

1С: Предприятие

Конфигурирование и администрирование
для начинающих



Э К С П Р Е С С
К У Р С

- ▶ Константы, справочники, документы, перечисления
- ▶ Запросы и отчеты
- ▶ Меню и панели инструментов
- ▶ Программные модули на встроенном языке
- ▶ Связь с внешними приложениями посредством OLE Automation

Владимир Дубянский

1С:Предприятие

**Конфигурирование и администрирование
для начинающих**

**ЭКСПРЕСС-
КУРС**

Санкт-Петербург

«БХВ-Петербург»

2005

УДК 681.3.06
ББК 32.973.26-018.2
Д79

Дубянский В. М.

Д79 **1С:Предприятие.** Конфигурирование и администрирование для начинающих. Экспресс-курс. — СПб.: БХВ-Петербург, 2005. — 176 с.: ил.
ISBN 5-94157-278-6

Описаны основные приемы программирования и конфигурирования, общие для всех компонент системы "1С:Предприятие" версии 7.7. 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. Приведены примеры конфигураций, внедренных на предприятиях.

Для широкого круга программистов и пользователей системы "1С:Предприятие"

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. гл. редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Елена Кашлакова</i>
Компьютерная верстка	<i>Натальи Караваевой</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Дизайн обложки	<i>Игоря Цырульниковой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 22.10.04.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 14,19.

Тираж 5000 экз. Заказ № 585

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

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

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

ISBN 5-94157-278-6

© Дубянский В. М.. 2005
О Оформление, издательство "БХВ-Петербург", 2005

Содержание

Введение	1
Глава 1. Вводные сведения	3
1.1. Почему именно "1С:Предприятие"?.....	3
1.2. Концепция системы "1С:Предприятие".....	3
1.3. Инсталляция.....	7
1.4. Что такое конфигурация.....	8
1.5. Функционирование системы.....	8
1.6. Создание новой конфигурации.....	9
1.7. Объекты метаданных.....	13
Глава 2. Начинаем новый проект	17
2.1. Константы.....	17
2.2. Периодические константы.....	20
2.3. Справочники.....	22
2.4. Документы.....	31
Глава 3. Начала программирования в системе "1С:Предприятие"	41
3.1. Встроенный язык системы.....	41
3.2. Программный модуль и простейшие программы.....	41
3.3. Заполнение документа методом подбора.....	49
3.4. Локальный контекст и доступность процедур и функций.....	56
3.5. Итоговые значения колонок документа.....	60
Глава 4. Вывод на печать	63
4.1. Понятие о выборке.....	63
4.2. Объект <i>Таблица</i> для построения отчета.....	65
4.3. Печатные формы справочников.....	68
4.4. Печатные формы документов.....	76
4.5. Печатные формы как отдельные файлы.....	83

Глава 5. Другие базовые объекты	87
5.1. Перечисление.....	87
5.2. Обработка.....	90
5.3. Журналы документов.....	96
5.4. Список значений.....	99
Глава 6. Запросы и отчеты	103
6.1. Начало создания отчета.....	103
6.2. Запрос.....	105
Глава 7. Пользовательский интерфейс и администрирование	113
7.1. Индивидуальный пользовательский интерфейс.....	113
7.2. Создание набора пользовательских прав.....	120
7.3. Создание нового пользователя.....	123
7.4. Заключительные замечания.....	126
Глава 8. Некоторые дополнительные сведения	127
8.1. Специальные свойства объекта конфигурации.....	127
8.2. Сохранение базы данных.....	127
8.3. Перенос справочников с данными.....	133
8.4. Загрузка данных в "1С:Предприятие" из Microsoft Excel.....	141
8.5. Экспорт данных из "1С:Предприятие" в Microsoft Excel.....	147
Глава 9. Примеры конфигураций, написанных на базовых объектах	157
9.1. Вводные замечания.....	157
9.2. Конфигурация Интернет-поддержка.....	157
9.3. Конфигурация Учет абонентской платы.....	163
Предметный указатель	170

Введение

Данная книга предназначена для изучения элементарной работы в конфигураторе "1С:Предприятие". Обучившись несложным вещам, вы достигнете нескольких целей:

- О не будете жестко привязаны к программисту или фирме-разработчику — большинство проблем решите сами;
- П. будете четко осознавать, что вам нужно от разработчика, как это удобнее для вас сделать, сколько это может занять времени и сколько стоит;
- П значительно повысится ваш статус как специалиста.

Конфигуратор "1С:Предприятие" — это эффективный инструмент продвинутого бухгалтера. Для качественной работы надо уметь использовать все возможности "1С:Предприятие", помимо работы в пользовательском режиме, конфигурировать и администрировать.

Основной упор сделан на разбор основополагающих моментов конфигурирования. Предполагается, что наличие такой обучающей книги позволит ускорить освоение технологии конфигурирования в "1С:Предприятие 7.7". Сквозной пример (www.bhv.ru/books/get_disk_data.php?id=4789), который мы будем разбирать на протяжении всей книги, вполне работоспособная конфигурация, правда, очень простая. Однако, имея уровень знаний, полученный в результате изучения этой книги, вы сможете самостоятельно совершенствоваться, улучшать рабочие конфигурации и писать новые.

Книга не является документацией, либо дополнением к документации к программному продукту. Для разработки приложений, безусловно, необходимо наличие документации, входящей в комплект поставки, в частности:

П конфигурирование и администрирование;

описание встроенного языка;

руководство по установке и запуску.

В процессе работы над данным пособием автор использовал материалы указанных книг, а также:

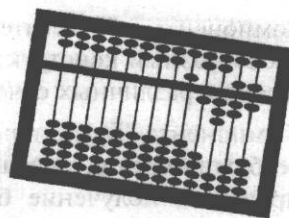
1. Байдаков В., Нуралиев С., Шевченко А. Введение в конфигурирование 1С бухгалтерии 7.7. - "1С", 2000.

2. Лихарев М., Нуралиев С., Русанов Д., Службин Д., Федоров Б., Филиппов В. "1С:Предприятие", версия 7.5. Методические рекомендации по конфигурированию и администрированию. — "1С", 1998.
3. Гвоздев Д. В., Гордеев О. В., Меломуд О. Л., Ожигов В. Л. Методические рекомендации по анализу и построению технической архитектуры в проектах внедрения информационных систем на базе системы программ "1С:Предприятие". — "1С", 2000.
4. Соловьев И. В. "1С:Предприятие" 7.7. Базовые объекты". — "1С", 2000.
5. Материалы сайта <http://training-1c.hotbox.ru>.

Автор предполагает, что читатели имеют элементарные навыки работы с персональным компьютером в среде MS Windows, то есть представляют структуру каталогов, могут создавать, копировать, удалять файлы и папки, знакомы с понятиями имени файла, расширения файла и пр.

Если, по мере прочтения книги, вы обнаружите какие-либо упущения, неточности, сообщите о них автору. Автор будет благодарен за замечания и предложения по развитию данной тематики.

Глава 1



Вводные сведения

1.1. Почему именно "1С:Предприятие"?

На рынке бухгалтерских программ достаточно большое разнообразие. "1С:Предприятие", по моему личному мнению, выгодно отличается от остальных следующими возможностями:

- О Очень гибкая настройка. Трудно представить себе изменения в законодательстве, в плане счетов и прочие, которые нельзя было бы реализовать в системе "1С:Предприятие", причем с сохранением всех ранее введенных данных.
- Возможность решения широкого круга учетных задач: бухгалтерия и торговля, управление персоналом, финансовый анализ, управление проектами и др.
- Открытая система — возможно взаимодействие с другими программами, обмен данными, написание внешних компонент, расширяющих возможности программы.
- Учет может вестись как на маленькой фирме с одним компьютером, так и на больших предприятиях, с сотнями рабочих мест.
- Возможность синхронизировать учет в территориально разделенных организациях, причем передача данных возможна любым путем: от дискеты до оптоволоконного кабеля.
- О Наиболее широкая и частая франчайзинговая сеть, а также развитая система поддержки конечных пользователей.

1.2. Концепция системы "1С:Предприятие"

Стандартное представление о программах фирмы "1С" — "1С:Бухгалтерия". Однако "1С:Предприятие" является системой программ для автоматизации различных областей экономической деятельности. "1С:Предприятие" имеет компонентную структуру. В настоящий момент существует три основные

компоненты: "Бухгалтерский учет", "Оперативный учет" и "Расчет". Это своеобразный конструктор из частей-компонент, которые по отдельности или в самых различных сочетаниях позволяют 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 законодательством и методологией учета.

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

Компонента "Оперативный учет" предназначена для учета наличия и движения средств в самых 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.

Обратим особое внимание на различия этих компонент. К примеру, пользуясь компонентой оперативного учета, можно эффективно учитывать товарно-материальные ценности на складах и вести торговлю, но нельзя создавать бухгалтерские операции и проводки. Если вам нужны проводки и другие моменты бухгалтерского учета, используйте еще и компоненту "Бухгалтерский учет". Вместе с тем, при небольших объемах торговых операций можно обойтись вообще одной компонентой "Бухгалтерский учет". Она позволяет вести бухгалтерский учет и расчет зарплаты. Сложную зарплату лучше начислять с помощью компоненты "Расчет".

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

По функциональной полноте "1С:Предприятие" различается следующими формами поставки:

О *Базовая*. Отличительной особенностью этой поставки является отсутствие возможности производить настройку в режиме конфигуратора.

- Стандарт.** В этой поставке:
 - 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.** Действуют 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 системы форма поставки может быть:

- локальная — на один компьютер. С информационной базой сможет работать только один пользователь;
- сетевая — неограниченное количество компьютеров;
- для трех пользователей (Оперативный учет и Расчет);
- SQL-версия — неограниченное количество компьютеров с использованием клиент-серверной технологии;
- набор для небольшой фирмы: сетевая бухгалтерия, Оперативный учет и Расчет для трех пользователей.

Примечание

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

Примеры поставок:

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

Выбрав, к примеру, инсталлятор "1С:Бухгалтерский учет", вы получаете компоненту "1С:Бухгалтерия", а если это был инсталлятор "1С:Предприятие", то сразу все три компоненты, возможен и промежуточный вариант. Возможно, для нужд вашего предприятия потребуется использовать две компоненты — "1С:Бухгалтерский учет" и "1С:Оперативный учет", а вот компо-

нента "1С:Расчет" не нужна. Что тогда делать? Тогда вы берете инсталлятор "1С:Бухгалтерский учет", затем инсталлятор "1С:Оперативный учет", причем устанавливаете компоненту обязательно в тот же каталог. При этом 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 уже сейчас оценить с профессиональной точки зрения то, что вы имеете, или запланировать, что нужно приобрести.

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

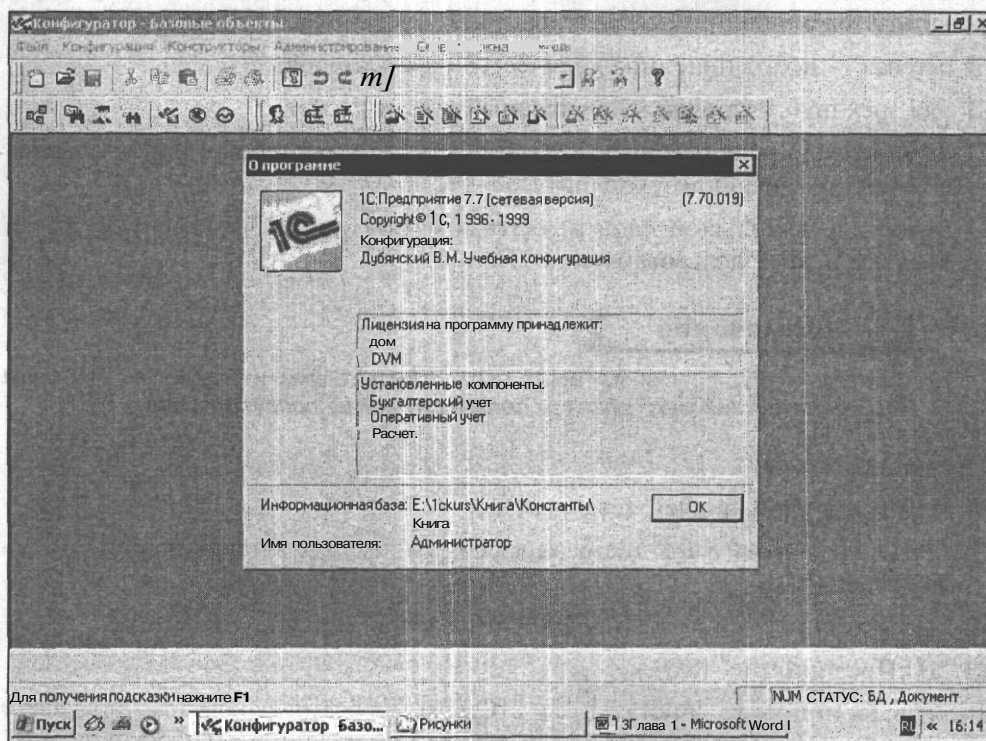


Рис. 1.1. Информация о программе

1.3. Инсталляция

Инсталляция программы поставляется на дискетах или компакт-диске. Компакт-диски или дискеты содержат программу установки Setup. Все версии работают под управлением операционной системы MS Windows. При первой установке программа Setup запрашивает имя пользователя и наименование компании. Программа сохраняет данную информацию и выводит ее каждый раз при запуске. Далее необходимо следовать инструкциям на экране. Когда программа инсталляции предложит 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 будем. При необходимости можете обратиться к документации программы. После инсталляции программы выключите компьютер, вставьте в порт принтера ключ защиты, если есть принтер, в ключ защиты вставьте разъем кабеля принтера, включите компьютер. Затем нажмите кнопку **Пуск** (Start), далее — **Программы | 1С:Предприятие 7.7 | Установка драйвера защиты** (рис. 1.2). Появится сообщение об успешной установке драйвера защиты. Теперь перезагрузите компьютер.

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

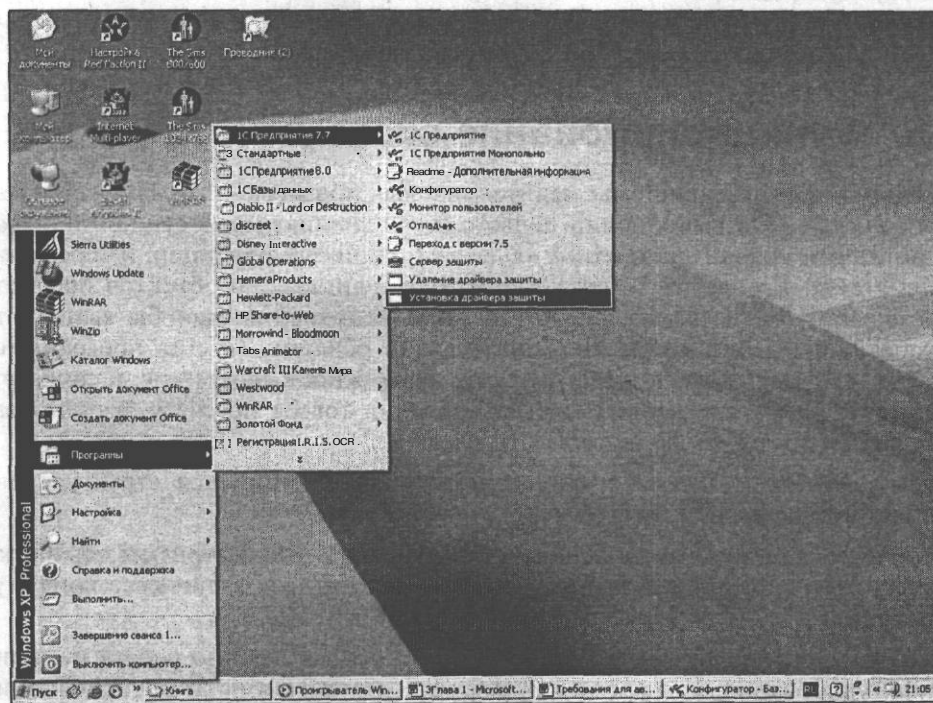


Рис. 1.2. Выбор пункта меню **Установка драйвера защиты**

1.4. 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, не зависящих от конкретного законодательства и методологии учета. Это "мотор" автоматизации предприятия. Действительно, ведь "1С:Предприятие" используется во всех странах СНГ и, бывает, дальнего зарубежья. Разработчикам невозможно предусмотреть все нюансы. А вот сам алгоритм 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 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.

1.5. Функционирование системы

Функционирование системы делится на два процесса — *конфигурирование* и *исполнение*. При выполнении первого этапа специалист, в сущности, строит описание модели предметной области (например, бухгалтерии) средствами системы. Его задача — создать такую конфигурацию объектов предметной области, чтобы пользователь в режиме исполнения смог бы выполнять обработку данных в 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.

В этом смысле в процессе конфигурирования формируется структура информационной базы и алгоритмы ее обработки.

В процессе же исполнения используются как штатные алгоритмы обработки данных (технологическая платформа), так и созданные 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.

1.6. Создание новой конфигурации

Для начала работы с собственной конфигурацией создайте на диске компьютера папку, в которой будет 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 выполнить следующую последовательность команд: **Пуск | Программы | 1С Предприятие 7.7 | 1С Предприятие**.

После поиска ключа защиты на экран будет выведен диалог запуска (рис. 1.3).

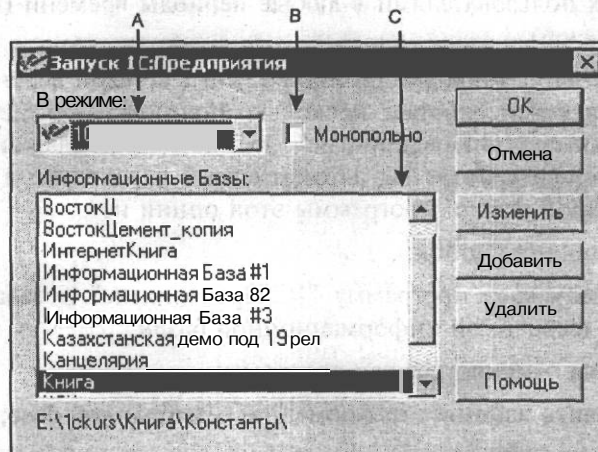


Рис. 1.3. Запуск системы "1С:Предприятие"

Исследуем появившееся окно. Область А определяет режим запуска программы. Область В используется в сетевой версии для запуска в монопольном режиме. Область С позволяет выбрать необходимую информационную базу, с которой пользователь собирается работать в данном сеансе.

Существует четыре режима работы программы:

- ❑ **"1С:Предприятие"** — основной режим работы с программой. В этом режиме работают пользователи системы, вводя первичную информацию и формируя необходимые отчеты. Работать в этом режиме 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 и восстановление данных, настройка списка пользователей, тестирование и исправление информационной базы).

- **Отладчик** — этот режим служит для облегчения разработки конфигурации. Имеются следующие возможности:
 - пошаговое выполнение модулей конфигурации;
 - расстановка точек останова;
 - прерывание и продолжение выполнения модуля;
 - вычисление выражений для анализа состояния переменных.

○ **Монитор** — можно просматривать список активных пользователей. Кроме того, монитор позволяет анализировать журнал регистрации действий, выполняемых пользователями в любые периоды времени (историю работы пользователей).

Некоторые действия с информационной базой в сетевой версии невозможно выполнить, если в ней работает несколько пользователей. Для такого рода действий в режиме "1С:Предприятие" предназначен флажок **Монопольно**. При работе в конфигураторе вы автоматически переходите в монопольный режим. В локальной версии программы этой опции нет.

Область управляющих кнопок:

- кнопка **ОК** запускает программу "1С:Предприятие" в заданном режиме для работы с выбранной информационной базой;
- кнопка **Отмена** отменяет запуск программы;
- кнопка **Изменить** изменяет информацию о выбранной базе;
- кнопка **Добавить** добавляет новую информационную базу. Следует ввести имя информационной базы и путь к конфигурации. Путь указывается так, чтобы в выбранном каталоге обязательно оказался файл настройки `1cv7.md`;
- кнопка **Удалить** удаляет информационную базу. Причем удаление производится не с диска, а только из списка доступных информационных баз. При желании можно восстановить эту базу кнопкой **Добавить**.

1.6.1. Информационные базы

Система "1С:Предприятие" позволяет работать с множеством информационных баз. Это могут быть базы одного предприятия за разные периоды или же текущие (рабочие) базы разных предприятий. При запуске программы вы должны выбрать нужную информационную базу нажатием на строке с названием базы. На рис. 1.4 выбрана информационная база "Книга", в нижней строке окна указан путь к информационной базе. В этом каталоге лежит вся информация, касающаяся данной базы.

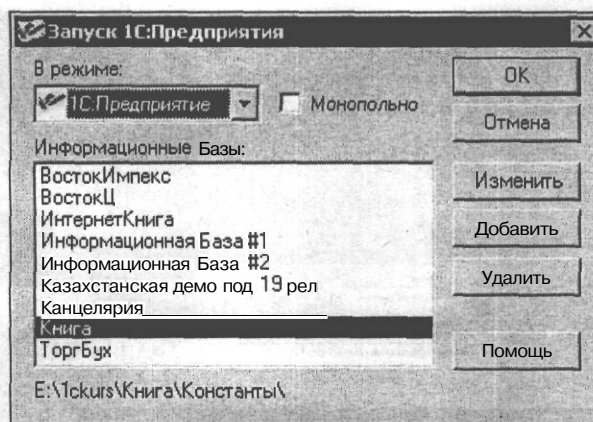


Рис. 1.4. Информационные базы системы

1.6.2. Регистрация информационной базы и запуск системы в режиме конфигуратора

Для того чтобы создать новую информационную базу, необходимо нажать на кнопку **Добавить** и в появившемся окне указать название и путь к предварительно созданному каталогу информационной базы (рис. 1.5).

После регистрации необходимо запустить конфигуратор, так как в нашем каталоге еще ничего нет. Поэтому мы нажмем на кнопку **ОК**. Система предложит выбрать формат хранения и *кодovou страницу* (см. рис. 1.6).

При создании новой конфигурации необходимо выбрать кодovou страницу, совпадающую с настройкой Windows, которая будет использоваться при запуске конфигурации. Рекомендуется в диалоге, который изображен на рис. 1.6, оставить кодovou страницу, предлагаемую системой, это значение соответствует Windows, и нажать на кнопку **ОК**. Если сделан выбор кодовой страницы и порядок сортировки, отличный от текущих системных значений, то корректное функционирование невозможно. В этом случае, скорее всего, будет выдано сообщение: "Порядок сортировки строк, установленный для базы данных, отличается от системного", после чего работа системы "1С:Предприятие" будет завершена. Следует изменить либо установки операционной системы, либо кодovou страницу информационной базы. Об этих изменениях будет сказано в гл. 7 книги, посвященной администрированию системы.

Замечание

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

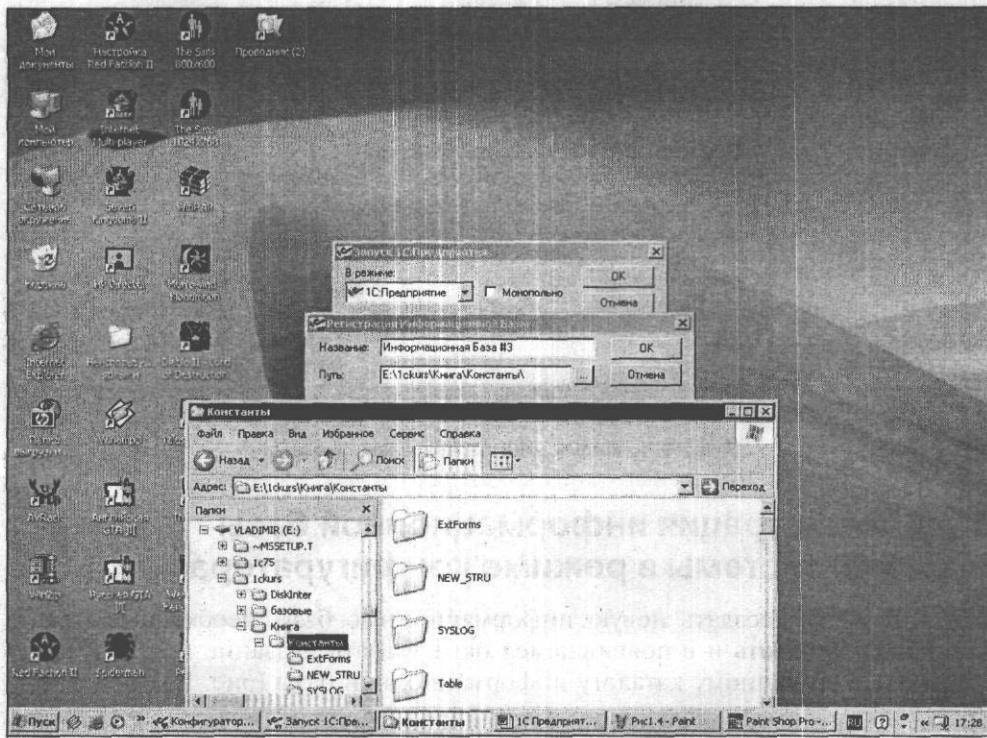


Рис. 1.5. Регистрация информационной базы



Рис. 1.6. Выбор кодовой страницы

После выбора кодовой страницы откроется окно configurатора. Здесь также есть меню, панели инструментов, рабочая область и строка состояния (рис. 1.7). Это основное окно, с которым вам придется иметь дело на протяжении всего сеанса работы с configurатором. Его можно открыть, либо используя пункт меню **Открыть конфигурацию**, либо нажатием соответствующей кнопки на панели инструментов.

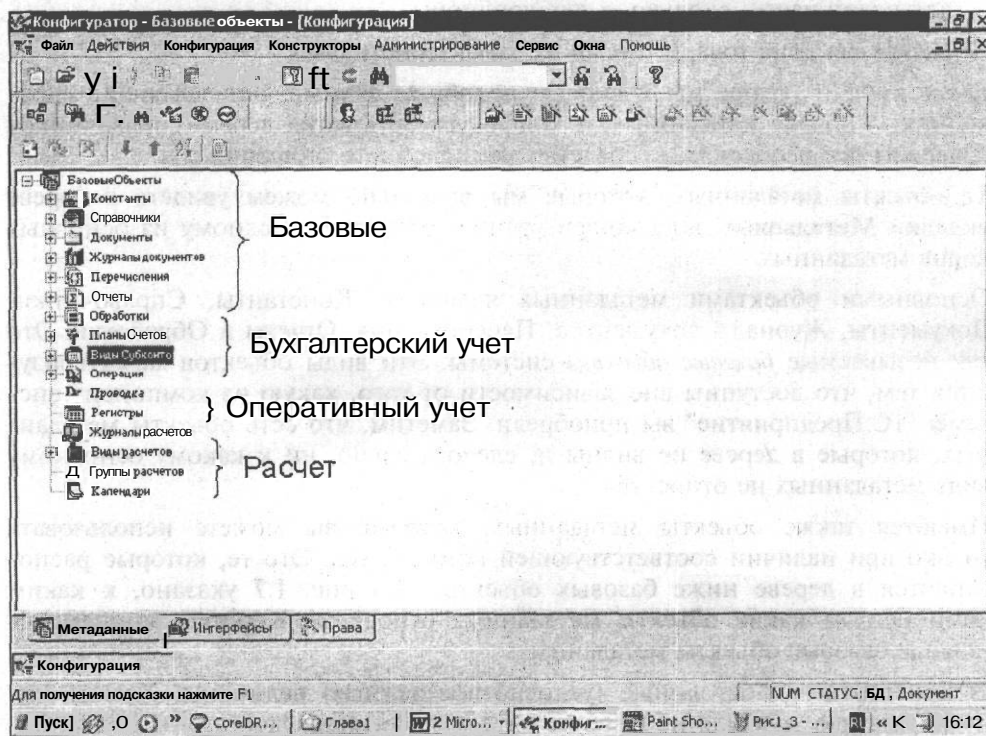


Рис. 1.7. Окно configurатора и группы метаданных конфигурации

В этом окне имеются три вкладки: **Метаданные**, **Интерфейсы**, **Права**. Пока мы будем работать с метаданными.

1.7. Объекты метаданных

В окне configurатора размещается стандартное дерево метаданных. Метаданные — это ветви дерева: **Константы**, **Справочники**, **Документы** и т. д. Объект метаданных — это некоторое формальное описание группы понятий предметной области со сходными характеристиками и одинаковым предназначением. Расшифруем эти определения на таком объекте метаданных, как

справочник. Допустим, необходимо создать справочник товаров. Какие данные он должен содержать? Как минимум, следующие:

- наименование товара, причем надо указать, что оно пишется символами (буквами, цифрами), и сколько символов отводится для этой записи;
- приходная цена товара, надо указать, что это числа, с запятой, разделяющей целую и дробную части, и обозначить, сколько цифр отводится для целой части, сколько — для дробной;
- расходная цена товара, с теми же характеристиками.

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

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

Основными объектами метаданных являются: Константы, Справочники, Документы, Журналы документов, Перечисления, Отчеты и Обработки. Это так называемые *базовые объекты* системы. Эти виды объектов характеризуются тем, что доступны вне зависимости от того, какую из компонент системы "1С:Предприятие" 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 базовые объекты метаданных:

- константы — постоянные (условно-постоянные) величины. Хранят информацию, которая не изменяется или изменяется достаточно редко: название организации, ее почтовый адрес и др.;
- справочники — списки однородных элементов данных для хранения нормативно-справочной информации;
- документы — служат для ввода информации о совершенных хозяйственных операциях;
- журналы — списки объектов данных для работы с документами;
- перечисления — списки значений, задаваемые на этапе конфигурации;
- отчеты — форма организации выходной информации;
- обработка — используется для выполнения различных действий над информационной базой.

Компонента "Оперативный учет" содержит регистры — средство накопления оперативной информации о наличии и движении средств.

Компонента "Бухгалтерский учет" оперирует следующими объектами:

- планы счетов — списки учетных регистров, по которым будет выполняться группировка средств при работе в системе "1С:Предприятие";
 - виды субконто — списки объектов для организации аналитического учета средств предприятия в совокупности с бухгалтерскими счетами;
 - операция — средство отражения в бухгалтерском учете хозяйственных операций, происходящих на предприятии. Это совокупности бухгалтерских проводок, выполняющие изменения в остатках и оборотах по бухгалтерским счетам;
- П проводка — предназначена для отражения в бухгалтерских итогах результатов хозяйственных операций.

Компонента "Расчет" содержит следующие объекты:

- П журналы расчетов — средство проведения периодических расчетов и хранения результатов прошлых периодов. Каждый журнал связан с одним из справочников объектов расчета. Каждая строка журнала — единичный акт расчета, который производится по определенному алгоритму, имеет протяженность во времени и результат;

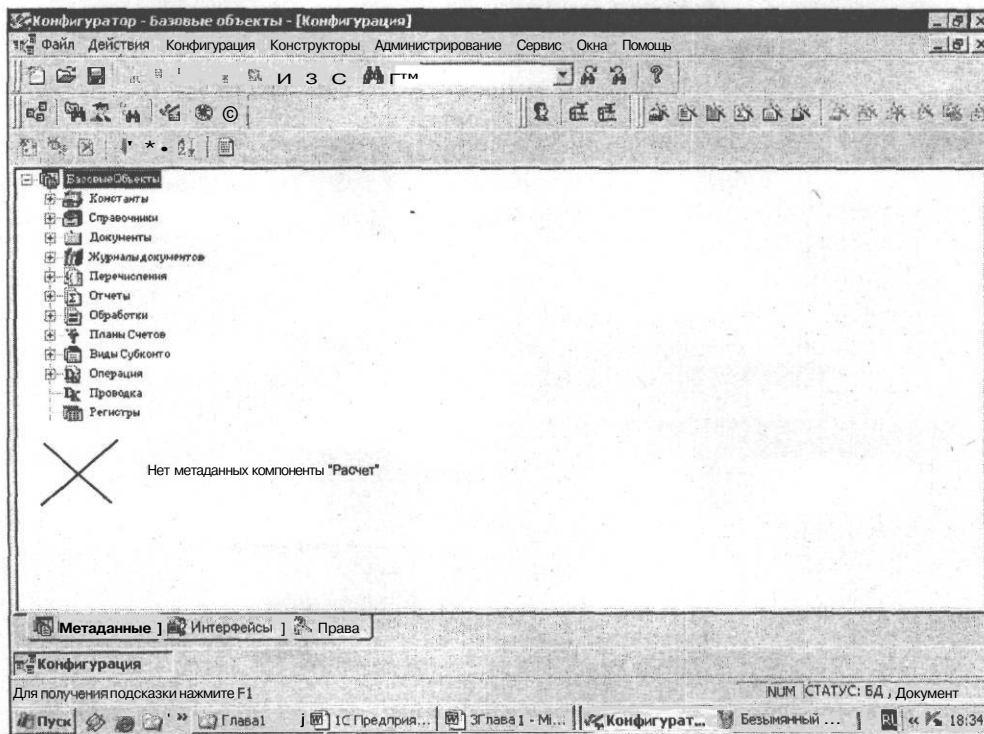
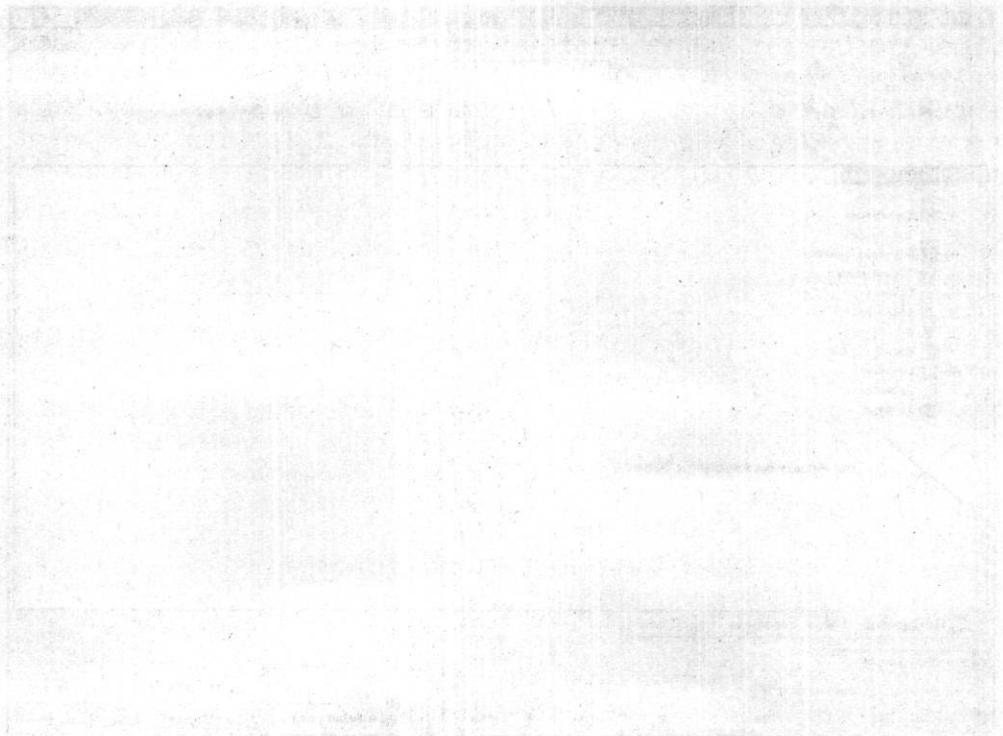


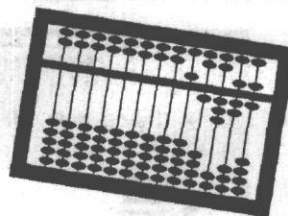
Рис. 1.8. Изменение дерева метаданных при отсутствии компоненты "Расчет"

- виды расчетов — алгоритмы, по которым могут проводиться расчеты записей журналов расчета;
- группы расчетов — средство логического объединения видов расчета по тем или иным признакам;
- календари — средство построения графиков работы предприятия, внешних организаций или отдельных категорий работников.

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



Глава 2



Начинаем новый проект

2.1. Константы

Выполним учет на небольшом предприятии, торгующем различным товаром. Начнем с того, без чего никогда и нигде невозможно обойтись — с бюрократии. 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 использоваться для хранения и обработки этих данных в системе "1С:Предприятие"? Согласно описаниям (см. гл. 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.

В открытом окне конфигурации найдите ветвь метаданных **Константы** и щелкните левой кнопкой мыши на ней, как на рис. 2.1. 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. 2.2). Зададим эти свойства.

Идентификатор — это название константы для программиста и для технологической платформы "1С:Предприятие". Как правило, идентификатором вы будете пользоваться 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. Пусть идентификатором нашей константы будет **ДатаРегистрации**.

Синоним — это другое название идентификатора, которое конечный пользователь будет видеть в окнах системы "1С:Предприятие" на месте идентификатора. Синоним запишем так: **Дата регистрации**.

Обратите внимание, что в синониме возможны пробелы.

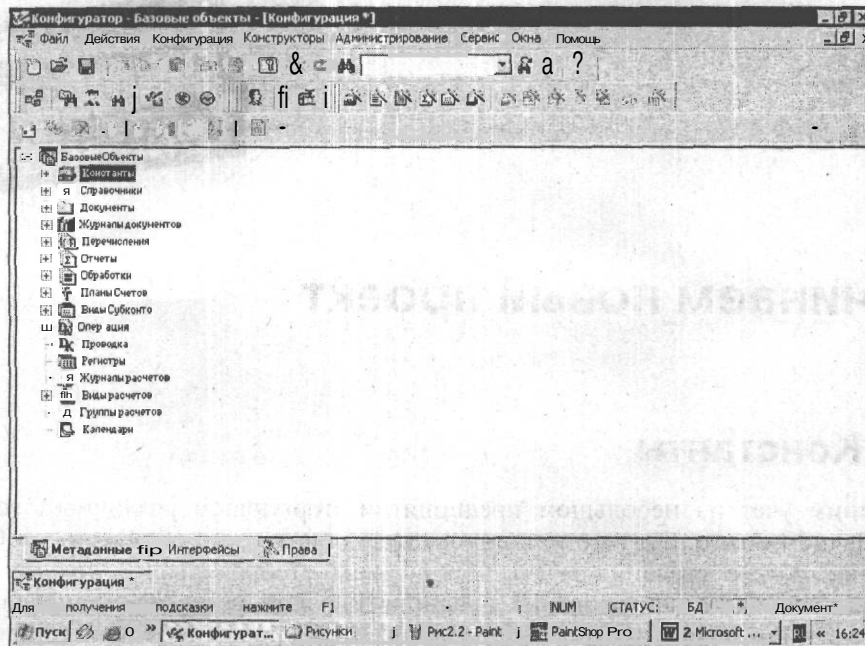


Рис. 2.1. Дерево метаданных конфигурации

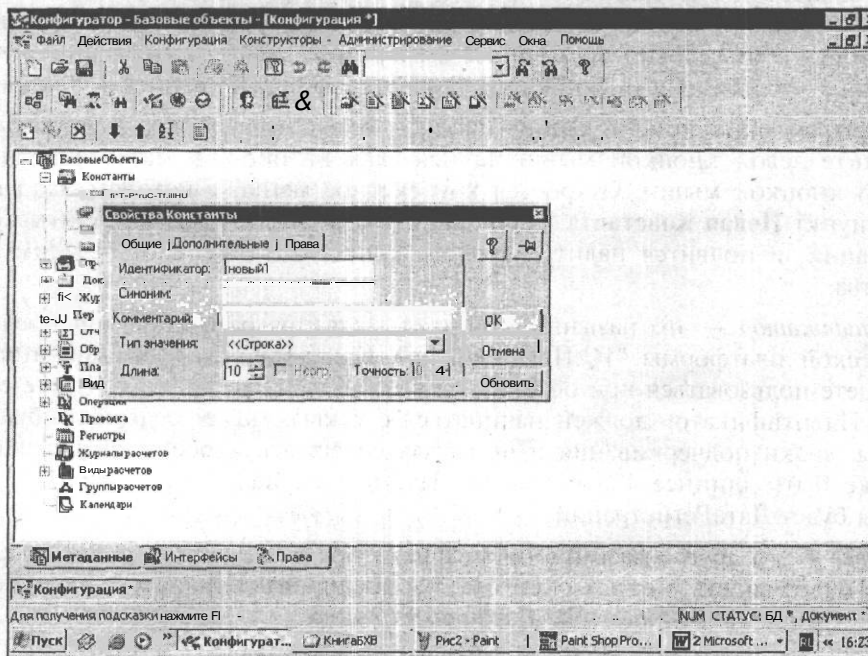


Рис. 2.2. Палитра свойств константы

Комментарий служит для расшифровки идентификатора объекта и выдается в различных диалогах системы вслед за идентификатором объекта. Комментарий может быть такой: Дата регистрации предприятия.

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

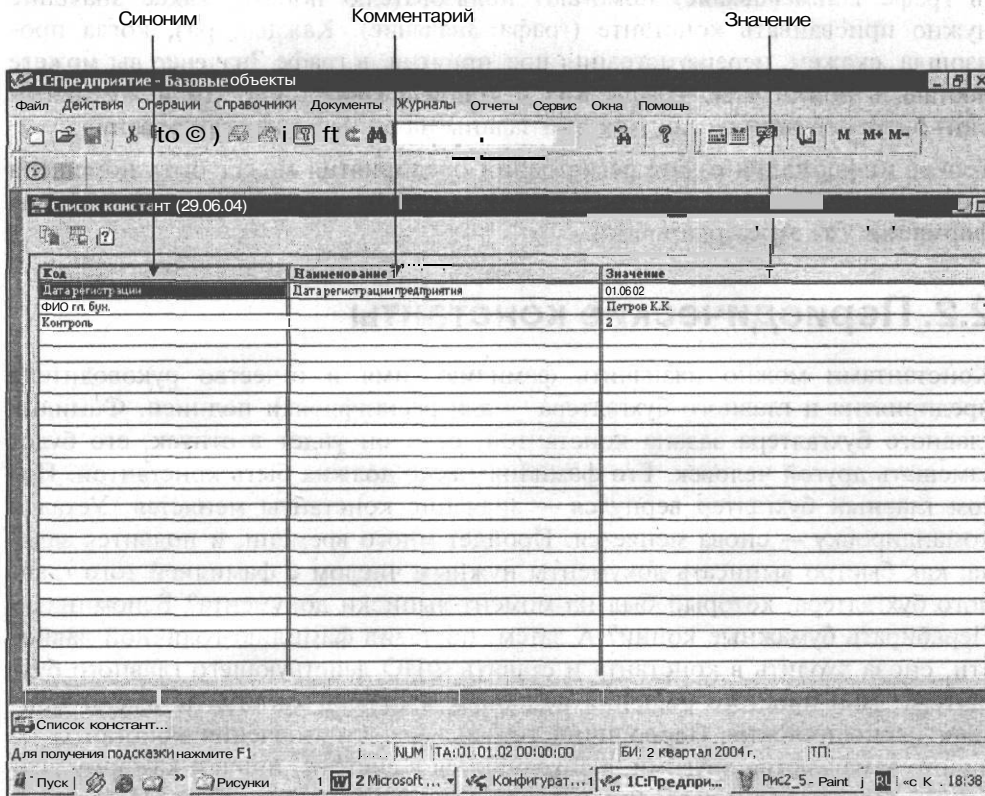


Рис. 2.3. Список констант конфигурации в пользовательском режиме

Теперь нажмем на палитре свойств кнопку ОК и запустим "1С:Предприятие" в пользовательском режиме. Это можно сделать через кнопку на панели инструментов или через пункт меню **Сервис | 1С:Предприятие**. Появится окно с сообщением: "При сохранении метаданных будет произведен анализ сделанных изменений... Выполнить сохранение метаданных?". Нажимайте кнопку Да. После этого конфигуратор выдаст сообщение: "Отсутствует старый словарь данных!". Нажимайте кнопку Да. Следующее окно показывает, какие изменения мы внесли. Проверяем — **Новая константа — ДатаРегистрации**.

Все правильно, нажимаем кнопку **Принять**. Последнее окно — сообщение: "Реорганизация информации закончена!". Нажимаем кнопку ОК, и "1С:Предприятие" переходит в пользовательский режим. Посмотрим результаты нашего конфигурирования. Выберем пункт меню (в пользовательском режиме!) **Операции | Константы....** Откроется список констант (рис. 2.3). 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.

2.2. Периодические константы

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 константой. Потом главный бухгалтер вернулся — значение константы меняется. Уехал в командировку — снова меняется. Пройдет много времени, и появится задача: как 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. 2.1):

Таблица 2.1. Значения периодической константы

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

Такие константы называются периодическими. 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 **Периодический** (рис. 2.4).

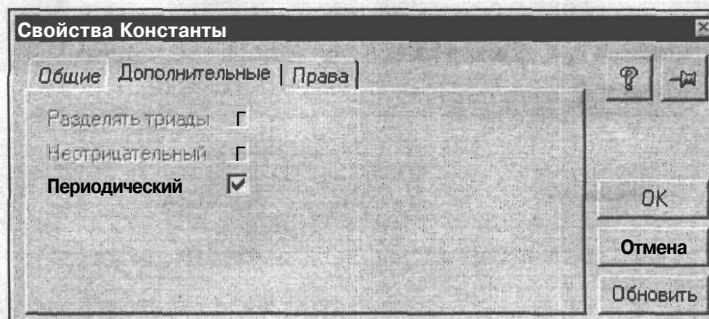


Рис. 2.4. Установка признака периодичности в свойствах константы

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. 2.5). Щелкните мышью на этой кнопке, откроется окно с таблицей, похожей на табл. 2.1, — окно истории значений константы. Заполните одну-две строки таблицы, как бы внесите значения константы задним числом. Обратите внимание, что в колонке **Дата** имеется кнопка с календарем для выбора даты. Затем закройте окно истории значений константы и внесите новое значение прямо в окне констант. Откройте историю значений константы и посмотрите — внесенному значению автоматически присвоена текущая дата системы. В дальнейшем так будет происходить постоянно.

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.

Как получить значения периодической константы, мы разберем позже.

