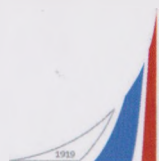


СРЕДНЕЕ  
ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАНИЕ

Электронно-  
Библиотечная  
Система  
znanium.com



ФИНАНСОВЫЙ  
УНИВЕРСИТЕТ  
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

Э.Г. Дадян

# РАЗРАБОТКА БИЗНЕС-ПРИЛОЖЕНИЙ НА ПЛАТФОРМЕ «1С:ПРЕДПРИЯТИЕ»



У Ч Е Б Н О Е   П О С О Б И Е



**Э.Г. ДАДЯН**

**РАЗРАБОТКА  
БИЗНЕС-ПРИЛОЖЕНИЙ  
НА ПЛАТФОРМЕ  
«1С:ПРЕДПРИЯТИЕ»**

УЧЕБНОЕ ПОСОБИЕ

*Рекомендовано Межрегиональным учебно-методическим советом профессионального образования в качестве учебного пособия для учебных заведений, реализующих программу среднего профессионального образования по укрупненной группе специальностей 38.02.00 «Экономика и управление» (протокол № 8 от 22.06.2020)*

**Электронно-  
Библиотечная  
Система**  
**znanium.com**

Москва  
ИНФРА-М  
2021

УДК 004(075.32)  
ББК 32.973.26-018я723  
Д14

ФЗ  
№ 436-ФЗ

Издание не подлежит маркировке  
в соответствии с п. 1 ч. 4 ст. 11

Рецензенты:

*Некрылов И.И.*, старший преподаватель Финансового университета при Правительстве Российской Федерации;

*Барабаш Д.А.*, кандидат экономических наук, доцент Финансового университета при Правительстве Российской Федерации

Далян Э.Г.

Д14

Разработка бизнес-приложений на платформе «1С:Предприятие» : учебное пособие / Э.Г. Далян. — Москва : ИНФРА-М, 2021. — 305 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование).

ISBN 978-5-16-016648-3 (print)


ISBN 978-5-16-109236-1 (online)

Учебное пособие по разработке бизнес-приложений на платформе «1С:Предприятие» позволит подготовленным пользователям самостоятельно и быстро изучить методы и средства разработки и редактирования прикладных решений в среде платформы «1С:Предприятие».

В учебном пособии проиллюстрированы решения задач по разработке реального бизнес-приложения. Выполнение упражнений позволит пользователю подробно ознакомиться с системой «1С:Предприятие», изучить объекты конфигурации и технологии работы с ними, освоить язык программирования 1С, методы и средства разработки и редактирования приложений.

Для студентов средних профессиональных и высших учебных заведений, изучающих экономические дисциплины, а также для всех желающих изучить средства и методы разработки и редактирования прикладных решений на платформе «1С:Предприятие».

УДК 004(075.32)  
ББК 32.973.26-018я723

Материалы, отмеченные знаком ,  
доступны в электронно-библиотечной системе Znaniium.com

ISBN 978-5-16-016648-3 (print)  
ISBN 978-5-16-109236-1 (online)

© Далян Э.Г., 2021

Подписано в печать 08.10.2020.  
Формат 60×90/16. Бумага офсетная. Печать цифровая.  
Гарнитура Newton. Усл. печ. л. 19,06.  
ППТ50. Заказ № 09271  
ТК 682798-961482-081020

ООО «Научно-издательский центр ИНФРА-М»  
127214, Москва, ул. Полярная, д. 31В, стр. 1.  
Тел.: (495) 280-15-96, 280-33-86. Факс: (495) 280-36-29.  
E-mail: books@infra-m.ru <http://www.infra-m.ru>

Отпечатано в типографии ООО «Научно-издательский центр ИНФРА-М»  
127214, Москва, ул. Полярная, д. 31В, стр. 1  
Тел.: (495) 280-15-96, 280-33-86. Факс: (495) 280-36-29

## **Предисловие**

В электронно-библиотечной системе Znanium.com содержатся материалы, предназначенные для самостоятельного изучения и исполнения. Прежде всего это 13 пар вариантов демонстрационной конфигурации и информационной базы, иллюстрирующие состояние разрабатываемого прикладного решения на момент прочтения соответствующего юнита или приложения. Конфигурации созданы в версии «1С:Предприятие 8.3».

Конфигурации не являются законченными (с предметной точки зрения) прикладными решениями и не предназначены для реального ведения учета. Они являются исключительно набором примеров, поясняющим текст книги.

Также электронная часть содержит все фрагменты листингов, приведенных в учебном пособии. Использование этих фрагментов может быть полезным как при чтении книги, так и в дальнейшей работе. Любой фрагмент листинга может быть просто перенесен мышью в произвольное место модуля.

Электронная часть включает также файлы картинок, используемые при разработке бизнес-приложения. Здесь же помещен словарь используемых терминов и выражений.

Курс позволяет студентам изучить основы конфигурирования в системе «1С:Предприятие 8.3» (в дальнейшем «1С:Предприятие»)

1. Обучение ведется по «лекционно-практической» технологии, когда после получения дозы теоретического материала и задания в виде соответствующего упражнения студент самостоятельно его выполняет.
2. Привычного деления на лекционные и практические занятия в курсе нет, студент выполняет задания сразу же, после получения и обсуждения с преподавателем текущей дозы теоретического материала. Каждое последующее задание строится на основе данных, полученных при правильном выполнении всех предыдущих.
3. Материалы курса поделены на темы-юниты. В состав каждого юнита входит теоретический материал, практические задания, система помощи и система контроля.
4. Обучение ведется на основе сквозного примера, т.е. студент шаг за шагом самостоятельно, опираясь на материалы курса, создает конфигурацию на платформе «1С:Предприятие», последовательно выполняя упражнения. Внешне упражнение имеет следующий вид:

### Упражнение № 1.1

---

1. Нажмите на кнопку «Добавить» в диалоге «Запуск «1С:Предприятие».
  2. В диалоге «Добавление информационной базы» (рис. 1.4) поставьте переключатель в положение «Создание новой информационной базы».
  3. Перейдите к следующему этапу регистрации, нажав на кнопку «Далее».
  4. Для того чтобы создать пустую ИБ, необходимо на этом этапе установить переключатель в положение «Создание информационной базы без конфигурации для разработки новой конфигурации или загрузки выгруженной ранее информационной базы» (рис. 1.5).
- 
5. Каждое упражнение имеет номер. В данном случае — это Упражнение № 1.1. Первая цифра соответствует номеру Юнита, вторая — порядковый номер упражнения данного юнита.
  6. Если заголовок выглядит как: **Упражнение № 1.1 (продолжение)** — часть упражнения. На части также делятся крупные упражнения для удобства их выполнения.
  7. Практическое выполнение упражнений является обязательным.

### Постановка задачи сквозного примера

1. Автоматизировать деятельность некоторой условной компьютерной организации — «Объединение «Содружество»».
2. В результате встречи с нашим условным заказчиком выяснилось, что из всех разделов учета, имеющихся на предприятии, необходимо выполнить автоматизацию управленческого учета только для основных средств (ОС).
3. Необходимо, чтобы конфигурация удовлетворяла следующим требованиям:
  - ✓ ввод всех данных и документов должен выполняться не ранее *даты регистрации* объединения;
  - ✓ во всех отчетах, формируемых ИБ, должно появляться название объединения;
  - ✓ для выполнения ряда операций необходимо пересчитывать рублевые суммы в условные единицы (УЕ). В связи с этим конфигурация должна формировать отчет «История курса УЕ» следующего содержания:

Объединение "Содружество"	
История курса УЕ	
Дата	Значение
28.12.2007	25.2600
01.01.2008	24.8500
01.01.2009	23.7844
31.01.2009	23.6222
02.03.2009	23.5729
01.04.2009	23.3801
01.05.2009	23.1021
31.05.2009	23.7090
30.06.2009	23.3483
30.07.2009	23.2450
18.08.2009	23.3532

4. Объединение состоит из нескольких организаций. При этом необходимо в одной ИБ организовать учет от нескольких организаций — «Фирм» как с возможностью получения консолидированных данных в целом по объединению, так и с детализацией по каждой фирме в отдельности.
5. Каждая фирма в своем составе может иметь несколько подразделений, каждое из которых, в свою очередь, также может состоять из нескольких подразделений. При этом количество подразделений и уровней иерархии заранее неизвестно.
6. Работники, занятые в объединении, могут числиться как на какой-то фирме в целом, так и в отдельном подразделении. При

этом система должна хранить в ИБ для каждого сотрудника табельный номер, уникальный в пределах всего объединения.

7. Учет основных средств (ОС) должен удовлетворять следующим условиям.
- 7.1. Инвентарный номер должен быть уникальным для каждой «Фирмы».
  - 7.2. Балансовая стоимость выражена в рублях и может изменяться с течением времени. Желательно, чтобы изменения были оформлены документами.
  - 7.3. Рыночная стоимость выражена в УЕ. Замечания, сделанные для балансовой стоимости, справедливы и для рыночной стоимости.
  - 7.4. Подразделение может изменяться с течением времени, т.е. возможна передача ОС из одного подразделения в другое.
  - 7.5. Каждое из основных средств должно принадлежать одной из групп ОС:
    - ✓ здания;
    - ✓ сооружения;
    - ✓ передаточные устройства;
    - ✓ машины и оборудование;
    - ✓ инструмент;
    - ✓ производственный и хозяйственный инвентарь;
    - ✓ прочие основные фонды.
  - 7.6. Для тех ОС из группы «Машины и оборудование», которые состоят из отдельных деталей (и для которых необходимо вести учет их комплектующих) в ИБ хранить список в виде следующей таблицы:

Деталь	Количество	Цена (У.Е.)	Сумма (У.Е.)
Системная плата с ISA шиной	1	115.00	115.00
Память DIMM 64Mb SDRAM	2	88.00	176.00
...	...	...	...
<b>Итого:</b>	<b>11</b>		<b>851.50</b>

- 7.7. Для каждого из ОС необходимо помнить Дату ввода в эксплуатацию и Дату списания.
- 7.8. Для каждой записи из справочника ОС должна быть возможность хранить в ИБ графическое изображение.
- 7.9. Для учета Компьютеров и оргтехники в ИБ должна быть заранее предусмотрена отдельная группа записей под именем «Компьютеры».

8. Для каждой Детали необходимо, помимо ее наименования, хранить в ИБ ее цену в У.Е.
9. В процессе функционирования объединения отдельные фирмы производят закупку как ОС, так и деталей к ним. Действия по поступлению ОС (Деталей) должны оформляться отдельным документом, табличная часть которого должна выглядеть следующим образом:

ОС (Деталь)	Количество	Цена (Руб.)	Сумма (Руб.)	Цена (У.Е.)	Сумма (У.Е.)
Принтер лазерный	1	13 038,00	13 038,00	410	410
Картридж	8	750	6 000,00	23,58	188,64
<b>Итого:</b>	<b>9</b>	<b>-</b>	<b>19 038,00</b>	<b>-</b>	<b>598,64</b>

При этом в одном документе могут присутствовать как ОС, так и детали. Для ОС количество в документе всегда должно равняться «1», так как по условиям постановки задачи в одной инвентарной карточке числится одно основное средство.

После закупки как ОС, так и отдельные детали числятся как «*Находящиеся в запасе*». При этом они могут храниться на складе фирмы, который специально предназначен для хранения ОС и (или) деталей, которые еще не эксплуатируются.

10. По мере необходимости купленные ОС могут вводиться в эксплуатацию, в результате чего конкретное средство записывается за конкретным подразделением.
11. В процессе эксплуатации тех ОС, для которых известен состав их деталей, могут производиться их модернизация (*upgrade*) или ремонт, которые оформляются внутрифирменным документом «Изменение ОС». В результате такой модернизации количество и состав деталей, которые находятся в запасе, могут изменяться.
12. После окончания эксплуатации (по разным причинам) основное средство списывается. При этом отдельные детали, из которых оно состояло, могут поступать в запас.

### **Некоторые замечания**

Нашу конфигурацию нужно рассматривать как учебную, а не как коммерческую конфигурацию, учитывающую текущее законодательство. Наша задача проиллюстрировать те или иные возможности конфигурирования и встроенного языка, а не создание коммерческого продукта.

**Переходим непосредственно к реализации поставленной задачи, но начнем с рассмотрения общих и в то же время необходимых для работы вопросов.**





довательно, и программной части системы «1С:Предприятие» функционировать не может. Поэтому наличие доступа к программной части системы на компьютере конкретного пользователя является обязательным.

Запуская систему, пользователь должен обязательно работать с какой-либо информационной базой (далее — ИБ). ИБ содержит Конфигурацию, Базу Данных (БД), список пользователей и некоторые другие параметры. Конфигурация состоит из отдельных объектов.

Среди всех объектов, которые есть в конфигурации, имеются объекты, относящиеся, согласно документации, к прикладным объектам системы. Как правило, каждому из прикладных объектов в конфигурации соответствует конкретная таблица в БД.

Набор таких таблиц должен представлять собой единую логически целостную конструкцию, которая должна определяться конкретной моделью предметной области.

Такая модель должна содержать описание объектов предметной области, структур информационных массивов (таблиц БД) и алгоритмов их обработки.

Конфигурация (в терминах документации «1С:Предприятия») представляет собой совокупность описания структуры данных, алгоритмов обработки, печатных и экранных форм, а также другой информации, необходимой для автоматизации той или иной прикладной задачи.

Как же выглядит ИБ «1С:Предприятия»?

Если Вы установили с установочного диска типовую конфигурацию, то содержащий ее каталог будет выглядеть так, как это показано на рис. 1.2.

Для удобства пользователей ИБ сохраняется в одном файле — «1cv8.2CD» и включает Конфигурацию. Такой подход исключает разрушение связей внутри ИБ в процессе ее копирования пользователем.

Для того чтобы увидеть конфигурацию, нам необходимо будет запустить «1С:Предприятие».

### **1.1.2. Режимы запуска системы «1С:Предприятие»**

Система «1С:Предприятие» имеет два основных режима запуска — «Конфигуратор» и «1С:Предприятие».

Первый из этих режимов предназначен для создания ИБ, внесения изменений в ее Конфигурацию, а также для выполнения административных функций.

В режиме «1С:Предприятие» пользователь запускает Конфигурацию на выполнение, как бы «проигрывая» файл ИБ. При этом программная часть системы использует структуры, созданные на этапе

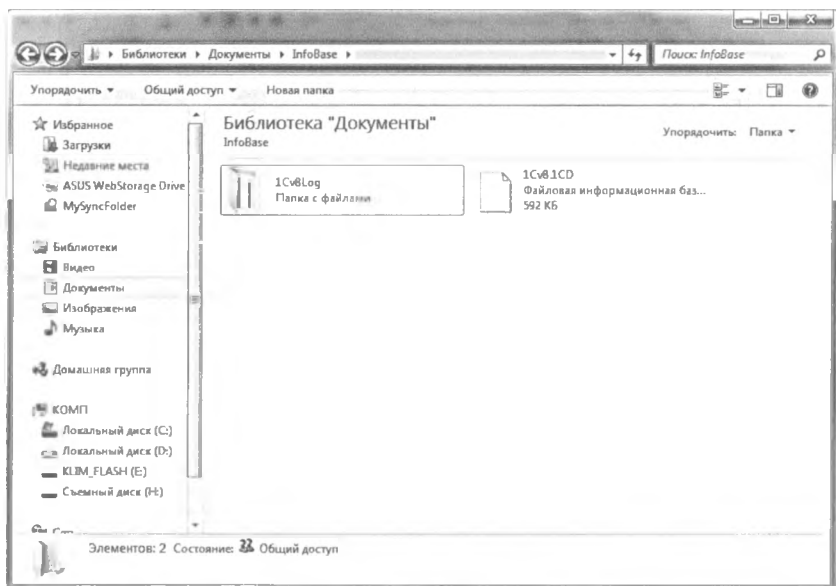


Рис. 1.2. Каталог ИБ

конфигурирования, предоставляя пользователю возможность заполнить их конкретными значениями.

Если на этапе конфигурирования с помощью встроенного языка определены соответствующие алгоритмы обработки, то в режиме «1С:Предприятие» пользователь будет вызывать их работу, давая системе соответствующие команды.

## 1.2. СОЗДАНИЕ НОВОЙ ИНФОРМАЦИОННОЙ БАЗЫ

Как было отмечено во Введении, мы будем строить свою учебную конфигурацию «с нуля». Давайте запустим систему. Мы будем считать, что у нас установлена только программная часть системы и нет ни одной ИБ.

Поэтому мы запустим систему в режиме «Конфигуратор». Для этого воспользуемся классическим способом запуска программ в MS Windows — через кнопку «Пуск» («Start»): Пуск — Программы — 1С: Предприятие 8.3 — Конфигуратор.

После чего на экран будет выведен диалог «Запуск 1С: Предприятия» (рис. 1.3).

В этом диалоге список «Информационные базы» содержит список зарегистрированных на данном компьютере Информационных

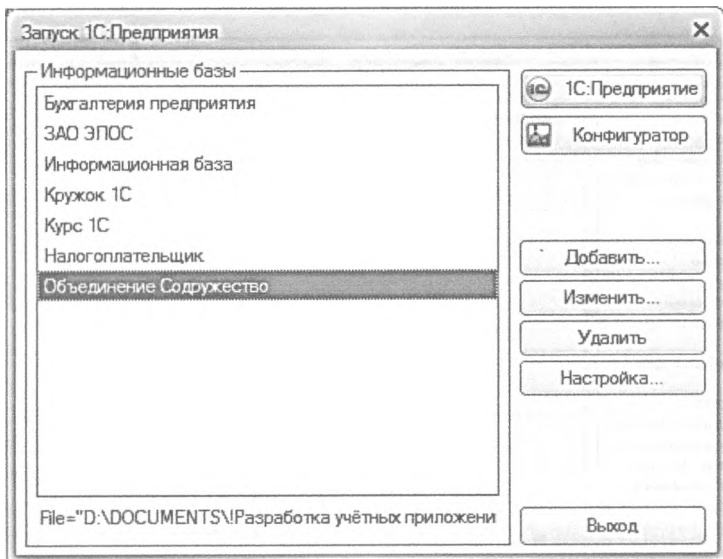


Рис. 1.3. Диалог «Запуск 1С:Предприятия»

Баз. По правилам работы системы каждая ИБ должна быть зарегистрирована. И неважно, существует ли реально ИБ или нет.

Каждая ИБ для файлового режима хранения данных характеризуется названием и каталогом, в котором она расположена. Процесс регистрации новой ИБ в «1С:Предприятии» версии 8.3 серьезно переработан по сравнению с версией 7.7, поэтому мы подробно его рассмотрим, выполнив упражнение 1.1.

### Упражнение № 1.1

1. Нажмите на кнопку «Добавить» в диалоге «Запуск 1С:Предприятие».
2. В диалоге «Добавление информационной базы» (рис. 1.4) поставьте переключатель в положение «Создание новой информационной базы».
3. Перейдите к следующему этапу регистрации, нажав на кнопку «Далее».
4. Для того чтобы создать пустую ИБ, необходимо на этом этапе установить переключатель в положение «Создание информационной базы без конфигурации для разработки новой конфигурации или загрузки выгруженной ранее информационной базы» (рис. 1.5).

Поясним те действия, которые мы только что проделали.

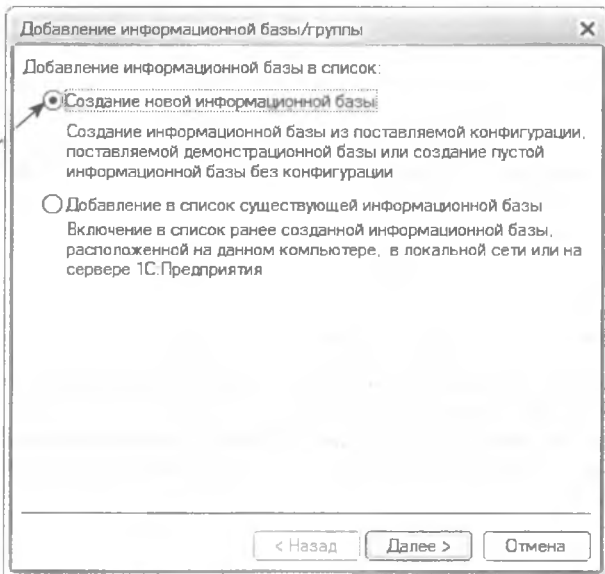


Рис. 1.4. Добавление новой информационной базы

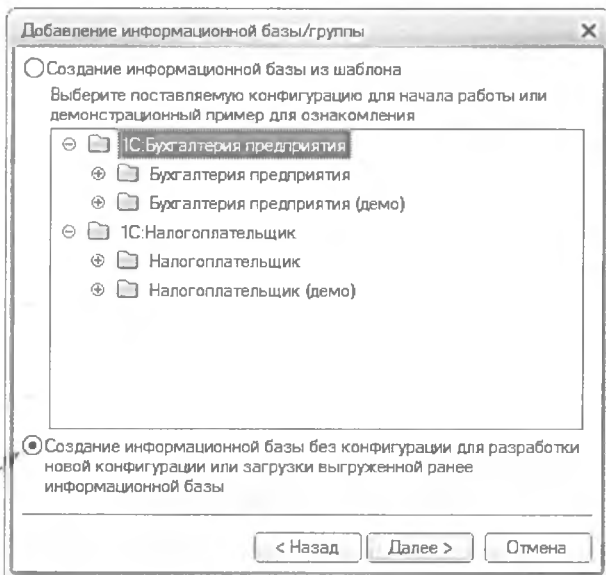


Рис. 1.5. Выбор способа создания ИБ

На первом этапе мы определили, что будем создавать новую информационную базу, а не регистрировать уже существующую. Если пойти по второму пути, то достаточно будет только указать, где находится ИБ.

Новой возможностью, которая появилась при создании новой информационной базы, является возможность создавать ИБ из шаблонов.

О том, как создать новый шаблон, описано в книге «1С:Предприятие 8. Руководство по установке и запуску».

### Упражнение № 1.1 (продолжение)

1. Нажмите на кнопку «Далее>» и укажите в качестве названия ИБ строку: «Прикладные объекты».
2. В качестве типа расположения ИБ оставьте переключатель в положении «На данном компьютере или на компьютере в локальной сети» (рис. 1.6).

Продолжим выполнение упражнения.

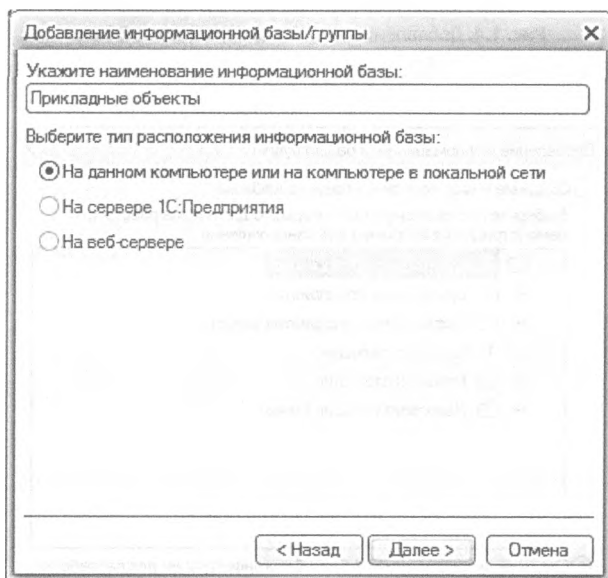


Рис. 1.6. Ввод названия и типа расположения ИБ

О том, как создавать ИБ, расположенные на сервере «1С:Предприятие», можно прочитать в документации к программе. В этом курсе мы не будем рассматривать этот тип расположения ИБ.

## Упражнение № 1.1 (продолжение)

На следующем этапе укажите (рис. 1.7):

Каталог информационной базы	C:\1cv8\PO\Base
Язык (Страна)	Русский (Россия)

Отметим, что выбор языка необходим, чтобы определить основной язык конфигурации. Нажмите на кнопку «Готово», чтобы завершить упражнение.

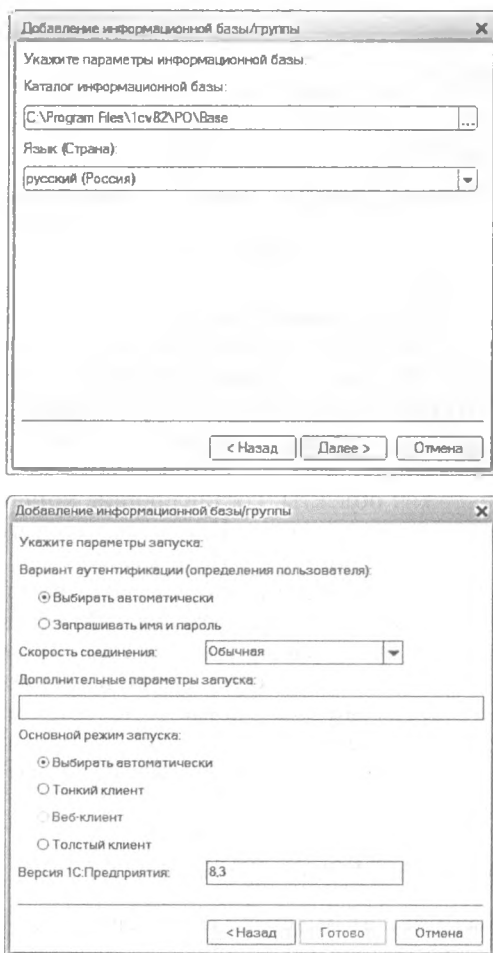


Рис. 1.7. Последний этап создания пустой ИБ

После регистрации необходимо запустить «Конфигуратор», используя одноименную кнопку.

### 1.3. ОКНО «КОНФИГУРАЦИЯ»

Окно программы «Конфигуратор» (рис. 1.8) похоже на многие другие программы MS Windows. Здесь есть меню, панели инструментов, рабочая область и строка состояния.

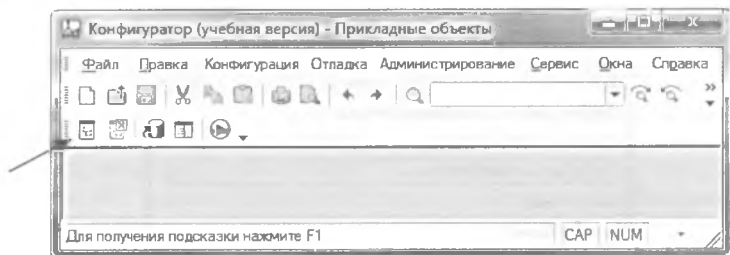



Рис. 1.8. Программа «Конфигуратор»

Основным окном, с которым Вам придется иметь дело на протяжении всего сеанса работы с Конфигуратором, — это окно «Конфигурация» (рис. 1.9). Его можно открыть, используя пункт меню «Конфигурация — Открыть конфигурацию», или нажав на кнопку  панели инструментов, которая выполняет те же функции, что и пункт меню.

Это окно содержит объекты, составляющие конфигурацию, которые отображаются в виде дерева. Каждая ветвь этого дерева предназначена для работы с объектами одного типа.

При разработке конфигурации «с нуля» в соответствующие ветви дерева мы будем добавлять новые объекты. При изложении материала мы в основном будем рассматривать *прикладные объекты* системы, полный список которых можно найти в документации.

Учитывая ограниченный объем данного пособия, мы более-менее подробно рассмотрим следующие типы прикладных объектов: *Константы*, *Справочники*, *Документы*, *Отчеты*, *Регистры сведений* и некоторые другие.

### 1.4. СВОЙСТВА ОБЪЕКТА КОНФИГУРАЦИИ

Каждый из объектов в этом дереве имеет свой набор свойств. Для того чтобы его увидеть, необходимо сначала выделить какой-либо из объектов в дереве, а затем нажать правую кнопку мыши. В открыв-



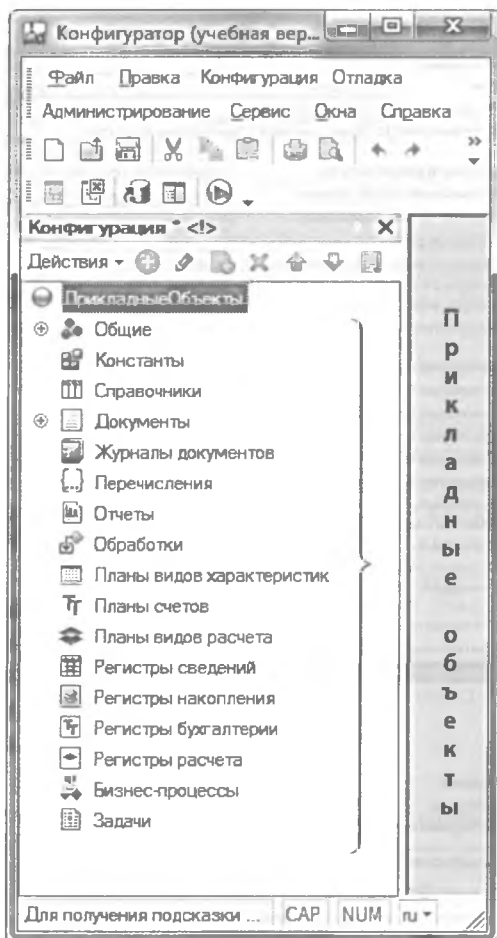


Рис. 1.9. Окно «Конфигурация»

шемся контекстном меню следует выбрать пункт «Свойства». Сразу после этого будет открыто окно «Свойства».

Основные приемы работы с окнами подробно описаны в документации по системе «1С:Предприятие». Здесь мы рассмотрим работу с этим окном на примере изменения свойств самой Конфигурации как объекта дерева.

Для того чтобы открыть свойства Конфигурации, необходимо на самой Конфигурации как на объекте сделать двойной клик мышью. После чего в Конфигураторе откроется окно «Свойства» (рис. 1.10).

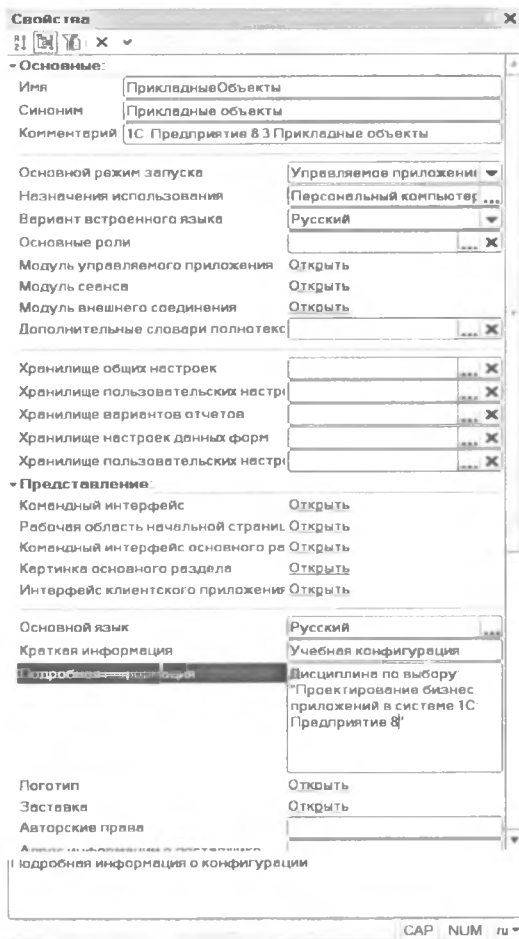


Рис. 1.10. Свойства объекта «Конфигурация»

Обратим внимание на то, что все свойства сгруппированы. Для данного объекта таких групп четыре — «Основные», «Представление», «Разработка» и «Справочная информация».

Заметим, что состав групп и свойств для каждого из объектов конфигурации был заранее определен еще на этапе разработки программной части системы «1С:Предприятие 8». Этот состав не может быть изменен пользователем (или настройщиком системы), но мы можем в Конфигураторе указать конкретные значения для каждого из свойств, определяя тем самым его поведение в режиме «1С:Предприятие».

### 1.4.1. Основные свойства

Для большинства объектов конфигурации в окне «Свойства» присутствует группа «Основные».

Как правило, в группе «Основные» присутствуют три свойства — это *Имя*, *Синоним* и *Комментарий*.

*Имя* — это краткое название объекта, которое записано по правилам записи идентификаторов в языках программирования. Оно должно начинаться с буквы (русской или латинской) или знака подчеркивания, может содержать буквы, цифры и знаки подчеркивания и не может содержать пробелов и спецсимволов. Имя является обязательным для заполнения свойством объекта — оно должно быть всегда. *Это имя Вы будете использовать при обращении к объекту из встроеного языка системы.*

*Синоним* — это другое название объекта, записанное, как правило, на языке конфигурации, который является *основным*. *Если кроме основного языка в конфигурации определены еще и другие языки, то синоним может быть записан индивидуально для каждого языка в отдельности.* Он может содержать пробелы и спецзнаки. Синоним записывают для того, чтобы в окнах «1С:Предприятия» вместо идентификатора пользователь мог видеть название объекта в читаемом для него виде.

*Комментарий* содержит дополнительную информацию об объекте. Вводится, если необходимо, разработчиком конфигурации.

#### Упражнение № 1.2

Укажите для объекта «Конфигурация» следующие основные свойства:

Свойство	Значение
Имя	ПрикладныеОбъекты
Синоним	Прикладные объекты
Комментарий	1С:Предприятие 8.3 Прикладные объекты

После выполнения данного упражнения в окне «Конфигурация» изменится название объекта.

В окне «Свойства» для данного объекта есть еще и другие свойства, которые расположены ниже Имени, Синонима и Комментария, — это свойства, специфичные для объекта данного типа. У других объектов конфигурации таких свойств нет.

Эти свойства мы рассмотрим позже, когда в этом возникнет необходимость.

### 1.4.2. Группа свойств «Представление»

Эта группа свойств есть у многих объектов в дереве конфигурации, но набор ее свойств очень сильно зависит от типа объекта, выделенного в дереве окна «Конфигурация».

Ниже мы рассмотрим некоторые специфические свойства объекта «Конфигурация» из этой группы свойств.

Свойство *Краткая информация* отображается системой при ее запуске в стартовом окне и может содержать любую информацию. Обычно здесь записывают строку текста, которая кратко характеризует назначение конфигурации.

В *Подробной информации* обычно записывается полное название конфигурации, дается информация об ее авторе. В это поле также можно поместить информацию о том, когда эта конфигурация была сделана.

*Логотип* и *Заставка* представляют собой рисунки, размещаемые внутри конфигурации для отображения, соответственно, логотипа конфигурации в диалоге «О программе» и заставки в стартовом окне режима «1С:Предприятие».

#### Упражнение № 1.3

Свойство	Значение
Краткая информация	Учебная конфигурация
Подробная информация	Дисциплина по выбору: «Проектирование бизнес приложений в системе «1С:Предприятие 8.3»

Заполните поля с краткой и подробной информацией вышеприведенными значениями.

Для выбора логотипа необходимо щелкнуть мышью ссылку **Открыть** и в открывшемся диалоге **Выбор картинки** следует с помощью кнопки **Выбрать из файла** указать на файл «Logo.bmp», который предварительно необходимо сохранить на Вашем компьютере. После этого диалог примет вид, изображенный на рис. 1.11. Теперь можно завершить ввод логотипа в конфигурацию путем нажатия на кнопку **OK**.

Аналогичным образом можно поступить и при выборе заставки. Для этого нужно опять нажать на ссылку **Открыть свойства Заставка**.

В этом случае Вам следует использовать файл «Slice.jpg», это позволяет «Установить прозрачный фон» с расширением .bmp, .ipg.

При выборе картинки существует возможность сделать какой-либо из цветов прозрачным. Поэтому, не закрывая диалог выбора заставки, нажмите на кнопку **Установить прозрачный фон** и выберите мышью белый фон картинки. Тогда диалог примет вид согласно рис. 1.12.

## Выбор картинки

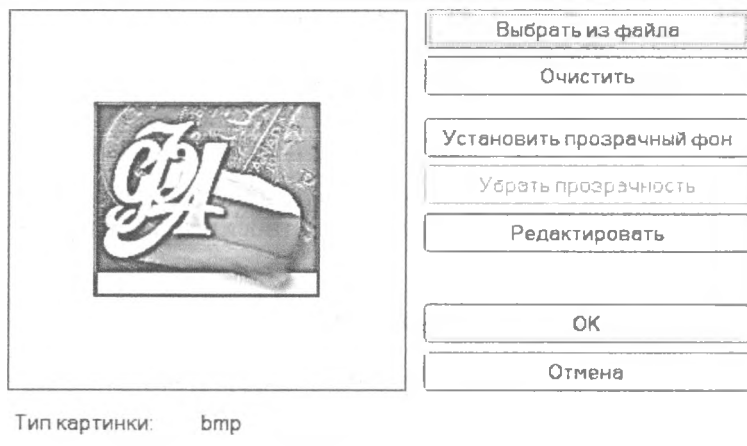


Рис. 1.11. Выбор логотипа для конфигурации

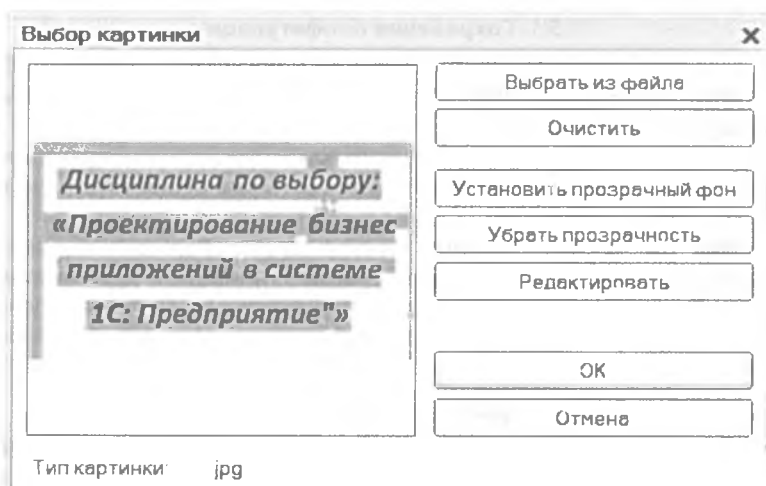


Рис. 1.12. Диалог выбора картинки после установки прозрачного фона

### 1.4.3. Группа свойств «Разработка»

Эта группа свойств позволяет идентифицировать разработчика Конфигурации, а также номер версии Конфигурации.

Заполните эти свойства, выполнив следующее упражнение.

Заполните следующие свойства конфигурации:

Свойство	Значение
Поставщик	Фирма "1С", 2012
Версия	Юнит 01

В каждом юните мы будем увеличивать номер версии на единицу, для того чтобы помнить, к какой теме относится рассматриваемая конфигурация.

## 1.5. ПУНКТ МЕНЮ «КОНФИГУРАЦИЯ»

Конфигуратор предоставляет некоторый набор команд, которые выполняют различные операции над конфигурацией.


С полным списком команд можно ознакомиться в книге «1С:Предприятие 8. Конфигурирование и администрирование». В этом разделе мы рассмотрим только минимально необходимый нам сейчас набор команд.

### 1.5.1. Сохранение Конфигурации


Когда Вы первый раз открываете в Конфигураторе окно «Конфигурация», система копирует содержимое *Конфигурации БД в Редактируемую Конфигурацию*.

Такой подход позволяет безопасно редактировать конфигурацию, не опасаясь случайным образом разрушить уже существующую настройку в ИБ.

Более того, все исправления производятся в оперативной памяти.

Поэтому система имеет команду «Конфигурация —  Сохранить конфигурацию», которая позволяет периодически сохранять редактируемую конфигурацию, не производя реструктуризацию информации в ИБ.

### 1.5.2. Обновление Конфигурации

Сохранение конфигурации не вызовет немедленных изменений в таблицах ИБ. Чтобы такие изменения произошли, необходимо выполнить пункт меню «Конфигурация — Обновить конфигурацию базы данных» или нажать на кнопку  панели инструментов. Редактируемая конфигурация будет помещена в ИБ.

После этого Конфигуратор произведет обновление Конфигурации. Поскольку никаких новых таблиц в ИБ в результате наших действий не возникло, то и никаких дополнительных сообщений Конфигуратор не выдаст.

### 1.5.3. Сохранение Конфигурации в файл

Так как конфигурация расположена внутри ИБ, то Конфигуратор позволяет извлечь ее из ИБ в файл, который может располагаться где угодно.

#### Упражнение № 1.5

---

---

Предварительно обновив Конфигурацию в ИБ, сохраните ее в отдельном файле: **C:\PO\Юнит1.cf**

---

---

## 1.6. АДМИНИСТРИРОВАНИЕ

Из возможностей администрирования системы в этом разделе мы рассмотрим только возможность создания резервной копии ИБ.

В процессе работы с ИБ, как в процессе разработки конфигурации, так и при эксплуатации пользователем уже готового решения, рекомендуется делать резервную копию всей ИБ.

За создание архива отвечает команда «Администрирование — Выгрузить информационную базу...», которая позволяет создать файл с расширением «\*.dt».

#### Упражнение № 1.6

---

---


Создайте резервную копию ИБ в файле: **C:\PO\Юнит1.dt**

---

---

## 1.7. ЗАПУСК В РЕЖИМЕ «1С:ПРЕДПРИЯТИЕ»

После обновления Конфигурации ее работу необходимо проверить в режиме «1С:Предприятие».

Сделать это можно разными способами. Но при разработке конфигурации наиболее эффективным способом с точки зрения поиска ошибок следует считать запуск режима «1С:Предприятие» под встроенным в Конфигуратор Отладчиком. Для этого достаточно выбрать пункт меню «Отладка — Начать отладку», или нажать на горячую клавишу F5, или нажать на пиктограмму .

Обратите внимание, что при запуске в стартовом окне приложения на некоторый момент времени появится наша заставка (рис. 1.13), а уже затем появится основное окно приложения в режиме «1С:Предприятие».

Обратите также внимание и на то, что под нашей заставкой «проевечивает» системная заставка «1С:Предприятие».

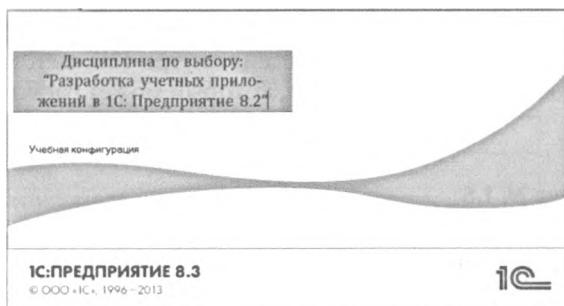


Рис. 1.13. Заставка учебной конфигурации

После того как программная часть системы загрузилась и открыла Информационную базу, то на экране появится окно системы в режиме «1С:Предприятие».

Если в основном меню этого приложения выбрать пункт «Справка — О программе», то мы увидим информацию, которую мы вводили в группу свойств «Представление» и «Разработка» (рис. 1.14).

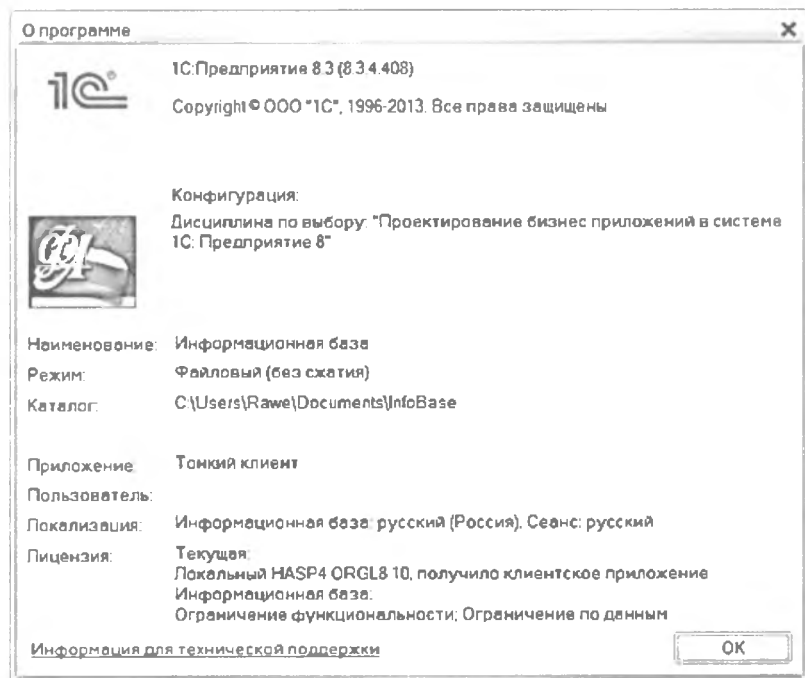


Рис. 1.14. Диалог «О программе» в режиме «1С:Предприятие»



Обратите внимание на то, что номер версии конфигурации отображается в скобках после ее названия.

## **1.8. ЧТО МЫ УЗНАЛИ**

В этой главе мы кратко ознакомились с концепциями, положенными в основу системы «ИС:Предприятие 8». Дали некоторые определения.

Разобрали регистрацию новой ИБ, а также связь этого понятия с Конфигурацией и программной частью системы. Рассмотрели создание новой ИБ «с нуля» и редактирование свойств объектов Конфигурации.

Мы создали новую информационную базу и произвели настройку свойств ее Конфигурации.

- ❑ Константы в системе «1С:Предприятие» относятся к *прикладным объектам* и предназначены для хранения постоянной или условно-постоянной информации. Важным является то, что *одна константа способна хранить только одно значение*.
- ❑ Константы создаются в Конфигураторе. Обращение к ним из встроенного языка осуществляется через соответствующее свойство глобального контекста.

### 2.1. СОЗДАНИЕ КОНСТАНТЫ

Константы располагаются в отдельной ветви дерева окна «Конфигурация». Для того чтобы создать новую константу, необходимо, используя правую кнопку мыши на ветви «Константы» дерева окна «Конфигурация», выбрать пункт выпадающего меню «Добавить» (рис. 2.1).

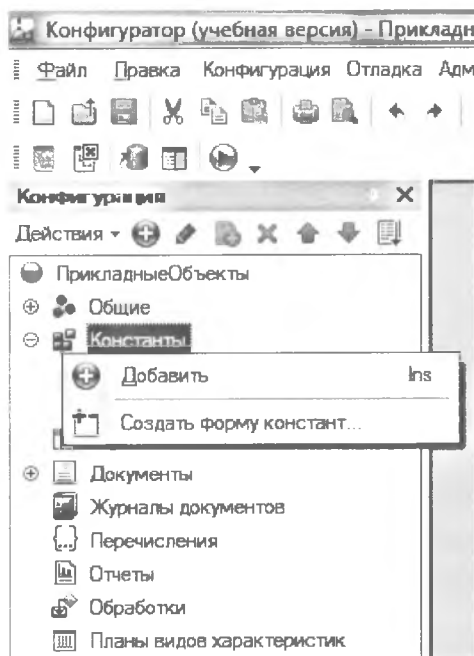


Рис. 2.1. Создание новой константы

В ответ на эту команду конфигуратор создаст в дереве новый объект — константу с именем «Константа1» и откроет окно с ее свойствами (рис. 2.2).

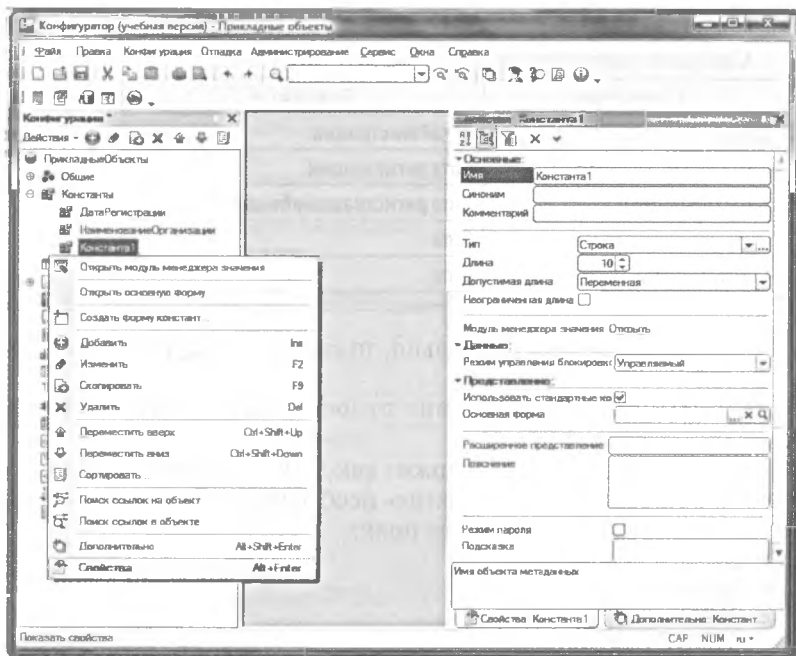


Рис. 2.2. Свойства константы

## 2.2. СВОЙСТВА КОНСТАНТЫ

Для новой константы необходимо настроить ее свойства.

Мы видим на рис. 2.2, что константы имеют две группы свойств — «Основные» и «Тип данных».

Группа «Основные» содержит свойства «Имя», «Синоним» и «Комментарий». Эти свойства мы подробно рассматривали в первой главе и здесь повторно их разбирать не будем.

А вот на группе «Тип данных» мы остановимся подробнее.

Поскольку значение константы сохраняется в ИБ, то для нее нужно обязательно указать тип значения, которое может в ней храниться. Это свойство должно быть обязательно выбрано.

Начало выпадающего списка из свойства «Тип» содержит примитивные типы данных: «Число», «Строка», «Дата», «Булево»...

Для закрепления только что рассмотренного материала и в соответствии с постановкой задачи создайте константу со следующими свойствами.

### Упражнение № 2.1

Создайте константу со следующими свойствами:

Свойство	Значение
Имя	<b>ДатаРегистрации</b>
Синоним	<b>Дата регистрации</b>
Комментарий	<b>Дата регистрации объединения</b>
Тип	<b>Дата</b>
Состав даты	<b>Дата</b>

Если Вы все сделали правильно, то свойства будут теми же, что и на рис. 2.3.

Здесь мы сделаем замечание относительно свойства «Состав даты».

Поскольку тип «Дата» содержит как дату, так и время, то пользователю в режиме «1С:Предприятие» необходим один из трех возможных способов заполнения этого поля:

- ✓ Дата.
- ✓ Время.
- ✓ Дата и время.

В соответствии с постановкой задачи создайте следующую константу самостоятельно.

### Упражнение № 2.2

Создайте константу и заполните ее свойства:

Свойство	Значение
Имя	<b>НаименованиеОрганизации</b>
Синоним	<b>Наименование организации</b>
Комментарий	<b>Полное наименование организации</b>
Тип	<b>Строка</b>

Какую длину для строки Вы выберете и почему?  
Сохраните и обновите конфигурацию.

Обратите внимание, что когда мы дали команду на обновление конфигурации, Конфигуратор проанализировал сделанные изменения и *выделил место хранения в ИБ для новых объектов.*

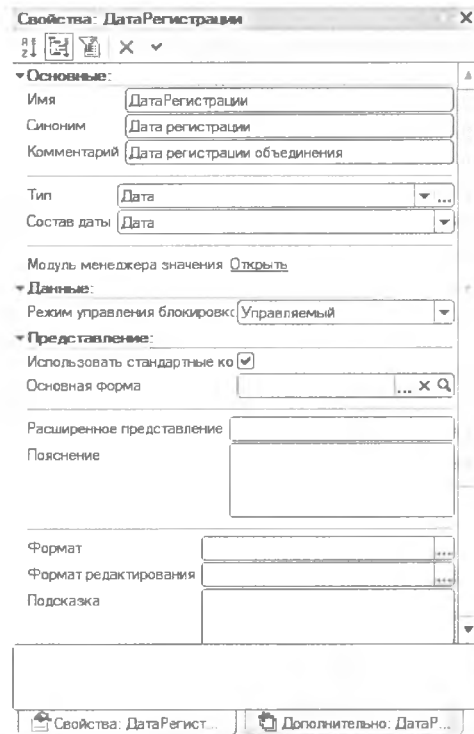


Рис. 2.3. Свойства константы «ДатаРегистрации»

Именно поэтому мы должны обязательно для строк указывать их длину, ибо Конфигуратор должен знать, сколько памяти выделить под хранение значений реквизитов этого типа.

