

# Владимир Дубянский, Людмила Скобликова РАЗРАБОТКА КОНФИГУРАЦИЙ В СРЕДЕ 1С:Предприятие



Бухгалтерский и оперативный учет Константы, справочники, документы, перечисления Запросы и отчеты Меню и панели инструментов Программные модули на встроенном языке Создание комплексной конфигурации Связь с внешними приложениями посредством OLE Automation



## Владимир Дубянский Людмила Скобликова

# Самоучитель разработка конфигураций в среде 1С:Предприятие 7.7

Санкт-Петербург «БХВ-Петербург» 2007 УДК 681.3.06

ББК 32.973.26-018.2

Д79

#### Дубянскиий, В. М.

Д79 Разработка конфигураций в среде 1С:Предприятие 7.7. Самоучитель /
 В. М. Дубянский, Л. К. Скобликова. — СПб.: БХВ-Петербург, 2007. — 448 с.: ил.

ISBN 978-5-94157-707-1

Описаны основные приемы программирования и конфигурирования компонент "1С:Предприятия" версии 7.7, как по отдельности, так и в различных сочетаниях. Приведены практические рекомендации по созданию констант, справочников, документов, перечислений, запросов и отчетов, меню и панелей инструментов, программных модулей на встроенном языке. Рассмотрена связь с внешними приложениями типа Microsoft Excel посредством OLE Automation. Основные приемы программирования и конфигурирования изложены с использованием сквозного примера. Каждое действие, описанное в книге, проиллюстрировано и сопровождается листингами процедур и функций.

Для программистов и пользователей системы "1С:Предприятие 7.7"

УДК 681.3.06 ББК 32.973.26-018.2

Главный редактор	Екатерина Кондукова	
Зам. главного редактора	Евгений Рыбаков	
Зав. редакцией	Григорий Добин	
Редактор	Татьяна Лапина	
Компьютерная верстка	Натальи Смирновой	
Корректор	Наталия Першакова	
Дизайн серии	Инны Тачиной	
Оформление обложки	Елены Беляевой	
Зав. производством	Николай Тверских	

#### Группа подготовки издания:

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 31.01.07. Формат 70×100<sup>1</sup>/<sub>16</sub>. Печать офсетная. Усл. печ. л. 36,12. Тираж 3000 экз. Заказ № "БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

> Отпечатано с готовых диапозитивов в ГУП "Типография "Наука" 199034, Санкт-Петербург, 9 линия, 12

## Оглавление

Введение 1			
Глава 1. Вводные сведения	3		
1.1. Концепция системы "1С:Предприятие"			
1.2. Инсталляция			
1.3. Что такое "Конфигурация"	8		
1.4. Функционирование системы	9		
1.5. Создание новой конфигурации	10		
1.6. Объекты метаданных системы "1С:Предприятие"	15		
Глава 2. Инструменты для разработки	21		
2.1. Конфигуратор	21		
2.2. Создание объектов метаданных	24		
2.3. Свойства конфигурации			
2.4. Сохранение базы данных			
2.5. Встроенный язык системы			
2.6. Диалог при сохранении конфигурации	43		
Глава 3. Пользовательский интерфейс и администрирование	45		
3.1. Индивидуальный пользовательский интерфейс	46		
3.2. Создание набора пользовательских прав	49		
3.3. Создание нового пользователя	51		
3.4. Общие замечания	54		
Глава 4. Сквозной пример	55		
4.1. Создание базовых объектов учета			
4.2. Организация бухгалтерского учета	57		
4.3. Организация оперативного учета			
4.4. Организация расчета заработной платы	59		
Глава 5. Константы и обработки	63		
5.1. Настройки первоначальных свелений об организации	63		
5.2. Примеры. Создание констант и обработок для Trade.Ltd	66		
Глава 6. Справочники и перечисления	81		
6.1. Общие свеления о справочниках			
6.1.1. Форма элемента			
6.1.2. Подчиненные справочники			
6.2. Понятие о выборке			
6.3. Объект Таблица для построения отчета	97		

1	100
6.4.1. Создание перечислений	101
6.5. Печатные формы справочников	102
6.6. Создание основных справочников конфигурации	112
Глара 7. Покумонти	117
тлава 7. документы	107
/.1. Построчное заполнение документа	127
7.2. Заполнение документа методом подоора	129
7.3. Локальный контекст и доступность процедур и функции	135
7.4. Итоговые значения колонок документа	139
7.5. Печатные формы документов	141
7.6. Печатные формы как отдельные фаилы	14/
/./. Список значении	149
7.8. Оощие реквизиты	153
7.10. Питета – сили состать сили в состать сили состать с	155
7.10. Другие документы учеоной конфигурации	159
Глава 8. Запросы и отчеты	167
8.1. Начало созлания отчета	167
8.2. Запрос	168
	101
Глава 9. Таблица значений	181
Глава 10. Бухгалтерский учет	193
10.1 Краткое описание компоненты	193
10.1. Краткое описание компоненты	193 195
10.1. Краткое описание компоненты 10.2. План счетов 10.3. Операции, проволки	193 195 197
<ul> <li>10.1. Краткое описание компоненты</li> <li>10.2. План счетов</li></ul>	193 195 197 197
<ul> <li>10.1. Краткое описание компоненты</li> <li>10.2. План счетов</li></ul>	193 195 197 197 198
<ul> <li>10.1. Краткое описание компоненты</li></ul>	193 195 197 197 198 199
<ul> <li>10.1. Краткое описание компоненты</li></ul>	193 195 197 197 198 199 200
<ul> <li>10.1. Краткое описание компоненты</li></ul>	193 195 197 197 198 199 200 201
<ul> <li>10.1. Краткое описание компоненты</li></ul>	193 195 197 197 198 199 200 201 223
<ul> <li>10.1. Краткое описание компоненты</li></ul>	193 195 197 197 197 198 198 199 200 201 223
<ul> <li>10.1. Краткое описание компоненты</li></ul>	
<ul> <li>10.1. Краткое описание компоненты</li></ul>	193 193 195 197 197 198 200 201 223 233 233 234 240 241 249 259
<ul> <li>10.1. Краткое описание компоненты</li></ul>	
<ul> <li>10.1. Краткое описание компоненты</li></ul>	
<ul> <li>10.1. Краткое описание компоненты</li></ul>	

12.2.2. Алгоритмы объектов	268
12.2.3. Особенности проведения документов	272
12.2.4. Последовательность документов	276
Глава 13. Запросы и отчетность	281
13.1. Понятие запросов	281
13.2. Примеры. Создание отчета "Взаиморасчеты"	286
13.3. Примеры. Создание отчета "Продажи"	292
Глава 14. Расчет	299
14.1. Краткое описание компоненты	299
14.2. Журналы расчетов, виды расчетов, календари	300
14.3. Применение компоненты "Расчет"	310
14.3.1. Создание объектов в сквозном примере компоненты "Расчет"	311
14.4. Написание алгоритмов в объектах компоненты "Расчет"	325
Глава 15. Отчетность в компоненте "Расчет"	351
15.1. Создание запросов по журналам расчетов	351
15.2. Примеры. Создание отчета "Расчетная ведомость"	356
15.3. Примеры. Создание отчета "Свод по заработной плате"	363
Глава 16. Некоторые дополнительные сведения	375
16.1. Экспорт данных с помощью файлов	375
16.2. Совместная работа системы "1С:Предприятие"	
и электронных таблиц Microsoft Excel	380
16.3. Работа с метаданными с помощью встроенного языка	392
Глава 17. Краткие сведения о работе с отладчиком	401
17.1. Замер производительности	405
Глава 18. Итоговоо описанио возвоботанной конфигурании	400
плава то. итотовое описание разраоотанной конфигурации	409
Заключение	426
Приложения	427
Приложение 1. Команды, выполняемые с помощью меню конфигуратора	429
Приложение 2. Быстрые (горячие) клавиши	434
Предметный указатель	440

## Введение

Книга предназначена для обучения элементарной работе в конфигураторе "1С:Предприятия 7.7". Она написана в форме самоучителя, но не является переработанной документацией к программе. Авторы попытались, используя опыт и материалы предыдущей книги одного из авторов "1С:Предприятие. Конфигурирование и администрирование для начинающих" ("БХВ-Петергбург", 2005), показать ключевые моменты конфигурирования совершенно не подготовленному в плане программирования читателю. Поэтому в книге много скриншотов, рисунков и листингов программ. Подробно обсуждаются базовые объекты, а для специализированных компонент "Бухгалтерский учет", "Оперативный учет" и "Расчет" приводятся основные сведения. Обучение основным навыкам приведет к следующим результатам:

- большинство текущих проблем конфигурирования вы сможете решить сами;
- □ будете четко осознавать, что вам нужно от разработчика, как это удобнее сделать для вашего конкретного случая, сколько это может занять времени и сколько это может стоить;
- □ значительно повысится ваш статус как специалиста.

Предполагается, что наличие такой обучающей книги позволит ускорить освоение технологии конфигурирования в системе "1С:Предприятие 7.7". Сквозной пример, который мы будем разбирать на протяжении всей книги, — вполне работоспособная конфигурация, правда, очень простая. При этом прикладным аспектам конфигурации как законченному программному продукту уделено минимальное внимание, а именно, лишь в пределах, необходимых для определенной стройности и логичности изложения. Основной упор сделан на описание работы с конкретными объектами: справочниками, документами и т. п. Такой подход позволяет использовать отдельные главы вне контекста книги для изучения только отдельных моментов в конфигурировании. Тем не менее, знания, полученные в результате изучения книги, помогут вам самостоятельно совершенствовать, улучшать ваши рабочие конфигурации и писать новые.

Книга не является ни документацией, ни дополнением к документации к программному продукту.

Для разработки приложений, безусловно, необходимо наличие книг, входящих в комплект поставки, в частности:

- □ "Конфигурирование и администрирование" (в 2-х частях);
- □ "Описание встроенного языка" (в 2-х частях);
- 🗖 "Руководство по установке и запуску".

В процессе работы над данным пособием авторы использовали материалы указанных книг, а также следующую литературу:

- 1. Байдаков В., Нуралиев С., Шевченко А. Введение в конфигурирование 1С:Бухгалтерии 7.7. М.: Фирма "1С", 2000.
- 2. Дубянский В. М. "1С:Предприятие". Конфигурирование и администрирование для начинающих. СПб.: БХВ-Петербург, 2005.
- Лихарев М., Нуралиев С., Службин Д., Русанов Д., Федоров Б., Филиппов В. "1С:Предприятие", версия 7.5. Методические рекомендации по конфигурированию и администрированию. — М.: Фирма "1С", 1998.
- Ожигов В. Л., Меломуд О. Л., Гвоздев Д. В., Гордеев О. В. Методические рекомендации по анализу и построению технической архитектуры в проектах внедрения информационных систем на базе системы программ "1С:Предприятие". — Икс Технологии, 2000.
- 5. Соловьев И. В. "1С:Предприятие" 7.7. Базовые объекты. М.: Фирма "1С", 2000.
- 6. Материалы сайта http://training-1c.hotbox.ru.

Авторы предполагают, что читатели имеют элементарные навыки работы с персональным компьютером в среде Windows'\*, т. е. представляют структуру каталогов, могут создавать, копировать, уничтожать файлы и папки, знакомы с понятием "Имя файла" и "Расширение файла" и т. д.

Если при прочтении книги вы обнаружите какие-либо упущения, неточности, сообщите о них по электронному адресу **DVM@MAIL.KZ**. Авторы будут благодарны за замечания и предложения.



## Вводные сведения

На современном рынке представлено достаточно большое количество разнообразных бухгалтерских программ. "1С:Предприятие" выгодно отличается от остальных следующими возможностями:

- возможность гибкой настройки трудно представить себе изменения в законодательстве, в плане счетов и т. п., которые нельзя было бы реализовать в программе "1С:Предприятие", причем с сохранением всех раннее введенных данных;
- возможность решения широкого круга учетных задач: бухгалтерия и торговля, управление персоналом, финансовый анализ, управление проектами и т. д.;
- открытая система возможно взаимодействие с другими программами, обмен данными, написание внешних компонент, расширяющих возможности программы;
- учет может вестись как в маленькой фирме с одним компьютером, так и на больших предприятиях с десятками рабочих мест;
- возможность синхронизировать учет в территориально разделенных организациях, причем передача данных возможна любым путем — от дискеты до оптоволокна;
- 🗖 развитая система поддержки конечных пользователей.

### 1.1. Концепция системы "1С:Предприятие"

Стандартное представление об экономических программах фирмы "1С" — "1С:Бухгалтерия", которая является программой, предназначенной для веде-

ния бухгалтерского учета. Однако это только одна из частей системы программ для автоматизации различных областей экономической деятельности — "1С:Предприятие". Система программ — потому, что "1С:Предприятие" имеет компонентную структуру. В настоящий момент фирмой "1С" разработаны три основные компоненты: "Бухгалтерский учет" ("1С:Бухгалтерия", о которой говорилось ранее) "Оперативный учет" и "Расчет". Это своеобразный конструктор из частей-компонент, используя которые по отдельности или в самых разных сочетаниях, мы можем решать различные задачи.

Компонента "Бухгалтерский учет" предназначена для ведения учета на основе бухгалтерских операций. Она обеспечивает ведение планов счетов, ввод проводок, получение бухгалтерских итогов. Компонента используется для автоматизации бухгалтерского учета в соответствии с любым законодательством и методологией учета.

Компонента "Оперативный учет" предназначена для учета наличия и движения средств в различных разрезах в *реальном времени*. Она используется для учета запасов товарно-материальных ценностей, взаиморасчетов с контрагентами и т. д. Компонента позволяет отражать в учете операции хозяйственной жизни предприятия непосредственно в момент их совершения.

Компонента "Расчет" предназначена для выполнения *сложных периодических расчетов*. Она может использоваться для расчета заработной платы любой сложности, для расчетов по ценным бумагам и для других видов расчетов.

Обратим особое внимание на различия этих компонент. К примеру, пользуясь компонентой "Оперативный учет", вы сможете эффективно учитывать ТМЦ на складах и вести торговлю, но не сможете создавать бухгалтерских операций и проводок. Если вам нужны проводки и другие возможности бухгалтерского учета — используйте дополнительно компоненту "Бухгалтерский учет". Вместе с тем, при небольших объемах торговых операций можно обойтись только одной компонентой "Бухгалтерский учет". С помощью этой компоненты можно вести бухгалтерский учет и расчет зарплаты. Но сложный расчет зарплаты лучше написать на компоненте "Расчет" — в конечном итоге это будет более эффективно при меньшей цене разработки.

В дополнение к этим "учетным" компонентам существуют еще и две "вспомогательные": "Управление распределенными информационными базами" и "WEB-расширение". "Управление распределенными информационными базами" (УРБД) предназначено для синхронизации территориально разделенных учетных баз. Допустим, у вашей организации есть центральный офис и несколько филиалов в разных городах, при этом весь учет вы ведете в центральном офисе. Тогда данные из учетных баз с периферии поступают в компоненту УРБД, где синхронизируются, и вы видите весь учет в центральном офисе. "WEB-расширение" позволяет вести работу с удаленными филиалами через Интернет. Например, если у вас есть магазин, а склад находится в другом месте, то организовать работу склада и магазина можно с помощью компоненты "WEB-расширение". Следует заметить, что деление на "учетные" и "вспомогательные" носит условный характер. А состав компонент системы постепенно расширяется.

Как видите, по своему составу "1С:Предприятие" довольно сложная система. Но это еще не все.

По функциональной полноте "1С:Предприятие" делится на следующие уровни:

- □ базовый отличительной особенностью этой поставки является невозможность пользователю производить настройку в режиме конфигуратора;
- стандарт (для бухгалтерской компоненты) урезаны некоторые возможности (соответственно в уровне "проф" они присутствуют):
  - количество субконто не может превышать трех;
  - можно оперировать только одним планом счетов;
  - невозможно использовать разделитель учета в проводках;
  - количество уровней вложенности в справочниках не больше трех;
- □ проф (для бухгалтерской компоненты) присутствуют все возможности компоненты "Бухгалтерский учет":
  - количество субконто до пяти;
  - можно оперировать несколькими планами счетов;
  - можно использовать разделитель учета в проводках;
  - количество уровней вложенности в справочниках до десяти.

По возможному количеству пользователей система "1С:Предприятие" делится на следующие:

- локальная на один компьютер. С информационной базой сможет работать только один пользователь;
- сетевая на неограниченное количество компьютеров;
- □ трехпользовательская поставка (для компонент "Оперативный учет" и "Расчет") — с информационной базой могут одновременно работать не более трех пользователей;
- □ SQL-версия на неограниченное количество компьютеров с использованием клиент-серверной технологии;
- набор для небольшой фирмы сетевая бухгалтерия плюс трехпользовательская поставка для компонент "Оперативный учет" и "Расчет".

Таким образом, окончательная комплектация может составляться из всевозможных вариантов, приведенных выше, но с учетом следующих замечаний:

- 1. Сетевые и SQL-версии не могут быть стандартными и базовыми.
- 2. Поставки "1С:Бухгалтерский учет" для трех пользователей не предусмотрено.

Примеры поставок:

- "1С:Бухгалтерский учет" (стандарт);
- "1С:Оперативный учет" (сетевая версия) + "1С:Бухгалтерский учет" сетевая версия;
- "1С:Бухгалтерский учет" (Проф) + "1С:Расчет" (Проф);
- "1С:Предприятие" (SQL).

О программе		×
10	1С:Предприятие 7.7 (сетевая версия) Copyright® '1C', 1996 - 1999	(7.70.025)
	Лицензия на программу принадлежит: дом DVM	
	Установленные компоненты: Бухгалтерский учет Оперативный учет Расчет	
Информационная база: E:\Самоучитель1С77\КонфБух\23080 ОК СамоучительБух Имя пользователя:		

Рис. 1.1. Информация о программе

Выбрав, к примеру, инсталлятор "1С:Бухгалтерский учет" и установив с него программу, вы получаете компоненту "1С:Бухгалтерия", а если это был инсталлятор "1С:Предприятие", то вы получите сразу все три компоненты. Возможен и промежуточный вариант. Для нужд вашего предприятия потребуется использовать две компоненты — "1С:Бухгалтерский учет" и "1С:Оперативный учет", а вот компонента "1С:Расчет" вам не нужна. Что

тогда делать? Тогда вы берете инсталлятор "1С:Бухгалтерский учет", устанавливаете эту компоненту, затем берете инсталлятор "1С:Оперативный учет" и также устанавливаете компоненту "1С:Оперативный учет", причем обязательно в тот же каталог, куда вы до этого установили компоненту "1С:Бухалтерский учет". Правда при этом вам надо соблюдать соответствие инсталляторов по функциональной полноте и по возможному количеству пользователей: нельзя ставить в один каталог программы "1С:Бухгалтерский учет" (стандарт) и "1С:Оперативный учет" (сетевая).

Знание этих нюансов позволит вам уже сейчас оценить с профессиональной точки зрения то, что вы имеете, или запланировать, что нужно приобрести.

В заключение раздела выясним, где можно посмотреть ту информацию о "1С:Предприятии", которую обсуждали выше. Откройте "1С:Предприятие". Войдите в меню **Помощь**, выберите пункт **О программе**. Появится окно, подобное рис. 1.1. В нем — необходимая информация об установленных компонентах. Как видите, установлены локальные (на один компьютер) "учетные" компоненты системы программ "1С:Предприятие".

### 1.2. Инсталляция

Если вы более или менее умеете работать с компьютером, тогда процесс установки программы у вас не вызовет затруднений. Инсталляция идет с дискет или компакт-дисков, которые содержат программу установки Setup. Все версии требуют наличия на компьютере Microsoft Windows'95 или Windows NT 3.51 или их более новых версий — Microsoft Windows XP, например. При первой установке программа Setup запрашивает имя пользователя и наименование компании. Программа сохраняет данную информацию и выводит ее каждый раз при запуске "1С:Предприятия". Далее необходимо следовать инструкциям на экране. Когда программа инсталляции предложит вам установить типовую конфигурацию — откажитесь. Нам она сейчас не нужна. Существуют особенности инсталляции сетевой и SQL-серверной версий, но в данной книге мы их описывать не станем. После завершения процедуры инсталляции "1С:Предприятие" будет установлено с собственной группой программ. После инсталляции программы выключите компьютер, вставьте в порт принтера ключ защиты. Для того чтобы подключить принтер, в ключ защиты вставьте кабель принтера, включите компьютер. Затем нажмите кнопку Пуск (Start), далее — Программы, 1С:Предприятие 7.7, Установка драйвера защиты (рис. 1.2). На экране должно появиться сообщение об успешной установке драйвера защиты. Перезагрузите компьютер.

Установка драйвера защиты требуется для того, чтобы программа "1С:Предприятие" обнаружила ключ защиты на вашем компьютере.



Рис. 1.2. Выбор пункта меню Установка драйвера защиты

#### Предупреждение

Если аппаратный ключ не установлен, либо не исправен, либо версия драйвера защиты не соответствует комплекту поставки, будет выдано сообщение: "Не обнаружен ключ защиты программы!!!", и запуск системы будет прерван.

Подробнее нюансы установки и работы с ключом защиты описаны в "Руководстве по установке и запуску", которое поставляется с программой в составе пакета документации.

### 1.3. Что такое "Конфигурация"

Все составляющие системы программ "1С:Предприятие" можно разделить на *технологическую платформу* и *конфигурации. Технологическая платформа* представляет собой набор различных механизмов, используемых для автоматизации экономической деятельности и не зависящих от конкретного законодательства и методологии учета. Другими словами, *технологическая плат*-

форма — это "мотор", который движет алгоритм автоматизации вашего предприятия. Действительно, ведь "1С:Предприятие" используется во всех странах СНГ и, бывает, дальнего зарубежья. Разработчикам невозможно предусмотреть всех нюансов. А вот сам алгоритм расчетов реализуют программисты на местах, посредством конфигураций, которые и являются собственно прикладными решениями. Каждая конфигурация ориентирована на автоматизацию определенной сферы экономической деятельности и, разумеется, отвечает принятому законодательству. Используя определения документации, конфигурация — это конкретный набор объектов, структур информационных массивов и алгоритмов их обработки. Понятно, что невозможно использовать только конфигурацию — она не будет работать без "мотора" — технологической платформы. А использование технологической платформы без конфигурации лишено смысла — она ведь не будет вести учет. Вместе с конфигурацией система программ "1С:Предприятие" выступает как некоторый готовый к использованию программный продукт, который может быть ориентирован как на конкретное предприятие, так и на определенные типы предприятий и классы решаемых задач. И это является основной особенностью системы.

### 1.4. Функционирование системы

Функционирование системы делится на два процесса — конфигурирование и исполнение. Результатом конфигурирования является некоторая конфигурация, включающая в себя структуру базы данных и алгоритмы ее обработки. При выполнении первого этапа специалист, осуществляющий конфигурирование, в сущности, строит описание модели предметной области (например, бухгалтерии) средствами системы. Его задача — создать такую конфигурацию объектов предметной области, используя которую пользователь в режиме исполнения смог бы выполнять обработку данных, используя понятия конкретной предметной области. То есть программист в конфигурации должен предусмотреть возможность создания, к примеру, документа "Счет", возможности резервирования товара по счету, а пользователь уже будет выписывать счета. Программист же при создании конфигурации не может вводить пользовательскую информацию в базу данных.

Таким образом, в процессе конфигурирования формируется структура информационной базы и алгоритмы ее обработки.

В процессе же исполнения используются как штатные алгоритмы обработки данных (реализуемые технологической платформой), так и алгоритмы, созданные программистом в процессе конфигурирования.

В данном пособии будет подробно рассматриваться именно работа на этапе конфигурирования, в то время как работе пользователя на этапе исполнения конфигурации будет уделено минимально необходимое внимание.

### 1.5. Создание новой конфигурации

В начале работы с собственной конфигурацией создайте на диске вашего компьютера папку, где будут храниться конфигурация и набор файлов информационной базы. Затем запустите систему "1С:Предприятие". Мы воспользуемся "классическим" способом запуска программ — через кнопку Пуск (Start). Для этого нужно выбрать следующий пункт меню: Пуск | 1С:Предприятие 7.7 | 1С:Предприятие.

После поиска ключа защиты на экран будет выведен диалог Запуск 1С:Предприятия (рис. 1.3).



Рис. 1.3. Диалоговое окно Запуск 1С:Предприятия

Исследуем появившееся окно. Область A определяет режим запуска программы. Область B используется в сетевой версии для запуска "1С:Предприятия" в монопольном режиме. Область C позволяет выбрать необходимую информационную базу, с которой пользователь собирается работать в данном сеансе. Область D — область кнопок, которые управляют окном запуска программы.

Существуют четыре режима работы программы "1С:Предприятие".

- □ 1С:Предприятие основной режим работы с программой. В этом режиме работают пользователи системы, вводя первичную информацию и формируя необходимые отчеты. Работать в нем можно только со сформированной настройкой. Изменять функциональные возможности настройки в данном режиме можно только в ограниченных пределах;
- Конфигуратор этот режим предназначен для конфигурирования информационной базы, придания необходимых свойств и функциональных возможностей рабочей настройке. Также в данном режиме осуществляется сервисное обслуживание (сохранение/восстановление данных, настройка списка пользователей, тестирование и исправление информационной базы и т. д.);
- □ Отладчик этот режим служит для облегчения разработки конфигурации. Имеет следующие возможности:
  - пошаговое выполнение модулей конфигурации;
  - расстановка точек останова;
  - прерывание и продолжение выполнения модуля;
  - вычисление выражений для анализа состояния переменных

ит.д;

- Монитор в этом режиме можно просматривать список активных пользователей, тех которые в данный момент работают с информационной базой. Кроме того, монитор позволяет анализировать журнал регистрации действий, выполняемых пользователями за любые периоды времени (историю работы пользователей);
- Монопольно в сетевой версии программы предполагается вариант работы с одной информационной базой сразу несколькими пользователями (сетевая версия собственно для этого и предназначена). Но некоторые действия с информационной базой невозможно выполнить, если с ней работают несколько пользователей. Для такого рода действий и предназначена опция Монопольно (опция действительна только для режима 1С:Предприятие). При работе в Конфигураторе вы автоматически переходите в монопольный режим. Если вы вошли в программу монопольно, то с этой информационной базой уже никто не сможет больше работать, пока вы не выйдете из нее. В локальной версии программы этой опции нет.

Область управляющих кнопок:

- □ **ОК** запускает программу "1С:Предприятие" в выбранном режиме для работы с выбранной информационной базой;
- **Отмена** отменяет запуск программы;

- **Изменить** изменяет информацию о выбранной информационной базе;
- □ Добавить добавляет новую информационную базу. В поле Название нужно ввести имя информационной базы. В поле Путь ввести (или выбрать) путь к конфигурации. Путь должен быть таким, чтобы в выбранном каталоге обязательно оказался файл настройки 1сv7.md;
- □ Удалить удаляет информационную базу. Причем удаление производится не с самого диска, а только из списка доступных информационных баз. При желании вы сможете восстановить эту базу кнопкой Добавить.

Программа "1С:Предприятие" позволяет работать с большим количеством информационных баз. Это могут быть базы одного предприятия за разные периоды или же текущие (рабочие) базы разных предприятий. Например, если ваше предприятие занимается аудиторской деятельностью, то у вас информационных баз может быть очень много. При запуске программы вы, как пользователь системы, должны выбрать нужную вам информационную базу по ее названию, выделить ее щелчком левой кнопки мыши и запустить, щелкнув кнопку **ОК** (или дважды щелкнув левой кнопкой мыши по названию базы) На рис. 1.4 выбрана информационная база **ИнтернетПоддержка**, внизу экрана указан путь к ней. В этом каталоге лежит вся информация, касающаяся данной базы.

🌌 Запуск 1С:Преді	
В режиме:	
🛩 1С:Предприятие	
Информационные Б	
ВостокЦ Гаап_З	
ИмПродБух ИмПродТорг	
Институт	
Интернет Годдержк Информационная Б ИП_Торговля	
Гланцелярия	
E:\StaryDisk\Tc77\Ir	

Рис. 1.4. Информационные базы "1С:Предприятия"

Для того чтобы создать новую информационную базу, необходимо нажать кнопку **Добавить** и в появившемся окне задать название и путь к каталогу, который вы уже предварительно создали (рис. 1.5). Название информационной базы может быть произвольным. Лучше, если оно будет соотносящимся

по значению. По умолчанию же информационной базе присваивается имя, состоящее из слов "Информационная база" и номера базы.



Рис. 1.5. Регистрация информационной базы

После регистрации вам необходимо запустить Конфигуратор, поскольку ваш каталог еще пуст. Для этого нажмите кнопку **ОК**. После этого система предложит выбрать формат хранения и *кодовую страницу* (рис. 1.6).

При создании новой конфигурации необходимо выбрать ту кодовую страницу, которая должна совпадать с кодовой страницей Windows, и которая будет использоваться при запуске конфигурации. Если эти кодовые страницы не совпадают, то работа "1С:Предприятия" невозможна. Рекомендуется в диалоге, который изображен на рис. 1.6, оставить кодовую страницу, предложенную системой, так как это значение извлечено из Windows, и нажать кнопку **ОК**. Если будет выставлен выбор кодовой страницы и порядок сортировки, отличный от установленных в системе на текущий момент, то корректное функционирование "1С:Предприятия" станет невозможно. Скорее всего, будет выдано сообщение "Порядок сортировки строк, установленный для базы данных, отличается от системного", после чего работа системы "1С:Предприятия" будет завершена. В этом случае следует изменить либо установки операционной системы, либо изменить кодовую страницу информационной базы.



Рис. 1.6. Выбор кодовой страницы

#### Замечание

Изменение кодовой страницы можно произвести в Конфигураторе (см. следующий абзац), вызвав меню Администрирование | Кодовая страница таблиц ИБ.

Следует помнить, что система "1С:Предприятие" не приспособлена для работы с несколькими кодовыми страницами. Поэтому в процессе работы с одной информационной базой недопустимо переключение национальных настроек операционной системы, приводящее к смене кодовых страниц.

После выбора кодовой страницы откроется окно **Конфигуратор**. Оно похоже на многие другие программы Windows. Здесь есть меню, панели инструментов, рабочая область и строка состояния (рис. 1.7). Основным окном, с которым вам придется иметь дело на протяжении всего сеанса работы с Конфигуратором — это окно **Конфигурация**. Его можно открыть используя пункт меню **Конфигурация** | **Открыть конфигурацию** либо нажав кнопку панели инструментов, которая выполняет те же функции, что и пункт меню.

В этом окне имеются три вкладки (внизу, под окном): Метаданные, Интерфейсы, Права. Это три взаимосвязанных части, которые вместе и образуют конфигурацию. Вкладки позволяют работать с названными частями создаваемой конфигурации.



**Рис. 1.7.** Окно конфигуратора и группы метаданных конфигурации

#### 1.6. Объекты метаданных системы "1С:Предприятие"

В окне Конфигуратора размещается стандартное Windows-дерево *метадан*ных. Метаданные — это ветви дерева: Константы, Справочники, Документы и т. д. Само понятие *метаданные*, т. е. данные о данных, документация определяет как совокупность объектов метаданных. Сам объект метаданных это некоторое формальное описание группы понятий предметной области со сходными характеристиками и одинаковым предназначением. Расшифруем эти определения на примере такого объекта метаданных, как справочник. Допустим, необходимо создать справочник "Товары". Какие данные должен содержать этот справочник? Как минимум:

- □ запись, куда вносится наименование товара; причем надо указать, что наименование товара пишется символами (буквами, цифрами), и сколько символов отводится для этой записи;
- □ запись, куда вносится приходная цена товара; надо указать, что приходная цена — это числа с запятой, разделяющей целую и дробную части, и обозначить, сколько цифр отводится для целой части, сколько — для дробной;

🗖 запись, куда вносится расходная цена товара, с теми же характеристиками.

Вот в конфигураторе мы и вносим подобные данные, описывающие конкретный справочник — объект метаданных. А конфигуратор в ветке дерева метаданных "Справочники" содержит все необходимые средства для подобного описания.

Те объекты метаданных, которые мы визуально можем увидеть в дереве вкладки **Метаданные** окна **Конфигурация**, относятся к одному из *основных видов метаданных*.

Такими основными видами являются: Константы, Справочники, Документы, Журналы документов. Перечисления, Отчеты и Обработки. Все перечисленные виды относятся к так называемым *базовым объектам* системы. Эти виды объектов характеризуются тем, что доступны вне зависимости от того, какую из компонент системы "1С:Предприятия" вы приобрели. Заметим, что есть объекты метаданных, которые в дереве не видны и, следовательно, ни к какому основному виду метаданных не относятся.

Имеются также объекты метаданных, которые вы можете использовать только при наличии соответствующей компоненты. Это те, которые располагаются в дереве ниже базовых объектов. На рис. 1.7 указано, к каким компонентам какие объекты метаданных относятся. Коротко опишем основные объекты метаданных:

- Константы постоянные (условно-постоянные) величины, хранят информацию, которая не изменяется или изменяется достаточно редко: название организации, ее почтовый адрес и т. д. Базовый объект;
- Справочники списки однородных элементов данных. Используются для хранения нормативно-справочной информации. Базовый объект;
- Документы служат для ввода информации о совершенных хозяйственных операциях. Базовый объект;
- Журналы списки объектов данных типа "Документ". Служат для работы с документами. Базовый объект;

- Перечисления списки значений, задаваемые на этапе конфигурирования. Базовый объект;
- Отчеты средство получения выходной информации. Источником данных для построения отчетов служат документы, справочники и регистры, также используется информация, хранящаяся в константах. Базовый объект;
- Обработка используются для выполнения различных действий над информационной базой. Базовый объект;
- Регистры средства накопления оперативной информации о наличии и движении средств. Компонента "Оперативный расчет";
- □ Планы счетов списки объектов данных типа "Бухгалтерский счет" учетных регистров, по которым будет выполняться группировка средств при работе в системе "1С:Предприятие". Компонента "Бухгалтерский учет";
- Виды субконто списки объектов аналитического учета. Используется для организации аналитического учета средств предприятия в совокупности с бухгалтерскими счетами. Компонента "Бухгалтерский учет";
- Операция средство отражения в бухгалтерском учете хозяйственных операций, происходящих на предприятии. Это совокупности бухгалтерских проводок, выполняющие изменения в остатках и оборотах по бухгалтерским счетам. Компонента "Бухгалтерский учет";
- □ Проводка предназначена для отражения в бухгалтерских итогах результатов хозяйственных операций. Компонента "Бухгалтерский учет";
- Журналы расчетов средство проведения периодических расчетов и хранения результатов расчета прошлых периодов. Каждый журнал связан с одним из справочников, который называется справочником объектов расчета. Каждая строка журнала — единичный акт расчета, который производится для определенного объекта по определенному алгоритму, имеет протяженность во времени и результат компонента "Расчет";
- Виды расчетов алгоритмы, по которым могут проводиться расчеты записей журналов расчета. Компонента "Расчет";
- □ Группы расчетов средство логического объединения видов расчета по тем или иным признакам. Компонента "Расчет";
- Календари средства построения графиков работы предприятия, внешних организаций или отдельных категорий работников. Компонента "Расчет".

Естественно, что наборы объектов метаданных будут изменяться в зависимости от того, какие компоненты установлены. Только базовые объекты будут оставаться при любой компонентной структуре "1С:Предприятия". На рис. 1.8 показано изменение дерева метаданных при отсутствии компоненты "Расчет".



Рис. 1.8. Изменение дерева метаданных при отсутствии компоненты "Расчет"

Теперь сохраните зарегистрированную конфигурацию. Для этого зайдите в меню **Файл** | **Сохранить** (или воспользуйтесь комбинацией клавиш <Ctrl>+<S>). На экране появится сообщение "1С:Предприятия": "При сохранении метаданных будет произведен анализ сделанных изменений и, в случае необходимости, реорганизация информационной базы, что может потребовать достаточно продолжительного интервала времени! Выполнить сохранение метаданных?" Ответьте Да. Затем появится сообщение конфигуратора: "Отсутствует старый словарь данных! Продолжить?" Снова ответьте Да. Появится окно **Реорганизация информации** как на рис. 1.9. Нажмите кнопку **Принять**. Наконец будет выведено сообщение "Реорганизация информации закончена!" Нажмите кнопку **ОК**. Конфигурация сохранена, можно закрывать приложение. Об описанном диалоге поговорим в *главе 2*.



Рис. 1.9. Диалоговое окно Реорганизация информации



## Инструменты для разработки

Как указывалось в *разд. 1.5*, реализация необходимых для решения поставленной задачи алгоритмов производится с помощью конфигурирования. Другими словами, для того чтобы настроить технологическую платформу "1С:Предприятие" на решение необходимой задачи, нужно:

- определить структуру метаданных, используемых для решения данной задачи;
- 🗖 описать алгоритмы обработки данных, вводимых пользователем в систему.

Для этого используется Конфигуратор — инструмент для работы программистанастройщика в "1С:Предприятие". Характерная черта Конфигуратора максимальная приспособленность для конструирования конфигурации в области учета. Это значительно облегчает работу программиста, делает ее производительней. Есть и минусы — узкая направленность ограничивает инструменты и функциональность системы. Задачи, при решении которых необходим анализ информации не в смысле бухгалтерского или торгового учета, требует уже значительных усилий со стороны программиста.

## 2.1. Конфигуратор

Запустим "1С:Предприятие" в режиме Конфигуратора. Это можно сделать двумя способами:

- в окне регистрации информационных баз выбрать этот режим и открыть окно "Конфигурация";
- в панели инструментов на рабочем столе Windows нажать кнопку Пуск (Start), в меню выбрать Программы | 1С Предприятие 7.7 | Конфигуратор. В этом случае в окне регистрации информационных баз в списке режимов будет сразу отмечен режим Конфигуратор.

Запустите учебную конфигурацию.

Откройте окно Конфигурация. Рассмотрим меню конфигуратора. Это полезно сделать для повышения эффективности дальнейшей работы. Возможно, не все команды меню на данный момент понятны. Кроме того, в процессе работы с конфигурацией при открытии разных окон состав меню может меняться. Далее разберем действия, вызываемые меню, более подробно.

Команды, выполняемые с помощью меню Конфигуратора, приводятся в *Приложении 1*. Горизонтальное меню конфигуратора (рис. 2.1) — основное меню и называется "меню первого уровня". Выпадающее при выборе пунктов горизонтального меню вертикальное, называется "меню второго уровня".



Рис. 2.1. Меню Конфигуратор | Конфигурация

Панели инструментов вызываются при помощи меню **Сервис | Панели инструментов**, они дублируют меню, позволяя более оперативно обращаться к часто используемым командам. По умолчанию Конфигуратор имеет восемь панелей инструментов, расположенных в диалоговом окне **Панели инструментов** на одноименной вкладке. Для того чтобы нужные панели инструментов появились в окне конфигуратора, их необходимо отметить флажками. Как видно из рис. 2.2, в окно Конфигуратора выведены панели Стандартная, Конфигурация, Администрирование, Конструкторы. Состав кнопок на панели инструментов можно менять, точно так же, как это делается в приложениях Microsoft Office. Для этого в диалоговом окне необходимо переключиться на вкладку Модификация.



Рис. 2.2. Диалоговое окно Панели инструментов

Список **Категории** соответствует названиям панелей инструментов. Справа от списка находятся изображения кнопок. Щелкнув мышью по изображению кнопки, в поле **Описание** можно прочитать описание функции кнопки. Нажав левую кнопку мыши, выделенную кнопку можно перетащить на любую панель управления. Перейдя на вкладку **Дополнительные**, можно определить положение панелей инструментов в окне конфигуратора.

Если подвести указатель мыши к кнопке на панели инструментов, то появится подсказка, определяющая функции кнопки. Одновременно в строке состояния окна Конфигуратора функция кнопки описывается более подробно.

## 2.2. Создание объектов метаданных

О дереве метаданных уже говорилось в *разд. 1.6.* Как и все стандартные деревья Windows, дерево метаданных перевернуто. "Корень" дерева находится вверху, ствол и ветви уходят вниз. Корнем дерева метаданных являются метаданные Задача. Это название всей конфигурации. Главные ветви — общие названия объектов метаданных — Константы, Справочники и т. п. Каждая главная ветвь может иметь дополнительные ответвления — созданные в процессе конфигурирования конкретные объекты метаданных, например, справочник "Номенклатура" на ветви Справочники или документ "Расходная накладная" на ветви Документы.

🚰 Конфигуратор - Бух и опер учет, расчет - [Конфигурация]	_ 8 ×
ষ্ট্রী Файл Действия Конфигурация Конструкторы Администрирование Сервис Окна Помощь	_ 8 ×
12 G 🖬 X h fi (G A ) 🖬 ⊅ ⊂ 🛤 Basalar 💽 🙀 🤽 🧏	
在前近外的站在台湾的有些。 电电子 医甲酮 医丁酮 医骨骨 医	
22 後 22   ↓ ↑ 44   目	
🖻 🔞 Учебная Компл учет	-
C I De Dominica Da Carla de Ca	
РФ Сотридники	
н 🖏 Контрагенты	
в 🔊 Основания	
в 🔊 Номенклатура	
- 🔊 Подразделения	
🖻 🎒 Графики	
🛱 🚔 Документы	
🕀 🛥 Общие реквизиты	
— Ш Нумераторы	
- В Последовательности	
П в Амортизация	
ны СлисаниеОС	
🗄 📶 Журналы документов	
🗄 🚍 Графы отбора	
— 🗐 УчетМатериалов	
- 🗎 Kacca	-
🔀 Метаданные 🔐 Интерфейсы 🕅 🖏 Права	
та та конфигурация	
Для получения подсказки нажинте F1	Документ
🖞 Пуск 🥥 🖓 🕙 » 🔐 Книга 🎲 Глава2 🖉 4 Глава 2 - М 🕅 Документ 1 📰 Раілt Shop Pro 🛛 🥰 Конфигурат 🕎 🖓 🦿 « 5	<u>34</u> 🌾 20:32

Рис. 2.3. Дерево объектов метаданных

В окне конфигурации можно раскрывать списки объектов. Объекты можно добавлять, удалять, редактировать, изменять их порядок, сортировать, составлять для них описание. Для того чтобы раскрыть списки объектов, надо щелкнуть мышью по знаку "+", расположенному слева от наименования ветви метаданных (рис. 2.3). Знак "+" появляется, если в ветви имеются созданные конкретные объекты метаданных, либо присутствует с начала создания конфигурации в таких ветвях, как Документы, Журналы документов, Планы счетов, Операция, Виды расчетов. У этих ветвей метаданных имеются ответвления (например, Графы отбора у Журнала документов) с момента создания конфигурации. Назначение некоторых из этих ответвлений мы обсудим позже.

Создавать объекты метаданных можно с помощью мастеров-конструкторов или вручную. Самый простой способ создания объекта метаданных — с помощью конструктора. Выберите в меню конфигуратора пункт Конструкторы, и выпадающее вертикальное меню покажет, какие объекты метаданных можно построить с помощью мастера. С помощью панели инструментов Конструкторы также вызывается мастер построения объекта метаданных (рис. 2.4).

🥳 Конфигуратор - Бух и опер учет, расчет - [Коно 📲 Файл Действия Конфигурация Конструкторы	игурация] Адиничисточорвание Сервис Окна Помощь	X X
ng Ilar Garde	Brasillox V & 2 ?	
Винак Конплунет     Винак Конплунет     Винак Конплунет     Сотранени     Основник Средства     Основник Средства     Основник     Основник     Одразделения     Одразделения     Одразделения     Одразделения     Одразделения     Одразделения     Одразделения     Винак Сотранения     Основания     Основания     Основания     Основания     Одразделения     Одерания     Одерания     Одразделения     Одраздел	Конструктор Справочника     ▼       Идентичикатор:     ☐065801       Снескник:	×
UverMarepuanos		-
🔣 Метаданные 🔐 Интерфейсы 🖄 Права		
ад получения полоказки наумите E1		NUM CTATYC: 5.4 ADDOMENT
	asa2 🔄 4E nasa 2 - Mi 🔄 Answeett 1 🚟 Paint Shon Pro	
	нова на полоде нист на макунских на макунских нарина и макунских на макунских	

Рис. 2.4. Пример мастера построения метаданных

Для того чтобы создать объект метаданных с помощью Конфигуратора, необходимо в режиме конструктора или вручную описать, по крайней мере, основные свойства этого объекта: идентификатор, синоним, комментарий.

Идентификатор — это название объекта метаданных для программиста и для технологической платформы "1С:Предприятие". Как правило, идентификатором пользуются при обращении к объекту на встроенном языке системы. Идентификатор должен начинаться с буквы, может содержать буквы, цифры, знаки подчеркивания и не может содержать пробелы и спецзнаки, а также быть длиннее 128 символов.

*Синоним* — это другое название идентификатора, которое пользователь будет видеть в окнах "1С:Предприятия" вместо идентификатора. Синоним может содержать пробелы.

*Комментарий* — служит для расшифровки идентификатора объекта и выдается в различных диалогах системы вслед за идентификатором объекта.

Как описать эти свойства, разберем в следующем разделе.

## 2.3. Свойства конфигурации

Щелкните правой кнопкой мыши по корню дерева объектов метаданных Задача. Появится контекстное меню (рис. 2.5). Выберите пункт **Свойства**. Этот пункт вызывает палитру свойств объекта метаданных.

#### Замечание

Палитру свойств можно вызвать, дважды щелкнув левой кнопкой мыши по объекту метаданных.

Палитра свойств любого объекта метаданных (рис. 2.6) всегда имеет как минимум одну вкладку **Общие**, на которой расположены поля для ввода идентификатора, синонима и комментария. Остальные вкладки могут отсутствовать или их количество и названия могут варьироваться в зависимости от вида объекта метаданных.

Заполним соответствующие поля следуюшим образом:

- **П Идентификатор** Учебная Компл учет;
- **Синоним** Бух и опер учет, расчет;

#### Внимание!

Синоним записан уже с пробелами.

□ Комментарий — Бухгалтерский и оперативный учет, расчет для самоучителя. Нажмем на палитре свойств кнопку Обновить. Название корня дерева метаданных сразу же стало соответствовать введенному идентификатору.



Рис. 2.5. Контекстное меню объекта метаданных

	Учебная Компл уч	ет
	Общие Задача	Автој
	Идентификатор:	Учеб
	Синоним:	Бух и
	Комментарий:	Бухг
l		
I		

Рис. 2.6. Палитра свойств объекта метаданных

Вкладка Задача позволяет выполнить следующие действия:

выбрать основной язык работы системы;

выбрать режим округления. Если режим округления выбран "Округлять 1,5 до 1", то при округлении числа до целого значения 0,5 включительно будут отбрасываться, а свыше 0,5 — округляться до целого. Например, сумма 1 рубль 50 копеек будет округлена до 1 рубля, а сумма 1 рубль 51 копейка — до 2-х рублей. Если выбрана установка "Округлять 1,5 до 2", то значения до 0,5 будут отбрасываться, а свыше 0,5 включительно округляться до целого. Например, сумма 1 рубль 49 копеек будет округлена до 1 рубля, а сумма 1 рубль 50 копеек — до 2-х рублей.

#### Замечание

Правило округления, установленное в свойствах конфигурации, при вычислении конкретного выражения можно изменить, используя соответствующую функцию встроенного языка системы "1С:Предприятие".

#### Предупреждение

Флажок Разрешить непосредственное удаление объектов позволяет удалять записи из информационной базы без контроля ссылочной целостности, т. е. без проверки, используются ли эти записи где-либо в базе. Устанавливать этот флажок можно только опытным специалистам по конфигурированию.

Кнопка Сменить пароль позволяет установить или сменить пароль на входе в конфигурацию. При нажатии этой кнопки появится поле для ввода пароля. Пароль представляет собой произвольную строку длиной не более 10 символов, состоящую из букв и (или) цифр. Пароль не должен содержать пробелов и специальных символов. При вводе пароль сразу же маскируется — отображается знаками "\*". После ввода пароля нажмите кнопку **ОК**, и программа запросит пароль еще раз. Повторите ввод пароля и для подтверждения нажмите кнопку **ОК**. Поле для ввода пароля будет закрыто и назначен новый пароль. Если пароль уже существовал, он будет удален и заменен новым.

Если вы передумали устанавливать пароль, нажмите кнопку **Отмена**. Если пароль установлен, конфигурацию невозможно открыть и редактировать без ввода этого пароля. Однако если пароль не известен или забыт, то "взломать" его может только квалифицированный специалист. В нашем примере пароль устанавливаться не будет.

Вкладка **Автор** служит для ввода сведений об авторе конфигурации. Непосредственно в палитре свойств показывается рисунок логотипа (если он есть). Щелкните кнопку **Изменить** и заполните диалоговое окно, как на рис. 2.7. Кнопки **Выбрать** вызывают диалоговое окно для выбора файла рисунков логотипа и заставки. Заставка появится при загрузке "1С:Предприятия". Логотип будет виден в окне "О программе". Сведения, введенные в этот диалог, можно защитить паролем. Ввод пароля осуществляется при нажатии кнопки **Пароль** аналогично вводу пароля на конфигурацию.

Учебная Компл у	чет 🗵
Общие Задача	Автор Заставка 🦿 🖓 – д
Автор конфигура	ции:
гурация © Дуб:	Автор конфигурации 🛛 🛛 🗙
Учебная конфиі бухгалтерский у	Краткая информация:
расчет.	Учебная конфигурация © Дубянский В.М., Скобликова Л. К.
	Развернутая информация:
Изменить	Учебная конфигурация, компоненты: бухгалтерский учет, оперативный учет, расчет.
	Поготип
	Очистить Выбрать Выбрать Для установки логотипа выберите картинку размером 64x64 пиксела в 256 цветах.
	_ Заставка
	Спредпроизтине Версия 7.7 Учебная Конфигурация
	Очистить Выбрать
	Пароль ОК Отмена

Рис. 2.7. Сведения об авторе конфигурации

Вкладка Заставка практически повторяет вкладку Автор, только вместо логотипа выводится рисунок заставки.

Кнопка с знаком "?" вызывает справочник помощи "1С:Предприятия", а кнопка со значком булавки позволяет оставить палитру свойств на экране при смене объекта метаданных. Если эта кнопка не нажата, то при работе с другим объектом метаданных палитра свойств закроется.
## 2.4. Сохранение базы данных

Один из главных факторов стабильной работы в "1С:Предприятии" версии 7.7 — регулярное сохранение информационной базы данных. Разрушение базы может происходить по разным причинам: поломка компьютера, вирусы, внезапное отключение питания, сбои в сети, да и просто непонятные причины. Разрушенную базу очень трудно восстановить, а часто и вообще невозможно. Пропадет работа многих специалистов, будет нарушено функционирование предприятия. Если же база данных сохраняется каждый день, как рекомендуют опытные пользователи, то, максимум, вам придется восстанавливать учет за один день. В самых трудных случаях вы можете просто переустановить программу на другом компьютере (ограничений на количество инсталляций "1С:Предприятие" не имеет), восстановить из сохраненной копии базу данных и продолжить работу.

Желательно иметь несколько архивных копий. База должна ежедневно копироваться на жесткий диск компьютера (или сервера) и на независимый носитель — дискеты, съемные диски, компакт-диски и пр. Самый простой способ сохранения базы — создание обычной копии всей папки. Это удобно при наличии места на носителях и относительно небольшом объеме базы. Каталог, в котором хранится информационная база, можно увидеть в окне **О** программе — меню Помощь | **О программе**.

"1С:Предприятие" имеет собственные встроенные средства архивирования информационных баз. Перед сохранением данных необходимо закрыть все открытые окна в Конфигураторе и закрыть "1С:Предприятие" в пользовательском режиме. Войти в меню Администрирование. В окне откроется список опций, в том числе и четыре пункта, относящиеся к сохранению данных, которые условно можно поделить на две группы в соответствии с режимом: Сохранить-Восстановить (Сохранить данные, Восстановить данные) и Выгрузить-Загрузить (Выгрузить данные, Загрузить данные) (рис. 2.8).

Оба режима предназначены для создания резервной копии информационной базы, восстановления информационной базы из резервной копии, переноса информационной базы на другой компьютер.

### Замечание

Режим Сохранить-Восстановить обеспечивает сохранение данных только в случаях формата DBF. Режим Выгрузить-Загрузить работает в формате DBF и MS SQL Server.

При выборе пункта меню **Сохранить данные** откроется диалог **Сохранение в архив**, который содержит список сохраняемых файлов и кнопки управления этим списком (рис. 2.9).



Рис. 2.8. Опции меню для сохранения или восстановления данных



Рис. 2.9. Диалоговое окно Сохранение данных

Изначально этот список содержит файлы, жизненно важные для функционирования системы "1С:Предприятие":

- \*.DBF файлы данных;
- □ 1CV7.MD описание конфигурации задачи;
- □ 1CV7.DD описание структуры файлов данных;
- □ 1CV7.SPL словарь, который используется для представления чисел, наименований денежных единиц и дат прописью.

Список сохраняемых файлов можно расширить, включив в него любые другие файлы.

Для добавления файла в список следует нажать кнопку **Добавить**, тогда на экран будет выдан диалог со списком файлов информационной базы.

Чтобы добавить файл в список сохраняемых файлов, необходимо выделить его клавишами или мышью и нажать кнопку **ОК**. Нажав кнопку **Отмена**, можно отказаться от выбора файла. Для исключения файла или группы файлов из списка необходимо выделить наименование файла в списке и нажать кнопку **Удалить**. Из списка нельзя удалить системные файлы и файлы данных — при выборе таких файлов кнопка **Удалить** станет недоступной.

После нажатия кнопки **Удалить** на экран будет выдан запрос для подтверждения удаления выбранных файлов. Нажатие кнопки **ОК** удаляет выбранные файлы, кнопка **Отмена** возвращает в режим **Сохранение в архив**.

Список сохраняемых файлов запоминается при выходе из режима **Сохранение в архив**. На экран выдается запрос "Сохранить список сохраняемых файлов?", при ответе **Да** список будет сохранен.

Обратите внимание на поле **Сохранять в**. При нажатии кнопки cnpaвa от этого поля откроется стандартное диалоговое окно Windows, в котором надо указать путь к месту (носителю), где будет сохранена архивируемая база. При этом если информационная база должна сохраняться на дискетах, программа архивации будет автоматически разбивать файл архива на части, соответствующие емкости дискет, и запрашивать новые дискеты по мере заполнения предыдущих.

В случае необходимости из архивной копии можно восстановить данные. Для этого выберите пункт **Восстановить данные** в меню **Администрирование** главного меню конфигуратора (рис. 2.10).

Если в открывшемся диалоге щелкнуть мышью кнопку ...., находящуюся справа от поля **Из архива**, то откроется стандартное диалоговое окно Windows, в котором нужно указать путь к файлу (носителю) с архивом информационной базы. После того как указано имя файла резервной копии, для запуска процесса восстановления следует нажать кнопку **Восстановить**.

Восстановление данных			
Восстанавливать			
	Из архива:	1Cv7.zip	
	Дата архива		

Рис. 2.10. Диалоговое окно Восстановление данных

Если происходит восстановление из резервной копии на дискетах, следует по запросам программы вставлять дискеты, содержащие части копии, в дисковод. Если дискеты вставляются в неверном порядке, программа выдаст соответствующее сообщение. При восстановлении информации из резервной копии на жестком диске или компакт-диске программа сразу начнет распаковку и запись в каталог информационной базы.

### Предупреждение

Если в каталоге текущей информационной базы существуют какие-либо файлы системы "1С:Предприятие", они будут полностью заменены информацией из резервной копии.

При выгрузке данных (меню Администрирование | Выгрузить данные) сразу появится диалог Выгрузка данных, в котором нужно указать путь к месту сохранения архивного файла и название самого файла. В отличие от предыдущего режима, никаких окон для настройки типов файлов сохраняемой информационной базы не предусмотрено. В этом режиме архивируются только жизненно важные для системы файлы.

Загрузка данных	
Загрузить из файла:	E:\Ca
Пароль ZIP:	

Рис. 2.11. Диалоговое окно Загрузка данных

Загрузка сохраненных данных производится при вызове меню Администрирование | Загрузить данные. В диалоговом окне (рис. 2.11) необходимо указать путь к архивному файлу и нажать кнопку ОК. После этого появится сообщение: "При загрузке данных все существующие данные будут уничто-

жены. Продолжить выполнение операции?" Если вы твердо решили загружать архивную копию, отвечаете Да. Если есть сомнения, ответьте **Нет** и все снова проверьте.

Основное отличие двух режимов архивирования информационной базы заключается в том, что режим **Сохранение**—Восстановление сохраняет информационную базу на уровне файлов, а режим Выгрузка—Загрузка — на уровне объектов метаданных. Это важный нюанс. При пользовании первым режимом восстановленная информационная база будет содержать все физические и логические ошибки исходной базы (за исключением индексных файлов). При использовании второго режима произойдет восстановление только корректных, или подлежащих автоматическому пересчету и исправлению данных. В целом режим Выгрузить—Загрузить является более надежным, а режим Сохранить—Восстановить — более удобным.

В заключение раздела замечу, что архивировать базу данных можно и с помощью любых внешних архиваторов (ZIP, RAR и т. п.), и с помощью специализированных программ (например, "Хранитель" фирмы "Гендальф", г. Ростов-на-Дону, www.gendalf.ru).

### Замечание

Неверная загрузка измененной конфигурации может повлечь разрушение базы, которое проявится только через несколько дней. В этих случаях наличие архивной копии базы за последний день не спасет положение. Исходя из опыта работы, можно рекомендовать сохранять копии информационных баз данных за последние пять-семь дней.

### 2.5. Встроенный язык системы

Встроенный язык системы "1С:Предприятие" является предметно-ориентированным, объектным языком программирования. Язык предназначен для описания алгоритмов функционирования прикладной задачи. Для облегчения написания текстов на этом языке в синтаксисе его реализована практически полная *двуязычность*. Одни и те же системные конструкции и зарезервированные слова языка могут быть записаны как по-русски, так и поанглийски, причем в одном тексте можно использовать как русские, так и английские синонимы. В документации к системе "1С:Предприятие" версии 7.7 описание встроенного языка приводится с русским и английским синтаксисом.

Язык обладает достаточным набором типов данных, которые вы можете использовать при обработке. Однако средствами самого языка вы не можете создавать новые типы данных. Эти типы вы будете создавать, используя визуальные средства настройки конфигуратора. В отличие от стандартных объектно-ориентированных языков метаданные задаются в Конфигураторе, и их виды жестко закреплены.

Процедуры и функции на встроенном языке пишутся в программных модулях. Место самого программного модуля в конфигурации — в тех точках, где требуется специфическая реакция системы на действия пользователя, определяемые условиями задачи. Например, при обработке данных, внесенных в документ (вычисление НДС), при проведении документа (ведь бухгалтерские проводки в той или иной степени специфичны для предприятия). Соответственно, у части метаданных (но не у всех) существуют следующие модули:

- модули формы объекта метаданных, предназначенные для обработки данных, внесенных в экранную форму. Выполняется при открытии (вызове) формы;
- модули проведения (объект метаданных Документ), предназначенные для проведения, отмены проведения и подобных действий;
- глобальный модуль, который находится в корневом разделе конфигурации и запускается при старте всей прикладной задачи. Он определяет глобальный контекст всей задачи (связывает работу частей конфигурации в единое целое).

Все модули имеют одинаковую структуру. Каждый модуль может состоять из трех разделов:

- раздел переменных в языке "1С:Предприятия" переменные не являются типом метаданных, они содержат значения, которые могут изменяться во время исполнения программы (потому так и называются). Переменные существуют в оперативной памяти компьютера и исчезают (в отличие от констант) при окончании работы модуля (процедуры) или после окончания работы системы;
- раздел процедур и функций отдельные действия, выполняемые в модуле, оформляются, как процедуры или функции с именем. Отличия процедур от функций следующие: процедуры выполняют некоторую задачу, например, напечатать документ. Функции выполняют вычисления и передают системе результат, например, вычислить дату и передать ее для дальнейшей обработки в процедуру;
- раздел основной программы обычно здесь происходит инициализация каких-либо переменных, необходимая еще до первого вызова процедуры или функции.

Каждый из разделов не является обязательным, но если присутствует, то в том порядке, как это перечислено.

Когда, в соответствии с логикой работы, должен быть вызван модуль, система выполняет следующие действия:

- 1. Находит текст модуля в конфигурации.
- Выполняет его компиляцию. На этом этапе происходит преобразование текста модуля во внутренний формат данных (он недоступен пользователю), который позволяет системе быстро выполнять алгоритм, не интерпретируя его текст много раз.
- Если компиляция завершилась удачно, т. е. не было найдено синтаксических ошибок, то управление получает та часть модуля, которая называется основной программой. Ее назначение — выполнить инициализацию модуля.
- Если в процессе компиляции найдены ошибки, то модуль считается неработоспособным, и, скорее всего, ваша конфигурация нормально работать не будет. Система выдаст сообщение об ошибке.

Разберем примеры работы в программном модуле. Примеры можно просмотреть в учебной конфигурации. Начнем со стандартной программы приветствия.

Откройте конфигурацию, щелкните правой кнопкой мыши по наименованию конфигурации, в контекстном меню выберите пункт **Глобальный модуль** (рис. 2.12). Откроется пустое окно — это и есть место для написания программы в глобальном модуле.

В окне введем следующий текст:

#### Листинг 2.1

```
//раздел переменных
Перем привет;
//раздел процедур и функций
Процедура приветствие()
Предупреждение (привет);
КонецПроцедуры
//раздел инициализации
привет="Здравствуйте!"; //инициализация переменной
приветствие(); //вызов процедуры
```

Правила работы в окне написания программ такие же, как в любом текстовом редакторе.



Рис. 2.12. Переход к глобальному модулю в конфигурации

Разберем приведенную программу.

Двойным слэшем (//) обозначают комментарии к программе. *Комментарии* — это заметки программиста, не предназначенные для исполнения системой, а пометки для себя, чтобы было удобнее пользоваться программой. Слова, написанные после знака //, система не интерпретирует, как программу. Наличие комментариев очень важно, и их написанием не стоит пренебрегать, иначе можно запутаться в собственной (а, тем более, в чужой, если необходимо внести изменения) программе. В приведенном примере (листинг 1) комментариями описана классическая структура программного модуля.

В разделе переменных с помощью ключевого слова перем мы объявили переменную с произвольным идентификатором (привет). В разделе процедур и функций создали процедуру с произвольным именем приветствие (). Двойные скобки после имени процедуры или функции обязательны. В них может заключаться список параметров. В теле процедуры вызываем системную функцию Предупреждение с параметром привет — окно предупреждения.

В разделе инициализации переменной привет присваивается значение выражение "Здравствуйте!" (инициализация переменной) и вызывается процедура приветствие(). Почему требуется вызывать процедуру? Глобальный модуль автоматически выполняется при старте системы. Но откуда система узнает, какую процедуру нужно выполнять первой, второй и т. д.? Из раздела инициализации, где указан порядок вызова процедур.

### Примечание

Точка с запятой является символом разделения операторов (отдельных команд программы).

Однако некоторые конструкции, например, операторы Процедура... Конец-Процедуры, не должны оканчиваться символом ;. Опыт показывает, что у начинающих программистов большая доля синтаксических ошибок падает именно на точку с запятой: либо ее забывают ставить в конце оператора, либо ставят, где не положено. Следите за этим.

В меню Действия | Синтаксический контроль осуществляется проверка синтаксиса программы. Если вы написали все верно, в окно сообщений будет выведена строка: "Синтаксических ошибок не обнаружено!" Однако осуществляется только проверка синтаксиса программы, а правильность написания программы можете проконтролировать только вы сами. В панели инструментов Текстовый редактор найдите кнопку с таким же рисунком, как у команды меню Синтаксический контроль. В дальнейшем при проверке синтаксиса можно использовать ее.

### Примечание

Если панель инструментов отсутствует, ее можно вызвать, открыв меню **Сервис** и выбрав пункт **Панели инструментов**.

Теперь запустим "1С:Предприятие" в режиме исполнения. Результат работы программы показан на рис. 2.13.

Текст программного модуля лучше делать разноцветным. Это очень удобно, поскольку структура программы становится более наглядной. Кроме того, синтаксическая ошибка в написании оператора изменяет его цвет, что позволяет сразу увидеть все ошибки программы.

### Примечание

Настроить выделение цветом синтаксических конструкций можно, зайдя в меню Сервис | Параметры и выбрав вкладку Модули.



Рис. 2.13. Приветствие системы

Дополнительно исследовать нашу программу можно с помощью синтакспомощника. В процедуре приветствие() щелкните мышью по системной процедуре Предупреждение(), затем правой кнопкой вызовите контекстное меню и выберите пункт Поиск в синтакс-помощнике (рис. 2.14). В правом нижнем окне синтакс-помощника помещено подробное описание системной процедуры Предупреждение().

Попробуйте самостоятельно поработать с параметром Таймаут.

С помощью синтакс-помощника можно разобраться в работе процедур и функций. Однако этот инструмент не заменяет книгу "Описание встроенного языка", а только дополняет ее. Изменения во встроенном языке, происходящие от релиза к релизу, отражаются в синтакс-помощнике. Но примеры использования операторов языка вы можете найти только в книге.

### Примечание

В синтакс-помощник можно попасть также через меню Сервис | Синтакс-помощник.



Рис. 2.14. Синтакс-помошник процедуры Предупреждение

Помимо процедур, которые мы создаем самостоятельно, в языке "1С:Предприятие" существуют *предопределенные процедуры*. Они имеют свои, зарезервированные системой названия и предназначены для обработки определенных действий пользователя. Какие действия пользователя обрабатывает предопределенная процедура, можно понять из ее названия, например, процедура ПриУдаленииДокумента() или ПриНачалеРаботыСистемы(). Тело предопределенной процедуры программист пишет сам в соответствии с алгоритмом, который необходим для решения задачи. Например, в процедуре ПриНачалеРаботыСистемы() можно программным путем узнать права пользователя, можно проверить, какие компоненты системы "1С:Предприятие" установлены на компьютере пользователя и т. п. Виды предопределенных процедур можно посмотреть в синтакс-помощнике: Встроенный язык | Глоб. модуль | предопределенные проц. для глобального модуля или при открытии ветви, относящейся к определенному виду метаданных (рис. 2.15).

Напишем полезную процедуру для завершения работы "1С:Предприятия". Если вы закрываете программу, она спрашивает: "Вы действительно хотите

закончить работу?" При ответе Да программа закрывается, при ответе Нет работа продолжается. Действительно, если открыто несколько окон различных приложений одновременно, вы можете ошибке по закрыть "1С:Предприятие". Наша процедура подстрахует от подобной случайности. Она задаст алгоритм окончания работы всей системы "1С:Предприятие". Просмотрим в синтакс-помощнике предопределенные процедуры глобального модуля. Предопределенная процедура ПриЗавершении Работы Системы () нам подходит. Напишем текст процедуры, разместив ее в разделе процедур и функций после процедуры Приветствие () (рис. 2.16).



Рис. 2.15. Описание предопределенных процедур модуля формы документа в синтакс-помощнике

#### Листинг 2.2

Процедура ПриЗавершении Работы Системы ()

Если Вопрос("Работа программы закончена?", "Да+Нет")="Нет" Тогда

```
Статусвозврата(0);
```

КонецЕсли;

Конецпроцедуры

Несмотря на то, что текст процедуры состоит из трех строк, она достаточно сложна для начинающих. Разберем конструкции, появившиеся в этой процедуре.



Рис. 2.16. Произвольные и предопределенные процедуры

Подробное описание системной функции Вопрос() посмотрите в синтакспомощнике. В нашем случае эта функция выводит на экран окно с вопросом и кнопками Да и Нет возвращает ответ пользователя. Если вы нажмете кнопку Да, условие Если... Тогда... КонецЕсли; не будет выполнено, и программа завершит работу. Ведь функция Вопрос вернет в процедуру значение Да, а выполнение условия — если ответ на вопрос вернет значение нет. Если же вы нажмете кнопку Нет (Вопрос (...)=нет), условие будет выполнено, управление перейдет к следующей важной системной функции — СтатусВозврата(). Эта функция предназначена для использования только в предопределенных процедурах. Если параметр функции задан нулевым, действие предопределенной процедуры отменяется, и программа продолжит работу, что нам и требуется в этом случае. Подробнее о функции Статус-Возврата() смотрите в документации или в синтакс-помощнике.

## 2.6. Диалог при сохранении конфигурации

В случае прекращения работы (закрытия) или запуска пользовательского режима "1С:Предприятие" из конфигурации при любых изменениях в объектах метаданных конфигурации будет выведен набор диалоговых окон, о котором говорилось в конце *главы 1*. Предупреждение "При сохранении метаданных будет произведен анализ" актуально, если разработчик не уверен в правильности своих действий. На данном этапе изменения в Конфигураторе можно отменить.

### Предупреждение

Сохранение метаданных в действующих конфигурациях с большими объемами данных может действительно занять очень много времени (вплоть до суток и более). Например, при удалении из какого-либо документа флажка вида учета (см. главу 7).

В следующем диалоговом окне **Реорганизация информации** визуально перечисляются все изменения в метаданных, которые выполнил разработчик. На данном этапе еще можно отказаться от них (кнопка **Отмена**), либо согласиться с ними (кнопка **Принять**). После нажатия кнопки **Принять** конфигурация будет переписана с внесенными изменениями. Если изменения не касались базовых функций метаданных, а были "косметическими", то в окне **Реорганизация информации** появится сообщение "Изменения в метаданных не вызвали изменений в данных".

При первом сохранении новой конфигурации появляется сообщение конфигуратора: "Отсутствует старый словарь данных! Продолжить?" Как правило, отвечать нужно "Да". В этот момент происходит создание файла словаря данных, который содержит описание структуры хранения информационной базы на специальном внутреннем языке описаний в терминах баз данных. Практическое применение этого файла описано в *Приложении 1*.

Этот диалог направлен на то, чтобы любые изменения делались разработчиком осознанно, и не произошло случайного изменения конфигурации, что может повлечь за собой искажение и разрушение информационной базы.

Глава 3



# Пользовательский интерфейс и администрирование

На предприятии каждый сотрудник выполняет свою часть работы. Сотрудники обладают разной квалификацией. У них могут быть определенные ограничения на доступ к конфиденциальной информации, находящейся в базе "1С:Предприятия". Наконец, им должно быть удобно работать. Соответственно, определенные категории сотрудников могут иметь возможность работы с ограниченным количеством метаданных — документов, справочников и т. п. Например, кладовщик может заполнять расходные документы, но не имеет права их проводить — это делает бухгалтер. Кладовщик может не иметь доступа к документам, справочникам и отчетам, относящимся к начислению и выплате заработной платы. Бухгалтер, начисляющий заработную плату, может не иметь возможности работать с приходно-расходными документами и т. д. Определенные ограничения в доступе к различной информации могут налагаться и только из соображений удобства работы: конфигурации, применяющиеся на крупных предприятиях, могут иметь десятки документов, справочников и отчетов. Меню, вызывающее их, будет очень громоздко и неудобно. В "1С:Предприятии" решение подобных задач производится с помощью развитых средств построения пользовательского интерфейса, который включает в себя меню и панели инструментов, набор пользовательских прав и набор пользователей.

В данной главе мы изучим основы работы с меню, пользовательскими правами и списком пользователей. Продолжим изучение этой темы после создания конфигурации, в *главе 18*.

## 3.1. Индивидуальный пользовательский интерфейс

Средства Конфигуратора позволяют создать столько различных пользовательских интерфейсов, сколько нужно. Интерфейс является средством администрирования системы "1С:Предприятие" на уровне разработчика конфигурации. Именно здесь разработчик создает набор интерфейсов, которые регулируют возможность доступа, изменения и просмотра базы данных системы через различные виды метаданных.

Для того чтобы создать новый интерфейс, нужно в окне **Конфигурация** выбрать вкладку **Интерфейсы** и нажать кнопку **Новый элемент** панели инструментов или выбрать пункт меню **Действия** | **Новый интерфейс**. После чего программа создаст новый объект с именем Новый и откроет окно панели инструментов **Свойства пользовательского интерфейса** (рис. 3.1):



Рис. 3.1. Палитра свойств пользовательского интерфейса

- □ **Название** это произвольная строка текста. Желательно, чтобы название интерфейса было осмысленным и понятным пользователю;
- флажок Отключить меню "Операции" включают в тех случаях, когда пользователю должен быть доступен для использования только ограниченный набор объектов.

### Предупреждение

При создании интерфейсов конфигураций важно помнить, чтобы среди них был хотя бы один интерфейс, для которого флажок **Использовать при отключенной авторизации** был включен. Это даст пользователю возможность корректно начать работать на пустой информационной базе, т. е. когда нет еще ни одного пользователя.

Когда вы создаете новый интерфейс, то в нем нет ни меню, ни панелей инструментов. Это все еще предстоит выбрать, вызывая **Редактор меню** и **Редактор панелей инструментов**.



Рис. 3.2. Палитра свойств меню

Для вызова редактора меню необходимо на соответствующем названии интерфейса из списка (а у нас создан только один интерфейс — Новый1) нажать правую кнопку мыши и выбрать пункт выпадающего меню **Редактировать меню**. После чего откроется окно редактора меню, где меню будет иметь вид дерева. Вначале разработаем меню первого уровня, т. е. то, которое будет выводиться в горизонтальной строке, наряду с уже имеющимися по умолчанию (рис. 3.2).

Для меню первого уровня можно указать только название. Все остальные поля палитры свойств будут недоступны. Если в палитре свойств установить флажок Системная колонка, то название меню первого уровня необходимо будет выбирать из выпадающего списка, содержащего названия четырех системных колонок главного меню системы "1С:Предприятие". В нашем примере флажок Системная колонка не устанавливаем.



Рис. 3.3. Тест вновь созданного меню

Создадим элементы меню первого уровня:

- Справочники;
- 🗖 Документы;

🗖 Отчеты

и т. д.

Теперь осталось проверить, как будет работать наше меню. Для этого нужно щелкнуть кнопку **Проверить меню** панели инструментов редактора меню или войти в меню конфигуратора **Действия** | **Тест**. Тогда редактор меню откроет так называемое **Тестовое окно** (рис. 3.3). В этом окне можно просматривать внешний вид меню, но нельзя его использовать.

В поле палитры свойств элемента меню "Тип", кроме типа "Команда" можно выбрать значения "Разделитель" и "Подменю". Эти типы служат для оптимизации внешнего вида меню и удобства в работе с ним. Пример использования этих типов элемента меню на рис. 3.4. Это образец меню типовой конфигурации. Мы не можем пока разворачивать далее наш интерфейс, так как еще не создали объекты метаданных.

Теперь можете запустить конфигурацию в пользовательском режиме. Просмотрите появившееся меню. Оно нам пригодится позже, когда будут создаваться объекты метаданных. Конфигурация приобрела стандартный вид компьютерной программы и стала значительно удобнее для пользователя.

# 3.2. Создание набора пользовательских прав

Права различным категориям пользователей на доступ к информации задаются в Конфигураторе с помощью формирования набора пользовательских прав. Доступ к списку набора прав осуществляется через вкладку **Права** окна Конфигурация. Окно редактора пользовательских прав похоже на окно Конфигурация — Метаданные, но содержит только те объекты метаданных, для которых могут быть назначены права (рис. 3.4).

В пустой информационной базе создается по умолчанию один набор прав — Администратор, со всеми правами на все виды объектов метаданных, т. е. на полный доступ к информации. Этот набор прав необходим для возможности дальнейшей разработки конфигурации.

Создавать и редактировать наборы прав можно так же, как и другие составляющие интерфейса, кнопками, расположенными на панели инструментов, двойным щелчком мыши или с помощью меню **Действия**.

Для того чтобы назначить права доступа к объекту метаданных, нужно дважды щелкнуть мышью на идентификаторе метаданных. Диалог **Свойства** элемента прав организован в виде картотеки, имеющей только одну вкладку — Доступ. Вкладка содержит окно со списком операций, которые можно осуществлять над выбранным объектом метаданных. Чтобы разрешить выполнение операции, необходимо напротив нее установить флажок. Разрешение или запрещение той или иной операции может автоматически запрещать или разрешать другие логически связанные операции. Например, отключение прав на Использование любых Внешних отчетов и Обработок автоматически отключает права и на Использование общих Внешних отчетов и Обработок. А вот повторное включение права на Использование любых Внешних отчетов и Обработок отнюдь не включает автоматически право на Использование общих Внешних отчетов и Обработок. Флажок для использования этого права необходимо ставить самостоятельно. При установке прав необходимо строго следить за подобными связками.



Рис. 3.4. Окно редактора пользовательских прав

Кнопка с двумя флажками позволяет установить сразу все права на объект. Кнопка с неустановленными флажками снимает все права с объекта метаданных.

### Предупреждение

Несмотря на кажущуюся простоту работы с правами, необходимо быть очень внимательным. Права должны быть логично увязаны не только с функциональными обязанностями пользователя, но и между собой. Например, не имеет смысла давать права на подчиненный справочник при отсутствии прав на справочник-владелец.

Когда пользовательский режим запускается из конфигуратора, не забудьте установить права на монопольный режим в корне дерева прав — самой конфигурации. Иначе при запуске будет выдано сообщение "Недостаточно прав!"

### 3.3. Создание нового пользователя

Новый пользователь может создаваться через меню Администрирование | Пользователи и через кнопки на панели инструментов. Имя пользователя — произвольная строка, состоящая из букв, цифр и знаков подчеркивания, длиной не более 24 символов. Имя желательно задавать осмысленное (рис. 3.5). Если несколько пользователей имеют одинаковые права, то можно копировать их (права) через буфер обмена, меняя только имя пользователя для соблюдения уникальности. У нового пользователя имеется своя палитра свойств. Их можно не заполнять сразу, а сделать это позднее, выделив имя пользователя щелчком левой кнопки мыши в списке. Палитру можно вызвать из выпадающего меню, пункт Свойства.

На вкладке Атрибуты находятся следующие поля:

- **П** Имя поле для задания и редактирования имени пользователя;
- □ Полное имя для расшифровки имени. Это произвольная строка, которая может содержать пробелы;
- Рабочий каталог каталог пользователя, где размещаются временные файлы и т. п. Если каталога нет, к примеру, после переноса конфигурации с одного компьютера на другой (или из сети в сеть), то система прекратит работу с выдачей соответствующего сообщения. Тогда надо либо создать такой же каталог (что неудобно), либо заново ввести пользователей.

Вкладка **Роль** предназначена для выбора наименования пользовательского интерфейса и набора прав, которые будут поставлены в соответствие имени пользователя:

□ **Права** — в данном поле необходимо указать название набора прав пользователей;

<i>≱</i> е Пользователи *	
≌≥ ↓ ↑ 4 🔓 🕮 🗰	
[и [п	
Свойства пользователя	
Атрибуты Роль	
Имя: Бухгалтер	
Полное имя: Бухгалтер на зарплате	
Рабочий каталог:	

Рис. 3.5. Создание нового пользователя

Конфигуратор - Бух и опер учет, расчет - [Пользователи *]	
со Файл действия конфигурация конструкторы Администриро	
42 46 <b>1</b> 42 49 19 10 10 10 10 10 10 10 10 10 10 10 10 10	植物灌木香油 各植物
Имя Полное имя	Роль ( Права, Интерфейс)
🛇 Новый1	Не заданы права.
Свої	іства пользователя
Ат	рибуты Роль 🛛 🖓 – 🖂
<b>P</b> -	
TIP	ава. Радминистратор
Ин	терфейс: Расчет
	Отключить контроль прав
	Обновить
7	
таконфигурация *	
Для получения подсказки нажмите F1	NUM СТАТУС: БД , Документ *
🏄 Пуск 🍏 🔄 🕒 🎽 🗋 3 Проводник 🔹 🖻 Документ1 - Мі	🖻 SГлава_З - Місг 🦗 Конфигурато 📓 Paint Shop Pro 🛛 🔃 🖓 🖏 🐂 🖬 🕼 🗃 🖓 🛄 19:49

Рис. 3.6. Назначение прав и интерфейса пользователю

- Интерфейс указывается название индивидуального пользовательского интерфейса;
- Отключить контроль прав данная опция отменяет для указанного пользователя контроль доступа к информации. Это все равно, как если бы пользователю были предоставлены все права (рис. 3.6).

С целью предотвращения входа пользователей в систему под чужими именами, что влечет за собой возможность использования чужих наборов прав и пользовательских интерфейсов, каждому пользователю может быть задан пароль для входа.

Файл Дей	йствия Опер-	ации Справочники	Документы Журна	алы Отчеты	Сервис Ок	на Помощь							
1 🖻	📰 🐰 🗈	6 6 8 1	⊅ ¢ #		- <b>M</b>	a ?	5 M 19		M M+ M-				
0						J.	1			J			
<u> </u>					_	_	_	_	_		_	_	
Con and	очник: контр	рагенты ПП V4 Съ съ ра		그미즈									
	N d d d d	1 vie nie 3 vie.											
KOA	Нани	енование											
	3 Авто 2 Молни	Альянс ООО я ТОО											
	1 Поста	өщик ЧП		_									
					Предприя	тие	×						
				<u> </u>	не	остаточно г	пав доступа!						
					<u>.</u>		····						
					[	ОК	1						
_													
_													
_													
_													
Эсправочник: Контраген													
Для получе	ния подсказки	1 нажмите F1				NU	M TA: 01.01.02	2 00:00:0	00 БИ	2 квартал 2004 г.	TT:		
🎘 Пуск	Ø @ O	» 🔄 Рисунки	🗺 1С Предпр	💇 9Глава 7 -	🦉 Бе	зымянны	Paint Shop	р Р 🧃	🧲 Конфигура	а 🦿 1С:Предпр	RU 🖓	₽ «	K 21:16

Рис. 3.7. Проверка прав пользователя

Для установки пароля выделите в списке пользователей имя пользователя, для которого необходимо установить пароль, и в меню **Действия** главного меню выберите пункт **Установить пароль**. На экран будет выдан запрос для ввода пароля. *Пароль* — произвольная строка длиной не более 10 символов, где допускаются буквы, цифры, знаки подчеркивания и т. п. кроме пробелов и специальных символов. При вводе пароль будет отображаться в виде звездочек, поэтому будьте внимательны. Для надежности система запросит пароль еще раз, снова введите точно такой же пароль. Если вы ошибетесь, система выдаст соответствующее сообщение.

Теперь можно запустить конфигурацию в пользовательском режиме и проверить наличие или отсутствие соответствующих прав пользователя (рис. 3.7).

## 3.4. Общие замечания

Функция создания списка пользователей и функция создания пользовательского интерфейса, как видно из изложенного, отделены одна от другой. Такой подход достаточно гибок и прост в использовании.

Можно создать несколько типовых пользовательских интерфейсов с различным объемом команд. Например, "Кладовщик", "Менеджер", "Бухгалтер". И каждому пользователю, принадлежащему к какой-нибудь из этих категорий, просто назначается соответствующий интерфейс. Если меняется интерфейс какой-то категории пользователей, то нет необходимости производить изменения для каждого пользователя отдельно — измененный интерфейс присвоится сразу всем пользователям. И, наконец, можно отдельным категориям пользователей давать различные права и интерфейсы. Например, главный бухгалтер может иметь права администратора, а интерфейс — главного бухгалтера.

Следует постоянно помнить о логической взаимосвязи всех составляющих пользовательского интерфейса. Будет лишено смысла, если в меню будет присутствовать справочник, на который у данного пользователя нет никаких прав.



## Сквозной пример

Для применения на практике возможностей системы "1С:Предприятие" создадим конфигурацию для ведения бухгалтерского учета, учета торговых операций, ведения расчета заработной платы в торговой фирме Trade Ltd.

Основная деятельность фирмы Trade Ltd заключается в том, что она закупает у своих поставщиков товары, которые затем продает.

Необходимо организовать хранение информации:

- 🗖 о номенклатуре товаров;
- 🗖 о контрагентах фирмы;
- 🗖 о сотрудниках фирмы.

Нужно организовать движение данных о деятельности фирмы таким образом, чтобы пользователю было удобно работать с программой и не приходилось вводить одну и ту же информацию дважды. Кроме того, в любой момент иметь возможность получения следующей информации:

• о взаиморасчетах (состояние задолженностей по торговым операциям);

🛛 о продажах (в разрезе покупателей и проданных им товаров);

🗖 о расчетах с сотрудниками фирмы.

Для простоты будем считать, что у нашей фирмы только один склад, торговля ведется от имени одного юридического лица, цены и стоимость товаров учитываются в одном виде валюты, без налогов.

Для сотрудников производятся расчеты оплаты труда по окладу, а также удержания подоходного налога по фиксированной ставке.

## 4.1. Создание базовых объектов учета

Рассматривая объекты метаданных *(см. разд. 1.6)* мы показали, что существуют объекты, используемые во всех компонентах. Это так называемые базовые объекты системы.

Создание конфигурации мы начнем с создания именно этих объектов.

Исходя из поставленной задачи, можно определить состав объектов информационной базы для ведения учета.