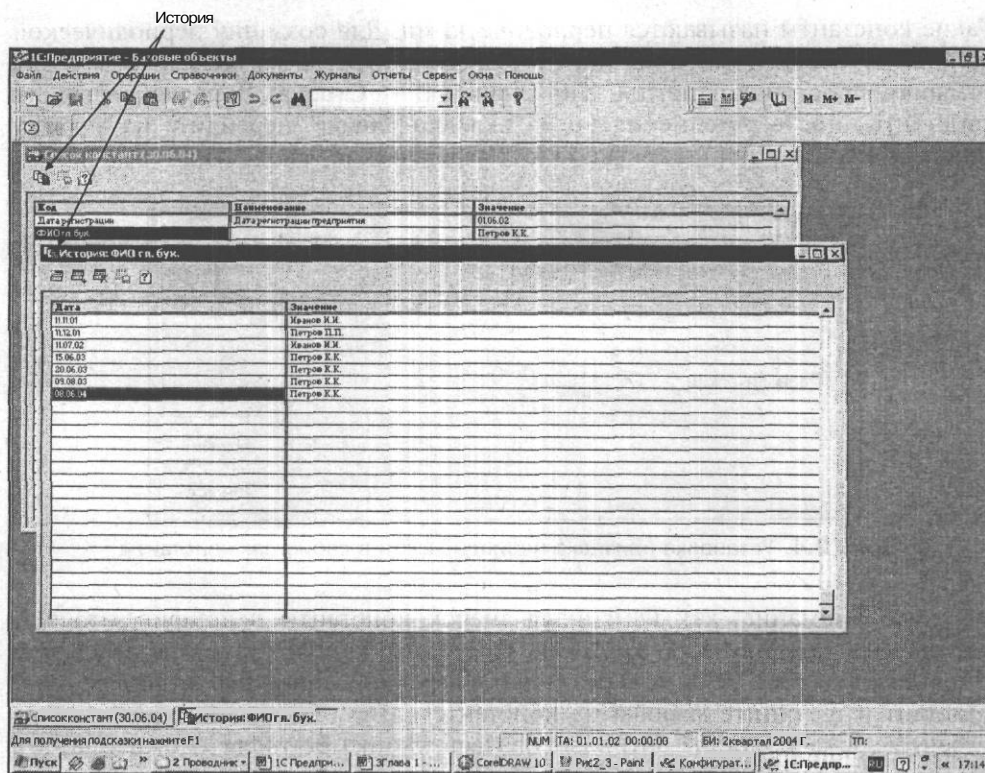


Рис. 2.5. История значений периодической константы

2.3. Справочники

Следующий шаг — подготовка справочников для ввода нормативно-справочной 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
- а Ставка НДС

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

В открытом окне конфигурации найдите ветвь метаданных **Справочники**. Щелкните левой кнопкой мыши на ней, и она выделится, как на рис. 2.6. Затем щелкните правой кнопкой мыши. Откроется контекстное меню, в котором надо выбрать пункт **Новый справочник**. Откроется первое окно конструктора справочника. Прочитайте подсказку в центре окна конструктора и заполните поля справочника следующими значениями:

- идентификатор — Номенклатура;
- синоним — Справочник номенклатуры;
- комментарий — Справочник ТМЦ.

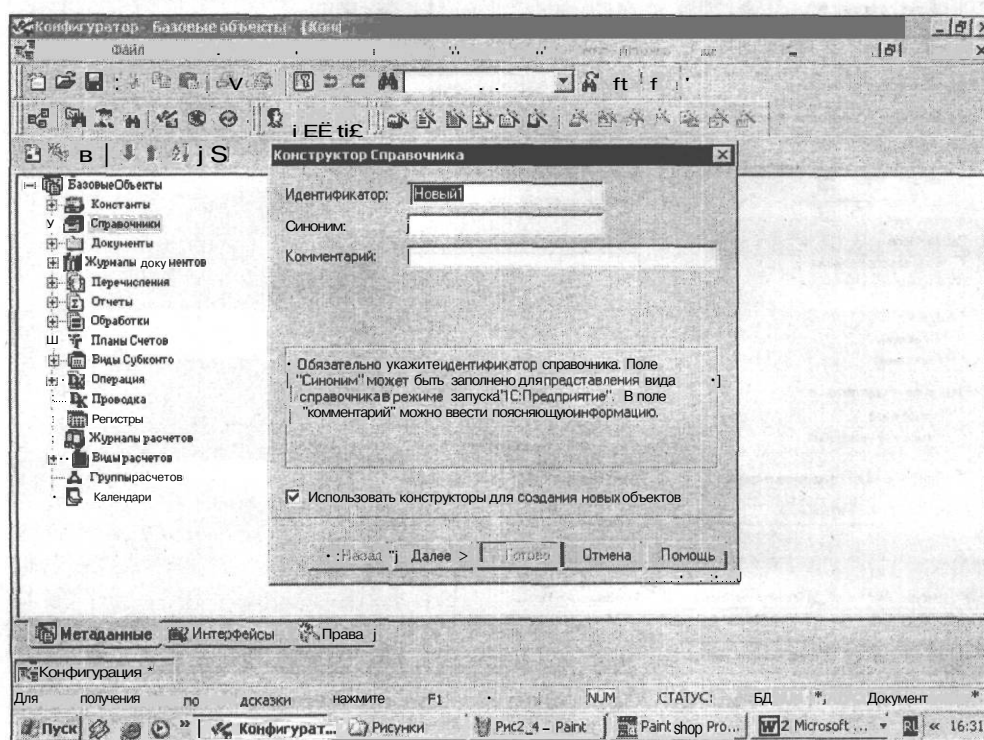


Рис. 2.6. Конструктор справочника

Замечание

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

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

Предупреждение

Вид метаданных Субконто не является базовым объектом и в дальнейшем не рассматривается.

В результате раскрывается окно редактирования справочника (рис. 2.7).

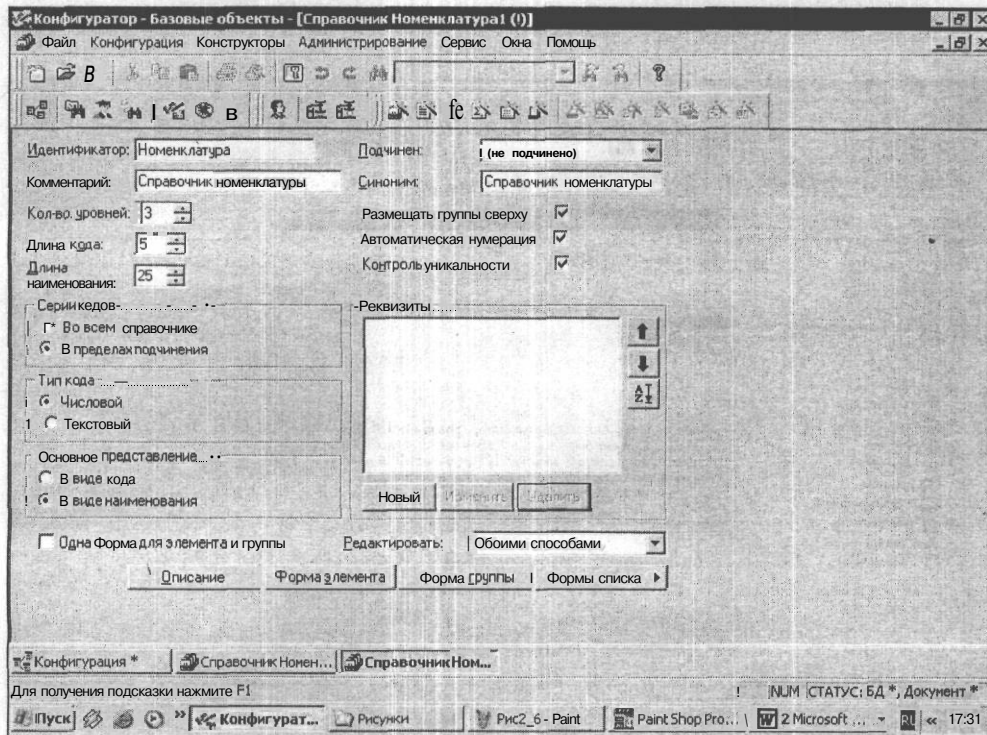


Рис. 2.7. Окно редактирования справочника

В окне выводятся и могут редактироваться идентификатор, синоним и комментарий справочника. Назначение поля **Подчинен** опишем позже. В поле

Кол-во 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. 2.8). Из опыта работы известно, что пользователю оптимально работать со справочниками, содержащими 3—5 уровней.

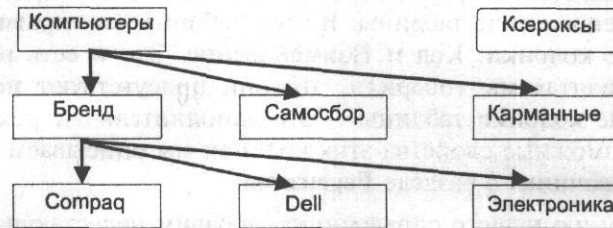


Рис. 2.8. Схема уровней справочника

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.

Длину кода оставим без изменения (5). Длину наименования поставим 50. Максимально возможная длина наименования — 100 знаков. Тип значения наименования всегда Строка. Тип значения кода может быть как Строка, так и Число. Тип значения кода осуществляется переключателем **Тип кода**. Включение флажка **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. В случае выбора 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.

Остальные свойства справочника разберем позже.

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**.

Согласно описанию нашего справочника, добавим недостающие 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**, и появится палитра свойств реквизита (рис. 2.9).

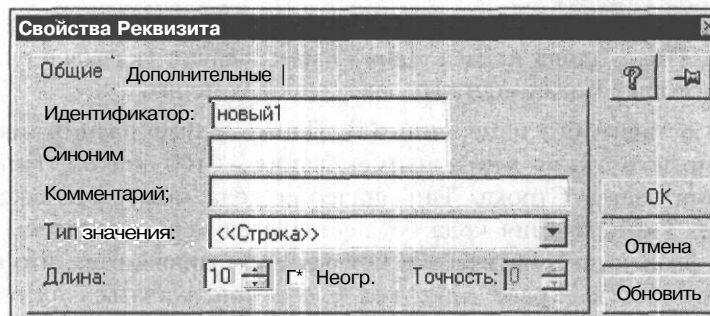


Рис. 2.9. Палитра свойств реквизита справочника

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, согласно инструкции в *разд. 2.1*. Идентификатор — ПолноеНаименование, синоним — Полное наименование, комментарий — Полное наименование. Тип значения — Строка (понятно, почему?). Длину установите 100. На вкладке **Дополнительные** ничего не устанавливайте.

Теперь создадим остальные реквизиты, согласно описанию справочника, данного 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. 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. 2.10). 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.

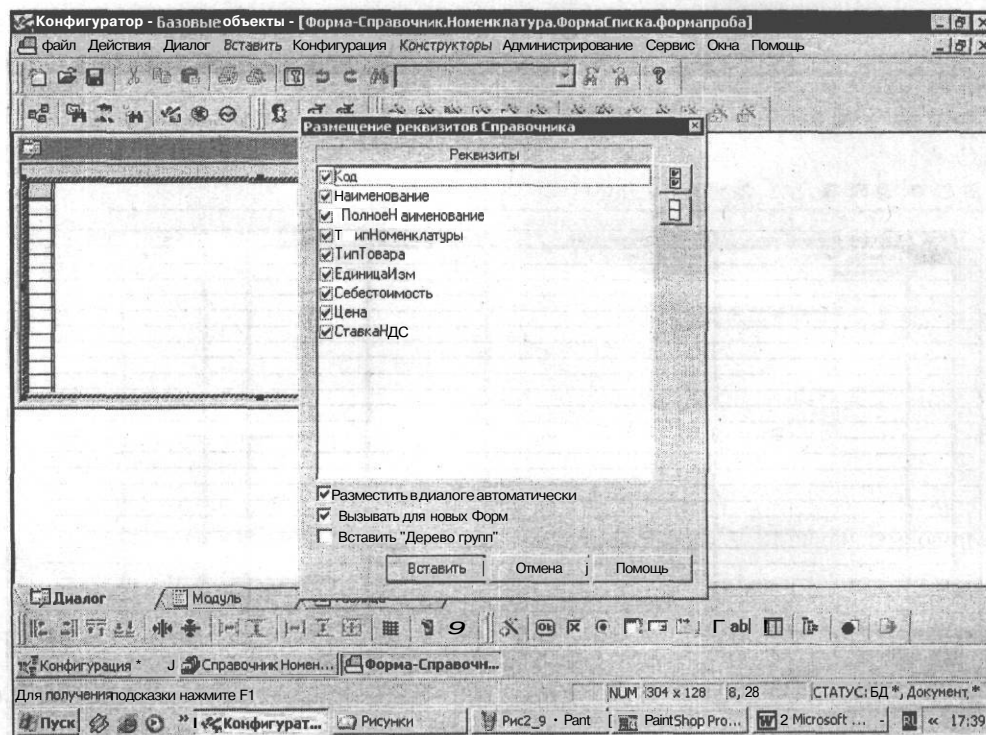


Рис. 2.10. Диалог размещения реквизитов справочника

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

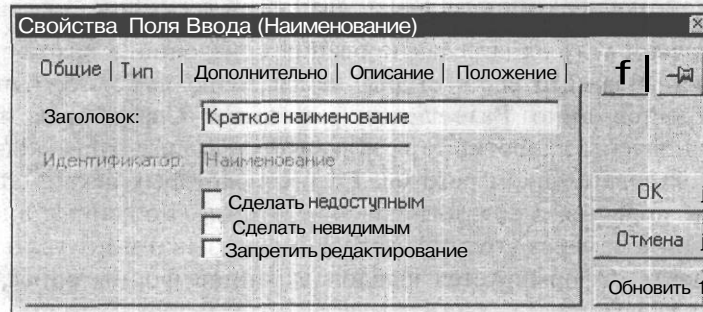


Рис. 2.11. Палитра свойств поля ввода

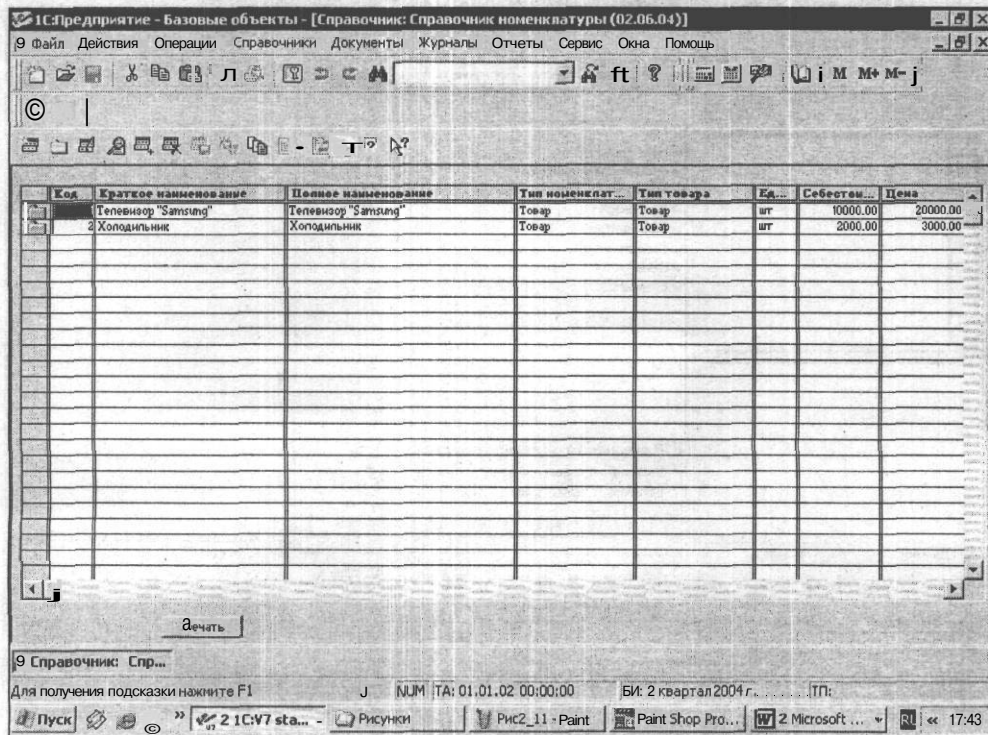


Рис. 2.12. Экранная форма списка справочника Номенклатура

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 Наименование.

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. 2.11). Напишите заголовок: Краткое наименование. Запустите "1С:Предприятие" в пользовательском режиме. Если все сделано правильно, экранная форма списка справочника Номенклатура будет выглядеть так, как на рис. 2.12.

Ставка НДС, как и ставки любых налогов, может периодически изменяться. То же касается и цены реализации товара. Аналогичные проблемы уже обсуждались в *разд. 2.2* — о периодических константах и решаются установлением свойства **Периодический** реквизитам Цена и Ставка НДС в палитре свойств реквизита через вкладку **Дополнительно**.

В примере к данному разделу создан справочник Контрагенты. Его можно создать самостоятельно, чтобы попрактиковаться. Пока в этот справочник нужно внести дополнительные реквизиты Полное Наименование и Адрес, с типом значения Строка и длиной 100 символов. В этом справочнике у нас будут храниться данные о контрагентах.

2.3.1. 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 это не совсем удобно при работе — требуется горизонтально прокручивать окно справочника. В таком случае можно использовать форму элемента справочника. Для ее вызова в окне редактирования справочника необходимо нажать кнопку **Форма элемента** (рис. 2.13).

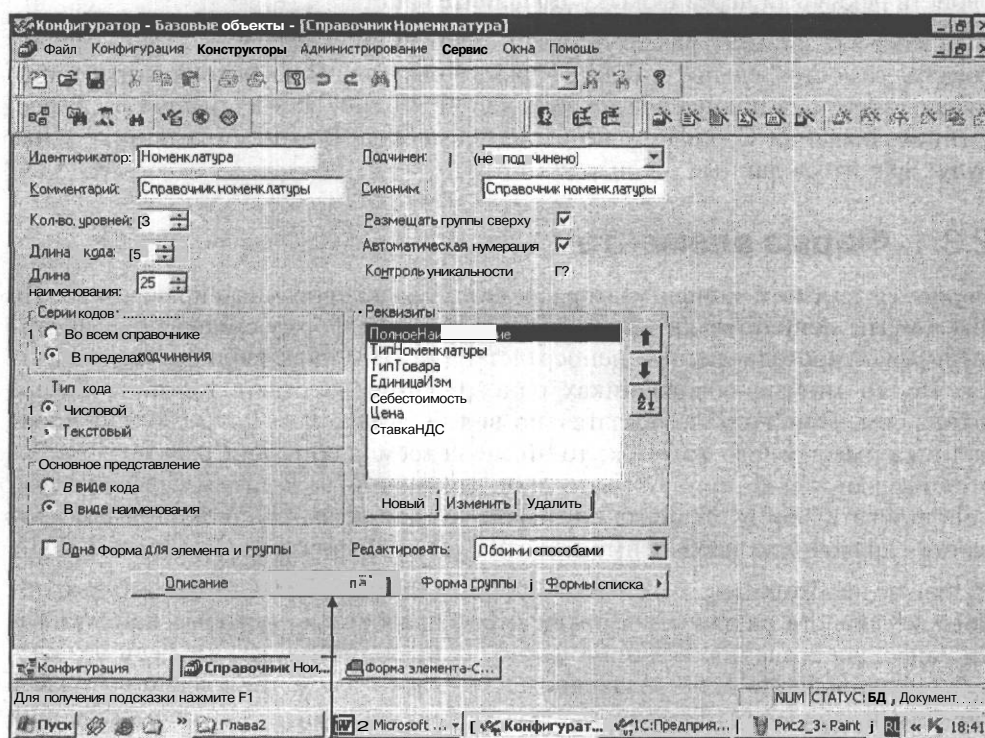
Вставить необходимые реквизиты в форму элемента можно с помощью такого же диалога **размещения** реквизитов справочника, которые использовали ранее для создания экранной формы списка (см. рис. 2.10). Если все сделано правильно, то окно редактирования формы элемента должно выглядеть так, как на рис. 2.14. Только на рисунке присутствует и диалог размещения реквизитов справочника, для наглядности.

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. Откройте экранную форму справочника. Щелкните по меню **Действия | Редактировать в диалоге**. Затем дважды щелкните по какой-нибудь строке (элементу) в экранной форме справочника. Откроется именно форма элемента (рис. 2.15). Если закрыть форму элемента и вновь просмотреть меню **Действия**, то напротив пункта **Редактировать в диалоге** будет установлен флажок. 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.



Кнопка "Форма элемента"

Рис. 2.13. Вызов формы элемента справочника

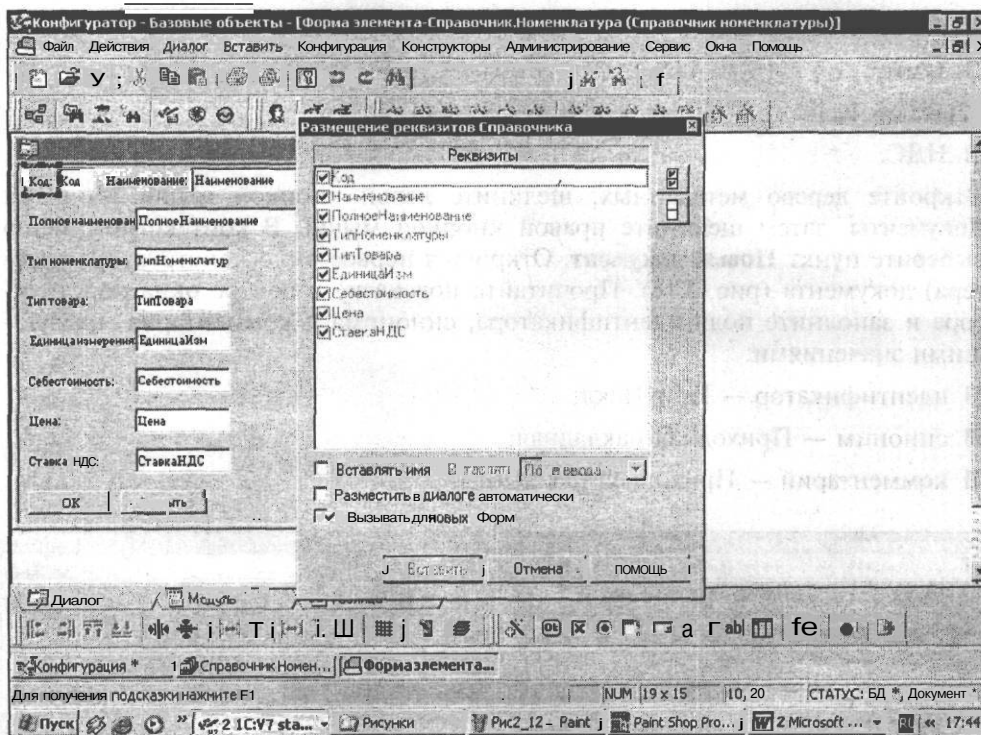


Рис. 2.14. Конструирование формы элемента справочника

2.4. Документы

Теперь начинаем создание документа.

Замечание

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. Однако следует помнить, что использование этого типа данных может выходить далеко за рамки простой фиксации информации о хозяйственных операциях.

Исходя из того, что документ аналогичен стандартному бумажному, для фиксации поступления товаров на фирму будем использовать Приходную накладную.

В документ Приходная накладная необходимо вносить следующие данные:

- наименование товара;
- цену товара;

- количество товара;
- сумму;
- ставку НДС;

а НДС.

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. 2.15). Прочитайте подсказку в центре окна конструктора и заполните поля идентификатора, синонима и комментария следующими значениями:

- идентификатор — ПрихНакл;
- синоним — Приходная накладная;
- комментарий — Приходная накладная.

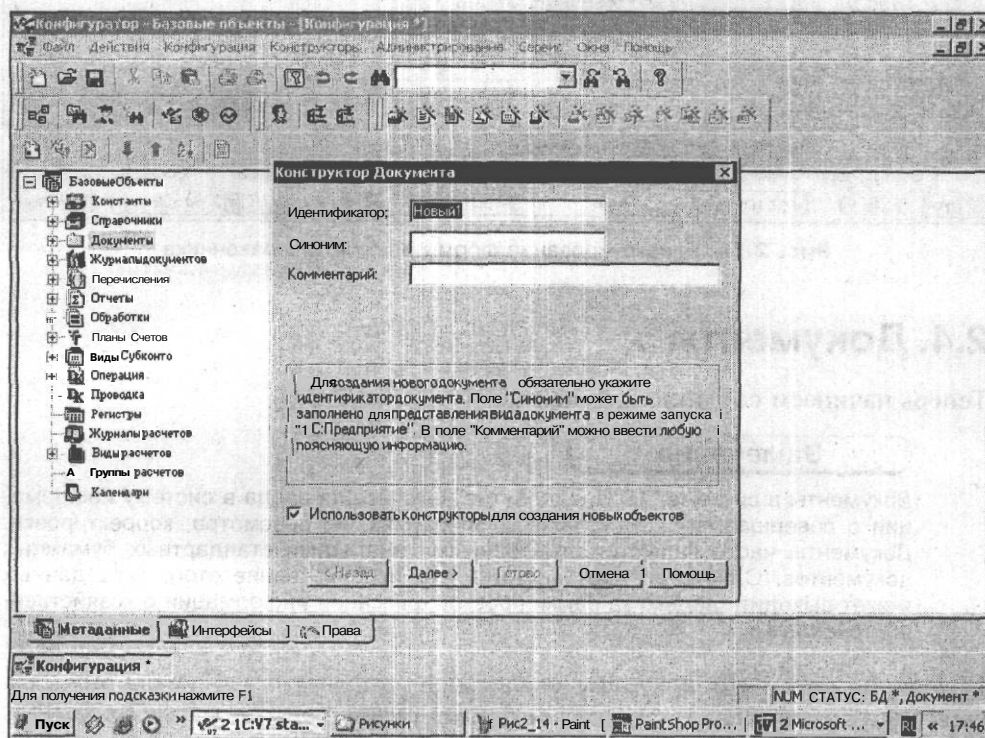


Рис. 2.15. Конструктор документа

Нажмите кнопку **Далее**. Флажки проведения и перепроведения документа установите, если они не выставлены. Нажмите еще раз 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 — **ПрихНакл**. Нажмите кнопку **Далее**, пропустите вставку документа в пользовательский интерфейс (у нас его пока нет).

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

На рис. 2.16 показано окно редактирования документа.

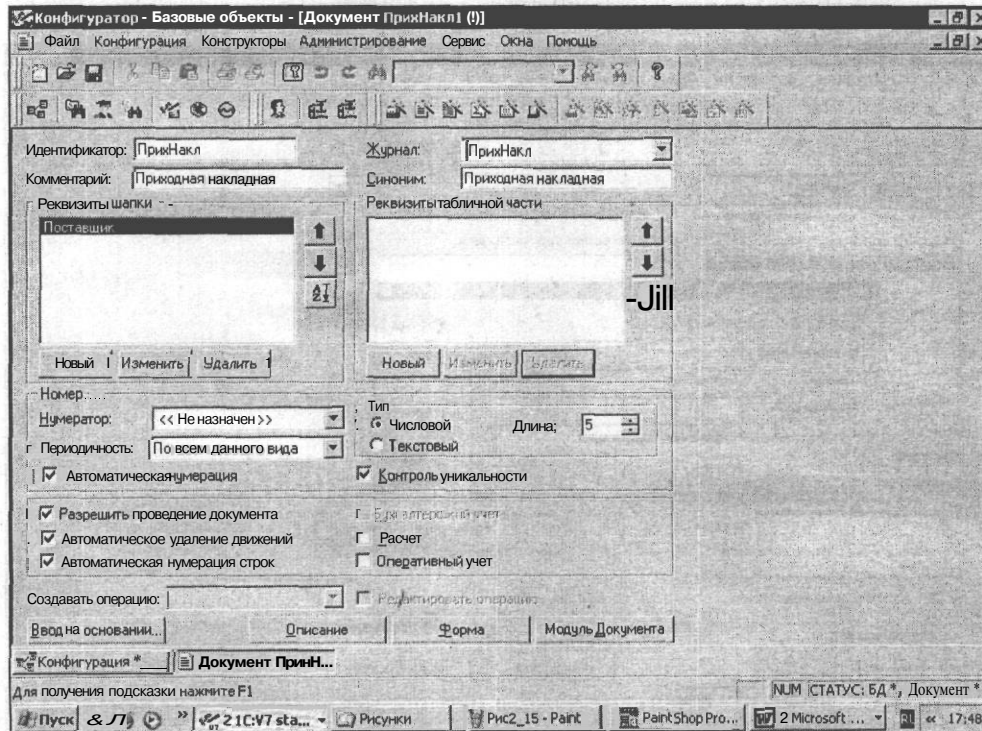


Рис. 2.16. Окно редактирования документа

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, то есть 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.

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

С Замечание

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

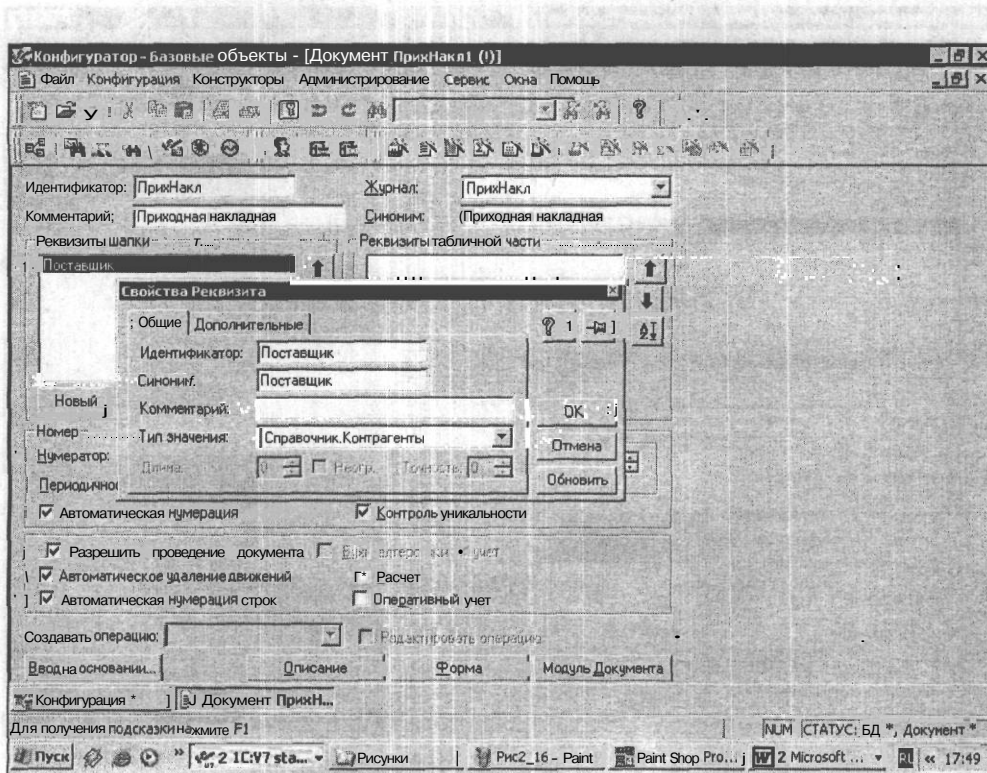


Рис. 2.17. Палитра свойств реквизита шапки документа

Создаем реквизиты шапки. Пока в ней будет только один реквизит — Поставщик. Нажмите кнопку **Новый** под полем **Реквизиты шапки**. Заполните палитру свойств реквизита (рис. 2.17). Обратите внимание на тип значения Справочник.Контрагенты (этот справочник вы должны были создать самостоятельно). 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 контрагент-

та-поставщика. Его достаточно просто выбрать из автоматически открывающегося при обращении к реквизиту справочника. Выбор типа значения Справочник означает, по существу, прямое обращение к данным справочника.

Если какие-либо данные хранятся в справочниках, старайтесь при использовании этих данных создавать реквизит с типом значения Справочник. Ваши Данные. Как правило, этот прием поможет экономить время пользователя и защитит ваши данные от излишних ошибок. Ведь если каждый раз в печатывать 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.

Теперь создадим реквизиты табличной части. В одной приходной накладной может быть несколько товаров, следовательно, товары у нас должны размещаться в табличной части. Нажмите кнопку **Новый** под полем **Реквизиты табличной части**. С помощью палитры свойств создадим реквизит Товары, тип значения — Справочник. Номенклатура. Затем реквизит Цена, тип значения — Число, длина 10, точность — 2. Реквизиты Количество, Сумма, НДС имеют такие же свойства, реквизит СтавкаНДС имеет длину 6 при точности 2. Создайте их самостоятельно. Если будут затруднения, обратитесь к учебному примеру.

Мы пока пропустим описание остальных полей в окне редактирования документа, хотя назначение части из них вы можете понять из названия или по аналогии со справочниками.

Нажмите на кнопку **Форма**, и у вас появится диалоговое окно размещения реквизитов документа, аналогичное такому же в конструкторе справочника (рис. 2.18).

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. 2.19.

Следующим шагом отредактируем экранную форму при помощи 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.

Запустите конфигурацию в пользовательском режиме и откройте новый документ Приходная накладная (**Операция | Журналы документов | Журнал приходных накладных**, в окне журнала выберите **Новый**). На рис. 2.20 показан пользовательский вид документа. Если щелкнуть по кнопке в поле **Поставщик**, откроется форма справочника Контрагенты для выбора поставщика. Если щелкнуть по строке в колонке Товар, можно выбрать товар для автоматической вставки его в табличную часть документа.

Автоматическое заполнение поля **Поставщик** 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.

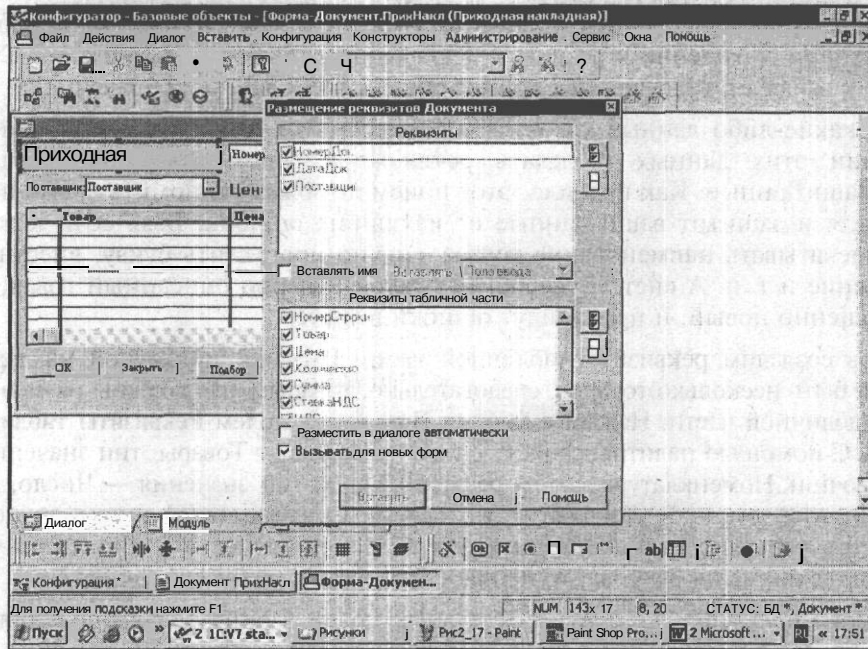


Рис. 2.18. Диалог размещения реквизитов справочника

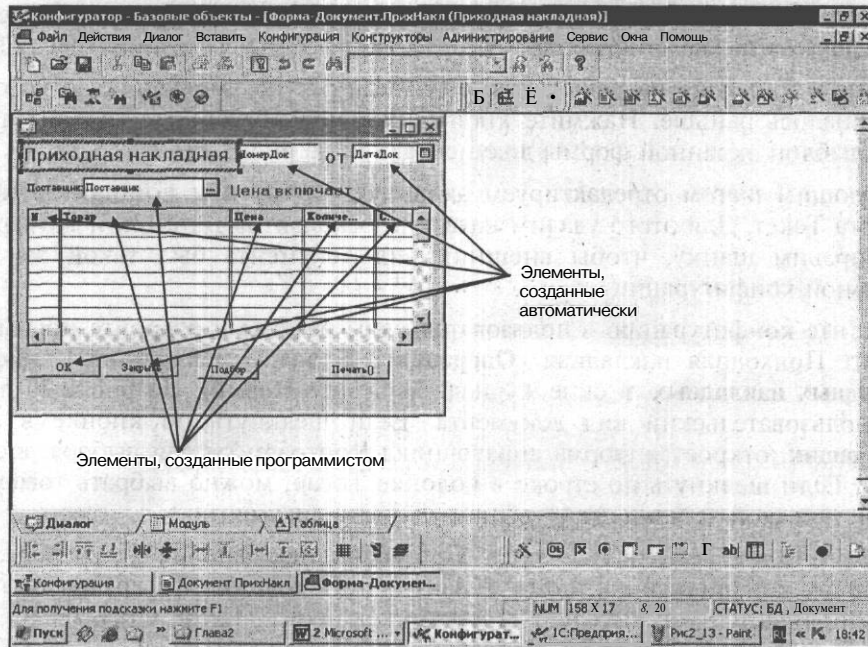


Рис. 2.19. Шаблон экранной формы документа

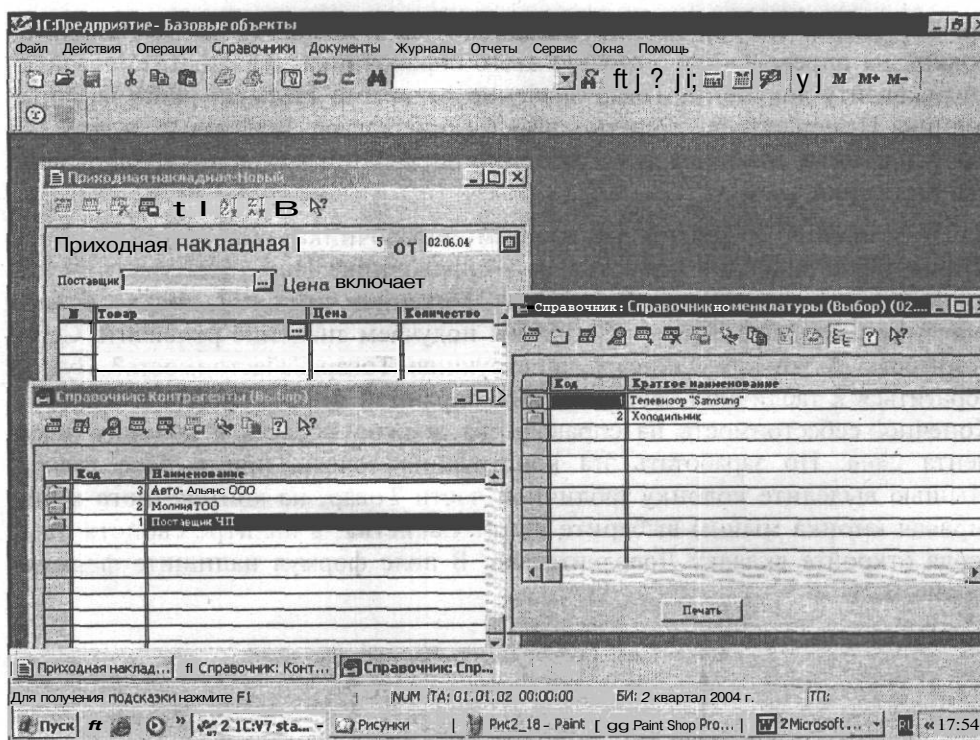


Рис. 2.20. Вид документа с открытыми справочниками для выбора

Существует несколько способов автоматического заполнения реквизитов табличной части документа. Мы рассмотрим два наиболее распространенных. Один — сейчас, второй — после изучения основ встроенного языка.

2.4.1. Построчное заполнение документа

Зайдите в шаблон экранной формы документа (см. рис. 2.19). 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 по документу **ПрихНакл** и в контекстном меню выберите пункт **Редактировать форму**.

Итак, отпускная цена товара и процент ставки НДС у нас уже внесены в справочник Номенклатура. Тип значения реквизита Товар приходной накладной тоже справочник Номенклатура. Когда мы щелкаем по строке документа в колонке **Товар**, мы открываем справочник Номенклатура и выбираем строку с нужным товаром в этом справочнике, то есть позиционируем его. Реквизит Товар с типом значения Справочник.Номенклатура — это

прямая ссылка на справочник номенклатуры с позиционированием (установкой на определенную строчку) этого справочника. Далее мы присваиваем реквизиту документа Товар значение реквизита Наименование из справочника Номенклатура, то есть наименование товара. Вспомните, в *разд. 2.3* мы установили основное представление справочника номенклатуры в виде наименования.

Для обращения к другим реквизитам справочника используется точка. Чтобы найти значение цены товара в справочнике Номенклатура, нам надо написать конструкцию Товар.Цена. Механизм прежний, только вместо значения реквизита Наименование мы получаем значение реквизита Себестоимость. А вот где написать конструкцию Товар.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. Мышь выделите колонку табличной части Товар, из контекстного меню (правая кнопка мыши) выберите пункт **Свойства**, в палитре свойств поля ввода откройте вкладку **Дополнительно**. В поле формул напишите формулу из рис. 2.21.

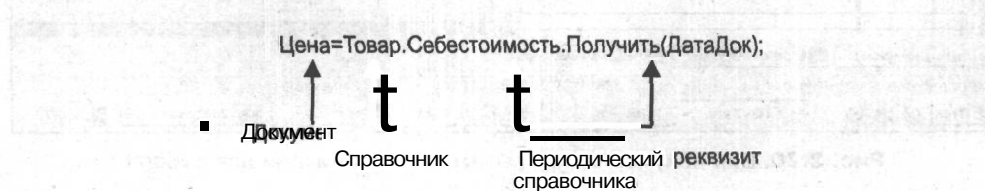


Рис. 2.21. Состав реквизитов в формуле

Знак равенства соответствует в данном случае обычному пониманию этого знака, но называется оператором присваивания.

Рис. 2.21 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. 2.2* структуре данных периодических констант. Для того чтобы получить значение периодического реквизита на определенную дату, нужно применить метод получить(дата). 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=Товар.СтавкаНДС.Получить(ДатаДок);

Количество товара мы можем внести только интерактивно. Но в переменной 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. 2.22), несколько строк.

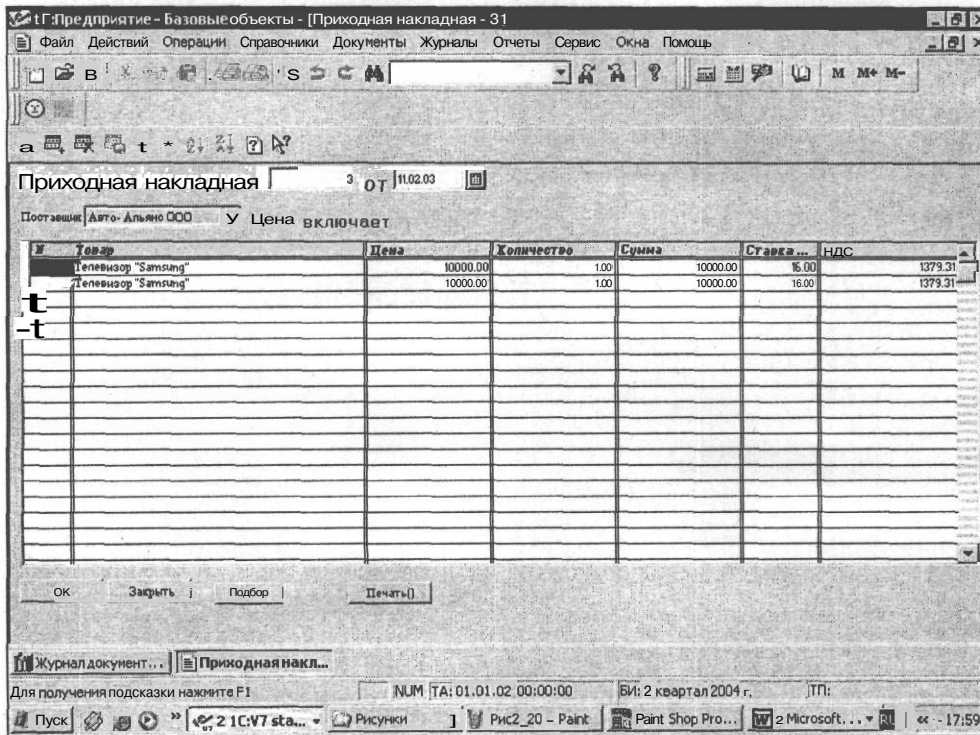


Рис. 2.22. Документ, заполненный автоматически

Этот способ достаточно простой, но не всегда эффективный. В реальных конфигурациях вычисление реквизитов часто невозможно с помощью простейших формул, предварительно нужно провести определенную программную обработку данных. Поэтому начинаем учиться программировать.

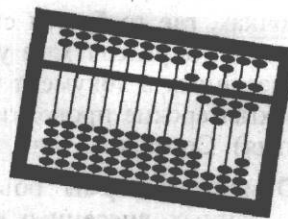
1. The first part of the document is a title page containing the name of the organization and the title of the report. It also includes the date of publication and the author's name.

Year	Q1	Q2	Q3	Q4
2018	100	100	100	100
2019	100	100	100	100
2020	100	100	100	100
2021	100	100	100	100
2022	100	100	100	100
2023	100	100	100	100
2024	100	100	100	100
2025	100	100	100	100
2026	100	100	100	100
2027	100	100	100	100
2028	100	100	100	100
2029	100	100	100	100
2030	100	100	100	100

The second part of the document is a table showing the results of the survey. The table is organized into columns representing different categories and rows representing different years. The data shows a steady increase in the number of respondents over time.

The third part of the document is a conclusion. It summarizes the findings of the survey and provides recommendations for future research. The conclusion states that the survey was successful in gathering valuable information about the topic.

Глава 3



Начала программирования в системе "1С:Предприятие"

Мы создали несколько объектов метаданных. В этой главе мы познакомимся с приемами программирования и взаимодействием программ и объектов метаданных в конфигурации.

3.1. 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 по-английски, 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 конфигулятора. В отличие от стандартных объектно-ориентированных языков, метаданные задаются в конфигураторе, и их виды жестко закреплены.

3.2. Программный модуль и простейшие программы

В программном модуле содержатся специфические алгоритмы функционирования, оформленные в виде процедур и функций, написанных на встроенном языке. 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:

- модули формы объекта метаданных, предназначенные для обработки данных, внесенных в экранную форму;
- модули объекта метаданных в виде документа;
- 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:

1. *Раздел переменных*; в языке "1С:Предприятие" 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.
2. *Раздел процедур и функций*; отдельные действия, выполняемые в модуле, оформляются как процедуры или функции и имеют имя. Отличия процедур от функций: 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.
3. *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:

- 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 называ-

ется *основной программой*. Ее назначение — выполнить инициализацию модуля.

- Если в процессе компиляции найдены ошибки, то модуль считается не-работоспособным, и, скорее всего, ваша конфигурация нормально работать не будет. Система выдаст сообщение об ошибке.

