

А. Гарнаев

Использование **MS Excel и VBA** в экономике и финансах

Мощные средства финансового анализа

Моделирование экономических сценариев

Поиск оптимальных решений

Более 100 примеров



Андрей Гарнаев

**Использование
MS Excel и VBA
в экономике и финансах**



Дюссельдорф Киев
Москва Санкт-Петербург

УДК 681.3.06

Книга предназначена для читателей, интересующихся применением вычислительной техники в менеджменте и экономике. В ней рассмотрено использование Excel и языка Visual Basic for Applications для решения широкого круга задач: от составления простого финансового отчета до создания сценариев серьезных экономических проектов. На большом количестве примеров описан современный инструментарий принятия оптимальных решений в экономике.

Книга имеет выраженную учебную направленность, поэтому будет полезна не только экономистам, специалистам по менеджменту, финансам и коммерции, но и преподавателям, студентам экономических специальностей.

Для широкого круга пользователей

Гарнаев А. Ю.

Использование MS Excel и VBA в экономике и финансах. — СПб.: БХВ — Санкт-Петербург, 1999. — 336 с., ил.

ISBN 5-8206-0024-X

© Гарнаев А. Ю., 1999

© Оформление, издательство "БХВ — Санкт-Петербург", 1999

Лицензия ЛР № 065953 от 15.06.98. Подписано в печать 24.02.99.

Формат 70×100 ¹/₁₆. Печать офсетная. Усл. печ. л. 27,1.

Тираж 3000 экз. Заказ

БХВ — Санкт-Петербург, 198005, С.-Петербург, Измайловский пр., 29.

Отпечатано с готовых диапозитивов
в Академической типографии "Наука" РАН
199034, Санкт-Петербург, 9 линия, 12.

Содержание

Предисловие	3
Глава 1. Основы работы в Excel	5
1.1. Рабочая книга Excel	5
1.2. Работа с файлами	8
1.2.1. Создание новой рабочей книги	8
1.2.2. Открытие существующей рабочей книги	10
1.2.3. Сохранение рабочей книги	11
1.2.4. Закрытие рабочей книги	12
1.2.5. Удаление файла рабочей книги	12
1.3. Окно приложения Excel	12
1.4. Операции с рабочими листами	17
1.5. Пример составления документа	20
1.5.1. Адресация ячейки	21
1.5.2. Ввод данных в ячейку и их редактирование	22
1.5.3. Ввод формулы в ячейку	24
1.5.4. Форматы данных	26
1.5.5. Диапазон ячеек	29
1.5.6. Выравнивание текста	31
1.5.7. Шрифты	32
1.5.8. Рамки и цвет фона	32
1.5.9. Изменение ширины столбцов и высоты строк	33
1.5.10. Специальные текстовые и графические эффекты	33
1.5.11. Параметры страницы, предварительный просмотр и печать документа	35
Глава 2. Построение графиков и решение нелинейных уравнений	38
2.1. Построение графика. Работа с мастером функций и мастером диаграмм	38
2.2. Построение графика функции с одним условием	47
2.3. Построение графика функции с двумя условиями	48
2.4. Построение двух графиков в одной системе координат	49
2.5. Построение поверхности	50
2.6. Нахождение корней уравнения	53

2.7. Нахождение корней уравнения методом деления отрезка пополам	55
2.8. Упражнения	57

Глава 3. Работа с массивами 65

3.1. Простейшие операции над массивами	65
3.2. Встроенные функции для работы с матрицами	67
3.3. Пошаговое решение системы линейных уравнений методом Гаусса	69
3.4. Упражнения	71

Глава 4. Финансовый анализ в Excel 74

4.1. Финансовая функция <i>ППЛАТ</i>	74
4.2. Пример расчета эффективности неравномерных капиталовложений с помощью функций <i>НПЗ</i> , <i>ВНДОХ</i> и <i>Подбор параметра</i>	76
4.3. Пример расчета эффективности капиталовложений с помощью функции <i>ПЗ</i>	79
4.4. Финансовые функции <i>ПЛПРОЦ</i> и <i>ОСНПЛАТ</i>	84
4.5. Финансовые функции <i>БЗ</i> , <i>КПЕР</i> и <i>НОРМА</i>	85
4.6. Функции для расчета амортизации <i>АМР</i> , <i>АМГД</i> , <i>ДОБ</i> и <i>ДДОБ</i>	88
4.7. Примеры простейших отчетных ведомостей	91
4.8. Пример отчетной ведомости по расчету просроченных платежей	98
4.9. Пример отчетной ведомости по расчету затрат на производство	100
4.10. Планирование рекламной кампании	102
4.11. Упражнения	106

Глава 5. Поиск решения 112

5.1. Линейная оптимизационная задача	112
5.1.1. Планирование производства красок	112
5.1.2. Элементы диалогового окна <i>Поиск решения</i>	116
5.1.3. Определение состава сплавов	120
5.1.4. Планирование штатного расписания	123
5.2. Транспортная задача	127
5.3. Решение системы нелинейных уравнений	130
5.4. Задача о назначениях	132
5.5. Уравнение регрессии	134
5.5.1. Общий подход к построению уравнения регрессии на примере линейной модели	135
5.5.2. Функции рабочего листа для уравнения линейной регрессии	136
5.5.3. Экспоненциальная модель	141

5.6. Транспортная задача с фиксированными доплатами	144
5.8. Упражнения	146
Глава 6. Сводные таблицы.....	161
Глава 7. Visual Basic for Applications (VBA)	170
7.1. Простейшие функции и процедуры. Типы переменных	171
7.1.1. Функция пользователя	171
7.1.2. Переменные и постоянные	175
7.1.3. Процедуры	178
7.2. Функции пользователя с операторами условного перехода <i>If-Then</i> и <i>If-Then-Else</i>	181
7.3. Функции пользователя с несколькими операторами условного перехода	184
7.4. Функции пользователя, используемые при расчете комиссионных	185
7.4.1. Функции пользователя с оператором условного перехода <i>If-Then</i>	185
7.4.2. Функции пользователя с оператором выбора <i>Select Case</i>	187
7.5. Стандартные функции для работы с массивами	188
7.6. Стандартные функции для работы с матрицами	190
7.7. Объекты, свойства и методы VBA.....	190
7.8. Операторы цикла	195
7.8.1. Оператор цикла <i>For-Next</i>	195
7.8.2. Оператор цикла <i>For-Each-Next</i>	199
7.8.3. Оператор цикла <i>While-Wend</i>	202
7.8.4. Операторы цикла <i>Do-Loop</i>	202
7.9. Панель инструментов <i>Элементы управления</i> . Процедуры перелистывания рабочих листов	206
7.9.1. Первый пример	207
7.9.2. Второй пример	209
7.9.3. Третий пример	211
7.9.4. Четвертый пример	211
7.9.5. Пятый пример	213
7.9.6. Шестой пример	214
7.10. Создание сценариев	215
7.11. Модель управления запасами	220
7.12. Задача об оптимальном раскрое	228
7.13. База данных отеля	231
7.14. Банковская база данных	238
7.15. Как разместить капитал?	255
7.16. Расчет краткосрочной ссуды с использованием диалогового окна	258

7.17. Расчет числа выплат с использованием диалогового окна	261
7.18. Расчет амортизации и ипотеки с использованием диалогового окна.....	264
7.18.1. Расчет амортизации	264
7.18.2. Расчет ипотеки.....	267
7.19. Расчет постоянных периодических выплат с построением диаграммы из диалогового окна.....	269
7.20. Программирование расчета годовой процентной ставки с помощью команды <i>Подбор параметра</i>	278
7.21. Программирование решения систем нелинейных уравнений с параметром.....	281
7.22. Оптимальное планирование сбыта продукции.....	284
7.23. Программирование рекламной кампании.....	291
7.24. Упражнения	294
Задания на вычисление комиссионных, программирование сценариев и оптимальное управление запасами.....	294
Задания по работе с массивами	296
Задания на работу с базами данных	301
Задания на нахождение оптимального раскроя	302
Задания на определение оптимальных капиталовложений.....	303
Приложение А. Отладка программ	307
Приложение В. Функции рабочего листа Excel.....	315
Литература	332

Предисловие

Успех в современном бизнесе и менеджменте во многом опирается на оперативный анализ экономической ситуации и выбор оптимального решения из возможных альтернатив. Одним из современных компьютерных средств для решения подобных задач является программа Microsoft Excel. Однако большинство книг, посвященных Excel, описывают лишь инструментальные возможности этой программы, не объясняя, как ими пользоваться при решении конкретных задач.

Восполнению этого пробела и посвящена данная книга. В ней представлена уникальная по количеству и разнообразию коллекция примеров, демонстрирующих возможности Excel по выполнению экономических расчетов и составлению финансовой документации. В частности, читатель научится решать следующие задачи:

- ❑ Планирование капиталовложений и сбыта продукции
- ❑ Управление запасами
- ❑ Планирование рекламных кампаний
- ❑ Создание сценариев экономических проектов
- ❑ Построение баз данных и работа с ними
- ❑ Разработка диалоговых окон для экономических расчетов

В главах 1–3 изложены основы работы в Excel. С функциями финансового анализа и их применением читатель познакомится в главе 4. В главе 5 описаны средства Excel, используемые для решения оптимизационных задач, а в главе 6 — средства для работы с базами данных. Глава 7 посвящена программированию на языке Visual Basic for Applications (VBA). Ознакомившись с этой главой, читатель научится более эффективно работать с электронными таблицами, а также решать многие прикладные задачи. В приложении А рассматриваются средства отладки программ, написанных на языке VBA, а в приложении В приводится список основных встроженных функций Excel.

В конце каждой главы имеется большое количество упражнений, позволяющих закрепить теоретический материал. Упражнения разбиты по вариантам, поэтому они могут быть использованы не только для самостоятельной работы, но и в качестве заданий на занятиях по информатике.

Данная книга может быть полезна не только тем, кто уже работает с программой Excel, но и тем, кто только приступает к ее изучению:

- ❑ Студентам — в процессе обучения и выполнения курсовых работ
- ❑ Преподавателям — для использования в лекциях и на практических занятиях
- ❑ Начинающим пользователям — для самостоятельного изучения
- ❑ Опытным пользователям — для расширения своих профессиональных возможностей

Типографские соглашения

Способ выделения	Применение
Полужирное начертание	Команды меню, кнопки панелей инструментов, заголовки и опции стандартных диалоговых окон Excel
Шрифт Courier	Заголовки и опции диалоговых окон, создаваемых пользователем; рабочие книги, листы, функции и формулы Excel; значения, вводимые пользователем или вычисляемые в рабочих таблицах; элементы языка программирования VBA, а также листинги и фрагменты программ
<клавиша>+<клавиша>	Комбинация клавиш, в которой сначала нажимается и удерживается первая клавиша, а затем нажимается вторая

Об авторе

Гарнаев Андрей Юрьевич, доктор физико-математических наук, профессор кафедры вычислительной математики Санкт-Петербургского Государственного Архитектурно-Строительного Университета. Область интересов: математическая кибернетика, программное обеспечение. Им опубликованы две монографии и более шестидесяти научных работ. Данная книга написана на основе многолетнего опыта преподавания Excel студентам экономических специальностей.

Глава 1



Основы работы в Excel

Добро пожаловать в изумительный и бескрайний мир Excel! Вы, дорогой читатель, можете быть как начинающим, делающим первые шаги пользователем, так и профессионалом, сломавшим большое число копий на захватывающем, полном неожиданностей и сюрпризов пути изучения Excel. Если вы начинающий пользователь, то наверняка вас заинтересует, что такое Excel, с какими документами он работает и как эти документы создавать. Первая глава как раз и посвящена этим вопросам, причем она составлена таким образом, что за минимальное время вы сможете приобрести максимальные практические навыки. Если вы профессионал, то можно пропустить эту главу и сразу перейти к интересующим вас специальным вопросам.

1.1. Рабочая книга Excel

Excel относится к программным продуктам, которые известны под названием электронные таблицы. Электронная таблица — это интерактивная программа, состоящая из набора строк и столбцов, изображенных на экране в специальном окне. Область, находящаяся на пересечении строки и столбца, называется ячейкой. В ячейке могут находиться число, текст или формула, с помощью которой осуществляются вычисления, относящиеся к одной или нескольким ячейкам. Ячейки можно копировать, перемещать, а также изменять их содержимое. При изменении содержимого ячейки производится автоматический пересчет содержимого всех ячеек, использующих в формулах измененную ячейку. На основе групп ячеек создаются диаграммы, сводные таблицы и карты. Электронную таблицу можно сохранить в отдельном файле для дальнейшего использования.

Создаваемые в Excel файлы называются *рабочими книгами* (WorkBook). Рабочая книга состоит из нескольких листов. Листы могут быть следующих типов:

- ❑ *Рабочий лист* (WorkSheet) — это самый распространенный лист, который обычно имеют в виду, говоря об электронной таблице
- ❑ *Лист модуля* (Module) — на этом листе пишутся программы Visual Basic for Applications
- ❑ *Лист диалогового окна* (DialogSheet) — на этом листе создаются диалоговые окна
- ❑ *Лист диаграмм* (ChartSheet) — на этом листе с помощью мастера диаграмм (Chart Wizard) можно строить диаграммы (обычно диаграммы строят на рабочем листе)

На рис. 1.1 показано окно Excel с тремя открытыми рабочими книгами. Окна рабочей книги функционируют точно так же, как и окна любой другой программы для Windows.

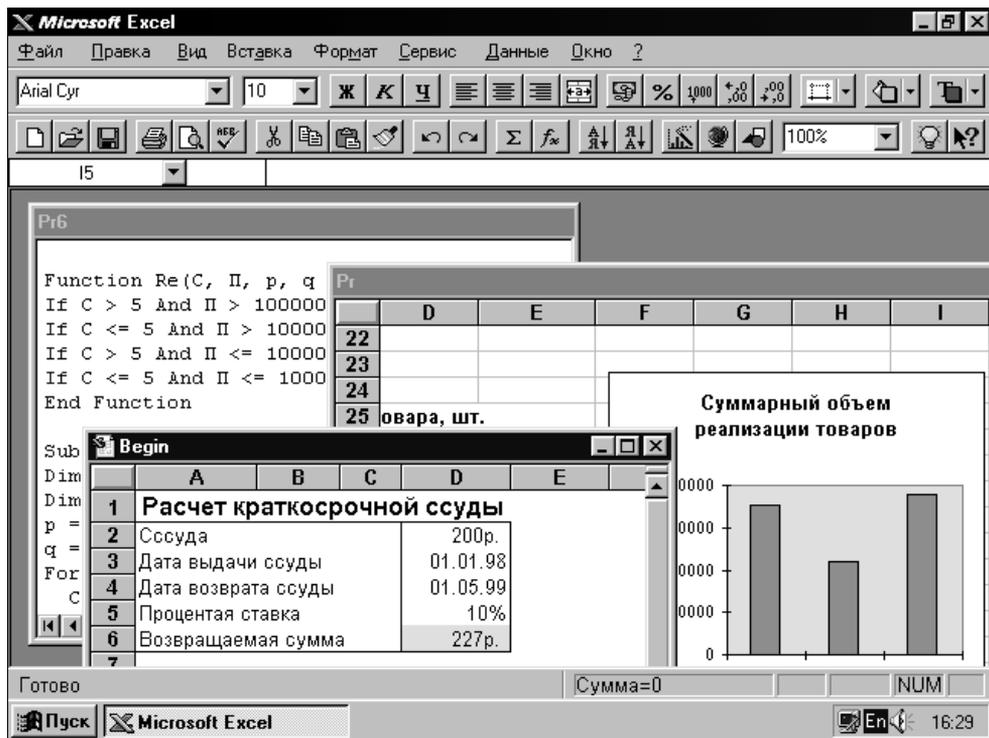


Рис. 1.1. Окно Excel с тремя открытыми рабочими книгами

Заголовок одного из окон выделен синим цветом — это активная рабочая книга, т. е. книга, с которой в данный момент вы работаете, например, вводите данные. В любой момент времени активной может быть только одна рабочая книга, и ее окно размещается поверх других открытых окон.

Для того чтобы окна всех открытых рабочих книг были видны на экране, необходимо воспользоваться командой **Окно, Расположить** (Window, Arrange). В открывшемся диалоговом окне **Расположение окон** (Arrange Windows) (рис. 1.2) предоставляется возможность выбрать вариант расположения окон: рядом, сверху вниз, слева направо или каскадом.

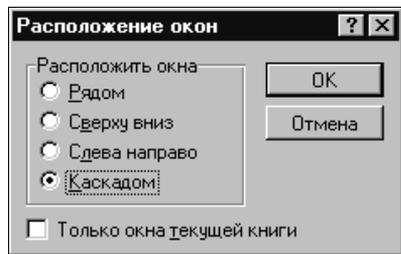


Рис. 1.2. Диалоговое окно **Расположение окон**

Команда **Окно** (Window) раскрывает меню **Окно**, в нижней части которого приводится список имен открытых в данный момент рабочих книг (рис. 1.3).

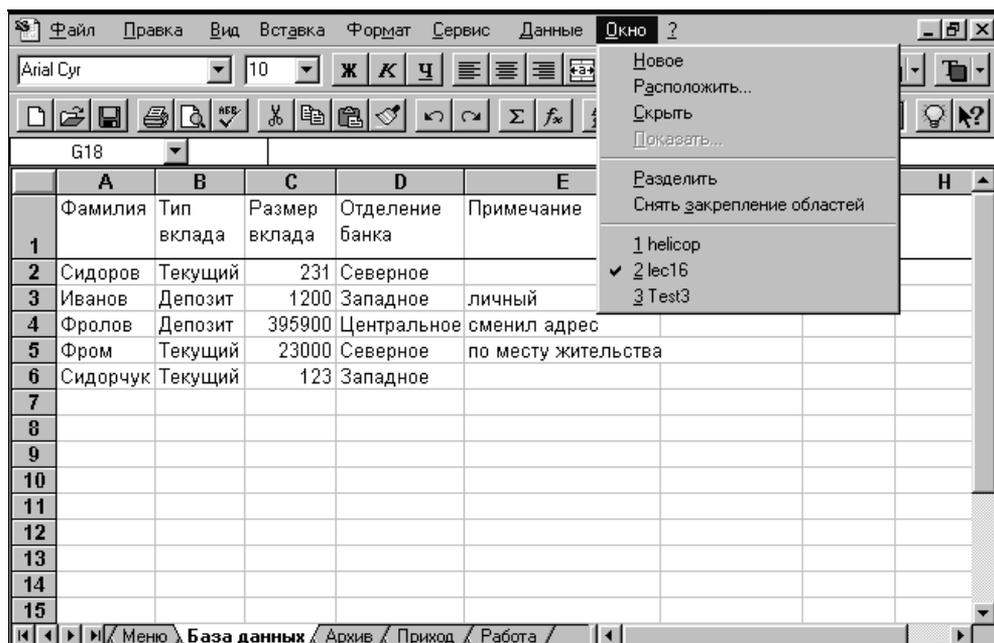


Рис. 1.3. Меню **Окно**

Для того чтобы активизировать окно нужной рабочей книги, необходимо щелчком мыши выбрать ее имя в этом списке (слева от имени появится галочка). Если нужное окно (хотя бы небольшая его часть) видно на экране, то его можно активизировать просто щелчком мыши по нему.

Окна открытых рабочих книг циклически активизируются нажатием клавиш <Ctrl>+<Tab> или <Ctrl>+<F6>. Нажатие клавиш <Ctrl>+<Shift>+<F6> производит активизацию этих окон в обратном порядке. Команда **Файл, Закрыть** (File, Close) или нажатие клавиш <Ctrl>+<F4> закрывает рабочую книгу. При этом Excel проверяет, были ли произведены изменения в рабочей книге со времени последнего сохранения файла. Если нет, то Excel закрывает файл без всяких сообщений. В противном случае он предложит сохранить файл (рис. 1.4).

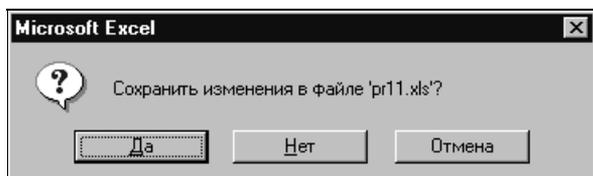


Рис. 1.4. Диалоговое окно, открывающееся при закрытии рабочей книги

1.2. Работа с файлами

Основными типами файлов, с которыми работает Excel, являются рабочая книга и файл шаблона, имеющие расширения .XLS и .XLT, соответственно.

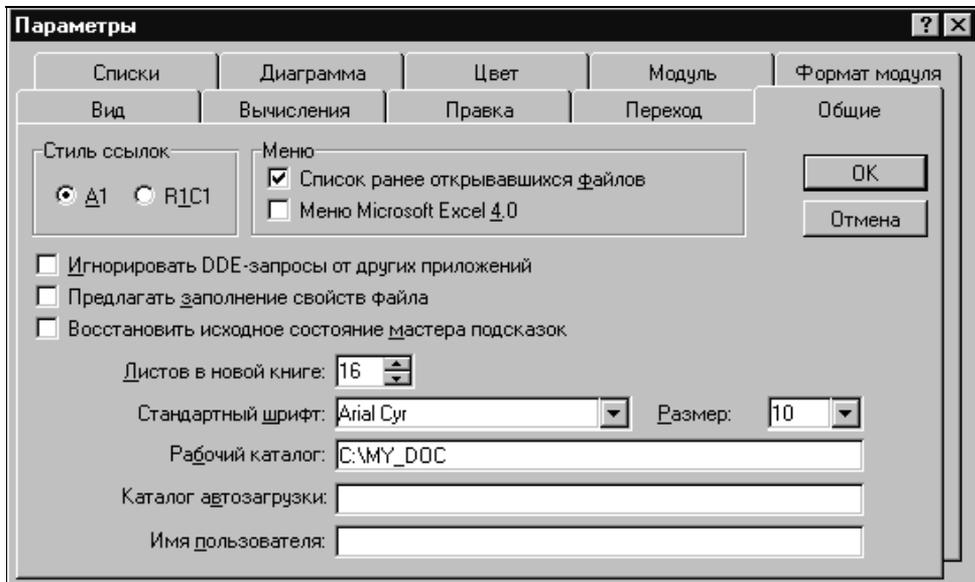
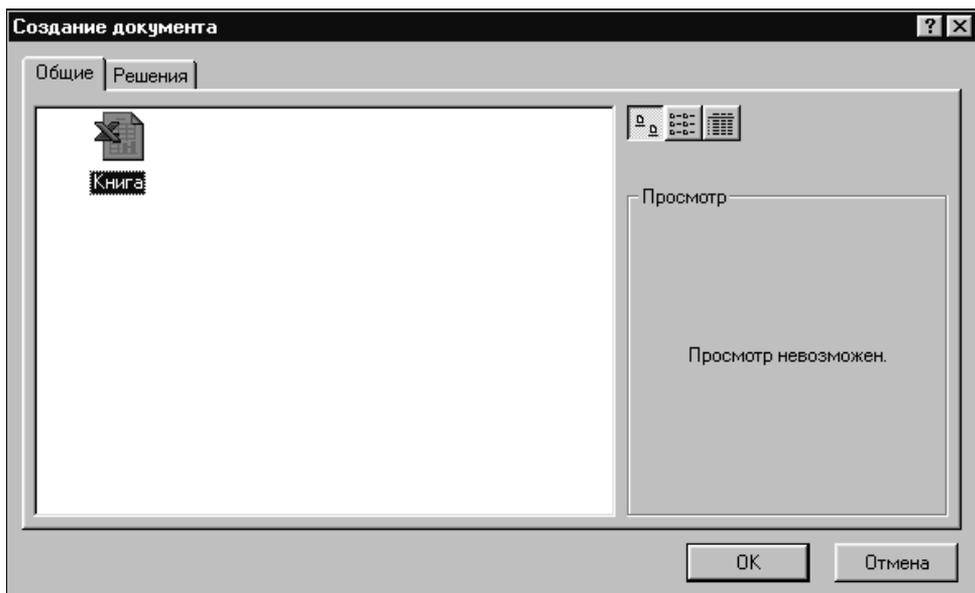
1.2.1. Создание новой рабочей книги

После запуска Excel автоматически создает новую рабочую книгу с именем Книга1. По умолчанию рабочая книга состоит из 16 рабочих листов с именами Лист1 (Sheet1), Лист2 (Sheet2), ..., Лист16 (Sheet16). Число листов задается счетчиком **Листов в новой книге** (Sheets In New Workbook) вкладки **Общие** (General) диалогового окна **Параметры** (Options), открываемого командой **Сервис, Параметры** (Tools, Options) (рис. 1.5).

Новую рабочую книгу можно также создать одним из следующих способов:

- Выбором команды **Файл, Создать** (File, New)
- Нажатием кнопки 
- Нажатием клавиш <Ctrl>+<N>

Каждое из перечисленных действий приводит к появлению диалогового окна **Создание документа** (New) (рис. 1.6).

Рис. 1.5. Вкладка **Общие** диалогового окна **Параметры**Рис. 1.6. Вкладка **Общие** диалогового окна **Создание документа**

На вкладке **Решения** (Spreadsheet Solutions) диалогового окна **Создание документа** (New) (рис. 1.7) выбирается шаблон рабочей книги: **Заказ**, **Счет** и т. д.

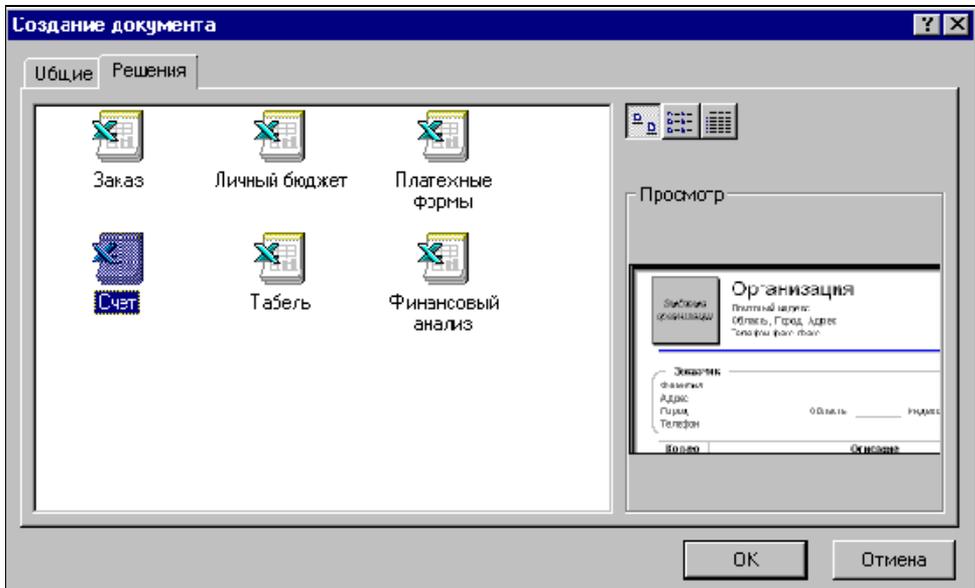


Рис. 1.7. Вкладка **Решения** диалогового окна **Создание документа**

1.2.2. Открытие существующей рабочей книги

Открыть существующую рабочую книгу можно одним из следующих способов:

- Выбором команды **Файл, Открыть** (File, Open)
- Нажатием кнопки 
- Нажатием клавиш <Ctrl>+<O>

В результате на экране появится диалоговое окно **Открытие документа** (Open) (рис. 1.8).

В поле **Папка** (Look in) указана папка, из которой считываются и в которой сохраняются по умолчанию рабочие книги. Если вы хотите изменить этот стандартный рабочий каталог, введите необходимые изменения в поле **Рабочий каталог** (Default File Location) вкладки **Общие** (General) диалогового окна **Параметры** (Options), открываемого командой **Сервис, Параметры** (Tools, Options) (рис. 1.5).

Открыть рабочую книгу можно также двойным щелчком на значке ее XLS-файла. Если вам нужно открыть файл, с которым недавно работали, то его имя вы найдете в меню **Файл** (File), где приводится список последних четырех открывавшихся файлов.

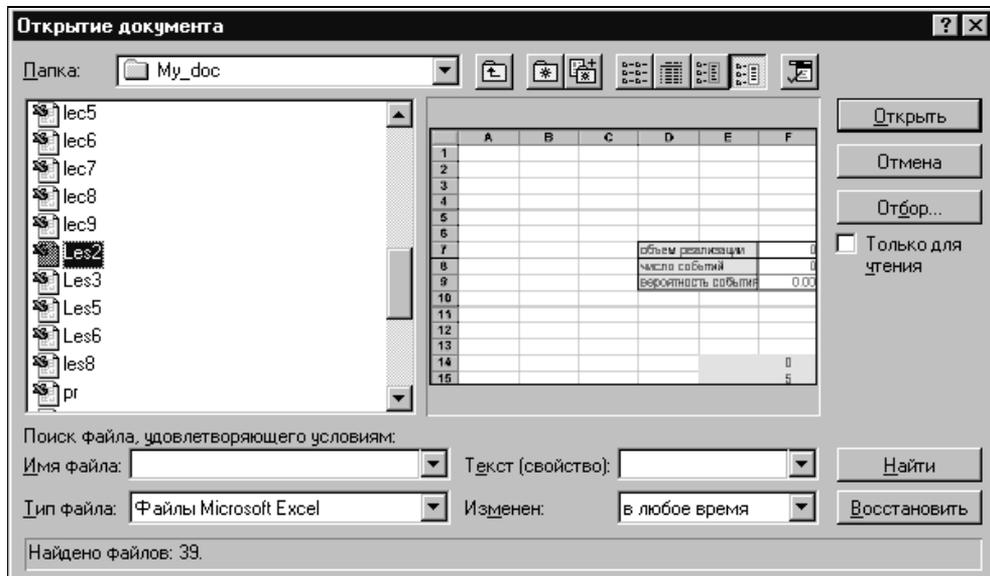


Рис. 1.8. Диалоговое окно **Открытие документа**

Диалоговое окно **Открытие документа** (Open) также позволяет производить поиск файла на диске. Для этого введите имя файла в поле **Имя файла** (File name) и нажмите кнопку **Найти** (Find).

1.2.3. Сохранение рабочей книги

Рабочая книга сохраняется одним из следующих способов:

- Выбором команды **Файл, Сохранить** (File, Save)
- Нажатием кнопки 
- Нажатием клавиш: <Ctrl>+<S> или <Shift>+<F12>

Если рабочая книга уже сохранялась ранее, то в результате выполнения одного из указанных действий она сохранится на диске под тем же именем. При первом сохранении рабочей книги на экране появится диалоговое окно **Сохранение документа** (Save As) (рис. 1.9).

В поле **Имя файла** (File name) вводится имя, под которым будет сохранена данная рабочая книга. По умолчанию Excel предлагает для рабочей книги следующие стандартные имена: Книга1 (Book1), Книга2 (Book2) и т. д., но использовать их не рекомендуется. Для облегчения работы с файлами предпочтительнее присваивать им такие имена, которые отражают их содержание. Для переименования рабочей книги необходимо воспользоваться командой **Файл,**

Сохранить как (File, Save As), при выполнении которой на экране также появится диалоговое окно **Сохранение документа** (Save As) (рис. 1.9).

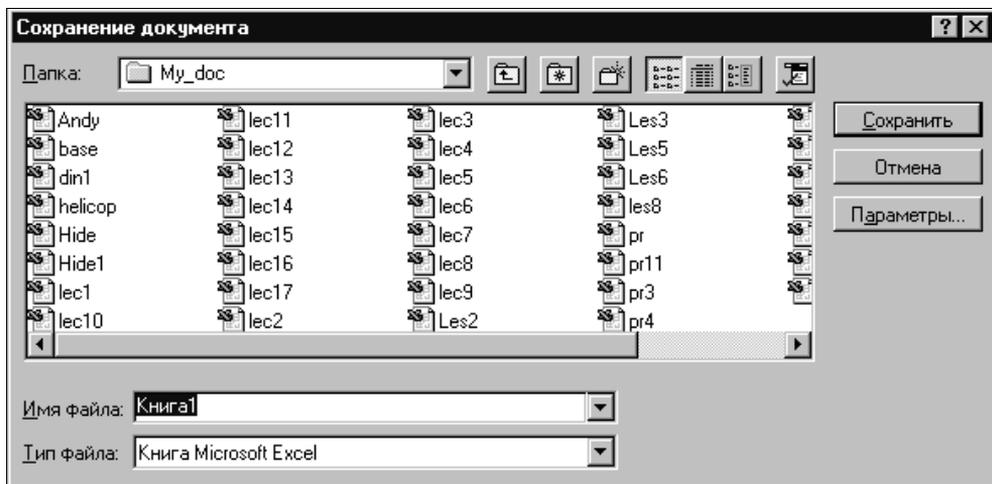


Рис. 1.9. Диалоговое окно **Сохранение документа**

1.2.4. Закрытие рабочей книги

Закрыть рабочую книгу можно:

- Выбором команды **Файл, Закрыть** (File, Close)
- Нажатием клавиш: <Ctrl>+<F4> или <Ctrl>+<W>

Если вы сделали в рабочей книге какие-либо изменения, то перед закрытием этой книги Excel выдаст запрос на ее сохранение (рис. 1.4).

1.2.5. Удаление файла рабочей книги

Если вам не нужен какой-либо файл рабочей книги и вы хотите его удалить, Excel предоставит вам и эту возможность. Для удаления файла в диалоговом окне **Открытие документа** (Open) щелкните правой кнопкой мыши на значке файла и в раскрывшемся контекстном меню выберите команду **Удалить** (Delete) (рис. 1.10).

1.3. Окно приложения Excel

На рис. 1.11 видно, что окно Excel содержит много элементов, присущих большинству программ в среде Windows (табл. 1.1).

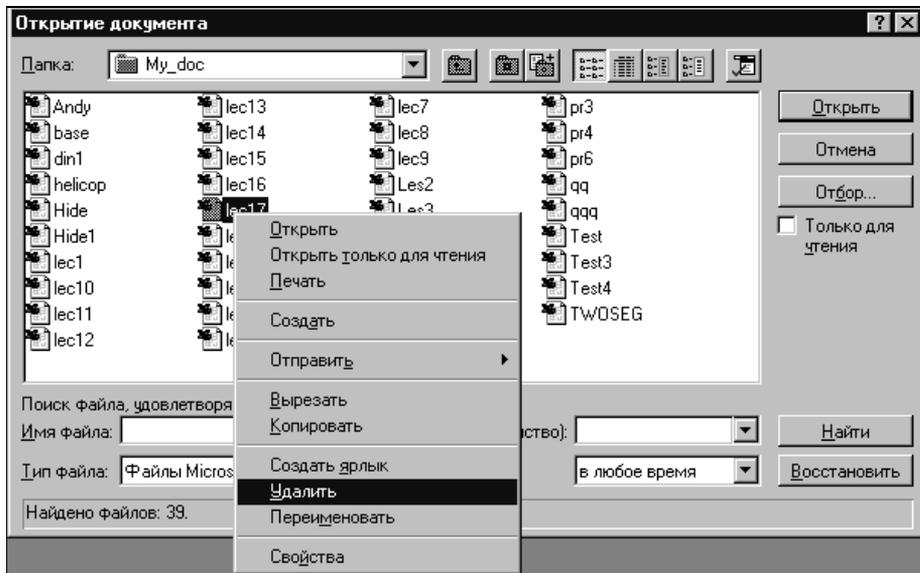


Рис. 1.10. Удаление файла в диалоговом окне **Открытие документа**

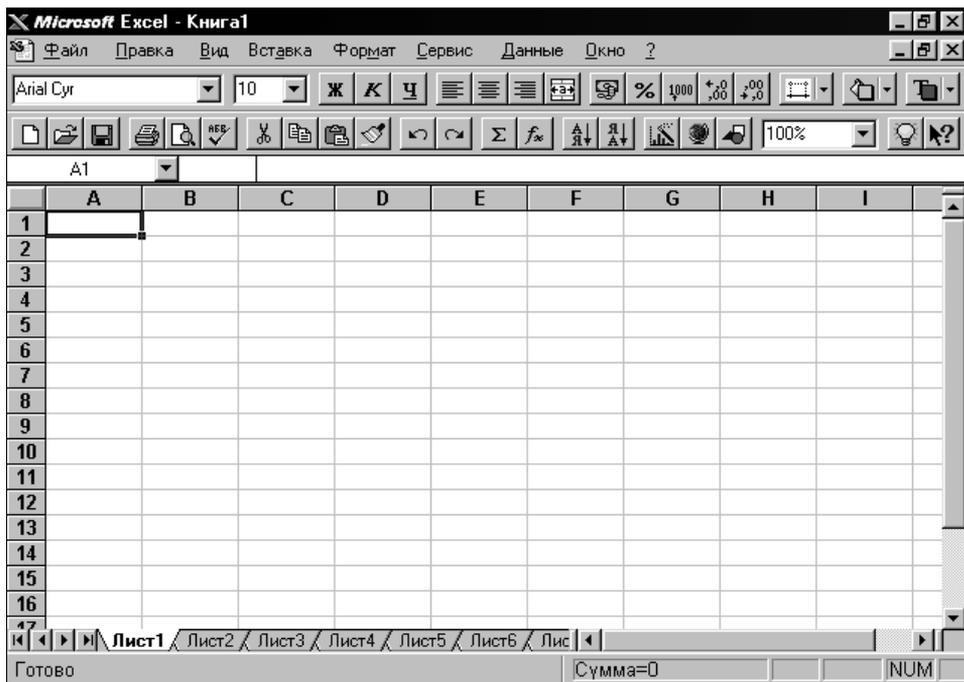


Рис. 1.11. Вид окна рабочей книги Excel

Таблица 1.1. Некоторые элементы интерфейса Windows

Элемент	Описание
Заголовок окна (Title bar)	Строка в верхней части окна приложения или рабочей книги. Заголовок окна приложения содержит название приложения и имя файла, открытого в этом окне. Заголовок окна рабочей книги содержит имя файла
Полоса меню (Menu bar)	Список команд меню. Обычно располагается под заголовком приложения
Меню (Menu)	Раскрывающийся список команд
Команда (Command)	Функция или действие, выбранное из раскрывающегося меню
Кнопка Свернуть (Minimize)	Левая кнопка из трех кнопок управления окном  в правой части заголовка. Уменьшает окно приложения или рабочей книги до кнопки на панели задач внизу экрана
Кнопка Развернуть (Maximize)	Средняя из трех кнопок управления окном в правой части заголовка. Разворачивает окно приложения или рабочей книги на все свободное пространство экрана
Кнопка Закреть (Close)	Правая из трех кнопок управления окном в правой части заголовка. Закрывает приложение Excel или рабочую книгу
Панель инструментов (Toolbar)	Панель с кнопками, нажатие которых обеспечивает быстрое выполнение определенных действий
Полоса прокрутки (Scroll bar)	Горизонтальная и вертикальная полосы, позволяющие с помощью мыши прокручивать экран по горизонтали и вертикали
Строка состояния (Status bar)	Строка в нижней части окна приложения, в которой появляются выделенные команды и даются указания и подсказки относительно следующего действия

Но разумеется, окно приложения Excel содержит и элементы, присущие именно этому приложению. Некоторые из них приведены в табл. 1.2.

Таблица 1.2. Некоторые элементы интерфейса Excel

Элемент	Описание
Ярлычки листов (Sheet Tabs)	Ярлычки с именами рабочих листов слева от полосы горизонтальной прокрутки
Кнопки прокрутки ярлычков (Tab Scrolling Buttons)	Кнопки слева от ярлычков рабочих листов  , помогающие быстро перебирать листы рабочей книги

Таблица 1.2 (окончание)

Элемент	Описание
Вешка ярлычков (Tab split box)	Небольшой прямоугольник между ярлычками листов и полосой горизонтальной прокрутки, перемещая который, можно увеличить место для ярлычков или увеличить размер полосы горизонтальной прокрутки
Строка формул (Formula bar)	Строка, в которой вводится текст, числа или формулы. Расположена ниже полосы меню и панели инструментов
Панель инструментов Стандартная (Standard)	
Панель инструментов Форматирование (Formatting)	

Как правило, в окне приложения Excel под полосой меню располагаются панели инструментов **Стандартная** (Standard) и **Форматирование** (Formatting). Панель инструментов **Рисование** (Drawing) может быть вызвана нажатием кнопки  панели инструментов **Стандартная** (Standard).

Вид окна Excel настраивается:

- ❑ В диалоговом окне **Панели инструментов** (Toolbars) (рис. 1.12), которое вызывается командой **Вид, Панели инструментов** (View, Toolbars). В этом окне имеется возможность управлять выводом панелей инструментов на экран. Флажки в нижней части окна задают размер и цвет кнопок. Флажок **Всплывающие подсказки** позволяет управлять появлением всплывающей подсказки с названием кнопки при установке на ней указателя мыши.
- ❑ На вкладке **Вид** (View) диалогового окна **Параметры** (Options), которое открывается командой **Сервис, Параметры** (Tools, Options) (рис. 1.13). В этом окне можно задать отображение на экране таких элементов окна, как сетка, строка формул, строка состояния, заголовки строк и столбцов, полосы прокрутки, ярлычки листов и т. д.

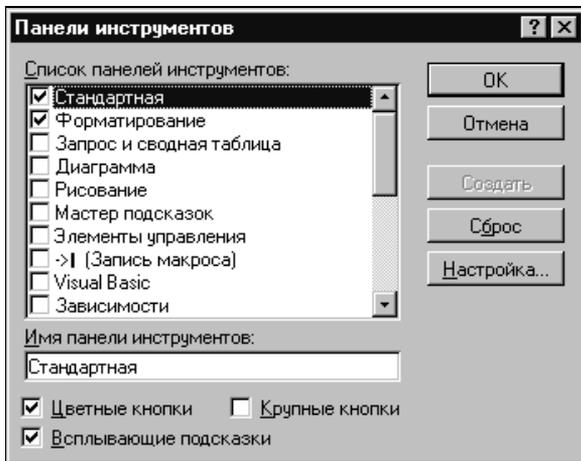


Рис. 1.12.
Диалоговое окно
Панели инструментов

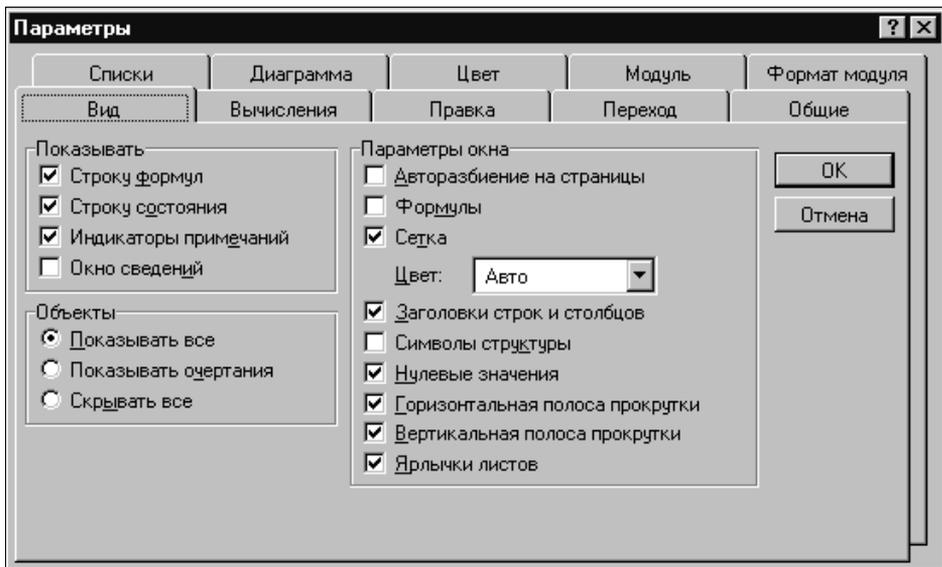


Рис. 1.13. Вкладка **Вид** диалогового окна **Параметры**

Основную часть окна Excel занимает рабочий лист, разделенный на отдельные ячейки. По умолчанию столбцы озаглавлены буквами, а строки — цифрами, однако, возможно обозначение строк и столбцов цифрами. Переход от одной системы ссылок на ячейку к другой осуществляется на вкладке **Общие** (General) диалогового окна **Параметры** (Options), которое открывается командой **Сервис, Параметры** (Tools, Options). В группе **Стиль ссылок** выбирается положение переключателя **A1** или **R1C1** (рис. 1.5). На одном рабочем листе имеется 256 столбцов и 16 386 строк.

1.4. Операции с рабочими листами

Последовательный переход от одного рабочего листа к другому осуществляется при помощи кнопок прокрутки ярлычков (табл. 1.2). Существует более быстрый способ перехода: щелкните правой кнопкой мыши на кнопке прокрутки ярлычков и в открывшемся контекстном меню из списка имен имеющихся рабочих листов выберите нужный лист (рис. 1.14).

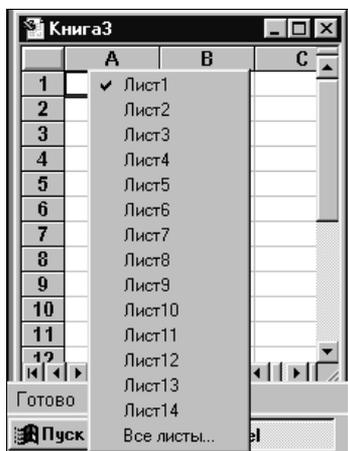


Рис. 1.14. Переход от листа к листу

Рабочие листы можно переименовывать, копировать, перемещать, удалять и скрывать.

Для переименования листа, например с именем *Лист1*, щелкните правой кнопкой мыши на ярлычке листа и в раскрывшемся контекстном меню выберите команду **Переименовать** (Rename) (рис. 1.15).

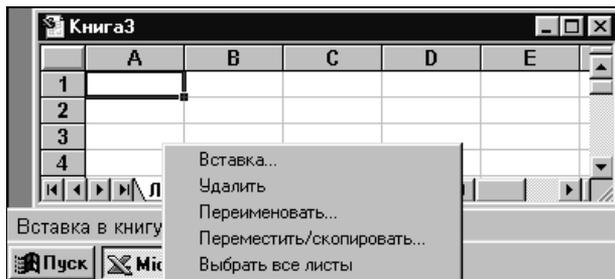


Рис. 1.15. Контекстное меню ярлычка

В открывшемся диалоговом окне **Переименование листа** (Rename Sheet) в поле **Имя листа** введите новое имя листа, например *Отчет*, и нажмите кнопку **ОК**. В результате произойдет переименование листа (рис. 1.16). Переименовать выделенный лист можно также командой **Формат, Лист, Переименовать** (Format, Sheet, Rename).

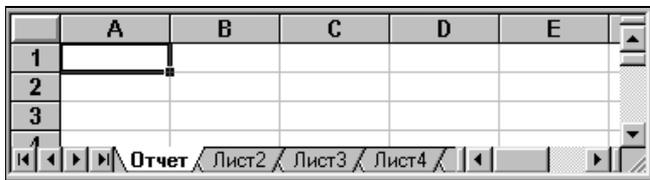


Рис. 1.16. Рабочий лист с новым именем

При необходимости удалить рабочий лист щелкните правой кнопкой мыши на ярлычке листа и в раскрывшемся контекстном меню выберите команду **Удалить** (Delete). Перед удалением Excel предупреждает, что восстановить удаленный лист невозможно. Удалить лист можно и командой **Правка, Удалить лист** (Edit, Delete Sheet).

Вставка нового листа осуществляется щелчком правой кнопки мыши на ярлычке листа, перед которым необходимо вставить новый лист. Выбор команды **Вставка** (Insert) в раскрывшемся контекстном меню открывает диалоговое окно **Вставка** (Insert), в котором следует выбрать тип вставляемого листа (рис. 1.17). Новый рабочий лист можно вставить и с помощью команды **Вставка, Лист** (Insert, Worksheet).

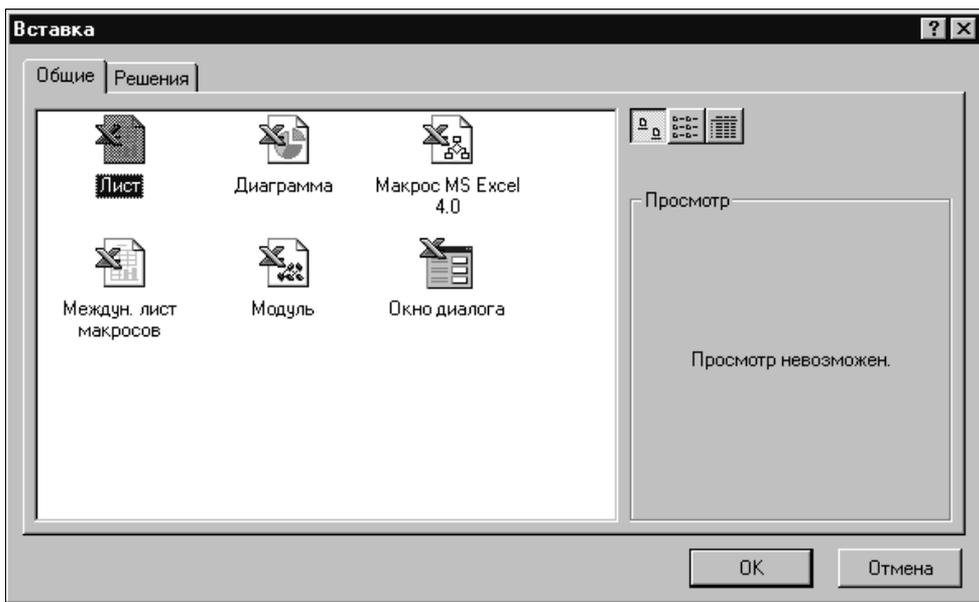


Рис. 1.17. Вкладка **Общие** диалогового окна **Вставка**

Для перемещения или копирования листа щелкните правой кнопкой мыши на ярлычке листа и в раскрывшемся контекстном меню выберите команду **Переместить/скопировать** (Move or Copy). В диалоговом окне **Переместить или скопировать** (Move or Copy Sheet) (рис. 1.18) укажите, в какую рабочую

книгу следует переместить выбранный лист и перед каким листом он должен располагаться. В случае копирования листа необходимо установить флажок **Создавать копию** (Create a Copy).

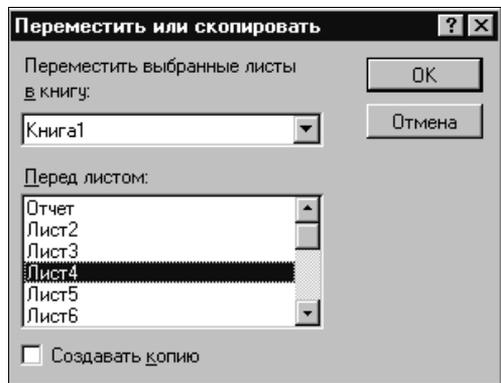


Рис. 1.18. Диалоговое окно **Переместить или скопировать**

Копировать или перемещать выделенный лист можно и с помощью команды **Правка, Переместить/скопировать** (Edit, Move or Copy).

Операцию перемещения листа удобно производить мышью. Для этого необходимо установить указатель мыши на ярлычке листа и, нажав левую кнопку мыши, перетащить ярлычок в нужное место. Подобным образом можно буксировать не только ярлычки листов, но и выделенные ячейки, их группы и любые графические объекты. Если буксировка производится при нажатой клавише <Ctrl>, то объект копируется.

Команда **Формат, Лист, Скрыть** (Format, Sheet, Hide) позволяет скрыть выделенный лист, а команда **Формат, Лист, Показать** (Format, Sheet, Unhide) — отобразить скрытый лист.

Перетаскивая **Вешку ярлычков** (Tab split box), можно изменить число ярлычков, отображаемых на экране (рис. 1.11).

Excel предоставляет возможность работать с группой листов как с одним в так называемом **Групповом режиме** (Group). Выделение группы листов осуществляется одним из следующих способов:

- Щелкните правой кнопкой мыши на ярлычке листа и в раскрывшемся контекстном меню выберите команду **Выбрать все листы** (Select all sheets)
- Добавить отдельный лист в группу можно, щелкнув на его ярлычке при нажатой клавише <Ctrl>
- Выделить в группу несколько подряд идущих листов можно, щелкнув первый и последний ярлычки группы, причем щелчок на последнем ярлычке необходимо выполнять при нажатой клавише <Shift>

Ярлычки листов, выделенных в группу, окрашены белым цветом (рис. 1.19). При желании выделенную группу листов можно удалять, перемещать, копировать, вводить в нее информацию как в один лист.



Рис. 1.19. Выделение ярлычков группы листов

Раскрывающийся список на панели **Стандартная** (Standard) позволяет изменять масштаб отображения рабочего листа. Для более точного масштабирования воспользуйтесь командой **Вид, Масштаб** (View, Zoom). В открывшемся диалоговом окне **Масштаб** (Zoom) установите необходимый масштаб от 10 до 400% (рис. 1.20).

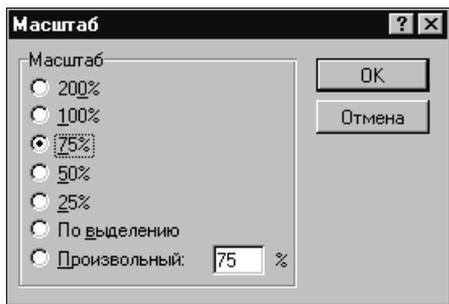


Рис. 1.20. Диалоговое окно **Масштаб**

1.5. Пример составления документа

Предположим, что ваш руководитель поручил вам рассчитать краткосрочную ссуду. Вы энергичны, полны амбиций и надеетесь получить прибавку к жалованию или повышение по службе. Несомненно, вы захотите подготовить документ, не только правильно рассчитывающий ссуду, но и весьма презентабельный, чтобы ваш руководитель с одного взгляда на него смог оценить вашу незаменимость и перспективность.

В данном разделе на примере создания таблицы расчета краткосрочной ссуды (рис. 1.21) рассмотрены основные понятия и приемы ввода, редактирования, форматирования и простейшей обработки данных, а также придания отчету презентабельного вида.



	A	B	C	D	E
1	Расчет краткосрочной ссуды				
2	Ссуда			200р.	
3	Дата выдачи ссуды			01.01.98	
4	Дата возврата ссуды			01.05.99	
5	Процентная ставка			10%	
6	Возвращаемая сумма			227р.	
7					
8					
9					
10					
11					
12					
13					
14					
15					

Рис. 1.21.
Расчет краткосрочной ссуды

Обратите внимание на удобное средство при работе в Excel: щелчок на кнопке  панели инструментов **Стандартная** (Standard) отменяет результат выполнения последней команды, а повторное выполнение последней команды осуществляется при нажатии на кнопку  той же панели. Эти кнопки позволяют отменить результаты некорректного редактирования данных и избежать повторного ввода с клавиатуры случайно удаленных данных.

1.5.1. Адресация ячейки

На активном рабочем листе одна ячейка является активной (active) или, другими словами, выделенной. Эта ячейка обрамлена черной рамкой. Перемещение черной рамки по рабочему листу осуществляется мышью или клавишами <↑>, <↓>, <←> и <→>. Каждая ячейка на активном рабочем листе определяется своим адресом, состоящим из имени столбца и номера строки, например A1. Ячейка на неактивном рабочем листе идентифицируется именем листа и ее адресом на листе, например Лист2!A1 (обратите внимание на разделительный восклицательный знак).

Существует другой способ адресации ячейки — по имени. Имя или адрес активной ячейки выводится в поле имен, которое расположено у левого края строки формул (рис. 1.22).

Для присвоения имени активной ячейке выберите команду **Вставка, Имя, Присвоить** (Insert, Name, Define) и в поле ввода **Имя** (Name) открывшегося диалогового окна **Присвоить имя** (Define Name) введите новое имя ячейки, например Первая_ячейка (рис. 1.23).

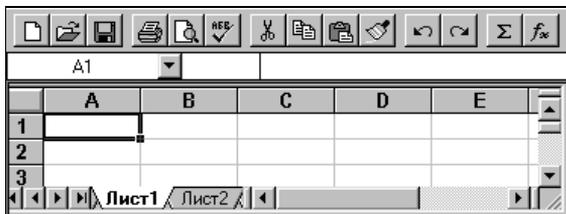
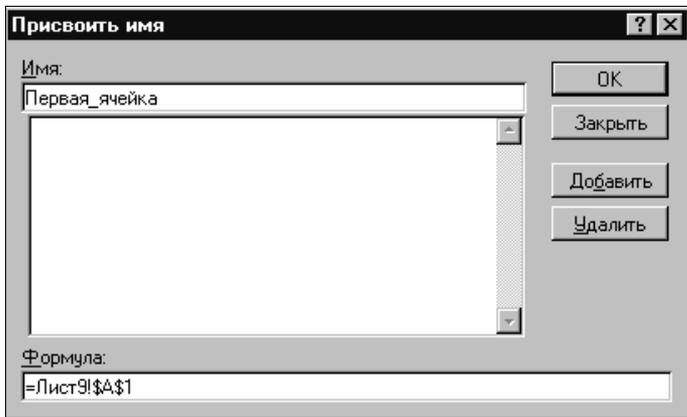


Рис. 1.22. Активная ячейка

Рис. 1.23.
Диалоговое окно
Присвоить имя

В именах ячеек не допускаются пробелы, поэтому в качестве разделителя между словами обычно используют символ подчеркивания "_". При нажатии кнопки **ОК** ячейке присваивается указанное имя. Конечно, активной ячейке можно присвоить имя и непосредственно, введя его в поле имени. Адресация ячейки по имени абсолютна, поэтому при ссылке на ячейку по имени на неактивном рабочем листе нет необходимости указывать имя этого листа.

1.5.2. Ввод данных в ячейку и их редактирование

Вернемся к нашему примеру — созданию таблицы расчета краткосрочной ссуды, приведенной на рис. 1.21. В новом созданном документе переименуем рабочий лист Лист1 в Начало и удалим остальные рабочие листы. Выделим ячейку A1 и с клавиатуры введем в нее текст Расчет краткосрочной ссуды. При вводе содержимое активной ячейки дублируется в строке формул (рис. 1.24).

Ввод текста заканчивается нажатием клавиши <Enter> или кнопки . Нажатие клавиши <Esc> или кнопки отменяет ввод или результат редактирования информации в ячейке. Хотя визуально текст, введенный в ячейку A1, занимает также ячейки B1 и C1, однако, эти ячейки пусты. В этом можно убедиться, сделав их активными и увидев, что в строку формул ничего не

выводится. Аналогично, в ячейки A2, A3, A4, A5 и A6 вводим текст, как показано на рис. 1.25.

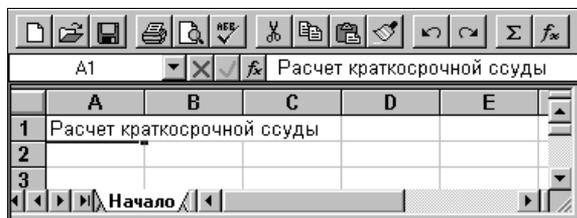


Рис. 1.24.
Ввод текста в ячейку

	A	B	C	D	E
1	Расчет краткосрочной ссуды				
2	Ссуда			200	
3	Дата выдачи ссуды			01.01.98	
4	Дата возврата ссуды			01.05.99	
5	Процентная ставка			0.1	
6	Возвращаемая сумма				

Рис. 1.25. Исходные данные для расчета краткосрочной ссуды

Для редактирования содержимого ячейки необходимо ее выделить и нажать клавишу <F2> или дважды щелкнуть ячейку. После этого содержимое ячейки редактируется стандартными средствами Windows непосредственно в ячейке или в строке формул, причем работать со строкой формул, как правило, удобнее. Для ускорения ввода информации и избежания ошибок при наборе пользуйтесь копированием через буфер обмена. В нашем примере содержимое ячеек A3 (Дата выдачи ссуды) и A4 (Дата возврата ссуды) очень похожи друг на друга. Введя текст в ячейку A3, выделите его, скопируйте в ячейку A4 и отредактируйте. Выделять текст в ячейке или строке формул можно стандартными средствами Windows:

- Перемещением указателя мыши при нажатой кнопке
- С помощью клавиш <↑>, <↓>, <←>, <→> при нажатой клавише <Shift>
- Двумя последовательными щелчками: одним в начале выделяемой области, а другим — в конце, но при нажатой клавише <Shift>

Выделенный фрагмент копируется в буфер обмена:

- Выбором команды **Правка, Копировать** (Edit, Copy)
- Нажатием клавиш <Ctrl>+<Insert> или <Ctrl>+<C>
- Нажатием кнопки 

Выделенный фрагмент копируется в буфер обмена с удалением:

- Выбором команды **Правка, Вырезать** (Edit, Cut)

- Нажатием клавиш <Shift>+<Delete> или <Ctrl>+<X>
- Нажатием кнопки 

Содержимое буфера обмена вставляется:

- Командой **Правка, Вставить** (Edit, Paste)
- Нажатием клавиш <Shift>+<Insert> или <Ctrl>+<V>
- Нажатием кнопки 

Использование буфера обмена резко ускоряет создание документа, однако, будьте осторожны: копируя через буфер обмена ошибки или опечатки, вы непроизвольно размножаете их по документу.

Продолжим работу с рассматриваемым примером составления отчетной таблицы расчета ссуды. В ячейки D2, D3, D4 и D5 (рис. 1.25) введем числа 200, 01.01.98, 01.05.99 и 0.1, соответственно (размер ссуды, даты выдачи и возврата ссуды и процентную ставку). Итак, ввод начальных данных в таблицу завершен. В следующем разделе объясняется, как по этим данным выполнить расчет ссуды.

1.5.3. Ввод формулы в ячейку

Для вычисления возвращаемой суммы необходимо ввести в ячейку D6 расчетную формулу. Формула в ячейке начинается со знака равенства "=" и может содержать операторы (табл. 1.3), пары круглых скобок, числа, адреса или имена ячеек, а также функции рабочего листа, вводимые в формулу командой **Вставка, Функция** (Insert, Function) или нажатием кнопки **Мастер функций** (Function Wizard) .

Таблица 1.3. Допустимые операторы

Оператор	Название
+	Сложение
-	Вычитание
*	Умножение
/	Деление
^	Возведение в степень
&	Конкатенация
=	Логическое сравнение
>	Логическое сравнение на больше
<	Логическое сравнение на меньше

Приведем несколько примеров формул:

=3*2	Умножение двух чисел
=A1+A2	Сложение содержимого двух ячеек
=A2^(2/3)	Возведение содержимого ячейки в степень 2/3

По умолчанию Excel создает в формулах относительные ссылки на адреса ячеек. Это означает, что Excel при копировании изменяет ссылки на ячейки в соответствии с новым положением формулы. Рассмотрим следующий пример: пусть в ячейки A1, B1, A2 и B2 введены числа 1, 2, 3 и 5, соответственно, а в ячейку C1 введена формула

=A1+B1

Скопируем эту формулу в ячейку C2 одним из двух способов:

1. Выделяем ячейку C1 и копируем в буфер обмена ее содержимое:

- При помощи команды **Правка, Копировать** (Edit, Copy)
- Нажатием клавиш <Ctrl>+<Insert> или <Ctrl>+<C>
- Нажатием кнопки 

Выделяем ячейку C2 и копируем в нее содержимое буфера обмена:

- При помощи команды **Правка, Вставить** (Edit, Paste)
- Нажатием клавиш <Shift>+<Insert> или <Ctrl>+<V>
- Нажатием кнопки 

2. Выделяем ячейку C1, располагаем указатель мыши на маркере заполнения выделенной ячейки и протаскиваем его вниз так, чтобы заполнить ячейку C2, после чего отпускаем мышь. (*Маркер заполнения* — это черный квадрат в нижнем правом углу выделенной ячейки или диапазона ячеек. При расположении указателя мыши на маркере заполнения он принимает вид черного креста.)

При перемещении формул все относительные ссылки перестраиваются так, чтобы в новом положении сохранились прежние связи. Это очень удобно при составлении различного рода таблиц.

Отметим, что второй способ копирования формулы значительно проще и элегантнее. В обоих случаях в ячейку C2 будет введена формула

=A2+B2

Для того чтобы скопировать формулу или ее часть без изменения адресации, необходимо сначала выполнить двойной щелчок на ячейке с формулой, а затем выделить в ячейке или в строке формул копируемую часть формулы.

Теперь нужно скопировать выделенный фрагмент в буфер обмена, нажав клавиши <Ctrl>+<Insert>. Затем выполните двойной щелчок на ячейке, в которую должна быть скопирована формула, и вставьте в нее содержимое буфера обмена, нажав клавиши <Shift>+<Insert>.

Временно прервем пополнение нашего теоретического багажа и вернемся к нашему примеру. Для вычисления возвращаемой суммы денег по ссуде введем в ячейку D6 формулу

$$=D2 * (1+D5) ^ ((D4-D3) / 365)$$

После нажатия клавиши <Enter> в этой ячейке будет вычислена искомая сумма (рис. 1.26). Итак, расчет краткосрочной ссуды завершен, однако, внешний вид отчета еще далек от совершенства. Как придать нашему документу презентабельный вид, будет изложено в следующих разделах.

	A	B	C	D	E
1	Расчет краткосрочной ссуды				
2	Ссуда			200	
3	Дата выдачи ссуды			01.01.98	
4	Дата возврата ссуды			01.05.99	
5	Процентная ставка			0.1	
6	Возвращаемая сумма			227.0028	

Рис. 1.26.
Результат расчета
краткосрочной ссуды

1.5.4. Форматы данных

Числовые значения, которые вводятся или вычисляются, как правило, не отформатированы (это просто последовательности цифр). Для более наглядного представления результатов вычисления их желательно отформатировать. Форматировать данные в выделенной ячейке можно одним из следующих способов:

- С помощью вкладки **Число** (Number) диалогового окна **Формат ячеек** (Format Cells), которое открывается командой **Формат, Ячейки** (Format, Cells) (рис. 1.27)
- Щелчком правой кнопки мыши на ячейке вызвать контекстное меню и выбрать команду **Формат ячеек** (Format Cells) (рис. 1.28). В результате откроется диалоговое окно **Формат ячеек** (Format Cells) (рис. 1.27)
- Нажатие клавиш <Ctrl>+<1> также открывает диалоговое окно **Формат ячеек** (Format Cells)

Рассмотрим основные типы числовых форматов:

- Общий** (General) — этот формат принят по умолчанию.

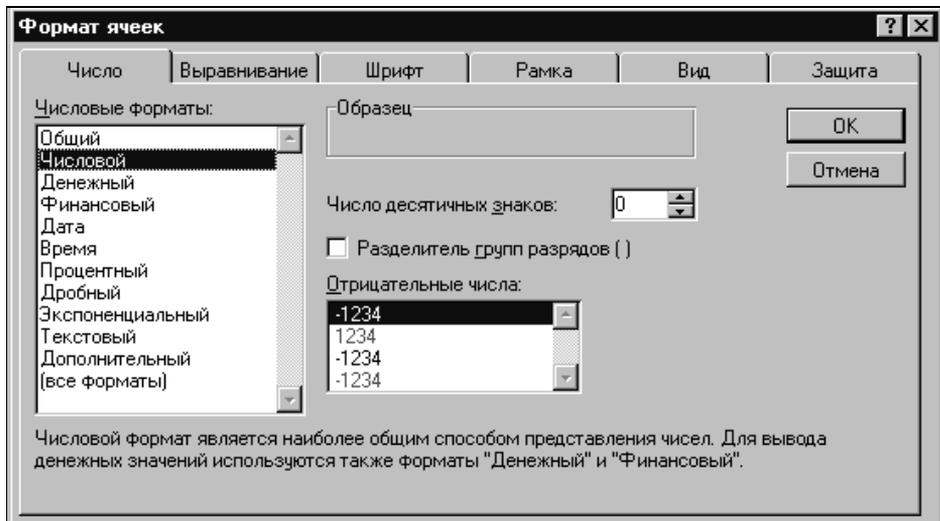


Рис. 1.27. Вкладка **Число** диалогового окна **Формат ячеек**

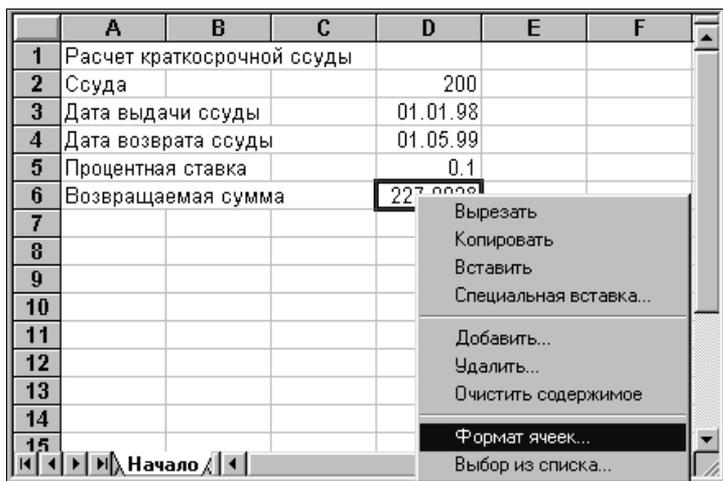


Рис. 1.28. Выбор команды **Формат ячеек** из контекстного меню

- **Числовой** (Number) — этот формат позволяет определить число выводимых знаков после десятичной точки, необходимость использования разделителя групп разрядов и способ вывода отрицательных чисел. Для того чтобы увеличить на единицу число разрядов после десятичной точки, нажмите кнопку , а чтобы уменьшить — кнопку . Вставка разделителя групп разрядов и отображение числового значения с двумя знаками после запятой производится нажатием кнопки .

- ❑ **Денежный (Currency)** — этот формат позволяет определить число выводимых знаков после десятичной точки, необходимость использования знака денежной единицы и способ вывода отрицательных чисел. Для того чтобы добавить знак денежной единицы, разделитель тысяч и отобразить число с точностью до двух знаков после запятой, просто нажмите кнопку .
- ❑ **Финансовый (Accounting)** — этот формат отличается от **денежного** тем, что числа выравниваются по десятичной точке.
- ❑ **Дата (Date)** — эта категория позволяет вывести дату в одном из 12 форматов (табл. 1.4). При работе с датами нужно иметь в виду, что Excel представляет каждую дату как ее порядковый номер в ряду дат, начиная с даты 1 января 1900 года, у которой первый порядковый номер. Такая особенность дат позволяет вычитать их друг из друга, для того чтобы найти количество прошедших дней, как это было сделано в формуле, введенной в ячейку D6 (рис. 1.26).
- ❑ **Время (Time)** — эта категория позволяет вывести время в одном из 8 форматов (табл. 1.4).

Таблица 1.4. Форматы Дата и Время

Дата	Время
7.5	15:23
7.5.98	3:23 PM
07.05.98	15:23:00
7 май	3:23:00 PM
7 май 98	23:00.0
07 май 98	15:23:00
май 98	07.05.98 3:23 PM
Май 98	07.05.98 15:23
7 Май, 1998	
07.05.98 12:00 AM	
07.05.98 0:00	
Май 7, 1998	

- ❑ **Процентный (Percentage)** — этот формат позволяет вывести число, умноженное на 100, со знаком процента и определить число выводимых зна-

ков после десятичной точки. Для того чтобы вывести число в этом формате без десятичных знаков после десятичной точки, просто нажмите кнопку .

- ❑ **Дробный** (Fraction) — эта категория позволяет вывести число в одном из 9 форматов. При вводе в ячейку чисел в этом формате отделяйте целую часть от дробной пробелом.
- ❑ **Экспоненциальный** (Scientific) — в этом формате число отображается в виде aEb , где a — мантисса, b — порядок. Запись aEb используется в программировании для представления чисел с плавающей десятичной точкой вида $a \times 10^b$. Например число 1.23×10^{-2} записывается как 1.23E-2. В этом формате можно для мантиссы определить число выводимых знаков после десятичной точки.
- ❑ **Текстовый** (Text) — применение этого формата к числовому значению позволяет Excel рассматривать его как текст.
- ❑ **Дополнительный** (Special) — в эту категорию входят четыре дополнительных формата: **Почтовый индекс** (Zip Code), **Индекс +4** (Zip Code+4), **Номер телефона** (Phone Number) и **Табельный номер** (Social security Number).
- ❑ **Все форматы** (Custom) — эта категория позволяет создавать свои форматы.

Вернемся к нашему примеру — созданию отчетной таблицы расчета краткосрочной ссуды (рис. 1.21). Отформатируем данные, полученные на рис. 1.26 в ячейках от D2 до D6, так, чтобы они приняли презентабельный вид, показанный на рис. 1.21. Предварительно изучим способы улучшения внешнего вида документа.

1.5.5. Диапазон ячеек

Форматировать данные можно как в одной ячейке, так и в диапазоне ячеек или группе несмежных диапазонов ячеек. *Диапазон* (range) — это прямоугольная группа ячеек. Для его адресации необходимо указать адреса верхней левой и нижней правой ячеек диапазона, разделенные двоеточием, например A1:C3 и C7:E10 (рис. 1.29).

Выделить диапазон можно одним из следующих способов:

- ❑ Щелкнув угловую ячейку диапазона и перетащив указатель мыши на диагонально противоположную ячейку диапазона
- ❑ Последовательно щелкнув две диагонально противоположные ячейки диапазона, причем щелчок на второй ячейке нужно выполнять при нажатой клавише <Shift>

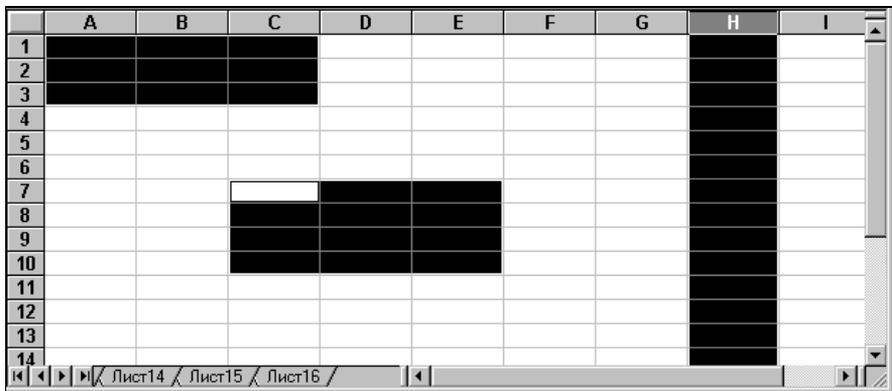


Рис. 1.29. Выделение диапазонов

- Для выделения целого столбца (строки) необходимо выполнить щелчок на его (ее) заголовке
- Нажатие клавиш <Ctrl>+<Пробел> и <Shift>+<Пробел> выделяет текущий столбец и строку, соответственно
- Весь рабочий лист выделяется нажатием клавиш <Ctrl>+<A>
- Выделение группы несмежных диапазонов производится последовательно. Для того чтобы не произошла отмена предыдущего выделения, щелкайте угловую ячейку очередного выделяемого диапазона при нажатой клавише <Ctrl>. Далее диапазон выделяется одним из вышеописанных способов. В формулах несмежные диапазоны объединяются знаком ";", например A1:C3;C7:E10.

Содержимое диапазона ячеек можно удалять, перемещать и копировать одним из стандартных способов Windows, описанных выше. Кроме того, Excel позволяет копировать содержимое ячейки или диапазона ячеек путем буксировки. Для этого выделите ячейку или диапазон, установите указатель мыши на одной из его сторон так, чтобы он превратился в стрелку, и перетяните выделенный диапазон на новое место при нажатой клавише <Ctrl>. Если клавиша <Ctrl> не нажата, диапазон перемещается на новое место без копирования.

Диапазон можно форматировать. В диапазон легко вводить повторяющиеся данные. Для этого достаточно его выделить, ввести данные и нажать клавиши <Ctrl>+<Enter>. Кроме того, как будет объяснено в последующих главах, диапазон удобно использовать в формулах.

При форматировании содержимого ячеек, кроме формата числа, можно определить способ выравнивания текста в ячейке, формат шрифта, наличие и

параметры рамки вокруг ячейки или диапазона, цвет фона ячейки. Рассмотрим подробнее способы их задания.

1.5.6. Выравнивание текста

По умолчанию текст, введенный в ячейку, выравнивается по ее левому краю, а числа — по правому краю (рис. 1.26). Изменить способ выравнивания в ячейке или диапазоне можно с помощью следующих кнопок панели инструментов **Форматирование** (Formatting):

- Выравнивание по левому краю — 
- Выравнивание по центру — 
- Выравнивание по правому краю — 

При необходимости отцентрировать текст в нескольких столбцах следует выделить нужный диапазон ячеек и нажать кнопку **Объединить и поместить в центре** (Merge and Center) .

Кроме того, вкладка **Выравнивание** (Alignment) диалогового окна **Формат ячеек** (Format Cells), открываемого командой **Формат, Ячейки** (Format, Cells), позволяет дополнительно изменять ориентацию текста в ячейке, выравнивать текст по вертикали и разрешать перенос текста по словам внутри ячейки (рис. 1.30).



Рис. 1.30. Вкладка **Выравнивание** диалогового окна **Формат ячеек**

1.5.7. Шрифты

Изменить тип и размер шрифта можно с помощью раскрывающихся списков:

Шрифт (Font) и **Размер шрифта** (Font Size) панели инструментов **Форматирование** (Formatting). Кнопки **Ж** **К** **Ч** изменяют начертание текста на **полужирное** (Bold), **курсивное** (Italic) и **подчеркнутое** (Underline), соответственно. Цвет текста выбирается с помощью кнопки .

Кроме того, вкладка **Шрифт** (Font) диалогового окна **Формат ячеек** (Format Cells), открываемого командой **Формат, Ячейки** (Format, Cells), позволяет дополнительно создавать эффект зачеркнутого шрифта, придавать тексту или числам формат нижнего или верхнего индекса (рис. 1.31).

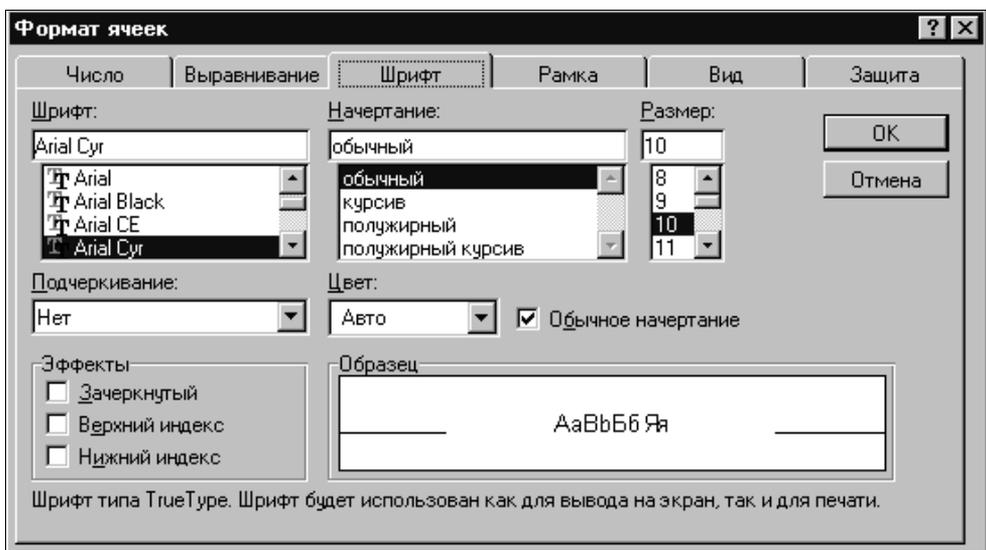


Рис. 1.31. Вкладка **Шрифт** диалогового окна **Формат ячеек**

1.5.8. Рамки и цвет фона

Рамка вокруг выделенной ячейки или диапазона ячеек создается с помощью кнопки  панели инструментов **Форматирование** (Formatting). Изменение цвета фона производится кнопкой .

Вкладка **Рамка** (Border) диалогового окна **Формат ячеек** (Format Cells), открываемого командой **Формат, Ячейки** (Format, Cells), позволяет дополнительно задавать тип и цвет линии обрамления.

1.5.9. Изменение ширины столбцов и высоты строк

Прежде всего столбцы необходимо выделить. После выделения можно воспользоваться одним из следующих способов:

- ❑ Перетаскивайте мышью правую границу заголовка любого из выделенных столбцов до тех пор, пока они не достигнут требуемых размеров
- ❑ В диалоговом окне **Ширина столбца** (Column Width), которое открывается командой **Формат, Столбец, Ширина** (Format, Column, Width), введите требуемую ширину
- ❑ Воспользуйтесь командой **Формат, Столбец, Автоподбор ширины** (Format, Column, AutoFit Selection), которая устанавливает ширину столбцов в соответствии с содержимым их ячеек (такой же результат достигается двойным щелчком на правой границе заголовка столбца)

Аналогично настраивается высота строк.

Если при расчетах в ячейке вместо числа появляется #####, это означает, что результат не помещается в ячейку и ее ширину необходимо увеличить.

Теперь вновь вернемся к нашему примеру — созданию отчетной таблицы расчета ссуды. Используя приобретенные навыки, отформатируем документ (рис. 1.26) так, чтобы он принял вид, показанный на рис. 1.21. Для этого увеличим размер шрифта заголовка таблицы, зададим для него полужирное начертание и выровняем его по центру таблицы. Применим к ячейкам таблицы обрамление, а ячейку с окончательным результатом выделим фоном. Действия, необходимые для окончательного завершения отчета, обсудим в следующем разделе.

1.5.10. Специальные текстовые и графические эффекты

Для придания отчету большей наглядности и презентабельности в него рекомендуется добавлять различные графические объекты. В качестве эмблем можно использовать рисунки, созданные в любом графическом редакторе, или уже готовые рисунки из коллекции, например **Microsoft ClipArt**, которая вызывается командой **Вставка, Объект, Картинка Microsoft ClipArt** (Insert, Object, Microsoft ClipArt Gallery 2.0) (рис. 1.32).