Состав реквизитов — произвольный, но достаточный для решения поставленных задач по учету и анализу информации.

Для хранения постоянных и условно-постоянных значений, используемых в расчетах, необходимы основные константы:

- Организация для вывода в печатных формах документов полного названия фирмы;
- □ Расчетный счет для вывода в печатных формах информации о банке и расчетном счете фирмы.

Для хранения списков однотипных значений необходимо создать справочники, используемые, в том числе для ведения аналитического учета на счетах плана счетов (субконто):

- "Материалы" для хранения списков материалов, информация по которым отражается в бухгалтерском учете;
- □ "Номенклатура" для хранения списков материалов, информация по которым отражается в бухгалтерском и оперативном учете;
- "Основные средства" для хранения списков основных средств, информация по которым отражается в бухгалтерском учете;
- □ "Сотрудники" для хранения списков сотрудников и их характеристик, информация по которым отражается в учете по заработной плате;
- □ "Контрагенты" для хранения списков организаций или частных лиц, взаимоотношения с которыми отражаются в бухгалтерском и оперативном учете;
- "Основания" справочник подчинен справочнику "Контрагенты" и предназначен для хранения информации о документах, служащих основанием операций с контрагентом.

Для отражения в учете операций будем использовать документы, а для этого создадим в конфигурации основные их виды:

□ "Поступление материалов", "Списание материалов" — для учета прихода/списания материалов и взаиморасчетов с поставщиками;

- Поступление товаров", "Реализация товаров" для учета прихода/реализации товаров и взаиморасчетов с контрагентами;
- □ ПКО, РКО кассовые документы для отражения прихода и расхода наличных денег;
- "Выписка банка" банковский документ для отражения движения денежных средств на расчетном счете фирмы;
- □ "Поступление ОС", "Списание ОС" для учета прихода/списания основных средств и взаиморасчетов с поставщиками.

Для хранения документов по видам будем создавать также Журналы документов, которые обеспечат удобство работы с ними. В системе необходимо наличие журналов по каждому разделу учета и Общий журнал, включающий все виды документов.

Не используя возможности отдельных компонент, создадим отчеты:

- Реестр документов отчет для формирования списка документов за выбранный период;
- □ Основные средства отчет для формирования списка основных средств по группам.

## 4.2. Организация бухгалтерского учета

Для простоты предполагаем, что фирма Trade Ltd. только открывается, и на первом этапе учет ведется по операциям с контрагентами по приобретению материалов, основных средств.

Дополним базовый набор объектов конфигурации объектами, характерными для компоненты "Бухгалтерский учет".

Для отражения бухгалтерских операций создадим *план счетов*. Для простоты не будем следовать национальным стандартам учета, а введем в план счетов буквенные счета. Обращаем внимание, что основная наша задача — изучить подходы к конфигурированию, а не познакомиться с методикой ведения бухгалтерского учета.

Исходя из потребностей анализа информации, на некоторых счетах плана счетов будет организовано ведение аналитического учета в разрезах *субконmo*. Для этого необходимо создать несколько *видов субконто*:

- □ Материалы для ведения учета в разрезе материалов;
- **О ОС** для ведения учета в разрезе основных средств;
- □ Контрагенты для ведения учета в разрезе контрагентов.

Дополним документы, присутствующие в конфигурации после этапа создания базовых объектов системы, алгоритмами формирования бухгалтерских операций. Дополнение алгоритмов произойдет для следующих документов:

- □ "Поступление материалов";
- "Списание материалов";
- □ ПКО;
- □ РКО;
- "Выписка банка";
- **П** "Поступление ОС";
- □ "Списание ОС";
- □ "Поступление товаров".

Дополним конфигурацию документами, имеющими непосредственное отношение к бухгалтерскому учету:

- "Начисление амортизации" регламентный документ, используемый в конце периода (месяц) для автоматического расчета износа основных средств;
- "Распределение на затраты" универсальный документ для распределения учетных сумм на счета учета затрат.

Для демонстрации возможностей конфигуратора в части выборки информации создадим "Бухгалтерский отчет", приближенный к привычному виду обортно-сальдовой ведомости.

## 4.3. Организация оперативного учета

Для организации учета по основному виду деятельности фирмы Trade Ltd. (торговле товарами), дополним состав объектов системы новыми объектами. Существующие документы дополним алгоритмами для ведения учета товаров и взаиморасчетов с контрагентами, используя возможности компоненты "Оперативный учет".

Создадим новую константу Метод списания, которая позволит выбрать метод расчета себестоимости реализованного товара согласно учетной политике фирмы.

Используем возможности компоненты "Оперативный учет" для отражения взаиморасчетов с контрагентами в разрезе договоров и для расчета себестоимости по выбранному методу списания. Для этого создадим регистры оперативного учета:

- □ Взаиморасчеты для учета взаиморасчетов с контрагентами по торговым операциям;
- **Партии** для ведения учета партий товаров.

Дополним алгоритмы отражения в оперативном учете операций по взаиморасчетам и движению товаров в следующих документах:

- □ "Поступление товаров";
- "Реализация товаров";
- □ ПКО;
- □ PKO;
- "Выписка банка".

Для удобства работы сотрудников, занимающихся отслеживанием взаиморасчетов с контрагентами, в Общем журнале документов организуем **Отбор по контрагентам**.

Используя информацию оперативного учета, дополним бухгалтерские отчеты оперативными, такими как:

- "Продажи" отчет о продажах и прибыли;
- "Взаиморасчеты" состояние взаиморасчетов с контрагентам по торговым операциям.

## 4.4. Организация расчета заработной платы

Для организации учета по сотрудникам фирмы Trade Ltd. дополним состав конфигурации новыми объектами, характерными для компоненты "Расчет". Существующие объекты дополним новыми реквизитами и алгоритмами, необходимыми для расчета заработной платы.

Сотрудникам фирмы согласно контракту устанавливается фиксированная сумма заработной платы (оклад) с шестидневной рабочей неделей, по 6 часов в день.

Для расчета заработной платы нам понадобятся постоянные значения, вернее, условно-постоянные, так как они устанавливаются, как правило, законодательно:

□ Ставка подоходного налога — периодический реквизит, показывающий процент удержания подоходного налога;

□ Минимальный расчетный показатель — периодический реквизит, показывающий значение минимальной льготы по подоходному налогу.

Для организации учета расчетов сотрудников нам также необходимы дополнительные справочники:

"Подразделения" — для учета сотрудников по подразделениям.

Дополним новыми реквизитами существующий справочник "Сотрудники":

- **Подразделение** периодический реквизит;
- **Оклад** периодический реквизит;
- □ Приказ о приеме на работу хранится документ о приеме на работу;
- □ **Льгота по подоходному налогу** периодический реквизит для указания наличия льготы на минимальный расчетный показатель у сотрудника;
- □ **Дата приема** дата приема сотрудника;
- **Дата увольнения** дата увольнения сотрудника.

Для ведения учета по сотрудникам с использованием возможностей компоненты "Расчет" дополним конфигурацию документами:

- □ "Приказ о приеме на работу" для занесения основных данных по сотруднику в справочник на момент приема;
- "Начало месяца" регламентный документ для ввода базовых записей для расчета заработной платы;
- □ "Выплата заработной платы" документ для расчета по сотрудникам сумм к выплате.
- Создадим и основные объекты компоненты "Расчет":
- журнал расчетов для хранения и отражения расчетов по заработной плате;
- календарь Основной для определения баланса рабочего времени сотрудников;
- виды расчетов:
  - По окладу для расчета основного начисления сотрудника;
  - Подоходный налог для расчета основного налога сотрудника по действующей ставке;
  - Выплата зарплаты для расчета суммы, полагающейся сотруднику к выплате.

Основной алгоритм этого раздела учета — расчет заработной платы — реализуем с помощью обработки **Расчет заработной платы**.

Для просмотра и анализа результатов расчетов с сотрудниками создадим в конфигурации отчеты, используя информацию из журнала расчетов:

- □ Расчетная ведомость отчет по сотрудникам за период с суммами начислений и удержаний в разрезе видов расчетов;
- □ Свод по зарплате итоговый отчет по фирме с суммами начислений и удержаний по видам расчетов.

В результате работы по созданию учебной конфигурации для учета фирмы Trade Ltd. вы получите практические навыки по организации автоматизированного учета с использованием возможностей всех трех компонент системы "1С:Предприятие 7.7".

Помимо перечисленных объектов метаданных в конфигурации для лучшего усвоения материала будут использоваться некоторые дополнительные объекты.



## Константы и обработки

С этой главы начинается непосредственное описание процесса создания учебной конфигурации. С помощью констант и обработок будет продемонстрирована связь между отдельными объектами метаданных в пределах конфигурации.

## 5.1. Настройки первоначальных сведений об организации

Начнем создавать конфигурацию в соответствии со сквозным примером, описанным в предыдущей главе. Какие сведения могут или должны отображаться в документах предприятия? Это наименование предприятия, дата регистрации предприятия, адреса — юридический и почтовый, ФИО руководителя и главного бухгалтера и т. п. Какие виды метаданных могут исобработки пользоваться для хранения И этих данных в системе "1С:Предприятие"? Согласно описаниям (см. главу 1) это константы и справочники. В константах хранится постоянная или условно-постоянная информация. Дата регистрации предприятия — постоянная информация. Эту дату мы будем хранить в константе.

Константами можно назначить фамилию, имя и отчество руководителя предприятия и главного бухгалтера — для расшифровки подписи. Фамилия главного бухгалтера задана константой. Если он уйдет в отпуск, его будет замещать другой человек. Его фамилия также должна быть задана константой. Потом главный бухгалтер вернулся — значение константы меняется. Уехал в командировку — снова меняется. Пройдет много времени и возникнет задача: быстро выписать документы нужным числом с фамилией того главного бухгалтера, который был на момент выписки документа. Вспоминать? Перебирать бумажные копии? А затем, поставив фамилию годичной

давности, снова входить в константу и ставить ФИО действующего главного бухгалтера? Для простого решения подобных вопросов константы имеют признак периодичности. Периодичность означает, что значения константы не просто замещаются другими, а сохраняются в системе, причем каждому значению соответствует дата, для которой это значение актуально. И если мы несколько раз изменили сведения о главном бухгалтере, то в периодической константе эти данные будут храниться в следующем виде (табл. 5.1).

Дата внесения значения	Значение
12.04.01	Иванов И. П.
15.12.01	Петров А. В.
01.09.02	Иванов И. П.

Таблица 5.1. Значения периодической константы

Такие константы называются периодическими.

Следующий вид метаданных, необходимый для построения конфигурации — Обработка. Обработки — вид метаданных, которые используются для выполнения различных действий над информационной базой. Те, кто имеет опыт работы в "1С:Предприятии", вероятно, знакомы с настройкой параметров учета, обработкой документов, с путеводителем по конфигурации и всевозможными помощниками. Область применения обработок весьма обширна. Мы познакомимся с этим видом метаданных, создав обработку "Настройка параметров учета".

В ветви метаданных Обработки создадим новую обработку с идентификатором НастройкаУчета и синонимом Настройка параметров учета. Для этого щелкнем левой кнопкой мыши по ветви Обработки. Щелчком правой кнопки мыши откроем контекстное меню и выберем пункт Новая обработка. В открывшейся палитре свойств установим значения идентификатора и синонима обработки. Комментарий не нужен. Перейдем на вкладку Права. В левом окне палитры свойств появился список прав, он состоит из одного значения Администратор. Этот набор пользовательских прав мы создали ранее (см. главу 3). В правом окне устанавливаются, непосредственно, права на обработку. В данном случае установлен флажок Использование, и мы его оставляем без изменений, так как планируем разрешить администратору использовать эту обработку, по крайней мере, на этапе разработки конфигурации (рис. 5.1). Ведь если снять этот флажок, обработка будет недоступна для проверки. Нажмем кнопку ОК, и обработка появится в дереве метаданных. Дважды шелкнув левой кнопкой мыши по идентификатору обработки в дереве метаданных, мы вызовем экранную форму обработки (рис. 5.2).





👺 Конфигуратор - Буж и опер учет, расчет - [Форма-Обработка.новый1]	- 8 ×
🖞 Файл действия Диалог Вставить Конфигурация Конструкторы Администрирование Сервис Окна Помощь	la X
12 26 日 26 19 10 10 10 10 10 10 10 10 10 10 10 10 10	
· \$P\$ \$P\$ \$P\$ \$P\$ \$P\$ \$P\$ \$P\$ \$P\$ \$P\$ \$P	
Выполнить	
Califinanor / Maguno / Alifadinaua /	
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	
ХадКонфитурация * ДФорна-Обработка.нов	
Для получения подоказки нажинте F1 [NUM (54 x 14 70, 142 СТАТУС: 5Д *, Док	/мент *
街 Tryck 🔊 😳 🏵 🐣 🗋 Kisera 🔄 『 T nasa5 👘 Документ 1 🞼 Kontykryp 🧱 Paint Shop Pro 🔃 🍞 🖑 🔍 5 西 宮 体 電気 🖓	) 14:29

Рис. 5.2. Экранная форма обработки

Экранная форма обработки — это заготовка, созданная системой. Мы можем самостоятельно изменять и дополнять ее. Щелкните по форме левой кнопкой мыши. Форма выделится голубым прямоугольником с рисками по углам и бокам. Если подвести указатель мыши к такой риске, то можно уве-
личить либо уменьшить форму, растянуть ее или сжать. Если щелкнуть мышью кнопку, то она тоже будет выделена подобным образом. Экранная форма служит для диалога системы с пользователем. Через экранные формы объектов метаданных пользователь вводит и изменяет данные. Все, что находится внутри экранной формы, — это элементы диалога. Элементами диалога в данном случае являются кнопки **Выполнить** и **Закрыть**. У новых элементов диалога, которые мы будем добавлять при разработке формы, также можно менять размеры и местоположение в форме. Добавить новые элементы диалога можно с помощью меню **Вставить** | **Реквизит диалога** либо кнопкой на панели инструментов **Элементы диалога** | **Реквизит диалога**.

#### Примечание

Палитра свойств реквизита диалога немного отличается от палитры свойств объекта метаданных. Например, тип значения устанавливается на вкладке **Тип**, а не на вкладке **Общие**.

# 5.2. Примеры. Создание констант и обработок для Trade.Ltd

В открытом окне конфигурации найдите ветвь метаданных **Константы** и щелкните левой кнопкой мыши на ней. Затем щелкните правой кнопкой мыши. Откроется контекстное меню, в котором надо выбрать пункт **Новая Константа**. На экран будут выведены все константы конфигурации и появится палитра свойств константы (рис. 5.3). Зададим эти свойства.

**Идентификатор** — ДатаРегистрации;

**Синоним** — Дата регистрации;

**С Комментарий** — Дата регистрации предприятия.

**Тип значения** показывает, какого типа данные могут храниться в константе. Дата регистрации предприятия состоит именно из даты. Щелкните по раскрывающемуся списку типа значения и выберите тип значения **Дата**.

Теперь нажмем на палитре свойств кнопку **ОК** и запустим "1С:Предприятие" в пользовательском режиме. Это можно сделать с помощью кнопки на панели инструментов или через пункт меню **Сервис** | **1С:Предприятие**.

#### Примечание

В диалоговом окне **Реорганизация информации** появятся сведения только о константе. Сведений о новой обработке не будет, так как в обработках, как и в отчетах, используются ссылки на уже созданные объекты метаданных без их изменения.



Рис. 5.3. Палитра свойств константы

Посмотрим результаты нашего конфигурирования. Выберем пункт меню (в пользовательском режиме!) Операции | Константы.... Откроется список констант (рис. 5.4). Обратите внимание: идентификатор константы вообще не виден — это внутреннее название константы. А вот синоним (в графе Код) и комментарий (в графе Наименование) помогают пользователю понять, какое значение нужно присваивать константе (графа Значение). Каждый раз, когда произошла, скажем, перерегистрация предприятия, в графе Значение вы можете поставить новую дату. Графы Код и Наименование (синоним и комментарий) будут неизменны, так как они заданы на этапе конфигурирования.

Теперь информация о дате регистрации предприятия может быть помещена в систему, удалена из нее, изменена, но в любом случае место для этой информации уже зарезервировано.

Подобным же образом нужно создать константу **Организация**. Идентификатор **Организация**, синоним и комментарий **Наименование организации**. Тип значения — "Строка", длина — 30 символов.

Advictami Onepauru Cipasovanu Aovymenta Xypuana Pesynatara pacieta Orverta Copadorni Copaci Orva Tonous	Alektreme Onepaum Opepaum Opepaum Opepaum Opepaum Agymental Kyphenal Pesynetatu packer of Oreana Opepaum Operational Contractional Contraction	Alektreme Onepaum Cropeosenson Agoymetha Wyphena Pesymetatu pacerer Onepai Orgadoria Cepeix Orea Dokous          Image:	Предприятие - Бух и опер учет	, расчет		
Image:	Image:	Image:	I Действия Операции Справочни	ики Документы Журналы Результатыр	асчета Отчеты Обработки Сервис Окна Помощь	
Конк констант (22.11.06)	Тисок констант (22.11.06)	Тисскі констант (22.11.06)	🖻 🖩 👗 🖪 🖓 🖓	M 5 C M	тйй? 🖬 🛒 💯 мм+м-	
Наменование       Значение       А         иналии       Органзация       Росчинай говара       FIFO         иналии       Росчинай говара       FIFO	А Наменование Значение     Органование ГРО     Органование ГРО     Органование ГРО     Органование Сеегорганизации     Органование Сеегоргание Сеегоргание Сеегоргание     Органование Сеегоргание Сеегоргание Сеегоргание Сеегоргание Сеегоргание     Органование Сеегоргание Сеегоргание Сеегоргание     Органование Сеегоргание Сеего	Политическование       Значение         Синима       Органовация       ГРО         Политическование       ГРО       Гро	писок констант (22.11.06)			
Наименование         Значение           нисация         Організация           ад слисання товара         FFD           рекладная         Наименование           рекладная         Мата слисання товара           рекладная         Наименование           рекладная         Наименование           рекладная         Наименование           рекладная         Мата слисання товара           рекладная         Пата начала отчета по унолично           ка подкождного налога         10.00           рекладная         Пата начала отчета по унолично           дата рекладноказтель            рекладноказтель            рекладноказтель            по сладоходного налога         10.00           по сладоходного налога	Видение         Значение           А         Наименование         Значение           Ализация         Органсоция         Срадина           Год. Стисания         Метод. Списания товра         FIFO           Уличение         Валение            Консания         Метод. Списания товра         FIFO           Уличение         Валение            Консания         Метод. Списания товра         FIFO           Уличение         Валение            Консания         Метод. Списания товра         FIFO           Консания         Метод. Списания товра            Консания         Метод. Списания товра            Консания         Метод. Списания товра            Консания         Мин. роектем.            Консания         Пато регистрации прадприятия            Стания         С             Стания         С             Стания         С             Стания         С             Стания         С             Стания         С	А Наименивание Значение Липонание Значение				
Наменование Значение	А Наменование Зночение	а Наменование Зночеше -				
оннования         Органования         ГГО           вельскания         Матад списания товара         ГГО           вельскания         Расчетный счет органовации	Рамъзина Органозция Градора ГГО по сликована М Матод списания повара ГГО ката начизая отчета Дага начизая отчета по умолнанию	Ранкация  Органозация  Органозация  Органозация  Ранкация  Ранкация Ранкация Ранкация Ранкация Ранкация Ранкация Ранкация Ранкация Ранкация Ранкация Ранкация Ранкация Ранкация Ранкация Ранкация Р	д	Наименование	Значение	
да списания         Метад списания товара         FIFO           Овласний         Метад списания товара         FIFO           Овласний         Полочина         Полочина           Полочина         Полочина         Полочина<	Гад слисания Метод слисания товара FIFO тод слисания Метод слисания товара FIFO жотный счет Рачитый счет организации жотный для начала отчет по умолнанно и ланизатов Мини роектный показатель вка падохадного налога Станка падохадного налога падетнистрации Дата регистрации предприятия га регистрации Пата регистрации предприятия падетнистрации Пата регистрации предприятия падетнистрации Пата регистрации предприятия падетнистрации Пата регистрации предприятия падетнистрации Пата регистрации предприятия падетнистрации Пата регистрации Пата регистрац	Гад слисания Метад слисания товара FIFO тов слисания от мата Пата начала отчета по унолизино кото налога Пата начала отчета по унолизино на растистрации Лата регистрации предприятия та регистрации Дата регистрации предприятия по напострации Пата регистрации предприятия по напострации по напострации предприятия по напострации по напострации п	ганизация	Организация		
етный счет Расчетный счет организации	ретный счет расный счет организации сонавидать начала отчета по моличноо сонавидать начала сонавидите на Почала сонавидите начала сонавидите начала сонавидите начала сонавидите начала сонавидите начала сонавидите нача сонавидите начала сонавидите начала сонавидите начала сонави начала сонавидите начала сонавидите начала сонавидите начала	ретный счет разначала отчета по умоличноо	тод списания	Метод списания товара	FIFO	
аная дата начала отчета по уноличнио	овная дата начала отчета и дита начала отчета по умолнанио	овная дата начала отчета о умоливию	четный счет	Расчетный счет организации		
расчетны показатель Мин расчетны показатель 10.00	разнатных показатель Мин. разчатыви показатель 10.00 в падюждито налота Ставка показатель 10.00 а регистрации Дата регистрации предприятия	раченны показатель Мин. расенны показатель в падюждится наюта Станая пасказатель а регистрация Дата регистрация предприятия	ювная дата начала отчета	Дата начала отчета по умолчанию		
			н. расчетный показатель	мин. расчетный показатель	10.00	
			а регистрации	Пата регистрации предприятия	10.00	
			а регистрации	зана регистрации предприятия		
			/22.11			
ок констант (22.11	кок констант (22.11	кок констант (22.11				
ток констант (22.11 учения подсолин муните F1 NBM_ТА: 29.09.06.15:29:36 БИ: 4 краутая 2006 г ТП: Остябрь 2006	жок констант (22.11 поченая подселия начантя F1 NBM Та: 29.09.06.15:20:36 БИ/: 4 клаптал 2006 г. ПП- Онтабль 2006 г.	кок констант (22.11 поченая подселов начанта F1 NUM_ТА: 29.09.06.15:29:36 FM: 4 клаптал 2006 г TD: Октабла 2006 г.	пучения подсказки нажите Е1		NEM TA: 29.09.06 15:29:36 EV: 4 квартар 2006 г. TD: O	ктябрь 2006 г.

Рис. 5.4. Список констант конфигурации в пользовательском режиме

Часто бывает удобным, а иной раз и необходимым, изменять (присваивать) значения констант из встроенного языка. Синтаксис такой процедуры:

Константа.Идентификатор = Переменная;

Обратите внимание: слово константа — это оператор встроенного языка, который указывает, что работа будет производиться с объектом метаданных типа "Константа". А через точку задается идентификатор константы.

Обратная процедура — присваивание переменной значения константы, в общем виде описывается так:

Переменная= Константа.Идентификатор;

Для создания периодической константы необходимо проделать все шаги, которые мы делали ранее для создания простой константы. Тип значения — "Строка". В палитре свойств константы после заполнения полей вкладки **Общие** перейдите на вкладку **Дополнительные** и поставьте флажок периодичности (рис. 5.5).



Рис. 5.5. Установка признака периодичности в свойствах константы

Обратите внимание, как изменился значок константы в конфигураторе — теперь он похож на набор карточек. Сохраните конфигурацию и перейдите в пользовательский режим, через меню **Операции** | Константы откройте окно констант и щелкните мышью на константе ГлБух. Теперь на панели инструментов окна констант активизировалась кнопка История (можно нажать клавишу <F5>) (рис. 5.6). Щелкните мышью эту кнопку — откроется окно с таблицей, похожей на табл. 5.1, — окно истории значений константы задним числом. Обратите внимание, что в колонке Дата имеется кнопка с календарем для выбора даты. Затем закройте окно истории значений константы и сторию значений константы и посмотрите — внесенному значению автоматически присвоена текущая дата системы. В дальнейшем так будет происходить постоянно.

пертириатие — вуя и опер учи Действия: Операции: Справо Правона — Справона — Справона Правона — Справона — С	ег, расчет ники Документы Журналы Результатырасч . 🔟 🖄 🖒 🏟 💽 🔽	ета Отчеты Обработки Сереис Окна Понзиць 🔏 🔏 😵 🛛 🔤 🕅 🕬 💭 М М+ М-	
*    <b>             </b>	国 5 c A	йй 🖇 🖇 🔲 🖬 🕬 🛄 м м+м-	
Список констант (23.11.06)			
a -2 (2)			
(	Hannahan anna	201200000	
лод Организация	паименование	Значение	
(PD			
летодилисания Расчетный Смет			
СтавкаПН		10.00	
Існовная дата начала отчета	Дата начала отчета по умолчанию	01.01.06	
Контроль		1	
ФИОТЛ. 0gx	чис гладного одягалтера		
ча история: чилота: о	y×		- <b>- ×</b>
- 一 二 二 二 二 二 二 二 二 二 二 二 二 二 二 二 二 二 二	Значение		
Aara	Значения	• 	
· · · · · · · · · · · · · · · · · · ·	3начеляне 	•	_ D X
Arra	3.607001846	•	
Aara	3637929800	•	X
····································	3.60%03868	•	
	3169901866	•	
문지 아이지 아이지 아이지 아이지 아이지 아이지 아이지 아이지 아이지 않는 것 같아.	3качение 3качение	•	
Arto Arto Arto	3.60%0.848	•	
	23tervenisee	•	
		•	
	3noresisce	•	
	3Horean Base 3Horean Base 	•	
	3novenses	•	
	3novenses	•	
	3noresee	•	
	3itovenaec	•	
	3novense	•	
	3noresisce	•	
	3novenses	•	
		•	
	3novenee  3novenee	•	
	236090360	•	
	794 3ttorwoitee 	•	

Глава 5

Рис. 5.6. История значений периодической константы

#### Замечание

На вкладке Дополнительные есть еще два флажка: Разделять триады и Неотрицательный. Эти свойства относятся только к объектам метаданных с типом значения "Число". Если установлен флажок Разделять триады, при вводе и показе значения объекта метаданных будут автоматически вставляться разделители между тройками цифр, разделяя тысячи, миллионы, миллиарды. Если установлен флажок Неотрицательный, система будет автоматически проверять, чтобы в качестве значения объекта метаданных было указано только неотрицательное число.

Процедура присваивания значения периодической константе с помощью встроенного языка выглядит так:

При этом дата — это выражение со значением требуемой даты, а переменная содержит необходимое значение. Другими словами, параметры в скобках соответствуют строке табл. 5.1.

Для получения актуального значения периодической константы используется метод Получить (). Этот метод возвращает значение периодической константы на заданную дату.

Переменная = Константа.Идентификатор.Получить (Дата);

Теперь вернемся к обработке, экранная форма которой так и открыта в конфигураторе. Если вы ее закрыли, найдите ветвь метаданных Обработки, щелкните по знаку "+", затем по названию обработки, и откроется ее экранная форма. В ней мы будем задавать значения констант. В нашей учебной конфигурации их пока три: Дата регистрации предприятия, ФИО главного бухгалтера, Организация. В форму обработки нужно добавить поля, чтобы вводить новые значения констант. Для этого используем возможность вставки реквизита диалога, как описывалось ранее. Для ввода значения константы ДатаРегистрации создадим реквизит диалога с идентификатором ДатРег, типом значения "Дата". Для удобства пользователя выставим в палитре свойств реквизита свойство Имеет кнопку выбора (рис. 5.7).



Рис. 5.7. Установка свойства реквизита Имеет кнопку выбора

Предполагается, что дата, введенная в этот реквизит, будет с помощью встроенного языка присвоена константе **ДатаРегистрации**. Для ввода наименования организации создадим реквизит **Организация** с типом значения "Строка" длиной 30 символов соответственно типу значения константы. Для ввода ФИО главного бухгалтера создадим реквизит диалога с идентификатором **ГлБух**, типом значения "Строка" длиной 20 символов соответственно типу значения константы **ГлБух**. Теперь напишем наименования созданных

реквизитов для пользователя. Используя команды меню Вставить | Текст или соответствующий значок на панели инструментов, вставим реквизит диалога Текст в экранную форму (рис. 5.8). В открывшихся свойствах текста, в заголовке, напишем наименование реквизита: Организация. Щелкнем кнопку ОК — и в экранной форме появится наименование реквизита. Таким же образом нужно создать поясняющие надписи к другим реквизитам.



Рис. 5.8. Реквизиты диалога в экранной форме обработки

Для того чтобы ровно расположить реквизиты диалога внутри экранной формы, их надо предварительно выделить. Сделать это можно двумя способами:

- □ удерживая нажатой клавишу <Shift> на клавиатуре компьютера, щелкните мышью по тем реквизитам, которые хотите выделить;
- подведя указатель мыши, например, к верхнему левому углу воображаемого прямоугольника, в котором находятся реквизиты в экранной форме, нажмите левую кнопку мыши и, не отпуская ее, ведите указатель мыши к

правому нижнему углу воображаемого прямоугольника по диагонали. При этом будут видны границы прямоугольника в виде пунктирных линий. Все реквизиты, оказавшиеся внутри этого прямоугольника, будут помечены (рис. 5.9).



Рис. 5.9. Выделение и пометка группы реквизитов диалога

Выровнять помеченные реквизиты можно, используя меню Диалог | Выровнять.

Помеченные реквизиты диалога можно перемещать внутри экранной формы (и, частично, за ее пределами) клавишами управления курсором.

Перейдем в модуль формы обработки. Для этого щелкнем по вкладке **Модуль** внизу экранной формы (рис. 5.10), в этой вкладке пишется программа на встроенном языке. Программа будет обрабатывать данные, введенные пользователем в форму, и реагировать на его действия (нажатие кнопок и прочее). Удалим созданную конструктором по умолчанию процедуру Выполнить () И создадим две новых предопределенных процедуры ПриОткрытии() И ПриЗакрытии().



Рис. 5.10. Модуль формы обработки

Назначение предопределенной процедуры ПриОткрытии () понятно из ее названия. Она выполняется при открытии формы обработки. С ее помощью мы помещаем в реквизиты **ДатРег**, **Организация** и **ГлБух** значения констант, актуальные на дату открытия обработки.

```
Листинг 5.1
```

```
Процедура ПриОткрытии()
Организация = Константа.Организация;
ДатРег = Константа.ДатаРегистрации;
ГлБух = Константа.ГлБух.Получить(РабочаяДата());
КонецПроцедуры
```

В этом и следующем (листинг 5.2) листингах демонстрируется практическая реализация работы с константой из встроенного языка, теоретический синтаксис которой приводился ранее.

Константа **ГлБух** периодическая, поэтому для получения ее актуального значения используется метод Получить ().

Дата задается системной функцией РабочаяДата(). Эта функция позволяет устанавливать дату непосредственно в системе "1С:Предприятие". Это необходимо, когда системная дата компьютера и дата каких-либо хозяйственных операций различны. Например, если ввод данных в информационную базу осуществляется задним числом. Рабочую дату мы можем установить, вызвав в пользовательском режиме меню Сервис | Параметры. За исключением специальных случаев, при разработке конфигураций лучше пользоваться системной функцией РабочаяДата().

В реквизитах обработки можно задать новые значения констант. При закрытии обработки эти значения присвоятся константам с помощью предопределенной процедуры ПриЗакрытии ():

```
Листинг 5.2
```

```
Процедура ПриЗакрытии()
Константа.ДатаРегистрации = ДатРег;
Константа.Организация. = Организация;
Константа.ГлБух.Установить (РабочаяДата(),ГлБух);
КонецПроцедуры
```

Для периодической константы ГлБух применен метод Установить (), который определяет значение константы на указанную дату. Дата также задается системной функцией РабочаяДата (), а новое значение константы берется из реквизита ГлБух.

Эта обработка должна вызываться при первом запуске конфигурации в пользовательском режиме. Пользователь, по желанию, введет необходимые значения констант при первом запуске либо позднее. Поэтому необходимо создать механизм, обеспечивающий вывод обработки при первом запуске либо при следующих запусках, иначе обработка будет вызываться пользователем через меню. Вернемся к экранной форме документа, щелкнув по вкладке **Диалог**.

Диалог вызова обработки реализуем посредством флажка. Его, так же, как и в другие элементы диалога, можно вставить в экранную форму, используя опции меню Вставить | Флажок или соответствующий значок на панели инструментов Элементы диалога. Идентификатор — Контроль, Заголовок — Показывать при запуске (рис. 5.11).



Рис. 5.11. Флажок для диалога вызова обработки

Если щелчком мыши мы установим флажок, то значение этого реквизита будет равно 1. Если флажок не установлен, его значение 0. Исходя из этого свойства, будем писать программу.

Значение данного флажка необходимо сохранять и обрабатывать. Для этого создадим новую константу-идентификатор **Контроль**, тип значения — "Число", длина — 1 символ. В этой константе мы будем хранить значение флажка до его изменения в результате События. Событие — щелчок мышью — обрабатывается процедурой Щелкнуть\_Флаг(). Каким образом системе становится понятно, что при щелчке мышью по данному флажку необходимо запустить именно процедуру Щелкнуть\_Флаг()? Ссылка на эту процедуру устанавливается в палитре свойств флажка, на вкладке Дополнительно. Щелкните мышью по этой вкладке и в поле **Формула** напишите ссылку на процедуру: Щелкнуть\_Флаг(). Не забудьте о скобках в конце наименования процедуры (рис. 5.12). Теперь в пользовательском режиме при щелчке мышью по флажку будет отрабатываться указанная процедура.

C	войства	ι φ <i>ι</i>	ажка
	Общие	Д	ополнительн
	Форму	ла:	Щелкнуть П Прижа
			🔲 Пропу

Рис. 5.12. Ссылка на процедуру обработки события

### Листинг 5.3 Процедура Щелкнуть\_Флаг() ПоказыватьНастройкуТорговли=?(Контроль=1, "1", "2"); Константа.Контроль = ПоказыватьНастройкуТорговли; КонецПроцедуры

В первой строке процедуры переменной Показывать НастройкуТорговли присваивается значение 1 или 2. Для чего присваиваются эти значения? Как уже говорилось, флажок **Контроль** может возвращать два значения: 0 и 1. Эти значения обрабатываются системой для выяснения, установлен флажок или снят. Но нужно еще выяснить, открывать ли обработку при каждом запуске конфигурации в пользовательском режиме? То есть появляется третья ситуация, для обработки которой мы вводим третье значение — 2. Все вышеописанное реализуется с помощью системной функции ? — вычислить выражение по условию. Эта функция вычисляет значение выражения, записанного первым в скобках (Контроль=1). Если выражение истинно, флажок установлен, переменной присваивается значение, следующее за выражением (1). Если выражение ложно — флажок не установлен, переменной присваивается последнее значение в скобках (2). В следующей строке процедуры константе Контроль присваивается значение переменной Показывать Настройку Торговли.

Проверять значение константы контроль необходимо при каждом запуске конфигурации в пользовательском режиме. При старте конфигурации первым запускается глобальный модуль *(см. разд. 3.2)*. Следовательно, в глобальном модуле и разместим процедуру проверки значения константы, воспользовавшись для этого предопределенной процедурой ПриНачалеРаботыСистемы().

```
Листинг 5.4
```

//предопределенная процедура
Процедура ПринАчалеРаботыСистемы()
ПоказыватьНастройкуТорговли=Константа.Контроль;
Если ПоказыватьНастройкуТорговли<2 Тогда
ОткрытьФормуМодально("Обработка.НастройкаУчета");
КонецЕсли;
КонецПроцедуры

Следует обратить внимание на то, что для открытия формы необходимо задать объект метаданных, точку и идентификатор объекта метаданных.

Таким образом, процедура ПриНачалеРаботыСистемы() покажет обработку только в том случае, если установлен флажок Показывать при запуске (значение константы контроль меньше 2). Если это условие не выполняется, обработка при запуске конфигурации в пользовательском режиме показана не будет. Но при необходимости ее можно открыть с помощью команды меню. При открытии формы необходимо опять проверить значение константы контроль и, соответственно, установить значение флажка Контроль. Допишем необходимые команды в процедуру ПриОткрытии() модуля формы обработки.

#### Листинг 5.5

```
Процедура ПриОткрытии()
ПоказыватьНастройкуТорговли = Константа.Контроль;
Контроль=?(Число(ПоказыватьНастройкуТорговли)<2, 1, 0);
ДатРег = Константа.ДатаРегистрации;
ГлБух = Константа.ГлБух.Получить(РабочаяДата());
КонецПроцедуры
```

78

Механизм работы процедуры тот же, что и в процедуре Щелкнуть\_Флаг().

Теперь можно проверить работу созданной обработки. Для этого нужно запустить конфигурацию в пользовательском режиме. При первом запуске обработка покажет введенные ранее значения констант (или пустые значения, если константы не заполнялись). Можно изменить/заполнить значения в обработке, снять флажок **Показывать при запуске** и в дальнейшем вызывать обработку только при необходимости внесения изменений. Для этого можно использовать меню **Операции | Обработка** либо вывести вызов обработки в пользовательский интерфейс.

Глава 6



# Справочники и перечисления

Справочники и перечисления — близкие по восприятию пользователя элементы метаданных. На самом деле между ними существенная разница. Справочник является списком возможных значений, который создается в конфигурации, но заполняется пользователем. Справочник может использоваться как самостоятельная единица конфигурации. Перечисление не используется самостоятельно, а применяется, в основном, в совокупности с другими типами данных. Перечисление — это список возможных значений реквизита, задаваемый на этапе конфигурирования без возможности редактирования в пользовательском режиме.

## 6.1. Общие сведения о справочниках

Фирма, автоматизацию которой мы организуем, занимается торговозакупочной деятельностью, поэтому для нее необходимо организовать систему ввода и хранения нормативно-справочной информации. Это могут быть сведения о товарах, о контрагентах — покупателях или поставщиках товара, информация о сотрудниках фирмы. Такого рода информация имеет следующие особенности:

- 🗖 как правило, заносится один раз, затем корректируется и дополняется;
- постоянно необходима в той или иной степени в процессе работы;
- понимание этой информации должно быть однозначным для всех пользователей системы. Например, сведения о контрагентах необходимы постоянно при выписке приходно-расходных документов, при формировании отчетности;
- □ эти сведения корректируются и дополняются (или убавляются) пользователями системы. Обратите внимание на этот момент!

#### Вот как выглядит информация в типичном справочнике (рис. 6.1).

<b>ॐ</b> 1С:Предприятие - Торговля	і+Ск	лад для Казахстана, редакция 1.0 - [Спр	оавочник: Контраге	нты]	_ & ×
🗐 Файл Действия Операции	Спр	авочники Документы Отчеты Сервис С	жна Помощь		
]" <b>`` ☞ </b> ♥  X ♥\$ ♥\$  @	\$ 69		_#`# ¥ []	Ш № U м м+ м-	
2 2 2 2 2	) 🕅	, TOE HARM INC. 🥠 🥔 🖹 🖂 🔜			
"自己思论自身吗?"	ie Di	e e e e v x			
П-СП Контрагенты		Наименование	РНН	Полное наименование	Код 🔺
Налоговые	61	Организации			00000001
— 🔄 Организации		Фирма Триада+ ТОО	600200037629	Фирма Триада+ ТОО	00000016
Пенсионные фонлы		"Мастер ИНК" ТОО	600700168038	"Мастер ИНК" ТОО	00000044
— Э Ууреантеан		«Издательский Дом «Алма-Медиа» ТОО	600900157702	«Издательский Дом «Алма-Медиа» ТОО	00000037
		JCF Electronics		JCF Electronics	00000022
	Ċ1	KAZAKHSTAN Realty	600400063911	KAZAKHSTAN Realty	00000028
		KULAN REALTY TOO	600900551280	KULAN REALTY TOO	00000039
	È	Shark TOO		Shark TOO	00000021
	È	Tenir-Project TOO	600500077211	Tenir-Project TOO	00000020
	È	Timer+ TOO	600700541616	Timer+ TOO	00000029
	È	Trodat		Trodat	00000007
	Ò	Алматинский Юридический колледж МВД РК	600800038523	Алматинский Юридический колледж МВД Республики Казахстан	00000034
	<u>Ò</u>	Альфатех ТОО		Альфатех ТОО	00000025
	<u>ì</u>	Бастау ТОО	600800046894	Бастау ТОО	00000038
	<u> </u>	Импрод ТОО	582100015007	ТОО Импрод	00000040
		Казинформателеком АО	600300037182	АВ Казинформтелеком	00000035
	<u> </u>	КазНЦКЗИ им. М. Айкимбаева		Казахский научный центр карантинных и зоонозных инфекций им. М. Ал	00000011
	<u> </u>	Мастер Инк. ТОО		Мастер Инк ТОО	00000014
		Меломан IUU, филиал		Видео-Аудио-Меломан IUU филиал	0000012
	<u> </u>	НефтьСтрой		НефтьСтрой	00000030
		НТЦ Элмас-БИСТОО	600800040576	НТЦ Элмас-БИСТОО	00000033
		НЧОУ "ТОЦ Казинформтелеком"	600900180256	НЧОУ "ТОЦ Казинформтелеком"	0000036
	Ļ	UAU ТексакаБанк		UAU Тексакаbанк	0000010
	무	Ортком ТОО	600400507153	Ортком ГОО	0000018
	븻	пульскар тоо	600300036461	Пульс-3р ТОО РЕКЕ КЦИК 214 жилли ICA 2 СТ 2000 10040	00000013
	4	ГГ N I NRU NJU ПРОЕКТ ILA2-L I -2000-10048	60000027469	ГТ NT NRUNOV Проект ILA2-L1-2000-10048	00000024
	는	Рынок Алтаи		Рынок Алтаи	00000027
	븡	Control Vision Control TOO	C0000004E72E	Control Mentany versions TOO	00000026
	-	Сехории 21 ТОО	E00300043735	Concerner 21 TOD	00000043
	岩	Соторик-27 ТОО Солта вои/интерневшие ТОО	60030032361	Соторик 21 гоо	00000015
	岩	TOO "K as HOIL High Tools for Human"	600700337233	TBD "K as HOIL High Tools for Human"	000000041
	H	TOO Kas. HOU, High Techtoi Huillan	000700220321	TOO Kas Hou High rechtor Hullian	
, телефон: 749212 Закрыть <u>О</u> снования Псправочник: Контраген	Pac	нетные счета Поиск <u>В</u> заиморасчет	ы		
Для получения подсказки нажмите	F1		NU	И ТА: 10.11.06 14:21:37 БИ: 4 квартал 2006 г. ТП:	
🖞 Пуск 🍏 🍙 🗿 » 🔄 К	нига	Конфигуратор - Бух	🖑 1С:Предприя	гие - Т 🖻 8Глава 6 - Microsoft 🛛 😰 🦿 🔍 😇 👬 🌾 🗃	19:05
	_				

Рис. 6.1. Форма списка справочника "Контрагенты"

Сведения о контрагентах однозначно понимаются всеми пользователями системы. Если уполномоченный пользователь (т. е. имеющий соответствующие права в системе) внес нового контрагента, то все понимают, что это именно "ОАО СтройСервис", а не "ЧП СтройСервис", и что это совершенно разные организации. Попутно отметим, что, так как ввод нормативносправочной информации осуществляется самими пользователями на этапе эксплуатации конфигурации, то сами пользователи и должны следить за тем, чтобы информация не дублировалась, была удобной для поиска. Например, лучше вносить данные как "СтройСервис ОАО" и "СтройСервис ЧП". "СтройСервисов" не так много, как ОАО и ЧП, поэтому поиск будет значительно проще. Для работы с подобного рода информацией используется объект метаданных Справочники.

В открытом окне конфигурации найдите ветвь метаданных Справочники. Щелкните левой кнопкой мыши на ней, и она будет выделена, как на рис. 6.2. Затем щелкните правой кнопкой мыши. Откроется контекстное меню, в котором надо выбрать пункт **Новый справочник**. Откроется первое окно конструктора справочника (внизу этого окна установлен флажок **Использовать** конструкторы для создания новых объектов. Не снимайте его.) Прочитайте подсказку в центре окна конструктора и заполните поля справочника следующими значениями:

- **Идентификатор** Номенклатура;
- **Синоним** Справочник номенклатуры;
- **С Комментарий** Справочник ТМЦ.

🚱 Конфигуратор - Бух и опер учет, расчет - [Конфигу	рация]	_ 8 ×
📲 Файл Действия Конфигурация Конструкторы Адм	инистрирование Сервис Окна Помощь	
]ª₩⊼₩४४० ]\$#EE	· 哈哈泽兰东海站 · 西西海南·	
⊡ - П Учебная Компл учет П - П - П - П - П - П - П - П - П - П -		
🗉 🚰 Справочники		
<ul> <li></li></ul>	Конструктор Справочника	
н	Идентификатор: Новый1	
🗄 📄 Обработки	Синонимс	
на траны Счетов на 📠 Виды Субконто	Комментарий:	
⊞-1 <u>ра</u> Операция ⊞-10 <sub>к</sub> Проводка		
на прегистры При Прегистры		
на на расчетов	Обязательно икажите илентиликатор справлиника. Поде	
Электрины расчетов Н	"Синонии" может быть заполнено для представления вида сплавленика в режиме заполнено для представления вида	
	"комментарий" можно ввести поясняющую информацию.	
	Использовать конструкторы для создания новых объектов	
	КНазад. Далее» Готово Отмена Помощь	
I Торава Права Права		
та Конфигурация		
Для получения подсказки нажиите F1	NUM (СТАТУС: БД , ДОК	умент
🟦 Пуск 🧃 🔄 💿 🎽 🏠 Главаб 🥰 Кон	ригурато 🖻 8 Глава 6 - Міст 🖉 Документ 1 - Мі 🕎 Paint Shop Pro 🛛 🔃 🖓 ኛ 🔍 🔁 🗃 K 🗃 🖳	19:07

Рис. 6.2. Конструктор справочника

#### Замечание

Понятия "идентификатор", "синоним", "комментарий" полностью соответствуют понятиям, описанным для констант, и применимы ко всем видам метаданных.

Следующий шаг — подготовка справочников для ввода нормативносправочной информации. Самый первый и необходимый справочник для торговой деятельности — справочник товаров, в более широком смысле, номенклатуры. Приступим к созданию справочника "Номенклатура".

В этом справочнике должны храниться данные о товарах или услугах, в том числе:

🛛 Код;

□ Краткое наименование;

□ Полное наименование;

□ Тип номенклатуры (товар, услуга);

Тип товара (собственный, покупной);

□ Единица измерения;

□ Себестоимость;

🗖 Цена;

□ Ставка НДС.

Для простоты считаем, что учет и расчеты ведутся только в национальной валюте.

Нажмите кнопку Далее. В следующем окне поставьте флажок в окне Создать новый вид субконто, если "1С:Предприятие" на компьютере имеет компоненту "Бухгалтерский учет". Нажмите кнопку Далее, вставьте справочник в пользовательский интерфейс (у нас пока только интерфейс Администратора). Нажмите кнопку Далее. Если имеется компонента "Бухгалтерский учет", заполните поля Синоним и Комментарий к субконто Номенклатура. Нажмите кнопку Готово.

#### Предупреждение

Вид метаданных Субконто рассматривается в главе 10.

В результате раскрывается окно редактирования справочника (рис. 6.3).

В окне выводятся и могут редактироваться идентификатор, синоним и комментарий справочника. Назначение поля **Подчинен** опишем позже. В поле **Количество уровней** выставляется количество уровней вложенности. Например, номенклатура товаров — оргтехника. Тогда для удобного внесения и поиска информации в справочнике можно создать уровень 1 с папками (группами) **Компьютеры**, **Ксероксы**. Далее, в группе **Компьютеры** создать уровень 2 — **Бренд**, **Самосбор** и т. п. (рис. 6.4). Из опыта известно, что пользователю оптимально работать со справочниками, содержащими 3—5 уровней.

🚰 Конфигуратор - Бух и опер учет, расчет -	[Справочник Новый1]	
🎒 Файл Конфигурация Конструкторы Админ	нистрирование Сервис Окна Помощь	. B ×
12 <b>- 1</b> × 6 6 4 5 1 5		
_ 68 (\$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$	· · · · · · · · · · · · · · · · · · ·	
Идентификатор: Новый1	Подчинен: (не подчинено)	
<u>К</u> омментарий:	Синонимс	
Кол-во, уровней: 1 😑	Резмещать группы сверху 🔽	
Длина кода: 🗧 芸	Автоматическая нумерация 🔽	
Длина наименования: 25 🚍	Контроль уникальности 🔽	
Серии кодов	Реквизиты	
С Во всем справочнике	t .	
• В пределах подчинения		
Tun koga		
С Текстовый		
Основное представление		
С В виде кода		
В виде наименования	Новый Изменить Удалить	
🔲 Одна форма для элемента и группы	Редактировать: В списке	
Плисание Форма з	аемента Фолма плилы Фолмы списка	
	Tenerua Lobud Thames Tobud Charles	
та Конфигурация * <b>Ду Справочник</b>	Новый1	
Для получения подсказки нажиите F1	NUM (СТАТУС: БД *, Докун	чент *
🕅 Пуск 🚿 🕞 🕞 » 🕞 Главаб	🕊 Конфигурато 🗟 В Лава 6 - Міст 🗟 Документі - Мі 📅 Раілі Shop Pro 🛛 🔟 🦿 🕊 🥵 🖀 🕮 🦄	19:08

Рис. 6.3. Окно редактирования справочника



Рис. 6.4. Схема уровней справочника

Если количество уровней более 1, активным становится поле с флажком **Размещать группы сверху**. Установим этот флажок так, чтобы названия групп справочника в любых режимах просмотра были сверху, что удобно для пользователя.

В справочнике всегда по умолчанию имеется два реквизита — Код и Наименование. Поля Длина кода и Длина наименования указывают количество знаков, из которых может состоять запись в этих полях. Поле Код необходимо для автоматического маркирования каждой записи в справочнике. Во встроенном языке "1С:Предприятие 7.7" есть системные функции для работы с полями Код и Наименование.

Длину кода оставим без изменения (5 символов). Длину наименования поставим 25 символов. Максимально возможная длина наименования — 100 символов. Тип значения наименования всегда "Строка". Тип значения кода может быть как "Строка", так и "Число". Тип значения кода осуществляется переключателем Тип кода. Включение опции Автоматическая нумерация устанавливает, что вновь введенному элементу в справочнике код будет присваиваться автоматически. Автоматически присвоенный код можно исправить вручную. Если установлен флажок Контроль уникальности, проверка кода на уникальность будет проводиться автоматически при вводе в справочник нового элемента. В этом случае невозможно присвоение одного и того же кода двум разным элементам справочника. Переключатель Серии кодов необходим для того, чтобы указать диапазон проверки кодов на уникальность. Если выбрана установка Во всем справочнике, то уникальность кода будет проверяться среди всех записей справочника. При выборе установки В пределах подчинения уникальность будет проверяться только в пределах группы, в которую помещается или в которой находится элемент. При этом необходимо учитывать, что одинаковые коды могут быть у элементов справочника, находящихся в разных группах. Переключатели Основное представление указывают, значение какого реквизита будет выведено при обращении к справочнику по умолчанию — Код или Наименование. Оставьте основное представление в виде наименования.

Установите флажки Автоматическая нумерация и Контроль уникальности.

Теперь добавим в справочник недостающие реквизиты. Работая в пользовательском режиме, вы, наверное, обратили внимание, что справочники могут быть представлены в виде таблицы. В этой таблице, как правило, присутствуют всегда две колонки: Код и Наименование. Это и есть реквизиты. Об этих двух реквизитах мы говорили, что они присутствуют по умолчанию. Дополнительные колонки таблицы — это дополнительные реквизиты справочника. Всевозможные свойства этих колонок мы описываем в окне редактирования справочника в разделе Реквизиты:

- □ Код этот реквизит уже существует по умолчанию, его свойства мы задали ранее;
- □ Краткое наименование его вполне можно поместить в Наименование, свойства которого мы задали ранее. Для Полного наименования мы должны создать новый реквизит.

Нажмите кнопку Новый — появится палитра свойств реквизита (рис. 6.5).

Палитра свойств реквизита очень похожа на палитру свойств константы, которую мы изучили. Следует подчеркнуть, что свойства объектов метаданных, приемы работы в конфигураторе и приемы программирования в значительной степени построены на аналогиях. Это позволяет экономить время изучения конфигурирования и программирования в среде "1С:Предприятие".

Заполните общие свойства данного реквизита. Идентификатор — ПолноеНаименование, Синоним — Полное наименование, Комментарий — Полное наименование. Тип значения — "Строка". Длину установите 100 символов. На вкладке Дополнительные ничего не устанавливайте.

🚰 Конфигуратор - Бух и опер учет, расчет - [Сп	равочник Номенк	латура]					_ <i>8</i> ×
🔊 Файл Конфигурация Конструкторы Админист	грирование Сервис	: Окна Помощь					_ <u>-</u>
	691	二百姓的	8 8				
<u> </u> 48  94 X 14  63 © 0    \$2   € €		· · · · · · · · · · · · · · · · · · ·	<b>治治增济</b> 法				
Идентификатор: Номенклатура	Подчинен: [	(не подчинено)	•				
Комментарий: Справочник номенклатуры	Еиноним: [	Справочник номенкла	туры				
Кол-во, уровней: 3 👘	<u>Р</u> азмещать группы	сверху 🔽					
Длина кода: 5 \Xi	Автоматическая ну	мерация 🔽					
Длина наименования: 25 💼	Контроль уникальн	ости 🔽					
Серии кодов	Реквизиты						
Во всем справочнике     В пределах подшинения	ПолноеНаименова Свойства Рекви	ание зита		×			
	Общие Лопол	нительные		اط اھ			
• Числовой	Илентификатог	п. ПодноеНаименова	HHR				
О Текстовый	Синоним:	Полное наименов	зние				
Основное представление	Комментарий:	Полное наименов	ание				
О В виде кода	Тип значения:	(<Строка>>	<b>T</b>				
• в виде наименования	Длина:	100 - Heorp.	Точность: 0 🚓				
🗌 Одна форма для элемента и группы 🛛 🖻	24			Обновить			
<u>О</u> писание Форма <u>э</u> лем	ента Форма гр	уппы 🖉 ормы спис	жа 🕨				
<u>та Конфигурация</u> Справочник Нов	менклат						
для получения подсказки нажмите F1	10	Discourse of Mary	The Desire Land Date			lalé	илим рстатус: БД , Документ
	Книга	er or hasa b - micr	Bill Pallic Shop Pro	) 🖏 конфигура	ато 🛚 🖳		· · · · · · · · · · · · · · · · · · ·

Рис. 6.5. Палитра свойств реквизита справочника

Теперь создадим остальные реквизиты согласно описанию справочника, данному в начале раздела. Временно реквизиты Тип номенклатуры, Тип товара, Единицы измерения будут иметь тип значения "Строка". Реквизиты Себестоимость, Цена, Ставка НДС имеют тип значения "Число". Реквизит Ставка НДС имеет длину 6, точность — 2 символа. Это связано с тем, что

ставка НДС указывается в процентах. Три знака резервируются на цифры от 0 до 100, один — для десятичного знака, два — для десятичной части.

В нижнем правом углу окна редактирования справочника находится кнопка Форма списка. Нажмите ее, в открывшемся меню выберите пункт Форма списка. Откроется окно Размещение реквизитов Справочника (рис. 6.6). Флажки установлены напротив тех реквизитов, которые будут видны в справочнике в пользовательском режиме. Если с некоторых реквизитов флажки снять, они не появятся в пользовательском режиме, но данные в них можно вводить (например, через команды встроенного языка) и получать. Нажмите кнопку Вставить. Сформируется шаблон экранной формы списка справочника "Номенклатура".



Рис. 6.6. Диалоговое окно Размещение реквизитов Справочника

Выполните запуск системы "1С:Предприятие" в пользовательском режиме. Выберите пункты меню Операция | Справочники, в окне Выбор справочника выберите строчку Номенклатура и нажмите кнопку ОК. Справочник откроется, введите в него несколько элементов. Они пригодятся для последующей работы.

Обратите внимание, что заголовки колонок — синонимы реквизитов. Если синонима по какой-то причине нет, то заголовком колонки будет идентификатор реквизита. В данном случае нет синонимов у реквизитов **Код** и **На-именование**.

Выйдем из пользовательского режима и вернемся в конфигуратор.

В форме списка справочника щелкните левой кнопкой мыши по колонке **Наименование**. Когда колонка будет выделена, вызовите контекстное меню и выберите пункт **Свойства**. Откроется палитра свойств поля ввода. Колонка для ввода информации называется так, в отличие от реквизита, где описывались как бы внутренние свойства этой колонки (рис. 6.7). Напишите заголовок **Краткое наименование**. Запустите "1С:Предприятие" в пользовательском режиме. Если все сделано правильно, экранная форма списка справочника "Номенклатура" будет выглядеть, как на рис. 6.8.

Ставка НДС, как и ставки любых налогов, может периодически изменяться. То же касается и цены реализации товара. Аналогичные проблемы (о периодических константах) уже обсуждались в *разд. 2.2* — они решаются установлением свойства **Периодический** реквизитам **Цена** и **СтавкаНДС** в окне свойств реквизита на вкладке **Дополнительно**.

🛱 Номенн	слатура	-		
Код	Краткое наименование			
	Сво	c .		
		:		
		l.		
		3		
Exc				
Печа	ть			

Рис. 6.7. Палитра свойств поля ввода

100 I I	Продари	атио - Бим и		THAT													
Файл	Анре-Апри Лействии	апистоужи Операции	Справочники	Документи	и Журналы	Результаты расчет	a 07	четы	Обработ	си Сег	вис О	кна П	омощь				
11 399	~ <b>m</b>	V Pro Pa			a.		 	ຄ		া দেটে। ভা দেটে।		M ML					
		~ 48 <b>U</b> ⊴			a)	ex	849	8			V2		III.	_	_	_	
<u>ق</u> ا	оменклат	ypa (28.11.	06)				l										
2	ù 🖉 🤉	9 4 4	Te 🗞 📭 🗈		) <i>P</i> 3												
	Код	Краткое на	именование	Eg	и Тип но	менклатуры 🔺											
	2	тов2		_													
	<u>'</u>	товарт															
ll⊢	- 1																
ll⊢					-												
ll⊢				-													
	• •				-												
<u> </u>																	
ிர	оменклат	ypa (28.11															
Для п	олучения п	одсказки наж	мите F1					-	NUM TA:	29.09.06	5 15:29:	36	БИ: «	квартал 200	5r.	ТП: Октябрь	2006 r.
â9⊓;	/ск 🔊 (	o o » c	2 Проводник	- W 2	Microsoft O	→ 🚟 Paint Shop Pro		Kor	нфигурат	эр [Г	<u>к</u> 1С:П	редпр	ия	RU (?) 🖗	« <u>9</u> 29	II K 🗆 🗉	16:33

Рис. 6.8. Экранная форма списка справочника "Номенклатура"

### 6.1.1. Форма элемента

Вернемся в режим конфигуратора, в окно редактирования справочника. До сих пор мы использовали только так называемую форму списка, т. е. вводили и редактировали данные непосредственно в ячейках таблицы справочника. Но во многих справочниках одновременно работают только с одним элементом. При этом не обязательно видеть остальные. Если таблица справочника имеет много колонок, то работать с ней неудобно — требуется горизонтально прокручивать окно. В таком случае можно использовать форму элемента справочника. Для ее вызова в окне редактирования справочника необходимо нажать кнопку **Форма элемента** (рис. 6.9).

Вставить необходимые реквизиты в форму элемента можно с помощью такого же диалога размещения реквизитов справочника, которые использовали ранее для формирования экранной формы списка (рис. 6.10). Если все сделано правильно, то окно редактирования формы элемента должно выглядеть, как на рис. 6.11. Только на рисунке для наглядности присутствует и диалог размещения реквизитов справочника.

В информационных полях диалога можно вывести разнообразную дополнительную информацию из других справочников, регистров, журналов и т. д.

Теперь в окне редактирования справочника можно в списке **Редактировать** выбрать значение **Обоими способами**. Запустим "1С:Предприятие" в пользовательском режиме.



Рис. 6.9. Вызов формы элемента справочника

Откройте экранную форму справочника. Щелкните меню Действия | Редактировать в диалоге. Затем дважды щелкните по какой-нибудь строке (элементу) в экранной форме справочника. Откроется именно форма элемента (рис. 6.12). Если закрыть форму элемента и вновь просмотреть меню Действия, то напротив пункта Редактировать в диалоге будет установлен флажок.

	_
🚰 Конфигуратор - Бух и опер учет, расчет - [Форма эленента-Справочник.Номенклатура (Справочник номенклатуры)] 📃	8 ×
🕒 Файл Действия Диалог Вставить Конфигурация Конструкторы Администрирование Сервис Окна Помощь	리의
☆ 🖬 🗟 🕾 🗷 ⇒ ⊂ 桷 🔽 ⊻ 糸 糸 💡	
· `` `` `` `` `` `` `` `` `` `` `` `` ``	
🛱 Нонсински Разнещение реквизитов Справочника 🛛 🗖 🗖 🖾	
Hareso     Percensaria       Unamenoidade acenceariae     Unamenoidade acenceariae       Unamenoidade acenceariae     Unamenoidade acencea	
Вставлять имя Вставлята Поле ввода	
Разместить в диалоге автоматически	
🔽 Вызывать для новых форм	
Встренить Отмена Помощь	
\СЭДналог / ШМодуль ) МТаблица /	
[『 레 〒 보   ◆ ◆   河 王 岡   ■   】 ● 【 ■   】 ● 【 ■   】 ● 】 ● 】	
а∰Конфигурация * 👔 Справочник Номенклатура 🔲 Форма элемента-Спра	
Для получения подсказки нажинте F1 NUM 19 x 15 63, 26 СТАТУС: БД , Докумен	т*
🔊 Пуск 🧑 🔄 🕙 🖹 2 Проводник 🔹 📝 2 Містозоft О • 🔡 Paint Shop Pro 🛛 🦿 Конфитурато 🚱 CorelDRAW 10 🔤 🕐 🗳 « Б Э Э К 🖓 🕿 🖓 👘 1	6:40

Рис. 6.10. Конструирование формы элемента справочника



Рис. 6.11. Окно редактирования формы элемента справочника

憖口 ГПлеяллиятие - Бухи и олен учет пасчет	₽ ×
стана редирилить Онгрании Справочники документы Журналы Результаты расчета Отчеты Обработки Сервис Окна Покощь Файл дейских Операции Справочники документы Журналы Результаты расчета Отчеты Обработки Сервис Окна Покощь	
1.2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
Поменклатура (28.11.06)	
19 1 1 2 2 4 4 4 4 5 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1	
Код Краткое наименование Еди Гип номенклатуры	
<u>аў Номенклатура : товар1 (28.11.06)</u>	
Наименование: товар1	
Полное наименование:	
Тип номенклатуры Тип товара	
Единица измерения.	
Себестоиность: 0.00 Ш Цена: 0.00 Ш	
ОК Закрыть	

Рис. 6.12. Форма элемента справочника в пользовательском режиме

Пока этот флажок присутствует в меню, редактирование записей будет происходить только в форме элемента. Если снять флажок (снова щелкнуть по пункту **Редактировать в диалоге**), редактирование записей будет происходить в форме списка.

#### Примечание

Для редактирования записи справочника только в форме списка или только в форме элемента необходимо в поле **Редактировать** окна редактирования справочника выбрать пункт **В списке** либо **В диалоге**.

Форма группы аналогична форме элемента, но предназначена для создания дизайна диалогового окна названия групп справочника. Часто она полностью соответствует форме элемента, если в окне редактирования справочника установлен флажок **Одна форма для элемента и группы**.

## 6.1.2. Подчиненные справочники

Посмотрите еще раз на рис. 6.1. Каждой записи в поле **Наименование** соответствует одна запись в каком-либо реквизите — **Адрес**, **ИНН** и т. д. Такое соответствие в информационной базе называется *отношение один-к-одному*. Другое дело, если каждому наименованию соответствуют несколько записей в реквизите. Например, с контрагентом может быть заключено насколько договоров на поставку или получение товаров и услуг. Такое соответствие называется отношение *один-ко-многим*. Что делать в этом случае? Вносить новую запись с одним и тем же контрагентом при подписании каждого нового договора? Это сильно затруднит работу со справочником. Ведь в различных случаях могут быть десятки записей в реквизите, относящиеся к одному. Для решения подобных проблем в системе "1С:Предприятие" существует понятие подчиненного справочника. В подчиненном справочнике хранится множество записей, относящихся к одной записи справочника-владельца (рис. 6.13).

В процессе конфигурирования одним из реквизитов справочника-владельца устанавливается подчиненный справочник, из множества значений которого пользователь может выбрать нужное в данный момент.



Рис. 6.13. Отношение один-ко-многим

Подчиненный справочник создается так же, как и обычный, только в поле **Подчинен** выбирается ранее созданный справочник-владелец. У неподчиненных справочников в этом поле установлено значение **Не подчинено**.

# 6.2. Понятие о выборке

Для отображения данных пользователя в печатной форме мы должны средствами встроенного языка сформировать отчет, который система уже сможет напечатать. Обычно алгоритм печати сводится к организации *выборки*. Выборку можно себе представить, как некоторое множество записей, подлежащих обработке. Это множество может включать в себя все записи таблицы из базы данных, какую-то часть записей, одну запись или даже ни одной. В последнем случае говорят, что выборка пуста. Система "1С:Предприятие" не имеет интерактивных средств выборки, т. е. выборки вручную. Поэтому задача печати данных фактически распадается на две:

организацию выборки и обработку записей;

🗖 отображение информации каждой записи в отчете.

Для того чтобы организовать выборку из объекта метаданных, надо вначале создать в оперативной памяти компьютера ссылку на этот объект (например, справочник). Для этого в тексте модуля необходимо написать обращение к системной функции СоздатьОбъект() и присвоить возвращаемое ей значение некоторой переменной:

```
об=СоздатьОбъект ("ИмяТипаАгрегатногоОбъекта");
```

Объект существует в оперативной памяти до тех пор, пока на него есть хотя бы одна ссылка. Как только переменная, содержащая ссылку, исчезнет, система сама автоматически удалит объект из оперативной памяти. Если по каким-либо причинам объект не нужен, то удалить его можно следующим образом:

об=0**;** 

Для объектов, которые имеют структуру хранения в информационной базе, обращаться к атрибутам объекта можно только после того, как объект будет позиционирован. До этого момента доступ к атрибутам будет вызывать сообщение об ошибке.

#### Примечание

Атрибуты описывают количественные или качественные характеристики объекта. Например, при обращении к справочнику **Код** и **Наименование** будут его атрибутами. При позиционировании данные, которые находятся на диске, переносятся в атрибуты объекта, расположенного в оперативной памяти. В информационной базе есть методы, отвечающие за позиционирование объектов. Список методов у каждого из объектов свой. Методы позиционирования указывают на определенную строку в записях. Реально это можно представить следующим образом: откройте любой справочник и выберите какую-нибудь строку. Вы позиционировали справочник вручную (рис. 6.14), а наша задача — написать программу, которая позиционировалась бы на записях справочника автоматически.

Вначале выборку нужно объявить:

об.ВыбратьЧто-то (...);

Метод возвращает 1, если в выборке есть хотя бы одна запись. Затем в цикле осуществляется вызов метода, который извлекает из выборки очередную запись и позиционирует на ней объект:

об.ПолучитьЧто-то (...);

1С:Предприятие - Бужгалт Файл Действия Операции	ерский учет для Казахстана, ред. 2.0: ТОС 1 Справочники Документы Журналы Отчи	<b>) "Авто - Альянс" - [Сп</b> еты Налоговый учет С	равочник:Контрагенты] ервис Окна Помощь	_ 8 ×
- *) =    X % (%) (*	5 & R > C M	- A A ?	₩ 🗭 💭 M M+ M-	
© III 7 2 4 #				
	* • • • • • • • *			
🗉 🖂 Контрагенты	Наименование	PHH	Полное наименование	Код 🔺
	① 0A0, 3A0			00027
	主 Организации			00001
	🔃 Пенсионные фонды			00002
	100 TOO			00026
	🔁 Учредители			00010
	Настные лица		-	00012
	І осударственный фонд социального стра и осударственный фонд социального стра	ахов 600400073391	I осударственный фонд социального страхования	00034
	Казэнерго филиал корпорации	181600400479	Филиал Корпорации "КазЭнерго"	00008
	Налоговый комитет по г.Астана	63010000005	Налоговый комитет по г.Астана	00022
	Налоговый комитет по г.9сть-Каменогор	оску 18160000011	Налоговый комитет по г.9сть-Каменогорску	00030
	Сибирь-Софт ООО	5018026285	Общество с ограниченной ответственностью "Сибирь-Софт"	00020
	Строительная фирма Россия	0221213333	Строительная фирма Россия	00023
	Тураналем	602301145741	вко чодо ванктураналем	00015
		1		
	Адрес: г.Астана, пр.Республики, 2			
Закрыть Основания	Расчетные счета Печать списка Ана	лиз сибконто "Контрасент	er [	
		and egatoring interruptioning	<u></u>	
genpasounate ton i pai en				
я получения подсказки нажиит	re F1	NL	М ТА: 01.01.2002 00:00:00 БИ: 4 квартал 2006 г. ТП:	
Пуск 🥥 🕞 🕑 » 🗀 2	2 Проводник 🔹 📝 2 Microsoft О 🔹 🧮 Ра	aint Shop Pro 🛛 🖓 3 1C	🗤 start 🎯 CorelDRAW 10 🔃 🕐 🗘 < 🙁 💷 🗷 🕷	🖳 🗃 🏟 🛅 16:58

Рис. 6.14. Выделенная строка (позиционирование) справочника

Метод также возвращает 1, если запись есть, и на нее удалось сослаться. Поэтому обычно цикл выборки выглядит следующим образом (рис. 6.14):

Если Об.ВыбратьЧто-то (...) =1 Тогда Пока Об. Об.ПолучитьЧто-то (...) =1 Цикл // объект позиционирован КонецЦикла;

# 6.3. Объект *Таблица* для построения отчета

Объект таблица служебный, поэтому его в дереве метаданных нет. В системе "1С:Предприятие" версии 7.7 этот объект можно использовать в двух режимах:

**П** для построения отчета;

для организации ввода информации.



Рис. 6.15. Открытие окна общих таблиц

В этой книге рассмотрим режим для построения отчета.

При работе с таблицей для построения отчета сначала нужно создать средствами табличного редактора шаблон печатной формы объекта метаданных. Большинство (но не все) видов объектов метаданных может иметь в своем составе шаблон печатной формы, который можно использовать для вывода на печать информации, содержащейся в объекте. В таких случаях говорят, что объект может иметь в своем составе шаблон печатной формы, принадлежащий локальному контексту текущего объекта. Справочник имеет шаблон печатной формы, с помощью которого можно формировать отчеты, относящиеся к данному справочнику.

#### Замечание

В тех случаях, когда локальный контекст модуля, в котором размещена процедура формирования отчета, не может содержать шаблон печатной формы, шаблон размещают в *глобальном контексте*, т. е. в общих таблицах. Для того чтобы открыть окно, содержащее общие таблицы, необходимо в окне конфигурации щелкнуть правой кнопкой мыши на объекте **Конфигурация** и выбрать в контекстно-зависимом меню пункт **Общие таблицы** (рис. 6.15). В результате будет открыто окно табличного редактора.

Посмотрим шаблон печатной формы справочника "Номенклатура". Для этого откроем в конфигураторе форму списка справочника и щелкнем по вкладке **Таблица**. В результате будет открыто окно табличного редактора. Визуально объект Таблица похож на электронную таблицу Microsoft Excel (рис. 6.16). Те же строки, колонки, строка формул и окно именования ячеек.

Однако различий гораздо больше, чем сходства. Таблицы в системе "1С:Предприятие" используются, в основном, для представления уже обработанной информации и чаще всего заполняются с помощью программных модулей на встроенном языке системы.

#### Замечание

В практической работе можно копировать таблицу с данными в буфер обмена и затем вставлять ее в MS Excel. Есть и специальные программы — конвертеры, передающие таблицы "1С:Предприятие" в MS Excel. Их преимущество по сравнению с простым копированием — грамотное форматирование передаваемых данных. Существуют и обратные конвертеры — из MS Excel в "1С:Предприятие". Однако такие конвертеры переносят только форму таблиц, без данных.

Элементарным элементом шаблона печатной формы является *ячейка*. Ячейка в шаблоне печатной формы имеет свою палитру свойств, которую можно вызвать стандартным способом — щелкнув по ячейке правой кнопкой мыши и выбрав из появившегося меню пункт **Свойства** (рис. 6.16). Рассмотрим некоторые свойства ячейки.

1	r 🔒	አ ቤ 🛍	60	1) 1) 1) 1)	#4		•	8 2 <b>9</b>								
₽/ <sup>8</sup>	94 X	°n ≪3 ≪	0		ĒĒ	×	in in di	1 1 1 1 1 1	1 第 4 4 1 1	治癒落	Ň					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1																
2																
3																-
4		_														
с а																
7																
8																
9																
10																_
11						Свойства яч	нейки					×				_
12						Toward	1		1			. 1				
14							ложение	шрифт   Рам	kalasob l							
15						Тип Текс	т 💽	<ul> <li>Контроль</li> </ul>	Авто	•						
16						_			·							
17										<u></u>						
18										-	OK					_
19						Paculation	OPY B			07540	- I					
20												10				
21						🔽 Защита						пь				
23					-		1									
24																
25																
26																_
27																
28																
30						-										
31																<u> </u>
32																
۹î.	1															ÞÉ
. EJ.	Диалог	). 🖺 Mi	одуль	). Al Kara	алог	). 🔊 Табли	ща1									

Рис. 6.16. Шаблон печатной формы справочника

Вкладка **Текст**. Текстом в данном случае мы будем называть любую комбинацию символов — буквы, цифры, знаки препинания, пробелы и служебные символы. Вводить и редактировать текст в ячейку можно в многострочном поле этой вкладки.

Важным является список поля Тип. Пункты этого списка устанавливают, какого рода информация введена в ячейку:

- □ **Текст** информация в ячейке в пользовательском режиме будет представлена так же, как введена в конфигураторе;
- □ Выражение информация в ячейке это выражение на встроенном языке. В пользовательском режиме вы увидите результаты вычисления этого выражения;
- Шаблон в ячейку вводится текст с включенными в него выражениями на встроенном языке. Выражения заключаются в квадратные скобки. В пользовательском режиме будут выводиться текст и результат вычисления выражений;
- □ Фиксированный шаблон место для вывода результатов выражений определяется пользователем с помощью расстояния между квадратными скобками, ограничивающими выражение в исходном тексте.

Вкладки Положение, Шрифт, Рамка, Узор служат для оформления отчета.

Поскольку практически все деловые документы имеют прямоугольную структуру, шаблон печатной формы представляет собой совокупность прямоугольных областей, каждая из которых служит для выдачи части готового отчета: заголовочной, табличной и других. Такие области называются *секциями*. Любой секции отчета можно присвоить уникальное имя и в дальнейшем ссылаться на поименованную область. Секции бывают горизонтальные и вертикальные. Они могут включать в себя несколько строк или колонок, но никак не меньше одной.

Сам процесс построения отчета происходит следующим образом. Сначала строится шаблон печатной формы из ячеек, строк, столбцов, секций. Одновременно пишется процедура, в которой реализуется алгоритм обработки шаблона печатной формы. В процессе выполнения процедуры из шаблона в нужном порядке извлекаются секции, транслируются (вместо имен переменных подставляются их значения) и копируются в таблицу готового отчета.

## 6.4. Перечисление

Подчеркнем еще раз, что при создании справочника в конфигураторе создается только шаблон для записи данных. Пользователь сам вносит данные и может ввести в справочник практически неограниченное количество записей.

Иногда нужно сделать выбор из конечного числа значений, причем не изменяя состав списка. Для этого следует создать Перечисление. В отличие от справочника, значения в перечислениях принадлежат конфигурации, поэтому пользователь в процессе своей работы не может изменить их состав. Зато список значений перечисления задается программистом, и он может использовать эти значения для проверки выбранного значения и описать действия, которые должны произойти после этого выбора.

Так же, в отличие от справочника, перечисление не может иметь несколько уровней. Все его значения находятся на одном уровне.

Перечисления удобно использовать в случаях, когда значения не будут меняться в будущем или требуется единообразие ввода. Например, перечисления Да, Нет, Юридическое лицо, Физическое лицо или группы основных средств. В последнем случае подразумевается, что группы основных средств не меняются в течение долгого времени. Желательно, чтобы в наименованиях групп основных средств соблюдалось единообразие. Ведь если наименование группы Здания один оператор внесет как "Здние", а другой — "Здание", то одна и та же группа в системе будет восприниматься уже как три, т. е. произойдет искажение учета. Неправильно использовать перечисления для изменяемых значений, например, для ставки НДС. При изменении в соответствии с законодательством значений ставки НДС, особенно в середине года, могут возникнуть серьезные проблемы по корректировке алгоритма программы. Гораздо удобнее использовать алгоритмы обработки справочника. Например, создать специальный справочник Ставкиндс. При любом изменении значений ставки их можно просто дописать в справочник самому пользователю.

## 6.4.1. Создание перечислений

На этом этапе мы разберем порядок создания перечислений, так как по логике нашей конфигурации они тесно связаны со справочниками, а именно являются составной частью некоторых их реквизитов. Самый простой пример — подготовка перечисления для будущего справочника "Контрагенты". В перечислении будет содержаться информация, является ли контрагент резидентом.

В Конфигураторе выберем ветвь метаданных Перечисление и создадим новое перечисление. В окне редактирования введем идентификатор — **Резидент**, синоним — **Резидент**, комментарий — Признак резидента. Нажмите кнопку **Новый** для ввода значений перечисления (рис. 6.17).

Заполните поле идентификатора — **Резидент** и поле представления — **Резидент**. Обратите внимание: в поле **Представление** вводится значение в том виде, в котором его будет видеть пользователь. В данном случае идентификатор и представление значения перечисления совпадают.

Нажмите кнопку **ОК** в диалоге **Свойства значения**. Идентификатор значения появится в поле диалога **Значения**. Добавьте новые значения:

#### Идентификатор — НеРезидент;

#### Представление — Не резидент.

В данном случае написание идентификатора и представления разное. Написание представления удобно для пользователя.

Следующее необходимое нам перечисление ЕдиницыИзмерения. Это перечисление понадобится в справочниках "Материалы" и "Номенклатура". Создайте новое перечисление. В окне редактирования введем идентификатор — **ЕдиницыИзмерения**, синоним — **Единицы измерения**, комментарий не нужен. Значения перечисления — это единицы измерения товаров и материалов. Создайте два новых значения: Штука с представлением Шт и Упаковка с представлением Уп. Вообще, данное перечисление формируется, исходя из специфики деятельности предприятия.
Перечисление ГруппыОС необходимо для справочника "Основные средства". Создайте это перечисление так же, как и два предыдущих. Значения следующие: Земля, Здания, Транспорт, Оборудование. Представления совпадают с идентификаторами.

2004	
Canonym yparup - byk konep ywer, parter - freprekonenike reskjent j Canonym yparup - byk konep ywer, parter - freprekonenike reskjent j Canonym yparup - byk konep yr canonym sector (chan Down)	
·····································	
Идентичикатор: Резидент Диноник. Резидент	
Комментариях Признак резидентства	
Значения	
Peswaerr 1	
Свойства Значения 🛛	
Общие 9 да	
Идентификатор: Резидент	
На Коминентарийс ОК	
Представление: Розмант Плана	
Оновить	
т. Жонфигурация * ПП Перечисление Резидент	
Для получения подсказки нажините F1 ЛИМ СТАТУС: БД. Доку	нент *
街 Пуск 🧑 🔄 \Theta 🔌 🗋 2. Проводник 🔹 🕅 2. Містозоft О 🛛 📅 Paint Shop Pro 🛛 🌾 2. 1 С.¥7 start 🎯 CorelDRAW 10 🗖 😨 🖗 🤹 🧃 🗐 🕷 🖓 🖙 🖓 🗑	j 17:05

Рис. 6.17. Ввод значений перечисления

Перечисление Затраты будет использоваться в дальнейшем для бухгалтерской аналитики по статьям затрат. Значения этого перечисления **Зарплата**, **ХозНужды, Износ, Себестоимость товаров, Материалы, Административные**.

# 6.5. Печатные формы справочников

Информация в системе "1С:Предприятие" может быть представлена не только в виде экранных форм, но и таблиц, которые можно выводить на принтер. Ведь часто так удобней работать, да и учетные документы требуется предоставлять заказчикам в распечатанном на бумаге виде. В "1С:Предприятии" имеется конструктор печати, наподобие конструктора справочников или документов. Конструктор позволяет создавать основы шаблонов печатных форм и прототипы алгоритмов печати. Однако их, как правило, приходится дорабатывать. Для понимания процесса полезно будет ознакомиться с созданием печатной формы и алгоритма печати вручную. С конструктором печати мы ознакомимся в *разд.* 7.5.

С основными принципами вывода информации в распечатанном виде мы познакомимся на примере создания печатной формы справочника.

Продолжим работу с вкладкой **Таблица** формы списка справочника "Номенклатура". Заметим, что название служебного объекта метаданных Таблица и шаблона печатной формы по умолчанию **Таблица** совпадают. Для того чтобы не путать их, переименуем шаблон печатной формы. Для этого щелкнем правой кнопкой мыши по вкладке **Таблица** и в открывшемся контекстном меню выберем пункт **Задать имя таблицы** (рис. 6.18). В появившемся окне напишем новое имя шаблона: **Каталог**.

Ko	нфигурато	ор - Бужиог	пер учет, ра	асчет - [Фо	рма-Справ	очник.Номе	нклатура.	ФормаСпис	ка.ФормаС	писка]						- 8 ×	
ЩФ. Шара	аил Деист	вия Габлиц и пъ «тъ	ца Вид Ко	нфигурация	Конструкт	горы Админ	истрирован	ие Сервис	Окна Пом	ющь							
10	GRI	8 49 G			<i>(</i> P9)		± é	5 76 <b>V</b>									
E,E	<b>A</b> X (	m ~ 6 *	0    Ω	巨匠	i k	n n n	8 8	13 Ka Ka	脑前面	š.							
R1C1		<b>▼</b> -{	<b>д =</b>														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	<u> </u>	
2		4															
3																	
4																	
5																	
7																	
8																	
10																	
11																	
12																	
14																	
15																	
16																	
18																	
19																	
20																	
21																	
23																	
24																	
25																	
27																	
28																	
29																	
31																_	
32																	
4								_								Þ	
\ E9.	Диалог	<u>}</u> ∭ M	одуль	👌 🖄 Ката	лог	🔪 🔊 Табль	Добави	/ пь таблиич									
$ [ ] \$		1 <u>12</u> 100		[]}		<b></b>	Удалит	ъ таблицу									
π⁄ड Ko	нфигурация	*	<b>Д</b> Справ	очник Номен	клатура	🖾 Форма-С	Задать	имя таблиць									
Созда	гь новый эл	емент					Переместить/Копировать			NUM СТАТУС: БД , Документ *							
街 Пуск 🥥 🍙 💿 🔌 🗀 З Проводник 🔹 💓 2 Microsoft 🔹 🎆 Рај			Разделить окно			CorelDRAW 10 🗷 Microsoft Exce 🔃 🕐 ኛ 😽 🔠 🌾 17:16											

Рис. 6.18. Переименование шаблона печатной формы

Начнем создание шапки нашего будущего отчета — заголовочной части. Посмотрим, где примерно должен располагаться заголовок, и впишем в ячейку название отчета: **Каталог товаров** (см. рис. 6.18). На следующей строке напишем названия столбцов отчета — реквизитов справочника: **Наименование**, **Тип номенклатуры**, **Тип товара** и остальных. Текст в ячейку можно вносить и без палитры свойств, просто щелкните мышью на нужной ячейке и вводите. Если тип "Текст", то этим можно и ограничиться, но если введенное значение имеет тип "Выражение", "Шаблон" или "Фиксированный шаблон", то эти типы необходимо устанавливать в окне **Свойства ячейки**.

Теперь нужно выделить строки с заголовком для создания секции документа. Для этого щелкните левой кнопкой мыши по номеру строки и протяните указатель мыши вниз до строки конца заголовочной части. В меню **Таблица** выберите пункт **Включить в секцию** и в диалоге ввода задайте имя секции: Шапка (рис. 6.19).



Рис. 6.19. Секция

Для лучшей читаемости шаблона печатной формы пропустим одну строку шаблона таблицы, а следующую выделим и создадим секцию **Строка**. В этой секции будут выводиться непосредственно строки справочника.

На данном этапе можно переходить к написанию процедуры печати и в дальнейшем одновременно создавать шаблон печатной формы и процедуру печати на встроенном языке.