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

Откройте конфигурацию, щелкните правой кнопкой мыши по наименованию конфигурации **Задача**, в контекстном меню выберите пункт **Глобальный модуль** (рис. 3.1). Откроется пустое окно — это и есть место для написания программы в глобальном модуле.

В окне введем следующий текст:

Листинг 3.1

```
//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.

Разберем приведенную программу.

Двойная наклонная черта // означает комментарии к программе. Комментарии — это заметки программиста не для исполнения системе, а для себя, чтобы было удобнее пользоваться программой. Слова, написанные после знака //, система не интерпретирует как программу. Наличие комментариев очень важно, и ими нельзя пренебрегать, иначе вы запутаетесь в собственной (тем более, в чужой) программе. В данном случае мы комментариями сопроводили классическую структуру программного модуля.

В разделе переменных при помощи ключевого слова **Перем** мы объявили переменную с произвольным идентификатором (**привет**). В разделе процедур

и функций мы создали процедуру с произвольным именем приветствие (). Парные 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.

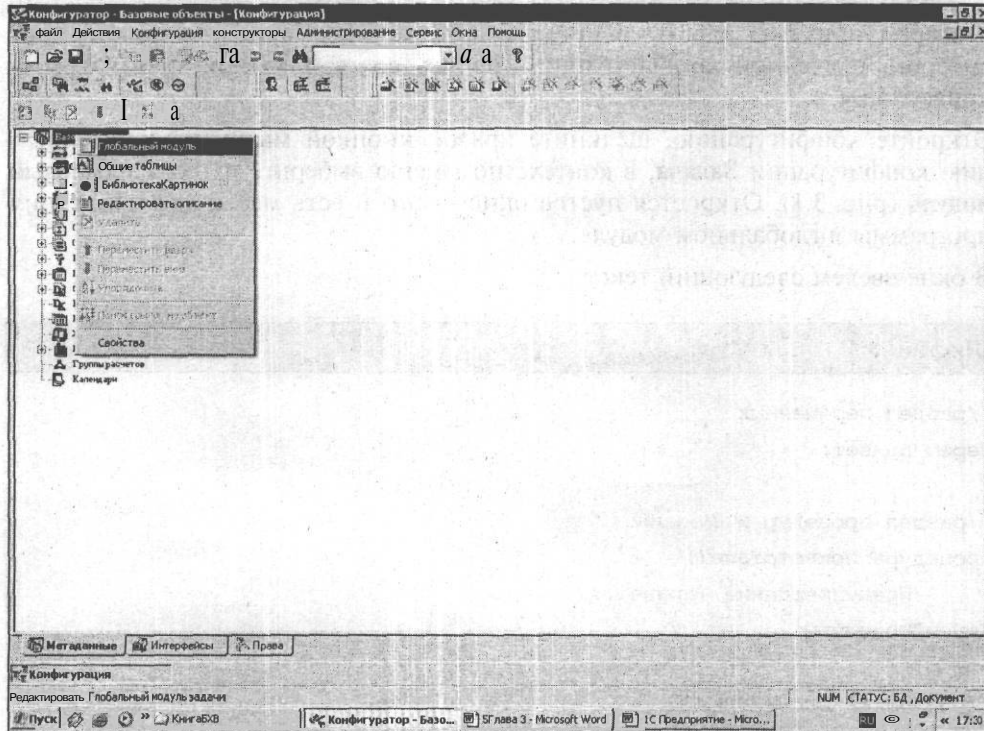


Рис. 3.1. Переход к глобальному модулю в конфигурации

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).

В меню Действия | Синтаксический контроль осуществляется проверка синтаксиса программы. Если вы написали все верно, в окне сообщений будет

выведена строка: "Синтаксических ошибок не обнаружено!". Однако осуществляется проверка только синтаксиса, а правильность написания программы можете проконтролировать только вы сами. В панели инструментов **Текстовый редактор** найдите кнопку с таким же рисунком, что у команды меню **Синтаксический контроль**. В дальнейшем при проверке синтаксиса можно использовать эту кнопку.

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 см. на рис. 3.2.

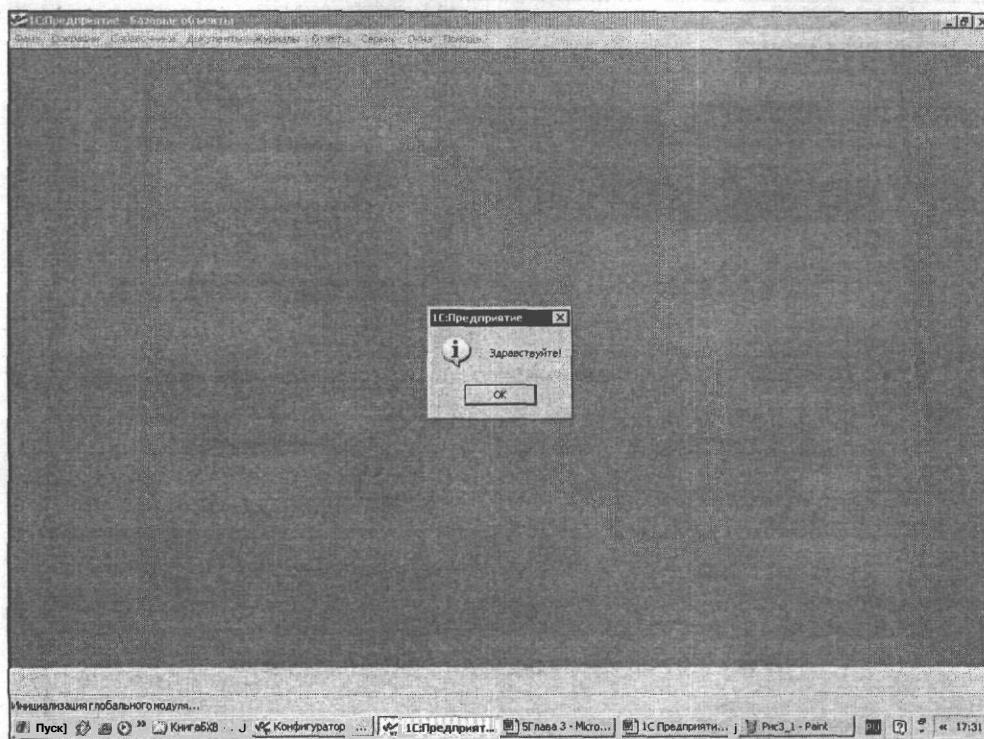


Рис. 3.2. Приветствие системы

Текст программного модуля разноцветный. Это очень удобно. Синтаксическая ошибка в написании оператора изменяет его цвет. И структура программы становится более наглядной.

Примечание

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

Дополнительно исследовать нашу программу можно с помощью синтакс-помощника. В процедуре приветствие () щелкните мышью по системной процедуре предупреждение (), затем правой кнопкой вызовите контекстное меню и выберите пункт **Поиск в синтакс-помощнике**. В правом нижнем окне (рис. 3.3) помещено подробное описание системной процедуры Предупреждение ().

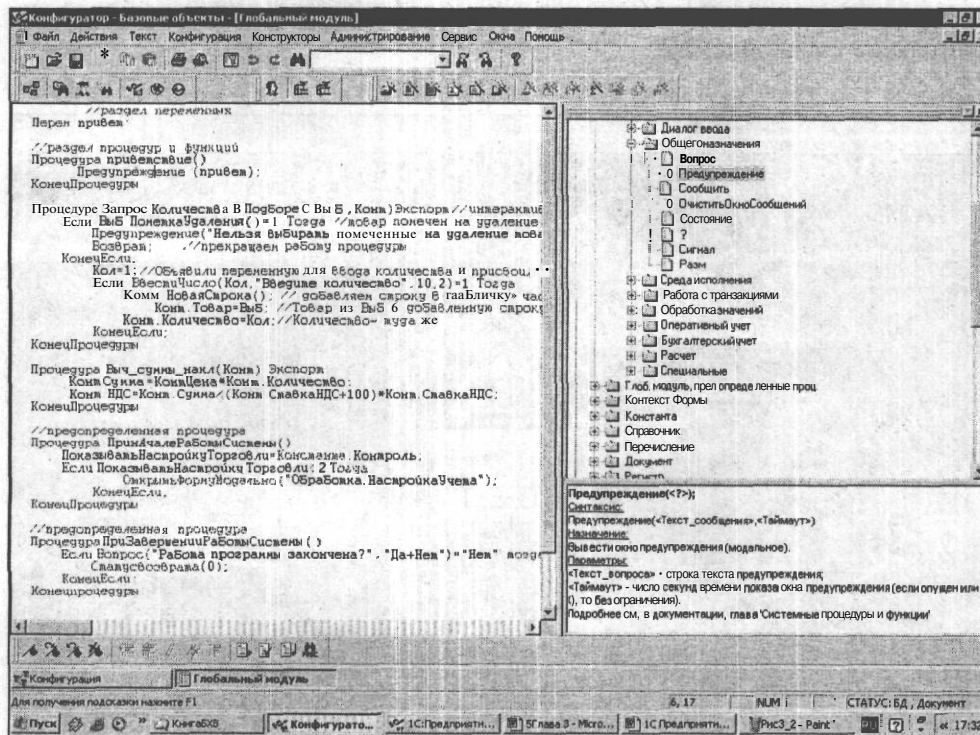


Рис. 3.3. Описание процедуры предупреждения

Попробуйте самостоятельно поработать с параметром тайм-аут.

С помощью синтакс-помощника можно разобраться в работе процедур и функций. Однако этот инструмент не заменяет руководства по описанию 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 синтаксис-помощник можно попасть также через меню **Сервис | Синтаксис-помощник**.

Помимо процедур, которые мы создаем самостоятельно, в языке "1С:Предприятие" существуют *предопределенные процедуры*. Эти процедуры имеют свои, зарезервированные системой названия. Они 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 вы можете посмотреть в синтаксис-помощнике для глобального модуля или при открытии ветви, относящейся к определенному виду метаданных (рис. 3.4).

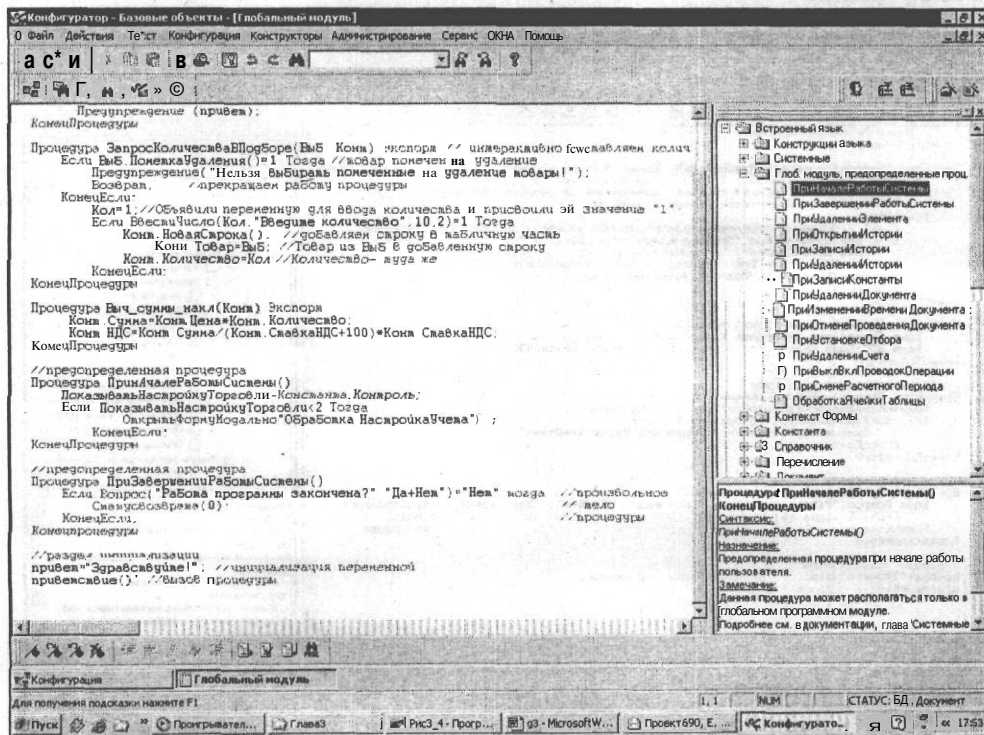


Рис. 3.4. Описание предопределенных процедур модуля формы документа

Напишем полезную процедуру для завершения работы "1С:Предприятие". 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 по ошибке закрыть "1С:Предприятие". Наша процедура подстрахует от подобной случайности.

Так как алгоритм должен выполняться при окончании работы всей системы "1С:Предприятие", посмотрим predeterminedные процедуры глобального модуля. Предeterminedная процедура ПриЗавершенииРаботыСистемы() нам подходит. Напишем текст процедуры, разместив ее в разделе процедур и функций, после процедуры приветствие о (рис. 3.5):

Листинг 3.2

Процедура ПриЗавершенииРаботыСистемы()

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

Статусвозврата(0);

КонецЕсли;

Конецпроцедуры

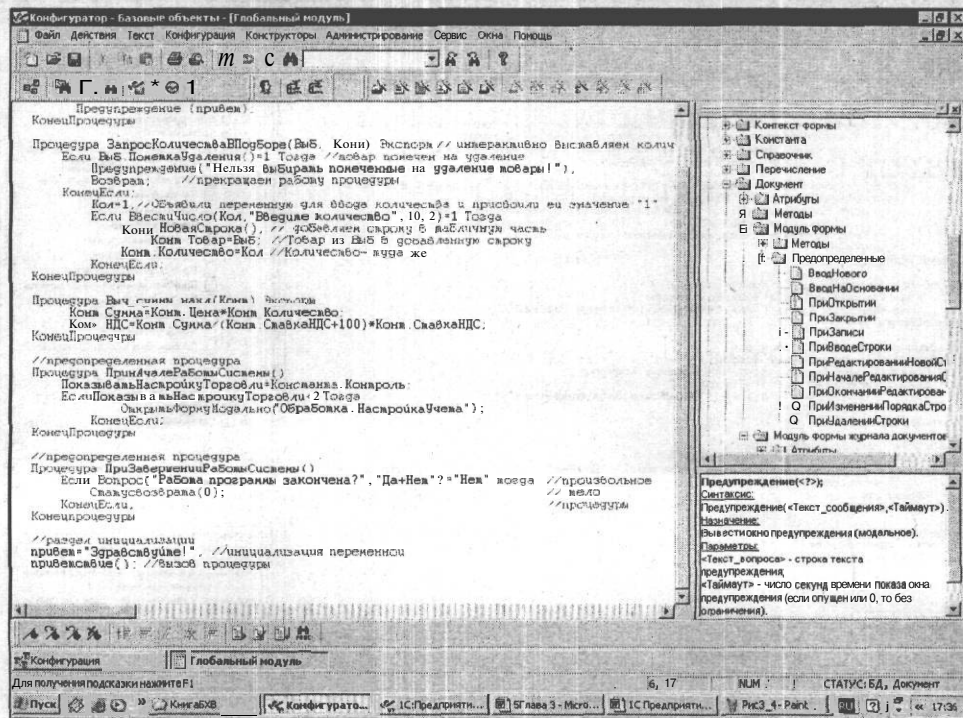


Рис. 3.5. Произвольные и predeterminedные процедуры

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

```
Если <Условие> Тогда <Операторы> КонечЕсли;
```

проверяет выполнение определенного условия. Если условие выполняется, тогда происходит какое-либо действие, заключенное между операторами Тогда И КонечЕсли.

Подробное описание системной функции вопрос () посмотрите в синтаксис-помощнике. В нашем случае эта функция выводит на экран окно с вопросом, кнопками (Да и Нет) и возвращает ответ пользователя. Если вы нажмете кнопку Да, условие ЕСЛИ... Тогда... КонечЕсли; 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 СтатусВозврата() смотрите в документации или в синтаксис-помощнике.

3.3. Заполнение документа методом подбора

В предыдущем разделе 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 при помощи следующих системных процедур:

Листинг 3.3

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

а также предопределенной процедуры обработкаподбора().

В режиме конфигурирования откройте модуль формы документа и создайте новую процедуру (листинг 3.4).

Листинг 3.4

```
Процедура Подбор ()
    ОткрытьПодбор ("Номенклатура", "ФормаСписка");
КонецПроцедуры
```

Метод `ОткрытьПодбор` 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 открывает выбранную экранную форму справочника.

Создадим следующую предопределенную процедуру:

Листинг 3.5

```
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. Согласно правилу, описанному в *разд. 3.2*, поскольку новые процедуры будут инициализироваться в `ОбработкаПодбора`, располагаться в модуле они должны ранее этой процедуры. Вставим товар из переменной `Выб` и его количество в табличную часть документа, в колонки `Товар` и `Количество` (листинг 3.6).

Листинг 3.6

```
Процедура ЗапросКоличестваВПодборе (Выб) //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
    НоваяСтрока (); //добавляем строку в табличную часть
    Товар=Выб; //Товар из Выб в добавленную строку
    Количество=Кол; //Количество - туда же
КонецЕсли;
КонецПроцедуры
```

Сразу же вернемся к процедуре обработкаПодбора () (см. листинг 3.5) и запишем в ней вызов новой процедуры:

Листинг 3.7

```
Процедура ОбработкаПодбора (Выб)
    ЗапросКоличестваВПодборе (Выб);
КонецПроцедуры
```

Кое-какие конструкции вам уже знакомы, их назначения понятны и из комментариев к процедуре. Обратите внимание, что русский синтаксис операторов языка несет смысловую нагрузку, и 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 на удаление. 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. Эта функция организует диалог ввода для ввода с клавиатуры какого-либо числового значения (рис. 3.6), в нашем случае — количества товара.

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, макси-

мальную длину **ВВОДИМОГО** числа и его точность. Согласно синтаксису этой функции, мы записали следующее: в произвольную переменную кол помещается число с максимальной длиной целой части семь знаков, длиной дробной части — два знака. Один знак зарезервирован для разделителя целой и дробной части числа. Дали заголовок окну диалога (см. рис. 3.6).

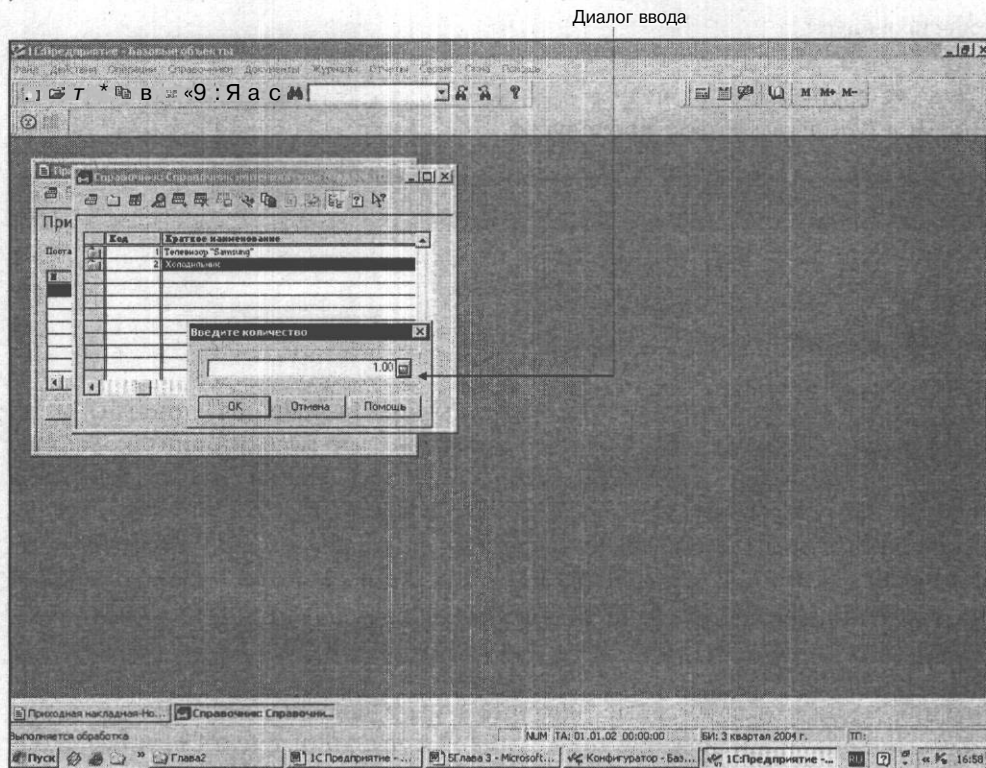


Рис. 3.6. Диалог ввода числа

Перед диалогом ввода мы программно задали переменной 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 документа новаяСтрока () добавляет новую строку в табличную часть документа. В эту строку затем вставляется наименование товара и его количество, что понятно из комментариев к процедуре.

Товар и количество уже введены в документ. Теперь из справочника Номенклатура внесем цену товара и ставку НДС товара (листинг 3.8).

Листинг 3.8

```

Процедура УстанЦеныПрих ()
    Цена=Товар.Себестоимость.Получить (ДатаДок); //Себестоимость товара из
спр. Номенклатура – в документ
    СтавкаНДС=Товар.СтавкаНДС.Получить (ДатаДок); //Ставку НДС из спр. Но-
менклатура – в документ
КонецПроцедуры

```

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

Последняя процедура — организация заполнения вычисляемых реквизитов документа. Это Сумма и НДС (листинг 3.9).

Листинг 3.9

```

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

```

Это стандартные арифметические выражения. Эту процедуру мы также вызываем из ОбработкаПодбора () (см. листинг 3.5), и ее окончательный вид представлен в листинге 3.10.

Листинг 3.10

```

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 при помощи инструмента **Кнопка** панели инструментов **Элементы диалога** или посредством 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. 3.7). 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. Обратите внимание, что в поле формул вкладки **Дополнительно** палитры свойств объекта можно писать не только формулы, но и ссылки на процедуры, которые будут выполняться при активизации этого объекта.

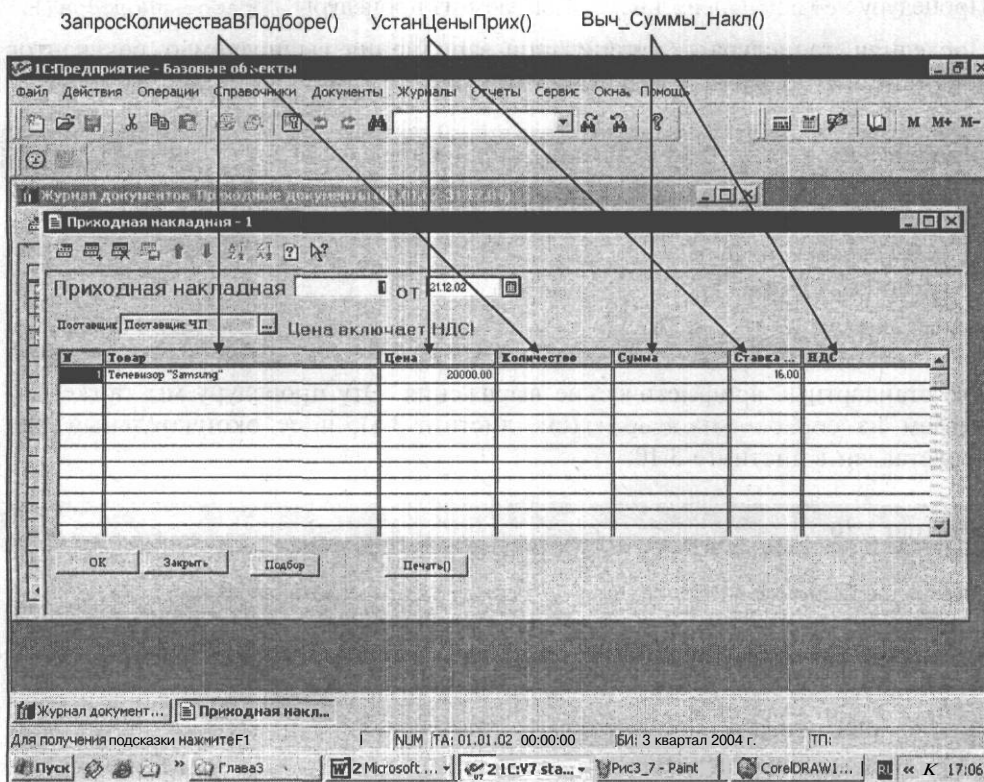


Рис. 3.7. Документ Приходная накладная

Следующий документ нашей учебной конфигурации — Расходная накладная. По составу реквизитов Расходная накладная совпадает с Приходной

накладной. Поэтому мы можем скопировать документ приходной накладной и вставить его в ветвь дерева метаданных **Документы**. Для выполнения этих операций откройте окно метаданных в конфигураторе. Выделите нужный объект метаданных — документ **ПрихНакл**. Скопируйте его в буфер обмена с помощью меню **Действия | Запомнить** либо посредством соответствующих кнопок на панели инструментов. Затем вставьте документ с помощью меню **Действия | Вставить** либо кнопок на панели инструментов. Новый документ будет вставлен под именем **ПрихНакл1**. Откройте окно редактирования этого документа и задайте новый идентификатор: **РасхНакл**, синоним и комментарий документа — **Расходная накладная**. Журнал для хранения расходных накладных в этот раз нужно создавать самостоятельно.

Закройте окно редактирования документа, перейдите в окно дерева метаданных и щелкните по ветви журналов документов правой кнопкой мыши. В контекстном меню выберите пункт **Конструктор журнала документов**. Заполните идентификатор: **РасхНакл**; синоним и комментарий: **Журнал расходных накладных**. Нажмите кнопку **Далее** и выберите вид журнала **Обычный**. Нажмите кнопку **Далее**. Пользовательского меню у нас пока нет, поэтому пропустите это окно и нажмите кнопку **Готово**. Раскроется окно редактирования журнала документов (рис. 3.8).

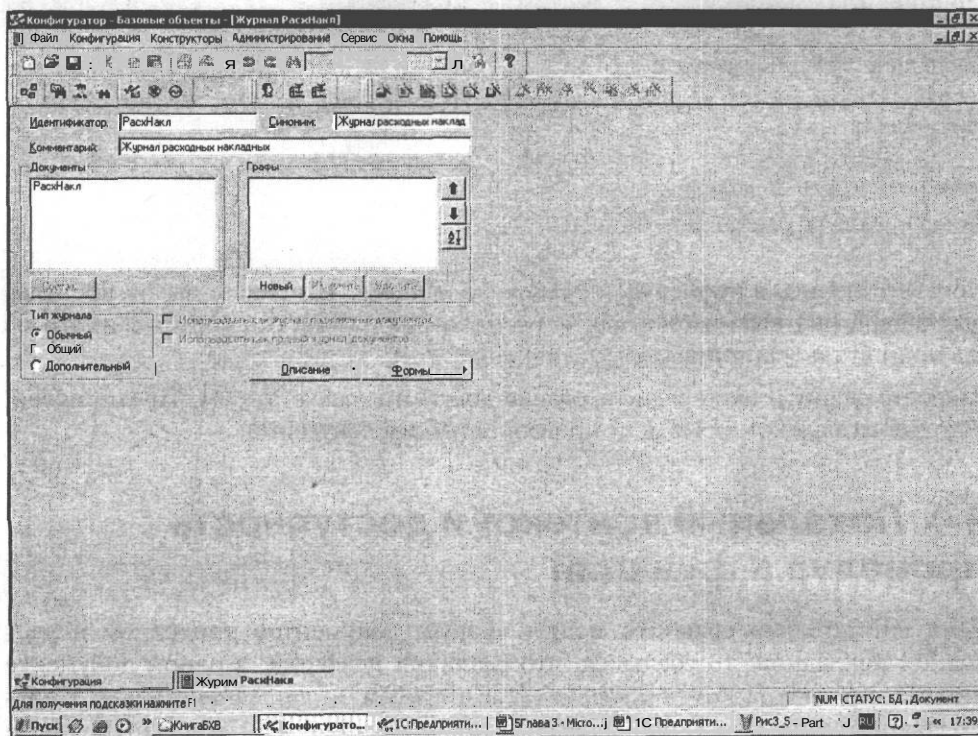


Рис. 3.8. Окно редактирования журнала документов

Больше ничего в журнале документов нам пока делать не надо. Закройте это окно. Откройте окно редактирования документа и в поле **Журнал** выберите журнал **РасхНакл**.

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

Благодаря вводу в работу нового документа в учете хозяйственной деятельности нашего предприятия произошли изменения. В частности, при оформлении отпуска товара документом **Расходная накладная** цену товара необходимо указывать отпускную. В нашем справочнике **Номенклатура отпускная** цена хранится в реквизите **Цена**. Поэтому в модуле формы документа **Расходная накладная** нужно сделать изменения. Процедуру **Устанценыприх()** переименуем в **УстанценыРасх()**, вместо реквизита справочника номенклатуры **Себестоимость** используем реквизит **Цена**. Процедура **УстанценыРасх()** будет выглядеть так, как в листинге 3.11.

Листинг 3.11

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
```

КонецПроцедуры

Соответственно, в процедуре **Обработкаподбора** о 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 работу документа.

3.4. Локальный контекст и доступность процедур и функций

Если внимательно сравнить модули формы документов приходной и расходной накладной, то убедимся, что различия невелики: в одном документе имеется процедура **УстанЦеныПрих()**, а в другом — **УстанценыРасхо**. Соответственно отличается процедура **Обработкаподбора()**. 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 37 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 может быть необходимым оперировать с сотнями тысяч записей и десятками тысяч документов. И если оператор выписывает весь 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 ушло впустую, а в пределах года это солидный убыток для предприятия.

Самая простая возможность в "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 общие для созданных 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 3.12).

Листинг 3.12

```
Процедура ЗапросКоличестваВПодборе(Выб) Экспорт //интерактивно выставляем
число
.....
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 Приходная накладная. 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 3.13

```
Процедура ОбработкаПодбора (Выб) //Элемент спр. Номенклатура – в перемен-
ную 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. Введем глобальный параметр, который при вызове процедуры будет иметь локальный контекст — конт (рис. 3.9). Формальному параметру Конт глобальной процедуры присваивается локальный контекст, т. е. указывается, из какого программного модуля 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 3.14.

Листинг 3.14

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

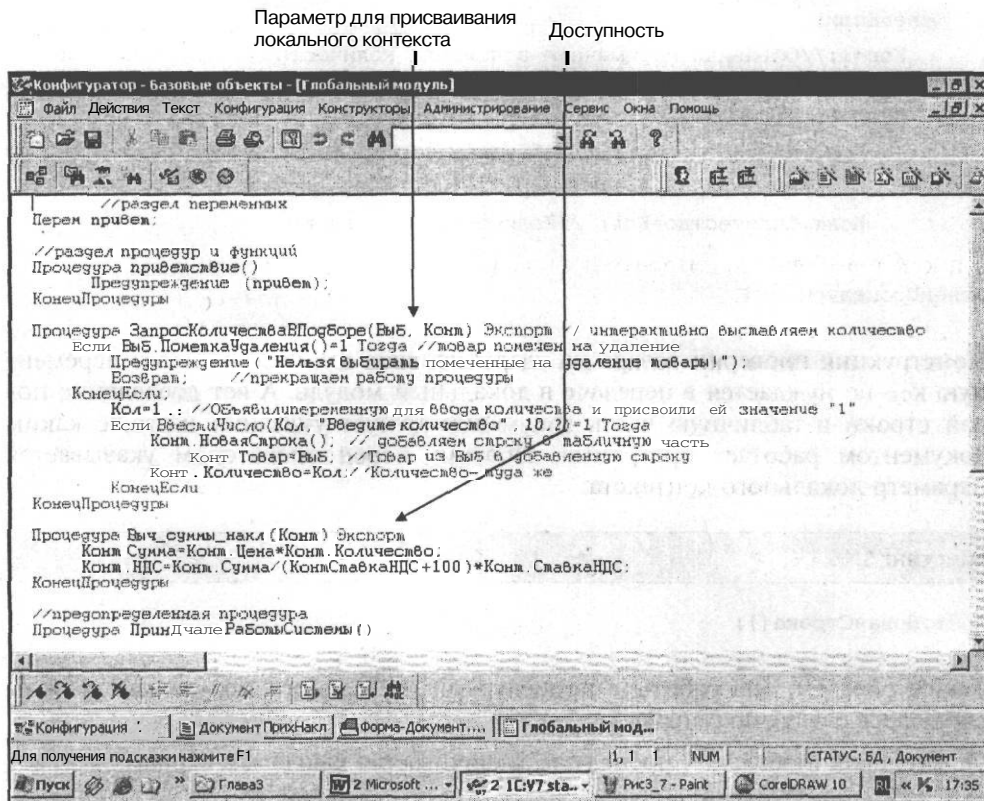



Рис. 3.9. Передача локального контекста программного модуля в качестве параметра

Эта глобальная процедура оперирует только локальными значениями, которые передаются либо из документа Приходная накладная, либо из документа Расходная накладная, либо из какого-нибудь 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 3.15

```

Процедура ЗапросКоличестваВПодборе (Выб, Конт) Экспорт //интерактивно вы-
ставляем 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.Товар=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 нуждается 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 3.16

```
Конт.НоваяСтрока ();
```

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

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

3.5. Итоговые значения колонок документа

Для повышения информативности документа было бы удобно иметь возможность получения итогов по колонкам табличной части документа. В документах Приходная накладная и Расходная накладная 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. 3.10). Перейдите на вкладку **Дополнительные** и устано-

вите флажок **Итог по колонке**. Повторите то же самое для реквизита НДС. Установите итог по колонкам и в документе Расходная накладная.

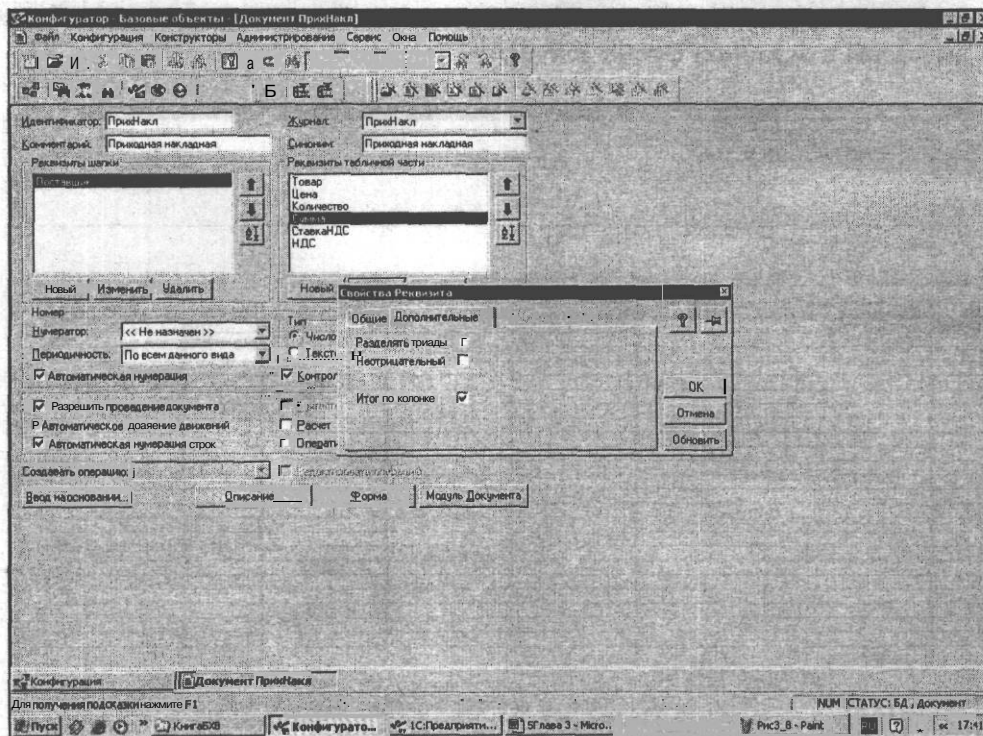


Рис. 3.10. Установка флажка **Итог по колонке**

В экранной форме документов под табличной частью добавьте элемент диалогов **Текст** (с помощью панели инструментов или меню). В палитре свойств текста очистите заголовок. Идентификатор нам тоже не нужен. Щелкните на вкладке **Дополнительные** и в поле формул напишите:

Листинг 3.17

Итог ("Сумма")

Создайте рядом еще один элемент диалогов **Текст** и запишите в нем формулу:

Листинг 3.18

Итог ("НДС")

Расположите оба элемента в форме так, чтобы они были удобны для восприятия. Запустите "1С:Предприятие" в пользовательском режиме, создайте документы Приходная накладная и Расходная накладная с произвольными значениями. Итоговые значения колонок будут автоматически посчитаны и выведены в экранной форме документа (рис. 3.11).

Расходная накладная № [2 от 23.05.04]

Поставщик: Поставщик ЧП Цена включает НДС

N	Товар	Цена	Количество	Сумма	Ставка НДС	НДС
1	Телевизор "Samsung"	20000.00	1.00	20000.00	16.00	2758.62
2	Холодильник	9000.00	1.00	9000.00	16.00	1137.79
		Итого		23000		
		Итого НДС				3172.41

Итого 23000
Итого НДС 3172.41

Журнал документов Жур... Расходная нак накладная

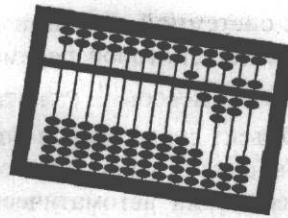
Для получения подсказки нажмите F1 NJM ПА: 01.01.02 00:00:00 БИ: 3 квартал 2004 г.Г ПП: ...

Пуск Глаз Microsoft Word Microsoft ... Конфигуратор - Баз... 1С:Предприятие... 12:29

Итоги по колонкам "Сумма" и "НДС"

Рис. 3.11. Итоги по колонкам Сумма и НДС в экранной форме документа

Глава 4



Вывод на печать

Информация в системе "1С:Предприятие" может быть представлена не только в виде экранных, но и табличных форм, которые можно выводить на принтер. Ведь часто так удобнее работать, да и учетные документы требуется предоставлять заказчикам в распечатанном на бумаге виде. В "1С:Предприятие" 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 позже.

4.1. 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, может включать 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. Поэтому задача печати данных фактически распадается на две:

□ организацию выборки и обработку записей;

О 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. 4.1), 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 (...);
```

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

```
Если Об.ВыбратьЧто-то (...) =1 Тогда
```

```
Пока Об. Об.ПолучитьЧто-то (...) =1 Цикл
```

```
//объект позиционирован
```

```
КонецЦикла;
```

Код	Имя организации	Полное наименование	Адрес	Примечание
3	Альра-Альра ООО	ООО Альра-Альра торговля автомобилями		
2	Минский ТСО	Москва		
1	Поставщик ЧП	Главный поставщик учебной конфигурации		

Рис. 4.1. Выделенная строка (позиционирование) справочника

4.2. 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:

- для построения отчета;
- для организации ввода информации.

В этой книге мы рассмотрим с вами первый режим.

При работе с таблицей для построения отчета сначала нужно создать средствами табличного редактора шаблон печатной формы объекта метаданных. Большинство 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. Для тех случаев, когда локальный контекст модуля, в котором размещена процедура формирования отчета, не может

содержать шаблон печатной формы, то такой шаблон размещают в *глобальном контексте*, т. е. в общих таблицах. Для того чтобы открыть окно, содержащее общие таблицы, необходимо в окне конфигурации нажать правую кнопку мыши на объекте **Конфигурация** и выбрать в контекстном меню пункт **Общие таблицы** (рис. 4.2). В результате будет открыто окно табличного редактора.

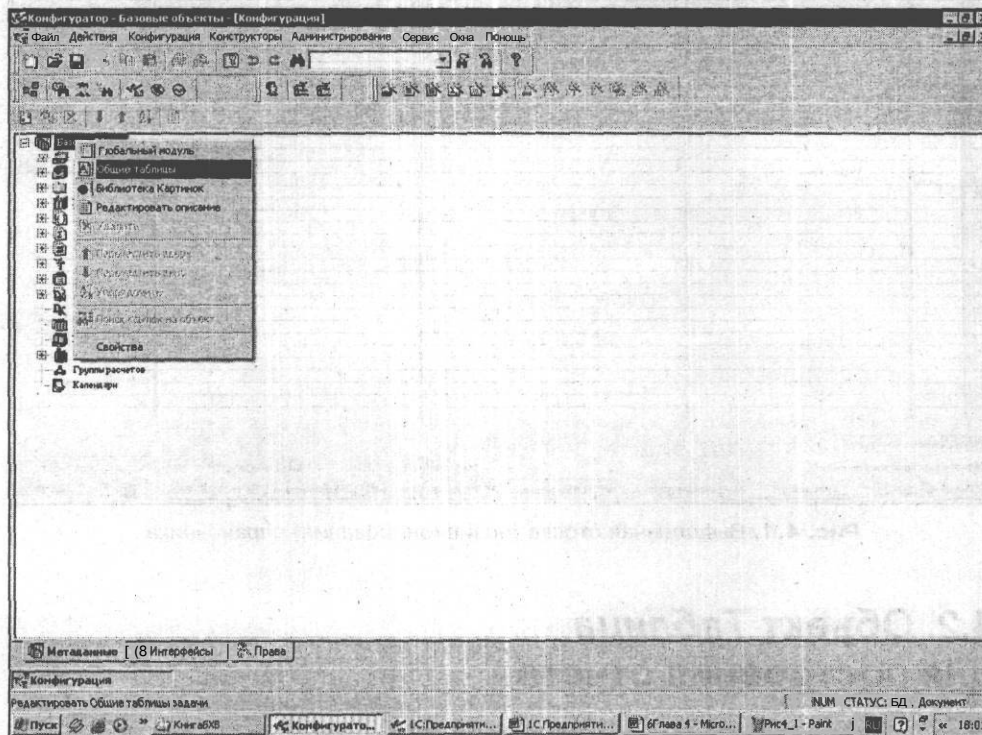


Рис. 4.2. Открытие окна общих таблиц

Посмотрим шаблон печатной формы, принадлежащий локальному контексту, например, справочнику 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. 4.3). Те же строки, колонки, строка формул и окно именованной ячейки.

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.

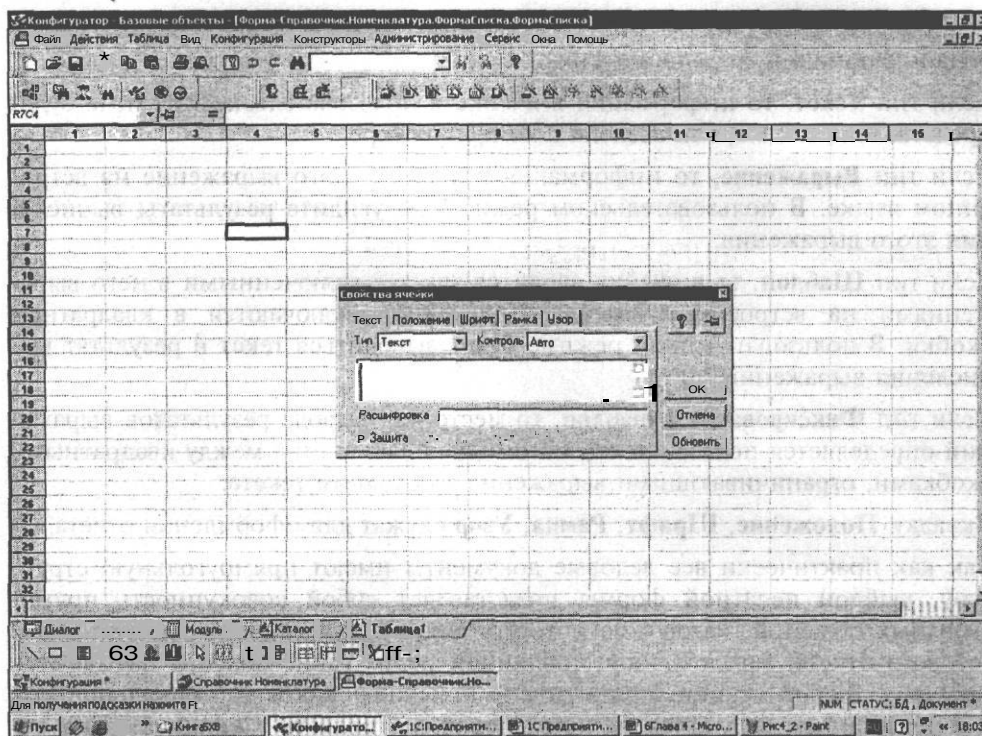


Рис. 4.3. Шаблон печатной формы справочника

Замечание

В практической работе можно копировать таблицу с данными в буфер обмена и затем вставлять ее в MS Excel. Есть и специальные программы — конверторы, передающие таблицы "1С:Предприятие" в MS Excel. Их преимущество, по сравнению с простым копированием, — грамотное форматирование передаваемых данных. Существуют и обратные конверторы — из MS Excel в "1С:Предприятие". Однако такие конверторы 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. 4.3). Рассмотрим некоторые свойства ячейки.

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

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

Если тип **Текст**, то информация в ячейке в пользовательском режиме будет представлена так же, как введена в конфигураторе.

Если тип **Выражение**, то информация в ячейке — это выражение на встроенном языке. В пользовательском режиме вы увидите результаты вычисления этого выражения.

Если тип **Шаблон**, то в ячейку вводится текст с включенными в него выражениями на встроенном языке. Выражения заключаются в квадратные скобки. В пользовательском режиме будет выводиться текст и результат вычисления выражений.

Если тип **Фиксированный шаблон**, то 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.

Так как практически все деловые документы имеют прямоугольную структуру, шаблон печатной формы представляет собой совокупность прямоугольных областей, каждая из которых служит для выдачи какой-то части готового отчета: заголовочной, табличной и др. Такие области называются *секциями*. Любой секции отчета можно присвоить какое-то 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.