Приложение **Microsoft WordArt2.0** позволяет создавать такие специальные текстовые эффекты, как фигурная ориентация текста, текст с тенью и т. д. (рис. 1.33).

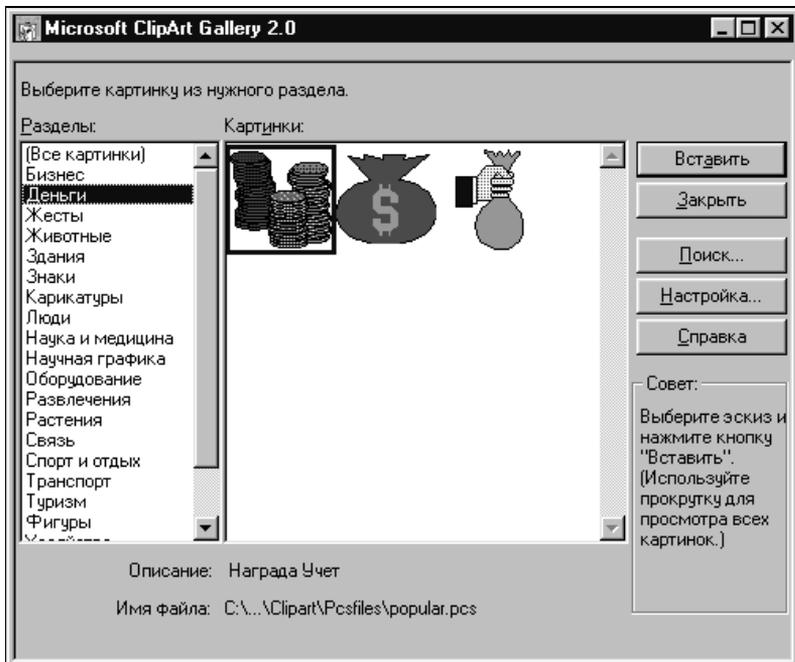


Рис. 1.32. Окно Microsoft ClipArt Gallery 2.0

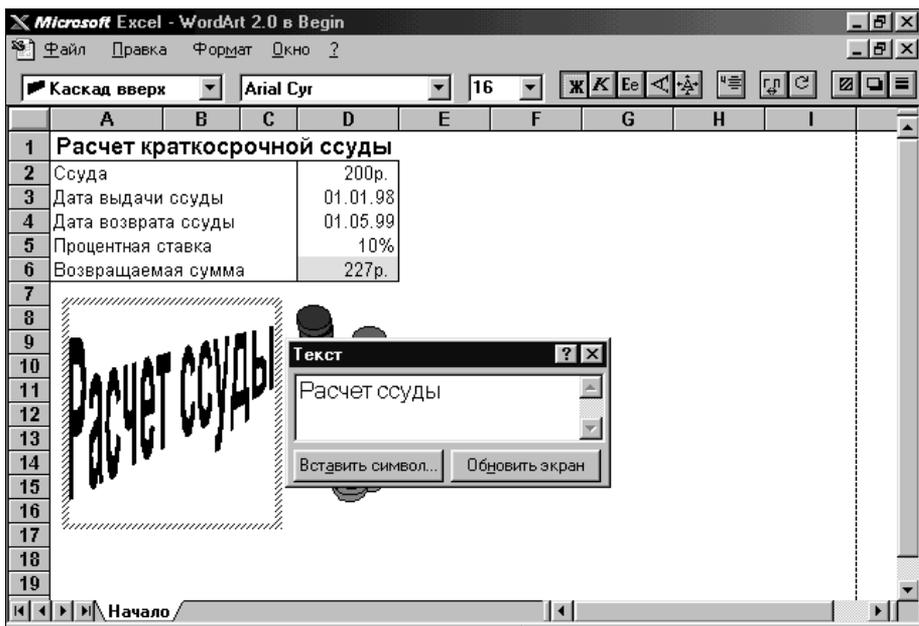


Рис. 1.33. Окно WordArt 2.0

Кроме того, Excel с помощью панели инструментов **Рисование** (Drawing) (рис. 1.34), вызываемой командой **Вид, Панели инструментов** (View, Toolbars) или нажатием кнопки  панели инструментов **Стандартная** (Standard), позволяет непосредственно на рабочем листе создавать простые рисунки.



Рис. 1.34. Панель инструментов **Рисование**

Приведем несколько простых советов по созданию графических объектов с помощью панели инструментов **Рисование** (Drawing):

- Для создания квадрата или круга рисуйте их при нажатой клавише <Shift>
- Для того чтобы расположить стрелку или отрезок под углом, кратным 45° , во время рисования объекта нажимайте клавишу <Shift>
- Для перемещения объекта строго по вертикали или горизонтали перетаскивайте его при нажатой клавише <Shift>
- Для привязки объекта к линиям сетки рисуйте его при нажатой клавише <Alt>
- Для того чтобы перемещать объект с привязкой к линиям сетки, перетаскивайте его при нажатой клавише <Alt>

1.5.11. Параметры страницы, предварительный просмотр и печать документа

В полученной таблице расчета краткосрочной ссуды осталось отключить сетку, сбросив флажок **Сетка** (Gridlines) на вкладке **Вид** (View) диалогового окна **Параметры** (Options), открываемого командой **Сервис, Параметры** (Tools, Options).

С помощью команды **Файл, Предварительный просмотр** (File, Print Preview) или нажатия кнопки  панели инструментов **Стандартная** (Standard) можно проверить, как документ будет располагаться на листе бумаги при печати (рис. 1.35). Выход из режима **Просмотр** (Print Preview) производится нажатием кнопки **Заккрыть** (Close).

Изменение параметров страницы осуществляется командой **Файл, Параметры страницы** (File, Page Setup). Диалоговое окно **Параметры страницы** (Page Setup) (рис. 1.36) позволяет задать ориентацию (книжная, альбомная), раз-

мер бумаги, качество печати, размеры полей и колонтитулов, нумерацию страниц и выводимый в колонтитулы текст. При нажатии кнопки **Печать** открывается диалоговое окно **Печать (Print)** (рис. 1.37), в котором можно задать печать не всего документа, а только выделенного диапазона ячеек.

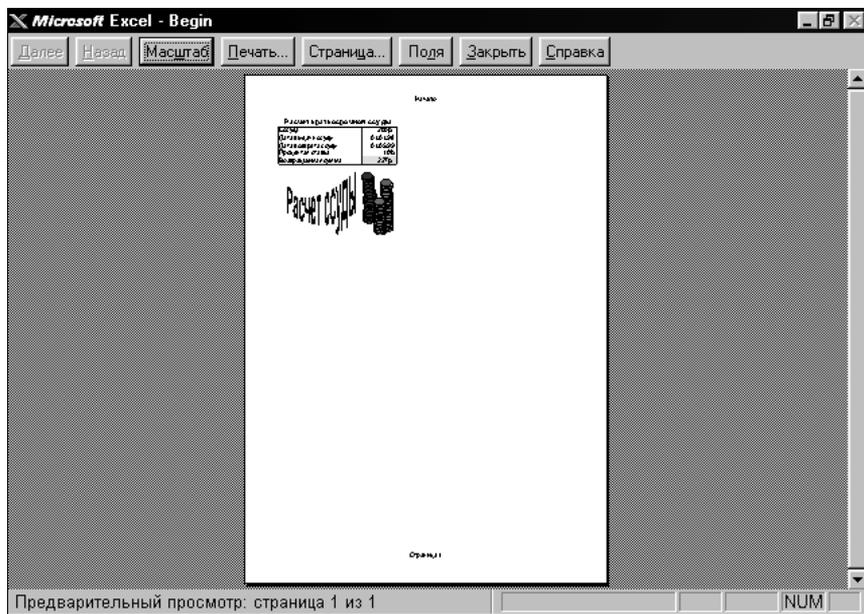


Рис. 1.35. Окно Предварительного просмотра

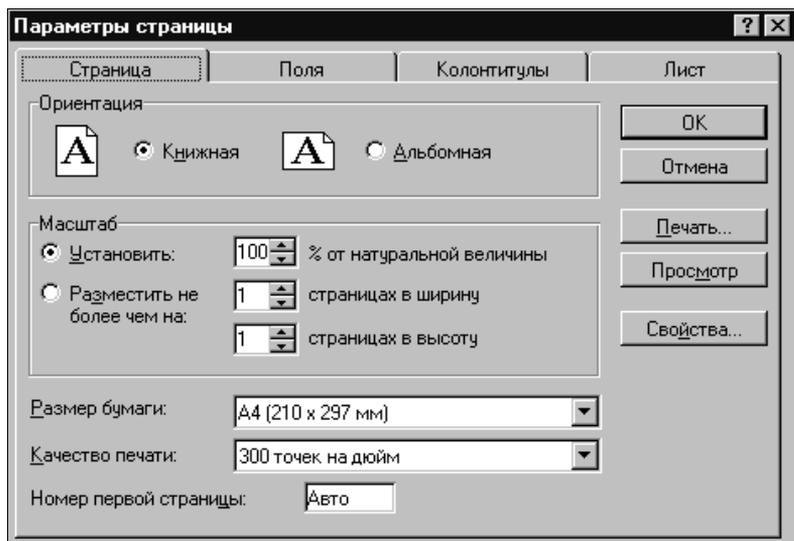


Рис. 1.36. Диалоговое окно Параметры страницы

Команда **Файл, Печать** (File, Print) и кнопка  панели инструментов **Стандартная** (Standard) управляют выводом документа на печать. С помощью диалогового окна **Печать** (Print) (рис. 1.37) можно задать принтер и выбрать вариант печати: **Всю книгу** (Entire Workbook), **Выделенные листы** (Selected Sheets) или **Выделенный диапазон** (Selection). Также можно задать диапазон страниц, выводимых на печать, и число печатаемых копий. Если нажать кнопку **Свойства** (Properties), открывается диалоговое окно, позволяющее скорректировать свойства выбранного принтера.

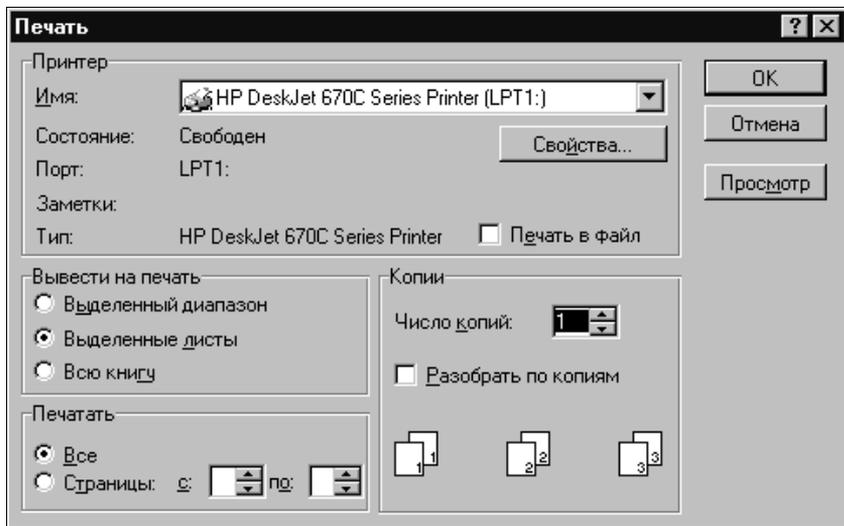
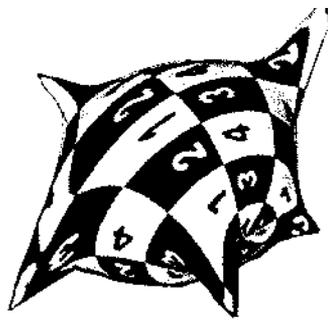


Рис. 1.37. Диалоговое окно **Печать**

Итак, ваш отчет по расчету краткосрочной ссуды создан и напечатан. Теперь с высоко поднятой головой вы можете его отнести вашему руководителю.

Глава 2



Построение графиков и решение нелинейных уравнений

В этой главе рассказывается, как с помощью электронных таблиц решаются две типичные задачи: построение графиков и решение уравнений с одним неизвестным. Для построения графиков Excel предоставляет прекрасное средство — мастер диаграмм с большим набором графиков и диаграмм, позволяющий представить данные наглядно, в наиболее выгодном свете.

Удивительно часто мы сталкиваемся с необходимостью решить то или иное уравнение, например, определить процентную ставку, при которой предлагаемая сделка выгодна, или определить скорость оборота капиталовложений. **Подбор параметра** (Goal Seek) как раз и является тем средством Excel, которое позволяет очень просто решать такие задачи.

2.1. Построение графика. Работа с мастером функций и мастером диаграмм

В качестве примера рассмотрим процедуру построения графика функции $y = \cos^2(\pi x)$ при $x \in [0, 1]$.

Для построения графика функции необходимо сначала построить таблицу ее значений при различных значениях аргумента, причем обычно аргумент изменяется с фиксированным шагом. Шаг выбирают небольшим, чтобы таблица значений функции отражала ее поведение на интервале табуляции. В нашем случае будем считать, что шаг изменения аргумента равен 0,1. Необходимо найти: $y(0)$, $y(0,1)$, $y(0,2)$, ..., $y(1)$. С этой целью в диапазон ячеек A1:A11 введем следующие значения переменной x : 0, 0,1, 0,2, ..., 1. Отме-

тим, что выбранные нами значения переменной образуют арифметическую прогрессию. Заполнение ячеек членами арифметической прогрессии в Excel можно осуществить двумя способами:

Первый способ. В ячейки A1 и A2 вводим первый и второй члены арифметической прогрессии и выделяем эти ячейки. После этого устанавливаем указатель мыши на маркере заполнения выделенного диапазона (рис. 2.1) и протаскиваем его вниз до тех пор, пока не получится числовой ряд нужной длины (рис. 2.2).

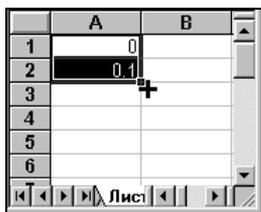


Рис. 2.1. Указатель мыши на маркере заполнения

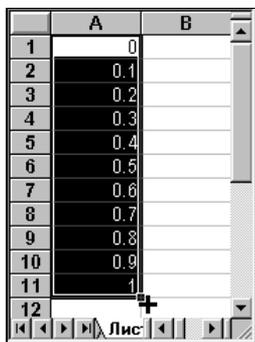


Рис. 2.2. Построение последовательности с помощью маркера заполнения

Второй способ. В ячейку A1 вводим первый член арифметической прогрессии. Выбираем команду **Правка, Заполнить, Прогрессия** (Edit, Fill, Series) и в открывшемся диалоговом окне **Прогрессия** (Series) (рис. 2.3) в группе **Расположение** (Arrange) устанавливаем переключатель в положение **По столбцам** (Columns), а в группе **Тип** (Type) — в положение **Арифметическая** (Linear). В поле **Шаг** (Step Value) вводим значение 0.1, а в поле **Предельное значение** (Stop Value) — 1. После нажатия кнопки **ОК** будет выполнено построение прогрессии.

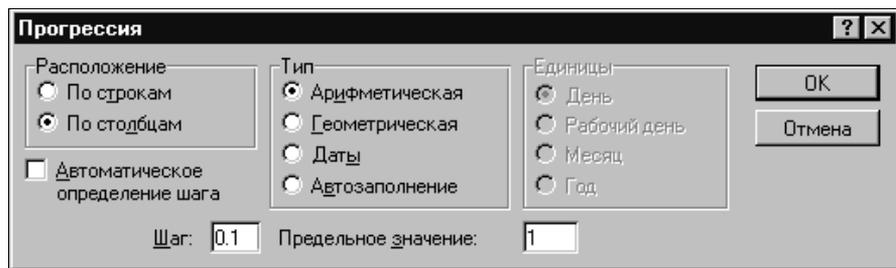


Рис. 2.3. Диалоговое окно **Прогрессия**

С помощью команды **Правка, Заполнить, Прогрессия** (Edit, Fill, Series) можно создавать также геометрические прогрессии. На рис. 2.4 приведен при-

мер построения арифметической и геометрической прогрессий по столбцам. Начальное значение обеих прогрессий равно 1, шаг — 1,2, а предельное значение — 4.

	А	В
1	Арифметическая	Геометрическая
2	1	1
3	2.2	1.2
4	3.4	1.44
5		1.728
6		2.0736
7		2.48832
8		2.985984
9		3.5831808

Рис. 2.4. Построение арифметической и геометрической прогрессий

Вернемся к нашему примеру построения графика. В ячейку В1 введем формулу

$=\text{COS}(\text{ПИ}() * \text{A1}) ^ 2$

Ввод формул в ячейку можно производить с клавиатуры или с помощью диалогового окна **Мастер функций** (Function Wizard), вызываемого командой **Вставка, Функция** (Insert, Function) или нажатием кнопки  панели инструментов **Стандартная** (Standard). Мастер функций содержит список всех встроенных в Excel функций, а также справки по синтаксису функций и примеры их применения.

Рассмотрим приемы работы с мастером функций на примере ввода вышеупомянутой формулы. Выделим ячейку В1 и нажмем кнопку  панели инструментов **Стандартная** (Standard). На экране появится первое диалоговое окно **Мастер функций** (Function Wizard) (рис. 2.5).

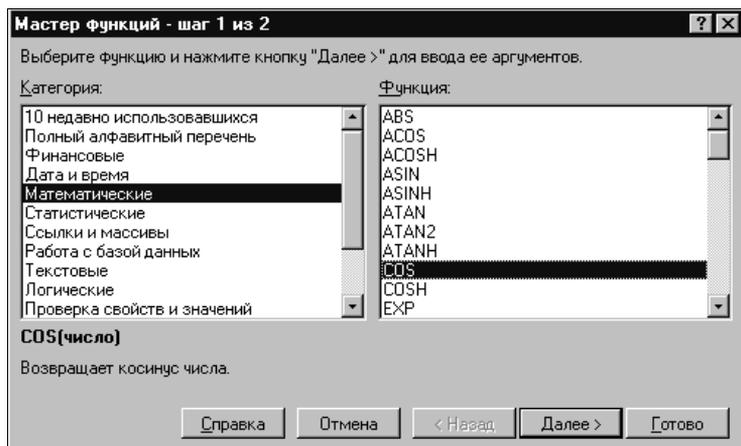


Рис. 2.5. Первое диалоговое окно **Мастер функций**

Это окно содержит два списка: **Категория** (Function Category) — список, включающий 11 категорий функций, и **Функция** (Function Name) — список имен функций, входящих в выбранную категорию. Категория **Полный алфавитный перечень** (All) содержит все встроенные функции Excel, имена которых упорядочены в алфавитном порядке. Категория **10 недавно использовавшихся** (Most Recently Used) содержит имена десяти недавно использовавшихся функций. Она ускоряет вызов функций, постоянно используемых пользователем.

Функция \cos относится к категории **Математические** (Math&Trig). Выберем эту функцию и нажмем кнопку **Далее** (Next). На экране появится второе диалоговое окно **Мастер функций** (Function Wizard) (рис. 2.6).

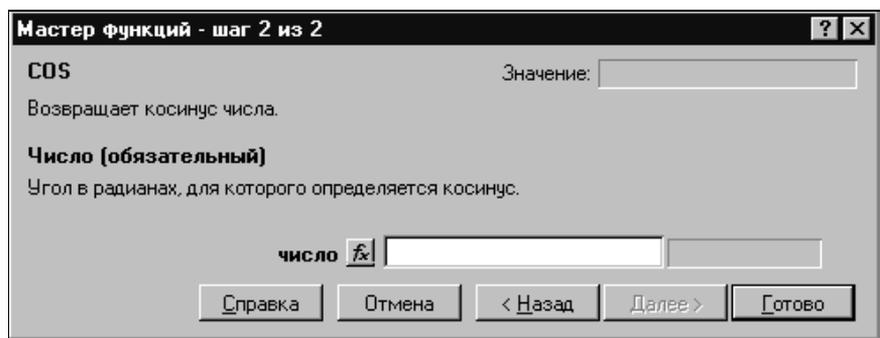


Рис. 2.6. Второе диалоговое окно **Мастер функций**

В поле **число** (Number) второго диалогового окна **Мастер функций** (Function Wizard) вводится аргумент функции. В рассматриваемом примере это $\pi() * A1$. Конечно, его можно ввести с клавиатуры. Однако этот аргумент содержит встроенную функцию $\pi()$, поэтому лучше, нажав кнопку **fx**, расположенную перед полем **число** (Number), еще раз вызвать мастер функций (эта кнопка предназначена для построения суперпозиции функций).

В первом диалоговом окне **Мастер функций** (Function Wizard) выберем функцию $\pi()$ из категории **Математические** (Math&Trig). Так как функция $\pi()$ не имеет аргументов, то нет необходимости переходить ко второму шагу мастера функций. Достаточно, нажав кнопку **Готово** (Finish), вернуться в диалоговое окно функции \cos (рис. 2.7).

Появившееся окно отличается от окна, представленного на рис. 2.6, тем, что в поле **число** (Number) введена функция $\pi()$. С помощью клавиатуры введем в это поле * и, щелкнув ячейку A1 рабочего листа, введем A1 (рис. 2.8). Конечно, ссылку на ячейку можно ввести также с помощью клавиатуры, однако, такой способ ввода (щелчком на соответствующей ячейке) дает дополнительную проверку правильности ввода.

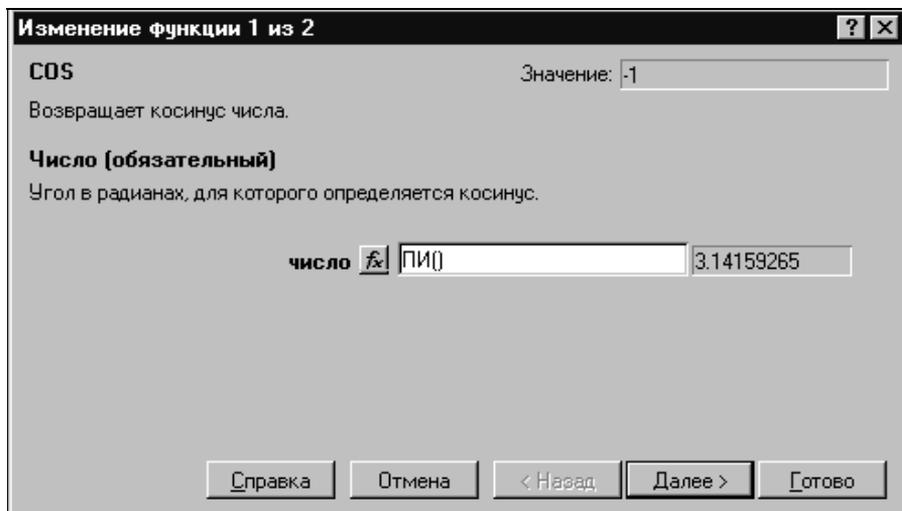


Рис. 2.7. Диалоговое окно **Изменение функции** для построения суперпозиции функций

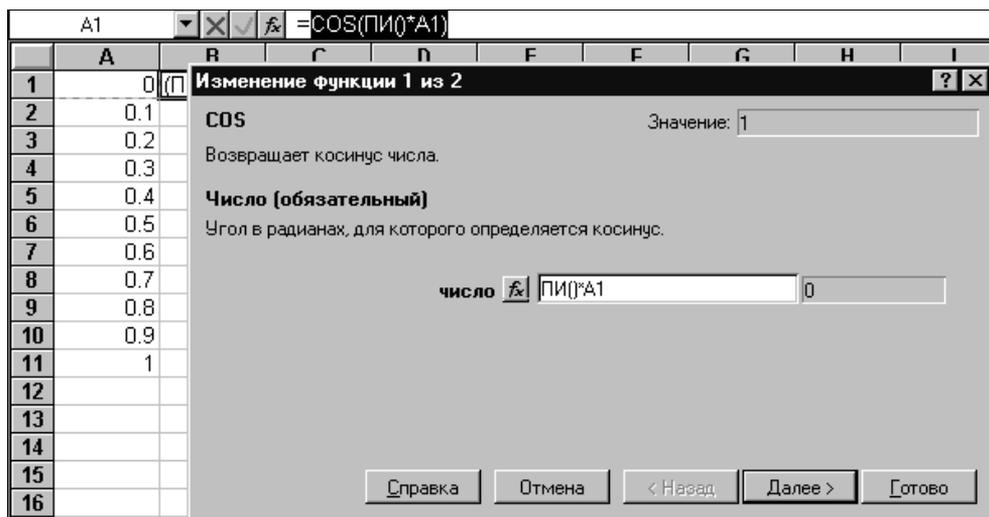


Рис. 2.8. Второе диалоговое окно **Мастер функций** с заполненным полем **число**

После нажатия кнопки **Готово** (Finish) в ячейку B1 будет введена формула $=\text{COS}(\text{ПИ}() * A1)$

Отметим, что при использовании мастера функций перед вводом формулы в ячейку не надо вводить знак "=", т. к. мастер функций введет его сам. Теперь

с помощью клавиатуры добавьте в эту формулу операцию возведения в квадрат. После всех описанных действий в ячейку B1 будет введена формула

$$=COS(ПИ()*A1)^2$$

Для того чтобы завершить процесс табулирования функции, выделим ячейку B1, установим указатель мыши на маркере заполнения этой ячейки и протащим его вниз до ячейки B11.

Итак, таблица значений функции создана. Для построения графика функции выделим диапазон ячеек A1:B11, содержащий таблицу значений функции и ее аргумента, и вызовем мастер диаграмм. Вызов мастера диаграмм производится либо с помощью команды **Вставка, Диаграмма, На этом листе** (Insert, Chart, On this Sheet) или команды **Вставка, Диаграмма, На новом листе** (Insert, Chart, As New Sheet), либо нажатием кнопки  панели инструментов **Стандартная** (Standard).

Протаскивая указатель мыши на рабочем листе, выделяем прямоугольную область, где будет построен график (рис. 2.9).

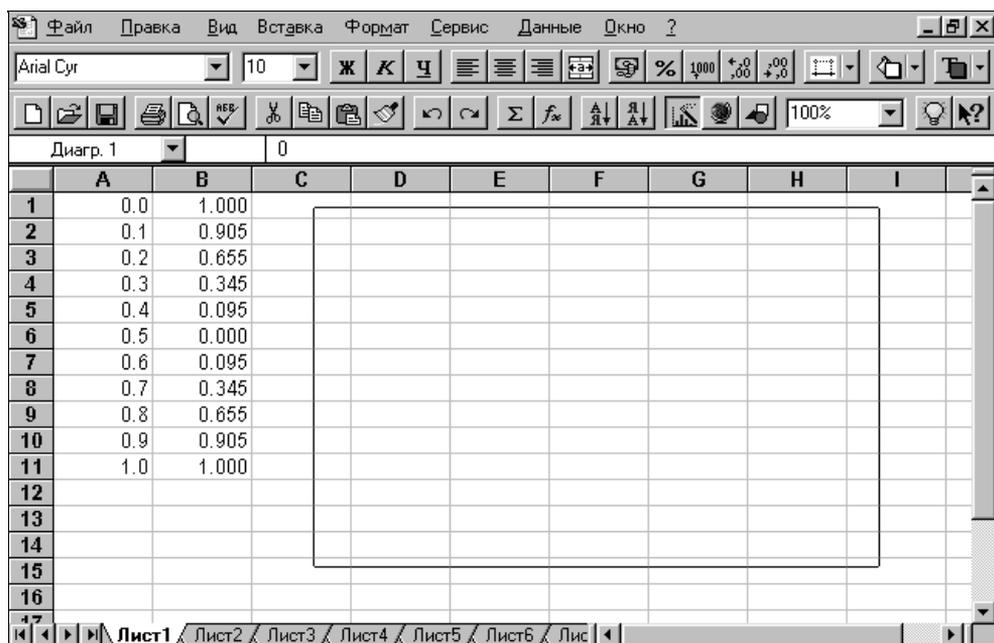


Рис. 2.9. Выделение указателем мыши области построения диаграммы

Проверим, правильно ли введен в поле ввода **Диапазон** (Range) первого диалогового окна **Мастер диаграмм** (Chart Wizard) диапазон ячеек, по которому строится график (рис. 2.10).

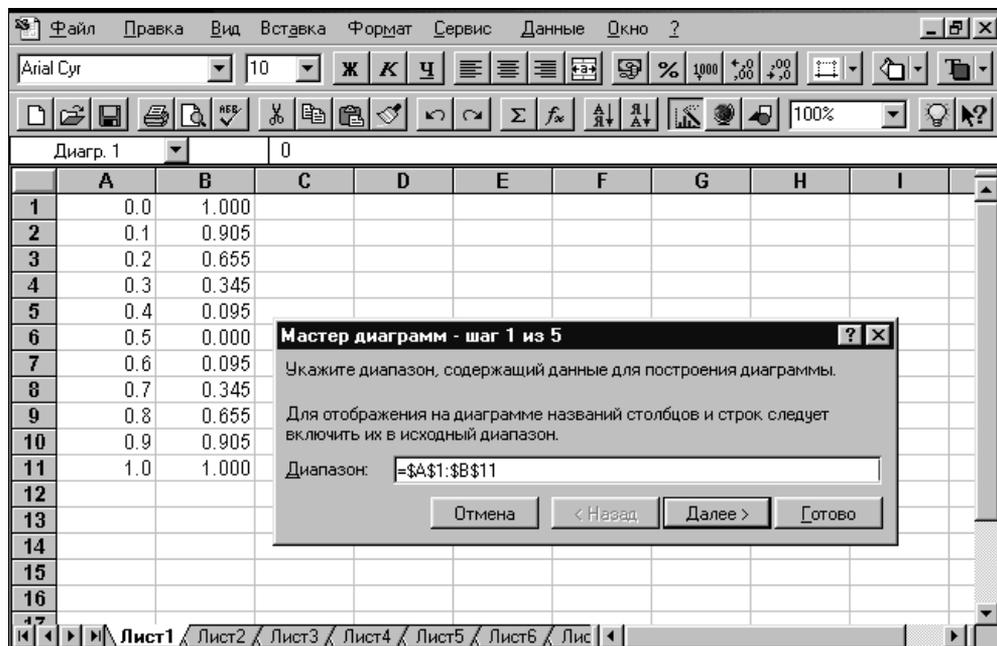


Рис. 2.10. Первое диалоговое окно **Мастер диаграмм**

На втором шаге мастера диаграмм выбираем тип диаграммы — **График** (Line) (рис. 2.11).

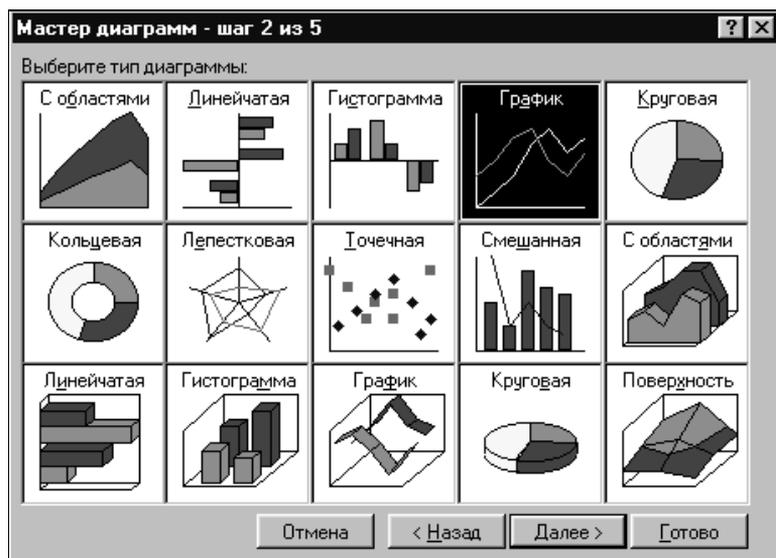


Рис. 2.11. Второе диалоговое окно **Мастер диаграмм**

На третьем шаге мастера диаграмм выбираем вид графика, например **10** — сглаженный график (рис. 2.12).

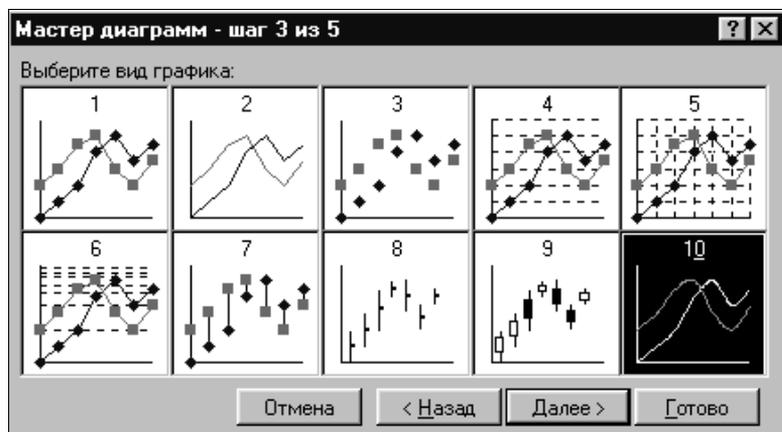


Рис. 2.12. Третье диалоговое окно **Мастер диаграмм**

На четвертом шаге мастера диаграмм заполняем диалоговое окно следующим образом. В группе **Ряды данных находятся** (Data Series in) устанавливаем переключатель в положение **В столбцах** (Columns). В поле **Считать метками оси X** введем 1 (номер столбца, из которого берутся метки оси X), а в поле **Считать метками легенды** — 0 (т. к. легенда — обозначение графиков различных функций разными цветами — нам не нужна) (рис. 2.13).



Рис. 2.13. Четвертое диалоговое окно **Мастер диаграмм**

На пятом шаге мастера диаграмм заполняем диалоговое окно следующим образом. В группе **Добавить легенду?** (Add a Legend?) установим переключат-

тель в положение **Нет** (No). В поле **Название диаграммы** (Chart Title) введем **График функции**, а в группе **Название по оси** (Axes Title) в поля **Категорий (X)** (Category (X)) и **Значений (Y)** (Value(Y)) введем x и y , соответственно (рис. 2.14).



Рис. 2.14. Пятое диалоговое окно **Мастер диаграмм**

Нажатие кнопки **Готово** (Finish) завершает построение графика (рис. 2.15).

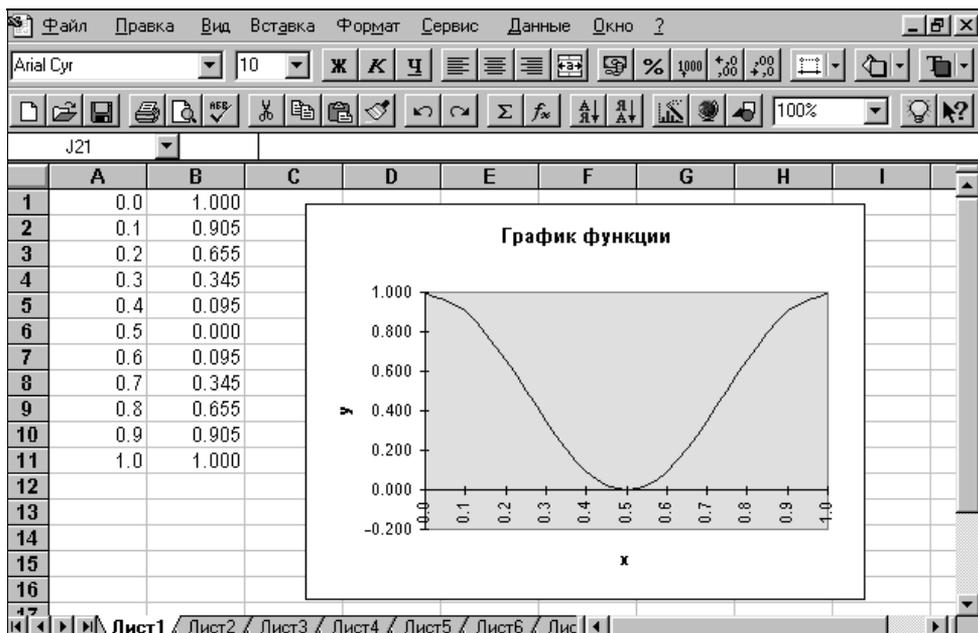


Рис. 2.15. Результат построения графика функции

Наиболее часто в Excel используются следующие стандартные математические функции:

ABS (аргумент)	Абсолютная величина
ACOS (аргумент)	Арккосинус
ASIN (аргумент)	Арсинус
ATAN (аргумент)	Арктангенс
COS (аргумент)	Косинус
EXP (аргумент)	Экспонента
LN (аргумент)	Натуральный логарифм
LOG10 (аргумент)	Десятичный логарифм
LOG (аргумент; основание)	Логарифм аргумента по данному основанию (если основание опущено, то оно полагается равным 10)
ПИ ()	π
SIN (аргумент)	Синус
TAN (аргумент)	Тангенс

2.2. Построение графика функции с одним условием

Рассмотрим пример построения графика функции

$$y = \begin{cases} \frac{1 + |0,2 - x|}{1 + x + x^2}, & x < 0,5, \\ x^{1/3}, & x \geq 0,5 \end{cases}$$

при $x \in [0, 1]$.

Этот график строится так же, как в разделе 2.1, за одним исключением — в ячейку B1 вводится формула:

=ЕСЛИ (A1<0.5 ; (1+ABS (0.2-A1)) / (1+A1+A1^2) ; A1^(1/3))

Синтаксис логической функции ЕСЛИ (IF):

ЕСЛИ (лог_выражение; значение_если_истина; значение_если_ложь)

Функция ЕСЛИ возвращает значение_если_истина, ЕСЛИ лог_выражение имеет значение ИСТИНА (TRUE), и значение_если_ложь — ЕСЛИ лог_выражение имеет

значение ЛОЖЬ (FALSE). Функция ЕСЛИ используется для проверки значений формул и организации переходов в зависимости от результатов этой проверки.

Другими логическими функциями являются:

И (логическое_значение1;
логическое_значение2; ...)
(AND)

Возвращает значение ИСТИНА, если все аргументы имеют значение ИСТИНА; возвращает значение ЛОЖЬ, если хотя бы один аргумент имеет значение ЛОЖЬ; например, И(2 + 2 = 4; 2 + 3 = 5) равняется ИСТИНА; если ячейка В4 содержит число между 1 и 100, то И(1<В4; В4<100) равняется ИСТИНА

ИЛИ (логическое_значение1;
логическое_значение2; ...)
(OR)

Возвращает ИСТИНА, если хотя бы один из аргументов имеет значение ИСТИНА; возвращает ЛОЖЬ, если все аргументы имеют значение ЛОЖЬ; например, И(2 + 2 = 4; 2 + 3 = 6) возвращает ИСТИНА; если ячейка В4 содержит число меньше 1 или больше 100, то ИЛИ(В4<1; В4>100) возвращает ИСТИНА

НЕ (логическое_значение) (NOT)

Меняет на противоположное логическое значение своего аргумента; например НЕ(2 + 2 = 5) возвращает ИСТИНА; если ячейка В4 содержит число меньше 1 или больше 100, то НЕ(ИЛИ(В4<1; В4>100)) возвращает ЛОЖЬ

2.3. Построение графика функции с двумя условиями

Рассмотрим пример построения графика функции

$$y = \begin{cases} 1 + \ln(1 + x), & x < 0,2, \\ \frac{1 + x^{1/2}}{1 + x}, & x \in [0,2, 0,8], \\ 2e^{-2x}, & x > 0,8 \end{cases}$$

при $x \in [0, 1]$.

График строится так же, как в разделе 2.1, только в ячейку В1 вводится формула:

=ЕСЛИ(A1<0.2;1+LN(1+A1);ЕСЛИ(И(A1>=0.2;A1<=0.8);(1+A1^(1/2))/(1+A1);2*EXP(-2*A1)))

Заметим, что в ячейку B1 можно ввести и более простую формулу, которая приведет к тому же результату:

```
=ЕСЛИ (A1<0.2;1+LN (1+A1) ;ЕСЛИ (A1<=0.8 ; (1+A1^(1/2)) / (1+A1) ;
2*EXP (-2*A1) ) )
```

2.4. Построение двух графиков в одной системе координат

Рассмотрим пример построения в одной системе координат графиков следующих двух функций

$$y = 2\sin(x) \text{ и}$$

$$z = 3\cos(2x) - \sin(x)$$

при $x \in [-3, 0]$.

В диапазон ячеек A2:A17 вводим значения переменной x от -3 до 0 с шагом $0,2$. В ячейки B1 и C1 вводим y и z , соответственно. В ячейки B2 и C2 вводим формулы

$$=2 * \text{SIN} (A2)$$

$$=3 * \text{COS} (2 * A2) - \text{SIN} (A2)$$

Выделим диапазон B2:C2, установим указатель мыши на маркере заполнения этого диапазона и протащим его вниз так, чтобы заполнить диапазон B2:C17. Выделим диапазон ячеек A1:C17, в который внесены: таблица значений двух функций, их общий аргумент и заголовки столбцов B и C, и вызовем мастер диаграмм. С помощью мыши на рабочем листе выделим прямоугольную область, где будет построен график. Проверим, правильно ли введен в первом диалоговом окне **Мастер диаграмм** (Chart Wizard) диапазон ячеек, по которому строится график.

На втором шаге мастера диаграмм выбираем тип диаграммы — **График** (Line).

На третьем шаге мастера диаграмм выбираем тип графика, например 1 (разновидность точечного графика).

На четвертом шаге мастера диаграмм заполняем диалоговое окно следующим образом. В группе **Ряды данных находятся** (Data series in) устанавливаем переключатель в положение **В столбцах** (Columns). В поле **Считать метками оси X** введем 1 (номер столбца), а в поле **Считать метками легенды** — 1 (номер строки) (рис. 2.13).

На пятом шаге мастера диаграмм заполняем диалоговое окно следующим образом. В группе **Добавить легенду?** (Add a Legend?) установим переключатель в положение **Да** (Yes). В поле **Название диаграммы** (Chart Title) введем

Графики функций, а в группе **Название по оси** (Axes Title) в поле **Категорий (X)** (Category (X)) введем x , а в поле **Значений (Y)** (Value (Y)) введем y и z . Нажатие кнопки **Готово** (Finish) завершает построение графика (рис. 2.16).

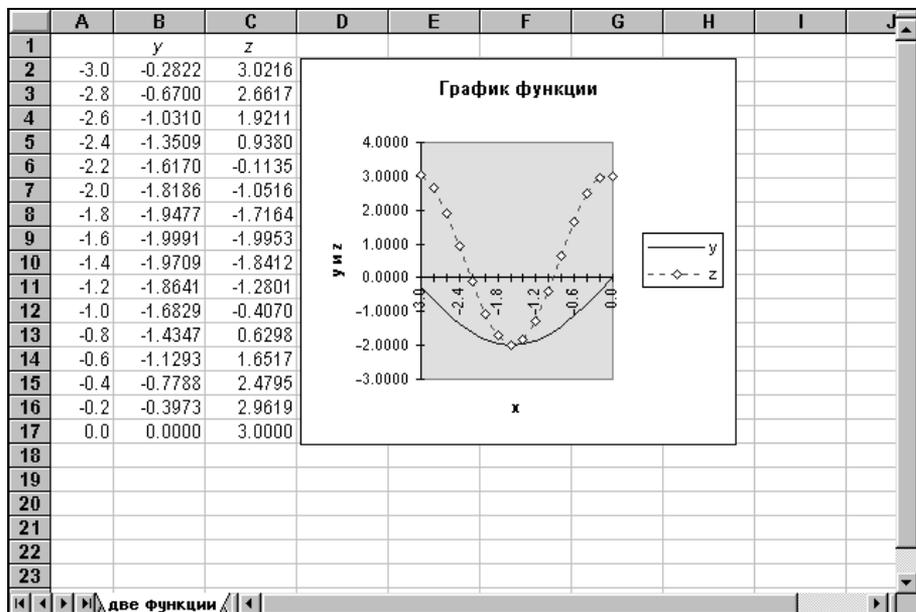


Рис. 2.16. Графики двух функций, построенные в одной системе координат

Как видно из рис. 2.16, графики функций y и z различаются, для наглядности, по типу линий. Как это делается?

График, внешний вид которого мы хотим изменить, выделяется и с помощью контекстного меню вызывается диалоговое окно **Форматирование элемента данных** (Format Data Entry) (рис. 2.17), которое позволяет изменять тип, толщину и цвет линии, а также тип, цвет и фон маркера.

2.5. Построение поверхности

Рассмотрим пример построения поверхности

$$z = x^2 - y^2$$

при $x, y \in [-1, 1]$.

В диапазон ячеек B1:L1 введем последовательность значений: $-1, -0.8, \dots, 1$ переменной x , а в диапазон ячеек A2:A12 — последовательность значений: $-1, -0.8, \dots, 1$ переменной y . В ячейку B2 введем формулу

$$= \$A2^2 - B\$1^2$$

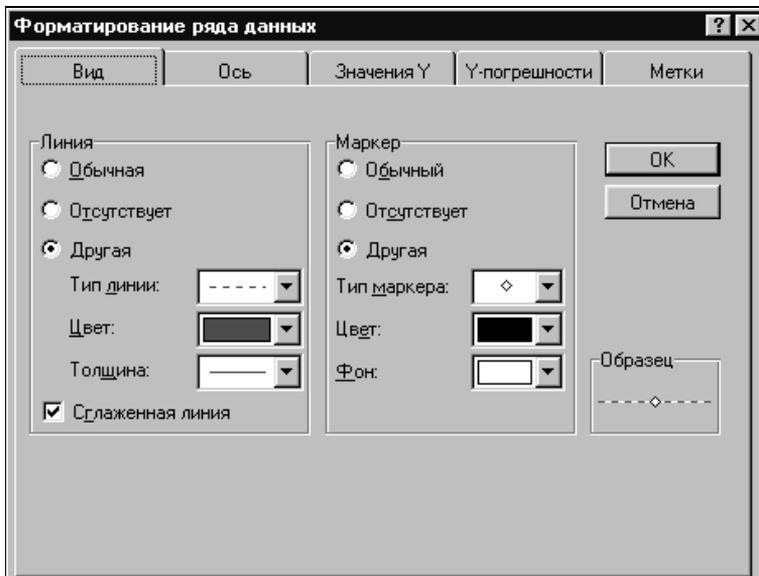


Рис. 2.17. Диалоговое окно **Форматирование элемента данных**

Выделим эту ячейку, установим указатель мыши на ее маркере заполнения и протаскиваем его так, чтобы заполнить диапазон B2:L12.

Знак \$, стоящий перед буквой в имени ячейки, дает абсолютную ссылку на столбец с данным именем, а знак \$, стоящий перед цифрой — абсолютную ссылку на строку с этим именем. Поэтому при протаскивании формулы из ячейки B2 в ячейки диапазона B2:L12 в них будет найдено значение z при соответствующих значениях x и y .

Итак, таблица значений функции z при различных значениях переменных создана (рис. 2.18).

Перейдем к построению поверхности. Выделим диапазон ячеек A1:L12, содержащий таблицу значений функции и ее аргументов, и вызовем мастер диаграмм. С помощью мыши на рабочем листе выделим прямоугольную область, где будет построен график. Проверим, правильно ли введен в первом диалоговом окне **Мастер диаграмм** (Chart Wizard) диапазон ячеек, по которому будет строиться поверхность.

На втором шаге мастера диаграмм выбираем тип диаграммы — **Поверхность** (3D Surface).

На третьем шаге мастера диаграмм выбираем вид поверхности, например **1**.

На четвертом шаге мастера диаграмм заполняем диалоговое окно следующим образом. В группе **Ряды данных находятся** (Data Series in) устанавливаем переключатель в положение **В столбцах** (Columns). В поле **Считать мет-**

ками оси **X** введем 1 (номер столбца), а в поле **Считать метками оси Значений** — 1 (номер строки) (рис. 2.13).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		-1.0	-0.8	-0.6	-0.4	-0.2	0.0	0.2	0.4	0.6	0.8	1.0		
2	-1.0	0.00	0.36	0.64	0.84	0.96	1.00	0.96	0.84	0.64	0.36	0.00		
3	-0.8	-0.36	0.00	0.28	0.48	0.60	0.64	0.60	0.48	0.28	0.00	-0.36		
4	-0.6	-0.64	-0.28	0.00	0.20	0.32	0.36	0.32	0.20	0.00	-0.28	-0.64		
5	-0.4	-0.84	-0.48	-0.20	0.00	0.12	0.16	0.12	0.00	-0.20	-0.48	-0.84		
6	-0.2	-0.96	-0.60	-0.32	-0.12	0.00	0.04	0.00	-0.12	-0.32	-0.60	-0.96		
7	0.0	-1.00	-0.64	-0.36	-0.16	-0.04	0.00	-0.04	-0.16	-0.36	-0.64	-1.00		
8	0.2	-0.96	-0.60	-0.32	-0.12	0.00	0.04	0.00	-0.12	-0.32	-0.60	-0.96		
9	0.4	-0.84	-0.48	-0.20	0.00	0.12	0.16	0.12	0.00	-0.20	-0.48	-0.84		
10	0.6	-0.64	-0.28	0.00	0.20	0.32	0.36	0.32	0.20	0.00	-0.28	-0.64		
11	0.8	-0.36	0.00	0.28	0.48	0.60	0.64	0.60	0.48	0.28	0.00	-0.36		
12	1.0	0.00	0.36	0.64	0.84	0.96	1.00	0.96	0.84	0.64	0.36	0.00		
13														
14														
15														
16														

Рис. 2.18. Результат табуляции функции двух переменных $z = x^2 - y^2$

На пятом шаге мастера диаграмм заполним диалоговое окно следующим образом. В группе **Добавить легенду?** (Add a Legend?) установим переключатель в положение **Нет** (No). В поле **Название диаграммы** (Chart Title) введем **Поверхность**, а в группе **Название по оси** (Axes Title) в поля **Категорий (X)** (Category(X)), **Значений (Z)** (Value (Z)) и **Рядов (Y)** (Series(Y)) введем **x**, **z** и **y**, соответственно. Нажатие кнопки **Готово** (Finish) завершает построение графика (рис. 2.19).

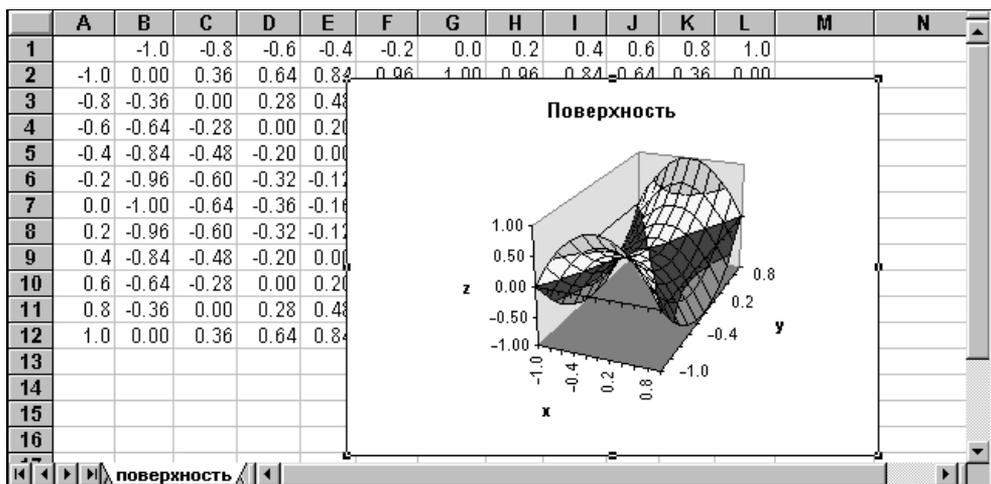


Рис. 2.19. Поверхность $z = x^2 - y^2$

2.6. Нахождение корней уравнения

Рассмотрим пример нахождения всех корней уравнения

$$x^3 - 0,01x^2 - 0,7044x + 0,139104 = 0.$$

Отметим, что у полинома третьей степени имеется не более трех вещественных корней. Для нахождения корней их предварительно нужно локализовать. С этой целью необходимо построить график функции или ее протабулировать. Например, протабулируем наш полином на отрезке $[-1, 1]$ с шагом $0,2$. Результат табуляции приведен на рис. 2.20, где в ячейку B2 введена следующая формула:

$$=A2^3-0.01*A2^2-0.7044*A2+0.139104$$

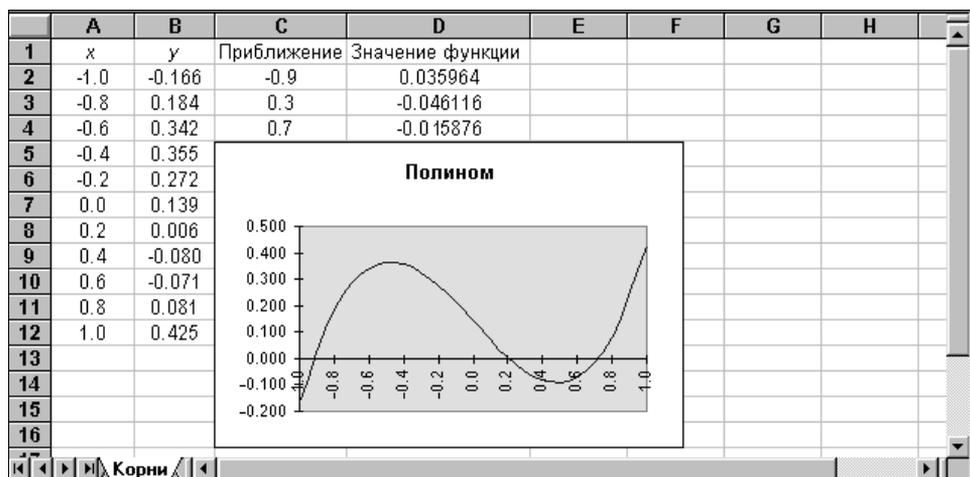


Рис. 2.20. Локализация корней полинома.

Из рис. 2.20 видно, что полином меняет знак на интервалах: $[-1, -0,8]$, $[0,2, 0,4]$ и $[0,6, 0,8]$. Это означает, что на каждом из них имеется корень данного полинома. Поскольку полином третьей степени имеет не более трех действительных корней, значит мы локализовали все его корни.

Найдем корни полинома методом последовательных приближений с помощью команды **Сервис, Подбор параметра** (Tools, Goal Seek). Относительная погрешность вычислений и предельное число итераций задаются на вкладке **Вычисления** (Calculation) диалогового окна **Параметры** (Options), открываемого командой **Сервис, Параметры** (Tools, Options) (рис. 2.21).

Зададим относительную погрешность и предельное число итераций, равными 0.00001 и 1000 , соответственно. В качестве начальных значений приближений к корням можно взять любые точки из отрезков локализации корней.

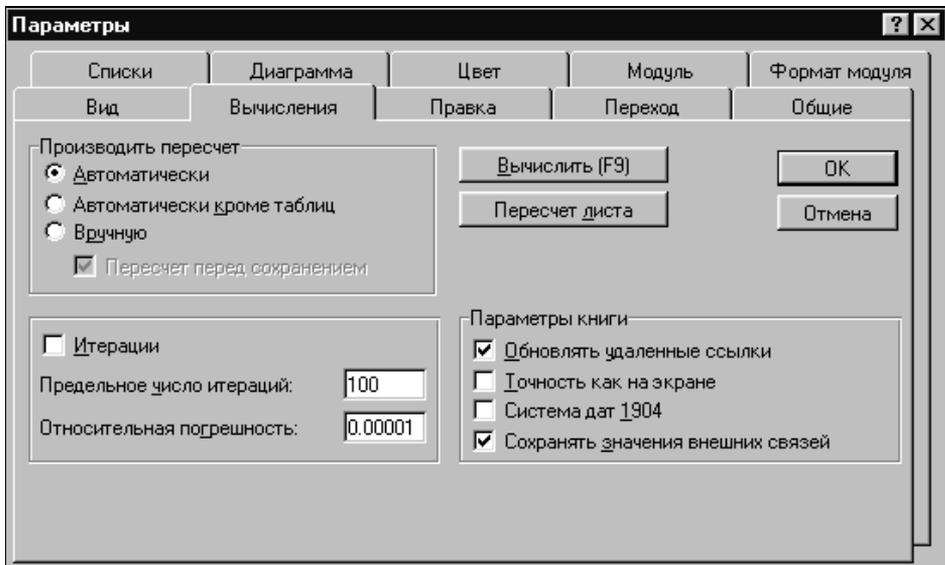


Рис. 2.21. Вкладка **Вычисления** диалогового окна **Параметры**

Возьмем, например, их средние точки: -0.9 , 0.3 и 0.7 и введем их в диапазон ячеек C2:C4. В ячейку D2 введем формулу

$$=C2^3 - 0.01 * C2^2 - 0.7044 * C2 + 0.139104$$

Выделим эту ячейку и с помощью маркера заполнения протащим введенную в нее формулу на диапазон D2:D4. Таким образом, в ячейках D2:D4 вычисляются значения полинома при значениях аргумента, введенного в ячейки C2:C4, соответственно.

Теперь выберем команду **Сервис, Подбор параметра** (Tools, Goal Seek) и заполним диалоговое окно **Подбора параметра** (Goal Seek) (рис. 2.22) следующим образом.

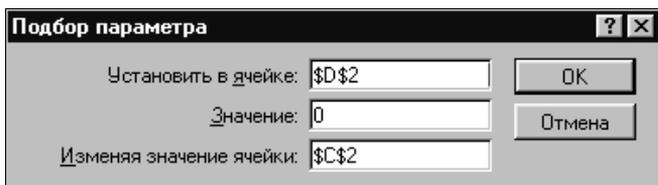


Рис. 2.22.
Диалоговое окно
Подбор параметра

В поле **Установить в ячейке** (Set Cell) введем D2. Отметим, что в этом поле дается ссылка на ячейку, в которую введена формула, вычисляющая значение левой части уравнения. Для нахождения корня уравнения с помощью

средства подбора параметров надо записать уравнение так, чтобы его правая часть не содержала переменную.

В поле **Значение** (To Value) вводим 0 (в этом поле указывается правая часть уравнения). В поле **Изменяя значение ячейки** (By Changing Cell) введем C2 (в этом поле дается ссылка на ячейку, отведенную под переменную).

Вводить ссылки на ячейки в поля диалогового окна **Подбор параметра** (Goal Seek) удобнее не с клавиатуры, а щелчком на соответствующей ячейке. При этом Excel автоматически будет превращать их в абсолютные ссылки (в нашем примере \$D\$2 и \$C\$2).

После нажатия кнопки **ОК** средство подбора параметров находит приближенное значение корня, которое помещает в ячейку C2. В данном случае оно равно -0.919999 . Как выглядит диалоговое окно **Результат подбор параметра** (Goal Seek Status) после успешного завершения поиска решения, показано на рис. 2.23.

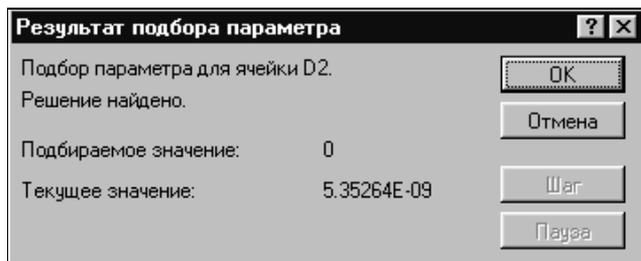


Рис. 2.23. Диалоговое окно **Результат подбор параметра**

Аналогично в ячейках C3 и C4 находим два оставшихся корня. Они равны 0.20999 и 0.71999 .

2.7. Нахождение корней уравнения методом деления отрезка пополам

Хорошим упражнением по работе в Excel является программирование на рабочем листе алгоритма нахождения корня уравнения $F(x) = 0$ методом деления отрезка пополам. Пусть непрерывная функция $F(x)$ имеет значения разных знаков на концах отрезка $[a, b]$, т. е.

$$F(a) F(b) < 0,$$

тогда уравнение $F(x) = 0$ имеет корень внутри этого отрезка, который называется отрезком локализации корня.

Пусть $c = (a + b)/2$ — середина отрезка $[a, b]$. Если $F(a) F(c) \leq 0$, то корень находится на отрезке $[a, c]$, который примем за новый отрезок локализации корня. Если $F(a) F(c) > 0$, то за новый отрезок локализации корня возьмем

[с, b]. Отметим, что новый отрезок локализации корня в два раза меньше первоначального.

Процесс деления отрезка локализации корня продолжаем до тех пор, пока его длина не станет меньше ε — точности нахождения корня. В этом случае любая точка отрезка локализации отличается от корня не более чем на $\varepsilon/2$.

На рис. 2.24 приведены результаты нахождения корня уравнения $x^2 - 2 = 0$ с точностью до 0,001 методом деления отрезка пополам.

	D	E	F	G	H	I	J	K
7	Точность нахождения корня	0.001						
8	a	b	c	Проверка знака	Значения функции в средней точке			
9	0	2	1	2	-1			
10	1	2	1.5	-0.25	0.25			
11	1	1.5	1.25	0.4375	-0.4375			
12	1.25	1.5	1.375	0.047851563	-0.109375			
13	1.375	1.5	1.4375	-0.007263184	0.06640625			
14	1.375	1.4375	1.40625	0.002456665	-0.02246094			
15	1.40625	1.4375	1.421875	-0.000488043	0.021728516			
16	1.40625	1.421875	1.414063	9.59635E-06	-0.00042725			
17	1.4140625	1.421875	1.417969	-4.54392E-06	0.010635376			
18	1.4140625	1.417969	1.416016	-2.17906E-06	0.00510025			
19	1.4140625	1.416016	1.415039	-9.97854E-07	0.002335548			
20	1.4140625	1.415039	1.414551	-4.07555E-07	0.000953913	Корень найден и равен	1.4146	
21								
22								
23								
24								

Рис. 2.24. Нахождение корня уравнения методом деления отрезка пополам

В ячейку E7 введена погрешность нахождения корня. За первоначальный отрезок локализации корня выбран отрезок [0, 2]. В ячейки D10, E10, F9, G9, H9 и I10 введены, соответственно, формулы

=ЕСЛИ (G9<=0; D9; F9)

=ЕСЛИ (G9<=0; F9; E9)

=(D9+E9) / 2

=(D9^2-2) * (F9^2-2)

=F9^2-2

=ЕСЛИ (E9-D9<=\$E\$7; "Корень найден и равен "&ТЕКСТ (F9; "0.0000") ; "")

которые протаскиваем вниз по столбцам до тех пор, пока не будет найден корень. Из рис. 2.24 видно, что корнем уравнения с точностью до 0,001 является 1,414.

Функция **ТЕКСТ** (Text) преобразует число в текстовую строку по указанному формату и имеет следующий синтаксис:

ТЕКСТ(значение; формат)

Аргументы:

значение Либо числовое значение, либо формула, вычисление которой дает числовое значение, либо ссылка на ячейку, содержащую числовое значение

формат Числовой формат с вкладки **Число** (Number) диалогового окна **Формат ячеек** (Format Cells), открываемого командой **Формат, Ячейки** (Format, Cells). Формат не может содержать звездочку (*) и не может быть **Общий** (General)

Приведем два примера работы функции **ТЕКСТ**:

ТЕКСТ(2.7153; "0.00р.") возвращает 2.72р.

ТЕКСТ(2.7153; "0.000") возвращает 2.715

2.8. Упражнения

Вариант 1

(а) Построить в разных системах координат при $x \in [-2, 2]$ графики следующих функций:

$$y = \sin(x)e^{-2x}, \quad g = \begin{cases} \frac{1+x^2}{\sqrt{1+x^4}}, & x \leq 0, \\ 2x + \frac{\sin^2(x)}{2+x}, & x > 0, \end{cases}$$

$$z = \begin{cases} \frac{1+|x|}{\sqrt[3]{1+x+x^2}}, & x \leq -1, \\ 2\ln(1+x^2) + \frac{1+\cos^4(x)}{2+x}, & x \in (-1, 0), \\ (1+x)^{3/5}, & x \geq 0. \end{cases}$$

(b) Построить в одной системе координат при $x \in [-2, 2]$ графики следующих двух функций:

$$y = 2\sin(x) \cos(x), \quad z = 3\cos^2(2x) \sin(x).$$

(c) Построить поверхность $z = x^2 - 2y^2$ при $x, y \in [-1, 1]$.

(d) Найти все корни уравнения $x^3 - 2,92x^2 + 1,4355x + 0,791136 = 0$.

Вариант 2

(a) Построить в разных системах координат при $x \in [-2, 2]$ графики следующих функций:

$$y = \frac{1 + x^2}{1 + 2x^2}, \quad g = \begin{cases} 3\sin(x) - \cos^2(x), & x \leq 0, \\ 3\sqrt{1 + x^2}, & x > 0, \end{cases}$$

$$z = \begin{cases} \frac{1 + x}{\sqrt[3]{1 + x^2}}, & x \leq 0, \\ -x + 2e^{-2x}, & x \in (0, 1), \\ |2 - x|^{1/3}, & x \geq 1. \end{cases}$$

(b) Построить в одной системе координат при $x \in [-2, 2]$ графики следующих двух функций:

$$y = 2\sin(\pi x) - 3\cos(\pi x), \quad z = \cos^2(2\pi x) - 2\sin(\pi x).$$

(c) Построить поверхность $z = 3x^2 - 2\sin^2(y)y^2$ при $x, y \in [-1, 1]$.

(d) Найти все корни уравнения $x^3 - 2,56x^2 - 1,3251x + 4,395006 = 0$.

Вариант 3

(a) Построить в разных системах координат при $x \in [-2, 1,5]$ графики следующих функций:

$$y = \frac{2 + \sin^2(x)}{1 + x^2}, \quad g = \begin{cases} \frac{3x^2}{1 + x^2}, & x \leq 0, \\ \sqrt{1 + \frac{2x}{1 + x^2}}, & x > 0, \end{cases}$$

$$z = \begin{cases} 3x + \sqrt{1 + x^2}, & x < 0, \\ 2 \cos(x)e^{-2x}, & x \in [0, 1], \\ 2 \sin(3x), & x > 1. \end{cases}$$

(b) Построить в одной системе координат при $x \in [-2, 2]$ графики следующих двух функций:

$$y = 5 \sin(\pi x) - \cos(3\pi x) \sin(\pi x), \quad z = \cos(2\pi x) - 2 \sin^3(\pi x).$$

(c) Построить поверхность $z = 5x^2 \cos^2(y) - 2y^2 e^y$ при $x, y \in [-1, 1]$.

(d) Найти все корни уравнения $x^3 + 2,84x^2 - 5,6064x - 14,766336 = 0$.

Вариант 4

(a) Построить в разных системах координат при $x \in [-1,5, 1,5]$ графики следующих функций:

$$y = \frac{1 + \cos(x)}{1 + e^{2x}}, \quad g = \begin{cases} \frac{3 + \sin^2(2x)}{1 + \cos^2(x)}, & x \leq 0, \\ 2\sqrt{1 + 2x}, & x > 0, \end{cases}$$

$$z = \begin{cases} \sqrt{1 + \frac{x^2}{1 + x^2}}, & x < 0, \\ 2 \cos^2(x), & x \in [0, 1], \\ \sqrt{1 + |2 \sin(3x)|^{1/3}}, & x > 1. \end{cases}$$

(b) Построить в одной системе координат при $x \in [-2, 2]$ графики следующих двух функций:

$$y = 3 \sin(2\pi x) \cos(\pi x) - \cos^2(3\pi x), \quad z = 2 \cos^2(2\pi x) - 3 \sin(3\pi x).$$

(c) Построить поверхность при $x, y \in [-1, 1]$

$$z = \begin{cases} 2x^2 - e^y, & |x + y| < 0,5, \\ x e^{2x} - y, & 0,5 \leq |x + y| < 1, \\ 2e^x - y e^y, & 1 \leq |x + y|. \end{cases}$$

(d) Найти все корни уравнения $x^3 + 1,41x^2 - 5,4724x - 7,380384 = 0$.

Вариант 5

(а) Построить в разных системах координат при $x \in [-1,8, 1,8]$ графики следующих функций:

$$y = \sqrt[4]{1 + e^{3x}}, \quad g = \begin{cases} \frac{3 + \sin(x)}{1 + x^2}, & x \leq 0, \\ 2x^2 \cos^2(x), & x > 0, \end{cases}$$

$$z = \begin{cases} |x|^{1/3}, & x < 0, \\ -2x + \frac{x}{1+x}, & x \in [0, 1), \\ \frac{|3-x|}{1+x}, & x \geq 1. \end{cases}$$

(б) Построить в одной системе координат при $x \in [0, 3]$ графики следующих двух функций:

$$y = 2\sin(\pi x) \cos(\pi x), \quad z = \cos^2(\pi x) \sin(3\pi x).$$

(с) Построить поверхность $z = 2x^2 \cos^2(x) - 2y^2$ при $x, y \in [-1, 1]$.

(д) Найти все корни уравнения $x^3 + 0,85x^2 - 0,4317x + 0,043911 = 0$.

Вариант 6

(а) Построить в разных системах координат при $x \in [-2, 1,8]$ графики следующих функций:

$$y = \frac{2+3x}{1+x+x^2}, \quad g = \begin{cases} \sqrt{1+2x^2-\sin^2(x)}, & x \leq 0, \\ \frac{2+x}{\sqrt[3]{2+e^{-0.1x}}}, & x > 0, \end{cases}$$

$$z = \begin{cases} \frac{1+x}{1+x^2}, & x < 0, \\ \sqrt{1+\frac{x}{1+x}}, & x \in [0, 1), \\ 2|\sin(3x)|, & x \geq 1. \end{cases}$$

(b) Построить в одной системе координат при $x \in [-3, 0]$ графики следующих двух функций:

$$y = 3\sin(3\pi x) \cos(2\pi x), \quad z = \cos^3(4\pi x) \sin(\pi x).$$

(c) Построить поверхность $z = 2e^{0,2x} x^2 - 2y^4$ при $x, y \in [-1, 1]$.

(d) Найти все корни уравнения $x^3 - 0,12x^2 - 1,4775x + 0,191906 = 0$.

Вариант 7

(a) Построить в разных системах координат при $x \in [-1,7, 1,5]$ графики следующих функций:

$$y = \frac{1+x}{1+\sqrt{2+x+x^2}}, \quad g = \begin{cases} \sqrt{1+x^2}, & x \leq 0, \\ \frac{1+x}{1+\sqrt[3]{1+e^{-0,2x}}}, & x > 0, \end{cases}$$

$$z = \begin{cases} \frac{1+x+x^2}{1+x^2}, & x < 0, \\ \sqrt{1+\frac{2x}{1+x^2}}, & x \in [0, 1), \\ 2|0,5 + \sin(x)|, & x \geq 1. \end{cases}$$

(b) Построить в одной системе координат при $x \in [-3, 0]$ графики следующих двух функций:

$$y = 2\sin(2\pi x) \cos(4\pi x), \quad z = \cos^2(3\pi x) - \cos(\pi x) \sin(\pi x).$$

(c) Построить поверхность $z = x^2 - 2e^{0,2y} y^2$ при $x, y \in [-1, 1]$.

(d) Найти все корни уравнения $x^3 + 0,77x^2 - 0,2513x + 0,016995 = 0$.

Вариант 8

(a) Построить в разных системах координат при $x \in [-1,5, 1,8]$ графики следующих функций:

$$y = \frac{1+x}{1+\sqrt{2+x+x^2}}, \quad g = \begin{cases} \sqrt{1+x^2}, & x \leq 0, \\ \frac{1+x}{1+\sqrt[3]{1+e^{-0,2x}}}, & x > 0, \end{cases}$$

$$z = \begin{cases} \frac{1+x+x^2}{1+x^2}, & x < 0, \\ \sqrt{1 + \frac{2x}{1+x^2}}, & x \in [0, 1), \\ 2|0.5 + \sin(x)|, & x \geq 1. \end{cases}$$

(b) Построить в одной системе координат при $x \in [-3, 0]$ графики следующих двух функций:

$$y = 2\sin(2\pi x) \cos(4\pi x), \quad z = \cos^2(3\pi x) - \cos(\pi x) \sin(\pi x).$$

(c) Построить поверхность $z = x^2 - 2e^{0,2y} y^2$ при $x, y \in [-1, 1]$.

(d) Найти все корни уравнения $x^3 + 0,77x^2 - 0,2513x + 0,016995 = 0$.

Вариант 8

(a) Построить в разных системах координат при $x \in [-1,5, 1,8]$ графики следующих функций:

$$y = \frac{1 + xe^{-x}}{2 + \sqrt{x^2 + \sin^2(x)}}, \quad g = \begin{cases} \sqrt{1 + |x|}, & x \leq 0, \\ \frac{1 + 3x}{2 + \sqrt[3]{1+x}}, & x > 0, \end{cases}$$

$$z = \begin{cases} 1 + \frac{3+x}{1+x^2}, & x < 0, \\ \sqrt{1 + (1-x)^2}, & x \in [0, 1), \\ \frac{1+x}{1 + \cos^2(x)}, & x \geq 1. \end{cases}$$

(b) Построить в одной системе координат при $x \in [0, 2]$ графики следующих двух функций:

$$y = \sin(3\pi x) + 2\sin(2\pi x) \cos(3\pi x), \quad z = \cos(\pi x) - \cos(3\pi x) \sin^2(\pi x).$$

(c) Построить поверхность при $x, y \in [-1, 1]$

$$z = \begin{cases} x - e^{2y}, & |x| + |y| < 0,5, \\ 2x^2 - e^y, & 0,5 \leq |x| + |y| < 1, \\ e^{2x} - y, & 1 \leq |x| + |y|. \end{cases}$$

(d) Найти все корни уравнения $x^3 + 0,88x^2 - 0,3999x + 0,037638 = 0$.

Вариант 9

(a) Построить в разных системах координат при $x \in [-1,4, 1,9]$ графики следующих функций:

$$y = \frac{1 + x e^{-x}}{2 + x^2} \sin^2(x), \quad g = \begin{cases} \frac{\sqrt{1 + |x|}}{2 + |x|}, & x \leq 0, \\ \frac{1 + x}{2 + \cos^3(x)}, & x > 0, \end{cases}$$

$$z = \begin{cases} \frac{1 + 2x}{1 + x^2}, & x < 0, \\ \sin^2(x) \sqrt{1 + x}, & x \in [0, 1), \\ \sin^2(x) e^{0,2x}, & x \geq 1. \end{cases}$$

(b) Построить в одной системе координат при $x \in [0, 2]$ графики следующих двух функций:

$$y = \cos(3\pi x) \sin(\pi x) + 2\sin(3\pi x) \cos(2\pi x), \quad z = \cos^2(\pi x) - \cos(3\pi x).$$

(c) Построить поверхность при $x, y \in [-1, 1]$

$$z = \begin{cases} x^2 - 3y^3, & x^2 + y^2 \leq 1, \\ 3x^2 - y^3, & x^2 + y^2 > 1. \end{cases}$$

(d) Найти все корни уравнения $x^3 + 0,78x^2 - 0,8269x + 0,146718 = 0$.

Вариант 10

(a) Построить в разных системах координат при $x \in [-1,4, 1,4]$ графики следующих функций

$$y = \frac{1 + x}{1 + \sqrt{|x|e^{-x} + |\sin(x)|}}, \quad g = \begin{cases} \sqrt[3]{1 + x^2}, & x \leq 0, \\ \sin^2(x) + \frac{1 + x}{1 + \cos^2(x)}, & x > 0, \end{cases}$$

$$z = \begin{cases} \frac{|x|}{1+x^2} e^{-2x}, & x < 0, \\ \sqrt{1+x^2}, & x \in [0, 1), \\ \frac{1+\sin(x)}{1+x} + 3x, & x \geq 1. \end{cases}$$

(b) Построить в одной системе координат при $x \in [0, 2]$ графики следующих двух функций:

$$y = 2\sin(2\pi x) \cos(\pi x) + \sin(3\pi x), \quad z = \cos(2\pi x) \sin^2(\pi x) - \cos(4\pi x).$$

(c) Построить поверхность $z = 3x^2 \sin^2(x) - 5e^{2y}$ при $x, y \in [-1, 1]$.

(d) Найти все корни уравнения $x^3 + 2,28x^2 - 1,9347x - 3,907574 = 0$.

Глава 3



Работа с массивами

Часто при работе с таблицами возникает необходимость применить одну и ту же операцию к целому диапазону ячеек или произвести расчеты по формулам, зависящим от большого массива данных. Для решения подобных задач Excel предоставляет простые и удобные средства, которые будут описаны в данной главе.

3.1. Простейшие операции над массивами

В качестве первого примера простой операции над массивами рассмотрим умножение массива A1:B2 на число 5. Выделите на рабочем листе область, например D1:E2, такого же размера, как и массив-множимое (рис. 3.1).

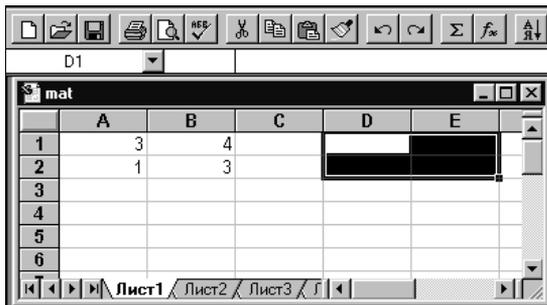


Рис. 3.1. Выделение диапазона для ввода результирующего массива

Теперь введите формулу

=A1:B2*5

Для этого установите курсор в строке формул и закончите ввод не как обычно, нажатием клавиши <Enter>, а нажатием клавиш <Ctrl>+<Shift>+

<Enter>). Таким образом вы сообщите программе, что необходимо выполнить операцию над массивом. При этом Excel заключит формулу в строке формул в фигурные скобки (рис.3.2):

```
{=A1:B2*5}
```

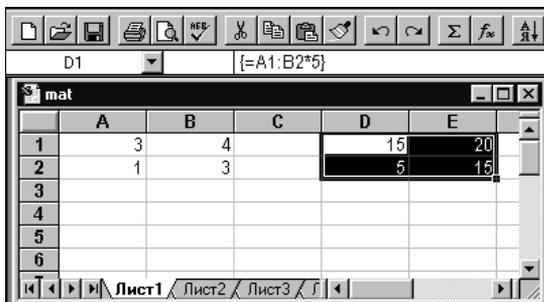


Рис. 3.2. Произведение массива на число

При работе с массивами формула действует на все ячейки диапазона. Нельзя изменять отдельные ячейки в операндах формулы. Аналогично можно вычислить:

Сумму (разность) массивов (рис. 3.3)

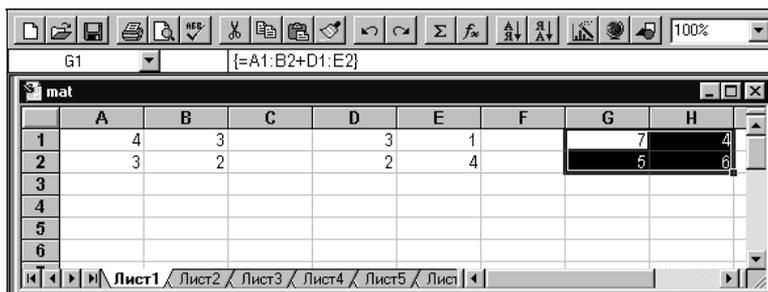


Рис. 3.3. Сумма двух массивов

Поэлементное произведение (деление) массивов (рис. 3.4)

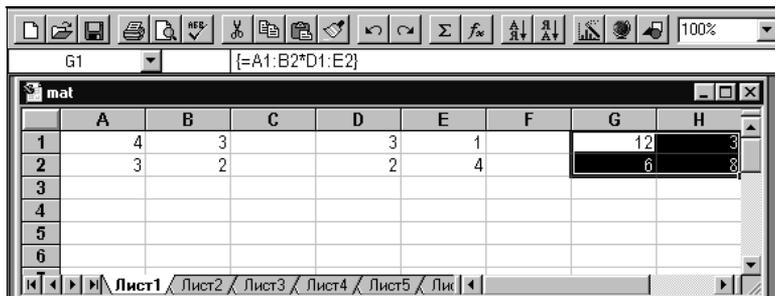


Рис. 3.4. Поэлементное произведение двух массивов

- Массив, каждый элемент которого связан посредством некоторой функции с соответствующим элементом первоначального массива (рис. 3.5)

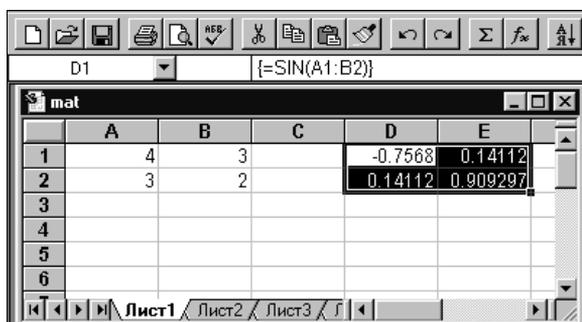


Рис. 3.5. Вычисление функции от каждого элемента массива

3.2. Встроенные функции для работы с матрицами

В Excel имеются следующие специальные функции для работы с матрицами:

МОБР (MINVERSE)	Обратная матрица
МОПРЕД (MDETERM)	Определитель матрицы
МУМНОЖ (MMULT)	Матричное произведение двух матриц
ТРАНСП (TRANSPOSE)	Транспонированная матрица

Во всех случаях при работе с матрицами перед вводом формулы надо выделить область на рабочем листе, куда будет выведен результат вычислений.

Решим в качестве примера систему линейных уравнений с двумя неизвестными, матрица коэффициентов которой записана в ячейки F1:G2, а свободные члены — в ячейки I1:I2 (рис. 3.6).

