Вячеслав Дорот
Федор Новиков

ТОЛКОВЫЙ СЛОВАРЬ

СОВРЕМЕННОЙ КОМПЬЮТЕРНОЙ ЛЕКСИКИ

Свыше 2500 терминов и словосочетаний

3-е издание

bhv®

Вячеслав Дорот
Федор Новиков

ТОЛКОВЫЙ СЛОВАРЬ

СОВРЕМЕННОЙ КОМПЬЮТЕРНОЙ ЛЕКСИКИ

3-е издание,
переработанное и дополненное

Санкт-Петербург
«БХВ-Петербург»
2004

УДК 681.3.06
ББК 32.973я26
Д69

Дорот В. Л., Новиков Ф. А.

Д69 Толковый словарь современной компьютерной лексики. — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2004. — 608 с.: ил.

ISBN 5-94157-491-6

Словарь является пособием, сочетающим толковательные, справочные и обучающие функции. Содержит определения, объяснения, примеры использования и английские эквиваленты свыше 2500 основных терминов и словосочетаний. Рассмотрены следующие темы: общие сведения об информации, о данных и их обработке; типы, модели, архитектура и основные параметры компьютеров; внешние устройства; операционные системы; методология, языки и системы программирования; базы данных и знаний; мультимедиа; работа с вычислительной системой и пользовательский интерфейс; сети ЭВМ и средства компьютерного общения. Многие статьи содержат примеры и иллюстрации. При отборе материалов авторы ориентировались в основном на пользователей персональных компьютеров, работающих с наиболее популярными приложениями Windows. Изложение вопросов, связанных с общей методикой программирования, сопровождается примерами на наиболее распространенных языках: Паскаль, Фортран, Си.

Для широкого круга пользователей

УДК 681.3.06
ББК 32.973я26

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Евгений Рыбаков*
Зав. редакцией	*Григорий Добин*
Редактор	*Юлия Гомулина*
Компьютерная верстка	*Натальи Смирновой*
Корректор	*Наталия Першакова*
Дизайн обложки	*Игоря Цырульникова*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.09.04.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 49,02.
Тираж 3000 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02
от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в ОАО "Техническая книга"
190005, Санкт-Петербург, Измайловский пр., 29.

ISBN 5-94157-491-6

© Дорот В. Л., Новиков Ф. А., 2004
© Оформление, издательство "БХВ-Петербург", 2004

Содержание

Предисловие **1**

Для кого и для чего нужен этот словарь 1
Как пользоваться словарем 3
Шрифтовые выделения, используемые в словарных статьях 7

Тематический указатель **9**

1. Общие сведения о компьютерах 9
   1.1. Автоматизация человеческой деятельности 9
   1.2. Обработка информации 11
   1.3. Представление данных 13
   1.4. Международные организации и стандарты 15
   1.5. Единицы измерения 17
2. Аппаратное обеспечение 18
   2.1. Компьютер и его структура 18
   2.2. Виды компьютеров 20
   2.3. Процессор 21
   2.4. Память 22
   2.5. Устройства ввода/вывода 25
   2.6. Носители информации 26
   2.7. Периферийное оборудование 27
   2.8. Устройства связи 28
   2.9. Компьютерные сети 29
3. Программное обеспечение 30
   3.1. Вычислительные процессы 30
   3.2. Архитектура программных систем 32
   3.3. Операционные системы 34
   3.4. Системы управления базами данных 39
   3.5. Текстовые процессоры 41
   3.6. Компьютерная графика 42
   3.7. Программное обеспечение сетей 44
   3.8. Различные приложения 45
4. Программирование 47
   4.1. Формальные языки 47
   4.2. Типы и структуры данных 50
   4.3. Программные конструкции 52
   4.4. Методология программирования 56
   4.5. Инструментальные средства программирования 59

5. Работа с вычислительной системой 62
5.1. Работа с персональным компьютером 62
5.2. Работа с компьютерной сетью 64
5.3. Управление файлами 66
5.4. Интерфейс пользователя 68
6. Компьютерные технологии 69
6.1. Компьютерная телефония 69
6.2. Компьютерная коммерция 70

А 71
Б 100
В 115
Г 146
Д 159
Е 186
Ж 186
З 187
И 202
К 228
Л 266
М 277
Н 308
О 315
П 354
Р 405
С 423
Т 459
У 475
Ф 486
Х 501
Ц 501
Ч 509
Ш 512
Щ 516
Э 516
Я 526

C 532
I 533
J 533
M 533
P 534
R 534
S 534
U 535
V 535
W 536

Англо-русский указатель терминов 537

Список литературы 601
Словари и справочные пособия 601
Учебные пособия, популярная и специальная литература 602

Предисловие

Уважаемый читатель! Вашему вниманию предлагается третье (исправленное и существенно дополненное) издание "Толкового словаря современной компьютерной лексики".

Для кого и для чего нужен этот словарь

Процессы развития компьютерной науки и техники столь стремительны, что русский язык постоянно пополняется новыми компьютерными терминами, постепенно освобождаясь от устаревших и часто изменяя значения уже сложившихся. Поэтому многие словари перестают удовлетворять пользователей вскоре после их издания. Среди наименее подверженных старению толковых словарей большинство ориентированы в основном на специалистов, занимающихся обработкой данных и программированием, и содержат, как правило, краткие определения, позволяющие начинающему пользователю компьютера уловить лишь общий смысл специальных терминов. Некоторые словари прежде всего отражают англо-американскую лексику, а не русскую, и также рассчитаны на специалистов — переводчиков и программистов. Кроме того, многие пояснения в этих словарях не дают полного толкования понятия, а только выделяют его важнейшие особенности.

Предлагаемый словарь рассчитан не только на специалистов, но и на школьников, студентов, аспирантов, преподавателей и начинающих пользователей вычислительной техники.

В настоящее время органами управления народным образованием рекомендовано к использованию в школах и других средних учебных заведениях несколько разных учебников по курсу "Основы информатики и вычислительной техники". Как показало сравнение этих пособий, школе не удалось создать единую программу, которая была бы реализована в каждом учебнике. Предлагаемый словарь охватывает материал всех школьных учебников по информатике и может служить пособием, по-своему дополняющим каждый из них.

В отличие от школьников студенты вузов изучают программирование и применение вычислительной техники на конкретных языках программирования высокого уровня и в основном на персональных компьютерах. Для них в словаре имеются сведения по представлению данных в компьютере, определения и описания основных синтаксических конструкций языков Паскаль, Фортран и Си. Статьи по методологии программирования, например, статьи "СТРУКТУРНОЕ ПРОГРАММИРОВАНИЕ", "МОДУЛЬНОЕ

ПРОГРАММИРОВАНИЕ", "ОТЛАДКА ПРОГРАММЫ" содержат советы, как писать хорошие программы по системе: алгоритм — язык программирования — стиль программирования, как их отлаживать и тестировать. На практических занятиях студентам пригодятся сведения об *операционных системах*, *системах программирования и стандартных программных средствах*. Кроме того, словарь поможет студентам в изучении всех предметов, так или иначе связанных с компьютером.

Студенты компьютерных специальностей найдут в словаре определения и объяснения многих профессиональных терминов и понятий, таких как *виртуальная память*, *обратное проектирование*, *макросредства*, *файловая система* и т. п., с которыми приходится сталкиваться при чтении учебников, пособий и документаций по различным разделам компьютерной науки, техники и технологии. Такие издания принадлежат как отечественным, так и зарубежным авторам, и, естественно, в них имеются разночтения, разобраться в которых также поможет словарь.

Быстро растет число людей, имеющих дома персональный компьютер или желающих его приобрести. К сожалению, в популярных книгах они не могут найти ответы на многие практические вопросы и вынуждены читать специальную литературу. При этом у начинающих пользователей персональных компьютеров и тех, кто заинтересовался увлекательным компьютерным делом, появляется немало терминологических вопросов. Кроме того, начинающий пользователь, чтобы найти нужную информацию, вынужден просматривать много пособий и справочников. Предлагаемый словарь, несомненно, будет полезен этому кругу читателей, т. к. сочетает в себе толковательные, справочные и обучающие функции и содержит в концентрированном виде сведения, относящиеся к современному аппаратному и программному обеспечению (архитектуре и конфигурации компьютеров, пользовательскому интерфейсу, текстовым редакторам, базам данных, компьютерным сетям, компьютерной телефонии, компьютерной коммерции и т. п.). Словарь окажет помощь аспирантам, преподавателям, специалистам и переводчикам, поскольку отражает состояние терминологии, сложившееся к 2004 г., и содержит определения, объяснения, примеры использования и английские эквиваленты свыше 2500 компьютерных терминов и словосочетаний. Кроме того, в конце книги дополнительно расположен алфавитный перечень английских эквивалентов представленных в словаре терминов и словосочетаний, позволяющий использовать книгу и как англо-русский словарь.

Словарь, помимо чисто познавательных, информативных функций, может способствовать самостоятельному обучению, для чего в нем широко используются перекрестные ссылки и указания, помогающие понять отношения между взаимосвязанными понятиями или различными аспектами одного понятия. С этой целью в словарь включен систематический указатель, в котором термины сгруппированы по темам.

Отбор материала производился из определенного круга изданных не ранее 1991 г. источников, которые указаны в списке использованной литературы. Среди них:

- наиболее распространенные учебники и учебные пособия по информатике, программированию, вычислительной технике и компьютерным сетям, а также книги для начинающих пользователей компьютера, которыми пользуются школьники и студенты;
- фундаментальная научно-техническая литература;
- документация к широко распространенным программным продуктам;
- электронные версии предметных указателей и глоссариев ко всем книгам по компьютерной тематике, изданным за последние несколько лет в издательстве "БХВ-Петербург"[1].

Отбирая материал, авторы старались отдавать предпочтение русским терминам, по возможности избегая англицизмов, не всегда корректных переводов англоязычных терминов и аббревиатур, которыми так насыщен проникающий в научную литературу программистский жаргон. В словарь попали лишь некоторые широко употребляемые в книгах и устной речи специалистов элементы такого жаргона, например, "горячие клавиши". Многие определения уже сложившихся терминов взяты из указанных в списке использованной литературы толковых словарей. При этом некоторые определения были сознательно упрощены или расширены ради простоты восприятия новичками. Наряду с устоявшимися, в книгу включены понятия, которые только начинают входить в широкий компьютерный обиход. Это термины, связанные с искусственным интеллектом, визуальным моделированием, сетью Интернет, компьютерной телефонией и мультимедиа.

Как пользоваться словарем

В словаре имеются *предисловие*, которое вы сейчас читаете, *тематический указатель компьютерных терминов*, *основная часть*, которая представляет собой перечень всех терминов с объяснением их значений, *указатель английских компьютерных терминов с их русскоязычными эквивалентами и список литературы*. Отобранные для словаря термины являются в настоящее время базовыми, наиболее важными в понятийной системе компьютерной терминологии, употребляемой в современной литературе.

---

[1] Авторы выражают глубокую благодарность издательству "БХВ-Петербург" за предоставление этих материалов.

Термины в словаре представлены в строго *алфавитном* порядке с учетом всех входящих символов, например:

АБОНЕНТ СЕТИ

АБСОЛЮТНЫЙ АДРЕС

АБСТРАКТНЫЙ КЛАСС

АБСТРАКТНЫЙ ТИП ДАННЫХ

Термины, начинающиеся с английских букв, например WEB-САЙТ, располагаются после всех "чисто русских" терминов в порядке сначала английского, а затем русского алфавитов, например:

...

WEB-САЙТ

WEB-СТРАНИЦА

Если термин состоит из нескольких слов, искать его следует, как видим, по первому слову.

При наличии у термина двух или более значений каждое из них выделяется цифрой, за которой следует определение:

АБОНЕНТ СЕТИ. 1. Пользователь, имеющий доступ к ресурсам компьютерной сети или к сети компьютерной связи. 2. Терминал, компьютер или рабочая станция, подключенные к вычислительной сети или сети компьютерной связи.

Если заголовочный термин встречается в статье повторно, он обозначается начальными буквами тех слов, из которых состоит:

АБСТРАКТНЫЙ ТИП ДАННЫХ. Понятие, введенное в конце 70-х годов в теоретических исследованиях по языкам программирования. А. т. д. рассматривается как множество значений и множество операций над этими значениями.

Словарные статьи можно разделить в зависимости от широты и глубины содержащихся в них сведений на общие, частные и отсылочные. К *общим* будем относить статьи, содержащие информацию по какому-либо широкому понятию, требующему для своего раскрытия привлечения многих данных и, как следствие этого, значительного числа других компьютерных терминов, например: **АВТОМАТИЗИРОВАННАЯ ОБУЧАЮЩАЯ СИСТЕМА**, **ВИРТУАЛЬНАЯ ПАМЯТЬ**, **ЗАГРУЗКА ОПЕРАЦИОННОЙ СИСТЕМЫ**.

К *частным* относятся словарные статьи, в которых объясняются компьютерные термины без привлечения или с малым привлечением других терминов:

АДРЕСНАЯ ШИНА. Шина или часть шины, предназначенная для передачи адреса.

Наконец, к *отсылочным* относятся словарные статьи, в которых не раскрывается или недостаточно раскрывается заголовочный термин, однако содержится отсылка на другой термин, где данное понятие толкуется с достаточной полнотой:

АРИФМЕТИЧЕСКОЕ УСТРОЙСТВО. То же, что *арифметико-логическое устройство.*

АРГУМЕНТ. Выражение, задающее объект или значение при обращении к процедуре, функции или макрокоманде, от которого зависит результат ее выполнения. См. *фактический параметр.*

В словаре показываются также системные отношения между компьютерными терминами. Сюда входят:

1. *Видородовые* связи, существующие прежде всего между заголовочным (видовым) термином и более широким (родовым) понятием (определителем), которое указывается в определении:

   ГЛАВНАЯ ПРОГРАММА. *Программа*, выполняемая первой и управляющая вызовом подпрограмм.

   ДИСТРИБУТИВ. *Программный продукт* в виде, поставляемом производителем (чаще всего на компакт-дисках).

2. *Синонимические* связи имеют место между терминами, называющими одно и то же понятие. Если синонимы стоят непосредственно при заголовочном термине, они выделяются полужирными строчными буквами и разделяются запятой:

   БЛОК ПРОГРАММЫ, **программный блок**, **блок**.

   Если при заголовочном слове дается только отсылка на синоним, объясняемый на своем алфавитном месте, используется помета *то же, что.*

   БЛОК [block]. 1. То же, что *аппаратный блок*. 2. То же, что *блок данных.* 3. То же, что *блок программы.*

3. *Словообразовательные* связи предполагают отношения между производными терминами и теми, от которых они образованы; обозначаются посредством выделения последних в определении курсивом:

   АРИФМЕТИКО-ЛОГИЧЕСКОЕ УСТРОЙСТВО. Часть процессора, выполняющая над данными *арифметические*, *логические* и другие *машинные операции.*

   ПОДМЕНЮ. *Меню*, вызываемое выбором пункта меню вышележащего уровня.

4. *Антонимические* связи устанавливаются между терминами, противопоставляемыми по значению; обозначаются пометой *противоп.* (при противопоставлении по всем параметрам) или конструкцией *чему противопоставляется что* (при логическом противопоставлении лишь по некоторым параметрам):

   ВЕРТИКАЛЬНОЕ МЕНЮ. Меню с вертикальным расположением пунктов меню (один под другим). На экране дисплея такое меню перекрывает выведенную ранее информацию. Противоп. *горизонтальное меню.*

   ЕСТЕСТВЕННЫЙ ЯЗЫК. Язык общения между людьми, правила которого основываются на текущем употреблении, а не на точном предварительном описании. Е. я. противопоставляется *формализованный язык.*

   Антонимы выделяются курсивом.

5. *Понятийные* связи предполагают отношения между сопредельными понятиями, когда одно из них нуждается в пояснении другим или когда одно из них входит в другое; обозначаются отсылкой *смотрите (см.)*, после которой указывается соответствующий термин, выделяемый курсивом:

   АНАЛОГОВОЕ УСТРОЙСТВО. Устройство, в котором информация представляется в виде непрерывно изменяющихся в некотором диапазоне физических величин, таких как электрическое напряжение или электрический ток. См. *аналоговый сигнал, аналоговая вычислительная машина.*

Для различения близких по звучанию или сходных по значению терминов используется отсылка *сравните (ср.)*, за которой следует указание на сравниваемый термин, выделяемый курсивом:

АВАРИЙНЫЙ ОСТАНОВ. Автоматическое прекращение работы компьютера из-за отказа аппаратных средств или программного обеспечения. Ср. *аварийное завершение.*

В результате структура словарной статьи получает следующий вид. В начале статьи находится заголовочный термин, который может быть одно- или многокомпонентным. Затем указываются синонимы, за синонимами в круглых скобках — краткая форма термина, за ней в квадратных скобках — английские эквиваленты. Основное место в словарной статье занимает объяснение термина; при этом отмечаются по возможности его системные отношения с другими терминами. В этих целях употребляются следующие пометы: *противоп.* при отсылке к словарным статьям, содержащим термины, противопоставляемые по значению, *см.* — при отсылке к словарным статьям, в которых находится дополнительная информация к тому, что содержится в определении данного термина, *ср.* — при сопоставлении или раз-

граничении близких по значению или созвучных терминов, *то же, что* — при тождестве значений у синонимов.

Шрифтовые выделения, используемые в словарных статьях

- **ПРОПИСНЫЕ ПОЛУЖИРНЫЕ БУКВЫ**, которыми выделяются заголовочные термины и их краткие формы.
- Строчные **полужирные буквы**, которыми выделяются синонимы, стоящие после заголовочного термина и английские эквиваленты.
- *Курсив*, которым выделяются: а) двусловные и многословные термины, толкуемые в словаре на своем алфавитном месте (однословные термины выделяются курсивом только в том случае, если они представляют родовое понятие по отношению к заголовочному слову, называющему видовое понятие); б) термины, в том числе однословные, находящиеся в словообразовательных и смысловых связях с заголовочным словом; в) термины, в том числе синонимы, стоящие за отсылками "то же, что", "см." и "ср."; г) антонимы, стоящие за ссылкой "противоп.".

Тематический указатель

1. Общие сведения о компьютерах

1.1. Автоматизация человеческой деятельности

АБСТРАКЦИЯ [abstraction]

АВТОМАТИЗИРОВАННАЯ СИСТЕМА [automated system]

АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ [computer-aided design (CAD)]

АВТОМАТИЧЕСКАЯ СИСТЕМА [automatic system]

АВТОМАТИЧЕСКОЕ УСТРОЙСТВО, автомат [automation]

АДАПТИВНАЯ СИСТЕМА [adaptive system]

АДАПТИВНОСТЬ [adaptivity]

АДРЕСАТ [target]

БЕСПЛАТНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, свободное программное обеспечение [public domain, freeware]

БЛОКИРОВКА ДАННЫХ [data interlock]

ВВОД ДАННЫХ [data input]

ВВОД/ВЫВОД [input-output]

ВСТРОЕННАЯ СИСТЕМА [embedded system]

ВСТРОЕННАЯ ЭВМ [build-in computer]

ВХОДНЫЕ ДАННЫЕ [input data]

ВЫХОДНЫЕ ДАННЫЕ [output data]

ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА [computer system]

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА [computer science, computing machinery]

ВЫЧИСЛИТЕЛЬНЫЙ КОМПЛЕКС, многомашинный вычислительный комплекс (ВК) [computer complex, multiple computer complex]

ВЫЧИСЛИТЕЛЬНЫЙ ПРОЦЕСС [calculation process]

ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР (ВЦ) [computer center]

ДИАЛОГ [dialog]

ДИАЛОГОВАЯ СИСТЕМА [dialog system, conversational system]

ДИСТРИБУТИВ [distribution kit]

ДОКУМЕНТ [document]

ЗАИНТЕРЕСОВАННЫЕ ЛИЦА [stockholders]

ЗАЩИТА ДАННЫХ [data protection]

ЗАЩИТА ОТ КОПИРОВАНИЯ [copy protection]

ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ [default value]

ИЗБЫТОЧНОСТЬ [redundancy]

ИНЖЕНЕР-ПРОГРАММИСТ [programmer-engineer]

ИНИЦИАЛИЗАЦИЯ [initialization]

ИНИЦИИРОВАНИЕ [initiation]

ИНТЕРАКТИВНАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА [interactive computer system]

ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР (ИВЦ) [information computer center]

ИСТОЧНИК ДАННЫХ [data source]

КОМПЛЕКС ПРОГРАММ [program complex, routine set]

КОМПЬЮТЕР [computer]

КОМПЬЮТЕРНАЯ ПЛАТФОРМА, ПЛАТФОРМА [computer platform, platform]

КОНЕЧНЫЙ ПОЛЬЗОВАТЕЛЬ [end user]

КОНТРОЛЬ КАЧЕСТВА [quality assurance (QA)]

КОНЦЕПТУАЛЬНЫЙ ПОИСК [concept search]

КРАКЕР [cracker]

МОРФОЛОГИЧЕСКИЙ ПОИСК [stemming]

МУЛЬТИМЕДИА [multimedia]

НАВИГАЦИОННЫЙ СПУТНИК [global positioning satellite (GPS)]

НЕЗАКОННОЕ КОПИРОВАНИЕ ПРОГРАММНЫХ СРЕДСТВ, несанкционированное копирование, пиратство [software piracy, piracy]

НЕСАНКЦИОНИРОВАННЫЙ ДОСТУП [unauthorized access, illegal access]

ОБРАБОТКА ДАННЫХ [data processing]

ОПЕРАТОР ЭВМ [operator]

ПОДДЕРЖКА [support]

ПОИСК В ОПРЕДЕЛЕННЫХ ПОЛЯХ [search by field]

ПОИСК ДАННЫХ [data search]

ПОИСК ПО КЛЮЧЕВЫМ СЛОВАМ [keyword search]

ПОИСК С ВОЗВРАТАМИ, обратное прослеживание [backtracking]

ПОИСКОВАЯ СИСТЕМА [search engine]

ПОЛНОТЕКСТОВАЯ ПОИСКОВАЯ СИСТЕМА [full-text search engine]

ПОЛЬЗОВАТЕЛЬ [client, user]

ПРЕДМЕТНАЯ ОБЛАСТЬ [domain]

ПРЕДМЕТНО-ОРИЕНТРОВАННЫЙ [domain specific]

ПРИЕМНИК ДАННЫХ (СООБЩЕНИЙ) [data (message) sink]

ПРОГРАММА [program]

ПРОГРАММИСТ [programmer]

ПРОГРАММНЫЙ ПРОДУКТ, программное изделие [program product, software]

ПСЕВДОНИМ [alias]

РАСПОЗНАВАНИЕ РЕЧИ [speech analysis, speech recognition]

РЕЗЕРВИРОВАНИЕ [standby]

РЕЗЕРВИРОВАНИЕ УСТРОЙСТВА [device allocation]

РЕЛЕВАНТНОСТЬ [relevancy]

РОБОТ ПОИСКОВОЙ СИСТЕМЫ [spider, grawler, bot, robot]

СИСТЕМА АВТОМАТИЗАЦИИ АВТОРСКОЙ РАБОТЫ, авторская система [authoring system]

СИСТЕМА АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ [automatic control system]

СИСТЕМА КОЛЛЕКТИВНОГО ПОЛЬЗОВАНИЯ, система коллективного доступа [multi-access system]

СИСТЕМНЫЙ АДМИНИСТРАТОР [system administrator]

СИСТЕМНЫЙ ИНЖЕНЕР [software engineer]

СИСТЕМНЫЙ ПРОГРАММИСТ [system programmer, software programmer, toolsmith]

СОПРОВОЖДЕНИЕ ПРОГРАММНОГО ИЗДЕЛИЯ [program product maintenance, support]

СТОП-СЛОВА [stop words]

ТВЕРДАЯ КОПИЯ [hard copy]

ТЕХНОЛОГИЯ [technology]

ТЕХНОЛОГИЯ PLUG AND PLAY [plug and play]

УНАСЛЕДОВАННЫЙ [legacy]

УПРАВЛЯЮЩИЙ СИГНАЛ [control signal]

УСЛОВНО-БЕСПЛАТНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ [Shareware]

ХАКЕР [hacker]

ЭЛЕКТРОННАЯ ВЫЧИСЛИТЕЛЬНАЯ МАШИНА (ЭВМ), [computer]

ЭЛЕКТРОННЫЙ ДОКУМЕНТ [electronic document]

1.2. Обработка информации

АЛГОРИТМ [algorithm]

АРИФМЕТИЧЕСКИЙ СДВИГ [arithmetic shift]

АРХИВ [archive]

АУДИОВИЗУАЛЬНЫЙ [audiovisual]

АУДИОДАННЫЕ, аудиоинформация [audio data, audio information]

БИНАРНАЯ ОПЕРАЦИЯ [binary operation]

БУЛЕВА АЛГЕБРА [Boolean algebra]

ВЕКТОР [vector]

ВИДЕОДАННЫЕ, видеоинформация [video data, video information]

ВИРТУАЛЬНЫЙ [virtual]

ГРАФИЧЕСКАЯ ИНФОРМАЦИЯ [graphic information, graphic data]

ДАННЫЕ [data]

ДВУМЕСТНАЯ ОПЕРАЦИЯ [dyadic operation]

ЕСТЕСТВЕННЫЙ ЯЗЫК [natural language]

ЗАПРОС [request, query]

ЗНАК [character, symbol]

ЗНАЧЕНИЕ [value]

ИЕРАРХИЯ [hierarchy]

ИМПЛИКАЦИЯ [implication]

ИМПОРТ [import]

ИНВЕРТИРОВАТЬ [invert]

ИНТЕРФЕЙС [interface]

ИНФОРМАТИКА [informatics, computer science]

ИНФОРМАЦИОННАЯ СИСТЕМА [information system]

ИНФОРМАЦИОННАЯ СРЕДА [information environment]

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ [information technology]

ИНФОРМАЦИОННОЕ СЛОВО [information word]

ИНФОРМАЦИЯ [information]

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, машинный интеллект [artificial intelligence, machine intelligence]

ИСХОДНЫЕ ДАННЫЕ [source data]

КИБЕРНЕТИКА [cybernetics]

КОЛОНКА, столбец [column]

КОНКАТЕНАЦИЯ [concatenation]

ЛЕКСИКОН ПРОГРАММИРОВАНИЯ [programming lexicon]

ЛОГИЧЕСКОЕ СЛОЖЕНИЕ, логическое "или", дизъюнкция [logical addition, logical "OR", dizjunction]

ЛОГИЧЕСКОЕ УМНОЖЕНИЕ, логическое "И", конъюнкция [logical multiplication, logical "AND", conjunction]

МЕТАДАННЫЕ [metadata]

МАШИНА ТЬЮРИНГА [Turing machine]

ОБМЕН ДАННЫМИ, обмен [exchange]

ОБРАБОТКА ЗАПРОСА [query processing]

ОБРАБОТКА ИНФОРМАЦИИ [information processing]

ОТНОШЕНИЕ [relation, relationship]

ПАРАЛЛЕЛИЗМ [concurrency]

ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ [parallel computing, parallel calculations]

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ [parallel algorithm]

ПОТОК ДАННЫХ [data flow, dataflow, data stream]

ПОТОК КОМАНД [instruction stream]

ПОТОМОК [descendant]

ПРЕДОК [ancestor]

ПРЕОБРАЗОВАТЕЛЬ, конвертор [converter]

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ [software]

ПРОГРАММНЫЕ СРЕДСТВА [software]

ПРОТОКОЛ [protocol]

СООБЩЕНИЕ [message]

СОРТИРОВКА ДАННЫХ, сортировка, упорядочение [data sorting, sort]

СТИРАНИЕ [erase]

ТЕКСТ [text]

ТЕКУЩЕЕ СОСТОЯНИЕ [current state]

УДАЛЕНИЕ [delete]

ЦИФРОВАЯ ПОДПИСЬ [digital signature]

ШИФРАТОР [scrambler, encipherer, encoder, encryptor]

ШИФРОВАНИЕ, кодирование [enciphering, encoding, encryption]

ШИФРОВАНИЕ С ЗАКРЫТЫМ КЛЮЧОМ, симметричное шифрование [secret key enciphering, secret key encryption, symmetric enciphering, symmetric encryption]

ШИФРОВАНИЕ С ОТКРЫТЫМ КЛЮЧОМ, несимметричное шифрование [open key enciphering, open key encryption, RSA enciphering, RSA encryption]

1.3. Представление данных

АДАПТИВНАЯ ДИФФЕРЕНЦИАЛЬНАЯ ИМПУЛЬСНО-КОДОВАЯ МОДУЛЯЦИЯ (АДИКМ) [adaptive differential pulse code modulation (ADPCM)]

БИТ ЧЕТНОСТИ [parity bit]

БЛОК ДАННЫХ, блок [data block, block]

ВЕРНЫЕ ЗНАЧАЩИЕ ЦИФРЫ [right significant digits]

ВИДЕОКОДЕК [videocodec]

ВОКОДЕР [vocoder]

ВОСЬМЕРИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ [octal number system]

ВОСЬМЕРИЧНОЕ ЧИСЛО [octal number]

ДВОИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ [binary number system]

ДВОИЧНОЕ ПРЕДСТАВЛЕНИЕ [binary-coded representation]

ДВОИЧНОЕ ЧИСЛО [binary number]

ДВОИЧНЫЙ КОД [binary code]

ДВОИЧНЫЙ РАЗРЯД [binary digit, bit]

ДВОЙНОЕ СЛОВО [double word]

ДЕСЯТИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ [decimal number system]

ДЕСЯТИЧНОЕ ПРЕДСТАВЛЕНИЕ [decimal representation]

ДИАПАЗОН ИЗМЕНЕНИЯ ЧИСЕЛ [number range]

ЗНАЧАЩАЯ ЦИФРА, значащий разряд [significant digit]

КОД [code]

КОДИРОВАНИЕ СИМВОЛОВ [character encoding]

КОДИРОВАНИЕ С ОГРАНИЧЕНИЕМ ДЛИНЫ ПОЛЯ ЗАПИСИ, RLL-кодирование [limited run-length encoding (RLL encoding)]

КОДОВАЯ СТРАНИЦА [code page]

КОМПРЕССОР-ДЕКОМПРЕССОР, кодек [compressor-decompressor, codec]

КОНТРОЛЬ ПО ЧЕТНОСТИ, контроль на четность, контроль четности [parity check]

МАНТИССА [fractional part, fraction]

МАСКА [mask]

МАШИННЫЙ НОЛЬ [computer zero]

МОДИФИЦИРОВАННАЯ АЛЬТЕРНАТИВНАЯ КОДИРОВКА ГОСТА [modified alternative code]

НОРМАЛИЗОВАННАЯ ФОРМА [normalized form]

НОРМАЛИЗОВАННОЕ ЧИСЛО [normalized number]

ПЕРЕПОЛНЕНИЕ [overflow, jverrun]

ПОЗИЦИОННАЯ СИСТЕМА СЧИСЛЕНИЯ [positional number system]

ПОЛЕ ДАННЫХ, поле [data field, field]

ПОЛУЛОГАРИФМИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ ЧИСЕЛ [floating-point number representation]

ПОРЯДОК ЧИСЛА [exponent]

ПРЕДСТАВЛЕНИЕ ДАННЫХ [data representation]

ПРЕДСТАВЛЕНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ТОЧКОЙ [floating-point representation]

РАЗРЯДНОСТЬ [capacity]

СЖАТИЕ ДАННЫХ, уплотнение данных [data compression, data compaction]

СИМВОЛ [character, symbol]

СИСТЕМА СЧИСЛЕНИЯ [number system]

СЛОВО [word]

УПАКОВЫВАТЬ [pack]

ФОРМАТИРОВАНИЕ ДАННЫХ [data formatting]

ЦИФРА [digit]

ЦИФРОВАЯ ФОРМА ДАННЫХ [digital data form]

ЦИФРОВОЙ КОД [numeric code]

ШЕСТНАДЦАТЕРИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ [hexadecimal number system]

ЭЛЕМЕНТ ДАННЫХ, элементарное данное [data item, item]

1.4. Международные организации и стандарты

АМЕРИКАНСКИЙ НАЦИОНАЛЬНЫЙ ИНСТИТУТ СТАНДАРТОВ [American National Standards Institute (ANSI)]

АМЕРИКАНСКИЙ СТАНДАРТНЫЙ КОД ОБМЕНА ИНФОРМАЦИЕЙ [American standard code for information interchange (ASCII)]

АРХИТЕКТУРА EISA [extended industry standard architecture (EISA)]

АССОЦИАЦИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ [association for computing machinery (ACM)]

АССОЦИАЦИЯ СТАНДАРТОВ ПО ВИДЕОТЕХНИКЕ, индустриальный консорциум VESA, ассоциация VESA [Video Electronics Standards Association (VESA)]

ГРУППА MPEG, группа экспертов в области движущихся изображений, экспертная группа по кинематографии [Moving Pictures Expert Group (MPEG)]

ГРУППА ОБЪЕДИНЕННЫХ ЭКСПЕРТОВ ПО ФОТОГРАФИИ, группа JPEG [Joint Pictures Entertainment Group (JPEG)]

ЕВРОПЕЙСКАЯ ЛАБОРАТОРИЯ ЯДЕРНЫХ ИССЛЕДОВАНИЙ (ЦЕРН) [European Particle Physics Laboratory, франц: Conseil Europeen pour la Recherche Nucleaire (CERN)]

ИНЖЕНЕРНЫЙ КОНСОРЦИУМ РАЗРАБОТЧИКОВ СТАНДАРТОВ ИНТЕРНЕТА, Инженерная комиссия Интернета, Инженерные силы Интернета, комитет IETF [Internet Engineering Task Force (IETF)]

ИНСТИТУТ ИНЖЕНЕРОВ ПО ЭЛЕКТРОТЕХНИКЕ И РАДИОЭЛЕКТРОНИКЕ (ИИЭР США) [Institute of Electrical and Electronics Engineers (IEEE)]

ИНСТИТУТ ТЕХНОЛОГИЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ [software engineering institute (SEI)]

ИНТЕГРИРОВАННАЯ ЦИФРОВАЯ СЕТЬ СВЯЗИ, сеть ISDN [integrated services digital network (ISDN)]

ИНТЕРФЕЙС ДЛЯ ПЕРЕДАЧИ ДАННЫХ ПО ВОЛОКОННО-ОПТИЧЕСКИМ КАНАЛАМ, стандарт FDDI [fiber distributed data interface (FDDI)]

КОДАСИЛ [CODASYL]

КОНСОРЦИУМ WORLD WIDE WEB [World Wide Web Consortium (WWWC, W3C)]

КОНСОРЦИУМ ПО ОБЪЕКТНЫМ ТЕХНОЛОГИЯМ, консорциум OMG [object management group (OMG)]

МЕЖДУНАРОДНАЯ АССОЦИАЦИЯ ПРОИЗВОДИТЕЛЕЙ КАРТ ПАМЯТИ ДЛЯ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ [Personal Computer Memory Card International Association (PCMCIA)]

МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ [International Standards Organization (ISO)]

МЕЖДУНАРОДНЫЙ КОНСУЛЬТАТИВНЫЙ КОМИТЕТ ПО ТЕЛЕГРАФИИ И ТЕЛЕФОНИИ [Consultative Committee International for Telephone and Telegraph (CCITT)]

МЕЖДУНАРОДНЫЙ СОЮЗ ТЕЛЕКОММУНИКАЦИЙ [International Telecommunication Union (ITU)]

МОДЕЛЬ OSI/ISO, модель взаимодействия открытых систем [International Organization for Standardization/Open System Interconnection model (OSI/ISO model)]

МОДЕЛЬ ЗРЕЛОСТИ ВОЗМОЖНОСТЕЙ, модель CMM [capability maturity model (CMM)]

РАСШИРЕННАЯ АРХИТЕКТУРА ПРОМЫШЛЕННОГО СТАНДАРТА, архитектура EISA [extended industry standard architecture (EISA)]

СЕАНСОВЫЙ УРОВЕНЬ [session layer]

СЕТЕВОЙ УРОВЕНЬ [network layer]

СПЕЦИФИКАЦИЯ IrDA [Infrared Data Association (IrDA)]

СПЕЦИФИКАЦИЯ ИНТЕРФЕЙСОВ СЕТЕВЫХ УСТРОЙСТВ, сетевой стандарт NDIS [network device interface specification (NDIS)]

СТАНДАРТЫ EDI [electronic data interchange (EDI)]

СТАНДАРТЫ H.323 [H.323]

СТАНДАРТЫ IEEE [IEEE standards]

СТАНДАРТЫ RFC [RFC standards, RFC documents]

СТАНДАРТ ADPCM [ADPCM]

СТАНДАРТ MIME [MIME]

СТАНДАРТ MPEG [MPEG]

СТАНДАРТ UNICODE [Unicode]

СТАНДАРТ XMI [XML Metadata Interchange]

ТЕЛЕКОММУНИКАЦИОННЫЙ СЕКТОР МЕЖДУНАРОДНОГО СОЮЗА ЭЛЕКТРОСВЯЗИ [Internacional Telecommunication Union — Telecommunication standardization sector (ITU-T)]

ТРАНСПОРТНЫЙ УРОВЕНЬ [transport layer]

УНИВЕРСАЛЬНАЯ ПОСЛЕДОВАТЕЛЬНАЯ ШИНА, стандарт USB [universal serial bus (USB)]

УПРАВЛЕНИЕ ДОСТУПОМ К СРЕДЕ, подуровень MAC [media access control (MAC)]

УПРАВЛЕНИЕ ЛОГИЧЕСКОЙ СВЯЗЬЮ, подуровень LLC [logical link control (LLC)]

УРОВЕНЬ КАНАЛА, канальный уровень [data link layer]

УРОВЕНЬ ПРИЛОЖЕНИЙ, прикладной уровень [application layer]

ФОНД СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ [Free Software Foundation (FSF)]

ФОРМАТ AVI [Audio Video Interleave (AVI)]

ФОРМАТ BMP [format BMP (сокращение от англ. bitmap) (BMP)]

ФОРМАТ GIF, формат обмена графическими данными [graphics interchange format (GIF)]

ФОРМАТ JPEG [format JPEG (JPEG)]

ФОРМАТ MPEG [format MPEG (MPEG)]

ФОРМАТ PCX [format PCX (PCX)]

ФОРМАТ RTF, расширенный текстовый формат [rich text format (RTF)]

ФОРМАТ TIFF, тегированный формат файлов изображений [tagged image file format, format TIFF, TIFF]

ФОРМАТ БУМАГИ [paper size]

1.5. Единицы измерения

БАЙТ (Б) [byte]

БИТ [bit]

БОД [baud]

ГЕРЦ (Гц) [hertz (Hz)]

ГИГАБАЙТ, Гбайт [gigabyte]

ГИГАФЛОПС, Гфлопс [GigaFLOPS, GFLOPS]

ДЮЙМ [inch, "]

ЕДИНИЦА ФУНКЦИОНАЛЬНОГО РАЗМЕРА [functional point (FP)]

КИЛОБАЙТ (Кбайт, Кб, К) [kilobyte (Kbyte, Kb, K)]

КИЛОБИТ В СЕКУНДУ (Кбит/с) [Kilobit per second (Kbit/s)]

КИЛОГЕРЦ (кГц) [kilohertz (kHz)]

КОЛИЧЕСТВО ПИКСЕЛОВ НА ДЮЙМ [pixels per inch (ppi)]

КОЛИЧЕСТВО ТОЧЕК НА ДЮЙМ [dots per inch (DPI)]

МЕГАБАЙТ (Мбайт, Мб, М) [megabyte]

МЕГАБАЙТ В СЕКУНДУ (Мбайт/с) [megabytes per second (MBps)]

МЕГАБИТ В СЕКУНДУ (Мбит/с) [megabits per second (Mbps, Mbit/s)]

МЕГАГЕРЦ (МГц) [megahertz (MHz)]

МЕГАФЛОПС, Мфлопс [MegaFLOPS, MFLOPS]

МИКРОСЕКУНДА (мксек, мкс) [microsecond (mcsec)]

МИЛЛИОН ОПЕРАЦИЙ В СЕКУНДУ [million instructions per second (MIPS)]

НАНОСЕКУНДА (нсек, нс) [nanosecond (nsec)]

ПУНКТ [point]

ТВИП [twip]

ТЕРАБАЙТ, Тб, Тбайт [terabyte (TB)]

ТЕРАФЛОПС, Тфлопс [TeraFLOPS, TFLOPS]

ФЛОПС [floating-point operations per second (FLOPS)]

ЧИСЛО СТРОК КОДА [lines of code (LOC)]

2. Аппаратное обеспечение

2.1. Компьютер и его структура

АВТОНОМНОЕ УСТРОЙСТВО [autonomy device, autonomy unit]

АДАПТЕР [adapter]

АДАПТЕР PCMCIA [PCMCIA-adapter]

АНАЛОГОВАЯ ФОРМА ДАННЫХ [analog data form]

АНАЛОГОВОЕ УСТРОЙСТВО [analog device]

АНАЛОГОВЫЙ СИГНАЛ, непрерывный сигнал [analog signal, continuous signal]

АНАЛОГО-ЦИФРОВОЙ ПРЕОБРАЗОВАТЕЛЬ (АЦП) [analog-to-digital converter]

АППАРАТНОЕ ОБЕСПЕЧЕНИЕ, техническое обеспечение [hardware]

АППАРАТНОЕ ПРЕРЫВАНИЕ [hardware interrupt]

АППАРАТНЫЕ СРЕДСТВА, технические средства [hardware]

АППАРАТНЫЙ [hardwired]

АППАРАТНЫЙ БЛОК, блок [hardware block, block]

АППАРАТНЫЙ КЛЮЧ [hardware key]

АППАРАТУРА [hardware, equipment]

АРХИТЕКТУРА КОМПЬЮТЕРА [computer architecture]

АРХИТЕКТУРА ФОН НЕЙМАНА, фон Неймановская архитектура [von Neumann architecture]

БЛОК УПРАВЛЕНИЯ [control block]

БУФЕРНЫЙ ПРОЦЕССОР, препроцессор [front-end processor]

БЫСТРОДЕЙСТВИЕ [performance, speed]

БЫСТРОДЕЙСТВИЕ ЭВМ, производительность ЭВМ [computer speed, computer performance]

ВИДЕОСИСТЕМА КОМПЬЮТЕРА [video system]

ВРЕМЯ ТАКТА [loop time]

ДЕШИФРАТОР, декодер [decoder]

ДИСК [disk]

ЖЕСТКИЙ МАГНИТНЫЙ ДИСК, жесткий диск [hard disk]

ИЕРАРХИЯ ПАМЯТИ [memory hierarchy, storage hierarchy]

ИНТЕГРАЛЬНАЯ СХЕМА (ИС) [integrated circuit]

КЛАССИФИКАЦИЯ ФЛИННА, таксономия Флинна [Flynn classification]

КОДЕК MPEG [MPEG-codec]

КОДЕР-ДЕКОДЕР, кодек [coder-decoder, codec]

КОМПЬЮТЕР С ОДНОРОДНЫМ ДОСТУПОМ К ПАМЯТИ, UMA-компьютер [uniform memory access computer, UMA-computer]

КОНТРОЛЛЕР [controller]

КОНТРОЛЛЕР ЖЕСТКОГО ДИСКА [hard disk controller]

КОНФИГУРАЦИЯ [configuration]

КОНФИГУРАЦИЯ КОМПЬЮТЕРА [configuration]

ЛОКАЛЬНАЯ ШИНА [local bus]

МАТЕРИНСКАЯ ПЛАТА [mother board]

МИКРОСХЕМА [chip, microcircuit]

НАРАБОТКА НА ОТКАЗ, среднее время между отказами [mean time between failures (MTBF)]

ОБОРУДОВАНИЕ [equipment]

ПИКОВАЯ ПРОИЗВОДИТЕЛЬНОСТЬ КОМПЬЮТЕРА, теоретическая производительность компьютера [computer peak performance]

ПЛАТА [circuit board, circuit card]

ПРОИЗВОДИТЕЛЬ ОБОРУДОВАНИЯ [original equipment manufacturer (OEM)]

ПРОФИЛЬ УСТРОЙСТВА [device profile]

СИГНАЛ [signal]

СИСТЕМНАЯ ШИНА [bus]

СИСТЕМНЫЙ БЛОК [system block]

СЛОТ [slot]

СОКЕТ [socket]

ТАЙМЕР [timer, timer clock]

ТАКТ, время такта [loop, loop time]

ТАКТОВАЯ ЧАСТОТА [clock rate]

УСТРОЙСТВО, техническое устройство [device, unit]

УСТРОЙСТВО СОПРЯЖЕНИЯ [interface device]

ФУНКЦИОНАЛЬНАЯ СХЕМА ЭВМ [computer functional diagram]

ФУНКЦИОНАЛЬНОЕ УСТРОЙСТВО [functional device, functional unit]

ЦИФРОАНАЛОГОВЫЙ ПРЕОБРАЗОВАТЕЛЬ (ЦАП) [digital-to-analog converter (DAC)]

ШИНА [bus]

ШИНА ДАННЫХ [data bus]

ШИНА VLB, локальная шина VESA [VESA local bus (VLB)]

ЭЛЕМЕНТНАЯ БАЗА [integrated circuit]

ЭЛЕМЕНТ РАСШИРЕНИЯ [add-on]

ЭЛЕКТРОННАЯ СХЕМА [electronic circuit]

ЦИФРОВОЙ СИГНАЛ, дискретный сигнал [digital signal, discrete signal]

2.2. Виды компьютеров

АНАЛОГОВАЯ ВЫЧИСЛИТЕЛЬНАЯ МАШИНА (АВМ) [analog computer]

АНАЛОГО-ЦИФРОВАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА, гибридная вычислительная система [hybrid computer]

БЕОВУЛЬФ-КЛАСТЕР [Beowulf cluster]

БЛОКНОТНЫЙ КОМПЬЮТЕР [hand-held PC (HPC)]

ВСТРОЕННАЯ ЭВМ [build-in computer]

ВЫЧИСЛИТЕЛЬНАЯ МАШИНА, компьютер [computer]

ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА MPP, система MPP [massively parallel processors system, MPP system]

ВЫЧИСЛИТЕЛЬНЫЙ КЛАСТЕР, кластер [computer cluster, cluster]

ГЕТЕРОГЕННАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА, разнородная вычислительная система [heterogeneous computer system]

ГЛАВНАЯ ЭВМ, хост [host computer, host]

КАЛЬКУЛЯТОР [calculator]

КЛАСТЕРНАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА [cluster computer system]

КОМПЬЮТЕР СО СЛОЖНОЙ СИСТЕМОЙ КОМАНД, CISC-компьютер [complex instruction set computer (CISC)]

КОМПЬЮТЕР С УПРОЩЕННОЙ СИСТЕМОЙ КОМАНД, RISC-компьютер [reduced instruction set computer (RISC)]

ЛЭПТОП [laptop]

МЕТАКОМПЬЮТЕР [metacomputer]

МИКРОКАЛЬКУЛЯТОР [microcalculator]

МИКРОЭВМ [microcomputer]

МНОГОПРОЦЕССОРНАЯ ЭВМ, мультипроцессорная ЭВМ [multiprocessor computer]

МОБИЛЬНЫЙ КОМПЬЮТЕР [mobile computer, mobile PC]

МЭЙНФРЕЙМ [mainframe]

НАСТОЛЬНЫЙ КОМПЬЮТЕР [desktop computer]

НОУТБУК [notebook]

ОРГАНИЗАТОР ЛИЧНОЙ ДЕЯТЕЛЬНОСТИ [personal digital assistant (PDA)]

ПАРАЛЛЕЛЬНЫЙ КОМПЬЮТЕР [parallel computer]

ПЕН-КОМПЬЮТЕР [pen computer]

ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР, персональная ЭВМ (ПК, ПЭВМ) [personal computer (PC)]

ПЕРСОНАЛЬНЫЙ IBM-СОВМЕСТИМЫЙ КОМПЬЮТЕР (IBM-ПК) [personal IBM-compatible computer (IBM PC)]

ПОКОЛЕНИЯ ЭВМ [computer generations]

ПОРТАТИВНЫЙ КОМПЬЮТЕР [portable computer]

СИММЕТРИЧНЫЙ МНОГОПРОЦЕССОРНЫЙ КОМПЬЮТЕР, SMP-компьютер [symmetric multi processor computer, SMP-computer]

СУПЕРЭВМ, суперкомпьютер [supercomputer]

ТРАНСПЬЮТЕР [transputer]

ЦИФРОВАЯ ВЫЧИСЛИТЕЛЬНАЯ МАШИНА (ЦВМ) [digital computer]

ЦИФРОВОЕ УСТРОЙСТВО [digital device]

ХОСТ [host computer, host]

MIMD-КОМПЬЮТЕР [MIMD-computer]

MISD-КОМПЬЮТЕР [MISD-computer]

SIMD-КОМПЬЮТЕР [SIMD-computer]

SISD-КОМПЬЮТЕР [SISD-computer]

SMP-КОМПЬЮТЕР [SMP-computer]

2.3. Процессор

АРИФМЕТИКО-ЛОГИЧЕСКОЕ УСТРОЙСТВО (АЛУ), арифметическое устройство (АУ) [arithmetic and logical unit (ALU)]

АРХИТЕКТУРА CISC [complex instruction set computing (CISC)]

АРХИТЕКТУРА RISC [reduced instruction set computing (RISC)]

ВЕКТОРНАЯ КОМАНДА [vector instruction, array instruction]

ВЕКТОРНЫЙ ПРОЦЕССОР [vector processor]

КОЛИЧЕСТВО ТАКТОВ НА КОМАНДУ [cycles per instruction (CPI)]

ЗАЩИЩЕННЫЙ РЕЖИМ [protected mode]

МАТЕМАТИЧЕСКИЙ СОПРОЦЕССОР [floating-point processor, mathematical coprocessor, math coprocessor]

МАТРИЧНЫЙ ПРОЦЕССОР [array processor]

МИКРОКОМАНДА [microinstruction]

МИКРОПРОГРАММА [micro program]

МИКРОПРОГРАММИРОВАНИЕ [microprogramming]

МИКРОПРОГРАММНАЯ ПАМЯТЬ [micro program store, control memory]

МИКРОПРОЦЕССОР [microprocessor]

НЕПРИВИЛЕГИРОВАННЫЙ РЕЖИМ [slave mode]

ПОСТПРОЦЕССОР [back-end processor, postprocessor]

ПРОЦЕССОР [processor]

ПРОЦЕССОРНЫЙ ЭЛЕМЕНТ [processor unit]

ПРОЦЕССОР СО СВЕРХДЛИННЫМ КОМАНДНЫМ СЛОВОМ, VLIW-процессор [very large instruction word processor, VLIW-processor]

ПРОЦЕССОР ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ (ЦОС) [digital signal processor (DSP)]

ПРОЦЕССОР MMX [multimedia extension processor (MMX processor)]

ПРОЦЕССОР POWERPC [power performance chip, PowerPC (PPC)]

РЕАЛЬНЫЙ РЕЖИМ [real mode]

СИСТЕМА КОМАНД [instruction set]

СОПРОЦЕССОР [coprocessor]

СУММАТОР [accumulator]

СУПЕРСКАЛЯРНАЯ АРХИТЕКТУРА [super scalar architecture]

СУПЕРСКАЛЯРНЫЙ ПРОЦЕССОР [superscalar processor]

ТЕХНОЛОГИЯ MMX [MultiMedia Extension (MMX)]

УСТРОЙСТВО УПРАВЛЕНИЯ [control unit (CU)]

ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР (ЦП) [central processing unit (CPU)]

2.4. Память

АБСОЛЮТНЫЙ АДРЕС, физический адрес [absolute address, physical address]

АДРЕС [address]

АДРЕСАЦИЯ [addressing]

АДРЕС КОМАНДЫ [instruction address]

АДРЕСНАЯ ШИНА [address bus]

АДРЕСНОЕ ПРОСТРАНСТВО ЦЕНТРАЛЬНОГО ПРОЦЕССОРА [CPU address space]

АДРЕСНЫЙ РЕГИСТР [address register]

АДРЕС ОПЕРАНДА [source address]

БАЗОВЫЙ АДРЕС [base address]

БАЗОВЫЙ РЕГИСТР, регистр базы [base register]

БАНК ПАМЯТИ, банк, секция памяти [memory bank, bank, memory section]

БЛОК ПАМЯТИ [memory block, storage block]

БУФЕР [buffer]

БУФЕРИЗАЦИЯ [buffering]

БУФЕРНАЯ ПАМЯТЬ, буфер [buffer memory, buffer]

ВЕКТОРНЫЙ РЕГИСТР [vector register, V-register]

ВЕРХНЯЯ ПАМЯТЬ [upper memory blocks, UMB]

ВИДЕОПАМЯТЬ [video memory, video RAM, video storage]

ВНЕШНЯЯ ПАМЯТЬ [backing storage, external memory]

ВНУТРЕННЯЯ ПАМЯТЬ [Internal memory]

ВЫБОРКА [fetching]

ДЕЙСТВИТЕЛЬНЫЙ АДРЕС [effective address, executive address]

ДИНАМИЧЕСКАЯ ОБЛАСТЬ ПАМЯТИ, проблемная область памяти [dynamic storage area]

ДОПОЛНИТЕЛЬНАЯ ПАМЯТЬ [add-in memory]

ЕМКОСТЬ ПАМЯТИ, объем памяти [memory capacity]

ЕМКОСТЬ РЕГИСТРА, длина регистра, размер регистра [register capacity, register length, register size]

ЗАПОМИНАЮЩЕЕ УСТРОЙСТВО (ЗУ) [storage device]

ИНДЕКСНАЯ АДРЕСАЦИЯ, адресация с индексированием [indexed addressing]

ИНДЕКСНЫЙ РЕГИСТР, регистр индекса, регистр B, регистр BX [index register, B-register]

ИНФОРМАЦИОННАЯ ЕМКОСТЬ, информационный объем [information capacity]

ИСПОЛНИТЕЛЬНЫЙ АДРЕС [effective (executive) address]

КЛАСТЕР [cluster]

КМОП-ПАМЯТЬ, CMOS-память [CMOS memory, CMOS RAM]

КОМПЛЕМЕНТАРНАЯ СТРУКТУРА МЕТАЛЛ-ОКСИД-ПОЛУПРОВОДНИК, КМОП-структура, CMOS-структура [complementary metal-oxide semiconductor (CMOS)]

КОСВЕННАЯ АДРЕСАЦИЯ [indirect addressing]

КЭШИРОВАНИЕ [caching]

КЭШИРОВАНИЕ С ОБРАТНОЙ ЗАПИСЬЮ [write-back caching]

КЭШИРОВАНИЕ С ПРЯМОЙ ЗАПИСЬЮ [write-through caching]

КЭШ-ПАМЯТЬ [cache memory]

ЛОГИЧЕСКИЙ АДРЕС, виртуальный адрес [logical address, virtual address]

ЛОГИЧЕСКАЯ ЗАПИСЬ [logical record]

МАССОВАЯ ПАМЯТЬ [mass storage]

МИКРОСХЕМА ПАМЯТИ [memory circuit]

МОДУЛЬ ПАМЯТИ [memory module]

НАСТРОЙКА АДРЕСОВ [address relocation]

НЕПОСРЕДСТВЕННЫЙ ОПЕРАНД [immediate operand]

НЕРЕЗИДЕНТНЫЙ ПУЛ [paged pool]

ОБЛАСТЬ ПАМЯТИ [storage area]

ОБНУЛЕНИЕ [zero filling]

ОБЩАЯ ПАМЯТЬ [shareable memory, shared memory]

ОПЕРАТИВНАЯ ПАМЯТЬ [main memory, main storage]

ОПЕРАТИВНАЯ ПАМЯТЬ ПЕРСОНАЛЬНЫХ IBM-СОВМЕСТИМЫХ КОМПЬЮТЕРОВ [IBM-PC main memory]

ОСНОВНАЯ ПАМЯТЬ [main memory, main storage]

ОТЛОЖЕННАЯ ЗАПИСЬ [lazy write]

ОТНОСИТЕЛЬНЫЙ АДРЕС [relative address]

ОТОБРАЖАЕМАЯ ПАМЯТЬ [expanded memory]

ПАМЯТЬ [memory, storage, store]

ПАМЯТЬ ПРОИЗВОЛЬНОГО ДОСТУПА, память прямого доступа [random access memory (RAM)]

ПАМЯТЬ ЭВМ [memory, storage, store]

ПОСЛЕДОВАТЕЛЬНЫЙ ДОСТУП [sequential access, serial access]

ПОСТОЯННОЕ ЗАПОМИНАЮЩЕЕ УСТРОЙСТВО (ПЗУ) [read only memory (ROM)]

ПРЕФИКС [prefix]

ПРОВЕРКА НА ЧЕТНОСТЬ [even parity]

ПРЯМАЯ АДРЕСАЦИЯ, непосредственная адресация [direct addressing, immediate addressing]

ПРЯМОЙ ДОСТУП [direct access]

ПУЛ ПАМЯТИ, пул [pool]

РАБОЧАЯ ОБЛАСТЬ [work area, workspace]

РАЗДЕЛ ДИСКА, раздел [partition]

РАЗДЕЛ ПАМЯТИ [division, partition]

РАЗДЕЛЯЕМАЯ ПАМЯТЬ, общая память [shareable memory, shared memory]

РАСПРЕДЕЛЕННАЯ ПАМЯТЬ [allocate memory]

РАСШИРЕННАЯ ПАМЯТЬ [extended memory (XMS)]

РЕГИСТР [register]

РЕГИСТР АДРЕСА, адресный регистр [address register]

РЕГИСТР БАЗЫ [base register]. То же, что базовый регистр

РЕГИСТР КОМАНД [instruction register]

РЕЗИДЕНТНЫЙ ПУЛ [nonpaged pool]

СБРОС КЭШ-ПАМЯТИ, сброс кэша [cache flushing]

СЕГМЕНТАЦИЯ [segmentation]

СЕГМЕНТ ПАМЯТИ, сегмент [memory segment]

СИСТЕМА УПРАВЛЕНИЯ ПАМЯТЬЮ [memory management system]

СМЕЩЕНИЕ [displacement, offset]

СТРАНИЦА ПАМЯТИ, страница [memory page, page]

ТОЧНАЯ ЗАПИСЬ [careful write]

ФИЗИЧЕСКАЯ ЗАПИСЬ [physical record]

ФИЗИЧЕСКАЯ ПАМЯТЬ [physical storage]

ФИЗИЧЕСКИЙ АДРЕС [physical address]

ЧЕРЕДУЕМАЯ ПАМЯТЬ [switching memory, flip-flops].

ЭЛЕКТРОННЫЙ ДИСК, виртуальный диск, псевдодиск [electronic disk, virtual disk, RAM disk]

ЯЧЕЙКА ПАМЯТИ [cell]

2.5. Устройства ввода/вывода

АДАПТЕР ДИСПЛЕЯ, видеоадаптер, видеокарта [display adapter, videoadapter, videocard]

АЛЬФА-КАНАЛ [alpha channel]

ВИДЕОТЕРМИНАЛ [video terminal]

ВИЗУАЛИЗАЦИЯ [rendering]

ВСТРОЕННЫЙ ИНТЕРФЕЙС НАКОПИТЕЛЕЙ, интерфейс IDE [integrated drive electronics, integrated device electronics (IDE)]

ВЫВОД ДАННЫХ, вывод [data output, output]

ГОЛОВКА ЗАПИСИ/ЧТЕНИЯ [read-write head]

ГРАФИЧЕСКИЙ АДАПТЕР [graphics adapter]

ГРАФИЧЕСКИЙ СОПРОЦЕССОР, графический акселератор [graphics coprocessor, graphics accelerator]

ДЖОЙСТИК [joystick]

ДИСКОВОД [disk drive]

ДИСПЛЕЙ [display]

ДИСПЛЕЙ НА ЖИДКИХ КРИСТАЛЛАХ, жидкокристаллический дисплей [liquid crystal display (LCD)]

ЗАПИСЬ [writing, record]

ЗНАКОГЕНЕРАТОР [character generator]

ЗНАКОМЕСТО [font reticle]

ИНТЕЛЛЕКТУАЛЬНЫЙ ТЕРМИНАЛ [intelligent terminal]

ИНФРАКРАСНЫЙ ИНТЕРФЕЙС [infrared interface, Ir interface]

ИНФРАКРАСНЫЙ ПОРТ [infrared port, Ir port]

КАЛИБРОВКА [calibration, tuning]

КАНАЛ ВВОДА/ВЫВОДА, канал [input-output channel, channel]

КЛАВИАТУРА [keyboard]

КОНСОЛЬ [console]

ЛИНЕЙНЫЙ ПРИНТЕР, устройство построчной печати, алфавитно-цифровое печатающее устройство (АЦПУ) [line printer (LPT)]

МОНИТОР [monitor]

МЫШЬ [mouse]

НАКОПИТЕЛЬ НА ГИБКИХ МАГНИТНЫХ ДИСКАХ (НГМД) [floppy disk drive]

НАКОПИТЕЛЬ НА МАГНИТНОЙ ЛЕНТЕ (НМЛ) [magnetic tape storage]

НАКОПИТЕЛЬ НА МАГНИТНЫХ ДИСКАХ (НМД) [magnetic disk storage]

ОБРАЩЕНИЕ К ВНЕШНЕМУ УСТРОЙСТВУ [device access]

ПАРАЛЛЕЛЬНЫЙ ПОРТ [parallel port]

ПЕРО [stylus]

ПОИСК ДОРОЖКИ, подвод головки [seek]

ПОРТ [port]

ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ [serial port]

ПРИВОД CD-ROM [CD-ROM drive, CD drive]

РАЗРЕШАЮЩАЯ СПОСОБНОСТЬ [resolution]

СВЕТОВОЕ ПЕРО [light pen]

СЕНСОРНЫЙ ЭКРАН [touch screen]

СТРИМЕР [streamer]

ТЕКСТОВЫЙ РЕЖИМ [character mode]

ТЕРМИНАЛ [terminal]

ТРЕКБОЛ, шаровой манипулятор [track ball]

УКАЗАТЕЛЬНОЕ УСТРОЙСТВО [pointing device]

УСТРОЙСТВО ВВОДА [input device]

УСТРОЙСТВО ВВОДА/ВЫВОДА [input-output device]

УСТРОЙСТВО ВЫВОДА [output device]

ЦИФРОВОЙ ФОТОАППАРАТ, цифровая камера, цифровая фотокамера [digital still camera, digital camera]

ЭКРАН [screen]

2.6. Носители информации

ВИДЕОДИСК [videodisk]

ВИНЧЕСТЕРСКИЙ ДИСК, винчестер [Winchester disk, hard disk]

ГЕОМЕТРИЯ ДИСКА [disk geometry]

ГИБКИЙ МАГНИТНЫЙ ДИСК, дискета [floppy disk, diskette]

ДИСК АВАРИЙНОГО ВОССТАНОВЛЕНИЯ [recovery disk]

ДИСК БЕРНУЛЛИ [Bernoulli disk]

ДОРОЖКА [track]

ЗАПОМИНАЮЩАЯ СРЕДА [storage medium]

ЗВУКОВАЯ ДОРОЖКА [audio track, sound track]

ЗОНА [region, band]

КОМПАКТ-ДИСК [compact disk (CD), CD-ROM]

ЛАЗЕРНЫЙ ДИСК, видеодиск, оптический диск, компакт-диск, [videodisk, optical disk, CD-ROM]

МАГНИТНЫЙ ДИСК [magnetic disk]

МАГНИТНАЯ КАРТА [magnetic card]

МАГНИТНАЯ ЛЕНТА (МЛ) [magnetic tape]

МАССИВЫ RAID [redundant array of independent disks (RAID)]

МОДИФИЦИРОВАННАЯ ЧАСТОТНАЯ МОДУЛЯЦИЯ (МЧМ) [modified frequency modulation (MFM)]

НАКОПИТЕЛЬ НА ЛАЗЕРНЫХ ДИСКАХ [laser disk drive]

НОСИТЕЛЬ ДАННЫХ [data medium]

ПАКЕТ МАГНИТНЫХ ДИСКОВ, пакет дисков [disk pack]

ПЕРФОКАРТА, перфорационная карта [card, punch card]

ПЛОТНОСТЬ ЗАПИСИ [packing density]

СЕКТОР [sector]

СМАРТ-КАРТА [smart card]

ФЛЭШ-ПАМЯТЬ [flash memory]

2.7. Периферийное оборудование

АКСЕССУАРЫ [accessory]

АЛФАВИТНО-ЦИФРОВОЕ ПЕЧАТАЮЩЕЕ УСТРОЙСТВО (АЦПУ) [line printer (LPT)]

ВНЕШНЕЕ ЗАПОМИНАЮЩЕЕ УСТРОЙСТВО (ВЗУ) [backing storage, external storage]

ВНЕШНЕЕ УСТРОЙСТВО [external device]

ГЕОМЕТРИЯ УСТРОЙСТВА, геометрия [device geometry, geometry]

ГРАФИЧЕСКИЙ ПЛАНШЕТ, цифровой планшет графического ввода, цифровой планшет [graphics tablet, digitizing tablet]

ГРАФОПОСТРОИТЕЛЬ, плоттер [plotter]

ДИГИТАЙЗЕР [digitizer]

ЗВУКОВАЯ КОЛОНКА [speaker]

ЗВУКОВАЯ ПЛАТА, звуковой адаптер, звуковая карта [sound board, sound card]

ИНТЕРФЕЙС МАЛЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ, интерфейс SCSI [small computer system interface (SCSI)]

КАРТРИДЖ [cartridge]

ЛАЗЕРНЫЙ ПРИНТЕР [laser printer]

МАТРИЧНЫЙ ПРИНТЕР [matrix printer]

МУЗЫКАЛЬНЫЙ СИНТЕЗАТОР [frequency modulation synthesizer]

ОБЩИЙ ПРИНТЕР, сетевой принтер [shared printer, network printer]

ПЕРИФЕРИЙНОЕ УСТРОЙСТВО [peripheral device]

ПРИНТЕР, печатающее устройство [printer]

ПРИНТЕР С СЕТЕВЫМ ИНТЕРФЕЙСОМ [network-interface printer]

ПРОГОН БУМАГИ [paper feed]

ПРОФИЛЬ ОБОРУДОВАНИЯ [equipment profile]

СКАНЕР [scanner]

СПИСОК СОВМЕСТИМОГО ОБОРУДОВАНИЯ [hardware compatibility list (HCL)]

СТРУЙНЫЙ ПРИНТЕР [ink-jet printer]

ТОНЕР [toner]

УСТРОЙСТВА SCSI [SCSI devices]

ЦИФРОВОЙ ИНТЕРФЕЙС МУЗЫКАЛЬНЫХ ИНСТРУМЕНТОВ [musical instrument digital interface (MIDI)]

2.8. Устройства связи

БРАНДМАУЭР [firewall]

ГОЛОСОВОЙ КАНАЛ [media stream]

ГОЛОСОВОЙ МОДЕМ [voice modem]

ДВУХКАНАЛЬНЫЙ КОНТРОЛЛЕР [dual channel controller]

КАНАЛ ПЕРЕДАЧИ ДАННЫХ [data link]

КАНАЛ СВЯЗИ [communication channel]

КВИТИРОВАНИЕ [handshaking]

КОНЦЕНТРАТОР, хаб [concentrator, hub]

ЛИНИИ ЗАПРОСА ПРЕРЫВАНИЙ, линии прерываний, линии IRQ [interrupt request lines (IRQ)]

ЛИНИЯ СВЯЗИ [link, circuit]

МАГИСТРАЛЬ [backplane, trunk, unibus]

МАРШРУТИЗАТОР [router]

МАРШРУТИЗАЦИЯ [routing]

МОДЕМ, модулятор-демодулятор [modem]

МОСТ [bridge]

ПЛАТА ИНТЕРФЕЙСА ISDN [ISDN interface card]

ПРОПУСКНАЯ СПОСОБНОСТЬ [bandwidth]

СЕГМЕНТ СЕТИ, сегмент [network segment, segment (SEG)]

СЕТЕВАЯ КАРТА, сетевая плата, сетевой адаптер [network card, network board]

СОКЕТ-КАНАЛ [socket]

УНИВЕРСАЛЬНЫЙ АСИНХРОННЫЙ ПРИЕМОПЕРЕДАТЧИК (УАПП) [universal asynchronous receiver-transmitter (UART)]

УПРАВЛЕНИЕ ПРОПУСКНОЙ СПОСОБНОСТЬЮ [bandwidth control]

ФАКС-АППАРАТ, факс-машина, факс [fax machine]

ФАКС-МОДЕМ [fax-modem]

ФАЙЛОВЫЙ СЕРВЕР, файл-сервер [file server]

ШИРИНА ПОЛОСЫ ПРОПУСКАНИЯ [bandwidth]

2.9. Компьютерные сети

АРХИТЕКТУРА СЕТИ, сетевая архитектура [network architecture]

АРХИТЕКТУРА ETHERNET [architecture Ethernet]

ВСЕМИРНАЯ ПАУТИНА [World Wide Web (WWW)]

ВЫЧИСЛИТЕЛЬНЫЙ УЗЕЛ, узел [node]

ГЛОБАЛЬНАЯ ВЫЧИСЛИТЕЛЬНАЯ СЕТЬ [global network]

ДОМАШНЯЯ СЕТЬ [home network]

ДРЕВОВИДНАЯ СЕТЬ, сеть древовидной топологии [tree network]

ДРЕВОВИДНАЯ ТОПОЛОГИЯ [tree topology]

ЗВЕЗДООБРАЗНАЯ СЕТЬ [star network]

ИНТЕРНЕТ [Internet]

ИНТРАНЕТ, интрасеть, корпоративная сеть [intranet]

КИБЕРПРОСТРАНСТВО [cyberspace]

КОЛЬЦЕВАЯ СЕТЬ, сеть кольцевой топологии [ring network]

КОЛЬЦЕВАЯ СЕТЬ С МАРКЕРНЫМ ДОСТУПОМ, кольцевая сеть с передачей маркера, маркерная кольцевая сеть [token ring network]

КОММУНИКАЦИОННЫЙ ДИАМЕТР [bandwidth, communication range]

КОМПЬЮТЕРНАЯ СВЯЗЬ [computer communication]

КОМПЬЮТЕРНАЯ СЕТЬ, вычислительная сеть, сеть ЭВМ [computer network]

ЛОКАЛЬНАЯ ВЫЧИСЛИТЕЛЬНАЯ СЕТЬ (ЛВС) [local area network (LAN)]

ОДНОРАНГОВАЯ ВЫЧИСЛИТЕЛЬНАЯ СЕТЬ [peer-to-peer network]

ПОРЯДОК УЗЛА

РАБОЧАЯ СТАНЦИЯ [work station]

РАДИАЛЬНАЯ СЕТЬ, звездообразная сеть, сеть звездообразной топологии [star network]

РЕГИОНАЛЬНАЯ ВЫЧИСЛИТЕЛЬНАЯ СЕТЬ [wide area network (WAN)]

СЕТЬ ДРЕВОВИДНОЙ ТОПОЛОГИИ [tree network]

СЕТЬ ЗВЕЗДООБРАЗНОЙ ТОПОЛОГИИ [star network]

СЕТЬ КОЛЬЦЕВОЙ ТОПОЛОГИИ [ring network]

СЕТЬ ПЕРЕДАЧИ ДАННЫХ [data transmission network]

СЕТЬ С КОММУТАЦИЕЙ ПАКЕТОВ, сеть с пакетной коммутацией [packet switching network]

СЕТЬ С МАРКЕРНЫМ ДОСТУПОМ, сеть с передачей маркера, маркерная сеть [token network]

СЕТЬ СТАНДАРТА Н.323, сеть Н.323 [Н.323 network, Н.323]

СЕТЬ ШИННОЙ ТОПОЛОГИИ [bus network]

СЕТЬ ЭВМ, вычислительная сеть [computer network]

СЕТЬ APPLETALK [AppleTalk]

СЕТЬ ETHERNET [Ethernet]

СЕТЬ EXTRANET [extranet]

СЕТЬ NETWARE [NetWare]

СИСТЕМНАЯ СЕТЕВАЯ АРХИТЕКТУРА, архитектура SNA, [Systems Network Architecture (SNA)]

ТОПОЛОГИЯ СЕТИ [network topology]

ТРАФИК [traffic]

УЗЕЛ СЕТИ [network node, node]

ШИННАЯ СЕТЬ С МАРКЕРНЫМ ДОСТУПОМ, шинная сеть с передачей маркера, маркерная шинная сеть [token bus network]

ШИРОКОВЕЩАТЕЛЬНАЯ КОМПЬЮТЕРНАЯ СЕТЬ, широковещательная сеть [broadcast network]

ЭКСТРАНЕТ, расширенная интрасеть, сеть extranet [extranet]

3. Программное обеспечение

3.1. Вычислительные процессы

АБСТРАКТНЫЙ ФАЙЛ [named pipe]

АВАРИЙНОЕ ЗАВЕРШЕНИЕ [abnormal end, abend]

АВАРИЙНЫЙ ОСТАНОВ, авост [abrupt end, abend]

АВАРИЯ [crash]

АКТИВАЦИЯ, фокус управления [activation, focus of control flow]

АКТИВИЗАЦИЯ [activation]

АКТИВНАЯ ПРОГРАММА [active program]

АТОМАРНЫЙ [atomic]

БЕНЧМАРКА [benchmark]

БУФЕРНЫЙ ФАЙЛ [spool file]

ВОЗВРАТ В ИСХОДНОЕ СОСТОЯНИЕ [reset]

ВОССТАНОВЛЕНИЕ, регенерация [recovery, regeneration]

ВРЕМЕННЫЙ ОБЪЕКТ [transient object]

ВРЕМЯ ВЫБОРКИ [fetch time]

ВРЕМЯ ДОСТУПА [access time]

ВРЕМЯ ОТВЕТА, время отклика, время реакции [response time]

ВРЕМЯ ПОИСКА [seek time]

ВРЕМЯ ПРОГОНА, время выполнения [running time, execute time]

ВРЕМЯ ПРОСТОЯ [downtime, idle time]

ВРЕМЯ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА, процессорное время [CPU time]

ВСТРОЕННЫЕ ПРОГРАММЫ, зашитые программы [firmware]

ВЫГРУЗКА [preemption]

ВЫПОЛНЕНИЕ ПРОГРАММЫ, исполнение программы [program run, computer run, run]

ВЫХОД ИЗ ПРИЛОЖЕНИЯ [quit]

ДВОЙНАЯ ОШИБКА [double fault]

ДИНАМИЧЕСКАЯ ЗАГРУЗКА ПРОГРАММЫ [dynamic program loading]

ДЛИНА КОНВЕЙЕРА [pipeline length]

ЗАВЕРШЕНИЕ ПРОГРАММЫ [completion, termination]

ЗАВИСАНИЕ [hang-up, starvation]

ЗАГРУЗКА [loading]

ЗАГРУЗКА ПРОГРАММЫ [program loading]

ЗАКОН АМДАЛА [Amdal's law]

ЗАПУСК ПРОГРАММЫ [program start]

ЗАХВАТ, монопольное использование [lockout]

ИМЕНОВАННЫЙ КАНАЛ, абстрактный файл [named pipe]

ИНСТАЛЛЯЦИЯ, установка [installation, setup]

ИСПОЛНЯЕМЫЙ ФАЙЛ [executable file]

КВАНТ ВРЕМЕНИ [time slice]

КВАНТОВАНИЕ ВРЕМЕНИ [time slicing]

КОМПОНОВКА, редактирование связей [linking, linkage editing]

КОНВЕЙЕР [pipeline]

КОНВЕЙЕРНАЯ ОБРАБОТКА [pipelined processing]

МАСТЕР-ПРОГРАММА [master program]

МАШИННОЕ ВРЕМЯ [computer time]

МАШИННЫЙ СБОЙ, аппаратный сбой, сбой [machine failure, failure, malfunction]

МУЛЬТИПРОГРАММИРОВАНИЕ [multiprogramming]

НЕВОССТАНОВИМАЯ ОШИБКА ПРИЛОЖЕНИЯ [Unrecoverable application error (UAE)]

НЕИСПРАВИМАЯ ОШИБКА, фатальная ошибка [unrecoverable error, fatal error, catastrophic error]

НОМЕР ПРОЦЕССА, ранг [rank]

ОБРАБАТЫВАЮЩАЯ ПРОГРАММА [processing program]

ОБРАБОТКА ПРЕРЫВАНИЙ [interrupt processing]

ОБЪЕКТ ПРОГРАММЫ, программный объект [program object]

ОЖИДАНИЕ [expectation]

ОТКАТ [backtracking, undo]

ПАКЕТ [batch, package, packet]

ПАКЕТНАЯ ОБРАБОТКА, пакетный режим [batch processing, batch mode]

ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА [concurrent processing, parallel processing, simultaneous processing]

ПАРАЛЛЕЛЬНЫЙ ВЫЧИСЛИТЕЛЬНЫЙ ПРОЦЕСС [parallel calculation process]

ПЕРЕЗАПИСЬ [rewrite]

ПЕРЕЗАПУСК, повторный пуск, рестарт [rerun, restart]

ПОТОК [flow, stream, thread]

ПОТОК ОБЪЕКТОВ [object flow]

ПОТОК УПРАВЛЕНИЯ [control flow, thread]

ПРОГОН ПРОГРАММЫ [program run, run]

ПРОЦЕСС [process]

ПУЛ [pool]

РАЗДЕЛЕНИЕ ВРЕМЕНИ [time sharing]

СВОПИНГ [swapping]

СЕГМЕНТ КОНВЕЙЕРА, ступень конвейера [pipeline stage]

СРЕДА [environment, medium]

СТРАНИЧНЫЙ ОБМЕН [page communication]

ТУПИКОВАЯ СИТУАЦИЯ, взаимная блокировка, тупик [deadlock]

ФАЙЛ ПРОТОКОЛА [log file]

ФОНОВАЯ ЗАДАЧА [background task]

ЭМУЛЯТОР [emulator]

ЭМУЛЯЦИЯ [emulation]

3.2. Архитектура программных систем

АВТОМАТИЗАЦИЯ [automation, OLE automation]

АВТОНОМНАЯ ПРОГРАММА [autonomy program]

АДРЕСНОЕ ПРОСТРАНСТВО ЗАДАЧИ, адресное пространство процесса [process address space]

АРХИТЕКТУРА [architecture]

АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ [computer system architecture]

АРХИТЕКТУРА КЛИЕНТ-СЕРВЕР [client-server architecture]

АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ [software architecture]

АРХИТЕКТУРА, УПРАВЛЯЕМАЯ МОДЕЛЬЮ [model driven architecture (MDA)]

БИБЛИОТЕКА ДИНАМИЧЕСКОЙ КОМПОНОВКИ, библиотека динамического связывания, динамически загружаемая библиотека [dynamic link library (DLL)]

БИБЛИОТЕКА ПОЛЬЗОВАТЕЛЯ [personal library, private library]

БИБЛИОТЕКА ПРОГРАММ, библиотека подпрограмм [program library, subroutine library]

БИБЛИОТЕЧНАЯ ПРОГРАММА [library program]

БИБЛИОТЕЧНЫЙ ФАЙЛ [library file]

ВЕРСИЯ [release, version]

ВИРТУАЛЬНАЯ МАШИНА [virtual machine]

ВИРТУАЛЬНАЯ ПАМЯТЬ [virtual storage, virtual memory]

ВНЕДРЕННЫЙ ОБЪЕКТ [embedded object]

ДИНАМИЧЕСКИЙ ОБМЕН ДАННЫМИ [dynamic data exchange (DDE)]

ДИНАМИЧЕСКОЕ РАЗМЕЩЕНИЕ МОДУЛЕЙ [dynamic modules allocation]

ДИНАМИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ ПАМЯТИ [dynamic memory allocation]

ЗАГРУЗОЧНЫЙ МОДУЛЬ [load module]

ЗАМÓК, ключ защиты памяти [lock]

ЗАЩИТА ПАМЯТИ [memory protection]

ИНТЕРФЕЙС ПЕРЕДАЧИ СООБЩЕНИЙ, технология MPI [message passing interface (MPI)]

КЛИЕНТ [client]

КЛЮЧ ЗАЩИТЫ [protection key]

КОМПОНЕНТ ПРОГРАММЫ, компонент [component]

КОМПОНЕНТНАЯ МОДЕЛЬ ОБЪЕКТОВ [Component Object Model (COM)]

КОНТЕЙНЕР [container]

КОРНЕВОЙ СЕГМЕНТ [root segment]

МНОГОУРОВНЕВАЯ АРХИТЕКТУРА ПРИЛОЖЕНИЯ [multi-tiered architecture]

МОДЕЛЬ COM [Component Object Model (COM)]

ОБЪЕКТНАЯ ПРОГРАММА [object program]

ОБЪЕКТНЫЙ БРОКЕР ЗАПРОСОВ [Object Request Broker (ORB)]

ОБЪЕКТНЫЙ МОДУЛЬ [object module]

ОВЕРЛЕЙНАЯ ПРОГРАММА, программа с оверлейной структурой, программа с перекрываемой структурой [overlay program]

ОВЕРЛЕЙНАЯ СТРУКТУРА, структура с перекрытием [overlay structure]

ОВЕРЛЕЙНОЕ РАЗМЕЩЕНИЕ МОДУЛЕЙ [overlay modules allocation]

ПАРАЛЛЕЛЬНАЯ ВИРТУАЛЬНАЯ МАШИНА, технология PVM [parallel virtual machine (PVM)]

ПЕРЕКРЫТИЕ, оверлей [overlay]

ПЛАТФОРМНАЯ НЕЗАВИСИМОСТЬ, межплатформность [platform independence, cross-platform]

РАСПРЕДЕЛЕННАЯ КОМПОНЕНТНАЯ МОДЕЛЬ ОБЪЕКТОВ [Distributed Component Object Model (DCOM)]

РАСПРЕДЕЛЕННОЕ ПРИЛОЖЕНИЕ [distributed application]

РЕЗИДЕНТНАЯ ПРОГРАММА [resident program, resident software]

СВЯЗАННЫЙ ОБЪЕКТ [linked object]

СВЯЗЫВАНИЕ И ВНЕДРЕНИЕ ОБЪЕКТОВ [object linking and embedding, OLE]

СЕГМЕНТ ПЕРЕКРЫТИЯ, оверлейный сегмент, оверлей [overlay segment]

СЕГМЕНТ ПРОГРАММЫ [program segment]

СЕРВЕР [server]

СОСТАВНОЙ ДОКУМЕНТ [compound document]

СТАТИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ ПАМЯТИ [static memory allocation]

СТАТИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ РЕСУРСОВ [static resource allocation]

ТЕХНОЛОГИЯ CORBA [Common Object Request Broker Architecture (CORBA)]

ТЕХНОЛОГИЯ OLE [OLE]

ЭЛЕМЕНТЫ УПРАВЛЕНИЯ ACTIVEX [ActiveX controls]

3.3. Операционные системы

АВТОРИЗАЦИЯ [authorization]

АГЕНТ [agent]

АУТЕНТИФИКАЦИЯ [authentication]

АУТЕНТИФИКАЦИЯ ПО СЕРТИФИКАТУ КЛИЕНТА [client certificate authentication]

АУТЕНТИФИКАЦИЯ С ЗАПРОСОМ И ОТВЕТОМ, аутентификация запрос-ответ [challenge/response authentication]

БАЗА ДАННЫХ КАТАЛОГА [directory database]

БАЗОВАЯ СИСТЕМА ВВОДА/ВЫВОДА [Basic Input/Output System (BIOS)]

БЛОКИРОВКА УЧЕТНОЙ ЗАПИСИ [account lockout]

БЮДЖЕТ ПОЛЬЗОВАТЕЛЯ [account, user account]

ВЕКТОР ПРЕРЫВАНИЙ [interrupt vector]

ВЕРИТЕЛЬНЫЕ ДАННЫЕ [credentials]

ВНЕШНЕЕ ПРЕРЫВАНИЕ [external interrupt]

ВНЕШНЯЯ КОМАНДА MS-DOS [MS-DOS external command]

ВНУТРЕННЕЕ ПРЕРЫВАНИЕ [internal interrupt]

ВНУТРЕННЯЯ КОМАНДА MS-DOS [MS-DOS internal command]

ВХОД В СИСТЕМУ [login]

ВЫТЕСНЯЮЩАЯ МНОГОЗАДАЧНОСТЬ [preemptive multitasking]

ВЫХОД ИЗ СИСТЕМЫ [logoff, logout]

ГЕНЕРАЦИЯ СИСТЕМЫ [system generation]

ГЕНЕРАЦИЯ ОПЕРАЦИОННОЙ СИСТЕМЫ, конфигурирование операционной системы [OS generation]

ГЛАВНАЯ ФАЙЛОВАЯ ТАБЛИЦА [master file table (MFT)]

ГЛОБАЛЬНО-УНИКАЛЬНЫЙ ИДЕНТИФИКАТОР [globally unique identifier (GUID)]

ГРУППА [group]

ДЕМОН [daemon]

ДЕМОН ЛИНЕЙНОГО ПРИНТЕРА [line printer daemon (LPD)]

ДЕСПУЛИНГ [despooling]

ДИНАМИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ РЕСУРСОВ [dynamic resource allocation]

ДИСПЕТЧЕР ВИРТУАЛЬНОЙ ПАМЯТИ [virtual memory manager (VMM)]

ДИСПЕТЧЕР РАСШИРЕННОЙ ПАМЯТИ [extended memory manager (XMM)]

ДИСПЕТЧЕР УСТРОЙСТВ [devices manager]

ДИСПЕТЧЕР УСТРОЙСТВ SCSI [SCSI manager]

ДОКУМЕНТООРИЕНТИРОВАННЫЙ ПОДХОД [data centric]

ДРАЙВЕР [driver]

ДРАЙВЕР ВИРТУАЛЬНОГО УСТРОЙСТВА [virtual anything driver (VxD)]

ДРАЙВЕР ЛОГИЧЕСКОГО УСТРОЙСТВА [type-specific driver (TSD)]

ДРАЙВЕР МИНИ-ПОРТА [miniport driver]

ДРАЙВЕР ПОРТА [port driver (PD)]

ДРАЙВЕР ПРИНТЕРА [printer driver]

ДРАЙВЕР-РУСИФИКАТОР [cyrillic driver]

ДРАЙВЕР УСТРОЙСТВА [device driver]

ДРАЙВЕР ФАЙЛОВОЙ СИСТЕМЫ [file system driver (FSD)]

ЗАГРУЖАЕМЫЙ ДРАЙВЕР [loadable driver]

ЗАГРУЗКА ОПЕРАЦИОННОЙ СИСТЕМЫ [operating sistem loading]

ЗАДАНИЕ [job]

ЗАДАЧА [task]

ИДЕНТИФИКАТОР БЕЗОПАСНОСТИ, идентификатор защиты, идентификатор доступа [security identifier, security ID (SID)]

ИМПЕРСОНАЦИЯ [impersonation]

ИМЯ ГРУППЫ [group name]

ИМЯ ПОЛЬЗОВАТЕЛЯ [user name]

ИНТЕРФЕЙС ГРАФИЧЕСКИХ УСТРОЙСТВ, интерфейс GDI [graphics device interface (GDI)]

ИНТЕРФЕЙС ПРИКЛАДНОГО ПРОГРАММИРОВАНИЯ [application programming interface (API)]

ИНТЕРФЕЙС ПРОГРАММИРОВАНИЯ ПРИЛОЖЕНИЙ ДЛЯ ПЕРЕДАЧИ СООБЩЕНИЙ, интерфейс MAPI [messaging application programming interface (MAPI)]

ИНТЕРФЕЙС ПРОГРАММИРОВАНИЯ ПРИЛОЖЕНИЙ ДЛЯ ТЕЛЕФОННОЙ СВЯЗИ, интерфейс TAPI [telephony application programming interface (TAPI)]

ИНТЕРФЕЙС FAX SERVICES API [fax services API]

ИНТЕРФЕЙС JTAPI [Java Telephony API (JTAPI)]

КЛЮЧ РЕЕСТРА [registry key]

КОД ПЕРЕКЛЮЧЕНИЯ, код трансформации [thunk]

КОМАНДА ОПЕРАЦИОННОЙ СИСТЕМЫ [executive directive]

КОМАНДНЫЙ ПРОЦЕССОР, процессор командного языка [command processor]

КОМАНДНЫЙ ФАЙЛ, пакетный файл [command file, batch file]

КОНФИГУРАЦИЯ БЕЗОПАСНОСТИ [security configuration]

КООПЕРАТИВНАЯ МНОГОЗАДАЧНОСТЬ [cooperative multitasking]

КЭШИРОВАНИЕ ДИСКА [disk caching]

КЭШИРОВАНИЕ ПАРОЛЕЙ [password caching]

ЛОГИЧЕСКОЕ УСТРОЙСТВО [logical device]

ЛОКАЛЬНАЯ ГРУППА [local group]

ЛОКАЛЬНЫЙ КОМПЬЮТЕР [local computer]

ЛОКАЛЬНЫЙ ПРОФИЛЬ ПОЛЬЗОВАТЕЛЯ [user local profile]

МАСШТАБИРУЕМОСТЬ [scalability]

МЕТАФАЙЛ [metafile]

МИНИ-ДРАЙВЕР [mini driver]

МНОГОЗАДАЧНАЯ ОПЕРАЦИОННАЯ СИСТЕМА [multitasking operating system]

МНОГОПОТОЧНОСТЬ [multithreading]

МНОГОЯЗЫКОВАЯ ПОДДЕРЖКА [national language support (NLS)]

МНОЖЕСТВЕННАЯ ЗАГРУЗКА [multiple boot configuration]

НЕАКТИВНАЯ ПРОГРАММА [inactive program]

ОБОЛОЧКА ОПЕРАЦИОННОЙ СИСТЕМЫ [shell]

ОБРАБОТЧИК ПРЕРЫВАНИЙ [interrupt handler]

ОБРАЗ ЗАДАЧИ [task image]

ОБЩИЙ РЕСУРС, разделяемый ресурс, совместно используемый ресурс [shareable resource, shared resource]

ОДНОЗАДАЧНАЯ ОПЕРАЦИОННАЯ СИСТЕМА [one task operating system]

ОКНО СИСТЕМНЫХ СООБЩЕНИЙ, системное модальное диалоговое окно, системное модальное окно [system modal message box,system modal dialog box, system modal window]

ОПЕРАЦИОННАЯ СИСТЕМА (ОС) [operating sistem (OS)]

ОПЕРАЦИОННАЯ СИСТЕМА LINUX (ОС LINUX), LINUX [operating system Linux (OS Linux), Linux]

ОПЕРАЦИОННАЯ СИСТЕМА MS-DOS (ОС MS-DOS), MS-DOS [operating sistem MS-DOS (OS MS-DOS)]

ОПЕРАЦИОННАЯ СИСТЕМА NETWARE [operating system NetWare, NetWare, Novel NetWare]

ОПЕРАЦИОННАЯ СИСТЕМА UNIX (ОС UNIX), UNIX [operating sistem (OS UNIX), UNIX]

ОПЕРАЦИОННАЯ СИСТЕМА WINDOWS (ОС WINDOWS) [operating system Windows (OS Windows)]

ОПЕРАЦИОННАЯ СРЕДА [operational environment]

ОЧЕРЕДЬ ЗАДАНИЙ НА ПЕЧАТЬ [print queue]

ОШИБОЧНОЕ ОБРАЩЕНИЕ К ОТСУТСТВУЮЩЕЙ СТРАНИЦЕ, ошибка страницы [page fault]

ПАКЕТ ЗАПРОСА ПРЕРЫВАНИЯ [interrupt request package (IRP)]

ПЕРЕХОДНИК [thunk]

ПИН-КОД, PIN-код [personal identification number]

ПЛАНИРОВАНИЕ [scheduling]

ПЛАНИРОВЩИК [scheduler]

ПОДКАЧКА [paging, swapping]

ПОДСИСТЕМА ПЕЧАТИ [printing subsystem]

ПОДСКАЗКА [help, prompt]

ПОЛИТИКА [policy]

ПОЛИТИКА БЕЗОПАСНОСТИ [security policy]

ПОЛИТИКА УЧЕТНЫХ ЗАПИСЕЙ [account policy]

ПРАВО ДОСТУПА [access right, access permission]

ПРЕРЫВАНИЕ [interrupt]

ПРЕРЫВАНИЕ ВВОДА/ВЫВОДА [input-output interrupt]

ПРЕРЫВАНИЕ ОТ СХЕМ КОНТРОЛЯ, прерывание по машинному сбою [machine-check interrupt]

ПРЕРЫВАНИЕ ПРИКЛАДНОЙ ПРОГРАММЫ [software interrupt]

ПРИВИЛЕГИРОВАННАЯ КОМАНДА [privileged instruction]

ПРИВИЛЕГИРОВАННЫЙ РЕЖИМ, режим операционной системы, режим ядра [privileged mode, kernel mode]

ПРИОРИТЕТНАЯ МНОГОЗАДАЧНОСТЬ [preemptive multitasking]

ПРОГРАММА БУФЕРИЗАЦИИ, спулер [spooler]

ПРОГРАММНАЯ ПЛАТФОРМА [program platform, platform]

ПРОГРАММНОЕ ПРЕРЫВАНИЕ [software interrupt]

ПРОГРАММНЫЙ ИНТЕРФЕЙС [program interface]

ПРОГРАММНЫЙ КАНАЛ, канал [pipe]

ПРОЦЕССОР ПЕЧАТИ [print processor]

РАБОЧИЙ НАБОР ПРОЦЕССА [working set]

РАСКЛАДКА КЛАВИАТУРЫ [keyboard layout]

РАСПРЕДЕЛЕНИЕ ПАМЯТИ [memory allocation, storage allocation, memory mapping]

РАСПРЕДЕЛЕНИЕ РЕСУРСОВ [resource allocation]

РАСТРОВЫЙ ДРАЙВЕР [raster driver]

РЕГИСТРАЦИЯ [login, logon]

РЕЕСТР, реестр Windows [Registry, Windows Registry]

РЕЖИМ ПОЛЬЗОВАТЕЛЯ, пользовательский режим [user mode]

РЕСУРС [resource]

СЕРВИС, служба [service]

"СИНИЙ ЭКРАН" [blue screen]

СИСТЕМА БЕЗОПАСНОСТИ [security system]

СИСТЕМА РАЗДЕЛЕНИЯ ВРЕМЕНИ [time-sharing operating system (TSM)]

СИСТЕМНАЯ БИБЛИОТЕКА [system library]

СИСТЕМНАЯ ОБЛАСТЬ [system area]

СИСТЕМНАЯ ПРОГРАММА [system program]

СИСТЕМНОЕ СООБЩЕНИЕ [system message]

СИСТЕМНЫЙ ФАЙЛ [system file]

СЛОВО СОСТОЯНИЯ [status word]

СЛУЖБА КАТАЛОГОВ ACTIVE DIRECTORY [Active Directory]

СПИН-БЛОКИРОВКА [spin lock]

СПУЛЕР ПЕЧАТИ, спулер [print spooler, spooler]

СПУЛИНГ [spooling]

СУПЕРВИЗОР [supervisor]

ТЕНЕВОЙ ФАЙЛ [shadow file]

ТОЧКА ПРЕРЫВАНИЯ [breakpoint]

УЛЕЙ [hive]

УНИВЕРСАЛЬНЫЙ ДРАЙВЕР ПРИНТЕРА [universal printer driver, unidriver]

УПРАВЛЕНИЕ ГОЛОСОВЫМ КАНАЛОМ [media control]

УПРАВЛЕНИЕ ДОСТУПОМ [access control]

УПРАВЛЕНИЕ ПАМЯТЬЮ [memory management]

УПРАВЛЕНИЕ СОЕДИНЕНИЯМИ [call control]

УЧЕТНАЯ ЗАПИСЬ ПОЛЬЗОВАТЕЛЯ, бюджет пользователя [account, user account]

ФАЙЛОВАЯ СИСТЕМА [file system]

ФАЙЛОВАЯ СИСТЕМА ПЕРСОНАЛЬНЫХ IBM-СОВМЕСТИМЫХ КОМПЬЮТЕРОВ [IBM-PC file system]

ФАЙЛОВАЯ СИСТЕМА FAT [file allocation table file system, FAT file system]

ФАЙЛОВАЯ СИСТЕМА HPFS [high-performance file system (HPFS)]

ФАЙЛОВАЯ СИСТЕМА WINDOWS NT, файловая система NTFS [new technology file system, Windows NT file system (NTFS)]

ФАЙЛ ПОДКАЧКИ, файл замещения страниц [swapping file, swap file, paging file]

ФАЙЛ РЕЕСТРА [REG file]

ФАЙЛ СПУЛИНГА, спул-файл [spool file]

ШИФРУЮЩАЯ ФАЙЛОВАЯ СИСТЕМА [encrypting file system (EFS)]

ЯДРО ОПЕРАЦИОННОЙ СИСТЕМЫ, ядро [kernel]

3.4. Системы управления базами данных

АДМИНИСТРАТОР БАЗЫ ДАННЫХ [database administrator]

АТРИБУТ [attribute]

БАЗА ДАННЫХ (БД) [database (DB)]

БАНК ДАННЫХ [data bank]

БЛОКИРОВКА ДАННЫХ [data interlock]

ВТОРИЧНЫЙ КЛЮЧ, внешний ключ [secondary key, foreign key]

ВЫБОРКА ДАННЫХ [data retrieval]

ДОМЕН [domain]

ЗАВЕРШЕНИЕ ТРАНЗАКЦИИ [commitment, transaction commitment]

ЗАХВАТ ЗАПИСИ, блокировка записи [record locking]

ИЕРАРХИЧЕСКАЯ БАЗА ДАННЫХ [hierarchical database]

ИМЯ ИСТОЧНИКА ДАННЫХ [data source name (DSN)]

ИСТОЧНИК ДАННЫХ ODBC [ODBC data source]

КЛЮЧ ЗАПИСИ [record key]

КОРТЕЖ [tuple]

МОДЕЛЬ ДАННЫХ [data model]

НЕОСНОВНОЙ КЛЮЧ, вторичный ключ [secondary key]

ОБРАБОТКА ТРАНЗАКЦИЙ [transaction processing]

ОБЪЕКТ ДАННЫХ [data entity]

ОСНОВНОЙ КЛЮЧ, первичный ключ [master key, primary key]

ОТКАТ ТРАНЗАКЦИИ [rollback, transaction rollback]

ОТКРЫТЫЕ БАЗЫ ДАННЫХ [open databases]

ОТКРЫТЫЙ ДОСТУП К БАЗАМ ДАННЫХ [open database connectivity (ODBC)]

ПОДТВЕРЖДЕННАЯ ТРАНЗАКЦИЯ [committed transaction]

РАСПРЕДЕЛЕННАЯ БАЗА ДАННЫХ [distributed database]

РЕЛЯЦИОННАЯ БАЗА ДАННЫХ [relational database]

РЕПЛИКА [replica]

РЕПЛИКАЦИЯ БАЗЫ ДАННЫХ [database replication, replication]

РЕПОЗИТОРИЙ [repository]

СЕТЕВАЯ БАЗА ДАННЫХ [network database]

СИНХРОНИЗАЦИЯ БАЗ ДАННЫХ [database synchronization]

СИСТЕМА УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ (СУБД) [data base management system (DBMS)]

СИСТЕМА УПРАВЛЕНИЯ ИЕРАРХИЧЕСКИМИ БАЗАМИ ДАННЫХ, иерархическая СУБД [hierarchical database management system (HDBMS)]

СИСТЕМА УПРАВЛЕНИЯ РЕЛЯЦИОННЫМИ БАЗАМИ ДАННЫХ (СУРБД), реляционная СУБД [relational data base management system (RDBMS)]

СИСТЕМА УПРАВЛЕНИЯ СЕТЕВЫМИ БАЗАМИ ДАННЫХ, сетевая СУБД [network database management system]

СОСТАВНОЙ КЛЮЧ [compound key]

СХЕМА ДАННЫХ [database scheme, data scheme]

ТРАНЗАКЦИЯ [transaction]

ЦЕЛОСТНОСТЬ ДАННЫХ [data integrity]

ЯЗЫК ЗАПРОСОВ SQL [Structured Query Language (SQL)]

ЯДРО СУБД [database engine]

3.5. Текстовые процессоры

АББРЕВИАТУРА [abbreviation]

АБЗАЦ [paragraph]

АЛЬБОМНАЯ ОРИЕНТАЦИЯ СТРАНИЦЫ, ландшафтная ориентация страницы [landscape page orientation]

БЛОК ОРФОГРАФИЧЕСКОГО КОНТРОЛЯ, корректор, проверка орфографии [spelling checker]

БУКВИЦА [dropped capital letter, drop cap]

БУЛЛИТ, маркер [bullet]

ВЕКТОРНЫЙ ШРИФТ [vector font]

ВЕРСТКА СТРАНИЦ [page makeup]

ВЕРХНИЙ КОЛОНТИТУЛ [header, running head]

ВИСЯЧАЯ СТРОКА [orphan, widow]

ВСТРОЕННОЕ ИЗОБРАЖЕНИЕ [in-line image]

ВЫРАВНИВАНИЕ, выключка [justify]

ВЫРАВНИВАНИЕ ТЕКСТА, автоматический переход на новую строку [word-wrap]

ГАРНИТУРА ШРИФТА, гарнитура [font typeface, typeface, type family]

ДИСКРЕТНЫЙ ПЕРЕНОС, мягкий перенос [discretionary hyphen]

ЗАГОЛОВОЧНЫЙ ТЕКСТ, фигурный текст [artistic text, word art]

ЗАГРУЖАЕМЫЙ ШРИФТ [downloadable font]

ЗАКЛАДКА [bookmark]

ИСПРАВЛЕНИЯ [revisions]

КАПИТЕЛЬ, малые прописные [small caps]

КЕРНИНГ [kerning]

КНИЖНАЯ ОРИЕНТАЦИЯ СТРАНИЦЫ, портретная ориентация страницы [portrait page orientation]

КОЛОНТИТУЛ [header, footer, running head, running foot]

КУРСИВ [italic]

ЛИНЕЙКА [ruler]

МАРКИРОВАННЫЙ СПИСОК [bulleted list]

МЕЖСТРОЧНЫЙ ИНТЕРВАЛ, интерлиньяж [line spacing]

НАЧЕРТАНИЕ ШРИФТА, стиль шрифта [font style, type style]

НЕПЕЧАТАЕМЫЕ СИМВОЛЫ [nonprinting characters]

НИЖНИЙ КОЛОНТИТУЛ [footer, running foot]

ОБРЕЗКА ИЗОБРАЖЕНИЯ, обрезка, кадрирование [crop, cropping]

ОТСТУП [indent]

ПЕРЕВОД СТРАНИЦЫ, подача страницы, прогон страницы [form feed (FF)]

ПЕРЕНОС СЛОВ [hyphenation]

ПОДГОТОВКА ТЕКСТОВ [word processing]

ПОДЧЕРКИВАНИЕ [underline]

ПОЛУЖИРНЫЙ ШРИФТ [bold, boldface]

РАЗБИЕНИЕ НА СТРАНИЦЫ [pagination]

РАЗМЕР ШРИФТА, кегль [font size, type size]

РАЗМЕТКА СТРАНИЦЫ [page layout]

РАЗРЯДКА [expanded font]

РЕЖИМ ВСТАВКИ [insert mode]

РЕЖИМ ЗАМЕЩЕНИЯ [overtype mode]

СИСТЕМА УПРАВЛЕНИЯ ЦВЕТОМ [color management system (CMS)]

СКРЫТЫЙ ТЕКСТ [hidden text]

СЛЭШ [slash]

СТИЛЬ [style]

ТЕЗАУРУС [thesaurus]

ТЕКСТОВЫЙ СИМВОЛ [Alphanumeric character, information character]

УПЛОТНЕНИЕ ШРИФТА [condensed font]

ФОРМАТИРОВАНИЕ ТЕКСТА [text formatting]

ШРИФТ [font]

ШРИФТ TRUETYPE [TrueType font]

3.6. Компьютерная графика

АДДИТИВНЫЙ ЦВЕТ [additive color]

АНИМАЦИЯ [animation]

АНИМАЦИОННАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ [animation path]

АППЛИКАЦИЯ, графическая заготовка, художественная заготовка, клипарт [clip art]

АЭРОГРАФ [air brush]

БИТОВАЯ ГЛУБИНА ЦВЕТА, глубина цвета, разрядность цвета [color depth, bit depth]

БИТОВЫЙ МАССИВ, битовая карта, растр, растровый образ [bitmap, pixel map, pixel image]

ВЕКТОРНАЯ ГРАФИКА [vector graphics]

ВЫДЕЛЕНИЕ ОБЛАСТИ, выбор объектов, выделение [marquee]

ВЫНОСКА [callout]

ГИСТОГРАММА [bar chart, bar graph]

ГРАДАЦИИ СЕРОГО, шкала серого [gray scale]

ГРАДИЕНТ [gradient]

ГРАФИК [chart]

ГРАФИЧЕСКАЯ ПОДСИСТЕМА [graphic subsystem]

ГРАФИЧЕСКИЙ ОБЪЕКТ [graphics object]

ГРАФИЧЕСКИЙ ПРИМИТИВ [graphics primitive]

ГРАФИЧЕСКИЙ ФАЙЛ [graphic file]

ГРАФИЧЕСКИЙ РЕДАКТОР, редактор изображений [graphics editor, bitmap editor]

ГРАФИЧЕСКИЙ РЕЖИМ [graphics mode]

ГРУППА [group]

ДИАГРАММА [chart]

ЗАКРАСКА [filling]

ЗАЛИВКА [flood filing, fill, paint bucket]

ЗЕРКАЛЬНОЕ ОТРАЖЕНИЕ ПО ВЕРТИКАЛИ, отражение сверху вниз [flip vertical]

ЗЕРКАЛЬНОЕ ОТРАЖЕНИЕ ПО ГОРИЗОНТАЛИ, отражение слева направо [flip horizontal]

ИНВЕРСИЯ ЦВЕТА [invert color]

ИНСТРУМЕНТЫ МАШИННОЙ ГРАФИКИ, графические инструменты [graphics tools]

ИНТЕНСИВНОСТЬ ЦВЕТА [chroma]

КАРАНДАШ [pencil]

КИСТЬ [paint brush]

КОМПЬЮТЕРНАЯ ГРАФИКА, машинная графика [computer graphics]

КОНТУР [path]

КОНТУРНЫЙ РЕЖИМ [wire frame]

КРИВАЯ БЕЗЬЕ [Bezier curve]

ЛАССО [lasso]

ЛАСТИК [eraser]

ЛЕГЕНДА [legend]

МОДУЛЬ ПОДБОРА ЦВЕТОВ [image color matcher (ICM)]

НАСЫЩЕННОСТЬ ЦВЕТА [color saturation]

НЕВИДИМАЯ ПОВЕРХНОСТЬ [hidden surface]

ОБВОДКА [outline]

ОПТИЧЕСКОЕ РАСПОЗНАВАНИЕ СИМВОЛОВ [optical character recognition (OCR)]

ПАЛИТРА [palette]

ПЕРСПЕКТИВА [perspective]

ПИКСЕЛ [pixel]

ПИПЕТКА [eye dropper]

ПРЕОБРАЗОВАНИЕ С ПРОМЕЖУТОЧНЫМИ ФОРМАМИ, твининг [tweening]

ПРОГРАММА РАСКРАСКИ [paint program]

ПРОГРАММА РИСОВАНИЯ [drawing program]

ПРОИЗВОЛЬНЫЙ ПОВОРОТ [free rotate]

РАСТР [raster]

РАСТРИРОВАНИЕ, растеризация [rasterization]

РАСТРОВАЯ ГРАФИКА [bit-mapped graphics, raster graphics]

РАСТРОВОЕ ИЗОБРАЖЕНИЕ, растровый образ [bitmap image]

РАСТРОВЫЙ ФАЙЛ [bitmap file]

РАСТРОВЫЙ ШРИФТ [bitmapped font, raster font]

РУКА [hand]

СГЛАЖИВАНИЕ [anti-aliasing, smooth]

СЕТКА [grid]

СПЕЦИАЛЬНЫЙ СИМВОЛ [special character]

СУБТРАКТИВНЫЙ ЦВЕТ [subtractve color]

ТОН [hue]

ТРАНСФОРМАЦИЯ [transformation]

ЦВЕТ [color]

ЦВЕТОВАЯ МОДЕЛЬ [color model]

ЦВЕТОВОЙ ОХВАТ [gamut]

ЦВЕТОВОЙ ПРОФИЛЬ [profile]

ЧЕРЕССТРОЧНАЯ ЗАГРУЗКА ИЗОБРАЖЕНИЯ [interlaced loading]

ЧЕРЕССТРОЧНЫЙ GIF [interlaced GIF]

ЯРКОСТЬ [brightness]

3.7. Программное обеспечение сетей

ИНТЕРФЕЙС CGI [Common Gateway Interface (CGI)]

МЕТОД МНОЖЕСТВЕННОГО ДОСТУПА С КОНТРОЛЕМ НЕСУЩЕЙ И РАЗРЕШЕНИЕМ КОНФЛИКТОВ [carrier sense multiple access with collision detection (CSMA/CD)]

МНОГОЦЕЛЕВЫЕ РАСШИРЕНИЯ ПОЧТЫ ИНТЕРНЕТА [multipurpose internet mail extension (MIME)]

ОБЩАЯ ОБЛАСТЬ [public area]

ПАКЕТНАЯ ПРОГРАММА ОТЛАДКИ СЕТЕЙ, утилита PING [packet internet groper (PING)]

ПОЧТОВЫЙ ПРОТОКОЛ POP [post office protocol, POP]

ПРОГРАММА ПРОСМОТРА, браузер [browser, viewer]

ПРОГРАММА CGI [CGI script]

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ WINDOWS SOCKETS, ПО Windows Socket [Windows Sockets (WinSoc)]

ПРОТОКОЛ МАРШРУТНОЙ ИНФОРМАЦИИ, протокол RIP [routing information protocol (RIP)]

ПРОТОКОЛ НАЧАЛЬНОЙ ЗАГРУЗКИ, протокол BOOTP [bootstrap protocol (BOOTP)]

ПРОТОКОЛ ПЕРЕДАЧИ ГИПЕРТЕКСТА, протокол HTTP [HyperText Transfer Protocol (HTTP)]

ПРОТОКОЛ УДАЛЕННЫХ КЛИЕНТОВ ЛИНЕЙНОГО ПРИНТЕРА, протокол LPR [line printer remote clients (LPR)]

ПРОТОКОЛЫ TCP/IP [Transmission Control Protocol/Internet Protocol (TCP/IP)]

ПРОТОКОЛ APPC [Advanced Program-to-Program Communications (APPC)]

ПРОТОКОЛ FTP [file transfer protocol (FTP)]

ПРОТОКОЛ NETBEUI [network basic input/output system extended user interface (NetBEUI)]

ПРОТОКОЛ POP, протокол почтового офиса, почтовый протокол [post office protocol (POP)]

ПРОТОКОЛ PPP [Point-To-Point Protocol (PPP)]

ПРОТОКОЛ SNA [system network architecture protocol, SNA protocol]

ПРОТОКОЛ SNMP [Simple Network Management Protocol (SNMP)]

ПРОТОКОЛ SSL [Secure Sockets Layer (SSL)]

ПРОТОКОЛ TELNET [Telnet]

СЕРВЕР ПЕЧАТИ [print server]

СЕРВЕР ПОЛНОМОЧИЙ [proxy server]

СЕРВЕР ПРОТОКОЛА ПЕРЕДАЧИ ГИПЕРТЕКСТА [HyperText Transfer Protocol Daemon (HTTPD)]

СЕРВЕР СЦЕНАРИЕВ WINDOWS [Windows scripting host (WSH)]

СЕРВЕР DNS, сервер имен доменов [domain name server, DNS name server]

СЕТЕВОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, сетевое ПО [netware]

СЕТЕВОЙ ПРОТОКОЛ [network protocol]

ТРАНСПОРТНЫЙ ПРОТОКОЛ, протокол транспортного уровня [transport protocol]

3.8. Различные приложения

АВТОМАТИЗИРОВАННАЯ ОБУЧАЮЩАЯ СИСТЕМА (АОС) [automated training system]

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ (АСУ) [automated control system]

АВТОРСКАЯ СИСТЕМА [authoring system]

АНТИВИРУСНАЯ ПРОГРАММА [antivirus program]

АППЛЕТ [applet]

АРХИВАТОР [archivator]

БАЗА ЗНАНИЙ [knowledge base]

БИБЛИОТЕКАРЬ [librarian]

БУФЕР ОБМЕНА [clipboard]

ГЕНЕРАТОР СЛУЧАЙНЫХ ЧИСЕЛ [random number generator]

ГИПЕРМЕДИА [hypermedia]

ГИПЕРТЕКСТ [hypertext]

ДЕФРАГМЕНТАТОР ДИСКА [disk defragmenter]

ДЕФРАГМЕНТАЦИЯ [defragmentation]

ДИСПЕТЧЕР [dispatcher, manager]

ДИСПЕТЧЕР ФАЙЛОВ [file manager]

ЗАГРУЗЧИК [loader]

ИГРОВАЯ ПРОГРАММА [game program]

ИНСТАЛЛЯТОР [installer]

ИНСТРУМЕНТАЛЬНЫЕ ПРОГРАММНЫЕ СРЕДСТВА [software tools, tools]

ИНТЕГРИРОВАННАЯ СИСТЕМА, интегрированный пакет [integrated system, integrated package]

ИНТЕГРИРОВАННАЯ СРЕДА [integrated environment]

ИНТЕГРИРОВАННАЯ СРЕДА РАЗРАБОТКИ [integrated development environment (IDE)]

ИНТЕГРИРОВАННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ [integrated software]

ИНТЕЛЛЕКТУАЛЬНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА [intelligent information system]

ИНТЕЛЛЕКТУАЛЬНАЯ ОБУЧАЮЩАЯ СИСТЕМА [intelligent tutorial system]

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ПРОГРАММИРОВАНИЯ [intelligent programming system]

ИНФИЦИРОВАННЫЙ ФАЙЛ [infected file]

КОМПОНОВЩИК, редактор связей [linker, linkage editor]

КОМПОНУЮЩИЙ ЗАГРУЗЧИК [linking loader]

КОМПЬЮТЕРНАЯ ИГРА [computer game]

КОМПЬЮТЕРНЫЙ ВИРУС, вирус [computer virus, virus]

КОНВЕРТОР [converter]

МАКРОВИРУС [macro virus]

МНОГОДОКУМЕНТНЫЙ ИНТЕРФЕЙС, архитектура MDI [multi-document interface, multiple document interface (MDI)]

НАДСТРОЙКА ОБОЛОЧКИ [shell extension]

НАСТОЛЬНАЯ РЕДАКЦИОННО-ИЗДАТЕЛЬСКАЯ СИСТЕМА [desktop publishing system]

НОБОТ [knowbot]

ОБОЛОЧКА [shell]

ОБОЛОЧКА NORTON COMMANDER [Norton Commander (NC)]

ОБСЛУЖИВАЮЩАЯ ПРОГРАММА, сервисная программа, утилита [service program, utility]

ОБУЧАЮЩАЯ ПРОГРАММА [learning program, on-line tutorial]

ОБУЧАЮЩЕЕ ПРИЛОЖЕНИЕ К ПРОГРАММНОМУ ПРОДУКТУ [computer based training system (CBT)]

ОФИСНОЕ ПРИЛОЖЕНИЕ [office application]

ПАКЕТ ПРИКЛАДНЫХ ПРОГРАММ (ППП) [application program package]

ПРЕДУПРЕЖДЕНИЕ [warning, alert, alert message]

ПРИКЛАДНАЯ ПРОГРАММА [application program]

ПРИЛОЖЕНИЕ [application]

ПРИЛОЖЕНИЕ, УПРАВЛЯЕМОЕ СОБЫТИЯМИ [event-driven application]

ПРОВОДНИК WINDOWS [Windows Explorer]

ПРОЕКТ GNU [GNU]

ПРОФИЛИРОВЩИК, профайлер [profiler]

РАБОЧАЯ КНИГА [workbook]

СВОДНАЯ ТАБЛИЦА [pivot table]

СИНТЕЗАТОР РЕЧИ [voice synthesizer]

СИНТЕЗ РЕЧИ [speech synthesis, speech generation]

СИСТЕМА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ (САПР) [computer-aided design (CAD)]

ТЕКСТОВЫЙ ПРОЦЕССОР [word processor]

ТЕКСТОВЫЙ РЕДАКТОР [text editor]

TEX [TeX]

ФАЙЛ УСТАНОВКИ [INF file]

ЭКСПЕРТНАЯ СИСТЕМА (ЭС) [expert system]

ЭЛЕКТРОННАЯ ТАБЛИЦА [spreadsheet]

4. Программирование

4.1. Формальные языки

АВТОКОД, язык ассемблера [autocode, assembly language]

АВТОРСКИЙ ЯЗЫК [authoring language]

АГРЕГАЦИЯ [aggregation]

АДА [Ada]

АЛГОЛ [Algol]

АЛГОРИТМИЧЕСКИЙ ЯЗЫК [algorithmic language]

АППАРАТНО-НЕЗАВИСИМЫЙ [hardware independent, device independent]

АССЕМБЛЕР [assembler]

АССОЦИАЦИЯ [association]

БАЙТ-КОД [bytecode]

БЕЙСИК [Basic, beginner's all-purpose symbolic instruction code]

ВАРИАНТ ИСПОЛЬЗОВАНИЯ [use case]

ВЗАИМОДЕЙСТВИЕ [interaction]

ВИЗУАЛЬНЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ [visual programming language]

ВИРТУАЛЬНАЯ МАШИНА JAVA [Java virtual machine, JVM]

ВНУТРЕННИЙ ЯЗЫК [internal language]

ВЫСОКОПРОИЗВОДИТЕЛЬНЫЙ ФОРТРАН, язык HPF [High Performance Fortran (HPF)]

ГРАФ ДЕЯТЕЛЬНОСТИ [activity graph]

ГРАФИЧЕСКАЯ НОТАЦИЯ [graphical notation]

ДЕЙСТВИЕ [action]

ДЕЙСТВУЮЩЕЕ ЛИЦО [actor]

ДЕЯТЕЛЬНОСТЬ [activity]

ДЖАВА [Java]

ДИАГРАММА ВЗАИМОДЕЙСТВИЯ [interaction diagram]

ДИАГРАММА ДЕЯТЕЛЬНОСТИ [activity diagram]

ДИАГРАММА ИСПОЛЬЗОВАНИЯ, диаграмма вариантов использования [use case diagram]

ДИАГРАММА КЛАССОВ [class diagram]

ДИАГРАММА КОМПОНЕНТОВ [component diagram]

ДИАГРАММА КООПЕРАЦИИ [collaboration diagram]

ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ [sequence diagram]

ДИАГРАММА РАЗМЕЩЕНИЯ, диаграмма развертывания [deployment diagram]

ДИАГРАММА СОСТОЯНИЙ [statechart diagram]

ИНСТРУКЦИЯ, команда [instruction, command]

ИНТЕРПРЕТИРУЕМЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ [interpretable programming language]

ИСХОДНЫЙ ЯЗЫК СИСТЕМЫ ПРОГРАММИРОВАНИЯ [source language]

КАНОНИЧЕСКАЯ НОТАЦИЯ ЯЗЫКА UML [canonical notation]

КЛАССИФИКАТОР [classifier]

КОБОЛ [Cobol]

КОД ОПЕРАЦИИ (КОП) [operation code]

КОМАНДА АССЕМБЛЕРА [assembly instruction]

КОМАНДА БЕЗУСЛОВНОЙ ПЕРЕДАЧИ УПРАВЛЕНИЯ, команда безусловного перехода [unconditional jump instruction, unconditional branch instruction]

КОМАНДА ВВОДА [input instruction]

КОМАНДА ВОЗВРАТА [return instruction]

КОМАНДА ВЫВОДА [output instruction]

КОМАНДА ПАУЗЫ [pause instruction]

КОМАНДА ПЕРЕДАЧИ УПРАВЛЕНИЯ, КОМАНДА ВЕТВЛЕНИЯ, команда перехода [jump instruction, branch instruction]

КОМАНДА ПЕРЕСЫЛКИ [sending instruction]

КОМАНДА ПРЕРЫВАНИЯ [trap instruction]

КОМАНДА УСЛОВНОГО ПЕРЕХОДА, команда условной передачи управления [condition jump instruction, condition branch instruction]

КОМПИЛИРУЕМЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ [compiled programming language]

КОМПОЗИЦИЯ [composition]

КООПЕРАЦИЯ [collaboration]

ЛИНИЯ ЖИЗНИ [lifeline]

ЛИСП [Lisp]

МАШИННАЯ КОМАНДА [computer instruction]

МАШИННАЯ ОПЕРАЦИЯ [computer operation]

МАШИННАЯ ПРОГРАММА [computer program, machine program]

МАШИННОЕ СЛОВО [computer word]

МАШИННО-НЕЗАВИСИМЫЙ ЯЗЫК [computer independent language]

МАШИННО-ОРИЕНТИРОВАННЫЙ ЯЗЫК, машинно-зависимый язык [computer-sensitive language, computer-oriented language]

МАШИННЫЙ ЯЗЫК [computer language, machine language]

МЕТАЯЗЫК [metalanguage]

НОТАЦИЯ [notation]

ОБОБЩЕНИЕ [generalization]

ОБЪЕКТНЫЙ ЯЗЫК ОГРАНИЧЕНИЙ, язык OCL [object constraint language (OCL)]

ОКНО ВЫПОЛНЕНИЯ [window of execution]

ПАСКАЛЬ [Pascal]

ПЛАВАТЕЛЬНАЯ ДОРОЖКА [swim lane]

ПОВЕДЕНИЕ [behavior]

ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЙ ЯЗЫК [problem-oriented language]

ПРОЛОГ [Prolog]

ПРОМЕЖУТОЧНЫЙ ЯЗЫК [intermediate language]

ПСЕВДОКОД [pseudocode]

СЕМАНТИКА ЯЗЫКА ПРОГРАММИРОВАНИЯ [programming language semantics]

СИ [C]

СИ++, Си плюс плюс [C++]

СИНТАКСИС [syntax]

СИНТАКСИС ЯЗЫКА ПРОГРАММИРОВАНИЯ [programming language syntax]

СТЕРЕОТИП [stereotype]

СТОРОЖЕВОЕ УСЛОВИЕ [guard condition]

УНИФИЦИРОВАННЫЙ ЯЗЫК МОДЕЛИРОВАНИЯ, язык UML [Unified Modeling Language (UML)]

УПАКОВАННЫЙ ФОРМАТ POSTSCRIPT-ФАЙЛОВ, формат EPS [encapsulated PostScript (EPS)]

ФОРМАЛИЗОВАННЫЙ ЯЗЫК [formalized language]

ФОРМАЛЬНЫЙ ЯЗЫК БЭКУСА—НАУРА, нормальная форма Бэкуса—Наура (БНФ) [Backus-Naur form (BNF)]

ФОРТРАН [Fortran]

ЭЛЕМЕНТ МОДЕЛИ [model element]

ЯЗЫК HPGL [Hewlett-Packard graphics language (HPGL)]

ЯЗЫК HTML [HyperText Markup Language (HTML)]

ЯЗЫК PERL [practical extraction and report language (Perl)]

ЯЗЫК POSTSCRIPT [PostScript]

ЯЗЫК SGML, стандартный обобщенный язык разметки [standard generalized markup language (SGML)]

ЯЗЫК UML [UML]

ЯЗЫК VISUAL BASIC [Visual Basic (VB)]

ЯЗЫК XML, расширяемый язык разметки [extensible markup language (XML)]

ЯЗЫК ВИЗУАЛЬНОГО МОДЕЛИРОВАНИЯ [visual modeling language]

ЯЗЫК МОДЕЛИРОВАНИЯ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ [Virtual Reality Modeling Language (VRML)]

ЯЗЫК ОПИСАНИЯ ИНТЕРФЕЙСОВ, язык IDL [interface definition language (IDL)]

ЯЗЫК ОПИСАНИЯ СТРАНИЦ [page description language (PDL)]

ЯЗЫК ПРОГРАММИРОВАНИЯ [programming language]

ЯЗЫК ПРОГРАММИРОВАНИЯ JAVA [Java programming language]

ЯЗЫК ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ [high-level programming language]

ЯЗЫК ПРОГРАММИРОВАНИЯ СЦЕНАРИЕВ, язык сценариев [scripting language]

ЯЗЫК РАЗМЕТКИ [markup language]

ЯЗЫК СПЕЦИФИКАЦИИ И ОПРЕДЕЛЕНИЯ, язык SDL [specification and definition language (SDL)]

ЯЗЫК СЦЕНАРИЕВ JAVASCRIPT, язык JavaScript [JavaScript]

ЯЗЫК СЦЕНАРИЕВ VBSCRIPT, язык VBScript [Microsoft Visual Basic scripting edition, VBScript]

ЯЗЫК УПРАВЛЕНИЯ ПЕЧАТЬЮ, язык PCL [printer control language (PCL)]

4.2. Типы и структуры данных

АРИФМЕТИЧЕСКОЕ ВЫРАЖЕНИЕ [arithmetic expression]

ВЕЩЕСТВЕННАЯ КОНСТАНТА, действительная константа [real constant]

ВЕЩЕСТВЕННАЯ ПЕРЕМЕННАЯ, действительная переменная [real variable]

ВЕЩЕСТВЕННОЕ ЧИСЛО, действительное число [real number]

ВЕЩЕСТВЕННЫЙ ТИП, действительный тип, тип Real [real type]

ВСТРОЕННЫЙ ТИП, стандартный тип [build-in type, standard type]

ДВУМЕРНЫЙ МАССИВ [two-dimensional array]

ЗАПИСЬ [record]

ИНДЕКС [index, subscript]

ЛОГИЧЕСКАЯ ПЕРЕМЕННАЯ [logical variable]

ЛОГИЧЕСКАЯ ФУНКЦИЯ, булева функция [Boolean function]

ЛОГИЧЕСКИЕ ОПЕРАЦИИ, булевы операции [logical operations, Boolean operations]

ЛОГИЧЕСКИЙ ТИП [logical type]

ЛОГИЧЕСКОЕ ВЫРАЖЕНИЕ, булево выражение [logical expression, Boolean expression]

ЛОГИЧЕСКОЕ ЗНАЧЕНИЕ, булево значение [logical value, Boolean value]

ОЧЕРЕДЬ [queue, first in first out (FIFO)]

ПЕРЕЧИСЛЯЕМЫЙ ТИП, перечислимый тип [enumerated type, ordinal type]

ПОРЯДКОВЫЙ ТИП [serial number type]

ПРИВЕДЕНИЕ ТИПОВ, преобразование типов [cast, coercion, type conversion]

ПРИМИТИВ [primitive]

ПРОИЗВОДНЫЙ ТИП ДАННЫХ [derived datatype]

РАЗМЕРНОСТЬ МАССИВА [dimension]

СЕМАФОР [semaphore]

СИМВОЛЬНАЯ КОНСТАНТА, текстовая константа [character constant]

СИМВОЛЬНАЯ ПЕРЕМЕННАЯ [character variable]

СИМВОЛЬНЫЙ ТИП, литерный тип, тип Char [character type]

СПИСОК [list]

СТЕК, магазин [stack, last in first out (LIFO)]

СТРОКА, строка символов [string, line]

СТРУКТУРА ДАННЫХ [data structure]

СТРУКТУРИРОВАННЫЙ ТИП [structured type]

ТИП ДАННЫХ [data type]

ТИП ДАННЫХ ЗАДАНИЯ НА ПЕЧАТЬ [print job data type]

ТИП "ЗАПИСЬ" [record type]

ФОРМА [form]

ЦЕЛАЯ ПЕРЕМЕННАЯ, переменная целого типа [integer variable]

ЦЕЛОЕ ЧИСЛО [integer]

ЦЕЛЫЙ ТИП, целочисленный тип [integer type]

ЧИСЛО [number]

ЧИСЛО ВЕЩЕСТВЕННОГО ТИПА [real type number]

ЧИСЛО С ПЛАВАЮЩЕЙ ТОЧКОЙ [floating-point number]

ЧИСЛО С ФИКСИРОВАННОЙ ТОЧКОЙ [fixed-point number]

ЧИСЛО ЦЕЛОГО ТИПА [integer type number]

4.3. Программные конструкции

АДРЕСНАЯ КОНСТАНТА [address constant, A-constant]

АДРЕСНАЯ ССЫЛКА [address reference]

АРГУМЕНТ [argument]

АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ [arithmetical operations]

АРИФМЕТИЧЕСКИЙ ОПЕРАТОР [arithmetic operator]

АРИФМЕТИЧЕСКИЙ УСЛОВНЫЙ ОПЕРАТОР [arithmetic IF statement]

АРИФМЕТИЧЕСКОЕ ВЫРАЖЕНИЕ [arithmetic expression]

БЕЗУСЛОВНАЯ ПЕРЕДАЧА УПРАВЛЕНИЯ, безусловный переход [unconditional branch]

БЛОК ПРОГРАММЫ, программный блок, блок [program block, block]

БЛОКИРОВАНИЕ [blocking]

БЛОКИРОВКА [locking, lockout]

ВКЛЮЧЕНИЕ ФАЙЛА [file inclusion]

ВЛОЖЕННЫЙ БЛОК, подблок [internal block]

ВЛОЖЕННЫЙ ЦИКЛ [internal cycle]

ВНЕШНЯЯ ССЫЛКА, межмодульная ссылка [external reference, intermodular reference]

ВНУТРЕННЯЯ ССЫЛКА [inline link]

ВОЗВРАТ [return]

ВСТРОЕННАЯ ПРОЦЕДУРА, стандартная процедура [nested procedure, standard procedure]

ВСТРОЕННАЯ ФУНКЦИЯ, стандартная функция [built-in function, standard function]

ВХОД В ПОДПРОГРАММУ [subroutine entry]

ВХОД В ПРОГРАММУ [program entry]

ВЫЗОВ ПОДПРОГРАММЫ, обращение к подпрограмме [call, subroutine call]

ВЫЗОВ ПРОЦЕДУРЫ, обращение к процедуре [procedure statement, procedure call]

ВЫЗОВ ФУНКЦИИ, обращение к функции [function call]

ВЫЗЫВАЕМАЯ ПОДПРОГРАММА [called subroutine]

ВЫЗЫВАЮЩАЯ ПРОГРАММА [calling program]

ВЫРАЖЕНИЕ [expression]

ВЫХОД [exit]

ВЫХОД ИЗ ЦИКЛА [loop termination]

ВЫЧИСЛЯЕМЫЙ ОПЕРАТОР ПЕРЕХОДА [evaluate go to statement]

ГЛАВНАЯ ПРОГРАММА, основная программа [main program, head program]

ГРАНИЦА ИНДЕКСА [subscript boundary]

ГРАНИЧНАЯ ПАРА [bound pair]

ДЕСТРУКТОР [destructor]

ДЖОКЕР [joker]

ЗАГОЛОВОК ПРОЦЕДУРЫ [procedure header]

ЗАГОЛОВОК ФУНКЦИИ [function header]

ЗАГОЛОВОК ЦИКЛА [loop header]

ЗАДАНИЕ НА ПЕЧАТЬ [print job]

ЗАКОММЕНТИРОВАТЬ [comment out]

ЗАКРЫТОЕ СВОЙСТВО КЛАССА, приватное свойство класса [private class property, private property]

ЗАКРЫТЫЙ МЕТОД КЛАССА, приватный метод класса [private class method, private method]

ЗАЩИЩЕННОЕ СВОЙСТВО КЛАССА [protected class property, protected property]

ЗАЩИЩЕННЫЙ МЕТОД КЛАССА [protected class method, protected method]

ЗНАК АРИФМЕТИЧЕСКОЙ ОПЕРАЦИИ [arithmetic operation character]

ЗНАК ЛОГИЧЕСКОЙ ОПЕРАЦИИ [logical operation character]

ЗНАЧЕНИЕ ТИПА ДАННЫХ [data value]

ИДЕНТИФИКАТОР [identifier]

ИМЕНОВАННОЕ ЗНАЧЕНИЕ [tagged value]

ИМЕНОВАННЫЕ АРГУМЕНТЫ [named arguments]

ИМЯ [name]

ИСКЛЮЧИТЕЛЬНАЯ СИТУАЦИЯ, особая ситуация [exception]

ИСПОЛНЯЕМЫЙ ОПЕРАТОР [imperative statement]

КЛАСС [class]

КЛЮЧЕВОЕ СЛОВО, зарезервированное слово, служебное слово [keyword, reserved word]

КОММЕНТАРИЙ [comment]

КОНСТАНТА [constant]

КОНСТРУКТОР [constructor]

ЛИТЕРАЛ [literal]

ЛОКАЛЬНЫЙ ВЫЗОВ ПРОЦЕДУРЫ [local procedure call (LPC)]

МАССИВ [array]

МЕЖМОДУЛЬНАЯ ССЫЛКА [inter modular reference]

МЕТКА [label]

МЕТОД КЛАССА [class method, method]

ОБЛАСТЬ ДЕЙСТВИЯ [owner scope, scope]

ОБЩИЙ БЛОК [common block]

ОДИНОЧКА [singleton]

ОДНОМЕРНЫЙ МАССИВ [one-dimensional array]

ОДНОМЕСТНАЯ ОПЕРАЦИЯ, унарная операция [unary operation, monadic operation]

ОПЕРАНД [operand]

ОПЕРАТОР [statement, operator]

ОПЕРАТОР БЕЗУСЛОВНОГО ПЕРЕХОДА [imperative go to statement]

ОПЕРАТОР ВВОДА [input statement]

ОПЕРАТОР ВОЗВРАТА [return statement]

ОПЕРАТОР ВЫБОРА, переключатель [case statement]

ОПЕРАТОР ВЫВОДА [output statement]

ОПЕРАТОР ВЫЗОВА ПОДПРОГРАММЫ, оператор обращения к подпрограмме [calling statement]

ОПЕРАТОР ЗАДАНИЯ ФОРМАТА, оператор формата [format statement]

ОПЕРАТОР ПРИСВАИВАНИЯ [assignment statement]

ОПЕРАТОР ПРОДОЛЖЕНИЯ [continue statement]

ОПЕРАТОР ПРОЦЕДУРЫ, оператор вызова процедуры, вызов процедуры [procedure statement]

ОПЕРАТОР УСЛОВНОГО ПЕРЕХОДА [conditional go to statement]

ОПЕРАТОР ЦИКЛА [cycle statement]

ОПЕРАТОР ЦИКЛА ЯЗЫКА ФОРТРАН [Fortran loop statement, DO loop]

ОПЕРАТОРНЫЕ СКОБКИ [statement brackets]

ОПЕРАТОР-ФУНКЦИЯ [statement function]

ОПЕРАТОРЫ ПЕРЕХОДА [GO TO statements]

ОПЕРАТОРЫ ЦИКЛА ЯЗЫКА ПАСКАЛЬ, предложения цикла языка Паскаль [Pascal loop statements, Pascal loop sentences]

ОПЕРАТОРЫ ЦИКЛА ЯЗЫКА СИ, предложения цикла языка Си [C loop statements, C loop sentences]

ОПЕРАЦИЯ [operation]

ОПЕРАЦИЯ СРАВНЕНИЯ, операция отношения [comparison operation]

ОПИСАНИЕ, объявление [declaration, specification]

ОПИСАНИЕ КЛАССА [class declaration]

ОПИСАНИЕ МАССИВА [array declaration]

ОПИСАНИЕ МЕТОК [label declaration]

ОПИСАНИЕ ПЕРЕМЕННОЙ [variable declaration]

ОПИСАНИЕ ПРОЦЕДУРЫ [procedure declaration]

ОПИСАНИЕ ТИПА, определение типа [type declaration]

ОПИСАНИЕ ФУНКЦИИ [function declaration]

ОПИСАТЕЛЬ, спецификатор [declarator, descriptor, specifier]

ОТКРЫТОЕ СВОЙСТВО КЛАССА [public class property, public property]

ОТКРЫТЫЙ МЕТОД КЛАССА [public class method, public method]

ПАРАМЕТР ЦИКЛА, переменная цикла, управляющая переменная цикла [cycle parameter, cycle index, loop variable]

ПЕРЕДАЧА УПРАВЛЕНИЯ [control transfer]

ПЕРЕКРЕСТНАЯ ССЫЛКА [cross reference]

ПЕРЕМЕННАЯ [variable]

ПЕРСИСТЕНТНЫЙ ОБЪЕКТ [persistent object]

ПОДКЛАСС [subclass]

ПОДПРОГРАММА [subroutine]

ПОЛИМОРФНАЯ ОПЕРАЦИЯ [polymorphic operation]

ПОМЕЧЕННЫЙ ОПЕРАТОР [labeled statement]

ПРЕДЛОЖЕНИЕ [sentence, statement]

ПРИСВАИВАНИЕ [assignment]

ПУСТОЙ ОПЕРАТОР [dummy statement, null statement]

РАЗВИЛКА [fork]

РАЗРУШЕННАЯ ССЫЛКА [broken link]

РЕКУРСИВНАЯ ПОДПРОГРАММА [recursive subroutine]

РЕКУРСИВНАЯ ПРОЦЕДУРА [recursive procedure]

РЕКУРСИВНАЯ ФУНКЦИЯ [recursive function]

СВОЙСТВО КЛАССА [class property, property]

СИГНАТУРА [signature]

СОЕДИНЕНИЕ [join]

СОЗДАНИЕ ЭКЗЕМПЛЯРА [instantiation]

СОСТАВНОЙ ОПЕРАТОР [compound statement]

СПИСОК ФАКТИЧЕСКИХ ПАРАМЕТРОВ, список аргументов [argument list]

СПИСОК ФОРМАЛЬНЫХ ПАРАМЕТРОВ, список параметров [parameter list]

СПИСОК ЦИКЛА [cycle list]

ССЫЛКА [reference, link]

СТАНДАРТНАЯ ПОДПРОГРАММА [standard subroutine]

СУПЕРКЛАСС [superclass]

ТЕЛО ПРОЦЕДУРЫ [procedure body]

ТЕЛО ФУНКЦИИ [function body]

ТЕЛО ЦИКЛА [loop body]

ТОЧКА ВОЗВРАТА [return point]

ТОЧКА ВХОДА [entry point]

ТОЧКА ВЫХОДА [exit point]

УКАЗАТЕЛЬ, индикатор [indicator, pointer]

УПРАВЛЯЮЩИЙ ОПЕРАТОР [control statement]

УСЛОВИЕ ВЫХОДА ИЗ ЦИКЛА, условие завершения цикла [truncation condition]

УСЛОВНЫЙ ОПЕРАТОР [conditional statement]

ФАКТИЧЕСКИЙ ПАРАМЕТР [actual parameter]

ФОРМАЛЬНЫЙ ПАРАМЕТР [formal parameter]

ФОРМАТ [format]

ФУНКЦИЯ [function]

ФУНКЦИЯ ОБРАТНОГО ВЫЗОВА [callback function]

ЦИКЛ [cycle, loop]

ЦИКЛ С ПАРАМЕТРОМ [cycle with loop variable, cycle with parameter]

ЦИКЛ С ПОСТУСЛОВИЕМ, цикл с нижним окончанием [repeat-until loop]

ЦИКЛ С ПРЕДУСЛОВИЕМ, цикл с верхним окончанием [while loop]

ЦИКЛ СО СЧЕТЧИКОМ, арифметический цикл [cycle with a counter, arithmetical cycle]

ЦИКЛИЧЕСКИЙ ПРОЦЕСС, цикл [cyclic process, cycle]

ЭКЗЕМПЛЯР КЛАССА [class instance, instance]

ЭЛЕМЕНТ МАССИВА, индексированная переменная, переменная с индексами [array element, subscripted variable]

4.4. Методология программирования

АБСТРАКТНЫЙ КЛАСС, базовый абстрактный класс [abstract class, abstract base class]

АБСТРАКТНЫЙ ТИП ДАННЫХ [abstract data type]

АВТОМАТИЧЕСКИЙ СИНТЕЗ ПРОГРАММ [automatic program synthesis]

АВТОНОМНЫЕ ИСПЫТАНИЯ ПРОГРАММЫ [stand-alone program tests]

АЛЬФА-ВЕРСИЯ [alpha release]

АЛЬФА-ТЕСТИРОВАНИЕ [alpha testing]

АРТЕФАКТ [artifact]

АСПЕКТО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ [aspect oriented programming (AOP)]

БЕСКОНЕЧНЫЙ ЦИКЛ [infinite loop]

БЕТА-ВЕРСИЯ [beta-release, beta-version]

БЕТА-ТЕСТИРОВАНИЕ [beta testing]

БЛОК-СХЕМА ПРОГРАММЫ, схема программы [program flowchart]

БЛОЧНАЯ СТРУКТУРА ПРОГРАММЫ [program block structure]

ВЕРИФИКАЦИЯ ПРОГРАММЫ [program verification]

ВЕТВЛЕНИЕ ПРОГРАММЫ [branching]

ВЕТВЬ АЛГОРИТМА [algorithm branch]

ВЕТВЬ ПРОГРАММЫ [program branch]

ВИРТУАЛЬНАЯ ФУНКЦИЯ [virtual function]

ВЛОЖЕННЫЙ АВТОМАТ [submachine]

ВНУТРЕННИЙ ПЕРЕХОД [internal transition]

ВОПЛОЩЕНИЕ [reification]

ВОСХОДЯЩЕЕ ПРОГРАММИРОВАНИЕ [bottom up programming]

ГЛОБАЛЬНАЯ ПЕРЕМЕННАЯ [global variable]

ДЕЙСТВИЕ ПРИ ВХОДЕ [entry action]

ДЕЛЕГИРОВАНИЕ [delegation]

ДЕМОНСТРАЦИОННАЯ ПРОГРАММА [demonstration program (demo program)]

ДЖАВА-СЦЕНАРИЙ, JAVA-сценарий [Java script]

ДИАГРАММА ПОТОКОВ ДАННЫХ [data flow diagram (DFD)]

ДОБАВЛЯЕМЫЙ МОДУЛЬ, добавочный модуль, надстройка [add-on module]

ДОКУМЕНТАЦИЯ ПРОГРАММНОГО ПРОДУКТА, ПРОГРАММНАЯ ДОКУМЕНТАЦИЯ [software documentation]

ЗАГЛУШКА [module stub]

ЗАКЛЮЧИТЕЛЬНОЕ СОСТОЯНИЕ [final state]

ЗАПЛАТА [patch]

ЗАЦИКЛИВАНИЕ [loophole]

ЗАЦИКЛИВАНИЕ ПРОГРАММЫ [cycling]

ИНВАРИАНТ [invariant]

ИНКАПСУЛЯЦИЯ [encapsulation]

ИСПЫТАНИЕ ПРОГРАММЫ, тестирование программы [program testing]

ИСХОДНЫЙ МОДУЛЬ [source module]

ИТЕРАЦИОННЫЙ ЦИКЛ [iterative loop]

КОМПЛЕКСНЫЕ ИСПЫТАНИЯ ПРОГРАММНОГО ПРОДУКТА [complex tests of the program product]

КОНЕЧНЫЙ АВТОМАТ [finite-state machine]

ЛИНЕЙНЫЙ АЛГОРИТМ [serial algorithm]

ЛОВУШКА [hook, trap]

ЛОГИЧЕСКАЯ ОШИБКА, ошибка алгоритма [logic error, algorithm error]

ЛОКАЛИЗАЦИЯ ОШИБКИ [error localization]

ЛОКАЛЬНАЯ ПЕРЕМЕННАЯ [local variable]

МЕТАКЛАСС [metaclass]

МЕТАМОДЕЛЬ [metamodel]

МЕТОДОЛОГИЯ ПРОГРАММИРОВАНИЯ [programming methodology]

МЕТРИКА [metric]

МНОЖЕСТВЕННОЕ НАСЛЕДОВАНИЕ [multiple inheritance]

МОДЕЛЬ [model]

МОДЕЛЬ ПЕРЕДАЧИ СООБЩЕНИЙ [massage passing model]

МОДЕЛЬ ПРОГРАММИРОВАНИЯ [programming model]

МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ [modular programming]

МОНИТОР [monitor]

НАСЛЕДОВАНИЕ [inheritance]

НАЧАЛЬНОЕ СОСТОЯНИЕ [initial state]

НЕВЫПОЛНЯЕМЫЙ ОПЕРАТОР [nonexecutable statement]

НЕЗАВИСИМОСТЬ ПРИЛОЖЕНИЙ [application independence]

НЕЯВНОЕ ОПИСАНИЕ [implicit declaration]

НИСХОДЯЩЕЕ ПРОГРАММИРОВАНИЕ [top-down programming]

ОБЛАСТЬ ВИДИМОСТИ, видимость [scope]

ОБЛАСТЬ ОПРЕДЕЛЕНИЯ, область существования [definitional domain]

ОБРАЗЕЦ ПРОЕКТИРОВАНИЯ, паттерн [design pattern]

ОБРАТНОЕ ПРОЕКТИРОВАНИЕ [reverse engineering]

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ (ООП) [object-oriented programming (OOP)]

ОГРАНИЧЕНИЕ [constraint]

ОПТИМИЗАЦИЯ [optimization]

ОТЛАДКА ПРОГРАММЫ [program debugging]

ОТЛОЖЕННОЕ СОБЫТИЕ [deferred event]

ОШИБКА В ПРОГРАММЕ, программная ошибка [program error, malfunction]

ПАРАДИГМА ПРОГРАММИРОВАНИЯ [programming paradigm]

ПЕРЕДАЧА ПАРАМЕТРОВ [parameter passing]

ПЕРЕКЛЮЧАЮЩЕЕ СОБЫТИЕ [trigger]

ПЕРЕХОД [transition]

ПЕРЕХОД ПО ЗАВЕРШЕНИИ [completion transition]

ПОСТУСЛОВИЕ [post condition]

ПРЕДУСЛОВИЕ [precondition]

ПРЕОБРАЗОВАНИЕ ОБЪЕКТОВ [object conversion]

ПРИНЦИП ПОДСТАНОВОЧНОСТИ [substitutability principle]

ПРОГРАММИРОВАНИЕ [programming]

ПРОГРАММИРОВАНИЕ, ОРИЕНТИРОВАННОЕ НА СОБЫТИЯ, событийно-управляемое программирование [event-driven programming]

ПРОГРАММНЫЙ МОДУЛЬ [module, program unit]

ПРОИЗВОДНЫЙ ЭЛЕМЕНТ [derived element]

ПРОСТРАНСТВО ИМЕН [name space]

ПРОТОТИП [prototype]

ПРОФИЛЬ ПРОГРАММЫ [program profile]

ПРОЦЕДУРА [procedure]

ПРОЦЕДУРНОЕ ПРИЛОЖЕНИЕ [procedural application]

СЕМАНТИЧЕСКАЯ ОШИБКА [semantic error]

СИНТАКСИЧЕСКАЯ КОНСТРУКЦИЯ [syntax construction]

СИНТАКСИЧЕСКАЯ ОШИБКА [syntax error]

СОБЫТИЕ [event]

СОВМЕСТИМОСТЬ [compatibility]

СОВМЕСТИМОСТЬ СВЕРХУ ВНИЗ [downward compatibility]

СОВМЕСТИМОСТЬ СНИЗУ ВВЕРХ [upward compatibility]

СОСТОЯНИЕ [state]

СТРУКТУРНОЕ ПРОГРАММИРОВАНИЕ [structured programming]

СЦЕНАРИЙ [script]

ТЕХНОЛОГИЯ ACTIVEMOVE [ActiveMovie]

ТЕХНОЛОГИЯ ACTIVEX [ActiveX]

УТОЧНЕНИЕ [refinement]

ЭЛЕМЕНТЫ БЛОК-СХЕМ [flowchart symbol]

MPI-ПРОГРАММА [MPI-program]

MPMD-МОДЕЛЬ [multiple progam multiple data (MPMD)]

PVM-ПРОГРАММА [PVM-program]

SPMD-МОДЕЛЬ [single progam multiple data (SPMD)]

4.5. Инструментальные средства программирования

АВТОМАТИЗИРОВАННАЯ РАЗРАБОТКА ПРОГРАМ, инструмент CASE [computer-aided software engineering (CASE), CASE-tool]

АНАЛИЗ ТРЕБОВАНИЙ [requirements analysis]

БЛОК УПРАВЛЕНИЯ [control block]

БЫСТРАЯ РАЗРАБОТКА ПРИЛОЖЕНИЙ [rapid application development (RAD)]

ВОДОПАДНЫЙ ПРОЦЕСС РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ [waterfall software development process]

ВРЕМЯ КОМПИЛЯЦИИ [compilation time, compiling time, compile time]

ДАМП [dump]

ДАМП ПАМЯТИ [memory dump, dump]

ДАМП ЭКРАНА [screen dump, dump]

ДЕСКРИПТОР, описатель [descriptor]

ИНКРЕМЕНТНЫЙ ПРОЦЕСС РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ [incremental software development process]

ИНТЕРПРЕТАТОР [interpreter]

ИНТЕРПРЕТАЦИЯ [interpretation]

ИСХОДНАЯ ПРОГРАММА [source program]

ИСХОДНЫЙ ТЕКСТ [incoming text, source text]

ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ, исходная программа [source program text]

КАРКАС [framework]

КОМПИЛЯТОР [compiler]

КОМПИЛЯЦИЯ [compilation]

КОНСТРУИРОВАНИЕ [construction]

КОНТРОЛЬНАЯ ТОЧКА [checkpoint]

КУЧА [heap]

ЛЕКСИЧЕСКИЙ АНАЛИЗАТОР, сканер [lexical analyzer, scanner]

ЛИСТИНГ [listing]

МАКРОАССЕМБЛЕР [macro assembler]

МАКРОБИБЛИОТЕКА [macro definition library]

МАКРОВЫЗОВ, макрокоманда [macro call, macro instruction, macrocode]

МАКРОГЕНЕРАТОР [macro generator]. То же, что макропроцессор

МАКРОГЕНЕРАЦИЯ [macro generation]

МАКРОКОМАНДА [macro, macro command, macrocode, macro instruction, macro call]

МАКРООПРЕДЕЛЕНИЕ [macro declaration, macro definition]

МАКРОПАРАМЕТР [macro parameter]

МАКРОПОДСТАНОВКА [macro generation]

МАКРОПРОГРАММА [macro program]

МАКРОПРОГРАММИРОВАНИЕ [macro programming]

МАКРОПРОЦЕССОР, макрогенератор [macro processor, macro generator]

МАКРОРАСШИРЕНИЕ [macro expansion]

МАКРОС [macros]

МАКРОСРЕДСТВА [macro facility]

МАКРОЯЗЫК [macro language]

МУСОР [garbage]

ОПТИМИЗИРУЮЩИЙ КОМПИЛЯТОР [optimizing compiler]

ОТЛАДОЧНЫЙ КОМПИЛЯТОР [checkout compiler]

ОТЛАДЧИК [debugger]

ПЕРЕМЕЩАЕМОСТЬ [relocatability]

ПЕРЕНОСИМОСТЬ ПРОГРАММЫ, портабильность программы, мобильность программы [program portability]

ПОВТОРНОЕ ИСПОЛЬЗОВАНИЕ [reuse]

ПРЕПРОЦЕССОР [preprocessor]

ПРИРАЩЕНИЕ [increment]

ПРОЕКТИРОВАНИЕ [design]

ПРОЦЕСС РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ [software development process]

РАЗВЕРТЫВАНИЕ [deployment]

РАСПЕЧАТКА [printing, listing]

РЕАЛИЗАЦИЯ [implementation]

РЕГРЕССИОННОЕ ТЕСТИРОВАНИЕ [regression testing]

РЕЕНТЕРАБЕЛЬНОСТЬ, повторная входимость [reenterability]

СИСТЕМА ПРОГРАММИРОВАНИЯ (СП) [programming system]

СИСТЕМА ПРОГРАММИРОВАНИЯ MPI [MPI programming system]

СИСТЕМА ПРОГРАММИРОВАНИЯ PVM [PVM programming system]

СИСТЕМА РАЗРАБОТКИ ПРОГРАММ [program development system]

СЛУЖБА КАТАЛОГОВ [directory services]

СРЕДА ПРОГРАММИРОВАНИЯ [programming environment]

ТРАНСЛЯТОР [translator]

ТРАНСЛЯЦИЯ ПРОГРАММЫ [program translation]

ТРАССИРОВКА ПРОГРАММЫ [program trace]

ТРЕБОВАНИЕ [requirement]

УПРАВЛЕНИЕ КОНФИГУРАЦИЕЙ [configuration management]

ЧИСТКА ПАМЯТИ, сборка мусора, [memory cleaning, garbage collection]

5. Работа с вычислительной системой

5.1. Работа с персональным компьютером

АВАРИЙНЫЙ СИГНАЛ [alarm]

АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ МЕСТО (АРМ) [automated working place, workstation]

АВТОМАТИЧЕСКИЙ ПОВТОР НАЖАТИЯ КЛАВИШИ [typematic]

АВТОМАТИЧЕСКОЕ СОХРАНЕНИЕ, автосохранение [autosave]

АКТИВНАЯ ЯЧЕЙКА [active cell]

АКТИВНОЕ ОКНО [active window]

АЛТ-ВВОД СИМВОЛОВ [character alt-input]

АППАРАТНЫЙ СБОЙ [failure]

АРХИВИРОВАНИЕ, архивация [archivation]

ВЕРХНИЙ РЕГИСТР [uppercase]

ВКЛЮЧИТЬ [switch on]

ВСТАВКА [paste]

ВТОРИЧНОЕ ОКНО [secondary window]

ВЫКЛЮЧИТЬ [switch off]

ВЫРЕЗАНИЕ [cut]

ГОРЯЧАЯ ОБЛАСТЬ, горячая точка [hot spot]

ГРАФИЧЕСКИЙ СИМВОЛ [graphic character]

ДВОЙНОЙ ЩЕЛЧОК [double click]

ДИАЛОГОВЫЙ ПРОЦЕССОР [conversational processor]

ДИАЛОГОВЫЙ РЕЖИМ, интерактивный режим [dialog mode, interactive mode]

ДИРЕКТИВА [directive]

ДОЧЕРНЕЕ ОКНО [child window]

КАРЕ [caret]

КЛАВИША, КЛЮЧ [key]

КЛАВИШИ ПИШУШЕЙ МАШИНКИ [type writer keys]

КЛАВИШИ РЕДАКТИРОВАНИЯ [edit keys]

КЛАВИШИ УПРАВЛЕНИЯ КУРСОРОМ [cursor control keys]

КОММЕРЧЕСКИЙ ПРОГРАММНЫЙ ПРОДУКТ [program product, software]

КОНФЛИКТНАЯ СИТУАЦИЯ, конфликт [conflict situation, conflict]

КОНФЛИКТ УСТРОЙСТВ [device conflict]

КОПИРОВАНИЕ [copy]

КУРСОР [cursor]

МАЛАЯ ЦИФРОВАЯ КЛАВИАТУРА [numeric keypad]

МОДАЛЬНОЕ ДИАЛОГОВОЕ ОКНО [modal dialog, modal window]

МОДИФИКАЦИЯ, обновление [update, updating]

МОИ ДОКУМЕНТЫ [my documents]

МОЙ КОМПЬЮТЕР [my computer]

НЕАКТИВНОЕ ОКНО [inactive window]

НИЖНИЙ РЕГИСТР [lowercase]

ОКНО [window]

ОКНО ДОКУМЕНТА [document window]

ОКНО ПРИЛОЖЕНИЯ [application window]

ОНЛАЙНОВЫЙ РЕЖИМ [online mode, on-line mode]

ОПЕРАТИВНАЯ ПОМОЩЬ [on-line help, help]

ОПЦИЯ [option]

ОШИБКА ДЕЛЕНИЯ НА НУЛЬ, деление на нуль [divide by zero error]

ОШИБКА КЭША, промах кэша [cache miss]

ОШИБКА НЕДОПУСТИМОГО КОДА ОПЕРАЦИИ [invalid opcode error]

ОШИБКА ПЕРЕПОЛНЕНИЯ [overflow error, overrun error]

ПАНЕЛЬ ЗАДАЧ [task bar]

ПАРОЛЬ [password]

ПЕРЕЗАГРУЗКА, повторная начальная загрузка [reboot]

ПОИСК ДОКУМЕНТА [document retrieval]

ПОЛНОМОЧИЕ [authorization, right]

ПРИГЛАШЕНИЕ [prompt]

ПРОФИЛЬ ПОЛЬЗОВАТЕЛЯ [user profile]

ПСЕВДОГРАФИЧЕСКИЙ СИМВОЛ [graphic character]

РАБОЧИЙ СТОЛ [desktop]

РОДИТЕЛЬСКОЕ ОКНО [parent window]

СЕАНС [session]

СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ [system software]

СИСТЕМНЫЙ ДИСК, системная дискета [system disk]

СЛИЯНИЕ [mail merge]

СПЕЦИАЛЬНЫЕ КЛАВИШИ [special keys]

СЦЕНАРИЙ РЕГИСТРАЦИИ [login script, logon script]

ТЕСТ ПРИ ВКЛЮЧЕНИИ ПИТАНИЯ, самодиагностика при включении питания, самотестирование при включении [power-on self test (POST)]

УКАЗАТЕЛЬ МЫШИ [mouse cursor]

УПРАВЛЯЮЩАЯ КЛАВИША [control key]

УПРАВЛЯЮЩИЙ СИМВОЛ [control character]

УСОВЕРШЕНСТВОВАНИЕ, обновление [upgrade, upgrading]

ФАЙЛ АВТОЗАПУСКА [autoexec file]

ФАЙЛ КОНФИГУРАЦИИ [configuration file]

ФОРМАТИРОВАНИЕ ДИСКА, форматирование [disk initialization, formatting]

ФУНКЦИОНАЛЬНЫЕ КЛАВИШИ [function keys]

ШАБЛОН [template]

ЭКРАННАЯ ЗАСТАВКА [screen saver]

5.2. Работа с компьютерной сетью

АБОНЕНТ СЕТИ [network abonent, user node]

АДРЕС IP [IP-address]

АСИНХРОННАЯ СВЯЗЬ, асинхронная передача [asynchronous connection, asynchronous transmission]

БАННЕР, банер [banner]

БЛОКИРОВКА БЮДЖЕТА [account lockout]

БРАУЗЕР, броузер [browser]

БРАУЗЕР WINDOWS NT [browser]

БЮДЖЕТ КОМПЬЮТЕРА, учетная запись компьютера [computer account]

ВЕБ-ДОКУМЕНТ, документ WEB, документ WWW [Web document, WWW document]

ВЕБ-ПАПКА, WEB-папка [Web folder]

ВЕБ-СЕРВЕР, WEB-сервер, интернет-сервер [Web server]

ВЕБ-СТРАНИЦА, страница WEB [Web page]

ВИРТУАЛЬНОЕ СОЕДИНЕНИЕ, логическое соединение [virtual connection]

ВЛОЖЕННЫЙ ФАЙЛ, прикрепленный файл, вложение [attached file, attachment]

ВОЗВРАТНЫЙ ВЫЗОВ, обратный вызов [callback]

ВЫГРУЗКА ПО ЛИНИИ СВЯЗИ [uploading]

ВЫДЕЛЕННАЯ ЛИНИЯ СВЯЗИ [dedicated line]

ГИПЕРССЫЛКА [hyperlink]

ГОЛОСОВАЯ ПОЧТА [voice mail]

ГРУППА [group]

ДВУХТОЧЕЧНЫЙ ОБМЕН [point-to-point exchange, point-to-point passing]

ДЕЙТАГРАММА [datagram]

ДЕРЕВО ДОМЕНОВ [domain tree]

ДИНАМИЧЕСКАЯ МАРШРУТИЗАЦИЯ [dynamic routing]

ДИНАМИЧЕСКАЯ ВЕБ-СТРАНИЦА [dynamic Web page]

ДОВЕРИТЕЛЬНОЕ ОТНОШЕНИЕ [trust]

ДОКУМЕНТ WEB, документ WWW [Web document, WWW document]

ДОМАШНЯЯ СТРАНИЦА, заглавная страница, начальная страница [home page]

ДОМЕН [domain]

ДОМЕННАЯ СИСТЕМА ИМЕН [domain name system (DNS)]

ЗАГРУЗКА ПО ЛИНИИ СВЯЗИ, скачивание [downloading]

ЗАЩИЩЕННЫЕ ЭЛЕКТРОННЫЕ ТРАНЗАКЦИИ, протокол SET [secure electronic transaction (SET)]

ЗВОНОК [call]

ИМЯ ДОМЕНА, доменное имя, имя DNS [domain name]

ИМЯ ПУНКТА ОБРАБОТКИ ПОЧТОВЫХ СООБЩЕНИЙ, MX-запись [mail exchanger record, MX record]

ИМЯ УЗЛА [node name]

ИНДЕКС ПОИСКОВОЙ СИСТЕМЫ, индекс [index]

ИНТЕРНЕТ-АДРЕС, адрес IP [IP address]

ИНТЕРНЕТ-ПРОВАЙДЕР [Internet provider]

ИНТЕРНЕТ-ТЕЛЕФОНИЯ, IP-телефония [IP-telephony]

ИНФОРМАЦИОННЫЙ СЕРВЕР ИНТЕРНЕТА, сервер IIS [Internet information server (IIS)]

КАДР [frame]

КАТЕГОРИИ [categories]

КЛАСС АДРЕСОВ [address class]

КЛИЕНТ ПЕЧАТИ [print client]

КЛИЕНТ-СЕРВЕРНАЯ ТЕЛЕФОНИЯ [client-server thelephony]

КОЛЛЕКТИВНЫЙ ОБМЕН [shared exchange]

КОММУНИКАТОР [communicator]

КОММУТАЦИЯ [switching]

КОММУТАЦИЯ ПАКЕТОВ [packet switching]

КОММУТАЦИЯ СОЕДИНЕНИЙ [connection switching]

КОНЕЦ ПЕРЕДАЧИ [end of transmission (EOT)]

МАРШРУТ [route]

МЕТАКОМПЬЮТИНГ [metacomputing]

МНОГОАДРЕСНАЯ РАССЫЛКА ПАКЕТОВ [multicasting]

МОДЕРАТОР [moderator]

МОСТ-МАРШРУТИЗАТОР, брутер [bridge-router, b-router, brouter]

ПАКЕТ [batch, package, packet]

ПЕРЕДАЧА ДАННЫХ [data communication, data transfer]

ПОДТВЕРЖДЕНИЕ ПРИЕМА, сигнал подтверждения приема, сигнал ACK [acknowledgement (ACK)]

ПОРТАЛ, веб-портал [portal, Web portal]

ПОСЕЩАЕМОСТЬ [traffic]

ПОСЕЩЕНИЕ [hit]

ПОСТАВЩИК СЕТЕВЫХ УСЛУГ, провайдер [service provider, provider]

ПОЧТОВЫЙ ЯЩИК [mailbox]

РАЗГОВОР В СЕТИ ИНТЕРНЕТ, чат [chat]

САЙТ [site]

САЙТ WEB [Web site]

СВЯЗЫВАНИЕ, привязка протокола [binding]

СЕАНС СВЯЗИ [session]

СИНХРОННАЯ СВЯЗЬ, синхронная передача [synchronous connection, synchronous transmission]

СКОРОСТЬ ПЕРЕДАЧИ ДАННЫХ [data transfer rate, bit rate]

СОЕДИНЕНИЕ [connection]

СПАМ [spam]

ТЕЛЕКОНФЕРЕНЦИЯ [computer conference, teleconference]

ТЕХНОЛОГИЯ DYNAMIC HTML [Dynamic HTML technology]

ТЕГ [tag]

УДАЛЕННЫЙ ВЫЗОВ ПРОЦЕДУР [remote procedure call (RPC)]

УДАЛЕННЫЙ ДОСТУП [remote access]

УНИВЕРСАЛЬНЫЙ УКАЗАТЕЛЬ РЕСУРСА, универсальный локатор ресурса [Uniform Resource Locator (URL)]

УЧЕТНАЯ ЗАПИСЬ КОМПЬЮТЕРА, бюджет компьютера, учетная запись, бюджет [computer account]

ХИТ [hit]

ЦИТИРОВАНИЕ [quoting]

ШИРОКОВЕЩАНИЕ [broadcasting]

ШИРОКОВЕЩАТЕЛЬНОЕ СООБЩЕНИЕ [broadcast message]

ШЛЮЗ [gateway]

ЭЛЕКТРОННАЯ ПОЧТА [electronic mail, E-mail]

ЯКОРЬ [anchor]

COOKIE-ИДЕНТИФИКАЦИЯ [cookie identification]

COOKIE-КОД [cookie]

COOKIE-ФАЙЛ [cookie]

5.3. Управление файлами

АРХИВНЫЙ ФАЙЛ [archived file]

БУКВА ДИСКА [disk letter]

ВОССТАНОВЛЕНИЕ ФАЙЛА [file recovery]

ВРЕМЕННЫЙ ФАЙЛ, рабочий файл [temporary file (temp file)]

ДЕРЕВО КАТАЛОГОВ [directory tree, tree]

ДЛИННОЕ ИМЯ [long name]

ЗАКРЫТИЕ ФАЙЛА [closing of a file]

ЗАКРЫТЫЙ ФАЙЛ [closed file]

ЗАЩИТА ФАЙЛА [file protection, file security]

ЗЕРКАЛЬНОЕ КОПИРОВАНИЕ, зеркалирование [mirroring]

ЗЕРКАЛЬНЫЙ НАБОР [mirror set]

КАТАЛОГ, директорий, папка [directory, folder]

КАТАЛОГИЗАЦИЯ [catalogization]

КЛАСТЕР [cluster]

КОНЕЦ ФАЙЛА [end of file (EOF)]

КОПИРОВАНИЕ КАТАЛОГА, копирование директории [directory copying]

КОПИРОВАНИЕ ФАЙЛА [file copying]

КОРЗИНА [basket, recycling bin]

КОРНЕВОЙ КАТАЛОГ, главный каталог [root directory, master directory]

ЛОГИЧЕСКИЙ ДИСК [logical disk]

ОБЩАЯ ПАПКА [shared folder]

ОТКРЫТИЕ ФАЙЛА, активизация файла [file activation]

ОТКРЫТЫЙ ФАЙЛ, активный файл [active file]

ОТМЕТКА ФАЙЛОВ И КАТАЛОГОВ [files and directories marking]

ПЕРЕИМЕНОВАНИЕ КАТАЛОГА [directory rename]

ПЕРЕИМЕНОВАНИЕ ФАЙЛА [file rename]

ПЕРЕНОС КАТАЛОГА, перемещение каталога [directory moving]

ПЕРЕНОС ФАЙЛА, перемещение файла [file moving]

ПОДКАТАЛОГ [subdirectory]

ПОИСК ФАЙЛА [file search]

ПОЛНОЕ ИМЯ ФАЙЛА [pathname]

ПРЕФИКС [prefix]

ПУТЬ [path]

ПУТЬ ДОСТУПА [pathname]

РАЗАРХИВИРОВАНИЕ [unpack]

РАСШИРЕНИЕ ИМЕНИ ФАЙЛА, расширение [file name extension, extension]

РЕЗЕРВНАЯ КОПИЯ [backup]

СЖАТИЕ ФАЙЛА [file compression, file compaction]

СКРЫТЫЙ ФАЙЛ [hidden file]

СОЗДАНИЕ КАТАЛОГА [directory creation]

СОРТИРОВКА ФАЙЛОВ [files sorting]

СОХРАНЕНИЕ [save]

ТАБЛИЦА РАЗМЕЩЕНИЯ ФАЙЛОВ, таблица FAT [file allocation table (FAT)]

ТЕКСТОВЫЙ ФАЙЛ [text file]

ТЕКУЩИЙ ДИСК, активный диск [current disk, active disc]

ТЕКУЩИЙ ДИСКОВОД [current disk drive]

ТЕКУЩИЙ КАТАЛОГ, рабочий каталог [current directory, working directory]

ТИП ФАЙЛА [file type]

УДАЛЕНИЕ КАТАЛОГА [directory delete]

УДАЛЕНИЕ ФАЙЛА [file delete]

ФАЙЛ [file]

ФАЙЛ ПОСЛЕДОВАТЕЛЬНОГО ДОСТУПА, последовательный файл [sequential file]

ФАЙЛ ПРЯМОГО ДОСТУПА, прямой файл [direct-access file]

ФАЙЛ AVI [AVI-file]

ФАЙЛ BMP [BMP file]

ФАЙЛ PCX [PCX-file]

ФАЙЛ RTF [RTF file]

ФАЙЛ TIFF [TIFF file]

ШАБЛОН ИМЕНИ ФАЙЛА [file name picture]

5.4. Интерфейс пользователя

АКТИВНЫЙ РАБОЧИЙ СТОЛ [active desktop]

АРХИТЕКТУРА MDI [multi-document interface, multiple document interface (MDI)]

БЕГУНОК, движок, ползунок [scroll box]

ВЕРТИКАЛЬНОЕ МЕНЮ [vertical menu]

ВСПЛЫВАЮЩЕЕ МЕНЮ [pop-up menu]

ВЫБОР ПО УМОЛЧАНИЮ [default option]

ВЫДЕЛЕНИЕ [select]

ГОРИЗОНТАЛЬНОЕ МЕНЮ [horizontal menu]

ГРАНИЦА ОКНА, ОБРАМЛЕНИЕ, ОКАНТОВКА [border]

ГРАФИЧЕСКИЙ ДИАЛОГ [graphic dialog]

ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ [graphic user interface (GUI)]

ДИАЛОГОВОЕ ОКНО С ВКЛАДКАМИ [tabbed dialog box]

ДРУЖЕСТВЕННЫЙ ИНТЕРФЕЙС [friendly interface]

ЗНАЧОК, иконка, пиктограмма [icon]

ИЗМЕНЕНИЕ РАЗМЕРА ОКНА [zooming]

ИНТЕЛЛЕКТУАЛЬНЫЙ ИНТЕРФЕЙС [intelligent interface]

ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ, пользовательский интерфейс [user interface]

ИНТЕРФЕЙС ADSI [active directory service interface (ADSI)]

КЛАВИШИ БЫСТРОГО ВЫЗОВА, быстрые клавиши, клавиши быстрого доступа, горячие клавиши [quick keys, access keys, hot keys]

КНОПКА [button, light button]

КОМАНДНАЯ СТРОКА [command line]

КОМАНДНАЯ КЛАВИША [accelerator key, keyboard accelerator, keyboard shortcut, shortcut key]

КОНТЕКСТНОЕ МЕНЮ [context menu, shortcut menu]

КОНТЕКСТНО-ЗАВИСИМАЯ СПРАВКА [context-sensitive help]

ЛИНЕЙКА ПРОКРУТКИ [scroll bar]

МАСТЕР [wizard]

МЕНЮ [menu]

НИСХОДЯЩЕЕ МЕНЮ, раскрывающееся меню, спускающееся меню [pull-down menu]

ПАНЕЛЬ ИНСТРУМЕНТОВ, инструментальная панель [toolbar]

ПАНЕЛЬ УПРАВЛЕНИЯ [control panel]

ПАНЕЛЬ ЭКРАНА, экранная панель, панель [screen panel, panel]

ПЕРЕКЛЮЧАТЕЛЬ, радиокнопка [option button]

ПЕРЕТАСКИВАНИЕ МЫШЬЮ, буксировка мышью [drag and drop]

ПИКТОГРАФИЧЕСКОЕ МЕНЮ [icon menu, key menu]

ПОДМЕНЮ, вложенное меню [submenu]

ПОЛЕ ВВОДА [input field]

ПОЛЕ СО СПИСКОМ [combo box]

ПОЛНЫЙ ЭКРАН [full screen view]

ПОЛОСА ПРОКРУТКИ [scroll bar]

ПРОГРЕСС-ИНДИКАТОР [progress indicator]

ПУНКТ МЕНЮ [choice]

РАБОЧАЯ СРЕДА ПОЛЬЗОВАТЕЛЯ, среда [working environment, user environment]

РАСКРЫВАЮЩИЙСЯ СПИСОК [drop-down list]

СКРОЛЛИНГ, прокрутка изображения, прокрутка [scrolling]

СПИСОК [list]

СПИСОК ВЫБОРА [pick list]

СПРАВКА, справочная система [help]

СТРОКА МЕНЮ [menu bar]

СТРОКА ПОДСКАЗКИ [help line]

СТРОКА СОСТОЯНИЯ [status bar]

ТЕКСТОВОЕ ПОЛЕ, поле [box, text box]

ФЛАЖОК [checkbox]

ФОКУС [focus]

ФОРМА, формуляр [form]

ФРЕЙМ, рамка [frame]

ЦВЕТОВОЙ МАРКЕР [color marker]

ЩЕЛЧОК [click]

ЭЛЕМЕНТ УПРАВЛЕНИЯ [control]

6. Компьютерные технологии

6.1. Компьютерная телефония

КОМПЬЮТЕРНО-ТЕЛЕФОННАЯ ИНТЕГРАЦИЯ [computer-telephony integration, (CTI)]

КОНФЕРЕНЦИЯ [conference]

ПРИВРАТНИК [gatekeeper]

ТЕЛЕКОММУНИКАЦИЯ [telecommunication]

ТЕЛЕФОННАЯ СЕТЬ [telephony network]

ТЕЛЕФОННОЕ ПРИЛОЖЕНИЕ [telephony application]

ТЕЛЕФОННЫЙ ИНТЕРФЕЙС ПРИКЛАДНОГО ПРОГРАММИРОВАНИЯ ДЛЯ ЯЗЫКА JAVA, технология JTAPI [Java Telephony API (JTAPI)]

ТЕЛЕФОННЫЙ СЕРВЕР [telephony server]

УКАЗАТЕЛЬ ВЫЗЫВАЮЩЕГО НОМЕРА [caller number delivery(CND)]

УСТРОЙСТВО УПРАВЛЕНИЯ КОНФЕРЕНЦИЯМИ [multipoint control unit (MCU)]

ФАКС [fax]

CTI-ПЛАТА [CTI board]

IP-ТЕЛЕФОН [IP-phone, IP-telephone]

6.2. Компьютерная коммерция

БАННЕРНАЯ РЕКЛАМА [banner advertising]

ТАРИФИКАЦИЯ, биллинг [billing]

ЦИФРОВОЙ СЕРТИФИКАТ [digital certificate]

ЭЛЕКТРОННАЯ КОММЕРЦИЯ [electronic commerce, e-commerce]

ЭЛЕКТРОННАЯ КОММЕРЦИЯ КЛАССА BUSINESS-TO-BUSINESS, электронная коммерция класса B2B [business-to-business e-commerce]

ЭЛЕКТРОННАЯ КОММЕРЦИЯ КЛАССА BUSINESS-TO-CONSUMER [business-to-consumer e-commerce]

ЭЛЕКТРОННЫЕ ДЕНЬГИ, цифровые деньги [digital money]

ЭЛЕКТРОННЫЙ ПАРТНЕР [electronic partner, e-partner]

ЭЛЕМЕНТ УПРАВЛЕНИЯ BUY NOW [buy now control]

А

АББРЕВИАТУРА [abbreviation]. Слово, составленное из первых букв или сокращенных частей слов, входящих в состав названий или словосочетаний, заменяющее эти названия или словосочетания. Например, ЭВМ — электронная вычислительная машина

АБЗАЦ [paragraph]. 1. Часть текста, ограниченная двумя отступами. 2. В *текстовых редакторах* — часть документа между двумя соседними маркерами конца А. В текстовом редакторе MS Word эти маркеры автоматически вносятся в текст при нажатии клавиши <Enter> (перевод строки) и видимы при нажатой кнопке ¶ (Непечатаемые знаки). Каждому А. можно придать свой формат, отличный от формата соседних абзацев

АБОНЕНТ СЕТИ [network abonent, user node]. 1. Пользователь, имеющий доступ к ресурсам *компьютерной сети* или к *сети компьютерной связи*. 2. Терминал, компьютер или *рабочая станция*, подключенные к *вычислительной сети* или сети компьютерной связи

АБСОЛЮТНЫЙ АДРЕС, физический адрес [absolute address, physical address]. 1. *Адрес* в виде *цифрового кода*, однозначно идентифицирующий реально существующие в *вычислительной системе* байт, *ячейку памяти* или устройство. А. а. непосредственно интерпретируется *аппаратными средствами* или микропрограммами *центрального процессора* вычислительной системы. А. а. либо указывается как неизменяемый адрес в *машинной команде*, либо вычисляется, исходя из *логического адреса*, и не может быть далее уточнен *программными средствами*. Ср. *логический адрес*. 2. Цифровой код, однозначно определяющий положение данных на *носителе данных*. Например, для *магнитного диска* А. а. представляет код, состоящий из номера поверхности, номера дорожки и номера сектора. См. *адресация*

АБСТРАКТНЫЙ КЛАСС, базовый абстрактный класс [abstract class, abstract base class]. В *языке программирования* C++ — *класс*, который не может быть использован для создания *объектов*, но выступает в качестве базы, на основе которой создаются другие классы. А. к. содержит хотя бы одну чисто *виртуальную функцию*. Ср. *абстрактный тип данных*

АБСТРАКТНЫЙ ТИП ДАННЫХ [abstract data type]. Понятие, введенное в конце 1970-х годов в теоретических исследованиях по *языкам программирования*. А. т. д. рассматривается как множество значений и множество операций над этими значениями, причем для описания семантики операций используются формальные математические, в частности, алгебраические, методы. Например, для А. т. д. "стек значений типа T" с операциями `push` (протолкнуть в стек) и `pop` (вытолкнуть из стека) выполняется аксиома $\forall x \in$

`T pop(push(x)) = x`. Результаты исследований по А. т. д. имеют большое значение для *объектно-ориентированного программирования*, *автоматического синтеза программ*, *искусственного интеллекта* и других разделов информатики

АБСТРАКТНЫЙ ФАЙЛ [named pipe]. То же, что *именованный канал*

АБСТРАКЦИЯ [abstraction]. Процесс выявления основных характеристик какой-либо сущности с целью обособления ее от других видов сущностей. При проведении А. рассматривается множество различных сущностей в поисках общих основных характеристик. А. подразумевает, что деление сущностей на некоторые виды происходит с некоторой точки зрения. Одна и та же сущность с разных точек зрения может принадлежать к разным абстрактным типам. А. присутствует в любом моделировании, часто на многих уровнях и для различных целей. А. является одним из основных понятий *унифицированного языка моделирования* UML

АВАРИЙНОЕ ЗАВЕРШЕНИЕ [abnormal end, abend]. Непредусмотренное прекращение *исполнения программы* из-за возникновения условий, делающих невозможным ее дальнейшее выполнение. Такими условиями могут быть *машинный сбой*, ошибка, обнаруженная *операционной системой*, переполнение, деление на ноль или извлечение квадратного корня из отрицательного числа и т. п. При А. з. операционная система предпринимает меры по сохранению работоспособности *вычислительной системы*, выдает информацию об аварийной ситуации, освобождает ресурсы, занятые задачей, и при возможности продолжает решение других задач. Ср. *аварийный останов*

АВАРИЙНЫЙ ОСТАНОВ, авост [abrupt end, abend]. Автоматическое прекращение работы компьютера из-за отказа *аппаратных средств* или *программного обеспечения*. Ср. *аварийное завершение*

АВАРИЙНЫЙ СИГНАЛ [alarm]. *Сигнал*, предупреждающий пользователя о возникновении чрезвычайной ситуации в работе *вычислительной системы*. Обычно это сообщение об ошибке или сбое в работе *программных* или *аппаратных средств*, сопровождаемое звуком. А. с. предваряет *аварийное завершение* или сопровождает *аварийный останов*

АВАРИЯ [crash]. Неожиданный выход из строя *программных* или *аппаратных средств*. А. программы приводит к потере всех несохраненных данных и может потребовать перезагрузки *операционной системы*. А. аппаратных средств может привести к потере работоспособности всей *вычислительной системы*. Последствия такой А. устраняются ремонтом или заменой неисправных блоков. При этом возможны информационные потери. Например, при А. диска могут разрушиться хранящиеся на нем данные

АВМ [analog computer]. То же, что *аналоговая вычислительная машина*

АВОСТ [abend, abrupt end]. То же, что *аварийный останов*

АВТОКОД, язык ассемблера [autocode, assembly language]. Простейший *язык программирования*, ориентированный на конкретный компьютер. По форме и содержанию А. наиболее близок к *машинному языку*. Структура команд А. определяется структурой команд и данными машинного языка, но, в отличие от последнего, А. допускает применение буквенных обозначений для *кодов операций* и адресов. Например, на А. формула $X = Y + Z$ программируется следующим образом:

```
MOV    AX, Y;        Загрузить Y в регистр
MOV    BX, Z;        Загрузить Z в регистр
ADD    AX, BX;       Добавить Z к Y
MOV    X, AX;        Сохранить результат в X
```

А. может иметь макросредства и средства управления трансляцией. Трансляция программы с А. осуществляется программой-ассемблером

АВТОМАТ [automation, automata]. 1. То же, что *автоматическое устройство*. 2. То же, что *конечный автомат*

АВТОМАТИЗАЦИЯ [automation, OLE automation]. 1. Система мероприятий по замене труда человека работой машин. 2. Способ управления программными объектами, используемый при *связывании и внедрении объектов*. А. позволяет одному приложению, которое называется *клиентом* А. или *контроллером* А., использовать объекты и средства другого приложения, которое называется *сервером* А. Например, все приложения пакета программ Microsoft Office являются как клиентами, так и серверами А. См. *связывание и внедрение объектов*

АВТОМАТИЗИРОВАННАЯ ОБУЧАЮЩАЯ СИСТЕМА (АОС) [automated training system, computer based training system (CBT)]. Комплекс программных, технических и учебно-методических средств, предназначенных для активного индивидуального обучения человека на основе программного управления этим обучением. Работа с АОС ведется в режиме диалога и напоминает диалог ученика с педагогом-репетитором. АОС делятся на специализированные, предназначенные для обучения одному какому-либо предмету, и универсальные, обеспечивающие возможность эффективного обучения нескольким предметам. Основу специализированных систем составляют *обучающие программы*, написанные на традиционных языках программирования, например, на Паскале или Си. К специализированным АОС относятся и тренажеры, управляемые компьютером. Универсальные системы располагают *проблемно-ориентированным языком* и специальными средствами программирования педагогами сценария обучения и программ управления обучением. В состав универсальных АОС входят также *библиотека программ* управления познавательной деятельностью, архив данных о процессе обучения каждого обучаемого, средства *дружественного интерфейса* и т. п. АОС развиваются на базе как серийной вычислительной техники *(персональные*

компьютеры, локальные вычислительные сети), так и специально созданных тренажеров. Обучаемый может работать с АОС как индивидуально, так и в условиях класса автоматизированного обучения. Развитием АОС являются *интеллектуальные обучающие системы.* Ср. *обучающая программа, обучающее приложение к программному продукту*

АВТОМАТИЗИРОВАННАЯ ПОДГОТОВКА [computer-based training (CBT)]. То же, что *обучающее приложение к программному продукту*

АВТОМАТИЗИРОВАННАЯ РАЗРАБОТКА ПРОГРАМ, инструмент CASE [computer-aided software engineering (CASE), CASE-tool]. Набор согласованных средств *разработки программ*, в число которых входят *языки визуального моделирования*, генераторы кода и документации, автоматизированные средства тестирования, редакторы, компиляторы и другие компоненты. Как правило, инструмент А. р. п. ориентирован на определенный *процесс разработки программного обеспечения* и компоненты инструмента А. р. п. строго соответствуют фазам выбранного процесса. Ср. *инструментальные программные средства*

АВТОМАТИЗИРОВАННАЯ СИСТЕМА [automated system]. Система *программных* и *аппаратных средств*, предназначенных для *автоматизации* процесса деятельности человека. В отличие от *автоматической системы* А. с. всегда функционирует при участии человека. И человек является ее главным звеном. См. *автоматизированная обучающая система, автоматизированная система управления, автоматизированное рабочее место, система автоматизированного проектирования*