Впрочем, последнее замечание относится и к другим типам, которые мы можем выбирать из выпадающего списка «Тип», — Конфигуратор всегда выделяет память в ИБ под хранение данных. Но в отличие от классических СУБД (Систем Управления БД), «ИС:Предприятие» позволяет размышлять в терминах предметной области, не задумываясь над тем, какие таблицы и какие поля выделить. Поэтому *на специалисте, занимающемся разработкой конфигурации, лежит обязанность правильно выбирать типы хранения данных в ИБ.*

*Обратите внимание еще на один важный момент — в Конфигураторе мы только разрабатываем структуру хранения, а конкретные значения пользователь вводит в режиме «ИС:Предприятие».*

А раз так, то мы должны предоставить пользователю некоторую форму диалога для редактирования значений констант.

## 2.3. ПОДСИСТЕМЫ

Подсистемы — основные элементы для построения интерфейса «1С:Предприятие». Поэтому первое, с чего следует начинать разработку конфигурации, — это проектирование состава подсистем.

При этом перед разработчиком стоит важная и ответственная задача — тщательно продумать состав Подсистем, и затем аккуратно и осмысленно привязать к подсистемам те объекты конфигурации, которые он будет создавать. В простых прикладных решениях подсистемы можно не использовать.

Подсистемы позволяют выделить в конфигурации функциональные части, на которые логически разбивается создаваемое прикладное решение.

Эти объекты располагаются в ветке объектов «Общие» и позволяют строить древовидную структуру, состоящую из подсистем и подчиненных подсистем, например, как на рис. 2.4.

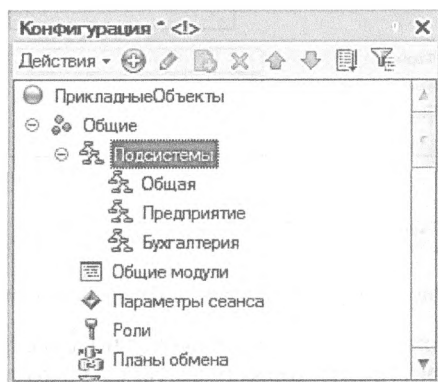


Рис. 2.4. Ветка «Общие»

Подсистемы верхнего уровня являются основными элементами интерфейса, так как образуют разделы прикладного решения.

Каждый объект конфигурации может быть включен в одну или сразу несколько подсистем, в составе которых он будет отображаться.

Забегая вперед, скажем, что с помощью подсистем, используя видимость по ролям, можно предоставить пользователю удобный и функциональный интерфейс, не содержащий лишних элементов. Например, кладовщик должен иметь возможность принять и выдать товар, и ему совсем не нужно видеть все, что относится к области бухгалтерского учета и оказанию услуг.

Таким образом, наличие подсистем определяет структуру прикладного решения, организует весь пользовательский интерфейс,

позволяет «рассортировать» различные документы, справочники и отчеты по логически связанным с ними разделам, в которых пользователю будет проще их найти и удобнее с ними работать. При этом каждому конкретному пользователю будут видны лишь те разделы, т.е. та функциональность прикладного решения, которые ему нужны в процессе работы.

Даже в такой небольшой конфигурации, как наша, можно выделить несколько функциональных частей, представляющих собой отдельные предметные области.

Так, можно выделить в отдельную подсистему все, что имеет отношение к бухгалтерскому учету.

### 2.3.1. Добавление подсистемы в режиме Конфигуратор

#### Упражнение № 2.3

Создайте в нашей конфигурации три новых объекта конфигурации подсистемы, которые будут иметь имена:

- Общая.
- Предприятие.
- Бухгалтерия.

Закроем приложение и вернемся в Конфигуратор. Чтобы создать новые подсистемы, раскроем ветвь «Общие» в дереве объектов конфигурации, нажав на «+» слева от нее.

Затем выделим ветвь «Подсистемы», вызовем ее контекстное меню и выберем пункт «Добавить» или нажмем соответствующую кнопку в командной панели окна конфигурации (рис. 2.5).

После этого система откроет окно редактирования объекта конфигурации.

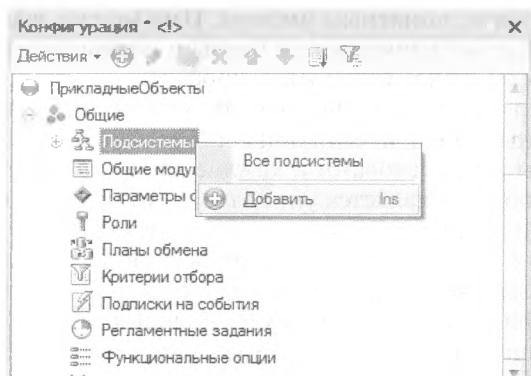


Рис. 2.5. Добавление подсистемы в режиме Конфигуратор

Оно предназначено специально для сложных объектов конфигурации и позволяет путем выполнения последовательных действий быстро создавать такие объекты.

Зададим имя подсистемы — «Общая», платформа автоматически создаст синоним — на основании имени «Общая» (рис. 2.6).

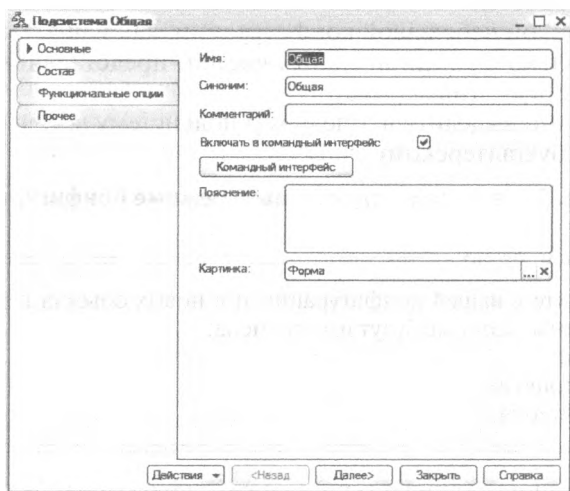


Рис. 2.6. Окно редактирования объекта конфигурации.

### 2.3.2. Имя и синоним объекта конфигурации

Имя является основным свойством любого объекта конфигурации. При создании нового объекта система автоматически присваивает ему некоторое имя.

Можно использовать имя, присвоенное системой, но лучше заменить его своим, понятным именем. Имя можно задавать любое, главное, чтобы оно начиналось с буквы и не содержало некоторых специальных символов (например, пробел).

Для удобства чтения конфигурации принято составлять интуитивно понятные имена и, если они состоят из нескольких слов, удалять пробелы между словами и каждое слово начинать с большой буквы. Имя объекта является уникальным и служит для обращения к свойствам и методам объекта на встроенном языке.

Свойство *Синоним* также есть у любого объекта конфигурации. Оно предназначено для хранения «альтернативного» наименования объекта конфигурации, которое будет использовано в элементах интерфейса нашей программы, т.е. будет показано пользователю. Поэтому на синоним практически нет никаких ограничений, и его можно задавать в привычном для человека виде.



В целях усовершенствования интерфейса приложения мы можем также задать картинку для отображения подсистемы. Нажмем кнопку выбора [...] в поле «Картинка» (см. рис. 2.6). В окне выбора картинки добавим картинку в список на закладке «Из конфигурации». Для этого нажмем кнопку «Добавить» (рис. 2.7) и в открывшемся окне «Общая картинка», используя кнопку «Выбрать из файла», находим требуемую картинку (рис. 2.8).

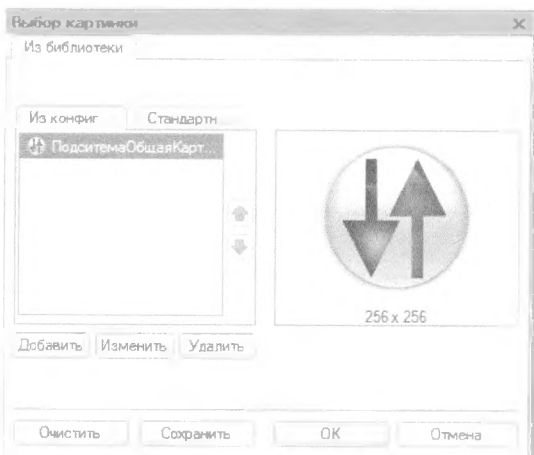


Рис. 2.7. Окно выбора картинки

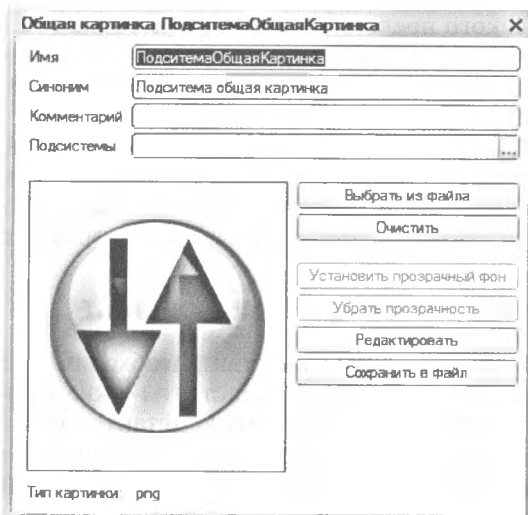


Рис. 2.8. Окно «Общая картинка»

По аналогии создайте подсистемы «Предприятие» и «Бухгалтерия».

В конечном итоге в режиме «1С:Предприятие» должен появиться интерфейс, представленный на рис. 2.9.

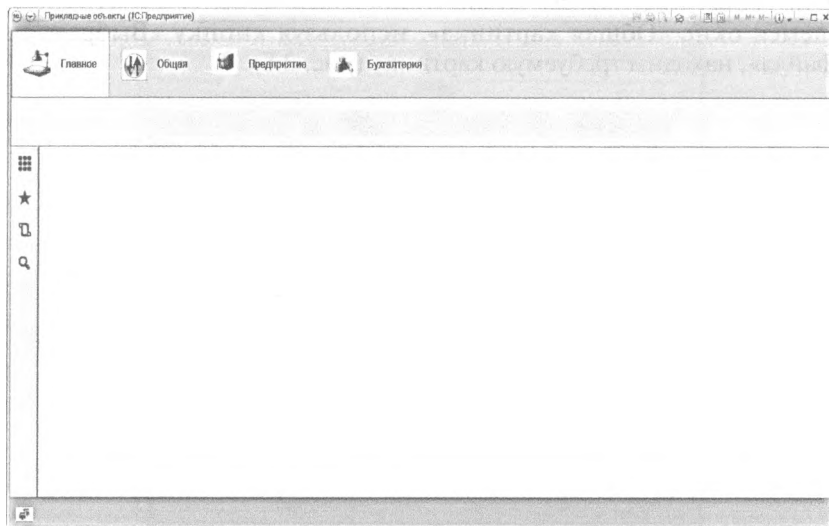


Рис. 2.9. Начальный интерфейс приложения

Подсистемы в управляемом приложении используются для удобства и логического представления информации. Тем не менее для первоначального изучения алгоритма разработки приложений прием более простую и хорошо отработанную методику создания «обычного» приложения.

Для этого в свойствах конфигурации значение параметра «Основной режим запуска» замените на «Обычное приложение» (рис. 2.10).

В дальнейшем для удобства изложения материала используется контекст «Приложение» вместо «Обычное приложение».

## 2.4. СОЗДАНИЕ ОСНОВНОЙ ФОРМЫ КОНСТАНТ

Для того чтобы создать форму констант, необходимо, используя правую кнопку мыши на поддереве «Константы» окна «Конфигурация», выбрать пункт меню «Создать форму констант...» (рис. 2.11).

При выборе этого пункта вызывается специальный «Конструктор общих форм» (рис. 2.12), который позволяет быстро построить форму, не вдаваясь на этом этапе в детали проектирования и автоматизируя многие ручные действия.

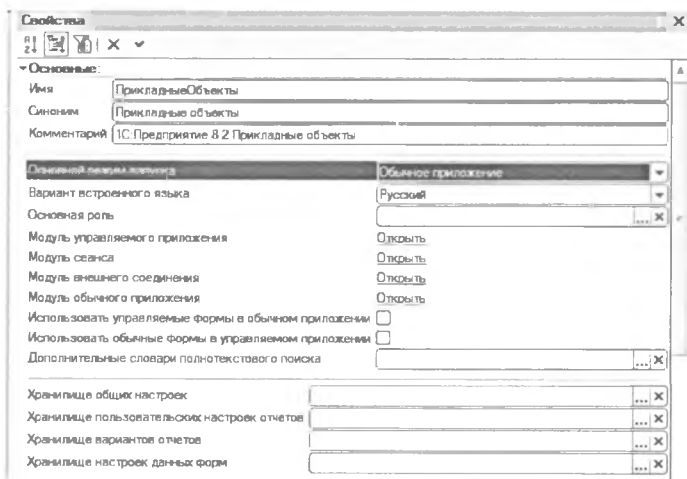


Рис. 2.10. Окно свойств приложения

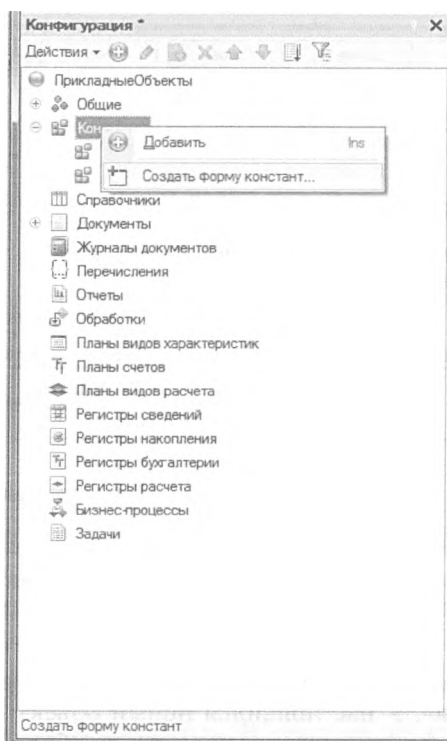


Рис. 2.11. Добавление новой формы констант

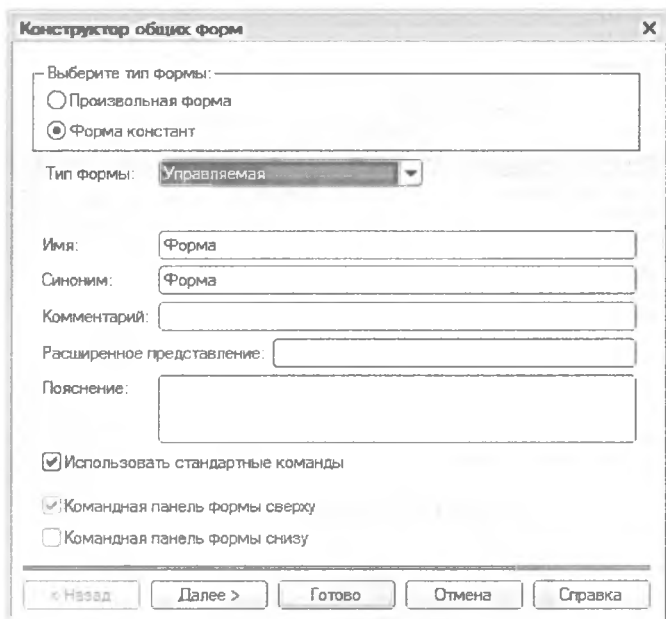


Рис. 2.12. Окно специального «Конструктора общих форм»

## Упражнение № 2.4

Добавьте новую форму констант в Конструкторе и укажите следующие значения в полях его диалога:

Свойство	Значение
Выберете тип формы	<b>Форма констант</b>
Тип формы	<b>Управляемая</b>
Имя	<b>Форма</b>
Синоним	<b>Форма</b>
Комментарий	<b>Форма констант</b>
Использовать стандартные команды	<input checked="" type="checkbox"/>

Нажмите на кнопку «Готово», для того чтобы Конструктор закончил построение формы и открыл ее для редактирования (рис. 2.13).

Прежде всего обратим внимание на то, что изменилось в дереве «Конфигурация». У нас появился новый объект — типа «Общая форма», который расположен в поддереве «Общие». Осталось сделать эту форму основной.

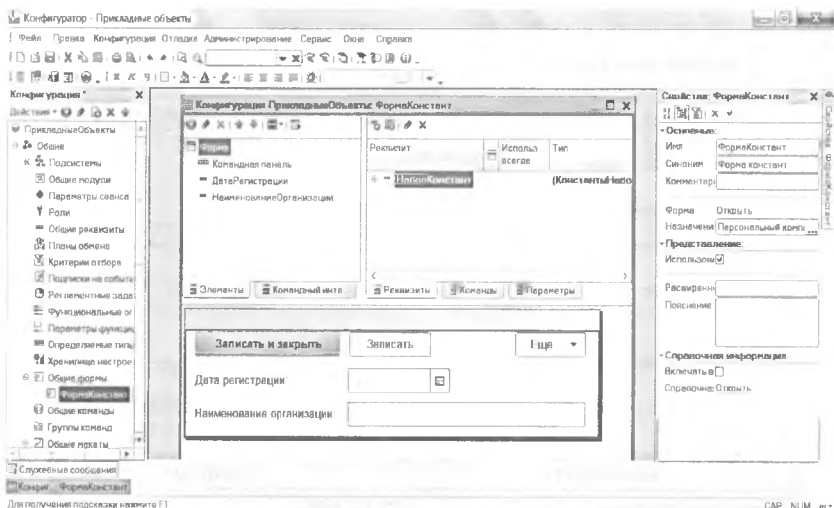


Рис. 2.13. Результат работы Конструктора формы

## Упражнение № 2.5

В свойствах конфигурации к параметру «Основная форма констант» подключите только что созданную управляемую форму «Форма». Для этого нажмите на кнопку ... и в появившемся диалоговом окне (рис. 2.14) выберите искомую форму.

Поскольку мы внесли в конфигурацию много изменений, то необходимо установить номер версии в свойствах конфигурации в значении «Юнит 2».

Сохраните и обновите базу данных.

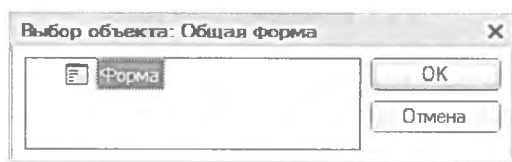


Рис. 2.14. Диалоговое окно выбора объекта

Теперь, после такой серьезной подготовительной работы мы сможем запустить конфигурацию в режиме «1С:Предприятие».

Запустим «1С:Предприятие» и в этом режиме выберем пункт меню «Операции — Константы». В ответ на эту команду система откроет только что спроектированную форму (рис. 2.15).

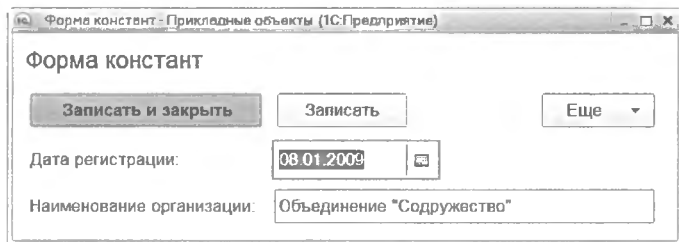


Рис. 2.15. Форма констант в режиме «1С:Предприятие»

## Упражнение № 2.6

Заполните в режиме «1С:Предприятие» константы следующими значениями:

Константа	Значение
Дата регистрации:	08.01.2009
Наименование организации:	Объединение «Содружество»

Нажмите на кнопку «Записать и закрыть», для того чтобы сохранить введенные значения в ИБ.

## 2.5. ОБРАЩЕНИЕ К ЗНАЧЕНИЯМ КОНСТАНТ ИЗ ВСТРОЕННОГО ЯЗЫКА

Для того чтобы обратиться к значениям констант, необходимо воспользоваться *свойством глобального контекста* Константы. Тип данных этого свойства — **КонстантаМенеджер**, который является, по сути, коллекцией значений.

### 2.5.1. Менеджер константы

Обратиться к конкретной константе можно, записав следующее выражение: **Константы.Имя**.

Такая запись дает нам доступ к объекту типа **КонстантаМенеджер**. Этот объект имеет два метода:

**Получить();**

**Установить(<Значение>);**

Эти методы позволяют получить и записать значение константы.

Приведем примеры обращения к константам, которые разместим в отдельной процедуре **ПримерыОбращенияККонстантам()** Модуля обычного приложения.

Для того чтобы открыть Модуль обычного приложения, нужно сначала выделить объект «Конфигурация» и, используя правую

кнопку мыши, выбрать пункт меню «Открыть модуль обычного приложения».

Вызов процедуры **ПримерыОбращенияККонстантам()** мы расположим в теле процедуры **ПриНачалеРаботыСистемы()**, для того чтобы ее вызов происходил при каждом запуске Конфигурации в режиме «IS: Предприятие».

## ЛИСТИНГ Ю\_2.1

```
// Модуль приложения
//ПримерыОбращенияККонстантам() — содержит примеры //обращения к
Константам
Процедура ПримерыОбращенияККонстантам()
// Если это первый запуск конфигурации, то константа
// содержит в качестве значения 1 января 0001 года.
Если Константы.ДатаРегистрации.Получить()='00010101' Тогда
// Тогда установим 1 января текущего года
Константы.ДатаРегистрации.Установить (НачалоГода(ТекущаяДата()));
КонецЕсли;
// распечатаем значения констант:
Сообщить(«Дата регистрации:»+Константы.ДатаРегистрации.Получить()+«.»);
Сообщить(«Наименование организации:»+Константы.НаименованиеОргани-
зации.Получить()+«.»);
КонецПроцедуры
```

```
Процедура ПриНачалеРаботыСистемы()
//Примеры:
ПримерыОбращенияККонстантам();
КонецПроцедуры
```

К этому тексту мы сделаем одно пояснение.

Часто бывает необходимо понять, ЧТО ввел пользователь в какой-то реквизит — там что-то есть или там «пусто»?

Когда Вы в Конфигураторе создаете какой-либо реквизит, то система при реструктуризации ИБ не только выделяет место, но и записывает в него начальное значение:

- ✓ для типа Булево, это будет Ложь;
- ✓ для чисел — это 0;
- ✓ для строк — это «» или строка, содержащая все пробелы;
- ✓ для даты — это «точка отсчета»: '00010101'.

Поэтому условие:

**Если Константы.ДатаРегистрации.Получить()='00010101' Тогда**

позволяет определить, что поле данных содержит начальное значение, которое могло быть туда записано системой или пользователем.

### 2.5.2. Набор констант

В процедуре, которая содержит примеры обращения к константам, мы использовали три операции считывания (метод `.Получить(...)`) и одну операцию записи (метод `.Установить(...)`).

Несмотря на то что наш пример учебный, такой подход (считывание и запись по одному значению) нельзя признать оптимальным. Поэтому во встроенном языке существуют способы, позволяющие считывать и записывать *набор значений*, выполняя необходимые действия по чтению и записи за одну операцию обращения к ИБ.

Сначала набор необходимо будет создать:

```
НаборКонстант = Константы.СоздатьНабор(«Список констант»);
```

При этом метод `СоздатьНабор()` создаст объект типа «КонстантыНабор». Свойствами этого объекта будут все имена констант, которые вошли в список констант (перечисляются через запятую), переданный методу `СоздатьНабор()` в качестве параметра.

В основном у этого объекта используются два метода:

```
Прочитать();
```

```
Записать();
```

Назначение этих методов понятно из их названия.

## 2.6. САМОСТОЯТЕЛЬНАЯ РАБОТА

Для закрепления только что рассмотренного материала выполните самостоятельно следующее упражнение.

### Упражнение № 2.7

---

Перепишите самостоятельно текст процедуры `ПримерыОбращенияККонстантам()`, используя для обращения к значениям набор констант.

Какие имена констант Вы внесете в список для метода: `СоздатьНабор()`?

Сколько раз Вы будете применять метод `.Прочитать()`?

Изменится ли способ записи обращения к константам?

Сколько раз Вы будете применять метод `.Записать()`?

---

Для надежности приводим текст данной процедуры (сравните со своим вариантом).

### ЛИСТИНГ Ю\_2.2

```
//ПримерыОбращенияККонстантам()
```

```
//- содержит примеры обращения к Константам
```

```
Процедура ПримерыОбращенияККонстантам()
```

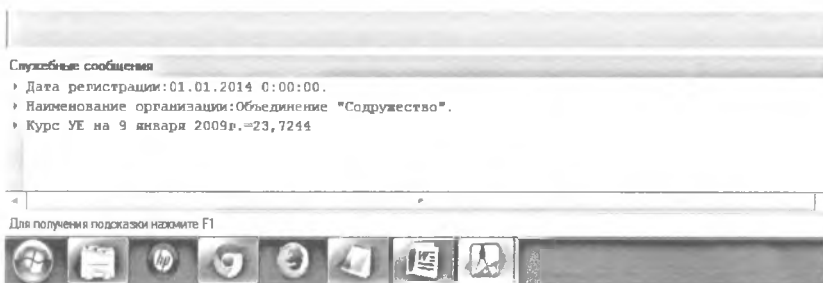
```
    //Создадим набор констант:
```



```

НаборКонстант=Константы.СоздатьНабор
(«ДатаРегистрации,НаименованиеОрганизации»);
НаборКонстант.Прочитать();//Прочитаем набор
// Если это первый запуск конфигурации, то константа
// содержит в качестве значения 1 января 0001 года.
Если НаборКонстант.ДатаРегистрации='00010101' Тогда
    // Тогда установим 1 января текущего года
НаборКонстант.ДатаРегистрации=НачалоГода(ТекущаяДата());
НаборКонстант.Записать();//Запишем набор
КонецЕсли;
// распечатаем значения констант:
Сообщить(«Дата регистрации:»+
НаборКонстант.ДатаРегистрации+«.»);
Сообщить(«Наименование организации:»+
НаборКонстант.НаименованиеОрганизации+«.»);
КонецПроцедуры

```



## 2.7. ЧТО МЫ УЗНАЛИ

В этой главе мы познакомились с прикладным объектом «Константы», который предназначен для сохранения в ИБ постоянной или условно-постоянной информации. Мы создали две константы и сконструировали в Конфигураторе форму для ввода их значений.

Мы также рассмотрели особенности обращения к значениям констант из встроенного языка системы, применив объекты типа **КонстантаМенеджер** и **КонстантыНабор**.

Объекты последнего типа применяются для оптимизации числа обращений к ИБ по *чтению* (записи), позволяя выполнить действия над необходимыми значениями в оперативной памяти.

Регистры сведений так же, как и константы, работу с которыми мы рассмотрели в предыдущем юните, относятся к *прикладным объектам* системы.

Константа способна хранить только одно значение. Что делать, если значение, которое необходимо сохранять в ИБ, должно изменяться с течением времени (быть *развернуто* по времени) или если оно (значение) зависит от ряда факторов?

В таких случаях можно говорить о том, что нам необходима некоторая *многомерная* структура хранения, способная сохранять различные сведения в зависимости от конкретных значений своих измерений.

Такой структурой хранения в «1С: Предприятии» являются регистры сведений, которые, выражаясь математическим языком, позволяют описать зависимость значения некоторой функции от своих аргументов в табличной форме.

В постановке задачи (п. 3) сказано, что «...необходимо пересчитывать рублевые суммы в условные единицы (УЕ)». Тогда для выполнения подобных операций необходимо помнить соответствующие значения курса. И зависимость этого курса от даты может быть выражена таблицей (рис. 3.1).

Объединение "Содружество"	
История курса УЕ	
Дата	Значение
28.12.2007	25.2600
01.01.2008	24.8500
01.01.2009	23.7844
31.01.2009	23.6222
02.03.2009	23.5729
01.04.2009	23.3801
01.05.2009	23.1021
31.05.2009	23.7090
30.06.2009	23.3483
30.07.2009	23.2450
18.08.2009	23.3532

Рис. 3.1. Значения курса УЕ

Из этой таблицы мы видим, что информация, представленная в колонке «Значение», *развернута* во времени.

Поэтому мы можем приступить к созданию соответствующего регистра сведений.

Регистры сведений располагаются в отдельной ветви дерева окна «Конфигурация». Для того чтобы создать новый регистр, необходимо, используя правую кнопку мыши, на ветви «Регистры сведений» дерева окна «Конфигурация» выбрать пункт выпадающего меню «Добавить».

### 3.1. ОКНО РЕДАКТИРОВАНИЯ РЕГИСТРА СВЕДЕНИЙ

Сразу после этого Конфигуратор создаст в дереве новый объект «РегистрСведений1» и откроет специальное *окно редактирования* для его настройки (рис. 3.2).

Сам регистр сведений — достаточно сложный объект, он содержит ряд подчиненных объектов. Кроме того, он должен быть связан с остальной конфигурацией в единое целое. Поэтому закладки слева позволяют не только настроить свойства и состав самого регистра, но и связать его с другими объектами в конфигурации.

Кнопка «Далее» позволяет последовательно пройти по всем этапам настройки регистра.

В этом юните мы рассмотрим только этапы «Подсистемы», «Основные», «Данные» и «Формы», а изложение остальных этапов в настройке регистра сведений мы рассмотрим в последующих юнитах.

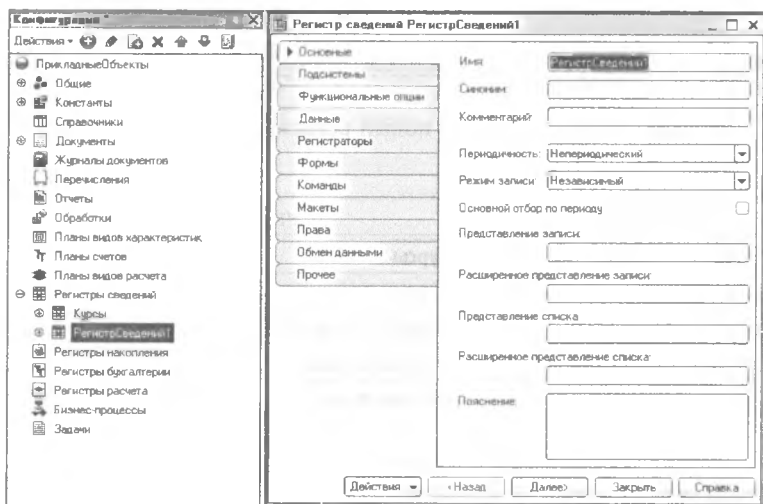


Рис. 3.2. Окно настройки Регистра сведений

### 3.1.1. Закладка «Основные»

На этой закладке собраны основные свойства регистра сведений, определение которых необходимо сделать до начала настройки остальных свойств.

Так же, как и у других объектов, у регистров сведений есть свойства: *Имя*, *Синоним* и *Комментарий*, которые мы уже рассматривали с вами ранее. Новыми для нас будут свойства *Периодичность* и *Режим записи*.

**Периодичность.** Это свойство позволяет определить, с какой степенью детализации необходимо помещать в регистр значения в зависимости от времени. Возможны следующие значения:

- ✓ Непериодический;
- ✓ В пределах секунды;
- ✓ В пределах дня;
- ✓ В пределах месяца;
- ✓ В пределах квартала;
- ✓ В пределах года.

Если регистр сделать непериодическим, то у него не будет разреза (изменения) «Период».

**Режим записи.** Это свойство определяет, каким образом будут вноситься изменения в регистр: вручную или документами.

Пока мы будем использовать только *независимый режим записи*, так как в нашей конфигурации еще нет ни одного документа.

Для самостоятельного освоения материала выполните следующее упражнение.

#### Упражнение № 3.1

---

Создайте в конфигурации новый регистр сведений со следующими свойствами:

Свойство	Значение
Имя	<b>Курсы</b>
Синоним	<b>Курсы</b>
Комментарий	<b>Содержит курсы УЕ</b>
Периодичность	<b>В пределах дня</b>
Режим записи	<b>Независимый</b>

Перейдите к закладке «Данные».

---

### 3.1.2. Закладка «Данные»

На этой закладке (рис. 3.3) мы должны определить структуру хранения нашего регистра.

Измерения представляют собой, по сути, аргументы функции, задаваемой в табличной форме.

Если регистр развернут по времени, то он уже имеет одно измерение — «Период». И такие регистры сведений называются *периодическими*.

Отметим, что в самом общем случае у неперіодических регистров измерения могут отсутствовать.

Ресурсы будут содержать искомые значения в зависимости от конкретных значений измерений.

Так же, как и в случае с измерениями, отметим, что в регистре могут отсутствовать и ресурсы.

Согласно таблице, которая изображена на рис. 3.1, нам потребуется:

- ✓ одно измерение — «Период» (уже есть!);
- ✓ один ресурс, содержащий курс УЕ на конкретную дату.

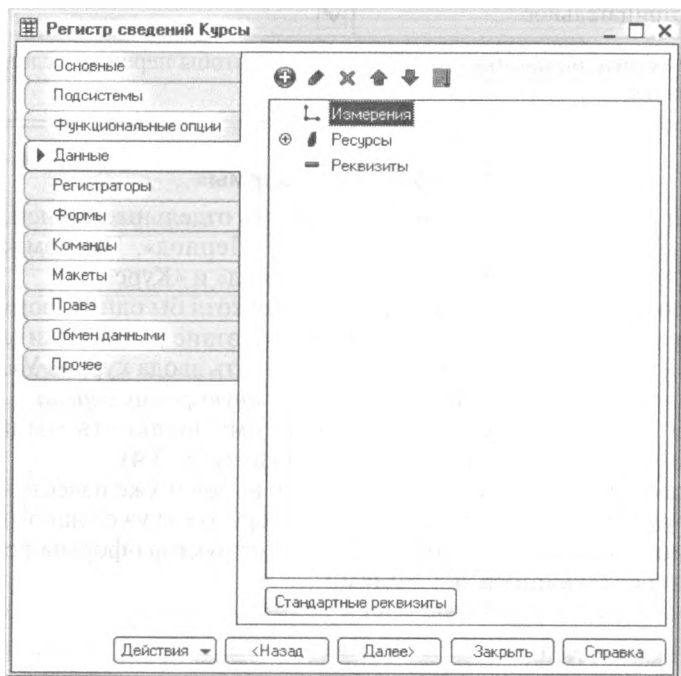


Рис. 3.3. Закладка «Данные»

Поэтому давайте создадим новый ресурс, выполнив следующее упражнение.

### Упражнение № 3.2

Выделите в окне «Регистр сведений: Курсы» закладку «Данные», затем выделите в окне ветвь «Ресурсы» и нажмите на кнопку Ins. После чего заполните свойства нового ресурса.


Основные	
Свойство	Значение
Имя	Курс
Синоним	Курс
Комментарий	Содержит курс валюты
Тип данных	
Свойство	Значение
Тип	Число
Длина	10
Точность	4
Неотрицательное	<input checked="" type="checkbox"/>

Нажмите на кнопку «Далее>», для того чтобы перейти к следующей закладке.

### 3.1.3. Закладка «Формы»

Наш регистр сведений будет содержать отдельные записи, которые будут упорядочены по измерению «Период». Причем каждая запись будет иметь две колонки — «Период» и «Курс».

Используем эту закладку для создания хотя бы одной формы для регистра сведений. Поскольку на данном этапе разработки конфигурации нам нужно обеспечить возможность ввода курсов УЕ на заданную дату, то мы создадим только *основную форму списка*.

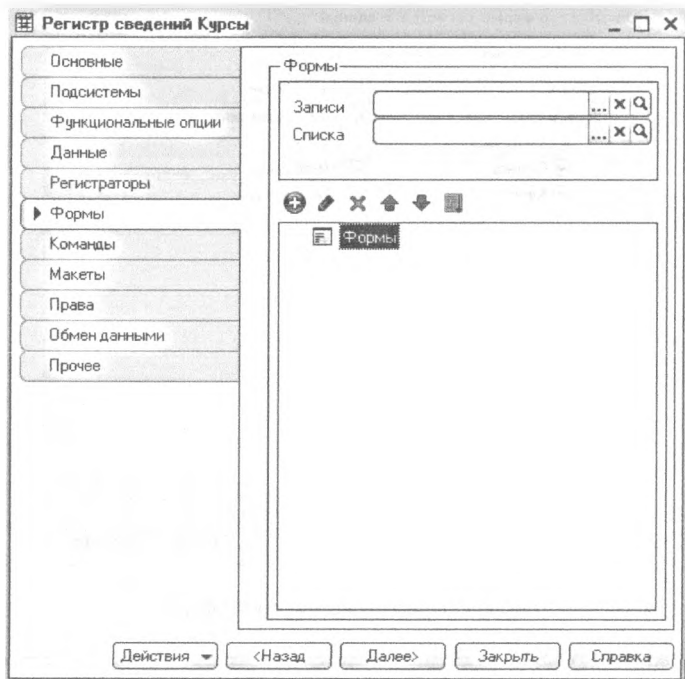
Чтобы создать эту форму, необходимо щелкнуть мышью на кнопке  поля «Основная форма списка» (рис. 3.4).

После таких действий Конфигуратор вызовет уже известный нам конструктор форм, с первым шагом которого мы уже знакомы.

Самостоятельно заполните поля Конструктора формы регистра сведений следующими значениями.

### Упражнение № 3.3

В Конструкторе укажите следующие значения в полях его диалога:



**Рис. 3.4.** Создание новой формы в закладке «Формы»

Свойство	Значение
Выберете тип формы	<b>Форма списка регистра сведений</b>
Тип формы	<b>Обычная</b>
Назначить форму основной	<input checked="" type="checkbox"/>
Имя	<b>ОсновнаяФормаСписка</b>
Синоним	<b>Основная форма списка</b>
Комментарий	<b>Основная форма списка регистра «Курсы»</b>

Нажмите на кнопку «Далее>», для того чтобы рассмотреть следующий шаг конструктора более подробно.

После указанных выше действий форма Конструктора формы списка примет вид, изображенный на рис. 3.5.

Как мы видим теперь, как и в Основной форме констант, эта форма содержит табличное поле, в котором мы можем по своему желанию (в некоторых пределах) настроить состав колонок, которые она будет содержать.

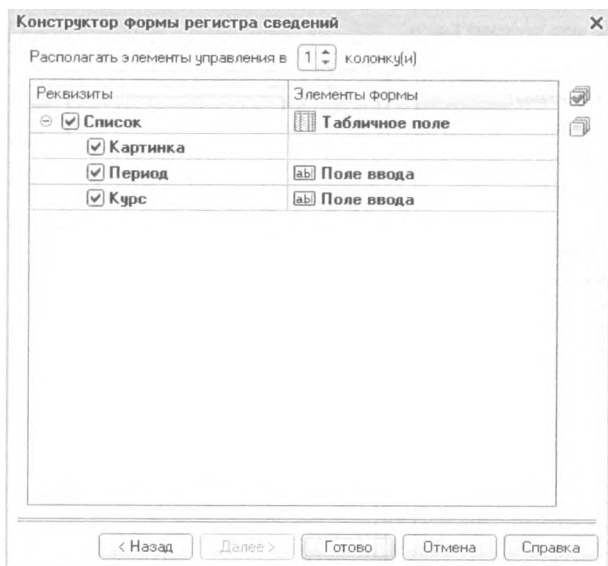


Рис. 3.5. Определение реквизитов формы

### Упражнение № 3.4

Нажмите на кнопку «Готово», чтобы оценить результат работы конструктора (рис. 3.6). В верхней части управляемой формы расположена командная панель, которая уже содержит необходимый набор кнопок управления для работы со списком записей регистра сведений, которые будут отображаться в табличном поле (оно расположено ниже).

Сохраните конфигурацию.

Закройте окно редактирования регистра сведений и обновите конфигурацию (обратите внимание на появление нового объекта в ИБ).

## 3.2. ВВОД ДАННЫХ В РЕГИСТР СВЕДЕНИЙ

Как уже отмечалось, в Конфигураторе мы только создали структуру хранения, а ее заполнением конкретными значениями будет заниматься пользователь в режиме «1С:Предприятие».

### Упражнение № 3.5

Запустите режим «1С: Предприятие».

Выберите пункт меню «Операции — регистр сведений», а в открывшемся диалоговом окне выбора объекта нажмите клавишу ОК.



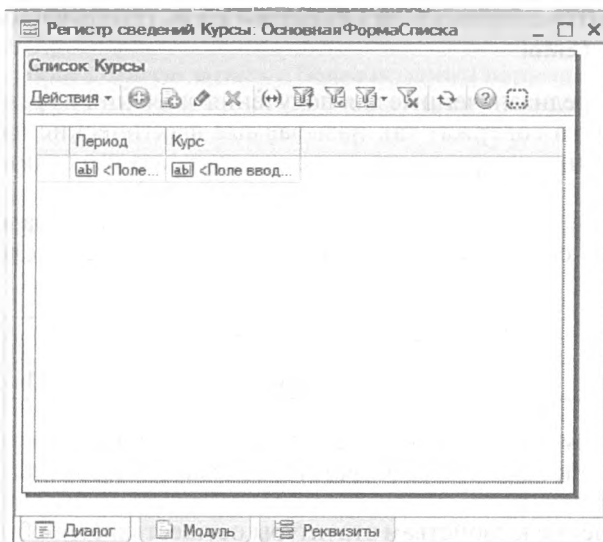


Рис. 3.6. Основная форма списка регистра сведений «Курсы» после работы конструктора

Введите в регистр новые значения, используя следующую таблицу:

Дата	Значение
28.12.2007	25,2600
01.01.2008	24,8500
01.01.2009	23,7844
31.01.2009	23,6222
02.03.2009	23,5729
01.04.2009	23,3801
01.05.2009	23,1021
31.05.2009	23,7090
30.06.2009	23,3483
30.07.2009	23,2450
18.08.2009	23,3532

**Совет:** для ввода новых записей удобно использовать кнопку **Ins** на клавиатуре. Кроме того, все возможные действия, которые можно выполнять в табличном поле, можно узнать, используя правую кнопку мыши.

### 3.3. РАБОТА С РЕГИСТРОМ СВЕДЕНИЙ ИЗ ВСТРОЕННОГО ЯЗЫКА СИСТЕМЫ

Методы, предназначенные для получения значений из регистра сведений, часто содержат так называемые фактические параметры. Многие из них возвращают в качестве результата тип данных «Структура».

Этот тип данных представляет собой запись в оперативной памяти (ОП) с произвольным числом полей (свойств) следующего вида:

Ключ:	Свойство1	Свойство2	...	СвойствоN
Значение:	Значение1	Значение2	...	ЗначениеN

При этом количество свойств и их имена (*ключи*), а также сами значения определяются соответствующими методами объекта типа данных «Структура».

Обращение к свойствам структуры осуществляется обычным для объектной технологии способом «через точку»: **ИмяСтруктуры.ИмяСвойства**.

#### 3.3.1. Получение отдельных значений

Как уже было отмечено выше, в периодическом регистре значения зависят от того периода времени, на который они были введены. У объекта данного типа имеются три схожих по применению метода, позволяющих получить значение типа «Структура», состав свойств которого полностью соответствует составу *ресурсов* регистра сведений, спроектированных на этапе конфигурирования.

Все эти методы начинаются со слова «Получить».

Метод **Получить(...)** пытается найти запись с указанным в качестве первого параметра моментом времени. Если запись есть, то возвращается структура, содержащая значения всех ресурсов. Если нет, то структура будет содержать пустые значения.

Приведем пример получения значения курса УЕ на конкретную дату:

```
//ПримерыОбращенияКРегиструСведений() — содержит примеры
// обращения к регистру сведений
Процедура ПримерыОбращенияКРегиструСведений()
// Найдем значение курса на конкретную дату:
Запись =РегистрыСведений.Курсы.Получить('20090101');
Сообщить(«Курс УЕ на 01.01.2009г.»+Запись.Курс);
КонецПроцедуры
```

А что делать, если записи на конкретный период нет? Тут возможны две стратегии:

- ✓ либо брать *первую* запись с более поздним периодом;
- ✓ либо брать *последнюю* запись с более ранним периодом.

Если записи, удовлетворяющие необходимому условию, нет, то, как и в предыдущем случае, возвращается структура, содержащая все пустые значения.

За первый способ получения записи отвечает метод **ПолучитьПервое(...)**, а за второй — **ПолучитьПоследнее()**.

Приведем пример:

**Процедура ПримерыОбращенияКРегиструСведений()**

// .....

// Найдем курс на сегодняшнее число:

**Запись=РегистрыСведений.Курсы.ПолучитьПоследнее(ТекущаяДата());**

**Сообщить(«Сегодня курс УЕ=»+Запись.Курс);**

**КонецПроцедуры**

Для закрепления материала выполните следующее упражнение.

### Упражнение № 3.6

---

---

Включите в процедуру — обработчик **ПриНачалеРаботыСистемы()** вызов процедуры: **ПримерыОбращенияКРегиструСведений()**.

В этой процедуре напишите фрагмент программного кода, который распечатывал бы в окне «Служебные сообщения» первое значение валютного курса в 2009 г.

Каким методом Вы воспользуетесь?

Обязательно проверьте работу фрагмента путем запуска режима «1С:Предприятие».

---

---

Для надежности приводим текст данного фрагмента программного кода (сравните со своим вариантом).

### ЛИСТИНГ Ю\_3.1

**Процедура ПримерыОбращенияКРегиструСведений()**

// Найдем первое значение валютного курса в 2009 г.

**Запись=РегистрыСведений.Курсы.ПолучитьПервое('20090101');**

**Сообщить(«Сегодня курс УЕ=»+Запись.Курс);**

**КонецПроцедуры**

### 3.3.2. Менеджер записи

Для того чтобы поместить в регистр сведений новую запись или заместить существующую, необходимо сначала создать объект типа «РегистрСведенийМенеджерЗаписи».

Делается это путем вызова соответствующего метода **СоздатьМенеджерЗаписи()**: для объекта типа **РегистрСведенийМенеджер**:

**Объект=РегистрыСведений.ИмяРегистра.СоздатьМенеджерЗаписи();**

После того как объект такого типа создан, Вы сможете применять его методы:

- ✓ **Прочитать();**
- ✓ **Записать();**
- ✓ **Удалить(),**

которые выполняют одноименные действия.

Перед вызовом этих методов необходимо предварительно заполнить свойства этого объекта, которыми будут измерения, ресурсы и реквизиты конкретного регистра сведений, менеджер записи которого Вы создали. Для периодических реквизитов у этого менеджера есть свойство «Период», которое позволяет определить конкретную запись, содержащую это значение.

Предположим, что мы хотим при **открытии формы** записать в регистр сведений «Курсы» значение 23.7244 на 9 января 2009 г., причем если запись на такое число уже есть, то мы должны оставить ее без изменений.

С точки зрения здоровой логики алгоритм решения этой задачи прост.

Попытаться прочитать запись из регистра на 09.01.2009 г.  
Убедиться, что такой записи нет.

Записать новую запись на указанную дату.

В решении второго пункта нам поможет метод **.Выбран()**, который поможет определить, считана ли запись.



## ЛИСТИНГ Ю\_3.2

**Процедура ПримерыОбращенияКРегиструСведений()**

// Сначала создадим менеджер записи:

**РегСведКурсы=РегистрыСведений.Курсы.СоздатьМенеджерЗаписи();**

// Затем определим период в который будем писать данные:

**РегСведКурсы.Период='20090109'; //9 января 2009 г.**

// 1. Пытаемся прочитать запись

**РегСведКурсы.Прочитать();**

// 2. Если записи нет

**Если Не РегСведКурсы.Выбран() Тогда**

**// 3. Запишем новую**

**РегСведКурсы.Период='20090109'; //9 января 2009 г.**

**РегСведКурсы.Курс=23.7244;**

**РегСведКурсы.Записать();**

**КонецЕсли;**

**Сообщить(«Сегодня курс УЕ=>»+ РегСведКурсы.Курс);**

**КонецПроцедуры;**  
**Процедура ПриНачалеРаботыСистемы()**  
    **ИмяПроцедуры();**  
**КонецПроцедуры**

### 3.3.3. Использование выборки

В предыдущих разделах мы обращались к конкретным записям (по одной), а что делать, если нужно обработать все или какую-то часть записей, удовлетворяющих каким-либо условиям?

Данная постановка реализуется с помощью открытия так называемой *выборки*.

Для регистра сведений это делается следующим образом:

```
ВыборкаСведений=РегистрыСведений.<ИмяРегистраСведений>.Выбрать(...);  
Пока ВыборкаСведений.Следующий() Цикл  
КонецЦикла;
```

По методу **.Выбрать()** мы получаем объект типа **РегистрСведенийВыборка**, для которого в цикле применяем его метод **.Следующий()**, чтобы извлечь из выборки очередную запись.

Внутри цикла Вы можете обращаться к значениям измерений и ресурсов просто как к свойству этого объекта.

Например: **ВыборкаСведений.Период**.

## 3.4. САМОСТОЯТЕЛЬНАЯ РАБОТА


Для закрепления только что рассмотренного материала выполните следующие упражнения самостоятельно.


### Упражнение № 3.7

---

Добавьте в процедуру Модуля Приложения **ПримерыОбращенияКРегиструСведений()** фрагмент, позволяющий записать в регистр сведений «Курсы» значение 23.7244 на 9 января 2009 г. Причем если запись на такое число уже есть, ее необходимо оставить без изменений.

Используя дерево окна «Конфигурация», откройте форму «**ОсновнаяФормаСписка**» в редакторе форм.

Выделите мышью кнопку  панели инструментов (рис. 3.7) и заполните ее свойства следующими значениями:

Свойство	Значение
Имя	ПечатьКурсаУЕ
Тип	Действие
Действие	ПечатьКурсаУЕ
Текст	Печать курса УЕ
Подсказка	Печать курса УЕ
Пояснение	Печать курса УЕ
Картинка	 Картинка: bmp
Отображение	Надпись и картинка


Для надежности приводим текст данного фрагмента программы (сравните со своим вариантом).

### ЛИСТИНГ Ю\_3.3

```

Процедура ПримерыОбращенияКРегиструСведений()
// записать в регистр сведений «Курсы» значение 23.7244 на 9 //января 2009г.
// Сначала создадим менеджер записи:
РегСведКурсы=РегистрыСведений.Курсы.СоздатьМенеджерЗаписи();
// Затем определим период в который будем писать данные:
РегСведКурсы.Период='20090109'; // 9 января 2009 г.
// 1. Пытаемся прочитать запись
РегСведКурсы.Прочитать();
// 2. Если записи нет
Если Не РегСведКурсы.Выбран() Тогда
// 3. Запишем новую
РегСведКурсы.Период='20090109'; // 9 января 2009 г.
РегСведКурсы.Курс=23.7244;
РегСведКурсы.Записать();
КонецЕсли;
КонецПроцедуры

```

**Совет:** для выбора картинки необходимо в диалоге «Выбор картин-ки» сначала выбрать закладку «Из библиотеки», а затем на закладке «Стандартные» картинку «PRINT» .

### Упражнение № 3.8

Переключитесь на закладку «Модуль» и напишите процедуру **Печать-КурсаУЕ()**, которая распечатывала бы в окне «Служебные сообщения» курс УЕ на каждую дату, что введена в регистре сведений «Курсы».

Проверьте работу кнопки  в режиме «1С:Предприятие».

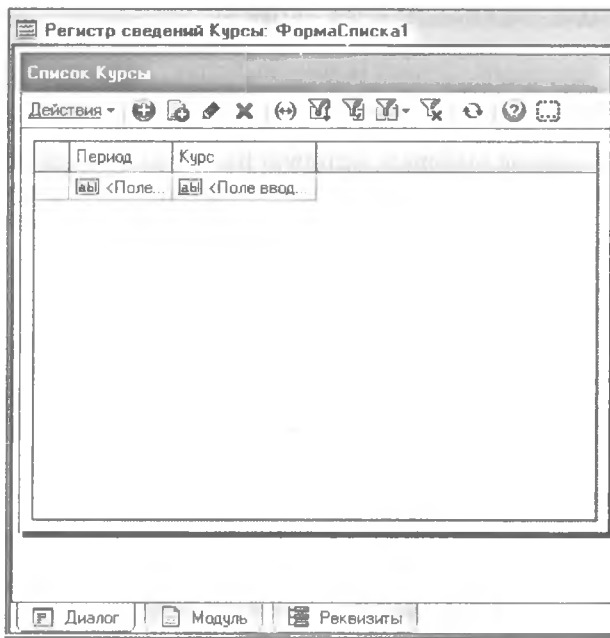


Рис. 3.7. Форма списка регистра сведений перед началом редактирования

По окончании отладки замените номер версии в свойствах конфигурации на значение «03».