Щелкните по вкладке **Модуль** экранной формы справочника. В модуле формы создайте процедуру печати (листинг 6.1).

```
Листинг 6.1
```

```
Процедура Печать()
```

Спр=СоздатьОбъект ("Справочник.Номенклатура"); //Создана переменная для ссылки на объект Справочник Номенклатура

Если Спр.ВыбратьЭлементы()=1 Тогда //справочник позиционирован

Пока Спр.ПолучитьЭлемент()=1 Цикл

КонецЦикла;

КонецЕсли;

КонецПроцедуры

В процедуре пока реализована только одна часть, согласно описанию в *разд.* 6.1.2 — выборка из справочника. В данном случае мы создали переменную Спр, которая содержит объект типа Справочник. Номенклатура. Теперь мы можем выполнять различные действия над этим объектом, указывая стандартные методы этого объекта. Название метода записывается через точку после самого объекта, и после него всегда ставятся круглые скобки. В скобках могут указываться параметры.

# Примечание

Обращение к объекту посредством создания переменной с произвольным именем позволяет использовать в одном модуле несколько однотипных объектов.

Подробнее разберем фрагмент процедуры.

С помощью конструкции (листинг 6.2).

#### Листинг 6.2

Если Спр.ВыбратьЭлементы()=1 Тогда

. . . .

КонецЕсли;

проверяем, есть ли хотя бы один элемент в справочнике. Если есть, позиционируем запись и начинаем перебор элементов в цикле, до тех пор, пока можно получить элемент (листинг 6.3):

#### Листинг 6.3

Пока Спр.ПолучитьЭлемент()=1 Цикл КонецЦикла;

Для вывода записей справочника в печатную форму все готово. Нужно дописать код для работы с шаблоном печатной формы.

Так как таблица — служебный объект метаданных, для работы с ней из встроенного языка мы должны создать соответствующую объектную переменную, как ранее для объекта Справочник. Номенклатура.

Таб=СоздатьОбъект ("Таблица");

Опишем действия, которые производятся с переменной таб.

Таб.ИсходнаяТаблица("Каталог");

В этой строке вызывается метод исходная Таблица (), который назначает исходный шаблон печатной формы. Ведь у одного справочника или документа может быть несколько печатных форм.

Таб.ВывестиСекцию("Шапка");

Эта строка выполняет включение в печатную форму подготовленной нами секции. Первой выводится секция Шапка с заголовком отчета. Вывод секции Строка обсудим чуть позже.

Закончить работу с переменной таб необходимо следующим набором строк (листинг 6.4):

#### Листинг 6.4

Таб.ТолькоПросмотр(1); Таб.Опции(0,0,0,0); Таб.Показать("Каталог товаров");

Метод ТолькоПросмотр (1) определяет, что печатная форма будет использоваться только для просмотра и печати, а не для редактирования. Метод Опции (...) многообразен, он позволяет, к примеру, показывать или очищать линии сетки в печатной форме, заголовки строк и столбцов и т. д. В данном случае поставлены стандартные значения метода. Метод Показать () выводит форму печатного документа на экран для просмотра и печати. Если этот метод не вызывать, то при работе процедуры мы так и не увидим печатную форму документа. В параметрах этого метода пишется название-подсказка окна печатной формы.

Теперь добавим операторы, формирующие печатную форму документа в процедуру Печать () (листинг 6.5):

#### Листинг 6.5

Процедура Печать()

```
Спр=СоздатьОбъект ("Справочник.Номенклатура"); //Создана переменная
для ссылки на объект Справочник Номенклатура
Таб=СоздатьОбъект ("Таблица");
Таб.ИсходнаяТаблица ("Каталог");
Таб.ВывестиСекцию ("Шапка");
Если Спр.ВыбратьЭлементы()=1 Тогда // справочник позиционирован
Пока Спр.ПолучитьЭлемент()=1 Цикл
КонецЦикла;
КонецЕсли;
Таб.ТолькоПросмотр(1);
Таб.Опции(0,0,0,0);
Таб.Показать ("Каталог товаров");
КонецПроцедуры
```

В процедуре не хватает только одной части — вывода записей справочника построчно. Перейдем в шаблон печатной формы. Ниже секции Шапка создадим новую секцию: Строка. В ячейках этой секции напишем выражения (т. е. вычисляемые значения). Для вывода наименования товара запишем выражение Спр. Наименование, Для вывода типа номенклатуры спр. ТипНоменклатуры и т. д. Подобная запись указывает системе, какие данные подставлять в выражение — в данном случае из объекта, который описывает переменная спр. Обратите внимание на запись выражения использованием периодических реквизитов: С Себестоимость. Цена, СтавкаНДС. Эти выражения должны записываться так: Спр.Себестоимость.Получить (РабочаяДата ()). Периодичность подразумевает наличие истории значений (см. разд. 5.1). Следовательно, системе необходимо указать, какую дату нужно выбрать.

В секции Строка за один раз отображается только одна запись справочника. Поэтому в процедуре Печать () вызов секции мы помещаем в цикл, действующий до тех пор, пока не перебраны все записи справочника. Этот цикл в процедуре уже записан, но пока работал вхолостую. Окончательный текст процедуры выглядит так:

#### Листинг 6.6

```
Процедура Печать()
```

Спр=СоздатьОбъект("Справочник.Номенклатура"); //Создана переменная для ссылки на объект Справочник Номенклатура

```
Таб=СоздатьОбъект ("Таблица");

Таб.ИсходнаяТаблица ("Каталог");

Таб.ВывестиСекцию ("Шапка");

Если Спр.ВыбратьЭлементы()=1 Тогда // справочник позиционирован

Пока Спр.ПолучитьЭлемент()=1 Цикл

Таб.ВывестиСекцию ("Строка");

КонецЦикла;

КонецЕсли;

Таб.ТолькоПросмотр(1);

Таб.Опции(0,0,0,0);

Таб.Показать ("Каталог товаров");
```

КонецПроцедуры

С помощью палитры свойств ячейки в шаблоне печатной формы нужно сделать отчет более удобным и читаемым. Для этого нужно выполнить следующие действия:

- 1. Подогнать ширину колонок (это можно сделать, перетащив мышью разделитель между колонками таблицы в верхнем заголовке — сером поле с номерами колонок).
- 2. Обвести рамкой ячейки.
- 3. Установить положение текста в ячейках.
- 4. Установить размер и вид шрифта.

Печатная форма закончена (рис. 6.20).

Но если сейчас открыть справочник в пользовательском режиме, печатной формы мы не увидим. Вызов печатной формы — это интерактивная функция нашей программы. Следовательно, необходимо создать механизм, вызывающий в нужный момент печатную форму.

Организовать вызов созданной процедуры можно с помощью кнопки, расположенной на экранной форме справочника. В режиме конфигуратора

нужно в форме списка справочника "Номенклатура" перейти на вкладку Диалог. Щелкнуть левой кнопкой мыши по краю диалоговой формы. Появится граница формы в виде синей линии, обводящей форму, с маркерами — прямоугольниками. Подведите курсор мыши к такому маркеру в нижней части диалоговой формы, и курсор изменит свою форму: будет выглядеть в виде стрелки. Зацепив маркер мышью, раздвиньте форму немного вниз (рис. 6.21). Таким образом появляется место для добавления других элементов диалога, в частности, элемента Кнопка. Для создания кнопки можно выбрать из меню Вставить пункт Кнопка либо нажать пиктограмму Кнопка на панели инструментов Элементы диалога (см. рис. 6.21). Далее нужно подвести курсор к левому верхнему углу того места, куда вы хотите вставить кнопку, нажать левую кнопку мыши и, не отпуская ее, передвинуть указатель мыши ниже и правее так, чтобы очертить прямоугольник кнопки, затем отпустить кнопку мыши.

<b>С</b> Конфи	гурато	р - Бух и опер учет, расчет - [(	Форма-Справоч	ник.Номен	клатура.Фо	ормаСписка	.ФормаСпи	ска]					_ 8 ×
Щ Файл Портеж	Дейст	зия Таблица Вид Конфигурац У Въ 🛱 / 🖾 да   🗊 🐀 /	ия Конструктор → АА	ы Админис	трирование	େ Cepвиc C	жна Помоц	ць					
			= ಆಗತ] ಕ    ಮಂನಂತ್ರತ್	a sta eta i		89 8 2 - 22 - 22 - 13	N .N. N.	1					
R24C5	- <i>4</i> 6	ena ™26809   3×  0± 02			n e e	N EN EN 14							
		1	2	3	4	5	6	7	8	9	10	11	12 🔺
Шапка	1												
	2			Катало	г товаро	ов							
	3	Наименование	номенклатуры	товара	Ед. изм.	Себестоим	Цена	ст. НДС.					
	4												
Строка	5	сСпр Наиманованиез	Con Turbioseus	Can TunTo	Con Enuna	Can Ceffer	Con Lleve I	Con Creek	HIC Dogu	urts (Deficilied	[ara())>		
o i posta	7	SCH03 IdvillionObanvio/	«спр.типпомен»	scrip.ruirro	копр.сдини	scrip.ceoec	«спр.щенал	чепр.етавк	arizaci nony 4	ить(гассчая	дата())#		
	8												
	10												
	11												
	12												
	14												
	15												
	16												
	18												
	19												
	20												
	21												
	23												
	24												
	25												
	27												
	28												
	30												
	31												-
•													Þ
\БЭДиал	ior	Moaguns )Ka   ∭ ] ] ∭_ [ ]	талог / ∄ 🔳 🐨 📼	5 150 m									
च <sup>∕च</sup> Конфи	урация	* 🎒 Справочник Ном	енклатура	Форма-Спр	авочник.Н	0							
Для получ	эния по,	асказки нажмите F1									NUM	СТАТУС: БД	, Документ *
<i>1</i> :Пуск	ø 0	) 🕑 » 🗋 4 Проводник 🗸 🛛	2 Microsoft	- 🚟 Paint	Shop Pro	2 1C:V	7 star •	CorelDRA	V 10	Microsoft Exc	e RU	? 🗘 « 🗄	3 30 🌾 17:28

Рис. 6.20. Готовый шаблон печатной формы

Sa Конф	куратор -	Базовые	е объе	кты	- [Форм	а-Спра	вочник	.Номен	клатура	.Форма	Списк	a.Фo	рмаС	писка	a]					- 5	×
🖪 Файл	Действия	Диалог	Вста	вить	Конфиг	урация	Конст	рукторы	Админ	истриров	ание	Cepa	вис	Окна	Пом	ющь				- 5	×
12 🖻	S .	Pa 💼	8	Ş.	1	c M				▼ #	2	?									
	4 Z %	~~ *	$\odot$										ß	e e	违	à	Š1	řÉ	ă.	ž.	ä
鄭									_ 🗆 ×												
E o	а Кратко	е наннени	рание		Почнос	наниен	оранне														
									=												
	+				<u> </u>				=												
									_												
									=												
	+				l																
									-												
	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	1104					,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,														
					È.					-											
े हिंदी <b>ग</b> क्त	-	/ El Mo	01105		Alk.	27 2005		Alt-	forma =1	/											_
( <u>C)</u> ((+ +)		<u>alu 2.</u>	i i i i i i i i i i i i i i i i i i i	er h		찌ㅣᄈ	/ # %a	<u>a</u> r	олицат	/	e	~	•		- 194		abi B	n i r		a l r	2
]]It+ +1		까지 주지	P**4	<u>e.</u>   !	1.2.1	22   #	*   3	-	-	∐ ° <b>r</b>	Ă	-			3		aul	11   12	-   -		20
<u>π{а</u> Конфи	гурация	<b>@</b> •<	правоч	ник Н	фмен	ЩΦор	има-Спр	авочн.						1.	_				_		_
Для получ	ения подска	азки нажи	ите F1		1	a ur		1	o c		NUM J	358 x	: 183 Dei -	0,1	0 191 c			:6Д,	Докул	ент	
и пуск	82 9 6	a <u>C</u>	/пав	54		0 2 Micr	osort		2 10:97	sta • _	1 PH	c4_4 ·	Pain		<b>3</b> (0	reiDRA	W 10		« 14	17:	53
				Ma	ркер					Кı	юпк	a									

Рис. 6.21. Создание кнопки в экранной форме

Теперь обратимся к палитре свойств созданной кнопки.

Щелкнем по прямоугольнику дважды левой кнопкой мыши либо правой кнопкой вызовем контекстное меню и выберем пункт Свойства. На вкладке Общие нужно изменить заголовок по умолчанию (Новый1) на Печать (рис. 6.22). Этот заголовок будет написан на кнопке. Пользователю будет понятно, для чего эта кнопка предназначена. Чтобы определить вызываемые кнопкой действия, необходимо переключиться на вкладку Дополнительные. Поле Формула нам уже знакомо. В этом поле нужно написать вызов процедуры Печать ().

# Замечание

Хороший стиль написания кода — строки пишутся одна под другой. Клавишей <Tab> делаются отступы для строк, находящихся внутри процедур и внутри циклов. Это делает текст модуля более наглядным. Чтобы соответствующим образом отформатировать модуль, можно выбрать из меню **Действия** пункт **Выделить все**, затем зайти в меню **Текст | Блок | Форматировать**. Все строки модуля будут выровнены в соответствии со стандартом. Теперь проверьте работу своей программы в пользовательском режиме. Если все верно, то при нажатии кнопки **Печать** на экран будет выведен готовый к распечатке табличный отчет по справочнику номенклатуры "Каталог товаров" (рис. 6.23).



Рис. 6.22. Вызов процедуры из палитры свойств кнопки

🗊 1С:Предприятие	-							_ 8 ×
Файл Действия Та	блица Вид Операции Справо	очники Документы Ж	урналы Отчеты	Сервис С	жна Помоц	ць		
1 🖻 🖬 🖌 🛙	16 6 6 🛛 🗅 C	<i>i</i> 44	■ # #	8			🗐 🎬 🎾 🛄 М М+ М	M-
		_	_	_	_	_		
🖨 Спра	вочник: Справочник номенкл	атуры (06.06.04)						
	₽₩₽₽₽₽	e 🖻 🗄 🕄 😽						
	од Краткое наименование 1 Телевизор "Samsung"		<u> </u>					
	2 Холодильник							
	🛃 Каталог товаров *							
		1						
		Кат	алог товаро	в				
	Наименование	номенклатуры товар	ед. изм.	Себестоим	Цена	ст. НДС.		
		- · · · ·						
	Teлeвизор "Samsung" Холодильник	Товар Товар Товар Товар	шт шт	2000	20000	16		
		1						
, <u> </u>								
	-1							
			-					
			-					
🗐 Справочник: Справ	вочник 🕅 🕅 Каталог товаро	в*						
Для получения подска	зки нажмите F1			NUM	TA: 01.01.0	2 00:00:00	БИ: 2 квартал 2004 г.	TN:
🏝 Пуск 🕼 🍏 🕃	) » 🔄 Paint Shop Pro 🗟 10	Предпр 🕅 6Глава	14 🦗 Конс	фигура	😤 1С:Пред	Inp 📰 Pai	nt Shop P 🦉 Рис4_8 - Paint	RU 🕐 🖡 « 🖳 9:08

Рис. 6.23. Отчет Каталог товаров

# 6.6. Создание основных справочников конфигурации

В этом разделе описывается создание основных справочников учебной конфигурации. Этот процесс аналогичен процессу создания справочника "Номенклатура". Сначала нам необходимо доработать этот справочник.

Откроем окно редактирования справочника "Номенклатура". Внесем изменения в реквизит ЕдиницыИзмерения. Щелкнем мышкой этот реквизит, затем кнопку Изменить. В палитре свойств реквизита перейдем к полю Тип значения. Откроем список значения этого поля. Помимо значения "Число", "Строка", "Дата" там появятся и значения типа "Перечисление". Дело в том, что этот объект метаданных является типообразующим. То есть, создавая очередное перечисление, мы создаем очередной тип данных для реквизита. Реквизит с этим типом данных ссылается на соответствующий объект метаданных и получает данные прямо из типообразующего объекта метаданных.

# Замечание

Ряд объектов метаданных, таких как Справочники, Перечисления, Документы, Счета, Планы счетов Субконто, являются типообразующими объектами метаданных.

Выберем для реквизита ЕдиницыИзмерения тип данных Перечисление.ЕдиницыИзмерения (рис. 6.24). Если все сделано правильно, то в пользовательском режиме при записи контрагента реквизит ЕдиницыИзмерения будет раскрываться в виде списка. Из этого списка можно выбрать любое из значений. Это удобно с двух сторон: ввод данных в реквизит ЕдиницыИзмерения ускоряется, и нет ошибок ввода.

Справочник "Материалы" предназначен для хранения информации о поступающих и исходящих материалах. По своей структуре он идентичен справочнику "Номенклатура". Имеет следующие реквизиты:

🛛 Код;

- □ Краткое наименование;
- □ Полное наименование;
- Единица измерения;
- □ Себестоимость;
- □ Ставка НДС.

Справочник двухуровневый. Имеет форму элемента и форму списка. Напомню, что два реквизита справочника созданы по умолчанию — **Код** и **Наименование**. Краткое наименование материалов можно записывать в реквизите Наименование. Для полного наименования следует создать новый реквизит ПолноеНаименование с типом значения "Строка". Длина максимальная — 100 знаков. Единица измерения, Ставка НДС и Себестоимость такие же реквизиты, как и в справочнике "Номенклатура": соответственно, типа "Строка" и "Число", длина 15, точность 2. Для этого справочника создается печатная форма, аналогичная форме справочника "Номенклатура".



Рис. 6.24. Справочник "Номенклатура" с реквизитом ЕдиницыИзмерения типа "Перечисление"

Справочник "Подразделения" служит для ведения списка подразделений фирмы. Справочник двухуровневый, не имеет дополнительных реквизитов, не имеет формы элемента. Форма списка состоит из реквизитов **Код** и **Наименование**.

Справочник "Основания" предназначен для хранения информации о договорах с контрагентами. Справочник двухуровневый, имеет два реквизита ДатаНачала — дата начала действия договора и ДатаОкончания — дата окончания действия договора. Тип значения обоих реквизитов — "Дата". Справочник "Контрагенты" предназначен для хранения информации о контрагентах фирмы. Справочник трехуровневый, имеет реквизиты:

- 🛛 Код;
- □ Краткое наименование;
- □ Полное наименование;
- 🛛 Резидент.

Реквизит Резидент имеет тип значения "Перечисление. Резидент", которое мы создали ранее.

Сохраним конфигурацию.

Вернемся к справочнику "Основания" и в окне редактирования, в поле Подчинен выберем справочник-владелец: "Контрагенты".

Теперь продолжим создание справочника "Контрагенты". Добавим новый реквизит "Основания", который имеет тип значения "Справочник.Основания", создадим форму элемента и печатную форму справочника.

Справочник "Сотрудники" предназначен для ведения списка сотрудников фирмы. Справочник одноуровневый (флажок **Размещать группы сверху** не активен). Имеет один реквизит — **Подразделение**, тип значения "Справочники.Подразделения".

Справочник "Основные средства" предназначен для хранения информации о поступающих и выбывающих основных средствах фирмы. Имеет два уровня. На данном этапе в него необходимо включить реквизиты:

- 🛛 Код;
- □ Наименование;
- □ Признак начисления амортизации;
- 🛛 Норма амортизации;
- **Группа ОС**;
- 🛛 НачальнаяСтоимость.

Код и Наименование, как известно, являются реквизитами справочника по умолчанию. Реквизит Признак начисления амортизации будет в виде флажка. Идентификатор — НачислятьАмортизацию, Синоним — Начислять амортизацию, Комментарий — Начислять амортизацию да\нет. Тип значения — "Число", длина 1, точность 0. Ранее флажок мы использовали в обработке. Напомню, что он может принимать два значения: 0 и 1. Отсюда следует и установленный тип значения с длиной и точностью.

Реквизит **Норма амортизации** имеет идентификатор **НормаАмортизации**, синоним **Норма амортизации в** %. Тип значения — "Число", длина 6 символов, точность 2. Норма амортизации назначается в %, поэтому 3 знака длины числа — это количество процентов (максимально 100), 1 знак — разделитель целой и дробной части, 2 знака — точность дробной части.

Реквизит Группа ОС имеет тип значения "Перечисление. ГруппыОС".

Реквизит **НачальнаяСтоимость** предназначен для ввода начальной стоимости ОС. Тип значения — "Число", длина 15 символов, точность 2.

Справочник имеет форму списка и форму элемента. В форме списка справочника не выводится реквизит **НачислениеАмортизации**. В форме элемента выводятся все реквизиты. При вставке реквизитов в форму элемента в диалоговом окне **Размещение реквизитов справочника** в поле **Вставлять** необходимо выбрать из списка значение **Флажок** (рис. 6.25). Вид формы списка должен получиться, как на рис. 6.26.



Рис. 6.25. Вставка реквизита в виде флажка

🚰 Конфигуратор - Бух и опер учет, расчет - [Форма элемента-Справочни	ик.ОсновныеСредства (Основные средства)]	_ 8 ×
🔄 Файл Действия Диалог Вставить Конфигурация Конструкторы Адми	нистрирование Сервис Окна Помощь	_ 8 ×
11 <b>2 2 3 4 6 6 6 6 8 1</b> 3 € 6		
白白白白白有 想出 6 ① ● 9 年代序 10	· · · · · · · · · · · · · · · · · · ·	
<b>邦</b>		
Код: Код Наименование: Наименование		
Норма амортизации: НормаАмортиза Счет затрат: СчетЗатрат 🛄 🗜	азмещение реквизитов Справочника 🛛	
Вид затрат. Вид Затрат		
Начисоять амортиза	✓НормаАмортизации ✓СчетЗатрат	
	ВидЗатрат	
ОК Закрыть	НачальнаяСтоимость	
	Вставлять имя Вставляты Поле ввода	
	П Разместить в диалоге автоматически	
	Вызывать для новых форм	
	Вставить Отмена Помощь	
<u>、日和anor (1 Maguns ) 然</u> Taónuua 第二部	● = = = = = = = = = = = = = = = = = = =	
로글 <sup>2</sup> Конфигурация * 🎒 Справочник ОсновныеСр 🛛 🖽 Форма элеми	ента-Спра	
Для получения подсказки нажните F1	NUM 89 x 13 57, 78 CTATYC: 5Д *, Д	жумент *
🟦 Пуск 🥥 🔄 🕑 🍟 🥰 Конфигур 🗀 2 Провод 🔻 🖭 8Глава 6	🖭 Документ1 🧱 Paint Shop Pro 🧟 CorelDRAW 🔃 😰 🗘 ሩ п 🗄 🗉 🌾 🖵 🏟	16:32

Рис. 6.26. Форма элемента справочника Основные средства

Сохраните конфигурацию. Запустите ее в пользовательском режиме. Теперь вы можете работать со справочниками, используя меню Справочники созданного ранее интерфейса.



# Документы

Документы в системе "1С:Предприятие" служат для ввода в систему информации о совершаемых хозяйственных операциях, ее просмотра, корректировки. Документы часто являются электронными аналогами стандартных бумажных документов. Например, "Приходная накладная" или "Расходный кассовый ордер". Однако следует помнить, что использование этого типа данных может выходить далеко за рамки простой фиксации информации о хозяйственных операциях.

Для знакомства с этим типом метаданных начнем создавать документ "Поступление товаров", для фиксации поступления товаров на фирму.

Откройте дерево метаданных, щелкните левой кнопкой мыши по ветви Документы, затем щелкните правой кнопкой мыши. В контекстном меню выберите пункт **Новый документ**. Откроется первое окно конструктора (мастера) документа (рис. 7.1). Прочитайте подсказку в центре окна конструктора и заполните поля идентификатора, синонима и комментария документа следующими значениями:

**Идентификатор** — ПоступлениеТоваров;

- **Синоним** Поступление товаров;
- **Комментарий** Поступление товаров.

Нажмите кнопку Далее. Флажки проведения и перепроведения документа установите, если они не выставлены. Документ будет проводиться и перепроводиться, так как, согласно *главе 4*, он будет отражаться в бухгалтерском и оперативном (торговом) учете. Нажмите кнопку Далее, в появившемся окне установите флажок Документ будет принадлежать журналу. Затем выберите пункт Создать новый журнал. По умолчанию идентификатор журнала такой же, как и у документа "ПоступлениеТоваров". Однако в нашем журнале планируется отражать все документы по учету товаров. Поэтому назовите журнал "УчетТоваров". Нажмите кнопку Далее, вставьте документ в пользовательский интерфейс.

😪 Конфигуратор		_ & ×
च्च Файл Действия Конфигурация Н	Конструкторы Администрирование Сервис Окна Помощь	_ & ×
Конфигуратор Табайл Действия Конфигурация 1 Сайл Действия Конфигурация 1 Сайл Действия Конфигурация 1 Сайл Действия Сонфигурация 1 Сайлание Сайлание С	Конструкторы Адининистрирование Сервис Окна Помощь Сонструктор Аланинистрирование Сервис Окна Помощь С С С С С С С С С С С С С С С С С С С	
		I
ны метаданные ни Интерфейсы		
च्≝Конфигурация *		
Для получения подсказки нажмите F1	NUM CTATY	С: БД *, Документ *
赴 Пуск 🖉 🎯 🕑 👋 🍕 2 1С:¥7	sta 🕜 Рисунки 🛛 🦉 Рис2_14 - Paint 🛛 🧱 Paint Shop Pro 🕅 🛛 Alicrosoft .	🔻 🔣 < 17:46

Рис. 7.1. Конструктор документа

Нажмите кнопку Далее. Этот шаг — заполнение свойств журнала. Идентификатор уже есть, синоним и комментарий заполните самостоятельно. Нажмите кнопку **Готово**.

Подробнее о журналах документов будет сказано далее.

На рис. 7.2 показано окно редактирования документа.

Окно редактирования документа похоже на аналогичное окно для справочников. Для добавления реквизитов предусмотрены две области: **Реквизиты шапки** и **Реквизиты табличной части**. Термин "шапка" используется для обозначения заголовочной части документа. Однако правильнее считать реквизитом шапки любой реквизит, значение которого встречается в документе только один раз. В табличной части размещаются реквизиты, которые могут иметь несколько значений. Обычно определить, будет ли реквизит относиться к шапке или к табличной части, достаточно просто. Если у вас есть бланк документа (накладной, счета), то из него видно, какие реквизиты будут относиться к табличной части, т. е. являться колонками в многострочной части документа. Если бланка под рукой нет, можно мысленно выделить группу реквизитов, которая будет иметь в документе единственное значение, и группу реквизитов, значения которых будут заполняться для каждой строки.

🚰 Конфигуратор - Бух и опер учет, расчет - [Документ ПоступлениеТоваров]	
Файл Конфигурация Конструкторы Администрирование Сервис Окна Помощь	
」"입ば묘↓⋡時記(ゟゟ)Ⅲ⊃⊂∰ <mark>BaseAox</mark>	8
长治的 各位的新加格 昭田 \$1 0000 年代度 90	· · · · · · · · · · · · · · · · · · ·
Идентификатор: ПоступлениеТоваров Журнал: УчетТоваров	•
Комментарий: Оступление товаров	
Реквизиты шапки	
1 4 2 Новый Изменить Удалить Новый Изменить Удалить	
Номер	
Нумератор: <<< Не назначен >> 💌 🔽 Иипловой Длина: 5 👘	
Периодичность: По всем данного вида 💌 С Текстовый	
🔽 Автоматическая нумерация 🔽 Контроль уникальности	
Разрешить проведение документа	
Автоматическое удаление движений <u>Расчет</u>	
Автоматическая нумерация строк 🔽 Опедативный учет	
Создавать операцию: Всегда 💌 🗖 Редактировать операцию	
Ввод на основании Описание Форма Модуль Докумен	та
📺 Конфигурация 🛛 📄 Документ Поступлени	
Для получения подсказки нажните F1	NUM СТАТУС: БД , Документ
🗶 Пуск 🥥 🔄 🐑 🗋 2 Проводник 🔻 🖻 Документ1 🖻 9Глава7 - Місг 🧱 Ра	int Shop Pro 🛛 🦿 2 1C:V7 star 🔛 CorelDRAW 10 🛛 💟 🗘 « 🔲 🗄 🌾 16:04

Рис. 7.2. Окно редактирования документа

Для значительной части документов табличная часть вообще не нужна. Например, платежное поручение не имеет многострочной части.

# Замечание

У любого документа существуют, по умолчанию, три реквизита шапки: дата, время и номер документа, и один реквизит табличной части — номер строки.

Следующая область окна редактирования документа — **Номер**. Каждый документ в системе может нумероваться автоматически. Для этого должен быть установлен флажок Автоматическая нумерация. Номер может быть двух типов: числовой и текстовый. Тип номера выбирается переключателем Числовой/Текстовый. В случае использования текстового типа номера возможно включать в номер не только цифры, но и буквы, например, различные префиксы. Это удобно для поиска и сортировки документов в сложных учетных базах. Однако работа с текстовым номером документа имеет специфические особенности. Мы не касаемся их в данной книге, эти особенности можно изучить в документации к программе. В данном случае вполне достаточно числового типа номера с длиной 5 символов. Возможная длина номера выставляется в поле Длина и может состоять не более чем из 20 символов.

Поле **Нумератор** предназначено для работы со специальным видом объектов метаданных Нумератор. В этом объекте метаданных можно описать правила нумерации документа более подробно, чем в окне редактирования документа. Порядок работы с нумераторами можно посмотреть в документации к программе.

В поле **Периодичность** устанавливается период повторяемости номеров документов. Возможные варианты:

- □ По всем данного вида;
- В пределах года;
- 🗖 В пределах квартала;
- В пределах месяца;
- В пределах дня.

Этот признак связан с признаками Автоматическая нумерация и Контроль уникальности. Если включен признак контроля уникальности, тогда Периодичность отслеживает границы осуществления этого контроля. В этих же границах осуществляется и автоматическая нумерация документа. Если Периодичность выбрана В пределах года, то с каждого следующего года номера документов будут начинаться с первого. Если же выбрано По всем данного вида, то номера будут продолжаться, пока работает база. В первом случае уникальность номера документа будет отслеживаться в пределах года, во втором — в пределах существования информационной базы.

Область, расположенная ниже в окне редактирования документа, касается параметров его проведения.

Если мы создаем документ для фиксирования каких-либо данных, то проведение его нам не требуется, и флажок **Разрешить проведение документа** может быть снят. Если же документ должен влиять на учет в нашей компании, например, генерировать бухгалтерские проводки, то флажок должен быть установлен, чтобы система могла задействовать необходимые механизмы. Если документ будет проводиться, то, в большинстве случаев должен быть установлен флажок **Автоматическое удаление** движений. Установка этого флажка означает, что изменения в учете, сделанные этим документом при проведении, будут отменены, если документ перепроводится, делается отмена проведения или документ удаляется.

Установка флажка **Автоматическая нумерация строк** означает, что система сама будет нумеровать строки в табличной части документа. Можно изменять порядок расположения строк, однако номера всегда будут отражать последовательность строк и не будет пропусков. При отключении этого режима номера строк могут редактироваться и нумерация может иметь пропуски.

Флажки с названиями компонент "1С:Предприятия" указывают, в каких именно системах учета документ будет выполнять изменения при проведении. Если установлен флажок **Бухгалтерский учет**, документ может генерировать операции и проводки.

# Предупреждение

Флажок Бухгалтерский учет может быть доступным, если в конфигурации создан хотя бы один план счетов.

Если установлен флажок **Оперативный учет**, документ при проведении может менять содержимое регистров оперативного учета. При установке флажка **Расчет** становятся доступными журналы расчетов.

# Предупреждение

Включение каждой из опций вызывает расход дополнительных ресурсов системы. Поэтому следует избегать включения "ненужных" компонент для документа.

Наличие флажка компоненты зависит от состава компонент системы "1С:Предприятие".

В самом низу окна редактирования документа находится управляющий элемент **Создавать операцию**. Он становится доступным, если для документа включена опция **Бухгалтерский учет**, и имеет три значения:

- □ Всегда операция записывается в момент записи документа, документ всегда присутствует в журнале операций;
- □ **Выборочно** конкретные документы могут создавать операцию либо при записи, либо при проведении;
- □ **Только при проведении** только проведенные документы могут иметь операцию.

Документы в системе "1С:Предприятие" могут создаваться на основании друг друга. Например, расходные документы можно выдавать на основании оплаченного счета. Технически это выглядит таким образом: счет имеет часть реквизитов, аналогичных реквизитам расходного документа. При выписке расходного документа на основании счета значения аналогичных реквизитов автоматически попадают в расходный документ. Это позволяет сэкономить время оператора и избежать ошибок. При вызове метода **Ввод** на основании через меню система покажет все имеющиеся документы конфигурации для выбора нужного. Чтобы ограничить количество документов в диалоге, в окне редактирования документа нужно настроить диалог **Ввод на** основании с помощью кнопки **Ввод на основании** (рис. 7.3). В левой части диалога можно отметить документы, для которых редактируемый документ является основанием, в правой — на основании которых будет вводиться редактируемый документ.

Кнопка Описание открывает текстовое окно, в котором можно напечатать описание работы с документом.

Сабонфигуратор - Бух и опер учет, расчет - [Документ ПоступлениеТоваров] Подал Конфигирация Конструкторы Аликиистрирование Селен- Окие Поняць	_ 8 ×
	- 면 스
· · · · · · · · · · · · · · · · · · ·	
Идентичикатор: ПоступлениеТоваров Журнал: УчетТоваров ▼ Комментарий: Диноним: Поступление товаров Реквизиты шалки Реквизиты паблиеной части	
Ввод на основании	
Может являться основанием для документа любого вида	
Является основанием для. Вводить на основании.	
Анхортизация Вылиска Валиска Валиск	
Создавать операцию: Всегда 🔄 🗖 Редектироветь операцию	
Вод на основании	
жа Конфигурация	
Для получения подсказки нажинте F1 [NUM_CTATYC: 5Д ,	Документ
🖞 Пуск 🧑 🏹 💿 🎽 🗋 2 Проводник 🔹 💆 Документ 1 🖉 9Глава7 - Міст 🕎 Pant Shop Pro 🛛 💞 2 I CiV7 star 🖉 CorelDRAW 10 🔟 🖓 🗳 « э	31 🌾 16:06

Рис. 7.3. Диалог Ввод на основании

Кнопка Форма открывает окно экранной формы документа.

Кнопка Модуль документа открывает окно модуля для описания процедуры проведения документа.

В документ "Поступление товаров" необходимо вносить следующие данные:

- □ наименование товара;
- 🗖 цену товара;
- 🗖 количество товара;
- □ сумму;
- □ ставку НДС;
- □ сумму НДС.

🧟 (кифигуратор - Бух и опер учет, расчет - [Документ ПоступлениеТоваров]	
В сели кондиграция конструкторы Адининстрирование Серенс Окна Поношь В сели кондиграция конструкторы Адининстрирование Серенс Окна Поношь Сона Поношь Сона Сона Сона Сона Сона Сона Сона Сона	
在杨汐林浩浩  各名的杨子语的  191 10 10 10 10 10 10 10 10 10 10 10 10 10	
Идентичикатор: ПоступлениеТ сварое Журнал: Учет Товарое	
Комментарий Динонини: Поступление товаров	
Реквизиты шалки Реквизиты табличной части	
Контралент Свойства Реквизита	
Общие Дополнительные 🦞 – 🖓	
Идентификатор: Контрагент	
Синонии: Контралент	
Илиние Комментарий. ОК	
измения Тип значения. Справочник.Контрагенты 🔽 Птинна	
Суртонатическая имеерация СКонторы инжальности	
№ Разрешить проведение документа IV Едухалятерский учет	
Г Автоманиеское уданение деяжении Г десчет О Автоманиеское уданение деяжении Г десчет	
вод на основании	
та Конфитурация 📄 Документ Поступлени	
Для получения подсказки нажните F1 ЛИМ (СТАТУС: БД., До	сумент
🖞 Пуск 🔊 🎧 💬 🐣 🗋 2 Проводник 🔹 🕅 Документ 1 🕅 9Глава7 - Міст 🗮 Рапк Shop Pro 🛛 🌾 2 1СкУ7 star 🕻 🖾 CorelDRAW 10 🔃 🖓 🖏 🦿 🗰	K 16:07

Рис. 7.4. Палитра свойств реквизита шапки документа

Создайте реквизиты шапки. Их будет только два — Контрагент и Основание. Нажмите кнопку Новый под полем Реквизиты шапки. Заполните окно свойств реквизита (рис. 7.4). Обратите внимание на тип значения: "Справочник. Контрагенты" (этот справочник был создан раньше, как и справочник "Основание").

В системе "1С:Предприятие" при создании справочника сразу образуется тип значения, соответствующий этому справочнику. Смысл выбора этого типа значения следующий: не нужно будет вручную вводить контрагентапоставщика. Его достаточно просто выбрать из автоматически открывающегося при обращении к реквизиту справочника. Выбор типа значения "Справочник" означает, по существу, прямое обращение к данным соответствующего справочника.

Если какие-либо данные хранятся в справочниках, старайтесь при их использовании создавать реквизит с типом значения "Справочник.ВашиДанные". Как правило, этот прием поможет экономить время пользователя и защитит ваши данные от излишних ошибок. Ведь если каждый раз впечатывать наименование товара, можно пропустить букву, внести сокращение и т п. А система воспримет такой неверно описанный товар, как совершенно новый, и произойдут ошибки в учете.



Рис. 7.5. Диалог размещения реквизитов справочника

Теперь создадим реквизиты табличной части. В одном документе может приходоваться несколько товаров, следовательно, товары у нас должны размещаться в табличной части. Нажмите кнопку **Новый** под полем **Реквизиты табличной части**. С помощью палитры свойств создайте реквизит **ТМЦ**, тип значения — "Справочник. Номенклатура". Затем реквизит **Цена**, тип значения — "Число", длина — 6 символов, точность — 2.

Создайте самостоятельно следующие реквизиты:

- □ Количество, длина 10, точность 2 символа;
- □ Сумма, длина 15, точность 2;
- **П НДС**, длина 15, точность 2;
- □ СтавкаНДС, длина 6, точность 2.

Если будут затруднения, обратитесь к учебному примеру.



Рис. 7.6. Шаблон экранной формы документа

Нажмите кнопку **Форма** в нижней части окна редактирования, и у вас появится диалоговое окно размещения реквизитов документа, аналогичное окну в конструкторе справочника (рис. 7.5).

Обратите внимание на реквизиты, образованные по умолчанию, о которых упоминалось раньше. Нажмите кнопку **Вставить**. Теперь вы должны получить шаблон экранной формы документа, как на рис. 7.6.

Следующим шагом отредактируем экранную форму с помощью элемента диалога **Текст**. Для этого удалим названия реквизитов **ДатаДок** и **НомерДок** и исправим шапку, чтобы внешний вид документа был такой же, как в учебной конфигурации.

💯 1С:Предприятие - Бух и опер учет, расчет	
Файл Действия Операции Справочники Документы Журна	лы Результаты расчета Отчеты Обработки Сервис Окна Помощь
▋▙▋▕▓▐▆▐▋ዿኇ▏▋ゝᢏ▓	ĨĂĂ ?
<b>Тії Журнал документов Полный (01.01.06-30.11.06)</b>	
Поступление товаров -Новый *	
●■■■副房屋車 ↓ ☆ ☆ 2 ℃	
Поступление товаров №	3 or 30.11.06
Контрагент:	
Основание:	Παρόορ
N ТМЦ Кол-во Цена	🚰 Номенклатура (Выбор) (30.11.06)
	低 し 思 勉 通 相 部 が あ の 取 2 2 え
	Код Краткое наименование Еди Тип номенклатуры
	С 2 тов2
	Экспорт информации
	Excel Jicropr
	Печать Кодовая страница 🔽 *.dbf
М Журнал документов Полн	]Номенклатура (Выбор
Для получения подсказки нажиите F1	NUM ТА: 29.09.06 15:29:36 БИ: 4 квартал 2006 г. ТП: Октябрь 2006 г.
🛃 Пуск 🧔 🔄 📀 🎽 🗋 2 Проводник 🔹 🖳 Документ1	🔄 9F naea7 - Micr 🛃 Paint Shop Pro 🛛 💞 3 1C:V7 star 🖉 CorelDRAW 10 🔃 🖓 ኛ « 🗋 🕮 🖡 16:18

**Рис. 7.7.** Вид документа с открытыми справочниками для выбора

Запустите конфигурацию в пользовательском режиме и откройте новый документ "Поступление Товаров" (Операция | Журналы документов | Журнал Учет товаров, в окне журнала выберите Новый либо откройте новый документ через меню Документы). На рис. 7.7 показан пользовательский вид документа. Если щелкнуть кнопкой мыши в поле **Контрагент**, откроется форма справочника **Контрагенты для выбора поставщика**. Так же можно добавить **Основание**. Если щелкнуть по строке в колонке **Товары**, можно выбрать товар для автоматической вставки в табличную часть документа.

Автоматическое заполнение поля **Контрагент** нас вполне устраивает. Но в рабочих конфигурациях, помимо заполнения колонки **Товар**, автоматически заполняются, как правило, и другие колонки — Цена, СтавкаНДС. Эти данные ведь тоже находятся в справочнике "Номенклатура", и нет необходимости их вводить вручную.

Существует несколько способов автоматического заполнения реквизитов табличной части документа. Мы рассмотрим два наиболее распространенных.

# 7.1. Построчное заполнение документа

Зайдите в шаблон экранной формы документа (см. рис. 7.6). Если она была закрыта, откройте окно Конфигурация (вкладка Метаданные), в дереве метаданных выберите Документы, раскройте эту ветвь (щелкните мышью по знаку +). Затем щелкните правой кнопкой ПоступлениеТоваров и в контекстном меню выберите пункт Редактировать форму.

Отпускная цена товара и процент ставки НДС у нас уже внесены в справочник "Номенклатура". Тип значения реквизита **ТМЦ** приходной накладной тоже внесен в справочник "Номенклатура". Когда вы щелкнете по строке документа в колонке **ТМЦ**, откроется справочник "Номенклатура". Выберите в нем строку с нужным товаром, т. е. позиционируйте его. Реквизит **ТМЦ** с типом значения "Справочник.Номенклатура" — это прямая ссылка на справочник номенклатуры с позиционированием (установкой на определенную строчку) этого справочника. Далее присваиваем реквизиту ТМЦ значение реквизита **Наименование** из справочника "Номенклатура", т. е. наименование товара. Вспомните, в *разд. 2.3* было установлено основное представление справочника номенклатуры в виде наименования. Поэтому в реквизите ТМЦ содержится не код товара из реквизита **Код справочника Номенклатура**, а наименование из реквизита **Наименование**.

Для обращения к другим реквизитам справочника используется точка. Поэтому подобный вид обращения называется "через точку". Чтобы найти значение цены товара в справочнике "Номенклатура", нам надо написать конструкцию тмц.цена. Механизм написания — прежний, только вместо значения реквизита Наименование справочника "Номенклатура" мы получаем значение реквизита Себестоимость того же справочника. А вот где написать конструкцию ТМЦ.Себестоимость? Нужно обратиться к табличной части шаблона экранной формы нашего документа. Конечно, себестоимость из справочника должна попасть в колонку документа "Цена". Но заработать эта конструкция должна при выборе товара. Мышью выделите колонку табличной части ТМЦ, из контекстного меню (правая кнопка мыши) выберите пункт Свойства, в палитре свойств поля ввода откройте вкладку Дополнительно. В поле формул напишите формулу Цена=Товар.Себестоимость.Получить (ДатаДок) (рис. 7.8).



Рис. 7.8. Состав реквизитов в формуле

# Примечание

Знак равенства в данном случае называется оператором присваивания.

Рисунок 7.8 дополнительно поясняет, каким образом в реквизит документа "Цена" попадает цена товара из справочника. Разберем подробнее обращение к периодическому реквизиту справочника. Структура данных периодических реквизитов аналогична описанной в *разд. 5.1* структуре данных периодических констант. Для того чтобы получить значение периодического реквизита на определенную дату, нужно применить метод Получить (Дата, на которую требуется получить значение реквизита). Это дата создания нашего документа. Она хранится в реквизите шапки документа "ДатаДок". Обращение к методам также осуществляется "через точку". Символ точки с запятой применяется для разделения нескольких формул в поле.

Процент ставки НДС передается в соответствующий реквизит документа аналогично цене. В поле **Формула** после формулы вычисления цены товара и точки с запятой напишите следующее:

СтавкаНДС= ТМЦ.СтавкаНДС.Получить (ДатаДок);

Количество товара мы можем внести только интерактивно. Но в реквизите **Количество** нужно написать формулы, при которых вычисляется сумма строки и НДС.

Сумма=Цена\*Количество;

```
НДС=Сумма/(СтавкаНДС+100)*СтавкаНДС;
```

Теперь, запустив в пользовательском режиме "1С:Предприятие", попробуйте заполнить строку документа (рис. 7.9).

🌠 1С:Предпр	иятие - Бух и опер учет, расчет - [Поступление то	варов -Новый *]				_ 8	X
🖹 Файл Дей	і́ствия Операции Справочники Документы Журнал	пы Результатыра	счета Отчеты С	бработки Сервис Окна	Помощь	_ 8	×
1 🖻 🔳	1 B B 6 8 B 5 C M	- 4	1 7A 🤋 🗉	і∭ 🕫 🕼 мм+	M-		
	品 取 卧 恒 ▲ 単 幻 ヌヌ ?) №?						
	Поступление товаров № 3	от  03.12.06	3				
Контрагент:			-				
Основание:		Подбор					
N	ТМЦ	Кол-во	Цена 200.00	Сумма 400.00	Ставка Н	НДС	<b>-</b>
2	проверка проверка 1	2.00	200.00	400.00	15.00	52.17	
	hpotopid 1	5.00	500.00	000.00	10.00	10.20	
					L		
		L				L I	-
OK	Закрыты Комментарий:						
📶 Журнал док	ументов Полн						
Для получения	подсказки нажмите F1		NUM	TA: 29.09.06 15:29:36	БИ: 4 квартал	2006 г. ТП: Октябрь 2006 г.	
街 Пуск 🍏	🖙 💿 🎽 🗋 2 Проводник 🗸 📄 9Глава7 - Мі 🛛 💆	] Документ1 🚆	Paint Shop Pro	🐾 Конфигурат 🐼 1С:	Предпр 🛛 🕅	U 🛛 🗘 « 🖻 🗉 🕅 🖓 🗃 🎲 16:	11

Рис. 7.9. Документ, заполненный автоматически

Этот способ достаточно простой, но не всегда эффективный. В реальных конфигурациях вычисление реквизитов с помощью простейших формул часто бывает невозможно; предварительно нужно провести определенную программную обработку данных. Поэтому изучим более совершенный, но сложный способ.

# 7.2. Заполнение документа методом подбора

В *разд.* 7.1 мы рассмотрели возможность заполнения строк документа с помощью формул, которые записаны в палитре свойств реквизитов. Стандартный метод автоматического заполнения документа — метод подбора — требует написания программ в модуле формы документа. Согласно документации, организовать подбор можно с помощью следующих системных процедур (листинг 7.1):

# Листинг 7.1

```
ОткрытьПодбор();
Активизировать();
АктивизироватьСтроку();
```

И предопределенной процедуры ОбработкаПодбора().

В режиме конфигуратора откройте модуль формы документа и создайте новую процедуру (листинг 7.2).

#### Листинг 7.2

```
Процедура Подбор()
```

```
ОткрытьПодбор("Номенклатура", "ФормаСписка");
```

КонецПроцедуры

Метод Открыть Подбор () в нашем случае имеет два параметра: имя справочника, из которого делается подбор, и название экранной формы этого справочника. Мы пока создали единственную экранную форму справочника форму списка. Поэтому ее и указали в параметрах метода. Этот метод откроет выбранную экранную форму справочника.

Создадим следующую предопределенную процедуру (листинг 7.3):

#### Листинг 7.3

Процедура ОбработкаПодбора (Выб)

КонецПроцедуры

Здесь обратите внимание на важный момент. В параметр процедуры мы ставим произвольную переменную выб. Эту переменную в данном случае не требуется предварительно объявлять. Вызов этой предопределенной процедуры происходит в момент выбора товара из формы для подбора. В этот момент в переменную выб происходит подстановка выбранного товара. Процедура ОбработкаПодбора() позволяет производить самые разные действия с заполнением табличной части документа. Сейчас мы организуем вызов других процедур, позволяющих автоматизировать заполнение документа (листинг 7.4). Согласно правилу, описанному в *разд. 2.5*, поскольку новые процедуры будут инициализироваться в процедуре ОбработкаПодбора(), располагаться в модуле они должны ранее ее. Товар из переменной Выб и его количество внесем в табличную часть документа, в колонки ТМЦ и Количество.

Листинг 7.4

Сразу же вернемся к процедуре ОбработкаПодбора() (листинг 7.3) и запишем в ней вызов новой процедуры (листинг 7.5):

Листинг 7.5

Процедура ОбработкаПодбора (Выб)

ЗапросКоличестваВПодборе (Выб);

КонецПроцедуры

Кое-какие конструкции вам уже знакомы, их назначения понятны из комментариев к процедуре. Обратите внимание, что русский синтаксис операторов языка несет смысловую нагрузку, назначение операторов часто понятно из их названия. У нас произошла передача параметра: из процедуры обработкаПодбора() В процедуру ЗапросКоличестваВПодборе(). Мы поместили выбранный товар в переменную выб и передали его в этой переменной, для последующей обработки, в процедуру ЗапросКоличестваВПодборе. Для этого переменная выб была установлена в качестве параметра обеих процедур и при вызове процедуры ЗапросКоличестваВПодборе(). Именно для резервирования места под возможные параметры мы и должны, согласно правилам языка, в конце наименования процедуры или функции писать круглые скобки. Условный оператор Если... Тогда... КонецЕсли реализует защиту от выбора товара, помеченного как удаленный. Процедура прекратит свою работу с соответствующим предупреждением, и ошибки учета удастся избежать.

Прекращение работы процедуры осуществляется ключевым словом возврат.

Для интерактивного ввода количества товара мы использовали системную функцию ВвестиЧисло(), ее описание посмотрите в синтакс-помощнике или в документации к программе. Эта функция организует диалог при вводе с клавиатуры числового значения (рис. 7.10), в нашем случае — количества товара.

🕼 ICПредприятие - Буж и опер учет, расчет - [Номениклатура (Подбор) (03.12.06)]												
		- <b>6</b> 8	?	1 🕫 🔟	M M+ M-							
		3	- 111	_, , _ ,								
Kon	Краткое наименование		IE	Единица из Тип номенклатуры								
3	проверка											
4	проверка 1 тов2											
1	товар1	Введите количество	×									
							<u> </u>					
Excel	Экспорт информации	Пекст										
Печать	Кодовая стран	iųa į ".dbr										
📶 Журнал документов Полн 🗈 Поступление товаров -Но 🖉 Номенклатура (Подбо												
Выполняется обработка			15:29	:36 БИ: 4 квар	тал 2006 г.	TП: Октябрь 2006 г.						
🖞 Пуск 🧉 🗋 🕤	» 📄 2 Проводник 🛛 🖭 9Глава7	- Мі 🖳 Документ1 🗮 Paint St	iop Pro 🍕	🕻 Конфигурат	🕰 1С:Предпр	RU (?) 🖡 (	× 🔢 K 🖵 🗃 🏟 16:13					

Рис. 7.10. Диалог для ввода числа

Синтаксис функции прост: в параметрах через запятую указываем:

- некую произвольную переменную, в которую будет помещено то число, которое мы наберем на клавиатуре;
- подсказку заголовок окна диалога;
- максимальную длину вводимого числа;
- его точность.

Согласно синтаксису этой функции, мы записали следующее: в произвольную переменную кол помещается число с максимальной длиной целой части 7 знаков, длиной дробной части — 2 знака. Один знак зарезервирован для разделителя целой и дробной части числа. Дали заголовок окну диалога (см. рис. 7.10).

Перед диалогом ввода мы программно занесли в переменную кол значение 1, которое и высвечивается при выводе диалога. Если у нас действительно оперируют хотя бы иногда единичными товарами, то благодаря этой строке программы мы экономим время пользователя.

Метод НоваяСтрока () добавляет новую строку в табличную часть документа. В эту строку затем вставляется наименование товара и его количество, что понятно из комментариев к процедуре.

Название товара и его количество внесены в документ. Далее из справочника "Номенклатура" внесем цену товара и ставку НДС (листинг 7.6).

# Листинг 7.6

```
Процедура УстанЦеныПрих ()
  Цена= ТМЦ. Себестоимость. Получить (ДатаДок);
                                              //Себестоимость товара из спр.
                                              //Номенклатура – в документ
  СтавкаНДС= ТМЦ.СтавкаНДС.Получить (ДатаДок);//Ставку НДС из спр.
                                               //Номенклатура — в документ
```

КонецПроцедуры

Обратите внимание: реквизит документа ТМЦ имеет тип значения "Справочник.Номенклатура". А в этом справочнике у нас есть и себестоимость, и ставка НДС. Смысл строк нашей процедуры — в реквизит Цена помещается себестоимость товара, выбранного в справочнике "Номенклатура" с помощью процедуры ОбработкаПодбора(). Аналогично для ставки НДС. Процедуру УстанЦеныПрих Также вызываем из процедуры ОбработкаПодбора ().

Последняя процедура — организация заполнения вычисляемых реквизитов документа. Это Сумма и НДС.

### Листинг 7.7

```
Процедура Выч_суммы_накл()
Сумма=Цена*Количество;
НДС=Сумма/(СтавкаНДС+100)*СтавкаНДС;
КонецПроцедуры
```

Листинг 7.7 демонстрирует возможность работы в 1С со стандартными арифметическими выражениями. Процедуру заполнения вычисляемых реквизитов мы также вызываем из ОбработкаПодбора () (листинг 7.3), ее окончательный вид представлен в листинге 7.8.

### Листинг 7.8

```
Процедура ОбработкаПодбора (Выб) //Элемент спр. Номенклатура —
//в переменную Выб
ЗапросКоличестваВПодборе (Выб); //передаем параметр
УстанЦеныПрих(); //вызов процедур обработки подбора
Выч_суммы_накл();
АктивизироватьСтроку();
КонецПроцедуры
```

Метод подбора АктивизироватьСтроку () предназначен для установки курсора в данном случае на текущей строке многострочной части документа.

Перейдите в диалог экранной формы. Создайте в нижней части формы новую кнопку с помощью инструмента **Кнопка** панели инструментов **Элемен**ты диалога или с помощью меню **Вставить** | **Кнопка**.

В палитре свойств кнопки задайте заголовок **Подбор**, в формуле — ссылку на процедуру Подбор(). Теперь сохраните конфигурацию и запустите ее в пользовательском режиме (рис. 7.11). При нажатии кнопки **Подбор** вы автоматически заполняете всю строку документа. Именно подобным образом организовано заполнение документов в конфигурациях "1С:Предприятие". Вы можете сравнить рабочую конфигурацию (копию) с учебной и, при необходимости, внести изменения в процедуры подбора. Правда, в реальных конфигурациях эти процедуры вынесены в глобальный модуль. В поле формул вкладки **Дополнительно** палитры свойств объекта можно записывать не только формулы, но и ссылки на процедуры, которые будут выполняться при активизации этого объекта.

	Выч_Суммы_Накл()							
ЗапросКоличестваВПодб	ope() Y	⁄станЦен	ыПрих()					
Осториятие - Бух и опер учет, расчет - [Поступление та файа Действиа Операции. Справодники Догуденты Хулика В файа Действиа. Операции.	оваров -Новый *] им. Результаты ра	чета Отчеты С	foaforrol Cepsur Orna	Помощь		_ 6 ×		
	vior resynetiatelpa	2 2 9 1 =	IMI 69 NO M M+	M-				
여러국립말과라 <sup>[1]</sup> · 44 전 3 년								
Поступления/товаров № 3 Контрагент: Макеления Основные	от 03.12.06 <u>с</u> Повбер	3						
	Кол-во	Цена	Сумма	Ставка Н Н	дс			
1 проверка 2 проверка	2.00	200.00	400.0	15.00		52.17		
3 проверка 1	1.00	300.00	300.0			70.20		
				+				
				+				
ОК Закрыть Коменентарий:				· · · ·				
МУхрыая документов Полн 🕪 Поступление товаров								
лат побора значения нахинте клапицу F4, для очистки значения - Shi	t+F4.	NUM	A: 29.09.06 15:29:36	БИ: 4 квартал 20	06 г. 111: Октябр	ь 2006 г.		
🕅 Пуск 🚎 🗁 🕞 » 🗀 2 Проводник - 🗟 🕅 9Глава7 - М 🗟	Документ1	Paint Shop Pro	🧏 Конфигурат 📈 10	Предпр 💷	2 K	33 16:15		

Рис. 7.11. Документ "Поступление товаров"

# 7.3. Локальный контекст и доступность процедур и функций

Процедуры и функции многих документов похожи. Например, если внимательно сравнить модули формы документов приходной и расходной накладной, то убедимся, что различия невелики: в одном документе имеется процедура установки цены прихода, а в другом — установки цены расхода. Соответственно отличается процедура ОбработкаПодбора(). А процедура подсчета НДС одинакова. Стоит ли в таком случае переписывать или копировать одинаковые процедуры из документа в документ? Уже сейчас общая длина текста процедур в модуле формы документов составляет десятки строк. Конфигурация будет усложняться, количество процедур и строк — расти. Чтобы быстро и правильно ориентироваться в модуле, стоит подумать о путях сокращения длины и количества процедур. Наконец, процедуры в модулях ветвей метаданных выполняются сравнительно медленно. А скорость выполнения, особенно для системы программ "1С:Предприятие", может иметь решающее значение. Ведь в реальном учете на предприятии бывает необходимо оперировать с сотнями тысяч записей и десятками тысяч документов. И если оператор выписывает целый день накладные, то от скорости их обработки напрямую зависит и оборот предприятия. Представьте, что выписывается 1000 накладных в день, и на каждой из-за неправильного алгоритма теряется дополнительно 2 секунды времени — значит, более получаса рабочего времени ушло впустую. Посчитайте, сколько времени потрачено зря за год — это серьезный убыток для предприятия.

Самая простая возможность системы "1С:Предприятие" ускорить исполнение программы и облегчить работу программисту — вынести часто повторяющиеся процедуры и функции в уже знакомый нам глобальный модуль конфигурации.

Перенесем в глобальный модуль наиболее общие для создаваемых нами документов процедуры: ЗапросКоличестваВПодборе (Выб), Выч\_суммы\_накл(). Для этого вырежем их из модуля формы документов и вставим в глобальный модуль между процедурами Приветствие() и ПриЗавершенииРаботыСистемы(). Теперь необходимо, чтобы эти процедуры были доступны из других программных модулей. В нашем примере — из модуля формы документов. Для этого после наименования процедуры и параметров пишется ключевое слово Экспорт (листинг 7.9).

### Листинг 7.9

```
Процедура ЗапросКоличестваВПодборе(Выб) Экспорт // интерактивно
// выставляем количество
.....
КонецПроцедуры
Процедура Выч_суммы_накл() Экспорт
```

. . . . . . . . .

КонецПроцедуры

# Предупреждение

Ключевое слово Экспорт имеет смысл только в глобальном программном модуле.

Теперь нужно указать, откуда эти процедуры будут получать значения параметров и переменных для вычислений. Ведь они видны всем документам, и нужно в конкретный момент указать, что работа ведется с приходной либо расходной накладной. Для этого в языке существует возможность передавать локальный контекст программного модуля в качестве параметра процедуры или функции.

Обратимся к модулю формы документа "Поступление Товаров". Укажем, что в процедуры глобального модуля ЗапросКоличестваВПодборе (Выб) Экспорт и Выч\_суммы\_накл() Экспорт необходимо передавать данные из локального модуля формы документа. Для этого в качестве фактического параметра при вызове процедур и функций используется ключевое слово Контекст. Эти процедуры вызываются из предопределенной процедуры модуля формы обработкаПодбора (Выб). Добавим в вызовы процедур ключевое слово Контекст (листинг 7.10).

### Листинг 7.10

```
Процедура ОбработкаПодбора (Выб) //Элемент спр. Номенклатура –
//в переменную Выб
ЗапросКоличестваВПодборе (Выб, Контекст); // передаем параметр
УстанЦеныПрих();//вызов процедур обработки подбора
Выч_суммы_накл (Контекст);
АктивизироватьСтроку();
КонецПроцедуры
```

Вернемся в глобальный модуль и внесем дополнения в глобальные процедуры. Допишем глобальный параметр, добавим, к примеру, конт (рис. 7.12). Формальному параметру конт глобальной процедуры присваивается локальный контекст, т. е. указывается, из какого программного модуля какой ветви метаданных передаются значения для вычислений. Кроме того, на протяжении всей процедуры мы должны указывать, когда работаем с переменными, командами и т. п. глобальной процедуры, а когда — с переданными в нее локальными. Для этого в глобальной процедуре мы обращаемся к локальной части через параметр конт с точкой, как в листинге 7.11.

### Листинг 7.11

```
Процедура Выч_суммы_накл(Конт) Экспорт
Конт.Сумма=Конт.Цена*Конт.Количество;
Конт.НДС=Конт.Сумма/(Конт.СтавкаНДС+100)*Конт.СтавкаНДС;
КонецПроцедуры
```


Рис. 7.12. Передача локального контекста программного модуля в качестве параметра

Эта глобальная процедура оперирует только локальными значениями, которые передаются либо из документа "ПоступлениеТоваров", либо из документа "РеализацияТоваров", либо из одного из будущих документов, ссылка на который попадет в параметр конт. Значения суммы и НДС, наоборот, будут отправлены в нужный локальный модуль по параметру конт.

Процедура ЗапросКоличестваВПодборе (Выб, Конт) Экспорт имеет операторы, относящиеся к локальному контексту, и операторы глобальной процедуры.

#### Листинг 7.12

Процедура ЗапросКоличестваВПодборе(Выб, Конт) Экспорт // интерактивно //выставляем количество

Если Выб.ПометкаУдаления()=1 Тогда //товар помечен на удаление
Предупреждение("Нельзя выбирать помеченные на удаление товары!");
Возврат; //прекращаем работу процедуры
КонецЕсли;
Кол=1; //Объявили переменную для ввода количества и присвоили //ей значение "1"
Если ВвестиЧисло(Кол, "Введите количество", 10, 2)=1 Тогда
Конт.НоваяСтрока(); // добавляем строку в табличную часть
Конт.ТМЦ=Выб; //Товар из Выб в добавленную строку
Конт.Количество=Кол;//Количество- туда же
КонецЕсли;
КонецПроцедуры

Конструкция проверки товара на удаление, ввод количества через переменную кол не нуждаются в передаче в локальный модуль. А вот добавление новой строки в табличную часть документа требует указания на то, с каким документом работает программа. Поэтому перед оператором указывается параметр локального контекста (листинг 7.13).

### Листинг 7.13

```
Конт.НоваяСтрока();
```

Таким образом, для работы с процедурами глобального модуля мы должны соблюдать следующие правила:

- **П** заголовок глобальной процедуры заканчивается ключевым словом Экспорт;
- □ в параметры глобальной процедуры добавляется параметр для присваивания глобального контекста конт;
- в параметры локальной процедуры добавляется ключевое слово Контекст;
- обращение к реквизитам и методам локального контекста происходит "через точку" после идентификатора Конт.

## 7.4. Итоговые значения колонок документа

Для повышения информативности документа было бы удобно иметь возможность получения итогов по колонкам табличной части документа. В документе "ПоступлениеТоваров" итоги интересны по колонкам Сумма и **НДС**. Конфигуратор "1С:Предприятие" позволяет получать итоги по колонкам без составления отдельных процедур на встроенном языке. Операторы встроенного языка потребуются только для обработки итогов.

Для того чтобы автоматически получать итог по колонке, необходимо перейти в окно редактирования документа "Приходная накладная". Затем выбрать реквизит табличной части (Сумма) и нажать кнопку Изменить. Откроется палитра свойств реквизита (рис. 7.13). Перейдите на вкладку Дополнительные и установите флажок Итог по колонке. Повторите то же самое для реквизита НДС.



Рис. 7.13. Установка флажка Итог по колонке

В экранной форме документов под табличной частью добавьте элемент диалога **Текст** (с помощью панели инструментов или меню). В палитре свойств текста очистите заголовок. Идентификатор нам тоже не нужен. Щелкните на вкладке **Дополнительно** и в поле формул напишите: **Итог (Сумма).** Создайте рядом еще один элемент диалога **Текст** и запишите в нем формулу

**Итог (НДС)**. Расположите оба элемента в форме так, чтобы они были удобны для восприятия. Запустите "1С:Предприятие" в пользовательском режиме, создайте несколько документов "Поступление Товаров" с произвольными значениями. Итоговые значения колонок будут автоматически посчитаны и выведены в экранной форме документа.

### 7.5. Печатные формы документов

Печатные формы документов создаются аналогично печатным формам справочников, хотя есть и определенные отличия.

В режиме конфигуратора необходимо открыть окно редактирования формы документа "ПоступлениеТоваров". Здесь начнем работать с вкладкой **Табли**ца. Будем создавать шаблон печатной формы документа так же, как описывалось в предыдущем разделе.

Большинство шаблонов печатной формы документа имеют три стандартные горизонтальные секции:

- □ Шапка для вывода в печатную форму заголовочной части документа;
- □ Строка для вывода многострочной части документа (если таковая имеется);

Подвал — для вывода итоговой части документа (если таковая имеется).

Как правило, секция Шапка существует всегда, остальные две — по необходимости.

В ячейке секции Шапка напишем заголовок документа, его номер и дату создания, использовав тип текста Шаблон (рис. 7.14).

Это сделано для того, чтобы заголовок, включающий обычный текст (наименование документа), и вычисляемое выражение (номер документа и дата документа) выглядели, как единое целое. Если заголовок, номер и дату документа разместить в отдельных ячейках, то потребуется дополнительная подгонка ширины колонок, и, тем не менее, заголовок все равно останется некрасивым.

Пропустим строку в секции Шапка и заполним названия колонок для многострочной части документа. Тип текста — Текст.

### Примечание

Для того чтобы вставить новую строку, нужно щелчком мыши выделить строку, выше которой предполагается вставка, и выбрать команды меню **Действия | Раздвинуть**.

🖉 Конфиг	урато	р-Бужи	опер учет, расчет - [Фо	рма-Документ.Постуг	лениеТова	ров]						
🕒 Файл	Дейст	вия Таб	ілица Вид Конфигурация	Конструкторы Админ	истрирование	е Сервис Окн	на Помощь					_ 8 ×
1 10		% Be (	868 8 > 0	<i>8</i> 4	• 6	¥ 8						
	<b>2</b> .8	n 19	® ⊖   \$   € €	à à à à à	<b>1</b> 8 <b>1</b> 8 <b>1</b>	<b>* * * *</b>	à à					
R2C2			-щ = Поступлен	ие товара № [НомерДок]	от [ДатаДок]							
		1	1		3	4	5	6	7	8	9	10 🔺
Шапка	1				-							
	2		СПОСТУПЛЕНИЕ	товара № [Но	јиерДо	к] от [Да	гаДок]>					
	3											
	4		Th	Свойства ячейки				×				
	5	Nº	IR	Tever		1			ндс			
Строка	7	<Номер(	<tmli< td=""><td>Текст Положение</td><td>шрифт   Ра</td><td>мка   930р  </td><td>1</td><td></td><td>и нлс&gt;</td><td></td><td></td><td></td></tmli<>	Текст Положение	шрифт   Ра	мка   930р	1		и нлс>			
	8			Тип Шаблон	<ul> <li>Контрол</li> </ul>	ь Авто	▼					
Подвал	9			Постипление товар	a Nº (Homen I	ок1 от Пата Пон	1					
	10			-			* =	0r [				
	12			-			<b>V</b>					
	13			Расшифровка				Отмена				
	14		Отпустил	. Зашита				06				
	15							ООНОВИТЬ				
	16											
	18											
	19											
	20											
	21											
	22											
	24											
	25											
	26											
	27											
	29											
	30											
	31							_				
•	32	1								1 1		
. С. Диал	or	λE	Модчль 🗼 🕅 Таб	лица /								
	E	1 S2 100	l   k      C ] ₽	EFC 10F								
च∕्च Конфиг	урация		📄 Документ Поступл	ениеТо	окумент.По	ст 📋 Глоба	льный модуль					
Для получе	ния по,	асказки н	ажмите F1							NUM	статус: БД	, Документ
<i>В</i> Пуск	ൽ ര	e »	2 Проводник 🚽 🕅 1	9Глава7 - Micr 🛛 🖃 Док	хмент1	Paint Shop	Pro Sec K	онфигурат	CorelDRAW 1	0 811 12	9 8	0 31 K 16:49

Рис. 7.14. Наименование документа имеет тип текста Шаблон

В секцию **Строка** помещаются значения из многострочной части документа. В отличие от справочника, при работе с документом не требуется создания объектной переменной, поэтому просто вставим названия реквизитов. Тип текста — **Выражение**.

В секцию **Подвал** помещаются итоговые значения документа, подписи и т. п. Все то, что обычно располагается в нижней части бумажного документа. В нашей печатной форме это будут итоги по колонкам суммы и НДС и подписи ответственных лиц в графах **Отпустил** и **Получил** (рис. 7.15).

Обратите внимание на синтаксис операторов для работы с итогами по колонкам. Для того чтобы получить итог по колонке (признак итогов должен быть предварительно установлен для реквизита, как указано в *разд.* 7.4), необходимо записать оператор итог, а далее, в скобках и кавычках, — наименование колонки:

```
Итог("Сумма");
```

- Koudur	VDato	n - Evy		VIRANUATORA	land						
Д Файл	Действ	зия Таб	лица Вид Конфигурация Конструкторы Адм	инистрирование	Сервис О	кна Помощь					_ @ ×
12 6		X 🗈 (	868 M 3 C A	- - -	8 8						
   [	172 3	ы <b>1</b> 85			8 - 28 - 18 - 18 - 18 - 18 - 18 - 18 - 1	( 茶茶)					
R20C5											
		1	2	3	4	5	6	7	8	9	10 🔺
Шапка	1										
	2		<Поступление товара № [Н	ІомерДо	к] от [Да	атаДок]>					
	3								-		
	5	N≘	ТМЦ	Кол-во	Цена	Сумма	Ст. НДС	НДС			
Comment	6		- THE				0				
строка	8	<номер	ч<тмц	<количеств	<цена>	<∪умма>	«ставкандо	«ндс»			
Подвал	9										
	10					<Сумма (Ито	r("Сумма") - шлсги	>			
	11					<в том числ	е ндсјито	г ("НДС") >	1		
	13										
	14		Отпустил	_	Получил			_			
	15										
	17										
	18										
	19					-	1				
	20					-					
	22										
	23								-		
	24										
	26										
	27										
	28										
	30								1		
	31										
•						· · · · · · · · · · · · · · · · · · ·					
СЭДиал	or E 🕰	\ [] Ω. 60	Модуль <u>М</u> Таблица 1 🗟 🔤 🛯 🗗 🖽 🖵 🗂 Хо Г	=							
च⁄∃Конфиг	урация	*	🖹 Документ ПоступлениеТо	Документ.По	ст 📋 Глоб	альный модуль					
Для получе	ния под	цсказки н	нажмите F1						NUM	СТАТУС: Б	ц, Документ *
<i>1</i> Нуск	ø 0	• • ×	2 Проводник - 🕑 9Глава7 - Місг 🖉 Д	окумент1	Paint Sho	ор Pro 🛛 😪 Кон	фигурат	CorelDRAW 1	0 RU (	7 <b>7</b> «	4 BI 🌾 16:53

**Рис. 7.15.** Шаблон печатной формы документа "ПоступлениеТоваров"

Ячейки секции Подвал, содержащие итоги, имеют тип текста Шаблон, так как содержат и текст, и выражение.

Теперь можно отредактировать шаблон печатной формы и подобрать шрифты.

Процедуру для вывода документа в печатную форму мы напишем в модуле формы документа (листинг 7.14).

### Листинг 7.14

```
Процедура Печать()
Таб=СоздатьОбъект ("Таблица");
Таб.ИсходнаяТаблица ("Таблица");
Таб.ВывестиСекцию("Шапка");
Если ВыбратьСтроки()=1 Тогда // многострочная часть позиционирована
Пока ПолучитьСтроку()=1 Цикл
```

```
Таб.ВывестиСекцию ("Строка");
КонецЦикла;
КонецЕсли;
Таб.ВывестиСекцию ("Подвал");
Таб.ТолькоПросмотр(1);
Таб.Опции(0,0,0,0);
Таб.Показать ("Приходная накладная");
КонецПроцедуры
```

Все строки процедуры, относящиеся к шаблону печатной формы, идентичны строкам процедуры печати справочников. Появилась новая строка для вывода секции Подвал. Нет объектных переменных для документа — они не нужны. Кроме того, отличаются операторы выборки: вместо ВыбратьЭлементы() — ВыбратьСтроки(), вместо ПолучитьЭлемент() — ПолучитьСтроку().

221 C D	равлонатиа - Бих и одар ушат, расцат - [Прихо		awaa *1									
கி நல	іредприятие бухи оперучет, расчет – піриход іп Лействия Табрица Вил Операции Справоци	ики Докуме	аноя ј анты Журнал	. Результать	прасчета (	Отнеты О	бработки	Сереис Окн	а Помош			
		ики докуно	лины журныл			mentiline.	opuboritar	- Copune Okn		0	-	
jn ⊡ ⊑	- L   & B B   O C A   B D C A		•	前前一岁		\$P 14	M M+	M-				
												<b>^</b>
	Поступление товара № 2 с	от 27.09	9.06									
No	T TMU	Кол.во	Пена	Сумма	Ст. НЛ	с н	лс	1				
1	Телевизор	1	2000	2000	0	0						
2	Холодильник	2	2400	4800	0	0		]				
				Cuuua 680	n							
				В том числ	ь 1е НДС О							
	0											
	Отпустил		Получил									
												-
•												
∖ ⊏		ff 🖂 🏹	3 FF									
📶 Жург	нал документов Полн 📄 Поступление товаров - 2	* 🔊 Прия	юдная накла	дная								
Для полу	учения подсказки нажните F1				NUM TA: 29	.09.06 15:2	29:36	БИ: 4 кварта	ал 2006 г.	ТП	: Октябрь 2006 г.	
â) Пусі	к 🍏 🏹 🕞 🎽 🗋 2 Проводник 🔹 👿 2 Містоз	soft 🔹 🚆	Paint Shop Pr	о 🥰 Конф	игурато	🖋 1С:Пр	едпри	CorelDRA	₩ 10	RU (?)	• • • •	17:08

**Рис. 7.16.** Печатная форма документа "Поступление товаров"

Кнопка **Печать** на экранную форму документа вставляется так же, как и на экранную форму справочника.

### Примечание

Правила работы с кнопками одинаковы для всех экранных форм "1С:Предприятия".

Запустив "1С:Предприятие" в пользовательском режиме, проверим сделанную нами работу (рис. 7.16).

Теперь автоматизируем процесс создания процедуры печати с помощью конструктора печати.

В режиме конфигуратора откройте форму документа "ПоступлениеТоваров". Находясь в модуле формы документа, выберите пункт меню Конструкторы | Печать. Появится поле ввода с именем готовящейся процедуры, по умолчанию — Печать. Нажмите кнопку ОК — появится первое окно диалоговой формы конструктора печати — Шапка (рис. 7.17). В правой части окна перечисляются реквизиты шапки документа, в левой — те реквизиты, которые мы будем использовать в шаблоне печатной формы.



Перенос реквизитов из правой части в левую осуществляется с помощью стрелок. По умолчанию в левой части окна уже есть обязательные реквизиты **НомерДок** и **ДатаДок**. Нам нужно перенести реквизит **Контрагент**. Выделите его щелчком левой кнопки мыши и нажмите кнопку . Точно так же перенесите реквизит **Основание**. Нажмите кнопку **Далее** и перенесите все реквизиты табличной части документа из правой части диалога в левую с помощью кнопки . Нажмите кнопку Далее и переместите в левую часть диалога реквизиты, которые будут помещены в подвал шаблона печатной формы документа: **Итог(Сумма)** и **Итог(Количество)**. Нажмите кнопку **Далее**, выставьте опции диалога, как указано на рис. 7.18, и нажмите кнопку **Готово**.



Рис. 7.18. Установка параметров печати и кнопок

Посмотрите результат работы конструктора в Конфигураторе и запустите "1С:Предприятие" в пользовательском режиме (рис. 7.19). Конструктор создает новый шаблон печатной формы с наименованием ПечатьАвто. Можно вручную произвести необходимые изменения в шаблоне печатной формы и в процедуре. При создании сложных документов объем редактирования шаблона печатной формы и процедуры возрастает, но в любом случае данный конструктор представляет дополнительные удобства для программиста.

100																																	
ि त	апред Эайл	прия 1ейств	ие-с	абли.	unej ia E	луче Ил П	г, ра Опер	ашии	Cnr	авочн	ь пос ники	Док	ления умент	атов њ. Х	аров Курна	алы алы	Pesv	льтат	ъ ра	счета	a 0'	гчеть	a Of	ірабо	тки	Cede	ис С	жна	Помоц	њ			- 레스
139	r≩∎	3	( Ba	R		40		1.5	è	AA [			,				- 12	ାଡ		-	 36	<b>201</b>	m	м	M+	M-							
	Пос Номер: Дата: 3 Контра Комме Основ:	2 27.09.0 гент: г нтари ание: д	<b>ілен</b> 6 маывл 7: огово	ние лыв	то	Ba	ро	B															-										-
	Ne			TN	ſЦ			Ko	пиче	ство		Цe	на			Сум	ма	a	вка	Н		НДС											
		Теле	визор	MP.				-		1			2	2000				2000		0				0									
	U T or C	умма:	6800																														
			-	er l	N		l e	7		0000	HTT:		8.4						_		_			_	_								
]] >					13		L	7	e,		8.1			11-				-															
Шж	урнал,	докуме	ентов І	Полн.	[	] Noc	тупл	ение	товар	ов - 2	2	≜]⊓e	ечаті	ь Пос	туп	лени	е т																
Для п	олучен	ия под	сказкі	и наж	мите	F1													NU	M TA	: 29.0	09.06	15:2	9:36		БИ:	4 квар	отал 2	006 r.		TTI: C	октябрь 200	06 г.
₫₽⊓	уск 👔	9 Ca	$\odot$	» (	21	Трово	дник		W 2	Micro	soft .		1	Paint :	Shop	Pro	19	Кон	фигу	рато.		1	С:Про	едпр	и	3	CorelD	RAW 1	0	RU	2 5	< 15 24	17:16

Рис. 7.19. Печатная форма документа "Поступление товаров", созданная с помощью конструктора

# 7.6. Печатные формы как отдельные файлы

До сих пор мы рассматривали создание печатных форм справочников и документов с использованием шаблонов в составе конфигурации. Это означает, что любые изменения, которые необходимо вносить в такой шаблон, требуют работы в режиме Конфигуратора. Существуют ситуации, когда удобнее редактировать шаблон, не трогая конфигурацию. Например, те отчетные формы, которые достаточно часто изменяются в соответствии с изменением законодательства. Или в случаях, когда пользователь сам вносит часто изменяющиеся данные: рекламный текст на счете, извещении об оплате за услуги. Для того чтобы создать отдельный файл печатной формы, войдите в режиме Конфигуратора в меню **Файл** и выберите пункт **Новый**. В появившемся диалоговом окне выберите пункт **Таблица** (рис. 7.20) и нажмите кнопку **ОК**.



Рис. 7.20. Диалоговое окно для выбора вида внешнего файла

Внешний вид шаблона печатной формы и правила работы с ним ничем не отличаются от работы с шаблонами печатной формы объектов метаданных. Созданный шаблон для удобства будем называть "внешней таблицей".

Откройте шаблон печатной формы документа "ПоступлениеТоваров", скопируйте содержимое в буфер обмена и вставьте его в открытый шаблон внешней таблицы. Копирование и последующая вставка производятся с помощью кнопки, находящейся в шаблоне печатной формы на пересечении номеров колонок и номеров строк.

Создайте в каталоге учебной базы папку Table и сохраните внешнюю таблицу в этой папке с помощью меню **Файл | Сохранить**. При сохранении дайте имя внешней таблице ПрихНакл.mxl.

В процедуре печати модуля формы документа "Поступление Товароов" внесите следующие изменения (листинг 7.15).

### Листинг 7.15

```
Процедура Печать()

Таб=СоздатьОбъект("Таблица");

// Таб.ИсходнаяТаблица("Таблица");

Таб.ИсходнаяТаблица("Table\ПрихНакл.mxl");

Таб.ВывестиСекцию("Шапка");

Если ВыбратьСтроки()=1 Тогда // справочник позиционирован

Пока ПолучитьСтроку()=1 Цикл

Таб.ВывестиСекцию("Строка");

КонецЦикла;

КонецЕсли;

Таб.ВывестиСекцию("Подвал");

Таб.ТолькоПросмотр(1);

Таб.Опции(0,0,0,0);

Таб.Показать("Приходная накладная");

КонецПроцедуры
```

Теперь необходимо указать путь к шаблону печатной формы. Если папка Table расположена не в каталоге рабочей базы, необходимо указывать полный путь к файлу, начиная с имени диска.

### Предупреждение

Название внешней таблицы обязательно указывать с расширением mxl.

Проверьте работу процедуры в пользовательском режиме. Попробуйте открыть файл и внести изменения в шаблон печатной формы. Сохраните шаблон. Закройте "1С:Предприятие", снова запустите его и проверьте внесенные изменения.

### 7.7. Список значений

Специальный агрегатный тип данных "Список значений" используется в особых случаях. Например, как указывается в документации к системе программ "1С:Предприятие", при формировании пользовательского интерфейса прикладной задачи, когда для удобства работы необходимо дать пользователю возможность выбирать какое-либо значение из заранее подготовленного списка. При этом невозможно или неудобно получить эти значения непосредственно при работе со стандартными визуальными представлениями журналов, справочников и перечислений.

Объект типа "Список значений" — это средство языка, он не сохраняется на жестком диске компьютера, а существует только в оперативной памяти во время исполнения соответствующей процедуры.

Используем список значений в нашем учебном примере для лучшей иллюстрации возможности печати документа. Мы осуществляли вывод печатной формы документа, используя встроенный в конфигуратор шаблон или внешний файл. Объединим обе возможности и предоставим пользователю выбор, какой из шаблонов печатной формы можно использовать. При нажатии кнопки **Печать** пользователь должен увидеть дополнительные опции вывода печатной формы: Встроенная печатная форма или Внешняя печатная форма.

Откройте модуль формы документа ПоступлениеТоваров и перейдите к процедуре Печать (). Так же, как и при работе со справочниками во встроенном языке, создайте объектную переменную Выбор, которой присвойте ссылку на объект Список значений:

```
Выбор=СоздатьОбъект ("СписокЗначений");
```

#### Теперь пополните этот список необходимыми нам значениями:

```
Выбор.ДобавитьЗначение(1, "Встроенная печатная форма");
```

```
Выбор.Добавить Значение (2, "Внешняя печатная форма");
```

Выражение (1 или 2) используется для указания системе выбранного значения, а символьное представление — для удобства сортировки и выбора значений из списка (отображается в диалоговом окне выбора).

Организуем выбор значения из списка:

```
Рез=Выбор.ВыбратьЗначение(Код, "", Поз,,1);
```

Метод ВыбратьЗначение() открывает окно для интерактивного выбора значений из списка. Он имеет следующие параметры:

- Значение идентификатор переменной, куда помещается результат выбора;
- □ Заголовок строковое выражение заголовка диалогового окна. Может использоваться для подсказки пользователю;
- позиция идентификатор переменной, куда помещается номер позиции выбранного значения в списке;

- Таймаут необязательный параметр. Числовое выражение, значение которого задает время ожидания системы (в секундах) на отклик пользователя;
- СпособВыбора необязательный параметр. Числовое выражение, значение которого задает способ выбора значения:
  - 0 в виде диалога, задан по умолчанию;
  - 1 выбор производится в виде меню, которое подстраивается по месту текущего элемента диалога или ячейки таблицы, откуда вызвано;
  - 2 выбор небольшим списком (список похож на выбор значения перечисления), также привязанным к позиции элемента диалога.

Возвращаемое значение метода — число:

- □ 1 если выбор произведен (нажата кнопка **ОК**);
- 0 если выбор не произведен (нажата кнопка ОТМЕНА);
- -1 закончилось время.

Для нашего случая необходимы параметры Значение, Позиция, СпособВыбора и возвращаемое значение.

Выбранное пользователем значение из списка попадет в переменную код, позиция — в переменную Поз, Способвыбора — 1.

Так как переменные поз и код не имеют справа от себя знака присваивания, то их требуется объявить в начале процедуры.