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ (АСУ) [automated control system]. Комплекс программных и технических средств *автоматизации* управления различными объектами. В отличие от *систем автоматического управления* в АСУ сбор и обработку информации, необходимой для выработки управляющих воздействий, осуществляют автоматические устройства — приборы и компьютеры, а окончательное решение принимает человек. Ср. *офисное приложение*

АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ [computer-aided design (CAD)]. См. *система автоматизированного проектирования*

АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ МЕСТО (АРМ) [automated working place, workstation]. 1. *Вычислительная система*, предназначенная для автоматизации профессиональной деятельности. Основу АРМ составляет ЭВМ (*персональный компьютер* или *рабочая станция*), снабженная специализированным *программным обеспечением.* Наибольшее распространение получили АРМ для проектировщиков, конструкторов, технологов, исследователей, бухгалтеров, администраторов и т. п. АРМ позволяет существенно увеличить производительность труда профессионала. Например, конструкторы имеют

возможность посвятить свое время целиком процессу конструирования, поскольку расчеты и подготовку чертежей машина "берет на себя". С помощью таких устройств, как *световое перо* или *графический планшет*, конструктор может быстро и легко вносить любые изменения в проект и тут же наблюдать результат на экране дисплея. При этом компьютер может представить какую-либо часть чертежа или проектируемой детали в увеличенном масштабе или под различными углами зрения. АРМ работают либо автономно, либо в составе *локальной вычислительной сети*. 2. *Программное обеспечение* автоматизации профессиональной деятельности

АВТОМАТИЧЕСКАЯ СИСТЕМА [automatic system]. Система *программных* и *аппаратных средств*, функционирующая самостоятельно, без участия человека. Ср. *автоматизированная система*

АВТОМАТИЧЕСКИЙ ПЕРЕХОД НА НОВУЮ СТРОКУ [word-wrap]. То же, что *выравнивание текста*. См. *перенос слов*

АВТОМАТИЧЕСКИЙ ПОВТОР НАЖАТИЯ КЛАВИШИ [typematic]. Автоматическое повторение ввода символа или команды, вызванное удерживанием соответствующей клавиши в нажатом состоянии. Например, при однократном нажатии клавиши <Page Down> на экране появится следующая страница, а если клавишу нажать и не отпускать, команда смены страницы будет автоматически повторяться, что приведет к листанию страниц. Продолжительность времени между двумя повторениями для клавиатуры *персонального компьютера* можно регулировать с помощью специальных команд *операционной системы*, например команды `mode` в MS-DOS или с помощью *Панели управления Windows*. Например, на рис. А.1 показано окно для настройки клавиатуры в операционной системе Windows

АВТОМАТИЧЕСКИЙ СИНТЕЗ ПРОГРАММ [automatic program synthesis]. Автоматическая генерация программы по некоторой *спецификации*. В зависимости от метода спецификации программы А. с. п. подразделяют на несколько категорий. Если спецификация задана в виде формального логического условия, связывающего *входные* и *выходные данные*, то говорят о дедуктивном А. с. п. Если спецификация задана в виде набора примеров, то говорят об индуктивном А. с. п., а если в виде формального, но неалгоритмического описания задачи, например, в виде диаграмм, то говорят о трансформационном А. с. п. Для А. с. п. обычно применяют методы *искусственного интеллекта*. В общем случае задача А. с. п. алгоритмически неразрешима, т. е. не существует алгоритма, который бы по произвольной спецификации строил соответствующую программу, однако известно множество частных, но практически важных случаев, в которых А. с. п. возможен. Результаты, получаемые в ходе теоретических исследований по А. с. п., находят практическое применение в *оптимизирующих компиляторах*, *электронных таблицах* и других областях. Ср. *Верификация*

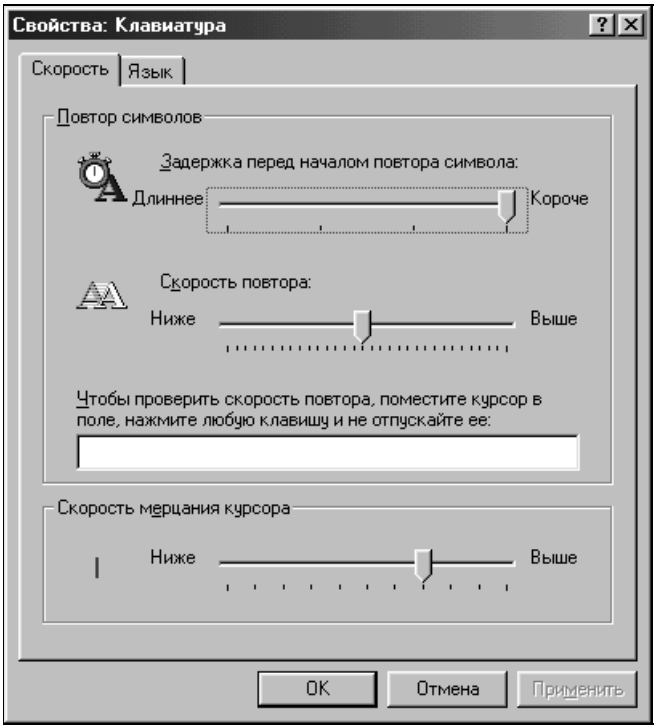

Рис. А.1. Настройка параметров клавиатуры с помошью Панели управления Windows

АВТОМАТИЧЕСКОЕ СОХРАНЕНИЕ, **автосохранение [autosave]**. *Сохранение*, выполняемое программой с заданной периодичностью или после определенного числа нажатий клавиш. А. с. служит гарантией от случайных потерь данных. Оно применяется в приложениях Windows, периодически сохраняя изменения, вносимые в документ. Например, в редакторе MS Word можно установить периодичность А. с. от одной минуты до двух часов

АВТОМАТИЧЕСКОЕ УСТРОЙСТВО, **автомат [automation]**. *Устройство*, способное функционировать без непосредственного участия человека

АВТОНОМНАЯ ПРОГРАММА [autonomy program, **stand-alone program]**. *Программа*, не зависящая от других программ в составе данной программной системы. Например, программа в виде *загрузочного модуля*, имеющая свои собственные средства инициализации и не требующая для своего выполнения управляющих действий *операционной системы*

АВТОНОМНОЕ УСТРОЙСТВО [autonomy device, autonomy unit]. *Устройство*, работающее независимо от других устройств *вычислительной системы*

АВТОНОМНЫЕ ИСПЫТАНИЯ ПРОГРАММЫ [stand-alone program tests]. *Испытания программы* по частям или целиком, но независимо от других программ, с которыми испытуемая программа должна взаимодействовать. Ср. *комплексные испытания программного продукта*. См. *испытание программы*

АВТОРИЗАЦИЯ [authorization]. 1. Процесс проверки имеющихся у пользователя прав и разрешений на доступ к ресурсу. См. *аутентификация*. 2. Предоставление пользователю определенных полномочий на выполнение некоторых работ в вычислительной системе. Ср. *права доступа*

АВТОРСКАЯ СИСТЕМА [authoring system]. То же, что *система автоматизации авторской работы*

АВТОРСКИЙ ЯЗЫК [authoring language]. *Язык программирования высокого уровня*, используемый для создания продуктов мультимедиа, презентаций, а также программ, *баз данных* и материалов для *автоматизированных обучающих систем*

АВТОСОХРАНЕНИЕ [autosave]. То же, что *автоматическое сохранение*

АГЕНТ [agent]. Программа, действующая от имени пользователя, выполняя специфические функции в фоновом режиме. Выполнив задачу, программа А. сообщает об этом пользователю

АГРЕГАЦИЯ [aggregation]. В *унифицированном языке моделирования* UML форма *ассоциации*, описывающая отношение типа "часть—целое" между объектами. Отношение А. является менее строгим по сравнению с отношением *композиции*: часть может принадлежать нескольким целым и прекращение существования целого не означает прекращения существования его частей

АДА [Ada]. Универсальный *язык программирования высокого уровня*, ориентированный на разработку *программного обеспечения* компьютеров, встроенных в системы автоматизированного управления. Назван в честь Августы Ады Лавлейс, считающейся первым в мире программистом. Первая версия А., разработанная в США под руководством Дж. Ишбиа, появилась в 1980 г. Последняя версия, Ада-95 — в 1995 г. А. обладает средствами программирования процессов, протекающих параллельно в реальном времени. Программы на языке А. имеют модульную структуру, допускающую раздельную компиляцию модулей. Кроме того, в стандарте языка А. помимо собственно языка программирования впервые была определена и его поддерживающая *среда программирования*, т. е. средства разработки, корректировки и модернизации программ. Все это позволяет создавать на языке А. очень большие программные проекты, например, программное обеспечение систем астронавигации

АДАПТЕР [adapter]. *Электронная схема*, позволяющая приспособить друг к другу устройства с различными способами представления данных. Например, *аналоговое устройство* нельзя напрямую соединить с *цифровым устройством*: для этого *аналоговый сигнал* должен быть сначала преобразован в цифровой или наоборот. Чтобы связать два несовместимых компонента *аппаратных средств*, требуется *устройство сопряжения*, важной составляющей которого и является А. Обычно А. монтируется на отдельной плате. См. *адаптер дисплея, звуковой адаптер*

АДАПТЕР ДИСПЛЕЯ, **видеоадаптер**, **видеокарта [display adapter**, **videoadapter**, **videocard]**. *Адаптер*, согласующий обмен *графической информацией* между *центральным процессором* и дисплеем. А. д. преобразует данные, поступающие из компьютера в цифровой форме, в соответствующее изображение на экране. *Цифровой код* текста или графического изображения размещается центральным процессором в видеопамяти. А. д. периодически "просматривает" содержимое видеопамяти или ее отдельных частей и по хранящемуся там коду "рисует" нужное изображение на экране дисплея. При этом он устанавливает формат экрана, управляет курсором и цветовыми характеристиками изображения. А. д., как правило, имеет несколько режимов работы, которые характеризуются типом отображаемой информации (текстовая или графическая), количеством используемых цветов, *разрешающей способностью* и размерами символов. См. *текстовый режим, графический режим*. А. д. современных *персональных компьютеров* позволяет выводить на экран дисплея изображения, получаемые с видеокамеры, видеомагнитофона или любого другого источника видеосигнала. Он состоит из четырех основных устройств: видеопамяти, видеоконтроллера, цифроаналогового преобразователя (ЦАП) и постоянного запоминающего устройства (ПЗУ), которые могут монтироваться на отдельной плате, называемой видеоплатой или видеокартой. Видеопамять служит для хранения изображения. Видеоконтроллер отвечает за вывод изображения из видеопамяти на монитор и обработку запросов центрального процессора. ЦАП служит для преобразования формируемых видеоконтроллером цифровых *видеоданных* в аналоговый видеосигнал. В ПЗУ записаны наиболее универсальные программы управления выводом изображения на экран, экранные шрифты, служебные таблицы и т. п. ПЗУ не используется видеоконтроллером напрямую — к нему обращается только центральный процессор, и в результате выполнения им программ происходят обращения из ПЗУ к видеоконтроллеру и видеопамяти. ПЗУ необходимо только для первоначального запуска адаптера и работы в режиме MS-DOS; операционные системы с графическим интерфейсом — Windows или OS/2 — не используют ПЗУ для управления адаптером. Для ускорения вывода изображения на экран на плату адаптера могут устанавливаться дополнительные микросхемы — акселераторы или графические сопроцессоры. В IBM PC применяются следующие А. д., указанные в порядке возрастания графических возможностей: EGA (Enhanced Graphics Adapter) —

улучшенный графический адаптер. Его максимальная разрешающая способность 640×350 пикселов. Может выводить одновременно 16 из 64 цветов. VGA (Video Graphics Adapter) — видеографический адаптер. Имеет текстовый режим 720×400 пикселов и графический режим 640×480 пикселов. Одновременно выводит 256 цветов. SVGA (Super VGA) — улучшенный VGA. Так обычно называются выпускаемые в последнее время различными фирмами платы, превосходящие по своим возможностям VGA. Разрешающая способность адаптеров некоторых фирм доходит до 1600×1280 и более пикселов. Количество выводимых цветов может достигать 16,7 млн. Также добавляются расширенные текстовые режимы вплоть до формата 132×50 знакомест. Многие фирмы выпускают адаптеры с улучшенными характеристиками, поддерживающие нестандартные режимы

АДАПТЕР PCMCIA [PCMCIA-adapter]. Тип *адаптера*, разработанный *Международной ассоциацией производителей карт памяти для персональных компьютеров* (PCMCIA). Предназначен для использования на портативных компьютерах. А. PCMCIA имеет небольшой размер и потребляет мало электроэнергии. Частота, на которой работает шина данных для его подключения, обычно равна 8,33 МГц

АДАПТИВНАЯ ДИФФЕРЕНЦИАЛЬНАЯ ИМПУЛЬСНО-КОДОВАЯ МОДУЛЯЦИЯ (АДИКМ) [adaptive differential pulse code modulation (ADPCM)]. Способ кодирования и *сжатия данных* о звуковых сигналах в цифровой форме. Позволяет передавать оцифрованную речь со скоростью свыше 32 Кбит/с. Применяется при записи звуковых фрагментов на *компакт-дисках*. Стандарт АДИКМ разработан *Международным советом по мультимедиа*

АДАПТИВНАЯ СИСТЕМА [adaptive system]. *Система*, автоматически настраивающаяся на условия применения. А. с. сама может изменять свои функциональные возможности в зависимости от изменения условий внешней среды. Например, *операционная система Windows* автоматически приспосабливается к имеющейся *конфигурации компьютера*

АДАПТИВНОСТЬ [adaptively]. Способность устройств или программ изменять свои параметры в связи с изменениями в них самих или в зависимости от условий применения для повышения эффективности

АДДИТИВНЫЙ ЦВЕТ [additive color]. *Цвет*, порожденный излучающим объектом. При смешении А. ц. осветляются, смешение трех основных А. ц. — красного, зеленого и синего — дает белый цвет. При попарном смешении основных А. ц. получают основные *субтрактивные цвета*

АДИКМ [ADPCM]. То же, что *адаптивная дифференциальная импульсно-кодовая модуляция*

АДМИНИСТРАТОР [administrator]. 1. То же, что *системный администратор*. 2. То же, что *администратор базы данных*

АДМИНИСТРАТОР БАЗЫ ДАННЫХ, **администратор [database administrator]**. Человек или группа лиц, ответственные за состояние, развитие и использование *базы данных* организации или учреждения. А. б. д. обеспечивает работоспособность базы данных, контролирует и поддерживает полноту, правильность, непротиворечивость и *целостность данных*, необходимый уровень *защиты данных*. Он взаимодействует с пользователями и программистами, программы которых используют доступ к базе данных. Роль А. б. д. аналогична роли *системного инженера*, сопровождающего *операционную систему*

АДРЕС [address]. 1. Номер, код или идентификатор места в памяти компьютера, где хранится или куда должна быть записана данная информация. Например, номер байта *оперативной памяти* или номер дорожки *магнитного диска*, имя *внешнего устройства* или *узла сети*. Различают *логические* и *физические* А. Логический А. указывается в *машинной программе*. При загрузке, а также при переадресации он преобразуется в зависимости от места, занимаемого программой и данными в *оперативной памяти*, в физический А., закрепленный в *вычислительной системе* за реально существующим *запоминающим устройством* или его частью. Это преобразование выполняется соответствующим *программным обеспечением* и *аппаратными средствами*. См. *адресация*. 2. Часть команды, определяющая местоположение операнда. 3. Часть сообщения, указывающая адресат

АДРЕС КОМАНДЫ [instruction address]. *Адрес ячейки памяти* или первого байта области памяти, которая занята *командой*

АДРЕС ОПЕРАНДА [source address]. *Адрес ячейки памяти* или первого байта области памяти, откуда извлекается *операнд*

АДРЕС IP [IP-address]. 1. Уникальный *адрес*, идентифицирующий узлы или компьютеры в сети, управляемой *протоколами TCP/IP*. В сети, представляющей собой объединение нескольких подсетей, указывает информацию маршрутизации. А. IP представляет собой 32-разрядное значение, которое идентифицирует данный компьютер. Каждый байт А. IP представляется десятичным числом и отделяется от других десятичной точкой. 2. То же, что *интернет-адрес*

АДРЕСАТ [target]. Обладатель *адреса* (лицо, программа, устройство или компьютер), которому предназначаются сообщение, команда или данные. Например, А. считается и *почтовый ящик*, в который приходит сообщение по *электронной почте*, и пользователь — владелец этого ящика. Также А. является дисковод, на который послал команду *центральный процессор*, и программа, для которой с диска считываются данные

АДРЕСАЦИЯ [addressing]. Определение операнда в *машинной команде*. Способ вычисления *физического адреса* и выполнения операций над *адресными регистрами*. Рассмотрим наиболее распространенную схему А. Чтобы при выполнении программы *устройство управления* могло определить, какую

команду необходимо выполнить в данный момент, и найти данные, указанные в этой команде, вся *основная память* машины делится на отдельные адресуемые ячейки — байты или *машинные слова*. Каждому байту или слову присвоен уникальный *физический адрес*. В традиционной схеме А. эти адреса являются номерами, начиная от нуля с шагом единица. Таким образом, к любой команде или элементу данных можно обратиться по адресу его самого левого элемента. При таком абсолютном методе А. адрес 1234 означает 1235-й байт или машинное слово при условии, что отсчет ведется от нулевой ячейки памяти. Однако при режиме мультипрограммирования место, занимаемое программой в *оперативной памяти*, зависит от наличия и расположения в ней других программ. Поэтому необходимо, чтобы программы были перемещаемыми, т. е. адреса внутри программ должны отсчитываться относительно некоторого начального *базового адреса* программы, который должен быть перемещаемым. Это же требуется и при независимой трансляции подпрограмм и отдельных *программных модулей*, адреса которых также отсчитываются от базового адреса и называются относительными адресами или смещениями. При *загрузке программы* в зависимости от места, занимаемого программой или ее сегментом, определяется значение ее базового адреса, которое посылается в *базовый регистр* процессора, и физический адрес ячейки интерпретируется аппаратурой как сумма содержимого базового регистра и *относительного адреса*. Например, если адресная часть команды содержит 1200, а содержимое базового регистра равно 300, то операнд будет выбираться из ячейки с физическим адресом 1500. Внутри циклов, которые есть почти в каждой программе, работает одна и та же последовательность команд, а изменяются только адрес или величина той или иной переменной. Для определения значения адреса, изменяемого в цикле, применяется так называемый *индексный регистр*, содержимое которого отражает значение параметра цикла. *Адрес операнда* в командах цикла определяется как указанный в команде относительный адрес плюс содержимое базового регистра плюс содержимое индексного регистра. При таком методе А. все данные могут смещаться при перемещении программы путем изменения значения базового регистра, а индексный регистр способствует простой организации выборки в цикле. При *трансляции программ*, написанных на *языках программирования высокого уровня*, компиляторы сами выбирают регистры, необходимые для базирования и индексации, а компоновщики и загрузчики устанавливают их нужные значения. Для уменьшения потерь времени на вычисление физических адресов во время выполнения программы такое вычисление производится специальными быстрыми *аппаратными средствами*

АДРЕСАЦИЯ С ИНДЕКСИРОВАНИЕМ [indexed addressing]. То же, что *индексная адресация*

АДРЕСНАЯ КОНСТАНТА [address constant, A-constant]. *Константа*, которая либо является *адресом*, либо используется при вычислении адресов. Обычно

это числовая константа или числовое выражение, значение которого является адресом или составной частью адреса

АДРЕСНАЯ ССЫЛКА [address reference]. *Ссылка* в форме *адреса*

АДРЕСНАЯ ШИНА [address bus]. *Шина* или часть шины, предназначенная для передачи адреса

АДРЕСНОЕ ПРОСТРАНСТВО ЗАДАЧИ, **адресное пространство процесса [process address space]**. Множество *логических адресов*, по которым может обращаться задача. *Система управления памятью* устанавливает соответствие между А. п. з. и *физической памятью* ЭВМ, и при выполнении задачи для каждого логического адреса вычисляет *физический адрес*. См. *адресация, управление памятью, виртуальная память*

АДРЕСНОЕ ПРОСТРАНСТВО ПРОЦЕССА [process address space]. То же, что *адресное пространство задачи*

АДРЕСНОЕ ПРОСТРАНСТВО ЦЕНТРАЛЬНОГО ПРОЦЕССОРА [CPU address space]. Множество *физических адресов* ячеек *оперативной памяти*, с которыми может работать *центральный процессор*. Для машин, работающих в *двоичной системе счисления*, размер А. п. ц. п. составляет 2^n ячеек, где n — разрядность *адресной шины* процессора. Очевидно, этот размер ограничивает емкость оперативной памяти, на которую рассчитан процессор и к которой он может иметь *прямой доступ*. Реально физическая оперативная память может быть меньше или больше той, которую позволяет иметь А. п. ц. п. Если количество физических адресов, с которыми работает центральный процессор, меньше его адресного пространства, то оперативную память компьютера можно расширить. Для увеличения размера памяти сверх А. п. ц. п. должны применяться специальные программные и аппаратные средства. См. *адресация, управление памятью, виртуальная память*

АДРЕСНЫЙ РЕГИСТР [address register]. То же, что *регистр адреса*

АКСЕССУАРЫ [accessory]. Добавочные принадлежности к компьютеру. Например, сумка для ноутбука

АКТИВАЦИЯ, фокус управления [activation, focus of control flow]. 1. То же, что *активизация*. 2. Выполнение операции. А. представляет собой период времени, в течение которого объект выполняет операцию (непосредственно или с помощью подчиненных операций). Посредством А. моделируется не только длительность выполнения операции, но и отношения *потока управления* между взаимодействующими объектами. А. является одним из основных понятий *унифицированного языка моделирования* UML. В обычных *языках программирования* А. соответствует *вызов процедуры*

АКТИВИЗАЦИЯ [activation]. Приведение объекта в состояние готовности к действию или использованию. Например, А. экранных объектов (панелей, меню, окон, кнопок и пр.) позволяет пользователю начать диалог с системой

АКТИВИЗАЦИЯ ФАЙЛА [file activation]. То же, что *открытие файла*

АКТИВНАЯ ПРОГРАММА [active program]. *Программа*, которой передано управление *центральным процессором*. Противоп. *неактивная программа*

АКТИВНАЯ ЯЧЕЙКА [active cell]. Ячейка электронной таблицы, с которой в настоящий момент работает или может работать пользователь. Она идентифицируется координатами строки и столбца, к которым принадлежит, и выделяется на экране контрастной рамкой (рис. А.2). В А. я. пользователь может ввести данные, отредактировать или отформатировать ее содержимое

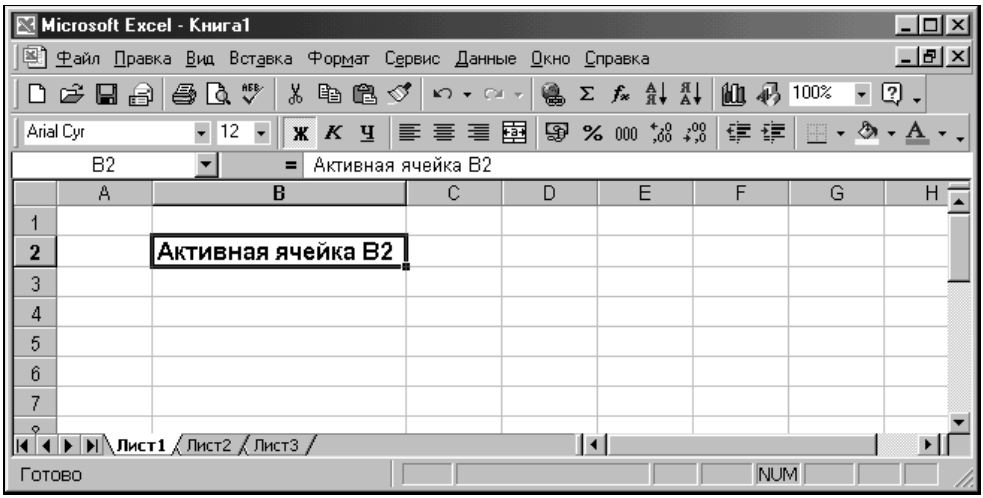

Рис. А.2. Активная ячейка на рабочем листе Microsoft Excel

АКТИВНОЕ ОКНО [active window]. *Окно*, с которым в настоящий момент работает или может работать пользователь. На нем активизированы элементы управления, через него возможен ввод данных. При одновременном отображении на экране нескольких окон (рис. А.3) А. о. не перекрывается другими окнами и выделяется контрастным цветом рамки или заголовка, наличием мерцающего курсора и т. п. Чтобы сделать окно активным, как правило, достаточно навести на него указатель мыши и нажать ее левую кнопку. Противоп. *неактивное окно*

АКТИВНЫЙ ДИСК [active disc]. То же, что *текущий диск*

АКТИВНЫЙ ФАЙЛ [active file]. То же, что *открытый файл*

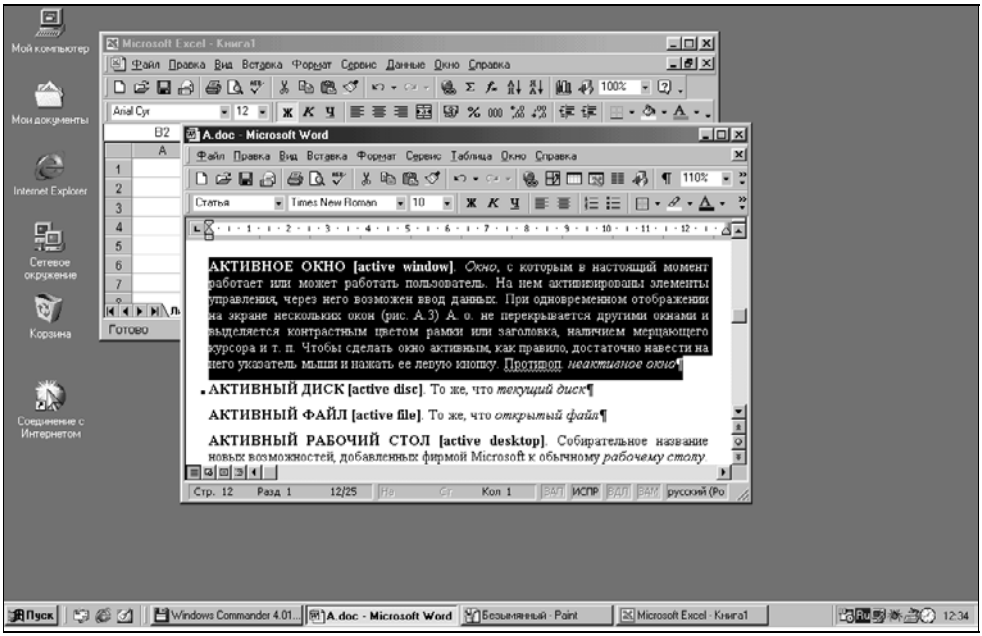

Рис. А.3. Активное окно Microsoft Word и неактивное окно Microsoft Excel

АКТИВНЫЙ РАБОЧИЙ СТОЛ [active desktop]. Собирательное название новых возможностей, добавленных фирмой Microsoft к обычному *рабочему столу*. На А. р. с. можно разместить любой объект *веб-страницы* (встроенную графику, гипертекстовые ссылки и т. п.), *браузера* и любой компонент *Всемирной паутины WWW* (например, доступные по WWW справочники). Веб-страницы, перечисленные в папке **Favorites** (Избранное) или размещенные на А. р. с., автоматически обновляются *операционной системой Windows*. Компонентами А. р. с. также являются веб-объекты, специально предназначенные для расположения на рабочем столе. Их содержимое обычно представляет собой новости, прогноз погоды, биржевые котировки и другую подобную информацию. В компонентах А. р. с. применяются те же технологии, что и в других веб-объектах

АЛГОЛ [Algol]. *Алгоритмический язык*, разработанный в 1958 г. Первоначально А. не был *языком программирования*, так как предназначался лишь для обмена вычислительными алгоритмами между людьми. Название языка произошло от английских слов ALGOrithmic Language (алгоритмический язык) В дальнейшем на основе А. были созданы языки программирования Алгол60 и Алгол68. Первая редакция *языка программирования* Алгол60, ориентированного на программирование численных решений научно-техни-

ческих задач, была разработана П. Науром, Дж. Бэкусом и др. и опубликована в 1960 г. В Алгол60 впервые были реализованы идеи *блочной структуры программ*, определения способов *передачи параметров* в процедуры, *рекурсивных процедур*. Для описания синтаксиса Алгол60 впервые был использован ставший классическим *формальный язык Бэкуса—Наура*. Идеи Алгол60 получили свое развитие во многих языках программирования *высокого уровня*. В 1966—1969 гг. на основе Алгол60 был разработан новый язык Алгол68. По сравнению с Алгол60 он обладает большими возможностями при программировании ввода и вывода данных и обработки текстов. Кроме того, в Алгол68 появились средства описания параллельно выполняемых операторов. Формальная строгость и общность описания сделали язык слишком сложным для широкого распространения. Это послужило толчком для создания более простого *языка Паскаль*

АЛГОРИТМ [algorithm]. Содержание и последовательность операций, точно определяющие решение задачи путем *вычислительного процесса*, преобразующего *исходные данные* в конечный результат. Характеристиками алгоритма являются: 1) однозначность результата при заданных исходных данных; 2) возможность разбиения процесса на конечное число отдельных операций, каждая из которых может быть выполнена человеком или *вычислительной машиной*; 3) способность получения результата для множества исходных данных, соответствующих множеству однотипных задач. Например, один из А. вычисления с помощью компьютера среднего арифметического трех чисел представляет собой следующую последовательность операций: ввод с клавиатуры трех чисел; вычисление суммы введенных чисел; деление полученной суммы на 3; вывод результата на экран дисплея. В приведенном примере для записи А. был применен русский язык. Существуют специально созданные *алгоритмические языки*. Вычислительной машине А. задается в виде программы. Могут существовать несколько А. решения одной и той же задачи. Среди них следует выбирать наиболее эффективный, для вычислительной реализации которого требуется наименьшее количество операций, машинного времени, памяти и т. п. Изучение условий существования и способов построения (разработки) эффективных А. составляет основу теории алгоритмов

АЛГОРИТМИЧЕСКИЙ ЯЗЫК [algorithmic language]. Язык, предназначенный для записи, разработки и изучения *алгоритмов*, а также для обмена алгоритмами между людьми. А. я. является важным средством обучения. Например, при изучении информатики в школах применяется т. н. школьный А. я., в котором используются понятные школьнику *служебные слова* на русском языке. В отличие от *языков программирования*, А. я. независим от ЭВМ. В нем нет деталей, связанных с устройством машины. Обычно при разработке *языка программирования высокого уровня* сначала создается А. я. с тем же названием. Кроме того, *языки программирования* используются для обме-

на алгоритмами и для их опубликования в научных изданиях. Поэтому понятие "А. я." часто применяется в качестве синонима понятия "язык программирования"

АЛТ-ВВОД СИМВОЛОВ [character alt-input]. Кодовый ввод символов с клавиатуры *персонального компьютера*. При нажатой клавише <Alt> следует набрать на *малой цифровой клавиатуре* код символа, после этого клавишу <Alt> отпустить. На экране появится тот символ, код которого был набран. Таким способом можно ввести любой допустимый *вычислительной системой* символ. Обычно при А.-в. с. применяется так называемая *модифицированная альтернативная кодировка ГОСТа* (см. табл. М.2), первые 128 кодов которой являются десятичным представлением двоичного *кода ASCII*

АЛУ [ALU]. То же, что *арифметико-логическое устройство*

АЛФАВИТНО-ЦИФРОВОЕ ПЕЧАТАЮЩЕЕ УСТРОЙСТВО (АЦПУ) [line printer (LPT)]. То же, что *линейный принтер*

АЛЬБОМНАЯ ОРИЕНТАЦИЯ СТРАНИЦЫ, **ландшафтная ориентация страницы [landscape page orientation]**. Такое расположение страницы, когда ее высота меньше ширины. Противоп. *книжная ориентация страницы*

АЛЬФА-ВЕРСИЯ [alpha release]. Предварительная версия программы, предназначенная для *альфа-тестирования*

АЛЬФА-КАНАЛ [alpha channel]. Дополнительный 8-разрядный канал, применяемый в некоторых графических платах кроме трех традиционных RGB-каналов. В таких платах для кодирования цвета пиксела используются 32 бита, причем младшие 24 бита содержат информацию о цвете, соответствующую RGB-каналам (по 8 бит на каждый цвет), а старшие 8 разрядов образуют А.-к. С его помощью можно формировать прозрачные области изображения, не зависящие от того или иного цвета

АЛЬФА-ТЕСТИРОВАНИЕ [alpha testing]. Предварительная стадия *испытания программы* или программного обеспечения, предназначенная для общей оценки будущего *программного продукта* и определения необходимости включения в него тех или иных свойств. Ср. *бета-тестирование*

АМЕРИКАНСКИЙ НАЦИОНАЛЬНЫЙ ИНСТИТУТ СТАНДАРТОВ (АНСИ) [American National Standards Institute (ANSI)]. Организация, определяющая стандарты, на основе которых в США разрабатываются *аппаратные средства* и некоторые виды *программного обеспечения*. Например, для языков С, Basic и SQL имеются стандарты, подготовленные ANSI

АМЕРИКАНСКИЙ СТАНДАРТНЫЙ КОД ОБМЕНА ИНФОРМАЦИЕЙ [American standard code for information interchange, **(ASCII)]**. Стандартная

схема кодирования текстовой информации, при которой каждый *текстовый* или *управляющий символ* представляется семиразрядным *двоичным кодом.* Этот код с отдельными модификациями используется в большинстве *вычислительных систем.* Многие клавиатуры при нажатии клавиши вводят в компьютер именно такой код. Зарубежные *персональные компьютеры* имеют набор ASCII-кодированных символов, в котором отсутствуют символы кириллицы (русские буквы), поэтому были созданы различные модификации таблицы А. с. к. о. и., дополненные буквами кириллицы. Все эти модификации имеют общую кодировку первых 128 символов, совпадающую с А. с. к. о. и. Эти символы и их коды приведены в табл. А.1. Чтобы определить код символа по этой таблице, нужно к первому числу столбца, в котором находится символ, приписать справа первое число его строки. Например, символ # находится на пересечении столбца, первое число которого есть 010, и строки, первое число которой — 0011. Следовательно, его код 0100011. В литературе по программированию и при *Алт-вводе символов* применяется десятичная запись двоичного кода (см. *двоичная система счисления*). Например, код 0100011 символа # в десятичной системе счисления записывается числом 35. См. *символ*

Таблица А.1. *Символы кода ASCII (общая часть всех кодировок)*

	000	001	010	011	100	101	110	111
0000	NUL	DLE	.	0	@	P	‘	p
0001	SON	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	-	o	DEL

Обозначения: NUL — знак пробела; SON — начало заголовка; STX — начало текста; ETX — конец текста; EOT — конец передачи; ENQ — запрос; ACK — подтверждение; BEL — звонок; BS — возврат на одну позицию; HT — горизонтальная табуляция; LF — перевод строки; VT — вертикальная табуляция; FF — подача бланка; CR — возврат каретки; SO — переход на верхний регистр; SI — переход на нижний регистр; DLE — переключение кода; DC1, DC2, DC3, DC4 — управление устройствами 1, 2, 3 и 4 соответственно; NAK — переспрос; SYN — режим синхронного ожидания; ETB — конец передачи блока; CAN — отмена; EM — конец носителя; SUB — замена; ESC — переход; FS — разделитель файла; GS — разделитель группы; RS — разделитель записи; US — разделитель блока; DEL — стирание

АНАЛИЗ ТРЕБОВАНИЙ [requirements analysis]. Фаза *процесса разработки программного обеспечения*, во время которой анализируются *требования* и *предметная область*. На фазе А. т. разработчики фокусируют свое внимание на том, что им предстоит сделать, а на фазе *проектирования* — каким образом они будут это осуществлять

АНАЛОГОВАЯ ВЫЧИСЛИТЕЛЬНАЯ МАШИНА (АВМ) [analog computer]. *Вычислительная машина*, которая обрабатывает информацию, представленную в форме непрерывно изменяемых физических величин, значения которых можно измерить. Такими величинами могут быть угловые или линейные перемещения, электрическое напряжение и пр., являющиеся аналогами математических переменных. Над аналоговыми величинами можно производить сложение, умножение, интегрирование и другие математические действия механическим или каким-либо физическим способом, измеряя результаты этих действий. Наибольшее распространение получили электронные АВМ. В них информация задается в виде электрического напряжения. Для решения математической задачи из элементов электрических цепей набирается электрическая схема, напряжение на выходе которой удовлетворяет тем же уравнениям, что и решение исходной математической задачи. Таким образом, напряжение является аналогом искомой математической переменной. Изменяя во времени параметры схемы, можно получать меняющиеся во времени решения. Поэтому физическое время также является аналоговой переменной в такой АВМ. К недостаткам АВМ относят сравнительно невысокую точность и малую универсальность, поэтому их используют в основном как управляющие устройства в различных системах управления и как измерительные устройства в системах сбора и обработки информации

АНАЛОГОВАЯ ФОРМА ДАННЫХ [analog data form]. Способ представления, хранения и *передачи данных* в виде непрерывно изменяющихся в некотором диапазоне физических величин, таких как электрическое напряжение или электрический ток, или *аналоговых сигналов*. А. ф. д. применяется в *ана-*

логовых вычислительных машинах и различных *аналоговых устройствах.* Противоп. *цифровая форма данных*

АНАЛОГОВОЕ УСТРОЙСТВО [analog device]. *Устройство*, в котором информация представляется в виде непрерывно изменяющихся в некотором диапазоне физических величин, таких как электрическое напряжение или электрический ток. См. *аналоговый сигнал, аналоговая вычислительная машина.* Противоп. *цифровое устройство*

АНАЛОГОВЫЙ СИГНАЛ, **непрерывный сигнал [analog signal**, **continuous signal]**. Носитель информации в виде непрерывно изменяющегося электрического напряжения или электрического тока. Переменная во времени амплитуда А. с. соответствует количественной характеристике переносимой им информации и часто представляет измеренную физическую величину, например, температуру, скорость и т. п. Для обработки переносимой А. с. информации на компьютере необходим *аналого-цифровой преобразователь.* См. *аналоговое устройство, аналоговая вычислительная машина.* Противоп. *дискретный сигнал, цифровой сигнал*

АНАЛОГО-ЦИФРОВАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА, **гибридная вычислительная система [hybrid computer]**. *Вычислительная система*, в которой сочетаются элементы *аналоговых* и *цифровых вычислительных машин.* Обычно в таких системах аналоговые устройства используются для решения уравнений, описывающих исследуемое явление или процесс в натуральном масштабе времени, а цифровые устройства — для хранения результатов решения, их последующей обработки и выдачи полученных данных в форме, удобной для восприятия человеком. Для преобразования непрерывно меняющихся во времени аналоговых величин в эквивалентные значения *цифровых кодов* здесь применяются специальные устройства — аналого-цифровые преобразователи. См. *аналоговый сигнал, цифровой сигнал*

АНАЛОГО-ЦИФРОВОЙ ПРЕОБРАЗОВАТЕЛЬ (АЦП) [analog-to-digital converter (ADC)]. Устройство для автоматического преобразования *аналогового сигнала* в *цифровой сигнал.* Например, для автоматической обработки экспериментальных данных поступающие с измерительных приборов непрерывно меняющиеся во времени аналоговые величины с помощью АЦП преобразуются в эквивалентные значения *цифровых кодов*, которые вводятся в компьютер. Противоп. *цифроаналоговый преобразователь*

АНИМАЦИОННАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ [animation path]. Последовательность изображений, быстрая смена которых на экране создает иллюзию движения объекта (рис. А.4). См. *анимация, твининг*

Рис. А.4. Анимационная последовательность

АНИМАЦИЯ [animation]. Создание иллюзии движения объекта на экране дисплея. В основе А. лежит быстрая смена последовательно смещаемых относительно друг друга образов. Существует несколько способов А., реализуемых в различных *системах программирования*. При одном способе на экране быстро рисуются и стираются последовательные фазы движения. При другом фазы заранее рисуются в виде полноэкранных кадров и запоминаются в памяти компьютера, чтобы затем последовательно отобразиться на экране. В развитых системах применяются средства А., для которых достаточно задать изображение объекта и его начальное и конечное положение. Качество А. зависит от скорости смены кадров. Чем выше скорость, тем плавнее движение. Современные компьютеры позволяют выполнять до 30 смен кадров в секунду

АНТИВИРУСНАЯ ПРОГРАММА [antivirus program]. *Обслуживающая программа*, предназначенная для поиска, диагностики, профилактики и "лечения" файлов, зараженных *компьютерным вирусом*. В процессе поиска и диагностики определяются зараженные файлы и тип вируса. Профилактика позволяет предотвратить заражение. Например, *резидентная* А. п. предотвращает несанкционированное пользователем изменение файлов операционной системы, запись в сектор начальной загрузки и т. п. Лечение подразумевает удаление вируса, восстановление поврежденных файлов и т. п.

АОС [automated training system]. То же, что *автоматизированная обучающая система*

АППАРАТНОЕ ОБЕСПЕЧЕНИЕ, **техническое обеспечение [hardware]**. Совокупность входящих в состав *вычислительной системы аппаратных средств*, необходимых для ее функционирования. А. о. составляют *процессоры*, *внешние устройства*, *линии связи* и т. д. Они технически обеспечивают эффективную работу системы, ее способность предоставлять пользователю определенные виды обслуживания. Этот термин определяет часть вычислительной системы, не являющуюся *программным обеспечением*

АППАРАТНОЕ ПРЕРЫВАНИЕ [hardware interrupt]. *Прерывание*, причиной которого является событие, произошедшее в *аппаратных средствах*. См. *обработка прерываний, внешнее прерывание, внутреннее прерывание, прерывание от схем контроля*

АППАРАТНО-НЕЗАВИСИМЫЙ [hardware independent, device independent]. Независимый от технических характеристик *аппаратных средств* конкретной *вычислительной системы*. Например, А.-н. формат хранения точечных рисунков позволяет воспроизвести рисунок максимально точно независимо от *разрешающей способности* и других параметров принтера, влияющих на качество изображения

АППАРАТНЫЕ СРЕДСТВА, технические средства, оборудование [hardware]. Электрические, электронные и механические схемы, блоки, приборы и устройства, составляющие материальную часть *вычислительной системы*. А. с. являются, например, и сам компьютер, и микросхемы, его составляющие. К А. с. также относятся дисплей, дисковод, принтер и пр. Противоп. *программные средства*

АППАРАТНЫЙ [hardwired]. Реализованный *аппаратными средствами*. Противоп. *программный*

АППАРАТНЫЙ БЛОК, блок [hardware block, block]. Конструктивно оформленная как единое целое совокупность взаимосвязанных устройств либо совокупность взаимосвязанных элементов или узлов одного устройства, выполняющих определенную функцию. Например, блок питания, обеспечивающий компьютер электроэнергией с заданными показателями, или *системный блок*, объединяющий основные устройства *персонального компьютера*

АППАРАТНЫЙ КЛЮЧ [hardware key]. Реализованный *аппаратными средствами* ключ *защиты от несанкционированного доступа* и *несанкционированного копирования*. Например, при защите программ А. к. часто представляет собой устройство, присоединяемое к параллельному порту персонального компьютера. Защищаемая программа обращается к параллельному порту и запрашивает секретный код. Если А. к. к компьютеру не присоединен, то защищаемая программа определяет ситуацию нарушения защиты и выполняет соответствующие действия. Обычно А. к. сконструирован таким образом, что его присоединение не мешает пользоваться другими устройствами, например, принтером

АППАРАТНЫЙ СБОЙ, машинный сбой, сбой [hardware failure, machine failure, failure]. Кратковременный отказ или ошибка в работе *аппаратных средств вычислительной системы*. Причинами, вызывающими А. с., как правило, являются изменение условий или несоблюдение правил эксплуатации оборудования, например, нестабильность питания, несоблюдение температурного режима, ненадежность соединений, попадание посторонних частиц в подвижные части и т. п.

АППАРАТУРА [hardware, equipment]. Приборы и устройства, входящие в состав *аппаратных средств вычислительной системы*. Физическое оборудование компьютера

АППЛЕТ [applet]. *Прикладная программа.* Термин А. получил распространение в последнее время в связи с технологиями *Всемирной паутины WWW*, позволяющей внедрять А. в *веб-документы* с целью построения интерактивных *веб-страниц.* Для создания таких А. обычно применяется *язык программирования Java*, поэтому часто их называют апплетами Java

АППЛИКАЦИЯ, **графическая заготовка**, **художественная заготовка**, **клипарт [clip art]**. Заранее приготовленный графический рисунок или его фрагмент, предназначенные для *вставки* в документ. Современные *графические* и *текстовые редакторы*, а также *настольные редакционно-издательские системы* поставляются с библиотеками А. На рис. А.5 показано диалоговое окно библиотеки А. приложения MS Word

АРГУМЕНТ [argument]. Выражение, задающее объект или значение при обращении к процедуре, функции или макрокоманде, от которого зависит результат ее выполнения. См. *фактический параметр*

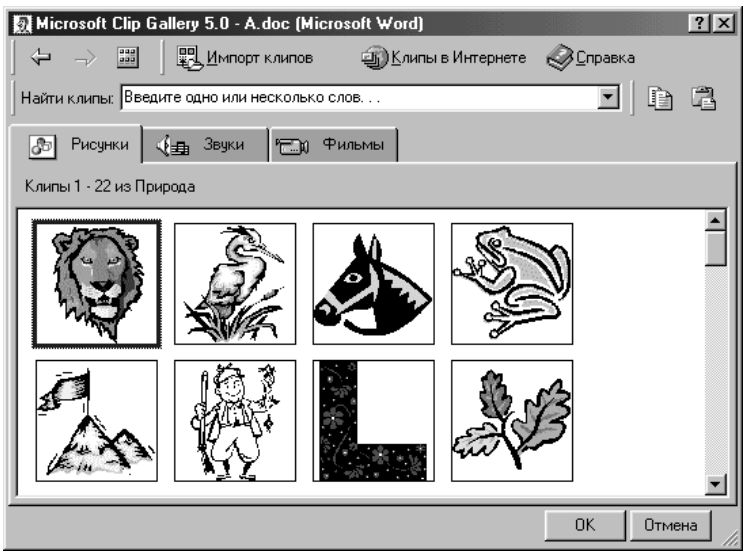

Рис. А.5. Диалоговое окно библиотеки аппликаций приложения Microsoft Word

АРИФМЕТИКО-ЛОГИЧЕСКОЕ УСТРОЙСТВО (АЛУ), **арифметическое устройство (АУ) [arithmetic and logical unit (ALU)]**. Часть процессора, выполняющая над данными *арифметические, логические* и другие *машинные операции.* Под контролем *устройства управления* в АЛУ из *оперативной памяти* поступают операнды, а из устройства управления — управляющие сигналы, указывающие, какую операцию над этими операндами следует вы-

полнить и куда поместить результат по окончании работы. Например, при выполнении сложения двух чисел устройство управления обеспечивает выборку этих чисел из соответствующих ячеек оперативной памяти в АЛУ. Затем оно посылает сигнал "сложить" и после выполнения АЛУ операции сложения пересылает результат в нужную *ячейку памяти* или *регистр*. См. *функциональная схема ЭВМ*

АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ [arithmetical operations]. *Операции*, выполняемые в соответствии с правилами арифметики. В *арифметико-логическом устройстве* операндами и результатом А. о. является число. В *языках программирования* операндами и результатом А. о. могут быть выражение, указатель и др. В набор А. о. обычно входят: сложение (обозначается знаком +), вычитание (-), умножение (× или знак *, который труднее спутать с буквой x), деление (/) и возведение в степень (** или ↑). В языке программирования Си введены дополнительно две *одноместные* и одна *двуместная* А. о.: операция увеличения, или инкремент (+ +) и операция уменьшения, или декремент (- -) и операция деления по модулю (%). Операция увеличения увеличивает, а операция уменьшения уменьшает значение своего операнда на единицу. Существуют две возможности использования этих операций: первая, когда символы + + (- -) находятся слева от операнда, — "префиксная" форма, и вторая, когда эти символы стоят справа от операнда, — "постфиксная" форма. Например, если a = 5 и b = 5, то в результате операций + + a и - - b переменные a и b получат значения: a = 6, b = 4. Тот же результат будет получен для a + + и b – -. Однако префиксная и постфиксная формы различаются между собой, когда операция и ее операнд являются частью выражения. При этом действует правило: в префиксной форме значение операнда изменяется перед его использованием в выражении, а в постфиксной форме — после того, как значение будет использовано в выражении. Например, при a = 5 после выполнения *оператора присваивания* c = + + a будет c = 6 и a = 6, а после выполнения оператора c = a + + будет c = 5, a = 6. Двуместная операция деления по модулю используется только в арифметике *целых чисел*. Ее результатом является остаток от деления целого числа, стоящего слева от знака операции %, на число, расположенное справа от него. Например, результатом *арифметического выражения* 7 % 3 (читается как "7 по модулю 3") будет 1, поскольку 7 = 2 % 3 + 1

АРИФМЕТИЧЕСКИЙ ОПЕРАТОР [arithmetic operator]. В *языках программирования — знак арифметической операции*

АРИФМЕТИЧЕСКИЙ СДВИГ [arithmetic shift]. Одновременный сдвиг всех разрядов числа без изменения позиции знака числа при сдвиге влево и с занесением значения знакового разряда в освободившиеся разряды при сдвиге вправо

АРИФМЕТИЧЕСКИЙ УСЛОВНЫЙ ОПЕРАТОР [arithmetic IF statement]. *Условный оператор* в языке Фортран, в котором проверяемое условие задается в форме *арифметического выражения*. А. у. о. имеет вид: `IF(A) m1, m2, m3`, где `A` — арифметическое выражение, `m1`, `m2`, `m3` — метки исполняемых операторов. Если значение выражения `A` отрицательно (`A < 0`), то выполнение программы продолжается с оператора, помеченного меткой `m1`, если `A = 0`, то управление передается оператору, помеченному меткой `m2`, если же `A > 0`, то оператору, помеченному меткой `m3`. Например, после оператора `IF (K – 2) 23, 7, 10` при `K < 2` управление передается оператору, помеченному меткой `23`, при `K = 2` — оператору, помеченному меткой `7`, и при `K > 2` — оператору, помеченному меткой `10`

АРИФМЕТИЧЕСКИЙ ЦИКЛ [arithmetical cycle]. То же, что *цикл со счетчиком*

АРИФМЕТИЧЕСКОЕ ВЫРАЖЕНИЕ [arithmetic expression]. Совокупность одной или нескольких констант, переменных, *переменных с индексами* и функций, соединенных знаками *арифметических операций* и скобками. Например, А. в. являются: `12.34`, `ALFA`, `(A + B – C) * cos(3.14 * B)`. Значением А. в. является число. Последовательность выполнения операций в А. в. определяется скобками, а при их отсутствии устанавливается следующий порядок: 1) вычисление функций; 2) возведение в степень; 3) умножение и деление; 4) сложение и вычитание. Однотипные операции обычно выполняются последовательно слева направо

АРИФМЕТИЧЕСКОЕ УСТРОЙСТВО (АУ) [arithmetical unit]. То же, что *арифметико-логическое устройство*

АРМ [automated working place]. То же, что *автоматизированное рабочее место*

АРТЕФАКТ [artifact]. Элемент информации, используемый или порождаемый в процессе разработки *программного обеспечения* (например, внешний документ или результат работы). А. может быть, например, модель, техническое задание или *программный продукт*

АРХИВ [archive]. Совокупность образовавшихся в результате деятельности пользователя *вычислительной системы* рабочих материалов и средств их долговременного хранения. Рабочими материалами являются копии документов, редко используемых программ, данных и т. п. В качестве средств долговременного хранения удобно использовать *магнитные ленты и магнитные диски*. В целях экономии памяти *архивные файлы* предварительно сжимают с помощью архиватора. Правильно организованный А. оберегает пользователя от потерь при любых аварийных ситуациях в компьютере. См. *архиватор*

АРХИВАТОР [archivator]. Программа или комплекс программ, выполняющие *сжатие (архивирование) файлов* для более компактного длительного хранения во *внешней памяти* и восстановление сжатых файлов в первоначальном виде *(разархивирование)*. В *персональных компьютерах* наиболее рас-

пространены А. pkzip.exe/pkunzip.exe, arj.exe, rar.exe и в последнее время winzip.exe и winrar.exe. Все они предоставляют возможность более компактного хранения файлов на дисках и позволяют: 1) упаковывать все, отдельные или соответствующие какому-либо шаблону файлы рабочего каталога в *архивный файл*; 2) добавлять файлы в архивный файл; 3) удалять файлы из архивного файла; 4) просматривать содержимое архивного файла; 5) извлекать из архивного файла отдельные файлы или распаковывать все файлы архива; 6) создавать самораспаковывающиеся архивные файлы. См. *сжатие данных*

АРХИВАЦИЯ [archivation]. То же, что *архивирование*

АРХИВИРОВАНИЕ, **архивация [archivation]**. *Сжатие* одного или более *файлов* с целью экономии памяти и размещение сжатых данных в одном *архивном файле*. А. выполняется специальной программой — *архиватором*

АРХИВНЫЙ ФАЙЛ [archived file]. 1. *Файл*, помещенный в *архив*. 2. *Файл*, являющийся результатом сжатия *архиватора*. В один А. ф. могут быть помещены несколько сжатых файлов. См. *архиватор, сжатие файла*

АРХИТЕКТУРА [architecture]. Организационная структура системы, включающая в себя разделение системы на части, связи между этими частями, механизмы взаимодействия и основные принципы проектирования системы. См. *архитектура вычислительной системы*, *архитектура компьютера*, *архитектура программного обеспечения*

АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ [computer system architecture]. Общая логическая организация *вычислительной системы*, определяющая процесс *обработки данных* и включающая *архитектуру ЭВМ*, а также структуру и характеристики *программного обеспечения* и принципы его взаимодействия с *аппаратными средствами*

АРХИТЕКТУРА КЛИЕНТ-СЕРВЕР [client-server architecture]. Способ организации взаимодействия программ или компонентов многокомпонентной программы, подразумевающий наличие программы или компонента программы, называемой *сервером*, и одной или нескольких других программ или компонентов, называемых *клиентами*. Клиент имеет возможность асинхронно для сервера инициировать выполнение процедур сервера и получать результаты их выполнения. Как правило, А. к.-с. обеспечивает возможность нескольким клиентам взаимодействовать с сервером параллельно и независимо друг от друга. Программы, являющиеся информационными системами для доступа к базам данных, часто построены по А. к.-с. На основе А. к.-с. организовано взаимодействие программ, обеспечивающих работу Интернета. См. *автоматизация, Всемирная паутина, связывание* и *внедрение объектов*

АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ [software architecture]. Общая логическая организация системы *программного обеспечения*, вклю-

чающая в себя разделение системы на модули, связи между ними, механизмы взаимодействия компонентов и основные принципы проектирования системы. См. *архитектура*

АРХИТЕКТУРА СЕТИ, **сетевая архитектура [network architecture]**. 1. Общая логическая и техническая организация *компьютерной сети*, выраженная в виде совокупности сетевых аппаратных и программных решений, методов доступа и используемых протоколов. 2. Совокупность сетевых аппаратных и программных решений, методов доступа и используемых протоколов. Например, см. *архитектура SNA*

АРХИТЕКТУРА, УПРАВЛЯЕМАЯ МОДЕЛЬЮ [model driven architecture (MDA)]. Концепция *архитектуры программного обеспечения*, в которой основным артефактом *процесса разработки программного обеспечения* является визуальная модель. В качестве *визуального языка моделирования* используется *унифицированный язык моделирования* UML. А. у. п. предложена *Консорциумом по объектным технологиям*

АРХИТЕКТУРА ФОН НЕЙМАНА, фоннеймановская архитектура [von Neumann architecture]. *Архитектура ЭВМ*, разработанная Джоном фон Нейманом. Она основана на следующих принципах: а) основными блоками вычислительной машины являются 1) *центральный процессор*, состоящий из *устройства управления* и *арифметико-логического устройства*, 2) *память* и 3) *устройства ввода/вывода*; б) программа и данные хранятся в одной и той же линейно адресуемой памяти; в) устройство управления выбирает команды из памяти по очереди, одну за другой, интерпретирует их, соответственно управляет арифметико-логическим устройством, которое выполняет арифметические и логические операции над вызываемыми из памяти данными, и возвращает результат в память; г) устройства ввода/вывода используются для ввода программ и данных и вывода результатов расчетов

АРХИТЕКТУРА ЭВМ [computer architecture]. 1. С точки зрения пользователя — это совокупность основных характеристик компьютера, таких как *система команд*, организация памяти, система адресации, операции ввода/вывода и управления и т. п. Компьютеры, имеющие одинаковую А. ЭВМ, с точки зрения программиста, являются совместимыми. Это означает, что независимо от конкретного исполнения их деталей и блоков компьютеры должны быть способны выполнить одну и ту же программу с одинаковым результатом. 2. С точки зрения разработчика *вычислительной системы* и конструктора *аппаратных средств* — это принцип действия, конфигурация и взаимное соединение основных логических узлов компьютера

АРХИТЕКТУРА CISC [complex instruction set computing (CISC)]. *Архитектура* процессора со сложной *системой команд*, каждая из которых выполняется в течение нескольких тактов процессора. Название дано в противоположность *архитектуре RISC* — процессоров с сокращенной системой команд,

большинство из которых выполняется за один такт. К процессорам А. CISC относятся, например, широко распространенные процессоры компании Intel: от 8086 до 80486, а также процессоры семейства Pentium. Но в элементах этих процессоров часто используется архитектура RISC

АРХИТЕКТУРА EISA [extended industry standard architecture (EISA)]. То же, что *расширенная архитектура промышленного стандарта*

АРХИТЕКТУРА ETHERNET [architecture Ethernet]. *Архитектура локальной вычислительной сети шинной топологии*, разработанная компанией Xerox в 1976 г. с последующей доработкой компаниями Intel и DEC. В А. Ethernet управление доступом основано на *методе CSMA/CD*. На основе А. Ethernet *Институтом инженеров по электротехнике и электронике* (IEEE) и *международной организацией по стандартизации* (ISO) были разработаны стандарты IEEE 802.3 и ISO 8802.3, регламентирующие технологию *сетей Ethernet*. Связь между компьютерами в такой сети выполнена на коаксиальных кабелях. Скорость *передачи данных* до 100 Мбит/с. В одной сети может работать до 1024 *рабочих станций*

АРХИТЕКТУРА MDI [multi-document interface, multiple document interface (MDI)]. То же, что *многодокументный интерфейс*

АРХИТЕКТУРА RISC [reduced instruction set computing (RISC)]. *Архитектура* процессора с сокращенной *системой команд*, которые используются последовательно для выполнения более сложных операций. А. RISC характеризуется командами фиксированной длины, большим набором операций типа регистр—регистр и отсутствием *косвенной адресации*. Благодаря несложности команд и большому количеству регистров почти все команды занимают только один такт процессора, и поэтому он может работать на более высокой тактовой частоте, чем большинство процессоров *архитектуры CISC*. А. RISC обеспечивает простоту конструкции процессора и облегчает разработку трансляторов

АРХИТЕКТУРА SNA [Systems Network Architecture (SNA)]. То же, что *системная сетевая архитектура*

АСИНХРОННАЯ ПЕРЕДАЧА [asynchronous transmission]. То же, что *асинхронная связь*

АСИНХРОННАЯ СВЯЗЬ, асинхронная передача [asynchronous connection, asynchronous transmission]. Режим *передачи данных*, при котором временной интервал передачи символа может изменяться, поскольку ограничен специальными синхронизирующими сигналами. Следовательно, при А. с. каждый передаваемый символ требует посылки дополнительных сигналов, но при этом не нужна синхронизация передающего и приемного устройств. Ср. *синхронная связь*

АСПЕКТО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ [aspect oriented programming (AOP)]. Новейшая *парадигма программирования*, нацеленная на упрощение программирования типичных и повторяющихся фрагментов программ. Например, пусть в программе требуется записывать определенную информацию в *файл протокола*. При обычном программировании потребуется вставить соответствующий фрагмент кода в каждое место программы, где требуется производить запись в файл протокола, причем эти фрагменты будут единообразны. В А. о. п. такой фрагмент (он называется аспектом) требуется написать только один раз, а также указать, в каком контексте данный аспект должен быть применен. После этого аспекто-ориентированная *система программирования* автоматически сгенерирует необходимые фрагменты исходного кода и вставит их в нужные места

АССЕМБЛЕР [assembler]. Общепринятое название транслятора с автокода

АССОЦИАЦИЯ [association]. В *унифицированном языке моделирования* UML семантическое отношение между двумя и более *классификаторами*, включающее в себя связи между их экземплярами. См. *агрегация*, *композиция*

АССОЦИАЦИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ [association for computing machinery (ACM)]. Профессиональная организация предприятий и частных лиц, ведущих деятельность в сфере компьютерных технологий, в частности, занимающихся *разработкой программ*

АССОЦИАЦИЯ СТАНДАРТОВ ПО ВИДЕОТЕХНИКЕ, индустриальный консорциум VESA, ассоциация VESA [Video Electronics Standards Association (VESA)]. Организация, занимающаяся разработкой стандартов видеоадаптеров и мониторов. Объединяет более 50 видеотехнических фирм. Консорциумом VESA была предложена специальная графическая *шина VLB*, обеспечивающая ускорение вывода изображения на экран

АССОЦИАЦИЯ VESA [Video Electronics Standards Association (VESA)]. То же, что *ассоциация стандартов по видеотехнике*

АСУ [automated control system]. То же, что *автоматизированная система управления*

АТОМАРНЫЙ [atomic]. Действие или операция, выполняемые в виде единого блока, которые нельзя выполнить частично или прервать внешним событием. Как правило, атомарные операции невелики по размеру и просты (например, присваивание, простые арифметические или строчные вычисления). Атомарное вычисление происходит в определенной точке последовательности выполнения программы

АТРИБУТ [attribute]. 1. Признак или свойство, характеризующие объект. Например, атрибутами выводимых на экран символов являются шрифт, цвет,

размер и т. п. 2. В *реляционных базах данных* — имя одного и того же поля всех *кортежей* одной таблицы. 3. В *языках разметки* — свойство элемента. 4. В *унифицированном языке моделирования* UML А. называется именованное свойство какого-либо класса. Каждый объект этого класса содержит отдельное значение данного свойства. Ср. *свойство класса*

АУ [arithmetic unit]. То же, что *арифметико-логическое устройство*

АУДИОВИЗУАЛЬНЫЙ [audiovisual]. Определение, относящееся к данным, которые содержат изображение и его звуковое сопровождение, или к аппаратуре, воспроизводящей такие данные

АУДИОДАННЫЕ, аудиоинформация [audio data, audio information]. Информация о звуковых образах, представленная в цифровой форме, предназначенная для хранения, обработки и воспроизведения средствами *вычислительной техники*, а также для передачи по *линиям связи*. Примерами А. являются человеческая речь или музыкальное сопровождение видеоизображений, введенные в память компьютера. Ср. *видеоданные*.

АУДИОИНФОРМАЦИЯ [audio information]. 1. Звуковые образы, человеческая речь. 2. То же, что *аудиоданные*

АУТЕНТИФИКАЦИЯ [authentication]. Процедура проверки правильности введенной пользователем регистрационной информации для *входа в систему*. А. используется для принудительного ограничения *прав доступа* к ресурсам и прав на выполнение операций в системе. См. *пароль*

АУТЕНТИФИКАЦИЯ ЗАПРОС/ОТВЕТ [challenge/response authentication]. То же, что *аутентификация с запросом и ответом*

АУТЕНТИФИКАЦИЯ ПО СЕРТИФИКАТУ КЛИЕНТА [client certificate authentication]. Метод *аутентификации*, в котором идентичность пользователя проверяется без использования пароля. Проверка выполняется путем вычислений, в которых используются открытый ключ, хранящийся в *цифровом сертификате* пользователя, и закрытый ключ, находящийся в компьютере пользователя. См. *шифрование с открытым ключом, шифрование с закрытым ключом*

АУТЕНТИФИКАЦИЯ С ЗАПРОСОМ И ОТВЕТОМ, аутентификация запрос/ответ [challenge/response authentication]. Защищенная процедура *аутентификации*, при которой сервер для контроля доступа использует алгоритмы типа запрос/ответ и *систему безопасности операционной системы*

АЦП [analog-to-digital converter]. То же, что *аналого-цифровой преобразователь*

АЦПУ [LPT]. То же, что *линейный принтер*

АЭРОГРАФ [air brush]. *Инструмент машинной графики*, предназначенный для имитации разбрызгивания *основным цветом*. На рис. А.6 для *графического редактора* Adobe Photoshop показаны кнопка вызова А. и диалоговые окна, позволяющие выбрать сечение сектора разбрызгивания и установить параметры А., определяющие степень разбрызгивания краски (нажим) и длину мазка (постепенный переход к прозрачному или к фону).

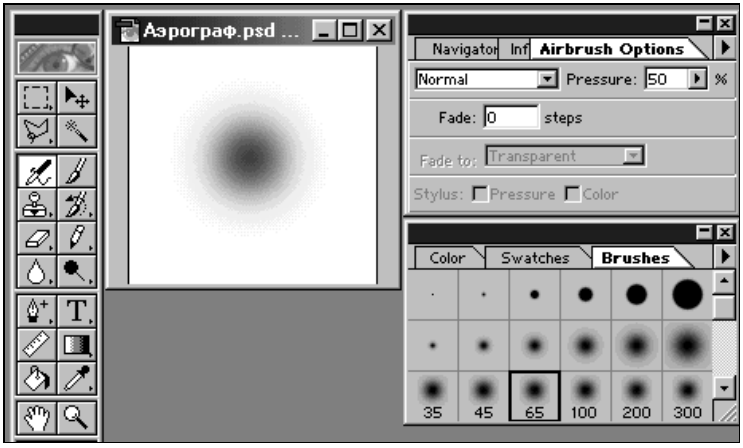

Рис. А.6. Нажатая кнопка вызова, образец работы и окна инструмента **Аэрограф** графического редактора Adobe Photoshop

Б

Б [byte]. То же, что *байт*

БАЗА ДАННЫХ (БД) [data base, database (DB)]. Один или несколько специальным образом организованных файлов, хранящих систематизированную информацию, для доступа к которой используются программные средства *системы управления базой данных* (СУБД). Б. д. представляет собой хранилище связанных между собой данных, не зависящих от программ, которые к ним обращаются. Например, в Б. д. предприятия хранятся конструкционные, технологические, производственные, эксплуатационные, экономические и т. п. характеристики изделий предприятия и их аналогов, выпускаемых конкурирующими фирмами. Такой Б. д. могут пользоваться специалисты из отдела маркетинга (для прогноза конкурентоспособности изделий на рынке), плановики (для определения необходимых объемов выпуска продукции), конструкторы (для анализа конструктивных особенностей изделий конкурентов) и т. д. Б. д. должна быть достаточно полной, правильно организованной, всегда соответствовать текущему моменту времени и удобной для использования. Содержащиеся в ней данные не должны противоречить друг

другу. Редактирование, пополнение и исключение данных, их поиск и сортировка осуществляются под управлением СУБД. Б. д. могут быть персональными и коллективного пользования. Сопровождение крупных Б. д. коллективного пользования осуществляет *администратор базы данных*. Ср. *банк данных*

БАЗА ДАННЫХ КАТАЛОГА [directory database]. *База данных*, содержащая информацию *системы безопасности* (имена *учетных записей пользователей* и пароли, а также параметры настройки *политики безопасности*). Б. д. к. Windows NT Workstation управляется с помощью утилиты User Manager, Б. д. к. домена Windows NT Server управляется с помощью утилиты User Manager for Domains

БАЗА ЗНАНИЙ [knowledge base]. Один или несколько специальным образом организованных файлов, хранящих систематизированную совокупность понятий, правил и фактов, относящихся к некоторой предметной области. Например, Б. з. по химии углеводородов. Содержимое Б. з. оформляется, связывается между собой и представляется таким образом, чтобы на его основе можно было с помощью специальных программ осуществлять рассуждения и делать выводы, получая сведения, которые в явном виде могут не присутствовать в Б. з. Для построения Б. з. применяются методы *искусственного интеллекта*, специальные языки описания знаний и *интеллектуальный интерфейс*. Б. з. являются основной содержательной частью *интеллектуальных информационных систем, интеллектуальных обучающих систем, интеллектуальных систем программирования* и *экспертных систем*, где с их помощью представляются навыки и опыт экспертов — специалистов в данной предметной области. Например, Б. з. в области медицины содержит накопленные медицинскими специалистами сведения о связях между болезнями, их симптомами и порождающими причинами, описания стереотипных ситуаций при течении болезни и рекомендуемых или предпринятых лечебных действиях и их результатах и т. п. На основе такой Б. з. разрабатываются экспертные диагностические и прогнозирующие медицинские системы

БАЗОВАЯ СИСТЕМА ВВОДА/ВЫВОДА [Basic Input/Output System (BIOS)]. Система программ на *машинном языке*, управляющая *передачей данных* между *аппаратными средствами вычислительной системы*. Точнее, BIOS управляет взаимодействием узлов компьютера при передаче данных между процессором, памятью, дисками или монитором, связывая их воедино. От качества BIOS существенно зависит *производительность компьютера*. В современных *персональных компьютерах* BIOS записана в *постоянное запоминающее устройство* — специальную микросхему, установленную на *материнской плате*. Как правило, BIOS поддерживает определенный стандарт интерфейса устройств и ориентируется на конкретную модель материнской платы. Программисты высокой квалификации, создавая *прикладные программы*, непосредственно используют некоторые функции BIOS

БАЗОВЫЙ АБСТРАКТНЫЙ КЛАСС [abstract base class]. То же, что *абстрактный класс*

БАЗОВЫЙ АДРЕС [base address]. *Адрес*, относительно которого идет отсчет *относительных адресов*, указанных в командах программы. Это может быть адрес начала *области памяти*, где располагается программа или данные. Б. а. позволяет процессору вычислять *физические адреса* команд и данных во время выполнения программы. Применение Б. а. делает программу независимой от места ее расположения в *оперативной памяти* во время выполнения. См. *адресация*

БАЗОВЫЙ РЕГИСТР, **регистр базы [base register]**. *Регистр*, предназначенный для хранения *базового адреса*

БАЙТ (Б) [byte]. 1. Фрагмент *двоичного кода*, состоящий из 8-ми соседних двоичных цифр. 2. Единица измерения количества информации, равная количеству информации, которое содержится в 8-ми *двоичных разрядах*. 3. В вычислительной технике байт представляет собой объединение из 8-ми соседних двоичных разрядов, которыми компьютер может оперировать как одним целым. Так, в *персональных компьютерах* каждый байт *оперативной памяти* имеет свой адрес и предназначен для хранения одной буквы, *специального символа* или двух десятичных или шестнадцатеричных цифр. 1 байт = 8 бит

БАЙТ-КОД [bytecode]. *Система команд виртуальной машины Java*

БАНК [bank]. 1. То же, что *страница памяти*. 2. То же, что *банк памяти*. 3. То же, что *банк данных*

БАНК ДАННЫХ [data bank]. Система файлов и *баз данных*, предоставляющая услуги по хранению и поиску данных по одной предметной области (например, астрономические наблюдения, статистика торговли и пр.). Пользователи Б. д. могут находиться в разных учреждениях и даже регионах, получая доступ к данным по *компьютерной связи* или по почте. Ср. *база данных*

БАНК ПАМЯТИ, банк, секция памяти [memory bank, bank, memory section]. Независимая область *оперативной памяти*, имеющая фиксированную длину. В многопроцессорных комплексах с *разделяемой памятью* деление на Б. п. позволяет совместить во времени обращение к памяти разных процессоров и функциональных устройств. Ячейки памяти могут быть пронумерованы непрерывным образом. Например, в первом Б. п. находятся ячейки с номерами от 0 до 255, во втором — от 256 до 511 и т. д. Если в комплексе есть *векторные процессоры,* то, как правило, Б. п. образуют *чередуемую память* с организацией, основанной на чередовании адресов. См. *чередуемая память*. Ср. *страница памяти*

БАННЕР, **банер [banner]**. 1. Заголовок статьи или документа, развернутый по всей ширине страницы. 2. Часть веб-страницы, содержащая рекламу и гиперссылку на веб-сайт рекламодателя (рис. Б.1)

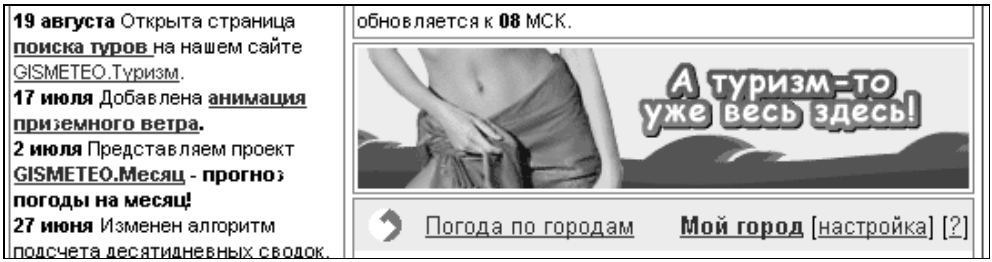

Рис. Б.1. Баннер на веб-странице сайта **www.gismeteo.ru**

БАННЕРНАЯ РЕКЛАМА [banner advertising]. Эффективный способ рекламы веб-сайта с помощью *баннеров*, расположенных на других сайтах. Служит для увеличения посещаемости рекламируемого сайта, а также является инструментом создания имиджа компании, продукции, услуг и т. д.

БД [database (DB)]. То же, что *база данных*

БЕГУНОК, **движок**, **ползунок [scroll box]**. Элемент управления, имеющий форму небольшого прямоугольника на *линейке прокрутки*, который используется для *скроллинга*. Двигая с помощью мыши Б. по линейке прокрутки, можно перемещаться по документу. Положение Б. на линейке прокрутки показывает относительное положение в документе отображаемой в окне информации (см. рис. Л.4). См. *элементы управления, графический интерфейс пользователя*

БЕЗУСЛОВНАЯ ПЕРЕДАЧА УПРАВЛЕНИЯ, **безусловный переход [unconditional branch]**. *Передача управления* по заранее определенному адресу. В *языках программирования* задается с помощью *оператора безусловного перехода*, который обычно имеет вид `go to m`, где `m` — неизменяемая (константная) метка

БЕЗУСЛОВНЫЙ ПЕРЕХОД [unconditional branch]. То же, что *безусловная передача управления*

БЕЙСИК [Basic, **beginner's all-purpose symbolic instruction code]**. *Язык программирования*, предназначенный для решения математических и экономических задач. Разработан в середине 1960-х гг. Напоминая упрощенную версию Фортрана, Б. обладает развитыми средствами диалога, а также *отладки* и *редактирования программ*. Простота синтаксиса, обеспечивающая сравнительно легкое и быстрое освоение, и относительная универсальность сделали Б. и его последующие модификации QBasic и Quick Basic широко распространенными языками программирования *персональных компьютеров*. Появление *системы программирования* Visual Basic, работающей в среде Windows, позволяет даже начинающим программистам разрабатывать программные изделия малой и средней степени сложности, ориентированные на работу в

этой операционной среде. В приложениях пакета Microsoft Office используется диалект языка Visual Basic for Applications, который разрешает пользователю автоматизировать применение продуктов пакета для создания *офисных приложений*

БЕНЧМАРКА [benchmark]. Эталонная программа, по которой судят о возможностях *вычислительной системы*. Как правило, Б. имеет простой алгоритм, регулярные структуры данных, значительную вычислительную емкость, возможность получения *производительности компьютера*, близкой к пиковой. В качестве Б. популярна программа LINPACK TPP, предназначенная для решения систем линейных алгебраических уравнений высокого порядка.

БЕОВУЛЬФ-КЛАСТЕР [Beowulf cluster]. *Вычислительный кластер*, составленный из *рабочих станций*. Название получил по одному из первых проектов *параллельных вычислительных систем*, созданному в центре NASA в 1994 г. Первый Б.-к. состоял из 16 вычислительных узлов с процессорами Intel 486DX4/100 МГц и *оперативной памятью* по 16 Мбайт, объединенных обычной сетью Ethernet.

БЕСКОНЕЧНЫЙ ЦИКЛ [infinite loop]. *Цикл*, который не может завершиться нормальным образом. В нем либо не предусмотрена проверка условия завершения и управление преднамеренно передается с конца *тела цикла* на начало, либо такая ситуация возникает в результате *логических* или *семантических ошибок*. Б. ц. может повторяться до тех пор, пока не будет прерван пользователем или *операционной системой*. См. *зацикливание, условие выхода из цикла*

БЕСПЛАТНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, **свободное программное обеспечение**. 1. **[public domain]**. Разнообразное бесплатно распространяемое программное обеспечение или документация, использование которых не оговаривается никакими условиями. 2. **[freeware]**. Бесплатно распространяемое программное обеспечение, использование которого оговаривается определенными условиями, например, запретом на коммерческое использование. Ср. *условно бесплатное программное обеспечение*. См. *Фонд свободного программного обеспечения*

БЕТА-ВЕРСИЯ [beta-release, **beta-version]**. Версия программы, предназначенная для *бета-тестирования*

БЕТА-ТЕСТИРОВАНИЕ [beta testing]. Пробная эксплуатация *программного продукта* перед его выпуском на рынок. Уже прошедшая испытания в фирме-изготовителе предварительная версия программы (т. н. *бета-версия*) вместе с предварительной версией документации официально рассылается наиболее авторитетным потенциальным пользователям, которые выступают в качестве экспертов. Они проверяют функциональные возможности про-

граммы, рассматривают, соответствует ли им документация. В процессе Б.-т. выявляются ошибки, связанные с непосредственным использованием продукта и не замеченные при разработке и испытаниях. По результатам Б.-т. фирма-разработчик корректирует программный продукт. После чего он тиражируется и поступает на рынок

БИБЛИОТЕКА ДИНАМИЧЕСКОГО СВЯЗЫВАНИЯ [dynamic link library (DLL)]. То же, что *библиотека динамической компоновки*

БИБЛИОТЕКА ДИНАМИЧЕСКОЙ КОМПОНОВКИ, **библиотека динамического связывания**, **динамически загружаемая библиотека [dynamic link library (DLL)]**. Один из способов организации *библиотек программ*, широко используемый в среде Windows. Б. д. к. обладает целым рядом преимуществ. 1) Связывание подпрограммы из Б. д. к. с вызывающей программой происходит динамически, т. е. во время выполнения программы в случае фактического *вызова подпрограммы* из Б. д. к. Это позволяет экономить ресурсы, поскольку не нужно загружать и связывать Б. д. к., пока не произошло фактического обращения к подпрограмме. 2) Подпрограммы из Б. д. к. обычно обладают свойством *реентерабельности*, поэтому один экземпляр Б. д. к., загруженный в память, позволяет обслуживать несколько вызывающих программ. 3) Подпрограммы в Б. д. к. удовлетворяют определенным соглашениям о вызове подпрограмм, поэтому программа, подготовленная в одной *системе программирования*, может вызывать подпрограммы из Б. д. к., подготовленной в другой системе программирования. Большинство современных систем программирования позволяют создавать Б. д. к. в виде файла с расширением dll

БИБЛИОТЕКА ПОДПРОГРАММ [subroutine library]. То же, что *библиотека программ*

БИБЛИОТЕКА ПОЛЬЗОВАТЕЛЯ [personal library, **private library]**. Составленное программистом и определенным образом организованное личное собрание программ, подпрограмм, процедур, макроопределений, текстов и наборов данных. Программист использует свою Б. п., либо запуская содержащиеся в ней программы, либо вставляя содержащиеся в ней элементы программ в разрабатываемую программу. Б. п. были распространены при работе с ЭВМ первого и второго поколений

БИБЛИОТЕКА ПРОГРАММ, **библиотека подпрограмм [program library**, **subroutine library]**. *Программный продукт*, содержащий собрание процедур, функций, макросов и т. п., предназначенных для использования при *разработке программ* в определенной *среде программирования*. Б. п. может содержать процедуры и функции, обеспечивающие выполнение определенных операций, например, работу с *графической информацией*, управление данными, математические расчеты и др. Достаточно с помощью специального оператора подключить Б. п. к разрабатываемой программе, чтобы содержа-

щиеся в ней процедуры или макросы можно было включать в состав программы пользователя. Б. п. представляет собой специальным образом организованный файл или набор файлов, содержащих, как правило, *объектные модули*, которые могут быть извлечены для присоединения к разрабатываемой программе на этапе компоновки. Б. п. выпускаются фирмами — разработчиками *программного обеспечения*. Ее дистрибутив обычно сопровождается печатным руководством, позволяющим освоить работу с библиотекой. Б. п. существенно облегчает программирование и поэтому является важной частью *программного обеспечения вычислительной системы*. Примером Б. п. может служить библиотека классов MFC *системы программирования* Microsoft Visual C++. Ее версия 4.0 содержит около 200 классов, каждый из которых включает от единиц до нескольких десятков *элементов данных* и методов работы с этими данными. Библиотека существенно упрощает разработку *пользовательского интерфейса* за счет поддержки разнообразных *элементов управления*, окон, разветвленной системы меню, редакторов данных, манипулятора "мышь", контекстных справочных систем и т. п. Кроме этого, библиотека позволяет управлять процедурами *обработки прерываний*, разрабатывать высококачественные *резидентные программы* при минимальных усилиях, поддерживать *виртуальную память* для хранения данных и многое другое

БИБЛИОТЕКАРЬ [librarian]. Управляющая программа, обеспечивающая организацию и обслуживание *системных библиотек* и *библиотек пользователя*. Б. позволяет выполнять такие операции, как включение элементов в библиотеку, удаление элементов, замена имен и др.

БИБЛИОТЕЧНАЯ ПРОГРАММА [library program]. *Программа*, которая может быть вызвана из *библиотеки программ*

БИЛЛИНГ [billing]. То же, что *тарификация*

БИНАРНАЯ ОПЕРАЦИЯ [binary operation]. 1. Операция над *двоичными числами*. 2. То же, что *двуместная операция*

БИС. См. *интегральная схема*

БИТ [bit]. 1. Один разряд *двоичного кода*. 2. Единица измерения количества информации, равная количеству информации, которое содержится в одном *двоичном разряде* или в ответе на вопрос, допускающий ответы "да" либо "нет" и никакого другого

БИТ ЧЕТНОСТИ [parity bit]. *Бит*, добавляемый к коду для контроля возможного искажения данных при хранении или пересылке. См. *контроль по четности*

БИТОВАЯ ГЛУБИНА ЦВЕТА, **глубина цвета**, **разрядность цвета [color depth**, **bit depth]**. Число двоичных разрядов (битов), предназначенных для кодиро-

вания цветов. Например, 1 бит обеспечивает два цвета, 8 битов — 256 цветов, а 24 бита — 16 777 216 цветов

БИТОВАЯ КАРТА [bitmap, pixel map]. То же, что *битовый массив*

БИТОВЫЙ МАССИВ, битовая карта, растр, растровый образ [bitmap, pixel map, pixel image]. *Структура данных*, которая описывает знак или графическое изображение в виде массива битов, задающих цвет каждой точки изображения. См. *растр, растровый файл*

БЛОК [block]. 1. То же, что *аппаратный блок*. 2. То же, что *блок данных*. 3. То же, что *блок программы*

БЛОК ДАННЫХ, блок [data block, block]. 1. Порция данных, передаваемая как единое целое при обмене данными между *внешними устройствами* или *центральным процессором* и внешними устройствами. 2. Объединение данных, хранящихся в соседних байтах или *ячейках памяти*, имеющее общий адрес и выступающее как одно целое в операциях обмена между *устройствами вычислительной системы*. На *жестких магнитных дисках* длина Б. д. не фиксирована и выбирается пользователем. На *гибких магнитных дисках* — фиксирована и составляет, как правило, 512 байт. 3. Единица измерения количества данных при обмене данными с диском или *магнитной лентой*

БЛОК ОРФОГРАФИЧЕСКОГО КОНТРОЛЯ, корректор, проверка орфографии [spelling checker]. Средство проверки орфографии, применяемое в *текстовых процессорах*, системах *электронной почты* и других приложениях. В простейших случаях Б. о. к. выделяет в проверяемом тексте слова, ищет их во встроенном словаре и сообщает пользователю о возможной ошибке, если не находит слова в словаре. В современных приложениях Б. о. к. может использовать сразу несколько словарей, относящихся к разным предметным областям и к различным языкам, может учитывать наличие словоформ и другие особенности языка, допускает пополнение словарей пользователем, исключает из проверки аббревиатуры, предлагает пользователю заменить неправильное слово сходным по написанию правильным словом и т. д. Например, на рис. Б.2 показан фрагмент документа MS Word, в котором Б. о. к. предлагает с помощью *контекстного меню* несколько вариантов замены неправильно написанного слова

БЛОК ПАМЯТИ [memory block, storage block]. Непрерывная *область памяти*, которой может быть присвоен ключ для ее идентификации или защиты. Выделяемая программе память состоит из одного или нескольких Б. п. Размер Б. п. может быть разным

БЛОК ПРОГРАММЫ, программный блок, блок [program block, block]. Относительно независимый фрагмент программы, ограниченный особыми символами *(операторными скобками)*. В *языках программирования высокого уровня*

Б. п. может содержать описания меток, констант, типов, переменных, процедур и функций, а также операторы и *вложенные блоки*. Имена, объявленные в Б. п., локализованы в нем самом и во всех вложенных в него блоках. Предполагается, что *оперативная память* для объектов, описанных в Б. п., выделяется динамически на период его выполнения и освобождается по завершении блока. Простейшим Б. п., в котором отсутствуют описания, является *составной оператор*

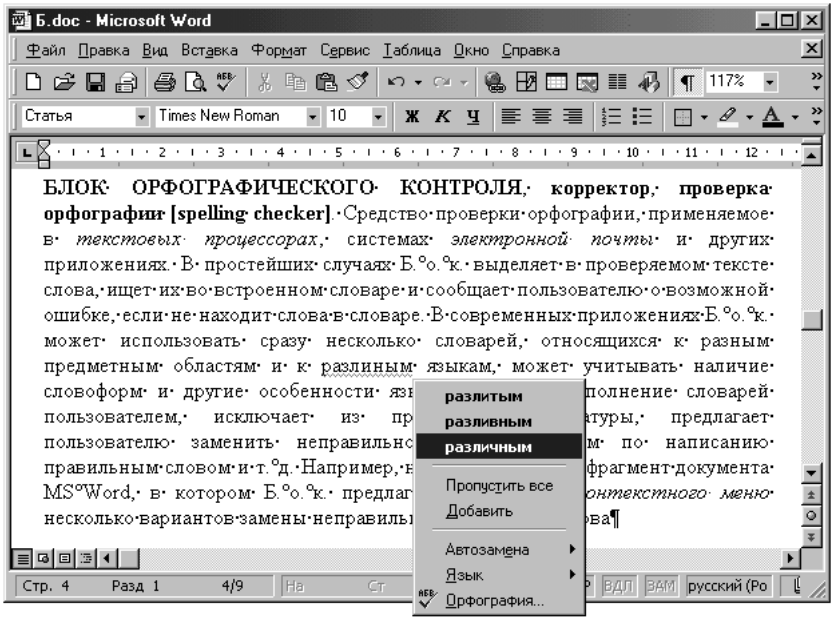

Рис. Б.2. Контекстное меню Microsoft Word для исправления орфографической ошибки

БЛОК УПРАВЛЕНИЯ [control block]. 1. Электронный блок, управляющий устройством. 2. Таблица, содержащая информацию, необходимую *операционной системе* для управления *вычислительным процессом*. Например, Б. у. *устройством ввода/вывода* — таблица, используемая драйвером ввода/вывода и содержащая данные о рабочих характеристиках устройства

БЛОКИРОВАНИЕ [blocking]. 1. Объединение элементов в один *блок*. См. *аппаратный блок, блок данных, блок программы*. 2. То же, что *блокировка*

БЛОКИРОВКА [locking, lockout]. Приведение программы или устройства в состояние, препятствующее выполнению определенных действий. Например, Б. клавиатуры препятствует вводу информации в компьютер. См. *блокировка данных*

БЛОКИРОВКА БЮДЖЕТА [account lockout]. Функция *системы безопасности*, блокирующая *пользовательский бюджет*, если в течение указанного времени происходит заданное количество неудачных попыток регистрации в системе. Б. б. базируется на установках блокировки, задаваемых *политикой безопасности*. Блокированный бюджет не позволяет пользователю зарегистрироваться в системе

БЛОКИРОВКА ДАННЫХ [data interlock]. Защита записи, *блока данных* или файла путем запрещения доступа к ним от всех пользователей, за исключением одного. Б. д. является механизмом организации контролируемого доступа к совместно используемым данным, предотвращающего одновременное обращение к ним двух и более программ

БЛОКИРОВКА ЗАПИСИ [record locking]. То же, что *захват записи*

БЛОКИРОВКА УЧЕТНОЙ ЗАПИСИ [account lockout]. Одно из средств обеспечения безопасности *операционной системы*, которое состоит в том, что в некоторых случаях не допускается регистрация пользователя в системе. Например, *операционная система Windows NT* осуществляет Б. у. з. пользователя в том случае, если в течение определенного интервала времени с помощью этой учетной записи было совершено определенное количество неудачных попыток *входа в систему*. Продолжительность интервала времени и предельное количество неудачных попыток входа в систему задаются соответствующими параметрами *политики учетных записей*. Снять Б. у. з. может *системный администратор*

БЛОКНОТНЫЙ КОМПЬЮТЕР, **карманный компьютер, наладонный компьютер [hand-held PC (HPC)]**. Малогабаритный *портативный компьютер*, который можно держать в руке во время работы (рис. Б.3). Б. к. применяются в тех случаях, когда использование иных видов *персональных компьютеров* неудобно по условиям работы. Например, при сборе данных в полевых условиях. Несмотря на малые размеры, Б. к. обладает достаточными возможностями для выполнения разнообразных функций сбора и первичной обработки информации, а также возможностями связи с другими компьютерами в стационарных условиях.

Рис. Б.3. Блокнотный компьютер

Многие Б. к. работают под управлением *операционной системы* Windows CE, ориентированной на управление компьютерами с ограниченными ресурсами. Некоторые Б. к. содержат встроенное программное обеспечение для распознавания рукописного текста

БЛОК-СХЕМА ПРОГРАММЫ, **схема программы [program flowchart]**. Графическое изображение алгоритма, подлежащего программированию. Б.-с. п. показывает последовательность основных шагов выполнения программы и применяется на начальном этапе ее разработки. Средством построения Б.-с. п. являются типовые символы, имеющие вид геометрических фигур (см. *элементы блок-схем*). Каждый символ обозначает определенные действия компьютера. Последовательность действий указывается линиями. Например, на рис. Б.4 приведена Б.-с. п. вычисления вещественных корней квадратного уравнения

БЛОЧНАЯ СТРУКТУРА ПРОГРАММЫ [program block structure]. Организация программы в виде последовательности смежных или *вложенных блоков программы*. Б. с. п. позволяет экономить *оперативную память* и реализована во многих *языках программирования*, например, в Паскале

БОД [baud]. Единица измерения скорости передачи информации, определяемая числом символов, передаваемых в секунду. Для каналов, передающих информацию в *двоичном коде*, 1 бод = 1 бит/с

БРАНДМАУЭР [firewall]. *Вычислительная система* или комбинация систем, создающая защитный барьер между двумя или бóльшим количеством сетей и предотвращающая нежелательные вторжения в частную сеть (интранет). Б. служат виртуальными барьерами для передачи пакетов из одной сети в другую и отслеживают движение данных между сетями Интернет и интранет

БРАУЗЕР [browser]. Программа чтения гипертекста. Браузеры, применяемые в Интернете (веб-браузеры), помимо чтения гипертекста, могут обладать следующими дополнительными возможностями: 1) воспроизводить звук и видеоданные (движущиеся изображения), т. е. воспроизводить гипермедиа; 2) устанавливать соединения с другими компьютерами сети и посылать работающим на них серверам запросы на веб-документы; 3) организовывать и поддерживать диалог с другими пользователями сети. Наиболее распространенные веб-браузеры для операционных систем Windows — Microsoft Internet Explorer и Netscape Navigator. Они поддерживают все возможности гипермедиа и устанавливают связь, а Netscape Communicator также позволяет редактировать документы и запускать доступные программы. В различных браузерах один и тот же документ может выглядеть по-разному. Это зависит от характеристик конкретного браузера и от установленных пользователем настроек. См. *гипермедиа*, *универсальный указатель ресурса*

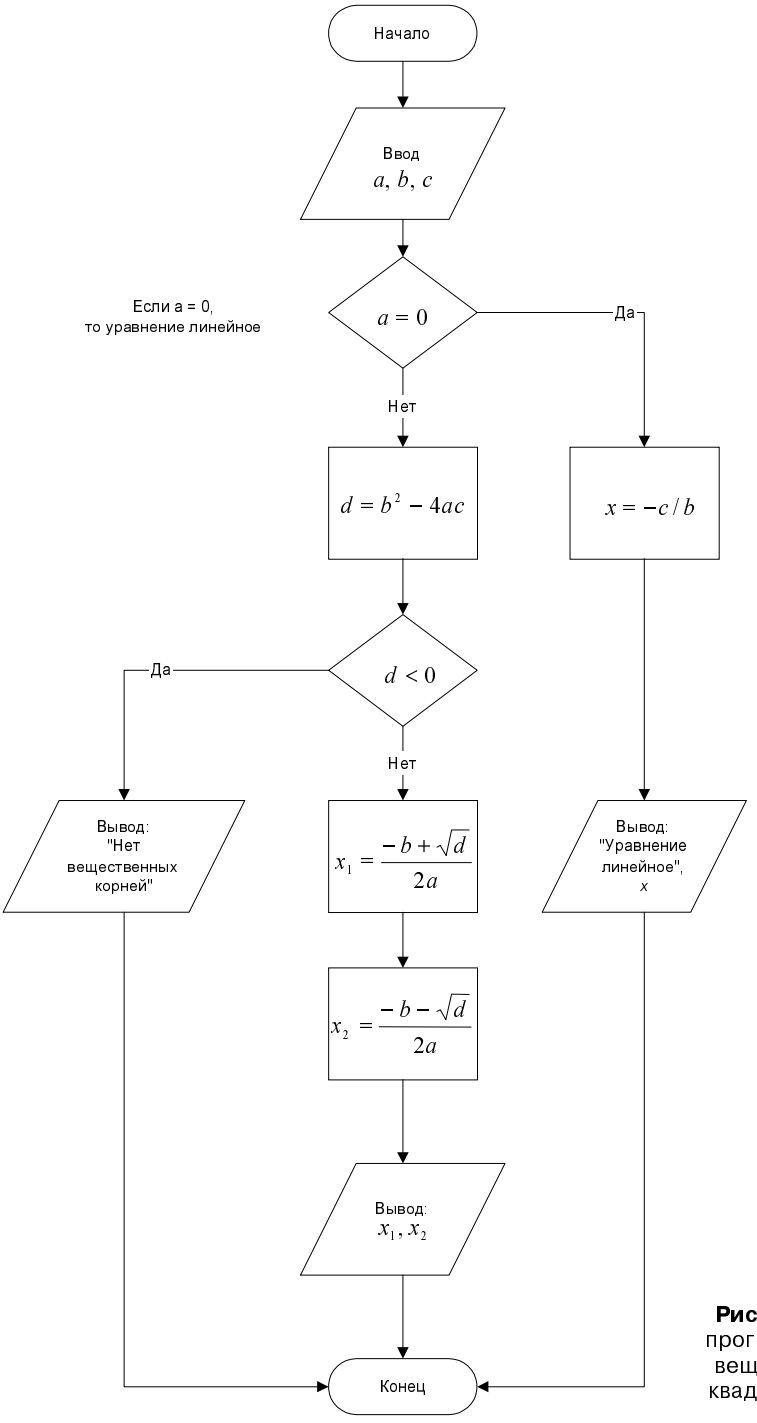

Рис. Б.4. Блок-схема программы вычисления вещественных корней квадратного уравнения

БРАУЗЕР WINDOWS NT [browser]. Один из специальных компьютеров в сети, которым ОС Windows NT делегирует выполнение просмотра (browsing) — формирования списка всех сетевых ресурсов, доступных в данный момент (списка просмотра, browsing list). 1) Главный браузер домена (domain master browser). Компьютер, отвечающий за поддержку списка главных браузеров всех подсетей. Он объединяет списки главных браузеров подсетей в общий список браузеров домена. Затем этот список направляется обратно, т. е. главному браузеру каждой подсети. По умолчанию главным браузером домена является первичный (главный) контроллер домена (Primary Domain Controller, PDC), который содержит главную БД домена. 2) Главный браузер (master browser). Компьютер, имеющийся в каждой подсети или рабочей группе и отвечающий за поддержку списка просмотра. При запуске любой компьютер подсети регистрируется главным браузером. Главный браузер ведет список всех зарегистрированных компьютеров, который отправляется главному браузеру домена. 3) Резервный браузер (backup browser). Помогает главному браузеру, по запросу отправляя клиентам список просмотра. Резервный браузер получает этот список от главного браузера. В случае отключения главного браузера его место занимает резервный. 4) Потенциальный браузер (potential browser). Компьютер, не выполняющий в данный момент функции браузера, но способный делать это в случае необходимости. 5) "Небраузер" (non browser). Компьютер, сконфигурированный таким образом, что он не принимает участия в поддержке списка просмотра сетевых ресурсов

БРОУЗЕР [browser]. То же, что *браузер*

БРУТЕР [bridge-router, **b-router**, **brouter]**. То же, что *мост-маршрутизатор*

БУКВА ДИСКА [disk letter]. Буква латинского алфавита, которой обозначается *диск* на *персональных компьютерах*. Диски обозначаются буквой, начиная с А, за которой следует двоеточие. См. *логический диск*

БУКВИЦА [dropped capital letter, **drop cap]**. Первая буква абзаца, специально отформатированная с увеличением размера и часто с использованием декоративного шрифта. На рис. Б.5 приведен пример Б., отформатированной в приложении Microsoft Word

БУКСИРОВКА МЫШЬЮ [drag and drop]. То же, что *перетаскивание мышью*

БУЛЕВА АЛГЕБРА [Boolean algebra]. Алгебра, в которой каждая переменная может принимать одно из двух значений: ИСТИНА или ЛОЖЬ. Операции над переменными в Б. а. называются *булевыми* или *логическими операциями*. Правила выполнения логических операций удобны для преобразования логических схем. Поэтому Б. а. явилась основой для разработки ЭВМ. См. *логические операции*

БУЛЕВА ОПЕРАЦИЯ [Boolean operation]. То же, что *логическая операция*

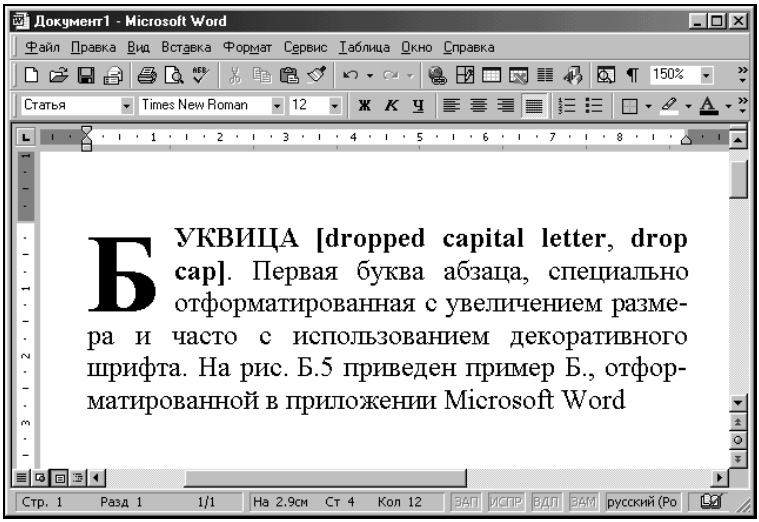

Рис. Б.5. Буквица в документе Microsoft Word

БУЛЕВА ФУНКЦИЯ [Boolean function]. То же, что *логическая функция*

БУЛЕВО ВЫРАЖЕНИЕ [Boolean expression]. То же, что *логическое выражение*

БУЛЕВО ЗНАЧЕНИЕ [Boolean value]. То же, что *логическое значение*

БУЛЛИТ, маркер [bullet]. Символ, выделяющий слева пункт в списке, абзац или фрагмент текста. В качестве Б. могут быть использованы различные значки и даже небольшие рисунки. На рис. Б.6 показано окно предложения MS Word, предлагающее на выбор различные типы Б. См. *маркированный список*

БУФЕР [buffer]. 1. То же, что *буферная память*. 2. *Электронная схема* или устройство, помещаемые между двумя другими устройствами для согласования различий в их рабочих характеристиках

БУФЕР ОБМЕНА [clipboard]. Средство *операционной системы Windows*, предназначенное для обмена данными между приложениями. Б. о. позволяет вырезать или скопировать выделенные данные в специальное место, отведенное для временного хранения данных, а затем вставить эти данные из буфера в другое место. В Б. о. вместе с данными хранится информация об их типе и формате, поэтому вставка данных из буфера может производиться различными способами, в зависимости от типа данных и возможностей приложения, в документ которого вставляются данные. См. *вставка, выделение, вырезание, копирование*

БУФЕРИЗАЦИЯ [buffering]. Применение *буферов*. См. *буферная память, спулинг*

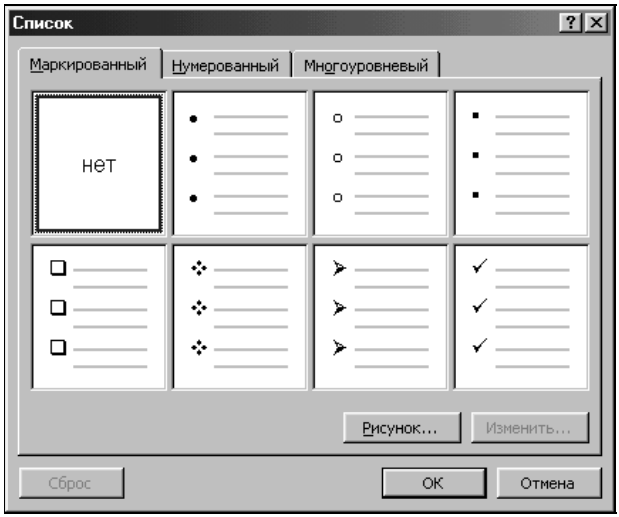

Рис. Б.6. Варианты буллитов приложения Microsoft Word

БУФЕРНАЯ ПАМЯТЬ, **буфер [buffer memory, buffer]**. *Память* для промежуточного хранения данных. Применяется при *обмене данными* между двумя устройствами, обладающими различной скоростью обработки информации, чтобы работа "тихоходного" устройства не снижала их общую производительность. Характерным примером использования Б. п. является выдача данных программой на принтер. Так как скорость передачи данных от программы к принтеру неизмеримо выше, чем скорость работы принтера, то, чтобы не замедлять работу программы на время выдачи данных, программа помещает данные в Б. п. и продолжает свою работу, а принтер "самостоятельно" извлекает их со своей скоростью и печатает, не требуя прерывания работы программы. Многие *внешние устройства* имеют собственную Б. п. кроме того, в качестве Б. п. может служить специально выделяемая *операционной системой* область *оперативной памяти*. См. *буфер обмена, кэш-память, спулинг*

БУФЕРНЫЙ ПРОЦЕССОР, **препроцессор [front-end processor]**. 1. Процессор, генерирующий или предварительно обрабатывающий данные для другого процессора. 2. *Сопроцессор*, выполняющий форматирование и предварительную обработку информации, вводимой в компьютер. Ср. *постпроцессор*

БУФЕРНЫЙ ФАЙЛ [spool file]. В *многозадачных операционных системах* — *файл*, предназначенный для временного запоминания *входных* или *выходных данных* задачи. См. *спулинг*

БЫСТРАЯ РАЗРАБОТКА ПРИЛОЖЕНИЙ [rapid application development (RAD)]. *Процесс разработки программного обеспечения*, ориентированный на максимальное ускорение. Ради сокращения времени разработки, как правило, жертвуют документацией, проектированием, расширяемостью и масштабируемостью *программного обеспечения*

БЫСТРОДЕЙСТВИЕ [performance, speed]. Характеристика скорости работы устройства. Например, Б. процессора характеризуется числом элементарных операций, выполняемых в секунду, или *тактовой частотой*, Б. принтера — числом знаков, печатаемых в секунду. См. *быстродействие компьютера*

БЫСТРОДЕЙСТВИЕ КОМПЬЮТЕРА [computer speed, computer performance]. То же, что *производительность компьютера*

БЫСТРОДЕЙСТВИЕ ЭВМ [computer speed, computer performance]. То же, что *производительность компьютера*

БЫСТРЫЕ КЛАВИШИ [quick keys]. То же, что *клавиши быстрого вызова*

БЮДЖЕТ КОМПЬЮТЕРА [computer account]. То же, что *учетная запись компьютера*

БЮДЖЕТ ПОЛЬЗОВАТЕЛЯ [account, user account]. То же, что *учетная запись пользователя*

В

ВАРИАНТ ИСПОЛЬЗОВАНИЯ [use case]. Спецификация последовательностей действий, которые могут осуществлять система, подсистема или класс, взаимодействуя с внешними *действующими лицами*

ВВОД ДАННЫХ [data input]. 1. Процесс записи данных в *память компьютера* с помощью *устройства ввода*. Некоторые устройства ввода требуют представления данных на специальных носителях, таких как *магнитные диски*, *магнитные ленты* и т. п., на которые данные должны быть предварительно записаны либо с помощью компьютера, либо на специальном устройстве подготовки данных. В. д. может осуществляться непосредственно человеком — голосом или с помощью клавиатуры. В *интерактивных вычислительных системах* В. д. происходит в *диалоговом режиме*. 2. Процесс *передачи данных* из *внешнего запоминающего устройства* в *оперативную память*

ВВОД/ВЫВОД [input-output]. Общий термин, определяющий процесс *обмена данными* между *оперативной памятью* и *внешними устройствами*

ВЕБ-ДОКУМЕНТ, документ WEB, документ WWW [Web document, WWW document]. *Документ* на *языке HTML*. В.-д. составляют информационную

основу *Всемирной паутины*. Они содержат *гипертекст*, позволяющий пользователю, указав на выделенное слово или фразу, получить доступ к данным, перейти в другую часть данного документа или в иной В.-д., связанный с данной *гиперссылкой*. В.-д. может также содержать данные *гипермедиа*, включающие текст, изображения, звук. Доступ к В.-д., их чтение или воспроизведение производятся с помощью *браузеров Интернета*. С понятием В.-д. тесно связаны понятия *веб-страницы* и *веб-сайта*. Значения этих терминов в настоящее время жестко не определены. Чаще всего термин "веб-страница " является синонимом термина В.-д., а термин "веб-сайт " относится к набору страниц, объединенных тематически либо принадлежащих одной организации, одному автору или пользователю. См. *динамическая веб-страница*

ВЕБ-ПАПКА, WEB-папка [Web folder]. *Папка* на *веб-сервере*. Файлы и папки на веб-серверах можно просматривать так же, как файлы и папки на серверах *локальной компьютерной сети*. При сохранении файла в В.-п. он помещается на веб-сервер, а не на *жесткий диск* компьютера

ВЕБ-ПОРТАЛ [portal, Web portal]. То же, что *портал*

ВЕБ-САЙТ [Web site]. То же, что *сайт*

ВЕБ-СЕРВЕР, WEB-сервер, интернет-сервер [Web server]. Компьютер, снабженный *программным обеспечением сервера* и предназначенный для получения, обработки и исполнения запросов от клиентов Интернета. В.-с. использует для связи с клиентами *протокол HTTP*. В.-с. обслуживается *системным администратором* или *интернет-провайдером*

ВЕБ-СТРАНИЦА, страница WEB [Web page]. См. *веб-документ*

ВЕКТОР [vector]. *Одномерный массив*. Упорядоченное объединение конечного числа однотипных *элементов данных*. Элементы В. располагаются в памяти последовательно один за другим. Начало В. определяется адресом его первого элемента, называемым базой вектора. В. широко используются в вычислительной технике, поскольку память по существу представляет собой В. байтов или слов. См. *банк памяти*

ВЕКТОРНАЯ КОМАНДА [vector instruction, array instruction]. *Команда*, среди операндов которой есть массивы *(векторы)*. См. *векторный процессор*

ВЕКТОРНЫЙ ПРОЦЕССОР [vector processor]. *Процессор*, в набор команд которого входят *векторные команды*. Их операндами являются массивы *(векторы)*. В. п. может обработать одной командой сразу множество значений, например, произвести поэлементное сложение двух массивов и присвоить полученные значения элементам третьего массива. Обычному процессору для этого пришлось бы многократно исполнить операцию сложения элементов двух массивов. Каждый операнд векторной команды хранится в особом векторном регистре большой емкости. Например, векторный регистр

компьютера CRAY Y-MP C90 позволяет хранить операнд длиной 128 64-разрядных слов.

ВЕКТОРНЫЙ РЕГИСТР [vector register, V-register]. См. *векторный процессор*

ВЕКТОР ПРЕРЫВАНИЙ [interrupt vector]. *Вектор* системных данных, используемый *операционной системой* для *обработки прерываний.* При работе со множеством разнотипных устройств, каждое из которых способно формировать сигналы прерывания, необходимо обрабатывать эти прерывания, причем для каждого типа устройства требуется своя уникальная программа *обработки прерываний.* В. п. обычно содержит адреса таких программ и, возможно, *слова состояния процессора*, используемые для организации соответствующего прерывания. См. *прерывание, обработка прерываний*

ВЕКТОРНАЯ ГРАФИКА [vector graphics]. Метод создания изображений в виде совокупности линий. Каждая линия рисунка представляется отрезками прямых (векторами) и сопрягающимися с ними отрезками стандартных геометрических кривых. Для определения формы и расположения отрезка используются математические описания. В. г. применяется в *программах рисования*, которые работают с рисунком как с совокупностью форм, а не как с группой точек. Рисунки, выполненные методами В. г., более компактны при хранении, допускают точное масштабирование и не зависят от *разрешающей способности* графических устройств. Ср. *растровая графика*

ВЕКТОРНЫЙ ШРИФТ [vector font]. *Шрифт*, в котором для получения изображения символа используются математические формулы, а не зафиксированные растровые рисунки. Символы воспроизводятся в порядке следования сегментов линий и могут масштабироваться в соответствии с нужным размером. Если устройство отображения не поддерживает воспроизведение В. ш., то предварительно производится растрирование. Ср. *растровый шрифт.* См. *шрифты TrueType*

ВЕРИТЕЛЬНЫЕ ДАННЫЕ [credentials]. Имя и пароль пользователя, *глобально-уникальный идентификатор*, или сертификат клиента, на основании которых проверяется подлинность пользователя. См. *аутентификация*

ВЕРИФИКАЦИЯ ПРОГРАММЫ [program verification]. Доказательство правильности программы, т. е. соответствия программы ее спецификациям. В. п. может осуществляться любым методом, который убеждает в том, что программа будет выполнять именно то, что от нее ожидается. Часто вывод о правильности программы делают на основе анализа результатов ее *прогонов (тестирования программы).* Такой подход, как правило, не охватывает всех возможных вариантов эксплуатации программы и не может служить доказательством ее правильности. Важнейшим является метод математического доказательства общей правильности программы как объекта, на который распространяются аксиомы и правила формальной логики. Для такого доказательства требуется тщательный учет структуры программы и семантики

языка программирования. Трудности, возникающие на этом пути, частично устраняются компьютерными системами автоматической В. п., которые способны, анализируя текст программы, формулировать условия верификации в форме теорем и либо доказывать, что эти условия выполнены, либо предоставлять их пользователю для неавтоматического доказательства. Ср. *испытание программы*

ВЕРНЫЕ ЗНАЧАЩИЕ ЦИФРЫ [right significant digits]. *N* первых *значащих цифр* приближенного числа называются В. з. ц., если абсолютная погрешность этого числа не превышает половины единицы разряда, выраженного *N*-й значащей цифрой, считая слева направо. Например, для точного числа 78.96 число 79.00 является приближением с тремя В. з. ц., т. к. $|79.00 - 78.96| = 0.04 \le 0.1/2$, где 0.1 — единица разряда третьей значащей цифры данного числа. Приведенный пример показывает, что в приближенном числе, имеющем *N* В. з. ц., не все *N* первых цифр должны совпадать с соответствующими цифрами точного числа. Однако во многих случаях В. з. ц. приближенного числа и соответствующие цифры точного числа одинаковы. При приближенном задании числа следует указывать только В. з. ц.

ВЕРСИЯ [release, version]. 1. Вариант *программного продукта* или *языка программирования.* Например, Word 7.0 для Windows 95 — седьмая версия текстового редактора Microsoft Word для Windows 95. Обычно используют номера версий, состоящие из двух частей. Первая часть, называемая старшим номером версии, изменяется в том случае, когда в возможности программы вносятся принципиальные изменения. Вторая часть, называемая младшим номером версии, указывает на наличие непринципиальных изменений, скажем, исправление замеченных ошибок, уточнения и т. д. Например, UML 1.3 — версия языка UML, в которой сделаны небольшие добавления и изменена форма описания, но сам язык по существу не изменился. Другой распространенный способ именования версий основан на указании года выпуска. Например, Algol-60 — версия языка Algol, принятая в 1960 г. 2. Файл, являющийся модификацией другого файла

ВЕРСТКА СТРАНИЦ [page makeup]. Полиграфический термин, означающий размещение текстового и графического материала на печатной странице в соответствии с определенными правилами. Правила зависят от типа издания, от национальных традиций и др. Например, в отечественной научной литературе принято выравнивать текст по ширине, а в англоязычной — по левому краю. В настоящее время В. с. выполняется, как правило, с помощью компьютера, поскольку специализированные *настольные редакционно-издательские системы* и *текстовые процессоры* имеют все необходимые средства и позволяют добиться более высокого качества В. с. При В. с. используются такие средства, как масштабирование иллюстраций, изменение *межстрочных интервалов,* отступы, выравнивание и др.

ВЕРТИКАЛЬНОЕ МЕНЮ [vertical menu]. *Меню* с вертикальным расположением *пунктов меню* (один под другим). На экране дисплея такое меню перекрывает выведенную ранее информацию (см. рис. М.5). Противоп. *горизонтальное меню*

ВЕРХНИЙ КОЛОНТИТУЛ [header, running head]. *Колонтитул*, расположенный в верхней части страницы, выше основного текста

ВЕРХНИЙ РЕГИСТР [uppercase]. Клавиатура компьютера (так же, как и любой пишущей машинки) имеет два регистра — верхний и нижний. В. р. служит для ввода прописных букв и других символов, указанных в верхней части клавиш. Нижний регистр предназначен для ввода строчных букв и других символов, указанных в нижней части клавиш. В исходном состоянии ввод производится в режиме нижнего регистра. Переход с нижнего регистра на верхний осуществляется нажатием клавиши <Shift>. При этом режим ввода в В. р. действует, только пока клавиша <Shift> удерживается нажатой. Клавиша <Caps Lock> служит для фиксации или отмены режима В. р. Действие нажатой клавиши <Caps Lock> распространяется только на буквенные клавиши. Противоп. *нижний регистр*

ВЕРХНЯЯ ПАМЯТЬ [upper memory blocks, UMB]. Область старших адресов *оперативной памяти персональных компьютеров* при работе под управлением *операционной системы* MS-DOS. В других ситуациях в настоящее время этот термин не используется. См. *оперативная память персональных IBM-совместимых компьютеров*

ВЕТВЛЕНИЕ ПРОГРАММЫ [branching]. Предусмотренная в *программе* возможность выполнения тех или иных действий в зависимости от результатов проверки некоторого условия. На *блок-схеме программы* В. п. изображается при помощи символа "Решение" (см. *элементы блок-схем*). Например, на рис. В.1 приведена блок-схема программы, определяющей координаты точки пересечения двух прямых на плоскости XOY, заданных уравнениями $y = ax + b$ и $y = cx + d$. Если прямые окажутся параллельными (при условии, что $a = c$), то программа сообщит об этом. В *языках программирования высокого уровня* В. п. программируется *оператором условного перехода*, который в *машинном языке* реализуется командами *условной передачи управления*

ВЕТВЬ АЛГОРИТМА [algorithm branch]. Одна из альтернативных последовательностей действий *алгоритма*, выбор которой зависит от некоторых условий. На блок-схеме — один из альтернативных путей, исходящих из точки ветвления. В программе В. а. соответствует *ветвь программы*. См. *ветвление программы*

ВЕТВЬ ПРОГРАММЫ [program branch]. 1. Участок *программы*, которому может быть передано управление при *ветвлении программы*. 2. Участок программы, который может выполняться параллельно с другими участками. На

блок-схеме В. п. изображается одним из альтернативных путей, исходящих из точки ветвления

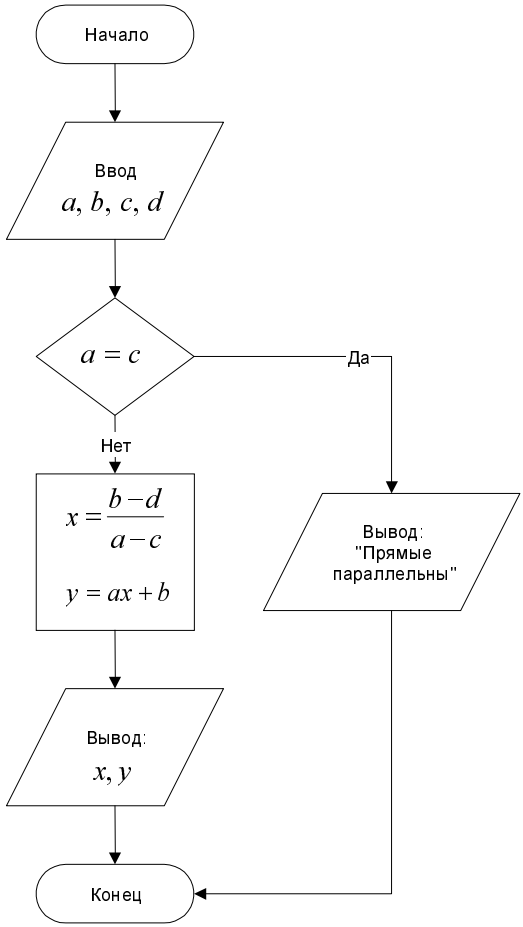

Рис. В.1. Блок-схема программы определения точки пересечения прямых

ВЕЩЕСТВЕННАЯ КОНСТАНТА, **действительная константа [real constant]**. *Константа*, представленная в программе в виде *числа вещественного типа*. См. *константа*

ВЕЩЕСТВЕННАЯ ПЕРЕМЕННАЯ, **действительная переменная [real variable]**. *Переменная*, значением которой могут быть только *числа вещественного типа*. В. п. вводятся в программу с помощью *описания переменной*, в котором указываются идентификатор (имя) переменной и *ключевое слово*, опреде-

ляющее *вещественный тип*, значения которого она может принимать. В качестве ключевых слов в описаниях В. п. применяются: `real` — в Фортране и Паскале и `float` либо `double` — в Си. См. *описание переменной*

ВЕЩЕСТВЕННОЕ ЧИСЛО, **действительное число [real number]**. *Число*, которое может быть представлено конечным или бесконечным количеством цифр в *системе счисления* с фиксированным основанием. В. ч. может быть целым, дробным, положительным, отрицательным или нулем. В программировании используются числа с конечным количеством цифр, поэтому термин "В. ч." применяется к *числам вещественного типа*, чтобы отличить их от *чисел целого типа*

ВЕЩЕСТВЕННЫЙ ТИП, **действительный тип**, **тип Real [real type]**. *Тип данных*, представляемых в виде *чисел с фиксированной* или *плавающей точкой*. В. т. составляют множество *вещественных чисел*, определяемое *диапазоном изменения чисел* данной *вычислительной системы*, и множество выполняемых над ними *арифметических операций* и *операций сравнения*. В качестве *ключевых слов* при *описании переменных* и других *объектов программы*, относящихся к В. т., применяются: `real` — в Фортране и Паскале и `float` либо `double` — в Си. Так как количество разрядов, отведенных для хранения каждого числа в памяти и арифметическом устройстве компьютера, ограничено, то операции с данными В. т. в большинстве случаев выполняются приближенно. Их точность зависит в основном от способа представления вещественных чисел в компьютере (принятой *системы счисления*, количества разрядов, отведенных для мантиссы и порядка), т. е. от того, с каким компьютером вы работаете. Поэтому в большинстве *языков программирования* имеется несколько встроенных В. т., которые различаются диапазоном изменения и точностью. Например, в языке Turbo Pascal предусмотрены три В. т.: `real`, `double`, `extended`. Ср. *целый тип*. См. *число вещественного типа, представление чисел с плавающей точкой, диапазон изменения чисел*

ВЗАИМНАЯ БЛОКИРОВКА [deadlock]. То же, что *тупиковая ситуация*

ВЗАИМОДЕЙСТВИЕ [interaction]. В *унифицированном языке моделирования* UML определение того, как объекты обмениваются сообщениями для достижения определенной цели. Взаимодействие описывается на *диаграммах кооперации* и *диаграммах последовательности*

ВЗУ [backing storage, **external storage]**. То же, что *внешнее запоминающее устройство*

ВИДЕОАДАПТЕР [video adapter]. То же, что *адаптер дисплея*

ВИДЕОДАННЫЕ, **видеоинформация [video data**, **video information]**. Информация о зрительных образах, представленная в цифровой форме, предназначенной для хранения, обработки и воспроизведения средствами вычислительной техники. Примерами В. являются введенные в память компьютера

рисунки, видеоизображения и т. п. В. можно передавать, хранить и обрабатывать в *вычислительной машине* с последующим воспроизведением на экране дисплея, принтере или другом отображающем В. устройстве. Для преобразования цифровых В. в аналоговый видеосигнал, управляющий изображением на экране дисплея, служит *адаптер дисплея*. Для длительного хранения В. применяются *магнитные* и *лазерные диски*. См. *видеодиск, мультимедиа*

ВИДЕОДИСК [videodisk]. *Лазерный диск* для *видеоданных*. На В. хранятся цифровые коды изображений, предназначенных для воспроизведения на экране дисплея, принтере или другом воспроизводящем видеозапись устройстве. Изображения могут быть статическими и динамическими, плоскими и объемными, монохромными и цветными. Они также могут иметь звуковое сопровождение. Четкость и красочность изображений, получаемых с В. на экране дисплея, превосходят качество изображений, получаемых типографскими методами. Для В. разработаны специальные форматы записи, например, формат VideoCD, используемый для хранения движущихся изображений

ВИДЕОИНФОРМАЦИЯ [video information]. 1. Зрительные образы, изображения. 2. То же, что *видеоданные*

ВИДЕОКАРТА [videocard]. То же, что *адаптер дисплея*. См. *мультимедиа*

ВИДЕОКОДЕК [videocodec]. *Кодек* для преобразования и передачи *видеоданных* в реальном времени

ВИДЕОПАМЯТЬ [video memory, **video RAM**, **video storage]**. *Память*, предназначенная для записи, хранения и считывания данных, определяющих изображение на экране дисплея. Эти данные отображаются на экране видеоадаптером. Как часть оперативной памяти В. является доступной *центральному процессору*, и программы могут непосредственно вводить в нее данные, изменяя изображение на экране. Организация В. зависит от режима работы видеоадаптера. Существуют два класса режимов: текстовые и графические. При текстовом режиме на экране могут отображаться только *текстовые* и *псевдографические символы*. Поэтому экран делится на определенное число символьных позиций (знакомест), каждому из которых в В. отводится определенное количество (обычно два) байт. Как правило, первый из них содержит код отображаемого символа, а последующие — атрибуты символов, определяющие шрифт, цвет и интенсивность символа, цвет и яркость фона, мигание и т. п. Благодаря такому режиму хранения информации достигается значительная экономия памяти. Распределение В. в графических режимах работы видеоадаптера иное. Здесь экран представляет собой *растр* — множество точек (пикселов), упорядоченных в строки и столбцы, а изображение формируется путем придания цвета и яркости каждой точке растра независимо. Поэтому каждому пикселу экрана соответствуют несколько бит В., которые содержат данные о его цвете и яркости. И хотя у видеоадаптеров

может быть несколько текстовых и несколько графических режимов, отличающихся деталями их реализации, существует прямая зависимость между качеством отображения и требуемым для этого *информационным объемом* В. Например, для вывода на экран 1024 точек по горизонтали, 768 — по вертикали и 65 536 возможных цветов каждой точки необходимы 2 Мбайт В. Для работы с бóльшим *разрешением* и большей *глубиной цвета* нужно, соответственно, иметь больший объем В. В большинстве режимов видеоадаптеров В. разделена на несколько страниц. При этом одна из страниц является активной и отображается на экране. В программе можно выводить информацию как в активную, так и в неактивные страницы и переключать активные страницы. Таким образом можно заранее подготовить изображения в нескольких страницах В. (несколько экранов), а затем менять их на экране дисплея

ВИДЕОПЛАТА [videocard]. То же, что *адаптер дисплея*

ВИДЕОСИСТЕМА КОМПЬЮТЕРА [video system]. Совокупность *аппаратных* и *программных средств* компьютера, предназначенных для создания, ввода, обработки и воспроизведения *видеоинформации*. Основными неотъемлемыми частями В. к. являются видеоадаптер, видеопамять и монитор. От их возможностей напрямую зависят возможности всей В. к. Качество В. к. определяется скоростью обработки видеоданных, разрешающей способностью и цветовой палитрой. См. *видеоданные, видеоадаптер, видеопамять, графический сопроцессор*

ВИДЕОТЕРМИНАЛ [video terminal]. *Терминал*, состоящий из дисплея и клавиатуры

ВИДИМОСТЬ [scope]. То же, что *область видимости*

ВИЗУАЛИЗАЦИЯ [rendering]. Процедура подготовки информации для отображения на графическом устройстве. Например, процедура создания *задания на печать*. Приложение обращается к *интерфейсу графических устройств*, который принимает информацию документа, отправленную ему приложением, вызывает *драйвер принтера*, ассоциированный с печатающим устройством, и строит задание на печать, используя язык принтера этого печатающего устройства

ВИЗУАЛЬНЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ [visual programming language]. Язык взаимодействия пользователя с *системой программирования*, реализуемый диалоговыми средствами *графического интерфейса пользователя*. Появление В. я. п. явилось следствием сложности программирования графического интерфейса программ, предназначенных для выполнения в операционных средах типа Windows. Чтобы облегчить этот процесс, в последнее время разрабатываются системы программирования, позволяющие программировать элементы графического интерфейса пользователя с помощью средств этого же интерфейса. Например, программирование диалого-

вого окна ввода данных сводится к рисованию его на экране или к выбору и корректировке предлагаемого прототипа в режиме диалога. При этом применяются *указатель мыши*, *кнопки*, *меню* и другие *элементы управления*. На В. я. п. можно разрабатывать как оболочки к уже существующим программам, так и новые *программные продукты*. Для этого в В. я. п. разработан удобный механизм написания и подключения процедур на базовом *языке программирования высокого уровня* (Бейсик, Паскаль, Си). При написании текстов процедур в В. я. п. может использоваться цвет, например, выделение цветом различных синтаксических конструкций (комментарии выделяются одним цветом, а управляющие операторы — другим). Одним из первых В. я. п. является язык интегрированной инструментальной оболочки Visual Basic, позволяющий даже начинающим программистам разрабатывать программные изделия малой и средней степени сложности, ориентированные на работу в среде Windows

ВИНЧЕСТЕР [Winchester]. То же, что *винчестерский диск*

ВИНЧЕСТЕРСКИЙ ДИСК, **винчестер**, **жесткий диск [Winchester disk**, **hard disk]**. Постоянно закрепленный в дисководе *жесткий магнитный диск* или пакет дисков, заключенный вместе с *головками записи/чтения* в герметически закрытый корпус, заполненный очищенным от пыли обычным воздухом под атмосферным давлением. При вращении диска над ним создается "воздушная подушка", обеспечивающая зависание головок над поверхностью диска на высоте порядка нескольких микрон. Такая конструкция позволила существенно уменьшить величину и массу головок, толщину магнитного покрытия диска и при этом значительно увеличить *плотность записи*, а следовательно, и емкость дисков. Конструкция разработана в 1973 г. в филиале фирмы IBM в г. Винчестер (Великобритания). В. д. широко применяется в *персональных компьютерах* для записи, долговременного хранения и быстрого чтения больших объемов информации. Емкость В. д. *персональных компьютеров* может достигать нескольких десятков гигабайт при времени записи/считывания одного бита, измеряемом в долях наносекунд

ВИРТУАЛЬНАЯ МАШИНА [virtual machine]. *Вычислительная система* заданной конфигурации, моделируемая для пользователя *программными* и *аппаратными средствами* конкретной реально существующей ЭВМ. Обычный программист редко имеет дело с *машинными командами*, которыми управляются конкретные электронные и механические устройства, составляющие реальную ЭВМ. Также он не работает с данными на уровне *ячеек памяти* и регистров. Между программистом и реальной машиной существует слой *программного обеспечения*, преобразующий для программиста аппаратную машину в виртуальную, которая обладает нужными ему ресурсами, алгоритмами, структурами данных и управляется более удобным, нежели машинные команды, способом. Конфигурация В. м. может существенно отличаться от реальной. Так, ее *оперативная память* может быть значительно больше (см.

виртуальная память), а работа отсутствующих в реальной ЭВМ устройств может выполняться за счет совместных действий других устройств или моделироваться программами (см. *эмуляция*). В результате пользователь воспринимает В. м. как компьютер, имеющий архитектуру, отличную от реально существующей. Например, пользователь Паскаля работает с В. м. Паскаля, которую определяют структуры данных и алгоритмы, написанные на этом языке. При этом серьезному пользователю необходимо глубокое понимание В. м. Паскаля и того, как она моделируется на реальном компьютере, но существенно меньший интерес представляют сведения о работе компилятора при переводе программы, написанной на Паскале, в программу на машинном языке. Кроме того, наличие *многозадачной операционной системы* позволяет выделять каждой из одновременно выполняемых на компьютере задач свою В. м., обеспечивая *распределение ресурсов* и определенный уровень *защиты данных*

ВИРТУАЛЬНАЯ МАШИНА JAVA [Java virtual machine (JVM)]. Описание *виртуальной машины*, определяющей семантику *языка программирования Java.* Разработана корпорацией Sun Microsystems и включает в себя: 1) описание формата файла, который должен создавать транслятор языка Java; 2) описание способа интерпретации исполняемой части этого файла (так называемый *байт-код*); 3) описание процедур проверки байт-кода на соблюдение требований безопасности, которые должны проводиться до начала интерпретации байт-кода. Существует множество реализаций JVM, разработанных как корпорацией Sun Microsystems, так и другими фирмами практически для всех программных и аппаратных платформ

ВИРТУАЛЬНАЯ ПАМЯТЬ [virtual storage, virtual memory]. 1. Механизм управления памятью *вычислительной системы*, позволяющий программе использовать память, размер которой больше реальной *оперативной памяти*, имеющейся у компьютера. Понятие "В. п." включает в себя как управляющие памятью *аппаратные* и *программные средства*, так и способ управления памятью, который состоит в следующем. Выполняемой задаче отводится *физическая память*, состоящая частично из оперативной памяти и частично из более медленных и более дешевых *внешних запоминающих устройств.* Эта память разбивается на страницы фиксированной длины или сегменты переменной длины, параметры которых записываются в специальные таблицы страниц или сегментов. С помощью этих таблиц устанавливается соответствие между *виртуальными адресами*, указанными в программе, и *физическими адресами* запоминающих устройств. Процессор работает с командами и данными, находящимися в оперативной памяти, до тех пор, пока не произойдет обращение к команде или операнду, содержащимся в странице или сегменте внешней памяти. В этом случае выполнение программы прерывается и происходит переписывание нужной страницы или сегмента из *внешней памяти* в оперативную. Если в оперативной памяти не оказалось свободного места, то оно предварительно высвобождается, для чего содержимое одной из

страниц или сегмента оперативной памяти сохраняется на внешнем устройстве. После такого обмена страницами или сегментами соответственно корректируются таблицы страниц или сегментов и выполнение программы продолжается. См. *свопинг, подкачка, страничный обмен.* 2. Память *виртуальной машины*

ВИРТУАЛЬНАЯ ФУНКЦИЯ [virtual function]. Функция *абстрактного класса*, для которой не указан метод реализации в рамках описания данного класса. Реализация В. ф. должна быть указана в одном из классов, которые *наследуют* данному классу

ВИРТУАЛЬНОЕ СОЕДИНЕНИЕ, логическое соединение [virtual connection]. Воображаемое *соединение* между двумя конечными точками в процессе *передачи данных* по *компьютерной сети.* При этом фактический маршрут следования данных выбирается по мере их передвижения. См. *сеть с коммутацией пакетов*, *дейтаграмма*

ВИРТУАЛЬНЫЙ [virtual]. Реально не существующий или воспринимаемый иначе, чем реализован. Например, понятие "виртуальная реальность" означает мир, созданный компьютерными средствами. Он реально не существует, однако компьютер может воздействовать на зрительные, слуховые и другие органы чувств человека, вызывая у него иллюзию погружения в этот мир. Кроме того, человек может влиять на события, происходящие в этом мире, что усиливает ощущение реальности. Простым примером погружения в виртуальную реальность является компьютерная игра

ВИРТУАЛЬНЫЙ АДРЕС [virtual address]. 1. *Адрес* в *виртуальной машине.* 2. То же, что *логический адрес*

ВИРТУАЛЬНЫЙ ДИСК [virtual disk]. То же, что *электронный диск*

ВИРУС [virus]. То же, что *компьютерный вирус*

ВИСЯЧАЯ СТРОКА [orphan, widow]. Нарушение правила *верстки страниц*, согласно которому при переносе *абзаца* с одной страницы на другую на странице должно присутствовать не менее определенного числа строк данного абзаца (обычно не менее двух строк). Таким образом, возможны два случая В. с.: при переносе абзаца на первой странице остается только одна строка ("сирота" — orphan) или на следующую страницу переносится только одна строка ("вдова" — widow). Современные *текстовые процессоры*, например MS Word, позволяют автоматически отследить выполнение этого правила и не допустить появления В. с.

ВКЛЮЧЕНИЕ ФАЙЛА [file inclusion]. Осуществляемое *препроцессором* включение содержимого указанного файла в заданное место *исходного текста программы.* Таким образом в программу вносится необходимая информация, содержащаяся в других файлах. Так, в языке Си для программирования В. ф. предназначена директива `#include`. Когда препроцессор "распознает" эту ди-

рективу, он ищет следующее за ней в угловых скобках или двойных кавычках имя файла и включает его содержимое в текущий файл перед его компиляцией. Например, чтобы иметь возможность пользоваться в программе функциями ввода и вывода символов (`getchar()` и `putchar()`), макроопределения которых обычно содержатся в *библиотечном файле* языка Си с именем stdio.h, достаточно в начале программы указать: `#include <stdio.h>`

ВКЛЮЧИТЬ [switch on]. Инициировать функционирование устройства. Например, при включении принтер становится доступным в используемой *конфигурации компьютера*. Противоп. *выключить*

ВЛОЖЕНИЕ [attachment]. То же, что *вложенный файл*

ВЛОЖЕННОЕ МЕНЮ [submenu]. То же, что *подменю*

ВЛОЖЕННЫЙ АВТОМАТ [submachine]. *Конечный автомат*, который может быть вызван как часть другого конечного автомата

ВЛОЖЕННЫЙ БЛОК, подблок [internal block, nested block]. *Блок программы*, определенный внутри другого блока. Любой блок можно охарактеризовать некоторым уровнем вложения.

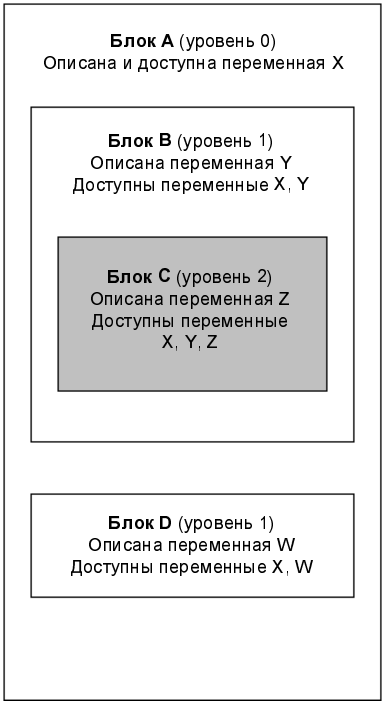

Рис. В.2. Структура вложенных блоков

Если саму программу, т. е. самый внешний блок отнести к уровню 0, то следующий уровень ее В. б. будет равен 1. В общем случае блок, описанный в блоке с уровнем k, будет иметь уровень вложения (k + 1). Структура В. б. показана на рис. В.2

ВЛОЖЕННЫЙ ФАЙЛ, **прикрепленный файл**, **вложение [attached file**, **attachment]**. *Файл*, посылаемый по *электронной почте* вместе с сообщением. См. *многоцелевые расширения почты Интернета*

ВЛОЖЕННЫЙ ЦИКЛ [internal cycle, **nested loop]**. *Цикл*, находящийся в *теле* другого *цикла*. Например, один и тот же алгоритм, содержащий В. ц. на языках Паскаль, Си и Фортран, соответственно, имеет вид:

Паскаль

```
for i:= 1 to 20 do
for k:= i +1 to 20 do
  begin
  w:= x[i,k];
  x[i,k]:= x[k,i];
  x[k,i]:= w
  end;
```

Си

```
for (i = 1; i <= 20; i++)
for (k = i + 1; k <= 20;
k++){
 w = x[i,k];
 x[i,k] = x[k,i];
 x[k,i] = w;
}
```

Фортран

```
    DO 10 I = 1, 20
    K = I + 1
    DO 10 J = K, 20
    W = X(I,J)
    X(I,J) = X(J,I)
10  X(J,I) = W
```

См. *операторы цикла языка Паскаль, операторы цикла языка Си, оператор цикла языка Фортран*

ВНЕДРЕННЫЙ ОБЪЕКТ [embedded object]. Объект, созданный средствами одного приложения (приложение-источник) и помещенный в документ другого приложения (документ-контейнер). Например, *диаграмма* Microsoft Excel, вставленная в документ Microsoft Word. При этом в документе-контейнере не хранится ссылка на документ-источник и В. о. не обновляется автоматически при его изменении в источнике. Ср. *связанный объект*. Для создания В. о. необходимо, чтобы оба приложения поддерживали *связывание и внедрение объектов* (OLE). *Двойной щелчок* по В. о. позволяет активизировать создавшее его приложение и приступить к редактированию объекта. Ср. *импорт*

ВНЕШНЕЕ ЗАПОМИНАЮЩЕЕ УСТРОЙСТВО (ВЗУ) [backing storage, **external storage]**. *Запоминающее устройство*, не относящееся к *оперативной памяти*. Набор ВЗУ реализует *внешнюю память вычислительной системы*, доступ к данным которой осуществляется средствами ввода/вывода. Внешняя память предназначена для длительного хранения информации, поэтому *носители данных*, применяемые в ВЗУ, не являются энергозависимыми — при выключении компьютера данные на них сохраняются. Существуют ВЗУ на *магнитных дисках*, *магнитных лентах* и т. п. В *персональных компьютерах* типичными ВЗУ являются *винчестерский диск, накопители на гибких магнитных дисках, лазерные диски, диски Бернулли* и т. д.

ВНЕШНЕЕ ПРЕРЫВАНИЕ [external interrupt]. *Прерывание*, инициируемое *внешним устройством*. Например, прерывание, вызванное таймером, указывающим, что истек квант времени *центрального процессора*, выделенный для выполнения данной программы, или прерывание, вызванное нажатием пользователем *персонального компьютера* клавиш <Ctrl> и <Break>

ВНЕШНЕЕ УСТРОЙСТВО [external device]. *Устройство* компьютера, не относящееся к *центральному процессору* и *оперативной памяти*, или *периферийное устройство вычислительной системы*. В. у. обычно являются *устройства ввода/вывода*: терминалы, дисководы, стримеры, принтеры, плоттеры, сканеры, дигитайзеры и т. п. Часто одним и тем же В. у. могут пользоваться несколько компьютеров

ВНЕШНИЙ КЛЮЧ [foreign key]. То же, что *вторичный ключ*

ВНЕШНЯЯ КОМАНДА MS-DOS [MS-DOS external command]. *Команда операционной системы MS-DOS*, не обрабатываемая непосредственно *командным процессором*. В. к. MS-DOS выполняет действия обслуживающего характера (например, форматирование или копирование дискет) и реализуется программой — *исполняемым файлом*, имя которого и является В. к. MS-DOS. Эта программа устанавливается с дистрибутива и поэтому входит в состав *операционной системы MS-DOS*. В остальном программа ничем не отличается от других программ, с которыми работают пользователи. Так же, как и другие программы, В. к. MS-DOS запускаются командным процессором command.com по команде пользователя, в которой имя команды совпадает с именем исполняемого файла. Такой механизм выполнения позволяет постоянно пополнять систему В. к. MS-DOS. В других ситуациях в настоящее время этот термин не используется. Противоп. *внутренняя команда MS-DOS*

ВНЕШНЯЯ ПАМЯТЬ [backing storage, external memory]. *Память*, к содержимому которой можно обратиться только при помощи операций ввода/вывода. В. п. реализуется набором *внешних запоминающих устройств вычислительной системы*. К данным, хранящимся во В. п., *центральный процессор* непосредственно обращаться не может. Для этого они должны быть предварительно считаны в *оперативную память*

ВНЕШНЯЯ ССЫЛКА, межмодульная ссылка [external reference, inter modular reference]. 1. То же, что *межмодульная ссылка*. *Ссылка* в одном *программном модуле* на объект, определенный в другом программном модуле, т. е. использование в одном модуле имени или метки, которые определены в другом модуле и значения которых определяются в процессе компоновки 2. Ссылка на одном документе или веб-странице, указывающая на ресурс на другом документе или веб-странице. Ср. *внутренняя ссылка*

ВНУТРЕННЕЕ ПРЕРЫВАНИЕ [internal interrupt]. *Прерывание*, инициируемое *центральным процессором*. Например, прерывание, вызванное делением на ноль

ВНУТРЕННИЙ ПЕРЕХОД [internal transition]. Переход в *конечном автомате*, у которого есть действие, но который не ведет при этом к изменению состояния

ВНУТРЕННИЙ ЯЗЫК [internal language]. *Язык программирования*, на который в ходе трансляции переводится *исходная программа*. Например, *машинный язык* или *промежуточный язык*

ВНУТРЕННЯЯ КОМАНДА MS-DOS [MS-DOS internal command]. *Команда операционной системы MS-DOS*, выполняемая непосредственно *командным процессором* MS-DOS (программой command.com). Программы выполнения В. к. MS-DOS являются составной частью командного процессора. По мысли авторов MS-DOS В. к. MS-DOS — наиболее часто используемые команды, программы выполнения которых должны постоянно находиться в *оперативной памяти* или быть быстро вызванными в нее. В других ситуациях в настоящее время этот термин не используется. Противоп. *внешняя команда MS-DOS*

ВНУТРЕННЯЯ ПАМЯТЬ [internal memory]. То же, что *основная память*

ВНУТРЕННЯЯ ССЫЛКА [inline link]. Ссылка, указывающая на объект в том же программном модуле, документе или на той же веб-странице. Ср. *внешняя ссылка, межмодульная ссылка*

ВОДОПАДНЫЙ ПРОЦЕСС РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ [waterfall software development process]. *Процесс разработки программного обеспечения*, состоящий из следующих фаз: *анализ требований*, *проектирование*, *реализация*, *тестирование*. Все фазы выполняются последовательно, возможно лишь небольшое наложение фаз. В. п. р. п. о. плохо подходит для случая, когда требования к *программному продукту* меняются в процессе разработки. Ср. *инкрементный процесс разработки программного обеспечения*

ВОЗВРАТ [return]. *Передача управления* из вызываемой программы или подпрограммы в *вызывающую программу*. См. *вызов подпрограммы*

ВОЗВРАТ В ИСХОДНОЕ СОСТОЯНИЕ [reset]. Приведение системы в исходное состояние. Выполняется, как правило, при возникновении сбоев, ошибок и тому подобных случаях. Так, нажатие кнопки Reset на *системном блоке персонального компьютера* приводит к перезагрузке *операционной системы*. См. *загрузка операционной системы*

ВОЗВРАТНЫЙ ВЫЗОВ, **обратный вызов [callback]**. Схема установления связи между *абонентами сети*, при которой модем вызываемого абонента вместо ответа на поступивший вызов требует, чтобы вызывающий абонент ввел *тональный код* и дал отбой. Далее модем проверяет код вызывающего абонента по имеющемуся набору телефонных номеров. Если полученный код соответствует какому-либо зарегистрированному номеру, модем набирает этот номер и открывает соединение с вызывающим абонентом. Модемы,

способные выполнить В. в. используются, когда требуется защитить данные от *несанкционированного доступа*. Ср. *функция обратного вызова*

ВОКОДЕР [vocoder]. *Аппаратное* или *программное средство*, реализующее алгоритм сжатия аудиоданных, содержащих преимущественно человеческую речь, для передачи по линиям связи

ВОПЛОЩЕНИЕ [reification]. Применительно к *объектно-ориентированному программированию* способ считать объектом некоторую сущность, которая обычно не рассматривается как объект. Примером может служить динамическое поведение. Большинство представляет себе операцию как объект, но что же тогда сказать о выполнении операции? Как правило, люди представляют это в виде процесса, но стоит дать имя В. этого процесса — оно называется *активацией* — и можно задавать ему свойства, изображать его отношения с другими объектами, манипулировать им. В. помогает трансформировать динамические процессы в структуры данных, которыми можно управлять. Оно представляет собой одну из наиболее мощных концепций в моделировании и программировании

ВОССТАНОВЛЕНИЕ, **регенерация [recovery**, **regeneration]**. Процесс возврата к нормальной работе после *машинного сбоя*, *программной ошибки* или *конфликтной ситуации*. Например, процесс В. данных, разрушенных в результате программной ошибки, путем считывания с носителя, содержащего *резервную копию* данных. Существуют специальные программы В., помогающие производить поиск и восстановление поврежденной информации. См. *возврат в исходное состояние, восстановление файла*. Ср. *перезапись*

ВОССТАНОВЛЕНИЕ ФАЙЛА [file recovery]. Процесс восстановления в *корневом каталоге магнитного диска* информации об удаленном файле. Например, в *операционной системе MS-DOS* для этого есть команда `undelete`. В *операционной системе Windows* В. ф. выполняется с помощью *Корзины*. См. *удаление файла*

ВОСХОДЯЩЕЕ ПРОГРАММИРОВАНИЕ [bottom up programming]. Способ *разработки программ*, при котором программирование ведется методом "снизу-вверх", от деталей к общему. Сначала разрабатываются и тестируются функции (подпрограммы) нижнего уровня. Затем на их основе программируются функции более высокого уровня и т. д. При этом структура и функциональное назначение функций более высоких уровней вытекают из функций нижних уровней. Противоп. *нисходящее программирование*

ВОСЬМЕРИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ [octal number system]. *Позиционная система счисления* с основанием системы $q = 8$. В качестве восьмеричных цифр используются символы 0, 1, 2, 3, 4, 5, 6, 7. Например, $(137)_8 = 1 \times 8^2 + 3 \times 8^1 + 7 \times 8^0 = (95)_{10}$

ВОСЬМЕРИЧНОЕ ЧИСЛО [octal number]. *Число*, представленное в *восьмеричной системе счисления*

ВРЕМЕННЫЙ ОБЪЕКТ [transient object]. Объект, который существует только во время выполнения создавшего его процесса

ВРЕМЕННЫЙ ФАЙЛ, **рабочий файл [temporary file (temp file)]**. Файл, создаваемый *операционной системой* или *прикладной программой* на время ее работы. В. ф. создается в оперативной памяти или на диске и по завершении работы программы уничтожается. Во многих операционных системах В. ф. имеют расширение tmp

ВРЕМЯ ВЫБОРКИ [fetch time]. Интервал времени между началом обработки запроса к *запоминающему устройству* и получением от него запрошенных данных. В. в. включает в себя время, которое требуется для поиска *элемента данных* в памяти, считывания и пересылки. Ср. *время доступа*

ВРЕМЯ ВЫПОЛНЕНИЯ [execute time, **running time**, **run time]**. 1. Обозначение времени, в течение которого программа непосредственно выполняется, в отличие от времени, когда она загружается или компилируется. Например, сообщение операционной системы "run time error" (ошибка времени выполнения) означает, что ошибка произошла (обнаружена) при выполнении программы. Ср. *время компиляции.* 2. То же, что *время прогона*

ВРЕМЯ ДОСТУПА [access time]. Интервал времени между моментом выдачи команды на ввод или вывод данных и моментом начала записи или считывания на запоминающем устройстве. В. д. характеризует скорость поиска нужного *элемента данных* и может исчисляться наносекундами при обращении к *оперативной памяти* или секундами при обращении к *магнитной ленте.* Ср. *время выборки*

ВРЕМЯ КОМПИЛЯЦИИ [compilation time, **compiling time**, **compile time]**. 1. Обозначение времени, в течение которого исходный модуль программы преобразуется в объектный модуль. Ср. *время выполнения.* 2. Величина интервала времени, в течение которого исходный модуль программы преобразуется в объектный модуль. Измеряется в секундах, микросекундах и т. п.

ВРЕМЯ ОТВЕТА, **время отклика**, **время реакции [response time]**. Интервал времени между вводом запроса в *вычислительную систему* и получением от нее ответной реакции. Например, время между вводом запроса на поиск некоторых данных и выводом первой строки запрошенных данных. В. о. является важнейшей характеристикой *диалоговой системы.* Если В. о. слишком велико, то система не отвечает своему назначению. Обычно приемлемым считается В. о., не превышающее нескольких секунд

ВРЕМЯ ОТКЛИКА [response time]. То же, что *время ответа*

ВРЕМЯ ПОИСКА [seek time]. Время, необходимое для перемещения *головки записи/чтения* дисковода к указанному месту на диске. См. *поиск дорожки*

ВРЕМЯ ПРОГОНА, **время выполнения [running time**, **execute time]**. Величина интервала времени между моментом запуска и моментом завершения программы. Измеряется в секундах, микросекундах и т. п. Ср. *время центрального процессора*

ВРЕМЯ ПРОСТОЯ [downtime, **idle time]**. Время, в течение которого *вычислительная система*, процесс или устройство не выполняли своих функций. Это может быть и время, в течение которого компьютер был выключен, и время, затраченное на устранение неисправности, и время работы вхолостую, когда исправный и включенный компьютер не выполнял команды и не передавал данные

ВРЕМЯ РЕАКЦИИ [response time]. То же, что *время ответа*

ВРЕМЯ ТАКТА [loop time]. То же, что *такт*

ВРЕМЯ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА, **процессорное время [CPU time]**. Ресурс, предоставляемый задаче *вычислительной системой.* Характеризует трудоемкость задачи. В *однозадачных системах* — время, в течение которого *центральный процессор* обслуживал данную задачу, включая время, в течение которого он находился в состоянии ожидания, например, при *обмене данными* между *оперативной памятью* и *внешними устройствами.* В *многозадачных системах* и *системах разделения времени* — суммарное время, в течение которого центральный процессор выполнял команды данной задачи. Ср. *время прогона*

ВСЕМИРНАЯ ПАУТИНА [World Wide Web (WWW)]. Гипертекстовая система поиска ресурсов в *Интернете* и доступа к ним. В. п. предоставляет набор услуг Интернета, позволяющий просмотреть любые данные, хранящиеся в компьютерах этой сети, через систему связывающих их *гиперссылок.* Основу *аппаратного обеспечения* В. п. образует множество объединенных в Интернете компьютеров, расположенных по всему миру. Информационную основу В. п. составляет множество хранящихся в памяти этих компьютеров и содержащих гипертекст документов, называемых *веб-документами.* В веб-документах пользователь может, указав на выделенное слово или фразу, переходить к документам или получать доступ к файлам, хранящимся в памяти компьютеров в разных точках земного шара. Поэтому пользователь воспринимает В. п. как Всемирную виртуальную информационную паутину. Поскольку В. п. использует компьютеры Интернета и стандартные протоколы для передачи файлов и документов, термин "В. п." часто относят как к Всемирной сети компьютеров, так и собственно к информации. Доступ к веб-документам осуществляется с применением *архитектуры "клиент-сервер".* Веб-сервер — это программа, запущенная на компьютере, предназначенном для предоставления документов другим машинам. Веб-клиент — программа,

которая дает возможность пользователю запрашивать документы с сервера. Обычно она называется *программой просмотра* или браузером. Для установления связи с сервером используется его сетевой адрес, который называется *универсальным указателем ресурса (URL).* Язык и правила, по которым общаются между собой клиенты и серверы В. п., определяются *протоколом передачи гипертекста* (HyperText Transmission Protocol, HTTP). HTTP позволяет запрашивать, получать и отображать данные *гипермедиа*, включающие текст, изображения, звук. См. *протоколы TCP/IP*

ВСПЛЫВАЮЩЕЕ МЕНЮ [pop-up menu]. *Меню*, появляющееся на экране дисплея в текущем положении курсора и исчезающее после выбора *пункта меню.* На экране дисплея такое меню перекрывает выведенную ранее информацию. Обычно в виде В. м. реализуются *контекстные меню*

ВСТАВКА [paste]. В приложениях *операционной системы Windows* — копирование содержимого *буфера обмена* в документ. При этом содержимое буфера обмена сохраняется и может быть вставлено произвольное количество раз. В зависимости от типа данных, находящихся в буфере обмена, и возможностей приложения, вставка может быть выполнена различными способами. Например, на рис. В.3 представлено диалоговое окно **Специальная вставка** приложения MS Word, показывающее различные способы вставки текста в документ MS Word. См. *вырезание, копирование*

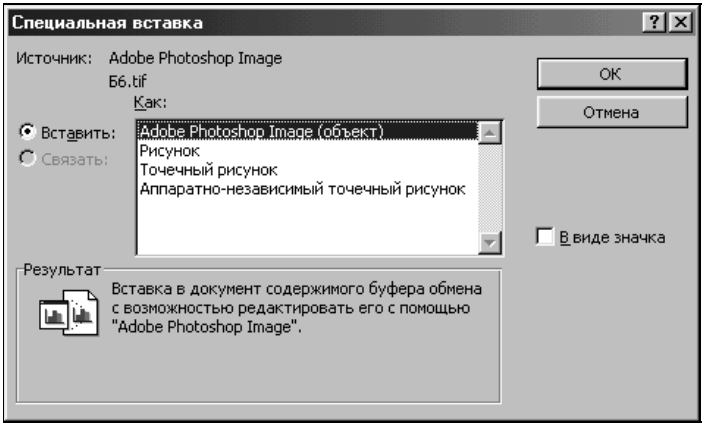

Рис. В.3. Диалоговое окно **Специальная вставка** приложения Microsoft Word

ВСТРОЕННАЯ ПРОЦЕДУРА, стандартная процедура [build-in procedure, standard procedure]. *Процедура*, действия которой определены *языком программирования.* Например, процедуры ввода/вывода языка Паскаль: `Read`, `Readln`, `Write`, `Writeln`. Реализация В. п. обеспечивается соответствующей

системой программирования, что делает В. п. одинаково доступной для всех программ на этом языке. Ее применение не требует *описания процедуры* в программе, т. к. вызов В. п. автоматически распознается и компилируется транслятором

ВСТРОЕННАЯ СИСТЕМА [embedded system]. *Вычислительная система*, выполненная на базе *встроенного компьютера* и ориентированная на специальное применение. Например, система управления транспортным средством или сложной бытовой техникой. Как правило, во В. с. применяется специализированное *программное* и *аппаратное обеспечение*

ВСТРОЕННАЯ ФУНКЦИЯ, **стандартная функция [built-in function**, **standard function]**. *Функция*, определенная версией *языка программирования*. Реализация такой функции обеспечивается соответствующей *системой программирования*. Например, математические функции `cos`, `sin`, `exp` и т. п. Их применение не требует *описания функции* в программе, поскольку вызов В. ф. автоматически распознается и компилируется транслятором. Практически в каждом языке программирования определен широкий набор математических В. ф., В. ф. обработки строк и других В. ф. преобразования данных

ВСТРОЕННАЯ ЭВМ [build-in computer]. То же, что *встроенный компьютер*

ВСТРОЕННОЕ ИЗОБРАЖЕНИЕ [in-line image]. Графический элемент, размещаемый в одной строке с текстом. Например, в данном абзаце имеется В. и. стандартной кнопки **Сохранить**

ВСТРОЕННЫЕ ПРОГРАММЫ, **зашитые программы [firmware]**. *Программное обеспечение*, хранящееся в микросхемах *постоянного запоминающего устройства*. Пользователь не может их заменить или модифицировать. В отличие от *оперативной памяти* постоянная память остается невредимой даже при отсутствии электропитания. Поэтому здесь хранятся жизненно важные компоненты *программного обеспечения* компьютера. В. п., в частности, являются программы запуска и команды ввода/вывода низкого уровня. С точки зрения сложности такие программы находятся между уровнем обычного программного обеспечения и уровнем *аппаратного обеспечения*

ВСТРОЕННЫЙ ИНТЕРФЕЙС НАКОПИТЕЛЕЙ, **интерфейс IDE [integrated drive electronics**, **integrated device electronics (IDE)]**. Тип *интерфейса* для подключения дисководов (дисковых накопителей, накопителей CD-ROM, стримеров и т. д.), у которых контроллер встроен в сам привод, т. е. без отдельной платы адаптера дисковода. Для подключения такого накопителя к компьютеру не требуется разъем шины, что более всего привлекает в этом интерфейсе. Подключение осуществляется через специальный разъем на *материнской плате*, служащий мини-разъемом, к которому подведены только нужные для В. и. н. сигналы

ВСТРОЕННЫЙ КОМПЬЮТЕР, встроенная ЭВМ [built-in computer]. *Компьютер*, являющийся неотъемлемой конструктивной частью прибора или оборудования. См. *встроенная система*

ВСТРОЕННЫЙ ТИП, **стандартный тип**, **примитивный тип [build-in type**, **standard type**, **primitive type]**. В *языках программирования высокого уровня — тип данных*, определенный стандартом данного языка. В большинство существующих языков программирования встроены *целый, вещественный, логический* и *символьный типы*

ВТОРИЧНОЕ ОКНО [secondary window]. Подчиненное окно, которое появляется поверх основного окна. Например, окно в справочной системе Windows (рис. В.4). В. о. может иметь различные положение, цвет и размер

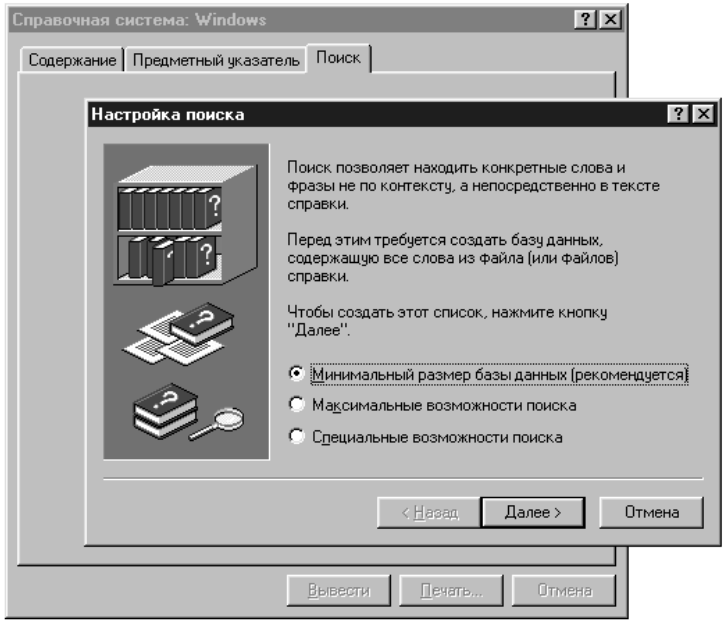

Рис. В.4. Вторичное окно

ВТОРИЧНЫЙ КЛЮЧ, **внешний ключ, неосновной ключ [secondary key**, **foreign key]**. *Поле записи*, которое содержит *первичный ключ* другого файла (или таблицы). Например, на рис. В.5 показана *схема данных*, в которой имеются две таблицы: "Отделы" и "Работники". В таблице "Работники" в поле "Отдел" для каждого работника хранится код отдела, в котором он работает. Здесь поле "Отдел" является В. к.

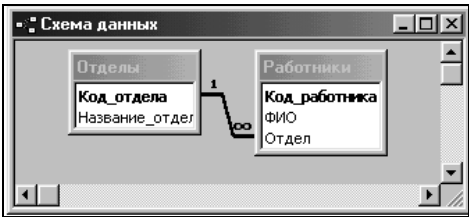

Рис. В.5. Схема данных в приложении Microsoft Access

ВХОД В ПОДПРОГРАММУ [subroutine entry]. См. *вход в программу*

ВХОД В ПРОГРАММУ [program entry]. Оператор или команда, которым передается управление при *запуске программы* или *вызове подпрограммы*. См. *точка входа*

ВХОД В СИСТЕМУ, **регистрация [login**, **logon]**. Процедура, обеспечивающая доступ к *вычислительной системе* зарегистрированных пользователей. Например, работа с системой начинается с сообщения ей имени и пароля. Обычно В. в с. предваряет начало сеанса работы пользователя за одним из удаленных терминалов или *рабочих станций локальной сети*. Противоп. *выход из системы*. См. *учетная запись пользователя*

ВХОДНЫЕ ДАННЫЕ [input data]. *Данные*, вводимые в *вычислительную систему* через *устройства ввода* для обработки или хранения. Ср. *исходные данные*

ВЫБОР ОБЪЕКТОВ [marquee]. То же, что *выделение области*

ВЫБОР ПО УМОЛЧАНИЮ [default option]. Выбор системой стандартных значений параметров или стандартных действий в случаях, когда пользователем не определены другие значения или действия. Например, если в *операторе цикла* языка Фортран не указан шаг изменения значения *параметра цикла*, то при трансляции он будет выбран равным единице. См. *значение по умолчанию*

ВЫБОРКА [fetching]. *Машинная операция*, с помощью которой осуществляется чтение команд или данных из *оперативной памяти* в регистры *устройства управления* или *арифметико-логического устройства*

ВЫБОРКА ДАННЫХ [data retrieval]. Процесс поиска и считывания данных из файла, группы файлов или *базы данных*

ВЫВОД [output]. То же, что *вывод данных*

ВЫВОД ДАННЫХ, **вывод [data output**, **output]**. Процесс *передачи данных* из *основной памяти* ЭВМ на экран дисплея, бумагу или другой подобный носи-

тель, а также на *внешние запоминающие устройства.* Выведенные из компьютера результаты *обработки данных* могут быть предназначены для последующего анализа человеком и для этого представляются в удобной для него форме (напечатанными или отображенными на экране дисплея), а могут быть подготовлены для ввода в другую *вычислительную систему* или для использования в другой программе и тогда записываются в закодированном виде на *магнитный диск, магнитную ленту* и т. п. В. д. производится *устройствами вывода*

ВЫГРУЗКА, **откачка [preemption]**. В *многозадачных системах* — перемещение задачи из *оперативной памяти* на диск с целью освобождения места для выполнения другой задачи с более высоким приоритетом. При этом состояние *вычислительного процесса* на момент В. сохраняется, чтобы его можно было продолжить

ВЫГРУЗКА ПО ЛИНИИ СВЯЗИ [uploading]. Процесс передачи копии файла из локального компьютера в удаленный компьютер. Местный компьютер через свой модем получает разрешение на В. п. л. с. от удаленного компьютера и начинает *передачу данных*, а удаленный компьютер сохраняет поступивший файл на диске. Ср. *загрузка по линии связи*

ВЫДЕЛЕНИЕ [select, **marquee]**. 1. В приложениях *операционной системы Windows* — особая операция *графического интерфейса пользователя*, с помощью которой отмечаются те данные, к которым в дальнейшем будут применяться другие операции. Например, перед тем как данные могут быть скопированы в *буфер обмена*, они должны быть выделены. Обычно выделенные данные отображаются особым образом, чтобы было видно, что именно выделено. Например, изображение выделенного текста инвертируется. На рис. В.6 представлен выделенный фрагмент документа Word и контекстное меню, показывающее различные операции, применимые к выделенному фрагменту. См. *вставка, вырезание, копирование* 2. То же, что *выделение области.* См. *лассо*

ВЫДЕЛЕНИЕ ОБЛАСТИ, **выбор объектов**, **выделение [marquee]**. *Инструмент машинной графики*, предназначенный для выделения на экране прямоугольной или эллиптической областей. После вызова инструмента В. о. можно с помощью мыши растянуть в нужном месте штриховой прямоугольник (или эллипс) так, чтобы заключить в него выделяемую область. На рис. В.7 для *графического редактора* Adobe Photoshop показана кнопка вызова инструмента В. о. и *диалоговое окно*, позволяющее установить форму, тип, пропорции и размеры выделенной области. См. *лассо*

ВЫДЕЛЕННАЯ ЛИНИЯ СВЯЗИ [dedicated line]. *Линия связи*, обслуживающая отдельного пользователя или зарезервированная для конкретного применения и не доступная другим абонентам сети

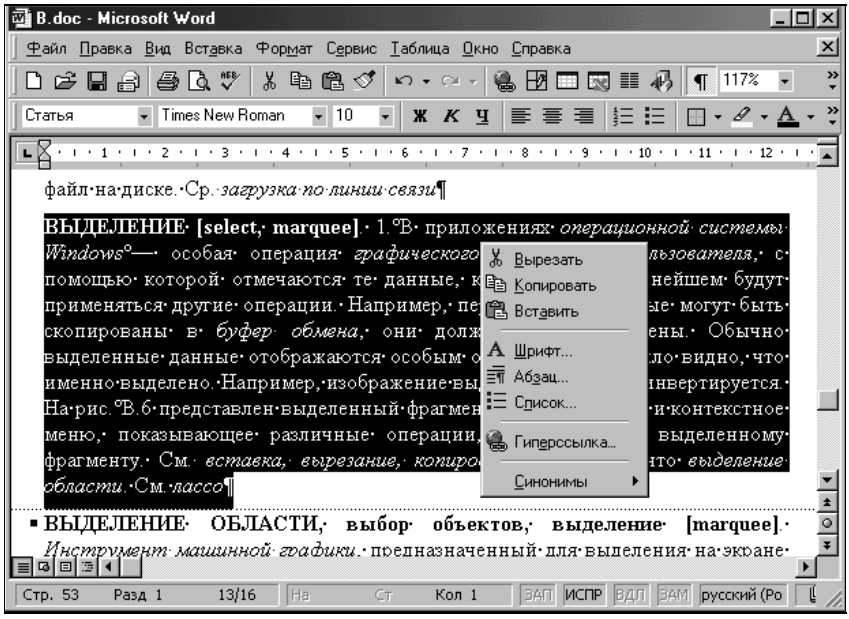

Рис. В.6. Выделенный фрагмент текста
и контекстное меню в приложении Microsoft Word

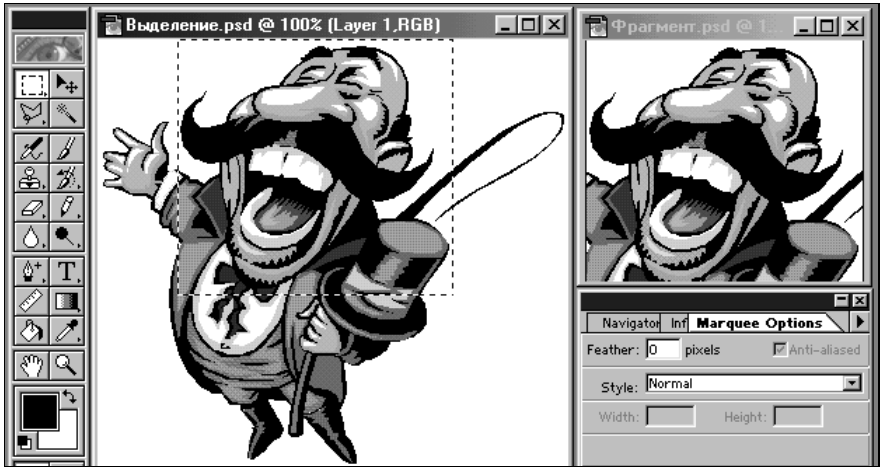

Рис. В.7. Результат выделения
и диалоговое окно инструмента "Прямоугольное выделение"
графического редактора Adobe Photoshop

ВЫЗОВ ПОДПРОГРАММЫ, **обращение к подпрограмме [call, subroutine call]**. 1. Последовательность *машинных команд вызывающей программы*, которые обеспечивают начало выполнения *подпрограммы*: *передачу* подпрограмме *исходных данных*, подготовку возврата в *вызывающую программу* и *передачу управления* на *вход в подпрограмму*, т. е. ее первой выполняемой команде. В *языках программирования высокого уровня* то же, что *оператор вызова подпрограммы, оператор процедуры* или *вызов функции*. 2. Передача подпрограмме исходных данных, подготовка возврата в вызывающую программу, передача управления на вход в подпрограмму, выполнение действий, запрограммированных в подпрограмме, и возврат в нужную точку программы. В языках программирования высокого уровня — выполнение *оператора вызова* подпрограммы: передача подпрограмме *фактических параметров*, передача управления на вход в подпрограмму, выполнение операторов подпрограммы и возврат управления вызывающей программе. См. *вызов процедуры, вызов функции, точка входа, точка возврата*

ВЫЗОВ ПРОЦЕДУРЫ, **обращение к процедуре [procedure statement, procedure call]**. 1. То же, что *оператор процедуры*. 2. Выполнение *оператора процедуры*: *передача* процедуре *фактических параметров*, *передача управления* первому выполняемому оператору *тела процедуры*, выполнение операторов и возврат управления вызывающей программе. См. *вызов подпрограммы*

ВЫЗОВ ФУНКЦИИ, **обращение к функции [function call]**. 1. Операнд выражения или оператор, предписывающий выполнение алгоритма *функции*. Состоит из имени, обозначающего функцию, и *списка* ее *фактических параметров*, определяющих конкретные данные, над которыми и выполняются запрограммированные функцией действия. Например, в *арифметическом выражении*, стоящем в правой части *оператора присваивания* `y:= x+sin(2*x)`, имеется В. ф. `sin(2*x)`. Здесь `sin` — имя функции, `2*x` — фактический параметр. Результатом выполнения функции является присвоенное имени функции значение, которое затем будет прибавлено к значению переменной `x`. В некоторых *языках программирования* допускается отсутствие списка фактических параметров. Правила записи В. ф. определяются синтаксисом конкретного языка программирования. Наиболее употребляемый вид В. ф.: `Name (x1, x2, ..., xn)` или `Name`, где `Name` — имя вызываемой функции или идентификатор дополнительной точки входа в функцию; `x1, x2, ..., xn` — фактические параметры, передаваемые процедуре. Они должны быть согласованы по количеству, порядку следования, классу и типу с соответствующими *формальными параметрами*, определенными в *описании функции*. В некоторых языках программирования (например, в Си) отсутствует понятие процедуры, поэтому В. ф. может употребляться в виде отдельного оператора *вызывающей программы*, аналогичного *оператору процедуры*. 2. Действия, связанные с выполнением функции: *передача* функции *фактических параметров*, *передача управления* первому выполняемому оператору *тела функции*, выполнение действий, указанных операторами тела функции,

возврат в *вызывающую программу* значения функции и возврат управления вызывающей программе. См. *вызов подпрограммы*

ВЫЗЫВАЕМАЯ ПОДПРОГРАММА [called subroutine]. *Подпрограмма*, которой передано управление *вызывающей программой*

ВЫЗЫВАЮЩАЯ ПРОГРАММА [calling program]. *Программа*, передающая управление другой программе или выполняющая *вызов подпрограммы*

ВЫКЛЮЧИТЬ [switch off]. Прекратить функционирование устройства. Например, при выключении устройство становится недоступным в используемой *конфигурации компьютера*. Это позволяет освободить занятые устройством ресурсы. Противоп. *включить*

ВЫКЛЮЧКА [justify, alignment]. То же, что *выравнивание*

ВЫНОСКА [callout]. Дополнительный текст на изображении, служащий для указания элементов изображения. Используются различные способы оформления выносок: линия с текстом, фигура с указательной стрелкой и др. (рис. В.8)

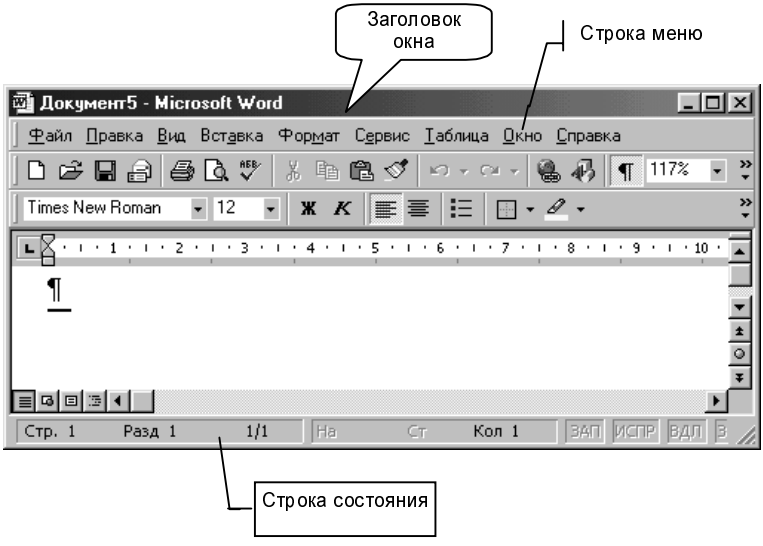

Рис. В.8. Различные способы оформления выносок

ВЫПОЛНЕНИЕ ПРОГРАММЫ, исполнение программы [program run, computer run, run]. Работа *компьютера* по заданной *программе*. См. *прогон программы*

ВЫПОЛНЯЕМЫЙ ОПЕРАТОР [imperative statement]. То же, что *исполняемый оператор*

ВЫРАВНИВАНИЕ, **выключка [justify]**. Полиграфический термин, означающий размещение текста на странице по определенным правилам. Различаются горизонтальное и вертикальное В. Горизонтальное В. может быть по левому краю, по правому краю, по ширине и по центру. В. по левому краю означает, что первые буквы последовательных строк текста находятся строго друг под другом и, таким образом, левый край текста оказывается выровненным. В. по правому краю означает, что последние буквы строк находятся друг под другом. В. по ширине означает одновременно В. по левому и по правому краю. В. по центру означает, что первая и последняя буквы строки находятся на одинаковом расстоянии от полей. В. применяется при *верстке страниц*, причем тип В. зависит от типа издания, национальных традиций и др. Например, в отечественной научной литературе принято выравнивать текст по ширине, а в англоязычной — по левому краю. На рис. В.9 приведены примеры различных типов горизонтального В. в приложении MS Word

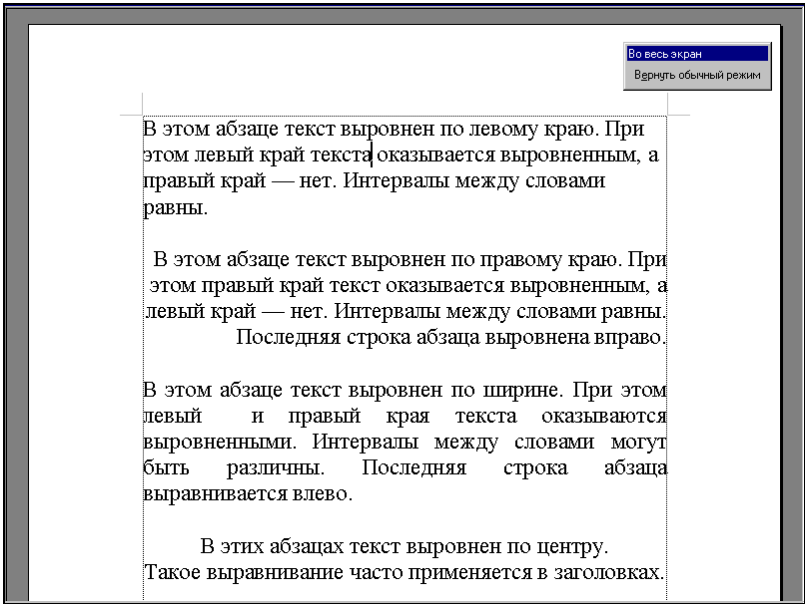

Рис. В.9. Различные типы горизонтального выравнивания текста

ВЫРАВНИВАНИЕ ТЕКСТА, **автоматический переход на новую строку [word-wrap]**. Режим *ввода текста*, при котором вводимый текст автоматически продолжается с новой строки по достижении правой границы области текста. В каком именно месте текста происходит переход на новую строку, зависит от применяемого типа *выравнивания* и от того, включен или нет ре-

жим автоматического *переноса слов*. В. т. используется во всех современных текстовых процессорах, например, в приложении MS Word

ВЫРАЖЕНИЕ [expression]. *Синтаксическая конструкция языка программирования высокого уровня*, имеющая вид формульной записи последовательности операций над данными (операндами). Выполняя эту последовательность операций, процессор вырабатывает значение В. В зависимости от типа вырабатываемого значения В. могут быть *арифметическими*, *логическими* и др. Правила записи В. устанавливаются конкретным языком программирования

ВЫРЕЗАНИЕ [cut]. В приложениях *операционной системы Windows* — копирование выделенных данных из документа в *буфер обмена* и последующее удаление скопированных данных из документа. См. *выделение, копирование*

ВЫСОКОПРОИЗВОДИТЕЛЬНЫЙ ФОРТРАН, язык HPF [High Performance Fortran (HPF)]. Расширение *языка программирования* Фортран для работы на *параллельных компьютерах*. HPF был разработан в середине 90-х годов прошлого столетия. Он основан на использовании специальных директив в комментариях к тексту программы, указывающих транслятору на параллелизм алгоритма. Ср. *технология MPI*, *технология PVM*

ВЫТЕСНЯЮЩАЯ МНОГОЗАДАЧНОСТЬ, **приоритетная многозадачность [preemptive multitasking]**. Поддерживаемый *многозадачной операционной системой* режим *мультипрограммирования*, при котором по истечении кванта времени решение о переходе *центрального процессора* с выполнения одной задачи на выполнение другой задачи принимает операционная система в соответствии с приоритетами задач. В. м. предоставляет операционной системе возможность захвата процессора в любой момент времени, независимо от состояния работающих приложений. Тем самым обеспечивается быстрая реакция, высокая пропускная способность и защищенность системы от действий некорректно работающих приложений. Ср. *кооперативная многозадачность*

ВЫХОД [exit]. Возврат в *вызывающую программу* из вызываемой программы. При В. происходит завершение выполнения вызванной программы и передача управления вызвавшей программе. Ср. *возврат*

ВЫХОД ИЗ ПРИЛОЖЕНИЯ [quit]. Корректная остановка *приложения*. По команде происходит нормальное завершение работы программы с передачей управления *операционной системе*

ВЫХОД ИЗ СИСТЕМЫ [logoff, logout]. Процедура завершения сеанса работы пользователя с *вычислительной системой*, происходившего по *линиям связи*. Обычно это окончание работы пользователя за одним из нескольких удаленных терминалов вычислительной системы или *рабочих станций локальной сети*, когда отключение терминала не означает выключение компьютера. Противоп. *вход в систему*. См. *учетная запись пользователя*

ВЫХОД ИЗ ЦИКЛА [loop termination]. Завершение *циклического процесса*. В. и. ц. может произойти либо после его выполнения, когда оказался исчерпанным *список цикла*, заданный в *операторе цикла*, либо с помощью *оператора условного перехода*, расположенного в *теле цикла*, и при достижении некоторого условия, задающего переход к оператору, расположенному вне цикла

ВЫХОДНЫЕ ДАННЫЕ [output data]. *Данные*, поступающие из компьютера на *устройства вывода* в результате выполнения программы

ВЫЧИСЛИТЕЛЬНАЯ МАШИНА, **компьютер [computer]**. *Устройство* или комплекс устройств, предназначенные для механизации или автоматизации обработки информации. Существуют два класса машин: *цифровые вычислительные машины* и *аналоговые вычислительные машины*. В настоящее время в основном используются *цифровые вычислительные машины*. Ср. *электронная вычислительная машина*

ВЫЧИСЛИТЕЛЬНАЯ СЕТЬ [computer network]. То же, что *компьютерная сеть*

ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА [computer system]. Совокупность *аппаратных* и *программных средств* компьютера, взаимодействующих для решения задач *обработки информации*. Например, В. с. является *персональный компьютер* с установленным на нем *программным обеспечением*

ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА MPP, система MPP [massively parallel processors system, MPP system]. Многопроцессорная *вычислительная система* с *распределенной памятью*, обладающая высокой степенью параллелизма. В. С. MPP может состоять из сотен и тысяч процессоров

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА [computer science, **computing machinery]**. 1. Совокупность научных дисциплин и отраслей техники, специализирующихся на создании *аппаратных средств* для обработки и хранения информации. Ср. *информатика*. 2. Совокупность вычислительных средств, предназначенных для автоматизации процесса решения задач: *компьютеров*, *внешних устройств*, абонентских пунктов, отдельных *терминалов*, средств *передачи данных*

ВЫЧИСЛИТЕЛЬНЫЙ КЛАСТЕР, кластер [computer cluster, cluster]. Совокупность компьютеров, объединенных в *локальную вычислительную сеть* для решения одной задачи путем проведения *параллельных вычислений*. В. к. нужно рассматривать как один многопроцессорный компьютер, не использующий систему разделения времени и работающий в режиме монопольного использования. В качестве *вычислительных узлов* обычно используются доступные на рынке однопроцессорные или *симметричные многопроцессорные компьютеры*. Каждый узел работает под управлением своей копии *операци-*

онной системы, чаще всего — стандартной: Linux, Windows NT и т. п. Состав и мощность узлов могут меняться даже в рамках одного В. к. Коммуникационная среда, обеспечивающая взаимодействие компьютеров в В. к., может создаваться с использованием стандартных сетевых технологий, исходя из соображений стоимости, производительности и масштабируемости. Применяются, например, и сети Ethernet со скоростью *передачи данных* 10 или 100 Мбит/с, и сети SCI со скоростью до 500 Мбит/с. Среди достоинств В. к. можно указать относительно низкую стоимость (В. к. из 10 *рабочих станций* дешевле десятипроцессорного суперкомпьютера), высокую надежность (выход из строя вычислительных узлов не обязательно приводит к остановке всей системы) и приспособленность к постепенному наращиванию и модификации без длительной остановки на замену оборудования. См. *Беовульф-кластер*

ВЫЧИСЛИТЕЛЬНЫЙ КОМПЛЕКС, **многомашинный вычислительный комплекс (ВК) [computer complex, multiple computer complex]**. Совокупность двух и более ЭВМ, работающих как единая *вычислительная система*

ВЫЧИСЛИТЕЛЬНЫЙ ПРОЦЕСС [calculation process]. Процесс решения задач на *компьютере*

ВЫЧИСЛИТЕЛЬНЫЙ УЗЕЛ, узел [node]. Компьютер, выполняющий вычисления в *компьютерной сети* или *вычислительном кластере*. Ср. *узел сети*

ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР (ВЦ) [computer center]. Самостоятельное учреждение или подразделение учреждения, оснащенное комплексом средств вычислительной техники и предназначенное для предоставления услуг по обработке и хранению данных. Если *ВЦ* снабжен *информационной системой*, то его называют *информационно-вычислительным центром*

ВЫЧИСЛЯЕМЫЙ ОПЕРАТОР ПЕРЕХОДА [evaluate `Go To` statement]. *Оператор перехода* в языке Фортран, имеющий вид:

```
GO TO (m1, m2, ..., mk), I
```

где `m1`, `m2`, ..., `mk` — метки операторов, `I` — переменная целого типа. Значение переменной `I` должно удовлетворять неравенству `1 ≤ I ≤ k`, где `k` — количество меток, заключенных в скобки. Если значение переменной `I` равно `i`, то выполнение программы продолжается с оператора, помеченного меткой `mi`, стоящей в скобках на *i*-м месте. Например, в операторе `GO TO (10, 7, 22)` переменная `I` может принимать значения 1, 2, 3. Если `I = 1`, то управление передается оператору, помеченному меткой `10`, если `I = 2`, то оператору с меткой `7`, и, наконец, если `I = 3`, — оператору с меткой `22`. В большинстве конкретных реализаций Фортрана в случае, если значение переменной `I ≤ 1` или `I ≥ k`, выполняется оператор, следующий за `GO TO`

Г

ГАРНИТУРА [font typeface, typeface, type family]. То же, что *гарнитура шрифта*

ГАРНИТУРА ШРИФТА, гарнитура [font typeface, typeface, type family]. Полиграфический термин, характеризующий стилистические особенности изображения символов шрифта. Шрифты разных гарнитур могут различаться, например, такими особенностями: наличие или отсутствие засечек на концах линий, постоянная или переменная толщина линий, соотношение между высотой и шириной символов, постоянная или переменная ширина символов и др. Каждая гарнитура имеет собственное имя, например, Times New Roman. Шрифты различных гарнитур существенно отличаются друг от друга по внешнему виду (рис. Г.1). В одну гарнитуру может входить несколько различных начертаний (см. рис. Н.1). См. *начертание шрифта, размер шрифта*

Этот текст набран в гарнитуре Times New Roman

Этот текст набран в гарнитуре Arial

Этот текст набран в гарнитуре Courier New

Рис. Г.1. Гарнитуры Times New Roman, Arial и Courier New

ГБ [GB]. То же, что *гигабайт*

ГБАЙТ [Gbyte]. То же, что *гигабайт*

ГЕНЕРАТОР СЛУЧАЙНЫХ ЧИСЕЛ [random number generator]. *Программа*, вырабатывающая последовательность случайных чисел. Так как в основе получения чисел лежит арифметический алгоритм, то, строго говоря, эти числа являются псевдослучайными. Тем не менее современные алгоритмы обеспечивают хорошее качество последовательности псевдослучайных чисел до 10^5 членов и выше. В приложениях Windows наиболее употребительны Г. с. ч., вырабатывающие числа, равномерно распределенные на отрезке [0; 1]. Например, такой Г. с. ч. есть в Microsoft Excel

ГЕНЕРАЦИЯ ОПЕРАЦИОННОЙ СИСТЕМЫ, конфигурирование операционной системы [OS generation]. Процесс формирования на базе полной *операционной системы* ее рабочей версии, которая по составу и структуре соответствует *конфигурации компьютера* и адаптирована к потребностям пользователя. Г. о. с. выполняется при *загрузке операционной системы*. Для *операционной системы MS-DOS* Г. о. с. выполняется по указаниям, заданным в *файле конфигурации* config.sys, содержимое которого может корректироваться пользователем или оператором *вычислительной системы*. *Операционная*

система Windows использует для генерации информацию в файле win.ini и в *реестре Windows*

ГЕНЕРАЦИЯ СИСТЕМЫ [system generation]. Процесс формирования на базе полной системы ее рабочей версии, которая по составу и структуре соответствует *конфигурации компьютера*, а также имеет программные параметры, настроенные согласно типу имеющейся *операционной системы*, классам решаемых задач и возможным режимам работы. См. *генерация операционной системы*

ГЕОМЕТРИЯ [geometry]. То же, что *геометрия устройства*

ГЕОМЕТРИЯ ДИСКА [disk geometry]. Набор таких показателей, как количество секторов на одной дорожке, количество головок (или сторон) на пластинах жесткого диска, а также общее количество цилиндров на диске

ГЕОМЕТРИЯ УСТРОЙСТВА, **геометрия [device geometry**, **geometry]**. Набор характеристик, определяющих конструктивную организацию устройства. Например, см. *геометрия диска*

ГЕРЦ (Гц) [hertz (Hz)]. Единица измерения частоты. Один Гц соответствует одному колебанию в секунду

ГЕТЕРОГЕННАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА, разнородная вычислительная система [heterogeneous computer system]. *Вычислительная система (вычислительный кластер)* на основе компьютерной сети, состоящей из разнородных *вычислительных узлов*. Принято выделять пять видов разнородности: 1) по архитектурам вычислительных узлов; 2) по формату представления данных на разных вычислительных узлах; 3) по быстродействию (прежде всего по производительности *центральных процессоров*); 4) по загрузке узлов; 5) по загрузке сети. В Г. в. с. могут работать компьютеры различных видов, на которых установлены разные *операционные системы* и *сетевые карты*, а, кроме того, на разных участках сети могут использоваться разные *сетевые протоколы*

ГИБКИЙ МАГНИТНЫЙ ДИСК, **дискета [floppy disk**, **diskette]**. *Магнитный диск*, предназначенный для долговременного хранения данных и для переноса данных с одного компьютера на другой. Представляет собой плоскую неразъемную кассету (рис. Г.2), состоящую из защитной оболочки, в которую заключен диск, покрытый с одной или с обеих сторон магнитным составом. В защитной оболочке имеются центральное отверстие для захвата диска приводным механизмом дисковода и радиальная прорезь (в двусторонних Г. м. д. имеются две такие прорези), через которые *головки записи/чтения* получают доступ к диску. Кроме того, на оболочке расположено устройство блокировки записи, с помощью которого можно предотвратить запись информации на диск, установленный в дисковод. В отличие от *жесткого магнитного диска* Г. м. д. вращается в дисководе не постоянно, а

только при записи или чтении. Данные записываются на диске по концентрическим дорожкам, которые делятся на секторы. *Информационная емкость* Г. м. д. зависит от *плотности записи.* Совершенствование методов записи приводит к тому, что емкость Г. м. д. постоянно увеличивается. В настоящее время применяются Г. м. д., имеющие разный диаметр (указывается в дюймах) и разную емкость: двусторонняя дискета размером 5,25" емкостью 360 Кбайт (40 дорожек на каждой из сторон по 9 секторов на каждой дорожке); дискета двойной плотности емкостью 760 Кбайт (80 + 80 дорожек по 9 секторов на каждой) размером 3,5"; дискеты высокой плотности: емкостью 1,2 Мбайт (80 + 80 дорожек по 15 секторов на каждой) размером 5,25" и емкостью 1.44 Мбайт, размером 3,5"; стандартный формат дискеты типа HD (High Density — высокая плотность) — 80 дорожек на каждой из сторон, 18 секторов по 512 байт на дорожке. Уплотненный формат — 82 или 84 дорожки, до 20 секторов по 512 байт, или до 11 секторов по 1024 байта. К Г. м. д. относятся и *диски Бернулли* (накопители Zip и Jaz) — сменные *запоминающие устройства*, емкость которых доходит до 230 Мбайт и выше. См. *форматирование диска*

Рис. Г.2. Дискеты

ГИБРИДНАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА [hybrid computer]. То же, что *аналого-цифровая вычислительная система*

ГИГАБАЙТ, **Гб**, **Гбайт [gigabyte]**. Единица измерения количества информации и *емкости памяти.* 1 Гбайт = 1024 Мбайт = 1 048 576 Кбайт = 1 073 741 824 байта

ГИГАФЛОПС, **Гфлопс [GigaFLOPS**, **GFLOPS]**. Единица измерения вычислительной *производительности компьютера*, равная миллиарду *арифметических операций* с плавающей точкой в секунду. 1 Гфлопс = 10^3 Мфлопс = 10^6 Кфлопс = 10^9 флопс

ГИПЕРМЕДИА [hypermedia]. Компьютерное представление данных различного типа, в котором автоматически поддерживаются смысловые связи между выделенными понятиями, объектами или разделами. Г. аналогично *ги-*

пертексту, но допускает в качестве связываемых частей не только фрагменты текста, а данные любой природы: графические изображения, видеоклипы, звуковые файлы и пр. В последнее время Г. применяется в Интернете. Для работы с Г. компьютер должен быть оборудован соответствующими мультимедийными устройствами. См. *мультимедиа*

ГИПЕРССЫЛКА [hyperlink]. Средство указания смысловой связи фрагмента одного документа с другим документом или его фрагментом. Г. являются характерным средством *гипертекста*. В настоящее время большинство приложений Windows позволяет размещать Г. в своих документах. Например, в приложении MS Word Г. обычно изображается в виде подчеркнутого текста синего цвета. Щелчок мышью на таком тексте вызывает переход к тому документу, на который указывает Г. В сети *Интернет* Г. является основным связующим средством *Всемирной паутины*

ГИПЕРТЕКСТ [hypertext]. 1. Компьютерное представление текста, в котором автоматически поддерживаются смысловые связи между выделенными понятиями, терминами или разделами. 2. Информационная программа, выводящая на экран дисплея Г. и выполняющая переходы по смысловым связям. Г. позволяет с помощью клавиатуры или мыши сразу получить на экране определения или пояснения слов или словосочетаний, выделенных в тексте подсветкой, ссылки на литературные источники и рекомендации по дальнейшему чтению. В форме Г. создаются справочные системы (Help) современных *программных продуктов*. Г. применяется в компьютерных обучающих системах, толковых словарях и т. п. Г. широко используется в Интернете

ГИСТОГРАММА [bar chart, bar graph]. Столбиковая диаграмма — графическое изображение зависимости между величинами, в котором каждая величина изображается в виде вертикально расположенного прямоугольника или прямоугольного параллелепипеда (рис. Г.3). Обычно Г. составляются для наглядного изображения и анализа статистических данных. Современное программное обеспечение позволяет выводить на дисплей или печатать Г., на которых прямоугольники могут отличаться друг от друга цветом, затенением или узором

ГЛАВНАЯ ПРОГРАММА, основная программа [main program, head program]. *Программа*, выполняемая первой и управляющая *вызовом подпрограмм*

ГЛАВНАЯ ФАЙЛОВАЯ ТАБЛИЦА [master file table (MFT)]. *База данных*, в которой *файловая система операционной системы Windows NT* хранит информацию о содержимом тома. Г. ф. т. представляет собой таблицу, строки которой соответствуют файлам тома, а столбцы — атрибутам файлов

ГЛАВНАЯ ЭВМ, хост [host computer, host]. 1. В *многомашинном вычислительном комплексе — ЭВМ*, выполняющая основную *обработку информации* и управляющая работой комплекса. 2. В *компьютерных сетях* — компьютер, занимающийся обслуживанием сети, управлением передачей сообщений и предоставляющий *удаленный доступ* к своим ресурсам

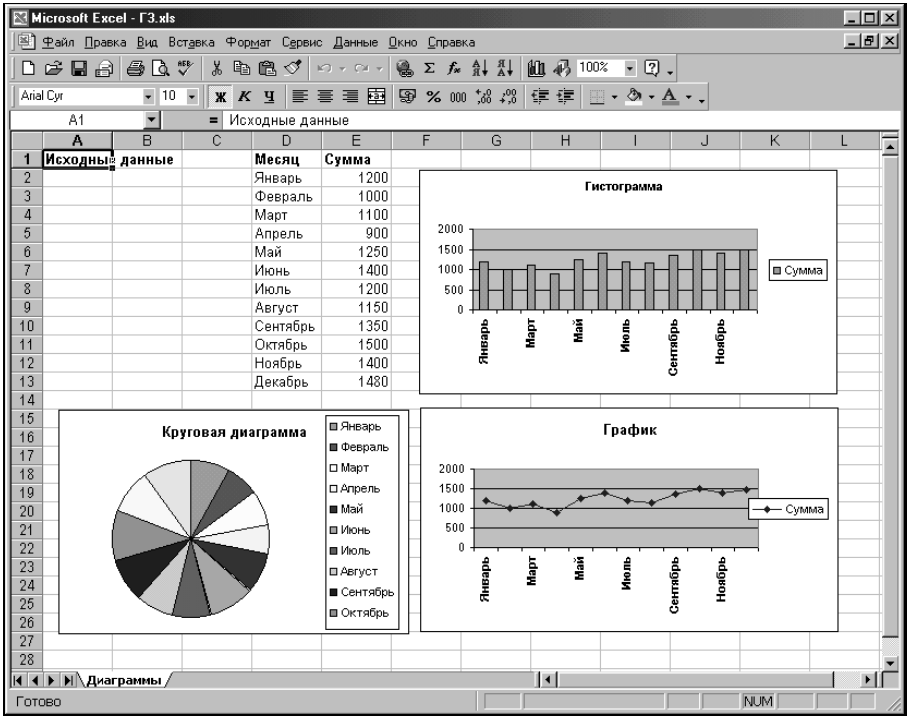

Рис. Г.3. Гистограмма, круговая диаграмма и график в Microsoft Excel

ГЛАВНЫЙ КАТАЛОГ [master directory, root directory]. То же, что *корневой каталог*

ГЛОБАЛЬНАЯ ВЫЧИСЛИТЕЛЬНАЯ СЕТЬ [global network]. *Региональная вычислительная сеть*, покрывающая значительную географическую территорию (регион, страну, ряд стран). *Интернет* является крупнейшей Г. в. с. Противоп. *локальная вычислительная сеть*

ГЛОБАЛЬНАЯ ПЕРЕМЕННАЯ [global variable]. *Переменная*, *областью определения* которой является вся программа или блок с вложенными в него подблоками. Ср. *локальная переменная*

ГЛОБАЛЬНО-УНИКАЛЬНЫЙ ИДЕНТИФИКАТОР [globally unique identifier (GUID)]. Строка, обычно числовая, которая уникальным образом идентифицирует элемент, объект, операцию или человека. Например, 128-битовая величина, которая однозначно определяет объекты при установлении связи между процессами и регистрации компонентов в *операционной системе*

ГЛУБИНА ЦВЕТА [color depth]. То же, что *битовая глубина цвета*

ГОЛОВКА ЗАПИСИ/ЧТЕНИЯ [read-write head]. Узел *запоминающего устройства*, непосредственно взаимодействующий с поверхностью *носителя информации*, записывая или читая данные. Принцип работы и конструкция Г. з./ч. зависит от типа и конструкции запоминающего устройства. В *накопителях на магнитных дисках*, *магнитных картах* или *магнитной ленте* применяются магнитные Г. з./ч., взаимодействующие с *запоминающей средой* посредством магнитного поля, а в *накопителях на лазерных дисках* — оптические, взаимодействующие посредством лазерного луча. Существуют фиксированные Г. з./ч., способные записывать и считывать данные только с одной дорожки носителя, и подвижные, механически передвигаемые от одной дорожки к другой. В *винчестерских дисках* применяются плавающие Г. з./ч. При вращении диска над ним создается тонкий слой плотного газа ("воздушная" подушка), обеспечивающий зависание головок над поверхностью диска на высоте порядка нескольких микрон

ГОЛОСОВАЯ ПОЧТА [voice mail]. Услуга *компьютерной сети* по пересылке *голосовых сообщений*. Г. п. позволяет оставить голосовое сообщение без прямого соединения. Сообщение сохраняется в персональном "ящике голосовой почты". Владелец ящика может удаленно прослушивать эти сообщения и стирать их

ГОЛОСОВОЕ СООБЩЕНИЕ [voice message]. *Сообщение*, содержащее запись человеческой речи в форме *аудиоданных*

ГОЛОСОВОЙ КАНАЛ [media stream]. Часть *линии связи*, отвечающая за передачу звука в телефонном соединении

ГОЛОСОВОЙ МОДЕМ [voice modem]. *Модем,* предоставляющий программный доступ к *голосовому каналу*. Программа, управляющая Г. м., может не только обеспечивать состояние соединения, но и управлять звуком в *линии связи* при передаче *голосовых сообщений*

ГОРИЗОНТАЛЬНОЕ МЕНЮ [horizontal menu]. *Меню* с горизонтальным расположением *пунктов меню* (один рядом с другим). Г. м., как правило, размещается в верхней или нижней части экрана и не перекрывает выве-

денную на него информацию (см. рис. М.5). Противоп. *вертикальное меню*. См. *строка меню*

ГОРЯЧАЯ ОБЛАСТЬ, **горячая точка [hot spot]**. Область на экране, в которой меняется вид *указателя мыши*. Если пользователь в этом месте щелкнет кнопкой мыши, то будет выполнено определенное действие. Чаще всего в Г. о. находится якорь гиперссылки

ГОРЯЧАЯ ТОЧКА [hot spot]. То же, что *горячая область*

ГОРЯЧИЕ КЛАВИШИ [hot keys]. То же, что *клавиши быстрого вызова*

ГРАДАЦИИ СЕРОГО, **шкала серого [gray scale]**. Характеристика яркости изображения, равная количеству одновременно отображаемых оттенков серого цвета, промежуточных между черным и белым. Например, существуют цветовые режимы графических отображающих устройств с 256 Г. с.