Каким типом объекта Вы воспользуетесь для решения поставленной задачи?

Для надежности приводим текст данной процедуры (сравните со своим вариантом).

#### ☁ ЛИСТИНГ Ю\_3.4

**Процедура ПечатьКурсаУЕ(Кнопка)**

**ВыборкаСведений=РегистрыСведений.Курсы.Выбрать();**

**Пока ВыборкаСведений.Следующий() Цикл**

**Сообщить(«»+ВыборкаСведений.Период+Символы.Таб+ВыборкаСведений.Курс);**

**КонецЦикла;**

**КонецПроцедуры**

### 3.5. ИСПОЛЬЗОВАНИЕ НАБОРА ЗАПИСЕЙ

Во всех предыдущих случаях (кроме выборки) мы считывали значения по одному, что нельзя признать эффективным способом обработки при большом или очень большом числе записей.

Использование выборки, которую Вы должны были освоить самостоятельно, позволяет несколько оптимизировать процесс чтения записей за счет того, что записи считываются отдельными *порциями*.

Но выборка, к сожалению, осуществляется только в прямом направлении. А в постановке задачи пример отчета содержит обратную сортировку.

Решить эту проблему можно путем использования *набора записей* регистра сведений. Методика применения здесь следующая:

```
НабКурсы=РегСведКурсы.СоздатьНаборЗаписей();
НабКурсы.Прочитать();
Для Каждого ТекСтрока из НабКурсы Цикл
Сообщить(«»+ТекСтрока.Период+Символы.Таб+ТекСтрока.Курс);
КонецЦикла;
```

Метод **.СоздатьНаборЗаписей()** создает объект типа **РегистрСведений-НаборЗаписей**, который будет содержать коллекцию (массив) отдельных записей.

Объект этого типа имеет достаточное число методов:

```
Вставить(<Индекс>);
Добавить();
Записать(<Замещать>);
Индекс(<Запись>);
Количество();
Очистить();
Прочитать();
Сдвинуть(<Запись>, <Смещение>);
Удалить(<Запись>),
```

которые позволяют работать как со всем набором записей в целом, так и с каждой из них в отдельности.

В частности, метод **.Прочитать()** позволяет считать всю таблицу из ИБ в оперативную память. Для последовательного перебора отдельных записей можно воспользоваться циклом «Для Каждого» (см. пример выше), который позволит последовательно обойти все записи из набора.

Однако это перебор записей в прямом направлении сортировки. Для того чтобы распечатать курсы в обратном порядке, нужно вспомнить, что набор записей — это коллекция. А следовательно, для доступа к отдельным записям (в «прямую») можно воспользоваться



операцией индексирования ([...]). Индексы в коллекциях отсчитываются от нуля. Поэтому, чтобы получить самую первую и самую последнюю запись, следует записать следующие выражения:

```
ПерваяЗапись = НабКурсы[0];  
ПоследняяЗапись = НабКурсы[НабКурсы.Количество()-1];
```

А цикл распечатывания курсов в обратном направлении будет выглядеть следующим образом:

```
РегСведКурсы=РегистрыСведений.Курсы;  
НабКурсы=РегСведКурсы.СоздатьНаборЗаписей();  
НабКурсы.Прочитать();  
к=НабКурсы.Количество();  
Для н=1 по НабКурсы.Количество() Цикл  
ТекСтрока=НабКурсы[к-н];  
Сообщить(«»+ТекСтрока.Период+Символы.Таб+ТекСтрока.Курс);  
КонецЦикла;
```

### Упражнение № 3.9

---

Измените текст процедуры **ПечатьКурсаУЕ()** в модуле формы списка регистра сведений, включив в него текст приведенный выше.

Проверьте работу процедуры после внесенных изменений.

---

Для надежности приводим текст данной процедуры (сравните со своим вариантом).

### ЛИСТИНГ Ю\_3.5

```
Процедура ПечатьКурсаУЕ(Кнопка)  
РегСведКурсы=РегистрыСведений.Курсы;  
НабКурсы=РегСведКурсы.СоздатьНаборЗаписей();  
НабКурсы.Прочитать();  
к=НабКурсы.Количество();  
Для н=1 по НабКурсы.Количество() Цикл  
ТекСтрока=НабКурсы[к-н];  
Сообщить(«»+ТекСтрока.Период+Символы.Таб+ТекСтрока.Курс);  
КонецЦикла;  
КонецПроцедуры
```

#### 3.5.1. Обработка событий

Объект типа **РегистрСведенийНаборЗаписей** предоставляет возможность обрабатывать два события:

- ✓ **ПередЗаписью;**
- ✓ **ПриЗаписи.**

Оба события позволяют контролировать процесс записи в ИБ.

Для того чтобы обработать нужное событие, необходимо написать процедуру с идентификатором соответствующего события в модуле набора записей регистра сведений.

В п. 1 постановки задачи сказано: «Ввод всех данных и документов должен выполняться не ранее *даты регистрации объединения*», для того чтобы решить поставленную задачу, выполните следующее упражнение.

### Упражнение № 3.10

---

В окне редактирования регистра сведений «Курсы» переключитесь на закладку «Прочее» (рис. 3.8).

Нажмите на кнопку «Модуль набора записей» и наберите следующий текст.



#### ЛИСТИНГ Ю\_3.6

```
// Модуль набора записей регистра сведений «Курсы»  
// ПередЗаписью() — проверяет дату периода перед записью в ИБ  
// Параметры  
// Отказ (Булево) позволяет отказаться от записи (Ложь-по умолчанию-  
// выполнение операции, Истина — отказ от операции.  
// Замещение (Булево) режим записи набора (Ложь-по умолчанию-дописывание  
//записи, Истина — замещение записи.
```

```
Процедура ПередЗаписью(Отказ, Замещение)  
ДатаРегистрации = Константы.ДатаРегистрации.Получить();  
Для каждого ТекЗапись Из ЭтотОбъект Цикл  
// Если период раньше даты регистраци  
Если ТекЗапись.Период < ДатаРегистрации Тогда  
// откажемся от записи  
Отказ = Истина;  
// сообщим об ошибке  
Сообщить(«Запись от «+ТекЗапись.Период+  
» с курсом «+ТекЗапись.Курс+  
» раньше даты регистрации объединения!»,  
СтатусСообщения.Важное);  
КонецЕсли;  
КонецЦикла;  
Если Отказ Тогда  
Сообщить(«Запись не выполнена!!!», СтатусСообщения.ОченьВажное);  
КонецЕсли;  
КонецПроцедуры
```

Проверьте работу процедуры путем ввода новой или редактирования существующей записи в регистре сведений «Курсы».

---

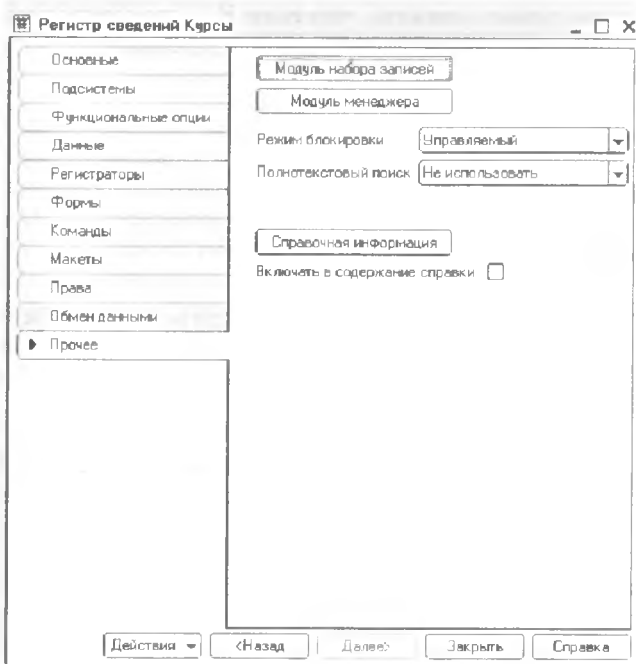


Рис. 3.8. Закладка «Прочее» окна редактирования

Дадим необходимые пояснения к нашим действиям.

Причина, по которой мы использовали именно такую процедуру, как обработчик событий, состоит в том, что запись новых данных и исправление существующих могут выполняться как вследствие интерактивных действий пользователя, так и программным путем.

Если бы проверка значения периода выполнялась только в модуле формы, то тогда мы смогли бы контролировать только интерактивные действия пользователя, связанные с непосредственным редактированием. При этом сохранялась бы возможность записи в реестр программным путем значений, которые не удовлетворяли бы условиям поставленной задачи.

Можно сказать, что модуль набора записей — это «последняя линия обороны» и его задача контролировать те данные, которые помещаются в реестр.

Формальный параметр **Отказ** анализируется системой после обработки события, поэтому

**Отказ = Истина;**

приведет к отмене транзакции.

При этом нужно помнить, что если Вы отменяете транзакцию записи в процедуре **ПередЗаписью(...)** или в **ПриЗаписи(...)**, то метод **.Записать()** следует вызвать внутри конструкции:

```
Попытка
...
Исключение
...
КонецПопытки;
```

В противном случае система будет генерировать ошибку времени выполнения.

### 3.6. ЧТО МЫ УЗНАЛИ

В этой главе мы познакомились с прикладным объектом «Регистры сведений», который предназначен для сохранения в ИБ значений, зависящих от значений различных параметров. Если для регистра свойство периодичность уставлено в какое-либо из значений «В пределах ...», то значения такого регистра будут развернуты во времени.

С помощью окна редактирования регистра мы последовательно определили: его свойства, структуру хранения (какие данные будет содержать регистр сведений) и сформировали, используя конструктор, его основную форму списка.

Эту основную форму списка мы использовали для ввода новых и редактирования существующих записей регистра сведений «Курсы», заполнив его в режиме «IS:Предприятие» конкретными значениями.

Мы также рассмотрели вопросы получения и записи отдельных записей для периодического регистра сведений, познакомились с методами организации выборки записей в цикле.

Работа с набором записей позволит достичь еще большей эффективности за счет того, что все (или часть) записей из таблицы, расположенной в ИБ, считываются в оперативную память по одной операции чтения. Обработка записей, образующих коллекцию, возможна как путем последовательного циклического перебора с использованием конструкции цикла для каждого, так и путем произвольного обращения по индексу в массиве.

В этой главе мы рассмотрели новую возможность — обработку событий записи в регистр, которая может выполняться в модуле набора записей регистра сведений.

В предыдущей главе мы использовали набор записей для того, чтобы распечатать содержимое регистра сведений, при этом мы считывали в ОП всю таблицу ИБ за одну операцию чтения.

Объект «Запрос» встроенного языка системы предоставляет еще более эффективный механизм обработки данных: для обработки необходимых данных объект формирует *временную таблицу*, структура и содержимое которой полностью определяются *текстом запроса*.

Текст запроса, по сути дела, определяет структуру некоторой временной таблицы, которая будет построена в результате работы объекта. Эта таблица имеет тип данных — РезультатЗапроса, который обладает достаточным набором методов для организации выборки, извлечения информации из временной таблицы.


Синтаксис языка запросов достаточно подробно описан в документации, поэтому в этой главе мы остановимся более подробно на методике его применения для формирования отчетов.

В этом юните мы рассмотрим основы применения объекта и написание текста запроса на примере п. 3 постановки задачи. Для этого мы напишем полный аналог процедуры **ПечатьКурсаУЕ(...)** из модуля формы списка регистра сведений «Курсы».

#### 4.1. СОЗДАНИЕ ЗАПРОСА «ВРУЧНУЮ»

Забегая вперед, скажем, что в системе имеется необходимый набор конструкторов, которые облегчают процесс создания объектов в конфигурации. Но для лучшего понимания работы Конструктора запросов (который мы рассмотрим в последующих юнитах) один из отчетов мы сделаем полностью «вручную» — практически без применения конструкторов.

##### Упражнение № 4.1

В форме списка регистра сведений «Курсы» выделите мышью кнопку  на нижней панели инструментов и заполните ее свойства следующими значениями:

Свойство	Значение
Имя	<b>ПечатьЧерезЗапрос</b>
Тип	<b>Действие</b>
Действие	<b>ПечатьЧерезЗапрос</b>

Свойство	Значение
Текст	Печать через запрос
Подсказка	Печать через запрос
Пояснение	Печать через запрос
Отображение	Надпись

После этих действий форма примет вид, изображенный на рис. 4.1.

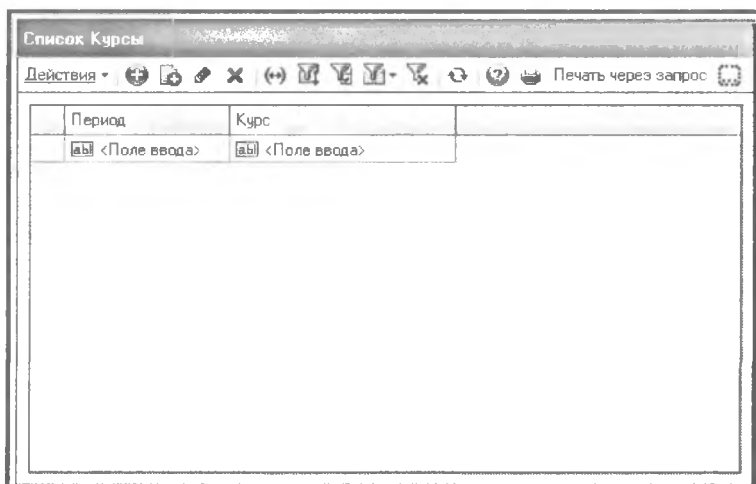


Рис. 4.1. Результат выполнения упражнения № 4.1

#### 4.1.1. Написание текста запроса и его выполнение

Методически работа с объектом типа «Запрос» строится следующим образом:

```
// Создаем объект запрос
Запрос = Новый Запрос();
Запрос.Текст = «Здесь мы запишем текст запроса»;
// Если текст запроса содержит некоторые параметры, то перед
// его выполнением необходимо им присвоить конкретные значения:
Запрос.УстановитьПараметр(«Параметр1»,ЗначениеПараметра1);
.....
Запрос.УстановитьПараметр(«ПараметрN»,ЗначениеПараметраN);
Результат = Запрос.Выполнить();
// Обработка результата
```

Текст запроса всегда начинается с ключевого слова **ВЫБРАТЬ** языка запросов, за которым через запятую следует указать: из каких реквизитов каких объектов запрос должен сделать выборку.

Так, в нашем случае нам нужны свойства *Период* и *Курс регистра сведений* **Курсы**. Поэтому мы можем записать следующий текст:

**Запрос.Текст=«ВЫБРАТЬ  
|РегистрСведений.Курсы.Период, РегистрСведений.Курсы.Курс»;**

Если мы сейчас выполним запрос, то объект построит следующую таблицу, содержащую результат запроса:

Период	Курс
28.12.2007	<b>25,2600</b>
01.01.2008	<b>24,8500</b>
01.01.2009	<b>23,7844</b>
31.01.2009	<b>23,6222</b>
02.03.2009	<b>23,5729</b>
01.04.2009	<b>23,3801</b>
01.05.2009	<b>23,1021</b>
31.05.2009	<b>23,7090</b>
30.06.2009	<b>23,3483</b>
30.07.2009	<b>23,2450</b>
18.08.2009	<b>23,3532</b>

Для удобства написания текста запроса можно таблице, из которой объект будет выбирать записи, дать краткое название — *псевдоним*, применив ключевое слово **КАК**:

**Запрос.Текст =**  
**« ВЫБРАТЬ Курсы.Период, Курсы.Курс**  
**|ИЗ РегистрСведений.Курсы КАК Курсы»;**

Сейчас в выборку попадают все записи, что есть в регистре (за все возможные периоды). Для того чтобы записи удовлетворяли какому-либо условию, можно в тексте запроса после оператора **ВЫБРАТЬ** применить ключевое слово **ГДЕ**, после которого написать условие отбора записей.

В учебных целях мы ограничим выборку *датой регистрации объединения*:

**Запрос.Текст =**  
**«ВЫБРАТЬ Курсы.Период, Курсы.Курс**

```
|ИЗ РегистрСведений.Курсы КАК Курсы  
|ГДЕ Курсы.Период >= &Период»;
```

Обратим внимание на знак «&» перед именем *параметра запроса*.

Чтобы запрос работал правильно, необходимо этому параметру присвоить значение перед выполнением запроса, поэтому полный текст будет выглядеть следующим образом:

```
// Создаем объект запрос  
Запрос = Новый Запрос();  
// Формируем текст запроса  
Запрос.Текст =  
«ВЫБРАТЬ Курсы.Период, Курсы.Курс  
|ИЗ РегистрСведений.Курсы КАК Курсы  
|ГДЕ Курсы.Период >= &Период»;  
// Устанавливаем значение параметра  
Запрос.УстановитьПараметр(«Период»,  
Константы.ДатаРегистрации.Получить());  
// И выполняем запрос  
Результат = Запрос.Выполнить();
```

Последнее, что нам осталось сделать, — это указать порядок, в котором записи будут располагаться в таблице, содержащей результат запроса:

```
Запрос.Текст =  
«ВЫБРАТЬ Курсы.Период, Курсы.Курс  
|ИЗ РегистрСведений.Курсы КАК Курсы  
|ГДЕ Курсы.Период >= &Период  
|УПОРЯДОЧИТЬ ПО Период УБЫВ»;
```

#### 4.1.2. Выборка данных из результата запроса

Метод **.Выбрать(...)** возвращает ссылку на объект типа «**Результат-Запроса**», который среди прочих имеет метод открывающий выборку результатов:

```
Выборка=Результат.Выбрать();  
Пока Выборка.Следующий() Цикл  
...=Выборка.Период;  
...=Выборка.Курс;  
.....  
КонецЦикла;
```



Именами свойств записи, которую мы выбрали по методу `.Следующий()`, будут те имена реквизитов источников данных (или их псевдонимов), которые мы указали в тексте запроса.

#### Упражнение № 4.2

---

Внесите соответствующие изменения в текст процедуры `ПечатьЧерезЗапрос(Кнопка)` и проверьте ее работоспособность.

По окончании отладки замените номер версии в свойствах конфигурации на значение «04».

---

Для надежности приводим текст данной процедуры (сравните со своим вариантом).



#### ЛИСТИНГ Ю\_4.1

**Процедура ПечатьЧерезЗапрос(Кнопка)**

// Вставить содержимое обработчика.

//Создаем объект запрос

Запрос = Новый Запрос();

//Формируем текст запроса

Запрос.Текст =

«ВЫБРАТЬ Курсы.Период, Курсы.Курс

|ИЗ РегистрСведений.Курсы КАК Курсы

|ГДЕ Курсы.Период >= &Период

|УПОРЯДОЧИТЬ ПО Период УБЫВ»;

//Устанавливаем значение параметра

Запрос.УстановитьПараметр(«Период»,

Константы.ДатаРегистрации.Получить());

//И выполняем запрос

Результат = Запрос.Выполнить();

// Обработка результата

Выборка=Результат.Выбрать();

Пока Выборка.Следующий() Цикл

Сообщить(«»+Выборка.Период+Символы.Таб+Выборка.Курс);

КонецЦикла;

**КонецПроцедуры**

## 4.2. ЧТО МЫ УЗНАЛИ

В этой главе мы рассмотрели основы работы с объектом «Запрос». Для выполнения запроса необходимо составить его текст на специальном языке запросов.

Более подробно работу с объектом и язык запросов мы рассмотрим в последующих юнитах.

В нашей конфигурации в настоящий момент имеются две формы, открытие которых мы делаем через пункт меню «Операции». Это не очень удобно для пользователя.

Кроме того, в модуле приложения мы реализовали процедуры **ПримерыОбращенияККонстантам()** и **ПримерыОбращенияКРегиструСведений()**, вызов которых расположен в процедуре — обработчике события **ПриОткрытии()**, что нельзя назвать хорошим решением, так как такой вызов происходит при каждом вызове «ИС:Предприятия».

Решить вышеуказанные проблемы можно путем создания в конфигурации своего интерфейса, используя который пользователь сможет давать системе определенные команды.

### 5.1. СОЗДАНИЕ НОВОГО ИНТЕРФЕЙСА

Для того чтобы создать новый интерфейс, необходимо раскрыть ветвь «Общие» дерева Конфигурации, выбрать там поддерево «Интерфейсы». Затем, используя правую кнопку мыши, выбрать пункт меню «Добавить» (рис. 5.1).

Сразу после этой команды система откроет окно Конструктора меню, который помогает построить новый интерфейс на начальном этапе (рис. 5.2).

Можно, конечно, на этом этапе сразу установить все «галочки» во всем списке «Подменю», но мы воспользуемся более тонкой настройкой, так как у нас в конфигурации еще нет всех типов объектов, которые упоминаются в списке «Команда» для каждого подменю.

Расставьте флажки самостоятельно, следуя приведенному ниже упражнению.

#### Упражнение № 5.1

Добавьте в Конфигурацию новый интерфейс и выберите в Конструкторе меню команды, руководствуясь следующей таблицей:

Подменю		Команда
Справочники	<input checked="" type="checkbox"/>	РегистрСведений.Курсы: Открыть регистр сведений
Прочие	<input checked="" type="checkbox"/>	Константы
Отчеты	<input checked="" type="checkbox"/>	Открыть отчет

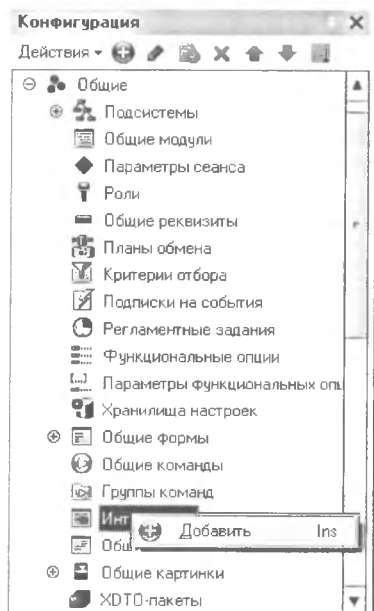


Рис. 5.1. Добавление нового интерфейса

Остальные пункты меню следует оставить без отметки. Нажмите на кнопку «Построить», чтобы закончить работу с Конструктором.

Сразу после нажатия на эту кнопку в дереве окна «Конфигурация» появится новый объект «Интерфейс 1», свойства которого система предложит заполнить.

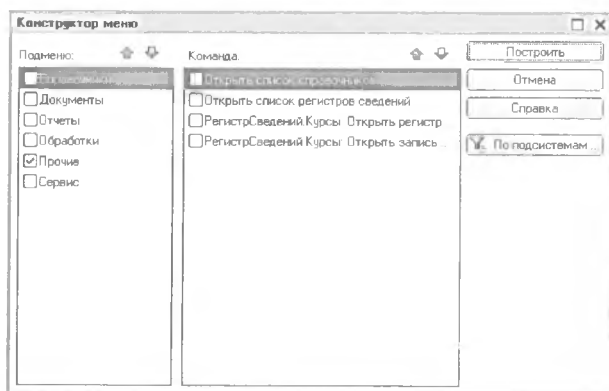


Рис. 5.2. Конструктор меню

## 5.2. СВОЙСТВА ИНТЕРФЕЙСА

Интерфейс имеет только группу свойств «Основные», большая часть из которых нам уже знакома, а остальные свойства мы рассмотрим, когда в этом возникнет необходимость.

Выполните следующее упражнение.

### Упражнение № 5.2

Заполните свойства нового интерфейса следующими значениями:

Свойство	Значение
Имя	<b>ОсновнойИнтерфейс</b>
Синоним	<b>Основной интерфейс</b>
Комментарий	<b>Основной интерфейс конфигурации</b>

Обратим внимание на то, что при завершении своей работы Конструктор меню открыл окно для редактирования интерфейса (рис. 5.3).

В этом окне мы видим, что Конструктор уже сформировал **ГлавноеМеню**, которое будет доступно пользователю. Причем в это меню вошли все те пункты, на которых мы поставили свои «галочки». Это главное меню (в терминах «1С: Предприятия» — Панель) вполне работоспособно.

Осталось сделать совсем немного — это слегка его доработать...

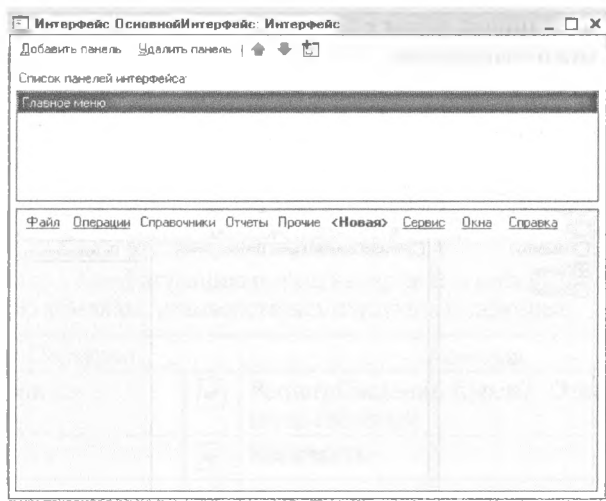


Рис. 5.3. Окно редактирования интерфейса

### 5.3. РЕДАКТИРОВАНИЕ ИНТЕРФЕЙСА

Каждый из пунктов меню имеет свой набор свойств (почти как и везде в «1С:Предприятии»). Поэтому если сделать двойной щелчок мышью на пункте меню «Справочники Курсы», откроется окно свойств для его настройки (рис. 5.4).

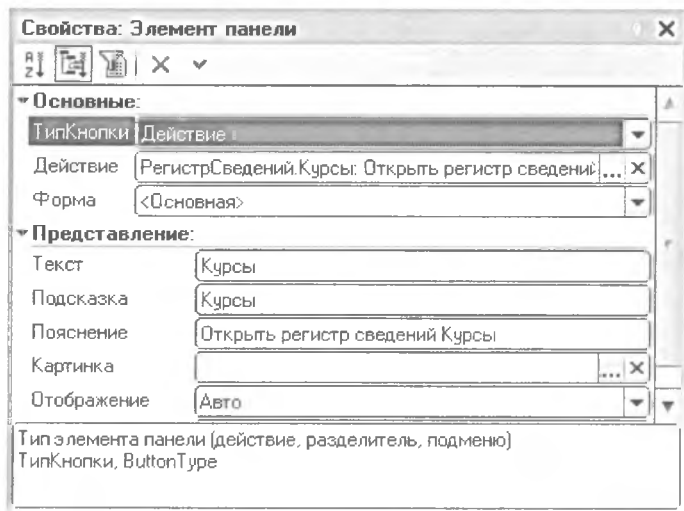


Рис. 5.4. Свойства пункта меню

Назначение большинства свойств понятно из их названия (рассмотрим их подробнее чуть позже).

#### 5.3.1. Редактирование свойств пункта меню

Редактирование большинства свойств пункта меню осуществляется обычным для окна свойств образом — путем выбора значения из выпадающего списка.

На данном этапе свойства тех пунктов меню, которые построены Конструктором, могут быть доработаны только в одном — мы можем добавить к ним *Картинку*, которая будет изображаться на пункте меню.

Для этого мы нажмем на кнопку свойства «Картинка». Сразу после этого откроется диалог «Выбор картинки» (рис. 5.5). Поскольку в нашей конфигурации пока отсутствуют какие-либо картинки, то мы можем воспользоваться стандартными изображениями. Для этого необходимо сначала переключиться на закладку «Из библиотеки», а затем вкладку «Стандартные».

Выполните следующее упражнение.

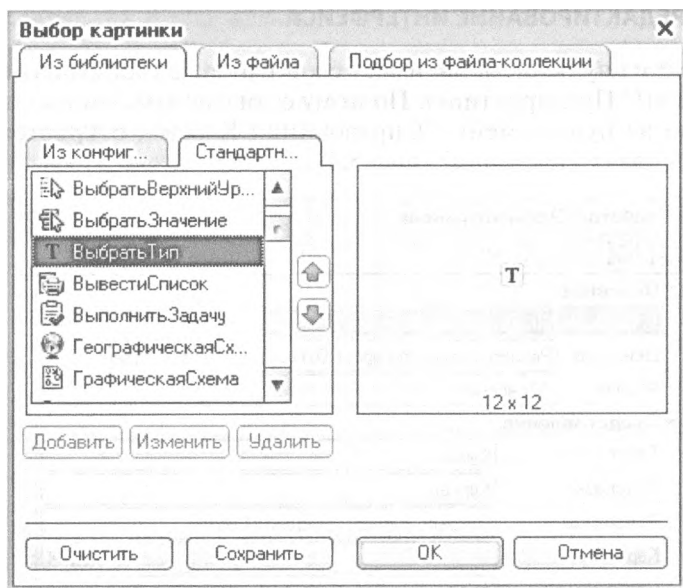




Рис. 5.5. Выбор картинки из стандартных изображений

### Упражнение № 5.3

Используя приведенный выше файл-коллекцию, назначьте указанным пунктам меню следующие картинки:

Для пункта меню «Справочники»	
Свойство	Значение
Текст	Курсы
Подсказка	Курсы
Пояснение	Открыть регистр сведений Курсы
Картинка	
Для пункта меню «Прочие»	
Свойство	Значение
Текст	Константы
Подсказка	Константы
Пояснение	Открыть форму редактирования констант
Картинка	

### 5.3.2. Свойство «Действие»

Если выбрать пункт меню «Константы» из нашего интерфейса и нажать на кнопку выбора для свойства Действие, то тогда система откроет диалог, который изображен на рис. 5.6.

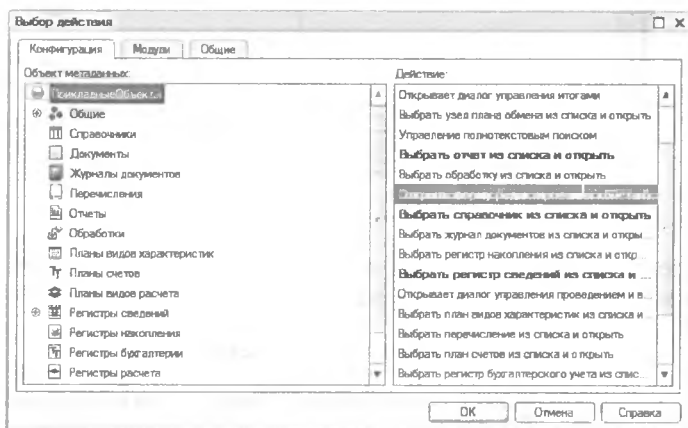


Рис. 5.6. Выбор действия

На этом рисунке хорошо видно, что действие системы при выборе этого пункта меню будет состоять в том, чтобы «Открыть форму редактирования констант». Таким образом, мы приходим к выводу, что в качестве действия может выступать *открытие какой-либо формы*.

Однако не всегда можно необходимое действие, которое следует выполнять при выборе пункта меню, свести к открытию необходимой формы. Достаточно часто эти действия приходится выражать «программным способом», располагая в модулях процедуры — обработки событий.

Так, например, в модуле приложения мы написали процедуры **ПримерыОбращенияККонстантам()** и **ПримерыОбращенияКРегиструСведений()**, вызов которых было бы удобно сделать через пункт меню, а не через процедуру **ПриНачалеРаботыСистемы()** при каждом запуске системы.

Если мы сейчас переключимся в диалоге «Выбор действия» на закладку «Модули», то соответствующий список будет пуст (рис. 5.7). Почему?

Ответ прост: потому что сейчас эти процедуры принадлежат *локальному контексту Модуля приложения*.

А раз так, то вызвать их можно только из какой-либо процедуры этого модуля.

Для того чтобы процедуры **ПримерыОбращенияККонстантам()** и **ПримерыОбращенияКРегиструСведений()** можно было использовать как действие

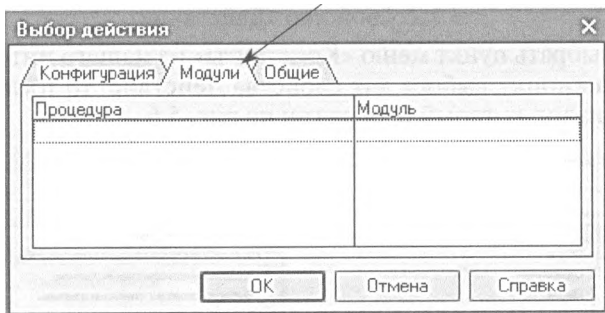


Рис. 5.7. В модуле приложения нет ни одной процедуры-действия

в пункте меню, их необходимо внести в *глобальный контекст*. Тогда эти процедуры появятся в списке возможных действий.

### 5.3.3. Создание новых пунктов меню

Окно редактирования интерфейса позволяет легко создавать новые пункты меню. В самом деле, если взглянуть на рис. 5.3, то можно увидеть, что среди всех пунктов Главного меню есть пункт «<Новая>», который и служит для создания новых пунктов.

Создайте новый пункт меню, выполнив следующее упражнение.

#### Упражнение № 5.4

Сделайте двойной щелчок мышью на пункте меню «<Новая>» и заполните его свойства следующими значениями:

Свойство	Значение
Тип	Подменю
Подменю	<Произвольное подменю>
Текст	Примеры
Подсказка	Примеры процедур
Пояснение	Вызов процедур, содержащих примеры

Обратите внимание, что после изменения свойств этого пункта меню окно редактирования интерфейса тоже изменилось — там добавился еще один пункт меню (рис. 5.8).

Мы опять выделим пункт «<Новая>» из меню «Примеры» и заполним его свойства, но предварительно выполним некоторую подготовительную работу в Модуле приложения.



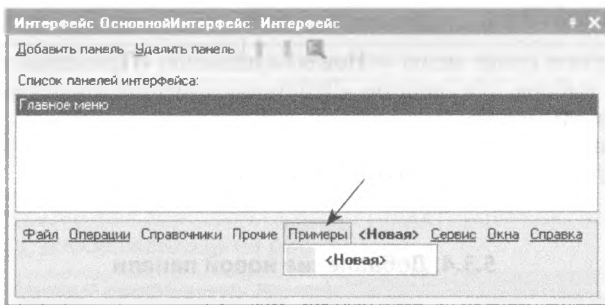


Рис. 5.8. Новое Подменю

### Упражнение № 5.5

Откройте Модуль обычного приложения и выполните следующие действия для процедур:

**ПримерыОбращенияККонстантам(),**  
**ПримерыОбращенияКРегиструСведений(),**

1. Для этих процедур укажите в конце их заголовков добавочное ключевое слово «Экспорт».
2. Удалите вызов этих процедур из процедуры — обработчика **ПриНачалеРаботыСистемы()**.

Теперь можно обеспечить вызов этих процедур из подпунктов этого меню.

Для этого нужно проделать следующее: в свойстве «Действие» нового пункта меню нажать на кнопку выбора и выбрать в открывшемся диалоге «Выбор действия» закладку «Модули» (рис. 5.9).

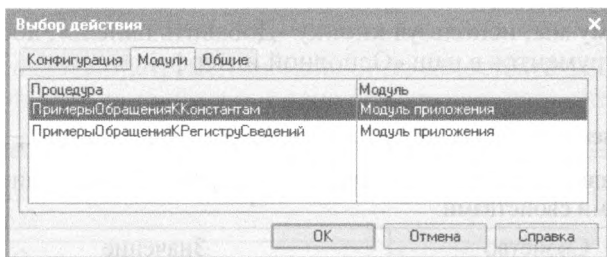


Рис. 5.9. Выбор процедуры в качестве действия

## Упражнение № 5.6

Выделите пункт меню «<Новая>» подменю «Примеры». Используя кнопку выбора, для свойства «Действия» выберите процедуру **Примеры-ОбращенияККонстантам()** из закладки «Модули».

Что произошло?

Для нового пункта меню выберите картинку.

### 5.3.4. Добавление новой панели

Все пункты меню, с которыми мы имели дело, входят в панель «Главное меню», в чем легко убедиться, если открыть ее свойства (рис. 5.10).

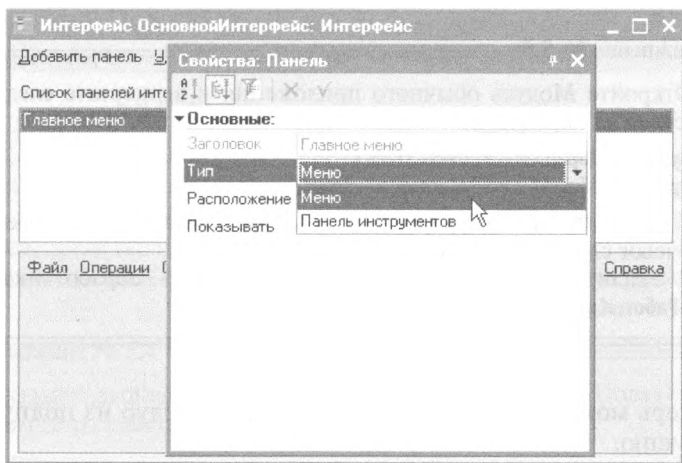


Рис. 5.10. Свойства панели в интерфейсе

Как видно из рисунка, панель может быть либо меню, либо панелью инструментов.

Поэтому мы, используя кнопку «Добавить панель», добавим панель инструментов в наш «Основной интерфейс».

## Упражнение № 5.7

Добавьте в интерфейс «ОсновнойИнтерфейс» новую панель со следующими свойствами:

Свойство	Значение
Заголовок	Главная панель
Тип	Панель инструментов

Свойство	Значение
Расположение	<b>Сверху</b>
С новой строки	<input checked="" type="checkbox"/>
Показывать	<input checked="" type="checkbox"/>

После указанных действий окно по редактированию интерфейса примет вид, изображенный на рис. 5.11.

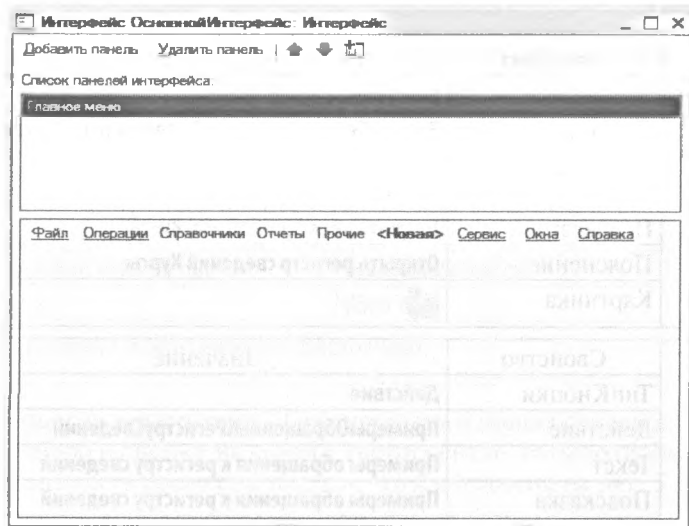



Рис. 5.11. Новая панель инструментов

Кнопка  используется для создания новых кнопок в панели инструментов по той же методике, что создание новых пунктов меню.


#### 5.4. САМОСТОЯТЕЛЬНАЯ РАБОТА

Создание панели инструментов выполните самостоятельно, используя для заполнения свойств ее кнопок следующее упражнение.


##### Упражнение № 5.8

Добавьте в панель инструментов «Главная панель» следующую кнопку:


Свойство	Значение
ТипКнопки	<b>Действие</b>
Действие	<b>ОбщаяФорма. Форма: Открыть форму</b>

Свойство	Значение
Текст	Константы
Подсказка	Константы
Пояснение	Открыть форму редактирования констант
Картинка	

Добавьте в панель инструментов «Главная панель» следующие кнопки:

Кнопка 1	Свойство	Значение
	ТипКнопки	Действие
	Действие	РегистрСведений.Курсы: Открыть регистр сведений
	Текст	Курсы
	Подсказка	Открыть регистр сведений Курсы
	Пояснение	Открыть регистр сведений Курсы
	Картинка	

Кнопка 2	Свойство	Значение
	ТипКнопки	Действие
	Действие	ПримерыОбращенияКРегиструСведений
	Текст	Примеры обращения к регистру сведений
	Подсказка	Примеры обращения к регистру сведений
	Пояснение	Примеры обращения к регистру сведений

Кнопка 3	Свойство	Значение
	ТипКнопки	Действие
	Действие	ПримерыОбращенияККонстантам
	Текст	Примеры обращения к константам
	Подсказка	Примеры обращения к константам
	Пояснение	Примеры обращения к константам
	Картинка	
	Отображение	Надпись и картинка

Какой объект метаданных вы выбрали на закладке «Конфигурация» диалога «Выбор действия» и почему? Что произошло, когда вы изменили свойство «Отображение»?

Сохраните и обновите конфигурацию.

## 5.5. ТЕСТИРОВАНИЕ ИНТЕРФЕЙСА

После того как разработка интерфейса завершена, его необходимо проверить. Никакого другого способа проверки, кроме запуска конфигурации в режиме «1С:Предприятие», не существует. Однако если сейчас мы выполним запуск, то в этом режиме нашего интерфейса не будет. Почему?

Потому что у нас нет ни одного пользователя (о пользователях — мы расскажем в следующих юнитах), а для конфигурации не назначен основной интерфейс.

Таким образом, нам необходимо в свойствах Конфигурации внести следующие изменения.

### Упражнение № 5.9

В свойствах конфигурации измените следующие свойства:

Свойство	Значение
Основной интерфейс	ОсновнойИнтерфейс
Версия	Юнит 05

Сохраните и обновите конфигурацию.

Теперь если запустить «1С:Предприятие» и попробовать все «наши» пункты меню и кнопки на панели инструментов, то окно приложения может выглядеть примерно так, как это изображено на рис. 5.12.

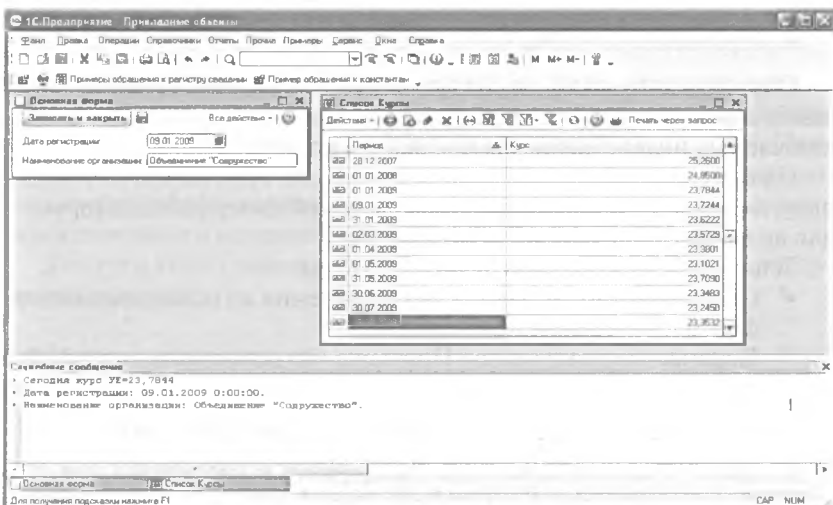


Рис. 5.12. Работа нового интерфейса в режиме «1С:Предприятие»

## 5.6. ПЕРЕКЛЮЧЕНИЕ ИНТЕРФЕЙСОВ

В нашем интерфейсе есть пункт меню «Примеры», которым мы будем пользоваться все же реже, чем обращаться к формам прикладных объектов. Поэтому само собой напрашивается решение — выделить этот пункт в отдельный интерфейс.

Что мы и сделаем.

### Упражнение № 5.10

Добавьте в конфигурацию новый интерфейс и выберите в конструкторе меню команды, руководствуясь следующей таблицей:

Подменю	Меню
Прочие	<b>Переключить интерфейс</b>
Прочие	<b>Интерфейс.Основной интерфейс: Переключить интерфейс</b>

Остальные пункты меню следует оставить без отметки. Нажмите на кнопку «Построить», чтобы закончить работу с Конструктором.

Заполните свойства нового интерфейса следующими значениями:

Свойство	Значение
Имя	<b>ИнтерфейсПримеров</b>
Синоним	<b>Интерфейс примеров</b>
Комментарий	<b>Интерфейс для вызова примеров</b>
Переключаемый	<input checked="" type="checkbox"/>

Обратим внимание на следующий момент: *конструктор меню всегда создает переключаемый интерфейс* (об использовании непереключаемых интерфейсов мы поговорим позже).

Именно поэтому мы вставили в наш интерфейс две команды: одну, позволяющую переключаться на любой интерфейс, а вторую — для возвращения в основной интерфейс.

Теперь нам нужно решить две проблемы:

- ✓ Обеспечить возможность переключения из основного интерфейса в интерфейс примеров.
- ✓ Перенести пункт меню «Примеры» из основного интерфейса в интерфейс примеров.

#### 5.6.1. Добавление команды переключения интерфейса

Откроем свойства основного интерфейса и убедимся в том, что это тоже переключаемый интерфейс (рис. 5.13).

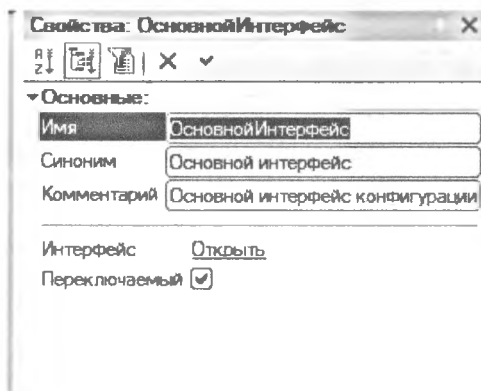


Рис. 5.13. Свойства основного интерфейса

А поскольку это так, то нам необходимо добавить пункт меню для переключения в интерфейс примеров в подменю «Прочие».

Можно, конечно, это сделать уже известным нам способом (используя кнопку выбора действия из соответствующего диалога «Выбор действия», который мы рассматривали в предыдущем разделе), но в Конфигураторе есть другой, более наглядный способ настроить взаимное переключение интерфейсов.

Для того чтобы воспользоваться этим способом, необходимо, используя правую кнопку мыши, выбрать пункт меню «Дополнительно» на объекте «ОсновнойИнтерфейс», тогда откроется специальное окно по дополнительной настройке объекта (рис. 5.14).

На закладке «Интерфейсы» в подокне «Команды» этого окна перечислены для интерфейсов возможные команды переключения.

Так, если выделить в окне «Конфигурация» «ОсновнойИнтерфейс» и для него в дополнительном окне выбрать «ИнтерфейсПримеров», то флаг напротив команды «Основной интерфейс: Переключить интерфейс» будет означать, что из интерфейса примеров можно переключиться в основной интерфейс (рис. 5.14а).

Для того чтобы выполнить обратное действие, выполните следующие действия.

### Упражнение № 5.11

Выделите в дереве «Конфигурация» объект «ИнтерфейсПримеров».

Затем в окне «Дополнительно: ИнтерфейсПримеров» выделите «ОсновнойИнтерфейс» и в подокне «Команды» поставьте флаг напротив команды «Интерфейс примеров: Переключить интерфейс».

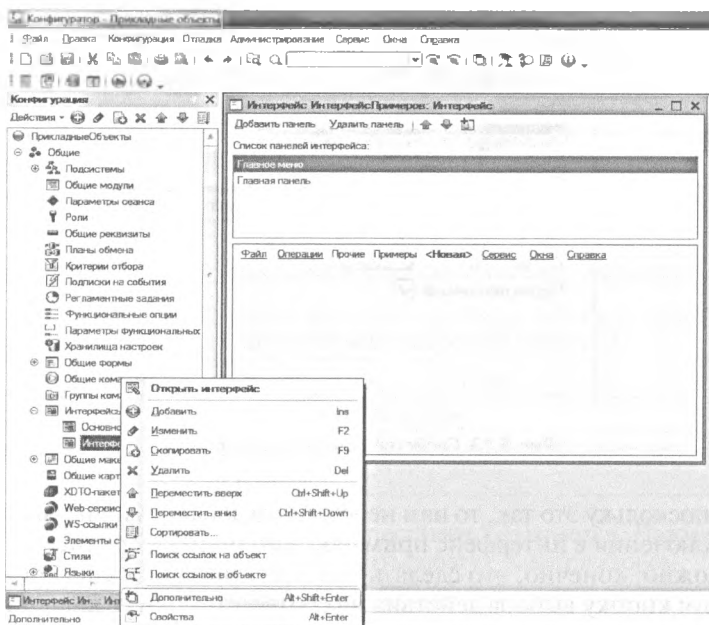


Рис. 5.14. Дополнительная настройка интерфейсов



Рис. 5.14а. Специальное окно по дополнительной настройке объекта

**Внимание!** Возможно, при выполнении задания окно «Дополнительно» не будет содержать никаких данных. Это ошибка программы. Закройте окно «Дополнительно» и окно Конфигурации. Затем откройте эти окна еще раз.

Если Вы выполнили упражнение правильно, то тогда вид окна будет таким, как это изображено на рис. 5.15.



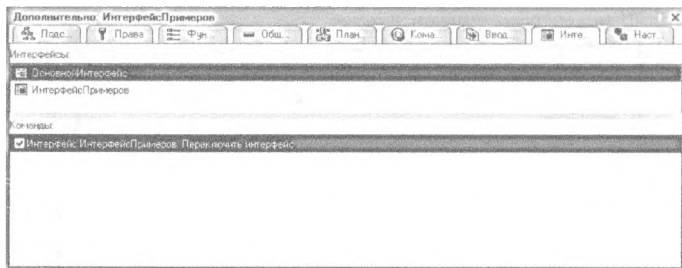


Рис. 5.15. Добавление команды переключения интерфейса в основной интерфейс

Обратимся к окну редактирования основного интерфейса (рис. 5.16) и увидим, что у нас добавился пункт в подменю «Прочие», который будет переключать интерфейс из основного в интерфейс примеров.

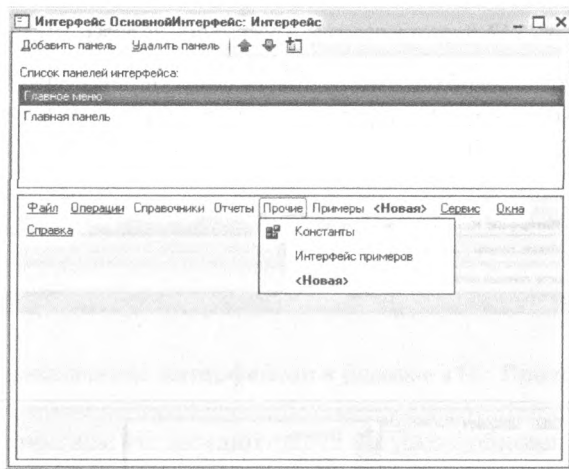


Рис. 5.16. Основной интерфейс после добавления команды переключения

### 5.6.2. Перенос подменю из одного интерфейса в другой

Теперь нам необходимо целое подменю «Примеры» перенести из основного интерфейса в интерфейс примеров.

Это можно сделать двумя способами — через буфер обмена и используя механизм «Перенеси и бросить».

Рассмотрим первый способ.

Выделим в окне редактирования основного интерфейса подменю «Примеры» и, используя правую кнопку мыши, выберем команду «Вырезать» (рис. 5.17).

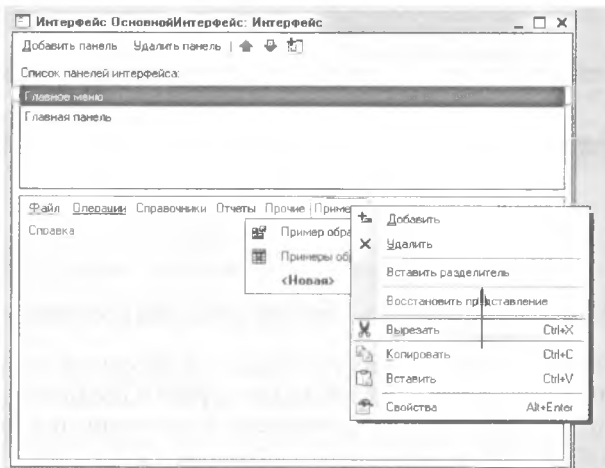


Рис. 5.17. Вырезать подменю и положить его в буфер обмена

После этого необходимо открыть (или переключиться в) окно редактирования для интерфейса примеров и на пункте меню «<Новая>», используя правую кнопку мыши, дать команду «Вставить» (рис. 5.18).