В переменную Рез у нас попадает возвращаемое значение. Если пользователь сделал выбор из списка, то значение созданного списка (1 или 2) попадет в переменную код. Если переменная код равна 1, тогда надо вывести встроенный шаблон печатной формы, если 2 — внешний файл. Окончательный вариант процедуры Печать должен получиться следующим (листинг 7.16):

### Листинг 7.16

```
Процедура Печать()
Перем код;Перем поз; //объявление переменных для метода
//"ВыбратьЗначение()"
Выбор=СоздатьОбъект("СписокЗначений");
Выбор.ДобавитьЗначение(1,"Встроенная печатная форма");
Выбор.ДобавитьЗначение(2,"Внешняя печатная форма");
```

```
Рез=Выбор.ВыбратьЗначение (Код, "", Поз, 1);
  Если Рез<>1 Тогда Возврат;КонецЕсли;
  Если Код=1 Тогда // пользователь выбрал первое значение
  Таб=СоздатьОбъект ("Таблица");
  Таб.ИсходнаяТаблица("Таблица");
   Таб.ВывестиСекцию ("Шапка");
   Если ВыбратьСтроки()=1 Тогда // справочник позиционирован
     Пока ПолучитьСтроку()=1 Цикл
      Таб.ВывестиСекцию("Строка");
     КонецЦикла;
   КонецЕсли;
  Таб.ВывестиСекцию("Подвал");
   Таб.ТолькоПросмотр(1);
  Таб.Опции (0,0,0,0);
  Таб.Показать ("Приходная накладная");
  КонецЕсли;
  Если Код=2 Тогда // пользователь выбрал второе значение
  Таб=СоздатьОбъект ("Таблица");
   //Таб.ИсходнаяТаблица ("Таблица");
  Таб.ИсходнаяТаблица ("Table\ПрихНакл.mxl");
  Таб.ВывестиСекцию ("Шапка");
   Если ВыбратьСтроки()=1 Тогда // справочник позиционирован
     Пока ПолучитьСтроку()=1 Цикл
      Таб.ВывестиСекцию("Строка");
     КонецЦикла;
   КонецЕсли;
  Таб. Вывести Секцию ("Подвал");
  Таб.ТолькоПросмотр(1);
  Таб.Опции (0,0,0,0);
  Таб.Показать ("Приходная накладная");
  КонецЕсли;
КонецПроцедуры
```

Для проверки запустите "1С:Предприятие" в пользовательском режиме. При нажатии кнопки **Печать** должно появиться меню (рис. 7.21).

	_
🧊 ICS Предприятие - Бух и опер учет, расчет 📑	×
чаил деиствия операции справочники документы журналы результаты расчета отчеты образотки сервис окна понощь	
「ひを聞 × 予読 金 通 り c A A 「 」 」 A A A S 「 国 世 S 「 U M M M M M	
и Журнал докунентов Полный (01.01.06-03.12.06)	
🖹 Поступление товаров - 2	
留俱想」到另可は★★如此の必	
Поступление товаров № 2 от 27.09.06	
Kowroaneer	
N         Imputes         KOFEO         Leftal         Imputes         Impute	
2 Холадильник 2.00 2.400.00 4,800.00	
ИК Закрыть Печать Коннентарий	
остроенная печатная офрива Внешная печатная форма	
📶 Журнал документов Полн 🗈 Поступление товаров	
NUM TA: 29.09.06 15:29:36 50/: 4 Keeptan 2006 r. TTI: OKta6pe 2006 r.	
🖞 Пуск 🚿 🖓 💮 » 🖹 2 Проводник 🔹 🕅 2 Містозоft 🔹 🕎 Paint Shop Pro 🛛 🌾 Конфитурато 🦿 1С:Предприя 🕼 CorelDRAW 10 🔟 🖓 🗳 « 🗄 🗄 🖡 1	23

Рис. 7.21. Список значений в процедуре печати

### 7.8. Общие реквизиты

Некоторые реквизиты могут быть одинаковыми для всех документов в конфигурации. Если в "1С:Предприятие" работает несколько человек одновременно, важно знать автора созданного документа. Следовательно, во всех документах должен быть реквизит **Автор**.

В документы бывает необходимо добавлять пометки произвольного вида. Для них можно создать реквизит **Комментарии**. Таких реквизитов может быть много, это зависит от специфики конфигурации. Чтобы сократить однотипную работу, можно использовать общие реквизиты (рис. 7.22).

В нашей конфигурации общий реквизит — Комментарий. Общий реквизит не появляется в списке реквизитов каждого документа, но он будет доступен при редактировании формы документа — в диалоговом окне Размещение реквизитов Документа (рис. 7.23), к нему можно обращаться с помощью встроенного языка из модуля документа.



Рис. 7.22. Общие реквизиты в дереве метаданных

Общий реквизит

	Λ
15 Kaukaran yang kuna ang unay parang 16 man Banangar Bertumanya Tapan	
🖂 Файл Действия диалог Вставить Конфигурация Конструкторы Администрирова	иль Серана: Окона Полощь
1 2 2 3 3 4 4 6 6 6 6 7 2 2 4 1 1 2 4 1 1 2 4 1 1 2 4 1 1 1 1	4 8
	24 24 24 24 24
поступление товаров	
Поступление товаров № НомерДох. от ДагаДок	
Контралент: Контралент	Разнещение реконзитов Документа
Основание ПечатьАвто Подбор/	Реконзиты
N INU Колео Цена Сумиа Ст НДС	ИномерДок
	– Komparent
	■ Ссное онне Коленонтарий
	🗏 Вставлять имя Вограмяты. Поле веда 💌
	Реканзиты табличной части
ОК Закрыть Печать Конментарий Конментарий	
7	■ EKonsecteo
	©Cyerva
	СтаркаНДС
	Разместить в диалоге автоматически     Ризползять адариальноски
	re personal e multiplea defeu
	Вставите Отмене Помощь
TEl Buaseer / El Monues Al Tañasua Al Desamánno /	/
□ 레큐르   � ♣   H 王   H 王 M   ■   \$ ₽    &   @ R ●	T T I I I I I I I I I I I I I I I I I I
Т. Конфигурация Документ ПоступлениеТо ФФорма-Документ.ПоступлениеТо	Tau
Для получения подоказки нажините F1	МЛМ 137×14 49, 19 СТАТУС: 5Д., Документ
🕂 Пуск 🍥 🔄 💮 * 🗋 2 Проводняк 🔹 📝 2 Microsoft 🔹 🚮 Paint Shop Pro	«Сконфигурат 😤 1С:Предприя 🖾 CoreDRAW 10 🔟 🕐 🤻 🔠 К 17:25

Рис. 7.23. Общий реквизит в диалоговом окне размещения реквизитов формы

### 7.9. Журналы документов

Журналы являются средством для работы с документами. Журнал — это список всех зарегистрированных в системе документов, которые можно помечать на удаление, проводить и отменять проведение, проверять проводки документа и т. п. Прямо в журнале можно создавать новый документ.

В нашей конфигурации уже существуют несколько журналов, созданных одновременно с документами. Они находятся в ветке дерева метаданных **Журналы документов** (рис. 7.24).



Рис. 7.24. Ветвь метаданных Журналы документов

В журнале Учет товаров (рис. 7.25) видны графы Дата, Время, Документ (вид документа), Номер и Сумма. Если нам нужны дополнительные сведения, то необходимо создать графы журнала. Дважды щелкнув мышью по ветви УчетТоваров, перейдем в окно редактирования формы журнала. Вызовем диалог для создания новых граф, щелкнув кнопку Новый. В появившемся

диалоговом окне информация о возможных значениях граф представлена в виде стандартного дерева Windows. Последовательно раскрывая уровни, выбираем необходимое значение (Контрагент), которое при щелчке по стрелке попадает в поле Выбранные значения (рис. 7.26). Нажимаем кнопку ОК. Теперь в окне редактирования журнала перейдем в форму списка (кнопки Формы | ФормаСписка), вставим реквизиты в форму списка и получим усовершенствованный вид журнала (рис. 7.27).

Журналы документов могут быть разных видов и играть разную роль в конфигурации. Углубленно изучить журналы документов и работу с ними можно с помощью документации к программе.

🗐 1С:П	редпри	тие -	Бухис	опер учет, р	асчет																_ 8 ×
Файл	Действия	ОП	ерации	Справочник	и Докуме	енты Жу	рналы	Резуль	ьтаты р	асчета	Отч	еты	Обраб	отки Се	рвис	Окна	Помощь				
m 🛛	¥ ∎	χ ⊑	b 62	661	i b c	#			[	- A	2	8		11 P	l to	ММ	(+ M-				
			_				_	_	_				111				_		_	_	_
	lin.	Учет	товаро	ов (01.01.06	5-03.12.00	5)	_	_	_	_			-1								
	2	1 22	a e	■ ® F	។ដូ 🕞	¤ <sub>K</sub> ⊡⊦	- 「 ?	ן און													
						1								_							
		Д	ата	Время	Докуме	ент	Ho	мер	Сумма												
	-	<u>/1 25</u> /1 27	09.06 30.00	12:00:00	Поступл	ение това	po	1	1	,020.00											
		1 29	.09.06	15:29:36	Реализа	ция товар	OE	1		250.00											
					-			_													
					1			-													
								-			-										
		_	_		_	_	_	_	_	_	_	_	_								
al cale o		(1)		_																	
∬∭)Уче	т товар	ов (О	1.01.0																	-	
для пол	учения по	дсказ	жи нажи	ите F1		o. 11			in the			, 	VUM  T/	4: 29.09.0	15:2	9:36	БИ: 	4 квартал 2006	r.	TTI: OKTA	орь 2006 г.
ду Пус	K 🥥 🕻	3 (0)	) " [	ј 2 Проводни	к <b>т</b> [W]	2 Microso	nt •	] <u>島</u> 首 Pa	aint Shop	o Pro	1	конфи	гурато	····   %	1С:Про	едпри.	. 🔛	CoreIDRAW 10	RU	2) ÷ «	<u>29 31</u> 🎽 18:40



Пока же нужно знать, что в системе "1С:Предприятие" возможно создание шести видов журналов документов.

Обычный журнал — основное средство для работы с документами. В учебной конфигурации журналы приходных и расходных документов — обычные. Такой журнал позволяет работать только с теми *документами, которым он назначен* в конфигурации. В нашем примере каждый журнал работает либо с приходными, либо с расходными документами.

Общий журнал позволяет работать со *всеми документами*. Кроме того, он позволяет выполнять отбор документов по значению какого-нибудь реквизита. Подробнее об организации такого отбора можно прочитать в документации. В нашем примере это журнал с идентификатором **Общий**.



Рис. 7.26. Создание графы журнала документов

Дополнительный журнал похож на обычный журнал. Но если обычный журнал можно создать в процессе создания документа с помощью конструктора, то выбор документов, доступных в дополнительном журнале, выполняется при создании и редактировании самого дополнительного журнала. Например, у нас есть журналы приходных и расходных документов, а можно создать дополнительный журнал Торговые документы, в котором будут видны оба вида документов.

Журнал подчиненных документов предназначен для работы с документами, подчиненными выбранному. В нашей конфигурации таких документов нет.

Полный журнал позволяет работать со всеми видами документов, существующими в конфигурации.



Рис. 7.27. Вид журнала Учет товаров

Журнал Прочие предназначен для работы с документами, для которых в процессе создания конфигурации не указаны конкретные журналы. Журнал Прочие всегда создается автоматически и имеет только экранную форму, которая не может настраиваться.

Таким образом, независимо от того, назначали ли в процессе разработки конфигурации документу журнал или нет, документ все равно будет фиксироваться в одном из журналов и никогда не потеряется.

### 7.10. Другие документы учебной конфигурации

Следующий документ нашей учебной конфигурации — "РеализацияТоваров". По составу реквизитов этот документ совпадает с "ПоступлениеТоваров" и вставить его в ветвь дерева метаданных Документы. Для выполнения этих операций откройте окно метаданных в конфигураторе. Выделите нужный объект метаданных — документ "ПоступлениеТоваров". Скопируйте его в буфер обмена с помощью меню Действия | Запомнить либо с помощью соответствующих кнопок на панели инструментов. Затем вставьте документ с помощью меню Действия | Вставить либо кнопок на панели инструментов. Новый документ будет назван "ПоступлениеТоваров1". Откройте окно редактирования этого документа и задайте новый идентификатор: РеализацияТоваров". Журнал для хранения документа — Учет товаров.

Откройте форму документа "Реализация товаров" и соответствующим образом исправьте название документа. Для этого щелкните правой кнопкой мыши по надписи Поступление товаров №, в контекстном меню выберите пункт Свойства и в палитре свойств в поле заголовка измените слово Поступление на Реализация.

Благодаря вводу в работу нового документа в учете хозяйственной деятельности нашего предприятия произошли изменения. В частности, при оформлении отпуска товара документом "Реализация товаров" цену товара необходимо указывать отпускную. В нашем справочнике "Номенклатура" отпускная цена хранится в реквизите Цена. Поэтому в модуле формы документа "Реализация товаров" нужно сделать изменения. Процедуру устанценыПрих() переименуем в устанЦеныРасх(), вместо реквизита справочника номенклатуры Себестоимость используем реквизит Цена.Процедура. Процедура устанЦеныРасх() будет выглядеть, как в листинге 7.17.

#### Листинг 7.17

Процедура УстанЦеныРасх()

Цена=Товар.Цена.Получить (ДатаДок); //Цену товара из спр. Номенклатура-//в документ

СтавкаНДС=Товар.СтавкаНДС.Получить (ДатаДок);//Ставку НДС из спр. //Номенклатура- в документ Соответственно, в процедуре ОбработкаПодбора () исправьте вызов процедуры: теперь она вызывается как УстанЦеныРасх () вместо скопированного из приходной накладной вызова УстанЦеныПрих ().

Закройте форму и окно редактирования документа, запустите "1С:Предприятие" в пользовательском режиме и проверьте как работет документ.

Документ "Амортизация" предназначен для начисления амортизации основных средств предприятия. На настоящем этапе разработки конфигурации укажем, что этот документ почти не имеет реквизитов за исключением установленных по умолчанию. Не имеет табличной части. "Рабочая" часть этого документа — модуль проведения. С программированием в модуле проведения мы познакомимся в *главе 10* Поэтому достаточно в диалоговом окне размещения реквизитов документа выбрать все реквизиты (номер, дату документа и общий реквизит — комментарий) и разместить их в форме документа, как показано на рис 7.28.



Рис. 7.28. Форма документа "Амортизация"

Документ "Поступление материалов" (Идентификатор — "ПоступлениеМатериалов") аналогичен документу "Поступление товаров". В табличной части документов реквизит ТМЦ имеет тип значения "Справочник. Материалы". Процедуры и функции для заполнения документа методом подбора также должны быть изменены в соответствии с отличиями в справочниках. Например, в процедуре Подбор необходимо указать справочник "Материалы". Процедуры ЗапросКоличестваВПодборе (Выб, Конт) Экспорт и Выч\_суммы\_накл(Конт) Экспорт остаются без изменений. Для того мы и создали эти универсальные процедуры в глобальном модуле, чтобы облегчить себе работу в дальнейшем.

Документ "Поступление ОС (основных средств)" (идентификатор — "ПоступлениеОС") несколько отличается от документов поступления товаров и материалов. В табличной части документов реквизит **ТМЦ** имеет тип значения "Справочник. ОсновныеСредства". Кроме того, добавлен специфический реквизит **Инвентарный номер**. В процедуре подбор необходимо указать справочник "Основные средства". НДС и, соответственно, ставка НДС не всегда нужны при учете ОС. Поэтому в справочнике "ОсновныеСредства" эти реквизиты отсутствуют. В процедуре УстанЦеныПрих() модуля формы документа строка для вычисления НДС отсутствует. При необходимости НДС вычисляется прямо в документе, как указано в *разд. 7.1.* Для этого в свойства поля ввода **СтавкаНДС** в поле **Формула** вставляется формула:

#### Листинг 7.18

НДС=Сумма/(СтавкаНДС+100)\*СтавкаНДС

В глобальный модуль также необходимо внести изменения. Каждому основному средству присваивается свой собственный инвентарный номер, поэтому количество основного средства всегда равно единице. Количество нужно будет нам только на этапе проведения документа, а пока этот реквизит отсутствует. Поэтому в процедуре ЗапросКоличестваВПодборе (Выб, Конт) Экспорт мы должны проверить, из какого документа вызывается эта процедура. Если документ — "ПоступлениеОС", тогда мы не должны работать с количеством ТМЦ.

Для выяснения вида документа, вызывающего процедуру ЗапросКоличестваВПодборе (Выб, Конт) Экспорт, используем метод Вид (), который возвращает строку с названием вида документа. Теперь листинг процедуры должен выглядеть так (листинг 7.19).

#### Листинг 7.19

```
Процедура ЗапросКоличестваВПодборе (Выб, Конт) Экспорт // интерактивно
                                                   //выставляем количество
  Если Выб.ПометкаУдаления()=1 Тогда //товар помечен на удаление
    Предупреждение ("Нельзя выбирать помеченные на удаление товары!");
    Возврат; //прекращаем работу процедуры
  КонецЕсли;
  Кол=1;//Объявили переменную для ввода количества и присвоили ей
        //значение "1"
  Если Конт.Вид()="ПоступлениеОС" Тогда
                                          //Поступление ОС не требует
                                          //количества
    Конт. НоваяСтрока (); // добавляем строку в табличную часть
    Конт.ТМЦ=Выб; //Товар из Выб в добавленную строку
  Иначе
              //а другие документы требуют
    Если ВвестиЧисло (Кол, "Введите количество", 10, 2)=1 Тогда
      Конт.НоваяСтрока(); // добавляем строку в табличную часть
      Конт.ТМЦ=Выб; //Товар из Выб в добавленную строку
      Конт.Количество=Кол; //Количество- туда же
    КонецЕсли;
  КонецЕсли;
КонецПроцедуры
В процедуре Выч суммы накл (Конт) Экспорт также не нужно использовать
количество и вычислять НДС. Здесь используем ту же проверку вида документа:
Листинг
Процедура Выч суммы накл(Конт) Экспорт
  Если Конт.Вид()="ПоступлениеОС" Тогда //Поступление ОС количество=1
    Конт.Сумма=Конт.Цена;
  Иначе
    Конт.Сумма=Конт.Цена*Конт.Количество;
  Конт.НДС=Конт.Сумма/(Конт.СтавкаНДС+100)*Конт.СтавкаНДС;
  КонецЕсли;
КонецПроцедуры
```

Обратите внимание на использование глобального параметра конт, с которым мы познакомились в *разд.* 7.3. Этот параметр передает в глобальный модуль не только значения реквизитов документа, но и вид самого документа.

Документ "Списание материалов" (идентификатор — "СписаниеМатериалов") относится к журналу Учет материалов. С помощью этого документа осуществляется хозяйственная операция по списанию материалов. В шапке документа используются реквизиты Счет затрат и Вид затрат. Реквизит Счет затрат имеет пока тип значения "Строка", длина 10 символов. После создания плана счетов тип значения этого реквизита нужно будет исправить. Реквизит Вид затрат имеет тип значения "Перечисление.Затраты" (см. разд. 6.2). Реквизиты табличной части:

**ТМЦ** — Справочник. Материалы;

**С Количество** — Число, длина 10, точность 2.

При списании материалов нам не нужны ставка НДС и сумма НДС, поэтому в табличной части остаются только реквизиты **Цена** и **Сумма**. Эти реквизиты аналогичны реквизитам документа "Поступление материалов".

В процедуре УстанЦеныПрих () модуля формы документа "Списание материалов" отсутствует строка для вычисления ставки НДС. В остальном процедура аналогична одноименной процедуре в модуле формы документа "Поступление материалов".

В глобальном модуле в процедуру Выч\_суммы\_накл (Конт) Экспорт нужно внести дополнения. Для документа "Списание материалов" нет необходимости вычислять НДС. Поэтому должен проверяться вид документа, и если документ — "Списание материалов", вычисляться должна только сумма. Однако в процедуре уже осуществляется проверка на один вид документа — "Поступление ОС". Проверку на следующий вид документа можно осуществить, используя оператор ИначеЕсли (листинг 7.20).

Листинг 7.20	
	0
Процедура выч суммы накл(Конт)	ЭКСПОРТ

```
Если Конт.Вид()="ПоступлениеОС" Тогда //Поступление ОС количество=1
Конт.Сумма=Конт.Цена;
ИначеЕсли
Конт.Вид()="СписаниеМатериалов" Тогда //не нужен расчет НДС
Конт.Сумма=Конт.Цена*Конт.Количество;
Иначе
Конт.Сумма=Конт.Цена*Конт.Количество;
Конт.НДС=Конт.Сумма/(Конт.СтавкаНДС+100)*Конт.СтавкаНДС;
КонецЕсли;
КонецПроцедуры
```

Обратите внимание: сначала проверяется вид одного определенного документа с помощью оператора Если, затем второго с помощью оператора ИначеЕсли. В случае, когда оба условия не выполнены, выполняется условие оператора Иначе.

Документ "Списание ОС" (идентификатор "СписаниеОС") относится к журналу Учет ОС. С помощью этого документа осуществляется хозяйственная операция по списанию основных средств. В шапке документа используются реквизиты Счет затрат и Вид затрат. Реквизит Счет затрат имеет пока тип значения "Строка", длина 10. После создания плана счетов тип значения этого реквизита исправим. Реквизит Вид затрат имеет тип значения "Перечисление.Затраты" (см. разд. 6.4). Реквизиты табличной части:

- □ ТМЦ Справочник. ОсновныеСредства;
- □ Сумма и Износ тип значения обоих реквизитов число, длина 15, точность 2.

Процедуры УстанЦеныПрих () и Подбор () такая же, как в модуле формы документа "Поступление ОС".

Количество ОС у нас всегда равно единице, поэтому процедура ЗапросколичестваВПодборе (Выб, Конт) Экспорт в глобальном модуле должна отрабатывать так же, как и для документа "Поступление ОС". В данном случае можно использовать логический оператор или. То есть условие должно быть усложнено и эта часть процедуры выполняется в случае, если вид документа "Поступление ОС" или "Списание ОС":

#### Листинг 7.21

Процедура ЗапросКоличестваВПодборе(Выб, Конт) Экспорт // интерактивно //выставляем количество
Если Выб.ПометкаУдаления()=1 Тогда //товар помечен на удаление
Предупреждение("Нельзя выбирать помеченные на удаление товары!");
Возврат; //прекращаем работу процедуры
КонецЕсли;
Кол=1;//Объявили переменную для ввода количества и присвоили ей //значение "1"
Если (Конт.Вид()="ПоступлениеОС") или (Конт.Вид()="СписаниеОС") Тогда

//Поступление ОС не требует количества

Конт. НоваяСтрока (); // добавляем строку в табличную часть

Конт.ТМЦ=Выб; //Товар из Выб в добавленную строку

#### Документы

```
Иначе
            //а другие документы требуют
    Если ВвестиЧисло (Кол, "Введите количество", 10, 2)=1 Тогда
      Конт. НоваяСтрока (); // добавляем строку в табличную часть
      Конт.ТМЦ=Выб; //Товар из Выб в добавленную строку
      Конт.Количество=Кол;//Количество- туда же
    КонецЕсли;
  КонецЕсли;
КонецПроцедуры
Процедура Выч суммы накл (Конт) Экспорт
  Если (Конт.Вид()="ПоступлениеОС") или (Конт.Вид()="СписаниеОС") Тогда
//Поступление ОС количество=1
    Конт.Сумма=Конт.Цена;
  ИначеЕсли
    Конт.Вид()="СписаниеМатериалов" Тогда //не нужен расчет НДС
    Конт.Сумма=Конт.Цена*Конт.Количество;
  Иначе
    Конт.Сумма=Конт.Цена*Конт.Количество;
  Конт. НДС=Конт. Сумма/(Конт. СтавкаНДС+100) *Конт. СтавкаНДС;
  КонецЕсли;
КонецПроцедуры
```

В процедуре Выч\_суммы\_накл (Конт) Экспорт также использован логический оператор или.

Теперь создадим общий журнал документов. В дереве объектов метаданных щелкнем правой кнопкой мыши по ветви **Журналы документов** и в контекстном меню выберем пункт **Конструктор Журнала документов**. Идентификатор и синоним — "Общий", комментарий "Общий журнал документов". В следующем диалоговом окне выбираем вид журнала **Общий**. В последнем диалоговом окне нажмите кнопку **Готово** и откроется окно редактирования общего журнала. Щелкните по кнопке **Формы**, выберите форму списка и заполните ее аналогично формы списка ранее изученных объектов метаданных (рис. 7.29).

Теперь сохраните конфигурацию. Запустив "1С:Предприятие" в пользовательском режиме, можно пользоваться общим журналом, например, для отбора документов (рис. 7.30).

		_					_					
се Re	энфигуратор Эайл Действ	- <b>Бужио</b> г 1я Диалог	ер учет, расчет - Вставить Конфи	Форма-Ж турация К	ирнал.Общик онструкторы	и.Форма.Форма Администриро	спискај вание Сег	овис Окна	Помощь			_ 레 ×
			68 8 5	¢ ḿ[			2 9					
				ar	rda sila rika		00 JO JO	 				
	194.40.9	ംലം		<u>⊢  </u> ] ar	er er er		<u>an 200 2</u> .		2			
530	Эбщий журна	л	-	_	_							
	Дата	Время	Документ	Номер	Коммента	арий 🔺						
				-								
				+	+							
				-								
3						<b>_</b>						
							1					
	Диалог	/ 🗂 Mo	адуль 🗼 🖄 Т	аблица								
	⇒] <u>∓∓</u> <u>±±</u>	*)[* *	$  = \Xi   = \Xi$	፼ #	۱ 🖉 📗	× • × •		🖺 I ab	🔟 📴 🗉	E9		
च∕ <sup>≣</sup> Ko	онфигурация '		🗐 Журнал Общи	í	Форма	а-Журнал.Общ	ий					
Для п	олучения под	казки наж	мите F1							NUM 304 × 134	8, 22	СТАТУС: БД , Документ *
â9∎;	уск 🥥 🗋	• » C	2 Проводник	- ₩2 M	crosoft Office.	🕶 🎆 Paint Sh	nop Pro	候 Кон	фигуратор	CorelDRAW 10 -	[E RU	? ኛ « <u>1 31</u> 🔏 18:45

Рис. 7.29. Форма списка общего журнала

<b>S</b> 1	С:Предприя	гие - Бух и	опер учет, расч	ет													_ 8 ×
Фай	п Действия	Операции	Справочники Д	цокументы Ж	Курналы Ре	зультаты расче	ета О	тчеты	Обработки	Сервис	Окна	Помощь					
		X 🖻 🛱	681	5 ¢ #		•	ดัล	8		92   W	М	M+ M-					
lít l	Общий журн	ал (01.01.0	06-03.12.06)			_ 0	×										
1.2			의 사 왕이 코 성	7 8. R B	- RF - HR	21 \?											
_		- <b></b>	1979 at 197	10 HA 1 HR 11	82. 1014) [[1112		_										
lг	Дата	Время	Документ	Номер	Комме	гарий 🔺	1										
15	01.06.06	12:00:00	Начало месяца	Î	1												
	08.06.06	12:00:00	Приказ о прием	е на г	1												
	30.06.06	12:00:00	Приказ о прием	е на р	2												
	30.06.06	12:00:10	Выплата зарабо	тной	2												
	<u>01.07.06</u>	12:00:00	Начало месяца		2												
	14.07.06	12:00:00	Выплата зарабо	тной	1												
	01.08.06	12:00:00	Поступление ОС		1												
	07.08.06	19:45:37	Поступление ма	териа	1												
	1 31.08.06	12:00:00	Амортизация	_	3												
	01.09.06	12:00:00	Начало месяца	_	3												
	13.09.06	12:00:00	Списание матер	юналог	1												
	25.09.06	12:00:00	Поступление тов	варов	2												
	27.03.06	12:00:00	Поступление тов	варов	2	•											
1							- 11										
			_														
1. Trill C	бщий журна	эл (01.01															
Для і	олучения под	сказки нажи	мите F1						NUM TA: 29.	09.06 15:	:29:36	БИ:	4 квартал 2006 г		тп: Ок	гябрь 2006	r.
đy	Іуск 🧭 🗋	ю » <u>с</u>	2 Проводник 👻	2 Micros	soft 🔹	Paint Shop Pro	e e	🛱 Конф	игурато	🖑 1С:П	редпр	и	CorelDRAW 10	RU	9 🕄	« <u>39</u> 31 )	18:45





### Запросы и отчеты

Автоматизация учета подразумевает не только ввод, хранение информации, но и получение ее с соответствующей обработкой и в удобном для просмотра виде. Обработанная информация в системе "1С:Предприятие" может быть получена в виде объекта метаданных Отчет. В определенных случаях выборка и обработка данных для отчета может производиться с помощью методов выборки соответствующих обрабатываемых объектов метаданных (например, справочников или документов). С подобными способами работы мы знакомились в *разд. 6.3.* Однако часто нужную информацию невозможно получить, работая непосредственно с документами, справочниками и т. п. Кроме того, во многих случаях использование подобных средств не эффективно с точки зрения производительности системы. Чтобы упростить построение отчетов, существует специальный агрегатный тип данных — "Запрос".

### Определение

Запрос — это агрегатный объект, предназначенный для выборки данных из справочников, документов, регистров, журналов расчетов, планов счетов, бухгалтерских проводок и операций, который предоставляет возможность группировки данных по различным критериям и накопления числовых данных с помощью функций.

Результатом работы запроса должна стать таблица, наполненная данными и расположенная на локальном компьютере пользователя.

### 8.1. Начало создания отчета

Объект метаданных Отчет является отдельной ветвью дерева метаданных и создается аналогично другим объектам метаданных с помощью конструктора. В качестве примера будет создан отчет о закупках, который обраба-

тывает необходимые нам сведения, содержащиеся в приходных документах. Идентификатор — ОтчетПоПрихДок, синоним — Отчет по приходным документам.

При создании самой первой диалоговой формы отчета пролистаем все остальные шаги конструктора (кнопкой **Далее**) и нажмем кнопку **Готово**. В результате получим основу диалоговой формы отчета (рис. 8.1).



Рис. 8.1. Основа диалоговой формы запроса

### 8.2. Запрос

Теперь в отчет необходимо вывести некоторые интересующие нас данные. Выборка данных из различных объектов метаданных в интересных для пользователя разрезах производится несколькими способами. Мы рассмотрим один из наиболее мощных методов — запрос.

Для написания запросов нужно выполнить следующие действия.

- 1. Объявить переменные запроса, как ссылки на отдельные атрибуты или реквизиты объектов.
- 2. Описать функции, которые будут рассчитывать (суммировать, вычислять среднее и т. п.) значения на основании данных числовых переменных запроса.
- 3. Описать группировки запроса по каким переменным запроса необходимо сгруппировать данные (выполнить функции запроса).
- 4. Описать условия какими значениями переменных нужно ограничить формирование результатов запроса.

Создайте запрос в нашей учебной конфигурации. Пусть это будет запрос по документу "ПоступлениеТоваров". Цель запроса — получить информацию о том, сколько и каких товаров поставили нам наши поставщики.

Конструктор Запросов : Пер	
Период будет выбираться в	
Период 🔽 🖉 Выражен ВыбНачПер	
По С Выражен ВыбКонПер	
ОбрабатыватьДокументы	
Обрабатывать	
Без итогов	
]	

Рис. 8.2. Конструктор запроса

В форме отчета перейдем в модуль. Выберем пункты меню Конструкторы | Запрос. В диалоговом окне зададим имя запроса. Можно оставить и имя по

умолчанию — Сформировать. Нажимаем кнопку **ОК** — раскрывается диалоговое окно конструктора запроса (рис. 8.2). Управление данным конструктором также осуществляется с помощью кнопок Далее, Назад, Отмена.

Отметим в окне конструктора, что период выборки будет выбираться в диалоге формы нашего отчета. В поле **ОбрабатыватьДокументы** выберем пункт **Все**, поскольку у нас еще нет конкретных процедур проведения документов. В поле **Обрабатывать** выберем **НеПомеченныеНаУдаление**. Документ, который собираются удалять, не должен участвовать в формировании отчета.

Нажимаем кнопку Далее и в следующем окне добавляем переменные в запрос. Для этого в правой части окна в дереве метаданных раскроем ветвь Документы, затем развернем ветвь документа "ПоступлениеТоваров", чтобы видеть реквизиты документа. Нажмем кнопку Добавить переменную, затем щелкнем мышью по реквизиту Контрагент и перенесем его в соседнее окно слева с помощью кнопки со стрелкой (рис. 8.3). Добавим переменные для выборки товара, количества и суммы. Нажмем кнопку Далее.

🚰 Конфигуратор - Бух и опер учет, расчет – [Форма-Отчет.новый1]	_ 8 ×
💾 Файл Действия Текст Конфигурация Конструкторы Администрирование Сервис Окна Помощь	_ 8 ×
122 ↓ 10 5 4 10 5 4 1 5 1 5	
在右边不在右 大台的盖袖袖 聪明 ③ ③●字 字代字 習	
//жжжжжжжжжжжжжжжжжжжжжжжжжжжжжжжжжжж	
КокецПроцедуры	
Конструктор Запросов : Переменные	
Имя переменной: Контрагент =	
Lossystem:       I Occrypnesse       Cocrypnesse       Cocrynnesse       Cocrypnesse       Cocrypnesse	
Добавить переменную Удалить переменную	
	Þ
\⊑Диалог ) ⊟ Модуль / Matafinua /	
	INVITED IN
	15:19

Рис. 8.3. Процедура добавления переменных в запрос

С помощью диалогового окна добавим функции для расчета количества товара и суммы, на которую поставлен товар (рис. 8.4).

Сконфигуратор - Бухи опер учет, расчет - [Форма-Отчет.Отчет.ПоПрихДок]	
// Процедура генерации запроса Сформироваль.	
Процедура СформироВаль() Конструктор Запросов : Функции	
Функция КоличествоСулима =	
Сумена 🔽 ( Количество 🔽 )	
Г когда :	
чулкция количествоцита = сулика(количество); Функция СуликаСулика = Сулика(Сулика);	
Î	
1	
Defense territory 1	
Тоозвиге фінктию адачице фінктию	
<Назад Далее> Отмена Помощь	
ТаБ Бивесли0Сехиция("ТМП"):	
КонецЦикла; // Заполнение полей Контрагент	
ТаБ. ВывеслиСекцию ("Конпразени");	▼ ►
СЭДиалог 🔪 🖹 Модуль 🖉 Таблица 💫 🖄 Сформировать 🦯	
▲ ≫ ≫ ≫ 幸 宇 〃 ※ 宇 <b>□ 말 □ 歳</b>	
жа Конфигурация * ДФорма-Отчет.ОтчетПо ДФорма-Отчет.новый1	
Для получения подсказки нажинте F1 I, 1 NUM СТАТУС: 5/	1*, Документ *
🖅 Пуск 😸 🔄 🐑 🕐 " 🗔 2 Проводник 🔹 💯 2 Місrosoft О 🛛 🧱 Paint Shop Pro 🔤 🌾 Конфигурато 🖓 CorelDRAW 10   🔃 🖞 🐇 🦷 🤄	15:20 📄 🖳 کې

Рис. 8.4. Процедура добавления функций в запрос

Перейдем в следующее окно и добавим группировки. Переменную **Контрагент** упорядочим по реквизиту справочника "Контрагенты" — "Наименование" — **Контрагент. Наименование** (т. е. поставщики будут выводиться в алфавитном порядке), а переменную **ТМЦ** упорядочим по реквизиту справочника "ТМЦ" — "Наименование" — **ТМЦ. Наименование**. Ведь эти переменные — прямые ссылки на соответствующие справочники. Посмотрите типы значений реквизитов **Контрагент** и **ТМЦ** документа "ПоступлениеТоваров", для выборки данных из которого делается отчет и запрос соответственно.

В следующем окне добавим еще одно условие. Для этого нажмем кнопку **Добавить условие**. Поставим флажки Элементарное условие и Выбирать в диалоге. Теперь выберем условие Контрагент в ВыбКонтрагент.

#### Замечание

Условие "в" не равнозначно знаку "=". Если вместо условия "в" поставить условие "=", то, в нашем случае, если пользователь выберет группу поставщиков, отчет будет пустой. Ведь товар не приобретается у группы поставщиков. Если пользователь вообще не выбрал поставщика, отчет также будет пустой. В данном случае считается, что запрос должен найти "пустого" поставщика. Если же использовать условие "в", то запрос будет работать по всем поставщикам.

Переходим в последнее окно конструктора. В нем уже выставлены все необходимые флажки. В верхнем окне — созданный конструктором текст запроса. Осталось нажать кнопку **Готово** — и отчет закончен. Его диалоговая форма изображена на рис. 8.5.

🖟 Конфигуратор - Бух и опер учет, расчет - [Форма-Отчет.ОтчетПоПрихДок]
🖞 Фэйл действия Диалог Вставить Конфигурация Конструкторы Адиинистрирование Сереис Окна Помощь 📃 🖻
□ ば 県 ※ 時 局 毎 年 時 ■ ■ ■ ば 所 済 8
植态学术长柏林 有右伤植伤病 聪慧 \$1 ① ● ● \$2 果代家 智
<sup>\$</sup>
K. Clerver Trillsuffe
Период с: ВыбНаиПериод по: ВыбКонПериод
Контрагент: ВыбКонтрагент
Сфорнировать Закрыть
СЭДналог / 🗍 Модуль / АЗТаблица / АСформировать /
E 41 77 22 4
ка Конфигурация * Дформа-Отчет.ВтчетПо Дформа-Отчет.новый1
ля получения подоказки нажинте F1 [10, 20 СТАТУС: БД *, Докунент
参 Dyck 🥘 🔄 💮 🎽 🗋 2 Dposodress 🔹 📝 2 Microsoft C 🔹 📆 Paint Shop Pro 🔤 🦿 🥵 Kondyaryparo 🔮 CorelDRAW 10 🗰 😨 🖓 🦿 🐺 Karalyaryparo

Рис. 8.5. Диалоговая форма отчета

Теперь отчет необходимо отредактировать вручную. Подправить диалоговую форму, если требуется — программу и шаблон печатной формы. Если мы начнем работы с отчетом в пользовательском режиме, то в диалоговой форме сразу обнаружатся неудобства. Можно выбрать различных поставщиков,

но нельзя очистить поле **Контрагенты**. То есть мы либо работаем по всему списку поставщиков, либо по каждому в отдельности. Для очистки поля **Контрагенты** добавим кнопку с названием "**X**". В палитре свойств кнопки в поле **Формула** напишем: **ВыбКонтрагент** = **0**. Теперь при нажатии этой кнопки поле **Контрагент** будет очищаться, и можно осуществлять различные выборки по поставщикам. Теперь отредактируем шаблон печатной формы. Сейчас он выглядит так (рис. 8.6): итоговые данные по каждому поставщику располагаются выше строк с товаром. Это непривычная для пользователя форма. Лучше разместить подитоги по поставщикам после перечисления поставленных ими товаров. Для этого перейдем в модуль отчета и исправим процедуру Сформировать. Найдем в ней фрагмент программы (листинг 8.1).



Рис. 8.6. Экранная и печатная формы отчета

#### Листинг 8.1

Пока Запрос.Группировка(1) = 1 Цикл

// Заполнение полей Контрагент
```
Таб.ВывестиСекцию("Контрагент");
Пока Запрос.Группировка(2) = 1 Цикл
// Заполнение полей Товар
Таб.ВывестиСекцию("ТМЦ");
КонецЦикла;
// Заполнение полей Поставщик
Таб.ВывестиСекцию("Поставщик");
КонецЦикла;
```

И выведем секцию с подитогами **Контрагент** после цикла с выводом в таблицу товаров данного поставщика. В результате фрагмент будет таким (листинг 8.2).

🚰 1 С:Предприятие - Бух и опер у	учет, расчет			_ 8 ×
Файл Действия Таблица Вид О	Операции Справочники Документы Журналы Рез	ультаты расчета. Отчеты. Обра	ботки Сервис Окна Помощь	
1) <b> 2   </b>	A ≥ ¢ A ≥ ¢ B	*************	M M+ M-	
🖪 Отчет По Приходным	_ D ×			
6. 63. 21 X				
01.01.02 - 16.03.06	а Сформировать *			_ 🗆 🗙
				<b>_</b>
Период с: 01.01.02 🔲 п	Контрагент Наименование	ТМЦ Наименование	КоличествоСумма Су	MMaCymma 400.00
	Поставщик	Панты	1.00	200.00
Контрагент:	Поставшик	TO AT DA	2.00	300.00
,	Пример	Кабель	1.00	100.00
	Пример		1.00	100.00
	Итого		3.00	400.00
<u>Сфорнировать</u> Закрыть				
	2 [ ] P = F = 16 F			
🗈 Отчет По Приходным 📃 🖄 🕻	Сформировать *			
Для получения подсказки нажмите F1	1	NUM TA: 16.03.06 19:2	28:16 БИ: 4 квартал 2006 г.	ТП: Март 2005 г.
🕂 Пуск 🍏 🔄 🕞 🔌 🗀 2 Про	оводник 🔹 😿 2 Microsoft Office 📲 🚟 Paint Sl	nop Pro 🛛 🖉 3 1C:V7 starte	er 🖾 CorelDRAW 10 - [E	RU ? 🗘 « T 🗄 K 15:35

#### Листинг 8.2

```
Пока Запрос.Группировка(1) = 1 Цикл
Пока Запрос.Группировка(2) = 1 Цикл
// Заполнение полей Товар
Таб.ВывестиСекцию("ТМЦ");
КонецЦикла;
// Заполнение полей Контрагент
Таб.ВывестиСекцию("Контрагент");
```

КонецЦикла;

И печатная форма отчета будет выглядеть более привычно (рис. 8.7).

Разумеется, экранную и печатную форму еще надо приводить в порядок — выравнивать, делать заголовки и т. п. Но это можно выполнить самостоятельно.

В листинге 8.3 приведена процедура сформировать с подробными комментариями для объяснения механизма ее работы.

# Листинг 8.3 // Процедура генерации запроса Сформировать. 11 Процедура Сформировать () Перем Запрос, ТекстЗапроса, Таб; //Создание объекта типа Запрос Запрос = СоздатьОбъект ("Запрос"); ТекстЗапроса = "//{{ЗАПРОС (Сформировать) с ВыбНачПериода по ВыбКонПериода; |Контрагент = Документ.ПоступлениеТоваров.Контрагент; |ТМЦ = Документ.ПоступлениеТоваров.ТМЦ; |Количество = Документ.ПоступлениеТоваров.Количество; |Сумма = Документ.ПоступлениеТоваров.Сумма; |Функция КоличествоСумма = Сумма (Количество); |Функция СуммаСумма = Сумма(Сумма);

```
Группировка Контрагент упорядочить по Контрагент. Наименование;
   Группировка ТМЦ упорядочить по ТМЦ. Наименование;
   Условие (Контрагент в ВыбКонтрагент);
   |"//}}ЗАПРОС
   // Если ошибка в запросе, то выход из процедуры
   Если Запрос.Выполнить (ТекстЗапроса) = 0 Тогда
      Возврат;
   КонецЕсли;
   // Подготовка к заполнению выходных форм данными запроса
  Таб = СоздатьОбъект ("Таблица");
   Таб.ИсходнаяТаблица ("Сформировать");
   // Заполнение полей "Заголовок"
  Таб.ВывестиСекцию("Заголовок");
   Состояние ("Заполнение выходной таблицы...");
   Таб.Опции(0, 0, Таб.ВысотаТаблицы(), 0);
   Пока Запрос. Группировка (1) = 1 Цикл
      Пока Запрос. Группировка (2) = 1 Цикл
         // Заполнение полей ТМЦ
         Таб.ВывестиСекцию("ТМЦ");
      КонецЦикла;
      // Заполнение полей Контрагент
      Таб. Вывести Секцию ("Контрагент");
   КонецЦикла;
   // Заполнение полей "Итого"
  Таб. Вывести Секцию ("Итого");
   // Вывод заполненной формы
  Таб.ТолькоПросмотр(1);
   Таб.Показать ("Сформировать", "");
КонецПроцедуры
```

Конструктор запроса удобен, но не универсален. Для сложных запросов с помощью конструктора можно создать основу и вручную дописывать необходимый код. Запрос можно редактировать с помощью конструктора.

#### Дополним код отчета еще двумя строками:

#### Листинг 8.4

```
ВыбНачПериода= Константа.ОсновнаяДатаНачалаОтчета;
ВыбКонПериода= ПолучитьДатуТА();
```

А конфигурацию дополним, соответственно, новой константой — Основная дата начала отчета. Теперь в пользовательском режиме нужно установить дату — значение этой константы, — например, на начало года. Отчет будет автоматически устанавливать период с заданной в константе даты по дату точки актуальности итогов.



Рис. 8.8. Переменная для запроса

### Определение

Точка актуальности итогов — понятие только оперативного учета. Показывает момент последнего изменения в регистрах оперативного учета. Например, момент последнего проведения документа.

### Предупреждение

При редактировании запроса с помощью конструктора изменения в тексте процедуры и в шаблоне печатной формы, внесенные вручную, могут быть потеряны.

Рассмотрим еще один вид отчета, построенный на запросе по справочнику. Создадим отчет по основным средствам, где укажем наименование ОС, группу ОС, норму амортизации и начальную стоимость ОС.

Для создания отчета используем конструктор отчета, как делали это ранее. Вызовем конструктор запросов. На его втором листе вместо документов для создания переменных используем справочник "Основные средства" (рис. 8.8).

В переменную будет помещаться текущий элемент справочника. Функции здесь не нужны, а группировать основные средства будем по инвентарному номеру — коду. В результате получится очень простой запрос с шаблоном печатной формы (рис. 8.9, листинг 8.5).

### Листинг 8.5

```
// Если ошибка в запросе, то выход из процедуры
       Если Запрос.Выполнить (ТекстЗапроса) = 0 Тогда
              Возврат;
       КонецЕсли;
       // Подготовка к заполнению выходных форм данными запроса
       Таб = СоздатьОбъект ("Таблица");
       Таб.ИсходнаяТаблица ("Сформировать");
       // Заполнение полей "Заголовок"
       Таб. Вывести Секцию ("Заголовок");
       Состояние ("Заполнение выходной таблицы...");
       Таб.Опции(0, 0, Таб.ВысотаТаблицы(), 0);
       Пока Запрос. Группировка (1) = 1 Цикл
              // Заполнение полей ТекущийЭлемент
              Таб. ВывестиСекцию ("ТекущийЭлемент");
       КонецПикла:
       // Вывод заполненной формы
       Таб.ТолькоПросмотр(1);
       Таб.Показать ("Сформировать", "");
КонецПроцедуры
```

Если сейчас запустить этот отчет, он выведет на печать только коды OC. Для того чтобы получить интересующие нас данные, нет необходимости дописывать код процедуры. Можно внести необходимые команды на встроенном языке прямо в шаблон печатной формы отчета.

В колонке Код ОС запишем Запрос. Текущий Элемент. Код.

В колонке Наименование ОС запишем Запрос. Текущий Элемент. Наименование.

В колонке Группа ОС запишем Запрос. Текущий Элемент. ГруппаОС.

В колонке Начальная стоимость ОС запишем Запрос. Текущий Элемент. Начальная Стоимость. Получить (ВыбКонПериода). Ведь начальная стоимость у нас — периодический реквизит. Это те же конструкции, что разбирались в *разд. 6.3.* 

В результате доработки таблицы получаем отчет по основным средствам (рис. 8.10).

🚰 Конфиг	урато	) - Б	ух и опер учет, расч	ет - [Форма-Отчет.(	ЭтчетПоОс (Отчет по с	основным средства	м)]						- 8 ×
🖽 Файл	Действ	ия	Таблица Вид Конфи	нгурация Конструкто	ры Администрирование	Сервис Окна Пом	ющь						_ @ ×
) 1 1 1 1		6	1666	Э С 🙀 Виль До	. <b>.</b>	ĥ ?							
	<b>R</b> %	4	~≦ ⊛ ⊝ 🛛 🎗	ee 🛛 🔺 🖻 1	8 <b>2 2 3 2 3</b>	a 答 答 答 答 X	ŝ.						
R12C4			→ -µ =										
		1	2	3	4	5	6	7	8	9	10	11	· · ·
	1	_											
заголовок	2												
	3	1	Текущий Элемент Код										
ТекущийЭл	4	<	Запрос.ЗначениеУпор										_
	6												
	7	-											
	8												
	9												
	10												_
	11	_											_
	12	_											
	14	-											
	15												
	16												
	17												
	18												
	19	_											_
	20	_											
	21	-											
	23												-
4													
С Диал	)r	$\overline{\lambda}$	Модуль 💦	🖄 Таблица 💦	Сформировать	🔪 🔊 Таблица 1							
	=	\$2	100   k   🔤   C	] & EFC	5 12 FF								
≅ <sup>∕≣</sup> Конфиг;	/рация	*	Форма-С	Этчет.ОтчетПо									
Для получе	ния под	сказ	ки нажмите F1							N	UM CTATYC	БД , Докуг	ент *
<i>1</i> Пуск	e) 🖸	$\odot$	» 📄 2 Проводник	- W 2 Microsof	t Office 🞆 Paint Sho	op Pro	C:V7 starter	• 🖾 co	orelDRAW 10	- [E RL	000	« 11 BI K	15:54

Рис. 8.9. Шаблон печатной формы отчета по справочнику "Основные средства"

💯 1 C:	Тредприят	ие - Бух и	опер	учет, рас	чет											_ 8 ×
Файл	Действия	Таблица	Вид	Операции	Справочни	и Документы	Журналы	Результат	ы расчета	Отчеты	Обработк	и Сервис	Окна Г	Томощь		
12	200	b R	8	& I	5 ¢ A		•	66	8 1	i 11 92	0 м	M+ M-				
Σ1 01	чет по ОС						- 15	l ×I	11				1			
٥F	Сформир	оватьОС	*										_	- 🗆 ×		
Ē														<u> </u>		
3	ТекүщийЭ	темент Код	а. П Те	ест			100	)		Пример	Поступле	Hune OC 1				
2			Te	ест2			200	)		Поставщик	Поступле	ние ОС 2				
							30	00.0000								
r i																
1 -																
			ß	E .	] 🖁 🗉	F 🗆 🍾	IF									
Σ. Οτ	ет по ОС		A	Сформир	оватьОС *											
Для по.	тучения под	сказки наж	кмите Г	F1					NUM	TA: 16.03.0	6 19:28:16	5 БИ:	4 кварта	л 2006 г.	ТП: Март	2005 r.
₫∕Пу	:к 🥥 🗋	• •	<u>2</u> 1	іроводник	- 10	Microsoft Office	• 📰 Pa	int Shop Pro		;; 3 1C:¥7	starter	🔹 🎯 Core	IDRAW 10	) - [E	u ?, 🖁 «	<u>6</u> 31 K 15:56

Рис. 8.10. Отчет по справочнику "Основные средства"



# Таблица значений

В *разд.* 7.7 мы рассматривали специальный агрегатный тип данных — список значений. В этой главе рассмотрим похожий тип данных — таблицу значений. Как и список значений, таблица значений — это средство языка. Она не сохраняется на жестком диске компьютера, а существует только в оперативной памяти во время исполнения соответствующей процедуры. В отличие от списка значений, таблица значений является двухмерным динамическим массивом, в котором можно редактировать, добавлять, удалять и сортировать элементы. Таблица значений может выглядеть для пользователя, как обычная табличная часть объекта метаданных. Типы хранимых значений могут быть разными: "Число", "Строка", "Справочник" и т. д. Этот тип данных применяется, как правило, в тех случаях, когда неудобно или невозможно использовать механизмы выборки из справочников, журналов, документов, перечислений.

Таблица значений может быть видимой для пользователя в экранных формах объектов метаданных, когда требуется интерактивное воздействие на данные таблицы, а может использоваться программистом только для решения жестко определенных задач — в этом случае пользователь и не подозревает о ее существовании. В этой главе мы познакомимся со вторым вариантом использования таблицы значений.

В качестве примера используем отчет по основным средствам (см. главу 8). Сделаем отчет более информативным, добавив сведения о поставщике основного средства. Эти сведения отсутствуют в справочнике "Основные средства", и извлечь их мы можем только из документа "Поступление ОС" (см. разд. 7.10). Здесь нам и пригодится таблица значений.

Данные о поставщике-контрагенте мы можем получить с помощью запроса по документу "Поступление ОС" (листинг 9.1).

#### Листинг 9.1

```
Перем Запрос, ТекстЗапроса, Таб;
   //Создание объекта типа Запрос
   Запрос = СоздатьОбъект ("Запрос");
  ТекстЗапроса =
   "//{{ЗАПРОС (ПервСтоимость)
Период с ВыбНачПериода по ВыбКонПериода;
|ТекущийДокумент = Документ.ПоступлениеОС.ТекущийДокумент;
|Контрагент = Документ.ПоступлениеОС.Контрагент;
|ОС = Документ.ПоступлениеОС.ТМЦ;
|Группировка ТекущийДокумент;
   |"//}}ЗАПРОС
   ;
   // Если ошибка в запросе, то выход из процедуры
   Если Запрос.Выполнить (ТекстЗапроса) = 0 Тогда
      Возврат;
   КонецЕсли;
```

На следующем этапе нам нужно в виртуальную таблицу значений собрать данные, выбранные запросом по справочнику "Основные средства" и выбранные запросом по документу "Поступление ОС". Полученную виртуальную таблицу (динамический массив) выведем на печать.

Виртуальный объект "Таблица значений" создается аналогично другим объектам в "1С:Предприятии" — с помощью команды Создать Объект:

```
ТаблОС=СоздатьОбъект ("ТаблицаЗначений");
```

Теперь определимся, из каких колонок таблица будет состоять. В таблице будут те же колонки, что и в уже созданном отчете по справочнику "Основные средства":

- Инвентарный номер;
- Наименование;
- 🗆 Дата ввода;
- Первоначальная стоимость

и дополнительные колонки, чтобы можно было легко определить документ прихода основного средства:

# □ Поставщик;

## 🛛 Документ.

Будет и отдельная вспомогательная колонка: **Флаг группы**. Она нужна для того, чтобы в дальнейшем при выводе на печать полученного отчета можно было разделить строки с наименованиями групп справочника (не путать с группами OC!) и строки с наименованием основного средства.

Колонки в таблице значений создаются с помощью оператора НоваяКолонка (листинг 9.2):

Листинг 9.2

```
ТаблОС=СоздатьОбъект ("ТаблицаЗначений");
ТаблОС.НоваяКолонка ("ИнвНомер");
ТаблОС.НоваяКолонка ("Иаименование");
ТаблОС.НоваяКолонка ("Группа");
ТаблОС.НоваяКолонка ("ДатаВвода");
ТаблОС.НоваяКолонка ("Стоимость");
ТаблОС.НоваяКолонка ("Поставщик");
ТаблОС.НоваяКолонка ("ФлагГруппы"); //группа справочника ВД
ТаблОС.НоваяКолонка ("Докум");
```

Это самый простой вариант использования этого оператора. Полный синтаксис таков: НоваяКолонка (<Идентификатор>, <Тип>, <Длина>, <Точность>, <Заголовок>, <Ширина>, <Формат>, <Положение>). Мы использовали только <Идентификатор>.

- □ <тип> необязательный параметр. Строка или вид субконто, задающий тип колонки. Если этот параметр не указан, то можно хранить любой тип.
- С <Длина> необязательный параметр. Длина для числовой или строковой колонки.
- <Точность> необязательный параметр. Точность (длина дробной части) для числовой колонки.
- Саголовок> необязательный параметр. Строковое выражение, содержащее заголовок колонки в элементе диалога типа "ТаблицаЗначений".

- «Ширина» необязательный параметр. Числовое выражение, содержащее ширину колонки (в символах) для представления колонки в элементе диалога типа "ТаблицаЗначений".
- <формат> необязательный параметр. Строковое выражение, содержащее форматную строку, которая будет использована при визуальном отображении значений данной колонки.
- Сположение> необязательный параметр. Определяет вариант выравнивания при визуальном отображении значений данной колонки. Число: 1 слева; 2 — справа.

#### Замечание

Чтобы виртуальный массив был доступен всем процедурам модуля, необходимо создать переменную, являющуюся ссылкой на таблицу значений, и объявить ее в начале модуля.

В нашем примере мы объявим переменную таблос в начале модуля, затем модернизируем процедуру сформировать () следующим образом (листинг 9.3).

#### Листинг 9.3

```
// Процедура генерации запроса Сформировать.
11
Процедура Сформировать1()
   Перем Запрос, ТекстЗапроса, Таб;
   ТаблОС=СоздатьОбъект ("ТаблицаЗначений");
   ТаблОС. НоваяКолонка ("ИнвНомер");
   ТаблОС. НоваяКолонка ("Наименование");
   ТаблОС. НоваяКолонка ("Группа");
   ТаблОС. НоваяКолонка ("ДатаВвода");
   ТаблОС. НоваяКолонка ("Стоимость");
   ТаблОС. НоваяКолонка ("Поставщик");
   ТаблОС.НоваяКолонка ("ФлагГруппы"); //группа справочника ВД
   ТаблОС.НоваяКолонка ("Докум");
   //Создание объекта типа Запрос
   Запрос = СоздатьОбъект ("Запрос");
   ТекстЗапроса =
   "//{{ЗАПРОС(Сформировать)
   Период с ВыбНачПериода по ВыбКонПериода;
```

```
|ТекущийЭлемент = Справочник.ОсновныеСредства.ТекущийЭлемент;
Группировка ТекущийЭлемент упорядочить по ТекущийЭлемент.Код;
|"//}}ЗАПРОС
// Если ошибка в запросе, то выход из процедуры
Если Запрос.Выполнить (ТекстЗапроса) = 0 Тогда
   Возврат;
КонецЕсли;
// Подготовка к заполнению выходных форм данными запроса
Таб = СоздатьОбъект ("Таблица");
Таб.ИсходнаяТаблица ("Сформировать");
// Заполнение полей "Заголовок"
Таб. ВывестиСекцию ("Заголовок");
Состояние ("Заполнение выходной таблицы...");
Таб.Опции(0, 0, Таб.ВысотаТаблицы(), 0);
Пока Запрос. Группировка (1) = 1 Цикл
   // Заполнение полей ТекушийЭлемент
   ТаблОС.НоваяСтрока();
                           //добавить строку в таблицу значений.
   Если Запрос.ТекущийЭлемент.ЭтоГруппа()=1 Тогда //Если название
                                  //группы элементов справочника, тогда
      ТаблОС.ФлагГруппы=1;
                               //поставим флаг группы и
      ТаблОС.ИнвНомер="";
      ТаблОС.Наименование=Запрос.ТекушийЭлемент; //запишем название
                                                  //группы
      ТаблОС.Группа="";
      ТаблОС.ДатаВвода="";
      ТаблОС.Стоимость=0;
      ТаблОС.Поставщик="";
   Иначе
                               // а если это элемент, тогда
      ТаблОС.ФлагГруппы=0; //флаг группы снимем, а остальные колонки
                             //заполним
      ТаблОС.ИнвНомер=Запрос.ЗначениеУпорядочивания (1,1);
      ТаблОС.Наименование=Запрос.ТекущийЭлемент;
      ТаблОС.Группа=Запрос.ТекущийЭлемент.ГруппаОС;
      ТаблОС.ДатаВвода=""; // не заполнив дату ввода и
```

ТаблОС.Стоимость=Запрос.ТекущийЭлемент.НачальнаяСтоимость.Получить (ВыбКон Периода); ТаблОС.Поставщик=""; //поставщика, так как нам пока их неоткуда //взять! КонецЕсли; КонецЦикла; // Вывод заполненной формы ПервСтоимость(); Печать(); ТаблОС=0; КонецПроцедуры

Посмотрите, после выполнения запроса вместо печатной формы заполняется созданная таблица значений. Для этого используется оператор НоваяСтрока (листинг 9.4).

#### Листинг 9.4

ТаблОС.НоваяСтрока(); //добавить строку в таблицу значений.

После этого мы определяем, является ли полученный из выборки элемент справочника названием группы справочника (листинг 9.5).

Листинг 9.5

```
Если Запрос.ТекущийЭлемент.ЭтоГруппа()=1 Тогда //Если название группы
//элементов справочника, тогда
ТаблОС.ФлагГруппы=1; //поставим флаг группы и
```

Если это группа, то стоимость этого элемента равна нулю, из всех остальных колонок заполняется только наименование. Если же полученный элемент — непосредственно основное средство, тогда флаг группы ставится равным нулю, заполняются все колонки, кроме даты ввода данных об основном средстве и поставщике основного средства, так как нам неоткуда взять эти данные. Эти недостающие сведения мы как раз и получим из процедуры ПервСтоимость () (см. листинг 9.1). Только теперь мы дополним программу специальными командами для работы с таблицей значений. Целиком процедура будет выглядеть так — листинг 9.6.

#### Листинг 9.6

```
// Процедура генерации запроса ПервСтоимость.
11
Процедура ПервСтоимость ()
   Перем НомСтр, НомКол;
  Перем Запрос, ТекстЗапроса, Таб;
   //Создание объекта типа Запрос
   Запрос = СоздатьОбъект ("Запрос");
  ТекстЗапроса =
   "//{{ЗАПРОС (ПервСтоимость)
Период с ВыбНачПериода по ВыбКонПериода;
|ТекущийДокумент = Документ.ПоступлениеОС.ТекущийДокумент;
|Контрагент = Документ.ПоступлениеОС.Контрагент;
ОС = Документ.ПоступлениеОС.ТМЦ;
|Группировка ТекущийДокумент;
   |"//}}ЗАПРОС
   ;
   // Если ошибка в запросе, то выход из процедуры
   Если Запрос.Выполнить (ТекстЗапроса) = 0 Тогда
      Возврат;
   КонецЕсли;
   Пока Запрос. Группировка (1) = 1 Цикл
      Если ТаблОС.НайтиЗначение (Запрос.ОС, НомСтр, 2)=1 Тогда
         ТаблОС.УстановитьЗначение (НомСтр, 6, Запрос.Контрагент);
         ТаблОС. Установить Значение (НомСтр, 8, Запрос. Текущий Документ);
        Состояние ("Установлен поставщик "+" "+Запрос.Контрагент+"
"+НомСтр);
   HOMCTP="";
      КонецЕсли;
   КонецЦикла;
КонецПроцедуры
```

При обработке данных запроса в цикле используется уже частично заполненная таблица значений. Нужно найти в таблице значений основное средство, соответствующее выбранному в запросе, и вставить данные о поставщике и документе, которым было оформлено поступление этого основного средства. Для этого мы используем оператор НайтиЗначение. В параметрах этого оператора сначала указывается искомое значение (в данном случае Запросос), затем идентификатор переменной, куда возвращается номер найденной строки.

### Замечание

Если при вызове метода передать в этот параметр номер строки, то поиск будет осуществляться только по указанной строке.

В начале процедуры объявлена переменная номстр для номера найденной строки. Последний параметр — идентификатор переменной, куда возвращается номер найденной колонки.

### Замечание

Если при вызове метода передать в этот параметр номер или идентификатор колонки, то поиск будет осуществляться только по указанной колонке.

Указана колонка под номером 2, т. е. поиск ведется по наименованиям основных средств.

Как только значение найдено, используется оператор установитьЗначение. Первый параметр оператора — номер строки, в которую нужно установить требуемое значение (в данном примере — номстр). Второй параметр — номер или идентификатор колонки (колонка № 6, куда помещается на-именование поставщика). Третий параметр — устанавливаемое значение (в данном примере — Запрос.Контрагент). Точно так же устанавливаются в соответствующую строку наименование и номер документа прихода основного средства.

Используя системную функцию Состояние(), выводим в строку состояния сведения о происходящем в информационной базе. Символы, заключенные в кавычки, будут выводиться в строке состояния так же, как они прописаны внутри кавычек. Выражения вне кавычек вычисляются по мере необходимости. Обратите внимание на символ " " (пробел, заключенный в кавычки). Он служит для разделения строки.

Затем переменной номСтр присваиваем пустое значение. Если этого не сделать, то поиск будет осуществляться только по номеру строки, указанному в переменной, и, соответственно, данные будут ошибочны.

Для печати отчета теперь нужно создать отдельную процедуру, выводящую в шаблон печатной формы данные из таблицы значений. Процедура Печать () (листинг 9.7) вызывается из процедуры Сформировать ().

#### Листинг 9.7

```
// Процедура печать
11
Процедура Печать ()
  Перем Таб;
   // Подготовка к заполнению выходных форм данными запроса
  Таб = СоздатьОбъект ("Таблица");
  Таб.ИсходнаяТаблица("Таблица");
   // Заполнение полей "Заголовок"
  Таб. ВывестиСекцию ("Заголовок");
  Состояние ("Заполнение выходной таблицы...");
  Таб.Опции(0, 0, Таб.ВысотаТаблицы(), 0);
  ТаблОС.ВыбратьСтроки();
   Пока ТаблОС.ПолучитьСтроку()=1 Цикл
      Если ТаблОС.ФлагГруппы=1 Тогда
         Таб. ВывестиСекцию ("ГруппаТМЦ");
         Продолжить;
      Иначе
         Таб. Вывести Секцию ("Текущий Элемент");
      КонецЕсли;
   КонецЦикла;
  Таб.ВывестиСекцию("Итог");
  // Вывод заполненной формы
  Таб.ТолькоПросмотр(1);
   Таб.Показать ("СформироватьОС", "");
КонецПроцедуры
```

### В этой процедуре обратите внимание на следующие строки (листинг 9.8):

#### Листинг 9.8

```
ТаблОС.ВыбратьСтроки();
Пока ТаблОС.ПолучитьСтроку()=1 Цикл
```

Здесь используются новые операторы для таблицы значений Метод Выбрать Строки () предоставляет возможность перебирать строки таблицы значений (открывает выборку). Дальнейшая выборка осуществляется с помощью метода Получить Строку (). Этот метод возвращает 1 — если строка выбрана успешно, 0 — если строка не выбрана (выборка закончилась). Поэтому цикл длится, пока метод Получить Строку () возвращает 1.

В этой процедуре применяется колонка таблицы значений **ФлагГруппы**. Если значение в колонке равно 1, выводится секция ГруппаТМЦ, в противном случае — секция ТекущийЭлемент.

<u> 7</u> Конфиг	урато	р - Бух и опер учет, расче	ет - [Форма-Отчет.Отче	тПоОс (Отчет по осн	ювным средствам)]						
🕒 Файл	Действ	зия Таблица Вид Конфи	гурация Конструкторы	Администрирование	Сервис Окна Помощь						_ & ×
) 🖻 🖻		x 6 6 6 6 🗉	⊃ ¢ ₩	■ 6 8	8						
	$\mathbb{R}^{3}$	n   % ⊗ ⊗ ] \$   0	ee    * * *	28 X 20 X	治治学论						
R7C4		<u></u> -µ = Ta6	5лОС.Итог("Стоимость")#№	1019.4							
0		1	2	3	4	5	6	7	8	9	10 🔺
Jaronobok	2	ТекушийЭлемент Кол									
ТекущийЭл	3	<ТаблОс.ИнвНомер>	<ТаблОС.Наименование>	<ТаблОС.ДатаВвода>	<ТаблОС.Стоимость>	<ТаблОС.По	<ТаблОС.До	кум≻			
[pynnaTML]	5		<ТаблОС.Наименов	ание>							
Итог	6				-ToBoOC Maron	' Tourson		10.45			
	7					Стоимос	ID JHINU	19.4/			
	9										
	10										
	12										
	13										
	14										
	16										
	17										
	19										
	20										
	21										
	23										
	24										
	25										
	27										
	28										
	29										
	31										
4	32								1		
П Пиал		) 📰 Молиль 🕥	Allafonua /Al	Сформировать	7						
	E G	£. 101 \≽ ⊠ [		۲ <u>۵</u> FF	/						
च∕ब Конфиг;	/рация	Форма-О	тчет.ОтчетПо								
Для получе	ния под	асказки нажмите F1							NUM CT	атус: БД , До	кумент
<i>1</i> 9 Пуск	<i>i</i>	🕑 » 🗋 2 Проводник	▼ 2 Microsoft 0 ▼	Paint Shop Pro	💞 2 1C:¥7 start •	CorelDRAW	V 10 RU	27	< <u> 15</u> 28 29	Kariga	14:39

Рис. 9.1. Шаблон печатной формы отчета

Для вывода данных в шаблоне печатной формы применяются выражения: Таблос.ИнвНомер,..., Таблос.Докум, т. е. осуществляется обращение непосредственно к значениям в колонках таблицы значений (рис. 9.1). Обратите внимание на итоговую строку. В ней пример правильного форматирования итога из таблицы значений.

#### Замечание

Подробно о форматировании строк и чисел можно прочитать в документации к "1С:Предприятие".

Теперь пользователь увидит в отчете более полную информацию по основным средствам (рис. 9.2).

2221 Г.П.	редприятие - Бух и г	опер учет, расчет						
Файл Д	цействия Таблица I	Вид Операции Справочни	ки Документы Журн	алы Результаты расч	ета Отчеты (	Обработки Сервис	Окна Помощь	
) 10 c	* 🖬   X 🖻 🖻	68 N > ¢ A		• # % ?	■ ¥ 92	🜘 м м+ м-		
<b>Б</b> ] Отч	ет по ОС							
1	СформироватьОС *	k						
3	Ина номер	Отчет по Основ	ным средствам <sub>Дата в вода в</sub>	Стоимость	Поставщик	По документь		
п		001	эксплуатацию	10000	вваиави Г	Поступление ОС 1		
2		0C2		0				
		Ø 图 [ 1 ] ₽ ] 目						
Σ] Отче	T NO OC	СформироватьОС *						
Для полу	чения подсказки наж	мите F1		N	JM  TA: 29.09.06	15:29:36 БИ	: 4 квартал 2006 г.	TП: Октябрь 2006 г.
₫⊖Пусн		2 Проводник 👻 💯 2 М	crosoft O 👻 🚊 Pain	t Shop Pro 🛛 🖓 2 10	:¥7 start + 🧕	CorelDRAW 10	RU 2 - « 🖪	<u>引 33</u> 🌾 🖀 🥵 🗐 🗐 15:05

**Рис. 9.2.** Отчет по справочнику "Основные средства" с дополнительной информацией о поставщике и документе прихода



# Бухгалтерский учет

Основное назначение компоненты "Бухгалтерский учет" — отражение хозяйственных операций предприятия в бухгалтерском учете. Возможности компоненты, которые мы будем рассматривать на сквозном примере, позволяют организовать ведение учета с использованием нескольких планов счетов, многомерной аналитики, различных валют.

# 10.1. Краткое описание компоненты

Основным понятием компоненты "Бухгалтерский учет" является **Счет**. Каждый счет собирает информацию о конкретном виде имущества предприятия или обязательствах. Совокупность счетов называется *планом счетов*. В программе "1С:Бухгалтерский учет" план счетов является не только списком счетов, но и служит своеобразным фундаментом конфигурации, определяющим правила хранения учетной информации.

Компонента "Бухгалтерский учет" позволяет в одной конфигурации вести учет по нескольким планам счетов, при этом для каждого плана счетов может строиться произвольная иерархия субсчетов большой вложенности. Бухгалтерские итоги хранятся системой раздельно для каждого плана счетов. Планы счетов предназначены для ведения синтетического учета.

Аналитический учет можно вести для любого счета или субсчета и назначать до 5 различных разрезов аналитики. Объектами аналитического учета могут служить элементы справочников, документы системы "1С:Предприятие" или произвольные числа, даты и строки.

Для любого счета можно назначить ведение количественного учета.

Возможности компоненты позволяют также вести многовалютный учет. Количество одновременно используемых валют при этом не ограничивается.

Включение аналитического и валютного учета выполняется при редактировании планов счетов.

Для организации аналитического учета используется механизм субконто.

Субконто в системе "1С:Предприятие" называется объект аналитического учета.

Видом субконто, в свою очередь, называется множество однотипных объектов аналитического учета. Конфигуратор позволяет организовать любое количество видов субконто в соответствии с требованиями по ведению аналитического учета предприятия.

Также к основным понятиям компоненты относится понятие *операции*. Операцией считается любое хозяйственное действие, изменяющее состояния средств предприятия.

Для ввода информации о хозяйственных операциях в системе "1С:Предприятие" используется документ специального вида — "Операция".

Чтобы введенная операция вызвала изменение в учетной информации (бухгалтерских итогах), она должна содержать *проводки*. Для этого используется объект метаданных Проводка.

📲 Конфигурация *
御祭図   ↓ ↑ 紅   目
🖃 🖬 Задача 🔼
📲 Константы
🗂 🗂 Справочники
—́—́—́—́ Документы
🗄 🎁 Журналы документов
💮 Перечисления
🖳 📄 Обработки
🚊 🖓 Планы Счетов
— ⊒н⊐ Реквизиты
🔚 Виды Субконто
🖻 🙀 Операция 📃
🚯 Типовые Операции
🖷 Графы журнала
🚍 🖓 Проводка
Корректные Проводки
📑 Метаданные 🎯 Интерфейсы 😤 Права

**Рис. 10.1.** Конфигурация с объектами компоненты "Бухгалтерский учет"

Только при наличии установленной компоненты эти объекты будут доступны для работы.

На рис. 10.1 показаны основные виды метаданных, характеризующие компоненту "Бухгалтерский учет": Планы Счетов, Виды Субконто, Операция и Проводка.

# 10.2. План счетов

Как видно на рис. 10.1, планы счетов представляют собой такой же вид объектов метаданных, как и остальные объекты в дереве.

При установленной компоненте "Бухгалтерский учет" конфигурация может содержать практически неограниченное количество планов счетов. Исключение составляет версия **Стандарт**, имеющая ограничение на поддержку только одного плана счетов (см. главу 1).

Максимальная длина кода счета: 🛛 7 📑	Длина наименования счета: 🛛 🗧 🕂
Планы счетов	- Реквизиты счетов
Основной	ПолноеНаименование
Новый Изменить Удалить Количественный учет	- Новый Изменить Удалить
🔽 Только по аналитике	Справочник валют: << Не назначен >> 💌
Аналитический циет	Курс: << Не назначен >> 💌
Макс. количество субконто: 1 📑	Кратность: << Не назначен >> •
Редактировать счета: В списке	Разделитель учета: << Не назначен >> 💌

Рис. 10.2. Окно редактирования Планы Счетов

Для настройки свойств плана счетов необходимо сделать двойной щелчок левой кнопкой мыши на ветви дерева Планы счетов, тогда конфигуратор откроет окно редактирования.

Код счета можно представить строкой маски следующего вида: Код счета.Код субсчета.Код субсчета...

При задании значения свойства Максимальная длина кода счета следует исходить из максимального количества символов в коде счета и в кодах субсчетов, учитывая количество точек между ними.

Напомним, что отдельный объект аналитического учета называется субконто. А список однотипных субконто называется *видом субконто*. Система позволяет назначить максимальное количество видов субконто, которое возможно присоединить к счетам плана счетов. Это значение определяется в характеристике **Макс. количество субконто** окна редактирования **Планы счетов** (рис. 10.2) и может изменяться от 0 до 5 (в стандартной версии от 0 до 3).

При выборе максимального значения количества субконто следует руководствоваться требованиями пользователя или нормативными документами.

Не нужно устанавливать значение "с запасом", предполагая, что это когданибудь понадобится. Необходимо помнить, что система заранее готовится для хранения итогов в указанном количестве независимых разрезов — отводит память в информационной базе. Поэтому неоправданное увеличение количества субконто ведет к увеличению размера информационной базы, а это отражается на общей производительности системы.

Для настройки валютного учета необходимо, чтобы конфигурация содержала справочник, который хранил бы курсы валют. Как правило, это справочник "Валюты". В нашем примере валютный учет не ведется.

По каждому счету плана счетов при соответствующей настройке система может накапливать бухгалтерские итоги как по сумме, так и по количеству. Как правило, количественный учет может вестись как с учетом аналитики, так и без учета. Это свойство назначается в окне редактирования Планы счетов в разделе настроек Количественный учет установкой флага Только по аналитике. Если количественный учет ведется по субсчетам вместо учета по аналитическим счетам, тогда необходимо сбросить флажок. В этом случае признак количественного учета будет устанавливаться непосредственно у соответствующих субсчетов.

Каждый план счетов состоит из списка счетов, а каждый счет имеет обязательные реквизиты (атрибуты) — **Код** и **Наименование**. В этом смысле план счетов похож на справочник (рис. 10.3).

<u>1</u> де	нтификат	ор: Основной	<u>С</u> иноним:	Основ	зной			
≦ом	ментарий	: План счетов нетрад	иционный					
∐ać	лон кода							
	Код	Наименование		K	3	A	Субконто1	
Ŷ	БНС	Безналичные средства				A	1.00	
Yr.	ДОХ	Доходы				Π		
Ŷ	3TP	Затраты				A	Затраты	
竹	ИЗН	Износ ОС и НМА				Π	00	
۳ř.	НАЛ	Наличные деньги		2		A	-	
۳r.	0000	Основные средства				A	OC	
Ýř.	PAC	Взаиморасчеты				AΠ	Контрагенты	
ŤŤ.	PAC.1	Расчеты с поставщика	ми			П	Контрагенты	
Ŷ	PAC.2	Расчеты с покупателям	14			A	Контрагенты	
Ýľ	ТМЦ	9чет ТМЦ				AΠ	- 007	
Ý	ТМЦ.1	Товары		30		AΠ		
YP.	ТМЦ.2	Материалы		+		AΠ	Материалы	

Рис. 10.3. План счетов сквозного примера

Для счетов, как и для справочников, можно создавать практически неограниченное количество дополнительных реквизитов. Такие дополнительные реквизиты не оказывают непосредственного влияния на хранение бухгалтерских итогов, а позволяют дополнительно описать свойства счета, не связанные напрямую с ведением бухгалтерского учета (например, **Полное наименование**).

# 10.3. Операции, проводки

Как уже упоминалось, основным понятием бухгалтерского учета в системе "1С:Предприятие" является *операция*.

# 10.3.1. Операции

Документ специального вида "Операция" позволяет занести в систему информацию о хозяйственной операции и содержит обязательные реквизиты (атрибуты): дату и время совершения операции, сумму и содержание операции.

а опе	раци	к ДатаОпе	рации		Номер опера Сумма опера	ации: Номе ации: Сумм	рДок наОперации	
N#	Дт	Субконто	Кт	Субконто	Колич	Сумма	ПланСчетов	

Рис. 10.4. Настройка формы списка операции

Операция имеет представление только в виде табличной части, в которую могут быть выведены реквизиты (рис. 10.4).

Операция может быть автоматически сформирована на основе документа (объект метаданных Документ). При этом средствами встроенного языка описывается заполнение реквизитов операции информацией из документа, породившего эту операцию.

Операция может вводиться пользователем вручную. При этом пользователь сам заполняет все реквизиты операции.

# 10.3.2. Проводки

Ввод бухгалтерских проводок реализуется с помощью объекта метаданных Проводка. Особенность проводок в системе "1С:Предприятие" состоит в том, что проводки всегда вводятся в составе операции.

Проводка имеет представление только в форме списка, в который выводятся реквизиты, однозначно определяющие связь проводки с операцией и планом счетов.

Если операция формируется документом, то документ, помимо заполнения реквизитов операции, может автоматически сформировать необходимые бухгалтерские проводки операции.

<b></b> Фо	орма	а-Проводка	.Форма	Списка.	Рорм	аСписка								l
翧														
	N	Документ	Дата	Номер	Дт	Субконт	Кт	Субконт	Кол-во	Сумма	Содержание	Содержа	П	
		5								2	2			_
E									-	-				_
		3									2			-
F							-							-
			-			1	1				-			
													<u>••</u>	
•														
	Диа	лог /	( 🛄 Mo	дуль	)	🔊 Таблица								

Рис. 10.5. Настройка формы списка проводки

# 10.4. Субконто, виды субконто

Для работы с видами субконто в дереве метаданных существует специальная ветвь, которая называется Виды субконто.

Объект не имеет структуры хранения, а предназначен для описания соответствия конкретного типа данных конфигурации конкретному виду субконто, который, в свою очередь, может быть указан при настройке аналитического учета для конкретного счета.

Таким образом, настройка аналитического учета с использованием субконто распадается на два этапа:

- 1. Создание нового вида субконто и настройка его свойств.
- 2. Назначение субконто из имеющихся видов в окне **План счетов** для ведения аналитического учета и его последующая настройка.

Напомним, что у любого типизованного объекта метаданных есть общие свойства: Идентификатор, Синоним, Комментарий и Тип значения (рис. 10.6).

Вид субконто имеет и специфические свойства:

- □ Отбор при назначении этого признака пользователь сможет сделать отбор по значениям данного вида субконто в Журнале проводок;
- □ Цена и Валютная цена указывается реквизит справочника вида субконто, в котором может быть задано значение цены или валютной цены.

- □ Ввод пустых субконто при назначении этой характеристики можно не указывать конкретное значение субконто при вводе проводок;
- □ **Представление** служит для создания шаблона текстового представления субконто.



Рис. 10.6. Окно Свойства Вида Субконто (все вкладки)

После создания вида субконто и определения всех его характеристик мы можем назначить его соответствующему разрезу аналитического учета, что мы и рассмотрим на примере.

# 10.5. Примеры применения компоненты "Бухгалтерский учет"

Напомним, что описанные в данном разделе примеры показывают подходы к конфигурированию в компоненте "Бухгалтерский учет" и не носят рекомендательного характера для организации учета предприятий.

Для работы запустите учебную конфигурацию в режиме **Конфигуратор** и откройте окно **Конфигурация**.

# 10.5.1. Создание объектов

Для настройки бухгалтерского учета в учебной конфигурации нам необходимо будет дополнить ее новыми объектами, а также существующие объекты дополнить новыми реквизитами.

Для организации бухгалтерского учета нам понадобится дополнить основные константы, описанные в сквозном примере.

Мы создадим новую константу **Расчетный счет**, как описано в *главе 5*, тип значения — "Строка", длина 10 символов.

План счетов является основополагающим объектом компоненты, поэтому начнем настройку объектов именно с него. Для того чтобы настроить свойства плана счетов, необходимо сделать двойной щелчок левой кнопкой мыши на ветви дерева Планы счетов, тогда откроется окно редактирования **Планы Счетов** (см. рис. 10.2).

Заполним основные характеристики плана счетов, исходя из правил, описанных в *разд. 10.2*.

Настройка реквизитов **Максимальная** длина кода счета и Длина наименования счета является общей для всех планов счетов, которые могут быть в Конфигураторе (рис. 10.7).

			_ <b>_ _ _ ×</b>
7	÷	Длина <u>н</u> аименования счета: — Рекризиты сцетов	25 🛨
	7	7 🚦	7 📑 Длина <u>н</u> аименования счета: Реквизиты счетов

Рис. 10.7. Код и наименования плана счетов

В правом окне списка можем задать дополнительные реквизиты счета.

Нажатием кнопки **Новый** создадим дополнительный реквизит — **Полное** наименование счета (рис. 10.8).

Установим флажок ведения количественного учета по аналитике.

Для организации аналитического учета *(см. разд. 10.2)* будем использовать только одно субконто. В соответствии с этим зададим значение **Макс. количество субконто** равным 1.

По условиям сквозного примера мы не будем вести валютный учет, и, значит, этот параметр у нас не будет заполнен.

После назначения основных параметров Планов счетов создадим основной план счетов.

ойства Реквизи	ra	
Общие Дополни	тельные	· · ·
Идентификатор:	ПолноеНаименование	
Синоним:	Полное наименование	
Комментарий:		ок
Тип значения:	<<Строка>>	] Отмена
Длина:	100 🗮 🗖 Неогр. Почность: 🛛 🚆	Обновить

Рис. 10.8. Диалоговое окно Свойства Реквизита на вкладке Общие

Для того чтобы создать новый план счетов, необходимо в окне редактирования нажать кнопку **Новый**. После этого Конфигуратор сообщит, что в конфигурацию был добавлен специальный вид документа — "Операция".

Этот вид документа имеет специальное назначение. Он предназначен для ввода проводок в информационную базу вручную и для записи проводок обычных документов при проведении (рис. 10.9).



Рис. 10.9. Предупреждение о добавлении в конфигурацию Документа операций

Такое предупреждение выдается только при создании в Конфигураторе первого плана счетов.

После ответа на предупреждение Конфигуратор создаст новый план счетов и откроет палитру свойств этого объекта (рис. 10.10).

### Замечание

Знак # означает, что в соответствующем месте кода счетов может стоять символ (либо цифра, либо буква). Символ точки отделяет один уровень кода счета от другого.

두 План счетов	Новый				_	ı×
<u>И</u> дентификатор:	Новый	<u>С</u> иноним:	_			
<u>К</u> омментарий:						
Шаблон кода:	###.###					
Код	Наименование		Кол.	Заб.	Акт.	
<u>∀r</u>						
						-
						-
						-
						⁻▾

Рис. 10.10. Свойства объекта План счетов

В нашем примере значение шаблона кода означает, что в плане счетов счета могут иметь максимально до 2-х уровней вложенности. И коды счета, и субсчета могут быть величиной не более трех символов.