ГРАДИЕНТ [gradient]. 1. В *компьютерной графике* — плавный переход от одного цвета к другому. Художники называют такой прием "растяжкой". 2. *Инструмент машинной графики*, выполняющий Г. выделенной контуром области экрана с указанием направления плавного цветового перехода. На рис. Г.4 для *графического редактора* Adobe Photoshop показаны кнопка вызова Г. и диалоговое окно, позволяющее установить параметры инструмента

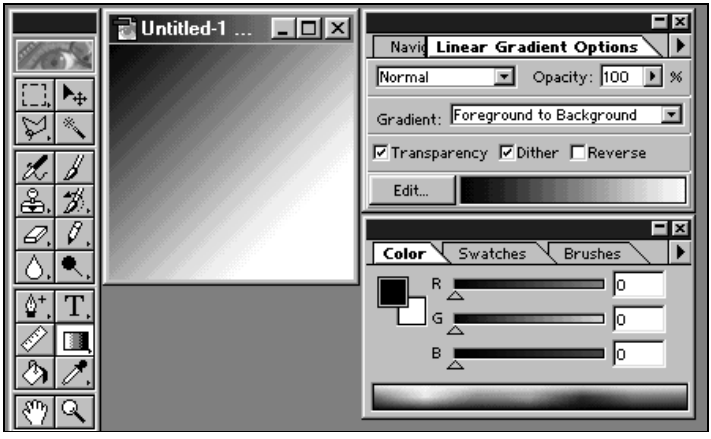

Рис. Г.4. Нажатая кнопка вызова, образец работы и окна инструмента "градиент" графического редактора Adobe Photoshop

ГРАНИЦА ИНДЕКСА [subscript boundary]. *Целое* число, указывающее наименьшее (нижняя граница) или наибольшее (верхняя граница) возможное значение индекса. Задавая в *описании массива* Г. и., мы сообщаем компилятору количество *элементов массива.* Кроме того, Г. и. требуются для автоматической проверки допустимости значений индекса, вычисленных во время исполнения программы. В некоторых *языках программирования* нижняя Г. и. является фиксированной, и в описаниях не указывается. Например, в языке Си значения индексов начинаются с нуля, а в Фортране — с единицы

ГРАНИЦА ОКНА, **обрамление**, **окантовка [border]**. В *операционной системе Windows* — рамка или часть рамки вокруг окна, на которой курсор или указатель мыши изменяют свою форму и, соответственно, выполняемую функцию. Например, при совмещении с вертикальной Г. о. указатель мыши приобретает вид, показывающий, что пользователь может изменить горизонтальный размер окна. См. табл. У.1

ГРАНИЧНАЯ ПАРА [bound pair]. В *языках программирования* — *синтаксическая конструкция*, определяющая нижнюю и верхнюю *границы индекса* массива. С помощью Г. п. в *описании массива* указывается допустимое множество значений индекса. Например, в описании на Паскале

```
var A: array [1..10, n..m ] of real,
```

где `1..10` указывает, что первый индекс массива `A` может принимать целые значения от 1 до 10, а Г. п. `n..m` означает, что нижняя граница второго индекса задана значением переменной `n`, а верхняя — `m`

ГРАФ ДЕЯТЕЛЬНОСТИ [activity graph]. Средство описания поведения в *унифицированном языке моделирования* UML. Особый вид *конечного автомата*, в котором все или бóльшая часть состояний являются состояниями деятельности и в котором все или бóльшая часть переходов запускаются при завершении деятельности в исходных состояниях. На графе деятельности изображается *поток управления.* Г. д. является завершенным блоком модели и отображается на *диаграмме деятельности.* Ср. *блок-схема*

ГРАФИК [chart]. *Диаграмма*, изображающая функциональную зависимость в виде кривой или ломаной линии. См. рис. Г.3

ГРАФИЧЕСКАЯ ЗАГОТОВКА [clip art]. То же, что *аппликация*

ГРАФИЧЕСКАЯ ИНФОРМАЦИЯ [graphic information, **graphic data]**. *Информация*, представленная в форме изображения — схем, графиков, диаграмм, рисунков, фотографий и т. п.

ГРАФИЧЕСКАЯ НОТАЦИЯ [graphical notation]. *Нотация формального языка*, в которой применяются графические обозначения: фигуры, значки, линии. Например, *блок-схемы программ* и *унифицированный язык моделирования* используют Г. н.

ГРАФИЧЕСКАЯ ПОДСИСТЕМА [graphic subsystem]. Часть *операционной системы*, обеспечивающая решение задач *компьютерной графики*. Сюда входят *программные средства* ввода, обработки и отображения *графической информации*, а также преобразования данных в графическую форму. Например, пакет OpenGL для Windows. Ср. *графический редактор*

ГРАФИЧЕСКИЕ ИНСТРУМЕНТЫ [graphics tools]. То же, что *инструменты машинной графики*

ГРАФИЧЕСКИЙ АДАПТЕР [graphics adapter]. *Адаптер дисплея*, поддерживающий *графический режим*. Устаревший термин, т. к. почти все современные видеоадаптеры являются графическими

ГРАФИЧЕСКИЙ АКСЕЛЕРАТОР [graphics accelerator]. То же, что *графический сопроцессор*

ГРАФИЧЕСКИЙ ДИАЛОГ [graphic dialog]. *Диалог*, при котором происходит обмен *графической информацией* между *пользователем* и *компьютером*

ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ [graphic user interface (GUI)]. *Интерфейс пользователя*, основанный на средствах *машинной графики*. Г. и. п. предполагает взаимодействие человека с компьютером в форме диалога с использованием ввода и вывода на экран дисплея *графической информации*, управления программами с помощью *кнопок*, меню, окон, *экранных панелей* и других *элементов управления*, а также выделением цветом определенных частей изображения или текста и т. п. В Г. и. п. используются мышь, *световое перо*, *цифровой планшет графического ввода* и другие средства ведения диалога с компьютером и формирования графических изображений

ГРАФИЧЕСКИЙ ОБЪЕКТ [graphics object]. Модель объекта реального мира, представленная в виде графического изображения

ГРАФИЧЕСКИЙ ПЛАНШЕТ, **цифровой планшет графического ввода**, **цифровой планшет [graphics tablet**, **digitizing tablet]**. *Указательное устройство* в виде соединенного с компьютером планшета, водя по которому специальной указкой, обычно называемой *пером*, можно управлять курсором или создавать контурные рисунки, которые воспроизводятся на экране дисплея в окне *графического редактора*. Далее эти рисунки можно обрабатывать с помощью графического редактора (например, окрашивать, перемещать, до-

полнять и т. д.). Поэтому Г. п. считается также *устройством ввода графической информации* (разновидностью дигитайзера). Г. п. при управлении курсором и исполнении рисунков полностью заменяет мышь, однако в отличие от нее является абсолютным указательным устройством, т. к. вводит в компьютер абсолютные координаты линий, по которым движется перо. Положив на планшет какой-либо рисунок и обводя его контуры, можно копировать на экран фрагменты этого рисунка. См. *перо*

ГРАФИЧЕСКИЙ ПРИМИТИВ, **примитив [graphics primitive, primitive]**. Элемент графического изображения, который рисуется и обрабатывается компьютером как одно целое. Например, точка, дуга, прямоугольник, буква и т. п. В *системах автоматизированного проектирования* можно по собственному усмотрению создавать новые Г. п.

ГРАФИЧЕСКИЙ РЕДАКТОР, **редактор изображений [graphics editor, bitmap editor]**. Программа создания и редактирования *графической информации.*

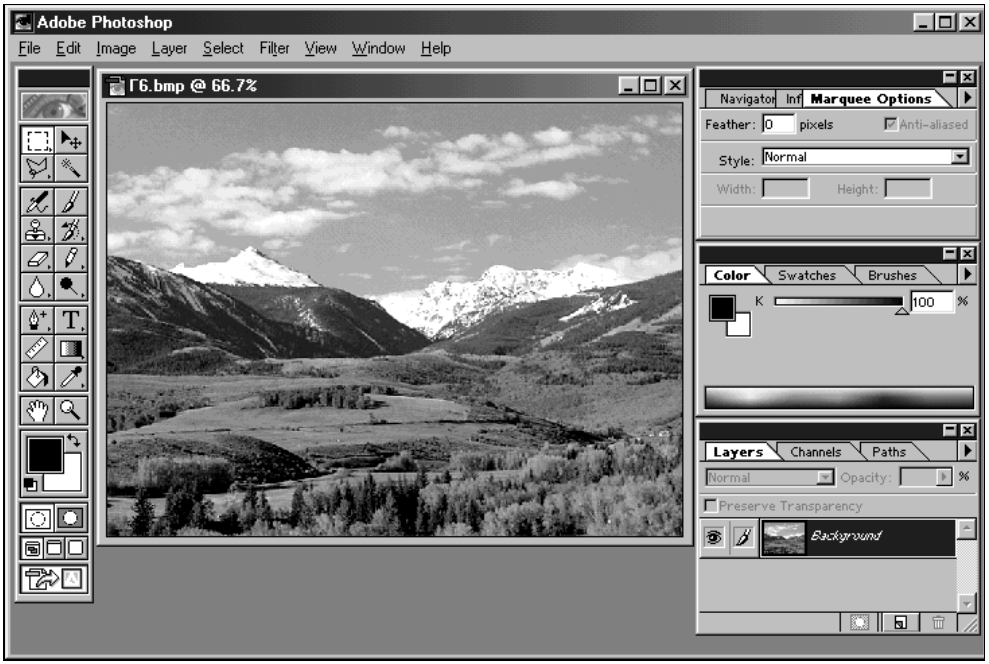

Рис. Г.5. Рабочее окно графического редактора Adobe Photoshop

Современные Г. р. позволяют создавать реалистические изображения, работать с цветными сканированными, а также цифровыми фото- и видеоизоб-

ражениями. Г. р. располагает широким набором *инструментов машинной графики*, с помощью которых можно подвергать эти изображения трансформации, цветовой коррекции, коллажированию и т. п. На рис. Г.5 показано рабочее окно Г. р. Adobe Photoshop

ГРАФИЧЕСКИЙ РЕЖИМ [graphics mode]. Режим работы *адаптера дисплея*, обеспечивающий вывод на экран *графической информации*. В Г. р. экран представляет собой растр — множество точек (пикселов), упорядоченных в строки и столбцы, а изображение формируется путем придания своего цвета и яркости каждой точке растра. Работающая программа определяет нужные цвет и яркость каждого пиксела. Эти данные заносятся в видеопамять *центральным процессором*, а видеоадаптер считывает их и последовательно, переходя от точки к точке по строке и от строки к строке, "рисует" изображение на экране. Ср. с *текстовым режимом*, при котором видеоадаптер одновременно "рисует" символ, занимающий целое знакоместо. Качество изображения, которое позволяет получить Г. р., определяется его *разрешающей способностью* и количеством используемых цветов. Разрешающая способность выражается произведением числа столбцов на число строк растра. Стандартные Г. р. видеоадаптеров, определенные фирмой IBM, обеспечивают разрешающую способность от 320×200 до 720×480 и выше при количестве изображаемых цветов от 16 до 256. Разрешение адаптеров специальных графических систем может быть 1024×768 и выше при количестве изображаемых цветов, доходящем до 16 млн. На монохромном дисплее различные цвета изображаются градациями серого, зеленого или коричневого цвета. См. *адаптер дисплея*

ГРАФИЧЕСКИЙ СИМВОЛ [graphic character]. Этим термином называют все допустимые *вычислительной системой* изображаемые на экране *символы*, кроме *псевдографических символов*. Например, любая буква или цифра. Ср. *управляющий символ*

ГРАФИЧЕСКИЙ СОПРОЦЕССОР, графический акселератор [graphics coprocessor, graphics accelerator]. *Сопроцессор*, устанавливаемый на плате видеоадаптера с целью повышения скорости вывода на экран графического изображения. Часть изображения может создаваться Г. с. без участия *центрального процессора*. Для этого на Г. с. посылаются специальные команды. Г. с. берет на себя выполнение операций перемещения фрагментов растрового изображения, рисования *графических примитивов*, *закраски* и *заливки* графических объектов и т. п. В результате разгружается центральный процессор и сокращается количество информации, передаваемой по *системной шине*. Все это повышает быстродействие видеоподсистемы компьютера

ГРАФИЧЕСКИЙ ФАЙЛ [graphic file]. *Файл*, содержащий графическое изображение. Г. ф. обычно создается с помощью *графического редактора*. Разновидностью Г. ф. являются *растровые файлы*. Используются различные форматы Г. ф., которые отличаются алгоритмами *сжатия данных* и другими параметрами. Например, на рис. Г.6 приведено графическое изображение, а в табл. Г.1 указаны размеры Г. ф. для этого изображения в разных форматах. См. *формат BMP, формат PCX, формат TIFF, формат GIF, формат JPEG*

Таблица Г.1. *Сравнительные размеры графических файлов в разных форматах*

Формат	BMP	PCX	TIFF	GIF	JPEG
Размер файла, Кбайт	901	929	688	178	155

Рис. Г.6. Пример растрового изображения

ГРАФОПОСТРОИТЕЛЬ, **плоттер [plotter]**. Устройство, предназначенное для изображения на бумаге или другом подобном ей носителе выводимых из компьютера данных в форме чертежа, графика или рисунка. Г. используются для выполнения машиностроительных и строительных чертежей, метеорологических карт, разного рода схем, для вывода результатов вычислений в

форме графиков или диаграмм. Изображение формируется пишущими элементами (перьями) различной толщины и цвета. Различают Г. планшетного и рулонного типа (рис. Г.7). Они работают в прямоугольной системе координат. В Г. планшетного типа лист бумаги неподвижно закреплен на планшете, а перо может передвигаться к любой точке, лежащей в пределах установленных диапазонов изменения координат *x* и *y*, касаясь поверхности бумаги и вычерчивая линию, или может двигаться над поверхностью в поднятом состоянии. В Г. рулонного типа двигаются и бумага, и перо

Рис. Г.7. Рулонные графопостроители

ГРУППА [group]. 1. Множество приложений в окне Диспетчера программ *операционной системы Windows* (Program Manager). 2. В *компьютерной графике* — временное объединение (фиксация) объектов, позволяющее применять всевозможные операции (например, перемещение, вращение и другие трансформации) к Г. как единому целому. На рис. Г.8 показана выделенная Г. графических объектов (видны маркеры, выделяющие границу Г.). 3. Совокупность пользователей или компьютеров. Г. могут использоваться для управления доступом или в качестве списков рассылки в *электронной почте*

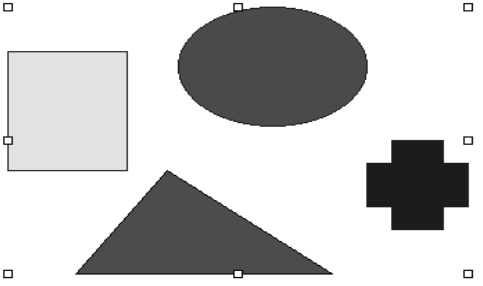

Рис. Г.8. Группа графических объектов

ГРУППА ОБЪЕДИНЕННЫХ ЭКСПЕРТОВ ПО ФОТОГРАФИИ, **группа JPEG [Joint Pictures Entertainment Group (JPEG)]**. Рабочий комитет *Международной организации по стандартизации* (ISO), разрабатывающий стандарты сжатия и восстановления сжатых данных в компьютерной графике. См. *формат JPEG*

ГРУППА ЭКСПЕРТОВ В ОБЛАСТИ ДВИЖУЩИХСЯ ИЗОБРАЖЕНИЙ [Moving Pictures Expert Group (MPEG)]. То же, что *группа MPEG*

ГРУППА JPEG [Joint Pictures Entertainment Group (JPEG)]. То же, что *группа объединенных экспертов по фотографии*

ГРУППА MPEG, **группа экспертов в области движущихся изображений**, **экспертная группа по кинематографии [Moving Pictures Expert Group (MPEG)]**. Рабочая группа экспертов цифрового видео, регулярно проводящая встречи своих специалистов под эгидой *Международной организации по стандартизации (ISO)*. Г. MPEG занимается разработкой эффективных методов хранения видеоданных. Например, при использовании *стандарта MPEG* на стандартный компакт-диск можно записать полнометражный фильм, причем качество изображения и звука будет высоким

ГФЛОПС [GFLOPS]. То же, что *гигафлопс*

Гц [Hz]. То же, что *герц*

Д

ДАМП [dump]. 1. Распечатка содержимого памяти компьютера или файла, обычно без учета внутренней структуры данных. Копия содержимого регистров, нужного участка оперативной памяти, блока данных или файла выводится на печать последовательно, байт за байтом, в форме *двоичного, восьмеричного* или *шестнадцатеричного* кодов. Д. применяется для анализа работы программных систем. Например, Д., полученный после *аварийного завершения* программы, служит материалом для выявления причин аварийного завершения. 2. То же, что *дамп памяти*. 3. То же, что *дамп экрана*

ДАМП ПАМЯТИ, **дамп [memory dump**, **dump]**. Копия содержимого *оперативной памяти* на внешнем носителе (бумаге, *магнитном диске* и т. п.)

ДАМП ЭКРАНА, **дамп [screen dump**, **dump]**. Копия содержимого экрана на внешнем носителе (бумаге, *магнитном диске* и т. п.)

ДАННЫЕ [data]. 1. Информация, подготовленная для передачи, хранения и обработки в *вычислительной машине*, т. е. представленная в символьной (цифровой) форме. Примерами Д. являются закодированные для ввода или

уже введенные в компьютер текст, речь, изображение, таблицы всевозможных величин и т. п. 2. В контексте отдельной программы или пакета программ слово "Д." означает все обрабатываемые программой объекты, отличные от ее команд. Эти Д. можно разбить на два класса: Д., определяемые программистом, и Д., определяемые системой. Д., определяемые программистом, состоят из элементов, которые программист явно определяет и над которыми производит действия в своей программе, например, числа, массивы, файлы и т. д. Д., определяемые системой, состоят из тех элементов, которые формируются для служебных целей во время выполнения программы, например, стеки *точек возврата*, дескрипторы структур данных, списки свободного пространства памяти, буферы ввода/вывода и т. д. Эти Д. обычно генерируются автоматически без явного указания программиста (часто он об этом даже не подозревает). 3. В качестве Д. могут выступать команды и даже программы. Например, коды программ на *исходном языке системы программирования* являются Д. для компилятора, а коды результирующей *объектной программы* — Д. для компоновщика. См. *видеоданные*

ДВИЖОК [scroll box]. То же, что *бегунок*

ДВОИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ [binary number system]. *Позиционная система счисления* с основанием $q = 2$. В качестве двоичных цифр используются символы 0 и 1. Например, двоичное число

$$(1101.01)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = (13.25)_{10}$$

Представление числа в Д. с. с. рассматривается как *двоичный код* этого числа. Поэтому Д. с. с. широко применяют для внутреннего представления чисел в компьютере. См. *система счисления*

ДВОИЧНОЕ ПРЕДСТАВЛЕНИЕ [binary-coded representation]. Представление информации в *двоичном коде*. Д. п. является основным способом представления данных в памяти компьютера

ДВОИЧНОЕ ЧИСЛО [binary number]. *Число*, представленное в *двоичной системе счисления*

ДВОИЧНЫЙ КОД [binary code]. *Код*, символами которого являются 0 и 1

ДВОИЧНЫЙ РАЗРЯД [binary digit, bit]. Место (позиция), занимаемое символом 0 или 1 в *двоичном коде*

ДВОЙНАЯ ОШИБКА [double fault]. Состояние *вычислительной системы*, возникающее, когда одновременно имеют место два прерывания и процессор не в состоянии их обработать. Этот вид ошибки практически всегда вызывается *аппаратным сбоем*

ДВОЙНОЕ СЛОВО [double word]. *Элемент данных*, состоящий из двух смежных *машинных слов*, который воспринимается памятью, *арифметическим устройством* или *устройством управления* ЭВМ как единое целое, имеющее определенное смысловое содержание. Например, *числа с плавающей точкой* одинарной и двойной точности *математический сопроцессор персонального компьютера* различает как слово (занимает 4 байта) и Д. с. (занимает 8 байт) соответственно

ДВОЙНОЙ ЩЕЛЧОК [double click]. Действие, которое заключается том, чтобы дважды быстро нажать и отпустить кнопку неподвижной мыши. Д. щ. применяется при выборе и активизации объектов, *запуске программ* и т. п. При этом *указатель мыши* должен быть позиционирован на выбираемый объект или программу. Ср. *щелчок*

ДВУМЕРНЫЙ МАССИВ [two-dimensional array]. Матрица. Массив с двумя измерениями (строками и столбцами)

ДВУМЕСТНАЯ ОПЕРАЦИЯ, **бинарная операция [dyadic operation**, **binary operation]**. *Арифметическая*, *логическая* или другая *операция* над двумя операндами (аргументами). Например, операция "логическая сумма" ("или") — двуместная. Ср. *одноместная операция*

ДВУХКАНАЛЬНЫЙ КОНТРОЛЛЕР [dual channel controller]. Схема или устройство, управляющее передачей сигнала по двум магистралям

ДВУХТОЧЕЧНЫЙ ОБМЕН [point-to-point exchange, point-to-point passing]. В *модели передачи сообщений* форма обмена сообщениями, в которой участвуют только два процесса, — процесс-отправитель и процесс-получатель. Имеется несколько разновидностей Д. о.: синхронный обмен, который сопровождается уведомлением об окончании приема сообщения; асинхронный обмен, который таким уведомлением не сопровождается; блокирующие прием—передача, которые приостанавливают выполнение процесса на время приема сообщения; неблокирующие прием—передача, при которых выполнение процесса продолжается, а программа в нужный момент может запросить подтверждение завершения приема сообщения. При реализации Д. о. обычно гарантируется сохранение порядка сообщений, которые не могут обгонять друг друга. Ср. *коллективный обмен*

ДЕЙСТВИЕ [action]. Выполнимое всегда завершающееся атомарное вычисление, которое приводит к изменению состояния системы или возврату значения. Выполнение Д. не может быть прервано событием. Д. является одним из основных понятий *унифицированного языка моделирования* UML. Ср. *деятельность*

ДЕЙСТВИЕ ПРИ ВХОДЕ [entry action]. Действие, осуществляемое при входе в состояние *конечного автомата*

ДЕЙСТВИТЕЛЬНАЯ ПЕРЕМЕННАЯ [real variable]. То же, что *вещественная переменная*

ДЕЙСТВИТЕЛЬНОЕ ЧИСЛО [real number]. То же, что *вещественное число*

ДЕЙСТВИТЕЛЬНЫЙ АДРЕС [effective address, executive address]. То же, что *исполнительный адрес*

ДЕЙСТВИТЕЛЬНЫЙ ТИП [real type]. То же, что *вещественный тип*

ДЕЙСТВУЮЩЕЕ ЛИЦО [actor]. Абстрактное описание сущности, находящейся вне моделируемой системы, которая напрямую взаимодействует с системой. Д. л. участвует в *варианте использования* или множестве вариантов использования с целью достижения определенного результата. Д. л. является одним из основных понятий *унифицированного языка моделирования* UML и отображается на *диаграммах использования*

ДЕЙТАГРАММА [datagram]. В *сети с коммутацией пакетов — пакет данных*, содержащий помимо передаваемой информации полный адрес пункта назначения, заданный пользователем, а не сетью. Д. передаются по сети независимо от других пакетов и без установки *виртуального соединения*

ДЕЛЕГИРОВАНИЕ [delegation]. Перепоручение выполнения функции одного объекта другому. Например, Д. операции, которую осуществляют над первым объектом, вызывает операцию над вторым объектом, которая и делает работу. Д. можно использовать в качестве альтернативы наследованию. В некоторых языках (таких как Self) Д. поддерживается механизмами наследования в самом языке. В большинстве остальных языков (например, Си++ и Smalltalk) Д. реализуется в виде *ассоциации* или *агрегации* с другим объектом

ДЕКОДЕР [decoder]. То же, что *дешифратор*

ДЕЛЕНИЕ НА НУЛЬ [divide by zero error]. То же, что *ошибка деления на нуль*

ДЕМОН [daemon]. Программа, используемая, как правило, в *операционных системах UNIX* для выполнения служебной функции без запроса со стороны пользователя и даже без его ведома. См. например, *демон линейного принтера*. Д. пребывает в *фоновом режиме* и активизируется только в случае необходимости

ДЕМОН ЛИНЕЙНОГО ПРИНТЕРА [line printer daemon (LPD)]. *Обслуживающая программа (демон)* сервера печати, позволяющая *линейному принтеру* принимать задания на печать от удаленных клиентов, используя *протокол LPR*. Удаленные клиенты часто представляют собой UNIX-системы, но существует программное обеспечение и для большинства других *операционных систем*, включая *операционную систему Windows NT*

ДЕМОНСТРАЦИОННАЯ ПРОГРАММА [demonstration program (demo program)]. *Программа*, демонстрирующая *интерфейс пользователя* с *программным*

продуктом либо возможности программного продукта. Как правило, это специально созданная программа-имитатор, иллюстрирующая некоторые экранные изображения, являющиеся результатом работы программного продукта. Иногда это бывает "урезанная" версия самого программного продукта

ДЕРЕВО ДОМЕНОВ [domain tree]. Иерархическая организация *доменов*. При этом домены прозрачно связываются друг с другом двусторонними, транзитивными *доверительными отношениями*, т. е. новый домен в дереве сразу же вступает в доверительные отношения с другими доменами, и все объекты доменов становятся доступными друг для друга. Таким образом, пользователю или группе в любом домене можно дать разрешения на доступ к любому объекту в дереве доменов, благодаря чему и обеспечивается возможность единой регистрации в сети

ДЕРЕВО КАТАЛОГОВ [directory tree, tree]. Графическое изображение структуры взаимного вложения *каталогов* и *подкаталогов*. Д. к. позволяет наглядно представить структуру *корневого каталога* и всех его подкаталогов, а также быстро найти нужный каталог и файл. На рис. Д.1 показан вид окна программы *Проводник Windows* на экране *персонального компьютера*.

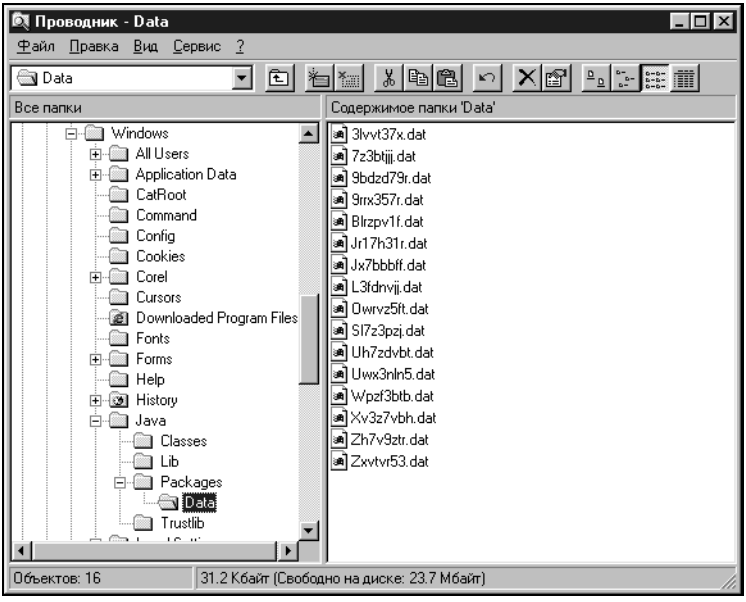

Рис. Д.1. Дерево каталогов в программе Проводник

На левой панели изображено Д. к. диска D:. Значок + слева от имени каталога означает, что этот каталог содержит вложенные каталоги, которые не

отображены в дереве в данный момент. Щелчок мышью по значку + раскрывает каталог. Имя *рабочего каталога* выделено на Д. к. *цветовым маркером.* Передвигая маркер по дереву, можно изменить рабочий каталог. При этом содержимое каталога появится на соседней панели

ДЕСКРИПТОР, **описатель [descriptor]**. 1. Определяемая и создаваемая *операционной системой* или *системой программирования* и хранимая в памяти служебная информация о программе, файле, данных или других объектах. Д. является "паспортом" объекта и содержит данные, необходимые системе для работы с указанными объектами. Например, Д. массива содержит тип элементов массива, адрес начального элемента массива, *размерность массива*, границы и шаг изменения индексов. Д. сегмента переменной длины обязательно содержит длину сегмента. 2. Слово, словосочетание или код языка информационно-поисковых систем, используемые для описания основного содержания документа. См. *информационная система* 3. В *унифицированном языке моделирования* UML *элемент модели*, описывающий общие свойства некоего множества экземпляров, включая их структуру, отношения, поведение, ограничения, цель и т. д.

ДЕСПУЛИНГ [despooling]. Процесс считывания содержимого *буферного файла* и отправки его на *печатающее устройство.* См. *буферная память, спулинг*

ДЕСТРУКТОР [destructor]. Операция с областью действия в масштабе класса, которая уничтожает экземпляр класса и высвобождает занимаемые им ресурсы

ДЕСЯТИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ [decimal number system]. *Позиционная система счисления* с основанием $q = 10$. Привычная нам система записи чисел, в которой используются цифры 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. В *языках программирования высокого уровня* Д. с. с. применяется для записи числовых констант

ДЕСЯТИЧНОЕ ПРЕДСТАВЛЕНИЕ [decimal representation]. Представление чисел в *десятичной системе счисления.* Д. п. применяется для записи числовых констант в программах на *языках программирования высокого уровня* и вводимых с клавиатуры числовых *исходных данных*

ДЕФРАГМЕНТАТОР ДИСКА, **дефрагментор диска [disk defragmenter]**. *Приложение*, выполняющее *дефрагментацию диска*

ДЕФРАГМЕНТАЦИЯ [defragmentation]. Перераспределение данных на устройстве длительного хранения информации (например, *гибком* или *жестком диске*) с целью устранения пустых мест между фрагментами, заполненными данными. Обычно файлы записываются на любое свободное место на диске. Поэтому частое удаление и создание файлов может привести к тому, что многие файлы окажутся фрагментированы (т. е. записаны несколькими порциями в разных местах диска). Д. выполняется с помощью специальных программ-дефрагментаторов путем последовательного копирования храня-

щихся на диске файлов в *оперативную память* или на другой *носитель данных* и повторного размещения этих файлов в последовательных (соседних) секторах, без пропусков. В результате, как правило, ускоряется доступ к данным — значительно возрастает скорость открытия файлов и запуска приложений, поскольку не тратится время на поиск фрагментов файла

ДЕФРАГМЕНТОР ДИСКА [disk defragmenter]. То же, что *дефрагментатор диска*

ДЕШИФРАТОР, **декодер [decoder]**. 1. Устройство или программа, преобразующая зашифрованные (закодированные) данные в их первоначальный вид. Противоп. *шифратор.* 2. Устройство, преобразующее *цифровой код* в управляющие сигналы. Например, имеющийся в *центральном процессоре* Д. команд распознает содержание команды и посылает соответствующие управляющие сигналы другим устройствам компьютера, обеспечивая ее выполнение. В этом смысле Д. является *цифроаналоговый преобразователь*

ДЕЯТЕЛЬНОСТЬ [activity]. Неатомарное вычисление, выполнение которого может быть прервано событием. Д. является одним из основных понятий *унифицированного языка моделирования* UML. Ср. *действие*

ДЖАВА [Java]. См. *язык программирования Java*

ДЖАВА-АППЛЕТ, JAVA-апплет [Java applet]. *Класс языка программирования Java*, встроенный в виде исполняемого модуля в документ, набранный на *языке HTML.* Д.-а. загружается с сервера на компьютер-клиент при запросе содержащей его страницы как *прикрепленный файл.* Д.-а. отличается от Java-приложения тем, что имеет ограниченный доступ к некоторым ресурсам компьютера-клиента и запрет на *обмен данными* с другими компьютерами сети (за исключением сервера, с которого он загружен). Д.-а. применяются, например, при организации на *веб-страницах* вычислений или диалога с пользователем, а также для применения мультимедийных эффектов в оформлении веб-страниц. См. *апплет*

ДЖАВА-СЦЕНАРИЙ, JAVA-сценарий [Java script]. *Сценарий* в виде текста на *языке программирования Java* или *языке сценариев JavaScript*, включаемый непосредственно в *веб-страницу* и загружаемый вместе с ней. Исполняется интерпретатором Java, который встраивается практически в каждый современный браузер

ДЖОЙСТИК [joystick]. Рычажковое *устройство ввода*, предназначенное для управления движением графических объектов на экране дисплея. Д. обычно выполняется в виде снабженной кнопками рукоятки, которой можно совершать наклонные и вращательные движения (рис. Д.2). Эти движения преобразуются в управляющие электрические сигналы, пропорциональные углам отклонения рукоятки от вертикали. Нажатием кнопок можно сообщить работающей программе некоторые сведения или отдать какие-либо команды. Д. различной конструкции широко применяются в компьютерных играх и тренажерах

Рис. Д.2. Джойстик

ДЖОКЕР [joker]. Идентификатор, обозначающий произвольное значение параметра

ДИАГРАММА 1. **[chart]**. Графическое изображение зависимости между величинами. На Д. числовые величины или их соотношения изображаются с помощью плоских или объемных геометрических фигур. На рис. Д.3 изображено диалоговое окно **Мастер диаграмм**, с помощью которого можно представлять в графической форме данные в Microsoft Excel.

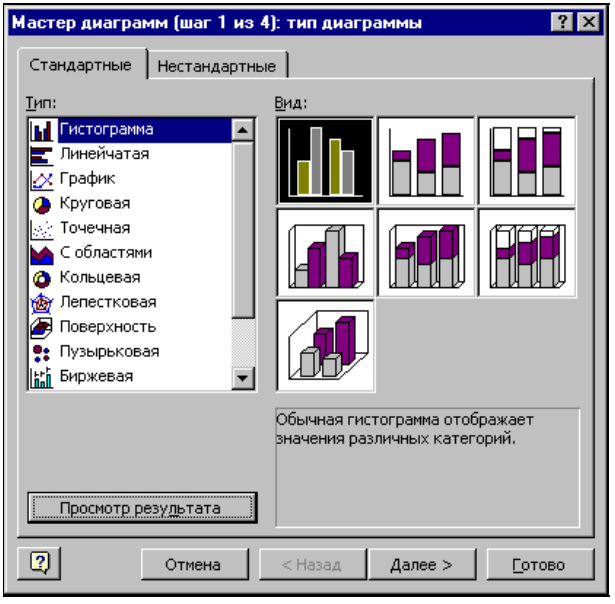

Рис. Д.3. Типы диаграмм Microsoft Excel

Оно предоставляет пользователю выбор из 11 типов диаграмм. Для каждого типа существует несколько разновидностей. 2. **[diagram]**. Графическая презентация элементов модели в *визуальном языке моделирования*. Изображается, как правило, в виде графа, состоящего из дуг (отношений) и вершин (сущностей). В *унифицированном языке моделирования* UML существуют *диаграммы классов*, *использования*, *последовательности*, *кооперации*, *состояний*, *деятельности*, *компонентов, размещения* и некоторые другие

ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ [use case diagram]. То же, что *диаграмма использования*

ДИАГРАММА ВЗАИМОДЕЙСТВИЯ [interaction diagram]. Общий термин *унифицированного языка моделирования* UML, который применяется к нескольким видам диаграмм, касающихся взаимодействия объектов. К Д. в. относятся *диаграммы кооперации* и *диаграммы последовательности*. С ними тесно связаны *диаграммы деятельности*

ДИАГРАММА ДЕЯТЕЛЬНОСТИ [activity diagram]. Диаграмма *унифицированного языка моделирования* UML, на которой изображается *граф деятельности* (рис. Д.4)

ДИАГРАММА ИСПОЛЬЗОВАНИЯ, диаграмма вариантов использования [use case diagram]. Диаграмма *унифицированного языка моделирования* UML, на которой изображены отношения, существующие между *действующими лицами* и *вариантами использования* системы (рис. Д.5)

ДИАГРАММА КЛАССОВ [class diagram]. Диаграмма *унифицированного языка моделирования* UML, в которой собраны декларативные (статичные) элементы, такие как классы, типы, а также их содержимое и отношения между ними (рис. Д.6)

ДИАГРАММА КОМПОНЕНТОВ [component diagram]. Диаграмма *унифицированного языка моделирования* UML, на которой изображены *компоненты программы* и зависимости между ними (рис. Д.7)

ДИАГРАММА КООПЕРАЦИИ [collaboration diagram]. Диаграмма *унифицированного языка моделирования* UML, на которой показано взаимодействие объектов в *кооперации*. В отличие от *диаграммы последовательности*, в Д. к. связи между объектами указываются явно (рис. Д.8). С другой стороны, на Д. к. время не изображается графически, поэтому последовательность сообщений нужно определять, исходя из их нумерации. Таким образом, на диаграммах последовательности и Д. к. изображается одна и та же информация, однако делается это по-разному

ДИАГРАММА ПОТОКОВ ДАННЫХ [data flow diagram (DFD)]. Диаграмма, отражающая перемещение данных между пользователями приложения, хранилищами данных и процедурами обработки данных. В *унифицированный язык моделирования* UML Д. п. д. в явном виде не включены

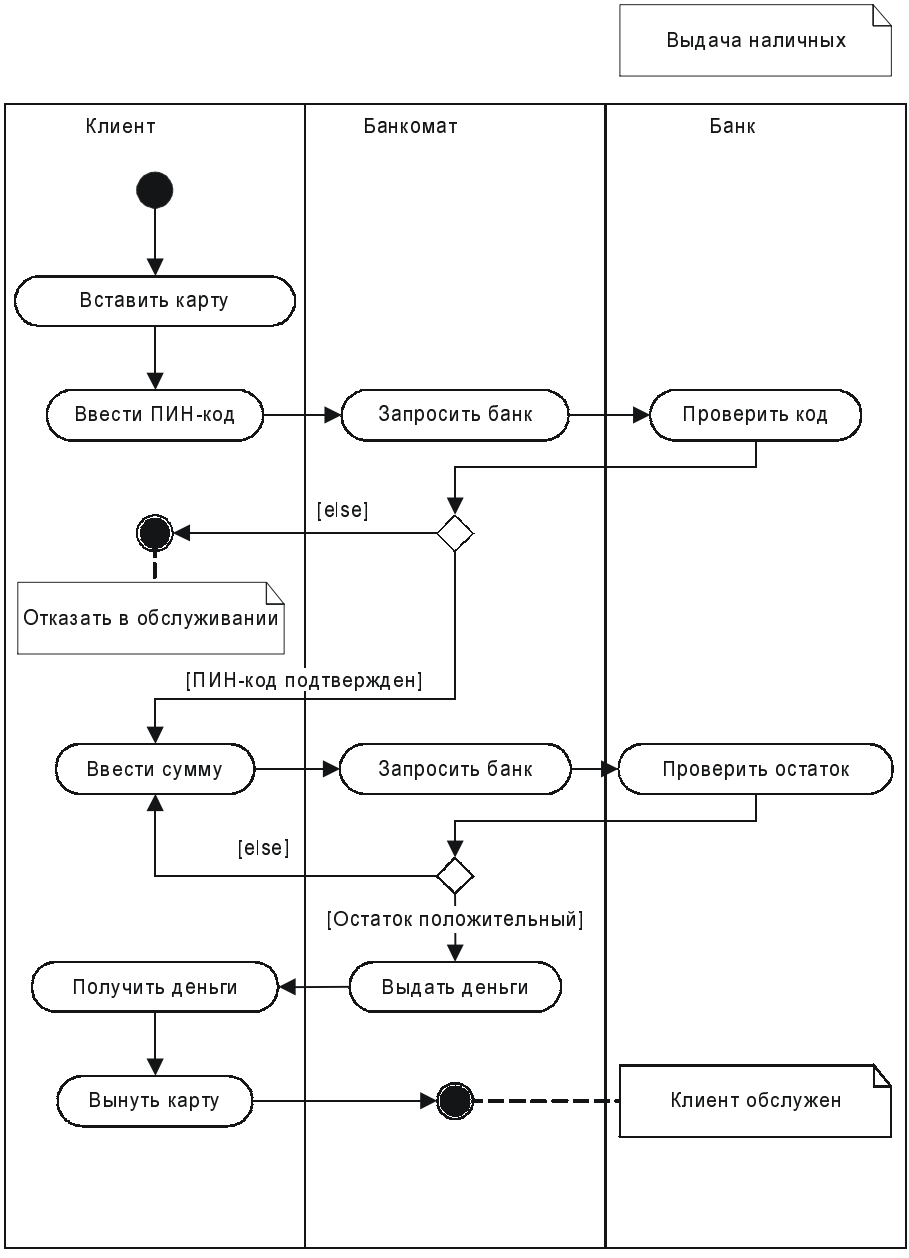

Рис. Д.4. Пример диаграммы деятельности унифицированного языка моделирования UML

Рис. Д.5. Пример диаграммы использования унифицированного языка моделирования UML

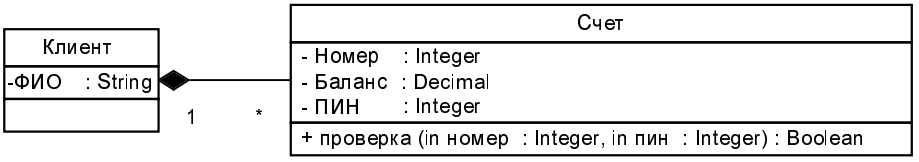

Рис. Д.6. Пример диаграммы классов унифицированного языка моделирования UML

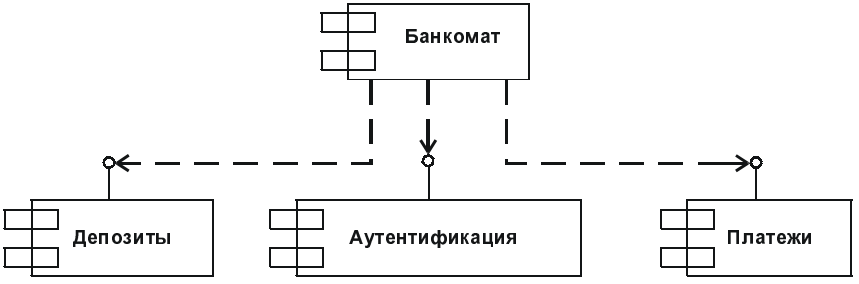

Рис. Д.7. Пример диаграммы компонентов унифицированного языка моделирования UML

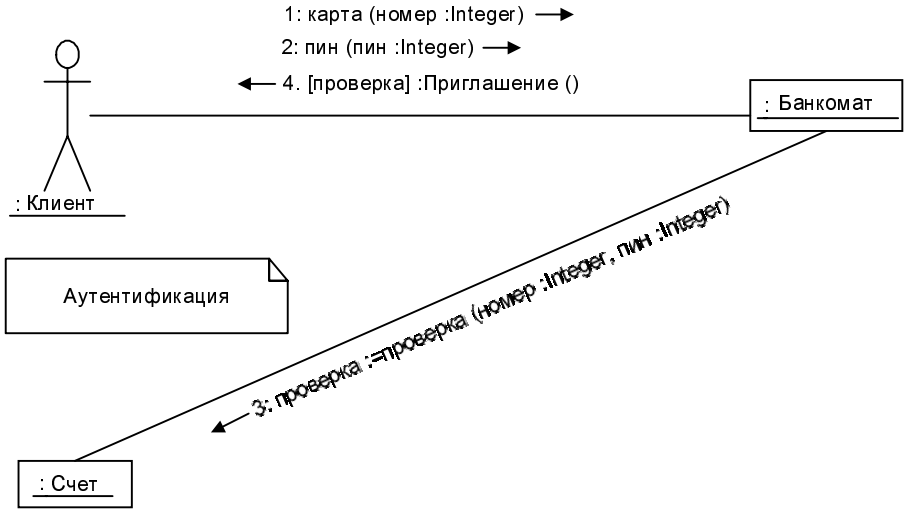

Рис. Д.8. Пример диаграммы кооперации унифицированного языка моделирования UML

ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ [sequence diagram]. Диаграмма *унифицированного языка моделирования* UML, на которой изображено упорядоченное во времени взаимодействие объектов. В частности, на ней изображаются сами объекты, их *линии жизни* и последовательность сообщений, которыми они обмениваются (рис. Д.9). Ср. *диаграмма кооперации*

Рис. Д.9. Пример диаграммы последовательности унифицированного языка моделирования UML

ДИАГРАММА РАЗВЕРТЫВАНИЯ [deployment diagram]. То же, что *диаграмма размещения*

ДИАГРАММА РАЗМЕЩЕНИЯ, диаграмма развертывания [deployment diagram]. Диаграмма *унифицированного языка моделирования* UML, на которой изображается конфигурация для работающих *вычислительных узлов* и *экземпляров компонентов* (рис. Д.10)

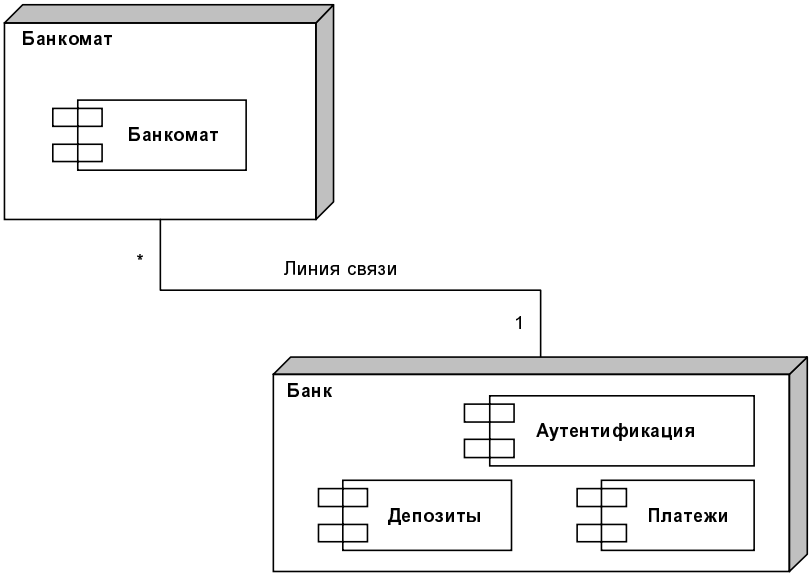

Рис. Д.10. Пример диаграммы размещения унифицированного языка моделирования UML

ДИАГРАММА СОСТОЯНИЙ [statechart diagram]. Диаграмма *унифицированного языка моделирования* UML, на которой изображается *конечный автомат*, его состояния и переходы (рис. Д.11). Изначальная концепция была разработана Дэвидом Харелом (David Harel)

ДИАЛОГ [dialog]. Двусторонний обмен информацией и *управляющими сигналами* между человеком и компьютером в форме вопросов и ответов в темпе, удобном для человека. Пользователь ведет Д. с помощью клавиатуры, мыши или микрофона. Программа выводит информацию на экран дисплея или использует *синтезатор речи* (см. *синтез речи*). Развитые формы Д. могут происходить с одновременным использованием разных устройств ввода/вывода. Например, вывод информации на экран сопровождается или перемежается речью. Существуют специальные *диалоговые системы*, позволяющие настроить Д. на потребности и стиль работы конкретного пользова-

теля. При участии в Д. опытного пользователя Д. управляется преимущественно человеком, в противном случае управление берет на себя компьютер. См. *диалоговый режим, графический диалог, интеллектуальный интерфейс*

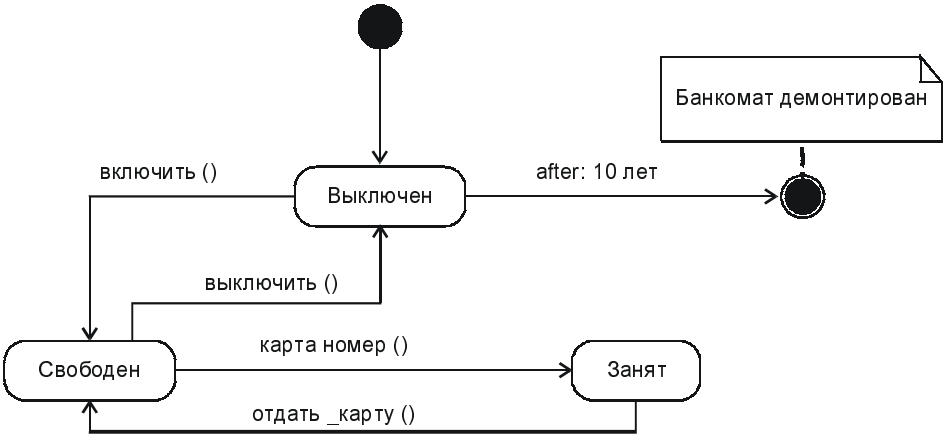

Рис. Д.11. Пример диаграммы состояний унифицированного языка моделирования UML

ДИАЛОГОВАЯ СИСТЕМА [dialog system, conversational system]. 1. Система программ и *аппаратных средств*, обеспечивающая *диалоговый режим* взаимодействия пользователя с исполняемой программой. 2. Программная система, управляемая пользователем в диалоговом режиме

ДИАЛОГОВОЕ ОКНО [dialog window]. См. *окно*

ДИАЛОГОВОЕ ОКНО С ВКЛАДКАМИ [tabbed dialog box]. Диалоговое окно, разделенное на секции, которые выглядят как карточки в картотеке с выступающими ярлычками (рис. Д.12). Для того чтобы раскрыть соответствующую вкладку, нужно щелкнуть на ярлычке или нажать комбинацию клавиш <Ctrl>+<Tab>

ДИАЛОГОВЫЙ ПРОЦЕССОР [conversational processor]. Программа — компонент *интеллектуального интерфейса*, представляющая собой интерпретатор естественного или профессионального языка пользователя. Д. п. распознает конструкции языка пользователя и преобразует их в описание задачи на *промежуточном языке интеллектуальной системы программирования*. По этому описанию специальными средствами системы создается программа решения задачи на основе информации, хранящейся в *базе знаний*

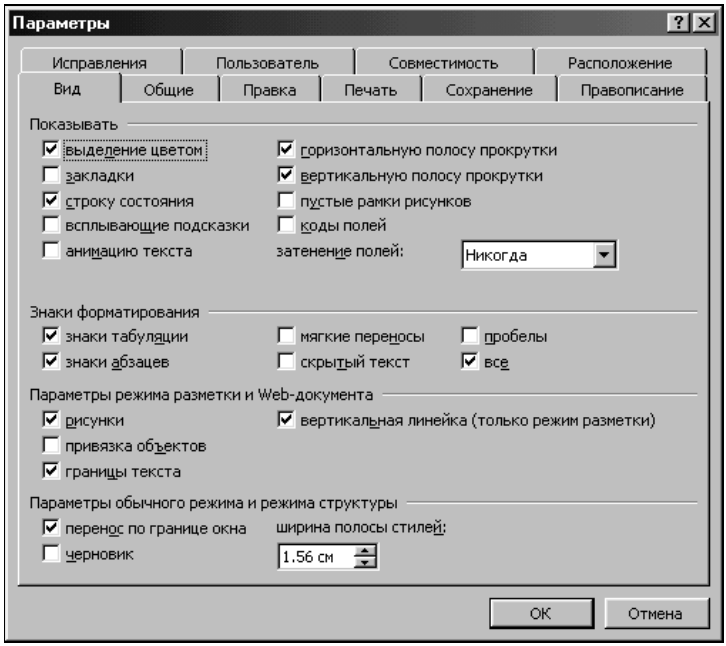

Рис. Д.12. Диалоговое окно с вкладками в приложении Word

ДИАЛОГОВЫЙ РЕЖИМ, **интерактивный режим [dialog mode**, **interactive mode]**. *Режим* взаимодействия пользователя с исполняемой *программой* или *вычислительной системой*, при котором пользователь управляет программой во время ее работы. Например, Д. р. используется для решения задач, программа или *исходные данные* которых в момент начала решения могут быть известны не полностью. Человек следит за процессом решения, фиксирует те или иные промежуточные результаты и по ходу решения задачи выдает программе инструкции. Д. р. обычно предполагает задание пользователем с терминала команд (запросов), вызывающих немедленные ответные действия, которые сопровождаются соответствующими сообщениями или подтверждениями компьютера. Внешним проявлением Д. р. является диалог пользователя с компьютером, для реализации которого могут применяться соответствующие средства. См. *диалог, интеллектуальный интерфейс*

ДИАПАЗОН ИЗМЕНЕНИЯ ЧИСЕЛ [number range]. Область значений, которые могут принимать числа при принятой в данной ЭВМ *системе счисления* и форме *представления чисел*. Д. и. ч. *целого типа* определяется неравенством $|N| \le q^L - 1$, где q — основание *системы счисления*; L — количество разрядов, отводимых под число, минус единица, соответствующая знаку. Например, если известно, что *число целого типа* в *двоичной системе счисления* занимает *машинное слово* длиной 4 байта = 32 бита, это значит, что его величина не

может превышать значения $2^{31} - 1 = 2\,147\,483\,647$. Д. и. ч. *с плавающей точкой* определяется неравенством

$$\left| i = q^{-q^p} \le |R| \le \left(1 - q^{-m}\right) q^{q^p - 1} \right| = s,$$

где i и s — нижний и верхний пределы ненулевых чисел с плавающей точкой; q — основание системы счисления; p — число разрядов, предназначенных для хранения абсолютной величины порядка; m — число разрядов, предназначенных для хранения абсолютной величины мантиссы. Для нуля существует особое представление. Представление чисел с плавающей точкой обеспечивает относительную ошибку порядка q^{-m}. Поэтому если известно, что в некоторой *вычислительной системе* двоичные числа с плавающей точкой занимают 8 байт = 64 бита, из которых 1 байт = 8 бит отводится на порядок и его знак и, следовательно, 7 байт = 56 бит — на мантиссу (включая знак числа), то диапазон изменения вещественных чисел в такой системе будет

$$0 \text{ и } i = 2^{-128} \le |R| \le (1 - 2^{-55}) \times 2^{127} = s,$$

где s и i — верхний и нижний пределы абсолютных значений ненулевых чисел с плавающей точкой. Таким образом, множество представимых в компьютере чисел с плавающей точкой отличается от множества *действительных чисел*. Оно ограничено и имеет конечное число элементов. При этом его плотность на числовой оси не везде одинакова. Действительно, числа с плавающей точкой более "сжаты" со стороны малых величин, чем со стороны больших величин: между i и $2i$ имеется столько же чисел, сколько между $2i$ и $4i$, между $4i$ и $8i$, ..., между $s/2$ и s (рис. Д.13). С другой стороны, ноль изолирован от других величин и должен обрабатываться отдельно. См. *представление чисел с плавающей точкой*

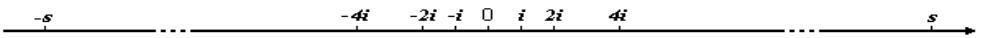

Рис. Д.13. Схема расположения на числовой оси представимых в ЭВМ чисел с плавающей точкой

ДИГИТАЙЗЕР [digitizer]. 1. Устройство, предназначенное для ввода чертежей с листа (рис. Д.14). Д. снабжен специальным инструментом, называемым *пером*. При перемещении пера над линиями чертежа производятся отсчеты его координат в близко расположенных точках. Эти данные вводятся в компьютер. Тем самым происходит преобразование готовых изображений в *цифровую форму данных*. Д. позволяют обрабатывать на компьютере, например, метеорологические данные, представленные в виде карт и графиков. См. *цифровой планшет графического ввода*. 2. Устройство, преобразующее непрерывный *аналоговый сигнал* в дискретную цифровую форму

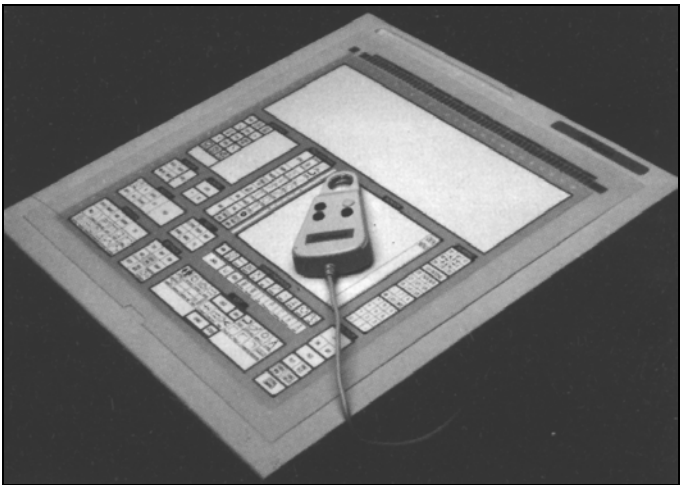

Рис. Д.14. Дигитайзер

ДИЗЪЮНКЦИЯ [disjunction]. То же, что *логическое "ИЛИ"*. См. *логические операции*

ДИНАМИЧЕСКАЯ ЗАГРУЗКА ПРОГРАММЫ [dynamic program loading]. Способ *загрузки программы*, при котором в *оперативную память* считывается не вся программа; модули, отсутствующие в памяти, подгружаются по мере обращения к ним. Д. з. п. может сопровождаться *динамическим размещением модулей* либо *оверлейным размещением модулей*. При динамическом размещении подгружаемые модули могут загружаться в разные свободные области оперативной памяти, и связи между ними устанавливаются динамически, *операционной системой* в процессе загрузки. При оверлейном размещении по мере необходимости подгружаются *сегменты перекрытия*, которые могут состоять из нескольких *программных модулей*. Они загружаются в одну и ту же область памяти, перекрывая ранее загруженные в нее сегменты той же программы. План такой загрузки и связи между оверлейно размещаемыми модулями устанавливаются статически, *системой программирования* до начала выполнения программы. См. *перекрытие, оверлейная программа*

ДИНАМИЧЕСКАЯ МАРШРУТИЗАЦИЯ [dynamic routing]. Процесс изменения в *таблицах маршрутизации*, происходящий в реальном масштабе времени в соответствии с изменениями в самой сети. Программное обеспечение изменяет маршруты, основываясь на служебных сообщениях, а затем распространяет сообщения об изменениях в своих маршрутах

ДИНАМИЧЕСКАЯ ОБЛАСТЬ ПАМЯТИ, проблемная область памяти [dynamic storage area]. *Область оперативной памяти*, предназначенная для разме-

щения *прикладных программ* на время их выполнения. После *завершения программы* Д. о. п. освобождается для загрузки другой программы. См. *распределение памяти*. Ср. *зона*

ДИНАМИЧЕСКАЯ ВЕБ-СТРАНИЦА [dynamic Web page]. *Веб-страница*, предназначенная для облегчения доступа к программам и данным. Д. в.-с., как правило, содержит очень мало собственно текста. Вместо этого здесь имеются встроенные ссылки на другие приложения, из которых Д. в.-с. извлекает информацию по запросам пользователя. Например, Д. в.-с. может предоставлять пользователю информацию определенного сорта, извлекая ее из различных уже существующих во *Всемирной паутине* баз данных

ДИНАМИЧЕСКИ ЗАГРУЖАЕМАЯ БИБЛИОТЕКА [dynamic link library (DLL)]. То же, что *библиотека динамической компоновки*

ДИНАМИЧЕСКИЙ ОБМЕН ДАННЫМИ [dynamic data exchange (DDE)]. Средство, позволяющее приложениям Windows обмениваться друг с другом данными и командами. Суть Д. о. д. состоит в следующем. Вначале первое приложение должно установить связь со вторым. Затем первое приложение может посылать второму команды для исполнения, данные для использования, запрашивать и получать результаты. В конце сеанс связи по DDE должен быть завершен. Д. о. д. поддерживают практически все приложения Windows, начиная с самых первых версий *операционной системы Windows*. Например, копирование данных из документа одного приложения в документ другого приложения с помощью *буфера обмена* в Windows основано на Д. о. д. В настоящее время Д. о. д. постепенно вытесняется более современными средствами, такими как *связывание и внедрение объектов* (OLE). Одна из причин состоит в том, что DDE не обеспечивает достаточных средств для обработки *исключительных ситуаций*

ДИНАМИЧЕСКОЕ РАЗМЕЩЕНИЕ МОДУЛЕЙ [dynamic modules allocation]. Размещение *программных модулей* в свободных областях оперативной памяти при *динамической загрузке программы*. При Д. р. м. связи между ними устанавливаются динамически, *операционной системой* в процессе динамической загрузки. Ср. *оверлейное размещение модулей*

ДИНАМИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ ПАМЯТИ [dynamic memory allocation]. Способ *распределения оперативной памяти*, при котором она выделяется и освобождается по запросам программы в ходе ее выполнения. Противоп. *статическое распределение памяти*. См. *распределение памяти, зона*

ДИНАМИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ РЕСУРСОВ [dynamic resource allocation]. Способ *распределения ресурсов*, при котором они выделяются и освобождаются по запросам программы в ходе ее выполнения. Противоп. *статическое распределение ресурсов*. См. *распределение ресурсов*

ДИРЕКТИВА [directive]. Предложение или команда, управляющие *вычислительной системой* или ее компонентами. Д. содержит указания системе о том, какие необходимо выполнить действия. Д. могут вводиться оператором ЭВМ или пользователем, а также содержаться в программе. В некоторых *языках программирования* предусмотрены Д., позволяющие управлять работой препроцессора, компилятора, компоновщика и других компонентов *системы программирования*. Например, в *языке ассемблера* с помощью Д. управляют последующей компоновкой программы, а в языке Си с помощью Д. управляют работой препроцессора. См. *включение файла*

ДИРЕКТОРИЯ [directory]. То же, что *каталог*

ДИСК [disk]. 1. *Носитель данных*, представляющий собой диск, поверхности которого покрыты материалом, способным запоминать информацию. В зависимости от физического явления, применяемого в процессе записи данных на Д., различают *магнитные* и *лазерные диски*. Считывание информации с Д. происходит при его вращении в электромеханическом устройстве, называемом приводом или дисководом. 2. То же, что *логический диск*

ДИСК АВАРИЙНОГО ВОССТАНОВЛЕНИЯ [recovery disk]. *Диск*, созданный *служебной программой* архивирования данных и содержащий значения текущих системных параметров *операционной системы Windows*. Этим диском можно воспользоваться при восстановлении переставшего загружаться компьютера, а также утраченных или поврежденных *системных файлов*

ДИСК БЕРНУЛЛИ [Bernoulli disk]. *Накопитель* на сменяемых *гибких магнитных дисках* большой емкости, разработанный компанией Iomega. В Д. Б. *головка записи/чтения* поддерживается над диском воздушным потоком в соответствии с принципами аэродинамики, сформулированными швейцарским математиком Д. Бернулли. По своим характеристикам Д. Б. не уступает *винчестерскому диску*, но в отличие от него имеет сменный *носитель данных*, похожий на обычную дискету, и в несколько раз превосходит его по долговечности. Разновидностями Д. Б. являются накопители Zip и Jaz, имеющие емкость 90, 100, 150, 230 Мбайт и 1 Гбайт на дискету

ДИСКЕТА [diskette]. То же, что *гибкий магнитный диск*

ДИСКОВОД [disk drive]. Устройство для записи и считывания информации с *магнитных* и *лазерных дисков*. Включает в себя механизм вращения дисков, *головки записи/чтения*, устройство, управляющее перемещением головок, и соответствующие *электронные схемы*. Существуют Д., предназначенные либо для одного диска, либо для *пакета магнитных дисков*

ДИСКРЕТНЫЙ ПЕРЕНОС, **мягкий перенос [discretionary hyphen]**. В *настольных редакционно-издательских системах* — знак потенциального переноса, который реализуется только при попадании слова в зону переноса именно в данном месте

ДИСКРЕТНЫЙ СИГНАЛ [discrete signal]. То же, что *цифровой сигнал*

ДИСПЕТЧЕР [dispatcher, manager]. Программа, управляющая распределением ресурсов. Например, Д. задач обслуживает очередь на использование центрального процессора — согласно приоритету выбирает задачи из очереди и передает им управление. См. *диспетчер файлов*

ДИСПЕТЧЕР ВИРТУАЛЬНОЙ ПАМЯТИ [virtual memory manager (VMM)]. *Драйвер устройства*, предназначенный для управления *дополнительной* (и в некоторых случаях *отображаемой*) *памятью*. Д. в. п. впервые появился еще в MS-DOS. В *операционных системах Windows* одной из функций Д. в. п. является работа с *файлом подкачки*

ДИСПЕТЧЕР РАСШИРЕННОЙ ПАМЯТИ [extended memory manager (XMM)]. *Драйвер*, обеспечивающий доступ к *расширенной памяти персонального компьютера*, работающего под управлением *операционной системы MS-DOS*. В других ситуациях в настоящее время этот термин не используется

ДИСПЕТЧЕР УСТРОЙСТВ [devices manager]. Программа административного управления устройствами компьютера. С помощью Д. у. можно просматривать и изменять свойства и обновлять *драйверы устройств*, а также выполнять их настройку и удаление из системы

ДИСПЕТЧЕР УСТРОЙСТВ SCSI [SCSI manager]. *Диспетчер устройств*, осуществляющий управление *устройствами SCSI*. В Windows NT в качестве Д. у. SCSI используется специальный *драйвер мини-порта*

ДИСПЕТЧЕР ФАЙЛОВ [file manager]. Модуль операционной системы, управляющий размещением файлов в физической памяти и доступом к ним. Д. ф. обрабатывает запросы на открытие и закрытие файлов, ведет список установленных дисков и т. д. См. *открытый файл*

ДИСПЛЕЙ [display]. Устройство визуального отображения информации, обрабатываемой компьютером. Представляет собой *устройство вывода* в виде экрана или табло, на котором можно временно отображать информацию как в буквенно-цифровой, так и в графической форме. В настоящее время наиболее широкое распространение получили Д. на основе электронно-лучевой трубки. Под управлением ЭВМ в качестве Д. может работать даже бытовой телевизор, однако специализированные Д. *(мониторы)* дают более четкое и устойчивое изображение, что позволяет избежать излишней утомляемости пользователя. Д. может быть цветным или монохромным (черно-белым). Основные характеристики Д. — размер по диагонали (в дюймах) и размер зерна (в миллиметрах). Информация, выводимая на экран Д., может быть текстовой (буквенно-цифровой) или графической (графики, диаграммы, рисунки и пр.). Работой Д. управляет *адаптер дисплея*, преобразующий данные, поступающие из компьютера в *цифровой форме*, в соответствующее изображение на экране. От адаптера дисплея зависит *разрешающая способ-*

ность — количество пикселов (точек), на которые разбит экран (указывается в виде произведения числа точек по горизонтали на число точек по вертикали). Современные видеоадаптеры обеспечивают разрешающую способность до 1024×768 пикселов и выше. В большинстве Д. имеются возможности для выделения или разделения различных участков текста или изображения путем мерцания, яркого свечения или окраски. Для уменьшения вредного воздействия излучения от Д. иногда применяется защитный экран. В настоящее время разработаны дисплеи с низким излучением, маркированные Low Radiation (LR). В *портативных компьютерах* применяются *дисплеи на жидких кристаллах*

ДИСПЛЕЙ НА ЖИДКИХ КРИСТАЛЛАХ, **жидкокристаллический дисплей [liquid crystal display (LCD)]**. Дисплей, изображение на котором создается посредством управляемых электронными сигналами жидких кристаллов. Жидкие кристаллы обладают способностью под воздействием электрического поля приобретать молекулярную структуру, пропускающую или не пропускающую поляризованный свет. Плоский экран Д. н. ж. к. состоит из тонкого слоя жидкокристаллических ячеек, заключенных между сетками горизонтальных и вертикальных электродов. Этот слой в свою очередь помещается между двумя фильтрами — поляризаторами света. По командам компьютера между электродами образуется электрическое поле, выборочно "включающее" или "выключающее" ячейки (пикселы). Тем самым формируется изображение на экране. У персональных компьютеров экран может освещаться расположенной сзади электролюминесцентной панелью и бывает цветным или монохромным

ДИСТРИБУТИВ [distribution kit]. *Программный продукт* в виде, поставляемом производителем (чаще всего на компакт-дисках). Дистрибутивный диск, как правило, содержит саму программу и *инсталлятор* (обычно с именем install или setup) для установки программы на *жестком диске* и настройки ее параметров. Иногда применяются дистрибутивные дискеты

ДЛИНА [length]. 1. Размер *области памяти*, отводимой объекту (программе, *блоку данных*, записи, *машинному слову*, регистру и т. д.). Измеряется в битах, байтах и т. п. 2. Число элементов, образующих объект. Например, длина блока данных — это и размер в байтах области памяти, отведенной блоку данных, и число машинных слов в блоке; длина файла — это и количество байтов, занимаемых файлом в памяти, и число блоков в файле; длина дорожки — это и *информационная емкость* дорожки диска, и число размещенных на ней секторов

ДЛИНА КОНВЕЙЕРА [pipeline length]. Общее число *сегментов конвейера*. См. *конвейерная обработка*

ДЛИНА РЕГИСТРА [register length]. То же, что *емкость регистра*

ДЛИННОЕ ИМЯ [long name]. *Имя* папки или файла, превышающее формат имени файла, принятый в *файловой системе FAT* (до восьми символов в имени с последующей точкой и трехсимвольным расширением). Начиная с Windows 95 операционные системы (в т. ч. *операционная система Windows 2000*) поддерживают имена файлов длиной до 255 символов

ДОБАВЛЯЕМЫЙ МОДУЛЬ, **добавочный модуль**, **надстройка [add-on module]**. *Модуль*, сделанный в дополнение к основному *программному продукту* или устройству для улучшения его свойств или возможностей. Многие приложения позволяют динамически устанавливать необходимые Д. м. Например, на рис. Д.15 показано окно для установки Д. м. в приложении Microsoft Excel

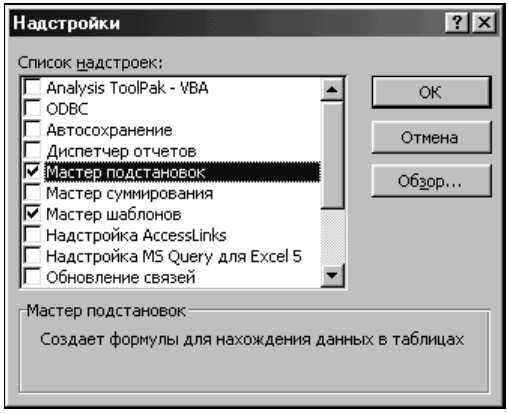

Рис. Д.15. Диалоговое окно **Надстройки** приложения Microsoft Excel

ДОБАВОЧНЫЙ МОДУЛЬ [add-on module]. То же, что *добавляемый модуль*

ДОВЕРИТЕЛЬНОЕ ОТНОШЕНИЕ [trust]. Отношение между пользователями в *компьютерной сети* с несколькими доменами Windows NT. Д. о. возникает тогда, когда пользователям одного домена разрешают обращаться к ресурсам другого домена без необходимости явной регистрации в этом домене. Домены Windows 2000 автоматически связываются между собой двусторонними, транзитивными отношениями при включении нового домена в *дерево доменов*

ДОКУМЕНТ [document]. 1. Важная деловая информация, облеченная в материальную форму. Например, Д. может являться содержащаяся на *носителе данных* в зафиксированном виде текстовая, графическая или звуковая информация (см. *электронный документ*). Д. может быть официальным и неофициальным. В официальных Д. информация оформляется установленным

порядком и имеет в соответствии с действующим законодательством правовое значение. 2. Текст, подготавливаемый в *текстовых процессорах* или *текстовых редакторах*

ДОКУМЕНТ WEB [Web document, WWW document]. То же, что *веб-документ*

ДОКУМЕНТ WWW [document WWW]. То же, что *веб-документ*

ДОКУМЕНТАЦИЯ ПРОГРАММНОГО ПРОДУКТА, программная документация [software documentation]. Совокупность *документов*, описывающих назначение, структуру и применение *программного продукта*. Д. п. п. предназначена для облегчения использования программного изделия и включает как печатные документы (описания, руководства, учебники, справочники и т. п.), так и *электронные документы* (выводимые на экран и распечатываемые тексты, а также *обучающие программы*, средства диалоговой подсказки и т. п.)

ДОКУМЕНТООРИЕНТИРОВАННЫЙ ПОДХОД [data centric] Подход, используемый современными *операционными системами*, ориентированный на работу с документами, а не работу с приложениями. Этот подход предоставляет возможность пользователю сосредоточиться на работе с конкретными документами, а не думать о том, какие приложения нужно использовать для выполнения задания

ДОМАШНЯЯ СЕТЬ [home network]. Локальная *сеть передачи данных*, обеспечивающая дистанционное управление бытовыми приборами и устройствами: домашним компьютером, видеомагнитофоном, охранной сигнализацией, системой освещения и микроклимата и т. д.

ДОМАШНЯЯ СТРАНИЦА, заглавная страница, начальная страница [home page]. Первая *веб-страница*, на которую попадает посетитель веб-сайта. С нее начинается просмотр сайта. По Д. с. посетитель обычно получает представление о том, куда он попал и что он может увидеть на других страницах сайта. На рис. Д.16 показан фрагмент Д. с. издательства "БХВ-Петербург"

ДОМЕН [domain]. 1. В *системах управления базами данных* — множество допустимых значений данного атрибута. Например, Д. атрибута "код города" может быть список всех допустимых трехзначных числовых телефонных кодов городов России, а Д. атрибута "день года" — список всех трехзначных чисел от 001 до 366. Понятие Д. имеет также скрытый смысл. При использовании *реляционных баз данных* часто приходится объединять два отношения или две таблицы, основываясь на соответствующих значениях в столбцах, обладающих одним и тем же Д. При этом нельзя объединять две таблицы на базе атрибутов, принадлежащих разным Д., даже если эти атрибуты имеют одинаковый формат. Например, атрибуты "код города" и "день года" объединить нельзя. 2. Составная часть имени компьютера в сети Интернет. См. *доменная система имен*

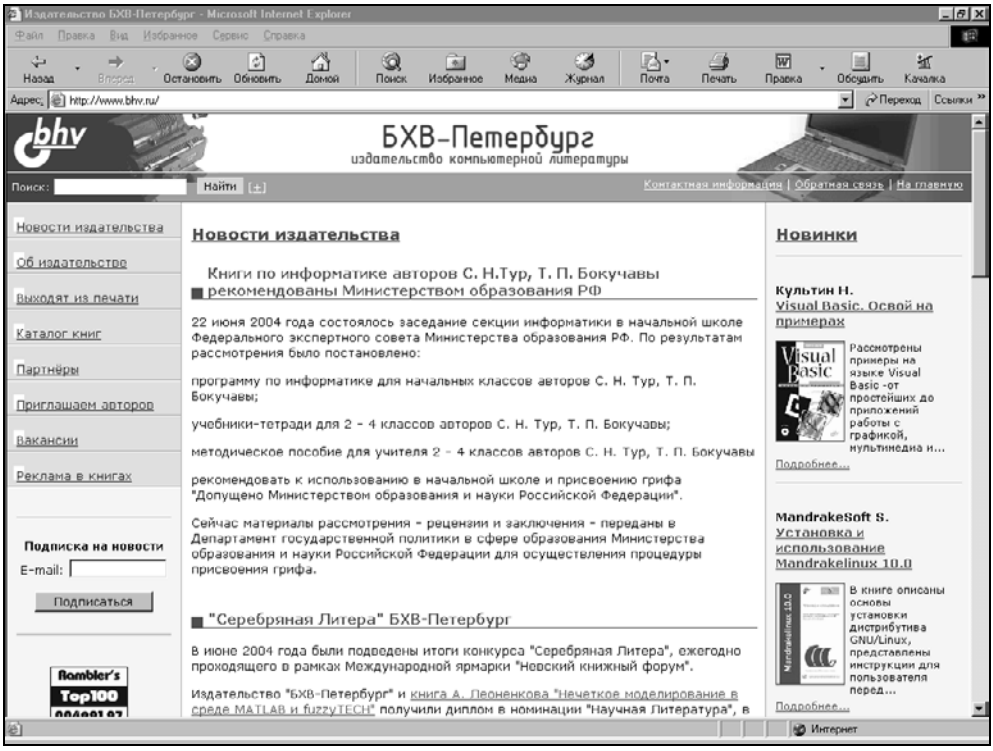

Рис. Д.16. Фрагмент домашней страницы издательства "БХВ-Петербург"

ДОМЕННАЯ СИСТЕМА ИМЕН [domain name system (DNS)]. Иерархическая система назначения уникальных имен каждому из компьютеров, связанных в сети, подобные Интернету. Имя состоит из доменов — разделенных точками частей, соответствующих возрастающим слева направо уровням иерархии. В имени может быть любое число доменов, но более пяти встречается редко. Значение каждого домена назначается лицами, отвечающими за соответствующий иерархический уровень. Например, в имени **cmp.ctd.unv.reg.ru** элемент **cmp** — имя реального компьютера. Это имя создано на уровне кафедры (**ctd**), где стоит компьютер. Кафедра принадлежит университету (**unv**), который подключен к региональной *компьютерной сети* (**reg**), находящейся в России (**ru**). Некоторые значения доменов верхних уровней, принятые в сети Интернет, приведены в табл. Д.1. По имени сервером Д. с. и. автоматически формируется числовой *интернет-адрес*, который используется аппаратурой и программами сети. Наряду с реальными именами существуют виртуальные имена, предоставляемые *поставщиками сетевых услуг* (провайдерами). Например, если реальное имя оказывается слишком длинным, провайдер может его заменить коротким виртуальным, которое распознается сервером Д. с. и. и преобразуется в реальный числовой адрес

Таблица Д.1. *Значения доменов верхнего уровня доменной системы имен Интернета*

Домен	Значение	Домен	Значение
com	Коммерческие организации	ru	Россия
edu	Учебные заведения	us	США
net	Сетевые ресурсы	ca	Канада
gov	Правительственные учреждения	uk	Великобритания
mil	Военные учреждения	au	Австралия
org	Прочие организации	cz	Республика Чехия

ДОМЕННОЕ ИМЯ [domain name]. То же, что *имя домена*

ДОПОЛНИТЕЛЬНАЯ ПАМЯТЬ [add-in memory]. *Запоминающее устройство*, подключаемое к компьютеру для увеличения объема *оперативной памяти*

ДОРОЖКА [track]. Часть поверхности движущегося *носителя данных* (*магнитного* или *лазерного диска*, *магнитной ленты* и т. п.), запись на которую или чтение с которой осуществляются одной *головкой записи/чтения*. Каждая Д. имеет свой номер (адрес)

ДОСТУП [access]. 1. Процедура установления связи с *запоминающим устройством* и размещенным на нем файлом для чтения, записи или изменения данных. 2. Возможность прочесть, записать или изменить данные. Обычно в этом случае указывается особенность или способ Д. Например, Д. к файлу только для чтения означает, что данные из файла можно только считывать, и в процессе считывания содержимое файла не изменяется и не стирается. См. *прямой доступ, последовательный доступ, несанкционированный доступ, удаленный доступ, права доступа*

ДОЧЕРНЕЕ ОКНО [child window]. *Окно*, которое принадлежит другому окну, называемому *родительским* окном. Например, окно рабочей книги MS Excel является Д. о. окна приложения MS Excel (рис. Д.17). См. *родительское окно*

ДРАЙВЕР [driver]. Управляющая программа. Обычно это программа *операционной системы*, обеспечивающая взаимодействие исполняемой программы с отдельным устройством и способствующая его удобному использованию. Например, существуют Д. клавиатуры, дисплея, мыши, принтера и т. п. Д. принимает запросы программ на обращение к устройству и преобразует их в команды управления устройством, а также обрабатывает прерывания от обслуживаемого устройства. При этом он учитывает конструктивные особенности устройства и особенности его работы в реальном масштабе времени.

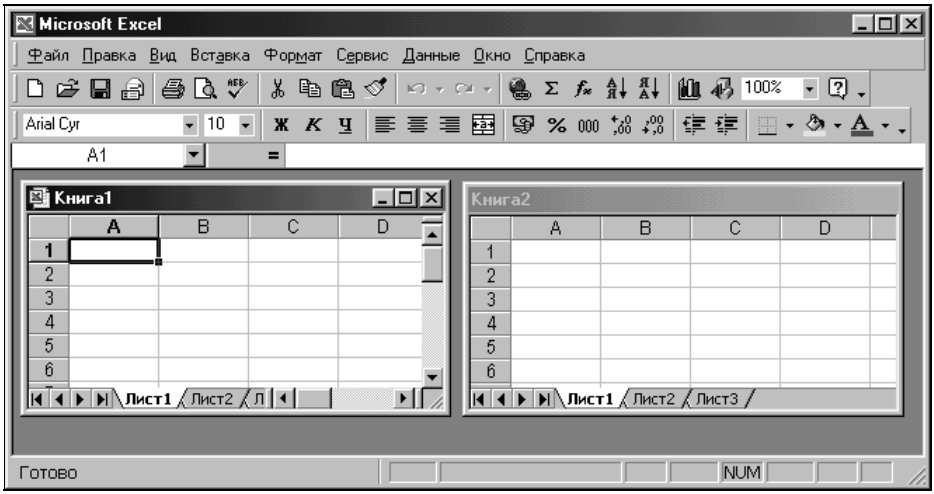

Рис. Д.17. Родительское окно приложения Microsoft Excel и дочерние окна двух открытых рабочих книг

Таким образом, Д. является связующим звеном между программой и устройством. Д. может быть оформлен как *резидентная программа* или как *загружаемый* Д. Наряду с реальными Д. может обслуживать и виртуальные устройства, как бы имитируя реальное физическое устройство. Д. считаются также программы, управляющие дополнительной памятью компьютера, и программы, управляющие другими программами, т. е. вызывающие их и задающие им входные параметры, например, драйвер базы данных ODBC. См. *драйвер виртуального устройства, драйвер логического устройства, драйвер устройства*

ДРАЙВЕР ВИРТУАЛЬНОГО УСТРОЙСТВА [virtual anything driver (VxD)]. Специальный класс *драйверов*, предоставляющий доступ к *виртуальным* устройствам, т. е. к устройствам, которые имитируются программным или аппаратным способом. Например, драйвер виртуального дисплея — программа, управляющая экраном дисплея. VxD — общее обозначение драйверов виртуальных устройств в *операционных системах Windows*. Так, драйвер виртуального дисплея обозначается VDD

ДРАЙВЕР ЛОГИЧЕСКОГО УСТРОЙСТВА [type-specific driver (TSD)]. *Драйвер*, обслуживающий (в отличие от драйверов физических устройств) *логические устройства*, относящиеся к одному типу. Например, для работы со всеми *жесткими дисками* служит один драйвер, для работы со всеми *гибкими дисками* — другой. Ср. *драйвер устройства*

ДРАЙВЕР МИНИ-ПОРТА [miniport driver]. Компонент *операционной системы Windows*, являющийся *драйвером логического устройства* и отвечающий за работу с устройствами определенного типа. Обычно все функции, не зави-

сящие от конкретной модели устройства, выполняются Д. м., а не *драйвером устройства*. См. *диспетчер устройств SCSI*

ДРАЙВЕР ПОРТА [port driver (PD)]. Компонент *операционной системы*, обеспечивающий доступ к *портам* устройства, подключенного к компьютеру. Д. п. зависит от конкретного типа и модели устройства. Например, для каждого дискового контроллера и накопителя используется отдельный Д. п.

ДРАЙВЕР ПРИНТЕРА [printer driver]. *Драйвер*, позволяющий приложениям корректно взаимодействовать с печатающим устройством, независимо от его типа и модели, а также используемого языкового интерпретатора. См. *задания на печать, очередь заданий на печать*

ДРАЙВЕР-РУСИФИКАТОР [Cyrillic driver]. *Драйвер*, поддерживающий ввод в память компьютера и вывод на экран дисплея символов — букв русского алфавита (кириллицы). Д.-р. обычно является *резидентной программой*, которая активизируется нажатием *командной клавиши* (либо сочетания клавиш), служащей командой смены латинского алфавита на русский. См. *раскладка клавиатуры*

ДРАЙВЕР УСТРОЙСТВА [device driver]. *Драйвер*, позволяющий конкретному устройству, такому как модем, *сетевая плата* или принтер, взаимодействовать с *операционной системой*. Если устройство включено в *список совместимого оборудования*, то драйвер такого устройства обычно входит в состав операционной системы. Драйверы устройств загружаются автоматически при запуске компьютера и с этого момента выполняются, оставаясь невидимыми для пользователя. Ср. *драйвер логического устройства*

ДРАЙВЕР ФАЙЛОВОЙ СИСТЕМЫ [file system driver (FSD)]. Компонент *файловой системы*, служащий для связи *операционной системы* с устройством длительного хранения данных (жестким или гибким диском). Кроме того, Д. ф. с. отвечает за поддержку *длинных имен файлов* и взаимодействие пользователя с конкретным устройством. Например, Д. ф. с. CD не позволяет записывать данные на устройства чтения компакт-дисков

ДРЕВОВИДНАЯ СЕТЬ, сеть древовидной топологии [tree network]. Локальная вычислительная сеть, узлы которой объединены в структуру типа "дерево". См. *топология сети*. Ср. *кольцевая сеть*, *радиальная сеть*, *шинная сеть*

ДРЕВОВИДНАЯ ТОПОЛОГИЯ [tree topology]. Топология сети, при которой ее узлы объединены в структуру типа "дерево"

ДРУЖЕСТВЕННЫЙ ИНТЕРФЕЙС [friendly interface]. *Интерфейс*, обеспечивающий человеку-пользователю не требующее специального обучения максимально удобное взаимодействие с программой или *вычислительной системой*. Это наглядные, простые и понятные для него изображения на экране, значки, пиктограммы, *кнопки*, меню, подсказки в диалоге, звуковое сопровождение и т. п. См. *графический интерфейс пользователя, интеллектуальный интерфейс*

ДЮЙМ [inch, ″]. Единица измерения длины. 1 дюйм = 1 inch = 1″= 25,4 мм

Е

ЕВРОПЕЙСКАЯ ЛАБОРАТОРИЯ ЯДЕРНЫХ ИССЛЕДОВАНИЙ (ЦЕРН) [European Particle Physics Laboratory, франц: **Conseil Europeen pour la Recherche Nucleaire (CERN)]**. Международный исследовательский центр физики высоких энергий (Швейцария). Место рождения *World Wide Web (WWW)*, созданной в 1989 г. как сеть для сотрудников центра

ЕДИНИЦА ФУНКЦИОНАЛЬНОГО РАЗМЕРА [functional point (FP)]. Единица измерения сложности *программного обеспечения*

ЕМКОСТЬ ПАМЯТИ, **объем памяти [memory capacity]**. Максимальное количество информации, которое может храниться в *запоминающем устройстве*. Е. п. определяет способность памяти разместить определенное количество единиц данных. Е. п. может измеряться в словах, битах, байтах и т. п. Е. п. ЭВМ должна соответствовать возможностям ее *центрального процессора*. Так, в *персональных IBM-совместимых компьютерах* на базе процессора 80286 трудно было разумно использовать больше 1 Мбайт *оперативной памяти*. В современных компьютерах на базе процессоров Pentium необходимый объем оперативной памяти определяется по потребностям используемых программ и доходит до нескольких гигабайт. Ср. *информационная емкость*

ЕМКОСТЬ РЕГИСТРА, **длина регистра**, **размер регистра [register capacity**, **register length**, **register size]**. Максимальное количество информации, которое можно разместить в *регистре*. Измеряется в битах, байтах, цифрах или *машинных словах*

ЕСТЕСТВЕННЫЙ ЯЗЫК [natural language]. Язык общения между людьми, правила которого основываются на текущем употреблении, а не на точном предварительном описании. Например, любой национальный разговорный язык. Е. я. противопоставляется *формализованный язык*

Ж

ЖЕСТКИЙ ДИСК [hard disk]. 1. То же, что *жесткий магнитный диск*. 2. То же, что *винчестерский диск*

ЖЕСТКИЙ МАГНИТНЫЙ ДИСК, **жесткий диск [hard disk]**. *Магнитный диск*, представляющий собой плоский диск из алюминиевого сплава, поверхность которого покрыта магнитным материалом. Для чтения и записи данных Ж. м. д. помещается в специальное устройство — *дисковод*, снабженное магнитными *головками записи/чтения*. В дисководе диск постоянно вращается с большой скоростью. Данные записываются на поверхности Ж. м. д. вдоль концентрических окружностей, называемых дорожками. Для

выбора нужной дорожки подвижная головка записи/чтения устанавливается в соответствующую позицию. Современные Ж. м. д. выполняются по винчестерской технологии. См. *винчестерский диск*

ЖИДКОКРИСТАЛЛИЧЕСКИЙ ДИСПЛЕЙ [liquid crystal display (LCD)]. То же, что *дисплей на жидких кристаллах*

З

ЗАВЕРШЕНИЕ ПРОГРАММЫ [completion, termination]. Последняя фаза обработки программы *операционной системой*, в ходе которой освобождаются запрошенные программой ресурсы, выводятся результаты, очищается выделенная программе *оперативная память*. З. п. может быть нормальным (после успешного выполнения программы) и аварийным. См. *аварийное завершение*. Ср. *прерывание*

ЗАВЕРШЕНИЕ ТРАНЗАКЦИИ [commitment, transaction commitment]. Действия, выполняемые СУБД при *обработке транзакций* в том случае, когда все операции транзакции успешно выполнены. З. т. заключается в том, что результаты всех операций транзакции фиксируются в *базе данных*, *блокировка данных* снимается и данные становятся доступными для обработки другими транзакциями. Ср. *откат транзакции*

ЗАВИСАНИЕ [hang-up, starvation]. 1. Состояние *вычислительной системы*, при котором она перестает выдавать результаты и реагировать на запросы извне, например, на нажатие пользователем *персонального компьютера* клавиш <Ctrl> и <Break>, или на другие *внешние прерывания*. З. может быть вызвано *аппаратным сбоем*, ошибками в *программном обеспечении* или попыткой обращения *прикладной программы* к устройству, не подключенному к *центральному процессору*. 2. Состояние, характеризующееся аномально большим временем ожидания задачи или процесса при обращении к какому-либо ресурсу

ЗАГЛАВНАЯ СТРАНИЦА [home page]. То же, что *домашняя страница*

ЗАГЛУШКА [module stub]. Фиктивная подпрограмма, имитирующая одну или несколько функций временно отсутствующего модуля разрабатываемой программы. Например, З. применяются при *модульном программировании,* для того чтобы можно было проводить компиляцию, тестирование и отладку незаконченной программы, не дожидаясь, когда некоторые отсутствующие в ней модули будут разработаны в надлежащем виде и испытаны. В зависимости от функций заменяемого модуля З. может выдавать либо правдоподобный результат (например, значения из таблицы), либо некоторое сообщение (например, фиксирующее вызов отсутствующей подпрограммы)

ЗАГОЛОВОК ПРОЦЕДУРЫ [procedure header]. Составная часть *описания процедуры*, в которой указываются имя процедуры и *список формальных параметров*, если они есть. Как правило, З. п. начинается *ключевым словом* `procedure`, за которым последовательно пишутся имя и список. Например, З. п. решения квадратного уравнения $ax^2 + bx + c = 0$ в языке Паскаль может иметь вид

```
procedure Quadratic_equation (a, b, c : Real, var x1, x2: Real, i: Integer);
```

Здесь указано, что процедура названа `Quadratic_equation` и имеет шесть формальных параметров. Первые пять из них *вещественного типа* соответствуют заданным трем коэффициентам уравнения и двум искомым величинам. Шестой параметр *целого типа* служит для указания, являются ли корни уравнения вещественными. Если `i = 1`, то `x1` и `x2` — два вещественных корня, если `i = 0`, то `x1` и `x2` — соответственно вещественная и мнимая части комплексно сопряженных корней. С помощью ключевого слова `var` указывается способ *передачи параметров*: первые три параметра передаются по значению, а последние три — по наименованию. См. *блок-схема программы* (см. рис. Б.4)

ЗАГОЛОВОК ФУНКЦИИ [function header]. Составная часть *описания функции*, в которой указываются имя функции и *список формальных параметров*, если они есть. Как правило, З. ф. начинается *ключевым словом* `function`, за которым последовательно пишутся имя и список. Например, З. ф., вычисляющей площадь треугольника по заданным длинам трех его сторон на языке Паскаль, может иметь вид

```
function Area (a, b, c: Real, var i: Integer): Real;
```

Здесь указано, что функция названа `Area` (площадь) и что она имеет четыре формальных параметра. Первые три из них имеют *вещественный тип* и соответствуют заданным длинам сторон. Четвертый параметр имеет *целый тип* и служит для контроля за правильностью полученного результата. Например, `i = 1`, если треугольник с заданными длинами сторон существует. Если же треугольник не существует или длины сторон заданы неверно, то `i = 0`. См. *испытание программы*

ЗАГОЛОВОК ЦИКЛА [loop header]. *Синтаксическая конструкция языка программирования*, указывающая начало либо начало и конец *тела цикла* и определяющая правила его повторения. З. ц. определяется конструкцией соответствующих предложений. В языке Фортран З. ц. является оператор `DO`, стоящий перед телом цикла. См. *оператор цикла языка Фортран*. В языке Паскаль в *цикле с предусловием* З. ц. является часть *оператора цикла* `while <логическое выражение> do`. В *цикле с параметром* — конструкция `for <управляющая переменная> := <начальное выражение> to <конечное выражение> do`. В цикле с постусловием З. ц. состоит из двух частей, разделенных телом цикла: *ключевого слова* `repeat`, стоящего перед телом цикла, и

конструкции until <логическое выражение>, стоящей после тела цикла. См. *операторы цикла языка Паскаль*. В языке Си в цикле с предусловием З. ц. является часть оператора цикла while (<выражение>). В цикле с параметром — конструкция for (<инициализирующее выражение>; <проверяемое выражение>; <корректирующее выражение>). В цикле с постусловием З. ц. состоит из двух частей, разделенных телом цикла: ключевого слова do, стоящего перед телом цикла, и конструкции while (<выражение>), стоящей после тела цикла. См. *операторы цикла языка Си*. Соответствующие примеры приведены в табл. 3.1

Таблица 3.1. *Примеры операторов цикла*

Язык	Цикл	Заголовок цикла
Фортран	DO 10 I = 1, N IF (A[I]<AMIN) GOTO 20 10 CONTINUE	DO 10 I = 1, N
Паскаль	while Abs(dx) > eps do begin dx:= (a/x – x) * 0.5; x:= x + dx end;	while Abs(dx) > eps do
	for l:= 10 downto k+1 do x[l]:= x[l] – x[k];	for l:= 10 downto k+1 do

	repeat
	dx:= (a/x – x) * 0.5;
	x:= x + dx
	until Abs(dx) < eps;
		Заголовок состоит из двух частей: repeat и until Abs(dx) < eps

Си	while (summa<1000){
	summa = summa + step;
	step = 2 * step
	};
		while (summa<1000)

	do {ch:=getchar(); putchar(ch);} while(ch!= '\n');	Заголовок состоит из двух частей: do и while(ch!= '\n');

	for (i = 1; i < = 10; i++){
	c = a[i]; a[i] = b[i]; b[i] = c
	}
		for (i = 1; i <= 10; i++)

ЗАГОЛОВОЧНЫЙ ТЕКСТ, **фигурный текст [artistic text, word art]**. В *настольных редакционно-издательских системах* — тип *текста*, который сочетает характеристики текста и графического объекта. В качестве графического объекта З. т. можно подвергать всем доступным трансформациям (рис. З.1). Ограничения связаны с объемом (З. т. может содержать не более 32 000 символов) и отсутствием возможности автоматической верстки. См. *верстка страниц*

Рис. З.1. Заголовочный текст, созданный программой Microsoft WordArt

ЗАГРУЖАЕМЫЙ ДРАЙВЕР [loadable driver]. *Драйвер*, не являющийся *резидентной программой операционной системы*, т. е. не находящийся постоянно в *оперативной памяти*. Он хранится во *внешнем запоминающем устройстве* (обычно на *жестком диске*) и при необходимости загружается в оперативную память. Для этого он должен быть указан в *файле конфигурации* операционной системы. Как правило, загружаемыми являются драйверы, дополняющие *систему ввода/вывода*. Это позволяет, не затрагивая *системные модули*, расширять возможности операционной системы, делая возможным подключение к *вычислительной системе* новых устройств или нестандартное использование имеющихся. Например, для подключения к компьютеру дисковода *лазерных дисков* (CD-ROM) или организации *электронного диска* нужно указать имена программ-драйверов этих устройств в файле конфигурации операционной системы

ЗАГРУЖАЕМЫЙ ШРИФТ [downloadable font]. *Шрифт*, который хранится на диске и может при необходимости быть загружен в память принтера для распечатки документа. З. ш. бывают различных начертаний и размеров, чаще всего они используются в *лазерных принтерах* и других принтерах постраничной печати. С некоторыми из З. ш. могут работать и *матричные принтеры*

ЗАГРУЗКА [loading]. Считывание данных из одного *запоминающего устройства* в другое, обычно обладающее большей оперативностью. Чаще всего этот термин обозначает копирование программ и данных из *внешнего запоминающего устройства* в *оперативную память*. См. *загрузка программы, загрузка операционной системы*

ЗАГРУЗКА ОПЕРАЦИОННОЙ СИСТЕМЫ [operating system loading]. Считывание *операционной системы* из *внешнего запоминающего устройства* в *опе-*

ративную память, ее настройка и запуск. Обычно З. о. с. осуществляется специальной программой, хранящейся в *постоянном запоминающем устройстве* компьютера. Эта программа сначала производит тестирование устройств компьютера, а затем считывает из внешнего запоминающего устройства в оперативную память — программу загрузчик основных модулей операционной системы и передает ей управление. После того как основные модули, составляющие *ядро операционной системы,* окажутся в оперативной памяти, дальнейшая загрузка и настройка составных частей системы могут осуществляться по мере считывания остальных ее модулей. При этом управление загрузкой может передаваться от одного считанного модуля к другому. Например, по аналогичной схеме происходит З. о. с. Windows, при этом вид экрана отражает этапы загрузки. Пока работает загрузчик, экран черный. После того как ядро операционной системы загружено, экран становится синим. После того как загружена оболочка операционной системы, экран приобретает цвет фона *рабочего стола*

ЗАГРУЗКА ПО ЛИНИИ СВЯЗИ, **скачивание [downloading]**. Процесс передачи по *линии связи* копии файла из удаленного компьютера в компьютер, выдавший запрос. Местный компьютер через свой модем выдает запрос и поручает удаленному компьютеру начать *передачу данных*, а затем сохраняет поступивший файл на диске. Термин З. п. л. с. употребляется также для обозначения передачи блока данных в подчиненное устройство, например, файла в принтер. Ср. *выгрузка по линии связи*

ЗАГРУЗКА ПРОГРАММЫ [program loading]. Считывание программы из *внешнего запоминающего устройства* в *оперативную память*, *настройка адресов* и, возможно, запуск. З. п. выполняется под управлением загрузчика — специальной программы, обычно входящей в состав *операционной системы*. Если в процессе загрузки программа записывается на свободное место оперативной памяти, то производится настройка указанных в программе *логических адресов*, которые в зависимости от места в оперативной памяти, занимаемого программой и отводимого ее данным, преобразуются в *физические адреса*. До момента запуска программа может загружаться целиком, а может происходить загрузка части программы. Тогда отсутствующие в оперативной памяти *программные модули* загружаются по мере обращения к ним. См. *динамическая загрузка, библиотека динамической компоновки*. Ср. *оверлейная структура*

ЗАГРУЗОЧНЫЙ МОДУЛЬ, образ задачи [load module]. *Программный модуль* в виде, пригодном для *загрузки* и выполнения. З. м. получается из *объектного модуля* при *редактировании связей* и представляет собой программу в виде последовательности *машинных команд*, имеющую принятый для данной *операционной системы* формат. З. м. обычно выполнен в *относительных* (иногда в *абсолютных*) *адресах* и содержит информацию для их настройки

загрузчиком в соответствии с отведенной для него областью *оперативной памяти*. З. м. являются *исполняемые файлы*

ЗАГРУЗЧИК [loader]. *Программа*, управляющая *загрузкой*. Обычно З. выполняет следующие основные функции: 1) распределяет пространство оперативной памяти для программы; 2) настраивает адреса подготовленной компилятором и компоновщиком *объектной программы*, заменяя все *относительные адреса* соответствующими адресами фактически распределенной памяти; 3) физически размещает подготовленную версию машинного кода программы в памяти и передает управление на первую команду программы. Существуют программы, сразу выполняющие и компоновку, и загрузку. См. *компонующий загрузчик*

ЗАДАНИЕ [job]. Совокупность программ и их данных, обрабатываемая *операционной системой* как единое целое в режиме *пакетной обработки*. Содержание З. определяется пользователем в виде последовательности *управляющих операторов*, определяющих выполняемые программы и используемые ими данные. Ср. *задача, процесс*

ЗАДАНИЕ НА ПЕЧАТЬ [print job]. Исходный код, который содержит как данные, так и команды для обработки в процессе печати. См. *драйвер принтера*, *спулер печати, очередь заданий на печать*

ЗАДАЧА [task]. Задание *вычислительной системе*, представленное в виде программы или части программы и данных, которое *операционная система* рассматривает как единое целое при распределении ресурсов. В ряде операционных систем термин "З." совпадает с понятием "процесс", в других — с понятием "задание". Ср. *задание, процесс*

ЗАИНТЕРЕСОВАННЫЕ ЛИЦА [stockholders]. Термин, которым обозначают группу лиц, которые так или иначе заинтересованы в успехе проекта. Например, для проекта по разработке *программного продукта* в число З. л. включаются заказчики, инвесторы, пользователи и разработчики

ЗАКЛАДКА [bookmark]. Место в тексте или выделенный фрагмент в документе, которому присвоено имя. З. используются во многих текстовых процессорах, например в Microsoft Word, а также в других приложениях. Имя З. может применяться для перехода к помеченному месту документа, для создания и нумерации *перекрестных ссылок* и пр.

ЗАКЛЮЧИТЕЛЬНОЕ СОСТОЯНИЕ [final state]. Особое состояние *конечного автомата*, переход в которое указывает на завершение выполнения этого автомата

ЗАКОММЕНТИРОВАТЬ [comment out]. Временно отменить действие одного или нескольких операторов программы, превратив их в *комментарий*. Для этого достаточно ограничить эти операторы в соответствии с синтаксисом комментария, предусмотренным *языком программирования*. Например, в

языке Паскаль нужно заключить исключаемый оператор или последовательность операторов в скобки `{...}` или `(*...*)`. Таким приемом часто пользуются для локализации ошибки при отладке программы. Чтобы восстановить первоначальный вид программы, достаточно убрать ограничители

ЗАКОН АМДАЛА [Amdal's law]. Коэффициент ускорения K, показывающий во сколько раз быстрее программа выполняется на параллельной машине, чем на последовательной; определяется формулой:

$$K = \frac{1}{S + P/N},$$

где S и P — относительные доли времени исполнения соответственно последовательной и параллельной частей программы на последовательной машине ($S + P = 1$), N — количество независимых ветвей параллельной части, распределенных по одной на N процессоров. Из З. А. следует, что для программ с небольшой долей параллелизма использование большого числа процессоров не дает сколько-нибудь значительного выигрыша в быстродействии. З. А. не учитывает время обмена данными, которое увеличивается при увеличении числа процессоров.

ЗАКРАСКА [filling]. В *компьютерной графике* — изменение цвета пикселов, расположенных внутри изображенного на экране графического объекта. Ср. *заливка*

ЗАКРЫТИЕ ФАЙЛА [closing of a file, file closing]. Операция завершения работы программы с файлом. При ее выполнении все связанные с файлом области *буферной памяти* очищаются, а информация о произведенных изменениях данных заносится во *внешнюю память* (обычно на диск). Противоп. *открытие файла*. См. *закрытый файл*

ЗАКРЫТОЕ СВОЙСТВО КЛАССА, приватное свойство класса [private class property, private property]. *Свойство класса*, *область видимости* которого распространяется только на определение *класса*. В большинстве языков *объектно-ориентированного программирования* в описании такого свойства используется ключевое слово `public`. Ср. *защищенное свойство класса*, *открытое свойство класса*

ЗАКРЫТЫЙ МЕТОД КЛАССА, приватный метод класса [private class method, private method]. *Метод класса*, *область видимости* которого распространяется только на определение *класса*. В большинстве языков *объектно-ориентированного программирования* в описании такого метода используется ключевое слово `public`. Ср. *защищенный метод класса*, *открытый метод класса*

ЗАКРЫТЫЙ ФАЙЛ [closed file]. Файл, над которым выполнена операция *закрытия файла*. Чтобы записи З. ф. снова стали доступны для чтения и обработки, его надо открыть. Противоп. *открытый файл*

ЗАЛИВКА [flood filing, fill, paint bucket]. 1. В *компьютерной графике* — однотонная закраска изображенного на экране плоского графического объекта, ограниченного четким контуром. Ср. *закраска*. 2. **[paint bucket]**. *Инструмент машинной графики*, выполняющий З. *основным цветом* выделенной контуром области экрана. На рис. 3.2 для *графического редактора* Adobe Photoshop показаны кнопка вызова З. и диалоговое окно, позволяющее установить параметры инструмента

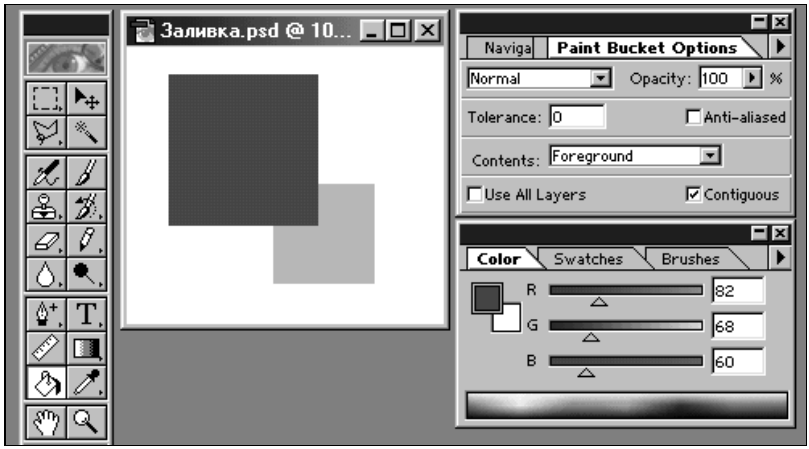

Рис. 3.2. Нажатая кнопка вызова, образец работы и диалоговые окна инструмента **Заливка** графического редактора Adobe Photoshop

ЗАМОК [lock]. 1. *Ключ защиты памяти*, определенная *структура данных* или программа, используемые для управления доступом к информации. 2. Операция, позволяющая только одному процессу иметь доступ к определенному ресурсу. См. *блокировка*

ЗАПИСЬ [writing, record]. 1. Процесс запоминания (фиксирования) данных в *запоминающей среде* или на *носителе данных* (writing). 2. Результат процесса записи (record). 3. Единица обмена данными между программой и *внешней памятью*, т. е. совокупность данных, совместно пересылаемых на *периферийные устройства* или с периферийных устройств одним *оператором ввода/вывода*. Поэтому файлы, хранимые во внешнем ЗУ, часто представляют собой набор З. Так, файл данных задачи, которые должны вводиться порциями в процессе ее выполнения, состоит из набора З. Каждая из таких З. содержит определенную порцию данных. Например, файл данных о служащих фирмы состоит из З., содержащих информацию о каждом служащем. Существует различие между совокупностью логически связанных данных и их физическим представлением в запоминающих устройствах. Как то, так и другое называют З., а чтобы различать эти понятия, обычно в первом случае

употребляют термин *"логическая З."*, а во втором — *"физическая З."*, которая представляет собой *блок данных* и на самом деле может состоять из нескольких логических З. 4. В *базах данных* — структурированная совокупность данных, составляющие которой, обычно называемые полями, могут иметь имя и значение. При этом допускается выполнение действий как над всей З., так и над ее полями. См. *кортеж*. 5. В *языках программирования высокого уровня* (например, в Паскале) — *структурированный тип* данных, компоненты значений которого, обычно называемые полями, могут иметь имя и принадлежать к различным *типам данных*. В таких языках допускается выполнение действий как над всей З., так и над ее полями. См. *тип записи, поле данных*

ЗАПЛАТА [patch]. Последовательность операторов или команд, добавленная к коду готовой программы для исправления имеющихся недостатков. Обычно добавляемая последовательность вносится в виде отдельного блока или файла, а в нужном месте программы помещается команда перехода. Часто З. является средством добавления новой функции к существующей версии программы до выхода новой версии, где эта функция будет вставлена обычным путем

ЗАПОМИНАЮЩАЯ СРЕДА [storage medium]. 1. Материал, обладающий физическими свойствами, позволяющими записывать, хранить и считывать информацию. Например, магнитный состав, которым покрывают *магнитные диски*. 2. Совокупность однотипных *носителей данных*, входящих в состав *запоминающего устройства*. Например, носители данных, составляющие *оперативную память*, обычно называют З. с.

ЗАПОМИНАЮЩЕЕ УСТРОЙСТВО (ЗУ) [storage device]. Техническое устройство, реализующее память. Это любое устройство, способное записывать данные и сохранять их для последующего считывания. ЗУ состоят из *запоминающей среды* или *носителя данных* и технических средств, осуществляющих запись информации в среду или на носитель и ее считывание. Например, ЗУ на *гибких магнитных дисках* состоит из дискет и дисковода. В компьютерах применяются разнообразные ЗУ с различными емкостями и скоростями работы — от ЗУ на полупроводниковых кристаллах со *временем выборки*, измеряемым в долях наносекунд (10^{-9} с) до устройств на *магнитных лентах* со временем выборки, измеряемым в секундах. Диапазон емкости различных устройств также широк. От 16-битовых регистров процессора до винчестеров, способных хранить несколько гигабайт. Существуют *постоянные запоминающие устройства*, предназначенные только для считывания

ЗАПРОС [request, query]. Обращение с целью получения ответа. Это может быть сигнал, с помощью которого *операционная система* запрашивает *периферийные устройства* об их готовности к *обмену данными*; обращение пользователя к *базе данных*, содержащее задание на поиск, чтение и обработку

нужной информации; обращение программы к *устройству ввода* с целью получения данных

ЗАПУСК ПРОГРАММЫ [program start]. *Загрузка программы* в *оперативную память* и *передача управления* команде программы, выполняемой первой

ЗАРЕЗЕРВИРОВАННОЕ СЛОВО [reserved word]. То же, что *ключевое слово*

ЗАХВАТ, **монопольное использование [lockout]**. Использование данного ресурса (файла, области памяти, порта ввода/вывода и т. п.) одним процессом с запрещением обращения к этому ресурсу других процессов. З. является способом организации контролируемого доступа к совместно используемому ресурсу, гарантирующим, что в данный момент только одна программа работает с этим ресурсом. См. *захват записи*. Ср. *блокировка*

ЗАХВАТ ЗАПИСИ, **блокировка записи [record locking]**. Монопольное использование записи базы данных одной прикладной программой. При этом для любой другой программы запись становится недоступной или доступной только для чтения. Ср. *блокировка данных*

ЗАЦИКЛИВАНИЕ [loophole]. То же, что *зацикливание программы*

ЗАЦИКЛИВАНИЕ ПРОГРАММЫ, **зацикливание [cycling**, **loophole]**. Бесконечное повторение одной и той же последовательности команд в ходе выполнения программы, которое может быть прекращено только принудительным прерыванием извне (*оператором ЭВМ* или таймером по истечении отведенного программе ресурса времени). Чаще всего З. п. является следствием неправильной *передачи управления* или не предусмотренного программистом *бесконечного цикла*. З. п. может явиться результатом как *синтаксической ошибки*, так и *логической ошибки*, возникающей обычно из-за невозможности предусмотреть все ситуации, которые могут возникнуть, и приводящей к повтору одних и тех же действий

ЗАШИТЫЕ ПРОГРАММЫ [firmware]. То же, что *встроенные программы*

ЗАЩИТА ДАННЫХ [data protection]. Меры, направленные против *несанкционированного доступа* к информации, хранящейся в памяти компьютера — так называемого компьютерного браконьерства. Одним из способов З. д., хранящихся в *вычислительной системе*, является использование символьных *паролей* или *цифровых кодов*. В этом случае вычислительная система допускает к информации только пользователей, знающих пароль. Кроме того, в памяти может храниться специальная таблица, в которой перечислены права каждого пользователя на доступ к тем или иным данным. В соответствии с этой таблицей разрешается (или не разрешается) доступ конкретному пользователю к определенным данным. При этом может производиться регистрация всех попыток и несанкционированного доступа. Для большей надежности информацию можно хранить в зашифрованном виде. Для этого применяются специальные шифры (тайнопись, криптография). Ср. *защита*

памяти. См. *аппаратный ключ, защита от копирования, защита файлов, права доступа*

ЗАЩИТА ОТ КОПИРОВАНИЯ [copy protection]. Меры, предотвращающие копирование *программного продукта* с целью его распространения или использования без согласия владельца авторских прав. З. о. к. может обеспечивать не только некопируемость дискеты дистрибутива при помощи стандартного дисковода, но и невозможность применения стандартных средств *отладки* или *трассировки программ* с целью изучения логики работы защищенной программы. З. о. к. реализуется как *программными*, так и *аппаратными средствами.* Чтобы предотвратить копирование дистрибутивной дискеты, часто применяются нестандартные нумерация и форматирование ее дорожек. Кроме того, программный продукт обычно содержит закодированную информацию (ключ), которая теряется при попытке копирования или отладки программы стандартными средствами. После этого программа становится непригодной к использованию. Существуют специально разработанные различными фирмами системы З. о. к. Некоторые авторы дополняют систему защиты блоком ответной реакции, который производит различные карательные действия при попытке использования нелегальной копии. См. *аппаратный ключ, пиратство*

ЗАЩИТА ПАМЯТИ [memory protection]. Меры, не позволяющие выполняемой задаче получить доступ к "чужой" памяти, т. е. к памяти, которая за ней не закреплена. При обращении к памяти осуществляется проверка принадлежности адреса, по которому происходит обращение, к *адресному пространству задачи.* Широко распространен способ З. п., когда программе присваивается *ключ защиты*, который должен совпадать с *ключами защиты памяти* всех *блоков памяти*, выделенных данной программе. Эти меры предотвращают порчу хранящихся в памяти программ и данных другими программами. З. п. реализуется как *аппаратными*, так и *программными средствами.* Для *оперативной памяти* защита реализуется аппаратно, чтобы поддерживать ее высокое быстродействие; для более медленных *запоминающих устройств* З. п. может быть полностью реализована программно. В настоящее время разработаны различные механизмы контроля за доступом к какой-либо области памяти с учетом допустимых режимов обращений. Широко распространен такой вид аппаратной З. п., как использование ограничительных регистров, содержимое которых обозначает границы *области памяти* с определенной формой доступа. Например, область, предназначенная для хранения подпрограмм общего использования, может быть отмечена *операционной системой*, как доступная только для исполнения, и данные из нее будут только считываться либо как команды, либо как операнды исполняемых команд. Для защиты *виртуальной памяти* также применяются дескрипторы страниц или сегментов. Нарушение системы З. п. исполняемой программой обычно вызывает ее *аварийное завершение.* Ср. *защита данных*

ЗАЩИТА ФАЙЛА [file security]. Меры, предотвращающие внесение изменений в файл или несанкционированное чтение данных из файла или, в случае *программных файлов*, их ошибочное или несанкционированное исполнение. З. ф. чаще всего реализуется *программными средствами*. Для этого в дескрипторе файла указываются операции над файлом, разрешенные для различных пользователей. См. *защита данных, защита от копирования, права доступа*

ЗАЩИЩЕННОЕ СВОЙСТВО КЛАССА [protected class property, protected property]. *Свойство класса*, *область видимости* которого распространяется на определение данного *класса* и на все классы, *наследующие* данному классу. В большинстве языков *объектно-ориентированного программирования* в описании такого свойства используется ключевое слово `protected`. Ср. *закрытое свойство класса*, *открытое свойство класса*

ЗАЩИЩЕННЫЕ ЭЛЕКТРОННЫЕ ТРАНЗАКЦИИ, протокол SET [secure electronic transaction (SET)]. Название *протокола* защиты платежей по кредитным картам в общедоступных сетях, таких как Интернет. Разработан компаниями Visa и MasterCard в 1996 г. Используется также при заключении торговых сделок через Интернет. Обеспечивает конфиденциальность финансовой информации и аутентификацию участников сделки

ЗАЩИЩЕННЫЙ МЕТОД КЛАССА [protected class method, protected method]. *Метод класса*, *область видимости* которого распространяется на определение данного *класса* и на все классы, *наследующие* данному классу. В большинстве языков *объектно-ориентированного программирования* в описании такого метода используется ключевое слово `protected`. Ср. *закрытый метод класса*, *открытый метод класса*

ЗАЩИЩЕННЫЙ РЕЖИМ [protected mode]. Основной режим работы процессоров *персональных IBM-совместимых компьютеров*, при котором программе доступны только определенные области *оперативной памяти*, а области памяти, предназначенные для специальных целей, защищены. В этом режиме можно полноценно использовать всю оперативную память. Возможно мультипрограммирование. В других ситуациях в настоящее время этот термин не применяется

ЗВЕЗДООБРАЗНАЯ СЕТЬ [star network]. То же, что *радиальная сеть*

ЗВОНОК [call]. Временное соединение двух и более абонентов *телефонной сети*

ЗВУКОВАЯ ДОРОЖКА [audio track, sound track]. *Дорожка*, на которой записан звук

ЗВУКОВАЯ КАРТА [sound card]. То же, что *звуковая плата*

ЗВУКОВАЯ КОЛОНКА [speaker]. Акустическая аппаратура, присоединяемая к *звуковой карте* для воспроизведения аналогового звукового сигнала. Как правило, к звуковой карте присоединяются две З. к. для создания стереоэффекта. Возможно присоединение к звуковой карте компьютера обычных устройств воспроизведения звука, применяемых в бытовой аудиотехнике, но чаще используют З. к., специально сконструированные для присоединения к компьютеру. См. *мультимедиа*

ЗВУКОВАЯ ПЛАТА, **звуковой адаптер**, **звуковая карта [sound board**, **sound card]**. *Плата*, позволяющая с помощью компьютера записывать и воспроизводить звуки, синтезировать музыку, управлять внешней акустической аппаратурой, подключенной к компьютеру. Все З. п. по назначению можно разделить на три группы: чисто звуковые, чисто музыкальные и комбинированные. Чисто звуковые позволяют только записывать или воспроизводить непрерывный звуковой поток, наподобие магнитофона. Вся работа по запоминанию записываемого и подготовке воспроизводимого потока возлагается на *программное обеспечение*. Некоторые платы имеют встроенные сигнальные процессоры для обработки звука в процессе его записи или воспроизведения. Чисто музыкальные содержат только *музыкальный синтезатор*. Такие платы ориентированы прежде всего на генерацию относительно коротких музыкальных звуков по командам от *центрального процессора*; при этом либо звуки создаются заново, либо воспроизводятся звуки, цифровые коды которых были заранее помещены в память синтезатора. Музыкальные платы не имеют возможности записи звука и не рассчитаны на воспроизведение непрерывного звукового потока. Некоторые музыкальные платы содержат эффект-процессор для обработки создаваемого звука. Комбинированные, или звукомузыкальные З. п. располагают блоком цифровой записи/воспроизведения, называемым также цифровым каналом или трактом, блоком музыкального синтезатора и блоками, осуществляющими регулирование уровней, коммутацию, усиление и сведение звуковых сигналов. См. *мультимедиа, звуковая колонка*

ЗВУКОВОЙ АДАПТЕР [sound board, **sound card]**. То же, что *звуковая плата*

ЗЕРКАЛИРОВАНИЕ [mirroring]. То же, что *зеркальное копирование*

ЗЕРКАЛЬНОЕ КОПИРОВАНИЕ, **зеркалирование [mirroring]**. Одновременная запись одних и тех же данных на несколько дисков или лент. Для З. к. требуется минимум два диска со своими адаптерами. Этот прием является разновидностью *технологии RAID* и применяется с целью повышения надежности *вычислительной системы*

ЗЕРКАЛЬНОЕ ОТРАЖЕНИЕ ПО ВЕРТИКАЛИ, **отражение сверху вниз [flip vertical]**. *Команда графического редактора*, выполняющая на экране вертикальное отражение рисунка или выделенной области (рис. З.3)

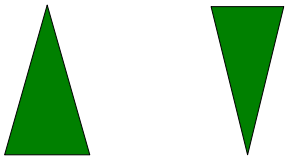

Рис. З.3. Зеркальное отражение по вертикали:
слева — исходная фигура, справа — фигура после исполнения команды

ЗЕРКАЛЬНОЕ ОТРАЖЕНИЕ ПО ГОРИЗОНТАЛИ, **отражение слева направо [flip horizontal]**. *Команда графического редактора*, выполняющая на экране горизонтальное отражение рисунка или выделенной области (рис. З.4)

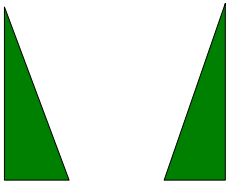

Рис. З.4. Зеркальное отражение по горизонтали:
слева — исходная фигура, справа — фигура после исполнения команды

ЗЕРКАЛЬНЫЙ НАБОР [mirror set]. Два раздела на разных дисках, один из которых используется в качестве копии другого. См. *зеркальное копирование*

ЗНАК [character, **symbol]**. Отдельный символ алфавита или элемент системы символов, используемой в *вычислительной системе*. См. *символ, знак арифметической операции, знак логической операции*

ЗНАК АРИФМЕТИЧЕСКОЙ ОПЕРАЦИИ [arithmetic operation character, **arithmetic operator]**. *Знак*, определяющий *арифметическую операцию*. Как правило, это знаки +, –, *, /

ЗНАК ЛОГИЧЕСКОЙ ОПЕРАЦИИ [logical operation character, **logical operator]**. Символ, определяющий *логическую операцию*. Как правило, это символы: "НЕ", "NOT", ¬ — для операции отрицания; "И", "AND", ∧ — для логического "И"; "ИЛИ", "OR", ∨ — для логического "ИЛИ"; "ЭКВИВАЛЕНТНО", ≡ — для операции эквивалентности. См. *логические операции*

ЗНАКОГЕНЕРАТОР [character generator]. Составная часть *адаптера дисплея*, преобразующая коды символов в их графическое изображение на экране дисплея. З. используется в *текстовом режиме* работы адаптера дисплея. В З. имеется таблица кодов и соответствующих им изображений символов, занимающих одно знакоместо на экране. По поступившему из видеопамяти коду

З. "выбирает" из таблицы символ, изображение которого затем формируется на экране. Программист может программировать З., изменяя в таблице изображения символов программным способом. Например, изменение шрифтов при работе в *текстовом редакторе* осуществляется путем смены таблиц З. В других ситуациях в настоящее время этот термин не используется

ЗНАКОМЕСТО [font reticle]. Прямоугольный участок поверхности экрана дисплея или бумаги, в котором размещается один выводимый *знак*. См. *текстовый режим*

ЗНАЧАЩАЯ ЦИФРА, **значащий разряд [significant digit]**. Любая цифра числа, записанного в позиционной системе счисления, начиная с первой слева ненулевой цифры. Это означает, что значащей называется любая цифра, отличная от нуля, и ноль, если он не стоит в числе до его первой отличной от нуля цифры. Например, в числе 0.012030 З. ц. являются последние пять цифр. Первые два нуля не являются З. ц. Они сами по себе не представляют количественную величину, а служат для определения разрядов других цифр (при записи этого же числа в виде $1.2030 \cdot 10^{-2}$ они отсутствуют). Остальные два нуля показывают, что в этих разрядах числа содержится именно 0, а не 1 и не 2 и т. д. В случае, если в данном числе последний ноль не является значащим, то оно должно быть записано в виде 0.01203. При написании больших целых чисел, например, числа 456000, нули справа могут служить как для обозначения З. ц., так и для определения разрядов остальных цифр. Чтобы избежать этой неопределенности, указанное число следует записать в виде $4.56 \cdot 10^5$, если оно имеет три З. ц., или $4.5600 \cdot 10^5$, если оно имеет пять З. ц. В случае приближенного задания числа не все З. ц. могут быть верными. См. *верные значащие цифры*

ЗНАЧАЩИЙ РАЗРЯД [significant digit]. То же, что *значащая цифра*

ЗНАЧЕНИЕ [value]. Смысл или величина, содержащаяся в *элементе данных*

ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ [default value]. Стандартное *значение*, присваиваемое параметру автоматически, если пользователь не задал ему иное допустимое значение

ЗНАЧЕНИЕ ТИПА ДАННЫХ [data value]. Экземпляр *типа данных*. Значение, не имеющее индивидуальности, в отличие от *экземпляра класса*

ЗНАЧОК, **пиктограмма**, **иконка [icon]**. Условное изображение программы, операции или информационного объекта на экране. Обычно З. нарисованы на *кнопках* или обозначают *пункты меню*. См. *графический интерфейс пользователя, пиктографическое меню*

ЗОНА [region, **band]**. 1. Область *оперативной памяти*, динамически выделяемая заданию или системной задаче в требуемых размерах независимо от других заданий и задач. См. *динамическое распределение памяти*. 2. Группа соседних дорожек на *магнитном* или *лазерном диске*

И

ИГРОВАЯ ПРОГРАММА [game program]. *Программа*, задающая алгоритм *компьютерной игры*

ИДЕНТИФИКАТОР [identifier]. Строка символов, предназначенная для обозначения *объекта программы* или *вычислительной системы*. В традиционных *языках программирования* И. употребляются в форме не содержащей пробелов последовательности букв латинского алфавита и цифр, начинающейся с буквы, например `X`, `y12`, `Summa`. В некоторых языках используются и другие знаки, например, `#`, `&`, `$`, `_`, а также русские буквы. При этом вместо слова "И." часто употребляется слово "имя" и устанавливается ограничение на длину И. См. *имя*

ИДЕНТИФИКАТОР БЕЗОПАСНОСТИ, **идентификатор защиты**, **идентификатор доступа [security identifier**, **security ID (SID)]**. *Идентификатор*, однозначно определяющий пользователя для *системы безопасности операционной системы Windows NT*. И. б. представляет собой структурированную запись переменной длины, идентифицирующую как самого пользователя, так и группы, к которым он принадлежит. Кроме того, И. б. может содержать информацию о привилегиях пользователя

ИДЕНТИФИКАТОР ДОСТУПА [security identifier, **security ID (SID)]**. То же, что *идентификатор безопасности*

ИДЕНТИФИКАТОР ЗАЩИТЫ [security identifier, **security ID (SID)]**. То же, что *идентификатор безопасности*

ИЕРАРХИЧЕСКАЯ БАЗА ДАННЫХ [hierarchical database]. *База данных*, организованная по иерархической *модели данных*. И. б. д. состоит из записей, которые связаны между собой явным образом, причем способ связывания строго ограничен, а именно, записи данных в иерархической базе данных образуют *иерархию*. Например, *реестр Windows* является И. б. д. И. б. д. имеет более простую структуру, а потому допускает наиболее эффективную реализацию по сравнению с другими базами данных. Ср. *реляционная база данных, сетевая база данных*

ИЕРАРХИЧЕСКАЯ СУБД [hierarchical database management system (HDBMS)]. То же, что *система управления иерархическими базами данных*

ИЕРАРХИЯ [hierarchy]. Частный случай отношения частичного порядка на множестве объектов, при котором для всех объектов, кроме одного, имеется некоторое количество (ноль или более) объектов, непосредственно следующих за данным объектом, и ровно один объект, непосредственно предшествующий данному объекту. Для одного выделенного объекта, который называется вершиной иерархии, нет предшествующего объекта. Отношения И. очень широко распространены как в обыденной жизни, так и в компьютер-

ной сфере. Например, И. являются следующие отношения в обыденной жизни: прямое наследование по мужской линии, субординация по службе, территориально-административное деление государства. Примеры И. в программировании: *дерево каталогов*, структура *вложенных блоков*, *оверлейная структура,* иерархия *наследования*. Графически И. обычно изображают в виде дерева (рис. И.1). Если в данной И. для двух объектов А и В либо А непосредственно предшествует В, либо существуют промежуточные объекты C_1, C_2, C_N, такие, что А непосредственно предшествует C_1, C_1 — предшествует C_2 и т. д., C_N непосредственно предшествует В, то говорят, что объект В является *потомком* объекта А, а объект А — *предком* объекта В. Например, на рис. И.1 объект "Директор" является предком всех объектов, объект "Служащий А" является потомком объекта "Начальник подразделения 1", но не является потомком объекта "Начальник подразделения 2"

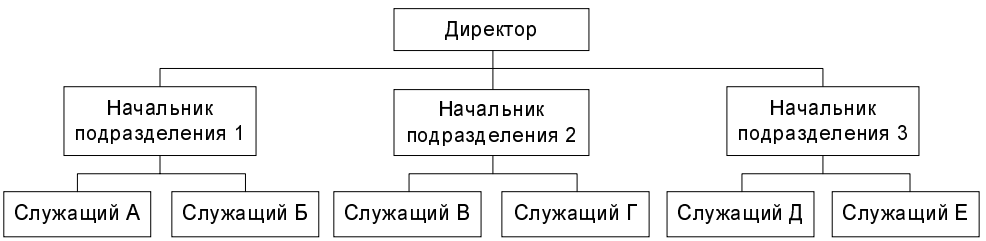

Рис. И.1. Организационная диаграмма как пример дерева иерархии

ИЕРАРХИЯ ПАМЯТИ [memory hierarchy, **storage hierarchy]**. Многоуровневая организация памяти *вычислительной системы*, при которой к низшему уровню памяти относятся *периферийные запоминающие устройства* с малым быстродействием *(магнитные ленты)*, к среднему уровню — *внешние устройства*, составляющие основной *объем памяти (магнитные диски)* и к высшему уровню — быстродействующие устройства *оперативной памяти* и внутренние регистры процессора. И. п. лежит в основе создания иерархической памяти вычислительной системы как совокупности взаимосвязанных запоминающих устройств, одни из которых имеют большое быстродействие, но малую емкость, а другие — большую емкость, но и большое время доступа. Это позволяет оптимизировать распределение ресурсов как между задачами, так и в процессе их выполнения. При этом *операционная система* или *аппаратные средства* перемещают *блоки данных* между уровнями иерархической памяти без специальных указаний *прикладной программы*, делая для нее И. п. незаметной

ИЗБЫТОЧНОСТЬ [redundancy]. Наличие в вычислительной системе дополнительных данных, программных и аппаратных средств, повышающих ее

надежность и защищенность. Например, наличие в информационной системе *резервных копий* файлов или дополнительных данных, удаление которых не нарушает полноты и правильности хранимых сведений о предметной области, но которые могут быть использованы в случае повреждения основных файлов. См. *резервирование*

ИЗМЕНЕНИЕ РАЗМЕРА ОКНА [zooming]. Многие виды *окон* допускают изменение своих размеров пользователем. Чтобы изменить размер окна, нужно подвести *указатель мыши* к границе окна так, чтобы он принял соответствующий вид (см. табл. У.1), нажать левую кнопку мыши и *перетащить мышью* границу окна в желаемом направлении. И. р. о. возможно не во всех случаях. Например, в *операционной системе Windows* окно может находиться в одном из трех состояний: обычном, свернутом и развернутом. В обычном состоянии окно занимает только часть возможного пространства на экране. Именно в этом состоянии видны границы окна и возможно И. р. о. В свернутом состоянии окно изображается в виде значка фиксированного размера, содержимое окна не отображается и И. р. о. невозможно. В развернутом состоянии окно занимает все возможное пространство, т. е. весь экран или все родительское окно, границы окна не отображаются и И. р. о. также невозможно. На рис. И.2 показано окно приложения MS Excel в развернутом состоянии, внутри него два окна электронных таблиц в обычном состоянии и одно окно — в свернутом состоянии, а также оконное меню свернутого окна. Для перевода окна из одного состояния в другое используются кнопки, расположенные в правом верхнем углу окна, или команды оконного меню

ИКОНКА [icon]. То же, что *значок*

ИМЕНОВАННОЕ ЗНАЧЕНИЕ [tagged value]. Термин *унифицированного языка моделирования* UML, обозначающий пару, состоящую из имени и значения. И. з. можно прикрепить к любому элементу модели для передачи дополнительной информации

ИМЕНОВАННЫЕ АРГУМЕНТЫ [named arguments]. Способ задания *фактических параметров* при *вызове функции*, *процедуры* или *макрокоманды*, при котором указывается имя *формального параметра* и его значение. Использование И. а. при вызове имеет следующие преимущества по сравнению с обычным позиционным способом указания значений аргументов: 1) аргументы можно указывать в произвольном порядке; 2) можно не указывать значения некоторых аргументов, если они имеют значения по умолчанию. И. а. используются, например, в языке Visual Basic. Функцию `MsgBox` языка Visual Basic, предназначенную для вывода сообщений и имеющую пять аргументов, можно вызвать с помощью такого выражения:

```
MsgBox (prompt:="Текст сообщения", buttons:= 0)
```

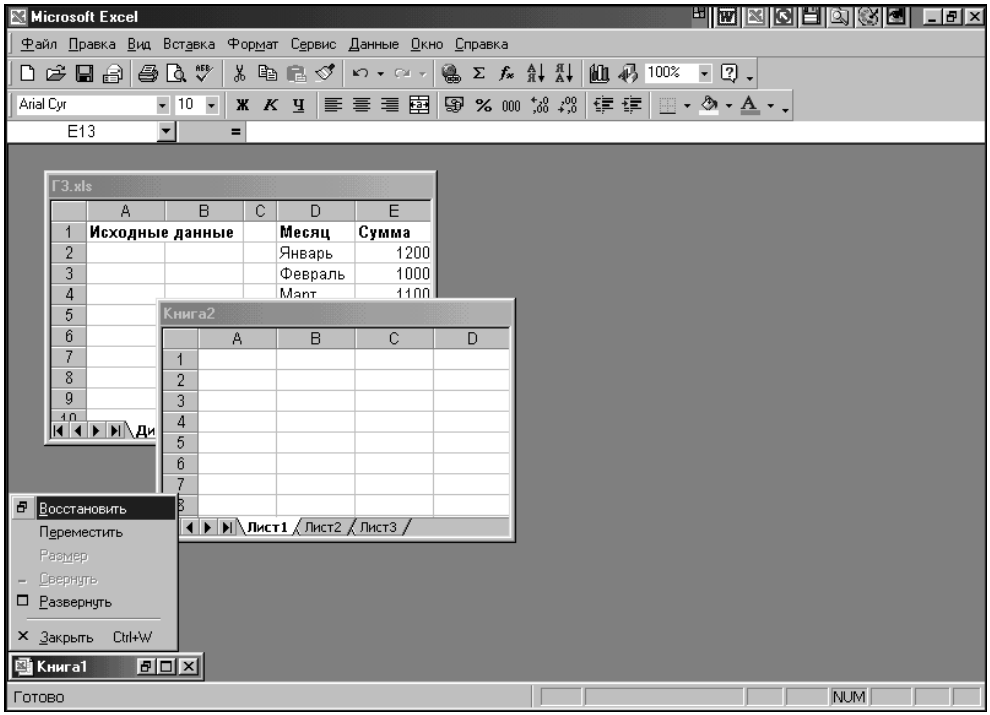

Рис. И.2. Окна в свернутом, развернутом и обычном состоянии

ИМЕНОВАННЫЙ КАНАЛ, **абстрактный файл [named pipe]**. Область памяти, которую может использовать один процесс при передаче информации другому. И. к. используется для организации *конвейера*. В операционных системах MS-DOS и OS/2 И. к. обозначается символом |

ИМПЕРСОНАЦИЯ [impersonation]. Функция *операционной системы Windows NT*, позволяющая одному процессу действовать от имени другого процесса с использованием его атрибутов безопасности

ИМПЛИКАЦИЯ [implication]. *Двуместная логическая операция*, результатом которой является *логическое значение* "ЛОЖЬ", только если первый аргумент имеет значение "ЛОЖЬ", а второй — "ИСТИНА". При всех других сочетаниях значений аргументов результатом И. является значение "ИСТИНА"

ИМПОРТ [import]. 1. Операция переноса данных из одной системы, программы или документа в другую систему, программу или другой документ. Для успешного И. необходимо, чтобы система, получающая данные, поддерживала формат или структуру передаваемых данных. В противном случае в процессе И. необходимо предусмотреть преобразование формата. См. *фор-*

матирование данных. 2. Данные, заимствованные системой, программой или документом из другой системы, программы или другого документа. Ср. *внедренный объект*

ИМЯ [name]. 1. То же, что *идентификатор.* 2. *Идентификатор*, присвоенный определенному *объекту программы* путем явного объявления, описания или по умолчанию. Вид именуемого объекта зависит от *языка программирования.* Это может быть переменная, массив, структура данных высокого уровня, файл, оператор, подпрограмма или сама программа

ИМЯ ГРУППЫ [group name]. Уникальное *имя*, определяющее в *операционной системе Windows 2000 локальную* или *глобальную группу.* И. г. не должно совпадать с любым другим И. г. или именем пользователя в домене или на компьютере

ИМЯ ДОМЕНА, **доменное имя**, **имя DNS [domain name]**. Уникальное символьное *имя*, присвоенное узлу *компьютерной сети* в соответствии с *доменной системой имен.* Например, **www.microsoft.com**. В сети Интернет это имя устройства, имеющего постоянный *IP-адрес.* Обычно оно характеризует общее положение узла. И. д. различаются по уровням. Имена первого уровня, национальные и международные (ru, com), находятся в ведении специальных организаций и не могут быть присвоены отдельным устройствам. И. д. второго уровня (amazon.com, ozon.ru) могут быть получены в собственность у этих организаций. Имена третьего уровня предоставляются владельцами имен второго уровня. Каждое И. д. регистрируется в сети и в *протоколе TCP/IP* служит идентификатором либо отдельного компьютера, либо нескольких компьютеров, объединенных в функциональную группу (домен). Ср. *интернет-адрес.* См. *доменная система имен*

ИМЯ ИСТОЧНИКА ДАННЫХ [data source name (DSN)]. *Логическое имя*, которое используется при *открытом доступе к базам данных* для получения доступа к *источнику данных.* И. и. д. применяется *информационным сервером Интернета* для установления связи с *источником данных*, например, с базой данных SQL Server

ИМЯ ПОЛЬЗОВАТЕЛЯ [user name]. Уникальное *имя*, определяющее *учетную запись пользователя* в *операционной системе Windows 2000.* И. п., определенное в учетной записи, не может совпадать с каким-либо другим *именем группы* или именем пользователя в том же домене или *рабочей группе*

ИМЯ ПУНКТА ОБРАБОТКИ ПОЧТОВЫХ СООБЩЕНИЙ, **MX-запись [mail exchanger record**, **MX record]**. Запись в *базе данных доменной системы имен*, указывающая почтовый сервер, который или произведет обработку почты, или осуществит ее перенаправление

ИМЯ УЗЛА [node name]. *Доменное имя* устройства в сети Windows 2000. Используется для поиска компьютера в сети. Чтобы поиск компьютера стал

возможен, данное имя должно быть известно *серверу DNS*. Для большинства компьютеров с *операционной системой Windows* это имя совпадает с именем компьютера

ИМЯ DNS [domain name]. То же, что *имя домена*

ИНВАРИАНТ [invariant]. *Ограничение*, которое должно соблюдаться постоянно

ИНВЕРСИЯ ЦВЕТА [invert color]. Изменение цвета на противоположный (на 180° по угловой шкале цветового круга). См. *цветовая модель*

ИНВЕРТИРОВАТЬ [invert]. Обратить, изменить на противоположный. Например, И. цвета на монохромном дисплее означает заменить черное на белое и, наоборот, белое на черное. Этот прием использован для выделения данных в приложениях *операционной системы Windows* (см. *выделение*). И. двоичный *цифровой код* означает заменить нули на единицы и, наоборот, единицы на нули. Существуют специальные электронные схемы — инверторы, которые реализуют подобную операцию над двоичным *цифровым сигналом*

ИНДЕКС [index, **subscript]**. 1. *Число целого типа* или *арифметическое выражение*, принимающее целочисленное значение, приписываемые элементу массива или другой конструкции данных для идентификации этого элемента. В *языке программирования* Фортран И. ставятся за идентификатором в круглых скобках, например, `A(1, 2), X(K), C2(I, J)`. В языках Си и Паскаль — в квадратных скобках, например, `a[1, 2], x[k], c2[i, j]`. См. *переменная с индексами*. 2. Совокупность указателей, при помощи которых можно найти запись в организованном специальным образом файле данных. 3. Адресная константа, используемая для модификации адреса. См. *индексная адресация* 4. То же, что *индекс поисковой системы*

ИНДЕКС ПОИСКОВОЙ СИСТЕМЫ, **индекс [index]**. Информационный массив (файл), в котором хранятся преобразованные особым образом текстовые составляющие всех посещенных и проиндексированных роботом *HTML-страниц* и *текстовых файлов*. Робот не только пополняет массив новыми поступлениями, но и регулярно обновляет уже имеющиеся в И. п. с. документы

ИНДЕКСИРОВАННАЯ ПЕРЕМЕННАЯ [subscripted variable]. То же, что *элемент массива*

ИНДЕКСНАЯ АДРЕСАЦИЯ, **адресация с индексированием [indexed addressing]**. Метод *адресации*, при котором формирование *исполнительного адреса* осуществляется путем добавления к *базовому адресу* содержимого *индексного регистра*

ИНДЕКСНЫЙ РЕГИСТР, **регистр индекса**, **[index register]**. *Регистр центрального процессора*, содержимым которого является значение индекса, используемое для *индексной адресации*

ИНДИКАТОР [indicator]. 1. *Элемент данных*, отражающий *текущее состояние* устройства или данных. Например, *поле данных*, в которое *системная* или *прикладная программа* помещает код ошибки, называют И. ошибки. 2. Циферблат или лампочка, сигнализирующие о состоянии устройства. Например, лампочку на дисководе, загорающуюся во время обращения к диску, называют И. активности дисковода. 3. То же, что *прогресс-индикатор*

ИНДУСТРИАЛЬНЫЙ КОНСОРЦИУМ VESA [Video Electronics Standards Association (VESA)]. То же, что *ассоциация стандартов по видеотехнике*

ИНЖЕНЕРНАЯ КОМИССИЯ ИНТЕРНЕТА [Internet Engineering Task Force (IETF)]. То же, что *Инженерный консорциум разработчиков стандартов Интернета*

ИНЖЕНЕРНЫЕ СИЛЫ ИНТЕРНЕТА [Internet engineering task force (IETF)]. То же, что *Инженерный консорциум разработчиков стандартов Интернета*

ИНЖЕНЕРНЫЙ КОНСОРЦИУМ РАЗРАБОТЧИКОВ СТАНДАРТОВ ИНТЕРНЕТА, **Инженерная комиссия Интернета**, **Инженерные силы Интернета**, **комитет IETF [Internet Engineering Task Force (IETF)]**. Организация, занимающаяся разработкой стандартов и протоколов Интернета. В рамках IETF создаются рабочие группы для решения инженерно-технических проблем. Одобренные IETF технологии обычно становятся мировым стандартом. На рис. И.3 представлена домашняя страница И. к. р. с. и. См. *стандарты RFC*

ИНЖЕНЕР-ПРОГРАММИСТ [programmer-engineer]. 1. Инженер, исполняющий функции *программиста*. 2. Инженер, имеющий квалификацию программиста

ИНИЦИАЛИЗАЦИЯ [initialization]. Приведение программы или устройства в состояние готовности к использованию. И. программы заключается в задании начальных значений или установке в ноль программных переменных (адресов, счетчиков, переключателей, указателей и т. п.) перед выполнением программы. Во многих *языках программирования* имеются средства определения начальных значений при первом описании переменной. И. *магнитного диска* включает его форматирование и запись управляющей информации (метки тома, описателей дорожек и т. п.). Ср. *Инициирование*

ИНИЦИИРОВАНИЕ [initiation]. Процесс запуска или перевода программы или устройства в активное состояние. Ср. *инициализация*

ИНКАПСУЛЯЦИЯ [encapsulation]. Термин *объектно-ориентированного программирования*, означающий структурирование программы на модули особого вида, называемые *классами* и объединяющие данные и процедуры их обработки, причем внутренние данные класса не могут быть обработаны иначе, кроме как предусмотренными для этого процедурами. Каждый такой класс имеет внутреннюю часть, называемую реализацией (или представле-

нием), и внешнюю часть, называемую интерфейсом. Доступ к реализации возможен только через интерфейс. Таким образом, реализация класса как бы заключена в капсулу и скрыта, откуда и происходит название И.

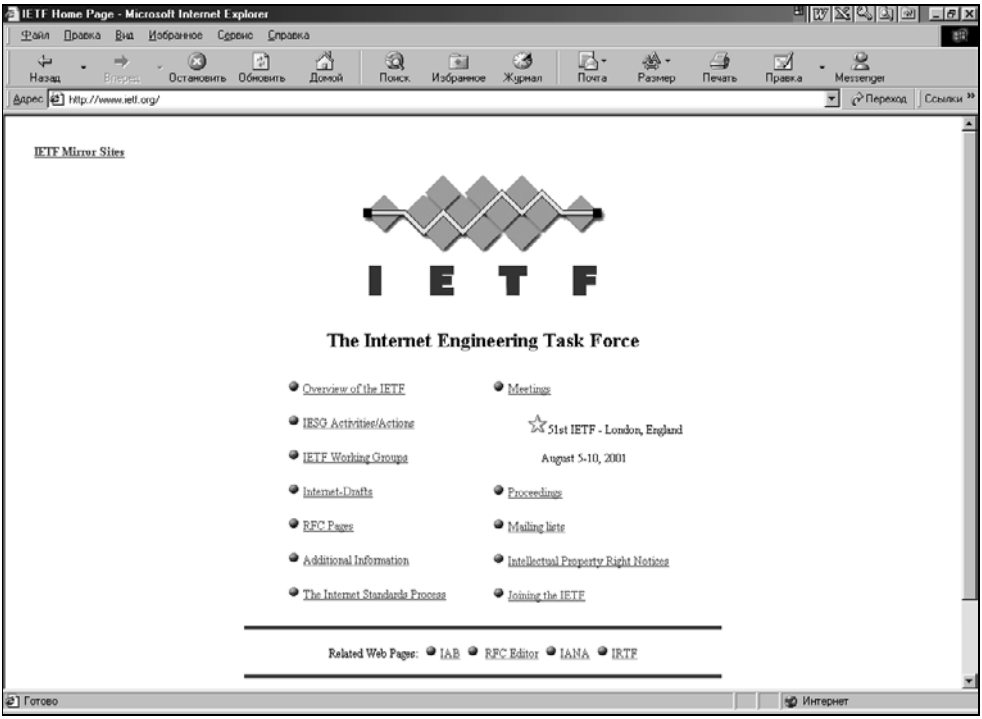

Рис. И.3. Домашняя страница консорциума IETF

ИНКРЕМЕНТНЫЙ ПРОЦЕСС РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ [incremental software development process]. *Процесс разработки программного обеспечения*, состоящий из итераций: на каждой итерации разрабатывается только часть необходимого программного обеспечения, которая называется *приращением*. При этом на каждой итерации последовательно выполняются все или только некоторые фазы процесса. Обычно используются следующие фазы: анализ *требований*, *конструирование*, *тестирование*, *развертывание*. И. п. р. п. о. хорошо подходит для случая, когда требования к *программному продукту* меняются в процессе разработки. Ср. *водопадный процесс разработки программного обеспечения*

ИНСТАЛЛЯТОР [installer]. Программа, выполняющая установку *программного продукта* на компьютер. См. *инсталляция*

ИНСТАЛЛЯЦИЯ, **установка [installation**, **setup]**. Установка *программного продукта* на компьютер. И. обычно выполняется под управлением *инсталлятора* — программы, которая приводит состав и структуру устанавливаемого программного изделия в соответствие с *конфигурацией компьютера*, а также настраивает программные параметры согласно типу имеющейся *операционной системы*, классам решаемых задач и режимам работы. Таким образом И. делает программный продукт пригодным для использования в данной *вычислительной системе* и готовым решать определенный класс задач в определенном режиме работы

ИНСТИТУТ ИНЖЕНЕРОВ ПО ЭЛЕКТРОТЕХНИКЕ И РАДИОЭЛЕКТРОНИКЕ (ИИЭР США) [Institute of Electrical and Electronics Engineers (IEEE)]. Международная организация, объединяющая технических специалистов из более чем 140 стран. Помимо издательской и образовательной деятельности разрабатывает стандарты в области электротехники и электроники, в том числе стандарты *компьютерных сетей*. См. *архитектура Ethernet*

ИНСТИТУТ ТЕХНОЛОГИЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ [software engineering institute (SEI)]. Институт, созданный с целью обеспечить качество программного обеспечения, разрабатываемого по заказам Министерства обороны США. Его достижения нашли применение в многочисленных организациях, не имеющих отношения к оборонной промышленности

ИНСТРУКЦИЯ [instruction]. 1. В некоторых *языках программирования* — то же, что *команда*, *предложение* или *оператор*. 2. Документ, указывающий и определяющий порядок работы при инсталляции и эксплуатации *программного продукта*

ИНСТРУМЕНТАЛЬНАЯ ПАНЕЛЬ [toolbar]. То же, что *панель инструментов*

ИНСТРУМЕНТАЛЬНЫЕ ПРОГРАММНЫЕ СРЕДСТВА [software tools, **tools]**. *Программные средства* (*сервисные программы*, *пакеты прикладных программ*, оболочки), предназначенные для автоматизации создания, редактирования и отладки различных *программных продуктов*: программ, *баз данных* и *знаний*, *электронных таблиц*, текстов, графических изображений и т. д. Например, *текстовый редактор*, позволяющий набирать текст программы, отладчик, облегчающий программисту выполнение отладки разрабатываемой программы, *библиотека программ*, облегчающая разработку *пользовательского интерфейса* программы и т. д.

ИНСТРУМЕНТЫ МАШИННОЙ ГРАФИКИ, **графические инструменты [graphics tools]**. *Инструментальные программные средства*, предназначенные для автоматизации создания и редактирования графических изображений. Для управления И. м. г. служат *панели инструментов*. Например, панель инструментов *графического редактора* Adobe Photoshop (рис. И.4) содержит кнопки управления инструментами для выделения, рисования и редактиро-

вания, а также для представления рисунка на экране. Каждая кнопка снабжена соответствующей пиктограммой. Инструмент включается щелчком мыши по кнопке. *Двойной щелчок* вызывает на экран *диалоговое окно* с установками, относящимися к данному инструменту. См. *аэрограф, ластик, пипетка, рука, лассо, выделение области, кисть, заливка, карандаш*

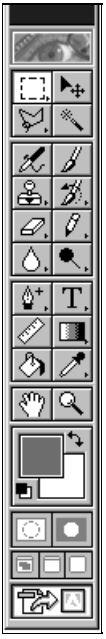

Рис. И.4. Панель инструментов графического редактора Adobe Photoshop

ИНСТРУМЕНТ CASE [computer-aided software engineering (CASE), CASE-tool]. То же, что *автоматизированная разработка программ*

ИНТЕГРАЛЬНАЯ СХЕМА (ИС), **чип [integrated circuit**, **chip]**. *Электронная схема*, выполненная на поверхности или внутри одного полупроводникового кристалла. ИС состоят из большого числа электронных элементов, способных выполнять *логические операции* или хранить информацию. Поэтому ИС могут реализовать различные функции обработки, хранения и передачи информации. Большие ИС (БИС) могут содержать тысячи электронных логических элементов на одном кристалле, а сверхбольшие (СБИС) — миллионы таких элементов. См. *микросхема, транспьютер*

ИНТЕГРИРОВАННАЯ СИСТЕМА, **интегрированный пакет [integrated system**, **integrated package]**. *Комплекс программ*, обеспечивающих различные информационные и вычислительные потребности пользователя, объединен-

ных единым *пользовательским интерфейсом* и единым способом представления данных. Обычно И. с. содержат *текстовый процессор*, средства работы с *электронными таблицами*, средства отображения и печати *графической информации*, простую *систему управления базами данных* и средства связи через модем. Наиболее ярким примером является пакет Microsoft Office. Ср. *интегрированное программное обеспечение*, *интегрированная среда*

ИНТЕГРИРОВАННАЯ СРЕДА [integrated environment]. *Интегрированная система*, включающая все необходимые пользователю *программные средства* и обеспечивающая единообразное взаимодействие с ними. Примером И. с. может служить *операционная система Windows* вместе с набором приложений, обеспечивающим широкие информационные и вычислительные потребности большинства пользователей *персональных компьютеров*. Все приложения Windows обладают единообразным *графическим интерфейсом пользователя* и могут обмениваться данными через *буфер обмена*

ИНТЕГРИРОВАННАЯ СРЕДА РАЗРАБОТКИ [integrated development environment (IDE)]. *Интегрированная система*, которая является *системой разработки программ* и снабжена *графическим интерфейсом пользователя*. Примером И. с. п. может служить *Microsoft Visual Studio*

ИНТЕГРИРОВАННАЯ ЦИФРОВАЯ СЕТЬ СВЯЗИ, сеть ISDN [integrated services digital network (ISDN)]. Стандартный тип сети *передачи данных*, обеспечивающей передачу данных разного вида (текстовых, звуковых, видео и др.) по специальным высокоскоростным цифровым телефонным каналам. Сеть ISDN построена на двух типах *каналов связи*: канал B, передающий голос, данные или видео со скоростью 64 Кбит/с, и канал D, передающий управляющую информацию со скоростью 16 Кбит/с. Компьютеры и другие устройства соединяются с сетью ISDN *платами интерфейса ISDN*

ИНТЕГРИРОВАННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ [integrated software]. Набор *программных модулей*, объединенных удобными средствами *передачи данных* от модуля к модулю и однотипным *пользовательским интерфейсом*. Для этого модули должны поддерживать единый внутренний формат или структуру передаваемых данных. Единый интерфейс существенно упрощает работу пользователя, т. к. программные модули могут выполнять различные виды работ. Например, имея интегрированные программы численного решения дифференциальных уравнений и построения графиков, пользователь может, решив уравнение, тут же представить численное решение в наглядном виде. Более высокой степенью интеграции и, соответственно, бóльшими возможностями обладают *интегрированные системы*. См. *интегрированная среда разработки*

ИНТЕГРИРОВАННЫЙ ПАКЕТ [integrated package]. То же, что *интегрированная система*

ИНТЕЛЛЕКТУАЛЬНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА [intelligent information system]. Автоматизированная *информационная система*, снабженная *интеллектуальным интерфейсом*, дающим возможность пользователю делать запросы на естественном или профессионально-ориентированном языке

ИНТЕЛЛЕКТУАЛЬНАЯ ОБУЧАЮЩАЯ СИСТЕМА [intelligent tutorial system]. *Автоматизированная обучающая система*, снабженная *интеллектуальным интерфейсом*, позволяющим обучаемому в процессе обучения вести диалог, отвечать на вопросы и выполнять задания на *естественном языке*

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ПРОГРАММИРОВАНИЯ [intelligent programming system]. *Система программирования*, в основе которой лежит естественный или профессионально-ориентированный язык. И. с. п. автоматически создает программу по сформулированному пользователем на естественном или профессионально-ориентированном языке описанию решаемой задачи. При этом пользователь освобождается от необходимости выполнять предварительную разработку алгоритма и программирование. См. *интеллектуальный интерфейс*

ИНТЕЛЛЕКТУАЛЬНЫЙ ИНТЕРФЕЙС [intelligent interface]. *Интерфейс*, обеспечивающий взаимодействие пользователя с компьютером на *естественном языке*. И. и., как правило, включает *диалоговый процессор*, интерпретирующий профессиональный язык пользователя, и планировщик, преобразующий описание задачи в программу ее решения на основе информации, хранящейся в *базе знаний*

ИНТЕЛЛЕКТУАЛЬНЫЙ ТЕРМИНАЛ [intelligent terminal]. *Терминал* с собственной памятью, процессором и *программными средствами*, позволяющими редактировать и обрабатывать данные независимо от работы ЭВМ. Примерами И. т. являются микроЭВМ или *персональный компьютер*, используемые в качестве терминала *суперкомпьютера*, а также банковские терминалы и терминалы торговых мест в супермаркетах

ИНТЕНСИВНОСТЬ ЦВЕТА [chroma]. Общее понятие, характеризующее мощность цвета. Термином "И. ц." пользуются в тех случаях, когда не уточняется, о какой именно из количественных характеристик цвета идет речь. Это может быть тон, насыщенность или яркость

ИНТЕРАКТИВНАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА [interactive computer system]. *Вычислительная система*, в которой пользователь может вести диалог с ЭВМ с терминала. Любой *персональный компьютер* является И. в. с.

ИНТЕРАКТИВНЫЙ РЕЖИМ [interactive mode]. То же, что *диалоговый режим*

ИНТЕРЛИНЬЯЖ [line spacing, leading]. То же, что *межстрочный интервал*

ИНТЕРНЕТ, сеть Internet [Internet]. Всемирная информационная *компьютерная сеть*. Представляет собой объединение множества региональных

компьютерных сетей и компьютеров, обменивающихся друг с другом информацией по каналам общественных телекоммуникаций (телефонной, радио- и спутниковой связи). И. появился в конце 70-х — начале 80-х гг. в результате постепенного объединения с помощью средств телекоммуникаций компьютерной сети Министерства обороны США, сети Национального научного фонда правительства США, региональных и даже *локальных вычислительных сетей.* Согласно официальным данным, в период с 1989 по 1995 гг. сеть И. росла, ежегодно удваивая свои размеры. В настоящее время сеть перешла на коммерческую основу, однако формально ее контролирует общественная организация ISOC (Internet SOCiety). Входящие в И. компьютерные сети взаимодействуют с помощью установленных в И. *протоколов TCP/IP*, которые позволяют связывать между собой компьютеры различной архитектуры, производимые разными фирмами. Под словом И. обычно подразумевают физический уровень сети, т. е. *аппаратное обеспечение*, состоящее из компьютеров, кабелей и других устройств *передачи данных.* Работу в И. обеспечивают базовые *программные средства.* Они осуществляют поиск нужной информации в архивах, размещенных внутри И., перемещают файлы из компьютера в компьютер, обеспечивают вход в другие компьютеры, доступ к множеству серверов и *баз данных.* С помощью аппаратных и программных средств И. предоставляет пользователю различные информационные услуги, среди которых *электронная почта*, службы электронных объявлений, телеконференций и рекламы. С начала 90-х гг. в И. существует сервис, называемый *Всемирной паутиной* (World Wide Web). Технология World Wide Web позволяет на основе гипертекста и гипермедиа создавать и хранить информацию в форме *веб-документов* и просматривать все документы Web, хранящиеся в компьютерах глобальной сети, через систему связывающих их ссылок. Подключить компьютер к И. и стать пользователем электронной почты, Всемирной паутины и других услуг И. помогают *поставщики сетевых услуг* (провайдеры)

ИНТЕРНЕТ-АДРЕС, **адрес IP [IP address]**. Уникальный номер, приписанный каждому компьютеру, непосредственно присоединенному к Интернету. И.-а. состоят из четырех чисел, соединенных точками. Например, 192.112.36.42. При передаче пакетов данных по сети эти числа определяют сеть и компьютер, откуда пакет отправлен, и сеть и компьютер — получателей пакета. И.-а. используется *программным* и *аппаратным обеспечением* сети. Пользователи работают с именами, которые взаимно однозначно связаны с И.-а. См. *доменная система имен*

ИНТЕРНЕТ-ПРОВАЙДЕР [Internet provider, **Internet services provider]**. См. *поставщик сетевых услуг*

ИНТЕРНЕТ-ТЕЛЕФОНИЯ, IP-телефония [IP-telephony]. Организация телефонной связи на основе *сетей передачи данных*, преимущественно по *протоколам TCP/IP.* Телефонная связь, построенная таким образом, позволяет

равномерно распределять нагрузку в сетях с *коммутацией пакетов*, применять *сжатие* голосовых *данных* с помощью кодеков, а также использовать единое оборудование для передачи речи и данных другого характера. Все это делает применение И.-т. экономически эффективным. См. *компьютерно-телефонная интеграция*

ИНТЕРПРЕТАТОР [interpreter]. *Транслятор* (программа или устройство), анализирующий команды или операторы *исходной программы* и немедленно выполняющий их. Таким образом, И. одновременно и транслирует, и выполняет заданную программу, делая это покомандно или пооператорно. Ср. с *компилятором*, который только транслирует всю программу без ее выполнения. И. применяются в большинстве современных программируемых микрокалькуляторов. См. *интерпретируемый язык программирования*

ИНТЕРПРЕТАЦИЯ [interpretation]. Метод выполнения *исходной программы*, при котором каждый отдельно взятый оператор транслируется и сразу выполняется, после чего осуществляется переход к следующему оператору. И. выполняется с помощью *программных* или *аппаратных средств*, называемых интерпретаторами. Достоинством И. является возможность пошагового прослеживания выполнения программы и модификации программы во время выполнения, недостатком — относительно малая скорость выполнения. Ср. *компиляция*. См. *интерпретируемый язык программирования*

ИНТЕРПРЕТИРУЕМЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ [interpretable programming language]. *Язык программирования* (как правило, *высокого уровня*), программы на котором выполняются методом *интерпретации*. Например, если язык программирования допускает конструкции, которые позволяют создавать новые подпрограммы или модифицировать имеющиеся динамически, во *время выполнения* программы, то *компиляция* такой программы становится затруднительной. Примером И. я. п. является ЛИСП. См. *интерпретатор*. Противоп. *компилируемый язык программирования*

ИНТЕРФЕЙС [interface]. Совокупность правил взаимодействия устройств и программ между собой или с пользователем и средств, реализующих это взаимодействие. Понятие И. включает в себя как сами *аппаратные* и *программные средства*, связывающие различные устройства или программы между собой или с пользователем, так и правила и алгоритмы, на основе которых эти средства созданы. Например, И. устройств — это и линии связи между ними, и *устройства сопряжения*, и способ преобразования передаваемых от устройства к устройству сигналов и данных, и физические характеристики *канала связи*. Программный И. — это и программы, обслуживающие *передачу данных* от одной задачи к другой, и *типы данных*, и список общих переменных и *областей памяти*, и набор допустимых процедур или операций и их параметров. И. пользователя с программой — это и изображенные на экране терминала *кнопки*, меню и другие элементы управления, с помощью которых пользователь управляет решением задачи, и сам терми-

нал и предусмотренные в программе операторы, позволяющие такое управление осуществить. См. *интерфейс пользователя, интерфейс прикладного программирования*

ИНТЕРФЕЙС ГРАФИЧЕСКИХ УСТРОЙСТВ, **интерфейс GDI [graphics device interface (GDI)]**. Один из самых важных компонентов семейства *операционных систем Windows*. Отвечает за управление всеми графическими элементами, выводящимися на экран или принтер. Приложения Windows используют для работы с графической информацией вызовы функций API, предоставляемые И. г. у.