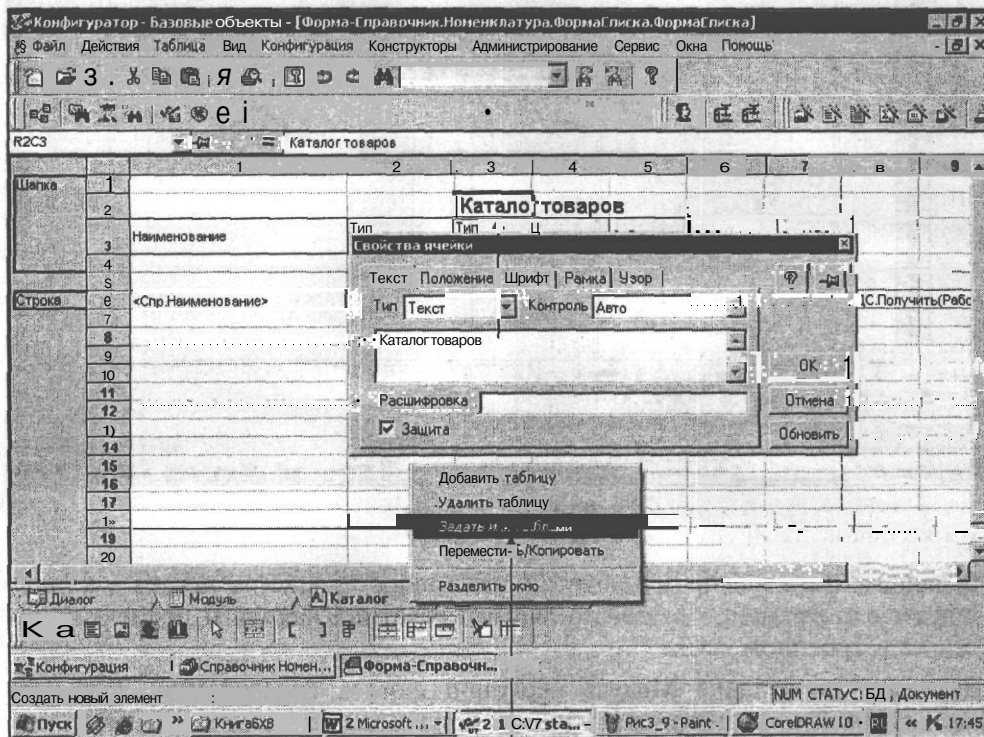
Сам процесс построения отчета происходит следующим образом. Сначала строится шаблон печатной формы из ячеек, строк, столбцов, секций. Одновременно пишется процедура, в которой реализуется алгоритм обработки шаблона печатной формы. В процессе выполнения процедуры из шаблона в нужном порядке извлекаются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

4.3. Печатные формы справочников

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. 4.4). В появившемся окне напишем новое имя шаблона: Каталог.

Начнем создание шапки нашего будущего отчета — заголовочной части. Посмотрим, где примерно должен располагаться заголовок, и впишем в ячейку

название отчета: Каталог товаров (рис. 4.4). 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**.



Меню для переименования шаблона
печатной формы

Рис. 4.4. Переименование шаблона печатной формы

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. 4.5).

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.

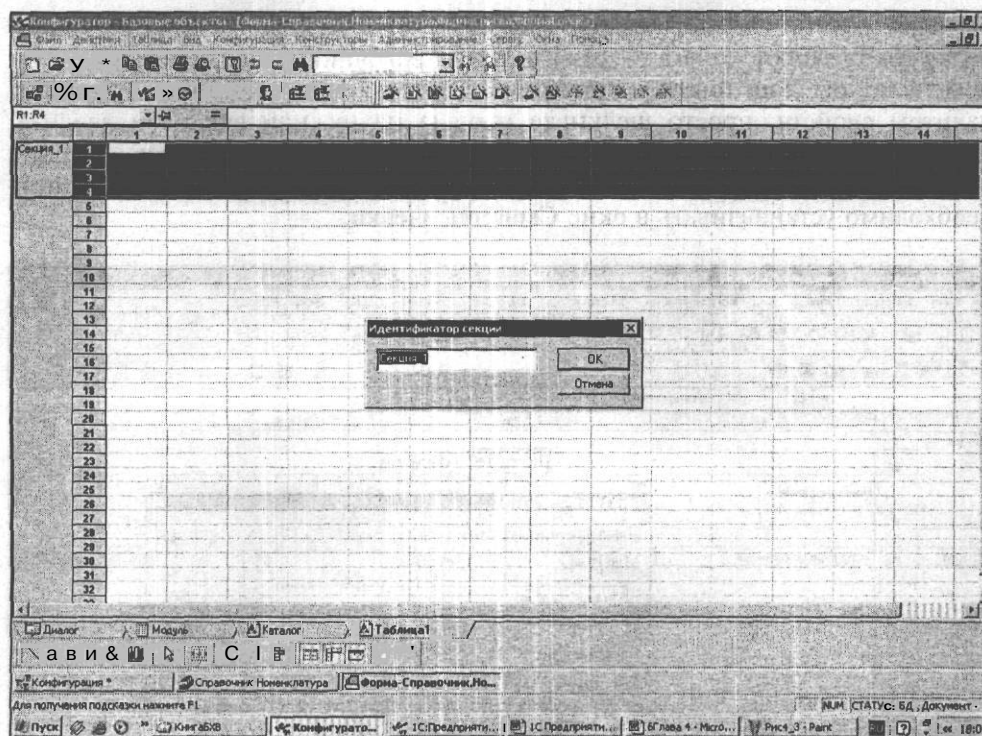


Рис. 4.5. Секция

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:

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. 4.2 — 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, указы-

вая 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

```
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:

```
Пока Спр.ПолучитьЭлемент 0=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 ИсходнаяТаблица(), 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:

```
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 подсказка окна печатной формы.

Теперь добавим операторы, формирующие печатную форму документа В Процедуру Печать () :

Листинг 4.1

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 0=1 Цикл
КонецЦикла;
КонецЕсли;
Таб.ТолькоПросмотр (1) ;
Таб.Опции (0, 0, 0, 0) ;
Таб.Показать ("Каталогтоваров" );
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 с использованием периодических реквизитов: Себестоимость, Цена, СтавкаНДС. Эти выражения должны записываться так: Спр.Себестоимость.Получить (РабочаяДата ()). Периодичность подразумевает наличие истории значений (см. разд. 2.2). 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, дейст-

вующий 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 4.2

```
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");
        КонечЦикла;
    КонечЕсли;
    Таб.ТолькоПросмотр (1);
    Таб.Опции (0, 0, 0, 0);
    Таб.Показать ("Каталог товаров");
КонечПроцедуры
```

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

- подогнать ширину колонок (это можно сделать, перетаскив мышью разделитель между колонками таблицы в верхнем заголовке — сером поле с номерами колонок);
- обвести рамкой ячейки;
- установить положение текста в ячейках;
- установить размер и вид шрифта.

Печатная форма закончена (рис. 4.6).

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 configurатора нужно в форме списка справочника Номенклатура перейти на вкладку **Диалог**. Щелкнуть левой кнопкой мыши по краю диалоговой формы. Появится гра-

нища 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, растяните форму немного вниз (рис. 4.7). 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 на пиктограмму **Кнопка** на панели инструментов **Элементы диалога** (рис. 4.7). Далее подвести курсор к левому верхнему углу нужного места, нажать левую кнопку мыши и, не отпуская ее, передвинуть указатель мыши ниже и **правее** так, чтобы очертить прямоугольник, затем отпустить кнопку мыши.

Теперь обратимся к палитре свойств созданной кнопки.

Щелчком 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**. Это заголовок кнопки. Пользователю будет понятно, для чего эта кнопка предназначена. Чтобы определить действия, необходимо переключиться на вкладку **Дополнительные**. Поле **Формула** нам уже знакомо. В этом поле нужно написать вызов процедуры печать () (рис. 4.8).

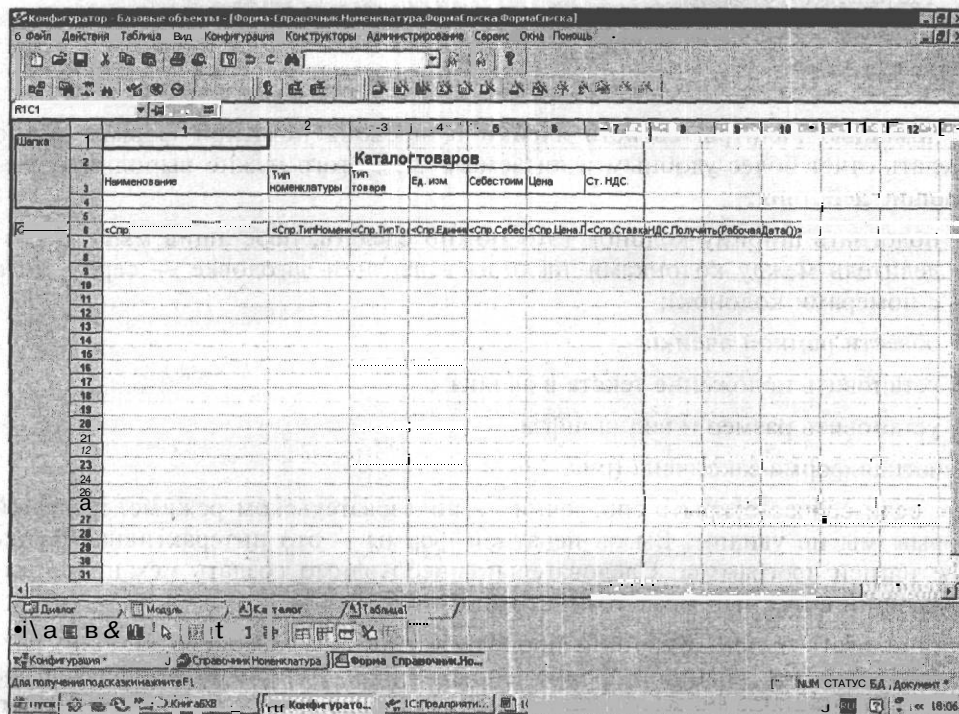


Рис. 4.6. Готовый шаблон печатной формы

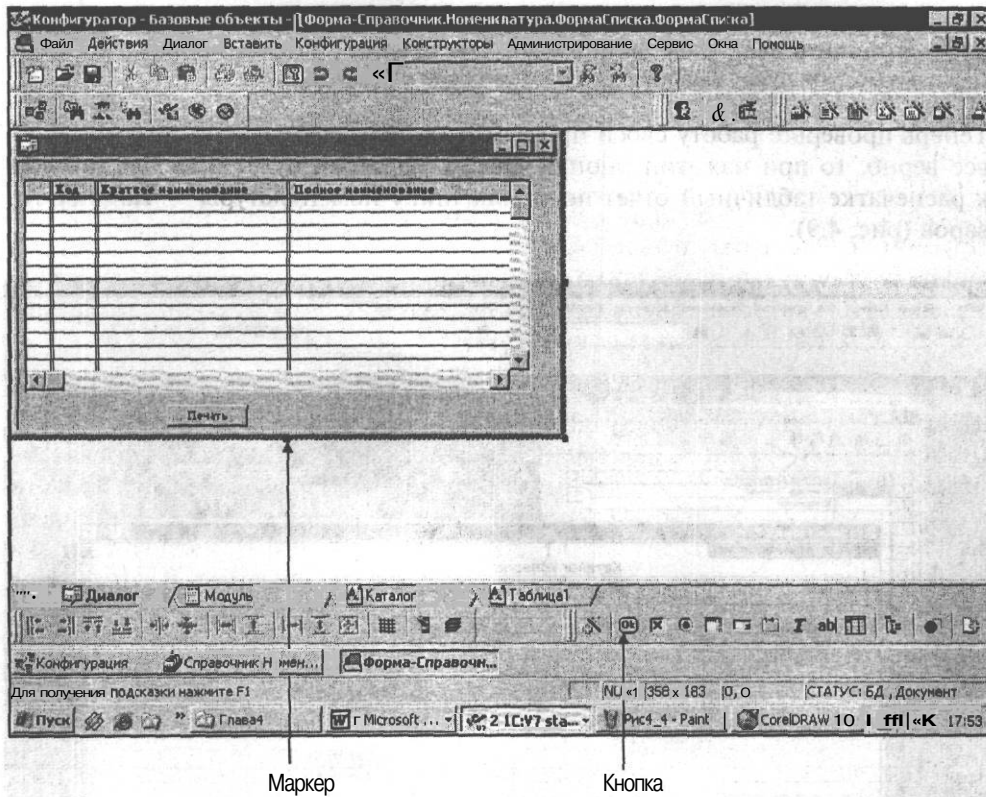


Рис. 4.7. Создание кнопки в экранной форме

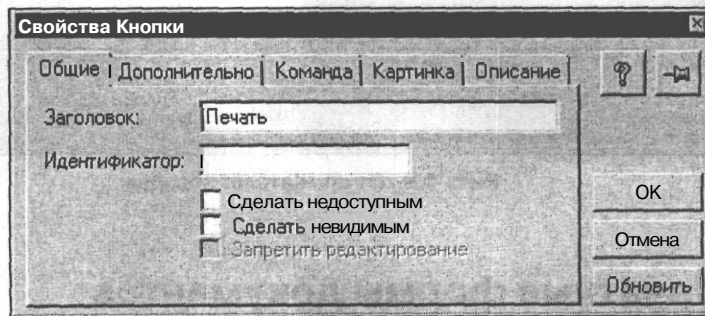


Рис. 4.8. Вызов процедуры из палитры свойств кнопки

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. 4.9).

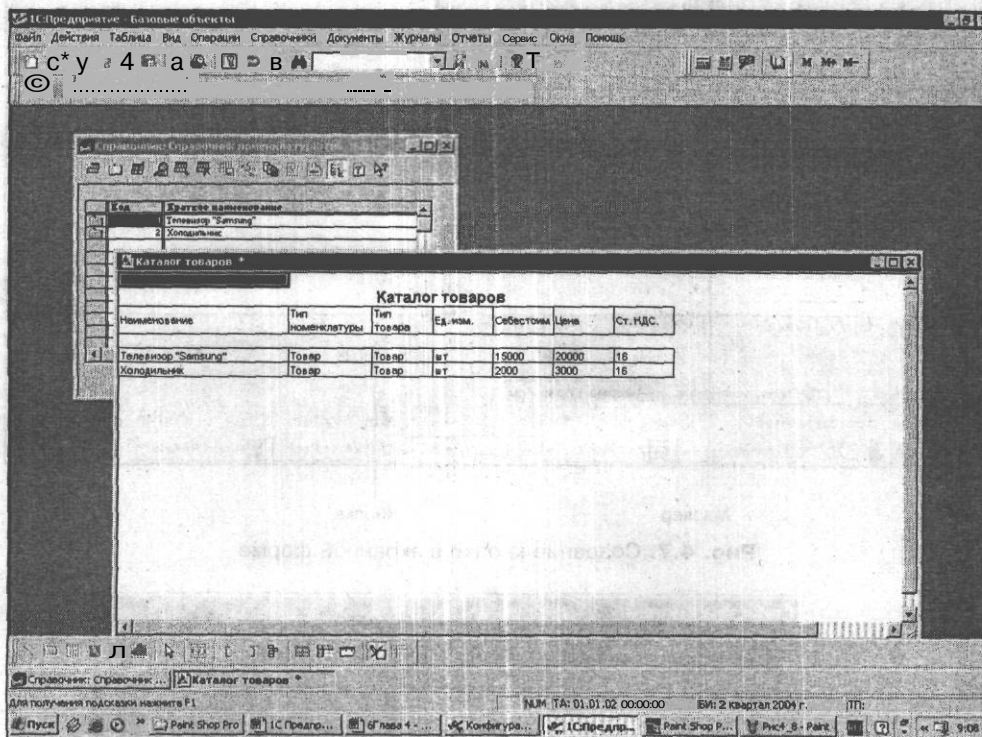


Рис. 4.9. Отчет Каталог товаров

4.4. 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 Приходная накладная. Здесь начнем работать с вкладкой **Таблица**. 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, используя тип текста **Шаблон** (рис. 4.10).

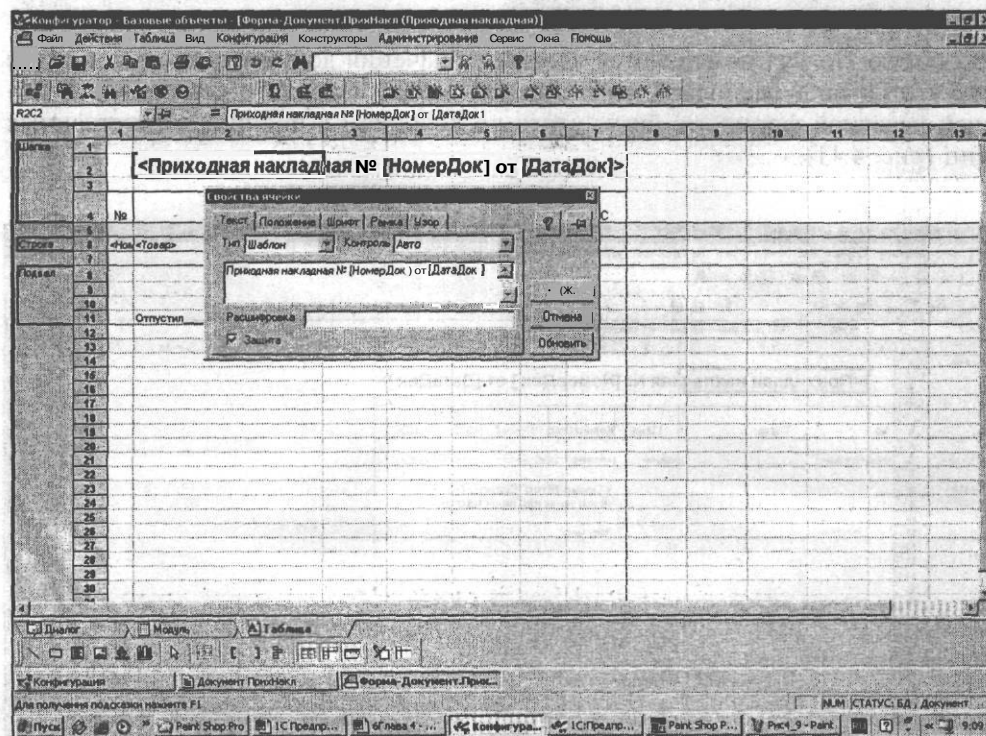


Рис. 4.10. Наименование документа имеет тип текста **Шаблон**

Это сделано для того, чтобы заголовок, включающий обычный текст (наименование документа) и вычисляемое выражение (номер документа и дата документа) выглядел как единое целое. Если заголовок, номер и дату документа поместить в отдельных ячейках, то потребуются дополнительная подгонка ширины колонок, тем не менее, заголовок все равно может быть 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**.

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

В секцию Подвал помещаются итоговые значения документа, подписи и все то, что обычно располагается в нижней части бумажного документа. В нашей печатной форме это будут итоги по колонкам и подписи ответственных лиц (рис. 4.11).

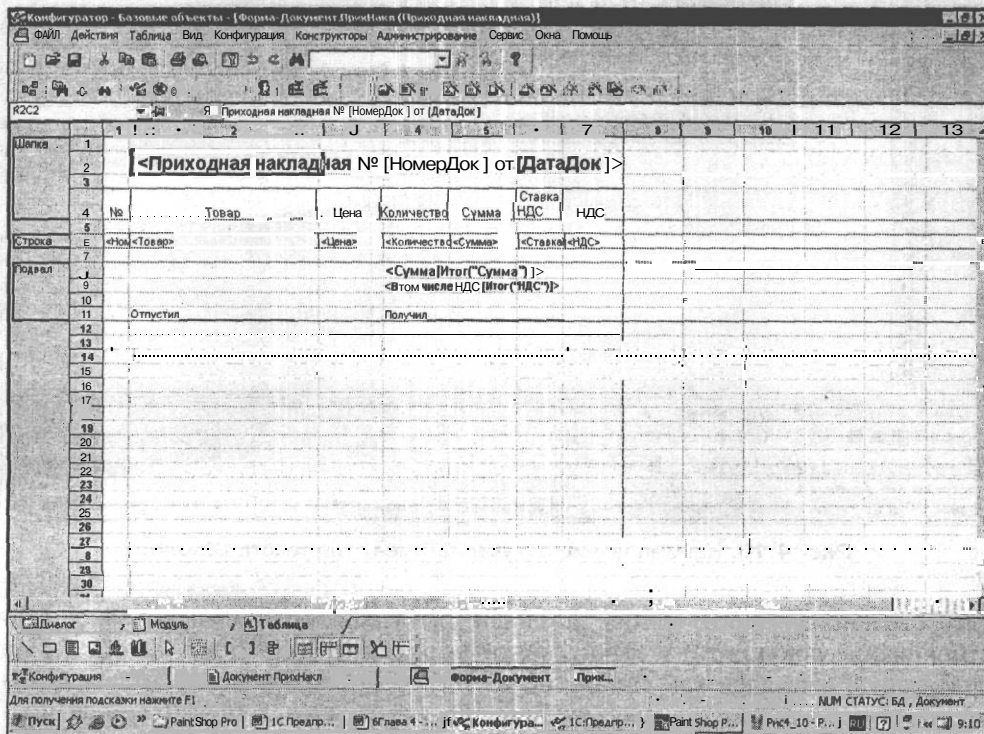


Рис. 4.11. Шаблон печатной формы документа Приходная накладная

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. 3.5*), необходимо написать оператор итог, а далее, в скобках и кавычках — наименование колонки:

```
Итог("Сумма");
```

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 4.3

```
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");
    КонечЦикла;
КонечЕсли;
Таб.ВывестиСекцию ("Подвал");
Таб.ТолькоПросмотр (1);
Таб.Опции (0,0,0,0);
Таб.Показать ("Приходнаянакладная");
КонечПроцедуры
```

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. И отличаются операторы выборки: вместо выбрать Элементы о — ВыбратьСтроки (), вместо Получить Элемент () — ПолучитьСтроку () .

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

Примечани

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

1С:Предприятие - базовые объекты - [Приходная накладная *]

А) файл Действия Таблица Вид Операции Справочники Документы Журналы Отчеты Сервис Окна Помощь

Приходная накладная № 3 от 11.02.03

№	Товар	Цена	Количество	Сумма	Ставка НДС	НДС
1	Телевизор "Samsung"	10000	1	10000	16	1379.31
2	Телевизор "Samsung"	10000	1	10000	16	1379.31
				Сумма 20000		
				в том числе НДС 2758,62		

Отпустил _____ Получил _____

Журнал документов Приходная накладная - 3 Приходная накладная -

Для получения подсказки нажмите F1

NUM ITA: 01.01.02 00:00:00 BI: 2 квартал 2004 Г. TP:

Рис. 4.12. Печатная форма документа Приходная накладная

Конструктор печати документа: Шапка

Реквизиты используемые в шапке документа

↑ ↓ × × Все

<ul style="list-style-type: none"> [-] НомерДок [-] ДатаДок 	<ul style="list-style-type: none"> [-] Номер [-] Дата [-] Поставщик [-] Итого("Цена") [-] Итого("Количество") [-] Итого("Сумма") [-] Итого("СтавкаНДС") [-] Итого("НДС")
-------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Далее> Отмена Помощь

Рис. 4.13. Диалговое окно конструктора печати документа

Запустив "1С:Предприятие" в пользовательском режиме, проверим сделанную нами работу (рис. 4.12).

Теперь автоматизируем процесс создания процедуры печати с помощью конструктора печати.

В режиме конфигуратора откройте форму документа Расходная накладная. Находясь в модуле формы документа, выберите пункт меню **Конструкторы | Печать**. Появится поле ввода с именем готовящейся процедуры, по умолчанию — Печать. Нажмите кнопку ОК, и появится первое окно диалоговой формы конструктора печати — **Шапка** (рис. 4.13). В правой части окна реквизиты шапки документа, в левой — те реквизиты, которые мы будем включать в шаблон печатной формы. Перенос реквизитов из правой части в левую осуществляется при помощи стрелок. По умолчанию в левой части окна уже есть обязательные реквизиты **НомерДок** и **ДатаДок**. Нам нужно перенести реквизит **Поставщик**. Нажмите кнопку **Далее** и перенесите все реквизиты табличной части документа из правой части диалога в левую при помощи стрелки с фигурной скобкой. Нажмите кнопку **Далее** и переместите в левую часть диалога реквизиты, которые будут помещены в подвал шаблона печатной формы документа: **Итог(Сумма)** и **Итог(НДС)**. Нажмите кнопку **Далее**, выставьте опции диалога, как указано на рис. 4.14, и нажмите кнопку **Готово**.

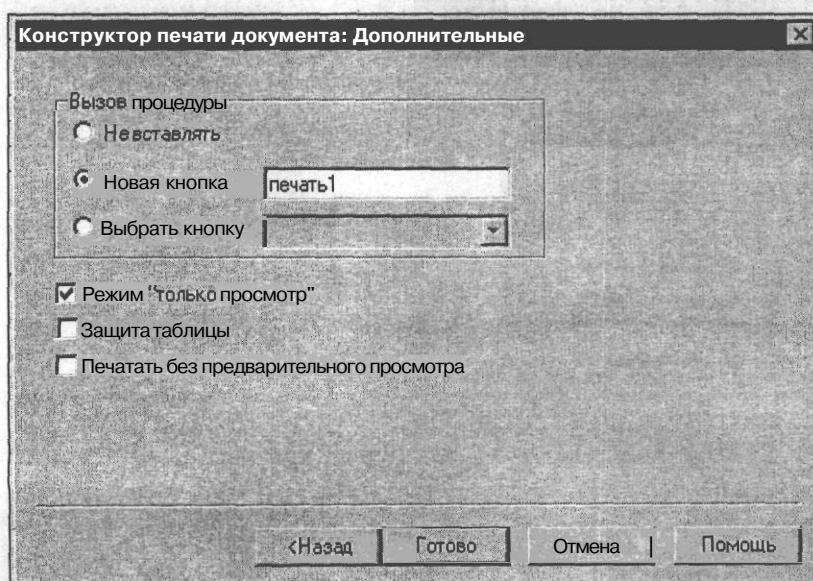


Рис. 4.14. Установка параметров печати и кнопок

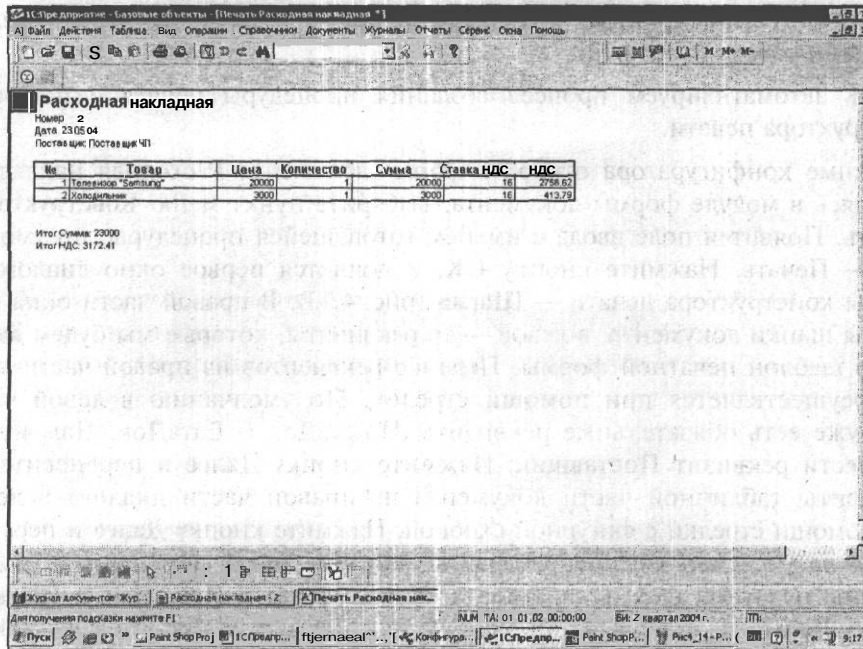


Рис. 4.15. Печатная форма документа Расходная накладная

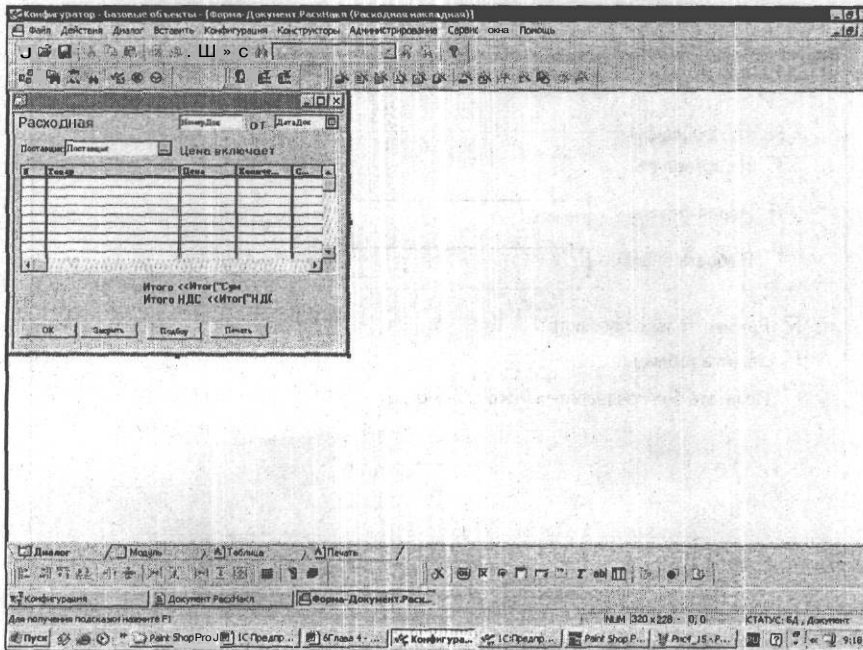


Рис. 4.16. Итоги по колонке в экранной форме документа (режим конфигулятора)

Посмотрите результат работы конструктора в конфигураторе и запустите "ИС:Предприятие" в пользовательском режиме (рис. 4.15). Конструктор создает новый шаблон печатной формы с наименованием Печать. 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.

Итоги по колонкам можно добавить в экранную форму документа. В нижней части экранной формы необходимо вставить элемент диалога **Текст**, в свойствах текста во вкладке **Дополнительно** в поле формул вписать (рис. 4.16):

```
Итог ("Сумма") ;
```

4.5. 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, в режиме конфигуратора войдите в меню **Файл** и выберите пункт **Новый**. В появившемся диалоговом окне выберите пункт **Таблица** (рис. 4.17) и нажмите кнопку **ОК**.

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 Приходная накладная, скопируйте содержимое в буфер обмена и вставьте 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**.

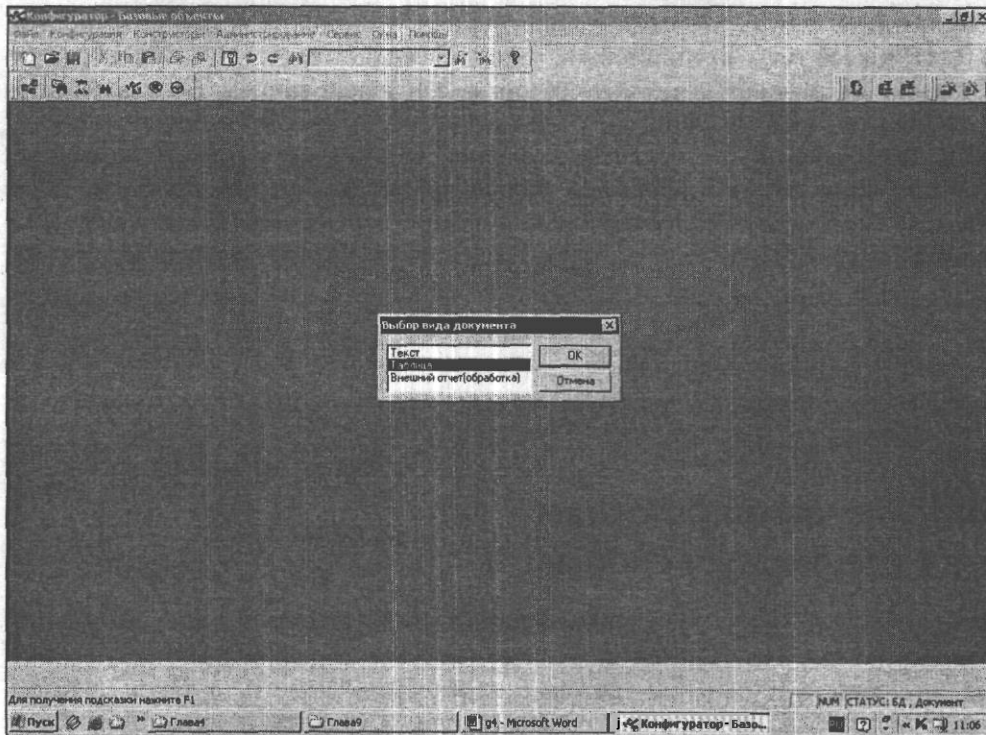


Рис. 4.17. Диалоговое окно для выбора вида внешнего файла

В процедуре печати модуля формы документа Приходная накладная внесите следующие изменения:

Листинг 4.4

Процедура Печать ()

```

Таб=СоздатьОбъект ("Таблица");
//Таб.ИсходнаяТаблица ("Таблица");
Таб.ИсходнаяТаблица ("Table\ПрихНакл.mxl");
Таб.ВывестиСекцию ("Шапка");
Если ВыбратьСтроки 0=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

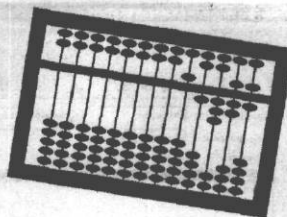
Теперь необходимо указывать путь к шаблону печатной формы. Если папка Table расположена не в каталоге рабочей базы, необходимо указывать полный путь к файлу, начиная с имени диска.

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 сделанные изменения.

Глава 5



Другие базовые объекты

В этой главе мы рассмотрим другие виды базовых объектов метаданных (см. гл. 1): Перечисления, Обработки, Журналы документов, а также специальный агрегатный тип данных Список значений. Объект метаданных Отчеты логичнее изучить несколько позже.

5.1. Перечисление

В гл. 2 мы создали несколько справочников. Как вы помните, в конфигураторе создается только шаблон для записи данных в справочник. Пользователь сам 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.

В разд. 2.3 был создан справочник Контрагенты с реквизитами Адрес и Полное наименование. Добавим реквизит, содержащий информацию, 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. 5.1).

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.

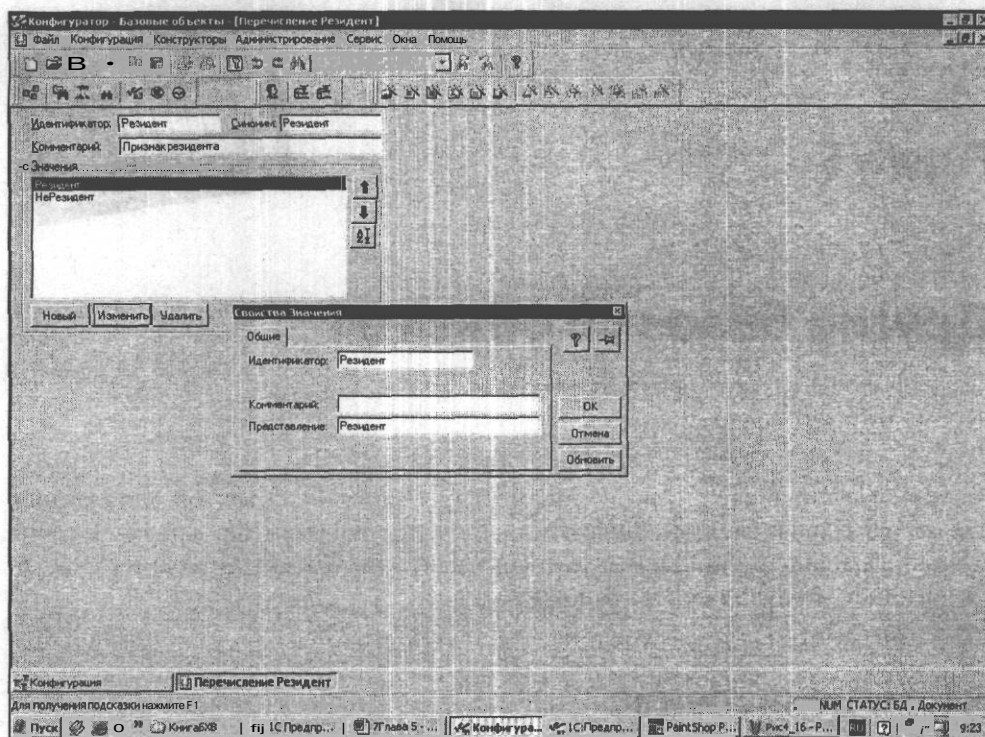


Рис. 5.1. Ввод значений перечисления

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

О Идентификатор — НеРезидент;

О Представление — Не резидент.

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

Откроем окно редактирования справочника **Контрагенты**. Добавим новый реквизит **Резидент**. Тип значения **Перечисление.Резидент** (рис. 5.2). Этот реквизит можно добавить в форму списка справочника с помощью кнопки **Реквизиты** панели инструментов **Элементы диалога** или через меню **Вставить**.

Если все сделано правильно, то в пользовательском режиме при записи контрагента реквизит **Признак резидента** будет раскрываться в виде списка. Из этого списка можно выбрать любое из двух значений: **Резидент** либо **Не резидент**. Это удобно с двух сторон: ввод данных в реквизит **Резидент** ускоряется, и нет ошибок ввода.

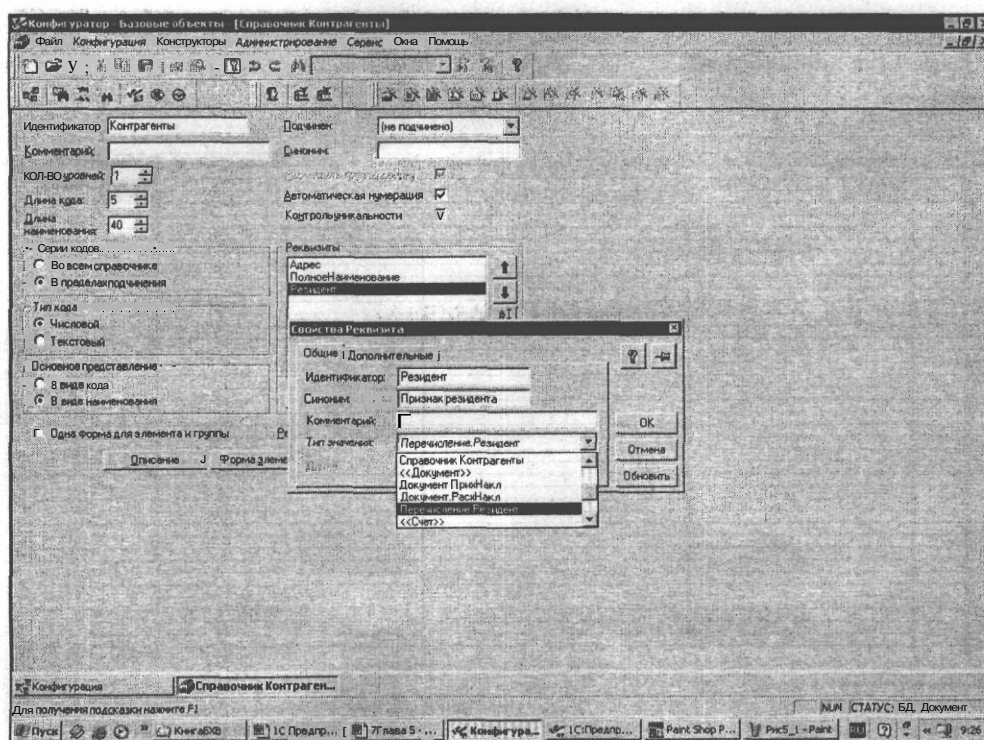


Рис. 5.2. Реквизит справочника типа Перечисление

Вообще 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, то есть 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.

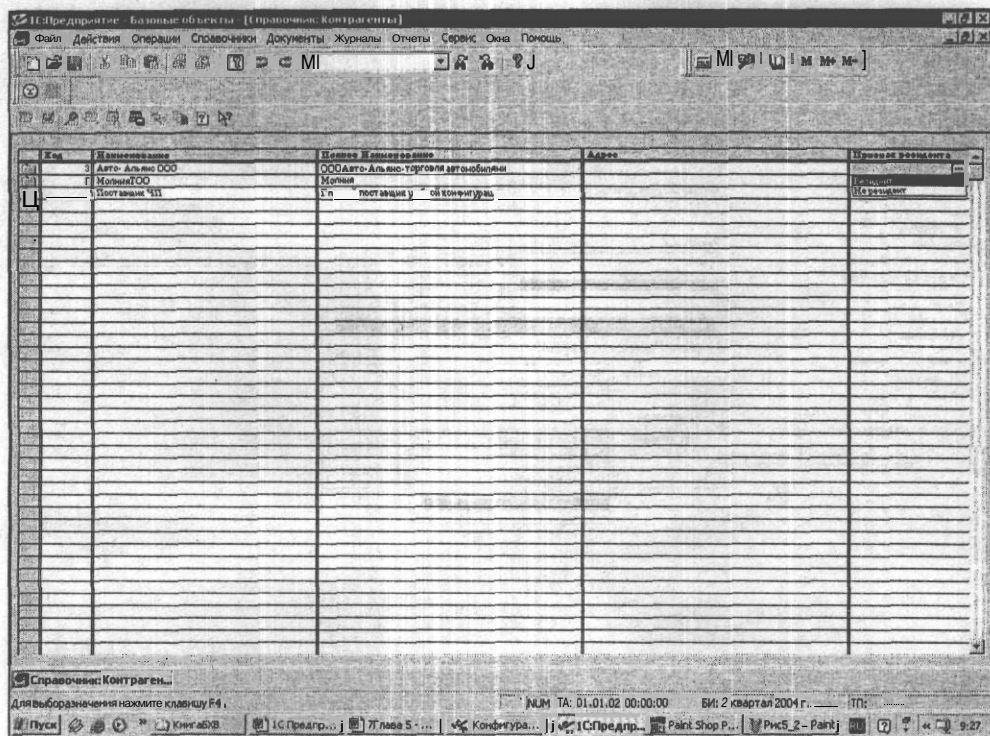


Рис. 5.3. Справочник Контрагенты с реквизитом Признак резидента типа Перечисление

5.2. Обработка

Обработки — вид метаданных, которые используются для выполнения различных действий над информационной базой. Те, кто имеет опыт работы в "1С:Предприятие", вероятно, знакомы с настройкой параметров учета, обработкой документов, путеводителем по конфигурации, 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. Дважды щелкнув левой кнопкой мыши по идентификатору обработки в дереве метаданных, мы вызовем форму обработки (рис. 5.4). В отличие от рассмотренных объектов метаданных, обработка не имеет окна редактирования.

В этой обработке мы будем задавать значения констант. В нашей учебной конфигурации их всего две: Дата регистрации предприятия и ФИО главного

бухгалтера. В форму обработки нужно добавить поля, чтобы вводить новые значения констант. Для этого используем реквизит диалога. Для его вставки через меню выберем опции **Вставить | Реквизит диалога** либо нажмем на панели инструментов **Элементы диалога** кнопку **Реквизит диалога**.

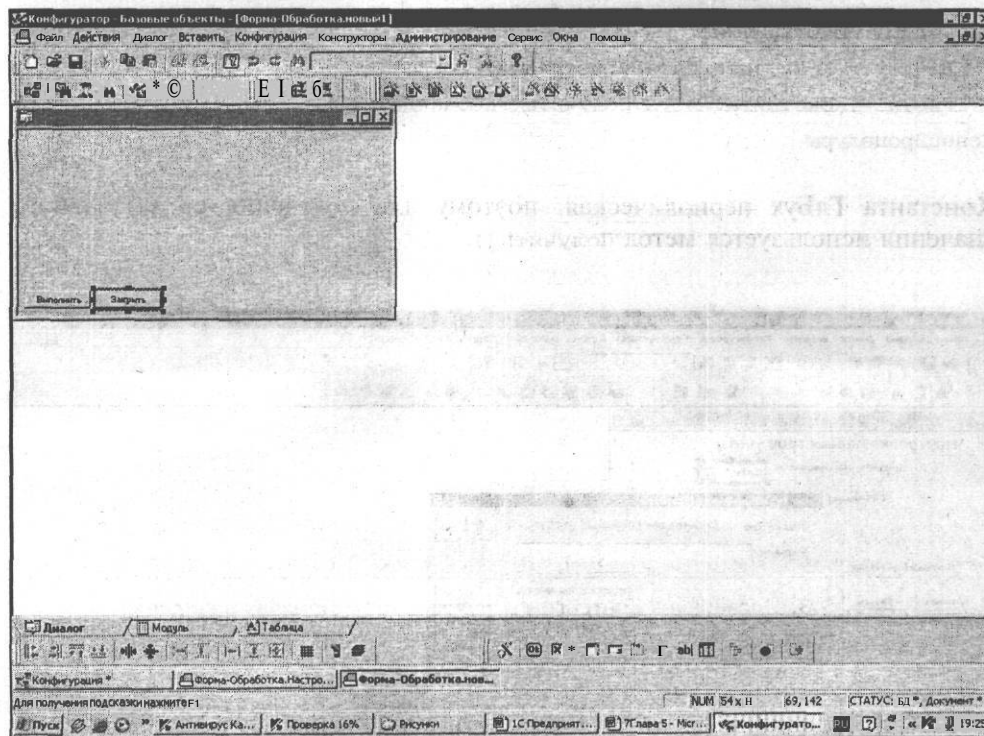


Рис. 5.4. Экранная форма обработки

Для ввода значения константы **ДатаРегистрации** создадим реквизит диалога с идентификатором **ДатРег**, типом значения **Дата**. Для удобства пользователя выставим в палитре свойств реквизита флажок **Имеет кнопку выбора** (рис. 5.5). Для ввода ФИО главного бухгалтера создадим реквизит диалога с идентификатором **ГлБух**, типом значения **Строка** длиной 20, соответственно типу значения константы **ГлБух**.

Перейдем в модуль формы обработки. Удалим созданную по умолчанию процедуру **выполнить ()** и создадим две новых predefined процедуры **ПриОткрытии ()** и **ПриЗакрытии ()**.