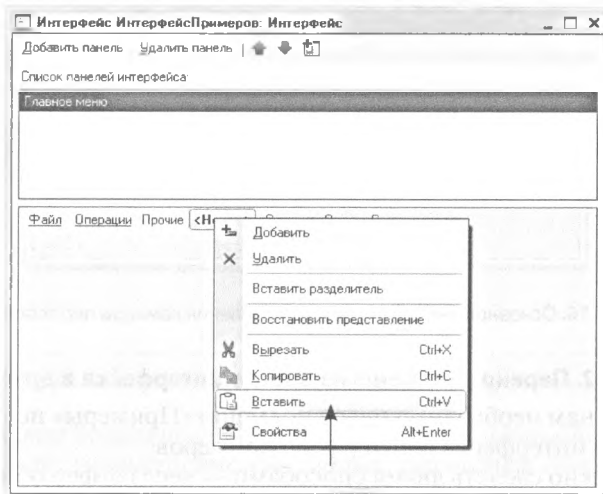


Рис. 5.18. Вставка пункта меню из буфера обмена

После описанных выше действий в интерфейсе примеров возникнет новое подменю «Примеры».

## Упражнение № 5.12

Самостоятельно перенесите из основного интерфейса кнопки панели «Главная панель», которые отвечают за вызов процедур примеров согласно рис. 5.19. Добавьте необходимые атрибуты.

Если Вы все сделали правильно, то оба интерфейса будут выглядеть так, как это изображено на рис. 5.19.

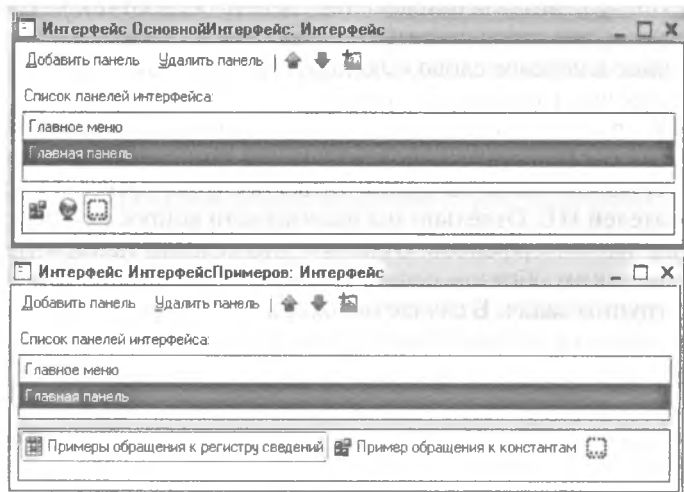


Рис. 5.19. Оба интерфейса после произведенных изменений

### 5.6.3. Переключение интерфейсов в режиме «1С: Предприятие»

Переключение интерфейсов проверьте самостоятельно в режиме «1С: Предприятие». Не забудьте перед запуском обновить конфигурацию в ИБ.

## 5.7. ЧТО МЫ УЗНАЛИ

Эту главу мы целиком посвятили настройке интерфейса. При этом мы рассмотрели Конструктор меню, работа которого существенно упрощает процесс создания нового интерфейса.

Мы рассмотрели назначение всех свойств интерфейса как объекта дерева «Конфигурация».

Каждый интерфейс состоит из отдельных панелей, которые содержат команды к системе. Панель может представлять собой либо меню, либо панель инструментов, которая содержит отдельные кнопки.

Мы подробно разобрали создание новых пунктов меню, кнопок на панели инструментов и как можно изменять их свойства.

Каждый элемент панели (будь то элемент меню или кнопка) имеет свойство «Действие», которое содержит то *действие*, которое должна выполнить система. Таким действием может быть открытие формы какого-либо объекта в конфигурации или вызов процедуры обработчика события.

В связи с последним мы рассмотрели вопрос, как можно «обычную» процедуру Модуля приложения сделать обработчиком события. Оказывается, для этого достаточно внести ее в глобальный контекст (добавочное ключевое слово «Экспорт») и выбрать ее в качестве действия в соответствующем диалоге.

При разработке интерфейсов следует помнить, что какой-либо интерфейс следует сделать основным для конфигурации, для того чтобы им можно было бы пользоваться в случае отсутствия списка пользователей ИБ. Отдельно мы рассмотрели вопрос разработки переключаемых интерфейсов. Отметим, что каждый такой интерфейс позволяет таким образом сосредоточиться на выполнении определенной группы задач. В случае необходимости пользователь сможет переключиться к выполнению другой группы задач, просто выбрав соответствующий пункт меню.

До сих пор для отображения информации мы использовали окно «Служебные сообщения», в которое можно помещать различные сообщения в виде строк текста. Но гораздо чаще информацию приходится представлять в табличном виде.

### 6.1. КОМАНДА «ВЫВЕСТИ СПИСОК» РЕЖИМА «1С:ПРЕДПРИЯТИЕ»

Режим «1С:Предприятие» умеет формировать отчет в табличной форме без какого-либо программирования, для того чтобы познакомиться с этой возможностью, выполните следующее упражнение.

#### Упражнение № 6.1

Запустите конфигурацию в режиме «1С:Предприятие». Откройте форму списка регистра сведений «Курсы» и выберите пункт меню «Действия – Вывести список» (рис. 6.1).

На втором этапе в диалоге «Вывести список» сбросьте флаг у колонки «Картинка» и нажмите на кнопку ОК (рис. 6.2).

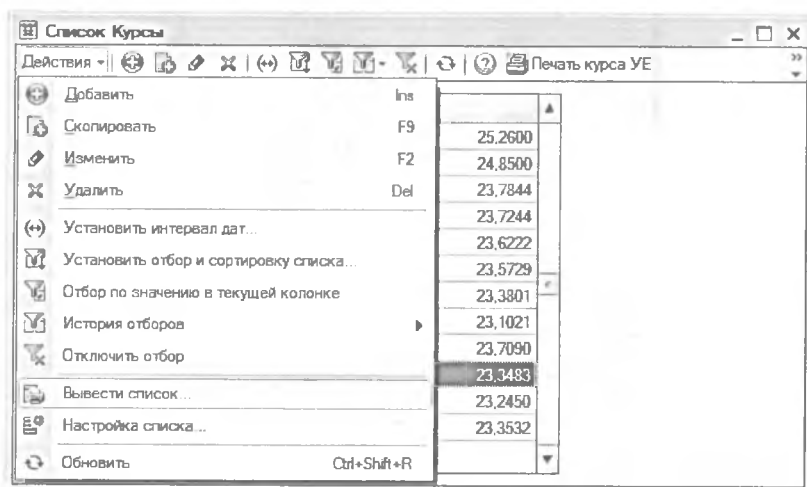


Рис. 6.1. Команда «Вывести список»

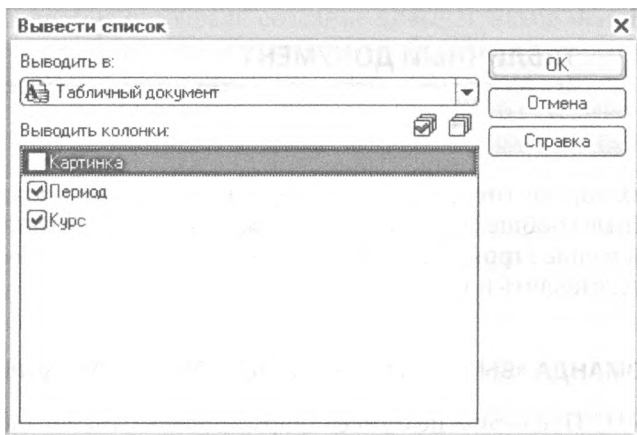


Рис. 6.2. Определение типа выходного документа и состава колонок

Если вы все сделали правильно, то результирующий табличный документ будет выглядеть так, как это изображено на рис. 6.3.

	1	2	3
1	Период	Курс	
2	28.12.2007	25,2600	
3	01.01.2008	24,8500	
4	01.01.2009	23,7844	
5	31.01.2009	23,6222	
6	02.03.2009	23,5729	
7	01.04.2009	23,3801	
8	01.05.2009	23,1021	
9	31.05.2009	23,7090	
10	30.06.2009	23,3483	
11	30.07.2009	23,2450	
12	18.08.2009	23,3532	
13			
14			
15			
16			
17			
18			
19			

Рис. 6.3. Результат выполнения упражнения № 6.1

Если содержимое табличного документа, полученное таким образом, устраивает пользователя, то настройку конфигурации можно оставить без изменения.

Однако постановка нашей задачи требует, чтобы «Во всех отчетах, формируемых ИБ, появлялось название объединения». Для решения этой задачи нам придется прибегнуть к программированию.

И в последующих разделах мы рассмотрим работу с объектом «ТабличныйДокумент» встроенного языка.

## 6.2. ОБЪЕКТ «ТАБЛИЧНЫЙДОКУМЕНТ»

Для работы с табличными документами в «1С:Предприятии» служит одноименный объект типа «ТабличныйДокумент», который относится к группе *Общие объекты*.

Табличный документ предназначен для формирования печатных форм первичных документов и отчетов. Табличные документы могут храниться в конфигурации в качестве макетов, которые служат прообразами для создания других табличных документов. Ячейки макета могут иметь свойство «Параметр», предназначенное для заполнения его конкретными данными. Перед выводом области макета в результирующую таблицу параметрам можно присвоить значения. Макеты могут иметь ячейки с установленным свойством «Параметр расшифровки», значения которых задаются аналогично параметрам. Расшифровка не влияет на внешний вид результирующей таблицы и используется для создания связанных отчетов, открытия форм или иных действий.

Прежде чем формировать отчет средствами встроенного языка с использованием объекта этого типа, необходимо его создать. Это можно сделать с помощью конструктора:

```
ТабДок = Новый ТабличныйДокумент();
```

Этот объект с помощью *коллекций* предоставляет доступ к объектам типа **ОбластьЯчеекТабличногоДокумента** и **РисунокТабличногоДокумента**. Он также имеет в своем составе объект типа **КолонтитулТабличногоДокумента** для доступа к его колонтитулам.

Сам объект будет содержать по окончании работы алгоритма на встроенном языке *результурующий табличный документ*, который может быть выведен в форму, просмотрен, распечатан и даже сохранен пользователем в виде отдельного файла.

Построение отчета возможно с использованием любого табличного документа как с установленным свойством «Макет», так и в том случае, если у исходного табличного документа это свойство выключено. Кроме того, возможно непосредственное формирование отчета, используя свойства и методы данного объекта.

В общем случае методику формирования отчета можно представить следующим образом:

```
ТабДок = Новый ТабличныйДокумент();  
// алгоритм формирования табличного документа  
ТабДок.Показать();
```

При этом метод **.Показать()** используется для отображения результата работы алгоритма в окне программы. В этом случае согласно терминологии, которая используется в документации к программе, можно говорить о том, что по этому методу в окне будет отображен результирующий табличный документ.

Впрочем, в конце работы алгоритма Вы можете использовать методы:

- ✓ **Записать(<Имя файла>, <Тип файла таблицы>)** для сохранения результатов работы во внешнем файле;
- ✓ **Напечатать(<Диалог печати>)** для вывода отчета на печать вместо метода **.Показать()**.

В этой главе мы рассмотрим формирование отчета как с использованием макета, так и без него.

### 6.3. ФОРМИРОВАНИЕ ОТЧЕТА С ИСПОЛЬЗОВАНИЕМ МАКЕТА

1. Одним из способов формирования результирующего табличного документа является использование заранее составленного макета.
2. Такой макет может как располагаться в составе конфигурации, так и быть внешним файлом табличного документа. Рассмотрение последнего варианта выходит за рамки нашего курса.
3. Макет или *исходный табличный документ* может располагаться как в общих объектах конфигурации, т.е. принадлежать глобальному контексту, так и внутри прикладного объекта в конфигурации.
4. Если макет находится в составе прикладного объекта метаданных, то тогда получить макет можно, используя метод **ПолучитьМакет()** соответствующего менеджера объекта метаданных:

```
ПолучитьОбщийМакет():  
//Получение макета  
Макет = ПолучитьОбщийМакет(«ИмяМакета»);
```

#### 6.3.1. Создание общего макета

Формальным признаком того, что макет стоит делать общим, является его использование (обращение к нему) из разных модулей в конфигурации. В этом разделе мы рассмотрим (в учебных целях) создание общего макета. О том, как использовать макет в составе прикладного объекта, мы расскажем в следующих юнитах.



Для того чтобы создать общий макет, необходимо в ветви «Общие» выбрать поддерево «Общие макеты» и, нажав на правую кнопку мыши, пункт «Добавить» (рис. 6.4).

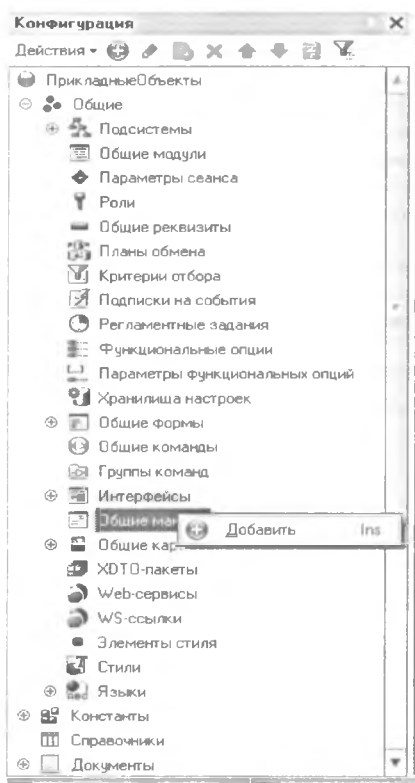


Рис. 6.4. Создание нового общего макета

Сразу после вышеописанных действий система откроет окно Конструктора макета.

### 6.3.2. Конструктор макета для создания табличного документа

С помощью этого конструктора можно создать макет (рис. 6.5):

- ✓ Табличного документа;
- ✓ Двоичных данных;
- ✓ Active document;
- ✓ HTML-документ.

В этой главе мы поговорим только об использовании табличного документа.

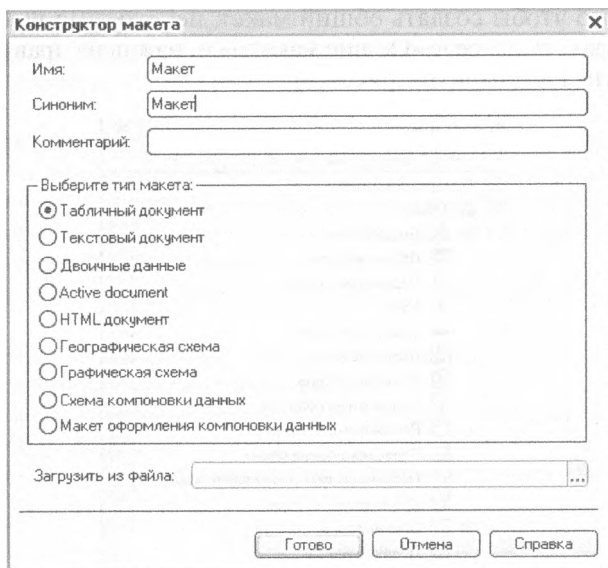


Рис. 6.5. Конструктор макета

## Упражнение № 6.2

Создайте новый общий макет и задайте в конструкторе следующие его свойства:

Свойство	Значение
Имя	КурсУЕ
Синоним	КурсУЕ
Комментарий	Табличный документ «Курс УЕ»
Выберите тип макета	Табличный документ

Нажмите на кнопку «Готово», для того чтобы завершить работу с Конструктором макета.

В результате работы Конструктора макета в дереве «Конфигурация» появится новый объект «КурсУЕ». Для удобства создания макета Конфигуратор сразу откроет окно Редактора табличного документа (рис. 6.6).

Окно Редактора табличного документа похоже на электронную таблицу. Однако это чисто визуальное сходство.

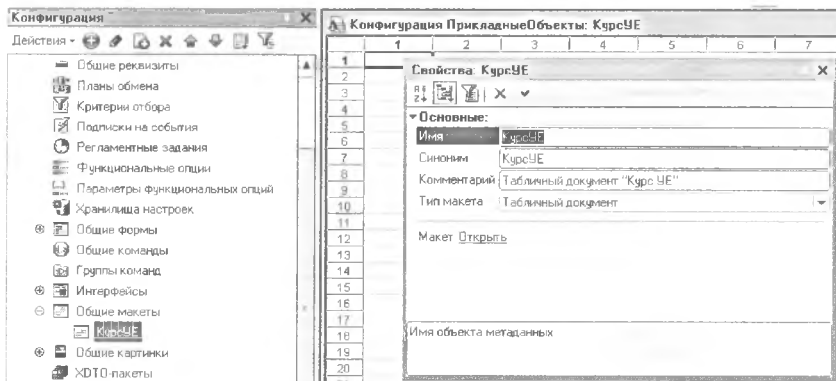


Рис. 6.6. Результат работы Конструктора макета

### 6.3.3. Свойства ячейки табличного документа

Прежде всего заметим, что каждая ячейка, как и любой объект в конфигурации, имеет свой набор свойств. Поэтому для разработки шаблона необходимо, последовательно открывая свойства соответствующих ячеек, вносить изменения в их свойства (рис. 6.7).

Каждая ячейка может содержать некоторый текст. В процессе формирования отчета этот текст с использованием методов объекта «ТабличныйДокумент» переносится из *макета* в *результатирующий табличный документ*.

В процессе такого переноса (вывода в результирующий табличный документ) система выполняет некоторые преобразования в зависимости от свойства «Заполнение» группы свойств «Макет».

Как мы видим на рисунке, возможны три варианта – «Текст», «Параметр» и «Шаблон». Что при этом происходит с текстом в каждом из случаев, мы рассмотрим ниже.

Ячейки, содержащие *текст*, при выводе в результирующий табличный документ будут перенесены без преобразования – точно так, как они заданы в макете.

В то же время ячейки, содержащие параметр или шаблон, будут преобразованы в текстовое представление согласно формату, который задан для параметров в одноименном свойстве «Формат».

Поэтому прежде чем рассматривать обращение к параметрам во встроенном языке системы, давайте создадим ячейку, содержащую параметр.

#### Упражнение № 6.3

Измените свойства левой верхней ячейки в макете «КурсУЕ» в соответствии со следующими значениями.

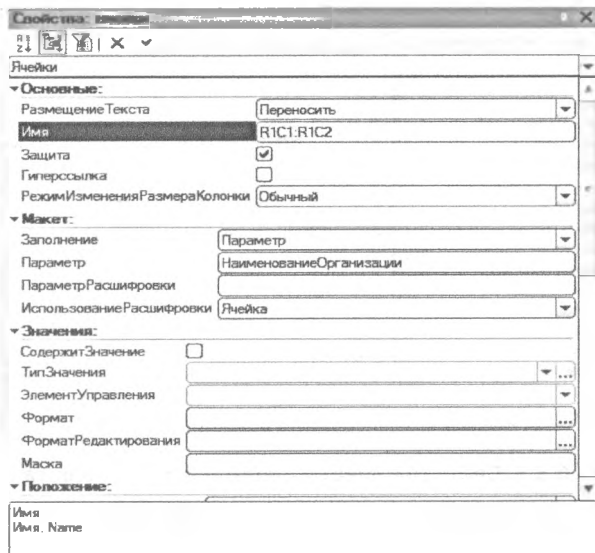


Рис. 6.7. Свойства ячейки табличного документа

Свойство	Значение
Группа свойств «Основные»	
Размещение	<b>Переносить</b>
Группа свойств «Макет»	
Заполнение	<b>Параметр</b>
Параметр	<b>НаименованиеОрганизации</b>
Группа свойств «Оформление»	
Шрифт	<b>Times New Roman, 11</b>

Остальные свойства можно оставить без изменения или выбрать по своему вкусу.

Если Вы все сделали правильно, то содержимое окна с макетом табличного документа будет выглядеть так, как это изображено на рис. 6.8.

*Параметрам*, которые заданы в ячейках макета, нужно присвоить конкретные значения до того, как часть табличного документа будет выведена в результирующий табличный документ.

Сделать это можно как с использованием именованных областей, так и без них.

В последнем случае, чтобы обратиться к свойствам ячейки, необходимо будет получить область табличного документа, используя ее адрес.

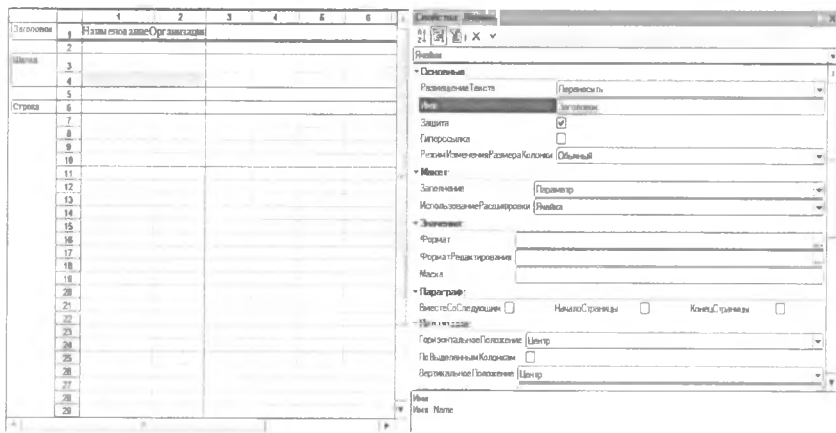


Рис. 6.8. Содержимое окна с макетом табличного документа.  
Свойства ячейки содержащей параметр

В следующем подразделе мы рассмотрим использование именованных областей для формирования результирующего табличного документа.

#### 6.3.4. Именованные области

Формирование результирующего табличного документа с использованием именованных областей заключается в последовательном переносе таких областей из исходного табличного документа.

Поэтому предварительно наш макет должен быть разбит на некоторое количество областей, каждой из которых мы можем дать некоторое *Имя*.

Область (именования) может быть отдельной ячейкой, может быть прямоугольной областью, а также целой строкой (строками) или целым столбцом (столбцами).

Поскольку мы будем формировать наш отчет построчно, наращивая его сверху вниз, то в этой главе в качестве именованной области мы используем только строки табличного документа.

Для того чтобы присвоить области имя, необходимо сначала ее выделить, затем выбрать пункт меню «Таблица – Имена – Назначить имя...». После чего Конфигуратор откроет диалог (рис. 6.9), в котором нужно дать области некоторое имя.

Выполните следующее упражнение.

#### Упражнение № 6.4

Выделите первую строку в макете «КурсыУЕ».  
Назначьте ей в качестве имени имя «Заголовок».

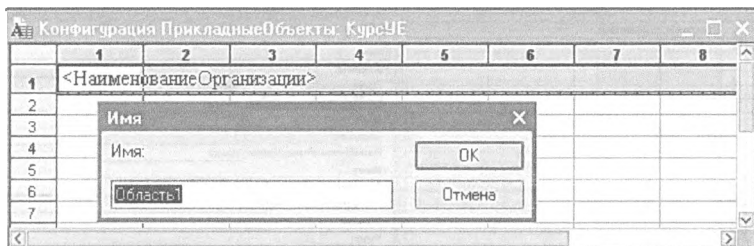


Рис. 6.9. Создание именованной области в макете

По окончании упражнения окно табличного документа примет вид, изображенный на рис. 6.10.

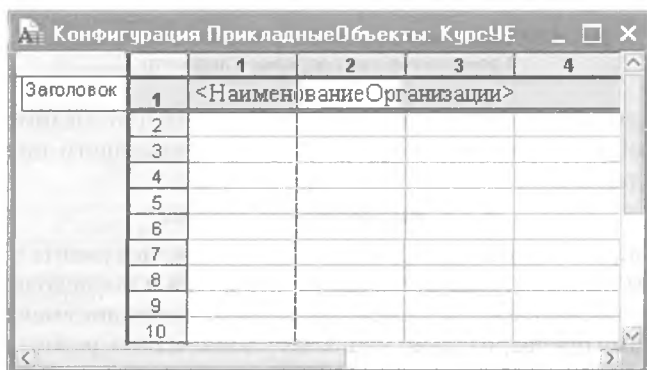


Рис. 6.10. Область Заголовок

Эту область мы поместим в результирующий табличный документ только один раз, используя для достижения этой цели методы **.ПолучитьОбласть()** и **.Вывести()** объекта «ТабличныйДокумент»:

```
//Получение части документа из макета
Область = Макет.ПолучитьОбласть(«Заголовок»);
// действия по заполнению параметров части документа...
ТабДок.Вывести(Область);
```

Метод **ПолучитьОбласть()** возвращает ссылку на объект типа **ТабличныйДокумент**, который представляет собой часть исходного табличного документа.

Для того чтобы присвоить параметрам значения части документа перед ее выводом в результирующий табличный документ, необходимо воспользоваться свойством **Параметры**:

```

ТабДок = Новый ТабличныйДокумент();
//Получение макета
Макет = ПолучитьОбщийМакет(«КурсУЕ»);
//Получение части документа из макета
Область = Макет.ПолучитьОбласть(«Заголовок»);
// действия по заполнению параметров значениями части
// документа
Область.Параметры.НаименованиеОрганизации =
Константы.НаименованиеОрганизации.Получить();
ТабДок.Вывести(Область);
//.....
ТабДок.Показать(«История курса УЕ»);

```

Если приведенный фрагмент программы добавить в процедуру **ПечатьКурсаУЕ()** и запустить ее на выполнение, то в результате будет открыто окно табличного документа, изображенное на рис. 6.11.

	1	2	3	4	5
1	Объединение "Содружество"				
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

**Рис. 6.11.** Результат вывода именованной области «Заголовок»

А пока самостоятельно создайте две области, выполнив следующее упражнение.

### Упражнение № 6.5

Создайте две области — «Шапка» и «Строка», разместив их в макете «КурсУЕ» так, как это изображено на рис. 6.12.

Самостоятельно внесите изменения в текст процедуры **ПечатьКурсаУЕ()** таким образом, чтобы она отображала курсы УЕ в табличном документе.

Проверьте работу процедуры в режиме «1С:Предприятие».

Заголовок	1	2
1	<Наименование Организации>	
2		
Шапка	История курса УЕ	
3	Дата	Значение
4		
5		
Строка	<Дата>	<Значение>
6		
7		
8		
9		
10		

Рис. 6.12. Области Шапка и Строка

Для надежности приводим текст данной процедуры (сравните со своим вариантом).

#### ЛИСТИНГ Ю\_6.1

```

Процедура ПечатьКурсаУЕ(Кнопка)
ТабДок = Новый ТабличныйДокумент();
//Получение макета
Макет = ПолучитьОбщийМакет(«КурсУЕ»);
//Получение части документа из макета
Область = Макет.ПолучитьОбласть(«Заголовок»);
// действия по заполнению параметров части документа
Область.Параметры.НаименованиеОрганизации =
Константы.НаименованиеОрганизации.Получить();
ТабДок.Вывести(Область);
Область = Макет.ПолучитьОбласть(«Шапка»);
ТабДок.Вывести(Область);
Область = Макет.ПолучитьОбласть(«Строка»);
РегСведКурсы=РегистрыСведений.Курсы;
НабКурсы=РегСведКурсы.СоздатьНаборЗаписей();
НабКурсы.Прочитать();
к=НабКурсы.Количество();
Для н=1 по НабКурсы.Количество() Цикл
ТекСтрока=НабКурсы[к-н];

```



```

//Сообщить(«»+ТекСтрока.Период+Символы.Таб+ТекСтрока.Курс);
Область.Параметры.Дата = ТекСтрока.Период;
Область.Параметры.Значение = ТекСтрока.Курс;
ТабДок.Вывести(Область);
КонецЦикла;
// Установка свойств табличного документа в целом перед его
// отображением: назначение этих параметров рассмотрим позднее
ТабДок.ИмяПараметровПечати = «ПараметрыКурсаУЕ»;
ТабДок.ИмяСохраненияПоложенияОкна = «ОкноКурсаУЕ»;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.ОтображатьСетку = Ложь;
ТабДок.СохранятьСвойстваОтображения = Истина;
ТабДок.ТолькоПросмотр = Истина;
ТабДок.ФиксацияСверху = 3;
//
ТабДок.Показать(«История курса УЕ»);
КонецПроцедуры

```

#### 6.4. УЛУЧШЕНИЕ ВИДА РЕЗУЛЬТИРУЮЩЕГО ТАБЛИЧНОГО ДОКУМЕНТА

Дата	Значение
18.08.2009 0:00:С	23,3532
30.07.2009 0:00:С	23,245
30.06.2009 0:00:С	23,3483
31.05.2009 0:00:С	23,709
01.05.2009 0:00:С	23,1021
01.04.2009 0:00:С	23,3801
02.03.2009 0:00:С	23,5729
31.01.2009 0:00:С	23,6222
01.01.2009 0:00:С	23,7844
01.01.2008 0:00:С	24,85
28.12.2007 0:00:С	25,26

Рис. 6.13. Первый вариант работы процедуры ПечатьКурсаУЕ()

Если Вы правильно выполнили предыдущее упражнение, то в режиме «1С:Предприятие» результат работы процедуры **ПечатьКурсаУЕ()** будет выглядеть примерно так, как это изображено на рис. 6.13.

Очевидно, что сформированный таким образом табличный документ обладает рядом недостатков.

1. Наименование организации с трудом помещается в одну ячейку.

2. Ширина столбца для колонок «Дата» и «Значение» не достаточна для отображения необходимой информации.

3. Числовые значения курса УЕ должны печататься с четырьмя знаками после запятой.

Последующие подразделы позволят решить указанные выше проблемы.

#### 6.4.1. Объединение ячеек

Наименование организации в отчете будет выглядеть лучше, если мы объединим две соседних ячейки.

Для этого, используя мышь или клавиатуру, выделите в макете КурсУЕ две первых ячейки. После чего в основном меню программы выберите пункт «Таблица – Объединить».

После таких действий эти две ячейки станут одной. Поэтому ее свойства следует «слегка» изменить, чтобы в режиме «1С:Предприятие» улучшить ее вид.

#### Упражнение № 6.6

Измените следующие свойства ячейки.

Группа свойств «Положение:»

Свойство	Значение
По горизонтали	Центр
По вертикали	Центр

Для закрепления этого приема разработки макетов выполните следующее упражнение.

#### Упражнение № 6.7

В области «Шапка» объедините две ячейки, которые должны содержать текст «История курса УЕ», и установите для этой ячейки те же свойства, что и в предыдущем упражнении.

Что изменилось в результате таких действий?

Как будет выглядеть теперь результирующий табличный документ в режиме «1С:Предприятие»?

## 6.4.2. Изменение ширины столбцов

Если выделить первый столбец нашего макета и, используя правую кнопку мыши, выбрать пункт контекстного меню «Ширина колонок...», то на экране появится одноименный диалог (рис. 6.14).

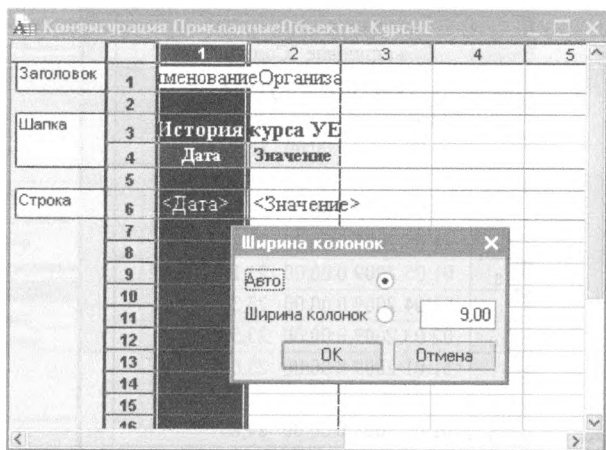


Рис. 6.14. Настройка ширины колонок

В этом диалоге мы можем указать ширину столбца в явном виде.

Единицей измерения, согласно документации, в данном случае служит *ширина символа шрифта по умолчанию*. Поэтому величина «9,00» означает, что по умолчанию девять символов шрифта с гарантией впишутся в ячейку этой колонки.

А какой шрифт является шрифтом по умолчанию?

Сразу после создания в конфигурации макета для него устанавливается шрифт *Arial высотой 8 пунктов*.

А у нас текстовое представление параметра Дата будет иметь 10 символов, при отображении которых мы использовали шрифт высотой 11 пунктов. Поэтому нам нужно установить ширину первого столбца, равной:  $10 \times 11/8 = 13,25$ . Округлим результат до 13 и выполним следующее упражнение.

### Упражнение № 6.8

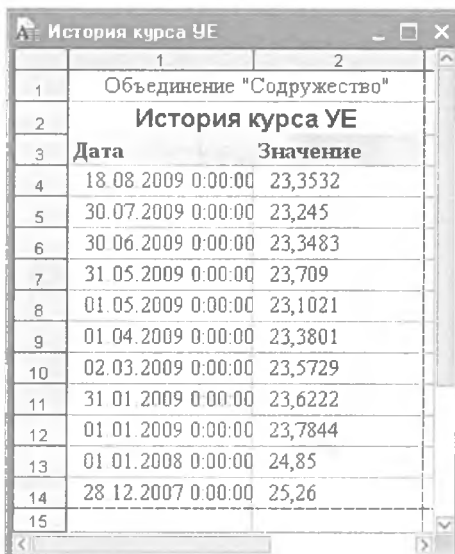
Установите ширину первой колонки равной 13. Рассчитайте минимальную ширину второй колонки. Измените ее ширину, используя диалог «Ширина колонок».

Какую ширину Вы установили для второй колонки?

Как будет выглядеть теперь результирующий табличный документ в режиме «1С:Предприятие»?

### 6.4.3. Форматирование содержимого ячейки

В результате выполнения упражнения № 6.8 окно табличного документа примет вид, изображенный на рис. 6.15.



	1	2
1	Объединение "Содружество"	
2	История курса УЕ	
3	Дата	Значение
4	18.08.2009 0:00:00	23,3532
5	30.07.2009 0:00:00	23,245
6	30.06.2009 0:00:00	23,3483
7	31.05.2009 0:00:00	23,709
8	01.05.2009 0:00:00	23,1021
9	01.04.2009 0:00:00	23,3801
10	02.03.2009 0:00:00	23,5729
11	31.01.2009 0:00:00	23,6222
12	01.01.2009 0:00:00	23,7844
13	01.01.2008 0:00:00	24,85
14	28.12.2007 0:00:00	25,26
15		

Рис. 6.15. Результат выполнения упражнения № 6.8


Глядя на этот рисунок, мы можем сделать выводы о *текстовом представлении* для данных типа Дата и Число по умолчанию.

По умолчанию при отображении типа данных Дата вне зависимости от ее состава текстовая строка будет содержать как дату, так и время.

При формировании текстового представления числа отбрасываются все пробелы слева и все незначащие нули справа.

Для явного указания формата, который будет использоваться при преобразовании значений параметров в текст ячейки, служит поле «Формат» группы свойств «Значения:».

В это поле следует вводить строку форматирования таким же образом, как это описано для функции **Формат(...)** встроенного языка системы.

Для упрощения заполнения этого свойства следует использовать кнопку выбора , которая есть в этом поле окна «Свойства», нажатие на эту кнопку открывает диалог «Редактор форматной строки».

Если мы откроем такой редактор для ячейки, которая содержит параметр «Дата», то, переключившись на закладку «Дата», мы сможем выбрать необходимый формат даты (рис. 6.16).

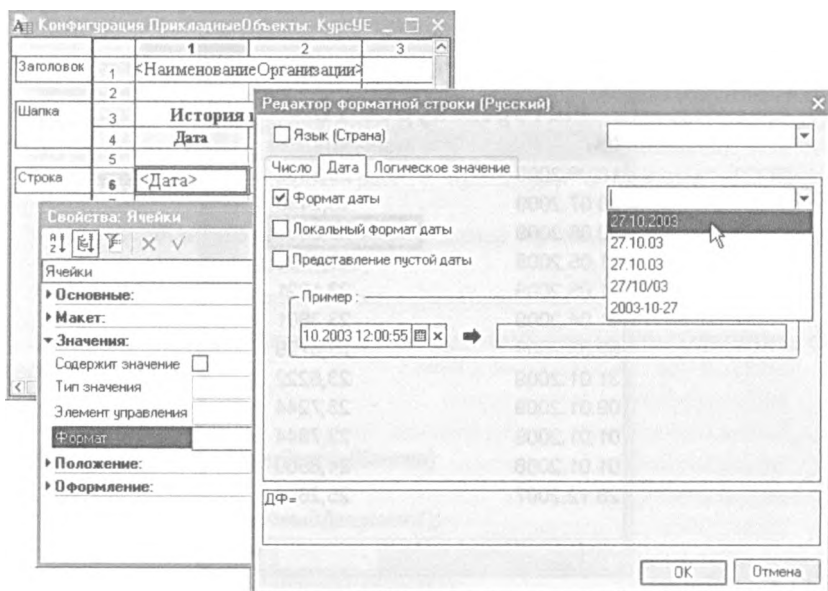


Рис. 6.16. Выбор формата для представления даты

### Упражнение № 6.9

Установите, используя диалог «Редактор форматной строки», необходимый формат для представления ячеек, содержащих параметры Дата и Значение.

Какие форматные строки Вы сформировали и почему?

Как будет выглядеть теперь результирующий табличный документ в режиме «ИС:Предприятие»?

В результате выполнения упражнения № 6.9 окно табличного документа примет вид, изображенный на рис. 6.17.

#### 6.4.4. Установка свойств табличного документа

Перед отображением результирующего табличного документа в окне программы (или перед его печатью) можно, а часто даже нужно установить свойства этого документа в целом.

Так, например, в процедуре **ПечатьКурсаУЕ()** перед применением метода **.Показать()** мы можем установить следующие свойства:

```
// Установка свойств табличного документа в целом
// перед его отображением:
```

История курса УЕ

Объединение "Содружество"

**История курса УЕ**

Дата	Значение
18.08.2009	23,3532
30.07.2009	23,2450
30.06.2009	23,3483
31.05.2009	23,7090
01.05.2009	23,1021
01.04.2009	23,3801
02.03.2009	23,5729
31.01.2009	23,6222
09.01.2009	23,7244
01.01.2009	23,7844
01.01.2008	24,8500
28.12.2007	25,2600

Рис. 6.17. Результат выполнения упражнения № 6.9

ТабДок.ИмяПараметровПечати = «ПараметрыКурсаУЕ»;  
 ТабДок.ИмяСохраненияПоложенияОкна = «ОкноКурсаУЕ»;  
 ТабДок.ОтображатьЗаголовки = Ложь;  
 ТабДок.ОтображатьСетку = Ложь;  
 ТабДок.СохранятьСвойстваОтображения = Истина;  
 ТабДок.ТолькоПросмотр = Истина;  
 ТабДок.ФиксацияСверху = 3;  
 ТабДок.Показать(«История курса УЕ»);

#### Упражнение № 6.10

Внесите в процедуру **ПечатьКурсаУЕ()** текст, приведенный выше, и проверьте ее работоспособность.

Как будет выглядеть теперь результирующий табличный документ в режиме «1С:Предприятие»?

## 6.5. САМОСТОЯТЕЛЬНАЯ РАБОТА

Для закрепления только что рассмотренного материала выполните следующее упражнение самостоятельно.

## Упражнение № 6.11

---

Внесите необходимые изменения в текст процедуры **ПечатьЧерезЗапрос(Кнопка)** модуля формы списка регистра сведений «Курсы», чтобы процедура формировала отчет, аналогичный рассмотренному выше.

Обязательно проверьте работу процедуры в режиме «1С:Предприятие».

По окончании отладки замените номер версии в свойствах конфигурации на величину «Юнит 6».

---

Для надежности приводим текст данной процедуры (сравните со своим вариантом).

### ЛИСТИНГ Ю\_6.2

#### Процедура ПечатьЧерезЗапрос(Кнопка)

```
ТабДок = Новый ТабличныйДокумент();
//Получение макета
Макет = ПолучитьОбщийМакет(«КурсУЕ»);
//Получение части документа из макета
Область = Макет.ПолучитьОбласть(«Заголовок»);
// действия по заполнению параметров части документа
Область.Параметры.НаименованиеОрганизации =
Константы.НаименованиеОрганизации.Получить();
ТабДок.Вывести(Область);
Область = Макет.ПолучитьОбласть(«Шапка»);
ТабДок.Вывести(Область);
Область = Макет.ПолучитьОбласть(«Строка»);
//Создаем объект запрос
Запрос = Новый Запрос();
//Формируем текст запроса
Запрос.Текст =
«ВЫБРАТЬ Курсы.Период, Курсы.Курс
|ИЗ РегистрСведений.Курсы КАК Курсы
|ГДЕ Курсы.Период >= &Период
|УПОРЯДОЧИТЬ ПО Период УБЫВ»;
//Устанавливаем значение параметра
Запрос.УстановитьПараметр(«Период»,
Константы.ДатаРегистрации.Получить());
//И выполняем запрос
Результат = Запрос.Выполнить();
// Обработка результата
```

```
Выборка=Результат.Выбрать();
Пока Выборка.Следующий() Цикл
//Сообщить(«»+Выборка.Период+Символы.Таб+Выборка.Курс);
Область.Параметры.Дата = Выборка.Период;
Область.Параметры.Значение = Выборка.Курс;
ТабДок.Вывести(Область);
КонецЦикла;
// Установка свойств табличного документа в целом
// перед его отображением:
ТабДок.ИмяПараметровПечати = «ПараметрыПечатиУЕ»;
ТабДок.ИмяСохраненияПоложенияОкна = «ОкноПечатиУЕ»;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.ОтображатьСетку = Ложь;
ТабДок.СохранятьСвойстваОтображения=Истина;
ТабДок.ТолькоПросмотр = Истина;
ТабДок.ФиксацияСверху = 3;
ТабДок.Показать(«История курса УЕ»);
КонецПроцедуры
```

## 6.6. ФОРМИРОВАНИЕ ОТЧЕТА БЕЗ ИСПОЛЬЗОВАНИЯ МАКЕТА

Объект **ТабличныйДокумент** позволяет строить отчеты без использования макета.

Пример, иллюстрирующий построение отчета без использования макета, можно найти в приложении В.

## 6.7. ЧТО МЫ УЗНАЛИ

В этой главе мы рассмотрели основы формирования отчетов с использованием объекта «ТабличныйДокумент».

Для формирования отчета мы использовали макет, который расположили в общих макетах конфигурации. Создание макетов в Конфигураторе происходит с использованием специального Конструктора. Сам макет представляет собой табличный документ с включенным свойством «Макет». Наличие этого режима делает возможным изменение в процессе его редактирования группы свойств «Макет». В эту группу входят свойства «Заполнение» и «Параметр», которые мы рассмотрели в этой главе.

Обычно при проектировании отчета его макет разбивают на отдельные области, каждой из которых дают *имя*. Это имя используют в дальнейшем для получения части макета (в качестве объекта типа «ТабличныйДокумент»), который выводят в *результатирующий табличный документ*.



Редактор табличных документов, который входит в состав программной части системы, содержит достаточный набор функций, позволяющий создать практически любое необходимое оформление (форматирование) документа.

В этой главе мы использовали формирование табличного документа, содержащего отчет из постановки задачи, вызываемый из формы списка регистра сведений «Курсы».

В этой главе мы рассмотрим, каким образом, используя возможности Конфигуратора, можно создавать новые справочники.

Справочники в системе «1С:Предприятие» предназначены для хранения нормативно-справочной информации. Записи в справочнике подобны карточкам в картотеке. Каждая запись имеет одинаковую структуру. Пользователь может поместить в справочник практически неограниченное число записей.

Существуют ситуации, когда пользователь должен сделать выбор из конечного числа значений, причем добавлять или изменять список этих значений нельзя. В такой ситуации специалист, осуществляющий конфигурирование системы «1С:Предприятие», должен создать Перечисление. В отличие от справочника, значения в перечислениях принадлежат конфигурации, поэтому пользователь в процессе своей работы не может изменить их состав.

В этой главе мы создадим, в соответствии с постановкой задачи, необходимое количество справочников и одно перечисление.

Для создания нового справочника необходимо выделить в дереве окна «Конфигурация» ветвь «Справочники» и, используя правую

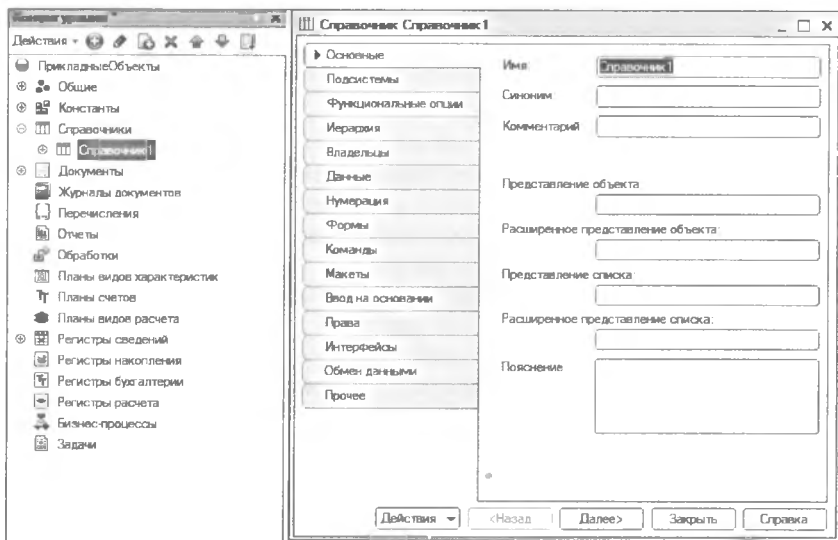


Рис. 7.1. Окно редактирования справочника

кнопку мыши, выбрать пункт выпадающего меню «Добавить». Сразу после этого Конфигуратор создаст в дереве новый объект «Справочник1» и откроет его окно редактирования (рис. 7.1).

Поскольку работа с таким окном нам уже знакома, то в этой главе мы сосредоточимся на создании конкретных справочников, а возможности, которые предоставляет окно редактирования справочника, будут рассмотрены на основе конкретных примеров.

## 7.1. СПРАВОЧНИК «ФИРМЫ»

В соответствии с п. 4 постановки задачи мы создадим новый справочник, выполнив упражнение № 7.1.

### Упражнение № 7.1

---

---

Создайте в конфигурации новый справочник со следующими свойствами:

Свойство	Значение
Имя	Фирмы
Синоним	Фирмы
Комментарий	Список организаций объединения

Перейдите к закладке «Данные».

---

---

#### 7.1.1. Закладка «Данные»

Эта закладка позволяет определить, что за реквизиты будут содержать записи справочника (рис. 7.2).

Справочник может иметь следующие реквизиты с заранее заданным назначением:

- ✓ Код;
- ✓ Наименование;
- ✓ Родитель;
- ✓ Владелец.

*Код* представляет собой краткое название объекта учета, обычно это его номер или внутренний учетный код. Код может быть числовым или текстовым.

Эта закладка позволяет настроить длину как кода, так и наименования. При выборе длины кода нужно руководствоваться:

- ✓ сколько элементов должен вмещать справочник;
- ✓ каким образом мы собираемся кодировать элементы.

Максимальная длина кода в справочнике — 50 символов.

*Наименование* — это название объекта учета. Обычно это название используется для быстрого поиска объекта по его названию.

Максимальная длина, которую можно установить для наименования, — 150.

Свойства «Родитель» и «Владелец» будут рассмотрены позже.

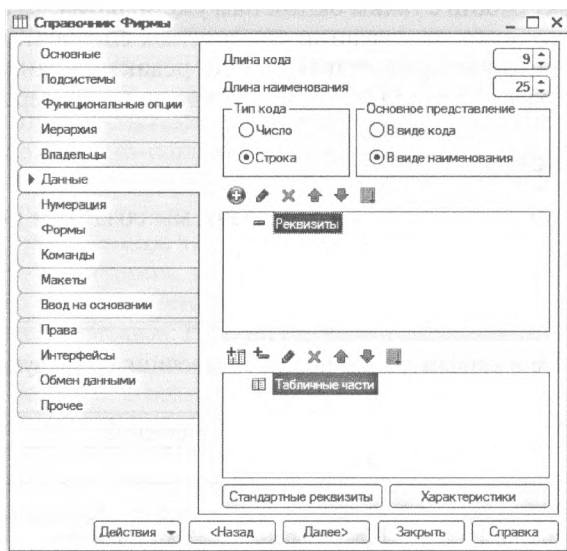


Рис. 7.2. Закладка «Данные»

На данном этапе создания конфигурации мы оставим содержимое этой закладки без изменения.

### 7.1.2. Закладка «Нумерация»

Эта закладка позволяет настроить свойства справочника, связанные с нумерацией реквизита «Код» (рис. 7.3).

Включенный флаг «Автонумерация» показывает, что новая запись, вводимая в справочник, будет иметь код на 1 больший, чем максимальное значение, которое было введено ранее.

Контроль уникальности, если флаг установлен, гарантирует, что в пределах серии кодов (настройка серий кодов будет рассмотрена позднее) не будет двух записей с одним кодом. Если этот флаг будет сброшен, то пользователь сможет поместить две (и более) имеющие одинаковый код, а возможно, и все другие реквизиты записи. Однако в этом случае эти две записи будут для системы тем не менее разными объектами. Система использует для контроля ссылочной целостности не Код, а внутреннее недоступное для нас поле. Именно по этой причине пользователь может свободно (если это не запрещено конфигурацией) изменять содержимое реквизита «Код», не опасаясь разорвать ссылки между объектами.

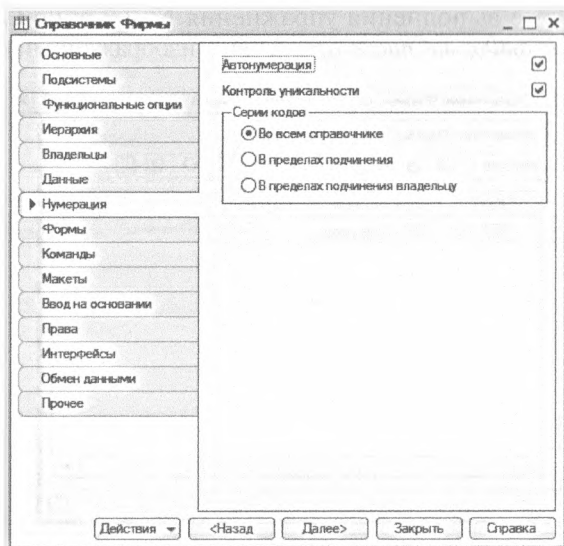


Рис. 7.3. Закладка «Нумерация»

### 7.1.3. Закладка «Формы»

Эта закладка нам уже знакома. Она позволяет создать формы справочника в необходимом для работы пользователя количестве.

#### Упражнение № 7.2

В окне редактирования справочника «Фирмы», используя кнопку открытия, создайте форму списка (она же форма выбора), установив в Конструкторе следующие свойства:

Свойство	Значение
Выберете тип формы	<b>Форма списка справочника</b>
Назначить форму основной	<input checked="" type="checkbox"/>
Основная форма списка и форма выбора	<input checked="" type="checkbox"/>
Имя	<b>ОсновнаяФормаСписка</b>
Синоним	<b>Основная форма списка</b>
Комментарий	<b>Форма списка и выбора</b>
Командная панель формы сверху	<input checked="" type="checkbox"/>
Командная панель формы снизу	<input checked="" type="checkbox"/>

Нажмите на кнопку «Готово».

В результате выполнения упражнения № 7.2 окно *Справочника Фирмы: ОсновнаяФормаСписка* примет вид, изображенный на рис. 7.3а.

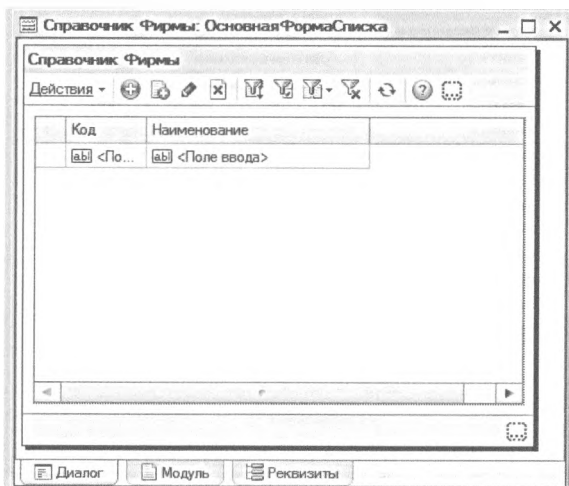


Рис. 7.3а. Результат выполнения упражнения № 7.2

#### 7.1.4. Закладка «Макеты»

Назначение этой закладки нам уже известно — она позволяет создать макет печатной формы справочника.