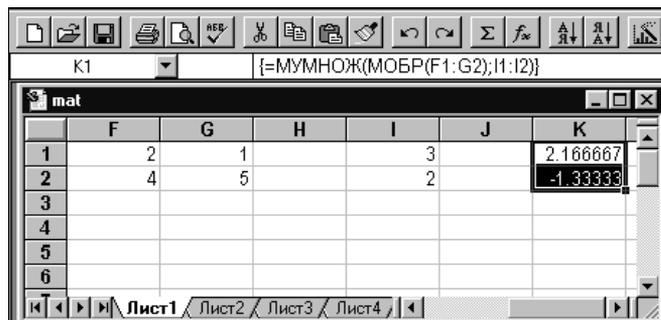


Рис. 3.6. Решение системы линейных уравнений

Для решения этой задачи вспомним, что решение линейной системы $AX = B$, где A — матрица коэффициентов, B — столбец (вектор) свободных членов, X — столбец (вектор) неизвестных, имеет вид $X = A^{-1}B$, где A^{-1} — матрица, обратная по отношению к A . Поэтому для решения нашей системы уравнений выделим под вектор решений диапазон K1:K2 и введем в него формулу, как показано на рис. 3.6.

Решим также систему линейных уравнений $A^2X = B$, где

$$A = \begin{pmatrix} 7 & 2 \\ 1 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

Для решения этой системы введем в диапазон ячеек A1:B2 элементы матрицы A , а в диапазон ячейки D1:D2 — элементы столбца свободных членов B . Выберем диапазон F1:F2, куда поместим элементы вектора решения, и введем следующую формулу:

```
{=МУМНОЖ(МОБР(МУМНОЖ(A1:B2;A1:B2));D1:D2)}
```

Рассмотрим пример вычисления квадратичной формулы $z = X^TAX$, где A — квадратная матрица, введенная в диапазон A1:B2, X — вектор, введенный в диапазон D1:D2, а символ $(^T)$ обозначает операцию транспонирования. Для вычисления z введем в ячейку F1 (рис. 3.7) формулу

```
{=МУМНОЖ(МУМНОЖ(ТРАНСП(D1:D2);A1:B2);D1:D2)}
```

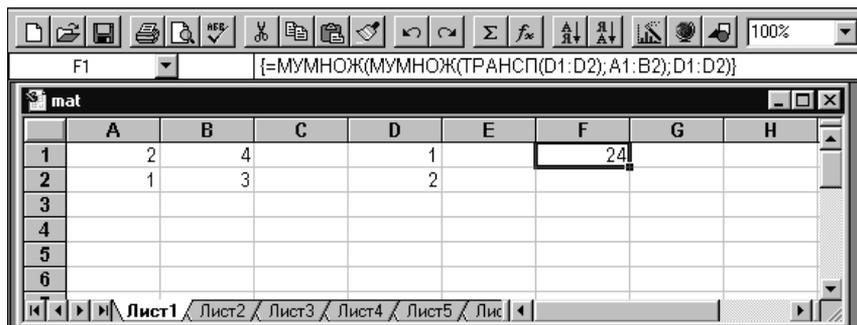


Рис. 3.7. Нахождение квадратичной формы

Хотя результатом этой формулы является число, не забудьте для ее ввода нажать клавиши $\langle \text{Ctrl} \rangle + \langle \text{Shift} \rangle + \langle \text{Enter} \rangle$. Если вы этого не сделаете, в ячейке F1 появится сообщение #ЗНАЧ!.

Вычислим значение квадратичной формы $z = Y^T A^T A Y$, где

$$A = \begin{pmatrix} 7 & 2 \\ 1 & 4 \end{pmatrix}, \quad Y = \begin{pmatrix} 3 \\ 1 \end{pmatrix}.$$

Для решения этой задачи введем в диапазон ячеек A1:B2 элементы матрицы A , а в диапазон D1:D2 — элементы столбца Y . Для вычисления квадратичной формы введем в ячейку F1 формулу

```
{=МУМНОЖ(ТРАНСП(D1:D2);МУМНОЖ(ТРАНСП(A1:B2);МУМНОЖ(A1:B2;D1:D2)))}
```

3.3. Пошаговое решение системы линейных уравнений методом Гаусса

Хорошим упражнением по работе с массивами является пошаговое программирование на рабочем листе решения системы линейных уравнений методом Гаусса.

На рис. 3.8 приведены результаты решения методом Гаусса следующей системы линейных уравнений:

$$2x_1 + 3x_2 + 7x_3 + 6x_4 = 1,$$

$$3x_1 + 5x_2 + 3x_3 + x_4 = 3,$$

$$5x_1 + 3x_2 + x_3 + 3x_4 = 4,$$

$$3x_1 + 3x_2 + x_3 + 6x_4 = 5.$$

	A	B	C	D	E	F	G	H	I	J	K
33	2	3	7	6	1		1	0	0	0	0.1857
34	3	5	3	1	3		0	1	0	0	0.7786
35	5	3	1	3	4		0	0	1	0	-0.6357
36	3	3	1	6	5		0	0	0	1	0.4571
37											
38	2	3	7	6	1						
39	0	0.5	-7.5	-8	1.5						
40	0	-4.5	-16.5	-12	1.5						
41	0	-1.5	-9.5	-3	3.5						
42											
43	2	3	7	6	1						
44	0	0.5	-7.5	-8	1.5						
45	0	0	-84	-84	15						
46	0	0	-32	-27	8						
47											
48	2	3	7	6	1						
49	0	0.5	-7.5	-8	1.5						
50	0	0	-84	-84	15						
51	0	0	0	5	2.2857						
52											
53											
54											
55											

Рис. 3.8. Пошаговое решение системы линейных уравнений методом Гаусса

В диапазоны ячеек A33:D36 и E33:E36 введены матрица коэффициентов и столбец свободных членов, соответственно. Содержимое ячеек A33:E33 скопировано в ячейки A38:E38, A43:E43 и A48:E48. В диапазон ячеек A39:E39 введена формула

$$={A34:E34-\$A\$33:\$E\$33*(A34/\$A\$33)}$$

обращающая в нуль коэффициент при x_1 во втором уравнении системы. Выделим диапазон A39:E39 и протащим маркер заполнения этого диапазона так, чтобы заполнить диапазон A39:E41. Это обратит в нуль коэффициент при x_1 в третьем и четвертом уравнениях системы. Скопируем значения из диапазона ячеек A39:E39 в диапазоны A44:E44 и A49:E49. Для копирования значений без формул воспользуйтесь командой **Правка, Специальная вставка** (Edit, Special Paste) и в открывшемся диалоговом окне **Специальная вставка** (Special Paste) в группе **Вставить** (Paste) установите переключатель в положение **Значения** (Value) (рис. 3.9).

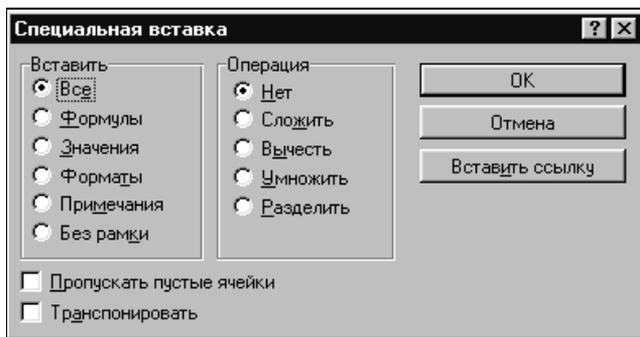


Рис. 3.9.
Диалоговое окно
Специальная вставка

В диапазон ячеек A45:E45 вводим формулу

$$={A40:E40-\$A\$39:\$E\$39*(B40/\$B\$39)}$$

Выделим диапазон A45:E45 и протащим маркер заполнения этого диапазона так, чтобы заполнить диапазон A45:E46. Это обратит в нуль коэффициент при x_2 в третьем и четвертом уравнениях системы. Копируем значения из диапазона ячеек A45:E45 в диапазон A50:E50. В диапазон ячеек A51:E51 вводим формулу

$$={A46:E46-\$A\$45:\$E\$45*(C46/\$C\$45)}$$

которая обращает в нуль коэффициент при x_3 четвертого уравнения системы.

Прямая прогонка метода Гаусса завершена. Обратная прогонка заключается в вводе в диапазоны G36:K36, G35:K35, G34:K34 и G33:K33, соответственно, следующих формул:

$$\{=A51:E51/D51\}$$

$$\{=(A50:E50-G36:K36*D50)/C50\}$$

$$\{=(A49:E49-G36:K36*D49-G35:K35*C49)/B49\}$$

$$\{=(A48:E48-G36:K36*D48-G35:K35*C48-G34:K34*B48)/A48\}$$

В диапазоне ячеек K33:K36 получено решение системы.

3.4. Упражнения

Вариант 1

Решить системы линейных уравнений $AX=B$, $A^3X=B$ и вычислить значение квадратичной формы $z = Y^T A^T A^2 Y$, где

$$A = \begin{pmatrix} 9 & 5 & 4 & 7 \\ 4 & 6 & 8 & 7 \\ 5 & 8 & 7 & 6 \\ 5 & 6 & 8 & 7 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 6 \\ 3 \\ 7 \end{pmatrix}, \quad Y = \begin{pmatrix} 2 \\ 6 \\ 2 \\ 3 \end{pmatrix}.$$

Вариант 2

Решить системы линейных уравнений $AX=B$, $A^2 A^T X=B$ и вычислить значение квадратичной формы $z = Y^T A^3 Y$, где

$$A = \begin{pmatrix} 9 & 6 & 3 & 8 \\ 4 & 6 & 7 & 4 \\ 2 & 3 & 5 & 3 \\ 4 & 8 & 3 & 7 \end{pmatrix}, \quad B = \begin{pmatrix} 3 \\ 1 \\ 4 \\ 2 \end{pmatrix}, \quad Y = \begin{pmatrix} 3 \\ 5 \\ 1 \\ 3 \end{pmatrix}.$$

Вариант 3

Решить системы линейных уравнений $AX=B$, $AA^T AX=B$ и вычислить значение квадратичной формы $z = Y^T A^T A^3 Y$, где

$$A = \begin{pmatrix} 2 & 3 & 5 & 2 \\ 5 & 2 & 7 & 5 \\ 4 & 2 & 1 & 7 \\ 7 & 5 & 4 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 3 \\ 2 \\ 3 \\ 2 \end{pmatrix}, \quad Y = \begin{pmatrix} 3 \\ 1 \\ 2 \\ 2 \end{pmatrix}.$$

Вариант 4

Решить системы линейных уравнений $AX = B$, $A^2 A^T AX = B$ и вычислить значение квадратичной формы $z = Y^T A^T AA^T Y$, где

$$A = \begin{pmatrix} 1 & 4 & 2 & 5 \\ 4 & 4 & 5 & 3 \\ 1 & 2 & 6 & 8 \\ 3 & 7 & 3 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 3 \\ 8 \\ 1 \\ 7 \end{pmatrix}, \quad Y = \begin{pmatrix} 1 \\ 2 \\ 5 \\ 1 \end{pmatrix}.$$

Вариант 5

Решить системы линейных уравнений $AX = B$, $AA^T A^2 X = B$ и вычислить значение квадратичной формы $z = Y^T A^3 A^T Y$, где

$$A = \begin{pmatrix} 9 & 6 & 3 & 8 \\ 4 & 6 & 7 & 4 \\ 2 & 3 & 5 & 3 \\ 4 & 8 & 3 & 7 \end{pmatrix}, \quad B = \begin{pmatrix} 3 \\ 1 \\ 4 \\ 2 \end{pmatrix}, \quad Y = \begin{pmatrix} 3 \\ 5 \\ 1 \\ 3 \end{pmatrix}.$$

Вариант 6

Решить системы линейных уравнений $AX = B$, $A^3 A^T X = B$ и вычислить значение квадратичной формы $z = Y^T A^2 A^T AY$, где

$$A = \begin{pmatrix} 2 & 4 & 7 & 4 \\ 4 & 1 & 6 & 2 \\ 8 & 3 & 6 & 7 \\ 6 & 3 & 5 & 7 \end{pmatrix}, \quad B = \begin{pmatrix} 2 \\ 0 \\ 3 \\ 1 \end{pmatrix}, \quad Y = \begin{pmatrix} 2 \\ 1 \\ 4 \\ 1 \end{pmatrix}.$$

Вариант 7

Решить системы линейных уравнений $AX = B$, $A^T A^3 X = B$ и вычислить значение квадратичной формы $z = Y^T AA^T A^2 Y$, где

$$A = \begin{pmatrix} 3 & 3 & 4 & 5 \\ 2 & 6 & 4 & 6 \\ 3 & 4 & 5 & 5 \\ 1 & 9 & 3 & 6 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 4 \\ 0 \\ 3 \end{pmatrix}, \quad Y = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 2 \end{pmatrix}.$$

Вариант 8

Решить системы линейных уравнений $AX = B$, $AA^T A^2 X = B$ и вычислить значение квадратичной формы $z = Y^T A^2 A^T A Y$, где

$$A = \begin{pmatrix} 2 & 1 & 5 & 2 \\ 5 & 2 & 2 & 6 \\ 2 & 2 & 1 & 2 \\ 1 & 3 & 3 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 3 \\ 0 \\ 2 \end{pmatrix}, \quad Y = \begin{pmatrix} 3 \\ 1 \\ 2 \\ 1 \end{pmatrix}.$$

Вариант 9

Решить системы линейных уравнений $AX = B$, $A^T A A^T X = B$ и вычислить значение квадратичной формы $z = Y^T A A^T A A^T Y$, где

$$A = \begin{pmatrix} 7 & 6 & 2 & 7 \\ 4 & 9 & 5 & 5 \\ 2 & 3 & 4 & 9 \\ 1 & 5 & 6 & 9 \end{pmatrix}, \quad B = \begin{pmatrix} 3 \\ 2 \\ 0 \\ 2 \end{pmatrix}, \quad Y = \begin{pmatrix} 1 \\ 1 \\ 3 \\ 3 \end{pmatrix}.$$

Вариант 10

Решить системы линейных уравнений $AX = B$, $A^2 A^T A X = B$ и вычислить значение квадратичной формы $z = Y^T A A^T A A^T Y$, где

$$A = \begin{pmatrix} 3 & 6 & 5 & 2 \\ 4 & 6 & 3 & 5 \\ 2 & 3 & 2 & 6 \\ 2 & 4 & 3 & 6 \end{pmatrix}, \quad B = \begin{pmatrix} 3 \\ 0 \\ 4 \\ 3 \end{pmatrix}, \quad Y = \begin{pmatrix} 2 \\ 0 \\ 1 \\ 1 \end{pmatrix}.$$

Глава 4



Финансовый анализ в Excel

Excel предоставляет большой спектр функций финансового анализа: от нахождения платы по процентам, амортизации оборудования, регулярных выплат по займу до оценки эффективности капиталовложений. В данной главе на большом количестве конкретных примеров рассмотрены функции финансового анализа Excel, а также приведены примеры составления финансовых отчетов: от расчета затрат на производство до оптимального планирования рекламной кампании.

4.1. Финансовая функция ППЛАТ

Рассмотрим пример расчета 30-летней ипотечной ссуды со ставкой 8% годовых при начальном взносе 20% и ежемесячной (ежегодной) выплате с помощью функции ППЛАТ (PMT).

Для приведенного на рис. 4.1 ипотечного расчета в ячейки введены формулы, показанные на рис. 4.2.

	A	B	C	D	E	F
1	Расчет ипотечной ссуды					
2						
3	Исходные данные					
4	Цена	201 900р.				
5	Первый взнос	20%				
6	Годовая процентная ставка	8%				
7	Размер ссуды	161 520р.				
8		Ежемесячные выплаты		Ежегодные выплаты		
9	Срок погашения ссуды	360	месяцев	30	лет	
10	Результат расчета					
11	Периодические выплаты	1 185р.		14 347р.		
12	Общая сумма выплат	426 664р.		430 422р.		
13	Общая сумма комиссионных	265 144р.		268 902р.		
14						

Рис. 4.1. Расчет ипотечной ссуды

	В	С	Д
7	=B4*(1-B5)		
8	Ежемесячные выплаты		Ежегодные выплаты
9	=D9*12		30
10			
11	=ППЛАТ(B6/12;D9*12;-B7)		=ППЛАТ(B6;D9;-B7)
12	=B9*B11		=D9*D11
13	=B12-\$B\$7		=D12-\$B\$7
14			
15			

Рис. 4.2. Формулы для расчета ипотечной ссуды

Функция ППЛАТ вычисляет величину постоянной периодической выплаты ренты (например, регулярных платежей по займу) при постоянной процентной ставке.

Синтаксис:

ППЛАТ(ставка; кпер; нз; бз; тип)

Аргументы:

ставка	Процентная ставка за период
кпер	Общее число периодов выплат
нз	Текущее значение, т. е. общая сумма, которую составят будущие платежи
бз	Будущая стоимость или баланс наличности, который нужно достичь после последней выплаты. Если аргумент бз опущен, то он полагается равным 0 (например, будущая стоимость займа равна 0)
тип	Число 0 или 1, обозначающее, когда должна производиться выплата. Если тип равен 0 или опущен, то оплата производится в конце периода, если 1 — то в начале периода

Если бз = 0 и тип = 0, то функция ППЛАТ вычисляет по формуле:

$$P \frac{i(1+i)^n}{(1+i)^n - 1},$$

где P — нз, i — ставка и n — кпер.

Отметим, что очень важно быть последовательным в выборе единиц измерения для задания аргументов ставка и кпер. Например, если вы делаете ежемесячные выплаты по четырехгодичному займу из расчета 12% годовых, то для задания аргумента ставка используйте 12%/12, а для задания аргу-

мента кпер — $4 \cdot 12$. Если вы делаете ежегодные платежи по тому же займу, то для задания аргумента ставка используйте 12%, а для задания аргумента кпер — 4.

Для нахождения общей суммы, выплачиваемой на протяжении интервала выплат, умножьте возвращаемое функцией ППЛАТ значение на величину кпер. Интервал выплат — это последовательность постоянных денежных платежей, осуществляемых за непрерывный период. Например, заем под автомобиль или заклад являются интервалами выплат. В функциях, связанных с интервалами выплат, выплачиваемые вами деньги, такие как депозит на накопление, представляются отрицательным числом, а деньги, которые вы получаете, такие как чеки на дивиденды, представляются положительным числом. Например, депозит в банк на сумму 1000 руб. представляется аргументом -1000 , если вы вкладчик, и аргументом 1000, если вы — представитель банка.

4.2. Пример расчета эффективности неравномерных капиталовложений с помощью функций НПЗ, ВНДОХ и Подбор параметра

Рассмотрим следующую задачу. Вас просят дать в долг 10 000 руб. и обещают вернуть через год 2000 руб., через два года — 4000 руб., через три года — 7000 руб. При какой годовой процентной ставке эта сделка выгодна?

В приводимом на рис. 4.3 расчете в ячейку В8 введена формула

=НПЗ (В7 ; В3 : В5)

	А	В	С
2	Размер ссуды	10 000.00р.	
3	1 год	2 000.00р.	
4	2 год	4 000.00р.	
5	3 год	7 000.00р.	
6	Срок		3 года
7	Годовая учетная ставка		11.79%
8	Чистый текущий объем вклада	10 000.00р.	
9			

Рис. 4.3. Расчет годовой процентной ставки

Кроме того, для автоматизации составления таблицы в ячейку С6 введена формула

=ЕСЛИ (В6=1 ; "год" ; ЕСЛИ (И (В6>=2 ; В6<=4) ; "года" ; "лет"))

Первоначально в ячейку В7 вводится произвольный процент, например 3%. После этого выбираем команду **Сервис, Подбор параметра** (Tools, Goal Seek) и заполняем открывшееся диалоговое окно **Подбор параметра** (Goal Seek), как показано на рис. 4.4.



Рис. 4.4. Диалоговое окно **Подбор параметра** при расчете годовой процентной ставки

В поле **Установить в ячейке** (Set Cell) даем ссылку на ячейку В8, в которой вычисляется чистый текущий объем вклада по формуле

$$=НПЗ(В7;В3:В5)$$

В поле **Значение** (To Value) указываем 10000 — размер ссуды. В поле **Изменяя значение ячейки** (By Changing Cell) даем ссылку на ячейку В7, в которой вычисляется годовая процентная ставка. После нажатия кнопки **ОК** средство подбора параметров определит, при какой годовой процентной ставке чистый текущий объем вклада равен 10000 руб. Результат вычисления выводится в ячейку В7. В нашем случае годовая учетная ставка равна 11,79%. Вывод: если банки предлагают большую годовую процентную ставку, то предлагаемая сделка не выгодна.

Функция **НПЗ** (NPV) возвращает чистый текущий объем вклада, вычисляемый на основе ряда последовательных поступлений наличных и нормы амортизации. Чистый текущий объем вклада — это сегодняшний объем будущих платежей (отрицательные значения) и поступлений (положительные значения). Например, вам предлагают следующую сделку. У вас берут в долг некоторую сумму денег и предлагают через k_1 лет вернуть сумму, равную P_{k1} , через k_2 лет — P_{k2} и т. д. и, наконец, через k_n лет — P_{kn} . Кроме данной сделки, у вас есть альтернативный способ использования ваших денег, например, положить их в банк под i процентов годовых. Тогда чистым текущим объемом вклада является та сумма денег, которой вам нужно располагать в начальный год, чтобы, положив их в банк под i процентов годовых, вы получили предлагаемую прибыль. В нашем случае чистый текущий объем вклада равен

$$\sum_{j=1}^n \frac{P_{k_j}}{(1+i)^{k_j}}.$$

Синтаксис:

НПЗ(ставка; 1-е значение; 2-е значение; ...)

Аргументы:

ставка

Процентная ставка за период

1-е значение,

От 1 до 29 аргументов, представляющих расходы и доходы.

2-е значение, ...

1-е значение, 2-е значение, ... должны быть равномерно распределены по времени и осуществляться в конце каждого периода. НПЗ использует порядок аргументов 1-е значение, 2-е значение, ... для определения порядка поступлений и платежей

Считается, что инвестиция, значение которой вычисляет функция НПЗ, начинается за один период до даты денежного взноса 1-го значения и заканчивается с последним денежным взносом в списке. Вычисления функции НПЗ базируются на будущих денежных взносах. Если первый денежный взнос приходится на начало первого периода, то первое значение следует добавить к результату функции НПЗ, но не включать в список аргументов.

Если n — это количество денежных взносов в списке значений, P_j — j -е значение и i — ставка, то функция НПЗ вычисляется по формуле

$$\sum_{j=1}^n \frac{P_j}{(1+i)^j}.$$

Функция НПЗ связана с функцией ВНДОХ (внутренняя скорость оборота). ВНДОХ — это скорость оборота, для которой НПЗ равняется нулю:

НПЗ(ВНДОХ(...); ...)=0.

Функция ВНДОХ (IRR) возвращает внутреннюю скорость оборота для ряда последовательных операций с наличными, представленными числовыми значениями. Объемы операций не обязаны быть регулярными, как в случае ренты. Внутренняя скорость оборота — это процентная ставка дохода, полученного от инвестиций, состоящих из выплат (отрицательные значения) и поступлений (положительные значения), которые происходят в регулярные периоды времени.

Синтаксис:

ВНДОХ(значения; прогноз)

Аргументы:

значения

Массив или ссылка на ячейки, содержащие числовые величины, для которых вычисляется внутренняя скорость оборота средств.

Значения должны включать по крайней мере одно положительное значение и одно отрицательное значение, для того чтобы можно было вычислить внутреннюю скорость оборота. Функция ВНДОХ использует порядок значений для интерпретации порядка денежных выплат или поступлений, поэтому нужно следить, чтобы значения выплат и поступлений вводились в правильном порядке

прогноз

Величина, о которой предполагается, что она близка к результату ВНДОХ

Если $n + 1$ — количество значений в списке, P_j — j -е значение, то ВНДОХ является корнем относительно i (ставки) следующего уравнения

$$\sum_{j=0}^n \frac{P_j}{(1+i)^j} = 0.$$

Для вычисления ВНДОХ Excel использует метод итераций. Начиная со значения прогноз, функция ВНДОХ выполняет циклические вычисления, пока не получит результат с точностью 0,00001. Если функция ВНДОХ не может получить результат после 20 попыток, возвращается значение ошибки #ЧИСЛО!

В большинстве случаев нет необходимости задавать прогноз для вычислений с помощью функции ВНДОХ. Если прогноз опущен, то он полагается равным 0,1 (10%).

Если ВНДОХ выдает значение ошибки #ЧИСЛО! или результат далек от ожидаемого, можно попытаться выполнить вычисления еще раз, но уже с другим значением аргумента прогноз.

Таким образом, задачу, сформулированную в самом начале данного раздела, можно решить также с помощью функции ВНДОХ. Для этого в ячейку В2 вместо 10000р. надо ввести -10000р., а в ячейку В7 — функцию ВНДОХ(В2:В5), которая и найдет минимальную годовую учетную ставку.

4.3. Пример расчета эффективности капиталовложений с помощью функции ПЗ

Рассмотрим следующую задачу. Допустим, что у вас просят в долг 10000 руб. и обещают возвращать по 2000 руб. в течение 6 лет. Будет ли выгодна эта сделка при годовой ставке 7%?

В приводимом на рис. 4.5 расчете в ячейку В5 введена формула

=ПЗ(В4;В2;-В3)

Кроме того, для автоматизации составления таблицы в ячейки С2 и В6 введены формулы

	A	B	C	D	E
1	Размер ссуды	10 000р.			
2	Срок	6 лет			
3	Ежегодно возвращаемые деньги	2 000р.			
4	Годовая процентная ставка	7%			
5	Чистый текущий объем вклада	9 533р.			
6	Вывод	Выгоднее деньги положить под проценты			
7					

Рис. 4.5. Расчет эффективности капиталовложений

=ЕСЛИ (B2=1; "год"; ЕСЛИ (И (B2>=2; B2<=4) ; "года"; "лет"))

=ЕСЛИ (B1<B5; "Выгодно дать деньги в долг"; ЕСЛИ (B5=B1;

"Варианты равносильны"; "Выгоднее деньги положить под проценты"))

Функция ПЗ (PV) возвращает текущий объем вклада на основе постоянных периодических платежей. Функция ПЗ аналогична функции НПЗ. Основное различие между ними заключается в том, что функция ПЗ допускает, чтобы денежные взносы происходили либо в конце, либо в начале периода. Кроме того, в отличие от функции НПЗ, денежные взносы в функции ПЗ должны быть постоянными на весь период инвестиции.

Синтаксис:

ПЗ (ставка; кпер; выплата; бз; тип)

Аргументы:

ставка	Процентная ставка за период
кпер	Общее число периодов выплат
выплата	Величина постоянных периодических платежей
бз	Будущая стоимость или баланс наличности, который нужно достичь после последней выплаты. Если аргумент бз опущен, он полагается равным 0 (например, будущая стоимость займа равна 0)
тип	Число 0 или 1, обозначающее, когда должна производиться выплата. Если тип равен 0 или опущен, то оплата производится в конце периода, если 1 — то в начале периода

Если тип = 0 и бз = 0, то функция ПЗ вычисляется по следующей формуле

$$A \frac{(1+i)^n - 1}{i(1+i)^n},$$

где A — выплата, i — ставка, n — кпер.

В данном разделе была рассмотрена задача с двумя результирующими функциями: числовой — чистым текущим объемом вклада и качественной, оценивающей, выгодна ли сделка. Эти функции зависят от нескольких параметров. Некоторыми из них вы можете управлять, например, сроком и суммой ежегодно возвращаемых денег. Часто бывает удобно проанализировать ситуацию для нескольких возможных вариантов параметров. Команда **Сервис, Сценарии** (Tools, Scenarios) предоставляет такую возможность с одновременным автоматизированным составлением отчета. Рассмотрим способ применения этой команды для следующих трех комбинаций срока и суммы ежегодно возвращаемых денег: 6, 2000; 12, 1500 и 7, 1500.

Выберем команду **Сервис, Сценарии** (Tools, Scenarios). В открывшемся диалоговом окне **Диспетчер сценариев** (Scenarios Manager) для создания первого сценария нажмите кнопку **Добавить** (Add) (рис. 4.6).

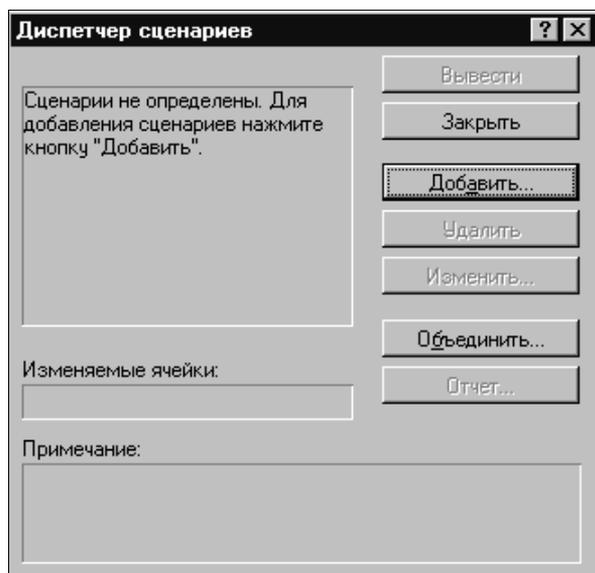


Рис. 4.6. Диалоговое окно **Диспетчер сценариев**

В диалоговом окне **Добавление сценария** (Add Scenario) в поле **Название сценария** (Scenario Name) введите, например пз 1, а в поле **Изменяемые ячейки** (Changing Cells) — ссылку на ячейки B2 и B3, в которые вводятся значения параметров задачи (срок и сумма ежегодно возвращаемых денег) (рис. 4.7).

После нажатия кнопки **ОК** появится диалоговое окно **Значения ячеек сценария** (Scenario Values), в поля которого введите значения параметров для первого сценария (рис. 4.8).

С помощью кнопки **Добавить** (Add) последовательно создайте нужное число сценариев. После этого диалоговое окно **Диспетчер сценариев** (Scenarios Manager) будет иметь вид, показанный на рис. 4.9.

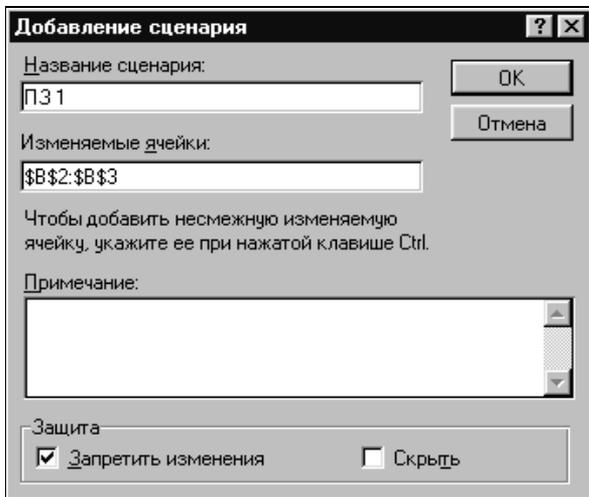


Рис. 4.7. Диалоговое окно **Добавление сценария**

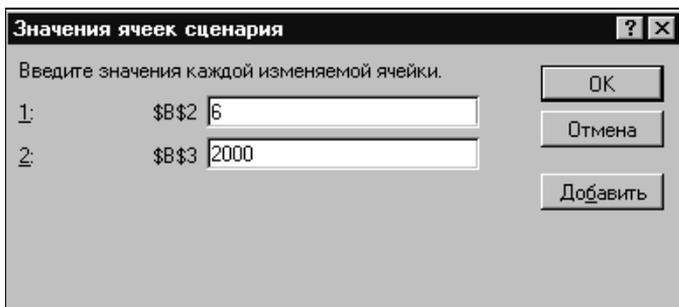


Рис. 4.8. Диалоговое окно **Значения ячеек сценария**

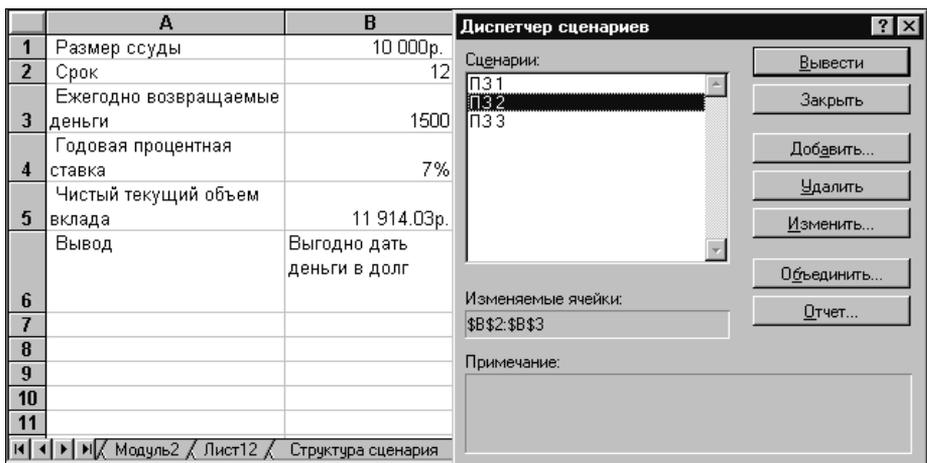


Рис. 4.9. Вывод сценариев на рабочий лист с помощью диалогового окна **Диспетчер сценариев**

С помощью кнопки **Вывести** (Show) можно вывести результаты, соответствующие выбранному сценарию. Нажатие кнопки **Отчет** (Summary) открывает диалоговое окно **Отчет по сценарию** (Scenario Summary) (рис. 4.10).

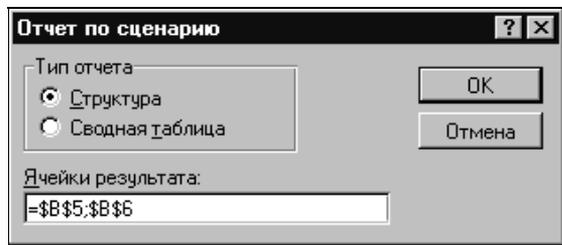


Рис. 4.10. Диалоговое окно **Отчет по сценарию**

В этом окне в группе **Тип отчета** (Scenario Type) необходимо установить переключатель в положение **Структура** (Scenario Summary) или **Сводная таблица** (Scenario Pivot Table), а в поле **Ячейки результата** (Result Cells) дать ссылку на ячейки, где вычисляются значения результирующих функций. После нажатия кнопки **ОК** создается отчет. На рис. 4.11 показан отчет по сценариям типа **Структура** (Scenario Summary).

	В	С	Д	Е	Ф	Г
1						
2	Структура сценария					
3	Текущие значения:			ПЗ 1	ПЗ 2	ПЗ 3
5	Изменяемые ячейки:					
6	\$B\$2	6		6	12	7
7	\$B\$3	2000		2000	1500	1500
8	Ячейки результата:					
9	\$B\$5	9 533.08р.		9 533.08р.	11 914.03р.	8 083.93р.
10	\$B\$6	Выгоднее деньги положить под проценты	Выгоднее деньги положить под проценты	Выгодно дать деньги в долг	Выгоднее деньги положить под проценты	Выгоднее деньги положить под проценты
11	Примечания: столбец "Текущие значения" представляет значения изменяемых ячеек в момент создания Итогового отчета по Сценарию. Изменяемые ячейки для каждого сценария выделены серым цветом.					
12						
13						
14						
15						
16						
17						
18						
19						
20						

Рис. 4.11. Отчет по сценарию типа **Структура**

4.4. Финансовые функции ПЛПРОЦ и ОСНПЛАТ

Рассмотрим пример вычисления основных платежей, платы по процентам, общей ежегодной платы и остатка долга на примере ссуды 100 000 руб. на срок 5 лет при годовой ставке 2% (рис. 4.12).

	А	В	С	Д	Е
1	Процент	2%			
2	Срок	5 лет			
3	Ежегодная плата	21 216р.			
4	Размер ссуды	100 000р.			
5	Год	Плата по процентам	Основная плата	Остаток долга	
6	0			100 000р.	
7	1	2 000р.	19 216р.	80 784р.	
8	2	1 616р.	19 600р.	61 184р.	
9	3	1 224р.	19 992р.	41 192р.	
10	4	824р.	20 392р.	20 800р.	
11	5	416р.	20 800р.	0р.	
12					

Рис. 4.12. Вычисление основных платежей и платы по процентам

Ежегодная плата вычисляется в ячейке В3 по формуле

=ПЛПРОЦ(процент; срок; -размер_ссуды),

где ячейки В1, В2 и В4 имеют имена: процент, срок и размер_ссуды, соответственно. Присвоение имени ячейке осуществляется с помощью команды **Вставка, Имя, Присвоить** (Insert, Name, Define). За первый год плата по процентам в ячейке В7 вычисляется по формуле

=D6*процент

Основная плата в ячейке С7 вычисляется по формуле

=ежегодная_плата-В7

где ежегодная_плата — имя ячейки В3. Остаток долга в ячейке D7 вычисляется по формуле

=D6-С7

В оставшиеся годы эти платы определяются с помощью протаскивания маркера заполнения выделенного диапазона В7:D7 вниз по столбцам. Отметим, что основную плату и плату по процентам можно было непосредственно найти с помощью функций ОСНПЛАТ (PPMT) и ПЛПРОЦ (IPMT), соответственно.

Функция ПЛПРОЦ возвращает платежи по процентам за данный период на основе периодических постоянных выплат и постоянной процентной ставки.

Синтаксис:

ПЛПРОЦ(ставка; период; кпер; нз; бз; тип)

Функция ОСНПЛАТ возвращает величину выплаты за данный период на основе периодических постоянных платежей и постоянной процентной ставки.

Синтаксис:

ОСНПЛАТ(ставка; период; кпер; нз; бз; тип)

Аргументы функций ПЛПРОЦ и ОСНПЛАТ:

период	Период, за который требуется найти прибыль (должен находиться в интервале от 1 до кпер)
ставка	Процентная ставка за период
кпер	Общее число периодов выплат
нз	Текущее значение, т. е. общая сумма, которую составят будущие платежи
бз	Будущая стоимость или баланс наличности, который нужно достичь после последней выплаты. Если аргумент бз опущен, он полагается равным 0 (например, будущая стоимость займа равна 0)
тип	Число 0 или 1, обозначающее, когда должна производиться выплата. Если тип равен 0 или опущен, то оплата производится в конце периода, если 1 — то в начале периода

Функции ПЛПРОЦ и ОСНПЛАТ тесно связаны между собой, а именно

$$\text{ПЛП}_j = i B_{j-1}, \text{ОСНП}_j = A - \text{ПЛП}_j, B_j = B_{j-1} - \text{ОСНП}_j \text{ при } j \in [0, n],$$

где j — номер периода, n — кпер, ПЛП_j , ОСНП_j и B_j — это ПЛПРОЦ, ОСНПЛАТ и остаток долга, соответственно, за j -й период, $\text{ПЛП}_0 = 0$, $\text{ОСНП}_0 = 0$, B_0 — нз, A — величина выплаты за один период годовой ренты на основе постоянных выплат и постоянной процентной ставки, вычисляемая с помощью функции ПЛЛАТ.

4.5. Финансовые функции БЗ, КПЕР и НОРМА

Функция ВЗ (FV) вычисляет будущее значение вклада на основе периодических постоянных платежей и постоянной процентной ставки. Функция ВЗ

подходит для расчета итогов накоплений при ежемесячных банковских взносах.

Синтаксис:

ВЗ (ставка; кпер; выплата; нз; тип)

Аргументы:

ставка	Процентная ставка за период
кпер	Общее число периодов выплат
выплата	Величина постоянных периодических платежей
нз	Текущее значение, т. е. общая сумма, которую составят будущие платежи
тип	Число 0 или 1, обозначающее, когда должна производиться выплата. Если тип равен 0 или опущен, то оплата производится в конце периода, если 1 — в начале периода

Если тип = 0 и нз = 0, то функция ВЗ вычисляется по следующей формуле

$$A \frac{(1+i)^n - 1}{i},$$

где A — выплата, i — ставка, n — кпер.

Приведем пример использования функции ВЗ. Предположим, вы хотите резервировать деньги для специального проекта, который будет осуществлен через год. Предположим, вы собираетесь вложить 1000 руб. при годовой ставке 6%. Вы собираетесь вкладывать по 100 руб. в начале каждого месяца в течение года. Сколько денег будет на счете в конце 12 месяцев?

С помощью формулы

=ВЗ(6%/12; 12; -100; -1000; 1)

получаем ответ: 2 301.40р.

Функция КПЕР (NPER) вычисляет общее количество периодов выплаты для данного вклада на основе периодических постоянных выплат и постоянной процентной ставки.

Синтаксис:

КПЕР (ставка; выплата; нз; бз; тип)

Аргументы:

ставка	Процентная ставка за период
выплата	Величина постоянных периодических платежей

нз	Текущее значение, т. е. общая сумма, которую составят будущие платежи
бз	Будущая стоимость или баланс наличности, который нужно достичь после последней выплаты. Если аргумент бз опущен, он полагается равным 0 (например, будущая стоимость займа равна 0)
тип	Число 0 или 1, обозначающее, когда должна производиться выплата. Если тип равен 0 или опущен, то оплата производится в конце периода, если 1 — то в начале периода

Если тип = 0 и бз = 0 функция КПЕР вычисляется по следующей формуле

$$\log_{1+i} \left(\frac{Pi}{A} + 1 \right),$$

где P — нз, i — ставка, A — выплата.

Например, если вы берете в долг 1 000 руб. при годовой ставке 1% и собираетесь выплачивать по 100 руб. в год, то число выплат вычисляется следующим образом:

=КПЕР(1%; -100; 1000)

В результате получаем ответ: 11.

Функция НОРМА (RATE) вычисляет процентную ставку за один период, необходимую для получения определенной суммы в течение заданного срока путем постоянных взносов. Следует отметить, что функция НОРМА вычисляет процентную ставку методом итераций, поэтому решение может быть и не найдено. Если после 20 итераций погрешность определения ставки превышает 0,000 000 1, то функция НОРМА возвращает значение ошибки #ЧИСЛО!.

Синтаксис:

НОРМА(кпер; выплата; нз; бз; тип; нач_прибл)

Аргументы:

кпер	Общее число периодов выплат
выплата	Величина постоянных периодических платежей
нз	Текущее значение, т. е. общая сумма, которую составят будущие платежи
бз	Будущая стоимость или баланс наличности, который нужно достичь после последней выплаты. Если аргумент бз опущен, он полагается равным 0 (например, будущая стоимость займа равна 0)

тип	Число 0 или 1, обозначающее, когда должна производиться выплата. Если тип равен 0 или опущен, то оплата производится в конце периода, если 1 — то в начале периода
нач_прибл	Предполагаемая величина нормы. Если нач_прибл опущено, то оно полагается равным 10%. Если функция НОРМА не сходится, следует попытаться использовать различные значения нач_прибл. Обычно функция НОРМА сходится, если нач_прибл имеет значение между 0 и 1

Если бз = 0 и тип = 0, функция НОРМА является корнем следующего уравнения

$$P = A \frac{(1+i)^n - 1}{i(1+i)^n},$$

где A — выплата, i — НОРМА, n — кпер, P — нз.

Рассмотрим пример использования функции НОРМА. Чтобы определить процентную ставку для четырехлетнего займа размером в 8000 руб. с ежемесячной выплатой 200 руб., можно использовать формулу

=НОРМА(48; -200; 8000)

В результате получаем: месячная (т. к. период равен месяцу) процентная ставка равна 0,77%.

4.6. Функции для расчета амортизации АМР, АМГД, ДОБ и ДДОБ

Под амортизацией подразумевается уменьшение (обычно — на единицу времени) стоимости имущества в процессе эксплуатации.

Функция АМР (SLN) возвращает величину амортизации имущества за один период времени, используя метод равномерной амортизации.

Синтаксис:

АМР(стоимость; остаток; время_эксплуатации)

Аргументы:

стоимость	Начальная стоимость имущества
остаток	Остаточная стоимость в конце амортизации (иногда называется ликвидной стоимостью имущества)
время_эксплуатации	Количество периодов, за которые собственность амортизируется (иногда называется периодом амортизации)

Рассмотрим на конкретном примере применение всех функций, описываемых в данном разделе.

Предположим вы купили за 6 000 руб. компьютер, который имеет срок эксплуатации 5 лет, после чего оценивается в 1 000 руб.

Снижение стоимости компьютера для каждого года эксплуатации вычисляется по формуле

```
=АМР(6000; 1000; 5)
```

В результате получаем: 1 000р.

Функция АМГД (SYD) возвращает годовую амортизацию имущества для указанного периода.

Синтаксис:

```
АМГД(стоимость; остаток; время_эксплуатации; период)
```

Аргументы:

стоимость	Начальная стоимость имущества
остаток	Остаточная стоимость в конце амортизации (иногда называется ликвидной стоимостью имущества)
время_эксплуатации	Количество периодов, за которые собственность амортизируется (иногда называется периодом амортизации)
период	Номер периода для вычисления амортизации (должен быть измерен в тех же единицах, что и время_эксплуатации)

Возвращаясь к предыдущему примеру, можно рассчитать амортизацию, например, за первый год эксплуатации компьютера по формуле

```
=АМГД(6000;1000;5;1)
```

и за последний год по формуле

```
=АМГД(6000;1000;5;5)
```

В результате получим: 1666.67р. и 333.33р., соответственно.

Функция ДОВ (DB) возвращает амортизацию имущества за заданный период, используя метод постоянного учета амортизации.

Синтаксис:

```
ДОВ(стоимость; остаток; время_эксплуатации; период; месяц)
```

Аргументы:

стоимость	Начальная стоимость имущества
остаток	Остаточная стоимость в конце амортизации (иногда называется ликвидной стоимостью имущества)
время_эксплуатации	Количество периодов, за которые собственность амортизируется (иногда называется периодом амортизации)
период	Номер периода для вычисления амортизации (измеряется в тех же единицах, что и время_эксплуатации)
месяц	Количество месяцев в первом году. Если аргумент месяц опущен, то предполагается, что он равен 12

Метод постоянного учета амортизации вычисляет амортизацию, используя фиксированную процентную ставку.

При расчете нашего примера получим амортизацию за время эксплуатации компьютера:

=ДОВ(6000;1000;5;1) вычисляет 1806.00р.

=ДОВ(6000;1000;5;2) вычисляет 1262.394р.

=ДОВ(6000;1000;5;3) вычисляет 882.41р.

=ДОВ(6000;1000;5;4) вычисляет 616.81р.

=ДОВ(6000;1000;5;5) вычисляет 431.15р.

Функция ДДОВ (DDV) возвращает величину амортизации имущества для указанного периода, используя метод двукратного (или k -кратного) учета амортизации.

Синтаксис:

ДДОВ(стоимость; остаток; время_эксплуатации; период; коэффициент)

Аргументы:

стоимость	Начальная стоимость имущества
остаток	Остаточная стоимость в конце амортизации (иногда называется ликвидной стоимостью имущества)
время_эксплуатации	Количество периодов, за которые собственность амортизируется (иногда называется периодом амортизации)
период	Номер периода для вычисления амортизации (должен быть измерен в тех же единицах, что и время_эксплуатации)

вдоль столбца (строки) до тех пор, пока не будет создан требуемый ряд. На вкладке **Списки** (Custom Lists) диалогового окна **Параметры** (Options) (рис. 4.14), открываемого командой **Сервис, Параметры** (Tools, Options), приведены встроенные в Excel списки, которые представляют собой последовательности названий месяцев и дней недели.

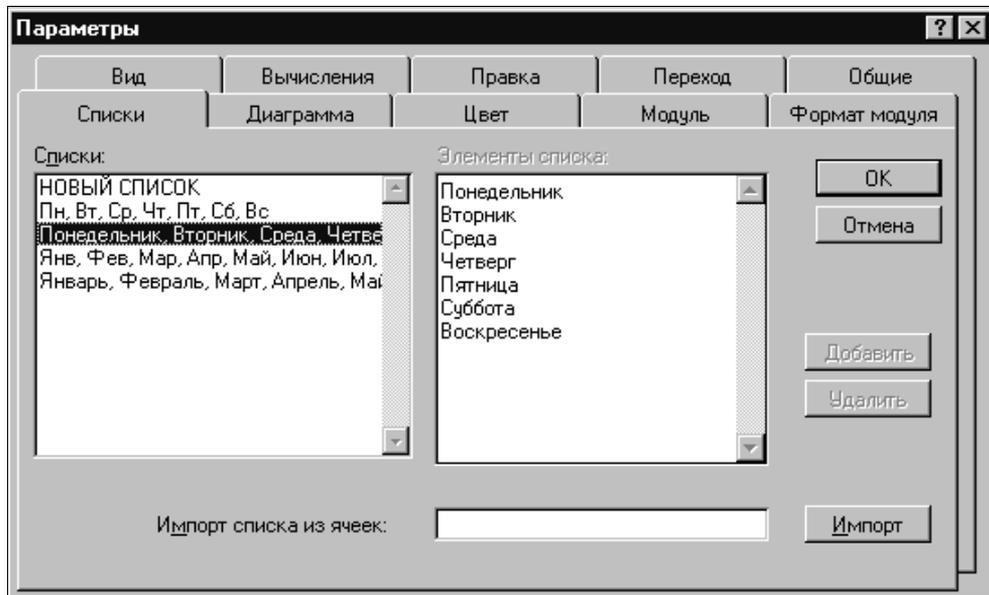


Рис. 4.14. Вкладка **Списки** диалогового окна **Параметры**

Используя вкладку **Списки** (Custom Lists), можно создавать пользовательские списки. Элементы списка пользователя надо ввести в поле **Элементы списка** (List Entries), причем каждый элемент вводится с новой строки. Если нажать кнопку **Добавить** (Add), то созданный список будет занесен в библиотеку списков. Список можно также добавить и непосредственно с рабочего листа, указав в поле **Импорт списка из ячеек** (Import List from Cells) диапазон, из которого импортируется список.

Кроме стандартных списков, занесенных в библиотеку, Excel позволяет легко создавать по приведенному выше алгоритму последовательности с текстом и порядковыми номерами (рис. 4.13, столбцы E, F и H). Если номера меняются с шагом, отличным от единицы, необходимо в две соседние ячейки ввести первые два члена последовательности, например, экспо 87 в G1 и экспо 89 в G2, затем выделить диапазон G1:G2, установить указатель мыши на маркер заполнения диапазона и протащить его вдоль столбца до тех пор, пока не будет создан требуемый ряд.

Перед началом создания первой отчетной ведомости пополним наши знания, рассмотрев синтаксис трех функций, наиболее часто встречающихся при расчетах в таблицах: СУММ, СРЗНАЧ и РАНГ.

Функция СУММ (SUM) находит сумму чисел из указанного диапазона ячеек.

Синтаксис:

СУММ(число1; число2; ...)

где число1, число2, ... — числа, которые суммируются.

Функция СРЗНАЧ (AVERAGE) находит среднее значение чисел из указанного диапазона ячеек.

Синтаксис:

СРЗНАЧ(число1; число2; ...)

Аргументы — те же, что и у функции СУММ.

Функция РАНГ (RANK) возвращает ранг числа в списке чисел. Ранг числа — это его величина относительно других значений в списке. (Если список отсортировать, то ранг числа будет его позицией).

Синтаксис:

РАНГ(число; ссылка; порядок)

Аргументы:

число	Число, для которого определяется ранг
ссылка	Массив или ссылка на список чисел. Нечисловые значения в ссылке игнорируются
порядок	Число, определяющее способ упорядочения. Если порядок равен 0 или опущен, то Excel определяет ранг числа так, как если бы ссылка была списком, отсортированным в порядке убывания. Если порядок — любое ненулевое число, то Excel определяет ранг числа так, как если бы ссылка была списком, отсортированным в порядке возрастания

Отметим, что функция РАНГ присваивает повторяющимся числам одинаковый ранг.

Вооружившись навыками работы со списками и знаниями о функциях СУММ, СРЗНАЧ и РАНГ перейдем к созданию двух отчетных таблиц.

Первый пример. Сначала обсудим, как создать отчетную ведомость о результатах работы сети магазинов, приведенную на рис. 4.15.



Рис. 4.15. Отчетная ведомость о работе сети магазинов за июнь — август

В ячейку E4 введем формулу

=СУММ(B4:D4)

которую с помощью маркера заполнения протащим на диапазон E4:E9.

В ячейку B10 введем формулу

=СУММ(B4:B9)

которую протащим на диапазон B10:E10.

В ячейку G4 введем формулу

=СРЗНАЧ(B4:D4)

которую протащим на диапазон G4:G9.

В ячейку H4 введем формулу

=E4/\$E\$10

которую протащим на диапазон H4:H9. После чего диапазону ячеек H4:H9 назначим формат с помощью кнопки **%**.

Заметим, что знак \$, стоящий перед буквой в имени ячейки, дает абсолютную ссылку на столбец с данным именем, а знак \$, стоящий перед цифрой в имени ячейки, дает абсолютную ссылку на строку с этим именем. Поэтому если в формуле буква, входящая в имя ячейки, окружена с двух сторон знаками \$, это означает, что в формуле дается не относительный, а абсолютный адрес ячейки, т. е. адрес, не подлежащий изменению при протаскивании формулы.

Для ввода в формулу абсолютного адреса ячейки достаточно после ввода ее относительного адреса нажать клавишу <F4>. Если бы в ячейку Н4 была введена формула =Е4/Е10, то ее протаскивание на ячейки Н5:Н9 привело бы к неверному результату. Присвоение ячейке имени с помощью команды **Вставка, Имя, Присвоить** (Inset, Name, Define) дает другой способ абсолютной адресации ячейки. Например, если бы ячейке Е10 было присвоено имя Итого, то в ячейку Н4 можно было бы ввести формулу

```
=Е4/Итого
```

которую затем протаскиваем на диапазон Н4:Н9.

Для нахождения места магазина по объему продаж введем в ячейку F4 формулу

```
=РАНГ(Е4; $Е$4: $Е$9)
```

которую протаскиваем на диапазон F4:F9.

С помощью функции ЧАСТОТА (FREQUENCY) подсчитаем для данного множества суммарных выручек магазинов, сколько значений попадает в интервалы от 0 до 1000, от 1001 до 1100, от 1101 до 1200 и свыше 1201 млн. руб. С этой целью в диапазон ячеек I4:I6 введем верхние границы этих интервалов 1000, 1100 и 1200, соответственно, а в диапазон ячеек J4:J7 введем формулу

```
{=ЧАСТОТА(Е4:Е9; I4: I6)}
```

Данная формула выведет в ячейку J4, сколько значений находится в интервале от 0 до 1000, в ячейку J5 — от 1001 до 1100, в ячейку J6 — от 1101 до 1200, в ячейку J7 — сколько значений будет не меньше 1201.

Функция ЧАСТОТА возвращает распределение частот в виде вертикального массива. Для данного множества значений и данного множества карманов (интервалов, в математическом смысле) частотное распределение подсчитывает, сколько исходных значений попадает в каждый интервал.

Синтаксис:

```
ЧАСТОТА(массив_данных; массив_карманов)
```

Аргументы:

массив_данных

Массив или ссылка на множество данных, для которых вычисляются частоты; если массив_данных не содержит значений, то функция ЧАСТОТА возвращает массив нулей

массив_карманов

Массив или ссылка на множество интервалов, в которые группируются значения аргумента массив_данных; если массив_карманов не содержит значений, то функция ЧАСТОТА возвращает количество элементов в аргументе массив_данных

Частоты можно также вычислить, воспользовавшись диалоговым окном **Анализ данных** (Data Analysis), которое открывается командой **Сервис, Анализ данных** (Tools, Data Analysis). Средство анализа данных является одной из надстроек Excel. Если в меню **Сервис** (Tools) отсутствует команда **Анализ данных** (Data Analysis), то для ее установки нужно выполнить команду **Сервис, Надстройки, Analysis ToolPak** (Tools, Add-ins, Analysis ToolPak).

После выбора пункта **Гистограмма** (Histogram) в диалоговом окне **Анализ данных** (Data Analysis) откроется диалоговое окно **Гистограмма** (Histogram) (рис. 4.16).

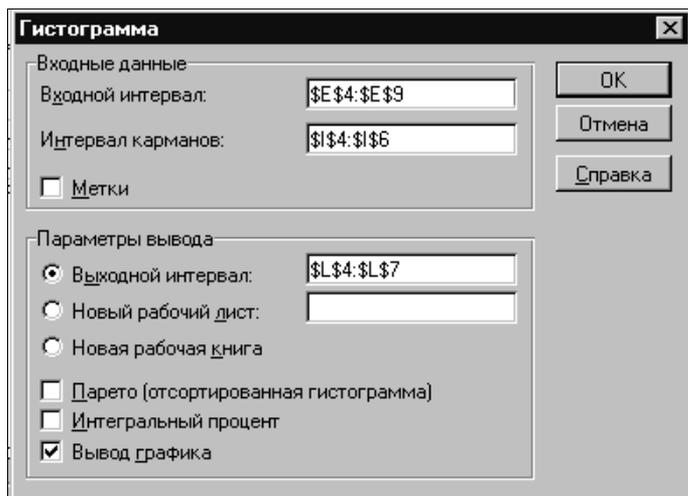


Рис. 4.16.
Диалоговое окно **Гис-
тограмма**

В поле **Входной интервал** (Input Range) введем диапазон E4:E9, по которому строим гистограмму. В поле **Интервал карманов** (Bin Range) введем диапазон I4:I6 со значениями верхних границ интервалов. Гистограмма строится на новом рабочем листе или на текущем листе с указанием диапазона ячеек для результата. В данном случае в поле ввода **Выходной интервал** (Output Range) введем диапазон L4:L7. На рис. 4.17 приведен результат построения гистограммы.

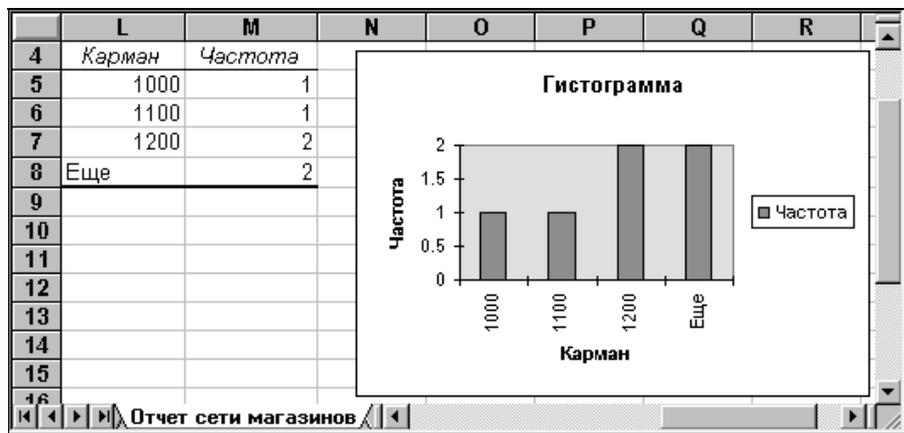


Рис. 4.17. Результат построения гистограммы

Второй пример. Рассмотрим еще один пример составления отчетной ведомости, в которой по объему реализованных товаров рассчитывается итоговая выручка (рис. 4.18).

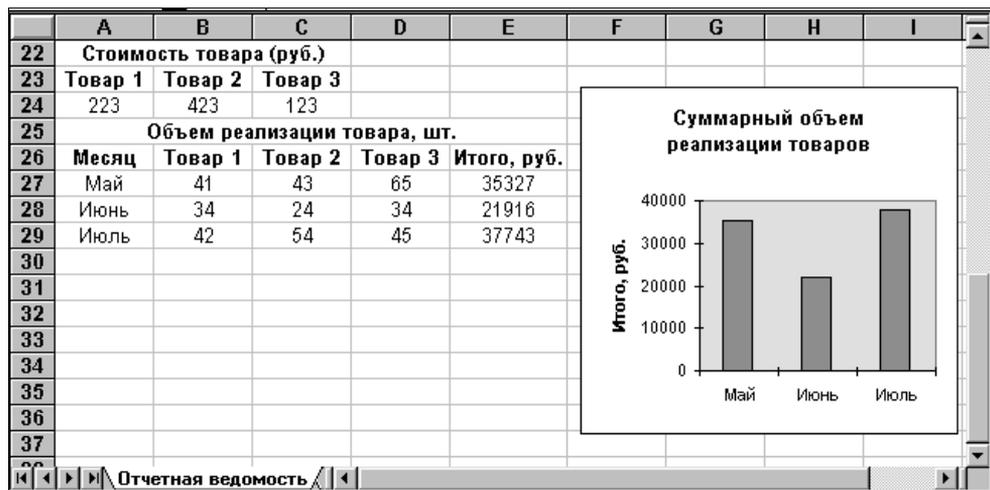


Рис. 4.18. Расчет итоговой выручки по объему реализации

В ячейки A24:C24 введены стоимости трех различных товаров, а в ячейки B27:D29 — объемы их реализации по месяцам. Для того чтобы вычислить суммарную стоимость реализованных товаров по месяцам, введем в ячейки E27:E29 формулу

```
{=МУМНОЖ(B27:D29;ТРАНСП(A24:C24)}
```

Отметим, что данную таблицу можно было заполнить и без привлечения матричных формул. Можно ввести в ячейку E27 формулу

=СУММПРОИЗВ(B27:D27;\$A\$24:\$C\$24)

и протянуть ее на диапазон E27:E29. Функция СУММПРОИЗВ (SUMPRODUCT) вычисляет сумму произведений элементов указанных диапазонов ячеек.

При построении гистограммы (рис. 4.18) в поле ввода первого диалогового окна **Мастер диаграмм** (Cart Wizard) введите диапазоны A27:A29; E27:E29. Напоминаем, что для одновременного выделения диапазонов, которые не примыкают друг к другу, сначала необходимо выделить первый диапазон, а потом при нажатой клавише <Ctrl> — второй.

4.8. Пример отчетной ведомости по расчету просроченных платежей

Рассмотрим пример составления отчетной ведомости фирмы, продающей компьютеры, позволяющей определить количество и сумму просроченных клиентами платежей (рис. 4.19).

	A	B	C	D	E	F	G	H
1	Название CPU	Цена	Дата продажи	Дата оплаты	Просрочка на дней	Дата переучета	Количество просроченных заказов	Срок просрочки, дней
2	AMD K5-100	67	12.06.98	12.06.98		31.07.98	3	до 29
3	Pentium I 233	98	12.06.98		49		2	от 30 до 39
4	Pentium II 266	209	14.06.98	14.06.98			4	свыше 40
5	Pentium II 233	182	14.06.98	20.06.98				
6	Pentium II 300	315	14.06.98		47			
7	Pentium II 333	403	17.06.98	17.06.98			Стоимость просроченных заказов	
8	Pentium II 266	209	18.06.98	21.07.98			945	до 29
9	Pentium II 266	209	19.06.98	19.07.98			276	от 30 до 39
10	AMD K5-100	67	20.06.98		41		2874	свыше 40
11	AMD K5-100	67	20.06.98		41			
12	Pentium II 266	209	20.06.98	20.06.98				
13	Pentium II 266	209	23.06.98		38			
14	AMD K5-100	67	24.06.98		37			
15	Pentium II 300	315	25.06.98	28.06.98				
16	Pentium II 266	209	25.06.98	25.06.98				
17	Pentium II 300	315	02.07.98		29			
18	Pentium II 300	315	02.07.98		29			
19	Pentium II 300	315	02.07.98	02.07.98				
20	Pentium II 300	315	03.07.98		28			

Рис. 4.19. Расчет просроченных платежей

Дата переучета введена в ячейку F2 с помощью формулы

```
=ДАТА(98;7;31)
```

Функция ДАТА (DATE) возвращает дату в числовом формате.

Синтаксис:

```
ДАТА(год; месяц; день)
```

Аргументы:

год	Число от 1900 до 2078
месяц	Число, представляющее номер месяца в году. Если оно больше 12, то прибавляется к первому месяцу указанного года. Например, ДАТА(96;14;2) возвращает числовой формат даты 2 февраля 1997 года
день	Число, представляющее номер дня в месяце. Если оно больше числа дней в указанном месяце, то прибавляется к первому дню указанного месяца

В ячейку E2 введена формула, определяющая срок просрочки

```
=ЕСЛИ(D2=0;$F$2-C2;"")
```

которая протаскивается на диапазон E3:E20.

В ячейки G8, G9 и G10 введены следующие формулы:

```
{=СУММ((E2:E20>0)*(E2:E20<=29)*(B2:B20))}
```

```
{=СУММ((E2:E20>=30)*(E2:E20<=39)*(B2:B20))}
```

```
{=СУММ((E2:E20>=40)*(B2:B20))}
```

вычисляющие суммарные стоимости просроченных оплат сроком до 29 дней, от 30 до 39 дней и свыше 40 дней.

Дадим пояснения к третьей из этих формул. Excel в формуле массива возвращает условие (E2:E20>=40) в виде массива, состоящего из 0 и 1, где 0 стоит на месте ячейки со значением меньше 40 и 1 — на месте ячейки со значением не меньше 40. Следовательно, данная формула вычисляет сумму произведений элементов массива (E2:E20>=40) (с единицами в случае просрочки на указанный срок и нулями — в противном случае) и массива B2:B20 (с ценами процессоров). Таким образом, третья формула возвращает суммарную стоимость заказов, просроченных не менее чем на 40 дней.

В ячейки G2, G3 и G4 введены формулы:

```
{=СУММ((E2:E20>0)*(E2:E20<=29))}
```

```
{=СУММ( (E2:E20>=30) * (E2:E20<40) ) }
=СЧЁТЕСЛИ(E2:E20;">=40")
```

вычисляющие количество просроченных оплат сроком до 29 дней, от 30 до 39 дней и свыше 40 дней.

Функция СЧЁТЕСЛИ (COUNTIF) возвращает количество ячеек внутри указанного интервала, удовлетворяющих заданному критерию.

Синтаксис:

СЧЁТЕСЛИ(интервал; критерий)

Аргументы:

интервал	Интервал, в котором нужно подсчитать ячейки
критерий	Критерий в форме числа, выражения или текста, который определяет, какие ячейки надо подсчитывать (например, критерий может быть выражен следующим образом: 17, "17", ">17", "Компьютер")

4.9. Пример отчетной ведомости по расчету затрат на производство

Рассмотрим пример составления отчетной ведомости по расчету затрат на производство товара (рис. 4.20).

	А	В	С	Д	Е	Ф
1	Заказ, шт.	4500		Диски, шт.	Оплата, руб./шт.	Оплата, руб.
2	Продовжная цена, руб.	10		0	0.3	300
3	Стоимость упаковки, руб./шт.	1		1000	0.4	400
4	Стоимость материала, руб./шт.	4		2000	0.5	500
5				3000	0.6	600
6	Стоимость упаковки	4500		4000	0.7	350
7	Стоимость материала	18000		5000	0.8	0
8	Зарплата	2150				
9	Общие издержки	24650				
10	Прибыль	20350				
11						
12						
13						
14						
15						
16						

Рис. 4.20. Расчет затрат на производство товара

Предположим, что фирма производит CD-диски. Упаковка диска обходится фирме в 1 руб./шт., стоимость материалов — 4 руб./шт. Готовые диски фирма продает по цене 10 руб./шт. Технические возможности фирмы позволяют выпускать до 5 тысяч дисков в день. Оплата труда рабочих является сдельной и зависит от количества выпущенных дисков. За первую тысячу дисков оплата труда рабочих составляет 0,3 руб./шт., за вторую тысячу дисков — 0,4 руб./шт., за третью тысячу дисков — 0,5 руб./шт., за четвертую тысячу дисков — 0,6 руб./шт. и свыше 4000 дисков — 0,7 руб./шт.

Фирме поступил заказ на изготовление 4500 CD-дисков. Необходимо подсчитать суммарные издержки и прибыль от выполнения данного заказа.

Для упрощения чтения формул присвоим с помощью команды **Вставка, Имя, Присвоить** (Insert, Name, Define) диапазонам D2:D7, E2:E7, F2:F7 и ячейке B1, соответственно, имена:

ДискиШт

ОплатаРубШт

ОплатаРуб

ЗаказШт

Зарплата рабочих, в зависимости от объема выпущенных дисков, находится в диапазоне F2:F7 по формуле

```
=ЕСЛИ(ЗаказШт-1000>ДискиШт;1000*ОплатаРубШт;
```

```
ЕСЛИ(ЗаказШт>ДискиШт;(ЗаказШт-ДискиШт)*ОплатаРубШт;0))
```

Заметим, что имя диапазона или ячейки удобнее вводить в формулу из диалогового окна **Вставка имени** (Paste Name), которое открывается командой **Вставка, Имя, Вставить** (Insert, Name, Paste), что помогает избежать ошибок при вводе с клавиатуры (рис. 4.21).

Стоимость упаковки и материалов вычисляются в ячейках B6 и B7 по формулам

```
=B1*B3
```

```
=B1*B4
```

Зарплата, общие издержки и прибыль вычисляются в ячейках B8, B9 и B10 по формулам

```
=СУММ(ОплатаРуб)
```

```
=СУММ(B6:B8)
```

```
=B1*B2-B9
```

Расчет прибыли и затрат на производство закончен.

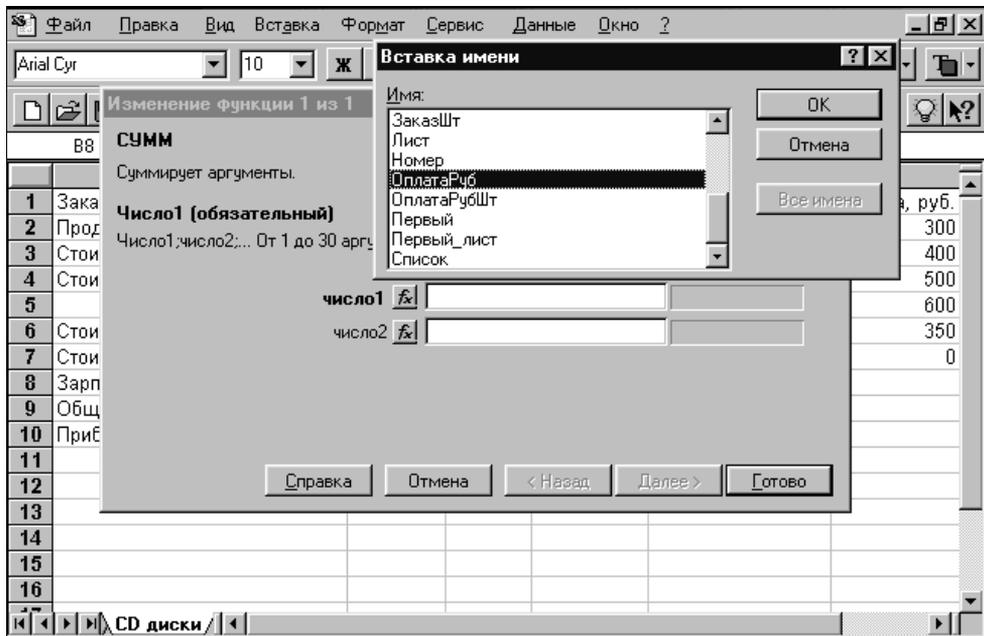


Рис. 4.21. Ввод имени в ячейку из диалогового окна **Вставка имени**

4.10. Планирование рекламной кампании

Прежде всего опишем функции **МАКС** (**MAX**) и **ПОИСКПОЗ** (**MATCH**), которые используются в дальнейшем при рассмотрении примера составления оптимального плана рекламной кампании.

Функция **МАКС** возвращает максимальный элемент массива. Функция **ПОИСКПОЗ** возвращает относительную позицию элемента массива, который соответствует указанному значению. Функция **ПОИСКПОЗ** используется вместо функций типа **ПРОСМОТР**, если нужна позиция элемента, а не сам элемент.

Синтаксис:

ПОИСКПОЗ(*искомое_значение*; *просматриваемый_массив*;
тип_сопоставления)

Аргументы:

искомое_значение

Значение, для которого ищется соответствие в аргументе *просматриваемый_массив*. Например, когда вы ищете номер телефона в телефонной книге, вы используете фамилию человека как *искомое_значение*.

Оно может быть значением (числом, текстом или логическим значением) или ссылкой на ячейку, содержащую число, текст или логическое значение

просматриваемый_массив Непрерывный интервал ячеек, который возможно содержит искомые значения. Просматриваемый_массив может быть массивом или ссылкой на массив

тип_сопоставления Число: -1, 0 или 1. Тип_сопоставления указывает, как Excel сопоставляет искомое_значение со значениями в аргументе просматриваемый_массив

Рассмотрим подробнее возможные варианты:

- Если тип_сопоставления равен 1, то функция ПОИСКПОЗ находит наибольшее значение, которое равно или меньше, чем искомое_значение (просматриваемый_массив должен быть упорядочен по возрастанию: ..., -2, -1, 0, 1, 2, ..., A-Z, ЛОЖЬ, ИСТИНА).
- Если тип_сопоставления равен 0, то функция ПОИСКПОЗ находит первое значение, которое в точности равно аргументу искомое_значение (просматриваемый_массив может быть расположен в любом порядке).
- Если тип_сопоставления равен -1, то функция ПОИСКПОЗ находит наименьшее значение, которое равно или больше, чем искомое_значение (просматриваемый_массив должен быть упорядочен по убыванию).
- Если тип_сопоставления опущен, то предполагается, что он равен 1.

Теперь рассмотрим следующий пример. Фирма еженедельно анализирует, как обстоят дела со сбытом одного из видов своей продукции и дает оценку: отличную ("о" — состояние 1), хорошую ("х" — состояние 2) или удовлетворительную ("у" — состояние 3). Необходимо принять решение о целесообразности рекламирования этой продукции с целью расширения ее сбыта.

Приведенные на рис. 4.22 в диапазонах В3:D5 и В6:D8 матрицы P^1 и P^2 определяют переходные вероятности без рекламы и при ее наличии в течение любой недели.

Так, $P^1_{22} = 0,5$ и $P^1_{23} = 0,5$ означает, что если в предыдущую неделю сбыт был хорошим, то и без рекламы на текущей неделе с равной вероятностью он останется хорошим или станет удовлетворительным. Соответствующие доходы заданы матрицами R^1 и R^2 в диапазонах Е3:G5 и Е6:G8. Отметим, что элементы матрицы R^2 учитывают затраты на рекламу. Необходимо спланировать оптимальную рекламную кампанию на последующие три недели.

Для общности предположим, что план составляется на N недель, а число состояний для каждого этапа равно m . Пусть $f_n(i)$ — оптимальный ожидаемый

	A	B	C	D	E	F	G	H	I	J	K
1	Варианты	P (Вероятности)			R (Прибыль)			Ожидаемая прибыль			
2		о	х	у	о	х	у		Неделя 3	Неделя 2	Неделя 1
3		0.2	0.5	0.3	7000	6000	3000		5300	8030	10381
4	1 х	0	0.5	0.5	0	5000	1000		3000	4750	6868
5	у	0	0	1	0	0	-1000		-1000	-600	1125
6	о	0.3	0.6	0.1	6000	5000	-1000		4700	8190	10736
7	2 х	0.1	0.6	0.3	7000	4000	0		3100	5610	7923
8	у	0.05	0.4	0.55	6000	3000	-2000		400	2125	4222
9								Варианты			
10	i (Состояние)	F(3,i)	к	F(2,i)	к	F(1,i)	к	1	5300	8030	10381
11	1	5300	1	8190	2	10736	2	2	4700	8190	10736
12	2	3100	2	5610	2	7923	2	1	3000	4750	6868
13	3	400	2	2125	2	4222	2	2	3100	5610	7923
14								1	-1000	-600	1125
15								2	400	2125	4222

Рис. 4.22. Планирование рекламной кампании

доход за этапы $n, n+1, \dots, N$ при условии, что система находится в состоянии i в начале n -й недели.

Тогда:

$$f_n(i) = \max_k \left\{ \sum_{j=1}^m p_{ij}^k \left(r_{ij}^k + f_{n+1}(j) \right) \right\}, \quad n \in [1, N],$$

где $f_{N+1}(j) = 0$ при всех j .

Пусть

$$v_i^k = \sum_{j=1}^m p_{ij}^k r_{ij}^k,$$

тогда

$$f_N(i) = \max_k \left\{ v_i^k \right\},$$

$$f_n(i) = \max_k \left\{ v_i^k + \sum_{j=1}^m p_{ij}^k f_{n+1}(j) \right\}, \quad n \in [1, N-1].$$

В ячейку I3 введена формула

=СУММПРОИЗВ (B3 : D3 ; E3 : G3)

вычисляющая v_1^1 , которая протаскивается на диапазон I4:I8 для вычисления v_2^1, \dots, v_3^2 .

В ячейки диапазона I10:I15 последовательно введены формулы

=I3

=I6

=I4

=I7

=I5

=I8

упорядочивающие ожидаемые доходы по следующим парам: первое состояние без рекламы и при ее наличии, второе состояние без рекламы и при ее наличии и третье состояние без рекламы и при ее наличии.

В ячейки диапазона B11:B13 введены формулы

=МАКС(I10:I11)

=МАКС(I12:I13)

=МАКС(I14:I15)

определяющие максимальную ожидаемую прибыль на третьей неделе, если на предыдущей неделе система находилась в первом, втором или третьем состоянии, соответственно. В ячейках диапазона C11:C13 по формулам

=ПОИСКПОЗ(B11;I10:I11;0)

=ПОИСКПОЗ(B12;I12:I13;0)

=ПОИСКПОЗ(B13;I14:I15;0)

определяется оптимальный вариант действий. Если 1, то деньги на рекламу не тратить, а если 2 — то тратить.

Перейдем ко второй неделе рекламной кампании. В ячейку J3 введена формула

=I3+МУМНОЖ(B3:D3; \$B\$11:\$B\$13)

вычисляющая

$$v_1^1 + \sum_{j=1}^3 p_{1j}^1 f_3(j),$$

которая протаскивается на диапазон J4: J8 для вычисления

$$v_2^1 + \sum_{j=1}^3 p_{2j}^1 f_3(j),$$

...

$$v_3^2 + \sum_{j=1}^3 p_{3j}^2 f_3(j).$$

В ячейки диапазона J10: J15 введены последовательно формулы

=J3

=J6

=J4

=J7

=J5

=J8

упорядочивающие ожидаемые доходы по следующим парам: первое состояние без рекламы и при ее наличии, второе состояние без рекламы и при ее наличии и третье состояние без рекламы и при ее наличии.

В ячейки диапазона D11:D13 введены формулы

=МАКС(J10:J11)

=МАКС(J12:J13)

=МАКС(J14:J15)

определяющие максимальную ожидаемую прибыль на второй неделе, если на предыдущей неделе система находилась в первом, втором или третьем состоянии, соответственно. В ячейках диапазона E11:E13 по формулам

=ПОИСКПОЗ(D11;J10:J11;0)

=ПОИСКПОЗ(D12;J12:J13;0)

=ПОИСКПОЗ(D13;J14:J15;0)

определяется оптимальный вариант действий. Аналогично проводятся расчеты для первой недели.

Из рис. 4.22 видно, что на первой и второй неделях необходимо использовать рекламу, не считаясь с состоянием системы, однако, на третьей неделе рекламу следует использовать только тогда, когда система находится во втором или третьем состояниях. Суммарный ожидаемый доход фирмы составит 10736 при отличной оценке, 7923 — при хорошей и 4222 — при удовлетворительной оценке.

4.11. Упражнения

(а) Вычислить n -годовичную ипотечную ссуду покупки квартиры за P руб. с годовой ставкой $i\%$ и начальным взносом $A\%$. Сделать расчет для ежемесячных и ежегодных выплат.

Вариант	n	P	i	A
1	7	170000	5	10
2	8	200000	6	10
3	9	220000	7	20
4	10	300000	8	20
5	11	350000	9	15
6	7	210000	10	15
7	8	250000	11	30
8	9	310000	12	30
9	10	320000	13	25
10	11	360000	14	25

(b) Вас просят дать в долг P руб. и обещают вернуть P_1 руб. через год, P_2 руб. — через два года и т. д., наконец, P_n руб. — через n лет. При какой годовой процентной ставке эта сделка имеет смысл?

Вариант	n	P	P_1	P_2	P_3	P_4	P_5
1	3	17000	5000	7000	8000		
2	4	20000	6000	6000	9000	7000	
3	5	22000	5000	8000	8000	7000	5000
4	3	30000	5000	10000	18000		
5	4	35000	5000	9000	10000	18000	
6	5	21000	4000	5000	8000	10000	11000
7	3	25000	8000	9000	10000		
8	4	31000	9000	10000	10000	15000	
9	5	32000	8000	10000	10000	10000	11000
10	3	36000	10000	15000	21000		

(с) Вас просят дать в долг P руб. и обещают возвращать по A руб. в течение n лет. При какой годовой процентной ставке эта сделка имеет смысл?

Вариант	n	P	A
1	7	170000	30000
2	8	200000	31000
3	9	220000	33000
4	10	300000	34000
5	11	350000	41000
6	7	210000	32000
7	8	250000	37000
8	9	310000	40000
9	10	320000	35000
10	11	360000	41000

(d) Вычислить основные платежи, плату по процентам, общую ежегодную выплату и остаток долга на примере ссуды P руб. под годовую ставку $i\%$ на срок n лет.

Вариант	n	P	i
1	7	170000	5
2	8	200000	6
3	9	220000	7
4	10	300000	8
5	11	350000	9
6	7	210000	10
7	8	250000	11
8	9	310000	12
9	10	320000	13
10	11	360000	14

(е) Вы берете в долг P руб. под годовую ставку $i\%$ и собираетесь выплачивать по A руб. в год. Сколько лет займут эти выплаты?

Вариант	P	A	i
1	170000	31000	3
2	200000	32000	4
3	220000	33000	5
4	300000	34000	6
5	370000	41000	7
6	210000	32000	8
7	260000	37000	9
8	310000	40000	10
9	320000	35000	4
10	360000	41000	5

(ф) Вас просят дать в долг P руб. в день D и обещают вернуть P_1 руб. в день D_1 , P_2 руб. — в день D_2 и т. д., наконец, P_n руб. — в день D_n . Имеет ли смысл эта сделка при годовой ставке $i\%$? (i, P, n, P_1, \dots, P_n взять из задания (б)).