Назначение predefined процедуры **ПриОткрытии ()** понятно из ее названия. Она выполняется при открытии формы обработки. С помощью

нее мы помещаем в реквизиты ДатРег и ГлБух значения констант, актуальные на дату открытия обработки:

Листинг 5.1

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

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

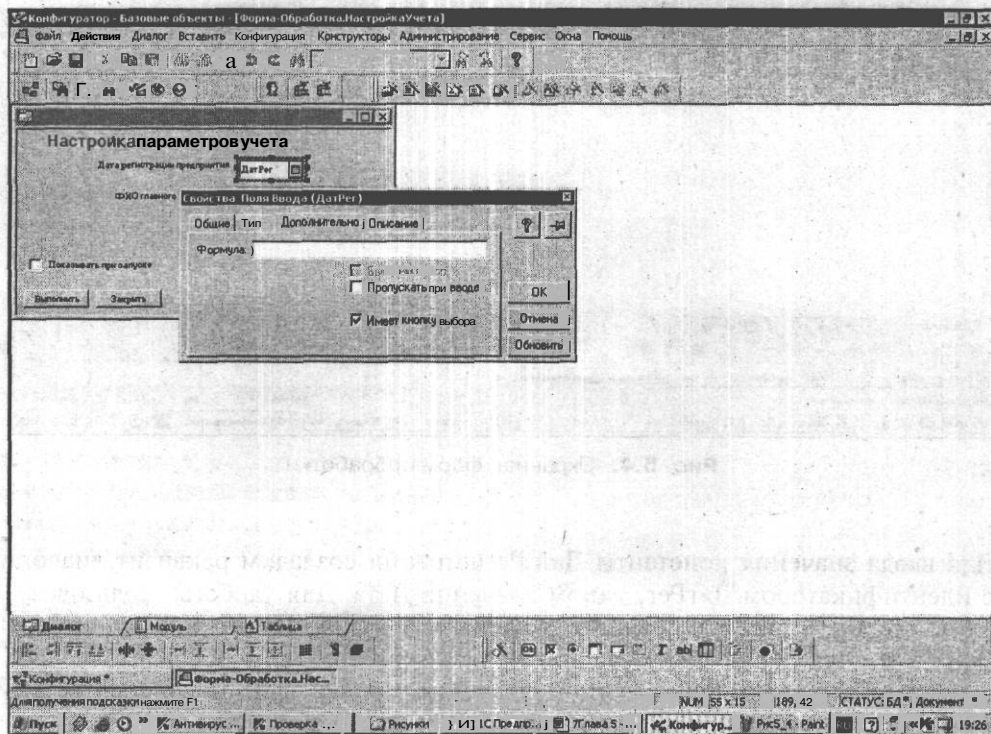


Рис. 5.5. Установка свойства флажка **Имеет кнопку выбора**

Этот метод возвращает значение периодической константы на заданную дату. Дата задается системной функцией РабочаяДата (). Данный код аналогичен коду периодических реквизитов справочников (см. разд. 4.4).

В реквизитах обработки можно задать новые значения констант. При закрытии обработки эти значения присвоятся константам при помощи предопределенной Процедуры ПриЗакрытии () :

Листинг 5.2

```
Процедура ПриЗакрытии ()
    Константа.ДатаРегистрации = ДатРег;
    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

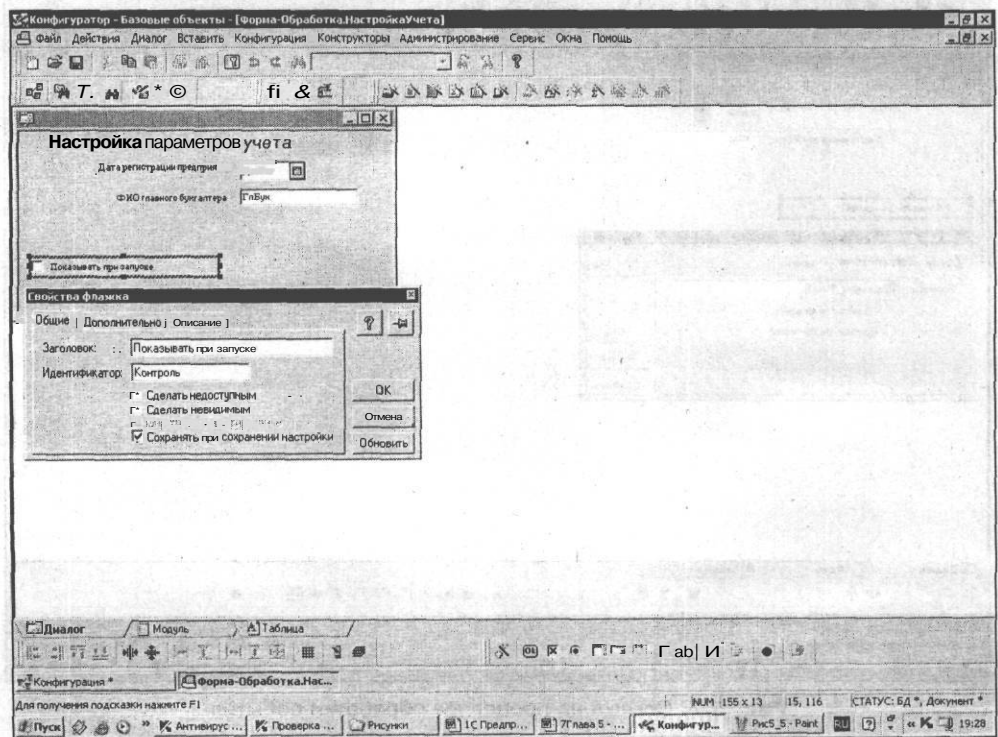


Рис. 5.6. Флажок для диалога вызова обработки

механизм, обеспечивающий вывод обработки при первом запуске либо при следующих запусках, иначе обработка будет вызываться пользователем через меню.

Диалог вызова обработки реализуем посредством флажка. Идентификатор — Контроль, заголовок — Показывать при запуске (рис. 5.6).

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 Контроль, тип значения — число, длина — 1. Событие — щелчок мышью — обрабатывается процедурой Щелкнуть_Флаг(). Ссылка на эту процедуру устанавливается в палитре свойств флажка, аналогично ссылке на процедуру в свойствах элемента диалога Кнопка (см. разд. 4.4) (рис. 5.7).

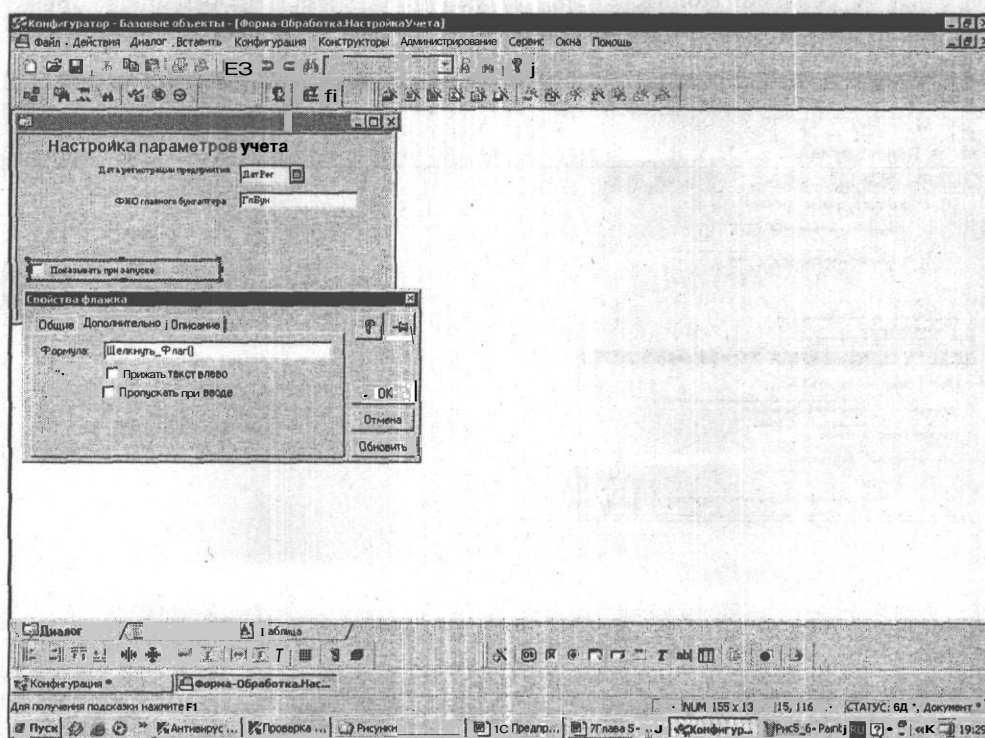


Рис. 5.7. Ссылка на процедуру обработки события

Листинг 5.3

```
Процедура Щелкнуть_Флаг()  
    ПоказыватьНастройкуТорговли=? (Контроль=1, "1", "2");  
    Константа.Контроль = ПоказыватьНастройкуТорговли;  
КонецПроцедуры
```

В первой строке процедуры переменной ПоказыватьНастройкуТорговли 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 ПоказыватьНастройкуТорговли.

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 проверку 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** (значения 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
```

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.

5.3. 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 два журнала: **ПрихНакл** и **РасхНакл**. Они находятся в ветке дерева метаданных **Журналы документов** (рис. 5.8).

В журнале приходных документов (рис. 5.9) видны графы **Дата**, **Время**, **Документ** (вид документа) и **Номер**. Если нам нужны дополнительные сведения, то необходимо создать графы журнала. Дважды щелкнув мышью по ветви **ПрихНакл**, перейдем в окно редактирования формы журнала. Вызовем диалог для создания новых граф, щелкнув по кнопке **Новый**. В появившемся диалоговом окне информация о возможных значениях граф представлена в виде стандартного дерева Windows. Последовательно раскрывая уровни, выбираем необходимое значение (**Поставщик**), которое при щелчке по стрелке попадает в поле **Выбранные значения** (рис. 5.10). 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. 5.11).

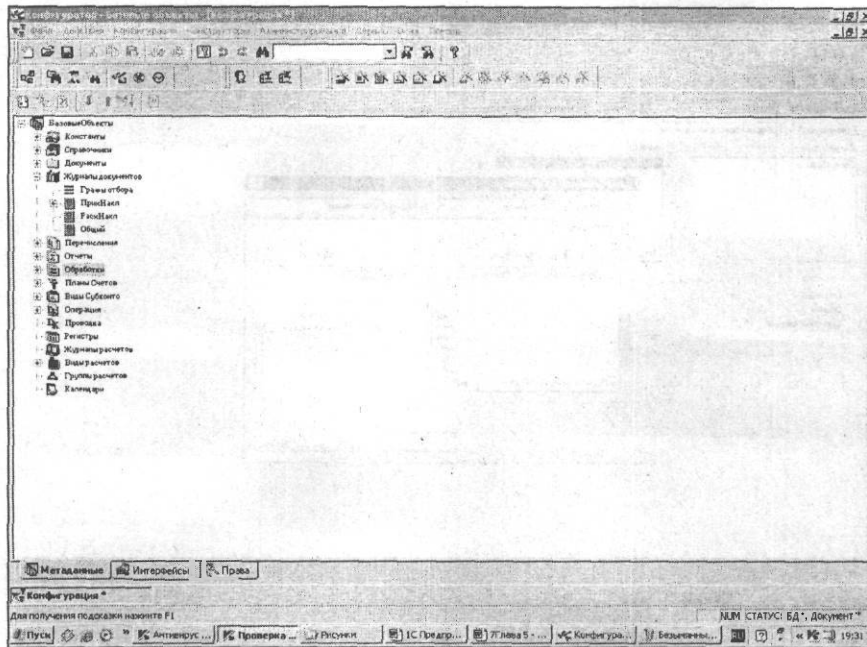


Рис. 5.8. Ветвь метаданных Журналы документов

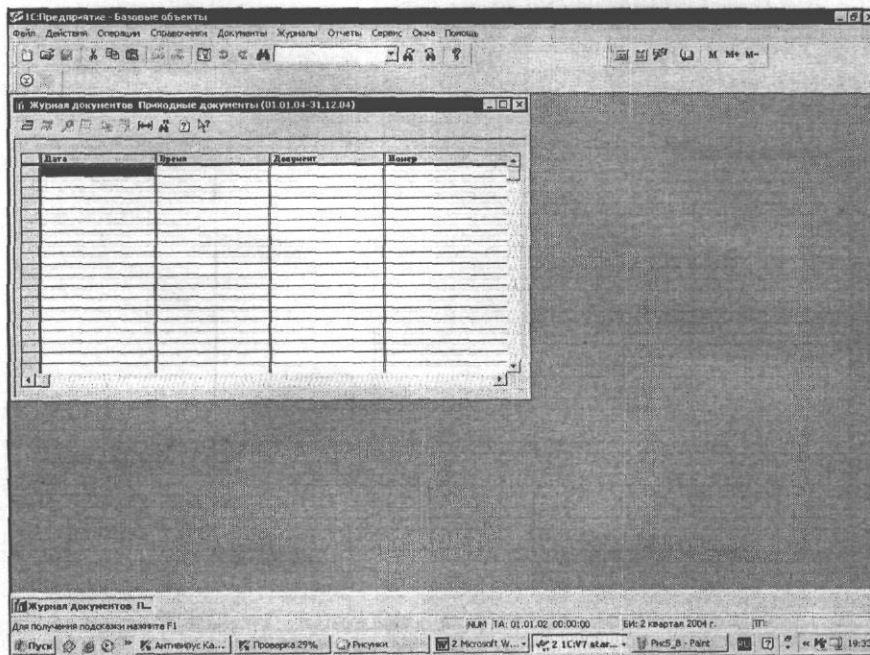


Рис. 5.9. Графы журнала приходных документов (в пользовательском режиме)

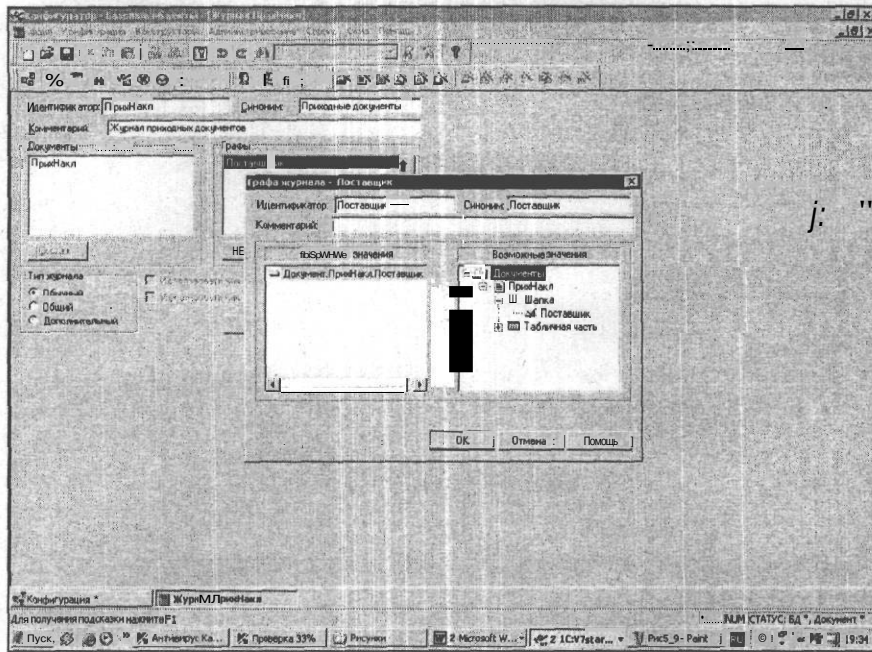


Рис. 5.10. Создание графы журнала документов

Рис. 5.11. Вид журнала приходных документов

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

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 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*.

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.

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.

5.4. 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 пользователя необходимо дать ему 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, не сохраняется на жестком диске компьютера, а существует только в оперативной памяти во время исполнения соответствующей процедуры.

Возможности применения списка значений весьма широки. Первоначальное представление об этом объекте дается здесь, а в гл. 8 приводится более сложный пример его использования.

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 Приходная накладная и перейдите к процедуре печать (). Так же, как и при работе со справочниками во встроенном языке, создадим объектную переменную Выбор, которой присвоим ССЫЛКУ на объект Список значений:

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

Теперь пополним этот список необходимыми нам значениями:

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

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

Выражение (1 или 2) используется непосредственно 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;
- Тайм-аут** — 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, задано по умолчанию;

- 1 — выбор производится в виде меню, которое подстраивается по месту текущего элемента диалога или ячейки таблицы, откуда вызвано;
- 2 — выбор маленьким 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 — первый.

Так как переменные поз и код не имеют справа от себя знака присваивания, то их следует 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 печати должен получиться таким:

Листинг 5.6

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\ax1");
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;
КонецПроцедуры

```

Для проверки запустите "1С:Предприятие" в пользовательском режиме. При нажатии кнопки **Печать** должно появиться меню, как на рис. 5.12.

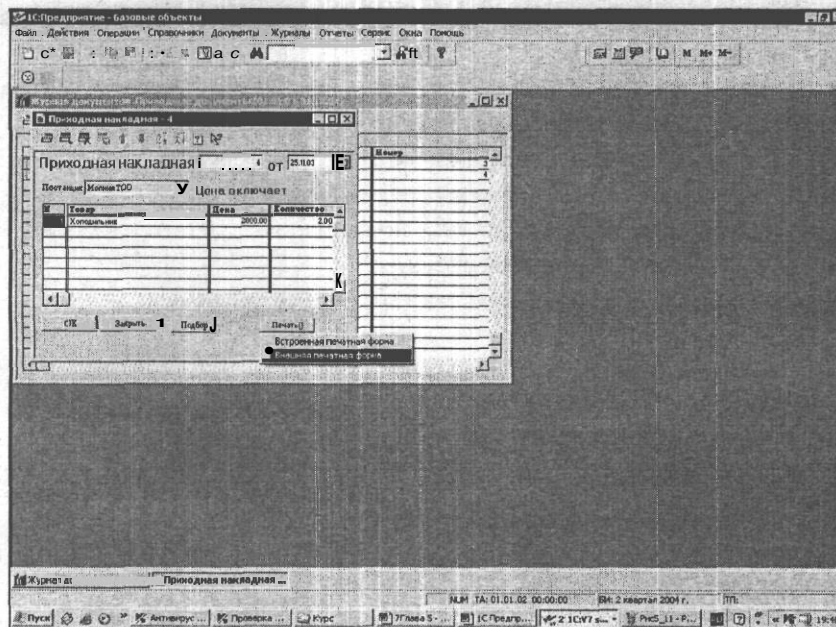
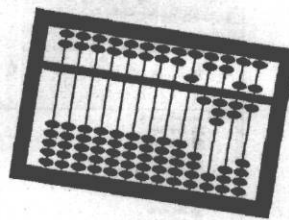


Рис. 5.12. Список значений в процедуре печати

Глава 6



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 знакомы в гл. 4. Однако часто нужную информацию невозможно получить, работая непосредственно с документами, справочниками и т. п. Кроме того, во многих случаях использование подобных средств неэффективно 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 позволяет группировки данных по различным критериям и накопления числовых функций.

Результатом работы запроса должна стать таблица, наполненная данными и расположенная на локальной машине пользователя.

6.1. 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, обрабатывающий необходимые нам сведения, содержащиеся в приходных документах. Идентификатор — ПриходТовара, синоним — Приход товара (рис. 6.1).

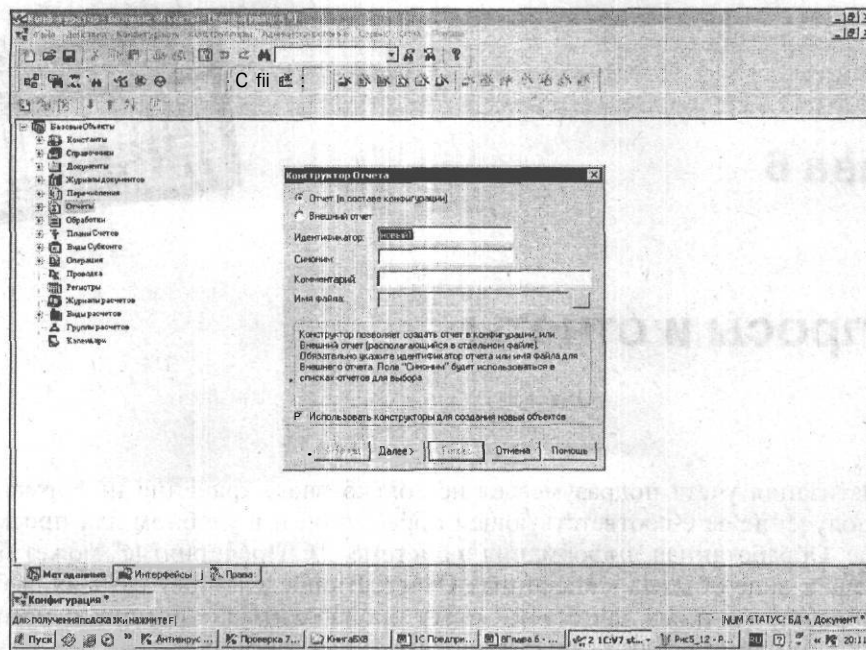


Рис. 6.1. Начало конструирования диалоговой формы отчета

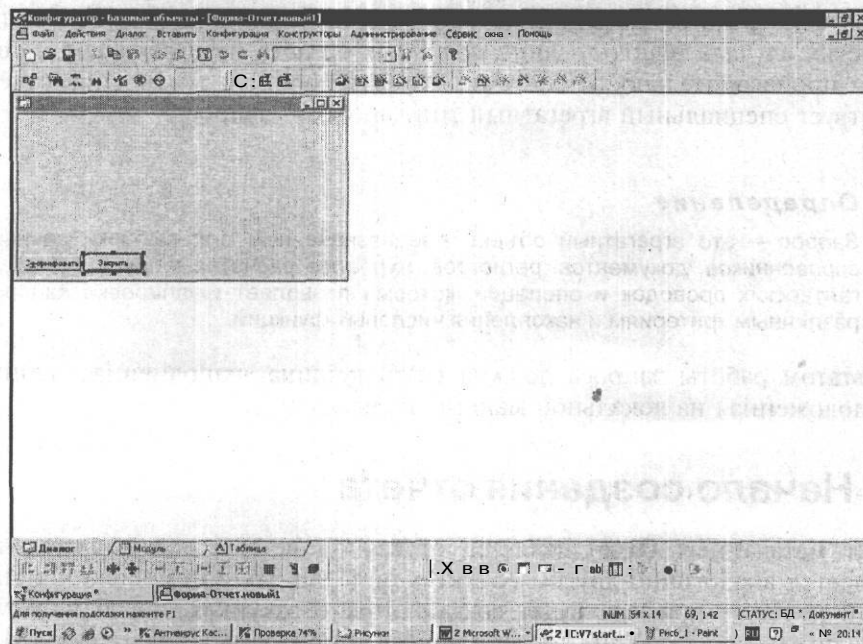


Рис. 6.2. Прототип диалоговой формы отчета

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. 6.2).

6.2. 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 существуют определенные правила:

- Сначала объявляются переменные запроса, как ссылки на какие-либо атрибуты или реквизиты объектов.
- Затем описываются функции, которые будут рассчитывать (суммировать, вычислять среднее и т. п.) значения, на основании данных числовых переменных запроса.
- Далее описываются группировки запроса, по каким переменным запроса необходимо сгруппировать данные (выполнить функции запроса).
- Описать условия, какими значениями переменных нужно ограничить формирование результатов запроса.

Создадим запрос в нашей учебной конфигурации. Это будет запрос по документу Приходная накладная. Цель запроса — получить информацию о том, сколько и каких товаров поставили нам наши поставщики.

В форме отчета перейдем в модуль. Выберем пункты меню **Конструкторы | Запрос**. В диалоговом окне зададим имя запросу. Можно оставить и имя по умолчанию — **Сформировать**. Нажимаем кнопку **ОК**, раскрывается диалоговое окно конструктора запроса (рис. 6.3). Управление конструктором осуществляется с помощью кнопок **Далее**, **Назад**, **Отмена**.

Отметим в окне конструктора, что период выборки будет назначаться в диалоге формы отчета. В поле **Обрабатывать Документы** выберем пункт **все**. Ведь 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 **ПрихНакл**, чтобы видеть реквизиты документа. Нажмем кнопку **Добавить переменную**, затем щелкнем мышью по реквизиту **Поставщик** и перенесем его в соседнее окно слева, с помощью стрелки (рис. 6.4). Теперь также добавим переменные для выборки товара, количества и суммы. Нажмем кнопку **Далее**.

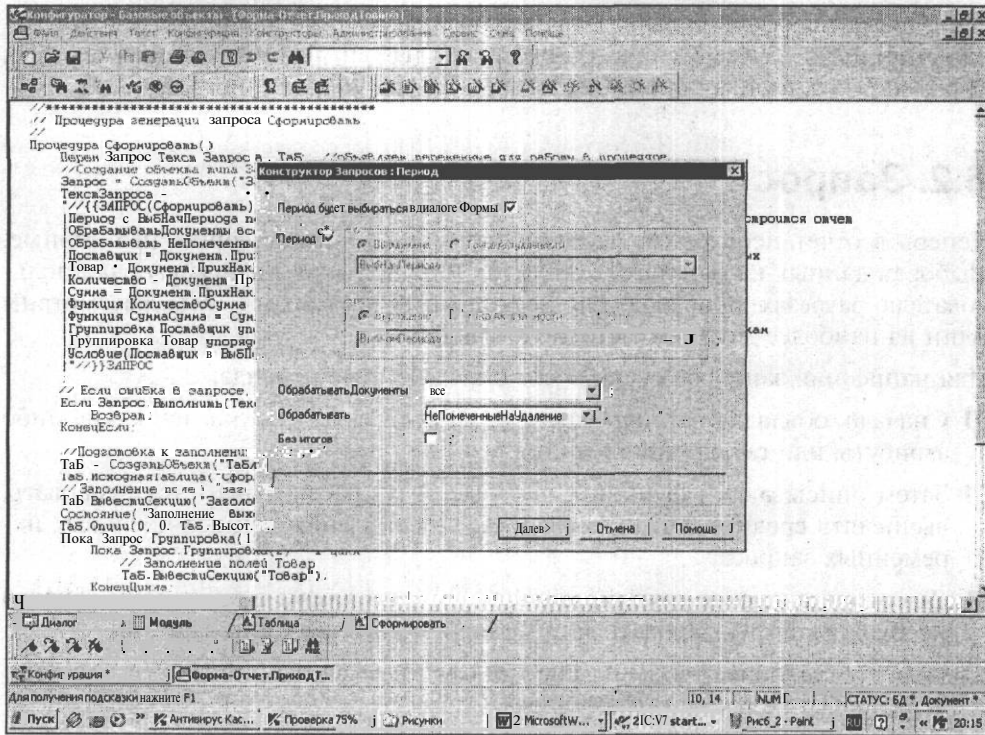


Рис. 6.3. Конструктор запроса

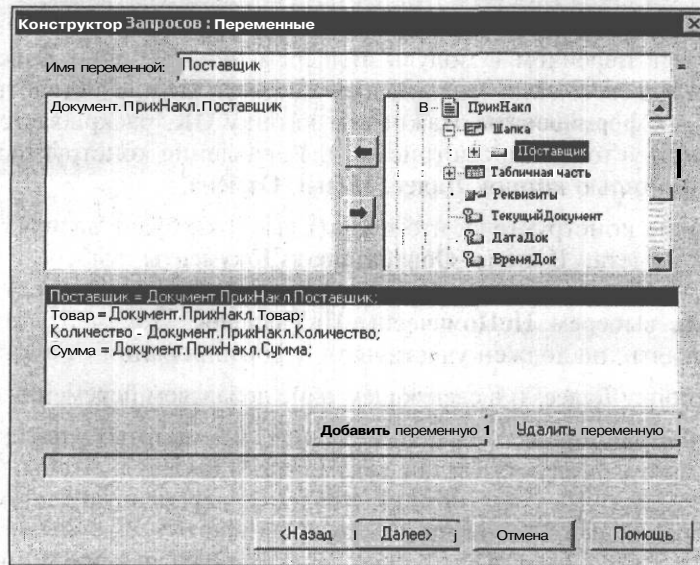


Рис. 6.4. Процедура добавления переменных в запрос

В диалоговом окне конструктора запроса добавим функции для расчета количества товара и суммы, на которую поставлен товар (рис. 6.5).

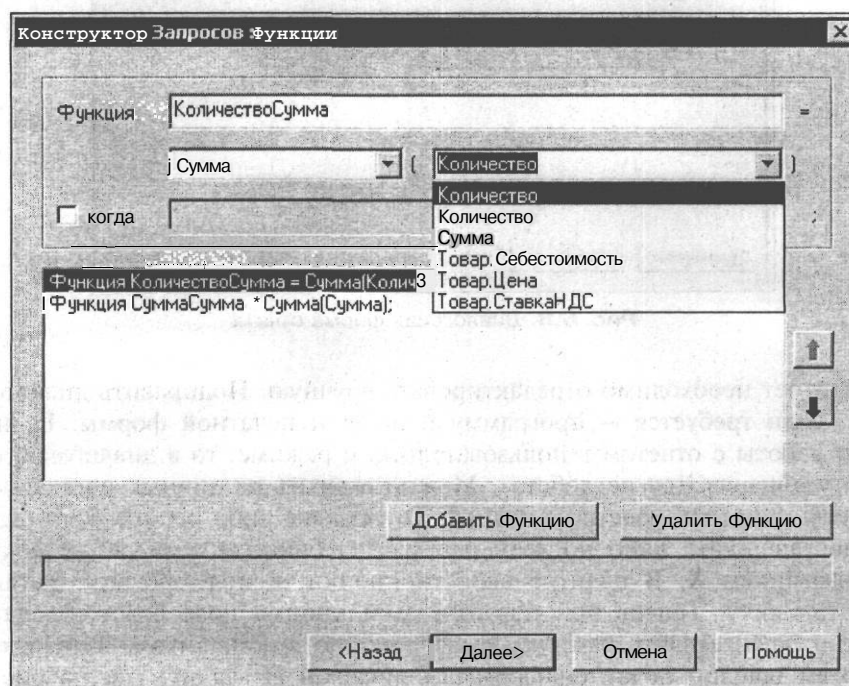


Рис. 6.5. Процедура добавления функций в запрос

Перейдем в следующее окно и добавим группировки. Переменные **Поставщик** и **Товар** выстроим в алфавитном порядке наименований.

В следующем окне добавим условие. Нажмем кнопку **Добавить условие**. Поставим флажки **Элементарное условие** и **Выбирать в диалоге**. Теперь выберем **условие** **Поставщик** в **ВыбПоставщик**.

С Замечание)

Условие **в** не эквивалентно знаку равенства. Если вместо условия **в** поставить условие **=**, то, 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 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 (рис. 6.6).

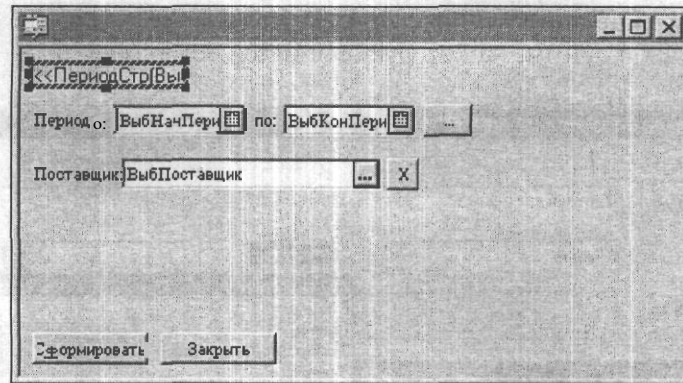


Рис. 6.6. Диалоговая форма отчета

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 поставщика. То есть мы либо работаем по отдельным поставщикам, либо по всем разом. Для очистки поля добавим кнопку с названием X. В палитре свойств кнопки, в поле формул напишем: `ВыбПоставщик=0`. Теперь при нажатии этой кнопки поле будет очищаться, и можно осуществлять различные выборки по поставщикам. Теперь отредактируем шаблон печатной формы. Сейчас он выглядит, как на рис. 6.7: итоговые данные по каждому поставщику располагаются выше строк с товаром. Это не привычная для пользователя форма. Лучше разместить итоги по поставщикам после перечисления поставленных ими товаров. Для этого перейдем в модуль отчета и исправим процедуру формирования. Найдем в ней фрагмент программы:

Листинг 6.1

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

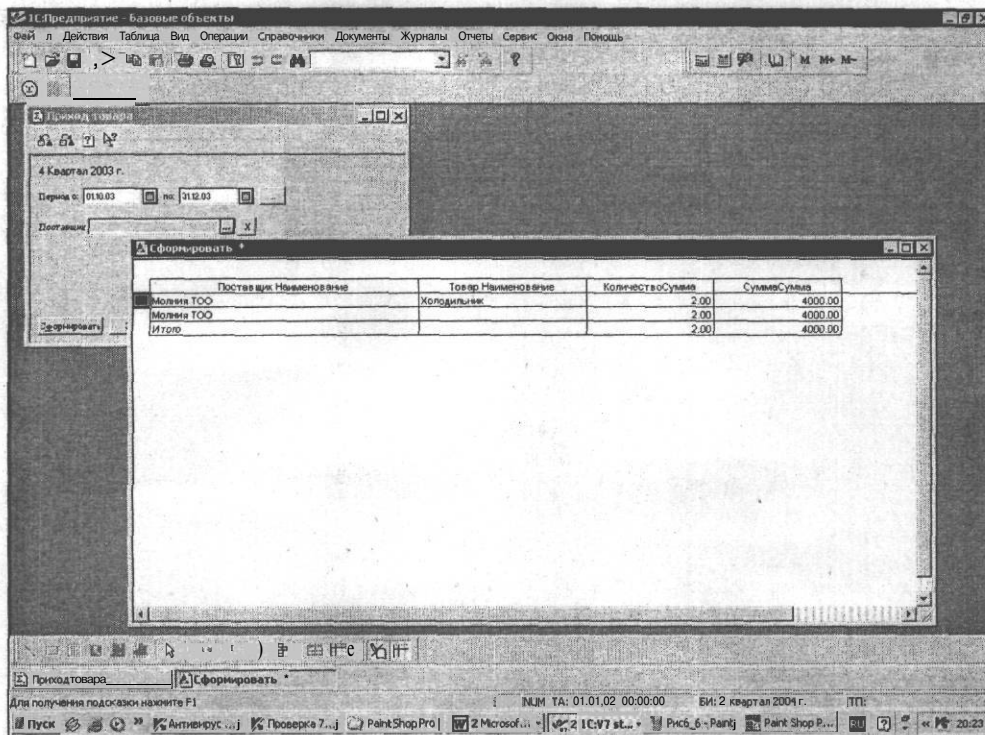


Рис. 6.7. Экранная и печатная форма отчета

И выведем секцию с итогами поставщика после цикла с выводом в таблицу товаров данного поставщика. В результате фрагмент будет таким:

Листинг 6.2

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

И печатная форма отчета будет в привычном виде (рис. 6.8).

Поставщик/Наименование	Товар/Наименование	Количество/Сумма	Сумма/Сумма
Азур-Альмас ООО	Телевизор "Samsung"	2.00	20000.00
Азур-Альмас ООО	Телевизор "Samsung"	2.00	20000.00
Молния ТОО	Холодильник	2.00	4000.00
Молния ТОО	Холодильник	2.00	4000.00
Поставщик ЧТ	Телевизор "Samsung"	5.00	100000.00
Поставщик ЧТ	Телевизор "Samsung"	5.00	100000.00
Итого		9.00	124000.00

Рис. 6.8. Печатная форма отчета

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

В листинге 6.3 приведена процедура сформировать () с подробными комментариями для объяснения механизма ее работы.

Листинг 6.3

```
//*****
//Процедура генерации запроса Сформировать.
//
Процедура Сформировать ( )
    Перец Запрос, ТекстЗапроса, Таб; //объявляем переменные для работы
    в процедуре
    //Создание объекта типа Запрос
    Запрос = СоздатьОбъект ("Запрос"); //Создаем объект Запрос
    ТекстЗапроса =
```