ИНТЕРФЕЙС ДЛЯ ПЕРЕДАЧИ ДАННЫХ ПО ВОЛОКОННО-ОПТИЧЕСКИМ КАНАЛАМ, **стандарт FDDI [fiber distributed data interface (FDDI)]**. Стандарт высокоскоростных оптоволоконных сетей, разработанный *Американским национальным институтом стандартов* (ANSI). Спецификации этого стандарта обеспечивают надежную *передачу данных* со скоростью 100 Мбит/с между *узлами сети*, отстоящими друг от друга на расстоянии до 100 км

ИНТЕРФЕЙС МАЛЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ, **интерфейс SCSI** (читается "интерфейс скази") **[small computer system interface (SCSI)]**. Стандарт *интерфейса*, который определяет работу электрических схем и соединений при *передаче данных* между устройствами *персональных компьютеров*. В основе стандарта лежит протокол, разработанный и запатентованный фирмами Shugart Associates и NCR Corporation примерно в то же время, когда IBM выпустила свой первый PC. Через некоторое время после этого данный стандарт был принят (и несколько изменен) *Американским национальным институтом стандартов*, который дал ему имя SCSI. Интерфейс разработан для связи компьютера с *периферийными устройствами*, например, с винчестером, принтером и т. п., а также для связи с другими компьютерами в *локальной вычислительной сети*. Через порт SCSI к компьютеру может быть подсоединено до семи устройств, но передачу данных одновременно может осуществлять только одно устройство. Поэтому каждое подсоединенное устройство наделяется приоритетным номером. Чем выше номер, тем выше приоритет устройства в очереди на передачу данных. Как правило, адаптеру шины SCSI, подключенному к системной шине PC, присвоен идентификатор 7. Стандарт SCSI имеет несколько спецификаций (первоначальная версия называется теперь SCSI-1), которые допускают различные режимы передачи данных. Эти режимы обеспечивают параметры передачи данных, приближающиеся к параметрам *системных шин*. Например, скорость передачи данных по стандарту SCSI-2 в 16-битном FastWide-режиме достигает 20 Мбайт/с (стандарт SCSI-1 обеспечивал скорость передачи не более 5 Мбайт/с — один байт за каждый такт передачи при тактовой частоте 5 МГц), что позволяет применять этот интерфейс для широкого класса компьютеров, включая суперкомпьютеры. Стандарт интерфейса SCSI-3 (Fast-40

Wide SCSI) позволяет передавать данные со скоростью 80 Мбайт/с по одиночному кабелю (проводу) SCSI. В настоящее время начинает применяться спецификация SCSI-3, предусматривающая возможность передачи данных по волоконно-оптическим линиям. См. *диспетчер устройств SCSI*

ИНТЕРФЕЙС ПЕРЕДАЧИ СООБЩЕНИЙ, технология MPI [message passing interface (MPI)]. Технология программирования *параллельных компьютеров* с *распределенной памятью*, основанная на *модели передачи сообщений. MPI-программа* порождает множество взаимодействующих между собой *параллельных вычислительных процессов.* Каждый процесс работает на своем процессоре в своем адресном пространстве. Никаких общих переменных или данных не используется. Основным способом *передачи данных* и обмена *управляющими сигналами* между процессами является явная посылка сообщений. Сообщение представляет собой набор данных некоторого типа, среди атрибутов которого есть идентификатор сообщения, *номер процесса* отправителя и *номер процесса* получателя. Взаимодействующие параллельные процессы можно объединять в группы, предоставляя им отдельную среду для общения — коммуникатор. Группы могут полностью входить одна в другую, пересекаться частично или не пересекаться. Каждый процесс имеет два основных атрибута: коммуникатор и номер процесса в коммуникаторе. Технология позволяет пользователю вводить *производные типы данных* и распределять процессы программы по физическим процессорам *вычислительной системы.* Практическое воплощение Т. MPI нашла в стандарте MPI, первая версия которого была разработана в 1993—1994 гг. группой MPI Forum. В стандарте специфицирован интерфейс, который должны соблюдать *системы программирования MPI* и их пользователи при создании своих *MPI-программ.* Интерфейс обеспечивает переносимость программ на уровне исходных кодов и их выполнение на *вычислительных кластерах* и *симметричных многопроцессорных компьютерах.* MPI поддерживает программирование для *гетерогенных вычислительных систем* (но только с разнородностью по форматам представления данных и в небольшой степени по архитектуре). Полный набор спецификаций интерфейса содержит описание более 120 функций. Реализации этих спецификаций представляют собой библиотеки подпрограмм и правила оформления языковых конструкций, позволяющие объединить средства обмена сообщениями и средства программирования обычных последовательных языков. В результате были созданы *системы программирования MPI* на основе языков СИ, СИ++ и Фортран. Ср. *параллельная виртуальная машина*

ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ, пользовательский интерфейс [user interface]. *Интерфейс* пользователя с программой, *пакетом прикладных программ* или *вычислительной системой.* См. *дружественный интерфейс, графический интерфейс пользователя, интеллектуальный интерфейс*

ИНТЕРФЕЙС ПРИКЛАДНОГО ПРОГРАММИРОВАНИЯ [application programming interface (API)]. Стандартизованный набор функций, объектов и других программных компонентов, с помощью которых приложение запрашивает и выполняет сервисные функции низшего уровня, такие как развертывание и свертывание окон, чтение с клавиатуры, интерпретация движений мыши и т. д. Обычно И. п. п. служит для связи *языков программирования* с низкоуровневыми компонентами *операционной системы* и указывает операционной системе очередность выполнения задач системного уровня. Основная цель И. п. п. — упростить создание приложений и обеспечить *переносимость программ*. Многие приложения (не только операционные системы) поддерживают И. п. п. и, таким образом, предоставляют возможность управлять своей работой программно, а не только с помощью *интерфейса пользователя*. См. *язык описания интерфейсов*

ИНТЕРФЕЙС ПРОГРАММИРОВАНИЯ ПРИЛОЖЕНИЙ ДЛЯ ПЕРЕДАЧИ СООБЩЕНИЙ, **интерфейс MAPI [messaging application programming interface (MAPI)]**. *Интерфейс прикладного программирования операционной системы Windows* для работы с различными почтовыми системами. При этом приложения могут использовать стандартные операции (такие как отправка сообщения) независимо от конкретной системы *электронной почты*. Необходимо только, чтобы система электронной почты также поддерживала MAPI

ИНТЕРФЕЙС ПРОГРАММИРОВАНИЯ ПРИЛОЖЕНИЙ ДЛЯ ТЕЛЕФОННОЙ СВЯЗИ, **интерфейс TAPI [telephony application programming interface (TAPI)]**. *Интерфейс прикладного программирования операционной системы Windows*, позволяющий компьютеру устанавливать связь с телефонной системой. Функции обеспечивают приложениям взаимодействия с различными типами коммуникационного оборудования (модемами и факс-модемами). И. TAPI разработан в 1993 г. компаниями Microsoft и Intel

ИНТЕРФЕЙС ADSI [active directory service interface (ADSI)]. *Интерфейс прикладного программирования* фирмы Microsoft, предназначенный для взаимодействия с различными *службами каталогов*, в частности, со *службой каталогов Active Directory*. Поддерживается начиная с *операционной системы Windows* 2000

ИНТЕРФЕЙС CGI [common gateway interface (CGI)]. Стандартный интерфейс *шлюза*, описывающий доступ HTTP-совместимых веб-серверов к внешним программам для передачи данных пользователю в форме автоматически генерируемой *веб-страницы*. Программы CGI, называемые *сценариями* или *скриптами* (CGI scripts), получают управление при заполнении пользователем экранной *формы*. Форма генерирует выходной пакет, обрабатываемый с помощью сценария, который при необходимости вызывает другие программы (средства поиска в *базе данных* или программы *электронной почты*). В результате пользователь получает новую веб-страницу, содержащую результаты поиска в базе данных или подтверждение отправки элек-

тронной почты. И. CGI служит для разработки серверных приложений HTTP. Стандарт разработан американской организацией Национальный центр приложений для суперкомпьютеров (NCSA)

ИНТЕРФЕЙС FAX SERVICES API [fax services API]. *Интерфейс прикладного программирования* для приема и отправки факсимильных сообщений *(факсов)*. Разработан фирмой Microsoft. Поддерживается *операционными системами Windows*

ИНТЕРФЕЙС GDI [graphics device interface (GDI)]. То же, что *интерфейс графических устройств*

ИНТЕРФЕЙС IDE [integrated drive electronics, integrated device electronics (IDE)]. То же, что *встроенный интерфейс накопителей*

ИНТЕРФЕЙС JTAPI [Java Telephony API (JTAPI)]. То же, что *телефонный интерфейс прикладного программирования для языка Java*

ИНТЕРФЕЙС MAPI [messaging application programming interface (MAPI)]. То же, что *интерфейс программирования приложений для передачи сообщений*

ИНТЕРФЕЙС TAPI [telephony application programming interface (TAPI)]. То же, что *интерфейс программирования приложений для телефонной связи*

ИНТЕРФЕЙС SCSI (читается "интерфейс скази") **[interface SCSI]**. То же, что *интерфейс малых вычислительных систем*

ИНТРАНЕТ, **интрасеть**, **корпоративная сеть, сеть intranet [intranet]**. *Локальная вычислительная сеть* организации или предприятия, использующая стандарты, технологии и *программное обеспечение* Интернета (в частности, *протоколы HTTP и FTP*). Обычно И. соединена с Интернетом через брандмауэр, который защищает ее от *несанкционированного доступа*. Как правило, И. пользуются только сотрудники организации, но может быть предоставлен доступ и ее деловым партнерам. Ср. *Интернет*, *сеть Ethernet*

ИНФИЦИРОВАННЫЙ ФАЙЛ [infected file]. *Файл*, "зараженный" *компьютерным вирусом*

ИНФОРМАТИКА [informatics, computer science]. Научное направление, изучающее свойства информации и способы ее представления, накапливания, автоматической обработки и передачи. И. начала формироваться в начале 1970-х гг., как дополнение и конкретизация *кибернетики* в связи с использованием ЭВМ в управлении, науке, проектировании, образовании, сфере услуг и т. д. В И. входит группа дисциплин, занимающихся различными вопросами, связанными с разработкой и применением *вычислительной техники*: прикладная математика, программирование, *искусственный интеллект*, *архитектура ЭВМ*, *вычислительные сети* и др. Современная прикладная И. занимается специальными *информационными системами*, основанными на ЭВМ и реализующими машинные *информационные технологии*. Эти системы

подразделяются на управленческие, административные, исследовательские, учебные, проектирующие, коммуникационные, системы обслуживания бытовой сферы, экологические, медицинские, военные и т. д. И. охватывает все аспекты их разработки, внедрения и влияния на развитие общества. Развитие вычислительной техники позволило И. перейти от изучения и разработки систем *обработки данных* к системам обработки знаний, музыкальных и художественных образов, т. е. к задействованию машин непосредственно в творческих процессах, широкому их использованию в качестве интеллектуальных помощников людей. Ср. *кибернетика*

ИНФОРМАЦИОННАЯ ЕМКОСТЬ, **информационный объем [information capacity]**. Способность *запоминающего устройства* разместить определенное количество информации. Измеряется максимальным количеством единиц данных (битов, байтов и т. д.), которое может храниться в запоминающем устройстве. См. *емкость памяти*

ИНФОРМАЦИОННАЯ СИСТЕМА [information system]. 1. Любая *система*, связанная с накапливанием, хранением или *обработкой информации*. В этом смысле И. с. являются и систематизированная картотека, и *банк данных*. 2. *Вычислительная система*, предназначенная для хранения, поиска и выдачи информации по запросам пользователей (людей и программ). Обычно И. с. включает в себя большие и сложные *базы данных* и *базы знаний* и обеспечивает информацией пользователей из нескольких организаций. Существуют информационно-поисковые системы, в которых поиск и отбор информации осуществляется по заданным в запросе признакам или условиям, и информационно-справочные системы, работающие в *интерактивном режиме* и обеспечивающие пользователей сведениями справочного характера

ИНФОРМАЦИОННАЯ СРЕДА [information environment]. Хранящаяся в компьютере, но не оформленная в виде *информационной системы* совокупность знаний, фактов и сведений, относящаяся к некоторой предметной области и используемая одним или несколькими пользователями

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ [information technology]. Совокупность методов, устройств и производственных процессов, используемых людьми для сбора, хранения, обработки и распространения информации. В широком смысле примером И. т. является использование конторских счет и книгопечатание. В узком смысле термин "И. т." употребляется в связи с применением современной электронной техники для *обработки информации* в целях снижения трудоемкости процессов, использующих эту информацию, повышения их надежности и оперативности. И. т. применяется в *вычислительной технике* и технике связи, в телевизионном и радиовещании, в науке, медицине, бытовой технике и т. д. Современная И. т. позволяет создавать на базе компьютеров *автоматизированные обучающие системы* и даже системы *искусственного интеллекта*

ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР (ИВЦ) [information computer center]. *Вычислительный центр*, снабженный *информационной системой* и предназначенный как для предоставления услуг по обработке и хранению данных, так и разнообразных информационных услуг

ИНФОРМАЦИОННОЕ СЛОВО [information word]. *Машинное слово*, представляющее собой объект обработки

ИНФОРМАЦИОННЫЙ ОБЪЕМ [information capacity]. То же, что *информационная емкость*

ИНФОРМАЦИОННЫЙ СЕРВЕР ИНТЕРНЕТА, **сервер IIS [Internet information server (IIS)]**. Программа — сетевой *сервер* файлов и приложений, разработанная корпорацией Microsoft для передачи сообщений через *Интернет*. Работает под управлением *операционной системы Windows* и поддерживает различные протоколы *передачи данных*

ИНФОРМАЦИЯ [information]. Совокупность знаний, фактов, сведений, представляющих интерес и подлежащих хранению и обработке. И. являются текст книги, научные формулы, поступления и выплаты по счету в банке, расписание занятий, сообщения измерительных комплексов о расстоянии между Землей и космической станцией и т. п. И., которая требуется для работы *вычислительной машины*, состоит из подлежащих обработке данных и программы, определяющей (или дающей пользователю возможность указать), что и в какой последовательности надо сделать с этими данными. И. можно передавать, запоминать, искать, копировать, принимать, обрабатывать и, наконец, создавать и разрушать. И. может создаваться и переноситься в форме световых, звуковых и радиоволн, электрического тока или напряжения, магнитных полей, знаков на бумаге. В принципе И. может переносить любая материальная структура или поток энергии. Масштабы использования И. характеризуют уровень развития общества

ИНФРАКРАСНЫЙ ИНТЕРФЕЙС [infrared interface, **Ir interface]**. *Интерфейс*, используемый для беспроводного подключения устройств. При этом связь между устройствами осуществляется посредством электромагнитных волн инфракрасного диапазона, излучаемых некоторыми видами светодиодов. Например, с использованием И. и. могут соединяться ноутбук и принтер, имеющие *инфракрасный порт*. Существует стандарт на инфракрасную *передачу данных*. См. *спецификация IrDA*

ИНФРАКРАСНЫЙ ПОРТ [infrared port, **Ir port]**. *Порт*, обеспечивающий *инфракрасный интерфейс* для беспроводного подключения периферийных устройств. Например, И. п. может использоваться в *мобильных компьютерах* для подключения принтера или для связи с *локальной вычислительной сетью*

ИС [integrated circuit, **chip]**. То же, что *интегральная схема*

ИСКЛЮЧИТЕЛЬНАЯ СИТУАЦИЯ, **особая ситуация [exception]**. *Событие* в программе, вызывающее *программное прерывание*, обработку которого предусматривает программист. См. *обработка прерываний*

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, **машинный интеллект [artificial intelligence**, **machine intelligence]**. 1. Область информатики, занимающаяся научными исследованиями и разработкой методов и средств для правдоподобной имитации отдельных функций человеческого интеллекта с помощью автоматизированных систем. В рамках И. и. создаются методы, программные и технические средства решения задач, для которых отсутствуют формальные алгоритмы: распознавание изображений, понимание *естественных языков* и речи, обучение с учетом способностей ученика, постановка диагнозов, доказательство теорем и т. п. Эти задачи обычно решаются человеком с привлечением подсознания и поэтому их довольно трудно моделировать. На основе методов И. и. разрабатываются программные интеллектуальные системы, например, *интеллектуальные информационные системы, интеллектуальные обучающие системы, интеллектуальные системы программирования* и др. Большинство таких систем используют для своей работы соответствующие *базы знаний*, которые также разрабатываются с привлечением методов И. и. Иногда программы И. и. служат для моделирования поведения человека, а иногда — для технических применений. Методы И. и. помогают и в программировании компьютерных игр. Термин "машинный интеллект", являясь синонимом И. и., чаще служит для указания только технологического аспекта проблемы И. и. 2. Свойство автоматических и автоматизированных систем выполнять отдельные функции интеллекта человека, например, выбирать и принимать оптимальные решения на основе ранее полученного опыта и анализа внешних воздействий

ИСПОЛНЕНИЕ ПРОГРАММЫ [program run]. То же, что *выполнение программы*

ИСПОЛНИТЕЛЬНЫЙ АДРЕС, **действительный адрес [effective address**, **executive address]**. *Адрес операнда* команды, содержащийся в ней или вычисляемый на основе содержимого ее полей; содержание адресной части модифицированной *команды*. Фактическое обращение к памяти выполняется именно по И. а. См. *адресация*

ИСПОЛНЯЕМЫЙ ОПЕРАТОР [imperative statement]. *Оператор* в программе, которому соответствуют одна или несколько последовательных операций, составляющих алгоритм решения задачи. Например, *оператор присваивания* или *оператор цикла*. Противоп. *невыполняемый оператор*

ИСПОЛНЯЕМЫЙ ФАЙЛ [executable file]. *Файл*, готовый к выполнению *операционной системой*. Например, в операционной системе Windows файлы, имеющие расширение exe, являются И. ф.

ИСПРАВЛЕНИЯ [revisions]. Режим *редактирования документа*, при котором внесенные изменения дополняют исходный текст и отмечаются особым образом (цветом, подчеркиванием, зачеркиванием, отметкой на полях и т. д.). В дальнейшем пользователь может принять внесенные изменения или отказаться от них. Режим записи И. очень удобен при работе нескольких авторов над одним документом, поскольку каждый автор может сразу увидеть, какие изменения внесены в документ соавторами. На рис. И.5 показан документ MS Word, в который внесены изменения в режиме записи И.

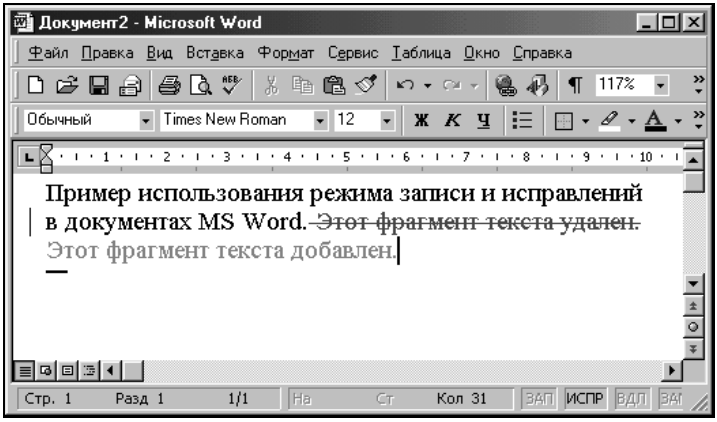

Рис. И.5. Исправления в документе Microsoft Word

ИСПЫТАНИЕ ПРОГРАММЫ, **тестирование программы [program testing]**. Проверка программы или ее составной части путем реального выполнения специально подобранных контрольных примеров. И. п. является важным этапом *разработки программы* и представляет собой проведение испытательных *прогонов программы* с целью убедиться, что она действительно решает ту задачу, для которой предназначена, и выдает правильный ответ при любых условиях. Если программа полностью транслирована и выдает некоторые результаты, это не означает, что результат ее работы правильный. Поэтому необходимы ее всесторонние испытания, направленные на то, чтобы гарантировать ее нормальную работу при всех предполагаемых практических ситуациях. В процессе И. п. устанавливается наличие или отсутствие в ней *семантических ошибок*. Удобству И. п. способствует *модульное программирование*, позволяющее проводить отдельно тестирование каждого *программного модуля*. При этом И. п. желательно проводить "сверху вниз", начиная с главной программы, заменив модули более низкого уровня (подпрограммы) имитирующими или подыгрывающими "заглушками", а затем постепенно подключать к испытаниям реальные модули. Заключительные испытания должны обеспечить проверку всей программы в целом. И. п. могут быть ав-

тономными или комплексными. *Автономные испытания программы* проводятся в процессе *отладки программы* или на начальном этапе И. п. При этом программа испытывается по частям (модулям, блокам) или целиком, но независимо от других программ, с которыми она должна взаимодействовать. *Комплексные испытания программного продукта* проводятся на завершающем этапе его разработки, выполняемом после отладки и автономных испытаний. Эти испытания предусматривают проверку правильности работы продукта в целом, взаимной увязки его отдельных частей, а также его взаимодействия с другими программами. Данные для И. п. необходимо готовить еще на этапе написания каждого модуля программы. Они должны готовиться так, чтобы испытать каждую ветвь программы. Здесь очень полезна *блок-схема программы*, показывающая возможные пути работы алгоритма. Тестовая информация должна также включать в себя все *типы данных* и допустимые диапазоны их значений. Необходимо заботиться о таком подборе тестовых примеров, чтобы они, с одной стороны, облегчали ручной контроль результатов, а с другой — не позволяли оставаться незамеченными ошибкам, непосредственно не влияющим на результат конкретной проверки. Объем проверок должен быть разумным и зависеть от предполагаемого срока эксплуатации программы и стоимости дополнительного И. п. В табл. И.1 приведен пример набора данных для тестирования модуля, вычисляющего площадь треугольника S по длинам его сторон a, b и c. Рабочая формула $S = \sqrt{p(p-a)(p-b)(p-c)}$, где $p = \frac{1}{2}(a+b+c)$. Ср. *верификация программы*

Таблица И.1. *Набор тестов программы, вычисляющей площадь треугольника по длинам трех его сторон*

Номер теста	a	b	c	Примечания
1	2	2	2	Начальный тест. Результат: $\sqrt{3} = 1.732$
2	3	4	5	Нормальный вариант. Результат: 6
3	0	0	0	Результат должен быть равен нулю
4	1	2	3	Результат должен быть равен нулю
5	0	1	2	Не треугольник. Что произойдет?
6	1	0	2	Не треугольник. Что произойдет?
7	1	2	0	Не треугольник. Что произойдет?
8	1	2	4	Не треугольник. Что произойдет?
9	3	–4	5	Неверные данные

ИСТОЧНИК ДАННЫХ [data source]. Человек, функциональное устройство, диск, файл, документ, страница в Интернете или любой другой набор данных, из которого извлекаются, копируются или перемещаются данные, используемые программой. Например, *источник данных ODBC*

ИСТОЧНИК ДАННЫХ ODBC [ODBC data source]. *Источник данных*, доступ к которому осуществляется с помощью *открытого доступа к базам данных*. При этом для источника данных имеется специальный *драйвер* ODBC, который поддерживает стандартный *интерфейс прикладного программирования* для доступа к данным. В настоящее время практически все *системы управления базами данных* имеют в своем составе драйверы ODBC и, таким образом, соответствующие *базы данных* являются И. д. ODBC. Драйверы ODBC имеются не только для баз данных, но и для данных, подготовленных самыми разными прикладными программами: *текстовых файлов*, *рабочих книг* MS Excel и др. Все они, таким образом, являются И. д. ODBC

ИСХОДНАЯ ПРОГРАММА [source program]. 1. Введенная в компьютер *программа* на *исходном языке системы программирования*. 2. То же, что *исходный текст программы*

ИСХОДНЫЕ ДАННЫЕ [source data]. *Данные*, необходимые для решения задачи. И. д. подготавливаются заранее и используются программой в ходе ее выполнения. И. д. вводятся *операторами ввода* программы или в *диалоговом режиме* — пользователем. При этом они становятся *входными данными*. И. д. для работы подпрограммы передаются ей в основном через механизм *формальных* и *фактических параметров*

ИСХОДНЫЙ МОДУЛЬ [source module]. *Программный модуль* на *исходном языке системы программирования*. При компиляции И. м. преобразуется в *объектный модуль*

ИСХОДНЫЙ ТЕКСТ [incoming text, source text]. *Текст*, подготовленный для ввода в компьютер

ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ, исходная программа [source program text]. *Текст* программы на *исходном языке системы программирования*

ИСХОДНЫЙ ЯЗЫК СИСТЕМЫ ПРОГРАММИРОВАНИЯ [source language]. *Язык программирования*, являющийся составной частью *системы программирования*. Используется для подготовки и ввода программы в компьютер с целью последующей ее компиляции *программными средствами* системы программирования. См. *исходный модуль*

ИТЕРАЦИОННЫЙ ЦИКЛ [iterative loop]. *Цикл*, число повторений в котором заранее не известно. Такие циклы встречаются при решении задач методами последовательных приближений (методами итераций), позволяющих

найти решение с наперед заданной точностью. Например, решение некоторого уравнения методом итераций с заданной точностью или суммирование сходящегося ряда с заданной точностью. Принцип построения логического условия, определяющего повторение циклического процесса вычислений в обоих случаях одинаков. Цикл выполняется, пока некоторая монотонно убывающая величина Δ остается по абсолютной величине больше наперед заданного положительного числа ε, т. е. $|\Delta| > \varepsilon$. Для метода последовательных приближений под Δ можно понимать величину, характеризующую погрешность полученного приближения, например, разность между двумя соседними приближениями. На рис. И.6 приведена блок-схема программы вычисления корня уравнения $x = \varphi(x)$ методом итераций.

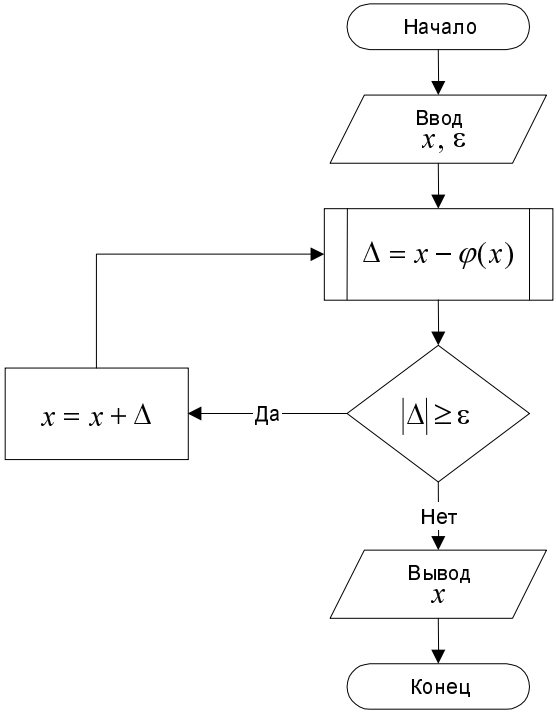

Рис. И.6. Блок-схема программы вычисления корня уравнения $x = \varphi(x)$ методом итераций

Начальное приближение x и требуемая точность ε вводятся. В случае суммирования сходящихся рядов это может быть очередной член ряда или некоторая величина, характеризующая погрешность полученной суммы. На рис. И.7 представлена блок-схема программы вычисления суммы ряда

$S = \sum_{k=1}^{\infty} u_k(x)$. Предполагается, что заданная точность ε не достигнута, если $|u_k| > \varepsilon$. В И. ц. *параметр цикла*, как правило, отсутствует. Программирование И. ц. см. *операторы цикла языка Паскаль, операторы цикла языка Си, оператор цикла языка Фортран*

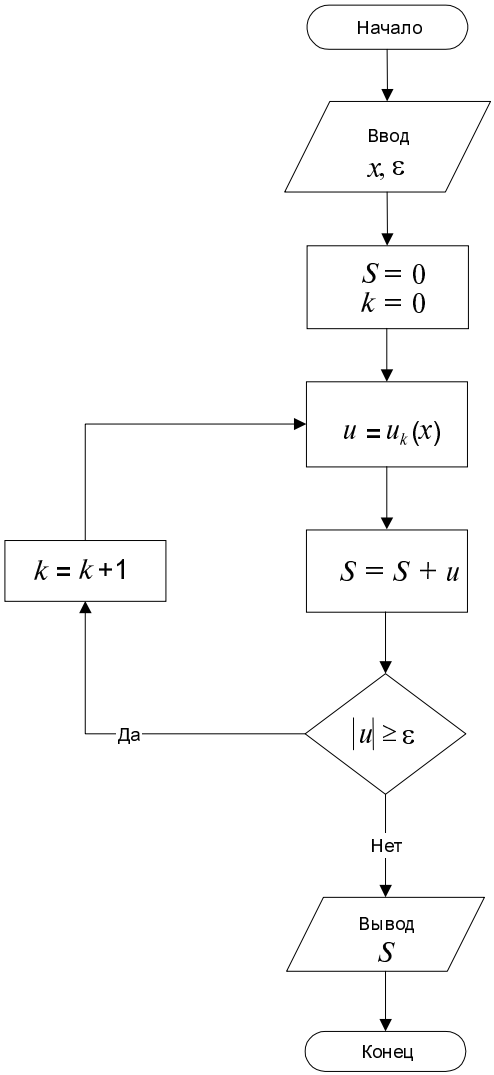

Рис. И.7. Блок-схема программы вычисления суммы ряда $S = \sum_{k=1}^{\infty} u_k(x)$

К

К [K]. То же, что *килобайт*

КАДР [frame]. 1. Содержимое экрана дисплея. 2. В *компьютерных сетях* — порция данных, передаваемая по *каналу связи*. 3. В некоторых *текстовых процессорах* и *настольных редакционно-издательских системах* — прямоугольная область, содержащая, например, иллюстрацию, которую при *верстке страниц* можно произвольным образом перемещать по странице независимо от остального текста. 4. То же, что *фрейм*

КАДРИРОВАНИЕ [crop, cropping]. То же, что *обрезка изображения*

КАЛИБРОВКА [calibration, tuning]. Процесс тонкой настройки устройств. Например, К. монитора, сканера, принтера и т. п. с целью более точной передачи цвета в *настольной редакционно-издательской системе* и соответствия его полиграфическим возможностям издательства (средств типографской печати)

КАЛЬКУЛЯТОР [calculator]. 1. Малогабаритное вычислительное устройство, выполняющее элементарные операции над числами, требующее ручного ввода чисел и команд. Различают К. программируемые и непрограммируемые. Непрограммируемые К. не имеют памяти для хранения программ, и каждая команда вводится вручную и тотчас выполняется. Программируемый К. обладает относительно небольшой памятью для хранения чисел и относительно несложных программ. Все К. выполняют команды методом интерпретации. См. *интерпретатор*. 2. То же, что *микрокалькулятор*

КАНАЛ [channel, data link, pipe]. 1. То же, что *канал ввода/вывода*. 2. То же, что *канал связи*. 3. То же, что *канал передачи данных*. 4. То же, что *программный канал*

КАНАЛ ВВОДА/ВЫВОДА, канал [input-output channel, channel]. Совокупность технических средств для *обмена данными* между *оперативной памятью* компьютера и *внешними устройствами*. Основу К. в./в. составляет специализированный процессор, обеспечивающий *форматирование данных* и управляющий *операциями ввода/вывода*

КАНАЛ ПЕРЕДАЧИ ДАННЫХ, канал [data link]. Физическое соединение, по которому осуществляется *передача данных* из одного устройства в другое. К. п. д. может связывать любые два устройства, способные принимать и посылать информацию, например, компьютер и присоединенный к нему принтер, или основной компьютер и терминал в сети. Иногда термин "К. п. д." подразумевает не только *линию связи*, но и любое оборудование, позволяющее принимать и передавать информацию, например, модем. Устройства, составляющие К. п. д., подчиняются протоколам, которые регламентируют процесс передачи. Ср. *канал связи*

КАНАЛ СВЯЗИ, канал [communication channel, channel]. Часть *линии связи*, обеспечивающая передачу независимого потока данных или управляющих сигналов. Совокупность технических средств для *передачи данных* в *компьютерной сети* от одного компьютера к другому. Ср. *канал передачи данных*

КАНАЛЬНЫЙ УРОВЕНЬ [data link layer]. То же, что *уровень канала*

КАНОНИЧЕСКАЯ НОТАЦИЯ ЯЗЫКА UML [canonical notation]. *Унифицированный язык моделирования* UML имеет стандартную К. н., согласно которой любая модель может быть описана монохромными линейными рисунками и текстом. Этот формат вполне подходит для вывода диаграмм на печать и для публикации. См. *элемент модели, графическая нотация*

КАПИТЕЛЬ, малые прописные [small caps]. Начертание шрифта, в котором для изображения строчных букв используются прописные буквы, но уменьшенного размера (см. рис. Н.1). В современной полиграфии применяется сравнительно редко

КАРАНДАШ [pencil]. *Инструмент машинной графики*, предназначенный для рисования при помощи мыши произвольных линий основным цветом. На рис. К.1 для *графического редактора* Adobe Photoshop показаны кнопка вызова инструмента К. и *диалоговые окна*, позволяющие установить толщину карандаша и другие параметры рисования

Рис. К.1. Нажатая кнопка вызова, образец работы и диалоговые окна инструмента **Карандаш** графического редактора Adobe Photoshop

КАРЕ [caret]. Символ ^, который вводится клавишами <Shift>+<6>

КАРКАС [framework]. Расширяемый шаблон для *разработки программ* в определенной *предметной области*

КАРМАННЫЙ КОМПЬЮТЕР [hand held PC]. То же, что *блокнотный компьютер*

КАРТРИДЖ [cartridge]. Сменный элемент устройства, содержащий расходный материал. Например, в *матричном принтере* — кассета с уложенной специальным образом красящей лентой, в копировальном аппарате — обойма с красящим порошком

КАТАЛОГ, **директорий**, **папка [directory**, **folder]**. 1. Список файлов и подкаталогов. 2. В *файловой системе* — файл, предназначенный для регистрации имен других файлов и обеспечения к ним доступа по зарегистрированным именам. К. содержит данные, описывающие группу файлов, размещенных на устройстве памяти с *прямым доступом* (обычно на *магнитном* или *лазерном диске*). Например, в *персональных компьютерах* в К. хранятся имена файлов, сведения об их типе (исполняемый, текстовый и т. п.), размере и времени последнего обновления, а также указатели места их расположения на диске. Если в К. хранится имя файла, то говорят, что этот файл находится или содержится в данном К. В любом К. кроме файлов могут содержаться другие К., называемые подкаталогами. На каждом диске имеется один главный, или *корневой*, К. Корневой К. обычно не имеет имени (считается, что его имя совпадает с именем диска, на котором он расположен) и занимает на этом диске специальное место. В корневом К. регистрируются файлы и подкаталоги (К. 1-го уровня). В К. 1-го уровня регистрируются файлы и К. 2-го уровня и т. д. Получается многоуровневая древовидная структура К. на диске. См. *дерево каталогов*

КАТЕГОРИЯ [category]. Раздел (обычно тематический) на поисковых системах Интернета

КБ [Kb]. То же, что *килобайт*

КБАЙТ [Kbyte]. То же, что *килобайт*

КБИТ/С [Kbit/s]. То же, что *килобит в секунду*

КВАНТ ВРЕМЕНИ [time slice]. Относительно короткий интервал *времени центрального процессора*, предоставляемый отдельной задаче при *квантовании времени*

КВАНТОВАНИЕ ВРЕМЕНИ [time slicing]. Режим работы *вычислительной системы*, при котором выделение центрального процессора задачам осуществляется на некоторый период — *квант времени*. После того как квант времени, выделенный задаче, истекает (при условии, что раньше не наступило прерывание), она временно откладывается и помещается в конец очереди готовых к работе задач. Следующий квант выделяется задаче, стоящей в очереди первой. Всем задачам может выделяться квант времени одной и той же длительности, и работающие задачи могут выбираться по циклическому алгоритму. См. *разделение времени*. Распространен также алгоритм К. в., при котором задачи, интенсивно использующие *устройства ввода/вывода*, поме-

щаются в очередь перед задачами чисто вычислительного характера, но получают более короткие кванты времени для выполнения

КВИТИРОВАНИЕ [handshaking]. Метод управления *передачей данных* между двумя устройствами, при котором одно из них передает данные, только если другое устройство подтверждает готовность к приему. При аппаратном К. по отдельному проводнику передается сигнал, уведомляющий о готовности принимающего устройства; при программном К. используются управляющие символы

КГц [KHz]. То же, что *килогерц*

КЕГЛЬ [font size, type size]. То же, что *размер шрифта*

КЕРНИНГ [kerning]. Изменение фактического интервала между некоторыми парами букв, для того чтобы добиться визуального выравнивания промежутков между буквами. Например, в паре букв "Тл" фактический интервал между буквами делается несколько меньше обычного, в противном случае из-за особенностей формы этих букв у читающего может создаться визуальное впечатление, что интервал больше обычного. В современных текстовых процессорах К. выполняется автоматически в соответствии с *гарнитурой* и *размером шрифта*. Ср. *разрядка, уплотнение шрифта*

КИБЕРНЕТИКА [cybernetics]. Наука об управлении и связи в природе и обществе. Основой К. явилось учение американского математика Н. Винера об обратной связи в сложных системах и организмах, опубликованное им в 1948 г. в книге "Кибернетика, или управление и связь в животном и машине". К. рассматривает сложные объекты природы и общества независимо от способа их организации, как большие кибернетические системы, состоящие из управляющих и управляемых элементов, между которыми существуют информационные прямая и обратная связи. С точки зрения К., системами являются часовой механизм, электронное устройство и человеческий организм. Такой подход привел к системному анализу — научному методу исследования структуры и функционирования сложных явлений, процессов и объектов природы и общества. Результатом анализа является математическое описание способов управления и распространения информации в системе. За анализом следует не только моделирование управления системами и прогнозирование их поведения, но и синтез новых систем. Кибернетический подход стал широко применяться в связи с созданием и развитием ЭВМ. Это привело к образованию ряда научных направлений К. Теоретическая К. на основе дискретной математики занимается теорией управления и теорией информации. Техническая К. занимается средствами *автоматизации* управления, включая ЭВМ и *автоматизированные системы управления*. Биологическая К. использует идеи теоретической К. в биологии и медицине. Экономическая К. занимается созданием математических моделей экономических процессов и применением компьютеров в экономических рас-

четах. Социальная К. строит и изучает математические модели управления различными процессами, протекающими в человеческом обществе. Многими задачами, поставленными К., в настоящее время занимается *информатика*

КИБЕРПРОСТРАНСТВО [cyberspace]. Термин, обозначающий весь диапазон информационных ресурсов, доступных через глобальные *компьютерные сети*. К. употребляется как синоним всего многообразия компьютерных сетей — Интернет, экстранет и т. д. Впервые этот термин появился в новелле "Neuromancer" Вильяма Гибсона (William Gibson)

КИЛОБАЙТ (Кбайт, Кб, К) [kilobyte (Kbyte, Kb, K)]. Единица количества информации; 1 Кбайт = 1024 байта. Пишется с большой буквы "К", чтобы отличить от сокращения "к" в метрической системе, обозначающего ровно 1000

КИЛОБИТ В СЕКУНДУ (Кбит/с) [Kilobit per second (Kbit/s)]. Единица скорости передачи данных, равная 1024 бит в секунду. Применяется для описания низкоскоростных *каналов связи*. Пишется с большой буквы "К", чтобы отличить от сокращения "к" в метрической системе, обозначающего ровно 1000. См. *мегабит в секунду*

КИЛОГЕРЦ (кГц) [kilohertz (kHz)]. Единица измерения частоты, эквивалентная 1000 колебаний в секунду; 1 КГц = 1000 Гц

КИСТЬ [paint brush]. *Инструмент машинной графики*, предназначенный для рисования мягкими мазками. Рисование производится при помощи мыши. Размеры К. и другие параметры инструмента задаются в *диалоговых окнах*. На рис. К.2 для *графического редактора* Adobe Photoshop показаны кнопка вызова инструмента К. и *диалоговые окна*, позволяющие установить параметры инструмента

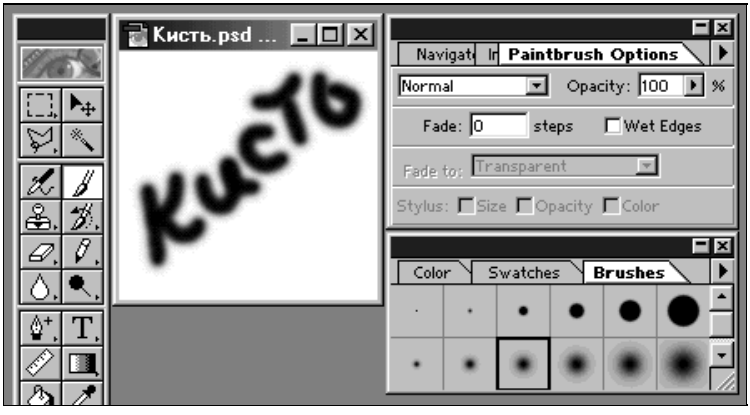

Рис. К.2. Нажатая кнопка вызова, образец работы и диалоговые окна инструмента **Кисть** графического редактора Adobe Photoshop

КЛАВИАТУРА [keyboard]. Устройство, предназначенное для непосредственного ввода команд и данных в компьютер. Представляет собой набор *клавиш*. Нажатие клавиши обеспечивает ввод одного, соответствующего этой клавише символа, или вызывает некоторое действие. Большинство современных К. имеют по меньшей мере 101 клавишу и несколько световых индикаторов, оповещающих о режимах работы клавиатуры. На К. имеются (рис. К.3): 1) стандартный набор *клавиш пишущей машинки* с цифрами, латинскими и русскими буквами и знаками препинания; 2) *малая цифровая клавиатура*; 3) *клавиши редактирования*; 4) *клавиши управления курсором*; 5) *функциональные клавиши*; 6) *специальные клавиши*. Буквенно-цифровые клавиши в основном используются для введения команд *операционной системы* и текстов. Малая цифровая К. дублирует имеющиеся на основной К. клавиши ввода цифр и удобна при вводе числовых данных. Клавиши редактирования и клавиши управления курсором служат для работы с информацией, выведенной на экран. Функциональные клавиши отведены для управления различными функциями вычислительной системы или отдельных программ. Специальные клавиши предназначены для управления компьютером и режимом работы К. Помимо упомянутых клавиш, которые присутствуют на большинстве моделей К., на некоторых моделях используются различные дополнительные клавиши. Например, на К., предназначенных для работы с *персональным компьютером* под *управлением операционной системы Windows*, существуют дополнительные клавиши, позволяющие без использования мыши нажать кнопку **Пуск** на *панели задач* или вызвать *контекстное меню*

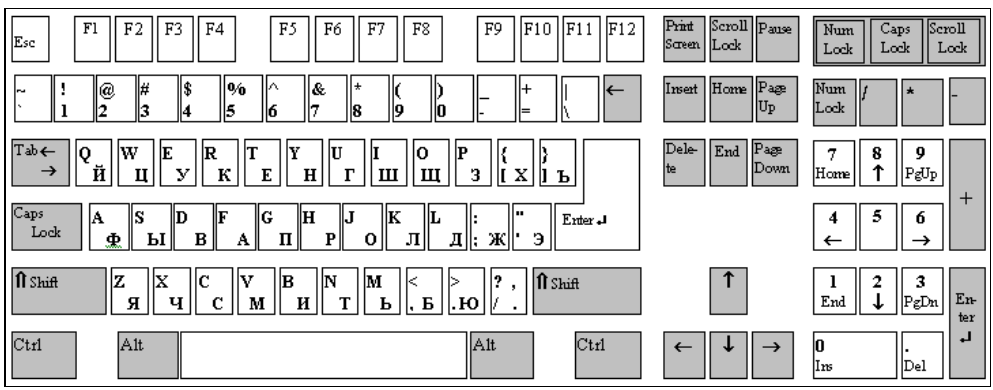

Рис. К.3. Клавиатура персонального компьютера

КЛАВИША [key]. Элемент клавиатуры, нажатием которого генерируется код соответствующего знака или инициируется определенное действие

КЛАВИШИ БЫСТРОГО ВЫЗОВА, **быстрые клавиши**, **клавиши быстрого доступа**, **горячие клавиши [quick keys**, **access keys**, **hot keys]**. *Командные клавиши*, одновременное нажатие на которые сразу вызывает определенные действия программы. В качестве К. б. в. обычно используются *функциональные* и *управляющие клавиши* клавиатуры в сочетании с буквенными. В табл. К.1 приведено краткое описание действия основных К. б. в. *в приложениях Microsoft Office*. Использование К. б. в. значительно ускоряет работу по сравнению с управлением программой путем выбора пунктов меню. Многие приложения позволяют переопределять имеющиеся и добавлять новые К. б. в.

Таблица К.1. *Основные клавиши быстрого вызова в приложениях Microsoft Office*

Сочетание клавиш	Действие
<Alt>+<Backspace>	Отмена последнего действия
<Ctrl>+<B>	Добавление полужирного начертания
<Ctrl>+<F>	Поиск слов и словосочетаний в документе
<Ctrl>+<I>	Добавление курсивного начертания
<Ctrl>+<N>	Создание нового документа
<Ctrl>+<O>	Открытие существующего документа
<Ctrl>+<P>	Печать документа
<Ctrl>+<U>	Добавление подчеркивания
<Ctrl>+<C>	Копирование выделенного текста или объекта в буфер обмена
<Ctrl>+<A>	Выделение всего документа
<Ctrl>+<X>	Вырезание выделенного текста или объекта в буфер обмена
<Ctrl>+<V>	Вставка текста или объекта из буфера обмена
<Shift>+<F12>	Сохранение документа
<F1>	Вызов справочной системы

КЛАВИШИ БЫСТРОГО ДОСТУПА [access keys]. То же, что *клавиши быстрого вызова*

КЛАВИШИ ПИШУЩЕЙ МАШИНКИ [type writer keys]. Стандартный набор К. п. м. с цифрами, латинскими и русскими буквами и знаками препинания, предназначенный для ввода символов, текстов и некоторых команд. Расположение латинских и русских букв на клавиатуре *персональных компьютеров* такое же, как на английской и русской пишущих машинках. Оно соответствует стандартам QWERTY и ЙЦУКЕН, названным так по расположению букв на первых шести клавишах первого буквенного ряда клавиа-

туры (см. рис. К.3). Символы — буквы русского алфавита (кириллицы) — могут быть введены непосредственно буквенными клавишами только в случае работы *операционной системы* с поддержкой кириллицы или специальной программы *драйвера-русификатора*. При этом для смены латинского алфавита на русский следует нажать одну из управляющих клавиш или сочетание определенных клавиш. Как и у любой пишущей машинки, клавиатура компьютера имеет два регистра — верхний и нижний. Верхний регистр служит для ввода прописных букв и других символов, указанных в верхней части клавиш. Переход с нижнего регистра на верхний осуществляется нажатием клавиши <Shift>. Клавиша <Caps Lock> служит для фиксации или отмены режима верхнего регистра, а клавиша <Enter> — для перехода на другую строку. Подробнее об этих клавишах см. *клавиши редактирования*

КЛАВИШИ РЕДАКТИРОВАНИЯ [edit keys]. Группа клавиш клавиатуры, позволяющих редактировать текст, выведенный на экран видеотерминала. В *персональных компьютерах* клавишами редактирования являются клавиши, перечисленные в табл. К.2 (см. рис. К.3). Эти клавиши в различных программах могут быть задействованы по-разному. Здесь приведены их типичные значения в программах — *текстовых редакторах*

Таблица К.2. *Клавиши редактирования в текстовых редакторах*

Клавиша	Действие
<Insert>	Переключение режимов вставки и замещения
<Delete>	Стирание символа, на который указывает курсор
<Backspace>	Стирание символа, расположенного в строке перед курсором. Иногда на этой клавише нарисована стрелка, направленная влево
<Home>	"Начало". После нажатия этой клавиши курсор перемещается в начало текущей строки
<End>	"Конец". После нажатия этой клавиши курсор перемещается в конец текущей строки
<Page Up>	"Страница вверх". После нажатия этой клавиши на экран выводится несколько строк (их количество равно числу строк в окне редактирования), находящихся в редактируемом тексте над строкой, которая была самой верхней на экране перед нажатием <Page Up>
<Page Down>	"Страница вниз". После нажатия этой клавиши на экран выводится несколько строк (их количество равно числу строк в окне редактирования), находящихся в редактируемом тексте под строкой, которая была самой нижней на экране перед нажатием <Page Down>
<↑>	"Строка вверх". После нажатия этой клавиши курсор перемещается на строку вверх. Если исходная строка — самая верхняя в окне редактирования, то курсор остается на месте и на экране происходит прокрутка — смещение редактируемого материала на одну строку вниз с заполнением верхней строки ранее невидимой строкой

Таблица К.2 *(окончание)*

Клавиша	Действие
<↓>	"Строка вниз". После нажатия этой клавиши курсор перемещается на строку вниз. Если исходная строка — самая нижняя в окне редактирования, то курсор остается на месте и на экране происходит прокрутка — смещение редактируемого материала на одну строку вверх с заполнением нижней строки ранее невидимой строкой
<→>	"Колонка вправо". После нажатия этой клавиши курсор перемещается на колонку вправо. Если исходная колонка крайняя правая в окне редактирования, то курсор остается на месте и на экране происходит одновременный сдвиг всех строк на одну колонку влево
<←>	"Колонка влево". После нажатия этой клавиши курсор перемещается на колонку влево. Если исходная колонка крайняя левая в окне редактирования, то курсор остается на месте и на экране происходит одновременный сдвиг всех строк на одну колонку вправо
<Tab>	Табуляция. Используется для передвижения курсора вправо на определенное число позиций
<Shift> (левая и правая клавиши)	Предназначены для ввода символа на верхнем регистре клавиатуры. При удерживаемой клавише <Shift> (безразлично, правой или левой) нажатие буквенной клавиши вызывает ввод прописной буквы, а нажатие символьно-цифровой клавиши — ввод символа, указанного в верхней части клавиши
<Caps Lock>	Включает и выключает режим работы клавиатуры на верхнем регистре. При включенном режиме Caps Lock (светится индикатор Caps Lock в правой верхней части клавиатуры) нажатие буквенной клавиши вызывает ввод прописной буквы, а нажатие символьно-цифровой клавиши — ввод символа, указанного в верхней части клавиши
<Spacebar>	Пробел (самая длинная клавиша клавиатуры)
<Enter>	Ввод признака конца строки. Иногда обозначается словом Return

КЛАВИШИ УПРАВЛЕНИЯ КУРСОРОМ [cursor control keys]. Клавиши видеотерминала или клавиатуры *персонального компьютера*, используемые для перемещения курсора по экрану. В *персональных компьютерах* это 4 клавиши со стрелками вверх, вниз, влево и вправо, а также клавиши <Home> (Начало), <End> (Конец), <Page Up> (Страницу вверх) и <Page Down> (Страницу вниз). К. у. к. дублируются на *малой цифровой клавиатуре* при выключенном режиме Num Lock. См. *клавиши редактирования*

КЛАСС [class]. 1. Группа объектов (понятий, задач, явлений, предметов, процессов), выделенная по определенному признаку. Например, К. операций над массивами. 2. В объектно-ориентированных языках программирования, например, в *языке программирования Си*++, — определяемый про-

граммистом *абстрактный тип данных*, позволяющий создать в программе новые *объекты данных* и ввести связанные с ними операции и функции. Программист описывает К. как сложный *структурированный тип*, состоящий из элементов, которые могут быть как собственно данными, т. е. значениями определенного типа данных, так и функциями, реализующими операции над элементами-данными. И те и другие элементы называются членами К. Элементы-данные также часто называются *свойствами К.*, а элементы-функции — *методами К.* Такое описание служит шаблоном для создания в программе конкретных объектов типа данного класса, имеющих свои конкретные имена. Когда создается объект типа данного класса (такой объект иногда называют *экземпляром К.*), для него выделяется память и инициализируются элементы данных, как компоненты объекта. Теперь к ним можно применять функции, заданные при описании К. Например, К. разноцветных точек на экране, каждая из которых характеризуется координатами X и Y и цветом Color, может быть описан следующим образом:

```
class Point
{                                //     свойства:
 int X, Y;                       //     координаты точки
 int Color;                      //     цвет точки
 public:                         //     методы:
 Point
(int x, int y, int color);       //     конструктор (резервирует память
                                 //     и инициализирует данные объекта)
 ~Point ();                      //     деструктор (уничтожает объект)
 void Show ();                   //     отображает точку на экране
 void Hide ();                   //     временно гасит точку на экране
 void MoveTo
(int NewX, int NewY);            //     перемещает точку на новое место
 void ChangeColor
(int NewColor);                  //     изменяет цвет точки
};
```

Теперь, инициализировав объект — точку А голубого цвета с начальными координатами $x = 5$, $y = 12$ при помощи конструктора `Point A (5, 12, BLUE);`, можно отобразить ее на экране оператором `A.Show();`, переместить оператором `A.MoveTo (7, 10);` и изменить ее цвет оператором `A.ChangeColor (GREEN);` и т. д. 3. В *унифицированном языке моделирования* UML описание множества объектов, обладающих общими атрибутами, операциями, методами, отношениями и поведением. Класс описывает некоторую концепцию в моделируемой системе. В зависимости от типа модели класс может отражать концепцию реального мира (например, для аналитической модели) или

содержать концепции, относящиеся к алгоритмической или компьютерной реализации (для модели проектирования)

КЛАСС АДРЕСОВ [address class]. Группа *интернет-адресов*, состоящая из адресов сетей определенного размера. К. а. определяется по первым восьми битам IP-адреса. Сети класса А (адреса, начинающиеся с 1—126) являются наиболее крупными (более 16 миллионов хостов), далее следуют сети класса В (128–191), которые могут насчитывать до 65 534 хостов, и, наконец, сети класса С (192–233), которые могут содержать до 254 хостов

КЛАССИФИКАТОР [classifier]. В *унифицированном языке моделирования* UML элемент модели, описывающий определенные черты структуры и поведения системы. К. являются классы, *действующие лица*, *типы данных*, интерфейсы, узлы, сигналы, подсистемы и *варианты использования*. Наиболее общими К. являются классы. Все прочие К. определяются относительно их сходства с классами, с учетом их ограничений по содержанию или использованию

КЛАССИФИКАЦИЯ ФЛИННА, таксономия Флинна [Flynn classification]. Классификация *архитектур вычислительных систем*, предложенная М. Флинном. На основе числа *потоков команд* и *потоков данных* Флинн выделяет четыре класса архитектур: 1) SISD (Single Instruction stream / Single Data stream) — одиночный поток команд и одиночный поток данных; 2) SIMD (Single Instruction stream / Multiple Data stream) — одиночный поток команд и множественный поток данных; 3) MISD (Multiple Instruction stream / Single Data stream) — множественный поток команд и одиночный поток данных; 4) MIMD (Multiple Instruction stream / Multiple Data stream) — множественный поток команд и множественный поток данных. См. *SISD-компьютер*, *SIMD-компьютер*, *MISD-компьютер*, *MIMD-компьютер*

КЛАСТЕР [cluster]. 1. Группа *блоков памяти* на диске, распределяемая *операционной системой* как единое целое. На диске К. являются один или более секторов. В один К. может быть записан только отдельный файл или его часть. Поэтому К. может рассматриваться как элемент размещения данных. 2. Группа процессоров в *многопроцессорной ЭВМ*. 3. То же, что *вычислительный кластер*

КЛАСТЕРНАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА [cluster computer system]. *Вычислительная система* на основе *вычислительного кластера*. К. в. с. при относительно невысокой стоимости может обладать производительностью, сравнимой с производительностью суперкомпьютера. Она имеет *распределенную память* и распределенное *программное обеспечение*. Наиболее распространенной *операционной системой*, под управлением которой работают К. в. с., является ОС LINUX

КЛИЕНТ [client]. 1. *Программа*, использующая определенные услуги другой программы, которая называется *сервером*. См. *автоматизация, архитектура*

"клиент-сервер", связывание и внедрение объектов. 2. Компьютер, на котором выполняется программа-клиент

КЛИЕНТ ПЕЧАТИ [print client]. *Компьютер*, который по сети отправляет *серверу печати* задания на печать. Иногда К. п. называются также клиентскими компьютерами. См. *задание на печать*

КЛИЕНТ-СЕРВЕРНАЯ ТЕЛЕФОНИЯ [client-server thelephony]. Способ организации компьютерного управления звонками на линиях абонентов согласно *архитектуре "клиент-сервер"*. Телефонная связь осуществляется с помощью компьютеров абонентов (клиентов), взаимодействующих с *телефонным сервером* посредством локальной *сети передачи данных*. При этом не нужны модемы на компьютерах абонентов. См. *компьютерно-телефонная интеграция*

КЛИПАРТ [clip art]. То же, что *аппликация*

КЛЮЧ [key]. 1. Совокупность символов или код, которые служат для идентификации и быстрого доступа к программам, данным *блокам памяти* или другим объектам, а также для их защиты. В качестве К. могут использоваться, например, несколько первых символов имени объекта. К. может присваиваться объекту операционной системой. См. *ключ записи, ключ защиты, ключ защиты памяти.* 2. Параметр шифрования данных, определяющий возможные варианты шифра

КЛЮЧ ЗАПИСИ [record key]. Совокупность символов, используемая для идентификации *записи* в файле и быстрого доступа к ней. См. *первичный ключ, составной ключ*

КЛЮЧ ЗАЩИТЫ [protection key]. 1. Код, присваиваемый программе, который должен совпадать с *ключами защиты памяти* всех *блоков памяти*, выделенных данной программе. Если программа обращается к данным, хранящимся не в "своем" блоке памяти, происходит *аварийное завершение.* Ср. *ключ защиты памяти.* 2. То же, что *аппаратный ключ*

КЛЮЧ ЗАЩИТЫ ПАМЯТИ [storage protection key]. Код, присваиваемый *блоку памяти*, выделенному программе, и используемый при обращении программы к памяти в целях ее защиты. Должен совпадать с *ключом защиты*, присвоенным программе. Ср. *ключ защиты*

КЛЮЧ РЕЕСТРА [registry key]. Запись в *реестре Windows*, содержащая уникальный идентификатор, присвоенный определенной части информации, находящейся в реестре. К. р. может содержать другие (вложенные) ключи и значимые элементы

КЛЮЧЕВОЕ СЛОВО, **зарезервированное слово**, **служебное слово [keyword**, **reserved word]**. 1. В *языках программирования* — *слово*, смысл которого зафиксирован правилами языка и по которому транслятор (или человек) распо-

знает основные языковые конструкции. Например, *оператор цикла* в языке Фортран распознается по К. с. `DO`, в языках Паскаль и Си — по К. с. `for`, а К. с. `integer` в традиционных языках программирования определяет числовые данные *целого типа*. Использование К. с. в качестве идентификаторов не разрешается, поскольку они являются зарезервированными (служебными) словами языка. 2. Слово или сочетание слов естественного языка, отражающее содержание текста

КМОП-ПАМЯТЬ, **CMOS-память [CMOS memory**, **CMOS RAM]**. *Микросхема оперативной памяти*, изготовленная на основе *комплементарных структур металл—оксид—полупроводник*. Характеризуется высокой плотностью размещения элементов, высокой скоростью и низким потреблением энергии. Часто используется в генераторе *тактовой частоты*

КМОП-СТРУКТУРА [complementary metal—oxide—semiconductor (CMOS)]. То же, что *комплементарная структура металл—оксид—полупроводник*

КНИЖНАЯ ОРИЕНТАЦИЯ СТРАНИЦЫ, **портретная ориентация страницы [portrait page orientation]**. Такое расположение страницы, когда ее высота больше ширины. Противоп. *альбомная ориентация страницы*

КНОПКА [button, **light button]**. 1. Клавиша или кнопка на аппаратуре, нажатие которой вызывает определенные действия. Например, пусковая К. вызывает запуск устройства. 2. В *интерактивных системах — элемент управления графического интерфейса пользователя*, выбор которого с помощью *указателя мыши* вызывает определенное действие системы. Чаще всего К. изображаются в виде рельефно выступающей кнопки или клавиши, на которой нарисован символический знак (пиктограмма) или написан текст. Но это может быть и просто символ, слово или часть текста, фрагмент рисунка и т. п. Указывая курсором на К. и нажимая клавишу <Enter> или кнопку мыши, пользователь выбирает и инициирует соответствующую программу или операцию либо задает их аргументы. См. *графический интерфейс пользователя, элементы управления*

КОБОЛ [Cobol]. *Проблемно-ориентированный язык программирования*, предназначенный для решения экономических задач и задач делопроизводства. Разработанный КОДАСИЛ в 1964 г., К. является одним из наиболее распространенных языков обработки коммерческой информации. Он обладает широким набором операторов для работы с файлами и записями, структура которых определяется программистом. Операторы, определяющие действия над данными, имеют форму, приближенную к английскому языку, и группируются в предложения, параграфы и секции. К. располагает удобными средствами редактирования, позволяющими в процессе определения данных задавать операции редактирования, которые будут выполняться при выводе данных. Кроме того, в К. предусмотрены средства управления видеотерми-

налами. Название языка происходит от COmmon Business-Oriented Language (универсальный язык, ориентированный на коммерческие задачи)

КОД [code]. 1. Система символов и однозначных правил их интерпретации, используемая для представления информации в виде данных. В современных компьютерах широкое распространение получил *двоичный* К. В нем каждая буква, цифра, знак препинания, пробел или какой-либо другой символ однозначно представляются (кодируются) комбинацией двоичных цифр — нулей и единиц. Например, буква S выражается К. 1010011, а буква О — 1001111. Поэтому сигнал терпящих бедствие SOS представляется в виде 101001110011111010011. См. *код ASCII.* 2. Программа, записанная на *языке программирования.* В таком смысле слово "К." применяется, чтобы подчеркнуть отличие готовой программы от ее блок-схемы или наброска, выполненного на некотором условном языке. См. *исходный текст программы*

КОД ОПЕРАЦИИ (КОП) [operation code (opcode)]. 1. *Код*, используемый для представления *операций* в командах на *машинном языке.* Каждая команда из *системы команд* ЭВМ обладает своим К. о., по которому процессор "распознает", какую следует выполнить операцию. 2. Часть *машинной команды* или *команды ассемблера*, определяющая тип выполняемой процессором операции. См. *команда*

КОД ПЕРЕКЛЮЧЕНИЯ, **код трансформации [thunk]**. Сегмент кода, способный обеспечить вызов 32-разрядного кода функции из 16-разрядного кода программы и наоборот. Применяется в Windows 95 для обеспечения взаимодействия 16- и 32-разрядных приложений, а также 16- и 32-разрядных сервисов *операционной системы*

КОД ТРАНСФОРМАЦИИ [thunk]. То же, что *код переключения*

КОД ASCII [ASCII code]. То же, что *американский стандартный код обмена информацией*

КОДАСИЛ [CODASYL]. 1. Американская организация, занимающаяся разработкой стандартов и созданием языковых средств *обработки* экономической и деловой *информации.* Например, К. разработала стандарт языка Кобол и занимается его усовершенствованием. Название К. происходит от Conference On DAta SYstem Language (Конференция по языкам информационных систем). 2. Разработанный К. набор стандартов для *сетевых баз данных*

КОДЕК [codec]. 1. То же, что *кодер-декодер.* 2. То же, что *компрессор-декомпрессор*

КОДЕК MPEG [MPEG-codec]. *Кодек*, разработанный согласно *стандарту MPEG.* См. *группа MPEG*

КОДЕР-ДЕКОДЕР, **кодек**, **[coder-decoder**, **codec]**. *Микросхема*, преобразующая *цифровые сигналы* в *аналоговые сигналы* и обратно. См. *аналого-цифровой преобразователь.* 2. То же, что *шифратор-дешифратор*

КОДИРОВАНИЕ С ОГРАНИЧЕНИЕМ ДЛИНЫ ПОЛЯ ЗАПИСИ, RLL-кодирование [run-length limited encoding (RLL encoding)]. Быстрый и высокоэффективный способ записи данных на *жесткий диск*. При К. с о. д. п. з. комбинации бит, представляющих информацию, не записываются точно бит за битом, а "кодируются" магнитным полем по схеме, построенной на "длине серии нулей". Изменение магнитного потока происходит в соответствии с количеством нулей, следующих один за другим. Такое кодирование позволяет записывать данные, существенно увеличивая емкость запоминающего устройства

КОДИРОВАНИЕ СИМВОЛОВ [character encoding]. Представление набора символов в виде последовательности цифр. Наиболее часто символами являются буквы, цифры, знаки препинания и т. п., которые представляются *двоичными числами*. См. *код ASCII, стандарт UNICODE*

КОДОВАЯ СТРАНИЦА [code page]. Средство поддержки наборов символов и *раскладок клавиатуры* для различных стран. К. с. — это таблица, связывающая используемые программой коды символов с клавишами клавиатуры и знаками на экране. Каждая К. с. имеет номер. Набор знаков и раскладку клавиатуры, принятую в России для *операционных систем Windows*, определяет К. с. 1251. Американская К. с. — 437

КОЛИЧЕСТВО ПИКСЕЛОВ НА ДЮЙМ [pixels per inch (PPI)]. Единица измерения *разрешающей способности* экрана и принтера, равная количеству пикселов, которые устройство может выводить на отрезке прямой длиной в один дюйм

КОЛИЧЕСТВО ТАКТОВ НА КОМАНДУ [cycles per instruction (CPI)]. Характеристика трудоемкости и длительности команды. Среднее значение К. т. н. к. может служить количественной характеристикой программы, т. к. в разных классах программ преобладают команды разной длительности

КОЛИЧЕСТВО ТОЧЕК НА ДЮЙМ [dots per inch (DPI)]. Единица измерения *разрешающей способности* экрана и принтера, равная количеству точек, которые устройство может выводить на отрезке прямой длиной в один дюйм

КОЛЛЕКТИВНЫЙ ОБМЕН [shared exchange]. В *модели передачи сообщений* форма обмена сообщениями, в которой участвуют более чем два процесса. Разновидностями К. о. являются: а) широковещательная передача, при которой сообщение передается от одного процесса ко всем; б) обмен с барьером, который происходит после того, как к определенной процедуре обратилось заданное число процессов; в) операции приведения, при которых всем процессам становится доступным одно значение, полученное после обработки данных, поступивших от нескольких процессов. При реализации К. о. обычно гарантируется сохранение порядка сообщений, которые не могут обгонять друг друга. Ср. *двухточечный обмен*

КОЛОНКА, **столбец [column]**. Вертикальный ряд объектов одной природы. Обычно К. образуют *элементы данных* или их позиции, расположенные на экране, бумаге или другом носителе вертикально — друг под другом. Например, столбец элементов матрицы, К. ячеек *электронной таблицы*, К. позиций перфокарты или бланка кодирования и т. п.

КОЛОНТИТУЛ [header, **footer**, **running head]**. Текст или изображение, которое размещается на верхнем или нижнем поле страницы вне области основного текста. К., размещенный на верхнем поле, называется *верхним колонтитулом* (header, running head), а размещенный на нижнем поле — *нижним колонтитулом* (footer). Обычно в К. помещают такую информацию, как номер страницы, название главы или части книги и т. п. Современные *настольные редакционно-издательские системы* имеют средства для автоматизированного формирования К.

КОЛЬЦЕВАЯ СЕТЬ, **сеть кольцевой топологии [ring network]**. *Локальная вычислительная сеть*, узлы которой соединены в кольцо. Сообщения в такой сети проходят в одном направлении от узла к узлу, каждый из которых проверяет адрес назначения, содержащийся в сообщении. Если адрес в сообщении совпадает с адресом узла, сообщение принимается, если нет, узел регенерирует сигнал и направляет сообщение к следующему узлу. См. *топология сети.* Ср. *древовидная сеть, радиальная сеть, шинная сеть*

КОЛЬЦЕВАЯ СЕТЬ С МАРКЕРНЫМ ДОСТУПОМ, **кольцевая сеть с передачей маркера**, **маркерная кольцевая сеть [token ring network]**. *Сеть с маркерным доступом*, выполненная по кольцевой топологии. Регламентируется *стандартом EEE* 802.5. См. сеть *кольцевой топологии*

КОЛЬЦЕВАЯ СЕТЬ С ПЕРЕДАЧЕЙ МАРКЕРА [token ring network]. То же, что *кольцевая сеть с маркерным доступом*

КОМАНДА [instruction, **command]**. 1. Элементарное предложение *машинного языка* или *автокода*, содержащее предписание *компьютеру* выполнить некоторую *машинную операцию* или действие. Определяя операцию или действие, К. несет следующую информацию: 1) *код операции*; 2) адреса или имена операндов, участвующих в операции. Например, К. "сложить содержимое ячейки 50 с содержимым ячейки 774" имеет следующую структуру: <Код операции "сложить"><Адрес:50><Адрес:774>. См. *система команд.* 2. *Управляющий сигнал*, инициирующий выполнение определенной операции в исполнительном устройстве

КОМАНДА АССЕМБЛЕРА [assembly instruction]. Основная конструкция (оператор) *языка ассемблера*, с помощью которой записывается программа на этом языке. Как правило, одна К. а. транслируется в одну эквивалентную *машинную команду*

КОМАНДА БЕЗУСЛОВНОГО ПЕРЕХОДА [unconditional jump instruction, unconditional branch instruction]. То же, что *команда безусловной передачи управления*. См. *команда передачи управления*

КОМАНДА БЕЗУСЛОВНОЙ ПЕРЕДАЧИ УПРАВЛЕНИЯ, команда безусловного перехода [unconditional jump instruction, unconditional branch instruction]. См. *команда передачи управления*

КОМАНДА ВВОДА [input instruction]. *Машинная команда*, определяющая *ввод данных* с *внешнего устройства* в *основную память*

КОМАНДА ВЕТВЛЕНИЯ [branch instruction]. То же, что *команда передачи управления*

КОМАНДА ВОЗВРАТА [return instruction]. *Команда перехода*, предписывающая выход из подпрограммы и возврат в *вызывающую программу*

КОМАНДА ВЫВОДА [output instruction]. *Машинная команда*, определяющая *вывод данных* из *основной памяти* во *внешнее устройство*

КОМАНДА ОПЕРАЦИОННОЙ СИСТЕМЫ [executive directive]. Предписание, инициирующее выполнение *операционной системой* определенных действий. Например, создать новый каталог или напечатать файл. К. о. с., как правило, состоит из имени команды и, возможно, параметров, содержащих необходимую уточняющую информацию. Например, команда печати файла, имеющего имя report.txt, в *операционной системе MS-DOS* имеет вид: `copy report.txt prn` и означает скопировать файл report.txt на принтер. Здесь `copy` — имя команды копирования файла, а `report.txt` и `prn` — параметры, указывающие, что копировать и куда. К. о. с. либо вводятся пользователем или *оператором ЭВМ* с клавиатуры в *командную строку* на экране дисплея, либо могут быть внесены в специальный *командный файл*, при исполнении которого они выполняются так, как будто вводятся с клавиатуры. К. о. с. обрабатываются *командным процессором*, являющимся составной частью операционной системы. Некоторые операционные системы имеют команды двух типов: внутренние, выполняемые самим командным процессором, и внешние, реализуемые отдельными специальными программами. См. *внутренняя команда MS-DOS, внешняя команда MS-DOS*. Повсеместное применение *оболочек операционных систем* привело к тому, что К. о. с. используются в основном в командных файлах, и рядовому пользователю для успешной работы на компьютере знание этих команд не обязательно. См. *интерфейс командной строки*

КОМАНДА ПАУЗЫ [pause instruction]. *Команда*, предписывающая временное прекращение выполнения программы. Например, на *персональных компьютерах* К. п. выдается нажатием клавиши <Pause> (<Break>). После чего программа прекращает работу, возобновить которую можно нажатием клавиши <Enter>

КОМАНДА ПЕРЕДАЧИ УПРАВЛЕНИЯ, **команда ветвления**, **команда перехода [jump instruction**, **branch instruction]**. *Команда*, предписывающая либо *передачу управления* без всякого условия, по заранее определенному адресу *(К. безусловной п. у.)*, либо предварительную проверку условия, которое должно быть удовлетворено, чтобы такая передача управления состоялась *(К. условной п. у.)*. Если условие передачи управления не выполнено, то управление передается команде, следующей сразу за К. п. у.

КОМАНДА ПЕРЕСЫЛКИ [sending instruction]. *Машинная команда*, определяющая пересылку данных из одного места *основной памяти* в другое

КОМАНДА ПЕРЕХОДА [jump instruction, **branch instruction]**. То же, что *команда передачи управления*

КОМАНДА ПРЕРЫВАНИЯ [trap instruction]. *Команда*, вызывающая *прерывание*

КОМАНДА УСЛОВНОГО ПЕРЕХОДА, **команда условной передачи управления [condition jump instruction**, **condition branch instruction]**. См. *команда передачи управления*

КОМАНДНАЯ КЛАВИША [accelerator key, **keyboard accelerator**, **keyboard shortcut**, **shortcut key]**. *Клавиша* или комбинация двух (максимум трех) клавиш, одновременное нажатие которых вызывает определенные действия программы или *вычислительной системы*. В качестве К. к. обычно используются *функциональные* и *управляющие клавиши* клавиатуры либо сочетания этих клавиш с буквенными. Например, одновременное нажатие сочетания клавиш <Ctrl>+<Alt>+<Del> вызывает перезагрузку *операционной системы MS-DOS*. К. к., управляющие конкретным приложением, обычно называют *клавишами быстрого вызова*

КОМАНДНАЯ СТРОКА [command line]. *Строка* экрана дисплея, предназначенная для записи вводимых пользователем *команд операционной системы* и *запуска программ* с клавиатуры. Когда операционная система готова к приему команд, она выдает в начале К. с. приглашение, которое может содержать некоторую нужную пользователю информацию о текущем состоянии системы. Например, в приглашении MS-DOS C:\EXE> сообщается о *текущем дисководе* C: и *текущем каталоге* EXE.

КОМАНДНЫЙ ПРОЦЕССОР, **процессор командного языка [command processor**, **shell]**. 1. Часть *операционной системы*, обрабатывающая *команды операционной системы*, вводимые пользователем с терминала или содержащиеся в *командном файле*. Например, К. п. *операционной системы MS-DOS* — программа command.com. 2. То же, что *контроллер*

КОМАНДНЫЙ ФАЙЛ, **пакетный файл [command file**, **batch file]**. *Исполняемый файл*, содержащий *команды операционной системы*. Обычно К. ф. имеет расширение bat и представляет собой *текстовый файл*, каждая строка кото-

рого — команда операционной системы или имя исполнимой программы. К. ф. выполняется *командным процессором.* При этом часто допускается использование в К. ф. условных операторов, циклов и других управляющих конструкций. В этом случае К. ф. также называют *сценарием.* Например, ниже приведен текст К. ф., реализующего следующий сценарий: если программа helpshow.exe доступна, то очищается экран, вызывается программа helpshow.exe с параметром read.me и еще раз очищается экран. В противном случае ничего не делается

```
@echo off
if not exist helpshow.exe goto Done
cls
helpshow read.me
cls
:Done
```

КОМИТЕТ ПО ИНЖЕНЕРНЫМ ВОПРОСАМ ИНТЕРНЕТА [Internet Engineering Task Force (IETF)]. То же, что *Инженерный консорциум разработчиков стандартов Интернета*

КОМИТЕТ IETF [Internet Engineering Task Force (IETF)]. То же, что *Инженерный консорциум разработчиков стандартов Интернета*

КОММЕНТАРИЙ [comment]. Ограниченный специальным образом фрагмент программы, предназначенный для чтения человеком и игнорируемый при трансляции. К. служит для включения в текст программы пояснений, облегчающих человеку ее чтение и анализ. В каждом *языке программирования* имеется свой синтаксис К. В языках Паскаль и Си К. заключают в скобки, например, `{...}`или `(*...*)` — в Паскале, `/*...*/` либо `//...//` — в Си. В Фортране каждый К. занимает отдельную строку и начинается с символа `C` или `*` в ее первой позиции. См. *закомментировать*

КОММЕРЧЕСКИЙ ПРОГРАММНЫЙ ПРОДУКТ [program product, software]. *Программный продукт*, изготовленный для продажи

КОММУНИКАТОР [communicator]. 1. *Программное средство* управления взаимодействием группы *параллельных вычислительных процессов* в *MPI-программах.* 2. Информационная *структура данных*, описывающая множество процессов, которые могут обмениваться сообщениями друг с другом

КОММУНИКАЦИОННЫЙ ДИАМЕТР [bandwidth, communication range]. Максимальный путь между любыми двумя *узлами сети*

КОММУТАЦИЯ [switching]. Установление связи между двумя или несколькими устройствами из определенного множества устройств путем создания временных, а не постоянных соединений

КОММУТАЦИЯ ПАКЕТОВ [packet switching]. Принцип организации связи по *сети передачи данных*, основанный на перемещении информации по сети в виде отдельных *пакетов*. При этом в каждом промежуточном *узле сети* определяется маршрут пакета, приходящего от соседнего узла к другому соседнему узлу. В результате пакеты от одного отправителя могут приходить к получателю разными путями, а нагрузка на *линии связи* и узлы сети распределяется более равномерно по сравнению с *коммутацией соединений*. С другой стороны, среднее время доставки пакетов оказывается выше, а сами пакеты могут приходить в произвольном порядке. Ср. *коммутация соединений*

КОММУТАЦИЯ СОЕДИНЕНИЙ [connection switching]. Принцип организации связи по *сети передачи данных*, основанный на предварительном установлении соединения между взаимодействующими узлами с последующей передачей информации по этому соединению. К. с. — основа телефонной связи. Ср. *коммутация пакетов*

КОМПАКТ-ДИСК [compact disk (CD), CD-ROM]. 1. *Лазерный диск* стандартного размера (5,25 дюйма), применяющийся в аудиосистемах и в *персональных компьютерах*. 2. То же, что *лазерный диск*

КОМПИЛИРУЕМЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ [compiled programming language]. *Язык программирования* (например, C++), который предполагает трансляцию программы из исходного кода в машинно-исполнимую форму. Процесс трансляции называется компиляцией. Компиляция должна быть выполнена до того, как программа может начать выполняться. См. *компиляция, компилятор*. Противоп. *интерпретируемый язык программирования*

КОМПИЛЯТОР [compiler]. Программа-*транслятор*, выполняющая *компиляцию программных модулей*. В отличие от *интерпретатора* К. только преобразует программу, составленную на *языке программирования высокого уровня*, в программу на *машинном языке* или языке, близком к машинному, не участвуя в ее исполнении. На вход К. поступает *исходный модуль*, который после компиляции преобразуется в *объектный модуль*. К. являются неотъемлемой частью *системы программирования*. Существуют различные виды К.: интерпретирующие (пошаговые) К., осуществляющие последовательную независимую компиляцию каждого отдельного оператора *исходной программы*; оптимизирующие К., повышающие эффективность *объектных программ*, например, за счет вынесения из циклов команд, результаты действия которых не изменяются при повторении цикла; *отладочные К.*, облегчающие пользователю *отладку программ*

КОМПИЛЯЦИЯ [compilation]. *Трансляция* программы или отдельного программного модуля, составленных на *языке программирования высокого уровня (исходная программа, исходный модуль)*, в программу или модуль на *машинном языке* или языке, близком к машинному *(объектная программа, объектный модуль)*. В процессе К. программа преобразуется в промежуточную форму, к

которой впоследствии необходимо присоединить библиотечные средства, содержащие *стандартные подпрограммы* и процедуры, а если нужно, то можно добавить любые другие модули, написанные самим пользователем и скомпилированные в объектные модули, возможно, с иных языков высокого уровня. Существуют *интерпретируемые языки программирования*, для которых К. затруднительна или невозможна, а потому не применяется. См. *компоновка*

КОМПЛЕКС ПРОГРАММ [program complex, routine set]. Совокупность *программ*, образующих в некотором смысле единое целое. Например, К. п. технического обслуживания компьютера представляет собой набор вспомогательных программ отладки, проверки и диагностики ЭВМ или ее отдельных устройств. Эти программы могут работать под управлением *операционной системы* или автономно

КОМПЛЕКСНЫЕ ИСПЫТАНИЯ ПРОГРАММНОГО ПРОДУКТА [complex tests of the program product]. Завершающий этап разработки *программного продукта*, состоящий в проведении испытательных прогонов программы с целью убедиться, что она действительно решает ту задачу, для которой предназначена, и выдает правильный ответ при любых условиях, а также оценить основные параметры функционирования. К. и. п. п. проводятся после отладки и *автономных испытаний*. Эти испытания предусматривают проверку правильности работы программы в целом, взаимной увязки ее отдельных частей, а также ее взаимодействия с другими программами, если это необходимо. Во время К. и. п. п. выясняется способность функционирования программы в условиях конкретной *вычислительной системы*: заданной *конфигурации компьютера*, определенной версии *операционной системы*, реальных наборов данных. При этом проводятся оценки основных параметров программы, например, требуемые минимальные и максимальные ресурсы памяти, времени и т. п. Ср. *автономные испытания программы*. См. *тестирование программы*

КОМПЛЕМЕНТАРНАЯ СТРУКТУРА МЕТАЛЛ—ОКСИД—ПОЛУПРОВОДНИК, КМОП-структура, CMOS-структура [complementary metal—oxide—semiconductor (CMOS)]. Полупроводниковое устройство, состоящее из двух металлоксидных транзисторов N-типа и P-типа, установленных на одном кремниевом кристалле. Для подобных устройств, обычно используемых в устройствах *оперативной памяти* и средствах коммутации, характерны очень высокая скорость срабатывания и крайне низкий уровень потребления питания. Однако они "боятся" воздействия статического электричества

КОМПОЗИЦИЯ [composition]. В *унифицированном языке моделирования* UML форма *ассоциации*, описывающая отношение типа "часть—целое" между объектами. Отношение К. является более строгим по сравнению с отношением *агрегации*: часть может принадлежать только одному целому и прекращение существования целого влечет прекращение существования его частей. К. может быть рекурсивной

КОМПОНЕНТ [component]. То же, что *компонент программы*

КОМПОНЕНТ ПРОГРАММЫ, **компонент [component]**. Часть программы, которая структурно отделена от остальных частей и может использоваться в нескольких программах. Например, *библиотека динамической компоновки* является К. п. *Распределенное приложение* обычно состоит из нескольких К. п., которые могут выполняться на разных компьютерах. См. *распределенная компонентная модель объекта*

КОМПОНЕНТНАЯ МОДЕЛЬ ОБЪЕКТОВ, **модель COM [Component Object Model (COM)]**. Разработанная корпорацией Microsoft технология, определяющая методы создания и взаимодействия *программных объектов* различных типов. К. м. о. является основой таких программных *технологий*, как OLE и ActiveX. Технология COM обеспечивает взаимодействие компонентов программного обеспечения, действующих в разных процессах или на разных компьютерах. Основная идея К. м. о. состоит в установлении эффективной связи между компонентами, разделенными границами *адресного пространства*. Технология COM отличается от *технологии CORBA* тем, что взаимодействие между объектами устанавливается непосредственно, а не с помощью промежуточного агента. Расширение этой технологии (см. *Распределенная компонентная модель объектов*) позволяет осуществлять взаимодействие между объектами, хранящимися на разных компьютерах. См. *связывание и внедрение объектов, элементы управления ActiveX*

КОМПОНОВКА, **редактирование связей [linking**, **linkage editing]**. Процесс сборки *загрузочного модуля* из полученных в результате раздельной компиляции *объектных модулей* с автоматическим поиском и присоединением *библиотечных подпрограмм* и процедур. В процессе К. программа собирается в единое целое. Устанавливаются связи между отдельными модулями с помощью *команд перехода* и *команд возврата*, у которых формируются *логические адреса*, происходит организация *общих блоков* и устанавливаются пути передачи данных от модуля к модулю и в подпрограммы. К. выполняется специальной программой — *компоновщиком*

КОМПОНОВЩИК, редактор связей [linker, **linkage editor]**. *Программа*, выполняющая *компоновку*. В результате работы К. из одного или нескольких *объектных модулей* с привлечением *библиотечных программ* и *стандартных подпрограмм* формируется *загрузочный модуль*, представляющий исполняемый вариант программы, который при необходимости может быть загружен в *оперативную память загрузчиком* для исполнения. Существуют программы, сразу выполняющие и компоновку, и загрузку. См. *компонующий загрузчик*

КОМПОНУЮЩИЙ ЗАГРУЗЧИК [linking loader]. *Загрузчик*, выполняющий *компоновку* задачи из *объектных модулей* непосредственно в *оперативной памяти* во время загрузки. Обычно К. з. выполняет следующие основные функции: 1) распределяет пространство оперативной памяти для програм-

мы; 2) связывает вместе части программы, представленные отдельными объектными модулями, вычисляя значения *внешних* (межмодульных) *ссылок* различных объектных модулей; 3) настраивает адреса подготовленной программы, заменяя все *относительные адреса*, выработанные компилятором, соответствующими адресами фактически распределенной памяти; 4) физически размещает подготовленную версию машинного кода программы в памяти и передает управление на первую команду программы. Ср. *загрузчик, компоновщик*

КОМПРЕССОР-ДЕКОМПРЕССОР, **кодек [compressor-decompressor**, **codec]**. 1. *Аппаратное* или *программное средство* преобразования исходного видео- или звукового *аналогового сигнала* в сжатую *цифровую форму данных* и обратно. 2. В *мультимедиа* — программа, которая сжимает звуковые и графические файлы для эффективного сохранения или передачи и распаковывает их для воспроизведения

КОМПЬЮТЕР [computer]. 1. То же, что *вычислительная машина*. 2. То же, что *электронная вычислительная машина*

КОМПЬЮТЕРНО-ТЕЛЕФОННАЯ ИНТЕГРАЦИЯ [computer-telephony integration, (CTI)]. Направление в современной телефонии, основанное на широком применении компьютерных технологий: передача звука по компьютерным сетям, управление телефонным оборудованием с помощью компьютерных программ, широкое применение *баз данных*, доступ к ресурсам Интернета и т. д. См. *интернет-телефония*

КОМПЬЮТЕР С ОДНОРОДНЫМ ДОСТУПОМ К ПАМЯТИ, UMA-компьютер [uniform memory access computer, UMA-computer]. *Многопроцессорная ЭВМ* с *разделяемой памятью*, имеющая единое *адресное пространство* и одинаковые для всех процессоров параметры доступа к модулям памяти

КОМПЬЮТЕР С УПРОЩЕННОЙ СИСТЕМОЙ КОМАНД, **RISC-компьютер [reduced instruction set computer (RISC)]**. *Компьютер*, построенный на базе процессора *архитектуры RISC*, как правило, имеет высокую тактовую частоту

КОМПЬЮТЕР СО СЛОЖНОЙ СИСТЕМОЙ КОМАНД, **CISC-компьютер [complex instruction set computer (CISC)]**. *Компьютер*, построенный на базе процессора *архитектуры CISC* со сложной и мощной системой команд

КОМПЬЮТЕРНАЯ ГРАФИКА, **машинная графика [computer graphics]**. 1. Система методов, алгоритмов, *программных и аппаратных средств* для ввода, обработки и отображения *графической информации*, а также для преобразования данных в графическую форму. К. г. является неотъемлемой составной частью современных *вычислительных систем*. На базе К. г. работают *системы автоматизированного проектирования* и мультимедиа. Средства К. г. используются в *дружественном* и *графическом интерфейсах пользователей*, а

также в *прикладных программах*, для того чтобы показать в наглядном виде математические или другие объекты, содержательно не являющиеся изображениями, например, графики функций, диаграммы, представленные в виде картинки ситуации в вычислительной системе и т. п. 2. Обработка *графической информации* с помощью *компьютера*. Благодаря К. г. изображения можно создавать, изменять и стирать точно так же, как и текст. Можно работать с цветом и яркостью всего изображения и отдельных его фрагментов. Наконец, можно создавать на экране дисплея движущееся цветное изображение. Наиболее распространенный вид К. г. связан с компьютерными играми

КОМПЬЮТЕРНАЯ ИГРА [computer game]. 1. Взаимодействие человека с компьютером по определенному игровому алгоритму с целью развлечения, обучения или тренировки. К. и. ведется под управлением *игровой программы* и обычно имеет форму диалога. На экране дисплея наглядно воспроизводится игровая ситуация. Играющий анализирует эту ситуацию и реагирует на нее с помощью клавиатуры, мыши, джойстика или других специальных *устройств ввода*. Ответные ходы или соответствующее изменение игровой ситуации компьютер воспроизводит на экране, часто со звуковым сопровождением. Игровой компонент выполняет важную роль в процессе компьютерного обучения. Хорошо спроектированные развлекательные К. и. способны развивать познавательные, художественные и профессиональные способности человека. Различают несколько типов К. и. 1) Логические, близкие по сюжетам и целям к традиционным математическим упражнениям и головоломкам. 2) Аркадные, цель которых — уничтожение движущихся объектов. 3) Приключенческие, предлагающие игроку действовать в соответствии с определенным сюжетом приключенческого, сказочного, фантастического или мистического характера. 4) Стратегические, основанные на моделировании управления городом, армией, страной и т. п. 5) Игры-имитации, воспроизводящие управление автомобилем, самолетом, вертолетом и т. п. Многие современные К. и. сочетают свойства перечисленных типов. Существуют К. и., в которых могут участвовать несколько партнеров, взаимодействующих между собой посредством *компьютерной сети*. 2. *Программный продукт*, обеспечивающий компьютерную игру. В настоящее время рынок насчитывает сотни тысяч К. и.: от простой программы-угадайки для дошкольников до шахматной *интерактивной системы* или использующей весь современный арсенал мультимедиа приключенческой игры, которая погружает игрока в виртуальную реальность

КОМПЬЮТЕРНАЯ ПЛАТФОРМА, **платформа [computer platform**, **platform]**. Тип *персонального компьютера*, на котором может быть установлен *программный продукт*. Например, IBM PC, Macintosh, и т. д. Ср. *программная платформа*

КОМПЬЮТЕРНАЯ СВЯЗЬ [computer communication]. Передача информации от компьютера к компьютеру по *каналам связи*. Связь осуществляется

электронными средствами либо по каналам общественных телекоммуникаций (телефонной, радио- и спутниковой связи), либо по специальным кабелям и проводам. К. с. отличается рядом достоинств: быстрый, точный и прямой обмен информацией; отправление и получение информации занимает значительно меньше времени, чем по почте или через посыльного; быстрая обработка больших объемов информации — объем информации, который может хранить и обрабатывать компьютер, колоссален; высокая скорость и простота поиска нужных данных. Чтобы осуществить связь между двумя компьютерами по телефону, необходимы аппаратное устройство, называемое *модемом*, и специальное *программное обеспечение*. Постоянно действующим средством К. с. являются *компьютерные сети*. В сетях К. с. имеются специальные компьютеры — *узлы сети* с установленным на них сетевым *программным обеспечением*, предназначенным для поддержки работы сети. Определенные *области памяти* этих компьютеров отводятся для сбора передаваемых сообщений, поэтому узлы сети иногда называют почтовыми ящиками. Средства К. с. могут работать в режимах *электронной почты* и телеконференции. См. *Интернет*

КОМПЬЮТЕРНАЯ СЕТЬ, **вычислительная сеть**, **сеть ЭВМ [computer network]**. *Вычислительная система*, включающая в себя несколько *компьютеров*, *терминалов* и других *аппаратных средств*, соединенных между собой обеспечивающими *передачу данных линиями связи*. Компьютеры могут соединяться друг с другом непосредственно либо через промежуточные компьютеры. Помимо компьютеров, за которыми работают люди *(рабочих станций)*, в К. с. могут быть компьютеры, за которыми никто не работает. Более того, у них могут отсутствовать видеомонитор и клавиатура. Такие компьютеры используются как управляющие центры в сети и концентраторы данных. Обычно их называют серверами. В зависимости от конструкции сети и возможностей сетевого *программного обеспечения* пользователь одного компьютера или терминала может получить доступ к данным, хранящимся в памяти другого компьютера, доступ к памяти и *центральному процессору* более мощного компьютера для выполнения своей *прикладной программы*, объединенные ресурсы всех включенных в нее компьютеров. Для организации дистанционной связи в сети необходимы *программные средства* передачи данных и аппаратные устройства — модемы, *сетевые карты*. Если в сеть объединяются компьютеры и терминалы, находящиеся в одном помещении, здании или на ограниченной территории, и при этом в качестве устройств передачи данных используются специальные кабели и провода, то говорят, что компьютеры образуют *локальную вычислительную сеть*. Обычно такая сеть состоит из одной мощной *главной ЭВМ* и нескольких менее мощных. При этом пользователи менее мощных компьютеров и терминалов имеют совместный доступ к вычислительным ресурсам главной ЭВМ и ее обширной памяти. Подобные сети часто организуются в учреждениях и учебных заведениях. Если в сеть объединяются ЭВМ, настолько удаленные друг от друга,

что для передачи данных между ними приходится использовать телекоммуникации (телефонную, радио- или спутниковую связь), то говорят, что компьютеры образуют *глобальную* (или *региональную*) *вычислительную сеть*. Для диспетчерского управления такой сетью или коммутации линий связи служат специально установленные компьютеры с сетевым программным обеспечением, называемые *узлами сети*. Примерами сетей служат единая система заказа билетов на авиалиниях или сеть, связывающая компьютеры банка и его филиалов, расположенных в разных городах и даже странах. Широкое распространение получила во всем мире глобальная К. с. Интернет

КОМПЬЮТЕРНЫЙ ВИРУС, **вирус [computer virus, virus]**. Программа или присоединяемый к другим программам набор команд, которые воспроизводят и распространяют свои копии в компьютерах или *компьютерных сетях* и выполняют некоторые действия, нежелательные для законных пользователей. К. в. "заражает" программы в *оперативной памяти* и на дисках. Кроме заражения вирус может вывести на экран "постороннюю" надпись, заставить неожиданно зазвучать динамик, испортить файлы, отформатировать диск или причинить иной ущерб. Уже известно множество типов К. в., различающихся способами распространения, "агрессивностью", способностью преодолевать защиту антивирусных программ и другими характеристиками. По методу заражения К. в. могут быть программными или загрузочными. Программный вирус записывается в тело программы и при загрузке ее в память тоже резидентно загружается и заражает все находящиеся в оперативной памяти программы, записывая себя в их файлы на диске. Загрузочный вирус записывает себя скрытым образом на диск, оставляя в дисковом загрузчике ссылку на себя, и активизируется во время первого обращения к диску. Он "заражает" каждый диск, к которому идет обращение. Существуют специальные *антивирусные программы*, позволяющие обнаруживать и устранять К. в. К сожалению, злонамеренными программистами непрерывно создаются новые вирусы, поэтому антивирусные программы и другие средства защиты нуждаются в постоянном обновлении. К. в. являются серьезной угрозой общественной безопасности. См. *макровирус*

КОНВЕЙЕР [pipeline]. 1. Последовательность команд, программ или процессов, построенная таким образом, что *выходные данные* одного из них являются *входными данными* для следующего. Конвейер реализуется обычно с помощью *именованного канала*. 2. *Архитектура ЭВМ*, в которой *центральный процессор* совмещает выполнение разных фаз нескольких команд. Например, одновременно с тем, как выполняется основное действие текущей команды, в память записывается результат выполнения предыдущей команды и одновременно из памяти читаются данные для выполнения следующей команды

КОНВЕЙЕРНАЯ ОБРАБОТКА [pipelined processing]. Способ организации исполнения команд или *обработки данных*, основанный на принципе *конвейера*. Каждая команда или операция над числами выполняется в виде по-

следовательности подопераций. Каждая из подопераций выполняется на отдельной части *арифметико-логического устройства*, называемой *ступенью* или *сегментом конвейера*, а общее число ступеней — *длиной конвейера*. Ступени расположены в порядке выполнения подопераций. После выполнения первой подоперации первая ступень передает результаты своей работы второй ступени, а сама берет новый объект обработки и т. д. При достижении объектом конца конвейера он окажется полностью обработанным. Например, пусть требуется сложить два массива чисел. Предположим, что операция суммирования включает три подоперации: выравнивание порядков, сложение мантисс и нормализацию. Если каждая подоперация выполняется за один такт, то вся операция сложения первых двух чисел выполнится за три такта. Однако каждый последующий такт на выходе из конвейерного устройства будет появляться сумма очередной пары элементов. Поэтому для сложения двух массивов по 100 элементов способом К. о. потребуется 102 такта вместо 300 тактов при использовании последовательного устройства. Ср. *параллельная обработка, параллельные вычисления*

КОНВЕРТОР [converter]. То же, что *преобразователь*

КОНЕЦ ПЕРЕДАЧИ [end of transmission (EOT)]. Символ, обозначающий прекращение передачи данных. См. *американский стандартный код обмена информацией*

КОНЕЦ ФАЙЛА [end of file (EOF)]. Символ, указывающий конец *текстового файла*. См. *американский стандартный код обмена информацией*

КОНЕЧНЫЙ АВТОМАТ, машина состояний [automaton, finite-state machine]. Один из способов формального задания алгоритмов. Суть работы К. а. сводится к следующему: автомат находится в одном из возможных состояний, число которых конечно, и последовательно воспринимает входные символы из конечного входного алфавита. Для каждой пары (состояние, входной символ) определена пара (выполняемое действие, новое состояние). Таким образом, воспринимая очередной входной символ, К. а. выполняет соответствующее действие и переходит в новое состояние (или остается в прежнем). Обычно К. а. задают в виде таблицы, строки которой помечены названиями состояний, столбцы помечены входными символами, а в ячейках таблицы указаны выполняемые действия и (если нужно) новые состояния. Например, приведенный в табл. К.3 К. а. вычисляет машинное представление *вещественного числа x* по его изображению в *позиционной двоичной системе счисления*, т. е. по последовательности двоичных цифр 0 и 1, среди которых может присутствовать один символ "." (точка). До начала работы автомата переменная x имеет значение 0, а вспомогательные переменные p и q имеют значения 1 и 0.5 соответственно (для записи действий использован синтаксис *языка программирования Паскаль*)

Таблица К.3. *Пример конечного автомата*

Состояния		Входные символы		
Название	**S**	**"0"**	**"1"**	**"."**
Начальное	1		x:=p; S:=2	S:=3
Целая часть	2	p:=p*2	x:=x+p; p:=p*2	S:=3
Дробная часть	3	q:=q/2	x:=x+q; q:=q/2	

К. а. не является универсальным способом задания алгоритмов, т. е. не всякий алгоритм может быть задан К. а., однако во многих случаях К. а. очень удобен, например, в задачах трансляции, преобразования форматов и др. К. а. можно задать не только таблицей, но и графом, вершинами которого являются состояния, а на дугах указаны условия перехода и выполняемые действия. Например, на рис. К.4 изображен в нотации *унифицированного языка моделирования* тот же К. а., что и в табл. К.3. Ср. *машина Тьюринга*

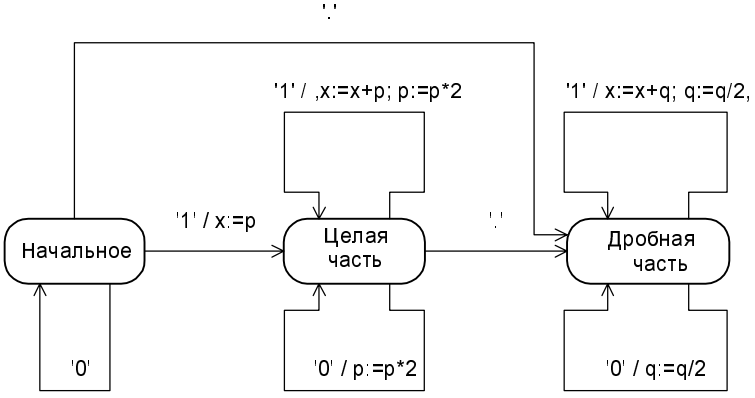

Рис. К.4. Пример конечного автомата

КОНЕЧНЫЙ ПОЛЬЗОВАТЕЛЬ [end user]. 1. Лицо, для которого предназначена информация, выводимая компьютером. Ср. *пользователь*. 2. Лицо, которое пользуется компьютером или программами (в отличие от тех, кто занимается их разработкой и сопровождением). Ср. *пользователь*

КОНКАТЕНАЦИЯ [concatenation]. Операция последовательного объединения двух или более *элементов данных* в один. Например, К. строк — объединение символьных или битовых строк в одну строку без нарушения порядка следования знаков в каждой строке. А К. файлов — последовательное слия-

ние содержимого двух однотипных файлов в один файл. В операционных системах и языках программирования существуют специальные команды или операции К. Например, в языке Паскаль операция + объединяет отдельные символьные данные в единую строку

КОНСОЛЬ [console]. 1. Пульт управления *вычислительной системой.* Обычно К. называется терминал, с которого пользователь (оператор) может управлять *операционной системой.* 2. Программа, которая позволяет работать с какой-либо системой в *интерактивном режиме.* Например, консоль PVM — программа, позволяющая работать с параллельной виртуальной машиной в интерактивном режиме.

КОНСОРЦИУМ ПО ОБЪЕКТНЫМ ТЕХНОЛОГИЯМ, консорциум OMG [object management group (OMG)]. Некоммерческая организация компаний, выпускающая стандарты в области *объектно-ориентированного программирования.* Например, стандарт *унифицированного языка моделирования* UML выпущен консорциумом OMG

КОНСОРЦИУМ OMG [object management group (OMG)]. То же, что *консорциум по объектным технологиям*

КОНСОРЦИУМ W3C [World Wide Web consortium, WWW consortium (W3C)]. То же, что *консорциум World Wide Web*

КОНСОРЦИУМ WORLD WIDE WEB, консорциум W3C [World Wide Web consortium, WWW consortium (W3C)]. Международная организация, объединяющая более 500 фирм во всех уголках мира, вклад которых в расширение Всемирной сети получил международное признание. Среди членов К. W3C корпорации Apple, IBM, Microsoft, Netscape, Sony, Xerox и др. К. W3C был основан в октябре 1994 г., чтобы способствовать развитию сетевых технологий (разработке общих протоколов, спецификаций, инструментария и т. п.) и тем самым полностью раскрыть потенциал сети как форума для информации, коммерции, коммуникации и общественного понимания. Например, за последние пять лет К. W3C разработано более 20 технических спецификаций для сетевой инфраструктуры (компьютеров, телекоммуникаций, технологий мультимедиа). На рис. К.5 представлена домашняя страница веб-сайта К. W3C

КОНСТАНТА [constant]. *Элемент данных*, присутствующий в тексте программы и не меняющий своего значения при многократном ее использовании. К. могут быть численными, например, *целая К.* (К. в виде *числа целого типа*) или *вещественная К.* (К. в виде *числа вещественного типа*), логическими *(логическое значение)*, символьными (строка) и адресными. Тип К. однозначно определяется способом ее записи. Например, в выражении `x + 3.1416` К. `3.1416` — число вещественного типа с фиксированной точкой. А в операторе `Writeln (Output, 'До свидания.')` К. является строка `До свидания.` См. *символьная константа*

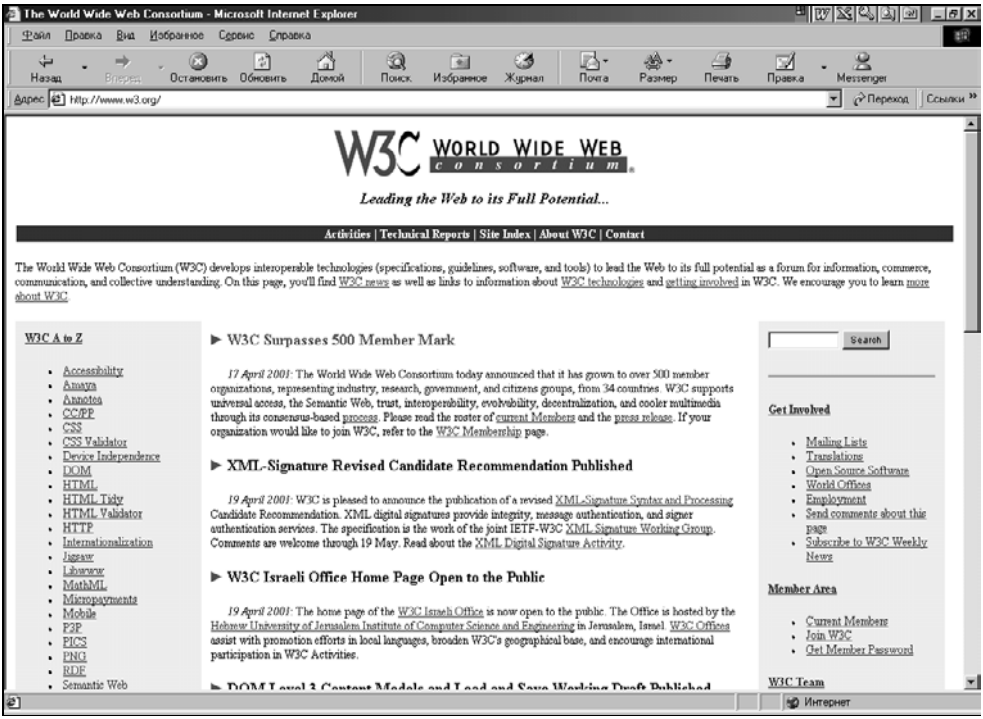

Рис. К.5. Домашняя страница W3C

КОНСТРУИРОВАНИЕ [construction]. Фаза *процесса разработки программного обеспечения*, в течение которой проводятся детальное *проектирование*, *реализация* и *тестирование*

КОНСТРУКТОР [constructor]. Операция с областью действия в масштабе класса, которая создает и инициализирует *экземпляр класса*

КОНТЕЙНЕР [container]. Объект, основным назначением которого является содержание других объектов. Например, папка в дереве каталогов является К., предназначенным для содержания других папок и файлов. Другой пример: документ, содержащий связанные или внедренные объекты, созданные другим приложением. См. *связывание и внедрение объектов*

КОНТЕКСТНОЕ МЕНЮ [context menu, shortcut menu]. Меню, которое содержит команды, применимые в данном контексте. Например, во многих приложениях Windows, если выделить какой-либо объект и щелкнуть правой кнопкой мыши, то появится К. м., содержащее команды, выполняемые с выделенным объектом в его текущем состоянии. На рис. К.6 приведен пример К. м. в приложении MS Word. К. м. является удобным средством *интерфейса пользователя*, заметно ускоряющим работу с приложением

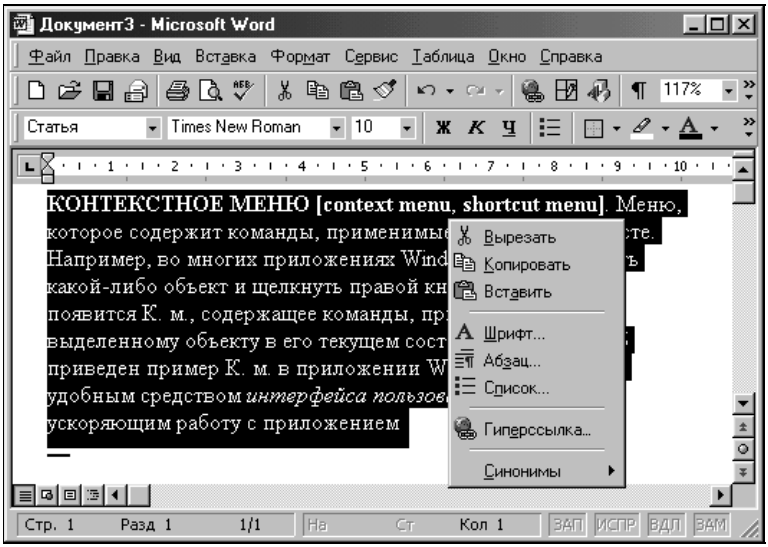

Рис. К.6. Контекстное меню для выделенного абзаца

КОНТЕКСТНО-ЗАВИСИМАЯ СПРАВКА [context-sensitive help]. *Справочная система*, которая выдает пользователю справочную информацию с учетом текущего контекста его работы. Например, если выделен некоторый объект, то при нажатии клавиши <F1> выводится список разделов справки, имеющих отношение к работе с объектами данного типа

КОНТРОЛЛЕР [controller]. Специализированный процессор, автоматически управляющий работой или согласующий работу подключенных к нему устройств. К К. обычно подключаются *внешние устройства*. К. также является составной частью *каналов связи*. Одна из его функций заключается в *форматировании данных* для передачи по каналу связи или для записи на *носитель данных*. Чаще всего К. технически выполняется в виде отдельной платы. См. *контроллер жесткого диска*

КОНТРОЛЛЕР ЖЕСТКОГО ДИСКА [hard disk controller]. В *персональных компьютерах* — *контроллер*, служащий для согласования работы *центрального процессора* и *жесткого диска*. К. ж. д. может располагаться на жестком диске, на *материнской плате*, на отдельной плате, вставляемой в разъем материнской платы. Для ускорения *выборки данных* с жесткого диска К. ж. д. может иметь *кэш-память*. В *персональных компьютерах* применяются контроллеры следующих типов (в порядке возрастания скорости передачи данных): IDE; ESDI; SCSI

КОНТРОЛЬ КАЧЕСТВА [quality assurance (QA)]. Процесс, с помощью которого можно обеспечить достижение заявленного уровня качества, в част-

ности, при *разработке программ.* Сокращение QA часто используется также для обозначения подразделения, обеспечивающего К. к.

КОНТРОЛЬ НА ЧЕТНОСТЬ [parity check]. То же, что *контроль по четности*

КОНТРОЛЬ ПО ЧЕТНОСТИ, **контроль на четность**, **контроль четности [parity check]**. Метод контроля правильности хранения или *передачи данных*, основанный на следующем приеме. К определенному количеству (группе) разрядов *двоичного кода* данных добавляется дополнительный контрольный бит. Значение этого бита устанавливается равным 0, если число единиц в группе бит четное, и равным 1, если число единиц нечетное. При расшифровке информации содержимое контрольного бита игнорируется. Такой прием приводит к расширению кода (биты, содержащие информацию, плюс контрольный бит). Проверка искажений при хранении или *передаче данных* сводится к подсчету количества единиц в расширенных группах бит. Оно всегда должно быть четным. В некоторых случаях, исходя из особенностей *аппаратных средств*, устанавливается контроль по нечетности, при котором значение дополнительного контрольного бита равно 0, если число единиц в группе бит нечетное, и равно 1, если число единиц четное

КОНТРОЛЬ ЧЕТНОСТИ [parity check]. То же, что *контроль по четности*

КОНТРОЛЬНАЯ ТОЧКА [checkpoint]. 1. Место в программе, где осуществляется проверка либо запись во *внешнюю память* данных, необходимых для повторного возобновления работы программы. Обычно считается, что К. т. располагается перед первым оператором или первой командой, осуществляющими такую проверку или запись. В случае *машинного сбоя* или нежелательного хода *вычислительного процесса* осуществляется откат и процесс восстанавливается. При этом результаты работы компьютера после "прохождения" последней К. т. теряются. Однако для больших задач такая методика эффективна, ибо в противном случае пришлось бы решать задачу с самого начала. См. *образ задачи, откат.* 2. То же, что *образ задачи.* 3. То же, что *точка прерывания.* См. *отладка программы*

КОНТУР [path]. Представление сложной линии средствами *векторной графики*, обычно основанными на использовании специального математического аппарата *кривых Безье*

КОНТУРНЫЙ РЕЖИМ [wire frame]. Способ отображения на экране объектов *векторной графики* в виде проволочных каркасов — без толщины и цвета обводки и без заливки

КОНФЕРЕНЦИЯ [conference]. 1. То же, что *телеконференция.* 2. Телефонное соединение, охватывающее трех и более *абонентов сети.* См. *устройство управления конференциями*

КОНФИГУРАЦИЯ [configuration]. 1. Компоновка системы с четким определением характера, количества, взаимосвязей и основных характеристик ее

функциональных элементов. Например, *конфигурация компьютера.* 2. Набор аппаратных и программных установок (например, положений переключателей, значений управляющих переменных, опций), задающих режим функционирования программы или устройства. Например, *конфигурация безопасности.* Ср. *профиль оборудования, профиль пользователя*

КОНФИГУРАЦИЯ БЕЗОПАСНОСТИ [security configuration]. Совокупность параметров выбранной *политики безопасности.* К. б. хранятся в виде файлов (имеются стандартные конфигурации), которые можно модифицировать, создавая собственные конфигурации, и применять к компьютерам.

КОНФИГУРАЦИЯ КОМПЬЮТЕРА [hardware configuration]. Совокупность *аппаратных средств* и соединений между ними в *вычислительной системе*, используемая в течение определенного периода эксплуатации. К. к. определяется набором характеристик устройств, входящих в состав вычислительной системы. Например, основными характеристиками *персональных компьютеров* являются тип и *тактовая частота* процессора; наличие *математического сопроцессора*; объем *оперативной памяти*; наличие и объем *кэш-памяти*; тип *системной шины*; наличие *локальной шины*; тип видеоадаптера и объем видеопамяти; тип контроллера и объем памяти *жесткого диска*; количество и тип накопителей для *гибких дисков*; количество и типы портов; тип (цветной или монохроматический), размер по диагонали и размер зерна дисплея. Так, указанная в рекламном объявлении К. к. "Celeron 633, 64Mb RAM, 10Gb HDD, CD 48x, Creative SB-128, fax-modem 56K Genius, монитор Hyundai S570 15" TCO-99, клавиатура, мышь, колонки 120 Вт" означает, что на компьютере установлены: центральный процессор марки Celeron, который работает с тактовой частотой 633 МГц; оперативная память емкостью 64 Мбайт; жесткий диск емкостью 10 Гбайт; привод CD-ROM со скоростью вращения в 48 раз больше стандартной, звуковая плата Sound Blaster, факс-модем со скоростью передачи данных 56 000 бод, монитор производства компании Hyundai с размером экрана 15 дюймов по диагонали; клавиатура; мышь и звуковые колонки мощностью 120 Вт

КОНФИГУРИРОВАНИЕ ОПЕРАЦИОННОЙ СИСТЕМЫ [OS generation]. То же, что *генерация операционной системы*

КОНФЛИКТ [conflict]. То же, что *конфликтная ситуация*

КОНФЛИКТНАЯ СИТУАЦИЯ, **конфликт [conflict situation**, **conflict]**. 1. Состояние *вычислительной системы*, когда две и более задачи одновременно претендуют на одни и те же ресурсы. 2. Ситуация, при которой взаимодействующие программы или разные модули одной программы используют одно имя для обозначения разных объектов. 3. Ситуация, при которой нескольким устройствам выделены (частично) совпадающие ресурсы. Например, разным *устройствам ввода/вывода* назначены пересекающиеся диапазоны адресов *оперативной памяти* 4. Ситуация при моделировании, при

которой атрибут или операция с одним и тем же именем наследуется от нескольких классов одновременно, или когда одно и то же событие запускает более одного перехода в *конечном автомате*, или же любая подобная ситуация, когда при соблюдении стандартных правил возможно появление результатов, обратных ожидаемым. В зависимости от семантики каждого вида элемента модели, конфликт разрешается с помощью определенного правила

КОНФЛИКТ УСТРОЙСТВ [device conflict]. *Конфликт*, происходящий в момент, когда два или более *устройств* пытаются получить доступ к одному и тому же системному ресурсу. Это могут быть линии запросов на прерывание, каналы *прямого доступа* к памяти, порты ввода/вывода и адреса памяти

КОНЦЕНТРАТОР, хаб [concentrator, hub]. *Устройство*, соединяющее несколько *узлов вычислительной сети*. Например, в сети *древовидной топологии* несколько *рабочих станций* подсоединяются по индивидуальным *каналам связи* к одному К., который, в свою очередь, соединяется одним каналом с остальной сетью. Такое использование К. позволяет достичь существенной экономии затрат на *аппаратное обеспечение* каналов связи. Кроме того, К. обычно предоставляет дополнительные возможности, например, автоматическое отслеживание разрыва связи и др.

КОНЦЕПТУАЛЬНЫЙ ПОИСК [concept search]. Поиск документов, имеющих прямое отношение к указанному поисковому слову, а не просто содержащих его. См. *поисковая система*. Ср. *поиск по ключевым словам, морфологический поиск, поиск в определенных полях*

КОНЪЮНКЦИЯ [conjunction]. То же, что *логическое умножение*. См. *логические операции*

КООПЕРАТИВНАЯ МНОГОЗАДАЧНОСТЬ [cooperative multitasking]. Поддерживаемый *многозадачной операционной системой* режим *мультипрограммирования*, при котором работающие приложения регулярно уступают *центральный процессор* операционной системе, чтобы могли выполняться другие приложения. Ср. *вытесняющая многозадачность*

КООПЕРАЦИЯ [collaboration]. В *унифицированном языке моделирования* UML описание взаимосвязи объектов, которые взаимодействуют для обеспечения некоторого поведения, такого как вариант использования или операция. К. изображается на *диаграмме кооперации*

КОП [operation code (opcode)]. То же, что *код операции*

КОПИРОВАНИЕ [copy]. В приложениях *операционной системы Windows* — копирование выделенных данных из документа в *буфер обмена*. См. *выделение*. Ср. *вырезание*

КОПИРОВАНИЕ ДИРЕКТОРИЯ [directory copying]. То же, что *копирование каталога*

КОПИРОВАНИЕ КАТАЛОГА, **копирование директория [directory copying, folder copying]**. Процедура воспроизведения содержимого одного *каталога* в другом каталоге или на другом *носителе данных* с сохранением исходного каталога. Каталог копируется вместе со всеми содержащимися в нем подкаталогами и файлами. При этом исходный каталог сохраняется в памяти на прежнем месте. Ср. *перенос каталога*. Копию каталога можно создать под другим именем. К. к. выполняется с помощью стандартных средств *операционной системы*. Например, в *операционной системе Windows* К. к. удобно выполнять с помощью *перетаскивания мышью* в программе Проводник. См. *копирование файла*

КОПИРОВАНИЕ ФАЙЛА [file copying]. Процедура переноса данных из одного *файла* в другой с сохранением исходного файла. Содержимое копируемого файла можно записать в виде файла с тем же именем в другой каталог того же *носителя данных* или на другой носитель данных либо, изменив имя, в любой каталог любого носителя данных. При этом исходный файл сохраняется в памяти. Ср. *перенос файла*. В одном каталоге можно создать несколько копий одного и того же файла, но под разными именами. Процедуру К. ф. можно осуществлять сразу над группой файлов. К. ф. выполняется с помощью стандартных средств *операционной системы*. Например, в *операционной системе Windows* К. ф. удобно выполнять с помощью *перетаскивания мышью* в программе Проводник (рис. К.7). Для этого значок файла нужно выделить на правой панели (см. *отметка файлов и каталогов*) и подвести указатель мыши к выделенным файлам, нажать левую кнопку мыши и, не отпуская ее, перетащить выделенный файл или группу файлов в нужный каталог на левой панели. Если рядом с указателем мыши имеется значок +, то произойдет копирование, в противном случае — перенос. Изменить режим по умолчанию (т. е. копирование на перенос и перенос на копирование можно, нажав и удерживая в процессе перетаскивания клавишу <Ctrl>). См. *копирование каталога*

КОРЗИНА [basket, recycle bin]. *Область памяти*, предназначенная для хранения ненужных данных. В *операционной системе Windows* К. оформлена в виде каталога (папки), обычно представленного на главной панели системы значком с изображением корзины для бумаг. В К. хранятся удаленные каталоги и файлы. Процесс удаления состоит в *перетаскивании мышью* удаляемого объекта и "сбрасывании" его в К. После этой процедуры удаленные данные продолжают занимать место в физической дисковой памяти, однако их имена не отображаются в обычном списке файлов и каталогов. Поэтому их прямое использование становится невозможным. Пользователь при необходимости может снова сделать находящиеся в К. объекты доступными для использования либо окончательно стереть, освободив дисковое пространство. См. *удаление, восстановление файла, стирание*

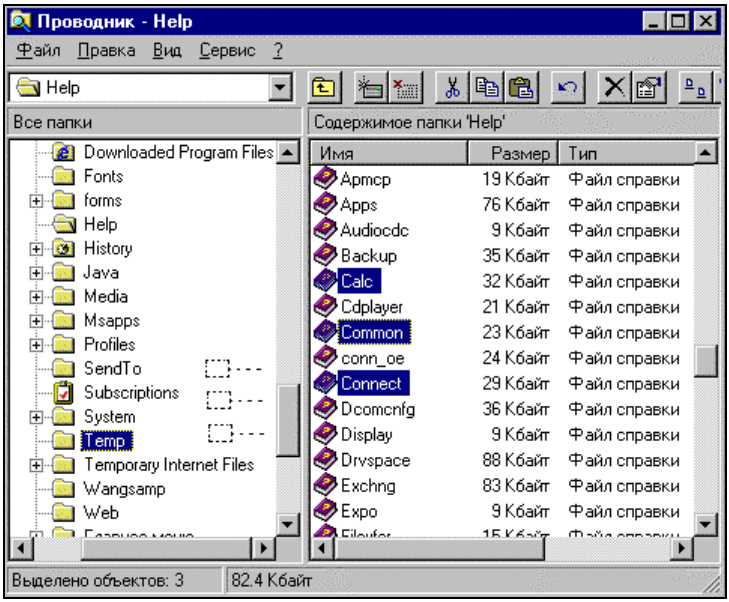

Рис. К.7. Копирование файлов с помощью программы Проводник

КОРНЕВОЙ КАТАЛОГ, главный каталог [root directory, master directory]. *Каталог*, в котором зарегистрированы все файлы и подкаталоги 1-го уровня иерархии в *файловой системе*. Считается, что имя К. к. совпадает с именем *запоминающего устройства* (диска), на котором он расположен. К. к. обычно занимает специальное место на "своем" диске

КОРНЕВОЙ СЕГМЕНТ [root segment]. Основная часть *оверлейной программы*, постоянно находящаяся в *оперативной памяти* в течение всего времени выполнения программы. См. *оверлейная структура*

КОРРЕКТОР [spelling checker]. То же, что *блок орфографического контроля*

КОРТЕЖ [tuple]. *Запись* в *реляционной базе данных*. Все К. одной таблицы реляционной базы данных имеют одну и ту же структуру полей

КОСВЕННАЯ АДРЕСАЦИЯ [indirect addressing]. Метод *адресации*, при котором адресная часть команды указывает на ячейку памяти, содержащую адрес операнда. Ср. *прямая адресация*

КРАКЕР [cracker]. 1. Название *хакера*, принятое в Интернете. В отличие от слова "хакер", которое здесь не носит негативного оттенка, К. называют "нехорошего человека", взламывающего защиту с целью *несанкционированного доступа*. 2. Хакерские программы, ломающие защиту от несанкционированного доступа

КРИВАЯ БЕЗЬЕ [Bezier curve]. Один из типов кривых, применяемых в *векторной графике*. К. Б. математически описывается интерполирующим алгебраическим многочленом 3-й степени, зависящим от заданных опорных точек. Это позволяет с помощью небольшого числа точек определить множество разнообразных гладких кривых. Выбирая положение этих точек на экране с помощью мыши или другого *указательного устройства*, можно придать К. Б. нужную форму. Поэтому К. Б. широко используются в системах *автоматизированного проектирования* и *программах рисования* для плавного сопряжения гладких кривых произвольной формы (рис. К.8)

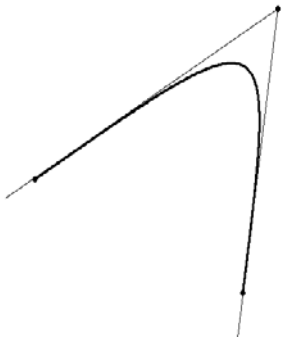

Рис. К.8. Кривая Безье

КУРСИВ [italic]. *Начертание шрифта*, при котором символы изображаются наклоненными вправо и скругленными. Например, в данном словаре К. используется для выделения ссылок на другие словарные статьи

КУРСОР [cursor]. 1. Световая отметка, обозначающая текущую позицию на экране, в которую будет произведен ввод символа при нажатии символьно-цифровой или буквенной клавиши. Обычно К. представляет собой яркий мигающий знак подчеркивания, вертикальную черту, прямоугольник или рамку. Передвижение К. по экрану осуществляется с помощью *клавиш управления курсором*. После ввода символа К. автоматически перемещается в следующую позицию на экране. 2. То же, что *указатель мыши*. 3. То же, что *цветовой маркер*. 4. В некоторых *системах управления базами данных* К. называют множество *записей*, полученных в результате выполнения *запроса*, к которым возможен *последовательный доступ* с помощью операций "перейти к следующей", "перейти к предыдущей". При этом СУБД "помнит" текущую запись в каждом из таких множеств записей, чем и объясняется использование термина К. в данном случае

КУЧА [heap]. *Область оперативной памяти*, отводимая программе для хранения данных, объем которых заранее не известен. Память из К. распределяется динамически в ходе выполнения программы. По запросу программы

ей из К. выделяется *блок памяти* нужной величины, в котором временно размещаются данные. После их использования память освобождается. В процессе работы программы может оказаться, что в К. свободного участка нужной длины нет, а есть несколько мелких участков между блоками. В этом случае производится дефрагментация К., при которой блоки располагаются встык друг к другу, освобождая память нужной длины. В языках Паскаль и Си предусмотрены специальные операторы запроса и освобождения памяти из К. Ср. *стек*. См. *динамическое распределение памяти, чистка памяти*

КЭШ [cache]. То же, что *кэш-память*

КЭШИРОВАНИЕ [caching]. Работа с использованием кэш-памяти. См. *кэширование диска*

КЭШИРОВАНИЕ ДИСКА [disk caching]. Хранение в *оперативной памяти* наиболее часто используемых *секторов диска* с целью увеличения скорости обмена данными между диском и оперативной памятью. В оперативной памяти *операционной системой* выделяется специальная область, в которой организуется *буферная память* — кэш. В нее помещаются данные, считанные с диска, и сохраняются там до тех пор, пока не будут вытеснены другими данными. Когда данные потребуются повторно, они могут быть считаны не с диска, а из кэша, а значит, быстрее. К. д. применяется не только при чтении, но и при записи. В этом случае данные могут сначала обновляться в кэше, а затем в "удобное" время копироваться на диск. Таким образом, К. д. не только увеличивает *производительность компьютера*, но и продлевает срок службы накопителей. Ср. *кэш-память*

КЭШИРОВАНИЕ ПАРОЛЕЙ [password caching]. Средство автоматического ввода паролей для доступа к различным объектам. При этом пользователю нужно помнить только один пароль — для регистрации в сети или в *операционной системе Windows*. Все остальные пароли пользователя хранятся в зашифрованном виде и становятся доступны только после регистрации в системе

КЭШИРОВАНИЕ С ОБРАТНОЙ ЗАПИСЬЮ [write-back caching]. Режим работы *кэш-памяти*, при котором в нее заносятся как данные операций чтения из *внешней памяти* (например, с диска), так и данные операций записи на диск. При этом запись во внешнюю память может быть отложена до выполнения определенных условий. Ср. *кэширование с прямой записью*

КЭШИРОВАНИЕ С ПРЯМОЙ ЗАПИСЬЮ [write-through caching]. Режим работы *кэш-памяти*, при котором в нее заносятся только данные операций чтения из *внешней памяти* (например, с диска), а данные операций записи сбрасываются на диск немедленно. При этом процессор должен ждать завершения записи. Ср. *кэширование с обратной записью*

КЭШ-ПАМЯТЬ, **кэш [cache memory]**. "Сверхоперативная" *буферная память*, предназначенная для промежуточного хранения наиболее часто используемых процессором данных. К.-п. служит для частичной компенсации разницы в скорости процессора и *основной памяти* и создается на *запоминающих устройствах* (обычно на регистрах) со скоростью чтения данных выше, чем у *оперативной памяти*. К.-п. работает следующим образом. Запрашивая данные, процессор сначала просматривает К.-п. и, если там эти данные есть, считывает их оттуда. Если их там не оказалось, то необходимые данные вызываются из *оперативной памяти* и параллельно копируются на свободное место в К.-п., где и хранятся до тех пор, пока не потребуется освободить место для записи новой порции данных. При этом в память, как правило, вытесняются наименее используемые данные. В итоге увеличивается *производительность процессора* за счет уменьшения общего *времени выборки* часто требуемых данных

Л

ЛАЗЕРНЫЙ ДИСК, **видеодиск**, **оптический диск**, **компакт-диск**, **[videodisk**, **optical disk**, **CD-ROM]**. *Носитель данных* в виде диска, считывание с которого производится посредством лазерного луча. Информация в *двоичном представлении* наносится на металлическую или полимерную поверхность Л. д. лучом мощного лазера в виде последовательности микроуглублений (питов, pits) и промежутков между ними. Запись производится вдоль спиральных дорожек, идущих от центра к периферии. На одном диске могут сосуществовать дорожки различных форматов, предназначенные для записи разных форм информации: текста, звука, изображения. В *персональных компьютерах* применяются серийные компактные акустические диски (CD), позволяющие хранить приблизительно 650 Мбайт информации. Такая емкость определена стандартом, однако при более плотном расположении дорожек или самих питов на диске может быть получен больший объем данных. Л. д. могут быть однократно записываемыми — CD-R (CD-Recordable — записываемый CD) и многократно записываемый — CD-RW (CD-Read/Write — CD с чтением/записью). Терминами CD-R и CD-RW обозначаются как устройства для записи, так и сами диски. Л. д. широко используются для долговременного хранения данных и как память для переноса данных с одного компьютера на другой, например, как носитель дистрибутивов для установки *программного обеспечения*, видеоигр и т. п.

ЛАЗЕРНЫЙ ПРИНТЕР [laser printer]. *Принтер*, в котором красящее вещество — тонер наносится на бумагу с помощью лазерного луча, а затем приплавляется за счет нагрева. Обладает высоким качеством печати, может быть цветным и черно-белым. *Разрешающая способность* современных Л. п. выше, чем у принтеров других типов, и достигает 1200 dpi и более

ЛАНДШАФТНАЯ ОРИЕНТАЦИЯ СТРАНИЦЫ [landscape page orientation]. То же, что *альбомная ориентация страницы*

ЛАССО [lasso]. *Инструмент машинной графики*, предназначенный для выделения на рисунке области произвольной формы. После вызова инструмента Л. с помощью мыши можно обрисовать нужную область рисунка. На рис. Л.1 для *графического редактора* Adobe Photoshop показаны кнопка вызова инструмента Л. и *диалоговое окно*, позволяющее установить радиусы скругления углов контура выделенной области

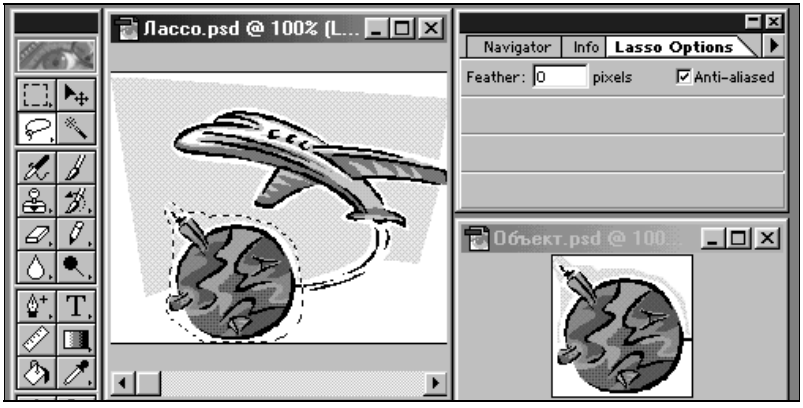

Рис. Л.1. Кнопка вызова, выделенный объект и диалоговое окно инструмента **Лассо** графического редактора Adobe Photoshop

ЛАСТИК [eraser]. *Инструмент машинной графики*, предназначенный для удаления точек рисунка с заменой их на точки с фоновым цветом или на прозрачные точки. На рис. Л.2 для *графического редактора* Adobe Photoshop показаны кнопка вызова Л. и диалоговое окно, позволяющее выбрать форму Л. (кисть, аэрограф, карандаш, блок) и установить его параметры, например, длину "мазка"

Рис. Л.2. Нажатая кнопка вызова и диалоговое окно инструмента **Ластик** графического редактора Adobe Photoshop

ЛВС [local area network (LAN)]. То же, что *локальная вычислительная сеть*

ЛЕГЕНДА [legend]. Дополнительное пояснение к графическому материалу. Например, область на диаграмме, в которой помещаются сведения, нужные для идентификации данных на диаграмме. Обычно для каждого ряда данных, отображаемых на диаграмме, в Л. имеется ключ, который показывает цвет и узор, использованный для отображения конкретного ряда данных, и текстовое описание этого ряда данных

ЛЕКСИКОН ПРОГРАММИРОВАНИЯ [programming lexicon]. Набор присущих программированию профессиональных терминов, которыми пользуются программисты в процессе своей работы. Содержание этого словаря частично отражает Л. п. Например, в Л. п. входят термины "программа", "файл", "цикл" и т. п., но кроме них в Л. п. есть *ключевые слова языков программирования* и многое другое

ЛЕКСИЧЕСКИЙ АНАЛИЗАТОР, **сканер [lexical analyzer**, **scanner]**. Составная часть *транслятора*, выполняющая первый этап трансляции *исходной программы*, в ходе которого распознаются и заменяются кодами так называемые лексемы, или токены. Например, для распространенных языков программирования лексемами являются литералы, константы, *ключевые слова* языка, идентификаторы, знаки операций. При этом Л. а. как бы сканирует текст программы, просматривая его посимвольно, чем и объясняется использование термина "сканер"

ЛИНЕЙКА [ruler]. Инструмент *графического интерфейса пользователя* в виде полосы с засечками. Может применяться для управления отступами, табуляцией, шириной страницы, положением столбцов в таблицах и пр. На рис. Л.3 показано использование Л. для установления красной строки в приложении Microsoft Word

ЛИНЕЙКА ПРОКРУТКИ, **полоса прокрутки [scroll bar]**. *Элемент управления*, служащий для управления *прокруткой изображения*. Обычно это расположенная на границе окна полоса с бегунком, сдвигая который при помощи мыши можно перемещать изображение или текст в окне по вертикали или по горизонтали. Например, если длинный текстовый документ не помещается сразу целиком на экране, то с помощью Л. п. можно быстро переместиться к другой части документа. На рис. Л.4 приведен пример использования Л. п. в программе MS Word. См. *скроллинг*

ЛИНЕЙНЫЙ АЛГОРИТМ [serial algorithm]. *Алгоритм*, не содержащий ветвей (см. *ветвь алгоритма*) и циклов. Все операции такого алгоритма выполняются последовательно, одна за другой. Блок-схема Л. а. представима в виде одного или нескольких следующих друг за другом символов "Процесс" и не содержит ни одного *символа* "Решение" (см. *элементы блок-схем*). Примером Л. а. может служить вычисление среднего арифметического трех чисел. Схема программы, реализующей этот алгоритм, приведена на рис. Л.5

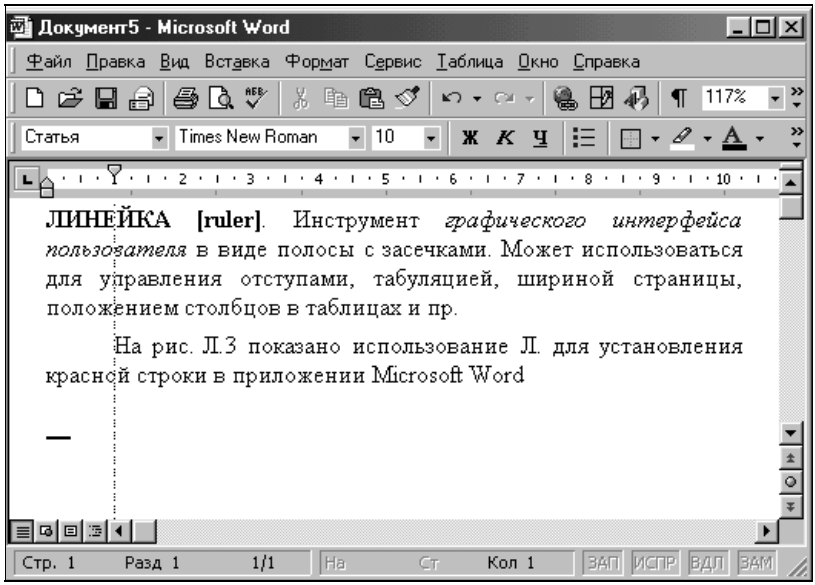

Рис. Л.3. Установка красной строки в документе Microsoft Word с помощью линейки

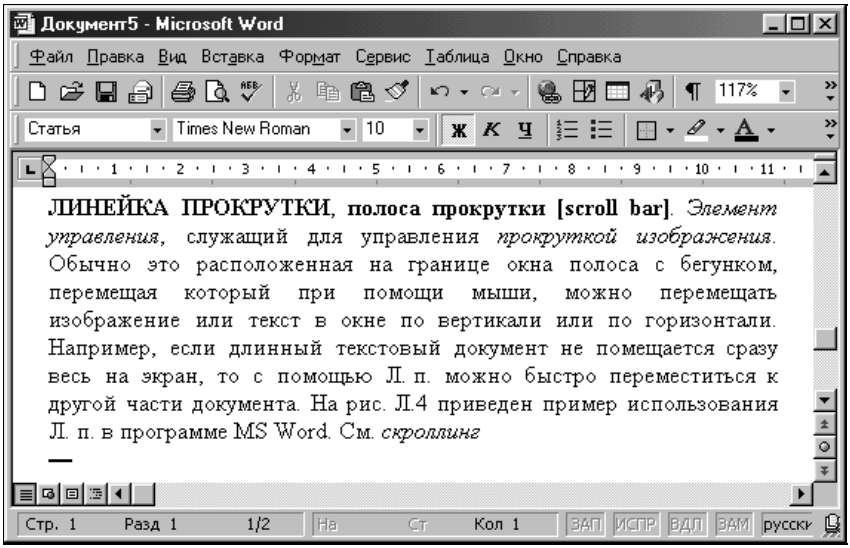

Рис. Л.4. Окно документа Microsoft Word с вертикальной и горизонтальной линейками прокрутки

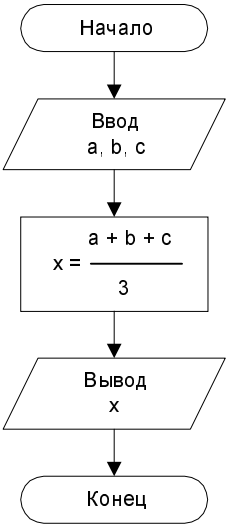

Рис. Л.5. Блок-схема линейного алгоритма

ЛИНЕЙНЫЙ ПРИНТЕР, **устройство построчной печати**, **алфавитно-цифровое печатающее устройство (АЦПУ) [line printer (LPT)]**. Быстродействующий *принтер*, печатающий сразу целую строку текста

ЛИНИИ ЗАПРОСА ПРЕРЫВАНИЙ, **линии прерываний**, **линии IRQ [interrupt request lines (IRQ)]**. Набор физических *линий связи*, используемых при взаимодействии центрального процессора с устройствами. По этим линиям *устройства ввода/вывода*, такие как клавиатура, принтер или дисковод, могут передавать прерывания *центральному процессору*. Если устройству необходимо получить или передать данные или выполнить какую-либо другую операцию, то оно активизирует соответствующую Л. з. п. Тем самым процессор информируется о необходимости обслужить устройство. В современных компьютерах есть 16 линий IRQ, но для работы с устройствами могут применяться только 15, поскольку одна линия используется особым образом

ЛИНИИ ПРЕРЫВАНИЙ [interrupt request lines (IRQ)]. То же, что *линии запроса прерываний*

ЛИНИЯ ЖИЗНИ [lifeline]. Пунктирная линия на *диаграмме последовательности унифицированного языка моделирования* UML, изображающая существование объекта на протяжении некоторого периода времени. Линия жизни параллельна оси времени

ЛИНИЯ СВЯЗИ [link, **circuit]**. Совокупность проводящей сигналы среды, преобразователей, усилителей и т. п. аппаратуры, которая образует и обеспечивает путь прохождения сигналов между двумя точками (например, меж-

ду двумя компьютерами, между компьютером и терминалом). Л. с. состоит из *каналов связи*, обеспечивающих передачу данных (речи, видео и т. д.) и управляющих сигналов. Обычно термин "Л. с." подразумевает возможность передачи сигналов в оба направления. В простейшем случае Л. с. может состоять только из двух проводов, а Л. с. компьютера с удаленным терминалом может быть создана, например, с использованием телефонных каналов общего назначения и системы спутниковой связи

ЛИНИИ IRQ [interrupt request lines (IRQ)]. То же, что *линии запроса прерываний*

ЛИСП [Lisp]. *Язык программирования*, предназначенный для обработки символьной информации. Название языка произошло от английских слов LISt Processing (обработка списков). Л. основан на понятии списка как упорядоченного множества данных, являющихся либо элементарными неделимыми данными, либо в свою очередь списками. Над списками выполняются элементарные функции, реализующие операции объединения, деления, группировки и пр. Программа на языке Л. представляет собой *рекурсивную функцию*, которая строится из элементарных функций. Л. обладает средствами работы в *диалоговом режиме*, сочетающем интерпретацию и трансляцию. Первая версия Л. разработана Дж. Маккарти в 1960 г. В настоящее время Л. применяется для программирования научно-исследовательских задач в области *искусственного интеллекта*

ЛИСТИНГ [listing]. 1. Печатный документ, формируемый *вычислительной системой* и содержащий данные о ходе и результатах разных этапов исполнения задачи. Например, Л., формируемый на этапе трансляции, обычно содержит текст *исходной программы* и результаты трансляции (сведения об ошибках или характеристики полученного *объектного модуля*). 2. То же, что *распечатка*

ЛИТЕРАЛ [literal]. Содержащийся в тексте программы *элемент данных*, представленный в виде имеющих самостоятельное значение символа или последовательности символов. Л. являются непосредственно указанные в программе адреса, числовые или *символьные константы*, но не имена констант, переменных или других объектов программы. Например, в написанных на языке Паскаль операторах `PI:= 3.14159; y:= 2*PI*x;` Л. является число `3.14159`, а символы `PI` ни в первом, ни во втором операторах не являются самостоятельно значимыми и, следовательно, не образуют Л. Это просто имя константы. Аналогично, в написанных на языке Си директиве `#define MSG "Пожалуйста, введите целое число."` и операторе `printf("%s\n", MSG)` Л. являются строка формата `"%s\n"` и символьная константа `"Пожалуйста, введите целое число."`

ЛИТЕРНЫЙ ТИП [character type]. То же, что *символьный тип*

ЛОВУШКА [hook, trap]. Прием программирования, используемый при *отладке программ* и для повышения их надежности. Если программисту известно, что при правильной работе программы должны выполняться некоторые условия, то невыполнение этих условий свидетельствует о некорректной работе программы. Например, если какая-то переменная используется для хранения величины площади геометрической фигуры, то отрицательное значение этой переменной свидетельствует о наличии ошибки, поскольку площадь не может быть отрицательной. В таком случае программист может поставить Л., т. е. определить специальную подпрограмму или другой фрагмент программы, которые должны выполняться в случае нарушения условий, характеризующих верное выполнение программы. Л. может содержать, например, выдачу пользователю диагностического сообщения о возникшей ситуации в работе программы или же более сложные действия по исправлению ошибки. Л. хороша тем, что позволяет "перехватить" ошибку, т. е. предпринять какие-то действия до того, как ошибка приведет к более тяжелым последствиям, например, к *аварийному завершению* программы или к выдаче пользователю неправильного результата. Для программирования Л. служат различные средства, зависящие от используемого *языка программирования* и *системы программирования*. Это может быть просто *условный оператор*, процедура реакции на *исключительную ситуацию*, определенное программистом *программное прерывание* и др.

ЛОГИЧЕСКАЯ ЗАПИСЬ [logical record]. *Запись*, определенная в программе одним *оператором ввода/вывода* и представляющая собой набор логически связанных данных независимо от их физического размещения. Является единицей *обмена данными* между программой и буфером ввода/вывода. Несколько Л. з. могут размещаться в одной *физической записи — блоке данных*. Противоп. *физическая запись*

ЛОГИЧЕСКАЯ ОШИБКА, ошибка алгоритма [logic error, algorithm error]. Ошибка, допущенная на этапе разработки алгоритма или вследствие его неправильного понимания. Л. о. не всегда приводит к остановке программы, и только часть результатов может оказаться неверной. Поэтому ее очень трудно найти. Как правило, она обнаруживается в процессе испытаний программы, а иногда — уже при эксплуатации

ЛОГИЧЕСКАЯ ПЕРЕМЕННАЯ [logical variable]. *Переменная*, принимающая только *логические значения*: "ИСТИНА" или "ЛОЖЬ", которые в компьютере могут быть представлены в виде 1 и 0. В *языках программирования* эти значения обычно обозначаются `1`, `true`, `T` или `0`, `false`, `F` соответственно, Л. п. вводится в программу с помощью *описания переменной*, в котором указываются идентификатор (имя) переменной и *ключевое слово*, определяющее *логический тип*. В качестве ключевых слов в описаниях Л. п. применяются: `logical` — в Фортране и `Boolean` — в Паскале. См. *описание переменной*

ЛОГИЧЕСКАЯ ФУНКЦИЯ, **булева функция [Boolean function]**. 1. *Функция*, принимающая *логические значения* "ИСТИНА" или "ЛОЖЬ" (1 или 0). 2. Функция в *булевой алгебре*

ЛОГИЧЕСКИЕ ОПЕРАЦИИ, **булевы операции [logical operations**, **Boolean operations]**. *Операции* над *логическими значениями*, выполняемые в соответствии с правилами *булевой алгебры*. Каждая *логическая переменная* принимает только одно из двух логических значений "ИСТИНА" или "ЛОЖЬ". В компьютере эти значения могут быть представлены соответственно в виде 1 или 0. Пусть А и В — логические переменные. Рассмотрим четыре основные Л. о., выполняемые компьютером: 1) Логическое "НЕ" (отрицание) — *одноместная операция*. "НЕ" А истинно, если А ложно и "НЕ" А ложно, если А истинно. 2) Логическое "И" (конъюнкция, логическое произведение) — *двуместная операция*. А "И" В истинно, если А и В оба истинны, и ложно во всех других случаях. 3) Логическое "ИЛИ" (дизъюнкция, логическое сложение) — двуместная Л. о. А "ИЛИ" В истинно, если либо А, либо В, либо и А и В — истинны. А "ИЛИ" В ложно лишь в том случае, когда А и В оба ложны. 4) Эквивалентность (тождество) — двуместная Л. о. А "ЭКВИВАЛЕНТНО" В истинно, если А и В оба истинны или оба ложны. В противном случае А "ЭКВИВАЛЕНТНО" В — ложно. Приведенные определения Л. о. сведены в табл. Л.1 (в скобках показаны обозначения операций, принятые в *булевой алгебре*)

Таблица Л.1. *Определение логических операций*

А	В	"НЕ" А ($\neg$А)	А "И" В (А $\wedge$ В)	А "ИЛИ" В (А $\vee$ В)	А "ЭКВИВАЛЕНТНО" В (А $\equiv$ В)
0	0	1	0	0	1
0	1	1	0	1	0
1	0	0	0	1	0
1	1	0	1	1	1

ЛОГИЧЕСКИЙ АДРЕС, **виртуальный адрес [logical address**, **virtual address]**. Символический или условный *адрес* ячейки или *области памяти*, устройства или *узла сети*, указанный в программе. Л. а. не зависит от конкретной *конфигурации компьютера* и мест в памяти, которые займут данные и программа в процессе ее выполнения. При *загрузке программы* в *оперативную память,* при *редактировании связей*, а также при переадресации Л. а. преобразуется соответствующим *программным обеспечением* и *аппаратными средствами* в *физический адрес*, закрепленный в *вычислительной системе* за реально существующим *запоминающим устройством* или его частью. Условность Л. а. и

механизм управления памятью *вычислительной системы* позволяют сделать программу относительно независимой от размера реальной оперативной памяти, имеющейся у компьютера. Противоп. *физический адрес*. См. *виртуальная память, относительный адрес*

ЛОГИЧЕСКИЙ ДИСК [logical disk]. Особым образом организованная область памяти на *жестком магнитном диске*, внешне проявляющая себя как самостоятельный жесткий диск. Один физический диск может быть разбит на несколько логических. Л. д. могут быть использованы для разделения жесткого диска между несколькими пользователями. Л. д. в *персональных компьютерах операционная система* присваивает имена C:, D:, E: и т. д. в алфавитном порядке. Л. д. может быть сжатым. Содержимое диска можно просмотреть, щелкнув по его значку в *Проводнике Windows* или в окне *Мой компьютер*. См. *сжатие данных*

ЛОГИЧЕСКИЙ ТИП [logical type]. *Тип данных*, принимающих только *логические значения* "ИСТИНА" или "ЛОЖЬ", над которыми выполняются *логические операции*. В *языках программирования* логические значения обычно обозначаются `1`, `true`, `T` или `0`, `false`, `F`. В компьютере эти значения могут быть представлены соответственно в виде 1 или 0. В некоторых языках Л. т. является *встроенным типом*, например, в языках Фортран и Паскаль. При этом в качестве ключевых слов в *описании переменных* и других *объектов программы*, относящихся к Л. т., применяются `logical` — в Фортране и `Boolean` — в Паскале. В языке Си можно создать Л. т. данных с помощью *описания типа*

ЛОГИЧЕСКОЕ ВЫРАЖЕНИЕ, **булево выражение [logical expression**, **Boolean expression]**. Совокупность одной или нескольких *логических переменных*, *логических функций* и отношений, соединенных *знаками логических операций* и скобками. Результатом Л. в. являются *логические значения* "ИСТИНА" или "ЛОЖЬ" ("да" или "нет"). Например, если координаты точки A (x, y) заданы так, что точка A лежит в области G (рис. Л.6), то Л. в. $x^2 + y^2 \leq$ $\leq 1 \wedge y \geq 0$ имеет значение "ИСТИНА".

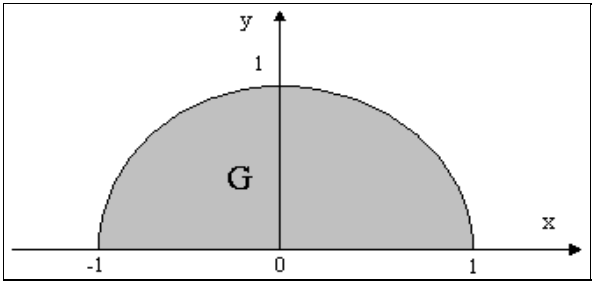

Рис. Л.6. Геометрия области, определяемой логическими выражениями $x^2 + y^2 \leq 1 \wedge y \geq 0$

Если последовательность выполнения операций в Л. в. не определяется скобками, то устанавливается следующий порядок: 1) вычисление логических функций; 2) вычисление отношений; 3) операции отрицания; 4) операции логическое "И"; 5) операции логическое "ИЛИ"; 6) операции "ЭКВИВАЛЕНТНОСТЬ". Однотипные операции выполняются последовательно слева направо

ЛОГИЧЕСКОЕ ЗНАЧЕНИЕ, **булево значение [logical value**, **Boolean value]**. Значение "ИСТИНА" или "ЛОЖЬ", которые могут принимать *логические выражения*. В компьютере эти значения могут быть представлены в виде 1 и 0. В *языках программирования* данные значения обычно обозначаются `1`, `true`, `T` или `0`, `false`, `F` соответственно

ЛОГИЧЕСКОЕ "И" [logical "AND"]. То же, что *логическое умножение*

ЛОГИЧЕСКОЕ "ИЛИ" [logical "OR"]. То же, что *логическое сложение*

ЛОГИЧЕСКОЕ ИМЯ [logical name]. *Имя* устройства *вычислительной системы*, указанное в программе. Например, Л. и. *внешнего запоминающего устройства*, используемое в программе при обращении к *внешней памяти*. В процессе выполнения программы одному Л. и. могут соответствовать несколько реальных физических устройств или, наоборот, нескольким Л. и. может соответствовать одно физическое устройство. Связь Л. и. с конкретным устройством устанавливается *оператором ЭВМ* или программным способом. Этим достигается независимость программы от конкретного устройства, т. е. от конкретной *конфигурации компьютера*. Например, *операционная система Windows* резервирует Л. и. С: для *жесткого диска* (если в компьютере несколько жестких дисков или на жестком диске сформировано несколько *логических дисков*, то для них используются Л. и. D:, Е: и т. д. по алфавиту)

ЛОГИЧЕСКОЕ СЛОЖЕНИЕ, **логическое "ИЛИ"**, **дизъюнкция [logical "OR"**, **disjunction]**. Двуместная *логическая операция*, результатом которой является "ЛОЖЬ", если оба операнда принимают значение "ЛОЖЬ", и "ИСТИНА" — в остальных случаях. В *языках программирования* обычно обозначается `OR` и служит для записи *логических выражений*. См. *логические операции*

ЛОГИЧЕСКОЕ УМНОЖЕНИЕ, **логическое "И"**, **конъюнкция [logical multiplication**, **logical "AND"**, **conjunction]**. Двуместная *логическая операция*, результатом которой является "ИСТИНА", если оба операнда принимают значение "ИСТИНА", и "ЛОЖЬ" — в остальных случаях. В *языках программирования* обозначается `AND`, `&` и др. Используется для записи *логических выражений*. См. *логические операции*

ЛОГИЧЕСКОЕ УСТРОЙСТВО [logical device]. 1. Системная программа, обеспечивающая взаимодействие приложения с конкретным физическим устройством. Например, программа (драйвер), выполняющая операции ввода/вывода по запросам приложения. 2. Функциональный эквивалент уст-

ройства, введенный программистом и рассматриваемый как реальная вещь. Например, *логический диск*

ЛОКАЛИЗАЦИЯ ОШИБКИ [error localization]. Определение относительно небольшого участка программы, в котором должна находиться ошибка. Например, определение оператора, содержащего Л. о., является составной частью *отладки программы*. См. *ошибка в программе*

ЛОКАЛЬНАЯ ВЫЧИСЛИТЕЛЬНАЯ СЕТЬ (ЛВС), **локальная компьютерная сеть**, **локальная сеть [local area network (LAN)**, **local network]**. *Компьютерная сеть*, состоящая из нескольких компьютеров, связанных между собой в пределах одного помещения, здания или ограниченной территории с целью *обмена данными* и совместного использования дорогостоящих устройств (принтеров, сканеров, плоттеров и т. п.). В состав ЛВС входят также программы и устройства, обеспечивающие эту связь. Обычно ЛВС состоит из одной более мощной *главной ЭВМ* и нескольких менее производительных компьютеров с установленными на них *сетевыми платами*, кабеля, подсоединенного к этим платам, и *программного обеспечения*. Общая схема физических соединений компьютеров в ЛВС обычно называется *топологией сети*. По типу топологии различаются *древовидные, кольцевые, радиальные* и *шинные сети*. Простейшим примером применения ЛВС является компьютерный учебный класс, в котором компьютеры учеников соединены между собой и с компьютером преподавателя, управляющего учебным процессом. Противоп. *региональная вычислительная сеть*

ЛОКАЛЬНАЯ ГРУППА [local group]. *Группа* на компьютере с *операционной системой Windows*, которой могут быть предоставлены разрешения и права

ЛОКАЛЬНАЯ КОМПЬЮТЕРНАЯ СЕТЬ [local computer network]. То же, что *локальная вычислительная сеть*

ЛОКАЛЬНАЯ ПЕРЕМЕННАЯ [local variable]. *Переменная* с ограниченной *областью определения* в программе. Л. п. существует только в том модуле или блоке, где она описана, а вне его является недоступной. О переменной, областью существования которой является блок, говорят, что она "локализована в блоке". Ср. *глобальная переменная*

ЛОКАЛЬНАЯ СЕТЬ [local network]. То же, что *локальная вычислительная сеть*

ЛОКАЛЬНАЯ ШИНА [local bus]. Специальная дополнительная *шина* с высокой пропускной способностью, связывающая *центральный процессор* с некоторыми компонентами компьютера. Например, в *персональных компьютерах* через Л. ш. возможно подключение видеоадаптера, *контроллера жесткого магнитного диска* и некоторых других устройств; в современных компьютерах распространены Л. ш. двух видов: VLB и PCI

ЛОКАЛЬНЫЙ ВЫЗОВ ПРОЦЕДУРЫ [local procedure call (LPC)]. Средство оптимизированной передачи сообщений, обеспечивающее связь между двумя процессами, которые выполняются на одном и том же компьютере. Защищенные подсистемы используют Л. в. п. для общения друг с другом и с клиентскими процессами. Фактически Л. в. п. — это разновидность *удаленного вызова процедуры*, оптимизированная для локального использования

ЛОКАЛЬНЫЙ КОМПЬЮТЕР [local computer]. *Компьютер*, на котором пользователь выполнил *вход в систему,* и доступный пользователю непосредственно, т. е. без коммуникационных линий и устройств, таких как сетевая плата или модем

ЛОКАЛЬНЫЙ ПРОФИЛЬ ПОЛЬЗОВАТЕЛЯ [user local profile]. *Профиль пользователя*, который создается автоматически при первом входе пользователя на компьютер с *операционной системой Windows*

ЛЭПТОП [laptop]. То же, что *портативный компьютер*

М

МАГАЗИН [stack, last in first out (LIFO)]. То же, что *стек*

МАГИСТРАЛЬ [backplane, trunk, unibus]. 1. То же, что *шина.* 2. Прямая *линия связи* между узлами *компьютерной сети*

МАГНИТНАЯ КАРТА [magnetic card]. *Носитель данных последовательного доступа* в виде пластиковой карточки стандартного размера, поверхность которой покрыта магнитным материалом. Применяется в *специализированных компьютерах*, например, в системе банковского обслуживания. Ср. *смарт-карта*

МАГНИТНАЯ ЛЕНТА (МЛ) [magnetic tape]. *Носитель данных последовательного доступа* в виде тонкой гибкой пластмассовой ленты, покрытой магнитным материалом. Применяется в *накопителях на* МЛ для реализации *внешней памяти.* Основным преимуществом МЛ по сравнению с другими носителями данных является ее относительно малая стоимость. Главный недостаток — наибольшее *время доступа* по сравнению с другими носителями

МАГНИТНЫЙ ДИСК [magnetic disk]. *Носитель данных*, представляющий собой диск, поверхности которого покрыты магнитным материалом. Для *чтения* и *записи данных* М. д. помещается в специальное устройство — дисковод, снабженное магнитными *головками записи/чтения*, где вращается с большой скоростью. Данные записываются на поверхности диска вдоль концентрических окружностей, называемых дорожками. Для выбора нужной дорожки подвижная головка устанавливается в соответствующую позицию.

М. д. имеют разновидности: *жесткий магнитный диск* и *гибкий магнитный диск*. В целях увеличения объема памяти несколько дисков объединяются в *пакет магнитных дисков*

МАКРОАССЕМБЛЕР [macro assembler]. *Транслятор* с *языка ассемблера*, включающий в себя макропроцессор, позволяющий выполнять *макроподстановки*. См. *макрос*

МАКРОБИБЛИОТЕКА [macro definition library]. *Программный продукт*, содержащий собрание макроопределений, предназначенных для использования при *разработке программ*. Достаточно подключить М. к разрабатываемой программе, чтобы содержащиеся в ней макросы с помощью макропроцессора можно было включать в состав программы. См. *макрос*

МАКРОВИРУС [macro virus]. *Компьютерный вирус*, реализованный в виде *макроса*. Многие современные приложения, например, все приложения пакета Microsoft Office, позволяют автоматизировать работу пользователя с помощью макросов. Средства макропрограммирования развиты до такой степени, что позволяют создавать очень сложные программы, чем пользуются злонамеренные авторы компьютерных вирусов. Особую опасность М. представляет потому, что заражение происходит не в результате запуска программы (осторожные пользователи стараются не запускать неизвестных им программ), а в результате выполнения внешне невинной операции открытия обычного документа, например, документа MS Word.

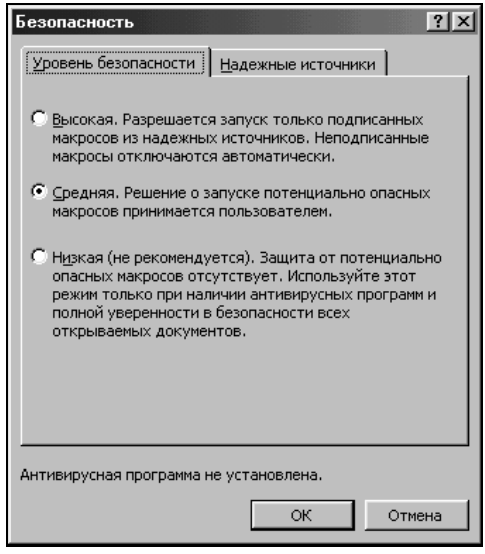

Рис. М.1. Диалоговое окно для управления режимом безопасного запуска макросов в Microsoft Office

Для борьбы с М. используются *антивирусные программы* и встроенные средства защиты. Например, на рис. М.1 показано диалоговое окно для управления режимом безопасности в приложениях Microsoft Office

МАКРОВЫЗОВ, **макрокоманда [macro call**, **macro instruction**, **macrocode]**. Один из компонентов *макроса*. Последовательность символов, заменяемая макропроцессором другой последовательностью символов. М. указывает макропроцессору: 1) место в *исходном тексте*, где должна быть сделана макроподстановка; 2) макроопределение, в котором содержится образец заменяющей последовательности символов. М., как правило, состоит из имени макроса и значений аргументов, по которым макропроцессор настраивает образец заменяющей последовательности символов перед макроподстановкой. Например, на языке Си: `PRINT (z);` — М. с именем `PRINT` и аргументом `z`. См. *макрос*

МАКРОГЕНЕРАТОР [macro generator]. То же, что *макропроцессор*

МАКРОГЕНЕРАЦИЯ [macro generation]. 1. Процесс выполнения *макроподстановок макропроцессором*. 2. То же, что *макроподстановка*. См. *макрос*

МАКРОКОМАНДА [macro, **macro command**, **macrocode**, **macro instruction**, **macro call]**. 1. В *интерактивной системе* — *команда*, вызывающая выполнение последовательности других команд. 2. В *машинно-ориентированных языках программирования* — команда или оператор, реализуемые несколькими *машинными командами*. С помощью М. можно заказать для задачи некоторые ресурсы, возбудить процесс ввода/вывода и т. п. М., как правило, наделяются высоким приоритетом. 3. То же, что *макровызов*. См. *макрос*

МАКРООПРЕДЕЛЕНИЕ [macro declaration, **macro definition]**. Определение *макроса*. Содержит данные, указывающие макропроцессору: 1) какая последовательность символов в *исходном тексте* является макровызовом; 2) какой последовательностью символов необходимо заменить этот макровызов. М. начинается с *ключевого слова*, за которым следуют образец макровызова и образец последовательности символов, заменяющей макровызов, отделенные друг от друга разделителями (обычно пробелами). Так, в написанном на языке Си М. (см. пример в статье *макрос*):

```
#define PRINT(x) printf(#x"=%d.\n",x)
```

`#define` — ключевое слово, указывающее, что это М.; `PRINT(x)` — образец макровызова; `printf(#x"=%d.\n",x)` — образец заменяющей последовательности символов; разделителями являются пробелы. Образец макровызова, как правило, состоит из имени макроса (в нашем примере — `PRINT`) и специальным образом выделенных аргументов — макропараметров (в нашем примере `x` — макропараметр, выделенный скобками). Образец замещающей последовательности также содержит макропараметры. Встретив в исходном тексте макровызов, макропроцессор выбирает из соответствующего М. обра-

зец заменяющей последовательности символов и подставляет в него указанные в макровызове значения макропараметров. Полученная в результате этого последовательность символов называется макрорасширением. Затем макропроцессор замещает макровызов макрорасширением. Если в М. макропараметры отсутствуют, то образец заменяющей последовательности символов и является макрорасширением. См. *макрос*

МАКРОПАРАМЕТР [macro parameter]. Символический аргумент в *макроопределении*, значение которому присваивает программист при написании макровызова. См. *макрос, макроопределение*

МАКРОПОДСТАНОВКА [macro generation]. Осуществляемая *макропроцессором* замена макровызова текстом, заданным в макроопределении. См. *макрос, макрогенерация*

МАКРОПРОГРАММА [macro program]. *Программа*, в которой используются *макросы*. Исходная программа для макропроцессора

МАКРОПРОГРАММИРОВАНИЕ [macro programming]. Разработка *макропрограмм*, т. е. разработка программ с использованием макросов. См. *макрос*

МАКРОПРОЦЕССОР, **макрогенератор [macro processor**, **macro generator]**. Программа, осуществляющая *макроподстановки*. Часть *программного обеспечения*, обеспечивающая функционирование макросов. Схема работы М. приведена на рис. М.2

Рис. М.2. Схема работы макропроцессора

М. имеет два входа: макроопределения и исходный текст. М. воспринимает макроопределения и затем считывает исходный текст, передавая его на выход без изменений, пока в нем не встретится макровызов. Когда это происходит, М., пользуясь соответствующим макроопределением, отыскивает в макровызове нужные значения макропараметров, и макровызов замещается в выходном тексте макрорасширением с соответствующей подстановкой их значений. Часто функции М. выполняет препроцессор, например, препроцессор языка Си. См. *макрос, препроцессор*

МАКРОРАСШИРЕНИЕ [macro expansion]. Последовательность символов, заменяющая *макровызов*, первоначально содержавшийся в тексте. М. формируется макропроцессором из заданного в макроопределении образца по-

следовательности символов. Иногда термин "макрорасширение" применяется вместо термина "макроподстановка". См. *макрос*

МАКРОС [macros]. 1. Средство замены одной последовательности символов другой. С помощью М. можно делать замены в текстах книг, писем, документов, программ и т. п. Компонентами М. являются макроопределение и макровызов. В *программном обеспечении* функционирование М. обеспечивается макропроцессором. Макропроцессору задается исходный текст, в котором есть подлежащие замене последовательности символов (макровызовы), и с помощью макроопределений указывается, чем они должны быть заменены. В результате работы макропроцессора получается текст, в котором выполнены необходимые замены, т. е. сделаны макроподстановки. Например, `3.1416` подставлено вместо `PI`. Большинство макропроцессоров позволяют перед заменой настроить заданный в макроопределении образец текста по указанным в макровызове значениям аргументов. В результате вместо макровызова `ПОЕЗД(16:30)` в текст можно подставить "поезд, отправляющийся в 16 часов 30 минут", а вместо `ПОЕЗД(00:15)` — "поезд, отправляющийся в 00 часов 15 минут". М. широко применяются в *языках программирования* с целью сокращения *исходного текста программы.* Например, при программировании на языке Си вывода числовых значений можно воспользоваться М., определив его с помощью макроопределения

```
#define PRINT(x) printf(#x"= %d.\n",x);
```

Теперь вместо оператора вывода `printf("z= %d.\n",z)` в тексте программы можно написать только макровызов `PRINT(z)`. Макропроцессор, содержащийся в препроцессоре Си, перед трансляцией автоматически заменит в тексте программы `PRINT(z)` на `printf("z= %d.\n",z)`. Такой прием позволяет сократить программу, если требуется выводить численные значения различных величин и в разных местах программы. Между М. и подпрограммой существует четкое отличие: М. — средство замены, применяемое до компиляции программы, а подпрограмма — средство замены, применяемое во время исполнения программы, поэтому М. сокращают *исходную программу*, а подпрограммы сокращают размер *исполняемого файла.* 2. Последовательность команд, запускаемая одним нажатием клавиши на клавиатуре или *кнопки на экране дисплея*

МАКРОСРЕДСТВА [macro facility]. Средства *макроязыка* и любые средства, обеспечивающие работу с *макросами.* См. *макрос*

МАКРОЯЗЫК [macro language]. 1. *Язык программирования*, предоставляющий возможность разрабатывать *макропрограммы.* 2. Подмножество языка программирования, позволяющее задавать макроопределения, макровызовы и условия макрогенерации. См. *макрос*

МАЛАЯ ЦИФРОВАЯ КЛАВИАТУРА [numeric keypad]. Группа клавиш клавиатуры, обычно расположенная справа и предназначенная для удобного

ввода цифр и *знаков арифметических операций.* Помимо цифровых клавиш и клавиш <+>, <->, <*>, </>, <.> содержит дополнительную клавишу <Enter> (Ввод), дублирующую основную клавишу <Enter> (см. *специальные клавиши*). На клавиатуре *персональных компьютеров* (см. рис. К.3) некоторые цифровые клавиши М. ц. к. дублируют *клавиши управления курсором.* Для перевода этих клавиш в режим набора цифр необходимо нажать *специальную клавишу* <Num Lock>

МАЛЫЕ ПРОПИСНЫЕ [small caps]. То же, что *капитель*

МАНИПУЛИРОВАНИЕ ДАННЫМИ [data manipulation]. То же, что *обработка данных*

МАНТИССА [fractional part, fraction]. Часть *числа с плавающей точкой*, содержащая *значащие разряды* этого *числа.* Например, М. числа $-0.1237 \cdot 10^{-5}$ есть -0.1237

МАРКЕР [mark, marker]. 1. То же, что *метка.* 2. То же, что *курсор.* См. *цветовой маркер.* 3. Служебная отметка на *носителе данных* или в передаваемых сообщениях, указывающая начало или конец всех хранимых или передаваемых данных либо их частей. Это может быть управляющая запись (особый код), специальный сигнал, читаемый *головкой записи/чтения*, либо отражающий или прозрачный участок поверхности носителя, воспринимаемый оптическим устройством, и т. п. С помощью М. указывают границы файлов, *блоков данных*, групп блоков, физическое начало дорожки, границы сегментов и т. п. 4. То же, что *буллит*

МАРКЕРНАЯ КОЛЬЦЕВАЯ СЕТЬ [token ring network]. То же, что *кольцевая сеть с маркерным доступом*

МАРКЕРНАЯ СЕТЬ [token network]. То же, что *сеть с маркерным доступом*

МАРКЕРНАЯ ШИННАЯ СЕТЬ [token bus network]. То же, что *шинная сеть с маркерным доступом*

МАРКИРОВАННЫЙ СПИСОК [bulleted list]. Список, каждый пункт которого отмечен *буллитом* (рис. М.3)

МАРШРУТ [route]. Путь *передачи данных* между *узлами* и терминалами *вычислительной сети*

МАРШРУТИЗАТОР [router]. Средство определения *маршрута передачи данных* между *узлами* и терминалами *вычислительной сети.* М. реализуется программно или программно-аппаратно

МАРШРУТИЗАЦИЯ [routing]. Определение *маршрута передачи данных* в сети. М. реализуется в каждом промежуточном *узле сети* направлением пакета или соединения, поступившего по одной из *линий связи,* на другую ли-

нию, исходящую из этого узла. См. *маршрутизатор*, *коммутация пакетов*, *коммутация соединений*

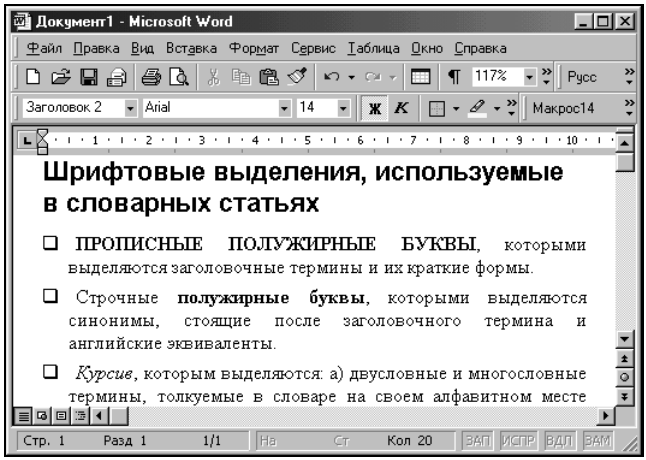

Рис. М.3. Маркированный список

МАСКА [mask]. Шаблонная комбинация битов или символов, используемая для анализа данных. Обычно с помощью М. производятся выделения, сравнения или исключения отдельных частей данных. М. участвует в операции над байтом, словом или *полем данных* и должна иметь соответствующий формат. Например, осуществляя поразрядное *логическое сложение* М. 00111000 с байтом данных 10101010, можно выделить значения его 3, 4 и 5-го битов. В результате получится 00101000. М. применяются подобным образом для анализа управляющей информации, содержащейся, например, в *слове состояния*

МАССИВ [array]. В *языках программирования* — совокупность чисел, *логических значений* или других однотипных *элементов данных*, которой присвоено имя. Каждый *элемент* М. однозначно определяется именем М. и набором целых чисел, называемых *индексами* М. Аналогами М. в алгебре могут служить векторы и матрицы, компоненты которых также обозначаются именем и индексами:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad A = \begin{pmatrix} a_{11} a_{12} a_{13} \\ a_{21} a_{22} a_{23} \\ a_{31} a_{32} a_{33} \end{pmatrix}$$

В программах эти объекты задаются М. Например, в Паскале приведенный выше вектор x может быть задан числовым массивом с тем же именем x,

элементы которого `x[1]`, `x[2]`, `x[3]` будут содержать соответствующие компоненты вектора. Над элементами М. можно выполнять различные операции. При этом в тексте программы элементы М. пишутся в виде *переменной с индексами*. Для правильной трансляции программа должна содержать *описание* М., в котором указываются имя массива, тип образующих массив элементов данных, *размерность массива* и *границы* изменения *индексов*

МАССИВЫ RAID [Redundant Array of Independent Disks (RAID)]. Система обеспечения надежности хранения данных, использующая, например, дополнительные *жесткие диски* для создания дублирующих копий информации или дополнительные биты четности для восстановления информации на случай потери данных на одном из дисков. См. *резервирование*

МАССОВАЯ ПАМЯТЬ [mass storage]. Обобщающий термин для внешней памяти компьютера (магнитных дисков, магнитных лент или оптических дисков), описывающий большие объемы данных по сравнению с емкостью традиционной памяти компьютера

МАСТЕР [Wizard]. Специальное приложение (или компонент приложения), которое упрощает использование или настройку системы. Например, Мастер установки принтеров упрощает процедуру установки принтера; Мастер сводных таблиц в MS Excel упрощает создание сводных таблиц. Обычно реализуется в виде последовательности диалоговых окон, которые содержат пояснения и простые вопросы и дают возможность пользователю шаг за шагом выполнить довольно сложную процедуру

МАСТЕР-ПРОГРАММА [master program]. Главная часть *параллельной программы*, работающей по схеме мастер/рабочие. М.-п. порождает *параллельные вычислительные процессы*, распределяет между ними обрабатываемые данные и собирает результаты обработки.

МАСШТАБИРУЕМОСТЬ [scalability]. 1. Свойство *операционной системы многопроцессорной ЭВМ*, обеспечивающей возможность запуска одних и тех же приложений как в однопроцессорном режиме, так и в многопроцессорном. 2. Свойство *программного* и *аппаратного обеспечения*, позволяющее решать данную задачу быстрее, увеличив количество *процессорных элементов* 3. Свойство *программного обеспечения*, позволяющее увеличивать количество одновременно обслуживаемых пользователей за счет привлечения дополнительных ресурсов.

МАТЕМАТИЧЕСКИЙ СОПРОЦЕССОР [floating-point processor, mathematical coprocessor, math coprocessor]. *Сопроцессор*, обеспечивающий высокоскоростную математическую обработку данных. М. с. существенно ускоряет работу компьютера при эксплуатации программ, производящих большой объем вычислений. В *персональных IBM-совместимых компьютерах* ранних моделей М. с. — микросхема, выполняющая математические действия над *числами с плавающей точкой*. В этих компьютерах применяются следующие

М. с., разработанные фирмой Intel (в порядке возрастания вычислительных возможностей): 80287; 80387SX; 80387DX; 80487SX. Сопроцессор 80287 — 16-разрядный, а все остальные — 32-разрядные, т. е. максимальные порции данных, с которыми сопроцессор может выполнять операции, составляют соответственно 16 бит (2 байта) и 32 бита (4 байта). В процессорах Pentium М. с. встроен в основной процессор и отдельно не применяется

МАТЕРИНСКАЯ ПЛАТА [mother board]. Главная *плата системного блока персональных компьютеров.* На М. п. *персональных компьютеров* расположены *центральный процессор, оперативная память, системная* и *локальная шины*, а также разъемы, в которые вставляются выполненные в виде отдельных плат *адаптер дисплея*, *контроллеры жесткого* и *гибких дисков* и порты. У некоторых персональных компьютеров адаптер, контроллеры и порты выполнены как части М. п. В настоящее время имеется тенденция производителей компьютеров бóльшую часть устройств компьютера размещать на М. п., поскольку возможности современных микросхем это позволяют, а такое техническое решение является более дешевым и более надежным

МАТРИЧНЫЙ ПРИНТЕР [matrix printer]. *Принтер*, в котором изображение формируется ударами иголок печатающей головки через красящую ленту. М. п. могут быть 9- и 24-игольчатыми. Качество печати значительно выше у последних. См. *разрешающая способность*

МАТРИЧНЫЙ ПРОЦЕССОР [array processor]. Набор связанных между собой идентичных *процессоров*, работающих одновременно под управлением общего *устройства управления.* Обычно в качестве управляющего устройства выступает *центральный процессор.* М. п. может быть рассчитан как на параллельное выполнение однотипных операций над *элементами массива* (например, матричных вычислений), так и на выполнение *параллельных вычислительных процессов.* М. п. используются для быстрого решения задач, связанных с проведением большого объема математических вычислений, или для ускорения обработки видеоинформации. М. п. являются мощным средством решения задач, обладающих *параллельным алгоритмом.* В настоящее время появились специализированные *интегральные схемы*, реализующие алгоритмы матричных вычислений аппаратно. См. *транспьютер*

МАШИНА СОСТОЯНИЙ [automaton, finite-state machine]. То же, что *конечный автомат*

МАШИНА ТЬЮРИНГА [Turing machine]. Получила свое название по имени английского математика Алана Тьюринга, предложившего в 1937 г. способ формального задания алгоритмов с помощью некоторой абстрактной машины. Суть работы М. Т. сводится к следующему. Имеется потенциально бесконечная лента, разбитая на ячейки, в каждой из которых может быть записан один символ из некоторого конечного алфавита. М. Т. имеет головку чтения/записи, которая позволяет прочитать символ в текущей ячейке,

записать символ в ячейку, а также сдвинуть головку в соседнюю ячейку (влево или вправо). Машина работает дискретно, по тактам и на каждом такте находится в одном из возможных состояний, число которых конечно. Для каждой пары (состояние, обозреваемый символ) определена тройка (записываемый символ, движение головки, новое состояние). До начала работы М. Т. находится в начальном состоянии, а головка чтения/записи обозревает на ленте самую левую непустую ячейку. Таким образом, обозревая очередной символ, М. Т. записывает новый символ (может быть, тот же самый), сдвигает головку влево, вправо или остается на месте и переходит в новое состояние (или остается в прежнем). Среди состояний выделяются заключительные, при достижении которых машина останавливается. Записанная при этом на ленте информация является результатом работы М. Т. Обычно М. Т. задают в виде таблицы, строки которой помечены названиями и номерами состояний, столбцы — входными символами, а в ячейках таблицы указаны: записываемый символ, направление движения головки (←, → или °) и номер нового состояния. Например, приведенная в табл. М.1 М. Т. вычисляет следующее натуральное число по его изображению в *позиционной двоичной системе счисления*, т. е. по последовательности двоичных цифр 0 и 1. Например, если на ленте были записаны символы 101, то после остановки машины окажутся записанными символы 110.

Таблица М.1. *Пример работы машины Тьюринга*

Состояния		**Входные символы**		
Название	**S**	**"0"**	**"1"**	**Пусто**
Начальное	1	'0', →, 1	'1', →, 1	пусто, ←, 2
Сложение	2	'1', °, 3	'0', ←, 2	'1', °, 3
Заключительное	3	'0', °, 3	'1', °, 3	пусто, °, 3

М. Т. является универсальным способом задания алгоритмов, т. е. всякий алгоритм может быть задан М. Т. Ср. *конечный автомат*

МАШИННАЯ ГРАФИКА [computer graphics]. То же, что *компьютерная графика*

МАШИННАЯ КОМАНДА [computer instruction]. *Команда*, которая может быть распознана и выполнена *центральным процессором*, т. е. команда, входящая в *систему команд* конкретной ЭВМ

МАШИННАЯ ОПЕРАЦИЯ [computer operation]. *Операция* над данными, допускаемая *системой команд* вычислительной машины. М. о. могут быть

большинство *арифметических* и *логических операций*, а также операции сравнения, выборки, пересылки данных из одной *ячейки памяти* в другую и т. п.

МАШИННАЯ ПРОГРАММА [computer program, machine program]. *Программа* на *машинном языке*. М. п. является последовательностью *машинных команд* и может быть распознана и выполнена *центральным процессором*

МАШИННОЕ ВРЕМЯ [computer time]. Время, затрачиваемое *компьютером* на выполнение определенного комплекса вычислений. Ср. *время прогона*, *время простоя* и *процессорное время*

МАШИННОЕ СЛОВО [computer word]. Определенной длины последовательность битов или символов, воспринимаемая памятью, *арифметическим устройством* или *устройством управления* компьютера как единое целое, имеющее определенное смысловое содержание. Например, М. с. являются *машинные команды*, из которых состоят *машинные программы*, а также операнды, над которыми производятся *машинные операции*. М. с., представляющее собой объект обработки, иногда называют *информационным словом*. М. с. может занимать одну *ячейку* (байт) *памяти* или несколько соседних ячеек (байтов). См. *двойное слово*

МАШИННО-ЗАВИСИМЫЙ ЯЗЫК [computer-sensitive language]. То же, что *машинно-ориентированный язык*

МАШИННО-НЕЗАВИСИМЫЙ ЯЗЫК [computer independent language]. *Язык программирования*, структура и средства которого не связаны ни с какой конкретной *архитектурой ЭВМ* и позволяют выполнять составленные на нем программы на любых компьютерах, снабженных трансляторами с этого языка. М.-н. я. являются все *языки программирования высокого уровня*. Противоп. *машинно-ориентированный язык*

МАШИННО-ОРИЕНТИРОВАННЫЙ ЯЗЫК, машинно-зависимый язык [computer-sensitive language, computer-oriented language]. *Язык программирования*, который отражает архитектуру данной ЭВМ или данного класса ЭВМ. М.-о. я. может обладать набором основных операторов типичного *языка программирования высокого уровня*, типы и структура данных которого соответствуют архитектуре определенного класса ЭВМ, и при этом иметь средства, позволяющие при программировании учитывать особенности *системы команд*, организации памяти, способа адресации и *представления данных*. Это дает возможность создавать программы, эффективность которых для данного класса компьютеров сравнима с эффективностью *машинных программ*. Примером М.-о. я. является любой *язык ассемблера*. Ср. *машинно-независимый язык*

МАШИННЫЙ ИНТЕЛЛЕКТ [machine intelligence]. То же, что *искусственный интеллект*

МАШИННЫЙ НОЛЬ [computer zero]. Представление нуля в *вычислительной системе*. М. н. обозначаются числа, абсолютная величина которых меньше наименьшей ненулевой величины из *диапазона изменения чисел* данной вычислительной системы

МАШИННЫЙ СБОЙ [machine failure, failure, malfunction]. То же, что *аппаратный сбой*

МАШИННЫЙ ЯЗЫК [computer language, machine language]. *Язык программирования*, предназначенный для представления программ и данных в форме, пригодной для непосредственного восприятия их устройствами *вычислительной машины*. Является *системой команд*, данных и инструкций, имеющих форму *цифровых кодов*, которые не требуют трансляции и непосредственно интерпретируются процессором ЭВМ. Программы, написанные на других языках программирования, сначала транслируются на М. я., а уже затем выполняются *вычислительной машиной*

МБ [megabyte (MB)]. То же, что *мегабайт*

МБАЙТ [megabyte]. То же, что *мегабайт*

МБАЙТ/С [MBps]. То же, что *мегабайт в секунду*

МБИТ/С [Mbps, Mbit/s]. То же, что *мегабит в секунду*

МГц [MHz]. То же, что *мегагерц*

МЕГАБАЙТ (Мбайт, Мб) [megabyte (MB)]. Единица количества информации; 1 Мбайт = 1024 Кбайт = 13 048 576 байт

МЕГАБАЙТ В СЕКУНДУ (Мбайт/с) [megabytes per second (MBps)]. Единица скорости передачи данных; 1 Мбайт/с = 1024 Кбайт/с = 13 048 576 байт/с

МЕГАБИТ В СЕКУНДУ (Мбит/с) [megabits per second (Mbps, Mbit/s)]. Единица скорости передачи данных; 1 Мбит/с = 1024 Кбит/с = 13 048 576 бит/с

МЕГАГЕРЦ (МГц) [megahertz (MHz)]. Единица измерения частоты, эквивалентная 1 миллиону колебаний в секунду; 1 МГц = 1000 КГц = 1 000 000 Гц

МЕГАФЛОПС, Мфлопс [MegaFLOPS, MFLOPS]. Единица измерения вычислительной *производительности компьютера*, равная миллиону *арифметических операций* с плавающей точкой в секунду. 1 Мфлопс = 10^6 флопс

МЕЖДУНАРОДНАЯ АССОЦИАЦИЯ ПРОИЗВОДИТЕЛЕЙ КАРТ ПАМЯТИ ДЛЯ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ [Personal Computer Memory Card International Association (PCMCIA)]. Международная организация, разрабатывающая стандарты на компьютерные платы, выполненные на *интегральных микросхемах*. Основана в 1989 г. См. *адаптеры PCMCIA*

МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ [International Standards Organization (ISO)]. Международная некоммерческая орга-

низация, работающая под эгидой Организации Объединенных Наций и разрабатывающая и распространяющая научные и технологические стандарты. ISO объединяет организации по стандартизации более 90 стран. Например, среди разработок ISO — средства концептуального построения компьютерных сетей. См. *модель ISO/OSI*

МЕЖДУНАРОДНЫЙ КОНСУЛЬТАТИВНЫЙ КОМИТЕТ ПО ТЕЛЕГРАФИИ И ТЕЛЕФОНИИ [Consultative Committee International for Telephone and Telegraph (CCITT)]. Организация, разрабатывающая протоколы для систем *передачи данных* — модемов, сетей и факсимильной связи — в Европе. Сейчас она переименована в ITU-T и является подразделением *Международного союза телекоммуникаций* ITU (International Telecommunications Union). Некоторые из стандартов CCITT для взаимодействия модемов через обыкновенную телефонную сеть приняты и в США

МЕЖДУНАРОДНЫЙ СОЮЗ ТЕЛЕКОММУНИКАЦИЙ [international telecommunication union (ITU)]. Международная организация, ответственная за стандартизацию программного и аппаратного обеспечения в области телекоммуникаций

МЕЖМОДУЛЬНАЯ ССЫЛКА [inter modular reference]. *Ссылка* в одном *программном модуле* на объект, определенный в другом программном модуле, т. е. использование в одном модуле имени или метки, которые определены в другом модуле и значения которых определяются в процессе компоновки. Ср. *внешняя ссылка*

МЕЖСТРОЧНЫЙ ИНТЕРВАЛ, **интерлиньяж [line spacing, leading]**. Полиграфический термин, означающий расстояние между базовыми линиями строк текста. Измеряется в пунктах (1/72 дюйма), в миллиметрах или в строках. В последнем случае подразумевается умножение обычного (одинарного) М. и. на некоторый коэффициент. Например, если для текста с размером шрифта в 10 пунктов одинарный М. и. равен 12 пунктов, то полуторный М. и. составит 18 пунктов. Современные *текстовые процессоры* позволяют настраивать М. и. в широких пределах, что используется при *верстке страниц.* На рис. М.4 показано *диалоговое окно* для настройки М. и. в приложении MS Word

МЕНЮ [menu]. Изображаемый на экране дисплея список режимов, команд или вариантов ответа, предлагаемых пользователю для выбора. Предлагаемые варианты называются *пунктами* М. Они определяют последующие действия системы. Если один из пунктов М. выделен другим цветом *(цветовым маркером)*, значит, М. активно. Для выбора того или иного пункта необходимо либо перевести на него маркер при помощи *клавиши управления курсором*, либо позиционировать на нем *указатель мыши.* Затем, нажав клавишу ввода или кнопку мыши, зафиксировать сделанный выбор. Возможны другие способы выбора, например, вводом указанных в М. номера или буквы.