После настройки свойств **Плана счетов** и создания **Основного плана счетов** можно приступать к вводу конкретных счетов. Эта операция на первый взгляд похожа на ввод записей в справочник. Но есть серьезные различия!

Важной особенностью ввода счетов является то, что счет как запись может быть задан и в пользовательском режиме, и в Конфигураторе. Записи же справочника задаются только в пользовательском режиме "1С:Предприятие".

Когда счет задан в Конфигураторе, то в пользовательском режиме "1С:Предприятие" нельзя внести изменения в его настройку. Это свойство счета следует использовать, когда:

- в нормативных документах конфигурации требуется строго определенная настройка конкретного счета;
- документы используют строго определенные счета при генерации проводок в модуле документа.

Ввод счета в конфигурации не препятствует пользователю вводить в план счетов свои счета или субсчета в пользовательском режиме.

Для того чтобы задать счет в конфигурации, необходимо открыть окно Плана счетов следующим образом:

- 1. Выберите план счетов Основной в окне редактирования Планы Счетов (рис. 10.11).
- 2. Нажмите кнопку Изменить. При этом откроется окно плана счетов Основной.

👎 Планы Счетов	
Максимальная длина кода счета: 7 🚖 Планы счетов Основной ВСНОВНОЙ ВСПОВНОЙ ВСНОВНОЙ ВСПОВНОВНОВ ВСПОВНОВНОВНОВНОВНОВ ВСПОВНОВНОВНОВНОВНОВНОВНОВНОВНОВНОВНОВНОВНО	Длина наименования счета: 25 🚖 Реквизиты счетов ПолноеНаименование Ф. Новый Изменить Эделить
Кол 🗄 План счетов Основной	
I Идентификатор: Основной	Синоним: Основной
Ана. Комментарий: План счетов нетрадици	энный
Мак Шаблон кода: ###.###	
Ведак         Код         Наименование           Основ         У́ř	К З А Субконто1

Рис. 10.11. Окно плана счетов Основной (до ввода счетов)

### Примечание

В окне, представленном на рис. 10.11, можно вводить новые счета и изменять свойства существующих счетов, принадлежащих нашей конфигурации.

Все счета, необходимые нам для организации бухгалтерского учета, зададим в Конфигураторе.

План счетов может содержать счета, которые называются по аналогии со справочниками, *счетом-группой*. Счет-группа — это счет, который может иметь подчиненные счета — *субсчета*. При этом счет-группа не может быть указан в корреспонденциях проводок.

Так же, как и для справочников, для счетов существует понятие *уровня вложенности счета*. Уровень, как говорилось ранее в *разд. 10.2*, определяется шаблоном кода.

Введем в план счетов **Основной** счета, необходимые для решения поставленных в сквозном примере задач.

Откройте окно **План счетов Основной**, нажмите кнопку **Ins** на клавиатуре или, используя правую кнопку мыши, выберите пункт меню **Новая строка** для ввода нового счета в этот план счетов.

Задайте код РАС и наименование счета Взаиморасчеты.

На вопрос конфигуратора "Счет будет иметь субсчета?" следует ответить Да, поскольку у этого счета будут еще 2 субсчета — РАС.1 и РАС.2 для учета взаиморасчетов с поставщиками и покупателями.

В результате план счетов будет содержать в конфигурации счет-группу **РАС**, что обозначается желтой буквой **Т**, на которой стоит символ галочки (рис. 10.12).

두 План счетов Основной							_ 🗆 🗙	
Γ	Идентификатор: Основной Синоним:				ной			
	Комментарий: План счетов нетрадиционный							
	Шаблон кода:  ###.###							
		Код	Наименование	K	3	A	Субконто1 🔺	
	۲r.	НАЛ	Наличные деньги			A		
	ΎΓ	0CC	Основные средства			A	00	
	Ϋ́r.	PAC	Взаиморасчеты			AΠ	Контрагенты	
	٦¥	PAC.1	Расчеты с поставщиками			Π	Контрагенты	
	٦¥	PAC.2	Расчеты с покупателями			A	Контрагенты	
	Ϋ́r.	ТМЦ	Учет ТМЦ			AΠ		
							<b>•</b>	
L								

Рис. 10.12. Результат создания счета-группы

Дополнительные свойства счетов — валютный, количественный, забалансовый, активность — устанавливаем в соответствующих столбцах выбором из возможных значений. Значения дополнительных реквизитов должны соответствовать постановке задачи. В частности, в нашем примере не предусмотрена возможность ведения забалансового учета. Например, на рис. 10.12 видно, что в колонке плана счетов **Заб.** не указан признак ведения забалансового учета знаком "+".

Для настройки аналитического учета *(см. разд. 10.5)* необходимо сначала создать **Виды Субконто**, так как в нашей конфигурации они еще отсутствуют.

Для создания нового вида субконто **Контрагенты** воспользуемся специальным конструктором, имеющимся в системе "1С:Предприятие".

Запустим конструктор, используя команду меню Конструкторы, выбрав пункт Новый Вид Субконто....

В открывшемся диалоге заполним свойства нового вида субконто (рис. 10.13).

Конструктор Вида	а Субконто	×	
Идентификатор:	Контрагенты	Конструктор Вида Субконто	×
Синоним:	Контрагенты	Использовать существующий Справочник	
Комментарий:	Контрагенты	С Создать новый Справочник	
Тип значения: Справочник  Справочник Перечисление Документ Для создания Вида Субконто следует задать ему идентификатор и выбрать тип данных, который бы		Вид Справочника: Контрагенты	
предлагает осн организации ан	овные типы данных, используемы алитического учета). ь конструкторы для создания новь	Вы можете использовать существующий Справочник, выбрав его из списка, или создать новый Справочник, задав его идентификатор. В дальнейшем вы сможете настроить свойства созданного Вида Субконто и свойства Справочника.	
< Назад	Далее > Готово Отмен		1
		< Назад Далее > Готово Отмена Помощь	

Рис. 10.13. Настроенные свойства нового вида субконто

В нашем случае в качестве типа значения мы выбрали "Справочник", поскольку конкретный список контрагентов будет изменяться пользователями с течением времени.

Нажмем кнопку Далее и продолжим заполнение свойств субконто.

На этом этапе конструктор предлагает нам или выбрать уже существующий в конфигурации справочник, или создать новый. Мы выбираем существующий справочник "Контрагенты".

## Замечание

В качестве типа значений субконто могут выступать и базовые типы данных ("Число", "Строка", "Дата"). Для чисел и строк действуют некоторые ограничения на длину и точность. Подробнее об этом можно прочитать в документации к системе "1С:Предприятие 7.7".

Нажмите кнопку Готово для окончания процесса создания нового вида субконто.

В дереве метаданных появится новый вид субконто Контрагенты (рис. 10.14).

Заполним дополнительные свойства вида субконто (см. разд. 10.3) на вкладке Дополнительные.

<sub>па</sub> Конфигурация *				X
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				
Справочники Документы Мурналы документов Перечисления Отчеты Обработки Обработки Контрагенты Опрация Опрация Проводка Регистры Курналы расчетов Виды расчетов	Свойства Вида С Общие Дополн Цена: Валютная цена:	убконто ительные Настройки Представление << Не назначена >> << Не назначена >> Ввод пустых Субконто Быстрый выбор	Х	
🚯 Метаданные 🔐 Интер	фейсы 🧏 🕅 Права			

Рис. 10.14. Окно конфигурации и свойств вида субконто

В соответствии со значением настраиваемого вида субконто зададим и значения свойств на вкладке Настройки (рис. 10.15).

Перейдем на вкладку **Представление** и определим текстовое представление субконто (см. рис. 10.15).

Свойства Вида Субконто		×
Общие Дополнительные Наст	ройки Представление	
Значение: 🔄 🖳 🖡 🖡	Заголовок:	
Код		
Адрес	🔲 С новой строки	ок
		Отмена
		Обновить

Рис. 10.15. Диалоговое окно Свойства Вида Субконто, вкладка Представление

В соответствии с условиями сквозного примера для практики можете дополнить самостоятельно конфигурацию видами субконто: Материалы, OC, Затраты.

Для вида субконто Материалы устанавливаем характеристику Цена.

Для вида субконто Затраты устанавливаем тип значения "Перечисление".

Структура метаданных нашей конфигурации пополнились новыми объектами, и дерево метаданных выглядит, как на рис. 10.16.



Рис. 10.16. Окно Конфигурация с видами субконто

После создания необходимых для ведения учета объектов аналитического учета определим для счетов плана счетов **Основной** требуемые субконто. Для этого щелкнем дважды левой кнопкой мыши по колонке **Субконто1**. Откроется окно со списком субконто, существующих в конфигурации. Вставить требуемое субконто в план счетов можно либо двойным щелчком мыши по нужному субконто, либо одинарным щелчком выделить субконто и нажать кнопку **OK** (рис. 10.17).

После ввода всех необходимых нам счетов и настройки их свойств окно **План счетов Основной** принимает вид, показанный на рис. 10.18.

Следующий этап — создание операций и проводок. В объекте метаданных Операция в конфигурации задаются различные свойства операций, формы для ввода, редактирования и просмотра операций. В этом случае Операция очень похожа на объект Документ.

В отличие от объекта Документ в конфигурации может быть только один объект Операция.

Ŧ	Т План счетов Основной					на на Сибконто	
Идентификатор: Основной Синоним: Основной					Контрагенты		
k	Комментарий: План счетов нетрадиционный				Материалы ОС		
Ľ	<u> </u> абл	лон кода:	###.###				Затраты
		Код	Наименование	Кол.	Заб.	Акт.	d I
	ΎΓ	БНС	Безналичные средства			A	
	ΎΓ	ДОХ	Доходы			Π	
	Ϋ́,	3TP	Затраты			А	
	ΎĽ	ИЗН	Износ ОС и НМА			Π	
	Yr.	НАЛ	Наличные деньги			A	
	Yr.	0CC	Основные средства			A	У эчет по сумме
	Ϋ́r	PAC	Взаиморасчеты			AΠ	Учет по валютной сумме
	Yr.	PAC.1	Расчеты с поставщиками			Π	🛛 🔽 Учет по количеству
	۲r	PAC.2	Расчеты с покупателями			A	
	Ý	ТМЦ	Учет ТМЦ			AΠ	ОК Отмена Очистить
	ΎΓ	ТМЦ.1	Товары			AΠ	
	ΎĽ	ТМЦ.2	Материалы	+		AΠ	Материалы 🔜 💌 🖛

Рис. 10.17. План счетов Основной — назначение субконто

ች በታ	두 План счетов Основной 💶 🗖 🗙							
<u>И</u> де <u>К</u> ом	Идентификатор: Основной <u>С</u> иноним: Основной Комментарий: План счетов нетрадиционный							
Шаб	лон кода:	###.###						
	Код	Наименование	Кол.	Заб.	Акт.	Субконто1		
¥r	БНС	Безналичные средства			A			
¥ŕ	ДОХ	Доходы			Π			
l ĭr	3TP	Затраты			A	Затраты		
¥ŕ	ИЗН	Износ ОС и НМА			Π	00		
¥ŕ	НАЛ	Наличные деньги			А			
¥r	OCC	Основные средства			А	00		
Ý	PAC	Взаиморасчеты			AΠ	Контрагенты		
¥r	PAC.1	Расчеты с поставщиками			Π	Контрагенты		
- Yr	PAC.2	Расчеты с покупателями			A	Контрагенты		
- Yr	ТМЦ	Учет ТМЦ			AΠ			
- Yr	ТМЦ.1	Товары			AΠ			
- Yr	ТМЦ.2	Материалы	+		AΠ	Материалы		
	II		11	1	11			

Рис. 10.18. План счетов Основной

Объект Операция не является самостоятельным объектом, поскольку он всегда принадлежит и создается объектом Документ. При этом Документ может иметь только одну Операцию.
Для ввода операций вручную в метаданных существует специальный вид документа "Операция", создаваемый системой после создания плана счетов. В служебном виде документа "Операция" отсутствуют реквизиты, форма документа и модуль документа. При обращении к документу "Операция" используются все параметры объекта Операция.

Таким образом, документ "Операция" обладает всеми свойствами объекта Документ — отображается в журналах документов, участвует в отборах, имеет номер, дату, время и т. п. — и объекта Операции — форму, реквизиты.

Возможность иметь операцию для документов конкретного вида задается при настройке в свойствах документа параметра **Бухгалтерский учет**, определяющего принадлежность к одноименной компоненте и возможность создавать операцию (рис. 10.19).

🖹 Документ Новый1	
Идентификатор: Новый1	
<u>К</u> омментарий:	
Реквизиты шапки	
Новый Изменить Уда.	
Номер	
<u>Н</u> умератор: << Не назнач	
<ul> <li>Автоматическая нумерация</li> </ul>	
Разрешить проведение док	
Автоматическое удаление с	
Создавать операцию: Всегда	
<u>В</u> вод на основании	

Как видно на рисунке, возможны три режима создания операции документа:

- □ Всегда операция будет записываться в момент записи документа, даже при отсутствии проводок;
- □ Выборочно запись операции описывается в модуле формы (при записи) или в модуле документа;
- □ **Только при проведении** операция будет записана только при проведении в модуле документа средствами встроенного языка.

Как и документ, операция состоит из заголовка (шапки), который содержит атрибуты (обязательные реквизиты) и дополнительные реквизиты. Табличная часть Операции представляет собой строки с проводками, значит, параметры и структура описываются в объекте метаданных Проводка.

Свойства операции редактируются в окне редактирования Операция (рис. 10.20).

В этом же окне по кнопкам **Форма операции** и **Формы журнала** можно создать новые или отредактировать открывшиеся формы.

<b>№</b> Операция	
Длина содержания: 50 📑	
Реквизиты	Сумма операции
<b>1</b>	Длина: 15 🕂
	Точность: 2 🐳
↓	Разд. триады 🔽
₽IJ	Отбор
	✓ По сумме
Новый Изменить Удалить	🔽 По содержанию
Графы журнала	Расчет суммы
<b>1</b>	О Не рассчитывать
	По всем проводкам
	О По первой проводке
2I	П Валютная симма
Новый Изменить Удалить	ј∙ выоор режима
<u>Р</u> едактировать операции док	ументов: Запретить 💌
Описание Форма опера	ации Формы журнала 🕨

Рис. 10.20. Окно редактирования Операция

В данной книге мы не будем рассматривать подробно создание форм проводки и операции. Подробно об операциях и проводках написано в документации "Конфигурирование и администрирование". Как правило, на практике не конфигурируют эти объекты "с нуля", а используют уже существующие в типовых конфигурациях.

Теперь, согласно условиям сквозного примера, нам понадобятся справочники для хранения списков объектов, участвующих в бухгалтерском учете. Используем справочники, созданные в *главе б*.

Для организации расчетов дополним справочник "Основные средства" несколькими дополнительными реквизитами:

- □ Счет Затрат для указания счета учета затрат по начисленному износу. Тип значения — Счет.Основной;
- □ Вид Затрат для указания элемента субконто Затраты счета учета затрат. Тип значения Перечисление.Затраты.

При этом значения реквизитов Счет Затрат и Вид Затрат являются периодическими.

Настроим экранную форму списка справочника "Основные средства", при необходимости выводя в нее дополнительные реквизиты (рис. 10.21).

	_		
舅日	Список	основных средств	
ŕ			
	Kon	Наименование	
	INCA.		
II-	-		
	-		
II-	-		
⊪–			
⊪–			
⊪–			
⊪–	-		
⊪–	-		
⊫–			
		II. I	

**Рис. 10.21.** Экранная форма справочника "Основные средства"

Дополним конфигурацию документами, имеющими непосредственное отношение к бухгалтерскому учету.

Документ **Затраты** предназначен для отнесения на затраты сумм с любых счетов плана счетов.

Для создания "с нуля" документа "Затраты" воспользуемся, как и ранее, конструктором создания документов *(см. главу 7)*.

После указания на первом шаге Конструктора названия документа на втором шаге Конструктора выбираем режим **Разработать структуру документа "от проводки"** (рис. 10.22). Именно этот режим отражает особенность конструирования бухгалтерских документов.



**Рис. 10.22.** Выбор режима конструирования бухгалтерского документа

Состав реквизитов понятен из формы диалога документа (рис. 10.23).

При создании документов конструктором у нас были созданы автоматически предопределенные процедуры ПриОткрытии () модуля формы и ОбработкаПроведения () модуля документа (листинги 10.1 и 10.2).

#### Листинг 10.1

```
Процедура ПриОткрытии()
```

```
ПриЗаписиПерепроводить (1);
```

КонецПроцедуры

<u>第</u> 10m	несение на затраты				
Отне	Отнесение на затраты М				
Комм	ентарий: Комментарий				
N	На вид затрат				
	H				
	ОК Закрыть				



#### Листинг 10.2

```
Процедура ОбработкаПроведения ()
  //{{ФОРМИРОВАНИЕ ОПЕРАЦИИ
  //Данный фрагмент построен конструктором.
  //При повторном использовании конструктора внесенные вручную
  // изменения будут потеряны!!!
  ВыбратьСтроки();
  Пока ПолучитьСтроку() = 1 Цикл
  Операция. Новая Проводка ();
  Операция.Дебет.Счет =
     СчетПоКоду ("ЗТР", ПланыСчетов. Основной);
   Операция. Дебет. Затраты = Затраты;
   Операция.Кредит.Счет = Сч1;
   Операция. Кредит. Субконто (1, Субконто1);
   Если (Операция.Дебет.Счет.Количественный = 1)
     Или (Операция.Кредит.Счет.Количественный = 1) Тогда
     Операция. Количество = Количество;
```

КонецЕсли; Операция.Сумма = Сумма; КонецЦикла; Операция.СуммаОперации = Итог("Сумма"); Операция.Записать();

//} }ФОРМИРОВАНИЕ\_ОПЕРАЦИИ

КонецПроцедуры

Поясним вышеприведенный алгоритм модуля проведения.

Метод ВыбратьСтроки() формирует выборку строк табличной части документа для последующей обработки. Обращение и позиционирование на каждой следующей строке выборки организовано в цикле методом Получить-Строку().

Далее в цикле приведены операторы создания проводок операции документа (объект Операция). Для объекта Операция по методу НоваяПроводка() создается проводка операции, для которой следующие за методом операторы назначают соответствующие значения атрибутам проводки: счет дебета, аналитика дебета, счет кредита, 1-е субконто кредита, количество для количественного счета, сумма проводки.

После создания проводки и завершении работы цикла записываем сумму всей операции (Операция. СуммаОперации).

Метод Записать () производит запись в базу данных сформированной операции.

После создания шаблона документа конструктором можем внести изменения.

Добавим дополнительный реквизит Основание в табличную часть документа для указания информации об основании возникновения затрат и затем выведем его в форму диалога.

Используя информацию по основанию затрат, добавим в цикл модуля документа оператор, заполняющий содержание проводки этой информацией:

Операция.СодержаниеПроводки = Основание;

Далее введем документы "РКО" и "ПКО", предназначенные для учета движения денежных средств на счетах плана счетов.

Для создания документа "ПКО" с использованием конструктора на шаге определения списка проводок документа можно определить основную корреспонденцию, как поступление наличных денег на Дебет счета НАЛ (Наличные деньги) с корреспондирующего счета Сч1, вводимого в документ:

Конструктор Докуме	нта	x
Список проводок		
Счет-дебет	Счет-кредит	Добавить Изменить Удалить
Введите одну или нес которые должны фор корреспонденциях ми конструкторе (выбор использовании докум	жолько корреспонденций г мироваться документом. С огут указываться непосред ом из плана счетов) или вв иента.	проводок, Счета в цственно в одиться при
Опишите проводку		×
Дебет	. Кредит	
План счетов Осно	рвной	

Рис. 10.24. Шаг конструктора Список проводок

朝			
Номер: Номе	р Да		
Cv1:	Cu1		
Субконто1:	Субко		
Количество:	Колич		
Сумма:	Сумма		
Комментарий			
ОК			

**Рис. 10.25.** Диалоговая форма документа "ПКО" после создания шаблона документа конструктором

После создания шаблона документа конструктором (рис. 10.25) можем внести изменения.

Добавим реквизиты в шапку документа и затем выведем их в форму диалога:

**ПринятоОт** — для указания информации о лице, оплатившем деньги;

**Основание** — для указания информации об основании платежа.

Диалоговая форма документа после дополнения реквизитов будет выглядеть следующим образом:



Рис. 10.26. Диалоговая форма ПКО

В листинге 10.3 представлен модуль формирования проводок документов, созданный конструктором.

```
Листинг 10.3

Процедура ОбработкаПроведения()

//{{ФОРМИРОВАНИЕ_ОПЕРАЦИИ

//Данный фрагмент построен конструктором.

//При повторном использовании конструктора внесенные вручную

//изменения будут потеряны!!!

Операция.НоваяПроводка();
```

```
Операция.Дебет.Счет = СчетПоКоду ("НАЛ",ПланыСчетов.Основной);
```

```
Операция.Кредит.Счет = Сч1;
Операция.Кредит.Субконто(1,Субконто1);
Операция.Сумма = Сумма;
Операция.СуммаОперации = Сумма;
Операция.Записать();
//}}ФОРМИРОВАНИЕ_ОПЕРАЦИИ
КонецПроцедуры
```

# Дополним работу конструктора операторами по заполнению информации по содержанию проводки:

Операция.СодержаниеПроводки = ПринятоОт;

#### и по заполнению содержания операции:

Операция.Содержание = Основание;

Идентификатор: Выписка Комментарий: Реквизиты шапки Изменить Идал Новый Изменить Идал Номер Нумератор: << Не назнач
Комментарий: Реквизиты шапки Новый Изменить Удел Номер Нумератор: << Не назнач
Реквизиты шапки Новый Изменить Удал Номер Нумератор: << Не назнач
Новый Изменить Удал Номер Нумератор: << Не назнач
Новый Изменить Удаг Номер Нумератор: << Не назнач
Новый Иаменить Удаг Номер Нумератор: << Не назнач
Новый Изменить Удал Номер Нумератор: << Не назнач
Новый Изменить Удал Номер Нумератор: << Не назнач
Новый Изменить Идал Номер Нумератор: << Не назнач
Номер Нумератор: << Не назнач
<u>Н</u> умератор: << Не назнач
Периодичность: По всем дан
. 🔽 Автоматическая нумерация
Разрешить проведение доку
Автоматическое удаление д Автоматическое удаление д
Автоматическая нумерация
Создавать операцию: Всегда
<u>В</u> вод на основании

Рис. 10.27. Свойства и состав документа "Выписка"

В операторе присваивания счета кредита операции заменим значение Cч1 на значение реквизита КорСчет:

Операция.Кредит.Счет = КорСчет;

По аналогии с документом "ПКО" можно создать документ "РКО" — для учета расхода наличных денег.

При этом корреспонденция проводок документа "РКО" (расход) будет противоположной корреспонденции документа "ПКО" (приход).

От кассовых документов документ "Выписка" отличается тем, что реквизиты движения денег расположены в его табличной части. Реквизиты табличной части служат для отражения информации как о поступлении, так и о расходе безналичных денежных средств.

Если воспользоваться любым из знакомых уже нам методов создания документов, можно получить следующую структуру документа (рис. 10.27).

Реквизиты документа в диалоговой форме после работы конструктора можно расположить самостоятельно более компактно (рис. 10.28).



Рис. 10.28. Диалоговая форма документа "Выписка"

При создании конструктором документа в модуле формы автоматически создается процедура Приоткрытии () (листинг 10.4).

#### Листинг 10.4

```
Процедура ПриОткрытии()
ПриЗаписиПерепроводить(1);
КонецПроцедуры
//*****
```

В листинге 10.5 представлен модуль формирования проводок документа "Выписка", созданный конструктором.

#### Листинг 10.5

```
Процедура ОбработкаПроведения ()
 //{{ФОРМИРОВАНИЕ ОПЕРАЦИИ
  //Данный фрагмент построен конструктором.
  //При повторном использовании конструктора внесенные вручную
  //изменения будут потеряны!!!
ВыбратьСтроки();
Пока ПолучитьСтроку() = 1 Цикл
  Если Приход>0 Тогда
   Операция. Новая Проводка ();
   Операция.Дебет.Счет =
     СчетПоКоду ("БНС", ПланыСчетов. Основной);
   Операция.Кредит.Счет = КорСчет;
   Операция. Кредит. Субконто (1, Субконто1);
   Операция.Сумма = Приход;
   Операция.СодержаниеПроводки = Основание;
  ИначеЕсли Расход>0 Тогда
   Операция. Новая Проводка ();
   Операция.Дебет.Счет = КорСчет;
   Операция. Дебет. Субконто (1, Субконто1);
   Операция.Кредит.Счет =
     СчетПоКоду ("БНС", ПланыСчетов. Основной);
```

Операция.Сумма = Расход;

```
Операция.СодержаниеПроводки = Основание;
КонецЕсли;
КонецЦикла;
Операция.Содержание = Комментарий;
Операция.СуммаОперации = Итог("Приход")+Итог("Расход");
Операция.Записать();
//}}ФОРМИРОВАНИЕ_ОПЕРАЦИИ
```

КонецПроцедуры

Как видно, алгоритм формирования проводок документа "Выписка" включает в себя:

алгоритмы проводок по приходу денег, выполняющиеся в случае выполнения условного оператора

Если Приход>0 Тогда;

алгоритмы проводок по расходу денег, выполняющиеся в случае выполнения условного оператора

ИначеЕсли Расход>0 Тогда.

В основном же действия, выполняемые операторами модуля, схожи с действиями документов "ПКО" по приходу и "РКО" по расходу денег.

Проводки формируются по каждой строке табличной части документа. После создания проводок операции заполняются атрибуты объекта операции: Содержание и СуммаОперации.

Во всех документах по движению денег присутствует реквизит и элемент диалога **КорСчет**, от выбора значения которого будет зависеть установка типа значения реквизита **Субконто1**, имеющего тип значения "Неопределенный". Назначение реквизиту типа "Неопределенны" предполагает возможность назначения ему любого типа в зависимости от возникших ситуаций.

Для назначения реквизиту Субконто1 нужного типа в диалоговой форме в свойствах реквизита КорСчет на вкладке Дополнительно зададим формулу:

НазначитьТип("Субконто1", КорСчет.ВидСубконто(1))

В случае выбора счета, у которого в качестве субконто используются справочники "Сотрудники" или "Контрагенты", можно будет автоматически заполнить реквизит **ПринятоОт**. В модуле формы опишем процедуру ПриВыбореСубконто(), запускаемую при выборе значения реквизита **Субконто1**.

#### Листинг 10.6

Процедура ПриВыбореСубконто1() // вызывается при выборе значения реквизита КорСчет Если ТипЗначенияСтр(Субконто1) = "Справочник" Тогда Если Субконто1.Вид() = "Сотрудники" Тогда ПринятоОт = Субконто1.Наименование; ИначеЕсли Субконто1.Вид() = "Контрагенты" Тогда Если ПустоеЗначение(СокрЛП(Субконто1.Наименование)) = 0 Тогда ПринятоОт = СокрЛП(Субконто1.Наименование); КонецЕсли; КонецЕсли; КонецЕсли;

КонецПроцедуры

Созданные ранее документы мы можем теперь дополнить алгоритмами формирования проводок, если это необходимо по условиям сквозного примера.

Документ "Амортизация" относится к регламентным документам и служит для ежемесячного отражения в бухгалтерском учете сумм начисленной амортизации основных средств.

Документ содержит только общие реквизиты: номер, дату, комментарий (рис. 10.29).



Рис. 10.29. Диалоговое окно документа "Амортизация"

Документ является регламентным и должен проводиться только последним числом месяца. Для назначения реквизиту ДатаДок нужного значения (последнее число месяца) в диалоговой форме в свойствах реквизита ДатаДок на вкладке Дополнительно зададим формулу:

```
ДатаДок=КонМесяца (ДатаДок) ;
```

В модуль формы добавим процедуру для назначения нужной даты документа при вводе нового:

```
Процедура ВводНового()
```

```
ДатаДок=КонМесяца(РабочаяДата());
```

```
КонецПроцедуры
```

Все расчеты амортизации будут производиться в момент проведения документа. Алгоритмы генерации проводок будут представлены в *разд. 10.5.2*.

При создании документов "Списание материалов", "Списание ОС" не были определены типы реквизитов, и теперь мы устанавливаем типы значений реквизитов шапки документа:

- □ СчетЗатрат для указания счета учета затрат по списанию материалов или ОС устанавливаем тип значения "Счет.Основной";
- □ ВидЗатрат для указания элемента субконто Затраты счета учета затрат устанавливаем тип значения "Перечисление.Затраты".

Сохраним внесенные в конфигурацию изменения.



Рис. 10.30. Свойства реквизита ВидЗатрат

### 10.5.2. Алгоритмы объектов

После создания в конфигурации необходимых объектов можно приступать к написанию процедур в них, используя при необходимости ссылки на существующие счета плана счетов **Основной**.

Прежде всего, перед дополнением алгоритмов документов, уже созданных в конфигурации, необходимо дополнить свойства документов для отражения их участия в бухгалтерском учете.

Мы помним, что появление в конфигурации **Плана Счетов** привело к появлению возможности отнесения документа к компоненте "Бухгалтерский учет", недоступной до этого момента (рис. 10.31).

Реквизиты шапки	
Новый Изменить Уда	
Номер	
<u>Н</u> умератор: << Не назнач	
Периодичность: По всем дан	
. Автоматическая нумерация	
🛛 🔽 Разрешить проведение доку	
🛛 🔽 Автоматическое удаление д	
🛛 🔽 Автоматическая нумерация	

Рис. 10.31. Окно свойств документа

Если по нашим условиям документы "Поступление материалов", "Списание материалов", "Поступление ОС", "Списание ОС" будут не только записываться, но и создавать проводки при проведении, то определим для них принадлежность к компоненте "Бухгалтерский учет".

Двойным щелчком мыши по документу в структуре метаданных конфигурации откроем окно "Документ" и в параметрах документа (см. рис. 10.31) установим флажок **Бухгалтерский учет**.

Для документов, созданных конструктором с использованием режима "от проводки" ("ПКО", "РКО", "Выписка банка", "Затраты"), принадлежность к компоненте "Бухгалтерский учет" проставляется автоматически.

На примере документа "Поступление материалов" дополним модуль проведения алгоритмами создания проводок. Для этого откроем модуль документа и выберем меню **Конструкторы** пункт **Формирование операции...**. В окне конструктора задаем флажок **Проводка по каждой строке документа** для организации цикла по табличной части документа.

С помощью конструктора формируем последовательность операторов для записи двух проводок: Доход от реализации и Себестоимость товара.

💣 Ko	онструкто	р формирования	і операці	и				×
Зна Выр Но Вве "Вь док	чение (Стр ражение мерДок+"( адите соде дите форм пражение", умента.	ока) от"+ДатаДок ржание операции в кулу для заполнения Для заполнения м	Вид с 	убконто убконто азад Далее> чение" или чия в поле ильзовать данные	Данные до Но Данные до Данные до Даниые до Да	кумента мер та мментарий нтрагент нование ого("Количеств ог("Цена") ог("Сумма")	50'')	ОК Отмена Обновить Помощь
- One Har Hot	рация именован иерДок+"с	ние л"+ДатаДок	ан счетов	Ссновной	Сумма операц Ітог("Сумма")	<b>ии</b> оводка по ка>	кдой строке док	умента
	Дебе	СубконтоДт	Кред	СубконтоКт	Количе	Сумма	Содержание	проводки 🔺
	DOX 3TP	СебестоимостьТс	РАС.2 ТМЦ.1	Контрагент		Итог("Сумм СуммаСеб	Доход от реали Себестоимость	IS3ЦИИ

Рис. 10.32. Конструктор формирования операций

Для документа "Поступление товаров" с помощью конструктора формируем модуль (листинг 10.7) для записи операции по поступлению товаров: общая сумма поступивших товаров от поставщика.

#### Листинг 10.7

Процедура ОбработкаПроведения()

```
// { \{ \Phi O P M P O B A H N E_O \Pi E P A L I N \}
```

//Данный фрагмент построен конструктором.

```
//При повторном использовании конструктора внесенные вручную //изменения будут потеряны!!!
```

```
Операция.НоваяПроводка();

Операция.Дебет.Счет = СчетПоКоду("ТМЦ.1",ПланыСчетов.Основной);

Операция.Кредит.Счет = СчетПоКоду("РАС.1",ПланыСчетов.Основной);

Операция.Кредит.Контрагенты = Контрагент;

Операция.Сумма = Итог("Сумма");

Операция.СодержаниеПроводки = "Поступление товаров";

Операция.Содержание = НомерДок+"от"+ДатаДок;

Операция.СуммаОперации = Итог("Сумма");

Операция.Записать();

//}}ФОРМИРОВАНИЕ_ОПЕРАЦИИ

КонецПроцедуры
```

Таким же образом с помощью конструктора формируем модуль документа "Поступление материалов" для записи операции по поступлению материалов: сумма, количество и наименование поступивших материалов от поставщика.

Отличительной особенностью учета материалов от учета товаров является то, что согласно плану счетов и условиям сквозного примера мы ведем количественный и аналитический учет по материалам на счетах бухгалтерского учета.

В алгоритме формирования проводок в цикле по каждой строке табличной части появятся операторы для заполнения аналитики проводки

```
Операция.Дебет.Материалы = ТМЦ;
```

и оператор для присваивания количественной характеристики проводки

Операция. Количество = Количество;

Весь листинг мы не приводим в связи с похожестью его на алгоритм документа "Поступление товара".

Документы реализации товаров и списания материалов используются для отражения в учете себестоимости реализованного или списанного TM3. Расчет себестоимости списанного материала рассмотрим в *разд. 11.2*, поскольку при расчете будет использоваться выборка данных бухгалтерского учета.

На этом этапе конфигурирования для расчета себестоимости списанного материала в соответствии с действующим методом списания "по средней" выведем только процедуру Рассчитать Себестоимость (Материал).

Алгоритмы процедуры будут рассмотрены далее, а на этом этапе пропишем возврат значения себестоимости из периодического реквизита справочника "Материалы", получаемого на дату документа.

#### Листинг 10.8

Алгоритм формирования проводок (листинг 10.9) аналогичен алгоритму формирования проводок при поступлении материала. Сумма проводки возвращается после расчетов функцией Рассчитать Себестоимость ().

#### Листинг 10.9

```
Процедура ОбработкаПроведения ()
  //{{ФОРМИРОВАНИЕ ОПЕРАЦИИ
  //Данный фрагмент построен Конструктором.
  //При повторном использовании конструктора внесенные вручную
  //изменения будут потеряны!!!
  CymmaCe6=0;
  ВыбратьСтроки();
  Пока ПолучитьСтроку() = 1 Цикл
  Операция. Новая Проводка ();
   Операция.Дебет.Счет =
     СчетПоКоду ("ЗТР", ПланыСчетов. Основной);
   Операция. Дебет. Затраты = ВидЗатрат;
   Операция.Кредит.Счет =
     СчетПоКоду ("ТМЦ.2", ПланыСчетов. Основной);
   Операция. Кредит. Материалы = ТМЦ;
   Операция.Количество = Количество;
   //рассчитаем себестоимость списания
   Себ=РассчитатьСебестоимость (ТМЦ);
```

```
Операция.Сумма = Себ;
Операция.СодержаниеПроводки = ТМЦ.Наименование;
// увеличим себестоимость подокументу
СуммаСеб=СуммаСеб+Себ;
КонецЦикла;
Операция.Содержание = "Списание материалов";
Операция.СуммаОперации = СуммаСеб;
Операция.Записать();
//}}ФОРМИРОВАНИЕ_ОПЕРАЦИИ
КонецПроцедуры
```

С помощью конструктора, используя имеющиеся у нас навыки, формируем модуль документа "Реализация товаров" для записи операции по списанию себестоимости товаров: общая сумма списанной на затраты себестоимости.

Для расчета себестоимости реализованного товара в соответствии с методом, указанным в константе Метод списания, выведем процедуру Рассчитать Себестоимость (Товар).

Алгоритмы процедуры будут рассмотрены в следующих разделах, а на этом этапе пропишем возврат значения себестоимости, равный общей сумме реализованного товара табличной части документа.

```
Листинг 10.10
```

Функция РассчитатьСебестоимость (Товар)

```
//здесь следует написать алгоритм расчета себестоимости товаров
Возврат Итог("Сумма");
КонецФункции
```

Алгоритм формирования проводок (листинг 10.11) аналогичен алгоритму формирования проводок при поступлении товаров.

Сумма по первой проводке определяет сумму дохода, полученную от контрагента-покупателя.

Сумма второй проводки определяет себестоимость всех товаров без детализации и возвращается после расчета функцией РассчитатьСебестоимость ().

#### Листинг 10.11

Процедура ОбработкаПроведения()

//{{ФОРМИРОВАНИЕ\_ОПЕРАЦИИ

//Данный фрагмент построен Конструктором.

```
//При повторном использовании конструктора внесенные вручную //изменения будут потеряны!!!
```

```
Операция.НоваяПроводка();
Операция.Дебет.Счет = СчетПоКоду("ДОХ",ПланыСчетов.Основной);
Операция.Кредит.Счет = СчетПоКоду("РАС.2",ПланыСчетов.Основной);
Операция.Кредит.Контрагенты = Контрагент;
Операция.Сумма = Итог("Сумма");
Операция.СодержаниеПроводки = "Доход от реализации";
```

```
Операция.НоваяПроводка();
Операция.Дебет.Счет = СчетПоКоду("ЗТР",ПланыСчетов.Основной);
Операция.Дебет.Затраты = Перечисление.Затраты.СебестоимостьТов;
Операция.Кредит.Счет = СчетПоКоду("ТМЦ.1",ПланыСчетов.Основной);
Операция.Сумма = РассчитатьСебестоимость (ТМЦ);
Операция.СодержаниеПроводки = "Себестоимость товаров";
```

```
Операция.Содержание = НомерДок+"от"+ДатаДок;
Операция.СуммаОперации = Итог("Сумма");
Операция.Записать();
//}}ФОРМИРОВАНИЕ ОПЕРАЦИИ
```

КонецПроцедуры

Переходим к формированию алгоритмов следующих документов. Документ "Поступление ОС" отражает в бухгалтерском учете операцию с суммой поступившего ОС от поставщика.

Согласно плану счетов и условиям сквозного примера мы не ведем количественный учет по ОС на счетах бухгалтерского учета.

Выберем знакомый уже нам пункт меню Конструкторы | Формирование операции....

После создания модуля документа конструктором добавим в построчный цикл оператор для записи периодического значения начальной стоимости ОС в справочник "Основные средства" (листинг 10.12).

#### Листинг 10.12

```
Процедура ОбработкаПроведения()

//{{фОРМИРОВАНИЕ_ОПЕРАЦИИ

//Данный фрагмент построен конструктором.

//При повторном использовании конструктора внесенные вручную

//изменения будут потеряны!!!

ВыбратьСтроки();

Пока ПолучитьСтроку() = 1 Цикл

Операция.НоваяПроводка();

Операция.Дебет.Счет = СчетПоКоду("ОС",ПланыСчетов.Основной);

Операция.Дебет.ОС = ТМЦ;

Операция.Кредит.Счет =

СчетПоКоду("РАС.1",ПланыСчетов.Основной);

Операция.Кредит.Контрагенты = Контрагент;

Операция.Сумма = Сумма;

Операция.СодержаниеПроводки = ТМЦ.Наименование;
```

//Запись начальной стоимости в справочник

Установить РеквизитСправочника (ТМЦ, "НачальнаяСтоимость", Сумма, ДатаДок);

КонецЦикла; Операция.Содержание = "Поступление ОС"; Операция.СуммаОперации = Итог("Сумма"); Операция.Записать(); //}}ФОРМИРОВАНИЕ\_ОПЕРАЦИИ Модуль документа "Списание ОС" (листинг 10.13) формируется также конструктором.

Операция по списанию ОС состоит из двух проводок:

- списывающей на счет основного средства (ОСС) сумму износа со счета накопления износа (ИЗН);
- списывающей на счет затрат (ЗТР) сумму остаточной стоимости ОС (Сумма – Износ) со счета учета основного средства (ОСС).

Проводки формируются по каждому основному средству, т. е. по каждой строке табличной части документа.

Для списанного OC отменяем участие в расчете амортизации, устанавливая значение периодического реквизита **НачислятьАмортизацию** равным 0.

Значения, используемые для расчета сумм проводок (Сумма и Износ), рассчитываются при заполнении табличной части документа. Эти алгоритмы будут описаны в *разд. 11.2.* На этом этапе мы предполагаем, что в табличной части документа все реквизиты заполнены до момента формирования проводок.

#### Листинг 10.13

Процедура ОбработкаПроведения()

//{{ФОРМИРОВАНИЕ ОПЕРАЦИИ

//Данный фрагмент построен конструктором.

//При повторном использовании конструктора внесенные вручную //изменения будут потеряны!!!

ВыбратьСтроки();

Пока ПолучитьСтроку() = 1 Цикл

Операция.НоваяПроводка();

Операция.Дебет.Счет = СчетПоКоду ("ИЗН", ПланыСчетов.Основной);

Операция.Дебет.ОС = ТМЦ;

Операция.Кредит.Счет = СчетПоКоду ("ОСС", ПланыСчетов.Основной);

Операция.Кредит.ОС = ТМЦ;

Операция.Сумма = Износ;

```
Операция.НоваяПроводка();

Операция.Дебет.Счет =

СчетПоКоду("ЗТР",ПланыСчетов.Основной);

Операция.Дебет.Затраты = ВидЗатрат;

Операция.Кредит.Счет =

СчетПоКоду("ОСС",ПланыСчетов.Основной);

Операция.Кредит.OC = ТМЦ;

Операция.Сумма = Сумма-Износ;

Операция.СодержаниеПроводки = "списание "+ТМЦ.Наименование;

//начисления амортизации списанного ОС в справочнике
```

```
Установить РеквизитСправочника (ТМЦ, "Начислять Амортизацию", 0, ДатаДок);
КонецЦикла;
```

```
Операция.Содержание = "Списание ОС";
Операция.СуммаОперации = Итог("Сумма")-Итог("Износ");
Операция.Записать();
//}}ФОРМИРОВАНИЕ_ОПЕРАЦИИ
```

КонецПроцедуры

232

Закончив редактирование документов, сохраните конфигурацию. Запустите программу в режиме "1С:Предприятие" и проверьте правильность работы алгоритмов.



# Бухгалтерские итоги, запросы и отчетность

Для работы с информацией, накопленной на счетах бухгалтерского учета, в системе "1С:Предприятие" существует специальный объект "Бухгалтерские итоги". Компонента предоставляет возможность организации выборки данных бухгалтерских итогов, используя запросы. Результаты выборки могут использоваться и в дальнейших расчетах, и при формировании различных форм отчетов.

# 11.1. Понятие о бухгалтерских итогах и запросах

Компонента "Бухгалтерские итоги" хранит бухгалтерские итоги (остатки и обороты по счетам) за любой рассчитанный период. Поэтому при построении отчетов можно быстро получить необходимые данные.

Для работы с объектом "Бухгалтерские итоги" используется метод встроенного языка Создать Объект ("Бухгалтерские Итоги").

Итоги по остаткам и оборотам хранятся с учетом аналитики плана счетов. Без учета аналитики хранятся итоги по оборотам между счетами.

В информационной базе существует возможность контроля расчета бухгалтерских итогов.

Запустите программу в режиме "1С:Предприятие" и в меню Операции выберите пункт Управление бухгалтерскими итогами.

В верхней строке диалогового окна показан период, по который в информационной базе имеются рассчитанные итоги. Это означает также, что при

обращении к бухгалтерским итогам без указания периода выборка будет строиться именно по информации указанного периода.



Рис. 11.1. Диалоговое окно Управление бухгалтерскими итогами

Чтобы установить следующий период, надо нажать кнопку Установить расчет, а затем кнопку Полный пересчет итогов.

y	правление бухгалт
	Расчет итогов устанс
	9становить расч
	Полный пересчет и

Рис. 11.2. Диалоговое окно Управление бухгалтерскими итогами с измененным периодом

Кнопкой пересчета итогов можно пользоваться независимо от кнопки установки расчета, например, в случае сбоя в данных.

## 11.2. Использование бухгалтерских запросов в алгоритмах документов

Для работы с объектом типа "Бухгалтерские итоги" имеется набор методов, которые позволяют обращаться к остаткам и оборотам по счетам. Имена

этих методов записываются через точку после имени переменной, содержащей ссылку на объект "Бухгалтерские итоги".

Описание методов работы с объектом "Бухгалтерские итоги" и их параметрами можно найти в документации по встроенному языку, поставляемой с программным обеспечением, или в синтакс-помощник.

Вернемся к условиям нашего примера.

Как и обещали в *славе 10*, добавим к документам алгоритмы или отдельные операторы, выбирающие данные из учета. Например, в документе "СписаниеОС" дополним модуль формы алгоритмами заполнения реквизитов Сумма и Износ в табличной части.

Для реквизита табличной части ТМЦ, имеющего тип "Справочник.ОсновныеСредства", пропишем процедуру ПриВыбореОС() для заполнения реквизитов строки.

В модуль формы добавим процедуру ПриВыбореОС() (листинг 11.1).

#### Листинг 11.1

Процедура ПриВыбореОС(ОС) КонецПроцедуры

Назначение процедуры — получить из бухгалтерских итогов значение начальной стоимости для заполнения реквизита табличной части Сумма и значение накопленного износа для заполнения реквизита табличной части Износ.

Создадим объект Бухгалтерские итоги:

БухИтоги=СоздатьОбъект ("БухгалтерскиеИтоги");

Из бухгалтерских итогов будем получать значения для конкретного элемента аналитики **ОС** вида субконто **ОС**, передаваемого в процедуру в виде параметра. Поэтому перед выполнением запроса установим режим отбора по виду субконто и по конкретному значению этого субконто:

БухИтоги.ИспользоватьСубконто(ВидыСубконто.ОС, ОС, 2);

Цифра 2 означает тип фильтра: режим отбора по указанному значению.

После этого можно выполнять запрос по указанным параметрам фильтра:

БухИтоги.ВыполнитьЗапрос(,ТекущийДокумент(),"ОСС,ИЗН", , , 1, , 1);

При этом обращаем внимание, что дату начала выборки не указываем, а дату конца выборки указываем, как позицию документа.

После выполнения запроса формируем выборку для работы:

БухИтоги.ВыбратьСубконто();

Если же по назначенному фильтру не найдено ни одного итога, тогда нет смысла и обращаться к ним. Проверим наличие в выборке записей по субконто **ОС** (листинг 11.2).

#### Листинг 11.2

```
Если БухИтоги.ПолучитьСубконто(,,ОС) = 0 Тогда
Сообщить("В учете нет информации по ОС - "
+СокрЛП(ОС.Наименование)+" Возможно документ не записан!");
Возврат;
КонецЕсли;
```

Метод Получить Субконто () позволяет позиционироваться на конкретной записи выборки, если таковая попала в запрос, если нет — завершить процедуру (Возврат).

Обращаем внимание, что если эта процедура будет вызвана для документа, не имеющего позиции (документ не записан), то выборка может оказаться также пустой. Поэтому, информируя пользователя об отсутствии в учете информации по основному средству, предлагаем ему проверить, записан ли документ. После записи документа в случае появления этого сообщения пользователь должен проверить наличие в учете информации по основному средству.

Как вы помните, в нашей выборке могут оказаться итоги по двум счетам указанного субконто:

- счет ОСС по дебету счета отражалась информация о стоимости основного средства;
- □ счет **ИЗН** по кредиту счета отражалась информация о накопленном износе.

Значение реквизита Износ получим так:

Износ=?(БухИтоги.ПолучитьСчет(, "ИЗН") = 1, БухИтоги.СНК(1), 0);

Метод ПолучитьСчет() позиционирует нас на строке с итогами по счету Изн. Условный оператор описывает следующие действия:

- □ если итог по счету найден, то мы берем значение сальдо, начального по кредиту (CHK(1));
- □ если итог не найден, значение равно 0. Параметр 1 при обращении к итогам означает обращение к итогам типа "Сумма".

Используя условный оператор, получим значение стоимости основного средства для заполнения реквизита Сумма:

```
Сумма=?(БухИтоги.ПолучитьСчет(, "ОСС") = 1, БухИтоги.СНД(1), 0);
```

На этом написание процедуры закончено.

Для документа "Списание материалов" дополним модуль документа алгоритмами расчета себестоимости списанных материалов.

Назначение процедуры РассчитатьСебестоимость () — получить из бухгалтерских итогов значение остатка материала и остаточной стоимости материала. Полученные значения передать для формирования параметров проводки в модуль проведения.

В связи с тем, что нам рассчитывать себестоимость материалов необходимо по каждой строке табличной части, выполним запрос один раз и перед обращением к итогам будем позиционироваться на итогах по конкретному значению материала.

Для этого в начале модуля документа объявим общую переменную:

Перем БухИтоги;

Операторы по запросу к бухгалтерским итогам разместим в конце модуля после всех процедур. Создадим вначале объект Бухгалтерские итоги:

БухИтоги =СоздатьОбъект ("БухгалтерскиеИтоги");

Из бухгалтерских итогов будем получать значения для вида субконто "Материалы", поэтому перед выборкой значений итогов установим фильтр по виду субконто:

БухИтоги.ИспользоватьСубконто (ВидыСубконто.Материалы,, 1);

Режим 1 в фильтре означает "разворачивать" по виду субконто, т. е. выбирать все возможные значения итогов для указанного вида субконто.

После этого можно выполнять запрос по указанным параметрам фильтра (листинг 11.3).

#### Листинг 11.3

```
БухИтоги.ВыполнитьЗапрос(ТекущийДокумент(), ТекущийДокумент(), "ТМЦ.2 ", , , 1, , 5);
```

При этом обращаем внимание, что дата начала выборки не устанавливается, так как нас будут интересовать остатки на конец периода (на позицию перед документом). Запрос будет выполняться по счету учета материалов. Цифра 5 означает тип выбираемых итогов: сумма и количество, поскольку на счете ведется количественный учет, и нам необходимы для работы оба вида итогов.

После написания операторов основного модуля переходим к процедуре расчета себестоимости. В процедуре мы обращаемся к результатам запроса и нам необходимо позиционироваться на итогах по конкретному материалу, переданному в виде параметра процедуры. Как и в документе "Списание-ОС", перед позиционированием формируем выборку

БухИтоги.ВыбратьСубконто();

и потом позиционируемся на значениях итогов по Материалу (листинг 11.4).

#### Листинг 11.4

```
Если БухИтоги.ПолучитьСубконто(,,Материал) = 0 Тогда
Сообщить("В учете нет информации по материалу – "
+СокрЛП(Материал.Наименование)
+" Возможно документ не записан!");
Возврат 0;
КонецЕсли;
```

Если документ записан и информация по материалу в учете присутствует, для определения остатка по количеству материалов используем итог по сальдо начальному дебетовому с указанием вида остатка:

ОстКол=БухИтоги.СКД("К");

По аналогии получим суммовой остаток:

ОстСумма=БухИтоги.СКД("С");

В зависимости от остатка материала на счете или спишем указанное в документе количество, или возвращаем нулевое значение и сообщим пользователю о нехватке материала, если остаток меньше списываемого количества. Если количество позволяет, рассчитываем сумму для списания, округляя результаты арифметических действий.

Таким образом, модуль выглядит как в листинге 11.5.

#### Листинг 11.5

```
Если ОстКол<Количество Тогда
Сообщить ("На складе нет необходимого количества материала - "
+СокрЛП (Материал.Наименование));
Возврат 0;
КонецЕсли;
Себестоимость=Окр (ОстСумма*Количество/ОстКол);
Возврат Себестоимость;
```

На этом написание процедуры расчета себестоимости закончено.

Внесем изменения в модуль проведения по тем операторам, которые связаны с выводом количества и суммы списанных материалов в проводке.

После строки

СуммаСеб=СуммаСеб+Себ;

дополним строки, заполняющие табличную часть документа, рассчитанными значениями для удобства просмотра документа (листинг 11.6).

```
Листинг 11.6
```

```
// заполним табличную часть результатами расчета – для справки 
Сумма=Себ;
```

При этом элементу диалога, соответствующего реквизиту табличной части Сумма, устанавливаем в свойствах запрет редактирования.

На этом написание модуля проведения документа "Списание материалов" можно считать законченным.

Для документа "Выписка" дополним экранную форму полями с информацией об остатках денежных средств на начало дня и на конец дня, получаемой из бухгалтерских итогов.

Чтобы рассчитать эти значения по бухгалтерским данным, в модуле формы перед началом расположения процедур поместим строку с описанием переменной:

Перем БухИтоги;

В модуле формы после текстов всех процедур поместим установку значений для переменной Бухитоги для возможности обращения к ней в контексте диалога (листинг 11.7).

```
Листинг 11.7
```

```
// Установка основных значений
БухИтоги=СоздатьОбъект("БухгалтерскиеИтоги");
БухИтоги.ВыполнитьЗапрос(ДатаДок,ДатаДок,СчетПоКоду("БНС"),,,1,,1);
```

Теперь к результатам запроса мы можем обращаться из любого места формы. Поэтому добавим в экранную форму диалога элементы типа "Текст" и опишем в его свойствах формулы (рис. 11.3).

Выписка банка № НомерДок о ДатаДо				
Свойства Текста (НачОст) Свойства Текста (КонОст)				
Общие Дополнительно Шрифт Оп Общие Дополнительно Шрифт Описание				
Формула: БухИтоги.СНД("С") Формула: БухИтоги.СКД("С")				
Положение: Прижать влево Положение: Прижать влево 💌				
🗖 Центр по вертикали 🗖 Центр по вертикали				
🗖 Прозрачный фон 📃 Прозрачный фон				
Начальный остаток: Конечный остаток:				
ОК Закрыть КНачОст> КонОст>				

Рис. 11.3. Формулы для начального и конечного остатка

Закончив редактирование документов, сохраните конфигурацию и, запустив программу в режиме "1С:Предприятие", проверьте правильность работы алгоритмов.

## 11.3. Понятие о бухгалтерских отчетах

В типовых решениях "1С:Бухгалтерский учет" содержится большое количество отчетов, которые позволяют получать всю необходимую информацию в различных разрезах.

Все отчеты можно разделить на три группы:

🗖 стандартные;

□ регламентированные;

🗖 специализированные.

Стандартные отчеты предназначены для выдачи остатков и оборотов в различных разрезах для любых счетов, видов субконто, валют, в сумме и в количестве независимо от вида деятельности организаций. Это отчеты по плану счетов, такие как: "Оборотно-сальдовая ведомость", "Анализ счета", "Карточка счета" и др.

Регламентированные отчеты предназначены для выдачи информации в соответствии с регламентами контролирующих инстанций: налоговые инспекции, фонды, органы статистики. Как правило, разработчиком этих отчетов для типовых конфигураций является фирма "1С". По мере изменения требований к отчетности производятся изменения отчетности в программе, которые распространяются среди зарегистрированных пользователей в виде обновлений (релизов).

Специализированные отчеты предназначены для выдачи информации по конкретному разделу учета, по специфической настройке плана счетов конкретной конфигурации. Как правило, отчеты создаются в том случае, если требуется специфическая выборка информации или особый вид печатной формы. В типовой конфигурации к такому виду отчета относится кассовая книга.

Обычно отчеты для бухгалтерского учета создаются в тех случаях, когда необходимо представить бухгалтерскую информацию в виде, доступном неспециалистам. В последнее время возросла потребность в отчетности по внутренним управленческим формам на основании бухгалтерской информации.

# 11.4. Примеры. Создание отчета "Ведомость амортизации"

В качестве примера рассмотрим создание отчета по начисленной амортизации основных средств. Отчет будет выводить следующую информацию по каждому основному средству на дату формирования отчета:

- первоначальная стоимость дебетовый оборот счета ОСС;
- П накопленный износ кредитовый оборот счета ИЗН;
- остаточная стоимость разница дебетового оборота счета ОСС и кредитового оборота счета ИЗН.

Отчет будем формировать методом обращения к бухгалтерским итогам с использованием фильтра по субконто ОС.

Для создания нового отчета в Конфигураторе откройте окно **Конфигурация** и перейдите к ветви **Отчеты** дерева метаданных. Выберите пункт **Новый эле-мент** из меню **Действия** главного меню Конфигуратора.

В открывшемся диалоговом окне конструктора отчетов укажите идентификатор **ВедомостьАмортизации** и синоним **Ведомость амортизации**. С помощью конструктора пропишем новый отчет в меню.

На шаге конструктора, где предлагается выбрать режимы работы, выберем режим Вызвать конструктор макета отчета.

Нажав кнопку Готово, мы заканчиваем работу в конструкторе и переходим в конструктор макета отчета (рис. 11.4). В открывшемся окне укажем название

процедуры формирования отчета. Можем оставить название **Сформировать**, предложенное конструктором.

Конструктор макета отче	
Заголовок: Сформироват	
Отчет содержит:	
О Отдельные значения	
🖲 Циклы по строкам	
О Циклы по строкам и г	
О Только заголовок	
🔲 Диаграмму	
Введите заголовок отчет схем построения отчета	
размещена диаграмма.	

Рис. 11.4. Диалоговое окно Конструктор макета отчета



Рис. 11.5. Форма конструктора для назначения параметров

Перейдем в следующее окно конструктора по кнопке Далее.

На следующем шаге в открывшейся форме Конструктор макета отчета (рис. 11.5) укажем содержание отчета — Циклы по строкам. В окне Параметры укажем количество циклов по строкам не менее 1, а количество колонок — не менее 4, для вывода названия, первоначальной стоимости, накопленного износа, остаточной стоимости.

После назначения дополнительных значений отчета — кнопка вызова, режим просмотра — работа конструктора закончена. Можно вносить изменения.

В экранную форму введем элемент типа "Текст" для заголовка отчета (рис. 11.6).

💻 Форма-Отчет. Ведомость			
Ведомость амортизации Ведомость аморт на дату ДатаОг	изации		×1
Сформировать Закрыть	Общие Дополн Заголовок: Идентификатор:	ительно Шрифт Описание Ведомость амортизации Сделать недоступным Сделать невидимым Запретить редактирование	
 \ 🗐 Диалог 🛛 🗍 Модуль	🔪 🖄 Таблица	🔪 🖄 Сформировать	🔪 🖄 ооблр 🛛 🖊

Рис. 11.6. Форма диалога с заголовком

Теперь настроим созданную конструктором печатную форму отчета. Конструктор сформировал макет печатной формы из трех секций: Шапка, Строка\_1 для табличной части и Подвал (рис. 11.7).

В шапку отчета выведем текст **Ведомость амортизации**. Строкой ниже выведем информацию о дате формирования отчета. Затем в строке с шапкой изменим названия столбцов. Измененная форма отчета показана на рис. 11.8.

На этом формирование шапки отчета закончено.

Теперь сформируем строку отчета в соответствии с шапкой табличной части.

В каждой ячейке укажем название переменной, указав тип значения ячейки "Выражение", т. е. эти значения будут вычисляться (рис. 11.9).

		1	2	3	4	5	6
	1						
Шапка	2						
	3						
	4		Колонка_1	Колонка_2	Колонка_З	Колонка_4	
	5						
Строка_1	6						
	7						
Подвал	8						
	9						

Рис. 11.7. Форма отчета, созданная конструктором

		1	2	3	4	5	6	
	1							
Шапка	2		Ведомость амортизации					
	3		<РабочаяДата()>					
	4		Основное средство	Первоначальная стоимость	Накопленный износ	Остаточная стоимость		
	5							
Строка_1	6							
	7							
Подвал	8							
	9							



📕 Форма-Отчет.Ведомость (Ведомость амортизации)										
		1	2	3	4	5	6 🔺			
	1									
Шапка	2			Ведомость амор	тизации					
	3		<РабочаяДата()>							
	4		Основное средство	Первоначальная стоимость	Накопленный износ	Остаточная стоимость				
	5			j						
Строка_1	6		<Наименование>	<начСтоимость#N015.2>	8/ 8/ 8/</th <th>оимость#N015.2&gt;</th> <th></th>	оимость#N015.2>				
-	7									
Подвал	8	ļ	Итоги:	<ИтонНачСтоимость#N01	<ИтогИзнос#N01	<ИтогОстСтоимос				
	9	-	Свойства ячей	(M		×				
	10									
	11		Текст Положе	Текст Положение Шрифт Рамка Узор 🛛 🧛 🗕						
	12		Тип Выражен							
	14									
	15		НачСтоимость#N015.2							
	16				_	пк 1				
	17				7					
	18		Расшифровка			Отмена	-			
<ul> <li>■</li> <li>ВЛ Диало</li> </ul>	рг	λ	📰 М 📃 Защита	,		Обновить				

Рис. 11.9. Форма отчета со строкой и свойствами ячейки

Для числовых значений укажем строку форматирования (подробнее смотрите в документации), определяющую общую длину числа (15 знаков), число знаков после запятой (2 знака) и разделитель целой и дробной части (знак "."):

В соответствии с правилами форматирования чисел в ячейках расположим их с выравниванием по правой стороне для удобства визуального восприятия чисел.

Устанавливаем для ячеек строки параметр Обвести.

В подвале отчета выводим информацию по итогам колонок, настраивая ячейки по аналогии с настройками табличной части.

Следующая задача — разработка алгоритма отчета. Перейдем на вкладку **Модуль**, где находится созданная конструктором заготовка процедуры (листинг 11.8).

#### Листинг 11.8

```
Процедура Сформировать()
```

```
//Конструкции <<...>> необходимо заменить на реальные логические условия //и значения переменных
```

```
Процедура Сформировать()
```

```
Таб = СоздатьОбъект ("Таблица");
```

```
Таб.ИсходнаяТаблица("Сформировать");
```

Таб.ВывестиСекцию("Шапка");

```
Таб.Опции (0,0,Таб.ВысотаТаблицы(),0);
```

Пока <<Условие>> Цикл

```
Таб.ВывестиСекцию("Строка 1");
```

КонецЦикла;

```
Таб.ВывестиСекцию("Подвал");
```

```
Таб.ТолькоПросмотр(1);
```

Таб.Показать ("Сформировать", "");

КонецПроцедуры

Поскольку будем работать с бухгалтерскими итогами, начнем с создания объекта:

```
БИ = СоздатьОбъект ("БухгалтерскиеИтоги");
```
Создание объекта предполагает работу со всеми итогами. Будем вводить дополнительные условия для ограничения выборки. Сначала укажем, что нам нужны итоги только по основным средствам. Этот метод мы уже использовали в алгоритмах документов. Сейчас он будет выглядеть так:

```
ЕИ.ИспользоватьСубконто (ВидыСубконто.ОС,,1);
```

Далее вводим ограничения выборки по периоду и счету. Эти параметры используются в момент выполнения запроса методом Выполнить Запрос(). Вводим следующую строку в процедуру:

```
БИ.ВыполнитьЗапрос(Дата("01.01.1990"), РабочаяДата(), «ОСС, ИЗН» , , , 1, , 1);
```

Для нашего примера мы указали дату возможного начала ведения учета (01.01.1990), параметр конечной даты (т. е. определили предельную дату расчета итогов) и коды счетов, по которым будем выбирать итоги.

В связи с тем, что мы будем обращаться к оборотам по счетам, задание начальной даты выборки является обязательным. Мы указали произвольную возможную дату, так как для каждого элемента аналитики она может быть другой. Остальные параметры метода не конкретизируем. Ознакомиться с ними можно в документации по встроенному языку или в синтакс-помощник. Некоторые параметры были описаны при дополнении алгоритмов в документах "Списание ОС".

После выполнения запроса мы получим набор итогов по указанным счетам (ОСС, ИЗН) по всем возможным элементам указанного вида субконто (ОС). Для начала обработки итогов по субконто введем оператор, открывающий обработку отобранных субконто:

БИ.ВыбратьСубконто();

Далее остаются созданные конструктором операторы формирования таблицы для печати и вывода шапки.

Перед операторами цикла подготовим переменные для сбора итоговых значений в отчете (листинг 11.9).

#### Листинг 11.9

```
//Начальные значения итогов:
ИтогНачСтоимость=0;
ИтогИзнос=0;
ИтогОстСтоимость=0;
```

В качестве условия работы цикла задаем условие позиционирования на очередном итоге бухгалтерского запроса, т. е. до тех пор, пока можно получить очередной итог методом ПолучитьСубконто().

Таким образом, в цикле организуем перебор всех попавших в запрос элементов для определения показателей, выводимых в отчет (листинг 11.10).

#### Листинг 11.10

```
Пока ЕИ.ПолучитьСубконто()=1 Цикл

// для каждого элемента аналитики определяем реквизиты

// для вывода в отчет

Наименование=ЕИ.Субконто();

Износ=?(БухИтоги.ПолучитьСчет(, "ИЗН") = 1, БухИтоги.КО(), 0);

НачСтоимость=?(БухИтоги.ПолучитьСчет(, "ОСС")=1,БухИтоги.ДО(),0);

ОстСтоимость=НачСтоимость-Износ;

// вывод строки отчета

Таб.ВывестиСекцию("Строка");

//Накопление итогов

ИтогНачСтоимость=ИтогНачСтоимость+НачСтоимость;

ИтогИзнос=ИтогИзнос+Износ;

ИтогОстСтоимость=ИтогОстСтоимость+ОстСтоимость;

КонецЦикла;
```

Как видно по приведенному тексту процедуры, по мере обработки каждого элемента запроса выводятся строки в табличную часть отчета и накапливаются итоговые значения для вывода в "подвале" отчета.

После формирования строк табличной части отчета выполняем вывод секций отчета с итоговыми данными, созданных конструктором (см. листинг 11.8).

Заключительными являются операторы назначения параметров и опций таблицы (см. листинг 11.8). Также заменим оператор назначения названия заголовка окна с готовым табличным документом на оператор:

Таб.Показать ("Ведомость амортизации основных средств");

Теперь изменим условия формирования отчета: дополним возможность формирования на текущую дату возможностью формирования на любую дату, указанную пользователем.

Для этого выполним следующие действия.

- 1. В экранную форму введем реквизит диалога типа "Дата".
- 2. Заменим в модуле отчета метод РабочаяДата() на идентификатор реквизита диалога ДатаОтчета.
- 3. Заменим в шапке таблицы метод РабочаяДата () на идентификатор реквизита диалога ДатаОтчета.
- 4. Добавим в модуль отчета процедуру для заполнения реквизита в момент открытия формы отчета (листинг 11.11).

#### Листинг 11.11

```
Процедура ПриОткрытии()
ДатаОтчета=РабочаяДата ();
КонецПроцедуры
```

Закончив редактирование документов, сохраните конфигурацию и, запустив программу в режиме "1С:Предприятие", проверьте правильность работы алгоритмов.

3	🛿 Ведо	мость амортизации	_ 🗆	X		
	6.61	S ≥				
		Ведомость амо				
	С <u>ф</u> ор	мировать Закрыть				
📓 Сформировать	-			-		
	Ведомость амортизации					
30.09.06						
Основное сре	дство	Первоначальная	Накопленный износ	Остаточная		
0001		10000.00	166.66	9833.34		
Итоги:		10000.00	166.66	9833.34		
•						

Рис. 11.10. Форма отчета в режиме предприятия

### 11.5. Примеры. Создание отчета "Оборотно-сальдовая ведомость"

В предыдущем примере мы познакомились с приемами построения отчета от макета отчета. Для создания отчета "Оборотно-сальдовая ведомость" снова воспользуемся возможностями конструктора и внесем изменения в соответствии с требованиями нашей учебной конфигурации.

Итак, создадим отчет вида оборотно-сальдовой ведомости с возможностью формирования в следующих разрезах:

- □ За выбранный период;
- □ По выбранному счету;
- По счету в целом или с детализацией по субсчетам;
- По выбранному виду субконто или без раскрытия по субконто;
- **П** по выбранному списку значений аналитики или по всем.

Как и в предыдущем примере создадим новый отчет конструктором. Назовем его "OCB", вставим команду в пользовательское меню, вызовем конструктор бухгалтерских запросов на одной из вкладок конструктора. После создания шаблона отчета конструктор перейдет к формированию процедуры (рис. 11.11).

Ę	<u>–</u> Форма-Отчет. О	ICB					- D ×
l	朝				- D ×		<b>_</b>
l							
l							
l		Новая процедура				×	
l		Введите имя	Сформировать				
l		_		_	_		
l				<b></b>		-	
l				UK		Отмена	
l	<sup>-</sup>						
l	1						_
	Сформировать	Закрыть 🥊					
L.	СЭ Диалог	/ 🗐 Модуль	🔪 🔊 Таблица				

Рис. 11.11. Форма отчета конструктора с началом запроса

Далее выберем вид обращения к бухгалтерским итогам, согласно пояснениям конструктора (рис. 11.12).



Рис. 11.12. Форма конструктора с выбором схемы обращения к бухгалтерским итогам

Выбираем период формирования отчета с возможностью произвольной установки его в форме диалога, указываем возможность формирования отчета по любому счету, выбранному пользователем. На той же форме конструктора назначаем возможность выбора в форме диалога Вида субконто (рис. 11.13) и возможность выбора в форме диалога конкретного значения субконто.

По условиям нашего примера и в соответствии с настройками плана счетов в Конфигураторе у нас не ведется валютный учет и не используется разделитель учета (*см. разд. 10.2*), поэтому в конструкторе не доступны для настройки возможности отбора по валюте и разделителю учета.

В нашем примере мы используем один план счетов, поэтому не используем возможность назначения одного из возможных планов счетов в форме диалога.

Конструктор бухгалтерских з		
Введите ограничения получени		
Получение итогов по счетам Получать итоги по всем сче Выбирать счет в диалоге ф Получать итоги по указанны Укажите "Выбирать в диалоге Выберите "Получать итоги по с или нескольким конкретным с		
Ограничить получение итогов і Вид:		
ВыбВидСубконто1		
ВыбВидСубконто1		
Нажмите кнопку "Добавить", ч конкретным значениям субкон		

**Рис. 11.13.** Форма конструктора с выбором счета и выбором вида субконто

На следующем шаге работы конструктора укажем разрезы получения бухгалтерских итогов согласно условиям примера, а именно: в разрезе счетов по субсчетам и в разрезе выбранного в форме диалога субконто (рис. 11.14).

На следующем шаге указываем необходимость формирования итоговых строк и вывода данных по количеству. Закончив конструирование отчета, сохраните конфигурацию и, запустив программу в режиме "1С:Предприятие", проверьте правильность работы алгоритмов.

Для понимания производимой далее настройки отчета приведем модуль, построенный конструктором (листинг 11.12).

Конструктор бухгалтер	ских запросов: Оборотка		x
Укажите в разрезе как Строки Субконто Счет Убконто В Авнные по валю Данные по валю Данные по коли Итоговая строка	их объектов получать бухгалтерские итоги  В разрезе субсчетов  обор вида субконто Выбор в диалоге Выражение 1-ый вид субконто выбранного счета Конкретный вид субконто Контрагенты Материалы ОС Затраты	Добавить Удалить Вверх Вниз	
		Помощь	

Рис. 11.14. Форма конструктора с выбором отбора по счетам и субконто

<b>Σ</b> , OCB				[	- II X	
a a ? X						
	3 Квартал	2006 г.				
Период с: 01.07.06	по: 30.09.06					
Счет: ОСС.	]					
ВыбВидСубконто1: Основные средства ВыбСубконто1: Х						
Сформировать Зан	крыты					
📓 Сформировать 🔹						_ [
		Оборотно	о-сальдовая ве	домость		
		за	3 Квартал 2006	г.		
		Счет ОС	С, Основные с	редства		
Счет	Нач. ос	таток	Оборо	рт	Кон. ост	гаток
	Дебет	Кредит	Дебет 10,000,00	Кредит	Дебет 10.000.00	Кре
000			10,000.00		10,000.00	
Итоги			10,000.00		10,000.00	

Рис. 11.15. Форма отчета в системе "1С:Предприятие"

#### Листинг 11.12

//Данный фрагмент построен конструктором.

//При повторном использовании конструктора внесенные вручную изменения //будут потеряны!!!

Процедура Сформировать()

Таб = СоздатьОбъект ("Таблица");

Таб.ИсходнаяТаблица ("Сформировать");

Ит = СоздатьОбъект ("БухгалтерскиеИтоги");

Ит.ИспользоватьСубконто (ВыбВидСубконто1, ВыбСубконто1, 1, 1);

Ит.ВключатьСубсчета(1);

Ит.Выполнить Запрос (ВыбНачПериода, ВыбКонПериода, ВыбСчет,,,1,, "СК");

Таб. ВывестиСекцию ("Шапка");

Таб.Опции (0,0,Таб.ВысотаТаблицы(),0);

Ит.ВыбратьСчета();

Пока Ит.ПолучитьСчет() = 1 Цикл

Таб. ВывестиСекцию ("Счет");

Ит.ВыбратьСубконто (ВыбВидСубконто1);

Пока Ит.ПолучитьСубконто (ВыбВидСубконто1) = 1 Цикл

Таб. Вывести Секцию ("Субконто1");

КонецЦикла;

КонецЦикла;

Таб. ВывестиСекцию ("Подвал");

Таб.ТолькоПросмотр(1);

Таб.Показать ("Сформировать", "");

КонецПроцедуры

//}}БУХГАЛТЕРСКИЙ ЗАПРОС

Процедура ПриОткрытии()

//{{ИНИЦИАЛИЗАЦИЯ БУХГАЛТЕРСКОГО ЗАПРОСА(Сформировать) ВыбНачПериода = НачалоПериодаБИ(); ВыбКонПериода = КонецПериодаБИ(); //{{ИНИЦИАЛИЗАЦИЯ БУХГАЛТЕРСКОГО ЗАПРОСА

КонецПроцедуры

Следующее действие — специфическая настройка отчета. Для корректировки созданного в Конфигураторе отчета откройте окно Конфигурация и перейдите к ветви Отчеты дерева метаданных. Выберите в списке отчетов "ОСВ", дважды щелкните по нему и в открывшейся форме перейдите на вкладку Модуль.

При конструировании мы задали режим выдачи данных по количеству, но количественный учет в нашем примере установлен не для всех счетов. Изменим модуль отчета таким образом, чтобы информация о количестве выводилась только для счетов, на которых ведется количественный учет по аналитике.

Для этого выполним следующие действия:

- 1. В таблице с шаблоном печатной формы разделим секции Субконто1 на две строки: строка с суммой и строка с количеством для возможности управления выводом в отчет каждого вида итогов. В секциях Счет, Подвал оставим строки только с итогами по сумме, так как количественный учет для планов счетов у нас назначен только по аналитике и по счету в целом не имеет смысла (рис. 11.16).
- 2. Заменим строку в модуле отчета (листинг 11.13), добавив операторы проверки ведения количественного учета при выводе информации по счету и субконто (листинг 11.14).

🖪 Форма	-Отчет	.OCE	}					<u>- 0 ×</u>
		1	2		3	4	5	6 🔺
	1							
Шапка	2					Оборотно	-сальдовая в	едомость
	3				<"за "+П	ериодСтр(Вь	ыбНачПериод	а, ВыбКонГ
	4				<"	Счет "+ВыбС	чет+", "+Выб	ВидСубкон
	5		Счет		Нач. ос	статок	060	рот
	6				Пебет	Крелит	Пебет	Кредит
-	7				Секции - (Гор	изонтальные)	2	<u>ا</u> ا
Счет	8		<Ит.Счет>	Ит	Г. Шапка		Выбрать	<u>1т.KO()#Ч015.2</u>
Субконто1	10		<Ит.Представление	Ит	Счет			1т.KO()#4015.2
Субконто1	11		Кол-во	Ит	Субкон	ro1_C	Изменить	<u>1т.KO(3)#Ч015.:</u>
Подвал	12 13		Итоги	Ит	Субкон	то1_К	Удалить	Ит.KO()#Ч015.2
	14				E Hogody		Помошь	
	15			_				
	17							
	18							<b>_</b>
•							Закрыты	•
\Е́∃Диало	рг	$ \lambda $	🛄 Модуль	λ				

Рис. 11.16. Форма таблицы после изменения

#### 255

#### Листинг 11.13

```
Таб.ВывестиСекцию("Субконто1 ");
```

#### Листинг 11.14

```
Таб.ВывестиСекцию("Субконто1 _С");
Если Ит.Счет.Количественный=1 тогда
Таб.ВывестиСекцию("Субконто1 _К");
КонецЕсли;
```

3. Организуем вывод данных в отчете по списку всех значений субконто или по списку всех субсчетов счета в зависимости от установленного флага. Для этого в экранную форму диалога добавим элементы диалога типа "Флажок" под реквизит ВыбСчет и под реквизит ВыбВидСубконто1. Присвоим идентификаторы флажкам соответственно ВклСубсчета и ВклСубконто.

После всех изменений экранная форма будет выглядеть как на рис. 11.17.

💻 Форма-Отчет. ОСВ	
🛱 Оборотно-сальдовая ведомость	
<<ПериодСтр(ВыбНачІ	
Период с: ВыбНачПерио, 🔠 по: Выб	
Счет:	
ВыбСчет	
🗖 разворачивать по субсчетам	
Сформировать Закрыть	
Г¶ Диалог / ШМодуль	

Рис. 11.17. Экранная форма отчета с внесенными изменениями

В зависимости от включения режима разворачивания по субконто будет меняться возможность выбора значения субконто. Это условие опишем в про-

цедуре УстФлаг () (листинг 11.15), приписанной в формуле реквизита ВклСубконто на вкладке Дополнительно. Поясним процедуру. Поскольку реквизиту ВыбСубконто1 определен в конфигураторе тип "Неопределенный", в зависимости от выбора счета (ВыбСчет) и установки соответствующего ему вида субконто (ВыбВидСубконто1) мы устанавливаем соответствующий тип реквизиту Субконто1. Делаем это прямо в формулах реквизитов:

#### □ для реквизита **ВыбСчет** описываем формулы в одной строке последовательно

```
Назначить Счет (ВыбВидСубконто1, ВыбСчет, 1);
Форма.ВыбСубконто1.Назначить Тип (ВыбВидСубконто1);
```

#### **П** для реквизита **ВыбВидСубконто1** описываем формулу

```
Форма.ВыбСубконто1.НазначитьТип(ВыбВидСубконто1);
```

Включаем в процедуру операторы проверки режима формирования отчета по конкретному виду субконто или в режиме разворачивания по всем возможным значениям. Эти операторы смотрите в листинге 11.15 после комментария "проверка на выборку по субконто". При проверке режима вывода субконто в случае отключенного флага и пустым значением реквизита **ВыбСубконто1** сообщаем пользователям о необходимости выбора субконто и завершаем процедуру (Возврат).

После комментария "проверка на включение в выборку субсчетов" В листинге 11.15 в операторах, внесенных вручную, добавлены операторы проверки условия детализации итогов счета по субсчетам.

Далее в цикле (листинг 11.15) при выводе строк отчета проверяем режим детализации по субконто:

по выбранному значению;

🗖 разворачивать по всем возможным значениям.

Далее последовательно представлены процедуры модуля формы отчета согласно описанным комментариям.

#### Листинг 11.15

```
//{{БУХГАЛТЕРСКИЙ ЗАПРОС(Сформировать)
```

//Данный фрагмент построен конструктором.

//При повторном использовании конструктора внесенные вручную изменения //будут потеряны!!!

```
//{{ Схема номер 1
```

//{{ ПодробноеПредставлениеСубконто

Процедура Сформировать()

```
Таб = СоздатьОбъект ("Таблица");
Таб.ИсходнаяТаблица ("Сформировать");
Ит = СоздатьОбъект ("БухгалтерскиеИтоги");
// ** проверка на выборку по субконто
Если ВклСубконто=1 тогда
Ит.ИспользоватьСубконто (ВыбВидСубконто1, , 1, 1);
Иначе
Если ВыбСубконто1.Выбран()=1 Тогда
Ит.ИспользоватьСубконто (ВыбВидСубконто1, ВыбСубконто1, 2, 1);
Иначе
   Сообщить ("Укажите субконто!");
  Возврат;
КонецЕсли;
КонецЕсли;
// ** проверка на включение в выборку субсчетов
Если ВклСубсчета=1 тогда
Ит.ВключатьСубсчета(1);
КонецЕсли;
Ит.Выполнить Запрос (ВыбНачПериода, ВыбКонПериода,
    ВыбСчет,,, 1,, "СК");
Таб.ВывестиСекцию("Шапка");
Таб.Опции (0,0,Таб.ВысотаТаблицы(),0);
Ит.ВыбратьСчета();
Пока Ит.ПолучитьСчет() = 1 Цикл
Таб.ВывестиСекцию("Счет");
 Если ВклСубконто=1 тогла
   // если выбор по субконто
  Ит.ВыбратьСубконто();
   Пока Ит.ПолучитьСубконто() = 1 Цикл
    Таб.ВывестиСекцию ("Субконто1 С");
    Если Ит.Счет.Количественный=1 тогда
      Таб.ВывестиСекцию ("Субконто1 К");
    КонецЕсли;
   КонецЦикла;
```

Иначе // если разворачивать по субконто Ит.ВыбратьСубконто (ВыбСубконто1); Пока Ит.ПолучитьСубконто (ВыбСубконто1) = 1 Цикл Таб.ВывестиСекцию ("Субконто1 С"); Если Ит.Счет.Количественный=1 тогда Таб.ВывестиСекцию ("Субконто1 К"); КонецЕсли; КонецЦикла; КонецЕсли; КонецЦикла; Таб.ВывестиСекцию("Подвал"); Таб. ТолькоПросмотр (1); Таб.Показать ("Сформировать", ""); КонецПроцедуры //}}БУХГАЛТЕРСКИЙ ЗАПРОС Процедура ПриОткрытии() //{{ИНИЦИАЛИЗАЦИЯ БУХГАЛТЕРСКОГО ЗАПРОСА(Сформировать) ВыбНачПериода = НачалоПериодаБИ();

ВыбКонПериода = КонецПериодаБИ();

//{{ИНИЦИАЛИЗАЦИЯ БУХГАЛТЕРСКОГО ЗАПРОСА

КонецПроцедуры //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Процедура УстФлаг() Если ВклСубконто=0 Тогда Форма.ВыбСубконто1.НазначитьТип(ВыбВидСубконто1); ИначеЕсли ВклСубконто=1 Тогда Форма.ВыбСубконто1.НазначитьТип("Неопределенный"); КонецЕсли; КонецПроцедуры

На этом мы заканчиваем настройку отчета. Сохраните конфигурацию и проверьте настройки в режиме "1С:Предприятие".



# Оперативный (торговый) учет

Из всего дерева структуры метаданных компоненте "Оперативный учет" принадлежит только один вид объектов — регистры. Этот вид объектов будет присутствовать в дереве только тогда, когда компонента загружена (см. елаву 1).

Наиболее важным преимуществом использования регистров является то, что они служат для хранения информации, к которой необходимо получать оперативный доступ. Уровень оперативности и, соответственно, целесообразности использования регистров должен определять специалист, выполняющий конфигурирование системы в соответствии с требованиями пользователей.

Кроме того, необходимо отметить, что регистры позволяют получать наиболее достоверную информацию о состоянии оперативного учета. Для работы в реальном времени система поддерживает точку актуальности итогов. Она может быть установлена принудительно, но ее могут изменять проводимые в потоке документы.

К особенностям использования регистров относится невозможность внесения информации напрямую непосредственно в регистры.

### 12.1. Регистры оперативного учета

Регистры используются для накопления информации о наличии и движении средств (товарных, денежных и т. п.). Именно регистры являются основным источником информации для анализа и формирования отчетов оперативно-го учета.

Регистры являются внутренним компонентом конфигурации, т. е. они недоступны для непосредственного заполнения, нет средств их просмотра. Только средствами встроенного языка можно записывать в регистры информацию и затем извлекать ее.

Можно сказать, что это N-мерная система координат, в узлах которой хранятся итоговые данные на определенный момент времени, причем доступ к данным осуществляется практически мгновенно.

Основные элементы регистров:

- измерения оси координат, на пересечении конкретных значений которых лежит информации;
- □ *ресурсы* собственно информация в виде итоговых данных.

Измерения можно определить как набор признаков, в разрезе которых информация должна храниться, извлекаться, анализироваться. Ресурсы — категории учета, которые должны быть представлены в числовом виде.

В системе "1С:Предприятие" существуют 2 типа регистров: регистры остатков и регистры оборотов. Различаются они характером хранимой информации: в регистрах остатков всегда хранится информация о конечном состоянии средств, а в регистрах оборотов — изменение этих средств.

Под термином "остаток" для регистра остатков следует понимать числовое значение, которое имеет ресурс на какой-то момент времени.

Кроме хранения остатка или оборота ресурса, регистры хранят все изменения (приход или расход), произведенные документами, т. е. сами движения регистра. Причем эти движения регистров хранятся со ссылками на те документы, которые их вызвали.

Отсюда и хранение регистров физически организовано в 2-х таблицах:

- 🗖 таблица остатков;
- 🗖 таблица движений.