```
"//{{ЗАПРОС(Сформировать)
|Период с ВыбНачПериода по ВыбКонПериода; //указываем временной пери-
од, внутри которого строится отчет
|ОбрабатыватьДокументы все; //используются все документы
|Обрабатывать НеПомеченныеНаУдаление; //кроме помеченных на удаление
|Поставщик = Документ.ПрихНакл.Поставщик; //информация о Поставщике из
приходных накладных
|Товар = Документ.ПрихНакл.Товар; //как и информация о товаре
|Количество = Документ.ПрихНакл.Количество; //о количестве товара
|Сумма = Документ.ПрихНакл.Сумма; //и сумме, на которую по-
ставлен товар
|Функция КоличествоСумма = Сумма(Количество); //суммируем для подгото-
гов и итогов
|Функция СуммаСумма = Сумма(Сумма);
|Группировка Поставщик упорядочить по Поставщик.Наименование; //суммы
выводим по поставщикам
|Группировка Товар упорядочить по Товар.Наименование; //потом
по товарам
|Условие(Поставщик в ВыбПоставщик);
//условие
|"/}}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 "Заголовки"
Таб.ВывестиСекцию("Заголовки");
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 Товар
        Таб.ВывестиСекцию("Товар");
    КонецЦикла;
//Заполнение полей Поставщик
Таб.ВывестиСекцию("Поставщик");
```

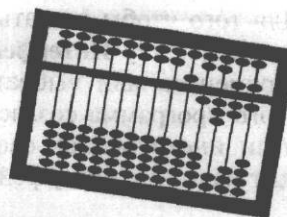
```
КонецЦикла;  
//Заполнение полей "Итого"  
Таб.ВывестиСекцию("Итого");  
//Вывод заполненной формы  
Таб.ТолькоПросмотр(1);  
Таб.Показать("Сформировать", "");  
КонецПроцедуры
```

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

Предупреждение

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

Глава 7



Пользовательский интерфейс и администрирование

На предприятии каждый сотрудник выполняет свой участок работы. Сотрудники обладают разной квалификацией. У них могут быть определенные ограничения на доступ к конфиденциальной информации, находящейся в базе "1С:Предприятие". Наконец, им должно быть удобно работать. Соответственно, определенные категории сотрудников могут иметь возможность работы с ограниченным количеством метаданных: документов, справочников и пр. 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 накладываться и из соображений удобства: конфигурации, автоматизирующие крупные предприятия, могут иметь десятки документов, справочников и отчетов. Меню вызова их будет громоздкое и неудобное. В "1С:Предприятие" решение перечисленных задач производится с помощью развитых средств построения пользовательского интерфейса, который включает в себя меню и панели инструментов, наборы пользовательских прав и наборы пользователей.

7.1. 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 создается 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 конфигурации выбрать вкладку **Интерфейсы** и нажать на кнопку **Новый элемент** панели инструментов или выбрать пункт меню **Действия | Новый интерфейс**. После этого программа создаст новый объект с именем Новый и откроет окно панели инструментов свойств пользовательского интерфейса, которое содержит название интерфейса и два флажка (рис. 7.1).

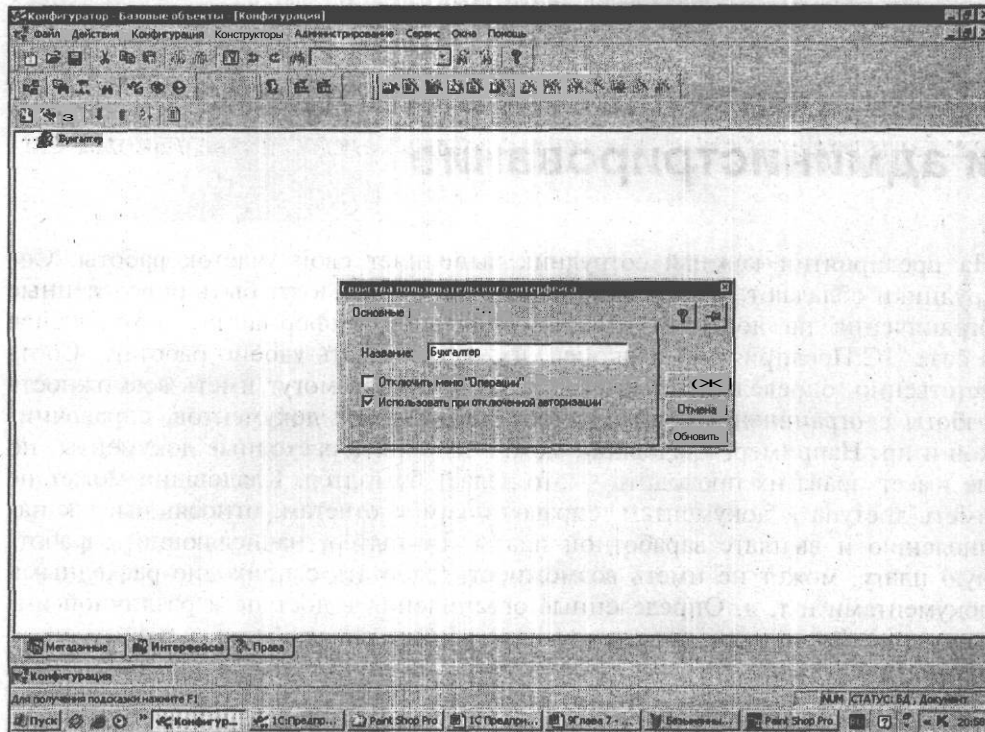


Рис. 7.1. Палитра свойств пользовательского интерфейса

Название интерфейса — это произвольная строка текста. Желательно, чтобы оно было осмысленным и понятным пользователю.

Флаг **Отключить меню "Операции"** устанавливается в тех случаях, когда пользователю доступен для использования только ограниченный набор объектов.

Предупреждение

При создании интерфейсов конфигураций важно помнить, чтобы среди них был хотя бы один интерфейс, для которого флаг **Использовать при отключенной авторизации** включен/Это даст возможность корректно начать работать в новой информационной базе до подключения остальных пользователей.

Когда вы создаете новый интерфейс, в нем нет ни меню, ни панелей инструментов. Поэтому интерфейс еще предстоит создать, вызывая *редактор меню* и *редактор панелей инструментов*.

Для вызова редактора меню необходимо на имени соответствующего интерфейса из списка нажать правую кнопку мыши и выбрать пункт выпадающего меню **Редактировать меню...**, после чего откроется окно палитры свойств меню (рис. 7.2). Меню в редакторе — в виде дерева. В самом начале разработаем меню первого уровня, которое будет выводиться в строке наряду с уже имеющимися по умолчанию.

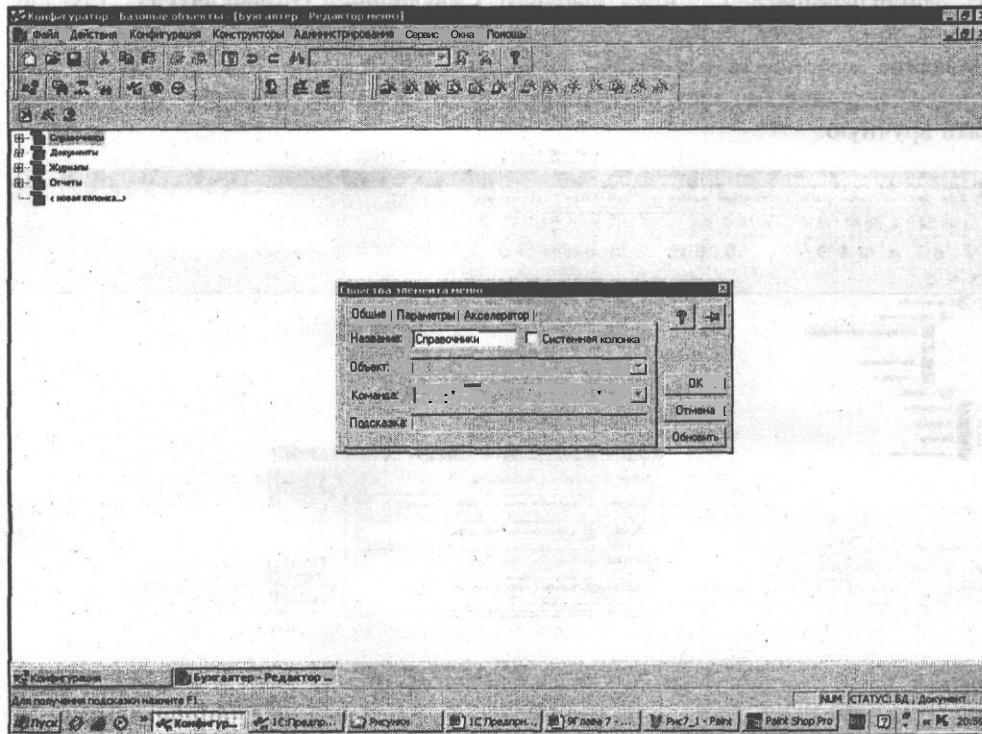


Рис. 7.2. Палитра свойств меню

Для меню первого уровня можно указать только название. Все остальные поля палитры свойств будут недоступны. Если в ней 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.

Создадим элементы меню первого уровня:

- справочники;
- документы;
- журналы;
- отчеты.

Щелкнем на пиктограмме плюса слева от ветви **Справочники**. Откроется ветвь этого элемента **<новый.>**. Щелкнем дважды по ней и заполним палитру свойств нового элемента выпадающего меню второго уровня (рис. 7.3). Тип так и останется по умолчанию — **Команда**, а в поле **Объект** из раскрывающегося списка выберем **Справочники.Номенклатура**. Так как справочник имеет синоним, этот синоним автоматически попадает в поле **Название**, которое пользователь и увидит в своем меню. Если синонима нет, в поле **Название** попадет идентификатор, и название придется отредактировать вручную.

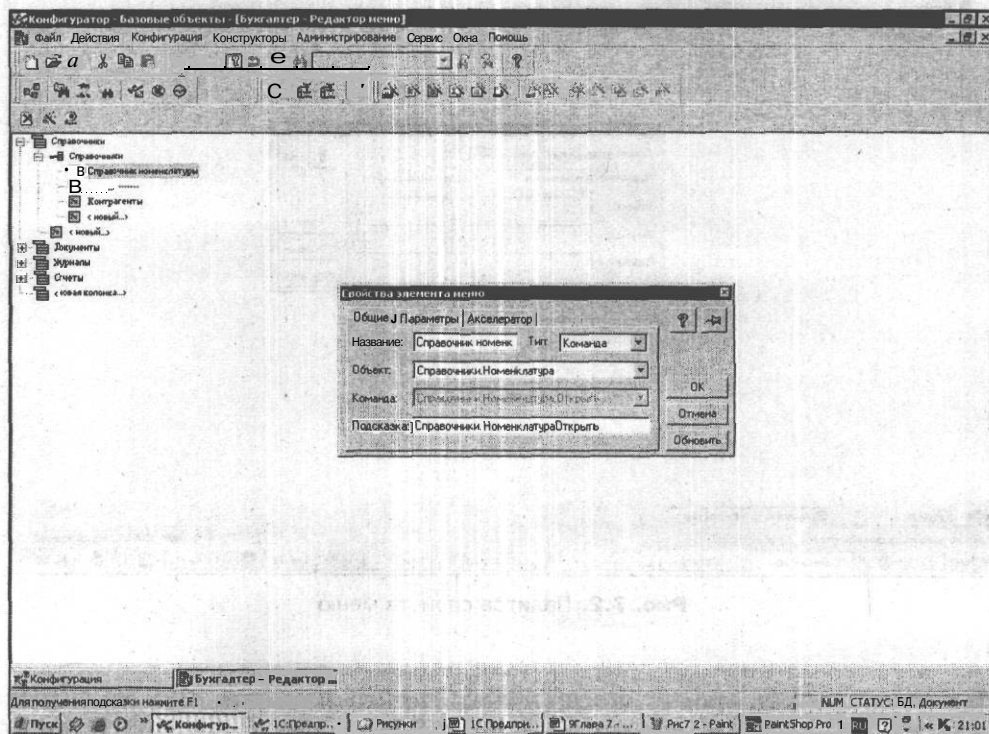


Рис. 7.3. Палитра свойств меню второго уровня

Добавьте в меню вызов всех справочников, и можно переходить к документам. Формирование выпадающего меню почти не отличается. При выборе

документа активизируется поле **Команда**, содержащее список из двух команд: **Документы... Ввести** и **Документы... ВвестиНаОсновании**. В нашем примере выбираем команду **Документы... Ввести**, так как в конфигурации нет документов, вводимых на основании других документов.

Меню **Журналы и Отчеты** создаются аналогично меню **Справочники**.

Теперь осталось проверить, как будет работать наше меню. Для этого нужно щелкнуть мышью на кнопке **Проверить меню** панели инструментов редактора меню или войти в меню **конфигуратора Действия | Тест**. Тогда редактор меню откроет **Тестовое окно** (рис. 7.4). В этом окне можно просматривать внешний вид меню, но нельзя его использовать.

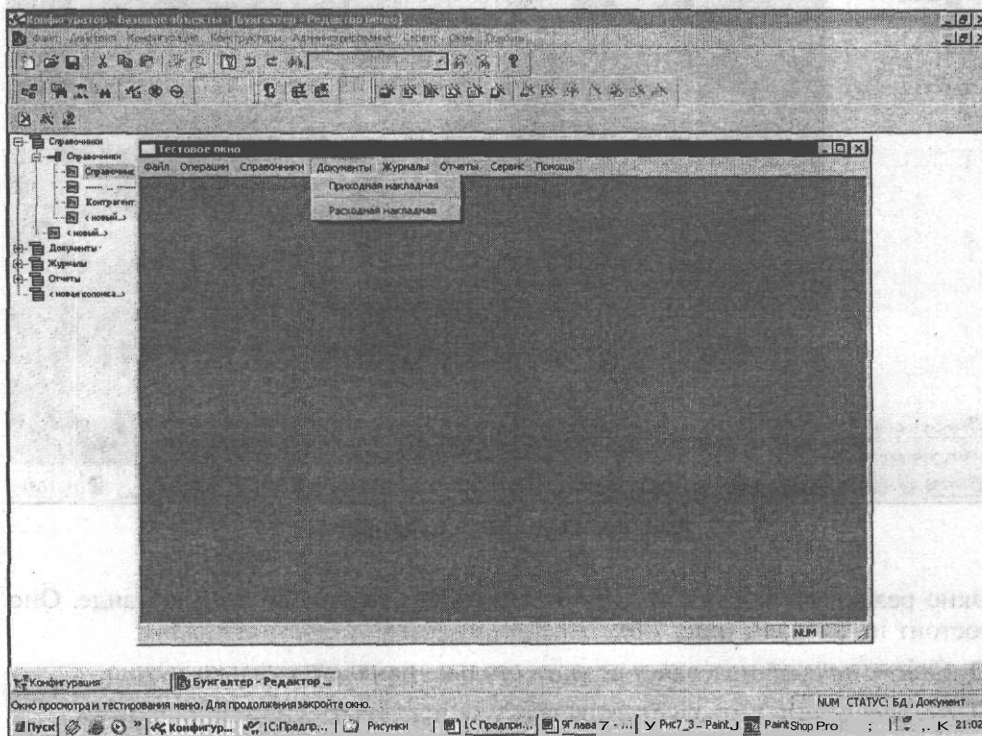


Рис. 7.4. Тестирование вновь созданного меню

В поле палитры свойств элемента меню **Тип**, кроме типа **Команда**, можно выбрать значения **Разделитель** и **Подменю**. Они служат для оптимизации внешнего вида меню и удобства в работе с ним. Пример использования приведен на рис. 7.5.

Для запуска редактора панели инструментов нужно выбрать вкладку **Интерфейсы** окна конфигурации. После чего, предварительно выбрав нужный ин-

терфейс, можно щелкнуть на нем правой кнопкой мыши и в открывшемся меню выбрать пункт **Редактировать панель инструментов**.

Подменю и разделитель

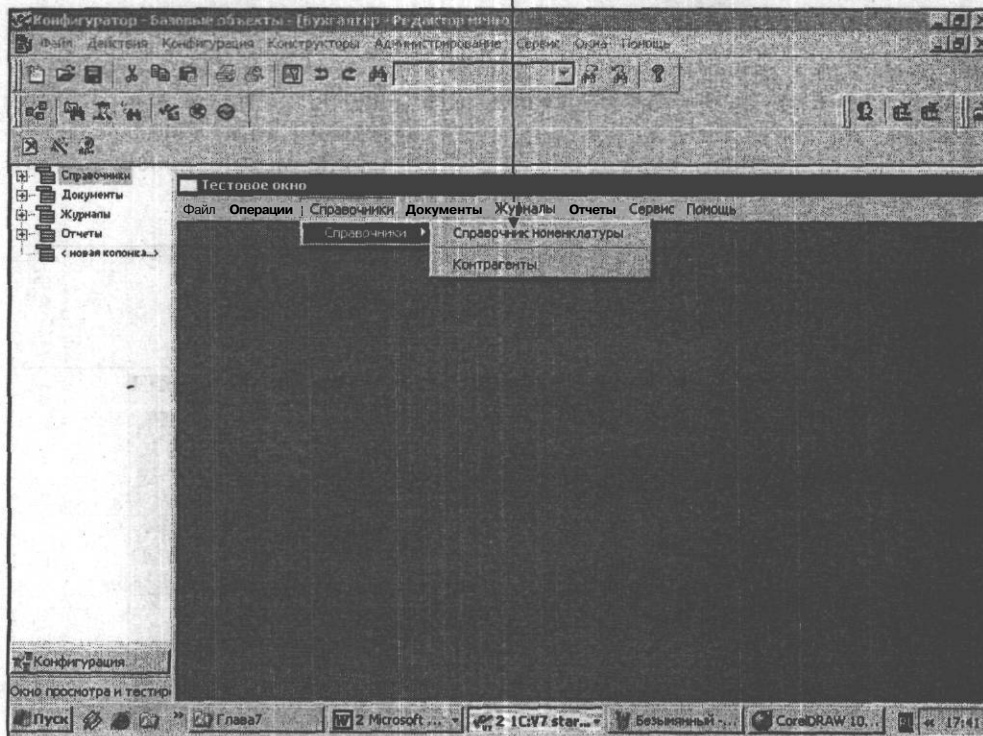


Рис. 7.5. Подменю и разделитель

Окно редактора панелей инструментов будет открыто по этой команде. Оно состоит из трех зон (рис. 7.6):

- 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**. В окне списка панелей нажмем кнопку **Добавить** либо зайдём в меню **Действия | Создать панель**. В появившейся палитре свойств заполним название панели: **Справочники**. Группу полей **По умолчанию** оставим без изменения. Нажмем кнопку **ОК**, и в списке панелей появится панель **Справочники**. Следующий шаг — сформировать состав панели. Кнопкой **Добавить** в окне состава панели либо через меню **Действия | Добавить кнопку** открываем палитру свойств новой

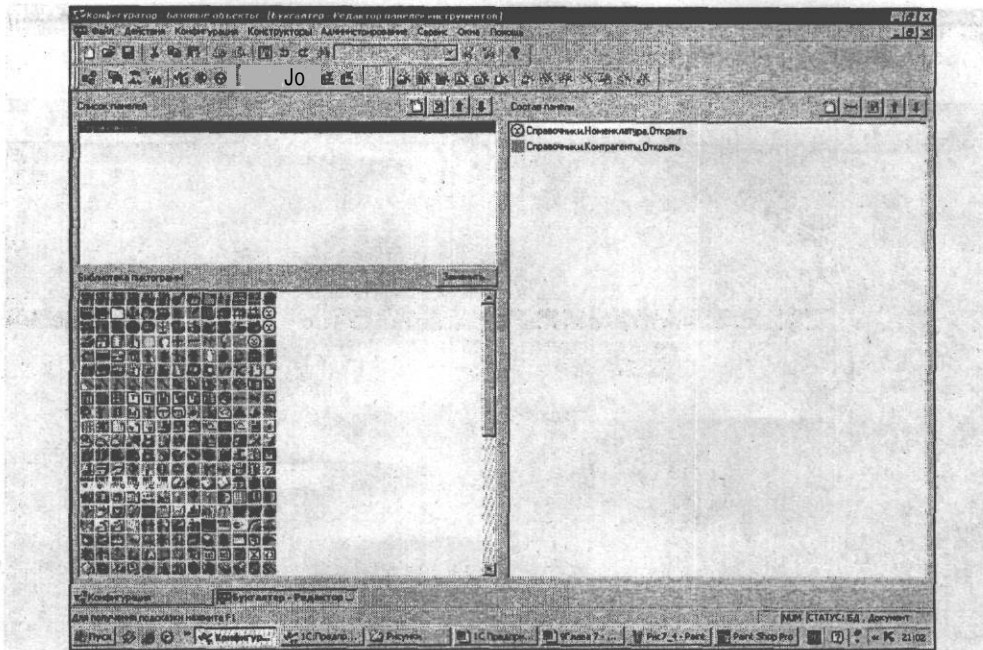


Рис. 7.6. Окно редактора панелей инструментов

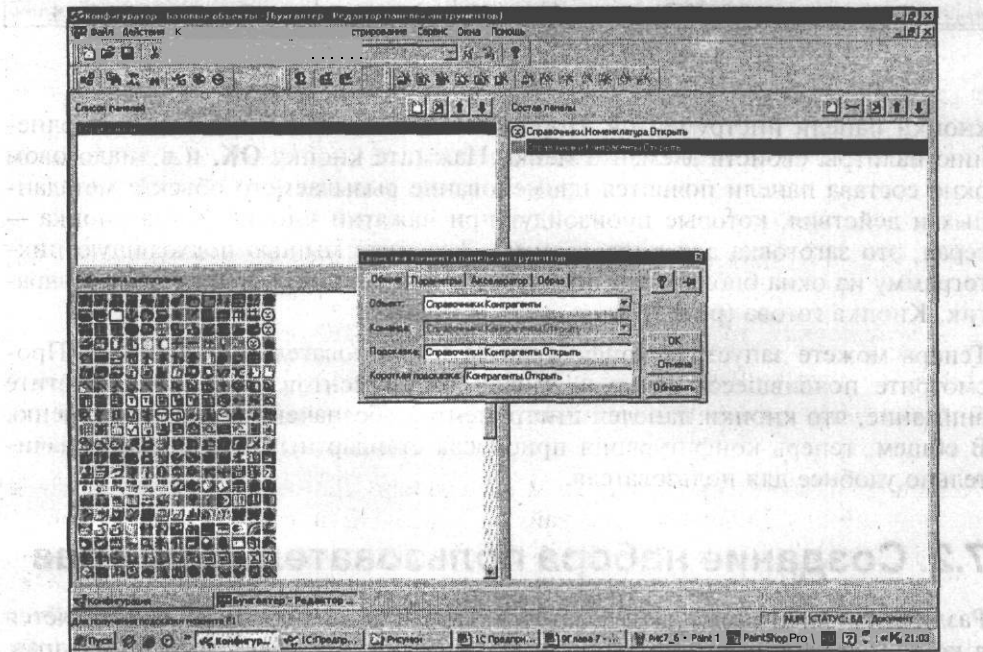


Рис. 7.7. Заполнение панели инструментов

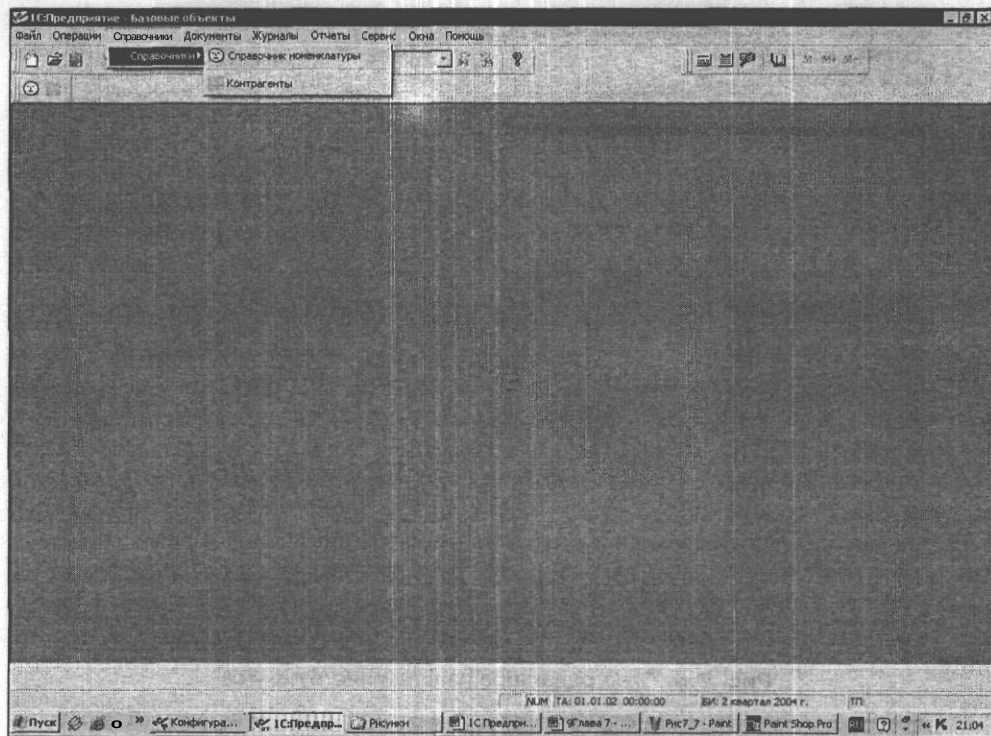


Рис. 7.8. Меню и панель инструментов

кнопки панели инструментов. Порядок ее заполнения аналогичен заполнению палитры свойств элемента меню. Нажмите кнопку ОК, и в диалоговом окне состава панели появится наименование вызываемого объекта метаданных и действия, которые произойдут при нажатии кнопки. Сама кнопка — серая, это заготовка для пиктограммы. Захватите мышью подходящую пиктограмму из окна библиотеки пиктограмм и перетащите ее на серый квадратик. Кнопка готова (рис. 7.7).

Теперь можете запустить конфигурацию в пользовательском режиме. Просмотрите появившееся меню и панели инструментов (рис. 7.8). Обратите внимание, что кнопки панелей инструментов обозначены и в пунктах меню. В общем, теперь конфигурация приобрела стандартный вид и стала значительно удобнее для пользователя.

7.2. Создание набора пользовательских прав

Различным категориям пользователей доступ к информации определяется в конфигураторе с помощью формирования набора пользовательских прав. Свойства элемента прав модифицируются во вкладке **Права** окна конфигу-

рации. Окно редактора похоже на окно **Метаданные**, но содержит только те объекты метаданных, для которых могут быть назначены права. Пример — документ. Нет ветвей для реквизитов шапки документа и табличной части документа, так как права могут быть назначены только целиком на документ (рис. 7.9).

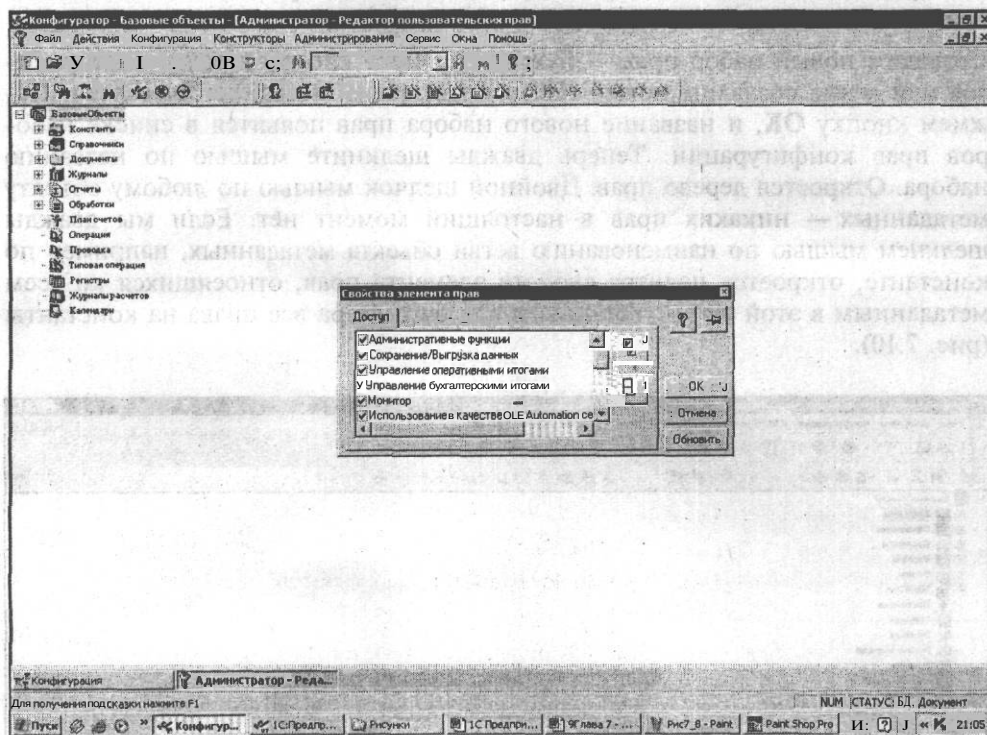


Рис. 7.9. Дерево прав конфигурации

В пустой базе создается один набор прав — Администратор, со всеми правами на виды объектов метаданных, т. е. полный доступ к информации.

Создавать и редактировать наборы прав можно так же, как и другие составляющие интерфейса — через кнопки на панели инструментов, двойной щелчок мыши или меню **Действия**.

Для того чтобы назначить права доступа к объекту метаданных, надо дважды щелкнуть мышью на идентификаторе метаданных. Диалог свойств 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 установить флажок напротив операции. Разрешение или запрещение той или иной операции может автоматически запрещать или

разрешать другие логически связанные операции. Например, включение флажка **Корректировка** автоматически включает и флажок **Чтение**. А отключение прав на чтение автоматически отключает все остальные права, ведь если нет прав на чтение информации, изменять ее как-либо уже невозможно. Кнопка с двумя установленными флажками позволяет установить сразу все права на объект. Кнопка с неустановленными флажками снимает все права с объекта метаданных.

Создадим новый набор прав — Бухгалтер. Через кнопку панели инструментов или меню создадим новый набор прав, дадим название: Бухгалтер. Нажмем кнопку ОК, и название нового набора прав появится в списке наборов прав конфигурации. Теперь дважды щелкните мышью по названию набора. Откроется дерево прав. Двойной щелчок мышью по любому объекту метаданных — никаких прав в настоящий момент нет. Если мы дважды щелкнем мышью по наименованию ветви объекта метаданных, например по константе, откроется палитра свойств элемента прав, относящихся ко всем метаданным в этой ветви. Установим для бухгалтера все права на константы (рис. 7.10).

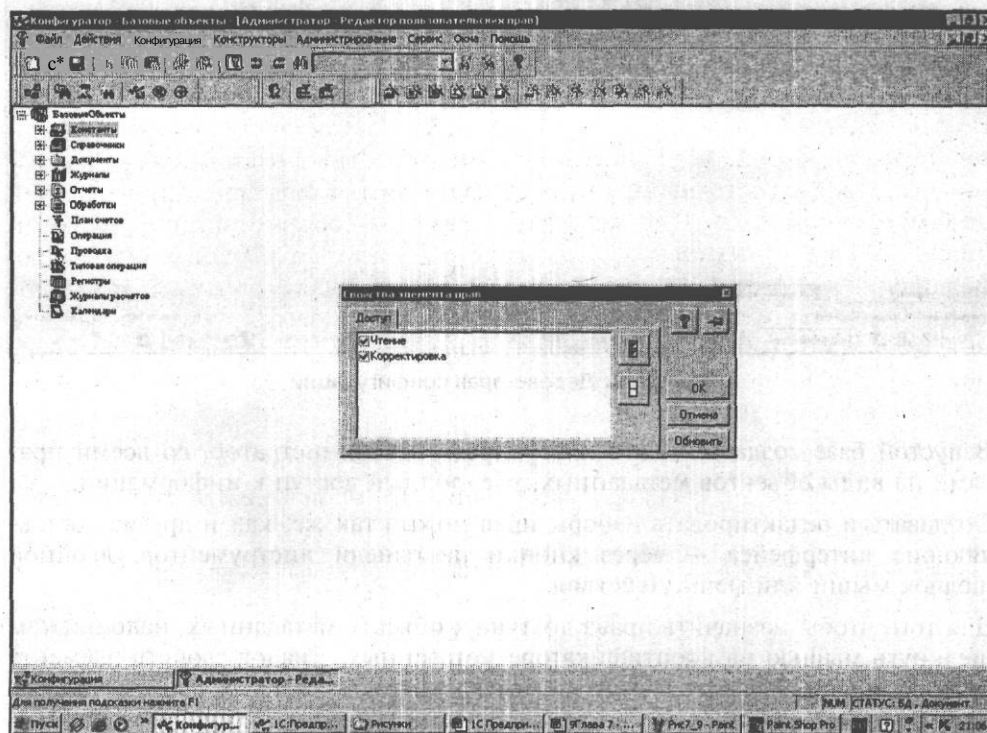


Рис. 7.10. Установка прав на константы

Проверьте, установлены ли теперь все права на каждую **константу**.

Теперь дадим права только на чтение справочника Номенклатура. Для этого зайдём в ветвь метаданных **Справочники**, затем непосредственно в справочник Номенклатура, дважды щелкнем мышью и установим флажок напротив права **Чтение**. На справочник Контрагенты установим все права: на все документы, журналы документов и отчеты.

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 основной.

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

7.3. Создание нового пользователя

Новый пользователь может быть создан в меню **Администрирование** | **Пользователи** или кнопками панели инструментов. Имя **пользователя** — произвольная строка, состоящая из букв, цифр и знаков подчеркивания, длиной не более 24 символов. Имя желательно давать в соответствии с функциями (рис. 7.11). Если несколько пользователей имеют одинаковую функцию, можно копировать имя в буфер обмена, меняя только окончание для соблюдения уникальности. У вновь созданного пользователя палитру свойств можно не заполнять сразу, а **позднее**, выделив этого пользователя. Палитру можно вызвать из выпадающего меню пунктом **Свойства**. Атрибуты, характеризующие нового пользователя:

- имя — для задания и редактирования имени пользователя;
- полное имя — для расшифровки имени. Это произвольная строка, которая может содержать пробелы;
- рабочий каталог — каталог, 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, либо заново ввести пользователей;
- роль — для выбора наименования пользовательского интерфейса и набора прав, которые будут поставлены в соответствие имени пользователя. В поле **Права** необходимо указать название набора прав пользователей.

В поле **Интерфейс** указывается название индивидуального пользовательского интерфейса. Опция **Отключить контроль прав** отменяет для указанного пользователя контроль доступа к информации. Это все равно, как если бы пользователю были предоставлены все права (рис. 7.12).

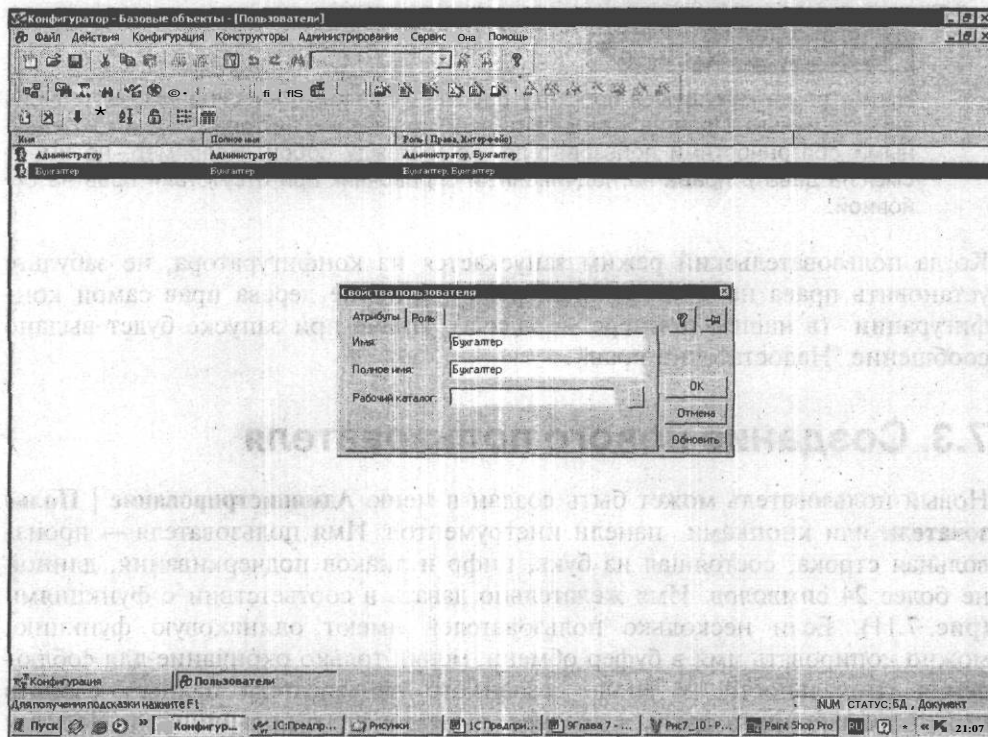


Рис. 7.11. Создание нового пользователя

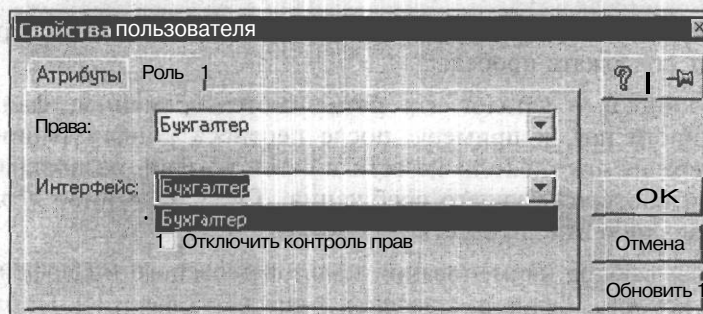


Рис. 7.12. Назначение прав и интерфейса пользователю

С целью предотвращения входа пользователей в систему под чужими именами, что влечет за собой и возможность использования чужих наборов прав и пользовательских интерфейсов, каждому пользователю может быть установлен пароль для входа. Для установки пароля выделите в списке имя пользователя 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, допускаются буквы, цифры, знаки подчеркивания и др., кроме пробелов и специальных символов. Для надежности система запросит ввод пароля еще раз. Если вы ошибетесь, система выдаст соответствующее сообщение.

Теперь можно запустить конфигурацию в пользовательском режиме и проверить наличие или отсутствие соответствующих прав пользователя (рис. 7.13).

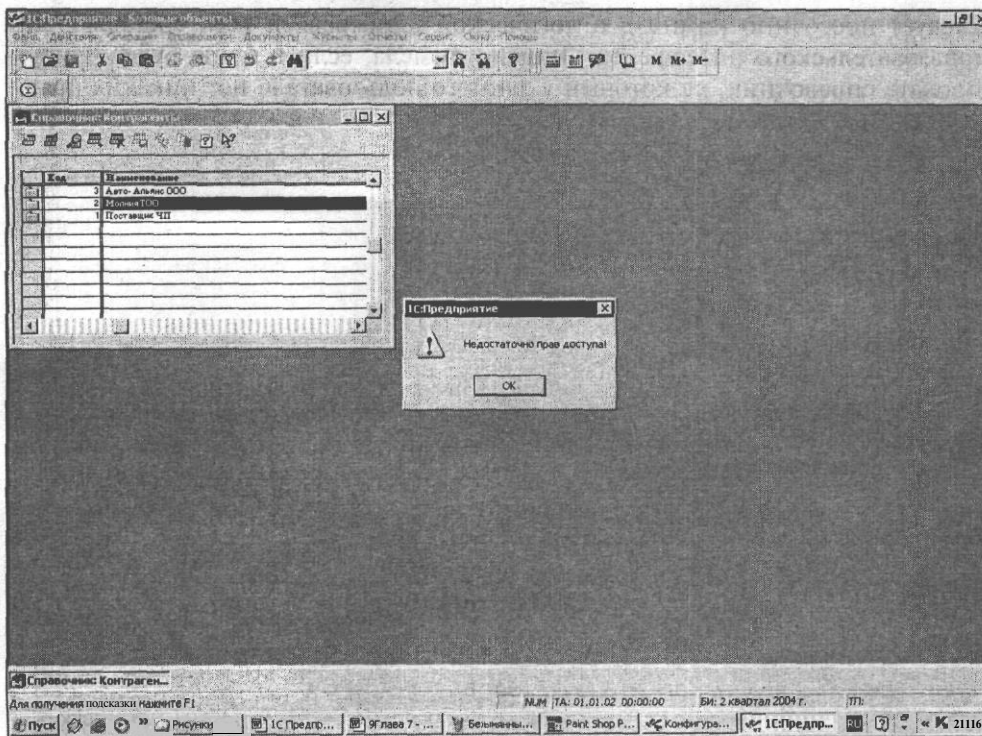


Рис. 7.13. Проверка прав пользователя

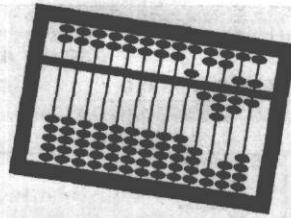
7.4. Заключительные замечания

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

Можно создать несколько типовых пользовательских интерфейсов с различным набором 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 будет сопоставлен 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. Лишено смысла, если в меню будет присутствовать справочник, на который у данного пользователя нет никаких прав.

Глава 8



Некоторые дополнительные сведения

В этой главе собраны разнообразные сведения, которые помогут, не выходя за рамки знаний, полученных ранее, улучшить конфигурацию и процесс сохранения и обработки данных.

8.1. Специальные свойства объекта конфигурации

Теперь можно изменить свойства самой конфигурации. Для этого нужно войти в палитру свойств корня дерева метаданных Задача. Задайте значение идентификатора и синонима, как в учебном примере. Обратите внимание на две вкладки: **Автор** и **Заставка**, которые позволяют ввести информацию об авторе конфигурации и свою заставку, которая будет появляться вслед за системной заставкой (рис. 8.1).

8.2. Сохранение базы данных

Один из главных факторов стабильной работы в "1С:Предприятие" версии 7.7 — регулярное сохранение информационной базы данных. Разрушение базы может происходить по разным причинам: поломки компьютера, вирусы, внезапное отключение питания, сбой в сети, вообще непонятные причины. Разрушенную базу бывает очень трудно восстановить, а часто и вообще невозможно. И 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.

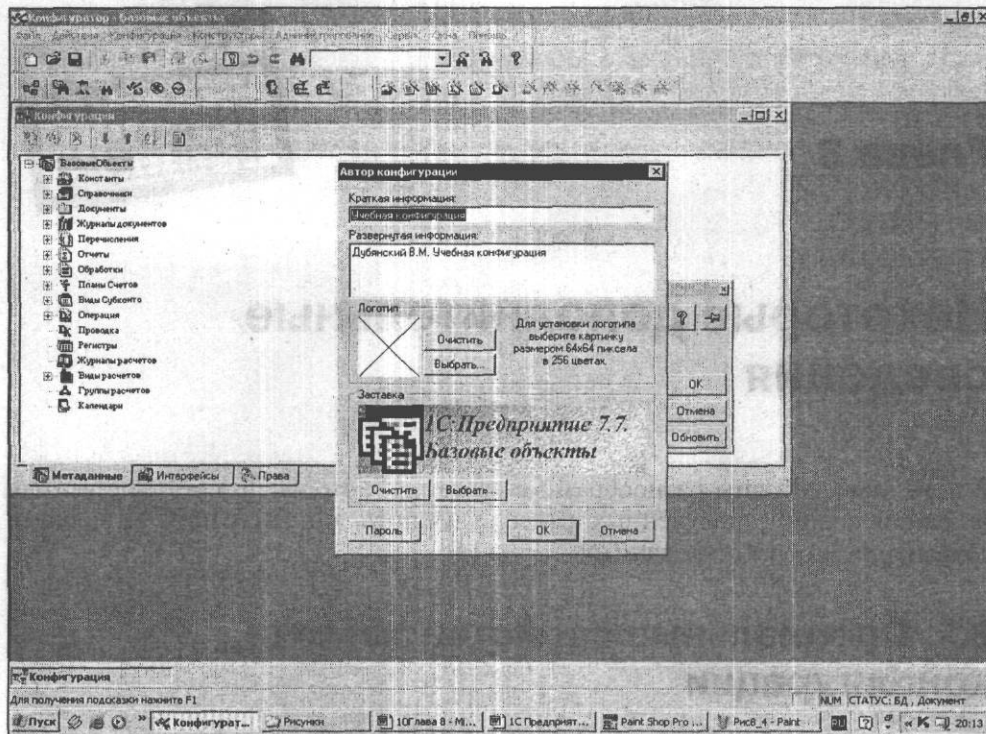


Рис. 8.1. Палитра свойств конфигурации

Желательно иметь несколько архивных копий. В самом простом случае база должна ежедневно копироваться на жесткий диск компьютера (или сервера) и на независимый носитель — дискеты, съемные диски, компакт-диски и пр. Неверная загрузка измененной конфигурации также 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.

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**. Откроется список опций, в том числе и четыре пункта, относящиеся к сохранению данных,

которые условно можно поделить на два режима. Режим **Сохранить/Восстановить (Сохранить данные, Восстановить данные)** и режим **Выгрузить/Загрузить (Выгрузить данные, Загрузить данные)** (рис. 8.2).

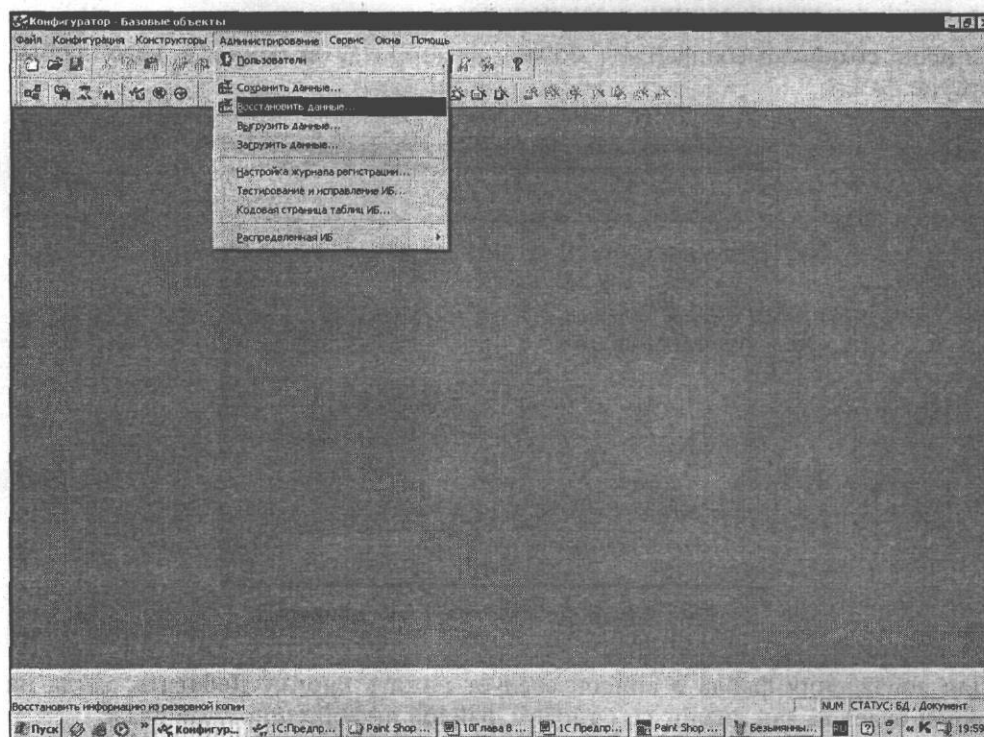


Рис. 8.2. Опции меню для сохранения или восстановления данных

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

С **Замечание**

Режим Сохранение/Восстановление обеспечивает сохранение данных только в случаях формата DBF. Режим Выгрузка/Загрузка работает в формате DBF и MS SQL Server.

При выборе пункта меню **Сохранить данные** откроется диалог **Сохранение данных**, который содержит список сохраняемых файлов и кнопки управления этим списком (рис. 8.3). Изначально этот список содержит файлы, жизненно важные для функционирования системы "1С:Предприятие":

О *.DBF — файлы данных;

- 1CV7.MD — описание конфигурации задачи;
- 1CV7.DD — описание структуры файлов данных;
- 1CV7.SPL — словарь, который используется для выдачи чисел, наименований денежных единиц и дат прописью.

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

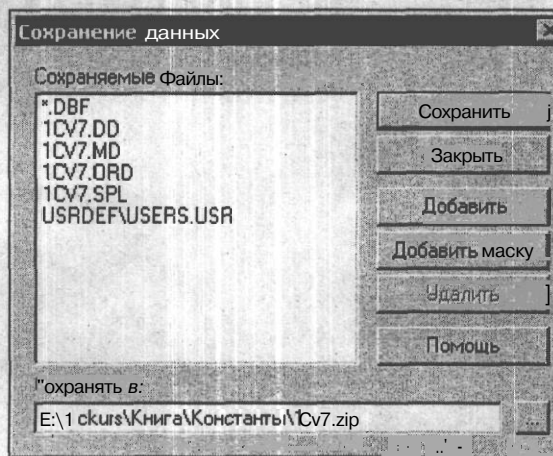


Рис. 8.3. Диалог Сохранение данных

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 **Сохранить в**. При нажатии кнопки справа 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,

профамма 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. 8.4).

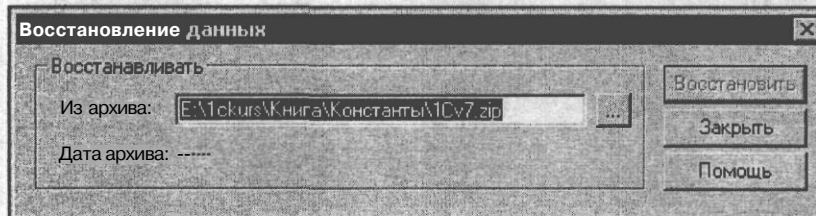


Рис. 8.4. Диалоговое окно **Восстановление данных**

Если в открывшемся диалоге щелкнуть мышью по кнопке, находящейся 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**.

Если происходит восстановление из резервной копии на дискетах, следует по запросам программы вставлять дискеты, содержащие части копии, в дискетод. Если дискеты вставляются в неверном порядке, профамма выдаст соответствующее сообщение. При восстановлении информации из резервной копии на жестком диске или компакт-диске профамма 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 сохраненных данных производится при вызове меню **Администрирование** | **Загрузить данные**. В диалоговом окне (рис. 8.5) необходимо указать путь к архивному файлу и нажать кнопку **ОК**. После этого появится сооб-

шение: "При загрузке данных все существующие данные будут уничтожены. Продолжить выполнение операции?". Если вы твердо решили загружать архивную копию, отвечаете Да. Если есть сомнения, ответьте Нет и все снова проверьте.

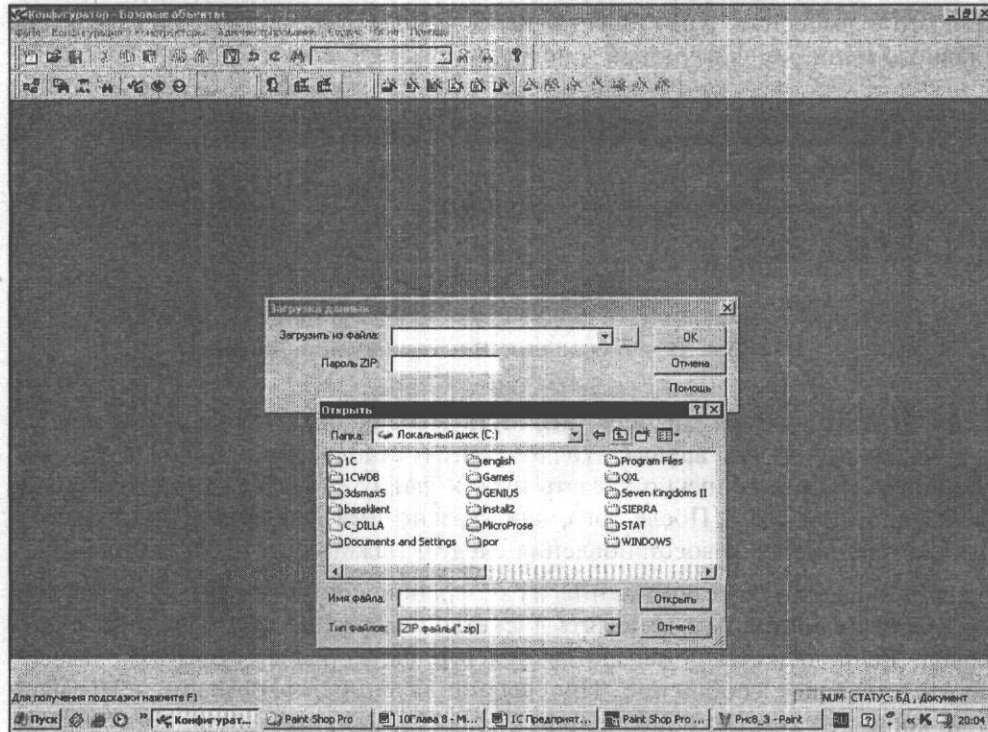


Рис. 8.5. Диалоговое окно **Загрузка данных**

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 **Выгрузка/Загрузка** является более надежным, а режим **Сохранение/Восстановление** — 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).

8.3. Перенос справочников с данными

Необходимость переноса справочников возникает достаточно часто. Например, надо создать пустую конфигурацию (копию рабочей), но перенести в нее из рабочей информационной базы справочники с данными. Действительно, ввод остатков в пустую информационную базу — часто возникающая задача и достаточно длительная работа. Особенно если необходима набивка справочника номенклатуры из 40 000 наименований. В данном разделе описан порядок работы по переносу справочника с данными для двух вариантов:

- справочник с данными переносится в пустую идентичную конфигурацию;
- добавляется как совершенно новый в произвольную конфигурацию.

8.3.1. Добавление нового справочника

Из условной конфигурации Бухгалтерский учет в нашу учебную конфигурацию перенесем справочник Коды бюджетной классификации. В учебной конфигурации этого справочника нет.

Вы, конечно же, можете использовать любой справочник из любой конфигурации.

Шаг 1. В режиме конфигуратора откройте учебную конфигурацию. Выберите в меню **Конфигурация** пункт **Объединение конфигураций** (рис. 8.6).

В стандартном диалоговом окне Windows **Открыть файл конфигурации** найдите папку с конфигурацией, в которой находится справочник, — в данном случае с конфигурацией Бухгалтерский учет. Выберите файл с расширением `md` и нажмите кнопку **Открыть** (рис. 8.7). Файл с расширением `md` содержит только описание конфигурации в особом формате: структуру метаданных, набор прав, пользовательские интерфейсы, но данные в этом файле не хранятся.

На экране появится диалоговое окно **Объединение конфигураций** (рис. 8.8).

Шаг 2. Уберите флажки со всех объектов в колонке **Объект**. Для этого щелкните левой кнопкой мыши по всем флажкам в квадратах слева от наименования объектов.

Шаг 3. Щелкните мышью по знаку **+** слева от ветви метаданных **Справочники**. У вас появится список справочников конфигурации Бухгалтерский учет.

Шаг 4. Найдите интересующий вас справочник (в нашем примере — Коды бюджетной классификации) и, щелкнув мышью по квадратику напротив него, установите флажок.

Шаг 5. Убедитесь, что флажок только напротив интересующего нас справочника, и нажмите кнопку **ОК**. Произойдет добавление справочника к учебной конфигурации, о чем система выдаст сообщение (рис. 8.9).

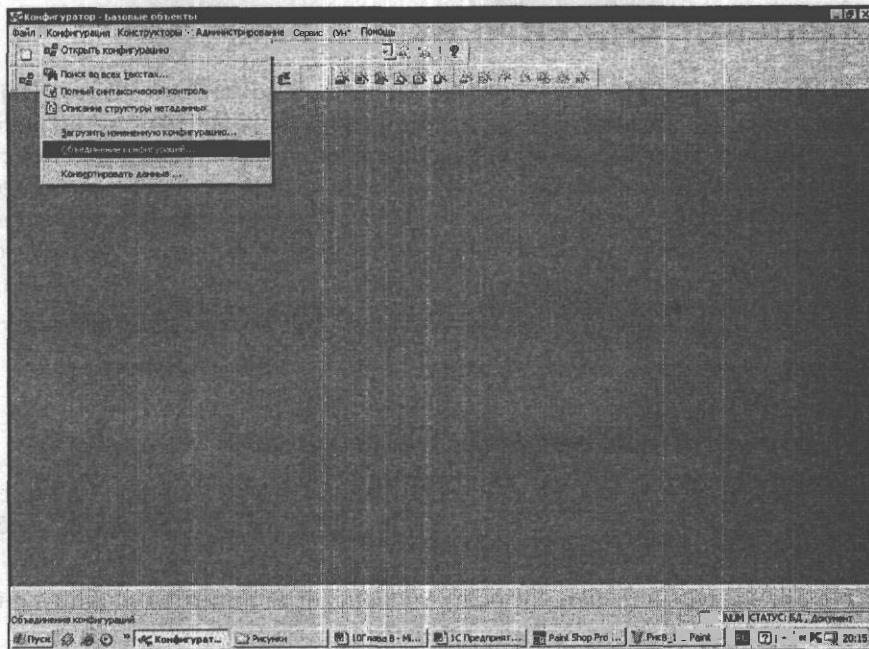


Рис. 8.6. Выбор меню **Объединение конфигураций**

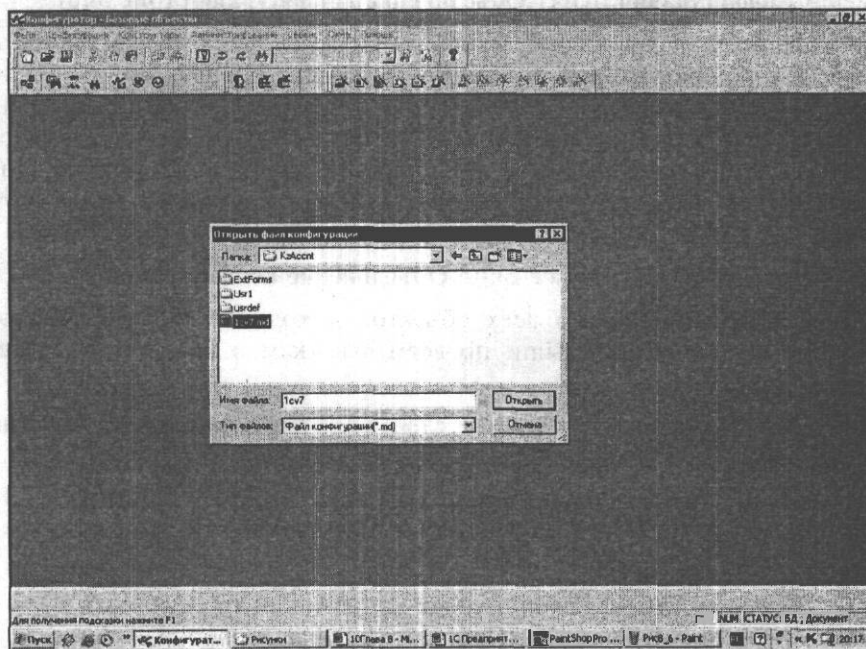


Рис. 8.7. Выбор файла с расширением mtd в диалоговом окне

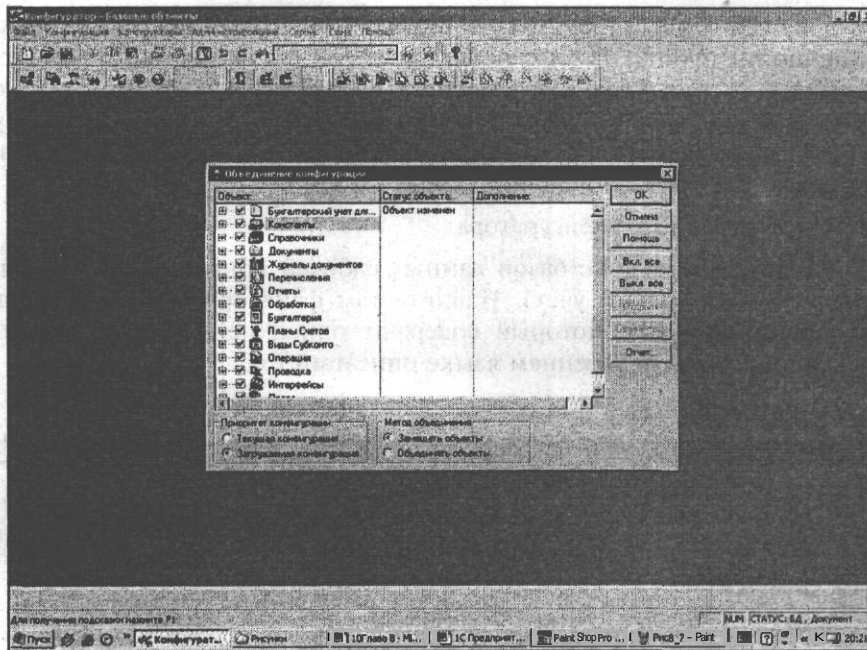


Рис. 8.8. Диалог **Объединение конфигураций** с развернутым списком справочников

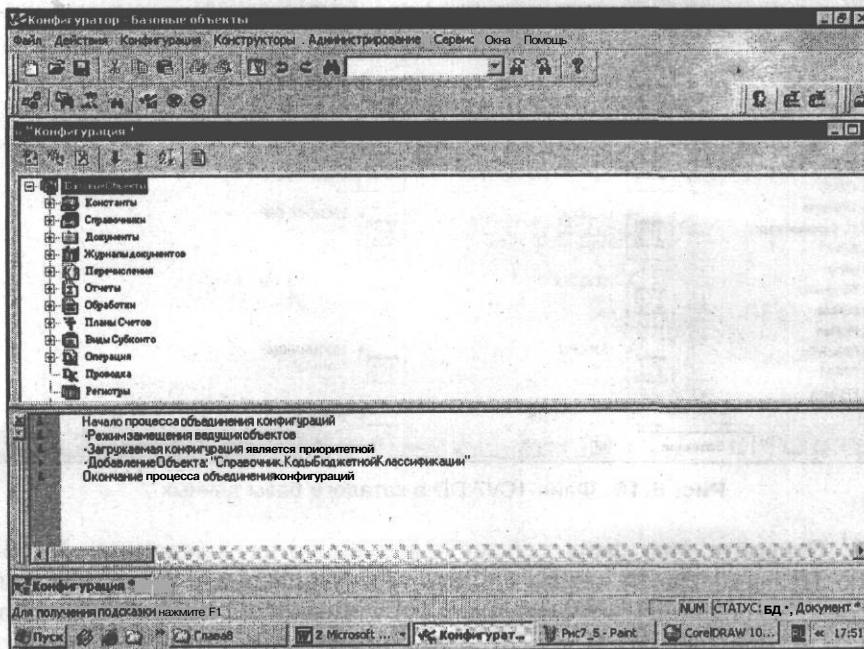


Рис. 8.9. Новый справочник добавлен в учебную конфигурацию

Теперь в нашей конфигурации есть справочник Коды бюджетной классификации, но он пустой. В этом нетрудно убедиться, если мы откроем этот справочник в пользовательском режиме. Справочник может иметь процедуры и функции в модулях формы и элемента, которые позднее нужно будет привести в соответствие с программой. А пока их лучше закомментировать, как это сделано в учебной конфигурации.

Шаг 6. Закройте окно конфигуратора.

Шаг 7. Зайдите в каталог с базой данных рабочей конфигурации (в нашем примере — Бухгалтерский учет). Найдите там файл 1CV7.DD — так называемый словарь данных, который содержит описание структуры хранения базы на специальном внутреннем языке описаний (рис. 8.10).

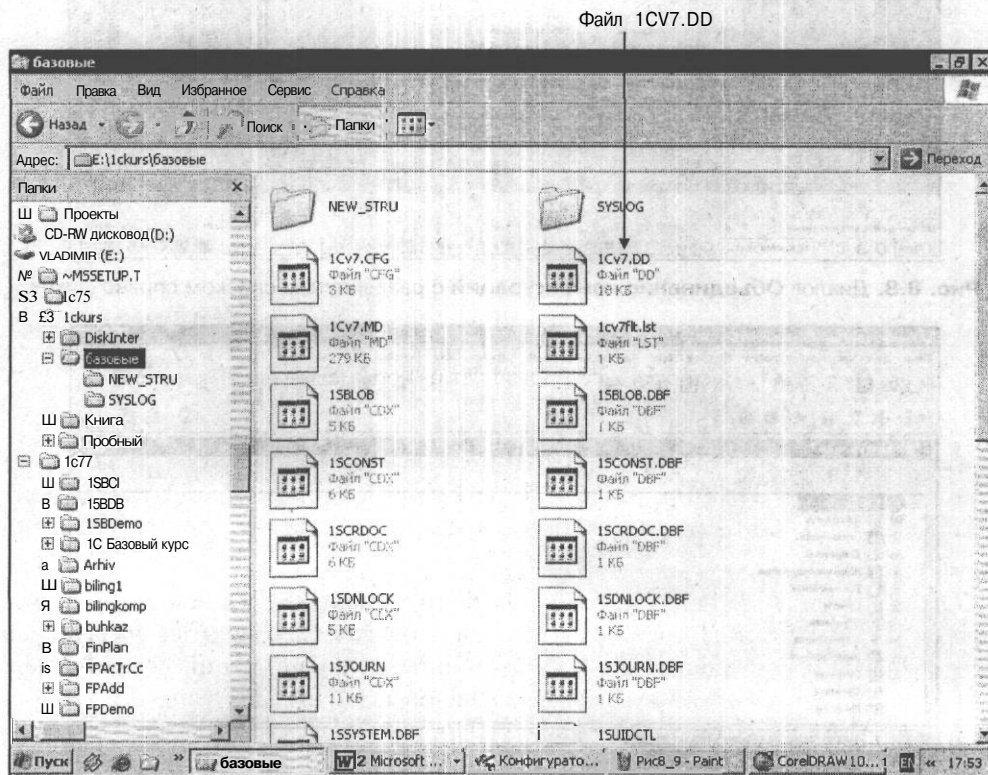


Рис. 8.10. Файл 1CV7.DD в каталоге базы данных

Шаг 8. Откройте этот файл с помощью стандартного редактора WordPad. В меню **Правка** | **Найти** перейдите к интересующему нас разделу Справочник Коды Назначения Платежей (рис. 8.11) и запомните (запишите) название файла, в котором хранятся записи этого справочника: SC4679 (у вас он может быть другой).