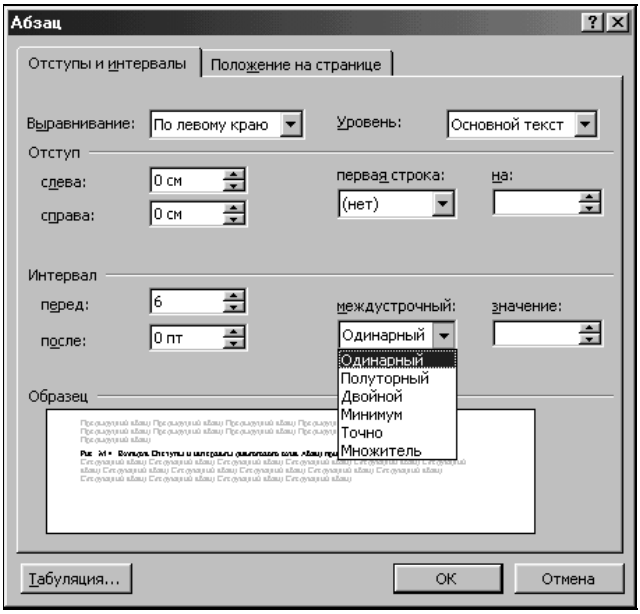

Рис. М.4. Вкладка **Отступы и интервалы** диалогового окна **Абзац** приложения Microsoft Word

Широкое распространение получили т. н. *нисходящие* М., обеспечивающие после выбора того или иного пункта появление нового М., вслед за которым может возникнуть еще одно, и т. д. Применение таких М. организует простой способ координации действий пользователя в сложных ситуациях, создавая условия для принятия им последовательности простых и правильных решений. Во время работы системы М. может постоянно находиться на экране вместе с выводимой информацией. В этом случае его пункты располагаются горизонтально *(горизонтальное М.)* в верхней или нижней части экрана и не перекрывают выведенную информацию. М., пункты которого располагаются вертикально, один под другим *(вертикальное М.)*, может перекрывать выведенную информацию. Поэтому такие М., как правило, возникают на экране в момент, когда пользователю нужно сделать выбор, и исчезают, как только этот выбор сделан. С развитием средств *графического интерфейса пользователя* в рамках *операционной системы Windows* широкое распространение получили такие формы М., как *контекстные меню*, *пиктографические меню*, *панели команд* и *панели инструментов*. Например, на рис. М.5 показаны некоторые типы меню, которые применяются в приложении MS Word. Существуют специальные *программные средства*, облегчающие программисту создание М. в разрабатываемой программе. Ср. *элементы управления*

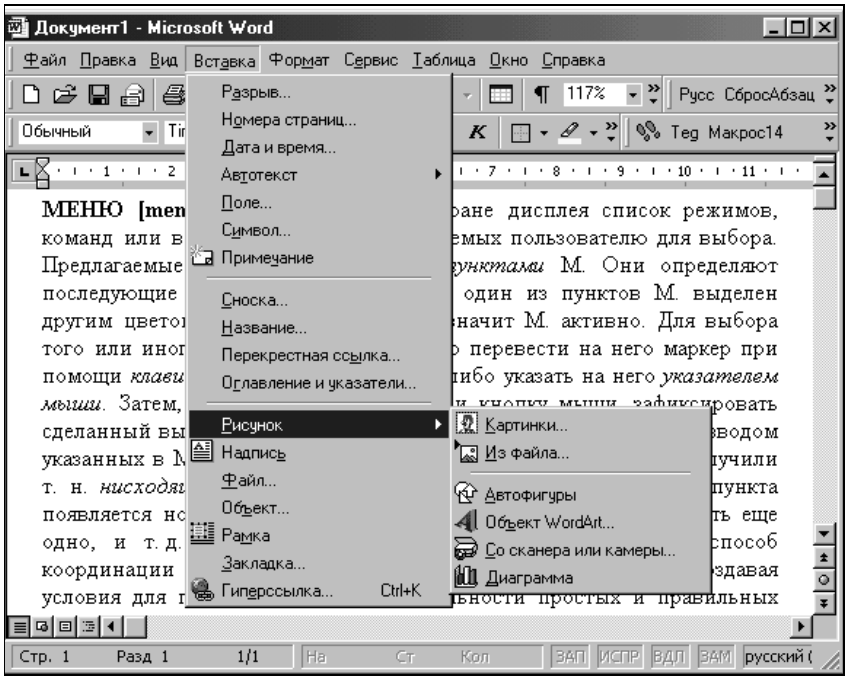

Рис. М.5. Горизонтальное и два раскрытых вертикальных нисходящих пиктографических меню в приложении Microsoft Word

МЕТАДАННЫЕ [metadata]. Информация о *данных*. Например, в *базах данных* — это таблицы описания хранимых данных и связей между ними, адресные таблицы и т. п. М. используются СУБД для поиска, просмотра и трансформации данных. М. необходимы *пакетам прикладных программ, оболочкам* и другим *инструментальным программным средствам* для работы с данными

МЕТАКЛАСС [metaclass]. Класс, экземпляры которого также являются классами. М. используются, как правило, при создании метамоделей

МЕТАМОДЕЛЬ [metamodel]. Модель, которая определяет допустимые элементы моделирования. Элементами М. являются, как правило, метаклассы. Например, М. *унифицированного языка моделирования* UML содержит такие метаклассы, как "класс", "ассоциация", "состояние", "переход" и т. д.

МЕТАКОМПЬЮТЕР [metacomputer]. См. *метакомпьютинг*

МЕТАКОМПЬЮТИНГ [metacomputing]. Использование *компьютерных сетей* для создания распределенной *вычислительной системы* национального и мирового масштаба. Цель М. заключается в том, чтобы создать *программное обеспечение*, объединяющее подключенные к Интернету пространственно

распределенные компьютеры и *периферийные устройства* большой мощности в сверхкомпьютер, или *метакомпьютер*, который для пользователей и программистов выступал бы в виде единой вычислительной среды, доступной с любого *персонального компьютера* или *рабочей станции*. При этом пользователь будет иметь полную иллюзию работы с одной машиной, но гораздо большей, чем та, что стоит у него на столе

МЕТАФАЙЛ [metafile]. 1. *Файл операционной системы*, который содержит каталоги и определяющие характеристики файлов, хранящихся на конкретном *запоминающем устройстве*. 2. В *операционной системе Windows* формат файла, позволяющий хранить разнородную информацию, например, изображение, звук и т. п.

МЕТАЯЗЫК [meta language]. Язык, используемый для описания других языков. Например, *формальный язык Бэкуса—Наура*

МЕТКА [label]. 1. Целое число без знака или *идентификатор*, приписанный оператору программы и используемый в других частях программы для обращения к этому оператору. М. обычно указывается перед оператором и либо отделяется от него двоеточием, как в Паскале и Си, либо ставится на определенном поле программы, как в Фортране. Оператор, снабженный меткой, называют *помеченным оператором*. Например, в Паскале *оператор присваивания* `25: k:= k+1;` помечен М. 25. 2. *Физическая запись* на внешнем *носителе данных*, определяющая начало или конец файла или тома

МЕТОД КЛАССА [class method, method]. Термин *объектно-ориентированного программирования*, которым обозначается функция (или процедура), являющаяся членом класса. См. *класс*. Ср. *свойство класса*

МЕТОД МНОЖЕСТВЕННОГО ДОСТУПА С КОНТРОЛЕМ НЕСУЩЕЙ И РАЗРЕШЕНИЕМ КОНФЛИКТОВ [carrier sense multiple access with collision detection (CSMA/CD)]. То же, что *метод CSMA/CD*

МЕТОДОЛОГИЯ ПРОГРАММИРОВАНИЯ [programming methodology]. Дисциплина, изучающая основополагающие принципы, которые служат методической основой конкретных технологий и инструментальных средств программирования. См. *парадигма программирования*

МЕТОД CSMA/CD, метод множественного доступа с контролем несущей и разрешением конфликтов [carrier sense multiple access with collision detection (CSMA/CD)]. Метод управления доступом в *широковещательных компьютерных* сетях, позволяющий нескольким *рабочим станциям* совместно использовать широковещательный канал. С помощью М. CSMA/CD каждый *узел сети* следит за состоянием линии и начинает передачу только тогда, когда линия освободится. Если конфликт возникает вследствие того, что два узла пытаются занять линию одновременно, оба узла прекращают передачу. Во избежание новых конфликтов они оба ждут в течение случайных

(неодинаковых) интервалов времени, а затем возобновляют попытки передачи. М. CSMA/CD используется в *сетях Ethernet*

МЕТРИКА [metric]. Количественная мера *артефакта* или фазы *процесса разработки программного обеспечения*. Например, *число строк кода* — метрика программного кода

МИКРОКАЛЬКУЛЯТОР, **калькулятор [micro calculator**, **calculator]**. Миниатюрная ЭВМ индивидуального пользования. Современные М. обладают памятью и, помимо выполнения *арифметических операций* и вычисления значений основных алгебраических и тригонометрических функций позволяют программировать на одном из *языков программирования*

МИКРОКОМАНДА [microinstruction]. *Команда микропрограммы*. См. *микропрограммирование*

МИКРОПРОГРАММА [micro program]. *Программа*, задающая последовательность элементарных действий процессора по выполнению *машинной команды*. Эти действия определяются на уровне внутренних регистров, имеющихся в отдельных блоках процессора, и сигналов, поступающих на эти регистры. Каждому такому действию соответствует одна *микрокоманда*. См. *микропрограммирование*

МИКРОПРОГРАММИРОВАНИЕ [microprogramming]. 1. Способ задания и изменения *системы команд* процессора, при котором каждая *машинная команда* задается *микропрограммой*. М. поддаются процессоры с микропрограммным способом управления, при котором каждая машинная команда при вводе ее в *регистр команд* определяет выполнение соответствующей микропрограммы — последовательности микрокоманд, в совокупности реализующей выполнение команды. Микропрограммы выполнения всех машинных команд хранятся в сверхбыстродействующей (обладающей более высоким быстродействием, нежели *оперативная память* компьютера) *микропрограммной памяти* процессора. Это создает возможность модификации системы команд посредством простого изменения содержимого микропрограммной памяти. 2. Процесс создания микропрограмм. Программирование алгоритмов выполнения процессором машинных команд

МИКРОПРОГРАММНАЯ ПАМЯТЬ [micro program store, **control memory]**. Сверхбыстродействующая *память* процессора, в которой хранятся *микропрограммы*, управляющие выполнением машинных команд. Она может быть фиксированной (ПЗУ) или гибкой, допускающей перезапись содержимого. См. *микропрограммирование*

МИКРОПРОЦЕССОР [microprocessor]. Одна или несколько микросхем, на которых реализуется *процессор*. Представляет собой полупроводниковый кристалл или комплект кристаллов, в которых по особой технологии изго-

товлены *арифметико-логическое устройство*, *устройство управления*, цепи *передачи данных*, регистры и память

МИКРОСЕКУНДА (мксек, мкс) [microsecond (mcsec)]. Единица измерения времени быстропротекающих процессов. 1 мксек = 10^{-6} сек

МИКРОСХЕМА, чип [chip, microcircuit]. *Интегральная схема*, реализующая некоторую сложную функцию. Например, одна из М., реализующих *оперативную память* современного компьютера, представляет собой сверхбольшую интегральную схему размером 1/2 дюйма на 1/4 дюйма, способную хранить многие миллионы бит информации

МИКРОСХЕМА ПАМЯТИ [memory circuit]. *Микросхема*, реализующая функцию *памяти произвольного доступа*. В М. п. используются два основных типа памяти: статическая (SRAM, Static RAM) и динамическая (DRAM, Dynamic RAM). *Ячейки* статической памяти построены на различных вариантах триггеров — схем с двумя устойчивыми состояниями. После записи бита в подобную ячейку она может пребывать в этом состоянии сколь угодно долго — необходимо только наличие питания. Ячейки статической памяти имеют малое время срабатывания (единицы-десятки наносекунд), однако микросхемы на их основе имеют низкую удельную плотность данных и высокое энергопотребление. Поэтому статическая память используется в основном в качестве *кэш-памяти*. В динамической памяти ячейки построены на основе областей с накоплением зарядов, занимающих гораздо меньшую площадь, нежели триггеры, и практически не потребляющих энергии при хранении. При записи бита в такую ячейку в ней формируется электрический заряд, который сохраняется в течение нескольких миллисекунд. Для постоянного сохранения заряда ячейки необходимо регенерировать — перезаписывать содержимое для восстановления зарядов. По сравнению со статической памятью ячейки динамической памяти имеют большее время срабатывания (десятки-сотни наносекунд), но бóльшую удельную плотность и меньшее энергопотребление. Динамическая память используется в качестве основной. В настоящее время применяются следующие М. п. FPM DRAM (Fast Page Mode DRAM, динамическая память с быстрым страничным доступом). Позволяет ускорить *страничный обмен* и снизить накладные расходы на регенерацию памяти. М. п. EDO (Extended Data Out, расширенное время удержания данных на выходе) работают в режиме простого конвейера: удерживают на выходах данных содержимое последней выбранной ячейки, в то время как на их входы уже подается адрес следующей выбираемой ячейки. Это позволяет примерно на 15% по сравнению с FPM ускорить процесс считывания последовательных массивов данных. BEDO (Burst EDO, EDO с блочным доступом) — память на основе EDO, работающая не одиночными, а пакетными циклами чтения/записи. Существуют также синхронные виды памяти, получающие внешний синхросигнал, к импульсам которого жестко привязаны моменты подачи адресов и обмена данными. Они позволяют бо-

лее полно использовать внутреннюю конвейеризацию и блочный доступ. SDRAM (Synchronous DRAM, синхронная динамическая память) — память с синхронным доступом, работающая быстрее обычной асинхронной (FPM/EDO/BEDO). VRAM (Video RAM) — микросхемы динамического типа с произвольной выборкой, используемые в видеоадаптерах. Основное требование к памяти типа VRAM — высокое быстродействие. Микросхемы памяти имеют четыре основные характеристики: тип, объем, структура и время доступа. Тип обозначает статическую или динамическую память, объем показывает общую емкость микросхемы, а структура — количество ячеек памяти и разрядность каждой ячейки

МИКРОЭВМ [microcomputer]. *Встроенная* или *персональная ЭВМ* малых размеров, в которой в качестве *арифметико-логического устройства* и *устройства управления* используется *микропроцессор.* В настоящее время термин употребляется редко из-за миниатюризации всех компьютеров

МИЛЛИОН ОПЕРАЦИЙ В СЕКУНДУ [million instructions per second (MIPS)]. Единица измерения скорости работы микропроцессора, равная одному миллиону простейших операций в секунду. Ср. *Мегафлопс*

МИНИ-ДРАЙВЕР [mini driver]. См. *универсальный драйвер принтера*

МНОГОАДРЕСНАЯ РАССЫЛКА ПАКЕТОВ [multicasting]. В *сети передачи данных* рассылка пакетов сразу нескольким (или даже всем) узлам. Ср. *широковещание*

МНОГОДОКУМЕНТНЫЙ ИНТЕРФЕЙС, **архитектура MDI [multi-document interface**, **multiple document interface (MDI)]**. Стандартная архитектура *интерфейса пользователя* для приложений Windows, которая допускает одновременное использование нескольких документов или форм. Каждый документ выводится на экран в *дочернем окне* основного *родительского окна* приложения

МНОГОЗАДАЧНАЯ ОПЕРАЦИОННАЯ СИСТЕМА [multitasking operating system]. *Операционная система*, способная одновременно поддерживать выполнение на компьютере нескольких задач. См. *мультипрограммирование*

МНОГОМАШИННЫЙ ВЫЧИСЛИТЕЛЬНЫЙ КОМПЛЕКС [multiple computer complex]. То же, что *вычислительный комплекс*

МНОГОПОТОЧНОСТЬ [multithreading]. Свойство *операционной системы*, позволяющее одному процессу иметь несколько *потоков управления* и, таким образом, разрешающее приложению выполнять несколько операций одновременно. Например, в современных *текстовых процессорах* можно вводить текст и одновременно проверять его орфографию. М. обеспечивает максимально эффективное использование кванта времени, выделяемого процессу операционной системой

МНОГОПРОЦЕССОРНАЯ ЭВМ, **мультипроцессорная ЭВМ [multiprocessor computer]**. *ЭВМ*, имеющая два и более *процессора*, функционирующих под управлением единой *операционной системы*

МНОГОУРОВНЕВАЯ АРХИТЕКТУРА ПРИЛОЖЕНИЯ [multi-tiered architecture]. Способ организации взаимодействия программ или *компонентов программы*. Как правило, М. а. п. используется в *распределенных приложениях*, компоненты которых выполняются на разных компьютерах. Частным случаем М. а. п. является *архитектура "клиент-сервер"*. В последнее время в *информационных системах* получила распространение архитектура, в которой распределенное приложение состоит из компонентов трех уровней: 1) компонент, ответственный за управление данными, выполняется на сервере баз данных; 2) компонент, осуществляющий обработку данных, выполняется на сервере приложений; 3) компонент, реализующий интерфейс с пользователем, исполняется на *рабочей станции*

МНОГОЦЕЛЕВЫЕ РАСШИРЕНИЯ ПОЧТЫ ИНТЕРНЕТА [multipurpose internet mail extension (MIME)]. Расширения возможностей *электронной почты* сети Интернет, связанные с пересылкой не только текстовых сообщений, но и любых двоичных данных, например, графических, видео и т. д. См. *стандарт MIME*

МНОГОЯЗЫКОВАЯ ПОДДЕРЖКА [national language support (NLS)]. Применяемые в *операционных системах Windows* средства, обеспечивающие возможность подготовки документов на различных языках. При этом можно использовать дополнительные *раскладки клавиатуры* и символы национальных алфавитов. М. п. не меняет язык выводимых на экран *системных сообщений*

МНОЖЕСТВЕННАЯ ЗАГРУЗКА [multiple boot configuration]. Метод, разрешающий устанавливать и использовать на одном компьютере несколько *операционных систем*. При этом на первом этапе загрузки пользователь должен выбрать конкретную операционную систему, после чего управление передается загрузчику этой операционной системы

МНОЖЕСТВЕННОЕ НАСЛЕДОВАНИЕ [multiple inheritance]. Допускаемый в некоторых языках *объектно-ориентированного программирования*, например, в *языке программирования Си++*, такой способ *наследования*, когда данный класс может иметь несколько классов, которым он непосредственно наследует

МОБИЛЬНОСТЬ ПРОГРАММЫ [program portability]. То же, что *переносимость программы*

МОБИЛЬНЫЙ КОМПЬЮТЕР [mobile computer, **mobile PC]**. *Персональный компьютер*, который легко можно перемещать, носить с собой и использовать в различных условиях, в том числе и при отсутствии электрической сети, в отличие от *настольного компьютера*, который обычно стационарно на-

ходится на рабочем месте и не перемещается. Например, *портативный компьютер, блокнотный компьютер, организатор личной деятельности*

МОДАЛЬНОЕ ДИАЛОГОВОЕ ОКНО, **модальное окно [modal dialog**, **modal dialog box**, **modal dialog window**, **modal window]**. *Диалоговое окно*, которое пользователь должен закрыть тем или иным способом (т. е. завершить выполнение функции М. д. о.), прежде чем продолжить работу с приложением. Другими словами, М. д. о. имеет *фокус* тогда и только тогда, когда оно открыто. В противоположность М. д. о. немодальное диалоговое окно позволяет чередовать выполнение функций с помощью данного окна и другую работу с приложением, при этом немодальное окно не закрывается, его можно в любой момент снова активизировать (переместить фокус) и выполнить его функцию при ранее установленных параметрах. Например, на рис. М.6 показано немодальное окно **Найти и заменить** в приложении MS Word, причем в данный момент окно не активно, хотя и открыто, а фокус имеет окно документа. На рис. М.7 показано модальное окно **Параметры** в приложении MS Word, причем в данный момент окно активно, а окно документа фокуса не имеет

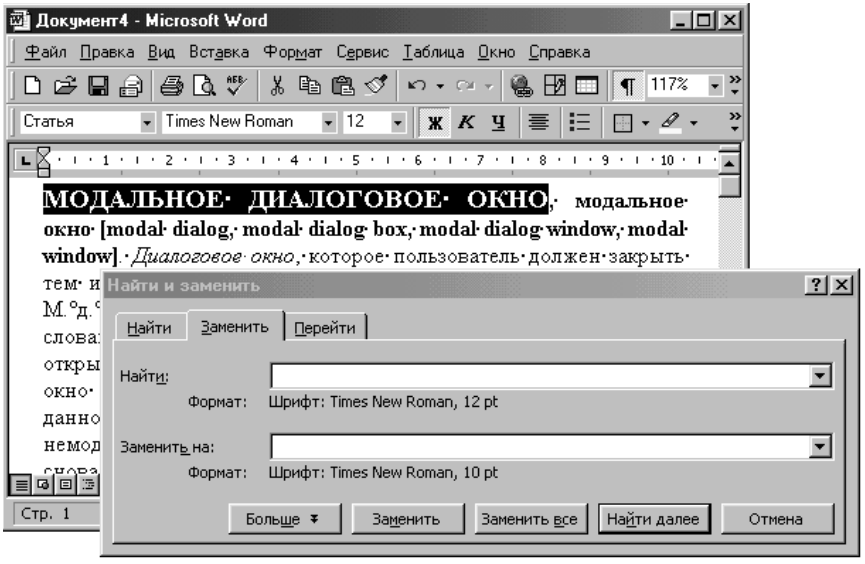

Рис. М.6. Немодальное окно **Найти и заменить** в приложении Microsoft Word

МОДАЛЬНОЕ ОКНО [modal window]. То же, что *модальное диалоговое окно*

МОДЕЛЬ [model]. Семантически законченная абстракция системы

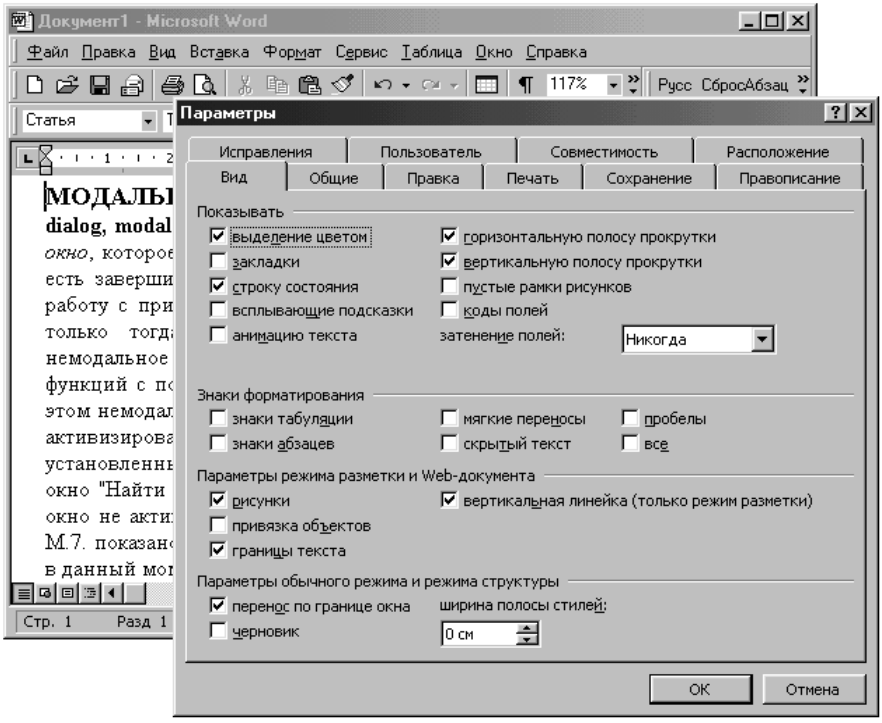

Рис. М.7. Модальное окно **Параметры** в приложении Microsoft Word

МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ ОТКРЫТЫХ СИСТЕМ [International Organization for Standardization/Open System Interconnection model (ISO/OSI model)]. То же, что *модель OSI/ISO*

МОДЕЛЬ ДАННЫХ [data model]. Совокупность принципов *организации базы данных*. Известно множество различных М. д., но самыми популярными являются иерархическая, сетевая и реляционная. Каждая М. д. предусматривает различные принципы определения, манипулирования и хранения данных в базе данных, но наиболее важным является принцип организации связей между данными в базе данных. М. д. отличаются друг от друга прежде всего способами организации связи между данными. В иерархической М. д. записи данных связаны между собой явным образом, причем способ связывания строго ограничен, а именно, записи данных в иерархической базе данных образуют *иерархию*. Например, *реестр Windows* является иерархической базой данных. В сетевой базе данных записи также явным образом связаны друг с другом, но способ связей не ограничен. Наибольшую известность получила сетевая М. д., описанная в стандартах *КОДАСИЛ*. В реляционной модели данных записи связаны друг с другом неявно, через

значения *атрибутов (полей кортежа)*. См. *иерархическая база данных, реляционная база данных, сетевая база данных*

МОДЕЛЬ ЗРЕЛОСТИ ВОЗМОЖНОСТЕЙ, модель CMM [capability maturity model (CMM)]. Метод, позволяющий оценить возможности организации в целом в области *процесса разработки программного обеспечения*. Создан в *Институте технологий разработки программного обеспечения*

МОДЕЛЬ ПЕРЕДАЧИ СООБЩЕНИЙ [massage passing model]. *Модель программирования параллельных алгоритмов*. Программа при выполнении порождает несколько задач, каждая из которых состоит из последовательного кода и локальной памяти и выполняется на отдельном процессоре. Обмен информацией между задачами (данными и управляющими сигналами) происходит посредством отправки и приема *сообщений*. Здесь архитектура компьютера отличается от фоннеймановской. Компьютер состоит из нескольких процессоров, снабженных своей собственной *оперативной памятью* и включенных в коммуникационную сеть. М. п. с. универсальна. Она может быть реализована на *параллельных компьютерах* как с *распределенной*, так и с *разделяемой памятью*, а также на *кластерных вычислительных системах*. Реализация осуществляется либо специальными языками параллельного программирования, либо расширением обычного последовательного языка средствами обмена сообщениями (например, языки CC+ и FORTRAN M), либо использованием специализированных библиотек в программах, написанных на обычных языках последовательного программирования (например, библиотеки PVM и MPI). См. *SPMD-модель*, *MPMD-модель*, *технология MPI*, *технология PVM*

МОДЕЛЬ ПРОГРАММИРОВАНИЯ [programming model]. Совокупность приемов программирования и *структур данных*, отвечающих архитектуре компьютера, предназначенного для выполнения определенного класса алгоритмов. Например, традиционная последовательная М. п. основана на *архитектуре фон Неймана*, следовательно, в любой момент времени выполняется только одна операция и только над одним элементом данных. Эта модель универсальна, реализуется стандартными языками программирования, например, Фортраном, Си, Си++. Программы отличаются хорошей переносимостью, но невысокой производительностью. Для реализации *параллельных алгоритмов* предназначены несколько параллельных М. п. В большинстве из них программа порождает несколько задач, каждая из которых состоит из последовательного кода и локальной памяти и выполняется на отдельном процессоре. При этом разные модели отличаются механизмом взаимодействия задач и базируются на архитектуре компьютера, отличной от фоннеймановской. См. *модель передачи сообщений*

МОДЕЛЬ CMM [capability maturity model (CMM)]. То же, что *модель зрелости возможностей*

МОДЕЛЬ COM [Component Object Model (COM)]. То же, что *компонентная модель объектов*

МОДЕЛЬ OSI/ISO, модель взаимодействия открытых систем [International Organization for Standardization/Open System Interconnection model (ISO/OSI model)]. Многоуровневая архитектура, разработанная *Международной организацией по стандартизации* (ISO), регламентирующая уровни обслуживания и типы взаимодействия компьютеров, обменивающихся информацией через сеть. *Модель OSI/ISO* делит коммуникации компьютер—компьютер на семь уровней, каждый из которых основан на стандартах, содержащихся в уровне ниже данного. Самый нижний из семи уровней относится исключительно к аппаратным средствам связи; самые высокие уровни — к взаимодействию программного обеспечения, в особенности прикладных программ. Ниже в порядке возрастания приведены краткие описания семи уровней М. *OSI/ISO*. 1) Физический уровень. Направляет неструктурированный поток данных через физическую среду передачи (сетевой кабель). Физический уровень описывает *топологию сети* и определяет метод, используемый для *передачи данных* по сетевому кабелю. 2) Канальный уровень. Осуществляет упаковку неструктурированных данных с физического уровня в т. н. фреймы данных, которые представляют собой логически упорядоченные, структурированные пакеты данных. Точный формат фрейма зависит от топологии сети. Канальный уровень обеспечивает безошибочную передачу фреймов между компьютерами через физический уровень. Фреймы содержат исходный адрес и адрес назначения, что позволяет принимающему компьютеру распознавать и извлекать "свои" фреймы. 3) Сетевой уровень. Отвечает за адресацию сообщений и преобразование логических адресов и имен в физические адреса. На этом уровне определяется *маршрут* от передающего компьютера к принимающему. Сетевой уровень также управляет трафиком сети, осуществляя переключение, маршрутизацию и управление буферизацией пакетов данных, объединяет небольшие фреймы данных, преобразует объемные фреймы в более мелкие пакеты. На принимающем компьютере сетевой уровень выполняет обратное преобразование. 4) Транспортный уровень. Отвечает за распознавание и коррекцию ошибок, а также гарантирует надежную доставку сообщений, создаваемых на уровне приложений. Транспортный уровень переупаковывает сообщения (подобно тому, как сетевой уровень обрабатывает фреймы данных), разрезая длинные сообщения на несколько пакетов и объединяя короткие сообщения. На принимающем компьютере транспортный уровень выполняет распаковку сообщений, сборку исходных сообщений и отправляет уведомление о приеме. 5) Сеансовый уровень. Позволяет двум приложениям на разных компьютерах установить, использовать и завершить соединение, называемое сеансом. Этот уровень обеспечивает распознавание имен и функции безопасности, выполняет синхронизацию пользовательских задач, помещая в поток данных контрольные точки. В случае сбоя выполняется повторная передача данных, следующих за последней контрольной точкой. На сеансовом уровне определяется передающая сторо-

на, момент начала передачи и ее продолжительность. 6) Уровень представления. Устанавливает форму, используемую для обмена данными между компьютерами, объединенными в сеть. На передающем компьютере этот уровень преобразует данные из формата, полученного с уровня приложений, в повсеместно распознаваемый промежуточный формат, а на принимающем — переводит данные из промежуточного формата в формат, распознаваемый уровнем приложения принимающего компьютера. Уровень представления управляет защитой данных в сети, обеспечивая шифрование данных, определяет правила передачи данных и осуществляет *сжатие данных.* 7) Уровень приложений. Обеспечивает доступ прикладных процессов к сетевому сервису и поддержку пользовательских приложений, таких как программное обеспечение для передачи файлов, доступ к базам данных и электронная почта. Ср. *системная сетевая архитектура*

МОДЕМ, **модулятор-демодулятор [modem]**. Устройство, преобразующее *цифровую форму данных* в непрерывный *аналоговый сигнал* и обратно для передачи от одного компьютера к другому через телефонную сеть

МОДЕРАТОР [moderator]. Администратор телеконференции, следящий за соответствием публикаций в ней ее тематике и правилам

МОДИФИКАЦИЯ, **обновление [update**, **updating]**. Изменение программы или данных с сохранением способа их организации и присвоенных идентификаторов. Так, М. файла подразумевает изменение его содержимого с целью отражения новых сведений. При этом название файла не изменяется. М. *программного продукта* подразумевает исправление ошибок, найденных после того, как программа выпущена, или незначительные усовершенствования алгоритма. М. программы либо не отражается в ее названии, либо приводит к небольшому изменению номера версии, например, вместо 3.0 появляется 3.1. Ср. *усовершенствование*

МОДИФИЦИРОВАННАЯ АЛЬТЕРНАТИВНАЯ КОДИРОВКА ГОСТА [modified alternative code]. Применяемая в России десятичная кодировка символов для *персональных компьютеров.* М. а. к. Г. представляет собой таблицу десятичных кодов, в которой каждому вводимому с помощью клавиатуры и изображаемому на экране дисплея символу соответствует код от 0 до 255. Эти символы вместе с кодами приведены в табл. М.2. Чтобы определить код символа по этой таблице, нужно сложить первые числа строки и столбца, на пересечении которых находится символ. Например, символ # находится на пересечении строки, обозначенной числом 30, и столбца — 5. Следовательно, его код 35. Первые 32 символа (коды 0—31) — рабочие символы *операционной системы*, а с помощью остальных пользователь может составлять свои тексты и таблицы. Символ с кодом 32 — пробел. Символы-буквы русского алфавита (кириллица) в компьютере должны быть загружены в знакогенератор видеоадаптера *операционной системой* с поддержкой кириллицы или специальной программой — *драйвером-русификатором.* Коды 176—223 соответствуют *псевдографическим символам*, предназначенным для

изображения рамок, таблиц и диаграмм. Символы с кодами 32—127 можно получить на экране дисплея, нажимая клавиши, на которых данные символы нарисованы. Эти и все остальные символы можно вводить с помощью т. н. *АЛТ-ввода символов*

Таблица М.2. *Модифицированная альтернативная кодировка ГОСТа*

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0										
10										
20										
30				!	"	#	$	%	&	‘
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	‘	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~	□	А	Б
130	В	Г	Д	Е	Ж	З	И	Й	К	Л
140	М	Н	О	П	Р	С	Т	У	Ф	Х
150	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
160	а	б	в	г	д	е	ж	з	и	й
170	к	л	м	н	о	п	¦	¦ –	¦ –	|
180	┤	╡	╢	╖	╕	╣	║	╗	╝	╜
190	╛	┐	└	┴	┬	├	─	┼	╞	╟
200	╚	╔	╩	╦	╠	═	╬	╧	╨	╤
210	╥	╙	╘	╒	╓	╫	╪	┘	г	¦
220	_	¦	¦	—	р	с	т	у	ф	х
230	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
240	Ё	ё	≥	≤	(	)	÷	~	°	•
250	·	√	n	2	¦					

МОДИФИЦИРОВАННАЯ ЧАСТОТНАЯ МОДУЛЯЦИЯ (МЧМ) [modified frequency modulation (MFM)]. Широко используемый метод записи данных на дисках. МЧМ базируется на более ранней методике, называемой частотной модуляцией (frequency modulation). Частота синусоидальной волны сигнала изменяется (модифицируется) в соответствии с записываемой информацией. МЧМ позволяет хранить на диске большее количество информации, чем кодирование частотной модуляцией, и применяется на *жестких дисках* со скоростями передачи до 5 Мбайт/с

МОДУЛЬ [module]. Часть какой-либо хорошо структурированной системы, выполняющая четко определенные функции. Внутреннее строение модуля для функционирования всей системы, как правило, значения не имеет. Например, в технике в этом качестве способно выступать конструктивно завершенное техническое устройство, которое легко заменить другим, выполняющим те же функции, а в программировании это может быть оформленная в виде файла часть программы (см. *программный модуль*)

МОДУЛЬ ПАМЯТИ [memory module]. *Плата* с размещенными на ней *микросхемами памяти*. С одной стороны платы расположены контактные площадки, которые вставляются в специальный разъем на *материнской плате*. В настоящее время используются в основном два типа модулей *оперативной памяти*: SIMM и DIMM. SIMM (Single In line Memory Module) — М. п. с одним рядом из 30 или 72 контактов. Применяется во всех современных материнских платах, а также во многих адаптерах, принтерах и прочих устройствах. SIMM может быть различного объема от 4 до 32 Мбайт. DIMM (Dual In line Memory Module) — М. п., похожий на SIMM, но с двумя рядами контактов (обычно по 84 контакта в ряду). За счет этого увеличивается разрядность или емкость памяти в модуле. М. п. DIMM могут иметь объем от 16 до 64 Мбайт. На SIMM в настоящее время устанавливаются преимущественно микросхемы FPM/EDO/BEDO, а на DIMM — EDO/BEDO/SDRAM

МОДУЛЬ ПОДБОРА ЦВЕТОВ [image color matcher (ICM)]. Специальный компонент *графической подсистемы*, позволяющий обеспечить *аппаратно-независимую* передачу цвета. Для этого применяются специальные таблицы цветов (цветовые профили устройств)

МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ [modular programming]. Способ *разработки программ*, при котором программа разбивается на относительно независимые составные части — *программные модули*. При этом каждый модуль может разрабатываться, программироваться, транслироваться и тестироваться независимо от других. Внутреннее строение модуля для функционирования всей программы, как правило, значения не имеет. При модификации алгоритма, реализуемого модулем, структура программы не должна меняться. См. *структурное программирование, нисходящее программирование*

МОДУЛЯТОР-ДЕМОДУЛЯТОР [modem]. То же, что *модем*

МОИ ДОКУМЕНТЫ [my documents]. Название *папки рабочего стола*, предназначенной для хранения документов, графических или других файлов, к которым требуется быстрый доступ. При сохранении файла в таких программах, как WordPad, файл автоматически сохраняется в папке М. д., если пользователь не указал другое место хранения

МОЙ КОМПЬЮТЕР [my computer]. Название *папки рабочего стола*, предназначенной для доступа к папкам, дискам и другим устройствам персонального компьютера, работающего под управлением операционной системы Windows. На рис. М.8 показано раскрытое окно **Мой компьютер**

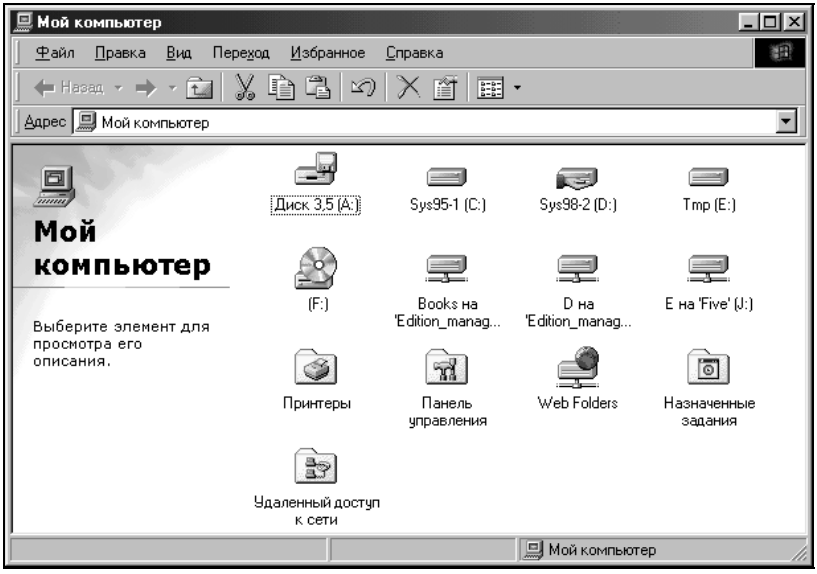

Рис. М.8. Окно **Мой компьютер**

МОНИТОР [monitor]. 1. *Терминал*, предназначенный для контроля либо для контроля и управления состоянием *вычислительной системы*. В качестве М. чаще всего применяется видеотерминал. Дисплей без клавиатуры может использоваться как дистанционный М., позволяющий наблюдать состояние системы на расстоянии. 2. Дисплей *персонального компьютера*. 3. Контролирующая или контролирующая и управляющая *программа*. М. называют программу, которая контролирует действия другой программы и управляет ее работой, например, М., контролирующий функционирование *операционной системы*. Также М. называют программу, контролирующую одну из фаз разных вычислительных процессов и управляющую этой фазой. Например,

в *системах программирования* на *языках программирования высокого уровня* М. обеспечивает организацию совместного использования памяти различными вычислительными процессами, включая возможность обмена параметрами между ними. М. может быть и программа пользователя, и часть операционной системы или системы программирования

МОНОПОЛЬНОЕ ИСПОЛЬЗОВАНИЕ [lockout]. То же, что *захват*

МОРФОЛОГИЧЕСКИЙ ПОИСК [stemming]. Возможность поисковой системы искать слово в документах не только в строго заданном виде, но и во всех его морфологических формах. См. *поисковая система*. Ср. *концептуальный поиск, поиск по ключевым словам, поиск в определенных полях*

МОСТ [bridge]. Совокупность технических и программных средств, обеспечивающая соединение двух и более *локальных вычислительных сетей* с одинаковым протоколом. М. превращает несколько малых сетей в одну сеть большего размера. При этом он пропускает через себя пакеты данных в обе стороны, не выбирая оптимальный маршрут. Ср. *шлюз*

МОСТ-МАРШРУТИЗАТОР, **брутер [bridge-router**, **b-router**, **brouter]**. Устройство, совмещающее функции *моста* и *маршрутизатора*. Используется для соединения двух *сегментов сети* и подключения к Интернету. Обычно М.-м. действует в качестве маршрутизатора для одного из транспортных *протоколов TCP/IP*, направляя пакеты этого формата к точным пунктам назначения. Для остальных типов пакетов М.-м. выполняет функции моста, просто передавая их дальше

МУЗЫКАЛЬНЫЙ СИНТЕЗАТОР [frequency modulation synthesizer]. *Периферийное устройство* или *микросхема*, генерирующая звук в *цифровой форме* по командам пользователя или программы. М. с. создает сложный сигнал путем комбинации цифровых импульсов, представляющих образы звука. Этот сигнал подается на *цифроаналоговый преобразователь*, с которого уже в форме звукового *(аналогового) сигнала* попадает на акустические системы. Используя данные, представляющие цифровой эквивалент нотного листа, компьютер с помощью М. с. может моделировать исполнение как одного музыканта, так и целого оркестра. Как периферийное устройство М. с. может подсоединяться к компьютеру с использованием *цифрового интерфейса музыкальных инструментов MIDI*. М. с. также является частью современных *звуковых плат*. В звуковых платах применяются два метода синтеза звука. 1) WT (Wave Table, таблица волн) — воспроизведение с разной скоростью самплов (от англ. *samples*), заранее записанных в цифровой форме звучаний различных музыкальных инструментов для разных диапазонов нот. Большинство плат содержат встроенный набор инструментов в ПЗУ и позволяют дополнительно загружать в память собственные инструменты. 2) FM (Frequency Modulation, частотная модуляция) — синтез при помощи нескольких генераторов сигнала (обычно синусоидального) со взаимной моду-

ляцией. Каждый генератор снабжается схемой управления частотой и амплитудой сигнала. Последовательность включения генераторов и их параметры (частота, амплитуда и закон их изменения во времени) определяют тембр звучания. При использовании в музыке звучаний реальных инструментов для синтеза лучше всего подходит метод WT; для создания же новых тембров более удобен FM, хотя возможности FM-синтезаторов звуковых карт сильно ограничены из-за своей простоты

МУЛЬТИМЕДИА [multimedia]. 1. Представление информации в форме видеоизображения с применением мультипликации и звукового сопровождения. 2. Комплекс *аппаратных* и *программных средств* для работы с видеоизображением и звуком. Компьютеры, оборудованные М., обычно имеют мощную видеосистему, возможность подключения видеомагнитофона и видеокамеры, аппаратные средства захвата изображения и записи его на *жесткий магнитный диск* в *цифровой форме*, средства наложения видеоизображений, *звуковую плату* для воспроизведения и синтеза звука, привод для считывания информации с *компакт-диска (CD-ROM)*, возможность для подключения акустической системы

МУЛЬТИПРОГРАММИРОВАНИЕ [multiprogramming]. 1. Режим или метод организации работы *вычислительной системы*, при котором одновременно выполняется несколько программ, попеременно использующих одни и те же ее ресурсы. При простейшей форме М. все активные программы находятся в *основной памяти*. При этом одни ресурсы (например, оперативная память и винчестер) делятся между программами, другие используются исключительно одной программой (например, графопостроитель), а третьи эксплуатируются совместно (например, *центральный процессор*). Строго говоря, программы не работают одновременно, а только попеременно эксплуатируют центральный процессор либо исходя из учета приоритетов, либо на основе *квантования времени*. Программы защищены друг от друга. Никакая программа, за исключением супервизора, не может захватить часть оперативной памяти, выделенную другой программе. М. позволяет повысить загруженность центрального процессора и всего оборудования вычислительной системы. Режим М. может быть организован в системе как с одним, так и с несколькими процессорами. Однако на каком-либо процессоре в каждый промежуток времени может выполняться какая-то одна программа. При М. процессоров всегда меньше, чем процессов. 2. Организация программы в виде нескольких взаимодействующих процессов, каждый из которых является последовательной программой. Такая программа выполняется компьютером в режиме М.

МУЛЬТИПРОЦЕССОРНАЯ ЭВМ [multiprocessor computer]. То же, что *многопроцессорная ЭВМ*

МУСОР [garbage]. Ненужные, не подлежащие дальнейшему использованию данные в памяти компьютера (недоступные, устаревшие, недостоверные

и т. п.). М. напрасно занимает место в памяти, поэтому ее следует регулярно очищать от М. Эту процедуру называют *сборкой мусора*. См. *чистка памяти*

МФЛОПС [MFLOPS]. То же, что *мегафлопс*

МЫШЬ [mouse]. *Указательное устройство*, которое представляет собой помещающуюся в ладони коробочку с датчиком перемещения в виде шарика и несколькими кнопками. Обычно М. соединяется с компьютером тонким кабелем через *последовательный порт*. При перемещении М. по плоскости вращение шарика преобразуется в электрические сигналы, управляющие положением *указателя мыши* на экране дисплея. Кнопки М. могут иметь различное назначение, зависящее от используемого драйвера М. и прикладной программы. Например, в текстовом редакторе MS Word левая кнопка служит для выбора объектов и выполнения действий, а правая — для вызова контекстного меню. Позиционированием указателя мыши в определенное место экрана и нажатием кнопки можно сообщить работающей программе некоторые сведения или отдать какие-либо команды. Описанная конструкция является наиболее распространенной, но имеются и другие. Существуют "бесхвостые" М., в которых применяется связь в диапазоне волн теплового излучения. В качестве датчика перемещения может использоваться не механическое, а оптическое устройство (оптические М.). Количество кнопок может быть больше двух, на дополнительные кнопки возлагаются различные специальные функции. Очень удобной является разработанная корпорацией Microsoft М. с дополнительным колесиком, вращение которого позволяет выполнять *скроллинг*. На рис. М.9 показаны различные модели устройства М. Разновидностью М. является *трекбол*. См. *джойстик, трекбол*

Рис. М.9. Различные модели мыши

МЭЙНФРЕЙМ [mainframe]. Универсальная ЭВМ большой мощности, обычно используемая одновременно многими пользователями, работающими на терминалах, подключенных к ней

МЯГКИЙ ПЕРЕНОС [discretionary hyphen]. То же, что *дискретный перенос*

Н

НАВИГАЦИОННЫЙ СПУТНИК [global positioning satellite (GPS)]. Специальный спутник, служащий для определения координат точки на поверхности Земли. Специальное оборудование и *программное обеспечение* позволяют связываться с Н. с. и с высокой точностью определять текущие координаты. Эта возможность часто применяется в бортовых компьютерах на современных автомобилях

НАДСТРОЙКА [add-on module]. То же, что *добавляемый модуль*

НАДСТРОЙКА ОБОЛОЧКИ [shell extension]. Специальное приложение, добавляющее дополнительные возможности *пользовательского интерфейса операционной системы*

НАКОПИТЕЛЬ НА ГИБКИХ МАГНИТНЫХ ДИСКАХ (НГМД) [floppy disk drive]. *Внешнее запоминающее устройство*, в котором *носителями данных* являются *гибкие магнитные диски.* Включает в себя дисковод и сменные дискеты. НГМД могут быть трех видов: для дисков 5.25″, емкостью 360 Кбайт; для дисков 5.25″, емкостью 1.2 Мбайт; для дисков 3.5″, емкостью 1.44 Мбайт. На накопителе для дисков 5.25″ — 1.2 Мбайт можно использовать диск 5.25″ — 360 Кбайт, а на накопителе для дисков 3.5″ — 1.44 Мбайт можно использовать диск 3.5″ — 760 Кбайт. К НГМД относятся и *диски Бернулли* (накопители Zip и Jaz), имеющие сменные *носители данных*, емкость которых доходит до 230 Мбайт и выше. См. *дисковод, гибкий магнитный диск*

НАКОПИТЕЛЬ НА ЛАЗЕРНЫХ ДИСКАХ [laser disk drive]. То же, что *привод CD-ROM*

НАКОПИТЕЛЬ НА МАГНИТНОЙ ЛЕНТЕ (НМЛ) [magnetic tape storage]. *Внешнее запоминающее устройство*, в котором *носителем данных* является *магнитная лента.* Существуют НМЛ, в которых данные хранятся на больших катушках, содержащих по несколько сотен мегабайт на 750 м магнитной ленты, и *стримеры* — НМЛ на стандартных звуковых кассетах. Так как записанная на ленте информация располагается в определенном порядке, то к данным, расположенным в конце ленты, можно получить доступ только прокрутив всю ленту, вплоть до начала нужного участка. Из-за большого *времени доступа* и риска случайного стирания данных и механических повреждений магнитной ленты НМЛ в настоящее время становятся все менее популярными. См. *стример*

НАКОПИТЕЛЬ НА МАГНИТНЫХ ДИСКАХ (НМД) [magnetic disk storage]. *Внешнее запоминающее устройство*, в котором *носителями данных* являются *магнитные диски.* НМД бывают со съемными и несъемными дисками. Как правило, в первом случае носителем данных является гибкий магнитный

диск, а во втором — жесткий. См. *винчестерский диск, накопитель на гибких магнитных дисках, пакет магнитных дисков*

НАНОСЕКУНДА (нсек, нс) [nanosecond (nsec)]. Единица измерения времени быстро протекающих процессов. 1 нсек = 10^{-9} сек

НАРАБОТКА НА ОТКАЗ, среднее время между отказами [mean time between failures (MTBF)]. Средний интервал времени до первого сбоя устройства, обычно выраженный тысячами или десятками тысяч часов. Н. н. о. определяется фирмой-изготовителем и является важнейшей характеристикой надежности устройства

НАСЛЕДОВАНИЕ [inheritance]. Понятие *объектно-ориентированного программирования*, которое состоит в том, что класс, определяемый на основе другого класса, наследует все или некоторые *свойства* и *методы* родительского класса. Применение Н. позволяет программисту сократить программу и сделать ее более наглядной, понятной и эффективной, поскольку общие свойства и методы для группы классов достаточно описать только один раз в их общем родительском классе. В большинстве случаев класс может иметь только один родительский класс, на основе которого непосредственно определен данный класс. Однако в некоторых языках программирования, например, в *языке программирования Си++*, допускается так называемое *множественное* Н., когда данный класс может иметь несколько родительских классов. Если класс А наследует классу В, то говорят, что класс А является *подклассом* класса В, а класс В является *суперклассом* класса А. См. *класс*

НАСТОЛЬНАЯ РЕДАКЦИОННО-ИЗДАТЕЛЬСКАЯ СИСТЕМА [desktop publishing system]. *Интегрированная система*, предназначенная для автоматизации всего комплекса процессов предпечатной подготовки изданий. В состав Н. р.-и. с. включены *текстовый процессор* и развитые средства *верстки страниц*, а также могут входить и многие другие средства, например, *графический редактор*, позволяющий готовить иллюстрации, редактор шрифтов, обеспечивающий модификацию существующих шрифтов, средства цветоделения, необходимые для подготовки цветных изданий, печатаемых на типографском оборудовании, и пр. Примерами известных Н. р.-и. с. являются PageMaker и Ventura Publisher. Некоторые современные текстовые процессоры, например, Microsoft Word, приближаются по своим возможностям к Н. р.-и. с.

НАСТОЛЬНЫЙ КОМПЬЮТЕР [desktop computer]. *Персональный компьютер*, размещаемый на рабочем столе стационарно и подключенный к электрической сети и, возможно, к коммуникационным сетям. Противоп. *мобильный компьютер*

НАСТРОЙКА АДРЕСОВ [address relocation]. Модификация адресов в *объектном* или *загрузочном модуле*, выполняемая *компоновщиком* или *загрузчиком* при размещении модуля в адресном пространстве. Настройке подвергаются

абсолютные адреса, указывающие внутрь модуля; к такому адресу прибавляется адрес начала модуля

НАСЫЩЕННОСТЬ ЦВЕТА [color saturation]. Количественная характеристика *цвета* в *машинной графике*. Одна из трех характеристик в *цветовой модели* HSB (тон-насыщенность-яркость), выраженная в процентах доля основного цвета в цветовом оттенке. См. *цвет, цветовая модель*

НАЧАЛЬНАЯ СТРАНИЦА [home page]. То же, что *домашняя страница*

НАЧАЛЬНОЕ СОСТОЯНИЕ [initial state]. Особое состояние *конечного автомата*, в котором автомат находится до начала работы, т. е. до обработки первого входного символа

НАЧЕРТАНИЕ ШРИФТА, **стиль шрифта [font style**, **type style]**. Различия в изображении шрифтов одной гарнитуры. Обычно используются следующие основные Н. ш.: 1) обычное или светлое; 2) полужирное; 3) наклонное или курсив. Например, в данном словаре заголовки словарных статей выделены полужирным начертанием, ссылки на другие статьи выделены курсивом, а основной текст оформлен обычным начертанием. В некоторых гарнитурах используются и другие начертания, а также их комбинации, например, *капитель*, полужирный курсив и пр. Таким образом, конкретный шрифт определяется тремя независимыми параметрами: гарнитурой, начертанием и размером. На рис. Н.1 представлены примеры использования различных Н. ш. гарнитуры Times New Roman

Гарнитура Times New Roman. Обычное светлое начертание.
Гарнитура Times New Roman. Полужирное начертание.
Гарнитура Times New Roman. Курсивное начертание.
Гарнитура Times New Roman. Полужирное курсивное начертание.
ГАРНИТУРА TIMES NEW ROMAN. ОБЫЧНАЯ КАПИТЕЛЬ.
ГАРНИТУРА TIMES NEW ROMAN. ПОЛУЖИРНАЯ КАПИТЕЛЬ.
ГАРНИТУРА TIMES NEW ROMAN. КУРСИВНАЯ КАПИТЕЛЬ.
ГАРНИТУРА TIMES NEW ROMAN. ПОЛУЖИРНАЯ КУРСИВНАЯ КАПИТЕЛЬ.

Рис. Н.1. Различные начертания шрифтов гарнитуры Times New Roman

НЕАКТИВНАЯ ПРОГРАММА [inactive program]. Загруженная в оперативную память *программа*, которая в текущий момент не управляет *центральным процессором*. Например, ожидающая обращения *резидентная программа*. Противоп. *активная программа*

НЕАКТИВНОЕ ОКНО [inactive window]. Окно, не используемое в настоящий момент. Н. о. может быть полностью или частично перекрыто другими окнами. Чтобы сделать его активным, как правило, достаточно навести на

него указатель мыши и нажать ее левую кнопку. Противоп. *активное окно* (см. рис. А.3)

НЕВИДИМАЯ ЛИНИЯ [hidden line]. Линия, которую можно или нужно скрыть при пространственном изображении объекта. Например, при изображении куба некоторые ребра будут Н. л. В *системах автоматизированного проектирования* и *программах рисования* существуют команды удаления Н. л.

НЕВИДИМАЯ ПОВЕРХНОСТЬ [hidden surface]. Поверхность, которую можно или нужно скрыть при пространственном изображении объекта. Например, при изображении куба некоторые грани будут Н. п. В *системах автоматизированного проектирования* и *программах рисования* существуют специальные команды удаления Н. п.

НЕВОССТАНОВИМАЯ ОШИБКА ПРИЛОЖЕНИЯ [unrecoverable application error (UAE)]. *Ошибка*, возникшая при выполнении команды процессора или доступе к защищенной области памяти. Как правило, такие ошибки возникают в том случае, когда приложение делает запрос, который система не может обработать. *Операционная система* обычно не может обнаружить Н. о. п., что приводит к нестабильной работе системы или ее полной остановке. См. *общая ошибка защиты*

НЕВЫПОЛНЯЕМЫЙ ОПЕРАТОР [nonexecutable statement]. *Оператор* в программе, не содержащий прямое указание на выполнение некоторого действия, определяющего алгоритм решения задачи, а предназначенный для описания объектов программы либо содержащий другую необходимую для трансляции и выполнения программы информацию. Например, *оператор задания формата* вводимых или выводимых данных. Противоп. *исполняемый оператор*

НЕЗАВИСИМОСТЬ ПРИЛОЖЕНИЙ [application independence]. Метод разработки *приложений*, позволяющий обеспечить независимость приложения от аппаратных и программных особенностей *вычислительной системы*. Для достижения этой цели обычно применяется *язык программирования высокого уровня* и набор *интерфейсов прикладного программирования* (API)

НЕЗАКОННОЕ КОПИРОВАНИЕ ПРОГРАММНЫХ СРЕДСТВ, **несанкционированное копирование**, **пиратство [software piracy**, **piracy]**. *Копирование программного продукта* без согласия владельца авторских прав. Н. к. п. с. производится с целью использования программы незаконными, незарегистрированными пользователями без какой-либо оплаты. Для предотвращения Н. к. п. с. программный продукт часто снабжается средствами *защиты от копирования*. В настоящее время Н. к. п. с. преследуется по закону

НЕИСПРАВИМАЯ ОШИБКА, **фатальная ошибка [unrecoverable error**, **fatal error**, **catastrophic error]**. Ошибка *операционной системы* или приложения,

последствия которой не могут быть устранены средствами операционной системы. Н. о. вызывает аварийный останов или аварийное завершение программы, возможно, с потерей данных, транзакций, файлов и т. п. Устранение последствий Н. о. требует вмешательства пользователя или *оператора ЭВМ*. См. *невосстановимая ошибка приложения*

НЕМОДАЛЬНОЕ ДИАЛОГОВОЕ ОКНО [non modal window]. См. *модальное диалоговое окно*

НЕОСНОВНОЙ КЛЮЧ [secondary key]. То же, что *вторичный ключ*

НЕПЕЧАТАЕМЫЕ СИМВОЛЫ [nonprinting characters]. Символы, которые могут отображаться на экране, но не должны появляться в печатной копии документа, например, пробелы, символы табуляции и конца абзаца и т. п.

НЕПОСРЕДСТВЕННАЯ АДРЕСАЦИЯ [immediate addressing]. То же, что *прямая адресация*

НЕПОСРЕДСТВЕННЫЙ ОПЕРАНД [immediate operand]. *Операнд*, представленный в команде непосредственно своим значением, а не адресом расположения в памяти. Например, в командах типа "проверить по маске" маска может задаваться как Н. о.

НЕПРЕРЫВНЫЙ СИГНАЛ [continuous signal]. То же, что *аналоговый сигнал*

НЕПРИВИЛЕГИРОВАННЫЙ РЕЖИМ [slave mode]. Режим работы *центрального процессора*, при котором выполняются *прикладные программы* и попытка выполнить *привилегированную команду* вызывает прерывание

НЕРЕЗИДЕНТНЫЙ ПУЛ [paged pool]. Часть *оперативной памяти*, страницы из которой могут быть откачаны на диск. Противоп. *резидентный пул*. См. *выгрузка*

НЕСАНКЦИОНИРОВАННОЕ КОПИРОВАНИЕ [software piracy]. То же, что *незаконное копирование программных средств*

НЕСАНКЦИОНИРОВАННЫЙ ДОСТУП [unauthorized access, illegal access]. *Доступ* с нарушением правил пользования данными, *программным продуктом* или сетью. Это может быть доступ лица или программы, которым пользование этими данными не разрешено, или доступ с применением неразрешенных средств вычислительной техники. См. *защита данных, защита памяти, защита от копирования*

НЕСИММЕТРИЧНОЕ ШИФРОВАНИЕ [open key enciphering, open key encryption, RSA enciphering, RSA encryption]. То же, что *шифрование с открытым ключом*

НЕЯВНОЕ ОПИСАНИЕ [implicit declaration]. В некоторых *языках программирования* допускается отсутствие *описания*. В этом случае характеристики

объекта программы, описание которого отсутствует, устанавливаются по виду его идентификатора согласно специальным правилам, принятым в данном языке. Например, в Фортране при отсутствии описания переменных действует такое правило: если идентификатор (имя) переменной начинается с одной из букв I, J, K, L, M, N, то это переменная *целого типа*, значения которой имеют длину 4 байта

НИЖНИЙ ИНДЕКС [subscript]. Числовой или буквенный указатель, напечатанный ниже базовой линии текста. Например, H_2O

НИЖНИЙ КОЛОНТИТУЛ [footer, running foot]. *Колонтитул*, расположенный в нижней части страницы, ниже основного текста

НИЖНИЙ РЕГИСТР [lowercase]. См. *верхний регистр*

НИСХОДЯЩЕЕ МЕНЮ, раскрывающееся меню, спускающееся меню [pull-down menu]. *Меню*, вызываемое указанием его заголовка, появляющееся непосредственно под этим заголовком и исчезающее после окончательного выбора одного из его пунктов. Первоначально Н. м. имеет вид *горизонтального меню*, обычно расположенного в верхней строке экрана. После выбора пункта этого меню появляется новое, как правило, *вертикальное меню*, накладывающееся на содержимое экрана, после выбора в этом меню может возникнуть еще одно поверх предыдущего и т. д. до окончательного выбора (см. рис. М.5). Затем все возникшие меню исчезают. Такие вложенные меню иногда называют подменю

НИСХОДЯЩЕЕ ПРОГРАММИРОВАНИЕ [top-down programming]. Способ *разработки программ*, при котором программирование ведется методом "сверху вниз", от общего к деталям. Алгоритм решения задачи разбивается на несколько более простых частей или подзадач. Их выделяют таким образом, чтобы программирование подзадач было независимым. При этом составляют план решения всей задачи, пунктами которого и выступают выделенные части. План записывают графически в виде *блок-схемы*, где выявляют головную и подчиненные подзадачи и связи между ними, т. е. интерфейс. Здесь же устанавливают, какие начальные данные (или аргументы) получает каждая подзадача для правильного функционирования и какие результаты она выдает. По блок-схеме составляется программа, в которой содержатся *вызовы подпрограмм* (процедур или функций), соответствующих выделенным подзадачам. Эту программу можно сразу отлаживать, временно заменив "заглушками" подпрограммы для подзадач. Аналогично производят детализацию и программирование каждой подзадачи. Процесс последовательной детализации идет до тех пор, пока не будет написана программа для каждого фрагмента алгоритма. При этом на каждом этапе Н. п. имеется действующий вариант программы, отладка которой ведется по ходу всей разра-

ботки программы. Противоп. *восходящее программирование.* См. *структурное программирование, модульное программирование, заглушка*

НОБОТ [knowbot]. Программа сбора информации. Термин образован по аналогии со словом "робот"

НОМЕР ПРОЦЕССА, ранг [rank]. *Идентификатор* процесса в *MPI-программе.* Н. п. является целым числом от 0 до $n - 1$, где n — число взаимодействующих процессов, входящих в один коммуникатор. См. *интерфейс передачи сообщений*

НОРМАЛИЗОВАННАЯ ФОРМА [normalized form]. Форма представления *числа с плавающей точкой*, при которой мантисса M удовлетворяет двум условиям: $|M| < 1$ и старшая цифра мантиссы отлична от нуля. Например, среди трех форм одного и того же числа с плавающей точкой: $1.2345 \cdot 10^2$, $0.012345 \cdot 10^4$ и $0.12345 \cdot 10^3$ Н. ф. является только третья

НОРМАЛИЗОВАННОЕ ЧИСЛО [normalized number]. *Число с плавающей точкой*, записанное в *нормализованной форме*

НОРМАЛЬНАЯ ФОРМА БЭКУСА—НАУРА (БНФ) [Backus—Naur form (BNF)]. То же, что *формальный язык Бэкуса—Наура*

НОСИТЕЛЬ ДАННЫХ [data medium]. Материальный объект или устройство с определенными физическими свойствами, позволяющими использовать его для записи, хранения и считывания данных. В качестве Н. д. в вычислительной технике применяются полупроводниковые кристаллы, *магнитные* и *лазерные диски*, *магнитные ленты*, *магнитные карты*, перфокарты и перфоленты, а также бумага (для распечатки). Совместно с устройствами, осуществляющими запись данных на Н. д. и их считывание, Н. д. образуют *запоминающие устройства*

НОТАЦИЯ [notation]. Правила записи предложений формального языка, в котором для передачи смысла используются не только тексты. Например, Н. для записи математических формул использует специальные символы и учитывает взаимное положение этих символов на плоскости (рис. Н.2). В *языках визуального моделирования* обычно применяется *графическая Н.* Ср. *синтаксис*

$$\frac{a \cdot b}{x \cdot y} \neq a \cdot b / x \cdot y$$

Рис. Н.2. Пример математической формулы

НОУТБУК [notebook]. То же, что *портативный компьютер*

О

ОБВОДКА [outline]. *Контур*, которому присвоены параметры толщины, цвета и пр.

ОБЛАСТЬ ВИДИМОСТИ, **видимость [visibility]**. Термин *объектно-ориентированного программирования*, обозначающий правила определения участков программы, на которые распространяется описание *свойств* и *методов* некоторого *класса* и где допустимо использование имен его свойств и методов. В большинстве языков различаются *открытые*, *защищенные* и *закрытые* свойства и методы. Открытые свойства и методы могут применяться везде, где виден сам класс. Защищенные свойства и методы — в самом классе и во всех классах, которые наследуют данному классу. Закрытые свойства и методы могут быть использованы только внутри данного класса

ОБЛАСТЬ ДЕЙСТВИЯ [owner scope, scope]. В *объектно-ориентированном программировании* — указание на то, имеют ли экземпляры класса индивидуальные значения свойства или метода (в таком случае говорят, что "О. д. является экземпляр"), или же все экземпляры класса совместно используют одно значение свойства или метода (в таком случае говорят, что "О. д. является класс")

ОБЛАСТЬ ОПРЕДЕЛЕНИЯ, **область существования [definitional domain]**. Участки программы, на которые распространяется описание некоторого объекта и где допустимо использование его имени. В большинстве *языков программирования* принято следующее правило. Определенные в описании имя константы, типа, переменной, процедуры или функции либо метки действительны во всем *программном блоке*, содержащем это описание, и во всех его *вложенных блоках* (подблоках), за исключением тех подблоков, где эти имя или метка определены заново. Если, например, переменная описана в некотором блоке, содержащем подблоки, то она доступна и в блоке, и во всех его подблоках, т. е. О. о. такой переменной является сам блок и все его подблоки. Такая переменная является *локальной переменной* для блока, в котором она описана, и *глобальной переменной* для каждого из подблоков. См. *вложенный блок*. Ср. *область видимости*

ОБЛАСТЬ ПАМЯТИ [storage area]. Последовательность адресуемых элементов *памяти* (байтов, *ячеек памяти* и т. п.) с адресами в заданном диапазоне, выделенная для размещения одной или нескольких порций данных

ОБЛАСТЬ СУЩЕСТВОВАНИЯ [definitional domain]. То же, что *область определения*

ОБМЕН ДАННЫМИ, **обмен [exchange]**. Обобщающее название любой *передачи* и приема *данных* в *вычислительной системе* или *компьютерной сети*, если при этом определены или подразумеваются *источник* и *приемник дан-*

ных. Например, О. д. между винчестером и *оперативной памятью.* Под О. д. обычно подразумевают весь процесс пересылки данных, включая кодирование, декодирование, буферизацию и *контроль по четности*

ОБНОВЛЕНИЕ 1. **[update]**. То же, что *модификация.* 2. **[upgrade]**. То же, что *усовершенствование*

ОБНУЛЕНИЕ [zero filling]. Присваивание нулевого значения или занесение непрерывной последовательности нулей в *поле данных* или *область памяти*

ОБОБЩЕНИЕ [generalization]. Отношение между более общим и более специализированным элементами. Более специализированный элемент полностью совместим с более общим и содержит дополнительную информацию. Экземпляр более специализированного элемента можно использовать везде, где объявлен более общий элемент. В *объектно-ориентированных языках программирования наследование* является О. В *унифицированном языке моделирования* UML О. является одним из основных отношений, применяемых на *диаграммах классов.* См. *принцип подстановочности*

ОБОЛОЧКА [shell]. Часть программы, программа или совокупность программ, предназначенная для комплексной автоматизации работы человека с другой программой или *комплексом программ.* О. является программным средством *дружественного интерфейса*, создающим для человека наглядные, простые и понятные изображения на экране, удобные средства управления: *кнопки*, *меню*, подсказки в диалоге и т. п. См. *оболочка операционной системы, графический интерфейс пользователя*

ОБОЛОЧКА NORTON COMMANDER [Norton Commander (NC)]. *Оболочка операционной системы MS-DOS персональных компьютеров*, разработанная фирмой Peter Norton Computing, Inc. в 1983 г. NC обладает характерным набором функций и элементов внешнего оформления, делающим работу с компьютером комфортной для пользователя. Среди них основными являются: *оперативная помощь*; *клавиши быстрого вызова*; *панели экрана*, несущие информацию о дисках; *нисходящее меню*; *командная строка*; окна; мышь. Подробная оперативная помощь вызывается нажатием клавиши <F1>. Управлять функциями NC можно с помощью "горячих клавиш" или меню. Иногда одни и те же действия вызываются и тем, и другим способом. Вид экрана NC с установленным нисходящим меню показан на рис. О.1. Горизонтальная строка нисходящего меню возникает в верхней (первой) строке экрана или становится активной (один из пунктов окрашивается в другой цвет) после нажатия клавиши <F9>. Нисходящее меню также можно вызвать на экран или сделать активным, если поместить *указатель мыши* в 1-ю строку и нажать кнопку мыши. Две прямоугольные панели располагаются в строках с 1 по 23. В них видна информация о каталогах и файлах, хранящихся на дисках (имена, размер в байтах, дата и время записи или создания). 24-я строка экрана называется командной строкой и предназначена для ввода *команд*

операционной системы и *запуска программ* с помощью клавиатуры. В 25-й строке находится краткое описание действия *функциональных клавиш*, а при нажатии клавиш <Alt> или <Ctrl> — описание действия функциональных клавиш в сочетании с указанными клавишами. Эти же действия можно вызвать, если поместить указатель мыши на 25-й строке в зону соответствующей клавиши и нажать кнопку мыши. В различных ситуациях работы NC на экране появляются окна, содержащие предупреждения, *поля ввода*, меню и установки. В полях ввода набирается запрашиваемая операционной системой текстовая информация, например, имя файла. Установки содержат текст, предлагающий тот или иной режим работы, и поле подтверждения — квадратные скобки, в которые вносится символ "x", если пользователь подтверждает установку, или пробел, если не подтверждает. В первом случае NC работает в соответствии с текстом установки, во втором установка игнорируется. Перемещения между полями ввода, установками и меню осуществляются клавишами <↑>, <↓>, <Tab>. Перемещения между *пунктами меню* — клавишами <←> и <→>. Интерфейс NC оказался настолько удачным и удобным в использовании, что практически без изменений (с добавлениями новых возможностей) воспроизводится в многочисленных современных оболочках операционных систем: FAR Manager, Windows Commander и др.

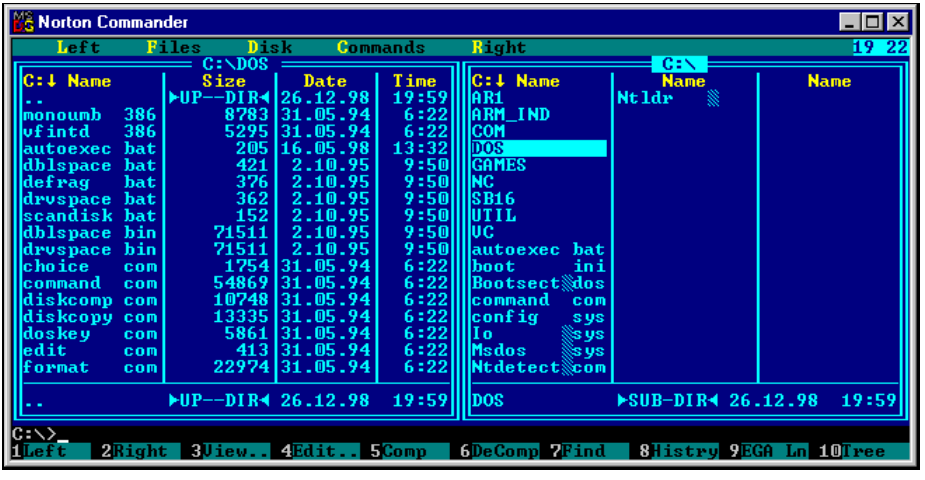

Рис. О.1. Вид экрана NC с панелями и строкой нисходящего меню

ОБОЛОЧКА ОПЕРАЦИОННОЙ СИСТЕМЫ [shell]. *Резидентная программа*, облегчающая взаимодействие человека-пользователя с *операционной системой*. Основное назначение О. о. с. — максимально упростить работу пользователя с файлами и каталогами. В примерный перечень функций, автоматизацию которых обеспечивает О. о. с., входят: *переименование файла*;

копирование файла (группы файлов); *перенос файла* (группы файлов) в другой каталог; *удаление файла* (группы файлов); *поиск файла* (группы файлов); *сортировка файлов* в каталоге; *создание каталога*; *переименование каталога*; *перенос каталога*; *удаление каталога*; просмотр *дерева каталогов*. Кроме этих у О. о. с. может быть еще много других *сервисных функций*, делающих работу с операционной системой комфортной. Широко известна *оболочка Norton Commander*, разработанная фирмой Peter Norton Computing, Inc. для *операционной системы MS-DOS*, и многочисленные современные О. о. с., поддерживающие аналогичный *интерфейс пользователя*: FAR Manager, Windows Commander и др.

ОБОРУДОВАНИЕ [equipment]. 1. Совокупность *аппаратных* и *программных средств*, направленных на решение компьютером определенной задачи. 2. То же, что *аппаратные средства*

ОБРАБАТЫВАЮЩАЯ ПРОГРАММА [processing program]. Программа, обрабатывающая данные или другие программы. Например, *текстовый редактор, компилятор, редактор связей, загрузчик*

ОБРАБОТКА ДАННЫХ, **манипулирование данными, обработка информации [data processing**, **data manipulation]**. Выполнение по заданной программе определенных действий над *данными*. К таким действиям относятся, например, *поиск данных*, *сортировка данных*, их анализ и объединение. Экономические и инженерные расчеты, решение научно-технических задач и задач управления производством тоже являются процессами О. д. В них *исходные данные* путем вычислений преобразуются в конечный результат

ОБРАБОТКА ЗАПРОСА [query processing]. Действия *операционной системы*, *диалоговой системы* или *системы управления базой данных* при подготовке запроса к выполнению, а также проведение завершающих операций. Действиями при подготовке могут быть чтение и интерпретация запроса, постановка в очередь, выделение ресурсов, выборка из очереди и т. п. Примером завершающей операции может быть выдача сообщения об исполнении запроса или о невозможности исполнения

ОБРАБОТКА ИНФОРМАЦИИ [information processing]. То же, что *обработка данных*

ОБРАБОТКА ПРЕРЫВАНИЙ [interrupt processing]. Стандартные действия, выполняемые *операционной системой* или *аппаратными средствами* при возникновении *прерываний*. Сначала производится запоминание состояния прерванной программы и операционной системы, и в зависимости от типа прерывания управление передается одной из программ — *обработчиков прерываний*, входящих в состав операционной системы (для некоторых прерываний обработчик прерываний может быть задан пользователем). Обработчик фиксирует информацию об условиях и причинах прерывания и выполняет необходимую программу О. п. в соответствии с характером пре-

рывания. Например, если поступило *прерывание от схем контроля* в результате *машинного сбоя*, то обработчик анализирует характер сбоя и пытается исправить ошибку повторением команды. Если это не удается сделать, то выполнение задачи, на которую действует ошибка, может быть прекращено. Если же последствия сбоя удалось устранить, то обработчик выдает сообщение об этом операционной системе и затем передает ей управление. Далее система возвращается в предыдущее состояние, что позволяет продолжить выполнение исходной задачи так, как будто ничего не случилось

ОБРАБОТКА ТРАНЗАКЦИЙ [transaction processing]. Действия СУБД по управлению параллельным выполнением нескольких *транзакций*. СУБД следит, к каким данным обращаются транзакции, выполняет *блокировку данных*, предотвращает *тупики* и разрешает *конфликты*, выполняет *завершение* и *откат транзакций*

ОБРАБОТЧИК ПРЕРЫВАНИЙ [interrupt handler]. Программа, выполняющая *обработку прерываний*

ОБРАЗЕЦ ПРОЕКТИРОВАНИЯ, паттерн [design pattern]. Документированный способ решения отдельной типовой задачи проектирования программного обеспечения. За последние годы были опубликованы сотни О. п., предложенных наиболее умелыми разработчиками *программного обеспечения*. Применение О. п. сокращает время на разработку и повышает надежность, поскольку используются проверенные решения. О. п. публикуются в различном виде, в частности, в форме текстовых описаний с примерами на *языках программирования* или в форме параметризованных *коопераций унифицированного языка моделирования* UML

ОБРАЗ ЗАДАЧИ [task image]. 1. Состояние *оперативной памяти* задачи, записанное во *внешнюю память*. О. з. включает содержимое *областей памяти*, отведенных для программы, *рабочих областей* и связанных с ними *блоков управления*, а также используемых программой регистров. *Многозадачная операционная система* формирует О. з. с целью освобождения места в оперативной памяти для задачи с более высоким приоритетом или в режиме *разделения времени*. Когда будет возможно, О. з. снова загрузится в оперативную память и выполнение задачи продолжится. Кроме того, О. з. может периодически формироваться операционной системой с целью восстановления *вычислительного процесса* в случае *машинного сбоя*, отказа оборудования или неверных ошибочных действий пользователя при *диалоговом режиме* управления программой. См. *откат, контрольная точка*. 2. То же, что *загрузочный модуль*

ОБРАМЛЕНИЕ, окантовка [border]. 1. То же, что *граница окна*. 2. Декоративная линия, ограничивающая страницу, рисунок и т. п.

ОБРАТНОЕ ПРОЕКТИРОВАНИЕ [reverse engineering]. Построение *артефакта* предыдущей фазы *процесса разработки программного обеспечения* по

артефактам последующей фазы, например, построение модели по программному коду

ОБРАТНОЕ ПРОСЛЕЖИВАНИЕ [backtracking]. То же, что *поиск с возвратами*

ОБРАТНЫЙ ВЫЗОВ [callback]. То же, что *возвратный вызов*

ОБРАЩЕНИЕ К ВНЕШНЕМУ УСТРОЙСТВУ [device access]. 1. Действия *вычислительной системы*, предпринимаемые или подразумеваемые при установлении связи с *внешним устройством*, включая выполнение устройством определенных функций. Например, обращение к *запоминающему устройству* предполагает доступ к устройству и полный цикл записи или чтения информации. 2. *Ссылка* на *внешнее устройство*

ОБРАЩЕНИЕ К ПОДПРОГРАММЕ [call, subroutine call]. То же, что *вызов подпрограммы*

ОБРАЩЕНИЕ К ПРОЦЕДУРЕ [procedure call]. То же, что *вызов процедуры*

ОБРАЩЕНИЕ К ФУНКЦИИ [function call]. То же, что *вызов функции*

ОБРЕЗКА [crop, cropping]. То же, что *обрезка изображения*

ОБРЕЗКА ИЗОБРАЖЕНИЯ, обрезка, кадрирование [crop, cropping]. Редактирование изображения путем выделения и сохранения его фрагмента нужного размера и удаления остатка. О. и. применяется при *верстке страниц* для укладки изображения в заданное пространство. В некоторых *графических редакторах* и *настольных издательских системах* для выполнения О. и. служит специальный *инструмент машинной графики*, обычно называемый "рамка". Кадрируемая область выделяется прямоугольной штриховой рамкой, снабженной квадратными маркерами. Перемещая маркеры *указателем мыши,* можно изменять размеры области, поворачивать и перемещать ее. Для завершения операции необходимо ввести указатель мыши в пределы рамки (указатель превращается в ножницы) и дважды щелкнуть

ОБСЛУЖИВАЮЩАЯ ПРОГРАММА, сервисная программа, служебная программа, утилита [service program, utility]. *Программа* в составе *вычислительной системы*, выполняющая общие вспомогательные функции. Например, программа *дефрагментации диска* или поиска и диагностики *компьютерного вируса.* Системные О. п. входят в состав *операционной системы* и выполняют повседневные задачи, например, перезапись данных с одного устройства на другое. Существуют автономные О. п., которые для своего выполнения не требуют *операционной системы*

ОБУЧАЮЩАЯ ПРОГРАММА [learning program, on-line tutorial]. 1. *Программа*, выполняющая в *диалоговом режиме* обучение человека по некоторой дисциплине или по одному из ее разделов. Например, программа обучения одному из *языков программирования высокого уровня.* О. п. предлагает обучае-

мому порции учебного материала и контрольные задания по ним в той последовательности, которую требует методика обучения данной дисциплине; определяет правильность выполнения контрольных заданий и указывает характер допущенных ошибок. В зависимости от достигнутых результатов О. п. выбирает дальнейшую последовательность прохождения курса и удобный для обучаемого темп подачи материала. О. п., как правило, обладают *дружественным интерфейсом*. К О. п. относятся также программы, управляющие обучением людей на тренажерах, позволяющих выработать у человека определенные навыки, необходимые в его профессиональной деятельности. Например, О. п. авиатренажеров. См. *автоматизированная обучающая система, интеллектуальная обучающая система*. 2. Часть *документации программного продукта* в виде программы, которая моделирует его работу и предлагает обучающемуся пользователю типичные ситуации, возникающие при работе, контролирует действия пользователя и исправляет ошибки. Например, программа, обучающая работе с мышью. См. *обучающее приложение к программному продукту*

ОБУЧАЮЩЕЕ ПРИЛОЖЕНИЕ К ПРОГРАММНОМУ ПРОДУКТУ, автоматизированная подготовка [computer based training system (CBT)]. *Обучающая программа*, предназначенная для обучения пользователя работе с конкретным *программным продуктом*. Как правило, О. п. к п. п. реализуется в виде надстройки, которая дает возможность пользователю во время обучения работать с самим программным продуктом, а не с его моделью. Ср. *обучающая программа*. См. *автоматизированная обучающая система*

ОБЩАЯ ОБЛАСТЬ [public area]. Множество веб-страниц или *дерево каталога*, содержание которых доступно для любого пользователя. При этом не требуется идентификации, аутентификации или разрешения.

ОБЩАЯ ПАМЯТЬ [shareable memory, shared memory]. То же, что *разделяемая память*

ОБЩАЯ ПАПКА [shared folder]. Находящаяся на данном или на другом компьютере *папка*, к которой открыт общий доступ

ОБЩИЙ БЛОК [common block]. В языке Фортран — *область памяти*, являющаяся общей как для *вызывающей программы*, так и для вызываемых подпрограмм. Использование О. б. экономит память и сокращает *машинное время*, которое в противном случае потребовалось бы на пересылку *фактических параметров* из программы в подпрограмму и наоборот. О. б. организуются оператором описания О. б., который относится к *невыполняемым операторам* и имеет форму:

```
COMMON / n1 / u1, u2, ..., un/ ... / nk / v1, v2, ..., vm
```

Здесь `n1, n2, ..., nk` — имена О. б.; `u1, u2, ..., un` — имена переменных или массивов или *описания массивов* блока `n1`; `v1, v2, ..., vm` — имена пе-

ременных или массивов или описания массивов блока `nk`. Имена О. б. могут отсутствовать. Массивы, для которых в операторе `COMMON` указаны только имена, должны быть описаны дополнительно. Например, оператор

```
COMMON /A/ X(10), F, R / / Y(15), SUMMA
```

описывает два О. б.: один с именем `A`, включающий массив `X`, состоящий из 10 элементов, и имена переменных или массивов `F` и `R`; другой — непоименованный О. б., включающий массив `Y`, состоящий из 15 элементов, и имя переменной или массива `SUMMA`. Если среди имен `F`, `R` и `SUMMA` есть имена массивов, то для них в программе должно быть соответствующее описание

ОБЩИЙ ПРИНТЕР, сетевой принтер [shared printer, network printer]. *Принтер*, на котором может осуществляться печать с нескольких компьютеров. Например, к принтеру, подключенному к некоторому компьютеру в *локальной вычислительной сети*, может быть открыт доступ для пользователей других компьютеров. См. *общий ресурс*

ОБЩИЙ РЕСУРС, разделяемый ресурс, совместно используемый ресурс [shareable resource, shared resource]. Любой *ресурс* (устройство, данные или программа), доступный для использования несколькими устройствами или программами. В *операционной системе Windows* О. р. называют любые ресурсы, доступные для пользователей сети, такие как папки, файлы, принтеры и *именованные каналы*. См. *общая папка, общий принтер*

ОБЪЕКТ [object]. 1. Общий термин, которым обозначается любая индивидуально выделяемая сущность. Предмет или явление, которому можно присвоить название. 2. Термин *объектно-ориентированного программирования*, которым обозначается *программный объект*, являющийся классом или *экземпляром класса*. См. *класс, метод класса, свойство класса*

ОБЪЕКТ ДАННЫХ [data entity]. Элемент данных, хранимый в *базе данных* и содержащий информацию о реальном процессе, предмете, явлении

ОБЪЕКТ ПРОГРАММЫ, программный объект [program object]. *Элемент данных*, доступный программе и обрабатываемый ею. Например, О. п. являются описанные в программе переменные, массивы, файлы, записи и т. п.

ОБЪЕКТНАЯ ПРОГРАММА [object program]. 1. *Объектный модуль*, представляющий законченную программу. 2. То же, что *объектный модуль*

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ (ООП) [object-oriented programming (OOP)]. Наиболее популярная в настоящее время *парадигма программирования*, являющаяся развитием *структурного программирования*. Центральной идеей ООП является *инкапсуляция*, т. е. структурирование программы на модули особого вида, объединяющие данные и процедуры их обработки, причем внутренние данные модуля не могут быть обработаны только предусмотренными для этого процедурами. В разных

вариациях ООП этот модуль называют по-разному: *класс*, *абстрактный тип данных*, кластер и др. Каждый такой класс имеет внутреннюю часть, именуемую реализацией (или представлением), и внешнюю часть, называемую интерфейсом. Доступ к реализации возможен только через интерфейс. Обычно в интерфейсе различают свойства (которые синтаксически выглядят как переменные) и методы (которые синтаксически выглядят как процедуры или функции). Класс может иметь методы — конструкторы и деструкторы, позволяющие во время выполнения программы динамически порождать и уничтожать *экземпляры класса*. Экземпляры одного класса сходны между собой (например, *наследуют методы класса*), но имеют различия (например, имеют разные значения свойств). Классы и экземпляры классов называют *объектами*, откуда и происходит название О.-о. п. См. *класс, метод класса, объект, область видимости, свойство класса, экземпляр класса*

ОБЪЕКТНЫЙ БРОКЕР ЗАПРОСОВ [object request broker (ORB)]. См. *технология CORBA*

ОБЪЕКТНЫЙ МОДУЛЬ, объектная программа [object module, program object]. *Программный модуль*, являющийся результатом компиляции *исходного модуля*. О. м. представляет собой последовательность *машинных команд*, готовую к объединению с другими О. м. с помощью *редактора связей* (компоновщика). О. м. реализует машинными командами в точности те действия, которые были описаны в исходном модуле посредством операторов *исходного языка системы программирования*. Обычно О. м. выполнен в *относительных* (иногда в *абсолютных*) *адресах* и обладает свойством перемещаемости. Если О. м. не нуждается в редактировании связей и представляет законченную программу, готовую к загрузке и исполнению, то он является *загрузочным модулем*. Так как О. м. не нужна трансляция и они могут непосредственно обрабатываться редактором связей, то в целях экономии *машинного времени* и *памяти ЭВМ* многие подпрограммы и часто используемые другими программами программные модули хранятся в *библиотеках программ* в виде О. м.

ОБЪЕКТНЫЙ ЯЗЫК ОГРАНИЧЕНИЙ, язык OCL [object constraint language (OCL)]. Разработанный компанией IBM *формальный язык* для записи *ограничений*, предназначенный для использования в *объектно-ориентированных языках программирования* и *визуального моделирования*

ОБЪЕМ ПАМЯТИ [capacity]. То же, что *емкость памяти*

ОБЪЯВЛЕНИЕ [declaration]. То же, что *описание*

ОВЕРЛЕЙ [overlay]. То же, что *перекрытие*

ОВЕРЛЕЙНАЯ ПРОГРАММА, программа с оверлейной структурой, программа с перекрываемой структурой [overlay program]. *Программа*, отдельные

части которой могут попеременно по мере необходимости загружаться в одну и ту же область *оперативной памяти*. См. *перекрытие, оверлейная структура*

ОВЕРЛЕЙНАЯ СТРУКТУРА, структура с перекрытием [overlay structure]. Структура *загрузочного модуля*, в котором одна выделенная часть программы *(корневой сегмент)* постоянно находится в *оперативной памяти*, а остальные части *(сегменты перекрытий)* попеременно загружаются в ходе выполнения программы в одну и ту же область оперативной памяти, сменяя и перекрывая друг друга. На рис. О.2 приведена схема распределения памяти с использованием О. с. СЕГМ1 — корневой сегмент. СЕГМ3 полностью перекрывается сегментом СЕГМ4, а СЕГМ2 частично перекрывается либо СЕГМ3, либо СЕГМ4. Общий размер памяти, необходимый такой программе, существенно меньше суммы размеров всех четырех сегментов. Конечно, такая экономия оперативной памяти происходит за счет накладных расходов времени, включающих время загрузки сегментов СЕГМ3 и СЕГМ4 при обращении к ним во время выполнения программы, а также время, требующееся на перезагрузку СЕГМ2. О. с. не создается автоматически, программист должен ее подробно разработать средствами *системы программирования*. См. *перекрытие, оверлейная программа*

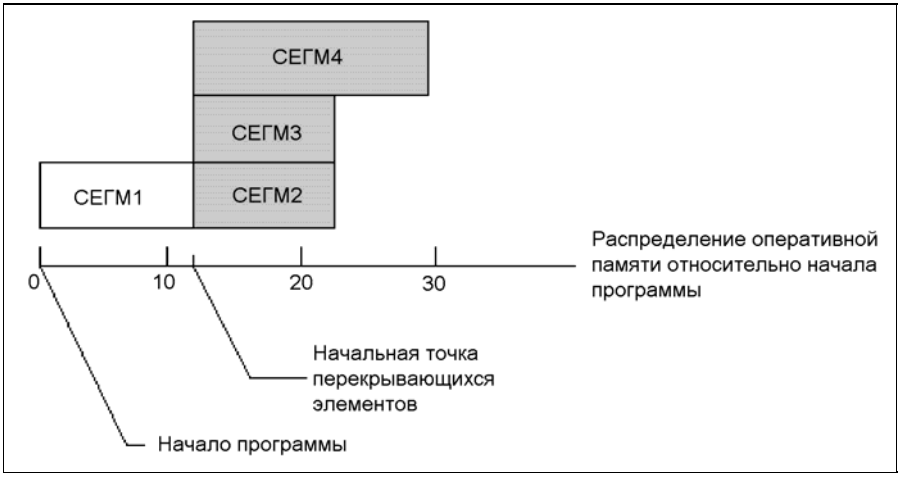

Рис. О.2. Схема распределения памяти с использованием оверлейной структуры

ОВЕРЛЕЙНОЕ РАЗМЕЩЕНИЕ МОДУЛЕЙ [overlay modules allocation]. Размещение модулей программы в одной и той же области *оперативной памяти* на различных этапах выполнения программы. Модули, содержащие подпрограммы или части данных, попеременно по мере надобности загружаются в одну и ту же область оперативной памяти. План такой загрузки и

связи между оверлейно размещаемыми модулями устанавливаются статически, *системой программирования* до начала выполнения программы. См. *перекрытие, динамическая загрузка*

ОГРАНИЧЕНИЕ [constraint]. Семантическое условие, представленное логическим выражением. В *унифицированном языке моделирования* UML для записи О. используется *объектный язык ограничений*

ОДИНОЧКА [singleton]. *Класс*, имеющий только один экземпляр. Класс-одиночка служит для представления таких общих конструкций программирования, как *глобальные переменные*, процедуры общего назначения, которые не являются методами и др., не выходя при этом за рамки *объектно-ориентированного программирования*

ОДНОЗАДАЧНАЯ ОПЕРАЦИОННАЯ СИСТЕМА [one task operating system]. Операционная система, допускающая единовременное выполнение только одной задачи

ОДНОМЕРНЫЙ МАССИВ [one-dimensional array]. *Массив* с одним индексом. См. *размерность массива*

ОДНОМЕСТНАЯ ОПЕРАЦИЯ, **унарная операция [unary operation**, **monadic operation]**. *Арифметическая* или *логическая операция*, выполняемая над одним операндом (аргументом). Например, операция логического отрицания ("НЕ"). Ср. *двуместная операция*

ОДНОРАНГОВАЯ ВЫЧИСЛИТЕЛЬНАЯ СЕТЬ [peer-to-peer network]. *Компьютерная сеть*, в которой все *узлы* равноправны, т. е. нет выделенных серверов. Например, *операционные системы Windows*, начиная с версии Windows 3.11, позволяют организовать О. в. с. *персональных компьютеров* без использования дополнительного *программного обеспечения*

ОЖИДАНИЕ [expectation]. Состояние *вычислительной системы*, при котором команды не выполняются, но могут выполняться разрешенные прерывания. Примером может служить О. *ввода данных* с клавиатуры: команды не выполняются, пока не поступит сигнал прерывания, вызванный нажатием клавиши

ОКАНТОВКА [border]. То же, что *обрамление*

ОКНО [window]. Выделенная часть экрана дисплея, с которой программа или пользователь работает как с отдельным независимым экраном, размеры и расположение которого он (пользователь) может изменять. Различают несколько типов окон: О. *приложения*, О. *документа*, *диалоговое* О. Например, в О. приложения MS Excel имеется возможность образовать несколько О. документов. Одно — для целевого документа, другие — для исходных и справочных материалов, на основании которых создается документ. На рис. О.3 представлено окно приложения MS Excel, в котором открыты два

окна документов и диалоговое окно **Найти**. Диалоговое О. служит для ввода и вывода информации. В *интерактивных вычислительных системах* в нужные моменты на экране появляются О., в которые выводится подсказка о возможных действиях пользователя. Специальные аппаратные и программные средства, называемые системой управления окнами, позволяют при помощи клавиш или мыши вызывать О. на экран и удалять с экрана, создавать и перемещать О. по экрану, изменять их размеры, переключать О. и т. п. О. на экране могут пересекаться, накладываться друг на друга. После удаления О. с экрана информация, ранее закрытая этим О., восстанавливается. Широкими возможностями управления О. обладает *операционная система Windows*. См. *модальное диалоговое окно*. Ср. *панель экрана, форма, фрейм*

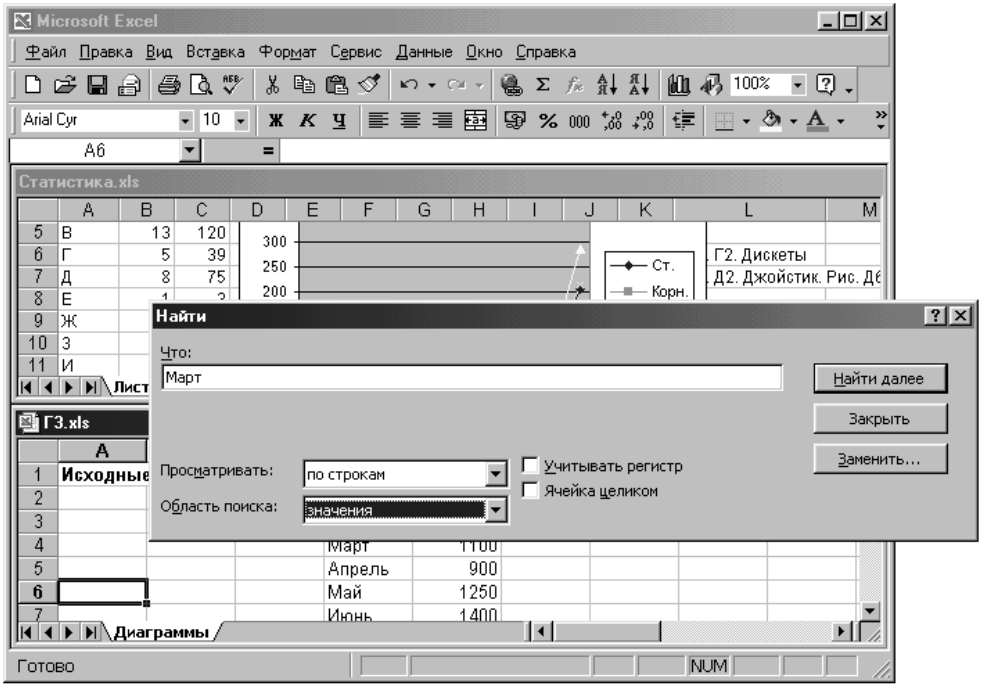

Рис. О.3. Примеры окон различных типов

ОКНО ВЫПОЛНЕНИЯ [window of execution]. Набор команд программы, из которого производится выборка команд для параллельного выполнения в данный момент. О. в. составляют команды — кандидаты на параллельное выполнение. Порядок выборки может не совпадать с порядком следования команд в программе, но результат выполнения должен совпадать с результатом строго последовательного выполнения. Чем больше О. в., тем эффек-

тивнее реализация параллельного выполнения команд. См. *параллельная обработка, параллельные вычисления, суперскалярный процессор*

ОКНО ДОКУМЕНТА [document window]. См. *окно*

ОКНО ПРИЛОЖЕНИЯ [application window]. См. *окно*

ОКНО СИСТЕМНЫХ СООБЩЕНИЙ, системное модальное диалоговое окно, системное модальное окно [system modal message box, system modal dialog box, system modal window]. *Модальное диалоговое окно операционной системы Windows*, содержащее *системное сообщение.* Появление О. с. с. препятствует выполнению любых действий пользователя до тех пор, пока окно не будет закрыто, обычно с помощью кнопок **OK** или **Отмена** (Cancel)

ОНЛАЙНОВЫЙ РЕЖИМ, режим on-line [online mode, on-line mode]. Постоянно включенный, оперативный режим работы. Например, в О. р. работает источник непрерывного питания, который фильтрует напряжение, поступающее от сети, а в случае отключения этого напряжения обеспечивает подачу питания от батареи; в О. р. работают компьютерные системы продажи и резервирования авиабилетов, оперативно реагируя на поступающие заказы и обеспечивая целостность и достоверность данных.

ООП [OOP]. То же, что *объектно-ориентированное программирование*

ОПЕРАНД [operand]. *Элемент данных*, участвующий в операции. Например, в операции вычитания участвуют два О. — уменьшаемое и вычитаемое, которые являются числами. В *машинной команде* О. могут задаваться явно, как ее составная часть, либо посредством указания адреса О. В *языках программирования* О. задаются в выражениях в виде констант, переменных, функций и т. п. Например, в арифметическом выражении `X+3.1416*SIN(Z)` О. являются переменная `X`, константа `3.1416` и функция `SIN(Z)`

ОПЕРАТИВНАЯ ПАМЯТЬ, основная память [main memory, main storage]. *Память*, непосредственно связанная с *центральным процессором вычислительной системы*. Только из О. п. вызываются в процессор подлежащие выполнению команды и операнды этих команд, и в нее же записываются результаты выполненных операций. О. п. включает в себя *постоянное запоминающее устройство*, в котором постоянно хранятся программы и данные, необходимые для работы самого компьютера. Остальные программы вместе с обрабатываемыми данными перед выполнением переписываются в О. п. из *внешней памяти. Запоминающие устройства*, из которых изготавливается О. п., должны обеспечивать высокую скорость чтения и записи, т. к. именно быстродействие О. п. ограничивает быстродействие всего компьютера. Поэтому О. п. является сравнительно дорогостоящей, и, чтобы не повышать стоимость компьютера, его память расширяют за счет менее дорогих устройств внешней памяти. В отличие от внешней памяти О. п. иногда называют внутренней

ОПЕРАТИВНАЯ ПАМЯТЬ ПЕРСОНАЛЬНЫХ IBM-СОВМЕСТИМЫХ КОМПЬЮТЕРОВ [IBM-PC main memory]. Хотя О. п. п. IBM-с. к. аппаратно однородна (за исключением ПЗУ), при работе под управлением *операционной системы MS-DOS* она может быть только разделенной на несколько неравных частей. Лишь 640 Кбайт *оперативной памяти* могут использоваться обычными программами для загрузки как самих программ, так и данных. Это т. н. *основная память.* Часть оперативной памяти от 640 Кбайт до 1 Мбайт называют *верхней памятью.* Адресное пространство сверх 1 Мбайт называют *расширенной памятью.* Использование расширенной памяти при работе в MS-DOS возможно с помощью драйвера himem.sys, но только для хранения данных, организации *электронного диска* или *кэш-памяти. Операционная система Windows* и программы, написанные для работы в среде Windows, работают в *защищенном режиме* и поэтому могут полноценно использовать всю имеющуюся оперативную память, поэтому в других ситуациях в настоящее время термин О. п. п. IBM-с. к. не применяется

ОПЕРАТИВНАЯ ПОМОЩЬ [on-line help, help]. Помощь, оказываемая пользователю системой в ходе его работы. О. п., как правило, предоставляется при нажатии *функциональной клавиши* <F1> или указании на экранную кнопку с нарисованным вопросительным знаком и имеет вид подсказки, выводимой на экран дисплея. Обычно содержание подсказок охватывает возможности системы, конструкцию операторов (управляющих команд), режимы работы, выход из ошибочных ситуаций. Ср. *справка*

ОПЕРАТОР [statement, operator]. 1. Допустимое в *языке программирования высокого уровня* предложение, задающее целостное законченное действие компьютера или представляющее набор описаний. Типичными О. в традиционных языках программирования являются О. *ввода/вывода*, *присваивания*, *перехода*, *цикла*, *процедуры* и др. Грамматическая конструкция каждого из них определяется синтаксисом конкретного *языка программирования.* О. условно делятся на *исполняемые* О. и *невыполняемые* О. Исполняемый О. представляет одну или несколько последовательных операций, составляющих алгоритм решения задачи. Невыполняемый О. непосредственно не задает алгоритм, а содержит описания *объектов программы* либо другую информацию, необходимую для трансляции и выполнения программы. 2. То же, что *оператор ЭВМ*

ОПЕРАТОР БЕЗУСЛОВНОГО ПЕРЕХОДА [imperative `go to` statement]. *Оператор перехода*, не содержащий никакого условия, а непосредственно указывающий, какой оператор должен быть выполнен следующим. В большинстве *языков программирования высокого уровня* О. б. п. состоит из слов "GO TO" (Перейти к), за которыми ставится метка. Например, в Фортране О. б. п. `GO TO 25` указывает, что следующим должен быть выполнен оператор, помеченный меткой `25`. Аналогичную форму имеет О. б. п. в Паскале и Си. *Помеченный оператор* может располагаться в программе до или после

О. б. п. При *трансляции программы* О. б. п. заменяется компилятором на *команду безусловной передачи управления*. Чрезмерное увлечение оператором `GO TO` приводит к созданию лабиринта в логике программы, делает программу трудночитаемой. Опытные программисты, пишущие программы на языках Паскаль или Си, практически не используют О. б. п. См. *операторы перехода, структурное программирование*

ОПЕРАТОР ВВОДА [input statement]. *Оператор* в программе, предписывающий *передачу данных* из *устройства ввода* или *внешней памяти* в *оперативную память* и делающий эти данные доступными программе. В *языке программирования* может быть несколько конструкций О. в. Вид этих операторов, а также информация, которая должна быть указана транслятору этими операторами, различны и определяются синтаксисом конкретного *языка программирования* и особенностями *внешних запоминающих устройств* и устройств ввода. Поэтому в некоторых *языках программирования высокого уровня* (например, в Си) О. в. не входят в состав основных определений языка. Их разработка возложена на программистов, проектирующих компиляторы *систем программирования*. Однако для обеспечения *переносимости программ* в различных *системах программирования*, ориентированных на один язык, применяется единый стандартный набор процедур или функций ввода. Обычно в О. в. задается список ввода — перечень переменных и массивов, значения которым должны быть присвоены в процессе *ввода данных*, и номер устройства или имя файла, из которого данные читаются. Иногда задается формат вводимых данных. Ниже приводятся примеры написания простейшего О. в. значений двух переменных `m` и `r` (`m` — *целого типа*, `r` — *вещественного типа*) на языках Фортран, Паскаль и Си. 1) На Фортране: `READ * m, r`. 2) На Паскале: `Read (m, r);`. Здесь тип переменных и формат не указан и определяется либо по умолчанию, либо заранее объявлен в программе. 3) На Си: `scanf(" %d %f", &m, &r);`. Здесь в кавычках задан формат вводимых данных: `%d` — десятичное целое число, `%f` — десятичное число с плавающей точкой

ОПЕРАТОР ВОЗВРАТА [return statement]. *Оператор*, завершающий выполнение процедуры или функции и передающий управление *вызывающей программе*. Как правило, О. в. обозначается ключевым словом `Return` (возврат) и располагается в *теле процедуры* или *функции*. Обычно О. в. передает управление либо оператору вызывающей программы, в котором содержался *вызов функции* (в случае функции), либо оператору, следующему за *оператором вызова процедуры* (в случае процедуры). В некоторых *языках программирования* (например, в Фортране) существует конструкция О. в., позволяющая передавать управление любому *помеченному оператору* вызывающей программы. Для этого его метка должна быть указана как *фактический параметр* в операторе вызова подпрограммы. В некоторых языках программирования (например, в Си) с помощью О. в. имени функции может прис-

ваиваться значение, возвращаемое в вызывающую программу как результат вычисления функции

ОПЕРАТОР ВЫБОРА, переключатель [case statement]. *Оператор* в программе, определяющий выбор одной из нескольких *ветвей алгоритма*. Например, с помощью О. в. программируется меню (выбор пользователем последовательности действий в соответствии с *пунктом меню*). О. в. содержит перечень альтернативных операторов, в которых определены действия, соответствующие различным ситуациям или условиям. Выбор необходимого варианта осуществляется с помощью указанного в О. в. выражения, обычно называемого *переключателем*. Выражение-переключатель конструируется так, чтобы оно принимало значение, определяющее номер или метку оператора, который нужно выполнить в создавшейся ситуации. Примерами О. в. являются оператор `case` в Паскале и оператор `switch` в языке Си. Оператор `case` имеет вид:

```
case n of
    n1: S1;
    n2: S2;
    ......
    nm: Sm
    otherwise SS
end;
```

где `n` — выражение (переключатель) *целого*, *символьного* или *перечисляемого типа*; `n1, n2, ..., nm` — константы того же типа, что и переключатель; `S1, S2, ..., Sm, SS` — операторы. Каждая константа сопоставляется с определенным оператором. Для выполнения выбирается тот оператор, с константой которого совпадает текущее значение переключателя. Если среди `n1, n2, ..., nm` нет константы, соответствующей текущему значению переключателя, то управление передается оператору после ключевого слова `otherwise` (если он присутствует). После выполнения выбранного оператора управление передается в конец конструкции `case`. Похожим образом работает оператор `switch`, общий вид которого:

```
switch (n)
{
    case m1: S1
       break;
    case m2: S2
       break;
    ......
    case mk: Sk
```

```
        break;
    default: SS
}
```

где n — выражение (переключатель) целого или символьного типа; m1, m2, ..., mk — метки того же типа, что и переключатель; S1, S2, ..., Sk, SS — операторы. Управление передается оператору, метка которого совпадает с текущим значением выражения n. Оператор break приводит к выходу из оператора switch и передает управление следующему за ним оператору. Операторы break могут отсутствовать. Если за оператором Si нет оператора break, то управление переходит к очередному оператору конструкции case и т. д., пока не будут выполнены оставшиеся операторы или не встретится оператор break. Если некоторому значению переключателя не соответствует никакая метка, управление передается оператору с меткой default (если такой имеется)

ОПЕРАТОР ВЫВОДА [output statement]. *Оператор* в программе, предписывающий *передачу данных* из *оперативной памяти* во *внешнюю память* или на *устройства вывода*. В *языке программирования* может быть несколько конструкций О. в. Вид этих операторов, а также информация, которая должна быть указана транслятору этими операторами, различны и определяются *синтаксисом* конкретного *языка программирования* и особенностями *внешних запоминающих устройств* и устройств вывода. Поэтому в некоторых *языках программирования высокого уровня* (например, в Си) О. в. не входят в состав основных определений языка. Их разработка возложена на программистов, проектирующих компиляторы *систем программирования*. Однако для обеспечения *переносимости программ* в различных *системах программирования*, ориентированных на один и тот же язык, применяется единый стандартный набор процедур или функций вывода. Обычно в О. в. задается список вывода — перечень переменных и массивов, значения которых должны быть выведены, имя или номер устройства либо имя файла, в который записываются данные. Кроме того, указывается формат выводимых данных. Ниже приводятся примеры написания простейшего О. в. значений двух переменных m и r (m — целого типа, r — вещественного типа) на языках Фортран, Паскаль и Си. 1) На Фортране: PRINT 10, m, r. Здесь метка 10 указывает на оператор задания формата выводимых величин, имеющий, например, вид: 10 FORMAT('m = ',I3,' r = ',F7.2), согласно которому при значениях m и r, равных, соответственно, 123 и -45.6, на экране появится строчка: m = 123 r = -45.60. 2) На Паскале: Write (m, r);. Тип переменных и формат не указан и определяется либо по умолчанию, либо заранее объявлен в программе. 3) На Си: printf(" n = %d r = %f \n", m, r);. Здесь в кавычках задан формат строки выводимых данных: %d — десятичное целое число, %f — десятичное число с плавающей точкой, \n — символ, предписывающий переход к следующей строке

ОПЕРАТОР ВЫЗОВА ПОДПРОГРАММЫ, оператор обращения к подпрограмме [call(ing) statement]. В *языках программирования оператор* `CALL`, задающий *вызов подпрограммы*: передачу подпрограмме *фактических параметров*, *передачу управления* на *вход в подпрограмму*, выполнение действий, запрограммированных в подпрограмме, и возврат управления *вызывающей программе*. Например, в языке Фортран О. в. п. имеет вид: `CALL name (x1, x2, ..., xn)` или `CALL name`, где `name` — имя вызываемой подпрограммы или идентификатор дополнительной точки входа в подпрограмму; `x1, x2, ..., xn` — фактические параметры, передаваемые подпрограмме, согласующиеся по количеству, порядку следования, классу и типу с соответствующими *формальными параметрами*. Ср. *оператор процедуры, вызов функции*

ОПЕРАТОР ВЫЗОВА ПРОЦЕДУРЫ, оператор обращения к процедуре [calling procedure statement]. То же, что *оператор процедуры*

ОПЕРАТОР ЗАДАНИЯ ФОРМАТА, оператор формата [format statement]. *Оператор*, задающий *формат* вводимых и выводимых данных. Он используется при работе *операторов ввода* и *вывода* и описывает форму *представления данных* и их расположение в записи ввода/вывода. Связь оператора ввода (или вывода) с О. з. ф. обычно осуществляется с помощью метки, записанной как в операторе ввода (или вывода), так и перед О. з. ф. Работу О. з. ф. можно объяснить только во взаимодействии с соответствующим ему оператором ввода или вывода. Например, в памяти компьютера хранятся значения переменных `K = 15` и `R = -123.45678`. Вывод этих величин на экран запрограммирован на языке Фортран оператором вывода и О. з. ф.

```
      PRINT 10, K, R
10    FORMAT('K =',I3,'R =',F7.2)
```

Эти операторы связаны между собой меткой `10`. В операторе `FORMAT` содержатся разделенные запятыми спецификации формата, предписывающие оператору `PRINT` вывести в текущую строку экрана: 1) в начале символы `K =` (*символьная константа* `'K ='`); 2) затем значение переменной `K` в форме *целого числа*, для которого на экране отводятся три знакоместа (спецификация `I3`); 3) снова символы `R =`, в том числе пробелы перед `R` и после `=` (символьная константа `'R ='`); 4) наконец, значение переменной `R` в форме *числа с фиксированной точкой*, занимающего семь знакомест экрана, так, чтобы дробная часть содержала две цифры (спецификация `F7.2`). В результате на экране должна появиться запись: `K = 25 R = -123.46`

ОПЕРАТОР ОБРАЩЕНИЯ К ПРОЦЕДУРЕ [procedure statement]. То же, что *оператор процедуры*

ОПЕРАТОР ПЕРЕХОДА [GO TO statement]. См. *операторы перехода*

ОПЕРАТОР ПРИСВАИВАНИЯ [assignment statement]. *Оператор*, передающий значение *арифметического*, *логического* или другого *выражения* одной или нескольким переменным либо имени функции. Действие присваивания обычно обозначается символами: = (например, в языках Фортран и Си) и := (в языке Паскаль). Слева от символа присваивания пишется имя переменной (или имена переменных), которой должно быть присвоено значение выражения, стоящего справа. При выполнении О. п. вначале вычисляется значение выражения, а затем оно передается переменной. Например, О. п. `k = k + 1` предписывает вычислить значение выражения `k + 1`, используя текущее значение переменной `k`; затем текущее значение `k` заменить значением только что вычисленного выражения. Короче говоря, увеличить `k` на единицу. Тип переменной, стоящей слева в О. п., может не совпадать с типом выражения, стоящего справа. Соответствующее *преобразование типов* обычно автоматически закладывается в *объектную программу* при компиляции. Однако во избежание возможной потери информации при преобразовании типов желательно, чтобы тип переменной и тип выражения были одинаковыми. В языке Си возможно присваивание одинакового значения нескольким переменным с помощью одного О. п. Например, `x = y = 0.0;`. В этом случае присваивание выполняется справа налево: значение `0.0` сначала получит переменная `y`, а затем переменная `x`. В Фортране и Паскале такое двойное присваивание запрещено. Кроме того, в языке Си введены дополнительные операции присваивания, обозначаемые символами: +=, —=, *=, /= и %=. При выполнении О. п. с такими символами переменной, стоящей слева от символа, присваивается новое значение, равное ее старому значению, скорректированному с помощью величины выражения, стоящего справа. Результат зависит от используемой операции. Пусть `x` — переменная, а `y` — выражение, тогда:

`x += y;` — то же самое, что `x = x + y;`

`x —= y;` — то же самое, что `x = x — y;`

`x *= y;` — то же самое, что `x = x * y;`

`x /= y;` — то же самое, что `x = x / y;`

`x %= y;` — то же самое, что `x = x % y;`

ОПЕРАТОР ПРОДОЛЖЕНИЯ [CONTINUE statement]. В языке Фортран *оператор* вида: `m CONTINUE`, где `m` — метка. О. п. не вызывает никаких действий, а используется для размещения метки. Обычно применяется в качестве последнего оператора *тела цикла*, чтобы избежать окончания цикла операторами, которые не могут быть последним *помеченным оператором* цикла (например, операторы `GO TO`, `RETURN`, `PAUSE`, `STOP` и др.). В языке Си оператор `CONTINUE` вызывает возобновление выполнения тела цикла с начала, т. е.

операторы, находящиеся между оператором `CONTINUE` и концом цикла, не выполняются

ОПЕРАТОР ПРОЦЕДУРЫ, оператор вызова процедуры, вызов процедуры [procedure statement]. В *языках программирования — оператор*, вызывающий процедуру. Выполнение О. п. эквивалентно такой последовательности действий: *передача* процедуре *фактических параметров*, *передача управления* на *вход в подпрограмму*, выполнение операторов, запрограммированных в *теле процедуры*, и возврат управления *вызывающей программе*. Правила записи О. п. определяются синтаксисом конкретного языка программирования. Наиболее распространенный вид О. п.: `Name (x1, x2, ..., xn)` или `Name`, где `Name` — имя вызываемой процедуры или идентификатор дополнительной *точки входа* в процедуру; `x1, x2, ..., xn` — фактические параметры, передаваемые процедуре, согласующиеся по количеству, порядку следования, классу и типу с соответствующими *формальными параметрами*. См. *оператор вызова подпрограммы, вызов процедуры*. Ср. *вызов функции*

ОПЕРАТОР УСЛОВНОГО ПЕРЕХОДА [conditional GO TO statement]. *Оператор перехода*, содержащий языковую конструкцию, реализующую проверку условия, при котором изменяется естественная последовательность выполнения операторов в программе. О. у. п. применяется при *ветвлении программы*, когда возникает необходимость осуществить переход в зависимости от результата проверки некоторого условия. Ниже приводятся примеры на языках Фортран, Паскаль и Си простейшего О. у. п., указывающего, что в случае, когда значение переменной `x` отрицательно, следующим должен выполняться оператор, помеченный меткой `m1` (или меткой `25` в примере для Фортрана), при $x \geq 0$ должен выполняться оператор, стоящий в программе сразу за О. у. п. 1) На Фортране: `IF (X<0) GO TO 25`. 2) На Паскале: оператор `if x<0 then go to m1;`. 3) На Си: `if (x<0) goto m1;`. О. у. п. транслируются компилятором с помощью *команд передачи управления*. См. *операторы перехода, арифметический условный оператор*

ОПЕРАТОР ФОРМАТА [format statement]. То же, что *оператор задания формата*

ОПЕРАТОР-ФУНКЦИЯ [statement function]. В языке Фортран *оператор*, имеющий вид *оператора присваивания* и определяющий внутри *программного модуля* функцию, заданную выражением. К этой функции можно обращаться по имени так же, как к *стандартным функциям*, если необходимо многократно в разных точках программного модуля вычислять значение этой функции при различных значениях ее аргументов. О.-ф. располагается в программном модуле перед первым *выполняемым оператором*, но после всех других *невыполняемых операторов*. Например, если в программе необходимо вычислять расстояния на плоскости от разных точек до разных прямых, то

удобно один раз перед первым выполняемым оператором программы написать О.-ф.:

```
D= ABS(Y-A*X-B)/SQRT(A*A+1.)
```

реализующую алгоритм вычисления расстояния от точки с координатами (x, y) до прямой, заданной уравнением $y = Ax + B$, а затем в нужных местах программы писать обращения к функции `D`, указывая вместо *формальных параметров* `X`, `Y`, `A` и `B` фактические значения аргументов или имена переменных, содержащих эти значения. Так, оператор `R = D(0, 0, 1.5, -3.5)` вычислит расстояние от начала координат до прямой $y = 1.5\,x - 3.5$ и присвоит это значение переменной `R`

ОПЕРАТОР ЦИКЛА [cycle statement]. *Оператор*, упрощающий программирование *цикла*. О. ц. формально состоит из *заголовка цикла* и *тела цикла*. Заголовок цикла указывает на последовательность повторяемых операторов, образующих тело цикла, и либо определяет множество значений *параметра цикла* и предписывает многократное выполнение тела цикла при этих значениях параметра, либо указывает условие повторного выполнения тела цикла. Правила написания О. ц. задаются синтаксисом конкретного *языка программирования* и типом программируемого цикла. См. *цикл, операторы цикла языка Паскаль, операторы цикла языка Си, оператор цикла языка Фортран*

ОПЕРАТОР ЦИКЛА ЯЗЫКА ФОРТРАН [Fortran loop statement, DO loop]. Для программирования циклов в языке Фортран предусмотрен *оператор цикла*, имеющий форму:

```
      DO m I = n1, n2, n3
      <оператор>
      . . . . .
m     <оператор>
```

Здесь `m` — метка последнего *исполняемого оператора* в *теле цикла*; `I` — управляющая переменная *целого типа (параметр цикла)*; `n1` — начальное значение параметра цикла; `n2` — конечное значение параметра цикла; `n3` — шаг изменения параметра цикла. `n1, n2, n3` — целые положительные константы или переменные целого типа, принимающие положительные значения; `n3` может отсутствовать, при этом подразумевается `n3 = 1`. Цикл `DO` является *циклом с постусловием*. При первом выполнении тела цикла `I = n1`. Затем значение параметра цикла увеличивается на шаг `n3` (или 1) и проверяется условие `I ≤ n2`. Пока оно истинно, выполняется тело цикла. Число выполнений тела цикла равно целой части от `(n2 – n1)/n3 + 1`, если `n2 > n1`, или 1, если `n2 < n1` (при этом `I = n1`). Оператор с меткой `m` не должен быть оператором цикла, оператором перехода, операторами `PAUSE` и `STOP`. Для обхода этого ограничения цикл заканчивают *оператором продолжения* `CONTINUE`.

Например,

```
     DO 10 I = 1, N
     IF (A[I]<AMIN) GOTO 20
10   CONTINUE
```

См. *вложенный цикл*

ОПЕРАТОР ЭВМ [operator]. Специалист, контролирующий *ЭВМ* и управляющий ее работой и работой ее отдельных устройств. В настоящее время О. ЭВМ работают в *вычислительных центрах* на крупных вычислительных комплексах или обслуживают *компьютерные сети*

ОПЕРАТОРНЫЕ СКОБКИ [statement brackets]. В *языках программирования — ключевые слова* `begin` и `end` или другие символы (например, фигурные скобки в языке Си), ограничивающие последовательность операторов программы. О. с. ставятся в целях объединения нескольких операторов в *составной оператор*, представляющий единую *синтаксическую конструкцию.* Например, в операторе цикла на языке Паскаль

```
for i:= 1 to 10 do begin x[i]:= cos(Alfa*i); y[i]:= sin(Alfa*i) end;
```

О. с. объединяют два оператора в повторяемый составной оператор *(тело цикла).* Без О. с. повторялся бы только оператор `x[i]:= cos(Alfa*i);`

ОПЕРАТОРЫ ПЕРЕХОДА [GO TO statements]. *Операторы*, изменяющие естественную последовательность действий в программе, указывая оператор, который должен быть выполнен следующим. В большинстве *языков программирования высокого уровня* О. п. содержит строку "`GO TO`" (Перейти к). Существуют *операторы безусловного перехода*, предписывающие переход в заданную точку программы без проверки выполнения каких-либо условий. При *ветвлении программы* возникает необходимость осуществить переход в зависимости от результата проверки некоторого условия. В этом случае могут применяться *операторы условного перехода* или другие операторы, явно или неявно включающие в себя языковую конструкцию, реализующую проверку условия. Например, в Паскале оператор `if x<0 then go to 25;` указывает, что в случае, когда значение переменной `x` отрицательно, следующим должен выполняться оператор, помеченный меткой `25`. Аналогичный оператор в Фортране имеет вид: `IF (X<0) GO TO 25`, и в Си он выглядит так же: `if (x<0) goto m25;`. Кроме того, в Фортране имеются *арифметический условный оператор*, определяющий переход в зависимости от значения, указанного в операторе *арифметического выражения*, и *вычисляемый оператор перехода*, удобный в случае, когда из нескольких расчетных формул необходимо выбрать одну. О. п. транслируются компилятором в виде одной или нескольких *команд передачи управления*

ОПЕРАТОРЫ ЦИКЛА ЯЗЫКА ПАСКАЛЬ, **предложения цикла языка Паскаль [Pascal loop statements, Pascal loop sentences]**. Для программирования *циклов* в языке Паскаль предусмотрены следующие основные *синтаксические конструкции* предложений:

Цикл с предусловием, общий вид которого:

```
while <логическое выражение> do <тело цикла>
```

Тело цикла будет повторно выполняться до тех пор, пока значение логического выражения не станет равным `false`. Если это выражение равно `false` при первой же проверке, то тело цикла не выполняется вообще. Телом цикла может служить одиночный оператор или несколько последовательных операторов, заключенных в операторные скобки `begin` и `end` *(составной оператор)*. Например, `while Abs(dx) > eps do begin dx:= (a / x – x) * 0.5; x:= x + dx end;`

Цикл с постусловием, общий вид которого:

```
repeat <тело цикла>until <логическое выражение>
```

Тело цикла будет повторно выполняться до тех пор, пока логическое выражение не примет значение `true`. Так как условие проверяется в конце тела цикла, само тело будет выполнено хотя бы один раз. Телом цикла может служить одиночный или составной оператор. Например,

```
repeat dx:= (a/x – x) * 0.5; x:= x + dx until Abs(dx) < eps;
```

Цикл с параметром, задающий повторные выполнения тела цикла до тех пор, пока *параметр (управляющая переменная) цикла* принимает ряд значений, образующих возрастающую либо убывающую арифметическую прогрессию. Соответственно, общий вид цикла имеет две формы:

```
for <управляющая переменная> := <начальное выражение> to
 <конечное выражение> do <тело цикла>;
```

или

```
for <управляющая переменная> := <начальное выражение> downto
 <конечное выражение> do <тело цикла>;
```

Начальное и конечное выражения являются выражениями *порядкового типа*. Управляющая переменная должна иметь тип, совпадающий или совместимый с типом выражений для начального и конечного значений. Если значение начального выражения больше значения конечного выражения для цикла `to` или, наоборот, значение начального выражения меньше значения конечного выражения для цикла `downto`, то тело цикла не выполняется ни разу, а управление передается следующему оператору. Ни один оператор в теле цикла не должен менять значение управляющей переменной.

Например,

```
for i:= 1 to 10 do begin x[i]:= cos(Alfa*i); y[i]:= sin(Alfa*i) end;
for l:= 10 downto k+1 do x[l]:= x[l] – x[k];
```

См. *вложенный цикл*

ОПЕРАТОРЫ ЦИКЛА ЯЗЫКА СИ, предложения цикла языка Си [C loop statements, C loop sentences]. Для программирования циклов в языке Си предусмотрены следующие основные конструкции предложений:

Цикл с предусловием, общий вид которого:

```
while (<выражение>) <тело цикла>
```

Тело цикла будет повторно выполняться до тех пор, пока выражение не станет ложным или равным нулю. Если это выражение ложно или равно нулю при первой же проверке, то тело цикла не выполняется вообще. Телом цикла может служить одиночный оператор или несколько последовательных операторов, заключенных в фигурные скобки `{` и `}` *(составной оператор)*. Например,

```
while (summa<1000) {summa = summa + step; step = 2 * step};
```

Цикл с постусловием, общий вид которого:

```
do <тело цикла> while (<выражение>);
```

Тело цикла будет повторно выполняться до тех пор, пока логическое выражение не станет ложным или равным нулю. Так как условие проверяется в конце тела цикла, само тело будет выполнено хотя бы один раз. Телом цикла может служить одиночный или составной оператор. Например,

```
do {ch:= getchar(); putchar(ch);} while(ch!= '\n');
```

Цикл с предусловием, который удобно применять, как *цикл с параметром*. Общий вид цикла имеет форму:

```
for ( <инициализирующее выражение>; <проверяемое выражение>;
 <корректирующее выражение> ) <тело цикла>
```

В операторе `for` используются три выражения, разделенные символом `;` (точка с запятой). Инициализирующее выражение вычисляется только один раз до начала выполнения какого-нибудь из операторов цикла. Если проверяемое выражение оказывается истинным или не равным нулю, тело цикла выполняется один раз. Затем вычисляется корректирующее выражение и снова вычисляется проверяемое выражение. Таким образом, тело цикла выполняется до тех пор, пока проверяемое выражение не станет ложным или равным нулю. Так как решение об очередной итерации цикла принимается до начала его прохождения, то может случиться, что тело цикла не выпол-

нится ни разу. Телом цикла может служить одиночный или составной оператор. Например,

```
for ( i = 1; i < = 10; i++) { c = a[i]; a[i] = b[i]; b[i] = c}
```

См. *вложенный цикл*

ОПЕРАЦИОННАЯ СИСТЕМА (ОС) [operating system (OS)]. *Комплекс программ*, организующих вычислительный процесс в *вычислительной системе*. Основными функциями ОС являются *распределение ресурсов* вычислительной системы между задачами с целью их наиболее эффективного использования и облегчение работы пользователя с вычислительной системой. Реализуя первую функцию, ОС учитывает и распределяет ресурсы, управляет *центральным процессором*, памятью, вводом/выводом, обеспечивает выполнение операций над файлами, выступает в роли диспетчера, запуская на выполнение *прикладные программы*, обеспечивает взаимодействие программ с *техническими устройствами* и пользователем. Выполняя вторую функцию, она предоставляет пользователю удобный интерфейс с *программным обеспечением* и устройствами компьютера, а также производит вспомогательные действия, такие как копирование или печать файлов. ОС может поддерживать в каждый момент работу на компьютере одной задачи — быть *однозадачной* — или одновременно работу нескольких программ — быть *многозадачной* (мультизадачной). Пользователь управляет ОС с помощью *команд операционной системы*. Для управления работой *персональных компьютеров* чаще всего применяются ОС *Windows*, Linux или UNIX различных версий и в различных конфигурациях

ОПЕРАЦИОННАЯ СИСТЕМА LINUX (ОС LINUX), LINUX [operating system Linux (OS Linux), Linux]. Многопользовательская многозадачная *операционная система*, представляющая собой полнофункциональную версию *операционной системы UNIX*, адаптированную для работы на многих платформах *персонального компьютера*. В Linux содержится набор инструментов, с помощью которых можно создавать приложения, документы, веб-страницы, презентации, чертежи и т. п. Кроме того, ОС Linux позволяет создавать интернет-узлы и FTP-серверы, а также поддерживает многопроцессорную архитектуру компьютера. Начало разработки Linux было положено Линусом Торвальдсом в 1991 г., а затем Linux развивался благодаря Интернету с помощью трудов многих энтузиастов. ОС Linux является некоммерческим продуктом и распространяется бесплатно и доступно в исходных текстах. Любая ее копия считается легальной. Вокруг Linux собирается интегрированная среда, состоящая из UNIX-совместимых приложений, разрабатываемых под эгидой *Фонда свободного программного обеспечения* в рамках *проекта GNU*

ОПЕРАЦИОННАЯ СИСТЕМА MS-DOS (ОС MS-DOS), MS-DOS [operating system MS-DOS (OS MS-DOS)]. Разработанная фирмой Microsoft

операционная система персональных IBM-совместимых компьютеров. За время активного использования этой ОС фирма выпустила несколько версий: 3.30, 4.01, 5.0 и 6.0, меняя которые, опытные пользователи IBM PC каждый раз получали заметные преимущества. MS-DOS 6.0 обеспечивает: автоматический запуск с первоначальным самотестированием *аппаратных* и *программных средств*, режим многовариантной *конфигурации компьютера*, диалог с пользователем, управление процессом выполнения операций, управление *внешними устройствами*, *обработку запросов* ввода/вывода, буферизацию, *блокировку данных*, планирование и *обработку прерываний*, организацию *разделов памяти*, сегментирование, динамическое и оверлейное размещение модулей, доступ к верхней и дополнительной памяти, диспетчеризацию очередей и многое другое. В связи с резко возросшими возможностями *персональных компьютеров* и широким распространением *графического интерфейса пользователя* MS-DOS применяется значительно реже, чем в предыдущие годы и пользователи MS-DOS в большинстве переходят на платформу Windows

ОПЕРАЦИОННАЯ СИСТЕМА NETWARE [operating system NetWare, NetWare, Novel NetWare]. Предназначенная для *локальных вычислительных сетей многозадачная операционная система*, созданная фирмой Novell. О. с. NetWare дает пользователям возможность совместно использовать файлы и ресурсы сети, например, жесткие диски, принтеры и др. Имеется семейство О. с. NetWare, распространенных на многих компьютерных платформах

ОПЕРАЦИОННАЯ СИСТЕМА UNIX (ОС UNIX), UNIX [operating system (OS UNIX), UNIX]. Интегрированная многозадачная *операционная система*, предоставляющая пользователю удобный язык команд и *диалоговый интерфейс.* Первоначально разработана в начале 1970-х гг. специалистами фирмы Bell Laboratories для ЭВМ серии PDP-11. В дальнейшем получила широкое распространение на компьютерах различных классов и типов. Выделенная мобильная основная часть ОС UNIX, написанная на языке Си, обеспечивает ее переносимость и *переносимость программ*, разработанных под UNIX, на машины с различными *системами команд.* В ОС UNIX впервые реализованы три основные концепции операционных систем: *файловой системы*, процесса и оболочки. *Файловая система* хранения программ, данных и текстов обеспечивает единый интерфейс всех видов ввода/вывода, а также средства защиты от *аппаратных сбоев* и *несанкционированного доступа.* Процесс, который ОС рассматривает как единое целое при распределении ресурсов, характеризуется последовательностью операций при выполнении программы, используемыми этими операциями данными, открытыми файлами, *текущим каталогом* и т. п. Оболочка имеет удобные средства диалога пользователя с системой, включая язык (SHELL, CSHELL), позволяющий создавать новые команды, изменяя и подстраивая *пользовательский интерфейс* по желанию пользователя. Кроме того, ОС UNIX предоставляет ряд средств: *разработки программ* на различных *языках программирования*, *машинной графики*, под-

держки *компьютерных сетей* и т. д. Важным свойством системы является ее открытость для расширения. Все это позволяет назвать ОС UNIX *интегрированной средой*, которая постоянно развивается и совершенствуется. ОС UNIX сыграла важную роль в создании и развитии сети Интернет. Для *персональных компьютеров* есть разные версии ОС UNIX. Наиболее распространенные из них — BSD, System V и Linux

ОПЕРАЦИОННАЯ СИСТЕМА WINDOWS (ОС WINDOWS) [operating system Windows (OS Windows)]. Собирательное название набора *операционных систем*, разработанных фирмой Microsoft для *персональных компьютеров.* Все ОС W. обеспечивают основные функции управления *вычислительной системой* и взаимодействия с пользователем. В настоящее время существует несколько вариантов ОС W., отличающихся по архитектуре и возможностям, а для каждого варианта выпущено несколько версий. Ниже приведен краткий обзор наиболее популярных ОС W. 1) Повсеместное признание ОС W. получила с появлением версий Windows 3.*x*, которые представляют собой 16-разрядные операционные системы, предназначенные для управления автономными персональными компьютерами и рабочими станциями в *локальной сети*. Главным достоинством этих операционных систем явилось использование *графического интерфейса пользователя*, что и обусловило их широкое распространение. 2) Появившаяся на рынке в 1995 г. Windows 95 представляет собой интегрированную 32-разрядную многозадачную операционную систему общего назначения. Она позволяет создать *информационную среду* с удобным, ориентированным на *рабочий стол дружественным интерфейсом.* Включает в себя средства мультимедиа и 32-разрядную *файловую систему* CD-ROM, обеспечивающие высококачественное воспроизведение цифровой видеозаписи, средства *компьютерной связи*, упрощающие подключение к Интернету, встроенные возможности приема и передачи факса и *электронной почты*. Кроме того, Windows 95 позволяет создать свою собственную *локальную вычислительную сеть.* Windows 98 — следующая после Windows 95 операционная система. По сравнению с Windows 95 она обеспечивает: ускоренную загрузку программ; интеграцию *браузера* Интернета в систему; *активный рабочий стол*; возможность самообновления; поддержку нескольких мониторов и до 127 внешних устройств; возможность установить расписание *чистки памяти.* При этом увеличивается место, занимаемое системой на жестком диске (до 295 Мбайт). 3) Параллельно с разработкой Windows 9*x* корпорация Microsoft создала семейство операционных систем Windows NT (New Technology). Существуют варианты, предназначенные для управления *рабочей станцией* (Windows NT Workstation), и варианты, предназначенные для управления *сервером* локальной сети (Windows NT Server). Эти 32-разрядные операционные системы отличает особо высокая надежность, устойчивость, безопасность и производительность, однако их требования к ресурсам вычислительной системы несколько выше, чем у Windows 9*x*. 4) В настоящее время корпорация Microsoft активно продвигает на рынок семейства

Windows 2000 и Windows XP, в которые входят несколько вариантов операционной системы, предназначенных для управления как рабочими станциями, так и серверами. Windows 2000 сохраняет преемственность с Windows 9*x* по принципам организации пользовательского интерфейса, выдвигает приемлемые требования к ресурсам вычислительной системы и в то же время использует многие хорошо зарекомендовавшие себя технологии семейства Windows NT. 5) Семейство Windows CE специально предназначено для управления *встроенными системами*, *организаторами персональной деятельности* и другими специализированными компьютерами с ограниченными ресурсами. Интерфейс этих операционных систем подобен интерфейсу других ОС W., но архитектура существенно отличается. Например, Windows CE может работать на компьютере, на котором отсутствуют *клавиатура* или *жесткий диск*

ОПЕРАЦИОННАЯ СРЕДА [operational environment]. *Операционная система* и ее *приложения*, которыми пользуется и во взаимодействии с которыми выполняется (работает) программа. О. с. накладывает свой отпечаток как на саму программу, так и на характер и качество ее работы. Например, поскольку текстовый редактор Microsoft Word для Windows работает в О. с. Windows, он обладает всеми преимуществами ее графического интерфейса

ОПЕРАЦИЯ [operation]. 1. Действие, выполняемое над данными. 2. Последовательность элементарных действий различных устройств компьютера, реализующая выполнение команды. См. *машинная операция* 3. Спецификация трансформации или запроса, которые должен выполнить вызываемый объект. О. имеет имя и список параметров. *Метод класса* — это процедура, которая реализует О. класса

ОПЕРАЦИЯ ОТНОШЕНИЯ [comparison operation]. То же, что *операция сравнения*

ОПЕРАЦИЯ СРАВНЕНИЯ, операция отношения [comparison operation]. *Операция*, позволяющая сравнить по величине или по значению два операнда. О. с. является *двуместной операцией*, которая проверяет, выполняется ли для данных операндов указанное в операции отношение. В набор О. с. входят: "равно" (обозначается знаком =), "больше" (>), "меньше" (<), "больше или равно" (≥ либо >=), "меньше или равно" (≤ либо <=), "не равно" (≠ либо <>, либо !=). Результатом О. с. может быть одно из двух значений: "да" или "нет", "ИСТИНА" или "ЛОЖЬ", 1 или 0. В *языках программирования высокого уровня* О. с. задаются с помощью языковых конструкций, называемых отношениями. О. с. могут производиться над операндами, имеющими численное значение, операндами, являющимися битовыми строками или кодами. См. *отношение*

ОПИСАНИЕ, объявление [declaration, specification]. Предложение или более сложная конструкция *языка программирования*, определяющая имя и основ-

ные характеристики переменных, массивов, меток, переключателей, файлов данных, подпрограмм и других объектов или элементов программы. В О. указывается имя объекта, его тип, атрибуты и *область определения*. В О. *процедуры* или *функции*, кроме того, задается реализуемый ими алгоритм. Информация, содержащаяся в О., необходима транслятору для *распределения памяти* и правильной *трансляции программы*. В тексте программы объект должен быть описан прежде, чем его имя встретится в каком-либо предложении, поэтому обычно О. располагаются в начале программы или *блока программы*. Правила О. различных объектов определяются синтаксисом конкретного языка. В некоторых языках О. в блоке должны следовать в строго определенной последовательности. Например, в Паскале принята следующая последовательность: О. *меток*, О. *констант*, О. *типов*, О. *переменных*, О. *процедур и функций*. См. *неявное описание*

ОПИСАНИЕ КЛАССА [class declaration]. Описание, в котором указываются имя класса и, если нужно, *суперкласс*, а также описания свойств класса, которые обычно синтаксически выглядят как *описания переменных*, и описания методов класса, которые обычно синтаксически выглядят как *описания функций*. См. пример О. к. в статье *класс*

ОПИСАНИЕ МАССИВА [array declaration]. *Описание*, в котором указываются имя массива, тип образующих массив *элементов данных*, *размерность массива* и границы изменения индексов. Например, на языке Паскаль описание

```
var A: array [1..20, 1..20] of real
```

характеризует массив `A` как двумерный, состоящий из 400 вещественных чисел. Причем, оба индекса, определяющие *элементы массива*, могут изменяться от 1 до 20

ОПИСАНИЕ МЕТОК [label declaration]. *Описание*, в котором указывается, какие идентификаторы являются метками, и задается их *область определения*. Любой оператор программы можно пометить, поставив перед ним метку, на которую можно ссылаться, например, в *операторе перехода*. В ряде *языков программирования*, например, в Фортране, не требуется специального О. м. Метка считается описанной в том *блоке программы*, где она стоит. Однако в некоторых языках каждую метку требуется описать в разделе О. м. Правила О. м. определяются синтаксисом конкретного языка. Например, в Паскале О. м. имеет вид: `label m1, m2, ..., mn;`, где `label` — ключевое слово (метка), `m1, m2, ..., mn` — метки, каждая из которых представляет собой целое число без знака

ОПИСАНИЕ ПЕРЕМЕННОЙ [variable declaration]. *Описание*, в котором указываются идентификатор (имя) переменной и *тип данных*, значения которого она может принимать. Обычно в одном О. п. могут быть определены

несколько переменных одного типа. В табл. О.1 приведены примеры О. п. на языках Паскаль и Си.

Таблица О.1. *Примеры описаний переменных*

Паскаль	Си
`var Root, Summa, x, y: real;`	`float Root, Summa, x, y;`
`Count, i: integer;`	`int Count, i;`
`Letter1, Letter2, Letter3: char;`	`char Letter1, Letter2, Letter3;`

Они определяют переменные `Root`, `Summa`, `x`, `y` как *вещественные переменные*, которые принимают значения *чисел с плавающей точкой*, имеющих стандартную длину, `Count`, `i` — как *целые переменные* стандартной длины и `Letter1`, `Letter2`, `Letter3` — как *символьные переменные*

ОПИСАНИЕ ПРОЦЕДУРЫ [procedure declaration]. Конструкция *языка программирования высокого уровня*, определяющая *процедуру*. Служит для присвоения имени процедуре, указания ее *формальных параметров* (если необходимо) и определения реализуемого процедурой алгоритма. О. п. может располагаться в программе в разделе описаний (например, в Паскале) или вслед за *главной программой*, оформленное в виде отдельного *программного модуля* (например, в Фортране). Форма О. п. устанавливается синтаксисом конкретного языка. Как правило, О. п. состоит из *заголовка процедуры* и *тела процедуры*. В заголовке указываются имя процедуры и, возможно, *формальные параметры*. В теле процедуры программируется выполняемый процедурой алгоритм. Например, процедура решения квадратного уравнения на языке Паскаль (см. *блок-схема*, рис. Б.4) имеет вид:

```
procedure Quadratic_equation (a, b, c: real; var x1, x2: real;
                                             var i: integer);
var d: real;
begin
  if a = 0 then begin
    if b = 0 then begin
      writeln ('Неверные данные'); i := 3
    end else begin
      writeln('Уравнение линейное'); x1 := – c / b; i := 1
    end
  end else begin
    d := b * 2 – 4 * a * c;
    if d < 0 then begin
```

```
      writeln ('Нет вещественных корней'); i := 0
    end else begin
      x1 := (- b + sqrt (d)) / (a + a);
      x2 := (- b — sqrt (d)) / (a + a);
      i := 2
    end { if d<0 }
  end { if a = 0 }
end { Quadratic_equation }
```

См. *заголовок процедуры, тело процедуры*

ОПИСАНИЕ ТИПА, определение типа [type declaration]. 1. *Синтаксическая конструкция языка программирования высокого уровня*, определяющая *тип данных*. С помощью О. т. в программе можно указать новое имя для уже существующего (стандартного для данного языка) типа данных либо ввести совершенно новый тип данных, присвоив ему имя. О. т. служит шаблоном для создания *объектов программы* этого типа, имеющих свои конкретные имена. В некоторых языках (например, в Паскале и Си) существуют средства конструирования *структурированных типов* данных. Поэтому для введения нового типа можно воспользоваться в О. т. этими средствами и стандартными типами данных. Кроме того, если необходимо, можно явно определить множество значений, допустимых для *элементов данных* вводимого типа, и множество применимых к ним операций. Введя таким образом новый тип данных, можно использовать его имя в описаниях. Например, тип данных "комплексные числа" можно ввести в программу на Паскале, как структурированный *тип "запись"* с двумя полями стандартного типа для вещественной и мнимой частей. См. пример в статье *тип "запись"*. 2. То же, что *описание*

ОПИСАНИЕ ФУНКЦИИ [function declaration]. Конструкция *языка программирования высокого уровня*, определяющая *функцию*. Служит для присвоения имени функции, указания ее *формальных параметров* (если необходимо) и определения реализуемого функцией алгоритма. О. ф. может располагаться в программе в разделе описаний (например, в Паскале) или вслед за *главной программой*, оформленное в виде отдельного *программного модуля* (например, в Фортране). Форма О. ф. устанавливается синтаксисом конкретного языка. Как правило, О. ф. состоит из *заголовка функции* и *тела функции*. В заголовке указываются имя функции и, возможно, *формальные параметры*. В теле программируется выполняемый функцией алгоритм. Например, функция, вычисляющая площадь треугольника по заданным длинам трех его сторон на языке Паскаль (см. *испытание программы*), имеет вид:

```
function Triangle_area (a, b, c: real; var i: integer): real;
var p: real;
```

```
begin
  if (a<0) or (b<0) or (c<0) then begin
    writeln ('Неверные данные');
    i := 2;
    Triangle_area:= 0
  end else
    if (a>b+c) or (b>a+c) or (c> a+b) then begin
      writeln ('Не треугольник');
      i := 0;
      Triangle_area:= 0
    end else begin
      i := 1;
      p := (a+b+c)*0.5;
      Triangle_area := sqrt(p*(p-a)*(p-b)*(p-c))
    end
end {Triangle_area}
```

См. *заголовок функции, тело функции*

ОПИСАТЕЛЬ [declarator, descriptor, specifier]. 1. То же, что *спецификатор*. 2. То же, что *дескриптор*

ОПРЕДЕЛЕНИЕ ТИПА [type declaration]. То же, что *описание типа*

ОПТИМИЗАЦИЯ [optimization]. Улучшение эффективности программы. Например, уменьшение *времени выполнения* за счет вынесения констант из циклов, объединения общих последовательностей операторов, подстановки *тела процедуры* вместо ее вызова и т. п. О. может выполняться вручную программистом на одном из завершающих этапов программирования или автоматически — транслятором с учетом специфики конкретной *архитектуры вычислительной системы*

ОПТИМИЗИРУЮЩИЙ КОМПИЛЯТОР [optimizing compiler]. *Компилятор*, выполняющий *оптимизацию* генерируемой *машинной программы*

ОПТИЧЕСКИЙ ДИСК [optical disk]. То же, что *лазерный диск*

ОПТИЧЕСКОЕ РАСПОЗНАВАНИЕ СИМВОЛОВ [optical character recognition (OCR)]. Процесс исследования изображений символов для выяснения их значения и представления в *цифровой форме данных*, понятных компьютеру. Программы О. р. с. позволяют после считывания *сканером* распознавать напечатанные или рукописные текстовые символы и вводить эти символы в компьютер так же, как если бы они вводились с обычной клавиатуры. Точность О. р. с. существенно зависит от качества изображения символов текста. Например, если О. р. с. применяется к тексту, напечатанному

типографским способом, то точность распознавания близка к 100%, в то время как точность распознавания рукописных символов обычно значительно меньше

ОПЦИЯ [option]. Дополнительный параметр, вариант режима или дополнительное средство, указываемое пользователем *обрабатывающей программе* или системе для модификации основного режима ее работы. Обычно О. указываются с помощью меню, списков, диалоговых окон или специальных команд. Например, О. программы MS Word указываются с помощью диалогового окна **Параметры** (рис. О.4)

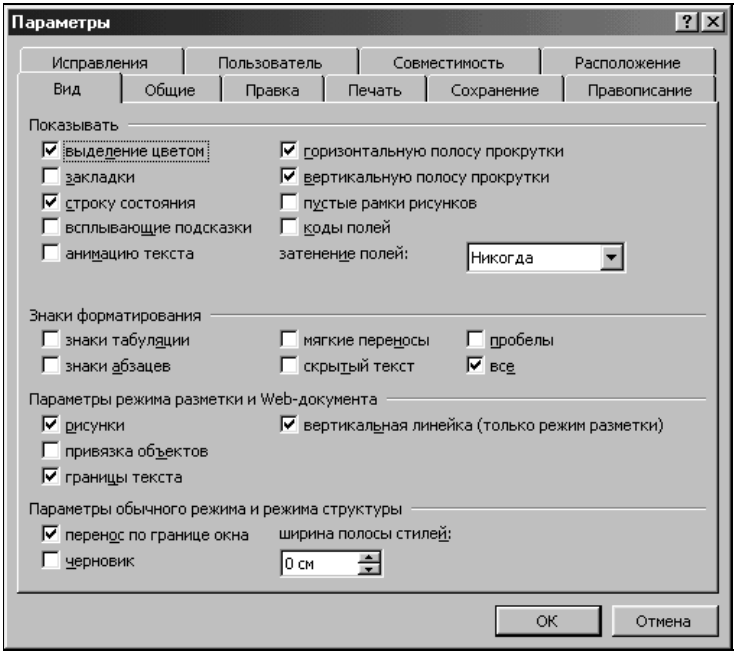

Рис. О.4. Диалоговое окно **Параметры** программы Microsoft Word

ОРГАНИЗАТОР ЛИЧНОЙ ДЕЯТЕЛЬНОСТИ [personal digital assistant (PDA)]. Карманный *персональный компьютер*, предназначенный для записи заметок, планирования встреч и выполнения других несложных задач. В О. л. д. обычно применяется специальная *операционная система* и отсутствует возможность установки дополнительных приложений. Часто в О. л. д. отсутствует клавиатура и ввод данных осуществляется с помощью сенсорного экрана. См. *пен-компьютер*

ОС [operating system (OS)]. То же, что *операционная система*

ОСНОВНАЯ ПАМЯТЬ [main memory, main storage]. 1. То же, что *оперативная память.* 2. Составная часть *оперативной памяти персональных IBM-совместимых компьютеров* при работе в среде MS-DOS (см. *оперативная память персональных IBM-совместимых компьютеров*). В настоящее время этот термин употребляется преимущественно в значении 1

ОСНОВНАЯ ПРОГРАММА [main program, head program]. То же, что *главная программа*

ОСНОВНОЙ КЛЮЧ [master key, primary key]. То же, что *первичный ключ*

ОСОБАЯ СИТУАЦИЯ [exception]. То же, что *исключительная ситуация*

ОСТАНОВ [stop, halt]. Прекращение автоматической выборки и выполнения команд *центральным процессором.* О. может быть вызван причинами, связанными с выходом из строя одного из устройств ЭВМ или *операционной системы — аварийный* О., а может быть программируемым — предусмотренным выполняемой программой. Ср. *ожидание, прерывание*

ОТКАТ [backtracking, undo, rollback]. 1. В *текстовых редакторах* и *системах программирования* — действия, отменяющие результат выполнения предыдущей команды или нескольких предыдущих команд и восстанавливающие измененное этими командами состояние обрабатываемого текста или переменных. 2. Возврат *вычислительного процесса* к ранее запомненному состоянию. О. выполняется с целью восстановления вычислительного процесса после *машинного сбоя*, отказа оборудования или ошибочных действий пользователя при *диалоговом режиме* управления программой. Возможность О. обеспечивается *операционной системой*, которая может периодически формировать *образ задачи*, либо заранее предусматривается программистом путем создания в программе *контрольных точек.* См. *откат транзакции*

ОТКАТ ТРАНЗАКЦИИ [rollback, transaction rollback]. Действия, выполняемые СУБД при *обработке транзакций* в том случае, когда успешного *завершения транзакции* не произошло. О. т. заключается в том, что отменяются результаты всех операций транзакции, а данные возвращаются в то исходное состояние, которое они имели до начала выполнения транзакции. О. т. вызывается различными причинами. 1) О. т. может быть осуществлен по инициативе самой транзакции, например, если в процессе ее выполнения обнаруживается, что завершение транзакции приведет к нарушению *целостности данных.* 2) О. т. может быть вызван *машинным сбоем* в процессе выполнения транзакции. 3) О. т. может быть произведен по инициативе СУБД для выхода из *тупиковой ситуации.* Ср. *завершение транзакции*

ОТКАЧКА [preemption]. То же, что *выгрузка*

ОТКРЫТИЕ ФАЙЛА, активизация файла [file activation]. Операция начала работы программы с *файлом.* Открыть файл, значит, связать программу с файлом: установить соответствие между физическим файлом на *внешнем за-*

поминающем устройстве, *буферной памятью* и набором данных программы, заполнить соответствующие поля в управляющих таблицах *операционной системы* и т. п. Иными словами, выполнить все операции, необходимые для обеспечения доступа к содержимому файла. Противоп. *закрытие файла*. См. *открытый файл*

ОТКРЫТОЕ СВОЙСТВО КЛАССА [public class property, public property]. *Свойство класса*, *область видимости* которого распространяется на все классы, из которых виден данный класс. В большинстве языков *объектно-ориентированного программирования* в описании О. с. к. используется ключевое слово `public`. Ср. *закрытое свойство класса, защищенное свойство класса*

ОТКРЫТЫЕ БАЗЫ ДАННЫХ [open databases]. *Базы данных*, которые поддерживают *открытый доступ к базам данных* (ODBC)

ОТКРЫТЫЙ ДОСТУП К БАЗАМ ДАННЫХ [open database connectivity (ODBC)]. Стандартный метод *доступа* к *базам данных*, разработанный фирмой Microsoft. В большинстве случаев доступ к данным при помощи О. д. к б. д. включает четыре этапа: 1) установку *драйвера* базы данных; 2) создание *источника данных ODBC* при помощи утилиты ODBC из *Панели управления* Windows; 3) установление *соединения* с источником данных; 4) использование инструкций *языка запросов SQL* для доступа к базе данных

ОТКРЫТЫЙ МЕТОД КЛАССА [public class method, public method]. *Метод класса*, чья *область видимости* распространяется на все классы, из которых виден данный класс. В большинстве языков *объектно-ориентированного программирования* в описании О. м. к. применяется ключевое слово `public`. Ср. *закрытый метод класса, защищенный метод класса*

ОТКРЫТЫЙ ФАЙЛ, **активный файл [active file]**. *Файл*, для которого выполнена операция *открытия файла*, в результате чего его записи стали доступными для чтения и обработки. Противоп. *закрытый файл*

ОТЛАДКА ПРОГРАММЫ [program debugging]. Этап *разработки программы*, состоящий в *локализации*, выявлении и устранении *программных ошибок*, факт существования которых уже установлен. О. п. имеет место тогда, когда очевидно, что программа либо не компилируется, либо работает неправильно. *Синтаксические ошибки* обычно выявляются в процессе *компиляции программы*. Наличие некоторых синтаксических, а также смысловых или *семантических ошибок* устанавливается в процессе *тестирования программы*. В *системах программирования* существуют специальные средства О. п. — отладчики, — которые позволяют в режиме интерпретации установить *контрольные точки*, выполнить отдельные участки программы и просмотреть результаты работы операторов. Очень эффективной может оказаться *трассировка программы*, но не будучи тщательно спланированной, она приводит к таким объемам выдаваемой информации, что разобраться в ней программисту бывает просто не под силу. Однако разработчику редко удается обойтись

стандартными средствами О. п. Желательно еще на этапе программирования предусматривать и вводить в программу собственные средства О. п., отслеживающие ход выполнения алгоритма, обращения к переменным и подпрограммам или воспроизводящие значения переменных. В этом случае места ошибок могут определяться с большой точностью. Удобству О. п. способствует *модульное программирование*, позволяющее проводить отдельно отладку каждого модуля программы. По мнению специалистов, О. п., как правило, занимает больше времени, чем все остальные стадии создания программы. Вот несколько советов программисту, данных в книге Д. Ван Тассела "Стиль, разработка, эффективность, отладка и испытание программ": первым делом проверяйте программу за столом; применяйте *отладочный компилятор*; вводите средства отладки как можно раньше; контролируйте правильность вводимых данных; используйте все доступные для вас средства отладки. Делайте программу правильной с самого начала

ОТЛАДОЧНЫЙ КОМПИЛЯТОР [checkout compiler]. *Компилятор*, обладающий средствами, облегчающими пользователю *отладку программы*. О. к. способен обнаруживать больше *синтаксических ошибок*, чем обычный, поскольку он детально анализирует синтаксическую конструкцию и взаимодействие операторов. Существуют О. к., способные устранять отдельные виды синтаксических ошибок. Кроме того, в процессе трансляции О. к. производит многочисленные проверки *исходной программы*, выявляющие переменные, для которых не заданы начальные значения, неправильные индексы и запрещенные переходы. Процесс *компиляции программ* О. к. осуществляется значительно медленнее по сравнению с обычным компилятором, и, как правило, О. к. не являются оптимизирующими

ОТЛАДЧИК [debugger]. *Программа*, облегчающая программисту выполнение *отладки* разрабатываемых им *программ*. О. помогает анализировать поведение отлаживаемой программы, обеспечивая ее трассировку, выполняя остановы в указанных точках или при заданных условиях, позволяя просмотреть текущие значения переменных, содержимое *ячеек памяти*, а иногда и регистров процессора, и при необходимости изменить эти значения. О. является важной составной частью *системы программирования*. См. *отладка программ*

ОТЛОЖЕННАЯ ЗАПИСЬ [lazy write]. Алгоритм обновления данных на диске, обладающий максимально возможной производительностью. *Файловые системы* с О. з. помещают изменения данных на диске в кэш-память и выводят оптимизированную информацию из кэша на диск, часто в фоновом режиме. Безопасность данных в файловых системах с О. з. обычно ниже, чем в файловых системах с прямой записью

ОТЛОЖЕННОЕ СОБЫТИЕ [deferred event]. Событие, распознавание которого отложено до тех пор, пока объект не выйдет из определенного состояния

ОТМЕТКА ФАЙЛОВ И КАТАЛОГОВ [files and directories marking, files and directories selection]. Выделение одного или группы *файлов* или *каталогов* с целью копирования, переноса или удаления. При работе с *оболочкой Norton Commander* (или другой, ей подобной) можно отметить файлы и каталоги по одному, нажимая клавишу <Insert>. При этом *цветовой маркер* на активной панели NC перемещается на строку вниз, а строка, на которой он находился, отмечается (окрашивается в другой цвет). Можно также отмечать файлы, используя *шаблон имени файла*. Для этого следует нажать на *малой цифровой клавиатуре* клавишу <+>. На экране появится окно с уже введенным шаблоном *.* (отметить все файлы). В этот шаблон можно внести изменения. Например, чтобы отметить все файлы с расширением txt, необходимо ввести шаблон *.txt. Далее следует нажать клавишу <Enter>. Окно исчезнет, а файлы, имена которых соответствуют шаблону, будут отмечены. Можно снять отметку, нажав клавишу <−> на цифровой клавиатуре. В *операционной системе Windows* О. ф. и к. выполняется с помощью мыши или другого *указательного устройства*. См. *выделение*

ОТНОСИТЕЛЬНЫЙ АДРЕС [relative address]. *Адрес*, выражающий разницу по отношению к значению *базового адреса*. Команды *объектных модулей* всегда содержат О. а., как правило, выражающие номера байтов или *ячеек памяти* относительно начала *области памяти*, занимаемой программой, или начала ее *рабочей области*. В процессе *загрузки программы* устанавливаются базовые адреса, соответствующие фактически распределенной памяти, и О. а. либо сразу пересчитываются в *физические адреса*, либо процессор вычисляет физические адреса команд и данных во время выполнения программы. Применение О. а. делает программу независимой от места ее расположения в *оперативной памяти* во время выполнения. См. *адресация*. Ср. *логический адрес*

ОТНОШЕНИЕ [relation, relationship]. 1. В математике — подмножество прямого (декартового) произведения множеств. Например, О. "больше" на множестве чисел: 5 больше 2, т. е. пара (5, 2) принадлежит отношению "больше". О. можно задать, явно перечислив все наборы элементов, которые принадлежат этому отношению. На этом обстоятельстве основаны *реляционные базы данных*. О. можно задать с помощью условия *(логического выражения)*, проверяющего, находятся ли указанные элементы в данном отношении. На этом обстоятельстве основано использование понятия О. в языках программирования, как указано в следующем пункте. 2. В *языках программирования высокого уровня* — два *арифметических выражения*, соединенные знаками *операций сравнения*, например, `x = 0.5, A + B < 1, cos(z) >= z`. Значением О. является *логическое значение* "ИСТИНА" или "ЛОЖЬ" ("да" или "нет"). Если условие О. выполнено, то результат — "ИСТИНА". В противном случае результат — "ЛОЖЬ". Например, если `x = 0.5`, то результатом О. `x = 0` будет "ЛОЖЬ". О. само является *логическим выражением* либо входит в него как составная часть. См. *логическое выражение* 3. В *унифициро-*

ванном языке моделирования UML семантическая связь между элементами модели. Существует несколько видов О. К ним относятся *ассоциация*, *обобщение*, и др.

ОТОБРАЖАЕМАЯ ПАМЯТЬ [expanded memory]. Дополнительная память, доступ к которой осуществляется путем копирования (отображения) ее страниц в *расширенную память*. О. п. реализуется в виде дополнительной платы памяти или эмулируется процессором 80386 в расширенной памяти. В других ситуациях в настоящее время этот термин не употребляется. См. *оперативная память персональных IBM-совместимых компьютеров*

ОТРАЖЕНИЕ СВЕРХУ ВНИЗ [flip vertical]. То же, что *зеркальное отражение по вертикали*

ОТРАЖЕНИЕ СЛЕВА НАПРАВО [flip horizontal]. То же, что *зеркальное отражение по горизонтали*

ОТСТУП [indent]. Расстояние от текста до поля страницы. Например, небольшой О. первой строки абзаца называется красной строкой. О. может быть слева и справа. Можно использовать отрицательные отступы для размещения текста на полях

ОФИСНОЕ ПРИЛОЖЕНИЕ [office application]. *Автоматизированная система управления*, применяемая для *автоматизации* конторского труда

ОЦИФРОВЫВАТЬ [digitize]. Преобразовывать любую непрерывно изменяемую *информацию* в *цифровую форму* данных. Так, сканер О. рисунок на бумаге, преобразуя нарисованные линии и цветовые оттенки в *двоичный код*, который вводится в память компьютера. Непрерывный звуковой сигнал О. в комбинацию нулей и единиц, которая записывается на лазерный диск. Процедура оцифровки обычно выполняется с помощью *аналого-цифрового преобразователя*

ОЧЕРЕДЬ [queue, first in first out (FIFO)]. Упорядоченный набор *элементов данных*, в котором можно удалять и добавлять элементы, причем новый элемент всегда записывается в конец набора, а очередной читаемый или удаляемый элемент всегда выбирается из начала набора. Таким образом, первый добавленный элемент О. является единственным доступным и первым удаляемым элементом (по принципу "первым вошел — первым ушел"). Ср. *стек*

ОЧЕРЕДЬ ЗАДАНИЙ НА ПЕЧАТЬ [print queue]. Последовательность документов, отправленных на печать и ждущих завершения печати других документов. В терминологии Windows О. з. н. п. представляет собой группу документов, ожидающую распечатки

ОШИБКА АЛГОРИТМА [algorithm error]. То же, что *логическая ошибка*

ОШИБКА В ПРОГРАММЕ, программная ошибка [bug, program error, malfunction]. Любая ошибка, содержащаяся в *программе*. Наличие О. в п. очевидно, когда программа либо не компилируется, либо работает неправильно. Различают *синтаксические* и *семантические ошибки*. Они выявляются при *трансляции* и *испытаниях программы* и устраняются в процессе *отладки программы*

ОШИБКА ДЕЛЕНИЯ НА НОЛЬ, деление на ноль [divide by zero error]. *Ошибка*, возникающая при выполнении команды деления, если делитель равен 0. Помимо *семантической ошибки* в программе О. д. н. н. может быть вызвана проблемами, требующими дальнейшего исследования (например, поврежденная память, аппаратные проблемы или программные сбои)

ОШИБКА КЭША, промах кэша [cache miss]. Неудачное обращение к *кэш-памяти* для чтения данных, которых там нет. Попытка осуществить доступ к части *кэшированного файла*, которая отсутствует в кэше. При попытке диспетчера кэша скопировать данные в пользовательский буфер происходит прерывание типа *"ошибка страницы"*. Тогда диспетчер *виртуальной памяти*, в свою очередь, вызывает соответствующий драйвер *файловой системы* для копирования файла с диска в кэш

ОШИБКА НЕДОПУСТИМОГО КОДА ОПЕРАЦИИ [invalid opcode error]. Ошибка, возникающая, когда процессор пытается выполнить команду, которая не определена. Эта ошибка практически всегда является следствием физического повреждения памяти или потери управления в программе

ОШИБКА ПЕРЕПОЛНЕНИЯ [overflow error, overrun error]. 1. Ошибка арифметического *переполнения*. Имеет место, когда результат выполнения арифметической операции настолько велик, что не укладывается в *диапазон изменения чисел* данного компьютера. Например, если известно, что *число целого типа* в *двоичной системе счисления* занимает *машинное слово* длиной 4 байта = 32 бита, то О. п. возникнет, если результат умножения целых чисел превышает 32-разрядное целое. Эта ошибка также может быть вызвана *аппаратным сбоем*. 2. Ошибка, которая происходит, когда устройство, принимающее данные, не может обрабатывать их так же быстро, как они поступают

ОШИБКА СТРАНИЦЫ [page fault]. То же, что *ошибочное обращение к отсутствующей странице*

ОШИБОЧНОЕ ОБРАЩЕНИЕ К ОТСУТСТВУЮЩЕЙ СТРАНИЦЕ, ошибка страницы [page fault]. *Прерывание*, возникающее при попытке программы считать или записать данные в участок *виртуальной памяти*, помеченный как "отсутствующий". Управляющее устройство системы виртуальной памяти оперирует информацией со статусом каждой страницы в виртуальном *адресном пространстве*. Страница либо проецируется на физический адрес, либо отсутствует в физической памяти. Когда обнаруживается обращение по

виртуальному адресу, не спроецированному на физический, устройство управления памятью генерирует это прерывание. *Операционная система* реагирует на ошибку тем, что подкачивает данные на эту страницу и обновляет данные о ее статусе в блоке управления памятью. См. *свопинг*

П

ПАКЕТ [batch, package, packet]. 1. Порция данных, передаваемая как одно целое. Например, в сети Интернет передаваемые данные разбиваются на П. размером от 40 до 32 000 байт в зависимости от *аппаратных средств* и *носителей информации*, и каждый П. перемещается по сети независимо от других П. 2. Совокупность программ, выполняемых компьютером в режиме *пакетной обработки* 3. *Программный модуль* в некоторых *языках программирования*, содержащий описания нескольких классов и определяющий *пространство имен.* 4. Группирующая сущность *унифицированного языка моделирования* UML. П. могут быть вложены в другой П. В П. могут находиться как *элементы модели*, так и *диаграммы*

ПАКЕТ ДИСКОВ [disk pack]. То же, что *пакет магнитных дисков*

ПАКЕТ ЗАПРОСА ПРЕРЫВАНИЯ [interrupt request package (IRP)]. *Структура данных*, используемая *драйверами устройств* и другими компонентами *режима ядра* для обмена данными, связанными с *обработкой прерывания*

ПАКЕТ МАГНИТНЫХ ДИСКОВ, пакет дисков [disk pack]. Сменный блок *магнитных дисков*, в котором несколько одинаковых *жестких магнитных дисков* расположены на равных расстояниях друг от друга. Диски жестко соединены между собой, чтобы сохранить устойчивость пакета при вращении с большой скоростью в дисководе. Информация записывается на обеих сторонах каждого диска вдоль концентрических дорожек, которые обычно делятся на секторы. Доступ к секторам осуществляется при вращении пакета. П. м. д. широко применялись в ЭВМ второго и третьего поколений. См. *поколения ЭВМ.* Все *головки записи/чтения* располагаются на одной линии, параллельной оси пакета, и имеют общее устройство позиционирования, которое обеспечивает их одновременное расположение над дорожками с одинаковыми номерами, находящимися на различных поверхностях дисков. Такая конструкция позволяет уменьшить *время выборки* взаимосвязанных записей, размещая их на одном цилиндре, образованном дорожками с одинаковыми номерами. П. м. д. проектируются с учетом международных стандартов, обеспечивающих их совместимость с дисководами как по механическим, так и по магнитным характеристикам. Необходимая для контроллера дисковода информация о шаге дорожек и размещении на них секторов записывается на специальной поверхности пакета, называемой сервоповерх-

ностью. Вне дисковода пакет хранится в пластмассовом кожухе, предохраняющем его от пыли и механических повреждений

ПАКЕТ ПРИКЛАДНЫХ ПРОГРАММ (ППП) [application program package]. Система *прикладных программ*, предназначенных для решения прикладных задач определенной *предметной области*, например, ППП, позволяющих проводить прочностные расчеты инженерных конструкций. Обычно программы, входящие в ППП, могут совместно использовать одни и те же данные и объединяются единым *пользовательским интерфейсом*. Развитые ППП содержат программу-монитор, программы планирования работ и допускают формулирование заданий в терминах предметной области

ПАКЕТНАЯ ОБРАБОТКА, пакетный режим [batch processing, batch mode]. Режим или метод организации работы *вычислительной системы*, при котором задания или накопленные заранее данные по определенным критериям объединяют в *пакет* для последующей автоматической обработки в соответствии с заданными приоритетами. При этом управление обработкой осуществляется *операционной системой*, и пользователь не может влиять на ход выполнения заданий, пока продолжается обработка пакета. П. о. широко применялась во втором *поколении ЭВМ* с целью снижения непроизводительных затрат *машинного времени* и эффективного использования оборудования. В настоящее время применяется П. о. *фоновых задач* в *системах коллективного пользования*

ПАКЕТНАЯ ПРОГРАММА ОТЛАДКИ СЕТЕЙ, утилита PING [packet Internet groper (PING)]. *Утилита* проверки возможности соединения с сервером локальной или глобальной сети (включая Интернет) с применением протокола TCP/IP. Программа посылает тестовый пакет и ожидает ответ от определенного рабочего компьютера в заданном промежутке времени

ПАКЕТНЫЙ РЕЖИМ [batch mode]. То же, что *пакетная обработка*

ПАКЕТНЫЙ ФАЙЛ [batch file]. То же, что *командный файл*

ПАЛИТРА [palette]. 1. Множество цветов, которые можно отобразить на экране дисплея. Богатство П. зависит как от аппаратного, так и от программного обеспечения. Монитор и видеоадаптер компьютера определяют *разрешающую способность* экрана и максимально возможное количество отображаемых на нем цветов. Однако имеющееся программное обеспечение вправе не использовать всех графических возможностей, предоставляемых аппаратурой, и ограничить число цветов, доступных для формирования изображения. Цвет каждого пиксела в *машинной программе* задается двоичным кодом. Поэтому количество цветов в П. определяется числом бит, отведенным для представления цвета одного пиксела. Например, если для представления цвета пиксела отводится 4 бита, то пиксел может иметь 16 цветов (коды от 0000 до 1111). Следовательно, 4-битовый пиксел допускает 16-цветную П. 2. Таблица соответствия между кодами цветов и цветами, ото-

бражаемыми на экране дисплея. 3. Набор цветов в *программах рисования* или набор цветов и инструментов рисования в *программах раскраски*. См. *закраска, заливка, цвет*

ПАМЯТЬ [memory, storage, store]. 1. Общее название для любых реальных или абстрактных средств и механизмов фиксации и сохранения информации. 2. То же, что *память ЭВМ*. 3. То же, что *запоминающее устройство*

ПАМЯТЬ ПРОИЗВОЛЬНОГО ДОСТУПА, память прямого доступа [random access memory (RAM)]. 1. То же, что *оперативная память*. 2. *Запоминающие устройства* на микросхемах или блоках микросхем (например, так называемые модули SIMM)

ПАМЯТЬ ПРЯМОГО ДОСТУПА [random access memory (RAM)]. То же, что *память произвольного доступа*

ПАМЯТЬ ЭВМ [memory, storage, store]. Функциональная часть ЭВМ, предназначенная для записи, хранения и считывания информации. П. Э. подразделяется на *основную*, или *оперативную*, *память* и *внешнюю память*. Технически реализуется в виде *запоминающих устройств*. См. *оперативная память, внешняя память, видеопамять, виртуальная память, верхняя память, функциональная схема ЭВМ*

ПАНЕЛЬ [panel]. То же, что *панель экрана*

ПАНЕЛЬ ЗАДАЧ [task bar]. *Панель экрана операционной системы Windows*, обычно расположенная горизонтально в нижней части *рабочего стола* (рис. П.1). Содержит кнопку **Пуск**, табло, показывающее текущее время, обозначенные значками кнопки управления системой и кнопки задач, которые появляются по мере запуска приложений и используются для переключения между ними. П. з. можно скрыть, переместить в верхнюю часть рабочего стола или настроить ее другими способами

Рис. П.1. Панель задач

ПАНЕЛЬ ИНСТРУМЕНТОВ, инструментальная панель [toolbar]. *Панель экрана* с размещенным на ней *пиктографическим меню*. На рис. П.2 приведена П. и. **Стандартная** текстового процессора Microsoft Word. На этой панели сосредоточены кнопки вызова инструментальных программных средств, с помощью которых выполняются стандартные действия, например, создание, открытие, сохранение, печать документа и т. д. Нажатие нужной кнопки с помощью мыши приводит к выполнению соответствующего действия

Рис. П.2. Панель инструментов **Стандартная** приложения Microsoft Word

ПАНЕЛЬ УПРАВЛЕНИЯ [control panel]. В *операционной системе Windows* — окно, в котором собраны значки различных *обслуживающих программ*, предназначенных для настройки *операционной системы* и *среды пользователя* (рис. П.3)

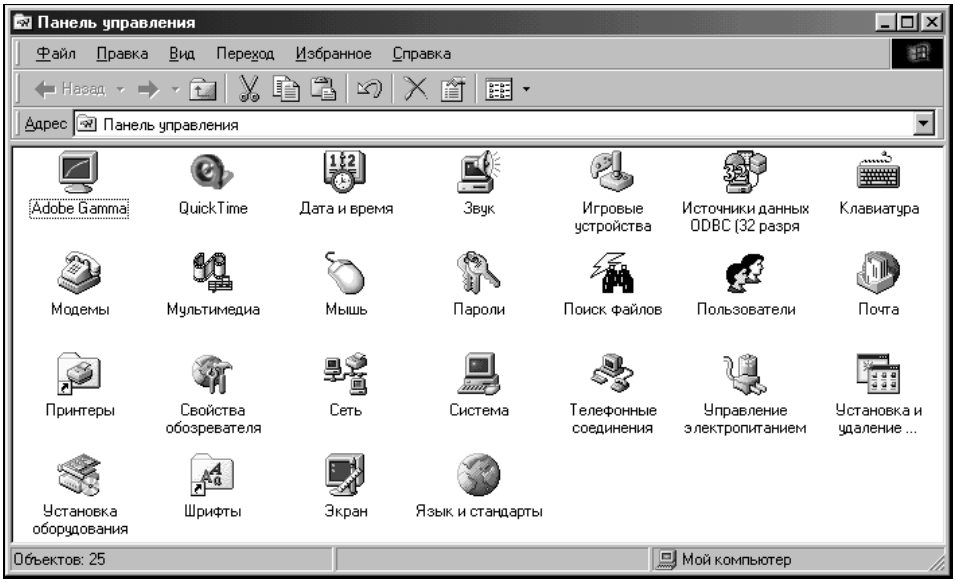

Рис. П.3. Панель управления Windows

ПАНЕЛЬ ЭКРАНА, экранная панель, панель [screen panel, panel]. Выделенная часть *экрана* дисплея или окна, предназначенная для размещения меню, управляющих *кнопок*, справочной и другой информации. Пользователь не может изменять размеры и расположение П. э. (а размер окна — может). Например, П. э. *оболочки Norton Commander*, несущие информацию о дисках (см. рис. О.1). Ср. *окно*

ПАПКА [folder]. 1. То же, что *каталог*. 2. *Каталог* в *операционных системах Windows*, имеющий экранное представление "деловой папки"

ПАРАДИГМА ПРОГРАММИРОВАНИЯ [programming paradigm]. Согласованный набор принципов и правил, которые используются при *разработке программ* и поддерживаются *инструментальными программными средствами*.

Например, *структурное программирование*, *объектно-ориентированное программирование*, *аспекто-ориентированное программирование*

ПАРАЛЛЕЛИЗМ [concurrency]. 1. Наличие в алгоритме или программе независимых друг от друга операций, которые можно выполнить одновременно. См. *параллельные вычисления, параллельный алгоритм* 2. Наличие среди обрабатываемых данных массивов, допускающих применение одной операции сразу к нескольким элементам. См. *векторная операция, векторный процессор.* 3. Способность *программного обеспечения* или аппаратуры реализовать *параллельную обработку* или *параллельные вычисления.* 4. Одновременное выполнение двух и более деятельностей, которые осуществляются независимо, за исключением явно заданных точек синхронизации. П. может быть достигнут за счет *параллельной обработки* или за счет *разделения времени*

ПАРАЛЛЕЛЬНАЯ ВИРТУАЛЬНАЯ МАШИНА, технология PVM [parallel virtual machine (PVM)]. Технология программирования *гетерогенных вычислительных систем*, основанная на *модели передачи сообщений.* Она позволяет объединить разнородный набор компьютеров (суперкомпьютеры, *рабочие станции* и т. п.) сетями любого вида (Ethernet, интранет и т. п.) в один вычислительный ресурс, который называют параллельной виртуальной машиной. П. в. м. скрывает от программиста реальную неоднородность находящегося в его распоряжении единого многопроцессорного вычислительного комплекса. Пользователь сам управляет конфигурацией виртуального параллельного компьютера, причем изменяя ее в процессе работы программы. *PVM-программа* порождает множество взаимодействующих между собой *параллельных вычислительных процессов*, называемых задачами. Задача в разные моменты времени может выполнять вычисления, обмениваться данными с другими задачами и запускать новые процессы или завершать выполнившие свою работу. В П. в. м. используются *двухточечный* и *коллективный обмены* разнотипными данными, ориентированные на выполнение программы в гетерогенной конфигурации. Идеи П. в. м. начала разрабатывать в 1989 г. группа специалистов из США. Результатом явились сопряженные с языками Си, Си++ и Фортран *системы программирования PVM*, которые представляют собой набор библиотек и утилит, предназначенных для разработки и отладки параллельных программ, а также для управления конфигурацией виртуальной вычислительной машины. Они работают с минимальной модификацией в любой операционной системе класса UNIX. Их отличает устойчивая работа, масштабируемость, поддержка любых типов гетерогенности и переносимость. Ср. *интерфейс передачи сообщений*

ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА [concurrent processing, parallel processing, simultaneous processing]. 1. Режим работы компьютера, при котором выполняются несколько *параллельных вычислительных процессов.* Этот режим может быть реализован как на *многопроцессорной ЭВМ*, так и на компьютере с одним процессором. В последнем случае необходима *многозадачная операцион-*

ная система, обеспечивающая *разделение времени* между программами. Поскольку *кванты времени*, предоставляемые чередующимся процессам, для человека неразличимо малы, эти программы кажутся выполняемыми параллельно. На многопроцессорной ЭВМ П. о. может означать как одновременное выполнение нескольких задач, так и решение одной задачи, *параллельный алгоритм* которой позволяет распределить отдельные ее сегменты по всем доступным в данный момент процессорам. См. *квантование времени, мультипрограммирование.* 2. То же, что *параллельные вычисления*

ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ [parallel computing, parallel calculations]. Реализация *параллельного алгоритма* на *многопроцессорной ЭВМ* или с одновременным привлечением нескольких компьютеров, объединенных в *вычислительную сеть*

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ [concurrent algorithm, parallel algorithm]. *Алгоритм*, в котором можно выделить и одновременно выполнить независимые друг от друга группы операций. Примером П. а. может служить алгоритм умножения матрицы на вектор, в котором каждый компонент результирующего вектора может быть вычислен независимо от других. П. а. может быть реализован в виде программы, задающей несколько *параллельных вычислительных процессов*

ПАРАЛЛЕЛЬНЫЙ ВЫЧИСЛИТЕЛЬНЫЙ ПРОЦЕСС [concurrent calculation process, parallel calculation process]. Один из двух или более *вычислительных процессов*, одновременно выполняющихся на компьютере за счет привлечения различных ресурсов *вычислительной системы.* Например, в *многопроцессорной ЭВМ*, используя разные процессоры, можно либо параллельно решать различные задачи, либо вести вычисления одной задачи согласно *параллельному алгоритму.* В этих случаях на разных процессорах будут выполняться П. в. п.