В этом разделе мы воспользуемся конструктором печати, чтобы сформировать макет печатной формы списка справочника «Фирмы».

#### Упражнение № 7.3

---

Выберите закладку «Макеты».

Используя кнопку «Конструкторы», выберите «Конструктор печати...».

В открывшемся окне диалога в группе «Создать новую процедуру» расположите переключатели так, как это изображено на рис. 7.4.

---

На этом этапе мы указали конструктору, что хотим иметь процедуру, которая распечатывала бы форму списка справочника. Причем она будет осуществлять печать списка фирм и располагаться в основной форме списка.

#### Упражнение № 7.3 (продолжение)

---

Нажмите на кнопку «Далее», чтобы продолжить работу с конструктором печати. Используя следующий шаг диалога, укажите в качестве рек-

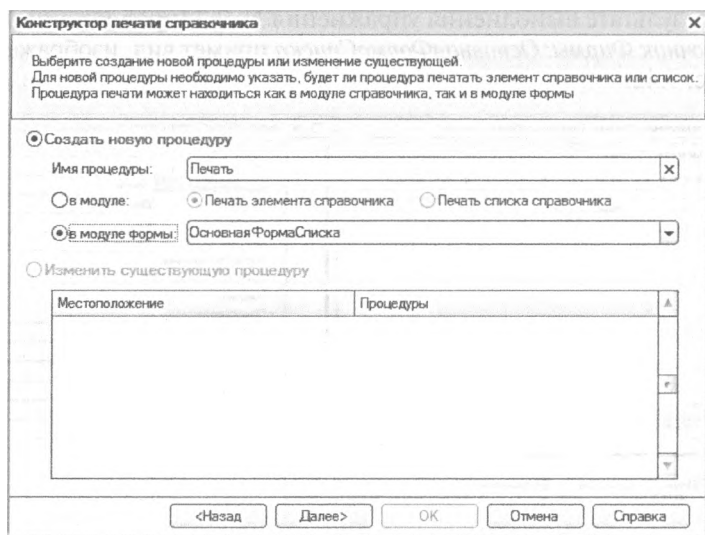


Рис. 7.4. Первый шаг конструктора печати справочника

визитов шапки реквизиты «Код» и «Наименование». Нажмите кнопку «Далее».

И на последнем этапе укажите название кнопки и свойства объекта **ТабличныйДокумент**:

Свойство	Значение
Вызов процедуры	<b>Не вставлять вызов процедуры</b>
Печать без предварительного просмотра	—
Таблица «Только для чтения»	<input checked="" type="checkbox"/>
Защита таблицы	—

Нажмите на кнопку ОК, чтобы завершить работу с конструктором.

Какие окна открыл Конструктор после окончания своей работы?

Добавьте в нижнюю командную панель формы кнопку «Печать».

Выберите для ее свойства «Действие» (из выпадающего списка) вызов процедуры Печать. Установите картинку «Печать» из стандартной коллекции и поменяйте отображение кнопки на «Надпись и картинка».

Откройте свойства формы в целом и установите:

Свойство	Значение
Исп. только видимую область	<input checked="" type="checkbox"/>

В результате выполнения упражнения № 7.3 (продолжение) окно *Справочник Фирмы: ОсновнаяФормаСписка* примет вид, изображенный на рис. 7.4а.

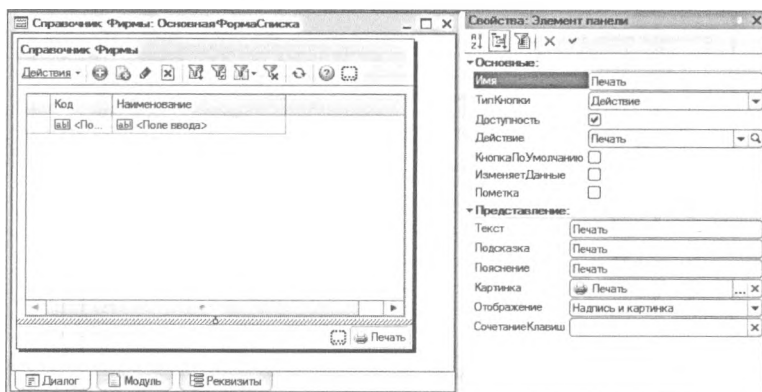


Рис. 7.4а. Результат выполнения упражнения № 7.3 (продолжение)

### 7.1.5. Закладка «Интерфейсы»

Эта закладка позволяет вставить необходимый пункт меню в интерфейс конфигурации, используя для этой цели окно редактирования справочника (рис. 7.5).

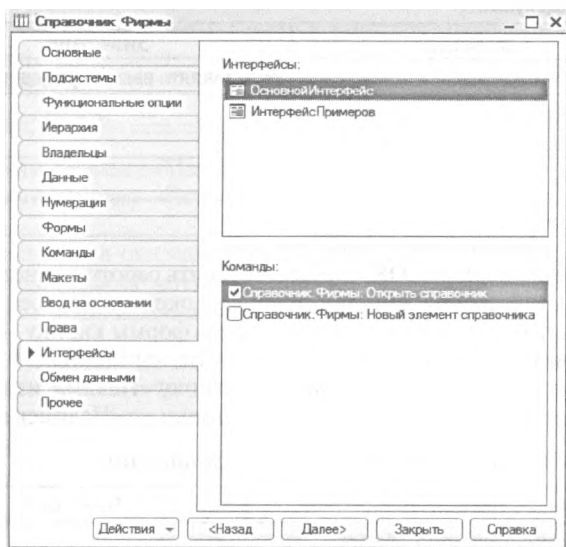


Рис. 7.5. Закладка «Интерфейсы» справочника «Фирмы»



## Упражнение № 7.4

Используя окна редактирования справочников, вставьте в основной интерфейс команду «Фирмы: Открыть справочник».

Откройте «ОсновнойИнтерфейс» и проверьте, соответствует ли интерфейс рис. 7.6. Если нет, то исправьте свойство «Текст» у только что созданного пункта.

Сохраните и обновите конфигурацию.

На этом мы пока закончим настройку справочника «Фирмы».

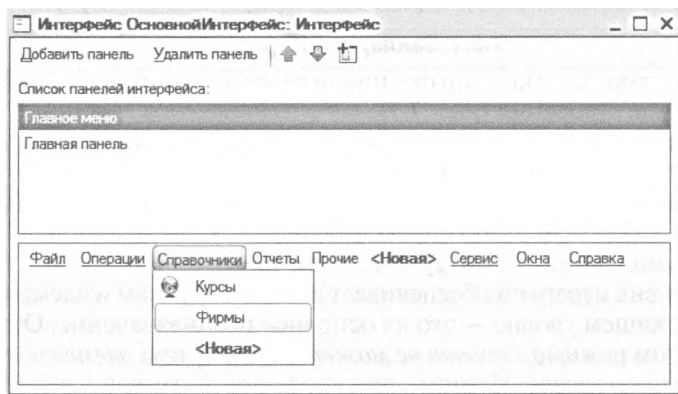


Рис. 7.6. Основной интерфейс после выполнения упражнения № 7.4

## 7.2. СПРАВОЧНИК «ПОДРАЗДЕЛЕНИЯ»

В п. 5 постановки задачи сказано, что «каждая фирма в своем составе может иметь несколько подразделений, каждое из которых, в свою очередь, также может состоять из нескольких подразделений...».

Для хранения такой информации в информационной базе нам понадобится иерархический справочник.

## Упражнение № 7.5

Создайте в конфигурации новый справочник со следующими свойствами:

Свойство	Значение
Имя	Подразделения
Синоним	Подразделения
Комментарий	—

Нажмите на кнопку «Далее>», чтобы перейти к следующему этапу создания справочника.

Справочник «Фирмы» из предыдущего раздела был одноуровневым. Записи, которые будут размещаться в нем, будут только *элементами*, что и понятно, так как все фирмы равноправны. Каждый элемент в справочнике соответствует одной инвентарной карточке или одному объекту аналитического учета.

Если между объектами учета *одного вида* существует отношение подчиненности, то такой справочник следует сделать иерархическим.

### 7.2.1. Закладка «Иерархия»

За настройку таких справочников отвечает одноименная закладка окна редактирования (рис. 7.7).

Установка свойства «Иерархический справочник» делает доступными остальные свойства этой группы.

Следующее свойство «Вид иерархии» позволяет определить, какие записи может содержать справочник.

По умолчанию — это *группы и элементы*.

Этот вид иерархии обеспечивает доступ к группам и элементам на нижележащем уровне — это их основное предназначение. Отметим, что в этом режиме *элемент не может иметь других элементов на нижележащем уровне*. Поэтому при вводе новой записи в справочник нужно заранее выбрать, что это будет — *элемент* или *группа*?

В случае если это свойство установлено в положение «Иерархия элементов», то тогда справочник будет содержать *только элементы*. Причем *элемент может иметь другие элементы на нижележащем уровне*. Следствием из этого утверждения будет то, что в справочник можно будет ввести только элементы.

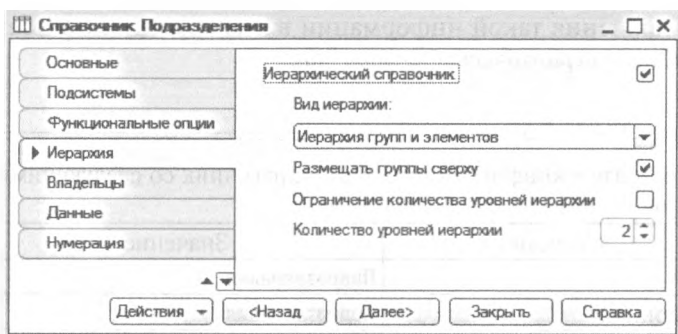


Рис. 7.7. Свойства иерархических справочников

Свойство «Ограничение уровней иерархии» позволяет ограничить глубину вложенностей путем установки свойства «Количество уровней иерархии». Если такое количество можно заранее оценить, то флажок следует установить и указать максимальное количество уровней от 2 до 10. В постановке задачи (п. 5) сказано, что количество уровней заранее неизвестно. Да и, кроме того, организационная структура организации может изменяться с течением времени — подразделения могут дробиться на более мелкие или объединяться, поэтому продолжим выполнение упражнения.

#### Упражнение № 7.6

---

Установите для справочника «Подразделения» следующие свойства:

Свойство	Значение
Иерархический справочник	<input checked="" type="checkbox"/>
Вид иерархии	<b>Иерархия элементов</b>
Ограничение количества уровней иерархии	—

Нажмите на кнопку «Далее>», чтобы перейти к следующему этапу создания справочника.

---

### 7.2.2. Закладка «Владельцы»

Справочник в системе «1С:Предприятие» *должен содержать сущности одной природы*.

Для того чтобы подчеркнуть подчиненность объекта учета одного вида объектами другого вида, необходимо определить для справочника состав его *владельцев*.

В самом деле подразделения «в воздухе не висят», а всегда относятся какой-либо «фирме». Поэтому для справочника «Подразделения» можно указать, что его элементами будут владеть записи из справочника «Фирмы».

#### Упражнение № 7.7

---

Используя кнопку окна редактирования справочника «Подразделения», укажите в качестве владельца Справочник «Фирмы».

Если вы все сделали правильно, то окно редактирования примет вид, изображенный на рис. 7.8.

---

Дадим небольшое пояснение значениям свойства «Использование подчинения». По умолчанию для подчиненных справочников

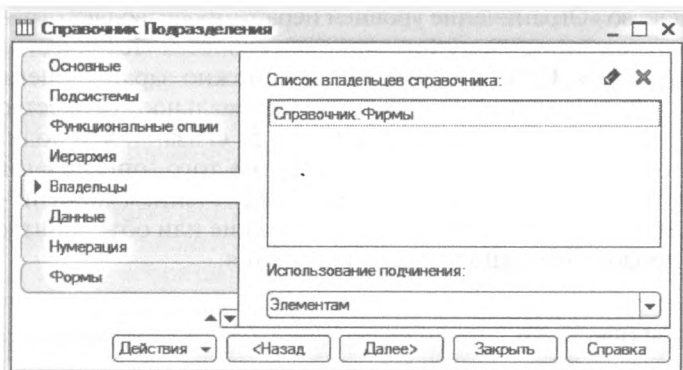


Рис. 7.8. Список владельцев справочника

это свойство установлено в положение «Элементам». Это приводит к тому, что свойство «Владелец» из данного справочника будет содержать только ссылки на элементы из справочников-владельцев.

Соответственно значение «Группам» приведет к тому, что это свойство сможет содержать только ссылки на группы из справочников-владельцев.

Значение «Группам и элементам» позволит пользователю выбирать в качестве значения свойства «Владелец» ссылки на любую запись из справочников-владельцев.

В случае справочника «Подразделения» единственно возможное значение — значение «Элементам». Почему?

### 7.2.3. Закладка «Нумерация»

В этом разделе мы подробнее рассмотрим переключатель «Серии кодов», который расположен на закладке «Нумерация» (рис. 7.9).

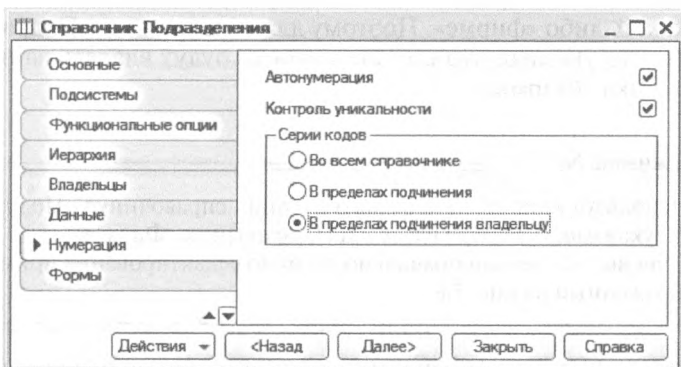


Рис. 7.9. Настройка серий кодов на закладке «Нумерация»

Прежде чем понять, как будет система нумеровать коды в справочнике при каждом из значений этого переключателя, вспомним, что у справочника есть два поля (или с точки зрения встроенного языка — свойств) — Родитель и Владелец.

Свойство «Родитель» содержит ссылку на запись в этом справочнике, которая находится на один уровень выше.

А свойство «Владелец» содержит ссылку на запись в другом справочнике.

Таким образом, запись из текущего справочника может быть подчинена как своему «Родителю», так и владельцу из другого справочника.

В случае если свойство «Серии кодов» установлено в положение «Во всем справочнике», то нумерация кодов будет выполняться системой *без учета значений как родителя, так и владельца*. Таким образом, это будет сквозная нумерация от «1» для всех записей в справочнике в порядке их создания, причем совсем не важно, на каком уровне вложенности вводится запись и какой у нее владелец.

Для положения «В пределах подчинения» нумерация кодов будет выполняться системой *с учетом значений как родителя, так и владельца*. Такая нумерация позволяет начинать нумерацию с «1» для подчиненных записей как для нового Владельца, так и для нового Родителя.

Для положения «В пределах подчинения владельцу» нумерация кодов будет выполняться системой *с учетом значения только владельца*. Такой способ позволяет организовать сквозную нумерацию кодов для каждого владельца в отдельности, причем совсем не важно, на каком уровне вложенности вводится запись.

По умолчанию свойство установлено в положение «Во всем справочнике», поэтому выполним упражнение.

### Упражнение № 7.8

---

---

Установите на закладке «Нумерация» переключатель «Серии кодов» в положение «В пределах подчинения владельцу».

---

---

### 7.2.4. Закладка «Формы»

Эта закладка нам уже знакома. Она позволяет создать формы справочника в необходимом для работы пользователя количестве.

### Упражнение № 7.9

---

---

В окне редактирования справочника «Подразделения» установите свойство «Способ редактирования» в положение «В диалоге».

После чего создайте основную форму списка со следующими свойствами (в конструкторе):

Свойство	Значение
Выберите тип формы	<b>Форма списка справочника</b>
Назначить форму основной	<input checked="" type="checkbox"/>
Основная форма списка и форма выбора	—
Имя	<b>ОсновнаяФормаСписка</b>
Синоним	<b>Основная форма списка</b>
Комментарий	—
Командная панель формы сверху	<input checked="" type="checkbox"/>
Командная панель формы снизу	<input checked="" type="checkbox"/>

Нажмите на кнопку «Далее» и установите на следующем шаге Конструктора флаг «Владелец» у реквизита «Список» и завершите работу.

При создании формы выбора мы предусмотрим, чтобы пользователь делал выбор не из табличной части, а из дерева.

#### Упражнение № 7.10

Для справочника «Подразделения» создайте основную форму выбора, указав на первом шаге конструктора следующие значения:

Свойство	Значение
Выберите тип формы	<b>Форма выбора справочника</b>
Назначить форму основной	<input checked="" type="checkbox"/>
Основная форма списка и форма выбора	—
Имя	<b>ОсновнаяФормаВыбора</b>
Синоним	<b>Основная форма выбора</b>
Комментарий	—
Командная панель формы сверху	<input checked="" type="checkbox"/>
Командная панель формы снизу	—

Нажмите на кнопку «Далее» и на следующем шаге конструктора укажите размещение реквизитов так, как это изображено на рис. 7.10. Нажмите на кнопку «Готово». Отредактируйте форму по своему вкусу.

Поскольку мы для справочника указали в качестве способа редактирования режим «В диалоге», то теперь нам необходимо создать форму элемента.

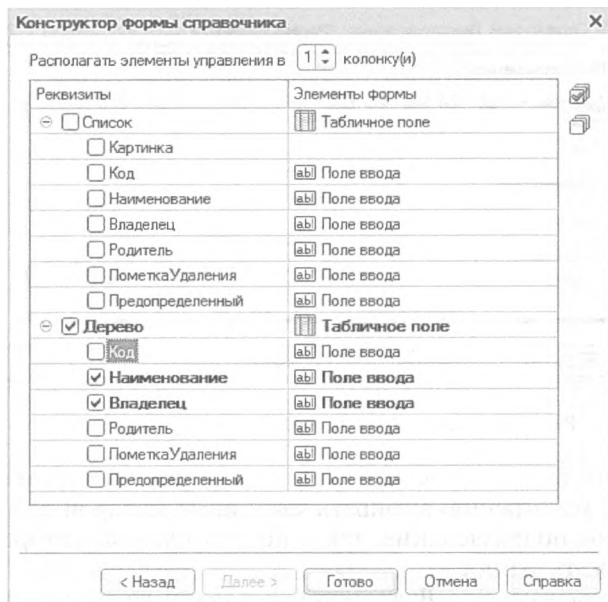


Рис. 7.10. Размещение реквизита «Наименование» только для просмотра в виде дерева

### Упражнение № 7.11

Для справочника «Подразделения» создайте основную форму элемента, указав на первом шаге конструктора следующие значения:

Свойство	Значение
Выберите тип формы	<b>Форма элемента справочника</b>
Назначить форму основной	<input checked="" type="checkbox"/>
Имя	<b>ОсновнаяФормаЭлемента</b>
Синоним	<b>Основная форма элемента</b>
Комментарий	—
Командная панель формы сверху	<input checked="" type="checkbox"/>
Командная панель формы снизу	<input checked="" type="checkbox"/>

Нажмите на кнопку «Готово» для завершения работы с Конструктором.

Отредактируйте форму по своему вкусу и сохраните конфигурацию.

Если Вы все сделали правильно, то ваша форма элемента будет похожа на ту, что изображена на рис. 7.11.

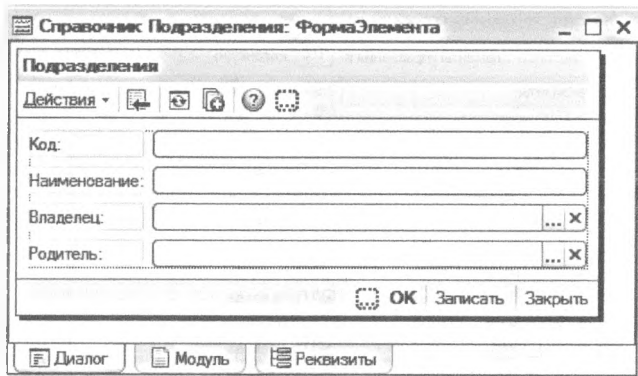


Рис. 7.11. Форма элемента справочника «Подразделения»

Обратим внимание на то, что в этой форме пользователь может по своему усмотрению выбирать как фирму, которой принадлежит конкретное подразделение, так и подразделение, которому будет принадлежать данное.

При этом реквизит «Владелец» будет давать возможность выбрать Firmу, а реквизит «Родитель» будет содержать ссылку на вышележащий уровень.

### 7.2.5. Закладка «Интерфейсы»

Выполните следующее упражнение.

#### Упражнение № 7.12

Вставьте в основной интерфейс команду: «Подразделения: Открыть справочник».

Откройте «ОсновнойИнтерфейс» и, если необходимо, отредактируйте свойство «Текст» у только что созданного пункта меню.

Сохраните и обновите конфигурацию.

На этом мы пока закончим настройку справочника «Подразделения».

## 7.3. САМОСТОЯТЕЛЬНАЯ РАБОТА: СПРАВОЧНИК «СОТРУДНИКИ»

Этот раздел предназначен для закрепления рассмотренного выше материала.

Продолжим чтение нашей постановки задачи. В ней сказано (п. 6), что «*сотрудники, работающие в объединении, могут числиться как на какой-то фирме в целом, так и в отдельном подразделении*».



Поэтому для хранения списка сотрудников самостоятельно создайте в конфигурации новый справочник, выполнив следующее упражнение.

### Упражнение № 7.13

Создайте в конфигурации новый справочник со следующим свой-ствами:

Свойство	Значение
Имя	<b>Сотрудники</b>
Синоним	<b>Сотрудники</b>
Комментарий	<b>Список сотрудников в объединении</b>
Иерархический справочник	<input checked="" type="checkbox"/>
Вид иерархии	<b>Иерархия групп и элементов</b>
Ограничение количества уровней иерархии	<input checked="" type="checkbox"/>
Количество уровней иерархии	<b>2</b>

При выборе владельцев справочника в диалоге «Выбор объекта» отметьте одновременно два справочника: «Фирмы» и «Подразделения» (рис. 7.12).

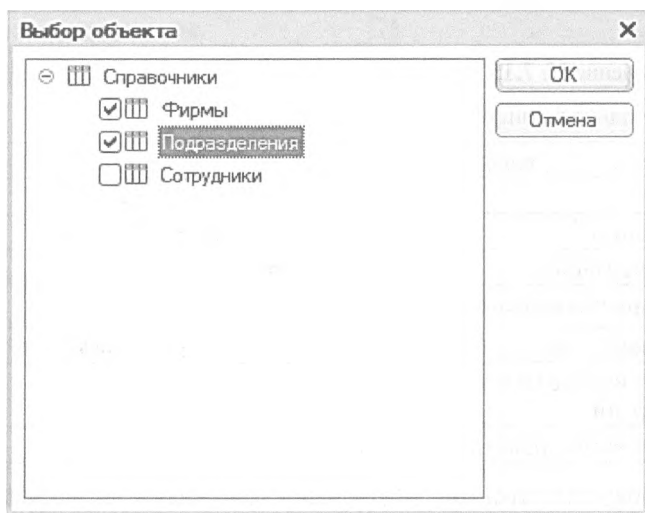


Рис. 7.12. Выбор нескольких владельцев для справочника

В этом случае одна запись в справочке «Сотрудники» может принадлежать либо справочнику «Фирмы», либо справочнику «Подразделения».

Такой способ настройки справочника приведет к тому, что для фирм, не имеющих организационной структуры, мы сможем указать список сотрудников, которые им принадлежат.

#### Упражнение № 7.14

В какое положение Вы установите на закладке «Нумерация» переключатель «Серии кодов»?

Для справочника «Сотрудники» создайте необходимое количество форм, используя закладку «Формы» его окна редактирования.

Какой режим редактирования Вы выберете?

Какие формы Вы создадите?

Необходимо ли что-то изменить на закладке «Интерфейсы»?

## 7.4. СПРАВОЧНИК «ОСНОВНЫЕ СРЕДСТВА»

В этом разделе мы начнем работу над новым справочником «Основные Средства», а в следующих юнитах полностью закончим его разработку.

Мы также изменим настройку ранее созданных справочников, добавив в них необходимое количество реквизитов.

#### Упражнение № 7.15

Создайте в конфигурации новый справочник:

Свойство	Значение
Имя	<b>Основные Средства</b>
Синоним	<b>Основные средства</b>
Комментарий	<b>Список ОС</b>
Иерархический справочник	<input checked="" type="checkbox"/>
Вид иерархии	<b>Иерархия групп и элементов</b>
Ограничение количества уровней иерархии	<input checked="" type="checkbox"/>
Количество уровней иерархии	<b>3</b>

Пропустите закладку «Владельцы» и перейдите на закладку «Данные». На этой закладке остановимся подробнее.

У справочника, как правило, имеются реквизиты (они же свойства соответствующего объекта) *Код* и *Наименование*.

Код может быть числовым или текстовым. В последнем случае он может содержать *префикс*. Например, предположим, что элементы нумеруются следующим образом: «АА-001», «АА-002», ..., «АА-999». Тогда «АА-» — это префикс.

Эта закладка позволяет настроить длину как кода, так и наименования. При выборе длины кода нужно руководствоваться следующим:

- ✓ сколько элементов должен вмещать справочник;
- ✓ каким образом мы собираемся кодировать элементы.

Максимальная длина кода в справочнике — 50 символов.

Давайте для определенности договоримся, что код будет содержать инвентарный номер (см. п. 7.1 в гл. 2) и первые три символа будут содержать префикс фирмы — 2 символа и знак «-» для разделения префикса и инвентарного номера.

Поэтому код в нашем справочнике будет иметь вид «ФФ-NNNN».

#### Упражнение № 7.16

Измените в окне редактирования длину кода и наименования справочника «ОсновныеСредства»:

Свойство	Значение
Длина кода	8
Длина наименования	75
Тип кода	Строка

#### 7.4.1. Реквизит «Фирма»

Поскольку по условию задачи ОС принадлежит какой-либо фирме, то мы должны создать в справочнике дополнительный реквизит, который бы хранил ссылки на элемент справочника «Фирмы».

#### Упражнение № 7.17

Используя кнопку «Добавить» окна редактирования справочника «ОсновныеСредства», добавьте в справочник новый реквизит:

Свойство	Значение
Группа свойств «Основные»	
Имя	Фирма
Синоним	Фирма
Комментарий	Фирма, которой принадлежит ОС
Индексировать	Индексировать

Свойство	Значение
Группа свойств «Тип»	
Тип	СправочникСсылка.Фирмы

Обратите внимание, что в качестве типа данных мы выбрали ссылку на элемент из справочника «Фирмы», что подчеркивает слово «Ссылка» в названии типа данных.

Отметим также, что установка свойства «Индексировать» в значение «Индексирование» приведет к тому, что пользователь сможет осуществлять сортировку в форме списка по значениям данного реквизита.

Префикс кода мы разместим в справочнике «Фирмы», выполнив упражнение 7.18.

#### Упражнение № 7.18

Добавьте для справочника «Фирмы» новый реквизит:

Свойство	Значение
Имя	Префикс
Синоним	Префикс
Комментарий	—
Индексировать	Индексировать
Тип	Строка
Длина	2
Допустимая длина	Фиксированная

Поскольку мы изменили структуру хранения справочника «Фирмы», то, следовательно, нам придется изменить и форму списка.

#### Упражнение № 7.18 (продолжение)

Откройте основную форму списка и выберите пункт меню Конфигуратора «Форма — Размещение данных...». В открывшемся диалоге «Размещение данных» установите пометку напротив реквизита «Префикс» табличного поля «Список» (рис. 7.13).

#### 7.4.2. Перечисление «ГруппыОС» и реквизит «ГруппаОС»

Для реализации п. 7.5 постановки задачи нам необходимо в конфигурации создать список возможных значений группы ОС, которой может принадлежать основное средства, т.е. перечисление.

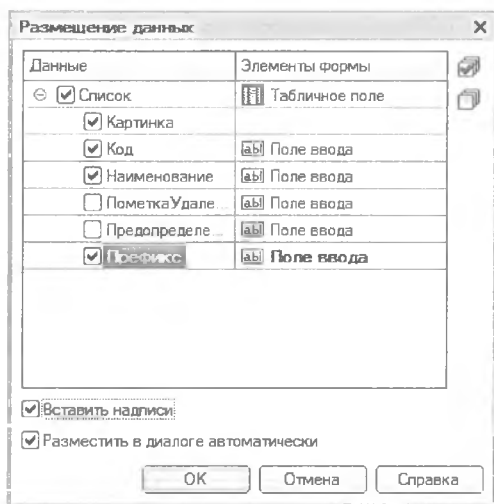


Рис. 7.13. Добавление новой колонки в табличное поле

Для того чтобы добавить в конфигурацию новое перечисление, необходимо выбрать одноименную ветвь дерева окна «Конфигурация» и, используя правую кнопку мыши, выбрать пункт «Добавить» (рис. 7.14). После чего конфигуратор откроет окно редактирования перечисления.

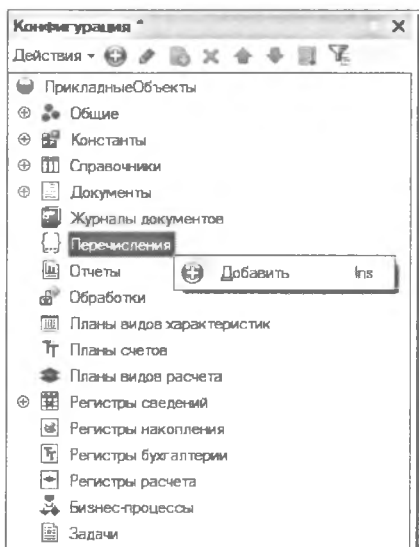


Рис. 7.14. Добавление нового перечисления

### Упражнение № 7.19

Создайте в конфигурации новое перечисление и заполните его свойствами следующими значениями.

Закладка «Основные»:

Свойство	Значение
Имя	Группы ОС
Синоним	Группы ОС
Комментарий	Группы основных средств

На закладке «Данные» необходимо ввести конкретные значения, заполнив их свойства в соответствии со следующей таблицей:

Имя	Синоним
Здания	Здания
Сооружения	Сооружения
Передаточные Устройства	Передаточные устройства
Машины И Оборудование	Машины и оборудование
Транспортные Средства	Транспортные средства
Инвентарь	Производств. и хозяйств. инвентарь
Прочие	Прочие основные фонды

Сохраните конфигурацию.

Теперь, после того как мы создали перечисление, мы можем создать реквизит в справочнике.

### Упражнение № 7.19 (продолжение)

Добавьте для справочника «Основные Средства» новый реквизит:

Свойство	Значение
Имя	Группа ОС
Синоним	Группа ОС
Комментарий	—
Индексировать	Индексирование
Тип	Перечисление Ссылка. Группы ОС

#### 7.4.3. Реквизит «Есть Детали»

В п. 7.6 постановки задачи сказано, что для тех ОС, которые состоят из отдельных деталей, и для тех из них, где такой учет необходим, нужно вести учет комплектующих, из которых они состоят.

Поэтому нам придется создать один реквизит, который мог бы хранить логический тип данных и одну табличную часть.

#### Упражнение № 7.20

Используя кнопку «Добавить» окна редактирования справочника «ОсновныеСредства», добавьте в справочник новый реквизит со следующими свойствами:

Свойство	Значение
Группа свойств «Основные»	
Имя	ЕстьДетали
Синоним	Есть детали
Комментарий	Признак того, что у ОС могут быть детали
Группа свойств «Тип»	
Тип	Булево

#### 7.4.4. Табличная часть «СписокДеталей»

Справочник в системе «1С:Предприятие» может иметь табличные части, поэтому добавим новую табличную часть, выполнив следующее упражнение.

#### Упражнение № 7.21

Используя кнопку «Добавить табличную часть» окна редактирования справочника «ОсновныеСредства», добавьте в справочник новую табличную часть:

Свойство	Значение
Имя	СписокДеталей
Синоним	Список деталей
Комментарий	—

Согласно постановке задачи нам нужно будет создать в этой табличной части 4 колонки: Деталь, Количество, Цена (УЕ) и Сумма (УЕ), что мы сделаем, выполнив следующее упражнение.

#### Упражнение № 7.22

Добавьте в табличную часть справочника «ОсновныеСредства» следующий реквизит:

Свойство	Значение
Имя	Деталь
Синоним	Деталь
Комментарий	—
Индексировать	Индексирование
Тип	СправочникСсылка

В качестве типа мы выбрали **СправочникСсылка**, поскольку пока у нас в конфигурации еще нет справочника «Детали» — он появится позже.

Добавьте реквизиты «Количество», «ЦенаУЕ» и «СуммаУЕ» с типом «Число» и длиной у всех 14. Для реквизита «Количество» точность должна быть равна 0, для остальных двух реквизитов должна быть равна 2.

#### 7.4.5. Реквизиты «ДатаВвода» и «ДатаСписания»

В соответствии с п. 7.7 постановки задачи нам необходимо еще создать два реквизита.

#### Упражнение № 7.23

Добавьте в справочник «ОсновныеСредства» реквизиты «ДатаВвода» и «ДатаСписания» со следующими свойствами:

Свойство	Значение
Индексировать	Индексировать
Тип	Дата
Состав даты	Дата

Свойства «Синоним» и «Комментарий» можно заполнить по своему усмотрению. Сохраните конфигурацию.

#### 7.4.6. Реквизит «Изображение»

В п. 7.8 постановки задачи сказано, что в каждой записи нужно хранить *изображение*. Для решения подобного рода задач в системе «1С:Предприятие» для реквизитов имеется специальный тип данных «ХранилищеЗначения».

#### Упражнение № 7.24

Добавьте в справочник «ОсновныеСредства» реквизит «Изображение».

Свойство	Значение
Имя	Изображение



Свойство	Значение
Синоним	Изображение
Комментарий	—
Использование	Для группы и элемента
Тип	ХранилищеЗначения

Обновите конфигурацию.

Свойство «Использование» может принимать три возможных значения:

- ✓ для элемента;
- ✓ для группы;
- ✓ для группы и элемента.

Это свойство позволяет определить состав реквизитов как для элемента, так и для группы. По умолчанию новые реквизиты создаются для использования только в элементах.

Для нашей постановки задачи логичным представляется использование такого реквизита как для группы, так и для элемента в справочнике «ОсновныеСредства», так как это даст возможность вносить изображение и для групп справочника.

#### 7.4.7. Закладка «Нумерация»

Очевидно, что для справочника «ОсновныеСредства» свойства «Автонумерация» и «Контроль уникальности» должны быть установлены.

#### Упражнение № 7.25

Проверьте, чтобы для справочника «ОсновныеСредства» были установлены следующие свойства:

Свойство	Значение
Автонумерация	<input checked="" type="checkbox"/>
Контроль уникальности	<input checked="" type="checkbox"/>
Серии кодов	Во всем справочнике

#### 7.4.8. Закладка «Формы»

Мы пропустим пока эту закладку, так как проектирование этого объекта еще не закончено.

## 7.5. САМОСТОЯТЕЛЬНАЯ РАБОТА: СПРАВОЧНИК «ДЕТАЛИ»

В п. 8 постановки задачи сказано: «Для каждой *Детали* необходимо, помимо ее наименования, хранить в ИБ ее цену в У.Е.». Самостоятельно создайте справочник «Детали».

### Упражнение № 7.26

Создайте самостоятельно в конфигурации новый справочник со следующими свойствами:

Свойство	Значение
Имя	Детали
Синоним	Детали
Комментарий	Список деталей
Иерархический справочник	<input checked="" type="checkbox"/>
Вид иерархии	Иерархия групп и элементов
Ограничение количества уровней иерархии	<input checked="" type="checkbox"/>
Количество уровней иерархии	4

Необходимо ли для справочника указывать владельцев?

Добавьте в справочник новый реквизит со следующими свойствами:

Свойство	Значение
Группа свойств «Основные»	
Имя	ЦенаУЕ
Синоним	Цена УЕ
Комментарий	Цена в УЕ
Группа свойств «Тип»	
Тип	Число
Длина	14
Точность	2
Неотрицательное	<input checked="" type="checkbox"/>

**Внимание!** Не забудьте изменить тип в реквизите «Деталь» справочника «ОсновныеСредства» на «СправочникСсылка.Детали».

## 7.6. РАБОТА СО СПРАВОЧНИКОМ ИЗ ВСТРОЕННОГО ЯЗЫКА СИСТЕМЫ

Для того чтобы обратиться к справочникам из встроенного языка системы, необходимо воспользоваться свойством *Справочники* глобального контекста. Это свойство имеет тип *СправочникиМенеджер*.

### 7.6.1. Менеджер справочника

Чтобы обратиться к конкретному справочнику, необходимо его имя записать «через точку», например: **Справочники.Фирмы**.

Такая запись имеет тип **СправочникМенеджер.<Имя справочника>** — это менеджер конкретного справочника, который имеет только методы:

- ✓ для организации выборки;
- ✓ для поиска отдельных записей;
- ✓ для получения макета;
- ✓ для получения формы справочника;
- ✓ для создания отдельных записей.

### 7.6.2. Организация выборки

Для того чтобы открыть выборку записей в справочнике, необходимо применить либо метод **.Выбрать(...)**, либо метод **.ВыбратьИерархически(...)**.

Приведем ряд примеров.

В предыдущем разделе мы с помощью конструктора печати написали в модуле формы списка справочника «Фирмы» процедуру **Печать()**, в которой цикл выборки выглядит следующим образом:

#### Процедура Печать(Элемент)

.....

**Выборка = Справочники.Фирмы.Выбрать();**

**ТабДок = Новый ТабличныйДокумент;**

**Пока Выборка.Следующий() = Истина Цикл**

**Объект = Выборка.ПолучитьОбъект();**

**Секция = Макет.ПолучитьОбласть(«Элемент»);**

**Секция.Параметры.Код = Объект.Код;**

**Секция.Параметры.Наименование = Объект.Наименование;**

**ТабДок.Вывести(Секция);**

**КонецЦикла;**

.....

Обычно метод **.Выбрать()** применяют для неиерархических справочников. Если справочник имеет два и более уровня, то в таких случаях лучше применять метод **.ВыбратьИерархически(...)**, поскольку он открывает выборку с учетом иерархии записей.

Для того чтобы привести пример иерархической выборки, сформируем с помощью Конструктора печати макет для формы списка справочника «Подразделения».

### Упражнение № 7.27

Для справочника «Подразделения» вызовите Конструктор печати и сформируйте макет печатной формы для печати списка записей. После чего «вручную» укажите вызов процедуры **Печать** из новой кнопки нижней командной панели инструментов.

Для надежности приводим макет печатной формы для печати списка записей (рис. 7.14а) и текст процедуры **Печать** (сравните со своим вариантом).

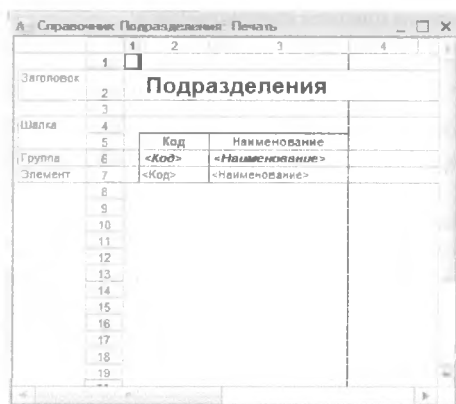


Рис. 7.14а. Макет печатной формы для печати списка записей справочника «Подразделения»



### ЛИСТИНГ Ю\_7.1

Процедура Печать(Элемент)

```
//{{ КОНСТРУКТОР_ПЕЧАТИ_СПИСОК(Печать)
```

```
// Данный фрагмент построен конструктором.
```

```
// При повторном использовании конструктора, внесенные вручную изменения  
будут утеряны!!!
```

```
ТабДок = Новый ТабличныйДокумент;
```

```
Макет = Справочники.Подразделения.ПолучитьМакет(«Печать»);
```

```
// Заголовок
```

```
Область = Макет.ПолучитьОбласть(«Заголовок»);
```

```
ТабДок.Вывести(Область);
```

```
// Шапка
```

```

Область = Макет.ПолучитьОбласть(«Шапка»);
ТабДок.Вывести(Область);
Выборка = Справочники.Подразделения.ВыбратьИерархически();
Группа = Макет.ПолучитьОбласть(«Группа»);
Элемент = Макет.ПолучитьОбласть(«Элемент»);
Пока Выборка.Следующий() Цикл
Если Выборка.ЭтоГруппа Тогда
Группа.Параметры.Заполнить(Выборка);
ТабДок.Вывести(Группа);
Иначе
Элемент.Параметры.Заполнить(Выборка);
ТабДок.Вывести(Элемент);
КонецЕсли;
КонецЦикла;
ТабДок.ОтображатьСетку = Ложь;
ТабДок.Защита = Ложь;
ТабДок.ТолькоПросмотр = Истина;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.Показать();
/}}_КОНСТРУКТОР_ПЕЧАТИ_СПИСОК
КонецПроцедуры

```

Откроем модуль основной формы списка справочника «Подразделения» и обратим внимание на цикл выборки из процедуры «Печать»:

```

Выборка =
Справочники.Подразделения.ВыбратьИерархически();
Группа = Макет.ПолучитьОбласть(«Группа»);
Элемент = Макет.ПолучитьОбласть(«Элемент»);
Пока Выборка.Следующий() Цикл
Если Выборка.ЭтоГруппа Тогда
Группа.Параметры.Заполнить(Выборка);
ТабДок.Вывести(Группа);
Иначе
Элемент.Параметры.Заполнить(Выборка);
ТабДок.Вывести(Элемент);
КонецЕсли;
КонецЦикла;

```

В этих примерах переменная «Выборка» имеет тип **СправочникВыборка**. Свойствами этого типа будут имена всех реквизитов справочника и его табличных частей. Кроме того, свойство **ЭтоГруппа** позво-

ляет определить, является ли текущая выбранная запись группой или это элемент.

Методов у этого типа данных всего три:

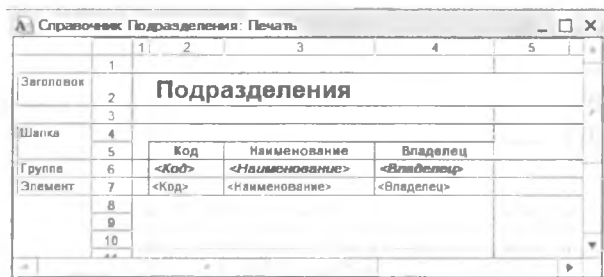
- ✓ Следующий();
- ✓ УровеньВВыборке();
- ✓ ПолучитьОбъект();

Метод **Следующий()** извлекает очередную запись из выборки, к полям которой можно обратиться через свойства объекта.

Метод **УровеньВВыборке()** имеет смысл применять, если выборка была открыта методом **ВыбратьИерархически(...)**. Номер уровня отсчитывается от 0.

### Упражнение № 7.28

Внесите изменения в алгоритм работы процедуры **Печать()** из модуля формы списка «Подразделения» (если необходимо) и его макет таким образом, чтобы результирующий табличный документ отображал колонку «Владелец» (рис. 7.146).



Шапка	Код	Наименование	Владелец
Группа	<Код>	<Наименование>	<Владелец>
Элемент	<Код>	<Наименование>	<Владелец>

Рис. 7.146. Измененный макет печатной формы для печати списка записей справочника «Подразделения»

### 7.6.3. Получение ссылки на элемент справочника

У менеджера справочника есть три метода, которые позволяют найти необходимую запись в справочнике и получить на нее ссылку:

- ✓ НайтиПоКоду(...);
- ✓ НайтиПоНаименованию(...);
- ✓ НайтиПоРеквизиту(...).

Все эти методы возвращают в качестве значения тип данных **СправочникСсылка.<Имя справочника>**.

При организации выборки Вы также можете получить ссылку на выбранную запись как на объект, если обратитесь к свойству **Ссылка** типа данных **СправочникВыборка.<Имя справочника>**.

#### 7.6.4. Работа с отдельными записями как с объектами

У объектов типа **СправочникСсылка.<Имя справочника>** и **СправочникВыборка.<Имя справочника>** есть метод **ПолучитьОбъект()**, который позволяет получить доступ к объекту типа **СправочникОбъект.<Имя справочника>**.

Объект этого типа обладает достаточным набором свойств и методов, которые позволяют не только прочесть все поля записи из ИБ, но изменить эту запись, сохранив сделанные в ОП изменения.

Свойствами этого объекта являются все реквизиты справочника, поэтому обращаться к ним можно «через точку» по имени (с полным набором методов можно ознакомиться в документации).

У объекта типа **СправочникМенеджер.<Имя справочника>** есть два метода, которые позволяют создавать новые записи в справочнике:

- ✓ **СоздатьГруппу();**
- ✓ **СоздатьЭлемент();**

Кроме того, новая запись может быть получена путем применения метода **.Скопировать()** у типов данных **СправочникОбъект** и **СправочникСсылка**.

При этом процесс создания новой записи программным путем может выглядеть следующим образом:

```
НовыйЭлемент=Справочники.Имя.СоздатьЭлемент();
НовыйЭлемент.Код = ...;
НовыйЭлемент.Наименование = ...;
НовыйЭлемент.... = ...;
.....
НовыйЭлемент.Записать();
```

Таким образом, мы приходим к выводу, что новый объект в системе может быть создан не только вследствие интерактивных действий пользователя, но и в результате выполнения программных действий.

Тот же вывод верен и для изменения (модификации) существующих записей.

Для того чтобы процесс создания новых и изменения существующих записей (как интерактивным, так и программным путем) выполнялся бы единообразным образом (работали бы одни и те же алгоритмы), в системе у **Справочников. Документов. Отчетов и Обработок** существует **Модуль Объекта**. Этот модуль предназначен для обработки системных событий, связанных с конкретной записью, ее созданием, заполнением, записью в ИБ и удалением.

### 7.6.5. Закладка «Прочие» и Модуль объекта

Если открыть окно редактирования справочника и выбрать закладку «Прочие» (рис. 7.15), то кнопка «Модуль объекта» позволит открыть одноименный модуль.

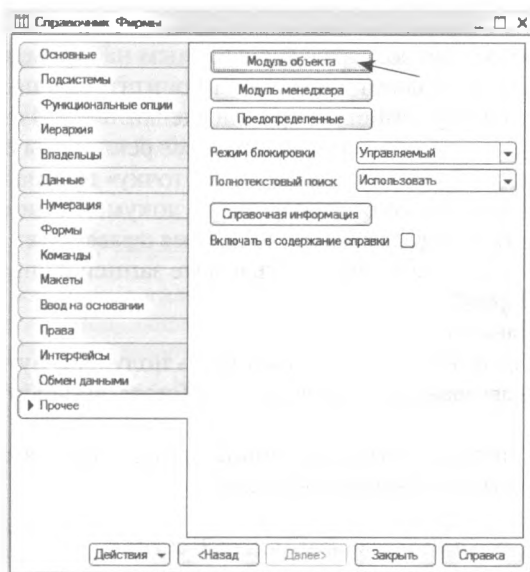


Рис. 7.15. Закладка «Прочие»

Для справочников этот модуль позволяет обрабатывать события, которые, напомним, могут возникать как интерактивным, так и программным способом.

Для их обработки в Модуле объекта можно использовать следующие обработчики событий (**имена фиксированы**):

- ПриУстановкеНовогоКода(<Стандартная обработка>, <Префикс>)
- ПриКопировании(<Объект копирования>)
- ОбработкаЗаполнения(<Основание>)
- ПередЗаписью(<Отказ>)
- ПриЗаписи(<Отказ>)
- ПередУдалением(<Отказ>)

Рассмотрим, как это можно сделать на следующем примере.

Предположим, что при копировании записи в справочнике «Фирмы» мы хотим, чтобы к наименованию добавлялся символ «!». Тогда текст модуля объекта этого справочника будет следующим:



// Модуль объекта справочника «Фирмы»  
Процедура ПриКопировании(ОбъектКопирования)  
Наименование = Наименование+«1»;  
КонецПроцедуры

#### Упражнение № 7.29

Откройте модуль справочника «Фирмы» и внесите в него текст процедуры **ПриКопировании(...)**.

## 7.7. ПРЕДОПРЕДЕЛЕННЫЕ ЗАПИСИ В СПРАВОЧНИКЕ

Закладка «Прочее» имеет кнопку «Предопределенные», которая открывает окно «Предопределенные элементы справочника», изображенное на рис. 7.16.

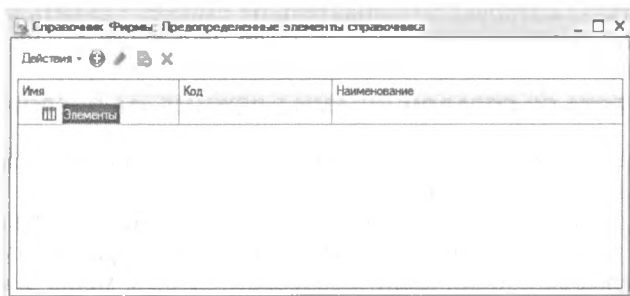


Рис. 7.16. Окно для редактирования списка предопределенных записей справочника

Поскольку по условиям нашей задачи справочник должен содержать группу «Компьютеры», то мы добавим ее, выполнив следующее упражнение.

#### Упражнение № 7.30

В окне «Предопределенные элементы справочника» для справочника «**ОсновныеСредства**» выберите пункт меню «Действия — Новая группа Ctrl+F9» и заполните окно «Элемент справочника».

Свойство	Значение
Родитель	Элементы
Имя	Компьютеры
Код	0000000
Наименование	Компьютеры и оргтехника

Нажмите на кнопку ОК, чтобы сохранить новую группу в справочнике.

Установите для версии конфигурации значение «Юнит 7» и обновите ее.

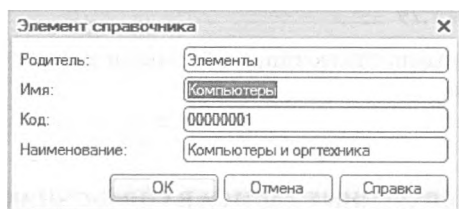


Рис. 7.17. Предопределенная запись

После выполнения этого упражнения справочник будет содержать группу с кодом «00000001» и наименованием «Компьютеры и оргтехника», которую пользователь не сможет удалить в режиме «1С:Предприятие».

## 7.8. РАБОТА СО СПРАВОЧНИКАМИ В РЕЖИМЕ «1С:ПРЕДПРИЯТИЕ»

В этом разделе мы рассмотрим, как можно работать с формами справочников в режиме «1С:Предприятие».

Прежде всего, отметим, что в результате установки флажков на закладке «Интерфейсы» в окне редактирования справочника основной интерфейс конфигурации содержит пункты меню для вызова формы списка справочников «Фирмы», «Подразделения», «Сотрудники» (рис. 7.18).

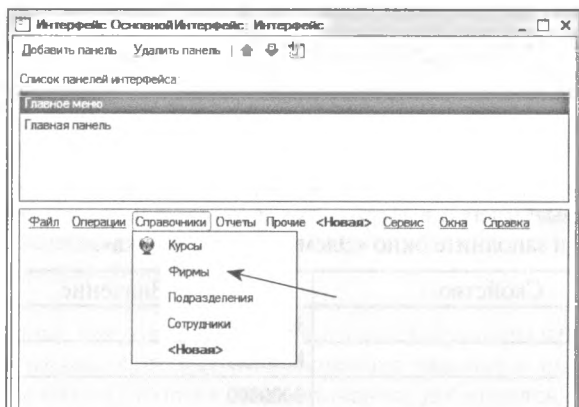


Рис. 7.18. Основной интерфейс конфигурации после создания справочников

### 7.8.1. Ввод элементов в справочник «Фирмы»

Сначала мы приступим к заполнению справочника «Фирмы».