Во встроенном языке системы "1С:Предприятие" есть методы, позволяющие получить из регистра все его движения, указав интересующий интервал времени. Изменение информации в регистрах происходит при выполнении системных предопределенных процедур в модуле документа: ОбработкаПроведения () И ОбработкаУдаленияПроведения ().

Помимо измерений и ресурсов, для регистра может быть создан набор реквизитов (рис. 12.1). Реквизиты позволяют включать в движения регистров различную дополнительную информацию. Реквизиты не влияют на значения ресурсов регистра и могут использоваться для анализа движений регистра при их выборке.

При работе с регистрами следует помнить, что *регистры остатков* хранят значения своих ресурсов на некоторый момент времени. Этот момент называется *точкой актуальности* итогов, обозначаемой как ТА. Документы, от-

носящиеся к компоненте "Оперативный учет", при проведении записывают в соответствующий регистр необходимые движения и затем уже передвигают ТА вперед по времени.

🗰 Регистр Новый1	
<u>И</u> дентификатор: Новый1	
 Комментарий:	
Измерения	
новый1	
Новый Изменить У	
Реквизиты	
Новый Изменить 9	

Рис. 12.1. Структура регистра — измерения, ресурсы, реквизиты

Работа регистра остатков в системе организована таким образом, что при обращении к состоянию ресурсов, т. е. остаткам, регистр практически мгновенно возвращает остаток на ТА.

### 12.2. Применение компоненты "Оперативный учет"

Напомним, что описанные примеры показывают подходы к конфигурированию в компоненте, а не описывают методику ведения учета.

Для работы запустите учебную конфигурацию в режиме **Конфигуратор** и откройте окно **Конфигурация**.

### 12.2.1. Создание объектов в компоненте "Оперативный учет"

Для настройки оперативного учета в учебной конфигурации нам так же, как и для бухгалтерского учета, необходимо будет дополнить ее новыми объектами, а также существующие объекты дополнить новыми реквизитами и алгоритмами.

Сначала нужно создать одну новую константу **Метод списания**, как описано в *главе 5*. Тип значения — "Строка", длина 10 символов.

Используем возможности оперативного учета для отражения взаиморасчетов с контрагентами в разрезе договоров и расчета себестоимости по выбранному методу списания. Для этого создадим регистры оперативного учета:

**Взаиморасчеты** — для учета взаиморасчетов с контрагентами;

**Партии** — для ведения учета партий товаров.

При создании регистра важно вначале для себя дать ответ на два основных вопроса:

- Какую информацию необходимо накапливать в регистре, и в каких разрезах требуется ее извлекать?
- □ Какие документы и какие движения ("приход" или "расход") этих документов должны изменять состояние регистров?

Разберем структуру регистра. При рассмотрении сквозного примера нам необходима информация в виде отчета по состоянию взаиморасчетов с контрагентами (остатки по взаиморасчетам), показывающего, сколько долгов по каждому контрагенту имеется в расчете на ТА.

Исходя из этого, создадим регистр Взаиморасчеты. При создании регистра используем основной метод в меню Действия | Новый элемент.

Регистр Взаиморасчеты будет иметь следующую структуру (рис. 12.2).

В рамках примера нам необходима информация для анализа прибыльности партий и возможность расчета себестоимости в зависимости от выбранного метода списания: FIFO, LIFO, по среднему.

Исходя из этой задачи, мы создадим регистр Партии для хранения информации о движении партий.

Регистр Партии будет иметь следующую структуру (рис. 12.3).

Теперь обсудим движения регистра. Регистр накапливает итоги и не хранит записей о тех значениях измерений, по которым нет ресурсов, т. е. регистр помнит только то, что необходимо. При этом обеспечивается компактная структура хранения информации, позволяющая быстро найти то, что нужно.

Ħ	Регистр Взаи	морасчеты
Γ	<u>И</u> дентификатор	р: Взаимора
	<u>К</u> омментарий:	Отражени
	Измерения	
	Контрагент Основание	
	Новый	Изменить
	Реквизиты	
	Новый	Изменить

Рис. 12.2. Окно Регистр Взаиморасчеты

Поэтому таблица остатков регистра при правильном построении конфигурации практически не растет и — как результат — документы проводятся быстро и отчеты формируются быстро.

При неправильном проектировании регистров и документов таблица остатков растет очень быстро и проведение документов и формирование отчетов замедляются.

#### Примечание

Чтобы не происходило того, что описано в предыдущем абзаце, нужно при проектировании регистра соблюдать следующее правило:

"Если для некоторого регистра остатков есть вид документа, записывающий движение «Приход», то обязательно должен быть вид документа, записывающий движение «Расход»".

Если это правило не выполняется хотя бы для одного регистра конфигурации, то об оперативном учете через некоторое время можно забыть. Это произойдет, потому что регистр будет постоянно накапливать итоги по всем возможным комбинациям измерений. Вследствие этого вырастет время, за которое регистр будет выдавать остаток.

🗰 Регистр Партии	
Идентификатор: Партии	
<u>К</u> омментарий: Партии	
Измерения	
Партия Номенклатура Поставщик	
Новый Изменить 5	
Новый Изменить L	

Рис. 12.3. Окно Регистр Партии

Представим в виде таблицы для каждого регистра примера возможные виды документов и их движения (табл. 12.1).

······································	Таблица	12.1.	Виды	док	ументов	в и их	движения
--	---------	-------	------	-----	---------	--------	----------

Регистр "Взаиморасчеты"				
Флаг движения	Вид документа			
"-" — расход	Поступление товаров			
"-" — расход	Поступление материалов			
"-" — расход	Поступление ОС			
"+" — приход	Реализация товаров			

#### Таблица 12.1 (окончание)

Регистр "Взаиморасчеты"					
Флаг движения	Вид документа				
"-" — расход	ПКО				
"+" — приход	РКО				
"-","+" — расход, приход	Выписка банка				
Регистр "Партии"					
"+" — приход	Поступление товаров				
"-" — расход	Реализация товаров				

При определении флага движения по регистру взаиморасчетов мы считаем, что:

□ *приход* — увеличение долга контрагента перед нашей организацией;

□ *расход* — уменьшение долга контрагента перед нашей организацией.

Дополним существующие документы конфигурации. В документы "РКО", "ПКО" в шапку добавляем реквизит **Договор** для ведения взаиморасчетов по договорам в регистре **Взаиморасчеты**. При этом необходимо в модуле формы описать условия заполнения нового реквизита для определенных счетов, имеющих аналитику **Контрагенты**.

С учетом этих условий форма документов будет выглядеть так (на примере "ПКО") — рис. 12.4.



Рис. 12.4. Форма документа "ПКО" со свойствами реквизита

Программный код модуля формы, дополненный операторами условия заполнения реквизита, выглядит следующим образом (листинг 12.1).

#### Листинг 12.1

```
Процедура ПриОткрытии()
  ПриЗаписиПерепроводить (1);
  // для реквизита Договор
  Если (Сч1=СчетПоКоду ("РАС.1", ПланыСчетов. Основной))
  или (Сч1=СчетПоКоду ("РАС.2", ПланыСчетов. Основной)) Тогда
   Форма.Договор.Доступность (1);
  Иначе
   Форма.Договор.Доступность (0);
  КонецЕсли;
КонецПроцедуры
//*********
Процедура ПриВыбореСубконто1()
  Если ТипЗначенияСтр(Субконто1) = "Справочник" Тогда
   Если Субконто1.Вид() = "Сотрудники" Тогда
     ПринятоОт = Субконто1.Наименование;
   ИначеЕсли Субконто1.Вид() = "Контрагенты" Тогда
     // для реквизита Договор
     Форма.Договор.Доступность (1);
   Если ПустоеЗначение (СокрЛП (Субконто1.Наименование)) = 0 Тогда
      ПринятоОт = СокрЛП (Субконто1. Наименование);
   КонецЕсли;
   КонецЕсли;
  КонецЕсли;
  КонецПроцедуры
```

Для документа "Выписка" добавляем реквизит **Договор** в табличную часть. Модуль формы документа при этом будет выглядеть следующим образом (листинг 12.2).

Листинг 12.2	1
--------------	---

Процедура ПриОткрытии()

```
ПриЗаписиПерепроводить (1);
```

```
// для реквизита Договор
  ВыбратьСтроки();
  Пока получитьСтроку()=1 Цикл
  Если (КорСчет=СчетПоКоду ("РАС.1", ПланыСчетов. Основной))
  или (КорСчет=СчетПоКоду ("РАС.2", ПланыСчетов. Основной)) Тогда
     Форма.Договор.Доступность (1);
  Иначе
     Форма.Договор.Доступность (0);
   КонецЕсли;
  КонецЦикла;
КонецПроцедуры
//*********
Процедура ПриВыбореСубконто1()
  // вызывается при выборе значения реквизита КорСчет
  Если ТипЗначенияСтр(Субконто1) = "Справочник" Тогда
   Если Субконто1.Вид() = "Сотрудники" Тогда
     ПринятоОт = Субконто1.Наименование;
  ИначеЕсли Субконто1.Вид() = "Контрагенты" Тогда
  // для реквизита Договор
     Форма.Договор.Доступность (1);
   Если ПустоеЗначение (СокрЛП (Субконто1.Наименование)) = 0 Тогда
      ПринятоОт = СокрЛП (Субконто1.Наименование);
   КонецЕсли;
   КонецЕсли:
  КонецЕсли;
КонецПроцедуры
```

Для документов "Поступление материалов", "Поступление ОС" добавим новый реквизит шапки **Основание** для учета взаиморасчетов в регистре **Взаиморасчеты**. После внесения реквизита экранная форма документов (на примере одного из них) будет выглядеть так (рис. 12.5).

Теперь реквизитный состав документов и алгоритмы записи движений регистров соответствуют друг другу.

Сохраним внесенные в конфигурацию изменения.

јПо	ступлен	ие материал	10B	
			Пос	тупл
Контр	агент:	Контрагент		
Осна	ование:	Основание		
N	ТМЦ			K
	OK	Закрыт	ь Ka	оммент

Рис. 12.5. Форма документа "Поступление материалов" со свойствами реквизита

### 12.2.2. Алгоритмы объектов

Прежде всего, перед дополнением алгоритмов документов, уже созданных в конфигурации, необходимо дополнить их характеристики для отражения их участия в оперативном учете. Для документов "Поступление товаров", "Реализация товаров", "Поступление материалов", "Поступление ОС", "ПКО", "РКО", "Выписка" определяем принадлежность к компоненте "Оперативный учет", т. е. документ будет не только записываться, но и создавать движения регистров при проведении.

Для документов "Поступление товаров", "Поступление материалов", "Поступление ОС", "ПКО", регистрирующих движения в регистре **Взаиморасчеты** с флажком **Приход**, вносим изменения в алгоритм проведения документов (модуль документа). В качестве примера используем документ "Поступление товаров".

Для прописывания движений по взаиморасчетам в модуле проведения к операторам формирования бухгалтерских проводок *(см. главу 10)* дополним операторы записи движений в регистр.

Для этого откроем модуль документа, установим курсор в то место, где необходимо добавить операторы, и выберем меню Конструкторы, пункт Движение регистров....

В предложенном окне выберем регистр и вид движения — расход, т. е. уменьшение долга контрагента перед нами.

🖹 Документ РК	(0		
<u>И</u> дентификатор:	РКО		
<u>К</u> омментарий:	Расходны	й касс	
Реквизиты ша	пки		
Сч1 Субконто1 Сумма Основание Выдать Договор			
Новый И	1зменить	Удал	
Номер			
<u>Н</u> умератор:	<< He	назнач	
<u>П</u> ериодичности	ы Повсе	:м дан	
🔽 Автоматич	еская нуме	рация	
🔽 Разрешить	проведени	ю доку	
🔽 Автоматическое удаление д			
🔽 Автоматическая нумерация			
Создавать опера	ацию: Всег	да	
<u>В</u> вод на основа	нии		

Рис. 12.6. Окно редактирования документа с установленным флажком Оперативный учет

Движения формируются по аналогии с конструктором формирования бухгалтерских операций (см. главу 10).

После работы конструктора в модуле документа появляются следующие строки перед операторами формирования бухгалтерских операций (можно и после них) (листинг 12.3).

#### Листинг 12.3

//}}ДВИЖЕНИЯ\_РЕГИСТРОВ

//Данный фрагмент построен конструктором.

//При повторном использовании конструктора внесенные вручную

```
//изменения будут потеряны!!!
```

```
Регистр.Взаиморасчеты.Контрагент = Контрагент;
Регистр.Взаиморасчеты.Основание = Основание;
Регистр.Взаиморасчеты.Сумма = Итог("Сумма");
Регистр.Взаиморасчеты.ДвижениеРасходВыполнить();
//}}движения РЕГИСТРОВ
```

Аналогично с помощью конструктора или методом копирования части модуля вносим операторы по отражению в регистрах движений в документы "Поступление материалов", "Поступление ОС".

Для документа "ПКО" добавить условные операторы для проверки выбранного счета, поскольку взаиморасчеты в регистрах будут изменяться только при выборе определенных счетов (листинг 12.4).

#### Листинг 12.4

```
Если (Сч1=СчетПоКоду("РАС.1", ПланыСчетов.Основной))
или (Сч1=СчетПоКоду("РАС.2", ПланыСчетов.Основной)) Тогда
Регистр.Взаиморасчеты.Контрагент = Субконто1;
Регистр.Взаиморасчеты.Основание = Договор;
Регистр.Взаиморасчеты.Сумма = Сумма;
Регистр.Взаиморасчеты.ДвижениеРасходВыполнить();
КонецЕсли;
```

Для документа "Поступление товаров" дополним движения по регистру **Партии**.

Для этого в уже сформированном модуле документа с движениями по регистру Взаиморасчеты вызовем конструктор через пункт меню Конструкторы | Движения регистров....

В открывшемся диалоговом окне нажатием кнопки **Новое** вызовем окно **Новое движение** для выбора регистра **Партии** и вида движения по нему.

Далее описываем движения по регистру с указанием режимов Движение по каждой строке документа, Привязывать строку.

После закрытия конструктора в модуле документа движения по регистрам вносим дополнения: в зависимости от заданного в константе **Метод списа**ния значения устанавливаем значение партиеобразующего документа. Модуль теперь будет выглядеть следующим образом (листинг 12.5).

#### Листинг 12.5

```
//Вынесли из цикла
  Если Константа. МетодСписания = Перечисление. МетодСписания. ПоСреднему
Тогла
   Регистр.Партии.Партия = "";
   Регистр. Партии. Поставщик = "";
  Иначе
   Регистр.Партии.Партия = ТекущийДокумент();
   Регистр.Партии.Поставщик = Контрагент;
  КонецЕсли;
// цикл по каждой строке
  ВыбратьСтроки();
  Пока ПолучитьСтроку() = 1 Цикл
   Регистр.Партии.Номенклатура = ТМЦ;
   Регистр.Партии.Сумма = Сумма;
   Регистр.Партии.Количество = Количество;
   Регистр. Партии. Привязывать Строку (НомерСтроки);
   Регистр.Партии.ДвижениеПриходВыполнить ();
  КонецЦикла;
  Регистр.Взаиморасчеты.Контрагент = Контрагент;
  Регистр.Взаиморасчеты.Основание = Основание;
```

Регистр.Взаиморасчеты.Сумма = Итог("Сумма");

```
Регистр.Взаиморасчеты.ДвижениеРасходВыполнить ();
```

Документы "Реализация товаров", "РКО" будут отражать приход по регистру Взаиморасчеты, т. е. увеличение долга клиента.

По аналогии с описанными документами добавим операторы движения по регистру взаиморасчетов, но уже с флажком **Приход** (листинг 12.6).

#### Листинг 12.6

- //}}ДВИЖЕНИЯ\_РЕГИСТРОВ
- //Данный фрагмент построен конструктором.
- //При повторном использовании конструктора внесенные вручную
- //изменения будут потеряны!!!

```
Регистр.Взаиморасчеты.Контрагент = Контрагент;
Регистр.Взаиморасчеты.Основание = Основание;
Регистр.Взаиморасчеты.Сумма = Итог("Сумма");
Регистр.Взаиморасчеты.ДвижениеПриходВыполнить();
//}}движения РЕГИСТРОВ
```

Для документа "РКО" добавить условные операторы для проверки выбранного счета по аналоги с документом "ПКО".

Для документа "Выписка" добавить условные операторы для проверки выбранного счета. В зависимости от выбранного счета и суммы прихода или расхода денежных средств записываем в регистр Взаиморасчеты движения с соответствующими флагами.

Модуль документа будет дополнен следующими операторами до формирования бухгалтерской операции (листинг 12.7).

#### Листинг 12.7

```
Регистр.Взаиморасчеты.Контрагент = Субконтоl;
Регистр.Взаиморасчеты.Основание = Договор;
Если (КорСчет=СчетПоКоду("РАС.1",ПланыСчетов.Основной)) Тогда
// взаиморасчеты с поставщиками
Регистр.Взаиморасчеты.Сумма = Расход;
Регистр.Взаиморасчеты.ДвижениеПриходВыполнить();
ИначеЕсли (КорСчет=СчетПоКоду("РАС.2",ПланыСчетов.Основной)) Тогда
// взаиморасчеты с покупателями
Регистр.Взаиморасчеты.Сумма = Приход;
Регистр.Взаиморасчеты.ДвижениеРасходВыполнить();
КонецЕсли;
```

При этом мы рассматриваем идеальный случай при отсутствии возвратов денежных средств. Усложнить задачу можно самостоятельно.

Сохраним внесенные в конфигурацию изменения.

### 12.2.3. Особенности проведения документов

В *главе 11* рассматривалось проведение документов в реальном времени, при этом данные из регистров вообще не использовались. В реальной работе

часто возникают ситуации, когда необходимо получить информацию о данных в регистрах не на ТА, а на конец дня, начало дня, на момент проведения конкретного документа.

Например, расчет себестоимости реализованного товара производится на основании данных регистра **Партии** в зависимости от метода списания и момента проведения документа реализации.

В этом случае нужно провести так называемый временный расчет одного или нескольких регистров, в которых хранится необходимая информация.

Для проведения временного расчета используется метод Временный Расчет(), который устанавливает флаг участия регистра во временном расчете. После этого можно выполнять расчеты с регистрами: Рассчитать РегистрыНа() и Рассчитать РегистрыПо(). Только после этого методы обращения к итогам регистра будут выдавать значения на заданный момент времени.

Для каждого регистра, участвующего во временном расчете, после проведения временного расчета формируется "фотография" состояния регистра. На эту "фотографию" не будут влиять все последующие изменения регистра.

Рассмотрим системные процедуры работы с регистрами. При применении процедуры в тексте программного модуля предварительная ссылка на объект не нужна.

Процедура Рассчитать РегистрыНа () рассчитывает все регистры, у которых установлен флаг участия во временном расчете на момент начала события, используемого в качестве параметра процедуры. Если в качестве параметра передана дата, то расчет временный на начало указанной даты, если параметр — ссылка на документ, то расчет производится на момент до проведения документа.

Процедура Рассчитать РегистрыПо() рассчитывает регистры соответственно на момент окончания события, переданного в качестве параметра процедуры.

Для документа "Реализация товара" мы добавим временный расчет регистра Партии для расчета себестоимости.

Учитывая сказанное, модуль документа можно начать следующими строками кода (листинг 12.8).

#### Листинг 12.8

```
// использование временного расчета
```

```
// запрещено дублирование товара в табличной части
```

```
Рег=СоздатьОбъект ("Регистр.Партии");
```

Если ИтогиАктуальны()=0 тогда // проведение задним числом

```
Рег.Временныйрасчет(1);
```

```
Рассчитать РегистрыНа (ТекущийДокумент ());
```

КонецЕсли;

Использование методов Рассчитать РегистрыНа () И Рассчитать РегистрыПо () позволяет иметь в системе *только один временный расчет.* 

Параллельное выполнение в системе временных расчетов одних и тех же регистров возможно при использовании вспомогательных объектов типа "Регистры". При этом набор регистров, входящих в состав одного объекта **Регистры**, воспринимается системой независимо от набора регистров, входящих в состав другого объекта **Регистры**.

Доступ к атрибутам и методам объекта типа "Регистры" может выполняться с помощью переменной со ссылкой на объект, создаваемый методом Создать Объект (). При создании объекта типа "Регистры" используется ключевое слово Регистры:

СпецРег=СоздатьОбъект("Регистры");

Для созданного таким образом объекта обращение к атрибутам и методам производится добавлением названия соответствующего метода или атрибута к имени объекта через точку.

Важным методом объекта "Регистры" является метод Актуальность (), позволяющий поддерживать итоги временного расчета регистров в актуальном состоянии все время проведения документа. При этом движения регистра будут изменять итоги временного расчета. Использование этого метода позволяет проводить документ с повторяющимися товарами в строках табличной части.

Учитывая эти дополнения, модуль документа "Реализация товаров" для расчета себестоимости по средней может выглядеть так (листинг 12.9).

Пистинг 12.9						
Перем ОбщРег, Рег;						
// **********						
Функция РассчитатьСебестоимость (Товар)						
//здесь следует написать алгоритм расчета себестоимости товаров						
Если Константа.МетодСписания =						
Перечисление.МетодСписания.ПоСреднему Тогда						
ОстСум=Рег.СводныйОстаток(,ТМЦ,,"Сумма");						

274

```
Кол=Рег.СводныйОстаток (, ТМЦ, , "Количество");
   Регистр.Партии.Партия = "";
   Регистр.Партии.Поставщик = "";
   Регистр.Партии.Сумма = Окр (ОстСум*Количество/Кол, 2);
   Регистр.Партии.Продажа = Сумма;
   Регистр. Партии. Количество = Кол;
   Регистр. Партии. Привязывать Строку (НомерСтроки);
   Регистр. Партии. Движение Приход Выполнить ();
  Иначе
   // рассчитать себестоимость по партиям
  КонецЕсли;
  Возврат ОстСум;
КонецФункции
//***********
11
Процедура ОбработкаПроведения ()
ОбщРег=СоздатьОбъект ("Регистры");
// поддержка итогов в актуальном состоянии
ОбщРег.Актуальность (1);
Рег=ОбщРег.Партии;
Если ИтогиАктуальны()=0 тогда // проведение задним числом
  Рег.Временныйрасчет(1);
  ОбщРег. Рассчитать РегистрыНа (ТекущийДокумент ());
КонецЕсли;
ВыбратьСтроки();
Пока ПолучитьСтроку() = 1 Цикл
  Регистр.Партии.Номенклатура = ТМЦ;
  // определим остаток
  ОстКол=Рег.СводныйОстаток(, ТМЦ,, "Количество");
  Если ОстКол > О Тогла
   Если ОстКол < Количество Тогла
     Сообщить ("Не хватило товара "+ТМЦ);
     СтатусВозврата(0);
```

```
Продолжить;
   КонецЕсли;
   СуммаСеб=РассчитатьСебестоимость (ТМЦ);
   Если СуммаСеб=0 Тогда
     Сообщить ("Не рассчитана себестоимость "+ТМЦ);
     СтатусВозврата(0);
     Продолжить;
   КонецЕсли;
   КонецЕсли;
КонецЦикла;
         // учет взаиморасчетов
  Регистр.Взаиморасчеты.Контрагент = Контрагент;
  Регистр.Взаиморасчеты.Основание = Основание;
  Регистр.Взаиморасчеты.Сумма = Итог ("Сумма");
  Регистр.Взаиморасчеты.ДвижениеПриходВыполнить ();
  //}}ДВИЖЕНИЯ РЕГИСТРОВ
```

```
//{{ФОРМИРОВАНИЕ ОПЕРАЦИИ
```

Далее следуют операторы формирования операции. Оператор формирования суммы списания себестоимости изменяем на следующий:

Операция.Сумма = СуммаСеб;

### 12.2.4. Последовательность документов

Важной особенностью компоненты "Оперативный учет" является понятие *точки актуальности итогов* (ТА).

ТА — это соответствующий документации момент времени (дата + время), на который все необоротные регистры содержат текущие значения своих ресурсов. В общем случае в качестве ТА берется дата и время самого "старшего" проведенного документа.

Благодаря наличию характеристик в виде даты и времени все документы в системе образуют хронологическую последовательность.

Алгоритмы проведения документа, как правило, отражают в учете данные, записанные в самом документе. В некоторых случаях алгоритмы проведения документа анализируют текущие итоги, используя их при проведении. На-

пример, при реализации товаров анализируется состояние по остаткам. Поэтому в этом случае важно, чтобы документы проводились в системе строго последовательно.

Однако на практике по ряду причин часто приходится вводить или исправлять документы задним числом. Разумеется, в этом случае движения регистров документов, расположенных после "нарушителя", становятся некорректными (неактуальными). Поэтому для актуализации состояния регистров все документы, анализирующие остатки и расположенные после "нарушителя", должны быть перепроведены.

Для облегчения этой работы в версии 7.7 системы "1С:Предприятие" для компонент "Оперативный учет" и "Бухгалтерский учет" был введен вспомогательный вид объекта метаданных Последовательности документов (рис. 12.7).



Рис. 12.7. Окно Последовательность объекта метаданных Последовательности документов

Каждая введенная в структуру метаданных последовательность документов обеспечивает контроль последовательного проведения документов указанных видов. Для настройки данного контроля необходимо указать в окне настройки последовательности документов следующие параметры (рис. 12.7):

**Виды** документов, которые входят в данную последовательность;

🗖 виды движений, влияющие на последовательность.

Виды движений определяют, какие из изменений итогов будут влиять на необходимость перепроведения документов последовательности, т. е. итоги каких механизмов учета используются документами данной последовательности при перепроведении.

Для нашего примера создадим последовательность **Товары**. Для организации правильного списания товаров укажем в качестве движения, влияющего на последовательность, регистр оперативного учета, в котором ведется учет товаров по стоимости (рис. 12.8).



Рис. 12.8. Движения, влияющие на последовательность

В качестве документов, на проведение которых будет влиять данная последовательность, укажем документ "Реализация товаров", при проведении которого анализируется указанный регистр (рис. 12.9).

Для каждой созданной последовательности система поддерживает автоматически *границу последовательности*. В качестве границы последовательности будет выступать позиция документа.

При последовательном проведении документов, входящих в последовательность, граница последовательности будет устанавливаться на каждый вновь проведенный документ.



**Рис. 12.9.** Документы, зависящие от последовательности

Проведение документов	
Документы Последовательное	
Восстановить последователь	
О Точку актуальности	
Последовательности докумен	
Выбрать все	
🗵 Выводить сообщения о пере	

Рис. 12.10. Восстановление последовательности документов

При проведении документов задним числом, а также при отмене проведения или удалении документов, если удаляются или записываются движения

регистров, указанных как влияющие на данную последовательность, *граница последовательности* отодвигается на момент измененного документа.

В режиме перепроведения документов существует специальная возможность восстановления последовательности проведения документов (рис. 12.10).

При использовании режима восстановления последовательности система автоматически выполняет перепроведение всех документов, входящих в выбранную последовательность от текущей позиции границы последовательности до указанного момента.

Восстановление последовательности может выполняться только в монопольном режиме, так как система должна гарантировать неизменность итогов на момент перепроведения документов.

Средства языка также предоставляют возможность непосредственного доступа к методам последовательностей. Все объявленные в Конфигураторе последовательности принадлежат к глобальному контексту конфигурации. Доступ к методам возможен в любом программном модуле. Во всех текстах программных модулей вызовы методов последовательностей можно записывать через точку после имени последовательности.

В рамках рассматриваемого примера мы не используем методы работы с последовательностями. Ознакомиться с ними можно в документации.



## Запросы и отчетность

Основная отличительная черта запросов — возможность создания выборки данных с произвольными группировками. При этом важной причиной использования механизма запросов при программировании модулей системы "1С:Предприятие" является их работа при обработке хранящихся данных, более быстрая, чем работа встроенных методов регистров.

### 13.1. Понятие запросов

Доступ к атрибутам и методам запросов может выполняться только через переменную, созданную методом СоздатьОбъект().

При написании запросов следует придерживаться следующей последовательности действий.

- 1. Создание объекта типа "Запрос".
- 2. Создание переменной с текстом запроса.
- 3. Выполнение запроса методом Выполнить ().
- 4. Организация циклической обработки результатов запроса.

Рассмотрим пример создания запроса для алгоритма сквозного примера.

Как вы помните, в модуле документа "Реализация товара" мы не выбрали вариант расчета себестоимости в случае выбираемого метода списания FIFO или LIFO. Для этих вариантов расчета необходимо создать выборку из остатков товаров в разрезе приходных партий. Воспользуемся запросом.

Дополним алгоритм модуля проведения документа "Реализация товара" *(см. разд. 12.3.3)*.

Объявим общую переменную Запрос перед процедурами.
Перед циклической обработкой табличной части документа (см. листинг 12.9) создадим запрос, к результатам которого будем обращаться в цикле для каждого выбираемого товара. Поскольку расчет себестоимости по средней мы уже запрограммировали, то запрос будем строить для расчета себестоимости по методу списания FIFO или LIFO (листинг 13.1).

#### Листинг 13.1

Если Константа.МетодСписания <> Перечисление.МетодСписания.ПоСреднему Тогда

// Выполним запрос для списания по партиям

КонецЕсли;

После строки комментария внутри группы условных операторов начнем описывать запрос.

Создадим объект типа "Запрос" и определим точку на оси времени, для которой будут выбираться итоги. Этой точкой является позиция текущего документа (листинг 13.2).

### Листинг 13.2

// Определим точку для выборки ТекДок=ТекущийДокумент(); // создадим объект типа Запрос Запрос=СоздатьОбъект("Запрос");

Следующий шаг — формирование текста запроса в идее переменной (листинг 13.3).

#### Листинг 13.3

// формируем текст запроса ТекстЗапроса= "//{{ЗАПРОС(Сформировать)"; // проверим, в реальном ли времени выполняется запрос //(актуальность) Если ИтогиАктуальны()=0 тогда // проведение задним числом // тогда укажем период запроса – документ ТекстЗапроса=ТекстЗапроса + " |Период с ТекДок по ТекДок;"; КонецЕсли; ТекстЗапроса=ТекстЗапроса + "

# Далее формируем текст запроса, включая комментарии (листинг 13.4).

### Листинг 13.4

```
// включаем партии
|Партия = Регистр.Партии.Партия;
// включаем товары
|Номенклатура = Регистр.Партии.Номенклатура;
// включаем значения сумм
|Сумма = Регистр.Партии.Сумма;
// включаем значения количества
Количество = Регистр.Партии.Количество;
// добавляем функции для получения конечных остатков
|Функция СуммаКонОст = КонОст (Сумма);
|Функция КоличествоКонОст = КонОст (Количество);
// указываем порядок группировки выборки
Группировка Номенклатура упорядочить по Номенклатура. Наименование без
групп;
|Группировка Партия;
// добавляем условие для ограничения выборки
|Условие (Номенклатура = ТМЦ);
|"//}}ЗАПРОС
```

```
;
```

Как видим, текст запроса нам уже понятен, так как аналогичные тексты мы получали, пользуясь конструкторами запросов.

Далее выполняем запрос и, если запрос не выполняется, процедура проведения документа прекращает свою работу (листинг 13.5).

#### Листинг 13.5

```
// выполняем запрос, и если не выполнен - прекращаем процедуру
Если Запрос.Выполнить (ТекстЗапроса)=0 Тогда
СтатусВозврата(0);
Возврат;
КонецЕсли:
```

Остальные операторы модуля проведения остаются без изменения.

Таким образом, мы создадим выборку оперативных итогов перед обработкой табличной части документа. Обрабатывая каждую строку табличной части нам необходимо рассчитать себестоимость товара выбранной строки. Для этих целей вызывается процедура РассчитатьСебестоимость (ТМЦ).

Теперь в процедуру (см. листинг 12.9) впишем расчет себестоимости списываемой партии товара.

Сначала в зависимости от метода списания определяем порядок просмотра партий: прямой порядок (FIFO) или обратный (LIFO) (листинг 13.6).

Листинг 13.6

```
// рассчитать себестоимость по партиям
// в зависимости от метода списания - порядок обработки выборки
Если Константа.МетодСписания =
Перечисление.МетодСписания.FIFO Тогда
Hanp=1;
Иначе
Hanp=-1;
КонецЕсли;
```

Используем выборку, созданную запросом. Переходим в начало выборки (листинг 13.7).

Листинг 13.7

// для выборки запроса позиционируемся на Товаре строки Запрос.Группировка ("Номенклатура");

Позиционируемся на группе элементов для товара строки, значение которого передается в процедуру в виде параметра (листинг 13.8).

### Листинг 13.8

Запрос.Получить (Товар,);

Если в выборку попал нужный нам товар, тогда получаем значения из переменных запроса для суммового остатка (Запрос.СуммаКонОст;) и для количественного остатка (Запрос.КоличествоКонОст) (листинг 13.9).

```
Запросы и отчетность
```

### Листинг 13.9

```
Если Запрос.Получить (Товар,)=1 Тогда

//нашли товар в выборке

ОстСум=Запрос.СуммаКонОст;

Кол=Запрос.КоличествоКонОст;

Иначе

// не нашли товара

ОстСум=0;

Кол=0;

КонецЕсли;
```

Запоминаем количество товара, необходимое для списания, поскольку может случиться, что списывать будем из нескольких партий (листинг 13.10).

Листинг 13.10

ОсталосьСписать=Количество;

Далее организовываем циклическое обращение к партиям выборки запроса (это второй уровень группировки запроса — см. листинг 13.4). В условиях выполнения цикла указываем следующие (листинг 13.11):

по всем записям выборки в направлении, указанном переменной Напр;

□ пока есть количество для списания.

#### Листинг 13.11

```
// списываем по партиям в зависимости от направления выборки
Пока (Запрос.Группировка("Партия", Напр)=1)и(ОсталосьСписать>0) Цикл
```

// списываем минимум из необходимого и имеющегося в партии СписКол=Мин (Запрос.КоличествоКонОст,ОсталосьСписать);

// определим сумму

СписСум=? (Запрос.КоличествоКонОст=СписКол,

Запрос.СуммаКонОст,

Окр (СписКол\*Запрос.СуммаКонОст/Запрос.КоличествоКонОст,2));

// собираем сумму себестоимости для бух. операции

Списано=Списано+СписСум;

// запишем измерения и ресурсы в регистр по этой партии Регистр.Партии.Партия = Запрос.Партия; Регистр.Партии.Поставщик = Запрос.Партия.Контрагент; Регистр.Партии.Сумма = СписСум; Регистр.Партии.Количество = СписКол; Регистр.Партии.Продажа = Цена\*СписКол; Регистр.Партии.Продажа = Цена\*СписКол; Регистр.Партии.ПривязыватьСтроку(НомерСтроки); Регистр.Партии.ДвижениеРасходВыполнить(); ОсталосьСписать=ОсталосьСписать-СписКол; Конециикла;

Комментарии в приведенном листинге помогут понять, для чего какие операторы программы использованы. Поясним логику выполнения расчета себестоимости.

Для партии определяем возможное количество для списания: минимальное значение из имеющегося в остатках партии и необходимого к списанию. Для этого количества определяем себестоимость. В регистр **Партии** записываем движение типа "Расход" и считаем количество к списанию с учетом списанного.

Если списаны все остатки товара по обрабатываемой партии и осталось еще некоторое количество товара для списания, тогда цикл обращается к следующей строке выборки и аналогично формируется следующее движение регистра **Партии**.

Оператор внутри цикла

Списано=Списано+СписСум;

собирает сумму для бухгалтерской операции, которая используется для передачи в модуль проведения.

На этом мы закончили настройку алгоритмов документов для компоненты "Оперативный учет".

# 13.2. Примеры. Создание отчета "Взаиморасчеты"

Назначение рассматриваемого в примере отчета — отражение информации по взаиморасчетам.

Для формирования отчета воспользуемся конструктором запросов в меню: Конструкторы | Запрос....

Используя навыки по работе с конструкторами, сформируем запрос, процедуру и таблицу, переходя последовательно по шагам конструктора, заполняя настройки следующим образом:

- □ Период задаем выбор периода в диалоге формы, результатом является вывод в форму отчета реквизитов начала (ВыбНачПериода) и конца (ВыбКонПериода) периода формирования отчета;
- □ Переменные задаем переменные запроса из измерений регистра Взаиморасчеты (Контрагент, Основание), ресурс регистра (Сумма) и атрибут ТекущийДокумент для расшифровки информации по документу, изменившему взаиморасчет с контрагентом;
- Функции задаем значения функций для получения итогов по регистру. В отчете нам необходимы значения начального состояния взаиморасчетов, сумма изменений взаиморасчетов за период, конечное состояние взаиморасчетов, поэтому используем функции HauOct(), Приход(), Pacход(), КонOct();
- Группировки задаем последовательно уровни группировок информации в запросе, используя заданные переменные запроса (см. шаг Переменные). Для нашего примера группируем запрос по переменной Контрагент, каждую переменную Контрагент группируем по Основанию, каждую запись по основанию — по ТекущемуДокументу, изменившему состояние регистра;
- □ Условия задаем условия, ограничивающие выборку информации в запрос. При указании режима Элементарное условие указываем для переменной Контрагент возможность выбора конкретного значения в форме диалога (режим Выбирать в диалоге) значения из реквизита ВыбКонтрагент. Указав знак сравнения равным "в", мы тем самым задали следующее условие, отличающее от сравнения вида "=": если не выбрано, значит для всех. Таким же образом задаем возможность для назначения в диалоге конкретного значения из реквизита ВыбОснование переменной запроса Основание;
- □ Текст запроса представляет текст сформированного по настройкам конструктора запроса. Нам остается согласиться (или не согласиться) с автоматической генерацией процедуры и таблицы.

В результате работы конструктора будет сформирована процедура, разделенная нами на несколько частей. Листинг 13.12 показывает часть процедуры, связанную с формированием запроса и его выполнением.

### Листинг 13.12

```
// Процедура генерации запроса Сформировать.
11
Процедура Сформировать ()
   Перем Запрос, ТекстЗапроса, Таб;
   //Создание объекта типа Запрос
   Запрос = СоздатьОбъект ("Запрос");
   ТекстЗапроса =
   "//{{ЗАПРОС(Сформировать)
   Период с ВыбНачПериода по ВыбКонПериода;
   |Контрагент = Регистр.Взаиморасчеты.Контрагент;
   Основание = Регистр.Взаиморасчеты. Основание;
   |ТекущийДокумент = Регистр.Взаиморасчеты.ТекущийДокумент;
   |Сумма = Регистр.Взаиморасчеты.Сумма;
   |Функция СуммаНачОст = НачОст (Сумма);
   |Функция СуммаПриход = Приход(Сумма);
   |Функция СуммаРасход = Расход(Сумма);
   |Функция СуммаКонОст = КонОст (Сумма);
   Группировка Контрагент упорядочить по Контрагент. Наименование без групп;
   Группировка Основание;
   |Группировка ТекущийДокумент;
   |Условие (Контрагент в ВыбКонтрагент);
   Условие (Основание в ВыбОснование);
   |"//}}ЗАПРОС
   // Если ошибка в запросе, то выход из процедуры
   Если Запрос.Выполнить (ТекстЗапроса) = 0 Тогда
      Возврат;
   КонецЕсли;
```

Далее (листинг 13.13) следуют операторы описания таблицы для вывода результатов запроса. Автоматически конструктор создал форму таблицы, представленную на рис. 13.1.

📕 Форма	🖣 Форма-Отчет.Взаиморасчеты							
		1 2	3	4	5	6	7	▲ B
	1							
Заголовок	2							
	3	Контрагент Наименование	Основание	ТекущийДокумент	СуммаНачОст	СуммаПриход	СуммаРасход	Суммаł
Итого	4	Итого		<Запрос.Сумм	аНачОст#N019.2>	аПриход#N019.2>	аРасход#N019.2>	ммаКонОс
Контрагент	5	<Запрос.ЗначениеУпорядоч		<Запрос.Сумм	аНачОст#N019.2>	аПриход#N019.2>	aPacxoд#N019.2>	ммаКонОс
Основание	6	<Запрос.ЗначениеУпорядоч	<Запрос.Зн	<Запрос.Сумм	aHaчOc⊤#N019.2>	аПриход#N019.2>	aPacxoд#N019.2>	ммаКонОс
ТекущийДо	7	<Запрос.ЗначениеУпорядоч	<Запрос.Зн	<Запрос.Текущий Д	аНачОст#N019.2>	аПриход#N019.2>	аРасход#N019.2>	ммаКонОс
	8							
	9							
	10							
\Е́見 Диалс	Сії Диалог 🗼 🗒 Модуль 🔪 🖄 Таблица 🖉 🖄 Сформировать 🖉							

Рис. 13.1. Форма таблицы, сформированная конструктором

# Листинг 13.13

```
// Подготовка к заполнению выходных форм данными запроса
Таб = СоздатьОбъект("Таблица");
Таб.ИсходнаяТаблица("Сформировать");
// Заполнение полей "Заголовок"
Таб.ВывестиСекцию("Заголовок");
Состояние("Заполнение выходной таблицы...");
Таб.Опции(0, 0, Таб.ВысотаТаблицы(), 0);
```

Потом конструктор формирует циклическую обработку строк запроса, организованную в виде трех вложенных циклов в соответствии с числом заданных группировок. Условия выполнения циклов заданы для каждого уровня группировки (1, 2 или3) методом Группировка(), позволяющим получать следующие значение выборки. Перед обращением к очередному уровню группировки выводятся строки в таблицу с итогами по группировке с помощью метода ВывестиСекцию() с указанием выводимой секции в скобках (листинг 13.14).

# Листинг 13.14

```
Пока Запрос.Группировка(1) = 1 Цикл

// Заполнение полей Контрагент

Таб.ВывестиСекцию("Контрагент");

Пока Запрос.Группировка(2) = 1 Цикл

// Заполнение полей Основание
```

```
Таб.ВывестиСекцию("Основание");
Пока Запрос.Группировка(3) = 1 Цикл
// Заполнение полей ТекущийДокумент
Таб.ВывестиСекцию("ТекущийДокумент");
КонецЦикла;
КонецЦикла;
```

После обработки всех записей запроса конструктор включил операторы для вывода итоговых строк секции **Итого**, назначения режима просмотра таблицы и вывод таблицы с названием окна **Сформировать** (листинг 13.15).

```
Листинг 13.15

// Заполнение полей "Итого"

Таб.ВывестиСекцию("Итого");

// Вывод заполненной формы

Таб.ТолькоПросмотр(1);

Таб.Показать("Сформировать", "");
```

КонецПроцедуры

Внесем некоторые усовершенствования в работу отчета. Во-первых, в модуль формы добавим процедуру ПриОткрытии () для установки начальных значений дат (листинг 13.16).

# Листинг 13.16

```
Процедура ПриОткрытии()
ВыбНачПериода= РабочаяДата();
ВыбКонПериода=ПолучитьДатуТА();
КонецПроцедуры
```

Для проверки правильности установки конечной даты формирования отчета, которая не может быть больше точки актуальности, введем новую процедуру, вызов которой осуществляется при изменении конечной даты формирования отчета в свойствах реквизита диалога **ВыбКонПериода** на вкладке **Дополнительно** (листинг 13.17).

### Листинг 13.17

Процедура УстДата()

Если ВыбКонПериода>ПолучитьДатуТА() Тогда

```
Предупреждение ("Нельзя устанавливать дату больше ТА!");
```

ВыбКонПериода=ПолучитьДатуТА();

КонецЕсли;

КонецПроцедуры

Теперь внесем изменения в отчет для выполнения требования по формированию отчета в режимах **Подробно по документам** или **Без детализации по документам**.

Для этого в экранную форму добавим элемент диалога **Флажок** для указания режима **Подробно по документам**. Определяем идентификатор элемента **Флажок** как **Подробно** для обращения в модуле к его значению. Тип элемента — "Число". Возможные значения элемента при этом:

0 — без детализации по документам;

1 — с детализацией по документам.

Далее в циклической обработке результатов выборки (листинг 13.18) добавляем вывод информации по документам в зависимости от установленного флажка **Подробно по документам**.

После внесения изменений третий уровень обработки выборки будет выглядеть следующим образом (листинг 13.18).

#### Листинг 13.18

```
Пока Запрос.Группировка(2) = 1 Цикл

// Заполнение полей Основание

Таб.ВывестиСекцию("Основание");

// вывод в режиме "подробно"

Если Подробно=1 Тогда

Пока Запрос.Группировка(3) = 1 Цикл

// Заполнение полей ТекущийДокумент

Таб.ВывестиСекцию("ТекущийДокумент");

КонецЦикла;

КонецЦикла;
```

Откорректируем форму отчета (см. рис. 13.1) для придания ей читаемой и компактной формы. После изменений она будет выглядеть так (рис. 13.2).

💻 Форма-Отчет. Взаиморасчеты								_ 🗆 ×	
		1	2 3	4	5	6	7	8	9 🔺
	1								
Заголовок	2		Ī		Вза	иморасчеты			
	3			<Период>					
	4		ĸ	онтрагент / Основание	Сальдо нач.	Приход	Расход	Сальдо кон.	
Итого	5		Итс	ого <Запрос.Су	ммаНачОст#N019.2>	имаПриход#N019.2>	маРасход#N019.2≻	ммаКонОст#N019.2>	
Контрагент	6		<3a	прос.Значение Упадалдшое Ба	ммаНачОст#N019.2>	имаПриход#N019.2>	маРасход#N019.2>	ммаКонОст#N019.2>	
Основание	7		<	Запрос.Значение Зверядо Фу	ммаНачОст#N019.2>	имаПриход#N019.2>	маРасход#N019.2>	ммаКонОст#N019.2>	
ТекущийДо	8			<Запрос.ТекущийДокумент	ммаНачОст#N019.2>	имаПриход#N019.2≻	маРасход#N019.2>	ммаКонОст#N019.2>	
	9								
	10								-
\Е́見Диало	ог		λI	🖺 Модуль 💦 🖄 Таб	блица 💦 🖄 С	формировать			

Рис. 13.2. Окончательный вид печатной формы отчета

Видоизменение таблицы для вывода печатной формы более подробно описано в следующем примере (см. разд. 13.3).

Для заполнения в секции **Заголовок** информации о периоде формирования отчета в модуле добавим операторы определения значения переменной **Пе-риод** в зависимости от параметров формирования отчета (листинг 13.19).

Листинг 13.19				
Период=ПериодСтр (ВыбНачПериода, ВыбКонПериода);				
Если Подробно=1 Тогда				
Период=Период+", по документам";				

КонецЕсли;

На этом формирование отчета закончено. Сохраните конфигурацию и проверьте работу отчета в режиме "1С:Предприятие".

# 13.3. Примеры. Создание отчета "Продажи"

Назначение рассматриваемого в примере отчета — отображение информации по продажам в разрезе поставщиков, партий и номенклатуры с расчетом прибыльности.

Для формирования отчета воспользуемся конструктором запросов в меню: Конструкторы | Запрос....

Используя навыки по работе с конструкторами (см. разд. 13.2), сформируем запрос, процедуру и таблицу (рис. 13.3).

Конструктор Запросов : Те
Партия = Резистр. Па Номенклатура = Рези Поставщик = Резистр Сумма = Резистр. Пар ТекущийДокумент = П Продажа = Резистр. Функция СуммаРасхо Функция ПродажаРася Группировка Постави Группировка Постави Группировка Партия
Условие (Поставщик в
9словие(Номенклатур
<ul> <li>Генерировать процедуру</li> <li>Генерировать таблици</li> </ul>
Генерировать таолицу
эдалять существующую т
1

Рис. 13.3. Результат работы конструктора

По аналогии с предыдущим отчетом внесем изменения для выполнения требования по формированию отчета в режиме Подробно по партиям или в режиме Без детализации по документам партии.

Для этого в экранную форму добавим элемент диалога **Флажок** (тип "Число") для указания режима **Подробно по партиям**.

В модуль формы добавим процедуру ПриОткрытии () для установки начальных значений дат и для проверки правильности установки конечной даты формирования отчета, которая не может быть больше точки актуальности (рис. 13.4).



Рис. 13.4. Форма диалога отчета по продажам

Далее мы представляем модуль формирования отчета с комментариями по внесенным изменениям (листинг 13.20).

### Листинг 13.20

```
// Процедура генерации запроса Сформировать.
11
Процедура Сформировать()
   Перем Запрос, ТекстЗапроса, Таб;
   //Создание объекта типа Запрос
   Запрос = СоздатьОбъект ("Запрос");
  ТекстЗапроса =
   "//{{ЗАПРОС(Сформировать)
   Период с ВыбНачПериода по ВыбКонПериода;
   |Партия = Регистр.Партии.Партия;
   |Номенклатура = Регистр.Партии.Номенклатура;
   |Поставщик = Регистр.Партии.Поставщик;
   |Сумма = Регистр.Партии.Сумма;
   |ТекущийДокумент = Регистр.Партии.ТекущийДокумент;
   |Продажа = Регистр.Партии.Продажа;
   // выбираем сумму себестоимости при реализации
   |Функция СуммаРасход = Расход(Сумма);
```

// выбираем сумму выручки при реализации |Функция ПродажаРасход = Расход (Продажа); Группировка Поставщик упорядочить по Поставщик. Наименование без групп; Группировка Номенклатура без групп; Группировка Партия упорядочить по Партия. ДатаДок; Условие (Поставшик в ВыбПоставшик); |Условие (Номенклатура в ВыбНоменклатура); |"//}}ЗАПРОС ; // Если ошибка в запросе, то выход из процедуры Если Запрос.Выполнить (ТекстЗапроса) = 0 Тогда Возврат; КонецЕсли; // выводим информацию о периоде отчета и детализации Период=ПериодСтр (ВыбНачПериода, ВыбКонПериода); Если Подробно=1 Тогда Период=Период+", по партиям"; КонецЕсли; // Подготовка к заполнению выходных форм данными запроса Таб = СоздатьОбъект ("Таблица"); Таб.ИсходнаяТаблица ("Сформировать"); // Заполнение полей "Заголовок" Таб. ВывестиСекцию ("Заголовок"); Состояние ("Заполнение выходной таблицы...");

Таб.Опции(0, 0, Таб.ВысотаТаблицы(), 0);

Поясним текст запроса, созданный конструктором в соответствии с условиями задачи.

В запросе используется функция **Расход**, поскольку при движении типа расход происходит запись в регистре значения рассчитанной себестоимости и партии, с которой идет списание. При расходе записываются в регистре себестоимость и сумма продажи.

В листинге 13.21 дано продолжение модуля формирования отчета, измененного по аналогии с предыдущим примером *(см. разд. 13.2)*, для вывода информации в различных режимах.

### Листинг 13.21

```
Пока Запрос.Группировка(1) = 1 Цикл

// Заполнение полей Поставщик

Таб.ВывестиСекцию("Поставщик");

Пока Запрос.Группировка(2) = 1 Цикл

// Заполнение полей Номенклатура

Таб.ВывестиСекцию("Номенклатура");

// вывод в зависимости от детализации

Если Подробно=1 Тогда

Пока Запрос.Группировка(3) = 1 Цикл

// Заполнение полей Партия

Таб.ВывестиСекцию("Партия");

КонецЦикла;

КонецЕсли;
```

КонецЦикла;

```
КонецЦикла;
```

```
// Заполнение полей "Итого"
Таб.ВывестиСекцию("Итого");
// Вывод заполненной формы
Таб.ТолькоПросмотр(1);
Таб.Показать("Сформировать", "");
```

КонецПроцедуры

Более подробно рассмотрим изменение таблицы печатной формы отчета (рис. 13.5).

		1	2	3	4	5	6
	1						
Заголовок	2						
	3		оставщик Ко	Партия ДатаДок	Номенклатура	СуммаРасход	ПродажаРасход
Итого	4		Итого		<3a⊓p	ос.СуммаРасход#N019.2>	.ПродажаРасход#N019.2>
Поставщик	5		<Запрос.Зна		<3a⊓p	ос.СуммаРасход#N019.2>	.ПродажаРасход#N019.2>
Партия	6		<Запрос.Зна	<Запрос.ЗначениеУпо	<3a⊓p	ос.СуммаРасход#N019.2>	.ПродажаРасход#N019.2>
Номенклат	7		<Запрос.Зна	<Запрос.ЗначениеУпо	<Запрос.ЗначениеУпорядо	ос.СуммаРасход#N019.2>	.ПродажаРасход#N019.2>
	8						

Для верхней строки секции Заголовок, объединяющей столбцы, отменим режим объединения через пункт меню Таблица | Разъединить.

После этого удалим столбцы **Партия ДатаДок** и **Номенклатура**. После колонки **ПродажаРасход** добавляем колонку **Прибыль**.

Каждой ячейке этой колонки всех секций настроим одинаковые свойства (рис.13.6).

		1	23 4	5	6	7
	1					
Заголовок	2			Отчет по пр	одажам	
	3		<Период>			
	4		Поставщик / Номенклатура	Себестоимость	Выручка	Прибыль
Итого	5		Итого <Запр	ос.СуммаРасход#N019.2>	.ПродажаРасход#N019.2>	<u>5.СүммаРасход)#N019.2&gt;</u>
Поставщик	6		<Запрос.ЗначениеУпорядо <b>чЮаар</b>	ос.СуммаРасход#N019.2>	.ПродажаРасход#N019.2>	с.СуммаРасход)#N019.2>
Номенклат	7		<Запрос.ЗначениеУпорядюзнава	ос.СуммаРасход#N019.2≻	.ПродажаРасход#N019.2>	с.СуммаРасход)#N019.2>
Партия	8		<Запрос.ЗначениеУпорядочия	ос.СуммаРасход#N019.2≻	.ПродажаРасход#N019.2>	с.СуммаРасход)#N019.2>
	9		Сво	йства ячейки	, 	×
	10					
	11		T	екст Положение Шриф	т Рамка Узор	ୁ ଜୁ –୍ଲା
	12		т			
	13			по рыражение	проле   Забиватенкрасне	
	14		F	Запрос ПродажаРасход-3	аппос СиммаРасход)#N01	92 📥 📃
	15			oanpooli ipagawar donog o	anpoologininar aonogymror	
	16					- OK
	17			I		
	18			Расшифровка		Отмена
	19			🗸 Зашита		06400475
	20					СОНОВИТВ

Рис. 13.6. Свойства ячейки таблицы

На вкладке **Текст** указываем значение ячейки — тип "Выражение" и его значение

(Запрос.ПродажаРасход-Запрос.СуммаРасход) #N019.2,

которое показывает формулу расчета прибыльности и формат вывода значения в таблицу.

На вкладке свойств ячейки Рамка задаем режим Обвести.

Такие же свойства задаем каждой ячейке на пересечении указанного столбца и каждой секции.

Изменим названия столбцов в шапке таблицы.

В верхней строке секции Заголовок выделим ячейки над столбцами таблицы (4 ячейки) и объединим их через пункт меню Таблица | Объединить.

В верхней строке секции **Заголовок** введем название отчета. Для этого в свойствах ячейки укажем тип значения "Текст" и введем значение — **Отчет** по продажам. Положение текста — центр по вертикали и горизонтали.

После настройки форма отчета выглядит как на рис. 13.7.

📕 Форма	🗏 Форма-Отчет.Продажи 📃 🗖 🗙								
		1	23	4	5	6	7		
	1								
Заголовок	2				Отчет по пре	одажам	~		
	3		<	Период>					
	4		1	Поставщик / Номенклатура	Себестоимость	Выручка	Прибыль		
Итого	5		Ит	того <Запр	ос.СуммаРасход#N019.2≻	.ПродажаРасход#N019.2>	с.СуммаРасход)#N019.2>		
Поставщик	6		<3	3апрос.ЗначениеУпорядо <b>чЮааар</b>	рс.СуммаРасход#N019.2≻	.ПродажаРасход#N019.2>	с.СуммаРасход)#N019.2>		
Номенклат	7		<	<Запрос.ЗначениеУпорядк <b>3нава</b>	рс.СуммаРасход#N019.2≻	.ПродажаРасход#N019.2>	с.СуммаРасход)#N019.2>		
Партия	8			<Запрос.ЗначениеУпорядочие	с.СуммаРасход#N019.2≻	.ПродажаРасход#N019.2>	с.СуммаРасход)#N019.2>		
	9								
	10								
	11								
	12								
\ЕЭ Диало	)r		λ	🗒 Модуль 💦 🔏 Табли	ица 💦 🛃 Сформ	ировать /			

Рис. 13.7. Форма таблицы отчета по продажам

Формирование отчета закончено. Сохраните конфигурацию и проверьте работу отчета в режиме **1С:Предприятие**.

# Глава 14



# Расчет

Компонента "Расчет" используется при решении задач с периодически повторяющимися расчетами для различных объектов. Возможности компоненты позволяют вести архив расчетов за прошедшие периоды. Компонента может применяться для различных областей учета, например, учета заработной платы, основных средств, ценных бумаг.

# 14.1. Краткое описание компоненты

Результаты расчета хранятся по периодам, по объектам, по видам расчетов. Отсюда основными характеристиками компоненты являются:

- **Период** различной длительности день, неделя, месяц, квартал, год;
- □ Объект как правило, объекты учета, для которых проводятся расчеты. Например: сотрудники, основные средства, ценные бумаги и т. п.;
- □ Виды расчетов алгоритмы для проведения расчетов с объектами, например:
  - для объекта Сотрудники начисление по окладу, удержания подоходного налога, начисление премии, выплата зарплаты и т. п.;
  - для объекта Основные средства начисление амортизации и т. п.;
  - для объекта Ценные бумаги начисление дивидендов и т. п.

Наиболее часто применяется компонента "Расчет" для решения задач по расчету заработной платы. Уместно использовать типовые решения на компоненте "Расчет" для расчета заработной платы при наличии на фирме разнообразных графиков работы (сменность), разнообразных видов начислений и удержаний, количества сотрудников более 50. "Расчет" — единственная компонента, позволяющая пользователю видеть всю хранимую информацию по расчетам с объектами (сотрудниками). Все итоги по объектам (сотрудникам) рассчитываются при формировании запросов, а не хранятся в отдельных файлах как в бухгалтерских итогах и регистрах оперативного учета.

Для получения результатов в журнале расчета необходимо сначала документом вести запись с нужным видом расчета и потом ее рассчитать. Особенностью компоненты "Расчет" является ввод записей в журнал расчетов только от документа, т. е. ручной ввод невозможен.

# 14.2. Журналы расчетов, виды расчетов, календари

Объектами метаданных, характерными для компоненты "Расчет", являются Журналы расчетов, Виды расчетов, Группы расчетов, Календари.

Только при наличии установленной компоненты эти объекты будут доступны для работы (рис. 14.1).





Ознакомимся с основными объектами компоненты.

*Журналы расчетов* — объекты метаданных, предназначенные для хранения, просмотра и редактирования результатов расчетов, произведенных в базе данных.

У каждого журнала существует *объект* — определенный справочник системы, для элементов которого и производятся расчеты. Журнал расчетов содержит записи, имеющие стандартные характеристики — *атрибуты*. Например, обязательные атрибуты следующие:

- □ Объект элемент справочника, для которого введена текущая запись журнала расчетов;
- **ВидРасчета** вид расчета текущей записи журнала расчетов;
- ДатаНачала дата начала действия записи журнала расчетов;
- **ДатаОкончания** дата окончания действия записи журнала расчетов;
- **Результат** значение результата расчета записи.

Форма-ЖурналРасчетов. Новый 1. 4		
	M	
Пиалог Модуль	_	

**Рис. 14.2.** Доступные атрибуты журнала расчетов

Эти атрибуты могут быть выведены в форму списка журнала расчета и доступны для просмотра пользователю (рис. 14.2). Кроме этого, все записи журнала расчетов имеют дополнительные признаки (это тоже атрибуты), отражающие их состояние, например:

- Документ документ-основание текущей записи журнала расчетов;
- РодительскийДокумент документ, который ввел текущую запись в журнал расчетов;
- □ **ПериодДействия** период действия записи журнала расчетов показывает, к какому периоду относится расчет записи;
- □ ПериодРегистрации период регистрации записи журнала расчетов показывает, в каком периоде расчетов запись была введена в журнал расчетов;
- **Сторно** признак сторнирующей записи;
- **Рассчитана** признак того, что запись рассчитана;
- Исправлена признак того, что запись исправлена вручную;

	X					
🗄 🖮 🖮 Корректные проводки	-					
📄 🖮 📾 Журнал расчетов						
📄 🖮 Атрибуты						
🗈 Документ						
РодительскийДокумент						
🛛 🖳 💮 Объект						
📗 🔤 🔤 ВидРасч						
📗 🔤 ДатаНачала						
📗 🔤 ДатаОкончания						
ПериодДействия						
ПериодРегистрации						
📗 🔄 🔤 Сторно						
Рассчитана						
Исправлена						
🛛 🔤 Фиксирована						
Перерасчет						
ПервичнаяЗапись						
Результат	-					
Покумент						
Синтаксис						
Документ						
Назначение:						
Документ-основание текущей записи журнала						
расчетов.						
Атрибут только для чтения.						
Журналами расчетов"						
	-					

- Фиксирована признак того, что результат расчета записи защищен от исправления;
- □ **Перерасчет** признак того, что запись является результатом перерасчета другой записи такого же вида расчета прошлого периода;

**Первичная** — первичная запись записи-перерасчета.

Эти атрибуты доступны только в модулях конфигурации. С описанием этих методов можно ознакомиться в синтакс-помощнике (рис. 14.3).

Журнал расчетов предназначен для проведения расчетов, повторяющихся через равные промежутки времени, называемый периодом расчета (**расчет-ный период**) (рис. 14.4). Все записи расположены в том или ином расчетном периоде.

📮 Журнал расчетов Зарп	
<u>И</u> дентификатор: Зарпл	
<u>К</u> омментарий:	
Справочник: Сотру	
Реквизиты	
Новый Изменить	
Графы отбора	
Подразделение	
ПрафикРаботы □Оклад	
□Лыготы	
ДатаПриема	

Рис. 14.4. Период журнала расчетов

Другая важная характеристика журнала расчетов — Дата отсчета — определяет точку начала отсчета периодов (рис. 14.5).

Обе эти характеристики не могут быть изменены при наличии записей в журнале расчетов.

📕 Журнал расче	етов Зарпла
<u>И</u> дентификатор	: Зарплат
<u>К</u> омментарий:	
<u>С</u> правочник:	Сотрудн
Реквизиты	
Новый Из	эменить 9,
– Графы отбора –	
Подразделе	ние
∐ ГрафикРабо □ Оклад	ты
Лыготы	
ДатаПриема	3

Рис. 14.5. Дата отсчета журнала расчетов

Как вы уже знаете, для каждого модуля формы существуют предопределенные процедуры, исполняемые при определенных событиях. Для модуля формы журнала расчетов это:

- □ ПриИсправленииРезультата отрабатывает в момент "ручного" исправления результата журнала расчетов;
- □ ПриОтменеИсправления отрабатывает в момент отказа от "ручного" исправления;
- ПриРасчете отрабатывает в момент выполнения одной из команд Рассчитать Запись, Рассчитать Объект, Рассчитать Документ;
- □ **ПриУстановкеОтбора** отрабатывает в момент установки отбора в форме журнала расчетов;
- □ ПриУстановкеГраницыПросмотра отрабатывает в момент изменения границы просмотра записей журнала расчетов;
- □ ПриУстановкеПредставления отрабатывает в момент изменения детализации представления записей журнала расчетов (по всем объектам расчетов, по одному объекту расчета, по одному документу расчета).

Как указывалось ранее, записи в журнал расчетов вводятся только документами. Для этого существуют специальные методы, описанные в документации. Рассмотрим основные методы:

```
ВвестиРасчет (Объект, ВидРасчета, НачалоДействия,
```

ОкончаниеДействия, Результат)

при использовании этого метода для ввода вида расчета, не участвующего в вытеснении, не будет анализироваться наличие аналогичных записей с тем же периодом действия, т. е. можно будет ввести запись с таким же видом расчета;

🗖 Записать Расчет (Объект, ВидРасчета, НачалоДействия,

ОкончаниеДействия, Результат)

при использовании этого метода для ввода вида расчета, участвующего в вытеснении, будут вытесняться аналогичные записи с тем же периодом действия, имеющие меньший или равный приоритет вытеснения (вид расчета вытесняет сам себя), т. е. не будет вводиться запись с таким же видом расчета.

Выборка записей из журнала расчета напрямую зависит от методов. В первую очередь от наличия у записи двух временных характеристик: периода действия и периода регистрации.

Наличие двух периодов объясняется наличием различных вариантов ввода записей в журнал расчетов документом, дата которого:

- принадлежит периоду действия, т. е. действующему в журнале расчетов периоду расчета на момент ввода документа;
- не принадлежит периоду действия, т. е. действующему в журнале расчетов периоду расчета на момент ввода документа.

Поэтому и выборка записей может производиться в зависимости от этого разными методами:

🗖 Выбрать Записи (Дата1, Дата2)

при использовании этого метода выбираются записи, период действия которых лежит в указанном интервале. Они могут быть введены в любом расчетном периоде;

🗖 ВыбратьПериод(Дата)

при использовании этого метода выбираются записи, которые были введены в том расчетном периоде, которому принадлежит дата. Период действия их может быть любым.

Следующим характерным для компоненты "Расчет" объектом метаданных являются ВидыРасчетов.

ВидыРасчетов — объекты метаданных, в которых описываются алгоритмы проведения расчетов для объектов и производится запись результата расчета в атрибут **Результат**. В свою очередь ВидыРасчетов имеют набор характеристик атрибутов:

- Код код вида расчета;
- □ Наименование соответствует комментарию, заданному при конфигурировании ВидаРасчета;
- Очередность число, соответствующее очередности расчета ВидаРасчета, заданной при конфигурировании;
- ПриоритетВытеснения число, соответствующее приоритету вытеснения ВидаРасчета, заданному при конфигурировании.

Обращение к ВидамРасчетов и их атрибутам возможно только в программных модулях компоненты.

Как правило, при расчете заработной платы встречаются виды расчетов, использующие результаты других расчетов, например, при расчете налогов используются результаты расчета начислений заработной платы. Поэтому важна очередность выполнения тех или иных расчетов.

Для организации последовательности проведения расчетов существует понятие *приоритета*. Установление приоритетов отражается в последовательности расположения записей в журнале расчета. В случае одинаковых приоритетов записи располагаются в хронологическом порядке, т. е. согласно последовательности ввода документов.

При указании приоритета виду расчета необходимо помнить: чем меньше приоритет, тем раньше он выполняется.

Приоритеты устанавливаются при настройке вида расчета в Конфигураторе (рис. 14.6).

Для исключения ввода в одном и том же интервале дат противоречащих друг другу расчетов служит *настройка вытеснения*. Например, в одном промежутке дат не могут быть одновременно произведены начисления по окладу и больничному. При этом вид расчета, который вытесняется (замещается), называется вытесняемым, а вид расчета, который вытесняет (замещает), вытесняющим.

Настройка вытеснения производится при настройке вида расчета. При нажатии кнопки **Настройка вытеснения** открывается форма настройки вытеснения, приведенная на рис. 14.6.

**Правила перерасчета** являются вспомогательным объектом метаданных у видов расчетов и предназначены для автоматического отслеживания правил влияния результатов расчета друг на друга.

Расчет

🖹 Вид расчета Новый1		
Идентификатор: Новый1	Синоним:	
<u>К</u> омментарий:		
Приоритет: 1 🛨	Настройка	вытеснения
	Порядок вытеснения - Новый 1	×
	Вытесняет виды расчета:	Вытесняется видами расчета: Новый1 Отмена Помощь

Рис. 14.6. Настройка правил вытеснения

Настройка правил перерасчета производится при настройке вида расчета. В Конфигураторе в дереве метаданных в разделе Виды расчетов двойным нажатием мыши на объекте Правила перерасчета открывается форма настройки правил, приведенная на рис. 14.7.

Правила перерасчетов определяют связь между ведущим (запускающими перерасчет) и зависимым (перерассчитываемым) видами расчетов, а также определяют тип перерасчета по периодам перерасчета (по текущему периоду, по будущим периодам).

При использовании в алгоритмах правил перерасчетов возможно обращение к их характеристикам — атрибутам:

- □ **Тип** позволяет прочитать/установить значение типа правила перерасчета. Данный атрибут может принимать следующие значения:
  - 0 при вводе записи журнала расчетов с любым периодом действия зависимые виды расчетов должны быть перерассчитаны только в текущем периоде журнала расчетов (фактически это приведет к снятию флажка "рассчитанности" записи);
  - 1 зависимые виды расчетов должны быть перерассчитаны в том же периоде, что и вводимая запись журнала расчетов;

- 2 зависимые виды расчетов должны быть перерассчитаны в нескольких периодах, следующих за периодом действия вводимой записи журнала расчетов (количество периодов задано атрибутом КоличествоПериодов);
- □ КоличествоПериодов позволяет прочитать/установить количество периодов перерасчета.

3==	🖥 Правило пере	ерасчета Но
	<u>И</u> дентификатор:	Новый1
	<u>К</u> омментарий:	
	При ввод	це видов расч
	⊡Новый1	
	Отолько в тен	кущем перио,
	•в том же пе	риоде журна)
	Ов следующи	x 1 🗄

Рис. 14.7. Настройка правил перерасчета

Более подробно о настройке и методах работы с правилами перерасчета можно прочитать в документации к программному обеспечению.

Отдельным объектом метаданных является объект **ГруппыРасчетов**, предназначенный для объединения видов расчетов по каким-либо признакам для облегчения дальнейшей работы с ними. Например, группа расчетов **ОблагаетсяНалогом** в нашем примере включает виды начислений, результаты которых участвуют в расчете подоходного налога.

Группы расчетов характеризуются двумя атрибутами:

- Код код группы видов расчета;
- □ Наименование соответствует комментарию, заданному при конфигурировании группы видов расчетов.

Обращение к **ГруппамРасчетов** и их атрибутам возможно только в программных модулях компоненты. На рис. 14.8 показано окно настройки состава группы расчета в конфигураторе.

•• Группа расчетов Новый1	
Идентификатор: Новый1	
<u>К</u> омментарий:	
Включает виды расч	



п( <sup>@</sup> Конфигурация * _□≍								
🔁 🍇 🛛 🖡 🕇 🕹 🗎								
🕂 🖓 Планы Счетов								
🔚 Виды Субконто								
🗄 🖳 🙀 Операция								
🗄 🖳 Проводка								
Пегистры								
🚊 🛺 Журналы расчетов	Свойства Календаря	x -						
Новый1								
🖃 🎼 Виды расчетов	Общие Календарь Права Миграция							
Правила перерасче	Длина дня							
інні Новыйі Пракод	Стартовая дата: 01.01.2006 📻 0.00 🐲							
Новыи	8.00							
— Да Труппы расчетов	8.00	ок 📗						
	8.00							
	8.00	Отмена						
новый1	0.00							
🚯 Метаданные 🙀 Интерфе	йсы 🕂 🕅 рава							

Рис. 14.9. Настройка календаря

Календари — объекты метаданных, предназначенные для вычисления интервалов времени различных периодов. Календари характеризуются двумя атрибутами:

**Дата** — дата текущей строки календаря; доступен только для чтения;

□ Значение — значение текущей строки календаря, на которой позиционирована выборка. Тип — число с двумя знаками после запятой.

Компонента содержит стандартный календарь **Праздники** для учета праздничных дней периода (см. рис. 14.1). В этом календаре атрибут **Значение** может быть равен 0 или 1, что соответствует ответам **Нет** или **Да** на вопрос: "Дата является праздником?"

Обращение к атрибутам календарей возможно только в режиме "1С:Конфигуратор" при настройке в окне свойств объекта. На вкладке **Общие** задаются идентификатор, синоним и комментарий. На вкладке определяются основные параметры календаря (рис. 14.9).

Заполнение календарей производится в режиме "1С:Предприятие" в соответствии с настройками (рис. 14.10).

📭 Календ	арь новый1	
<	★         >>>           rsd5pe         2006         >>           p         HT         C6         Bc           7         28         29         30         1           4         5         6         7         8           1         12         13         14         15           8         19         20         21         22           25         26         27         28         29	Автозаполнение Учитывать праздники О дней 0.00 часов 0.00 значение по
30 31	1 2 3 4 5 Автозаполнени За период С: 01.10.06	е календаря

Рис. 14.10. Заполнение календаря

# 14.3. Применение компоненты "Расчет"

Напомним, что описанные примеры показывают лишь подходы к конфигурированию на примере расчета заработной платы.

Для работы запустите учебную конфигурацию в режиме Конфигуратор.

Откройте окно Конфигурация.

# 14.3.1. Создание объектов в сквозном примере компоненты "Расчет"

Для настройки расчета заработной платы в нашей конфигурации нам необходимо будет дополнить ее как новыми объектами, так и существующие уже объекты дополнить новыми реквизитами и алгоритмами.

Нам понадобятся дополнительно две константы (см. описание сквозного примера в *славе 4*). Мы создадим их так, как описано в *славе 5*. Результат представлен на рис. 14.11.

<sub>ка</sub> Конфигурация * ∎□⊠								
🖹 🏷 🖄 🖡 🕇 🏭								
	войства Констан Общие Дополни Идентификатор:	пты тельные Права Миграция ) СтавкаПН	× N N					
<ul> <li>Э Основнаяд</li> <li>МРП</li> <li>СтавкаПН</li> <li>Справочники</li> <li>Справочники</li> <li>Документы</li> <li>Документы</li> <li>Ти Журналы докум</li> <li>Перечисления</li> </ul>	Синоним: Комментарий: Тип значения: Длина:	Ставка подоходного налог Ставка подоходного налога <<Число>> 5 • Неогр. Точность: 2 •	ОК Отмена Обновить					
<ul> <li>⊕ ∑ Отчеты</li> <li>⊕ ⊇ Обработки</li> <li>⊡ Ш Метаданные 200 И</li> </ul>	нтерфейсы	Права	T					

Рис. 14.11. Константы для расчета зарплаты

Согласно условиям примера нам понадобятся справочники для ведения списка подразделений предприятия и определения сотрудникам основного графика работы. Новые справочники создаем в конфигураторе по методике, описанной в *главе 5*.

Для нашего примера эти справочники не содержат дополнительных реквизитов — только **Код** и **Наименование**. Структура справочников одноуровневая, представление элементов организуем в форме списка.

Создадим справочник Подразделения таким, как показано на рис. 14.12 и 14.13.

Для этого выведем реквизиты в форму списка, добавим название списка и заголовок экранной формы (рис. 14.13).

Справочник Подразделени:	
Идентификатор: Подразделения	
<u>К</u> омментарий:	
Кол-во, уровней: 💈 \Xi	
Длина кода: 🛛 🗧 🚊	
Длина наименования: 25 📑	
Серии кодов	
О Во всем справочнике	
В пределах подчинения	
- Тип кода	
• Числовой	
С Текстовый	
Основное представление	
С В виде кода	
• В виде наименования	
🗖 Одна форма для элемента і	
<u>О</u> писание	

Рис. 14.12. Окно Справочник Подразделения

🚛 Форм	ла-Спра	авочник	
🗐 Пол	цраздел	тения	
Попразлелен			
Ko	д I	Таимен	
,			
4			
\ 🗐 Ди	алог		

Рис. 14.13. Форма списка справочника Подразделения

Справочник графиков рабочего времени (Графики) аналогичен справочнику Подразделения.

При работе конструктора в интерфейс **Расчетчик** добавляются пункты меню для доступа к вновь создаваемым справочникам, формируя интерфейс пользователя для расчета заработной платы (рис. 14.14).



Рис. 14.14. Меню Расчетчика

Согласно сквозному примеру, для расчета заработной платы нам понадобятся дополнительные характеристики сотрудников. Для этого существующие справочники дополним новыми реквизитами. Новые реквизиты в справочники дополняем в Конфигураторе по методике, описанной в *главе 5*.

Добавим следующие реквизиты сотрудника, используемые при расчете заработной платы. Часть реквизитов будет изменяться со временем. Это условно постоянные или периодические реквизиты:

- **Подразделение** тип "Справочник. Подразделения";
- График работы тип "Справочник. Графики";
- □ **Оклад** тип "Число" (9,2);

□ **Льгота** — тип "Перечисление.Булево", т. е. значение вида "Да" или "Нет". Часть реквизитов будет постоянной:

- **Дата приема** и **Дата увольнения** тип "Дата";
- **Приказ о приеме** тип "Документ Приказ о приеме";

Полностью состав реквизитов представлен на рис. 14.15.

🔊 Справочник Сотрудники
Идентификатор: Сотрудники
<u>К</u> омментарий:
Кол-во. уровней: 🚺 🌐
Длина кода: 🗧 🗧
Длина наименования: 25 📑
Серии кодов
О Во всем справочнике
💿 В пределах подчинения
– Тип кода
• Числовой
О Текстовый
Основное представление
О В виде кода
В виде наименования
📔 📋 Одна форма для элемента
<u>О</u> писание
1

Рис. 14.15. Структура справочника "Сотрудники"

Настроим экранную форму справочника "Сотрудники", выведя в нее дополнительные реквизиты (рис. 14.16).

Информацию о расчетах в журнал расчетов можно разместить только из документа *(см. разд. 14.1)*, поэтому для ведения расчетов зарплаты нам понадобятся документы. Создадим эти документы и их экранные формы.

Разработку программных модулей документов будем описывать в разд. 14.4.

Новые документы в конфигурации создаем по методике, описанной в *славе 5*, т. е. с помощью конструктора.

При работе с конструктором новых документов создаем сразу журналы для каждого вида документа. При решении задачи сквозного примера создаем обычные журналы для хранения документов. Эти журналы будут содержать графы, необходимые для организации удобного поиска документов.

💻 Форма элемен	та-Спр	авочник.Со			
📴 Сотрудник					
	Код:	Код			
Фамилия, имя, отч	ество:	Наименова			
Подразделение:	Подраз	деление			
ГрафикРаботы:	График	Работы			
ОК	Закј	рыть			
\ 🗐 Диалог		Тодуль			

Рис. 14.16. Экранная форма справочника "Сотрудники"

Как вы уже знаете, конструктор позволяет также создать пункты меню в интерфейсах для работы с новыми объектами.

А теперь создадим конструктором "заготовки" документа "Приказ о приеме на работу".

Документ "Приказ о приеме на работу" будет служить для ввода в справочник "Сотрудники" информации по сотрудникам на дату приема на работу. Эта информация используется при расчете зарплаты и может в дальнейшем изменяться уже в справочнике.

В каждом документе присутствуют по умолчанию атрибуты ДатаДок и НомерДок.

Перечень реквизитов шапки документа "Приказ о приеме на работу":

- ДатаПриема тип "Дата";
- **Подразделение** тип "Справочник. Подразделения";
- **Сотрудник** тип "Справочник.Сотрудники";
- Льготы тип "Перечисление.Булево";
- **Оклад** тип "Число", неотрицательное;
- ГрафикРаботы тип "Справочник. Графики".

Табличной части документ не имеет. На рис. 14.17 представлено окно с характеристиками и реквизитами документа.

🖹 Документ Пр	иказОПр	иеме
Идентификатор:	ПриказОГ	Трием
<u>К</u> омментарий:	,	
– Реквизиты шаг	, іки — — —	
ДатаПриема Подразделени Сотрудник Льготы Оклад ГрафикРаботь	ie M	
Новый И	зменить	Уда/
Номер		
<u>Ну</u> мератор:	<< He	назнач
Периодичность	По все	м дан
🔽 Автоматиче	ская нуме	рация
<ul><li>Разрешить</li><li>Автоматиче</li><li>Автоматиче</li></ul>	проведени ское удал ская нуме	ю доку ение д рация
Создавать опера	цию:	
<u>В</u> вод на основан	ии	

Рис. 14.17. Настройка документа "Приказ о приеме на работу"

	💻 Форма-Докуме 👼 Приказ о прие	нт.При эме на					
	Приказ о пр						
	Сотрудник	Сотр					
I	Подразделение	Подр					
	График работы	Граф					
	ок	Закры					

Рис. 14.18. Экранная форма документа "Приказ о приеме на работу"

Выведем реквизиты в экранную форму документа и оптимизируем их расположение, например, как показано на рис. 14.18.

Автоматически конструктором был создан новый журнал документов Приказы о приеме. Создадим в журнале графы, как показано на рис. 14.19.

📑 Журнал ПриказОПри	еме	
Идентификатор: Прик	азОПриеме <u>С</u> иноним: Прика:	зы о приеме
<u>К</u> омментарий:		
_ Документы	Графы	
ПриказОПриеме	Сотрудник	
	Графа журнала - Сотрудник	×
	Идентификатор: Сотрудник	Синоним: Сотрудник
	Комментарий:	
Состав	Выбранные значения	Возможные значения
Тип журнала	🗅 Документ.ПриказОПриеме.Сотру	— 🔄 Документы — 🖹 ПриказОПриеме
💿 Обычный		
О Общий	•	🛏 🗌 — 🔤 ДатаПриема
О Дополнительный		Подразделение
ļ		Льготы
		— 💷 Оклад 🚽

Рис. 14.19. Настройка журнала Приказы о приеме

После этого настроим форму списка журнала с учетом удобства поиска, например, как на рис. 14.20. При этом для журналов, в которых выводится только один вид документа (обычный), можно не выводить колонку **Вид документа** для экономии места.

Теперь будем создавать новый регламентный документ "Начало месяца".

Записи в журнал расчетов могут вводиться только документами. С целью ввода записей-заготовок для расчета и создадим этот документ. После этого можно будет описывать обработку расчета введенных записей.

Документ "Начало месяца" должен формироваться пользователями в начале каждого месяца после смены расчетного периода.

Документ может формировать записи как по всем сотрудникам в целом, так и по выбранному подразделению или сотруднику.
💻 Форма-Журнал.ПриказОІ				
🛱 Приказы о приеме				
П	риказы о при	еме		
Дата приказа Номе				
llŀ				
IF				
IE				
IF				
IL				
1				
•				
🗐 Диалог 🖉 Моду				

Рис. 14.20. Форма списка журнала Приказы о приеме

Поэтому для документа "Начало месяца" введем следующие реквизиты шапки документа:

**Подразделение** — тип "Справочник.Подразделения";

**Сотрудник** — тип "Справочник.Сотрудники".

Табличной части документ не имеет.

Выведем реквизиты в экранную форму документа, например, как на рис. 14.21.

После расчета всех записей по заработной плате можно формировать документ по выдаче заработной платы сотрудникам. Поэтому нам необходимо создать в Конфигураторе документ "Выплата зарплаты".

По рис. 14.22 можете определить самостоятельно реквизитный состав документа.

Все реквизиты можете самостоятельно вывести в экранную форму, например, как на рис. 14.23.

Поясним размещение реквизита **Окончательный Расчет**. Тип реквизита — "Число", количество знаков — 1, возможные значения "1" или "2". При расположении реквизита в форме диалога указываем вставку элемента типа "Переключатель" (рис. 14.24).



Рис. 14.21. Экранная форма документа "Начало месяца"

🖹 Документ Вы	ілатаЗар	плат
<u>И</u> дентификатор:	Выплата З	арпла
<u>К</u> омментарий:		
– Реквизиты шап	ш	
Подразделени: Окончательны	е йРасчет	
Новый Из	менить	Удал
Номер		
<u>Н</u> умератор:	<< He ⊨	назнач
Периодичность:	По все	мдан
🔽 Автоматиче	ская нумер	рация
<ul> <li>Разрешить г</li> <li>Автоматичен</li> <li>Автоматичен</li> </ul>	проведени ское удале ская нумер	е док <u>у</u> эние д рация
Создавать опера	цию:	
<u>В</u> вод на основани	ни	

Рис. 14.22. Настройка документа "Выплата зарплаты"

📕 Форма-Д	lокумент.Вь					
舅 Выплата	а заработно					
Выплата з	аработной г					
Подразде.	ление Подраз					
– Вид выпла	ты					
Окончал	гельный расче					
О Промеж	(уточная выпл					
N Co	трудник					
Рассчитать						
🖓 🗐 Диалог 🛛 🖊 🧮						

Рис. 14.23. Экранная форма документа "Выплата зарплаты"



Рис. 14.24. Добавление реквизита Окончательный Расчет

Ниже добавляем элемент диалога типа "Переключатель". Делаем надписи и создаем рамку вокруг реквизитов, как показано на рис. 14.23. При такой последовательности добавления реквизитов в форму диалога настраивается режим "переключателя", т. е. одномоментно может быть "включен" только один из флагов. Реквизит, являющийся первым в группе (см. рис. 14.24), должен иметь идентификатор. Именно ему присваивается значение 2 при включении и 1 — при отключении (включении второго).

321

Дополнительно в экранную форму документа добавлены кнопки (см. рис. 14.23):

**Заполнить** — для заполнения табличной части документа;

**Печать** — для формирования печатной формы документа.

По примеру журнала **Приказы о приеме** создайте журнал **Выплата зарплаты** для документов по выплате заработной платы. Воспользуйтесь для этого помощью конструктора **Новый журнал документов...**.

Как уже говорилось, одной конфигурации может быть произвольное количество журналов расчета. В нашем примере мы создадим единственный журнал расчета зарплаты сотрудников. Этот журнал будет иметь периодичность, равную месяцу.