ПАРАЛЛЕЛЬНЫЙ КОМПЬЮТЕР [parallel computer]. *Вычислительная система*, допускающая проведение *параллельных вычислений.* См. *симметричный многопроцессорный компьютер*

ПАРАЛЛЕЛЬНЫЙ ПОРТ [parallel port]. *Порт*, передающий данные порциями по несколько бит параллельно по нескольким *линиям связи.* Например, если процессор соединен с *внешним устройством* восемью проводами через П. п., то он будет способен передавать одновременно 8 бит информации — по одному в каждую линию. Противоп. *последовательный порт*

ПАРАМЕТР ЦИКЛА, переменная цикла, управляющая переменная цикла [cycle parameter, cycle index, loop variable]. *Переменная*, используемая в *операторе цикла* для управления повторением *тела цикла.* Например, в написанных на языках Фортран, Паскаль и Си операторах цикла `DO 10 K = 1,50...`, `for i:= 1 to 65 do...` и `for(count = 1; count <= number; count++)...` П. ц. являются, соответственно, переменные `K`, `i` и `count`

ПАРОЛЬ [password]. Уникальная последовательность символов, которую необходимо ввести по запросу компьютера, чтобы получить доступ к системе, программе или данным. Система сравнивает введенный П. с хранящимся в памяти образцом. Если произошло совпадение кодов, пользователь получает желаемый доступ. Предполагается, что П. известен ограниченному кругу лиц и поэтому считается средством ограничения доступа к вычислительной системе и ее файлам. См. *защита данных, несанкционированный доступ*

ПАСКАЛЬ [Pascal]. *Язык программирования высокого уровня.* Предварительная версия П., разработанная швейцарским ученым Н. Виртом, была опубликована в 1968 г. и предназначалась для обучения программированию как систематической дисциплине. Впоследствии, претерпев некоторые изменения, П. стал одним из основных языков программирования вычислительных и информационно-логических задач. *Структуры данных* (массивы, записи, файлы и типы данных, определяемые пользователем), а также процедуры и основные управляющие конструкции языка *(условные операторы*, *операторы цикла* и *вызова процедур)* позволяют использовать весь набор методов *структурного программирования* и создавать простые, понятные и удобочитаемые программы. П. был относительно легко реализован на многих типах ЭВМ. Особенно популярны разрабатываемые фирмой Borland International Inc. *среды программирования* для *персональных компьютеров*, основанные на языке П. (Turbo Pascal, Borland Pascal, Delphi), которые постоянно развиваются и широко применяются многими разработчиками *программного обеспечения*

ПАТТЕРН [design pattern]. То же, что *образец проектирования*

ПЕН-КОМПЬЮТЕР [pen computer]. *Портативный компьютер*, в котором основным устройством ввода является *перо*, а не клавиатура и мышь. П.-к. оснащен *дисплеем на жидких кристаллах*, *сенсорный экран* которого позволяет вводить данные и управлять компьютером с помощью пера. Для этого используется специальное *программное обеспечение*. П.-к. выполняет функции записной книжки и незаменим во время деловых встреч, при проведении социологических опросов и тому подобных работах. П.-к. может иметь жесткий диск и обмениваться данными с другими компьютерами, для чего применяются кабели или модем

ПЕРВИЧНЫЙ КЛЮЧ, основной ключ [master key, primary key]. *Ключ*, однозначно идентифицирующий запись в файле или таблице базы данных. Кроме П. к. в записи могут быть предусмотрены вторичные (внешние) ключи. См. *вторичный ключ*, *составной ключ* (см. рис. В.5)

ПЕРЕВОД СТРАНИЦЫ, подача страницы, прогон страницы [form feed (FF)]. Команда для принтера, предписывающая ему перейти к началу следующей страницы. В наборе *кодов ASCII* команде П. с. соответствует *управляющий символ* FF (см. табл. А.1), имеющий десятичное значение 12 (шестнадцатеричное 0Ch). Кроме того, он часто называется символом выдачи страницы

(page-eject character), т. к. его назначение заключается в том, чтобы начать печатать на следующей странице

ПЕРЕДАЧА ДАННЫХ [data communication, data transfer]. Пересылка данных от устройства или процесса — *источника данных* к устройству или процессу — *приемнику данных.* В *вычислительных сетях* П. д. обычно осуществляется в соответствии с правилами, определяющими формат и процедуры *обмена данными* между двумя или более независимыми устройствами или процессами. Совокупность таких правил называется протоколом. Ср. *обмен данными.* См. *протокол*

ПЕРЕДАЧА ПАРАМЕТРОВ [parameter passing]. Механизм замены *формальных параметров* на *фактические параметры* при *обращении к процедуре* или *к функции.* Различают три вида П. п.: по значению, по наименованию и по ссылке. При П. п. по значению в памяти выделяется место для всех значений параметров (как для локальных переменных), при этом в данную область памяти передается только копия значения фактического параметра. Если фактический параметр является выражением, то сначала вычисляется его текущее значение, которое затем копируется в выделенную память. После выхода из процедуры память, отведенная для значений параметров, освобождается, и произведенные процедурой изменения значений не отражаются на исходных объектах *вызывающей программы.* Например, если фактическим параметром является имя переменной, то значение этой переменной после выполнения процедуры не изменится. По значению могут передаваться лишь исходные данные процедуры. П. п. по наименованию имеет место, если формальный параметр описан как параметр-переменная. При этом происходит как бы подстановка текста фактического параметра во все места *тела процедуры*, где расположен соответствующий формальный параметр. Поэтому если фактическим параметром является выражение, то его значение при выполнении процедуры будет вычисляться столько раз, сколько в теле процедуры встречается формальный параметр. Рекомендуется использовать П. п. по наименованию для возврата в вызывающую программу результатов выполнения процедуры. П. п. по ссылке означает, что вызываемой процедуре передается адрес *объекта вызывающей программы.* Любые операции, в записи которых участвует соответствующий формальный параметр, выполняются непосредственно над объектом, адрес которого указан в качестве фактического параметра

ПЕРЕДАЧА УПРАВЛЕНИЯ [control transfer]. Изменение естественной последовательности выполнения команд в программе. Необходимость П. у. обуславливается *ветвлением программы.* П. у. задается с помощью *команд* П. у. Термин "П. у." обычно применяется, если речь идет о программировании на *машинных языках.* В *языках программирования высокого уровня* чаще употребляется эквивалентный термин *"переход".* См. *поток управления*

ПЕРЕЗАГРУЗКА, повторная начальная загрузка [reboot]. Повторная *загрузка операционной системы*. Как правило, выполняется при зависании *вычислительной системы* или иных нарушениях функционирования компьютера, когда другие способы восстановления нормальной работы не дают результатов. Экстренная П. *персонального компьютера* вызывается либо одновременным нажатием клавиш <Ctrl>+<Alt>+<Delete>, либо нажатием кнопки Reset на *системном блоке*. В рамках *операционной системы Windows* П. выполняется по команде **Перезагрузить компьютер** в окне **Завершение работы Windows**. Ср. *перезапуск*

ПЕРЕЗАПИСЬ [rewrite]. Повторная запись информации на *носитель данных*. Это может быть повторная запись на тот же носитель с целью восстановления поврежденных данных или запись заново при сбое записывающего устройства, а может быть П. на другой носитель с целью получения *резервной копии*. Ср. *восстановление*

ПЕРЕЗАПУСК, повторный пуск, рестарт [rerun, restart]. Повторный *запуск программы*. П. выполняется либо после *аварийного завершения*, вызванного *машинным сбоем* или сбоем в работе *операционной системы*, либо после временной остановки, возникшей, например, из-за нежелательного хода вычислительного процесса. В первом случае П. может быть произведен автоматически операционной системой, во втором также автоматически, если это заранее предусмотрено в задании, либо вручную — пользователем. Во многих программах программистами устанавливаются *контрольные точки*, позволяющие осуществлять повторный пуск не с самого начала, т. е., не повторяя *загрузку программы*. См. *образ задачи, откат*. Ср. *перезагрузка*

ПЕРЕИМЕНОВАНИЕ КАТАЛОГА [directory rename]. Присвоение нового имени ранее созданному каталогу. П. к. выполняется с помощью стандартных средств *операционной системы*. На *персональных компьютерах* каталог переименовывается так же, как и файл. См. *переименование файла*

ПЕРЕИМЕНОВАНИЕ ФАЙЛА [file rename]. Присвоение нового имени определенному ранее файлу. П. ф. выполняется с помощью стандартных средств *операционной системы*. Например, в *операционной системе Windows* П. ф. осуществляется следующим образом. Нужно выделить переименовываемый файл (в окне Проводника, в окне папки или в другом месте) и один раз щелкнуть кнопкой мыши. При этом система перейдет в режим редактирования имени файла. Изменив имя файла, нужно нажать клавишу <Enter>

ПЕРЕКЛЮЧАТЕЛЬ [switch]. 1. *Устройство*, изменяющее электрическую цепь. 2. То же, что *оператор выбора*. 3. Выражение в *операторе выбора*, управляющее выбором нужной *ветви программы*. 4. То же, что *радиокнопка*

ПЕРЕКЛЮЧАЮЩЕЕ СОБЫТИЕ [trigger]. Событие, наступление которого делает возможным запуск перехода в *конечном автомате*

ПЕРЕКРЕСТНАЯ ССЫЛКА [cross reference]. Набор *атрибутов* (тип, адрес, место определения, список мест применения) *идентификатора*, используемого в программе

ПЕРЕКРЫТИЕ, оверлей [overlay]. 1. Метод использования одних и тех же областей *оперативной памяти* на различных этапах выполнения программы. При этом программа разбивается на сравнительно независимые части *(сегменты перекрытий)*, которые попеременно по мере надобности загружаются в одну и ту же область оперативной памяти. *Системы программирования* позволяют программисту заранее придать программе *оверлейную структуру*, описание которой специальными командами передается *редактору связей*. См. *оверлейная программа, оверлейная структура*. 2. То же, что *сегмент перекрытия*. 3. Наложение одного изображения на другое или его часть

ПЕРЕМЕННАЯ [variable]. *Элемент данных* в программе, которому присвоено имя. П. различаются по имени и принимают разные значения, т. е. не являются постоянными. Значение П. может быть получено и изменено программой. *Тип данных*, к которому могут принадлежать значения П., устанавливается *описанием переменной*. В некоторых языках (например, в Фортране) допускается *неявное описание* П. В зависимости от типа принимаемых значений П. могут быть *целыми, вещественными, логическими* и т. д.

ПЕРЕМЕННАЯ С ИНДЕКСАМИ [subscripted variable]. То же, что *элемент массива*

ПЕРЕМЕННАЯ ЦЕЛОГО ТИПА [integer variable]. То же, что *целая переменная*

ПЕРЕМЕННАЯ ЦИКЛА [loop variable]. То же, что *параметр цикла*

ПЕРЕМЕЩАЕМОСТЬ [relocatability]. Свойство программы или *программного модуля* быть загруженными в любом разрешенном месте *оперативной памяти* и при необходимости быть перемещенными из одного участка памяти в другой. Свойством П., как правило, обладают *объектные* и *загрузочные модули* и *программы*

ПЕРЕМЕЩЕНИЕ КАТАЛОГА [directory moving]. То же, что *перенос каталога*

ПЕРЕМЕЩЕНИЕ ФАЙЛА [file moving]. То же, что *перенос файла*

ПЕРЕНОСИМОСТЬ ПРОГРАММЫ, мобильность программы, портабильность программы [program portability]. Свойство программы, позволяющее переносить ее без изменений или с минимальными переделками с одного компьютера на другой, в особенности когда эти ЭВМ имеют различную архитектуру. П. п. обеспечивается *системами программирования*

ПЕРЕНОС КАТАЛОГА, перемещение каталога [directory moving]. Процедура воспроизведения содержимого одного *каталога* в другом каталоге или на другом *носителе данных* без сохранения исходного каталога в памяти на прежнем месте. Каталог переносится вместе со всеми содержащимися в нем

подкаталогами. Ср. *копирование каталога.* П. к. выполняется с помощью стандартных средств *операционной системы.* Например, на *персональных компьютерах* П. к. удобно выполнять в *оболочке Norton Commander* с помощью *горячих клавиш* аналогично тому, как производится *перенос файла.* В *операционной системе Windows* П. к. удобно осуществлять с помощью *перетаскивания мышью* в программе *Проводник Windows*

ПЕРЕНОС СЛОВ [hyphenation]. Разбиение слова на части в соответствии с правилами морфологии конкретного языка, которые определяют, в каких случаях начальную часть слова можно оставить в конце одной строки текста, а конечную часть слова поместить в начале следующей строки текста. Современные *текстовые процессоры*, например, MS Word, поддерживают режим автоматического и ручного П. с. (рис. П.4)

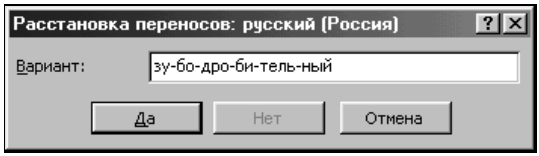

Рис. П.4. Диалоговое окно расстановки переносов текстового процессора Microsoft Word

ПЕРЕНОС ФАЙЛА, перемещение файла [file moving]. Процедура перезаписи *файла* из одного каталога в другой. Переносимый файл можно записать в иной каталог того же *носителя данных* или на другой носитель данных либо под тем же именем, либо изменив имя. При этом исходный файл не сохраняется в памяти. Ср. *копирование файла.* Процедуру П. ф. можно выполнять сразу над группой файлов. Для этого группу переносимых файлов надо предварительно отметить (см. *отметка файлов и каталогов*). П. ф. выполняется с помощью стандартных средств *операционной системы.* Например, на *персональных компьютерах* П. ф. удобно выполнять в *оболочке Norton Commander* с помощью *горячих клавиш.* В *операционной системе Windows* П. ф. удобно выполнять с помощью *перетаскивания мышью* в программе *Проводник Windows.* Ср. *копирование файла*

ПЕРЕПОЛНЕНИЕ [overflow, overrun]. 1. Ситуация, при которой результат операции превышает по абсолютной величине наибольшую представимую в *вычислительной системе* величину. Обычно П. приводит к *аварийному завершению* задачи. См. *диапазон изменения чисел.* 2. В *передаче данных* — ситуация, при которой устройство, принимающее данные, не может обрабатывать их так же быстро, как они поступают

ПЕРЕТАСКИВАНИЕ МЫШЬЮ, буксировка мышью [drag and drop]. Прием *графического интерфейса пользователя.* П. м. — это комплексное действие,

которое состоит из трех неразрывных этапов. Сначала нужно подвести *указатель мыши* к перетаскиваемому объекту на экране и нажать левую кнопку мыши. На жаргоне это называется "зацепить объект мышью". Как правило, при этом указатель мыши меняет свою форму, чтобы пользователь мог видеть, что зацепление произошло. На этом заканчивается первый этап. Затем нужно переместить указатель мыши в другое место на экране, удерживая нажатой кнопку мыши. При этом обычно изображение перетаскиваемого объекта (или иной символ) перемещается по экрану вместе с указателем мыши, чтобы пользователь мог видеть, что и куда он перетаскивает. Это второй этап — буксировка (drag). Наконец, следует отпустить кнопку мыши, как бы роняя перетаскиваемый объект в новом месте (drop). В этот момент и будет фактически выполнена операция, определяемая перетаскиванием. В зависимости от того, что и куда перетаскивается, с помощью П. м. можно инициировать выполнение самых разнообразных функций, например, *копирование* и *перенос файлов*. Кроме того, выполняемое действие может зависеть от того, была ли нажата одна из клавиш <Ctrl>, <Alt> или <Shift> при П. м. Например, в программе Проводник операционной системы Windows П. м. значка файла соответствует переносу файла, а если при этом нажата клавиша <Ctrl> — копированию файла

ПЕРЕХОД 1. **[branching, jump]**. Изменение естественной последовательности выполнения операторов в программе. Необходимость П. обуславливается *ветвлением программы*. П. задается с помощью *операторов* П. Термин "П." обычно употребляется, если речь идет о программировании на языках высокого уровня. В *машинных языках* чаще применяют эквивалентный термин *"передача управления"*. 2. **[transition]**. В *конечном автомате* — отношение между двумя состояниями, показывающее, что объект, находящийся в первом состоянии, будет выполнять некоторые *действия* и перейдет в другое состояние, как только наступит некоторое *переключающее событие*, и при этом будут удовлетворены необходимые *сторожевые условия*

ПЕРЕХОДНИК [thunk]. *Программный интерфейс*, преобразующий 32-разрядные данные и вызовы *системных функций* в 16-разрядные аналоги и наоборот. Используется для обеспечения совместимости с приложениями *операционной системы Windows* 3.*x*

ПЕРЕХОД ПО ЗАВЕРШЕНИИ [completion transition]. Переход в *конечном автомате*, у которого нет явного *переключающего события* и который поэтому запускается по окончании деятельности в исходном состоянии. В *унифицированном языке моделирования* UML применяется в *графах деятельности*

ПЕРЕЧИСЛЯЕМЫЙ ТИП, **перечислимый тип [enumerated type, ordinal type]**. *Тип данных*, определяемый путем перечисления принадлежащих ему значений, каждое из которых задается с помощью уникального имени константы. Множество значений П. т. является упорядоченным и описывается именами

констант. Примерами описаний П. т. могут служить следующие *описания типа* на языке Паскаль:

```
type Color = (White, Red, Orange, Yellow, Green, Blue, Violet, Black);
type Month = (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec);
```

При этом константам ставятся в соответствие порядковые номера, определяемые очередностью их записи при описании типа. Поэтому на множестве значений любого П. т. могут быть определены *операции сравнения*, например, `White < Red < Orange` или `Mar > Feb > Jan`. Ср. *порядковый тип*

ПЕРИФЕРИЙНОЕ УСТРОЙСТВО [peripheral device]. *Устройство*, подключаемое к основному блоку компьютера посредством кабеля или проводных *линий связи*. Как правило, П. у. имеет собственный блок управления и выполняет задания *центрального процессора*, не требуя его постоянного вмешательства. Для соединения П. у. с компьютером оно имеет так называемый порт, или многоразрядные вход и выход. Примерами П. у. могут служить принтер, сканер, мышь, стример, *внешние запоминающие устройства*

ПЕРО [stylus]. Элемент *указательного устройства*, похожий на ручку или карандаш. Используется в *графических планшетах*. Обычно соединяется с планшетом проводом. Провод может отсутствовать, если планшет сенсорный. Касаясь острием П. поверхности планшета, можно на экране дисплея перемещать курсор (аналог *указателя мыши*) или вычерчивать линии. Выбор на экране *элементов управления* производится перемещением курсора и нажатием кнопки на П., либо нажатием П. на поверхность планшета. См. *световое перо*

ПЕРСИСТЕНТНЫЙ ОБЪЕКТ [persistent object]. Объект, который продолжает свое существование даже после завершения создавшего его процесса

ПЕРСОНАЛЬНАЯ ЭВМ (ПЭВМ) [personal computer (PC)]. То же, что *персональный компьютер*

ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР, персональная ЭВМ (ПК, ПЭВМ) [personal computer (PC)]. ЭВМ универсального назначения, рассчитанная на одного пользователя и управляемая одним человеком. В класс ПК входят различные ЭВМ, например, дешевые игровые приставки с бытовым телевизором в качестве дисплея, *мобильные компьютеры* и *настольные компьютеры*

ПЕРСОНАЛЬНЫЙ IBM-СОВМЕСТИМЫЙ КОМПЬЮТЕР [personal IBM compatible computer (IBM PC)]. *Персональный компьютер*, на котором возможно использование программ для персональных компьютеров производства фирмы IBM. Все IBM PC можно условно разделить на три группы: 1) собственно IBM-совместимые компьютеры настольного исполнения — наиболее многочисленная группа компьютеров, совместимых между собой программно и аппаратно (т. е. в двух компьютерах этой группы с одинако-

выми техническими характеристиками возможна перестановка отдельных плат и устройств из одного компьютера в другой и наоборот). К этой группе можно отнести и компьютеры семейства PS-1 фирмы IBM, и компьютеры фирм Compaq, Dell и др., лишь частично аппаратно совместимые с продукцией большинства производителей; 2) *портативные компьютеры* — компьютеры классов laptop (наколенный компьютер) и notebook (блокнот), легкие, малогабаритные и способные работать от автономного источника питания; 3) настольные компьютеры семейства PS-2 фирмы IBM, аппаратно не совместимые с компьютерами других фирм. В других ситуациях в настоящее время этот термин не используется

ПЕРСПЕКТИВА [perspective]. *Команда графического редактора*, позволяющая создать у рисунка или выделенной области искажения, характерные для пространственной перспективы

ПЕРФОКАРТА, перфорационная карта [card, punch card]. *Носитель данных* в виде карты прямоугольной формы из тонкого картона. Данные наносятся на П. в *двоичном коде* путем пробивки сквозных отверстий — перфораций. Наличие отверстия означает 1, а его отсутствие — 0. Стандартная П. имеет 80 колонок по 12 позиций для пробивки в каждой. Каждая колонка содержит код одного символа. В настоящее время П. используются редко

ПЕРФОРАЦИОННАЯ КАРТА [card, punch card]. То же, что *перфокарта*

ПЕЧАТАЮЩЕЕ УСТРОЙСТВО [printer]. То же, что *принтер*

ПЗУ [ROM]. То же, что *постоянное запоминающее устройство*

ПИКОВАЯ ПРОИЗВОДИТЕЛЬНОСТЬ КОМПЬЮТЕРА, теоретическая производительность компьютера [computer peak performance]. Максимально возможная *производительность компьютера.* Эта величина вычисляется в предположении, что все *функциональные устройства* компьютера работают в максимально производительном режиме, а связи между функциональными устройствами (и, соответственно, потери на этих связях) отсутствуют. Не учитывается и загруженность функциональных устройств в процессе совместной работы. Так, если в процессоре есть два конвейерных устройства, то рассматривается режим, когда оба конвейера работают одновременно и с максимальной нагрузкой. Если в компьютере есть 100 таких процессоров, то пиковая производительность одного процессора просто умножается на 100

ПИКСЕЛ [pixel]. Элемент *графической информации.* Минимальный участок изображения, которому независимым образом можно задать цвет, яркость и другие характеристики. П. является элементом растра. Поэтому П. называются точки, на которые делится экран дисплея при *графическом режиме* работы, или точки, из которых формируется изображение на *струйных* или *лазерных принтерах.* См. *растр*

ПИКТОГРАММА [icon]. То же, что *значок*

ПИКТОГРАФИЧЕСКОЕ МЕНЮ [icon menu, key menu]. *Меню*, пункты которого изображаются на экране дисплея в виде *кнопок* с нарисованными на них *пиктограммами*. П. м. наряду с обычными меню и *горячими клавишами* широко применяются в качестве элементов *пользовательского интерфейса программных продуктов*, ориентированных на работу в среде Windows. П. м. предоставляет пользователю набор возможностей программного продукта. С каждой кнопкой связана некоторая команда, операция или вызов *инструментальных программных средств*, а рисунок на этой кнопке передает суть команды. Некоторые кнопки П. м. могут дублировать наиболее часто используемые команды, доступные в обычных меню, т. к. вызов команды с помощью кнопки гораздо быстрее, чем выбор в меню. Многие приложения, работающие в системе Windows, разрешают пользователю формировать собственные П. м. См. *панель инструментов*

ПИН-КОД, PIN-код [personal identification number]. Цифровой пароль. Уникальный набор цифр, предназначенный для идентификации пользователя при доступе к некоторой услуге

ПИПЕТКА [eye dropper]. *Инструмент машинной графики*, предназначенный для переноса цвета из рисунка в поля образцов основного и фонового цветов и для определения параметров цвета любого пиксела.

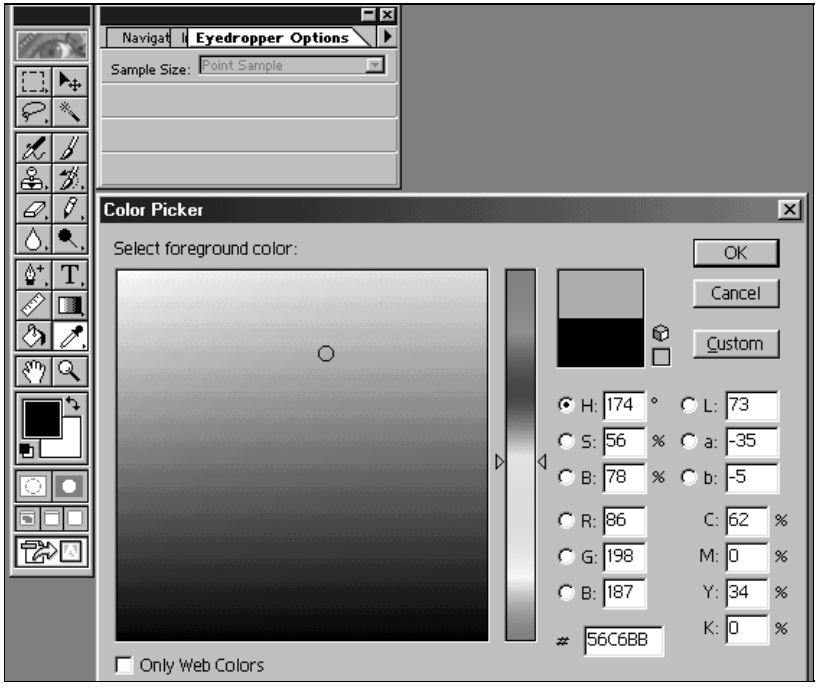

Рис. П.5. Кнопка вызова и диалоговые окна инструмента **Пипетка** графического редактора Adobe Photoshop

На рис. П.5 для *графического редактора* Adobe Photoshop показаны кнопка вызова П. и диалоговые окна, позволяющие установить площадь охвата П. (например, 3×3 пиксела) и определить параметры текущего цвета в двух *цветовых моделях*

ПИРАТСТВО [piracy]. То же, что *незаконное копирование программных средств*

ПК [personal computer (PC)]. То же, что *персональный компьютер*

ПЛАВАТЕЛЬНАЯ ДОРОЖКА [swim lane]. Термин *унифицированного языка моделирования* UML, означающий условные части *графа деятельности*. Сами по себе П. д. не имеют определенного значения и являются формой графического комментария, адресованного читателю модели. Например, *диаграмму деятельности* некоторого процесса в организации можно разделить на П. д., чтобы показать, что разные деятельности выполняются в различных подразделениях организации (см. рис. Д.4)

ПЛАНИРОВАНИЕ [scheduling]. Планирование *операционной системой* очередности выполнения процессов в *вычислительной системе*, позволяющее оптимизировать использование ее ресурсов. Прежде всего речь идет о порядке предоставления различным процессам *центрального процессора*, *оперативной памяти* и *устройств ввода/вывода*. П. выполняется планировщиками — специальными программами, входящими в состав операционной системы. См. *планировщик*

ПЛАНИРОВЩИК [scheduler]. *Программа*, входящая в состав *операционной системы*, определяющая порядок предоставления некоторого общего ресурса, в первую очередь *центрального процессора*, различным процессам. П. верхнего уровня определяет совокупность задач, выполняемых *вычислительной системой*. Он поддерживает двустороннюю связь с пользователем вычислительной системы, принимает от него запросы и команды, проверяет их и инициирует запрошенные действия. П. нижнего уровня, часто называемый *диспетчером*, определяет, какой задаче или какому ожидающему своей очереди процессу будет предоставлен ресурс (центральный процессор, *оперативная память*, *внешнее устройство*)

ПЛАНШЕТ ГРАФИЧЕСКИЙ [graphics tablet, digitizing tablet]. То же, что *графический планшет*

ПЛАТА [circuit board, circuit card]. Жесткая панель (обычно тонкий прямоугольный лист) из токоизолирующего материала, на которой монтируется *электронная схема*, выполняющая определенную функцию при работе компьютера. На одном из концов платы имеется соединитель, обеспечивающий все необходимые соединения с другими схемами компьютера. См. *звуковая плата, материнская плата*

ПЛАТА ИНТЕРФЕЙСА ISDN [ISDN interface card]. Выполненное в виде платы *устройство сопряжения*, предназначенное для подключения компью-

тера к *интегрированной цифровой сети связи*. По своим функциям П. и. ISDN сходна с модемом

ПЛАТФОРМА [computer platform, platform]. 1. То же, что *компьютерная платформа*. 2. То же, что *программная платформа*

ПЛАТФОРМНАЯ НЕЗАВИСИМОСТЬ, межплатформность [platform independence, cross-platform]. Свойство *программного обеспечения*, делающее его исполнимым на более чем одном типе компьютеров (компьютерной платформе), например, на IBM PC, Macintosh и т. д., или под управлением более чем одной *операционной системы* (программной платформы), например, под управлением Windows, UNIX и т. д.

ПЛОТНОСТЬ ЗАПИСИ [packing density]. Количество информации, приходящееся на единицу длины, площади или объема *носителя данных*. Например, одна из единиц измерения П. з. — число бит на дюйм. П. з. определяется физическими свойствами носителя и форматом данных. Она является важной характеристикой качества как носителя данных, так и всего *запоминающего устройства*, т. к. от П. з. зависит *информационная емкость*. Например, П. з. дискет размером 3.5″ значительно выше, чем у дискет размером 5.25″. Поэтому последние обладают меньшей информационной емкостью. Некоторые дисководы могут выполнять операции *форматирования диска*, читать и записывать данные с разной П. з.

ПЛОТТЕР [plotter]. То же, что *графопостроитель*

ПОВЕДЕНИЕ [behavior]. В *унифицированном языке моделирования* UML П. называются наблюдаемые эффекты выполнения операции или обработки события, в том числе получаемые при этом результаты и изменение состояния системы

ПОВТОРНАЯ НАЧАЛЬНАЯ ЗАГРУЗКА [reboot]. То же, что *перезагрузка*

ПОВТОРНО ВХОДИМЫЙ [reenterable]. См. *реентерабельность*

ПОВТОРНОЕ ИСПОЛЬЗОВАНИЕ [reuse]. Использование уже существующего артефакта

ПОВТОРНЫЙ ПУСК [rerun]. То же, что *перезапуск*

ПОДАЧА СТРАНИЦЫ [form feed (FF)]. То же, что *перевод страницы*

ПОДБЛОК [internal block]. То же, что *вложенный блок*

ПОДВОД ГОЛОВКИ [seek]. То же, что *поиск дорожки*

ПОДГОТОВКА ТЕКСТОВ [word processing]. Ввод и редактирование *текста* с помощью *текстового редактора* или *текстового процессора*

ПОДДЕРЖИВАТЬ [support]. Предоставлять необходимые средства и обеспечивать необходимые условия функционирования. Например, фраза: "*Много-*

задачная операционная система поддерживает одновременную работу на компьютере нескольких программ" означает, что операционная система предоставляет необходимые ресурсы и обеспечивает необходимые условия для одновременного выполнения на компьютере нескольких программ

ПОДДЕРЖКА [support]. 1. См. *поддерживать*. 2. То же, что *сопровождение программного изделия*

ПОДКАТАЛОГ [subdirectory]. См. *каталог*

ПОДКАЧКА, **свопинг [paging, swapping]**. Считывание в *оперативную память* страницы или сегмента *виртуальной памяти* или ранее сохраненного во *внешней памяти образа задачи*. В системах с виртуальной памятью П. страницы или сегмента может осуществляться либо по обращению к ним из программы, либо с упреждением, когда *операционная система* определяет, к каким страницам или сегментам наиболее вероятны обращения в ближайшее время, и подкачивает их. См. *свопинг, страница, страничный обмен*

ПОДКЛАСС [subclass]. Класс, который является *потомком* данного класса в *иерархии наследования*

ПОДМЕНЮ, вложенное меню [submenu]. *Меню*, вызываемое выбором *пункта меню* вышележащего уровня. Обычно пункты меню, в результате выбора которых раскрывается П., помечаются многоточием или стилизованной стрелкой. См. *нисходящее меню*

ПОДПРОГРАММА [subroutine]. Выделенная часть *программы*, реализующая определенный алгоритм и допускающая обращение из разных мест остальной части программы. Применение П. сокращает текст программы, если на разных этапах решения задачи требуется выполнить один и тот же алгоритм, т. к. при этом исключается необходимость многократного повторения в программе одних и тех же групп команд или операторов. Команды или операторы, задающие алгоритм П., выписывают один раз, а в нужных местах помещают *вызов* П. — команды либо оператор, осуществляющий переход к П. Например, если по ходу решения задачи необходимо несколько раз вычислить кубический корень, то рекомендуется написать П. извлечения кубического корня из числа и в нужных местах программы поместить вызов этой П., указав, из какого числа этот корень надо извлечь. Аналогично следует поступить, если в программе требуется найти корни нескольких квадратных уравнений. Здесь при вызове П. указываются коэффициенты конкретного уравнения. П., как часть программы, должна быть оформлена в соответствии с правилами *языка программирования* так, чтобы она в нужном месте вызывалась, получала *исходные данные* (например, в приведенных примерах ими являются число, из которого извлекается корень, или значения коэффициентов уравнения), выполняла запрограммированные действия и возвращала управление в нужное место *вызывающей программы*. Важным средством, облегчающим программирование, являются *стандартные* П.

В *языках программирования высокого уровня* различают два вида П.: процедуры, вызов которых имеет форму отдельного оператора (самостоятельного предложения) программы, и функции, обращения к которым являются составной частью *арифметического*, *логического* или другого *выражения*. См. *процедура, функция*

ПОДСИСТЕМА ПЕЧАТИ [printing subsystem]. Часть *операционной системы*, выполняющая функции, связанные с выводом данных на принтер. Ср. *видеосистема*

ПОДСКАЗКА [help, prompt]. 1. Дополнительная функция системы, состоящая в оказании пользователю помощи в виде сообщений, выводимых на экран дисплея (help). 2. Приглашение пользователю со стороны *операционной системы* на ввод команды или со стороны *диалоговой системы* на ввод данных. См. *приглашение*

ПОДТВЕРЖДЕНИЕ ПРИЕМА, сигнал подтверждения приема, сигнал ACK [acknowledgement (ACK)]. В сетях *передачи данных* — сообщение или сигнал, который получатель *пакета данных* передает отправителю для подтверждения того, что пакет получен полностью и в нем не обнаружено ошибок. В сети Интернет П. п. регламентировано протоколом TCP. П. п. увеличивают поток данных в сети, снижая скорость передачи, но повышая надежность

ПОДТВЕРЖДЕННАЯ ТРАНЗАКЦИЯ [committed transaction]. *Транзакция*, о которой в журнале транзакций сделана запись, что она завершена и записана в *кэш-память*

ПОДУРОВЕНЬ LLC [logical link control (LLC)]. То же, что *управление логической связью*

ПОДУРОВЕНЬ MAC [media access control (MAC)]. То же, что *управление доступом к среде*

ПОДЧЕРКИВАНИЕ [underline]. Операция *текстового редактора*, добавляющая черту под текстом. Некоторые *текстовые процессоры* позволяют выполнять П. двойными, пунктирными и волнистыми линиями

ПОЗИЦИОННАЯ СИСТЕМА СЧИСЛЕНИЯ [positional number system]. *Система счисления*, в которой значение каждой цифры числа зависит от занимаемого ею места — позиции и определяется этим местом. Например, в десятичной системе счисления число 55.5 трижды содержит одну и ту же цифру. Однако первая цифра обозначает 5 десятков, т. к. занимает позицию десятков, вторая — 5 единиц, поскольку занимает позицию единиц, а третья — 5 десятых, из-за того, что занимает позицию десятых долей. Любое число R в П. с. с. с основанием q имеет вид

$$R = (\pm a_m a_{m-1} \ldots a_1 a_0 \,.\, a_{-1} a_{-2} \ldots a_{-n})_q,$$

где ± — знак числа (отсутствие знака обычно обозначает плюс), a_m, a_{m-1}, ... — цифры, "." — точка (или запятая), разделяющая число на целую и дробную части. Скобки и основание q могут быть опущены, если заранее известно, в какой системе счисления написано данное число. Приведенная запись означает, что

$$R = \pm\,(a_m q^m + a_{m-1}q^{m-1} + ... + a_1 q + a_0 + a_{-1}q^{-1} + a_{-2}q^{-2} + ... + a_{-n}\,q^{-n})$$

$(-273.45)_{10} = -(2\cdot 10^2 + 7\cdot 10^1 + 3 + 4\cdot 10^{-1} + 5\cdot 10^{-2})$. П. с. с. с основанием q содержит ровно q цифр. Например, в *десятичной системе счисления* применяются цифры 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. См. *восьмеричная, двоичная, десятичная, шестнадцатеричная системы счисления*

ПОИСК В ОПРЕДЕЛЕННЫХ ПОЛЯХ [search by field]. Некоторые *поисковые системы* позволяют производить поиск не по всему *HTML-документу*, а выборочно, по определенным полям. Например, можно искать по заглавию документа (title), в этом случае учитываются только *ключевые слова*, встречающиеся именно в заглавии документа. См. *поисковая система*. Ср. *концептуальный поиск, поиск по ключевым словам, морфологический поиск*

ПОИСК ДАННЫХ [data search]. Процесс отыскания *данных* в памяти компьютера при условии, что в тот момент, когда эти данные необходимы для обработки, их адрес неизвестен. П. д. осуществляется автоматически *операционной системой* или *программными средствами информационной системы* по запросам пользователей. Для П. д. задаются один или несколько критериев, которым должны удовлетворять искомые данные. Например, *поиск файла* может производиться по шаблону имени файла, поиск записи в файле — по начальным знакам записи, а поиск числовых данных — по маске, определяющей формат и допустимые значения данных. Существуют различные алгоритмы П. д. Они зависят от характера искомых данных, критериев поиска и т. п.

ПОИСК ДОКУМЕНТА [document retrieval]. Процесс отыскания нужного *документа* в памяти компьютера. Некоторые приложения Windows выполняют автоматический П. д. по информации, задаваемой пользователем. Такой информацией могут служить заголовок документа, фамилия автора, дата создания, заранее оговоренные ключевые слова и т. п.

ПОИСК ДОРОЖКИ, подвод головки [seek]. Процесс перемещения *головки записи/чтения* в дисководе для установки к указанной *дорожке* диска при выполнении операции записи или чтения

ПОИСКОВАЯ СИСТЕМА [search engine]. Система, выполняющая автоматический поиск информации по ключевым словам, темам и т. д. П. с., осуществляющая поиск данных в WWW, содержит в себе три компонента: 1) *робот поисковой системы*, который исследует *веб-сайты* и заносит страницы в индекс системы; 2) *индексы поисковой системы*, где хранятся преобра-

зованные особым образом текстовые составляющие всех посещенных и проиндексированных роботом *HTML-страниц* и *текстовых файлов*; 3) собственно программу поиска, которая, получив запрос пользователя, находит в индексе документы, отвечающие критериям запроса, и выводит список найденных документов в порядке убывания релевантности

ПОИСК ПО КЛЮЧЕВЫМ СЛОВАМ [keyword search]. Поиск документов, которые содержат указанные пользователем *ключевые слова*. См. *поисковая система*. Ср. *концептуальный поиск, поиск в определенных полях, морфологический поиск*

ПОИСК С ВОЗВРАТАМИ, обратное прослеживание [backtracking]. Метод решения переборных задач в искусственном интеллекте, основанный на следующей идее. Находясь в исходной ситуации, пробуем изменить ее допустимым образом в надежде найти решение. Если изменение не привело к успеху, то возвращаемся в исходную ситуацию (отсюда и название метода "поиск с возвратами") и пробуем изменить ее другим образом, и так до тех пор, пока не будут исчерпаны все возможности

ПОИСК ФАЙЛА [file search]. Процесс отыскания нужного *файла* в памяти компьютера. П. ф. выполняется с помощью стандартных средств *операционной системы*. На *персональных компьютерах* его удобно производить в *оболочке Norton Commander* (или подобной ей) с помощью *горячих клавиш*. NC предоставляет возможность П. ф. по *шаблону имени файла* и по отрывку содержащегося в нем текста. Например, чтобы найти на диске все файлы с расширением txt, необходимо при нажатой клавише <Alt> нажать клавишу <F7>. В поле ввода **File(s) to find:** (Файл(ы) для поиска) появившегося окна поиска ввести шаблон Readme.*, подтвердить опцию, нажав клавишу <Enter>. В результате в окне поиска появится перечень найденных файлов. С помощью *цветового маркера* можно выбрать среди них нужный файл и нажатием клавиши <Enter> сделать его текущим на панели NC. В операционной системе Windows П. ф. выполняется с помощью утилиты программы *Проводник Windows*, диалоговое окно которой показано на рис. П.6

ПОКОЛЕНИЯ ЭВМ [computer generations]. Классы в классификации *вычислительных систем* по степени развития *аппаратных и программных средств*. П. ЭВМ определяются *элементной базой*, архитектурой и производительностью. В настоящее время различают пять П. ЭВМ. Характерными чертами первого П. ЭВМ, существовавшего в 50—60-е гг., были применение ламповой техники, быстродействие до нескольких десятков тысяч операций в секунду (оп./сек), программирование преимущественно *машинными командами* и работа программиста за пультом. Для второго П. ЭВМ (60—70-е гг.) было характерно применение полупроводниковой техники, быстродействие до 1 млн оп./сек, программирование на *языках высокого уровня*, наличие *операционных систем* и *пакетная обработка* заданий. Третье П. ЭВМ (80-е гг.) отличалось применением больших *интегральных схем*, быстродействием до не-

скольких сот млн оп./сек, развитыми системами *программного обеспечения* и *диалоговым режимом* общения программиста с вычислительной системой. Для четвертого П. ЭВМ (в настоящее время) характерны применение сверхбольших интегральных схем (СБИС) и микропроцессорной техники, быстродействие свыше миллиарда оп./сек, многомашинные и многопроцессорные комплексы в сочетании с *персональными компьютерами*, широкое применение СУБД, объектно-ориентированного программирования, *компьютерных сетей* и *сетей компьютерной связи*. В настоящее время интенсивно создается пятое П. ЭВМ, от которого ожидаются разработка и применение систем с *искусственным интеллектом* и систем решения задач на основе информации, хранящейся в *базе знаний*

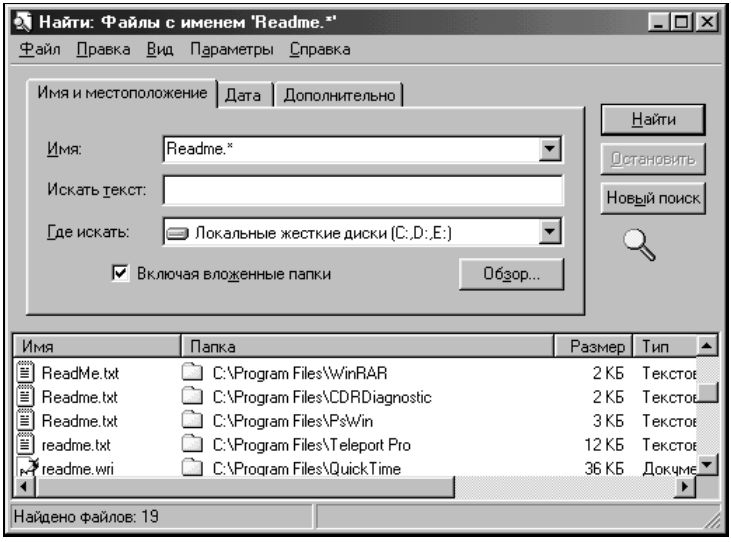

Рис. П.6. Диалоговое окно для поиска файлов

ПОЛЕ [field]. 1. Участок памяти или экрана дисплея. 2. То же, что *поле данных*. 3. То же, что *текстовое поле*

ПОЛЕ ВВОДА [input field]. Часть окна на экране дисплея, предназначенная для ввода текста. Обычно имеет вид выделенной пустой или частично заполненной строки, на которой располагается мигающий курсор. Курсор указывает текущую позицию П. в., в которую будет произведен ввод символа при нажатии символьно-цифровой или буквенной клавиши. Ср. *текстовое поле*

ПОЛЕ ДАННЫХ, поле [data field, field]. Часть записи, команды или заполняемой формы, имеющая самостоятельное значение. П. д. представляет не-

сколько смежных позиций (бит, символов), выделяемых заданием формата, и обрабатывается компьютером или воспринимается пользователем как отдельный *элемент данных*. В одной записи файла могут размещаться несколько элементарных данных, занимающих различные П. д., каждое из которых выделяется форматом записи, например, в одной записи файла — списка учеников могут присутствовать поля фамилии и инициалов, домашнего адреса, телефона и т. п. *Форматом команды* выделяются поле *кода операции*, содержащее код выполняемой процессором операции, и поле адреса, в котором размещается адрес операнда. А форматом *числа с плавающей точкой* выделяются поле мантиссы и поле порядка

ПОЛЕ СО СПИСКОМ [combo box]. *Элемент управления*, предназначенный для ввода, отображения и редактирования значений, а также для выбора одного или нескольких значений из списка. Является комбинацией элементов управления *поле* и *список*

ПОЛЗУНОК [scroll box]. То же, что *бегунок*

ПОЛИМОРФНАЯ ОПЕРАЦИЯ [polymorphic operation]. Термин *объектно-ориентированного программирования*, который обозначает такую операцию обобщающего класса, которая может иметь различные реализации в обобщаемых классах

ПОЛИТИКА [policy]. 1. Набор правил или комплекс алгоритмов, в соответствии с которыми осуществляется управление вычислительными системами, сетями и т. п. См. *политика учетных записей*. 2. Программная реализация правил и алгоритмов управления вычислительными системами, сетями и т. п.

ПОЛИТИКА БЕЗОПАСНОСТИ [security policy]. Принятая в конкретной организации *политика* управления получением, обработкой, передачей секретной информации, или информации, рассчитанной на ограниченный круг пользователей. См. *защита данных*

ПОЛИТИКА УЧЕТНЫХ ЗАПИСЕЙ [account policy]. Набор параметров, который определяет способ *входа в систему* или *права доступа* для всех или некоторых пользователей *локальной сети* или *вычислительной системы*. Позволяет установить минимальную длину *пароля*, периодичность смены пользовательских паролей и возможность повторного использования паролей. П. у. з. может быть установлена *системным администратором* для всех *учетных записей пользователей*

ПОЛНОЕ ИМЯ ФАЙЛА [pathname]. Составное имя, включающее последовательно записываемые через разделитель "\" имя диска, последовательность имен *вложенных каталогов*, через которые проходит путь к файлу (начиная с высшего в их иерархии), и имя файла с расширением. Таким образом,

П. и. ф. состоит из пути поиска файла и имени с расширением. Например, в П. и. ф. C:\SYS\DRV\800.com путем является C:\SYS\DRV

ПОЛНОМОЧИЕ [authorization, right]. Право пользоваться *компьютерной сетью*, *вычислительной системой*, программой или защищенными данными. Необходимость в получении П. возникает, например, для доступа к вычислительной системе и хранящимся в ней файлам в случае, если круг пользователей ограничен. Обычно П. даются пользователю администратором системы в виде пароля, который необходимо ввести по запросу компьютера, чтобы получить доступ к системе, программе или данным. См. *защита данных, несанкционированный доступ, пароль*

ПОЛНОТЕКСТОВАЯ ПОИСКОВАЯ СИСТЕМА [full-text search engine]. *Поисковая система*, которая индексирует все слова на веб-странице и учитывает порядок их расположения

ПОЛНЫЙ ЭКРАН [full screen view]. Режим, при котором на экране отображается только документ и не отображаются *панели инструментов*, *строка меню* и прочие элементы *интерфейса пользователя*

ПОЛОСА ПРОКРУТКИ [scroll bar]. То же, что *линейка прокрутки*

ПОЛУЖИРНЫЙ ШРИФТ [bold, boldface]. *Начертание шрифта*, при котором для изображения символов используются более толстые линии, а ширина символов несколько увеличена. В данном словаре П. ш. выделены, например, заголовки словарных статей, имена диалоговых окон, полей ввода и других элементов графического интерфейса. Ср. *курсив*

ПОЛУЛОГАРИФМИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ ЧИСЕЛ [floating-point number representation]. Способ представления *числа с плавающей точкой* в виде двух компонентов (P, M), где P — *порядок числа*, M — его мантисса. Обычно порядок выбирается так, чтобы $|M|<1$, и при записи опускаются нулевая целая часть и точка мантиссы. Например, $0.123 \cdot 10^{-5}$ имеет П. п. ч. (−5, 123). П. п. ч. применяется в ЭВМ, т. к. обеспечивает высокую точность в широком *диапазоне изменения чисел* при небольшом количестве цифр. См. *представление чисел с плавающей точкой*

ПОЛЬЗОВАТЕЛЬ [client, user]. 1. Человек или организация, пользующиеся *вычислительной системой* или *программным средством*. 2. Программа, использующая данный *программный модуль*

ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС [user interface]. То же, что *интерфейс пользователя*

ПОМЕЧЕННЫЙ ОПЕРАТОР [labeled statement]. *Оператор* в программе, снабженный *меткой*

ПОРТ [port]. 1. *Устройство сопряжения*, с помощью которого *центральный процессор* или *оперативная память* компьютера могут быть связаны с други-

ми устройствами с целью *передачи данных*. Например, через П. подключаются к шине процессора *устройства ввода/вывода*, а программа может посылать данные в П. или получать данные из П. Обычно один и тот же П. может работать на ввод или вывод. По способу передачи данных различают *параллельный* и *последовательный порты*. 2. В сети Интернет — число, которое идентифицирует (определяет) конкретную прикладную программу Интернета

ПОРТАБИЛЬНОСТЬ ПРОГРАММЫ [program portability]. То же, что *переносимость программы*

ПОРТАЛ, **веб-портал**, **Web-портал [portal**, **Web portal]**. *Веб-сайт*, предоставляющий пользователям комплекс услуг сети Интернет, среди которых: *поисковая система*, предлагающая широкий выбор тем для поиска в World Wide Web; *электронная почта*, служба разговоров (чаты); справочная служба, позволяющая получить, например, метеопрогноз, котировку валют и т. п.; индивидуальная подписка на новости по различной тематике; форум пользователей; *баннерная реклама* и т. д. Обычно П. в той или иной степени отражает круг интересов его авторов или владельцев и собирает "вокруг себя" более или менее постоянное множество пользователей, для которых он служит как бы входом во Всемирную паутину. На рис. П.7 показана веб-страница П. Яндекс

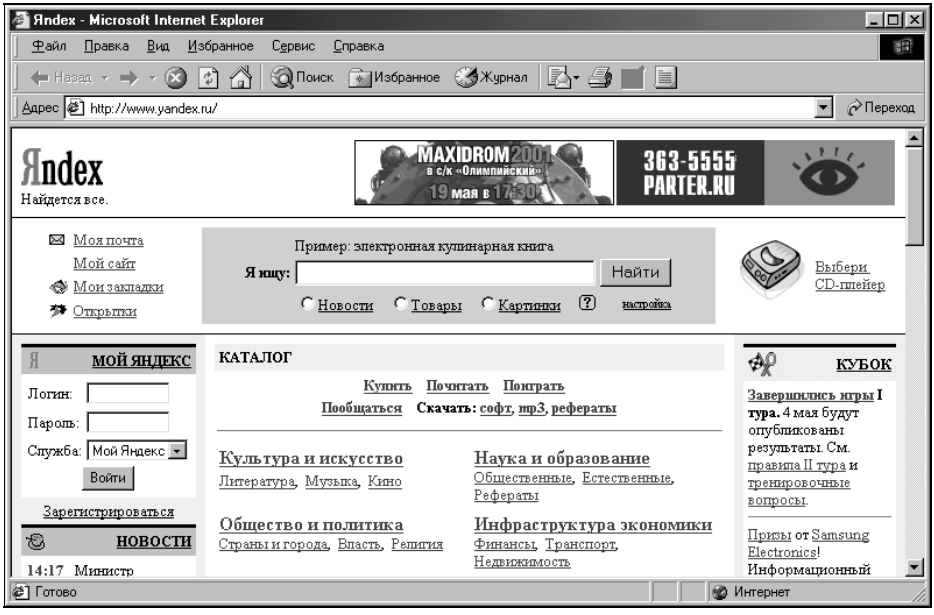

Рис. П.7. Портал Яндекс

ПОРТАТИВНЫЙ КОМПЬЮТЕР [portable computer]. Малогабаритный переносной *персональный компьютер*, способный работать от автономного источника питания. Современные П. к., как правило, ни в чем не уступают настольным. Они имеют сравнимые с ними по характеристикам процессор и *оперативную память*, плоский дисплей на жидких кристаллах, клавиатуру, энергонезависимую память и средства подключения *внешних устройств*. Различают П. к. классов laptop (дорожный компьютер) и notebook (блокнот). К П. к. относятся и т. н. пен-компьютеры (pen computer — перьевой компьютер). Это электронные записные книжки, в которых первичным устройством ввода является перо, а не клавиатура. Ср. *мобильный компьютер*

ПОРТРЕТНАЯ ОРИЕНТАЦИЯ СТРАНИЦЫ [portrait page orientation]. То же, что *книжная ориентация страницы*

ПОРЯДКОВЫЙ ТИП [serial number type]. Обобщающее название для *типа данных*, каждому значению которого ставится во взаимно однозначное соответствие некоторый порядковый номер. К П. т. относятся стандартные типы: *целый тип* (каждое целое число может рассматриваться, как порядковый номер), *символьный тип* (код символа может рассматриваться, как порядковый номер) и *логический тип* с двумя порядковыми номерами 0 и 1. Также к П. т. относится любой *перечисляемый тип*

ПОРЯДОК УЗЛА [node order**]**. Количество узлов сети, с которыми данный узел имеет прямое соединение

ПОРЯДОК ЧИСЛА [exponent]. Часть *числа с плавающей точкой*, определяющая значение степени, в которую должно быть возведено основание *системы счисления*, чтобы при умножении на мантиссу получить истинное значение числа. Например, порядок числа $0.123 \cdot 10^{-5}$ равен −5, порядок числа +3.45E+2 равен 2

ПОСЕЩАЕМОСТЬ [traffic]. Количество посетителей *веб-сайта* или *веб-страницы* за единицу времени (день, месяц и т. д.)

ПОСЕЩЕНИЕ [hit]. Обращение к *веб-сайту* или *веб-странице*. Число П. служит показателем популярности страницы. Ср. *хит*

ПОСЛЕДОВАТЕЛЬНЫЙ ДОСТУП [sequential access, serial access]. 1. Способ *доступа* к данным, при котором записи файла считываются в том же порядке, в котором они были записаны при его создании. Такая логическая последовательность записей вовсе не означает, что и физически записи расположены последовательно, например, для файла, размещенного на *магнитном диске*. Впрочем, часто логическая и физическая последовательности совпадают. 2. Способ *доступа*, при котором данные считываются в *оперативную память* в порядке их физического размещения на *носителе данных внешнего запоминающего устройства*. При этом для обращения к нужной записи необходимо последовательно "просмотреть" все участки носителя, на-

чиная либо с самого первого, либо со следующего за тем, к которому было предыдущее обращение. Такой способ доступа осуществляется, например, к данным, хранящимся на *магнитных лентах*. Ср. *прямой доступ*

ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ [serial port]. *Порт*, передающий данные только последовательно, бит за битом. Ср. *параллельный порт*

ПОСЛЕДОВАТЕЛЬНЫЙ ФАЙЛ [sequential file]. То же, что *файл последовательного доступа*

ПОСТАВЩИК СЕТЕВЫХ УСЛУГ, провайдер [service provider, provider]. Лицо или организация, предоставляющая услуги по подключению к *компьютерным сетям*. Например, интернет-провайдер предоставляет следующие, как правило, платные, услуги по подключению и доступу в Интернет: 1) снабжает пользователя именем, паролем, *каналом связи* или номером телефона для соединения с Интернетом; 2) помогает настроить сетевое *программное обеспечение* и модем на компьютере пользователя. Многие П. с. у. за дополнительную плату обеспечивают аренду пользователем *области памяти* на своем интернет-сервере для размещения *почтового ящика* или публикации *веб-документов* и помогают в их создании. Перечень и качество таких услуг зависят от конкретного поставщика. Кроме того, у П. с. у. можно получить профессиональную консультацию по вопросам, связанным с пользованием сетью

ПОСТОЯННОЕ ЗАПОМИНАЮЩЕЕ УСТРОЙСТВО (ПЗУ) [read only memory (ROM)]. Память, предназначенная только для чтения. Специальное *запоминающее устройство*, в которое при его изготовлении навсегда записаны определенные неизменяемые данные. Эти данные в последующем невозможно ни стереть, ни изменить, их можно только считывать, т. к. ПЗУ не способно выполнять операцию записи. Такая память обычно реализуется на микросхемах или *лазерных дисках* (CD-ROM) и предназначена для хранения программ и данных, необходимых для работы компьютера

ПОСТПРОЦЕССОР [back-end processor, postprocessor]. 1. *Сопроцессор*, выполняющий дополнительную обработку результатов работы основного процессора или осуществляющий некоторую вспомогательную функцию, освобождающую основной процессор для другой работы. Например, П. является встроенный в *звуковую плату* сигнальный процессор, обрабатывающий оцифрованный звук в течение его воспроизведения. 2. *Машинная программа*, выполняющая дополнительную обработку результатов. Ср. *препроцессор*

ПОСТУСЛОВИЕ [post condition]. Ограничение, которое должно соблюдаться после завершения операции

ПОТОК [flow, stream, thread]. 1. То же, что *поток данных*. 2. То же, что *поток команд*. 3. То же, что *поток управления*

ПОТОК ДАННЫХ [data flow, dataflow, data stream]. Логическая последовательность данных, обрабатываемых программой. Этот термин широко применяется в *параллельных вычислениях*, в которых П. д. определяет последовательность операций, а именно: выполнение каждой операции происходит при готовности всех ее операндов. Если готовы данные для нескольких операций, они выполняются параллельно. Ср. *поток управления*

ПОТОК КОМАНД [instruction stream]. Последовательность команд, обрабатываемая процессором. См. *поток управления*. Ср. *поток данных*

ПОТОК ОБЪЕКТОВ [object flow]. Отношение между объектом и деятельностью, которая либо создает объект (в качестве значения на выходе), либо использует его (в качестве значения на входе). Ср. *поток управления*

ПОТОК УПРАВЛЕНИЯ [control flow, thread]. Последовательность выполнения операторов в программе. Обычно операторы в программе на *языке программирования высокого уровня* (или команды в программе на *машинном языке*) выполняются по очереди в естественном порядке (от начала к концу). В этом случае П. у. просто совпадает с последовательностью операторов в программе. Однако в целом ряде случаев П. у. может не совпадать с последовательностью операторов в программе. Например, при выполнении оператора *вызова подпрограммы* выполнение операторов программы приостанавливается, управление передается в подпрограмму, т. е. в П. у. попадают операторы подпрограммы, а при *выходе* из подпрограммы возобновляется выполнение операторов программы. Аналогично, на П. у. оказывают влияние *операторы перехода*, *условные операторы*, *операторы цикла*. Кроме того, на П. у. влияют факторы, находящиеся вне данной программы, например, П. у. меняется при *обработке прерываний* и при возникновении *событий*. При *параллельной обработке* имеется несколько П. у.: каждый *параллельный вычислительный процесс* обладает своим П. у. Существуют *операционные системы*, позволяющие одному процессу иметь несколько П. у. См. *многопоточность,* ср. *поток команд*

ПОТОМОК [descendant]. См. *иерархия*

ПОЧТОВЫЙ ПРОТОКОЛ [post office protocol (POP)]. То же, что *протокол POP*

ПОЧТОВЫЙ ПРОТОКОЛ POP [post office protocol, POP]. *Протокол*, используемый при отправке и получении сообщений через *электронную почту* Интернета. Существуют различные версии этого протокола, в настоящее время применяется протокол POP3, позволяющий осуществлять динамический доступ в *почтовый ящик* с *рабочей станции*

ПОЧТОВЫЙ ЯЩИК [mailbox]. *Область памяти*, предназначенная для хранения сообщений, документов или данных, переданных по *электронной почте*. П. я. имеет свой адрес в системе электронной почты и может разме-

щаться на специальном компьютере — *узле сети* вместе с программой-сервером, предоставляющей почтовые услуги пользователям. Пользователь-адресат получает доступ к П. я. через свой терминал. См. *сервер*

ПО Windows Sockets [Windows Sockets (WinSoc)]. То же, что *программное обеспечение Windows Sockets*

ПРАВО ДОСТУПА [access right, access permission]. 1. Право, предоставленное пользователю на санкционированное использование определенных программ и данных, хранящихся в *вычислительной системе*. 2. Правила, определяющие, какие пользователи или процессы могут получать *доступ* к объекту (каталогу, файлу, принтеру и т. п.) и каким образом

ПРЕДЛОЖЕНИЕ [sentence, statement]. *Синтаксическая конструкция*, обладающая для данного *языка программирования* смысловой законченностью. Это — синтаксически правильная часть программы, определяющая единицу действия для транслятора или исполнителя программ. П. являются, например, команды *машинного языка* или автокода, описания *объектов программы* и операторы в *языках программирования высокого уровня*. Понятие П. обычно уточняется правилами каждого конкретного языка программирования

ПРЕДЛОЖЕНИЯ ЦИКЛА ЯЗЫКА ПАСКАЛЬ [Pascal loop sentences]. То же, что *операторы цикла языка Паскаль*

ПРЕДМЕТНАЯ ОБЛАСТЬ [domain]. Конкретная область применения *информационных технологий*. Например, здавоохранение, транспорт и т. п.

ПРЕДМЕТНО-ОРИЕНТРОВАННЫЙ [domain specific]. Ориентированный на применение в конкретной *предметной области*. Например, ГАС "Выборы" является П.-О. автоматизированной системой. Ср. *проблемно-ориентированный*

ПРЕДОК [ancestor]. См. *иерархия*

ПРЕДСТАВЛЕНИЕ ДАННЫХ [data representation]. Вид, форма, правила кодирования и образования конструкций данных в *памяти ЭВМ* или на другом конкретном уровне *вычислительной системы*. Например, *вещественные числа*, вводимые в систему с клавиатуры, представляются в *десятичной системе счисления*, в форме *чисел с фиксированной* или *плавающей точкой*, а их представление в памяти компьютера (система счисления, код и формат) определяются типом компьютера. См. *диапазон изменения чисел, представление чисел с плавающей точкой*

ПРЕДСТАВЛЕНИЕ ЧИСЕЛ С ПЛАВАЮЩЕЙ ТОЧКОЙ [floating-point representation]. Запись *чисел с плавающей точкой* в *памяти ЭВМ* в коде, и в формате, определяемых типом компьютера *(компьютерной платформой)*. Ниже приведены схемы размещения (форматы) чисел с плавающей точкой, которые различает *математический сопроцессор* серии 80x87.

а) Формат числа одинарной точности (4 байта)

1 бит	8 бит	23 бита
Знак	Порядок	Мантисса

б) Формат числа двойной точности (8 байтов)

1 бит	11 бит	52 бита
Знак	Порядок	Мантисса

в) Формат числа расширенной точности (10 байтов)

1 бит	15 бит	64 бита
Знак	Порядок	Мантисса

Они соответствуют числам одинарной, двойной и расширенной точности (часто называются также коротким вещественным, длинным вещественным и расширенным вещественным числами соответственно). Каждый формат состоит из трех полей: знак, порядок и мантисса. Содержимое каждого поля является *двоичным числом*, порядок определяет степень числа 2, а не 10. Числа в этих форматах занимают в памяти 4, 8 и 10 байтов соответственно. В языке Turbo Pascal эти форматы соответствуют вещественным типам `single`, `double` и `extended`. Следует заметить, что ради дополнительного бита точности числа одинарной и двойной точности хранятся без старшего бита мантиссы, который из-за используемой *нормализованной формы* всегда равен единице и поэтому опускается (этот бит называется неявным). Для чисел расширенной точности хранятся все биты мантиссы. Таким образом, размер мантиссы коротких вещественных чисел 24, а длинных — 53 бита. По умолчанию принят формат двойной точности. Одинарный формат применяется при ограниченной оперативной памяти. Формат расширенной точности применяется внутри сопроцессора для повышения точности результатов операций над числами. Гарантированные диапазоны изменения десятичных чисел с плавающей точкой приведены в табл. П.1. См. *диапазон изменения чисел*

Таблица П.1. *Гарантированное количество десятичных значащих цифр и диапазоны степени числа 10 для чисел с плавающей точкой математического сопроцессора серии 80x87*

Формат числа	Значащих цифр	Наименьшая степень числа 10	Наибольшая степень числа 10
Короткое	7	–37	38
Длинное	15	–307	308
Расширенное	19	–4931	4932

ПРЕДУПРЕЖДЕНИЕ [warning, alert, alert message]. Предупреждающее *сообщение* обрабатывающей программы. Оно может быть текстовым, графическим, сопровождаться звуковым или речевым сигналом. В английском языке слово *warning* — соответствует П. о незначительной ошибке в программе, при обнаружении которой выполнение программы может быть продолжено. Например, П. компилятора об имеющихся в программе объявленных, но не использованных переменных. Слова *alert*, *alert message* — соответствуют П. о серьезной ошибке, сбойной ситуации в системе или недопустимости операции, которую попытался выполнить пользователь

ПРЕДУСЛОВИЕ [precondition]. Ограничение, которое должно соблюдаться перед выполнением операции

ПРЕОБРАЗОВАНИЕ ОБЪЕКТОВ [object conversion]. Метод изменения формата *объекта* и его свойств. При этом объект, созданный в одном приложении, преобразуется в формат другого приложения. Например, при вставке в документ MS Word рисунка, подготовленного в графическом редакторе Microsoft Paint, рисунок можно вставить как объект Paintbrush, а можно выполнить П. о. и вставить как точечный рисунок (рис. П.8). При П. о. могут быть утрачены элементы, отвечающие за представление данных, но не сами данные

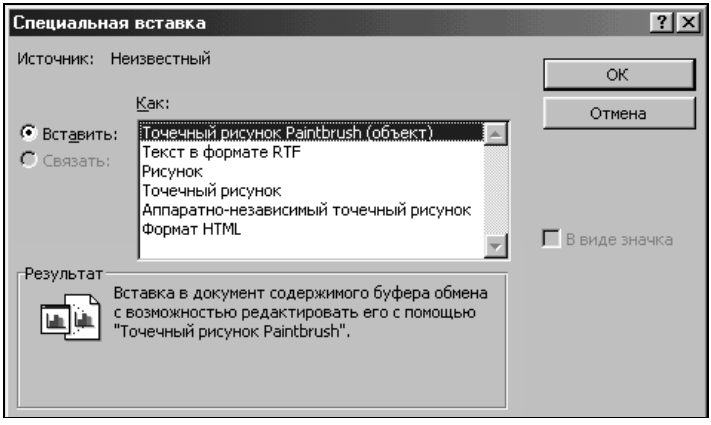

Рис. П.8. Диалоговое окно **Специальная вставка** позволяет выполнять преобразование объектов

ПРЕОБРАЗОВАНИЕ С ПРОМЕЖУТОЧНЫМИ ФОРМАМИ, твининг [tweening]. В графической программе: метод преобразования одного объекта изображения в другой через последовательность промежуточных форм. Например, пользователь может дать программе команду преобразовать квадрат в

круг. Тогда программа рассчитывает промежуточные формы (число которых иногда можно указать точно), необходимые для трансформации

ПРЕОБРАЗОВАНИЕ ТИПОВ [cast, coercion, type conversion]. То же, что *приведение типов*

ПРЕОБРАЗОВАТЕЛЬ, конвертор [converter]. 1. Устройство, преобразующее электрические сигналы из одной формы в другую. См. *аналого-цифровой преобразователь*, *цифроаналоговый преобразователь*. 2. Программа или устройство, преобразующее данные из одной формы представления в другую. По характеру выполняемых преобразований можно выделить следующие типы программ-П. 1) П., изменяющие формат файла. Например, программа, преобразующая файл типа "текст DOS" (файл с расширением txt) в формат документа Microsoft Word (файл с расширением doc). 2) П., изменяющие способ представления данных. Например, программа, выполняющая перевод чисел из *десятичной системы счисления* в двоичную. 3) Трансляторы с одного *языка программирования*, как правило, *высокого уровня*, на другой язык того же уровня. Сюда можно отнести П., изменяющий машинный код программы, работающей на одном компьютере, для переноса на другой компьютер

ПРЕПРОЦЕССОР [preprocessor]. 1. *Программа*, выполняющая предварительную обработку *входных данных* для другой программы. Предварительная обработка может заключаться, например, в простом изменении форматов данных с целью подготовки их для ввода в программу. Разновидностью П. является макропроцессор, осуществляющий макроподстановки перед обработкой программы компилятором. Наиболее известен П. языка Си, который просматривает программу до компилятора и заменяет в программе определенные сочетания символов (символические аббревиатуры) на соответствующие директивы. Он отыскивает и подключает к программе необходимые файлы и может изменить условия компиляции. См. *макропроцессор*. Ср. *постпроцессор*. 2. То же, что *буферный процессор*

ПРЕРЫВАНИЕ [interrupt]. 1. Временное прекращение выполнения команд программы с сохранением информации о ее *текущем состоянии* и передачей управления специальной программе — *обработчику прерываний*. Причинами П. могут быть события, происходящие в *технических устройствах* или программах. Например, П. может вызвать терминал, которому необходимо принять данные, сбой в работе *устройства ввода/вывода* или попытка деления на ноль при выполнении вычислений. Инициировать П. может пользователь, желающий начать взаимодействие с системой, или таймер по истечении заданного времени. После П. происходит его обработка. В зависимости от причины П. обработка может выполняться *аппаратными* или *программными средствами*. При этом сохраняется информация о текущем состоянии прерванной программы, фиксируются причина и условия прерывания и определяются дальнейшие действия. См. *обработка прерываний, программное прерывание, внешнее прерывание, внутреннее прерывание, прерывание от схем*

контроля. Ср. *завершение программы.* 2. Сигнал, вырабатываемый *аппаратными* или *программными средствами*, инициирующий временное прекращение выполнения команд программы

ПРЕРЫВАНИЕ ВВОДА/ВЫВОДА [input-output interrupt]. *Внешнее прерывание*, связанное с работой *устройства ввода/вывода.* См. *обработка прерываний*

ПРЕРЫВАНИЕ ОТ СХЕМ КОНТРОЛЯ, прерывание по машинному сбою [machine-check interrupt]. *Прерывание*, вызванное схемами контроля компьютера в случае любого *машинного сбоя*, например, такого как ошибка при *контроле по четности* или падение напряжения в сети

ПРЕРЫВАНИЕ ПО МАШИННОМУ СБОЮ [machine-check interrupt]. То же, что *прерывание от схем контроля*

ПРЕРЫВАНИЕ ПРИКЛАДНОЙ ПРОГРАММЫ [software interrupt]. То же, что *программное прерывание*

ПРЕФИКС [prefix]. 1. Приставка, начальная часть имени, указывающая на принадлежность к определенной программе либо на место расположения в памяти. Например, в иерархической *файловой системе* П. имени файла — часть *полного имени файла*, определяющая имя каталога, в котором он расположен. 2. Содержимое специального регистра, используемое при автоматическом вычислении *абсолютных адресов*

ПРИВЕДЕНИЕ ТИПОВ, преобразование типов [cast, coercion, type conversion]. Преобразование значения из формата, принятого для представления значений одного *типа данных,* в формат, принятый для представления значений другого типа данных. П. т. бывает явное, т. е. указанное программистом, и неявное, т. е. автоматически производимое транслятором языка программирования. В каждом *языке программирования высокого уровня* существуют свои правила неявного П. т. Так, компиляторы с языка Си не считают неправильной программу, в которой есть смешение типов в одном выражении. В этом случае используется набор правил для автоматического П. т. Основными из них являются следующие. 1) Если операция выполняется над данными двух различных типов, то обе величины приводятся к "высшему" из двух, и результат является величиной "высшего" типа. "Высшим" считается тип, диапазон изменения значений которого больше. Например, из целого (integer) и вещественного (float) типов "высший" — float. 2) В *операторе присваивания* результат вычисления выражения преобразуется к типу переменной, которой он должен быть присвоен. Кроме П. т., которые компилятор выполняет автоматически, в *языке программирования* Си введены специальные *стандартные функции* и операции, позволяющие в программе точно указать тип данных, к которому следует привести некоторую величину. В результате при выполнении фрагмента программы:

```
int i = 3; float f1, f2; f1 = i/2; f2 = ((float) i)/2;
```

переменная `f1` получит значение `1.0`, а переменная `f2` — значение `1.5`. Это происходит по следующей причине. В первом *операторе присваивания* как переменная `i`, так и константа `2` имеют значения *целого типа*, поэтому операция деления выполняется как деление нацело, т. е. с остатком, и дает в результате целое число `1`, которое автоматически приводится к значению *вещественного типа* `1.0` и присваивается переменной `f1`. Во втором операторе присваивания явно указано, что значение переменной `i` целого типа должно быть сначала преобразовано в значение вещественного типа, поэтому операция деления выполняется как деление вещественных чисел и дает в результате значение `1.5`, которое присваивается переменной `f2` без всякого преобразования

ПРИВИЛЕГИРОВАННАЯ КОМАНДА [privileged instruction]. *Команда* процессора, которая может быть вызвана к исполнению только *ядром операционной системы* и некоторыми *системными программами*. К таким командам относятся, например, команды управления памятью

ПРИВИЛЕГИРОВАННЫЙ РЕЖИМ, режим операционной системы, режим ядра [privileged mode, kernel mode]. Режим работы *центрального процессора*, при котором разрешены *привилегированные команды*. В П. р. исполняется код *ядра операционной системы*. При этом *поток управления* имеет доступ и к *системной памяти*, и к аппаратуре

ПРИВОД CD-ROM, накопитель на лазерных дисках [CD-ROM drive, CD drive]. *Устройство*, предназначенное для считывания информации с *компакт-диска*. Состоит из платы электроники, двигателя, системы оптической считывающей головки и системы загрузки диска. На плате электроники размещены все управляющие схемы привода. Двигатель служит для приведения диска во вращение с постоянной или переменной линейной скоростью. Сохранение постоянной линейной скорости требует изменения угловой скорости диска в зависимости от положения оптической головки. При поиске фрагментов диск может вращаться с большей скоростью, нежели при считывании. Система оптической головки позволяет считывать данные, несмотря на вертикальные биения диска, даже при значительных скоростях вращения. Загрузка компакт-диска *персонального компьютера* осуществляется при помощи выдвижного лотка, на который кладется сам диск. На передней панели привода обычно расположены кнопка для загрузки и выгрузки диска, индикатор обращения к приводу и гнездо для подключения наушников с регулятором громкости. При стандартной скорости вращения скорость передачи данных составляет около 150 Кбайт/сек. В двух- и более скоростных CD-ROM диск вращается с пропорционально большей скоростью, и пропорционально повышается скорость передачи данных (например, 1200 Кбайт/сек для 8-скоростного привода)

ПРИВРАТНИК [gatekeeper]. Узел *сети H.323*, управляющий группой других узлов. П. преобразует их телефонные номера в *IP-адреса* и отвечает за реги-

страцию узлов, тарификацию звонков, контроль прав доступа. Он также участвует в установлении соединений, фактически играя в сети Н.323 роль телефонной станции. См. *компьютерно-телефонная интеграция*

ПРИГЛАШЕНИЕ, подсказка [prompt]. Текст, значок или другое изображение на экране дисплея, указывающее, что программа или система ожидает ввода пользователем команд или данных. П. может содержать некоторую нужную пользователю информацию о текущем состоянии системы. Например, в П. *операционной системы MS-DOS*: C:\EXE> сообщается о *текущем дисководе* C: и *текущем каталоге* EXE

ПРИЕМНИК ДАННЫХ (СООБЩЕНИЙ) [data (message) sink]. Функциональное устройство, получающее передаваемые данные

ПРИКЛАДНАЯ ПРОГРАММА [application program]. 1. *Программа*, предназначенная для решения задачи или класса задач в определенной области науки, техники, экономики, искусства, народного хозяйства, связанных с применением вычислительной системы. Например, программа расчета прочности стержневых конструкций, программа резервирования железнодорожных билетов и т. п. 2. Программа, использующая средства, предоставляемые *системной программой*. В этом смысле термин "П. п." применяется к программе, чтобы отличить ее от системных программ, таких как *операционная система* или средства *разработки* и *отладки программ*

ПРИКЛАДНОЙ УРОВЕНЬ [application layer]. То же, что *уровень приложений*

ПРИКРЕПЛЕННЫЙ ФАЙЛ [attached file]. То же, что *вложенный файл*

ПРИЛОЖЕНИЕ [application]. *Прикладная программа*, предназначенная для выполнения под управлением определенной *операционной системы* с использованием конкретного типа компьютера. Например, *текстовый редактор* Microsoft Word 7.0 является приложением Windows 95

ПРИЛОЖЕНИЕ, УПРАВЛЯЕМОЕ СОБЫТИЯМИ [event-driven application]. *Приложение*, в котором пользователь, а не программа, управляет процессом выполнения. Пользователь управляет приложением, воздействуя на *элементы управления графического интерфейса пользователя*. А приложение выполняет соответствующие процедуры реакции (см. *программирование, ориентированное на события*). П., у. с., реагирует также на системные события, например, прерывание от таймера, или программные события, например, событие открытия формы. Ср. *процедурное приложение*

ПРИМИТИВ [primitive]. 1. Элемент, который нельзя разложить на более простые формы. В *языках программирования высокого уровня* этот термин служит для обозначения встроенных и неделимых (т. е. не имеющих определенной в языке внутренней структуры) элементов языка. П. являются *ключевые слова* и *встроенные типы*, такие как *целый тип* или *вещественный тип*, в то время как *массив* не является П., потому что имеет видимую

структуру (состоит из элементов). Аналогично операция сложения (+) является П., потому что встроена в язык, и ее реализация скрыта от программиста, в то время как определенная программистом функция не является П. 2. То же, что *графический примитив*

ПРИМИТИВНЫЙ ТИП [primitive type]. То же, что *встроенный тип*

ПРИНТЕР, печатающее устройство [printer]. *Устройство*, предназначенное для вывода *текстовой и графической информации* из компьютера в печатном виде на бумагу или другой подобный носитель. П. используются чаще всего для вывода текстов и текстовых документов. Распространены в основном принтеры трех видов: матричные, струйные и лазерные. Качество печати определяется *разрешающей способностью* П. См. *лазерный принтер, матричный принтер, струйный принтер*

ПРИНТЕР С СЕТЕВЫМ ИНТЕРФЕЙСОМ [network-interface printer]. Принтер, подключенный непосредственно в компьютерную сеть через собственную сетевую карту

ПРИНЦИП ПОДСТАНОВОЧНОСТИ [substitutability principle]. Принцип *объектно-ориентированного программирования*, согласно которому любой экземпляр любого потомка класса *X* может использоваться в качестве фактического значения переменной или параметра, объявленного как имеющего тип *X*, не нарушая при этом семантику объявления или использования. Другими словами, экземпляр класса-потомка можно подставить вместо экземпляра класса-предка. Термин был предложен Барбарой Лисков (Barbara Liskov). См. *наследование*

ПРИОРИТЕТ [priority]. Присваиваемый задаче, программе или операции признак, определяющий очередность их выполнения или обслуживания *вычислительной системой*. П. может быть определен количественной величиной. Чем выше П. программы, тем бóльшим преимуществом она обладает в очереди на выполнение, получение ресурсов и т. п. Поэтому в режиме мультипрограммирования программам можно так назначить П., что срочные работы не будут задерживаться вспомогательными задачами. См. *мультипрограммирование*

ПРИОРИТЕТНАЯ МНОГОЗАДАЧНОСТЬ [preemptive multitasking]. То же, что *вытесняющая многозадачность*

ПРИРАЩЕНИЕ [increment]. Часть *программного продукта*, которая разрабатывается за одну итерацию *инкрементального процесса разработки программного обеспечения*

ПРИСВАИВАНИЕ [assignment]. Операция задания значения регистру, переменной, *элементу массива* или другому *элементу данных*. Например, фраза: "Переменной *x* присвоено значение 3.5" означает, что с момента П. значе-

ние *x* равно 3.5, и *x* сохранит это значение до следующей операции П. В *языках программирования* П. задается с помощью *оператора* П.

ПРОБЛЕМНАЯ ОБЛАСТЬ ПАМЯТИ [dynamic storage area]. То же, что *динамическая область памяти*

ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЙ [problem-oriented]. Ориентированный на решение задач определенного класса. Например, "1С Бухгалтерия" является П.-о. *пакетом прикладных программ*, ориентированным на решение задач бухгалтерского учета в различных *предметных областях*

ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЙ ЯЗЫК [problem-oriented language]. *Язык программирования высокого уровня*, предназначенный для решения определенного класса задач (проблем). П.-о. я. по возможности использует символику и систему понятий, принятую при решении задач данного класса. Например, для подготовки на компьютере научно-технических книг, статей и других публикаций, содержащих математические формулы, предназначен П.-о. я. *Tex*, в который в качестве *ключевых слов* входят некоторые полиграфические и математические термины. См. *Tex*

ПРОВАЙДЕР [provider]. То же, что *поставщик сетевых услуг*

ПРОВЕРКА НА ЧЕТНОСТЬ [even parity]. Проверка правильности хранимых или пересылаемых данных методом *контроля по четности*

ПРОВЕРКА ОРФОГРАФИИ [spelling checker]. См. *блок орфографического контроля*

ПРОВОДНИК WINDOWS [Windows Explorer]. *Приложение*, входящее в состав *операционной системы Windows*, которое предназначено для управления файлами, в том числе находящимися в *локальной сети*, и фактически выполняет функции *оболочки операционной системы*. П. Windows позволяет выполнять все функции, которые реализует, например, о*болочка Norton Commander*, но при этом снабжен удобным *графическим интерфейсом*. На рис. Д.1 показано окно П. Windows, которое состоит из двух панелей: слева *дерево каталогов*, а справа — содержимое текущего каталога. См. *копирование файла, перенос файла, поиск файла, переименование файла*

ПРОГОН БУМАГИ [paper feed]. Продвижение бумаги в принтере, графопостроителе и т. п. на определенное число строк

ПРОГОН ПРОГРАММЫ [program run, run]. Однократное *выполнение программы*

ПРОГОН СТРАНИЦЫ [form feed (FF)]. То же, что *перевод страницы*

ПРОГРАММА [program]. Последовательность указаний (команд или описаний и операторов), задающая алгоритм *вычислительной машине*. П. указывает, в каком порядке над какими данными и какие операции должны быть

выполнены компьютером и в какой форме должен быть выдан результат. *Устройство управления* воспринимает П., заданную в виде последовательности *машинных команд.* Составление П. на *машинном языке* — неудобный и трудоемкий процесс. Поэтому обычно П. для компьютера составляется человеком на одном из *языков программирования*, а затем сам компьютер переводит (транслирует) эту программу на машинный язык. См. *исходная программа, машинная программа*

ПРОГРАММА БУФЕРИЗАЦИИ, спулер [spooler]. Программа, управляющая процессом *буферизации.* Например, П. б. перехватывает данные, направляемые к драйверу принтера, и записывает их в буфер, чтобы передать на принтер, когда он освободится. См. *спулинг, спулер печати*

ПРОГРАММА ПРОСМОТРА, браузер, броузер. 1. **[browser]**. *Программа* чтения *гипертекста* (browser). См. *браузер* 2. **[viewer]**. Средства просмотра (viewers) файлов различных типов. Так, в состав ОС Windows 95 входит утилита QuickView Plus для просмотра текстовых и графических файлов. Средства предварительного просмотра файлов различных форматов поддерживаются современными *текстовыми* и *графическими процессорами*, программами обработки *электронных таблиц*

ПРОГРАММА РАСКРАСКИ [paint program]. Приложение, которое позволяет создавать методом *растровой графики* изображения, хранимые в памяти в виде набора пикселов, а не отдельных линий, фигур и прочих графических объектов. С помощью П. р. можно рисовать от руки. Для этого пользователю предоставляется набор инструментов: шаблонов, цветов, форм кисточек и линий разной толщины и т. п. П. р. обрабатывает картинку как группу точек и часто допускает поточечную перерисовку небольших фрагментов рисунка. В П. р. при выделении и перемещении фрагмента рисунка на его месте остается неокрашенная область пикселов. Примером П. р. является Microsoft Paint. Ср. *программа рисования*

ПРОГРАММА РИСОВАНИЯ [drawing program]. Приложение, которое позволяет создавать изображения методом *векторной графики.* Эти изображения хранятся в памяти как набор математических описаний отдельных линий и геометрических фигур, которые пользователь рисует на экране дисплея с помощью мыши или другого *указательного устройства.* П. р. позволяет легко перемещать выделенные геометрические элементы и блоки текста, обращаться с ними, как с независимыми объектами, меняя их размеры и ориентацию, а также производить *заливку* очерченных областей нужным цветом. Для этого пользователю предоставляется набор инструментов: шаблонов фигур и линий разной толщины и цвета, набор цветов для заливки и т. п. П. р. обрабатывает картинку как группу линий и допускает перерисовку отдельно выделяемых составляющих рисунка. Примером П. р. является CorelDRAW. Ср. *программа раскраски*

ПРОГРАММА С ОВЕРЛЕЙНОЙ СТРУКТУРОЙ [overlay program]. То же, что *оверлейная программа*

ПРОГРАММА С ПЕРЕКРЫВАЕМОЙ СТРУКТУРОЙ [overlay program]. То же, что *оверлейная программа*

ПРОГРАММИРОВАНИЕ, разработка программ [programming]. 1. Процесс создания *программы*. П. включает в себя анализ требований к программе и все стадии ее разработки и реализации: выбор алгоритма, *структуры данных* и *системы программирования*; написание (кодирование) программы и подготовку данных; *отладку* и *испытания программы*; создание сопровождающей программу документации. В более узком смысле П. — это написание программы *на языке программирования*, ее отладка и испытания. Если алгоритм сложен, то П. рекомендуется выполнять способом *модульного* П. и начинать с составления *блок-схемы программы*. При этом пользуются методами *восходящего* или *нисходящего* П. 2. Наука, занимающаяся разработкой методов и средств получения и реализации программ для *вычислительных машин*

ПРОГРАММИРОВАНИЕ, ОРИЕНТИРОВАННОЕ НА СОБЫТИЯ, событийно-управляемое программирование [event-driven programming]. Способ структуризации программного кода, основанный на следующей идее. Имеется некоторое предопределенное множество поименованных *событий*. События могут быть явным (или неявным) образом связаны с *программными объектами*, а могут быть связаны с неявными объектами, в таком случае события обычно называют системными. События способны "возникать". Возникновение события подразумевает, что состояние системы изменилось определенным образом. С событием связывается процедура, называемая "реакцией" на событие, которая автоматически вызывается при возникновении события. В современных *системах программирования*, поддерживающих П., о. н. с., предусматривается большое число самых разнообразных событий, реакции на которые могут быть определены в программе, например, нажатие клавиши на *клавиатуре*, перемещение *указателя мыши* в определенную область экрана, достижение внутренним *таймером* заданного значения, открытие заданного файла и т. д. В программе, целиком управляемой событиями, нет основного *потока управления*, он находится вне программы (в *операционной системе* или в административной системе *времени выполнения*, т. е. там, где реализован механизм возникновения событий). Управление в программу попадает только в форме *вызова процедуры* реакции. П., о. н. с., подразумевает следующую методику разработки и использования программы. Программист формирует *графический интерфейс* программы с помощью стандартных или специально разработанных *элементов управления*, а также определяет необходимые процедуры реакции для событий, связанных с использованными элементами управления. Пользователь, воздействуя на элементы управления, вызывает те или иные события и тем самым управляет ходом работы программы

ПРОГРАММИСТ [programmer]. Лицо, выполняющее разработку и отладку программ

ПРОГРАММНАЯ ДОКУМЕНТАЦИЯ [software documentation]. То же, что *документация программного продукта*

ПРОГРАММНАЯ ОШИБКА [program error]. То же, что *ошибка в программе*

ПРОГРАММНАЯ ПЛАТФОРМА [program platform, platform]. Тип *операционной системы* и *интерфейса прикладного программирования*, с которыми может работать *программный продукт*. Например, Win32, MS-DOS, Linux и т. д. Ср. *компьютерная платформа*

ПРОГРАММНОЕ ИЗДЕЛИЕ [software]. То же, что *программный продукт*

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ [software]. Совокупность входящих в состав *вычислительной системы программных средств*, т. е. программ, данных и документов к ним. П. о. обеспечивает эффективную работу компьютера и предоставляет пользователю определенные виды обслуживания. Этот термин определяет часть вычислительной системы, не являющуюся *аппаратным обеспечением*. Различают *системное программное обеспечение*, которое является необходимым дополнением аппаратных средств, и прикладное программное обеспечение, которое определяется ролью вычислительной системы в данной организации или потребностями пользователей. Системное программное обеспечение, важнейший компонент которого — *операционная система*, обычно поставляется самим производителем вычислительной системы. Прикладное программное обеспечение состоит из *баз данных*, пакетов и библиотек *прикладных программ* и т. п.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ WINDOWS SOCKETS, ПО Windows Socket [Windows Sockets (WinSoc)]. *Сетевое программное обеспечение* для *операционных систем Windows*, позволяющее приложениям использовать *сокет-каналы* при работе в *компьютерных сетях*. ПО Windows Socket обеспечивает интерфейс между прикладными программами, исполняемыми на разных компьютерах

ПРОГРАММНОЕ ПРЕРЫВАНИЕ, прерывание прикладной программы [software interrupt]. *Прерывание*, вызванное либо обращением программы к *аппаратным средствам* (например, ввод *исходных данных*, вывод результатов, очистка экрана и т. п.), либо в связи с некорректным представлением или использованием команд или данных. Обычно это происходит при ошибочно заданном *параметре цикла*, неправильном вычислении индекса массива, неверном *формате данных* и т. п. Регистрация некоторых из этих ситуаций может быть отменена *командами операционной системы*. В таком случае ошибки в программе не регистрируются и за правильность вычислений ответственность несет программист. См. *исключительная ситуация*

ПРОГРАММНЫЕ СРЕДСТВА [software]. Программы, данные и документы к ним, входящие в состав *программного обеспечения вычислительной системы*, обеспечивающие ее эффективную работу и предоставляющие пользователю определенные виды обслуживания. Это части вычислительной системы, не являющиеся *аппаратными средствами*

ПРОГРАММНЫЙ [softwired]. Реализованный *программными средствами*. Ср. *аппаратный*

ПРОГРАММНЫЙ БЛОК, блок [program block, block]. То же, что *блок программы*

ПРОГРАММНЫЙ ИНТЕРФЕЙС [program interface]. 1. *Интерфейс* между программами. 2. Программная реализация интерфейса. См. *интерфейс прикладного программирования, программный канал*

ПРОГРАММНЫЙ КАНАЛ, канал [pipe]. Специальный *тип данных операционной системы*, служащий средством обмена данными между родственными процессами в компьютере или в *компьютерной сети*. Процесс может читать данные из канала и передавать данные в канал, как при работе с файлами. П. к. может быть односторонним (один процесс может только читать из него, а другой — только записывать) и двусторонним. Частными случаями П. к. являются: *именованный канал, конвейер, сокет-канал*

ПРОГРАММНЫЙ МОДУЛЬ [module, program unit]. *Программа* или часть программы, оформленная в виде, допускающем ее независимую трансляцию. П. м. должен выполнять четко определенную функцию, быть оформлен по правилам *языка программирования* или в соответствии с внутренними стандартами *операционной системы*, иметь хорошо определенный интерфейс с другими П. м. и *внешними устройствами* и относительно небольшие размеры. Программы можно конструировать из П. м., как из составных частей, каждая из которых может разрабатываться, программироваться, транслироваться и тестироваться независимо от других. Внутреннее строение П. м. для функционирования всей программы, как правило, значения не имеет. При модификации алгоритма работы модуля структура программы не должна меняться. Простейшим примером П. м. является подпрограмма. См. *модульное программирование*

ПРОГРАММНЫЙ ОБЪЕКТ [program object]. То же, что *объект программы*

ПРОГРАММНЫЙ ПРОДУКТ, программное изделие [program product, software]. *Программа (пакет программ)*, предназначенная для продажи или передачи в эксплуатацию другим лицам и удовлетворяющая ряду требований, важнейшие из которых: 1) сама программа и прилагаемая к ней инструкция должны содержать достаточное количество данных для своего полноценного использования; 2) программа должна сопровождаться производителем, т. е. замеченные ошибки должны устраняться бесплатно для покупателей;

3) программа должна поставляться в удобном для установки и использования виде, как правило, на *гибких* или *лазерных дисках* с инструкцией и защитной упаковкой (см. *дистрибутив*); 4) программа должна быть изготовлена с помощью законно приобретенных *программных средств* и запатентована

ПРОГРЕСС-ИНДИКАТОР [progress indicator]. Элемент *графического интерфейса пользователя*, показывающий ход выполнения некоторого процесса. Например, на рис. П.9 изображено диалоговое окно с П.-и., показывающим ход выполнения операции *копирования файла*

Рис. П.9. Прогресс-индикатор в диалоговом окне **Копирование**

ПРОЕКТИРОВАНИЕ [design]. Фаза *процесса разработки программного обеспечения*, в которой описывается, как система будет реализована на логическом уровне, еще до написания программного кода. Во время этой фазы принимаются стратегические решения относительно того, как воплотить в жизнь требования к системе. Ср. *конструирование*, *реализация*

ПРОЕКТ GNU [GNU] (аббревиатура раскрывается рекурсивно: "GNU's Not Unix" — "GNU — это не UNIX"). Выполняемый *Фондом свободного программного обеспечения* комплекс работ по созданию полной интегрированной программной системы, средства которой совместимы с возможностями среды *операционной системы UNIX*. Как правило, возможности программ GNU шире возможностей аналогов среды UNIX. Хотя П. GNU развивается по плану, в него принимаются для свободного распространения и программы, разработанные фирмами и частными лицами по собственной инициативе. См. *операционная система Linux*

ПРОИЗВОДИТЕЛЬНОСТЬ КОМПЬЮТЕРА, быстродействие компьютера, быстродействие ЭВМ, производительность ЭВМ [computer speed, computer performance]. Характеристика скорости и показатель качества работы ЭВМ. Часто, говоря о П. к., подразумевают способность *центрального процессора* быстро выполнять команды. Величину П. к. измеряют числом операций в секунду или *тактовой частотой*. Однако при этом следует учитывать, что разные операции имеют разную длительность, а производительность цен-

трального процессора зависит от таких факторов, как длина *машинного слова*, *система команд*, *время доступа* к памяти, наличие *кэш-памяти* и т. п. Поэтому оценку П. к. производят по совокупности параметров, которая включает тактовую частоту процессора, время выполнения операций сложения, пропускную способность шины, скорость обмена с *жестким диском* и, наконец, время обработки эталонных тестов. См *бенчмарка*. Ср. *пиковая производительность компьютера*

ПРОИЗВОДИТЕЛЬНОСТЬ ЭВМ [computer speed, computer performance]. То же, что *производительность компьютера*

ПРОИЗВОДИТЕЛЬ ОБОРУДОВАНИЯ [original equipment manufacturer (OEM)]. Термин, используемый для обозначения фирм, занимающихся разработкой и производством компьютерного оборудования. Иногда этот термин обозначает фирму, занимающуюся предустановкой программного обеспечения на компьютеры других производителей

ПРОИЗВОДНЫЙ ТИП ДАННЫХ [derived datatype]. *Тип данных*, образованный от каких-либо других типов данных. Например, *структурированный тип*

ПРОИЗВОДНЫЙ ЭЛЕМЕНТ [derived element]. Элемент, который можно вычислить по другим элементам. Вводится для большей ясности или каких-либо других целей, несмотря на то, что не несет собственной семантической нагрузки

ПРОИЗВОЛЬНЫЙ ПОВОРОТ [free rotate]. *Команда графического редактора*, позволяющая поворачивать на экране выделенную область в ручном режиме. По команде П. п. вокруг выделенной области появляются маркеры, с помощью которых ее можно свободно вращать мышью. См. *выделение области*

ПРОКРУТКА [scrolling]. То же, что *скроллинг*

ПРОКРУТКА ИЗОБРАЖЕНИЯ [scrolling]. То же, что *скроллинг*

ПРОЛОГ [Prolog]. *Язык программирования высокого уровня*, предназначенный для разработки программ и систем *искусственного интеллекта*. П. позволяет описать решаемую задачу на точном логическом языке, указывая конечную цель разработки, не предписывая заранее детальный способ ее достижения. При этом возможности П. расширены рядом *встроенных процедур*: ввода, вывода, выполнения *арифметических операций*, управления ходом доказательства и т. п. П. обеспечивает легкость написания, понимания и модификации программ, возможность программирования *параллельных алгоритмов*. П. применяется для создания *экспертных систем*, *интеллектуальных информационных систем*, *интеллектуальных обучающих систем*, *интеллектуальных систем программирования*. Имеется ряд версий П., которые реализованы на *персональных компьютерах*. П. разработан в 1970-х гг. коллекти-

вом французских ученых во главе с А. Калмероэ. Свое название получил от сокращения английских слов PROgramming in LOGic

ПРОМАХ КЭША [cache miss]. То же, что *ошибка кэша*

ПРОМЕЖУТОЧНЫЙ ЯЗЫК [intermediate language]. *Язык программирования*, на который производится трансляция с *языка программирования высокого уровня* и с которого, в свою очередь, выполняется трансляция на *машинный язык*. См. *диалоговый процессор*

ПРОПУСКНАЯ СПОСОБНОСТЬ [bandwidth]. Наибольшее количество данных, которое можно передать по *каналу связи* или *компьютерной сети*. Обычно измеряется в битах в секунду

ПРОСТРАНСТВО ИМЕН [name space]. Термин, используемый в *объектно-ориентированных языках программирования*, в *языках разметки* и др. для обозначения способа интерпретации составных имен. В одном П. и. действуют обычные правила интерпретации имен, например, все имена однородных объектов должны быть уникальны или должны подчиняться правилам *области определения* имен. Имена в различных П. и. не зависят друг от друга, в частности, для обозначения различных объектов могут использоваться одинаковые имена. При указании имени объекта, определенного в другом П. и., используется составное имя, образованное из имени П. и. и имени объекта в этом П. и. Таким образом, хотя различные объекты могут иметь совпадающие имена, неоднозначности не возникает

ПРОТОКОЛ [protocol]. 1. Результат регистрации в хронологическом порядке информации о некотором процессе. Например, П. компиляции — формируемый *вычислительной системой* листинг, содержащий исходный текст программы, ее эквивалент в виде *объектной программы*, таблицу соответствия имен и *адресных ссылок*, диагностические сообщения об ошибках и другую информацию. 2. Совокупность правил, определяющих алгоритм взаимодействия устройств, программ, систем *обработки данных*, процессов или пользователей. Например, П. *линии связи* — это правила, регламентирующие структуру и методы кодирования порции данных и процесс ее передачи по линиям связи. П. международной компьютерной сети Интернет (так называемые IP-протоколы) определяют правила взаимодействия между собой как объединенных в Интернете региональных сетей, так и подключенных к сети отдельных компьютеров

ПРОТОКОЛ МАРШРУТНОЙ ИНФОРМАЦИИ, протокол RIP [routing information protocol (RIP)]. *Протокол*, позволяющий маршрутизатору обмениваться информацией о маршруте передаваемого сообщения с ближайшим маршрутизатором. Такая информация может содержать сведения о возможных вариантах маршрута и соответствующем каждому варианту количестве переходов через другие маршрутизаторы. По этим сведениям обычно выбирается путь с наименьшим числом переходов

ПРОТОКОЛ НАЧАЛЬНОЙ ЗАГРУЗКИ, протокол BOOTP [bootstrap protocol (BOOTP)]. *Сетевой протокол* из набора *протоколов TCP/IP*, используемый для конфигурирования и начальной загрузки сетевых бездисковых *рабочих станций*. В таких станциях загрузка программ и все действия с файлами производятся с *файлового сервера*

ПРОТОКОЛ ПЕРЕДАЧИ ГИПЕРТЕКСТА, протокол HTTP [HyperText Transfer Protocol (HTTP)]. Стандартный *протокол* передачи *гипертекста* и *гипермедиа* в сети Интернет, поддерживаемый *Комитетом IETF* и описывающий обмен информацией во *Всемирной паутине*

ПРОТОКОЛ ПОЧТОВОГО ОФИСА [post office protocol (POP)]. То же, что *протокол POP*

ПРОТОКОЛ ТРАНСПОРТНОГО УРОВНЯ [transport protocol]. То же, что *транспортный протокол*

ПРОТОКОЛ УДАЛЕННЫХ КЛИЕНТОВ ЛИНЕЙНОГО ПРИНТЕРА, протокол LPR [line printer remote clients (LPR)]. Один из *сетевых протоколов* в семействе *протоколов TCP/IP*, определяющий стандарт передачи заданий на печать между компьютерами. Широко используется в Интернете для связи с *демонами линейных принтеров*. П. LPR опубликован в *RFC* 1179. См. *стандарты RFC*

ПРОТОКОЛ APPC [advanced program-to-program communications (APPC)]. Протокол, созданный фирмой IBM для *системной сетевой архитектуры* (архитектуры SNA), предназначенный для обеспечения связи между прикладными программами, работающими на разных компьютерах, и для непосредственного обмена данными. П. APPC предоставляет в распоряжение прикладных программ набор правил, а также обобщенный язык, которым они могут пользоваться, не обращаясь для этого к сетевым функциям низкого уровня

ПРОТОКОЛ BOOTP [bootstrap protocol (BOOTP)]. То же, что *протокол начальной загрузки*

ПРОТОКОЛ FTP [file transfer protocol (FTP)]. Один из стандартных *протоколов передачи данных*, поддерживаемый *комитетом IETF* и обеспечивающий эффективную передачу файлов по телефонной линии. П. FTP используются при переносе файлов с одного компьютера на другой или при транспортировке файлов между различными узлами сети Интернет. Чтобы установить соединение и обменяться файлами в Интернете согласно П. FTP, необходимо запустить специальную прикладную программу, так называемую "клиентскую часть FTP". Клиентское программное обеспечение FTP устанавливается вместе с коммуникационными утилитами TCP/IP. Получить доступ к другому компьютеру для обмена файлами можно, указав пользовательское имя и пароль. Существуют так называемые анонимные

FTP-серверы, позволяющие получать файлы любому пользователю, обладающему клиентской частью FTP. В этом случае в качестве пользовательского имени вводится "anonymous", а в качестве пароля — адрес электронной почты пользователя П. FTP

ПРОТОКОЛ LPR [line printer remote clients (LPR)]. То же, что *протокол удаленных клиентов линейного принтера*

ПРОТОКОЛ NETBEUI [network basic input/output system extended user interface (NetBEUI)]. Основной *транспортный протокол*, используемый *операционной системой Windows NT* для небольших локальных сетей, объединяющих до 200 пользователей

ПРОТОКОЛ POP, **протокол почтового офиса, почтовый протокол [post office protocol (POP)]**. *Протокол* сети Интернет, позволяющий осуществлять динамический доступ в *почтовый ящик* сервера с *рабочей станции*. Работает на основе *протокола TCP/IP*. Имеется несколько версий этого протокола, определенных в *стандартах RFC*

ПРОТОКОЛ PPP [point-to-point protocol (PPP)]. Набор промышленных стандартов, описывающих процедуры *передачи данных*. П. PPP связывает конфигурационные параметры нескольких уровней *модели OSI*

ПРОТОКОЛ RIP [routing information protocol (RIP)]. То же, что *протокол маршрутной информации*

ПРОТОКОЛ SET [secure electronic transaction (SET)]. То же, что *защищенные электронные транзакции*

ПРОТОКОЛ SNA [system network architecture protocol, SNA protocol]. Стандартный *сетевой протокол*, построенный в соответствии с *архитектурой SNA*. Широко применяется в компьютерах фирмы IBM. Пользователи *персональных компьютеров* обычно используют этот протокол для связи с мэйнфреймами при помощи *удаленного доступа*

ПРОТОКОЛ SNMP [simple network management protocol (SNMP)]. Дословный перевод с английского: "простой протокол управления сетью". Сетевой протокол, предназначенный для управления компонентами сети, мониторинга загруженности сети и выполнения других задач

ПРОТОКОЛ SSL [secure sockets layer (SSL)]. Протокол безопасной передачи данных по *Всемирной паутине*, разработанный корпорацией Netscape Communications. Он не зависит от приложения и обеспечивает: 1) *шифрование* (безопасный канал, защищенный от вмешательства посторонних); 2) *аутентификацию* (применение электронных сертификатов для удостоверения каждой из сторон, вступающих в контакт); 3) целостность сообщений (гарантию того, что сообщение не было изменено при передаче). П. SSL функционирует на уровне более низком, чем протоколы приложений, такие

как HTTP или FTP, но в то же время более высоком, чем протокол связи TCP/IP. Такая стратегия позволяет ему действовать независимо от любого протокола Интернета. Если П. SSL реализован одновременно на клиентской и серверной сторонах, информация между ними передается в зашифрованном виде, что гарантирует конфиденциальность

ПРОТОКОЛ TELNET [Telnet]. *Протокол удаленного доступа*, используемый в Интернете. Позволяет пользователю одного компьютера устанавливать связь по сети с удаленной *вычислительной системой*. Когда связь установлена, пользователь может работать так, как будто его клавиатура, мышь и монитор подключены непосредственно к удаленному компьютеру

ПРОТОКОЛЫ TCP/IP [transmission control protocol/Internet protocol (TCP/IP)]. Набор базовых *протоколов Всемирной компьютерной сети Интернет*, которые определяют правила взаимодействия между собой как объединенных в Интернете региональных сетей, так и подключенных к сети отдельных компьютеров. Межсетевым протоколом IP устанавливается, что информация передается *пакетами* размером от 40 до 32000 байт в зависимости от *аппаратных средств* и *носителей информации*, и каждый пакет перемещается по сети независимо от других пакетов. Такая технология не позволяет одному пользователю монополизировать сеть. Пакеты снабжаются *интернет-адресами*, которые содержат информацию, достаточную для доставки пакета. Протокол управления *передачей данных* TCP определяет правила разбиения данных на пакеты и их нумерацию, а также правила, согласно которым при получении отдельных пакетов проверяется, вся ли информация получена, а данные располагаются в правильном порядке. Если оказывается, что при передаче пакета произошла ошибка, то запрашивается его повторная передача. В соответствии с П. TCP/IP работает *аппаратное* и *программное обеспечение*, реализующее передачу данных по Интернету

ПРОТОТИП [prototype]. 1. Макет, на котором можно продемонстрировать или проиллюстрировать некоторые аспекты разрабатываемого *программного продукта*. 2. Программный продукт, который берется в качестве образца при разработке нового программного продукта

ПРОФАЙЛЕР [profiler]. То же, что *профилировщик*

ПРОФИЛИРОВЩИК, профайлер [profiler]. *Программа* определения *профиля программы* или *профиля оборудования*. П. собирает сведения, например, о порядке выполнения модулей, об использовании программой тех или иных ресурсов, о характеристиках работы оборудования в различных режимах и т. д.

ПРОФИЛЬ ОБОРУДОВАНИЯ [equipment profile]. Набор значений, параметров и характеристик конкретного *оборудования* компьютера. Эти данные могут быть использованы, например, для настройки компьютера на работу с

периферийными устройствами. Для получения П. о. применяются специальные программы — *профилировщики.* См. *конфигурация компьютера*

ПРОФИЛЬ ПОЛЬЗОВАТЕЛЯ [user profile]. Информация о доступных *пользователю* ресурсах *вычислительной системы* или *компьютерной сети* и о настройках *рабочей среды пользователя.* Эта информация представляет собой набор параметров *рабочей среды пользователя*, загружаемый при входе пользователя в систему и определяющий доступные для пользователя группы программ и файлов, цвета экрана, сетевые подключения дисков и принтеров, свойства мыши, размеры и положение окон и т. п. На основе П. п. программируется *сценарий регистрации*

ПРОФИЛЬ ПРОГРАММЫ [program profile]. Информация о ходе выполнения программы. Включает сведения о последовательности и количестве выполнений *программных модулей*, операторов (либо всех, либо указанных групп операторов), количестве обращений к переменным и другие данные, необходимые для анализа поведения программы. Для получения П. п. применяются специальные программы — *профилировщики*

ПРОФИЛЬ УСТРОЙСТВА [device profile]. Файл, сохраняющий описание параметров конкретного устройства ввода или вывода (монитора, сканера, принтера). Используется при автоматической настройке *вычислительной системы* (см. *технология Plug and Play*) и в *системе управления цветом*

ПРОЦЕДУРА [procedure]. Конструкция многих *языков программирования высокого уровня* (например, Паскаль), соответствующая понятию *подпрограммы.* Представляет собой поименованную часть программы (*блок программы* или группу описаний и операторов), которая может выполнять некоторые четко заданные действия над условными данными, определяемыми с помощью *формальных параметров.* Выполнение П. может быть инициировано из любого места программы одним *оператором* (см. *вызов процедуры*). При вызове П. вместо формальных параметров указываются *фактические параметры*, определяющие конкретные данные, над которыми и выполняются запрограммированные в П. действия. Возможны П., в которых нет формальных параметров. В них операторы сразу задают действия над объектами, определенными в *главной программе.* П. вводятся в программу с помощью *описания* П., которое обычно располагается в разделе описаний. Описание П. состоит из *заголовка* П. и *тела* П. Заголовок служит для присвоения П. некоторого имени и, возможно, указания формальных параметров. В теле программируется выполняемый П. алгоритм. Ср. *функция.* См. *описание процедуры, вызов процедуры*

ПРОЦЕДУРНОЕ ПРИЛОЖЕНИЕ [procedural application]. *Приложение*, в котором инициатива принадлежит программе, а не пользователю. Программа определяет, что следует делать дальше, а когда программе нужно полу-

чить данные от пользователя, она выводит *диалоговое окно*. Ср. *приложение, управляемое событиями*

ПРОЦЕСС [process]. Последовательность операций при выполнении программы или ее части вместе с используемыми данными. *Операционная система* рассматривает П. как единое целое при распределении ресурсов. См. *задача*

ПРОЦЕССОР [processor]. Устройство, выполняющее команды. Обязательными компонентами П. являются *арифметико-логическое устройство* и *устройство управления*. П. характеризуются архитектурой, набором выполняемых команд, скоростью их выполнения и длиной *машинного слова*. Архитектура определяет типы обрабатываемых данных, регистры, стеки, систему адресации и т. п. Набор команд может быть фиксированным или с возможностью микропрограммирования. Микропрограммируемые П. имеют встроенную память, в которой хранятся микропрограммы, определяющие набор выполняемых команд. Среди фиксированных команд, как правило, есть команды *арифметических* и *логических операций*, *передачи управления* и перемещения данных между регистрами, стеками, памятью и портами ввода/вывода. Команды П. обычно обеспечивают обработку слов следующей длины: бит, полубайт (4 бита), байт (8 бит), два, четыре или восемь байтов. Существуют *векторные процессоры*, команды которых позволяют одновременно обрабатывать массивы данных. В современных компьютерах в качестве процессоров применяются микропроцессоры. В разговорном языке под словом П. чаще всего подразумевается *центральный процессор*

ПРОЦЕССОР КОМАНДНОГО ЯЗЫКА [command processor]. То же, что *командный процессор*

ПРОЦЕССОРНОЕ ВРЕМЯ [CPU time]. То же, что *время центрального процессора*

ПРОЦЕССОРНЫЙ ЭЛЕМЕНТ [processor unit]. 1. *Арифметико-логическое устройство* с собственной памятью, работающее наряду с другими подобными устройствами под управлением одного *устройства управления*. Устройство управления выдает *поток команд*, синхронно исполняемый П. э. 2. *Процессор*, не являющийся *центральным процессором*

ПРОЦЕССОР ПЕЧАТИ [print processor]. Компонент *спулера печати — библиотека динамической компоновки*, которая обрабатывает *задания на печать* перед посылкой их на исполнение печатающим устройством. П. п. вносит в задание на печать необходимые изменения, основываясь на *типе данных задания на печать*

ПРОЦЕССОР СО СВЕРХДЛИННЫМ КОМАНДНЫМ СЛОВОМ, VLIW-процессор [very large instruction word processor, VLIW-processor]. *Процессор*, реализующий параллельное выполнение нескольких операций, заданных

одной машинной командой. Обнаружение параллелизма в исходной программе происходит в процессе трансляции. Транслятор анализирует программу, определяя, какие операции могут выполняться параллельно, и "упаковывает" их в различные поля одной длинной команды. Каждое поле отвечает за свой набор операций, исполняемых определенной функциональной частью процессора. Если на данном этапе выполнения программы какая-то часть процессора не востребована, то соответствующее поле команды не задействуется. После того как такая команда выбрана из памяти, составляющие ее обычные команды выполняются параллельно. Ср. *суперскалярный процессор.* См. *параллельная обработка, параллельные вычисления*

ПРОЦЕССОР ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ (ЦОС) [digital signal processor (DSP)]. Специализированный микропроцессор, предназначенный для обработки в реальном времени преобразованных в цифровую форму аналоговых сигналов, например, оцифрованного звука или видео. ЦОС используется в звуковых и видеокартах, разгружая центральный процессор, а также в телефонах и системах сбора данных и обработки видеоизображений.

ПРОЦЕССОР MMX [multimedia extension processor (MMX processor)]. Серия процессоров фирмы Intel, реализующих *технологию MMX.* П. MMX обладает расширенным набором команд и *типов данных*, предназначенных для быстрого выполнения *параллельных алгоритмов* в приложениях мультимедиа и *компьютерных играх.* Важно, что эти команды и типы данных полностью совместимы с существующими программами и *операционными системами*

ПРОЦЕССОР POWERPC [power performance chip, PowerPC (PPC)]. *Микропроцессор суперскалярной архитектуры RISC*, разработанный в 1992 г. фирмами Motorola и IBM при участии фирмы Apple. П. PowerPC имеет 64-битную *шину данных* и 32-битную *адресную шину*, раздельное кэширование данных и команд, мультиустройства обработки *целых чисел* и *чисел с плавающей точкой.* PowerPC — зарегистрированная торговая марка фирмы IBM

ПРОЦЕСС РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ [software development process]. Регламентированная и стандартизованная *разработка программы* управляемым и воспроизводимым образом. Цель П. р. п. о. — успех всей работы и качество конечного программного продукта. П. р. п. о. содержит способ выделения фаз или стадий процесса, возможные последовательности фаз, а также условия перехода к следующей стадии и типы характерных артефактов для каждой фазы. Исторически первым явился *водопадный* П. р. п. о. В настоящее время наиболее распространен *инкрементный* П. р. п. о.

ПРЯМАЯ АДРЕСАЦИЯ, непосредственная адресация [direct addressing, immediate addressing]. Метод *адресации*, при котором адресная часть команды содержит *адрес* операнда. Ср. *косвенная адресация*

ПРЯМОЙ ДОСТУП [direct access]. Способ *доступа* к данным, при котором все *элементы данных* или записи в файле являются равнодоступными, т. е. время доступа к элементу или записи не зависит от расположения данных в памяти и для доступа к любому из них не требуется "просмотр" других элементов или записей. При П. д. обращение к данным осуществляется либо по адресу, либо по *ключу записи*, если при этом *время доступа* не зависит от значения адреса или ключа предыдущего обращения. Например, по принципу П. д. организуется обмен данными между *оперативной памятью* и процессором. Ср. *последовательный доступ*

ПРЯМОЙ ФАЙЛ [direct-access file]. То же, что *файл прямого доступа*

ПСЕВДОГРАФИЧЕСКИЙ СИМВОЛ [graphic character]. *Символ*, позволяющий строить простейшие графические изображения на бумаге или экране дисплея в *текстовом режиме*. П. с. применяются при наборе текста для изображения рамок, таблиц и диаграмм, чтобы придать тексту удобную для чтения форму. В табл. П.2 приведены стандартные П. с. и *коды ASCII* этих символов. Код символа равен сумме шестнадцатеричных чисел в первом столбце и первой строке. Ср. *графический символ, управляющий символ*

Таблица П.2. *Псевдографические символы*

hex	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
A0		¦	¦	¦	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛
B0	┐	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬
C0	╧	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	¦	_	¦	‾
D0	¦															

ПСЕВДОДИСК [virtual disk]. То же, что *электронный диск*

ПСЕВДОКОД [pseudocode]. Язык, подобный естественному и вместе с тем достаточно формальный для того, чтобы на нем можно было описывать алгоритмы

ПСЕВДОНИМ [alias]. Имя или метка, применяемые в *информационных системах*, *системах управления базами данных* и в *компьютерных сетях* в качестве альтернативного способа указания кого-либо или чего-либо. В сетях П. идентифицирует отдельного пользователя или группу пользователей. При использовании группового П. сообщение попадает ко всем членам группы, что упрощает рассылку информации нескольким получателям

ПУЛ [pool]. То же, что *пул памяти*

ПУЛ ПАМЯТИ, пул [pool]. Область *оперативной памяти*, выделяемая для организации нескольких процессов

ПУНКТ [point]. Основная единица полиграфической системы мер. Равна 1/72 дюйма. Служит в основном для измерения размеров шрифта. См. *твип*

ПУНКТ МЕНЮ [choice, menu item]. Элемент *меню*. Один из его вариантов, предлагаемых пользователю для выбора. Каждый П. м. обозначает какое-то действие или набор действий системы, поэтому, выбирая тот или иной П. м., пользователь тем самым определяет последующие действия системы

ПУСТОЙ ОПЕРАТОР [dummy statement, null statement]. *Оператор*, не задающий никаких действий и не представляющий описания данных. Например, в языке Фортран *оператор продолжения* не вызывает никаких действий, а используется только для размещения метки. П. о. называют отсутствие оператора на месте, предписываемом правилами синтаксиса. Например, в программе на языке Си возможен такой *оператор цикла*, определяющий первый делитель целого значения переменной `nmb`:

```
for (div = 2; nmb % div != 0; div++);
```

Здесь все необходимые вычисления выполняются внутри *списка цикла*, поэтому *тело цикла* (между закрывающейся скобкой и точкой с запятой) является П. о. Ср. *невыполняемый оператор*

ПУТЬ [path]. *Данные*, указывающие *операционной системе* место в памяти, где следует искать файл. Обычно П. указывает *логическое имя* накопителя и последовательность имен вложенных друг в друга каталогов, в последнем из которых содержится нужный файл. Например, в *персональных компьютерах* П., по которому отыскивается файл, содержится в *полном имени файла*

ПУТЬ ДОСТУПА [access path]. 1. Список каталогов, в которых следует искать файл. 2. *Маршрут*, состоящий из *узлов сети*, через которые должно пройти передаваемое сообщение

ПЭВМ (ПК) [personal computer (PC)]. Персональная *электронная вычислительная машина*. То же, что *персональный компьютер*

Р

РАБОЧАЯ КНИГА [workbook]. Название документа приложения Microsoft Excel. Р. к. состоит из нескольких рабочих листов, каждый из которых является *электронной таблицей* с широкими возможностями обработки данных и построения *диаграмм*. На рис. Р.1 представлен вид открытой Р. к. MS Excel

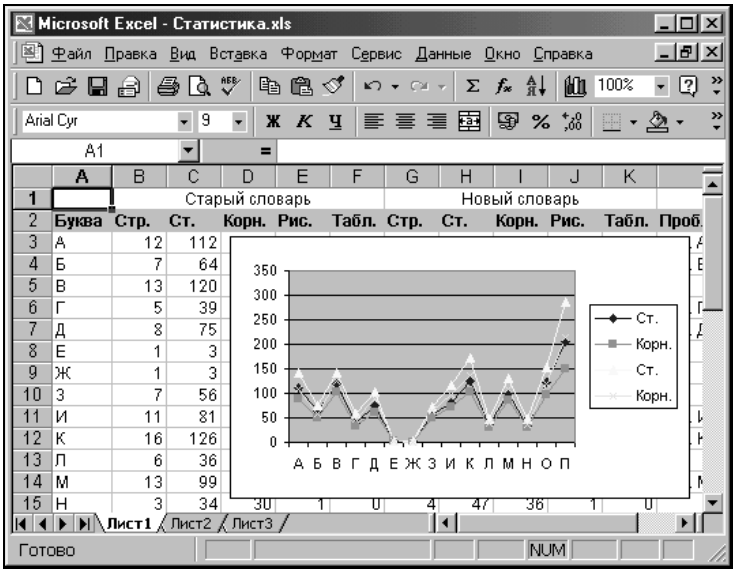

Рис. Р.1. Рабочая книга Microsoft Excel

РАБОЧАЯ ОБЛАСТЬ [work area, workspace]. *Область оперативной памяти*, выделяемая *прикладной программе* для временного размещения данных в течение обработки. См. *распределение памяти*

РАБОЧАЯ СРЕДА ПОЛЬЗОВАТЕЛЯ, среда [working environment, user environment]. Набор параметров *операционной системы* и совокупность *программных средств*, с которыми обычно работает пользователь. Р. с. п. устанавливается с помощью *сценария регистрации* при *входе в систему*. См. *профиль пользователя*

РАБОЧАЯ СТАНЦИЯ [work station]. 1. ЭВМ, рассчитанная на одного пользователя, но более мощная, чем *персональный компьютер*. Обычно используется как профессионально ориентированное *автоматизированное рабочее место*. 2. Компьютер, включенный в *локальную вычислительную сеть*

РАБОЧИЙ КАТАЛОГ [working directory]. То же, что *текущий каталог*

РАБОЧИЙ НАБОР ПРОЦЕССА [working set]. Набор страниц *виртуальной памяти*, принадлежащих данному процессу и всегда находящихся в *оперативной памяти*. Р. н. п. обязательно должен присутствовать в памяти, чтобы задача могла выполняться

РАБОЧИЙ СТОЛ [desktop]. Метафорическое название приема организации *графического интерфейса пользователя*, существующего, например, в *операционной системе Windows*. В начальный момент работы системы на экране компьютера располагаются *значки* приложений, документов и т. д. Пользователь имеет удобный доступ к объектам, значки которых размещены на Р. с. Например, двойной щелчок кнопкой мыши запускает приложение или открывает документ. Пользователь может добавлять на Р. с. значки, удалять их с Р. с., перемещать значки, располагая их удобным для себя образом, и т. д.

РАБОЧИЙ ФАЙЛ [temporary file (temp file)]. То же, что *временный файл*

РАДИАЛЬНАЯ СЕТЬ, звездообразная сеть, сеть звездообразной топологии [star network]. *Локальная вычислительная сеть*, в которой каждый *узел сети* соединен с центральным компьютером (*концентратором* или *хабом*). В такой сети сообщения проходят от узла к концентратору, который вырабатывает маршрут передачи сообщения другому узлу. См. *топология сети*. Ср. *древовидная сеть, кольцевая сеть, шинная сеть*

РАДИОКНОПКА, переключатель [option button]. *Элемент управления*, предназначенный для выбора лишь одного из нескольких возможных значений. На рис. Э.2 приведен пример использования Р. в *диалоговом окне*. Ср. *флажок*

РАЗАРХИВИРОВАНИЕ [unpack]. Процедура, обратная *архивированию*, — восстановление сжатых файлов из *архивного файла*. См. *архив, сжатие файлов*

РАЗБИЕНИЕ НА СТРАНИЦЫ [pagination]. Операция, в процессе которой определяется, где в документе начинается и заканчивается каждая страница. При *верстке страниц* изменения, внесенные в одну из первых страниц документа, могут существенно повлиять на Р. н. с. в остальной части документа. Некоторые *тестовые процессоры* производят Р. н. с. автоматически, в фоновом режиме. Другие приложения производят Р. н. с. по явной команде пользователя

РАЗВЕРТЫВАНИЕ [deployment]. Фаза *процесса разработки программного обеспечения*, в которой описывается конфигурация системы, работающей в реальном окружении. На стадии Р. принимаются решения относительно параметров конфигурации, размещения ресурсов, распределения оборудования и параллелизма

РАЗВИЛКА [fork]. Ситуация, при которой *поток управления* разветвляется на несколько потоков управления. В *унифицированном языке моделирования* UML Р. обозначается символом "развилка" (см. *элемент модели*) и применяется на *диаграммах деятельности* для описания *параллелизма*. Противоп. *соединение*. Ср. *ветвление*

РАЗГОВОР В СЕТИ ИНТЕРНЕТ, чат [chat]. Текстовый диалог в реальном времени между пользователями Интернета. Каждый участник разговора

"высказывается" с помощью клавиатуры, получая ответы в специальном окне экрана

РАЗДЕЛ [partition]. 1. То же, что *раздел диска*. 2. То же, что *раздел памяти*

РАЗДЕЛ ДИСКА, раздел [partition]. *Раздел памяти*, выделяемый на *жестком диске* для определенного использования. Например, *загрузочный раздел*. Р. д. фактически являются *логические диски*, на которые обычно делят жесткий диск большой емкости

РАЗДЕЛЕНИЕ ВРЕМЕНИ [time sharing]. Режим мультипрограммирования, при котором *время центрального процессора* предоставляется различным задачам (процессам) последовательно *квантами времени*. Р. в. реализуется *системами* Р. в. — *многозадачными операционными системами*, способными обеспечить этот режим. При Р. в. процессорное время распределяется по некоторому алгоритму. Чаще всего применяется циклический алгоритм, который состоит в следующем. Каждый процесс периодически получает короткий квант времени, в течение которого он имеет возможность использовать *центральный процессор*, причем предотвращаются любые прерывания, связанные с событиями, происходящими вне этого процесса. Как только процесс полностью израсходует это время, он прерывается таймером и временно откладывается, даже если он не закончил свою работу. Если же процесс прерывается из-за внутренних событий и блокируется до истечения кванта времени, то оставшаяся часть этого кванта оказывается для него потерянной и центральный процессор предоставляется следующему по очереди процессу. Очередь готовых к работе процессов организуется в виде кольца. При этом планировщик задач каждый раз продвигается по кольцу на один шаг, чтобы запустить следующий процесс. Когда ранее заблокированный процесс оказывается готовым к работе, он добавляется в это кольцо перед текущим выполняемым процессом, т. е. в конец кольцевой очереди. Если какой-то процесс отложен, а в оперативной памяти не хватает места для следующего, то *образ* отложенной *задачи* запоминается на *внешнем устройстве прямого доступа* и память освобождается (см. *свопинг*). Длительность кванта времени, обычно составляющая доли секунды, и возможное количество одновременно обслуживаемых задач подбираются так, чтобы центральный процессор оказывался все время занятым, а *время прогона* каждой задачи было разумным. На основе Р. в. работают многие *системы коллективного пользования*. Ср. *квантование времени*

РАЗДЕЛ ПАМЯТИ [division, partition]. *Область памяти*, выделяемая для определенного использования. В режиме мультипрограммирования — область *оперативной памяти*, выделяемая отдельному заданию

РАЗДЕЛЯЕМАЯ ПАМЯТЬ, общая память [shareable memory, shared memory]. 1. В *многозадачной операционной системе — память,* доступная для использования несколькими программами. 2. В многопроцессорной ЭВМ —

память, доступная для использования несколькими процессорами. В простейшем случае это оперативная память, с которой процессоры соединены одной общей шиной. Физическая память может состоять из нескольких модулей, хотя виртуальное *адресное пространство* остается общим. В этом случае вместо общей шины применяется переключатель, направляющий запросы от процессора к нужному модулю памяти. Ср. *чередуемая память, распределенная память*. См. *разделяемый ресурс*

РАЗДЕЛЯЕМЫЙ РЕСУРС [shareable resource, shared resource]. То же, что *общий ресурс*

РАЗМЕРНОСТЬ МАССИВА [dimension]. Количество индексов, необходимое для однозначной идентификации любого *элемента массива*. Массив, элемент которого — переменная с одним индексом, называется *одномерным массивом*, с двумя индексами — двумерным и т. д. Одномерный массив можно представить себе как строку элементов:

```
T(1) T(2) T(3) T(4)
```

Двумерный массив можно представить себе состоящим из горизонтальных строк и вертикальных столбцов:

```
T(1,1) T(1,2)
T(2,1) T(2,2)
```

В таком случае первый из двух индексов задает номер строки, второй — номер столбца. В *описании массива* Р. м. указывается числом *граничных пар* или верхних *границ индексов*

РАЗМЕР РЕГИСТРА [register size]. То же, *что емкость регистра*

РАЗМЕР ШРИФТА, кегль шрифта [font size, type size]. Высота символов шрифта, измеряемая в пунктах (1 пункт = 1/72 дюйма, или примерно 0,36 мм). Наиболее ходовым размером шрифта считается 10 пунктов. Более мелкие шрифты (8 пунктов и менее) трудно читаются. Следует иметь в виду, что Р. ш. определяет только высоту рамки, в которую должны быть вписаны символы шрифта, но отнюдь не физическую высоту каждого отдельного символа. На рис. Р.2 для сравнения приведено одно и то же слово, набранное в разных шрифтах одного размера. См. *гарнитура, начертание шрифта*

10 пунктов Times New Roman. 10 пунктов Arial. 10 пунктов Courier New.

Рис. Р.2. Сравнение различных шрифтов одного размера

РАЗМЕТКА СТРАНИЦЫ [page layout]. 1. Параметры размещения текстового и иллюстративного материала на странице. В число этих параметров

входят, например, размеры полей страницы, расстояния между *колонтитулами* и основным текстом и т. п. 2. Режим отображения страницы на экране компьютера, при котором видны все параметры Р. с. На принтере страница печатается обычно в режиме Р. с., т. е. с полями, колонтитулами и пр., в то время как на экране страница может отображаться и без учета Р. с., например, без колонтитулов, без изображения пустого пространства полей и пр.

РАЗНОРОДНАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА [heterogeneous computer system]. То же, что *гетерогенная вычислительная система*

РАЗРАБОТКА ПРОГРАММ [program development]. То же, что *программирование*

РАЗРЕШАЮЩАЯ СПОСОБНОСТЬ, разрешение [resolution]. Характеристика качества изображения, которое может быть достигнуто устройством визуального отображения (дисплеем, принтером и т. п.). Р. с. зависит от числа элементов, из которых складывается изображение: чем больше число элементов и чем меньше размер каждого элемента, тем выше качество изображения. Поэтому Р. с. экрана дисплея определяется количеством пикселов (точек), на которые разбит экран (указывается в виде произведения числа точек по горизонтали на число точек по вертикали). Р. с. *принтера* определяется количеством точек на *дюйм* (dpi, dots per inch). См. *графический режим, количество пикселов на дюйм, количество точек на дюйм*

РАЗРЕШЕНИЕ [resolution]. То же, что *разрешающая способность*

РАЗРУШЕННАЯ ССЫЛКА [broken link]. *Ссылка* на ресурс, который не может быть найден. Р. с. возникает, например при работе в Интернете, если *универсальный указатель ресурса* (URL) недействителен, ресурс, на который указывает ссылка, не существует, или же сервер, содержащий ресурс занят или неисправен.

РАЗРЯД [digit]. 1. Место (позиция), занимаемое цифрой при записи числа в *позиционной системе счисления*. 2. Место (позиция) в ячейке памяти для одного бита информации

РАЗРЯДКА [expanded font]. Увеличение интервалов между буквами по сравнению с нормальными интервалами. Р. на значительную величину (сравнимую с шириной букв) может использоваться для выделения фрагментов текста. Р. на небольшую величину используется при *верстке страниц* для уменьшения слишком больших интервалов между словами. В настоящее время этот прием форматирования текста используется сравнительно редко. Противоп. *уплотнение шрифта*. Ср. *кернинг*

РАЗРЯДНОСТЬ [capacity]. Количество разрядов, отведенное для чего-либо. Например, Р. цвета — число двоичных разрядов (битов), предназначенных для кодирования цветов

РАЗРЯДНОСТЬ ЦВЕТА [bit depth]. То же, что *битовая глубина цвета*

РАМКА [frame]. То же, что *фрейм*

РАНГ [rank]. То же, что *номер процесса*

РАСКЛАДКА КЛАВИАТУРЫ [keyboard layout]. Параметр *операционной системы*, определяющий, *коды символов* какого алфавита будут вводиться в память компьютера при нажатии клавиш *клавиатуры*. В отечественной практике используются, как правило, минимум две раскладки: русская (символы кириллицы) и английская (латинские символы). Переключение Р. к. осуществляется *командной клавишей* или комбинацией клавиш. На рис. Р.3 показано диалоговое окно Windows для управления раскладкой клавиатуры. См. *драйвер-русификатор*

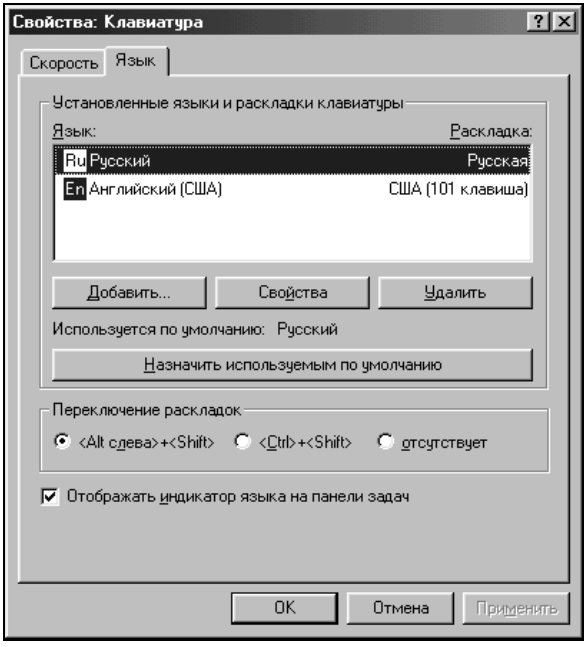

Рис. Р.3. Диалоговое окно для управления раскладкой клавиатуры

РАСКРЫВАЮЩЕЕСЯ МЕНЮ [pull-down menu]. То же, что *нисходящее меню*

РАСКРЫВАЮЩИЙСЯ СПИСОК [drop-down list]. *Элемент управления*, предназначенный для выбора одного или нескольких значений из списка вариантов. Аналогичен элементу управления *список*, но занимает меньше места

на экране. На рис. Э.2 приведен пример использования Р. с. в *диалоговом окне*. Ср. *список*

РАСПЕЧАТКА [printing, listing]. 1. Процесс вывода данных из компьютера с помощью принтера. 2. Данные, выведенные из компьютера в печатном виде (на бумаге или другом подобном носителе). См. *листинг*

РАСПОЗНАВАНИЕ РЕЧИ [speech analysis, speech recognition]. Процесс анализа человеческой речи с целью выделения отдельных фонем, слов предложений и в конечном счете перевод аналогового сигнала в связный текст. Р. р., являясь классической областью применения методов искусственного интеллекта, в настоящее время находит практическое применение. Например, некоторые версии операционной системы Windows снабжены программами Р. р., которые в состоянии правильно понимать и выполнять команды пользователя, подаваемые голосом в микрофон. Последняя версия программы MS Word имеет возможность вводить текст с голоса, т. е. под диктовку пользователя, без клавиатуры. Ср. *синтез речи*

РАСПРЕДЕЛЕНИЕ ПАМЯТИ [memory allocation, storage allocation, memory mapping]. Действия транслятора или *операционной системы* по выделению *областей памяти* программам и обрабатываемым этими программами данным. Различают *статическое и динамическое* Р. п. Статическое Р. п. выполняется до начала выполнения программы, а динамическое — во время выполнения. Область оперативной памяти, предназначенная для размещения прикладных программ, называется *динамической областью памяти*, а область, выделяемая программе для временного размещения данных, — *рабочей областью*. Основными элементами программы и данных, для которых необходима память во время выполнения программы, являются: 1) сама исполняемая программа пользователя; 2) *системные программы*, которые производят вспомогательные действия во время выполнения программы пользователя (от простых *библиотечных программ*, например, вычисляющих значение синуса угла, до программ, управляющих динамическим Р. п.); 3) *структуры данных*, создаваемые или объявленные программой пользователя, и используемые ею константы; 4) *точки входа* и *возврата* для подпрограмм; 5) временная память для промежуточных результатов вычисления выражений и *фактических параметров* подпрограмм; 6) *буферная память* ввода/вывода; 7) различные системные данные (дескрипторы структур данных, списки свободного пространства памяти, информация о состоянии *устройств ввода/вывода* и т. п.). Существуют следующие основные схемы Р. п.: выделение одного непрерывного раздела, разбиение на разделы, разбиение на страницы, сегментация. Выделение одного непрерывного раздела применяется в однозадачных операционных системах. При этом одна выполняемая задача монополизирует всю память и все устройства. При разбиении на разделы каждой задаче выделяется непрерывно адресуемая часть памяти, границы которой могут быть либо фиксированными (мультипрограммиро-

вание с фиксированным числом задач), либо динамически изменяющимися в зависимости от текущей загруженности задачами *вычислительной системы* (мультипрограммирование с переменным числом задач). Схема страничной организации памяти предполагает разбиение адресуемого пространства каждой задачи на страницы, а *физической памяти* — на блоки оперативной памяти или секторы на *магнитном диске*. При этом любая страница может быть помещена в любой блок или сектор. Таким образом организуется *виртуальная память*, которая значительно облегчает реализацию мультипрограммирования и повышает производительность вычислительной системы. В случае схемы сегментации памяти оперативная память делится на сегменты фиксированной длины. Для аппаратуры адрес памяти состоит из двух частей: номера (имени) сегмента и смещения внутри сегмента. Программа и данные также разбиваются на сегменты и могут быть размещены в нескольких сегментах памяти. Учетом и распределением оперативной памяти между программами управляет супервизор основной памяти. См. *управление памятью, виртуальная память, супервизор*

РАСПРЕДЕЛЕНИЕ РЕСУРСОВ [resource allocation]. Действия *операционной системы* по выделению *ресурсов* отдельным заданиям. Выделение ресурсов программе может происходить как до начала ее выполнения *(статическое распределение ресурсов)*, так и в ходе ее выполнения *(динамическое распределение ресурсов)*. Р. р. и управление ими осуществляют специальные *резидентные программы* операционной системы. Когда в режиме мультипрограммирования ресурсы *вычислительной системы* делятся между несколькими программами, то каждая программа должна быть защищена от воздействия со стороны другого задания. Например, каналы ввода/вывода — это ресурсы, которые разрешается совместно использовать несколькими заданиями. Поэтому программа пользователя не может непосредственно сама инициировать операцию ввода/вывода, а должна обратиться к супервизору ввода/вывода и предоставить всю информацию, необходимую для этой операции. См. *распределение памяти, мультипрограммирование, супервизор*

РАСПРЕДЕЛЕННАЯ БАЗА ДАННЫХ [distributed database]. *База данных*, в которой информация распределена между несколькими узлами *компьютерной сети*. Не следует путать Р. б. д. с *сетевой базой данных*

РАСПРЕДЕЛЕННАЯ КОМПОНЕНТНАЯ МОДЕЛЬ ОБЪЕКТОВ [distributed component object model (DCOM)]. Расширение *модели COM*, специально предназначенное для работы в сетевом окружении и Интернете. Технология DCOM определяет методы создания и взаимодействия объектов, хранящихся на разных компьютерах. Средой для передачи данных между объектами является локальная или глобальная вычислительная сеть. При этом поддерживается как синхронная, так и асинхронная передача данных, что позволяет предотвратить остановку работы клиентского приложения при ожидании

отклика от сервера. Технология DCOM может использоваться для интеграции приложений Web. См. *компонентная модель объектов*

РАСПРЕДЕЛЕННАЯ ПАМЯТЬ [allocate memory]. В многопроцессорных *вычислительных системах* — совокупная *оперативная память*, состоящая из памяти отдельных процессоров. При этом каждый процессор имеет доступ только к своей памяти, а обмен данными между процессорами обычно осуществляется при помощи сообщений-посылок по коммуникационной сети, связывающей процессоры. В этом случае нет конфликтов по доступу к памяти и отпадает необходимость в шине или переключателе. Однако обмен данными по сети требует дополнительных затрат времени. Ср. *разделяемая память*

РАСПРЕДЕЛЕННОЕ ПРИЛОЖЕНИЕ [distributed application]. Приложение, которое состоит из нескольких компонентов, выполняющихся на различных компьютерах. См. *архитектура "клиент-сервер", многоуровневая архитектура приложения*

РАСТЕРИЗАЦИЯ [rasterization]. То же, что *растрирование*

РАСТР [raster]. Двумерный массив точек (элементов растра, *пикселов*), упорядоченных в ряды (строки) и столбцы, предназначенный для представления изображения путем окраски каждой точки в определенный цвет. С помощью Р. создается изображение на экране дисплея, *лазерные* и *струйные принтеры* выводят растровое изображение на бумагу. См. *растровая графика.* 2. То же, что *битовый массив*

РАСТРИРОВАНИЕ, растеризация [rasterization]. Преобразование изображения *векторной графики* в *растровое изображение* заданного размера и формата

РАСТРОВАЯ ГРАФИКА [bit-mapped graphics, raster graphics]. Метод создания изображений в виде *растра* — набора разноцветных точек (пикселов), упорядоченных в строки и столбцы. В памяти компьютера такие изображения хранятся в виде битовых последовательностей, которые описывают цвет отдельных пикселов. При этом на каждый пиксел приходится конкретное число бит, определяющих ту или иную характеристику цвета в соответствии с принятой *цветовой моделью* и допустимой палитрой. Р. г. применяется в *программах раскраски*, которые работают с рисунком как с группой точек, а не с совокупностью форм. На принципах Р. г. основана работа экрана дисплея, а также *лазерных* и *струйных принтеров*. Ср. *векторная графика.* См. *цвет, палитра*

РАСТРОВОЕ ИЗОБРАЖЕНИЕ, растровый образ [bitmap image]. Изображение, созданное методом *растровой графики*

РАСТРОВЫЙ ДРАЙВЕР [raster driver]. См. *универсальный драйвер принтера*

РАСТРОВЫЙ ОБРАЗ [bitmap image, pixel image]. 1. То же, что *растровое изображение*. 2. То же, что *битовый массив*

РАСТРОВЫЙ ФАЙЛ [bitmap file]. *Графический файл*, содержащий *битовый массив (растр)* изображения. Кроме того, Р. ф. обладает информацией о *разрешающей способности* устройства, на котором этот массив был создан, о размерах массива и доступной палитре цветов. Среди Р. ф. наиболее распространены файлы стандартных форматов BMP, PCX и TIFF. См. *файл BMP*, *файл PCX* и *файл TIFF*

РАСТРОВЫЙ ШРИФТ [bitmapped font, raster font]. *Шрифт*, символы которого хранятся и воспроизводятся как *растровые изображения*. Р. ш. трудно масштабируются. Ср. *векторный шрифт*

РАСШИРЕНИЕ [extension]. То же, что *расширение имени файла*

РАСШИРЕНИЕ ИМЕНИ ФАЙЛА, расширение [file name extension, extension]. Необязательная часть *имени файла*, предназначенная для указания *типа файла* или описания его содержания. В *операционной системе MS-DOS* Р. и. ф. записывается после имени, отделяется точкой, за которой следуют от 1 до 3 символов. Символы в Р. и. ф. могут быть прописными и строчными латинскими буквами, цифрами и символами: -, _, $, #, &, @, !, %, (,), {, }, ', ~, ^. Пробелы в Р. и. ф. не допускаются. Многие программы, создавая файлы, устанавливают Р. и. ф., по которому можно узнать, что за программа создала файл. Например, exe, com, bat — расширения командных файлов, причем exe относится к программам, созданным компиляторами, doc — документы для редактора MS Word, for, pas, c — программы на Фортране, Паскале и Си соответственно. Многие программы используют расширение bak для копий файла, создаваемых перед его изменением. Пользователь, сохраняя файл, может указать свое расширение, например, let для писем. В других операционных системах действуют иные принципы использования Р. и. ф.

РАСШИРЕННАЯ АРХИТЕКТУРА ПРОМЫШЛЕННОГО СТАНДАРТА, архитектура EISA [extended industry standard architecture (EISA)]. Стандарт на *архитектуру* шины подключения *периферийных устройств*, введенный в 1988 г. Консорциумом компаний — производителей вычислительной техники (так называемой "группой девяти" — AST Research, Compaq, Epson, Hewlett-Packard, NEC, Olivetti, Tandy, Wise, Zenith). Шины EISA являются 32-разрядными и обеспечивают скорость передачи данных 132 Мбайт/сек

РАСШИРЕННАЯ ИНТРАСЕТЬ [extranet]. То же, что *экстранет*

РАСШИРЕННАЯ ПАМЯТЬ [extended memory (XMS)]. Составная часть *оперативной памяти персональных IBM-совместимых компьютеров* при работе в среде MS-DOS. В других ситуациях в настоящее время этот термин не упот-

ребляется. См. *оперативная память персональных IBM-совместимых компьютеров*

РАСШИРЕННЫЙ ТЕКСТОВЫЙ ФОРМАТ [rich text format (RTF)]. То же, что *формат RTF*

РАСШИРЯЕМЫЙ ЯЗЫК РАЗМЕТКИ [extensible markup language (XML)]. То же, что *язык XML*

РЕАЛИЗАЦИЯ [implementation]. 1. Определение конструкции или алгоритма вычисления чего-либо. Например, класс является Р. типа, метод — Р. операции, лазерный диск — Р. памяти. Ср. *спецификация*. 2. Фаза *процесса разработки программного обеспечения*, в которой описывается функционирование системы в некой среде (например, на *языке программирования*, в *базе данных* или *аппаратном обеспечении*)

РЕАЛЬНЫЙ РЕЖИМ [real mode]. Режим работы *центрального процессора персонального IBM-совместимого компьютера*, при котором могут выполняться программы, написанные для компьютеров, имеющих устаревший процессор 8088/86. Чтобы обеспечить такую возможность, современные микропроцессоры при работе в среде MS-DOS вынуждены имитировать (эмулировать) микропроцессор 8088/86 со всеми его ограничениями. Например, в Р. р. можно полноценно использовать только 640 Кбайт *оперативной памяти*. Начиная с версии 3.0, Windows не поддерживает этот режим работы. В других ситуациях в настоящее время данный термин не применяется

РЕГЕНЕРАЦИЯ [regeneration]. То же, что *восстановление*

РЕГИОНАЛЬНАЯ ВЫЧИСЛИТЕЛЬНАЯ СЕТЬ [wide area network (WAN)]. Нелокальная *вычислительная сеть*, соединяющая территориально удаленные компьютеры, что позволяет им совместно использовать *программное обеспечение, базы данных,* файлы и т. д. Современная Р. в. с. служит для связи нескольких *локальных вычислительных сетей*. Бóльшая часть Р. в. с. эксплуатирует для *передачи данных* волоконно-оптические кабели, выделенные телефонные линии или ультракоротковолновые радиоканалы. Противоп. *локальная вычислительная сеть*

РЕГИСТР [register]. Внутреннее *запоминающее устройство* процессора или адаптера для временного хранения обрабатываемой или управляющей информации и обеспечения быстрого доступа к ней. Часто Р. располагает тем же размером, что и *машинное слово*, но может иметь и бóльшую или меньшую длину, равную целому числу байтов. Р. служат для выполнения определенных функций. Существуют Р., на которые из *оперативной памяти* вызываются исполняемые команды или их составные части *(регистр команд, регистр адреса)*, Р., на которые вызываются операнды, Р., играющие роль счетчиков, стеков и т. п.

РЕГИСТР АДРЕСА, адресный регистр [address register]. *Регистр* внутренней памяти *центрального процессора*, содержимое которого соответствует некоторому *адресу*. Это может быть *адрес команды*, вызываемой для выполнения, или адрес операнда, указанный в команде. Р. а. служат для вычисления *физических адресов* команд и операндов выполняемой программы. Иногда Р. а. рассматривается как часть *регистра команды*. См. *адресация*

РЕГИСТРАЦИЯ [login, logon]. То же, что *вход в систему*

РЕГИСТР БАЗЫ [base register]. То же, что *базовый регистр*

РЕГИСТР КОМАНД [instruction register]. *Регистр* процессора, в котором находится исполняемая в данный момент команда. Она вызывается на Р. к. из *оперативной памяти*

РЕГРЕССИОННОЕ ТЕСТИРОВАНИЕ [regression testing]. Процесс проверки того факта, что добавление кода в программу, находящуюся в разработке, не ухудшает тех возможностей, которыми программа уже обладает. См. *испытание программы*

РЕДАКТИРОВАНИЕ [editing]. Внесение изменений в текст или преобразование программ или данных к виду, требуемому для их дальнейшего использования. Р. текста состоит из перемещения, удаления и изменения его отдельных символов и фрагментов, а также ввод нового текста и изменение формата. Р. текста выполняется с помощью специальных программ — *текстовых редакторов* и *текстовых процессоров*. Р. *исходного текста программ* и подготовка *исходных данных* выполняется с помощью редакторов, входящих в состав *системы программирования*

РЕДАКТИРОВАНИЕ СВЯЗЕЙ [linkage editing]. То же, что *компоновка*

РЕДАКТОР ИЗОБРАЖЕНИЙ [graphics editor, bitmap editor]. То же, что *графический редактор*

РЕДАКТОР СВЯЗЕЙ [linkage editor]. То же, что *компоновщик*

РЕДАКЦИОННО-ИЗДАТЕЛЬСКАЯ СИСТЕМА [publishing system]. См. *настольная редакционно-издательская система*

РЕЕНТЕРАБЕЛЬНОСТЬ, повторная входимость [reenterability]. Свойство подпрограммы или *программного модуля*, позволяющее нескольким вызывающим программам независимо друг от друга вызывать один и тот же экземпляр Р. подпрограммы. Например, в многопоточной среде реентерабельная подпрограмма может быть вызвана из одной программы, затем управление может быть передано в другую программу, в которой снова может быть вызвана та же самая подпрограмма до того, как закончится выполнение подпрограммы, инициированное первым вызовом. Обычно Р. достигается тем, что совокупность данных, от которых зависит ход выполнения

подпрограммы, в частности, значения *локальных переменных* и *фактические параметры* подпрограммы, хранятся в вызывающей программе, а не в подпрограмме. Использование реентерабельных подпрограмм позволяет экономить ресурсы оперативной памяти, необходимой для размещения программного кода, поэтому Р. часто используется в *системных программах*

РЕЕСТР, реестр Windows [Registry, Windows Registry]. *Иерархическая база данных*, в которой *операционная система Windows* централизованным образом хранит всю системную информацию, например, *конфигурацию* вычислительной системы, сведения об установленных программах, значения различных параметров, настраиваемых пользователем, и пр. Пользователь может менять информацию в Р., например, с помощью программы regedit, окно которой показано на рис. Р.4, однако следует иметь в виду, что неквалифицированное изменение данных в Р. может привести *вычислительную систему* к неработоспособному состоянию

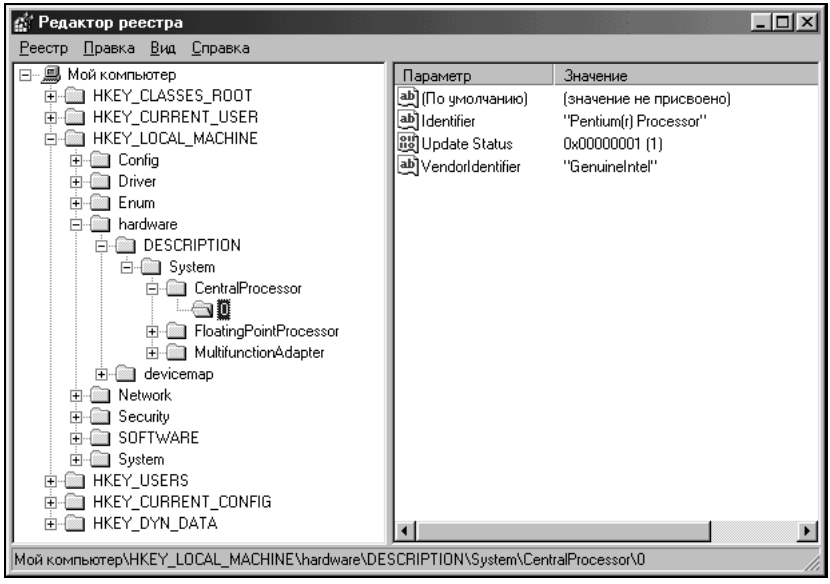

Рис. Р.4. Редактор реестра Windows

РЕЕСТР WINDOWS [Registry, Windows Registry]. То же, что *реестр*

РЕЖИМ ВСТАВКИ [insert mode]. Режим работы клавиатуры, при котором вводимый символ помещается между соседними символами: символы строки, которые находились над курсором и правее, и сам курсор сдвигаются на

один символ вправо, а вводимый символ становится в позицию, в которой находился курсор. Ср. *режим замещения*

РЕЖИМ ЗАМЕЩЕНИЯ [overtype mode]. Режим работы клавиатуры, при котором вводимый символ замещает собой символ, указываемый текстовым курсором. Замещенный символ теряется. Ср. *режим вставки*

РЕЖИМ ОПЕРАЦИОННОЙ СИСТЕМЫ [privileged mode, kernel mode]. То же, что *привилегированный режим*

РЕЖИМ ПОЛЬЗОВАТЕЛЯ, пользовательский режим [user mode]. *Непривилегированный режим* работы процессора, в котором исполняется код *прикладных программ*. *Процесс*, выполняющийся в Р. п., может получить доступ к системным ресурсам только посредством вызова сервисов *операционной системы*

РЕЖИМ ЯДРА [privileged mode, kernel mode]. То же, что *привилегированный режим*

РЕЖИМ ON-LINE [online mode, on-line mode]. То же, что *онлайновый режим*

РЕЗЕРВИРОВАНИЕ [standby]. Включение в систему дублирующих, резервных устройств для повышения ее надежности. См. *избыточность*. 2. Создание *резервной копии*

РЕЗЕРВИРОВАНИЕ УСТРОЙСТВА [device allocation]. Закрепление устройства за процессом или программой с целью *монопольного использования*

РЕЗЕРВНАЯ КОПИЯ [backup]. Запасная копия содержимого диска, программы, файла, документа или некоторых данных, создаваемая для использования в случае повреждения оригинала. Например, Р. к. *базы данных* используется *системой управления базой данных* для восстановления базы после *аппаратного сбоя* или *программных ошибок*. Некоторые программы автоматически создают Р. к. обрабатываемого файла, сохраняя на диске и его текущее состояние, и предыдущий вариант. В архиве рекомендуется хранить Р. к. наиболее ценных файлов

РЕЗИДЕНТНАЯ ПРОГРАММА [resident program, resident software]. *Программа*, постоянно присутствующая в *оперативной памяти*. После считывания в оперативную память Р. п. не выполняется до конца, а остается в состоянии готовности к продолжению выполнения до своей выгрузки, перезагрузки или выключения компьютера. Она активизируется и выполняет заданные действия либо при обращении к ней, например, нажатием определенной клавиши или сочетания клавиш, либо при достижении компьютером определенного состояния (например, по сигналу таймера). Примерами Р. п. могут служить программы обработки ситуаций нажатия пользователем клавиш на клавиатуре. Такая программа проверяет, не нажата ли опреде-

ленная комбинация клавиш, и если да, то вызывает некоторую связанную с этой комбинацией программу, а если нет — передает управление *операционной системе* для обработки нажатия клавиш. Р. п. должны занимать небольшой *объем памяти* и нетривиально взаимодействовать с операционной системой. Поэтому они разрабатываются квалифицированными программистами, как правило, на языках ассемблера и Си

РЕЗИДЕНТНЫЙ ПУЛ [nonpaged pool]. Часть *оперативной памяти*, страницы которой никогда не откачиваются на диск. Противоп. *нерезидентный пул*. См. *выгрузка*

РЕКУРСИВНАЯ ПОДПРОГРАММА [recursive subroutine]. *Подпрограмма*, которая вызывает саму себя. В *языках программирования высокого уровня* понятию Р. п. соответствуют понятия *рекурсивная процедура* и *рекурсивная функция*. Например, написанная на языке Паскаль функция `Factorial(n)`, вычисляющая значение `n!`:

```
function Factorial(n: integer): integer;
if n = 0 then Factorial := 1 else Factorial := n * Factorial(n-1);
```

содержит в *теле функции* обращение к самой себе: `Factorial(n-1)`. Не всякий язык программирования допускает Р. п., т. к. для этого требуется специальное *программное* и *аппаратное обеспечение*, позволяющее "развернуть" подпрограмму при обращении к ней и затем восстановить ее первоначальное состояние. Например, Паскаль и Си допускают Р. п., а Фортран — нет. Применение Р. п. приводит не только к более простому написанию программы, но и к ее более легкому пониманию пользователями, не участвовавшими в разработке этой программы. Однако на большинстве компьютеров Р. п. выполняется существенно медленнее, чем ее нерекурсивный аналог, и отладка Р. п. может оказаться весьма затруднительной. Поэтому без нужды не следует пользоваться рекурсивными алгоритмами. Например, для вычисления `n!` предпочтительным является применение цикла:

```
function Factorial(n: integer): integer;
var i, f: integer;
begin
  f := 1;
  for i:= 1 to n do f:= f * i;
  Factorial:= f
end;
```

С другой стороны, существует целый ряд случаев, когда замена рекурсии циклом затруднительна или невозможна

РЕКУРСИВНАЯ ПРОЦЕДУРА [recursive procedure]. *Процедура*, в которой имеется обращение к самой себе. См. *рекурсивная подпрограмма*

РЕКУРСИВНАЯ ФУНКЦИЯ [recursive function]. *Функция*, в которой имеется обращение к самой себе. См. *рекурсивная подпрограмма*

РЕЛЕВАНТНОСТЬ [relevancy]. Мера, определяющая, насколько полно тот или иной документ отвечает критериям, указанным в запросе пользователя. Разумеется, далеко не всегда документ, признанный *поисковой системой* наиболее релевантным, будет таким по мнению самого пользователя

РЕЛЯЦИОННАЯ БАЗА ДАННЫХ [relational database]. *База данных*, организованная по реляционной *модели данных*. Р. б. д. состоит из таблиц, каждая из которых представляет собой множество *кортежей* одинаковой структуры. Такую таблицу можно интерпретировать как отношение (в математическом смысле) между *атрибутами* таблицы. Отсюда происходит название Р. б. д. (от англ. *relation*). Связи между кортежами различных таблиц в Р. б. д. устанавливаются неявно, по совпадению значений атрибутов в разных таблицах (в отличие от явных связей между записями в иерархической и сетевой базах данных). Ср. *иерархическая база данных, сетевая база данных*

РЕЛЯЦИОННАЯ СУБД [relational data base management system (RDBMS)]. То же, что *система управления реляционными базами данных*

РЕПЛИКА [replica]. См. *репликация базы данных*

РЕПЛИКАЦИЯ БАЗЫ ДАННЫХ [database replication, replication]. Специальная операция, которая применяется при ведении *распределенных баз данных*. При Р. б. д. создается копия базы данных, которая называется *репликой*. После этого как в оригинал базы данных, так и в реплику могут независимо вноситься различные изменения. Помимо самих изменений в базе данных сохраняется информация о моментах времени, когда производились изменения. Затем выполняется операция *синхронизации* реплики с оригинальной базой данных, которая переносит все изменения из реплики в оригинальную базу данных с учетом изменений, сделанных в оригинальной базе с момента создания реплики. Таким образом, после синхронизации оригинальная база данных оказывается в точности в том состоянии, которое бы она имела, если бы все изменения (в том числе изменения, произведенные в реплике) выполнялись непосредственно в оригинальной базе данных. Р. б. д. применяется, например, в том случае, когда пользователь корпоративной базы данных временно не может иметь к ней доступа (находится в командировке). Тогда можно произвести Р. б. д., скопировать реплику на *портативный компьютер* этого пользователя, а в тот момент, когда снова появляется возможность доступа к основной базе (сотрудник вернулся из командировки), производится операция синхронизации

РЕПОЗИТОРИЙ [repository]. 1. Каталог для хранения, обслуживания, управления и защиты моделей *баз данных*. В Р. хранятся определения данных и их взаимосвязей, код приложения и информация о *правах доступа* к базе данных. 2. Место для хранения моделей, интерфейсов, компонентов, программ и других артефактов, возникающих в *процессе разработки программного обеспечения*

РЕСТАРТ [restart]. То же, что *перезапуск*

РЕСУРС [resource]. Время, данные, *аппаратные*, *программные* и другие *средства*, которые могут быть предоставлены компонентами *вычислительной системы или сети* вычислительному процессу или пользователю. Например, Р. являются *время центрального процессора*, *область* оперативной или *внешней памяти* и *устройства ввода/вывода*, которые могут быть выделены для работы некоторой программы, а также общедоступная информация, хранящаяся на некотором сервере Интернета

РОБОТ ПОИСКОВОЙ СИСТЕМЫ [spider, grawler, bot, robot]. Неотъемлемая составляющая *поисковой системы* — программа, которая посещает *веб-страницы*, считывает (индексирует) полностью или частично их содержимое и далее следует по ссылкам, найденным на данной странице. Р. п. с. возвращается к индексированной странице через определенные периоды времени (например, каждый месяц) и индексирует страницу снова. Вся информация заносится роботом в *индексы поисковой системы*

РОДИТЕЛЬСКОЕ ОКНО [parent window]. *Окно*, которое имеет *дочерние окна*. Другими словами, окно, которое "обладает" другими окнами. Например, окно приложения MS Excel является Р. о. и может иметь несколько окон документов, которые являются дочерними окнами

РУКА [hand]. *Инструмент машинной графики*, предназначенный для передвижения рисунка, превышающего размер текущего рабочего окна. Р. является альтернативой *линейки прокрутки*. Кроме того, Р. позволяет масштабировать рисунок относительно рабочего окна. На рис. Р.5 для *графического редактора* Adobe Photoshop показана кнопка вызова Р.

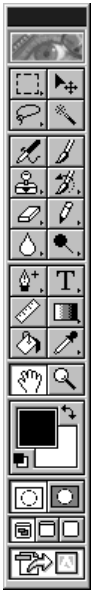

Рис. Р.5. Нажатая кнопка вызова инструмента **Рука** графического редактора Adobe Photoshop

С

САЙТ, веб-сайт, сайт Web [site, Web site]. Русская транскрипция английского слова "site" (буквально переводится с английского словом "место"). Обычно используется словосочетание веб-сайт. Место во *Всемирной паутине*, где можно найти какую-либо информацию. В зависимости от контекста термин С. относится либо к набору страниц, объединенных тематически, принадлежащих одному автору, организации или пользователю, либо к компьютеру *(узлу сети)*, на котором хранится определенная информация и к которому открыт доступ через *Интернет*. См. *веб-документ*

САМОДИАГНОСТИКА ПРИ ВКЛЮЧЕНИИ ПИТАНИЯ [power-on self test (POST)]. То же, что *тест при включении питания*

САМОТЕСТИРОВАНИЕ ПРИ ВКЛЮЧЕНИИ [power-on self test (POST)]. То же, что *тест при включении питания*

СБОЙ [failure]. То же, что *машинный сбой*

СБОРКА МУСОРА [garbage collection]. То же, что *чистка памяти*

СБРОС КЭША [cache flushing]. То же, что *сброс кэш-памяти*

СБРОС КЭШ-ПАМЯТИ, **сброс кэша [cache flushing]**. Принудительная запись на диск всего содержимого *кэш-памяти*

СВЕТОВОЕ ПЕРО [light pen]. *Устройство ввода* координатной информации. По форме напоминает карандаш, снабженный кнопкой и соединенный проводом с компьютером. На конце С. п. находится светочувствительный элемент. С его помощью на экране дисплея можно определить координаты точки, на которую указывает С. п., и ввести их в память компьютера. Поэтому С. п. является *указательным устройством* абсолютного типа. Конец С. п. можно использовать в качестве курсора, а кнопку — вместо клавиши <Enter>. Кроме того, С. п. можно "рисовать" изображение на экране, сохраняя его в памяти компьютера. См. *графический интерфейс пользователя*. Ср. *перо, сенсорный экран, графический планшет*

СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ [freeware, public domain]. То же, что *бесплатное программное обеспечение*

СВОДНАЯ ТАБЛИЦА [pivot table]. Интерактивная таблица на рабочем листе MS Excel, которая позволяет представлять данные других таблиц в компактном и удобном для анализа виде. При составлении С. т. используются различные суммирующие и обрабатывающие функции. Достоинством С. т. является возможность легкого изменения структуры представления данных с помощью *перетаскивания мышью* полей сводной таблицы

СВОЙСТВО КЛАССА [class property, property]. Термин *объектно-ориентированного программирования*, которым обозначается элемент данных, являющийся членом класса. См. *класс*. Ср. *метод класса*

СВОПИНГ [swapping]. 1. Метод эффективного использования *оперативной памяти*, заключающийся в переписывании из нее информации во *внешнюю память* в те периоды, когда эта информация не нужна, и считывании информации обратно в оперативную память, когда в ней снова возникает необходимость. При помощи С. *операционная система* обеспечивает присутствие используемой в данный момент информации в оперативной памяти. На основе С. организуется *виртуальная память* путем перемещения страниц *(страничный обмен)* или сегментов. Кроме того, С. применяется в *многозадачных операционных системах* в случае, когда необходимо освободить место в оперативной памяти под размещение более приоритетной задачи. Для чего менее приоритетная или ждущая задача сохраняется во внешней памяти, а на ее место переписывается более приоритетная. При этом может происходить перемещение *образов задач*. 2. То же, что *подкачка*

СВЯЗАННЫЙ ОБЪЕКТ [linked object]. *Объект*, вставленный в документ-контейнер, но остающийся в исходном файле-источнике. При этом в документе-контейнере хранится ссылка на документ-источник. Связанные данные автоматически обновляются при их изменении в источнике. Ср. *внедренный объект*

СВЯЗЫВАНИЕ И ВНЕДРЕНИЕ ОБЪЕКТОВ, технология OLE [object linking and embedding, OLE]. Технология, разработанная корпорацией Microsoft и реализованная в *операционной системе Windows* и программном обеспечении Macintosh фирмы Apple, используемая для создания динамически обновляемых связей между документами, а также для внедрения документа, созданного одним приложением, в документ, образованный другим приложением. Документ, в который внедрен объект, называется контейнером, а документ, из которого взят внедренный объект, именуется источником. Если объект связан по технологии OLE с неким контейнером, то последний содержит ссылку на этот объект, а любые изменения в источнике отражаются в контейнере. Если объект внедрен по технологии OLE, то контейнер содержит копию объекта, и последующие изменения объекта в источнике не будут отражены в контейнере. См. *элементы управления ActiveX, внедренный объект, связанный объект*

СВЯЗЫВАНИЕ, ПРИВЯЗКА ПРОТОКОЛА [binding]. Процесс, который устанавливает *канал связи* между драйвером протокола (например, *протокола TCP/IP*) и *сетевой картой*

СГЛАЖИВАНИЕ [anti-aliasing, smooth]. 1. Удаление ступенчатости в изображении линий. Причинами ступенчатости могут быть: недостаточная *разрешающая способность* экрана, негладкое сопряжение кривых или некачест-

венное рисование линии на экране с помощью инструмента построения кривых, например, *карандаша*. Для устранения указанных недостатков изображения линий в *графических редакторах* существуют команды С. линий. 2. Смягчение цветового перехода между различными участками изображения. Это достигается путем усреднения *тона, насыщенности, яркости цвета* пикселов или других элементов изображения, находящихся в местах такого перехода. В *графических редакторах* предусматриваются специальные команды или опции, вызывающие процесс С. Например, в инструменте "заливка" для смягчения перехода между областью заливки и рисунком обычно предусматривается опция С. При этом граничные точки делаются полупрозрачными и сливаются с точками рисунка

СЕАНС [session]. 1. Время, в течение которого пользователю выделены ресурсы *вычислительной системы* (от момента входа в систему до момента выхода из нее). 2. Время, в течение которого существует соединение между приложениями, компьютерами или *узлами сети*

СЕАНС СВЯЗИ [session]. Процесс установки, использования и завершения связи между приложениями, устройствами, компьютерами или *узлами сети* для *передачи данных*

СЕАНСОВЫЙ УРОВЕНЬ [session layer]. Пятый из семи уровней *модели ISO/OSI*, стандартизирующей межкомпьютерную связь. Позволяет двум приложениям на разных компьютерах установить, использовать и завершить соединение, называемое сеансом. На С. у. определяется передающая сторона, момент начала передачи и ее продолжительность. С. у. координирует и регулирует *передачу данных*, работает с элементами, которые должны быть согласованы, и с устройствами обмена. Обеспечивает распознавание имен и функции безопасности, выполняет синхронизацию пользовательских задач, помещая в *поток данных контрольные точки*. В случае сбоя выполняется повторная передача данных, следующих за последней контрольной точкой

СЕГМЕНТ [segment]. 1. Часть программы, данных, памяти или изображения на экране дисплея. См. *сегмент программы, сегмент перекрытия, корневой сегмент, сегмент памяти*. 2. То же, что *сегмент сети*

СЕГМЕНТАЦИЯ [segmentation]. 1. Деление памяти, программы, данных на *сегменты*. 2. Способ *распределения памяти операционной системой*. См. *распределение памяти*

СЕГМЕНТ КОНВЕЙЕРА, ступень конвейера [pipeline segment]. Участок *конвейера*, на котором выполняется одна подоперация. См. *конвейерная обработка*

СЕГМЕНТ ПАМЯТИ, сегмент [memory segment]. 1. Часть *оперативной памяти* фиксированного размера, используемая при формировании *адресного пространства центрального процессора* или *задачи* таким образом, что для

аппаратуры адрес памяти состоит из двух частей: номера (имени) сегмента и смещения внутри сегмента. Обычно С. п. имеет размер максимально возможного смещения (часто это 64 Кбайт). См. *распределение памяти.* 2. Логическая или физическая единица подкачки в системе с *виртуальной памятью.* Часть физической или виртуальной памяти, используемая как единое целое при перемещении участков программ или данных из *внешней памяти* в оперативную память и обратно. С. п. отличается от страницы тем, что его размер не зафиксирован, и пользователь в определенной мере может влиять на этот размер. См. *виртуальная память, подкачка, свопинг*

СЕГМЕНТ ПЕРЕКРЫТИЯ, оверлейный сегмент, оверлей [overlay segment]. *Сегмент машинной программы*, загружаемый в ходе ее выполнения в область *оперативной памяти*, ранее занятую другими сегментами этой же программы. См. *оверлейная программа, оверлейная структура, перекрытие*

СЕГМЕНТ ПРОГРАММЫ [program segment]. Часть программы, отдельно обрабатываемая загрузчиком. В *оверлейных программах* существуют *корневой сегмент* и *сегменты перекрытий*

СЕГМЕНТ СЕТИ, сегмент [network segment, segment (SEG)]. Отрезок сети между двумя компьютерами

СЕКТОР [sector]. Участок *дорожки магнитного диска*, имеющий свой *физический адрес.* Таким образом С. является минимальным физически адресуемым элементом памяти на магнитном диске. См. *форматирование диска*

СЕКЦИЯ ПАМЯТИ [memory section]. То же, что *банк памяти*

СЕМАНТИКА ЯЗЫКА ПРОГРАММИРОВАНИЯ [programming language semantics]. Совокупность правил, определяющих смысл чисел, переменных, выражений, операторов, процедур и других элементов и предложений данного *языка программирования*

СЕМАНТИЧЕСКАЯ ОШИБКА [semantic error]. Смысловая ошибка в программе, не связанная с нарушением *синтаксиса языка программирования.* К таким ошибкам относятся неправильное описание алгоритма решения задачи, неверное определение типа или значения переменных, несогласованность *исходных данных* с алгоритмом, неверное использование правильных *синтаксических конструкций языка.* Наличие С. о. устанавливается в процессе *испытания программы*, а их устранение производится при *отладке программы.* Ср. *логическая ошибка*

СЕМАФОР [semaphore]. Тип переменной, используемой в *языках программирования* для синхронизации взаимодействующих процессов

СЕНСОРНЫЙ ЭКРАН [touch screen]. *Экран дисплея*, способный определять координаты точки прикосновения к его поверхности. Пользователь карандашом или указкой, касаясь поверхности экрана, выбирает *элементы управле-*

ния графического интерфейса, двигает курсор, пишет текст или рисует. С. э. часто применяется в *портативных компьютерах*, заменяя мышь и другие *указательные устройства*. Ср. *графический планшет*, *световое перо*

СЕРВЕР [server]. 1. *Программа*, предоставляющая определенные услуги другим программам, которые называются *клиентами*. Программа-С. и программы-клиенты могут выполняться на одном или на разных компьютерах. Например, инсталлированная на компьютере программа *электронной почты* является программой-клиентом. Она контактирует с *почтовым ящиком* через программу-сервер, установленную на компьютере, где размещен *почтовый ящик*. См. *автоматизация, архитектура "клиент-сервер", связывание и внедрение объектов*. 2. Компьютер, на котором выполняется программа-сервер

СЕРВЕР ИМЕН ДОМЕНОВ [domain name server, DNS name server]. То же, что *сервер DNS*

СЕРВЕР ПЕЧАТИ [print server]. 1. *Узел сети* (компьютер), который управляет одним или несколькими печатающими устройствами при их совместном использовании другими подключенными к сети компьютерами. 2. Специальное *аппаратное устройство*, которое подключает печатающее устройство к сети. 3. Серверная *программа*, управляющая *сетевыми принтерами*

СЕРВЕР ПОЛНОМОЧИЙ [proxy server]. *Программа*, управляющая связью защищенной сети с остальной частью *Интернета*. Многие корпорации пользуются С. п., создающими защитный слой между корпоративной сетью и потенциально опасными приложениями. См. *брандмауэр*

СЕРВЕР ПРОТОКОЛА ПЕРЕДАЧИ ГИПЕРТЕКСТА [HyperText Transfer Protocol Daemon (HTTPD)]. *Веб-сервер*, первоначально разработанный в CERN (Conseil Europeen pour la Recherche Nucleaire, Европейский центр ядерных исследований) и названный CERN HTTPD. Позже С. п. п. г. был независимо разработан в NASA для *операционной системы UNIX*

СЕРВЕР СЦЕНАРИЕВ WINDOWS [Windows scripting host (WSH)]. Средство *операционных систем Windows*, позволяющее выполнять *сценарии*, написанные на *языках VBScript и JavaScript*; при этом поддерживаются и традиционные сценарии *командной строки* MS-DOS, и исполняемые файлы

СЕРВЕР DNS, сервер имен доменов [domain name server, DNS name server]. *Сервер*, преобразующий *имена доменов* в *IP-адреса*. Эти серверы имеют в своем распоряжении часть *базы данных доменной системы имен*, устанавливающей соответствие между именами компьютеров и их числовыми IP-адресами. Кроме того, С. DNS предоставляют имена клиентам системы, запрашивающим разрешение на использование имени в *сети Интернет*

СЕРВИС, служба [service]. Обслуживающая программа операционной системы, резидентно присутствующая в памяти и обеспечивающая выполнение определенных функций по запросу приложений или пользователя. На

рис. С.1 представлено диалоговое окно управления С. операционной системы Windows NT

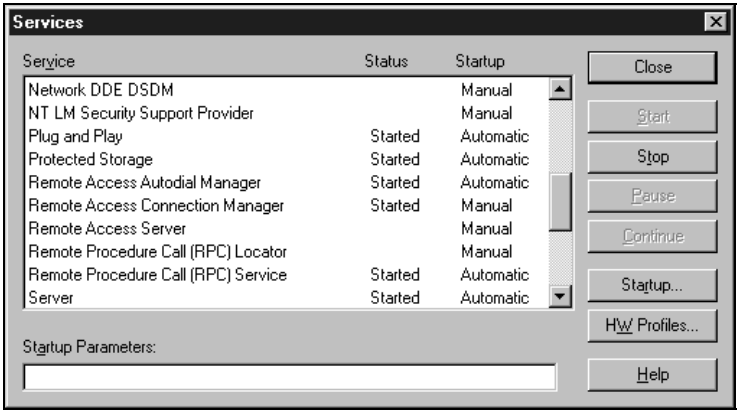

Рис. С.1. Окно управления сервисами операционной системы Windows NT

СЕРВИСНАЯ ПРОГРАММА [service program]. То же, что *обслуживающая программа*

СЕТЕВАЯ АРХИТЕКТУРА [network architecture]. То же, что *архитектура сети*

СЕТЕВАЯ БАЗА ДАННЫХ [network database]. *База данных*, организованная по сетевой *модели данных*. С. б. д. состоит из наборов записей, которые связаны между собой так, что некоторые записи могут содержать явные ссылки на другие наборы записей. Тем самым наборы записей образуют сеть (в математическом смысле), откуда и происходит название С. б. д. С. б. д. отличается от *иерархической базы данных* тем, что связи между записями могут быть совершенно произвольными, и отличается от *реляционной базы данных* тем, что связи явно присутствуют и хранятся в базе данных. Ср. *распределенная база данных*

СЕТЕВАЯ КАРТА, сетевая плата, сетевой адаптер [network card, network board]. Выполненный в виде *платы адаптер*, согласующий работу компьютера и *канала связи* при *передаче данных* от одного компьютера к другому в *компьютерной сети*. С. к. преобразует параллельно поступающие из шины компьютера сигналы, соответствующие *двоичным разрядам информационных слов*, в высокочастотные импульсы, передаваемые последовательно по *линии связи*. См. *шина, локальная вычислительная сеть*

СЕТЕВАЯ ПЛАТА [network board]. То же, что *сетевая карта*

СЕТЕВАЯ СУБД [network DBMS]. То же, что *система управления сетевыми базами данных*

СЕТЕВОЕ ПО [netware]. То же, что *сетевое программное обеспечение*

СЕТЕВОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, сетевое ПО [netware]. *Программное обеспечение*, организующее работу компьютеров в *компьютерной сети*

СЕТЕВОЙ АДАПТЕР [network card]. То же, что *сетевая карта*

СЕТЕВОЙ ПРИНТЕР [network printer]. См. *общий принтер*

СЕТЕВОЙ ПРОТОКОЛ [network protocol]. *Протокол*, определяющий правила взаимодействия между собой отдельных компонентов *компьютерной сети*. Например, С. п. SNMP регламентирует обмен пакетами данных между двумя *рабочими станциями* одной сети, а набор С. п. TCP/IP определяет правила взаимодействия между собой как объединенных в Интернете региональных сетей, так и подключенных к сети отдельных компьютеров. Ср. *межсетевой протокол*

СЕТЕВОЙ СТАНДАРТ NDIS [NDIS]. То же, что *спецификация интерфейсов сетевых устройств*

СЕТЕВОЙ УРОВЕНЬ [network layer]. Третий из семи уровней *модели OSI/ISO*, стандартизирующий связи компьютер—компьютер. Информация не всегда передается по прямому маршруту от отправителя к получателю: по пути она может маршрутизироваться от одного узла к другому, а также разбиваться на пакеты. Назначение сетевого уровня — устанавливать, поддерживать и хранить открытым маршрут для передачи информации

СЕТКА [grid]. Совокупность вертикальных и горизонтальных координатных линий, на фоне которых размещаются какие-либо *графические объекты*. Использование С. упрощает выравнивание графических объектов за счет так называемой "привязки к сетке", при которой границы объектов должны точно попадать на линии С. С. может быть отображаемой и скрытой, равномерной и неравномерной. Примером С., не связанной с компьютерами, является линованная бумага — "миллиметровка". С. часто применяется в таких приложениях, как *системы автоматизированного проектирования*, *графические редакторы*, *текстовые процессоры*, *электронные таблицы*. На рис. С.2 приведен пример сетки, используемой в графическом редакторе Visio

СЕТЬ ДРЕВОВИДНОЙ ТОПОЛОГИИ [tree network]. То же, что *древовидная сеть*

СЕТЬ ЗВЕЗДООБРАЗНОЙ ТОПОЛОГИИ [star network]. То же, что *радиальная сеть*

СЕТЬ КОЛЬЦЕВОЙ ТОПОЛОГИИ [ring network]. То же, что *кольцевая сеть*

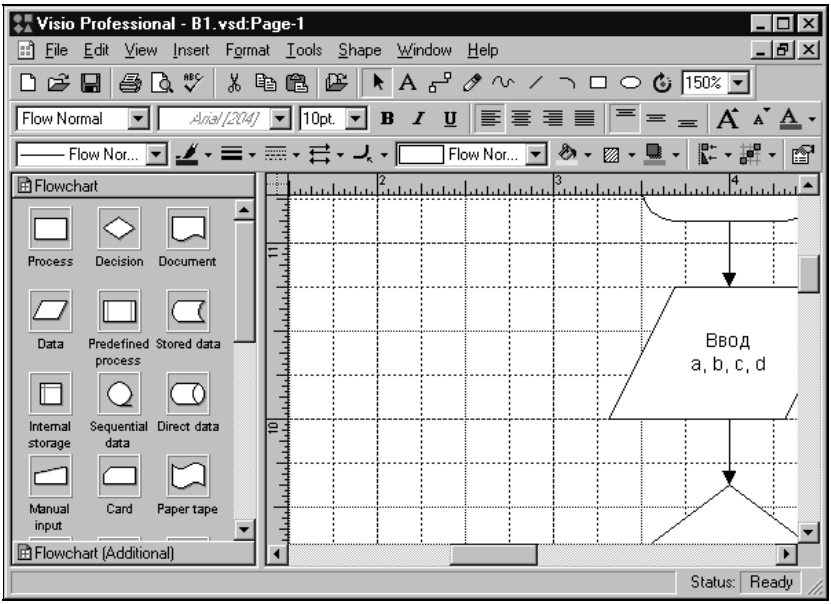

Рис. С.2. Сетка на рабочем поле приложения Visio

СЕТЬ ПЕРЕДАЧИ ДАННЫХ [data transmission network]. Совокупность *линий связи* и устройств коммутации, обеспечивающая *передачу цифровых данных* любой природы между электронными устройствами. Примером локальной С. п. д. является *сеть Ethernet*, а глобальной — Интернет. Обычно данные передаются между *узлами сети* в виде пакетов. С. п. д. могут быть основаны на *коммутации пакетов* и *коммутации соединений*

СЕТЬ С КОММУТАЦИЕЙ ПАКЕТОВ, сеть с пакетной коммутацией [packet switching network]. *Компьютерная сеть*, в которой передаваемые данные разбиваются на несколько специально оформленных *пакетов*, каждый из которых передается по сети независимо от других

СЕТЬ С МАРКЕРНЫМ ДОСТУПОМ, сеть с передачей маркера, маркерная сеть [token network]. *Локальная вычислительная сеть*, в которой управление *передачей данных* осуществляется с помощью определенной последовательности символов, называемой маркером. Маркер, как эстафета, передается по сети от одной *рабочей станции* к другой. Станция, получившая маркер, получает право на передачу данных. С. с м. д. регламентируются *стандартами IEEE*

СЕТЬ С ПАКЕТНОЙ КОММУТАЦИЕЙ [packet switching network]. То же, что *сеть с коммутацией пакетов*

СЕТЬ С ПЕРЕДАЧЕЙ МАРКЕРА [token network]. То же, что *сеть с маркерным доступом*

СЕТЬ СТАНДАРТА H.323, сеть H.323 [H.323 network, H.323]. *Телефонная сеть,* соотвествующая *стандартам H.323*, организующая телефонные соединения на основе *сетей передачи данных.* С. H.323 работают поверх любых сетей передачи данных, например, *сетей Ethernet.* Узлами С. H.323 являются *IP-телефоны*, компьютеры с запущенной программой *интернет-телефонии*, *привратники*, *шлюзы* и *устройства управления конференциями.* Функции *телефонной станции* распределены между разными узлами, причем каждый узел может быть многократно продублирован. Наиболее распространенным транспортным протоколом С. H.323 является *протокол TCP/IP*

СЕТЬ ШИННОЙ ТОПОЛОГИИ [bus network]. То же, что *шинная сеть*

СЕТЬ ЭВМ [computer network]. То же, что *компьютерная сеть*

СЕТЬ APPLETALK [AppleTalk]. Недорогая *локальная компьютерная сеть*, которая была разработана компанией Apple Computer и может использоваться как компьютерами Apple, так и другими компьютерами для связи и совместного использования ресурсов, подобных принтерам и *файл-серверам.* Компьютеры Macintosh включаются в сеть через порт принтера; компьютеры других фирм, например, IBM PC, должны быть оборудованы устройствами AppleTalk и соответствующими программами. В С. AppleTalk используется немодулированная *передача данных* со скоростью 230 Кбит/сек; в ней можно связать от 1 до 32 устройств (узлов) на расстоянии около 350 м. В сети использован иерархический набор протоколов, аналогичный *модели ISO/OSI*, а информация передается *пакетами данных* — кадрами. С. AppleTalk поддерживает соединение с другими С. AppleTalk с помощью мостов, а также с сетями иных типов через шлюзы

СЕТЬ ETHERNET [Ethernet]. *Широковещательная компьютерная сеть*, имеющая *архитектуру Ethernet*

СЕТЬ EXTRANET [extranet]. То же, что *экстранет*

СЕТЬ H.323 [H.323 network, H.323]. То же, что *сеть стандарта H.323*

СЕТЬ INTERNET [Internet]. То же, что *Интернет*

СЕТЬ INTRANET [intranet]. То же, что *интранет*

СЕТЬ ISDN [integrated services digital network (ISDN)]. То же, что *интегральная цифровая сеть связи*

СЕТЬ NETWARE [NetWare]. *Локальная вычислительная сеть*, работающая под управлением *операционной системы NetWare*

СЖАТИЕ ДАННЫХ, уплотнение данных [data compression, data compaction]. Преобразование *данных* в более компактную форму без потери содержащей-

ся в них информации. С. д. позволяет экономить память и повышать эффективность *передачи данных*. В теории информации существует много методов С. д., позволяющих путем изменения кода сократить их избыточность. На основе этих методов разработаны специальные программы — архиваторы, выполняющие *архивирование* файлов. Кроме того, методы С. д. позволяют существенно увеличивать емкость *внешней памяти*. Например, в *операционные системы MS-DOS 6.0* и *Windows 95* включена программа, увеличивающая как минимум вдвое емкость *жесткого* и *гибкого магнитных дисков* с помощью алгоритма MRCI (Microsoft Real-time Compression Interface). Эта программа занимает часть дискового пространства под специальный файл, где хранится сжатая информация. Для пользователя этот файл отображается в виде *логического диска*, но при обращении к нему выполняется архивирование/разархивирование в режиме реального времени. Очевидно, на такой диск не имеет смысла записывать *архивные файлы*. Существуют специальные методы и программы С. д., позволяющие компактно записывать, а затем воспроизводить графическую, звуковую и видеоинформацию. См. *сжатие файла, формат JPEG*

СЖАТИЕ ФАЙЛА [file compression, file compaction]. Преобразование *файла* в более компактную форму без потери содержащейся в нем информации с целью экономии памяти. С. ф. применяется при *архивировании файлов*. См. *сжатие данных*

СИ [C]. *Язык программирования высокого уровня*, предназначенный для разработки *системных программ* различного назначения. На С. пишутся *операционные системы*, компиляторы, редакторы, *системы управления базами данных*, программы мультимедиа, программы решения научных и инженерных задач и программы обработки текстов. С. был разработан в 1972 г. Д. Ричи в фирме Bell Laboratories и получил свое название как преемник ранее разработанного в этой фирме малоизвестного языка Би (B). С. обладает широким спектром стандартных *типов данных* и удобным механизмом композиции стандартных типов в более сложные. Язык дает программисту широкую свободу в выборе синтаксических конструкций, в частности, допускает произвольное использование *указателей*, что затрудняет работу неопытных программистов и, наоборот, облегчает опытным написание *системных программ*. В языке С. имеется около 40 *арифметических*, *логических*, *символьных* и других *операций* над данными (например, сдвиг кода, поразрядные операции, арифметические операции над указателями и т. д.). Управляющие конструкции языка допускают *структурное* и *модульное программирование* с раздельной трансляцией модулей. В С. имеются средства работы с системными и пользовательскими *библиотеками программ*. Дальнейшим развитием языка С. стал язык программирования общего назначения Си++

СИ++, **Си плюс плюс [C++]**. *Язык программирования высокого уровня*, происходящий от *языка программирования Си*. Обеспечивая основные *типы дан-*

ных, операции, синтаксис операторов и структуру программ языка Си, язык Си++ к своему "родительскому" языку добавил новый тип данных — класс. При помощи классов программист может определять сложные типы, которые включают не только элементы данных, но и функции, применимые к этому типу. И главное, классы создают механизм для абстракции данных и приводят к объектно-ориентированному программированию, при котором программа представляется как набор объектов — экземпляров некоторого класса. Классы могут определяться на основе других классов, наследуя свойства и методы базовых классов. Такой подход обеспечивает высокую модульность и структурность программ. Кроме того, в Си++ имеются функции и операторы, позволяющие встроить в задачу приспособленную к ней схему управления памятью компьютера. Все *описания функций* в Си++ должны содержать информацию о типах аргументов. Это позволяет не только проверять типы аргументов для предотвращения ошибок, но и переопределять функции, сохраняя стандартные имена функций и знаки операций. Дополнительные (по сравнению с Си) черты языка Си++ расширяют Си-подобные части языка, одновременно способствуя использованию новых приемов и стилей программирования и делая Си++ значительно богаче по своим возможностям большинства языков высокого уровня. Из числа систем программирования на языке Си++ (и Си) наибольшей популярностью у программистов пользуются *среда программирования* Visual C++ для Windows, разработанная фирмой Microsoft, и среда программирования Watcom C/C++ фирмы Watcom International Corp.