#### Упражнение № 7.31

Запустите режим «1С:Предприятие».

В основном меню программы выберите пункт «Справочники — Фирмы» и заполните справочник следующими значениями:

Код	Наименование	Префикс
1	АО «Альтернатива»	АЛ
2	ЗАО «Бухгалтер»	БХ

Проверьте работу кнопки «Печать».

Если Вы все сделали правильно, то форма списка справочника «Фирмы» в режиме «1С:Предприятие» будет выглядеть так, как это изображено на рис. 7.19.

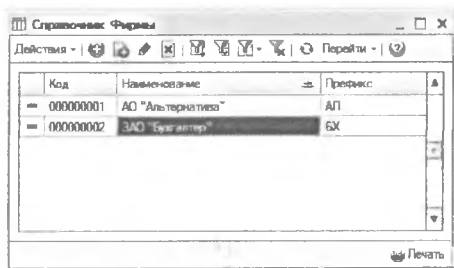


Рис. 7.19. Форма списка справочника «Фирмы» в режиме «1С:Предприятие»

### 7.8.2. Ввод элементов в справочник «Подразделения»

Приступим к заполнению справочника «Подразделения».

#### Упражнение № 7.32

Откройте форму списка справочника «Подразделения», используя пункт меню «Справочники».

Нажмите на клавишу Ins и в открывшейся форме элемента заполните его реквизит следующими значениями:

Реквизит	Значение
Код	1
Наименование	Администрация
Владелец	АО «Альтернатива»
Родитель	—

Нажмите на кнопку ОК, чтобы сохранить элемент в ИБ.  
Введите в справочник следующий элемент:

Реквизит	Значение
Код	1
Наименование	Администрация
Владелец	ЗАО «Бухгалтер»
Родитель	—

Нажмите на кнопку ОК, чтобы сохранить элемент в ИБ.

---

---

Для того чтобы вести новый элемент на втором уровне, необходимо в форме сначала включить *Просмотр по владельцу*, который по умолчанию отключен. Именно поэтому форма отображала записи, подчиненные всем фирмам.

#### Упражнение № 7.32 (продолжение)

---

---

В справочнике «Подразделения» установите курсор в строку с ЗАО «Бухгалтер».

Включите режим просмотра по владельцам, используя команду «Действия — Просмотр по владельцу».

Введите в справочник следующий элемент:

Реквизит	Значение
Код	2
Наименование	Дирекция
Владелец	ЗАО «Бухгалтер»
Родитель	Администрация

При выборе Родителя нажмите кнопку F4 и выберите соответствующий элемент (рис. 7.20).

Опять нажмите на клавишу Ins и введите новый элемент:

Реквизит	Значение
Код	3
Наименование	Секретариат
Владелец	ЗАО «Бухгалтер»
Родитель	Администрация

Нажмите на кнопку ОК, чтобы сохранить элемент в ИБ.

Используя клавиши «Ctrl+Стрелка вверх», вернитесь на первый уровень и введите новый элемент со следующими свойствами:

Реквизит	Значение
Код	4
Наименование	Бухгалтерия
Владелец	ЗАО «Бухгалтер»
Родитель	—

Проверьте работу кнопки «Печать».

Рис. 7.20. Использование формы выбора справочника

### 7.8.3. Ввод записей в справочник «Сотрудники»

Записи в этом справочнике могут принадлежать как справочнику «Фирмы», так и справочнику «Подразделения», что определяется содержимым реквизита «Владелец». Этот реквизит имеет так называемый *составной тип данных*.

Для данного справочника это означает, что при заполнении этого реквизита в форме сначала придется выбрать необходимый тип данных, а уже затем его конкретное значение: ссылка на элемент справочника «Фирмы» или ссылка на элемент справочника «Подразделения».

#### Упражнение № 7.33

В форме списка справочника «Фирмы» установите курсор на записи «АО «Альтернатива»» и выберите пункт меню «Перейти — Сотрудники».

Введите новый элемент со следующими свойствами:

Реквизит	Значение
Код	1
Наименование	Никитина Алла Андреевна
Владелец	АО «Альтернатива»

Обратите внимание, что реквизит «Владелец» уже заполнен. Закройте все окна.

Продолжим ввод данных в справочник «Сотрудники». Нажмите на кнопку Ins и введите следующий элемент в справочник:

Реквизит	Значение
Код	2
Наименование	Баранов Борис Борисович

Для заполнения поля «Владелец» нажмите в нем на кнопку F4, после чего выберите в диалоге «Выбор типа данных» (рис. 7.21) справочник «Подразделения».

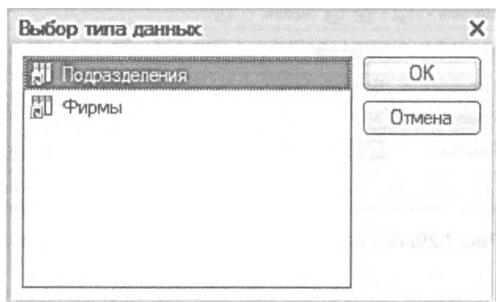


Рис. 7.21. Диалог «Выбор типа данных»

По окончании выбора типа еще раз нажмите на кнопку F4, тогда система откроет форму со списком подразделений (рис. 7.22).

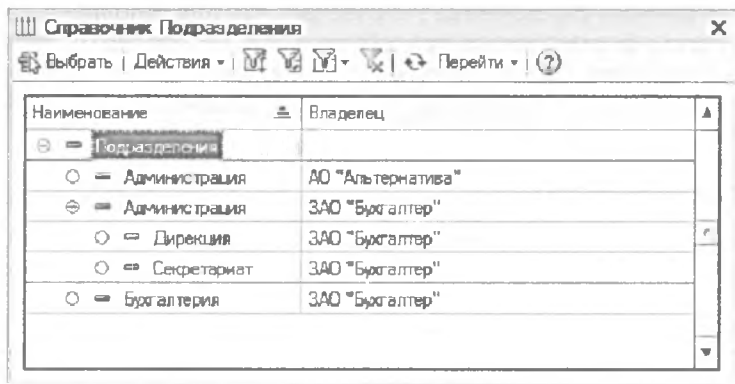


Рис. 7.22. Форма выбора справочника «Подразделения»

Нажмите на кнопку «+» дополнительной клавиатуры и выберите подразделение «Дирекция».

## 7.9. ИСПОЛЬЗОВАНИЕ ЯЗЫКА ЗАПРОСОВ


В этом разделе мы рассмотрим использование языка запросов для формирования печатных форм справочников «Фирмы» и «Подразделения», для обращения к данным указанных справочников.

### 7.9.1. Конструктор запросов

Рассмотрим использование конструктора запросов на примере печати списка записей справочника «Фирмы».

Начнем с формирования подготовительных действий, выполнив упражнение № 7.34.

#### Упражнение № 7.34

В форме списка справочника «Фирмы» выделите мышью кнопку  на нижней панели инструментов и заполните ее свойства следующими значениями:

Свойство	Значение
Имя	ПечатьЧерезЗапрос
Тип	Действие
Действие	ПечатьЧерезЗапрос
Текст	Печать через запрос
Подсказка	Печать через запрос
Пояснение	Печать через запрос
Отображение	Надпись и картинка

Наберите в модуле следующий текст процедуры:

```
// Печать списка справочника, используя запрос  
Процедура ПечатьЧерезЗапрос(Кнопка)  
Запрос = Новый Запрос();  
Запрос.Текст =  
КонецПроцедуры
```

Для того чтобы сформировать текст запроса, необходимо установить курсор в то место процедуры, где по смыслу должен располагаться его текст и, используя правую кнопку мыши, вызвать конструктор.

#### Упражнение № 7.35

Установите курсор в конце строки `Запрос.Текст=` и нажмите на правую кнопку мыши (рис. 7.23).

Сразу после этих действий Конструктор запросит о необходимости создания нового запроса.

В ответ на вопрос «Создать новый запрос?» нажмите на кнопку «Да», для того чтобы Конструктор открыл свое окно (рис. 7.24).

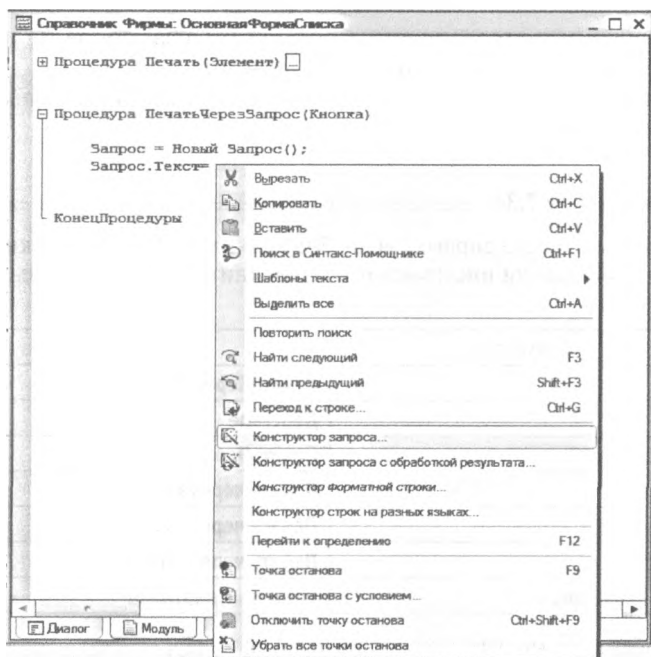


Рис. 7.23. Вызов конструктора запросов



Рис. 7.24. Окно конструктора запросов



Это окно содержит закладки, позволяющие написать текст запроса визуальным образом.

Как правило, текст запроса начинают формировать с закладки «Таблицы и поля».

На первом этапе Конструктор предложит указать таблицы и поля, которые должен обработать объект **Запрос**.

### Упражнение № 7.35 (продолжение)

Используя кнопки «>» в окне диалога Конструктора выберите в качестве полей «Код» и «Наименование» из справочника «Фирмы» (рис. 7.25). Нажмите на кнопку «Вперед>».

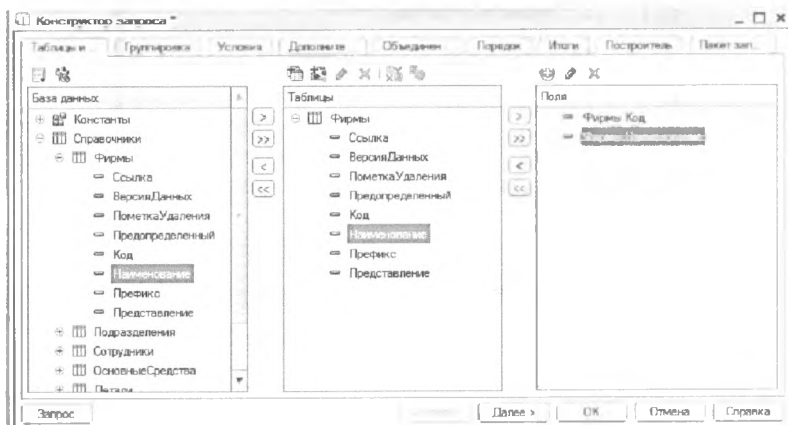


Рис. 7.25. Выбор таблиц и полей запросов

Поскольку наш справочник одноуровневый, то мы сразу можем перейти к закладке «Порядок».

Используя кнопки «>» в окне диалога Конструктора, выберите в качестве порядка сортировки поле «Код» (рис. 7.26).

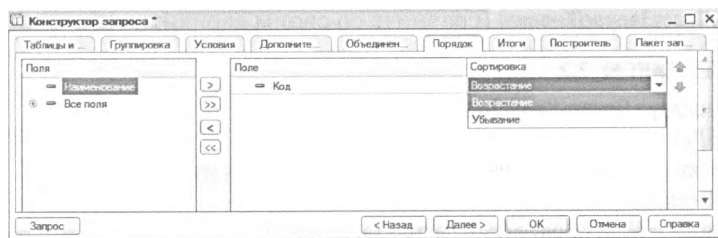
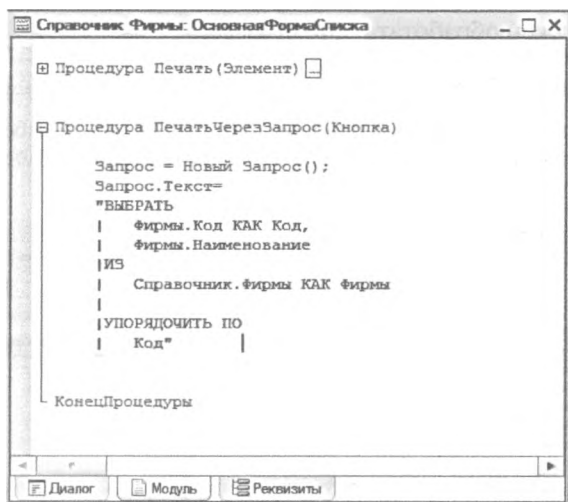


Рис. 7.26. Выбор порядка расположения записей в запросе

Нажмите на кнопку ОК.

Закладку «Итоги» также можно пропустить, так как этот справочник не содержит числовых реквизитов.

После завершения упражнения в текст нашей процедуры добавились строки:



### Упражнение № 7.36

Используя полученные ранее знания, доработайте текст процедуры **ПечатьЧерезЗапрос(Кнопка)** так, чтобы она формировала результирующий табличный документ, как и тот, что получается в результате работы процедуры **Печать(Элемент)**.

Проверьте работу обеих кнопок в режиме «1С:Предприятие».

Для надежности приводим доработанный текст процедуры **ПечатьЧерезЗапрос(Кнопка)** (сравните со своим вариантом).

### ЛИСТИНГ Ю\_7.2

```
Процедура ПечатьЧерезЗапрос(Кнопка)
ТабДок = Новый ТабличныйДокумент;
Макет = Справочники.Фирмы.ПолучитьМакет(«Печать»);
// Заголовок
Секция = Макет.ПолучитьОбласть(«Заголовок»);
ТабДок.Вывести(Секция);
// Шапка
Секция = Макет.ПолучитьОбласть(«Шапка»);
ТабДок.Вывести(Секция);
```

```

Запрос = Новый Запрос();
Запрос.Текст = «ВЫБРАТЬ
| Фирмы.Код КАК Код,
| Фирмы.Наименование
| ИЗ
| Справочник.Фирмы КАК Фирмы
|
| УПОРЯДОЧИТЬ ПО
| Код»;
//И выполняем запрос
Результат = Запрос.Выполнить();
Секция = Макет.ПолучитьОбласть(«Элемент»);
// Обработка результата
Выборка=Результат.Выбрать();
Пока Выборка.Следующий() Цикл
Секция.Параметры.Код = Выборка.Код;
Секция.Параметры.Наименование = Выборка.Наименование;
ТабДок.Вывести(Секция);
КонецЦикла;
ТабДок.ОтображатьСетку = Ложь;
ТабДок.Защита = Ложь;
ТабДок.ТолькоПросмотр = Истина;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.Показать();
КонецПроцедуры

```

## 7.9.2. Конструктор выходных форм

Выходную форму вместе с запросом можно создавать, используя для ускорения процесса конфигурирования Конструктор выходных форм.

Вызвать этот Конструктор можно разными способами. Наиболее простой — из окна «Конфигурация» (рис. 7.27).

Сразу после описанных выше действий Конструктор откроет диалог по выбору места расположения выходной формы (рис. 7.28).

### Упражнение № 7.37

---

Запустите Конструктор выходных форм для справочника «Подразделения». Укажите на первом этапе для новой выходной формы название — «Подразделения».

Используя кнопки «>» в окне диалога Конструктора, выберите в качестве полей: Код, Наименование и Владелец из справочника «Подразделения» (рис. 7.29).

Переключитесь на закладку «Порядок».

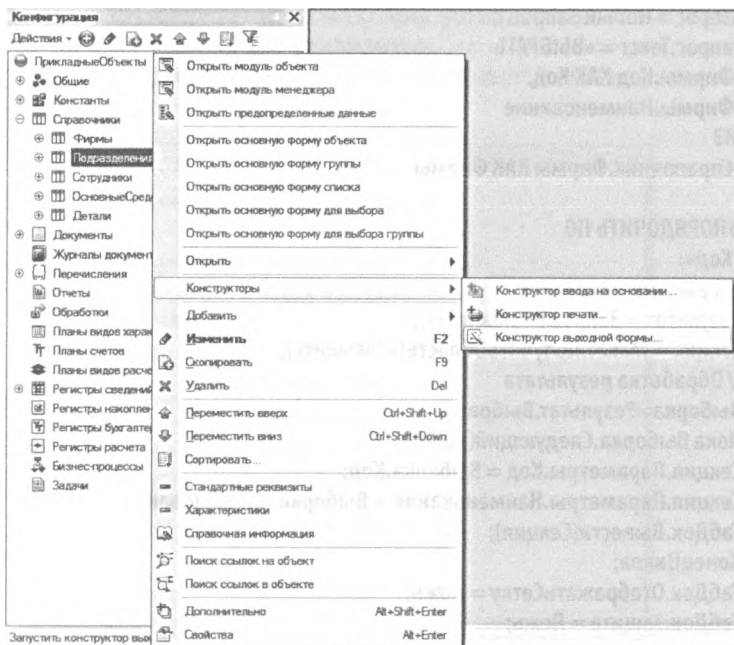


Рис. 7.27. Вызов Конструктора выходных форм

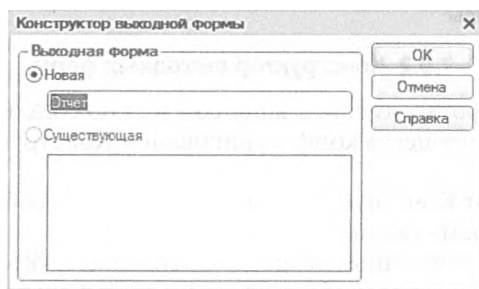


Рис. 7.28. Выбор места расположения выходной формы

Очевидно, что записи в запросе следует сначала упорядочить по названию фирм, а потом уже «внутри» каждой фирмы следует сохранить иерархию записей справочника «Подразделения».

На закладке укажите следующий порядок сортировки (рис. 7.30).

Поле	Сортировка
ВладелецПредставление	<b>Возрастание</b>
Наименование	<b>Иерархия</b>

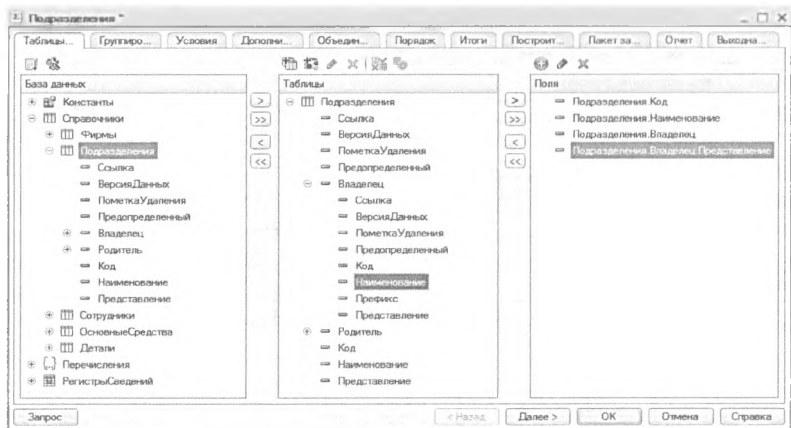


Рис. 7.29. Выбор таблиц и полей запросов

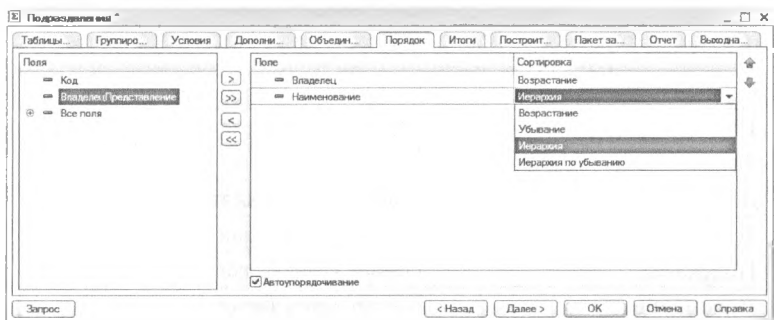


Рис. 7.30. Выбор порядка сортировки

Перейдите к закладке «Выходная форма».


На закладке «Выходная форма» мы сможем указать конструктору, где расположить процедуру и нужно ли связать ее соответствующей кнопкой.

Установите переключатель «Расположение элементов управления» в положение «Новая форма» (рис. 7.31).

И нажмите на кнопку ОК.

Итак, выходная форма создана. Теперь необходимо добавить ее вызов из формы списка справочника «Подразделения».

### Упражнение № 7.38

Выделите мышью кнопку  на нижней панели инструментов и заполните ее свойства следующими значениями:

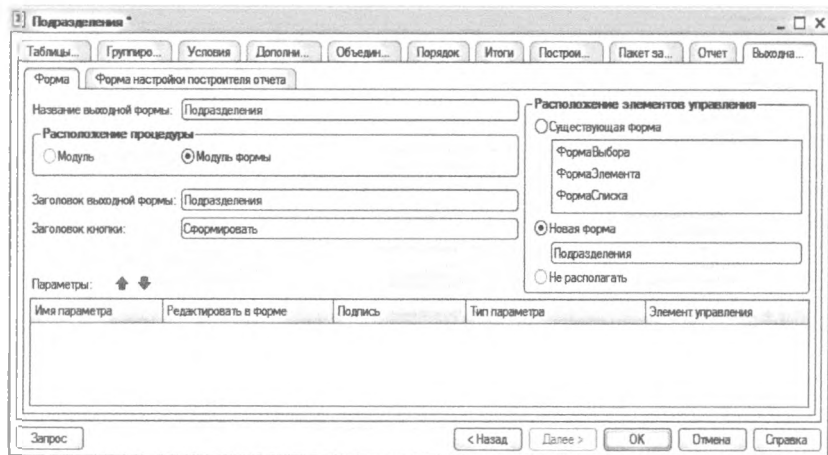


Рис. 7.31. Настройка выходной формы

Свойство	Значение
Имя	ПечатьЧерезЗапрос
Тип	Действие
Действие	СформироватьНажатие
Текст	Печать через запрос
Подсказка	Печать через запрос
Пояснение	Печать через запрос

Перейдите на закладку «Модуль» формы списка. В процедуру **СформироватьНажатие(Кнопка)** добавим следующий текст:

**Процедура СформироватьНажатие(Кнопка)**  
**Справочники.Подразделения.ПолучитьФорму(«Подразделения»),).Открыть();**  
**КонецПроцедуры**

Запустите режим «1С:Предприятие» и проверьте работу новой кнопки.

## 7.10. ЧТО МЫ УЗНАЛИ

В этой главе мы познакомились с прикладным объектом «Справочник», который предназначен для хранения в ИБ нормативно-справочной информации.

Новые справочники мы создавали в соответствии с постановкой задачи (Юнит 2), используя для этой цели окно редактирования справочника. В этом окне мы подробно рассмотрели состав и назначение его закладок: «Основные», «Иерархия», «Владельцы», «Данные», «Нумерация», «Формы», «Макеты» и «Интерфейсы».

Справочник может быть многоуровневым, как с ограничением числа уровней, так и без ограничения этого числа. Когда в справочнике два и более уровней, то реквизит «Родитель» содержит ссылку на запись из вышележащего уровня. Таким образом система реализует механизм подчиненности в пределах одного справочника.

Для указания отношения подчиненности между записями различных справочников служит закладка «Владельцы», которая позволяет указать, каким справочникам могут подчиняться записи данного справочника. Отметим, что в общем случае разные записи из одного справочника могут быть подчинены записям из разных справочников, но одна запись всегда подчинена одной записи из другого справочника, на которую содержит ссылку реквизит «Владелец».

Так же как и регистр сведений, справочник может иметь дополнительные реквизиты, назначение которых описывать какие-либо свойства объекта учета. Справочник в системе «1С:Предприятие 8» может иметь несколько табличных частей.

В этой главе мы также рассмотрели работу со справочниками из встроенного языка системы. При этом мы подробно рассмотрели методику организации выборки записей из справочника.

Особое внимание мы уделили работе с типом данных «СправочникОбъект», который позволяет работать с отдельной записью как с объектом. Мы рассмотрели, какие события могут возникать при работе с этим объектом. Напомним, что изменять запись из справочника можно как вручную (интерактивным образом через форму), так и программным путем. Именно поэтому Модуль объекта способен обрабатывать события, связанные с изменением конкретной записи.

В этой главе мы также рассмотрели использование запросов (языка запросов) для получения данных из справочника. Для этой цели мы воспользовались как Конструктором запросов, так и Конструктором выходных форм.

В этой главе мы рассмотрим некоторые приемы разработки форм на примере решения проблем, которые мы обнаружили в предыдущей главе.

### 8.1. ИЗ ЧЕГО СОСТОИТ ФОРМА

Для того чтобы понять, из чего состоит форма, откроем в Конфигураторе основную форму списка справочника «Подразделения» (рис. 8.1).

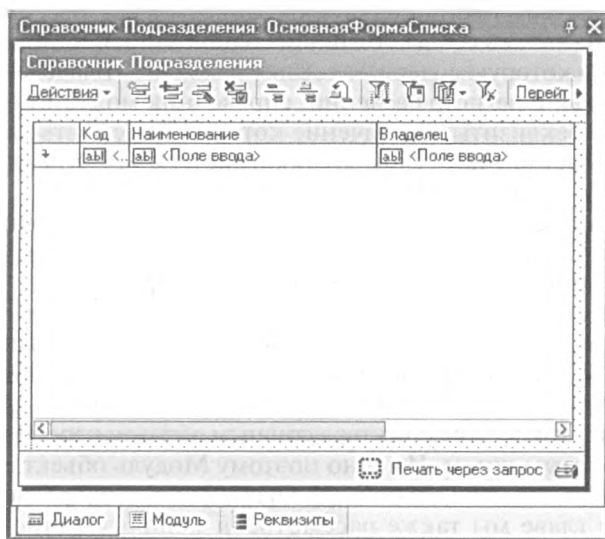


Рис. 8.1. Окно редактора форм Конфигуратора

Мы видим, что это окно имеет внизу три закладки:

- ✓ Диалог;
- ✓ Модуль;
- ✓ Реквизиты.

Диалог состоит из отдельных элементов диалога, которые размещены в форме.

Каждый из элементов и сама форма в целом обладают необходимым набором свойств, которые позволяют задать поведение формы в режиме «1С:Предприятие».



В большинстве случаев настройка свойств позволяет обойтись без программирования — можно создавать достаточно сложные формы, в которых модуль будет пустым.

Модуль в форме предназначен в основном для размещения в форме процедур — обработчиков событий.

При необходимости в этом модуле могут быть размещены вспомогательные процедуры и функции, которые вызываются из процедур — обработчиков событий. Если разработчик конфигурации предполагает осуществлять открытие формы программным путем (метод **ПолучитьФорму(...)**), то в модуле формы можно разместить дополнительные переменные, процедуры и функции с дополнительным ключевым словом **Экспорт**, обращение к которым возможно через объект типа «Форма».

## 8.2. РЕКВИЗИТЫ ФОРМЫ

Закладка «Реквизиты» (рис. 8.2) определяет, какие данные должна отображать форма и как она должна себя вести в целом.

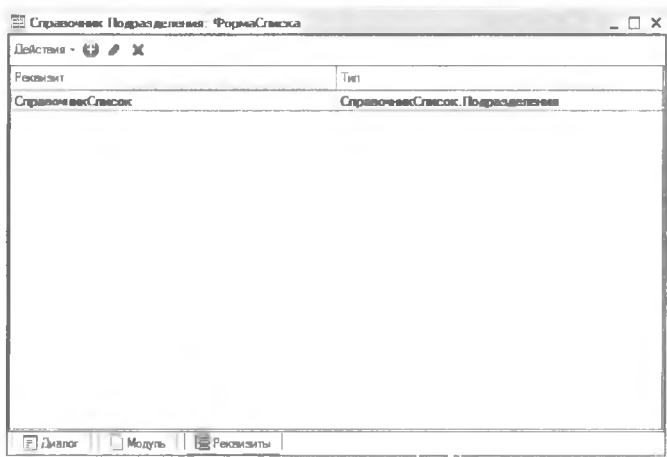


Рис. 8.2. Реквизиты формы

Реквизитов может быть много. Из этого следует, что в одной форме можно одновременно отображать несколько разных объектов (необязательно как-то связанных друг с другом).

Если при создании формы Конструктором Вы выбрали тип формы, отличный от положения «Произвольная форма», то в реквизитах всегда будет **основной реквизит**, который система всегда отображает жирным шрифтом.

Именно основной реквизит определяет поведение формы в целом.

### 8.2.1. Два объекта (реквизита) в одной форме и связь между ними

В предыдущей главе при работе со справочником «Подразделения» в его форме списка мы вынуждены были либо включать отбор по владельцу, либо отключать его, чтобы работать либо со всем списком подразделений, либо со списком конкретной фирмы. Но при этом мы не могли в этом же окне видеть список всех фирм.

Представляется разумным разместить в форме два *табличных поля*. Слева расположить элемент, отображающий список *фирм*, а справа — список *подразделений*.

Прежде чем вставить в форму элемент *табличное поле*, необходимо на закладке «Реквизиты» добавить новый реквизит.

#### Упражнение № 8.1

Откройте форму «ОсновнаяФормаСписка» справочника «Подразделения». Откройте закладку «Реквизиты» и нажмите на кнопку Ins на клавиатуре, после чего заполните свойства нового реквизита следующими значениями:

Свойство	Значение
Имя	СписокФирм
Тип	СправочникСписок.Фирмы

#### Упражнение № 8.1 (продолжение)

Переключитесь на закладку «Диалог» и, увеличив предварительно размер формы, освободите слева место для вставки нового элемента формы (рис. 8.3).

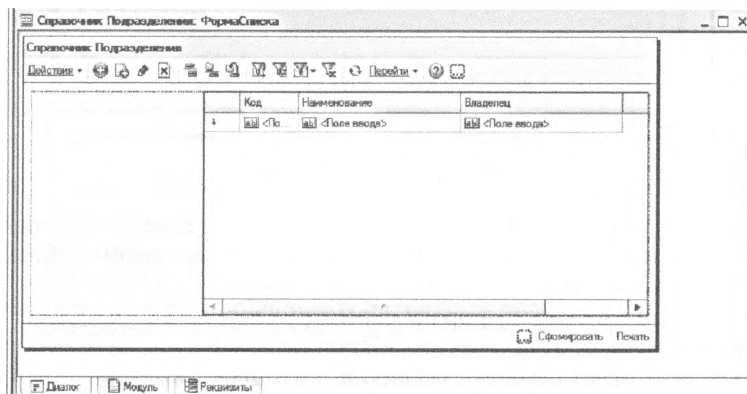


Рис. 8.3. Форма списка справочника «Подразделения» после изменений

Вставьте в форму элемент управления «Табличное поле», используя команду «Форма — Вставить элемент управления».

В открывшемся диалоге выберите элемент управления «Табличное поле» и заполните его свойства следующими значениями:

Свойство	Значение
Имя	СписокФирм
Тип значения	ТаблицаЗначений

Нажмите на кнопку ОК и расположите элемент в свободной области формы (рис. 8.4).

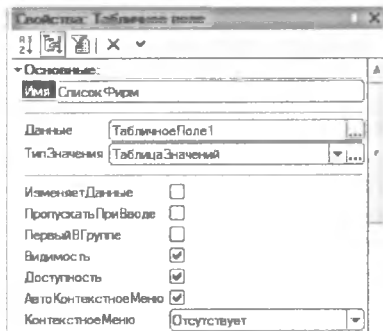
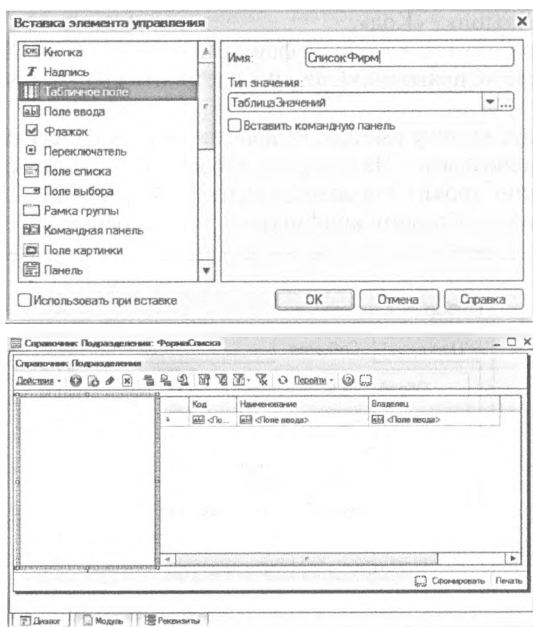


Рис. 8.4. Форма списка после вставки второго табличного поля

Сейчас это поле никак не связано с реквизитами формы. Чтобы установить такую связь, мы изменим свойства элементов формы «СписокФирм» и «СправочникСписок».

### Упражнение № 8.2

Установите для элемента формы «СписокФирм» свойство «Данные» в значение «СписокФирм».

Что произошло с табличным полем?

Установите для элемента «СписокФирм» свойство «Только просмотр».

Удалите колонку «Код».

Для того чтобы между двумя формами была связь, необходимо в табличном поле «СправочникСписок» настроить свойство «Связь по владельцу».

Используя кнопку выбора, в свойстве «Связь по владельцу» отойдите диалог «Выбор связи». На закладке «События» выберите событие «ПриАктивизацииСтроки» для объекта «СписокФирм» (рис. 8.5).

Сохраните и обновите конфигурацию и проверьте работу формы.

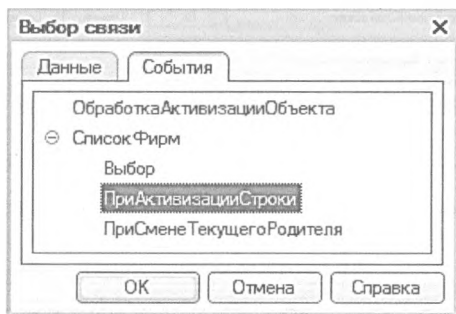


Рис. 8.5. Диалог «Выбор связи»

### 8.3. ИСПОЛЬЗОВАНИЕ ОБЪЕКТА ТИПА «ДЕРЕВОЗНАЧЕНИЙ»

В предыдущем разделе мы рассмотрели создание и настройку формы без использования программирования. Фактически поведение формы и ее элементов мы определяли путем заполнения их свойств конкретными значениями. При этом если модуль формы остается пустым, — это идеальная ситуация.

В формах достаточно часто встречаются элементы формы типа «Табличное поле». Указав для этого поля свойство «Данные», мы устанавливаем связь между списком записей какого-либо прикладного объекта и элементом формы. При этом заполнение этого эле-

мента конкретными строками берет на себя сама форма — программировать ничего не нужно.

Однако так бывает не всегда. Иногда заполнять табличное поле приходится программным путем, так как сама форма не в состоянии отобразить в одном поле записи из разных объектов конфигурации.

В этом разделе мы рассмотрим указанную проблему на примере формы списка справочника «Сотрудники».

В форме списка справочника «Сотрудники» слева расположено *дерево*, которое отображает группы справочника. Нашему пользователю будет удобнее, если слева будут отображаться в одном дереве как *фирмы*, так и *подразделения*, которые им принадлежат. Это должно выглядеть так, как это изображено на рис. 8.6.

К сожалению, сама форма (без программирования) не может отобразить в этом элементе формы необходимые нам данные.

Для достижения поставленной цели мы изменим свойства этого элемента формы. А затем напишем в модуле формы процедуру заполнения дерева необходимыми данными.

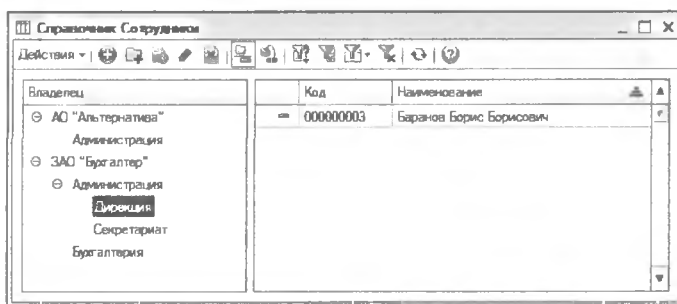


Рис. 8.6. Дерево значений, которое содержит список фирм и подразделений

Но сначала нам нужно будет выполнить подготовительные действия, выполнив следующее упражнение.

### Упражнение № 8.3

Удалите реквизит «СправочникДерево» из списка реквизитов формы «ОсновнаяФормаСписка» справочника «Сотрудники».

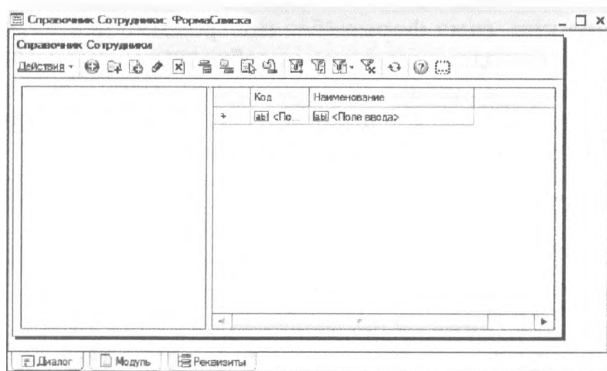
Следующие действия мы проделаем уже на закладке «Диалог». Здесь мы исправим свойства элемента диалога «Дерево».

### Упражнение № 8.4

У элемента формы «СправочникДерево» установите следующие свойства:

Свойство	Значение
Имя	<b>Дерево</b>
Данные	—
Тип значения	<b>ДеревоЗначений</b>
Только просмотр	<input checked="" type="checkbox"/>
Дерево	<input checked="" type="checkbox"/>

Если все сделали правильно, то форма будет выглядеть так, как это изображено на рис. 8.7.



**Рис. 8.7.** Форма списка после выполнения упражнения № 8.4

Для того чтобы дерево значений могло что-то отображать, необходимо в этом объекте вставить *колонку*.

### Упражнение № 8.5

Используя правую кнопку мыши, добавьте новую колонку и установите ее свойства:

Свойство	Значение
Имя	<b>Ссылка</b>
Данные	<b>Ссылка</b>
Только просмотр	<input checked="" type="checkbox"/>
Текст шапки	<b>Владелец</b>
Отображать иерархию	<input checked="" type="checkbox"/>
Ширина	<b>140</b>

Результат выполненного упражнения — на рис. 8.8.

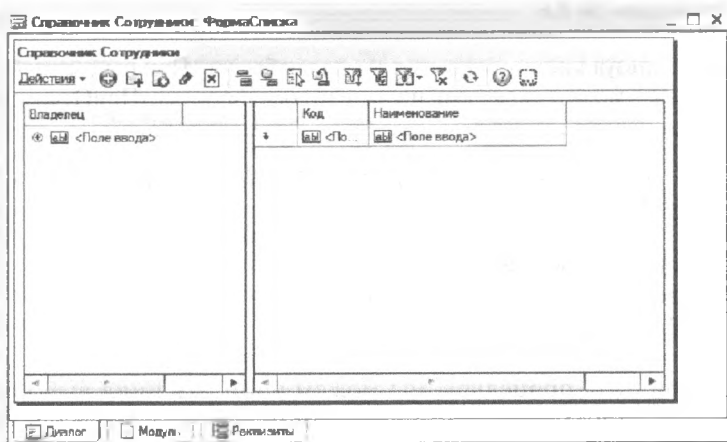


Рис. 8.8. Новая колонка в дереве значений

Теперь нам предстоит заполнить дерево конкретными значениями. Но когда (в какой момент) мы можем это сделать?

Для того чтобы понять, какое событие использовать, следует выделить саму форму как объект и в окне «Свойства» обратиться к группе свойств «События» (рис. 8.9).

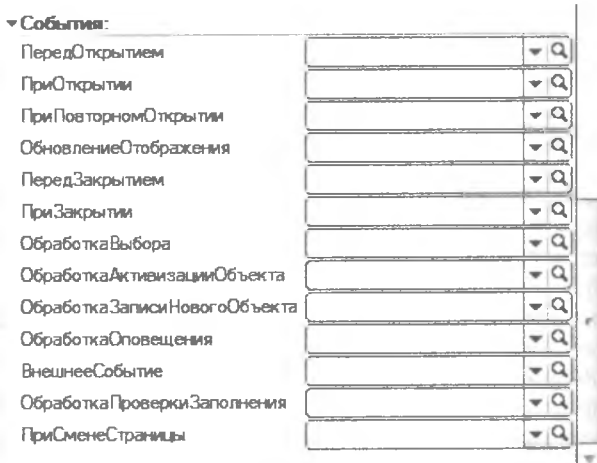


Рис. 8.9. События формы

Дерево мы будем заполнять *пока* однократно — «при открытии» формы, поскольку список фирм и подразделений с течением времени изменяется мало.

## Упражнение № 8.6

Используя кнопку открытия , для события «При открытии» сформируйте в модуле заголовок процедуры обработчика «ПриОткрытии».

После выполнения этого упражнения модуль формы будет содержать следующую заготовку:

```
Процедура ПриОткрытии()  
// Вставить содержимое обработчика.  
КонецПроцедуры
```

Внутри этой процедуры мы можем написать цикл выборки из справочника «Фирмы»:

```
//ПриОткрытии() – заполняет элемент формы «Дерево»  
//конкретными значениями  
Процедура ПриОткрытии()  
ВыборкаФирм = Справочники.Фирмы.Выбрать();  
Пока ВыборкаФирм.Следующий() Цикл  
Ссылка = ВыборкаФирм.Ссылка;  
КонецЦикла;  
КонецПроцедуры
```

Помещать *ссылку* мы будем в строку объекта типа «ДеревоЗначений», используя для хранения *ссылки* одноименную колонку.

Элемент формы «Дерево» расположен в самой форме. Для получения доступа к нему как к интерфейсному расширению соответствующего объекта в языке придется писать выражение *ЭлементыФормы.Дерево*. У этого объекта есть свойство *Значение*, которое и будет содержать ссылку на объект типа «ДеревоЗначения».

Сам объект *Дерево значений* имеет две коллекции (свойства) — *Колонки* и *Строки*. В нашем случае необходимо только увеличивать количество строк в дереве. Поэтому нам придется записать в модуле: *ЭлементыФормы.Дерево.Значение.Строки*.

Это выражение по типу данных в системе называется «*Коллекция-СтрокДереваЗначений*». У всех коллекций в языке «1С:Предприятия» есть метод *Добавить()*. Для данного типа данных этот метод вернет ссылку на добавленную строку.

Список фирм, который должен находиться на первом уровне вложенности в дереве, мы сможем получить, выполнив упражнение № 8.7.



## Упражнение № 8.7

---

Наберите в модуле формы следующий текст.



### ЛИСТИНГ Ю\_8.1

```
//ПриОткрытии() – заполняет элемент формы «Дерево»  
// конкретными значениями  
ВыборкаФирм = Справочники.Фирмы.Выбрать();  
Пока ВыборкаФирм.Следующий() Цикл  
СтрокаСФирмой = ЭлементыФормы.Дерево.Значение.Строки.Добавить();  
СтрокаСФирмой.Ссылка = ВыборкаФирм.Ссылка;  
КонецЦикла;
```

Обновите конфигурацию и проверьте работу формы списка справочника «Сотрудники».

---

Теперь необходимо внутри цикла «по фирмам» открыть аналогичный цикл «по подразделениям». Здесь мы сталкиваемся с необходимостью правильно выбрать метод выборки:

- ✓ либо это — Выбрать(Родитель, Владелец, Отбор, Порядок);
- ✓ либо это — ВыбратьИерархически(Родитель, Владелец, Отбор, Порядок).

Метод **.Выбрать(...)** будет выбирать записи, которые расположены на одном уровне. А метод **.ВыбратьИерархически(...)** будет выбирать все записи, расположенные на текущем и нижележащем уровнях.

При этом первый и второй параметры дадут нам возможность ограничить выборку (установить *отбор*) записями, которые удовлетворяют соответствующим значениям свойств «Родитель» и «Владелец».

Добавить новый уровень в дерево значений мы сможем, если для конкретной строки обратимся к ее свойству Строки, которое специально предназначено для обращения к строкам на нижележащем уровне:

```
СтрокаСПодразделением = СтрокаСФирмой.Строки.Добавить();
```

Поскольку справочник «Подразделения» многоуровневый с неограниченным количеством уровней, то нам удобнее будет организовать добавление нового подразделения в дерево в виде рекурсивного вызова одной процедуры:

```
//ДобавитьУровеньПодразделений(...) добавляет в «Дерево» подразделения  
Процедура ДобавитьУровеньПодразделений(  
СтрокаДерева, //Строка дерева куда добавляем  
Знач РодительПодразделения, //Уровень в справочнике  
Знач ВладелецПодразделения)//Владелец справочника  
ВыборкаПодразделений =  
Справочники.Подразделения.Выбрать(
```

```

РодительПодразделения, ВладелецПодразделения);
Пока ВыборкаПодразделений.Следующий() Цикл
Строка(Подразделением = СтрокаДерева.Строки.Добавить());
Строка(Подразделением.Ссылка = ВыборкаПодразделений.Ссылка;
ДобавитьУровеньПодразделений(
Строка(Подразделением,
ВыборкаПодразделений.Ссылка,
ВладелецПодразделения);
КонецЦикла;
КонецПроцедуры

```

У этой процедуры три параметра.

Первый задает строку в дереве, для которой мы будем добавлять новые строки на нижележащем уровне. Второй параметр указывает, для какого Родителя мы будем делать выборку. Назначение третьего параметра — определить *владельца* записей, которые мы будем выбирать.

Для того чтобы выбрать первый уровень в справочнике «Подразделения», нужно в качестве второго параметра передать *пустую ссылку*.

#### Упражнение № 8.7 (продолжение)

Наберите в модуле формы следующий текст.



#### ЛИСТИНГ Ю\_8.2

```

//ПриОткрытии() – заполняет элемент формы «Дерево»
// конкретными значениями
Процедура ПриОткрытии()
ВыборкаФирм = Справочники.Фирмы.Выбрать();
Пока ВыборкаФирм.Следующий() Цикл
Строка(Фирмой = ЭлементыФормы.Дерево.Значение.Строки.Добавить());
Строка(Фирмой.Ссылка = ВыборкаФирм.Ссылка;
ДобавитьУровеньПодразделений(
Строка(Фирмой,
Справочники.Подразделения.ПустаяСсылка(),
ВыборкаФирм.Ссылка);
КонецЦикла;
КонецПроцедуры
//ДобавитьУровеньПодразделений(...) добавляет в «Дерево» подразделения
Процедура ДобавитьУровеньПодразделений(
СтрокаДерева, //Строка дерева куда добавляем
Знач РодительПодразделения, //Уровень в справочнике
Знач ВладелецПодразделения) //Владелец справочника
ВыборкаПодразделений =
Справочники.Подразделения.Выбрать(
РодительПодразделения, ВладелецПодразделения);

```

```

Пока ВыборкаПодразделений.Следующий() Цикл
СтрокаПодразделением = СтрокаДерева.Строки.Добавить();
СтрокаПодразделением.Ссылка = ВыборкаПодразделений.Ссылка;
ДобавитьУровеньПодразделений(
СтрокаПодразделением,
ВыборкаПодразделений.Ссылка,
ВладелецПодразделения);
КонецЦикла;
КонецПроцедуры

```

Нам осталось сделать совсем немного, а именно: указать системе, что список, расположенный справа, связан с колонкой «Ссылка» дерева, которое расположено слева.

### Упражнение № 8.8

Выделите в элементе формы «Дерево» колонку «Ссылка» и установите, используя кнопку выбора, для нее свойство «Тип значения»:

Свойство	Значение
Тип значения	СправочникСсылка.Фирмы, СправочникСсылка.Подразделения

Чтобы проделать это, необходимо в диалоге «Редактирование типа данных» установить флаг «Составной тип» (рис. 8.10).

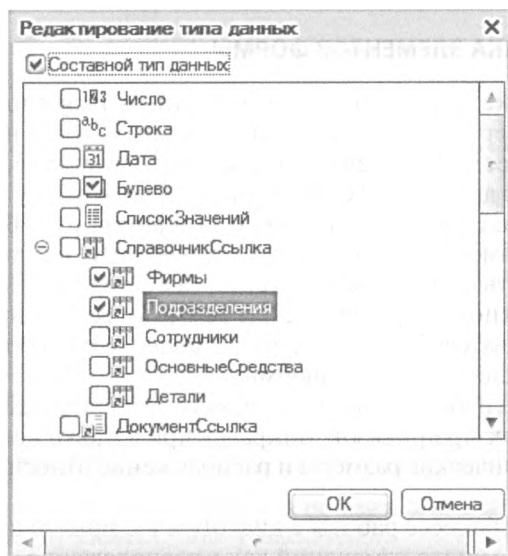


Рис. 8.10. Выбор составного типа данных

Указанные выше действия были необходимы для того, чтобы установить связь по владельцу в табличном поле «Список».

Выделите в диалоге табличное поле с именем «СправочникСписок».

Очистите свойство «Связь по родителю», используя для этого кнопку очистки. Используя кнопку выбора, установите свойство «Связь по владельцу» в значение: **ЭлементыФормы.Дерево.ТекущиеДанные.Ссылка** (рис. 8.11).

Проверьте работоспособность формы в режиме «1С:Предприятие».

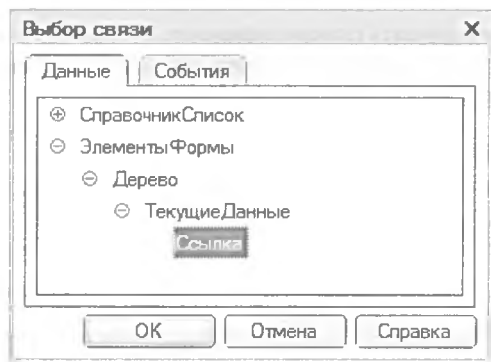


Рис. 8.11. Выбор связи с Владейцем

#### 8.4. ПРИВЯЗКА ЭЛЕМЕНТОВ ФОРМЫ

При разработке диалога формы важно стремиться к тому, чтобы начальный размер формы был бы минимальным, в ней не было много свободного места и элементы, размещенные в форме, не пересекались бы друг с другом при ее отображении на экране у пользователя.

При работе с диалогом формы в Конфигураторе Вы задаете его начальный размер, но пользователь в режиме «1С:Предприятие» способен изменять размер как по высоте, так и по ширине формы.

А что должно происходить с элементами формы при таких действиях пользователя? На этот вопрос и позволяет ответить настройка привязки каждого элемента формы.

В процессе работы с формой редактор диалогов самостоятельно выбирает характер привязок, опираясь при вставке нового элемента на его геометрические размеры и расположение относительно других элементов.

Однако в процессе работы с диалогом вы можете вносить достаточно большое число изменений как в расположение элементов, так и в их состав. Поэтому в таких случаях рекомендуется перед завер-

шением разработки формы обязательно тестировать ее поведение при изменении размеров.

Сделать это можно, вызвав пункт меню «Форма — Проверить», который позволяет увидеть, как будут изменяться размеры элементов при изменении размеров формы.

### Упражнение № 8.9

Откройте форму списка справочника «Сотрудники» и проверьте ее работу, вызвав пункт меню «Форма — Проверить»

Если вы все сделали правильно, вполне возможно, что окно с проверкой формы у вас выглядит так, как это изображено на рис. 8.12.

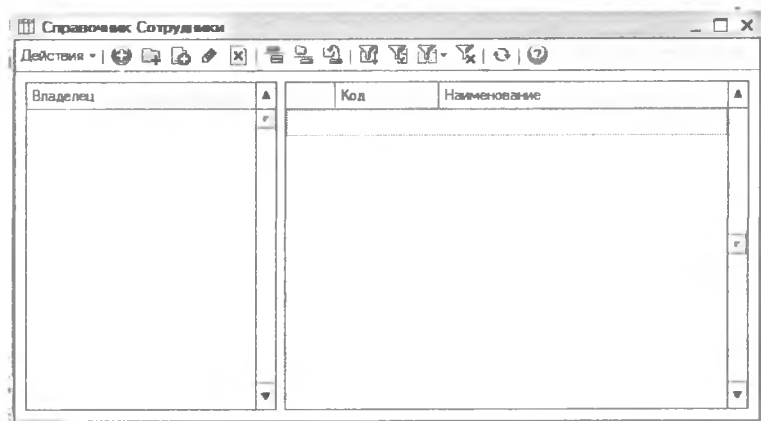


Рис. 8.12. Проверка формы списка справочника «Сотрудники»

Эта форма имеет один небольшой недостаток: у пользователя нет возможности регулировать взаимный размер двух табличных полей.