Вариант	D	D_1	D_2	D_3	D_4	D_5
1	12.02.98	22.10.98	11.05.99	25.12.99		
2	13.02.98	23.10.98	12.05.99	26.12.99	12.07.01	
3	14.02.98	24.10.98	13.05.99	27.12.99	13.07.01	11.05.02
4	15.02.98	25.10.98	14.05.99	28.12.99		
5	16.02.98	26.10.98	15.05.99	29.12.99	12.07.01	
6	17.02.98	27.10.98	16.05.99	30.12.99	13.07.01	11.05.02
7	18.02.98	28.10.98	17.05.99	31.12.99		
8	19.02.98	29.10.98	18.05.99	01.01.00	12.07.01	
9	20.02.98	30.10.98	19.05.99	02.01.00	13.07.01	11.05.02
10	21.02.98	31.10.98	20.05.99	03.01.00		

(г) Составить отчетную ведомость реализации товаров n магазинами с месяца A по месяц B , приведенную на рис. 4.15.

Вариант	A	B	n
1	май	декабрь	3
2	июнь	январь	4
3	июль	октябрь	5
4	август	январь	6
5	сентябрь	декабрь	7
6	октябрь	март	8
7	ноябрь	март	9
8	декабрь	июль	10
9	январь	июль	4
10	февраль	август	5

В качестве стоимостей товаров введите произвольные трехзначные числа, а в качестве объемов их реализации — произвольные двузначные числа.

(h) Вы берете в долг P руб. под годовую ставку $i\%$ и собираетесь отдавать по A руб. в год. Сколько лет займут выплаты?

Вариант	1	2	3	4	5	6	7	8	9	10
A	200	190	178	164	146	243	320	423	521	711
P	1000	1700	3100	5900	6190	6509	6860	7246	7670	8138
i	2	3	4	2	2	3	4	5	6	7

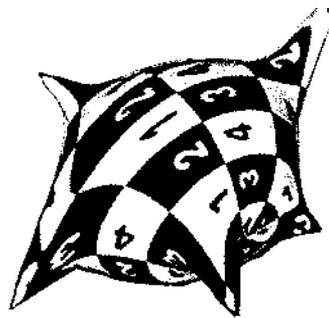
(i) Вы собираетесь вкладывать по A руб. в течение n лет при годовой ставке $i\%$. Сколько денег будет на счете через n лет?

Вариант	1	2	3	4	5	6	7	8	9	10
A	200	190	178	164	146	243	320	423	521	711
n	10	11	12	13	14	8	9	10	11	12
i	2	3	4	2	2	3	4	5	6	7

(j) Определить процентную ставку для n -летнего займа в P руб. с ежегодной выплатой в A руб.

Вариант	1	2	3	4	5	6	7	8	9	10
A	200	190	178	164	146	243	320	423	521	711
P	1000	1700	3100	5900	6190	6509	6860	7246	7670	8138
i	2	3	4	2	2	3	4	5	6	7

Глава 5



Поиск решения

Просто удивительно, насколько часто в повседневной жизни мы сталкиваемся с необходимостью решать оптимизационные задачи. Так, каждый раз, когда мы заходим в магазин, перед нами встает одна и та же проблема: как максимально удовлетворить потребности, соизмеряясь с возможностями кошелька. Что же говорить о менеджерах, экономистах, которые постоянно сталкиваются с разнообразными проблемами, начиная с планирования штата сотрудников, фонда зарплаты и заканчивая составлением оптимального плана производства, планированием рекламной кампании по продвижению продукции на рынок и оптимизацией капиталовложений.

Несмотря на все многообразие таких задач, встречающихся в жизни и экономике на каждом шагу, Excel предлагает единый мощный инструмент их решения — средство поиска решений. От вас требуется только грамотно сформулировать для Excel вашу задачу, а оптимальное решение будет найдено быстро и точно. Данная глава на большом количестве конкретных примеров научит вас правильно решать все вышеперечисленные задачи.

5.1. Линейная оптимизационная задача

В данном разделе рассмотрим, как с помощью средства поиска решений решаются линейные оптимизационные задачи на примере трех типичных ситуаций: планирование производства, составление сплавов или смесей и планирование штатного расписания.

5.1.1. Планирование производства красок

Рассмотрим следующую задачу планирования производства. Небольшая фабрика выпускает два типа красок: для внутренних (I) и наружных (E) работ.

Продукция обоих видов поступает в оптовую продажу. Для производства красок используются два исходных продукта А и В. Максимально возможные суточные запасы этих продуктов составляют 6 и 8 тонн, соответственно. Расходы продуктов А и В на 1 т соответствующих красок приведены в табл. 5.1.

Таблица 5.1. Исходные данные задачи о планировании производства красок

Исходный продукт	Расход исходных продуктов на тонну краски, т		Максимально возможный запас, т
	краска Е	краска I	
А	1	2	6
В	2	1	8

Изучение рынка сбыта показало, что суточный спрос на краску I никогда не превышает спроса на краску Е более чем на 1 т. Кроме того, установлено, что спрос на краску I никогда не превышает 2 т в сутки. Оптовые цены одной тонны красок равны: 3000 руб. для краски Е и 2000 руб. для краски I. Какое количество краски каждого вида должна производить фабрика, чтобы доход от реализации продукции был максимальным?

Для решения этой задачи необходимо построить математическую модель. Процесс построения модели можно начать с ответа на следующие три вопроса:

1. Для определения каких величин строится модель (т. е. каковы переменные модели)?
2. В чем состоит цель, для достижения которой из множества всех допустимых значений переменных выбираются оптимальные?
3. Каким ограничениям должны удовлетворять неизвестные?

В нашем случае фабрике необходимо спланировать объем производства красок так, чтобы максимизировать прибыль. Поэтому переменными являются: x_I — суточный объем производства краски I и x_E — суточный объем производства краски Е.

Суммарная суточная прибыль от производства x_I краски I и x_E краски Е равна $z = 3000 x_E + 2000 x_I$. Целью фабрики является определение среди всех допустимых значений x_E и x_I таких, которые максимизируют суммарную прибыль, т. е. целевую функцию z .

Перейдем к ограничениям, которые налагаются на x_E и x_I . Объем производства красок не может быть отрицательным, следовательно:

$$x_E, x_I \geq 0.$$

Расход исходного продукта для производства обоих видов красок не может превосходить максимально возможный запас данного исходного продукта, следовательно:

$$x_E + 2x_I \leq 6,$$

$$2x_E + x_I \leq 8.$$

Кроме того, ограничения на величину спроса на краски таковы:

$$x_I - x_E \leq 1,$$

$$x_I \leq 2.$$

Таким образом, математическая модель данной задачи имеет следующий вид:

максимизировать

$$z = 3000 x_E + 2000 x_I$$

при следующих ограничениях:

$$x_E + 2x_I \leq 6,$$

$$2x_E + x_I \leq 8,$$

$$x_I - x_E \leq 1,$$

$$x_I \leq 2,$$

$$x_I, x_E \geq 0.$$

Заметим, что данная модель является линейной, т. к. целевая функция и ограничения линейно зависят от переменных.

Решим данную задачу с помощью команды **Сервис, Поиск решения** (Tools, Solver). Средство поиска решений является одной из надстроек Excel. Если в меню **Сервис** (Tools) отсутствует команда **Поиск решения** (Solver), то для ее установки необходимо выполнить команду **Сервис, Надстройки, Поиск решения** (Tools, Add-ins, Solver).

Отведем ячейки A3 и B3 под значения переменных x_E и x_I (рис. 5.1).

	А	В	С
1	Переменные:		
2	x_E	x_I	
3			
4	Функция цели:		=3000*A3+2000*B3
5			
6	Ограничения:		
7	=A3+2*B3	6	
8	=2*A3+B3	8	
9	=B3-A3	1	
10	=B3	2	
11			

Рис. 5.1. Диапазоны, отведенные под переменные, целевую функцию и ограничения

В ячейку С4 введем функцию цели

$$=3000*A3+2000*B3$$

в ячейки А7:А10 введем левые части ограничений

$$=A3+2*B3$$

$$=2*A3+B3$$

$$=B3-A3$$

$$=B3$$

а в ячейки В7:В10 — правые части ограничений.

После этого выберем команду **Сервис, Поиск решения** (Tools, Solver) и заполним открывшееся диалоговое окно **Поиск решения** (Solver), как показано на рис. 5.2 (подробно о работе с этим окном будет рассказано в следующем разделе).

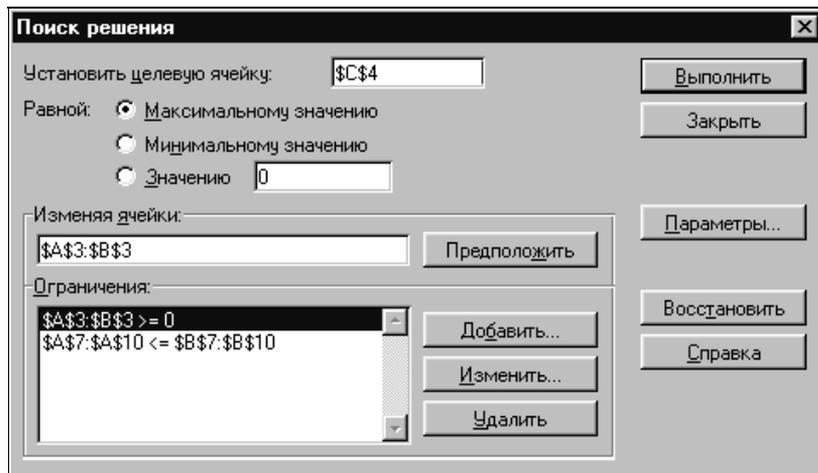


Рис. 5.2. Диалоговое окно **Поиск решения** задачи о планировании производства красок

После нажатия кнопки **Выполнить** (Solve) открывается окно **Результаты поиска решения** (Solver Results), которое сообщает, что решение найдено (рис. 5.3).

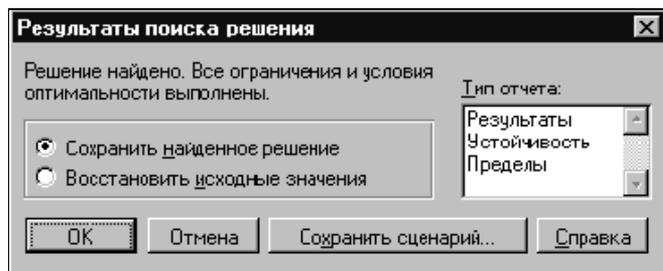


Рис. 5.3. Диалоговое окно **Результаты поиска решения**

Результаты расчета нашей задачи (оптимальный план производства и соответствующая ему прибыль) представлены на рис. 5.4. Как видно из рисунка, оптимальным является производство $3\frac{1}{3}$ т краски Е и $1\frac{1}{3}$ т краски I в сутки. Этот объем производства принесет фабрике $12\frac{2}{3}$ тыс. руб. прибыли.

	А	В	С
1	Переменные:		
2	x _E	x _I	
3	3.33333333	1.33333333	
4	Функция цели:		12666.67
5	Ограничения:		
7		6	6
8		8	8
9		-2	1
10	1.33333333	2	
11			

Рис. 5.4. Результаты расчета с помощью средства поиска решений для задачи планирования производства красок

5.1.2. Элементы диалогового окна *Поиск решения*

Рассмотрим более подробно элементы диалогового окна **Поиск решения** (Solver) (рис. 5.2).

В поле **Установить целевую ячейку** (Set Target Cell) диалогового окна **Поиск решения** (Solver) дается ссылка на ячейку с функцией, для которой будет находиться максимум, минимум или заданное значение. В задаче о производстве красок в поле **Установить целевую ячейку** (Set Target Cell) вводится $\$C\4 (рис. 5.2).

Тип взаимосвязи между решением и целевой ячейкой задается путем установки переключателя в группе **Равной** (Equal To). Для нахождения максимального или минимального значения целевой функции этот переключатель ставится в положение **Максимальному значению** (Max) или **Минимальному значению** (Min), соответственно. Для нахождения значения целевой функции, заданного в поле группы **Равной** (Equal To), переключатель ставится в положение **Значению** (Value of). В нашей задаче о красках установим переключатель в положение **Максимальному значению** (Max), т. к. планируем производство, обеспечивающее максимальную прибыль.

В поле **Изменяя ячейки** (By Changing Cells) указываются ячейки, которые должны изменяться в процессе поиска решения задачи, т. е. ячейки отведенные под переменные задачи. В нашем случае введем в поле **Изменяя ячейки** (By Changing Cells) диапазон $\$A\$3:\$B\3 .

Ограничения, налагаемые на переменные задачи, отображаются в поле **Ограничения** (Subject to the Constraints) (рис. 5.2). Средство поиска решений допускает ограничения в виде равенств, неравенств, а также позволяет ввести требование целочисленности переменных. Ограничения добавляются по одному. Для ввода ограничений нажмите кнопку **Добавить** (Add) в диалоговом окне **Поиск решения** (Solver) (рис. 5.2) и в открывшемся диалоговом окне **Добавление ограничения** (Add Constraint) (рис. 5.5) заполните поля.

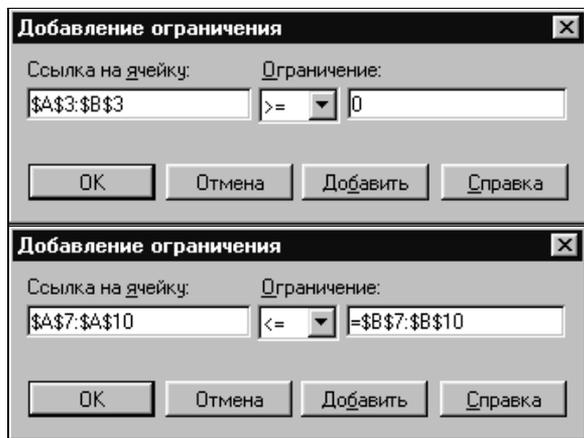


Рис. 5.5. Диалоговое окно **Добавление ограничения**. Ввод двух групп ограничений

В поле **Ссылка на ячейку** (Cell Reference) введите левую часть ограничения — $\$A\$3:\$B\3 , а в поле **Ограничение** (Constraints) — правую часть, в нашем примере — 0. С помощью раскрывающегося списка \leq вводится тип соотношения между левой и правой частями ограничения. В нашем примере это \geq . Таким образом, требование неотрицательности переменных задано.

Нажмите кнопку **Добавить** (Add) в диалоговом окне **Добавление ограничения** (Add Constraint) (рис. 5.5) и введите вторую группу ограничений, налагаемых на переменные. Нажатие кнопки **ОК** завершает ввод ограничений. Обратите внимание на то, что ограничения удобнее задавать в виде диапазонов, как это сделано в рассматриваемом примере.

Теперь нажмите кнопку **Параметры** (Options) в диалоговом окне **Поиск решения** (Solver) (рис. 5.2), для того чтобы проверить, какие параметры заданы для поиска решений.

В открывшемся диалоговом окне **Параметры поиска решения** (Solver Options) (рис. 5.6) можно изменять условия и варианты поиска решения исследуемой задачи, а также загружать и сохранять оптимизируемые модели. Значения и состояния элементов управления, используемые по умолчанию, подходят для решения большинства задач.

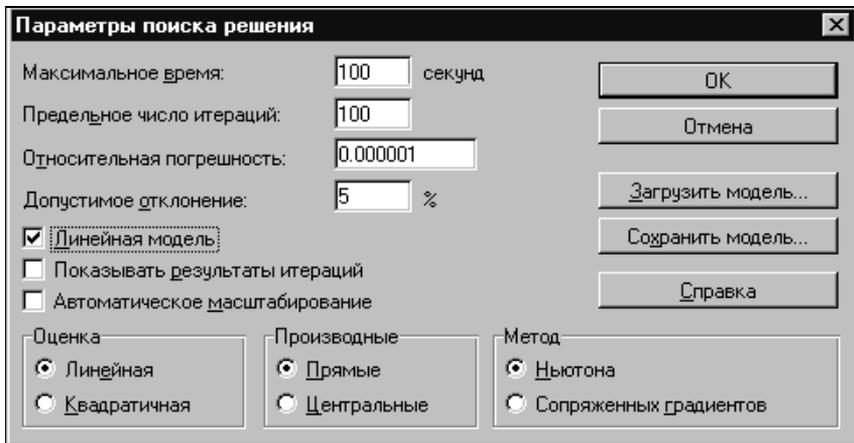


Рис. 5.6. Диалоговое окно **Параметры поиска решения**

Рассмотрим элементы этого окна:

- Поле **Максимальное время** (Max Time) служит для ограничения времени, отпускаемого на поиск решения задачи
- Поле **Предельное число итераций** (Iteration) служит для ограничения числа промежуточных вычислений
- Поля **Относительная погрешность** (Precision) и **Допустимое отклонение** (Tolerance) служат для задания точности, с которой ищется решение. Рекомендуется после нахождения решения с величинами данных параметров, заданными по умолчанию, повторить вычисления с большей точностью и меньшим допустимым отклонением и сравнить с первоначальным решением. Использование подобной проверки особенно рекомендуется для задач с требованием целочисленности переменных
- Флажок **Линейная модель** (Assume Linear model) служит для поиска решения линейной задачи оптимизации или линейной аппроксимации нелинейной задачи. В случае нелинейной задачи этот флажок должен быть сброшен, в случае линейной задачи — установлен, т. к. в противном случае возможно получение неверного результата
- Флажок **Показывать результаты итераций** (Show Iteration Results) служит для приостановки поиска решения и просмотра результатов отдельных итераций
- Флажок **Автоматическое масштабирование** (Use Automatic Scaling) служит для включения автоматической нормализации входных и выходных значений, качественно различающихся по величине, например, при максимизации прибыли в процентах по отношению к вложениям, исчисляемым в миллионах рублей
- Группа **Оценка** (Estimates) служит для выбора метода экстраполяции

Группа **Производные** (Derivatives) служит для выбора метода численного дифференцирования

Группа **Метод** (Search) служит для выбора алгоритма оптимизации

Для того чтобы вывести отчет о результатах решения задачи выберите в диалоговом окне **Результаты поиска решения** (Solver Results) (рис. 5.3) требуемый тип отчета: **Результаты, Устойчивость, Пределы** (Answer, Sensitivity, Limit). Соответствующие отчеты приведены на рис. 5.7–5.9.

Microsoft Excel 7.0 Отчет по пределам
Рабочий лист: [qq.xls]краски
Отчет создан: 22.7.98 20:19

Целевое		
Ячейка	имя	Значение
\$C\$4	Функция цели:	12666.66667

Ячейка	Изменяемое имя	Значение	Нижний предел	Целевой результат	Верхний предел	Целевой результат
\$A\$3	xЕ	3.333333333	0.333333333	3666.666667	3.333333333	12666.66667
\$B\$3	xI	1.333333333		10000	1.333333333	12666.66667

Рис. 5.7. Отчет по пределам

Microsoft Excel 7.0 Отчет по устойчивости
Рабочий лист: [qq.xls]краски
Отчет создан: 22.7.98 20:19

Изменяемые ячейки

Ячейка	Имя	Результ. значение	Редуц. стоимость	Целевой Коэффициент	Допустимое Увеличение	Допустимое Уменьшение
\$A\$3	xЕ	3.333333333	0	3000	1000	2000
\$B\$3	xI	1.333333333	0	2000	4000	500

Ограничения

Ячейка	Имя	Результ. значение	Теневая Цена	Ограничение Правая часть	Допустимое Увеличение	Допустимое Уменьшение
\$A\$7	Ограничения:	6	333.3333333	6	1	2
\$A\$8	Ограничения:	8	1333.333333	8	4	2
\$A\$9	Ограничения:	-2	0	1	1E+30	3
\$A\$10	Ограничения:	1.333333333	0	2	1E+30	0.666666667

Рис. 5.8. Отчет по устойчивости

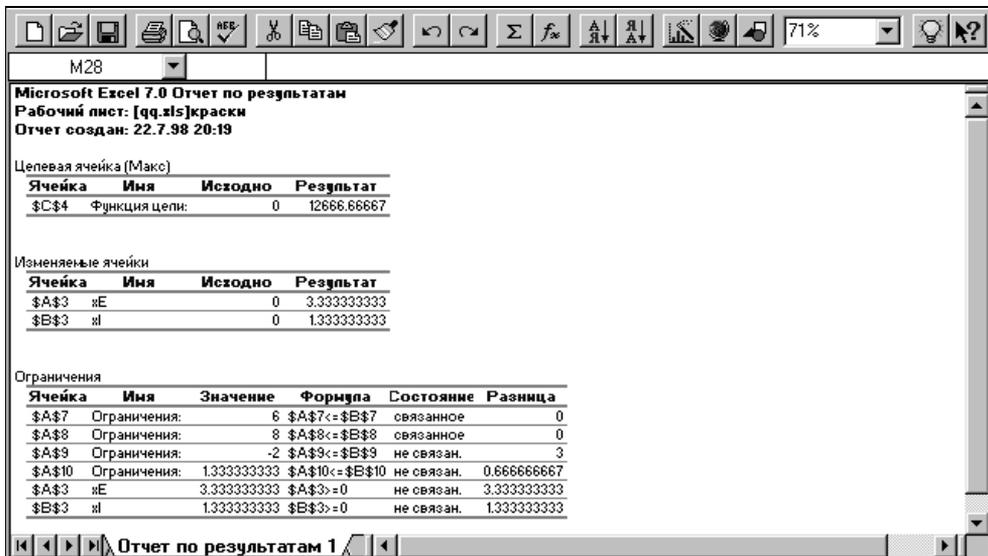


Рис. 5.9. Отчет по результатам

5.1.3. Определение состава сплавов

Рассмотрим задачу: для получения сплавов A и B используются четыре металла I, II, III и IV, требования к содержанию которых в сплавах A и B приведены в табл. 5.2.

Таблица 5.2. Требования к содержанию металлов в задаче определения состава сплавов

Сплав	Требования к содержанию металла
A	Не более 80% металла I
	Не более 30% металла II
B	От 40 до 60% металла II
	Не менее 30% металла III
	Не более 70% металла IV

Характеристики и запасы руд, используемых для производства металлов I, II, III и IV, указаны в табл. 5.3.

Таблица 5.3. Характеристики и запасы руд в задаче об определении состава сплавов

Руда	Максимальный запас, т	Состав, %					Цена, \$/т
		I	II	III	IV	Другие компоненты	
1	1000	20	10	30	30	10	30
2	2000	10	20	30	30	10	40
3	3000	5	5	70	20	0	50

Пусть цена 1 т сплава A равна 200 долларов, а 1 т сплава B — 210 долларов. Необходимо максимизировать прибыль от продажи сплавов A и B .

Обозначим через x_{1A} , x_{2A} , x_{3A} , x_{4A} и x_{1B} , x_{2B} , x_{3B} , x_{4B} количество I, II, III и IV металлов, используемых для получения сплавов A и B , соответственно. Количество использованной i -й руды обозначим y_i , $i \in [1, 3]$.

Тогда математическая модель данной задачи имеет вид:

максимизировать:

$$z = 200(x_{1A} + x_{2A} + x_{3A} + x_{4A}) + 210(x_{1B} + x_{2B} + x_{3B} + x_{4B}) - 30y_1 - 40y_2 - 50y_3$$

при ограничениях на состав сплавов (на основании данных из табл. 5.2):

$$x_{1A} \leq 0,8(x_{1A} + x_{2A} + x_{3A} + x_{4A}),$$

$$x_{2A} \leq 0,3(x_{1A} + x_{2A} + x_{3A} + x_{4A}),$$

$$x_{2B} \leq 0,6(x_{1B} + x_{2B} + x_{3B} + x_{4B}),$$

$$x_{2B} \geq 0,4(x_{1B} + x_{2B} + x_{3B} + x_{4B}),$$

$$x_{3B} \geq 0,3(x_{1B} + x_{2B} + x_{3B} + x_{4B}),$$

$$x_{4B} \leq 0,7(x_{1B} + x_{2B} + x_{3B} + x_{4B}),$$

на характеристики и состав руды (на основании данных из табл. 5.3):

$$x_{1A} + x_{1B} \leq 0,2y_1 + 0,1y_2 + 0,05y_3,$$

$$x_{2A} + x_{2B} \leq 0,1y_1 + 0,2y_2 + 0,05y_3,$$

$$x_{3A} + x_{3B} \leq 0,3y_1 + 0,3y_2 + 0,7y_3,$$

$$x_{4A} + x_{4B} \leq 0,3y_1 + 0,3y_2 + 0,2y_3,$$

а также на диапазоны использования переменных:

$$x_{iA} \geq 0, x_{iB} \geq 0, \quad i \in [1, 4],$$

$$0 \leq y_1 \leq 1000,$$

$$0 \leq y_2 \leq 2000,$$

$$0 \leq y_3 \leq 3000.$$

Отведем под переменные x_{iA} , x_{iB} , $i \in [1, 4]$ диапазон ячеек C3:D6, а под переменные y_i , $i \in [1, 3]$ — диапазон ячеек F3:F5 (рис. 5.10).

	A	B	C	D	E	F	G	H	I	J
1			Сплав							
2	М		A	B		Руда		Запас руды		
3	е	1	550	0	1	1000		1000		
4	т	2	0	650	2	2000		2000		
5	а	3	2025	975	3	3000		3000		
6	л	4	1500	0						
7	л									
8	О		-2710							
9	г		-1223		Функция цели		896250			
10	р		-325							
11	а		0							
12	н		-488							
13	и		-1138							
14	ч		0							
15	е		0							
16	н		0							
17	и		0							
18	я									
19										
20										
21										
22										
23										

Рис. 5.10. Результат расчета в задаче определения состава сплавов

В ячейку G9 введем функцию цели

$$=200*\text{СУММ}(C3:C6)+210*\text{СУММ}(D3:D6)-30*F3-40*F4-50*F5$$

В диапазон ячеек C8:C17 введем левые части ограничений, причем преобразуем их к виду, когда все переменные находятся слева, а все знаки неравенств — меньше или равно:

$$=C3-0.8*\text{СУММ}(C3:C6)$$

$$=C4-0.3*\text{СУММ}(C3:C6)$$

$$=D4-0.6*\text{СУММ}(D3:D6)$$

$$=0.4*\text{СУММ}(D3:D6)-D4$$

$$=0.3*\text{СУММ}(D3:D6)-D5$$

$=D6-0.7*СУММ(D3:D6)$
 $=СУММ(C3:D3)-0.2*F\$3-0.1*F\$4-0.05*F\$5$
 $=СУММ(C4:D4)-0.1*F\$3-0.2*F\$4-0.05*F\$5$
 $=СУММ(C5:D5)-0.3*F\$3-0.3*F\$4-0.7*F\$5$
 $=СУММ(C6:D6)-0.3*F\$3-0.3*F\$4-0.2*F\$5$

В диапазон ячеек Н3:Н5 введем количество имеющихся запасов руд. Выберем команду **Сервис, Поиск решения** (Tools, Solver) и заполним диалоговое окно **Поиск решения** (Solver), как показано на рис. 5.11.

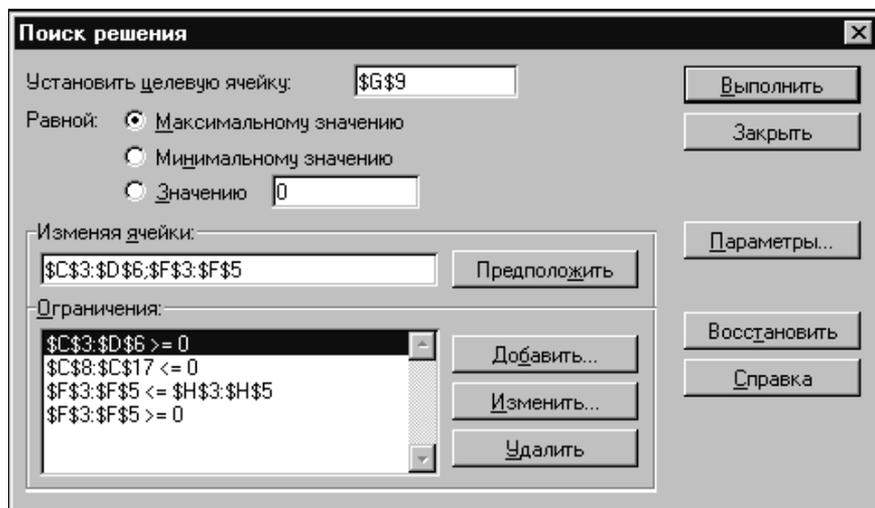


Рис. 5.11. Диалоговое окно **Поиск решения** в задаче определения состава сплавов

Результаты работы средства поиска решений представлены на рис. 5.10.

5.1.4. Планирование штатного расписания

Авиакомпаниям требуется определить, сколько стюардесс следует принять на работу в течение шести месяцев при условии, что любая из них должна пройти предварительную подготовку. Потребности в количестве человеко-часов летного времени для стюардесс известны: в январе — 8000, в феврале — 9000, в марте — 8000, в апреле — 10000, в мае — 90000 и в июне — 12000.

Подготовка стюардессы к выполнению своих обязанностей занимает один месяц. Следовательно, прием на работу должен, по крайней мере, на один месяц опережать ввод стюардессы в строй. Кроме того, каждая стюардесса должна в течение месяца, отведенного на ее подготовку, пройти 100-часо-

вую практику непосредственно во время полетов. Таким образом, за счет каждой обучаемой стюардессы в течение месяца освобождается 100 человеко-часов летного времени, отведенного для уже обученных стюардесс.

Каждая полностью обученная стюардесса в течение месяца может иметь налет до 150 часов. Авиакомпания в начале января уже имеет 60 опытных стюардесс. При этом ни одну из них не снимают с работы. Установлено также, что приблизительно 10% обучаемых стюардесс по окончании обучения увольняются по каким-либо обстоятельствам. Опытная стюардесса обходится авиакомпании в \$800, а обучаемая — в \$400 в месяц. Необходимо спланировать штат авиакомпании таким образом, чтобы минимизировать издержки за отчетные шесть месяцев.

Для данной задачи также можно разработать математическую модель, но ее удобнее проанализировать в более развернутой форме. Отведем диапазон ячеек В3:В8 под число новых стюардесс, принимаемых на работу с января по июнь (рис. 5.12).

	А	В	С	Д	Е	Ф	Г	Н
1	Месяц	Число новых стюардесс	Требуемое число, ч	Число постоянно работающих стюардесс	Фактическое число, ч	Затраты		
2	Декабрь	60						
3	Январь		8000	60	9000	48000		
4	Февраль		9000	60	9000	48000		
5	Март		8000	60	9000	48000		
6	Апрель		10000	60	9000	48000		
7	Май		9000	60	9000	48000		
8	Июнь		12000	60	9000	48000		
9					Итоговые затраты	288000		
10				Затраты на одну стюардессу		Разрешенный налет, ч		
11				Обучение	Работа	Обучение	Работа	
12				400	800	100	150	
13								
14								
15								
16								
17								
18								

Рис. 5.12. Исходные данные задачи о планировании штатного расписания

В ячейку В2 введем число стюардесс, работающих в декабре. В диапазоне ячеек D3:D8 вычислим число стюардесс, постоянно работающих в текущем месяце, введя в ячейки D3 и D4 формулы

=B2

=D3+0.9*B3

и протаскивая последнюю из них на диапазон D5:D8. В диапазоне E3:E8 вычислим налет по месяцам, введя в ячейку E3 формулу

$$=D3*\$G\$12+B3*\$F\$12$$

и протаскивая ее на диапазон E4:E8, где в ячейки F12 и G12 введены допустимый налет обучаемой и работающей стюардесс. В диапазоне F3:F8 вычислим затраты по месяцам, введя в ячейку F3 формулу

$$=D3*\$E\$12+B3*\$D\$12$$

и протаскивая ее на диапазон F3:F8, где в ячейки D12 и E12 введены затраты на обучение и работу стюардессы. Вычислим суммарные затраты за планируемый период в ячейке F9 по формуле

$$=СУММ(F3:F8)$$

Теперь все готово, чтобы перейти к выбору команды **Сервис, Поиск решения** (Tools, Solver). Заполним открывшееся диалоговое окно **Поиск решения** (Solver), как показано на рис. 5.13.

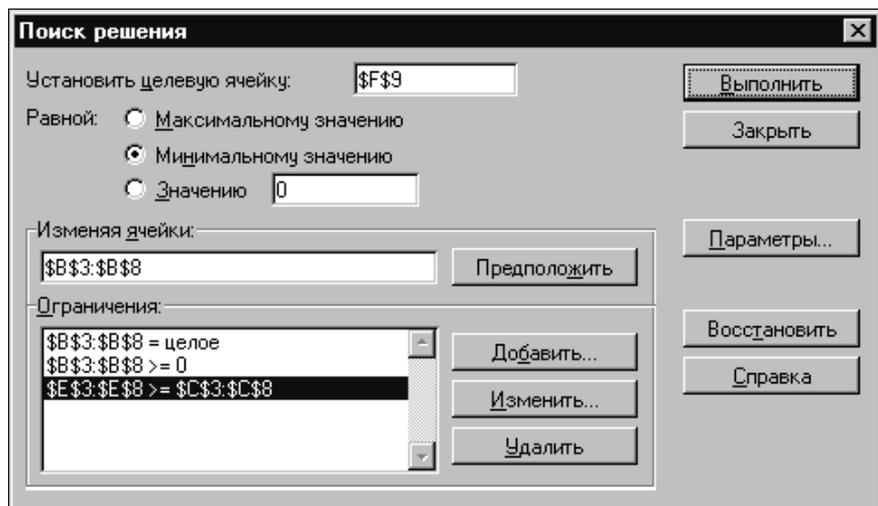


Рис. 5.13. Диалоговое окно **Поиск решения** задачи о планировании штатного расписания

Результаты расчета оптимального штата стюардесс приведены на рис. 5.14.

Согласно этим расчетам фирма в последний месяц планового периода должна взять на обучение 17 новых стюардесс.

Предположим, что авиакомпания по какой-то причине решила не брать в июне на обучение новых стюардесс. Тогда в поле **Ограничения** (Subject to the

Constraints) диалогового окна **Поиск решения (Solver)** надо добавить $B8=0$. Оптимальное решение при таком дополнительном ограничении представлено на рис. 5.15, из которого видно, что оно приведет к временному повышению текущих затрат.

	A	B	C	D	E	F	G	H
1	Месяц	Число новых стюардесс	Требуемое число, ч	Число постоянно работающих стюардесс	Фактическое число, ч	Затраты		
2	Декабрь	60						
3	Январь	0	8000	60	9000	48000		
4	Февраль	0	9000	60	9000	48000		
5	Март	0	8000	60	9000	48000		
6	Апрель	10	10000	60	10000	52000		
7	Май	0	9000	69	10350	55200		
8	Июнь	17	12000	69	12050	62000		
9					Итоговые затраты	313200		
10				Затраты на одну стюардессу		Разрешенный налет, ч		
11				Обучение	Работа	Обучение	Работа	
12				400	800	100	150	
13								

Рис. 5.14. Результаты расчета с помощью средства поиска решений в задаче о планировании штатного расписания

	A	B	C	D	E	F	G	H
1	Месяц	Число новых стюардесс	Требуемое число, ч	Число постоянно работающих стюардесс	Фактическое число, ч	Затраты		
2	Декабрь	60						
3	Январь	0	8000	60	9000	48000		
4	Февраль	0	9000	60	9000	48000		
5	Март	0	8000	60	9000	48000		
6	Апрель	10	10000	60	10000	52000		
7	Май	13	9000	69	11650	60400		
8	Июнь	0	12000	81	12105	64560		
9					Итоговые затраты	320960		
10				Затраты на одну стюардессу		Разрешенный налет, ч		
11				Обучение	Работа	Обучение	Работа	
12				400	800	100	150	
13								

Рис. 5.15. Решение задачи о планировании штатного расписания в том случае, если набор стюардесс на обучение в июне не производится

5.2. Транспортная задача

Рассмотрим еще один пример, где используется средство поиска решений. Предположим, что фирма имеет 4 фабрики и 5 центров распределения ее товаров. Фабрики фирмы располагаются в Денвере, Бостоне, Новом Орлеане и Далласе с производственными возможностями 200, 150, 225 и 175 единиц продукции ежедневно, соответственно. Центры распределения товаров фирмы располагаются в Лос-Анджелесе, Далласе, Сент-Луисе, Вашингтоне и Атланте с потребностями в 100, 200, 50, 250 и 150 единиц продукции ежедневно, соответственно. Хранение на фабрике единицы продукции, не поставленной в центр распределения, обходится в \$0,75 в день, а штраф за просроченную поставку единицы продукции, заказанной потребителем в центре распределения, но там не находящейся, равен \$2,5 в день. Стоимость перевозки единицы продукции с фабрик в пункты распределения приведена в табл. 5.4.

Таблица 5.4. Транспортные расходы

	1	2	3	4	5
	Лос-Анджелес	Даллас	Сент-Луис	Вашингтон	Атланта
1 Денвер	1.5	2	1.75	2.25	2.25
2 Бостон	2.5	2	1.75	1	1.5
3 Новый Орлеан	2	1.5	1.5	1.75	1.75
4 Даллас	2	0.5	1.75	1.75	1.75

Необходимо так спланировать перевозки, чтобы минимизировать суммарные транспортные расходы.

Поскольку данная модель сбалансирована (суммарный объем произведенной продукции равен суммарному объему потребностей в ней), то в этой модели не надо учитывать издержки, связанные как со складированием, так и с недопоставками продукции. В противном случае в модель нужно было бы ввести:

- В случае перепроизводства — фиктивный пункт распределения, стоимость перевозок единицы продукции в который полагается равной стоимости складирования, а объемы перевозок — объемам складирования излишков продукции на фабриках
- В случае дефицита — фиктивную фабрику, стоимость перевозок единицы продукции с которой полагается равной стоимости штрафов за недопоставку продукции, а объемы перевозок — объемам недопоставок продукции в пункты распределения

Для решения данной задачи построим ее математическую модель. Неизвестными в данной задаче являются объемы перевозок. Пусть x_{ij} — объем перевозок с i -й фабрики в j -й центр распределения. Функция цели — это суммарные транспортные расходы, т. е.

$$z = \sum_{i=1}^4 \sum_{j=1}^5 c_{ij} x_{ij},$$

где c_{ij} — стоимость перевозки единицы продукции с i -й фабрики в j -й центр распределения.

Неизвестные в данной задаче должны удовлетворять следующим ограничениям:

- Объемы перевозок не могут быть отрицательными
- Так как модель сбалансирована, то вся продукция должна быть вывезена с фабрик, а потребности всех центров распределения должны быть полностью удовлетворены

В результате имеем следующую модель:

минимизировать:

$$z = \sum_{i=1}^4 \sum_{j=1}^5 c_{ij} x_{ij},$$

при ограничениях:

$$\sum_{i=1}^4 x_{ij} = b_j, \quad j \in [1, 5],$$

$$\sum_{j=1}^5 x_{ij} = a_i, \quad i \in [1, 4],$$

$$x_{ij} \geq 0, \quad i \in [1, 4], j \in [1, 5].$$

где a_i — объем производства на i -й фабрике, b_j — спрос в j -м центре распределения.

Для решения этой задачи с помощью средства поиска решений введем данные, как показано на рис. 5.16.

В ячейки A1:E4 введены стоимости перевозок. Ячейки A6:E9 отведены под значения неизвестных (объемы перевозок). В ячейки G6:G9 введены объемы

производства на фабриках, а в ячейки A11:E11 введена потребность в продукции в пунктах распределения. В ячейку F10 введена целевая функция

=СУММПРОИЗВ (A1 : E4 ; A6 : E9)

	A	B	C	D	E	F	G	H
1	1.5	2	1.75	2.25	2.25			
2	2.5	2	1.75	1	1.5			
3	2	1.5	1.5	1.75	1.75			
4	2	0.5	1.75	1.75	1.75			
5								
6						0	200	
7						0	150	
8						0	225	
9						0	175	
10	0	0	0	0	0	0		
11	100	200	50	250	150			

Рис. 5.16. Исходные данные транспортной задачи

В ячейки A10:E10 введены формулы

=СУММ (A6 : A9)

=СУММ (B6 : B9)

=СУММ (C6 : C9)

=СУММ (D6 : D9)

=СУММ (E6 : E9)

определяющие объем продукции, ввозимой в центры распределения.

В ячейки F6:F9 введены формулы

=СУММ (A6 : E6)

=СУММ (A7 : E7)

=СУММ (A8 : E8)

=СУММ (A9 : E9)

вычисляющие объем продукции, вывозимой с фабрик.

Теперь выберем команду **Сервис, Поиск решения** (Tools, Solver) и заполним открывшееся диалоговое окно **Поиск решения** (Solver), как показано на рис. 5.17.

Не забудьте в диалоговом окне **Параметры поиска решения** (Solver Options) (рис. 5.6) установить флажок **Линейная модель** (Assume Linear Model). После нажатия кнопки **Выполнить** (Solve) средство поиска решений находит оптимальный план поставок продукции и соответствующие ему транспортные расходы (рис. 5.18).

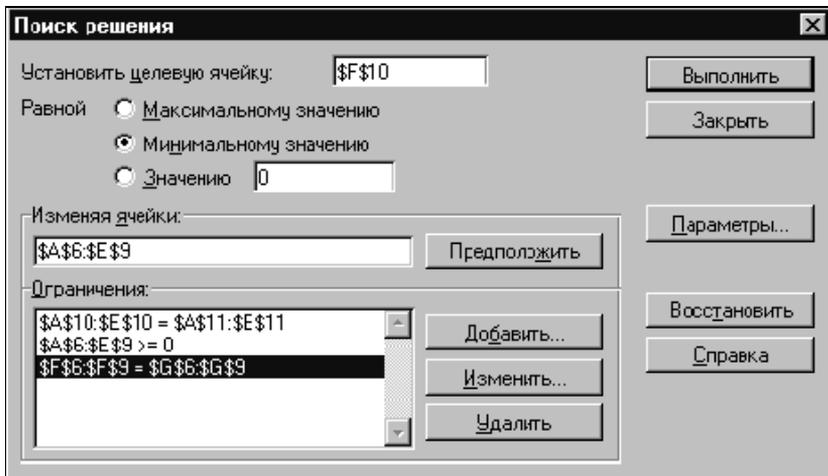


Рис. 5.17. Диалоговое окно Поиск решения для транспортной задачи

	A	B	C	D	E	F	G	H
1	1.5	2	1.75	2.25	2.25			
2	2.5	2	1.75	1	1.5			
3	2	1.5	1.5	1.75	1.75			
4	2	0.5	1.75	1.75	1.75			
5								
6	100	25	50	0	25	200	200	
7	0	0	0	150	0	150	150	
8	0	0	0	100	125	225	225	
9	0	175	0	0	0	175	175	
10	100	200	50	250	150	975		
11	100	200	50	250	150			

Рис. 5.18 Оптимальное решение транспортной задачи

5.3. Решение системы нелинейных уравнений

Кроме оптимизационных задач, средство поиска решений позволяет находить решения систем нелинейных уравнений. Рассмотрим, как это делается, на примере решения следующей системы уравнений:

$$\begin{cases} x^2 + y^2 = 3, \\ 2x + 3y = 1. \end{cases} \quad (5.1)$$

Напомним, что пара (x, y) является решением системы (5.1) тогда и только тогда, когда она является решением следующего уравнения с двумя неизвестными:

$$(x^2 + y^2 - 3)^2 + (2x + 3y - 1)^2 = 0. \quad (5.2)$$

С помощью средства поиска решений вместо системы (5.1) будем решать равносильное ей уравнение (5.2). Отметим, что решением системы уравнений (5.1) являются точки пересечения окружности с радиусом, равным 3, и прямой. Следовательно, уравнение (5.2) имеет не более двух различных решений.

Определяемое решение нелинейной задачи зависит от начального приближения, удачный подбор которого очень важен. В данном случае локализовать корни можно, например, протабулировав левую часть уравнения (5.2) по переменным x и y на отрезке $[-3, 3]$ с шагом 1,5. Результат табуляции приведен на рис. 5.19.

	A	B	C	D	E	F
1		-3.0	-1.5	0.0	1.5	3.0
2	-3.0	481.0	200.3	85.0	74.3	229.0
3	-1.5	237.1	74.5	16.6	2.5	93.1
4	0.0	136.0	30.8	10.0	12.8	100.0
5	1.5	117.1	8.5	4.6	44.5	189.1
6	3.0	241.0	68.3	61.0	158.3	421.0

Рис. 5.19. Результат табуляции левой части уравнения (5.2)

В ячейки A2:A6 и B1:F1 введены значения x и y , соответственно. В ячейку B2 введена формула

$$=(\$A2^2+B\$1^2-3)^2+(2*\$A2+3*B\$1-1)^2$$

вычисляющая правую часть уравнения (5.2) при значениях x и y из ячеек A2 и B1, соответственно. Протащим эту формулу на диапазон B2:F6. Из рис. 5.19 видно, что за начальное приближение к корню разумно выбрать следующие пары значений $(-1,5; 1,5)$, $(1,5; 0)$ и $(1,5; -1,5)$. Можно убедиться, что две последние пары начальных приближений с помощью средства поиска решений будут приводить к нахождению одного и того же решения.

Для нахождения первого корня отведем под переменные x и y ячейки A10 и B10, соответственно, и введем в них начальные приближения -1.5 и 1.5 . В ячейку C10 введем формулу

$$=(A10^2+B10^2-3)^2+(2*A10+3*B10-1)^2$$

вычисляющую значение правой части уравнения (5.2) для этих значений неизвестных. Затем вызовем команду **Сервис, Поиск решения** (Tools, Solver).

В открывшемся диалоговом окне **Поиск решения** (Solver) в поле **Установить целевую ячейку** (Set Target Cell) вводим C10. В поле **Изменяя ячейки** (By

Changing Cells) вводим диапазон ячеек A10:B10. В группе **Равной** (Equal To) установим переключатель в положение **Значению** (Value of), в поле ввода которого вводим 0. Не забудьте убедиться, что в диалоговом окне **Параметры поиска решения** (Solver Options) (рис. 5.6) снят флажок **Линейная модель** (Assume Linear Model).

После нажатия на кнопку **Выполнить** (Solve) средство поиска решений находит решение, которое помещает в ячейки A10 и B10. В данном случае это будут значения -1.268 и 1.179 . В ячейке C10 вычисляется значение правой части уравнения (5.2) при этих значениях неизвестных. Так как средство поиска решений находит решение приближенно, то в ячейке C10 в общем случае будет число отличное от нуля, но достаточно близкое к нему. В нашем примере это $8.89E-09$, т. е. 8.89×10^{-9} . Если в ячейке C10 будет большое число, то решение найдено не верно.

Аналогично находится второе решение, используя начальное приближение (1,5; -1,5). Решением будут значения 1.576 и -0.717. Убедитесь, что начальное приближение (1,5; 0) приводит к тому же решению.

5.4. Задача о назначениях

Рассмотрим пример решения задачи о назначениях. Четверо рабочих могут выполнять четыре вида работ. Стоимости c_{ij} выполнения i -м рабочим j -й работы приведены в ячейках диапазона A1:D4 (рис. 5.20).

	A	B	C	D	E	F	G	H	I	J
1	1	4	6	3		Стоимость работ =				0
2	9	10	7	9						0
3	4	5	11	7						0
4	8	7	8	5						0
5										0
6										0

Рис. 5.20. Стоимости работ в задаче о назначениях

В этой таблице строки соответствуют рабочим, а столбцы — работам. Необходимо составить план выполнения работ так, чтобы все работы были выполнены, каждый рабочий был загружен только на одной работе, а суммарная стоимость выполнения всех работ была минимальной. Отметим, что данная задача является сбалансированной, т. е. число работ совпадает с числом рабочих. Если задача не сбалансирована, то перед началом решения ее необходимо сбалансировать, введя недостающее число фиктивных строчек или столбцов с достаточно большими штрафными стоимостями работ.

Для решения данной задачи построим ее математическую модель. Пусть переменная $x_{ij} = 1$, если i -м рабочим выполняется j -я работа, и $x_{ij} = 0$, если i -м рабочим не выполняется j -я работа. Тогда модель имеет следующий вид:

минимизировать:

$$z = \sum_{i=1}^4 \sum_{j=1}^4 c_{ij} x_{ij},$$

при ограничениях:

$$\sum_{i=1}^4 x_{ij} = 1, \quad j \in [1, 4],$$

$$\sum_{j=1}^4 x_{ij} = 1, \quad i \in [1, 4],$$

$$x_{ij} \in \{0, 1\}, \quad i \in [1, 4], \quad j \in [1, 4].$$

Для решения этой задачи с помощью средства поиска решений отведем под неизвестные диапазон ячеек F2:I5. В ячейку J1 введем целевую функцию

=СУММПРОИЗВ(F2:I5;A1:D4)

вычисляющую стоимость работ. Введем формулы, задающие левые части ограничений (рис. 5.21).

	F	G	H	I	J
1		Стоимость работ =			=СУММПРОИЗВ(F2:I5;A1:D4)
2	1	0	0	0	=СУММ(F2:I2)
3	0	0	1	0	=СУММ(F3:I3)
4	0	1	0	0	=СУММ(F4:I4)
5	0	0	0	1	=СУММ(F5:I5)
6	=СУММ(F2:F5)	=СУММ(G2:G5)	=СУММ(H2:H5)	=СУММ(I2:I5)	

Рис. 5.21. Левые части ограничений в задаче о назначениях

Затем выберем команду **Сервис, Поиск решения** (Tools, Solver) и заполним открывшееся диалоговое окно **Поиск решения** (Solver), как показано на рис. 5.22.

Не забудьте в диалоговом окне **Параметры поиска решения** (Solver) установить флажок **Линейная модель** (Assume Linear Model). После нажатия кнопки **Выполнить** (Solve) средство поиска решений найдет оптимальное решение (рис. 5.23).

Заметим, что флажок **Формулы** диалогового окна **Параметры** (Options), открываемого командой **Сервис, Параметры** (Tools, Options), обеспечивает отображение формул в ячейках, если они там находятся (рис. 5.21).

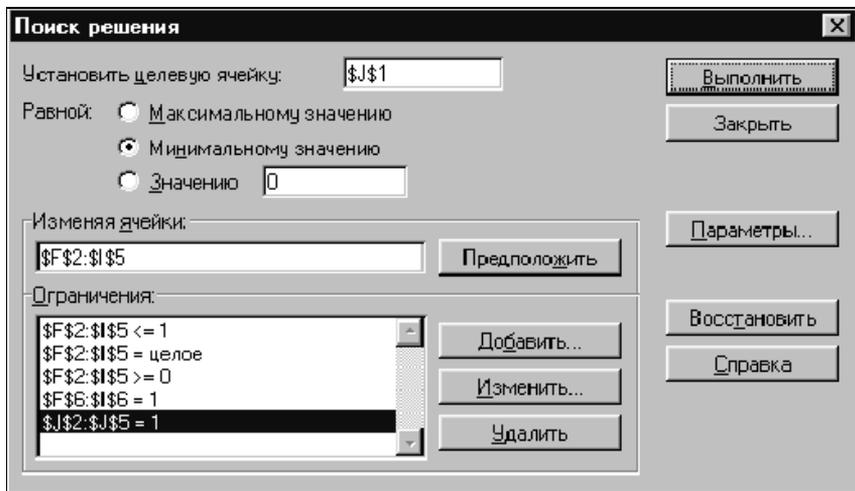


Рис. 5.22. Диалоговое окно **Поиск решения** задачи о назначениях

	A	B	C	D	E	F	G	H	I	J
1	1	4	6	3		Стоимость работ =				18
2	9	10	7	9		1	0	0	0	1
3	4	5	11	7		0	0	1	0	1
4	8	7	8	5		0	1	0	0	1
5						0	0	0	1	1
6						1	1	1	1	

Рис. 5.23. Оптимальный план работ в задаче о назначениях

5.5. Уравнение регрессии

В данном разделе будет рассмотрена задача построения регрессионной модели. В разделе 5.5.1 с помощью средства поиска решений будет решена задача нахождения уравнения регрессии для одной зависимой и одной независимой переменных. Хотя рассмотренная модель имеет очень специфический вид, описанный подход позволяет исследовать любое уравнение регрессии. В разделе 5.5.2 приведены функции рабочего листа, непосредственно вычисляющие различные характеристики линейного уравнения регрессии, а в разделе 5.5.3 — экспоненциального уравнения регрессии, которые позво-

ляют значительно упростить процедуру регрессионного анализа для этих наиболее часто встречающихся на практике моделей.

5.5.1. Общий подход к построению уравнения регрессии на примере линейной модели

Рассмотрим, как решается задача нелинейной оптимизации с помощью средства поиска решений на примере построения линейного уравнения регрессии. Имеются две наблюдаемые величины x и y , например, объем реализации фирмы, торгующей поддержанными автомобилями, за шесть недель ее работы. Значения этих наблюдаемых величин приведены на рис. 5.24, где x — отчетная неделя, а y — объем реализации за эту неделю.

	A	B	C	D	E	F
	x	y	Теор.значе- ние y	m	b	
1						
2	1	7	7	1.885714	5.4	
3	2	9	9	1.885714	5.4	1.77143
4	3	12	11			
5	4	13	13			
6	5	14	15			
7	6	17	17			

Рис. 5.24. Исходные данные для построения линейной модели

Необходимо построить линейную модель $y = mx + b$, наилучшим образом описывающую наблюдаемые значения. Обычно m и b подбираются так, чтобы минимизировать сумму квадратов разностей между наблюдаемыми и теоретическими значениями зависимой переменной y , т. е. минимизировать

$$z = \sum_{i=1}^n (y_i - m x_i - b)^2,$$

где n — число наблюдений (в данном случае $n = 6$).

Для решения этой задачи отведем под переменные m и b ячейки D3 и E3, соответственно, а в ячейку F3 введем минимизируемую функцию

```
{=СУММКВРАЗН (B2 : B7 ; E3 + D3 * A2 : A7) }
```

Функция СУММКВРАЗН (SUMSQ) вычисляет сумму квадратов разностей для элементов указанных массивов.

Теперь выберем команду **Сервис, Поиск решения** (Tools, Solver) и заполним открывшееся диалоговое окно **Поиск решения** (Solver), как показано на рис. 5.25.

Отметим, что на переменные m и b ограничения не налагаются. В результате вычислений средство поиска решений найдет: $m = 1,88571$ и $b = 5,400$.

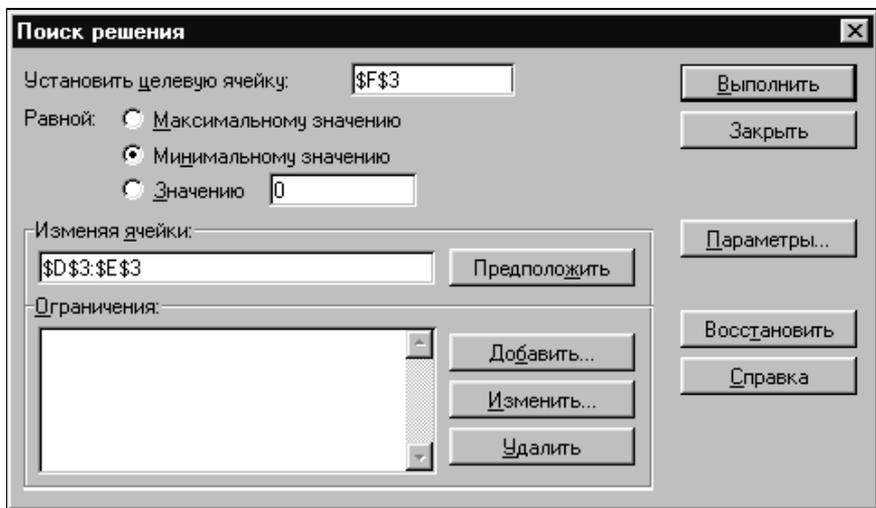


Рис. 5.25. Диалоговое окно Поиск решения для расчета уравнения регрессии

5.5.2. Функции рабочего листа для уравнения линейной регрессии

Параметры m и b линейной модели $y = mx + b$ из предыдущего раздела можно определить с помощью функций НАКЛОН (SLOPE) и ОТРЕЗОК (INTERCEPT).

Функция НАКЛОН (SLOPE) определяет коэффициент наклона линейного тренда.

Синтаксис:

НАКЛОН(известные_значения_y; известные_значения_x)

Функция ОТРЕЗОК (INTERCEPT) определяет точку пересечения линии линейного тренда с осью ординат.

Синтаксис:

ОТРЕЗОК(известные_значения_x; известные_значения_y)

Аргументы функций НАКЛОН (SLOPE) и ОТРЕЗОК (INTERCEPT):

известные_значения_y Массив известных значений зависимой наблюдаемой величины

известные_значения_x Массив известных значений независимой наблюдаемой величины. Если аргумент известные_значения_x

опущен, то предполагается, что это массив {1; 2; 3;...} такого же размера, как и аргумент известные_значения_y

Функции НАКЛОН и ОТРЕЗОК вычисляются по следующим формулам:

$$m = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2},$$

$$b = \bar{y} - m \bar{x},$$

$$\text{где } \bar{x} = \frac{\sum_{i=1}^n x_i}{n}, \quad \bar{y} = \frac{\sum_{i=1}^n y_i}{n}.$$

В ячейках D2 и E2 (рис. 5.24) найдены m и b , соответственно, по формулам:

=НАКЛОН (B2:B7;A2:A7)

=ОТРЕЗОК (B2:B7;A2:A7)

Коэффициенты m и b можно найти и другим способом. Постройте точечный график по диапазону ячеек A2:B7, выделите точки графика двойным щелчком, а затем щелкните их правой кнопкой мыши. В раскрывшемся контекстном меню выберите команду **Линии тренда** (Trendline) (рис. 5.26).

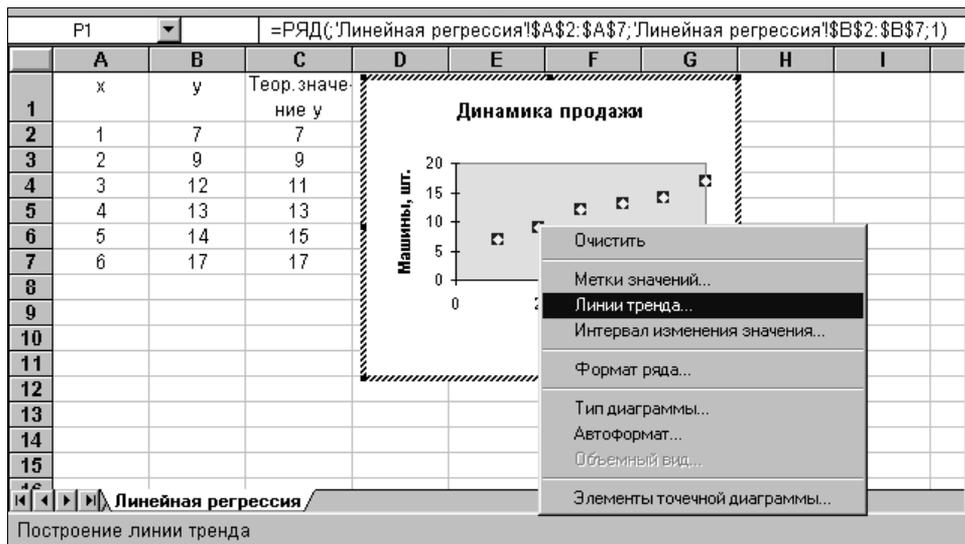


Рис. 5.26. Начало построения линии тренда

В диалоговом окне **Линия тренда** (Trendline) на вкладке **Тип** (Type) в группе **Построение линии тренда (аппроксимация и сглаживание)** (Trend/Regression type) выберите параметр **Линейная** (Linear) (рис. 5.27), а на вкладке **Параметры** (Options) установите флажки **Показывать уравнение на диаграмме** (Display Equation on Chart) и **Поместить на диаграмму величину достоверности аппроксимации (R²)** (Display R-squared) (т. е. на диаграмму необходимо поместить значение квадрата коэффициента корреляции) (рис. 5.28).

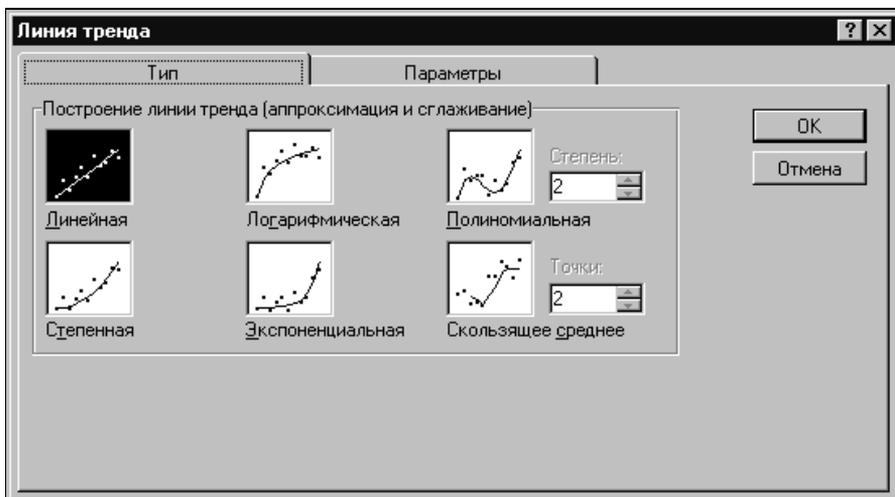


Рис. 5.27. Вкладка **Тип** диалогового окна **Линия тренда**

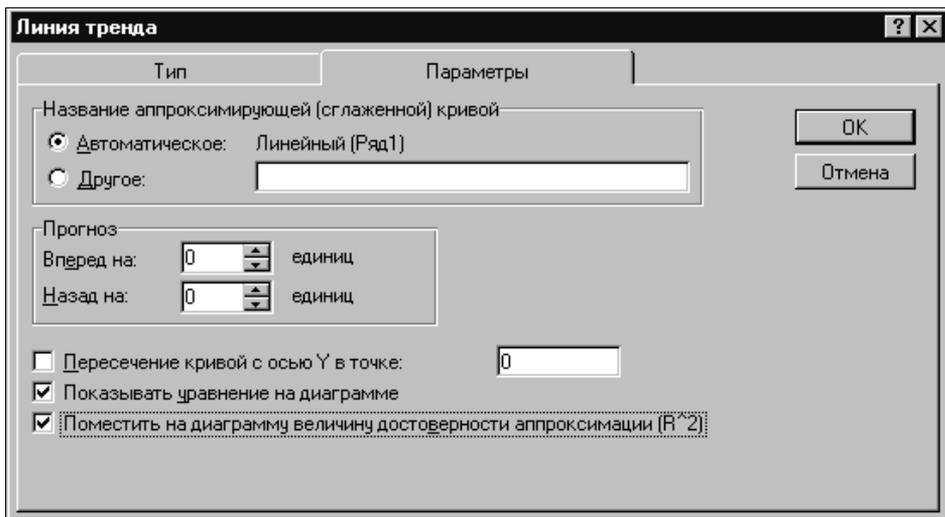


Рис. 5.28. Вкладка **Параметры** диалогового окна **Линия тренда**

По коэффициенту корреляции можно судить о правомерности использования линейного уравнения регрессии. Если он лежит в диапазоне от 0,9 до 1, то данную зависимость можно использовать для предсказания результата. Чем ближе к единице коэффициент корреляции, тем более обоснованно это указывает на линейную зависимость между наблюдаемыми величинами. Если коэффициент корреляции близок к -1 , то это говорит об обратной зависимости между наблюдаемыми величинами.

Флажок **Пересечение кривой с осью Y в точке** (Set Intercept) (рис. 5.28) устанавливается только в случае, если эта точка известна. Например, если этот флажок установлен и в его поле введен 0, это означает, что ищется модель $y = mx$.

Результат выполнения команды **Линии тренда** (Trendline) приведен на рис. 5.29.



Рис. 5.29. График линии тренда

Как видно из рисунка, квадрат коэффициента корреляции равен 0.9723, следовательно, линейная модель может быть использована для предсказания результатов.

На основе найденных коэффициентов уравнения регрессии можно определить теоретическое значение наблюдаемой величины y . Вычислим теоретическое значение y в ячейке C2 (рис. 5.24) при x из A2 по формуле

$$=D\$2*A2+E\$2$$

Однако теоретическое значение y в фиксированной точке можно вычислить и без предварительного определения коэффициентов линейной модели с помощью функции ПРЕДСКАЗ (FORECAST).

Синтаксис:

ПРЕДСКАЗ (t; известные_значения_y; известные_значения_x)

Аргументы:

t	Точка данных, для которой предсказывается значение
известные_значения_y	Массив известных значений зависимой наблюдаемой величины
известные_значения_x	Массив известных значений независимой наблюдаемой величины. Если аргумент <code>известные_значения_x</code> опущен, то предполагается, что это массив {1; 2; 3; ...} такого же размера, как и массив <code>известные_значения_y</code>

Например, теоретическое значение в ячейке C2 (рис. 5.24) можно также определить по формуле

=ПРЕДСКАЗ (A2 ; \$B\$2 : \$B\$7 ; \$A\$2 : \$A\$7)

Функция ТЕНДЕНЦИЯ (TREND) вычисляет значения уравнения линейной регрессии для целого диапазона значений независимой переменной как для одномерного, так и для многомерного уравнения регрессии. Многомерная линейная модель регрессии имеет вид:

$$y = m_1x_1 + \dots + m_nx_n + b.$$

Синтаксис:

ТЕНДЕНЦИЯ (известные_значения_y; известные_значения_x;
новые_значения_x; конст)

Аргументы:

известные_значения_y	Массив известных значений зависимой наблюдаемой величины
известные_значения_x	Массив известных значений независимой наблюдаемой величины. Если аргумент <code>известные_значения_x</code> опущен, то предполагается, что это массив {1; 2; 3; ...} такого же размера, как и массив <code>известные_значения_y</code>
новые_значения_x	Новые значения x, для которых функция ТЕНДЕНЦИЯ возвращает соответствующие значения y
конст	Логическое значение, которое указывает, требуется ли, чтобы константа <i>b</i> была равна 0. Если аргумент <code>конст</code> имеет значение ИСТИНА или опущен, то <i>b</i> вычисляется обычным образом. Если <code>конст</code> имеет значение ЛОЖЬ, то <i>b</i> полагается равным 0

Если строится многомерная линейная модель, то аргументы `известные_значения_x` и `новые_значения_x` должны содержать столбец (или строку) для

каждой независимой переменной. Если аргумент `новые_значения_x` опущен, то предполагается, что он совпадает с аргументом `известные_значения_x`.

Функция `ЛИНЕЙН` (`LINEST`) возвращает массив $\{m_n, \dots, m_1, b\}$ значений параметров уравнения многомерной линейной регрессии.

Синтаксис:

`ЛИНЕЙН(известные_значения_y; известные_значения_x;
конст; статистика)`

Аргументы:

<code>известные_значения_y</code>	Массив известных значений зависимой наблюдаемой величины
<code>известные_значения_x</code>	Массив известных значений независимой наблюдаемой величины. Если аргумент <code>известные_значения_x</code> опущен, то предполагается, что это массив $\{1; 2; 3; \dots\}$ такого же размера, как и <code>известные_значения_y</code>
<code>конст</code>	Логическое значение, которое указывает, требуется ли, чтобы константа <code>b</code> была равна 0. Если аргумент <code>конст</code> имеет значение <code>ИСТИНА</code> или опущен, то <code>b</code> вычисляется обычным образом. Если <code>конст</code> имеет значение <code>ЛОЖЬ</code> , то <code>b</code> полагается равным 0
<code>статистика</code>	Логическое значение, которое указывает, требуется ли вывести дополнительную статистику по регрессии, например, коэффициент корреляции. Если <code>статистика</code> имеет значение <code>ИСТИНА</code> , то функция <code>ЛИНЕЙН</code> возвращает дополнительную регрессионную статистику. Если аргумент <code>статистика</code> имеет значение <code>ЛОЖЬ</code> или опущен, то функция <code>ЛИНЕЙН</code> возвращает только значения коэффициентов

5.5.3. Экспоненциальная модель

Другой часто встречающейся на практике регрессионной моделью является экспоненциальная модель, которая описывается уравнением

$$y = bm^x$$

Значения экспоненциального тренда можно предсказывать с помощью функции `РОСТ` (`GROWTH`).

Синтаксис:

`РОСТ(известные_значения_y; известные_значения_x; новые_значения_x; конст)`

Аргументы:

известные_значения_y	Массив известных значений зависимой наблюдаемой величины
известные_значения_x	Массив известных значений независимой наблюдаемой величины. Если аргумент <code>известные_значения_x</code> опущен, то предполагается, что это массив {1; 2;3; ...} такого же размера, как и <code>известные_значения_y</code>
новые_значения_x	Новые значения x , для которых <code>ТЕНДЕНЦИЯ</code> возвращает соответствующие значения y
конст	Логическое значение, которое указывает, требуется ли, чтобы константа b была равна 0. Если аргумент <code>конст</code> имеет значение <code>ИСТИНА</code> или опущен, то b вычисляется обычным образом. Если <code>конст</code> имеет значение <code>ЛОЖЬ</code> , то b полагается равным 0

Значения параметров экспоненциальной модели определяются с помощью функции `ЛГРФПРИБЛ` (`LOGEST`).

Синтаксис:

`ЛГРФПРИБЛ(известные_значения_y; известные_значения_x; конст; статистика)`

Аргументы:

известные_значения_y	Массив известных значений зависимой наблюдаемой величины
известные_значения_x	Массив известных значений независимой наблюдаемой величины. Если аргумент <code>известные_значения_x</code> опущен, то предполагается, что это массив {1; 2; 3;...} такого же размера, как и <code>известные_значения_y</code>
конст	Логическое значение, которое указывает, требуется ли, чтобы константа b была равна 0. Если аргумент <code>конст</code> имеет значение <code>ИСТИНА</code> или опущен, то b вычисляется обычным образом. Если <code>конст</code> имеет значение <code>ЛОЖЬ</code> , то b полагается равным 0
статистика	Логическое значение, которое указывает, требуется ли вывести дополнительную статистику по регрессии, например, коэффициент корреляции. Если <code>статистика</code> имеет значение <code>ИСТИНА</code> , то функция <code>ЛИНЕЙН</code> возвращает дополнительную регрессионную статистику. Если <code>статистика</code> имеет значение <code>ЛОЖЬ</code> или опущена, то функция <code>ЛИНЕЙН</code> возвращает только значения коэффициентов

Кроме того, одномерную экспоненциальную модель можно построить графически (рис. 5.27). На рис. 5.30 приведены результаты построения экспо-

ннциального уравнения тренда продажи подержанных автомобилей за 7, 8 и 9-ю недели торговли (задача из рис. 5.24).

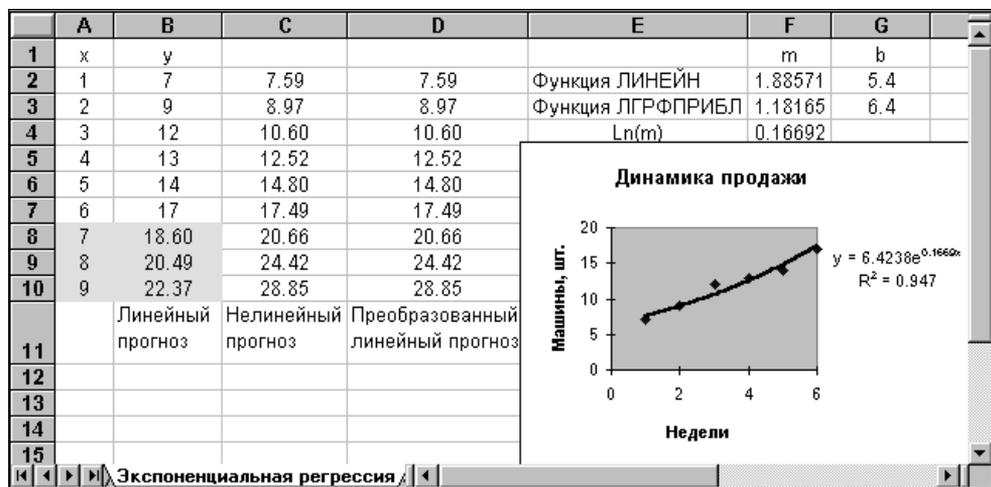


Рис. 5.30. Экспоненциальная линия тренда

В диапазоне ячеек B8:B10 введена формула построения линейного тренда

`{=ТЕНДЕНЦИЯ(B2:B7;A2:A7;A8:A10)}`

В диапазоне ячеек C2:C10 введена формула построения экспоненциального тренда

`{=РОСТ(B2:B7;A2:A7;A2:A10)}`

Линейный и экспоненциальный тренды тесно связаны между собой. В диапазон ячеек D2:D10 введена формула

`{=ЭКСП(ТЕНДЕНЦИЯ(LN(B2:B7);A2:A7;A2:A10))}`

Как видно из рисунка, значения в диапазонах C2:C10 и D2:D10 совпадают. В диапазоны ячеек F2:G2 и F3:G3 введены формулы

`{=ЛИНЕЙН(B2:B7;A2:A7)}`

`{=ЛГРФПРИВЛ(B2:B7;A2:A7)}`

для определения параметров линейной и экспоненциальной моделей.

Квадрат коэффициента корреляции экспоненциальной модели равен 0,947 (рис. 5.30) и меньше квадрата коэффициента корреляции линейной модели

(= 0,9923) (рис. 5.29). Таким образом, в данном примере линейная модель более достоверно описывает зависимость между наблюдаемыми величинами.

5.6. Транспортная задача с фиксированными доплатами

Рассмотрим пример нелинейной задачи, которая после введения вспомогательных переменных становится линейной. Пусть, как обычно в транспортной задаче, имеется m пунктов производства некоторого товара с объемом производства a_i в i -м пункте и n пунктов его потребления с объемом потребления b_j в j -м пункте. Требуется найти величины x_{ij} — объемы перевозок из пункта i в пункт j , удовлетворяющие естественным транспортным ограничениям

$$\begin{aligned} \sum_{i=1}^m x_{ij} &= b_j, \quad j \in [1, n], \\ \sum_{j=1}^n x_{ij} &= a_i, \quad i \in [1, m], \\ x_{ij} &\geq 0, \quad i \in [1, m], \quad j \in [1, n] \end{aligned} \quad (5.3)$$

и минимизирующие функцию

$$z = \sum_{i=1}^m \sum_{j=1}^n c_{ij}(x_{ij}),$$

где

$$c_{ij}(x_{ij}) = \begin{cases} 0, & x_{ij} = 0, \\ c_{ij} x_{ij} + d_{ij}, & x_{ij} > 0 \end{cases}$$

(c_{ij} — затраты на перевозку единицы товара из пункта i в пункт j).

Под d_{ij} можно понимать, например, плату за аренду транспортных средств, не зависящую от загрузки. Очевидно, что если все d_{ij} равны нулю, то данная задача превращается в обычную транспортную задачу. В противном случае эта задача в силу разрывности функции цели вообще выпадает из рамок линейного программирования. Однако путем введения дополнительных целочисленных переменных ее удастся свести к частично целочисленной задаче линейного программирования. Пусть

$$M = \max\{a_i, b_j, i \in [1, m], j \in [1, n]\}$$

Рассмотрим задачу минимизации функции

$$z = \sum_{i=1}^m \sum_{j=1}^n c_{ij}x_{ij} + d_{ij}y_{ij}$$

при ограничениях (5.3) и дополнительных условиях

$$y_{ij} \in \{0, 1\},$$

$$x_{ij} - My_{ij} \leq 0 \quad (5.4)$$

при $i \in [1, m], j \in [1, n]$.

Легко видеть, что частично целочисленная задача эквивалентна исходной задаче.

Рассмотрим пример решения частично целочисленной задачи. Пусть имеется два пункта производства с объемами производства по 200 единиц товара и три пункта потребления с объемами потребления 100, 200 и 100 единиц.

Затраты на перевозку единицы груза c_{ij} находятся в ячейках A1:C2, а плата за аренду транспортных средств d_{ij} — в ячейках E1:G2. Значения объемов производства введем в ячейки D1:D2. Значения объемов потребления введем в ячейки A1:C3. Под объемы перевозок x_{ij} отведем ячейки A5:C6, а под дополнительные целочисленные переменные y_{ij} — ячейки E5:G6. Под функцию цели отведем ячейку D7, в которую введем формулу

$$=СУММПРОИЗВ(A1:C2;A5:C6) + СУММПРОИЗВ(E1:G2;E5:G6)$$

В ячейки A7:C7 введем формулы

$$=СУММ(A5:A6)$$

$$=СУММ(B5:B6)$$

$$=СУММ(C5:C6)$$

а в ячейки D5:D6 — формулы

$$=СУММ(A5:C5)$$

$$=СУММ(A6:C6)$$

которые представляют собой левые части формул транспортных ограничений. В диапазон ячеек A9:C10 введем формулу

$$\{=A5:C6-200*E5:G6\}$$

которая представляет собой левую часть ограничений (5.4) при $M = 200$.

Заполним диалоговое окно **Поиск решения** (Solver), как показано на рис. 5.31. Заметим, что в этом окне в поле **Ограничения** (Subject to the Constraints) не поместилось еще одно ограничение: $\$E\$5:\$G\$6 \geq 0$.

В диалоговом окне **Параметры поиска решения** (Solver Options) (рис. 5.6) не забудьте установить флажок **Линейная модель** (Assume Linear Model). После нажатия на кнопку **Выполнить** (Solve) средство поиска решений найдет оптимальное решение (рис. 5.32).

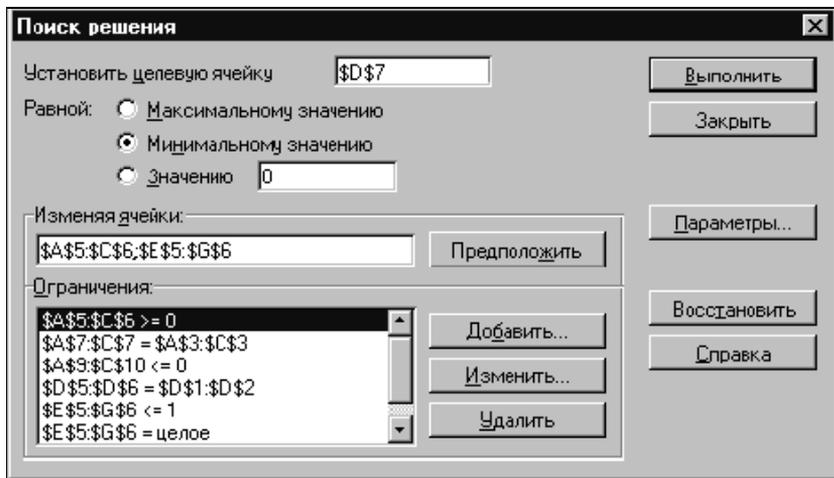


Рис. 5.31. Диалоговое окно **Поиск решения** для транспортной задачи с фиксированными доплатами

	A	B	C	D	E	F	G
1	1	2	3	200	1	2	2
2	2	4	3	200	3	1	1
3	100	200	100				
4							
5	0	200	0	200	0	1	0
6	100	0	100	200	1	0	1
7	100	200	100	906			
8							
9	0	0	0				
10	-100	0	-100				
11							
12							

Рис. 5.32. Решение транспортной задачи с фиксированными доплатами

5.8. Упражнения

(а) **Транспортная задача.** Имеются n пунктов производства и m пунктов распределения продукции. Стоимость перевозки единицы продукции с i -го пункта производства в j -й центр распределения c_{ij} приведена в таблице, где под строкой понимается пункт производства, а под столбцом — пункт рас-

пределения. Кроме того, в этой таблице в i -й строке указан объем производства в i -м пункте производства, а в j -м столбце указан спрос в j -м центре распределения. Необходимо составить план перевозок по доставке требуемой продукции в пункты распределения, минимизирующий суммарные транспортные расходы.

(b) **Задача о назначениях.** Имеются n рабочих и m видов работ. Стоимость c_{ij} выполнения i -м рабочим j -й работы приведена в таблице, где рабочему соответствует строка, а работе — столбец. Необходимо составить план работ так, чтобы все работы были выполнены, каждый рабочий был занят только на одной работе, а суммарная стоимость выполнения всех работ была бы минимальной.

(c) **Линейная оптимизационная задача.**

(d) **Система нелинейных уравнений.** Найти все решения системы нелинейных уравнений.

(e) **Уравнение регрессии.** Построить линейную модель для двух наблюдаемых величин (например, объем реализованных фирмой подержанных автомобилей за указанное число недель).

Вариант 1

(a) Транспортная задача:

Стоимость перевозки единицы продукции				Объемы производства
1	3	4	5	20
5	2	10	3	30
3	2	1	4	50
6	4	2	6	20
Объемы потребления	30	20	60	15

(b) Задача о назначениях:

Стоимость выполнения работ					
Рабочие	3	6	2	5	11
	1	2	7	11	3

5	12	11	9	1
2	4	2	10	5

Виды работ

(с) Линейная оптимизационная задача:

Предприятие электронной промышленности выпускает две модели радиоприемников, причем каждая модель производится на отдельной технологической линии. Суточный объем производства первой линии — 60 изделий, второй линии — 75 изделий. На радиоприемник первой модели расходуется 10 однотипных элементов электронных схем, на радиоприемник второй модели — 8 таких же элементов. Максимальный суточный запас используемых элементов равен 800 единицам. Прибыль от реализации одного радиоприемника первой и второй моделей равна 30 и 20 долларов, соответственно. Определить оптимальный суточный объем производства первой и второй моделей.