СИГНАЛ [signal]. 1. Носитель информации в виде изменяющейся физической величины, используемый для передачи данных. Например, в *вычислительных машинах* применяются электрические С. в виде непрерывно или дискретно изменяющегося электрического напряжения или электрического тока. См. *аналоговый сигнал*, *цифровой сигнал*. 2. То же, что *программное прерывание*

СИГНАЛ ПОДТВЕРЖДЕНИЯ ПРИЕМА [acknowledgement (ACK)]. См. *подтверждение приема*

СИГНАЛ АСК [acknowledgement (ACK)]. См. *подтверждение приема*

СИГНАТУРА [signature]. 1. То же, что *электронная подпись*. 2. Имя и *формальные параметры* операции или процедуры. С. может включать в себя необязательные типы возвращаемых значений

СИМВОЛ. 1. **[character]**. Элемент допустимого *вычислительной системой* набора знаков, который может быть закодирован, введен в компьютер и отображен на экране его дисплея. С. являются буквы, цифры, скобки, *знаки* препинания и *арифметических операций*, а также *специальные*, *управляющие* и *псевдографические символы*. Все допустимые вычислительной системой С. могут быть введены в память компьютера с помощью клавиатуры

нажатием либо одной клавиши, либо группы клавиш. Большинство вычислительных систем для представления С. используют с отдельными модификациями так называемые *коды ASCII*, *в которых* закодированы 256 С. числами от 0 до 255. В России для этого применяется *модифицированная десятичная кодировка ГОСТа*. Для составления и распечатки текстов служат специальные программы — *текстовые редакторы* и *текстовые процессоры*, многие из которых позволяют вводить и отображать на экране и принтере С., отсутствующие в стандартном наборе, например, буквы греческого алфавита. См. *стандарт UNICODE*. 2. **[symbol]**. Последовательность из одного или нескольких стоящих друг за другом условных знаков, используемая для обозначения чего-либо

СИМВОЛЬНАЯ КОНСТАНТА, текстовая константа [character constant]. *Константа*, значением которой является *символ* или строка символов. В программах заключается в апострофы или кавычки. Например, `'Y'`, `'123'`, `"Ответьте ДА или НЕТ"`

СИМВОЛЬНАЯ ПЕРЕМЕННАЯ [character variable]. *Переменная*, значением которой является *символ* или строка символов

СИМВОЛЬНЫЙ ТИП, литерный тип, тип Char [character type]. *Перечисляемый тип* данных, определенный заданием *символьных констант* и идентификаторов. В большинстве *языков программирования* значения С. т. изображаются одиночным символом, заключенным в апострофы, например, `'S'`, `'5'`, `'*'`. Для обозначения самого апострофа его либо повторяют дважды (в Паскале), либо перед ним ставят обратную косую черту, например, `''''` или `''\''`. Обычно множество значений С. т. составляют символы, предназначенные для связи *вычислительной системы* с пользователем. Все эти символы есть на *устройствах ввода* (например, на клавиатуре) и *устройствах вывода* (например, на принтере). Как сами символы, так и порядок их следования среди значений перечисляемого типа зависят от конкретной вычислительной системы. Обычно значения С. т. представляются в виде целых чисел без знака с помощью *кода ASCII* (либо иной кодировки) и считаются упорядоченными в соответствии с возрастанием значений этих чисел. Каждое из значений С. т. размещается в одном байте (или в двух байтах, если используется *стандарт UNICODE*) памяти. Для *описания переменных* С. т. обычно применяется *ключевое слово* `char`. Ниже приведены примеры описания переменных С. т. на языках Паскаль и Си:

```
var Letter, Signal: char;
char Letter, Signal;
```

СИММЕТРИЧНОЕ ШИФРОВАНИЕ [secret key enciphering, secret key encryption, symmetric enciphering, symmetric encryption]. То же, что *шифрование с закрытым ключом*

СИММЕТРИЧНЫЙ МНОГОПРОЦЕССОРНЫЙ КОМПЬЮТЕР, SMP-компьютер [symmetric multi processor computer, SMP-computer]. *Многопроцессорный компьютер* с *разделяемой памятью*, имеющий одну *операционную систему* и одну подсистему ввода/вывода. Слово "симметричный" означает, что каждый процессор работает как процессор общего назначения и, имея доступ к общему набору модулей памяти, может делать то же, что и любой другой. Операционная система должна обеспечивать равномерную загрузку процессоров, динамически перемещая процессы на освободившиеся процессоры

"СИНИЙ ЭКРАН" [blue screen]. Ситуация аварийной остановки операционной системы, при которой ее работа полностью прекращается и информация о причинах остановки выводится на экран с синим фоном. См. *аварийный останов*

СИНТАКСИС [syntax]. Правила записи предложений *формального языка* в виде текста. Ср. *нотация*

СИНТАКСИС ЯЗЫКА ПРОГРАММИРОВАНИЯ [programming language syntax]. Совокупность правил написания чисел, переменных, выражений, операторов, процедур и других элементов и предложений *(синтаксических конструкций)* данного *языка программирования*. Ср. *графическая нотация*

СИНТАКСИЧЕСКАЯ КОНСТРУКЦИЯ [syntax construction]. Допустимое *синтаксисом языка программирования* сочетание символов или символов и *ключевых слов*. С. к. являются как законченные предложения языка (операторы, описания и т. д.), так и их составные части. Например, С. к. в Паскале являются: `y:= y+x, 1 .. 20, for k: = 1 to p do`

СИНТАКСИЧЕСКАЯ ОШИБКА [syntax error]. *Ошибка в программе*, связанная с нарушением *синтаксиса языка программирования*. К таким ошибкам относятся нарушения грамматических правил: неправильное написание констант, переменных, выражений, операторов, процедур и других синтаксических конструкций. Все без исключения С. о. обнаруживаются транслятором в процессе компиляции при *отладке программы*

СИНТЕЗАТОР РЕЧИ [voice synthesizer]. Устройство или программа, осуществляющие *синтез речи* человека на основе *цифрового кода*

СИНТЕЗ РЕЧИ [speech synthesis, speech generation]. Процесс создания звуков, имитирующих человеческий голос. Устройство, именуемое *синтезатором речи*, по сигналам, поступающим из систем *обработки данных* или систем управления, формирует речевые сообщения, воспроизводимые динамиком компьютера. С. р. производится путем "сборки" сообщений из элементов набора основных звуков, которые либо могут иметь искусственное происхождение, либо представлять собой записанные на носитель фрагменты естественной человеческой речи. Ср. *распознавание речи*

СИНХРОНИЗАЦИЯ БАЗ ДАННЫХ [database synchronization]. См. *репликация базы данных*

СИНХРОННАЯ ПЕРЕДАЧА [synchronous transmission]. То же, что *синхронная связь*

СИНХРОННАЯ СВЯЗЬ, синхронная передача [synchronous connection, synchronous transmission]. Режим *передачи данных*, при котором временной интервал передачи символа совпадает с временным интервалом его приема. Это требует синхронизации работы передающего и приемного устройств перед каждой передачей пакета данных. Ср. *асинхронная связь*

СИСТЕМА АВТОМАТИЗАЦИИ АВТОРСКОЙ РАБОТЫ, авторская система [authoring system]. *Программное обеспечение*, позволяющее автоматизировать процесс создания интерактивных продуктов мультимедиа, включающих анимацию, звук, видео и другие изобразительные средства. С. а. а. р. применяются для создания звуковых видеоклипов, а также при разработке диалоговых средств обучения и справочных систем

СИСТЕМА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ (САПР) [computer-aided design (CAD)]. Комплекс *программных* и *аппаратных средств*, предназначенных для автоматизации процесса проектирования человеком технических изделий или продуктов интеллектуальной деятельности. САПР широко применяются в архитектуре, электротехнике, электронике, машиностроении, авиакосмической технике, математике и программировании. Предоставляя человеку решение творческих задач, САПР берет на себя выполнение на компьютере поддающихся алгоритмизации сложных вспомогательных операций, например, проведение проектных расчетов, выполнение и корректировку чертежей, оптимизацию формы изделий или оптимизацию раскроя материала и т. п. Пополняемая *база данных* САПР снабжает человека необходимой информацией об изобретениях, существующих стандартах и проектных решениях, типовых элементах конструкций и комплектующих изделиях, материалах, технологиях исполнения, а также о различных законодательных положениях, руководствах, инструкциях и т. п. Развитые САПР способны по задаваемым проектным требованиям автоматически выдавать решения, достаточные для рассмотрения и окончательной корректировки специалистом. *Аппаратное обеспечение* САПР составляет компьютер с набором устройств, необходимых для ввода и вывода *графической информации* (графопостроитель, *световое перо*, *графический планшет* и др.), образующие *автоматизированное рабочее место* (АРМ проектировщика). Для крупных САПР основой служит *локальная вычислительная сеть*, в которой *главная ЭВМ* обслуживает несколько АРМов, в этом случае САПР является *системой коллективного пользования*

СИСТЕМА АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ [automatic control system]. Система, управляющая объектом без участия человека. Ср. *автоматизированная система управления*

СИСТЕМА БЕЗОПАСНОСТИ [security system]. Совокупность организационных мер, технических и программных средств, направленных на реализацию *политики безопасности*

СИСТЕМА КОЛЛЕКТИВНОГО ДОСТУПА [multi-access system]. То же, что *система коллективного пользования*

СИСТЕМА КОЛЛЕКТИВНОГО ПОЛЬЗОВАНИЯ, система коллективного доступа [multi-access system]. *Вычислительная система*, обеспечивающая одновременную работу нескольких *пользователей*. Каждый пользователь имеет терминал, связанный с *главной ЭВМ* через *линию связи*, по которой осуществляется *передача данных*. С. к. п. работает в режиме *разделения времени*, при котором *время центрального процессора* распределяется между пользователями периодически, но столь короткими квантами, что у каждого пользователя создается иллюзия собственной вычислительной системы, находящейся в его полном распоряжении. Сообщив через терминал свое имя и пароль, пользователь может вести постоянный диалог с машиной, вводить команды, программы, запросы, сообщения и получать ответы в приемлемое для него время. В связи с большим количеством пользователей в С. к. п. важное значение имеет решение вопросов *защиты памяти* и *защиты данных*

СИСТЕМА КОМАНД [instruction set]. Полный набор *команд*, допустимых в *машинном языке* данной ЭВМ. Представляет собой список всех команд, которые способен выполнять конкретный компьютер, с указанием *кодов операций* и разрешенных способов адресации

СИСТЕМА ПРОГРАММИРОВАНИЯ (СП) [programming system]. Совокупность *языка программирования* и *системы разработки программ*, обеспечивающих автоматизированную *разработку* и *выполнение программ* на данном языке и подготовку соответствующей документации. Как правило, С. п. включает не эталонный вариант языка, а его версию, содержащую определенные упрощения или расширения. Некоторые С. п. могут поддерживать разработку программ на нескольких языках. Наиболее известные С. п. для *персональных компьютеров*: Visual Studio, созданная фирмой Microsoft, поддерживающая языки программирования Basic, Java, C++; Delphi фирмы Inprise (Borland International Inc.) и др.

СИСТЕМА ПРОГРАММИРОВАНИЯ MPI [MPI programming system]. *Система программирования*, обеспечивающая разработку и выполнение *MPI-программ*. В настоящее время получили распространение С. п. MPI, сопряженные с языками Фортран, Си и Си++. См. *интерфейс передачи сообщений*

СИСТЕМА ПРОГРАММИРОВАНИЯ PVM [PVM programming system]. *Система программирования*, обеспечивающая разработку и выполнение *PVM-программ*. В настоящее время получили распространение С. п. PVM, сопряженные с языками Фортран, Си и Си++. См. *параллельная виртуальная машина*

СИСТЕМА РАЗДЕЛЕНИЯ ВРЕМЕНИ [time-sharing operating system (TSS)]. *Многозадачная операционная система*, обеспечивающая работу компьютера в режиме *разделения времени*. На основе С. р. в. работают различные *системы коллективного пользования*

СИСТЕМА РАЗРАБОТКИ ПРОГРАММ [program development system]. Совокупность *программных средств*, обеспечивающих *разработку программ*. Обычно С. р. п. состоит из *текстового редактора*, *транслятора*, *компоновщика*, *библиотеки стандартных подпрограмм*, средств *трассировки* и *отладки программ*, системы подготовки документации, а также других вспомогательных программ. С. р. п. может обладать оболочкой, обеспечивающей удобное и единообразное взаимодействие пользователя с программами. Если при этом С. р. п. включает все необходимые пользователю программные средства, позволяющие программисту выполнять свою работу, не выходя в *операционную систему*, то такая С. р. п. представляет собой *интегрированную систему* и называется *средой программирования*. Ср. *интегрированная среда разработки*

СИСТЕМА СЧИСЛЕНИЯ [number system]. Совокупность символов и правил написания чисел. Наибольшее распространение получила *десятичная система счисления*. В *вычислительной технике* применяются также *двоичная*, *восьмеричная и шестнадцатеричная системы счисления*. Перечисленные С. с. являются *позиционными системами счисления*

СИСТЕМА УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ (СУБД) [data base management system (DBMS)]. *Комплекс программ* и языковых средств, предназначенных для создания, ведения и использования *баз данных*. СУБД дает программистам возможность разрабатывать новые, более совершенные средства *обработки данных*, а пользователям — возможность непосредственно управлять хранящимися в базе данными, для чего обрабатывает обращения к базе данных *прикладных программ* и пользователей и при этом обеспечивает целостность, восстановление, проверку правильности и секретность данных. Таким образом, СУБД имеет свойства, характерные как для компиляторов, так и для *операционных систем*. В наиболее полном варианте СУБД может состоять из следующих компонентов: 1) *интерфейса*, реализующего непосредственное управление данными с клавиатуры; 2) *языка программирования* прикладных *информационных систем*, реализованного как интерпретатор, что позволяет быстро создавать и отлаживать программы обработки данных; 3) компилятора для придания готовой программе формы *исполняемого файла*; 4) *ядра СУБД*, осуществляющего операции манипулирования данными; 5) *обслуживающих программ* (утилит) быстрого программирования рутинных

операций (генераторов отчетов, экранов, меню и других приложений). В реальных СУБД обычно присутствуют не все указанные компоненты. Одной из важнейших характеристик любой СУБД является используемый в ней тип транслятора (компилятор или интерпретатор). Программы, написанные для интерпретатора, исполняются лишь в присутствии самой системы, а система-компилятор может создавать автономные *программные изделия*. Среди наиболее известных СУБД для *персональных компьютеров* — система FoxPro фирмы Fox Software, которая является интерпретирующей с высокими скоростными характеристиками, однако в ней имеется компилятор, позволяющий при желании сформировать EXE-файлы готовых программ. Система Clipper фирмы Nantucket — только компилирующая, формирующая модули типа EXE, которые могут запускаться самостоятельно без "родительской" СУБД. Интерпретирующая система Access фирмы Microsoft обладает развитым графическим интерфейсом, облегчающим пользователю управление данными. Другой важнейшей характеристикой СУБД является *модель данных*, положенная в основу. В настоящее время наибольшее распространение получили *реляционные СУБД*, например, все упомянутые выше СУБД являются реляционными. См. *система управления иерархическими базами данных, система управления реляционными базами данных, система управления сетевыми базами данных*

СИСТЕМА УПРАВЛЕНИЯ ИЕРАРХИЧЕСКИМИ БАЗАМИ ДАННЫХ, иерархическая СУБД [hierarchical database management system (HDBMS)]. Система управления *иерархическими базами данных*, т. е. базами данных, построенными согласно иерархической *модели данных*. Иерархическая база данных имеет более простую структуру, а потому С. у. и. б. д. допускает наиболее эффективную реализацию по сравнению с другими СУБД. Ср. *реляционная система управления базами данных, сетевая система управления базами данных*

СИСТЕМА УПРАВЛЕНИЯ ПАМЯТЬЮ [memory management system]. Совокупность *аппаратных* и *программных средств*, обеспечивающих *управление памятью*

СИСТЕМА УПРАВЛЕНИЯ РЕЛЯЦИОННЫМИ БАЗАМИ ДАННЫХ (СУРДБ), реляционная СУБД [relational data base management system (RDBMS)]. Система управления *реляционными базами данных*, т. е. базами данных, построенными согласно реляционной *модели данных*. В настоящее время С. у. р. б. д. являются наиболее распространенными и популярными среди СУБД. Ср. *иерархическая система управления базами данных, сетевая система управления базами данных*

СИСТЕМА УПРАВЛЕНИЯ СЕТЕВЫМИ БАЗАМИ ДАННЫХ, сетевая СУБД [network database management system]. Система управления *сетевыми базами данных*, т. е. базами данных, построенными согласно сетевой *модели данных*. В настоящее время С. у. с. б. д., как правило, строятся в соответствии со

стандартами *КОДАСИЛ*. Ср. *иерархическая система управления базами данных, сетевая система управления базами данных*

СИСТЕМА УПРАВЛЕНИЯ ЦВЕТОМ [color management system (CMS)]. Совокупность программных средств, обеспечивающих соответствие цветов, на устройствах, которые используются на всех этапах работы *настольной редакционно-издательской системы* от сканирования до типографской печати

СИСТЕМНАЯ БИБЛИОТЕКА [system library]. 1. *Библиотека*, хранящаяся в *системном каталоге*. 2. *Библиотека*, в которой хранятся *системные файлы*

СИСТЕМНАЯ ДИСКЕТА [system disk]. То же, что *системный диск*

СИСТЕМНАЯ ОБЛАСТЬ [system area]. *Область памяти* в начале *магнитного диска*, используемая *операционной системой* при работе с файлами, записанными на диск. Включает таблицу размещения файлов на диске и *корневой каталог*. С. о. *системного диска* содержит загрузчик операционной системы. См. *загрузка операционной системы*

СИСТЕМНАЯ ПАПКА [system folder]. То же, что *системный каталог*

СИСТЕМНАЯ ПРОГРАММА [system program]. 1. *Программа*, входящая в состав *операционной системы*. Например, драйвер клавиатуры. 2. Программа, используемая для разработки и выполнения *прикладных программ*. Например, транслятор с *языка программирования* на *машинный язык*. Понятия "системный" и "прикладной" относительны: транслятор является прикладной программой по отношению к операционной системе и системной — по отношению к транслируемой программе

СИСТЕМНАЯ СЕТЕВАЯ АРХИТЕКТУРА, архитектура SNA [Systems Network Architecture (SNA)]. Широко используемая спецификация связи, разработанная фирмой IBM для определения функций сети и установления стандартов различных моделей *вычислительных систем*, предназначенных для обмена и *обработки данных*. С. с. а. делит сетевую связь на пять уровней, каждый из которых представляет отдельный функциональный уровень, начинающийся от физических связей до прикладного *программного обеспечения*. С. с. а. не распространяется на *аппаратные средства*, сети самого высокого уровня и *прикладные программы*. Ниже в порядке убывания приведены пять уровней С. с. а.: 5) Функциональное управление (отображение данных, интерфейс между сетью и пользователем); 4) Управление потоком данных (обработка потока данных во время сеанса связи); 3) Контроль передачи (оценка статуса и темпа сеанса связи); 2) Управление маршрутом (маршрутизация данных); 1) Управление каналом связи (адресует текущую передачу данных). Ср. *модель ISO/OSI*

СИСТЕМНАЯ ШИНА [bus]. *Шина персональных компьютеров*. Распространены 16-разрядная шина ISA и 32-разрядная шина EISA. Очевидно, что вторая производительнее, однако даже этой производительности недоста-

точно для быстрой передачи данных в видеосистему и обмена данными с винчестером. Для решения этой проблемы разработаны *локальные шины*

СИСТЕМНОЕ МОДАЛЬНОЕ ДИАЛОГОВОЕ ОКНО [system modal dialog box]. То же, что *окно системных сообщений*

СИСТЕМНОЕ МОДАЛЬНОЕ ОКНО [system modal window]. То же, что *окно системных сообщений*

СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ [system software]. Вся совокупность *системных программ*, входящих в состав *вычислительной системы*. Это *программное обеспечение*, необходимое для эксплуатации и технического обслуживания компьютера, для организации вычислительных работ и автоматизации разработки *прикладных программ*. Важнейший компонент С. п. о. — *операционная система* — является необходимым дополнением аппаратных средств и обычно поставляется самим производителем вычислительной системы. Остальные компоненты С. п. о. устанавливаются на компьютер самим пользователем для создания удовлетворяющей его вычислительной системы. См. *программное обеспечение, системная программа*

СИСТЕМНОЕ СООБЩЕНИЕ [system message]. *Сообщение*, выдаваемое *операционной системой*. См. *окно системных сообщений*

СИСТЕМНЫЙ АДМИНИСТРАТОР [system administrator]. Человек или группа лиц, ответственные за состояние, развитие и использование *системного программного обеспечения* организации или учреждения. С. а. ведет *учетные записи пользователей*, определяет *политику учетных записей*, обеспечивает работоспособность *операционных систем* и необходимый уровень *защиты данных*. Он взаимодействует с *системными инженерами*, *системными программистами* и пользователями данной организации. Ср. *администратор базы данных*

СИСТЕМНЫЙ БЛОК [system block]. Заключенный в металлический или пластмассовый корпус *аппаратный блок*, содержащий основные устройства *персонального компьютера*. Существуют несколько типов компоновки С. б.: горизонтальная — типов desktop ("настольный"), slim line ("тонкая линия") и вертикальная — типов big tower ("большая башня"), midi tower ("средняя башня"), mini tower ("малая башня"). В С. б. находятся блок питания, *центральный процессор*, *постоянное запоминающее устройство* (ПЗУ), *оперативная память*, *кэш-память* (возможна конфигурация без кэш-памяти), *адаптер дисплея*, *жесткий магнитный диск*, накопители для *гибких магнитных дисков* и *лазерных дисков*, *системная шина*, *локальная шина* (возможна конфигурация без локальной шины), *параллельный порт* (один или несколько), *последовательный порт* (один или несколько), платы дополнительных устройств. В некоторых компьютерах центральный процессор, оперативная память, системная и локальная шины размещены на так называемой *материнской плате*, а адаптер дисплея, контроллеры жесткого и гибкого дисков и порты

выполнены в виде отдельных плат и вставляются в разъемы (слоты) на материнской плате. В других компьютерах адаптер, контроллеры и порты могут быть выполнены как части материнской платы. См. *конфигурация компьютера*

СИСТЕМНЫЙ ДИСК, системная дискета [system disk]. *Магнитный* или *оптический диск*, на котором размещены модули *операционной системы* и с которого производятся ее загрузка и запуск. См. *загрузка операционной системы*

СИСТЕМНЫЙ ИНЖЕНЕР [software engineer]. Лицо (обычно *инженер-программист*), занимающееся эксплуатацией и сопровождением *системного программного обеспечения*. В некоторых случаях С. и. выполняет также функции *системного администратора*. Ср. *системный программист*

СИСТЕМНЫЙ КАТАЛОГ, системная папка [system catalogue, system folder]. Каталог *системных файлов*

СИСТЕМНЫЙ ПРОГРАММИСТ [system programmer, software programmer, toolsmith]. *Программист*, занимающийся разработкой *системного программного обеспечения*. Ср. *системный инженер*

СИСТЕМНЫЙ ФАЙЛ [system file]. *Файл*, содержащий один из модулей *операционной системы* или набор данных, которые она использует или поддерживает

СКАНЕР [scanner]. 1. *Устройство ввода* в компьютер *графической* и *текстовой информации* с листа бумаги, пленки и т. п. С помощью С. в память компьютера можно ввести в виде растра двумерное изображение (печатный текст, рисунок, карту, фотографию и т. п.). С. бывают ручные и настольные. При работе с ручным С. его перемещают по вводимому документу вручную. В настольном С. ввод (сканирование) осуществляется автоматически. Соответственно, точность ввода в настольном С. выше. С. бывают цветные (способные вводить цветные изображения) и черно-белые. Для ввода текстовых документов помимо С. необходимо *программное обеспечение* для распознавания символов. 2. В *системах программирования* то же, что *лексический анализатор*

СКАЧИВАНИЕ [download]. То же, что *загрузка по линии связи*

СКОРОСТЬ ПЕРЕДАЧИ ДАННЫХ [data transfer rate, bit rate]. Среднее количество битов, символов или блоков, передаваемых в системе *передачи данных* за единицу времени. Для каналов, передающих информацию в *двоичном коде*, С. п. д. измеряется числом бит в секунду (бит/с). См. *бод*

СКРИПТ [script]. То же, что *сценарий*

СКРОЛЛИНГ, прокрутка изображения, прокрутка [scrolling]. Вертикальное или горизонтальное перемещение изображения в окне экрана. Управление С. осуществляется *клавишами управления курсором* или при помощи мыши путем перемещения *бегунка* на *линейках прокрутки*. Очень удобной является

разработанная корпорацией Microsoft мышь с дополнительным колесиком, вращение которого позволяет осуществлять С.

СКРЫТЫЙ ТЕКСТ [hidden text]. Фрагмент *текста электронного документа*, оформленный таким образом, что он по умолчанию невидим на экране и не выводится на печать. Например, в виде С. т. в документ могут быть вставлены авторские комментарии

СКРЫТЫЙ ФАЙЛ [hidden file]. *Файл*, имя которого согласно *политике безопасности* не отражается в списке файлов каталога. Для этого он снабжается специальным признаком

СЛИЯНИЕ 1. **[mail merge]**. Процесс формирования серийных документов, в результате которого основной документ объединяется с данными из заданного источника. Каждая запись в источнике данных порождает один результирующий документ. Например, если в таблице *базы данных* собраны индивидуальные данные о некоторых людях, а основной документ представляет собой текст приглашения на конференцию, то с помощью С. можно автоматически получить нужное количество именных приглашений. 2. **[merge]**. Ситуация, при которой сливаются несколько альтернативных *потоков управления*. В *унифицированном языке моделирования* UML С. обозначается символом "решение" (см. *элемент модели*) и применяется на *диаграммах деятельности* для описания *ветвления алгоритма*. Противоп. *ветвление*. Ср. *соединение*

СЛОВО [word]. 1. Не содержащая пробелов последовательность символов в некотором алфавите, имеющая определенное смысловое значение. 2. То же, что *машинное слово*. См. *слово состояния*

СЛОВО СОСТОЯНИЯ [status word]. *Слово*, которое содержит определенным образом закодированную информацию о текущем состоянии устройства, программы или процесса. С. с. формируются и хранятся в специальных внутренних регистрах *центрального процессора* и позволяют в случае прерывания восстановить приостановленную последовательность действий компьютера. Например, С. с. программы содержит адрес выполняемой команды, адрес следующей команды, биты, указывающие на состояние *арифметико-логического устройства*, а также сведения о прерывании и, возможно, другую информацию, необходимую для восстановления процесса выполнения программы по окончании *обработки прерывания*. Для этого при наступлении прерывания программы или процесса С. с. автоматически запоминается в *оперативной памяти*, а затем в нужный момент снова считывается в соответствующий регистр. См. *прерывание*

СЛОТ [slot]. Разъем на *материнской плате*, в который устанавливаются вспомогательные платы для расширения возможностей компьютера (например, *звуковая плата*, модем и т. п.)

СЛУЖБА [service]. То же, что *сервис*

СЛУЖБА КАТАЛОГОВ [directory services]. *Программное обеспечение*, предоставляющее услугу по структурированному хранению и представлению учетных данных, например, С. к. пользователей сети. См. *служба каталогов Active Directory*, *интерфейс ADSI*

СЛУЖБА КАТАЛОГОВ ACTIVE DIRECTORY [Active Directory]. *Служба каталогов*, входящая в состав серверной версии *операционной системы Windows* фирмы Microsoft. См. *интерфейс ADSI*

СЛУЖЕБНАЯ ПРОГРАММА [service program, utility]. То же, что *обслуживающая программа*

СЛУЖЕБНОЕ СЛОВО [reserved word]. То же, что *ключевое слово*

СЛЭШ [slash]. Косая черта /, используемая в качестве символа или знака операции деления. Символ \ называют обратным слэшем

СМАРТ-КАРТА [smart card]. *Носитель данных* в виде пластиковой карточки стандартного размера, в которую вмонтирована микросхема. В области применения является аналогом *магнитной карты*, но обладает более широкими возможностями

СМЕЩЕНИЕ [displacement, offset]. Число, указывающее положение элемента данных в *памяти ЭВМ* относительно *базового адреса. Физический адрес* равен сумме базового адреса и С. См. *адресация, распределение памяти, сегмент памяти*

СОБЫТИЕ [event]. Одномоментное идентифицируемое изменение состояния некоторой системы. В *программировании*, *ориентированном на события*, предусматривается множество разнообразных С., для которых программист может определять процедуры реакции на С. См. *программное прерывание*

СОБЫТИЙНО-УПРАВЛЯЕМОЕ ПРОГРАММИРОВАНИЕ [event-driven programming]. То же, что *программирование*, *ориентированное на события*

СОВМЕСТИМОСТЬ [compatibility]. Свойство *программы* (или *аппаратного устройства*) правильно работать вместе с программой (или устройством) другого производителя или с другой версией программы (моделью устройства) того же производителя. Например, большинство моделей *персональных компьютеров* программно С., т. е. программа, работающая на одной модели такого компьютера, будет работать и на другой модели. Однако отношение С. не всегда является симметричным. Например, программы, разработанные для Windows 3.*x*, могут работать под управлением Windows 9.*x*, но программы, разработанные для Windows 9.*x* не могут выполняться под управлением Windows 3.*x*. С. является полезным и важным для потребителя свойством, поскольку позволяет потребителю выбирать различные модели и версии нужных ему продуктов различных производителей. Для обеспечения С. про-

изводители стараются строго следовать промышленным стандартам, изготавливают специальные *преобразователи*, *эмуляторы* и т. п. См. *совместимость сверху вниз, совместимость снизу вверх*

СОВМЕСТИМОСТЬ СВЕРХУ ВНИЗ [downward compatibility]. *Совместимость* ряда версий программы (моделей устройства), при которой младшие версии совместимы со старшими. Например, все версии программы MS Word, начиная с версии 6.0 С. с. в. Другими словами, документ, подготовленный в старшей версии Word, может быть сохранен таким образом, что его можно открыть и использовать младшей версией Word. Ср. *совместимость снизу вверх*

СОВМЕСТИМОСТЬ СНИЗУ ВВЕРХ [upward compatibility]. *Совместимость* ряда версий программы (моделей устройства), при которой старшие версии совместимы с младшими. Например, все модели процессора Pentium С. с. в. Другими словами, программа, разработанная для младшей версии процессора, будет работать и на старшей версии. Обеспечение С. с. в является общепринятой практикой в настоящее время. Ср. *совместимость сверху вниз*

СОВМЕСТНО ИСПОЛЬЗУЕМЫЙ РЕСУРС [shareable resource, shared resource]. То же, что *общий ресурс*

СОЕДИНЕНИЕ [connection]. 1. Связь, установленная между устройствами для *передачи данных*. См. *канал передачи данных*. 2. Связь, установленная между программами для *передачи данных*. Например, С. ODBC. См. *открытый доступ к базам данных*. 3. **[join]**. Ситуация, при которой соединяются несколько параллельных *потоков управления*, происходит их *синхронизация*, и образуется один поток управления. В *унифицированном языке моделирования* UML С. обозначается символом "соединение" (см. *элемент модели*) и применяется на *диаграммах деятельности* для описания *параллелизма*. Противоп. *развилка*. Ср. *слияние*

СОЗДАНИЕ КАТАЛОГА [directory creation]. Процедура, результатом которой является выделение памяти для нового *каталога* и присвоение ему имени. С. к. выполняется с помощью стандартных средств *операционной системы*. Например, в *оболочке Norton Commander* для этого следует нажать клавишу <F7>, в поле ввода появившегося окна набрать имя нового каталога и нажать клавишу <Enter>. В Проводнике Windows достаточно щелкнуть правой кнопкой мыши и выбрать соответствующую команду в *контекстном меню*

СОЗДАНИЕ ЭКЗЕМПЛЯРА [instantiation]. Создание нового *экземпляра класса*

СОКЕТ [socket]. 1. Гнездо — контактный разъем, в который вставляется *интегральная схема* или плата. 2. То же, что *сокет-канал*

СОКЕТ-КАНАЛ [socket]. Технология, используемая для связи компьютеров в *вычислительной сети*. Представляет собой аналог двустороннего *программного канала*, примененного для обмена данными между прикладными программами, исполняемыми на разных компьютерах. С.-к. описаны в *интерфейсе прикладного программирования* Berkeley Sockets, разработанном в университете штата Калифорния (США). См. *программное обеспечение Windows Sockets*

СООБЩЕНИЕ [message]. 1. *Информация* о ходе или состоянии *вычислительного процесса*, выдаваемая пользователю компонентами *вычислительной системы*. Например, С. об ошибке, выдаваемое *операционной системой*, *обрабатывающей* или *обслуживающей программой* и содержащее сведения о месте, типе и характере ошибки в программе пользователя. См. *системное сообщение*. 2. Порция данных, оформленная для *передачи данных* принятым образом в этой *компьютерной сети* или системе *компьютерной связи*. Например, С. *электронной почты*. С. имеет, как правило, заголовок и признак окончания С. Заголовок содержит информацию об отправителе и получателе С. (например, их имена и адреса), сведения о содержании и длине С. и времени его отправления. 3. Средство синхронизации и обмена информацией между *параллельными вычислительными процессами*. 4. Передача информации от одного объекта другому в расчете на то, что за этим последует некоторая деятельность. С. может быть сигнал или *вызов процедуры*. Получение С., как правило, трактуется как *событие*

СОПРОВОЖДЕНИЕ ПРОГРАММНОГО ИЗДЕЛИЯ, поддержка [program product maintenance, support]. Меры, направленные на поддержание в работоспособном состоянии находящейся в эксплуатации программы: устранение выявленных в ходе эксплуатации ошибок и внесение изменений в программу с целью улучшения ее функциональных возможностей в соответствии с изменением предъявляемых к ней требований. С. п. и. обычно ведется либо специально выделенным квалифицированным программистом из состава пользователей, либо фирмой — производителем или распространителем данного изделия

СОПРОЦЕССОР [coprocessor]. Специализированный *процессор*, дополняющий функциональные возможности *центрального процессора*. С. расширяет набор команд, которыми может пользоваться программист. Когда центральный процессор получает команду, которая не входит в его рабочий набор, он передает управление С., в рабочий набор которого эта команда включена. В *вычислительной системе* могут применяться два и более С. Например, один может быть *математическим сопроцессором*, а другой — обеспечивать управление *базой данных*

СОРТИРОВКА [sort]. То же, что *сортировка данных*

СОРТИРОВКА ДАННЫХ, сортировка, упорядочение [data sorting, sort]. Распределение *элементов данных* по группам в соответствии с определенными правилами. Разновидностью С. д. является размещение элементов данных на экране или в памяти в порядке возрастания или убывания значений некоторого признака. Например, в результате сортировки массива чисел по возрастанию их значений формируется новый массив чисел, первым элементом которого является наименьшее число, а последним — наибольшее. С. д. *литерного типа* приводит к их расположению в алфавитном порядке, т. к. значения кодов английских букв возрастают от первой буквы алфавита к последней, а значения кодов русских букв, отличающихся по изображению от английских, больше значений кодов английских букв (см. *модифицированная альтернативная кодировка ГОСТа*). С. д. — записей файла может быть проведена по убыванию или возрастанию значений одного или нескольких полей записи. Предварительная сортировка может существенно ускорить *поиск* и *обработку данных*. Средства С. д. предусматриваются как в *информационных системах*, так и в *операционных системах* и в *языках программирования*, ориентированных на задачи обработки данных (например, в языке Кобол). См. *сортировка файлов*

СОРТИРОВКА ФАЙЛОВ [files sorting]. Упорядочение файлов каталога по имени, расширению, размеру или по времени создания. С. ф. выполняется с помощью стандартных средств *операционной системы*. Например, в *оболочке Norton Commander* нажатие комбинации клавиш <Ctrl>+<F3> вызывает С. ф. по имени в алфавитном порядке; <Ctrl>+<F4> — по расширению в алфавитном порядке; <Ctrl>+<F5> — по времени создания; <Ctrl>+<F6> — по размеру. В *Проводнике Windows* достаточно щелкнуть правой кнопкой мыши и выбрать соответствующую команду из *контекстного меню*

СОСТАВНОЙ ДОКУМЕНТ [compound document]. *Документ*, который содержит связанные и/или внедренные объекты, подготовленные в других приложениях. Например, документ MS Word, который содержит диаграммы, подготовленные в MS Excel, и отчеты, сгенерированные в MS Access. См. *связывание и внедрение объектов*

СОСТАВНОЙ КЛЮЧ [compound key]. *Ключ* записи, состоящий из ее нескольких ключевых полей

СОСТАВНОЙ ОПЕРАТОР [compound statement]. Последовательность *операторов*, формально объединенных в одну *синтаксическую конструкцию* заключением в *операторные скобки*. Пример см. в статье *операторные скобки*

СОСТОЯНИЕ [state]. Ситуация в жизненном цикле объекта, во время которой он выполняет определенную деятельность или ожидает какого-либо события

СОХРАНЕНИЕ [save]. Запись группы данных из *оперативной памяти* в файл, находящийся во *внешней памяти*

СПАМ [spam]. *Сообщение*, которое принудительно посылается пользователям каких-либо услуг компьютерной сети. Например, С. рассылается подписчикам телеконференций с целью напомнить тематику дискуссий (как правило, это делается модератором телеконференции). Некоторые фирмы и пользователи злоупотребляют рассылкой по *электронной почте* рекламного С.

СПЕЦИАЛЬНЫЕ КЛАВИШИ [special keys]. Группа клавиш клавиатуры, предназначенных для управления работой компьютера или инициализации стандартных действий. На клавиатурах *персональных компьютеров*, как правило, имеются следующие С. к. (см. рис. К.3):

- <Esc> — клавиша отказа от выполняемого действия (от англ. *escape* — побег, спасение). В программах используется для отмены различных действий или выхода из программы;
- <Print Screen> — клавиша печати экрана на принтере. При работе компьютера под управлением *операционной системы MS-DOS* для печати экрана, работающего в *текстовом режиме*, нужно нажать клавишу <Print Screen> при включенном принтере. На листе будет напечатано изображение экрана. Для печати экрана, работающего в *графическом режиме*, необходимо до нажатия <Print Screen> дать команду MS-DOS: graphics /r. В случае работы под управлением Windows нажатие <Print Screen> помещает изображение экрана в *буфер обмена*, откуда другие программы Windows могут его извлечь. Если нажать сочетание клавиш <Alt>+<Print Screen>, то в буфер обмена помещается изображение *активного окна*;
- <Scroll Lock> — клавиша прокрутки (от англ. *scroll* — прокручивать, просматривать). Включает режим Scroll Lock (светится индикатор Scroll Lock). В программах может использоваться для включения и отключения прокрутки текста по экрану в вертикальном направлении и для других целей;
- <Ctrl>, <Alt> — эти клавиши применяются только в сочетании с другими клавишами для того, чтобы дать программе сигнал выполнить то или иное действие. Например, одновременное нажатие клавиш <Ctrl>, <Alt> и <Del> (сочетание <Ctrl>+<Alt>+<Del>) приводит к перезагрузке компьютера, работающего под управлением операционной системы MS-DOS. Клавиша <Alt> служит для *Алт-ввода символов*. На клавиатуре имеются две клавиши <Alt> и две клавиши <Ctrl> (правые и левые). Действие правых и левых клавиш одинаково;
- <Enter> — предназначена для ввода команд или строк. На некоторых клавиатурах обозначается <Return> или <CR>. Нажав ее после набора команды или строки, мы делаем их доступными для программы. В *текстовых редакторах* служит для перевода строки, как на обычной пишущей машинке. (см. *клавиатура пишущей машинки*) Для удобства ввода на клавиатуре имеются две клавиши <Enter>, действие которых одинаково;

- <Pause>/<Break> — эта клавиша имеет два указанных названия и при нажатии вызывает приостановку действий *центрального процессора*. Возобновить деятельность процессора можно нажатием клавиши <Enter>. Одновременное нажатие клавиш <Ctrl> и <Break> (сочетание <Ctrl>+ +<Break>) часто используется в программах для прерывания работы этих программ;
- <Num Lock> — включает и выключает режим ввода цифр с *малой цифровой клавиатуры*. При включенном режиме светится индикатор Num Lock

СПЕЦИАЛЬНЫЙ СИМВОЛ [special character]. Графический символ, не являющийся ни буквой, ни цифрой, ни пробелом

СПЕЦИФИКАТОР, описатель [declarator, specifier]. В *языках программирования* — *ключевое слово*, определяющее *тип данных*. Например, `real`, `integer`, `array`

СПЕЦИФИКАЦИЯ [specification]. 1. Точное и полное формальное описание свойств, характеристик и функций программы, *элемента данных* или другого объекта. Например, С. программы — точное описание того результата, который необходимо достичь с помощью программы. Она содержит полную информацию, необходимую для построения алгоритма, т. е. определяет, что должна выполнять программа, не указывая, как она должна это делать. Если программа начинает свою работу с ввода *исходных данных* и заканчивает выводом результатов, ее С. может иметь форму описания множеств входных и выходных величин. Если программа осуществляет действия, не связанные с преобразованием исходных данных в конечный результат, то С. содержит точное описание отдельных функций, реализуемых программой. В *языках программирования* С. — определение типа данных с помощью *ключевых слов*. 2. Перечень документов, определяющих *программный продукт* (это значение термина устанавливалось требованиями *Единой системы программной документации* и в настоящее время используется сравнительно редко)

СПЕЦИФИКАЦИЯ ИНТЕРФЕЙСОВ СЕТЕВЫХ УСТРОЙСТВ, сетевой стандарт NDIS [network device interface specification (NDIS)]. Стандарт, определяющий взаимодействие драйверов *сетевых адаптеров* и стеков протоколов в *локальных вычислительных сетях*. Драйвер, написанный в этом стандарте, может поддерживать *транспортный протокол* для всех сетевых адаптеров. Стандарт разработан совместно компаниями Microsoft и 3Com

СПЕЦИФИКАЦИЯ IrDA [Infrared Data Association (IrDA)]. Стандарт на беспроводную передачу данных с использованием инфракрасного излучения. Разработан Ассоциацией по средствам передачи данных в инфракрасном диапазоне (Infrared Data Association). Применяется при разработке средств *инфракрасного интерфейса* (*инфракрасных портов* и *программного обеспечения* к ним)

СПИН-БЛОКИРОВКА [spin lock]. Механизм синхронизации, используемый ядром и другими частями *операционной системы Windows NT*. С.-б. обеспечивает взаимоисключающий доступ к системным *структурам данных* в многопроцессорной среде

СПИСОК [list]. 1. *Элемент управления*, предназначенный для выбора одного или нескольких значений из их предопределенного набора. Ср. *раскрывающийся список*. То же, что *список выбора*. 2. *Структура данных*, состоящая из некоторого количества однородных элементов данных, каждый из которых содержит *указатель* на следующий элемент

СПИСОК ВЫБОРА [pick list]. *Элемент управления*, предназначенный для выбора одного или нескольких значений из их предопределенного набора. Ср. *раскрывающийся список*

СПИСОК АРГУМЕНТОВ [argument list]. То же, что *список фактических параметров*

СПИСОК ПАРАМЕТРОВ [parameter list]. То же, что *список формальных параметров*

СПИСОК СОВМЕСТИМОГО ОБОРУДОВАНИЯ [hardware compatibility list (HCL)]. Список всех *аппаратных средств*, тестированных компанией Microsoft на предмет совместимости с *операционной системой Windows 2000*. Следует свериться с ним перед инсталляцией системы

СПИСОК ФАКТИЧЕСКИХ ПАРАМЕТРОВ, список аргументов [argument list]. Последовательность *фактических параметров*, указываемая в *обращении к процедуре* или *функции*. Правила написания С. ф. п. определяются синтаксисом конкретного *языка программирования*. Обычно С. ф. п. следует за именем процедуры или функции и заключается в круглые скобки. Параметры отделяются друг от друга запятыми или другими разделителями. Если у процедуры (функции) нет формальных параметров, у нее не должно быть и параметров фактических. Так как С. ф. п. содержит фактические параметры, подставляемые при *вызове процедуры* вместо формальных, содержащихся в *списке формальных параметров*, то соответствие двух параметров — формального и фактического — определяется очередностью их перечисления в соответствующих списках. См. *оператор процедуры, вызов функции, именованные аргументы, значение по умолчанию*

СПИСОК ФОРМАЛЬНЫХ ПАРАМЕТРОВ, список параметров [parameter list]. Последовательность *формальных параметров*, указанная в *заголовке процедуры* или *функции*. Правила написания С. ф. п. определяются синтаксисом конкретного *языка программирования*. Обычно С. ф. п. следует за именем процедуры или функции и заключается в круглые скобки. Параметры отделяются друг от друга запятыми или другими разделителями. В некоторых

языках (например, в Паскале) после каждого параметра указывается его тип. См. *описание процедуры, описание функции*

СПИСОК ЦИКЛА [cycle list]. Часть *заголовка цикла*, определяющая правило вычисления значений *параметра цикла* и *условие выхода из цикла*. Например, в написанных на языках Фортран, Паскаль и Си заголовках цикла `DO 10 K = 1, 50`, `for i:= 1 to 65 do` и `for (count = 1; count <= number; count++)` С. ц. являются соответственно: `K = 1, 50`, `i:= 1 to 65` и `count = 1; count <= number; count++`

СПРАВКА, справочная система [help]. Программное средство, предназначенное для предоставления пользователю информации о некотором приложении и о порядке работы с ним в *интерактивном режиме*, т. е. непосредственно во время работы с приложением. Современная С., как правило, является *гипертекстом* и допускает несколько различных способов получения информации. Например, С. многих приложений Windows позволяет найти нужный раздел по иерархическому оглавлению, по предметному указателю, содержащему *ключевые слова*, и просто по набору слов, встречающихся в тексте. См. *контекстно-зависимая справка*. Ср. *оперативная помощь*

СПРАВОЧНАЯ СИСТЕМА [help]. То же, что *справка*

СПУЛЕР [spooler]. 1. То же, что *программа буферизации*. 2. То же, что *спулер печати*

СПУЛЕР ПЕЧАТИ, спулер [print spooler, spooler]. Часть *операционной системы*, управляющая процессом буферизации данных с последующей их выдачей на печать. Представляет собой набор *драйверов виртуальных устройств* и *библиотек динамической компоновки*, объединенных общей архитектурой. С. п. обеспечивает печать в *фоновом режиме* за счет использования фоновой обработки *потоков управления*. Это означает, что спулер передает данные на принтер только тогда, когда принтер готов к получению дополнительной информации. В составе *операционной системы Windows NT* компоненты С. п. реализованы как сервис, который можно остановить или запустить. См. *процессор печати*

СПУЛИНГ [spooling]. Способ применения *буферной памяти* при организации *ввода* и *вывода данных* в компьютерах с *многозадачной операционной системой*. При С. все *выходные данные* задачи временно запоминаются в *буферном файле*, а после ее окончания выводятся независимо от других задач. Аналогично все *входные данные* задачи загружаются в буферный файл до ее выполнения. Причем эти действия совмещаются с выполнением процессором команд той же или другой задачи. Применение С. особенно эффективно в случае использования принтеров, графопостроителей, *устройств ввода* с перфокарт и других медленных устройств. См. *буферная память, буферизация*

СПУЛ-ФАЙЛ [spool file]. То же, что *файл спулинга*

СПУСКАЮЩЕЕСЯ МЕНЮ [pull-down menu]. То же, что *нисходящее меню*

СРЕДА [environment, medium]. 1. Окружение (environment), в котором функционирует объект. Например, С. выполнения *прикладной программы* является *вычислительная система*. 2. То же, что *носитель данных* (medium). 3. То же, что *рабочая среда пользователя*

СРЕДА ПРОГРАММИРОВАНИЯ [programming environment]. Интегрированная *система разработки программ*. В С. п. все *программные средства* имеют единый *пользовательский интерфейс*, общую *базу данных* и не требуют специального вызова, так что программист может выполнять свою работу, не выходя в *операционную систему*. Таким образом, С. п. способна удовлетворить все потребности программиста, связанные с написанием, отладкой и выполнением программы. Ср. *система разработки программ*

СРЕДНЕЕ ВРЕМЯ МЕЖДУ ОТКАЗАМИ [mean time between failures (MTBF)]. То же, что *наработка на отказ*

ССЫЛКА [reference, link]. 1. *Имя*, *указатель* или *адрес* в программе, указывающий на *объект программы*, подпрограмму, другую программу или устройство. С помощью С. программируются *обращения к внешним устройствам*, *вызовы подпрограмм*, указываются *элементы данных* и т. п. 2. То же, что *гиперссылка*

СТАНДАРТНАЯ ПОДПРОГРАММА [standard subroutine]. *Библиотечная программа*, оформленная и описанная так, чтобы пользователи могли применять ее без необходимости знаний о тексте программы. Существуют С. п., которые автоматически встраиваются в процессе трансляции в *объектную программу*. См. *встроенная процедура, встроенная функция*

СТАНДАРТНАЯ ПРОЦЕДУРА [standard procedure]. То же, что *встроенная процедура*

СТАНДАРТНАЯ ФУНКЦИЯ [standard function]. То же, что *встроенная функция*

СТАНДАРТНЫЙ ОБОБЩЕННЫЙ ЯЗЫК РАЗМЕТКИ [standard generalized markup language (SGML)]. То же, что *язык SGML*

СТАНДАРТНЫЙ ТИП [standard type]. То же, что *встроенный тип*

СТАНДАРТЫ EDI [electronic data interchange (EDI)]. Набор стандартов электронного обмена деловыми и финансовыми документами (такими, как заказы на покупку, котировки, накладные и счета-фактуры) между компьютерными программами различных предприятий.

СТАНДАРТЫ H.323 [H.323]. Семейство стандартов *интернет-телефонии*, принятых *Телекоммуникационным сектором Международного союза электросвязи* (ITU-T) в 1992 г. и дополненных в 1998 г. С. H.323 описывают различ-

ные аспекты построения *телефонных сетей* поверх *сетей передачи данных.* На основе С. Н.323 можно строить внутренние телефонные сети, объединять различные сегменты традиционных телефонных сетей, а также организовывать прямые телефонные соединения между пользователями сети Интернет. См. *компьютерно-телефонная интеграция*, *сеть* Н.323

СТАНДАРТЫ IEEE [IEEE standards]. Стандарты на электронную технику (включая *компьютерные сети* и их элементы). Например, С. IEEE 802, регламентируют технологию *сетей Ethernet*

СТАНДАРТЫ ISO [ISO standards]. См. *Международная организация по стандартизации*

СТАНДАРТЫ RFC [RFC standards, RFC documents]. Названия официальных документов *комитета IETF*, содержащих подробное описание стандартов и протоколов Интернета, например, *протоколов TCP/IP.* С. RFC публикуются в изданиях комитета IFTF, которые называются Request for Comments (RFC) — запрос на комментарии (можно перевести, как "предлагается к обсуждению") и имеют порядковый номер. Например, стандарт *протокола удаленных клиентов линейного принтера* опубликован в RFC 1179

СТАНДАРТ ADPCM [ADPCM]. Разработанный *международным советом по мультимедиа* стандарт кодирования и *сжатия данных* о звуковых сигналах в цифровой форме. См. *адаптивная дифференциальная импульсно-кодовая модуляция*

СТАНДАРТ FDDI [fiber distributed data interface (FDDI)]. То же, что *интерфейс для передачи данных по волоконно-оптическим каналам*

СТАНДАРТ MIME [MIME]. Стандарт на кодирование в одном сообщении *электронной почты* сети Интернет текста и нетекстовых двоичных (например, графических) данных. Используется для кодирования и декодирования *вложенных файлов*, отправляемых по электронной почте, а также для чтения и публикации таких файлов в сети Интернет. Заголовок файла с двоичными данными содержит специальный *тип данных* MIME, что служит указанием для программ-клиентов обрабатывать данные иначе, чем текст. См. *многоцелевые расширения почты Интернета*

СТАНДАРТ MPEG [MPEG]. Разработанный *группой MPEG* стандарт на сжатие, хранение и воспроизведение движущихся изображений. В настоящее время существуют четыре С. MPEG, различающихся требованиями к качеству *(разрешающей способности)* цифрового видео и *шириной полосы пропускания.* Все С. MPEG являются открытыми, т. е. за их использование не требуется выплат. См. *группа MPEG, формат MPEG*

СТАНДАРТ UNICODE [Unicode]. Стандарт кодировки, использующий 2 байта для каждого символа. Предложен в 1991 г. некоммерческой организацией Unicode Consortium. Применение этого стандарта позволяет закодировать

очень большое число символов из разных письменностей: в документах Unicode могут соседствовать китайские иероглифы, математические символы, буквы греческого алфавита и кириллицы. При этом становятся ненужными *кодовые страницы*. Коды в С. Unicode разделены на несколько областей. Область с кодами от 0000 до 007F содержит символы набора ASCII с соответствующими кодами. Далее расположены области знаков различных письменностей, знаки пунктуации и технические символы; часть кодов зарезервирована для использования в будущем. Символам кириллицы выделены коды от 0400 до 0451

СТАНДАРТ USB [universal serial bus (USB)]. То же, что *универсальная последовательная шина*

СТАНДАРТ XMI [XML Metadata Interchange]. Предложенный *консорциумом по объектным технологиям* (OMG) стандарт для обмена *метаданными* с помощью *расширяемого языка разметки* XML

СТАТИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ ПАМЯТИ [static memory allocation]. Способ *распределения оперативной памяти*, при котором размещение в ней программы или объектов одной программы (переменных, констант, процедур и т. п.), а также объем занимаемой ими памяти определяются заранее и не меняются в процессе выполнения. Ср. *динамическое распределение памяти*. См. *распределение памяти*

СТАТИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ РЕСУРСОВ [static resource allocation]. Способ *распределения ресурсов*, при котором они назначаются программе до начала ее выполнения. Ср. *динамическое распределение ресурсов*. См. *распределение ресурсов*

СТЕК, магазин [stack, last in first out (LIFO)]. Упорядоченный набор *элементов данных*, в котором можно удалять и добавлять элементы, причем новый элемент всегда записывается в его конец, а очередной читаемый или удаляемый элемент также выбирается из его конца. Таким образом, последний добавляемый элемент С. является единственным доступным и первым удаляемым элементом (по принципу "последним вошел — первым ушел"). Вставку и удаление элемента иногда называют соответственно проталкиванием (push) в С. и выталкиванием (pop) из С. С. нередко используются при проведении вычислений, в частности, для реализации *рекурсивных процедур* или *функций*. В программах С. реализуется в виде списка или в виде массива с двумя указателями — указателем на первый элемент (дно С.) и указателем на последний элемент (вершину С.). С. может быть реализован аппаратными средствами в виде безадресной (т. н. магазинной) памяти с работой в режимах вставки/удаления. Ср. *очередь*

СТЕРЕОТИП [stereotype]. В *языках визуального моделирования* новый вид *элемента модели*, созданный на основе существующих элементов. Стереоти-

пы могут расширять семантику классов *метамодели*, но не ее структуру. Ср. *макрос*

СТИЛЬ [style]. Набор правил для отображения шрифтов, выравнивания текста, параметров текста, фонового рисунка и других объектов документа. Ср. *начертание шрифта*

СТИЛЬ ШРИФТА [font style, type style]. То же, что *начертание шрифта*

СТИРАНИЕ [erase]. Применительно к *программному обеспечению* С. означает физическое уничтожение объекта таким образом, что его дальнейшее использование становится невозможным. Например, С. файла означает удаление информации о файле из каталога и физическую запись вместо информации файла какой-то другой информации, так что восстановление после С. становится невозможным. Ср. *удаление*

СТОЛБЕЦ [column]. То же, что *колонка*

СТОП-СЛОВА [stop words]. Для экономии места и увеличения производительности некоторые *поисковые системы* не включают в индексы слова, встречающиеся на *веб-страницах* очень часто. Например, артикли "a", "the" и т. д.

СТОРОЖЕВОЕ УСЛОВИЕ [guard condition]. Условие, которое должно быть выполнено для того, чтобы был запущен ассоциированный с ним переход в *конечном автомате*

СТРАНИЦА [page]. 1. Совокупность строк печатного документа, в *текстовых редакторах* определяемая набором атрибутов (размерами полей, наличием и содержанием колонтитулов и т. д.). 2. То же, что *страница памяти*. 3. То же, что *веб-страница*

СТРАНИЦА ПАМЯТИ, страница [memory page, page]. Область памяти, имеющая фиксированную длину, ограниченная виртуальным адресным пространством и передаваемая в качестве единицы информации между *внешней* и *оперативной* памятью в системе со *страничным обменом*.

СТРАНИЦА WEB [Web page]. См. *веб-документ*

СТРАНИЧНЫЙ ОБМЕН [page communication]. Автоматический обмен *страницами виртуальной памяти* между *внешней* и *оперативной памятью* компьютера. См. *свопинг*

СТРИМЕР [streamer]. *Устройство* для записи и считывания информации на кассеты с *магнитной лентой*. Часто применяется для *резервного копирования*

СТРОКА, строка символов [string, line]. 1. Последовательность слов, букв, цифр или других знаков, написанных в одну линию. Например, битовая строка — последовательность двоичных цифр (нулей и единиц). 2. *Тип данных*, значениями которого являются последовательности знаков. Обычно реализуется как *одномерный массив* переменной длины, минимальный раз-

мер которого равен единице, а максимальный размер, т. е. длина С., может изменяться. 3. Горизонтальная линия на экране дисплея или на бумаге, заполненная последовательностью знаков или предназначенная для такого заполнения. Например, *командная* С., С. *подсказки* или С. текста при работе в *текстовом редакторе* либо напечатанная принтером

СТРОКА МЕНЮ [menu bar]. *Элемент управления* в *графическом интерфейсе пользователя*, реализующий функции *горизонтального меню*. Ср. *панель инструментов*

СТРОКА ПОДСКАЗКИ [help line]. В *интерактивных системах* — *строка* на экране дисплея, указывающая доступные команды и их смысл. Например, при наведении *указателя мыши* на одну из *кнопок пиктографического меню* в С. п. появляется текст, поясняющий, какие действия вызовет "нажатие" этой кнопки

СТРОКА СОСТОЯНИЯ [status bar]. Строка в окне приложения (как правило, в нижней части окна), в которой отображается информация о выбранной команде, нажатой кнопке *панели инструментов*, режиме работы приложения или о выполняемом в данный момент действии

СТРУЙНЫЙ ПРИНТЕР [ink-jet printer]. *Принтер*, в котором изображение создается струей чернил. Обладает высоким качеством печати, может быть цветным и черно-белым

СТРУКТУРА ДАННЫХ [data structure]. 1. Способ объединения, взаимосвязь или взаимное расположение нескольких *элементов данных*, рассматриваемых как одно целое. С. д. выражает, как из элементов может быть составлена некоторая величина или как ее разделить на элементы. Например, С. д. *двумерного массива* может быть представлена как прямоугольная таблица его элементов, имеющая определенные количества строк и столбцов. При этом первый *индекс массива* рассматривается как номер строки, а второй — как номер столбца. Наличие у данных определенной структуры ведет к необходимости выбора определенной структуры их размещения в памяти, а также и обеспечения набора процедур, которые реализуют допустимые операции над элементами данных с учетом выбранной структуры хранения. Чтобы упростить программирование операций над такими данными, в некоторых языках программирования высокого уровня вводятся специальные *структурированные типы* данных. См. *структурированный тип, массив, класс.* 2. Множество данных, относящихся к определенному *структурированному типу*

СТРУКТУРА С ПЕРЕКРЫТИЕМ [overlay structure]. То же, что *оверлейная структура*

СТРУКТУРИРОВАННЫЙ ТИП [structured type]. *Тип данных*, значения которого состоят из определенного количества компонентов. При этом допускается выполнение операций (действий) как над всем *объектом данных*,

принадлежащим к С. т., так и над его отдельными компонентами. К С. т., например, относятся: массив, компонентами которого являются *элементы массива*, и запись, компонентами которой являются *поля данных*. Любой С. т. характеризуется количеством и типом (типами) входящих в него компонентов, а также способом создания структуры. Некоторые *языки программирования высокого уровня* допускают С. т., компоненты которых также принадлежат к С. т. См. *структура данных, массив, класс*

СТРУКТУРНОЕ ПРОГРАММИРОВАНИЕ [structured programming]. *Парадигма программирования*, направленная на создание логически простых и понятных программ. С. п. основано на предположении, что логичность и понятность программы облегчает разработку, доказательство правильности и последующее *сопровождение программы*, а также обеспечивает ее надежность. Характерными принципами С. п. являются следующие. 1) *Нисходящее программирование*, при котором задача разбивается на несколько более простых частей или подзадач, программируемых в виде подпрограмм (процедур или функций). 2) *Модульное программирование*, при котором относительно независимые подзадачи программируются в виде отдельных *программных модулей*. 3) Использование при программировании трех структур управления (следование, выбор и повторение). Любой алгоритм состоит из последовательности действий, разветвлений и повторений. Их можно описать основными управляющими структурами. Структура "следование" ("цепочка") предполагает естественную последовательность выполнения операторов. Структура "выбор" ("ветвление") задается схемой "если — то — иначе". Ей соответствует *условный оператор* Паскаля `if <условие> then <оператор1> else <оператор2>`. Структуре "повторение" сопоставлен *оператор цикла*. Из этих и только из этих структур может быть построена программа любой подзадачи. 4) Отказ от *безусловных передач управления* и ограниченное использование *глобальных переменных*. С помощью *вызовов подпрограмм* (процедур или функций) все подзадачи связываются в одно целое — структурную программу

СТУПЕНЬ КОНВЕЙЕРА [pipeline segment]. То же, что *сегмент конвейера*

СУБД [DBMS]. То же, что *система управления базой данных*

СУБТРАКТИВНЫЙ ЦВЕТ [subtractive color]. Цвет, получаемый на бумаге с помощью полиграфических красок (голубой, пурпурной, желтой и черной)

СУММАТОР [accumulator]. *Регистр центрального процессора*, используемый для выполнения *арифметических* и *логических операций*

СУПЕРВИЗОР [supervisor]. Управляющая *резидентная программа* в составе *операционной системы*, координирующая распределение и использование ресурсов *вычислительной системы*. В операционной системе может быть несколько С. Например, С. ввода/вывода контролирует состояние средств ввода/вывода, инициирует и прекращает процессы ввода/вывода. С. *основной*

памяти осуществляет учет и *динамическое распределение* области *оперативной памяти*, в которую загружаются программы пользователей и некоторые *системные программы*. С. страниц организует *страничный обмен виртуальной памяти*. Наконец, С. прерываний анализирует прерывания и выбирает необходимый вид *обработки прерываний*. В этом значении термин С. был введен в употребление с появлением операционных систем ЭВМ третьего поколения. В других ситуациях в настоящее время этот термин не употребляется. См. *распределение ресурсов, обработка прерываний*

СУПЕРКЛАСС [superclass]. Класс, являющийся *предком* данного класса в *иерархии наследования*

СУПЕРКОМПЬЮТЕР [supercomputer]. То же, что *суперЭВМ*

СУПЕРСКАЛЯРНАЯ АРХИТЕКТУРА [super scalar architecture]. *Архитектура процессора*, обрабатывающего данные одновременно в двух и более параллельных конвейерах. С. а. позволяет выполнять более одной команды за такт. См. *суперскалярный процессор*

СУПЕРСКАЛЯРНЫЙ ПРОЦЕССОР [super scalar processor]. *Процессор суперскалярной архитектуры*, реализующий параллельное выполнение команд программы, машинный код которой не содержит специально организованную информацию о параллелизме. Обнаружение параллелизма в коде и соответствующая выборка команд делается аппаратно в процессе выполнения программы. Основными компонентами С. п. являются устройства для интерпретации команд, снабженные логикой, позволяющей определить, являются ли команды независимыми, и несколько исполняющих устройств. В исполняющих устройствах могут быть конвейеры. Большинство современных микропроцессоров, включая Pentium, — суперскалярные. Ср. *процессор со сверхдлинным командным словом*. См. *окно выполнения, параллельная обработка, параллельные вычисления*

СУПЕРЭВМ, суперкомпьютер [supercomputer]. ЭВМ, относящаяся к классу самых мощных в настоящее время. Это дорогая *многопроцессорная ЭВМ*, обладающая самыми высокими быстродействием и емкостью памяти. С. может выполнять огромные объемы вычислений за сравнительно короткий промежуток времени. Поэтому С. обычно используется для проведения сложных расчетов, необходимых, например, для управления движением космического корабля, составления прогноза погоды, проведения крупного научного вычислительного эксперимента и т. д.

СХЕМА ДАННЫХ [database scheme, data scheme]. Описание данных и взаимосвязей между ними в *базе данных*. Часто это описание представляют в графической форме, в виде диаграммы специального вида (см. рис. В.5, *вторичный ключ, первичный ключ*)

СЦЕНАРИЙ, скрипт [script, scenario]. 1. План работы программы во взаимодействии с пользователем. Например, создание *компьютерной игры* начинают с разработки С., описывающего, что и в какой последовательности должен выполнять компьютер в зависимости от игровой ситуации и реакций играющего. 2. Описание последовательности действий, выполнение которых приводит к определенному результату. При этом сами действия считаются определенными и их детальное описание в С. не включается. Например, *командный файл* является С. работы *операционной системы*. 3. Программа (особый вид программного кода), обычно написанная на некотором *интерпретируемом* (не на *компилируемом*) *языке* и содержащая отдельные команды-инструкции. Например, С. включается в текст *веб-страницы* в виде исходного кода и интерпретируется системой, установленной на компьютере удаленного пользователя, запросившего эту страницу. См. *Java-сценарий, сценарий регистрации, язык программирования сценариев*

СЦЕНАРИЙ РЕГИСТРАЦИИ [login script, logon script]. *Программа*, *командный файл* или *макрос*, назначенный пользователю *системным администратором* и выполняемый при входе пользователя в *вычислительную систему* или *компьютерную сеть*. С. р. можно редактировать так, чтобы каждому пользователю при *входе в систему* создавалась индивидуальная пользовательская среда. Например, в *операционных системах Windows* применяются различные С. р., позволяющие настраивать пользовательскую среду и запускать различные программы. Эти сценарии (командные файлы с расширением bat или исполняемые — с расширением exe) выполняются при запуске или выключении системы, а также при регистрации пользователя или его выходе из системы. См. *профиль пользователя*

СЦЕНАРИЙ CGI, программа CGI [CGI script]. См. *интерфейс CGI*

Т

ТАБЛИЦА РАЗМЕЩЕНИЯ ФАЙЛОВ, таблица FAT [file allocation table (FAT)]. Таблица для динамического распределения пространства *жесткого диска*. Единицей распределяемой памяти является *кластер*, для которого в таблице указываются номер и состояние (свободен, испорчен и т. д.). Это один из наиболее старых методов разметки дисков. См. *файловая система FAT*

ТАБЛИЦА FAT [file allocation table (FAT)]. То же, что *таблица размещения файлов*

ТАЙМЕР [timer, timer clock]. Программируемое устройство отсчета времени, часы. Т. обеспечивает измерение астрономического времени и позволяет предварительно установить временной интервал, по истечении которого может быть выдан соответствующий сигнал. Этот сигнал можно использо-

вать в программах для прерывания или начала некоторых запрограммированных действий

ТАКСОНОМИЯ ФЛИННА [Flynn classification]. То же, что *классификация Флинна*

ТАКТ, время такта [loop, loop time]. Время выполнения элементарной внутренней операции процессора. Ср. *тактовая частота*

ТАКТОВАЯ ЧАСТОТА [clock rate]. Частота повторения сигналов, синхронизирующих работу микросхем компьютера. Эти сигналы вырабатываются тактовым генератором *центрального процессора* и благодаря своей стабильной частоте используются для создания единого стандарта времени с целью управления всеми процессами, происходящими в устройствах компьютера. Сама Т. ч. применяется для управления теми устройствами ЭВМ, которые имеют наибольшее быстродействие, тогда как работа устройств меньшего быстродействия осуществляется на частотах, получаемых делением Т. ч. Следовательно, скорость выполнения компьютером элементарных операций, а значит, и *производительность компьютера* пропорциональны Т. ч. Поэтому она является основной характеристикой скорости работы центрального процессора. При описании характеристик *персонального компьютера* (см. *конфигурация компьютера*) Т. ч. в мегагерцах указывается сразу за указанием типа центрального процессора. Например, Celeron 733 означает, что Т. ч. используемого в компьютере центрального процессора равна 733 МГц

ТАРИФИКАЦИЯ, биллинг [billing]. Начисление денег и выставление счета за услуги, например, за услуги *интернет-провайдера*

ТБ [terabyte]. То же, что *терабайт*

ТБАЙТ [terabyte]. То же, что *терабайт*

ТВЕРДАЯ КОПИЯ [hard copy]. Копия программы, документа или некоторых данных, выведенная из компьютера на бумагу, пленку или перфокарты. Термин "Т. к." противопоставляется термину "электронная копия", относящемуся к электронным и магнитным *носителям данных*

ТВИНИНГ [tweening]. То же, что *преобразование с промежуточными формами*

ТВИП [twip]. Единица полиграфической системы мер, равная одной двадцатой пункта. Дюйм содержит 1440 твипов. Используется в основном для измерения размеров шрифта

ТЕГ [tag]. 1. В *языках разметки* — код (набор символов), идентифицирующий некоторый элемент документа (заголовок, список и т. д.) и указывающий *браузеру* способ отображения этого элемента. Т. ограничиваются угловыми скобками и могут содержать атрибуты (параметры). Например, браузер показывает только ту часть документа HTML, которая заключена

между Т. <BODY> и </BODY>. 2. *Идентификатор* сообщения в *MPI-программе*. Т. задается пользователем и является целым числом от 0 до 32 767. По Т. процесс, принимающий сообщение, может, например, различить два сообщения, пришедших к нему от одного и того же процесса. См. *интерфейс передачи сообщений*

ТЕЗАУРУС [thesaurus]. Автоматизированный словарь синонимов. Представляет собой файл синонимов и программу, осуществляющую поиск синонимов

ТЕКСТ [text]. Написанная, напечатанная, отображенная на экране или закодированная последовательность символов: букв, цифр, скобок, знаков препинания и *знаков арифметических операций*, *специальных (управляющих)* и *псевдографических символов*. Т. представляет собой информацию в неструктурированном виде. Все допустимые *вычислительной системой* символы могут быть введены в *память ЭВМ* с помощью клавиатуры нажатием либо одной клавиши, либо группы клавиш. Большинство вычислительных систем для представления Т. используют с отдельными модификациями так называемые *коды ASCII символов* и кодировку в *стандарте UNICODE*. Для составления и распечатки Т. служат специальные программы — *текстовые редакторы* и *текстовые процессоры*

ТЕКСТОВАЯ КОНСТАНТА [character constant]. То же, что *символьная константа*

ТЕКСТОВОЕ ПОЛЕ, поле [box, text box]. *Элемент управления*, предназначенный для ввода, отображения и редактирования небольших текстовых значений. На рис. Э.2 приведен пример использования Т. п. в диалоговом окне. Ср. *поле ввода*

ТЕКСТОВЫЙ ПРОЦЕССОР [word processor]. *Текстовый редактор*, дополненный разнообразными средствами оформления текста, среди которых следует отметить: 1) использование библиотек шрифтов; 2) возможность создания и дальнейшего применения шаблонов документов; 3) вставка графических изображений и данных из других программ; 4) орфографический и грамматический контроль, словарь синонимов; 5) автоматическое формирование оглавлений и указателей; 6) создание сносок, *верхних и нижних колонтитулов* и др. Из Т. п., предназначенных для *персональных компьютеров*, наиболее известны Word фирмы Microsoft, WordPerfect фирмы WordPerfect и AmiPro фирмы Lotus. См. *блок орфографического контроля, верстка страниц, внедренный объект, колонтитулы*

ТЕКСТОВЫЙ РЕДАКТОР [text editor]. *Программа*, с помощью которой можно подготовить и распечатать текстовые данные. Этими данными могут быть программа, написанная на *языке программирования высокого уровня*, финансовый документ, научная статья, книга стихов и т. п. Т. р. являются неотъемлемой частью любой *диалоговой системы*. У всех Т. р. есть как ми-

нимум следующие возможности: 1) создание нового *текстового файла*; 2) просмотр и редактирование существующего файла; 3) ввод текста как в *режиме вставки*, так и в *режиме замещения*; 4) поиск, выделение, копирование, удаление, замена, перенос части строки или нескольких строк; 5) поиск, выделение, копирование, перенос части файла в другой файл; 6) разбиение на страницы; 7) печать редактируемого файла. Т. р. могут быть построчными (текст воспринимается, как последовательность строк, разделенных маркерами конца строки), символьными (текст рассматривается как поток символов, в котором признаки конца строки или страницы тоже являются символами) или экранными (экран дисплея образует как бы перемещаемое по тексту окно, в котором можно передвигать курсор и устанавливать его в нужные позиции). Многие современные приложения содержат Т. р. в качестве встроенного компонента

ТЕКСТОВЫЙ РЕЖИМ [character mode]. Режим работы *адаптера дисплея*, при котором на экран выводятся изображения только *текстовых символов* и *псевдографических символов*. При Т. р. экран делится на определенное число символьных позиций (знакомест), которые группируются в строки. Например, стандартный Т. р. позволяет вывести на экран 25 строк по 80 знакомест. Каждое знакоместо представляет собой прямоугольник, состоящий из определенного количества пикселов (точек, имеющих цвет и яркость). В каждом знакоместе может находиться один символ. Работающая программа определяет сам символ, его цвет и цвет фона символа. Эти данные заносятся *центральным процессором* в видеопамять, а затем считываются и преобразуются видеоадаптером в изображение на экране. Качество изображения текста зависит от размера матрицы пикселов, определяющей количество пикселов, приходящихся на одно знакоместо. Если для изображения символа используется матрица 8×8 пикселов (8 столбцов по 8 пикселов в каждом столбце), то на экране можно различить отдельные пикселы, из которых состоит символ. Современные видеоадаптеры могут поддерживать Т. р., в котором матрица пикселов имеет размер 9×16, что обеспечивает высокое качество отображения текста. В других случаях в настоящее время этот термин не используется

ТЕКСТОВЫЙ СИМВОЛ [alphanumeric character, information character]. Буква, цифра, скобка, знак препинания или знак *арифметической операции*. Являясь составной частью текста, Т. с. выступает и частью его содержания, в отличие от *управляющих* и *псевдографических символов*, которые служат для придания тексту удобной для восприятия формы

ТЕКСТОВЫЙ ФАЙЛ [text file]. 1. *Файл*, содержащий *текст*. 2. Файл, состоящий исключительно из *кода ASCII*

ТЕКУЩЕЕ СОСТОЯНИЕ [current state]. Состояние системы, программы, устройства или информации в данный момент времени. См. *слово состояния*

ТЕКУЩИЙ ДИСК, активный диск [current disk, active disc]. *Магнитный* или *лазерный диск*, доступный в данный момент для использования. В *персональных компьютерах* это *текущий дисковод* или *логический диск*

ТЕКУЩИЙ ДИСКОВОД [current disk drive]. *Дисковод*, с которым в данный момент непосредственно работает *операционная система*. В *персональных компьютерах* это дисковод, с которым в настоящий момент работает пользователь. При работе в *операционной системе MS-DOS* Т. д. указан в приглашении, расположенном в *командной строке*

ТЕКУЩИЙ КАТАЛОГ, рабочий каталог [current directory, working directory]. *Каталог*, файлы которого в данный момент непосредственно доступны программам и пользователю. В *персональных компьютерах* это каталог, с которым в настоящий момент работает пользователь. При этом если пользователь указывает в *команде операционной системы* имя файла без *пути к файлу*, то операционная система будет искать или создавать этот файл в Т. к. При работе в *операционной системе MS-DOS* имя Т. к. указано в приглашении, расположенном в *командной строке*

ТЕЛЕКОММУНИКАЦИОННЫЙ СЕКТОР МЕЖДУНАРОДНОГО СОЮЗА ЭЛЕКТРОСВЯЗИ [International Telecommunication Union — Telecommunication standardization sector (ITU-T)]. То же, что *Международный консультативный комитет по телеграфии и телефонии*

ТЕЛЕКОММУНИКАЦИЯ [telecommunication]. Дистанционная связь; передача всех форм информации, включая данные, голос, видео и т. п., на расстояние по *линиям связи*. К Т. имеют отношение телеграфная и телефонная связь, всемирная сеть Интернет

ТЕЛЕКОНФЕРЕНЦИЯ, конференция [computer conference, conference, teleconference]. Набор услуг *компьютерной сети*, обеспечивающий групповое общение пользователей сети по избранной теме. Каждый участник Т. может послать свое сообщение в адрес Т., и оно может быть прочитано любым другим участником. Таким образом, Т. — это "электронная доска объявлений и дискуссий", которая отличается своей довольно обширной рубрикой-направлением. В рамках этой рубрики между участниками Т. проводится ряд дискуссий по конкретным темам, помещаются объявления, советы и т. п. Например, в сети Интернет Т. по направлению "*Вычислительная техника* и смежные области" включает дискуссии по основам вычислительной техники, сведения о новом *программном обеспечении*, информацию об эксплуатации *аппаратных* и *программных средств* и т. д. Специальные программные средства позволяют организовать для пользователей сети большое количество различных Т. и при этом обеспечивают ведение протокола, пересылку и сохранение сообщений. В отличие от *электронной почты* здесь не надо посылать одну и ту же статью поочередно каждому участнику. От-

правка статьи требует времени и действий не более, чем отправка письма по электронной почте. Кроме того, не надо принимать все поступающие в сеть материалы Т. Установленная на вашем компьютере программа чтения Т. отделяет сообщения, которые вы уже видели, и показывает только новые, которые поступили после окончания последнего сеанса. Из них можно выбирать и читать только то, что вас интересует

ТЕЛЕФОННАЯ СЕТЬ [telephony network]. Сеть связи, поддерживающая динамические соединения (звонки) между ее узлами и двустороннюю передачу информации между ними в реальном времени. Узлами Т. с. являются телефон, факс, модем или телефонная станция, обеспечивающая коммутацию между другими узлами. Существуют Т. с. общего пользования и внутренние, используемые непосредственно эксплуатирующей ее организацией для собственных нужд

ТЕЛЕФОННОЕ ПРИЛОЖЕНИЕ [telephony application]. Компьютерная программа, управляющая некоторой телефонной системой. Т. п. могут работать в *диалоговом режиме* или автоматически. Например, существуют Т. п., обеспечивающие пользователю визуальный контроль состояния звонков и "ручное" управление звонками с помощью *графического интерфейса пользователя*. А есть Т. п., предназначенные для автономного управления факсом, автоответчиком, автосекретарем, *голосовой почтой* и т. п., для регулярной проверки состояния банковского счета по телефону. Получили распространение Т. п., реализующие на основе *CTI-плат* все основные функции автоматической телефонной станции для внутренней *телефонной сети* организации или учреждения. См. *компьютерно-телефонная интеграция*, *телефонная сеть*

ТЕЛЕФОННЫЙ ИНТЕРФЕЙС ПРИКЛАДНОГО ПРОГРАММИРОВАНИЯ ДЛЯ ЯЗЫКА JAVA, интерфейс JTAPI [Java Telephony API (JTAPI)]. Элемент *компьютерно-телефонной интеграции — интерфейс программирования приложений для телефонной связи* на *языке Java*. Содержит *управление соединениями* и *управление голосовым каналом*. Допускает *клиент-серверную телефонию*. Служит не только для чисто компьютерных приложений, но благодаря свойствам *виртуальной машины Java* может применяться, например, в сотовых телефонах

ТЕЛЕФОННЫЙ СЕРВЕР [telephony server]. Компьютер, на котором исполняются серверные *телефонные приложения*. Т. с. может быть подключен к управляемой им автоматической телефонной станции (АТС) или же представлять собой собственно АТС на основе *CTI-плат*. См. *компьютерно-телефонная интеграция*, *телефонная сеть*

ТЕЛО ПРОЦЕДУРЫ [procedure body]. Исполняемая часть *процедуры*, представляющая последовательность операторов, реализующую алгоритм процедуры. В некоторых *языках программирования* (например, в Паскале) Т. п.

является *блоком программы*, содержащим раздел описаний, где вводятся *локальные переменные* и другие локализованные в Т. п. объекты. Некоторые языки программирования (например, Паскаль) допускают в Т. п. наличие нескольких *точек входа* в процедуру и *точек выхода* из нее. См. *описание процедуры*

ТЕЛО ФУНКЦИИ [function body]. Исполняемая часть *функции*, представляющая последовательность операторов, реализующую алгоритм функции. В некоторых *языках программирования* (например, в Паскале) Т. ф. является *блоком программы*, содержащим раздел описаний, где вводятся *локальные переменные* и другие локализованные в Т. ф. объекты. Среди операторов, реализующих алгоритм обработки данных, в Т. ф. должен быть оператор, присваивающий имени функции значение, возвращаемое в *вызывающую программу* как результат вычисления функции. Это требование не является обязательным в языках программирования, допускающих, чтобы *вызов функции* не только употреблялся в качестве операнда выражения, но и был отдельным оператором вызывающей программы. Кроме того, правила многих языков программирования требуют наличия в Т. ф. *оператора возврата*, завершающего выполнение функции. Таких операторов может быть несколько. См. *описание функции*

ТЕЛО ЦИКЛА [loop body]. Последовательность повторяемых в *цикле* команд или операторов. В циклах, написанных на *языках программирования высокого уровня*, Т. ц. определяется конструкцией соответствующих предложений. В языке Фортран Т. ц. составляют операторы, стоящие сразу за оператором `DO` вплоть до последнего повторяемого оператора, помеченного указанной в операторе `DO` меткой. См. *оператор цикла языка Фортран*. В языке Паскаль в циклах с параметром и с предусловием Т. ц. образует одиночный или *составной оператор*, стоящий за *ключевым словом* `do`. В цикле с постусловием Т. ц. заключено между ключевыми словами `repeat` и `until`. См. *операторы цикла языка Паскаль*. В языке Си в циклах с параметром и с предусловием Т. ц. образует одиночный или составной оператор, стоящий за закрывающейся скобкой, а в цикле с постусловием Т. ц. заключено между ключевыми словами `do` и `while`. См. *операторы цикла языка Си*. Соответствующие примеры приведены в табл. Т.1

Таблица Т.1. *Примеры циклов*

Язык	Цикл	Тело цикла

Фортран	`DO 10 I = 1, N`
	`IF (A[I]<AMIN) GOTO 20`
	`10 CONTINUE`
		`IF (A[I]<AMIN) GOTO 20`
		`10 CONTINUE`

Таблица Т.1 *(окончание)*

Язык	Цикл	Тело цикла

Паскаль	`while abs(dx) > eps do begin`
	`dx := (a/x – x) * 0.5;`
	`x : = x + dx`
	`end;`
		`begin`
		`dx := (a / x – x) * 0.5;`
		`x := x + dx`
		`end`

	`for l := 10 downto k+1 do` `x[l]:= x[l] – x[k];`	`x[l]:= x[l] – x[k]`

	`repeat`
	`dx : = (a/x – x) * 0.5;`
	`x := x + dx`
	`until Abs(dx) < eps;`
		`dx : = (a / x – x) * 0.5;`
		`x := x + dx`

Си	`while (summa<1000) {`
	`summa = summa + step;`
	`step = 2 * step;`
	`}`
		`{`
		`summa = summa + step;`
		`step = 2 * step;`
		`}`

	`do{ch=getchar(); putchar(ch);}` `while(ch!= '\n');`	`{ch=getchar(); putchar(ch);}`

	`for ( i = 1; i < = 10; i++) {`
	`c = a[i]; a[i] = b[i]; b[i] = c;`
	`}`
		`{`
		`c = a[i]; a[i] = b[i]; b[i] = c;`
		`}`

ТЕНЕВОЙ ФАЙЛ [shadow file]. *Файл*, в который в процессе *спулинга* записывается административная информация к *заданию на печать* (имя пользователя, имя документа и тип данных). В *операционной системе Windows NT* по умолчанию Т. ф. имеет расширение shd

ТЕОРЕТИЧЕСКАЯ ПРОИЗВОДИТЕЛЬНОСТЬ КОМПЬЮТЕРА [computer peak performance]. То же, что *пиковая производительность компьютера*

ТЕРАБАЙТ, Тб, Тбайт [terabyte (TB)]. Единица измерения количества информации и *емкости памяти*. 1 Тбайт = 1024 Гбайт = 1 048 576 Мбайт = = 1 073 741 824 Кбайт = 1 099 511 627 776 байт

ТЕРАФЛОПС, Тфлопс [TeraFLOPS, TFLOPS]. Единица измерения вычислительной *производительности суперкомпьютера*, равная триллиону *арифметических операций* с плавающей точкой в секунду. 1 Тфлопс = 10^3 Гфлопс = = 10^6 Мфлопс = 10^{12} флопс

ТЕРМИНАЛ [terminal]. Устройство или набор устройств, предназначенных для взаимодействия пользователя с *вычислительной системой* или *компьютерной сетью*. Т. состоит из *устройства ввода* (чаще всего это клавиатура) и одного или нескольких *устройств вывода* (дисплей, принтер и т. п.). Наиболее распространен видеотерминал, состоящий из дисплея и клавиатуры. В качестве Т. большой вычислительной системы или сети ЭВМ может выступать подключенный к ним и управляемый *центральным процессором персональный компьютер* (см. *интеллектуальный терминал*). Т. называют также любое удаленное от компьютера устройство, являющееся источником или получателем данных, например, вынесенный в другое помещение принтер

ТЕСТИРОВАНИЕ ПРОГРАММЫ [program testing]. То же, что *испытание программы*

ТЕСТ ПРИ ВКЛЮЧЕНИИ ПИТАНИЯ, самодиагностика при включении питания, самотестирование при включении [power-on self test (POST)]. Набор диагностических и конфигурационных процедур BIOS, которые хранятся в ПЗУ и автоматически выполняются при включении или перезапуске компьютера. Во время Т. п. в. п. определяется реальный *объем памяти* и наличие необходимых *аппаратных средств*. При этом происходит тестирование разнообразных систем компьютера (например, *оперативной памяти*, дисководов, клавиатуры и т. д.) и проверка их правильного подключения. В случае обнаружения неисправностей вырабатывается соответствующий предупреждающий сигнал. Если проверка прошла нормально, начинается *загрузка операционной системы*

TEX [TeX]. *Язык программирования* печатных документов. Предназначен для подготовки на компьютере научно-технических публикаций. Т. был разработан Д. Кнутом в конце 1970-х гг. и до сих пор конкурирует с современными *настольными редакционно-издательскими системами*. Свое название Т. получил по первым трем буквам греческого слова, обозначающего искусство, и созвучию с корнем английского слова *technology*. На языке Т. можно создавать программы, обеспечивающие печать книг, статей, научной документации и т. п. полиграфических изданий как на компьютерных принтерах, так и типографским способом. Т. позволяет разрабатывать различные стили оформления страниц с текстом, сравнимым по внешнему виду с выходящим из-под руки высококлассного мастера набором. Кроме того, в Т. легко набирать технические тексты с большим количеством математических формул, вставлять в текст рисунки, графики и диаграммы. Т. является *проблемно-ориентированным языком*. Его управляющие операторы содержат *ключевые слова* — аналоги профессиональных научных и полиграфических терминов, например, `\int` вставляет в формулу знак интеграла, а `\title`, `\author` указывают транслятору, как обрабатывать следующий за ними текст. Т. также является средством обмена научной информацией. Например, автор может послать по *электронной почте* в редакцию журнала свою статью в виде *ис-*

ходной программы на языке Т. Там внесут в эту программу лишь одну команду, задающую соответствующий журналу стиль оформления, и статью можно тиражировать

ТЕХНИЧЕСКИЕ СРЕДСТВА [hardware]. То же, что *аппаратные средства*

ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ [hardware]. То же, что *аппаратное обеспечение*

ТЕХНИЧЕСКОЕ УСТРОЙСТВО [device, unit]. То же, что *устройство*

ТЕХНОЛОГИЯ [technology]. Система взаимосвязанных правил, стандартов и приемов изготовления продукции в производственном процессе

ТЕХНОЛОГИЯ ACTIVEMOVIE [ActiveMovie]. Технология цифрового видео, позволяющая просматривать видеофильмы в форматах AVI, QuickTime или MPEG прямо на веб-странице. Поддерживается многими *компьютерными платформами*

ТЕХНОЛОГИЯ ACTIVEX [ActiveX]. Разработанная корпорацией Microsoft на основе *компонентной модели объектов* технология взаимодействия *программных объектов*, обеспечивающая удобное встраивание готовых компонентов в *приложения*. См. *элементы управления ActiveX*

ТЕХНОЛОГИЯ CORBA [common object request broker architecture (CORBA)]. Модель объектов и технология ее поддержки, разработанная *консорциумом по объектным технологиям* (OMG). Основу Т. CORBA составляет так называемый *объектный брокер запросов* (Object Request Broker, ORB), который управляет взаимодействием клиентов и серверов *распределенного приложения*. ORB должен быть установлен на каждом компьютере, где исполняются программы, использующие Т. CORBA. ORB реализован для всех основных аппаратных и программных платформ, поэтому распределенные приложения, использующие Т. CORBA, могут выполняться в гетерогенных (неоднородных) сетях. Ср. *распределенная компонентная модель объектов*

ТЕХНОЛОГИЯ DYNAMIC HTML [Dynamic HTML technology]. Расширение языка HTML, предложенное и впервые реализованное корпорацией Microsoft в браузере Internet Explorer 4.0. Т. Dynamic HTML позволяет автору *веб-страницы* запрограммировать управление отображением страницы, задавая процедуры реакции на события `OnLoad`, `OnExit` и `MouseOver`. Ранее интерактивные возможности узла Web реализовывались с помощью *языков сценариев* (Perl, JavaScript, VBScript) или обеспечивались приложениями, созданными на языках Visual Basic или Java

ТЕХНОЛОГИЯ MMX [MultiMedia Extension (MMX)]. Разработанная корпорацией Intel технология ускорения обработки *мультимедиа*. Т. MMX ускоряет обработку таких ключевых элементов, как аудио- и видеоинформация, трехмерная графика, анимация и распознавание образов. Т. MMX применя-

ется в *процессорах MMX*, Pentium II и др. Т. MMX использует свойство параллелизма, присущее большинству алгоритмов обработки мультимедийной информации

ТЕХНОЛОГИЯ MPI [message passing interface (MPI)]. То же, что *интерфейс передачи сообщений*

ТЕХНОЛОГИЯ OLE [OLE technology]. То же, что *связывание и внедрение объектов*

ТЕХНОЛОГИЯ PLUG AND PLAY [plug and play] (англ. "подключи и работай"). *Технология*, базирующаяся на возможностях BIOS, *операционной системы* и внешних устройств и позволяющая компьютерам автоматически обнаруживать и настраивать подключенные устройства и устанавливать соответствующие драйверы. Автоматическое конфигурирование устройств помогает избежать конфликтов между адаптерами и оптимально настроить систему в целом. Основу Т. Plug and Play составляет набор спецификаций, разработанный корпорацией Intel

ТЕХНОЛОГИЯ PVM [parallel virtual machine (PVM)]. То же, что *параллельная виртуальная машина*

ТИП ДАННЫХ [data type]. Понятие *языков программирования высокого уровня*, обозначающее множество допустимых в *вычислительной системе* значений, объединенных совокупностью применимых к ним операций. Указывая, к какому Т. д. относится *объект программы* или выражение, программист определяет все множество значений, которые они могут принимать в данной вычислительной системе, и совокупность допустимых над ними операций. Например, *переменные целого типа* могут принимать только целочисленные значения — положительных, отрицательных чисел или нуля, над которыми допустимы *арифметические операции* и *операции сравнения*. А переменные *логического типа* принимают только два значения — "ИСТИНА" или "ЛОЖЬ", над которыми допустимы *логические операции*. Благодаря разработанной системе Т. д. в языках высокого уровня можно работать с данными, не вникая в детали их представления, хотя в большинстве языков программирования Т. д. позволяет указать и форму представления *элемента данных* в памяти. В каждом языке определен ряд Т. д., называемых *встроенными* или *стандартными* Т. д. В большинство существующих языков программирования встроены целый, вещественный, логический и символьный Т. д. Тип присутствующей в программе константы компилятор может распознать по ее "внешнему виду". Однако в случае переменной необходимо, чтобы ее тип был объявлен в описании. Объявление Т. д. нужно компилятору, чтобы не только верно компилировать программу, но и контролировать правильность использования данных в ней. Для объявления встроенных типов применяются *ключевые слова* — описатели, например, `integer`, `real`, `boolean`, `character` и т. п. или их сокращения. Кроме встроенных Т. д. во

многих языках программирования (например, языках Паскаль, Си и др.) имеются средства конструирования Т. д., удобных для программиста. Такие Т. д. включаются в программу с помощью *описаний типа*. См. *структурированный тип*

ТИП ДАННЫХ ЗАДАНИЯ НА ПЕЧАТЬ [print job data type]. *Тип данных*, обозначающий множество допустимых значений, которые может принимать *задание на печать*, и определяющий действия *спулера печати* по обработке этих заданий. Например:

- тип PSCRIPT1 указывает, что задание на печать состоит из кода на *языке PostScript*, поступившего от клиента Macintosh, а печатающее устройство, на которое должно быть направлено это задание, не является принтером PostScript. Спулер печати интерпретирует код PostScript и создает растровый образ страницы. *Интерфейс графических устройств* и *драйвер принтера* смогут преобразовать этот растр в язык, используемый печатающим устройством;
- тип RAW указывает спулеру, что он не должен вносить никаких изменений в задание на печать, т. к. оно уже имеет формат, в котором должно выводиться на распечатку;
- тип RAW [FF Appended] указывает спулеру на то, что задание на печать поступило от приложения, которое не добавляет заключительный символ подачи страницы в конец каждого задания на печать, и спулер добавляет в конец задания символ подачи страницы, но не вносит никаких других изменений;
- тип RAW [FF Auto] дает спулеру указания проверять наличие завершающего символа подачи страницы в конце поступающих заданий. Если символ подачи страницы уже присутствует, спулер не добавляет его. Никаких других изменений в задание спулер не вносит.

ТИП "ЗАПИСЬ" [record type]. *Структурированный тип* данных, значения которых состоят из определенного числа компонентов, называемых полями. В отличие от массивов поля могут принадлежать к различным *типам данных,* и индексировать их с помощью выражения нельзя. Поэтому каждое поле записи имеет свое имя. При *описании типа* определяются имя типа и список полей с указанием имени и типа каждого поля. *Область определения* имени поля — запись, в которой оно определено. Над переменной Т. з. в некоторых языках (например, в Паскале) можно выполнять только две операции — присваивание и выборку значений компонентов. Т. з. является очень удобным механизмом представления данных, позволяющим вводить в программу нестандартные типы данных. Например, тип данных "комплексные числа" можно ввести в программу на Паскале, как Т. з. с двумя полями стандартного типа `real`, с помощью следующего описания типа:

```
type Complex = record Re, Im: real end;
```

где `type` (тип) — *ключевое слово* описания типа, `Complex` — имя создаваемого типа, `record` (запись) — ключевое слово, обозначающее Т. з., а `Re`, `Im` — имена полей записи, каждое из которых принадлежит к вещественному типу. Если затем поместить в программу *описание переменных*

```
var x, y, z: Complex;
```

то в программу будут введены переменные `x`, `y`, `z`, относящиеся к типу `Complex`, т. е. записи, состоящие из двух полей, с которыми можно обращаться как с вещественной и мнимой частями. Например, операторы `x.Re = 5; x.Im = -3;` зададут переменной `x` значение `5-3i`

ТИП ФАЙЛА [file type]. Характеристика *файла*, отражающая его назначение и область применения, например, командный, текстовый, фортран-программа и т. п. Обычно Т. ф. указывается в *расширении имени файла*

ТИП CHAR [character type]. То же, что *символьный тип*

ТИП REAL [real type]. То же, что *вещественный тип*

ТОН [hue]. Одна из трех количественных характеристик цвета в *цветовой модели* HSB (тон—насыщенность—яркость), позволяющая отличить один цвет от других. Физически зависит от частоты световой волны. Определяется по угловой шкале цветового круга: 0° — красный, 60° — желтый, 120° — зеленый, 180° — голубой, 240° — синий, 300° — фиолетовый

ТОНЕР [toner]. Порошковый краситель, используемый для печати в копировальных аппаратах и лазерных принтерах. Т. наносится на бумагу в соответствии с печатаемым изображением, а затем путем нагрева расплавляется и закрепляется на бумаге

ТОПОЛОГИЯ СЕТИ [network topology]. Общая схема *компьютерной сети*, отображающая физическое расположение *узлов сети* и соединений между ними. См. *древовидная сеть, кольцевая сеть, радиальная сеть, шинная сеть*

ТОЧКА ВОЗВРАТА [return point]. Место в *вызывающей программе*, в которое осуществляется *возврат* из подпрограммы после ее завершения. В машинной программе это адрес. В программе на *языке программирования высокого уровня* считается, что либо Т. в. находится непосредственно за *оператором процедуры* или *вызовом функции*, либо Т. в. является *помеченный оператор*, по метке которого осуществляется возврат. См. *вызов подпрограммы, вызов процедуры, вызов функции*

ТОЧКА ВХОДА [entry point]. 1. Место (адрес) в подпрограмме, куда передается управление из *вызывающей программы*. 2. *Синтаксическая конструкция*, отмечающая начало последовательности действий в процедуре. Если в процедуре одна Т. в., считается, что она находится непосредственно перед первым *исполняемым оператором*. См. *вызов подпрограммы, вызов процедуры, вызов функции*

ТОЧКА ВЫХОДА [exit point]. Точка, в которой происходит передача управления из подпрограммы

ТОЧКА ПРЕРЫВАНИЯ, контрольная точка [breakpoint]. 1. Место в программе, где ее выполнение приостанавливается, с тем чтобы пользователь мог проверить значения переменных и осуществить другие действия. См. *отладка программы*. 2. Адрес команды, на которой остановлено выполнение программы для *обработки прерывания*. По завершении обработки с этого адреса будет продолжено выполнение программы, если прерывание не было связано с *аварийным завершением* программы

ТОЧНАЯ ЗАПИСЬ [careful write]. Алгоритм обновления данных на диске, предотвращающий недопустимые нарушения целостности *файловой системы* в случае сбоя. Файловая система с Т. з. упорядочивает запросы на ввод/вывод и организует изменения данных на диске таким образом, чтобы любые все же возникшие ошибки можно было полностью исправить в удобное время

ТРАНЗАКЦИЯ [transaction]. Совокупность операций с *базой данных*, которые должны быть выполнены обязательно до конца, чтобы база данных оказалась в непротиворечивом состоянии. Например, в финансовой базе данных при переводе денег с одного счета на другой должны быть выполнены по меньшей мере две операции: увеличение суммы на одном счете и соответствующее уменьшение суммы на другом. Если по какой-либо причине одна из этих операций будет выполнена, а другая — не будет, то база данных окажется в неправильном, противоречивом состоянии, т. е. окажется нарушенной *целостность данных*. На время выполнения Т. данные, с которыми производятся операции, блокируются, с тем чтобы другие программы не могли получить доступ к данным, находящимся в противоречивом состоянии. По *завершении Т.* блокировка снимается. Если же успешно завершить Т. не удается, то СУБД производит *откат Т.*, возвращая базу данных в исходное непротиворечивое состояние. См. *захват записи, обработка транзакций*

ТРАНСЛЯТОР [translator]. *Программа* или *техническое устройство*, выполняющее *трансляцию программы*. Т. является одним из основных средств автоматизации программирования. Применение Т. не только облегчает составление отдельной программы, но и позволяет использовать для разных компьютеров один и тот же алгоритм, написанный на некотором *языке программирования*. Различают два вида Т.: интерпретатор, который одновременно и транслирует, и выполняет заданную программу, делая это покомандно или пооператорно, и компилятор, который преобразует исходную программу, составленную на *языке программирования высокого уровня*, в программу на *машинном языке* или языке, близком к машинному *(объектная программа)*, не участвуя в ее исполнении. Т. являются неотъемлемой частью *систем программирования*. Многие Т. выполняют оптимизацию объектной програм-

мы, улучшая некоторые ее характеристики, например, сокращая *время выполнения* программы. Такие Т. называются оптимизирующими. См. *интерпретация, компиляция, система разработки программ*

ТРАНСЛЯЦИЯ ПРОГРАММЫ [program translation]. Перевод *программы* с одного *языка программирования* на другой. Обычно Т. п. является преобразованием программы, написанной на *машинно-независимом языке*, в эквивалентную программу на *машинном языке* конкретной ЭВМ. Т. п. осуществляется самим компьютером по специальной программе, называемой *транслятором*. См. *компиляция, интерпретация*

ТРАНСПОРТНЫЙ ПРОТОКОЛ, протокол транспортного уровня [transport protocol]. *Сетевой протокол*, регламентирующий передачу (транспортировку) сообщений по сети. Т. п. устанавливает способы разделения сообщений на пакеты для передачи и соединения их при приеме, а также правила задания приоритета сообщений и управления их потоком между *узлами сети*. См. *транспортный уровень, сетевой уровень*

ТРАНСПОРТНЫЙ УРОВЕНЬ [transport layer]. Четвертый из семи уровней *модели ISO/OSI*, регламентирующий межкомпьютерную связь. Т. у. отвечает за распознавание и коррекцию ошибок, а также гарантирует надежную доставку сообщений, создаваемых на *уровне приложений*. Т. у. переупаковывает сообщения (подобно тому, как *сетевой уровень* обрабатывает *фреймы данных*), разрезая на несколько пакетов длинные сообщения и объединяя короткие. На принимающем компьютере Т. у. выполняет распаковку сообщений, сборку исходных сообщений и отправляет уведомление о приеме

ТРАНСПЬЮТЕР [transputer]. Сверхбольшая *интегральная схема*, содержащая все основные элементы компьютера: микропроцессор, собственную *оперативную память* и средства доступа к *внешней памяти*. Кроме того, Т. имеет *каналы* межпроцессорной *связи*, которые позволяют монтировать несколько Т. на одной плате, создавая *многопроцессорные ЭВМ*. Т. применяются в качестве плат-ускорителей для повышения производительности существующих *персональных компьютеров*; в качестве *встроенных компьютеров*, например, в многопроцессорных ЭВМ общего назначения для организации *параллельных вычислительных процессов* и в *лазерных принтерах*

ТРАНСФОРМАЦИЯ [transformation]. Изменение положения и формы объекта на экране (перемещение, вращение, наклон, отражение и т. д.) с помощью команд или *инструментами машинной графики*

ТРАССИРОВКА ПРОГРАММЫ [program trace]. Выполнение программы или ее участка, сопровождающееся выводом на экран, принтер или другой регистрацией в хронологической последовательности информации о событиях, связанных с выполнением программы. Т. п. применяется при *отладке* или *тестировании программы*, когда программа пользователя или ее отлаживаемый участок выполняются под управлением специальной программы-

трассировщика. При этом, например, можно выводить на экран или в некоторый регистрирующий файл все встретившиеся при выполнении программы *помеченные операторы* в той последовательности, в которой они фактически выполняются. Таким образом может отслеживаться программная логика. При Т. п. можно контролировать и значения переменных, важных для поиска ошибки, и т. д. Информация об отладочных действиях задается трассировщику, который изменяет *объектную программу* в *оперативной памяти*, размещая в точках трассировки команды перехода на программу трассировки, выполняющую требуемую регистрацию. Более сложные трассировщики могут также изменять объектную программу, исключать и вставлять операторы, позволяя таким образом программисту в течение одного и того же *прогона программы* не только локализовать ошибки, но и попытаться исключить их и посмотреть на результат этого исключения

ТРАФИК [traffic]. Поток данных по *линии связи* или в *сети передачи данных*

ТРЕБОВАНИЕ [requirement]. Черта, свойство или поведение, которое желательно иметь в системе. См. *анализ требований*

ТРЕКБОЛ, шаровой манипулятор [track ball]. *Указательное устройство*, которое отличается от мыши тем, что его шарик находится сверху и вращается пальцами (рис. Т.1). Обычно применяется в *портативных компьютерах*

Рис. Т.1. Конструкции трекбола

ТУПИК [deadlock]. То же, что *тупиковая ситуация*

ТУПИКОВАЯ СИТУАЦИЯ, взаимная блокировка, тупик [deadlock]. Ситуация, возникающая при *параллельной обработке*, когда несколько процессов, использующих общие ресурсы, не позволяют друг другу продолжать работу. Например, пусть есть два процесса, А и Б, которым требуются ресурсы X и Y. Предположим, процесс А сначала запрашивает ресурс X и захватывает его, а процесс Б параллельно захватывает ресурс Y, который в это время еще свободен. В результате возникает ситуация, когда ни один из двух процессов не может продолжать работу: процесс А ждет, когда процесс Б закончит свою работу и освободит ресурс Y, а процесс Б ждет, когда процесс А закончит свою работу и освободит ресурс X. Т. с. может возникнуть при работе

операционной системы, *системы управления базой данных* и др. Известны различные методы предотвращения Т. с. и выхода из Т. с. Например, если СУБД обнаруживает, что возникла Т. с. при работе двух *транзакций* из-за взаимной *блокировки записей*, то СУБД выбирает одну из транзакций и производит ее *откат*, в результате чего записи освобождаются и вторая транзакция завершает свою работу. После этого СУБД повторно запускает первую транзакцию

ТФЛОПС [TFLOPS]. То же, что *терафлопс*

У

УДАЛЕНИЕ [delete]. Применительно к *программному обеспечению* У. означает такое изменение состояния объекта, что его дальнейшее использование становится невозможным. У., как правило, не подразумевает физического уничтожения объекта, поэтому в некоторых случаях удаленный объект может быть восстановлен, т. е. переведен в такое состояние, в котором возможно его использование. См. *удаление каталога, удаление файла, Корзина.* Ср. *стирание*

УДАЛЕНИЕ КАТАЛОГА [directory delete]. Процедура исключения каталога. У. к. выполняется с помощью стандартных средств *операционной системы.* Например, в *оболочке Norton Commander* У. к. удобно выполнять с помощью *горячих клавиш.* Чтобы удалить каталог, необходимо в активной панели NC установить *цветовой маркер* на строку с именем нужного каталога и нажать клавишу <F8>. В *операционной системе Windows* У. к. удобно выполнять с помощью программы *Проводник Windows.* Для этого следует *выделить* удаляемый каталог или группу каталогов и нажать клавишу <Delete> (или щелкнуть правой кнопкой мыши и выбрать соответствующую команду из контекстного меню). См. *удаление файла, Корзина.* Ср. *копирование каталога, перенос каталога, создание каталога*

УДАЛЕНИЕ ФАЙЛА [file delete]. Процедура исключения файла из каталога. У. ф. выполняется с помощью стандартных средств *операционной системы.* Например, в *оболочке Norton Commander* У. ф. удобно выполнять с помощью *горячих клавиш.* Чтобы удалить файл, необходимо в активной панели NC установить *цветовой маркер* на строку с именем нужного файла и нажать клавишу <F8>. В *операционной системе Windows* У. ф. удобно выполнять с помощью программы *Проводник Windows.* Для этого следует *выделить* удаляемый файл или группу файлов и нажать клавишу <Delete> (или щелкнуть правой кнопкой мыши и выбрать соответствующую команду из контекстного меню). У. ф. не означает его физическое стирание с носителя данных. В момент У. ф. стирается только информация о нем в *каталоге* диска. По-

этому, если вы какой-либо файл удалили случайно и сразу заметили это, файл можно восстановить. См. *восстановление файла, Корзина*

УДАЛЕННЫЙ ВЫЗОВ ПРОЦЕДУР [remote procedure call (RPC)]. Средство, позволяющее вызывать функции и обрабатывать данные, расположенные на другом компьютере, так, как если бы они были расположены локально. У. в. п. обычно применяется в *распределенных приложениях*. У. в. п. существенным образом используется в *распределенной компонентной модели объектов* и *технологии CORBA*

УДАЛЕННЫЙ ДОСТУП [remote access]. *Доступ* к программам и данным, осуществляемый с терминала, удаленного на значительное расстояние и соединенного с *вычислительной системой линией связи*. У. д. производится с помощью *программных средств* У. д. и позволяет, сидя за клавиатурой одного компьютера, входить по линиям связи в удаленную (другую) вычислительную систему. Связь может устанавливаться с компьютером, стоящим в этой же комнате, этом же городе или (например, с помощью сети Интернет) в отдаленном уголке земного шара. Когда связь установлена, вы можете работать так, как будто ваши клавиатура и монитор подключены непосредственно к удаленному компьютеру

УЗЕЛ [node]. 1. То же, что *узел сети*. 2. То же, что *вычислительный узел*

УЗЕЛ СЕТИ, узел [network node, node]. 1. Специальный компьютер с установленным на нем сетевым *программным обеспечением*, предназначенный для диспетчерского управления или коммутации *линий связи* в *компьютерной сети* или *сети передачи данных*. 2. То же, что *вычислительный узел*

УКАЗАТЕЛЬ [indicator, pointer]. 1. То же, что ссылка. Если в *структуре данных* поле некоторого элемента А содержит адрес элемента В, то говорят, что в А содержится У. на В, или А указывает на В. 2. Переменная в программе, значениями которой являются адреса. В некоторых *языках программирования* (например, в Си) имеется *тип данных* У. Значениями переменных типа У. служат адреса переменных, массивов и т. п. *объектов программы*. С У. связаны операции получения адреса и операция косвенной адресации. В Си операция получения адреса обозначается символом `&`. Если за этим символом стоит *имя переменной*, результатом такой операции является адрес указанной переменной. Например, `ptr = &summ;` присваивает У. `ptr` адрес переменной `summ`. Операция косвенной адресации обозначается символом `*`. Если за этим знаком следует У. на переменную, то результатом операции будет величина, помещенная в память по адресу, содержащемуся в У. Например, если после оператора `ptr = &summ;` поместить оператор `x = *ptr;`, то переменная `x` получит то же значение, что и переменная `summ`. Существуют и другие операции над У., аналогичные *арифметическим операциям*. У. открывают большие возможности при программировании *передачи данных* в подпрограммы, работе с многомерными массивами и т. п.

УКАЗАТЕЛЬ ВЫЗЫВАЮЩЕГО НОМЕРА [caller number delivery(CND)]. Сервисная услуга телефонных компаний США и некоторых других стран, с помощью которой телефонные станции вместе с вызывным сигналом передают информацию о вызывающем абоненте. Основной частью этой информации является телефонный номер звонящего абонента или его идентификатор. Некоторые модемы могут принимать эту информацию для последующего анализа программным обеспечением и выводить на экран терминала. Отличие от российских АОНов состоит в том, что информация присылается в другом формате, автоматически и до того, как модем ответит на вызов

УКАЗАТЕЛЬ МЫШИ [mouse cursor]. Значок на экране дисплея, передвигающийся при движении *мыши* по плоскости. В *графическом режиме* работы экрана У. м. обычно представляет собой яркую, ограниченную четким контуром стрелку. В *текстовом режиме* У. м. имеет вид яркого прямоугольника, подсвечивающего одно знакоместо. В приложениях Windows для *персональных компьютеров* вид У. м. может изменяться в зависимости от режима работы и выполняемых функций. В табл. У.1 приведены разновидности У. м. и его функции, принятые в приложениях Windows. См. *мышь*

Таблица У.1. *Вид указателя мыши и его функции, принятые в приложениях Windows*

Вид указателя	Режим работы или выполняемая функция
	Основной режим (выбор команд)
	Предоставление справочной информации об элементе экрана
	Фоновый режим (выполняется какая-то операция в фоновом режиме, пользователь не должен ждать ее завершения)
	Система недоступна (выполняется какая-то операция и пользователь должен ждать ее завершения)
	Режим графического рисования
	Режим выделения текста
	Операция невозможна
	Изменение вертикального размера
	Изменение горизонтального размера
	Изменение размера по диагонали

Таблица У.1 *(окончание)*

Вид указателя	Режим работы или выполняемая функция
	Изменение размера по диагонали
	Перемещение объекта
	Сдвиг вертикальной границы
	Сдвиг горизонтальной границы
	Переход по гиперссылке

УКАЗАТЕЛЬНОЕ УСТРОЙСТВО [pointing device]. *Устройство ввода* координатной информации. Различаются относительные и абсолютные У. у. Например, относительным У. у. является *мышь*, поскольку положение курсора на экране связано с ее перемещением относительно исходного положения. Если перенести мышь в другое место на столе, то положение курсора не изменится. *Сенсорный экран* является примером абсолютного У. у., поскольку прикосновение указки к экрану определяет абсолютные координаты точки на экране. См. *джойстик, трекбол, графический планшет*

УЛЕЙ [hive]. Часть *реестра*, хранящаяся в виде файла на *жестком диске*. Дерево реестра подразделяется на У. (названные так по аналогии с сотовой структурой пчелиного улья). У. представляет собой множество ключей и значений, берущее начало в вершине иерархии реестра

УНАРНАЯ ОПЕРАЦИЯ [unary operation]. То же, что *одноместная операция*

УНАСЛЕДОВАННЫЙ [legacy]. Термин, которым обозначают программное и аппаратное обеспечение, разработанное ранее, но по-прежнему используемое, причем требуется обеспечить совместимость с вновь устанавливаемым программным или аппаратным обеспечением. В паре "новое — прежнее" термин "У." применяют к прежнему программному или аппаратному обеспечению в том случае, когда по тем или иным причинам оно не может быть изменено и должно использоваться "как есть", а совместимость достигается за счет приспособления нового программного и аппаратного обеспечения

УНИВЕРСАЛЬНАЯ ПОСЛЕДОВАТЕЛЬНАЯ ШИНА, стандарт USB [universal serial bus (USB)]. Разработанный корпорацией Intel стандарт обмена данными по недорогой шине между компьютером и *периферийными устройствами*, такими как клавиатура, монитор, *указательные устройства*, *внешние запоминающие устройства* и т. п. С. USB поддерживает скорость *передачи*

данных до 12 Мбайт/с. При этом периферийные устройства соединяются посредством порта, кабелей и соединителей, имеющих единый С. USB, что обеспечивает быструю их замену

УНИВЕРСАЛЬНЫЙ АСИНХРОННЫЙ ПРИЕМОПЕРЕДАТЧИК (УАПП) [universal asynchronous receiver-transmitter (UART)]. *Микросхема*, преобразующая данные, поступающие по параллельным *линиям связи,* в данные, передаваемые последовательно, и наоборот. УАПП содержит цепи приема и передачи, требуемые для *асинхронной связи*. Два компьютера, каждый из которых имеет УАПП, могут связываться простым проводным соединением. УАПП — самая распространенная схема, используемая в модемах *персональных компьютеров*

УНИВЕРСАЛЬНЫЙ ДРАЙВЕР ПРИНТЕРА [universal printer driver, unidriver]. *Драйвер принтера*, способный работать с большинством типов принтеров. Каждый поставщик печатающего устройства предоставляет для своего устройства специальный файл — т. н. мини-драйвер, работающий совместно с У. д. п. для обеспечения взаимодействия с поставляемым устройством. У. д. п. иногда называется растровым драйвером, поскольку он поддерживает печать растровой графики

УНИВЕРСАЛЬНЫЙ ЛОКАТОР РЕСУРСА [Uniform Resource Locator (URL)]. То же, что *универсальный указатель ресурса*

УНИВЕРСАЛЬНЫЙ УКАЗАТЕЛЬ РЕСУРСА, универсальный локатор ресурса [Uniform Resource Locator (URL)]. Строка символов, обозначающая документ или ресурс, запрашиваемый пользователем *Всемирной паутины.* Чтобы получить доступ к нужному ресурсу, достаточно ввести У. у. р. в *поле ввода программы просмотра* WWW. Форматы У. у. р. стандартизованы, поэтому они распознаются всеми программами просмотра и другими *инструментальными программными средствами* системы WWW. Структура У. у. р. имеет следующий вид:

<Тип ресурса или **протокол>://<Имя в Интернете>/<Путь доступа>**,

где **<Тип ресурса** или **протокол>** определяет метод доступа к ресурсу, **<Имя в Интернете>** представляет собой сетевое имя компьютера, на котором находятся искомые данные (см. *доменная система имен*), **<Путь доступа>** может обозначать полное имя файла, порт, с которым необходимо соединиться, либо текст, по которому должен производиться поиск в базе данных. Например, У. у. р. для гипертекстовой страницы информационного агентства CNN: **http://www.cnn.com/cnnsi/basketball/nba/news/1999**

УНИФИЦИРОВАННЫЙ ЯЗЫК МОДЕЛИРОВАНИЯ, язык UML [Unified Modeling Language (UML)]. *Формальный язык визуального моделирования*, предназначенный для проектирования и построения моделей сложных систем, в т. ч. и прежде всего программных систем. Модель в UML отображает-

ся графически в виде совокупности *диаграмм*, описывающих различные аспекты структуры и поведения сложной системы. Будучи наглядными и лаконичными, модели UML являются в то же время строго определенными и даже допускают автоматическую генерацию программного кода. См. *элемент модели*

УПАКОВАННЫЙ ФОРМАТ POSTSCRIPT-ФАЙЛОВ, формат EPS [encapsulated PostScript (EPS)]. Набор команд *языка PostScript*, который можно использовать как независимый объект, представляющий графическое изображение. Изображение в Ф. EPS может быть отправлено по *линии связи* и распечатано на принтере, снабженном интерпретатором PostScript

УПАКОВЫВАТЬ [pack]. Выполнять *сжатие данных*. См. *архивирование*

УПЛОТНЕНИЕ ДАННЫХ [data compaction]. То же, что *сжатие данных*

УПЛОТНЕНИЕ ШРИФТА [condensed font]. Уменьшение интервалов между буквами по сравнению с нормальными интервалами. У. ш. на значительную величину (сравнимую с шириной букв) может использоваться для создания текстовых эффектов от наложения букв. У. ш. на небольшую величину используется при *верстке страниц* для уменьшения общей длины текста, что позволяет вместить его в нужные границы. В настоящее время этот прием форматирования текста используется сравнительно редко. Противоп. *разрядка*. Ср. *кернинг*

УПОРЯДОЧЕНИЕ [sort]. То же, что *сортировка данных*

УПРАВЛЕНИЕ ГОЛОСОВЫМ КАНАЛОМ [media control]. Часть *интерфейса прикладного программирования*, относящаяся к записи, воспроизведению и анализу звука в *голосовом канале*. Обычно противопоставляется *управлению соединениями*, поскольку требует высокоскоростного соединения компьютера с управляемой телефонной системой. К У. г. к. относятся также *анализ* и *синтез речи*. Примером интерфейса прикладного программирования, поддерживающего У. г. к., является *интерфейс TAPI*

УПРАВЛЕНИЕ ДОСТУПОМ [access control]. Методы и алгоритмы ограничения *доступа* к определенным *элементам данных* или средствам на основе *учетных записей пользователей* и их участия в различных предопределенных группах. У. д. обычно используется *системными администраторами* для разграничения доступа пользователей к ресурсам сети, таким как серверы, каталоги и файлы, и реализуется в виде системы разрешений, предоставляемых пользователям и группам для работы с конкретными объектами

УПРАВЛЕНИЕ ДОСТУПОМ К СРЕДЕ, подуровень MAC [media access control (MAC)]. Второй из двух подуровней, на которые делят уровень канала стандарты IEEE 802. Обеспечивает доступ к сети и обнаружение конфликтов

УПРАВЛЕНИЕ КОНФИГУРАЦИЕЙ [configuration management]. Процесс, регламентирующий управление версиями и сопровождение различных артефактов проекта по *разработке программы*

УПРАВЛЕНИЕ ЛОГИЧЕСКОЙ СВЯЗЬЮ, подуровень LLC [logical link control (LLC)]. Высший из двух подуровней, на которые делят уровень канала стандарты IEEE 802. У. л. с. охватывает связи "станция—станция", генерацию кадров и контроль ошибок

УПРАВЛЕНИЕ ПАМЯТЬЮ [memory management]. Учет и распределение между программами имеющейся памяти, а также обеспечение к ней доступа. При этом может подразумеваться управление как всей совокупностью *запоминающих устройств* (ЗУ) *вычислительной системы*, так и отдельными компонентами памяти, например, управление *оперативной памятью*. Целью У. п. является как обеспечение максимальной эффективности обращений к ЗУ каждого типа, так и обеспечение рационального распределения между программами имеющейся памяти. В первом случае примером может служить *кэширование диска*, при котором организуется хранение в оперативной памяти наиболее часто используемых секторов диска. Во втором — организация *виртуальной памяти* со страничным обменом между диском и оперативной памятью. У. п. может осуществляться *аппаратными* и *программными средствами*, а также действиями пользователя. Например, перемещение данных из оперативной памяти в кэш-память выполняется аппаратно, а кэшированием дисков управляет резидентная программа, называемая администратором или диспетчером кэша. Указание на организацию кэширования дисков (если оно предусмотрено в *операционной системе*) дается пользователем. Программы, выполняющие функцию У. п., обычно называются *супервизорами* и входят в состав *операционной системы*. Например, учетом и распределением оперативной памяти между программами управляет *супервизор основной памяти*. См. *распределение памяти, супервизор, виртуальная память*

УПРАВЛЕНИЕ ПРОПУСКНОЙ СПОСОБНОСТЬЮ [bandwidth control]. Определение текущей *пропускной способности* и возможное ее изменение. Так, пропускная способность сервера в сети Интернет может быть сделана меньше его фактической пропускной способности, чтобы зарезервировать часть ресурсов этого компьютера для других целей, например, для *электронной почты*

УПРАВЛЕНИЕ СОЕДИНЕНИЯМИ [call control]. Часть *интерфейса прикладного программирования*, относящаяся к созданию, приему и коммутации *звонков*, а также контролю их параметров и управлению состоянием *линий связи*. Обычно противопоставляется *управлению голосовым каналом*. Примером интерфейса прикладного программирования, поддерживающего У. с., является *интерфейс TAPI*

УПРАВЛЯЮЩАЯ КЛАВИША [control key]. Клавиша <Ctrl>. Клавиша, которая, будучи нажатой в сочетании с другой клавишей, придает последней альтернативный смысл. См. *клавиши быстрого вызова*

УПРАВЛЯЮЩАЯ ПЕРЕМЕННАЯ ЦИКЛА [loop variable]. То же, что *параметр цикла*

УПРАВЛЯЮЩИЙ ОПЕРАТОР [control statement]. Оператор, влияющий на порядок выполнения других операторов. Например, *оператор перехода*, *условный оператор*

УПРАВЛЯЮЩИЙ СИГНАЛ [control signal]. Сигнал, поступающий на объект управления для выполнения определенных действий

УПРАВЛЯЮЩИЙ СИМВОЛ [control character]. *Символ*, который при вводе с клавиатуры или передаче *внешнему устройству* вызывает выполнение вполне определенной функции. Например, если при наборе текста нажимается клавиша <Enter>, то в *текстовый файл* заносится невидимый на экране У. с. (точнее, код У. с.) перевода строки, который при передаче текста на принтер вызовет перемещение бумаги в печатающем устройстве на одну строку вперед. Другой пример: при вводе символа с кодом 07 подается звуковой сигнал, а на экране ничего не отображается. См. *код ASCII*. Ср. *графический символ*

УРОВЕНЬ КАНАЛА, канальный уровень [data link layer]. Второй из семи уровней модели ISO/OSI, применяемой в целях стандартизации межкомпьютерной связи. У. к. расположен непосредственно над физическим уровнем. Он участвует в упаковке информации в кадры, формировании адресов, а также в управлении потоком отдельных передач по линиям связи. Это самый низкий из трех уровней (канала, сетевой, транспортный), занимающихся фактической пересылкой информации из одного устройства в другое. Имеет два подуровня: *управление логической связью* и *управление доступом к среде*

УРОВЕНЬ ПРИЛОЖЕНИЙ, прикладной уровень [application layer]. Уровень сетевых стандартов, который связан с предоставлением сервиса пользователям сети, работающим с приложениями. Это седьмой, самый высший уровень *модели ISO/OSI*. Он базируется на возможностях стандартов более низких уровней, но принимает самое минимальное участие в работе *аппаратного обеспечения* сети. Круг выполняемых на У. п. задач зависит от характера использования сети; это могут быть процедуры регистрации в сети, *электронная почта*, эмуляция терминалов, управление *базами данных* и функционирование *файл-серверов* и *серверов печати*

УСЛОВИЕ ВЫХОДА ИЗ ЦИКЛА, условие завершения цикла [truncation condition]. Условие, в зависимости от выполнения которого происходит или повторение выполнения *тела цикла*, или *выход из цикла*. Если проверка

этого условия осуществляется после выполнения тела цикла, то У. в. и. ц. называют условием окончания цикла, или постусловием. Если же проверка У. в. и. ц. производится перед выполнением тела цикла, то его называют условием продолжения цикла или предусловием. См. *цикл с верхним окончанием, цикл с нижним окончанием*

УСЛОВИЕ ЗАВЕРШЕНИЯ ЦИКЛА [truncation condition]. То же, что *условие выхода из цикла*

УСЛОВНО БЕСПЛАТНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ [Shareware]. Программы, которые доступны в Интернете. Такую программу можно скачать бесплатно и опробовать. Если программа понравилась и принята в эксплуатацию, то пользователь добровольно принимает на себя обязательство послать плату ее автору. Размер платы, имя и адрес автора находятся в специальном файле, распространяемом вместе с самой программой. Ср. *бесплатное программное обеспечение*

УСЛОВНЫЙ ОПЕРАТОР [conditional statement]. *Оператор*, определяющий условие и действия, выполняемые в зависимости от истинности или ложности условия. Таким образом, У. о. реализует выбор вариантов действий при *ветвлении программы*. На *блок-схемах программ* такой оператор изображается символом "Решение" (см. *элементы блок-схем*). Основным элементом У. о. является *синтаксическая конструкция*, состоящая из *ключевого слова* `if` (если), за которым находится проверяемое условие в форме *логического выражения*. Ниже приводятся простейшие формы записи У. о. на языках Фортран, Паскаль и Си. 1) На Фортране: `IF (L) S`, где `L` — логическое выражение, `S` — любой *исполняемый оператор*, кроме *оператора цикла* или другого У. о. Оператор действует следующим образом: если `L` истинно, выполняется оператор `S`; если `L` ложно, то оператор `S` игнорируется (У. о. в этом случае эквивалентен *пустому оператору*). Например, оператор `IF (X.GE.EPS) SUM = SUM + X` указывает: если `X ≥ EPS`, то значения переменных `X` и `SUM` складываются, в противном случае значение переменной `SUM` не изменится. Кроме рассмотренного оператора в языке Фортран существует У. о., в котором в качестве условия используется арифметическое выражение (см. *арифметический* У. о.). 2) На Паскале: `if L then S1 else S2`, где `S1` и `S2` — любые, в том числе составные операторы. Перед словом `else` в языке Паскаль разделитель `;` не ставится. Оператор действует следующим образом: если `L` истинно, выполняется оператор `S1`; если `L` ложно, то — оператор `S2`. Например, У. о. `if x<0 then y:= x*x +1 else y:= 1;` задает вычисление функции

$$y(x) = \begin{cases} x^2 + 1, & \text{если } x < 0\,; \\ 1, & \text{если } x \geq 0\,. \end{cases}$$

У. о. может иметь сокращенную форму: `if L then S1`, в которой отсутствует часть, начинающаяся словом `else`. Эта форма равносильна полной форме У. о. с пустым оператором `S2`. 3) На языке Си: `if (L) S1 else S2` или, в сокращенной форме, `if (L) S1`. Действия, указываемые этими операторами, аналогичны действиям соответствующих операторов языка Паскаль. Например, вычисление приведенной выше функции $y(x)$ на Си программируется У. о. `if (x<0) y = x*x +1; else y = 1;`

УСОВЕРШЕНСТВОВАНИЕ, обновление [upgrade, upgrading]. 1. Замена устройства или программного продукта на его более новую, более мощную версию. Например, У. *вычислительной системы* за счет замены жесткого диска на более быстрый и более емкий или ее У. путем замены операционной системы Windows 95 на Windows 98. 2. Новая версия устройства или программного продукта, обладающая новыми свойствами и новыми возможностями. Например, ОС Windows 98 является У. ОС Windows 95

УСТАНОВКА [setup]. То же, что *инсталляция*

УСТРОЙСТВА SCSI (читается "устройства скази") **[SCSI devices]**. *Периферийные устройства*, связь компьютера с которыми определяется стандартом *интерфейса SCSI*. См. *интерфейс малых вычислительных систем*

УСТРОЙСТВО, техническое устройство [device, unit]. Элемент *аппаратных средств*, представляющий законченную техническую конструкцию, имеющую определенное функциональное значение. Например, принтер, дисковод. См. *устройство ввода, устройство вывода, устройство сопряжения*

УСТРОЙСТВО ВВОДА [input device]. Любое *техническое устройство*, позволяющее осуществлять *ввод данных* в компьютер. Наиболее удобными У. в. являются видеотерминалы и устройства речевого ввода данных, с которых данные вводятся в компьютер без предварительной подготовки, а также У. в. графической информации (сканеры, дигитайзеры, *цифровые планшеты графического ввода*). У. в. позволяет преобразовать данные, записанные на *носителе данных* или поступающие через микрофон, видеокамеру либо по каналам телефонной связи, в форму *цифрового сигнала*, пригодную для записи в память. К У. в. относятся и такие *внешние запоминающие устройства*, как дисководы *магнитных* и *лазерных дисков* (CD), стримеры, У. в. с магнитных карт и уходящие в прошлое У. в. с перфокарт

УСТРОЙСТВО ВВОДА/ВЫВОДА [input-output device]. *Устройство*, обеспечивающее *обмен данными* между *оперативной памятью* и *периферийным устройством*. Этот термин употребляется также в собирательном смысле для обозначения любого из *устройств ввода* или *вывода*

УСТРОЙСТВО ВЫВОДА [output device]. *Техническое устройство*, позволяющее осуществлять вывод из компьютера результатов расчетов или *обработки данных*, программ и другой информации, хранившейся в ЭВМ. У. в.

дает возможность преобразовать поступающие из *оперативной памяти* данные в *цифровой форме* в форму, удобную для восприятия человеком (текст, графические изображения, звук). К таким У. в. относятся дисплеи, принтеры, графопостроители, синтезаторы звука и речи. В качестве У. в. могут выступать и *внешние запоминающие устройства*: дисководы *магнитных дисков*, стримеры, устройства записи на *магнитные карты* и уходящие в прошлое У. в. на перфокарты

УСТРОЙСТВО ПОСТРОЧНОЙ ПЕЧАТИ [line printer (LPT)]. То же, что *линейный принтер*

УСТРОЙСТВО СОПРЯЖЕНИЯ [interface device]. *Устройство*, обеспечивающее интерфейс двух несовместимых *компонентов аппаратных средств*. Например, чтобы использовать телефонную сеть для *передачи данных* от компьютера к компьютеру, нельзя просто соединить выходной порт компьютера проводами с телефонной розеткой. Ведь выводимые из компьютера данные имеют форму дискретного *цифрового сигнала*, а большинство устройств телефонной сети пока работают с непрерывными *аналоговыми сигналами* и не способны передавать цифровой сигнал. Эта несовместимость преодолевается с помощью У. с., называемого *модемом*. См. также *адаптер, сетевая плата*

УСТРОЙСТВО УПРАВЛЕНИЯ [control unit (CU)]. Функциональная часть *центрального процессора* ЭВМ, управляющая работой всех остальных устройств и частей, а также потоками информации внутри *вычислительной машины*. Действия У. у. определяются командами, из которых состоит программа. Команды одна за другой выбираются из *оперативной памяти* в *регистр команд*. У. у. распознает содержание команды и посылает соответствующие управляющие сигналы другим устройствам компьютера, обеспечивая ее выполнение. Например, если команда требует обращения к памяти, то по сигналам У. у. будут произведены все необходимые операции по определению *физического адреса* требуемого данного, считыванию данного из памяти и записи в соответствующий регистр *арифметико-логического устройства*. См. *функциональная схема ЭВМ*

УСТРОЙСТВО УПРАВЛЕНИЯ КОНФЕРЕНЦИЯМИ [multipoint control unit (MCU)]. Узел *сети стандарта H.323*, отвечающий за организацию *конференций*

УТИЛИТА [utility]. То же, что *обслуживающая программа*

УТИЛИТА PING [packet Internet groper (PING)]. То же, что *пакетная программа отладки сетей*

УТОЧНЕНИЕ [refinement]. Более полная и детальная спецификация того, что уже было описано на менее детальном уровне

УЧЕТНАЯ ЗАПИСЬ [computer account]. То же, что *учетная запись компьютера*

УЧЕТНАЯ ЗАПИСЬ КОМПЬЮТЕРА, бюджет компьютера, учетная запись, бюджет [computer account]. Запись в базе данных *локальной вычислительной сети*, в которой регистрируются компьютер и его активность. Каждый компьютер, работающий под управлением *операционной системы Windows NT* и являющийся членом домена, имеет собственную У. з. к., которая создается, когда компьютер впервые идентифицируется в домене в процессе установки сети. Термин "бюджет" обычно относится к содержанию У. з. к.

УЧЕТНАЯ ЗАПИСЬ ПОЛЬЗОВАТЕЛЯ, бюджет пользователя [account, user account]. Совокупность данных, которая определяет пользователя в *операционной системе*. Включает в себя имя и *пароль* пользователя, необходимые для его *регистрации*, группы, к которым принадлежит пользователь, *права доступа* к ресурсам, предоставляемые пользователю для работы с системой и другую информацию. У. з. п. ведет, как правило, *системный администратор*

Ф

ФАЙЛ [file]. Информация, размещаемая на *внешних запоминающих устройствах*, снабженная идентификатором и оформленная как единое целое средствами *операционной системы* или *языка программирования*. Содержимым Ф. могут быть программы, данные, тексты и любая другая информация. По способу доступа к данным различают *файлы последовательного доступа* и *файлы прямого доступа*. В *персональных компьютерах* Ф. хранятся на *магнитных дисках* и рассматриваются в процессах *ввода* или *вывода* данных как единое целое. Работа с персональным компьютером — это работа с файлами. См. *файловая система, файловая система персональных компьютеров*

ФАЙЛ АВТОЗАПУСКА [autoexec file]. *Командный файл*, автоматически выполняемый при *загрузке операционной системы*. Его содержимое заранее программируется пользователем или оператором *вычислительной системы*. Так как Ф. а. выполняется при каждом запуске компьютера, то в него обычно записывают *команды операционной системы*, позволяющие окончательно подготовить вычислительную систему к работе, приспособив ее к нуждам пользователя. Например, в Ф. а. *операционной системы MS-DOS* autoexec.bat удобно включить: 1) команды, устанавливающие список каталогов, в которых производится поиск программ в случае, если пользователь введет имя команды без указания, в каком каталоге находится соответствующая программа; 2) команду, определяющую вид приглашения *операционной системы*; 3) команды запуска драйвера-русификатора, *оболочки Norton Commander*, программы защиты от *компьютерного вируса* и других программ, создающих привычную рабочую обстановку на компьютере. См. *загрузка операционной системы*

ФАЙЛ КОНФИГУРАЦИИ [configuration file]. *Системный файл*, задающий изменения стандартных параметров при *генерации операционной системы*. В *операционной системе MS-DOS* имеет стандартное имя config.sys. По указаниям Ф. к. в процессе *загрузки операционной системы* создается ее рабочая версия, которая по составу и структуре соответствует *конфигурации компьютера* и адаптирована к потребностям пользователя. Например, устанавливается необходимое количество буферов для операций ввода/вывода с диском, максимальное количество одновременно *открытых файлов* и т. п. В Ф. к. указывается также, какие программы, расширяющие возможности операционной системы, необходимо загрузить в оперативную память. Обычно ими являются драйверы устройств, подключенных к компьютеру. Ф. к., как правило, является *текстовым файлом*. Его содержимое заранее программируется пользователем или оператором *вычислительной системы* и может корректироваться перед каждой перезагрузкой. См. *загрузка операционной системы, генерация операционной системы*

ФАЙЛОВАЯ СИСТЕМА [file system]. 1. Система организации файлов и каталогов. Включает в себя структуру каталогов и файлов, а также правила их создания и манипулирования ими. 2. Часть *операционной системы*, обеспечивающая выполнение операций над файлами

ФАЙЛОВАЯ СИСТЕМА ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ [IBM-PC file system]. В *персональных компьютерах* файлы хранятся на *магнитных дисках* и рассматриваются в процессах *ввода* или *вывода данных* как единое целое. Файл имеет имя и расширение, разделенные символом "." (точка). Файл характеризуется размером, датой создания и другими атрибутами. Расширение дает возможность пользователю и программам распознавать характер информации, содержащейся в файле. Сгруппировать отдельные виды файлов и отделить одну группу файлов от другой можно с помощью каталогов. На каждом *магнитном* или *лазерном диске* имеется один *корневой каталог*. В нем регистрируются файлы и подкаталоги (каталоги 1-го уровня). В каталогах 1-го уровня регистрируются файлы и каталоги 2-го уровня и т. д. Получается многоуровневая древовидная структура каталогов на диске. См. *каталог, дерево каталогов*. Различные операционные системы используют разные файловые системы для управления файлами. См. *файловая система FAT, файловая система HPFS, файловая система NTFS*

ФАЙЛОВАЯ СИСТЕМА FAT [file allocation table file system, FAT file system]. *Файловая система*, существующая в *операционной системе MS-DOS* для хранения информации на дисках. Ф. с. FAT основана на использовании *таблиц размещения файлов* (FAT, File Allocation Table). Ф. с. FAT представляет собой структуру данных, создаваемую на диске при форматировании. Во время сохранения файла на диске MS-DOS помещает в Ф. с. FAT информацию, по которой этот файл можно найти и прочитать. В Ф. с. FAT для имен файлов применяется так называемый формат 8.3 (до восьми символов в имени с

последующей точкой и трехсимвольным расширением). *Операционные системы Windows*, имеющие собственные файловые системы, тем не менее совместимы с Ф. с. FAT

ФАЙЛОВАЯ СИСТЕМА HPFS [high-performance file system (HPFS)]. 1. *Файловая система операционной системы OS/2.* 2. Метод разметки *жесткого диска*, используемый *операционной системой* OS/2. (Операционная система Windows NT версий 3.1—3.51 также поддерживает HPFS.) Хотя данная файловая система позволяет значительно увеличить скорость выполнения файловых операций, только операционная система OS/2 и приложения, разработанные для этой системы, могут обращаться к дискам, размеченным данным способом

ФАЙЛОВАЯ СИСТЕМА NTFS [Windows NT file system (NTFS)]. То же, что *файловая система Windows NT*

ФАЙЛОВАЯ СИСТЕМА WINDOWS NT, файловая система NTFS [NT file system, Windows NT file system (NTFS)]. 1. *Файловая система операционной системы Windows NT.* Поддерживает объектно-ориентированные приложения, представляя все файлы как объекты, имеющие атрибуты, определяемые пользователями системы. Обладает возможностями восстановления. Обеспечивает *защиту данных*, надежность файловой системы и все возможности других файловых систем *персональных компьютеров.* 2. Метод разметки *жесткого диска*, используемый *операционной системой Windows NT.* Хотя этот метод позволяет значительно увеличить скорость выполнения файловых операций, только операционная система Windows NT и приложения, разработанные для этой системы, могут обращаться к дискам, размеченным данным способом

ФАЙЛОВЫЙ СЕРВЕР, файл-сервер [file server]. Компьютер, содержащий файлы для общего пользования, предоставляющий доступ к ним и обеспечивающий централизованное управление файлами в сети. В традиционной сети *персональных компьютеров* центральный Ф. с. позволяет совместно использовать файлы, посылая их на сервер. После этого можно либо переслать файлы на другой компьютер, либо сохранить их на сетевом диске. Традиционные приложения типа *"клиент-сервер"* имеют одну из двух стандартных архитектур: либо приложение целиком выполняется на сервере, либо целиком выполняется на *рабочей станции*, получая только данные с сервера. См. *архитектура "клиент-сервер", одноранговые вычисления*

ФАЙЛ ПОДКАЧКИ, файл замещения страниц [swapping file, swap file, paging file]. *Файл* на диске, в который осуществляется *свопинг.* В Ф. п. размещаются активные страницы памяти *операционной системы* или приложения, для которых в данный момент не оказалось места в *оперативной памяти*

ФАЙЛ ПОСЛЕДОВАТЕЛЬНОГО ДОСТУПА, последовательный файл [sequential file]. *Файл*, записи которого можно читать только последовательно, друг за другом, в порядке их расположения. В Ф. п. д. определены начальная и конечная записи, а относительно любой другой записи введены понятия "текущая", "предыдущая" и "последующая" записи. Ф. п. д. являются, например, файлы, хранящиеся на *магнитных лентах*. Ср. *файл прямого доступа*

ФАЙЛ ПРЯМОГО ДОСТУПА, прямой файл [direct-access file]. *Файл*, записи которого можно читать в произвольном порядке. В нем каждая запись снабжена своим номером — ключом, по которому и осуществляется поиск нужной записи. Ф. п. д. организуются на *магнитных* и *лазерных дисках*. Ср. *файл последовательного доступа*

ФАЙЛ ПРОТОКОЛА [log file]. Файл, в который записывается информация о ходе выполнения программы. Например, в Ф. п. можно записывать, когда и какие модули программы выполнялись, как изменялись значения определенных переменных и т. д. В случае сбоя программы можно будет проанализировать Ф. п. и установить причину сбоя

ФАЙЛ РЕЕСТРА [REG file]. *Файл*, предназначенный для хранения разделов и значений *реестра Windows* в текстовом представлении. Ф. р. используются для импорта и экспорта данных реестра Windows

ФАЙЛ-СЕРВЕР [file server]. То же, что *файловый сервер*

ФАЙЛ СПУЛИНГА, спул-файл [spool file]. *Файл*, в который в процессе *спулинга* записывается содержимое *задания на печать*. В случае отключения питания в процессе печати наличие Ф. с. предотвращает потерю данных, т. к. выполнение задания на печать может быть возобновлено после включения питания. Ср. *теневой файл*

ФАЙЛ УСТАНОВКИ [INF file]. Специальный файл, в котором хранится информация, необходимая для установки приложения, драйвера или другого компонента. Например, в этом файле указываются папки, в которые должны быть скопированы файлы приложения. Кроме того, в Ф. у. содержатся данные, используемые для регистрации устанавливаемого компонента

ФАЙЛ AVI [AVI-file]. *Файл*, в котором в стандартном *формате AVI* хранится информация о форме звукового и видеосигнала. Имя Ф. AVI имеет расширение avi

ФАЙЛ BMP [BMP file]. *Растровый файл*, выполненный в стандартном *формате BMP*. Имя Ф. BMP имеет расширение bmp

ФАЙЛ PCX [PCX-file]. *Растровый файл*, выполненный в стандартном *формате PCX*. Имя Ф. PCX имеет расширение pcx

ФАЙЛ RTF [RTF file]. *Файл*, выполненный в *формате RTF*. Имя Ф. RTF имеет расширение rtf

ФАЙЛ TIFF [TIFF file]. *Растровый файл*, выполненный в стандартном *формате TIFF*. Имя Ф. TIFF имеет расширение tif

ФАКС [fax]. 1. *Факсимильное сообщение*, передаваемое с помощью *факс-модема* или *факс-аппарата* через телефонное соединение. Обычно в виде Ф. передаются точные копии документов, схем, чертежей, фотографий. 2. То же, что *факс-аппарат*

ФАКСИМИЛЬНОЕ СООБЩЕНИЕ [fax message]. Сообщение, представляющее графически точно воспроизведение рукописи, документа, подписи и т. п. Происходит от латинского *fac simile* — делай подобное

ФАКС-АППАРАТ, факс-машина, факс [fax machine]. *Устройство*, предназначенное для приема и передачи *факсимильных сообщений (факсов)* через телефонное соединение. Ф.-а. включает в себя встроенные сканер и принтер, а также электронные преобразователи цифровых сигналов в аналоговые и обратно. Обычно Ф.-а. объединяется с телефонным аппаратом. Для передачи и приема факсов может использоваться снабженный *факс-модемом* компьютер

ФАКС-МАШИНА [fax machine]. То же, что *факс-аппарат*

ФАКС-МОДЕМ [fax-modem]. *Модем*, снабженный устройством для приема и передачи *графической информации* и *факсимильных сообщений*

ФАКТИЧЕСКИЙ ПАРАМЕТР [actual parameter]. Выражение или имя, задающее значение, объект программы или другую процедуру при *обращении к процедуре*, функции или макрокоманде. Все сказанное ниже о Ф. п. процедуры относится и к Ф. п. функции. Ф. п. определяют реальные *исходные данные*, над которыми производятся вычисления или другие действия, определяемые процедурой, а также указывают переменные, массивы и другие объекты *вызывающей программы*, которым должны быть присвоены результаты выполнения процедуры. Кроме того, Ф. п. может быть имя процедуры (функции), алгоритм которой должен быть использован в данной процедуре. Ф. п. указываются в *списке* Ф. п. *оператора вызова процедуры* или *вызова функции* в том же порядке, в котором расположены соответствующие *формальные параметры* в *списке формальных параметров описания процедуры* или *описания функции*. Этим обеспечивается необходимое соответствие между описанием процедуры и ее вызовом. Помимо этого, каждый Ф. п. должен соответствовать своему формальному параметру по классу. В табл. Ф.1 показано правильное соответствие классов фактических и формальных параметров в языке Паскаль. Ср. *формальный параметр*

Таблица Ф.1. *Соответствие классов фактических и формальных параметров в языке Паскаль*

Класс параметров	Формальный параметр	Фактический параметр
Параметр-значение	Имя переменной	Выражение
Параметр-переменная	Имя переменной	Переменная
Параметр-процедура	Заголовок процедуры	Имя процедуры
Параметр-функция	Заголовок функции	Имя функции

ФАТАЛЬНАЯ ОШИБКА [fatal error]. См. *неисправимая ошибка*

ФИГУРНЫЙ ТЕКСТ [artistic text, word art]. То же, что *заголовочный текст*

ФИЗИЧЕСКАЯ ЗАПИСЬ [physical record]. *Блок данных*, в котором размещаются одна или несколько *логических записей*

ФИЗИЧЕСКАЯ ПАМЯТЬ [physical storage]. *Оперативная* и *внешняя память* ЭВМ

ФИЗИЧЕСКИЙ АДРЕС [physical address]. 1. То же, что *абсолютный адрес*. 2. Уникальный идентификатор, однозначно определяющий конкретное *внешнее устройство*

ФЛАЖОК [checkbox]. *Элемент управления*, предназначенный для выбора одного из двух возможных значений. На рис. Э.2 приведен пример использования Ф. Ср. *переключатель*

ФЛОПС [floating-point operations per second (FLOPS)]. Единица измерения вычислительной *производительности компьютера*, равная одной *арифметической операции* с плавающей точкой в секунду

ФЛЭШ-ПАМЯТЬ [flash memory]. Запоминающее устройство, подобное *микросхемам памяти*, но являющееся энергонезависимым. Применяется в качестве *внешнего запоминающего* устройства в *персональных компьютерах*, а также в качестве оперативной памяти в различных цифровых устройствах, например, *цифровых фотоаппаратах*. Емкость Ф.-п. достигает нескольких Гбайт, однако скорость чтения/записи пока уступает обычным микросхемам памяти. На рис. Ф.1. показано устройство Ф.-п. для *персонального компьютера*, подсоединяемое к порту USB. См. *универсальная последовательная шина*

ФОКУС [focus]. Говорят, что *диалоговое окно* или *элемент управления* имеет Ф., если этот элемент активен и действия пользователя (ввод с клавиатуры, *щелчок* кнопкой мыши) относятся именно к данному элементу и должны быть им восприняты и обработаны. Например, в *текстовом редакторе* MS Word Ф. обычно имеет *окно документа,* и нажатие клавиш приводит к вводу символов в текст. Однако если передать Ф. *строке меню* (путем нажа-

тия клавиши <Alt>), то нажатие клавиш приведет к выполнению команды меню, но не к вводу символа в текст. См. *модальное диалоговое окно*

Рис. Ф.1. Флэш-память

ФОКУС УПРАВЛЕНИЯ [focus of control flow]. То же, что *активация*

ФОНД СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ [Free Software Foundation (FSF)]. Организация, деятельность которой направлена на устранение ограничений на копирование, распространение, анализ и модификацию компьютерных программ путем стимулирования разработки и использования свободного программного обеспечения. Под свободными понимаются программы, которые можно беспрепятственно копировать и изменять по собственному желанию при абсолютном доступе к их первоначальному тексту. Основателем и руководителем Ф. с. п. о. является Ричард Столлман, который еще в 1985 г. выдвинул идеи неприятия права собственности на программы. Эти идеи привели к *проекту GNU* — основному детищу фонда. См. *бесплатное программное обеспечение*, *операционная система Linux*

ФОННЕЙМАНОВСКАЯ АРХИТЕКТУРА [von Neumann architecture]. То же, что *архитектура фон Неймана*

ФОНОВАЯ ЗАДАЧА [background task]. *Задача* с низким приоритетом, выполняемая *вычислительной системой* в режиме мультипрограммирования в те моменты времени, когда выполнение высокоприоритетных задач по тем или иным причинам невозможно. В качестве Ф. з. выбирают задачи, не требующие диалога с пользователем или предназначенные для *пакетной обработки*. Выполнение Ф. з. позволяет снизить непроизводительные затраты *машинного времени* и эффективно использовать оборудование

ФОРМА, формуляр [form]. 1. Документ строго определенного формата (электронный бланк), который подлежит заполнению по определенным правилам. Часто применяется в делопроизводстве. 2. Набор *элементов управ-*

ления, организованный приложением в виде *диалогового окна*, для взаимодействия приложения с пользователем. Ф. используется как один из основных приемов организации *интерфейса пользователя* в самых разнообразных случаях: для взаимодействия с веб-страницей, для ввода информации в базу данных и т. д. Иногда размеры Ф. можно менять, что отличает ее от диалогового окна

ФОРМАЛИЗОВАННЫЙ ЯЗЫК [formalized language]. Язык, полностью или частично определенный по некоторым формальным правилам. К Ф. я. относятся искусственно созданные *алгоритмические языки* и *языки программирования*, основанные на точном предварительном описании алфавита, синтаксиса и семантики. Ф. я. противопоставляется *естественный язык*

ФОРМАЛЬНЫЙ ПАРАМЕТР [formal parameter]. Условное имя, определенное в процедуре или функции для обозначения аргумента процедуры (функции) или результата ее работы. Все сказанное ниже о Ф. п. процедуры относится и к Ф. п. функции. В *заголовке процедуры* Ф. п. определяет тип *элемента данных*, который должен быть передан процедуре при ее вызове. В *теле процедуры* операторы задают алгоритм как действия над Ф. п., что обеспечивает возможность подстановки различных *исходных данных* путем замены Ф. п. на *фактические параметры*, указанные в *операторе вызова процедуры*. Например, в процедуре решения квадратного уравнения $ax^2 + bx + c = 0$ (см. *процедура*, *заголовок процедуры*) Ф. п.-аргументами являются a, b и c, обозначающие коэффициенты уравнения, действия над которыми определяют решение задачи. Существуют четыре класса Ф. п., допустимые в большинстве традиционных *языков программирования*: параметры-значения, параметры-переменные, параметры-процедуры и параметры-функции. Ф. п.-значение представляет в вызываемой процедуре некоторую *локальную переменную*. Этой переменной в качестве начального значения при вызове процедуры присваивается значение соответствующего фактического параметра, который должен быть выражением. Параметры-значения нельзя использовать для представления результата выполнения процедуры. Ф. п.-переменная при вызове замещается фактической переменной *вызывающей программы*. В этом случае любые операторы, в записи которых участвует Ф. п., выполняются непосредственно над фактическим параметром. Параметры-переменные используются также для представления результатов выполнения процедуры. В *списке* Ф. п. параметры-процедуры и параметры-функции, как правило (например, в языке Паскаль), представляются своими заголовками. При *вызове процедуры* в качестве фактического параметра передается имя определенной процедуры или функции. Ср. *фактический параметр*. См. *вызов процедуры, передача параметров*

ФОРМАЛЬНЫЙ ЯЗЫК БЭКУСА—НАУРА, нормальная форма Бэкуса—Наура (БНФ) [Backus-Naur form (BNF)]. Формальная система обозначений для описания *синтаксиса языков программирования*. Разработана Дж. Бэкусом для

описания синтаксиса языка Алгол-60. Ее варианты используются до сих пор. В БНФ одни синтаксические категории определяются через другие, последовательно. Например,

```
<цифра>:: = 0 |1 |2 |3 |4 |5 |6 |7 |8 |9
<целое число без знака>:: = <цифра> |<цифра> <целое число без знака>
<знак>:: = + |-
<целое число>::= <целое число без знака> |<знак> <целое число без знака>
```

где имя определяемой синтаксической категории стоит слева от символа `::=`, который означает "определяется как". Справа стоит определение. Символ | читается как "или"

ФОРМАТ [format]. 1. Способ представления или схема расположения данных в памяти компьютера либо на других *носителях данных*. Ф. определяет количество и порядок размещения кодов, символов или *элементов данных* в *машинном слове*, записи, файле и других структурах данных в *оперативной памяти* и на внешних носителях данных. Ф. может определяться как конструктивными особенностями носителя, так и характером данных. Например, Ф., в котором хранятся *вещественные числа* в *основной памяти* IBM PC, строго определен (см. *представление чисел с плавающей точкой*), а Ф. печати конкретных вещественных чисел задается программистом средствами *языка программирования*. На экран или бумагу их можно вывести либо в форме *числа с фиксированной точкой*, либо в форме *числа с плавающей точкой*. При этом можно устанавливать длину числа, наличие или отсутствие знака, если число положительное, и т. п. 2. *Синтаксическая конструкция* языка программирования, символически описывающая Ф. данных (представление или расположение данных в файле) при вводе или выводе. Например, вывод результата перевода секунд в минуты и секунды в языке Си может быть задан оператором

```
printf("%d сек. — это %d мин. %d сек. \n", sec, min, add);
```

в котором синтаксическая конструкция `"%d сек. — это %d мин. %d сек. \n"` задает Ф. выводимой записи. При значениях переменных `sec = 235`, `min = 3`, `add = 55` на дисплей или в файл будет выведена следующая строка:

```
"235 сек. — это 3 мин. 55 сек."
```

В языке Фортран для установления нужного Ф. ввода/вывода служит *оператор задания формата*. 3. Форма, внешний вид документа или отдельных его частей. Ф. характеризуется совокупностью параметров. Например, Ф. абзаца определяется видом и размером шрифта, величиной межстрочного интервала, способом выравнивания текста и т. д. В *текстовых редакторах* эти характеристики задаются установкой значений соответствующих параметров. См. *форматирование текста, формат бумаги*

ФОРМАТ БУМАГИ [paper size]. Используемый принтером стандартный размер листа бумаги. Согласно международному *стандарту ISO* лист формата А0 имеет размер (840×1189) мм^2 ≈ 1 м^2. Каждый последующий формат имеет вдвое меньшую площадь и получается из предыдущего уменьшением вдвое его наибольшей стороны. Следовательно, А1 имеет размер (594×840) мм^2 ≈ 1/2 м^2, А2 — (420×594) мм^2 ≈ 1/4 м^2, А3 — (297×420) мм^2 ≈ ≈ 1/8 м^2. Наиболее распространенным в России Ф. б. является А4 — (270×297) мм^2 ≈ 1/16 м^2

ФОРМАТИРОВАНИЕ [formatting]. 1. То же, что *форматирование диска* 2. То же, что *форматирование текста*

ФОРМАТИРОВАНИЕ ДАННЫХ [data formatting]. Преобразование формы и *структуры данных* в соответствии с предписанным *форматом*

ФОРМАТИРОВАНИЕ ДИСКА, форматирование [disk initialization, formatting]. Процедура, состоящая из разметки *магнитного диска* на адресуемые элементы (дорожки и секторы), проверки их качества и создания на нулевой дорожке *системной области*. Ф. д. выполняется как для *жестких*, так и для *гибких магнитных дисков*. Оно необходимо для подготовки нового диска к первому его использованию. При форматировании ранее использованного диска все имевшиеся на нем записи уничтожаются, поэтому применять Ф. д. для жесткого диска без крайней необходимости не рекомендуется. В *персональных компьютерах* Ф. д. выполняется *сервисными программами* или *командами операционных систем*, например, по команде MS-DOS

```
format a: /q /u /s /v:disk_1
```

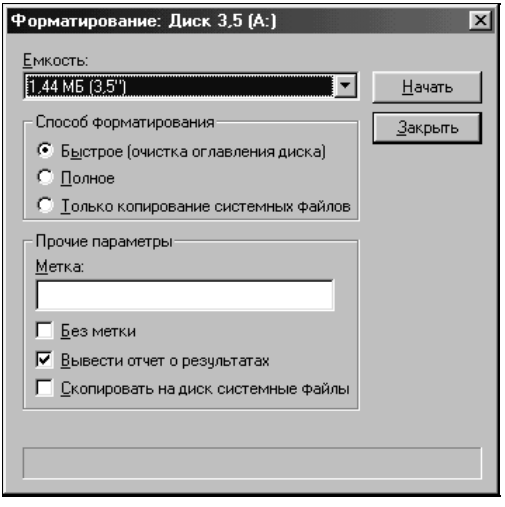

Рис. Ф.2. Диалоговое окно для форматирования диска

осуществляется быстрое, невосстанавливаемое Ф. д., установленного в дисковод А:. При этом дискете назначается метка `disk_1`, и на нее переносятся основные файлы операционной системы. В *операционной системе Windows* Ф. д. выполняется с помощью *диалогового окна*, показанного на рис. Ф.2, которое вызывается, например, командой *контекстного меню* щелчком правой кнопкой мыши по значку дисковода в программе *Проводник Windows*

ФОРМАТИРОВАНИЕ ТЕКСТА [text formatting]. В *текстовых редакторах* — автоматическое придание тексту определенной формы — *формата*. Различают два способа Ф. т.: прямое и стилевое. При прямом форматировании с помощью меню и диалоговых окон пользователь устанавливает шрифт, его размер, размер строки и другие параметры выделенного участка текста. При стилевом форматировании символу, отдельному абзацу или тексту всего документа назначается заранее заготовленный набор параметров, называемый стилем. Пользователь делает такое назначение, указывая лишь имя стиля. Применение стилей упрощает процесс Ф. т. и способствует унификации оформления документов. Текстовые редакторы позволяют изменять существующие стили и создавать новые

ФОРМАТ ОБМЕНА ГРАФИЧЕСКИМИ ДАННЫМИ [graphics interchange format (GIF)]. То же, что *формат GIF*

ФОРМАТ AVI [Audio Video Interleave (AVI)]. Стандартный *формат* цифровой записи и сжатия движущихся видеоизображений с синхронным звуковым сопровождением. Ф. AVI основан на методе чередования аудио- и видеоданных, разработанном фирмой Microsoft. Он позволяет хранить в компьютере и воспроизводить в окне *операционной системы Windows* цветные звуковые фильмы. Для этого не нужна специальная аппаратура — все происходит с использованием только программных средств. В настоящее время Ф. AVI является самым популярным при передаче мультимедиа через Интернет. См. *файл AVI*

ФОРМАТ BMP [format BMP (сокращение от англ. *bitmap*), **(BMP)]**. Стандартный *формат растровой графики* в *операционных системах Windows*, согласно которому в *битовом массиве* на изображение точки отводится 4, 8 и 24 разряда. Этот формат графических данных не подразумевает никакой формы *сжатия данных*. Другие операционные системы могут также использовать указанный формат данных, чтобы поддерживать графику различных типов. См. табл. Г.1, *файл BMP*. Ср. *формат PCX*

ФОРМАТ EPS [encapsulated PostScript (EPS)]. То же, что *упакованный формат PostScript-файлов*

ФОРМАТ GIF, формат обмена графическими данными [graphics interchange format (GIF)]. Формат *графических файлов*, разработанный фирмой CompuServe. Поддерживает 24-разрядную 256-цветную палитру с размером изображения до 64×64 Кбайт пикселов. Один из самых распространенных графических

форматов в Интернете. Наиболее популярен для создания баннеров, т. к. позволяет создавать анимацию, прозрачные области, *чересстрочную загрузку изображения*. Имеет хороший алгоритм сжатия. Существует несколько распространенных версий этого формата, последняя из которых — GIF89a — наиболее популярна в настоящее время. Ф. GIF первоначально использовался CompuServe для уменьшения времени загрузки графических файлов и коррекции возможных ошибок. Другое достоинство Ф. GIF — это поддержка анимированных иллюстраций. При этом в одном графическом файле находится несколько графических изображений, а также служебная информация, управляющая последовательностью смены кадров, их расположением и т. д. Браузер или другое специально разработанное приложение будут отображать графические изображения по одному в том порядке, в котором они находятся внутри файла, чтобы создать эффект анимации

ФОРМАТ JPEG [format JPEG (JPEG)]. *Формат графических файлов*, применяемый для описания, сжатия и хранения цветных изображений с большим количеством цветов. Разработан *группой JPEG*. Формат базируется на сжатии изображения с потерей качества, поэтому может использоваться только для тех изображений, которые не содержат множества мелких, четко очерченных значащих частей. Чаще всего применяется для сохранения фотоизображений. Степень сжатия и, соответственно, потери качества указываются пользователем при создании файла. См. табл. Г.1, ср. *формат BMP*, *формат TIFF*

ФОРМАТ MPEG [format MPEG (MPEG)]. Разработанный *группой MPEG формат* хранения сжатого файла. Допускает сжатие до 1:200. См. *стандарт MPEG*

ФОРМАТ PCX [format PCX (PCX)]. Широко распространенный *формат растровой графики*, применяемый для описания и хранения цветных графических изображений. Ф. PCX поддерживает различную *глубину цвета* и методы *сжатия данных*. Формат разработан компанией Zsoft для *графического редактора* PaintBrush. В настоящее время используется во многих графических редакторах и программах. См. табл. Г.1, *файл PCX*. Ср. *формат BMP*

ФОРМАТ RTF, расширенный текстовый формат [rich text format (RTF)]. Формат файлов, содержащих форматированный текст с включенными в него графическими изображениями. Разработан корпорацией Microsoft для обмена документами между прикладными программами. Реализован на многих компьютерных платформах. См. *файл RTF*

ФОРМАТ TIFF, тегированный формат файлов изображений [tagged image file format, format TIFF, (TIFF)]. *Формат растровых файлов*, используемый для описания, сжатия и хранения цветных и полутоновых графических изображений. Разработан компаниями Aldus и Microsoft. Обычно файлы в Ф. TIFF имеют размер меньше, чем файлы в графических форматах, в которых не применяется сжатие данных. Ф. TIFF часто используется в устройствах по-

лучения изображений, таких как *сканеры* и *цифровые фотоаппараты*. См. табл. Г.1. Ср. *формат BMP, формат JPEG*

ФОРТРАН [Fortran]. *Язык программирования* научно-технических расчетов. Название языка произошло от английских слов FORmula TRANslator (транслятор формул). Запись фортрановских программ напоминает алгебраические формулы, что в значительной мере облегчает научному работнику программирование вычислений. Основной информационной структурой языка является массив, который соответствует матрицам, широко используемым в научных расчетах. В Ф. впервые была реализована важная идея независимой компиляции подпрограмм, что дало возможность создавать библиотеки научных подпрограмм. Первая версия Ф. I была разработана фирмой IBM в 1956 г. В 1958 г. появилась версия Ф. II, которую в свою очередь сменил язык Ф. IV, стандартизованный в 1966 г. Американским национальным институтом стандартов и поэтому называемый также Ф. 66. В 1977 г. был принят новый стандарт языка Ф. 77, в котором появился *символьный тип* данных. Однако этот стандарт, как и все предшествующие, не отвечал требованиям методологии *структурного программирования*. На создание следующего стандарта ушло 13 лет. В Ф. 90 появились *структурированные типы* данных, описываемые оператором `STRUCTURE`, и средства реализации методологии структурного программирования. Ф. остается до настоящего времени широко распространенным языком программирования, особенно среди пользователей, занимающихся вопросами численного моделирования. Это объясняется несколькими причинами: существованием огромных фондов *прикладных программ* на Ф., накопленных за годы, а также наличием значительной армии программистов, использующих этот язык, наличием эффективных трансляторов Ф. на всех типах компьютеров, причем версии для различных типов машин стандартизованы и перенос программ с машины на машину не составляет больших трудностей. Ф. получил широкое распространение и на персональных компьютерах, благодаря различным версиям транслятора Ф. фирмы Microsoft

ФРЕЙМ, рамка [frame]. *Панель экрана*, которая не может перекрывать другие панели и окна. Часто применяется для форматирования *веб-страниц*. Ср. *форма*

ФУНКЦИОНАЛЬНАЯ СХЕМА ЭВМ [computer functional diagram]. Условное изображение структуры ЭВМ, отражающее основное функциональное назначение составных частей компьютера и связи между ними. Обычно на Ф. с. ЭВМ основные функциональные компоненты изображаются в виде геометрических фигур, а связи между ними обозначаются стрелками. ЭВМ состоит из шести основных функциональных частей: *устройства управления* и *арифметико-логического устройства*, *оперативной памяти*, *внешней памяти*, а также *устройств ввода* и *вывода данных*. На рис. Ф.3 схематически изображены эти части и связи между ними. Одиночными стрелками показаны пути прохождения сигналов управления, двойными — потоки обрабатываемой информации

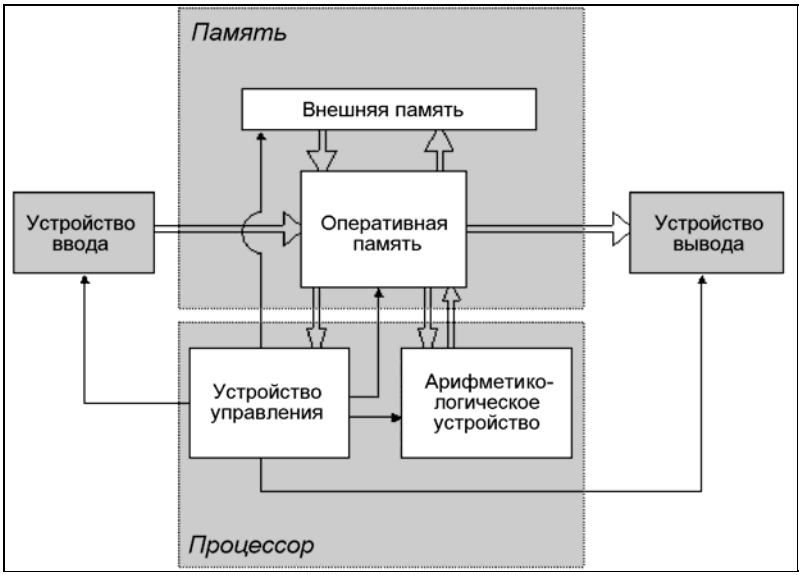

Рис. Ф.3. Функциональная схема ЭВМ

ФУНКЦИОНАЛЬНОЕ УСТРОЙСТВО [functional device, functional unit]. *Устройство* или набор устройств *вычислительной системы*, реализующий одну или несколько функций. К таким функциям относятся, например, исполнение конкретных арифметических или логических операций, ввод, вывод, пересылка данных в память или извлечение данных из нее и т. п. Различают простые Ф. у. и конвейерные. Простое Ф. у. одновременно может реализовать только одну операцию. Простым устройством можно считать многофункциональный процессор, если он не способен начать выполнение последующей операции раньше, чем закончится предыдущая. В отличие от простого Ф. у., конвейерное Ф. у. способно одновременно выполнять несколько операций. Конвейерное Ф. у. часто является цепочкой простых Ф. у., последовательно реализующих отдельные этапы исполнения функции или операции. Например, при сложении чисел с плавающей точкой соответствующие элементарные устройства последовательно реализуют выравнивание порядков, сложение мантисс, нормализацию и т. п. Возможны и более сложные конвейерные Ф. у. См. *конвейерная обработка данных*

ФУНКЦИОНАЛЬНЫЕ КЛАВИШИ [function keys]. Специальные клавиши на клавиатуре, которые могут быть использованы при работе с программой. Действие, вызываемое нажатием Ф. к., может быть заранее запрограммировано так, чтобы инициировать выполнение некоторой операции. Например, на клавиатурах *персональных компьютеров* функциональными являются клавиши <F1>—<F12> (см. рис. К.3). Эти клавиши используются в программах

для различных целей. Так, клавиша <F1>, как правило, вызывает *оперативную помощь* (Help) при работе с программой, <F2> — запись данных в файл (сохранение) и т. д.

ФУНКЦИЯ [function]. 1. Одно из предназначений устройства, программы, системы. Например, Ф. дисплея — вывод на экран текстовой и графической информации. 2. В *языках программирования высокого уровня* (например, в Паскале и Си) аналогичная процедуре конструкция, соответствующая понятию подпрограммы. Представляет собой именованную часть программы (*блок программы* или группу операторов), результатом выполнения которой является значение, присваиваемое имени Ф., поэтому *вызов* Ф. используется в качестве операнда в выражении. Алгоритм Ф. может быть задан в виде действий над условными данными, определяемыми с помощью *формальных параметров*. При вызове Ф. вместо формальных параметров указываются *фактические параметры*, определяющие конкретные данные, над которыми и выполняются запрограммированные Ф. действия. Возможны Ф., в которых нет формальных параметров. В них операторы сразу задают действия над *объектами программы*, определенными в *главной программе*. В некоторых языках программирования (например, в Си) отсутствует понятие *процедуры*, а вызов Ф. может не только употребляться в качестве операнда выражения, но и быть отдельным оператором *вызывающей программы*. При этом значение, которым в результате вызова обладает имя Ф., никуда не передается, а результат выполнения Ф. может состоять, например, в изменении значений некоторых фактических параметров или *глобальных переменных*. Аналогичное применение Ф. допускается в последних версиях Паскаля. Ф. вводятся в программу с помощью *описания* Ф., которое обычно располагается в разделе описаний. Описание Ф. состоит из *заголовка* Ф. и *тела* Ф. Заголовок служит для присвоения Ф. некоторого имени и, возможно, указания формальных параметров. В теле программируется выполняемый Ф. алгоритм. Важную роль в программах играют т. н. *встроенные* или *стандартные* Ф. языка программирования. Они не требуют описаний и автоматически распознаются транслятором. Ср. *процедура*. См. *описание функции, вызов функции, встроенные функции*

ФУНКЦИЯ ОБРАТНОГО ВЫЗОВА [callback function]. Функция, которая используется для организации связи между *вызывающей программой* и *вызываемой подпрограммой* таким образом, чтобы вызывающая программа могла обрабатывать промежуточные результаты работы вызываемой подпрограммы до завершения работы последней. Для этого в вызывающей программе определяется Ф. о. в, которая передается в качестве одного из параметров вызываемой подпрограмме. Подпрограмма в процессе своей работы вызывает переданную ей Ф. о. в., которая выполняет необходимые действия и возвращает управление вызвавшей ее подпрограмме. В основную программу управление возвращается, как обычно, после завершения работы подпрограммы. Такой прием часто используется для организации *интерфейса прикладного программирования* между различными программными компонентами

Х

ХАБ [hub]. См. *концентратор*

ХАКЕР [hacker]. Программист-фанатик, поглощенный доскональным изучением и совершенствованием программного обеспечения вычислительных систем. В настоящее время этот термин приобрел отрицательный оттенок из-за участившихся попыток тайного проникновения Х. в чужие компьютеры и *компьютерные сети*, "взлома" *защиты данных* и т. п. неблаговидных поступков. Ср. *кракер*

ХИТ [hit]. Загрузка любого элемента *веб-страницы* (*HTML-документа*, *графического файла*, *Джава-апплета* и т. д.) пользователем. Например, если на веб-странице присутствуют 15 графических элементов, то при ее загрузке сервер зарегистрирует 16 хитов — 15 картинок и один HTML-документ. Ср. *посещение*

ХОСТ [host]. 1. То же, что *главная ЭВМ*. 2. *Узел сети*, к которому подключаются терминалы. 3. Компьютер, на котором размещается *программное обеспечение*, предназначенное для использования в *режиме удаленного доступа*

ХУДОЖЕСТВЕННАЯ ЗАГОТОВКА [clip art]. То же, что *аппликация*

Ц

ЦАП [DAC]. То же, что *цифроаналоговый преобразователь*

ЦВЕТ [color]. 1. Свойство тела вызывать определенное зрительное ощущение. Многообразие Ц., которое видит человеческий глаз, определяется спектральным составом отраженного или испускаемого телом электромагнитного излучения — от красного Ц., которому соответствуют наиболее длинные электромагнитные волны, воспринимаемые глазом, до фиолетового, соответствующего наиболее коротковолновому излучению. Существуют две основные группы технических способов получения всевозможных оттенков Ц. 1) Аддитивное смешение Ц. создается либо путем быстрого чередования во времени излучений различного Ц., либо путем изображения мелких разноцветных точек, расположенных рядом. В обоих случаях глаз получает ощущение некоторого "среднего" Ц. 2) Субтрактивное смешение Ц. возникает либо при последовательном прохождении света через несколько разноцветных светофильтров, либо при механическом смешении красок. В принтерах и компьютерных видеосистемах применяются оба типа способов смешения Ц. Основными характеристиками Ц. являются тон — собственно Ц., определяемый длиной световой волны; насыщенность — "густота" Ц., например, красный Ц. насыщеннее розового; яркость, определяемая содержанием белого Ц. Для программного управления процессом получения нужного Ц. в компьютерной графике применяются различные способы опи-

сания Ц., или *цветовые модели.* 2. *Атрибут* выводимых на экран символов или элементов изображений. В программах может задаваться ключевым словом, например, `blue` — голубой, или *цифровым кодом.* См. *палитра*

ЦВЕТОВАЯ МОДЕЛЬ [color model]. 1. Физический метод формирования цветов на экране дисплея или в цветном принтере. Во многих цветных дисплеях используется Ц. м. RGB (red-green-blue) — красный-зеленый-голубой. В этой модели конкретный цвет или оттенок получается путем аддитивного смешения разных долей красного, зеленого и голубого цветов. Причем полное отсутствие этих цветов в смеси дает черный цвет, а сложение 100% всех цветов — белый цвет. Во многих системах печати используется Ц. м. CMY (cyan-magenta-yellow) — голубой-пурпурный-желтый, основанная на поглощающих свойствах красителей. В противоположность RGB-модели, здесь цвета и оттенки получаются путем вычитания из белого цвета определенных долей голубого, пурпурного и желтого. Если вычесть все три цвета полностью, остается черный цвет. 2. Способ описания и программирования цветов в *компьютерной графике.* Для описания цвета в компьютерной графике широко применяется Ц. м. HSB (hue-saturation-brightness) — тон-насыщенность-яркость. Каждый цвет или оттенок задается тремя количественными характеристиками. Тон определяется по угловой шкале цветового круга: 0° — красный, 60° — желтый, 120° — зеленый, 180° — голубой, 240° — синий, 300° — фиолетовый. Насыщенность и яркость выражаются в процентах. Например, яркость 0% соответствует черному цвету, а 100% — белому. См. *насыщенность цвета, палитра, тон, цвет, яркость*

ЦВЕТОВОЙ МАРКЕР [color marker]. Цветовая отметка, выделяющая один из *пунктов меню.* Ц. м. передвигается по пунктам меню при помощи *клавиш управления курсором* или *указателя мыши* и позволяет сделать нужный выбор. Для выбора того или иного пункта необходимо перевести на него Ц. м. Затем, нажав клавишу ввода или кнопку мыши, зафиксировать сделанный выбор

ЦВЕТОВОЙ ОХВАТ [gamut]. Диапазон цветов, которые способны обеспечить *устройства ввода* (сканер, цифровая камера) и *вывода* (монитор, принтер)

ЦВЕТОВОЙ ПРОФИЛЬ [profile]. *Файл*, в котором хранится информация о *цветовом охвате* конкретного *устройства ввода* или *вывода*

ЦВМ [digital computer]. То же, что *цифровая вычислительная машина*

ЦЕЛАЯ ПЕРЕМЕННАЯ, переменная целого типа [integer variable]. *Переменная*, значением которой могут быть только *числа целого типа.* Ц. п. вводятся в программу с помощью *описания переменных*, в котором указываются идентификатор (имя) переменной и *ключевое слово*, определяющее *целый тип*, значения которого она может принимать. В качестве ключевых слов в описаниях Ц. п. применяются: `integer` — в Фортране и Паскале и `int`, `long` или `short` — в языке Си, в зависимости от размера элемента памяти, отводимого под целое число. См. *описание переменной*

ЦЕЛОЕ ЧИСЛО [integer]. То же, что *число целого типа*

ЦЕЛОСТНОСТЬ ДАННЫХ [data integrity]. Непротиворечивость, полнота и сохранность данных. Например, Ц. д., хранящихся в *базе данных*, означает, что находящиеся в ней данные должны быть полностью согласованы между собой, содержать все сведения, необходимые для выполнения возложенных на базу данных функций, и при этом должно быть исключено их случайное уничтожение (стирание) или искажение. Все это обеспечивается *системой управления базой данных*

ЦЕЛОЧИСЛЕННЫЙ ТИП [integer type]. То же, что *целый тип*

ЦЕЛЫЙ ТИП, целочисленный тип [integer type]. *Тип данных*, множество значений которого составляют *числа целого типа*. Над этими числами допустимы *арифметические операции* и *операции сравнения*. Множество представимых в компьютере чисел Ц. т. ограничено и определяется *диапазоном изменения чисел* данной *вычислительной системы*, а выполняемые над ними *арифметические операции* осуществляются машиной по правилам элементарной арифметики: сложение и вычитание производятся по разрядам, умножение эквивалентно последовательности сложений числа с самим собой, а деление — последовательности вычитаний делителя из делимого. Результаты операций — также числа Ц. т. Например, 5/2 = 2 (но 5.0/2.0 = 2.5). В большинстве *языков программирования* Ц. т. является *встроенным типом*. При *описании переменных* и других *объектов программы*, относящихся к Ц. т., применяются *ключевые слова*: `integer` — в языках Фортран и Паскаль, `int`, `long` и `short` — в языке Си, в зависимости от размера элемента памяти, отводимого под целое число. Ср. *вещественный тип*

ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР (ЦП) [central processing unit (CPU)]. 1. Главный рабочий *процессор* компьютера или *вычислительной системы*, выполняющий основные функции по *обработке данных* и управлению работой других устройств. В *персональном компьютере* это микросхема, управляющая работой компьютера и производящая основные вычисления. Функционально ЦП подразделяется на *устройство управления* и *арифметико-логическое устройство*, выполняющие операции в соответствии с программой. Кроме того, ЦП имеет внутреннюю процессорную память, состоящую из регистров и предназначенную для временного хранения данных, непосредственно используемых в выполнении операций. В *персональных компьютерах* применяются следующие процессоры, разработанные фирмой Intel (в порядке возрастания вычислительных возможностей): 80286; 80386SX; 80386DX; 80486SX; 80486DX; Pentium, Pentium II, Pentium III. Процессор 80286 — 16-разрядный, а все остальные — 32-разрядные, т. е. порции данных, с которыми ЦП может выполнять различные операции, составляют соответственно 16 бит (2 байта) и 32 бита (4 байта). В процессорах 80486 и Pentium встроены: *математический сопроцессор* — для ускорения выполнения операций над *числами с плавающей точкой*; *кэш-память* — для быстрой выборки данных из

оперативной памяти. Основная характеристика этих процессоров — *тактовая частота.* См. *функциональная схема ЭВМ, конфигурация компьютера.* В настоящее время в основном применяются процессоры Pentium и аналогичные процессоры других производителей. В процессорах Pentium применена 64- или 128-разрядная магистраль, значительно ускоряющая обмен данными с оперативной памятью. Здесь также одно исполнительное устройство заменено на два — U и V, каждое — со своим собственным конвейером; оба параллельно ведут выборку, расшифровку и выполнение команд; устройство U является основным и может выполнять все команды, устройство V — вспомогательным и выполняет только наиболее часто встречающиеся типы команд. Внутренняя кэш-память разделена на кэш команд и кэш данных. Есть система предсказания переходов путем опережающего просмотра программы, что позволяет в случае верного предсказания выполнить переход за один такт. Улучшен (по сравнению с 80486) *математический сопроцессор.* В процессорах Pentium MMX реализованы расширения MultiMedia eXtension (мультимедийные расширения) — дополнительные возможности, ориентированные на обработку цифрового изображения и звука, которые включают в себя свыше 50 новых команд, предназначенных для обработки звуковых и видеосигналов. Использование MMX позволяет перенести основную нагрузку по обработке изображения и звука на ЦП, оставив видео- и звуковым адаптерам только преобразование аналоговых сигналов в цифровые. 2. В многомашинном вычислительном комплексе или в *компьютерной сети* ЦП называют *главную ЭВМ*, которая кроме выполнения основной обработки информации управляет работой других ЭВМ

ЦЕРН [European Particle Physics Laboratory, франц.: **Conseil Europeen pour la Recherche Nucleaire (CERN)]**. То же, что *Европейская лаборатория ядерных исследований*

ЦИКЛ [cycle, loop]. 1. То же, что *циклический процесс.* 2. Многократно выполняющийся участок алгоритма или программы. Нужное количество повторений можно обеспечить, осуществляя переход на начало повторяемого участка программы в зависимости от выполнения (или невыполнения) некоторого *условия выхода из цикла.* Проверка этого условия может выполняться в конце повторяемого участка. Такой Ц. называют *циклом с постусловием*, а проверяемое условие — условием повторения (окончания). Схему Ц. с постусловием см. на рис. Ц.1. Можно организовать Ц. так, чтобы решение о его выполнении или невыполнении принималось по условию, расположенному в начале повторяемого участка. Такой Ц. называют *циклом с предусловием*, а проверяемое условие — условием выполнения (продолжения). Схему Ц. с предусловием см. на рис. Ц.2. В зависимости от программируемого алгоритма число необходимых повторений может быть либо заранее известным, либо неизвестным и определяться по мере достижения требуемого результата. В *языках программирования* существуют *операторы цикла* — специальные программные конструкции, обеспечивающие при определенных

условиях многократное последовательное выполнение одного и того же участка программы. См. *операторы цикла языка Паскаль, операторы цикла языка Си, оператор цикла языка Фортран*

ЦИКЛИЧЕСКИЙ ПРОЦЕСС, цикл [cyclic process, cycle]. Процесс многократного последовательного выполнения одного и того же участка алгоритма или программы. Так как компьютер производит операции над данными с огромной скоростью, то ему выгодно поручать задачи, которые требуют многократного повторения действий (при различных значениях обрабатываемых данных), дающие возможность с помощью краткой программы задать большой объем вычислений. Программные конструкции, позволяющие программировать Ц. п., называются *циклами*. Они включают в себя команды или операторы, непосредственно реализующие повторяемые действия алгоритма — т. н. *тело цикла*, и команды или операторы, управляющие повторением и осуществляющие *выход из цикла* в нужный момент. В зависимости от программируемого алгоритма количество повторений может быть известно заранее (например, *цикл со счетчиком*), а может зависеть от вычисленных в ходе Ц. п. величин (например, *итерационный цикл*). См. *цикл*

ЦИКЛ С ВЕРХНИМ ОКОНЧАНИЕМ [while loop]. То же, что *цикл с предусловием*

ЦИКЛ С НИЖНИМ ОКОНЧАНИЕМ [repeat-until loop]. То же, что *цикл с постусловием*

ЦИКЛ СО СЧЕТЧИКОМ, арифметический цикл [cycle with a counter, arithmetical cycle]. Разновидность *цикла с параметром*. При выполнении Ц. с. с. определенная часть программы периодически повторяется, а число повторений (проходов, итераций) регистрируется счетчиком. По достижении нужного количества проходов выполнение цикла прекращается. Текущее значение счетчика часто используется в *теле цикла*. Ц. с. с. могут быть как *циклами с предусловием*, так и *циклами с постусловием*. В операторе *языка программирования высокого уровня*, реализующем такой цикл, как правило, нужно задать начальное и конечное значения счетчика. Например, вычисление суммы кубов первых ста чисел натурального ряда $S = 1^3 + 2^3 + 3^3 + ... + 100^3$ программируется в форме Ц. с. с., записи которого на языках Фортран, Паскаль и Си имеют вид, представленный ниже:

Фортран

```
   DO 99 I=1,100
   S=S+I*I*I
99 CONTINUE
```

Паскаль

```
S =0;
for i:=1 to 100 do
 S=S+i*i*i;
```

Си

```
for(S=0,i=1; i<=100; i++)
 S+=i*i*i;
```

См. *операторы цикла языка Паскаль, операторы цикла языка Си, оператор цикла языка Фортран*

ЦИКЛ С ПАРАМЕТРОМ [cycle with loop variable, cycle with parameter]. *Цикл*, в котором явно задается управляющая количеством повторений переменная *(параметр цикла)*, правило вычисления последовательности ее значений и использующее эти значения *условие выхода из цикла*. В операторе цикла указанные данные обычно называются списком цикла. Например, в списке Ц. с п. для управляющей переменной могут быть заданы ее начальное и конечное значения и шаг изменения. Текущее значение параметра цикла можно использовать в *теле цикла*, но изменять здесь это значение не рекомендуется. Количество повторений в Ц. с п., как правило, заранее известно или может быть вычислено. Такой цикл — непременная часть алгоритмов табулирования функций, вычисления конечных сумм, произведений и т. д. Примером Ц. с п. является *цикл со счетчиком*. Ср. *итерационный цикл*

ЦИКЛ С ПОСТУСЛОВИЕМ, цикл с нижним окончанием [repeat-until loop]. *Цикл*, в котором проверка *условия выхода из цикла* осуществляется в конце выполнения *тела цикла*. Это условие может формулироваться либо как условие повторения тела цикла, либо как условие окончания цикла. В первом случае оно имеет вид логического выражения, значение которого "ИСТИНА" в случае, если тело цикла нужно повторить, и "ЛОЖЬ", если этого делать не надо. Во втором — логического выражения, значение которого "ИСТИНА", если тело цикла больше не надо повторять. Блок-схема Ц. с п. приведена на рис. Ц.1. Так как указанные условия располагаются в конце тела цикла, то тело Ц. с п. должно выполниться хотя бы один раз. См. *операторы цикла языка Паскаль, операторы цикла языка Си, оператор цикла языка Фортран*

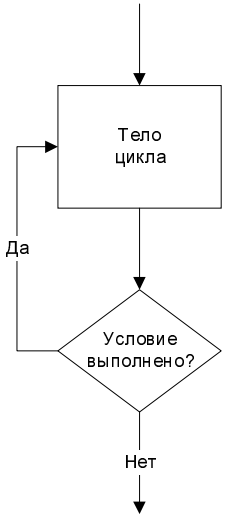

Рис. Ц.1. Блок-схема цикла с постусловием

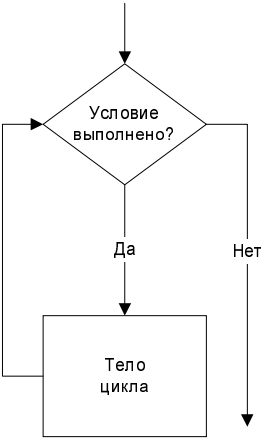

Рис. Ц.2. Блок-схема цикла с предусловием

ЦИКЛ С ПРЕДУСЛОВИЕМ, цикл с верхним окончанием [while loop]. *Цикл*, в котором проверка *условия выхода из цикла* осуществляется в начале выполнения *тела цикла*. Это условие обычно формулируется как условие выполнения тела цикла в виде *логического выражения*, значение которого "ИСТИНА" в случае, если тело цикла нужно выполнить, и "ЛОЖЬ", если этого делать не надо. Блок-схема Ц. с п. приведена на рис. Ц.2. Правила работы цикла таковы: тело цикла выполняется, пока условие сохраняет силу (значение условного выражения — "ИСТИНА"); если же с самого начала условие не соблюдается, то тело цикла не выполнится ни разу. См. *операторы цикла языка Паскаль, операторы цикла языка Си, оператор цикла языка Фортран*

ЦИТИРОВАНИЕ [quoting]. Включение части полученного сообщения *электронной почты* в ответное сообщение. Ц. упрощает чтение ответа на сообщение, поскольку отсутствует необходимость одновременно смотреть на исходное сообщение

ЦИФРА [digit]. *Символ*, используемый для написания чисел в *позиционной системе счисления*

ЦИФРОАНАЛОГОВЫЙ ПРЕОБРАЗОВАТЕЛЬ (ЦАП) [digital-to-analog converter (DAC)]. Устройство для автоматического преобразования *цифрового сигнала* в *аналоговый*. Например, ЦАП видеоадаптера преобразует цифровые данные об интенсивности цвета в уровни непрерывного сигнала, подаваемые на монитор. Противоп. *аналого-цифровой преобразователь*

ЦИФРОВАЯ ВЫЧИСЛИТЕЛЬНАЯ МАШИНА (ЦВМ) [digital computer]. *Вычислительная машина*, которая обрабатывает информацию в цифровой (дискретной) форме. В отличие от *аналоговой вычислительной машины* в ЦВМ все вычисления производятся над конечными числами и с ограниченной точностью, определяемой разрядностью и формой представления чисел. ЦВМ являются как механический арифмометр, так и современный компьютер. См. *цифровая форма данных*

ЦИФРОВАЯ КАМЕРА [digital still camera, digital camera]. То же, что *цифровой фотоаппарат*

ЦИФРОВАЯ ПОДПИСЬ, электронная подпись [digital signature, signature]. Метод удостоверения подлинности сообщения, основанный на применении метода *шифрования с открытым ключом*. Суть этого метода состоит в следующем. Отправитель сообщения кодирует сообщение S своим закрытым ключом и отправляет получателю подписанное сообщение (само сообщение S и его код C). Получатель сообщения еще раз кодирует код C с помощью открытого ключа отправителя, получая сообщение S'. Если S и S' совпали, то это значит, что (нешифрованное!) сообщение S не было искажено злоумышленником при передаче и действительно было отправлено отправителем, опубликовавшим свой открытый ключ. Подделать Ц. п. практически невозможно

ЦИФРОВАЯ ФОРМА ДАННЫХ [digital data form]. Способ представления, хранения и *передачи данных* в виде *цифрового кода* или *цифровых сигналов*. Ср. *аналоговая форма данных*

ЦИФРОВАЯ ФОТОКАМЕРА [digital still camera, digital camera]. То же, что *цифровой фотоаппарат*

ЦИФРОВОЕ УСТРОЙСТВО [digital device]. *Устройство*, в котором информация представляется в виде дискретно изменяющихся *цифровых сигналов*. Это устройство, в котором для обработки, хранения, отображения или передачи информации используется *цифровая форма данных*. Ср. *аналоговое устройство*

ЦИФРОВОЙ ИНТЕРФЕЙС МУЗЫКАЛЬНЫХ ИНСТРУМЕНТОВ [musical instrument digital interface (MIDI)]. Стандарт *интерфейса* электромузыкальной аппаратуры. MIDI-стандарт определяет совокупность правил цифрового кодирования звуков, формат и процедуры передачи звуковых данных, а также архитектуру MIDI-*аппаратных средств*: типы *каналов ввода/вывода* звуковых данных и типы разъемов и кабелей, соединяющих компьютеры с музыкальными синтезаторами и другими устройствами, создающими или воспроизводящими звук. В MIDI стандартизованы коды многих музыкальных инструментов и тембров их звучания. В памяти компьютера звук хранится в MIDI-файлах, содержание которых с помощью звуковой платы воспроизводится акустической системой

ЦИФРОВОЙ КОД [numeric code]. *Код*, набор знаков которого состоит только из цифр

ЦИФРОВОЙ ПЛАНШЕТ [digital tablet]. То же, что *графический планшет*

ЦИФРОВОЙ ПЛАНШЕТ ГРАФИЧЕСКОГО ВВОДА [digital tablet]. То же, что *графический планшет*

ЦИФРОВОЙ СЕРТИФИКАТ [digital certificate]. *Текстовый файл*, в котором содержится информация о владельце и организации, выдавшей и хранящей сертификат, а также, в зашифрованном виде, сведения для идентификации полномочий владельца. Ц. с. используется в средствах защиты веб-сайтов *электронной коммерции* наряду с *шифрованием с открытым ключом*

ЦИФРОВОЙ СИГНАЛ, дискретный сигнал [digital signal, discrete signal]. Электрический сигнал, напряжение которого может изменяться только дискретно, находясь в каждый момент времени на одном из нескольких (обычно двух) уровней. Каждый уровень имеет свои границы — пороги и соответствует определенной цифре. В вычислительной технике применяются двухуровневые Ц. с., которые иногда называют двоичными, поскольку в них один уровень (обычно высокий) соответствует цифре 1 (истина), а другой — 0 (ложь). Ср. *аналоговый сигнал*

ЦИФРОВОЙ ФОТОАППАРАТ, цифровая камера, цифровая фотокамера [digital still camera, digital camera]. Фотокамера (рис. Ц.3), использующая для получения изображения ПЗС-матрицу, с которой изображение в цифровой форме в сжатом виде записывается в энергонезависимую память (например, на специальную дискету). Отснятые кадры через последовательный порт можно ввести в компьютер для последующей обработки с помощью *графического редактора*

Рис. Ц.3. Цифровые фотоаппараты

ЦОС [digital signal processor (DSP)]. То же, что *процессор цифровой обработки сигналов*

ЦП [CPU]. То же, что *центральный процессор*

Ч

ЧАТ [chat]. То же, что *разговор в сети Интернет*

ЧЕРЕДУЕМАЯ ПАМЯТЬ [switching memory, flip-flops]. *Разделяемая память*, адресное пространство которой организовано по принципу чередования адресов. Ч. п. делится на *банки памяти*, запросы к которым могут обрабатываться одновременно. Принято соглашение, что ячейка памяти с номером i находится в банке с номером $i \bmod n$, где n — количество банков памяти, а *mod* — операция вычисления остатка от деления. При этом подряд расположенные n ячеек памяти попадают в разные банки. Например, если имеется 8 банков памяти, то первому банку будут принадлежать ячейки с номерами 0, 8, 16, ..., второму — 1, 9, 17, ... и т. д. Чаще всего число банков памяти равно степени двойки: $n = 2^m$, и номер необходимого банка просто равен числу, записанному в m младших разрядов адреса. При такой организации максимально эффективна *параллельная обработка* векторов.