Конструктор жур	нал		
Идентификатор:	He		
Синоним:			
Комментарий:			
Обязательно ука "Синоним" може	эжит ят бы		
расчетов в режи "комментарий" м	ме з можн		
	_		
🔽 Использовать	конс		
< Назад	Да.		

Рис. 14.25. Создание конструктором журнала расчетов

Для создания журнала расчетов воспользуемся конструктором из пункта меню **Конструкторы | Новый журнал расчетов...**. На рис. 14.25 представлены основные шаги работы конструктора.

После создания конструктором объекта в конфигурации и вывода в меню соответствующих пунктов настроим экранную форму журнала — форму списка (рис. 14.26).



Рис. 14.26. Форма списка журнала расчетов

Для решения нашего примера необходимо создать несколько видов расчетов:

- **П** по начислениям **ПоОкладу;**
- по удержаниям Подоходный Налог;
- результирующий ВыплатаЗарплаты.

Определим общие правила настройки приоритетов, т. е. последовательности выполнения расчетов:

- виды расчетов по начислениям будут иметь приоритеты от 1 до 100;
- 🗖 виды расчетов по удержаниям будут иметь приоритеты от 101 и выше.

Не рассматриваем алгоритмы вытеснения для видов расчета, поскольку ни один из видов расчета в нашем примере не исключает введение другого. Настройку и использование алгоритмов вытеснения можно изучить по документации, поставляемой с программным обеспечением.

Рассмотрим пример создания вида расчета **ПоОкладу**. Вид расчета создается в конфигурации в дереве метаданных. Выберите группу объектов **Виды расчетов** и в пункте меню **Действия** выберите подпункт **Новый элемент**. В открывшемся окне настройки укажите необходимые параметры, как показано на рис. 14.27.

🖹 Вид расчета	ПоОкладу
Идентификатор:	ПоОкладу
<u>К</u> омментарий:	По окладу
Приоритет:	1 🗧
Pro	
	ючать в групп

Рис. 14.27. Настройка вида расчетов ПоОкладу

Для вида расчета Подоходный Налог задаем приоритет 101, а для вида расчета Выплата Зарплаты — 500.

Аналогично создаем еще два вида расчета согласно условиям примера.

Для упрощения написания алгоритмов расчета, например, подоходного налога, создаем группу расчетов **ОблагаетсяНалогом**. Группа создается методом **Новый элемент** и настройкой в диалоговом окне группы расчетов. Настроим название объекта и перенесем налево виды расчетов, результаты по которым будут участвовать в расчете налога (рис. 14.28).

Создаем еще две группы расчетов: Начисления и Удержания. После распределения видов расчетов по группам расчетов изменяется и настройка видов расчетов, так как становится заполненным окно **Включать в группы**. Посмотрим это на примере вида расчета **ПоОкладу** (рис. 14.29).

•	• Группа расче	тов Облага
ſ	Идентификатор:	Облагается
	<u>К</u> омментарий:	Облагается
	Включа ПоОкладу	ает виды расч

Рис. 14.28. Настройка группы расчетов ОблагаетсяНалогом

🖹 Вид расчета	ПоОкладу
<u>И</u> дентификатор:	ПоОкладу
<u>К</u> омментарий:	По окладу
<u>П</u> риоритет:	1 🕂
Вкл	ючать в групп
Начисления ОблагаетсяНа	алогом

Рис. 14.29. Создание групп расчетов Начисление и Удержание

Создавая календарь, опишем *правила автозаполнения* для удобства работы пользователей.

н <sup>#</sup> Конфигурация *	
● 後 図 ↓ ↑ 41 ■	
Перечисления	
⊕…∑ Отчеты     Свойства Календаря	×
🕀 🐨 📅 Планы Счетов 🛛 Оощие Колендоро   Права   Миграция	V V
🕀 💼 Виды Субконто Длина дня	
🕀 🕰 Операция — Стартовая дата; 01.01.2006 📺 0.00 🚎	
	ок (
Энварь 2006 6.67 Энварь 2006 6.67 Энварь 2006 6.67	
Виды расчетов 26.27.28.29.30.31 11 6.67	Отмена
⊕ 4 Группы расчетс 2 3 4 5 6 7 8 0.00	
🖻 📮 Календари	Обновить
Праздники 16 17 18 19 20 21 22	
Основной 23 24 25 26 27 28 29	
<b>30 31</b> 1 2 3 4 5	
🚯 Метаданные 🎯 Интерфейсы 🐥 Права	

Рис. 14.30. Настройка календаря Основной

В нашем примере создаем один календарь **Основной** и устанавливаем для него стартовую дату. С нее будет начинаться автозаполнение. Также указываем длину дня в календаре (рис. 14.30).

Сохраним внесенные в конфигурацию изменения.

# 14.4. Написание алгоритмов в объектах компоненты "Расчет"

После создания в Конфигураторе необходимых объектов можно приступать к написанию процедур в них, используя при необходимости ссылки на существующие объекты и виды расчетов.

Напомним, что при создании документов конструктором у нас были только предопределенные процедуры ПриОткрытии() модуля формы (листинг 14.1) и ОбработкаПроведения() модуля документа (листинг 14.2).

#### Листинг 14.1

```
Процедура ПриОткрытии()
```

```
ПриЗаписиПерепроводить (1) ;
```

```
КонецПроцедуры
```

#### Листинг 14.2

```
Процедура ОбработкаПроведения()
```

//Здесь следует написать алгоритм проведения документа

КонецПроцедуры

Основное *назначение документов* — сохранить вводимую информацию и результаты расчетов. При проведении документов компоненты "Расчет" могут сохраняться изменения реквизитов справочников (например, **ПриказОприемеНаРаботу**) и вводиться записи в журнал расчетов. При этом, как правило, результаты этих записей равны нулю, поскольку многие расчеты используют результаты других расчетов. Определение значений результатов происходит при окончательном расчете зарплаты, когда последовательно рассчитываются записи журнала расчетов. Алгоритмы расчетов результатов описаны в соответствующих видах расчетов в модулях.

Дополним документы сквозного примера алгоритмами.

Начнем с документа "Приказ о приеме на работу", основное назначение которого — записать в справочник "Сотрудники" значения периодических реквизитов, необходимых для расчета заработной платы: оклад, льготы, график работы, подразделение.

Проведение документа не отражается на результатах расчета, поэтому можно не определять его принадлежность к компоненте "Расчет" (см. рис. 14.17).

В модуле формы документа конструктором создана одна предопределенная процедура Приоткрытии () (см. листинг 14.1).

Дополним модуль формы следующими процедурами:

Вводнового() — для заполнения реквизитов документа при создании нового документа. В этой же процедуре (листинг 14.3) проверяется ситуация создания документа копированием.

#### Листинг 14.3

Процедура ВводНового (Копирование)

```
Если Копирование = О тогда
```

// если не скопирован - заполняем реквизиты

```
ДатаДок = РабочаяДата();
```

```
ДатаПриема = ДатаДок;
```

```
Льготы = Перечисление.Булево.Да;
```

КонецЕсли;

КонецПроцедуры

Опишем проверку заполнения обязательных реквизитов. Используем для этого предопределенную процедуру ПриЗаписи() (листинг 14.4). Если реквизиты не заполнены, документ не записывается. В этом случае метод СтатусВозврата() возвращает значение 0, т. е. процедура не завершена. Использование метода СтатусВозврата() имеет смысл только в предопределенных процедурах, которые формируют сигнал системе о нормальном завершении. По умолчанию СтатусВозврата() = 1.

#### Листинг 14.4

```
Процедура ПриЗаписи()
  // Проверка заполнения сотрудника
  Если Сотрудник.Выбран()=0 Тогда
  Сообщить ("Выберите сотрудника!");
  СтатусВозврата (0);
  КонецЕсли;
  // Проверка заполнения Подразделения
  Если Подразделение.Выбран()=0 Тогда
  Сообщить ("Выберите подразделение!");
  СтатусВозврата (0);
  КоненЕсли:
  // Проверка заполнения оклада
  Если Оклад=0 Тогда
  Сообщить ("Укажите оклад!");
  СтатусВозврата(0);
  КонецЕсли;
  // Проверка заполнения График работы
  Если ГрафикРаботы.Выбран()=0 Тогда
  Сообщить ("Выберите График!");
   СтатусВозврата(0);
  КонецЕсли;
КонецПроцедуры
```

В модуле документа в процедуре обработкаПроведения () организуем запись периодических реквизитов существующего сотрудника (листинг 14.5) в справочнике "Сотрудники". Для записи периодических реквизитов используем метод установить РеквизитСправочника (). Использование этого метода обеспечивает автоматическое удаление записанных значений реквизитов при отмене проведения документа.

#### Листинг 14.5

```
Процедура ОбработкаПроведения ()
  //Здесь следует написать алгоритм проведения документа
  СпрСотрудники=СоздатьОбъект ("Справочник.Сотрудники");
  СпрСотрудники. НайтиЭлемент (Сотрудник);
  ТекСотрудник=СпрСотрудники.ТекущийЭлемент();
  Установить РеквизитСправочника (СпрСотрудники,
                                "Оклад", Оклад, ДатаПриема);
  Установить РеквизитСправочника (СпрСотрудники,
                   "Подразделение", Подразделение, ДатаПриема);
  Установить РеквизитСправочника (СпрСотрудники,
                                "Льготы", Льготы, ДатаПриема);
  Установить РеквизитСправочника (СпрСотрудники,
                    "ГрафикРаботы", ГрафикРаботы, ДатаПриема);
   СпрСотрудники.ДатаПриема = ДатаПриема;
   СпрСотрудники.ПриказОПриеме = ТекущийДокумент();
   Попытка
     СпрСотрудники.Записать ();
  Исключение
   Сообщить ("Неудачная попытка записи элемента "
           + СокрЛП (Сотрудник. Наименование)
           +". Возможно, объект заблокирован...", "!");
     СтатусВозврата(0);
     Возврат;
   КонецПопытки;
```

КонецПроцедуры

Обращаем внимание, что в модуле использована конструкция языка по обработке исключительной ситуации:

Попытка

·····•

Исключение

.....

КонецПопытки;

#### Расчет

Обрабатывается ситуация записи элемента справочника. Действие может быть не выполнено, например, если форма элемента открыта (объект заблокирован). В этом случае не должно быть аварийных или непонятных завершений работы алгоритмов. После оператора Исключение описываются действия по информированию пользователя и завершению процедуры.

При отмене проведения документа "Приказ о приеме" автоматически очистятся значения периодических реквизитов элемента справочника "Сотрудники". Для удаления значений непериодических реквизитов справочника опишем необходимые действия в процедуре ОбработкаУдаленияПроведения () (листинг 14.6).

#### Листинг 14.6

```
Процедура ОбработкаУдаленияПроведения()
СпрСотрудники=СоздатьОбъект("Справочник.Сотрудники");
СпрСотрудники.НайтиЭлемент(Сотрудник);
СпрСотрудники.ДатаПриема = Дата(0);
СпрСотрудники.ПриказОПриеме = ПолучитьПустоеЗначение();
Попытка
СпрСотрудники.Записать();
Исключение
Сообщить("Неудачная попытка записи элемента "+
СокрЛП(СпрСотрудники.Наименование)+
". Возможно, объект заблокирован...","!");
СтатусВозврата(0);
Возврат;
КонецПопытки;
```

КонецПроцедуры

Основное назначение документа "Начало месяца", как уже описывалось ранее, занести в журнал расчетов для каждого сотрудника записи по видам расчетов, необходимые для нормального режима расчета зарплаты. Нормальный режим — режим расчета основных начислений и удержаний без учета вводимых в течение периода промежуточных или дополнительных расчетов: оплата по среднему, дополнительные начисления и удержания. Результаты этих записей равны нулю, записи имеют признак нерассчитанных.

В модуле формы документа прописываем необходимые предопределенные процедуры, как в предыдущем документе. Модуль формы выглядит при этом следующим образом (листинг 14.7).

#### Листинг 14.7

Наличие общей переменной ЖрнЗарплата позволяет обращаться к параметру журнала НачалоТекущегоПериода() при установке даты документа. Этой установкой даты в пределах текущего расчетного периода мы регламентируем ввод документа, а значит, и записей, только в текущем расчетном периоде.

В зависимости от условий заполнения реквизитов **Подразделение** и **Сотрудник** при проведении документа будут созданы записи журнала расчетов с нулевым результатом по выбранному сотруднику или по сотрудникам выбранного подразделения. Для каждого сотрудника будет создана запись с основным начислением по окладу и основным удержанием по подоходному налогу.

Записи будут вводиться только сотрудникам, работающим в текущем расчетном периоде.

При повторном вводе документа "Начало месяца" в расчетном периоде у нас не должны создаваться новые записи журнала расчетов, т. е. система должна учесть наличие аналогичных записей в журнале расчетов.

Для невытесняющих друг друга видов расчета будем использовать метод ввода записей в журнал расчетов Записать Расчет(). Этот метод позволяет не создавать дублирующих записей в том же расчетном периоде *(см. разд. 14.2)*.

Прокомментируем модуль проведения (листинг 14.8) для описанных условий. Перед процедурами в модуле проведения объявим общую переменную ЖрнЗарплата для доступа к ее атрибутам в процедурах.

Перем ЖрнЗарплата;

В процедуру проведения, во-первых, добавляем проверку даты документа и даты начала расчетного периода, так как при перепроведении документа не в своем периоде будут удалены результаты расчетов, а также могут произойти перерасчеты взаимосвязанных записей и другие неприятности, связанные с настройкой расчетов. В нашем примере ничего подобного произойти не должно, но регламент необходимо соблюдать.

Комментировать модуль будем, разбив его на части.

### Листинг 14.8 Процедура ОбработкаПроведения() //Здесь следует написать алгоритм проведения документа // общая проверка на возможность проведения документа Если ДатаДок<>ЖрнЗарплата.НачалоТекущегоПериода() Тогда Сообщить ("Документ нельзя перепроводить задним числом!","!!!"); СтатусВозврата(0); Возврат; КонецЕсли;

После этого можно начинать расчеты в зависимости от назначенного реквизита Сотрудник (листинг 14.9).

#### Листинг 14.9

```
Если Сотрудник.Выбран()=1 Тогда

// Если по выбранному сотруднику

ВвестиПоСотруднику(Сотрудник);

Иначе

// по всем сотрудникам

СпрСотрудники=СоздатьОбъект("Справочник.Сотрудники");

СпрСотрудники.ВыбратьЭлементы();

Пока СпрСотрудники.ПолучитьЭлемент()=1 Цикл
```

Сотр=СпрСотрудники.ТекущийЭлемент();

Для каждого элемента справочника "Сотрудники" проверяем принадлежность к подразделению, выбранному в документе (листинг 14.10). Реквизит **Подразделение** является периодическим, поэтому мы воспользуемся методом получить () для выборки значения, действующего на конец расчетного периода.

#### Листинг 14.10

```
// проверим - есть ли ограничения на подразделение
Если (Подразделение.Выбран()=1) и
(Сотр.Подразделение.Получить(ЖрнЗарплата.КонецТекущегоПериода())<>
Подразделение) Тогда
// сотрудник не участвует в расчете
Продолжить;
КонецЕсли;
ВвестиПоСотруднику(Сотр);
КонецЦикла;
Конецесли;
```

```
ЖрнЗарплата= СоздатьОбъект ("ЖурналРасчетов.Зарплата");
```

Действия по выводу видов расчета в журнал расчетов выведены в отдельную процедуру ВвестиПоСотруднику(), обращение к которой происходит в двух местах модуля, но с разными параметрами (см. листинги 14.9 и 14.10).

Процедуру располагаем в модуле проведения выше процедуры Обработка-Проведения().

В процедуре (листинг 14.11) вначале проверяем, работал ли сотрудник в текущем расчетном периоде. Если не работал, тогда прекращаем работу процедуры.

Если сотрудник работал в текущем расчетном периоде, необходимо определить, с какой даты он работал, для правильного расчета оплаты по окладу за фактически отработанное время.

#### Листинг 14.11

```
// ввод записей по сотруднику
```

Процедура ВвестиПоСотруднику (ВыбранСотр)

// сотрудник работает?

Если (ВыбранСотр.ДатаПриема<ЖрнЗарплата.НачалоТекущегоПериода()) и (ВыбранСотр.ДатаПриема>ЖрнЗарплата.КонецТекущегоПериода()) Тогда Сообщить ("Сотрудник не работает в текущем периоде!","!"); Возврат;

КонецЕсли;

Если (ВыбранСотр.ДатаПриема>=ЖрнЗарплата.НачалоТекущегоПериода()) и (ВыбранСотр.ДатаПриема<=ЖрнЗарплата.КонецТекущегоПериода()) Тогда ДатаС=ВыбранСотр.ДатаПриема;

КонецЕсли;

Для работающих сотрудников продолжаем обработку в процедуре и выводим записи в журнал расчетов по окладу и по подоходному налогу (листинг 14.12).

#### Листинг 14.12

```
// выводим записи в журналы расчетов
ЖрнЗарплата.ЗаписатьРасчет (ВыбранСотр,ВидРасчета.ПоОкладу,
ДатаС, ЖрнЗарплата.КонецТекущегоПериода(),0);
ЖрнЗарплата.ЗаписатьРасчет (ВыбранСотр,ВидРасчета.ПодоходныйНалог,
ЖрнЗарплата.НачалоТекущегоПериода(),
ЖрнЗарплата.КонецТекущегоПериода(),0);
```

КонецПроцедуры

Как видно по листингу 14.12 в методе Записать Расчет () последним параметром указано значение 0, т. е. документ создает записи с нулевыми результатами. Период действия записей определяется датой начала (ЖрнЗарплата.НачалоТекущегоПериода ()) и датой окончания (ЖрнЗарплата.КонецТекущегоПериода ()). Период регистрации записей будет определяться датой документа.

Перейдем к документу "Выплата зарплаты", который будет использоваться для расчета суммы, подлежащей выплате сотрудникам, и для формирования печатной формы документа. Этот документ также должен отразить в журнале расчетов выплаченные суммы. Результаты вводимых в журнал расчета записей не равны нулю, поскольку сумма будет выплачена и изменена быть не может. Результат рассчитывается в документе как разность между суммой начислений и суммой удержаний. Традиционно конструктором добавляется процедура ПриОткрытии () (см. листинг 14.1). Добавляем процедуру Вводнового () для начального заполнения документа (листинг 14.13). Также в модуле формы в начале перед процедурами задаем общую переменную ЖрнЗарплата и присваиваем ей значение методом СоздатьОбъект () после всех процедур (листинг 14.13).

#### Листинг 14.13

```
Перем ЖрнЗарплата;
//
Процедура ВводНового(Копирование)
Если Копирование = 0 тогда
ДатаДок = РабочаяДата();
ОкончательныйРасчет=1;
КонецЕсли;
КонецПроцедуры
ЖрнЗарплата= СоздатьОбъект("ЖурналРасчетов.Зарплата");
```

Для заполнения табличной части документа и расчета сумм к выплате в модуле формы добавим процедуру Заполнить (), отрабатывающую по кнопке Заполнить (листинг 14.14). При заполнении табличной части элементами справочника "Сотрудники" элементы проверяются на соответствие следующим условиям:

**П** Сотрудник не уволен на начало расчетного периода;

**П** Сотрудник принят на работу до конца расчетного периода;

□ Сотрудник работает в выбранном подразделении.

Если сотрудник соответствует всем вышеперечисленным условиям, тогда мы создаем новую строку табличной части и присваиваем реквизиту **Сотрудник** значение текущего элемента справочника (листинг 14.14).

#### Листинг 14.14

```
Процедура Заполнить ()
СпрСотрудники=СоздатьОбъект ("Справочник.Сотрудники");
СпрСотрудники.ВыбратьЭлементы();
Пока СпрСотрудники.ПолучитьЭлемент()=1 Цикл
Сотр=СпрСотрудники.текущийЭлемент();
Если (Сотр.ДатаУвольнения=Дата("...")) или
```

(Сотр.ДатаУвольнения>=ЖрнЗарплата.НачалоТекущегоПериода()) Тогда
Если (Сотр.ДатаПриема<=ЖрнЗарплата.КонецТекущегоПериода()) Тогда
Если Сотр.Подразделение.Получить (ЖрнЗарплата.КонецТекущегоПериода())=
Подразделение Тогда
НоваяСтрока();
Сотрудник=Сотр;
КонецЕсли;
КонецЕсли;
КонецЕсли;
КонецЦикла;
КонецПроцедуры

При проведении документа с помощью кнопки **Ввести расчеты** в журнал расчетов будут введены записи с нулевым результатом для окончательной выплаты или с результатом, указанным в табличной части документа.

Переходим к написанию модуля проведения.

В случае выплаты заработной платы нам необходимо вывести записи в журнал расчетов. Результат у этих записей нулевой, и расчет суммы будет производиться в момент выполнения алгоритма вида расчета ВыплатаЗарплаты.

Период действия записей может не совпадать с текущим расчетным периодом, поскольку зарплата, как правило, выплачивается за предыдущий период.

В нашем примере рассматриваются два варианта выплаты заработной платы: промежуточная выплата и окончательный расчет.

Для промежуточной выплаты период действия записи будет равен периоду регистрации. Периодом, за который выплачивается зарплата при окончательном расчете, будет являться месяц, предшествующей месяцу даты документа.

Промежуточных выплат может быт несколько, а окончательная выплата в расчетном периоде только одна.

Для ввода записей по выплате заработной платы в журнал расчетов можно использовать только метод ВвестиРасчет() *(см. разд. 14.2)*, поскольку в одном периоде могут присутствовать две или более записи с одинаковыми видами расчетов, например, промежуточная выплата и окончательный расчет.

Модуль проведения в этом случае выглядит, как представлено в листингах 14.15, 14.16. Прокомментируем листинги. Перед вводом записей в журнал расчетов проверяем дату документа на принадлежность текущему расчетно-

му периоду, так как пересчитывать выплату можно только в текущем расчетном периоде (см. листинг 14.15).

#### Листинг 14.15

```
Процедура ОбработкаПроведения()

ЖрнЗарплата= СоздатьОбъект("ЖурналРасчетов.Зарплата");

//Здесь следует написать алгоритм проведения документа

Если (ДатаДок < ЖрнЗарплата.НачалоТекущегоПериода())

Или (ДатаДок > ЖрнЗарплата.КонецТекущегоПериода()) Тогда

Предупреждение("Документ не принадлежит текущему расчетному

|периоду журнала Зарплата

|Документ не проводится!");

СтатусВозврата(0);

Возврат;

КонецЕсли;
```

После проверки документа обрабатываем каждую строку табличной части (листинг 14.16) и в зависимости от режима выплаты устанавливаем значение в журнал расчета.

#### Листинг 14.16

```
ВыбратьСтроки();
Пока ПолучитьСтроку()=1 Цикл
Если ОкончательныйРасчет=1 Тогда
КВыплате=0;
Иначе
КВыплате=Сумма;
КонецЕсли;
ЖрнЗарплата.ВвестиРасчет (Сотрудник,ВидРасчета.ВыплатаЗарплаты,
ЖрнЗарплата.НачалоТекущегоПериода(),
ЖрнЗарплата.КонецТекущегоПериода(),КВыплате);
КонецЦикла
КонецПроцедуры
```

Результаты выплаты зарплаты связаны с учетом денежных средств, поэтому необходимо обеспечить неизменность результата при изменении начислений. Для этого дополним документ алгоритмом закрепления рассчитанного результата.

В экранную форму документа введем кнопку **Рассчитать**, при нажатии которой происходит вызов процедуры Рассчитать () для записей, введенных документом в журнал расчета.

Для расчета записей, введенных в журнал расчетов этим документом, воспользуемся методом Выбрать ЗаписиПоДокументу(). Сформировав выборку этим методом, организуем последовательную обработку записей, получая доступ к ним методом Получить Запись ().

Каждую запись выборки рассчитываем методом Выполнить Расчет (). Если метод по какой-либо причине не выполняется, прерываем выполнение процедуры.

Для того чтобы при расчете записей журнала расчета не было обращений к ним, например, других пользователей, занимающихся расчетом зарплаты, воспользуемся методом обработки транзакции. С этой целю перед началом массовой обработки записей вводим оператор процедуры Начать Транзакцию ().

Для фиксирования изменений данных при нормальном расчете всех записей выборки применим процедуру Зафиксировать Транзакцию().

В листинге 14.17 представлена процедура по расчету записей.

Для формирования печатной формы документа и написания процедуры печати воспользуемся конструктором: меню **Конструкторы** | **Печать...**.



Рис. 14.31. Конструктор печати документа: Шапка



Рис. 14.32. Конструктор печати документа: Таблица



Рис. 14.33. Конструктор печати документа: Подвал



Рис. 14.34. Конструктор печати документа: Дополнительные

На первом шаге конструктора — Шапка (рис. 14.31) — определяем реквизиты, которые будем выводить в заголовке отчета.

На следующем шаге — **Таблица** (рис. 14.32) — выводим реквизиты для таблицы отчета.

Итоговые значения отчета устанавливаются на шаге конструктора Подвал (рис. 14.33).

В последней форме настройки Дополнительные задаются режимы вывода таблицы и вызова процедуры печати (рис. 14.34).

В результате работы конструктора мы получили дополнительно в модуле формы процедуру печати (листинг 14.18).

#### Листинг 14.18

//{{ПРОЦЕДУРА ПЕЧАТИ(Печать)

//Данный фрагмент построен конструктором.

```
//При повторном использовании конструктора внесенные вручную изменения //будут потеряны!!!
```

Процедура Печать ()

```
Таб = СоздатьОбъект ("Таблица");
```

```
Таб.ИсходнаяТаблица ("Печать");
```

```
Таб.ВывестиСекцию("Шапка");
```

Таб.Опции (0,0,0,0);

ВыбратьСтроки();

```
Пока ПолучитьСтроку() = 1 Цикл
```

```
Таб.ВывестиСекцию("Строка");
```

КонецЦикла;

```
Таб. ВывестиСекцию ("Подвал");
```

Таб.ТолькоПросмотр(1);

```
Таб.Показать ("Печать Выплата заработной платы", "");
```

#### КонецПроцедуры

//} }ПРОЦЕДУРА\_ПЕЧАТИ

//\*\*\*\*\*\*

В связи с тем, что в печатную форму должны попадать результаты из журнала расчетов, изменим выборку данных для вывода на печать. Вместо операторов выборки строк табличной части

```
ВыбратьСтроки();
Пока ПолучитьСтроку() = 1 Цикл
```

введем операторы, выбирающие записи из журнала расчетов и формирующие переменные для таблицы печати (листинг 14.19).

#### Листинг 14.19

```
ЖрнЗарплата.ВыбратьЗаписиПоДокументу (ТекущийДокумент());
НомерСтроки=0;
ИтогСумма=0;
Пока ЖрнЗарплата.ПолучитьЗапись() = 1 Цикл
НомерСтроки=Номерстроки+1;
Сотрудник=ЖрнЗарплата.Объект;
Сумма=ЖрнЗарплата.Результат;
ИтогСумма=ИтогСумма+Сумма;
```

Сформированную конструктором печатную форму дополняем, выведя печать итоговой суммы прописью, добавив в конце документа подписи руководителей, а в табличной части колонку для подписи сотрудников. Поработав над размещением информации в печатной форме, получим следующий вид таблицы (рис. 14.35).

		1	2	3	4	5	6	7
	1							
Шапка	2		<Пл	атежная ведомость № [Номе]	оДок] от	[ДатаДок]	>	
	3				<за [ЖрнЗарпла	ата.ПериодПоДа	ге(ЖрнЗарі	плата.Н
	4			< [Подразделение.Наименование]>				
	5		≺Вид вь	платы: [?(ОкончательныйРасчет=1, "Окончательный	расчет", "Промеж	(уточная выплат:	 \$")]>	
	6							
	8		N⊆	Сотрудник	Сумма	Подпись		
	9							
Строка	10		Строки≻	<Сотрудник.Наименование>	<cyмма></cyмма>		į	
	11							
Подвал	12			Итого:	< [ИтогСумма]>			
	13							
	14			<Руководитель : /		⊳		
	15							
	16							

Рис. 14.35. Таблица печатной формы документа "Выплата зарплаты"

В экранной форме документа для удаления строк табличной части у реквизита диалога кнопки **Очистить** опишем действия, выполняемые при нажатии на нее (рис. 14.36).

Таким образом, мы вплотную подошли к написанию алгоритмов видов расчетов.

💻 Форма-Документ.ВыплатаЗарплаты	_ 🗆 🗵
👼 Выплата заработной платы	<b>_</b>
Выплата заработной платы № НомерДок от ДатаДок Подразделение Подразделение Вид выплаты О Окончательный расчет Свойства Кнопки О Промежуточная выплат Общие Дополнительно Команда Картинка Описание Формула: УдалитьСтроки()	×
Пропускать при вводе	ОК Отмена Обновить
Лиалог Диалог Диало	• •

Рис. 14.36. Свойства кнопки Очистить

При расчете зарплаты модули видов расчетов будут выполняться в порядке приоритетов, поэтому начнем их создавать в том же порядке.

Для вычисления результата вида расчета **ПоОкладу** нам необходимо знать количество отработанных дней и сумму оклада.

Используя методы работы с объектами **Календарь**, определяем по Сотруднику количество его рабочих дней для начисления суммы по окладу. Количество этих дней определяется от даты начала периода по дату конца, указанных в параметрах записи журнала расчетов. Количество дней определяется по календарю сотрудника, указанного для его графика работы в справочнике "Сотрудники" (листинг 14.20).

```
Листинг 14.20
```

```
Процедура ПровестиРасчет()
```

// Процедура выполняется при проведении расчета

ЖрнЗарплата = СоздатьОбъект("ЖурналРасчетов.Зарплата");

Если Объект.ГрафикРаботы.Получить (ЖрнЗарплата.КонецТекущегоПериода (

)).Календарь.Выбран()=0 Тогда

	Сообщить ();		
	Возврат;		
	Иначе		
	Календ=Объект.ГрафикРаботы.Получить (ЖрнЗарплата.КонецТекущегоПериода (		
))	)).Календарь;		
	КонецЕсли;		
	Дней = Календ. Дней (ДатаНачала, ДатаОкончания);		

Далее определяем общее количество дней в расчетном периоде по календарю графика работы сотрудника (листинг 14.21).

Листинг 14.21				
ВсегоДней = Календ.Дней(НачалоПериодаПоДате(ДатаНачала),				
КонецПериодаПоДате (ДатаНачала));				

Получаем оклад из периодического реквизита **Оклад** объекта расчета. Рассчитываем сумму по окладу, если определено общее количество рабочих дней в расчетном периоде, иначе — весь период был нерабочим и результат равен нулю (листинг 14.22).

Листинг 14.22

```
Оклад=Объект.Оклад.Получить (ЖрнЗарплата.КонецТекущегоПериода());
Если ВсегоДней>0 Тогда
Результат=Окр(Оклад*Дней/ВсегоДней);
Иначе
Результат=0;
КонецЕсли;
КонецПроцедуры
```

Для вычисления результата подоходного налога в модуле вида расчета **Подо**ходный Налог нам необходимо рассчитать сумму начислений в расчетном периоде, подлежащую налогообложению. Ставка подоходного налога фиксирована и указана в константе.

Модуль вида расчета с комментариями представлен в листингах 14.23-14.25.

Для расчетов определим необходимые переменные (листинг 14.23): дату начала месяца, дату конца месяца, начальное значение накопленной суммы за период.

#### Листинг 14.23

```
Процедура ПровестиРасчет()

// Процедура выполняется при проведении расчета

ЖрнЗарплата = СоздатьОбъект("ЖурналРасчетов.Зарплата");

СуммаМес=0;

КонМес=КонецПериодаПоДате(ДатаНачала);

НачМес=НачалоПериодаПоДате(ДатаНачала);
```

Для суммирования результатов начислений выбираем все записи по объекту, относящиеся к периоду действия с определенными датами начала и конца периода методом Выбрать ЗаписиПообъекту () (листинг 14.24).

#### Листинг 14.24

ЖрнЗарплата. ВыбратьЗаписиПоОбъекту (Объект, НачМес, КонМес);

Далее в цикле просматриваем все записи, попавшие в выборку. Суммируем результаты по видам расчета, принадлежавшим группе расчетов **Облагает-сяНалогом**. По полученному результату рассчитываем сумму налога.

#### Листинг 14.25

Пока ЖрнЗарплата.ПолучитьЗапись()>0 Цикл

Если ЖрнЗарплата.ВидРасч.ВходитВГруппу (ГруппаРасчетов.ОблагаетсяНалогом)

=1 Тогда

СуммаМес=СуммаМес+ЖрнЗарплата.Результат;

КонецЕсли;

КонецЦикла;

Результат=Окр (СуммаМес\*Константа.СтавкаПН.Получить (КонМес)/100);

КонецПроцедуры

В модуле вида расчета **ВыплатаЗарплаты** при вычислении результата будем рассчитывать сумму начислений в расчетном периоде, сумму удержаний в расчетном периоде и сумму выплат через кассу за расчетный период.

Для подсчета суммы начислений нам необходимо организовать выборку по журналу расчета. Можно воспользоваться одним из двух методов *(см. разд. 14.2)* для выборки записей по конкретному сотруднику: ВыбратьЗаписиПо-Объекту () ИЛИ ВыбратьПериодПоОбъекту ().

В первом случае мы выберем записи, период действия которых совпадает с выбранным для работы периодом журнала расчетов.

Обращаем внимание, что при более сложных расчетах заработной платы при использовании этого метода могут существовать записи, не учтенные при анализе. Например, доплата за январь, введенная в феврале после выплаты заработной платы. В документ выплаты заработной платы за январь эта сумма *еще* не попадает, а в документ выплаты зарплаты за февраль *уже* не попадает (другой анализируемый период).

В случае использования второго метода — Выбрать Период ПоОбъекту () в расчете будут участвовать записи, которые введены в выбранном периоде журнала расчетов, т. е. их период регистрации совпадет с периодом расчетов. В нашем примере доплата за январь будет выплачена в феврале.

В листинге 14.26 представлен модуль вида расчета с комментариями.

#### Листинг 14.26

Процедура ПровестиРасчет()			
// Процедура выполняется при проведении расчета			
ЖрнЗарплата= СоздатьОбъект("ЖурналРасчетов.Зарплата");			
Если Документ.ОкончательныйРасчет=1 Тогда			
Результат=0;			
Итог=0;			
// рассчитываем результат итоговой выплаты и для этого:			
//выбираем все записи расчетного периода по сотруднику,			
// который находится в рассчитываемой строке			
ЖрнЗарплата.ВыбратьПериодПоОбъекту(Объект,ДатаНачала);			
Пока ЖрнЗарплата.ПолучитьЗапись()=1 Цикл			
// Итог к выплате собирается как сумма всех начислений			
// за минусом всех удержаний			

И

Если (ЖрнЗарплата.ВидРасч.ВходитВГруппу(Группарасчетов.Удержания)

=1) Тогда

Итог=Итог-ЖрнЗарплата.Результат;

ИначеЕсли

(ЖрнЗарплата.ВидРасч.ВходитВГруппу (Группарасчетов.Начисления)

=1) Тогда

Итог=Итог+ЖрнЗарплата.Результат;

КонецЕсли;

КонецЦикла;

// выбираем записи сотрудника по выплатам в расчетный период ЖрнЗарплата.ВыбратьЗаписиПоОбъекту(Объект,

ДатаНачала, ДатаОкончания);

Пока ЖрнЗарплата.ПолучитьЗапись()=1 Цикл

// Итог к выплате собирается как сумма всех начислений и

// за минусом всех удержаний и

//за минусом всех выплат

Если ЖрнЗарплата.ВидРасч=

ВидРасчета.ВыплатаЗарплаты Тогда

Итог=Итог-ЖрнЗарплата.Результат;

КонецЕсли;

КонецЦикла;

// к выплате может быть сумма 0 или положительная

Итог=Мак(,Окр(Итог));

Результат=Итог;

ИначеЕсли Документ.ОкончательныйРасчет=2 Тогда

// промежуточную выплату не рассчитываем, т.к.

// она уже рассчитана при проведении документа

Возврат;

КонецЕсли;

КонецПроцедуры

Прокомментируем выборку записей для расчета суммы к выплате.

Расчет суммы к выплате производится только в случае указания в документе режима окончательного расчета (Окончательный Расчет=1). Для выборки записей с начислениями и удержаниями используется метод Выбрать Период-

ПоОбъекту() для выборки всех записей, зарегистрированных в расчетном периоде.

После расчета сумм к выплате, как разности между начислениями и удержаниями, выбираются виды расчетов с выплатами за расчетный период, поэтому используем метод ВыбратьЗаписиПоОбъекту(), чтобы получить список всех выплат, относящихся к указанному периоду.

В остальном комментарии в модуле помогают понять логику построения программы.

После написания модулей документов и видов расчета можно перейти непосредственно к написанию модуля расчета заработной платы, который предназначен для получения результатов по введенным в журнал расчетов записям.

Расчет заработной платы происходит при исполнении команды Рассчитать. Эта команда приводит в действие процедуру ПровестиРасчет для тех видов расчетов, ссылки на которые присутствуют в записях журнала расчетов за текущий расчетный период.

Для расчета заработной платы текущего расчетного периода создадим обработку и опишем алгоритмы ее работы.

Необходимо помнить, что рассчитать можно только записи, уже введенные в журнал расчетов. Это означает, что в течение текущего периода должен быть создан и проведен документ "Начало месяца".

Расчет заработной платы производится не документом, а обработкой, поскольку фиксировать и записывать уже ничего не надо. Основная задача обработки — запустить обработку расчета.

Новая обработка создается через меню Действия | Новый элемент.

В обработке присутствуют следующие элементы диалога (рис. 14.37):

**Сотрудник** — элемент справочника "Сотрудники";

**Подразделение** — элемент справочника "Подразделения".

Алгоритм расчета зарплаты представлен в листинге 14.27. Прокомментируем основные принципы его работы.

Для выборки записей, которые подлежат расчету, воспользуемся методом Выбрать Период () *(см. разд. 14.2)*. В этом случае в выборку попадут те записи, которые введены в выбранном периоде журнала расчетов.

Если в диалоге формы выбран сотрудник, воспользуемся методом Выбрать-ПериодПоОбъекту () для выборки записей по конкретному сотруднику.

💻 Форма-Обработка.Расчет	
📮 Расчет заработной плат	
Строка(ЖрнЗарплата	
Сотрудник. Сотр	
Подразделение : Под	
Выполнить Закрыть	
1	
🔪 🗐 Диалог 🛛 🖉 Моду	

Рис. 14.37. Экранная форма обработки Расчет зарплаты

Для обработки полученной выборки с записями журнала расчетов сначала позиционируемся на конкретной записи методом Получить Запись (). Полученную запись рассчитываем с помощью метода Выполнить Расчет ().

### Листинг 14.27 Перем ЖрнЗарплата, ДатаНачалаПериода, ДатаКонцаПериода; Процедура Выполнить () Док=СоздатьОбъект ("Документ.НачалоМесяца"); Если Док.ВыбратьДокументы()=0 Тогда // документов нет Предупреждение ("Не найден документ начала месяца! |Заработная плата не может быть рассчитана!"); Возврат; КонецЕсли; Если Сотрудник.Выбран()=1 Тогда ЖрнЗарплата.ВыбратьПериодПоОбъекту (Сотрудник,ДатаНачалаПериода); Иначе ЖрнЗарплата.ВыбратьПериод(ДатаНачалаПериода); КоненЕсли:

При написании модуля расчета заработной платы использованы знания, полученные при написании модулей в рамках сквозного примера.

Сохраните конфигурацию и проверьте работоспособность алгоритмов в режиме "1С:Предприятие".

Глава 15



## Отчетность в компоненте "Расчет"

Как мы уже знаем из предыдущих глав (см. разд. 11 и 13), существуют варианты отчетов, которые сложно, долго и нецелесообразно создавать с помощью методов, рассмотренных в этих разделах. В таких случаях рекомендуется использовать запросы.

# 15.1. Создание запросов по журналам расчетов

Рассмотрим пример создания отчета по журналу расчетов с помощью конструктора, вызываемого из меню Конструкторы и его пункта Новый отчет... (рис. 15.1).

При переходе конструктора к следующему шагу укажем необходимость вывода в меню команды вызова отчета и далее, в знакомой уже нам форме, укажем необходимость использования конструктора запросов, который начнет работу после создания объекта конфигурации **Отчет**. При этом после окончания работы конструктора нового отчета открывается окно конструктора запроса с формой для задания имени запроса (рис. 15.2).

Последовательность прохождения конструктора по шагам описана в *разд.* 13.2. Поэтому будем комментировать только основные настройки каждого шага.

После задания имени запроса переходим к шагу **Период**, где задаем правила определения периода запроса.

Конструктор Отчета
Отчет (в составе ко
C Внешний отчет
Идентификатор: 🔟
Синоним:
Комментарий:
Имя файла:
Конструктор позволя Внешний отчет (распо
Обязательно укажите Внешнего стиста. Пос
списках отчетов для і
Использовать коно
< Назад Да.

Рис. 15.1. Конструктор отчета

朝		
Новый запрос		
Ввелите имя заплоса:		
bbqg/ric kink/ calipsed.		
Сформировать Закрыть		

Рис. 15.2. Задание имени нового запроса

Указав правила выбора периода запросов, на следующем шаге **Переменные** создаем внутренние переменные запроса (рис. 15.3):

- □ сотр включает в выборку всех сотрудников, встречающихся как объекты в журнале расчетов Зарплата за период выборки;
- ВидР включает в выборку все виды расчета из журнала расчетов Зарплата за период выборки;



Рис. 15.3. Создание переменных запроса

Рез — включает в выборку все результаты из журнала расчетов Зарплата за период выборки.

Перейдя к следующему шагу работы конструктора, определим функции, используемые в запросе. Для переменной запроса Рез зададим функцию суммирования (рис. 15.4).

Задав необходимые в запросе функции, определяем способ и порядок группировки данных. При назначении нескольких группировок возможно организовать последовательную обработку результатов запроса. При этом необходимо учитывать, что при циклической обработке результатов запроса сначала производится обращение к группировкам вышестоящих уровней. Пример настройки группировок приведен на рис. 15.5.

После описания группировок запроса можно определить ограничивающие условия по включению в выборку данных на следующем шаге конструктора (рис. 15.6):

- □ включить в выборку только те виды расчетов, которые входят в группу расчетов **ОблагаютсяНалогом**;
- включить в выборку только те результаты, которые имеют ненулевое значение.
| Ka | нструктор З        | апросов:Фу     |  |  |  |
|----|--------------------|----------------|--|--|--|
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    | -                  | DeeConnue      |  |  |  |
|    | Функция            | гезсумма       |  |  |  |
|    |                    |                |  |  |  |
|    |                    | -              |  |  |  |
|    |                    | Сумма          |  |  |  |
|    |                    | , -            |  |  |  |
|    |                    |                |  |  |  |
|    | I когда (          |                |  |  |  |
|    |                    | 1              |  |  |  |
|    |                    |                |  |  |  |
|    | <b>.</b> .         |                |  |  |  |
|    | <u>Функция Рез</u> | Lymma = Lymmal |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
| -  |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |
|    |                    |                |  |  |  |

Рис. 15.4. Функции запроса



Конструктор Запросов : Усл		
🗖 Элементарное условие		
Условие ( (ВР.ВходитВГруп		
Условие(Рез > 0);		
Условие((ВР.ВходитВГруппу(Г		



Конструктор Запросов : Теі	
Период с {КонПерио; Сомр = ЖурналРасчел	
BugP = ЖурналРасчел Рез = ЖурналРасчелс	
Функция РезСумма = Группировка Сомр уг	
9словие(Pes > 0); Условие((BP.BxogumE	
•	
🔽 Генерировать процедуру	
🔽 Генерировать таблицу	
🔽 Удалять существующую т	

В результате работы конструктора запросов на последнем шаге мы получили следующий текст запроса (рис. 15.7).

Определив потребность в формировании печатной формы отчета, мы получаем таблицу для вывода печатной формы и процедуру для формирования запроса и вывода результатов в таблицу, алгоритм формирования печатной формы отчета по результатам запроса.

## 15.2. Примеры. Создание отчета "Расчетная ведомость"

Отчет, рассматриваемый в рамках примера, предназначен для вывода на печать данных о рассчитанной заработной плате и выплате ее за определенный период.



Рис. 15.8. Запрос для расчетной ведомости

Для формирования расчетной ведомости необходимо знать начисления и удержания по каждому сотруднику. При формировании запроса воспользу-

емся конструктором запросов, работа с которым описана в *разд. 15.1.* Выполняя шаги в соответствии с постановкой задачи, получаем следующий вид запроса (рис. 15.8).

После создания конструктором текста запроса нам остается указать режимы создания процедур модуля, а также назначить, в случае необходимости, кнопку в форме диалога для вызова нашей процедуры.

В диалоге формы отчета оптимизируем размещение реквизитов, добавленных по условиям отбора в запросе (Сотрудник и Подразделение) и для выбора периода запроса. Эти реквизиты включены в форму диалога конструктором.

Форма диалога может выглядеть так, как показано на рис. 15.9.



Рис. 15.9. Форма диалога отчета "Расчетная ведомость"

Выведенные конструктором реквизиты выбора периода формирования отчета заменим возможностью выбора периода, что более соответствует методике ведения расчетов компоненты "Расчет".

В форму диалога введем реквизиты, как показано на рис. 15.10, и прокомментируем их.

Элемент диалога типа "Текст" между кнопками < и > содержит формулу в свойствах ЖрнЗарплата.ОписательПериода (ДатаОтчета).

Метод ОписательПериода () возвращает строковое написание расчетного периода даты, указанной в качестве параметра.

Кнопки навигации по периодам позволяют:

Указать предшествующий период по формуле МинусПериод (ЖрнЗарплата, ДатаОтчета) в свойствах кнопки (рис. 15.11); > — указать последующий период по формуле ПлюсПериод (ЖрнЗарплата, ДатаОтчета) в свойствах кнопки (см. рис. 15.11).



**Рис. 15.10.** Реквизиты для выбора периода отчета

💻 Форма-Отчет. РасчетнаяВедом	
🛱 Расчетная ведомость	
Расчетная	
< тлата	
Свойства Кнопки	
Общие Дополнительно Команда	
Формула: МинусПериод(ЖрнЗарг	
🔲 Пропускать при вво	

Рис. 15.11. Свойства кнопок изменения периода

Для изменения периода отчета необходимо изменить значение переменной **ДатаОтчета** для формулы с методом ОписательПериода().

При движении назад с использованием методов мы уменьшаем значение ДатаОтчета на 1 день и получаем дату предыдущего периода, поскольку изначально ДатаОтчета — это первое число периода (листинг 15.1). Используя метод НачалоПериодаПодате(), получаем первое число расчетного периода.

### Листинг 15.1

// процедуры предназначены для изменения периода формирования отчета Процедура МинусПериод(ЖрнЗарплата,ДатаОтчета)

```
ДатаОтчета = ЖрнЗарплата.НачалоПериодаПоДате (ДатаОтчета-1);
КонецПроцедуры
```

При движении вперед с использованием методов мы увеличиваем значение последнего числа расчетного периода, полученного методом конецПериода-подате(). Увеличиваем дату на 1 день и получаем дату следующего периода, которая будет использована при описании периода (листинг 15.2).

### Листинг 15.2

```
Процедура ПлюсПериод(ЖрнЗарплата,ДатаОтчета)
ДатаОтчета = ЖрнЗарплата.КонецПериодаПоДате(ДатаОтчета)+1;
КонецПроцедуры
```

Для доступа в формулах элементов диалога переменных жрнЗарплата и ДатаОтчета необходимо объявить в модуле формы. Поэтому введем в начальных строках модуля формы оператор объявления этих переменных:

Перем ЖрнЗарплата;

Перем ДатаОтчета;

В модуле формы после расположения всех процедур разместим операторы определения этих переменных для доступности их значений во всех процедурах модуля формы и в формулах реквизитов диалога:

```
ЖрнЗарплата = СоздатьОбъект ("ЖурналРасчетов.Зарплата");
```

ДатаОтчета = ЖрнЗарплата.НачалоТекущегоПериода();

Для создания окончательного вида печатной формы, созданной конструктором (рис. 15.12), видоизменим ее, как мы уже описывали в *разд. 13.3*.

Поясним внесенные изменения по шагам:

- 1. В секции Заголовок в верхней строке отменяем режим объединения.
- 2. Удалим лишние столбцы и добавляем при необходимости новые столбцы.

- 3. В секции Заголовок в верхней строке назначаем режим объединения.
- 4. В секции Заголовок в верхней строке вводим название отчета и форматируем шрифт.
- 5. В секции Заголовок во второй строке назначаем режим объединения.
- 6. В секции Заголовок в верхней строке вводим название периода и форматируем шрифт (рис. 15.13).

R21C7			= ₩-			
		1	2	3	4	5
	1					
Заголовок	2					
	3		Сотр Наименование	Начислено	Удержания	
Итого	4		Итого	прос.Начислено#N019.2>	прос.Удержания#N019.2>	
Сотр	5		<Запрос.ЗначениеУпорядочивани	прос.Начислено#N019.2>	прос.Удержания#N019.2>	
	6					
	7					
	8					
	9					
	10					
	11					

Рис. 15.12. Форма таблицы отчета "Расчетная ведомость", созданная конструктором

	1		2	3	4		5
1							
2				Расчетная вед	домость		
3		•	<за [ЖрнЗарг	лата.Описателы	Периода(Да	таОтчет	a)]>
4		C	Свойства ячейки				x выплате
5				- x - x - x x	_,		ос.Начислено-3
6		<Запрос.Зн≀	Текст Положени	ю Шрифт Рамка Узор			ос.Начислено-3
7			Тип Шабари	📕 Контроль Алло			
8			тин Гшаолон				
9			за [ЖрнЗарплата	а. ОписательПериода(Дата	Отчета)] 📃		
10						OK	1
11					<b>v</b>		
12			Provinterative D			0	1
13			гасшифровка				
14			🔽 Защита			Обновить	1
15						Соновить	J



6. В секции Заголовок в строке с шапкой таблицы изменим названия колонок и форматируем шрифт.

На рис. 15.14 представлен вариант формы отчета после редактирования.

R20C6			-µ =								
		1	2	3	4	5	6	7			
	1										
Заголовок	2			Расчетная ведомость							
	3		<за [ЖрнЗарг	<за [ЖрнЗарплата.ОписательПериода(ДатаОтчета)]>							
	4		Сотрудник	Начисления	Удержания	К выплате					
Итого	5		Итого	прос.Начислено#N019.2>	прос.Удержания#N019.2>	<(Запрос.Начислено-З	апрос.Удер	жания)#∿			
Сотр	6		<Запрос.ЗначениеУпорядочивани	прос.Начислено#N019.2>	прос.Удержания#N019.2>	<(Запрос.Начислено-З	апрос.Удер	жания)#∿			
	7										



Процедура печати для рассматриваемого примера достаточно проста и мы не вносим в нее никаких изменений (листинг 15.3).

### Листинг 15.3

```
// Процедура генерации запроса Сформировать.
Процедура Сформировать ()
   Перем Запрос, ТекстЗапроса, Таб;
   КонПериода = ЖрнЗарплата.КонецПериодаПоДате (ДатаОтчета);
   //Создание объекта типа Запрос
   Запрос = СоздатьОбъект ("Запрос");
   ТекстЗапроса =
   "//{{ЗАПРОС(Сформировать)
   Период с ВыбНачПериода по ВыбКонПериода;
   |Сотр = ЖурналРасчетов.Зарплата.Объект;
   |BP = ЖурналРасчетов.Зарплата.ВидРасч;
   |Рез = ЖурналРасчетов.Зарплата.Результат;
   |Подр = ЖурналРасчетов.Зарплата.Объект.Подразделение;
   |Функция Начислено = Сумма (Рез)
            когда (BP. ВходитВГруппу (ГруппаРасчетов. Начисления) =1);
   Функция Удержания = Сумма (Рез)
                когда (BP. ВходитВГруппу (ГруппаРасчетов. Удержания) =1);
   Группировка Сотр упорядочить по Сотр. Наименование;
   |Условие (Рез > 0);
```

```
|Условие ( (ВР. ВходитВГруппу (ГруппаРасчетов. Начисления) =1)
              или (BP=ВходитВГруппу (ГруппаРасчетов.Удержания)=1));
   |Условие (Сотр в ВыбСотр);
   |Условие (Подр в ВыбПодр);
   |"//}}ЗАПРОС
  ;
   // Если ошибка в запросе, то выход из процедуры
   Если Запрос.Выполнить (ТекстЗапроса) = 0 Тогда
      Возврат;
   КонецЕсли;
   // Подготовка к заполнению выходных форм данными запроса
  Таб = СоздатьОбъект ("Таблица");
   Таб.ИсходнаяТаблица ("Сформировать");
   // Заполнение полей "Заголовок"
   Таб. ВывестиСекцию ("Заголовок");
   Состояние ("Заполнение выходной таблицы...");
   Таб.Опции(0, 0, Таб.ВысотаТаблицы(), 0);
   Пока Запрос. Группировка (1) = 1 Цикл
      // Заполнение полей Сотр
       Таб. ВывестиСекцию ("Сотр");
   КонецЦикла;
   // Заполнение полей "Итого"
  Таб.ВывестиСекцию ("Итого");
   // Вывод заполненной формы
  Таб.ТолькоПросмотр(1);
   Таб.Показать ("Сформировать", "");
КонецПроцедуры
```

Структура модуля, создаваемая конструктором, рассмотрена нами в *разд. 13.2.* На этом мы заканчиваем рассмотрение первого примера по компоненте "Расчет".

## 15.3. Примеры. Создание отчета "Свод по заработной плате"

Данный отчет предназначен для вывода сведений о суммах начислений по видам расчетов за определенные периоды. Диапазон периодов выбирается в диалоге формы.

При этом мы выбираем в запрос записи, *зарегистрированные* в выбранном диапазоне периодов. Если бы нам необходимо было выбрать записи, рассчитанные для выбранного периода, в том числе относящиеся к прошлым периодам, мы бы выбирали в запрос записи по **периоду действия**.

При формировании запроса снова воспользуемся конструктором запросов, работа с которым помогает заменить ввод основных операторов вручную. Выполняя шаги в соответствии с постановкой задачи, обратим внимание на шаг создания переменных (рис. 15.15).

Конструктор Запросов : Пер		
Имя переменной: ПериодРе		
ЖурналРасчетов.Зарплата.Г		
•		
ВидРасч = ЖурналРасчетов, Результат = ЖурналРасчето ПериодРегистрации = Журн		

Рис. 15.15. Создание переменных запроса отчета "Свод по заработной плате"



Рис. 15.16. Создание группировок запроса отчета "Свод по заработной плате"

Конструктор Запросов : Тек		
Период с ВыБНамПери		
ВидРасч = ЖурналРас		
Результат = ЖурналF		
Функция РезульмамСу		
Группировка BugPacy		
i pgilliupooka ilepuogr		
4		
<b>—</b> -		
І енерировать процедуру		
Генерировать таблицу		
🔽 Удалять существующую т		

Рис. 15.17. Результат работы конструктора запроса отчета "Свод по заработной плате"

Одна из переменных запроса — ПериодРегистрации — введена для выборки записей по принадлежности к периоду регистрации. В качестве доступного для выбора атрибута журнала расчетов мы видим ПериодДействия (рис. 15.15).

В зависимости от условий задачи конструирования запроса мы можем использовать различные временные характеристики для отбора записей журнала расчетов (см. разд. 14.2): ПериодРегистрации, ПериодДействия, ДатаНачала, ДатаОкончания.

Согласно постановке задачи определяем группировки запроса: по видам расчета и по периодам действия (рис. 15.16).

В результате работы конструктора получаем следующий вид запроса (рис. 15.17).

Результаты работы конструктора преобразованы также в виде модуля формы отчета, показанного в листинге 15.4.

### Листинг 15.4

```
// Процедура генерации запроса Сформировать.
11
Процедура Сформировать ()
   Перем Запрос, ТекстЗапроса, Таб;
   //Создание объекта типа Запрос
   Запрос = СоздатьОбъект ("Запрос");
   ТекстЗапроса =
   "//{{ЗАПРОС(Сформировать)
   Период с ВыбНачПериода по ВыбКонПериода;
   |ВидРасч = ЖурналРасчетов.Зарплата.ВидРасч;
   |Результат = ЖурналРасчетов.Зарплата.Результат;
   ПериодРегистрации = ЖурналРасчетов.Зарплата.ПериодРегистрации;
   |Функция РезультатСумма = Сумма (Результат);
   |Группировка ВидРасч;
   |Группировка ПериодРегистрации;
   |"//}}ЗАПРОС
   // Если ошибка в запросе, то выход из процедуры
   Если Запрос.Выполнить (ТекстЗапроса) = 0 Тогда
     Возврат;
   КонецЕсли;
```

```
// Подготовка к заполнению выходных форм данными запроса
   Таб = СоздатьОбъект ("Таблица");
   Таб.ИсходнаяТаблица ("Сформировать");
   // Заполнение полей "Заголовок"
   Таб.ВывестиСекцию("Заголовок");
   Состояние ("Заполнение выходной таблицы...");
   Таб.Опции(0, 0, Таб.ВысотаТаблицы(), 0);
   Пока Запрос. Группировка (1) = 1 Цикл
      // Заполнение полей ВидРасч
      Таб.ВывестиСекцию("ВидРасч");
      Пока Запрос. Группировка (2) = 1 Цикл
         // Заполнение полей ПериодРегистрации
         Таб. ВывестиСекцию ("ПериодРегистрации");
      КонецЦикла;
   КонецЦикла;
   // Заполнение полей "Итого"
   Таб.ВывестиСекцию("Итого");
   // Вывод заполненной формы
   Таб. ТолькоПросмотр (1);
   Таб.Показать ("Сформировать", "");
КонецПроцедуры
```

Таблица отчета, созданная конструктором, представлена на рис. 15.18.

R18C8			<b>▼</b> -⊠	=		
		1	2	3	4	5
	1					
Заголовок	2					
	3		ВидРасч	иодРегистра	РезультатСумма	
Итого	4		Итого	<3anpoc.l	езультатСумма#N019.2>	
ВидРасч	5		<Запрос.В	<3anpoc.l	езультатСумма#N019.2>	
ПериодРели	6		<Запрос.В	<Запрос.Пе	езультатСумма#N019.2>	
	7					
	8					

Рис. 15.18. Таблица отчета "Свод по заработной плате"

Экранная форма диалога отчета показана на рис. 15.19.

鄭	
Период с: ВыбНач	
<<ПериодСтр(ВыбНа	
Сформировать	

Рис. 15.19. Форма диалога отчета "Свод по заработной плате"

Мы не будем комментировать результаты работы конструктора, а перейдем сразу к изменениям.

Для понимания изменений в процедуре формирования отчета изменим структуру таблицы отчета.

Поясним вносимые изменения по шагам.

- 1. В секции Заголовок в верхней строке отменяем режим объединения.
- 2. Удаляем столбец ПериодРегистрации и расширяем столбец ВидРасч.
- 3. В секции Заголовок в левой ячейке вносим название отчета, расположив его слева, и форматируем шрифт (рис. 15.20).
- 4. В секции Заголовок в строке с шапкой таблицы изменяем названия колонок и форматируем шрифт. При этом вместо названия второй колонки **РезультатСумма** задаем выражение для вывода описателя периода (рис. 15.21).

Вывод результатов будет производиться в секции **ПериодРегистрации** следующим образом:

- □ название вида расчета в соответствующей колонке;
- в той же строке последовательно по периодам суммы начислений по виду расчета;
- □ в последней колонке итоговая сумма за весь диапазон периодов по виду расчета.

		ВилВ	Периол		
		оидг	период		
	1	2	3		4
1	L		L		
2		Свод по заработ	ной плате		
4					X
5					
6		Текст Положение Шрифт	Рамка Узор		9 0
7			- · · ·		
8		Тип Текст 💌 Контр	оль Авто		
9					
10		свод по заработной плате			
11					ок
12				-	
13		Расшифровка			Отмена
14		🔽 Защита			
15		и защига			Обновить
16					



	1	2	3		4				
1									
2		Свод по заработ							
4		Виды расчета	<Описатель>						
5		Свойства ячейки 🗵							
7 8		Текст Положение Шрифт	Рамка   Узор		? Ø				
9		Тип Выражение 💌 Контроль Обрезать 💌 Описатель							
10 11									
12				-	OK				
13 14		Расшифровка		-	Отмена				
15 16		🗹 Защита			Обновить				

Рис. 15.21. Свойства ячейки таблицы отчета для назначения названия колонки

Для циклического вывода в таблицу колонки с результатами начислений выделим в таблице вертикальную секцию следующим образом:

- 1. Выделим нужный столбец таблицы.
- 2. Выберем пункт меню Таблица | Включить в секцию;
  - 3. В открывшемся окне (рис. 15.22) задаем название секции Период.

			Секция_5	
	1	2	3	4
1				
2		Свод по зараб	отной плате	
4		Виды расчета	<Описатель>	
5		Итого	езультатСумма#N019.2>	
6		<Запрос.ВидРасч>		
7		<Запрос.ВидРасч>	Идентификатор секции	×
8			<b>_</b>	
9			Период	OK
10				0
11				Отмена

Рис. 15.22. Создание секции таблицы

Теперь создадим еще одну вертикальную секцию **ВидР** для начальных колонок, для того чтобы можно было к ним обратиться в модуле отчета.

Удаляем горизонтальные секции (строки) ВидРасч и Итого.

Изменяем выражение в ячейке вывода результата на значение переменной, которая будет формироваться в модуле отчета (рис. 15.23).

	1	2	3		4				
1									
2		Свод по заработ							
4		Виды расчета	<Описатель>						
5		<Запрос.ВидРасч>	<u>ззультатСумма#N019.2&gt;</u>						
6		Свойства ячейки		×					
7		·	·	-					
8		Текст Положение Шрифт Рамка Узор 🧖							
9									
10									
11		Запрос_РезультатСумма#I	A						
12		+			οκ 1				
14		t		<u> </u>					
15		Расшифровка		_	Отмена				
16		22000 ga							
17		ј• Защига			Обновить				

Рис. 15.23. Свойство ячейки таблицы с результатом

На рис. 15.24 представлен вариант формы отчета после редактирования.

В форму диалога внесем только изменения по расположению реквизитов и добавим элемент с названием отчета (рис. 15.25).

			ВидР	Период	
		1	2	3	4
	1				
Заголовок	2		Свод по заработной плате		
	4		Виды расчета	<Описатель>	
ПериодРеги	5		<Запрос.ВидРасч>	езультатСумма#N019.2>	
	6				

Рис. 15.24. Таблица отчета "Свод по заработной плате" после редактирования

舅 Свод		
Св	),	
пл	<b>.</b>	
с ВыбНач	le	
Сформировать		

Рис. 15.25. Окончательная форма диалога отчета "Свод по заработной плате"

Учитывая особенности компоненты "Расчет", определим формулы в диалоге для реквизитов начала и окончания периода выборки таким образом, чтобы дата начала выборки всегда была равна первому числу месяца (начало расчетного периода), а дата конца выборки — последнему числу месяца (конец расчетного периода).

Для этого в свойствах реквизитов на вкладке **Дополнительно** указываем соответствующие операторы:

- ВыбНачПериода=НачМесяца (ВыбНачПериода) для реквизита ВыбНачПериода;
- ВыбКонПериода=КонМесяца (ВыбКонПериода) для реквизита ВыбКонПериода.

Внесем в модуль формы знакомые уже нам операторы определения общей переменной жрнЗарплата в начале модуля и оператор присваивания переменной значения после всех процедур (листинг 15.5). Сразу же добавим процедуру ПриОткрытии() для заполнения дат периода выборки данных: даты начала текущего расчетного периода и даты конца текущего расчетного периода соответственно (листинг 15.5).

### Листинг 15.5

ЖрнЗарплата = СоздатьОбъект ("ЖурналРасчетов.Зарплата");

Переходим к изменениям алгоритмов в процедуре формирования отчета. Основные изменения будут производиться для вывода результатов начислений. Конструктор создал форму отчета с выводом результатов запроса в виде строк (см. рис. 15.18).

Изменим программный модуль таким образом, чтобы результаты выводились в одной строке, путем добавления колонок с результатами по периодам.

После комментария // Заполнение полей "Заголовок" (см. листинг 15.4) добавляем операторы формирования колонок заголовка в зависимости от выбранного диапазона периодов.

Первоначально объявим переменную Список для хранения списка всех периодов диапазона отчета. Этот список необходим для того, чтобы выводить пустые ячейки в случае отсутствия результатов начислений в этом периоде.

Затем выведем строку и колонку с названием вида расчета секции Заголовок. Использование метода ВывестиСекцию() позволяет добавить новую строку секции Заголовок и колонки ВидР в этой строке (листинг 15.6).

### Листинг 15.6

```
// Заполнение полей "Заголовок"
```

```
Список=Создать<br/>Объект ("СписокЗначений") ;
```

```
Таб.ВывестиСекцию("Заголовок|ВидР");
```

Потом определим дату из первого периода диапазона отчета для вычисления названия колонки:

ДатаКон=ВыбНачПериода;

Организуем последовательный вывод всех возможных периодов диапазона в цикле, пока не превысим дату конца выборки.

1. Определяем переменную для вывода названия колонки:

Описатель=ЖрнЗарплата.ОписательПериода (ДатаКон);

 Сохраняем в списке периодов дату конца периода, по которому уже выведена колонка:

Список.ДобавитьЗначение (КонМесяца (ДатаКон));

 К выведенной секции с названием вида расчета добавляем колонку следующего периода. При этом метод Присоединить Секцию() означает добавление секции в той же строке:

Таб.ПрисоединитьСекцию("Заголовок|Период");

 Рассчитываем возможную дату следующего периода как последнее число расчетного периода плюс один день:

ДатаКон=ЖрнЗарплата.КонецПериодаПоДате (КонМесяца (ДатаКон))+1;

 Переходим на начало цикла и проверяем, не превысили ли мы дату конца выборки:

КонецЦикла;

6. После вывода колонок для всех периодов выводим колонку для итогов, в которой название задаем текстом, и присоединяем секцию в той же строке:

```
Описатель="Итого по виду расч.";
```

Таб.ПрисоединитьСекцию("Заголовок | Период");

В связи с разделением строк на горизонтальные секции заменим строку (листинг 15.4)

```
Таб.ВывестиСекцию("ВидРасч");
```

на оператор

Таб. ВывестиСекцию ("ПериодРегистрации | ВидР");

Внутри первой группировки обрабатываются значения по группировке **ПериодРегистрации**. Мы уже оговорили ситуацию отсутствия в периоде начислений по видам расчета, поэтому по каждому результату периода будем проверять, к какому периоду относится результат запроса. Перед началом выборки присвоим номеру элемента списка значение 1 и начнем выборку:

```
Номер=1;
Пока Запрос.Группировка(2) = 1 Цикл
```

Теперь проверим, соответствует ли дата окончания периода регистрации из запроса дате окончания периода из списка:

Если Запрос.ПериодРегистрации.ДатаОкончания >

Список.ПолучитьЗначение (Номер) Тогда

Если получен период более дальний (с пропуском), то мы выводим пустые ячейки (Запрос\_РезультатСумма=0), пока не доходим до нужного периода (листинг 15.7).

### Листинг 15.7

```
ДатаСлед=Запрос.ПериодРегистрации.ДатаОкончания;
Пока Список.ПолучитьЗначение(Номер)<ДатаСлед Цикл
Запрос_РезультатСумма=0;
Таб.ПрисоединитьСекцию("ПериодРегистрации|Период");
//ДатаСлед=КонМесяца(ДатаСлед)+1;
Номер=Номер+1;
КонецЦикла;
КонецЕсли;
```

После вывода пустых ячеек выводим ячейку с результатами запроса и, перейдя к следующему периоду списка периодов, изменим значение переменной номер (листинг 15.8).

### Листинг 15.8:

```
// Заполнение полей ПериодРегистрации
Запрос_РезультатСумма=Запрос.РезультатСумма;
Таб.ПрисоединитьСекцию("ПериодРегистрации|Период");
Номер=Номер+1;
КонецЦикла;
```

Закончив обработку результатов по группировке **ПериодРегистрации**, мы можем оказаться в ситуации, когда в последних периодах нет результатов начислений. Мы не сможем вывести пустые ячейки внутри цикла. (В этой ситуации после цикла по второй группировке выводим оставшиеся пустые ячейки, если количество выведенных периодов (Homep -1) меньше размера списка переменной Список (листинг 15.9)).

### Листинг 15.9

Если (Номер-1)<Список.РазмерСписка() Тогда Пока (Номер-1)<Список.РазмерСписка() Цикл Запрос\_РезультатСумма=0; Таб.ПрисоединитьСекцию("ПериодРегистрации|Период"); Номер=Номер+1; КонецЦикла; КонецЕсли;

После вывода пустых ячеек присоединяем строку с итогами и заканчиваем цикл обработки результатов запроса (листинг 15.10).

### Листинг 15.10

```
Запрос РезультатСумма=Запрос.РезультатСумма;
```

```
Таб. Присоединить Секцию ("Период Регистрации | Период");
```

КонецЦикла;

После обработки результатов запроса оставляем операторы вывода таблицы и установки режима вывода (листинг 15.11).

### Листинг 15.11

```
// Вывод заполненной формы
Таб.ТолькоПросмотр(1);
Таб.Показать("Свод по периодам", "");
КонецПроцедуры
```

При разработке сквозного примера нами рассматривалась идеальная ситуация для ознакомления с возможностями программного продукта "1С:Предприятие". При желании можно самостоятельно усложнить некоторые решения, внеся изменения в алгоритмы.

Сохраните конфигурацию и проверьте работоспособность алгоритмов в режиме **1С:Предприятие**.

Глава 16



# Некоторые дополнительные сведения

Система программ "1С:Предприятие" является открытой. Это значит, что возможно осуществлять обмен информацией между различными информационными базами "1С:Предприятие" либо между системой "1С:Предприятие" и другими программными продуктами. В главе приводится пример экспорта данных из "1С:Предприятия" во внешние файлы и обмен данными с Microsoft Excel.

## 16.1. Экспорт данных с помощью файлов

Для связи с другими программами могут использоваться встроенные средства загрузки-выгрузки информации в текстовом формате и формате DBF. Подавляющее большинство программных продуктов позволяет импортировать данные с помощью этих форматов. Для работы с текстами в системе используется специальный агрегатный тип данных — "Текст". Средства языка имеют возможность не только вывода строк в текстовые файлы, но и считывания имеющихся файлов с последующим разбором его по строкам. Для работы с файлами в системе используется специальный агрегатный тип данных — "Хbase".

В любом случае для организации обмена данными может понадобиться изучить работу с файлами. Для работы с файлами в системе используется специальный агрегатный тип данных — "ФС".

Обращение к агрегатным типам данных стандартное — Создатьобъект ("ИмяОбъекта"). Для текстовых файлов и для файлов dbf актуально понятие Кодовой страницы: КодоваяСтраница(<Режим>). Кодовая страница

\_ 8 × \_ 🗆 X 📕 Кодировк - Блокнот Файл Правка Формат Вид Справка ГЪ ЎҐ«м 🔻 🛃 Переход . Ц«Ёвл ïa®ĭҐa€ ĭa®ўҐa€ 1 🎒 кодВи - Блокнот \_ 🗆 × Файл Правка Формат Вид Справка Кабель Ппиты проверка проверка 1 Документы - 1 🖬 SC65 Сетевое окружение 式 SC86 Корзина 🗟 SC86 Папка выгрузки Share-to-Web 로 SC88 network.assistant.4.0.build.2383 🖬 SC88 Setun 🗐 коаВи Военно-Исторический Форум ВИФ2 🗐 Кодирс Неиспользуемые ярлыки 🗊 проба 🗐 пртекс СетеварМарк • ÞÍ 🐮 Пуск 🧉 🏹 🕞 🔌 🗀 2 Проводник 🔹

определяет, в какой кодировке будут выводиться данные: Windows-кодировка — режим 0, DOS-кодировка — режим 1.

Рис. 16.1. Записи в Windows-кодировке и в DOS-кодировке

Рассмотрим некоторые возможности использования текстовых и DBFфайлов на примере выгрузки данных из справочника "Номенклатура". Выгрузку для пользователя организуем следующим образом: создадим в форме списка три флажка.

- **Кодовая страница** идентификатор КодС;
- **П Текст** идентификатор ЭкспТекст;
- □ \*.dbf идентификатор ЭкспДбф.

При установке первого флажка кодовая страница будет равна 1, т. е. информация выводится в DOS-кодировке, иначе — в Windows-кодировке. При установке второго флажка информация выводится в текстовый файл. При установке третьего флажка информация выводится в DBF-файл. Экспорт информации осуществляется при нажатии кнопки Экспорт (рис. 16.2).



Рис. 16.2. Кнопки для выгрузки данных справочника во внешние приложения

Откроем в Конфигураторе форму списка справочника "Номенклатура" и перейдем в модуль формы списка. Создадим новую процедуру Выгрузить ().

```
Листинг 16.1

СтрЗап=""; НомЗап=0;ИмяФайла="";

Спр=СоздатьОбъект ("Справочник.Номенклатура"); //Создана переменная для

//ссылки на объект "Справочник Номенклатура"

Если ЭкспТекст=1 Тогда

ВвестиСтроку (ИмяФайла, "Назовите текстовый файл", 8,);

Если КодС>0 тогда КодСтр=1 Иначе КодСтр=0 КонецЕсли;

Текст= СоздатьОбъект ("Текст");

Текст.Открыть (ИмяФайла);

Текст.КодоваяСтраница (КодСтр);

Если Спр.ВыбратьЭлементы()=1 Тогда // справочник спозиционирован
```

```
Пока Спр.ПолучитьЭлемент()=1 Цикл
НомЗап=НомЗап+1;
СтрЗап=Строка(СокрЛП(Спр.Наименование));
Текст.ДобавитьСтроку(СтрЗап);
Текст.Записать(ИмяФайла);
Состояние("Экспортировано "+НомЗап+" строк");
СтрЗап="";
КонецЦикла;
КонецЕсли;
```

Создаем объект "Справочник. Номенклатура" — хорошо знакомое действие. Дальше проверяем, активен ли флажок ЭкспТекст. Если пользователь установил этот флажок, вводим имя файла, используя переменную, объявленную в начале процедуры. После этого проверяем, какая кодовая страница используется для вывода информации, и присваиваем переменной КодСтр значение либо 1, либо 0, в соответствии с выбором пользователя — значением флажка Кодовая страница. С помощью переменной Текст мы создаем текстовый файл с ранее введенным именем. Вид кодовой страницы для объекта Текст содержится в переменной КодСтр.

Используем знакомую структуру выборки из справочника. Обратите внимание на операторы (листинг 16.2).

```
Листинг 16.2
```

```
Текст.ДобавитьСтроку (СтрЗап);
Текст.Записать (ИмяФайла);
```

Для того чтобы запись из справочника, хранимая в переменной **СтрЗап**, добавилась в текстовый файл, используется оператор Добавить Строку (). А для того чтобы строка сохранилась в текстовом файле, необходимо записывать (сохранять) файл каждый раз после добавления новой строки.

Если пользователь активизировал флажок ЭкспДбф, работает часть процедуры, выводящая записи справочника в файл \*.dbf.

### Листинг 16.3

```
ИначеЕсли ЭкспДбф=1 Тогда
ВвестиСтроку(ИмяФайла,"Назовите файл базы данных",8,);
Если КодС>0 тогда КодСтр=1 Иначе КодСтр=0 КонецЕсли;
База=СоздатьОбъект("XBase");
```

```
База.ДобавитьПоле ("Name", 2, 30, 0);
База.ДобавитьПоле("Price",1,10,2);
ИмяФайла=ИмяФайла+".dbf";
База. Создать Файл (Имя Файла);
База.ЗакрытьФайл();
База. Открыть Файл (Имя Файла);
  Если Спр.ВыбратьЭлементы()=1 Тогда // справочник спозиционирован
   Пока Спр.ПолучитьЭлемент()=1 Цикл
     База.Добавить (); //добавляет новую запись в файл
  База.УстановитьЗначениеПоля ("Name", Строка (СокрЛП (Спр. Наименование)));
//в новую запись, в поле "Name" заносит значение поля Наименование из
//справочника.
  База.УстановитьЗначениеПоля ("Price", Спр.Цена.Получить (РабочаяДата()));
//в новую запись, в поле "Price" заносит значение поля Цена
//из справочника.
      База.Записать(); //сохраняет запись в файле
   КонецЦикла;
  КонецЕсли;
   База.ЗакрытьФайл(); //после выборки из справочника все записи внесены
                        //и файл закрывается.
```

КонецЕсли;

КонецПроцедуры

Сначала мы описываем файл \*.dbf. Переменной База присваиваем ссылку на объект **ХВаse**. Описываем поля в файле (листинг 16.4).

### Листинг 16.4

```
База.ДобавитьПоле("Name",2,30,0);
База.ДобавитьПоле("Price",1,10,2);
```

Метод добавить Поле добавляет поле в описание структуры файла. Данный метод можно использовать только перед созданием новой базы. В скобки вносятся параметры поля:

- Строковое выражение имя создаваемого поля;
- □ <тип> строковое или числовое выражение тип создаваемого поля. Допустимые значения:
  - 1 или N –число;
  - 2 или S строка;

- 3 или D дата;
- 4 или L логическое;
- 5 или F то же, что и 1, т. е. число;
- 6 или М тето (не поддерживается);

Слина> — числовое выражение — общая длина создаваемого поля;

- «Точность» — числовое выражение — длина создаваемого поля после десятичной точки (только для числовых полей).

В нашем примере название поля — "Name", "2" указывает на то, что поле будет строкового типа, "30" — длина поля, "0" — точность. Поле текстовое, так что точность для этого поля — понятие, лишенное смысла.

В поле Price указывается точность, так как поле имеет числовой тип.

Далее добавляем к названию файла, содержащегося в переменной Имяфайла, расширение dbf, только теперь создаем файл с помощью функции База.СоздатьФайл (ИмяФайла); Файл тут же закрываем оператором База.ЗакрытьФайл ();

Теперь система может вносить новые записи в созданный файл. Порядок действий понятен из комментариев в листинге 16.3.

## 16.2. Совместная работа системы "1С:Предприятие" и электронных таблиц Microsoft Excel

Система "1С:Предприятие" поддерживает современные стандарты интеграции программ: DDE, OLE Automation. Одним из примеров использования этих возможностей может служить организация работы системы "1С:Предприятие" с распространенной офисной программой Microsoft Excel. В приведенном примере мы продолжим работать со справочником "Номенклатура". Загрузка записей справочника из Microsoft Excel будет осуществляться при нажатии кнопки **Excel**.

Рассмотрим алгоритм загрузки данных из файла Microsoft Excel. В модуль формы документа добавляем переменные Имяфайла и ИмяКаталога. Область использования этих переменных — весь модуль формы, поэтому в соответствии с правилами написания программных модулей эти переменные должны объявляться в первой строке модуля (листинг 16.5).

### Листинг 16.5

Перем ИмяФайла, ИмяКаталога;

### Листинг 16.6

Процедура ВыбратьФайл()

Файл=СоздатьОбъект ("ФС");

Если Файл.ВыбратьФайл(0, ИмяФайла, ИмяКаталога,

"Выберите файл, содержащий данные справочника",

```
"Файлы Microsoft Excel|*.xls|Все файлы|*.*", "xls")<>1 Тогда Возврат;
КонецЕсли;
```

ИмяФайла=ИмяКаталога+ИмяФайла;

КонецПроцедуры

Смысл процедуры — организовать диалог для поиска файла Microsoft Excel'97, из которого будет производиться загрузка в справочник. Для работы с файлами используется агрегатный тип данных "ФС", о котором говорилось в начале главы. В функции Имяфайла указываются параметры открытия диалогового окна: 0 — диалог типа "открыть", 1 — диалог типа "сохранить". Используем 0.

- «ИмяФайла» имя переменной, содержащей строковое значение с именем файла. В эту же переменную система возвращает имя выбранного файла.
- «ИмяКаталога» имя переменной, содержащей строковое значение с именем начального каталога. В эту же переменную система возвращает имя выбранного каталога.
- СаголовокОкна> строковое выражение, с помощью которого можно задать заголовок открываемого окна — Выберите файл, содержащий данные справочника.
- □ <Фильтр> строковое выражение, с помощью которого можно задать список фильтров отбора файлов — Файлы Microsoft Excel|\*.xls|Все файлы|\*.\*.
- СРасширение> строковое выражение, с помощью которого можно задать расширение файла по умолчанию, которое используется системой при записи файла. Используем "xls".

Процедура Загрузка () осуществляет непосредственно перенос значений из Microsoft Excel в справочник "Номенклатура". Поскольку наименование товара будет первоначально записываться (или проверяться на наличие) в справочнике "Номенклатура", в процедуре создается объект языка Справочник. Номенклатура. Далее необходимо создать объект языка Excell.Application (Приложение Excel). При работе с этим объектом используются операторы языка Visual Basic for Application (VBA).



Рис. 16.3. Лист Microsoft Excel с данными для справочника "Номенклатура"

Далее следует написать функцию, которая позволяет получать значения отдельной ячейки листа Microsoft Excel с использованием VBA.

### Листинг 16.7

Функция ПолучитьЗначение(Excel,y,x)

```
Cell=Excel.Cells(y,x);
```

Возврат Cell.Value;

КонецФункции

Это стандартная функция, которую можно применять в любой программе, осуществляющей загрузку данных из Microsoft Excel. В соответствии с правилами написания модуля эта функция должна располагаться перед процедурой Загрузка().

Листинг 16.9 — текст процедуры Загрузка() с комментариями. Для подготовленного читателя этого достаточно. Но для лучшего понимания опишем ключевые моменты алгоритма.

Переменной і присваивается значение 1, если данные, которые необходимо загрузить в "1С:Предприятие", располагаются, начиная с первой строки. Если номер строки листа Microsoft Excel иной, переменной і присваивается соответствующее значение.

Загрузка данных будет производиться до тех пор, пока ячейка с координатами номер строки=I, номер колонки=1 будет заполнена. Как только ячейка будет пустой, загрузка прекратится. Поэтому используется системная функция "1С:Предприятия": ПустаяСтрока(), которая возвращает значение 1, если в ячейке нет никаких данных. На этом алгоритме основана конструкция, представленная в листинге 16.8.

### Листинг 16.8

```
Пока Пустаястрока(Стр)=0 Цикл
.....КонецЦикла;
```

Обратите внимание, что значения ячейки листа Microsoft Excel мы получаем с помощью описанной в предыдущем шаге функции Получить Значение ().

Внутри цикла проверяется, имеется ли товар, загруженный из листа Microsoft Excel, в справочнике "Номенклатура". Если такого товара нет, он записывается в справочник вместе со значениями остальных реквизитов.

Конечно, в данном примере отсутствует целый ряд полезных сервисных функций. Например, интерактивная настройка загрузки данных из ячеек Microsoft Excel. Однако мы сознательно опустили эти детали, чтобы сделать задачу более понятной. Кроме того, в частном случае при донастройке под конкретное предприятие можно обойтись и без этих сервисных функций.

### Листинг 16.9

```
Процедура Загрузка()
Спр=СоздатьОбъект("Справочник.Номенклатура"); //Создадим объект
//Справочник Номенклатура
```

384

```
Спр.ИспользоватьДату (РабочаяДата (), 1);
                              //найдем файл с документом Microsoft Excel
ВыбратьФайл();
Excel=CosgatbOGbert("Excel.Application");
                                            // создадим объект -
                                             //рабочая книга Excel
РабочаяКнига=Excel.Workbooks;
РабочаяКнига.Open(ИмяФайла,0,1);// откроем рабочую книгу - выбранный
                                 //файл Excel
  i=1; // с 1 строки начинаются данные в документе Microsoft Excel
   Стр=(СокрЛП(Строка(ПолучитьЗначение(Excel, i, 1))));
    Пока ПустаяСтрока (Стр) = 0 Цикл
                                              //пока непустые ячейки
                                              //в строках - загружаем
    Стр=(СокрЛП(Строка(ПолучитьЗначение(Excel, i, 1)));
      Если ПустаяСтрока(Стр)=1 Then //пустая ячейка - конец загрузки
       Сообщить ("Загрузка закончена", "i");
       Прервать;
      КонецЕсли;
//
              Добавляем элемент в справочник
 НомерСтроки=Стр;
ТоварСпр=(СокрЛП(Строка(ПолучитьЗначение(Excel,i,2)));
 Если Спр.НайтиПоНаименованию (ТоварСпр, 0)=0 Тогда // Если товара
                                        //в справочнике Номенклатура нет
    Спр.Новый ();
                               // тогла занесем его тула
Спр.Наименование=(СокрЛП(Строка(ПолучитьЗначение(Excel,i,2)));
Спр.Себестоимость= Число (Получить Значение (Excel, i, 3));
Спр.СтавкаНДС= Число (ПолучитьЗначение (Excel, i, 4));
КонецЕсли;
    Спр.Записать ();
```

і=і+1; КонецЦикла;

КонецПроцедуры

Если все сделано правильно, то при запуске конфигурации в пользовательском режиме при нажатии кнопки **Excel** справочник "Номенклатура" должен автоматически заполняться (рис. 16.4).