(d) Система нелинейных уравнений:

$$2x^2 + 5y^2 = 3,$$

$$5x + 9y = 3.$$

(e) Уравнение регрессии:

Неделя	1	2	3	4	5	6	7	8
Количество машин	13	19	26	30	37	44	49	55

Вариант 2

(a) Транспортная задача:

Стоимость перевозки единицы продукции				Объемы производства
2	7	7	6	20
1	1	1	2	50
5	5	3	1	10
2	8	1	4	20

3	2	1	5	17
40	30	20	20	

**Объемы
потребления**

(b) Задача о назначениях:

Стоимость выполнения работ

Рабочие

1	3	6	5	7
5	2	7	8	3
3	5	1	9	2
6	4	2	10	5

Виды работ

(c) Линейная оптимизационная задача:

Процесс изготовления двух видов промышленных изделий состоит в последовательной обработке каждого из них на трех станках. Время использования этих станков для производства данных изделий ограничено 10-ю часами в сутки. Время обработки и прибыль от продажи одного изделия каждого вида приведены в табл. 5.5. Найти оптимальный объем производства изделий каждого вида.

Таблица 5.5. *Время обработки и прибыль от продажи одного изделия*

Изделие	Время обработки одного изделия, мин			Удельная прибыль, \$
	Станок 1	Станок 2	Станок 3	
1	10	6	8	2
2	5	20	15	3

(d) Система нелинейных уравнений:

$$3x^2 + 4y^2 = 4,$$

$$3x + 4y = 2.$$

радиорекламы. Определить оптимальное распределение ежемесячно отпускаемых средств между радио- и телерекламой.

(d) Система нелинейных уравнений:

$$5x^2 + 2y^2 = 4,$$

$$2x + 7y = 1.$$

(e) Уравнение регрессии:

Неделя	1	2	3	4	5	6	7	8	9
Количество машин	7	17	19	28	35	42	41	52	57

Вариант 4

(a) Транспортная задача:

	Стоимость перевозки единицы продукции				Объемы производства
	5	1	7	6	30
	1	5	8	1	40
	5	6	3	3	10
	2	5	1	4	18
	3	7	9	1	10
Объемы потребления	20	40	30	20	

(b) Задача о назначениях:

Стоимость выполнения работ

Рабочие

8	6	2	5
5	2	9	8
3	8	1	9
1	4	2	3
3	7	10	5

Виды работ

(с) Линейная оптимизационная задача:

Фирма производит два вида продукции — А и В. Объем сбыта продукции А составляет не менее 60% общего объема реализации продукции обоих видов. Для изготовления продукции А и В используется одно и то же сырье, суточный запас которого ограничен величиной 100 кг. Расход сырья на единицу продукции А составляет 2 кг, а на единицу продукции В — 4 кг. Цены продукции А и В равны 20 и 40 долларов, соответственно. Определить оптимальное распределение сырья для изготовления продукции А и В.

(d) Система нелинейных уравнений:

$$4x^2 + 5y^2 = 3,$$

$$5x + 3y = 1.$$

(e) Уравнение регрессии:

Неделя	1	2	3	4	5	6	7	8	9
Количество машин	12	21	30	36	44	54	61	70	78

Вариант 5

(a) Транспортная задача:

Стоимость перевозки единицы продукции

Объемы производства

3	9	4	5	40
1	8	5	3	10
7	2	1	4	30
2	4	10	6	20

Объемы потребления

50 10 35 10

(b) Задача о назначениях:

Стоимость выполнения работ

Рабочие

10	8	6	2	7
6	2	9	8	3
3	7	1	10	5
9	10	2	3	4

Виды работ

(с) Линейная оптимизационная задача:

Фирма выпускает ковбойские шляпы двух фасонов (А и В). Трудоемкость изготовления шляпы фасона А вдвое выше трудоемкости изготовления шляпы фасона В. Если бы фирма выпускала только шляпы фасона А, суточный объем производства мог бы составить 500 шляп. Суточный объем сбыта шляп обоих фасонов ограничен диапазоном от 150 до 200 штук. Прибыль от продажи шляпы фасона А равна \$8, а фасона В — \$5. Определить, какое количество шляп каждого фасона следует изготовить, чтобы максимизировать прибыль.

(d) Система нелинейных уравнений:

$$5x^2 + 6y^2 = 3,$$

$$7x + 3y = 1.$$

(e) Уравнение регрессии:

Неделя	1	2	3	4	5	6	7	8	9	10	11
Количество машин	12	17	23	30	35	40	48	54	59	65	72

Вариант 6

(a) Транспортная задача:

Стоимость перевозки единицы продукции				Объемы производства
6	1	3	1	20
3	4	5	8	30
5	9	3	2	20
2	4	8	4	20
3	2	1	5	17
Объемы потребления	50	30	20	20

(b) Задача о назначениях:

Стоимость выполнения работ

Рабочие	9	3	2	7
	5	4	9	8
	7	8	1	10
	1	9	10	3
	2	7	8	5

Виды работ

(c) Линейная оптимизационная задача:

Изделия четырех типов проходят последовательную обработку на двух станках. Время обработки одного изделия каждого типа на каждом из станков приведено в табл. 5.6.

Таблица 5.6. *Время обработки одного изделия*

Станок	Время обработки одного изделия, ч			
	Тип 1	Тип 2	Тип 3	Тип 4
1	2	3	4	2
2	3	2	1	2

Затраты на производство одного изделия каждого типа определяются как величины, прямо пропорциональные времени использования станков (в машино-часах). Стоимость машино-часа составляет 10 и 15 долларов для станков 1 и 2, соответственно. Допустимое время использования станков для обработки изделий всех типов ограничено следующими значениями: 500 машино-часов для станка 1 и 380 машино-часов для станка 2. Цены изделий типов 1, 2, 3 и 4 равны 65, 70, 55 и 45 долларов, соответственно. Составить план производства, максимизирующий чистую прибыль.

(d) Система нелинейных уравнений:

$$3x^2 + 5y^2 = 3,$$

$$5x + 2y = 2.$$

(e) Уравнение регрессии:

Неделя	1	2	3	4	5	6	7	8	9
Количество машин	10	18	22	28	34	39	46	51	54

Вариант 7

(a) Транспортная задача:

Стоимость перевозки единицы продукции

**Объемы
производства**

5	9	4	5	30
1	5	5	6	20
2	2	10	4	30
3	7	2	6	40
Объемы потребления	20	50	20	35

(b) Задача о назначениях:

Стоимость выполнения работ

Рабочие

9	4	6	2	10
6	2	10	8	4
3	7	1	10	5
7	10	5	3	9

Виды работ

(c) Линейная оптимизационная задача:

Завод выпускает изделия трех моделей (I, II и III). Для их изготовления используются два вида ресурсов (A и B), запасы которых составляют 4000 и 6000 единиц. Расходы ресурсов на одно изделие каждой модели приведены в табл. 5.7.

Таблица 5.7. Расходы ресурсов

Ресурс	Расход ресурса на одно изделие данной модели		
	I	II	III
A	2	3	5
B	4	2	7

Трудоемкость изготовления изделия модели I вдвое больше, чем изделия модели II, и втрое больше, чем изделия модели III. Численность рабочих завода позволяет выпускать 1500 изделий модели I. Анализ условий сбыта показывает, что минимальный спрос на продукцию завода составляет 200, 200 и 150 изделий моделей I, II и III, соответственно. Однако соотношение выпуска

изделий моделей I, II и III должно быть равно 3:2:5. Удельная прибыль от реализации изделий моделей I, II и III составляет 30, 20 и 50 долларов, соответственно. Определить выпуск изделий, максимизирующий прибыль.

(d) Система нелинейных уравнений:

$$7x^2 + 6y^2 = 3,$$

$$5x + 3y = 2.$$

(e) Уравнение регрессии:

Неделя	1	2	3	4	5	6	7	8
Количество машин	12	18	25	32	40	46	53	60

Вариант 8

(a) Транспортная задача:

Стоимость перевозки единицы продукции

Объемы производства

7	1	3	2	30
8	4	5	8	20
5	2	3	7	10
5	5	8	4	27
1	9	7	5	30

Объемы потребления

30 40 50 10

(b) Задача о назначениях:

Стоимость выполнения работ

Рабочие

10	3	2	4
5	9	10	8
7	8	1	9
11	10	9	12
2	7	8	10

Виды работ

(c) Линейная оптимизационная задача:

Требуется распределить имеющиеся денежные средства по четырем альтернативным вариантам. Игра имеет три исхода. В табл. 5.8 приведены размеры выигрыша (или проигрыша) на каждый доллар, вложенный в соответствующий альтернативный вариант, для каждого из трех исходов. У игрока имеется \$500, причем использовать их в игре можно только один раз. Точный исход игры заранее неизвестен. Учитывая эту неопределенность, распределить деньги так, чтобы максимизировать минимальную отдачу от этой суммы.

Таблица 5.8. Возможные выигрыши и проигрыши

Исход	Выигрыш или проигрыш на каждый доллар, вложенный в данный вариант			
	1	2	3	4
1	-3	4	-7	15
2	5	-3	9	4
3	3	-9	10	-10

(
d) Система нелинейных уравнений:

$$5x^2 + 6y^2 = 3,$$

$$3x + 2y = 2.$$

(e) Уравнение регрессии:

Неделя	1	2	3	4	5	6	7	8	9
Количество машин	14	23	30	39	45	54	63	70	78

Вариант 9

(a) Транспортная задача:

	Стоимость перевозки единицы продукции				Объемы производства
7	9	1	5	20	
2	7	5	6	30	
3	5	10	8	40	
3	7	4	5	30	
Объемы потребления	40	30	30	42	

(b) Задача о назначениях:

Стоимость выполнения работ

Рабочие	5	4	12	2	10
	6	5	10	8	4
	3	7	11	10	8
	10	1	5	11	9

Виды работ

(c) Линейная оптимизационная задача:

Небольшая фирма выпускает два типа автомобильных деталей (А и В). Для этого она закупает литье, подвергаемое токарной обработке, сверловке и шлифовке. Данные, характеризующие производительность станочного парка фирмы, приведены в табл. 5.9.

Таблица 5.9. Производительность станков

Станки	Деталь А, шт./ч	Деталь В, шт./ч
Токарный	25	40
Сверлильный	28	35
Шлифовальный	35	25

Каждая отливка, из которой изготавливают деталь А, стоит \$2. Стоимость отливки для детали В — \$3. Продажная цена деталей равна, соответственно, 5 и 6 долларов. Стоимость часа станочного времени составляет по трем типам используемых станков 20, 14 и 17,5 долларов, соответственно. Предполагая, что можно выпускать для продажи любую комбинацию деталей А и В, нужно найти план выпуска продукции, максимизирующий прибыль.

(d) Система нелинейных уравнений:

$$3x^2 + 2y^2 = 2,$$

$$2x + 7y = 3.$$

(e) Уравнение регрессии:

Неделя	1	2	3	4	5	6	7	8	9	10	11
Количество машин	15	22	26	33	40	45	51	58	63	69	78

Вариант 10

(a) Транспортная задача:

Стоимость перевозки единицы продукции

Объемы производства

5	9	3	10	10
3	10	5	9	30
7	2	3	8	20
8	5	11	2	32
5	9	10	5	20

**Объемы
потребления**

50 10 30 10

(b) Задача о назначениях:

Стоимость выполнения работ

Рабочие

5	12	2	7
10	9	7	12
7	8	11	9
2	10	9	13
12	7	8	3

Виды работ

(c) Линейная оптимизационная задача:

Бройлерное хозяйство птицеводческой фермы насчитывает 20 000 цыплят, которые выращиваются до 8-недельного возраста и после соответствующей обработки поступают в продажу. Хотя недельный рацион цыплят зависит от их возраста, в дальнейшем будем считать, что в среднем (за 8 недель) он составляет 1 фунт.

Для того чтобы цыплята достигли к восьмой неделе необходимого веса, кормовой рацион должен удовлетворять определенным требованиям по питательности. Этим требованиям могут соответствовать смеси различных видов кормов или ингредиентов. Ограничим наше рассмотрение только тремя ингредиентами: известняком, зерном и соевыми бобами. В табл. 5.10 приведены данные, характеризующие содержание (по весу) питательных веществ в каждом из ингредиентов и удельную стоимость каждого ингредиента.

Таблица 5.10. Содержание (по весу) питательных веществ

Ингредиент	Содержание питательных веществ, фунт/фунт ингредиента			Стоимость, \$/фунт
	Кальций	Белок	Клетчатка	
Известняк	0,38	—	—	0,04
Зерно	0,001	0,09	0,02	0,15
Соевые бобы	0,002	0,5	0,08	0,4

Смесь должна содержать:

- Не менее 0,8%, но и не более 1,2% кальция
- Не менее 22% белка
- Не более 5% клетчатки

Необходимо определить количество каждого из трех ингредиентов, образующих смесь минимальной стоимости, при соблюдении требований к общему расходу кормовой смеси и ее питательности.

(d) Система нелинейных уравнений:

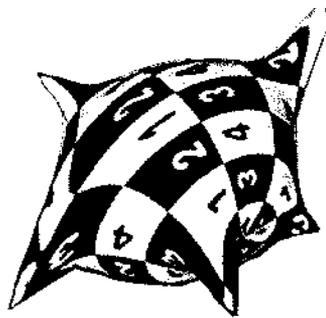
$$5x^2 + y^2 = 3,$$

$$3x + 5y = 2.$$

(e) Уравнение регрессии:

Неделя	1	2	3	4	5	6	7	8
Количество машин	9	15	24	29	38	46	52	58

Глава 6



Сводные таблицы

Одним из наиболее мощных средств Excel по работе с базами данных являются сводные таблицы. Они полезны как для анализа, так и для обобщения информации, хранящейся в базе данных, на рабочих листах, во внешних файлах. Сводные таблицы являются динамическими объектами, позволяющими выводить информацию с различной степенью детализации.

Проанализируем базу данных, представленную на рис. 6.1.

	A	B	C	D	E	F
1	Фамилия	Тип вклада	Размер вклада	Отделение банка	Примечание	
2	Сидоров	Текущий	23100000	Северное	Временно выехал	
3	Фролов	Депозит	4159000	Центральное		
4	Сидорчук	Текущий	123	Западное		
5	Вагнер	Текущий	3888000	Северное	Сменил адрес	
6	Петров	Текущий	21758	Северное		
7	Иванов	Депозит	559000	Центральное		
8	Григорьев	Текущий	5000500	Центральное		
9	Чернов	Текущий	73000000	Северное		
10	Кравцов	Текущий	1760300	Северное	Перевел в другое отделение	
11	Рябов	Текущий	77200000	Западное		
12	Степанов	Депозит	45300000	Центральное		
13	Иванов	Депозит	50959000	Центральное		
14	Круглов	Депозит	12859000	Центральное		
15	Федоров	Текущий	34500000	Северное		
16	Тараканов	Текущий	47600000	Северное		
17	Сидоров	Депозит	29800000	Западное		
18	Козлов	Депозит	67300000	Западное		

Рис. 6.1. База данных

Эта база данных создана на рабочем листе с именем База данных с помощью сводной таблицы для нахождения суммы вкладов, хранящихся в различных отделениях банка.

Выполним команду **Данные, Сводная таблица** (Data, Pivot Table). В первом диалоговом окне **Мастер сводных таблиц** (PivotTable Wizard) (рис. 6.2) установим переключатель в положение, показывающее, откуда берутся данные для сводной таблицы:

- В списке или базе данных Microsoft Excel** — если данные берутся с одного рабочего листа
- Во внешнем источнике данных** — если данные берутся из внешней базы данных
- В нескольких диапазонах консолидации** — если данные берутся с нескольких рабочих листов
- В другой сводной таблице** — если сводная таблица создается на основе данных другой сводной таблицы

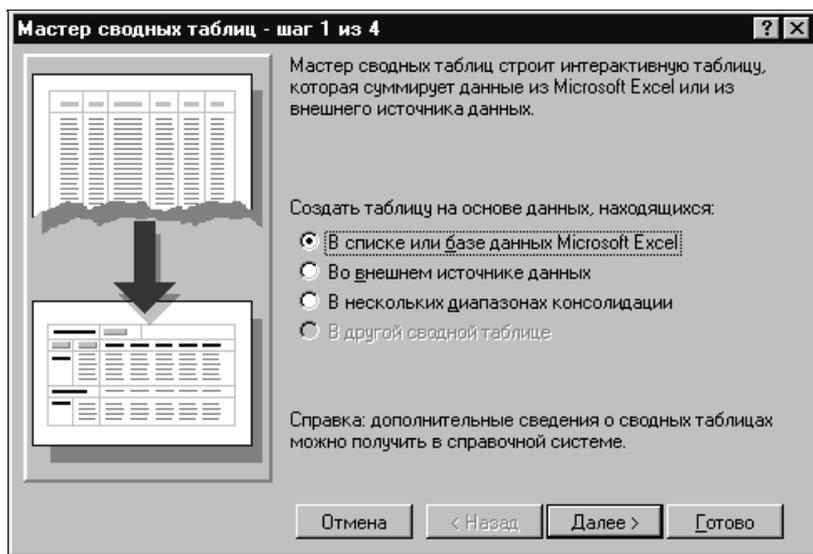


Рис. 6.2. Первое диалоговое окно **Мастер сводных таблиц**

В рассматриваемом примере нужно установить переключатель в положение **В списке или базе данных Microsoft Excel**.

В поле **Диапазон** (Range) второго диалогового окна **Мастер сводных таблиц** (PivotTable Wizard) укажите, на основе какого диапазона строится сводная таблица (рис. 6.3). В рассматриваемом примере необходимо указать диапазон:

'База данных'!\$A\$1:\$E\$18

В третьем диалоговом окне **Мастер сводных таблиц** (PivotTable Wizard) необходимо указать структуру сводной таблицы (рис. 6.4).

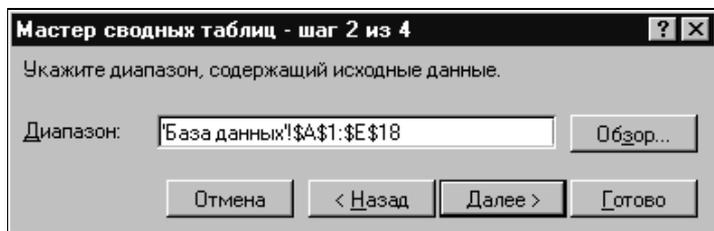


Рис. 6.3. Второе диалоговое окно **Мастер сводных таблиц**

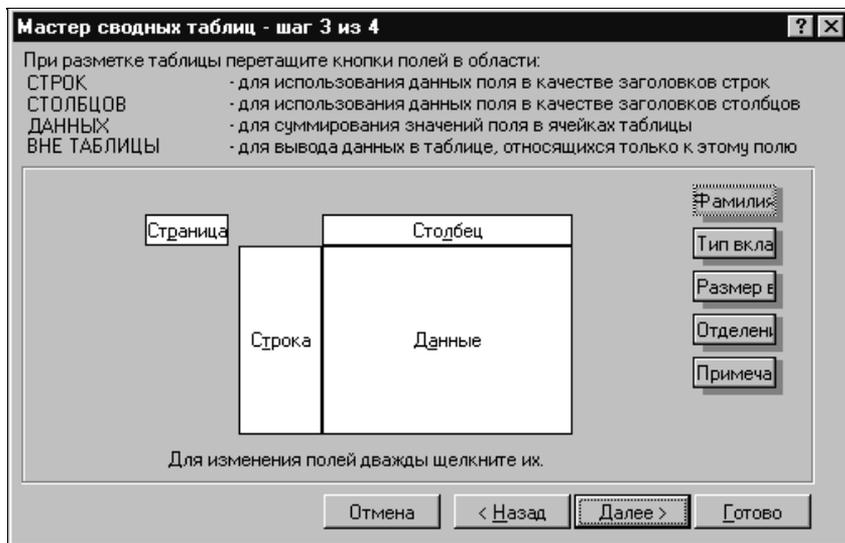


Рис. 6.4. Третье диалоговое окно **Мастер сводных таблиц**

Поля базы данных, на основе которой строится сводная таблица, представлены в этом окне в виде кнопок с названиями этих полей. Надо так перетащить их в соответствующие области, чтобы сводная таблица приобрела необходимую структуру. В окне имеются четыре области:

- Строка** (Row) — для использования данных поля, расположенного в этой области, в качестве заголовков строк
- Столбец** (Column) — для использования данных поля, расположенного в этой области, в качестве заголовков столбцов
- Данные** (Data Area) — для суммирования значений поля, расположенного в этой области, в ячейках сводной таблицы
- Страница** (Page) — для обеспечения возможности вывода данных сводной таблицы, относящихся только к полю, расположенному в этой области

В рассматриваемом примере создадим рабочую таблицу, состоящую из строк с названиями отделений с выводом суммы вкладов, находящихся в них (рис. 6.5).

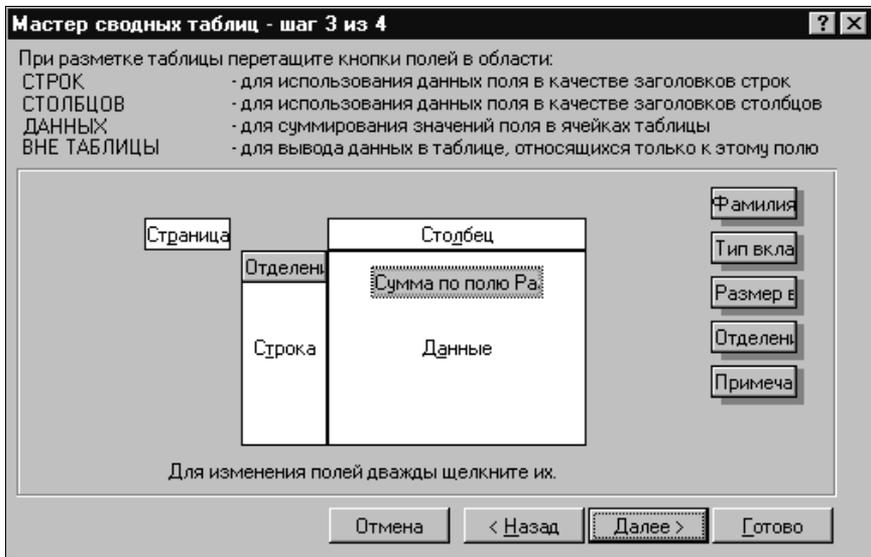


Рис. 6.5. Третье диалоговое окно **Мастер сводных таблиц** после перетаскивания кнопок полей в области сводной таблицы

Двойной щелчок на кнопке, перемещенной в область **Данные** (Data), открывает диалоговое окно **Вычисление поля сводной таблицы** (PivotTable Field), позволяющее выбрать правило, по которому подводятся итоги в сводной таблице (рис. 6.6).

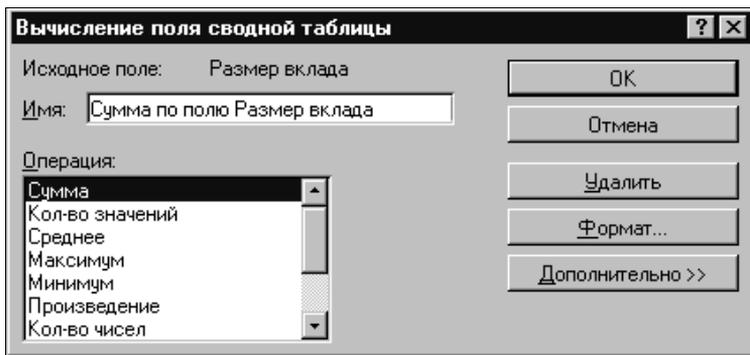


Рис. 6.6. Диалоговое окно **Вычисление поля сводной таблицы**

Допустимыми операциями подведения итогов являются: сумма, количество значений, среднее арифметическое, максимальное и минимальное значение,

произведение, количество чисел, несмешанное и смешанное отклонение, несмешанная и смешанная дисперсия. В нашем примере выберем в качестве допустимой операции сумму.

В четвертом диалоговом окне **Мастер сводных таблиц** (PivotTable Wizard) можно задать некоторые параметры, определяющие вид сводной таблицы (рис. 6.7).

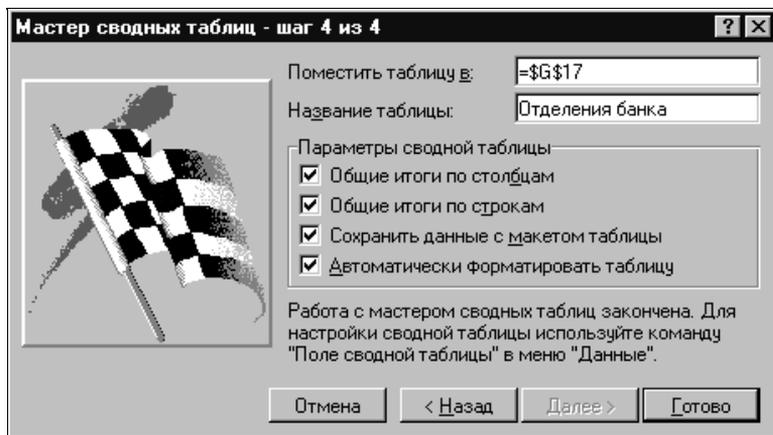


Рис. 6.7. Четвертое диалоговое окно **Мастер сводных таблиц**

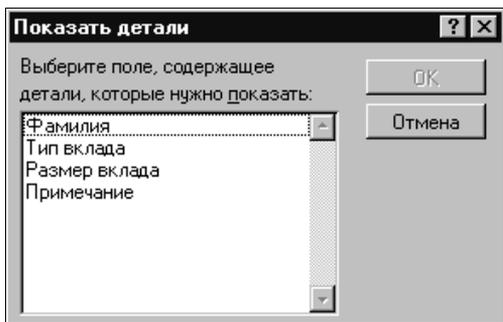
- В поле ввода **Поместить таблицу в** (PivotTable Starting Cell) указывается адрес верхней левой ячейки таблицы. Если это поле оставить пустым, то в текущую рабочую книгу будет вставлен новый рабочий лист со сводной таблицей, левый верхний угол которой расположен в ячейке A1
- В поле ввода **Название таблицы** (PivotTable Name) указывается название сводной таблицы. По умолчанию сводные таблицы называются: СводнаяТаблица1 (PivotTable1), СводнаяТаблица2 (PivotTable2) и т. д.
- Устанавливая флажок **Общие итоги по столбцам** (Grand Totals for Columns) или флажок **Общие итоги по строкам** (Grand Totals for Rows), можно подвести итоги по столбцам или строкам в сводной таблице
- Установка флажка **Сохранить данные с макетом таблицы** (Save Data With Table Layout) создает дополнительную копию данных, позволяющую быстрее пересчитывать сводную таблицу при ее изменении
- Установка флажка **Автоматически форматировать таблицу** (AutoFormat Table) позволяет использовать средства автоформата Excel

После щелчка на кнопке **Готово** (Finish) на рабочем листе будет создана сводная таблица, представленная на рис. 6.8.

	Г	Н
17	Сумма по полю Размер вклада	
18	Отделение банка	Всего
19	Западное	174300123
20	Северное	183870058
21	Центральное	118836500
22	Общий итог	477006681
23		
24		
25		

Рис. 6.8. Пример сводной таблицы

Для детализации итогов сводной таблицы необходимо дважды щелкнуть на названии поля (например, Западное) и в открывшемся диалоговом окне **Показать детали** (Show Detail) (рис. 6.9) указать, по каким признакам необходимо выполнить детализацию.

Рис. 6.9. Диалоговое окно **Показать детали**

В рассматриваемом примере детализируем сводную таблицу по типу вклада. Результат представлен на рис. 6.10. Теперь двойным щелчком на названии поля **Отделение банка** можно скрывать или показывать его детализацию.

	Г	Н	И
17	Сумма по полю Размер вклада		
18	Отделение банка	Тип вклада	Всего
19	Западное	Депозит	97100000
20		Текущий	77200123
21	Западное Всего		174300123
22	Северное	Текущий	183870058
23	Северное Всего		183870058
24	Центральное	Депозит	118836000
25		Текущий	5000500
26	Центральное Всего		118836500
27	Общий итог		477006681

Рис. 6.10. Детализированная сводная таблица

При изменении данных в базе данных, на основе которой была построена сводная таблица, не происходит автоматического пересчета итогов в сводной таблице. Для ее обновления необходимо выполнить команду **Данные, Обновить данные** (Data, Refresh Data) или щелкнуть правой кнопкой мыши любую ячейку сводной таблицы и в раскрывшемся контекстном меню выбрать команду **Обновить данные** (Refresh Data).

Для удаления поля из сводной таблицы достаточно щелкнуть его название и перетащить его за пределы сводной таблицы.

Перемещая названия полей, можно менять их местоположение. Например, переместив название поля **Тип вклада** из строчки в столбец, получим измененную сводную таблицу (рис. 6.11).

	G	H	I	J
17	Сумма по полю Размер вклада	Тип вклада		
18	Отделение банка	Депозит	Текущий	Общий итог
19	Западное	97100000	77200123	174300123
20	Северное	0	183870058	183870058
21	Центральное	113836000	5000500	118836500
22	Общий итог	210936000	266070681	477006681

Рис. 6.11. Сводная таблица после перемещения названия поля **Тип вклада** из строчки в столбец

Для редактирования сводной таблицы ее необходимо выделить и выполнить команду **Сервис, Сводная таблица** (Tools, Pivot Table). В результате открывается третье диалоговое окно **Мастер сводных таблиц** (PivotTable Wizard). Преобразуем, например, данную сводную таблицу в таблицу, подсчитывающую количество вкладов различных типов в различных отделениях с учетом примечаний (рис. 6.12).

На рис. 6.13 показан результат создания такой таблицы.

Изменить поле сводной таблицы можно, дважды щелкнув его название. Открывшееся диалоговое окно **Вычисление поля сводной таблицы** (PivotTable Field) (рис. 6.6) предоставляет возможность изменить имя поля и операцию, на основе которой составляется сводная таблица.

Двойной щелчок на ячейке сводной таблицы вызывает создание новой таблицы (на новом рабочем листе), содержащей обобщенную информацию об этой ячейке. Например, двойной щелчок на ячейке **H19** таблицы, представленной на рис. 6.11, создает таблицу, показанную на рис. 6.14.

Excel предоставляет удобное средство для группировки данных. Например, для того чтобы в нашей итоговой сводной таблице объединить северное и западное отделения банка в одно северо-западное, необходимо выбрать на-

звания полей с этими двумя отделениями и выполнить команду **Данные, Структура, Сгруппировать** (Data, Group and Outline, Group). Результат ее выполнения показан на рис. 6.15.

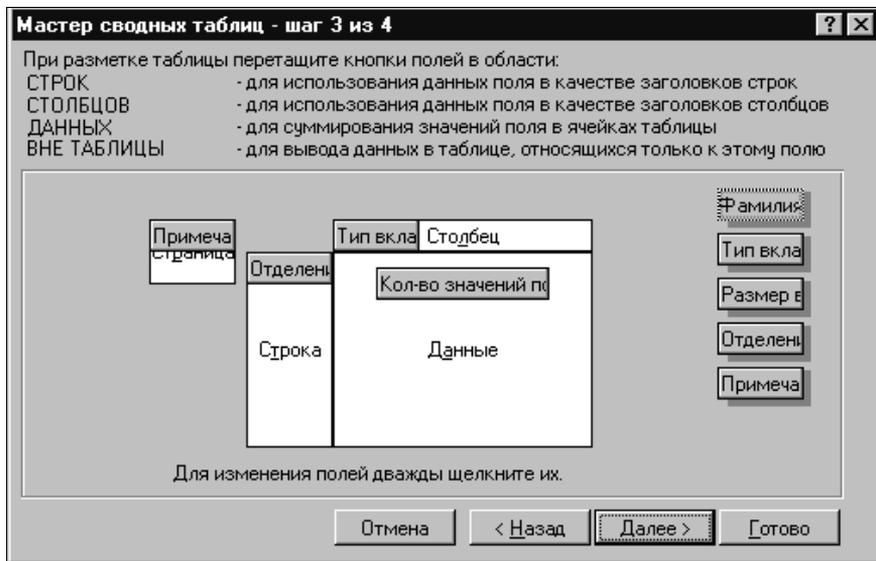


Рис. 6.12. Преобразование сводной таблицы

	G	H	I	J
15	Примечание	(Все)		
16				
17	Кол-во значений по полю Тип вклада	Тип вклада		
18	Отделение банка	Депозит	Текущий	Общий итог
19	Западное	2	2	4
20	Северное	0	7	7
21	Центральное	5	1	6
22	Общий итог	7	10	17

Рис. 6.13. Результат преобразования сводной таблицы

	A	B	C	D	E
1	Фамилия	Тип вклада	Размер вклада	Отделение банка	Примечание
2	Сидоров	Депозит	29800000	Западное	
3	Козлов	Депозит	67300000	Западное	
4					

Рис. 6.14. Развернутое содержание ячейки сводной таблицы

	G	H	I	J	K
17	Сумма по полю Размер вклада		Тип вклада		
18	Отделение банка2	Отделение банка	Депозит	Текущий	Общий итог
19	Группа1	Западное	97100000	77200123	174300123
20		Северное	0	183870058	183870058
21	Центральное	Центральное	113836000	5000500	118836500
22	Общий итог		210936000	266070681	477006681

Рис. 6.15. Группировка данных в сводной таблице

Изменить название поля Отделение банка2 на Северно-западное можно в диалоговом окне **Вычисление поля сводной таблицы** (PivotTable Field) (рис. 6.6), которое открывается двойным щелчком на этом названии.

Для того чтобы разгруппировать данные, необходимо выбрать названия соответствующих полей и выполнить команду **Данные, Структура, Разгруппировать** (Data, Group and Outline, Ungroup).

Глава 7



Visual Basic for Applications (VBA)

Материал предыдущих глав, несомненно, дает уверенность в том, что составление любого документа в Excel теперь не вызовет проблем. Ни одна задача не сможет поставить вас в тупик. Однако если вы стремитесь к совершенству, то Visual Basic For Applications (VBA) — это то, что вам нужно.

Visual Basic For Applications — это сочетание одного из самых простых языков программирования и всех вычислительных возможностей Excel. С помощью VBA вы сможете легко и быстро создавать разнообразные приложения, даже не являясь специалистом в области программирования. VBA содержит графическую среду, позволяющую наглядно конструировать экранные формы и управляющие элементы. Использование VBA придаст вашим документам неповторимую изюминку и элегантность. Visual Basic For Applications позволяет с легкостью решать многие задачи, о возможности выполнения которых средствами Excel вы раньше даже и не подозревали.

При изучении программирования на языке VBA важнейшую роль играют примеры. Как правило, удачно подобранный пример может передать основную идею лучше, чем длинные теоретические рассуждения. Кроме того, как показывает опыт, теоретическая рутинка часто действует усыпляюще на начинающего пользователя. Для того чтобы держать читателя бодрствующим и заинтересованным, мы подходим к изучению VBA в этой главе "от конкретных примеров".

Глава насыщена большим количеством полностью разработанных приложений, одни из которых можно использовать без всяких изменений, а другие — после соответствующей корректировки в соответствии с вашими потребностями.

7.1. Простейшие функции и процедуры. Типы переменных

7.1.1. Функция пользователя

Простейшей задачей, решаемой на VBA, является создание функций пользователя, имеющих имена. С функциями пользователя можно работать с помощью мастера функций точно так же, как и со встроенными функциями рабочего листа. Создадим, например, функцию пользователя, математически определенную следующим образом:

$$y = \sin(\pi x) e^{-2x}$$

Функции пользователя создаются на рабочем листе с названием модуль VBA, после чего с ними можно работать с помощью мастера функций. Для создания модуля VBA необходимо щелкнуть правой кнопкой мыши ярлычок листа, перед которым вы хотите разместить данный модуль, и из раскрывшегося контекстного меню выбрать команду **Вставка (Insert)**. На вкладке **Общие (General)** открывшегося диалогового окна **Вставка (Insert)** необходимо выбрать тип листа — **Модуль (Module)**. После нажатия кнопки **ОК** произойдет вставка листа модуля. По умолчанию Excel присваивает листам модуля имена: Модуль1, Модуль2 (Module1, Module2) и т. д. Теперь на листе модуля, как это показано на рис. 7.1, можно ввести текст следующей программы:

```
Function Y(x)
    Y = Sin(Application.Pi() * x) * Exp(-2 * x)
End Function
```

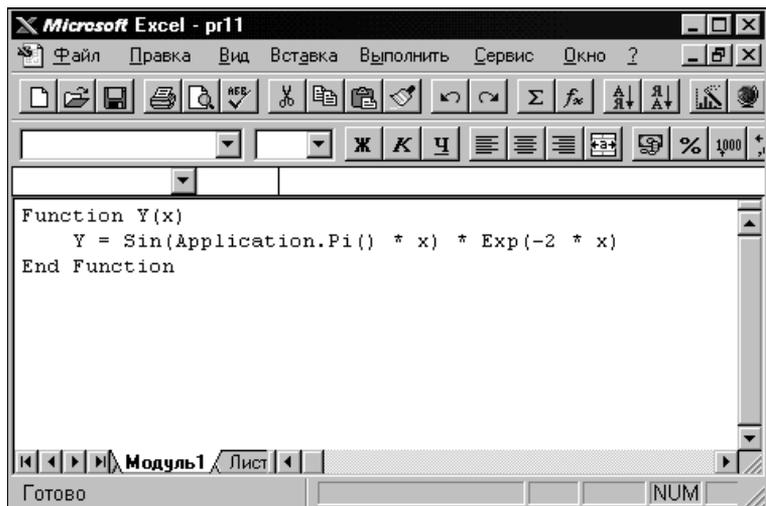


Рис. 7.1.
Лист модуля

Функция $\text{Pi}()$ в VBA возвращает значение постоянной π (в Excel — функция $\text{ПИ}()$). Заметим, что поскольку функция $\text{Pi}()$ не является внутренней функцией VBA, то ее необходимо записать в форме $\text{Application.Pi}()$. Об объекте Application см. раздел 7.7.

По умолчанию созданная функция при выборе ее с помощью мастера функций относится к категории **Определенные пользователем** (User Defined). Можно изменить категорию принадлежности функции, а также добавить необходимые комментарии в мастере функций для удобства пользования созданной функцией, если выполнить следующие действия. При активном листе Модуль1, на котором введена функция Y , нажмите клавишу $\langle F2 \rangle$. На экране появится диалоговое окно **Просмотр объектов** (Object Browser) (рис. 7.2).

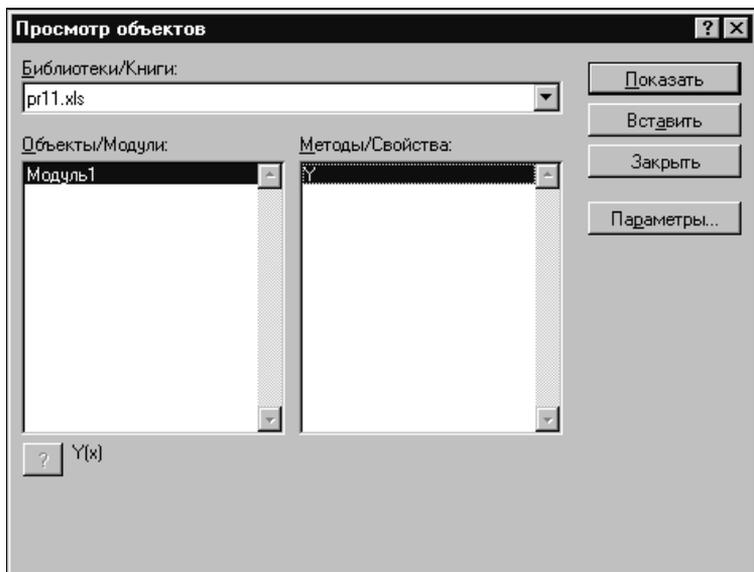


Рис. 7.2. Диалоговое окно **Просмотр объектов**

При нажатии кнопки **Параметры** (Options) открывается диалоговое окно **Параметры макроса** (Macro Options). В этом окне в поле **Описание** (Description) введите текст комментария, например, *Вспомогательная функция* (рис. 7.3).

С помощью диалогового окна **Параметры макроса** (Macro Options) можно также изменить категорию, к которой принадлежит данная функция, и связать с ней файл справки с подробным описанием данной функции. После внесения необходимой информации нажмите кнопку **ОК** диалогового окна **Параметры макроса** (Macro Options), а затем кнопку **Закрыть** (Close) диалогового окна **Просмотр объектов** (Object Browser). Теперь все готово для работы с функцией Y .

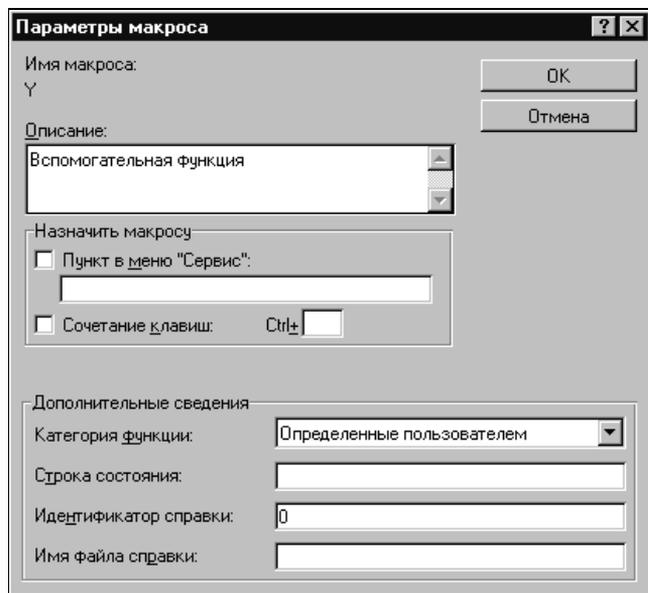


Рис. 7.3.
Ввод описания функции
в диалоговом окне
Параметры макроса

Пусть в ячейку A1 записано число 0,2, и вам нужно вычислить в ячейке B1 значение функции Y при $x = 0,2$. Для этого достаточно в ячейку B1 ввести формулу $=Y(A1)$ (рис. 7.4). Это можно сделать теперь с помощью мастера функций, который будет содержать функцию Y наряду с другими встроенными функциями Excel.

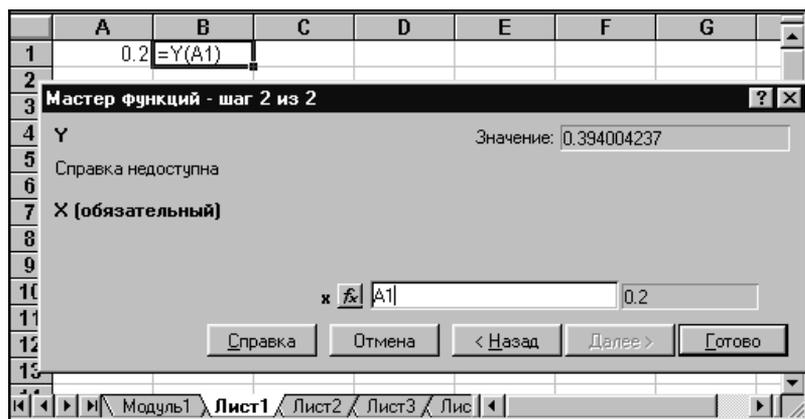


Рис. 7.4. Ввод функции Y в ячейку с помощью мастера функций

Конечно, тот же результат можно получить, и не создавая функцию пользователя, а просто введя в ячейку B1 следующую формулу

$=\text{SIN}(\text{ПИ}() * A1) * \text{EXP}(-2 * A1)$

Однако при постоянном ее использовании лучше все же создать соответствующую функцию пользователя, чтобы не вводить эту формулу каждый раз заново.

Общий вид функции пользователя:

```
Function ИМЯ_ФУНКЦИИ(СПИСОК_ПАРАМЕТРОВ)
    ТЕЛО_ФУНКЦИИ
End Function
```

ТЕЛО_ФУНКЦИИ состоит из описательной части и блока операторов, выполняющихся один за другим. Если необходимо прекратить выполнение функции в некотором конкретном месте, это можно сделать с помощью оператора Exit Function. При определении функции иногда бывает удобно описать типы параметров и вычисляемого значения функции.

Именем функции или переменной может быть любой идентификатор, определенный пользователем. Идентификатор — это последовательность букв, цифр и символа подчеркивания, начинающаяся с буквы. Отметим, что пробелы внутри идентификаторов недопустимы, поэтому они заменяются символом подчеркивания. Для упрощения чтения и понимания значения идентификаторов на практике рекомендуется при написании сложных идентификаторов использовать строчные и прописные буквы, а также символ подчеркивания. Например, вместо идентификатора `datetoday` лучше написать `DateToday` или `Date_Today`.

Носителем возвращаемого значения функции пользователя является имя функции. Поэтому в теле функции пользователя, вычисляющей некоторое значение, должен присутствовать, по крайней мере, один оператор, присваивающий имени функции значение какого-либо выражения.

С помощью ключевого слова `Optional` можно сделать некоторые переменные в функции пользователя необязательными (необязательными являются те из них, которые можно при вызове не задавать). Необязательные переменные должны иметь тип `Variant` (см. раздел 7.1.2).

Рассмотрим функцию пользователя `Miss_Side_Tr`, которая находит длину недостающей стороны прямоугольного треугольника, где переменные *A* и *B* отведены под длины катетов, а переменная *C* — под гипотенузу. Например, формула `=Miss_Side_Tr(;B2;C2)` вычисляет катет *A* по введенным в ячейки *B2* и *C2* катету *B* и гипотенузе *C*. При работе с необязательными переменными необходимо использовать функцию `IsMissing`, возвращающую `True`, если соответствующий аргумент не был передан в процедуру, и `False` — в противном случае.

```
Function Miss_Side_Tr(Optional A; Optional B; Optional C)
    If Not (IsMissing(A)) And Not (IsMissing(B)) Then
```

```

        Miss_Side_Tr = Sqr(A ^ 2 + B ^ 2)
    End If
    If Not (IsMissing(A)) And Not (IsMissing(C)) Then
        Miss_Side_Tr = Sqr(C ^ 2 - A ^ 2)
    End If
    If Not (IsMissing(B)) And Not (IsMissing(C)) Then
        Miss_Side_Tr = Sqr(C ^ 2 - B ^ 2)
    End If
End Function

```

7.1.2. Переменные и постоянные

Все переменные в VBA имеют тип. Тип указывает, что может хранить переменная: целое число, строку, дату и т. д. Базовые типы переменных VBA приведены в табл. 7.1.

Таблица 7.1. Типы переменных

Тип данных	Занимаемый размер памяти (байт)	Диапазон
Boolean (Логический)	2	True (Истина) или False (Ложь)
Integer (Целое)	2	От -32 768 до 32 768
Long (Длинное целое)	4	От -2 147 483 648 до 2 147 483 647
Single (Число с плавающей точкой)	4	По абсолютной величине от 1,401298E-45 до 3,402823E38
Date (Дата)	8	От 1 января 100 г. до 31 декабря 9 999 г.
Array (Массив)	Зависит от размера массива и типа элементов массива	
Double (Число с плавающей точкой двойной точности)	8	По абсолютной величине от 4,94065645841247E-324 до 1,79769313486232E308
Object (Объект)	4	Любой определенный объект
String (строка переменной длины)	10 + длина строки	От 0 до 2×10 ⁹
Currency (Денежный)	8	От -922 337 203 685 477,5808 до 922 337 203 685 477,5807
Variant (Вариант)	Зависит от содержимого переменной	

Если не указан тип переменной, то по умолчанию используется тип Variant. Переменные этого типа могут хранить все, что в них поместят, т. е. их тип изменяется в зависимости от последнего присвоения. Поэтому тип Variant очень удобен. Однако указание конкретного типа позволяет делать функции более быстрыми и надежными.

Переменную в VBA можно описать с помощью следующей конструкции:

```
Dim ИМЯ_ПЕРЕМЕННОЙ As ТИП_ПЕРЕМЕННОЙ
```

Например, возможны следующие описания переменных:

```
Dim A As Integer
```

```
Dim C, D As Integer, E As Single
```

По умолчанию строковая или текстовая переменная (String) является массивом переменной длины, который содержит символы. Однако текстовая переменная может быть определена и фиксированной длины. В следующем примере объявляется символьный массив размером в 25 символов:

```
Dim S As String*25
```

В этом случае, если вы присвоите переменной S строку длиной более 25 символов, то она будет усечена.

Как и в других языках программирования, в VBA вы можете использовать массивы. Примеры объявления массивов:

```
Dim B(3, 3) As Single
```

```
Dim A(12) As Integer
```

Первая строка объявляет двухмерный массив 3×3 (матрицу), состоящий из действительных чисел. Вторая строка объявляет одномерный массив (вектор), состоящий из 12 целых чисел, причем по умолчанию первый элемент массива будет $A(0)$, а последний — $A(11)$. В этом случае говорят, что 0 — базовый индекс. Можно изменить базовый индекс, написав в начале листа модуля оператор Option Base 1. После этого индексы массивов A и B будут начинаться с единицы. Другим способом изменения базового индекса является использование ключевого слова To при объявлении массива, например

```
Dim B(1 To 3, 1 To 3) As Single
```

```
Dim A(1 To 12) As Integer
```

Массив в программе определяется поэлементно, например

```
Dim B(1 To 2, 1 To 2) As Single
B(1,1)=2
B(1,2)=4
B(2,1)=1
B(1,2)=6
```

Удобным способом определения одномерных массивов является функция `Array`, преобразующая список элементов, разделенных запятыми, в вектор из этих значений и присваивающая ему тип `Variant`, например

```
Dim A As Variant
A = Array(10,20,30)
B = A(2)
```

Иногда в процессе выполнения программы требуется изменять размер массива. В этом случае его объявляют как динамический. Для этого при объявлении массива не нужно указывать размерность, например

```
Dim R( ) As Single
```

Затем в программе следует вычислить необходимый размер массива, присвоив его некоторой переменной, например n , и указать размер динамического массива с помощью оператора `ReDim`

```
ReDim R(n, n)
```

При описании типов переменных, являющихся параметрами функций или процедуры, `Dim` опускается. В примере, приведенном в разделе 7.1.1, параметр x функции Y и вычисляемое в функции значение по умолчанию имеют тип `Variant`. Так как по смыслу задачи x и Y являются действительными числами, то данную функцию пользователя можно также записать следующим образом (с явным указанием типов):

```
Function Y(x As Double) As Double
    Y = Sin(Application.Pi() * x)*Exp(-2 * x)
End Function
```

Знак "=" является оператором присваивания, который предписывает выполнить выражение, заданное в его правой части, и присвоить результат пере-

менной, идентификатор которой расположен в левой части. Например, в результате действия следующей пары операторов

```
X=2
```

```
X=X+2
```

переменной *x* будет присвоено значение 4.

Часто при написании программ необходимо использовать одни и те же постоянные значения: числа, строки, даты и т. д. В этом случае вместо них лучше использовать имена, которые обозначают эти значения. В VBA можно задать постоянную с помощью одной из следующих конструкций:

```
Const ИМЯ_ПОСТОЯННОЙ = ВЫРАЖЕНИЕ
```

```
Const ИМЯ_ПОСТОЯННОЙ As ТИП_ПОСТОЯННОЙ = ВЫРАЖЕНИЕ
```

Примеры возможного определения постоянных:

```
Const FileName = "test.xls"
```

```
Const PI As Double = 3.14159
```

7.1.3. Процедуры

Кроме функций пользователя, на листе модуля можно записывать процедуры (подпрограммы). Процедуры имеют стандартное оформление:

```
Sub ИМЯ_ПРОЦЕДУРЫ(СПИСОК_ПАРАМЕТРОВ)
```

```
    ТЕЛО_ПРОЦЕДУРЫ
```

```
End Sub
```

Процедура — это совокупность операторов VBA, выполняющих определенные действия. Например, процедура *Сигнал* использует оператор *Веер* для синтеза звукового сигнала:

```
Sub Сигнал()
```

```
    Веер
```

```
End Sub
```

Досрочное завершение процедуры возможно с помощью оператора `Exit Sub`.

Следует отметить, что кроме выполнения определенных действий процедура, так же, как и функция, может возвращать значения, которые присваиваются параметрам внутри процедуры. В отличие от функций пользователя, процедуры нельзя вызывать из ячеек рабочего листа.

Приведем некоторые общие сведения об организации программ в VBA. Программа VBA состоит из одного или нескольких модулей. Модуль — это лист с текстом программы, вставленный в рабочую книгу. Обычно текст программы VBA начинается с опций, которые управляют описанием переменных, способом сравнения строк и т. д. Затем следует объявление глобальных переменных или констант для данного модуля, т. е. таких переменных, которые используются во всех процедурах модуля. Далее располагается непосредственно текст функций пользователя и процедур, составляющих саму программу.

При написании программ удобно внутри текста помещать комментарии. Комментарии — это пояснительный текст, который можно записать в любом месте программ. Каждая строка комментариев начинается со знака апострофа. Комментарии игнорируются компилятором, и поэтому никакого влияния на программу не оказывают. Комментарии удобно использовать также при отладке операторов для их временного отключения.

Для более компактной записи программы используется символ двоеточия ":" в качестве разделителя операторов в одной строке.

Пример организации модуля:

```
Option Base 1
Option Explicit
Const PI As Double = 3.14159
' PI - глобальная постоянная
Dim x As Double
' x - глобальная переменная
Function Disc(R As Double) As Double
' Disc вычисляет площадь круга
    x = 2
    Disc = PI * R ^ 2
End Function

Function Rec(a, b, c As Double) As Double
' Rec вычисляет площадь треугольника
Dim p As Double
' p - локальная переменная
    p = (a + b + c) / 2
    Rec = Sqr(p * (p - a) * (p - b) * (p - c))
End Function
```

```

Sub Results()
    Dim R_1, R_2, a, b, c As Double
' R_1, R_2, a, b, c - локальные переменные
    R_1 = Disc(2.5)
    x = x + 2
    MsgBox("Площадь круга = " & CStr(R_1) & ",x=" & CStr(x))
    a = 1: b = 1: c = Sqr(2)
    R_2 = Rec(a, b, c)
    MsgBox ("Площадь треугольника = " & CStr(R_2))
End Sub

```

Здесь инструкция `Option Explicit` указывает на необходимость описания типов всех переменных, используемых на данном листе модуля. Функция `CStr` переводит числовой формат в строковый. (Функция `Str` также переводит числовой формат в строковый, но только для чисел с десятичной точкой.)

Перечислим другие функции преобразования типов:

<code>CBool</code>	Преобразует выражение в тип <code>Boolean</code>
<code>CCur</code>	Преобразует выражение в тип <code>Currency</code>
<code>CDate</code>	Преобразует выражение в тип <code>Date</code>
<code>CLng</code>	Преобразует выражение в тип <code>Long</code>
<code>CInt</code>	Преобразует выражение в тип <code>Integer</code>
<code>CDbl</code>	Преобразует выражение в тип <code>Double</code>
<code>CSng</code>	Преобразует выражение в тип <code>Single</code>
<code>CVar</code>	Преобразует выражение в тип <code>Variant</code>

Функция `Sqr` вычисляет квадратный корень аргумента. Процедура `MsgBox` выводит текстовую информацию в окно вывода (подробнее см. раздел 7.11).

В результате работы программы появятся сообщения, представленные на рис. 7.5.

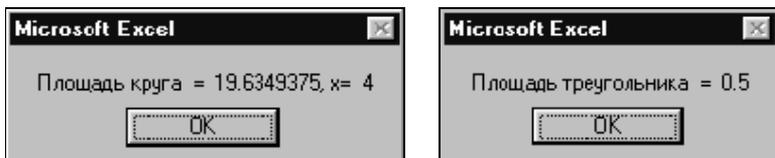


Рис. 7.5. Результаты работы программы `Results`

В общем случае при вызове процедур передаваемые параметры не надо заключать в скобки. Приводимый ниже пример показывает основные способы передачи параметров в процедуры.

```
Sub Assistant(a, b)
    c = a + b
    MsgBox CStr(c)
End Sub

Sub MAIN()
Dim x, y As Double
    Assistant 1, 3
    x = 1: y = 3
    Assistant x, y
    Assistant a:=1, b:=3
End Sub
```

В данном примере при третьем вызове процедуры `Assistant` передача параметров осуществляется присвоением значений именам параметров с помощью оператора присваивания " :=" (который применяется только по отношению к параметрам процедуры, в отличие от ранее описанного оператора присваивания "="). При таком способе передачи параметров порядок присваивания значений не важен, т. к. все определяется именами параметров.

7.2. Функции пользователя с операторами условного перехода *If-Then* и *If-Then-Else*

Рассмотрим пример функции пользователя

```
Function G(y)
    If y <= 0 Then G = (1 + y ^ 2) / (1 + y ^ 4) ^ (1/2) _
        Else G = 2 * y + Sin(y) ^ 2 / (2 + y)
End Function
```

вычисляющей следующую функцию с двумя условиями:

$$g = \begin{cases} \frac{1 + y^2}{\sqrt{1 + y^4}}, & y \leq 0, \\ 2y + \frac{\sin^2(y)}{2 + y}, & y > 0. \end{cases}$$

Заметим, что знак подчеркивания "_" в конце строки обозначает, что следующая строка является продолжением предыдущей.

Пусть в ячейку A1 записано число 0,2, и вам нужно вычислить в ячейке B1 значение функции G при $y = 0,2$. Для этого достаточно в ячейку B1 ввести формулу $=G(A1)$. Тот же результат можно получить, если ввести в ячейку B1 следующую формулу

`=ЕСЛИ(A1<=0; (1+A1^2) / (1+A1^4)^(1/2); 2*A1+SIN(A1)^2 / (2+A1))`

В функции G использовался условный оператор If-Then-Else, обеспечивающий передачу управления в программе в зависимости от выполнения условия.

Синтаксис:

`If УСЛОВИЕ Then ОПЕРАТОР_1 Else ОПЕРАТОР_2`

В операторе условия ОПЕРАТОР_1 выполняется, если УСЛОВИЕ истинно, в противном случае выполняется ОПЕРАТОР_2.

УСЛОВИЕ — это выражение логического типа. Результат выражения всегда имеет булевский тип. Выражение может быть простым и сложным. При записи простых условий могут использоваться все возможные операции отношения, указанные в табл. 7.2.

Таблица 7.2. Логические отношения

Операция	Название	Выражение	Результат
=	Равно	$A=B$	True, если A равно B
<>	Не равно	$A<>B$	True, если A не равно B
>	Больше	$A>B$	True, если A больше B
<	Меньше	$A<B$	True, если A меньше B
>=	Больше или равно	$A>=B$	True, если A больше или равно B
<=	Меньше или равно	$A<=B$	True, если A меньше или равно B

Сложные условия образуются из простых путем применения логических операций и круглых скобок. Список логических операций приведен в табл. 7.3.

Таблица 7.3. Логические операции

Операция	Название	Выражение	A	B	Результат
Not	Логическое отрицание	Not A	False		True
			True		False
And	Логическое И	A And B	True	True	True
			True	False	False
			False	True	False
			False	False	False
Or	Логическое ИЛИ	A Or B	True	True	True
			True	False	True
			False	True	True
			False	False	False

В условном операторе допустимо использование блока операторов вместо любого из операторов. В этом случае условный оператор имеет вид:

```
If УСЛОВИЕ Then
    БЛОК_ОПЕРАТОРОВ_1
Else
    БЛОК_ОПЕРАТОРОВ_2
End If
```

Ветвь Else в условном операторе является необязательной. Таким образом, возможен следующий синтаксис оператора условия:

```
If УСЛОВИЕ Then ОПЕРАТОР
If УСЛОВИЕ Then
    БЛОК_ОПЕРАТОРОВ
End If
```

Заметим, что в операторе условия после Then можно разместить несколько операторов, для того чтобы все они выполнялись, если условие истинно. В этом случае они должны располагаться в одну строчку и быть разделены двоеточием, например

```
If A > 10 Then A = A + 1 : B = B + A : C = C + B
```

В условном операторе может проверяться несколько условий. В этом случае условный оператор имеет вид:

```
If УСЛОВИЕ_1 Then
    БЛОК_ОПЕРАТОРОВ_1
ElseIf УСЛОВИЕ_2 Then
    БЛОК_ОПЕРАТОРОВ_2
ElseIf УСЛОВИЕ_3 Then
    ...
Else
    БЛОК_ОПЕРАТОРОВ
End If
```

В данном условном операторе выполняется БЛОК_ОПЕРАТОРОВ_i, если истинно УСЛОВИЕ_i, в противном случае выполняется БЛОК_ОПЕРАТОРОВ.

Рассмотрим использование условного оператора и оператора Exit Function на примере функции пользователя, вычисляющей квадратный корень числа.

```
Function Sq(x)
    If x < 0 Then Exit Function Else Sq = x ^ (1/2)
End Function
```

7.3. Функции пользователя с несколькими операторами условного перехода

Рассмотрим пример функции пользователя, вычисляющей значение следующей функции с тремя условиями:

$$z = \begin{cases} \frac{1 + |t|}{\sqrt[3]{1 + t + t^2}}, & t \leq -1, \\ 2 \ln(1 + t^2) + \frac{1 + \cos^4(t)}{2 + t}, & t \in (-1, 0), \\ (1 + t)^{3/5}, & t \geq 0. \end{cases}$$

Функцию z можно вычислить с помощью следующей функции пользователя:

```
Function z(t)
    If t <= -1 Then z = (1+Abs(t))/(1+t+t^2)^(1/3)
    If t > -1 And t < 0 Then z = 2*Application.Ln(1+t^2) + _
        (1+Cos(t)^4)/(2+t)
    If t >= 0 Then z = (1+t)^(3/5)
End Function
```

Функция Ln не является внутренней функцией VBA, поэтому для ее вызова нужно пользоваться конструкцией Application.Ln.

Существует также другой способ вычисления функции z :

```
Function z(t)
    If t <= -1 Then
        z = (1 + Abs(t)) / (1 + t + t^2)^(1/3)
    ElseIf t < 0 Then z = 2 * Application.Ln(1 + t^2) + _
        (1 + Cos(t)^4)/(2 + t)
    Else
        z = (1 + t)^(3/5)
    End If
End Function
```

Пусть в ячейку A1 записано число 0.2, и вам нужно вычислить в ячейке B1 значение функции z при $t = 0.2$. Для этого достаточно в ячейку B1 ввести формулу =z(A1). Тот же результат можно получить, если ввести в ячейку B1 следующую формулу:

```
=ЕСЛИ(A1<=-1;(1+ABS(A1))/(1+A1+A1^2)^(1/3);ЕСЛИ(A1<0;
2*LN(1+A1^2)+(1+COS(A1)^4)/(2+A1);(1+A1)^(3/5)))
```

7.4. Функции пользователя, используемые при расчете комиссионных

7.4.1. Функции пользователя с оператором условного перехода *If-Then*

Рассмотрим пример функции пользователя VBA, вычисляющей комиссионные, начисляемые по следующему правилу:

- Если продукции продано не меньше, чем на 1000 000 руб., то комиссионные составляют 2% от стоимости реализованной продукции

- ❑ Если продукции продано меньше, чем на 1 000 000 руб., то комиссионные составляют 1% от стоимости реализованной продукции
- ❑ Если стаж работы в фирме не меньше 5 лет, то выплачивается доплата в размере 0,5% от стоимости реализованной продукции

Первый способ составления функции пользователя:

```
Function Премия(стаж, продажа)
    If стаж >= 5 And продажа >=1000000 Then Премия=0.025*продажа
    If стаж < 5 And продажа >=1000000 Then Премия=0.02*продажа
    If стаж >= 5 And продажа <1000000 Then Премия=0.015*продажа
    If стаж < 5 And продажа <1000000 Then Премия=0.01*продажа
End Function
```

Второй способ:

```
Function Премия(стаж As Integer, продажа As Integer) as Integer
Dim sAge, sSale As Integer
    If стаж >= 5 Then sAge = 0.005 * продажа
    If продажа >= 1000000 Then
        sSale = 0.02 * продажа
    Else
        sSale = 0.01 * продажа
    End If
    Премия = sAge + sSale
End Function
```

Пусть в ячейку A1 введена величина стажа работы в фирме, равная 15, а в ячейку B1 — объем реализованной продукции, равный 5000000. Для того чтобы вычислить комиссионные в ячейке C1, достаточно в эту ячейку ввести формулу =Премия(A1;B1). Тот же результат получается при вводе в ячейку C1 следующей формулы:

```
=ЕСЛИ(И(A1>=5;B1>=1000000);B1*0.025;
ЕСЛИ(И(A1<5;B1>=1000000);B1*0.02;
ЕСЛИ(И(A1>=5;B1<1000000);B1*0.015;
ЕСЛИ(И(A1<5;B1<1000000);B1*0.01;)))
```

или формулы

```
ЕСЛИ(A1>=5; B1*0.005)+ЕСЛИ(B1>=1000000;B1*0.02; B1*0.01)
```

7.4.2. Функции пользователя с оператором выбора *Select Case*

Оператор выбора `Select Case` удобно использовать, когда в зависимости от значения некоторого выражения, имеющего конечное множество допустимых значений, необходимо выполнить разные действия.

Синтаксис:

```
Select Case ТЕСТИРУЕМОЕ_ВЫРАЖЕНИЕ
    Case УСЛОВИЕ_ВЫБОРА_1
        БЛОК_ОПЕРАТОРОВ_1
        ...      ....      ...
    Case УСЛОВИЕ_ВЫБОРА_n
        БЛОК_ОПЕРАТОРОВ_n
    Case Else
        БЛОК_ОПЕРАТОРОВ
End Select
```

После каждого оператора `Case` может находиться произвольное количество других операторов, и все они будут выполняться, если условие оператора `Case` истинно. При использовании одного оператора его можно поместить в одну строку с оператором `Case`. `Is` является ключевым словом VBA, обозначающим `ТЕСТИРУЕМОЕ_ВЫРАЖЕНИЕ` в операторе `Case`. В операторе `Case` допустимы составные условия, например

```
Case 4, 7 To 8, 11 To 12, 15
```

Проверяет, принадлежит ли `ТЕСТИРУЕМОЕ_ВЫРАЖЕНИЕ` одному из отрезков: от 7 до 8 и от 11 до 12 или равняется одному из значений: 4 и 15

```
Case 5, 6, 9 To 10, 13, 14, Is >= 16
```

Проверяет, принадлежит ли `ТЕСТИРУЕМОЕ_ВЫРАЖЕНИЕ` отрезку от 9 до 10 или равняется одному из значений: 5, 6, 13 и 14, или оно не меньше, чем 16

Рассмотрим пример начисления комиссионных на основе оператора выбора `Select Case`. В этом примере размер комиссионных зависит только от объема проданной продукции по правилу, приведенному в табл. 7.4.

Таблица 7.4. Правило начисления комиссионных

Объем продаж, тыс. руб.	Комиссионные, %
0 – 9999	8
10000 – 39999	10
40000 и более	14

Задачу начисления комиссионных решает следующая функция:

```
Function Премия(продажа)
    Select Case продажа
        Case 0 To 9999
            Премия = 0.08*продажа
        Case 10000 To 39999
            Премия = 0.1*продажа
        Case Is >= 40000
            Премия = 0.14*продажа
    End Select
End Function
```

Пусть в ячейку A1 введен объем продаж, равный 150000. Для того чтобы вычислить комиссионные в ячейке B1, достаточно в эту ячейку ввести формулу =Премия(A1). Тот же результат получается при вводе в ячейку B1 следующей формулы

```
=ЕСЛИ(И(A1>=0;A1<10000);A1*0.08;ЕСЛИ(И(A1>=10000;A1<40000);A1*0.1;ЕСЛИ(A1>=40000;A1*0.14)))
```

7.5. Стандартные функции для работы с массивами

Используются следующие функции рабочего листа для работы с массивами:

Count	Количество чисел в массиве (в русскоязычной версии Excel имеется аналогичная функция СЧЕТ)
CountA	Количество элементов массива (СЧЕТЗ)
Sum	Сумма элементов массива (СУММ)

SumProduct	Сумма произведений элементов массивов (СУММПРОИЗВ)
SumSq	Сумма квадратов элементов массива (СУММКВ)
SumXmY2	Сумма квадратов разностей элементов двух массивов (СУММКВРАЗН)
SumX2mY2	Сумма разностей квадратов элементов двух массивов (СУММРАЗНКВ)

Приведем пример функции пользователя VBA, вычисляющей коэффициент корреляции (заметим, что коэффициент корреляции можно вычислить и непосредственно функцией КОРРЕЛ).

```
Function R(x As Variant, y As Variant) As Double
Dim n As Integer
Dim sx; sy; sxy; sx2; sy2 As Double
    n = Application.Count(x)
    sx = Application.Sum(x)
    sy = Application.Sum(y)
    sxy = Application.SumProduct(x, y)
    sx2 = Application.SumSq(x)
    sy2 = Application.SumSq(y)
    R = (n*sxy - sx*sy)/((n*sx2 - sx^2)*(n*sy2 - sy^2))^(1/2)
End Function
```

Коэффициент корреляции двух последовательностей $x_i, y_i, i \in [1, n]$, вычисляется по следующей формуле:

$$R = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sqrt{\left(n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2 \right) \left(n \sum_{i=1}^n y_i^2 - \left(\sum_{i=1}^n y_i \right)^2 \right)}}$$

Пусть, например, необходимо вычислить коэффициент корреляции двух последовательностей: 1, 2, 3, 4, 5, 6 и 5, 8, 11, 12, 18, 21, записанных в диапазоны ячеек А3:А8 и В3:В8, соответственно. Результат необходимо поместить в ячейку С1. Для этого достаточно в ячейку С1 ввести формулу =R(А3:А8;В3:В8). Тот же результат можно получить, если ввести в ячейку С1 следующую формулу

$$= (\text{СЧЕТ}(A3:A8) * \text{СУММПРОИЗВ}(A3:A8; B3:B8) - \text{СУММ}(A3:A8) * \text{СУММ}(B3:B8)) /$$

$$((\text{СЧЕТ}(A3:A8) * \text{СУММКВ}(A3:A8) - \text{СУММ}(A3:A8)^2)$$

$$* (\text{СЧЕТ}(A3:A8) * \text{СУММКВ}(B3:B8) - \text{СУММ}(B3:B8)^2))^{(1/2)}$$

7.6. Стандартные функции для работы с матрицами

Используются следующие функции рабочего листа для работы с матрицами:

Mmult	Произведение двух матриц
Minverse	Обратная матрица
Transpose	Транспонированная матрица
Mdeterm	Определитель матрицы

Приведем пример функции пользователя VBA, решающей систему линейных уравнений $AX = B$, где A — матрица коэффициентов, B — столбец свободных членов, X — столбец неизвестных.

```
Function Реш_лин_сис(A As Variant, B As Variant) As Variant
    Реш_лин_сис = Application.MMult(Application.MInverse(A), B)
End Function
```

Рассмотрим пример функции пользователя VBA, вычисляющей квадратичную форму $z = X^TAX$, где A — матрица коэффициентов, X — столбец неизвестных:

```
Function Квадр_форма(A As Variant, Z As Variant) As Variant
    Квадр_форма = Application.MMult _
        (Application.MMult(Application.Transpose(Z), A), Z)
End Function
```

7.7. Объекты, свойства и методы VBA

Одним из основных понятий VBA является объект. Объект — это то, чем вы управляете с помощью программы на языке VBA, например, диалог, рабочий лист, диапазон ячеек. Каждый объект обладает некоторыми характеристиками или свойствами. Например, диалог может быть видимым или невидимым.

димым в данный момент на экране. Можно узнать текущее состояние диалога с помощью свойства `Visible`. Другим примером свойства объекта может быть шрифт, используемый для отображения информации в ячейке (объекте) рабочего листа. Изменяя свойства, вы можете менять характеристики объекта.

Объект содержит также список методов, которые к нему применимы. Методы — это то, что вы можете делать с объектом. Например, показать диалог на экране или убрать его можно с помощью методов `Show` и `Hide`, соответственно.

Таким образом, объект — это программный элемент, который имеет свое отображение на экране, содержит некоторые переменные, определяющие его свойства, и некоторые методы для управления объектом. Объект является "кирпичиком" построения программ VBA. В VBA имеется много встроенных объектов, например

<code>Range</code>	Диапазон ячеек (может включать только одну ячейку)
<code>Cells</code>	Ячейка
<code>Sheet</code>	Лист
<code>Worksheet</code>	Рабочий лист
<code>DialogSheet</code>	Диалоговое окно

Большинство объектов принадлежит к группе подобных объектов. Эти группы называются наборами. Например, все рабочие листы рабочей книги образуют набор, называемый `Worksheets`. Наборы используются одним из двух способов: либо какое-либо действие совершается над всеми объектами набора, например, удалить, либо со ссылкой на набор выбирается конкретный объект для работы с ним. Второй способ реализован в следующей конструкции:

```
Worksheets ("Первый")
```

выбирающей рабочий лист `Первый` из рабочей книги. Другими примерами наборов являются:

<code>Sheets</code>	Листы
<code>DialogSheets</code>	Диалоговые окна
<code>DrawingObjects</code>	Графические объекты

Изменяя свойства, можно изменять характеристики объекта или набора объектов. Установка значений свойств — это один из способов управления объектами. Для установки свойства необходимо ввести имя объекта, затем

поставить точку и за ней — имя свойства. Далее должен следовать знак равенства и значение свойства. Синтаксис установки значения свойства объекта выглядит следующим образом:

```
Объект.Свойство = Выражение
```

В приведенном ниже примере для свойства Value диапазона ячеек Начальные_данные устанавливается значение 0,1 (т. е. в ячейках этого диапазона будет записано число 0.1):

```
Range("Начальные_данные").Value = 0.1
```

В следующем примере в ячейку A2 вставляется формула путем изменения свойства Formula (формула):

```
Range("A2").Formula = "=СУММ(A1:C1)"
```

+

Некоторые свойства являются неизменяемыми, т. е. допустимыми только для чтения. Имеется в виду, что значение свойства можно узнать, но нельзя изменить. Например, для диапазона, состоящего из одной ячейки, свойства Row (строка) и Column (столбец) являются неизменяемыми. Другими словами, можно узнать, в какой строке и в каком столбце находится ячейка, но изменить ее положение путем изменения этих свойств нельзя. Синтаксис чтения свойств объекта выглядит следующим образом:

```
Переменная = Объект.Свойство
```

В следующем примере переменной Процентная_ставка присваивается значение из ячейки A1 текущего рабочего листа:

```
Процентная_ставка = Range("A1").Value
```

или другим способом:

```
Процентная_ставка = Cells(1,1).Value
```

Кроме свойств, как уже отмечалось выше, у объектов есть ряд методов, т. е. команд, применяемых к объекту. Например, у объекта — диапазон ячеек — имеется метод Clear, позволяющий очистить содержимое диапазона. Приводимый ниже пример показывает, как можно очистить диапазон Начальные_данные:

```
Range("Начальные_данные").Clear
```

А в примере

```
Range("A10:B12").Select
```

выбирается диапазон ячеек A10:B12. Таким образом, синтаксис вызова объекта имеет следующий вид:

```
Объект.Метод
```

В Excel имеется много объектов, причем некоторые из них содержат другие объекты. Например, рабочая книга содержит рабочие листы, рабочий лист содержит диапазон ячеек и т. д. Объектом самого высокого уровня является Application (приложение). Если вы изменяете его свойства или вызываете его методы, то результат применяется к текущей работе Excel. Например, можно завершить работу с Excel, применив метод Quit (выход) к объекту Application:

```
Application.Quit
```

Как было отмечено, точка после имени объекта указывает на то, что далее следует имя свойства или метода. Но после точки можно указать и имя объекта для перехода от одного объекта к другому. Например, следующее выражение очищает вторую строку рабочего листа Май в рабочей книге Отчет:

```
Application.Workbooks("Отчет").Worksheets("Май").Rows(2).Delete
```

Если вспомнить, что строка содержит отдельные ячейки, свойства которых тоже можно устанавливать, то выражение становится еще длиннее:

```
Application.WorkBooks("Отчет").Sheets("Май")._
    Rows(2).Cells(1).Value = "Да"
```

Таким образом, ссылки на объекты могут быть слишком громоздкими. Если данный объект часто используется в программе, то разумно создать объектную переменную с помощью команды Set. Например, предыдущий пример можно записать в следующем виде:

```
Dim R As Object
Set R = Application.WorkBooks("Отчет")._
    Sheets("Май").Rows(2).Cells(1)
R.Value = "Да"
```

Приводимые выше примеры можно записать значительно короче:

- Можно не писать имя объекта Application, т. к. это подразумевается по умолчанию
- При работе с подобъектом уже активизированного объекта нет необходимости указывать содержащий его объект
- VBA использует некоторые свойства и методы, которые возвращают объект, к которому они относятся

Использование последнего обстоятельства позволяет быстро указывать нужный объект. Так, в следующем примере устанавливается значение активной ячейки

```
ActiveCell.Value = "Да"
```

ActiveCell (активная ячейка), ActiveSheet (активный лист), ActiveWorkBook (активная рабочая книга) и Selection (выбор — указывает на выбранный объект) являются примерами свойств, возвращающих объект.

Можно использовать диалоговое окно **Просмотр объектов** (Object Browser), для того чтобы узнать, какие свойства и методы связаны с конкретным объектом. Это особенно удобно для методов, имеющих несколько аргументов, поскольку в модуль будут вставлены имена аргументов. Для того чтобы вызвать окно диалога **Просмотр объектов** (Object Browser), необходимо при активном листе модуля нажать клавишу <F2>. В открывшемся окне в поле **Библиотеки/Книги** (Libraries/ Workbooks) в раскрывающемся списке выберите **Excel**, а в списке **Объекты/Модули** (Objects/Modules) — имя объекта, справку о котором ищете, например **Worksheets** (рис. 7.6).

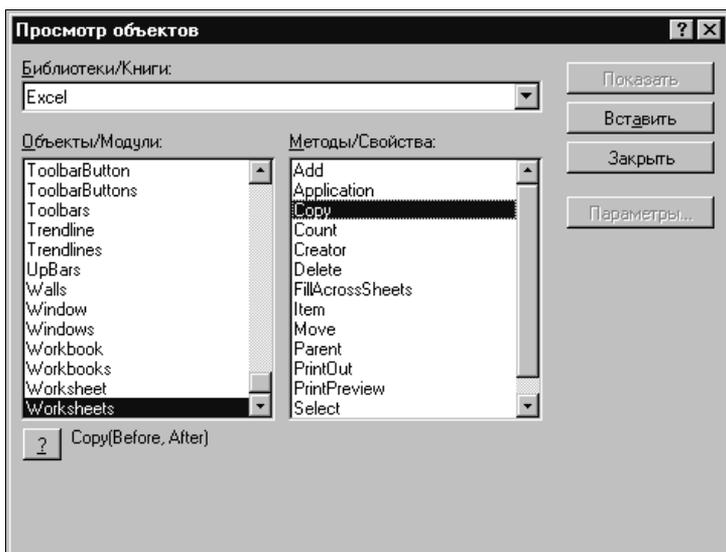


Рис. 7.6.
Диалоговое окно
Просмотр объектов

В поле **Методы/Свойства** (Methods/Properties) выводится список методов и свойств выбранного объекта. Нажав кнопку **?** (Help) можно получить краткую справочную информацию о данном методе, а нажав кнопку **Вставить** (Paste), можно вставить данный метод в лист модуля. Например, для метода Copy в лист модуля вставляется следующий фрагмент:

```
Copy(Before:=, After:=)
```

7.8. Операторы цикла

7.8.1. Оператор цикла For-Next

Для многократного выполнения одного оператора или их группы служит оператор цикла For-Next.

Синтаксис:

```
For СЧЕТЧИК = НАЧАЛЬНОЕ_ЗНАЧЕНИЕ To КОНЕЧНОЕ_ЗНАЧЕНИЕ Step ШАГ
    БЛОК_ОПЕРАТОРОВ
    [Exit For]
    БЛОК_ОПЕРАТОРОВ
Next СЧЕТЧИК
```

Цикл For-Next обеспечивает многократное выполнение блока операторов при последовательном изменении счетчика от начального до конечного значения с указанным шагом изменения. Если Step ШАГ в конструкции отсутствует, то по умолчанию считается, что шаг равен 1. По оператору Exit For можно выйти из оператора цикла до того, как СЧЕТЧИК достигнет последнего значения.

Рассмотрим использование оператора цикла For-Next на примере построения функции пользователя, вычисляющей разность между текущим объемом вклада и размером ссуды при постоянной годовой процентной ставке и неравномерных платежах, т. е. функции пользователя, вычисляемой по следующей формуле:

$$F = \sum_{j=1}^n \frac{P(j)}{(1+i)^{(d(j)-d(1))/365}},$$

где $P(1)$ и $d(1)$ — размер и дата выдачи ссуды, причем $P(1)$ берется со знаком минус, $P(j)$, $d(j)$ — размер и дата j -й выплаты, $n-1$ — число выплат, i — годовая процентная ставка.

```

Option Explicit
Option Base 1
Function Доход(процент As Double, платеж As Variant, _
год As Variant) As Double
Dim i, j, n As Integer, s As Double
    n = платеж.Rows.Count
    s = 0
    For i = 1 To n
        s = s + платеж(i) / _
            (1 + процент)^((год(i) - год(1))/365)
    Next i
    Доход = s
End Function

```

Решим с помощью функции Доход следующую задачу. Предположим, что 11.01.97 у вас берут в долг 10 000 руб. и предлагают вернуть: 20.12.97 — 2000 руб., 18.10.98 — 4000 руб., 12.04.99 — 7000 руб. Имеет ли смысл эта сделка при годовой ставке 10%?

Для решения этой задачи введем данные, как показано на рис. 7.7.