```

1-IDD      |of ID      |0      |ID      |
I-CODE     |of CODE    |0      |CODE (UPPER) |
I-DESCR    |of DESCR   |0      |DESCR (UPPER) |
#
#-----
#==TABLE no 25 : Справочник КодыБюджетнойКлассификации
# Name      |Descr      |Type[A/S/U] |DBTableName|ReUsable
T=SC6794    |Справочник КодыБюджетнойКлассификации|A          |SC6794      |1
#-----
# Fields
# Name      |Descr      |Type|Length|Precision
F-ID        |ID object  |C   |19    |10
F-CODE      |object code|C   |16    |10
F-DESCR     |object description |C   |1100  |10
F-ISMARK    |Flag Object is Marked |C   |1     |10
F-VERSTAMP  |Version stamp |C   |16    |10
#-----
# Indexes
# Name      |Descr      |Unique|Indexed fields
I-ID        |of ID      |0     |ID
I-CODE      |of CODE    |0     |CODE (UPPER)
I-DESCR     |of DESCR   |0     |DESCR (UPPER)
#-----
#==TABLE no 26 : Справочник КодыНазначенияПлатежей
# Name      |Descr      |Type[A/S/U] |DBTableName|ReUsable
T=SC6796    |Справочник КодыНазначенияПлатежей|A          |SC6796      |1
#-----
# Fields
# Name      |Descr      |Type|Length|Precision
F-ID        |ID object  |C   |19    |10
F-CODE      |object code|C   |13    |10

```

Рис. 8.11. Информация о справочнике КодыНазначенияПлатежей

Шаг 9. Перейдите в каталог с учебной базой и откройте файл 1CV7.DD, записанный в ней. Найдите раздел, описывающий справочник КодыБюджетнойКлассификации, и введите имя файла, предназначенного для хранения данных SC104.

Шаг 10. Войдите в папку с базой учебной конфигурации и переименуйте файл SC104.dbf таким образом, чтобы он подсказывал прежнее название и не совпадал ни с одним возможным файлом конфигурации — например, SC104a. Обратите внимание, мы переименовываем только файл с расширением dbf, именно в файлах с таким расширением хранятся записи справочника.

Шаг 11. Скопируйте из папки с базой данных рабочей конфигурации файл справочника SC4679.dbf или соответствующий ему в вашей конфигурации в папку с базой данных учебной конфигурации.

Шаг 12. Переименуйте скопированный файл SC4679.dbf в файл SC104.dbf, т. е. замените пустой файл данных файлом с записями.

Шаг 13. Удалите файл SC104a.dbf или соответствующий ему в конфигурации.

Шаг 14. Удалите в папке с базой данных учебной конфигурации все файлы с расширением `cdx`.

Шаг 15. Запустите учебную конфигурацию в пользовательском режиме. После переиндексации вы увидите уже заполненный справочник с данными (рис. 8.12).

Код	Наименование
10101	Подоходный налог с юр.лиц-резидентов
10102	Подоходный НАЛОГ с юр.лиц-нерезидентов
10103	Подоходный налог с юр.лиц-резидентов, удерж.
10104	Подоходный налог с юр.лиц-нерезидентов, удерж.
101201	Подоходный налог с физ.лиц, удерж. у источника
101202	Подоходный налог с физ.лиц, занимающихся предпринимательской деятельностью
102100	Сбор на социальное обеспечение
102100	Социальный налог
104101	Налог на имущество юр. лиц
104102	Налог на имущество физ. лиц
104201	Сбор за регистрацию эмиссии ценных бумаг
104301	Земельный налог на земли с/х назначения
104302	Земельный налог на земли населенных пунктов
104303	Земельный налог на земли промышленности
104304	Земельный налог на земли лесного фонда
104305	Земельный налог на земли водного фонда
104401	Налог на транспортные средства с юр.лиц
104402	НАЛОГ НА транспорт средства с физ. лиц
105101	НДС на товары внутреннего производства
105102	Налог на добавленную стоимость НА товары, ввозимые на территорию РК
105201	Все виды спирта из пищевого сырья
105202	Водка
105203	Ликероводочные изделия
105204	Вина
105205	Коньяки

Рис. 8.12. Справочник Коды бюджетной классификации, перенесенный в учебную конфигурацию

8.3.2. Копирование справочников из другой конфигурации

Рассмотрим иную ситуацию. Например, существует рабочая (в нашем случае учебная) конфигурация и ее пустая копия, т. е. только объекты метаданных, без пользовательских записей. Чтобы перенести справочники Номенклатура, Контрагенты и др. из рабочей конфигурации в пустую, нужно повторить шаги 7 и 8 предыдущего примера. При этом помните, что вы ищете файл нужного вам справочника, а не то название, что указано в книге (оно достаточно условно).

Шаг 9. Так как пустая конфигурация — точная копия рабочей, перенесите файл справочника из каталога рабочей конфигурации прямо в каталог пустой. Наименования файлов одинаковые, система запросит подтверждение на перезапись уже имеющегося файла. Подтвердите перезапись.

Шаг 10. Удалите в папке с базой данных учебной конфигурации все файлы с расширением `cdx`.

Шаг 11. Запустите учебную конфигурацию в пользовательском режиме. После переиндексации вы увидите уже заполненный справочник с данными.

На компакт-дисках ИТС (Информационно-технологического сопровождения), которые поставляются фирмой "ИС" официальным пользователям программы по подписке, предлагается универсальная (т. е. работающая в любой конфигурации) обработка, которая позволяет выгружать в текстовый формат структуру и содержимое выбранных справочников. А также заполнять справочники данными из текстового файла. Обработка позволяет переносить данные из разных конфигураций и справочников. При этом соответствие реквизитов справочников и режим загрузки устанавливается пользователем (рис. 8.13).

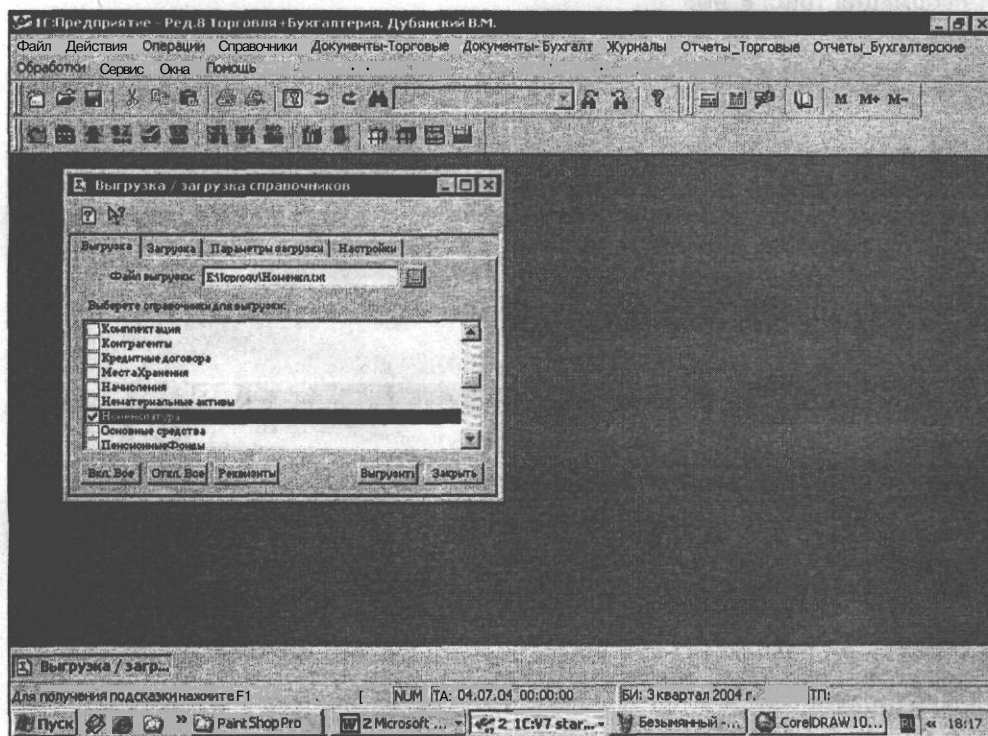


Рис. 8.13. Выгрузка справочника Номенклатура из произвольной конфигурации с помощью универсальной обработки

Для того чтобы выгрузить данные из справочников, нужно указать, какие справочники и реквизиты будут выгружаться, а также имя и путь файла, в который будут выгружены данные.

Для указания подлежащих выгрузке справочников надо выбрать вкладку **Выгрузка** и в списке отметить требуемые справочники. Нажав кнопку

Реквизиты, пользователь может выбрать выгружаемые реквизиты текущего справочника. Для выполнения выгрузки следует указать имя файла, в который нужно выгрузить данные, и нажать кнопку **Выгрузить**. В процессе выгрузки будет создан файл с указанным именем, содержащий выгруженные данные в специальном формате.

Для загрузки данных в справочники на вкладке **Загрузка** укажите файл, в котором они содержатся, и нажмите кнопку **Анализ**. При этом будет заполнена таблица соответствий справочников и таблицы соответствий их реквизитов. В колонке **Справочник.загруж.** содержится набор справочников, полученных из указанного файла загрузки. В колонке **Справочник.текущ.** соответствующие им справочники из текущей конфигурации, в которые будет загружена информация (рис. 8.14).

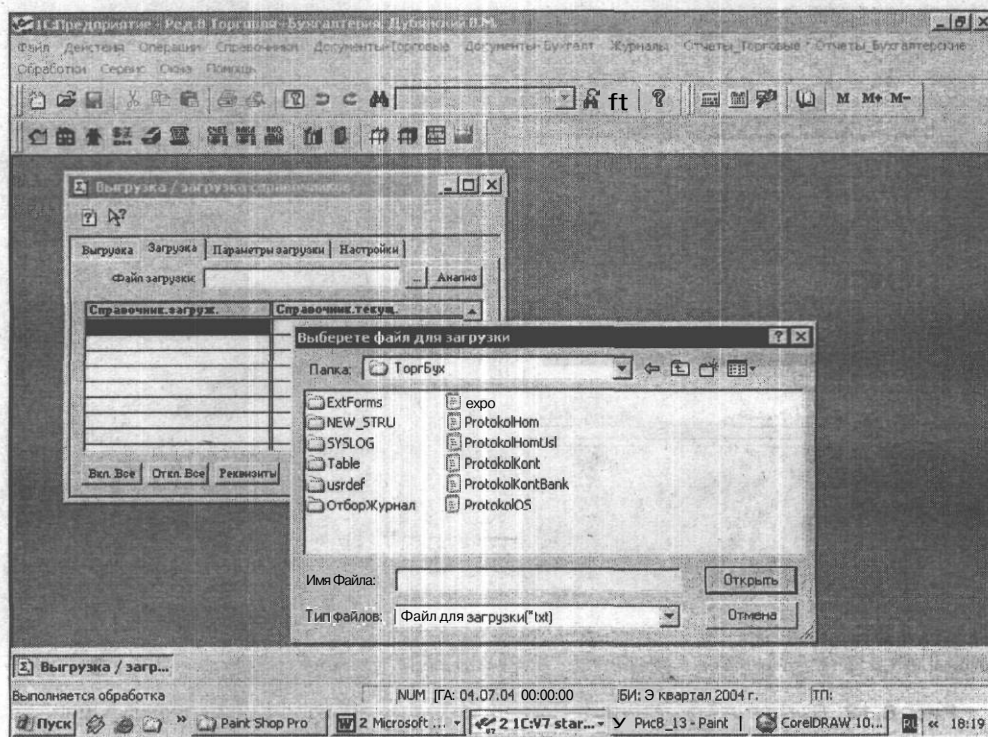


Рис. 8.14. Загрузка данных в учебную конфигурацию с помощью универсальной обработки

Подробное описание работы с этой обработкой можно найти на диске ИТС (выпуска февраля 2002 г.).

8.4. Загрузка данных в "1С:Предприятие" из Microsoft Excel

Система программ "1С:Предприятие" является открытой. Для связи с другими программами могут использоваться встроенные средства загрузки-выгрузки информации в текстовом формате и формате DBF, система поддерживает современные стандарты интеграции программ: DDE, OLE Automation. Одним из примеров использования этих возможностей может служить организация работы "1С:Предприятие" с распространенной системой электронных таблиц Microsoft Excel.

В учебной конфигурации пример для изучения данного раздела — документ Приходная накладная, кнопка **Загрузка из Excel**. Файл примера в Microsoft Excel — **ЗагрузкаНакл.xls**.

В данном примере мы рассмотрим ввод приходных накладных с помощью стандартного сканера в систему программ "1С:Предприятие". Такая задача актуальна для небольших предприятий, работающих с отдельными накладными, включающими несколько десятков или сотен наименований товара. Для решения требуется:

П поступление от поставщиков качественно распечатанных приходных накладных;

- наличие сканера;
- наличие установленной программы распознавания текста Fine Reader v. 4.5 или аналогичной;

П наличие установленной программы MS Excel 97 или более новой.

Рассмотрим алгоритм загрузки данных из файла электронных таблиц MS Excel в документ Приходная накладная.

Реквизит документа Товар имеет тип значения **Справочник.Номенклатура**. Поэтому при загрузке колонки накладной с наименованием товара нужно проверить, есть ли этот товар в справочнике Номенклатура. Если его нет, то добавить в этот справочник. Из справочника номенклатуры товар, соответствующий наименованию в приходной накладной, должен загрузиться в экранную форму документа Приходная накладная. Значения реквизитов Цена, Количество, Сумма и др. должны загружаться непосредственно из соответствующих колонок электронных таблиц. Загрузка номера и даты документа не нужна — в "1С:Предприятие" свой порядковый номер и дата ввода документа, а также итоговые значения. Загрузку наименования поставщика пропустим для упрощения примера.

Приходную накладную необходимо сканировать с распознаванием и поместить документ в лист рабочей книги Microsoft Excel и сохранить его (рис. 8.15).

Microsoft Excel - ЗагрузкаИмя

Файл Правка Вид Вставка Сервис Данные Ввод Формат Окно Справка

Англо-Русский Общий

С15

1 Расходная накладная № 1 от 30.11.02

2

3

4 Клиент: Альфа-Ромео

5

№	Товар	Цена	Количество	Сумма	Ставка НДС	НДС
1	Телевизор	40000	5	200000	16	27686.21
2	Холодильник	30000	1	30000	16	4137.93

Итого: 6 Итого: 230000
..... в том числе НДС: 31724.14

Готово

Пуск

Рис. 8.15. Лист Microsoft Excel с накладной

В "1С:Предприятие" формируется механизм загрузки данных из файла Microsoft Excel. Для этого в модуль формы документа Приходная накладная добавляется ряд процедур и функций (все изменения выполняются на копии информационной базы!).

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 (рис. 8.16):

```
Перем ИмяФайла, ИмяКаталога;
```

Мы не будем описывать здесь операторы процедуры — их можно посмотреть в синтаксис-помощнике либо в документации. Смысл же процедуры — организовать диалог для поиска файла Microsoft Excel, из которого будет производиться загрузка в приходную накладную. Имя указанного в диалоге файла и будет помещено в переменную `ИмяФайла`.

Процедура `Загрузка()` осуществляет непосредственно перенос значений из Microsoft Excel в документ Приходная накладная. Так как наименование товара будет первоначально записываться (или проверяться на наличие)

в справочнике Номенклатура, в процедуре создается объект языка Справочник.номенклатура. Далее необходимо создать объект языка Excel.Application (листинг 8.1). При работе с этим объектом используются операторы языка Visual Basic for Applications (VBA), на котором пишутся приложения в Microsoft Excel.

Перем ИмяФайла, ИмяКаталога

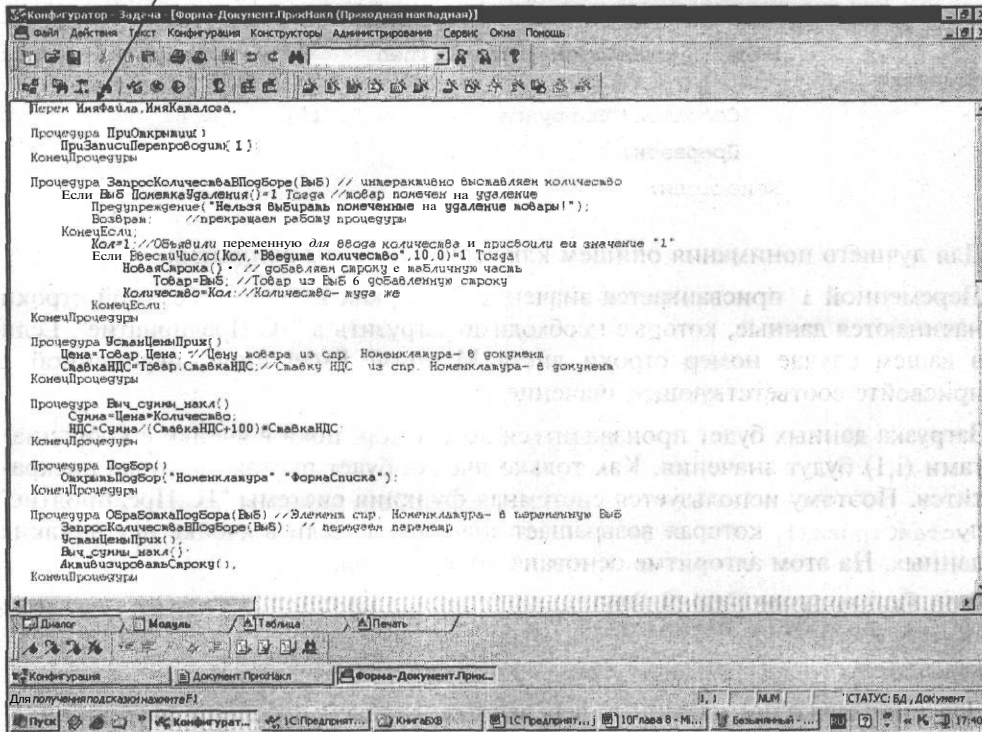


Рис. 8.16. Объявление переменных модуля формы

Листинг 8.1

```

Процедура Загрузка()
    Спр=СоздатьОбъект("Справочник.Номенклатура"); //Создадим объект - Спра-
    вочник Номенклатура
    ВыбратьФайл(); //найдем файл с докумен-
    том Microsoft Excel
    Excel=СоздатьОбъект("Excel.Application"); //создадим объект - ра-
    бочая книга Excel
    РабочаяКнига=Excel.Workbooks;

```

```

РабочаяКнига.Открыть(ИмяФайла, 0, 1); //откроем рабочую книгу - выбранный
файл Excel
i=7; //с 7 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;

```

Для лучшего понимания опишем ключевые моменты алгоритма.

Переменной *i* присваивается значение 7, так как именно с седьмой строки начинаются данные, которые необходимо загрузить в "1С:Предприятие". Если в вашем случае номер строки листа Microsoft Excel иной, переменной *i* присвойте соответствующее значение.

Загрузка данных будет производиться до тех пор, пока в ячейке с координатами (*i*,1) будут значения. Как только ячейка будет пустой, загрузка прекратится. Поэтому используется системная функция системы "1С:Предприятие" ПустаяСтрока, которая возвращает значение 1, если в ячейке нет никаких данных. На этом алгоритме основана конструкция:

```

Пока ПустаяСтрока(Стр)=0 Цикл
.....
КонецЦикла;

```

Для инициализации названных переменных напишем процедуру выбратьФайл (листинг 8.2). Процедура предназначена для открытия стандартного диалога Windows для выбора файла определенного вида.

Листинг 8.2

```

Процедура ВыбратьФайл()
Файл=СоздатьОбъект("ФС");
Если Файл.ВыбратьФайл(0, ИмяФайла, ИмяКаталога,
"Выберите файл, содержащий накладную",
"Файлы Microsoft Excel|*.xls|Все файлы|*.*", "xls")<>1 Тогда Возврат;
КонецЕсли;
ИмяФайла=ИмяКаталога+ИмяФайла;
КонецПроцедуры

```

Далее следует написать функцию, которая позволяет получать значения отдельной ячейки листа Microsoft Excel с использованием языка программирования VBA:

Листинг 8.3

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 ().

Обратите внимание, что значения ячейки листа Microsoft Excel мы получаем при помощи описанной в предыдущем шаге функции получитьЗначение ().

Далее внутри цикла проверяется, имеется ли товар, загруженный из листа Microsoft Excel, в справочнике Номенклатура. Если такого товара нет, он записывается в справочник. Затем этот товар сопоставляется реквизиту Товар документа Приходная накладная.

Листинг 8.4

```
НоваяСтрока (); //_____Добавляем строку в приходную накладную
НомерСтроки=Стр;
ТоварСпр=(СокрЛП(Строка(ПолучитьЗначение(Excel,i,2))));
Если Спр.НайтиПоНаименованию(ТоварСпр,0)=0 Тогда //Если товара
в справочнике Номенклатура нет
    Спр.Новый (); //тогда занесем его туда

Спр.Наименование=(СокрЛП(Строка(ПолучитьЗначение(Excel,i,2))));
Спр.Себестоимость=Число(ПолучитьЗначение(Excel,i,3));
Спр.СтавкаНДС=Число(ПолучитьЗначение(Excel,i,6));
КонецЕсли;
Спр.Записать ();
Спр.НайтиПоНаименованию(ТоварСпр,0); //теперь найдем товар из Excel
уже в справочнике Номенклатура
тов=Спр.Наименование;
Товар=Спр.ТекущийЭлемент (); //и внесем его в строку приход-
ной накладной
```

Значения остальных реквизитов документа Приходная накладная загружаются непосредственно из соответствующих ячеек листа Microsoft Excel с использованием системных функций преобразования типов данных:

Листинг 8.5

```

Цена=Число (ПолучитьЗначение (Excel, i, 3)); //организуем ввод из
Excel и других колонок
Количество=Число (ПолучитьЗначение (Excel, i, 4));
Сумма=Число (ПолучитьЗначение (Excel, i, 5));
СтавкаНДС=Число (ПолучитьЗначение (Excel, i, 6));
НДС=Число (ПолучитьЗначение (Excel, i, 7));
i=i+1;
КонецЦикла;
КонецПроцедуры

```

Добавим в экранную форму документа Приходная накладная кнопку **Загрузка из Excel**. Не забудьте в палитре свойств кнопки, во вкладке **Дополнительно**, в поле **Формула** указать ссылку на процедуру Загрузка о (рис. 8.17).

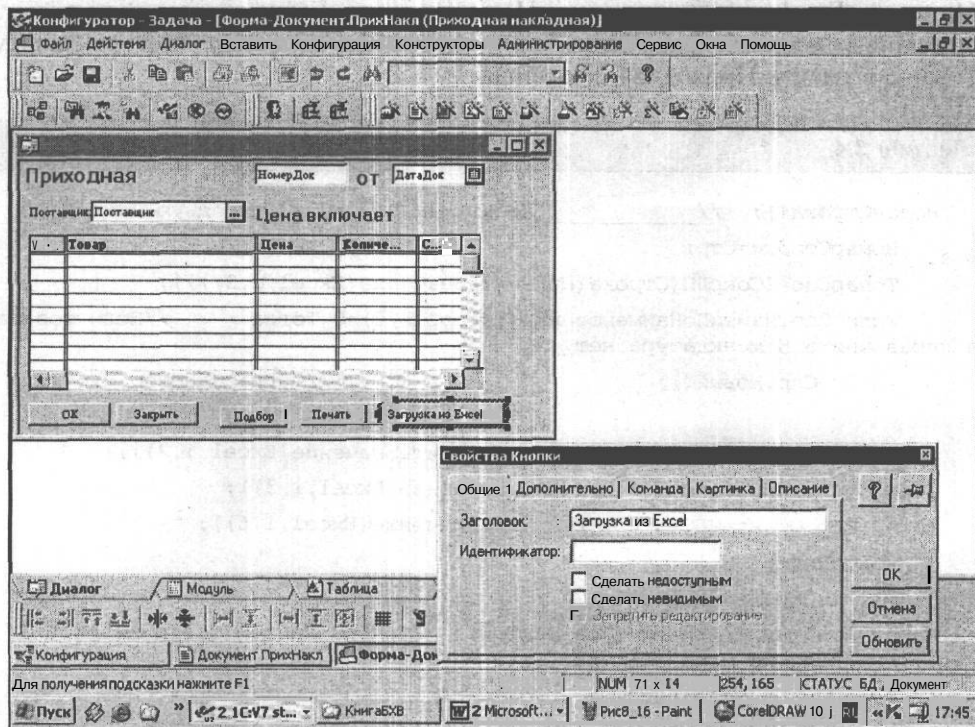


Рис. 8.17. Интерфейс для вызова процедуры загрузки данных из Microsoft Excel

Конечно, в данном примере отсутствует целый ряд полезных сервисных функций. Например, проверка, помечен ли товар в справочнике Номенклатура на удаление, и обработка этой ситуации, интерактивная настройка загрузки данных из ячеек Microsoft Excel и многое другое. Однако мы сознательно пропустили эти детали, чтобы сделать задачу более понятной. Кроме того, в частном случае при настройке конкретного предприятия можно обойтись и без этих сервисных функций.

Если все сделано правильно, то при запуске конфигурации в пользовательском режиме при нажатии кнопки **Загрузка из Excel** приходная накладная и справочник номенклатуры должны автоматически заполниться (рис. 8.18).

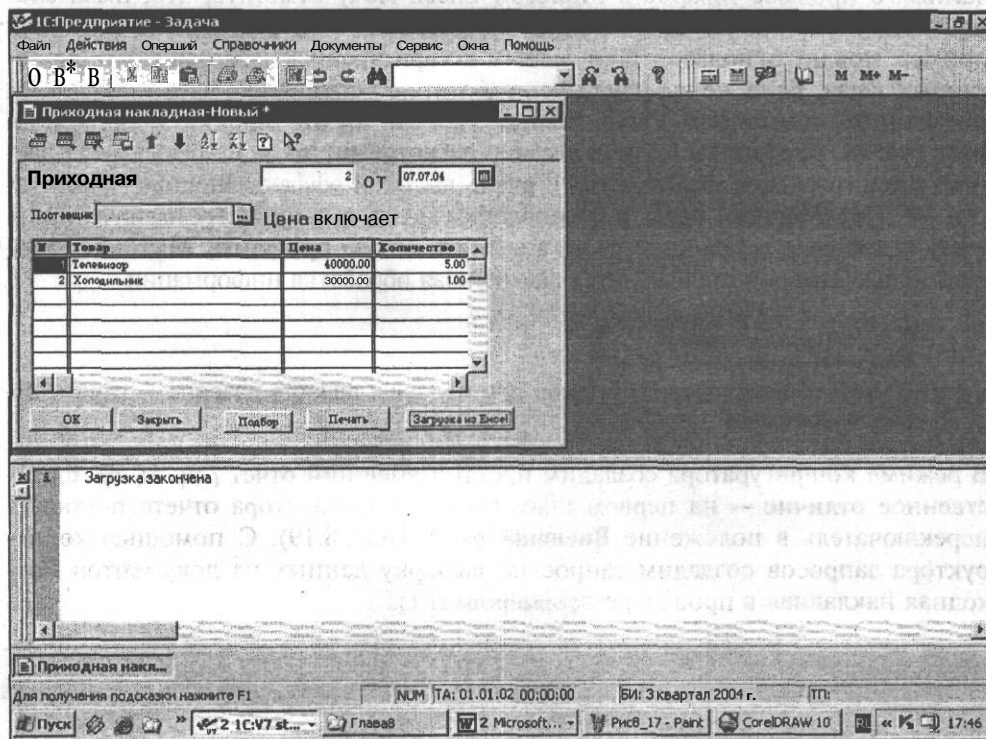


Рис. 8.18. Загрузка документа из Microsoft Excel в пользовательском режиме

8.5. Экспорт данных из "1С:Предприятие" в Microsoft Excel

В данном разделе мы разберем противоположный случай — экспорта данных из "1С:Предприятие" 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" — система для учета, а не для анализа. Компенсировать этот недостаток можно с помощью Microsoft Excel, используя механизм OLE Automation.

Учитывая, что наша конфигурация очень простая, мы не сможем рассчитывать сложные управленческие задачи. Решим простой пример экспорта данных о продаже товаров в Microsoft Excel. Хочу отметить, что, имея элементарные знания по составлению запросов в "1С:Предприятие", данный пример можно использовать в любых конфигурациях для решения аналогичных задач. Если же вы научитесь работать со специфическими объектами метаданных компонент "Оперативный учет" и "Бухгалтерский учет" — планом счетов, операциями, проводками, регистрами, то использовать изучаемый здесь механизм можно будет с очень высокой эффективностью. Фактически "1С:Предприятие" будет использоваться по своему прямому назначению — учету и поставке результатов учета в аналитическую профамму Microsoft Excel, с помощью которой производится дальнейшая обработка информации.

Примечание

В учебной конфигурации пример для изучения данного раздела — отчет Расход Товара Excel.

В режиме конфигуратора создадим простой внешний отчет (см. гл. 6). Единственное отличие — на первом шаге работы конструктора отчета поставьте переключатель в положение **Внешний отчет** (рис. 8.19). С помощью конструктора запросов создадим запрос на выборку данных из документов Расходная накладная в процедуре сформировать ():

Листинг 8.6

Процедура Сформировать ()

```

Перем Запрос, ТекстЗапроса, Таб;
//Создание объекта типа Запрос
Запрос = СоздатьОбъект("Запрос");
ТекстЗапроса =
"//{{ЗАПРОС(Сформировать)
|Период с ВыбНачПериода по ВыбКонПериода;
|ОбрабатыватьДокументы все;

```

```
|Обработать НеПомеченныеНаУдаление;  
I Товар = Документ.РасхНакл.Товар;  
|Количество = Документ.РасхНакл.Количество;  
I Сумма = Документ.РасхНакл.Сумма;  
|Функция КоличествоСумма = Сумма (Количество);  
|Функция СуммаСумма = Сумма (Сумма);  
|Группировка Месяц;  
|Группировка Товар упорядочить по Товар.Наименование;  
|"/})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 ("Сформировать1");  
//Заполнение полей "Заголовки"  
Таб.ВывестиСекцию ("Заголовки");  
Состояние ("Заполнение выходной таблицы...");  
Таб.Опции (0, 0, Таб.ВысотаТаблицы(), 0);  
Пока Запрос.Группировка (1) = 1 Цикл  
    //Заполнение полей Месяц  
    Таб.ВывестиСекцию ("Месяц");  
    Пока Запрос.Группировка (2) = 1 Цикл  
        //Заполнение полей Товар  
        Таб.ВывестиСекцию ("Товар");  
    КонецЦикла;  
КонецЦикла;  
//Заполнение полей "Итого"  
Таб.ВывестиСекцию ("Итого");  
//Вывод заполненной формы  
Таб.ТолькоПросмотр (1);  
Таб.Показать ("Сформировать1", "");  
КонецПроцедуры
```

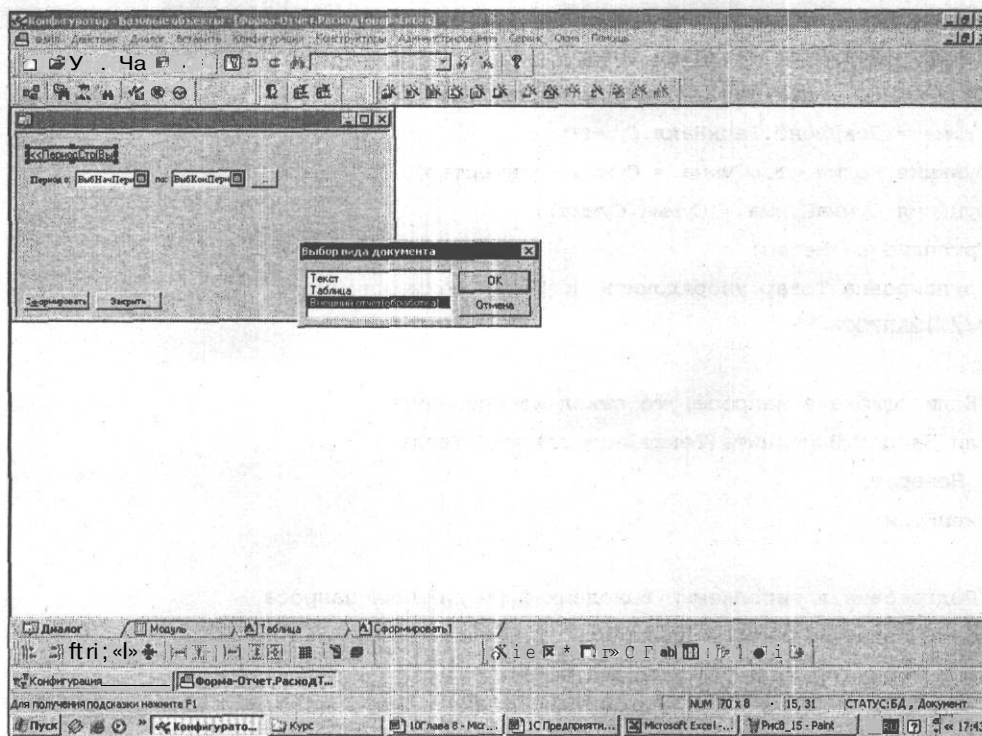


Рис. 8.19. Отчет для экспорта данных в Microsoft Excel, созданный с помощью конструктора

Запрос выбирает данные по продажам каждого товара в определенном месяце выбранного периода.

Начинаем работать с Microsoft Excel. Создадим объектную переменную для работы с книгой Excel и установим параметры книги:

Листинг 8.7

```
//Создадим объект Excel
Excel = СоздатьОбъект("Excel.Application");
//Ограничим книгу Excel одним листом
Excel.SheetsInNewWorkbook = 1;
//Установим заголовок книги
Excel.Caption = "Отчет по продажам";
//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 внесем в него наименования столбцов. Затем с помощью цикла перенесем названия столбцов из списка значений в первую строку листа рабочей книги Microsoft Excel (рис. 8.20) (листинг 8.8).

Листинг 8.8

```
СписокСтолбцов = СоздатьОбъект ("СписокЗначений");  
СписокСтолбцов.ДобавитьЗначение ("Товар");  
СписокСтолбцов.ДобавитьЗначение ("Месяц");  
СписокСтолбцов.ДобавитьЗначение ("Продано");  
СписокСтолбцов.ДобавитьЗначение ("На сумму");  
  
//Заполним наименование столбцов:  
Для Столбец = 1 По СписокСтолбцов.РазмерСписка () Цикл  
    Ячейка = Excel.Cells (1,Столбец);  
    Ячейка.Value = СписокСтолбцов.ПолучитьЗначение (Столбец);  
КонечЦикла;
```

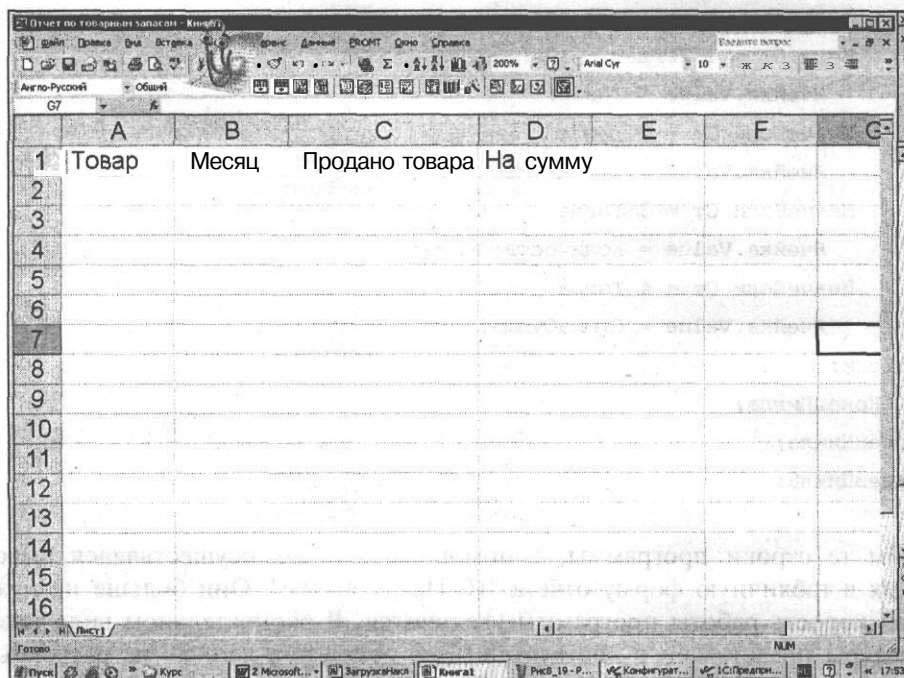


Рис. 8.20. Результат работы фрагментов программы

Теперь организуем вывод данных по месяцам. В данном фрагменте программы обратите внимание на заполнение ячеек таблицы Microsoft Excel:

Листинг 8.9

```

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;
        //Организм 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 Тогда
                Ячейка.Value = Строка(Месяц);
            ИначеЕсли Ст = 3 Тогда
                Ячейка.Value = КоличествоСумма;
            ИначеЕсли Ст = 4 Тогда
                Ячейка.Value = СуммаСумма;
        КонецЕсли;
    КонецЦикла;
КонецЦикла;
КонецЦикла;

```

Удалим те строки программы, с помощью которых осуществлялся вывод данных в табличную форму отчета "1С:Предприятие". Они больше не нужны, а скорость работы программы увеличится. В окончательном виде текст процедуры представлен в листинге 8.10. В результате данные из расходных накладных помещены в таблицу MS Excel (рис. 8.21).