Для того чтобы пользователь получил возможность «двигать границу» между элементами, в форму между ними следует вставить элемент управления типа «Разделитель», который обладает способностью перемещаться пользователем мышью по форме.

### Упражнение № 8.10

Используя пункт меню «Форма — Вставить элемент управления...», вставьте в форму элемент типа «Разделитель».

Новый элемент следует расположить между элементами «Дерево» и «СправочникСписок» (рис. 8.13).

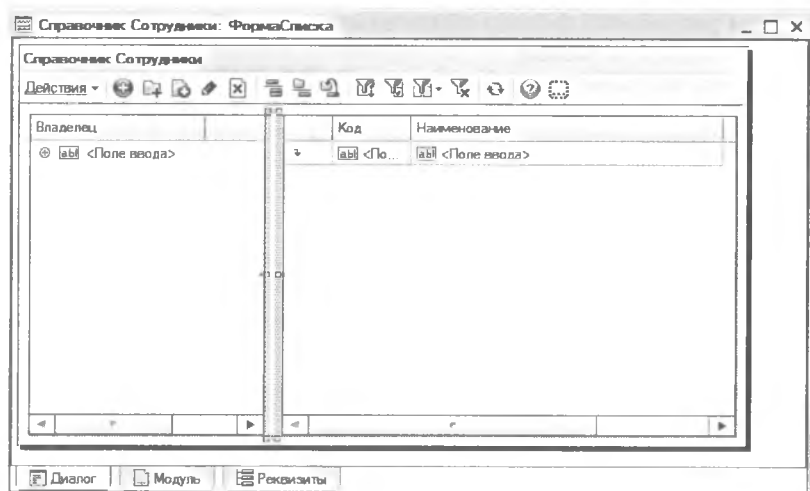


Рис. 8.13. Расположение разделителя в форме

При этом необходимо, чтобы вертикальный размер всех трех элементов был одинаковым, а расположение элементов не должно пересекаться.

Проверьте работу формы, в том числе и в режиме «1С:Предприятие».

Следующее упражнение предназначено для самостоятельного закрепления только что рассмотренного материала.

### Упражнение № 8.11

Вставьте в справочник «Подразделения» вертикальный разделитель. Сделайте его границу невидимой.

После завершения работы установите в свойствах Конфигурации номер версии, равным значению «Юнит 8».

Проверьте работу формы, в том числе и в режиме «1С:Предприятие».

## 8.5. ЧТО МЫ УЗНАЛИ

В этой главе мы рассмотрели основные приемы работы с формой как в Конфигураторе, так и в режиме «1С:Предприятие».

Напомним, что форма состоит из Диалога, Модуля и Реквизитов.

Диалог формы содержит отдельные элементы управления. Поведение самой формы в целом и ее элементов настраивается с помощью свойств этих объектов. Для большого числа случаев такая настройка может быть выполнена без применения встроенного языка системы.

В процессе работы пользователя с формой возникают различные события, которые относятся как к самой форме в целом, так и к ее отдельным элементам управления. Для обработки возникающих событий предназначен модуль формы, который хранит, в большинстве своем, процедуры — обработчики событий.

Использование реквизитов формы в сочетании со свойством «Данные» позволяет описать поведение как формы в целом, так и отдельных ее элементов. При этом реквизиты формы по сути своей определяют, какие данные должна отображать как форма в целом, так и конкретный реквизит в частности.

Для типовых ситуаций настройка списка реквизитов формы и свойства «Данные» позволяют обойтись без программирования. В качестве примера в этой главе рассматривалось отображение в одной форме одновременно списков двух справочников.

Приведем фрагмент постановки задачи:

«7.2. *Балансовая стоимость* выражена в рублях и может изменяться с течением времени. Желательно, чтобы изменения были оформлены документами.

7.3. *Рыночная стоимость* выражена в УЕ. Замечания, сделанные для балансовой стоимости, справедливы и для рыночной стоимости.

7.4. *Подразделение* может изменяться с течением времени, т.е. возможна передача ОС из одного подразделения в другое».

К сожалению, мы не можем создать такие реквизиты в справочнике, так как не сможем обеспечить хранение истории их изменения в этом объекте.

В тех случаях, когда возникает подобная задача, необходимо создавать два объекта — справочник и соответствующий ему периодический регистр сведений.

### 9.1. РЕГИСТР СВЕДЕНИЙ «ДанныеОС»

Поскольку Вы уже имеете опыт создания регистра сведений, постарайтесь выполнить следующее упражнение самостоятельно.

#### Упражнение № 9.1

Добавьте в Конфигурацию новый регистр сведений со следующими свойствами:

Свойство	Значение
Имя	ДанныеОС
Синоним	Данные ОС
Комментарий	—

Какие значения Вы выберете для свойств «Периодичность» и «Режим записи»?

Нажмите на кнопку «Далее».

Как мы помним, на втором этапе нужно определиться с измерениями, ресурсами и Реквизитами. Поскольку такое понятие, как «Балансовая стоимость», относится к конкретному основному средству, то необходимо в регистре создать соответствующее измерение,



для того чтобы установить логическую связь между этим двумя понятиями.

### Упражнение № 9.2

Добавляем в регистр сведений новое измерение:

Свойство	Значение
Группа свойств «Основные»	
Имя	ОС
Синоним	ОС
Комментарий	Основное средство
Ведущее	<input checked="" type="checkbox"/>
Запрет незаполненных значений	<input checked="" type="checkbox"/>
Группа свойств «Тип»	
Тип	СправочникСсылка.Основные-Средства

Дадим небольшое пояснение.

Свойство «Ведущее», будучи установленным, приводит к тому, что при удалении записи из справочника (т.е. из ведущего объекта) запись регистра, содержащая ссылку на этот элемент, также будет удалена автоматически.

Установка запрета незаполненных значений вполне логична. Ведь если в записи регистра не указано ОС, то данные о нем «повиснут в воздухе», чего не должно быть.

На этом измерения у нас закончатся.

В соответствии с постановкой задачи добавьте в регистр три ресурса.

### Упражнение № 9.2 (продолжение)

Добавляем в регистр сведений новый ресурс:

Свойство	Значение
Группа свойств «Основные»	
Имя	БалансоваяСтоимость
Синоним	Балансовая стоимость
Комментарий	Балансовая стоимость ОС
Группа свойств «Тип»	
Тип	Число

Свойство	Значение
Длина	14
Точность	2

Новые объекты в конфигурации можно создавать копированием, поэтому создайте новый ресурс путем копирования. Используйте комбинации клавиш Ctrl+C (скопировать объект) и Ctrl+V (добавить объект).

Свойство	Значение
Имя	<b>РыночнаяСтоимость</b>
Синоним	<b>Рыночная стоимость</b>
Комментарий	<b>Стоимость ОС в У.Е.</b>

И добавьте последний ресурс:

Свойство	Значение
Группа свойств «Основные»	
Имя	<b>Подразделение</b>
Синоним	<b>Подразделение</b>
Комментарий	—
Группа свойств «Тип»	
Тип	<b>СправочникСсылка.Подразделения</b>

Обратите внимание, что в последнем случае мы, добавив ресурс «Подразделение», привели пример использования в качестве ресурса нечислового типа данных.

Реквизитов в регистре не будет, так как они используются при проведении документов, а их у нас пока нет.

Следующее упражнение предназначено для самостоятельного закрепления рассмотренного в предыдущих юнитах материала.

### Упражнение № 9.3

Перейдите на закладку «Формы» и создайте основную форму списка регистра сведений.

Какие реквизиты Вы вставили в форму?

Измените, если хотите, геометрические размеры колонок табличного поля.

Эту форму мы создали для того, чтобы пользователь мог видеть весь регистр или какую-то часть записей «как есть». Здесь он может также редактировать записи. Однако для редактирования больше

подходит форма элемента из справочника «Основные Средства», так как наши данные связаны с конкретным объектом ОС.

## 9.2. ФОРМА ЭЛЕМЕНТА СПРАВОЧНИКА «ОСНОВНЫЕ СРЕДСТВА»

В этом разделе мы углубим наши знания по работе с формой.

Сначала мы создадим сам диалог, а затем используем необходимые модули для обработки событий.

### 9.2.1. Разработка диалога

Начнем с создания новой формы и размещения в ней элементов диалога.

#### Упражнение № 9.4

Вызовите конструктор формы и создайте для справочника «Основные Средства» форму, предварительно установив переключатель «Редактировать» в положение «В диалоге»:

Свойство	Значение
Выберите тип формы	Форма элемента справочника
Назначить форму основной	<input checked="" type="checkbox"/>
Имя	Основная Форма Элемента
Синоним	Основная форма элемента
Комментарий	–
Командная панель формы сверху	<input checked="" type="checkbox"/>
Командная панель формы снизу	<input checked="" type="checkbox"/>

Нажмите на кнопку «Далее» и сбросьте все флаги (рис. 9.1).

Такие действия объясняются просто — форма у нас будет достаточно сложная, поэтому мы будем вставлять реквизиты по одному, постепенно наращивая ее сложность.

Если вы все сделали правильно, то форма будет выглядеть так, как это изображено на рис. 9.2.

Перед началом работы с формой увеличим ее размер.

#### Упражнение № 9.5

Установите следующие свойства формы:

Свойство	Значение
Ширина	480
Высота	360

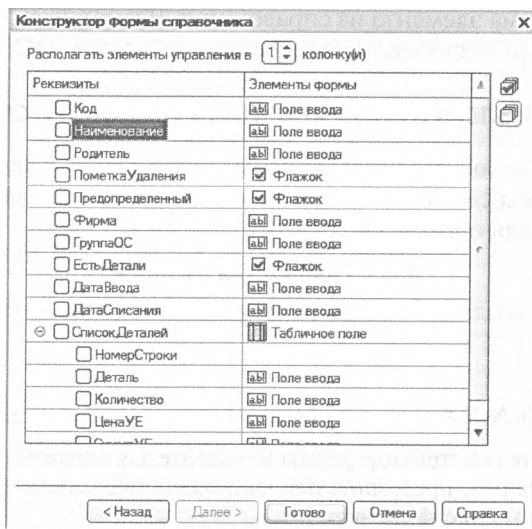


Рис. 9.1. Не вставлять в форму реквизиты!

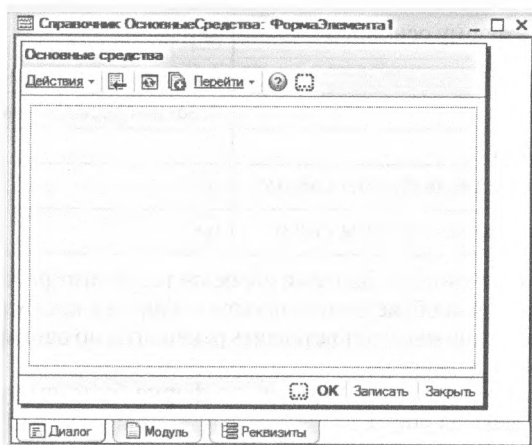


Рис. 9.2. Форма элемента перед размещением реквизитов

Самостоятельно, используя пункт меню «Форма — Размещение данных...», вставьте реквизиты «Фирма» и «Родитель» в соответствии с рис. 9.3.

Мы вставили эти реквизиты сверху потому, что пользователь должен сначала решить, к какой фирме относится новый элемент, в какую группу его следует записать, а потом уже вводить другие данные.

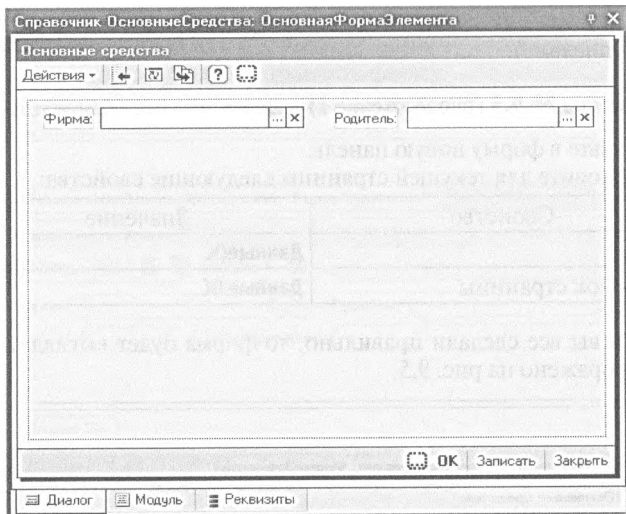


Рис. 9.3. Форма элемента после вставки реквизита «Фирмы» и «Родитель»

#### Упражнение № 9.5 (продолжение 1)

Вставьте реквизиты «Код» и «Наименование» (рис. 9.4).

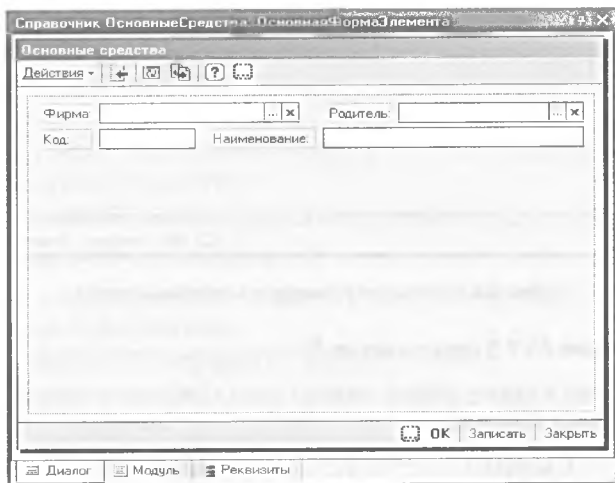


Рис. 9.4. Реквизиты «Код» и «Наименование»

У нас в этой форме будут два табличных поля:

- ✓ одно поле содержит записи из регистра сведений;
- ✓ другое поле — записи из табличной части справочника.

Поэтому мы сначала разместим в форме панель (не путать с командной панелью!).

### Упражнение № 9.5 (продолжение 2)

Вставьте в форму новую панель.

Установите для текущей страницы следующие свойства:

Свойство	Значение
Имя	ДанныеОС
Заголовок страницы	Данные ОС

Если вы все сделали правильно, то форма будет выглядеть так, как это изображено на рис. 9.5.

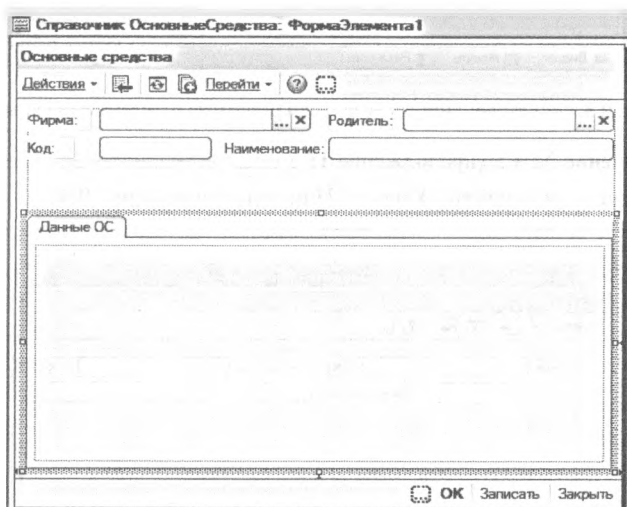


Рис. 9.5. Панель для размещения табличных полей

### Упражнение № 9.5 (продолжение 3)

Вставьте в форму новый элемент типа «Табличное поле», указав ему следующие свойства:

Свойство	Значение
Имя	НаборДанныеОС
Тип значения	РегистрСведенийНаборЗаписей. ДанныеОС
Вставить командную панель	<input checked="" type="checkbox"/>

И разместите элементы в панели, расположенной внизу формы. Удалите колонку «ОС» и измените ширину столбцов (рис. 9.6) для того, чтобы удобнее было настраивать форму. Что изменилось в списке реквизитов формы?

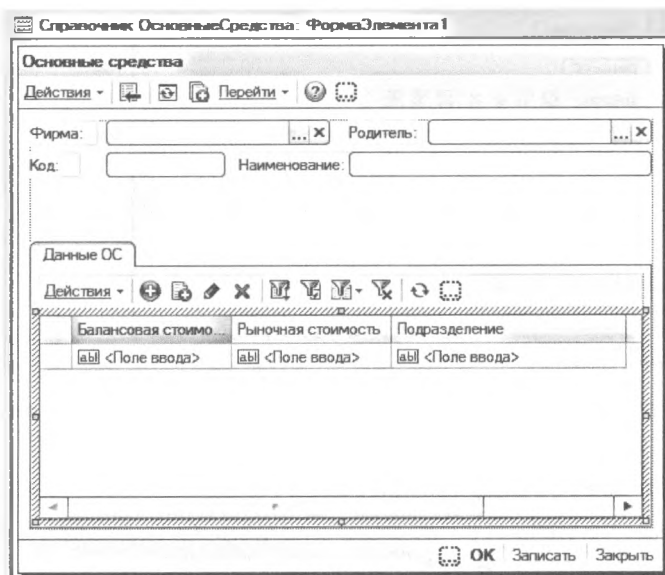


Рис. 9.6. Форма после вставки табличного поля

Колонку «ОС» мы удалили потому, что этот реквизит будем заполнять программным путем.

#### Упражнение № 9.5 (продолжение 4)

Самостоятельно вставьте в форму реквизиты «ГруппаОС», «ЕстьДетали», «ДатаВвода» и «ДатаСписания».

Для реквизита «ЕстьДетали» установите свойство:

Свойство	Значение
Положение заголовка	Текст слева

После вставки реквизитов измените их геометрические размеры и расположение таким образом, чтобы в форме не осталось «пустого» места (рис. 9.7).

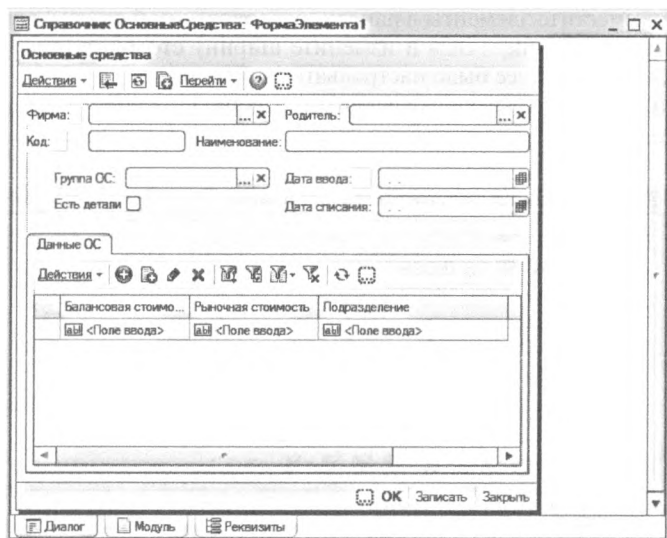


Рис. 9.7. Форма элемента после вставки всех реквизитов справочника

### Упражнение № 9.5 (продолжение 5)

Для того чтобы вставить табличную часть «СписокДеталей», добавьте в панели к странице «Данные ОС» еще одну страницу:

Свойство	Значение
Имя	СписокДеталей
Синоним	Список деталей

Вставьте на эту страницу все реквизиты из табличной части «Список-Деталей».

Для того чтобы пользователь мог управлять этим табличным полем, вставьте еще одну командную панель:

Свойство	Значение
Имя	КоманднаяПанель3
Заполнять автоматически	<input checked="" type="checkbox"/>
Источник действий	СписокДеталей
Вспомогательная	<input checked="" type="checkbox"/>

На этом мы закончили упражнение.

На этом мы завершили размещение элементов формы (рис. 9.8).



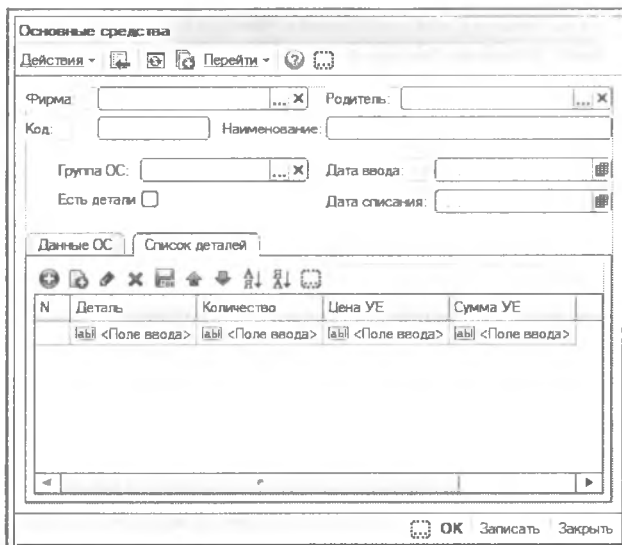


Рис. 9.8. Окончательный вид диалога

### 9.2.2. Обработка событий от элементов формы

На следующем этапе создания формы мы напишем обработчики событий от элементов формы.

Начнем с элемента «Фирма».

Если выделить этот элемент и открыть группу его свойств «События», то среди всех событий для таких элементов наиболее часто применяется обработка события «При изменении».

#### Упражнение № 9.6

Установите для элемента формы «Фирма» следующие свойства:

Свойство	Значение
Быстрый выбор	<input checked="" type="checkbox"/>
При изменении	<b>ПриИзмененииФирмы</b>

Переключитесь на закладку «Модуль». Что произошло?

Сейчас наш модуль содержит заготовку:

```
Процедура ПриИзмененииФирмы(Элемент)
// Вставить содержимое обработчика.
КонiecПроцедуры
```

Достаточно часто у начинающих программирование в этой системе возникает вопрос: «А что я могу использовать в этой точке программы?».

Этот вопрос сводится к другому вопросу: а что входит в локальный контекст этого модуля?

Для того чтобы понять это в «1С: Предприятие 8.3», нужно приложить минимум усилий.

### Упражнение № 9.7

Удалите текст комментария и нажмите Ctrl+пробел.

Что произошло?

Как мы видим на рис. 9.9, в текущий контекст попали как реквизиты нашего объекта (Фирма), так и методы справочника (УстановитьНовыйКод).

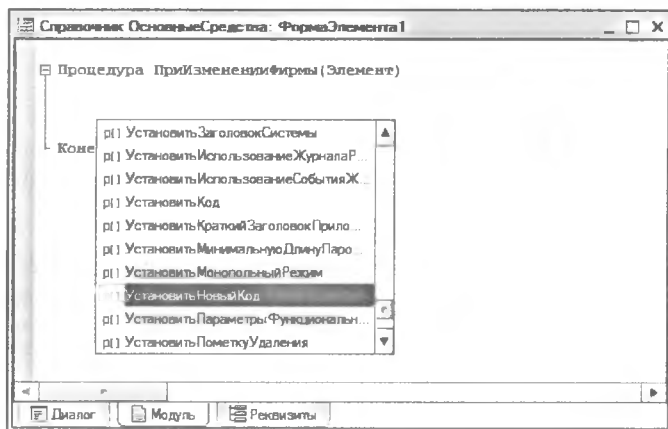


Рис. 9.9. Получение контекста в любой точке программы

Поэтому логично предположить, что текст обработчика может быть следующим:

```
Процедура ПриИзмененииФирмы(Элемент)
    УстановитьНовыйКод(Фирма.Префикс+«-»);
КонецПроцедуры
```

Отметим, что вызов метода **УстановитьНовыйКод()** вызовет изменение реквизита **Код**, т.е. одно из полей записи изменится.

Но что нам делать, если где-либо в конфигурации нужно написать:

**НовыйЭлемент=Справочники.ОсновныеСредства.СоздатьЭлемент();**

.....  
**НовыйЭлемент.Фирма = Фирма;**

.....  
Тогда вызов:

**НовыйЭлемент.УстановитьНовыйКод(Фирма.Префикс+«-»);**

придется писать еще раз ... и так много раз.

С методической точки зрения правильнее будет поместить эту обработку в Модуль объекта, так как она предназначена для изменения поля этого объекта.

Поэтому:

**Упражнение № 9.7 (продолжение)** \_\_\_\_\_

Наберите в Модуле Объекта:



**ЛИСТИНГ Ю\_9.1**

**// Модуль справочника «ОсновныеСредства»**

**//УстановитьКод() – устанавливает новый код**

**//для записи в справочнике**

**Процедура УстановитьКод() Экспорт**

**Если ЭтоГруппа=Ложь Тогда**

**// это элемент справочника**

**УстановитьНовыйКод(Фирма.Префикс+«-»);**

**КонецЕсли;**

**КонецПроцедуры**

Затем исправьте модуль формы:

**Процедура ПриИзмененииФирмы(Элемент)**

**УстановитьКод();**

**КонецПроцедуры**

Обратите внимание на добавочное ключевое слово **Экспорт** у процедуры **УстановитьКод()**. Оно привело к тому, что объект типа **СправочникОбъект.ОсновныеСредства** получил новый метод.

Аналогичную методику можно применить и для табличного поля «НаборДанныеОС».

**Упражнение № 9.8** \_\_\_\_\_

Выделите поле ввода «БалансоваяСтоимость» из табличного поля «НаборДанныеОС» и установите для этого элемента формы следующие свойства (рис. 9.10):

Свойство	Значение
При изменении	<b>ПриИзмененииБалансовойСтоимости</b>

Переключитесь на закладку «Модуль».  
Попробуйте самостоятельно написать процедуру:

```
//ПриИзмененииБалансовойСтоимости() – рассчитывает для записи  
//ее рыночную стоимость  
//ТекЗапись – запись, для которой производится расчет  
Процедура ПриИзмененииБалансовойСтоимости(Элемент)  
КонецПроцедуры
```

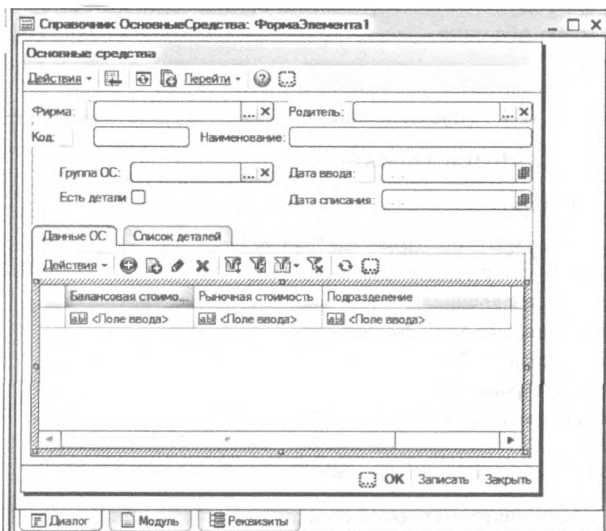


Рис. 9.10. Особенность выделения поля ввода в табличном поле формы

В табличном поле есть поле ввода «Подразделение», которое имеет тип СправочникСсылка.Подразделения, а справочник «Подразделения» подчинен справочнику «Фирмы», поэтому:

### Упражнение № 9.9

Установите у поля ввода «Подразделение» свойство «Связь по владельцу». В качестве владельца выступает справочник «Фирма».

В постановке задачи сказано, что «для тех ОС из группы «Машины и оборудование», что состоят из отдельных деталей...».

Таким образом, если в реквизите «ГруппаОС» выбрано значение «Машины и оборудование», то тогда пользователь должен иметь возможность обращаться к реквизитам формы «ЕстьДетали» и «Список-

Деталей», т.е. реквизиты должны быть видимы, а в противном случае — нет.

### Упражнение № 9.10

Выделите поле ввода «ГруппаОС» и установите для этого элемента формы следующие свойства:

Свойство	Значение
При изменении	ПриИзмененииГруппыОС

Переключитесь на закладку «Модуль» и наберите следующий текст:

#### ЛИСТИНГ Ю\_9.2

```
//ПриИзмененииГруппыОС – управляет видимостью
// реквизита «ЕстьДетали» и страницы «СписокДеталей»
Процедура ПриИзмененииГруппыОС(Элемент) Экспорт
_Видимость = (ГруппаОС =Перечисления.ГруппыОС.МашиныИОборудование);
ЭлементыФормы.ЕстьДетали.Видимость = _Видимость;
ЭлементыФормы.Панель1.Страницы.СписокДеталей.Видимость = _Видимость;
КонецПроцедуры
```

Сделаем следующие пояснения.

1. Ключевое слово **Экспорт** мы применили для того, чтобы можно было делать вызов этого алгоритма из внешней процедуры:

```
Форма =Справочники.ОсновныеСредства.ПолучитьФормуНовогоЭлемента();
Форма. ....
Форма.ПриИзмененииГруппыОС(Неопределено);
```

2. В этой процедуре мы обошлись без оператора «Если». Следующее упражнение сделайте самостоятельно.

### Упражнение № 9.11

Выделите поле ввода «ЕстьДетали» и установите для этого элемента формы следующие свойства:

Свойство	Значение
При изменении	ПриИзмененииФлага

Реализуйте самостоятельно текст этой процедуры.

После чего исправьте текст процедуры **ПриИзмененииГруппыОС(...)**:

```
Процедура ПриИзмененииГруппыОС(Элемент) Экспорт
_Видимость = (ГруппаОС =Перечисления.ГруппыОС.МашиныИОборудование);
ЭлементыФормы.ЕстьДетали.Видимость = _Видимость;
ЭлементыФормы.Панель1.Страницы.СписокДеталей.Видимость = _Видимость;
```

### 9.2.3. Самостоятельная работа

Выполните следующие упражнения самостоятельно.

#### Упражнение № 9.12

---

---

Сделайте так, чтобы при выборе детали в поле ввода «Деталь» цена из справочника «Детали» помещалась в реквизит «ЦенаУЕ» табличной части справочника «ОсновныеСредства».

---

---

В каком модуле вы расположили текст процедуры-обработчика?

#### Упражнение № 9.13

---

---

Сделайте так, чтобы при изменении в форме реквизитов **Количество**, **ЦенаУЕ** и **СуммаУЕ** производился как прямой расчет суммы, так и обратный расчет цены.

За выполнение расчета должна отвечать одна процедура **ПересчетСтроки(Элемент)**, которая должна являться обработчиком для трех полей ввода.

*Совет:* для того чтобы определить имя колонки, из которой сделан вызов, нужно написать следующее выражение:

**ИмяКолонки = ЭлементыФормы.СписокДеталей.ТекущаяКолонка.Имя;**

Нужно ли изменить текст обработчика поля ввода «Деталь»?

---

---

### 9.2.4. Обработка событий табличного поля

В нашей форме вставлены два табличных поля «НаборДанныеОС» и «СписокДеталей», каждое из которых в составе своих свойств имеет группу «События» (рис. 9.11).

В этом подразделе мы рассмотрим только некоторые из них.

«Разбор полетов» будем делать на примере табличного поля «НаборДанныеОС», тип значения которого — «РегистрСведенийНаборЗаписей». А раз так, то все записи из регистра сведений будут считываться за «один раз». Но все записи нам не нужны, поэтому:

#### Упражнение № 9.14

---

---

Установите для поля «НаборДанныеОС»:

Свойство	Значение
Обновлять при изменении отбора	<input checked="" type="checkbox"/>

---

---

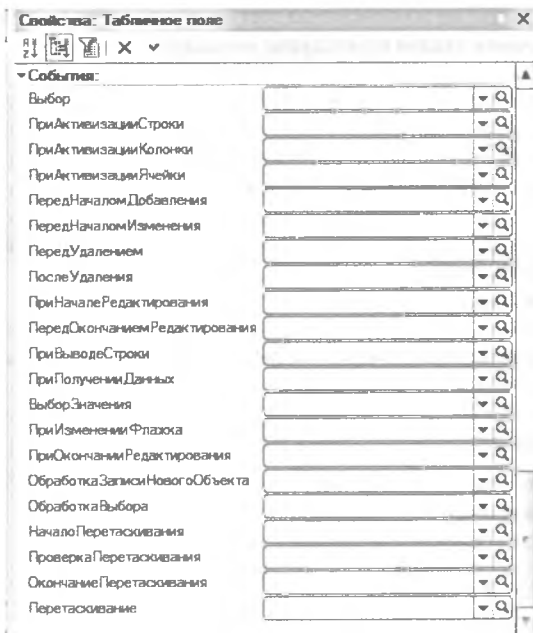


Рис. 9.11. Список событий табличного поля как элемента формы

### Предварительные рассуждения.

Пользователь начнет заполнять табличное поле «НаборДанныеОС» путем ввода новых строк, а раз так, то нужно реквизиту «ОС» регистра сведений «ДанныеОС» задать ссылку на основной реквизит формы рассматриваемого справочника «Основные средства», имеющего тип «СправочникОбъект.ОсновныеСредства».

Но ссылку мы сможем записать только в том случае, если элемент справочника уже существует, т.е. уже записан. Для управления ситуацией выполняем упражнение № 9.15.

### Упражнение № 9.15

Используя кнопку , установите для реквизита «НаборДанныеОС» события:

Свойство	Значение
Перед началом добавления	НаборДанныеОСПередНачаломДобавления
При начале редактирования	НаборДанныеОСПриНачалеРедактирования

Имена процедур в этом случае будут сгенерированы автоматически.

Первое событие возникает до того, как появится новая строка, и поэтому основная задача процедуры: отказать в добавлении новой строки в табличном поле.

Наберите в модуле следующий текст процедуры:



### ЛИСТИНГ Ю\_9.3

```
//НаборДанныеОСПередНачаломДобавления – проверяет
// возможность добавления новой строки
Процедура НаборДанныеОСПередНачаломДобавления(Элемент, Отказ, Копирование)
Если ЭтоНовый() Тогда
// откажем в добавлении строки:
Отказ = Истина;
Предупреждение(«Прежде чем добавлять запись необходимо записать элемент
справочника!»);
КонецЕсли;
КонецПроцедуры
```

А вот вторую процедуру мы можем использовать для заполнения ее реквизитов значениями по умолчанию.

Наберите в модуле следующий текст процедуры:



### ЛИСТИНГ Ю\_9.4

```
//НаборДанныеОСПриНачалеРедактирования – заполняет реквизит
// ОС ссылкой на текущий элемент справочника
Процедура НаборДанныеОСПриНачалеРедактирования(Элемент, НоваяСтрока,
Копирование)
Если НоваяСтрока Тогда
ТекСтрока = ЭлементыФормы.НаборДанныеОС.ТекущаяСтрока;
ТекСтрока.ОС = Ссылка;
КонецЕсли;
КонецПроцедуры
```

Для табличного поля «НаборДанныеОС» больше не требуется обрабатывать какие-либо события. А вот для поля «СписокДеталей» необходимо обеспечить, чтобы поле ввода «Деталь» всегда содержало какое-либо значение. Поэтому выполним следующее упражнение.

### Упражнение № 9.16

Нажмите на кнопку открытия  и установите для реквизита «СписокДеталей» событие:

Свойство	Значение
Перед окончанием редактирования	СписокДеталейПередОкончаниемРедактирования

И наберите следующий текст:





## ЛИСТИНГ Ю\_9.5

```
//СписокДеталейПередОкончаниемРедактирования – проверяет
// правильность заполнения строки
Процедура СписокДеталейПередОкончаниемРедактирования(Элемент,
НоваяСтрока, ОтменаРедактирования,Отказ)
// Если пользователь отказался от ввода новой строки
Если НоваяСтрока И ОтменаРедактирования Тогда
// то нам все равно
Возврат;
КонецЕсли;
// Если пользователь подтверждает ввод строки
Если Не ОтменаРедактирования Тогда
ТекСтрока = ЭлементыФормы.СписокДеталей.ТекущаяСтрока;
Если ТекСтрока.Деталь.Ссылка.Пустая() Тогда
Предупреждение(«Не выбрана деталь!»);
Отказ = Истина;
Возврат;
КонецЕсли;
КонецЕсли;
КонецПроцедуры
```

---

Нужно иметь в виду, что эта процедура будет получать управление в двух случаях:

- ✓ когда пользователь отменил редактирование, нажав Esc;
- ✓ когда пользователь закончил редактирование, нажав Enter.

Именно поэтому в процедуре проверяется соответствующая комбинация входных параметров **НоваяСтрока** и **ОтменаРедактирования**.

На этом мы закончим рассматривать обработку событий табличного поля и перейдем к рассмотрению событий самой формы в целом.

### 9.2.5. Обработка событий от формы в целом

Напомним список основных событий, которые есть в форме:

- ✓ Перед открытием;
- ✓ При открытии;
- ✓ Обновление отображения;
- ✓ Перед закрытием;
- ✓ При закрытии;
- ✓ При изменении данных;
- ✓ Перед записью;
- ✓ При записи;
- ✓ После записи.

Здесь мы перечислили наиболее часто встречающиеся события, которые возникают практически во всех формах.

Давайте разберемся по порядку.

**Перед открытием.** Возникает до того, как диалог будет отображен на экране. Обычно используется для того, чтобы отказаться от открытия формы.

А раз так, то нужно до открытия формы установить отбор в поле «НаборДанныеОС».

#### Упражнение № 9.17

---

Включите для формы обработку события «Перед открытием».

```
//ПередОткрытием – установим отбор в наборе записей регистра
Процедура ПередОткрытием(Отказ, СтандартнаяОбработка)
НаборДанныеОС.Отбор.ОС.Установить(Ссылка);
КонецПроцедуры
```

В нашем случае мы всегда будем открывать форму. Поэтому формальный параметр «Отказ» оставляем без изменения.

---

**При открытии.** Возникает также до открытия диалога, но после события «Перед открытием», если мы не отказались от открытия диалога. Обычно это событие используется для подготовительных действий перед открытием формы. Такими действиями могут быть:

- ✓ заполнение реквизитов формы начальными значениями;
- ✓ установка свойств формы и ее элементов в начальные значения.

Когда мы можем заполнить реквизиты начальными значениями? Тогда и только тогда, когда мы открываем форму нового объекта (записи) в ИБ.

Для справочников, точнее у объекта типа «СправочникОбъект», есть метод **ЭтоНовый()**, который возвращает *истину*, если объект был только что создан, но еще не записан.

#### Упражнение № 9.18

---

Нажмите на кнопку  для события «При открытии» в свойствах формы и наберите в модуле формы следующий текст:



#### ЛИСТИНГ Ю\_9.6

```
Процедура ПриОткрытии()
Если ЭтоНовый() Тогда
// Заполним реквизиты нового объекта начальными значениями
Фирма = Константы.ОсновнаяФирма.Получить();
УстановитьКод();
КонецЕсли;
```

// Установим свойство видимость у элементов диалога перед его

// открытием

ПриИзмененииГруппыОС(Неопределено);

КонецПроцедуры

Поскольку начальное значение реквизита «Фирма» мы будем сохранять в константе **ОсновнаяФирма**, то добавьте ее в конфигурацию:

Свойство	Значение
Имя	<b>ОсновнаяФирма</b>
Синоним	<b>Основная фирма</b>
Комментарий	<b>Значение по умолчанию</b>
Тип	<b>СправочникСсылка.Фирмы</b>

Создайте новую основную форму констант и установите ее высоту равной 200 пикселям (**Константы – Правая кнопка – Создать форму констант**), установите флажки в соответствии с рис. 9.12. Удалите старую форму констант для управляемого приложения (**Общие – Общие формы – ОсновнаяФорма**).

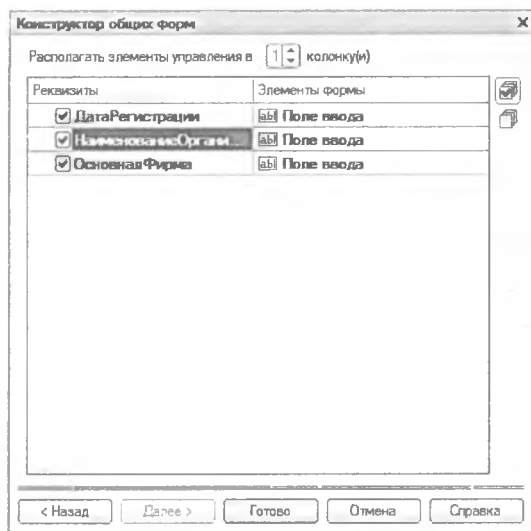


Рис. 9.12. Выбор данных и элемента формы

Если вы все сделали правильно, то основная форма констант будет выглядеть так, как это изображено на рис. 9.13.

Теперь мы можем поработать с формой элемента из режима «1С:Предприятие».

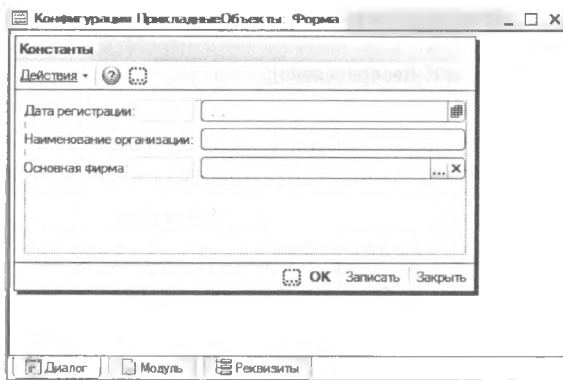


Рис. 9.13. Основная форма констант после доработки

### Упражнение № 9.19

В режиме «ИС:Предприятие», используя пункт меню «Прочие — Константы», выберите в качестве значения константы «Основная фирма» фирму АО «Альтернатива» и сохраните сделанные изменения в ИБ.

Используя пункт меню «Операции — Справочник...», откройте форму списка справочника «Основные средства» и добавьте в него новый элемент, заполнив его реквизиты следующими значениями:

Реквизит	Значение
Фирма	АО «Альтернатива»
Родитель	—
Код	АЛ-00001
Наименование	Головной офис
Группа ОС	Здания
Дата ввода	02.02.2008

Попробуйте добавить на закладке «Данные ОС» новую строку. Нажмите на кнопку ОК для сохранения элемента и записи его в ИБ.

Как мы видим, наши процедуры-обработчики успешно отработали и выполнили все возложенные на них функции.

Но вернемся к рассмотрению событий формы.

После того как форма открыта, пользователь работает с формой, в результате чего возникают события, как относящиеся к форме в целом, так и порождаемые элементами формы.

Если пользователь изменил объект ИБ, ему будет предложено сохранить введенные данные. В случае положительного ответа на вопрос системы будет последовательно возникать события:

- ✓ Перед записью;
- ✓ При записи;
- ✓ После записи.

При этом нужно помнить, что в модуле объекта вы также можете обрабатывать события:

- ✓ Перед записью;
- ✓ При записи.

При этом обработчики событий в форме получают управление раньше одноименных обработчиков событий в модуле объекта.

Задача обработчика **ПередЗаписью(...)** — проверить возможность записи объекта в ИБ и, если есть необходимость, *отказать* в этом.

Поэтому мы в модуле объекта реализуем функцию **ЕстьОшибка()**, которая будет возвращать Имя реквизита, содержащего с точки зрения программиста неверные данные.

Тогда при обработке события «Перед записью», как в форме, так и в модуле объекта, ее можно вызвать, чтобы определить — есть ли ошибки в объекте или нет.

На первый взгляд кажется, что оба обработчика выполняют одни и те же функции, но это не так.

Обработчик **ПередЗаписью(Отказ)** из модуля формы в случае обнаружения ошибки сможет активизировать соответствующий реквизит формы, установив туда курсор, и выдать при этом предупреждение пользователю.

Обработчик **ПередЗаписью(Отказ)** из модуля объекта в случае обнаружения ошибки сможет лишь только сообщить об этом в окне сообщений. Предпринимать какие-либо интерактивные действия в этом обработчике нельзя, так как он может циклически вызываться из какой-либо обработки верхнего уровня.

## Упражнение № 9.20

---

Наберите в модуле объекта следующий текст.



### ЛИСТИНГ Ю\_9.7

```
//ЕстьОшибка() – возвращает имя реквизита с ошибкой
//Если ошибок нет, то: «»
Функция ЕстьОшибка() Экспорт

// Группу записываем всегда!
Если ЭтоГруппа Тогда
Возврат «»;
КонецЕсли;
//Ошибкой будем считать пустые реквизиты
//«Фирма» и «ГруппаОС»
Если Фирма.Ссылка.Пустая() Тогда
```

```
Возврат «Фирма»;
КонецЕсли;
Если ГруппаОС.Пустая() Тогда
Возврат «ГруппаОС»;
КонецЕсли;
// ошибок нет
Возврат «»;
КонецФункции
```

Теперь в том же модуле можно и обработчик добавить.

---

### Упражнение № 9.20 (продолжение)

---

Наберите в модуле объекта следующий текст.



#### ЛИСТИНГ Ю\_9.8

```
// Проверяет правильность заполнения элемента справочника
Процедура ПередЗаписью(Отказ)
Имя = ЕстьОшибка();
Если Имя = «Тогда»
Возврат; // ошибок нет
КонецЕсли;
Сообщить(«Реквизит»+Имя+« не содержит данных!»);
СтатусСообщения.Важное);
Сообщить(«Запись»+ ЭтотОбъект+« справочника не записана!»);
СтатусСообщения.ОченьВажное);
Отказ = Истина;
КонецПроцедуры
```

Обработчик в форме будет получать управление до обработчика из модуля объекта, поэтому в случае отказа в этом обработчике «дело не дойдет» до модуля объекта.

Включите в модуле формы обработку события «Перед записью» и наберите следующий текст.



#### ЛИСТИНГ Ю\_9.9

```
//ПередЗаписью - проверяет правильность заполнения элемента справочника
Процедура ПередЗаписью(Отказ)
Имя = ЕстьОшибка();
Если Имя = «Тогда»
Возврат; // т.к. нет ошибок
КонецЕсли;
Предупреждение(«Реквизит »+Имя+« не содержит данных!»);
// установим курсор в нужное поле;
ТекущийЭлемент = ЭлементыФормы[Имя];
```

**Отказ = Истина;**  
**КонецПроцедуры**

---

---

Наша форма содержит два реквизита: «**НаборДанныеОС**» и «**СправочникОбъект**», причем последний из них является *основным реквизитом*. Именно его система будет записывать по команде «Обновить».

Поэтому только событие «При записи» может быть использовано для записи «НаборДанныеОС».

### Упражнение № 9.21

---

---

Включите в модуле формы обработку события «При записи» и наберите следующий текст.

#### ЛИСТИНГ Ю\_9.10

```
//ПриЗаписи – записывает набор записей регистра сведений
Процедура ПриЗаписи(Отказ)
Попытка
НаборДанныеОС.Обновить();
Исключение
Предупреждение(«Не удалось сохранить«Данные ОС»!»);
Отказ = Истина
КонецПопытки;
КонецПроцедуры
```

---

---

Процедура **ПослеЗаписи()** получает управление только тогда, когда запись успешно завершена (транзакция закрыта).

В нашем случае это событие можно и нужно применить для того, чтобы переустановить отбор в наборе записей, как минимум, в случае, если записали новый объект.

### Упражнение № 9.22

---

---

Включите в модуле формы обработку события «После записи» и наберите следующий текст.

#### ЛИСТИНГ Ю\_9.11

```
//ПослеЗаписи() – переустанавливает отбор после записи
Процедура ПослеЗаписи()
НаборДанныеОС.Отбор.ОС.Установить(Ссылка);
КонецПроцедуры
```

---

---

## 9.2.6. Изменение привязок элементов

Когда Вы заполняли форму элемента в режиме «1С: Конфигуратор», то, наверное, обратили внимание на то, что пропорции некоторых элементов и их расположение не соответствуют тому, что есть в режиме «1С:Предприятие».

Это объясняется тем, что привязка, установленная Конфигуратором по умолчанию, не соответствует вашему ожидаемому поведению формы при изменении ее размеров.

### Упражнение № 9.23

Запустите основную форму элемента на проверку в режиме «Конфигуратор».

Что вы видите (рис. 9.14)? Какие элементы формы требуют доработки?

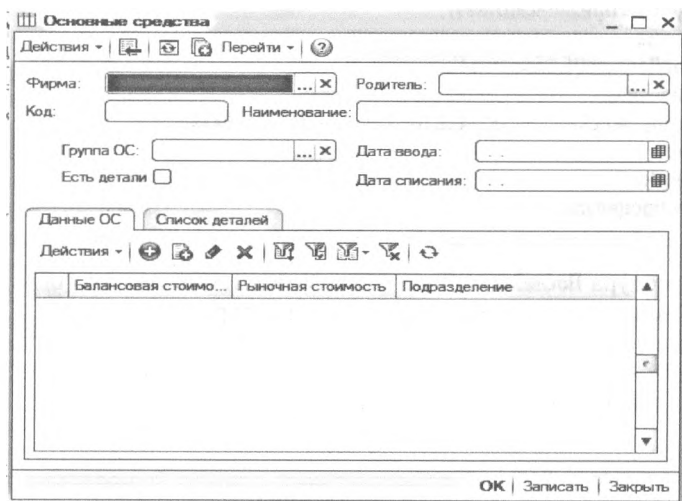


Рис. 9.14. Окно проверки основной формы элемента

### Упражнение № 9.23 (продолжение)

Будем действовать последовательно.

Для элемента формы «Фирма» в его свойствах для свойства «Привязка границ» откройте диалог «Привязка границ...», нажав на ссылку «Установить правила», и на закладке «Сложная» установите свойства согласно рис. 9.15.



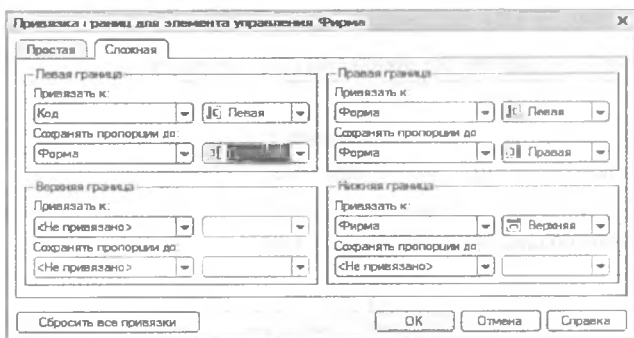


Рис. 9.15. Привязки элемента «Фирма»

Проверьте работу формы. Что изменилось?

Установка привязки левой границы к элементу «Код» позволит зафиксировать левую границу этого элемента, а сохранение горизонтальных пропорций для правой границы формы приведет к тому, что ширина элемента будет изменяться пропорционально.

Самостоятельно, используя закладку «Сложная привязка», добейтесь такого поведения формы, чтобы при увеличении размеров она выглядела согласно рис. 9.16.

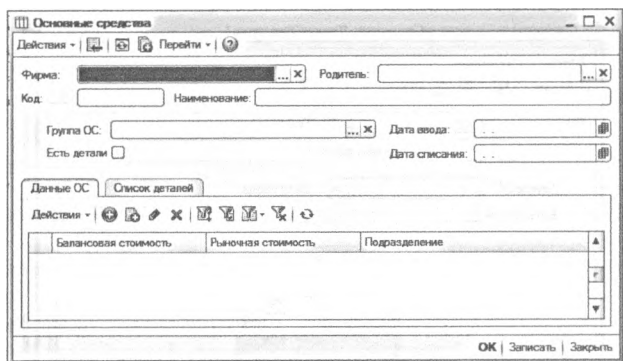


Рис. 9.16. Работа привязок элементов формы

### 9.2.7. Отображение картинки

Непосредственно вставить в форму реквизит «Изображение» мы не можем, так как его тип «ХранилищеЗначения», в котором может быть сохранено все что угодно.

Но мы можем вставить в форму элемент формы типа «Поле картинки», в котором можно будет отобразить изображение.

## Упражнение № 9.24

Добавьте в панель, расположенную в форме, новую закладку:

Свойство	Значение
Имя	Картинка
Заголовок страницы	Картинка

После чего вставьте на эту страницу элемент управления типа «Поле картинки» со следующими свойствами:

Свойство	Значение
Имя	ПолеКартинки
Рамка	Выпуклая
Картинка	—
Размер картинки	Пропорционально
Гиперссылка	<input checked="" type="checkbox"/>

И расположите ее на свободном пространстве (рис. 9.17).