	А	В	С	Д
1	Неравномерные платежи			
2	Ссуда	-10 000.00р.		11.01.97
3	Возврат	2 000.00р.	Даты	20.12.97
4		4 000.00р.		18.10.98
5		7 000.00р.		12.04.99
6				
7	Годовая процентная ставка	10%		
8	Разность между чистым текущим объемом вклада и размером ссуды	857.91р.		

Рис. 7.7. Решение задачи о неравномерных платежах

Ссуда введена в ячейке В2 со знаком минус, т. к. эти деньги у вас забирают. В ячейку В8, где вычисляется разность между текущим объемом вклада и размером ссуды, введем формулу =Доход(В7;В2:В5;D2:D5). В данном случае найденное значение равно 857.91. Так как результат положителен, данная сделка выгодна.

Отметим, что метод Rows возвращает строки диапазона платеж, а свойство Count считает число элементов объекта. Таким образом, платеж.Rows.Count определяет число строк в диапазоне платеж. Если бы вы хотели найти число

столбцов диапазона платежей, то вам нужно было бы использовать конструкцию платежей.Columns.Count.

Рассмотрим пример использования оператора Exit For. Функция Тест определяет номер первого вхождения элемента b в вектор a . Если среди компонент вектора a нет элементов, равных b , функция Тест принимает значение, равное -1 .

```
Function Тест(a As Variant, b As Variant) As Integer
Dim i, n As Integer, t As Boolean
    n = a.Rows.Count*a.Columns.Count
    t = False
    For i = 1 To n
        If a(i) = b Then
            Тест = i
            t = True
            Exit For
        End If
    Next i
    If t = False Then Тест = -1
End Function
```

Приведем еще один пример использования оператора цикла. Предположим, нужно вычислить

$$s = \frac{2 \sum_{i=1}^n x_i + \left(\sum_{i=1}^m \sum_{j=1}^m b_{ij} c_{ij} \right)^2}{1 + \sum_{i=1}^n x_i^2},$$

где x — вектор из n компонент, b и c — матрицы размерности $m \times m$, причем $n = 3$, $m = 2$ и

$$x = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}, \quad b = \begin{pmatrix} 2 & 1 \\ 2 & 5 \end{pmatrix}, \quad c = \begin{pmatrix} 1 & 1 \\ 4 & 5 \end{pmatrix}.$$

Введем в диапазон A1:A3 компоненты вектора x , а в диапазоны B1:C2 и D1:E2 компоненты матриц b и c , соответственно. Найдем значение s тремя способами.

Первый способ. На листе модуля вводится следующая функция пользователя:

```
Option Base 1
Function Q(x, b, c As Variant) As Double
Dim s1, s2, s3 As Double, i, j, n, m As Integer
    n = x.Rows.Count
    m = b.Rows.Count
    s1 = 0
    For i = 1 To n
        s1 = s1 + x(i)
    Next i
    s2 = 0
    For i = 1 To m
    For j = 1 To m
        s2 = s2 + b(i, j) * c(i, j)
    Next j
    Next i
    s3 = 0
    For i = 1 To n
        s3 = s3 + x(i) ^ 2
    Next i
    Q = (2 * s1 + s2 ^ 2) / (1 + s3)
End Function
```

В ячейку A8 введем формулу:

=Q(A1:A3;B1:C2;D1:E2)

Второй способ. В ячейку A6 введем формулу

=(2*СУММ(A1:A3)+СУММПРОИЗВ(B1:C2;D1:E2)^2)/(1+СУММКВ(A1:A3))

или

{=(2*СУММ(A1:A3)+СУММ(B1:C2*D1:E2)^2)/(1+СУММ(A1:A3^2))}

Третий способ. На лист модуля вводится функция пользователя:

```
Function S(x, b, c As Variant) As Double
Dim s1, s2, s3 As Double
```

```

s1 = Application.Sum(x)
s2 = Application.SumProduct(b, c)
s3 = Application.SumSq(x)
S = (2 * s1 + s2 ^ 2) / (1 + s3)

```

End Function

В ячейку A7 введем формулу:

```
=S(A1:A3;B1:C2;D1:E2)
```

7.8.2. Оператор цикла *For-Each-Next*

Для перебора объектов из группы подобных объектов, например, ячеек из диапазона или элементов массива, удобно использовать оператор цикла For-Each-Next.

Синтаксис:

```

For Each Элемент In Группа
    БЛОК_ОПЕРАТОРОВ
    [Exit For]
    БЛОК_ОПЕРАТОРОВ
Next Элемент

```

Например, приводимая ниже процедура `Знак` заменяет все положительные числа диапазона ячеек A1:B2 знаком "+", все отрицательные числа — знаком "-", а нули оставляет без изменения:

```

Option Explicit
Sub Знак()
Dim c As Object
For Each c In Worksheets("Лист1").Range("A1:B2")
If IsNumeric(c.Value) Then
    If c.Value > 0 Then c.Value = "+"
    End If
If IsNumeric(c.Value) Then
    If c.Value < 0 Then c.Value = "-"
    End If
If IsNumeric(c.Value) Then
    If c.Value = 0 Then c.Value = 0

```

```
End If
Next c
End Sub
```

В процедуре `Знак` использовалась функция `IsNumeric(выражение)`, которая возвращает `True`, если выражение может быть описано как числовое, и `False` — в противном случае.

Следующая процедура `Цвет` в зависимости от содержимого ячейки выделенной области, которая возвращается методом `Selection`, изменяет цвет ее фона. Кроме того, если содержимое ячейки положительно, то изменяется цвет, размер и тип шрифта.

```
Sub Цвет()
Dim a As Object
For Each a In Selection
    If IsNumeric(a.Value) Then
        If a.Value > 0 Then
            a.Interior.ColorIndex = 8
            a.Font.Bold = True
            a.Font.ColorIndex = 5
            a.Font.Size = 20
        End If
    End If
    If IsNumeric(a.Value) Then
        If a.Value < 0 Then a.Interior.ColorIndex = 4
    End If
    If IsNumeric(a.Value) Then
        If a.Value = 0 Then a.Interior.ColorIndex = 6
    End If
End For
Next a
End Sub
```

В данной процедуре `Font` — шрифт диапазона ячеек — является объектом. Этот объект имеет следующие свойства:

<code>Size</code>	Размер шрифта, например <code>Worksheets("Лист1").Range("B2").Font.Size = 12</code>
<code>Bold</code>	Жирный шрифт
<code>Italic</code>	Курсивный шрифт

- ColorIndex** Цвет символов. Следующий пример задает зеленый цвет:
`Worksheets ("Лист1").Range ("B2").Font.ColorIndex=4`
- Цвет в VBA задается целым числом от 1 до 56. В табл. 7.5 приведены некоторые из них
- Underline** Подчеркнутый шрифт. Данное свойство может принимать одно из следующих значений:
- `xlNone` — отсутствие подчеркивания
 - `xlSingle` или `xlSingleAccounting` — одинарное подчеркивание
 - `xlDouble` или `xlDoubleAccounting` — двойное подчеркивание

Таблица 7.5. Таблица кодов цвета

Число	Цвет
1	Черный
2	Белый
3	Красный
4	Зеленый
5	Синий
6	Желтый
7	Фиолетовый
8	Голубой

В приведенной выше программе `Interior` также является объектом, характеризующим фон указанного диапазона и имеющим следующее свойство:

- ColorIndex** Цвет. Например, зеленый цвет можно задать ячейке следующим образом:
`Range ("B2").Interior.ColorIndex = 4`

Отметим, что диапазон ячеек обладает следующими свойствами:

- NumberFormat** Числовой формат. Следующий пример задает числовой формат с двумя знаками после десятичной точки:
`Range ("B2").NumberFormat = "0.00"`
- WrapText** Многострочный текст, например
`Range ("B2").Value = "Многострочный текст"`
`Range ("B2").WrapText = True`

7.8.3. Оператор цикла *While-Wend*

Оператор цикла *While-Wend* используется для организации цикла с неизвестным заранее числом шагов.

Синтаксис:

```
While УСЛОВИЕ
    БЛОК_ОПЕРАТОРОВ
Wend
```

Цикл *While-Wend* обеспечивает многократное выполнение блока операторов, пока УСЛОВИЕ принимает значение True.

Приведенная ниже функция *Доход_2* вычисляет то же значение, что и функция *Доход*, но с использованием цикла *While-Wend*.

```
Option Base 1
Function Доход_2(процент As Double, платеж As Variant, _
    год As Variant) As Double
Dim i, j, n As Integer, s As Double
n = платеж.Rows.Count
s = 0
i = 1
While i <= n
    s = s+платеж(i) / (1+процент) ^ ((год(i)-год(1))/365)
    i = i + 1
Wend
Доход_2 = s
End Function
```

7.8.4. Операторы цикла *Do-Loop*

В VBA для организации циклов с неизвестным заранее числом шагов используются и другие операторы цикла: *Do While-Loop*, *Do Until-Loop*, *Do-Loop While* и *Do-Loop Until*.

Синтаксис перечисленных операторов:

```
Do While УСЛОВИЕ
    БЛОК_ОПЕРАТОРОВ
```

```
[Exit Do]
БЛОК_ОПЕРАТОРОВ
```

```
Loop
```

```
Do Until УСЛОВИЕ
    БЛОК_ОПЕРАТОРОВ
    [Exit Do]
    БЛОК_ОПЕРАТОРОВ
```

```
Loop
```

```
Do
    БЛОК_ОПЕРАТОРОВ
    [Exit Do]
    БЛОК_ОПЕРАТОРОВ
```

```
Loop While УСЛОВИЕ
```

```
Do
    БЛОК_ОПЕРАТОРОВ
    [Exit Do]
    БЛОК_ОПЕРАТОРОВ
```

```
Loop Until УСЛОВИЕ
```

Оператор `Do While-Loop` обеспечивает многократное выполнение блока операторов до тех пор, пока УСЛОВИЕ соблюдается, а оператор `Do Until-Loop` — пока УСЛОВИЕ не соблюдается. Операторы `Do-Loop While` и `Do-Loop Until` отличаются от перечисленных выше двух операторов тем, что сначала блок операторов выполняется по крайней мере один раз, а потом проверяется УСЛОВИЕ. Оператор `Exit Do` обеспечивает досрочный выход из оператора цикла.

Приведем пример использования цикла `Do-Loop While` при нахождении корня уравнения $f(x) = 0$ методом Ньютона. Для нахождения приближенного значения корня уравнения методом Ньютона необходимо задать начальное приближение x_0 , а затем определить последующие приближения к корню методом итераций по формуле

$$x_n = x_{n-1} - f(x_{n-1})/f'(x_{n-1}), \text{ где } n = 1, 2, \dots,$$

Процесс останавливается при выполнении неравенства $|x_n - x_{n-1}| \leq \varepsilon$, где ε — точность определения корня.

В следующей программе методом Ньютона решается уравнение $x^2 = 2$. За начальное приближение к корню взято число 3, а корень вычисляется с точностью до 0,00001. Найденное значение корня с помощью метода `ActiveCell` выводится в активную ячейку.

```
Option Explicit
Function F(x As Double) As Double
    F = x ^ 2 - 2
End Function

Function DF(x As Double) As Double
    DF = 2 * x
End Function

Sub MN()
Const Eps = 0.00001
Dim x0, x As Double
    x = 3
    Do
        x0 = x
        x = x - F(x) / DF(x)
    Loop While Abs(x - x0) > Eps
ActiveCell.Value = x
End Sub
```

Примером использования цикла Do-Loop Until может служить следующая программа нахождения корня уравнения $f(x) = 0$ методом деления отрезка пополам. В этой программе находится корень уравнения $x^2 = 2$ с точностью до $\varepsilon = 0,00001$. За отрезок начальной локализации корня берется отрезок $[0, 2]$.

```
Function F(x)
    F = x ^ 2 - 2
End Function

Sub BM()
Dim c, FC As Double
Dim a, b, Eps, BM As Double
a = 0: b = 2: Eps = 0.00001
If F(a) * F(b) >= 0 Then
    MsgBox "Функция не меняет знак " & _
        "на концах отрезка локализации корня."
Exit Sub
End If
Do
```

```
c = (a + b) / 2
FC = F(c) * F(a)
If FC < 0 Then b = c Else a = c
Loop Until b - a < Eps

BM = c
ActiveCell.Offset(rowOffset:=-1, columnOffset:=0).Value = _
    "Корень"
ActiveCell.Value = BM
ActiveCell.Offset(rowOffset:=-1, columnOffset:=1).Value = _
    "Значение функции"
ActiveCell.Offset(rowOffset:=0, columnOffset:=1).Value = FC
End Sub
```

В процедуре `BM` использовался метод `Offset`. Метод `Offset` возвращает диапазон, сдвинутый на указанное число строк и столбцов по отношению к данному диапазону.

Синтаксис:

```
Offset(rowOffset, columnOffset)
```

Аргументы:

<code>rowOffset</code>	Число строк, на которое будет сдвигаться диапазон
<code>columnOffset</code>	Число столбцов, на которое будет сдвигаться диапазон

Использование метода `Offset` позволяет в процедуре `BM` в активную ячейку вводить найденное приближенное значение корня, а в ячейку, находящуюся справа от активной — соответствующее значение функции. В ячейки, расположенные над этими двумя, вводятся текстовые строчки `Корень` и `Значение функции`.

Рассмотрим еще один пример применения цикла `Do-Loop Until` — для расчета корня уравнения $F(x) = 0$ методом Ньютона с выводом на рабочем листе промежуточных результатов. Данную задачу решает следующая программа:

```
Function F(x)
    F = x ^ 3 - 3 * x + 1
End Function

Function DF(x)
```

$$DF = 3 * x ^ 2 - 3$$

```
End Function
```

```
Sub MN()
```

```
    eps = Cells(2, 4).Value
```

```
    Range("A3:C1000").Clear
```

```
    x = Cells(2, 1).Value
```

```
    Cells(2, 2).Value = F(x)
```

```
    Cells(2, 3).Value = DF(x)
```

```
    i = 3
```

```
    Do
```

```
        x0 = x
```

```
        x = x - F(x) / DF(x)
```

```
        Cells(i, 1).Value = x
```

```
        Cells(i, 2).Value = F(x)
```

```
        Cells(i, 3).Value = DF(x)
```

```
        i = i + 1
```

```
    Loop Until Abs(x - x0) <= eps
```

```
    i = i - 1
```

```
    Set Reg = Range(Cells(i, 1), Cells(i, 2))
```

```
    Reg.Font.Size = 14
```

```
    Reg.Interior.ColorIndex = 6
```

```
End Sub
```

Функция DF вычисляет значение производной функции F. Точность вычислений введена в ячейку D2, а начальное приближение — в ячейку A2 рабочего листа. Найденные значения (корень и значение функции) выводятся увеличенным шрифтом, причем фон ячеек окрашен в желтый цвет.

7.9. Панель инструментов *Элементы управления*. Процедуры перелистывания рабочих листов

В данном разделе приведены примеры создания автоматизированного оглавления рабочей книги с помощью объектов панели инструментов **Элементы управления** (Forms), которая состоит из следующих кнопок:

	Надпись		Поле со списком
	Поле		Поле с раскрывающимся списком
	Группа		Полоса прокрутки
	Кнопка		Счетчик
	Флажок		Свойства элементов управления
	Переключатель		Текст программы
	Раскрывающийся список		Сетка привязки
	Список		Запуск окна диалога

Давайте попробуем создать рабочую книгу с пятью рабочими листами с именами *Оглавление*, *Май*, *Июнь*, *Июль* и *Вспомогательный*.

7.9.1. Первый пример

На лист модуля введите следующие три процедуры:

```
Sub Май ()
    Sheets ("Май").Activate
End Sub

Sub Июнь ()
    Sheets ("Июнь").Activate
End Sub

Sub Июль ()
    Sheets ("Июль").Activate
End Sub
```

В этих процедурах использовался метод `Activate`. Это метод активизирует объект, в данном случае — рабочий лист.

С помощью кнопки  создайте на ячейках C2, D3 и E4 рабочего листа Оглавление три кнопки (рис. 7.8).

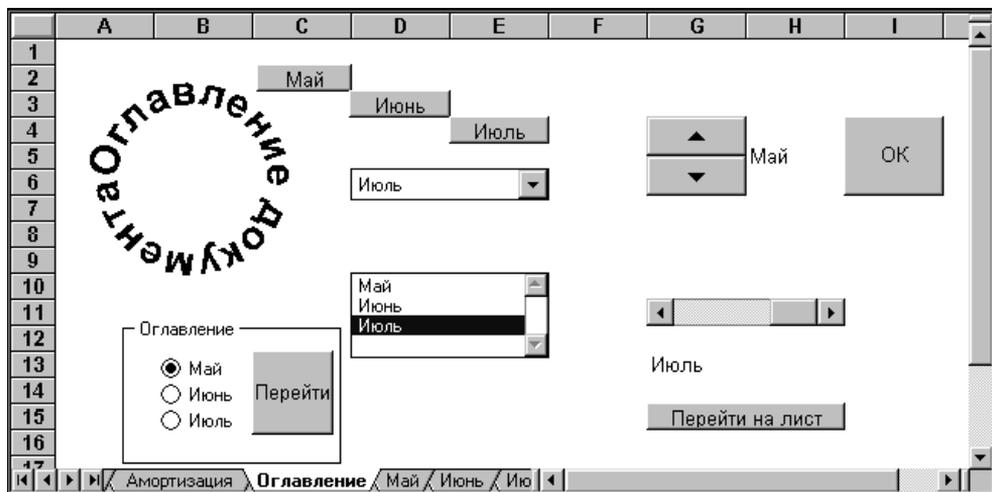


Рис. 7.8. Примеры автоматизированных оглавлений

По умолчанию создаваемые кнопки имеют имена Кнопка 1, Кнопка 2 и Кнопка 3, которые выводятся в поле имен и на поверхности кнопок. Щелкнув правой кнопкой мыши кнопку с именем Кнопка 1 и установив на ней указатель мыши, исправьте надпись с Кнопка 1 на Май. Эта операция не изменяет имени кнопки (оно будет прежним — Кнопка 1, и именно оно будет выводиться в поле имен). Щелкнув правой кнопкой мыши кнопку Май, раскройте контекстное меню, в котором выберите команду **Назначить макрос** (Assign Macro) (рис. 7.9).



Рис. 7.9. Выбор команды **Назначить макрос** с помощью контекстного меню

В поле ввода **Имя макроса/ссылка** (Macro Name/Reference) открывшегося диалогового окна **Назначить макрос объекту** (Assign Macro) введите Май (рис. 7.10).

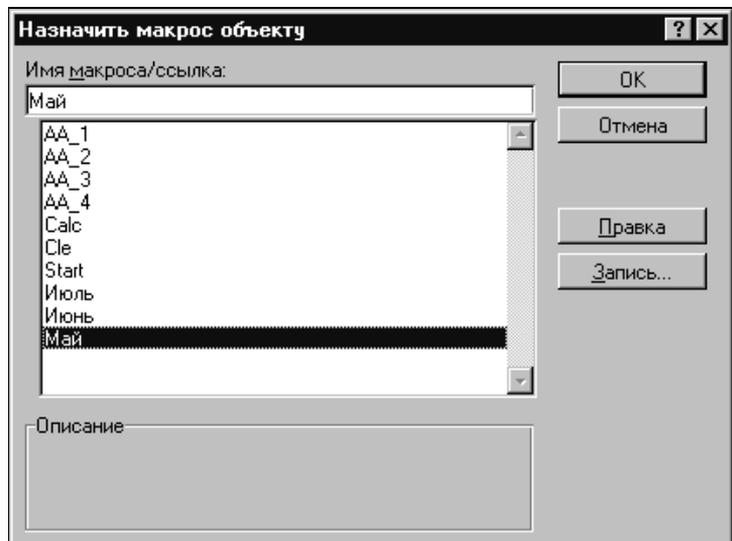


Рис. 7.10.
Диалоговое окно
Назначить макрос объекту

Нажмите кнопку **ОК**. Теперь кнопке Май назначена процедура Май, активизирующая рабочий лист с именем Май, т. е. нажатие этой кнопки будет приводить к переходу на лист с именем Май. Аналогично назначьте кнопкам Июнь и Июль процедуры Июнь и Июль, соответственно.

7.9.2. Второй пример

На рабочем листе **Вспомогательный** в ячейки А1, А2 и А3 введите:

Май
Июнь
Июль

С помощью команды **Вставка, Имя, Присвоить** (Insert, Name, Define) присвойте диапазону А1:А3 имя **Список**, ячейке А6 — имя **Номер**, а ячейке А7 — имя **Лист** (рис. 7.11).

В ячейку А7 введите формулу

=ИНДЕКС (Список;Номер;1)

	A	B	C	D	E
1	Май				
2	Июнь				
3	Июль				
4					
5					
6		3		1	
7	Июль			3	

Рис. 7.11. Данные на рабочем листе Вспомогательный

В общем случае синтаксис функции ИНДЕКС (INDEX) имеет следующий вид:

ИНДЕКС(массив; номер_строки; номер_столбца)

Функция ИНДЕКС возвращает значение ячейки диапазона массив с заданными номерами строки и столбца. Таким образом, при введении в ячейку Номер целого числа от 1 до 3 функция ИНДЕКС будет возвращать в ячейку Лист содержимое соответствующей ячейки диапазона Список. Например, если в ячейку Номер ввести число 3, то в ячейке Лист будем иметь Июль (рис. 7.11).

На листе модуля введите следующую процедуру

```
Sub AA_1()
Dim s As String
    s = Range("Лист").Value
    Sheets(s).Select
End Sub
```

Процедура AA_1 считывает значение из ячейки Лист в строковую переменную s и активизирует лист с именем, записанным в эту переменную.

С помощью кнопки  создайте на ячейках D6 и E6 рабочего листа Отглавление раскрывающийся список (рис. 7.8). Назначьте этому раскрывающемуся списку процедуру AA_1. Теперь, щелкнув список правой кнопкой мыши, выберите в раскрывшемся контекстном меню команду **Формат объекта** (Format Object). Откроется диалоговое окно **Форматирование объекта** (Format Object). Это окно можно открыть и другим способом: щелкнуть правой кнопкой мыши раскрывающийся список и нажать кнопку .

В диалоговом окне **Форматирование объекта** (Format Object) на вкладке **Элемент управления** (Control) в поле **Форматировать список по диапазону** (Input Range) введите Список, в поле **Помещать результат в ячейку** (Cell Link) — Номер, а в поле **Количество строк списка** (Lines) — 3, как показано на рис. 7.12.

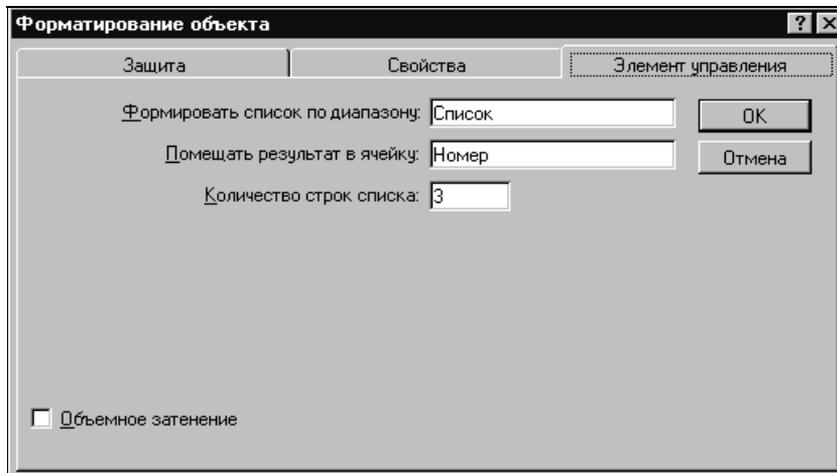


Рис. 7.12. Диалоговое окно **Форматирование объекта** для раскрывающегося списка

После нажатия кнопки **ОК** раскрывающийся список будет выводить значения, записанные в диапазон `Список`, а номер выбранного элемента из этого списка — в ячейку `Номер`. Таким образом будет обеспечиваться переход на страницу с именем, выбранным в раскрывающемся списке.

7.9.3. Третий пример

С помощью кнопки  создайте список на ячейках D10:E12 рабочего листа `Отглавление` (рис. 7.8). Назначьте этому списку процедуру `aa_1`. Кроме того, щелкнув список правой кнопкой мыши, выберите в раскрывшемся контекстном меню команду **Формат объекта** (Format Object). В открывшемся диалоговом окне **Форматирование объекта** (Format Object) в поле **Форматировать список по диапазону** (Input Range) введите `Список`, в поле **Помещать результат в ячейку** (Cell Link) — `Номер`. Установите переключатель в положение **Только одного значения** (рис. 7.13) и нажмите кнопку **ОК**.

Список выведет значения, записанные в диапазон `Список`, а номер выбранного элемента из этого списка — в ячейку `Номер`. Таким образом, данная конструкция будет обеспечивать переход на страницу с именем, выбранным в списке.

7.9.4. Четвертый пример

Присвойте ячейке `С6` рабочего листа вспомогательный имя `Первый_лист`. С помощью кнопки  создайте на диапазоне I4:I6 рабочего листа `Отглавление` кнопку, на поверхности которой напишите `OK` (рис. 7.8).

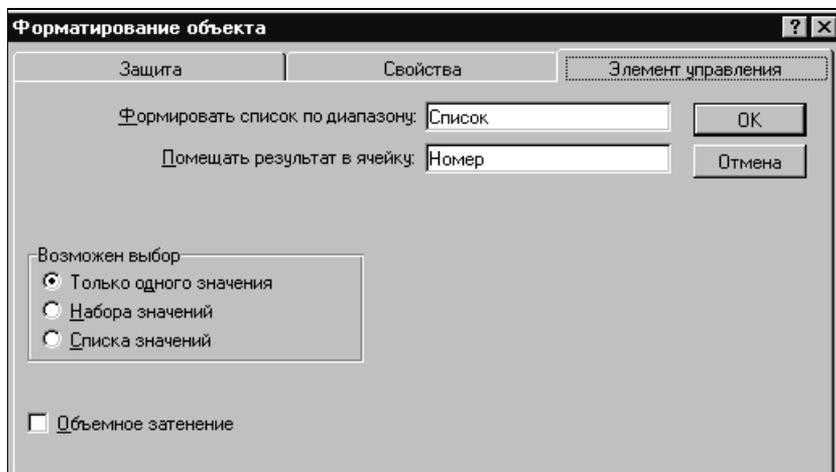


Рис. 7.13. Диалоговое окно **Форматирование объекта** для списка

С помощью кнопки  создайте счетчик на диапазоне G4:G6 (рис. 7.8). Ячейке H5 рабочего листа Оглавление присвойте имя Первый и введите в нее формулу

```
=ИНДЕКС(Список;Первый_лист;1)
```

На листе модуля введите процедуру

```
Sub AA_2()
Dim s As String
    s = Range("Первый").Value
    Sheets(s).Activate
End Sub
```

Назначьте эту процедуру созданной вами кнопке ОК. Щелкнув счетчик правой кнопкой мыши, выберите в раскрывшемся контекстном меню команду **Формат объекта** (Format Object). В открывшемся диалоговом окне **Форматирование объекта** (Format Object) в поле **Начальное значение** (Value) введите 2, в поле **Минимальное значение** (Min) — 1, в поле **Максимальное значение** (Max) — 3, в поле **Шаг изменения** (Step Value) — 1 и в поле **Помещать результат в ячейку** (Input Range) — Первый_лист (рис. 7.14).

Таким образом, счетчик будет изменять значения ячейки Первый_лист в диапазоне от 1 до 3, функция ИНДЕКС — выводить соответствующее значение диапазона Список в ячейку Первый, а процедура AA_2 — по щелчку на кнопке ОК осуществлять переход на лист с именем, введенным в ячейку Первый.

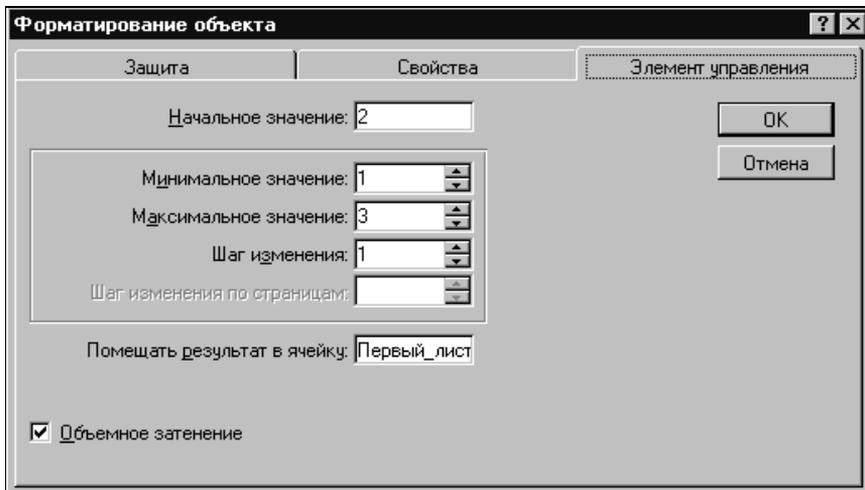


Рис. 7.14. Диалоговое окно **Форматирование объекта** для счетчика

7.9.5. Пятый пример

Присвойте ячейке **C7** рабочего листа **Вспомогательный имя Второй_лист**. С помощью кнопки  создайте на ячейках **G15:H15** рабочего листа **Оглавление** кнопку, на поверхности которой напишите **Перейти на лист**. С помощью кнопки  создайте полосу прокрутки на ячейках **G11:H11** (рис. 7.8). Ячейке **G13** рабочего листа **Оглавление** присвойте имя **Второй** и введите в нее формулу

```
=ИНДЕКС (Список;Второй_лист;1)
```

На листе модуля введите процедуру

```
Sub AA_3 ()
    Sheets (Range ("Второй").Value).Activate
End Sub
```

Назначьте эту процедуру кнопке **Перейти на лист**. Щелкнув правой кнопкой мыши полосу прокрутки, выберите в раскрывшемся контекстном меню команду **Формат объекта** (Object Format). Заполните открывшееся диалоговое окно **Форматирование объекта** (Format Object) так же, как в предыдущем примере, только в поле **Помещать результат в ячейку** (Input Range) введите **Второй_лист**.

7.9.6. Шестой пример

С помощью кнопки  создайте на ячейках C13:C15 рабочего листа **Оглавление** кнопку, на поверхности которой напишите **Перейти**. С помощью кнопки  создайте на диапазоне B12:C16 группу, а с помощью кнопки  на ячейках B13, B14 и B15 — три положения переключателя (рис. 7.8). На листе модуля введите процедуру

```
Sub AA_4()
Dim s As String
    Set NS = Sheets("Оглавление")
    If NS.OptionButtons(1).Value = xlOn Then s = "Май"
    If NS.OptionButtons(2).Value = xlOn Then s = "Июнь"
    If NS.OptionButtons(3).Value = xlOn Then s = "Июль"
    Sheets(s).Select
End Sub
```

Назначьте эту процедуру кнопке **Перейти**.

Приведем необходимые пояснения к процедуре AA_4.

OptionButtons (номер)	Объект управления — кнопка переключателя с указанным номером
Свойство Value объекта OptionButtons	Описывает положение переключателя: выбрана данная кнопка или нет
Постоянная xlOn, равная 1	Используется, чтобы описать ситуацию, когда кнопка переключателя выбрана

По мере создания элементов управления (полей, списков, переключателей и т. д.) Excel по умолчанию присваивает им общие порядковые номера, которые входят в имена, отображаемые в поле имени, например **Кнопка 1**, **Полоса прокр. 2**, **Перекл. 3**. Кроме того, внутри каждой группы однотипных элементов управления (например, группы переключателей) эти элементы также имеют порядковые номера. Например, если группа переключателей состоит из трех переключателей с именами **Перекл. 13**, **Перекл. 14** и **Перекл. 15**, то номера этих переключателей внутри группы будут **1**, **2** и **3**, соответственно. Вызвать объект управления можно как по порядковому номеру в группе однотипных элементов, так и по имени, т. е. допустима, например, следующая запись:

```
If NS.OptionButtons("Переключ. 13").Value = xlOn Then s = "Май"
If NS.OptionButtons("Переключ. 14").Value = xlOn Then s = "Июнь"
If NS.OptionButtons("Переключ. 15").Value = xlOn Then s = "Июль"
```

Внося изменения в поле имен, можно изменить имена объектов управления. Таким образом, процедура `AA_4`, в зависимости от включенного переключателя, присваивает переменной `s` одно из возможных значений `Май`, `Июнь` или `Июль`, а кнопка `Перейти` активизирует лист с соответствующим именем.

7.10. Создание сценариев

Рассмотрим пример процедуры пользователя для анализа сценариев. Создадим процедуры `Вариант_1`, `Вариант_2` и `Вариант_3`. При запуске этих процедур в ячейки будут введены значения, соответствующие этим вариантам. На рис. 7.15 приведен пример рабочей таблицы, обрабатывающей простую производственную модель с пятью переменными:

Стоимость_услуг	Ячейка B1
Стоимость_материала_1	Ячейка B2
Стоимость_материала_2	Ячейка B3
Количество_материала_1	Ячейка B5
Количество_материала_2	Ячейка B6

	А	В	С
1	Стоимость услуг	31	Вариант 1
2	Стоимость материала 1	59	Вариант 2
3	Стоимость материала 2	27	Вариант 3
4	Норма времени (часы)	12	
5	Количество материала 1	3	
6	Количество материала 2	4	
7	Себестоимость	657	
8	Товарная наценка	5%	
9	Отпускная цена	690	
10	Прибыль на одно изделие	33	
11	Количество изделий	36	
12	Суммарная прибыль	1183	

Рис. 7.15. Запрограммированные сценарии

Предположим, что компания производит изделие, для изготовления которого нужно затратить определенное количество рабочего времени и материалов. Себестоимость изделия вычисляется в ячейке `B7` по формуле

`=СУММПРОИЗВ(B1:B3;B4:B6)`

Отпускная цена вычисляется в ячейке B9 по формуле

$$=B7 * (1+B8)$$

Прибыль на одно изделие вычисляется в ячейке B10 по формуле

$$=B9-B7$$

Суммарная прибыль вычисляется в ячейке B12 по формуле

$$=B10*B11$$

Менеджер пытается предсказать суммарную прибыль, однако, у него нет точных сведений о том, каковы будут почасовая оплата труда, стоимость единицы материала и количество материалов. Эти данные определяются по трем сценариям, приведенным в табл. 7.6.

Таблица 7.6. Возможные варианты цен

Сценарии	1	2	3
Почасовая оплата	23	29	31
Цена единицы материала 1	62	55	59
Цена единицы материала 2	32	24	27
Количество материала 1	5	3	3
Количество материала 2	3	6	4

Следующие три процедуры, связанные с соответствующими кнопками, осуществляют ввод требуемых значений в ячейки таблицы:

```
Sub Вариант_1()
```

```
    Range("Стоимость_услуг") = 23
```

```
    Range("Стоимость_материала_1") = 62
```

```
    Range("Стоимость_материала_2") = 32
```

```
    Range("Количество_материала_1") = 5
```

```
    Range("Количество_материала_2") = 3
```

```
End Sub
```

```
Sub Вариант_2()
```

```
    Range("Стоимость_услуг") = 29
```

```
    Range("Стоимость_материала_1") = 55
```

```
Range("Стоимость_материала_2") = 24
Range("Количество_материала_1") = 3
Range("Количество_материала_2") = 6
End Sub
```

```
Sub Вариант_3()
    Range("Стоимость_услуг") = 31
    Range("Стоимость_материала_1") = 59
    Range("Стоимость_материала_2") = 27
    Range("Количество_материала_1") = 3
    Range("Количество_материала_2") = 4
End Sub
```

Данную задачу составления сценариев можно решить и более изящно с помощью следующей программы:

```
Sub Товар(a, p1, p2 As Double, n1, n2 As Integer)
    Range("Стоимость_услуг") = a
    Range("Стоимость_материала_1") = p1
    Range("Стоимость_материала_2") = p2
    Range("Количество_материала_1") = n1
    Range("Количество_материала_2") = n2
End Sub
```

```
Sub Вариант_1()
    Товар 23, 62, 32, 5, 3
End Sub
```

```
Sub Вариант_2()
    Товар 29, 55, 24, 3, 6
End Sub
```

```
Sub Вариант_3()
    Товар 31, 59, 27, 3, 4
End Sub
```

Рассмотрим еще один пример составления сценариев — начисления комиссионных. Пусть в некотором магазине продавцам начисляют комиссионные по следующему правилу:

- ❑ Если объем продаж не менее 100 000 руб., то комиссионные равны p_1 процентов от объема продаж
- ❑ Если объем продаж от 50 000 руб. до 100 000 руб., то комиссионные равны p_2 процентов от объема продаж
- ❑ Если объем продаж менее 50 000 руб., то комиссионные равны p_3 процентов от объема продаж

Приведенные ниже процедуры вычисляют комиссионные для двух сценариев: $p_1 = 4\%$, $p_2 = 3\%$, $p_3 = 2\%$ и $p_1 = 6\%$, $p_2 = 4\%$, $p_3 = 2\%$ (рис. 7.16). Заметим, что комиссионные, начисленные по максимальному проценту, будут окрашены красным цветом, по наименьшему — черным, а по среднему — синим.

	A	B	C	D	E
1	ФИО	Выручка	Комиссионные	Вариант 1	
2	Петров Н.П.	1000000	40000		
3	Иванов А.С.	5500000	220000	Вариант 2	
4	Сидоров Т.Е.	340000	13600		
5	Федоров А.Г.	390000	15600		
6	Кравцов Е.Л.	60600	1818		
7	Суслов Е.Г.	560000	22400		
8	Чичиков У.Ф.	5000	100		
9	Туркин Г.Т.	234600	9384		
10					
11					

Рис. 7.16. Сценарии начисления комиссионных

```

Dim fl As Integer
Function Re(i, p1, p2, p3 As Double, fl As Integer)
Dim s1, s2, s3 As Double
    If i >= 100000 Then
        s1 = p1
        fl = 1
    End If
    If i < 100000 And i >= 50000 Then
        s2 = p2
        fl = 2
    End If
    If i < 50000 Then
        s2 = p3
        fl = 3
    End If
    Re = (s1 + s2 + s3) * i

```

```
End Function

Sub Та(p1, p2, p3 As Double)
Dim i As Double
Dim j, n As Integer
n = Application.CountA(Sheets("Комиссионные").Range("B:B"))
For j = 2 To n
    i = Cells(j, 2).Value
    With Cells(j, 3)
        .Value = Re(i, p1, p2, p3, fl)
        Select Case fl
            Case Is = 1
                .Font.ColorIndex = 3
            Case Is = 2
                .Font.ColorIndex = 5
            Case Is = 3
                .Font.ColorIndex = 1
        End Select
    End With
Next j
End Sub

Sub Ком1()
    Та 0.04, 0.03, 0.02
End Sub

Sub Ком2()
    Та 0.06, 0.04, 0.02
End Sub
```

Приведем необходимое пояснение к данной программе. В VBA диапазон "B:B" обозначает столбец B. Функция СЧЕТЗ в русскоязычной версии Excel подсчитывает число непустых ячеек в выделенном диапазоне. Ее эквивалентом в англоязычной версии является функция CountA. Таким образом, функция

```
Application.CountA(Sheets("Комиссионные").Range("B:B"))
```

вычисляет число непустых ячеек в столбце B.

7.11. Модель управления запасами

Рассмотрим следующую простую модель управления запасами. Уличный продавец покупает журналы у издательства по цене 20 руб. за штуку, а продает — по 23 руб. за штуку. В случае, если товар не удастся реализовать, продавец возвращает его издательству по цене 17 руб. за штуку. Необходимо определить, сколько журналов следует закупать продавцу, чтобы его ожидаемая прибыль была максимальна.

Продавцу никогда не удавалось продать более 20 журналов, а в среднем за 37 дней объемы реализации и число соответствующих событий показаны на рис. 7.17 (для простоты учитываются только пачки по пять журналов).

	D	E	F	G	H	I	J	K	L
5	Продажа	Покупка	Возврат		Start				
6	23	20	17						
7	Объем реализации		0	5	10	15	20		
8	Число событий		0	7	9	15	6		
9	Вероятность события		0.00	0.19	0.24	0.41	0.16		
10			Продажа						
11			0	5	10	15	20		
12		0	0	0	0	0	0	Покупка	Прибыль
13		5	-15	15	15	15	15	0	0.00
14	Покупка	10	-30	0	30	30	30	5	15.00
15		15	-45	-15	15	45	45	10	24.32
16		20	-60	-30	0	30	60	15	26.35
17			Максимальная прибыль			26.35		20	16.22
18			Оптимальный объем			15			

Рис. 7.17. Модель управления запасами

В ячейку F9 введена формула

$$=F8/СУММ(\$F\$8:\$J\$8)$$

вычисляющая вероятность события. Эту формулу протаскиваем за маркер заполнения на диапазон F9:J9. Стоимость продукции вводится в ячейки D6 (продажа), E6 (покупка) и F6 (возврат) из диалоговых окон ввода (рис. 7.18) с помощью процедуры Calc, назначенной кнопке Start. Кроме того, процедура Calc выводит найденные значения максимальной прибыли и оптимального объема закупки газет в диалоговое окно, представленное на рис. 7.18.

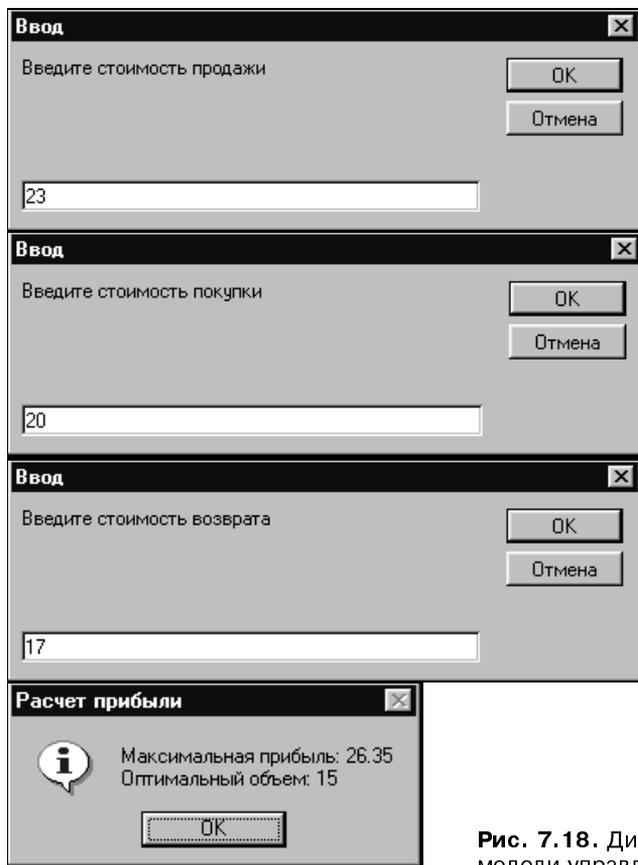


Рис. 7.18. Диалоговые окна модели управления запасами

```

Sub Calc()
Dim r, v As Double
    Range("продажа").Value = _
        InputBox("Введите стоимость продажи")
    Range("покупка").Value = _
        InputBox("Введите стоимость покупки")
    Range("возврат").Value = _
        InputBox("Введите стоимость возврата")
    r = Range("максимальная_прибыль").Value
    v = Range("оптимальный_объем").Value
    r = Format(r, "#.##")
    MsgBox "Максимальная прибыль: " & r & Chr(13) & _
        "Оптимальный объем: " & _
        v, vbInformation, "Расчет прибыли"
End Sub

```

В ячейках F12:J16 (рис. 7.17) с помощью функции пользователя Прибыль, которая будет описана ниже, вычисляются финансовые исходы при всех возможных вариантах событий покупки журналов и их реализации.

Option Base 1

Function Прибыль(покуп As Variant) As Variant

Dim NRows, i, j, Цпрод, Цпок, Цсдач As Integer, Res() As Integer

NRows = покуп.Rows.Count

Цпрод = Range("продажа").Value

Цпок = Range("покупка").Value

Цсдач = Range("возврат").Value

ReDim Res(NRows, NRows)

For i = 1 To NRows

For j = 1 To NRows

If i <= j Then Res(i, j) = покуп(i)*(Цпрод - Цпок)

If i > j Then Res(i, j) = покуп(j) * _
(Цпрод - Цпок) - (покуп(i) - покуп(j)) * (Цпок - Цсдач)

Next j

Next i

Прибыль = Res

End Function

В ячейках L13:L17 с помощью формулы

{=МУМНОЖ(F12:J16;ТРАНСП(F9:J9))}

находим ожидаемую прибыль, соответствующую различным вариантам покупки журналов. В ячейке I17 (максимальная прибыль) с помощью формулы

=НАИБОЛЬШИЙ(L13:L17;1)

вычисляем максимальную прибыль. Ее также можно найти, воспользовавшись функцией МАКС, находящей максимальный элемент из списка

=МАКС(L13:L17)

В ячейке I18 по формуле

=(ПОИСКПОЗ(НАИБОЛЬШИЙ(L13:L17;1);L13:L17;0)-1)*5

вычисляем соответствующий оптимальный объем покупок газет. Затем процедура Calc выводит эти оптимальные значения в окне сообщений.

Функция `НАИБОЛЬШИЙ` (`LARGE`) возвращает k -е наибольшее значение из множества данных. Эта функция используется, чтобы выбрать значение по его относительному местоположению. Например, функцию `НАИБОЛЬШИЙ` можно использовать, чтобы определить наилучший, второй или третий результат в баллах, показанный при тестировании. Приведем синтаксис использованной выше функции `НАИБОЛЬШИЙ`.

Синтаксис:

`НАИБОЛЬШИЙ` (массив; k)

Аргументы:

Массив	Массив или диапазон, для которых определяется k -е наибольшее значение
k	Позиция (начиная с наибольшей) в массиве или диапазоне

Приведем некоторые пояснения к процедуре Calc. Функция `MsgBox`, которая осуществляет вывод информации в окне сообщений, имеет следующий синтаксис:

`MsgBox` (текстовая_строка, кнопки, заголовок)

Аргументы:

текстовая_строка	Обязательный аргумент, задающий в окне выводимое информационное сообщение. Может состоять из нескольких текстовых строк, объединенных знаком <code>&</code> . Использование в этом аргументе <code>Chr(13)</code> приводит к переходу на новую строку при выводе информации
кнопки	Значения этого аргумента определяют категории появляющихся в окне кнопок. От значения аргумента <code>кнопки</code> зависит также, появляется ли в окне какой-либо значок. Если не указано, какие кнопки необходимо отображать в окне сообщений, то используется значение по умолчанию, соответствующее кнопке <code>ОК</code> . В табл. 7.7 приведены возможные комбинации кнопок и значков в окне сообщений
заголовок	Задает заголовок окна. По умолчанию используется заголовок <code>Excel</code>

Таблица 7.7. Допустимые значения переменной кнопки

Отображение	Аргумент
Кнопка ОК	VbOKOnly
Кнопки ОК и Отмена	VbOKCancel
Кнопки Да и Нет	VbYesNo
Кнопки Да, Нет и Отмена	VbYesNoCancel
Кнопки Прекратить, Повторить и Игнорировать	VbAbortRetryIgnore
Кнопки Повторить и Отмена	VbRetryCancel
Информационный знак	VbInformation
Знак вопроса	VbQuestion
Знак восклицания	VbExclamation

Функция `InputBox` осуществляет ввод значений с помощью окна ввода.

Синтаксис:

```
InputBox(текстовая_строка, заголовок)
```

Аргументы:

текстовая_строка	Обязательный аргумент. Задает в окне информационное сообщение, обычно поясняющее смысл вводимой величины
заголовок	Задает заголовок окна. По умолчанию используется заголовок Ввод

Функция `Format` осуществляет форматирование значения вычисляемого выражения по указанному формату.

Синтаксис:

```
Format(выражение, формат)
```

Аргументы:

выражение	Любое допустимое в VBA выражение
формат	Любое допустимое или определенное пользователем значение формата. Примером встроенного числового формата является <code>Fixed</code> , резервирующий две цифры после десятичной точки и, как минимум, одну — до нее

При построении пользовательского числового формата можно использовать следующие символы.

- Символ "0" резервирует позицию цифрового разряда. Отображает цифру или нуль. Если у числа, представленного аргументом, есть какая-нибудь цифра в той позиции разряда, в которой в строке формата находится "0", то функция отображает именно эту цифру аргумента, если нет — в этой позиции отображается нуль
- Символ "#" резервирует позицию цифрового разряда. Отображает цифру или ничего не отображает. Если у числа, представленного аргументом, есть какая-нибудь цифра в той позиции разряда, в которой в строке формата находится "#", функция отображает эту цифру аргумента, если нет — в этой позиции не отображается ничего. Действие этого символа аналогично действию "0", за исключением того, что нули в начале и конце числа не отображаются
- Символ "." резервирует позицию десятичного разделителя. Указание точки в строке формата определяет, сколько разрядов необходимо отображать слева и справа от десятичной точки
- Символ "%" резервирует процентное отображение числа
- Символ "," — разделитель разряда сотен от разряда тысяч
- Символ ":" — разделитель часов, минут и секунд в категории форматов **Время**
- Символ "/" — разделитель дня, месяца и года в категории форматов **Дата**
- Символы "E+", "E-", "e+", "e-" — разделители мантиссы и порядка в экспоненциальном формате

В процедуре Calc функция `Format(result, "#.##")` осуществляет форматирование значения, записанного в переменную `result`. Ее можно было бы записать с использованием встроенного формата следующим образом: `Format(result, "Fixed")`. Под дробную часть этого значения отводится две позиции и не менее одной позиции — под его целую часть. Приведем другие встроенные форматы:

General Number	Отображает число, как оно есть
Currency	Денежный формат с двумя цифрами после десятичной точки
Standard	Отображает число с разделителем тысяч и двумя цифрами после десятичной точки
Percent	Отображает число в формате процентов с двумя цифрами после десятичной точки
Scientific	Отображает число в стандартном экспоненциальном формате

Yes/No	Отображает No, если 0 и Yes — если 1
True/False	Отображает False, если 0 и True — если 1
On/Off	Отображает Off, если 0 и On — если 1
General Date	Отображает дату или время
Long Date	Отображает дату в длинном формате согласно системным установкам
Medium Date	Отображает дату в среднем формате согласно системным установкам
Short Date	Отображает дату в коротком формате согласно системным установкам
Long Time	Отображает время в длинном формате согласно системным установкам
Medium Time	Отображает время в среднем формате согласно системным установкам
Short Time	Отображает время в коротком формате согласно системным установкам

Отметим, что задачу о продаже журналов можно решить, не создавая массив финансовых исходов при всех возможных вариантах событий покупки журналов и их реализации, при помощи следующей программы:

```
Option Base 0
Sub Sale()
Dim пр, пр, вз As Double
Dim пр(4), SS(4) As Double
Dim i, j, n As Integer
Dim s1, s2 As Double
    n=4
    пр=Range("продажа").Value
    прк=Range("покупка").Value
    вз=Range("возврат").Value
    For j=0 To n
        For i=0 To n
            пр(i)=Cells(9, i + 6).Value
        Next i
        s1=0
        For i = 0 To j
```

```

        s1=s1+5*(i*(np-пк)-(j-i)*(пк-вз))*pr(i)
    Next i
    s2=0
    For i=j+1 To n
        s2=s2+5*j*(np-пк)*pr(i)
    Next i
    SS(j)=s1+s2
    Cells(j + 22, 14).Value = 5 * j
    Cells(j + 22, 15).Value = SS(j)
Next j
For j=0 To n
    If Cells(j+22, 15).Value= Application.Max _
        (Range(Cells(22, 15), Cells(n + 22, 15))) Then
        Cells(j + 22, 15).Font.ColorIndex = 3
        Cells(j + 22, 14).Font.ColorIndex = 3
    End If
Next j
End Sub

```

Программа Sale определяет оптимальный объем продаж и максимальную ожидаемую прибыль, выделяя их красным цветом.

Приведем еще один пример процедуры построения матрицы. Рассмотрим задачу построения матрицы G_C размерности $n_C \times n_C$

$$G_C(i, j) = \begin{cases} C^2(i), & i \leq j, \\ C(i - j) + C^3(i), & i > j, \end{cases}$$

где $n_C = 7$ и $C = (4, 3, 2, 1, 4, 3, 2)$.

На листе модуля напишем следующую функцию пользователя:

```

Option Base 1
Option Explicit
Function G(C As Variant) As Variant
Dim N, i, j As Integer, R() As Integer
    N = C.Columns.Count
    ReDim R(N, N)
    For i=1 To N
    For j = 1 To N
        If i <= j Then R(i, j) = C(i)^2

```

```

        If i > j Then R(i, j) = C(i - j) - C(i)^3
    Next j
Next i
G = R
End Function

```

Введем в диапазон ячеек A1:G1 компоненты вектора C . Выделим диапазон ячеек A3:G9, куда будет записана матрица G . В этот диапазон введем формулу $\{=G(A1:G1)\}$.

7.12. Задача об оптимальном раскрое

Рассмотрим следующую задачу: бумажная фабрика выпускает свою продукцию в виде бумажных рулонов стандартной длины, равной 20 футам. По специальным заказам потребителей поставляются рулоны и других размеров, для чего производится раскрой стандартных рулонов. Типичные заказы на рулоны нестандартных размеров приведены в табл. 7.8. Необходимо удовлетворить заказы, минимизируя отходы.

Таблица 7.8. Исходные данные задачи об оптимальном раскрое

Заказ	Требуемая длина рулона, фут	Требуемое количество рулонов
1	5	210
2	7	150
3	8	200
4	10	250

Для построения математической модели надо перебрать все возможные варианты раскроя рулона стандартной длины на рулоны требуемой длины. Данную задачу решает следующая программа:

```

Option Explicit
Sub Раскрой()
Dim R, i1, i2, i3, i4, s, t As Integer
Dim l, a1, a2, a3, a4, a5, m As Integer
    l = 20
    a1 = 5: a2 = 7
    a3 = 8: a4 = 10

```

```

R = 3
m = Application.Min(a1, a2, a3, a4)
t = Application.Floor(1/m, 1)
For i1 = 0 To t
For i2 = 0 To t
For i3 = 0 To t
For i4 = 0 To t
    s = 20 - a1*i1 - a2*i2 - a3*i3 - a4*i4
    If s >= 0 And s < m Then
        Cells(R, 1).Value = R - 2
        Cells(R, 2).Value = i1
        Cells(R, 3).Value = i2
        Cells(R, 4).Value = i3
        Cells(R, 5).Value = i4
        Cells(R, 6).Value = s
        R = R + 1
    End If
Next i4
Next i3
Next i2
Next i1

```

End Sub

где функция `Floor` возвращает наибольшее целое число, не превышающее данное. Результаты расчета процедуры `Раскрой` представлены на рис. 7.19 (они заполняют диапазон ячеек `A3:F12` рабочего листа).

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Заказанные длины рулонов								Заказанное количество				
2	Вариант	5	7	8	10	Остаток			210	150	200	250	Отходы
3	1	0	0	0	2	0	П	100	210	150	200	250	100
4	2	0	0	1	1	2	е	50					
5	3	0	0	2	0	4	р	0					
6	4	0	1	0	1	3	е	0	Варианты				
7	5	1	1	1	0	0	м	150					
8	6	1	2	0	0	1	е	0					
9	7	2	0	0	1	0	н	0					
10	8	2	0	1	0	2	н	0					
11	9	2	1	0	0	3	ы	0					
12	10	4	0	0	0	0	е	15					

Рис. 7.19. Варианты решения в задаче об оптимальном раскрое

Пусть x_j — количество стандартных рулонов, разрезанных по варианту j , где $j \in [1, 10]$. Ограничения, налагаемые на переменные x_j , связаны с требованием обеспечить изготовление заказанного количества нестандартных рулонов. Функция цели учитывает суммарные отходы, получаемые при выполнении заказа. Таким образом, имеем следующую математическую модель:

минимизировать:

$$z = 2x_2 + 4x_3 + 3x_4 + x_6 + 2x_8 + 3x_9 + \\ + 5(x_5 + x_6 + 2x_7 + 2x_8 + 2x_9 + 4x_{10} - 210) + \\ + 7(x_4 + x_5 + 2x_6 + 2x_9 - 150) + \\ + 8(x_2 + 2x_3 + x_5 + x_8 - 200) + \\ + 10(2x_1 + x_2 + x_4 + x_7 - 250)$$

при ограничениях: x_i — целые при $i \in [1, 10]$ и

$$\begin{aligned} x_5 + x_6 + 2x_7 + 2x_8 + 2x_9 + 4x_{10} &\geq 210, \\ x_4 + x_5 + 2x_6 + x_9 &\geq 150, \\ x_2 + 2x_3 + x_5 + x_8 &\geq 200, \\ 2x_1 + x_2 + x_4 + x_7 &\geq 250. \end{aligned}$$

Отведем диапазон ячеек Н3:Н12 под переменные. Введем в диапазон ячеек I3:L3 левые части ограничений, определенные следующими формулами:

=СУММПРОИЗВ(\$Н\$3:\$Н\$12;В3:В12)

=СУММПРОИЗВ(\$Н\$3:\$Н\$12;С3:С12)

=СУММПРОИЗВ(\$Н\$3:\$Н\$12;D3:D12)

=СУММПРОИЗВ(\$Н\$3:\$Н\$12;Е3:Е12)

В ячейку М3 введем функцию цели

=СУММПРОИЗВ(\$Н\$3:\$Н\$12;F3:F12)

+В2*(СУММПРОИЗВ(\$Н\$3:\$Н\$12;В3:В12)-I2)

+С2*(СУММПРОИЗВ(\$Н\$3:\$Н\$12;С3:С12)-J2)

+D2*(СУММПРОИЗВ(\$Н\$3:\$Н\$12;D3:D12)-K2)

+Е2*(СУММПРОИЗВ(\$Н\$3:\$Н\$12;Е3:Е12)-L2),

где в ячейки В2:Е2 введены длины, а в ячейки I2:L2 — количество заказанных рулонов.

Выберем команду **Сервис, Поиск решения** (Tools, Solver) и заполним открывшееся диалоговое окно **Поиск решения** (Solver), как показано на

рис. 7.20. Результаты расчета приведены на рис. 7.19. Оптимальным является использование первого варианта раскроя для 100 стандартных рулонов, второго — для 50, пятого — для 150 и десятого — для 15 стандартных рулонов.

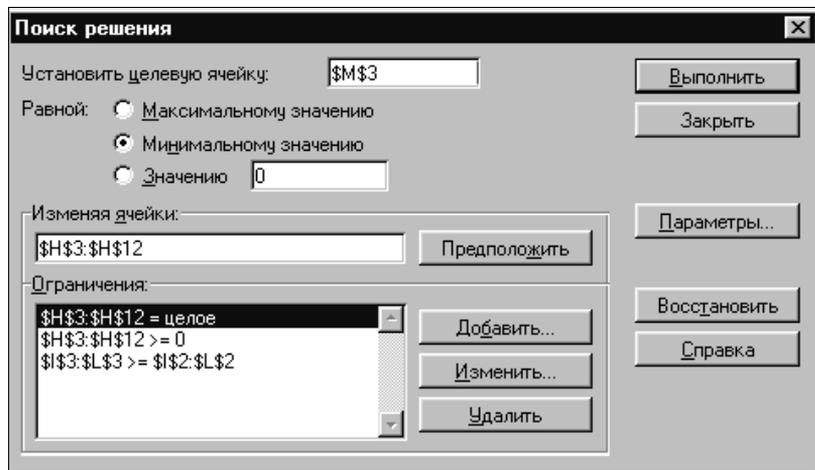


Рис. 7.20. Диалоговое окно **Поиск решения** в задаче об оптимальном раскрое

7.13. База данных отеля

Приведем пример программы, создающей простую базу данных регистрации клиентов отеля на рабочем листе **База данных** (рис. 7.21).

	A	B	C	D	E	F	G	H
	Фамилия	Имя	Пол	Тип номера	Оплачено	Паспорт сдан	Продолжительность проживания, суток	Ввод данных
1								
2	Петров	Иван	муж	Двухместный	да	нет	11	
3	Федоров	Петр	муж	Люкс	нет	да	13	
4	Крепс	Ирина	жен	Одноместный	да	нет	15	
5								
6								
7								

Рис. 7.21. База данных отеля

База данных будет заполняться программой. Программе не требуются названия полей, но для облегчения ориентации пользователя в первой строке рабочего листа **База данных** введем их названия, как показано на рис. 7.21.

Выделите вторую строку рабочего листа База данных и выполните команду **Окно, Закрепить области** (Window, Freeze Panes). Теперь заголовок будет присутствовать на экране постоянно при просмотре любых записей базы.

Нажмем кнопку **Ввод данных**. Теперь с помощью открывшегося диалогового окна **Регистрация клиентов** будет заполняться база данных (рис. 7.22).

Рис. 7.22. Диалоговое окно **Регистрация клиентов**

Диалоговое окно **Регистрация клиентов** создается на листе диалога **Регистрация** средствами панели инструментов **Элементы управления** (Forms). Напомним, что диалоговый лист создается командой **Вставка, Макрос, Диалоговое окно** (Insert, Macro, DialogSheet) (рис. 7.23). С помощью вкладки **Элемент управления** (Control) диалогового окна **Форматирование объекта** (Format Object) убедитесь в правильном назначении функций кнопкам диалогового окна.

Для работы с раскрывающимся списком на вспомогательном рабочем листе **Переменные** в диапазон A1:A3 с именем **Типы_номеров** вводим последовательно типы номеров: **Люкс**, **Одноместный**, **Двухместный**. Ячейке A15 присваиваем имя **Номер**, а ячейке A14 — имя **Тип_номера**. Кроме того, в ячейку A14 вводим следующую формулу

```
=ИНДЕКС(Типы_номеров;Номер;1)
```

Щелкнув список правой кнопкой мыши, с помощью контекстного меню вызовем диалоговое окно **Форматирование объекта** (Format Object). В поле **Форматировать список по диапазону** (Input Range) введем имя диапазона **Типы_номеров**, а в поле **Помещать результат в ячейку** (Cell Link) введем имя ячейки **Номер**. Затем назначим процедуру **EnterData** кнопке **Ввод данных**. Со счетчиком диалогового окна ввода данных свяжем процедуру **DoSpinner**, выводящую в третье поле ввода текущее значение счетчика, а с третьим полем

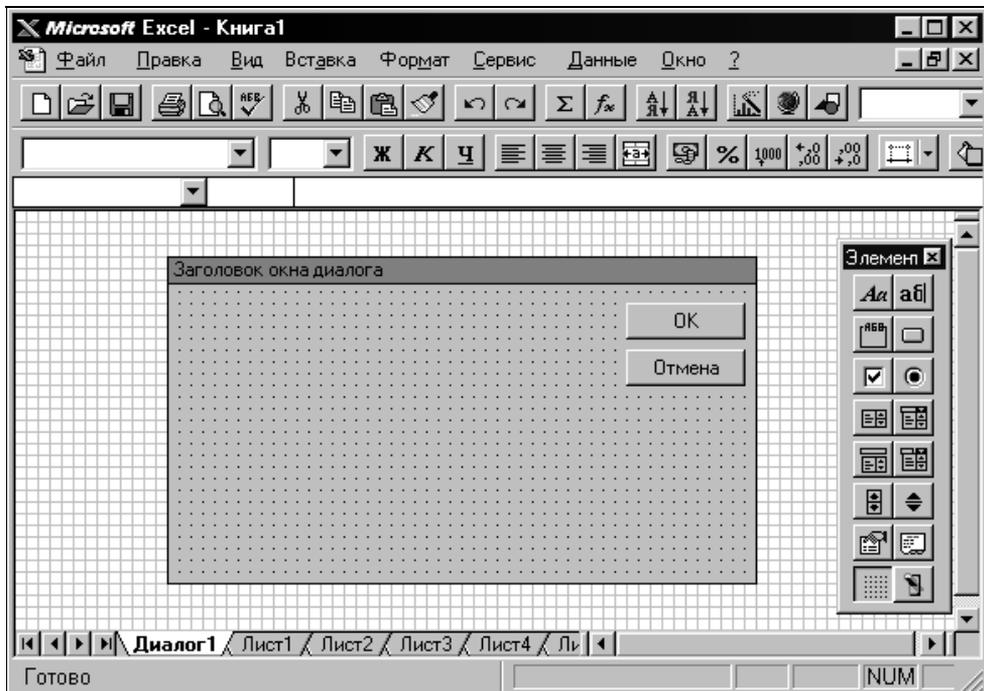


Рис. 7.23. Диалоговый лист

ввода — процедуру BackSpinner для синхронизации значения счетчика с содержимым поля ввода.

```

Sub EnterData()
Dim Фамилия, Имя, Срок, Оплачено, Паспорт As String
Dim Выб_номер, Пол As String
Dim DboxOK As Boolean
Dim Row As Integer
Start:
    With DialogSheets("Регистрация")
        .EditBoxes(1).Text=""
        .EditBoxes(2).Text=""
        .EditBoxes(3).Text=""
        .CheckBoxes(1).Value =xlOff
        .CheckBoxes(2).Value =xlOff
        .Spinners(1).Min=1
        .Spinners(1).Max=100
    End With

```

```
DboxOK = DialogSheets("Регистрация").Show
If Not DboxOK Then Exit Sub
Row=Application.CountA(Sheets("База данных").Range("с:с"))+1
With DialogSheets("Регистрация")
    Фамилия=.EditBoxes(1).Text
    Имя=.EditBoxes(2).Text
    Срок=.EditBoxes(3).Text
    If .OptionButtons(1)=xlOn Then Пол="муж"
    If .OptionButtons(2)=xlOn Then Пол="жен"
    If .CheckBoxes(1)=xlOn Then
        Оплачено="да"
    Else
        Оплачено="нет"
    End If
    If .CheckBoxes(2)=xlOn Then
        Паспорт="да"
    Else
        Паспорт="нет"
    End If
End With
Выб_номер = Worksheets("Переменные").Range("Тип_номера")
With Sheets("База данных")
    .Cells(Row, 1).Value = Фамилия
    .Cells(Row, 2).Value = Имя
    .Cells(Row, 3).Value = Пол
    .Cells(Row, 4).Value = Выб_номер
    .Cells(Row, 5).Value = Оплачено
    .Cells(Row, 6).Value = Паспорт
    .Cells(Row, 7).Value = Срок
End With
GoTo Start
End Sub

Sub DoSpinner()
    With ActiveDialog
        .EditBoxes(3).Text = CStr(.Spinners(1).Value)
    End With
End Sub
```

```
Sub BackSpinner()  
    With ActiveDialog  
        .Spinners(1).Value = CInt(.EditBoxes(3).Text)  
    End With  
End Sub
```

Дадим необходимые пояснения к приведенной выше программе. CheckBoxes, EditBoxes и Spinners являются наборами объектов управления: флажок, поле и счетчик, соответственно. Другими наборами объектов управления являются:

ListBoxes	Список
DropDowns	Раскрывающийся список
ScrollBars	Полоса прокрутки
Labels	Надпись

Свойства Min и Max определяют минимальное и максимальное значения счетчика с указанным номером, т. е. конструкция

```
DialogSheets("Регистрация").Spinners(1).Min = 1
```

задает минимальное значение первого счетчика, равное 1.

Конструкция

```
DialogSheets("Регистрация").EditBoxes(1).Text = ""
```

очищает поле ввода с номером 1.

Оператор With-End With используется для удобства при работе с несколькими свойствами или методами одного и того же объекта. Например, конструкция

```
With DialogSheets("Регистрация")  
    .EditBoxes(1).Text = ""  
    .EditBoxes(2).Text = ""  
    .EditBoxes(3).Text = ""  
    .CheckBoxes(1).Value = xlOff  
    .CheckBoxes(2).Value = xlOff  
    .Spinners(1).Min = 1  
    .Spinners(1).Max = 100  
End With
```

эквивалентна следующей конструкции (но без использования With-End With):

```
DialogSheets ("Регистрация").EditBoxes(1).Text = ""
DialogSheets ("Регистрация").EditBoxes(2).Text = ""
DialogSheets ("Регистрация").EditBoxes(3).Text = ""
DialogSheets ("Регистрация").CheckBoxes(1) = xlOff
DialogSheets ("Регистрация").CheckBoxes(2) = xlOff
DialogSheets ("Регистрация").Spinners(1).Min = 1
DialogSheets ("Регистрация").Spinners(1).Max = 100
```

В переменную Row вводится номер первой пустой строки на рабочем листе База данных. Оператор безусловного перехода GoTo позволяет изменять порядок выполнения команд. Так, в нашем примере после выполнения оператора GoTo Start управление передается оператору с меткой Start. Метка в тексте программы отличается от других идентификаторов тем, что после нее стоит двоеточие.

Рассмотрим подробнее следующую конструкцию:

```
DboxOK = DialogSheets ("Регистрация").Show
If Not DboxOK Then Exit Sub
```

Свойство Show возвращает True, если диалоговое окно открыто, и False — в противном случае. Таким образом, данная конструкция при нажатии в диалоговом окне кнопки Отмена присваивает переменной DboxOK значение False, что приводит к завершению работы процедуры EnterData. Если кнопка Отмена не нажата, то переменной DboxOK присваивается значение True, и процедура EnterData продолжает работу по заполнению базы данных.

Рассмотрим другой способ выполнения этой же операции — без использования дополнительной переменной и оператора безусловного перехода:

```
Do While DialogSheets ("Регистрация").Show
    БЛОК_ОПЕРАТОРОВ
Loop
```

Метод Show позволяет отобразить диалоговое окно. Например, диалоговое окно Регистрация может быть отображено на экране с помощью команды

```
Sheets ("Регистрация").Show
```

Диалоговое окно останется на экране до тех пор, пока не будет нажата кнопка **OK** или кнопка **Отмена**. Применяв метод `Hide`, можно также закрыть это окно. Команда

```
Sheets("Регистрация").Hide
```

закроет ранее открытое диалоговое окно. Существует различие между методом `Show`, который отображает диалоговое окно, и методом `Activate`, активизирующим лист диалогового окна. Метод `Show` отображает диалоговое окно в рабочем режиме для работы с данными, а метод `Activate` возвращает его в режиме конструирования как часть листа диалогового окна.

Конструкция

```
If .OptionButtons(1) = xlOn Then Пол = "муж"
```

присваивает переменной `Пол` значение "муж", если переключатель поставлен в первое положение.

Следует также отметить, что при обращении из программы к элементам управления в диалоге важно знать порядок их расположения. Порядок расположения элементов — это порядок их обхода с помощью клавиши `<Tab>`. Изначально порядок расположения управляющих элементов соответствует порядку их создания. Затем его можно изменять с помощью диалогового окна **Последовательность перехода** (Tab Order), вызываемого командой **Сервис, Последовательность перехода** (Tools, Tab Order), не затрагивая при этом самих элементов (рис. 7.24).

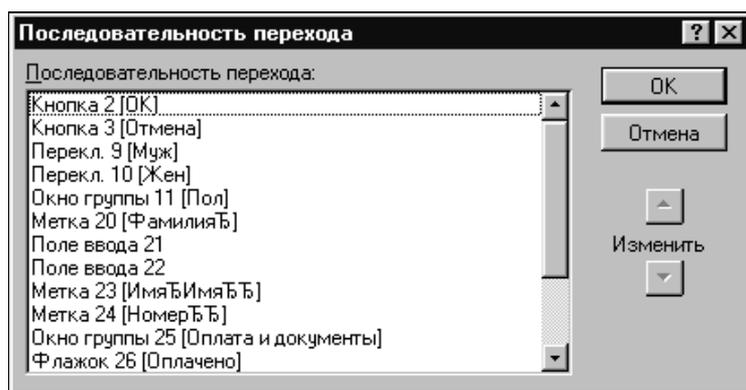


Рис. 7.24. Диалоговое окно **Последовательность перехода**

Отметим, что порядковый номер элементов управления при их выделении выводится в поле имен.

7.14. Банковская база данных

Рассмотрим простую базу данных учета клиентов банка. База данных будет заполняться на рабочем листе База данных. В рабочий лист Архив записывается информация об операциях со счетами клиентов. На рабочем листе Меню имеются четыре кнопки: Выход, Изменение информации, Прием вклада и Переучет архива, управляющие работой с базой данных (рис. 7.25).

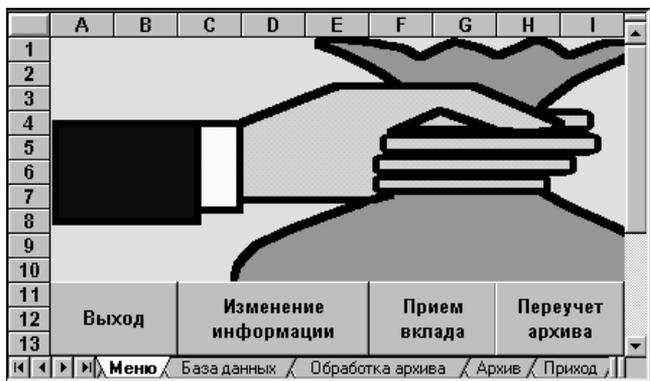


Рис. 7.25.
Рабочий лист Меню
банковской базы данных

На листе диалога Приход создано диалоговое окно внесения поступивших денег в базу данных. Если в базе данных клиент не обнаружен, то появляется соответствующее сообщение (рис. 7.26).

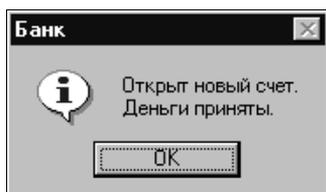


Рис. 7.26. Сообщение
об открытии счета

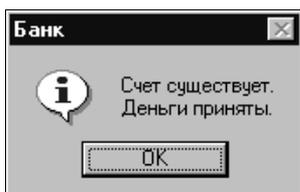


Рис. 7.27. Сообщение о принятии
денег на имеющийся счет

Если в базе данных такой клиент имеется, то его вклад добавляется к текущему счету и появляется сообщение (рис. 7.27).

На рабочем листе База данных введите заголовки, как это показано на рис. 7.28.

Будем считать, что тип вклада может быть Текущий, Депозит или Срочный, а отделение банка — Северное, Западное или Центральное.

На рабочем листе Архив введите заголовки, как показано на рис. 7.29.

На листе диалога Приход создайте диалоговое окно ПРИНЯТИЕ ВКЛАДА (рис. 7.30).

	A	B	C	D	E	F
	Фамилия	Тип вклада	Размер вклада	Отделение банка	Примечание	
1						
2	Сидоров	Текущий	23100000	Северное	Временно выехал	
3	Фролов	Депозит	4159000	Центральное		
4	Сидорчук	Текущий	123	Западное		
5	Вагнер	Текущий	3888000	Северное	Сменил адрес	
6	Петров	Текущий	21758	Северное		
7	Иванов	Депозит	559000	Центральное		
8	Григорьев	Текущий	5000500	Центральное		
9	Чернов	Текущий	73000000	Северное		
10	Кравцов	Текущий	1760300	Северное	Перевел в другое отделение	
11	Рябов	Текущий	77200000	Западное		

Рис. 7.28. Рабочий лист База данных банковской базы данных

	A	B	C	D	E	F	G
	Получатель	Дата	Фамилия	Тип вклада	Выданные деньги	Отделение банка	Примечание
1							
2	Петров	02.06.98	Фролов	Депозит	100	Западное	По доверенности
3	Смирнов	05.06.98	Петров	Текущий	5198	Западное	По доверенности
4		09.06.98	Иванов	Депозит	3572	Западное	По доверенности
5	Петров	09.08.98	Иванов	Депозит	572	Западное	По доверенности
6		10.08.98	Сидоров	Депозит	572	Западное	Лично
7		11.08.98	Федотов	Депозит	572	Западное	Лично
8		13.08.98	Федотов	Депозит	1572	Западное	Лично
9	Смирнов	27.07.98	Петров	Текущий	3000	Северное	По доверенности
10		10.08.98	Сидоров	Депозит	572	Западное	Лично
11		01.08.98	Петров	Текущий	2	Северное	
12		01.08.98	Фром	Текущий	230000	Северное	

Рис. 7.29. Рабочий лист Архив банковской базы данных

ПРИНЯТИЕ ВКЛАДА [X]

Фамилия

Тип вклада

Размер вклада

Отделения банка

Северное

Западное

Центральное

Примечание

Рис. 7.30. Диалоговое окно ПРИНЯТИЕ ВКЛАДА

Полю фамилия присвойте имя фамилия. Для переименования поля выделите его и введите новое имя в поле имен, расположенное левее строки формул. Раскрывающему списку присвойте имя Тип_вклада, полям Размер вклада и Примечание — имена Размер_вклада и Remarks, соответственно. В VBA в именах переменных и объектов буква "ч" часто воспринимается программой как ошибка. Поэтому вместо русского имени Примечание используем английское Remarks. Переключателям присваиваем имена Северное, Западное и Центральное. Щелкнув правой кнопкой мыши поле Remarks, с помощью контекстного меню вызовем диалоговое окно **Форматирование объекта** (Format Object), в котором полю Remarks зададим режим **Многострочный текст** (Multiline Edit) (рис. 7.31).

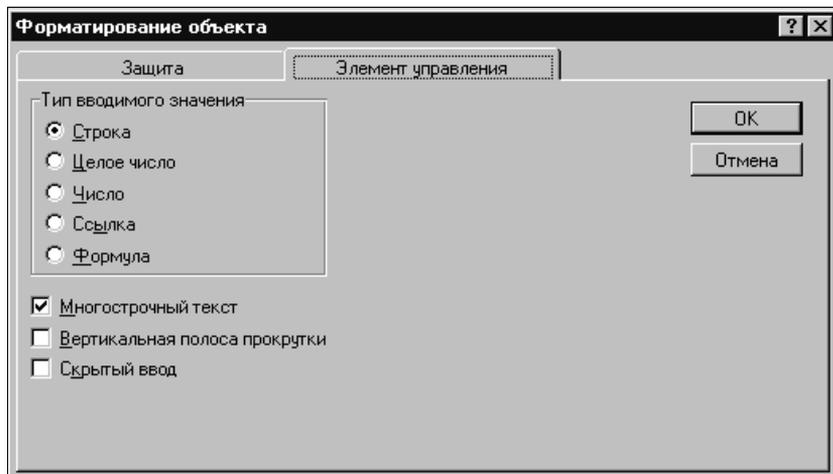


Рис. 7.31. Диалоговое окно **Форматирование объекта** для поля

На листе диалога Работа создайте диалоговое окно Работа со вкладом (рис. 7.32).

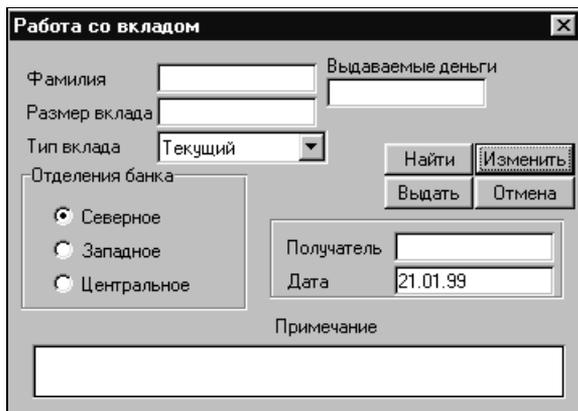


Рис. 7.32. Создаваемое диалоговое окно для работы со вкладом

Присвойте объектам те же имена, что и в диалоговом окне ПРИНЯТИЕ ВКЛАДА (рис. 7.30). Кроме того, полям Выдаваемые деньги, Получатель и Дата присвойте имена Выдаваемые_деньги, Фамилия_п и Дата, соответственно. Для поля ввода Remarks задайте режим **Многострочный текст** (Multiline Edit) в диалоговом окне **Форматирование объекта** (Format Object) (рис. 7.31). С помощью окна диалога **Форматирование объекта** (Format Object) назначьте кнопке Выдать функцию **Закреть** (Dismiss), кнопке Изменить — функции **Не определена** и **Закреть** (Dismiss), кнопке Отмена — функцию **Отмена** (Cancel) и кнопке Найти — ни одной функции (рис. 7.33).

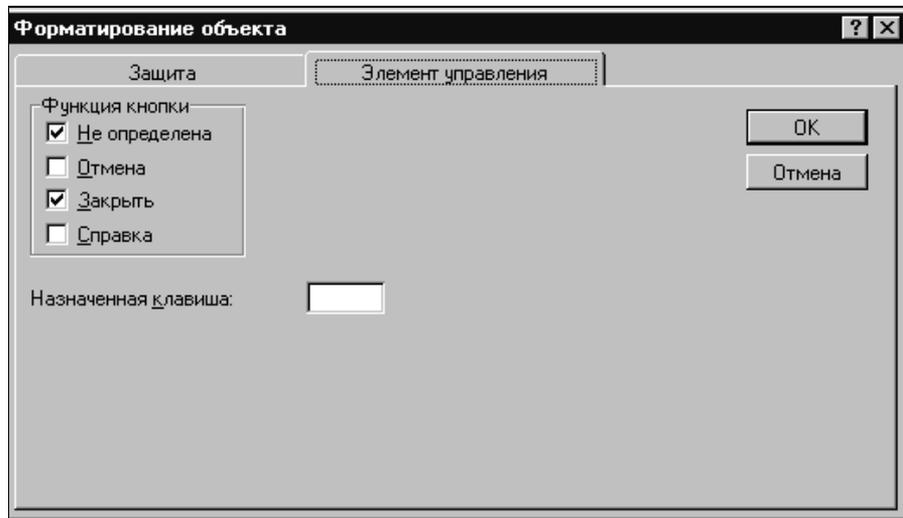


Рис. 7.33. Диалоговое окно **Форматирование объекта** для кнопки

На листе диалога **Обработка архива** создайте окно диалога со списком фамилий, показанное на рис. 7.34.