ЧЕРЕССТРОЧНАЯ ЗАГРУЗКА ИЗОБРАЖЕНИЯ [interlaced loading]. См. *чересстрочный GIF*

ЧЕРЕССТРОЧНЫЙ GIF [interlaced GIF]. *Графический файл* в *формате GIF*, отображение которого в *программе просмотра* (браузере) происходит постепенно с нарастающим уровнем детализации. Это достигается чересстрочной записью изображения в файл. В самом начале файла хранятся строки изображения с номерами, кратными восьми, затем с номерами, кратными четырем, и т. д. Соответственно, и загрузка, называемая чересстрочной, происходит подобным образом. Сначала отображаются строки рисунка, кратные восьми, затем следующие, улучшая детализацию. Данное представление рисунка позволяет увидеть его приблизительное отображение до завершения полной загрузки

ЧИП [chip]. То же, что *микросхема*

ЧИСЛО [number]. *Символ* или объединение нескольких символов, представляющие количественную величину в определенной *системе счисления*. Например, Ч. "десять" в римской системе счисления обозначается одним символом X, а в арабской — двумя: 10. Последние два символа в *двоичной системе счисления* обозначают Ч. "два". В *языках программирования высокого уровня* применяются в основном *целые*, *вещественные* (действительные) и в некоторых — *комплексные* Ч., для записи которых используется *десятичное представление*. Любому Ч. (кроме определенного как Ч. *без знака*) может предшествовать знак + или −. Вещественное число записывается в формах Ч. *с фиксированной* или *плавающей точкой*. Кроме десятичных в программировании используются Ч. в *восьмеричной* и *шестнадцатеричной системах счисления*. Для внутреннего представления Ч. в компьютере применяется *двоичная система счисления*. См. *число вещественного типа, число целого типа, вещественное число, представление чисел с плавающей точкой*

ЧИСЛО ВЕЩЕСТВЕННОГО ТИПА [real type number]. Рациональное число, которое может иметь либо форму *числа с фиксированной точкой*, либо форму *числа с плавающей точкой*

ЧИСЛО С ПЛАВАЮЩЕЙ ТОЧКОЙ [floating-point number]. Рациональное число, представленное в форме, содержащей явно заданный *порядок числа*, например $0.12345 \cdot 10^3$. В общем случае Ч. с п. т. имеет вид $M \cdot q^p$, где M — *число с фиксированной точкой*, называемое мантиссой, q — основание системы счисления, p — *число целого типа*, называемое порядком. В приведенном примере 0.12345 — мантисса, 10 — основание десятичной системы счисления, 3 — порядок. Запись числа в форме Ч. с п. т. может быть сделана многими способами. Например, то же число может быть записано как $0.012345 \cdot 10^4$ и $1.2345 \cdot 10^2$. Чтобы избежать такой неоднозначности, обычно применяется *нормализованная форма* $0.12345 \cdot 10^3$. В *языках программирования* Паскаль, Фортран и др. вместо основания системы 10 в десятичных Ч. с п. т. пишут букву E или e, а знак умножения после мантиссы не ставится, например 0.123E−5 или 0.123e−5 вместо $0.123 \cdot 10^{-5}$. *Арифметические операции* над Ч. с п. т. выполняются отдельно над мантиссой и порядком, на-

пример, при умножении таких чисел их мантиссы перемножаются, а порядки складываются. Поэтому при размещении Ч. с п. т. в памяти ЭВМ оно разбивается на мантиссу и порядок, которые раздельно помещаются в отведенные для них части *ячейки памяти*. Этот способ называется *полулогарифмическим представлением чисел*. Конечно, при размещении чисел в памяти ЭВМ используется *двоичное представление* мантиссы и порядка. См. *представление чисел с плавающей точкой*

ЧИСЛО СТРОК КОДА [lines of code (LOC)]. Количественная характеристика программы. Часто используют единицу KLOC, означающую тысячу строк *исходного кода*. См. *метрика*

ЧИСЛО С ФИКСИРОВАННОЙ ТОЧКОЙ [fixed-point number]. Рациональное число, представленное в форме, содержащей целую и дробную части, разделенные точкой, например 123.45, 1.0, −0.329, +0.001. Ч. с ф. т. в основном используются при программировании для записи констант и при выводе данных

ЧИСЛО ЦЕЛОГО ТИПА [integer type number]. Положительное или отрицательное целое число, включая ноль, например, 0, 7, 003, +500, −4096. Ч. ц. т. не должно содержать никаких других символов, кроме знака и цифр, например, числа 14.0 и $2 \cdot 10^5$ не являются Ч. ц. т. В ЭВМ Ч. ц. т. обычно представляются в *двоичном коде* и имеют свойства, близкие к свойствам натуральных чисел. Например, ниже показано размещение числа 5 в байте памяти в форме числа целого типа (двоичное представление):

	0	1	0	1
Знак	Абсолютная величина			

ЧИСТКА ПАМЯТИ, сбор мусора, сборка мусора [memory cleaning, garbage collection]. Операция по выявлению и освобождению выделенных программе, но фактически не используемых блоков *оперативной памяти*. Ч. п. выполняется в программах, которые используют *динамическое распределение памяти*. Так называемый мусор — это выделенная программе память, которая на самом деле не используется. В программах на *языках программирования высокого уровня* (например, Паскале, Си) мусор может появляться следующим образом. Программа может запросить *блок памяти* и поместить адрес этого блока в переменную, являющуюся указателем. Затем указателю можно присвоить другое значение, в результате чего выделенный блок памяти будет недоступен программе, если только на него не указывает какой-то другой указатель. Образование мусора может не быть ошибкой в программе, но накопление мусора приводит к нерациональному использованию ресурсов, в результате чего программа может не закончить свою работу из-за нехватки памяти, хотя на самом деле имеющейся памяти достаточно для

решения задачи. Существуют два основных способа решения этой проблемы. Во-первых, программисту рекомендуется следить за использованием памяти в программе и явным образом освобождать блоки памяти, которые более не нужны. Во-вторых, некоторые *системы программирования* в случае нехватки памяти автоматически выполняют Ч. п., т. е. выявляют выделенные блоки памяти, к которым программа не имеет доступа, и возвращают эти блоки в число свободных

Ш

ШАБЛОН [template]. Документ, используемый в качестве образца для создания новых документов. Например, приложение Microsoft Word всегда использует некоторый Ш. при создании нового документа. Пользователь может либо выбрать имеющийся Ш., либо применить Ш. по умолчанию

ШАБЛОН ИМЕНИ ФАЙЛА [file name picture]. Последовательность знаков, позволяющих обозначить *имена* сразу нескольких *файлов*. Ш. и. ф. образуется из символов, присутствующих в имени файла, и знаков, указывающих, что на их месте в имени символы либо отсутствуют, либо могут стоять любые допустимые символы. Большинство операционных и поисковых систем позволяют создавать Ш. и. ф., используя знаки * и ?. Знак * заменяет любое количество любых символов в имени файла или в расширении имени файла, а знак ? — один произвольный символ или его отсутствие. Например, Ш. и. ф. *.* обозначает все файлы текущего каталога; Ш. и. ф. *.exe — все файлы с расширением exe; Ш. и. ф. a:\doc\ab??.t* — все файлы из каталога a:\doc с именем, начинающимся с ab и состоящим не более чем из 4-х символов, и расширением, начинающимся с буквы t

ШАРОВОЙ МАНИПУЛЯТОР [track ball]. То же, что *трекбол*

ШЕСТНАДЦАТЕРИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ [hexadecimal number system]. *Позиционная система счисления* с основанием системы $q = 16$. В качестве шестнадцатеричных цифр используются символы 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Например, $(B3D)_{16} = 11 \cdot 16^2 + 3 \cdot 16^1 + 13 \cdot 16^0 = (2877)_{10}$

ШИНА [bus]. *Устройство*, служащее для *передачи данных* и *управляющих сигналов* между компонентами компьютера. Состоит из линий электрических соединений. По каждой линии Ш. обычно передается один *двоичный разряд информационного слова*, являющегося элементом данных или адресом. Поэтому максимальная пропускная способность Ш. обеспечивается при числе линий, равном сумме числа бит в *машинном слове* и максимального числа *двоичных разрядов* адреса, плюс оптимальное количество линий для передачи управляющих сигналов. Существует способ организации ЭВМ, при котором все ее компоненты подключаются к единой Ш. Это так называемая шинная

архитектура компьютера. В *персональных компьютерах* применяются Ш. высокой пропускной способности. См. *адресная шина, локальная шина, системная шина, магистраль*

ШИНА ДАННЫХ [data bus]. *Шина* или часть шины, предназначенная для параллельной передачи двоичных разрядов данных между устройствами компьютера

ШИНА EISA [EISA-bus]. См. *расширенная архитектура промышленного стандарта*

ШИНА VLB, локальная шина VESA [VESA local bus (VLB)]. *Локальная шина*, стандарт которой разработан *ассоциацией VESA*. Служит для подключения высокоскоростных видеоплат

ШИННАЯ СЕТЬ, сеть шинной топологии [bus network]. *Локальная вычислительная сеть*, в которой все *узлы сети* подключены к основной *линии связи* (шине). В Ш. с. каждый узел следит за работой линии. Сообщения регистрируются всеми узлами, но принимаются только теми из них, кому они адресованы. Поскольку Ш. с. базируется на общей магистрали данных, неисправный узел просто выходит из связи, не нарушая общей работы, как это было бы в *кольцевой сети*, где сообщения передаются от одного узла к следующему. Во избежание проблем при попытке двух или более узлов одновременно воспользоваться линией в Ш. с. для регулирования потока сообщений обычно применяется механизм разрешения конфликтов или передачи маркера. См. *топология сети*. Ср. *древовидная сеть, кольцевая сеть, радиальная сеть*

ШИННАЯ СЕТЬ С МАРКЕРНЫМ ДОСТУПОМ, шинная сеть с передачей маркера, маркерная шинная сеть [token bus network]. *Сеть с маркерным доступом,* выполненная по шинной топологии. Регламентируется *стандартом EEE* 802.4. См. *сеть шинной топологии*

ШИННАЯ СЕТЬ С ПЕРЕДАЧЕЙ МАРКЕРА [token bus network]. То же, что *шинная сеть с маркерным доступом*

ШИРИНА ПОЛОСЫ ПРОПУСКАНИЯ [bandwidth]. В системах связи разность между верхней и нижней частотными границами данного диапазона. В компьютерных сетях большая *ширина полосы пропускания* означает возможность более быстрой *передачи данных*

ШИРОКОВЕЩАНИЕ [broadcasting]. Способ *передачи данных*, при котором посылка данных достигает всех компьютеров в данной *компьютерной сети*

ШИРОКОВЕЩАТЕЛЬНАЯ КОМПЬЮТЕРНАЯ СЕТЬ, широковещательная сеть [broadcast network]. *Компьютерная сеть*, в которой возможно *широковещание*. К такой сети подключено более двух маршрутизаторов, и имеется возможность передать одно физическое сообщение всем подсоединенным к ней маршрутизаторам. Примером Ш. с. является *сеть Ethernet*. См. *широковещательное сообщение*

ШИРОКОВЕЩАТЕЛЬНАЯ СЕТЬ [broadcast network]. То же, что *широковещательная компьютерная сеть*

ШИРОКОВЕЩАТЕЛЬНОЕ СООБЩЕНИЕ [broadcast message]. Сетевое *сообщение*, рассылаемое по всем устройствам, расположенным в пределах общих с отправляющим компьютером *сегментов сети*

ШИФРАТОР [scrambler, encipherer, encoder, encryptor]. 1. Программа, выполняющая *шифрование*. 2. Блок ЭВМ, выполняющий преобразование входных сигналов. Например, *аналого-цифровой преобразователь*, преобразующий *аналоговый сигнал*, поступающий на компьютер, в *цифровой сигнал*. Противоп. *дешифратор*

ШИФРОВАНИЕ [enciphering, encoding, encryption]. Алгоритмическое преобразование данных для защиты от несанкционированного прочтения и использования. Выполняется вручную или с помощью программы-*шифратора*. Широко применяется при передаче конфиденциальной информации по каналам общественных телекоммуникаций (телефонной, радио- и спутниковой связи), а также при ее сохранении на магнитном носителе. Ш. выполняется с помощью шифра — кода, основные элементы которого (секретный ключ и, возможно, правила кодирования) известны ограниченному кругу лиц. Расшифровка производится также вручную или с помощью программы-*дешифратора*. Известны две основные группы программных методов Ш.: с *открытым ключом* и с *закрытым ключом*. Ср. *кодирование символов*. См. *защита данных*

ШИФРОВАНИЕ С ЗАКРЫТЫМ КЛЮЧОМ, симметричное шифрование [secret key enciphering, secret key encryption, symmetric enciphering, symmetric encryption]. Метод *шифрования*, при котором для зашифровки и расшифровки сообщений применяется один и тот же секретный ключ. Наиболее надежным из методов Ш. с з. к. в настоящее время считается метод DES (Data Encryption Standard), в котором используется ключ длиной 64 бита. Вскрыть такой шифр можно только подбором ключа, для чего нужно перебрать 2^{64} возможных комбинаций, что практически невозможно даже с применением *суперкомпьютеров*. Ср. *шифрование с открытым ключом*

ШИФРОВАНИЕ С ОТКРЫТЫМ КЛЮЧОМ, несимметричное шифрование [open key enciphering, open key encryption, RSA enciphering, RSA encryption]. Метод *шифрования*, при котором для зашифровки сообщений применяется открытый ключ, который получатель шифрованных сообщений сообщает всем желающим отправить сообщение, а расшифровка сообщения производится получателем сообщения с помощью закрытого ключа, который держится в секрете. Злоумышленник, перехвативший шифрованное сообщение, не сможет его расшифровать, даже зная открытый ключ, которым сообщение было зашифровано, и метод шифрования. Этот метод основан на использовании таких функций, которые сами вычисляются сравнительно про-

сто, но обратные к ним, хотя и существуют, требуют настолько трудоемких вычислений, что это практически невозможно сделать даже с применением *суперкомпьютеров*. Существование таких функций было обнаружено математиками сравнительно недавно. В настоящее время Ш. с о. к. бурно развивается, поскольку это очень надежный, удобный (открытый ключ можно, например, публиковать в сети Интернет) и сравнительно простой метод. Иногда Ш. с о. к. называют методом RSA по начальным буквам фамилий авторов одного из первых вариантов этого метода: Rivest, Shamir и Adelman. См. *цифровая подпись*, ср. *шифрование с закрытым ключом*

ШИФРУЮЩАЯ ФАЙЛОВАЯ СИСТЕМА [encrypting file system (EFS)]. Надстройка *файловой системы NTFS*, позволяющая пользователю прозрачно шифровать содержимое файлов и каталогов, делая их защищенными от прочтения

ШЛЮЗ [gateway]. Совокупность технических и программных средств, обеспечивающая сопряжение нескольких сетей различной архитектуры

ШРИФТ [font]. Конкретный способ изображения символов из некоторого набора. Обычно в Ш. входят буквы одного или нескольких алфавитов, цифры и специальные знаки. Кстати, этот словарь набран шрифтами именно такого состава. Бывают также специальные шрифты, в которые входят, например, *псевдографические символы*, математические или астрономические символы и т. п. Кроме набора отображаемых символов Ш. характеризуется множеством других свойств, из которых важнейшими являются *гарнитура Ш.*, *начертание Ш.* и *размер Ш.* По способу представления в компьютере различаются растровые Ш. и векторные Ш. В первом случае для представления шрифтов используются методы *растровой графики* и символы Ш. представляют собой массивы *пикселов*. Во втором случае используются методы *векторной графики* и форма символов Ш. описывается математическими функциями. Свойства растровых и векторных Ш. существенно различны. Например, векторный Ш. допускает произвольное масштабирование (т. е. изменение размера), в то время как растровый Ш. не допускает масштабирования или допускает масштабирование только с определенными коэффициентами. С другой стороны, работа с растровым Ш. значительно быстрее, поскольку при этом не требуется дополнительной обработки для вывода символа Ш. на экран или бумагу. Среди векторных Ш. наибольшее распространение получили Ш. типа TrueType

ШРИФТ TRUETYPE [TrueType font]. Масштабируемый *векторный шрифт*. Знак Ш. TrueType задается программной процедурой, описывающей его контур, составленный из гладко сопрягаемых прямых и кривых. Кроме знаковых процедур Ш. TrueType содержит информацию о способах изображения знаков малых размеров на экране и при печати. В *операционную систему* встроен генератор шрифтов, а знаковые процедуры и информация о шрифтах хранятся на диске в виде отдельных файлов. Такая технология позволяет

включать в операционные системы высококачественные шрифты, хорошо воспринимаемые глазом, особенно на экране. Первоначально технология Ш. TrueType была разработана для *операционной системы Windows*. В настоящее время она применяется и в других системах

Щ

ЩЕЛЧОК [click]. Действие, которое заключается в том, чтобы быстро нажать и отпустить кнопку неподвижной мыши. Щ. применяется при задании команды с помощью *элемента управления*, быстрой активизации окон и т. п. При этом *указатель мыши* должен указывать на выбранный элемент, *значок*, программу и т. п. Ср. *двойной щелчок*

Э

ЭВМ [computer]. То же, что *электронная вычислительная машина*

ЭКЗЕМПЛЯР КЛАССА [class instance, instance]. Термин *объектно-ориентированного программирования*, которым обозначается конкретный объект, созданный конструктором некоторого класса. Э. к. обладает свойствами, которые хранят данные, присущие именно этому объекту, и методами, которые определены в классе или унаследованы от родительских классов. См. *класс, метод класса, наследование, свойство класса*

ЭКРАН [screen]. Устройство отображения текста или *графической информации*. См. *дисплей*

ЭКРАННАЯ ЗАСТАВКА [screen saver]. Движущийся рисунок или узор, появляющийся на экране, если в течение указанного периода времени пользователь не выполняет действия с мышью или клавиатурой. Помимо декоративной функции, Э. з. служит сохранению свойств электронно-лучевой трубки *дисплея*, которые могут пострадать при длительном отображении постоянного изображения. На рис. Э.1 приведено диалоговое окно операционной системы Windows для управления Э. з.

ЭКРАННАЯ ПАНЕЛЬ [screen panel]. То же, что *панель экрана*

ЭКСПЕРТНАЯ ГРУППА ПО КИНЕМАТОГРАФИИ [Moving Pictures Expert Group (MPEG)]. То же, что *группа MPEG*

ЭКСПЕРТНАЯ СИСТЕМА (ЭС) [expert system]. Система *программных* и *аппаратных средств*, включающая *базу знаний*, способная на основании методов *искусственного интеллекта* и предоставляемых пользователем фактов идентифицировать ситуацию, поставить диагноз, сделать прогноз, сгенери-

ровать решение или дать рекомендацию для выбора действия. ЭС обычно ориентируется на некоторую предметную область (например, ЭС диагностики и лечения сердечно-сосудистых заболеваний) и способна получать, накапливать, корректировать знания из этой области, выводить новые знания из уже известных, решать практические задачи на основе этих знаний и объяснять ход решения. Помимо базы знаний, хранящей факты, закономерности и правила, в ЭС входят программы-решатели, реализующие функции планирования, поиска решения задачи, механизма логического вывода из часто неполных и нечетких знаний. Создание ЭС начинается с разработки ее первоначального варианта — прототипа ЭС. Далее следует длительный, многоэтапный процесс испытаний и совершенствования. Для программирования ЭС применяются в основном языки Лисп, Пролог, Си

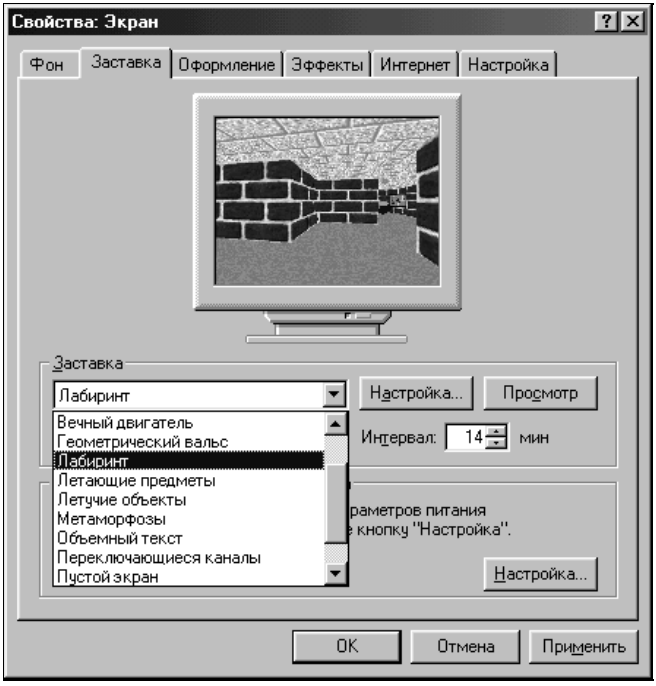

Рис. Э.1. Диалоговое окно для управления экранной заставкой

ЭКСТРАНЕТ, расширенная интрасеть, сеть extranet [extranet]. Корпоративная сеть *интранет*, в которой используются технологии Интернета для связи с деловыми партнерами (заказчиками, дилерами, поставщиками и т. п.)

ЭЛЕКТРОННАЯ ВЫЧИСЛИТЕЛЬНАЯ МАШИНА (ЭВМ), компьютер [computer]. *Вычислительная машина*, основными элементами которой являются электронные приборы (транзисторы, *микросхемы*, магнитные элементы и т. п.). В настоящее время под ЭВМ понимают в основном электронные *цифровые вычислительные машины*. Термин "ЭВМ" чаще применяется при рассмотрении внутреннего устройства вычислительной машины

ЭЛЕКТРОННАЯ КОММЕРЦИЯ [electronic commerce, e-commerce]. Деловое взаимодействие между коммерческими партнерами с помощью *компьютерных сетей*. Основой Э. к. являются сети Интернет, интранет, экстранет, а также частные интегрированные сети с дополнительными услугами. Э. к. позволяет автоматизировать, а следовательно, ускорить и удешевить, процессы купли-продажи. При этом осуществляется электронный обмен транзакциями и деловыми документами между компьютерными программами покупателя, продавца и его партнеров по бизнесу. Для повышения эффективности систем Э. к. разрабатываются стандартные формы такого обмена. Одним из способов стандартизации электронного обмена деловыми документами как внутри организации, так и между организациями является применение *стандартов EDI*. См. *электронная коммерция класса business-to-business*, *электронная коммерция класса business-to-consumer*

ЭЛЕКТРОННАЯ КОММЕРЦИЯ КЛАССА BUSINESS-TO-BUSINESS, электронная коммерция класса B2B [business-to-business e-commerce]. *Электронная коммерция* между деловыми партнерами, основанная на *сети extranet*. Э. к. к. B2B осуществляется между ограниченным количеством доверенных партнеров, а платежи производятся в соответствии с заранее определенными правилами кредитования. Большинство предприятий, участвующих в Э. к. к. B2B, создают электронные каталоги, помогающие *электронным партнерам* проводить быстрый поиск и сравнение необходимой продукции. Для безопасного обмена конфиденциальной информацией между партнерами системы Э. к. к. B2B должны быть защищены брандмауэрами от нежелательных вторжений в частную сеть, а также применением паролей, аутентификации, шифрования и проверки полномочий. Ср. *электронная коммерция класса business-to-consumer*

ЭЛЕКТРОННАЯ КОММЕРЦИЯ КЛАССА BUSINESS-TO-CONSUMER, электронная коммерция класса B2C [business-to-consumer e-commerce]. *Электронная коммерция* между продавцом и массовым покупателем, основанная на сети Интернет. Э. к. к. B2C предполагает неограниченный доступ и верифицированные платежи по кредитным картам. Веб-сайт Э. к. к. B2C обычно представляет собой виртуальную витрину. Покупатель щелкает мышью на изображении товара, который он желает приобрести, и в появившееся *диалоговое окно* вводит адрес доставки и данные своей кредитной карты. Ср. *электронная коммерция класса business-to-business*

ЭЛЕКТРОННАЯ КОММЕРЦИЯ КЛАССА B2B [business-to-business e-commerce]. То же, что *электронная коммерция класса business-to-business*

ЭЛЕКТРОННАЯ КОММЕРЦИЯ КЛАССА B2C [business-to-consumer e-commerce]. То же, что *электронная коммерция класса business-to-consumer*

ЭЛЕКТРОННАЯ ПОДПИСЬ [digital signature, signature]. То же, что *цифровая подпись*

ЭЛЕКТРОННАЯ ПОЧТА [electronic mail, E-mail]. 1. Набор услуг *компьютерной сети* по пересылке сообщений между ее пользователями. Э. п. является средством быстрой доставки писем, текстов программ, документов и другой подобной корреспонденции. Э. п. кроме своей основной функции отправления и приема почты предоставляет пользователю следующие возможности: 1) получать и хранить сообщения в систематизированном виде, "раскладывая" их в папки; 2) пересылать всю почту на другой компьютер; 3) рассылать копии одного сообщения сразу нескольким лицам с помощью списка рассылки. Отдельные программные пакеты обеспечивают отправителю автоматическое уведомление о получении после того, как посланная почта помещена в почтовый ящик получателя, а также уведомление о прочтении после того, как получатель вывел посланное сообщение на экран своего дисплея. 2. Система *сервисных программ*, обеспечивающая пересылку сообщений по *компьютерным сетям* от пользователя к пользователю, а также чтение и хранение этих сообщений. Почтовая программа пользователя должна быть настроена на почтовый сервер *поставщика сетевых услуг* (провайдера). При передаче сообщения по Э. п. передающий и принимающий компьютеры не обязательно взаимодействуют друг с другом непосредственно. Почта передается от одного компьютера к другому, пока не попадет на почтовый сервер провайдера. При этом она может передаваться из одной компьютерной сети в другую. Пункты соединения между сетями представляют собой компьютеры, называемые *шлюзами*. Они должны изменить формат поступающего сообщения в соответствии с требованиями сети назначения

ЭЛЕКТРОННАЯ СХЕМА [electronic circuit]. Полупроводниковые приборы, соединенные проводниками по определенной схеме для выполнения заданной функции. Э. с. применяются в ЭВМ для генерирования, усиления, преобразования, хранения и передачи электрических сигналов различной формы, мощности и частоты. С помощью этих сигналов представляется обрабатываемая компьютером информация и управляются различные устройства

ЭЛЕКТРОННАЯ ТАБЛИЦА [spreadsheet]. Работающая в *диалоговом режиме* программа *обработки данных*, обеспечивающая взаимодействие с пользователем при помощи выводимых на экран дисплея прямоугольных таблиц. Ячейки таблиц могут содержать строки, числа или формулы, задающие за-

висимость значений одних ячеек от значений других ячеек. Пользователь может просматривать, задавать и изменять содержимое ячеек с немедленным отображением результата этих действий на экране. Э. т. обеспечивают также задание формата отображаемых данных, поиск и сортировку данных, вывод результатов на печать. Кроме того, Э. т. могут дополняться различными *программными средствами* (например, *текстовым процессором*), для того чтобы пользователи могли получать информацию, не выходя из Э. т. Такая Э. т. становится *интегрированной средой*. Наиболее известные интегрированные Э. т.: Works и Excel фирмы Microsoft; Lotus 1-2-3 фирмы Lotus Development; Super Calc фирмы Computer Associates; Quatro Pro фирмы Borland

ЭЛЕКТРОННЫЕ ДЕНЬГИ, цифровые деньги [digital money, digital cash]. Средства расчета, применяемые в *электронной коммерции*. Простейшей формой Э. д. являются зашифрованные информационные пакеты, передаваемые по сети. При этом применяются *шифрование с открытым ключом*, *цифровая подпись* и другие средства защиты. Предполагаемая защищенность от подделки примерно соответствует защищенности долларовых купюр

ЭЛЕКТРОННЫЙ ДИСК, виртуальный диск, псевдодиск [electronic disk, virtual disk, RAM disk]. Особым образом организованная область *оперативной памяти*, внешне проявляющая себя как *магнитный диск*. Скорость *обмена данными* с Э. д. намного больше, чем с обычным. Один из вариантов использования Э. д. таков: если имеется *унаследованное* приложение, рассчитанное на малый объем оперативной памяти и потому хранящее временные данные на диске, то, используя Э. д., можно значительно увеличить производительность работы подобного приложения. В *операционной системе MS-DOS* Э. д. организуется программно, с использованием загружаемого драйвера Э. д. ramdrive.sys. В других ситуациях в настоящее время этот термин не употребляется

ЭЛЕКТРОННЫЙ ДОКУМЕНТ [electronic document]. Совокупность данных в памяти *вычислительной системы*, предназначенная для просмотра и прослушивания человеком с помощью *программных и аппаратных средств*. Э. д. может содержать текстовую, графическую и звуковую информацию. Э. д. может быть оформлен в виде гипертекста

ЭЛЕКТРОННЫЙ ПАРТНЕР [electronic partner, e-partner]. Доверительные стороны, участвующие в *электронной коммерции класса business-to-business*

ЭЛЕМЕНТАРНОЕ ДАННОЕ [data item, item]. То же, что *элемент данных*

ЭЛЕМЕНТ ДАННЫХ, элементарное данное [data item, item]. Поименованная или непоименованная составная часть *данных*, воспринимаемая как неделимый объект обработки. Например, число, *логическое значение*, *строка символов*, переменная, *элемент массива*, *поле данных* и т. п. Э. д. называют и

сложную часть данных, составленную из других частей, если в контексте она рассматривается как неделимый объект обработки, например, выводимая на экран запись

ЭЛЕМЕНТ МАССИВА, индексированная переменная, переменная с индексами [array element, subscripted variable]. *Элемент данных*, образующих *массив*. Составная часть массива, определяемая именем массива и индексами. В традиционных *языках программирования* Э. м. имеет форму *переменной с индексами* `a(i1,i2,...,in)`, где `a` — имя массива, `i1,i2,...,in` — индексы, например, `a(1,2)`, `b(k+1)`, `c2(i,j)`. В некоторых языках (например, в Паскале) вместо круглых скобок применяются квадратные: `a[1,2]`, `b[k+1]`, `c2[i,j]`

ЭЛЕМЕНТ МОДЕЛИ [model element]. Элементарная составляющая модели в *визуальном языке моделирования*. На диаграммах Э. м. изображается с помощью символа, фигуры или линии. Основные Э. м. *унифицированного языка моделирования* UML приведены в табл. Э.1.

Таблица Э.1. *Основные символы унифицированного языка моделирования*

Элемент	Название	Назначение
Вариант использования	Вариант использования [use case]	Последовательность производимых системой действий, доставляющая значимый для некоторого действующего лица результат
Действующее лицо	Действующее лицо [actor]	Сущность, находящаяся вне моделируемой системы и непосредственно взаимодействующая с ней
Класс -Атрибут +Операция ()	Класс [class]	Множество объектов с общими атрибутами и операциями
Объект :Класс	Объект [object]	Экземпляр класса
Компонент	Компонент [component]	Физически заменяемый артефакт, реализующий некоторый набор интерфейсов

Таблица Э.1 *(продолжение)*

Элемент	Название	Назначение
Узел	Узел [node]	Физический вычислительный ресурс
Состояние	Состояние [state]	Период в жизненном цикле объекта, в котором объект выполняет деятельность или ожидает события
Деятельность	Деятельность [activity]	Состояние, в котором выполняется определенная работа, а не просто пассивно ожидается наступление события
Примечание	Примечание [note]	Применяется для размещения произвольных дополнительных сведений
	Развилка [fork]	Разветвление одного потока управления на несколько параллельных потоков управления
	Соединение [join]	Соединение нескольких параллельных потоков управления в один
	Решение [decision]	Разветвление потока управления на несколько альтернативных потоков

Таблица Э.1 *(окончание)*

Элемент	Название	Назначение
	Слияние [merge]	Слияние альтернативных потоков управления

ЭЛЕМЕНТНАЯ БАЗА [integrated circuit]. Набор основных функциональных элементов, на основе которого создаются устройства *вычислительной техники*. Например, электронные лампы, полупроводниковые диоды и триоды, микросхемы, интегральные схемы и т. д. Э. б. является одним из признаков, по которому классифицируются *поколения ЭВМ*

ЭЛЕМЕНТ РАСШИРЕНИЯ [add-on]. Устройство, добавляемое в стандартную *конфигурацию компьютера* (например, плата расширения или внешний жесткий диск) для дополнения или совершенствования его возможностей

ЭЛЕМЕНТ УПРАВЛЕНИЯ [control]. Средство *графического интерфейса пользователя*, предназначенное для ввода/вывода информации и управления работой программы. В настоящее время применяется множество различных Э. у., например, *кнопки*, *флажки*, *переключатели*, *поля*, *списки*, *полосы прокрутки*. На рис. Э.2 приведено диалоговое окно, демонстрирующее пример использования различных Э. у.

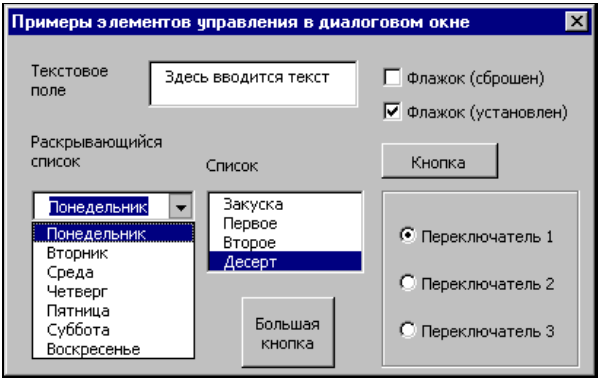

Рис. Э.2. Примеры элементов управления в диалоговом окне

ЭЛЕМЕНТ УПРАВЛЕНИЯ BUY NOW [buy now control]. *Элемент управления ActiveX*, который позволяет покупателю приобретать товары с веб-страницы в Интернете. Когда покупатель щелкает мышью на изображении продукта, появляется *диалоговое окно* для ввода адреса доставки и данных о кредитной карте покупателя

ЭЛЕМЕНТЫ БЛОК-СХЕМ [flowchart symbol]. Стандартные графические символы, из которых состоит *блок-схема программы*. Представляют собой содержащие текст геометрические фигуры и снабжаемые стрелками линии. Основные Э. б.-с. приведены в табл. Э.2. Символ "Пуск-останов" применяется для обозначения начала, конца или прерывания процесса выполнения программы. Внутри эллипса обычно помещают слова "начало", "конец" или "стоп", поясняющие смысл конкретного символа. Символ "Процесс" обозначает операции или группы операций над данными. Выполняемые операции указываются внутри прямоугольника, который имеет один вход — сверху и один выход — вниз. Символ "Решение" служит для обозначения на блок-схемах точек *ветвления программы* и имеет один вход — сверху и соответственно два выхода — вниз и направо или налево. Внутри ромба указывается условие, определяющее, которой из двух *ветвей программы* может быть передано управление. Это условие обычно имеет форму *логического выражения*. Рядом с выходом, соответствующим передаче управления в случае, если указанное логическое выражение истинно, обычно пишут слово "да". Рядом с другим выходом — "нет". Символ "Ввод/вывод" предназначен для указания места ввода или вывода данных, а также для обозначения, какие данные вводятся или какие искомые результаты должны быть выведены. Эту информацию можно написать внутри фигуры. Символ "Предопределенный процесс" изображает использование ранее созданных и отдельно описанных алгоритмов или программ (подпрограмм). Символ "Линия потока" является линией, связывающей между собой отдельные Э. б.-с. и указывающей последовательность определенных этими элементами действий. Если стрелка не указана, то предполагается естественная последовательность сверху вниз и слева направо. Место слияния нескольких линий потока, каждая из которых направлена к одному и тому же Э. б.-с., обозначается точкой. При большой насыщенности блок-схемы элементами или при невозможности разместить блок-схему на одной странице допускаются обрывы линий потока между удаленными друг от друга Э. б.-с. При этом в начале и в конце обрыва должны быть изображены символы "Соединитель", внутри которых цифрой или буквой обозначается разорванная линия. Если пояснение к некоторому Э. б.-с. не помещается внутри символа, то оно размещается на свободном месте схемы и соединяется с поясняемым элементом символом "Комментарий". См. *блок-схема программы* и рис. Б.4

Таблица Э.2. *Основные символы блок-схем*

Элемент	Название	Назначение
Начало	Пуск-останов [terminator]	Применяется для обозначения начала, конца или прерывания процесса выполнения программы
Условие	Решение [decision]	Служит для обозначения мест ветвления программы и имеет один вход — сверху и соответственно два выхода — вниз и направо или налево
Данные	Ввод/вывод [data]	Предназначен для указания места ввода или вывода данных
Название подпрограммы	Предопределенный процесс [predefined process]	Изображает использование ранее созданных и отдельно описанных алгоритмов или программ (подпрограмм)
↓	Линия потока [line connector]	Является линией, связывающей между собой отдельные элементы блок-схемы и указывающей последовательность определенных этими элементами действий
A	Соединитель [connector]	Служит для обозначения разрыва линий
Примечание	Комментарий [comment]	Применяется для размещения дополнительных сведений, не поместившихся в основной элемент

ЭЛЕМЕНТЫ УПРАВЛЕНИЯ ACTIVEX [ActiveX controls]. Специальный вид *элементов управления*, использующий *технологию ActiveX.* Ранее были известны как элементы управления OLE или OCX. Как и другие элементы управления, Э. у. ActiveX имеют набор свойств, с помощью которых можно изменить их внешний вид, определить процедуры *обработки событий* и т. п. Э. у. ActiveX являются динамическими модулями и могут использоваться для расширения функциональных возможностей любого приложения или *языка программирования*, поддерживающего *связывание и внедрение объектов*

ЭМУЛЯТОР [emulator]. *Программа* или *микросхема*, позволяющая осуществить *эмуляцию*

ЭМУЛЯЦИЯ [emulation]. Точное выполнение *вычислительной машиной* программы или ее части, записанной в *системе команд* другой ЭВМ. При этом на обоих компьютерах при одинаковых *входных данных* результаты долж-

ны быть одинаковыми. Э. осуществляется с помощью *эмуляторов* — специальных программ или микросхем, выполняющих каждую команду *исходной программы* посредством одной или нескольких команд машины, на которой происходит Э. С помощью программной Э. можно имитировать отсутствующие в *вычислительной системе аппаратные средства*. Например, для выполнения программ, производящих вычисления над *числами с плавающей точкой*, на компьютерах без *математического сопроцессора* необходим программный эмулятор математического сопроцессора, входящий в состав компилятора программ

Я

ЯДРО [kernel]. То же, что *ядро операционной системы*

ЯДРО ОПЕРАЦИОННОЙ СИСТЕМЫ, ядро [kernel]. Часть *операционной системы*, постоянно находящаяся в *оперативной памяти*. Я. о. с. определяет очередность выполнения процессов *центральным процессором*, распределяет оперативную память и другие ресурсы *вычислительной системы*, обрабатывает прерывания и иные *исключительные ситуации*. См. *режим ядра*

ЯДРО СУБД [database engine]. Набор системных программ, входящих в состав СУБД и осуществляющих операции манипулирования данными, т. е. сохранение, поиск и обновление данных в *базе данных*. Какие именно операции над данными поддерживает Я. СУБД, определяется конкретной СУБД и используемой *моделью данных*. Функции Я. СУБД, как правило, вызываются не напрямую, а через *интерфейс пользователя* или из программы на *языке программирования* соответствующей СУБД. См. *система управления базой данных*

ЯЗЫК АССЕМБЛЕРА [assembly language]. То же, что *автокод*

ЯЗЫК ВИЗУАЛЬНОГО МОДЕЛИРОВАНИЯ [visual modeling language]. *Формальный язык*, имеющий *графическую нотацию* и предназначенный для визуального моделирования и спецификации *программных и аппаратных средств*. Примером может служить *язык спецификации и определения* SDL. В настоящее время наибольшее распространение получил *унифицированный язык моделирования* UML

ЯЗЫК ЗАПРОСОВ SQL [Structured Query Language (SQL)]. Специализированный язык общения пользователя с *информационными системами*. SQL используется при организации запросов, обновлении и управлении *реляционными базами данных*. Он берет начало от исследовательского проекта фирмы IBM, по которому в 1970 г. был разработан структурированный английский язык запросов (Structured English QUEry Language, SEQUEL). В настоящее время SQL принят в качестве фактического стандарта в *системах управления*

базами данных. Не будучи *языком программирования*, как, например, Си или Паскаль, SQL может применяться при составлении интерактивных запросов или встраиваться в *прикладную программу* в виде команд. Стандарт SQL также содержит средства определения, управления и *защиты данных*

ЯЗЫК МОДЕЛИРОВАНИЯ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ [Virtual Reality Modeling Language (VRML)]. Формальный язык, в своих основных принципах сходный с языком HTML. Я. м. в. р. используется для создания трехмерных изображений, т. н. виртуальных миров, в которых пользователь может перемещаться и взаимодействовать с объектами. Первая версия языка, VRML 1.0, позволяла создавать статические миры, в которых пользователь мог только перемещаться. (Эта статичность породила в свое время шутливую расшифровку аббревиатуры VRML — Virtual Reality Museum Language — Язык музея виртуальной реальности.) Версия VRML 2.0 включила в себя определение реакции объектов на события и их перемещение. Таким образом, не только пользователь может взаимодействовать с объектами виртуального мира, но и они могут взаимодействовать друг с другом

ЯЗЫК ОПИСАНИЯ ИНТЕРФЕЙСОВ, язык IDL [interface definition language (IDL)]. *Формальный язык*, предназначенный для описания *интерфейсов прикладного программирования*. Используется для формальной спецификации интерфейсов программных систем

ЯЗЫК ОПИСАНИЯ СТРАНИЦ [page description language (PDL)]. *Язык программирования*, применяемый для описания данных, выводимых на принтер или на экран. Например, *язык PostScript* или *язык PCL*. Я. о. с. подразумевает *интерпретацию* программ при выводе данных на *графическое устройство* и по этому признаку противопоставляется *языку разметки*

ЯЗЫК ПРОГРАММИРОВАНИЯ [programming language]. *Формальный язык*, предназначенный для связи человека с *вычислительной машиной*. На Я. п. компьютеру задаются информация (данные) и алгоритм *обработки данных* (программа). Процессор ЭВМ непосредственно воспринимает программу, представленную на *машинном языке*, программирование на котором весьма неудобно для человека. Поэтому разработаны *языки программирования высокого уровня*, существенно упрощающие процесс программирования и не зависящие от архитектуры конкретной ЭВМ. Также существуют *проблемно-ориентированные языки программирования*, специально приспособленные для решения задач определенного класса. С развитием *интеллектуальных систем программирования* в качестве Я. п. станет употребляться *естественный язык*. См. *машинно-ориентированный язык, машинно-независимый язык, визуальный язык программирования*

ЯЗЫК ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ [high-level programming language]. *Язык программирования*, в который введены элементы, допускающие описание задачи в наглядном, легко воспринимаемом виде,

упрощающие и автоматизирующие процесс программирования. Управляющие конструкции и *структуры данных* Я. п. в. у. отражают естественные для человека понятия, а не *архитектуру вычислительной системы*. Поэтому программа, составленная на Я. п. в. у., сначала транслируется самим компьютером на *машинный язык*, а затем выполняется. Я. п. в. у. определяется заданием алфавита (или словаря исходных символов), точным описанием правил построения предложений (синтаксис) и правил, определяющих смысл предложений (семантика). В алфавит Я. п. в. у. могут входить буквы, цифры, математические символы и даже так называемые *ключевые слова*, например, `if` (если), `then` (тогда), `else` (иначе) и т. п. Из исходных символов по правилам синтаксиса строятся предложения, обычно называемые операторами, например, `if x<1 then y:=x+1 else y:=x-1;`. Приведенный оператор на языке Паскаль имеет следующий смысл. Если значение переменной `x` меньше 1, тогда значение переменной `y` нужно вычислять по формуле `y = x + 1`, в противном случае (если `x` ≥1) следует воспользоваться формулой `y = x - 1`. Наиболее распространенные Я. п. в. у.: Паскаль, Фортран и Си

ЯЗЫК ПРОГРАММИРОВАНИЯ СЦЕНАРИЕВ, язык сценариев [scripting language]. *Язык программирования высокого уровня*, предназначенный для программирования *сценариев*. Как правило, в Я. п. с. не включают сложных структур данных и операций манипулирования с объектами низкого уровня, поэтому Я. п. с. проще и меньше универсального языка программирования высокого уровня. Например, язык *командных файлов* является Я. п. с. См. пример в статье *командный файл*. См. *язык сценариев JavaScript, язык сценариев VBScript*

ЯЗЫК ПРОГРАММИРОВАНИЯ JAVA [Java programming language]. Разработан компанией Sun Microsystems. Это современный *объектно-ориентированный язык программирования высокого уровня*, синтаксис которого напоминает синтаксис *Си++*. В целях повышения надежности программ из Я. п. J. специально исключены некоторые конструкции, являющиеся потенциальным источником ошибок, например, указатели. Транслятор Java на основе исходной программы генерирует т. н. *байт-код*, формат которого определен в спецификации *виртуальной машины Java* (JVM). Полученный байт-код интерпретируется с помощью конкретной реализации JVM, и таким образом происходит выполнение программы. Поскольку реализации JVM существуют для всех основных платформ, программы на Java мобильны (переносимы) и могут исполняться практически на любом компьютере. Благодаря этому свойству Java широко используется для программирования приложений для *Интернета*

ЯЗЫК РАЗМЕТКИ [markup language]. *Формальный язык*, предназначенный для описания структуры и содержания *электронных документов*. К наиболее известным Я. р. относятся *языки SGML*, *HTML*, *XML*. Я. р. не предписывает способ интерпретации своих конструкций, возлагая это на приложение, ин-

терпретирующее документ, и тем самым обеспечивает независимость размеченного электронного документа от приложений. По этому признаку Я. р. противопоставляются интерпретируемым командным языкам. Ср. *TeX, формат RTF*

ЯЗЫК СПЕЦИФИКАЦИИ И ОПРЕДЕЛЕНИЯ, язык SDL [specification and definition language (SDL)]. *Язык визуального моделирования*, разработанный *Международным союзом телекоммуникаций* и предназначенный для описания и спецификации программного и аппаратного обеспечения. В основе данного языка лежат понятия *конечного автомата* и структурной декомпозиции

ЯЗЫК СЦЕНАРИЕВ [scripting language]. То же, что *язык программирования сценариев*

ЯЗЫК СЦЕНАРИЕВ JAVASCRIPT, язык JavaScript [JavaScript]. *Язык программирования сценариев*, созданный компанией Netscape. Является совместимым с *языком программирования Java*, но более простым. Широко применяется для написания встроенных в тело *веб-страницы сценариев*, которые интерпретируются при просмотре страницы. См. *сервер сценариев Windows*

ЯЗЫК СЦЕНАРИЕВ VBSCRIPT, язык VBScript [Microsoft Visual Basic scripting edition, VBScript]. *Язык программирования сценариев*, созданный компанией Microsoft. Представляет собой подмножество *языка Visual Basic*, предназначенное для написания встроенных в тело *веб-страницы сценариев*. Поддерживается браузером MS Internet Explorer. Является альтернативой *языку сценариев JavaScript*. См. *сервер сценариев Windows*

ЯЗЫК УПРАВЛЕНИЯ ПЕЧАТЬЮ, язык PCL [printer control language (PCL)]. *Язык описания страниц*, разработанный фирмой Hewlett-Packard для принтеров этой фирмы (LaserJet, DeskJet и RuggedWriter). Так как эти принтеры доминируют на рынке, язык PCL стал фактическим стандартом. Команды Я. у. п. задают число копий, разрешение печати, а также *размер страницы*, шрифт, интерлиньяж, кегль и т. п.

ЯЗЫК HPGL [Hewlett Packard graphics language (HPGL)]. *Язык программирования*, разработанный в компании Hewlett-Packard для обработки и хранения графических изображений. Первоначально он предназначался для изображений, выводимых на графопостроители

ЯЗЫК HPF [High Performance Fortran (HPF)]. То же, что *высокопроизводительный Фортран*

ЯЗЫК HTML [HyperText Markup Language (HTML)]. Язык разметки *гипертекста*. Стандартный язык, используемый во *Всемирной паутине WWW* для создания и публикации *веб-документов*. Я. HTML поддерживает основные функции, необходимые для создания гипермедиа-документов: форматирование текста, возможности включения графики, подключения видео и звука, создания гиперссылок, списков, запуска программ поиска

информации в WWW. Существуют программы-редакторы, предназначенные для создания документов на Я. HTML, и программы-конверторы для преобразования уже готовых документов в HTML-формат. Например, Microsoft FrontPage, Netscape Navigator Gold и Macromedia Dreamweaver являются наиболее распространенными HTML-редакторами для *операционных систем Windows*, а текстовый процессор Microsoft Word обладает встроенным HTML-конвертором

ЯЗЫК JAVASCRIPT [JavaScript]. То же, что *язык сценариев JavaScript*

ЯЗЫК PCL [printer control language (PCL)]. То же, что *язык управления печатью*

ЯЗЫК PERL [practical extraction and report language (PERL)]. *Язык программирования*, часто применяющийся в *сценариях* (скриптах CGI). Программы на Я. Perl исполняются методом интерпретации. См. *интерфейс CGI, язык программирования сценариев*

ЯЗЫК POSTSCRIPT [PostScript]. *Язык описания страниц* для печатающих устройств. Впервые был реализован фирмой Adobe Systems в 1982 г. Предлагает гибкую методику обработки контурных масштабируемых шрифтов и высококачественных графических изображений. Команды Я. PostScript включаются непосредственно в документ и выполняются интерпретатором PostScript, которым должен быть снабжен принтер. См. *упакованный формат PostScript-файлов*

ЯЗЫК SDL [specification and definition language (SDL)]. То же, что *язык спецификации и определения*

ЯЗЫК SGML, стандартный обобщенный язык разметки [standard generalized markup language (SGML)]. Утвержденный *Международной организацией по стандартизации* (ISO) набор правил и шаблонов для описания содержания, структуры и формата любых *электронных документов*, независимо от конечного способа их представления (на экране или в печатном виде). На основе Я. SGML разработаны *язык HTML* и *язык* XML

ЯЗЫК UML [UML]. То же, что *унифицированный язык моделирования*

ЯЗЫК VBA [Visual Basic for Applications (VBA)]. То же, что *Visual Basic для приложений*

ЯЗЫК VBSCRIPT [VBScript]. То же, что *язык сценариев VBScript*

ЯЗЫК VISUAL BASIC [Visual Basic (VB)]. Язык программирования высокого уровня, разработанный корпорацией Microsoft на основе языка *Бейсик*. Является развитым языком *объектно-ориентированного программирования*. См. *язык сценариев VBScript, Visual Basic для приложений*

ЯЗЫК XML, расширяемый язык разметки [extensible markup language (XML)]. Предложенный *Консорциумом World Wide Web* язык обмена данными и форматирования *документов WWW*. Как и *язык HTML*, Я. XML является подмножеством *языка SGML*. Важное отличие состоит в том, что набор *тегов* не является фиксированным (как в HTML), а может быть описан с помощью специального языка DTD (Document Type Definition). Таким образом, Я. XML пригоден для описания структуры и содержания любых *электронных документов* (а не только веб-страниц) и позволяет осуществить независимый от приложений и *компьютерных платформ* обмен данными

ЯКОРЬ [anchor]. Видимая часть *гиперссылки* на *веб-странице*. Признаком Я. гипертекстовой ссылки является изменение внешнего вида *указателя мыши* (на "указательный палец"), когда тот проходит над ссылкой

ЯРКОСТЬ [brightness]. Количественная характеристика *цвета* в *машинной графике*. Одна из трех характеристик в *цветовой модели* HSB (тон-насыщенность-яркость) — выраженная в процентах; доля белого цвета в цветовом оттенке. См. *цвет*, *цветовая модель*

ЯЧЕЙКА ПАМЯТИ [cell]. 1. Имеющий определенный адрес элемент *оперативной памяти* или регистр, содержимое которого выбирается *центральным процессором* за одно обращение. Во многих ЭВМ Я. п. соответствует одной *машинной команде* или ее одному типовому операнду. В большинстве *персональных компьютеров* ячейка *оперативной памяти* содержит один байт информации. 2. Элемент *памяти ЭВМ*, отведенный для хранения одного данного. 3. Элемент данных *электронной таблицы*

C

CISC-КОМПЬЮТЕР [complex instruction set computer (CISC)]. То же, что *компьютер со сложной системой команд*

CMOS-ПАМЯТЬ [CMOS memory, CMOS RAM]. То же, что *КМОП-память*

CMOS-СТРУКТУРА [complementary metal-oxide-semiconductor (CMOS)]. То же, что *комплементарная структура металл-оксид-полупроводник*

COOKIE-ИДЕНТИФИКАЦИЯ [cookie identification]. Метод идентификации, при котором "личность" пользователя выясняется не по его имени, а при помощи *глобально-уникального идентификатора* (GUID). GUID хранится в *cookie-файле*, используемом браузером пользователя

COOKIE-КОД [cookie]. Постоянный идентификационный код, назначаемый пользователю и позволяющий контролировать посещения пользователем веб-сайта. См. *cookie-идентификация*

COOKIE-ФАЙЛ [cookie]. *Файл*, создаваемый сервером на компьютере клиента и предназначенный для использования сервером. Например, сервер Интернета создает на компьютере клиента С.-ф., предназначенный для идентификации клиента. В файле хранятся данные о клиенте, включая информацию о количестве посещений клиентом веб-страниц и о действиях, предпринятых посетителем. Эта информация автоматически передается обратно с клиента на сервер при всех соединениях клиента с сервером. См. *cookie-идентификация, cookie-код*

CTI-плата [CTI board]. Устанавливаемая внутри компьютера *плата* с телефонным оборудованием, обеспечивающим создание *телефонных приложений*, имеющих прямой доступ к данным, поступающим по *голосовым каналам*. Это позволяет автоматически обрабатывать звонки с помощью компьютерных программ. В отличие от *голосового модема* CTI-п. имеет возможность коммутации внутренних линий пользователя, например, можно подключить факс или автосекретаря и даже переложить на компьютер все основные функции автоматической телефонной станции для внутренней *телефонной сети* организации или учреждения. См. *компьютерно-телефонная интеграция*

I

IP-АДРЕС [IP address]. То же, что *интернет-адрес*

IP-ТЕЛЕФОН [IP-phone, IP-telephone]. Телефонный аппарат, подключенный непосредственно к *сети передачи данных* и выполняющий функции терминала *интернет-телефонии*. См. *компьютерно-телефонная интеграция*, *сеть H.323*

IP-ТЕЛЕФОНИЯ [IP-telephony]. То же, что *интернет-телефония*

J

JAVA-АППЛЕТ [Java applet]. То же, что *джава-апплет*

JAVA-СЦЕНАРИЙ [Java script]. То же, что *джава-сценарий*

M

MAC-АДРЕС [MAC address]. *Физический адрес* устройства, присоединенного к *компьютерной сети*. Уникальное 48-битное число, присваиваемое *сетевой карте* ее производителем. MAC-а. используются для однозначной идентификации устройства в локальных сетях, управляемых *протоколами TCP/IP*. См. *управление доступом к среде*

MIMD-КОМПЬЮТЕР [MIMD-computer]. Согласно *классификации Флинна* — *компьютер*, имеющий MIMD-архитектуру (Multiple Instruction stream/Multiple Data stream) — множественный *поток команд* и множественный *поток данных*. В таком компьютере есть несколько устройств обработки команд, объединенных в единый комплекс и работающих каждое со своим потоком команд и данных. К этому классу компьютеров относится большинство параллельных *вычислительных систем*, *рабочие станции* с несколькими процессорами, кластеры рабочих станций и т. д. Ср. *MISD-компьютер, SIMD-компьютер, SISD-компьютер*

MISD-КОМПЬЮТЕР [MISD-computer]. Согласно *классификации Флинна* — *компьютер*, имеющий MISD-архитектуру (Multiple Instruction stream/Single Data stream) — множественный *поток команд* и одиночный *поток данных*. Такие компьютеры должны состоять из многих процессоров, обрабатывающих один и тот же поток данных. Некоторые теоретики считают, что таких компьютеров нет. Другие относят к этому классу конвейерные машины, составленные из *процессорных элементов*, расположенных в узлах сети, представ-

ляющей регулярную решетку. Роль ребер в ней играют межпроцессорные соединения. Все процессорные элементы управляются общим тактовым генератором. Каждый процессорный элемент получает данные от своих соседей, выполняет одну команду и передает результат соседям. Ср. *MIMD-компьютер, SIMD-компьютер, SISD-компьютер*

MPI-ПРОГРАММА [MPI-program]. *Программа параллельных вычислений*, написанная в соответствии с технологией *"интерфейс передачи сообщений"* (технологией MPI)

MPMD-МОДЕЛЬ [multiple progam multiple data (MPMD)]. Частный случай *модели передачи сообщений*, в рамках которой программа порождает фиксированный набор процессов, причем каждый процесс выполняется на своем процессоре. В этих процессах могут выполняться разные программы. Ср. *SPMD-модель*

MX-ЗАПИСЬ [MX record]. То же, что *имя пункта обработки почтовых сообщений*

P

PIN-КОД [personal identification number]. То же, что *ПИН-код*

PVM-ПРОГРАММА [PVM-program]. *Программа параллельных вычислений*, написанная в соответствии с технологией *параллельная виртуальная машина* (технологией PVM)

R

RLL-КОДИРОВАНИЕ [RLL encoding]. То же, что *кодирование с ограничением длины поля записи*

S

SIMD-КОМПЬЮТЕР [SIMD-computer]. Согласно *классификации Флинна — компьютер*, имеющий SIMD-архитектуру (Single Instruction stream/Multiple Data stream) — одиночный поток команд и множественный поток данных. SIMD-к. состоит из одного *командного процессора* (контроллера) и нескольких *процессорных элементов*, выполняющих *обработку данных*. Контроллер выбирает, анализирует и выполняет команды. Если команда является операцией над данными, то эта операция выполняется на одном или нескольких процессорных элементах под руководством контроллера. Это позволяет обра-

ботать одной командой сразу множество значений, например, произвести поэлементное сложение двух массивов. Ср. *SISD-компьютер, MIMD-компьютер, MISD-компьютер*

SPMD-МОДЕЛЬ [single progam multiple data (SPMD)]. Частный случай *модели передачи сообщений*, в рамках которой для всех *параллельных вычислительных процессов* используется один и тот же код. Это возможно, если исходная проблема разбивается на несколько частей-подзадач, каждая из которых может быть решена применением одного и того же алгоритма к различным фрагментам набора данных. Однотипные параллельные процессы запускаются выделенной управляющей подзадачей — *мастер-программой*. Ср. *MPMD-модель*

SISD-КОМПЬЮТЕР [SISD-computer]. Согласно *классификации Флинна* — *компьютер*, имеющий SISD-архитектуру (Single Instruction stream/Single Data stream) — одиночный поток команд и одиночный поток данных. SISD-к. — это классический последовательный компьютер, выполняющий в каждый момент времени только одну операцию над одним элементом данных. При этом для увеличения скорости обработки команд и скорости выполнения арифметических операций может применяться *конвейерная обработка*. Большинство современных бытовых персональных ЭВМ являются SISD-к. Ср. *SIMD-компьютер, MISD-компьютер, MIMD-компьютер*

SMP-КОМПЬЮТЕР [SMP-computer]. 1. То же, что *симметричный многопроцессорный компьютер*. 2. Альтернативное название любых компьютеров с общей памятью. При этом аббревиатура SMP расшифровывается как Shared Memory Processors — процессоры с общей памятью

U

UMA-КОМПЬЮТЕР [UMA-computer]. То же, что *компьютер с однородным доступом к памяти*

V

VISUAL BASIC ДЛЯ ПРИЛОЖЕНИЙ, язык VBA [Visual Basic for applications (VBA)]. Подмножество *языка Visual Basic*, реализованное в составе Microsoft Office. Используется многими приложениями в качестве стандартного *языка программирования*. Программы на VBA исполняются не сами по себе, а в среде того приложения Microsoft Office, в котором они были созданы. Тем не менее программы на VBA легко переносятся на Visual Basic, что позволяет после

минимальной переделки использовать такие программы в качестве самостоятельных приложений

VLIW-ПРОЦЕССОР [VLIW-processor]. То же, что *процессор со сверхдлинным командным словом*

W

WEB-ПАПКА [Web folder]. То же, что *веб-папка*

WEB-СЕРВЕР, интернет-сервер [Web server]. То же, что *веб-сервер*

Англо-русский указатель терминов

A

Abbreviation — аббревиатура

Abend — аварийный останов

Abnormal end, abend — аварийное завершение

Abrupt end — аварийный останов

Absolute address — абсолютный адрес

Abstract base class — базовый абстрактный класс

Abstract class — абстрактный класс

Abstract data type — абстрактный тип данных

Abstraction — абстракция

Accelerator key — командная клавиша

Access control — управление доступом

Access keys — клавиши быстрого доступа

Access path — путь доступа

Access permission — право доступа

Access right — право доступа

Access time — время доступа

Access — доступ

Accessory — аксессуары

Account — учетная запись пользователя, бюджет пользователя

Account lockout — блокировка учетной записи, блокировка бюджета

Account policy — политика учетных записей

Accumulator — сумматор

ACK (ACKnowledgement) — подтверждение приема, сигнал подтверждения приема, сигнал ACK

Acknowledgement (ACK) — подтверждение приема, сигнал подтверждения приема, сигнал ACK

ACM (Association for Computing Machinery) — ассоциация вычислительной техники

A-constant — адресная константа

Action — действие

Activation — активация, активизация

Active cell — активная ячейка

Active desktop — активный рабочий стол

Active directory service interface (ADSI) — интерфейс ADSI

Active disc — активный диск

Active File — открытый файл, активный файл

Active program — активная программа

Active window — активное окно

ActiveX controls — элементы управления ActiveX

ActiveX — технология ActiveX

Activity diagram — диаграмма деятельности

Activity graph — граф деятельности

Activity — деятельность

Actor — действующее лицо

Actual parameter — фактический параметр

Ada — Ада

Adapter — адаптер

Adaptive Differential Pulse Code Modulation (ADPCM) — адаптивная дифференциальная импульсно-кодовая модуляция

Adaptive system — адаптивная система

Adaptivity — адаптивность

ADC (Analog-to-Digital converter) — аналого-цифровой преобразователь (АЦП)

Add-in memory — дополнительная память

Additive color — аддитивный цвет

Add-on module — добавляемый модуль, добавочный модуль, надстройка

Add-on — элемент расширения

Address — адрес

Address Bus — адресная шина

Address class — класс адресов

Address constant — адресная константа

Address reference — адресная ссылка

Address register — регистр адреса, адресный регистр

Address relocation — настройка адресов

Addressing — адресация

Administrator — администратор

ADPCM (Adaptive Differential Pulse Code Modulation) — адаптивная дифференциальная импульсно-кодовая модуляция, стандарт ADPCM

ADSI (Active directory service interface) — интерфейс ADSI

Advanced Program-to-Program Communications (APPC) — протокол APPC

Agent — агент

Aggregation — агрегация

Air brush — аэрограф

Alarm — аварийный сигнал

Alert message — предупреждение

Alert — предупреждение

Algol — Алгол

Algorithm Branch — ветвь алгоритма

Algorithm error — ошибка алгоритма

Algorithm — алгоритм

Algorithmic language — алгоритмический язык

Alias — псевдоним

Alignment — выравнивание, выключка

Allocate memory — распределенная память

Alpha channel — альфа-канал

Alpha release — альфа-версия

Alpha testing — альфа-тестирование

Alphanumeric character — текстовый символ

ALU (Arithmetic and Logical Unit) — арифметико-логическое устройство (АЛУ), арифметическое устройство (АУ)

American National Standards Institute (ANSI) — Американский национальный институт стандартов (ANSI)

American Standard Code Information Interchange (ASCII) — Американский стандартный код обмена информацией

Analog computer — аналоговая вычислительная машина (АВМ)

Analog data form — аналоговая форма данных

Analog device — аналоговое устройство

Analog signal — аналоговый сигнал, непрерывный сигнал

Analog-to-Digital converter (ADC) — аналого-цифровой преобразователь (АЦП)

Ancestor — предок

Anchor — якорь

Animation Path — анимационная последовательность

Animation — анимация

ANSI (American National Standards Institute) — Американский национальный институт стандартов

Anti-aliasing — сглаживание

Antivirus program — антивирусная программа

AOP (Aspect Oriented Programming) — аспекто-ориентированное программирование

API (Application Programming Interface) — интерфейс прикладного программирования

APPC (Advanced Program-to-Program Communications) — протокол APPC

Applet — апплет, аплет

AppleTalk — сеть APPLETALK

Application independence — независимость приложений

Application layer — уровень приложений, прикладной уровень

Application program package — пакет прикладных программ (ППП)

Application program — прикладная программа

Application programming Interface (API) — интерфейс прикладного программирования

Application window — окно приложения

Application — приложение

Architecture — архитектура

Archivation — архивирование, архивация

Archivator — архиватор

Archive — архив

Archived File — архивный файл

Argument — аргумент

Argument list — список аргументов, список фактических параметров

Arithmetic and Logical Unit (ALU) — арифметико-логическое устройство (АЛУ), арифметическое устройство (АУ)

Arithmetic cycle — арифметический цикл

Arithmetic expression — арифметическое выражение

Arithmetic IF statement — арифметический условный оператор

Arithmetic operation character — знак арифметической операции

Arithmetic operations — арифметические операции

Arithmetic operator — арифметический оператор, знак арифметической операции

Arithmetic shift — арифметический сдвиг

Arithmetic Unit — арифметическое устройство (АУ)

Arithmetical cycle — арифметический цикл

Arithmetical operations — арифметические операции

Arithmetical Unit — арифметическое устройство (АУ)

Array — массив

Array declaration — описание массива

Array element — элемент массива, индексированная переменная, переменная с индексами

Array instruction — векторная команда

Array processor — векторный процессор

Array processor — матричный процессор

Artifact — артефакт

Artificial intelligence — искусственный интеллект

Artistic text — заголовочный текст, фигурный текст

ASCII (American Standard Code Information Interchange) — Американский стандартный код обмена информацией

ASCII code (American Standard Code Information Interchange) — код ASCII

Aspect Oriented Programming (AOP)] — аспекто-ориентированное программирование

Assembler — ассемблер

Assembly instruction — команда ассемблера

Assembly language — язык ассемблера

Assignment statement — оператор присваивания

Association for Computing Machinery (ACM) — ассоциация вычислительной техники

Association — ассоциация

Asynchronous connection — асинхронная связь

Asynchronous transmission — асинхронная передача

Atomic — атомарный

Attached File — вложенный файл, прикрепленный файл

Attachment — вложение

Attribute — атрибут

Audio data — аудиоданные

Audio information — аудиоинформация

Audio track — звуковая дорожка

Audio Video Interleave (AVI) — формат AVI

Audiovisual — аудиовизуальный

Authentication — аутентификация

Authoring language — авторский язык

Authoring system — авторская система, система автоматизации авторской работы

Authorization — авторизация

Autocode — автокод

Autoexec File — файл автозапуска

Automata — автомат

Automated Control system — автоматизированная система управления (АСУ)

Automated system — автоматизированная система

Automated training system — автоматизированная обучающая система (АОС)

Automated working place — автоматизированное рабочее место (АРМ)

Automatic control system — система автоматического управления

Automatic program synthesis — автоматический синтез программ

Automatic system — автоматическая система

Automation — автоматизация

Automaton — конечный автомат

Autonomy device — автономное устройство

Autonomy program — автономная программа

Autonomy unit — автономное устройство

Autosave — автоматическое сохранение, автосохранение

AVI (Audio Video Interleave) — формат AVI

AVI-File — файл AVI

B

Back-end processor — постпроцессор

Background task — фоновая задача

Backing storage — внешняя память, внешнее запоминающее устройство (ВЗУ)

Backplane — магистраль

Backtracking — откат, поиск с возвратами, обратное прослеживание

Backup — резервная копия

Backus-Naur form (BNF) — формальный язык Бэкуса—Наура, нормальная форма Бэкуса—Наура (БНФ)

Band — зона

Bandwidth control — управление пропускной способностью

Bandwidth — пропускная способность, ширина полосы пропускания

Bank — банк, банк данных, страница памяти

Banner — баннер, банер

Banner advertising — баннерная реклама

Bar chart — гистограмма

Bar graph — гистограмма

Base address — базовый адрес

Base register — базовый регистр, регистр базы

Basic Input/Output System (BIOS) — базовая система ввода/вывода

Basic — Бе(э)йсик

Basket — корзина

Batch — пакет

Batch File — пакетный файл

Batch mode — пакетный режим

Batch processing — пакетная обработка

Baud — бод

Beginner's all-purpose symbolic instruction code — Бе(э)йсик

Behavior — поведение

Benchmark — бенчмарка

Beowulf cluster — беовульф-кластер

Bernoulli disk — диск Бернулли

Beta testing — бета-тестирование

Beta-release — бета-версия

Beta-version — бета-версия

Bezier curve — кривая Безье

Billing — биллинг, тарификация

Binary code — двоичный код

Binary digit — двоичный разряд, бит

Binary number system — двоичная система счисления

Binary number — двоичное число

Binary operation — бинарная операция

Binary-coded representation — двоичное представление

Binding — связывание, привязка протокола

BIOS (Basic Input/Output System) — базовая система ввода/вывода

Bit — бит, двоичный разряд

Bit depth — битовая глубина цвета, глубина цвета, разрядность цвета

Bit rate — скорость передачи данных

Bitmap editor — редактор изображений, графический редактор

Bitmap File — растровый файл

Bitmap image — растровое изображение, растровый образ

Bitmap — битовый массив, битовая карта, растр, растровый образ

Bitmapped font — растровый шрифт

Bit-mapped graphics — растровая графика

Block — блок

Blocking — блокирование

Blue screen — "синий экран"

BMP (сокращение от англ. BitMaP) — формат BMP

BMP File — файл BMP

BNF (Backus—Naur Form) — формальный язык Бэкуса—Наура, нормальная форма Бэкуса—Наура (БНФ)

Bold — полужирный шрифт

Boldface — полужирный шрифт

Bookmark — закладка

Boolean algebra — булева алгебра

Boolean expression — булево выражение

Boolean function — логическая функция, булева функция

Boolean operation — булева операция

Boolean value — булево значение

Bootstrap protocol (BOOTP) — протокол начальной загрузки, протокол BOOTP

Border — граница окна, обрамление, окантовка

Bot — робот поисковой системы

Bottom up programming — восходящее программирование

Bound pair — граничная пара

Box — поле

Branch instruction — команда ветвления, команда передачи управления, команда перехода

Branching — ветвление программы, переход

Breakpoint — точка прерывания

B-register — индексный регистр, регистр индекса, регистр B, регистр BX

Bridge — мост

Bridge-router — мост-маршрутизатор, брутер

Brightness — яркость

Broadcast message — широковещательное сообщение

Broadcast network — широковещательная компьютерная сеть, широковещательная сеть

Broadcasting — широковещание

Broken link — разрушенная ссылка

B-router, brouter (Bridge-router) — мост-маршрутизатор, брутер

Browser — браузер, Web-браузер, браузер Windows NT

Buffer memory — буферная память

Buffer — буфер

Buffering — буферизация

Build-in computer — встроенная ЭВМ

Build-in procedure — встроенная процедура

Build-in type — встроенный тип

Built-in function — встроенная функция

Bullet — буллит, маркер

Bulleted list — маркированный список

Bus — шина, системная шина

Bus network — шинная сеть, сеть шинной топологии

Business-to-Business e-commerce — электронная коммерция класса business-to-business, электронная коммерция класса B2B

Business-to-consumer e-commerce — электронная коммерция класса business-to-consumer

Button — кнопка

Buy now control — элемент управления buy now

Byte — байт (б)

Bytecode — байт-код

C

C — Си

C++ — Си++

Cache — кэш, кэш-память

Cache flushing — сброс кэш-памяти, сброс кэша

Cache memory — кэш-память, кэш

Cache miss — ошибка кэша, промах кэша

Caching — кэширование

CAD (Computer-Aided Design) — автоматизированное проектирование, система автоматизированного проектирования (САПР)

Calculator — калькулятор

Calculation process — вычислительный процесс

Calibration — калибровка

Call — вызов подпрограммы, обращение к подпрограмме

Call control — управление соединениями

Callback — возвратный вызов, обратный вызов

Callback function — функция обратного вызова

Called subroutine — вызываемая подпрограмма

Calling procedure statement — оператор вызова процедуры, оператор обращения к процедуре

Calling program — вызывающая программа

Callout — выноска

Canonical notation — каноническая нотация языка UML

Capability Maturity Model (CMM) — модель зрелости возможностей

Capacity — объем памяти, разрядность

Card — перфокарта, перфорационная карта, карта

Careful write — точная запись

Caret — каре

Carrier sense multiple access with collision detection (CSMA/CD) — метод множественного доступа с контролем несущей и разрешением конфликтов, метод CSMA/CD

Cartridge — картридж

CASE (Computer-Aided Software Engineering) — автоматизированная разработка програм

Case statement — оператор выбора, переключатель

CASE-tool — автоматизированная разработка програм

Cast — приведение типов, преобразование типов

Catalogization — каталогизация

Catastrophic error — неисправимая ошибка, фатальная ошибка

CBT (Computer Based Training system) — обучающее приложение к программному продукту, автоматизированная подготовка

CCITT — Международный консультативный комитет по телеграфии и телефонии

CD (Compact Disk) — компакт-диск

CD drive (Compact Disk drive) — привод CD-ROM

CD-ROM (Compact Disk Read Only Memory) — компакт-диск

CD-ROM drive(Compact Disk Read Only Memory drive) — привод CD-ROM

Cell — ячейка памяти

Central Processing Unit (CPU) — центральный процессор (ЦП)

CERN (франц: Conseil Europeen pour la Recherche Nucleaire) — Европейская лаборатория ядерных исследований (ЦЕРН)

CGI (Common Gateway Interface) — интерфейс CGI

CGI script (Common Gateway Interface script) — сценарий CGI, программа CGI

Challenge/response authentication — аутентификация с запросом и ответом, аутентификация "запрос—ответ"

Channel — канал

Character — символ

Character alt-input — Алт-ввод символов

Character constant — символьная константа, текстовая константа

Character encoding — кодирование символов

Character generator — знакогенератор

Character mode — текстовый режим

Character type — символьный тип, литерный тип, тип char

Character variable — символьная переменная

Chart — график, диаграмма

Chat — разговор в сети Интернет, чат

Checkbox — флажок

Checkout compiler — отладочный компилятор

Checkpoint — контрольная точка

Child window — дочернее окно

Chip — микросхема

Choice — выбор, пункт меню

Chroma — интенсивность цвета

Circuit board — плата

Circuit — линия связи, канал связи

Circuit card — плата

CISC (Complex Instruction Set Computer, **Complex Instruction Set Computing)** — компьютер со сложной системой команд, CISC-компьютер, архитектура CISC

Class — класс

Class declaration — описание класса

Class diagram — диаграмма классов

Class instance — экземпляр класса

Class method — метод класса

Class property — свойство класса

Classifier — классификатор

Click — щелчок

Client — клиент

Client certificate authentication — аутентификация по сертификату клиента

Client-Server architecture — архитектура "клиент-сервер"

Client-server thelephony — клиент-серверная телефония

Clip art — аппликация, графическая заготовка, художественная заготовка, клипарт

Clipboard — буфер обмена

Clock rate — тактовая частота

C Loop sentences — операторы цикла языка Си, предложения цикла языка Си

Closed file — закрытый файл

Closing of a file — закрытие файла

Cluster — кластер

Cluster computer system — кластерная вычислительная система

CMM (Capability Maturity Model) — модель зрелости возможностей

CMOS (Complementary metal-oxide-semiconductor) — комплементарная структура металл-оксид-полупроводник, КМОП-структура, CMOS-структура

CMOS memory, CMOS RAM (Complementary metal-oxide semiconductor memory) — КМОП-память, CMOS-память

CMS (Color Management System) — система управления цветом

Cobol — Кобол

CODASYL (Conference On Data System Language) — КОДАСИЛ, Конференция по языкам информационных систем

Code — код

Code page — кодовая страница

Codec (Coder-DECoder, Compressor-DECompressor) — кодер-декодер, компрессор-декомпрессор, кодек

Coder-decoder (codec) — кодер-декодер, кодек

Coercion — приведение типов, преобразование типов

Collaboration diagram — диаграмма кооперации

Collaboration — кооперация

Color — цвет

Color depth — битовая глубина цвета, глубина цвета, разрядность цвета

Color management system (CMS) — система управления цветом

Color marker — цветовой маркер

Color model — цветовая модель

Color saturation — насыщенность цвета

Column — колонка, столбец

COM (Component Object Model) — модель COM

Combo box — поле со списком

Command — инструкция, команда

Command File — командный файл

Command line — командная строка

Command processor — командный процессор, процессор командного языка

Comment — комментарий

Comment out — закомментировать

Commitment — завершение транзакции

Committed Transaction — подтвержденная транзакция

Common block — общий блок

Common gateway interface (CGI) — интерфейс CGI

Common object request broker architecture (CORBA) — технология CORBA

Communication channel — канал связи

Compact Disk (CD, CD-ROM) — компакт-диск

Comparison operation — операция отношения, операция сравнения

Compatibility — совместимость

Compilation Time — время компиляции

Compilation — компиляция

Compile Time — время компиляции

Compiled programming language — компилируемый язык программирования

Compiler — компилятор

Compiling Time — время компиляции

Complementary metal-oxide-semiconductor (CMOS) — комплементарная структура металл-оксид-полупроводник, КМОП-структура, CMOS-структура

Completion — завершение программы

Completion transition — переход по завершении

Complex instruction set computer (CISC) — компьютер со сложной системой команд, CISC-компьютер

Complex instruction set computing (CISC) — архитектура CISC

Complex tests of the program product — комплексные испытания программного продукта

Component — компонент программы, компонент

Component diagram — диаграмма компонентов

Component object model (COM) — компонентная модель объектов, модель COM

Composition — композиция

Compound document — составной документ

Compound key — составной ключ

Compound statement — составной оператор

Compressor—decompressor (codec) — компрессор—декомпрессор, кодексистемная дискета

Computer — вычислительная машина, компьютер, электронная вычислительная машина (ЭВМ)

Computer account — бюджет компьютера, учетная запись компьютера

Computer architecture — архитектура компьютера

Computer based training system (CBT) — обучающее приложение к программному продукту, автоматизированная подготовка

Computer center — вычислительный центр (ВЦ)

Computer cluster, cluster — вычислительный кластер

Computer communication — компьютерная связь

Computer complex — вычислительный комплекс, многомашинный вычислительный комплекс (ВК)

Computer conference — телеконференция

Computer functional diagram — функциональная схема компьютера

Computer game — компьютерная игра

Computer generations — поколения компьютеров

Computer graphics — компьютерная графика, машинная графика

Computer independent language — машинно-независимый язык

Computer instruction — машинная команда

Computer language — машинный язык

Computer network — сеть ЭВМ, вычислительная сеть, компьютерная сеть

Computer operation — машинная операция

Computer peak performance — пиковая производительность компьютера

Computer performance — производительность компьютера, производительность ЭВМ

Computer platform — компьютерная платформа, платформа

Computer program — машинная программа

Computer run — выполнение программы, исполнение программы, прогон программы

Computer science — информатика, вычислительная техника (наука)

Computer speed — производительность компьютера, производительность ЭВМ

Computer system architecture — архитектура вычислительной системы

Computer system — вычислительная система

Computer Time — машинное время

Computer UMA — компьютер UMA, компьютер с однородным доступом к памяти

Computer virus — компьютерный вирус, вирус

Computer word — машинное слово

Computer zero — машинный ноль

Computer-aided design (CAD) — автоматизированное проектирование, система автоматизированного проектирования (САПР)

Computer-Aided Software Engineering (CASE) — автоматизированная разработка программ

Computer-based training (CBT) — автоматизированная подготовка

Computer-oriented language — машинно-ориентированный язык

Computer-sensitive language — машинно-зависимый язык

Computer-telephony integration — компьютерно-телефонная интеграция

Computing machinery — вычислительная техника

Concatenation — конкатенация

Concentrator — концентратор

Concept search — концептуальный поиск

Concurrent algorithm — параллельный алгоритм

Concurrent calculation process — параллельный вычислительный процесс

Concurrent processing — параллельная обработка

Condition Branch instruction — команда условного перехода, команда условной передачи управления

Condition JUMP instruction — команда условного перехода, команда условной передачи управления

Conditional GO TO statement — оператор условного перехода

Conditional statement — условный оператор

Conference — конференция

Configuration File — файл конфигурации

Configuration management — управление конфигурацией

Configuration — конфигурация

Conflict situation — конфликтная ситуация, конфликт

Conflict — конфликтная ситуация, конфликт

Conjunction — конъюнкция

Connection — соединение

Connection switching — коммутация соединений

Console — консоль

Constant — константа

Constraint — ограничение

Construction — конструирование

Constructor — конструктор

Consultative Committee International for Telephone and Telegraph (CCITT) — Международный консультативный комитет по телеграфии и телефонии

Container — контейнер

Context menu — контекстное меню

Context-sensitive help — контекстно-зависимая справка

CONTINUE statement — оператор продолжения

Continuous signal — непрерывный сигнал

Control — элемент управления

Control block — блок управления

Control character — символ

Control flow — поток управления

Control key — управляющая клавиша

Control memory — микропрограммная память

Control panel — панель управления

Control signal — управляющий сигнал

Control statement — управляющий оператор

Control transfer — передача управления

Control unit (CU) — устройство управления

Controller — контроллер

Conversational processor — диалоговый процессор

Conversational system — диалоговая система

Converter — преобразователь, конвертор

Cookie — cookie-код, cookie-файл

Cookie identification — cookie-идентификация

Cooperative multitasking — кооперативная многозадачность

Coprocessor — сопроцессор

Copy — копирование

Copy protection — защита от копирования

CORBA (Common Object Request Broker Architecture) — технология CORBA

CPI (cycles per instruction) — количество тактов на команду

CPU (Central Processing Unit) — ЦП, центральный процессор

CPU address space (Central Processing Unit address space) — адресное пространство центрального процессора

CPU Time (Central Processing Unit Time) — время центрального процессора, процессорное время

Crash — авария

Credentials — верительные данные

Crop — обрезка изображения, обрезка, кадрирование

Cropping — обрезка изображения, обрезка, кадрирование

Cross reference — перекрестная ссылка

Cross-platform — платформная независимость, межплатформность

CSMA/CD (Carrier Sense Multiple Access with Collision Detection) — метод множественного доступа с контролем несущей и разрешением конфликтов, метод CSMA/CD

CTI — компьютерно-телефонная интеграция

CTI board — CTI-плата

CU (Control Unit) — устройство управления

Current directory — текущий каталог

Current disk drive — текущий дисковод

Current disk — текущий диск

Current state — текущее состояние

Cursor control keys — клавиши управления курсором

Cursor — курсор

Cut — вырезание

Cybernetics — кибернетика

Cycle — цикл

Cycle index — параметр цикла

Cycle list — список цикла

Cycle parameter — параметр цикла

Cycle statement — оператор цикла

Cycle with a counter — цикл со счетчиком

Cycle with loop variable, Cycle with parameter — цикл с параметром

Cycles per instruction (CPI) — количество тактов на команду

Cyclic process — циклический процесс

Cycling — зацикливание программы

Cyrillic driver — драйвер-русификатор

D

DAC (Digital-to-Analog Converter) — ЦАП, цифроаналоговый преобразователь

Daemon — демон

Data — данные

Data bank — банк данных

Data block — блок данных

Data Bus — шина данных

Data centric — документо-ориентированный подход

Data communication — передача данных

Data compaction — уплотнение данных

Data compression — сжатие данных

Data entity — объект данных

Data field — поле данных

Data flow diagram (DFD) — диаграмма потоков данных

Data flow, Dataflow — поток данных

Data formatting — форматирование данных

Data input — ввод данных

Data integrity — целостность данных

Data interlock — блокировка данных

Data item — элемент данных, элементарное данное

Data link layer — уровень канала, канальный уровень

Data link — канал передачи данных

Data manipulation — манипулирование данными, обработка данных

Data medium — носитель данных

Data model — модель данных

Data output — вывод данных

Data Processing — обработка данных

Data protection — защита данных

Data representation — представление данных

Data retrieval — выборка данных

Data scheme — схема данных

Data search — поиск данных

Data sorting — сортировка данных, упорядочение

Data source name (DSN) — имя источника данных

Data source — источник данных

Data stream — поток данных

Data structure — структура данных

Data transfer rate — скорость передачи данных

Data transfer — передача данных

Data type — тип данных

Data value — значение типа данных

Database (DB) — база данных (БД)

Database administrator — администратор базы данных

Database engine — ядро системы управления базой данных, ядро СУБД

Database management system (DBMS) — система управления базой данных (СУБД)

Database replication — репликация базы данных

Database scheme — схема данных

Database synchronization — синхронизация баз данных

Datagram — дейтаграмма

DB (Data Base) — база данных (БД)

DBMS (Data Base Management System) — система управления базой данных (СУБД)

DCOM (Distributed Component Object Model) — распределенная компонентная модель объектов

DDE (Dynamic Data Exchange) — динамический обмен данными

Deadlock — тупиковая ситуация, взаимная блокировка, тупик

Debugger — отладчик

Decimal number system — десятичная система счисления

Decimal representation — десятичное представление

Declaration — описание, объявление

Declarator — спецификатор, описатель.

Decoder — дешифратор, декодер

Dedicated line — выделенная линия связи

Default option — выбор по умолчанию

Default value — значение по умолчанию

Deferred event — отложенное событие

Definitional domain — область определения, область существования

Defragmentation — дефрагментация

Delegation — делегирование

Delete — удаление

Demo program — демонстрационная программа

Demonstration program — демонстрационная программа

Deployment diagram — диаграмма размещения

Deployment — развертывание

Derived data type — производный тип данных

Derived element — производный элемент

Descendant — потомок

Descriptor — дескриптор, описатель

Design pattern — образец проектирования

Design — проектирование

Desktop computer — настольный компьютер

Desktop publishing system — настольная редакционно-издательская система

Desktop — рабочий стол

Despooling — деспулинг

Destructor — деструктор

Device — устройство, техническое устройство

Device access — обращение к внешнему устройству

Device allocation — резервирование устройства

Device conflict — конфликт устройств

Device driver — драйвер устройства

Device geometry — геометрия устройства

Device independent — аппаратно-независимый

Devices manager — диспетчер устройств

DFD (Data Flow Diagram) — диаграмма потоков данных

Diagram — диаграмма

Dialog mode — диалоговый режим

Dialog system — диалоговая система

Dialog window — диалоговое окно

Dialog — диалог

Digit — разряд, цифра

Digital camera — цифровой фотоаппарат, цифровая камера, цифровая фотокамера

Digital certificate — цифровой сертификат

Digital computer — цифровая вычислительная машина (ЦВМ)

Digital data form — цифровая форма данных

Digital device — цифровое устройство

Digital money — цифровые деньги

Digital signal processor (DSP) — процессор цифровой обработки сигналов (ЦОС)

Digital signal — цифровой сигнал, дискретный сигнал

Digital signature — цифровая подпись

Digital still camera — цифровой фотоаппарат, цифровая камера, цифровая фотокамера

Digital tablet — цифровой планшет графического ввода

Digital-to-analog converter (DAC) — цифроаналоговый преобразователь (ЦАП)

Digitize — оцифровывать

Digitizer — дигитайзер

Dimension — размерность (массива)

Direct access — прямой доступ

Direct addressing — прямая адресация

Direct-access File — файл прямого доступа, прямой файл

Directive — директива

Directory — каталог, директорий

Directory copying — копирование каталога, копирование директория

Directory creation — создание каталога

Directory database — база данных каталога

Directory delete — удаление каталога

Directory moving — перенос каталога, перемещение каталога

Directory rename — переименование каталога

Directory tree — дерево каталогов

Discrete signal — дискретный сигнал

Discretionary hyphen — дискреционный перенос, мягкий перенос

Disjunction — дизъюнкция

Disk caching — кэширование диска

Disk defragmenter — дефрагментатор диска, дефрагментор диска

Disk drive — дисковод

Disk geometry — геометрия диска

Disk initialization — форматирование диска

Disk letter — буква диска

Disk pack — пакет магнитных дисков, пакет дисков

Disk — диск

Diskette — дискета

Dispatcher — диспетчер

Displacement — смещение

Display — дисплей

Display adapter — адаптер дисплея, видеоадаптер, видеокарта

Distributed application — распределенное приложение

Distributed component object model (DCOM) — распределенная компонентная модель объектов

Distributed database — распределенная база данных

Distribution kit — дистрибутив

Divide by zero error — ошибка деления на ноль, деление на ноль

Division — раздел памяти

DLL (Dynamic Link Library) — библиотека динамической компоновки

DNS (Domain Name System) — доменная система имен

DNS name server — сервер DNS, сервер имен доменов

DO Loop — оператор цикла языка Фортран

Document retrieval — поиск документа

Document window — окно документа

Document WWW — документ WWW

Document — документ

Domain — домен, предметная область

Domain name server — сервер DNS, сервер имен доменов

Domain name system (DNS) — доменная система имен

Domain name — имя домена, доменное имя

Domain specific — предметно-ориентированный

Domain tree — дерево доменов

Dots per inch (DPI) — количество точек на дюйм

Double click — двойной щелчок

Double fault — двойная ошибка

Double word — двойное слово

Downloadable font — загружаемый шрифт

Downloading — загрузка по линии связи, скачивание

Downtime — время простоя

Downward compatibility — совместимость сверху вниз

DPI (Dots Per Inch) — количество точек на дюйм

Drag and drop — перетаскивание мышью, буксировка мышью

Drawing program — программа рисования

Driver — драйвер

Drop cap — буквица

Drop-down list — раскрывающийся список

Dropped capital letter — буквица

DSN (Data Source Name) — имя источника данных

DSP (digital signal processor) — процессор цифровой обработки сигналов (**ЦОС)**

Dual channel Controller — двухканальный контроллер

Dummy statement — пустой оператор

Dump — дамп

Dyadic operation — двуместная операция, бинарная операция

Dynamic data exchange (DDE) — динамический обмен данными

Dynamic HTML — технология Dynamic HTML

Dynamic link library (DLL) — библиотека динамической компоновки, библиотека динамического связывания, динамически загружаемая библиотека

Dynamic memory allocation — динамическое распределение памяти

Dynamic modules allocation — динамическое размещение модулей

Dynamic program loading — динамическая загрузка программы

Dynamic resource allocation — динамическое распределение ресурсов

Dynamic routing — динамическая маршрутизация

Dynamic storage area — динамическая область памяти, проблемная область памяти

Dynamic Web page — динамическая веб-страница

E

E-commerce — электронная коммерция

EDI (electronic data interchange) — стандарты EDI

Edit keys — клавиши редактирования

Editing — редактирование

Effective address — действительный адрес

EFS (Encrypting File system) — шифрующая файловая система

EISA (Extended Industry Standard Architecture) — расширенная архитектура промышленного стандарта, архитектура EISA

Electronic circuit — электронная схема

Electronic commerce — электронная коммерция

Electronic data interchange (EDI) — стандарты EDI

Electronic disk — электронный диск

Electronic document — электронный документ

Electronic mail — электронная почта

Electronic partner — электронный партнер

E-mail — электронная почта

Embedded object — внедренный объект

Embedded system — встроенная система

Emulation — эмуляция

Emulator — эмулятор

Encapsulated PostScript (EPS) — упакованный формат PostScript-файлов, формат EPS

Encapsulation — инкапсуляция

Encipherer — шифратор

Enciphering — шифрование

Encoder — кодировщик, шифратор

Encoding — кодирование, шифрование

Encrypting File system (EFS) — шифрующая файловая система

Encryption — шифрование

End of file (EOF) — конец файла

End of transmission (EOT) — конец передачи

End user — конечный пользователь

Entry action — действие при входе

Entry point — точка входа

Enumerated type — перечисляемый тип, перечислимый тип

Environment — среда, окружение

EOF (End Of File) — конец файла

EOT (End Of Transmission) — конец передачи

E-partner — электронный партнер

EPS (Encapsulated PostScript) — упакованный формат PostScript-файлов, формат EPS

Equipment proFile — профиль оборудования

Equipment — аппаратное обеспечение, аппаратура, оборудование, технические средства

Erase — стирание

Eraser — ластик

Error localization — локализация ошибки

Ethernet — сеть Ethernet

European Particle Physics Laboratory (CERN) — Европейская лаборатория ядерных исследований (Церн)

Evaluate GO TO statement — вычисляемый оператор перехода

Even parity — проверка на четность

Event — событие

Event-driven application — приложение, управляемое событиями

Event-driven programming — программирование, ориентированное на события, событийно-управляемое программирование

Exception — исключительная ситуация, особая ситуация

Exchange — обмен данными, обмен

Executable File — исполняемый файл

Execute Time — время выполнения, время прогона

Executive address — исполнительный адрес

Executive directive — команда операционной системы

Exit — выход

Exit point — точка выхода

Expanded memory — отображаемая память

Expectation — ожидание

Expert system — экспертная система (ЭС)

Exponent — порядок числа

Expression — выражение

Extended industry standard architecture (EISA) — расширенная архитектура промышленного стандарта, архитектура EISA

Extended memory (XMS) — расширенная память

Extended memory manager (XMM) — диспетчер расширенной памяти

Extensible markup language (XML) — язык XML, расширяемый язык разметки

Extension — расширение, расширение имени файла

External device — внешнее устройство

External interrupt — внешнее прерывание

External memory — внешняя память, внешнее запоминающее устройство (ВЗУ)

External reference — внешняя ссылка

External storage — внешняя память, внешнее запоминающее устройство (ВЗУ)

Extranet — экстранет, расширенная интрасеть, сеть extranet

Eyedropper — пипетка

F

Failure — сбой

FAT (File Allocation Table) — таблица размещения файлов, таблица FAT

FAT File system — файловая система FAT

Fatal error — неисправимая ошибка, фатальная ошибка

Fax — факс

Fax machine — факс-аппарат, факс-машина, факс

Fax services API — интерфейс fax services API

Fax-modem — факс-модем

FDDI (Fiber Distributed Data Interface) — интерфейс для передачи данных по волоконно-оптическим каналам, стандарт FDDI

Fetch Time — время выборки

Fetching — выборка

FF (Form Feed) — перевод страницы, подача страницы, прогон страницы

Fiber distributed data interface (FDDI) — интерфейс для передачи данных по волоконно-оптическим каналам, стандарт FDDI

Field — поле

FIFO (First In First Out) — очередь

File — файл

File activation — открытие файла, активизация файла

File allocation table (FAT) — таблица размещения файлов, таблица FAT

File allocation table File system — файловая система FAT

File closing — закрытие файла

File compaction — сжатие файла

File compression — сжатие файла

File copying — копирование файла

File delete — удаление файла

File inclusion — включение файла

File manager — диспетчер файлов

File moving — перенос файла, перемещение файла

File name extension — расширение имени файла, расширение

File name picture — шаблон имени файла

File recovery — восстановление файла

File rename — переименование файла

File search — поиск файла

File security — защита файла

File server — файловый сервер, файл-сервер

File system driver (FSD) — драйвер файловой системы

File system — файловая система

File transfer protocol (FTP) — протокол FTP

File type — тип файла

Files and directories marking — отметка файлов и каталогов

Files sorting — сортировка файлов

Fill — заливка

Filling — закраска

Final state — заключительное состояние

Finite-state machine — конечный автомат

Firewall — брандмауэр

Firmware — встроенные программы, зашитые программы

First in first out (FIFO) — очередь

Fixed-point number — число с фиксированной точкой

Flash memory — флэш-память

Flip horizontal — зеркальное отражение по горизонтали, отражение слева направо

Flip vertical — зеркальное отражение по вертикали, отражение сверху вниз

Floating-point number representation — полулогарифмическое представление чисел

Floating-point number — число с плавающей точкой

Floating-point operations per second (FLOPS) — флопс

Floating-point processor — математический сопроцессор

Floating-point representation — представление чисел с плавающей точкой

Flood filling — заливка

Floppy Disk drive — накопитель на гибких магнитных дисках (НГМД)

Floppy disk, **diskette** — гибкий магнитный диск, дискета

FLOPS (Floating-Point Operations Per Second) — флопс

Flowchart symbol — элементы блок-схем

Flynn classification — классификация Флинна, таксономия Флинна

Focus — фокус

Focus of control flow — фокус управления

Folder — папка

Font — шрифт

Font condensing — уплотнение шрифта

Font expansion — разрядка

Font reticle — знакоместо

Font size — размер шрифта, кегль шрифта

Font style — начертание шрифта, стиль шрифта

Font typeface — гарнитура шрифта

Footer — нижний колонтитул

Foreign key — внешний ключ

Fork — развилка

Form — форма

Form feed (FF) — перевод страницы, подача страницы, прогон страницы

Formal parameter — формальный параметр

Formalized language — формализованный язык

Format — формат

Format BMP (BMP) (сокращение от англ. *bitmap*) — формат BMP

Format JPEG (JPEG) — формат JPEG

Format MPEG (MPEG) — формат MPEG

Format PCX (PCX) — формат PCX

Format statement — оператор задания формата, оператор формата

Format TIFF (TIFF) — формат TIFF, тегированный формат файлов изображений

Formatting — форматирование

Forms — формы

Fortran loop statement — оператор цикла языка Фортран

Fortran — Фортран

FP (Functional Point) — единица функционального размера

Fraction — мантисса

Fractional part — мантисса

Frame — кадр, фрейм, рамка

Framework — каркас

Free rotate — произвольный поворот

Free software foundation (FSF) — Фонд свободного программного обеспечения

Freeware — бесплатное программное обеспечение, свободное программное обеспечение

Frequency modulation synthesizer — музыкальный синтезатор

Friendly interface — дружественный интерфейс

Front-end processor — буферный процессор, препроцессор

FSD (File System Driver) — драйвер файловой системы

FSF (Free Software Foundation) — Фонд свободного программного обеспечения

FTP (File Transfer Protocol) — протокол FTP

Full screen view — полный экран

Full-text search engine — полнотекстовая поисковая система

Function — функция

Function body — тело функции

Function call — вызов функции, обращение к функции

Function declaration — описание функции

Function header — заголовок функции

Function keys — функциональные клавиши

Functional device — функциональное устройство

Functional Point (FP) — единица функционального размера

Functional unit — функциональное устройство

G

Game program — игровая программа

Gamut — цветовой охват

Garbage collection — сборка мусора

Garbage — мусор

Gatekeeper — привратник

Gateway — шлюз

Gb (Gigabyte) — Гб, гигабайт

Gbyte (Gigabyte) — Гбайт, гигабайт

GDI (Graphics Device Interface) — интерфейс GDI, интерфейс графических устройств

Generalization — обобщение

Geometry — геометрия устройства, геометрия

GFLOPS (GigaFLOPS) — Гфлопс, гигафлопс

GIF (Graphics Interchange Format) — формат GIF, формат обмена графическими данными

Gigabyte (Gbyte, Gb) — гигабайт

GigaFLOPS (GFLOPS) — гигафлопс, Гфлопс

Global network — глобальная вычислительная сеть

Global positioning satellite (GPS) — навигационный спутник

Global variable — глобальная переменная

Globally unique identifier (GUID) — глобально-уникальный идентификатор

GNU — проект GNU

GO TO statement — оператор перехода

GPS (Global Positioning Satellite) — навигационный спутник

Gradient — градиент

Graphical notation — графическая нотация

Graphic character — графический символ

Graphic data — графическая информация, графические данные

Graphic dialog — графический диалог

Graphic File — графический файл

Graphic information — графическая информация

Graphic subsystem — графическая подсистема

Graphic user interface (GUI) — графический интерфейс пользователя

Graphics accelerator — графический акселератор, графический сопроцессор

Graphics adapter — графический адаптер

Graphics coprocessor — графический сопроцессор

Graphics device interface (GDI) — интерфейс графических устройств, интерфейс GDI

Graphics editor — графический редактор, редактор изображений

Graphics interchange format (GIF) — формат GIF, формат обмена графическими данными

Graphics mode — графический режим

Graphics object — графический объект

Graphics primitive — графический примитив

Graphics tablet — графический планшет, цифровой планшет графического ввода, цифровой планшет

Graphics tools — инструменты машинной графики, графические инструменты

Grawler — робот поисковой системы

Gray scale — градации серого, шкала серого

Grid — сетка

Group name — имя группы

Group — группа

Guard condition — сторожевое условие

GUI (Graphic User Interface) — графический интерфейс пользователя

GUID (globally unique identifier) — глобально-уникальный идентификатор

H

H.323 — стандарты H.323

H.323 network — сеть стандарта H.323

Hacker — хакер

Halt — останов

Hand — рука

Hand-held PC (HPC) — блокнотный компьютер, карманный компьютер

Handshaking — квитирование

Hang-up — зависание

Hard copy — твердая копия

Hard disk controller — контроллер жесткого диска

Hard disk — жесткий магнитный диск, жесткий диск

Hardware block — аппаратный блок

Hardware compatibility list (HCL) — список совместимого оборудования

Hardware configuration — конфигурация компьютера

Hardware failure — аппаратный сбой

Hardware independent — аппаратно-независимый

Hardware interrupt — аппаратное прерывание

Hardware key — аппаратный ключ

Hardware — аппаратное обеспечение, аппаратура, технические средства

Hardwired — аппаратный

HCL (Hardware Compatibility List) — список совместимого оборудования

HDBMS (Hierarchical Database Management System) — иерархическая СУБД

Head program — главная программа, основная программа

Header — верхний колонтитул

Heap — куча

Help line — строка подсказки

Help — справка, справочная система, оперативная помощь

Hertz (Hz) — герц (Гц)

Heterogeneous computer system — гетерогенная вычислительная система, разнородная вычислительная система

Hewlett-Packard graphics language (HPGL) — язык HPGL

Hexadecimal number system — шестнадцатеричная система счисления

Hidden File — скрытый файл

Hidden line — невидимая линия

Hidden surface — невидимая поверхность

Hidden text — скрытый текст

Hierarchical database management system (HDBMS) — система управления иерархическими базами данных, иерархическая СУБД

Hierarchical database — иерархическая база данных

Hierarchy — иерархия

High performance Fortran (HPF) — высокопроизводительный Фортран, язык HPF

High-level programming language — язык программирования высокого уровня

High-performance File system (HPFS) — файловая система HPFS

Hit — посещение

Hive — улей

Home network — домашняя сеть

Home page — домашняя страница, заглавная страница, начальная страница

Hook — ловушка

Horizontal menu — горизонтальное меню

Host — хост

Host computer — главная ЭВМ, хост

Hot keys — горячие клавиши

Hot spot — горячая область, горячая точка

HPC (Hand-held PC) — блокнотный компьютер, карманный компьютер

HPF (High performance Fortran) — высокопроизводительный Фортран, язык HPF

HPFS (High-Performance File System) — файловая система HPFS

HPGL (Hewlett-Packard Graphics Language) — язык HPGL

HTML (Hypertext Markup Language) — язык HTML

HTTP (Hypertext Transfer Protocol) — протокол передачи гипертекста, протокол HTTP

HTTPD (Hypertext Transfer Protocol Daemon) — сервер протокола передачи гипертекста, сервер HTTPD

Hub — хаб, концентратор

Hue — тон

Hybrid Computer — аналого-цифровая вычислительная система, гибридная вычислительная система

Hyperlink — гиперссылка

Hypermedia — гипермедиа

Hypertext markup language (HTML) — язык HTML

Hypertext transfer protocol (HTTP) — протокол передачи гипертекста, протокол HTTP

Hypertext transfer protocol daemon (HTTPD) — сервер протокола передачи гипертекста

Hypertext — гипертекст

Hyphenation — перенос слов

Hz — герц (Гц)

I

IBM PC — персональный IBM-совместимый компьютер

IBM-PC File system — файловая система персональных IBM-совместимых компьютеров

IBM-PC main memory — оперативная память персональных IBM-совместимых компьютеров

ICM (Image Color Matcher) — модуль подбора цветов

Icon menu — пиктографическое меню

Icon — значок, пиктограмма, иконка

IDE (Integrated Development Environment) — интегрированная среда разработки

IDE (Integrated Drive Electronics) — встроенный интерфейс накопителей, интерфейс IDE

Identifier — идентификатор

Idle Time — время простоя

IEEE (Institute of Electrical and Electronics Engineers) — Институт инженеров по электротехнике и радиоэлектронике (ИИЭР США)

IEEE standards — стандарты IEEE

IETF (Internet Engineering Task Force) — инженерный консорциум разработчиков стандартов Интернета, инженерная комиссия Интернета, комитет IETF

IIS (Internet Information Server) — информационный сервер Интернета, сервер IIS

Illegal access — несанкционированный доступ

Image color matcher (ICM) — модуль подбора цветов

Immediate addressing — прямая адресация, непосредственная адресация

Immediate operand — непосредственный операнд

Imperative GO TO statement — оператор безусловного перехода

Imperative statement — выполняемый (исполняемый) оператор

Impersonation — имперсонация

Implementation — реализация

Implication — импликация

Implicit declaration — неявное описание

Import — импорт

Inactive program — неактивная программа

Inactive window — неактивное окно

Inch, ″ — дюйм

Incoming text — исходный текст

Increment — приращение

Incremental software development process — инкрементный процесс разработки программного обеспечения

Indent — отступ

Index — индекс, индекс поисковой системы

Index register — индексный регистр, регистр индекса

Indexed addressing — индексная адресация, адресация с индексированием

Indicator — индикатор

Indirect addressing — косвенная адресация

INF File — файл установки

Infected File — инфицированный файл

Infinite loop — бесконечный цикл

Informatics — информатика

Information capacity — информационная емкость, информационный объем

Information character — текстовый символ

Information computer center — информационно-вычислительный центр (ИВЦ)

Information environment — информационная среда

Information processing — обработка информации

Information system — информационная система

Information technology — информационная технология

Information word — информационное слово

Information — информация

Infrared data association (IrDA) — спецификация IrDA

Infrared interface (Ir Interface) — инфракрасный интерфейс

Infrared port (Ir port) — инфракрасный порт

Inheritance — наследование

Initial state — начальное состояние

Initialization — инициализация

Initiation — инициирование

Ink-jet printer — струйный принтер

Input data — входные данные

Input device — устройство ввода

Input field — поле ввода

Input instruction — команда ввода

Input statement — оператор ввода

Input-output channel — канал ввода/вывода

Input-output device — устройство ввода/вывода

Input-output interrupt — прерывание ввода/вывода

Input-output — ввод/вывод

Insert mode — режим вставки

Installation, setup — инсталляция, установка

Installer — инсталлятор

Instance — экземпляр класса

Instantiation — создание экземпляра

Institute of electrical and electronics engineers (IEEE) — Институт инженеров по электротехнике и радиоэлектронике (ИИЭР США)

Instruction address — адрес команды

Instruction register — регистр команд

Instruction set — система команд

Instruction stream — поток команд

Instruction — инструкция, команда

Integer type number — число целого типа

Integer type — целый тип, целочисленный тип

Integer variable — целая переменная, переменная целого типа

Integer — целое число

Integrated circuit — интегральная схема (ИС), элементная база

Integrated development environment (IDE) — интегрированная среда разработки

Integrated device electronics — встроенный интерфейс накопителей, интерфейс IDE

Integrated drive electronics (IDE) — встроенный интерфейс накопителей, интерфейс IDE

Integrated environment — интегрированная среда

Integrated package — интегрированный пакет

Integrated services digital network (ISDN) — интегрированная цифровая сеть связи, сеть ISDN

Integrated software — интегрированное программное обеспечение

Integrated system — интегрированная система

Intelligent information system — интеллектуальная информационная система

Intelligent interface — интеллектуальный интерфейс

Intelligent programming system — интеллектуальная система программирования

Intelligent terminal — интеллектуальный терминал

Intelligent tutorial system — интеллектуальная обучающая система

Inter modular reference — межмодульная ссылка

Interaction diagram — диаграмма взаимодействия

Interaction — взаимодействие

Interactive computer system — интерактивная вычислительная система

Interactive mode — интерактивный режим

Interface device — устройство сопряжения

Interface — интерфейс

Interlaced GIF — чересстрочный GIF

Intermediate language — промежуточный язык

Internal block — вложенный блок, подблок

Internal cycle — вложенный цикл

Internal interrupt — внутреннее прерывание

Internal language — внутренний язык

Internal memory — внутренняя память

Internal transition — внутренний переход

International organization for standardization/open system interconnection model (OSI/ISO model) — модель взаимодействия открытых систем, модель OSI/ISO

International standards organization (ISO) — Международная организация по стандартизации

International Telecommunication Union-Telecommunication standardization sector (ITU-T) — Телекоммуникационный сектор Международного союза электросвязи

International Telecommunication Union (ITU) — международный союз телекоммуникаций

Internet — Интернет

Internet engineering task force (IETF) — Инженерный консорциум разработчиков стандартов Интернета, Инженерная комиссия Интернета, Инженерные силы Интернета, комитет IETF

Internet information server (IIS) — информационный сервер Интернета, сервер IIS

Internet provider — интернет-провайдер

Interpretable programming language — интерпретируемый язык программирования

Interpretation — интерпретация

Interpreter — интерпретатор

Interrupt — прерывание

Interrupt handler — обработчик прерываний

Interrupt processing — обработка прерываний

Interrupt request lines (IRQ) — линии запроса прерываний, линии прерываний, линии IRQ

Interrupt request package (IRP) — пакет запроса прерывания

Interrupt vector — вектор прерываний

Intranet — интранет, интрасеть, корпоративная сеть

Invalid opcode error — ошибка недопустимого кода операции

Invariant — инвариант

Invert color — инверсия цвета

Invert — инвертировать

IP address — интернет-адрес, адрес IP

IP-phone — IP-телефон

IP-telephone — IP-телефон

IP-telephony — IP-телефония, интернет-телефония

Ir Interface (InfraRed interface) — инфракрасный интерфейс

Ir port (InfraRed port) — инфракрасный порт

IrDA (Infrared Data Association) — спецификация IrDA

IRP (Interrupt Request Package) — пакет запроса прерывания

IRQ (Interrupt Request Lines) — линии запроса прерываний, линии прерываний, линии IRQ

ISDN (Integrated Services Digital Network) — интегрированная цифровая сеть связи, сеть ISDN

ISDN Interface card — плата интерфейса ISDN

ISO (International Organization for Standardization) — Международная организация по стандартизации

ISO standards — стандарты ISO

Italic — курсив

Item — элемент данных, элементарное данное

Iterative loop — итерационный цикл

ITU (International Telecommunication Union) — Международный союз телекоммуникаций

ITU-T — Телекоммуникационный сектор Международного союза электросвязи

J

Java programming language — язык программирования Java

Java Telephony API (JTAPI) — телефонный интерфейс прикладного программирования для языка Java

Java virtual machine (JVM) — виртуальная машина Java

JavaScript — язык сценариев JavaScript, язык JavaScript

Job — задание

Join — соединение

Joint pictures entertainment group (JPEG) — группа объединенных экспертов по фотографии, группа JPEG

Joker — джокер

Joystick — джойстик

JPEG (Joint Pictures Entertainment Group) — группа JPEG, группа объединенных экспертов по фотографии, формат JPEG

JTAPI (Java Telephony API) — телефонный интерфейс прикладного программирования для языка Java, интерфейс JTAPI

Jump instruction — команда передачи управления, команда ветвления, команда перехода

Jump — ветвление программы, переход

Justify — выравнивание, выключка

JVM (Java Virtual Machine) — виртуальная машина Java

K

K — К, килобайт

Kb — Кб, килобайт

Kbit/s — Кбит/с, килобит в секунду

Kbyte — Кбайт, килобайт

Kernel mode — привилегированный режим, режим операционной системы, режим ядра

Kernel — ядро операционной системы, ядро

Kerning — кернинг

Key menu — пиктографическое меню

Key — клавиша, ключ

Keyboard accelerator — командная клавиша

Keyboard layout — раскладка клавиатуры

Keyboard shortcut — командная клавиша

Keyboard — клавиатура

Keyword search — поиск по ключевым словам

Keyword — ключевое слово

Khz — кГц, килогерц

Kilobit per second — килобит в секунду (Кбит/с)

Kilohertz — килогерц, кГц

Knowledge base — база знаний

L

Label declaration — описание меток

Label — метка

Labeled statement — помеченный оператор

LAN (Local Area Network) — локальная вычислительная сеть (ЛВС)

Landscape page orientation — альбомная ориентация страницы, ландшафтная ориентация страницы

Laptop — лэптоп

Laser disk drive — накопитель на лазерных дисках

Laser printer — лазерный принтер

Lasso — лассо

Last In First Out (LIFO) — стек, магазин

Lazy write — отложенная запись

LCD (Liquid Crystal Display) — дисплей на жидких кристаллах, жидкокристаллический дисплей

Leading — межстрочный интервал, интерлиньяж

Learning program — обучающая программа

Legacy — унаследованный

Legend — легенда

Length — длина

Lexical analyzer — лексический анализатор

Librarian — библиотекарь

Library program — библиотечная программа

Lifeline — линия жизни

LIFO (Last In First Out) — стек, магазин

Light button — кнопка

Light pen — световое перо

Line — линия, строка, строка символов

Line printer (LPT) — алфавитно-цифровое печатающее устройство (АЦПУ)

Line printer daemon (LPD) — демон линейного принтера

Line printer remote clients (LPR) — протокол удаленных клиентов линейного принтера, протокол LPR

Line spacing — межстрочный интервал, интерлиньяж

Lines of code (LOC) — число строк кода

Link — ссылка, линия связи

Linkage editor — редактор связей, компоновщик

Linked object — связанный объект

Linker — компоновщик, редактор связей

Linking loader — компонующий загрузчик

Linking, **linkage editing** — компоновка, редактирование связей

Linux — операционная система Linux (ОС Linux), Linux

Liquid crystal display (LCD) — дисплей на жидких кристаллах, жидкокристаллический дисплей

Lisp — Лисп

List — список

Listing — листинг, распечатка

Literal — литерал

LLC (Logical Link Control) — управление логической связью, подуровень LLC

Load module — загрузочный модуль

Loadable driver — загружаемый драйвер

Loader — загрузчик

Loading — загрузка

LOC (Lines of code) — число строк кода

Local area network (**LAN**) — локальная вычислительная сеть (ЛВС)

Local Bus — локальная шина

Local computer network — локальная компьютерная сеть

Local computer — локальный компьютер

Local group — локальная группа

Local network — локальная сеть

Local procedure call (LPC) — локальный вызов процедуры

Local variable — локальная переменная

Lock — замóк

Locking — блокировка

Lockout — захват, монопольное использование, блокировка

Log File — файл протокола

Logic error — логическая ошибка

Logical "AND" — логическое умножение, логическое "И", конъюнкция

Logical "OR" — логическое сложение, логическое "ИЛИ", дизъюнкция

Logical address — логический адрес, виртуальный адрес

Logical device — логическое устройство

Logical disk — логический диск

Logical expression — логическое выражение

Logical link control (LLC) — управление логической связью, подуровень LLC

Logical multiplication — логическое умножение, логическое "И", конъюнкция

Logical name — логическое имя

Logical operation character — знак логической операции

Logical operations — логические операции, булевы операции

Logical operator — знак логической операции

Logical record — логическая запись

Logical type — логический тип

Logical value — логическое значение, булево значение

Logical variable — логическая переменная

Login script — сценарий регистрации

Login — вход в систему, регистрация

Logoff, logout — выход из системы

Logon script — сценарий регистрации

Logon — вход в систему, регистрация

Long name — длинное имя

Loop — время такта, такт, цикл

Loop body — тело цикла

Loop header — заголовок цикла

Loop termination — выход из цикла

Loop time — время такта, такт

Loop variable — управляющая переменная цикла

Loop — цикл

Loophole — зацикливание, зацикливание программы

Lowercase — нижний регистр

LPC (Local Procedure Call) — локальный вызов процедуры

LPD (Line Printer Daemon) — демон линейного принтера

LPR (Line Printer Remote clients) — протокол удаленных клиентов линейного принтера, протокол LPR

LPT (Line Printer) — алфавитно-цифровое печатающее устройство (АЦПУ)

M

MAC (Media Access Control) — управление доступом к среде, подуровень MAC

Machine failure — машинный сбой

Machine intelligence — машинный интеллект

Machine language — машинный язык

Machine program — машинная программа

Machine-check interrupt — прерывание от схем контроля, прерывание по машинному сбою

Macro — макрокоманда

Macro assembler — макроассемблер

Macro call — макровызов, макрокоманда

Macro declaration — макроопределение

Macro definition library — библиотека макроопределений, макробиблиотека

Macro definition — макроопределение

Macro expansion — макрорасширение

Macro facility — макросредства

Macro generation — макрогенерация, макроподстановка

Macro generator — макрогенератор

Macro instruction — макрокоманда

Macro language — макроязык

Macro parameter — макропараметр

Macro processor — макропроцессор

Macro program — макропрограмма

Macro programming — макропрограммирование

Macro virus — макровирус

Macrocode — макровызов, макрокоманда

Macros — макрос

Magnetic card — магнитная карта

Magnetic disk storage — накопитель на магнитных дисках (НМД)

Magnetic disk — магнитный диск

Magnetic tape storage — накопитель на магнитной ленте (НМЛ)

Magnetic tape — магнитная лента (МЛ)

Mail exchanger record (MX record) — имя пункта обработки почтовых сообщений, MX-запись

Mail merge — слияние

Mailbox — почтовый ящик

Main memory — оперативная память, основная память

Main program — главная программа, основная программа

Main storage — оперативная память, основная память

Mainframe — мэйнфрейм

Malfunction — машинный сбой, аппаратный сбой

Manager — диспетчер

MAPI (Messaging Application Programming Interface) — интерфейс программирования приложений для передачи сообщений, интерфейс MAPI

Mark — маркер

Marker — маркер

Markup language — язык разметки

Marquee — выделение области, выбор объектов, выделение

Mask — маска

Mass storage — массовая память

Master directory — главный каталог, корневой каталог

Master File table (MFT) — главная файловая таблица

Master key — основной ключ, первичный ключ

Master program — мастер-программа

Math coprocessor — математический сопроцессор

Mathematical coprocessor — математический сопроцессор

Matrix printer — матричный принтер

MB (MegaByte) — мегабайт (Мбайт, Мб)

Mbit/s — мегабит в секунду, Мбит/с

MBps — мегабайт в секунду, Мбайт/с

Mbps — мегабит в секунду, Мбит/с

Mcsec — микросекунда (мксек, мкс)

MCU (multipoint control unit) — устройство управления конференциями

MDA (Model Driven Architecture) — архитектура, управляемая моделью

MDI (Multiple Document Interface) — архитектура MDI

Mean Time between failures (MTBF) — наработка на отказ, среднее время между отказами

Media access control (MAC) — управление доступом к среде, подуровень MAC

Media control — управление голосовым каналом

Media stream — голосовой канал

Medium — среда

Megabyte (MB) — мегабайт (Мбайт, Мб)

MegaFLOPS (MFLOPS) — мегафлопс (Мфлопс)

Megahertz (MHz) — мегагерц (МГц)

Memory — память компьютера, память ЭВМ, память

Memory allocation — распределение памяти

Memory bank — банк памяти, страница памяти

Memory block — блок памяти

Memory capacity — емкость памяти, объем памяти

Memory circuit — микросхема памяти

Memory cleaning — чистка памяти, сбор мусора, сборка мусора

Memory dump — дамп памяти, дамп

Memory hierarchy — иерархия памяти

Memory management system — система управления памятью

Memory management — управление памятью

Memory mapping — распределение памяти

Memory module — модуль памяти

Memory protection — защита памяти

Memory section — секция памяти

Memory segment — сегмент памяти, сегмент

Menu bar — строка меню

Menu item — пункт меню

Menu — меню

Merge — слияние

Message — сообщение

Message passing interface (MPI) — интерфейс передачи сообщений, технология MPI

Message passing model — модель передачи сообщений

Messaging application programming Interface (MAPI) — интерфейс программирования приложений для передачи сообщений, интерфейс MAPI

Metaclass — метакласс

Metacomputer — метакомпьютер

Metacomputing — метакомпьютинг

Metadata — метаданные

MetaFile — метафайл

Metalanguage — метаязык

Metamodel — метамодель

Method — метод класса, метод

Metric — метрика

MFLOPS (MegaFLOPS) — мегафлопс (Мфлопс)

MFM (Modified Frequency Modulation) — модифицированная частотная модуляция (МЧМ)

MFT (Master File Table) — главная файловая таблица

MHz (Megahertz) — мегагерц (МГц)

Micro calculator — микрокалькулятор

Micro program store — микропрограммная память

Micro program — микропрограмма

Microcircuit — микросхема

Microcomputer — микроЭВМ

Microinstruction — микрокоманда

Microprocessor — микропроцессор

Microprogramming — микропрограммирование

Microsecond (Mcsec) — микросекунда (мксек, мкс)

Microsoft Visual Basic scripting edition — язык сценариев VBScript, язык VBScript

MIDI (Musical Instrument Digital Interface) — цифровой интерфейс музыкальных инструментов, интерфейс MIDI

Million instructions per second (MIPS) — миллион операций в секунду

MIMD-computer — MIMD-компьютер

MIME (Multipurpose Internet Mail Extension) — многоцелевые расширения почты Интернета, стандарт MIME

Mini driver — мини-драйвер

Miniport driver — драйвер мини-порта

MIPS (million instructions per second) — миллион операций в секунду

Mirror set — зеркальный набор

Mirroring — зеркальное копирование, зеркалирование

MISD-computer — MISD-компьютер

MMX (MultiMedia Extension) — технология MMX

MMX processor (MultiMedia Extension processor) — процессор MMX

Mobile computer — мобильный компьютер

Mobile PC — мобильный компьютер

Modal dialog box — модальное диалоговое окно, модальное окно

Modal dialog window — модальное диалоговое окно, модальное окно

Modal dialog — модальное диалоговое окно, модальное окно

Modal window — модальное диалоговое окно, модальное окно

Model Driven Architecture (MDA) — архитектура, управляемая моделью

Model — модель

Modem — модем, модулятор-демодулятор

Moderator — модератор

Modified alternative code — модифицированная альтернативная кодировка ГОСТа

Modified frequency modulation (MFM) — модифицированная частотная модуляция (МЧМ)

Modular programming — модульное программирование

Module stub — заглушка

Module — модуль

Module — программный модуль

Monadic operation — одноместная операция, унарная операция

Monitor — монитор

Mother board — материнская плата

Mouse — мышь

Mouse cursor — указатель мыши

Moving pictures expert group (MPEG) — группа MPEG, группа экспертов в области движущихся изображений, экспертная группа по кинематографии, стандарты MPEG, формат MPEG

MPEG-codec — кодек MPEG

MPI (message passing interface) — технология MPI, интерфейс передачи сообщений

MPI-program — MPI-программа

MPMD (multiple progam multiple data) — MPMD-модель

MS-DOS external command — внешняя команда MS-DOS

MS-DOS internal command — внутренняя команда MS-DOS

MS-DOS — операционная система MS-DOS (ОС MS-DOS)

MTBF (Mean Time Between Failures) — наработка на отказ, среднее время между отказами

Multi-access system — система коллективного пользования, система коллективного доступа

Multicasting — многоадресная рассылка пакетов

Multi-document interface (MDI) — архитектура MDI

Multimedia extension (MMX) — технология MMX

Multimedia extension processor (MMX processor) — процессор MMX

Multimedia — мультимедиа

Multiple boot configuration — множественная загрузка

Multiple computer complex — многомашинный вычислительный комплекс

Multiple document interface (MDI) — архитектура MDI

Multiple inheritance — множественное наследование

Multiple progam multiple data (MPMD) — MPMD-модель

Multipoint control unit (MCU) — устройство управления конференциями

Multiprocessor computer — многопроцессорная ЭВМ, мультипроцессорная ЭВМ

Multiprogramming — мультипрограммирование

Multipurpose Internet mail extension (MIME) — многоцелевые расширения почты Интернета

Multitasking operating system — многозадачная операционная система

Multithreading — многопоточность

Multi-tiered architecture — многоуровневая архитектура приложения

Musical instrument digital interface (MIDI) — цифровой интерфейс музыкальных инструментов

MX record (Mail eXchanger record) — имя пункта обработки почтовых сообщений, MX-запись

My computer — Мой компьютер

My documents — Мои документы

N

Name — имя

Name space — пространство имен

Named arguments — именованные аргументы

Named pipe — именованный канал, абстрактный файл

Nanosecond (nsec) — наносекунда (нсек, нс)

National language support (NLS) — многоязыковая поддержка

Natural language — естественный язык

NC (Norton Commander) — оболочка Norton Commander

NDIS (Network Device Interface Specification) — спецификация интерфейсов сетевых устройств, сетевой стандарт NDIS

Nested block — вложенный блок, подблок

Nested Loop — вложенный цикл

NetBEUI (Network Basic input/output system Extended User Interface) — протокол NetBEUI

Netware — сетевое программное обеспечение, сетевое ПО

NetWare — сеть NetWare, операционная система NetWare

Network abonent — абонент сети

Network basic input/output system extended User Interface (NetBEUI) — протокол NetBEUI

Network board — сетевая карта, сетевая плата, сетевой адаптер

Network card — сетевая карта, сетевая плата, сетевой адаптер

Network database management system (Network DBMS) — система управления сетевыми базами данных, сетевая СУБД

Network database — сетевая база данных

Network DBMS (Network Database Management System) — система управления сетевыми базами данных, сетевая СУБД

Network device interface specification (NDIS) — спецификация интерфейсов сетевых устройств, сетевой стандарт NDIS

Network layer — сетевой уровень

Network monitor — монитор сети, сетевой монитор

Network node — узел сети

Network printer — сетевой принтер

Network protocol — сетевой протокол

Network segment (SEG) — сегмент сети, сегмент

Network topology — топология сети

Network-interface printer — принтер с сетевым интерфейсом

New technology File system (NTFS) — файловая система Windows NT, файловая система NTFS

NLS (National Language Support) — многоязыковая поддержка

Node — узел, узел сети, вычислительный узел

Node name — имя узла

Non modal window — немодальное диалоговое окно

Nonexecutable statement — невыполняемый оператор

Nonpaged pool — резидентный пул

Nonprinting characters — непечатаемые символы

Normalized form — нормализованная форма

Normalized number — нормализованное число

Norton Commander (NC) — оболочка Norton Commander

Notation — нотация

Notebook — ноутбук, портативный компьютер

Novel NetWare — операционная система NetWare

Nsec (Nanosecond) — наносекунда (нсек, нс)

NTFS (New Technology File System) — файловая система Windows NT, файловая система NTFS

Null statement — пустой оператор

Number — число

Number range — диапазон изменения чисел

Number system — система счисления

Numeric code — цифровой код

Numeric keypad — малая цифровая клавиатура

O

Object — объект

Object Constraint Language (OCL) — объектный язык ограничений

Object conversion — преобразование объектов

Object flow — поток объектов

Object linking and embedding (OLE) — связывание и внедрение объектов

Object Management Group (OMG) — Консорциум по объектным технологиям

Object module — объектный модуль

Object program — объектная программа

Object request broker (ORB) — объектный брокер запросов

Object-oriented programming (OOP) — объектно-ориентированное программирование (ООП)

OCL (Object Constraint Language) — объектный язык ограничений

OCR (Optical Character Recognition) — оптическое распознавание символов

Octal number system — восьмеричная система счисления

Octal number — восьмеричное число

ODBC (Open Database Connectivity) — открытый доступ к базам данных

ODBC data source — источник данных ODBC

OEM (Original Equipment Manufacturer) — производитель оборудования

Office application — офисное приложение

Offset — смещение

OLE (Object Linking and Embedding) — связывание и внедрение объектов

OLE automation (Object Linking and Embedding automation) — автоматизация

OLE technology (Object Linking and Embedding technology) — технология OLE, связывание и внедрение объектов

OMG (Object Management Group) — консорциум по объектным технологиям

One task operating system — однозадачная операционная система

One-dimensional array — одномерный массив

On-line help — оперативная помощь

Online mode, On-line mode — онлайновый режим

On-line tutorial — обучающая программа

OOP (Object-Oriented Programming) — объектно-ориентированное программирование (ООП)

Opcode (Operation code) — код операции (КОП)

Open database connectivity (ODBC) — открытый доступ к базам данных

Open databases — открытые базы данных

Open key enciphering — шифрование с открытым ключом, несимметричное шифрование

Open key encryption — шифрование с открытым ключом, несимметричное шифрование

Operand — операнд

Operating system (OS) — операционная система (ОС)

Operating system Linux (OS Linux) — операционная система Linux (ОС Linux), Linux

Operating system loading — загрузка операционной системы

Operating system MS-DOS (OS MS-DOS) — операционная система MS-DOS (ОС MS-DOS)

Operating system NetWare — операционная система NetWare

Operating system UNIX (OS UNIX) — операционная система UNIX (ОС UNIX)

Operating system Windows (OS Windows) — операционная система Windows (ОС Windows)

Operation code (opcode) — код операции (КОП)

Operation — операция

Operational environment — операционная среда

Operator — оператор, оператор ЭВМ

Optical character recognition (OCR) — оптическое распознавание символов

Optical disk — оптический диск, лазерный диск

Optimization — оптимизация

Optimizing compiler — оптимизирующий компилятор

Option button — радиокнопка, переключатель

Option — опция

ORB (Object Request Broker) — объектный брокер запросов

Ordinal type — перечисляемый тип, перечислимый тип

Original equipment manufacturer (OEM) — производитель оборудования

Orphan — висячая строка

OS (Operating System) — операционная система (ОС)

OS generation (Operating System generation) — генерация операционной системы, конфигурирование операционной системы

OSI/ISO model (International Organization for Standardization/Open System Interconnection model) — модель OSI/ISO, модель взаимодействия открытых систем

OS Linux (Operating System Linux) — операционная система Linux (ОС Linux), Linux

OS MS-DOS (Operating System MS-DOS) — операционная система MS-DOS (ОС MS-DOS)

OS UNIX (Operating System UNIX) — операционная система UNIX (ОС UNIX)

OS Windows (Operating System Windows) — операционная система Windows (ОС Windows)

Outline — обводка

Output data — выходные данные

Output device — устройство вывода

Output instruction — команда вывода

Output statement — оператор вывода

Output — вывод

Overflow error — ошибка переполнения

Overflow — переполнение

Overlay modules allocation — оверлейное размещение модулей

Overlay program — оверлейная программа, программа с оверлейной структурой, программа с перекрываемой структурой

Overlay segment — сегмент перекрытия, оверлейный сегмент, оверлей

Overlay structure — оверлейная структура, структура с перекрытием

Overlay — перекрытие, оверлей

Overrun error — ошибка переполнения

Overrun — переполнение

Overtype mode — режим замещения (при вводе текста)

Owner scope — область действия

P

Pack — упаковывать

Package — пакет

Packet internet groper (PING) — пакетная программа отладки сетей, утилита PING

Packet switching — коммутация пакетов

Packet switching network — сеть с коммутацией пакетов, сеть с пакетной коммутацией

Packet — пакет

Packing density — плотность записи

Page — страница

Page communication — страничный обмен

Page description language (PDL) — язык описания страниц

Page fault — ошибочное обращение к отсутствующей странице, ошибка страницы

Page layout — разметка страницы

Page makeup — верстка страниц

Paged pool — нерезидентный пул

Pagination — разбиение на страницы

Paging File — файл подкачки, файл замещения страниц

Paging — подкачка

Paint brush — кисть

Paint bucket — заливка

Paint program — программа раскраски

Palette — палитра

Panel — панель

Paper feed — прогон бумаги

Paper size — формат бумаги

Paragraph — абзац

Parallel algorithm — параллельный алгоритм

Parallel calculation process — параллельный вычислительный процесс

Parallel calculations — параллельные вычисления

Parallel computer — параллельный компьютер

Parallel computing — параллельные вычисления

Parallel port — параллельный порт

Parallel processing — параллельная обработка

Parallel virtual machine (PVM) — параллельная виртуальная машина, технология PVM

Parameter list — список формальных параметров, список параметров

Parameter passing — передача параметров

Parent window — родительское окно

Parity bit — бит четности

Parity check — контроль по четности, контроль на четность, контроль четности

Partition — раздел памяти, раздел диска, раздел

Pascal — Паскаль (язык программирования)

Pascal loop sentences — операторы цикла языка Паскаль, предложения цикла языка Паскаль

Password — пароль

Password caching — кэширование паролей

Paste — вставка

Patch — заплата

Path — контур, путь

Pathname — полное имя файла

Pause instruction — команда паузы

PCL (Printer Control Language) — язык управления печатью, язык PCL

PCMCIA (Personal Computer Memory Card International Association) — Международная ассоциация производителей карт памяти для персональных компьютеров

PCMCIA adapter (Personal Computer Memory Card International Association adapter) — адаптер PCMCIA

PCX File — файл PCX

PCX — формат PCX

PD (Port Driver) — драйвер порта

PDA (Personal Digital Assistant) — организатор личной деятельности

PDL (Page Description Language) — язык описания страниц

Peer-to-peer network — одноранговая вычислительная сеть

Pen computer — пен-компьютер

Pencil — карандаш

Performance — производительность, быстродействие

Peripheral device — периферийное устройство

PERL (Practical Extraction and Report Language) — язык PERL

Persistent object — персистентный объект

Personal computer (PC) — персональный компьютер, персональная ЭВМ (ПК, ПЭВМ)

Personal computer memory card international association (PCMCIA) — Международная ассоциация производителей карт памяти для персональных компьютеров

Personal digital assistant (PDA) — организатор личной деятельности

Personal IBM-compatible computer (IBM PC) — персональный IBM-совместимый компьютер

Personal identification number — PIN-код

Personal library — библиотека пользователя

Perspective — перспектива

Physical address — физический адрес

Physical record — физическая запись

Physical storage — физическая память

Pick list — список выбора

PING (Packet Internet Groper) — пакетная программа отладки сетей, утилита PING

Pipe — программный канал, канал

Pipeline length — длина конвейера

Pipeline segment — сегмент конвейера, ступень конвейера

Pipeline — конвейер

Piracy — пиратство, незаконное копирование программных средств

Pivot table — сводная таблица

Pixel — пиксел

Pixel image — битовый массив, битовая карта, растр, растровый образ

Pixel map — битовый массив, битовая карта, растр, растровый образ

Pixels per inch (PPI) — количество пикселов на дюйм

Platform independence — платформная независимость, межплатформность

Platform — компьютерная платформа, программная платформа, платформа

Plotter — графопостроитель, плоттер

Plug and Play — технология Plug and Play

Point — пункт

Pointer — указатель

Pointing device — указательное устройство

Point-to-point exchange — двухточечный обмен

Point-to-point passing — двухточечный обмен

Point-to-point protocol (PPP) — протокол PPP

Policy — политика

Polymorphic operation — полиморфная операция

Pool — пул памяти, пул

POP (Post office protocol) — почтовый протокол POP

Pop-up menu — всплывающее меню

Port — порт

Port driver (PD) — драйвер порта

Portable computer — портативный компьютер

Portal — портал, Web-портал

Portrait page orientation — книжная ориентация страницы, портретная ориентация страницы

Positional number system — позиционная система счисления

POST (Power-on self test) — тест при включении питания, самодиагностика при включении питания, самотестирование при включении

Post condition — постусловие

Post office protocol (POP) — почтовый протокол POP

Postprocessor — постпроцессор

PostScript — язык PostScript

Power performance chip (PPC) — процессор PowerPC

Power-on self test (POST) — тест при включении питания, самодиагностика при включении питания, самотестирование при включении

PowerPC (PPC) — процессор PowerPC

PPC (Power performance chip) — процессор PowerPC

PPI (Pixels per inch) — количество пикселов на дюйм

PPP (Point-to-point protocol) — протокол PPP

Practical extraction and report language (PERL) — язык PERL

Precondition — предусловие

Preemption — выгрузка

Preemptive multitasking — вытесняющая многозадачность, приоритетная многозадачность

Prefix — префикс

Preprocessor — препроцессор

Primary key — первичный ключ

Primitive type — встроенный тип, стандартный тип

Primitive — примитив

Print client — клиент печати

Print job data type — тип данных задания на печать

Print job — задание на печать

Print processor — процессор печати

Print queue — очередь заданий на печать

Print server — сервер печати

Print spooler — спулер печати, спулер

Printer control language (PCL) — язык управления печатью, язык PCL

Printer driver — драйвер принтера

Printer — принтер, печатающее устройство

Printing subsystem — подсистема печати

Printing — распечатка

Priority — приоритет

Private class method — закрытый метод класса, приватный метод класса

Private class property — закрытое свойство класса, приватное свойство класса

Private library — библиотека пользователя

Private method — закрытый метод, приватный метод

Private property — закрытое свойство, приватное свойство

Privileged instruction — привилегированная команда

Privileged mode — привилегированный режим, режим операционной системы, режим ядра

Problem-oriented — проблемно-ориентированный

Problem-oriented language — проблемно-ориентированный язык

Procedural application — процедурное приложение

Procedure — процедура

Procedure body — тело процедуры

Procedure call — вызов процедуры, обращение к процедуре

Procedure declaration — описание процедуры

Procedure header — заголовок процедуры

Procedure statement — вызов процедуры, обращение к процедуре, оператор процедуры

Process address space — адресное пространство задачи

Process — процесс

Processing program — обрабатывающая программа

Processor — процессор

Processor unit — процессорный элемент

ProFile — цветовой профиль

ProFiler — профилировщик, профайлер

Program — программа

Program block structure — блочная структура программы

Program block — блок программы, программный блок

Program Branch — ветвь программы

Program complex — комплекс программ

Program debugging — отладка программы

Program development — разработка программ

Program development system — система разработки программ

Program entry — вход в программу

Program error — ошибка в программе, программная ошибка

Program flowchart — блок-схема программы, схема программы

Program Interface — программный интерфейс

Program library — библиотека программ, библиотека подпрограмм

Program loading — загрузка программы

Program object — объект программы, программный объект

Program platform — программная платформа

Program portability — переносимость программы, мобильность программы

Program product maintenance — сопровождение программного изделия

Program product — (коммерческий) программный продукт, программное изделие

Program proFile — профиль программы

Program run — выполнение программы, исполнение программы, прогон программы

Program segment — сегмент программы

Program start — запуск программы

Program testing — испытание программы, тестирование программы

Program trace — трассировка программы

Program translation — трансляция программы

Program unit — программный модуль

Program verification — верификация программы

Programmer — программист

Programmer-engineer — инженер-программист

Programming — программирование

Programming environment — среда программирования

Programming language semantics — семантика языка программирования

Programming language syntax — синтаксис языка программирования

Programming language — язык программирования

Programming lexicon — лексикон программирования

Programming methodology — методология программирования

Programming model — модель программирования

Programming paradigm — парадигма программирования

Programming system — система программирования (СП)

Progress indicator — прогресс-индикатор

Prolog — Пролог (язык программирования)

Prompt — подсказка

Prompt — приглашение ОС, подсказка

Property — свойство класса, свойство

Protected class method — защищенный метод класса

Protected class property — защищенное свойство класса

Protected method — защищенный метод класса

Protected mode — защищенный режим

Protected property — защищенное свойство класса

Protection key — ключ защиты

Protocol — протокол

Prototype — прототип

Provider — провайдер

Proxy server — сервер полномочий

Pseudocode — псевдокод

Pseudographics character — псевдографический символ

Public area — общая область

Public class method — открытый метод класса

Public class property — открытое свойство класса

Public domain — бесплатное программное обеспечение, свободное программное обеспечение

Public method — открытый метод класса

Public property — открытое свойство класса

Publishing system — редакционно-издательская система

Pull-down menu — нисходящее меню, раскрывающееся меню, спускающееся меню

Punch card — перфокарта, перфорационная карта

PVM (parallel virtual machine) — параллельная виртуальная машина, технология PVM

PVM-program — PVM-программа

PVM programming system — система программирования PVM

PC (Personal Computer) — персональный компьютер, персональная ЭВМ (ПК, ПЭВМ)

Q

Quality Assurance (QA), QA — контроль качества

Query processing — обработка запроса

Queue — очередь

Quick keys — клавиши быстрого вызова, быстрые клавиши

Quit — выход из приложения

Quoting — цитирование

R

RAD (Rapid Application Development) — быстрая разработка приложений

RAID (Redundant Array of Independent Disks) — массивы RAID

RAM (Random Access Memory) — память произвольного доступа, память прямого доступа

RAM disk (Random Access Memory disk) — электронный диск, виртуальный диск, псевдодиск

Random number generator — генератор случайных чисел

Rank — номер процесса, ранг

Rapid Application Development (RAD) — быстрая разработка приложений

Raster font — растровый шрифт

Raster graphics — растровая графика

Raster — растр

Rasterization — растрирование, растеризация

RDBMS (Relational Data Base Management System) — реляционная СУБД, система управления реляционными базами данных (СУРБД)

Read only memory (ROM) — постоянное запоминающее устройство (ПЗУ)

Read-write head — головка чтения/записи

Real constant — вещественная константа, действительная константа

Real mode — реальный режим

Real number — вещественное число, действительное число

Real type number — число вещественного типа

Real type — вещественный тип, действительный тип, тип Real

Real variable — вещественная переменная, действительная переменная

Reboot — перезагрузка, повторная начальная загрузка

Record key — ключ записи

Record locking — захват записи, блокировка записи

Record type — тип "запись"

Recovery — восстановление, регенерация

Recovery disk — диск аварийного восстановления

Recursive function — рекурсивная функция

Recursive procedure — рекурсивная процедура

Recursive subroutine — рекурсивная подпрограмма

Reduced instruction set computer (RISC) — компьютер с упрощенной системой команд, RISC-компьютер, архитектура RISC

Redundancy — избыточность

Redundant array of independent disks (RAID) — массивы RAID

Reenterability — реентерабельность, повторная входимость

Reenterable — повторно входимый

Reference — ссылка

Refinement — уточнение

REG File — файл реестра

Regeneration — регенерация, восстановление

Region — зона

Register — регистр

Register capacity — емкость регистра

Register length — длина регистра

Register size — размер регистра

Registry — реестр

Registry key — ключ реестра

Regression testing — регрессионное тестирование

Reification — воплощение

Relation — отношение

Relational data base management system (RDBMS) — реляционная СУБД, система управления реляционными базами данных (СУРБД)

Relational database — реляционная база данных

Relationship — отношение

Relative address — относительный адрес

Release — версия

Relevancy — релевантность

Relocatability — перемещаемость, переносимость

Remote access — удаленный доступ

Remote procedure call (RPC) — удаленный вызов процедур

Rendering — визуализация

Repeat-until loop — цикл с постусловием, цикл с нижним окончанием

Replica — реплика

Replication — репликация

Repository — репозиторий

Request, **query** — запрос

Requirement — требование

Requirements analysis — анализ требований

Rerun, **restart** — перезапуск, повторный пуск, рестарт

Reserved word — зарезервированное слово, служебное слово

Reset — возврат в исходное состояние

Resident program — резидентная программа

Resident software — резидентная программа

Resolution — разрешающая способность, разрешение

Resource — ресурс

Resource allocation — распределение ресурсов

Response Time — время ответа, время отклика, время реакции

Restart — рестарт

Return — возврат

Return instruction — команда возврата

Return point — точка возврата

Return statement — оператор возврата

Reuse — повторное использование

Reverse engineering — обратное проектирование

Revisions — исправления

Rewrite — перезапись

RFC document — документ RFC

RFC standard — стандарт RFC

Rich text format (RTF) — формат RTF, расширенный текстовый формат

Right — полномочие

Right significant digits — верные значащие цифры

Ring network — кольцевая сеть, сеть кольцевой топологии

RIP (Routing information protocol) — протокол маршрутной информации, протокол RIP

RISC (Reduced instruction set computer) — компьютер с упрощенной системой команд, RISC-компьютер, архитектура RISC

RLL encoding (Run-length limited encoding) — кодирование с ограничением длины поля записи, RLL-кодирование

Robot — робот поисковой системы

Rollback — откат транзакции, откат

ROM (Read Only Memory) — постоянное запоминающее устройство (ПЗУ)

Root directory — корневой каталог, главный каталог

Root segment — корневой сегмент

Route — маршрут

Router — маршрутизатор

Routine set — комплекс программ

Routing — маршрутизация

Routing information protocol (RIP) — протокол маршрутной информации, протокол RIP

RPC (Remote procedure call) — удаленный вызов процедур

RSA enciphering — шифрование с открытым ключом, несимметричное шифрование

RSA encryption — шифрование с открытым ключом, несимметричное шифрование

RTF (Rich text format) — формат RTF, расширенный текстовый формат

RTF File — файл RTF

Ruler — линейка

Run — выполнение программы, исполнение программы, прогон программы

Run time — время прогона, время выполнения

Run-length limited encoding (RLL encoding) — кодирование с ограничением длины поля записи, RLL-кодирование

Running foot — нижний колонтитул

Running head — верхний колонтитул

Running time — время прогона, время выполнения

S

Save — сохранение

Scalability — масштабируемость

Scanner — сканер

Scenario — сценарий

Scheduler — планировщик

Scheduling — планирование

Scope — область действия

Scrambler — шифратор

Screen — экран

Screen dump — дамп экрана, дамп

Screen panel, — панель экрана, экранная панель, панель

Screen saver — экранная заставка

Script — скрипт, сценарий

Scripting language — язык программирования сценариев, язык сценариев

Scroll bar — линейка прокрутки, полоса прокрутки

Scroll box — бегунок, движок

Scrolling — скроллинг, прокрутка (изображения, текста)

SCSI (Small computer system interface) — интерфейс SCSI (читается "интерфейс скази"), интерфейс малых вычислительных систем

SCSI devices — устройства SCSI (читается "устройства скази")

SCSI manager — диспетчер устройств SCSI (читается "диспетчер устройств скази")

SDL (Specification and definition language) — язык спецификации и определения

Search by field — поиск в определенных полях

Search engine — поисковая система

Secondary key — неосновной ключ, вторичный ключ

Secondary window — вторичное окно

Secret key enciphering — шифрование с закрытым ключом

Secret key encryption — шифрование с закрытым ключом

Sector — сектор

Secure electronic Transaction (SET) — защищенные электронные транзакции, протокол SET

Secure sockets layer (SSL) — протокол SSL

Security ID (SID) — идентификатор безопасности, идентификатор защиты, идентификатор доступа

Security identifier (SID) — идентификатор безопасности, идентификатор защиты, идентификатор доступа

Security policy — политика безопасности

Security system — система безопасности

Seek — поиск дорожки, подвод головки

Seek Time — время поиска

SEG (network Segment) — сегмент сети, сегмент

Segment — сегмент памяти, сегмент сети, сегмент

Segmentation — сегментация

SEI (Software Engineering Institute) — институт технологии разработки программного обеспечения

Select — выделение

Semantic error — семантическая ошибка

Semaphore — семафор

Sending instruction — команда пересылки

Sentence — предложение, оператор

Sequence diagram — диаграмма последовательности

Sequential access — последовательный доступ

Sequential file — файл последовательного доступа, последовательный файл

Serial access — последовательный доступ

Serial algorithm — линейный алгоритм

Serial number type — порядковый тип

Serial port — последовательный порт

Server — сервер

Service — сервис, служба

Service program — обслуживающая программа, сервисная программа, утилита

Service provider — поставщик сетевых услуг, провайдер

Session layer — сеансовый уровень

Session — сеанс, сеанс связи

SET (Secure electronic transaction) — защищенные электронные транзакции, протокол SET

Setup — установка

SGML (Standard generalized markup language) — язык SGML, стандартный обобщенный язык разметки

Shadow file — теневой файл

Shareable memory — разделяемая память, общая память

Shareable resource — общий ресурс, разделяемый ресурс, совместно используемый ресурс

Shared folder — общая папка

Shared memory — разделяемая память, общая память

Shared printer — общий принтер

Shared resource — общий ресурс, разделяемый ресурс, совместно используемый ресурс

Shareware — условно бесплатное программное обеспечение

Shell — оболочка (операционной системы)

Shell extension — надстройка оболочки

Shortcut key — командная клавиша

Shortcut menu — контекстное меню

SID (Security IDentifier) — идентификатор безопасности, идентификатор защиты, идентификатор доступа

Signal — сигнал

Signature — сигнатура

Significant digit — значащая цифра, значащий разряд

SIMD-computer — SIMD-компьютер

Simple network management protocol (SNMP) — протокол SNMP

Simultaneous processing — параллельная обработка

Single progam multiple data (SPMD) — SPMD-модель

Singleton — одиночка

SISD-computer — SISD-компьютер

Site — сайт

Slash — слэш

Slave mode — непривилегированный режим

Slot — слот

Small caps — капитель, малые прописные

Small computer system interface (SCSI) — интерфейс малых вычислительных систем, интерфейс SCSI (читается "интерфейс скази")

Smooth — сглаживание

SMP-computer — SMP-компьютер

SNA (Systems network architecture) — системная сетевая архитектура, архитектура SNA

SNA protocol — протокол SNA

SNMP (Simple network management protocol) — протокол SNMP

Smart card — смарт-карта

Socket — сокет, сокет-канал

Software architecture — архитектура программного обеспечения

Software development process — процесс разработки программного обеспечения

Software documentation — документация программного продукта, программная документация

Software engineer — системный инженер

Software Engineering Institute (SEI) — институт технологии разработки программного обеспечения

Software interrupt — прерывание прикладной программы, программное прерывание

Software piracy — незаконное копирование программных средств, несанкционированное копирование, пиратство

Software programmer — системный программист

Software tools — инструментальные программные средства

Software — программное изделие, программное обеспечение, программные средства

Softwared — программный

Sort — сортировка, упорядочение

Sound board — звуковая плата, звуковой адаптер

Sound card — звуковая карта, звуковой адаптер

Sound track — звуковая дорожка

Source address — адрес операнда

Source data — исходные данные

Source language — исходный язык системы программирования

Source module — исходный модуль

Source program text — исходный текст программы, исходная программа

Source program — исходная программа

Source text — исходный текст

Source — источник

Spam — спам

Speaker — звуковая колонка

Special character — специальный символ

Special keys — специальные клавиши

Specification and definition language (SDL) — язык спецификации и определения

Specification — спецификация, описание

Specifier — спецификатор, описатель

Speech analysis — распознавание речи

Speech generation — синтез речи

Speech recognition — распознавание речи

Speech synthesis — синтез речи

Speed — быстродействие

Spelling checker — блок орфографического контроля, корректор, проверка орфографии

Spider — робот поисковой системы

Spin lock — спин-блокировка

SPMD (Single progam multiple data) — SPMD-модель

Spool File — файл спулинга, буферный файл, спул-файл

Spooler — программа буферизации, спулер, спулер печати

Spooling — спулинг

Spreadsheet — электронная таблица

SQL (Structured query language) — язык запросов SQL (читается "сиквэл")

SSL (Secure Sockets Layer) — протокол SSL

Stack — стек, магазин

Stand-alone program tests — автономные испытания программы

Standard function — стандартная функция

Standard generalized markup language (SGML) — язык SGML, стандартный обобщенный язык разметки

Standard procedure — стандартная процедура

Standard subroutine — стандартная подпрограмма

Standard type — стандартный тип

Standby — резервирование

Star network — радиальная сеть, звездообразная сеть, сеть звездообразной топологии

Starvation — зависание

State — состояние

Statechart diagram — диаграмма состояний

Statement — оператор

Statement brackets — операторные скобки

Statement function — оператор-функция

Static memory allocation — статическое распределение памяти

Static resource allocation — статическое распределение ресурсов

Status bar — строка состояния

Status Word — слово состояния

Stemming — морфологический поиск

Stereotype — стереотип

Stockholders — заинтересованные лица

Stop — останов

Stop words — стоп-слова

Storage — память компьютера, память ЭВМ, память

Storage allocation — распределение памяти

Storage area — область памяти

Storage block — блок памяти

Storage device — запоминающее устройство (ЗУ)

Storage hierarchy — иерархия памяти

Storage medium — запоминающая среда

Storage protection key — ключ защиты памяти

Store — память компьютера, память ЭВМ, память

Streamer — стример

String — строка, строка символов

Structure type — структурированный тип

Structured programming — структурное программирование

Structured query language (SQL) — язык запросов SQL (читается "сиквэл")

Style — стиль

Stylus — перо

Subclass — подкласс

Subdirectory — подкаталог

Submachine — вложенный автомат

Submenu — подменю

Subroutine call — вызов подпрограммы, обращение к подпрограмме

Subroutine entry — вход в подпрограмму

Subroutine library — библиотека подпрограмм

Subroutine — подпрограмма

Subscript — нижний индекс

Subscript boundary — граница индекса

Subscripted variable — индексированная переменная, переменная с индексами

Substitutability principle — принцип подстановочности

Subtractive color — субтрактивный цвет

Super scalar architecture — суперскалярная архитектура

Superclass — суперкласс

Supercomputer — суперЭВМ, суперкомпьютер

Superscalar processor — суперскалярный процессор

Supervisor — супервизор

Supervisor interrupt — прерывание по обращению к супервизору

Support — поддерживать, поддержка, сопровождение программного изделия

SVC-interrupt — прерывание по обращению к супервизору

Swap file, Swapping file — файл подкачки, файл замещения страниц

Swapping — свопинг, подкачка

Swim lane — плавательная дорожка

Switch — переключатель

Switch off — выключить

Switch on — включить

Switching — коммутация

Symbol — символ

Symmetric enciphering — симметричное шифрование

Symmetric encryption — симметричное шифрование

Symmetric multi processor computer — симметричный многопроцессорный компьютер

Synchronous connection — синхронная связь, синхронная передача

Synchronous transmission — синхронная связь, синхронная передача

Syntax — синтаксис

Syntax construction — синтаксическая конструкция

Syntax error — синтаксическая ошибка

System administrator — системный администратор

System area — системная область

System block — системный блок

System catalogue — системный каталог

System disk — системный диск,

System file — системный файл

System folder — системная папка

System generation — генерация системы

System library — системная библиотека

System message — системное сообщение

System modal dialog box — системное модальное диалоговое окно

System modal message box — модальное окно системных сообщений

System modal window — системное модальное окно

System network architecture protocol — протокол SNA

System program — системная программа

System programmer — системный программист

System software — системное программное обеспечение

Systems network architecture (SNA) — системная сетевая архитектура, архитектура SNA

T

Tabbed dialog box — диалоговое окно с вкладками

Tag — тег

Tagged image file format (TIFF) — формат TIFF, тегированный формат файлов изображений

Tagged value — именованное значение

TAPI (Telephony application programming interface) — интерфейс программирования приложений для телефонной связи, интерфейс TAPI

Target — адресат

Task bar — панель задач

Task image — образ задачи

Task — задача

TB (Terabyte) — Терабайт, Тб, Тбайт

TCP/IP (Transmission control protocol/Internet protocol) — протоколы TCP/IP

Technology — технология

Teleconference — телеконференция

Telephony application — телефонное приложение

Telephony application programming interface (TAPI) — интерфейс программирования приложений для телефонной связи, интерфейс TAPI

Telephony server — телефонный сервер

Telnet — протокол Telnet

Temp file — временный файл, рабочий файл

Template — шаблон

Temporary file — временный файл, рабочий файл

Terabyte (TB) — Терабайт, Тб, Тбайт

TeraFLOPS — Терафлопс, Тфлопс

Terminal — терминал

Termination — завершение программы

Text box — текстовое поле

Text editor — текстовый редактор

Text File — текстовый файл

Text formatting — форматирование текста

Text — текст

TFLOPS — Терафлопс, Тфлопс

Thesaurus — тезаурус

Thread — поток управления

Thunk — код переключения, код трансформации, переходник

TIFF (Tagged image file format) — формат TIFF, тегированный формат файлов изображений

TIFF File — файл TIFF

Time sharing — разделение времени

Time slice — квант времени

Time slicing — квантование времени

Timer — таймер

Timer clock — таймер

Time-sharing operating system (TSS) — система разделения времени

Token Bus network — шинная сеть с маркерным доступом, шинная сеть с передачей маркера, маркерная шинная сеть

Token network — сеть с маркерным доступом, сеть с передачей маркера, маркерная сеть

Token ring network — кольцевая сеть с маркерным доступом, кольцевая сеть с передачей маркера, маркерная кольцевая сеть

Toner — тонер

Toolbar — панель инструментов, инструментальная панель

Tools — инструментальные программные средства

Toolsmith — системный программист

Top-down programming — нисходящее программирование

Touch screen — сенсорный экран

Track — дорожка

Track ball — трекбол, шаровой манипулятор

Traffic — трафик, посещаемость

Transaction commitment — завершение транзакции

Transaction Processing — обработка транзакций

Transaction rollback — откат транзакции

Transaction — транзакция

Transformation — трансформация

Transient object — временный объект

Transition — переход

Translator — транслятор

Transmission control protocol/Internet protocol (TCP/IP) — протоколы TCP/IP

Transport layer — транспортный уровень

Transport protocol — транспортный протокол, протокол транспортного уровня

Transputer — транспьютер

Trap instruction — команда прерывания

Trap — ловушка

Tree network — древовидная сеть, сеть древовидной топологии

Tree — дерево

Tree topology — древовидная топология

Trigger — переключающее событие

TrueType font — шрифт TrueType

Truncation condition — условие выхода из цикла, условие завершения цикла

Trunk — магистраль

Trust — доверительное отношение

TSD (Type-Specific Driver) — драйвер логического устройства

TSS (Time-sharing operating system) — система разделения времени

Tuning — калибровка

Tuple — кортеж

Turing machine — машина Тьюринга

Tweening — преобразование с промежуточными формами, твининг

Twip — твип

Two-dimensional array — двумерный массив

Type conversion — приведение типов, преобразование типов

Type declaration — описание типа, определение типа

Type family — гарнитура шрифта

Type size — размер шрифта, кегль шрифта

Type style — начертание шрифта, стиль шрифта

Type writer keys — клавиши пишущей машинки

Typeface — гарнитура шрифта

Typematic — автоматический повтор нажатия клавиши

Type-specific driver (TSD) — драйвер логического устройства

TeX — TEX

U

UAE (Unrecoverable application error) — невосстановимая ошибка приложения

UART (Universal asynchronous receiver-transmitter) — универсальный асинхронный приемопередатчик (УАПП)

UMA-computer — UMA-компьютер, компьютер с однородным доступом к памяти

UMB (Upper memory blocks) — верхняя память

UML (Unified modeling language) — унифицированный язык моделирования, язык UML

Unary operation — одноместная операция, унарная операция

Unauthorized access — несанкционированный доступ

Unconditional Branch instruction — команда безусловной передачи управления, команда безусловного перехода

Unconditional branch — безусловная передача управления, безусловный переход

Unconditional JUMP instruction — команда безусловной передачи управления, команда безусловного перехода

Undefined format — неопределенный формат

Underline — подчеркивание

Undo — откат

UniBus — магистраль

Unicode — стандарт Unicode

Unidriver — универсальный драйвер принтера

Unified modeling language (UML) — унифицированный язык моделирования, язык UML

Uniform memory access computer — компьютер с однородным доступом к памяти

Uniform resource locator (URL) — универсальный указатель ресурса, универсальный локатор ресурса

Unit — устройство, техническое устройство

Universal asynchronous receiver-transmitter (UART) — универсальный асинхронный приемопередатчик (УАПП)

Universal printer driver — универсальный драйвер принтера

UNIX — операционная система UNIX (ОС UNIX)

Unpack — разархивирование

Unrecoverable application error (UAE) — невосстановимая ошибка приложения

Unrecoverable error — неисправимая ошибка

Update — модификация, обновление

Updating — модификация, обновление

Upgrade — усовершенствование, обновление

Upgrading — усовершенствование, обновление

Uploading — выгрузка по линии связи

Upper memory blocks (UMB) — верхняя память

Uppercase — верхний регистр

Upward compatibility — совместимость снизу вверх

URL (Uniform resource locator) — универсальный указатель ресурса, универсальный локатор ресурса

Use case diagram — диаграмма использования

Use case — вариант использования

User — пользователь

User account — учетная запись пользователя, бюджет пользователя

User environment — рабочая среда пользователя, среда

User interface — интерфейс пользователя, пользовательский интерфейс

User local profile — локальный профиль пользователя

User mode — режим пользователя, пользовательский режим

User name — имя пользователя

User node — абонент сети

User profile — профиль пользователя

Utility — утилита, обслуживающая программа, сервисная программа

V

Value — значение

Variable — переменная

Variable declaration — описание переменной

VB (Visual Basic) — язык Visual Basic

VBScript — язык сценариев VBScript, язык VBScript

Vector — вектор

Vector font — векторный шрифт

Vector graphics — векторная графика

Vector instruction — векторная команда

Vector processor — векторный процессор

Vector register — векторный регистр

Version — версия

Vertical menu — вертикальное меню

Very large instruction word processor — процессор со сверхдлинным командным словом

VESA (Video Electronics Standards Association) — ассоциация стандартов по видеотехнике, индустриальный консорциум VESA, ассоциация VESA

VESA local Bus (VLB) — шина VLB, локальная шина VESA

Video adapter — видеоадаптер

Video data — видеоданные

Video electronics standards association (VESA) — ассоциация стандартов по видеотехнике, индустриальный консорциум VESA, ассоциация VESA

Video information — видеоинформация

Video memory — видеопамять

Video RAM — видеопамять

Video storage — видеопамять

Video system — видеосистема компьютера

Video terminal — видеотерминал

Videocard — адаптер дисплея, видеоадаптер, видеокарта

Videocodec — видеокодек

Videodisk — лазерный диск, видеодиск

Viewer — программа просмотра

Virtual address — виртуальный адрес

Virtual anything driver (VxD) — драйвер виртуального устройства (x — начальная буква конкретного устройства, например, VPD — драйвер виртуального принтера)

Virtual connection — виртуальное соединение, логическое соединение

Virtual disk — виртуальный диск, псевдодиск

Virtual function — виртуальная функция

Virtual machine — виртуальная машина

Virtual memory — виртуальная память

Virtual memory manager (VMM) — диспетчер виртуальной памяти

Virtual reality modeling language (VRML) — язык моделирования виртуальной реальности

Virtual storage — виртуальная память

Virtual — виртуальный

Virus — вирус

Visibility — область видимости

Visual Basic (VB) — язык Visual Basic

Visual modeling language — язык визуального моделирования

Visual programming language — визуальный язык программирования

VLB (VESA Local Bus) — шина VLB, локальная шина VESA

VLIW-processor — процессор со сверхдлинным командным словом, VLIW-процессор

VMM (Virtual memory manager) — диспетчер виртуальной памяти

Vocoder — вокодер

Voice mail — голосовая почта

Voice modem — голосовой модем

Voice synthesizer — синтезатор речи

Von Neumann architecture — архитектура фон Неймана, фон-неймановская архитектура

V-register — векторный регистр

VRML (Virtual Reality Modeling Language) — язык моделирования виртуальной реальности

VxD — драйвер виртуального устройства (x — начальная буква конкретного устройства, например, VPD — драйвер виртуального принтера)

W

W3C (World Wide Web Consortium) — Консорциум World Wide Web

WAN (Wide area network) — региональная вычислительная сеть

Warning — предупреждение

Waterfall software development process — водопадный процесс разработки программного обеспечения

Web document — документ Web, документ WWW

Web page — страница Web, веб-страница

Web portal — портал, Web-портал

Web site — сайт, веб-сайт, сайт Web,

While loop — цикл с предусловием, цикл с верхним окончанием, цикл "пока"

Wide area network (WAN) — региональная вычислительная сеть

Widow — висячая строка

Winchester disk — винчестерский диск, винчестер

Window — окно

Window of execution — окно выполнения

Windows — операционная система Windows, окна

Windows Explorer — проводник Windows

Windows NT File system (NTFS) — файловая система Windows NT, файловая система NTFS

Windows registry — реестр Windows

Windows scripting host (WSH) — сервер сценариев Windows

Windows Sockets (WinSoc) — программное обеспечение Windows Sockets, ПО Windows Socket

WinSoc (Windows Sockets) — программное обеспечение Windows Sockets, ПО Windows Socket

Wire frame — контурный режим

Wizard — мастер

Word — слово

Word art — заголовочный текст, фигурный текст

Word processing — подготовка текстов

Word processor — текстовый процессор

Word wrap — выравнивание текста, автоматический переход на новую строку

Work area — рабочая область

Workbook — рабочая книга

Working directory — рабочий каталог

Working environment — рабочая среда пользователя, среда

Working set — рабочий набор процесса

Workspace — рабочая область

Workstation — рабочая станция, автоматизированное рабочее место (АРМ)

World Wide Web (WWW) — Всемирная паутина

World Wide Web Consortium (WWWC, W3C) — Консорциум World Wide Web

Write-back caching — кэширование с обратной записью

Write-through caching — кэширование с прямой записью

Writing, record — запись

WSH (Windows Scripting Host) — сервер сценариев Windows

WWW (World Wide Web) — Всемирная паутина

WWW Document — документ Web, документ WWW

WWWC (World Wide Web Consortium) — Консорциум World Wide Web

X

XMI (XML metadata interchange) — стандарт XMI

XML (Extensible markup language) — язык XML, расширяемый язык разметки

XMM (Extended memory manager) — диспетчер расширенной памяти

XMS (Extended memory) — расширенная память

Z

Zero filling — обнуление

Zooming — изменение размера окна

Список литературы

Словари и справочные пособия

1. Англо-русский словарь пользователя IBM PC: 8200 слов и выражений / Сост. Е. З. Друкер, П. З. Друкер, В. В. Собоцинский. — М.: ВАРД, 1992. — 192 с.
2. Англо-русский толковый словарь терминов//"Мир ПК" № 1, 4, 8, 9, 10. — М.: АО "Информэйшн. Компьютер Энтерпрайз", 1994.
3. Борковский А. Б. Англо-русский словарь по программированию и информатике (с толкованиями). Ок. 6000 терминов. — М.: Московская международная школа переводчиков, 1992. — 335 с.
4. Борковский А. Б., Зайчик Б. И., Боровикова Л. Т. Словарь по программированию (английский, русский, немецкий, французский). Ок. 5000 терминов. — М.: Русский язык, 1991.
5. Ваулина Е. Ю. Толковый словарь пользователя PC. — СПб.: Изд-во "Антон", 1998. — 176 с.
6. Дорот В. Л., Новиков Ф. А. Толковый словарь современной компьютерной лексики. — СПб.: БХВ — Санкт-Петербург, 1999. — 384 с.
7. Дорот В. Л., Новиков Ф. А. Толковый словарь современной компьютерной лексики. — 2-е изд., перераб. и доп. — СПб.: БХВ — Санкт-Петербург, 2001. — 512 с.
8. Ершов А. П., Шанский Н. М., Окунева А. П., Баско Н. В. Школьный терминологический словарь по основам информатики и вычислительной техники / Под ред. А. П. Ершова. — М.: Просвещение, 1991. — 159 с.
9. Информатика: Энциклопедический словарь для начинающих / Сост. Д. А. Поспелов. — М.: Педагогика-Пресс, 1994. — 352 с.
10. Лемешко Е. Б., Лемешко Н. Н. Англо-русский словарь новейшей компьютерной терминологии / Под ред. К. Ахметова. — М.: Компьютер-Пресс, 1998. — 224 с.
11. Максимов В. И., Дорот В. Л., Зайцев Б. Е. и др. Иллюстрированный толковый словарь русской научной и технической лексики: 7671 терминов / Под ред. В. И. Максимова. — М.: Русский язык, 1994. — 800 с.

12. Мячев А. А., Красавин А. Н., Алексеев Е. С. Персональные ЭВМ. Толковый словарь. Англо-русский словарь сокращений. — М.: Радио и связь, 1993. — 96 с.

13. Microsoft Press. Толковый словарь по вычислительной технике: Пер. с англ. Под ред. А. И. Козлова. — М.: Издательский отдел "Русская редакция" ТОО "Channel Trading Ltd.", 1995. — 496 с.

14. Першиков В. И., Савинков В. М. Толковый словарь по информатике. — М.: Финансы и статистика, 1995. — 544 с.

15. Пройдаков Э. М., Теплицкий Л. А. Англо-русский толковый словарь по вычислительной технике, Интернету и программированию. — 2-е изд., испр. и доп. — М.: Издательско-торговый дом "Русская редакция", 2000. — 448 с.

16. Толковый словарь по вычислительным системам / Под ред. В. Иллингуорта и др.: Пер. с англ. А. К. Белоцкого и др. / Под ред. Е. К. Масловского. — М.: Машиностроение, 1991. — 560 с.

17. Травин А. Толковый словарь "Термины и жаргон Интернета" из кн. "Популярные браузеры World Wide Web". — М.: Изд-во ABF, 1997.

Учебные пособия, популярная и специальная литература

1. Брауде Э. Технология разработки программного обеспечения. — СПб.: Питер, 2004. — 655 с.

2. Рамбо Дж., Якобсон А., Буч Г. UML: специальный справочник. — СПб.: Питер, 2002. — 656 с.

3. Васильев В. П., Малиновский А. Н. Основы работы на ПК. — СПб.: БХВ — Санкт-Петербург, 2000. — 448 с.

4. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. — СПб.: БХВ-Петербург, 2002. — 608 с.

5. Галичский К. В. Компьютерные системы в телефонии. — СПб.: БХВ-Петербург, 2002. — 400 с.

6. Дорстер А. IBM-совместимые персональные компьютеры: Краткое руководство пользователя. — СПб.: Издание автора, 1994. — 105 с.

7. Дьюхарст С., Старк К. Программирование на C++: Пер. с англ. — К.: "ДиаСофт", 1993. — 272 с.

8. Жаров А. Железо IBM. — М.: фирма "Микроарт", 1996. — 200 с.

9. Йенсен К., Вирт Н. Паскаль. Руководство для пользователя: Пер. с англ. М. Л. Сальникова, Ю. В. Сальниковой. — М.: Компьютер, 1993. — 256 с.

10. Кокорева О. И. Реестр Windows 2000. — СПб.: БХВ — Санкт-Петербург, 2000. — 352 с.

11. Колесниченко О. В., Шишигин И. В. Аппаратные средства PC. — 3-е изд., перераб. и доп. — СПб.: БХВ — Санкт-Петербург, 1999. — 800 с.

12. Крол Э. Все об Internet: Пер. с англ. — К.: Торгово-издательское бюро BHV, 1995. — 592 с.

13. Марченко А. И., Пасько В. П. Word 7.0 для Windows 95. — К.: Торгово-издательское бюро BHV, 1996. — 464 с.

14. Матросов А. В., Новиков Ф. А., Усаров Г. Е., Харитонова И. А. MS Office XP: разработка приложений / Под ред. Новикова Ф. А. — СПб.: БХВ-Петербург, 2003. — 944 с.

15. Немнюгин С. А., Стесик О. Л. Параллельное программирование для многопроцессорных вычислительных систем. — СПб.: БХВ-Петербург, 2002. — 400 с.

16. Новиков Ф. А., Яценко А. Д. Microsoft Office в целом. — СПб.: BHV — Санкт-Петербург, 1995. — 331 с.

17. Новиков Ф. А., Яценко А. Д. Microsoft Office 97 в целом. — СПб.: BHV — Санкт-Петербург, 1998. — 624 с.

18. Нортон П., Гудман Дж. Персональный компьютер: аппаратно-программная организация: Пер. с англ. — СПб.: BHV — Санкт-Петербург, 2000. — 848 с.

19. Омельченко Л. Н. Федоров А. Ф. Самоучитель Windows 2000 Professional. — СПб.: БХВ — Санкт-Петербург, 2000. — 528 с.

20. Пилгрим А. Персональный компьютер: модернизация и ремонт: Пер. с англ. — СПб.: BHV — Санкт-Петербург, 2000. — 528 с.

21. Пономаренко С. И. Adobe Photoshop 5.0. — СПб.: БХВ — Санкт-Петербург, 1999. — 512 с.

22. Пономаренко С. И. CorelDRAW 9. — СПб.: БХВ — Санкт-Петербург, 1999. — 560 с.

23. Райманс Х. Г. Вводный курс Visual Basic: Пер. с нем. — К.: Торгово-издательское бюро BHV, 1993. — 272 с.

24. Фигурнов В. Э. IBM PC для пользователя. — М.: Финансы и статистика, 1990. — 240 с.

25. Хеслоп Б., Бадник Л. HTML с самого начала: Пер. с англ. — СПб.: Питер, 1997. — 416 с.

26. Школьник и компьютер: учимся друг у друга / Под ред. В. В. Прохорова. — М.: Физматлит, 1993. — 208 с.

27. Щербаков А. Защита от копирования. — М.: Изд-во "Эдэль", 1992. — 80 с.

28. Электронная коммерция. B2B-программирование: Пер. с англ. — СПб.: БХВ-Петербург, 2001. — 368 с.