🖾 I С:Предприятие - I	Бух и опер учет, расчет - [Номенклатура (01.12.06)]			15 ×
🗐 Файл Действия 🤇	Операции Справочники Документы Журналы Результатырасчета С	этчеты Обработки Серви	с Окна Попощь	8  ×
1 🖙 🖩 👗 🖻	16 6 8 2 C A 2 C A 2 C A 3 C A	8 SIP ()	M M+ M-	
	·····································			
Код	Краткое наименование	Единица из	Тип номенклатуры	-
	3 проверка.			
	2 Tou2			- 61
	1 Toeap1			-81
				-8
				-81
				-2
				-8
				-81
				-8
	//			-81
	/			-8
	2050000000000			-81
	опруженные			-91
	Элементы			
				-81
				-81
				-81
				-21
				•
Ехсеі Печать	- Экспорт интроризации ЭкспортГ Текст Кодорая страница*dbl			
Номенклатура (0)	1.12			
для получения подсказк	и наконте F1	NUM TA: 29.09.06 15:2	9:36 БИ: 4 квартал 2006 г.   TTI: Октябрь 2006 г.	
<i>Ш</i> Пуск 慮 🔾 🔾	Э Проводник • 12 2 Microsoft Of • 44 3 10:47 start • )	😔 Cnpaska Windows 🛛 📰 F	Paint Shop Pro 🔟 🕐 🦿 🔍 🐨 🐨 🖉 🗐 🗃	16:28

Рис. 16.4. Загрузка справочника из Microsoft Excel в пользовательском режиме

Разберем противоположный случай — выгрузки данных из "1С:Предприятия" 7.7 в Microsoft Excel для последующего анализа.

Не секрет, что возможности управленческого учета, анализа и прогноза деятельности предприятия в стандартных конфигурациях "1С:Предприятие" невелики. Не отличаются разнообразием и средства визуализации информации (представление в виде диаграмм, интерактивная и программная возможность изменения типа диаграмм и выборки данных). Еще раз отметим, что "1С:Предприятие" — программа для учета, а не для анализа. Компенсировать этот недостаток можно с помощью Microsoft Excel, используя механизм OLE Automation.

Учитывая, что наша конфигурация проста и мы не сможем рассчитывать сложные управленческие задачи, ограничимся простым примером — выгрузим данные о продаже товаров в Microsoft Excel. Хочу отметить, что данный пример можно использовать в любых конфигурациях для решения аналогичных задач. Если же вы научитесь работать с специфическими объектами метаданных компонент "Оперативный учет" и "Бухгалтерский учет" — планом счетов, операциями, проводками, регистрами, — то использовать изучаемый здесь механизм можно будет с высокой эффективностью. Фактически "1С:Предприятие" будет использоваться по своему прямому назначению — учету и доставке результатов учета в аналитическую программу — Microsoft Excel, с помощью которой производится дальнейшая обработка информации.

В режиме конфигуратора создадим простой внешний отчет. Он создается так же, как было описано в *главе 8*. Единственное отличие — на первом шаге работы конструктора отчета поставьте переключатель в положение **Внеш**ний отчет, а не Отчет в составе конфигурации (рис. 16.5). С помощью конструктора запросов создадим запрос на выборку данных из документов "Реализация товаров" в процедуре Сформировать ().



Рис. 16.5. Отчет для экспорта данных в Microsoft Excel создан с помощью конструктора

#### Листинг 16.10

```
Процедура Сформировать()
Перем Запрос, ТекстЗапроса, Таб;
  //Создалим объект Excel
Excel = CosgatbOGbekt("Excel.Application");
//Ограничим книгу Excel одним листом
Excel.SheetsInNewWorkbook = 1;
//Установим заголовок книги
Excel.Caption = "Отчет по реализации товаров";
//Создадим новую рабочую книгу
NewBook = Excel.Workbooks.Add;
 СписокСтолбцов = СоздатьОбъект ("СписокЗначений");
 СписокСтолбцов. Добавить Значение ("Товар");
 СписокСтолбцов. Добавить Значение ("Месяц");
 СписокСтолбцов.ДобавитьЗначение ("Продано товара");
  СписокСтолбцов.ДобавитьЗначение("На сумму");
 //Заполним наименование столбцов: товар, месяц и т.п.
 Для Столбец = 1 По СписокСтолбцов. РазмерСписка () Цикл
  Ячейка = Excel.Cells(1,Столбец);
  Ячейка.Value = СписокСтолбцов.ПолучитьЗначение (Столбец);
КонецЦикла;
  //Создание объекта типа Запрос
  Запрос = СоздатьОбъект ("Запрос");
  ТекстЗапроса =
  "//{{ЗАПРОС(Сформировать)
  Период с ВыбНачПериода по ВыбКонПериода; //указываем временной период,
                                          //внутри которого строится отчет
```

ОбрабатыватьДокументы все; // используются все документы
Обрабатывать НеПомеченныеНаУдаление; // кроме помеченных на удаление
Говар = Документ.РеализацияТоваров.ТМЦ; // как и информация о товаре
Количество = Документ.РеализацияТоваров.Количество;// о количестве
// товара

```
|Сумма = Документ.РеализацияТоваров.Сумма;
                                                   // и сумме, на которую
                                                   // поставлен товар
  Функция КоличествоСумма = Сумма (Количество); //суммируем для подитогов
                                                 //и итогов
  Функция СуммаСумма = Сумма (Сумма);
  |Группировка Месяц;
  |Группировка Товар упорядочить по Товар.Наименование; // потом по
товарам
  |"//}}ЗАПРОС
  ;
  // Если ошибка в запросе, то выход из процедуры
  Если Запрос.Выполнить (ТекстЗапроса) = 0 Тогда
  Возврат;
  КонецЕсли;
  // Подготовка к заполнению выходных форм данными запроса
  Состояние ("Заполнение выходной таблицы...");
  НомерСтроки=1;
  Пока Запрос. Группировка (1) = 1 Цикл
    // Заполнение полей Товар
  Пока Запрос. Группировка (2) = 1 Цикл
   Товар = Запрос.ЗначениеУпорядочивания (2); // (1)
  Месяц = Запрос.Месяц;
   КоличествоСумма= Запрос.КоличествоСумма;
   СуммаСумма = Запрос.СуммаСумма;
   НомерСтроки = НомерСтроки + 1;
   //Организуем заполнение ячеек Excel
   Для Ст = 1 По СписокСтолбцов.РазмерСписка() Цикл
     Ячейка = Excel.Cells(НомерСтроки, Ст);
    Если Ст = 1 Тогла
     Ячейка.Value = Товар;
    ИначеЕсли Ст = 2 Тогда
     Ячейка.Value = Строка (Месяц);
    ИначеЕсли Ст = 3 Тогда
     Ячейка.Value = КоличествоСумма;
```

```
ИначеЕсли Ст = 4 Тогда
Ячейка.Value = СуммаСумма;
КонецЕсли;
КонецЦикла;
КонецЦикла;
КонецЦикла;
Excel.Visible = 1;
```

Excel = 0; КонецПроцедуры

Запрос выбирает данные по продажам каждого товара в определенном месяце выбранного периода. Разберем процедуру более подробно.

Начинаем работать с Microsoft Excel. Создадим объектную переменную для работы с книгой Excel и установим параметры книги (листинг 16.11).

### Листинг 16.11

```
//Создадим объект Excel
Excel = СоздатьОбъект("Excel.Application");
//Ограничим книгу Excel одним листом
Excel.SheetsInNewWorkbook = 1;
//Установим заголовок книги
Excel.Caption = "Отчет по реализации товаров";
//Создадим новую рабочую книгу
NewBook = Excel.Workbooks.Add;
```

Озаглавим столбцы листа Microsoft Excel, в которые предполагается передавать данные. Для этого создадим объект типа "Список значений" и занесем в него наименования столбцов. Затем с помощью цикла перенесем названия столбцов из списка значений на первую строку листа рабочей книги Microsoft Excel (листинг 16.12).

### Листинг 16.12

```
СписокСтолбцов = СоздатьОбъект ("СписокЗначений");
СписокСтолбцов.ДобавитьЗначение ("Товар");
СписокСтолбцов.ДобавитьЗначение ("Месяц");
```
```
СписокСтолбцов.ДобавитьЗначение ("Продано товара");
СписокСтолбцов.ДобавитьЗначение ("На сумму");
//Заполним наименование столбцов:
Для Столбец = 1 По СписокСтолбцов.РазмерСписка() Цикл
Ячейка = Excel.Cells(1,Столбец);
Ячейка.Value = СписокСтолбцов.ПолучитьЗначение(Столбец);
КонецЦикла;
```

Теперь организуем вывод данных помесячно (листинг 16.13). В данном фрагменте программы обратите внимание на заполнение ячеек таблицы Microsoft Excel.

#### Листинг 16.13

НомерСтроки=1;

```
Пока Запрос. Группировка (1) = 1 Цикл
    // Заполнение полей Товар
   Пока Запрос. Группировка (2) = 1 Цикл
   Товар = Запрос.ЗначениеУпорядочивания (2); // (1)
   Месяц = Запрос.Месяц;
   КоличествоСумма= Запрос.КоличествоСумма;
   СуммаСумма = Запрос.СуммаСумма;
   НомерСтроки = НомерСтроки + 1;
   //Организуем заполнение ячеек Excel
   Для Ст = 1 По СписокСтолбцов.РазмерСписка() Цикл
     Ячейка = Excel.Cells (НомерСтроки, Ст);
    Если Ст = 1 Тогда
     Ячейка.Value = Товар;
    ИначеЕсли Ст = 2 Тогла
     Ячейка.Value = Строка (Месяц);
    ИначеЕсли Ст = 3 Тогла
     Ячейка.Value = КоличествоСумма;
    ИначеЕсли Ст = 4 Тогда
     Ячейка.Value = СуммаСумма;
```

КонецЕсли;

КонецЦикла;

КонецЦикла;

КонецЦикла;

<b>1</b>	тчет по	реализа	ции това	ара-Книга1												_ 8 ×
:2	<u>Ф</u> айл	Правка	<u>В</u> ид В	ставка Формат	Сервис	Данные	PROMT	<u>О</u> кно	⊆правка					Введите в	onpoc	- 8 ×
1	12	l 🖪 🖨	1 🖪 🗋	1 🥙 🔊 - (°	- Σ -	Al I 🛍	100% -	2 14	vrial Cyr	- 10	- ж	К П ≣	등 등 强	<b>9</b> % 🚦	- 11	🗞 - 🗛 - 👔
AH	ло-Русск	ий	• Общи	й	- 🌚 🖷 🛙	3 🔚 I	🗊 🐼 🗉		a iii 🔬	8 🖸 🔍						
	R7C2	•	fx													× 7
	1		2		3		4	5	6	7	8	9	10	11	12	
1	Товар	Mec	ац	Продано	товара	Насум	му									
2		_														
4		-						-			_					
5																
6																
7				_											_	
8								-				_			-	
10																
11																
12																
13		_													_	
14		_						-				_			-	
16		_														
17		_														
18																
19																
20		_						-							_	
22		-						-			_				-	
23																
24																
25		_														
26																
27		_														
20		_														
30																
31																
32																
H 4	• • • • • • • • • • • • • • • • • • •	иист1/				3		4			•					
÷Де	йс <u>т</u> вия ▼	AB1	гофи <u>с</u> уры	• \ \ [](	اله 🖻 (	6° 🔒	🛛 🖓 -	<u>⊿</u> • <u>∔</u>	<b>\</b> - =	5 🖬 🗐	Ŧ					
Гото	60														NUM	
<b>#</b> 7	Іуск 🧝	) 🔄 🕑	» 🚞	3 Пров 🗕 👿 2	Micro +	<mark> 🦑 2</mark> 1C:	:V • 🤌	Справка	📆 Pain	t Sh 🧕 🥰 C	orelDR	🕙 Книга1	RU 🛛	🛱 « 12	<u> 10 32 K (</u>	) 🗐 🏾 16:50

Рис. 16.6. Результат работы фрагментов программы — листингов 16.12 и 16.13

В результате данные из расходных накладных помещены в таблицу Excel (рис. 16.7).

Обратите внимание на две последние строки процедуры.

Строка Excel.Visible = 1; делает окно Excel видимым и активным. Если ваша процедура в силу каких-либо причин не срабатывает до этой строки, то это может вызвать переполнение оперативной памяти компьютера открытыми, но невидимыми книгами MS Excel, что приведет к зависанию компьютера.

Строка Excel = 0; освобождает оперативную память компьютера, которая резервировалась для объектной переменной Excel.

<b>X</b> 0	тчет по	pea	ализации	товара - Кі	ига1												_ 8 ×
:	<u>Ф</u> айл	Пра	авка <u>В</u> ид	ц Вст <u>а</u> вка	Формат	Сервис	Данные	PROMT OK	ю <u>С</u> правка					BB	едите вопрос	•	- 8 ×
1	💕 😡			i 🛕 💝 I	<b>17 -</b> (*)	- Σ	- 🛔   🛄	100% 🔹	Arial Cyr		• 10 • 1	жкч		- 🔜 i 🥶	%	🗄 • 🔕 •	<u>A</u> - 🔋
AH	ло-Русск	ий	• (	Общий	-	🖽 🖽	💹 🔚 🛛	i 🐼 🛄 🔽	1 🛐 🗰 1	r 🛛 🖸	Q 😡 -					100	<u> </u>
	R1C1		-	∱ Товар								,				5	7
	1			2	3		4	5	6	7	8	9	10	11	12	1.	14
1	Товар	_	Месяц		Продано	товара	На сумму										
2	товарт		Сентярр	6 Ub		10	4	:50									
4		-															
5																	
6																	
7																	
8																	
10																	
11																	
12																	
13																	
14																	
16		-															
17																	
18																	
19																	
20																	
22																	
23																	
24		_															
25																	
20																	
28																	
29																	
30																	
31																	
H 4	► H	Лис	r1/								•						- FI
Де	йс <u>т</u> вия •	3	Автофи	Суры • 🗸	100	) 🚵 👍	1 🗘 🛽	🎿 i 🧆 🗸	• <u>A</u> • =	≡ <b>≓</b> ∅							
Гото	60															NUM	
de r	Іуск 🥖	5 🖸	) 🕑 👋	🗋 З Пров.	• 👿 2	Micro •	• 👯 3 1C:	V 🖌 🤣 Cnp	вка 📰	Paint Sh	🍯 CorelDR.	🛛 💌 OTY	ют п 🛛	2 :	SE 0E EI >>	K 🕲 🗐 1	<b>1</b> 6:51

Рис. 16.7. Лист Microsoft Excel с данными из системы "1С:Предприятие"

В заключение хотелось бы обратить внимание на следующие моменты:

- малоподготовленного пользователя не должно пугать большое количество строк кода программы. Значительная часть работы выполняется системой "1С:Предприятие" автоматически, с помощью конструкторов в конфигураторе;
- использование набора относительно простых отчетов для вывода информации в Microsoft Excel позволяет использовать все богатство аналитических инструментов и без особых финансовых затрат иметь эффективный бизнес-анализ.

# 16.3. Работа с метаданными с помощью встроенного языка

Язык "1С:Предприятие" позволяет работать не только с отдельными атрибутами метаданных, но и непосредственно с самой структурой метаданных конфигурации. Это может быть необходимо, например, для создания универсальных отчетов или обработок, которые обращаются к объектам различных видов (документам, справочникам и т. п.), если необходимо подсчитать количество реквизитов в справочнике или найти реквизит по идентификатору. Работа с метаданными выполняется через глобальный атрибут **Метаданные**, имеющий тип "Метаданные".

### Замечание.

Объект типа "Метаданные" нельзя создать, используя метод СоздатьОбъект.

Объект типа "Метаданные" имеет атрибуты для доступа к свойствам объекта метаданных и методы для доступа к массивам подчиненных объектов метаданных. Глобальный атрибут "Метаданные" предоставляет доступ к общим свойствам конфигурации и к основным объектам метаданных (документам, справочникам и т. п.). Полученный из глобального атрибута подчиненный объект метаданных, например, документ, предоставляет в свою очередь доступ к свойствам документа и к объектам метаданных, являющихся реквизитами документа.

Отметим, что в документации к системе "1С:Предприятие" не приводится полного описания работы с метаданными, поэтому приходится использовать информацию из других источников.

Для лучшего понимания назначения метаданных и способа работы с ними из встроенного языка используем следующий пример.

В *разд. 2.3* при описании свойств конфигурации указывается на свойство **Непосредственное удаление объектов**. Как вы помните, было рекомендовано не устанавливать этот флажок. Если же в какой-то момент работы с конфигурацией необходимо непосредственно удалять объекты, то сделать это можно только из встроенного языка (листинг 16.14).

### Листинг 16.14

ВыбМетодУдаления = Метаданные.НепосредедственноеУдалениеОбъектов;

Действительно, не может же пользователь в какой-то момент, исходя из логики программы, остановить работу "1С:Предприятия", войти в конфигуратор, установить нужный флажок и продолжить работу.

Рассмотрим работу с метаданными на примере обработки "Обработка документов". Это достаточно стандартный сервис в типовых конфигурациях, позволяющий проводить различные манипуляции (проведение, удаление и т. п.) сразу с целыми группами документов. В нашей конфигурации обработка сильно упрощена. Тем не менее, в ней присутствует код, позволяющий работать со списками документов. Это является иллюстрацией возможностей работы со списком значений. Разбор этого кода не входит в нашу задачу, сделайте это самостоятельно. В обработке, в пользовательском режиме, задается период, за который выбираются обрабатываемые документы, и из правого окна в левое окно переносятся виды обрабатываемых документов. При нажатии двойной стрелки в левое окно переносятся все документы, при нажатии одинарной — только отмеченный документ. Стрелки, направленные вправо, переносят документы из левого окна в правое, т. е. исключают их из списка обрабатываемых документов (рис. 16.8).

🎯 IC:Предприятие - Бух и опер учет, расчет 💦 🚺 🔀
Файл Действия Операции Справочники Документы Журналы Результатырасчета Отчеты Обработки Сервис Окна Помощь
2 2 日 米 助 記 示 係 面 つ と 執 2 月 系 系 ? 日 当 第 10 M M+ M-
С С С С С С С С С С С С С С С С С С С
[2] Обработка документов           Для выборь значення назначте клавншу F4,
🤰 Пуск 🥑 🍙 🕢 » 🛄 3 Пров • 📈 2 Місто • 🧖 3 1 С.Ү • 🤣 Справка 🗮 Paint Sh 🕼 CorelDR 🔟 Orver n 🔃 🕐 🦿 🤞 🖓 🗰 16:53

Рис. 16.8. Обработка документов в пользовательском режиме

Выполняет обработка только одно действие: составляет реестр документов. Код этой процедуры рассмотрим более подробно.

## Листинг 16.15

Процедура Выполнить (Обр)

```
ВидыДок = "";
       Для А=1 По ВыбранныеДокументы. РазмерСписка() Цикл
               ВидыДок = ВидыДок+Документы. Получить Значение
               (ВыбранныеДокументы.ПолучитьЗначение (А)) +", ";
       КонецЦикла;
       Если ПустаяСтрока (ВидыДок) = 1 Тогда
               Возврат;
       КонецЕсли:
               Док=СоздатьОбъект ("Документ");
               Док.ВыбратьДокументы (Дата1,Дата2);
       Если Обр="Печать" Тогда
               Таб=СоздатьОбъект ("Таблица");
"+СокрП (ВыбКонтрагент.Наименование) +".";
               Таб.ВывестиСекцию("Отчет");
       Иначе
               Начать Транзакцию ();
       КонецЕсли:
       H\Pi\Pi=0;
       НД=0; //сколько доков ВД
       Пока Док.ПолучитьДокумент()=1 Цикл
               //Проверка вида документа
               Если Найти (ВидыДок, Док.Вид()+", ")=0 Тогда
                       Продолжить;
               КонецЕсли:
               Состояние ("Обработка "+Док.ДатаДок);
               H\Pi\Pi = H\Pi\Pi + 1;
                       НД=НД+1;
                       ПечЗнак="";
                       РеквСум-
ма=Метаданные.Документ (Док.Вид()).РеквизитТабличнойЧасти
        ("Cymma");
```

та=Метаданные.Документ (Док.Вид()).РеквизитШапки ("Валюта"); Если РеквСумма.Выбран()=1 Тогда ПечСумма=? (РеквСумма.ИтогПоКолонке=1, Формат (Док.Итог ("Сумма"), "414.2-,"),""); Если РеквВалюта.Выбран()=1 Тогда // проверим, а есть ли валюта в документе ВД ЕстьВал=1; Если Док.Валюта=Константа.БазоваяВалюта Тогда ПечСуммаДляИтога=? (РеквСумма.ИтогПоКолонке=1,Док.Итог ("Сумма"), ""); Иначе ПечСуммаДляИтога=?(РеквСумма.ИтогПоКолонке=1,Док.Итог("Сумма"), ""); ПечСуммаДляИтога=ПечСуммаДляИтога\*Док.Курс; КонецЕсли; Иначе // валюты нет ЕстьВал=0; ПечСуммаДляИтога=? (РеквСумма.ИтогПоКолонке=1,Док.Итог ("Сумма"), ""); КонецЕсли; //проверим, а есть ли валюта //в документе ВД ПечИтогСумма=ПечИтогСумма+Число(ПечСуммаДляИтога); //ВД Иначе РеквСумма=Метаданные.Документ (Док.Вид()).РеквизитШалки ("Сумма"); Если РеквСумма.Выбран()=1 Тогда ПечСумма=Формат (Док.Сумма, "414.2-,"); Если РеквВалюта.Выбран()=1 Тогда // проверим, а есть ли валюта в документе ВД ЕстьВал=1;

РеквВалю-

Если

Док.Валюта=Константа.БазоваяВалюта Тогда

ПечСуммаДляИтога= Док.Сумма; //ВД

Иначе

ПечСуммаДляИтога= Док.Сумма; //ВД

ПечСуммаДляИто-

га=ПечСуммаДляИтога\*Док.Курс;

КонецЕсли;

Иначе // валюты нет

ЕстьВал=0;

ПечСуммаДляИтога= Док.Сумма; //ВД

КонецЕсли; //проверим, а есть ли //валюта в документе ВД

ПечИтогСумма=ПечИтогСумма+Число(ПечСуммаДляИтога); //ВД

КонецЕсли;

КонецЕсли;

РеквКлиент=Метаданные.Документ (Док.Вид()).РеквизитШапки

("Клиент");

Если РеквКлиент.Выбран()=1 Тогда

ПечКлиент="Клиент: "+СокрЛП (Док.Клиент);

КонецЕсли;

Таб. ВывестиСекцию ("Докум");

КонецЕсли;

КонецЦикла;

11

Если Обр="Печать" Тогда

Таб.ВывестиСекцию ("КонецОтчета"); Таб.Опции (0,0,5,0,"ОпцииЛечатиРеестра"); Таб.ТолькоПросмотр (1); Таб.Показать ("Реестр Документов", ""); Иначе

```
Зафиксировать Транзакцию ();
```

```
Предупреждение ("Обработка закончена!");
```

КонецЕсли;

КонецПроцедуры

В первых двадцати строках программы проверяется наличие в левом окне наименований видов документов для обработки. Если таковые имеются, то их идентификаторы помещаются в строку (переменная ВидыДок). Далее, если мы хотим сформировать реестр, то программа сразу же подготовит шапку печатной формы этого реестра. Если планируются какие-то иные действия (здесь опущенные), начинается транзакция.

Затем в цикле происходит обработка каждого документа. Здесь вспомним, что каждый документ имеет обязательные реквизиты шапки: номер, дату, а также идентификатор. Эти данные попадут в реестр для любого документа. Другое дело — сумма документа. У некоторых документов этого реквизита нет в принципе (например, документ "Начало месяца"). У других реквизит присутствует, но может располагаться в табличной части (документ "Реализация товаров") или в шапке (в нашей конфигурации такого примера нет). Для обработки таких ситуаций и используется работа с метаданными (от строки РеквСумма=Метаданные.Документ (Док.Вид()).РеквизитТабличнойЧасти ("Сумма");). В этой строке переменной РеквСумма присваивается реквизит Сумма табличной части документа — не только значение, но и все свойства реквизита. Переменной РеквВалюта присваиваются все свойства реквизита Валюта шапки документа. Далее, если реквизит Сумма существует, нужно проверить, установлен ли для него флаг Итог по колонке. Если установлен — вносим итоговое значение в переменную ПечСумма, если не установлен — переменная Печ-Сумма пуста.

#### Листинг 16.16

Если РеквСумма.Выбран()=1 Тогда

```
ПечСумма=?(РеквСумма.ИтогПоКолонке=1,Формат(Док.Итог("Сумма"), "Ч14.2-,"),"");
```

Далее следует обработка ситуаций с валютой.

Если переменная РеквСумма отсутствует, проверяется, а есть ли реквизит Сумма в шапке документа. Если реквизит найден, его значение вносим в переменную ПечСумма.

Таким образом, происходит поиск одинаковых реквизитов в различных документах и производится единообразная обработка этих документов, что позволяет получить реестр с необходимыми данными (рис. 16.9).

🌿 1С:Пре	дприяти	е - Бух и опер учет, ј	расчет - [Рее	стр Документов	*]					_ 8 ×
者 Файл	Действия	я Таблица Вид Оп	ерации Справ	зочники Документь	ы Журналы Результат	ы расчета Отче	ты Обрабо	отки Серв	ис Окна Помощь	_ 8 ×
12 🚔	🔛   X	B B 8 8 0		<b>å</b>	■ # % ?	🔜 🗎 🇭	🛄 м	M+ M-		
11						111				
			Peed	тр Документ	0В					
			c	11.01.98 по 29.09.06						
								Валюта		
Nº ⊓/⊓	Дата	Документ	Номер	Сумма	Инфо	рмация	4	цокумента		
1	01.06.06	Начало месяца	1							
2 <sup>2</sup>	00.00.00	работу	· · · · ·							
3	30.06.06	Приказ о приеме на	2							
4	30.06.06	работу Выплата	2	0.00						
	00.00.00	заработной платы	-							
5	01.07.06	Начало месяца	2	0-00						
Б	14.07.06	выплата заработной платы	1	1,900-00						
7	01.08.06	Поступление ОС	1	10,000-00						
8	07.08.06	Поступление	1	2,500-00						
9	31.08.06	Амортизация	3	2,500-00						
10	01.09.06	Начало месяца	3	2,500-00						
11	13.09.06	Списание	1	1,250-00						
12	25.09.06	Поступление	1	1,020-00						
		товаров								
13	27.09.06	Поступление	2	1,250-00						
14	29.09.06	Реализация товаров	1	250-00						
										_
4										
			[] 🖁	= F 📼 🔀 I	F					
Σ) Οбрабо	тка докум	ентов	р Документо	)B *						
Для получе	ния подск	сазки нажмите F1				NUM TA: 29.09.0	6 15:29:36	БИ: -	4 квартал 2006 г. ТП: Окт	ябрь 2006 г.
者 Пуск	<i>(</i> )	🕞 🔌 🚞 З Пров 🤜	2 Micro	🥂 3 1C:V	🔗 Справка 🞇 Раі	nt Sh 🛛 🍯 Core	elDR	Отчет п	■ [] [] [] [] [] [] [] [] [] [] [] [] []	<b>( ()</b> 🗐 🌆 16:54

Рис. 16.9. Образец печатной формы реестра документов



### Краткие сведения о работе с отладчиком

Создание конфигурации, как и любого прикладного программного обеспечения, можно условно разделить на несколько этапов. Например, постановка задачи, создание технического задания, создание объектов метаданных и написание кода программы, тестирование и отладка кода, тестовая эксплуатация, опытная эксплуатация, сдача продукта заказчику. На всех этих этапах могут выявляться самые разнообразные ошибки в программе. Синтаксические (несмотря на, казалось бы, постоянное применение синтаксического контроля), неверное переопределение переменных, логические ошибки постоянные спутники программиста. Выявлять ошибки можно, например, с помощью оператора Сообщить (значение параметра), прописанного в подозрительных местах. Но "1С:Предприятие" имеет специальную программу для отладки кода: она так и называется "Отладчик". Отладчик может выполнять несколько полезных функций, в частности, производить "пошаговое" выполнение программы: выполняется строка кода — остановка, можно просмотреть значения переменных, затем выполняется следующая строка опять остановка и т. д.

Запускается отладчик стандартным способом, так же, как Конфигуратор или пользовательский режим системы "1С:Предприятие" (рис. 17.1). Для запуска можно воспользоваться и соответствующими пунктами меню **Сервис** главного меню Конфигуратора. Кроме того, Отладчик может быть запущен при редактировании текста модуля с помощью пункта **Открыть в отладчике** меню **Действия** главного меню Конфигуратора.

Предварительно нужно запустить Конфигуратор (ведь в нем исправляют ошибки) и систему "1С:Предприятие" в пользовательском режиме.

8 режиме:	_	OK
🞢 Конфигуратор	🗕 🔽 Монопольно	Отмена
쭏 1С:Предприятие		
📆 Конфигуратор	ий учет для Каза: 🔺	Изменить
😻 Отладчик	ий учет для Каза:	
😔 Монитор	лад для Казахст	Добавить
Орбита Полит		
проверочнаякит Пистал		Удалить
РезкаВitex		
РезкаКИТ		
Самоучитель		Помощь

Рис. 17.1. Запуск Отладчика

В Отладчике есть специальная панель инструментов. Разберем несколько основных команд с использованием этой панели и меню. Более подробно с работой отладчика можно ознакомиться в документации к программе.

В окне Отладчика имеется список всех модулей, содержащихся в текущей конфигурации. Список модулей имеет древовидную структуру и практически идентичен структуре метаданных, которую можно видеть в окне Конфигуратора. Кроме того, окно Отладчика содержит ветвь **Внешние отчеты (обработки)** для работы с модулями внешних отчетов и глобальный модуль.

Так же, как и в Конфигураторе, при щелчке мышью по знаку "+" слева от наименования ветви дерева ветвь раскрывается вплоть до программных модулей.

Для открытия окна с текстом конкретного модуля необходимо дважды щелкнуть мышью на наименовании нужного программного модуля.

Определите примерный участок кода, где может прятаться ошибка. На однудве строки раньше (может быть и по-другому, тут надо исходить из конкретного контекста программного модуля) установите Точку останова. *Точкой останова* называется пометка, устанавливаемая слева от конкретной строки программного модуля. При достижении точки останова исполнение программного модуля останавливается и управление передается Отладчику.

Установить или снять точку останова можно двойным щелчком мыши на желтой полосе в левой части окна модуля, воспользовавшись меню **Отладка** | **Точка останова**.

### Замечание

Команды с клавиатуры для работы с Отладчиком можно посмотреть в Приложении 1.



Рис. 17.2. Окно Отладчика

Запустите систему "1С:Предприятие" в пользовательском режиме (можно воспользоваться кнопкой на панели инструментов или пунктом меню **Сервис | "1С:Предприятие"**) и выполните действия пользователя над тем объектом метаданных, который собираетесь отлаживать. Например, выполните процедуру Подбор. Как только программа выполнит действие процедуры и дойдет до выполнения строки, отмеченной в отладчике Точкой останова, работа системы "1С:Предприятие" в пользовательском режиме прекратится и откроется окно Отладчика.

Теперь имеет смысл воспользоваться следующей командой Отладчика — **Шагнуть в** (панель инструментов, либо меню **Отладка** | **Шагнуть в**). Это одна из команд режима пошагового выполнения программы. При выполнении этой программы Отладчик переходит к следующему оператору и прекращает работу программы до следующей команды пользователя. Если следующим выполняемым оператором модуля является вызов функции или процедуры, начинается ее пошаговое выполнение. Результаты вычислений программы в пошаговом режиме можно контролировать с помощью специального окна — Табло. Вызов табло на экран выполняется выбором команды Табло в меню Отладка главного меню Отладчика или нажатием кнопки на панели инструментов. Обычно в табло помещаются выражения, результаты вычисления которых необходимо контролировать в процессе пошагового выполнения модуля. Каждое выражение записывается отдельной строкой. Для ввода в табло нового выражения следует поместить курсор на первую свободную строку, набрать или скопировать из кода и вставить в табло выражение и нажать клавишу <Enter>. Справа от выражения появится знак "=" и результат вычисления выражения. Если выражение не может быть вычислено, вместо результата появится сообщение "Ошибка в выражении!". Обновление табло выполняется автоматически при выполнении любого оператора модуля, после которого управление передается Отладчику. Проконтролировав результаты вычислений, можно нажать кнопку Продолжить на панели инструментов. Эта команда прервет пошаговое выполнение программы и продолжится работа в обычном режиме.

🌠 Отладчик - Бук и опер учет, расчет - [Обработка.СкоростьВыпол	інения.Форна]	_ 8 ×
Файл Действия Текст Отладка Сервис Окна Помощь		_8×
	■異常 \$    電為野園で 国際してけれ *   55 / 2/2   名	
//***********************************		
Nep1=10/5		
Сообщинь (Пер1);		
КожецПроцедары		
Процедура Пример2() Перем Пер1;		
Для к=1 no 10 Цикл Пер1=10/ж;		
Ec.nu Hep1=2 Tosga CooSquub(Hep1);	Точка останова	
Прербаль; КонсцЕсли;		
Конеццикла; КожецПроцедуры		
		-
Для получения подсказки наконите F1	2.1 NUM	READ
🔊 Пуск 🖉 🖓 🕙 🎽 🗅 2 Проводняк 🔹 🕅 2 Матовоft О 🔹 📰	Paint Shop Pro 🦉 Konderyparop 🖓 Отладчек - Б 🔟 🍸 🤻 🔟 🗄 🖉 🕷 🖀 🌒	14:23

Рис. 17.3. Точка останова

В режиме пошагового выполнения возможен еще ряд команд:

- Шагнуть через если следующим выполняемым оператором модуля является вызов функции или процедуры, она выполняется целиком (не пошагово) и Отладчик переходит к следующему оператору;
- □ Шагнуть из прервать пошаговое выполнение функции или процедуры и остановиться на первом операторе, находящемся после ее вызова;
- □ Идти до курсора прервать пошаговое выполнение модуля, выполнять все операторы до той строки, на которой стоит курсор.

Точку останова можно убрать, воспользовавшись командой меню, панелью управления, либо дважды щелкнув по ней левой кнопкой мыши.

Существует еще ряд команд отладчика, познакомиться с которыми можно в документации к программе.

### 17.1. Замер производительности

Известно, что многие задачи в программировании можно решать несколькими способами. Одни из этих способов будут оптимальны по одним параметрам, другие — по другим. Для системы "1С:Предприятие" наиболее критичный параметр, как показывает практика, — скорость работы (производительность). Проиллюстрируем замер производительности на специальном примере. В структуру учебной базы входит специальная обработка Скорость Выполнения. Обработка состоит из двух процедур, в каждой их них нужно в переменную поместить число "2", но сделано это разными способами (листинг 17.1).

### Листинг 17.1

```
Процедура Пример1()
Пер1=10/5;
Сообщить (Пер1);
КонецПроцедуры
Процедура Пример2()
Перем Пер1;
Для к=1 по 10 Цикл
Пер1=10/к;
Если Пер1=2 Тогда
Сообщить (Пер1);
```

```
Прервать;
```

КонецЕсли;

КонецЦикла;

КонецПроцедуры

Для того чтобы замерить скорость выполнения этих процедур, сделаем следующее:

- 1. Откроем Конфигуратор.
- 2. Из Конфигуратора запустим Отладчик (Сервис | Отладчик).
- 3. Из Отладчика запустим "1С:Предприятие".
- 4. В Отладчике запустим Замер производительности (Отладка | Замер производительности либо кнопка на панели инструментов).
- 5. В "1С:Предприятии" запустим обработку Скорость Выполнения, пример 1.

🌠 Отлад	цчик - Бух и опер учет, расчет				_ 8 ×
Файл Д	ействия Отладка Сервис Окна Помощь				
2 🖻	🖬 🕮 🖻 🕾 🗢 🛎 🗛 📃	✓ Ă Ă ?	🕄 A 🦻 🐺 💿 💷 🗗 ?	) (P-1) + 🖬 🕍	<u>/1 🔟 🧏 🖌 🖌 🖌 🖌 🖌 🎸 🖌 🖌 🖌 🖌 🖌 🖌 🖌</u>
Спис	эк модулей		Замар произво		
B 📋 C	бработка.СкоростьВыполнения.Форма		Danop riporisoo		
	🗴 Замер1 *			_ 🗆 ×	
	Модуль	Строка	Кол.	Врем. %(Врем.)	
	Обработка.СкоростьВыполнения.Форма(5):	Сообщить(Пер1);	1	0.0 99.04	
	Обработка. Скорость Выполнения. Форма(4): Обработка. Скорость Выполнения. Форма(7):	Пер1=10/5; КонеиПроцедуры	1	0.0 0.41	
	обработ налакарост воблютна излюбриа (7).	Konceripoectypor	1	0.00	
	😨 Замер2 *			_ 🗆 ×	
	Модуль	Строка	Кол.	Врем. %(Врем.)	
	Обработка. Скорость Выполнения. Форма(14): Обработка. Скорость Выполнения. Форма(12):	Cooбщить(Пер1); Dep1=10/c;	1	0.0 97.96	
	Обработка. Скорость Выполнения. Форма (12).	Для к=1 по 10 Цикл	5	0.0 0.42	
	Обработка.СкоростьВыполнения.Форма(13):	Если Пер1=2 Тогда	5	0.0 0.26	
	Обработка. Скорость Выполнения. Форма (17):	КонецЦикла;	4	0.0 0.24	
	Обработка.Скороствовполнения.Форма(10).	КонецСон,	1	0.0 0.07	
	Обработка. Скорость Выполнения. Форма (15):	Прервать;	1	0.0 0.02	
•					
-					
	Кол. 0 Врем. 0.000000 %(Врем.)	0.00 🔽 Для вызовов процедур	и функций включать время выполн	ения	
Course	модулей	(1) Bawen1 *	Samen2 *		
Начать За	кончить замер производительности	]().			NI IM
	По со	oft O 🖌 🎆 Paint Shop Pro	1C:V7 start 😂 CorelDRAW 10		75 K 🖀 🔿 🗐 14:28

Рис. 17.4. Замеры производительности процедур Пример1 и Пример2

6. Перейдем в Отладчик и отожмем кнопку (пункт меню) Замер производительности. На экране появиться окно со скоростью выполнения именно одной процедуры Пример1.

В окне Замер производительности содержится следующая информация:

- Модуль содержит название модуля и номер строки;
- Строка текст данной строки модуля;
- Кол. количество вызовов данной строки за время замера;
- **Врем.** суммарное время (сек.) выполнения данной строки за время замера;
- **%(Врем.)** процент суммарного времени выполнения данной строки к общему времени замера (общее время замера равно сумме всех промежутков времени, в которые выполнялся код конфигурации).

🧏 Отладчик - Буж и опер учет, расчет - [Заме]	p3 *]			
о файл действия отладка сервис окна п		-		
^2 ਯ ⊌   X № ₽   ⊕ &   ⊃ ⊂ /	M		+ 0+ 0+ *0 →   <b>≥</b> 2	<u> // ¥   %  </u>
Модуль	Строка	Кол. Врем.	%(Врем.)	A
Глобальный модуль(43):	ОткрытьФормуМодально("Обработка.НастройкаУ	1 1.3	38.85	
Документ.ПоступлениеМатериалов.Форма(20):	ЗапросКоличестваВПодборе(Выб, Контекст); // пе	1 1.0	30.44	
Глобальный модуль(19):	Если ВвестиЧисло(Кол, "Введите количество", 10, 2)	1 1.0	30.41	
Глобальный модуль(6):	Предупреждение (привет);	1 1.0	29.77	
Документ.ПоступлениеМатериалов.Форма(16):	ОткрытьПодбор("Материалы","ФормаСписка");	1 0.0	0.89	
Обработка.НастройкаУчета.Форма(16):	Константа.Организация = Организация;	1 0.0	0.02	
Глобальный модуль(10):	Если Выб.ПометкаУдаления()=1 Тогда //товар пом	1 0.0	0.02	
Документ.ПоступлениеМатериалов.Форма(22):	Выч_суммы_накл(Контекст);	1 0.0	0.01	
Глобальный модуль(21):	Конт.ТМЦ=Выб; //Товар из Выб в добавленную строку	1 0.0	0.01	
Глобальный модуль(41):	ПоказыватьНастройкуТорговли=Константа.Контроль;	1 0.0	0.00	
Документ.ПоступлениеМатериалов.Форма(21):	УстанЦеныПрих();//вызов процедур обработки по	1 0.0	0.00	
Документ.ПоступлениеМатериалов.Форма(9):	Цена=ТМЦ.Себестоимость.Получить(ДатаДок); //	1 0.0	0.00	
Глобальный модуль(20):	Конт.НоваяСтрока(); // добавляем строку в табли	1 0.0	0.00	
Обработка.НастройкаУчета.Форма(7):	ПоказыватьНастройкуТорговли = Константа.Конт	1 0.0	0.00	
Глобальный модуль(15):	Если (Конт.Вид()="ПоступлениеОС") или (Конт.Ви	1 0.0	0.00	
Глобальный модуль(28):	Если (Конт.Вид()="ПоступлениеОС") или (Конт.Ви	1 0.0	0.00	
Глобальный модуль(35):	Конт.НДС=Конт.Сумма/(Конт.СтавкаНДС+100)*К	1 0.0	0.00	
Глобальный модуль(34):	Конт.Сумма=Конт.Цена*Конт.Количество;	1 0.0	0.00	
Глобальный модуль(30):	ИначеЕсли	1 0.0	0.00	
Документ.ПоступлениеМатериалов.Форма(4):	ПриЗаписиПерепроводить(1);	1 0.0	0.00	
Глобальный модуль(22):	Конт.Количество=Кол;//Количество- туда же	1 0.0	0.00	
Глобальный модуль(55):	привет="Здравствуйте!"; //инициализация переменной	1 0.0	0.00	
Обработка.НастройкаУчета.Форма(8):	Контроль=?(Число(ПоказыватьНастройкуТорговл	1 0.0	0.00	
Глобальный модуль(56):	приветствие(); //вызов процедуры	1 0.0	0.00	
Глобальный модуль(42):	Если ПоказыватьНастройкуТорговли <2 Тогда	1 0.0	0.00	
Документ.ПоступлениеМатериалов.Форма(23):	АктивизироватьСтроку();	1 0.0	0.00	
Глобальный модуль(7):	КонецПроцедуры	1 0.0	0.00	
Документ.ПоступлениеМатериалов.Форма(17):	КонецПроцедуры	1 0.0	0.00	
Глобальный модуль(14):	Кол=1;//Объявили переменную для ввода количе	1 0.0	0.00	
Глобальный модуль(25):	КонецПроцедуры	1 0.0	0.00	
Глобальный модуль(45):	КонецПроцедуры	1 0.0	0.00	
Обработка.НастройкаУчета.Форма(18):	КонецПроцедуры	1 0.0	0.00	
Глобальный модуль(13):	КонецЕсли;	1 0.0	0.00	
Глобальный модуль(23):	КонецЕсли;	1 0.0	0.00	
Глобальный модуль(24):	КонецЕсли;	1 0.0	0.00	
Глобальный модуль(36):	КонецЕсли;	1 0.0	0.00	
Глобальный модуль(37):	КонецПроцедуры	1 0.0	0.00	
Глобальный модуль(44):	КонецЕсли;	1 0.0	0.00	-
Обработка.НастройкаУчета.Форма(11):	КонецПроцедуры	1 0.0	0.00	
Документ.ПоступлениеМатериалов.Форма(5):	КонецПроцедуры	1 0.0	0.00	-
Локумент.ПоступлениеМателиалов.Фолма(11):	КоненПроцелуры	1 0.0	0.00	-
Кол. 0 Врем. 0.000000 %(Врем.)	0.00 🔽 Для вызовов процедур и функций включать вре	мя выполнения		
🖫 Список модулей 🛛 📋 Обработка. Скор	оостьВып 🔯 ЗамерЗ *			
Для получения подсказки нажните F1				NUM
🥂 Пуск 🍈 🕞 🕞 » 🕞 2 Поводник 🔹 🕅	2 Microsoft O V 🐨 Paint Shop Pro	CorelDRAW 10		

 Снова нажмем в Отладчике кнопку (пункт меню) Замер производительности, перейдем в "1С:Предприятие", нажмем в обработке кнопку Пример2. Вернемся в Отладчик, отожмем кнопку (пункт меню) Замер производительности, появится второе окно с производительностью второй процедуры. Разница результатов хорошо заметна (рис. 17.4).

Теперь просмотрим производительность конфигурации в некоторых режимах. Например, запуск конфигурации, заполнение приходного документа и окончание работы конфигурации. Для этого сначала выйдем из системы "1С:Предприятие", в Отладчике нажмем кнопку **Замер производительности**, запустим "1С:Предприятие" в пользовательском режиме. Заполним приходный документ и выйдем из пользовательского режима. В Отладчике уже будет окно с замером производительности конфигурации на протяжении всего периода работы (рис. 17.5).

Общий принцип замера производительности таков: замер производительности в Отладчике включается перед частью конфигурации, которую требуется проанализировать, и отключается после отработки этой части. Глава 18



### Итоговое описание разработанной конфигурации

В *главе 3* мы ознакомились с порядком создания пользовательского интерфейса. Поскольку конфигурация готова, можно организовать полный набор пользовательских интерфейсов в соответствии с кругом задач, предполагаемых к решению пользователями.

В процессе создания документов и справочников конструкторами нам предоставлялась возможность сразу включать новые объекты в существующие интерфейсы (рис. 18.1). Если эта возможность не была использована, тогда будем создавать новые интерфейсы. Если же что-то было уже включено в меню — будем редактировать существующие интерфейсы.

Предположим, что требуется создать три интерфейса: Бухгалтер, Менеджер и Расчетчик. При этом пользователи интерфейса Бухгалтер будут иметь доступ ко всем справочникам, документам и отчетам конфигурации. Интерфейс Менеджер дает доступ к работе с документами оплаты и движения товаров, а также с соответствующими им справочниками и отчетами. Интерфейс Расчетчик позволяет работать с документами, справочниками и отчетами, относящимися к задачам расчета заработной платы.

В *главе 3* описано создание нового интерфейса. Согласно этому описанию, создайте новый интерфейс с идентификатором **Бухгалтер**. Щелкните по созданному интерфейсу правой кнопкой мыши и в контекстном меню выберите пункт **Редактировать меню**. После того как в окне редактора меню появилось основание дерева меню, используем конструктор. Он вызывается через меню **Действия** | **Конструктор** или соответствующей кнопкой на панели инструментов (рис. 18.2).

В диалоговом окне конструктора указаны колонки (меню первого уровня) и пункты (меню второго уровня) (рис. 18.3).

Конструктор Справо		
🔲 Вставить команду		
□Расчет		
Конструктор позволя справочника в меню интерфейсы, в меню этого справочника. К "Справочники".		
< Назад Да		

Рис. 18.1. Добавление в интерфейс новых объектов конструктором

в, <mark>#</mark> Конфигурация *		<u>- 0 ×</u>
1 4 2 1 1	źĮ ⊨	
— 🕵 Бухгалтер	🛐 Бухгалтер - Редактор меню 📃 🗆 🗙	
	≥ ≪ 2	
	<ul> <li>Спос Док;</li> <li>Удалить</li> <li>Док;</li> <li>Удалить</li> <li>Конструктор</li> <li>Резу <ul> <li>Док;</li> <li>Удалить</li> <li>Док;</li> <li>Удалить</li> <li>Хур</li> <li>Конструктор</li> <li>Док;</li> <li>Док;</li> <li>Удалить</li> <li>Док;</li> <li>Удалить</li> <li>Хур</li> <li>Конструктор</li> <li>Док;</li> <li>Конструктор</li> <li>Док;</li> <li>Конструктор</li> <li>Док;</li> <li>Док;</li> <li>Док;</li> <li>Конструктор</li> <li>Док;</li> <li>Док;</li> <li>Конструктор</li> <li>Док;</li> <li>Док;</li></ul></li></ul>	
Метаданные 1	Развернуть все Свернуть все Свернуть все Свернуть все	

Рис. 18.2. Вызов конструктора меню

на Конфигурация * 🔁 🍬 🖻 🖡 🕇 🏭 🗎		
🗟 Бухгалтер - Редактор меню		
Колструктор меню Колонки Справочники Спр	Пункты Справочники. Открыть Справочники. Материалы. Открыть Справочники. ОсновныеСредства. Открыть Справочники. Сотрудники. Открыть Справочники. Контрагенты. Открыть Справочники. Основания. Открыть Справочники. Номенклатура. Открыть Справочники. Подразделения. Открыть Справочники. Прафики. Открыть Справочники. Графики. Открыть	Построить Отмена Помощь





Рис. 18.4. Состав колонки Документы

Для просмотра состава колонок нужно щелкнуть мышкой по соответствующему названию колонки в диалоговом окне конструктора.

Конструктор автоматически включает все справочники, отчеты и обработки в колонки меню первого уровня.

Для пункта меню **Справочники** оставим все справочники, имеющиеся в конфигурации.

При конструировании состава меню Документы следует учесть, что для документов имеются два вида команд меню:

- □ Ввести;
- 🗖 Ввести на основании.



Рис. 18.5. Состав меню Бухгалтер

Поэтому для настройки колонки меню Документы необходимо флажками указать нужную команду. По умолчанию для документов установлена команда **Ввести** (рис. 18.4).

Команда **Ввести на основании** используется в том случае, когда для документа не предусмотрен режим **Ввод нового**.

Для остальных колонок меню оставляем полный состав соответствующих объектов конфигурации.

В результате меню для бухгалтера будет выглядеть, как показано на рис. 18.5.

Интерфейс бухгалтера представляет собой полный набор объектов. Любой другой интерфейс можно рассматривать как его часть.

Создадим интерфейс **Менеджер** на основе полного интерфейса **Бухгалтер**. Для этого щелкните правой кнопкой мыши и в контекстном меню выберите пункт **Новый подчиненный интерфейс**. Задайте название и укажите режим **Отключить меню "Операции"**. Этот режим закрывает доступ ко всем объектам конфигурации через общий пункт меню **Операции**. Остается возможность работать только с объектами, выведенными в интерфейсе.

ы. ака Конфигурация *	_ 🗆 ×
管 ‰ ⊠ ↓ ↑ ④ 目	
⊡…∰ Бухгалтер ∭ Новый1	
Свойства пользовательского интерфейса	×
Основные	§ -#
Название: Менеджер	
🔽 Отключить меню "Операции"	ОК
Использовать при отключенной авторизации	Отмена
	Обновить
🚯 Метаданные 🛛 🛱 Интерфейсы 🖏 Права	

Рис. 18.6. Свойства подчиненного интерфейса

После создания подчиненного интерфейса щелкните правой кнопкой мыши и в контекстном меню выберите пункт **Редактировать**. В открывшемся окне со списком возможных объектов необходимо указать те позиции главного

интерфейса, которые должны быть доступны пользователю с подчиненным интерфейсом (рис. 18.7).

e( <mark>#</mark> Ko	онфигу	урация *		<u>- 🗆 ×</u>	
	≷ ≥	1   1 社 社   11			
	👷 Бухг 🏥 🎊 Г	алтер Менеджер			
	讍 Me	енеджер			
	Кома	нда	Парам	Пункт меню и подсказки	Акселератор 🔺
		Документы.Амортизация.Ввести		Амортизация; Документы.Амортиз	
🗹 Документы.Выписка.Ввести			Выписка банка; Документы.Выпис		
		Документы.ВыплатаЗарплаты.Ввести		Выплата заработной платы; Докум	
		Документы.Затраты.Ввести		Распределение на затраты; Докум	
		Документы.НачалоМесяца.Ввести		Начало месяца; Документы.Начало	
		Документы.Операция.Ввести		Операция; Документы.Операция.Вв	
		Документы.ПКО.Ввести		Приходный кассовый ордер; Докум	
		Документы.ПоступлениеМатериало		Поступление материалов; Докумен	
		Документы.ПоступлениеОС.Ввести		Поступление ОС; Документы.Посту	
5		Документы.ПоступлениеТоваров.Вв		Поступление товаров; Документы	<b>_</b>
	1				

Рис. 18.7. Окно редактирования подчиненного интерфейса

≓ <mark>⊭</mark> Конфигура	ация *	
100 10a DA	📘 🛧 🐴 📄 Свойства пользовательского интерфейса	
	<ul> <li>Z : Ш</li> <li>Основные</li> <li>Название: Гость</li> <li>Название: Гость</li> <li>Отключить меню "Операции"</li> <li>Ф Отключить меню "Операции"</li> <li>Использовать при отключенной авторизации</li> </ul>	
	Документы.Выписка	
	Документы.ВыплатаЗарплаты.Ввести	Выпл
	Документы. Затраты. Ввести	Распр
	Документы.НачалоМесяца.Ввести	Начал
	Документы.Операция.Ввести	Опера
	Документы.ПКО.Ввести	Прихс
	Документы.ПоступлениеМатериало	Посту
	Документы.ПоступлениеОС.Ввести	Посту
		► //.

Рис. 18.8. Интерфейс Гость

Для интерфейса Бухгалтер создадим еще один подчиненный интерфейс Гость и назначим ему свойства Отключить меню операции и Использовать при отключенной авторизации. В списке редактирования состава меню уберем пометки у всех позиций (рис. 18.8). Таким образом, пользователь, попавший в программу без авторизации, окажется с пустым интерфейсом, т. е. лишенным возможности обратиться к какой-либо информации.



Рис. 18.9. Окно редактирования интерфейса Расчетчик

Создадим новый интерфейс **Расчетчик**, используя конструктор так же, как для создания интерфейса **Бухгалтер**. По каждой колонке меню оставим позиции, необходимые для расчета заработной платы. В результате настройки получаем меню, представленное на рис. 18.9. После создания интерфейсов перейдем к созданию панелей инструментов.

Для запуска редактора панели инструментов на вкладке **Интерфейсы** окна **Конфигурация** следует выбрать нужный интерфейс и щелкнуть на нем правой кнопкой мыши. В открывшемся меню выбрать пункт **Редактировать па**нель инструментов.

Окно редактора панелей инструментов, открывшееся по этой команде, состоит из трех частей (рис. 18.10).

Список панелей позволяет создать столько панелей, сколько нужно. Состав панели позволяет включать или исключать кнопки из панели. Библиотека пиктограмм позволяет выбрать необходимое изображение для кнопки.

Создадим панель инструментов Справочники. Для этого в окне диалога Список панелей нажмем кнопку Добавить либо зайдем в меню Действия | Создать панель. В появившейся палитре свойств (см. рис. 18.10) заполним название панели Справочники.

в¦# Конфигурация *		
🖹 🍬 🖻 🖡 🕇 🛃 🗎		
🖃 🚭 Бухгалтер		
🐺 Бухгалтер - Редакто	р панелей инструментов	
Список панелей 🛅	Cостав панели	
Новый1		
Библиотека пиктограмм	Заменить	
	Свойства панели инструментов	×
	Основные	_ ? <b>-</b> ⊨
<b>ZARKON</b>	Название: Справочники	
	По умолчанию	
	Положение Сверху	ок 1
	Показывать Располагать с новой строки	Отмена
		Обновить

Рис. 18.10. Редактор панелей инструментов

Группу полей **По умолчанию** оставим без изменения. Нажмем кнопку **ОК**, и в списке панелей появится панель **Справочники**.

Следующий шаг — сформировать состав панели. Используя кнопку Добавить в окне диалога Состав панели, либо через меню Действия | Добавить кнопку, открываем палитру свойств новой кнопки панели инструментов. Ее заполнение аналогично заполнению палитры свойств элемента меню (рис. 18.11). Нажмите кнопку ОК — и в диалоговом окне Состав панели появятся наименование вызываемого объекта метаданных и название действия, которое произойдет при нажатии данной кнопки (рис. 18.11).



Рис. 18.11. Заготовка для кнопки панели инструментов Справочники

	💷 Бухгалтер - Редактор панелей и
h	
	Список панелей
	Справочники
	Библиотека пиктограмм
	1. 9 a. a 9 3 4 4 3 3 A 4
	アミッノミンゆりり/シン
	🔜 🗢 🚯 🛼 🕸 🖄 🐼 🖉 🖓 🕅 🖓 🌆
1	//////////////////////////////////////
1	X COM NO VAR DING 21. COM
1	
1	1.2 mai 1998 1. 1991 1 A. 1942 1 241 251 1991 2881 2001
D!	

Рис. 18.12. Состав панели инструментов Справочники

Сама кнопка — серая, поскольку является заготовкой для пиктограммы. Зацепите мышкой подходящую пиктограмму из окна диалога **Библиотека пиктограмм** и перетащите ее на серый квадратик. Кнопка готова (рис. 18.12). Таким же образом выведем кнопки для основных, наиболее часто используемых справочников.

Таким же способом создадим панели инструментов **Документы** (рис. 18.13) и **Отчеты** (рис. 18.14), выведя в пункты меню основные или наиболее часто используемые объекты.



Рис. 18.13. Состав панели инструментов Документы

100	💯 Бухгалтер - Редактор панелей и
ſ	Список панелей
I	Справочники
	Документы Отчеты
l	1
	Библиотека пиктограмм
I	
I	喝噌

Рис. 18.14. Состав панели инструментов Отчеты

После создания интерфейсов и панелей инструментов перейдем к созданию наборов прав пользователей.

В *славе 3* даны начальные сведения по созданию наборов прав. Более подробно с правилами настройки прав пользователей можно ознакомиться в дополнительной книге к программному обеспечению "Конфигурирование и администрирование. Часть 1". Наша задача — создать необходимые наборы прав согласно условиям сквозного примера.

Для создания набора прав на вкладке **Права** окна **Конфигурация** следует выбрать пункт меню **Действия | Новый** и в открывшемся окне задать имя набора прав (рис. 18.15).

<sub>п(п</sub> Конфигурация *	_ 🗆 ×
"≧ 🍬 🖻 🖡 🕇 🛃 🗎	
Наборы прав	
🕎 Администратор	
😵 Новый1	
Свойства набора прав	×
Основные	ଡ଼ –⊨
Название: Новый1	
	ок
	Отмена
	Обновить
📳 Метаданные 🔤 Интерфейсы 👔 🦄 Права	

Рис. 18.15. Свойства набора прав

Изначально автоматически создается набор Администратор, включающий в себя полный список прав. Создадим дополнительные наборы прав с ограничениями по доступу:

- □ Менеджер без возможности удаления объектов и без возможности редактирования справочника "Сотрудники";
- □ Расчетчик без возможности удаления объектов, без возможности ввода и редактирования документов по кассе, но с возможностью их просмотра в журнале.

Остальным пользователям будут назначаться полные права — набор прав Администратор.

Создадим новый набор прав **Менеджер**. При создании нового набора ему не устанавливается никаких прав. Для настройки прав доступа очень удобно действовать методом отмены прав по запрещенным операциям. Для этого первоначально устанавливаются все права следующим образом: нажатием правой кнопки мыши на выбранном наборе прав вызываем контекстное меню и выбираем пункт **Установить все права** (рис. 18.16).



Рис. 18.16. Установить все права

После этого переходим к редактированию прав по объектам. Для этого двойным щелчком курсора по выбранному набору прав вызываем окно редактора пользовательских прав (рис. 18.17) и корректируем права по работе с объектами.

Для нашего примера в наборе прав **Менеджер** мы корректируем свойства конфигурации, отменив право на удаление помеченных объектов (рис. 18.18) и любое другое право по вашему усмотрению и отменив права на работу со справочником "Сотрудники", назначив только режим **Чтение** (рис. 18.19).

По аналогии самостоятельно настройте права доступа Расчетчик.

н <sup>ж</sup> Конфигурация *
12 🍬 🖻 🖡 🕇 🛃 🖹
Наборы прав
👔 Администратор
🍸 Менеджер 🌱 Менеджер - Редактор пользова 🗖 🗖 🗙
🖃 🔂 Учебная Компл учет
🕀 💼 Константы
🕀 😁 Справочники
⊡ Документы ∴ <b>А</b> агуу
Штирани Паранании Паранании Паранании
Ш Ф Пари систор
Thian everope
Типовая операция
П П Регистры
🕀 🛄 Журналы расчетов
🗄 📮 Календари
🚯 Метаданные 🔐 Интерфейсы 🔅 Права

Рис. 18.17. Редактор пользовательских прав

к <mark>#</mark> Конфигурация *	
™ 🗞 🖻 🖡 🕇 🛃 🗎	
Наборы прав	
🖞 Администратор	
🚏 Менеджер	
Менеджер - Редактор пользовательских	
🖃 🎼 Учебная Компл учет	
н Спреволники	
Свойства элемента прав	X
Доступ	<u>ଡ</u> ଼ା–⊯
Сохранение/Выгрузка данных	
Управление оперативными итогами	
Управление бухгалтерскими итогами	
Монитор Н Н	OK
✓Использование в качестве ULE Automation ce	Отмена
	Обновить

Рис. 18.18. Свойства элемента прав (конфигурация)



Рис. 18.19. Свойства элемента прав (справочник)

После создания наборов прав и интерфейсов можно ссылаться на них в текстах модулей. Для этого используются методы, описанные в синтакс-помощнике в разделе Встроенный язык | Системные | Функции/процедуры | Среда исполнения (рис. 18.20) и в аналогичных разделах документации по описанию встроенного языка.

Использование методов встроенного языка позволяет программно скорректировать права пользователей дополнительно к имеющимся настройкам.

Например, в глобальном модуле процедуру ПриЗавершенииСистемы() дополним проверкой набора прав пользователя. Для пользователя с интерфейсом Бухгалтер напомним о необходимости создать резервную копию (листинг 18.1).

```
Листинг 18.1
```

//предопределенная процедура

Процедура ПриЗавершении Работы Системы ()

```
Если Вопрос("Работа программы закончена?", "Да+Нет")="Нет" тогда //произвольное
```

Статусвозврата (0);

КонецЕсли; //процедуры

Если НазваниеИнтерфейса()="Бухгалтер" Тогда

Предупреждение ("Бухгалтер - не забудь создать копию!");

КонецЕсли;

Конецпроцедуры



Рис. 18.20. Методы среды исполнения

После создания в конфигураторе интерфейсов и наборов прав можно сформировать список пользователей с требуемыми ролями. Комбинации из интерфейсов и наборов прав позволяют создать различные варианты ролей для пользователей.

Для ввода пользователя выберем пункт меню **Администрирование** | **Пользователи**. В открывшемся окне (рис. 18.21) создадим новый элемент (**Действия** | **Новый**).

// тело

При этом зададим свойства нового пользователя на вкладке **Атрибуты** (см. рис. 18.21).

В поле **Рабочий каталог** можно указать имя каталога, который будет использоваться для размещения файлов, создаваемых системой "1С:Предприятие".

После этого на вкладке Роль назначим пользователю интерфейс и права (рис. 18.22).

Включение режима Отключить контроль прав равносильно тому, как если бы пользователю были предоставлены все права.

🕼 Пользователи *	
1 2 I I I I I I	8-5 5-6
Имя	
🛇 Иванов	
Свойства пользовател	ns
Атрибуты Роль	
Имя: Ива	знов
Полное имя: Ива	анов
Рабочий каталог: 🗍	anov
<u> </u>	

Рис. 18.21. Создание нового пользователя

🕢 Пользователи *	
1 x I t 4 6 8	
Имя	
🕵 Иванов	
Свойства пользователя	
Атрибуты Роль	
Права: Менеджер	
Интерфейс: Менеджер	
🗖 Отключить	
र	

Рис. 18.22. Назначение роли пользователя

Для пользователя можно задать пароль через меню Действия | Изменить пароль (рис. 18.23).

Пользователи, которым назначен пароль, в списке помечаются значком "за-мок" (рис. 18.23).

После формирования и сохранения списка пользователей запуск программы будет происходить с авторизацией (рис. 18.24).

🐉 Пользоват	ели *	_ 🗆 🗵
10 🛛 🖡	↑ 2I 🔓 🚟 🗂	
Имя	Полное имя	Роль ( Права, Интерфейс)
🛃 Иванов	Иванов Иван	Менеджер, Менеджер
👮 Петров	Петров Петр	Администратор, Бухгалтер
	Смена пароля Введите новый пароль для пользователя Петров: [	СК Отмена Помощь
•		•

Рис. 18.23. Назначение пароля пользователю

10		1691	редп	рият	ие
and and				Зерсия	7.7
Автори	зация доступ	а		X	1 推行
	Пользователь: Пароль:	Иванов Петров ОК	Отмена	Помощь	1
<i>Лиценз<mark>ия на про</mark> Skoblikova Skoblikova Загрузка конфи</i>	ограмму принадл ггурационной инф	ежит: © 199 рормации	6-1999 фирма 1	С. Все права за	цищены.

Рис. 18.24. Запуск программы с авторизацией
# Заключение

Описанные в книге приемы программирования и конфигурирования — это только элементарные начала. Авторы будут считать большой удачей, если знания, полученные из данного самоучителя, послужат читателям базой для дальнейшего изучения программирования и конфигурирования в системе "1С:Предприятие". Для дальнейшего совершенствования полученных навыков мы рекомендуем:

использовать готовые конфигурации — внедренные решения. Начинать понемногу усовершенствовать их, так как там наверняка найдется обширное поле для вашей деятельности. Одновременно изучать, каким образом составлены программы по тем или иным моментам более опытными коллегами. Можно вести специальный журнал, в котором описаны удачные решения (например, организация работы со справочником, проведение документа и т. д.) в изученных конфигурациях. Эти решения можно привязывать к собственным разработкам. Очень редко в работе программиста приходится изобретать что-то новое. В большинстве случаев нужно уметь грамотно использовать уже известные наработки;

постоянно пользоваться документацией к программе;

- использовать ресурсы Интернета, обсуждать возникшие проблемы на форумах. Особенно если вы — начинающий разработчик;
- читать специализированную литературу, причем лучше иметь несколько книг разных авторов.



# Приложения

Приложение 1



## Команды, выполняемые с помощью меню конфигуратора

Горизонтальное меню — меню первого уровня	Вертикальное меню — меню второго уровня	Выполняемые действия
Файл	Новый	Используется для создания нового файла текста, таблицы или внешнего отчета. Не создает новую конфигурацию
	Открыть	Используется для открытия существующих файлов текста, таблицы или внешнего отчета. Вызывает стандартное диалоговое окно Windows
	Закрыть	Используется для закрытия объекта в ак- тивном окне
	Сохранить	Используется для сохранения изменений в базе данных, текста, таблицы или внеш- него отчета
	Сохранить как	Используется для сохранения изменений объекта в активном окне в файл — текста, таблицы или внешнего отчета
	Сравнить файлы	Вызывает режим сравнения файлов. По- зволяет сравнить, например, тексты про- граммных модулей двух версий внешних отчетов и визуально определить изменен- ные участки кода
	Печать	Используется для печати таблиц отчетов и т.п.

Горизонтальное меню — меню первого уровня	Вертикальное меню — меню второго уровня	Выполняемые действия
	Просмотр	Используется для предварительного про- смотра перед печатью печатных форм
	Параметры стра- ницы	Используется для установки параметров печати: размер страницы, ориентация бу- маги, имя принтера и т. д.
	Список файлов	Список последних открытых файлов. Слу- жит для быстрого повторного открытия этих файлов
	Выход	Используется для завершения работы с программой
Конфигурация (см. рис. 2.1)	Открыть конфигу- рацию	Используется для открытия окна конфигу- рации
	Поиск во всех тек- стах	Используется для поиска заданной поль- зователем строки в модулях, диалогах, таблицах и описаниях конфигурации
	Полный синтакси- ческий контроль	Выполняет синтаксический контроль, т. е. проверку на наличие всех операторов в сложных конструкциях встроенного языка, правильности построения конструкций во всех модулях конфигурации
	Описание структу- ры метаданных	Выводит в текстовый файл информацию обо всех объектах метаданных конфигу- рации
	Загрузить изме- ненную конфигу- рацию	Выполняет импорт конфигурации, заменяя имеющийся вариант на импортируемый
	Объединение кон- фигураций	Позволяет импортировать выборочный раздел новой конфигурации в имеющуюся
	Конвертировать данные	Вызывает конвертер данных из предыду- щих версий программ фирмы "1С"
Конструкторы	Новый справочник	Вызывает конструктор (мастер создания) справочника
	Новый документ	Вызывает конструктор (мастер создания) документа

Горизонтальное меню — меню первого уровня	Вертикальное меню — меню второго уровня	Выполняемые действия
	Новый Журнал документов	Вызывает конструктор (мастер создания) Журнала документов
	Новый отчет	Вызывает конструктор (мастер создания) отчета
	Новый Вид субконто	Вызывает конструктор (мастер создания) нового вида субконто (для компоненты "Бухгалтерский учет")
	Новый Журнал расчетов	Вызывает конструктор (мастер создания) журнала расчетов для создания "Журнал расчетов" (компонента "Расчет")
	Печать	Вызывает конструктор (мастер создания) печатной формы и процедуры на встроен- ном языке, связанной с ней
	Макет отчета	Вызывает конструктор (мастер создания) отчета. Создает шаблон печатной формы отчета и процедуру на встроенном языке, связанную с этим шаблоном
	Бухгалтерский запрос	Вызывает конструктор (мастер создания) бухгалтерского запроса (компонента "Бух- галтерский учет")
	Запрос	Вызывает конструктор (мастер создания) запроса
	Ввод на основании	Вызывает конструктор (мастер создания) механизма ввода документа на основании
	Формирование операции	Вызывает конструктор (мастер создания) формирования бухгалтерской операции (компонента "Бухгалтерский учет")
	Движения регистров	Вызывает конструктор (мастер создания) механизма движения регистров (компо- нента "Оперативный учет")
Администри- рование	Пользователи	Открывает окно для работы со списком пользователей
	Сохранить данные	Вызывает диалоговое окно для архивиро- вания данных

Горизонтальное меню — меню первого уровня	Вертикальное меню — меню второго уровня	Выполняемые действия
	Восстановить дан- ные	Вызывает диалоговое окно для восстанов- ления архивированных данных
	Выгрузить данные	Вызывает диалоговое окно для выгрузки информационной базы в файл переноса данных
	Загрузить данные	Вызывает диалоговое окно для загрузки информационной базы из файла перено- са данных
	Настройка журна- ла регистрации	Вызывает диалоговое окно для настройки журнала регистрации событий в инфор- мационной базе
	Тестирование и исправление БД	Вызывает диалоговое окно для установки параметров и запуска тестирования и ис- правления информационной базы
	Кодовая страница таблиц ИБ	Вызывает диалоговое окно для установки кодовой страницы
	Распределенная ИБ	Подменю для работы с распределенными информационными базами (только для компоненты УРБД)
Сервис	Калькулятор	Вызов калькулятора
	Календарь	Вызов календаря
	Синтакс- помош- ник	Вызов Синтакс-помощника
	Настройка шабло- нов	Вызов окна для настройки шаблонов встроенного языка
	Временная блоки- ровка	Вызов режима временной блокировки
	Параметры	Настройка параметров системы
	Панели инстру- ментов	Настройка панелей инструментов
	1С:Предприятие	Вызов "1С:Предприятия" в пользователь- ском режиме
	Отладчик	Вызов Отладчика

(окончание)

Горизонтальное меню — меню первого уровня	Вертикальное меню — меню второго уровня	Выполняемые действия
	Монитор пользова- телей	Вызов Монитора пользователей
Окна	Каскад	Расположить все открытые окна каскадом
	Закрыть	Закрыть текущее окно
	Закрыть все	Закрыть все окна
	Закрыть окно со- общений	Закрыть окно сообщений
	Панель окон	Управление панелью окон
	Окна	Работа со списком открытых окон
Помощь	Помощь	Вызов помощи
	Содержание	Вызов содержания помощи
	Поиск в Синтакс- помощнке	Контекстный поиск в Синтакс-помощнике
	Советы дня	Вызов текущего совета дня
	О системе "1С:Предприятие"	Вызов общего описания системы "1С:Предприятие"
	О программе	Актуальные данные о технологической платформе и текущей информационной базе

### Приложение 2



# Быстрые (горячие) клавиши

Клавиша (сочетание клавиш)	Действие	Примечание
	Конфигуратор	
	Конфигурация	
<alt>+<k>+<enter></enter></k></alt>	Открыть конфигурацию	
<alt>+<k>+<t></t></k></alt>	Поиск во всех текстах	
<alt>+<k>+<o></o></k></alt>	Объединение конфигураций	
<alt>+<k>+&lt;3&gt;</k></alt>	Загрузить измененную конфигурацию	
<alt>+<k>+<e></e></k></alt>	Конвертирование данных	
Администрирование		
<alt>+<a>+&lt;∏&gt;</a></alt>	Пользователи	
<alt>+<a>+<x></x></a></alt>	Сохранить данные	
<alt>+<a>+<b></b></a></alt>	Восстановить данные	
<alt>+<a>+<bl></bl></a></alt>	Выгрузить данные	
<alt>+<a>+<f></f></a></alt>	Загрузить данные	
<alt>+<a>+<h></h></a></alt>	Настройка журнала регистрации	
<alt>+<a>+<p></p></a></alt>	Распределенная ИБ	
<alt>+<a>+<p>+<y></y></p></a></alt>	Управление	
<alt>+<a>+<p>+<a></a></p></a></alt>	Автообмен	

Клавиша (сочетание клавиш)	Действие	Примечание	
Действия			
<alt>+&lt;Д&gt;+<t></t></alt>	Открыть в отладчике		
<alt>+&lt;Д&gt;+&lt;К&gt;</alt>	Синтаксический контроль запросов		
<alt>+&lt;Д&gt;+&lt;И&gt;</alt>	Синтаксический контроль		
<alt>+&lt;Д&gt;+<p></p></alt>	Процедуры и функции модуля		
	Файл		
<ctrl>+<n></n></ctrl>	Новый		
<ctrl>+<o></o></ctrl>	Открыть		
<ctrl>+<s></s></ctrl>	Сохранить		
<ctrl>+<p></p></ctrl>	Печать		
<alt>+&lt;Φ&gt;+&lt;Ρ&gt;</alt>	Сравнить файлы		
<alt>+&lt;Φ&gt;+&lt;3&gt;</alt>	Закрыть		
<ctrl>+<p></p></ctrl>	Печать		
<alt>+&lt;Φ&gt;+<p>+<p>+ +<enter></enter></p></p></alt>	Просмотр		
<alt>+&lt;Φ&gt;+<m></m></alt>	Параметры страницы		
	Работа с вкладками		
<shift>+<f2></f2></shift>	Предыдущая вкладка		
<alt>+<f2></f2></alt>	Вкладка		
<f2></f2>	Следующая вкладка		
Работа с блоком			
<ctrl>+<shift>+<f></f></shift></ctrl>	Форматировать		
<tab></tab>	Сдвинуть вправо		
<shift>+<tab></tab></shift>	Сдвинуть влево		
<ctrl>+<num></num></ctrl>	Добавить комментарий		
<ctrl>+<shift>+<num></num></shift></ctrl>	Удалить комментарий		

Клавиша (сочетание клавиш)	Действие	Примечание	
Окна			
<ctrl>+<f4></f4></ctrl>	Закрыть окно		
<ctr>l+<shift>+<f6></f6></shift></ctr>	Следующее окно		
<ctrl>+<f6></f6></ctrl>	Предыдущее окно		
<alt>+<o>+<k></k></o></alt>	Каскадом		
<ctrl>+<shift>+<z></z></shift></ctrl>	Закрыть окно сообщений		
<ctrl>+<alt>+<f></f></alt></ctrl>	Следующая страница		
<ctrl>+<alt>+<b></b></alt></ctrl>	Предыдущая страница		
<alt>+<o>+<p></p></o></alt>	Разделить окно		
	Поиск и замена		
<ctrl>+<shift>+<f3></f3></shift></ctrl>	Заменить		
<ctrl>+<f3></f3></ctrl>	Искать		
<f3></f3>	Повторить поиск		
<alt>+<f3></f3></alt>	Искать назад		
<shift>+<f3></f3></shift>	Искать вперед		
<ctrl>+<shift>+<l></l></shift></ctrl>	Переход к строке		
	Сервис		
<ctrl>+<f2></f2></ctrl>	Калькулятор		
<alt>+<c>+<c></c></c></alt>	Синтакс-помощник		
<alt>+<c>+<h></h></c></alt>	Настройка шаблонов		
<alt>+<c>+<b></b></c></alt>	Временная блокировка		
<f11></f11>	Запуск предприятия		
<f12></f12>	Запуск отладчика		
Помощь			
<f1></f1>	Помощь		
<shift>+<f1></f1></shift>	Содержание		
<ctrl>+<f1></f1></ctrl>	Поиск в синтакс-помощнике		
<alt>+&lt;Π&gt;+<o></o></alt>	О программе		

Клавиша (сочетание клавиш)	Действие	Примечание	
	Общие		
<alt>+<enter></enter></alt>	Свойства		
<alt>+<f4></f4></alt>	Выход		
	Предприятие		
	Документ		
<ctrl>+<enter></enter></ctrl>	Кнопка по умолчанию		
<f4></f4>	Выбор поля (альтернатива мыши)		
<tab></tab>	Прямой обход полей		
<shift>+<tab></tab></shift>	Обратный обход полей		
	Справочник		
	Действия		
<ins></ins>	Новый элемент		
<ctrl>+<f9></f9></ctrl>	Новая группа		
<shift>+<enter></enter></shift>	Изменить элемент		
<alt>+&lt;Д&gt;+<p></p></alt>	Просмотр элемента или группы		
<f9></f9>	Копировать элемент или группу		
<del></del>	Пометить элемент/ Снять пометку		
<f5></f5>	История значения		
<alt>+&lt;Д&gt;+<a></a></alt>	Найти в дереве		
<ctrl>+<f5></f5></ctrl>	Перенести в другую группу		
<alt>+&lt;Д&gt;+&lt;Ч&gt;</alt>	Подчиненный справочник		
Сортировка			
<ctrl>+&lt;1&gt;</ctrl>	Сортировка справочника по коду		
<ctrl>+&lt;2&gt;</ctrl>	Сортировка справочника по наименованию		

Клавиша (сочетание клавиш)	Действие	Примечание
	Сортировка	
<ctrl>+&lt;3&gt;</ctrl>	Сортировка справочника по реквизиту	
<ctrl>+&lt;^&gt;</ctrl>	На уровень вверх	
Ctrl+<↓>	На уровень вниз	
	Отладчик	
<e9></e9>	Установить точку останова	
<ctrl>+<shift>+<f9></f9></shift></ctrl>	Отключить точку останова	
<f11></f11>	Запуск "1С:Предприятия"	
<f8></f8>	Шагнуть в	Если следующим выполняе- мым оператором модуля является вызов функции или процедуры, начинается ее пошаговое выполнение, ина- че Отладчик переходит к следующему оператору
<f5></f5>	Продолжить	Прервать пошаговое выпол- нение модуля и продолжить свободное выполнение
<f10></f10>	Шагнуть через	Если следующим выполняе- мым оператором модуля является вызов функции или процедуры, она выполняется целиком (не пошагово) и Отладчик переходит к сле- дующему оператору
<shift>+<f7></f7></shift>	Шагнуть из	Прервать пошаговое выпол- нение функции или про- цедуры и остановиться на первом операторе, находя- щемся после ее вызова
<f7></f7>	Идти до курсора	Прервать пошаговое выпол- нение модуля, выполнять все операторы до той строки, на которой стоит курсор

окончание	
entern landie,	

Клавиша (сочетание клавиш)	Действие	Примечание
	Windows	
WB (логотип Windows)	Открывает главное меню	
<wb>+<e></e></wb>	Вызывает проводник	
<wb>+<f></f></wb>	Вызывает поиск файлов	
<wb>+<m></m></wb>	Сворачивает все окна	
<wb>+<shift>+<m></m></shift></wb>	Снова их развертывает	
<wb>+<r></r></wb>	Вызывает окно <b>Выполнить</b>	
<wb>+<break></break></wb>	Вызывает окно <b>Система</b>	
<wb>+<tab></tab></wb>	Особым образом переклю- чает задачи	
<ctrl>+<a></a></ctrl>	Выделить все	
<ctrl>+<f></f></ctrl>	Поиск	
<ctrl>+<h></h></ctrl>	История	
<ctrl>+<l></l></ctrl>	Диалоговое окно открытия документа	
<ctrl>+<b></b></ctrl>	Упорядочить избранное	
<ctrl>+<w></w></ctrl>	Закрыть текущее окно	
<shift>+<f10></f10></shift>	При работе с проводником	То же самое, что и щелчок правой кнопкой мыши
<doubleclick></doubleclick>		По активной панели меняет его режим с полноэкранного режима на окно и наоборот
<alt>+<doubleclick></doubleclick></alt>		Вызывает окно свойств

#### Примечание

Собрано из http://klerk.ru/soft/1c/?8362 и авторская доработка.

# Предметный указатель

### A

Автоматизация учета 167 Автоматическое заполнение реквизитов 127 Агрегатный тип данных 181 Алгоритм: расчета себестоимости 237 формирования проводок 221 Аналитический учет 193, 205 Архив расчетов 299

### Б

Базовые объекты системы 56 Бухгалтерские итоги 233

### B

Валютный учет 194, 201 Вид обращения "через точку" 127 Виды расчета 17 вытесняющие 330 по начислениям 322 по удержаниям 322 Внешний отчет 386 Временный расчет 273 Выборка 95 данных с произвольными группировками 281 оперативных итогов 284 Вычисляемые реквизиты документа 133

### Γ

Главный интерфейс 414 Глобальные процедуры 137 Глобальный: модуль конфигурации 136 параметр 137 Граница последовательности 280 Группы расчетов 17

### Д

Движения регистра 262 Дерево метаданных 24 Диапазон периодов 363 Динамический массив 182 Документы 16

#### Ж

Журнал: документов 16 общий 157 подчиненных документов 158 полный 158 расчетов 17, 351 объект 301 период регистрации 305 период действия 305 атрибуты 301

### 3

Забалансовый учет 205 Запросы 351

### И

Идентификатор 26, 117, 409 Интерактивная настройка загрузки данных 383 Информационные базы 12 Итог по колонке 24

### K

Календари 17 Кнопка: пересчета итогов 234 установки расчета 234 Кодовая страница 13 Комментарий 26, 117 Компонента: "WEB-расширение" 4 "Бухгалтерский учет" 4 "Оперативный учет" 4 "Расчет" 4 "Управление распределенными информационными базами" 4 Константы 16 Конструктор: запросов 178 отчета 178 печати 145 Конфигуратор 14, 21 Конфигурация 9

### Μ

Метаданные 15 Метод подбора 129 Методы встроенного языка 422 Многовалютный учет 193 Модуль: проведения 215 формы обработки 73

### H

Набор прав пользователя 419 Настройка прав доступа 420

### 0

Оборот ресурса 260 Обработки 16, 64 Оперативный учет 259 Операция 17, 194 Организация выборки 95 Отладчик 401 Отчеты 16, 57

### Π

Палитра свойств 26 Панели инструментов 418 Пароль для входа 53 Переменная запроса 365 Перепроведение 280 Перерасчет 307 Перечисление 16, 81, 100 Период запроса 351 Периодическая: константа 71 реквизиты 313 План счетов 17, 57, 193, 204 Полчиненный: интерфейс 414 справочник 94 счет 204 Пользовательские: права 45 интерфейс 409 Порядок группировки данных 353 Пошаговое выполнение программы 405 Правила перерасчета 307 Предопределенные процедуры 41 Приоритет вида расчета 306 Проведение документа 120 Проверка синтаксиса программы 38 Проводка 17, 194

### Р

Расчет заработной платы 299, 311 Регистр 17, 259 взаиморасчетов 271 реквизиты 260 ресурсы 260 измерения 260 Реестр 398 Режимы: для построения отчета 98 архивирования информационной базы 34 работы программы "1С:Предприятие" 11 Реквизиты условно постоянные 313

### C

Секции отчета 100 Синоним 26, 117 Синтакс-помощник 39 Список: пользователей 45 проводок документа 215 Справочники 16, 56, 81 Субконто 194 виды 17 Субсчет 204 Счет-группа 204

### Т

Таблица: движений 260 значений 181 остатков 260, 263 Технологическая платформа 8 Типообразующий объект метаданных 112 Точка: актуальности итогов 276 останова 402

### У

Универсальные отчеты 393 Управляющие кнопки 11 Уровень вложенности счета 204 Установка драйвера защиты 7 Установки режима вывода 374

### Ц

Циклическая обработка строк запроса 289

### Ш

Шаблон печатной формы 98, 150

#### Э

Экранная форма обработки 65 Экспорт данных 375