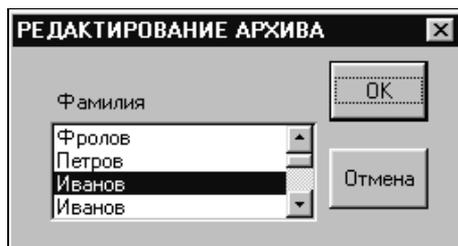


Рис. 7.34. Диалоговое окно для редактирования архива

Для этого списка с помощью диалогового окна **Форматирование объекта** (Format Object) установите переключатель **Возможен выбор** (Selection) в положение **Списка значений** (MultiSelect) (рис. 7.35).

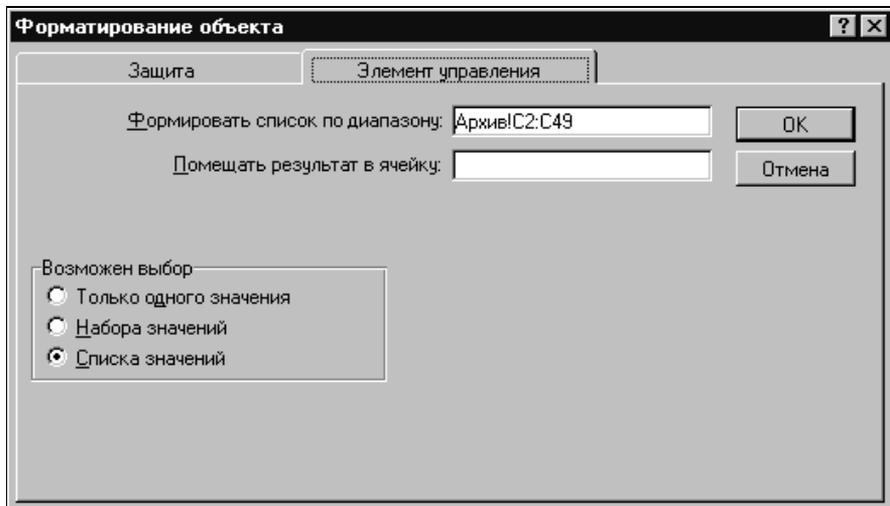


Рис. 7.35. Диалоговое окно **Форматирование объекта** для списка

Данная установка позволит с помощью шелчка мышью выбирать из списка архива группу фамилий вкладчиков и объединять их операции. Объединенный результат операций будет помещен в архив в строчку, соответствующую первому клиенту из выбранной группы.

На листе `Модуль1` введите следующую программу для работы с базой данных:

```
Option Explicit
```

```
Option Base 1
```

```
Dim Данные(5) As Variant
```

```
Dim Получатель(2) As String
```

```
Dim Текущий As Integer
```

```
Dim Область As Object
```

```
Dim Число As Long
```

```
Dim Количество As Integer
```

```
Dim Номер, Вклад As Integer
```

```
Const Диалог_прихода As Integer = 1
```

```
Const Диалог_обработки As Integer = 2
```

```
Sub Подготовка(Диалог)
```

```
    Dim Объект_диалога As Object
```

```
    Select Case Диалог
```

```
        Case Диалог_прихода
```

```
Set Объект_диалога = DialogSheets("Приход")
Case Диалог_обработки
Set Объект_диалога = DialogSheets("Работа")
With Объект_диалога
.DrawingObjects("Фамилия_п").Text = ""
.DrawingObjects("Дата").Text = Format(Date, "Short Date")
End With
End Select
```

```
With Объект_диалога
.DrawingObjects("Фамилия").Text = ""
.DrawingObjects("Тип_вклада").DropDownLines = 3
.DrawingObjects("Тип_вклада").List = _
Array("Текущий", "Депозит", "Срочный")
.DrawingObjects("Тип_вклада").Value = 1
.DrawingObjects("Размер_вклада").Text = ""
.DrawingObjects("Remarks").Text = ""
End With
```

```
If Диалог = Диалог_обработки Then
With DialogSheets("Работа")
.DrawingObjects("Выдаваемые_деньги").Text = ""
.DrawingObjects("Фамилия_п").Text = ""
End With
End If
```

```
End Sub
```

```
Sub Загрузка_диалога(Диалог As Integer)
```

```
Dim Объект_диалога As Object
```

```
Select Case Диалог
```

```
Case Диалог_прихода
```

```
Set Объект_диалога = DialogSheets("Приход")
```

```
Case Диалог_обработки
```

```
Set Объект_диалога = DialogSheets("Работа")
```

```
With Объект_диалога
```

```
Получатель(1) = .DrawingObjects("Фамилия_п").Text
```

```
Получатель(2) = .DrawingObjects("Дата").Text
```

```
End With
```

```
End Select
```

```
With Объект_диалога
Данные(1) = .DrawingObjects("Фамилия").Text
Данные(2) = .DrawingObjects("Тип_вклада") _
.List(.DrawingObjects("Тип_вклада").ListIndex)
Данные(3) = .DrawingObjects("Размер_вклада").Text
If .DrawingObjects("Северное") = xlOn Then
Данные(4) = "Северное"
End If
If .DrawingObjects("Западное") = xlOn Then
Данные(4) = "Западное"
End If
If .DrawingObjects("Центральное") = xlOn Then
Данные(4) = "Центральное"
End If
Данные(5) = .DrawingObjects("Remarks").Text
End With
```

```
End Sub
```

```
Sub Запись_клиента()
    Подготовка(Диалог_прихода)
    DialogSheets("Приход").Show
End Sub
```

```
Sub Работа_с_клиентом()
    Подготовка(Диалог_обработки)
    Текущий = 0
    DialogSheets("Работа").Show
End Sub
```

```
Sub Выход()
    ThisWorkbook.Close
End Sub
```

```
Sub Взнос_вклада()
Dim Найдено As Boolean
Dim Столбец As Integer
    Загрузка_диалога (Диалог_прихода)
    Найдено = False
Set Область = Sheets("База данных").Cells(1, 1).CurrentRegion
```

```
Количество = Область.Rows.Count
Номер = 2
With Sheets("База данных")
While Not Найдено And Номер <= Количество
If .Cells(Номер, 1) = Данные(1) Then
Найдено = True
Else
Номер = Номер + 1
End If
Wend
If Найдено Then
.Cells(Номер, 3) = .Cells(Номер, 3) + Данные(3)
MsgBox "Счет существует." & Chr(13) & _
        "Деньги приняты.", vbInformation, "Банк"
Else
For Столбец = 1 To 5
.Cells(Количество + 1, Столбец) = Данные(Столбец)
Next Столбец
MsgBox "Открыт новый счет." & Chr(13) & _
        "Деньги приняты.", vbInformation, "Банк"
End If
End With
End Sub
```

```
Sub Поиск_клиента()
Dim Measure As Long
Dim Найдено As Boolean
    Загрузка_диалога (Диалог_обработки)
    Найдено = False
Set Область = Sheets("База данных").Cells(1, 1).CurrentRegion
Количество = Область.Rows.Count
Номер = 2
With Sheets("База данных")
While Not Найдено And Номер <= Количество
If .Cells(Номер, 1) = Данные(1) Then
Найдено = True
End If
Номер = Номер + 1
Wend
```

```
If Найдено Then
Текущий = Номер - 1
Select Case .Cells(Текущий, 2)
Case Is = "Текущий"
    Measure = 1
Case Is = "Депозит"
    Measure = 2
Case Is = "Срочный"
    Measure = 3
End Select
DialogSheets("Работа"). _
    DrawingObjects("Тип_вклада").Value = Measure
Select Case .Cells(Текущий, 4)
Case Is = "Северное"
    DialogSheets("Работа"). _
        DrawingObjects("Северное").Value = xlOn
Case Is = "Западное"
    DialogSheets("Работа"). _
        DrawingObjects("Западное").Value = xlOn
Case Is = "Центральное"
    DialogSheets("Работа"). _
        DrawingObjects("Центральное").Value = xlOn
End Select
DialogSheets("Работа"). _
DrawingObjects("Remarks").Text = .Cells(Текущий, 5)
If DialogSheets("Работа").DrawingObjects("Размер_вклада"). _
Text <> "" Then
Число = DialogSheets("Работа"). _
DrawingObjects("Размер_вклада").Text
Else
Число = 0
DialogSheets("Работа"). _
DrawingObjects("Размер_вклада").Text = .Cells(Текущий, 3)
End If
If Число <= .Cells(Текущий, 3) Then
MsgBox "Клиент найден." & Chr(13) & _
    "Данные о счете помещены в окно диалога.", _
vbExclamation, "Банк"
Else
```

```
DialogSheets("Работа").DrawingObjects("Размер_вклада") _  
.Text = .Cells(Текущий, 3)  
MsgBox "Клиент найден." & Chr(13) & _  
    "На счету имеется меньше денег," & Chr(13) & _  
    "чем запрашивает клиент." & Chr(13) & _  
    "Данные о счете помещены в окно диалога.", _  
    vbExclamation, "Банк"  
  
End If  
Else  
MsgBox "Такой клиент не найден.", vbCritical, "Банк"  
End If  
End With  
  
End Sub  
  
Sub Снятие_денег()  
Dim Строка_архива As Long  
    If Текущий = 0 Then  
        MsgBox "Перед выдачей денег надо найти счет.", _  
            vbExclamation, "Банк"  
    Else  
        Загрузка_диалога (Диалог_обработки)  
        If DialogSheets("Работа"). _  
            DrawingObjects("Выдаваемые_деньги").Text <> "" Then  
            Число = CLng(DialogSheets("Работа"). _  
                DrawingObjects("Выдаваемые_деньги").Text)  
            If Число <= Sheets("База данных").Cells(Текущий, 3) Then  
                Set Область = Sheets("Архив").Cells(1, 1).CurrentRegion  
                Строка_архива = Область.Rows.Count + 1  
                With Sheets("База данных")  
                    .Range(.Cells(Текущий, 1), .Cells(Текущий, 5)).Copy  
                End With  
                With Sheets("Архив")  
                    .Paste destination:= _  
                    .Range(.Cells(Строка_архива, 3), .Cells(Строка_архива, 7))  
                End With  
                With Worksheets("Архив")  
                    .Cells(Строка_архива, 1).Value = Получатель(1)  
                    .Cells(Строка_архива, 2).Value = Получатель(2)  
                    .Cells(Строка_архива, 5).Value = _
```

```
DialogSheets("Работа"). _  
DrawingObjects("Выдаваемые_деньги").Text  
.Cells(Строка_архива, 7).Value = _  
DialogSheets("Работа"). _  
DrawingObjects("Remarks").Text  
.Cells(Строка_архива, 5).Value = Число  
End With  
With Sheets("База данных")  
.Cells(Текущий, 3).Value = .Cells(Текущий, 3).Value - Число  
If .Cells(Текущий, 3).Value = 0 Then  
.Cells(Текущий, 3).EntireRow.Delete  
MsgBox "Деньги выданы." & Chr(13) & _  
    "Счет закрыт в связи с отсутствием на нем денег.", _  
    vbInformation, "Банк"  
Else  
MsgBox "Деньги выданы." & Chr(13) & _  
    "На счету имеется остаток " & _  
    & CStr(.Cells(Текущий, 3).Value) & _  
    " р.", vbInformation, "Банк"  
End If  
End With  
Else  
MsgBox "Деньги не выданы." & Chr(13) & _  
    "На счету нет необходимой суммы денег." & Chr(13) & _  
    "Клиент запрашивает на " & CStr(Число - _  
Sheets("База данных").Cells(Текущий, 3).Value) & _  
    "р. больше чем имеется на счету." & _  
    , vbExclamation, "Банк"  
End If  
Else  
MsgBox "Нельзя выдать деньги" & Chr(13) & _  
    "без указания суммы", vbExclamation, "Банк"  
End If  
End If
```

```
End Sub
```

```
Sub Изменить()
```

```
    If Текущий = 0 Then  
        MsgBox "Перед изменением описания" & Chr(13) & _
```

```
"следует найти клиента", vbExclamation, "Банк"  
Else  
With DialogSheets("Работа")  
Sheets("База данных").Cells(Текущий, 5).Value = _  
    .DrawingObjects("Remarks").Text  
Select Case Sheets("База данных").Cells(Текущий, 2)  
Case "Северное"  
    .DrawingObjects("Северное").Value = xlOn  
Case "Западное"  
    .DrawingObjects("Западное").Value = xlOn  
Case "Центральное"  
    .DrawingObjects("Центральное").Value = xlOn  
End Select  
End With  
MsgBox "Изменено описание доступных " _  
& Chr(13) & _  
"для изменения данных.", vbInformation, "Банк"  
End If
```

```
End Sub
```

На листе Модуль2 введите следующую программу для работы с архивом:

```
Dim Area As Object  
Dim n, i As Integer  
Dim s() As Integer  
  
Sub Раб_арх()  
    Set Area=Sheets("Архив").Cells(1, 1).CurrentRegion  
    n=Area.Rows.Count  
    DialogSheets("Обработка архива"). _  
        ListBoxes(1).MultiSelect = xlExtended  
    Do While DialogSheets("Обработка архива").Show=True  
        DialogSheets("Обработка архива"). _  
            ListBoxes(1).RemoveAllItems  
        DialogSheets("Обработка архива"). _  
            ListBoxes(1).ListFillRange=_  
            "=Архив!c2:c" & LTrim(CStr(n))  
        DialogSheets("Обработка архива"). _  
            ListBoxes(1).Selected(3)=True
```

```

Loop
End Sub

Sub Обработка()
Dim Su, Fl As Integer
Dim a As Boolean
    Su = 0
    For i = 1 To n - 1
        If DialogSheets("Обработка архива"). _
            ListBoxes(1).Selected(i)=True Then
            Su=Su+Sheets("Архив").Cells(i + 1, 5).Value
        End If
    Next i
    For i=1 To n-1
        If DialogSheets("Обработка архива"). _
            ListBoxes(1).Selected(i)=True Then
            Fl=i
            Exit For
        End If
    Next i
    Sheets("Архив").Cells(Fl+1, 5).Value=Su
    For i=1 To n-1
        If DialogSheets("Обработка архива"). _
            ListBoxes(1).Selected(i)=True _
            And i>Fl Then
            Sheets("Архив").Cells(i+1,1).EntireRow.Delete
        End If
    Next i
End Sub

```

Для завершения создания системы учета денежных вкладов необходимо связать данные программы с кнопками в соответствии с табл. 7.9, а программу Обработка — со списком диалогового окна РЕДАКТИРОВАНИЕ АРХИВА.

Таблица 7.9. Соответствие между процедурами и кнопками

Кнопка	Программа
Выход	Выход
Изменение информации	Работа_с_клиентом

Таблица 7.9 (окончание)

Кнопка	Программа
Прием вклада	Запись_клиента
Принять	Внос_вклада
Найти	Поиск_клиента
Выдать	Снятие_денег
Изменить	Изменить
ОК	Раб_арх

Приведем некоторые пояснения к тексту программы.

Метод `DrawingObjects` (переменная) возвращает объект управления или графический объект с именем или номером, помещенным в переменную. Свойство `CurrentRegion` возвращает объект — диапазон ячеек, окружающий заданную ячейку и ограниченный пустыми ячейками. Таким образом в следующей группе операторов переменной `Область` назначается диапазон ячеек рабочего листа `База данных`, содержащий всю внесенную информацию о клиентах банка. В переменную `Количество` записывается число заполненных строк в базе данных

```
Dim Область As Object
Set Область = Sheets("База данных").Cells(1, 1).CurrentRegion
Количество = Область.Rows.Count
```

Копирование содержимого диапазона ячеек производится методом `Copy`. Например, в нашей программе копирование диапазона ячеек из строки с номером `Текущий` осуществляется с помощью следующей конструкции:

```
Sheets("База данных"). _
Range(.Cells(Текущий, 1), .Cells(Текущий, 5)).Copy
```

Метод `Paste` копирует содержимое буфера обмена, записанное во встроенную переменную `destination`, в указанный диапазон ячеек. Таким образом, следующая конструкция записывает информацию из буфера обмена в диапазон ячеек рабочего листа `Архив`:

```
Sheets("Архив").Paste destination:= _
.Range(.Cells(Строка_архива, 3), .Cells(Строка_архива, 7))
```

Отметим, что существует более компактный способ копирования диапазонов ячеек, например, следующая процедура

```
Sub Co()
    Range("A1:C1").Copy (Range("B2"))
End Sub
```

копирует содержимое диапазона ячеек A1:C1 рабочего листа в диапазон B2:D2 того же листа.

Процедура

```
Sub Mu()
    Range("A1:C1").Cut (Range("B2"))
End Sub
```

перемещает диапазон ячеек.

Метод `EntireRow` возвращает строку, содержащую указанную строку. Таким образом, конструкция

```
Sheets("Архив").Cells(i+1,1).EntireRow.Delete
```

удаляет строку, содержащую ячейку `Cells(i+1, 1)`.

Приведем краткий обзор методов и свойств следующих объектов: список и раскрывающийся список.

Метод <code>RemoveAllItems</code>	Удаляет из списка все элементы
Метод <code>RemoveItem(index, count)</code>	Удаляет из списка элементы, число которых равно <code>count</code> , начиная с номера <code>index</code> . Если <code>count</code> опущен, то удаляется только один элемент
Свойство <code>DropDownLines</code>	Определяет число элементов, выводимых в раскрывающемся списке
Свойство <code>ListCount</code>	Возвращает число элементов списка
Свойство <code>ListIndex</code>	Определяет (как и свойство <code>Value</code>) номер активного элемента раскрывающегося списка (например <code>DrawingObjects("Тип_вклада").ListIndex=1</code>)
Свойство <code>MultiSelect</code>	Позволяет выбирать несколько элементов из списка. Свойство <code>MultiSelect</code> может принимать одно из следующих значений: <code>xlNone</code> — для выбора одного элемента из списка, а <code>xlSimple</code> или <code>xlExtended</code> — для выбора группы элементов из списка

Свойство Selected Возвращает True, если элемент списка с указанным номером выбран, и возвращает False — в противном случае

Таким образом, конструкция

```
For i=1 To n-1
If DialogSheets("Обработка архива").ListBoxes(1).Selected(i) _
    = True Then
    Su = Su + Sheets("Архив").Cells(i+1, 5).Value
End If
Next i
```

суммирует значения ячеек пятого столбца, соответствующие выбранным элементам раскрывающегося списка.

Задать элементы списка можно одним из следующих способов:

Метод ListFillRange Назначает списку указанный диапазон. Например, в нашей программе конструкция

```
DialogSheets("Обработка архива"). _
ListBoxes(1).ListFillRange = _
"=Архив!c2:c" & LTrim(CStr(n))
```

назначает диапазон ячеек C2:Cn рабочего листа Архив. Функция LTrim возвращает копию строки без предшествующих пробелов. Другими полезными функциями работы со строкой являются RTrim и Trim, возвращающие копии строки без пробелов в конце строки и без пробелов в обоих концах строки, соответственно

Свойство List Задает все элементы списка или элемент с указанным номером. Таким образом, следующая конструкция создает раскрывающийся список:

```
.DrawingObjects("Тип_вклада"). _
List = Array("Текущий", "Депозит", "Срочный")
```

Метод AddItem(text, index) Определяет в качестве элемента списка с номером, записанным в переменную index, значение текстовой переменной text. Например, следующая конструкция определяет тот же раскрывающийся список, что и в предыдущем примере:

```
.DrawingObjects("Тип_вклада"). _
AddItem 1, "Текущий"
.DrawingObjects("Тип_вклада"). _
AddItem 2, "Депозит"
.DrawingObjects("Тип_вклада"). _
AddItem 3, "Срочный"
```

Опишем действие подпрограмм, входящих в программу работы с банковской базой данных.

Подготовка (Диалог)	Готовит диалоговое окно для нового взноса денег, если переменная <code>Диалог</code> равна постоянной <code>Диалог_прихода</code> , и для работы со счетом, если переменная <code>Диалог</code> равна постоянной <code>Диалог_обработки</code> . Кроме того, она считывает информацию в массив <code>Данные</code> из объектов диалогового окна
Выход	Закрывает рабочую книгу
Запись_клиента	Активизирует диалоговое окно <code>Приход</code> и подготавливает его к работе
Взнос_вклада	Осуществляет поиск клиента по фамилии. Если такого клиента нет, то она осуществляет запись счета нового клиента в первую свободную строку базы данных и выдает соответствующее сообщение (рис. 7.26). Если такой клиент существует, то подпрограмма <code>Взнос_вклада</code> прибавляет взнос к ранее открытому счету клиента и выдает соответствующее сообщение (рис. 7.27)
Работа_с_клиентом	Активизирует диалоговое окно <code>Работа со вкладом</code> (рис. 7.32) и подготавливает его к работе
Поиск_клиента	Осуществляет поиск клиента по фамилии. Если клиент найден, то появляется сообщение, показанное на рис. 7.36, а если клиент не найден — сообщение, показанное на рис. 7.37
Изменить	Позволяет изменить примечание и запись об отделении банка. После ее выполнения появляется сообщение, показанное на рис. 7.38

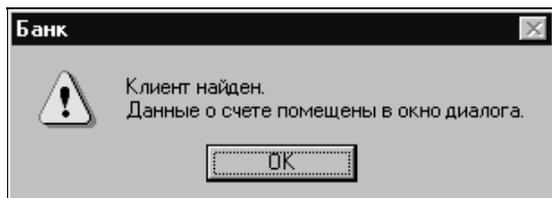


Рис. 7.36. Результат поиска клиента (если клиент найден)

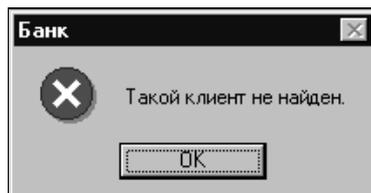


Рис. 7.37. Результат поиска клиента (если клиент не найден)

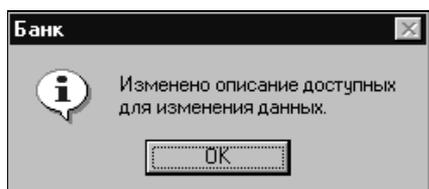


Рис. 7.38. Сообщение об изменении описания

При работе с базой данных возможны также следующие сообщения:

- ❑ При нажатии кнопок в диалоговом окне Работа со вкладом (рис. 7.32) без предварительного поиска клиента возможно появление сообщения, показанного на рис. 7.39.

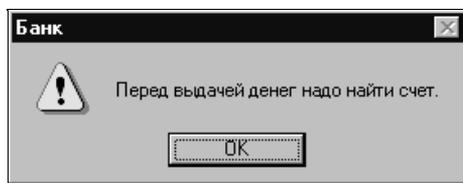


Рис. 7.39. Сообщение, предупреждающее о необходимости поиска счета

- ❑ При работе программы Снятие_денег с найденным счетом возможно появление одного из сообщений, показанных на рис. 7.40.

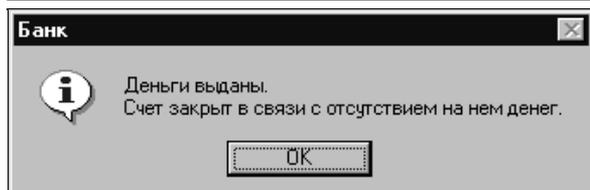
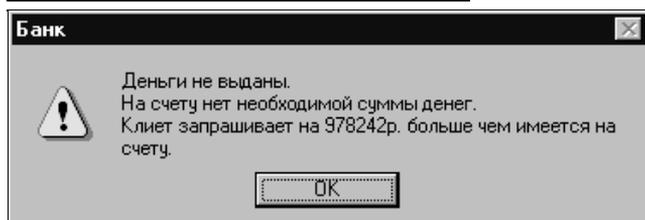
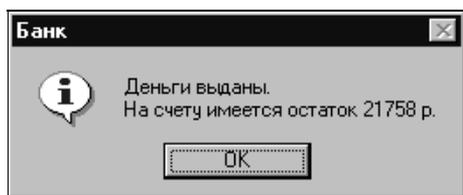


Рис. 7.40. Сообщения о результате выдачи денег

7.15. Как разместить капитал?

Рассмотрим следующую задачу. Фирма по производству быстрого питания намерена сделать капиталовложения в размере 5 млн. руб. в расширение производства. Фирма имеет четыре филиала, расположенные в различных городах. В каждом из филиалов проведено изучение рынка и найдены мате-

математические ожидания прибыли как функции капиталовложений, приведенные в табл. 7.10. Необходимо выработать оптимальный план капиталовложений, максимизирующий ожидаемую прибыль.

Таблица 7.10. Ожидаемая прибыль в зависимости от капиталовложений

Филиалы	Млн. руб.					
	0	1	2	3	4	5
1-й	0	0.28	0.45	0.65	0.78	0.90
2-й	0	0.25	0.41	0.55	0.65	0.75
3-й	0	0.15	0.25	0.40	0.50	0.62
4-й	0	0.20	0.33	0.42	0.48	0.53

Пусть:

- $R(i, j)$ — прибыль, получаемая от вложения i млн. руб. в j -й филиал, где $i \in [0, 5]$, $j \in [1, 4]$
- $F(A, 1, 2)$ — оптимальное распределение средств, когда A млн. руб. вкладываются в 1 и 2-й филиалы вместе
- $F(A, 1, 2, 3)$ — оптимальное распределение средств, когда A млн. руб. вкладываются в 1, 2 и 3-й филиалы вместе
- $F(A, 1, 2, 3, 4)$ — оптимальное распределение средств, когда A млн. руб. вкладываются в 1, 2, 3 и 4-й филиалы вместе

Очевидно, что

$$F(A, 1, 2) = \max\{R(A - i, 1) + R(i, 2): i \in [0, A]\},$$

$$F(A, 1, 2, 3) = \max\{F(A - i, 1, 2) + R(i, 3): i \in [0, A]\},$$

$$F(A, 1, 2, 3, 4) = \max\{F(A - i, 1, 2, 3) + R(i, 4): i \in [0, A]\}.$$

Значения i , при которых достигается максимум, определяют оптимальные капиталовложения в филиалы.

Введите на листе модуля следующую программу, вычисляющую оптимальные капиталовложения по указанным выше формулам:

```
Option Explicit
```

```
Option Base 1
```

```
Sub DP()
```

```
Dim i, j, k, n, p, l, t As Integer
```

```
Dim m, R(), A() As Double
```

```
k = 5
ReDim R(k+1, 4), A(k+1)
For i = 1 To k+1
For j = 2 To 5
    R(i, j-1) = Cells(i+2,j).Value
Next j
Next i
t=2
For p=2 To 4
    If p=2 Then
        For j=1 To k+1
            A(j) = Cells(j+2,2).Value
        Next j
    End If
    If p > 2 Then
        For j=1 To k+1
            A(j) = Cells(j+2,p+3).Value
        Next j
    End If
    For n=1 To k+1
        m=-1
        For j=1 To n
            If m < A(j)+R(n+1-j,p) Then
                m = A(j)+R(n+1-j, p)
            End If
        Next j
        Cells(n+2, 4+p).Value = m
        l=t
        For j=1 To n
            If m = A(j)+R(n+1-j,p) Then
                Cells(n+5+k,l).Value = j-1
                Cells(n+5+k,l+1).Value = n-j
                l=l+2
            End If
        Next j
    Next n
    t=1
Next p
End Sub
```

Переменной k присваивается значение, равное объему капиталовложений. В массив R с рабочего листа Капиталовложения (рис. 7.41) вводится ожидаемая прибыль, распределенная по филиалам.

	A	B	C	D	E	F	G	H	I
1		Прибыль				Максимумы			
2	Млн. руб.	1	2	3	4	1 и 2	1, 2 и 3	1, 2, 3 и 4	
3	0	0	0	0	0	0	0	0	
4	1	0.28	0.25	0.15	0.2	0.28	0.28	0.28	
5	2	0.45	0.41	0.25	0.33	0.53	0.53	0.53	
6	3	0.65	0.55	0.4	0.42	0.7	0.7	0.73	
7	4	0.78	0.65	0.5	0.48	0.9	0.9	0.9	
8	5	0.9	0.75	0.62	0.53	1.06	1.06	1.1	
9									
10									
11	0	0	0	0	0	0	0		Распределение капиталовложений
12	1	1	0	1	0	1	0		
13	2	1	1	2	0	2	0		
14	3	2	1	3	0	2	1		
15	4	3	1	4	0	4	0		
16	5	3	2	5	0	4	1		
17									
18	Млн. руб.	1	2	1 и 2	3	1, 2 и 3	4		

Рис. 7.41. Результат нахождения оптимальной схемы капиталовложений

В диапазон ячеек F3:H8 программа DP выводит значения $F(A, 1, 2)$, $F(A, 1, 2, 3)$ и $F(A, 1, 2, 3, 4)$. В диапазон ячеек B11:G16 программа DP выводит оптимальное распределение капиталовложения по филиалам. Из рис. 7.41 видно, что максимальная ожидаемая прибыль равна 1,1 млн. руб., а оптимальные капиталовложения состоят в выделении 1-му филиалу 3 млн. руб., 2-му филиалу — 1 млн. руб. и 4-му филиалу — 1 млн. руб.

7.16. Расчет краткосрочной ссуды с использованием диалогового окна

В данном разделе рассмотрим пример оформления расчета краткосрочной ссуды с использованием диалогового окна.

Создайте рабочую книгу, состоящую из трех листов: рабочего — Расчет ссуды, диалогового — Ссуда и листа модуля. В ячейку D1 рабочего листа Расчет ссуды введем размер ссуды, в ячейку D2 — дату выдачи ссуды, в ячейку D3 — дату возврата ссуды, в ячейку D4 — процентную ставку и в ячейку D5 — возвращаемую сумму денег. На листе диалога Ссуда создайте диалоговое окно Расчет краткосрочных ссуд, как показано на рис. 7.42.



Рис. 7.42. Расчет краткосрочной ссуды

Все расчеты и заполнение ячеек рабочего листа будет осуществлять следующая программа, которую необходимо набрать на листе модуля:

```

Sub Диалог()
    DialogSheets("Ссуда").EditBoxes(5).Enabled = False
    DialogSheets("Ссуда").Show
End Sub

Sub Ссуда()
    Dim P, R, i As Double
    Dim D_beginning, D_end As Date
    With DialogSheets("Ссуда")
        P = CDb1(.EditBoxes(1).Text)
        D_beginning = CDate(.EditBoxes(2).Text)
        D_end = CDate(.EditBoxes(3).Text)
        i = CDb1(.EditBoxes(4).Text)/100
    End With
    If D_end < D_beginning Then
        MsgBox "Ошибка в датах", _
            vbExclamation, "Расчет ссуды"
        Exit Sub
    End If
    R = P*(1+i)^((D_end - D_beginning)/365)

```

```

R = Format(R, "Fixed")
DialogSheets("Ссуда").EditBoxes(5).Text = CStr(R)
Range("D1").Value = P
Range("D2").Value = D_beginning
Range("D3").Value = D_end
Range("D4").Value = i
Range("D5").Value = R

```

End Sub

Sub C1()

```

With DialogSheets("Ссуда")
    .EditBoxes(1).Text = ""
    .EditBoxes(2).Text = ""
    .EditBoxes(3).Text = ""
    .EditBoxes(4).Text = ""
    .EditBoxes(5).Text = ""

```

End With

```
Range("D1:D5").Clear
```

End Sub

Назначьте процедуры следующим кнопкам:

Диалог	Кнопке Расчет ссуды рабочего листа Расчет ссуды
Ссуда	Кнопке Вычислить диалогового окна Расчет краткосрочных ссуд
C1	Кнопке Очистить диалогового окна Расчет краткосрочных ссуд

В процедуре Ссуда проверяется корректность ввода дат, которая в случае несогласования дат выдает сообщение (рис. 7.43).

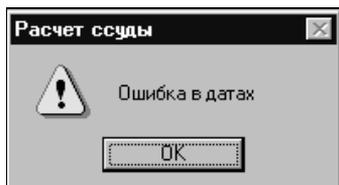


Рис. 7.43. Сообщение о некорректном вводе даты

Приведем необходимые пояснения к программе.

Свойство Enabled определяет возможность внесения изменений в объекты управления (изменение положение переключателя, ввод текста в поле ввода

и т. д.). Задание этого свойства равным `True` позволяет вносить изменения в объект, а `False` — не позволяет. В нашем примере в пятое поле ввода выводятся числа, вычисленные программой. Для того чтобы их невозможно было изменить, в процедуру `Диалог` включена команда

```
DialogSheets ("Студа").EditBoxes (5).Enabled = False
```

7.17. Расчет числа выплат с использованием диалогового окна

В данном разделе рассмотрим пример оформления расчета числа периодических выплат при фиксированной процентной ставке с использованием диалогового окна.

Создайте рабочую книгу, состоящую из трех листов: рабочего — `Расчет срока платежей`, диалогового — `Выплаты` и листа модуля. В ячейку `D1` рабочего листа `Расчет срока платежей` введем размер ссуды, в ячейку `D2` — процентную ставку, в ячейку `D3` — величину периодических выплат и в ячейку `D5` — вычисляемое число выплат. На листе диалога `Выплаты` создайте диалоговое окно `Расчет числа выплат`, как показано на рис. 7.44.

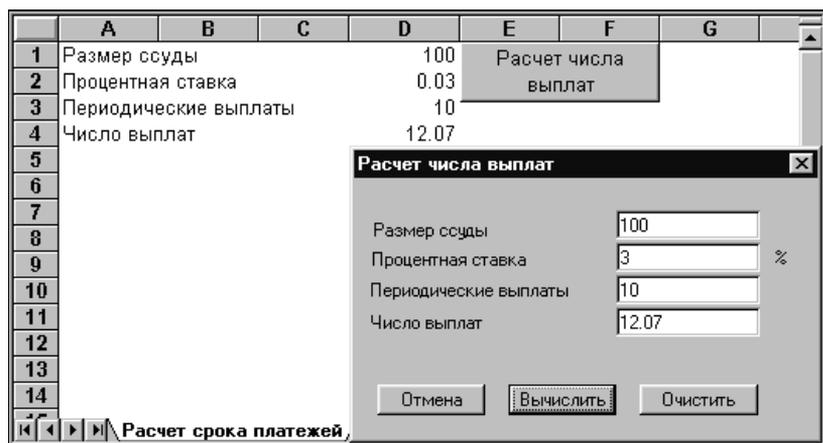


Рис. 7.44. Расчет числа выплат

Все расчеты и заполнение ячеек рабочего листа будет осуществлять следующая программа, которую необходимо набрать на листе модуля:

```
Sub Start ()
    DialogSheets ("Выплаты").Show
```

```
DialogSheets("Выплаты").EditBoxes(4).Enabled = False
End Sub

Sub Calc()
Dim A, P, n, i As Double
With DialogSheets("Выплаты")
    P = CDBl(.EditBoxes(1).Text)
    i = CDBl(.EditBoxes(2).Text)/100
    A = CDBl(.EditBoxes(3).Text)
End With
If Application.IsNumber(Application.NPer(i, -A, P)) = _
    False Then
MsgBox "Ошибка в данных." & Chr(13) & _
"Долг растёт быстрее, чем производятся выплаты.", _
vbExclamation, "Расчет числа выплат"
Exit Sub
End If
n = Application.NPer(i, -A, P)
n = Format(n, "Fixed")
DialogSheets("Выплаты").EditBoxes(4).Text = CStr(n)
With Sheets("Расчет срока платежей")
    .Range("D1").Value = P
    .Range("D2").Value = i
    .Range("D3").Value = A
    .Range("D4").Value = n
End With
End Sub

Sub CLL()
With DialogSheets("Выплаты")
    .EditBoxes(1).Text = ""
    .EditBoxes(2).Text = ""
    .EditBoxes(3).Text = ""
    .EditBoxes(4).Text = ""
End With
Range("D1:D4").Clear
End Sub
```

Назначьте процедуры следующим кнопкам:

Start	Кнопке Расчет числа выплат рабочего листа Расчет срока платежей
Calc	Кнопке Вычислить диалогового окна Расчет числа выплат
CLL	Кнопке Очистить диалогового окна Расчет числа выплат

Приведем необходимые пояснения к программе.

Число выплат вычисляется функцией `NPer`. В процедуру `Calc` введена проверка корректности ввода данных, которая в случае неверного ввода выдает соответствующее сообщение (рис. 7.45).

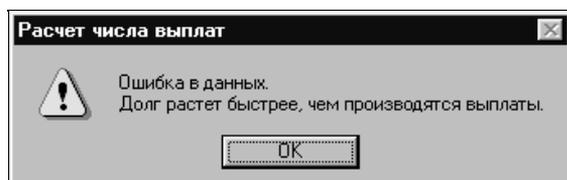


Рис. 7.45. Сообщение об ошибке в данных

Некорректность данных проверяется с помощью функции `IsNumber`. Функция `IsNumber` принимает значение `True`, если ее аргумент — число, и `False` — в противном случае. Функцией рабочего листа, соответствующей `IsNumber`, является `ЕЧИСЛО`.

Приведем другие функции рабочего листа, которые используются для проверки типа значения или ссылки. Каждая из этих функций проверяет тип значения и, в зависимости от него, возвращает логическое значение `ИСТИНА` или `ЛОЖЬ`. Например, функция `ЕПУСТО` возвращает логическое значение `ИСТИНА`, если проверяемое значение является ссылкой на пустую ячейку, в противном случае возвращается логическое значение `ЛОЖЬ`.

ЕЛОГИЧ	<code>ISLOGICAL</code>	Проверяемая величина является логическим значением
ЕНД	<code>ISNA</code>	Проверяемая величина является значением ошибки #Н/Д
ЕНТЕКСТ	<code>ISNOTEXT</code>	Проверяемая величина не является текстом
ЕОШ	<code>ISERR</code>	Проверяемая величина является одним из возможных ошибочных значений Excel за исключением #Н/Д
ЕОШИБКА	<code>ISERROR</code>	Проверяемая величина является одним из возможных ошибочных значений Excel
ЕПУСТО	<code>ISBLANK</code>	Проверяемая величина является ссылкой на пустую ячейку

ЕССЫЛКА	ISREF	Проверяемая величина является ссылкой
ЕТЕКСТ	ISTEXT	Проверяемая величина является текстом
ЕЧИСЛО	ISNUMBER	Проверяемая величина является числом

7.18. Расчет амортизации и ипотеки с использованием диалогового окна

7.18.1. Расчет амортизации

В данном разделе рассмотрим пример оформления расчета амортизации двумя методами с использованием диалогового окна. Выбор методов амортизации производится с помощью переключателя. В случае использования k -кратного учета амортизации дополнительно вводится значение k .

Создайте рабочую книгу, состоящую из трех листов: рабочего — Расчет амортизации, диалогового — Амортизация и листа модуля (рис. 7.46).

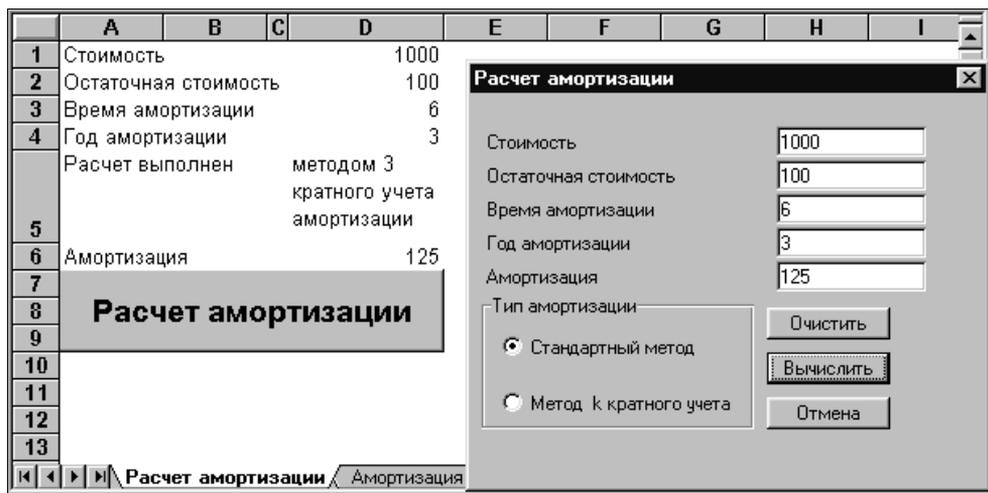


Рис. 7.46. Расчет амортизации

В ячейку D1 рабочего листа Расчет амортизации введем стоимость, в ячейку D2 — остаточную стоимость, в ячейку D3 — полное время амортизации, в ячейку D4 — год амортизации, в ячейку D5 — метод амортизации и в ячейку D6 — величину амортизации. На листе диалога Амортизация создайте диалоговое окно Расчет амортизации, как показано на рис. 7.46. Все расче-

ты и заполнение ячеек рабочего листа будет осуществлять следующая программа, набираемая на листе модуля

```
Sub Start()  
DialogSheets("Амортизация").Show  
DialogSheets("Амортизация").OptionButtons(1).Value = xlOn  
DialogSheets("Амортизация").EditBoxes(5).Enabled = False  
DialogSheets("Амортизация").EditBoxes(6).Visible = False  
DialogSheets("Амортизация").Labels(6).Visible = False  
DialogSheets("Амортизация").Spinners(1).Visible = False  
End Sub  
  
Sub Calc()  
Dim B, E, a As Double  
Dim Ye, Yc, k As Integer  
Dim Flag As Boolean  
    With DialogSheets("Амортизация")  
        B = CDBl(.EditBoxes(1).Text)  
        E = CDBl(.EditBoxes(2).Text)  
        Ye = CInt(.EditBoxes(3).Text)  
        Yc = CInt(.EditBoxes(4).Text)  
        .Spinners(1).Min = 2  
    End With  
    If B < E Then  
        MsgBox "Остаток больше начальной стоимости", _  
            vbExclamation, "Амортизация"  
        Exit Sub  
    End If  
    If Ye < Yc Then  
        MsgBox "Ошибка в сроке амортизации", _  
            vbExclamation, "Амортизация"  
        Exit Sub  
    End If  
    If DialogSheets("Амортизация").OptionButtons(1).Value _  
        =xlOn Then  
        Flag=True  
    Else  
        Flag=False  
    End If
```

```
If Flag=True Then
a=Application.Syd(B, E, Ye, Yc)
Else
k=CInt(DialogSheets("Амортизация").EditBoxes(6).Text)
a=Application.Ddb(B, E, Ye, Yc, k)
End If
If a>= 0.01 Then a=Format(a, "Fixed") Else a=0
DialogSheets("Амортизация").EditBoxes(5).Text=CStr(a)
Range("D1").Value=B
Range("D2").Value=E
Range("D3").Value=Ye
Range("D4").Value=Yc
Range("D6").Value=a
If Flag=True Then
    Range("D5").Value="стандартным методом"
Else
    Range("D5").Value="методом " & CStr(k) & _
        " кратного учета амортизации"
End If
End Sub

Sub SWI()
    DialogSheets("Амортизация").Labels(6).Visible = True
    DialogSheets("Амортизация").EditBoxes(6).Visible = True
    DialogSheets("Амортизация").Spinners(1).Visible = True
End Sub

Sub SWI1()
    DialogSheets("Амортизация").Labels(6).Visible = False
    DialogSheets("Амортизация").Spinners(1).Visible = False
    DialogSheets("Амортизация").EditBoxes(6).Visible = False
End Sub

Sub Cle()
Dim i As Integer
    With DialogSheets("Амортизация")
        For i = 1 To 6
            .EditBoxes(i).Text = ""
        Next i
    End With
End Sub
```

```
End With
Range("D1:D6").Clear
End Sub

Sub DoSp()
    With ActiveDialog
        .EditBoxes(6).Text = CStr(.Spinners(1).Value)
    End With
End Sub

Sub BackSp()
    With ActiveDialog
        .Spinners(1).Value = CInt(.EditBoxes(6).Text)
    End With
End Sub
```

Процедуру `Start` назначьте кнопке `Расчет` амортизации рабочего листа `Расчет` амортизации. Процедуру `Calc` свяжите с кнопкой `Вычислить`, а процедуру `Cle` — с кнопкой `Очистить` окна диалога `Расчет` амортизации.

Процедуры `SWI` и `SWI1` свяжите с первой и второй кнопками переключателя, соответственно. Они обеспечивают появление в диалоговом окне и исчезновение из него поля ввода `Кратность` метода в зависимости от того, стоит ли переключатель в положении `Метод k` кратного учета.

Появлением и исчезновением в диалоговом окне элементов управления управляет свойство `Visible`. Это свойство возвращает `True`, если графический объект видим, и `False` — в противном случае. Процедуры `DoSp` и `BackSp` обеспечивают работу счетчика.

7.18.2. Расчет ипотеки

Вид диалоговых окон можно оживлять, внедряя в них через буфер обмена графические объекты и художественно оформленные тексты. На рис. 7.47 приведен пример оформления диалогового окна расчета ипотечной ссуды. Программа расчета ипотечной ссуды приведена ниже.

```
Sub Ипотека_диалог()
DialogSheets("Диалог_ипотеки").Show
DialogSheets("Диалог_ипотеки").EditBoxes(5).Enabled = False
DialogSheets("Диалог_ипотеки").EditBoxes(6).Enabled = False
DialogSheets("Диалог_ипотеки").EditBoxes(7).Enabled = False
```

	A	B	C	D	E
1	Расчет ипотечной ссуды				
2	Ипотечная ссуда				
3	И				
4	Цена			Вычислить	
5	Первый и	Цена	10000	Отмена	
6	Годовая	Первый взнос, %	20	<input checked="" type="radio"/> Ежемесячные <input type="radio"/> Ежегодные	
7	Размер с	Годовая процентная ставка	20		
8		Срок погашения ссуды (лет)	10		
9	Срок пог	Вычисляемые параметры			
10		Периодические выплаты	155		
11	Периоди	Общая сумма выплат	18 553		
12	Общая с	Общая сумма комиссионных	10553		
13	Общая с				

Рис. 7.47. Расчет ипотечной ссуды

```
End Sub
```

```
Sub Ипотека_расчет()
```

```
Dim A, P, i, iFirst, PSum, PPer As Double
```

```
Dim n As Integer
```

```
Dim Fl As Boolean
```

```
With DialogSheets("Диалог_ипотеки")
```

```
    P = Cdbl(.EditBoxes(1).Text)
```

```
    iFirst = Cdbl(.EditBoxes(2).Text)/100
```

```
    i = Cdbl(.EditBoxes(3).Text)/100
```

```
    n = CInt(.EditBoxes(4).Text)
```

```
    If .OptionButtons(1).Value = xlOn Then
```

```
        Fl = True
```

```
    Else
```

```
        Fl = False
```

```
    End If
```

```
End With
```

```
PS = P * (1 - iFirst)
```

```
If Fl = True Then
```

```
    A = Application.Pmt(i/12, n * 12, -PS)
```

```
    PSum = A * n * 12
```

```
Else
```

```
    A = Application.Pmt(i, n, -PS)
```

```

        PSum = A * n
    End If
    PPer = PSum - PS
    A = Format(A, "###.##")
    PSum = Format(PSum, "###.##")
    PPer = Format(PPer, "###.##")
    With DialogSheets("Диалог_ипотеки")
        .EditBoxes(5).Text = CStr(A)
        .EditBoxes(6).Text = CStr(PSum)
        .EditBoxes(7).Text = CStr(PPer)
    End With
End Sub

```

7.19. Расчет постоянных периодических выплат с построением диаграммы из диалогового окна

В данном разделе рассмотрим пример оформления расчета постоянных периодических выплат с построением диаграммы из диалогового окна.

Создайте рабочую книгу, состоящую из трех листов: рабочего — Гр_программа, диалогового — Период. выплаты и листа модуля (рис. 7.48).

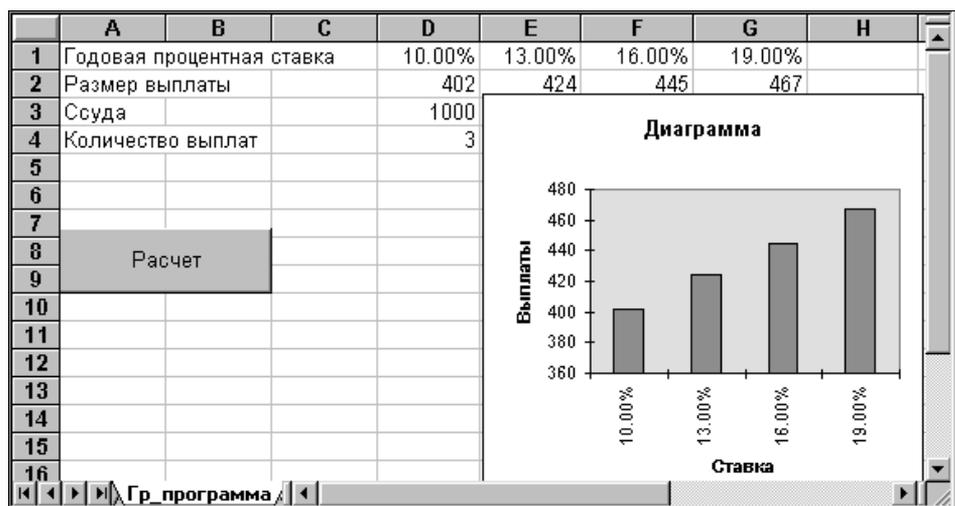


Рис. 7.48. Расчет постоянных периодических выплат

В ячейку D3 рабочего листа Гр_программа введем размер ссуды, в ячейку D4 — количество выплат. В диапазон ячеек D1:G1 введем процентные ставки, а в диапазон D2:G2— соответствующие постоянные периодические выплаты. На листе диалога Период. выплаты создайте диалоговое окно Расчет периодических выплат, как показано на рис. 7.49.

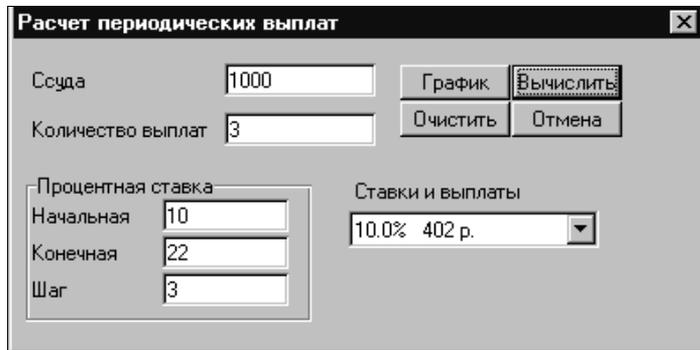


Рис. 7.49. Диалоговое окно расчета постоянных периодических выплат

Все расчеты и заполнение ячеек рабочего листа будет осуществлять следующая программа, которую необходимо набрать на листе модуля:

```
Sub Дисплей()
    DialogSheets("Период. выплаты").Show
End Sub

Sub Выплаты()
    Dim P, ib, ie, ist As Double
    Dim k, i, n, m As Integer
    Dim A(), B() As Double
    Dim S(), Bl() As String
    Dim Area As Object

    With DialogSheets("Период. выплаты")
        P = CDb1(.EditBoxes(1).Text)
        k = CInt(.EditBoxes(2).Text)
        ib = CDb1(.EditBoxes(3).Text)/100
        ie = CDb1(.EditBoxes(4).Text)/100
        ist = CDb1(.EditBoxes(5).Text)/100
    End With

    If ie < ib Or ie < ib + ist Then
        MsgBox "Ошибка в ставках", vbInformation
    End If
End Sub
```

```
Exit Sub

End If
Set Area = ActiveSheet.Cells(1, 4).CurrentRegion
n = Area.Columns.Count
ActiveSheet.Range(Cells(1, 4), Cells(2, n+3)).Clear
m = (ie - ib)/ist + 1
ReDim A(1 To m)
ReDim B(1 To m)
ReDim B1(1 To m)
With ActiveSheet
    For i = 1 To m
        B(i) = ib + ist*(i - 1)
        A(i) = Application.Pmt(B(i), k, -P)
        A(i) = Format(A(i), "##")
        B1(i) = CStr(Format(B(i), "00.0%"))
        .Cells(1, 4 + i - 1).Value = B1(i)
        .Cells(2, 4 + i - 1).Value = A(i)
    Next i
End With
With ActiveSheet
    .Range("D3").Value = P
    .Range("D4").Value = k
End With
ReDim S(1 To m)
For i = 1 To m
    S(i) = CStr(B1(i)) & " " & CStr(A(i)) & " p."
Next i
With DialogSheets("Период. выплаты")
    .DropDowns(1).RemoveAllItems
    For i = 1 To m
        .DropDowns(1).AddItem S(i), i
    Next i
    .DropDowns(1).Value = 1
End With
ActiveSheet.ChartObjects.Delete

End Sub

Sub График()
Dim Area As Object
```

```

Dim n, i As Integer
Set Area = ActiveSheet.Cells(1, 4).CurrentRegion
n = Area.Columns.Count
ActiveSheet.ChartObjects.Add(195, 30, 200, 190).Select
Application.CutCopyMode = False
ActiveChart.ChartWizard Source:= _
    Range(Cells(1, 4), Cells(2, n + 3)), _
    Gallery:=xlColumn, Format:=1, _
    PlotBy:=xlRows, CategoryLabels:=1, _
    SeriesLabels :=0, HasLegend:=2, _
    Title:="Диаграмма", _
    CategoryTitle:="Ставка", _
    ValueTitle:="Выплаты", _
    ExtraTitle:=""
End Sub

Sub Clea()
Dim Area As Object
Dim n, i As Integer
For i = 1 To ActiveSheet.ChartObjects.Count
    ActiveSheet.ChartObjects(i).Delete
Next i
Set Area = ActiveSheet.Cells(1, 4).CurrentRegion
n = Area.Columns.Count
With ActiveSheet
    .Range(Cells(1, 4), Cells(2, n + 3)).Clear
    .Range("D3").Clear
    .Range("D4").Clear
End With
End Sub

```

Процедуры данной программы выполняют следующие действия:

- | | |
|---------|--|
| Дисплей | Назначена кнопке Расчет рабочего листа Гр_программа. Активизирует диалоговое окно Расчет периодических выплат |
| Clea | Назначена кнопке Очистить диалогового окна. Очищает поля диалогового окна, ячейки рабочего листа и удаляет с рабочего листа все диаграммы |
| График | Назначена кнопке График диалогового окна. Строит график |

Выплаты

Назначена кнопке **Вычислить** диалогового окна **Расчет периодических выплат**. Проверяет правильность ввода процентных ставок. В случае некорректного ввода появляется сообщение, показанное на рис. 7.50. При правильном вводе начальной и конечной процентных ставок и шага их изменения процедура **Выплаты** очищает первую и вторую строку рабочего листа **Гр_программа** от результатов предыдущих вычислений, вычисляет размер периодических выплат, заполняет ячейки первой и второй строк значениями процентной ставки и соответствующими величинами выплат и выводит их в раскрываемом списке **Ставки и выплаты**

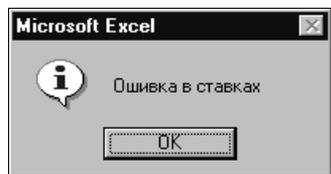


Рис. 7.50. Сообщение об ошибке ввода начальных данных

Приведем необходимые пояснения к программе.

Для заполнения раскрывающегося списка **Ставки и выплаты** в процедуре **Выплаты** использована следующая конструкция:

```
ReDim S(1 To m)
For i = 1 To m
    S(i) = CStr(B1(i)) & " " & CStr(A(i)) & " p."
Next i
With DialogSheets("Период. выплаты")
    .DropDowns(1).RemoveAllItems
    For i = 1 To m
        .DropDowns(1).AddItem S(i), i
    Next i
    .DropDowns(1).Value = 1
End With
```

ChartObjects возвращает набор всех диаграмм, а примененный к нему метод **Delete** их удаляет. Следовательно, конструкция:

```
ActiveSheet.ChartObjects.Delete
```

использованная в процедуре **Выплаты**, удаляет все ранее созданные на активном листе диаграммы.

Метод **Add**, примененный к **ChartObjects**, создает новый графический объект, определяя координату его левого верхнего угла, высоту и ширину. Еди-

нищей измерения является $1/72$ дюйма, а точка с координатами (0, 0) располагается в левом верхнем углу ячейки A1. В общем случае синтаксис метода Add имеет вид:

```
Add(left, top, width, height)
```

где left, top — координаты левого верхнего угла, width и height — ширина и высота создаваемого графического объекта. Например, использованная в процедуре График конструкция

```
ActiveSheet.ChartObjects.Add(195, 30, 200, 190).Select
```

создает на активном листе пустую диаграмму с левым верхним углом в точке с координатами (195, 30), шириной 200 и высотой 190.

Метод ChartWizard вводит данные и форматирует ранее созданную диаграмму. В процедуре График строится график с помощью следующей конструкции:

```
ActiveChart.ChartWizard Source:= _
Range(Cells(1, 4), Cells(2, n + 3)), _
Gallery:=xlColumn, Format:=1, _
PlotBy:=xlRows, CategoryLabels:=1, _
SeriesLabels :=0, HasLegend:=2, _
Title:="Диаграмма", _
CategoryTitle:="Ставка", _
ValueTitle:="Выплаты", _
ExtraTitle:=""
```

Опишем переменные метода ChartWizard:

Source	Определяет диапазон ячеек, по которому строится диаграмма
Gallery	Определяет тип диаграммы из второго диалогового окна мастера диаграмм. В табл. 7.11 приведены допустимые значения этой переменной
Format	Определяет вид диаграммы из третьего диалогового окна мастера диаграмм. Ей присваивается порядковый номер этого вида
PlotBy	Определяет, располагаются данные по строкам (xlRows) или по столбцам (xlColumns)
CategoryLabels	Определяет количество столбцов (строк), отводимых под метки по оси x

SeriesLabels	Определяет количество столбцов (строк), отводимых под легенду
HasLegend	Определяет, нужно отображать легенду (в этом случае переменной присваивается значение 1) или нет (присваивается 2)
Title	Определяет текст заголовка диаграммы
CategoryTitle	Определяет название оси <i>x</i>
ValueTitle	Определяет название оси <i>y</i>
ExtraTitle	Определяет название оси <i>z</i>

Таблица 7.11. Типы диаграмм

Тип диаграммы	Значение переменной <i>Gallery</i>
Гистограмма	xlColumn
График	xlLine
Круговая	xlPie
Кольцевая	xlDoughnut
Лепестковая	xlRadar
Точечная	xlXYScatter
Смешанная	xlCombination
С областями	xl3Darea
Линейчатая	xl3Dbar
Гистограмма	xl3Dcolumn
График	xl3Dline
Круговая	xl3Dpie
Поверхность	xl3Dsurface

Как видно из предыдущих примеров, при написании приложений на VBA довольно часто приходится использовать громоздкие конструкции. Для облегчения их составления рекомендуется использовать макрорекодер. С помощью макрорекодера все действия, производимые пользователем, записываются на языке VBA. Программа, создаваемая макрорекодером, называется макрос. Создавая макросы, можно ускорить написание программ и разобраться, как работает та или иная процедура. Однако надо помнить, что макрорекодер создает непропорционально длинные программы даже для

простых задач. Рассмотрим процесс работы с ним на примере создания макроса построения графика.

Запись макроса начинается с помощью команды **Сервис, Макрос, Начать запись** (Tools, Record Macro, Record New Macro). В поле **Имя макроса** (Macro Name) открывшегося диалогового окна **Запись макроса** (Record Macro) вводится имя макроса (рис. 7.51).

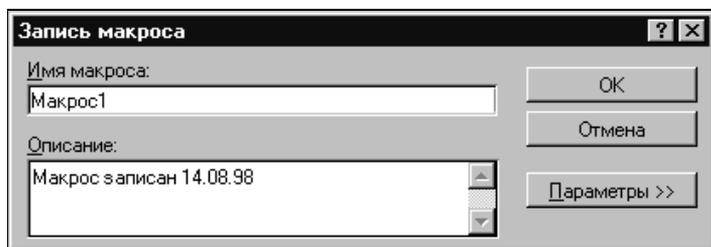


Рис. 7.51.
Диалоговое окно
Запись макроса

По умолчанию VBA присваивает макросам имена: Макрос1, Макрос2 (Macro1, Macro2) и т. д. Нажатие кнопки **Параметры** (Options) открывает расширенное диалоговое окно **Запись макроса** (Record Macro), позволяющее установить требуемые параметры макроса (рис. 7.52).

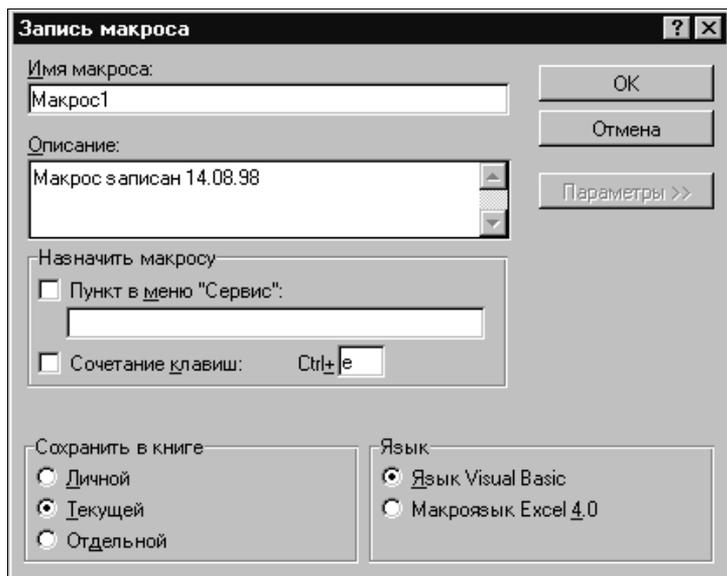


Рис. 7.52. Диалоговое окно параметров макроса

После нажатия кнопки **ОК** создается лист модуля, на котором будет записываться макрос. Кроме того, на экране появится окно с кнопкой (рис. 7.53), нажатие которой завершает запись макроса.

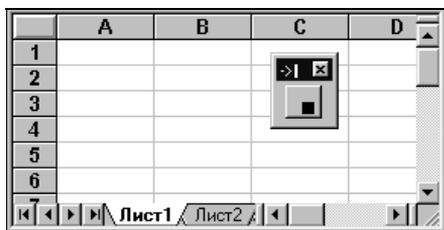


Рис. 7.53. Кнопка завершения записи макроса

Другим способом завершения записи является команда **Сервис, Запись макроса, Остановить запись** (Tools, Record Macro, Stop Recording).

В данном примере на листе Модуль1 будет создан следующий макрос:

```
'
' Макрос1 Макрос
' Макрос записан 14.08.98
'
'
Sub Макрос1()
Range("A1").Select
ActiveCell.FormulaR1C1 = "1"
Selection.AutoFill Destination:=Range("A1:D1"), Type:=xlFillSeries
Range("A1:D1").Select
Range("A2").Select
ActiveCell.FormulaR1C1 = "2"
Range("B2").Select
ActiveCell.FormulaR1C1 = "4"
Range("C2").Select
ActiveCell.FormulaR1C1 = "6"
Range("D2").Select
ActiveCell.FormulaR1C1 = "7"
Range("D3").Select
ActiveSheet.ChartObjects.Add(198.75, 54.75, 168, 117.75).Select
Application.CutCopyMode = False
ActiveChart.ChartWizard Source:= Range("A1:D2"), _
Gallery:=xlColumn, Format:=1, _
PlotBy:=xlRows, CategoryLabels:=1, _
SeriesLabels :=0, HasLegend:=2, _
Title:="Диаграмма", CategoryTitle:="Недели", _
ValueTitle:="Сбыт", _
ExtraTitle:=""
```

End Sub

Для выполнения макроса необходимо воспользоваться командой **Сервис, Макрос** (Tools, Macro). В открывшемся диалоговом окне **Макрос** (Macro) в поле **Имя макроса/Ссылка** (Macro Name/Reference) следует указать имя макроса и щелчком на кнопке **Выполнить** (Run) запустить его на выполнение (рис. 7.54).

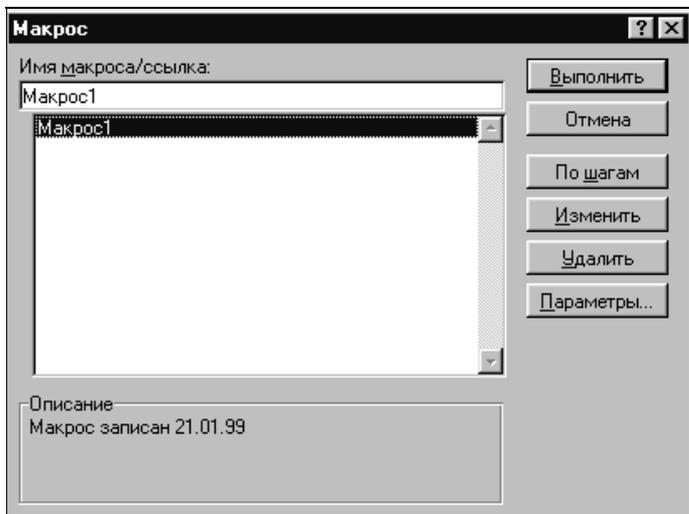


Рис. 7.54. Диалоговое окно **Макрос**

7.20. Программирование расчета годовой процентной ставки с помощью команды *Подбор параметра*

В данном разделе рассматривается задача нахождения годовой процентной ставки, при которой вы планируете накопить 1000 руб. в течение 10 лет. Предлагается 6 вариантов величины начального вклада в банк: 150, 200, 250, 300, 350 и 400 руб. Конечно, данную задачу можно решить с помощью функции ПЗ, выполнив команду **Подбор параметра** (Goal Seek) шесть раз, но это довольно утомительно.

Значительно проще составить программу. В ячейку C3 рабочего листа введите сумму накоплений, а в ячейку D3 — число лет. Введите в диапазон ячеек A3:A8 размеры первоначальных вкладов, а диапазон ячеек B3:B8 отведите под соответствующие годовые процентные ставки (рис. 7.55).

	A	B	C	D	E
1					
2	Ежегодная выплата	Годовая ставка	Сумма накоплений	Число лет	Старт
3	150	0.081441678	1000	10	150
4	200	0.150984145			200
5	250	0.214064943			250
6	300	0.273198463			300
7	350	0.329753139			350
8	400	0.384548196			400
9					

Рис. 7.55. Расчет годовой процентной ставки

На листе модуля введите следующую программу и назначьте процедуру Уравнение кнопке Старт рабочего листа Нел ур.

```
Sub Уравнение()
    Dim i, j As Integer
    Dim P, S As Double
    Dim T, Ti As String
    For i = 1 To 6
        j = i + 2
        With Sheets("Нел ур")
            S = Cells(i + 2, 1).Value
            Ti = "B" & LTrim(Str(j))
            T = "=C3*" & Ti & _
                "*" & (1 + Ti & ")^D3/((1 + Ti & ")^D3 - 1)"
            .Cells(i + 2, 5).Formula = T
            .Cells(i + 2, 5).GoalSeek _
                Goal := S, ChangingCell := .Cells(i + 2, 2)
        End With
    Next i
End Sub
```

Результаты расчета процедуры Уравнение представлены на рис. 7.55.

Приведем необходимые пояснения к данной процедуре.

Следующие свойства позволяют вводить в ячейку формулы:

Formula	Вводит формулу в ячейку, например
	Range("B1").Formula = "=SIN(A1)"

FormulaLocal	Эквивалентно свойству ячейки Formula и предназначено для ввода в ячейку неанглоязычных встроенных функций рабочего листа
FormulaArray	Вводит формулу в диапазон, например Range("E1:E3").FormulaArray = "=Sum(R1C1:R3C3)"
FormulaR1C1	Вводит формулу в ячейку в R1C1 формате, например Range("B1").FormulaR1C1 = "=SQRT(R1C1)"

Таким образом, конструкция:

```
j = i + 2
S = .Cells(i + 2, 1).Value
Ti = "B" & LTrim(Str(j))
T = "=C3*" & Ti & "*(1+" & Ti & ")^D3/((1+" & Ti & ")^D3-1)"
Sheets("Нел yp").Cells(i + 2, 5).Formula = T
```

при данном значении i вводит в ячейку Sheets("Нел yp").Cells(i+2, 5) формулу. Например, если $i = 2$, то в ячейку E3 вводится

```
=C3*B3*(1+B3)^D3/((1+B3)^D3-1)
```

Данная формула позволяет найти размер ежегодных выплат при годовой ставке, введенной в ячейку B3, сумме накоплений, введенной в ячейку C3 и числе выплат, введенном в ячейку D3.

В VBA выполнение команды **Подбор параметра** (Goal Seek) обеспечивается следующей конструкцией:

```
Уст_в_Яч.GoalSeek Goal:=Значение, ChangingCell:= Изм_зн_яч
```

Опишем содержащиеся в ней переменные:

Уст_в_Яч	Ссылка на ячейку, содержащую левую часть уравнения, корень которого определяется
Значение	Правая часть уравнения, не содержащая неизвестной величины
Изм_зн_яч	Ссылка на ячейку, содержащую неизвестную

7.21. Программирование решения систем нелинейных уравнений с параметром

В данном разделе программируется решение следующей системы нелинейных уравнений:

$$\begin{cases} x^2 + y^2 = 1, \\ 2x + 3y = a, \end{cases}$$

где $a \in \{0,1; 0,5; 0,9; 1,3; 1,7; 2,1; 2,5; 2,9; 3,3\}$.

Введите на листе модуля следующую программу:

```
Sub Система()
Dim F, XY, S As String
Dim i, n As Integer
Dim a As Double
    n=9
    For i=1 To n
        a=Sheets("Поиск_решения").Cells(5, i + 1).Value
        F="R8C" & LTrim(Str(i + 1))
        XY="R6C" & LTrim(Str(i + 1)) & _
            "R7C" & LTrim(Str(i + 1))
        S="(R6C" & LTrim(Str(i + 1)) & _
            "^2 + R7C" & LTrim(Str(i + 1)) & _
            "^2-1)^2 +" & _
            "(2*R6C" & LTrim(Str(i + 1)) & _
            " + 3*R7C" & LTrim(Str(i + 1)) & _
            "- " & Trim(Str(a)) & ")^2"
        Sheets("Поиск_решения").Cells(8, i + 1).Formula =S
        Application.ExecuteExcel4Macro String:= _
            "' [SOLVER.XLA] SOLVER' !ПОИСК. РЕШЕНИЯ. ПАРАМЕТРЫ" & _
            "(100,100,0.000001,FALSE,FALSE,1,1,1,0.005,FALSE)"
        Application.ExecuteExcel4Macro String:=_
            "' [SOLVER.XLA] SOLVER' !ПОИСК. РЕШЕНИЯ. ОК(!" & _
            & F & ",3,0,([qq.xls]Поиск_решения!" & XY & ")")
        Application.ExecuteExcel4Macro String:= _
            "' [SOLVER.XLA] SOLVER' !ПОИСК. РЕШЕНИЯ. ВЫПОЛНИТЬ(1)"
    Next i
End Sub
```

Программа Система выводит результат расчета на рабочем листе Поиск_решения (рис. 7.56).

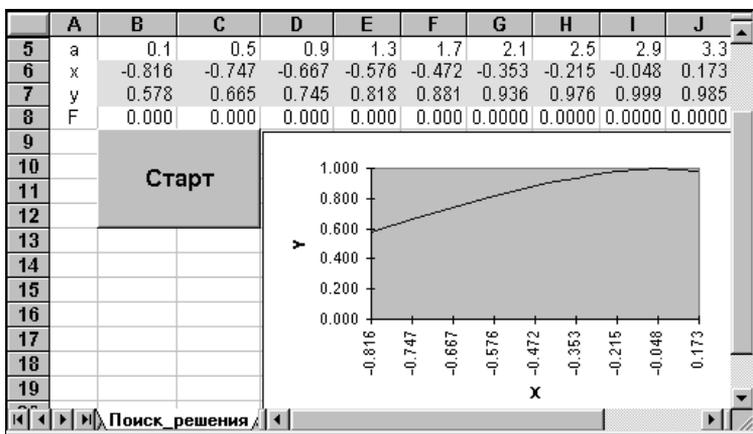


Рис. 7.56. Решение системы нелинейных уравнений с параметром

Программа Система выводит в диапазон B5:J5 значения параметра a , а в диапазоны B6:J6 и B7:J7 — соответствующие значения неизвестных x и y . В качестве начального приближения принято $(x, y) = (-1, 1)$. В диапазон B8:J8 программа вводит следующие целевые функции:

$$=(\$B\$6^2 + \$B\$7^2-1)^2 + (2*\$B\$6 + 3*\$B\$7-0.1)^2$$

$$=(\$C\$6^2 + \$C\$7^2-1)^2 + (2*\$C\$6 + 3*\$C\$7-0.5)^2$$

.....

$$=(\$J\$6^2 + \$J\$7^2-1)^2 + (2*\$J\$6 + 3*\$J\$7-3.3)^2$$

Команда

```
Application.ExecuteExcel4Macro String:= _
    "' [SOLVER.XLA] SOLVER' !ПОИСК.РЕШЕНИЯ.ПАРАМЕТРЫ" & _
    "(100, 100,0.000001,FALSE,FALSE,1,1,1,0.005,FALSE) "
```

задает следующие параметры средства поиска решения:

Максимальное время (Max Time)	100 с
Предельное число итераций (Iterations)	100
Относительная погрешность (Tolerance)	0,000001
Флажок Линейная модель (Assume Linear Model)	Снят

Флажок Показывать результаты итераций (Show Iteration Results)	Снят
Флажок Автоматическое масштабирование (Use Automatic Scaling)	Снят
Положение Линейная (Tangent) переключателя группы Оценка (Estimates)	Установлено
Положение Прямые переключателя группы Производные (Derivatives)	Установлено
Положение Ньютона (Newton) переключателя группы Метод (Search)	Установлено

Команда

```
Application.ExecuteExcel4Macro String:= _
    "' [SOLVER.XLA] SOLVER' !ПОИСК.РЕШЕНИЯ.ОК(!" _
    & F & ",3,0,([qq.xls]Поиск_решения!" & XY & ")")"
```

устанавливает опции диалогового окна **Поиск решения** (Solver). Ее параметрами являются:

F	Ссылка на ячейку поля Установить целевую ячейку (Set Target Cell)
XY	Диапазон ячеек поля Изменяя ячейки (By Changing Cells)
qq	Рабочая книга, в которой решается оптимизационная задача
3	Номер положения, в которое установлен переключатель группы Равной (Equal To). В данном случае переключатель стоит в положении Значению (Value of)
0	Число поля Значению (Value of)

Команда

```
String:="' [SOLVER.XLA] SOLVER' !ПОИСК.РЕШЕНИЯ.ВЫПОЛНИТЬ(1)"
```

запускает команду **Поиск решения** (Solver) на выполнение. Значение параметра, равное 1, указывает на то, что по завершении работы команды на экран не выводится диалоговое окно **Результаты поиска решения** (Solver Results) и в этом окне переключатель установлен в положение **Сохранить найденное решение** (Keep Solver Solution).

Если в диапазон B8:J8 независимо от программы введены вышеупомянутые целевые функции, то программа значительно упрощается:

```
Sub Система()
Dim F, XY As String
```

```

Dim i, n As Integer
Dim a As Double
    n=9
    For i=1 To n
        a=Sheets("Поиск_решения").Cells(5, i + 1).Value
        F="R8C" & LTrim(Str(i + 1))
        XY="R6C" & LTrim(Str(i + 1)) & ":R7C" & _
        LTrim(Str(i + 1))
        Application.ExecuteExcel4Macro String:= _
        "' [SOLVER.XLA] SOLVER'!ПОИСК.РЕШЕНИЯ.ПАРАМЕТРЫ(" _
        & "100,100,0.000001,FALSE,FALSE,1,1,1,0.005,FALSE)"
        Application.ExecuteExcel4Macro String:= _
        "' [SOLVER.XLA] SOLVER'!ПОИСК.РЕШЕНИЯ.ОК(!" _
        & F & ",3,0,([qq.xls]Поиск_решения!" & XY & "))"
        Application.ExecuteExcel4Macro String:= _
        "' [SOLVER.XLA] SOLVER'!ПОИСК.РЕШЕНИЯ.ВЫПОЛНИТЬ(1)"
    Next i
End Sub

```

7.22. Оптимальное планирование сбыта продукции

Рассмотрим следующую задачу. Фирма производит N видов продукции, аналогии которых выпускаются и другими фирмами. На основе анализа продаж можно оценить коэффициенты эффективности рекламы по каждому виду продукции (S_i), показывающие, какая доля рынка принадлежит фирме в зависимости от ее затрат на рекламу. Так, если фирма тратит x_i , а ее конкуренты — a_i на рекламу продукции i , то объем сбыта составит:

$$V_i = \frac{S_i x_i}{x_i + a_i}.$$

Примем издержки производства и сбыта (исключая затраты на рекламу) такими, что если фирма продает свою продукцию i -го вида в количестве V_i , то ее валовая прибыль составит $g_i V_i$, где g_i — прибыль с реализации единицы продукции. Таким образом, чистая прибыль фирмы равна:

$$P = \sum_{i=1}^N \frac{g_i S_i x_i}{x_i + a_i} - \sum_{i=1}^N x_i .$$

Предположим, что фирма решила потратить на рекламу сумму не более X . Таким образом, на переменные x_i накладываются следующие ограничения:

$$\sum_{i=1}^N x_i \leq X ,$$

$$x_i \geq 0 , i \in [1, N].$$

Необходимо найти оптимальное распределение средств на рекламу, максимизирующее валовую прибыль.

Для решения этой задачи создайте рабочую книгу с рабочим листом **Сбыт** (рис. 7.57) и двумя листами окон диалога **Диалог1** и **Диалог2**, на которых создайте диалоговые окна, показанные на рис. 7.58 и 7.59, соответственно.

	A	B	C	D	E	F	G
1		2000	30000	4000	2000	1500	
2		222	233	453	234	200	
3		1	2	1	2	2.2	
4		444.33	3505.98	893.11	733.47	612.40	
5							
6	Валовая прибыль	59579.41					
7	Реальные расходы на рекламу	6189		Количество типов продукции и максимальный расход на рекламу			
8	Максимальные расходы на рекламу	9890		Ввод S, a, g и расчет оптимальных затрат			
9	Число типов продукции	5					

Рис. 7.57. Планирование сбыта продукции

Сбыт [X]

Количество типов продукции

5 [↑] [↓]

Максимальные расходы на рекламу

9890 [↑] [↓]

[OK]

[Отмена]

Рис. 7.58. Диалоговое окно на листе **Диалог1**

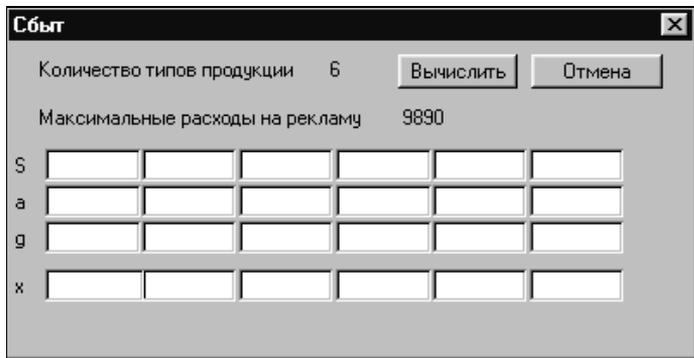


Рис. 7.59.
Диалоговое окно
на листе Диалог2

На листе модуля введите следующую программу:

```
Function P(s, g, x, a As Variant, n As Integer) As Double
Dim i As Integer
Dim t As Double
    t = 0
    For i = 1 To n
        t = t + g(i) * s(i) * x(i) / (x(i) + a(i))
    Next i
    For i = 1 To n
        t = t - x(i)
    Next i
    P = t
End Function

Sub Окно()
    Sheets("Диалог1").Show
End Sub

Sub Ввод()
Dim n, M As Integer
    n = CInt(Sheets("Диалог1").EditBoxes(1).Text)
    M = CInt(Sheets("Диалог1").EditBoxes(2).Text)
    Sheets("Сбыт").Cells(9, 2).Value = n
    Sheets("Сбыт").Cells(8, 2).Value = M
End Sub

Sub DoSpinner1()
    With ActiveDialog
```

```
.Spinners(1).Min = 1
.Spinners(1).Max = 6
.EditBoxes(1).Text = CStr(.Spinners(1).Value)
End With
End Sub
Sub BackSpinner1()
    With ActiveDialog
        .Spinners(1).Value = CInt(.EditBoxes(1).Text)
    End With
End Sub
Sub BackSpinner2()
    With ActiveDialog
        .Spinners(2).Value = CInt(.EditBoxes(2).Text)
    End With
End Sub
Sub DoSpinner2()
    With ActiveDialog
        .EditBoxes(2).Text = CStr(.Spinners(2).Value)
    End With
End Sub
Sub Второй_ввод()
Dim n, M, i As Integer
    n = Sheets("Сбыт").Cells(9, 2).Value
    M = Sheets("Сбыт").Cells(8, 2).Value
    With Sheets("Диалог2")
        .Labels(2).Text = CStr(n)
        .Labels(6).Visible = False
        .Labels(8).Text = CStr(M)
    End With
    With Sheets("Диалог2")
        For i = 1 To n
            .EditBoxes(i).Visible = True
            .EditBoxes(i + 6).Visible = True
            .EditBoxes(i + 12).Visible = True
        Next i
        For i = n + 1 To 6
            .EditBoxes(i).Visible = False
```

```

        .EditBoxes(i + 6).Visible = False
        .EditBoxes(i + 12).Visible = False
    Next i
    For i = 1 To 6
        .EditBoxes(i + 18).Visible = False
    Next i
End With
Sheets("Диалог2").Show
End Sub

Sub Расчет()
Dim n, M As Integer
Dim F, TT, R2L, R2R, H, l As String
Dim s, g, x, a As Variant
Dim Ar As Object
Dim Q As Double

    Set Ar = Range("A1").CurrentRegion
    Num = Ar.Columns.Count
    Range(Cells(1, 2), Cells(4, Num)).Clear
    n = Sheets("Сбыт").Cells(9, 2).Value
    Range(Cells(4, 2), Cells(4, n + 1)).NumberFormat = "0.00"
    M = Sheets("Сбыт").Cells(8, 2).Value
    F = "R6C2"
    TT = "R4C2:R4C" & LTrim(CStr(n + 1))
    R2R = "=R8C2"
    R2L = "R7C2"
    With Sheets("Диалог2")
        For i = 1 To n
            Sheets("Сбыт").Cells(1, 1 + i).Value = _
                .EditBoxes(i).Text
            Sheets("Сбыт").Cells(2, 1 + i).Value = _
                .EditBoxes(i + 6).Text
            Sheets("Сбыт").Cells(3, 1 + i).Value = _
                .EditBoxes(i + 12).Text
        Next i
    End With

' в переменную l, в зависимости от величины n,
' вводится имя последнего столбца,
' содержащего значения векторов g, a и S

```

```
Select Case n
    Case Is = 1
        l = "B"
    Case Is = 2
        l = "C"
    Case Is = 3
        l = "D"
    Case Is = 4
        l = "E"
    Case Is = 5
        l = "F"
    Case Is = 6
        l = "G"
    Case Is = 7
        l = "H"
End Select
Sheets("Сбыт").Cells(6, 2).FormulaLocal = _
    "=P(B1:" & l & "1;B3:" & l & "3;B4:" & l & _
    "4;B2:" & l & "2;B9)"
Sheets("Сбыт").Cells(7, 2).FormulaLocal = _
    "=СУММ(B4:" & l & "4)"
Application.ExecuteExcel4Macro String:= _
    "' [SOLVER.XLA] SOLVER'!ПОИСК.РЕШЕНИЯ.ОК(!" & _
    & F & ",1,0,([pr5.xls]Сбыт!" & TT & "))"
Application.ExecuteExcel4Macro String:= _
    "' [SOLVER.XLA] SOLVER'!ПОИСК.РЕШЕНИЯ.ДОБАВИТЬ(!" & _
    TT & ",3,""=0"")"
Application.ExecuteExcel4Macro String:= _
    "' [SOLVER.XLA] SOLVER'!ПОИСК.РЕШЕНИЯ.ДОБАВИТЬ(!" & _
    & R2L & ",1,""" & R2R & """)"
Application.ExecuteExcel4Macro String:= _
    "' [SOLVER.XLA] SOLVER'!ПОИСК.РЕШЕНИЯ.ПАРАМЕТРЫ" & _
    "(100,100,0.000001,FALSE,FALSE,1,1,1,0.05,FALSE)"
Application.ExecuteExcel4Macro String:= _
    "' [SOLVER.XLA] SOLVER'!ПОИСК.РЕШЕНИЯ.ВЫПОЛНИТЬ(2)"
Sheets("Сбыт").Cells(4, 2 + n).Clear
With Sheets("Диалог2")
    .Labels(6).Visible = True
    For i = 1 To n
```

```

        .EditBoxes(i + 18).Visible = True
        Q = Sheets("Сбыт").Cells(4, 1 + i).Value
        Q = Format(Q, "Fixed")
        .EditBoxes(i + 18).Text = CStr(Q)
    Next i
End With
End Sub

```

Процедуры данной программы выполняют следующие функции:

Р	Вычисляет чистую прибыль фирмы
Окно	Назначена кнопке Количество типов продукции и максимальный расход на рекламу. Активизирует диалоговое окно на листе Диалог1
Ввод	Назначена кнопке ОК диалогового окна на листе Диалог1. Осуществляет ввод данных в ячейки рабочего листа Сбыт
DoSpinner1, BackSpinner1, DoSpinner2, BackSpinner2	Назначены счетчикам диалогового окна на листе Диалог1 для ввода с их помощью данных
Второй_ввод	Назначена кнопке Ввод S, a, g и расчет оптимальных затрат. Активизирует диалоговое окно на листе Диалог2
Расчет	Назначена кнопке Вычислить диалогового окна на листе Диалог2. Производит расчет оптимальных вложений средств и заполнение рабочего листа Сбыт

Приведем некоторые пояснения к программе.