Листинг 8.10

Процедура Сформировать ()

```
    Перемен Запрос, ТекстЗапроса, Таб;  
    //Создадим объект Excel  
Excel = СоздатьОбъект ("Excel.Application");  
//Ограничим книгу Excel одним листом  
Excel.SheetsInNewWorkbook = 1;  
//Установим заголовок книги  
Excel.Caption = "Отчет по товарным запасам";  
//Создадим новую рабочую книгу  
NewBook = Excel.Workbooks.Add;  
  
    СписокСтолбцов = СоздатьОбъект ("СписокЗначений");  
    СписокСтолбцов.ДобавитьЗначение ("Товар");  
    СписокСтолбцов.ДобавитьЗначение ("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);  
    КонечЦикла;  
  
    //Создание объекта типа Запрос  
    Запрос = СоздатьОбъект ("Запрос");  
    ТекстЗапроса =  
    "///{{ЗАПРОС (Сформировать)  
    |Период с ВыбНачПериода по ВыбКонПериода; //указываем временной пери-  
од, внутри которого строится отчет  
    |ОбрабатыватьДокументы все; //используются все документы  
    |Обрабатывать НеПомеченныеНаУдаление; //кроме помеченных на удаление  
    |Товар = Документ.РасхНакл.Товар; //как и информация о товаре  
    |Количество = Документ.РасхНакл.Количество; //о количестве товара  
    |Сумма = Документ.РасхНакл.Сумма; //и сумме, на которую по-  
ставлен товар  
    |Функция КоличествоСумма = Сумма (Количество); //суммируем для подгото-  
гов 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;
        //Организм 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 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
```


	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Товар	Месяц	Продано товара	На сумму									
2	Телевизор "Samsung"	Февраль 03		20000									
3	Телевизор "Samsung"	Май 04		20000									
4	Холодильник	Май 04		3000									
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													
21													
22													
23													
24													
25													
26													
27													
28													
29													
30													
31													
32													
33													

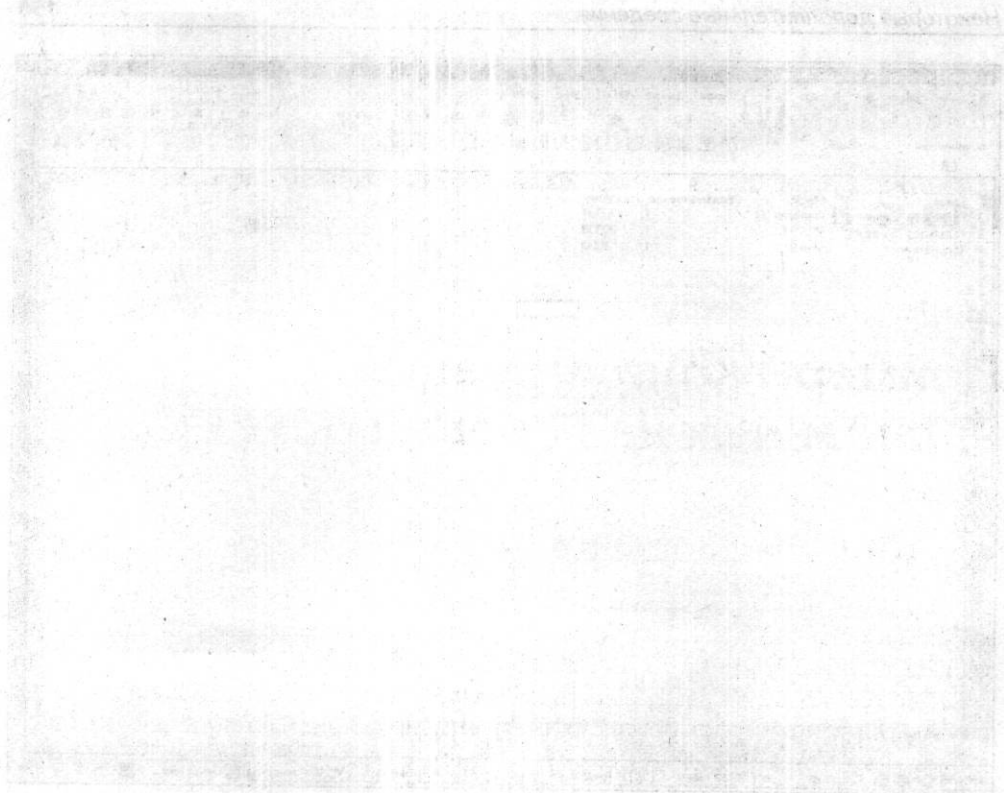
Рис. 8.21. Лист Microsoft Excel с данными "1С:Предприятие"

Обратите внимание на две последние строки процедуры. Команда `Excel.Visible = 1;`

делает окно электронных таблиц видимым и активным. Команда `Excel = 0;`

освобождает оперативную память компьютера, которая резервировалась для объектной переменной `Excel`.

В заключение хотелось бы обратить внимание на следующие моменты: неподготовленного 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.



The following information is provided for your reference:

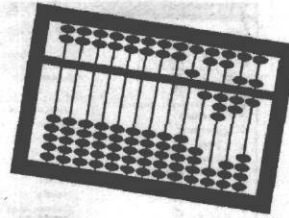
On the basis of the information received from the various sources, it is estimated that the total amount of the grant is approximately \$100,000. This amount is to be distributed over a period of three years, with the first payment of \$30,000 being made at the end of the first year.

The following table shows the estimated distribution of the grant over the three-year period:

Year	Estimated Amount
Year 1	\$30,000
Year 2	\$35,000
Year 3	\$35,000
Total	\$100,000

It is noted that the above figures are only estimates and are subject to change. The actual amounts will be determined by the grant-making authority.

Глава 9



Примеры конфигураций, написанных на базовых объектах

9.1. Вводные замечания

Ряд задач можно автоматизировать в системе "1С:Предприятие", используя только базовые объекты. При этом конфигурации будут работать независимо от того, какая компонента установлена у пользователя. Это значительно увеличивает круг потребителей подобных конфигураций. В данной главе рассмотрены примеры простой и довольно сложной настроек, внедренных на предприятиях и написанных только с использованием базовых объектов.

9.2. Конфигурация Интернет-поддержка

Эта небольшая конфигурация предназначена для быстрого обращения к ресурсам сети Интернет прямо из системы "1С:Предприятие". По сути, это аналог папки Избранное (Favorites) в браузерах. Конфигурация может использоваться как отдельно, так и включаться в любую пользовательскую. Отличительная особенность — возможность автоматической загрузки Web-адресов из папки Избранное. Мысль о написании данной конфигурации пришла автору в связи с тем, что в процессе разработки приходилось часто обращаться к специализированным ресурсам сети, а это занимало достаточно времени.

Ранее мы знакомились с совместной работой системы "1С:Предприятие" и MS Excel. В данной конфигурации приводится пример работы с загрузкой информации из текстовых файлов.

Состав конфигурации:

- справочник Интернет, содержащий Web-адреса;
- обработка Интернет, содержащая информацию для пользователя и кнопку для открытия справочника Web-адреса;
- внешний отчет IntText, предназначенный для автоматической загрузки либо очистки справочника.

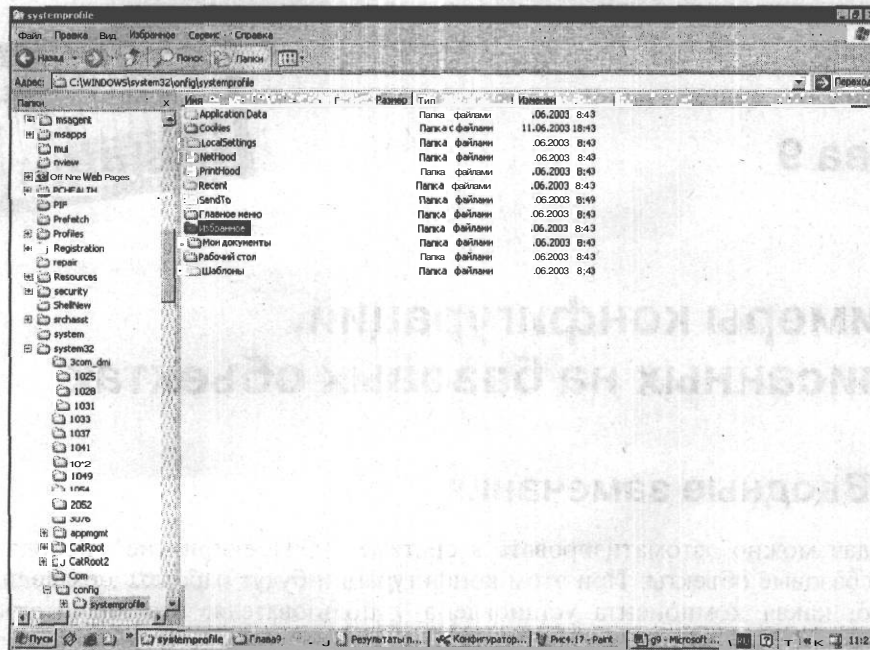


Рис. 9.1. Расположение папки Избранное

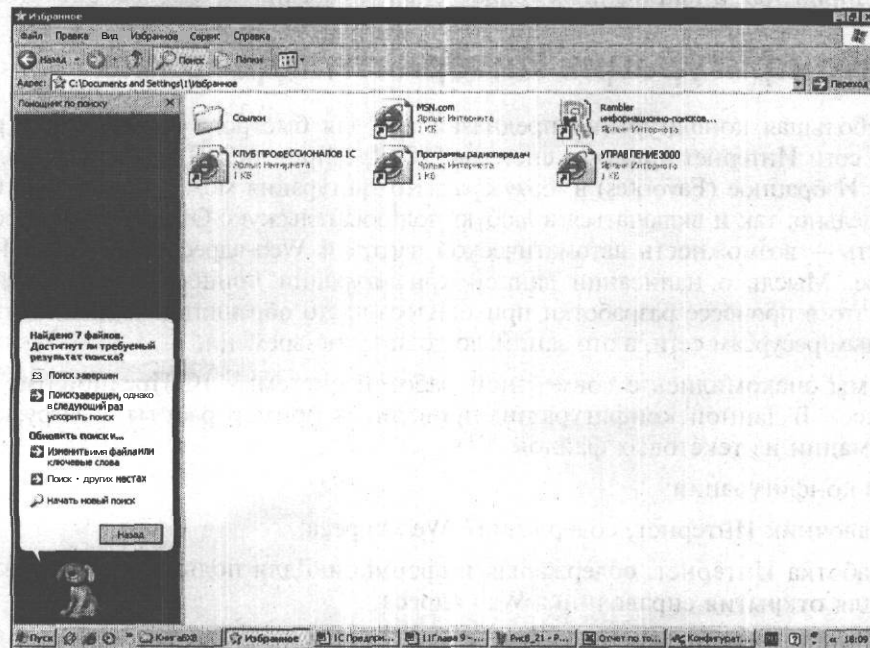


Рис. 9.2. Адреса в папке Избранное

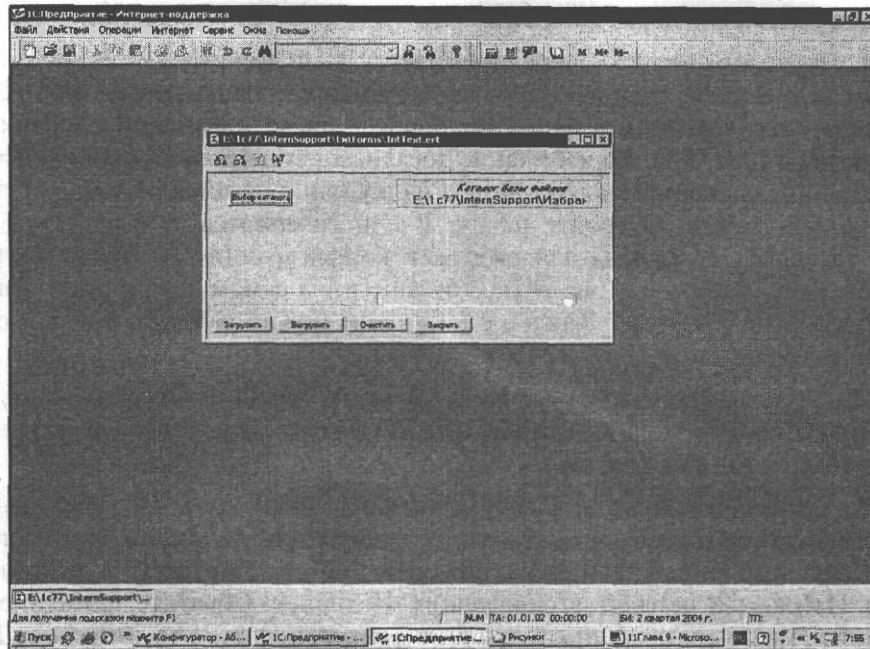


Рис. 9.3. Внешний отчет IntText

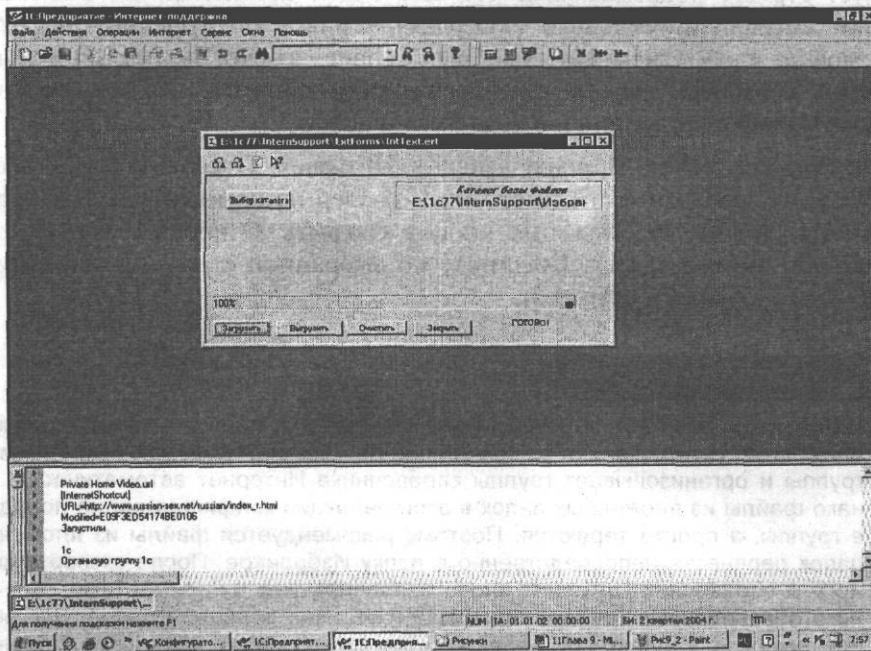


Рис. 9.4. Информация о ходе загрузки файлов

Порядок работы с конфигурацией.

Шаг 1. Предположим, что вы решили внести в справочник Интернет собственные Web-адреса. Найдите на своем компьютере папку Избранное. Скорее всего, она находится в папке Windows на диске с файлами операционной системы (рис. 9.1). Во избежание досадных случайностей, скопируйте ее в какую-нибудь специально созданную папку или в папку ExtForms каталога этой конфигурации. Откройте папку. В ней содержатся Web-адреса. Если какие-то адреса не нужны для работы в конфигурации "1С:Предприятие", можете сразу удалить их (рис. 9.2). Эти адреса, на самом деле, представляют собой обычные текстовые файлы, с которыми можно работать при помощи средств встроенного языка "1С:Предприятие".

Шаг 2. В пользовательском режиме в меню **Файл | Открыть** в папке ExtForms откройте внешний отчет IntText.ert (рис. 9.3). Очистите справочник, нажав на кнопку **Очистить**.

Шаг 3. С помощью кнопки **Выбор каталога** укажите путь к файлам папки Избранное. Необходимо открыть папку и щелкнуть на имени какого-либо из файлов (см. рис. 9.3). Теперь нажмите кнопку **Загрузить**, и файлы из папки Избранное попадут в справочник Интернет. Обратите внимание на дополнительные средства, позволяющие контролировать ход загрузки. Данные о загрузке пишутся в окне сообщений. Кроме того, в форме отчета существует строка, показывающая ход загрузки в процентах. Об окончании загрузки сигнализирует слово "ГОТОВО!" (рис. 9.4). Конечно, без всего этого можно и обойтись. Но подобный сервис необходим при длительных процессах обработки данных, для наглядности процесса. Кстати, код этой процедуры свободно доступен в сети Интернет.

Шаг 4. Закрыть отчет. В меню выбрать **Интернет**. Откроется окно предупреждения о необходимости иметь Web-браузер на компьютере, с которого запускается программа. Нажмите кнопку **Открыть**. Откроется форма справочника для выбора адреса. Щелкните по выбранной строке, и автоматически запустится браузер (рис. 9.5).

Предупреждение

Папка Избранное является обычным каталогом Windows и может содержать вложенные папки. Данная конфигурация рассматривает вложенные папки как группы и организует группы справочника Интернет автоматически. Однако файлы из вложенных папок в этой редакции конфигурации не попадают в группы, а просто теряются. Поэтому рекомендуется файлы из вложенных папок перенести непосредственно в папку Избранное. После экспорта каталога в справочник Интернет вновь заполните уже инструментами справочника образованные группы. Или самостоятельно напишите процедуру, обеспечивающую корректный перенос вложенных папок каталога Избранное в группы справочника Интернет.

Процедура загрузки добавляет новые файлы из каталога, который указан в отчете, к содержащимся в справочнике. Поэтому можно собрать данные из нескольких папок, содержащих ссылки на ресурсы сети Интернет.

В данной редакции процедура очистки справочника производится без дополнительного уведомления пользователя и не может быть отменена после нажатия кнопки **Очистить**.

В предлагаемом примере собрано более 40 ссылок на популярные Web-ресурсы. Не все эти ссылки работают в настоящее время. Пополнять и исправлять список автор предлагает самим пользователям. Для более полноценного использования данной конфигурации можно создать и процедуру автоматической выгрузки справочника Интернет в папку Избранное.

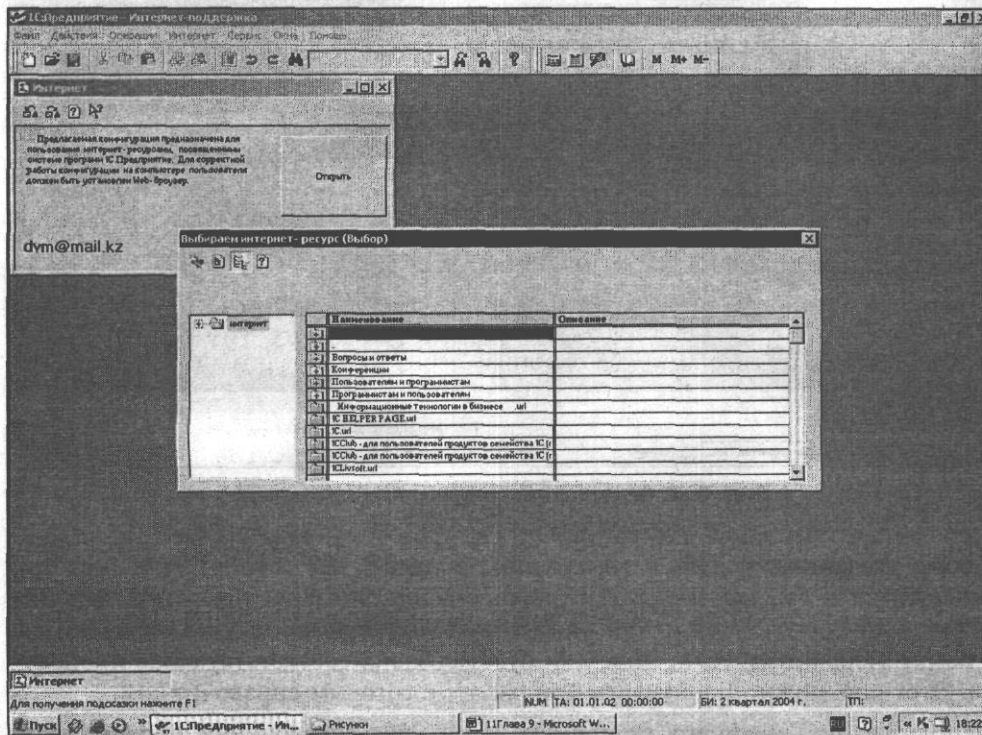


Рис. 9.5. Выбор Web-ресурса в конфигурации

Шаг 5. Форма справочника для выбора содержит два поля: название сайта и произвольное — описание сайта, которое делает сам пользователь, для большего удобства. Это поле заполняется в стандартной форме списка справочника, которую можно открыть в пользовательском режиме через меню **Операции | Справочники**. В данном режиме доступно три поля: **Наименование**, **Описание**, которое заполняется вручную, и **Адрес**, которое заполняется

автоматически и содержит непосредственно адрес сайта. Помимо заполнения поля **Описание**, в этой форме справочника можно также и произвести очистку справочника (рис. 9.6).

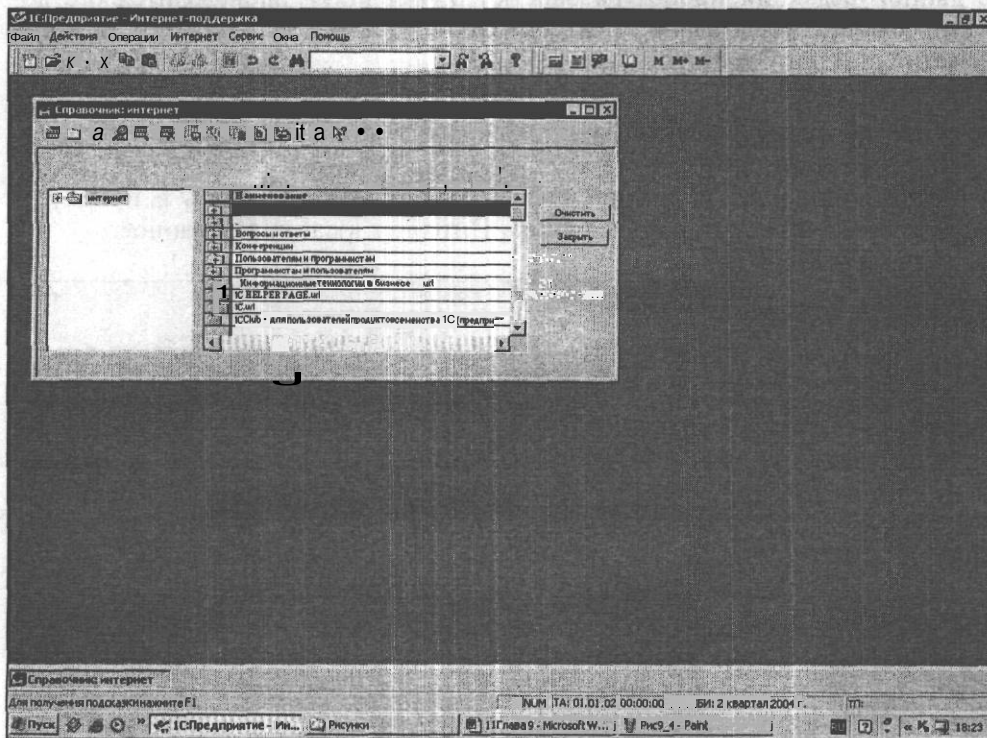


Рис. 9.6. Форма списка справочника Интернет

Кроме указанных полей, в справочник попадает вся информация, которая содержится в ссылке на сайт. Однако она не доступна пользователю. Эта информация служебная и пригодится для выгрузки данных из справочника в папку Избранное, если, конечно, будет создана соответствующая процедура. Этот шаг можно пропустить при работе с готовым справочником.

Использование внешнего отчета для обработки справочника продиктовано тем соображением, что данная конфигурация может включаться в любую пользовательскую. Поэтому встраивание данной конфигурации в пользовательскую сводится к необходимости добавления двух объектов метаданных — справочника Интернет и обработки Интернет. Можно воспользоваться возможностью конфигуратора Объединение конфигураций, либо просто перенести эти объекты метаданных из данной конфигурации в пользовательскую через буфер обмена. Если требуется перенести справочник Интернет вместе с данными, то порядок подобного переноса описан в предыдущей главе.

Непосредственно процедуры очистки справочника и загрузки можно оставить в виде внешнего отчета — его можно скопировать в нужный каталог пользовательской конфигурации. А можно, например, в виде обработки перенести непосредственно в пользовательскую конфигурацию, как удобнее.

9.3. Конфигурация Учет абонентской платы

Алгоритм настройки позволяет осуществлять прием самых разных абонентских платежей: от платы за электроэнергию до платы за кабельное телевидение. Мы подробно рассмотрим версию, внедренную на одном из предприятий. Это учет абонентской платы за отопление и снабжение квартиросъемщиков горячей водой. Абонентский отдел рассчитывает платежи за услуги для нескольких Кооперативов Собственников Квартир (КСК).

Возможности

Можно производить начисления за коммунальные услуги для каждого квартиросъемщика, а также вести непрерывный учет долгов и **переплат**, автоматический настраиваемый расчет пени за просроченные платежи.

Производится автоматическая распечатка квитанций (Счет-извещение и Счет-квитанция), которая возможна как для всех квартиросъемщиков, так и по участкам, и индивидуально для каждого квартиросъемщика в отдельности. Поддерживается корректировка начисления.

При необходимости можно получать основные отчеты по начислению:

- оборотные ведомости по отдельному участку в разрезе лицевых счетов;
- П оборотные ведомости по КСК в разрезе лицевых счетов;
- О краткую оборотную ведомость по КСК;
- П оборотную ведомость об оплате коммунальных услуг на текущее число рабочего месяца;
- сведения об оплате коммунальных услуг и долгах (переплатах) любого квартиросъемщика по выбору;
- П карточка квартиросъемщика — сведения об оплате услуг за определенный период (сличительная ведомость).

Дополнительные отчеты с возможностью сортировки по участкам:

- П список квартиросъемщиков;
- П список должников за коммунальные услуги.

Поддерживается начисление оплаты за коммунальные услуги как по количеству потребителей, проживающих в квартире, так и по индивидуальным приборам учета.

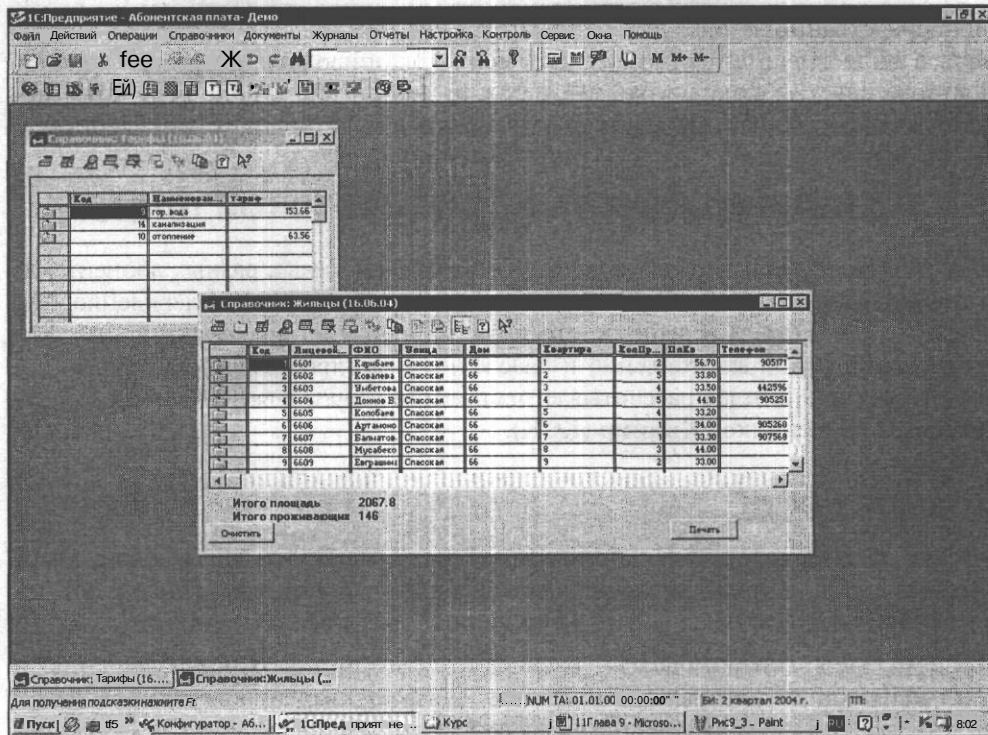


Рис. 9.7. Экранные формы справочников Тарифы и Жильцы

Работа с настройкой начинается с заполнения основных справочников: Тарифы и Жильцы (рис. 9.7).

При нажатии кнопки **Печать** получаем дополнительный отчет Список жильцов КСК, сортированный по лицевым счетам. Если отмечена опция **Сортировать по участку**, список жильцов будет отсортирован по участкам.

Долги потребителей за коммунальные услуги рассчитываются и хранятся в справочнике Долги. Справочник Долги заполняется автоматически во время работы программы (рис. 9.8 и 9.9).

Для начисления ежемесячной платы за коммунальные услуги используется документ Оплата коммунальных услуг (рис. 9.10). В табличной части документа при первом заполнении заполняются поля **Дата платежа**, **Оплата...** (Оплата услуг потребителем) и **Долг...** (Долг за услугу потребителя). Остальные поля заполняются автоматически. При дальнейшей работе поля **Долг...** также заполняются автоматически.

У потребителей, имеющих индивидуальные приборы учета (счетчики), начисление не производится. Сумму начислений проставляет оператор из счета-квитанции квартиросъемщика.

Участок	Номер деп.	ФИО	Дата плат.	Гр	Долг в	Отно	Плат(Отно)	Канализ.	Водост.	Газовые	Контроль
6601	30	Королева Е.Т.	21.12.01	459.14	459.14	3705.37	401.13				
6602	30	Колобаева И.М.	21.12.01			103.24	3.94			86.00	33
6603	30	Ульбикова Е.Т.	17.12.01			2128.26				34.00	54
6604	30	Долгова В.А.	24.12.01			2.30				104.00	107.00
6605	30	Колобаева Р.К.	26.12.01							480.00	105.00
6606	30	Артamonov A.M.	17.12.01	41.48	41.48	5314.59	91.85			25.00	44.00
6607	30	Валитова Т.В.	21.12.01							23.00	22.00
6608	30	Мусабоева К.И.	26.12.01							12.00	62.00
6609	30	Евгoлoвa C.Э.	17.12.01			4078.08	37.63			41.00	41.00
6610	30	Елисеева А.В.	22.12.01		3579.96	3323.47	308.86			41.00	77
6611	30	Пашова Е.С.	26.12.01							41.00	41.00
6612	30	Усолова А.П.	21.12.01							41.00	41.00
6613	30	Усолова А.П.	21.12.01							41.00	41.00
6614	30	Дьячкова В.Н.	06.12.01		11783.20	8666.29				41.00	41.00
6615	30	Кудряшова С.Д.	17.12.01		7233.93	15401.03				41.00	41.00
6616	30	Колесова Ж.С.	17.12.01		1161.20	4616.43				41.00	41.00
6617	30	Мусабоева М.	25.12.01							41.00	41.00
6618	30	Байсаяпова Т.	17.12.01		1318.40	4102.78				41.00	41.00
6619	30	Власова Н.А.	17.12.01			5283.33				41.00	41.00
6620	30	Талиева Ч.	21.12.01		15.37					41.00	41.00
6621	30	Пунтатарова Г.	17.12.01		1428	2122.8				41.00	41.00
6622	30	Усолова М.Б.	26.12.01			2764.86				41.00	41.00
6623	30	Жамалова Е.Е.	29.12.01							41.00	41.00
6624	30	Лобарева А.А.	25.12.01			1836.32				41.00	41.00
6625	30	Мельников В.И.	28.12.01			-0.96				41.00	41.00

Рис. 9.8. Экранная форма справочника Долги

Участок	Лицевой	ФИО	Дата платежа	Долг гор. вода	Долг отопление	Долг Канализация	Под документу номер
1	6601	Королева Е.Т.	21.12.01	459.14	3705.37		38
1	6602	Колобаева И.М.	24.12.01		103.24		38
1	6603	Ульбикова Е.Т.	17.12.01		2128.26		38
1	6604	Долгова В.А.	24.12.01		2.3		38
1	6605	Колобаева Р.К.	26.12.01				38
1	6606	Артamonov A.M.	17.12.01	41.48	5314.59		38
1	6607	Валитова Т.В.	21.12.01				38
1	6608	Мусабоева К.И.	26.12.01				38
1	6609	Евгoлoвa C.Э.	17.12.01		4078.08		38
1	6610	Елисеева А.В.	22.12.01	3579.96	3323.47		38
1	6611	Пашова Е.С.	26.12.01				38
1	6612	Усолова А.П.	21.12.01				38
1	6613	Усолова А.П.	21.12.01				38
1	6614	Дьячкова В.Н.	06.12.01		11783.2	8666.29	38
1	6615	Кудряшова С.Д.	17.12.01		7233.93	15401.03	38
1	6616	Колесова Ж.С.	17.12.01		1161.20	4616.43	38
1	6617	Мусабоева М.	25.12.01				38
1	6618	Байсаяпова Т.	17.12.01		1318.4	4102.78	38
1	6619	Власова Н.А.	17.12.01			5283.33	38
1	6620	Талиева Ч.	21.12.01		15.37		38
1	6621	Пунтатарова Г.	17.12.01		1428	2122.8	38
1	6622	Усолова М.Б.	26.12.01			2764.86	38
1	6623	Жамалова Е.Е.	29.12.01				38
1	6624	Лобарева А.А.	25.12.01			1836.32	38
1	6625	Мельников В.И.	28.12.01			-0.96	38

Рис. 9.9. Печатная форма ведомости должников

КСК "Бастай" дата коммунальных услуг: Май 2004 г.

Пользователь: [] Выделенный: [] Для обработки: [] Контрактная переплата: []
 Без комиссии: []

Долг город: 459.14 К 459.14
 Долг старый: 3705.97 К 7309.82
 Долг старый: 0 К 0

№	Лицевой	ФИО	Дата плат	Тар.Без НДС	Оплата де...	Оплата ГВ	Долг ГВ	Плата ГВ	Платежи ГВ	Коллектор ГВ	Итого ГВ
1	6601	Владимир Е.Т.	16.06.04				459.14				459.14
2	6602	Кочанова Н.И.	16.06.04								
3	6603	Ульбегова Е.Т.	16.06.04								
4	6604	Дюнов Б.А.	16.06.04								
5	6605	Кочубеева Р.К.	16.06.04								
6	6606	Артгановой А.И.	16.06.04					4148			4148
7	6607	Валытова Т.В.	16.06.04								
8	6608	Мусабекжол И.	16.06.04								
9	6609	Багратова С.З.	16.06.04								
10	6610	Султанова С.	16.06.04				3,579.96	290.42			3,579.96
11	6611	Павлова Е.С.	16.06.04								
12	6612	Усупова А.И.	16.06.04								
13	6613	Дмитриев В.И.	16.06.04				1,783.20	343.20			1,783.20
14	6614	Кубошова С.Д.	16.06.04				2,233.93	280.70			2,233.93
15	6615	Коченова С.	16.06.04				1,812.4	94.24			1,812.4
16	6616	Молдыбаев М.	16.06.04								
17	6617	Валибаева Т.	16.06.04				1,38.40	38.40			1,38.40
18	6618	Власова Л.	16.06.04								
19	6619	Урманов Ч.	16.06.04				85.37	85.37			85.37
20	6620	Тунгатарова Г.	16.06.04				142.90	142.1			142.90
21	6621	Лавринов А.В.	16.06.04								
22	6622	Энриков М.В.	16.06.04								
23	6623	Кудамбаев К.	16.06.04								
24	6624	Любева А.А.	16.06.04								

СК | Закрыть | Печать книги | По участку | Оборотно-сальдо | По участку

ОпконУсТариф-Новый

Для получения подсказки нажмите F1

МММ ТА: 01.01.00 00:00:00 БИ: 2 квартал 2004 г.

1 Глава 9 - Микро... Рис 9 - Райк 8:09

Рис. 9.10. Экранная форма документа Оплата коммунальных услуг

Документ заполняется оператором расчетного центра один раз в месяц, причем не обязательно одновременно. Заполнение может происходить по мере поступления документов об оплате услуг за предыдущий месяц.

Существует специальный настраиваемый механизм расчета пени за неоплаченные услуги. В частности, пеня может начисляться и за неоплаченные услуги при наличии индивидуальных приборов учета. При этом настройки позволяют автоматизировать процесс начисления пени в соответствии с законодательством и внутриведомственными инструкциями.

Корректировочная часть в шапке документа позволяет изменить начисления не полностью предоставленных услуг, сделать перерасчет в случае установки или снятия индивидуальных приборов учета.

Информационная часть в шапке документа позволяет оперативно получать информацию о состоянии платежей каждого потребителя. Кроме того, не выходя из документа можно получать основные отчетные формы, как:

- полная оборотная ведомость по расчетам со всеми потребителями услуг в КСК (рис. 9.11). Представляет целостную картину полноты и структуры оплаты;

КСК
расчетам с квартиросъемщиками
за Декабрь 2000 г.

Участок	Лицевой	Гор. вода	Отопление	Канализация
38	Вход. сальдо	0	0	0
	К оплате	0	0	0
	Оплачено	0	0	0
	Оплачено ИПУ	нет ИПУ	0	нет ИПУ
	Оплачено Пени	ОплатаПени	ОплатаПени	ОплатаПени
6601	Сальдо	ОплатаГ+Q	ОплатаОтоп+Q	
	Вход. сальдо	0	-2081.82	0
	К оплате	381	5164.03	0
	Оплачено	0	2087	0
	Оплачено ИПУ	201	0	0
3802	Сальдо	ОплатаГ+Q	ОплатаОтоп+Q	
	Вход. сальдо	0	0	0
	К оплате	635	1837.37	0
	Оплачено	0	1837.37	0
	Оплачено ИПУ	635	0	0
5603	Сальдо	ОплатаГ+Q	ОплатаОтоп+Q	
	Вход. сальдо	-303	-7.95	0
	К оплате	303	1824.02	0
	Оплачено	0	1824.02	0
	Оплачено ИПУ	0	0	0

Рис. 9.11. Полная оборотная ведомость

КСК
Оборотная ведомость
по расчетам с квартиросъемщиками
38 Декабрь 2000 г.

Услуга	Входящее сальдо	Начислено Пени	Начислено с учетом пени	К оплате	Оплачено с учетом пени	Оплачено пени	Сальдо
Гор. вода	-37787.84	0	4131.84	56112.28			
Отопление	-73455.41	0	111916.33	183577.86			

Гл. бухгалтер
Председатель

Рис. 9.12. Краткая оборотная ведомость

ИСПредприятие - Абонентская плата - Демо - [Квиток *1]

Файл Действия Таблица Вид Операции Справочники Документы Журналы Отчеты Настройка Контроль Сервис Окна Помощь

RTC1

ТОО "Бастау" РИН 600800046894, Р/счет 802467664 АФ ОАО Банк "ТуранАлен" г. Алматы
оплату производить в филиале с/б №

С Ч Е Т - И З В Е Щ Е Н И Е лицевой счет 6601 Тел. плате. 905171

Платательщик							Телефон	36891 1	
Адрес	Спасская 66 кв. 1	Полезная	площадь	56.7	начисл.	долг	Пеня	Кол человек	2
Углы	последняя	предыдущ	коп-во	тариф	0	0	0	к оплате	Оплачено
Горячая вода	87	0		153.66					
				т/куб.м.					
отопление				63.56	3603.85	5105.84	102.12	6811.81	
				т/м2		долг			
горячая вода				0	0	долг		0	
				т/чел		долг			
Срок оплаты д	31 12 01	За Декабрь 2001 г.						Итого	8811.81

С 1 декабря 2001 года вводится пеня на просроченные платежи.

С Ч Е Т - К В И Т А Н Ц И Я лицевой счет 6601 Тел. плате. 905171

Платательщик							Телефон	368911	
Адрес	Спасская 66 кв. 1	Полезная	площадь	56.7	начисл.	долг/переп	Пеня	Кол человек	2
Углы	последняя	предыдущ	коп-во	тариф	0	0	0	к оплате	Оплачено
Горячая вода	87	0		153.66					
				т/куб.м.					
отопление				63.56	3603.85	5105.84	102.12	8811.81	
				т/м2		долг			
горячая вода				0	0	долг		0	
				т/чел		долг			
Срок оплаты д	31 12 01	За Декабрь 2001 г.						Итого	8311.81

Журнал документов Сервис... | Сервис/Стариф - 33 | Квиток *1

Для получения подсказки нажмите F1 | NUM TA: 01.01.00.00:00:00 | БИ: 2 квартал 2004 г. | ИП:

Пуск | Конфигуратор - Аб... | ИСПредприятие... | Курс | 1 Глава9 - Microso... | Рк9_12 - Park | В: IS

Рис. 9.14. Печатная форма счета-квитанции и счета-извещения

Необходимо помнить и о том, что настройка, использующая базовые объекты, не реализует специфические возможности компонент. В частности, в данной конфигурации нет возможности работы с бухгалтерскими проводками. Если нужно вести бухгалтерский учет коммунальных платежей, настройку необходимо внедрить в конфигурацию бухгалтерского учета.

Предметный указатель

А

- Автоматически присвоенный код 25
- Автоматически настраиваемый расчет пени 163
- Автоматическое заполнение реквизитов 37
- Алгоритм обработки шаблона печатной формы 68

Б

- Базовые объекты метаданных 14

В

- Ввод остатков 133
- Ветви дерева справочников 14
- Ветвь метаданных 17
- Вид журнала 96
- Визуальные средства настройки конфигуратора 41
- Встроенные средства архивирования 128
- Встроенный язык системы 41
- Выбор кодовой страницы, 11
- Выборка из объекта метаданных 63
- Выборка из справочника 70
- Выгружаемые реквизиты 140
- Выгрузка данных 131
- Вызов процедуры 73
- Вычисление НДС 42

Г

- Глобальная процедура 58
- Глобальный контекст 42
- Глобальный модуль 42, 54, 57, 58
- Графы журнала 96, 98
- Группировки запроса 105
- Группы справочника Интернет 160

Д

- Дата регистрации предприятия 17
- Добавление справочника 133
- Документ Приходная накладная 31, 35
- Дополнительные опции вывода 100
- Дополнительные реквизиты 26, 29
- Доступ к атрибутам 64

Е

- Единица измерения 22

Ж

- Журнал:
 - документов 56
 - расчетов 15

З

- Заголовок диалогового окна 100
- Загрузка сохраненных данных 131
- Запрос ввода пароля 125
- Запрос на выборку данных 148
- Зарезервированные слова языка 41

И

Идентификатор 17, 20, 23, 26, 32, 33
Идентификатор реквизита 29
Имя секции 69
История значений константы 21
Итог по колонке 60
Итоговая часть документа 77
Итоговые значения 60, 62

К

Каталог товаров 69, 71—73, 76
Ключ защиты 7
Ключевое слово 57, 58, 60
Код 20, 22, 26, 29
Команды встроенного языка 27
Комментарий 19, 23, 32, 43
Компонента:
 "Бухгалтерский учет, 4, 15, 24
 "Оперативный учет" 4, 14
 "Расчет" 4, 15
Конструктор:
 запроса 106, 112
 печати 63, 81
Контроль уникальности 25, 26
Конфигурация объектов 8
Копии информационных баз 128
Краткая оборотная ведомость 167, 168
Краткое наименование 22, 26, 29

Л

Локальный контекст 58
Локальный контекст модуля 65

М

Меню:
 второго уровня 116
 первого уровня 115, 116
Механизм OLE 148
Механизм:
 загрузки данных 142
 расчета пени 166
Многострочная часть документа 77, 78
Модуль отчета 108

Н

Набор:
 интерфейсов 113
 пользовательских прав 120, 113

О

Оборотная ведомость 163, 168
Обработка 87, 90
Объект метаданных 13, 138
Объект языка 143
Ограничения в доступе 113
Окно:
 именования ячеек 66
 конструктора запроса 105
 редактирования формы 76
Описание конфигурации 133
Отчет о закупках 103

П

Палитра свойств:
 константы 21
 поля ввода 29
 реквизита 26
Панель инструментов 118
Параметр процедуры 50
Перебор элементов в цикле 71
Переменные запроса 105
Перечисление 87—90
Периодическая константа 21
Поле формул 61
Полная оборотная
 ведомость 166, 167
Полное наименование 22, 26
Предопределенная процедура 47
Признак периодичности 20
Приходная накладная 147
Проверка синтаксиса
 программы 44
Проводка 15, 96
Программа архивации 131
Простой внешний отчет 148
Прототипы алгоритмов печати 63
Процедура автоматической
 выгрузки 161

Р

- Раздел:
 - основной программы 42
 - переменных 42
 - процедур и функций 42
- Разделители между тройками цифр 21
- Режим конфигуратора 73, 76, 81, 83
- Резервная копия информационной базы 129

С

- Свойства ячейки 67
- Себестоимость 23, 27, 38
- Секция Подвал 78
- Сервисное обслуживание 9
- Сетевая бухгалтерия 5
- Синоним 17
- Синтакс-помощник 46
- Скорость работы программы 152
- Словарь данных 136
- Служебный объект метаданных 71
- Создание печатных форм 83
- Создание списка пользователей 126
- Состав конфигурации 157
- Сохранение информационной базы 127
- Сохранение метаданных 19
- Список:
 - активных пользователей 10
 - констант 20
 - наборов прав 122
 - сохраняемых файлов 130
- Справочник Контрагенты 29
- Справочник товаров 22
- Средства загрузки-выгрузки 141
- Ставка НДС 23, 27, 29
- Стандарты интеграции 141

Т

- Табличная часть документа 50, 52, 60
- Табличный отчет 76
- Текстовый формат 139
- Технологическая платформа 8
- Тип значения 19, 34, 35
 - Строка 29
 - Число 21
- Тип номенклатуры 22, 27, 72
- Тип товара 22, 27

Ф

- Фактический параметр 58
- Флажок периодичности 21
- Форма элемента 29, 30
- Формальный параметр 58
- Формирование результатов запроса 105
- Формы поставки 4

Ц

- Цена 23, 27, 29, 35, 38, 39
- Цикл выборки 64

Ш

- Шаблон печатной формы 65, 66, 68, 70-72, 74, 76, 79, 81, 83, 85, 101, 108
- Шаблон экранной формы 27, 35, 37

Э


- Экранная форма списка 29
- Экспорт данных 147

Дубянский Владимир Маркович, разработчик программного обеспечения, преподаватель учебного центра группы компаний "Glotur", автор курсов по основам программирования в системе "1С:Предприятие". Разработал и внедрил более 30 приложений в области бухгалтерского учета, статистики и системного анализа. Постоянный автор журнала "Бухгалтер плюс компьютер", "Computer club magazine", в которых опубликовал более 50 статей.

1 С: Предприятие

Конфигурирование в администрирование для начинающих

Э К С П Р Е С С - К У Р С



Автор делится с читателями богатым опытом программиста и преподавателя. В доступной форме излагает основы конфигурирования и программирования в системе "1С:Предприятие" версии 7.7. Все действия расписаны пошагово, с иллюстрациями к каждому шагу и листингами процедур. Книга адресована тем, кто никогда не программировал, но хочет самостоятельно изучить основы разработки конфигураций. Особенно она может быть полезна бухгалтерам, менеджерам, руководителям, которые постоянно используют "1С: Предприятие" в работе, а также студентам, изучающим программирование в этой среде.

БХВ Петербург

190005, Санкт-Петербург,
Измайловский пр., 29

E-mail: mail@bhv.ru
Internet: www.bhv.ru

тел.: (812) 251-42-44
факс: (812) 251-12-95

ISBN 5-94157-278-6



9 785941 572786