В программе использовано свойство `Visible`, задающее видимое и невидимое состояния графического объекта. Управление этим свойством определяет вид диалогового окна. Например, окно диалога на листе Диалог2 (рис. 7.59) с пятью типами продукции при вводе данных имеет вид, показанный на рис. 7.60, а при выводе — вид, показанный на рис. 7.61.

Команда

```

Application.ExecuteExcel4Macro String:= _
    "' [SOLVER.XLA] SOLVER' !ПОИСК.РЕШЕНИЯ.ДОБАВИТЬ(!" _
    & R2L & ",1,'" & R2R & "''")"

```

задает ограничения в диалоговом окне **Поиск решения** (Solver).

Сбыт

Количество типов продукции 5

Максимальные расходы на рекламу 9890

S	2000	30000	4000	2000	1500
a	222	233	453	234	200
g	1	2	1	2	2.2

Рис. 7.60.
Диалоговое окно
на листе Диалог2
при вводе данных

Сбыт

Количество типов продукции 5

Максимальные расходы на рекламу 9890

S	2000	30000	4000	2000	1500
a	222	233	453	234	200
g	1	2	1	2	2.2
x	444.33	3505.98	893.11	733.47	612.4

Рис. 7.61.
Диалоговое окно
на листе Диалог2
при выводе данных

Строковой переменной $R2L$ присвоена ссылка на диапазон ячеек, вводимых в поле **Ссылка на ячейку** (Cell Reference), а переменной $R2R$ присвоена ссылка на диапазон ячеек, вводимых в поле **Ограничение** (Constraint) диалогового окна **Добавление ограничения** (Add Constraint). Тип ограничения задается цифрой, а именно:

- | | |
|---|-------|
| 1 | <= |
| 2 | = |
| 3 | => |
| 4 | Целое |

7.23. Программирование рекламной кампании

В данном разделе рассмотрим задачу программирования рекламной кампании, рассмотренную в разделе 4.10. Напомним, что в этой задаче фирма

еженедельно оценивает, как обстоят дела со сбытом одного из видов своей продукции и дает оценку: отличную ("о" — состояние 1), хорошую ("х" — состояние 2) или удовлетворительную ("у" — состояние 3). Необходимо принять решение о целесообразности рекламирования этой продукции с целью расширения ее сбыта. Приведенные на рис. 7.62 в диапазонах В3:D5 и В6:D8 матрицы P^1 и P^2 определяют переходные вероятности без рекламы и при ее наличии в течение любой недели.

	А	В	С	Д	Е	Ф	Г	Н	І
1	Варианты	Р (Вероятности)			R (Прибыль)			Вычислить	
2		о	х	у	о	х	у		
3	1	о	0.2	0.5	0.3	7000	6000	3000	
4		х	0	0.5	0.5	0	5000	1000	
5		у	0	0	1	0	0	-1000	
6	2	о	0.3	0.6	0.1	6000	5000	-1000	
7		х	0.1	0.6	0.3	7000	4000	0	
8		у	0.05	0.4	0.55	6000	3000	-2000	
9	i (Состояние)	F(3,i)	к	F(2,i)	к	F(1,i)	к		
10	1	5300	1	8190	2	10735.5	2		
11	2	3100	2	5610	2	7922.5	2		
12	3	400	2	2125	2	4222.25	2		

Рис. 7.62. Программирование рекламной кампании

Соответствующие доходы заданы матрицами R^1 и R^2 в диапазонах Е3:G5 и Е6:G8. Процедура Реклама находит оптимальную рекламную кампанию на последующие N недель. Результаты расчета приведены на рис. 7.62, где $F(n, i)$ — оптимальный ожидаемый доход за недели $n, n+1, \dots, N$ при начальном состоянии i . Если $k = 1$, то нет необходимости рекламировать продукцию, а если $k = 2$, то ее рекламировать надо.

```
Sub Реклама()
Dim FP(1 To 3) As Double ' m вектор
Dim F(1 To 3) As Double ' m вектор
Dim NU(1 To 3, 1 To 2) As Double ' m x s матрица
Dim i, j, k, N, t, tt, s, m As Integer
Dim p, r, g As Double
' s - количество матриц перехода
' m - количество состояний в матрице перехода
' N - количество временных периодов
s = 2: m = 3: N = 3
For t = 1 To s
For i = 1 To m
    NU(i, t) = 0
```

```
For j = 1 To m
    p = Cells(i + 2 + (t - 1)*m, j + 1).Value
    r = Cells(i + 2 + (t - 1)*m, j + m + 1).Value
    NU(i, t) = NU(i, t) + p*r
Next j
Next i
Next t
' в массив NU вводятся ожидаемые прибыли на первом шаге
For i = 1 To m
    F(i) = 0
Next i
For t = N To 1 Step -1
    For i = 1 To m
        FP(i) = F(i)
    Next i
    For i = 1 To m
        F(i) = NU(i, 1)
        For j = 1 To m
            p = Cells(i + 2, j + 1).Value
            F(i) = F(i) + p*FP(j)
        Next j
        k = 1
        For tt = 2 To s
            g = NU(i, tt)
            For j = 1 To m
                p = Cells(i + 2 + m*(tt - 1), j + 1).Value
                g = g + p * FP(j)
            Next j
            If g > F(i) Then
                F(i) = g
                k = tt
            End If
        Next tt
        Cells(i + 3 + m * s, 2 * N + 3 - 2 * t - 1).Value = F(i)
        Cells(i + 3 + m * s, 2 * N + 3 - 2 * t).Value = k
    Next i
Next t
End Sub
```

7.24. Упражнения

Задания на вычисление комиссионных, программирование сценариев и оптимальное управление запасами

(а) Составить таблицу начисления премии по итогам работы сети n магазинов с месяца A по месяц B по следующему правилу:

- Если продукции продано не меньше, чем на C руб., то комиссионные составляют $i\%$
- За первое место дополнительно начисляется $j_1\%$, за второе место — $j_2\%$ и т. д., за k -е место дополнительно начисляется $j_k\%$

Вариант	C	i	j_1	j_2	j_3	j_4	n	A	B
1	23 000	1	3	1,5			3	Май	Июль
2	44 000	2	4	2			4	Июнь	Август
3	65 000	3	5	2,5	1,25		5	Июль	Ноябрь
4	86 000	4	6	3	1,5	0,75	6	Август	Февраль
5	107 000	5	7	3,5	1,75	0,88	7	Май	Декабрь
6	128 000	1	3	1,5	0,75	0,38	8	Октябрь	Январь
7	149 000	2	4	2	1	0,50	3	Ноябрь	Май
8	170 000	3	5	2,5	1,25	0,63	4	Декабрь	Май
9	191 000	4	6	3	1,5		5	Январь	Май
10	212 000	5	7	3,5			6	Февраль	Июнь

(б) Составить таблицу начисления премии по итогам работы сети n магазинов с месяца A по месяц B по следующему правилу: если продукции продано меньше, чем на C_1 руб., то премия составляет $i_1\%$ от стоимости реализованной продукции; если продукции продано не меньше, чем на C_1 руб., но меньше, чем на C_2 руб., то премия составляет $i_2\%$, и т. д.; если продукции продано не меньше, чем на C_k руб., то премия составляет $i_{k+1}\%$. N , A и B берутся из задания (а).

Вариант	C_1	C_2	C_3	i_1	i_2	i_3	i_4
1	10 000	20 000		1	1,5	2,3	
2	20 000	40 000		2	3,0	4,5	
3	30 000	60 000	120 000	3	4,5	6,8	7,4
4	40 000	80 000	160 000	4	6,0	9,0	9,9
5	50 000	100 000	200 000	5	7,5	11,3	12,4
6	60 000	120 000	240 000	6	9,0	13,5	14,9
7	70 000	140 000	280 000	1	1,5	2,3	2,5
8	80 000	160 000	320 000	2	3,0	4,5	5,0
9	90 000	180 000	360 000	3	4,5	6,8	7,4
10	100 000	200 000		4	6,0	9,0	

(с) Составить ведомость расчета прибыли от товара, производимого из n видов комплектующих. В этой ведомости учесть k вариантов стоимости комплектующих и стоимость работы по комплектации. В качестве стоимостей взять любые двузначные числа.

Вариант	1	2	3	4	5	6	7	8	9	10
n	2	4	5	6	3	4	5	6	3	4
k	3	4	5	6	7	8	9	10	11	12

(d) Рассмотрим следующую простую модель управления запасами. Уличный продавец покупает журналы у издательства по цене A руб. за штуку, а продает по цене B руб./шт. Если товар не реализован, то продавец возвращает его издательству по цене C руб. за штуку. Считаем, что продавец реализует журнал пачками по m штук. Продавец заметил, что за отчетный период он не реализовал ни одной пачки P_0 раз, одну пачку — P_1 раз и т. д., k пачек — P_k раз и при этом более P_k пачек он никогда не продавал. Определить оптимальный объем закупки журналов, максимизирующий ожидаемую прибыль.

Вариант	A	B	C	m	P_0	P_1	P_2	P_3	P_4	P_5
1	18	20	10	3	3	10	17	21	15	3
2	17	22	9	4	2	7	14	10	4	
3	19	24	8	5	3	8	16	14	9	4

(продолжение)

Вариант	A	B	C	m	P ₀	P ₁	P ₂	P ₃	P ₄	P ₅
4	21	26	10	6	4	6	15	12	7	
5	20	28	9	7	5	11	15	18	11	3
6	19	30	8	3	6	9	19	15	7	
7	16	20	11	4	1	5	16	21	17	4
8	15	22	10	5	2	4	12	17	11	5
9	14	24	9	6	3	9	14	21	14	6
10	15	26	7	7	4	6	17	14	7	

Задания по работе с массивами

Пусть C — строка, состоящая из n_C элементов, и G_C — квадратная матрица $n_C \times n_C$.

Вариант 1

(а) Построить

$$G_C(i, j) = \begin{cases} C^2(i), & i \leq j, \\ \sin(C(i)C(i-j)) + C^2(i), & i > j, \end{cases}$$

где $n_C = 3$ и $C = (3, 4, 1)$.

(б) Вычислить

$$s = \frac{2 \sum_{i=1}^n x_i y_i + \left(\sum_{i=1}^m \sum_{j=1}^m b_{ij} \right)^2}{3 + \sum_{i=1}^n x_i},$$

где x, y — векторы с n компонентами, b — матрица размерности $m \times m$, причем $n = 4, m = 2$,

$$x = (3, 1, 2, 3), \quad y = (1, 7, 2, 3), \quad b = \begin{pmatrix} 4 & 1 \\ 2 & 5 \end{pmatrix}.$$

Вариант 2

(a) Построить

$$G_c(i, j) = \begin{cases} \sin^2(C(i)), & i \leq j, \\ C(i - j) + \cos(C(i)), & i > j, \end{cases}$$

где $n_c = 4$ и $C = (2, 4, 1, 7)$.

(b) Вычислить

$$s = \frac{2 \sum_{i=1}^m a_i + \left(\sum_{i=1}^n \sum_{j=1}^n c_{ij} \right)^2}{\left(1 + \sum_{i=1}^m a_i \right) \left(1 + \sum_{i=1}^m a_i^2 \right)},$$

где a — вектор из m компонентов, c — матрица размерности $n \times n$, причем $n = 3$, $m = 4$,

$$a = (3, 1, 2, 3), c = \begin{pmatrix} 2 & 2 & 4 \\ 2 & 4 & 6 \\ 2 & 5 & 3 \end{pmatrix}.$$

Вариант 3

(a) Построить

$$G_c(i, j) = \begin{cases} C(i) + i + j^2, & i - j \leq 3, \\ C(i - j) + C(i)^3, & i - j > 3, \end{cases}$$

где $n_c = 5$ и $C = (6, 4, 2, 7, 6)$.

(b) Вычислить

$$s = \frac{\sum_{i=1}^n x_i + 2 \sum_{i=1}^n y_i^2 + 5 \left(\sum_{i=1}^m \sum_{j=1}^m b_{ij} \right)^3}{3 + \sum_{i=1}^n y_i},$$

где x , y — векторы из n компонентов, b — матрица размерности $m \times m$, причем $n = 4$, $m = 2$,

$$x = (1, 2, 7, 4), y = (1, 7, 2, 3), b = \begin{pmatrix} 4 & 1 \\ 2 & 5 \end{pmatrix}.$$

Вариант 4

(a) Построить

$$G_c(i, j) = \begin{cases} C(i)C(j), & i \leq j + 2, \\ C(i - j)^2 C(i), & i > j + 2, \end{cases}$$

где $n_c = 4$ и $C = (1, 2, 1, 7)$.

(b) Вычислить

$$s = 3 \sum_{i=1}^m a_i^2 + 7 \sum_{i=1}^n \sum_{j=1}^n c_{ij} - \left(1 + \sum_{i=1}^n \sum_{j=1}^n c_{ij} \right)^2,$$

где a — вектор из m компонентов, c — матрица размерности $n \times n$, причем $n = 3$, $m = 4$,

$$a = (3, 1, 2, 3), c = \begin{pmatrix} 2 & 2 & 4 \\ 2 & 4 & 6 \\ 2 & 5 & 3 \end{pmatrix}.$$

Вариант 5

(a) Построить

$$G_c(i, j) = \begin{cases} \sin(C(i))C(j), & i \leq j + 2, \\ |C(i - j) - C(i)|, & i > j + 2, \end{cases}$$

где $n_c = 6$ и $C = (1, 3, 1, 7, 2, 2)$.

(b) Вычислить

$$s = \sum_{i=1}^n x_i + 2 \sum_{i=1}^n y_i^2 + \left(\sum_{i=1}^m \sum_{j=1}^m b_{ij} \right) \left(2 + \sum_{i=1}^n x_i \right) - 2 \left(1 + \sum_{i=1}^n x_i y_i \right),$$

где x , y — векторы из n компонентов, b — матрица размерности $m \times m$, причем $n = 4$, $m = 2$,

$$x = (1, 2, 7, 4), y = (1, 7, 2, 3), b = \begin{pmatrix} 4 & 1 \\ 2 & 5 \end{pmatrix}.$$

Вариант 6

(a) Построить

$$G_c(i, j) = \begin{cases} 3C(i) - 2C(j), & i \leq j + 2, \\ |C(i - j) - C(i)^3|, & i > j + 2, \end{cases}$$

где $n_c = 7$ и $C = (1, 4, 1, 3, 4, 5, 7)$.

(b) Вычислить

$$s = \left(\sum_{i=1}^m a_i \right)^2 + \sum_{i=1}^n \sum_{j=1}^n c_{ij}^2 - \left(3 + \sum_{i=1}^n \sum_{j=1}^n c_{ij} \right) \left(1 + \sum_{i=1}^m a_i^2 \right),$$

где a — вектор из m компонентов, c — матрица размерности $n \times n$, причем $n = 3$, $m = 4$,

$$a = (3, 3, 1, 3), c = \begin{pmatrix} 2 & 2 & 4 \\ 2 & 4 & 6 \\ 2 & 5 & 3 \end{pmatrix}.$$

Вариант 7

(a) Построить

$$G_c(i, j) = \begin{cases} C(i) \cos^2(C(j)), & i \leq j + 1, \\ |C(i - j)^3 - C(i)|, & i > j + 1, \end{cases}$$

где $n_c = 4$ и $C = (1, 4, 8, 7)$.

(b) Вычислить

$$s = \left(2 \sum_{i=1}^n x_i + \sum_{i=1}^n x_i y_i \right) \left(2 - \sum_{i=1}^n x_i \right) + 3 + \sum_{i=1}^n x_i^2,$$

где x, y — векторы из n компонентов, причем $n = 4$,

$$x = (1, 2, 7, 4), y = (1, 7, 2, 3).$$

Вариант 8

(a) Построить

$$G_c(i, j) = \begin{cases} \sin(C(i) + C(j)), & i \leq j + 1, \\ C(i - j) + 3C(i) - 7C(j), & i > j + 1, \end{cases}$$

где $n_c = 5$ и $C = (2, 9, 1, 7, 3)$.

(b) Вычислить

$$s = \left(1 + \sum_{i=1}^m a_i\right)^2 \left(1 + \sum_{i=1}^m a_i^2\right) - \left(1 + \sum_{i=1}^n \sum_{j=1}^n c_{ij}\right) \left(1 + \sum_{i=1}^n \sum_{j=1}^n c_{ij}^2\right),$$

где a — вектор из m компонентов, c — матрица размерности $n \times n$, причем $n = 2, m = 4$,

$$a = (1, 4, 1, 3), c = \begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix}.$$

Вариант 9

(a) Построить

$$G_c(i, j) = \begin{cases} 3C(i) + 2C(j), & i \leq j + 2, \\ \sin(C(i - j)) + |3C(i) - 7C(j)|, & i > j + 2, \end{cases}$$

где $n_c = 6$ и $C = (1, 2, 1, 7, 8, 2)$.

(b) Вычислить

$$s = \left(\sum_{i=1}^n x_i^2 + 5 \sum_{i=1}^n x_i y_i\right) \left(1 + \sum_{i=1}^n x_i + \sum_{i=1}^n y_i\right) - 3 + \sum_{i=1}^n x_i^2,$$

где x, y — векторы из n компонентов, причем $n = 4$,

$$x = (7, 5, 7, 4), y = (2, 4, 2, 3).$$

Вариант 10

(a) Построить

$$G_c(i, j) = \begin{cases} |C(i) - 5C(j)|, & i \leq j+1, \\ C(i-j) + 4 \sin(C(i)) - 7C(j), & i > j+1, \end{cases}$$

где $n_c = 7$ и $C = (5, 1, 1, 7, 1, 2, 1)$.

(b) Вычислить

$$s = \left(1 + \sum_{i=1}^m a_i\right)^2 \left(1 + \sum_{i=1}^n \sum_{j=1}^n c_{ij}^2\right) - 1 - \sum_{i=1}^m a_i^2 + 4 \sum_{i=1}^n \sum_{j=1}^n c_{ij},$$

где a — вектор из m компонентов, c — матрица размерности $n \times n$, причем $n = 3$, $m = 4$,

$$a = (2, 1, 1, 3), \quad c = \begin{pmatrix} 5 & 2 & 4 \\ 1 & 4 & 2 \\ 7 & 3 & 5 \end{pmatrix}.$$

Задания на работу с базами данных

Построить диалог построения базы данных (рис. 7.63), имеющий n полей ввода, m раскрывающихся списков, k групп по два флажка, p групп по два переключателя и q полей ввода со счетчиками, используя следующую таблицу:

Вариант	n	m	k	p	q
1	2	4	2	2	2
2	3	3	3	3	2
3	4	2	4	2	3
4	2	2	4	4	3
5	3	3	3	2	4
6	4	4	2	4	4
7	2	3	2	2	3
8	3	2	2	3	2
9	4	4	3	4	3
10	4	3	3	4	2

Рис. 7.63. Пример окна диалога построения базы данных

Задания на нахождение оптимального раскроя

Продукция бумажной фирмы выпускается в виде бумажных рулонов стандартной длины, равной L футам. По специальным заказам фирма поставяет потребителям рулоны и других размеров, для чего производится раскрой стандартных рулонов. Необходимо удовлетворить заказ, минимизируя отходы.

Вариант	L	Дл 1	Шт 1	Дл 2	Шт 2	Дл 3	Шт 3	Дл 4	Шт 4
1	22	3	100	5	150	6	320	7	320
2	23	3	120	4	160	5	325	6	330
3	24	4	140	5	170	6	330	8	340
4	25	4	160	6	180	7	335	8	350
5	26	5	180	6	190	8	340	9	360
6	27	7	200	8	200	9	345	10	370
7	28	4	220	6	210	9	350	11	380
8	29	5	240	7	220	9	355	10	390
9	30	4	260	11	230	12	360	13	400
10	18	3	280	4	240	5	365	6	410

Задания на определение оптимальных капиталовложений

Фирма по производству быстрого питания намерена вложить капитал в размере A млн. руб. в расширение производства. Фирма имеет n филиалов, расположенных в различных городах. В каждом из этих филиалов проведено изучение рынка и найдены математические ожидания прибыли как функции капиталовложений. Необходимо выработать оптимальный план капиталовложений, максимизирующий ожидаемую прибыль.

Вариант 1

Филиалы	Млн. руб.						
	0	1	2	3	4	5	6
I	0	0,31	0,45	0,65	0,69	0,90	1,02
II	0	0,22	0,41	0,55	0,65	0,75	0,80
III	0	0,18	0,24	0,40	0,50	0,62	0,73
IV	0	0,20	0,39	0,42	0,48	0,53	0,56
V	0	0,17	0,33	0,42	0,48	0,53	0,56

Вариант 2

Филиалы	Млн. руб.						
	0	1	2	3	4	5	6
I	0	0,31	0,45	0,65	0,69	0,90	1,02
II	0	0,22	0,41	0,55	0,65	0,75	0,80
III	0	0,18	0,24	0,40	0,50	0,62	0,73
IV	0	0,20	0,39	0,42	0,48	0,53	0,56
V	0	0,17	0,33	0,42	0,48	0,53	0,56

Вариант 3

Филиалы	Млн. руб.						
	0	1	2	3	4	5	6
I	0	0,13	0,13	0,14	0,14	0,15	0,15
II	0	0,14	0,15	0,15	0,15	0,15	0,15
III	0	0,12	0,12	0,13	0,13	0,13	0,14
IV	0	0,15	0,17	0,19	0,21	0,23	0,25
V	0	0,12	0,12	0,13	0,13	0,14	0,14
VI	0	0,13	0,14	0,14	0,15	0,15	0,16

Вариант 4

Филиалы	Млн. руб.							
	0	1	2	3	4	5	6	7
I	0	0,13	0,13	0,14	0,14	0,15	0,15	0,16
II	0	0,11	0,12	0,12	0,12	0,12	0,12	0,12
III	0	0,13	0,13	0,14	0,14	0,14	0,15	0,15
IV	0	0,14	0,16	0,18	0,20	0,22	0,24	0,27
V	0	0,13	0,13	0,14	0,14	0,15	0,15	0,16
VI	0	0,12	0,13	0,13	0,14	0,14	0,15	0,15

Вариант 5

Филиалы	Млн. руб.					
	0	1	2	3	4	5
I	0	0,15	0,15	0,16	0,16	0,17
II	0	0,16	0,17	0,17	0,17	0,17
III	0	0,15	0,15	0,16	0,16	0,16
IV	0	0,17	0,19	0,21	0,23	0,25
V	0	0,16	0,16	0,17	0,17	0,18

Вариант 6

Филиалы	Млн. руб.						
	0	1	2	3	4	5	6
I	0	0,20	0,21	0,21	0,22	0,22	0,23
II	0	0,21	0,22	0,22	0,23	0,23	0,24
III	0	0,19	0,19	0,20	0,20	0,21	0,21
IV	0	0,18	0,20	0,22	0,24	0,26	0,29
V	0	0,21	0,21	0,22	0,22	0,23	0,23
VI	0	0,23	0,24	0,24	0,25	0,26	0,26

Вариант 7

Филиалы	Млн. руб.							
	0	1	2	3	4	5	6	7
I	0	0,11	0,11	0,12	0,12	0,13	0,13	0,14
II	0	0,12	0,13	0,13	0,13	0,13	0,13	0,13
III	0	0,18	0,18	0,19	0,19	0,20	0,20	0,20
IV	0	0,20	0,22	0,24	0,26	0,28	0,31	0,33
V	0	0,17	0,17	0,18	0,18	0,19	0,19	0,20
VI	0	0,17	0,18	0,18	0,19	0,19	0,20	0,21

Вариант 8

Филиалы	Млн. руб.						
	0	1	2	3	4	5	6
I	0	0,31	0,32	0,32	0,33	0,33	0,34
II	0	0,29	0,30	0,31	0,32	0,32	0,33
III	0	0,29	0,29	0,30	0,30	0,31	0,31
IV	0	0,28	0,30	0,32	0,34	0,37	0,39
V	0	0,32	0,33	0,33	0,34	0,34	0,35

Вариант 9

Филиалы	Млн. руб.						
	0	1	2	3	4	5	6
I	0	0,15	0,15	0,16	0,16	0,17	0,17
II	0	0,14	0,15	0,15	0,15	0,15	0,15
III	0	0,16	0,16	0,17	0,17	0,17	0,18
IV	0	0,12	0,14	0,16	0,18	0,20	0,22
V	0	0,16	0,16	0,17	0,17	0,18	0,18

Вариант 10

Филиалы	Млн. руб.					
	0	1	2	3	4	5
I	0	0,34	0,35	0,35	0,36	0,37
II	0	0,32	0,33	0,34	0,35	0,36
III	0	0,32	0,33	0,33	0,34	0,34
IV	0	0,29	0,31	0,33	0,35	0,38
V	0	0,33	0,34	0,34	0,35	0,35
VI	0	0,34	0,35	0,35	0,36	0,37

Приложение А



Отладка программ

При написании программ часто допускаются те или иные ошибки. На некоторые ошибки VBA сразу указывает, выдавая сообщение об ошибке синтаксиса (рис. А.1).

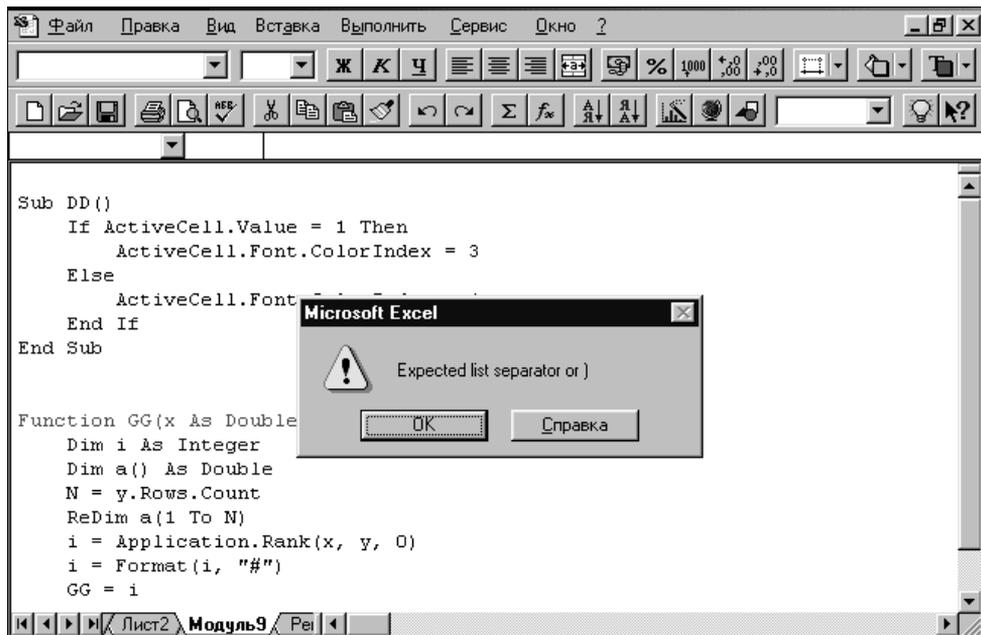


Рис. А.1. Пример сообщения об ошибке (о недостатке скобок в выражении)

Такие ошибки обычно легко исправить, т. к. их местоположение локализовано и указана их возможная причина.

Другой тип ошибок возникает при запуске программ (рис. А.2).

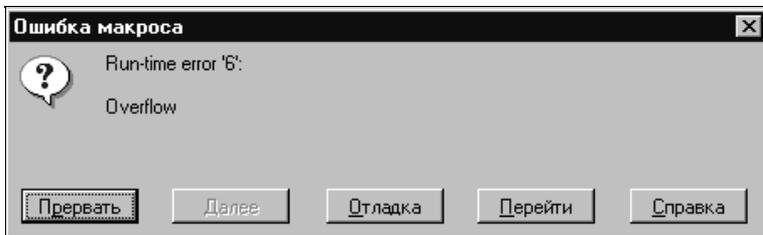


Рис. А.2. Диалоговое окно **Ошибка макроса**

При нажатии кнопки **Справка** (Help) VBA предложит возможные причины появления данной ошибки. Нажатие кнопки **Перейти** активизирует лист модуля в том месте, где допущена ошибка, с выделением ее синим цветом (рис. А.3).

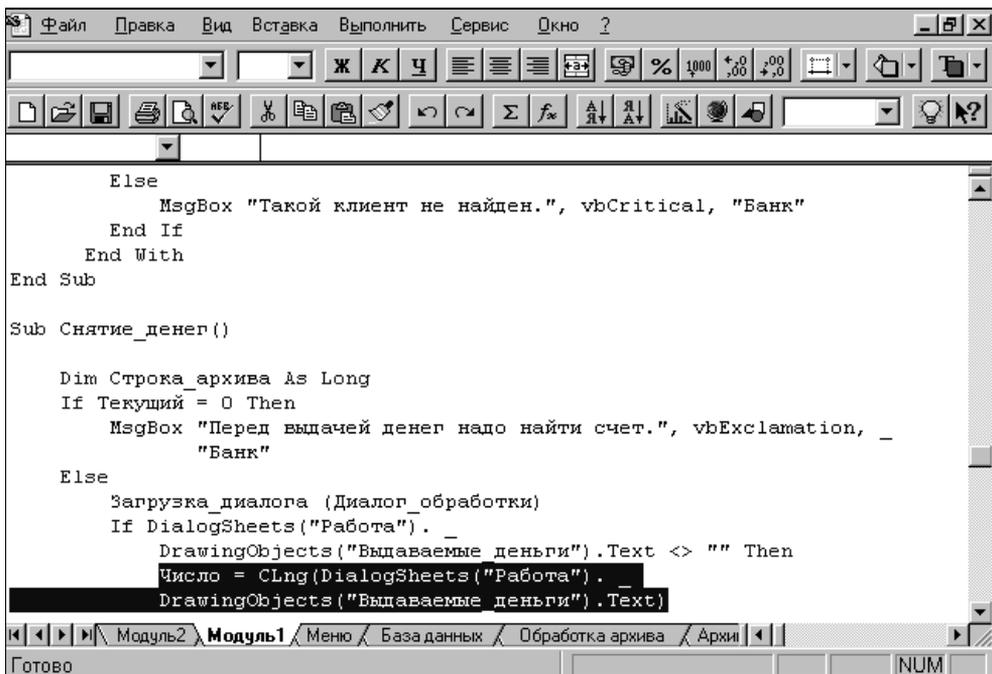


Рис. А.3. Лист модуля с выделенной ошибкой

В данном случае ошибка вызвана тем, что переменной `Число` типа `Integer` было присвоено слишком большое значение. Для того чтобы исправить эту ошибку, достаточно присвоить переменной `Число` тип `Long`.

Щелчок на кнопке **Отладка** (Debug) открывает окно **Отладка** (Debug), в котором место ошибки выделено рамкой (рис. А.4).

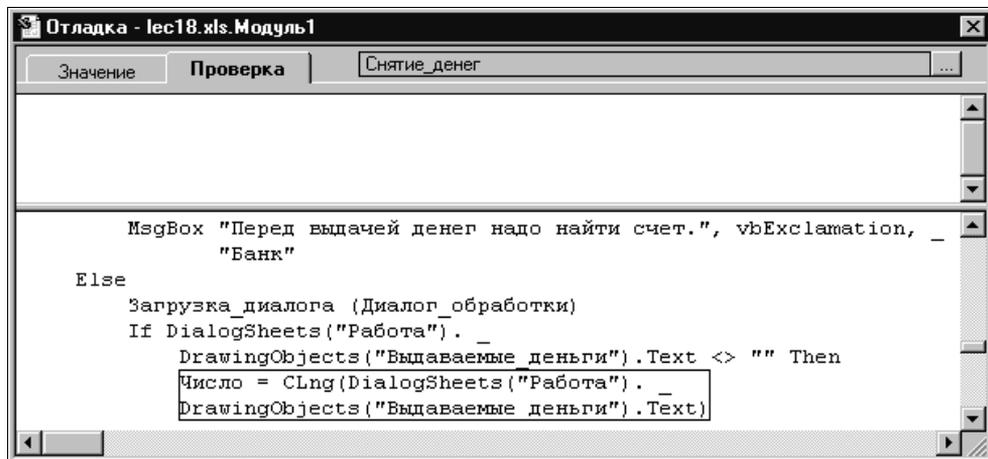


Рис. А.4. Окно **Отладка**

Наиболее коварными являются логические ошибки (когда программа работает, но выдает неверные результаты). В этом случае очень полезными являются средства отладки, которые позволяют лучше понять, что действительно происходит в программе, проследить за ее работой шаг за шагом. Окно **Отладка** (Debug) вызывается нажатием клавиш <Ctrl>+<G> или командой **Вид, Окно отладки** (View, Debug Window).

Окно **Отладка** (Debug) разбито на две панели: нижняя панель называется **Панель кода** (Code Pane) и выполняет роль листа модуля, а верхняя панель может отображать одну из двух вкладок: **Проверка** (Immediate) или **Значение** (Watch).

Вкладка **Значение** (Watch) позволяет отслеживать изменения значений выбранных переменных.

Вкладка **Проверка** (Immediate) предназначена для тестирования одного оператора, проверки значения переменной. Текст, введенный на этой вкладке, не изменяет текста программы, записанного на листе модуля. Вводимый оператор выполняется по нажатию клавиши <Enter>. Рассмотрим конкретные примеры работы с вкладкой **Проверка** (Immediate) (рис. А.5).

В окне **Отладка** (Debug) на вкладке **Проверка** (Immediate) введите следующий оператор присваивания:

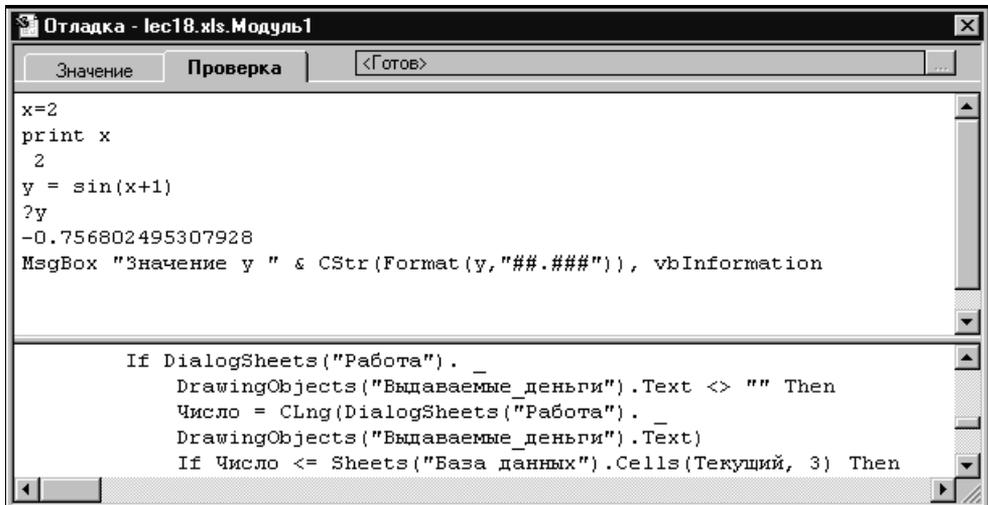


Рис. А.5. Работа с вкладкой **Проверка** окна **Отладка**

При нажатии клавиши <Enter> VBA выполнит этот оператор, т. е. переменной x будет присвоено значение 2. Для того чтобы проверить, какое значение присвоено переменной x , на новой строке введите оператор

```
print x
```

который при нажатии клавиши <Enter> выведет на вкладке **Проверка** (Immediate) значение этой переменной. Оператор `print` не позволяет вывести значение на листе модуля, однако, используя конструкцию

```
Debug.Print имя_переменной
```

можно вывести это значение на вкладке **Проверка** (Immediate).

Другим способом вывода значения выражения является использование вопросительного знака. Например, введите на вкладке **Проверка** (Immediate)

```
y = sin(x+1)
?y
```

VBA будет помнить значение переменной x до тех пор, пока не будет закрыто окно **Отладка** (Immediate). Поэтому переменной y будет присвоено значение $\sin(3)$. Для выполнения операторов после ввода каждой строки не забывайте нажимать клавишу <Enter>. После выполнения оператора `?y` на вкладке **Проверка** (Immediate) будет выведено значение переменной y .

На вкладке **Проверка** (Immediate) можно выводить значения выражений также посредством диалогового окна вывода, например

```
MsgBox "Значение y " & CStr(Format(y,"##.###")), vbInformation
```

Для проверки последовательности выполнения операторов в программе необходимо поместить курсор в процедуру, которую необходимо отладить, и осуществить одно из перечисленных ниже действий:

- Выбрать команду **Выполнить, Выполнить пошагово** (Run, Step Into)
- Нажать клавишу <F8>
- Нажать кнопку **Войти в процедуру** (Step Into)  (Все кнопки, упоминаемые в этом приложении, находятся на панели **Visual Basic**)

В результате откроется окно **Отладка** (Immediate), отображающее выбранную процедуру на **Панели кода** (Code Pane). Первая строка процедуры будет выделена рамкой. Это означает, что она будет выполнена следующей (рис. А.6).

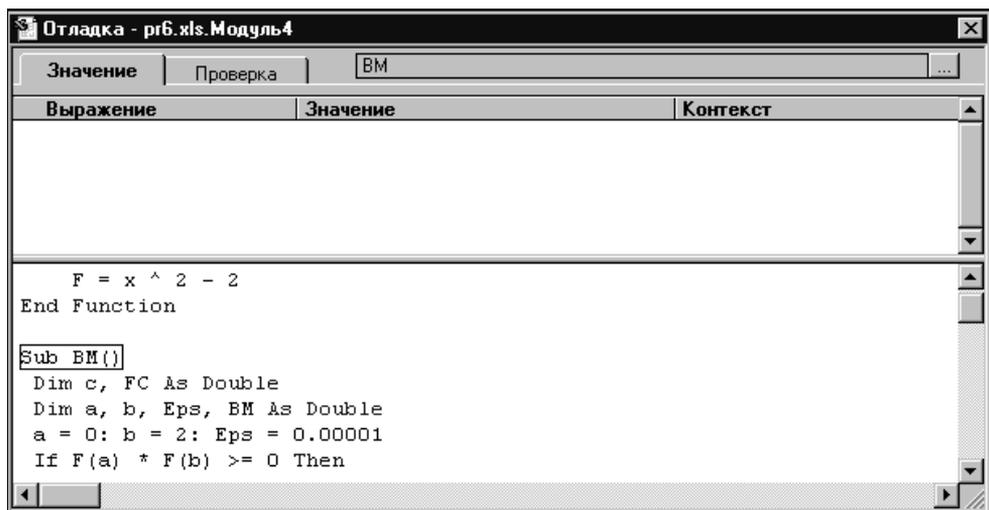


Рис. А.6. Пошаговая отладка программы

Для выполнения этой строки надо повторить одно из перечисленных выше действий. После выполнения последнего оператора процедуры окно **Отладка** (Immediate) автоматически закрывается.

Если при пошаговом выполнении программы нет необходимости отслеживать выполнение какой-либо вызываемой процедуры, ее пошаговое выполнение можно пропустить, выполнив одно из перечисленных ниже действий:

- Выбрать команду **Выполнить, Перешагнуть** (Run, Step Over)

- Нажать клавиши <Shift>+<F8>
- Нажать кнопку **Обойти процедуру** (Step Over) 

Программу можно выполнять по шагам с любого места, используя точку прерывания, которую устанавливают в программе, предварительно поместив курсор в нужной строке и выполнив одно из следующих действий:

- Выбрать команду **Выполнить, Точка останова** (Run, Toggle Breakpoint)
- Нажать клавишу <F9>
- Нажать кнопку **Точка останова** (Toggle Breakpoint) 

После установки точки прерывания цвет строки меняется на коричневый (рис. А.7).

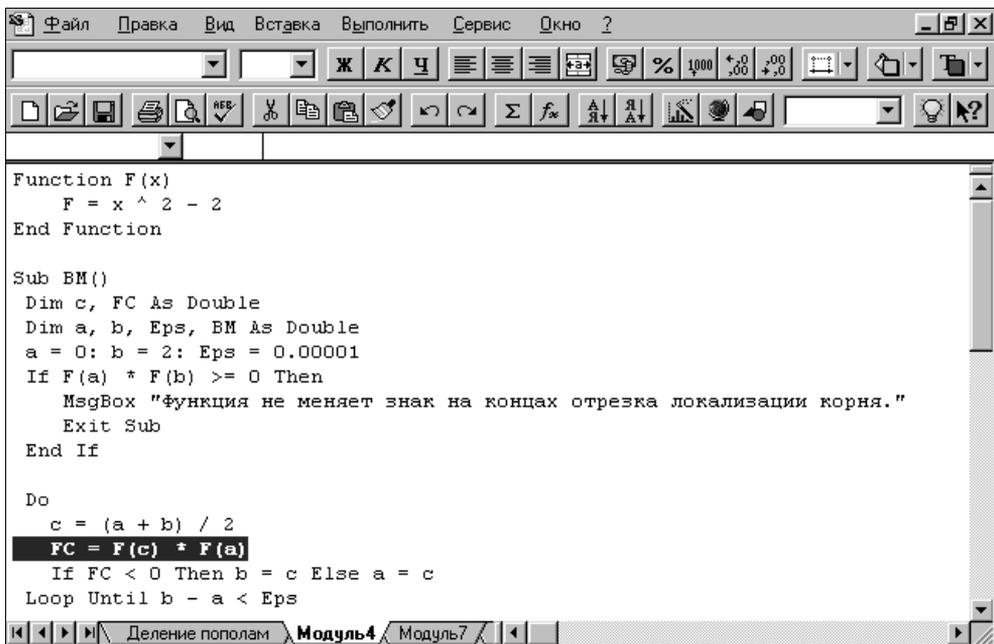


Рис. А.7. Точка прерывания

После указания точек прерывания запуск программы производится одной из следующих операций:

- Выбрать команду **Выполнить, Начать** (Run, Start)
- Нажать клавишу <F5>
- Нажать кнопку **Выполнить макрос** (Run Macro) 

Программа прервется на месте, где установлена точка прерывания. Для удаления точки прерывания надо поместить курсор в строке с точкой прерывания и повторить одно из описанных выше действий по ее установке.

Отметим, что можно размещать точки прерывания программно на листе модуля, используя операторы `Stop`.

При пошаговой отладке программы можно отслеживать изменения значений переменных с помощью вкладки **Значение** (Watch) окна **Отладка** (Immediate). Для этого надо выделить переменную в программе на **Панели кода** и выполнить команду **Сервис, Добавление контрольного значения** (Tools, Add Watch) (рис. А.8).

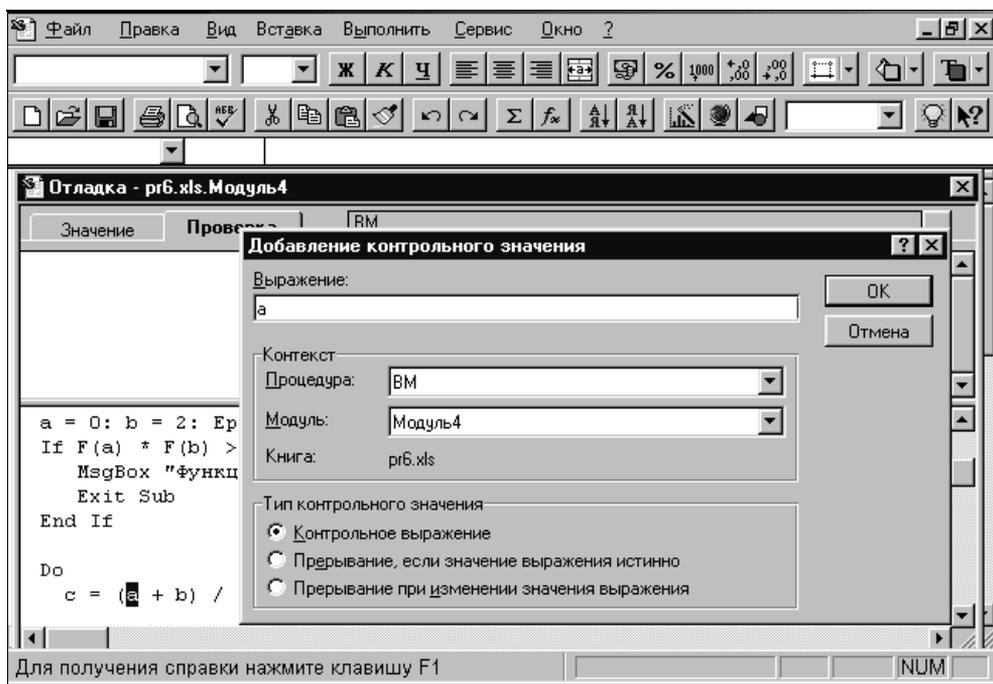


Рис. А.8. Диалоговое окно **Добавление контрольного значения**

После нажатия кнопки **ОК** данная переменная помещается на вкладку **Значение** (Watch). Для того чтобы добавить наблюдения за другой переменной, надо повторить с ней описанные выше действия или выбрать ее и щелкнуть кнопку **Контрольное значение** (Instant Watch) . Щелчок на кнопке **Добавить** (Add) открывшегося диалогового окна **Просмотр значения выражения** (Instant Watch) (рис. А.9) добавит новую переменную на вкладку **Значение** (Watch).

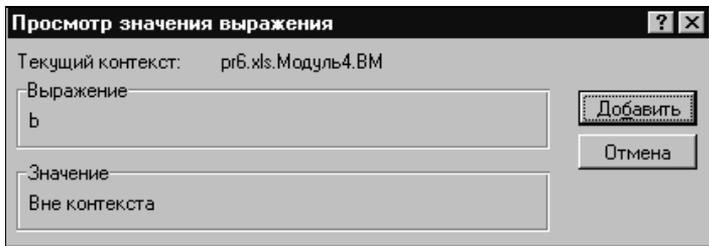


Рис. А.9. Диалоговое окно **Просмотр значения выражения**

После этого закройте окно **Отладка** (Immediate) и выполните в пошаговом режиме программу. Теперь текущие изменения выбранных переменных будут отображаться на вкладке **Значение** (Watch) (рис. А.10).

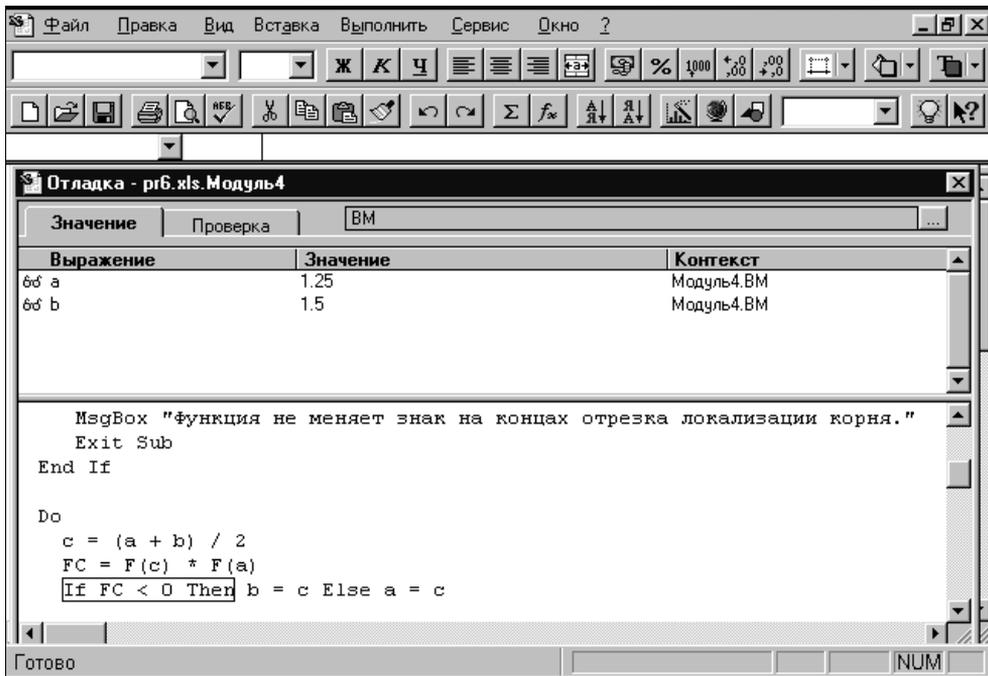


Рис. А.10. Отображение текущих значений выбранных переменных

Для удаления переменной с вкладки **Значение** (Watch) достаточно расположить курсор на строке с этой переменной и нажать клавишу <Delete>. С помощью команды **Сервис, Изменить контрольное значение** (Tools, Change Watch) можно изменить тестируемую переменную.

Приложение В



Функции рабочего листа Excel

В данном приложении содержится список основных функций рабочего листа Excel. Для получения более подробной информации о конкретной функции и ее аргументах выберите ее в диалоговом окне **Мастер функций** (Function Wizard) и нажмите кнопку **Справка** (Help).

*Таблица В.1. Функции категории **Математические** (Math&Trig)*

Функция (рус.)	Функция (англ.)	Назначение
ABS	ABS	Возвращает модуль (абсолютную величину) числа (см. раздел 2.1)
ACOS	ACOS	Возвращает арккосинус числа. Если нужно преобразовать результат из радиан в градусы, то умножьте его на $180/\text{ПИ}()$ (см. раздел 2.1)
ACOSH	ACOSH	Возвращает гиперболический арккосинус числа (см. раздел 2.1)
ASIN	ASIN	Возвращает арксинус числа (см. раздел 2.1)
ASINH	ASINH	Возвращает гиперболический арксинус числа (см. раздел 2.1)
ATAN	ATAN	Возвращает арктангенс числа (см. раздел 2.1)
ATAN2	ATAN2	Возвращает арктангенс для заданных координат X и Y (см. раздел 2.1)
ATANH	ATANH	Возвращает гиперболический арктангенс числа (см. раздел 2.1)

Таблица В.1 (продолжение)

Функция (рус.)	Функция (англ.)	Назначение
COS	COS	Возвращает косинус заданного угла (см. раздел 2.1)
COSH	COSH	Возвращает гиперболический косинус числа
EXP	EXP	Возвращает число "е", возведенное в указанную степень (см. раздел 2.1)
LN	LN	Возвращает натуральный логарифм числа
LOG	LOG	Возвращает логарифм числа по заданному основанию. Если основание опущено, то оно полагается равным 10 (см. раздел 2.1)
LOG10	LOG10	Возвращает десятичный логарифм числа (см. раздел 2.1)
SIN	SIN	Возвращает синус заданного угла (см. раздел 2.1)
SINH	SINH	Возвращает гиперболический синус числа
TAN	TAN	Возвращает тангенс заданного угла (см. раздел 2.1)
TANH	TANH	Возвращает гиперболический тангенс числа
ГРАДУСЫ	DEGREES	Преобразует радианы в градусы
ЗНАК	SIGN	Возвращает 1, если число положительное, 0, если число равно 0 и -1, если число отрицательное
КОРЕНЬ	SQRT	Возвращает положительное значение квадратного корня из неотрицательного числа
МОБР	MINVERSE	Возвращает обратную матрицу (см. раздел 3.2)
МОПРЕД	MDETERM	Возвращает определитель матрицы (см. раздел 3.2)
МУМНОЖ	MMULT	Возвращает произведение матриц (см. раздел 3.2)
НЕЧЁТ	ODD	Возвращает число, округленное до ближайшего нечетного целого
ОКРВВЕРХ	CEILING	Возвращает результат округления с избытком до ближайшего числа, кратного точности

Таблица В.1 (продолжение)

Функция (рус.)	Функция (англ.)	Назначение
ОКРВНИЗ	FLOOR	Возвращает результат округления числа до заданной точности с недостатком
ОКРУГЛ	MROUND	Округляет число до указанного количества десятичных разрядов
ОКРУГЛВВЕРХ	ROUNDUP	Округляет число до ближайшего большего по модулю целого
ОКРУГЛВНИЗ	ROUNDDOWN	Округляет число до ближайшего меньшего по модулю целого
ОСТАТ	MOD	Возвращает остаток от деления числа на делитель
ОТБР	TRUNC	Усекает число до целого, отбрасывая его дробную часть
ПИ	PI ()	Возвращает значение числа π (см. раздел 2.1)
ПРОИЗВЕД	PRODUCT	Возвращает произведение чисел, заданных в качестве аргументов
ПРОМЕЖУТОЧ- НЫЕ .ИТОГИ	SUBTOTAL	Возвращает промежуточный итог в список или базу данных
РАDIАНЫ	RADIANS	Преобразует градусы в радианы
РИМСКОЕ	ROMAN	Преобразует число из арабской записи в римскую
СЛЧИС	RAND	Возвращает равномерно распределенное случайное число, большее либо равное 0 и меньшее 1
СТЕПЕНЬ	POWER	Возвращает результат возведения числа в степень
СУММ	SUM	Возвращает сумму всех чисел, входящих в список аргументов (см. раздел 4.7)
СУММЕСЛИ	SUMIF	Возвращает сумму значений в ячейках, специфицированных заданным критерием
СУММКВ	SUMSQ	Возвращает сумму квадратов аргументов (см. раздел 7.5)
СУММКВРАЗН	SUMXMY2	Возвращает сумму квадратов разностей соответствующих значений в двух массивах (см. раздел 7.5)

Таблица В.1 (окончание)

Функция (рус.)	Функция (англ.)	Назначение
СУММПРОИЗВ	SUMPRODUCT	Возвращает сумму произведений соответствующих элементов массивов (см. раздел 4.7)
СУММРАЗНKB	SUMX2MY2	Возвращает сумму разностей квадратов соответствующих значений в двух массивах (см. раздел 7.5)
СУММСУММКВ	SUMX2PY2	Возвращает сумму сумм квадратов соответствующих элементов двух массивов (см. раздел 7.5)
СЧЁТЕСЛИ	COUNTIF	Возвращает количество непустых ячеек заданного диапазона, удовлетворяющих заданному критерию (см. раздел 4.8)
СЧИТАТЬПУСТОТЫ	COUNTBLANK	Возвращает количество пустых ячеек в заданном диапазоне
ФАКТР	FACT	Возвращает факториал числа
ЦЕЛОЕ	INT	Возвращает число, округленное до ближайшего меньшего целого
ЧЁТН	EVEN	Возвращает число, округленное до ближайшего четного целого
ЧИСЛКОМБ	COMBIN	Возвращает количество комбинаций для заданного числа объектов

Таблица В.2. Функции категории **Дата и время** (Date & Time)

Функция (рус.)	Функция (англ.)	Назначение
ВРЕМЗНАЧ	TIMEVALUE	Возвращает порядковый номер для указанного времени в текстовом формате
ВРЕМЯ	TIME	Возвращает порядковый номер для указанного времени
ГОД	YEAR	Возвращает год, соответствующий указанной дате
ДАТА	DATE	Возвращает порядковый номер указанной даты (см. раздел 4.8)
ДАТАЗНАЧ	DATEVALUE	Возвращает порядковый номер указанной даты, представленной в текстовом формате

Таблица В.2 (окончание)

Функция (рус.)	Функция (англ.)	Назначение
ДЕНЬ	DAY	Возвращает день месяца указанной даты
ДЕНЬНЕД	WEEKDAY	Возвращает день недели указанной даты
ДНЕЙ360	DAYS360	Возвращает количество дней между двумя датами на основе 360-дневного года
МЕСЯЦ	MONTH	Возвращает месяц, соответствующий указанной дате
МИНУТЫ	MINUTE	Возвращает минуты, соответствующие указанной дате
СЕГОДНЯ	TODAY	Возвращает текущую дату в числовом формате
СЕКУНДЫ	SECOND	Возвращает секунды, соответствующие указанной дате
ТДАТА	NOW	Возвращает текущую дату и время в числовом формате
ЧАС	HOUR	Возвращает час, соответствующий указанной дате

Таблица В.3. Функции категории Логические (Logical)

Функция (рус.)	Функция (англ.)	Назначение
ЕСЛИ	IF	Возвращает первое значение, если логическое выражение при вычислении дает значение ИСТИНА, и второе значение, если ЛОЖЬ (см. раздел 2.2)
И	AND	Возвращает значение ИСТИНА, если все аргументы имеют значение ИСТИНА; возвращает значение ЛОЖЬ, если хотя бы один аргумент имеет значение ЛОЖЬ (см. раздел 2.2)
ИЛИ	OR	Возвращает ИСТИНА, если хотя бы один из аргументов имеет значение ИСТИНА; возвращает ЛОЖЬ, если все аргументы имеют значение ЛОЖЬ (см. раздел 2.2)
ИСТИНА	TRUE	Возвращает логическое значение ИСТИНА (см. раздел 2.2)

Таблица В.3 (окончание)

Функция (рус.)	Функция (англ.)	Назначение
ЛОЖЬ	FALSE	Возвращает логическое значение ЛОЖЬ (см. раздел 2.2)
НЕ	NOT	Меняет на противоположное логическое значение аргумента (см. раздел 2.2)

Таблица В.4. Функции категории **Ссылки и массивы** (Reference)

Функция (рус.)	Функция (англ.)	Назначение
АДРЕС	ADDRESS	Возвращает адрес ячейки в виде текста, используя номер строки и номер столбца
ВПР	VLOOKUP	Просматривает левый столбец массива в поисках определенного значения и возвращает значение из указанной ячейки
ВЫБОР	CHOOSE	Возвращает значение из списка аргументов с данным номером индекса
ГПР	HLOOKUP	Просматривает верхнюю строку массива в поисках определенного значения и возвращает значение из указанной ячейки
ДВССЫЛ	INDIRECT	Возвращает значение, находящееся в ячейке, ссылка на которую находится в указанной ячейке
ИНДЕКС	INDEX	Возвращает значение указанной ячейки или массив значений в аргументе массив (см. раздел 7.9.2)
ОБЛАСТИ	AREAS	Возвращает количество областей в ссылке
ПОИСКПОЗ	MATCH	Возвращает относительную позицию элемента массива, который соответствует указанному значению указанным образом (см. раздел 4.10)
ПРОСМОТР	LOOKUP	Ищет значения в векторе или массиве. Функция ПРОСМОТР имеет две синтаксические формы: вектор и массив
СМЕЩ	OFFSET	Возвращает ссылку заданной высоты и ширины, отстоящую от другой ссылки на заданное количество строк и столбцов

Таблица В.4 (окончание)

Функция (рус.)	Функция (англ.)	Назначение
СТОЛБЕЦ	COLUMN	Возвращает номер столбца по заданной ссылке
СТРОКА	ROW	Возвращает номер строки по заданной ссылке
ТРАНСП	TRANSPOSE	Возвращает транспонированный массив (см. раздел 3.2)
ЧИСЛСТОЛБ	COLUMNS	Возвращает количество столбцов в ссылке
ЧСТРОК	ROWS	Возвращает количество строк в ссылке

Таблица В.5. Функции категории *Текстовые* (Text)

Функция (рус.)	Функция (англ.)	Назначение
ДЛСТР	LEN	Возвращает количество символов в текстовой строке
ЗАМЕНИТЬ	REPLACE	Заменяет часть текстовой строки на другую текстовую строку
ЗНАЧЕН	VALUE	Преобразует текст в число
КОДСИМВ	CODE	Возвращает числовой код первого символа в текстовой строке
ЛЕВСИМВ	LEFT	Возвращает первые символы текстовой строки
НАЙТИ	FIND	Находит вхождение одной текстовой строки в другую и возвращает номер символа, с которого начинается первое вхождение искомой строки
ПЕЧСИМВ	CLEAN	Удаляет из текста все непечатные символы
ПОВТОР	REPT	Повторяет текст заданное число раз
ПОДСТАВИТЬ	SUBSTITUTE	Заменяет текст в строке
ПОИСК	SEARCH	Возвращает позицию первого вхождения символа или текстовой строки при поиске слева направо
ПРАВСИМВ	RIGHT	Возвращает последние символы текстовой строки

Таблица В.5 (окончание)

Функция (рус.)	Функция (англ.)	Назначение
ПРОПИСН	UPPER	Делает все буквы в тексте прописными
ПРОПНАЧ	PROPER	Первая буква в тексте, а также все первые буквы, следующие за символами, отличными от букв, делаются прописными. Все прочие буквы в тексте делаются строчными
ПСТР	MID	Возвращает заданное число символов из строки текста, начиная с указанной позиции
РУБЛЬ	DOLLAR	Преобразует число в текст, используя денежный формат с округлением до заданного числа десятичных знаков
СЖПРОБЕЛЫ	TRIM	Удаляет из текста все пробелы, за исключением одиночных пробелов между словами
СИМВОЛ	CHAR	Возвращает символ с заданным кодом
СОВПАД	EXACT	Сравнивает две строки текста и возвращает значение ИСТИНА, если они в точности совпадают и ЛОЖЬ — в противном случае
СТРОЧН	LOWER	Преобразует символы в текстовой строке из верхнего регистра в нижний
СЦЕПИТЬ	CONCATENATE	Объединяет несколько текстовых элементов в один
Т	T	Если значение является текстом или ссылается на текст, то функция Т возвращает само это значение. Если значение не является текстом и не ссылается на текст, то функция Т возвращает " " (пустой текст)
ТЕКСТ	TEXT	Преобразует значение в текст в заданном числовом формате (см. раздел 2.7)
ФИКСИРОВАННЫЙ	FIXED	Округляет число до заданного количества десятичных цифр, форматирует число в десятичном формате с использованием запятых и точек и возвращает результат в виде текста

Таблица В.6. Функции категории *Финансовые* (Finance)

Функция (рус.)	Функция (англ.)	Назначение
АМГД (стоимость; остаточная стоимость; время эксплуатации; период)	SYD	Возвращает годовую амортизацию имущества для указанного периода (см. раздел 4.6)
АМР (стоимость; остаток; период)	SLN	Возвращает величину непосредственной амортизации имущества за один период (см. раздел 4.6)
БЗ (ставка; кпер; плата; нз; тип)	FV	Возвращает будущее значение вклада на основе периодических постоянных платежей и постоянной процентной ставки (см. раздел 4.5)
ВНДОХ (значения; прогноз)	IRR	Возвращает внутреннюю скорость оборота для ряда последовательных операций с наличными, представленными числовыми значениями (см. раздел 4.2)
ДДОБ (нач_стоимость; ост_стоимость; время_эксплуатации; период; коэффициент)	DDB	Возвращает величину амортизации имущества для указанного периода, используя метод двукратного учета амортизации или иной явно указанный метод (см. раздел 4.6)
ДОБ (нач_стоимость; ост_стоимость; время_эксплуатации; период; месяц)	DB	Возвращает амортизацию имущества на заданный период, используя метод постоянного учета амортизации (см. раздел 4.6)
КПЕР (ставка; платеж; нз; бз; тип)	NPER	Возвращает общее количество периодов выплаты для данного вклада на основе периодических постоянных выплат и постоянной процентной ставки (см. раздел 4.5)
МВСД (значения; ф_ставка; р_ставка)	MIRR	Возвращает модифицированную внутреннюю скорость оборота средств для ряда последовательных периодических операций с наличными
НОРМА (кпер; выплата; нз; бз; тип; нач_прибл)	RATE	Возвращает процентную ставку за один период при выплате ренты (см. раздел 4.5)

Таблица В.6 (окончание)

Функция (рус.)	Функция (англ.)	Назначение
НПЗ (ставка; значение1; значение2; ...)	NPV	Возвращает чистый текущий объем вклада, вычисляемый на основе ряда последовательных поступлений наличных и нормы амортизации (см. раздел 4.2)
ОСНПЛАТ (ставка; период; кпер; нз; бз; тип)	PMT	Возвращает величину выплаты на данный период на основе периодических постоянных платежей и постоянной процентной ставки (см. раздел 4.4)
ПДОВ (стоимость; остаток; период; нач_период; кон_период; коэфф; без_переключения)	VDB	Возвращает значение амортизации имущества за данный период, включая конкретные периоды, используя метод двойного процента со снижающегося остатка или иной явно указанный метод
ПЗ (ставка; кпер; выплата; бз; тип)	PV	Возвращает текущий объем вклада (см. раздел 4.3)
ПЛПРОЦ (ставка; период; кпер; нз; бз; тип)	IPMT	Возвращает платежи по процентам за данный период на основе периодических постоянных выплат и постоянной процентной ставки (см. раздел 4.4)
ППЛАТ (ставка; кпер; нз; бз; тип)	PMT	Возвращает величину выплаты за один период годовой ренты на основе постоянных выплат и постоянной процентной ставки (см. раздел 4.1)

Таблица В.7. Функции категории **Работа с базой данных** (Database)

Функция (рус.)	Функция (англ.)	Назначение
ВДДИСП	DVAR	Оценивает дисперсию генеральной совокупности по выборке, используя числа в указанном поле всех записей базы данных, удовлетворяющих критерию
ВДДИСПП	DVARP	Вычисляет дисперсию генеральной совокупности, используя числа в указанном поле всех записей базы данных, удовлетворяющих критерию

Таблица В.7 (окончание)

Функция (рус.)	Функция (англ.)	Назначение
ВДПРОИЗВЕД	DPRODUCT	Перемножает значения в указанном поле всех записей, удовлетворяющих критерию
ВВСУММ	DSUM	Суммирует числа в указанном поле всех записей базы данных, удовлетворяющих критерию
ВИЗВЛЕЧЬ	DGET	Извлекает отдельное значение из базы данных, удовлетворяющее критерию
ВСЧЁТ	DCOUNT	Подсчитывает количество ячеек, содержащих числа, во всех записях базы данных, удовлетворяющих критерию
ВСЧЁТА	DCOUNTA	Подсчитывает непустые ячейки в указанном поле всех записей базы данных, удовлетворяющих критерию
ДМАКС	DMAX	Возвращает наибольшее число в указанном поле всех записей базы данных, удовлетворяющих критерию
ДМИН	DMIN	Возвращает наименьшее число в указанном поле всех записей базы данных, удовлетворяющих критерию
ДСРЗНАЧ	DAVERAGE	Возвращает среднее значение в указанном поле всех записей, удовлетворяющих критерию
ДСТАНДОТКЛ	DSTDEV	Возвращает стандартное отклонение для выборки из генеральной совокупности, используя числа в указанном поле всех записей базы данных, удовлетворяющих критерию
ДСТАНДОТКЛП	DSTDEVP	Возвращает стандартное отклонение генеральной совокупности, используя числа в указанном поле всех записей базы данных, удовлетворяющих критерию

Таблица В.8. Функции категории **Проверка свойств и значений** (Information)

Функция (рус.)	Функция (англ.)	Назначение
ЕЛОГИЧ	ISLOGICAL	Возвращает логическое значение ИСТИНА, если проверяемая величина является логическим значением; в противном случае возвращается логическое значение ЛОЖЬ (см. раздел 7.17)

Таблица В.8 (продолжение)

Функция (рус.)	Функция (англ.)	Назначение
ЕНД	ISNA	Возвращает логическое значение ИСТИНА, если проверяемая величина является значением ошибки #Н/Д; в противном случае возвращается логическое значение ЛОЖЬ (см. раздел 7.17)
ЕНЕТЕКСТ	ISNOTEXT	Возвращает логическое значение ИСТИНА, если проверяемая величина не является текстом; в противном случае возвращается логическое значение ЛОЖЬ (см. раздел 7.17)
ЕОШ	ISERR	Возвращает логическое значение ИСТИНА, если проверяемая величина является одним из возможных ошибочных значений Excel, за исключением #Н/Д; в противном случае возвращается логическое значение ЛОЖЬ (см. раздел 7.17)
ЕОШИБКА	ISERROR	Возвращает логическое значение ИСТИНА, если проверяемая величина является одним из возможных ошибочных значений Excel; в противном случае возвращается логическое значение ЛОЖЬ (см. раздел 7.17)
ЕПУСТО	ISBLANK	Возвращает логическое значение ИСТИНА, если проверяемая величина является ссылкой на пустую ячейку; в противном случае возвращается логическое значение ЛОЖЬ (см. раздел 7.17)
ЕССЫЛКА	ISREF	Возвращает логическое значение ИСТИНА, если проверяемая величина является ссылкой; в противном случае возвращается логическое значение ЛОЖЬ (см. раздел 7.17)
ЕТЕКСТ	ISTEXT	Возвращает логическое значение ИСТИНА, если проверяемая величина является текстом; в противном случае возвращается логическое значение ЛОЖЬ (см. раздел 7.17)
ЕЧИСЛО	ISNUMBER	Возвращает логическое значение ИСТИНА, если проверяемая величина является числом; в противном случае возвращается логическое значение ЛОЖЬ (см. раздел 7.17)
ИНФОРМ	INFO	Возвращает информацию о текущей операционной среде

Таблица В.8 (окончание)

Функция (рус.)	Функция (англ.)	Назначение
НД	NA	Возвращает значение ошибки #Н/Д. #Н/Д, которое означает, что "нет доступного значения"
ТИП	TYPE	Возвращает тип значения
ТИП.ОШИБКИ	ERROR.TYPE	Возвращает номер, соответствующий одному из возможных ошибочных значений Excel
Ч	N	Возвращает значение, преобразованное в число
ЯЧЕЙКА	CELL	Возвращает информацию о формате, местоположении или содержимом левой верхней ячейки в ссылке

Таблица В.9. Функции категории Статистические (Statistical)

Функция (рус.)	Функция (англ.)	Назначение
ФРАСП	FDISP	Возвращает F-распределение вероятности
ФРАСПОВР	FINV	Возвращает обратное значение для F-распределения вероятностей
ЗТЕСТ	ZTEST	Возвращает двустороннее Р-значение z-теста
БЕТАОВР	BETAINV	Возвращает обратную функцию к интегральной функции плотности бета-вероятности
БЕТАРАСП	BETADIST	Возвращает интегральную функцию плотности бета-вероятности
БИНОМРАСП	BINOMDIST	Возвращает отдельное значение биномиального распределения
ВЕЙБУЛЛ	WEIBULL	Возвращает распределение Вейбулла
ВЕРОЯТНОСТЬ	PROB	Возвращает вероятность того, что значение из интервала находится внутри заданных пределов
ГАММАНЛОГ	GAMMALN	Возвращает натуральный логарифм гамма функции $\Gamma(x)$

Таблица В.9 (продолжение)

Функция (рус.)	Функция (англ.)	Назначение
ГАММАОВР	GAMMAINV	Возвращает обратное гамма-распределение
ГАММАРАСП	GAMMADIST	Возвращает гамма-распределение
ГИПЕРГЕОМЕТ	HYPGEOMDIST	Возвращает гипергеометрическое распределение
ДИСП	VAR	Возвращает дисперсию по выборке
ДИСПР	VARP	Возвращает дисперсию для генеральной совокупности
ДОВЕРИТ	CONFIDENCE	Возвращает доверительный интервал для среднего генеральной совокупности
КВАДРОТКЛ	DEVSQ	Возвращает сумму квадратов отклонений точек данных от их среднего
КВАРТИЛЬ	QUARTILE	Возвращает квартиль набора данных
КВПИРСОН	RSQ	Возвращает квадрат коэффициента корреляции Пирсона
КОВАР	COVAR	Возвращает ковариацию, т. е. среднее произведений отклонений для каждой пары точек данных
КОРРЕЛ	CORREL	Возвращает коэффициент корреляции (см. раздел 7.5)
КРИТБИНОМ	CRITBINOM	Возвращает наименьшее значение, для которого интегральное биномиальное распределение больше или равно заданному критерию
ЛГРФПРИВЛ	LOGEST	Возвращает экспоненциальное уравнение регрессии (см. раздел 5.5.3)
ЛИНЕЙН	LINEST	Возвращает линейное уравнение регрессии (см. раздел 5.5.2)
ЛОГНОРМОВР	LOGINV	Возвращает обратную функцию логарифмического нормального распределения
ЛОГНОРМРАСП	LOGNORMDIST	Возвращает интегральное логарифмическое нормальное распределение
МАКС	MAX	Возвращает максимальное значение из списка аргументов (см. раздел 4.10)

Таблица В.9 (продолжение)

Функция (рус.)	Функция (англ.)	Назначение
МЕДИАНА	MEDIAN	Возвращает медиану заданного набора чисел
МИН	MIN	Возвращает наименьшее значение в списке аргументов
МОДА	MODE	Возвращает наиболее часто встречающееся значение набора данных
НАИБОЛЬШИЙ	LARGE	Возвращает k-е наибольшее значение из множества данных (см. раздел 7.11)
НАИМЕНЬШИЙ	SMALL	Возвращает k-е наименьшее значение в множестве данных
НАКЛОН	SLOPE	Возвращает наклон линии линейной регрессии (см. раздел 5.5.2)
НОРМАЛИЗАЦИЯ	STANDARDIZE	Возвращает нормализованное значение для распределения, характеризуемого средним и стандартным отклонением
НОРМОБР	NORMINV	Возвращает обратное нормальное распределение для указанного среднего и стандартного отклонения
НОРМРАСП	NORMDIST	Возвращает нормальную функцию распределения для указанного среднего и стандартного отклонения
НОРМСТОБР	NORMSINV	Возвращает обратное значение стандартного нормального распределения
НОРМСТРАСП)	NORMSDIST	Возвращает стандартное нормальное интегральное распределение
ОТРВИНОМРАСП	NEGBINOMDIST	Возвращает отрицательное биномиальное распределение
ОТРЕЗОК	INTERCEPT	Возвращает отрезок, отсекаемый на оси линией линейной регрессии (см. раздел 5.5.2)
ПЕРЕСТ	PERMUT	Возвращает количество перестановок для заданного числа объектов, которые выбираются из общего числа объектов
ПЕРСЕНТИЛЬ	PERCENTILE	Возвращает k-ю перцентиль для значений из интервала

Таблица В.9 (продолжение)

Функция (рус.)	Функция (англ.)	Назначение
ПИРСОН	PEARSON	Возвращает коэффициент корреляции Пирсона
ПРЕДСКАЗ	FORECAST	Возвращает предсказанное значение функции в данной точке на основе уравнения линейной регрессии (см. раздел 5.5.2)
ПРОЦЕНТРАНГ	PERCENTRANK	Возвращает процентное содержание значения в множестве данных
ПУАССОН	POISSON	Возвращает распределение Пуассона
РАНГ	RANK	Возвращает ранг числа в списке чисел. Ранг числа — это его величина относительно других значений в списке (см. раздел 4.7)
РОСТ	GROWTH	Возвращает значения в соответствии с экспоненциальным уравнением регрессии (см. раздел 5.5.3)
СКОС	SKEW	Возвращает асимметрию распределения
СРГАРМ	HARMEAN	Возвращает среднее гармоническое множества данных
СРГЕОМ	GEOMEAN	Возвращает среднее геометрическое значений
СРЗНАЧ	AVERAGE	Возвращает среднее (арифметическое) значение (см. раздел 4.7)
СРОТКЛ	AVEDEV	Возвращает среднее абсолютных значений отклонений точек данных от среднего
СТАНДОТКЛОН	STDEV	Возвращает стандартное отклонение по выборке
СТАНДОТКЛОНП	STDEVP	Возвращает стандартное отклонение по генеральной совокупности
СТОШУХ	STEXY	Возвращает стандартную ошибку предсказанных значений y для каждого значения x в регрессии
СТЬЮДРАСП	TDIST	Возвращает t -распределение Стьюдента
СТЬЮДРАСПОВР	TINV	Возвращает обратное распределение Стьюдента для заданного числа степеней свободы

Таблица В.9 (окончание)

Функция (рус.)	Функция (англ.)	Назначение
СЧЁТ	COUNT	Возвращает количество чисел в списке аргументов (см. раздел 7.5)
СЧЁТЗ	COUNTA	Возвращает количество непустых значений в списке аргументов (см. раздел 7.5)
ТЕНДЕНЦИЯ	TREND	Возвращает значения в соответствии с линейным уравнением регрессии (см. раздел 5.5.2)
ТТЕСТ	TTEST	Возвращает вероятность, соответствующую критерию Стьюдента
УРЕЗСРЕДНЕЕ	TRIMMEAN	Возвращает усеченное среднее значение
ФИШЕР	FISHER	Возвращает преобразование Фишера
ФИШЕРОБВ	FISHERINV	Возвращает обратное преобразование Фишера
ФТЕСТ	FTEST	Возвращает результат F-теста
ХИ2ОБР	CHIINV	Возвращает обратную функцию для χ^2 -распределения
ХИ2РАСП	CHIDIST	Возвращает одностороннюю вероятность χ^2 -распределения
ХИ2ТЕСТ	CHITEST	Возвращает тест на независимость
ЧАСТОТА	FREQUENCY	Возвращает распределение частот в виде вертикального массива
ЭКСПРАСП	EXPONDIST	Возвращает экспоненциальное распределение
ЭКСЦЕСС	KURT	Возвращает эксцесс множества данных

Литература

1. Бунин Э. Excel Visual Basic для приложений. М.: Бином, 1996, 351 с.
2. Карлберг К. Excel 5.0 для Windows в вопросах и ответах. СПб.: ВНВ — Санкт-Петербург, 1995, 411 с.
3. Карлберг К. Бизнес-анализ с помощью Excel. Киев: Диалектика, 1997, 448 с.
4. Комягин В. Б. Программирование в Excel 5 и Excel 7 на языке Visual Basic. М.: Радио и связь, 1996, 319 с.
5. Комягин В. Б., Коцюбинская А. О. Excel 7 в примерах. М.: Нолидж, 1996, 429 с.
6. Николь Н., Альбрехт Р. Электронные таблицы Excel 5.0. М.: Эком, 1995, 343 с.
7. Николь Н., Альбрехт Р. Электронные таблицы Excel 5.0 для квалифицированных пользователей. М.: Эком, 1995, 301 с.
8. Персон Р. Microsoft Excel 97 в подлиннике. Том I, II. СПб.: ВНВ — Санкт-Петербург, 1997, 1272 с.
9. Таха Х. Введение в исследование операций. Том I. М.: Мир, 1985, 479 с.
10. Уокенбах Д. Библия пользователя Excel 7.0. М.: Эком, 1996, 565 с.
11. Фойц С. Windows 95 для пользователя. Киев: ВНВ, 1995, 557 с.