Теперь необходимо установить привязки у этого элемента формы, чтобы его размеры также пропорционально изменялись с изменением размеров формы.

Установите привязки у поля картинки согласно рис. 9.18.

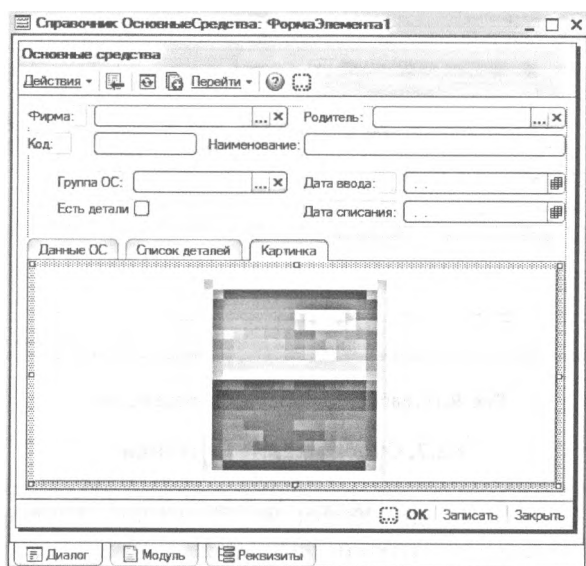


Рис. 9.17. Поле картинки в форме

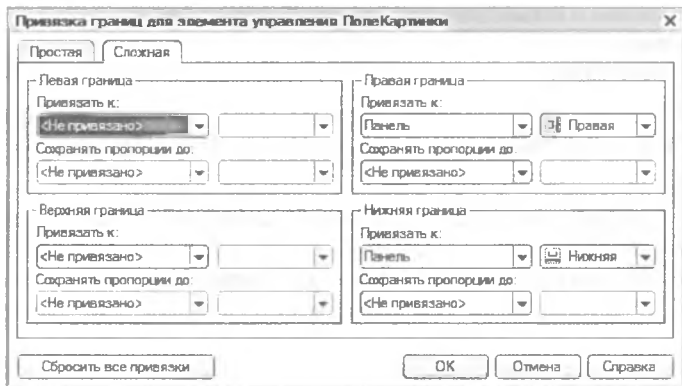


Рис. 9.18. Привязки поля картинки

Теперь нам нужно решить две проблемы.

1. Как прочитать новую картинку из файла в элементе формы и сохранить ее в хранилище значений.

2. Как извлечь картинку из хранилища и отобразить ее.

Для того чтобы прочитать картинку из файла, давайте обработаем *нажатие* на картинку в форме.

### Упражнение № 9.25

Установите для элемента формы «ПолеКартинки» событие:

Свойство	Значение
Нажатие	ПолеКартинкиНажатие

И наберите следующий текст.

### ЛИСТИНГ Ю\_9.12

```
//ПолеКартинкиНажатие – читает картинку в поле картинки
Процедура ПолеКартинкиНажатие(Элемент)
// Открываем диалог выбора файла
Диалог = Новый ДиалогВыбораФайла(РежимДиалогаВыбораФайла.Открытие);
Диалог.Заголовок = «Выберите файл картинки»;
Диалог.ПредварительныйПросмотр = Истина;
Диалог.Фильтр = «Точечный рисунок(*.jpg)*.jpg»;
// Если пользователь сделал выбор тогда
Если Диалог.Выбрать() Тогда
// Прочитаем картинку в ОП
НоваяКартинка = Новый Картинка(Диалог.ПолноеИмяФайла);
// и отобразим в форме:
ЭлементыФормы.ПолеКартинки.Картинка = НоваяКартинка;
```

**Изображение = Новый ХранилищеЗначения(НоваяКартинка);**  
**КонецЕсли;**  
**КонецПроцедуры**

Запустите режим «1С:Предприятие» и проверьте работу алгоритма.  
На закладке «Картинка» добавьте фотографию Финакадемии.

---

---

При записи объекта реквизит «Изображение» будет сохранен в ИБ.  
А вот когда мы будем открывать форму, то нужно отобразить реквизит «Изображение» в картинку, если объект неновый. Если объект новый, то тогда следует в картинку отобразить *пустую картинку*.

### Упражнение № 9.26

---

---

Исправьте текст процедуры **ПриОткрытии()**.



#### ЛИСТИНГ Ю\_9.13

```
// ПриОткрытии – действия в момент открытия формы
Процедура ПриОткрытии()
Если ЭтоНовый() Тогда
// Заполним реквизиты нового объекта начальными значениями
Фирма = Константы.ОсновнаяФирма.Получить();
УстановитьКод();
// запишем пустую картинку
Изображение = Новый ХранилищеЗначения(Новый Картинка());
КонецЕсли;
// Установим свойство видимость у элементов диалога перед его
// открытием
ПриИзмененииГруппыОС(Неопределено);
// отобразим картинку
ЭлементыФормы.ПолеКартинки.Картинка = Изображение.Получить();
КонецПроцедуры
```

Запустите режим «1С:Предприятие» и проверьте работу алгоритма.

---

---

## 9.3. САМОСТОЯТЕЛЬНАЯ РАБОТА

Следующие упражнения предназначены для закрепления материала, изложенного выше.

### Упражнение № 9.27

---

---

Самостоятельно создайте для справочника «**ОсновныеСредства**» основную форму группы (рис. 9.19).

Какие события вы будете обрабатывать в этой форме?

От каких ее элементов?

Сделайте так, чтобы коды групп в справочнике «ОсновныеСредства» нумеровались в системе в следующем порядке: «00000001», «00000002», и т.д. до «00099999».

Чтобы проверить работу формы, введите в справочник новую группу с наименованием «Здания и сооружения».

Нужно ли что-то изменить в модуле формы или в модуле объекта?

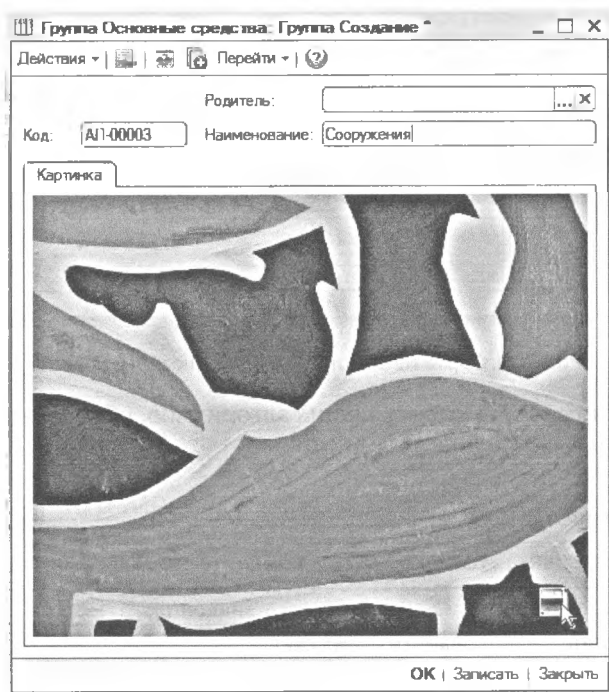


Рис. 9.19. Форма группы

### Упражнение № 9.28

Самостоятельно создайте для справочника «ОсновныеСредства» основную форму списка (рис. 9.20).

Кнопка «Печать» нам понадобится позже.

Вызовите Конструктор печати и создайте печатную форму списка справочника, которая должна вызываться по кнопке «Печать».

На втором шаге выберите следующие реквизиты:

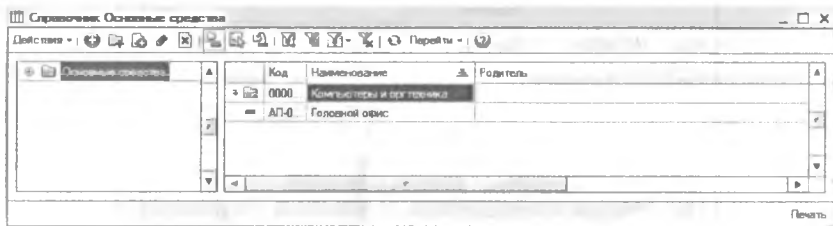
- Код;
- Наименование;

- **Фирма;**
- **ГруппаОС;**
- **ДатаВвода;**
- **ДатаСписания.**

Обратите внимание, что Конструктор печати при распечатывании формы списка справочника не включает в процедуру печати возможность распечатывать табличные части справочников, если они есть.

Вставьте открытие этой формы в основной интерфейс конфигурации.

Проверьте работу кнопки в режиме «1С:Предприятие».



**Рис. 9.20.** Форма списка справочника «ОсновныеСредства»

Чтобы проверить работу наших алгоритмов, обеспечьте наличие в справочнике следующих записей.

### Упражнение № 9.29

Для элемента «Головной офис» измените его Родителя:

Реквизит	Значение
Родитель	<b>Здания и сооружения</b>

На закладке «Данные ОС» наберите:

Период	Балансовая стоимость	Рыночная стоимость	Подразделение
31.12.2008	10 000 000,00	402 414,49	—

Нажмите на кнопку ОК. Что изменилось в ИБ?

Проверим установку префикса для другой «Фирмы»:

Реквизит	Значение
Фирма	<b>ЗАО «Бухгалтер»</b>
Родитель	<b>Здания и сооружения</b>
Код	<b>БХ-00001</b>
Наименование	<b>Крытый павильон</b>

Группа ОС	Сооружения
Дата ввода	14.01.2008
Дата списания	—

И на закладке «Данные ОС» наберите:

Период	Балансовая стоимость	Рыночная стоимость	Подразделение
14.01.2009	200 000,00	8 408,87	—
28.12.2009	250 000,00	10 060,36	Администрация

Нажмите на кнопку ОК. Что изменилось в ИБ?

Давайте введем элемент с деталями:

Реквизит	Значение
Фирма	АО «Альтернатива»
Родитель	Компьютеры и оргтехника
Код	АЛ-00002
Наименование	Компьютер
Группа ОС	Машины и оборудование
Дата ввода	10.12.2008
Дата списания	—
Есть детали	<input checked="" type="checkbox"/>

Переходим на закладку «Список деталей»:

Деталь	Количество	Цена УЕ	Сумма УЕ
Материнская плата	1	115,00	115,00
Процессор 3ГГц	2	103,00	206,00
ОП DIMM/DDR 256 Мб	2	88,00	176,00
Винчестер 40Гб	1	100,00	100,00
CD-ROM 52x	1	32,00	32,00
FDD 3.5	1	10,00	10,00

Сохраните элемент в ИБ.

#### 9.4. МЕТОД СРЕЗПОСЛЕДНИХ(...) МЕНЕДЖЕРА РЕГИСТРА СВЕДЕНИЙ

В модуле формы списка мы поместили кнопку «Печать», работа которой приводит к формированию табличного документа (рис. 9.21).

Код	Наименование	Фирма	Группа ОС	Дата ввода	Дата списания
0000002	Здания и сооружения				
АЛ-00018	Головной офис	АО "Альтернатива"	Здания	02.02.2008	
БХ-00002	Крытый павильон	ЗАО "Бухгалтер"	Сооружения	14.01.2008	
0000001	Компьютеры в оргтехнике				
АЛ-00002	Компьютер	АО "Альтернатива"	Машины и оборудование	10.12.2008	

Рис. 9.21. Результат работы кнопки «Печать»

Однако в этой печатной форме нет информации о балансовой и рыночной стоимости, которая хранится в регистре сведений.

Поэтому можно доработать отчет, поместив в него информацию из регистра сведений «ДанныеОС». Так как этот регистр периодический, то извлекать данные можно только на какую-то дату. Для упрощения задачи мы будем строить отчет на *Рабочую дату*.

Считывать данные по одной записи, используя метод **ПолучитьПоследнее(...)**, — не очень эффективное решение, поэтому мы воспользуемся другим методом: **СрезПоследних(...)**.

Этот метод возвращает объект типа «Таблица значений». Он может содержать переменное число строк и столбцов.

### Упражнение № 9.30

Измените настройку ячеек в столбцах «Фирма», «Группа ОС», «Дата ввода» и «Дата списания», чтобы улучшить представленные в них данные.

Добавьте три столбца «Балансовая стоимость», «Рыночная стоимость» и «Подразделение» (рис. 9.22).

1	2	3	4	5	6	7	8	9	10
Основные средства									
Код	Наименование	Фирма	Группа ОС	Дата ввода	Дата списания	Балансовая стоимость	Рыночная стоимость	Подразделение	
«АЛ»	«Здания и сооружения»	«АО "Альтернатива"»	«Здания»	«02.02.2008»	«14.01.2008»	«02.02.2008»	«14.01.2008»	«АО "Альтернатива"»	«ЗАО "Бухгалтер"»
«БХ»	«Крытый павильон»	«ЗАО "Бухгалтер"»	«Сооружения»	«14.01.2008»					
«АЛ»	«Компьютеры в оргтехнике»	«АО "Альтернатива"»	«Машины и оборудование»	«10.12.2008»					
«АЛ»									
«АЛ»									
«АЛ»									
«АЛ»									
«АЛ»									
«АЛ»									
«АЛ»									
«АЛ»									
«АЛ»									

Рис. 9.22. Макет печатной формы после доработки

Для того чтобы получить срез последних записей, нам необходимо вставить в процедуру «Печать» до начала выборки записей из справочника следующую строку текста:



// Получение среза

ТаблицаСреза = РегистрыСведений.ДанныеОС.СрезПоследних(РабочаяДата);

Фрагмент процедуры, который выводит реквизиты элемента справочника, будет выглядеть следующим образом:

Выборка = Справочники.ОсновныеСредства.ВыбратьИерархически();

.....

Объект = Выборка.ПолучитьОбъект();

.....

Секция = Макет.ПолучитьОбласть(«Элемент»);

Секция.Параметры.Код = Объект.Код;

Секция.Параметры.Наименование = Объект.Наименование;

Секция.Параметры.Фирма = Объект.Фирма;

Секция.Параметры.ГруппаОС = Объект.ГруппаОС;

Секция.Параметры.ДатаВвода = Объект.ДатаВвода;

Секция.Параметры.ДатаСписания = Объект.ДатаСписания;

// Найдем строку содержащую ОС в таблице значений

СтрокаСреза = ТаблицаСреза.Найти(Объект.Ссылка, «ОС»);

Если СтрокаСреза<>Неопределено Тогда

Секция.Параметры.БалансоваяСтоимость = СтрокаСреза.БалансоваяСтоимость;

Секция.Параметры.РыночнаяСтоимость = СтрокаСреза.РыночнаяСтоимость;

Секция.Параметры.Подразделение = СтрокаСреза.Подразделение;

КонецЕсли;

ТабДок.Вывести(Секция);

Упражнение № 9.30 (продолжение) \_\_\_\_\_

Внесите необходимые изменения в текст процедуры **Печать()** и проведите, если необходимо, ее отладку.

Для надежности приводим доработанный текст процедуры **Печать()** (сравните со своим вариантом).

 ЛИСТИНГ Ю\_9.14

//Печать – печатает записи из справочника «ОсновныеСредства»

// и срез записей из регистра сведений «ДанныеОС»

Процедура Печать(Элемент)

ТабДок = Новый ТабличныйДокумент;

Макет =

Справочники.ОсновныеСредства.ПолучитьМакет(«Печать»);

// Получение среза

```

ТаблицаСреза =
РегистрыСведений.ДанныеОС.СрезПоследних(РабочаяДата);
// Заголовок
Секция = Макет.ПолучитьОбласть(«Заголовок»);
ТабДок.Вывести(Секция);
// Шапка
Секция = Макет.ПолучитьОбласть(«Шапка»);
ТабДок.Вывести(Секция);
Выборка =
Справочники.ОсновныеСредства.ВыбратьИерархически();
Пока Выборка.Следующий() Цикл
Объект = Выборка.ПолучитьОбъект();
Если Объект.ЭтоГруппа Тогда
Секция = Макет.ПолучитьОбласть(«Группа»);
Секция.Параметры.Код = Объект.Код;
Секция.Параметры.Наименование =
Объект.Наименование;
ТабДок.Вывести(Секция);
Иначе
Секция = Макет.ПолучитьОбласть(«Элемент»);
Секция.Параметры.Код = Объект.Код;
Секция.Параметры.Наименование =
Объект.Наименование;
Секция.Параметры.Фирма = Объект.Фирма;
Секция.Параметры.ГруппаОС = Объект.ГруппаОС;
Секция.Параметры.ДатаВвода = Объект.ДатаВвода;
Секция.Параметры.ДатаСписания =
Объект.ДатаСписания;
// Найдем строку содержащую ОС в таблице значений
СтрокаСреза = ТаблицаСреза.Найти(Объект.Ссылка, «ОС»);
Если СтрокаСреза <> Неопределено Тогда
Секция.Параметры.БалансоваяСтоимость =
СтрокаСреза.БалансоваяСтоимость;
Секция.Параметры.РыночнаяСтоимость =
СтрокаСреза.РыночнаяСтоимость;
Секция.Параметры.Подразделение =
СтрокаСреза.Подразделение;
КонецЕсли;
ТабДок.Вывести(Секция);
КонецЕсли;
КонецЦикла;
ТабДок.ОтображатьСетку = Ложь;
ТабДок.Защита = Ложь;

```

ТабДок.ТолькоПросмотр = Истина;  
ТабДок.ОтображатьЗаголовки = Ложь;  
ТабДок.Показать();  
КонецПроцедуры

### Упражнение № 9.31

Добавьте в форму списка справочника «ОсновныеСредства» новую кнопку:

Свойство	Значение
Имя	ПечатьЗапросом
Заголовок	Печать запросом
Нажатие	ПечатьЗапросомНажатие

И расположите ее около кнопки «Печать» (рис. 9.23).

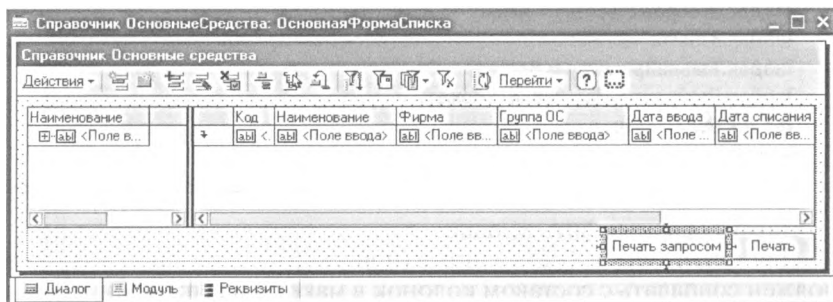


Рис. 9.23. Кнопка «Печать запросом»

## 9.5. СОЕДИНЕНИЕ В ЯЗЫКЕ ЗАПРОСОВ

В предыдущем подразделе при выводе информации в табличный документ мы *соединили* в одном отчете данные из разных таблиц ИБ.

Аналогичные действия можно делать и в языке запросов. Поэтому в этом подразделе мы построим точно такой же отчет, но путем формирования запроса к ИБ.

Сначала выполните подготовительные действия.

### Упражнение № 9.31 (продолжение 1)

Переключитесь на закладку «Модуль» и наберите «заготовку» для написания текста запроса.

## ЛИСТИНГ Ю\_9.15

```
//ПечатьЗапросомНажатие – выполняет те же действия, что и
//процедура Печать(...), но с использованием объекта запрос
Процедура ПечатьЗапросомНажатие(Элемент)
ТабДок = Новый ТабличныйДокумент;
Макет = Справочники.ОсновныеСредства.ПолучитьМакет(«Печать»);
// Заголовок
Секция = Макет.ПолучитьОбласть(«Заголовок»);
ТабДок.Вывести(Секция);
// Шапка
Секция = Макет.ПолучитьОбласть(«Шапка»);
ТабДок.Вывести(Секция);
Запрос = Новый Запрос();
Запрос.Текст =
«»;
// Выполняем запрос
Результат = Запрос.Выполнить();
ТабДок.ОтображатьСетку = Ложь;
ТабДок.Защита = Ложь;
ТабДок.ТолькоПросмотр = Истина;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.Показать();
КонецПроцедуры
```

---

Давайте размышлять дальше.

Для того чтобы выбрать данные (причем состав колонок в запросе должен совпадать с составом колонок в макете) из справочника, мы должны написать следующий текст.

## ЛИСТИНГ Ю\_9.16

```
«ВЫБРАТЬ
| ОсновныеСредства.Ссылка КАК Ссылка,
| ОсновныеСредства.Код КАК Код,
| ОсновныеСредства.Наименование КАК Наименование,
| ОсновныеСредства.Фирма КАК Фирма,
| ОсновныеСредства.ГруппаОС КАК ГруппаОС,
| ОсновныеСредства.ДатаВвода КАК ДатаВвода,
| ОсновныеСредства.ДатаСписания КАК ДатаСписания
|
| ИЗ
| Справочник.ОсновныеСредства КАК ОсновныеСредства
```

Для того чтобы получить срез последних записей из регистра сведений мы должны написать:

## «ВЫБРАТЬ

| ДанныеОС.БалансоваяСтоимость КАК БалансоваяСтоимость,  
| ДанныеОС.РыночнаяСтоимость КАК РыночнаяСтоимость,  
| ДанныеОС.Подразделение КАК Подразделение  
| ИЗ  
| РегистрСведений.ДанныеОС.СрезПоследних(&Период) КАК ДанныеОС

Тут следует обратить внимание на то, что запись **РегистрСведений.ДанныеОС.СрезПоследних(&Период)** представляет собой виртуальную таблицу, а **&Период** — это параметр ее создания.

При создании текста с помощью Конструктора не забывайте нажимать на кнопку «Параметры виртуальной таблицы» (рис. 9.24).

Если соединить два текста в один запрос, то в результате получится «каша» из записей справочника и виртуальной таблицы.

Для того чтобы какой-либо записи из справочника соответствовала какая-нибудь (если есть) запись, полученная из регистра сведений, необходимо установить соединение между этими исходными таблицами.

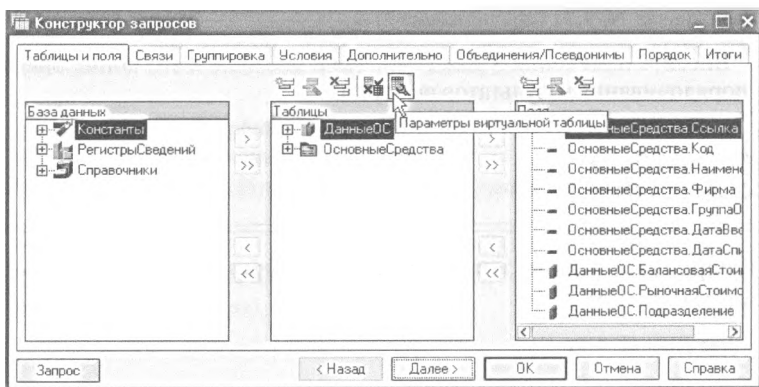


Рис. 9.24. Кнопка настройки виртуальной таблицы

В тексте запроса это будет выглядеть так:

## «ВЫБРАТЬ

| ОсновныеСредства.Ссылка КАК Ссылка,  
| ОсновныеСредства.Код КАК Код,  
| ОсновныеСредства.Наименование КАК Наименование,  
| ОсновныеСредства.Фирма КАК Фирма,  
| ОсновныеСредства.ГруппаОС КАК ГруппаОС,  
| ОсновныеСредства.ДатаВвода КАК ДатаВвода,  
| ОсновныеСредства.ДатаСписания КАК ДатаСписания,

```

| ДанныеОС.БалансоваяСтоимость КАК БалансоваяСтоимость,
| ДанныеОС.РыночнаяСтоимость КАК РыночнаяСтоимость,
| ДанныеОС.Подразделение КАК Подразделение
| ИЗ
| Справочник.ОсновныеСредства КАК ОсновныеСредства
| ЛЕВОЕ СОЕДИНЕНИЕ
| РегистрСведений.ДанныеОС.СрезПоследних(&Период) КАК
| ДанныеОС
| ПО ОсновныеСредства.Ссылка = ДанныеОС.ОС

```

Слово ЛЕВОЕ означает, что *все* записи из справочника попадут в результат. Они будут соединены в отношении «1 ко многим» с записями из виртуальной таблицы.

Условие такого соединения: «ОсновныеСредства.Ссылка = ДанныеОС.ОС».

Для нашего случая виртуальная таблица РегистрСведений.ДанныеОС.СрезПоследних будет для каждого ОС содержать максимально одну запись.

#### Упражнение № 9.31 (продолжение 2)

Внесите изменения в текст запроса и добавьте в его конце оператор упорядочивания результатов выборки:

```

|
| упорядочить по
| Ссылка ИЕРАРХИЯ»;

```

Теперь текст вызова запроса и цикл выборки будут выглядеть следующим образом.

#### ЛИСТИНГ Ю\_9.17

```

// Выполняем запрос
Запрос.УстановитьПараметр(«Период»,РабочаяДата);
Результат = Запрос.Выполнить();
//Откроем выборку
Выборка = Результат.Выбрать();
Пока Выборка.Следующий() Цикл
Если Выборка.Ссылка.ЭтоГруппа Тогда
Секция = Макет.ПолучитьОбласть(«Группа»);
Секция.Параметры.Заполнить(Выборка);
ТабДок.Вывести(Секция);
Иначе
Секция = Макет.ПолучитьОбласть(«Элемент»);
Секция.Параметры.Заполнить(Выборка);

```

**ТабДок.Вывести(Секция);**  
**КонецЕсли;**  
**КонецЦикла;**

Следует обратить внимание на метод **.Заполнить(...)**, который позволяет заполнить параметры области макета табличного документа, сократив объем фрагмента их заполнения до одной строки!

### Упражнение № 9.31 (продолжение 3) ---

Закончите формирование текста процедуры. Проверьте ее работу и сравните результаты печатной формы с предыдущим алгоритмом.

По окончании выполнения упражнения измените версию конфигурации на значение — «Юнит 9».

---

Для надежности приводим доработанный полный текст процедуры **ПечатьЗапросомНажатие** (сравните со своим вариантом).

#### ЛИСТИНГ Ю\_9.18

```
//ПечатьЗапросомНажатие – выполняет те же действия, что и процедура  
// Печать(...), но с использованием объекта запрос
```

**Процедура ПечатьЗапросомНажатие(Элемент)**

**ТабДок = Новый ТабличныйДокумент;**

**Макет = Справочники.ОсновныеСредства.ПолучитьМакет(«Печать»);**

**// Заголовок**

**Секция = Макет.ПолучитьОбласть(«Заголовок»);**

**ТабДок.Вывести(Секция);**

**// Шапка**

**Секция = Макет.ПолучитьОбласть(«Шапка»);**

**ТабДок.Вывести(Секция);**

**Запрос = Новый Запрос();**

**Запрос.Текст =**

**«ВЫБРАТЬ**

**{ ОсновныеСредства.Ссылка КАК Ссылка,**

**| ОсновныеСредства.Код КАК Код,**

**| ОсновныеСредства.Наименование КАК Наименование,**

**| ОсновныеСредства.Фирма КАК Фирма,**

**| ОсновныеСредства.ГруппаОС КАК ГруппаОС,**

**| ОсновныеСредства.ДатаВвода КАК ДатаВвода,**

**| ОсновныеСредства.ДатаСписания КАК ДатаСписания,**

```
| ДанныеОС.БалансоваяСтоимость КАК БалансоваяСтоимость,  
| ДанныеОС.РыночнаяСтоимость КАК РыночнаяСтоимость,  
| ДанныеОС.Подразделение КАК Подразделение  
| ИЗ  
| Справочник.ОсновныеСредства КАК ОсновныеСредства  
| ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.ДанныеОС.СрезПоследних(&Период)  
КАК ДанныеОС  
| ПО ОсновныеСредства.Ссылка = ДанныеОС.ОС  
|  
| УПОРЯДОЧИТЬ ПО  
| Ссылка ИЕРАРХИЯ»;
```

```
// Выполняем запрос
```

```
Запрос.УстановитьПараметр(«Период»,РабочаяДата);  
Результат = Запрос.Выполнить();
```

```
//Откроем выборку
```

```
Выборка = Результат.Выбрать();  
Пока Выборка.Следующий() Цикл
```

```
Если Выборка.Ссылка.ЭтоГруппа Тогда
```

```
Секция = Макет.ПолучитьОбласть(«Группа»);
```

```
Секция.Параметры.Заполнить(Выборка);
```

```
ТабДок.Вывести(Секция);
```

```
Иначе
```

```
Секция = Макет.ПолучитьОбласть(«Элемент»);
```

```
Секция.Параметры.Заполнить(Выборка);
```

```
ТабДок.Вывести(Секция);
```

```
КонецЕсли;
```

```
КонецЦикла;
```

```
ТабДок.ОтображатьСетку = Ложь;
```

```
ТабДок.Защита = Ложь;
```

```
ТабДок.ТолькоПросмотр = Истина;
```

```
ТабДок.ОтображатьЗаголовки = Ложь;
```

```
ТабДок.Показать();
```

```
КонецПроцедуры
```

## 9.6. ЧТО МЫ УЗНАЛИ

В этой главе мы рассмотрели совместное применение в конфигурации справочников и регистров сведений.



В качестве примера мы добавили в конфигурацию периодический регистр сведений «**ДанныеОС**», в котором разместили реквизиты, имеющие отношение к основному средству. Этот регистр имеет два измерения — «Период» и «ОС».

В этой главе мы закончили проектирование справочника «**ОсновныеСредства**» рассмотрев более подробно разработку формы элемента, где уделили внимание обработке событий, как от элементов формы, так и от самой формы в целом.

Здесь же мы рассмотрели особенности применения добавочного ключевого слова «Экспорт» при расположении процедуры (или функции) в модуле объекта. Использование такой процедуры позволяет применить ее в качестве метода объекта, определяемого пользователем.

В этой же главе мы рассмотрели особенности применения метода **СрезПоследних(...)** регистра сведений.

В заключение этой главы мы рассмотрели применение в языке запросов виртуальной таблицы «**СрезПоследних**», а также особенности соединения двух таблиц для получения необходимых данных.

Система «1С:Предприятие 8» использует концепцию работы «от документа». Это значит, что документ служит не столько для формирования печатной формы некоторого первичного «бумажного» документа, сколько для отражения в системе некоторого акта хозяйственной деятельности, произошедшего на предприятии.

### 10.1. ОСНОВНЫЕ ПОНЯТИЯ

В этом разделе мы рассмотрим принципы, положенные в основу системы для работы с документами.

Сначала отметим, что каждый сохраненный в ИБ документ имеет некоторую позицию на оси времени. Эта позиция: тип данных **МоментВремени**, который содержит *дату*, включая *время*, и ссылку на объект ИБ. Именно таким образом система определяет взаимный порядок расположения двух документов на оси времени.

Время в реквизите «Дата» может быть использовано не только для отражения реального (астрономического) времени происшедшего акта хозяйственной деятельности, но и для указания взаимной хронологии двух событий, если реальное время совершения хозяйственных актов пользователю неизвестно. Тогда пользователь, явно указывая в реквизите «Дата» конкретное время, только относит (регистрирует) акт хозяйственной деятельности, который реально произошел, происходит или еще будет происходить на предприятии.

Очень важным является понятие о *проведении* документа.

Если документ проведен, то это означает, что его содержимое отражено в *регистрах* конфигурации.

При этом проведенный документ может иметь несколько связанных с ним учетных записей, которые документация к программе называет *Движениями* (рис. 10.1).

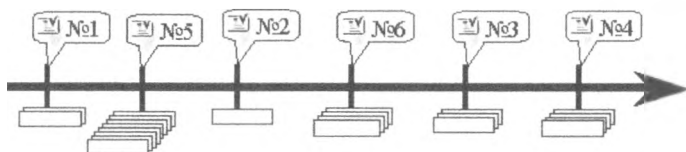


Рис. 10.1. Расположение документов на воображаемой оси времени

Движения очень тесно связаны с документом. Так, например, для большинства видов документов в конфигурации, если произвести его исправление с последующим повторным проведением (уже проведенного документа), то система удалит предварительно сформированные старые движения и на их место разместит новые.

Такой способ поведения документа система устанавливает по умолчанию для новых видов документов. И этот способ гарантирует, что либо документ будет проведен, либо состояние ИБ вернется в момент до его проведения.

Впрочем, возможна такая настройка документа, что при его проведении Вы сами будете решать: какие движения (записи) сохранить, какие добавить, а какие из уже имеющихся движений удалить.

В процессе проведения документа система открывает не явным образом *транзакцию*. Такой подход гарантирует, что либо документ будет проведен, либо состояние системы вернется в момент до его проведения, что обеспечивает целостность вносимых в объекты ИБ изменений.

Проведение документа одним пользователем *не блокирует* проведение другого документа другим пользователем. Поэтому два и более пользователя могут работать с документами практически независимо друг от друга.

Как следствие, два и более документа могут иметь одну и ту же дату (и время), и тем не менее у них моменты времени будут различны.

Их взаимное расположение (один раньше — другой позже) можно определить с использованием методов типа данных «МоментВремени».

Все документы, как уже было сказано, образуют единую хронологическую последовательность, поэтому в режиме «1С:Предприятие» существует возможность их группового проведения. Такая возможность имеется в системе из-за того, что запись документа и его проведение могут выполняться как отдельные этапы работы с ним в соответствии с порядком заполнения данных, который определяется конфигурацией. В силу этой возможности любой документ (если это не запрещено на этапе конфигурирования) может быть изменен и проведен «задним числом».

Но что должно произойти с документами, которые расположены позже по оси времени? Ответ на этот вопрос зависит от характера прикладной задачи.

Если нет зависимости между документами, то ничего делать не нужно, так как такое проведение задним числом следует рассматривать как отражение акта хозяйственной деятельности, который уже произошел на предприятии.

Если проведение одного документа зависит от данных, формируемых при проведении других документов, то в этом случае можно восстановить хронологию обработки актов хозяйственной деятельности путем повторного проведения необходимых документов.

Идеальной можно считать ситуацию, когда все документы проводятся последовательно друг за другом. В этом случае можно на этапе конфигурирования включить механизм *оперативного проведения*, который позволяет при проведении документа текущей датой устанавливать его время так, чтобы он всегда располагался последним документом в хронологической последовательности.

Если для документа включено оперативное проведение и он вводится в систему задним числом, то при его проведении можно определить *режим проведения* и тем самым оптимизировать время обработки. Заметим, что обработка проведения должна выполняться максимально быстро, так как при большом времени ее работы пользователь будет каждый раз ожидать ее завершения.

В последующих разделах мы подробно рассмотрим конфигурирование документов на основе постановки задачи.

## 10.2. ИЗМЕНЕНИЕ НАСТРОЙКИ РЕГИСТРА СВЕДЕНИЙ «ДАННЫЕОС»

На данном этапе создания конфигурации регистр сведений «ДанныеОС» пользователь заполняет вручную. Для того чтобы в этот регистр документы могли бы помещать движения, необходимо в его настройках изменить режим записи.

### Упражнение № 10.1

Установите для регистра сведений «ДанныеОС» следующее свойство:

Свойство	Значение
Режим записи	<b>Подчинение регистратору</b>

Такие действия приведут к тому, что в регистре появится специальное поле (свойство) — Регистратор. Оно будет содержать ссылку на документ, которому принадлежит данная запись, и редактирование отдельных записей в форме списка или набора записей станет невозможным.

Если мы указываем для регистра подчинение регистратору, то это будет означать, что он будет содержать *движения документов*. При этом нужно помнить, что на специалисте, осуществляющем конфигурирование, лежит ответственность за создание достаточного коли-

чества видов документов, для того чтобы содержимое регистра полностью формировалось автоматически при проведении документов.

Впрочем, система предоставляет возможность и ручного редактирования движений в форме документа, но в этом случае ответственность за полноту и целостность содержимого регистров лежит как на пользователе, который выполнил изменения вручную, так и на специалисте, предоставившем пользователю такую возможность.

Отметим, что этот регистр можно было бы сделать и непериодическим, так как необходимая дата и время имеются в документе, но в этом случае у нас будет нарушена работа объектов, построенных на предыдущих этапах работы.

Поскольку мы изменили структуру регистра, то необходимо внести небольшие изменения и в его форму.

### 10.2.1. Изменение формы списка

Для того чтобы наш пользователь мог редактировать содержимое регистра сведений в полном объеме, необходимо вставить в его форму списка колонку «Регистратор».

#### Упражнение № 10.2

Увеличьте ширину формы списка регистра сведений «ДанныеОС» до величины 600. После чего вставьте в табличное поле, используя команду «Форма — Размещение данных...», колонку «Регистратор» и расположите ее в форме согласно рис. 10.2.

Сохраните и обновите конфигурацию.

При завершении этого упражнения Конфигуратор при выполнении операции обновления выдаст сообщение об ошибке:

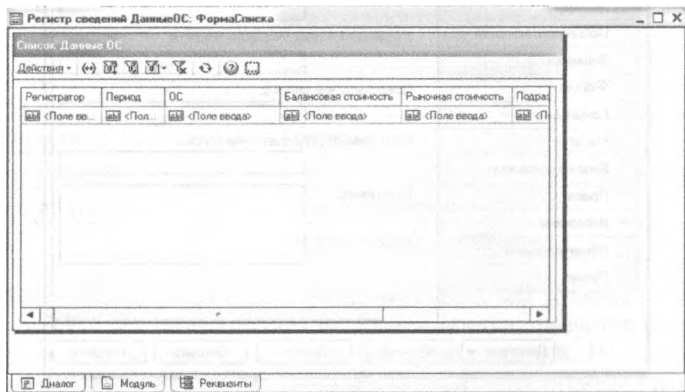


Рис. 10.2. Основная форма списка регистра сведений

**РегистрСведений.Данные0С:** Ни один из документов не является регистратором для регистра.

В данном случае это нормально, поскольку теперь мы должны добавить в нашу конфигурацию хотя бы один вид документов, который будет формировать движения по этому регистру сведений.

### 10.3. ДОКУМЕНТ «ПОСТУПЛЕНИЕ»

В учебных целях мы добавим в нашу конфигурацию документ «Поступление». Он будет формировать движения в регистре сведений.

Создание новых и редактирование существующих видов документов в конфигурации удобно производить через окно редактирования.

Структура окна и методика работы с ним нам уже знакомы по таким типам данных, как «Регистры сведений» и «Справочники». Поэтому в этом разделе мы напомним назначение уже известных вам закладок и рассмотрим специфику настройки документов.

#### Упражнение № 10.3

Используя правую кнопку мыши на ветви дерева «Документы» окна «Конфигурация», выберите пункт меню «Добавить». Таким образом, Вы открыли окно редактирования нового документа (рис. 10.3).

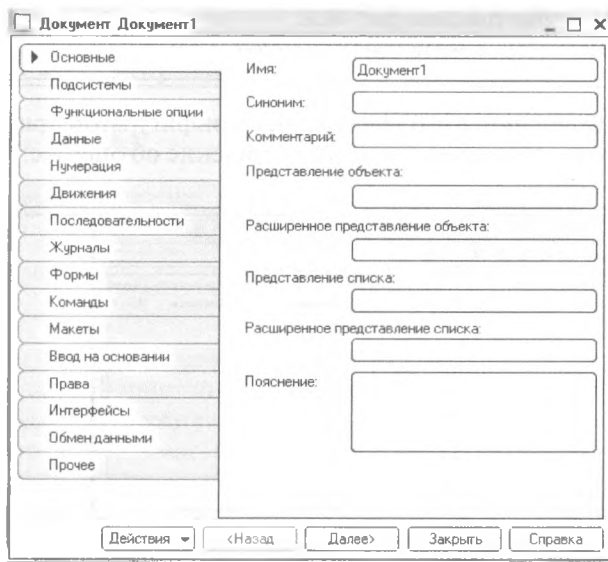


Рис. 10.3. Окно редактирования нового документа

### 10.3.1. Закладка «Основные»

Эта закладка позволяет заполнить основные свойства документа «Имя», «Синоним» и «Комментарий». Эти свойства вам хорошо знакомы и не нуждаются в каких-либо пояснениях.

В соответствии с п. 9 постановки задачи мы создадим в конфигурации новый вид документа под именем «Поступление».

#### Упражнение № 10.3 (продолжение)

Включите в конфигурацию новый документ со следующими основными свойствами:

Свойство	Значение
Имя	Поступление
Синоним	Поступление
Комментарий	Поступление ОС и деталей к ним

### 10.3.2. Закладка «Данные»

Документ имеет два реквизита с заранее заданным назначением — это «Номер» и «Дата». Все остальные реквизиты необходимо будет добавить к свойствам объекта, используя эту закладку.

Переключимся на эту закладку и прежде, чем что-то вводить, задумаемся над тем, какие реквизиты нам нужны.

Прежде всего вспомним, что согласно постановке задачи у нас в одной ИБ должны присутствовать документы нескольких *фирм*, образующих объединение. Поэтому нам понадобится реквизит «Фирма».

#### Упражнение № 10.4

Добавьте в шапку документа следующий реквизит:

Свойство	Значение
Имя	Фирма
Синоним	Фирма
Комментарий	—
Индексировать	Индексировать
Тип	СправочникСсылка.Фирмы

Далее. По условию задачи рублевые суммы мы должны пересчитывать в УЕ, следовательно, для выполнения таких расчетов нам нужно будет получать значение курса из регистра сведений. Но выполнение такой операции при редактировании формы будет отнимать некоторое время.

Поэтому разумнее будет иметь реквизит «Курс» в самом документе и заполнять его программным путем всего один раз, предоставив при этом возможность пользователю самостоятельно выбирать дату курса.

Добавьте в шапку документа следующие реквизиты:

Свойство	Значение
Имя	<b>ДатаКурса</b>
Синоним	<b>Дата курса</b>
Комментарий	—

Свойство	Значение
Имя	<b>Курс</b>
Синоним	<b>Курс</b>
Комментарий	—

Какие типы данных вы назначите для новых реквизитов шапки?

На этом реквизиты шапки, очевидно, закончатся, а содержимое табличной части фактически представлено в постановке задачи.

Добавьте в документ табличную часть:

Свойство	Значение
Имя	<b>Список</b>
Синоним	<b>Список</b>
Комментарий	<b>Список ОС и деталей</b>

Теперь мы должны будем наполнить табличную часть реквизитами согласно постановке задачи.

Колонку, которая может содержать как ОС, так и деталь, мы назовем «ОС\_Деталь».

Добавьте в табличную часть реквизит:

Свойство	Значение
Имя	<b>ОС_Деталь</b>
Синоним	<b>ОС/Деталь</b>
Комментарий	<b>ОС или Деталь</b>
Индексировать	<b>Индексировать</b>
Тип	<b>СправочникСсылка.ОсновныеСредства, СправочникСсылка.Детали</b>

*Совет:* при выборе типа данных используйте кнопку выбора и не забудьте в диалоге «Выбор типа» установить флаг «Составной тип данных».

Теперь нам предстоит добавить в табличную часть реквизиты: **Количество**, **ЦенаРуб.**, **СуммаРуб.**, **ЦенаУЕ** и **СуммаУЕ**.



Однако часть из таких реквизитов у нас уже была в справочнике «ОсновныеСредства». Поэтому, используя команды работы с буфером обмена, скопируйте реквизиты табличной части «СписокДеталей» справочника «ОсновныеСредства» в табличную часть «Список». Измените, если нужно, общие свойства этих реквизитов.

Если вы правильно выполнили последние действия, то закладка «Данные» нашего документа будет выглядеть так, как это изображено на рис. 10.4.

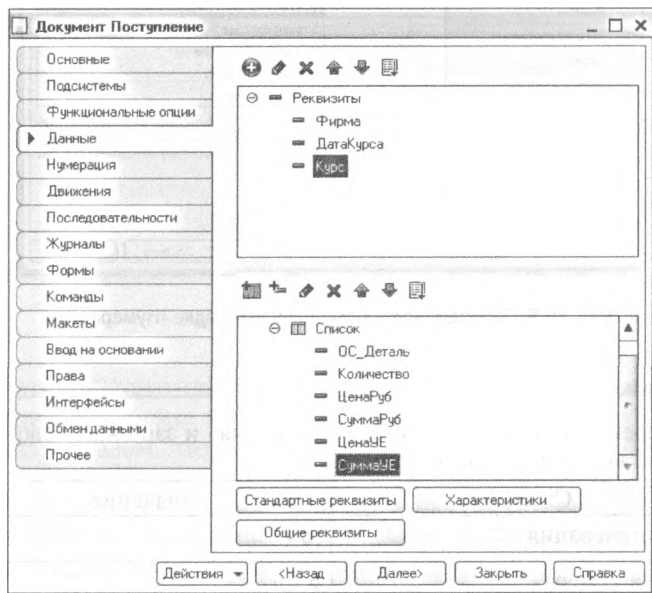


Рис. 10.4. Закладка «Данные» после выполнения упражнения

### 10.3.3. Закладка «Нумерация»

На закладке «Нумерация» мы можем установить тип и длину номера документа подобно тому, как мы настраивали тип и длину кода в справочнике.

Дополнительно, используя эту закладку, можно установить периодичность номеров (рис. 10.5).

Поскольку в нашей информационной базе будут присутствовать документы от разных фирм, то мы, как и в случае со справочником «ОсновныеСредства», будем использовать префикс кода.

Для удобства пользователя мы включим периодичность номеров.

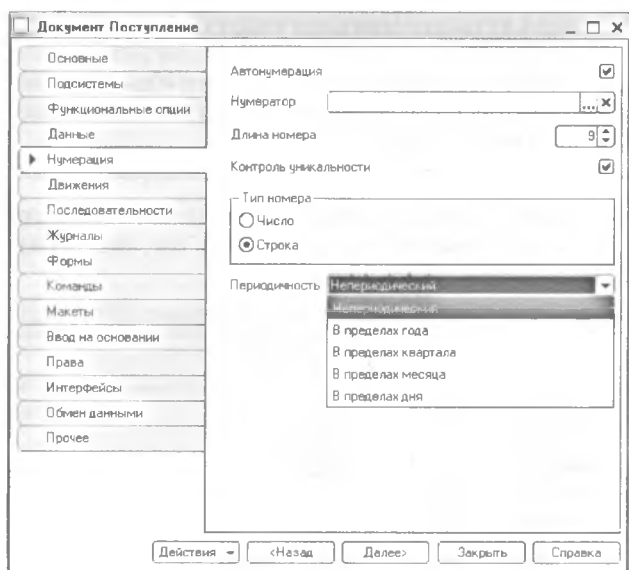


Рис. 10.5. Периодичность номеров на закладке «Нумерация»

### Упражнение № 10.5

Переключитесь на закладку «Нумерация» и заполните свойства документа следующими значениями:

Свойство	Значение
Автонумерация	<input checked="" type="checkbox"/>
Длина номера	8
Контроль уникальности	<input checked="" type="checkbox"/>
Тип номера	Строка
Периодичность	В пределах года

### 10.3.4. Закладка «Движения»

На этой закладке можно настроить проведение документа (рис. 10.6).

Свойство «Проведение» может принимать два значения: «Разрешить» и «Запретить». Если установлено первое значение — документ может быть проведен, если установлено второе значение — нет. При этом в обоих случаях документ может иметь печатную форму, настройка которой выполняется на закладке «Макеты».

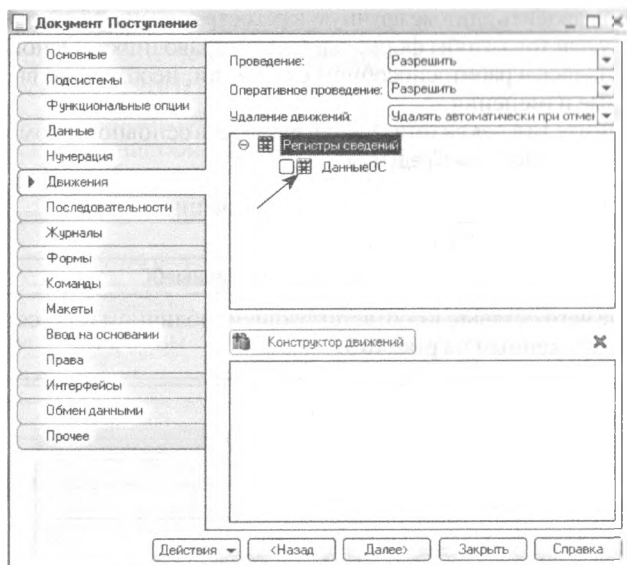


Рис. 10.6. Закладка «Движения»

Свойство «Оперативное проведение» указывает системе, что при проведении документа текущей датой его время можно устанавливать таким образом, чтобы он занял последнее положение в хронологической последовательности документов.

Свойство «Удаление движений» по умолчанию установлено в положение «Удалять движения автоматически». Это приводит к тому, что, когда пользователь тем или иным способом дает команду на повторное проведение документа, его движения сначала удаляются. Установка этого свойства в положение «Не удалять движения автоматически» даст возможность программисту самостоятельно решать, какие из движений уже проведенного документа нужно удалить, какие добавить, а какие оставить без изменения при проведении документа.

Изображенный выше список регистров позволяет отметить те регистры, в которые документу *разрешено* записывать движения.

### Упражнение № 10.6

Установите на закладке «Движения» флаг для регистра «ДанныеОС». Сохраните и обновите конфигурацию.

Обратите внимание, что записи в регистре сведений конфигуратор удалил!

Теперь вводить данные вручную в регистр сведений будет уже нельзя. Поэтому для того чтобы форма элемента справочника «ОсновныеСредства» осталась в работоспособном состоянии, необходимо внести в нее некоторые изменения.

Измените тип реквизита «НаборДанныеОС» в основной форме элемента справочника «ОсновныеСредства»:

Свойство	Значение
Имя	НаборДанныеОС
Тип	РегистрСведенийСписок.ДанныеОС

После чего оставьте в соответствующем табличном поле состав колонок, изображенный на рис. 10.7.

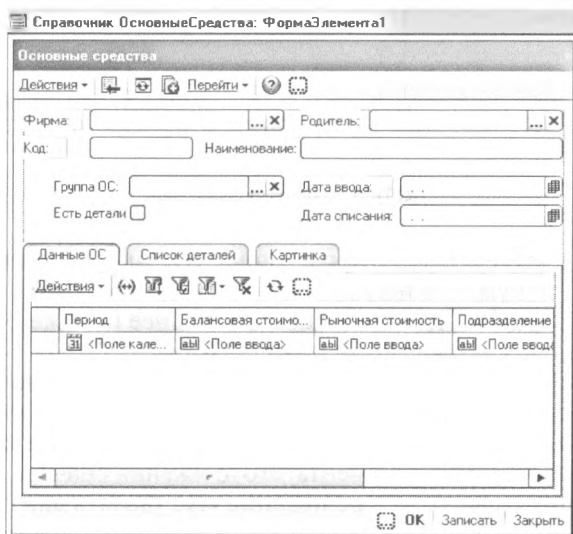


Рис. 10.7. Форма элемента справочника после изменения типа одного из реквизитов

### 10.3.5. Закладка «Последовательности»

Позволяет при наличии *последовательностей* в конфигурации более «тонко» настроить систему на восстановление хронологической последовательности документов.

### 10.3.6. Закладка «Журналы»

При наличии журналов в конфигурации позволяет указать журналы, в которых будут отображаться документы данного вида в режиме «1С:Предприятие».

### 10.3.7. Самостоятельная работа: закладка «Формы»

Какие формы нам будут нужны? Очевидно, что нам будет нужна *форма списка*, для того чтобы видеть список уже ранее введенных документов, и *форма самого документа*.

Теперь нам понадобится создать форму документа.

#### Упражнение № 10.7

Добавьте новую форму со следующими свойствами:

Свойство	Значение
Выберите тип формы	<b>Форма документа</b>
Назначить форму основной	<input checked="" type="checkbox"/>
Имя	<b>ОсновнаяФормаДокумента</b>
Синоним	<b>Основная форма документа</b>
Комментарий	<b>Поступление ОС и деталей</b>
Командная панель сверху	<input checked="" type="checkbox"/>
Командная панель снизу	<input checked="" type="checkbox"/>

Нажмите на кнопку **Далее>**. Поскольку наша форма будет достаточно сложная, то сбросьте на этом этапе все флажки.

Используя пункт меню «Форма — Размещение данных...», вставьте и расположите элементы формы так, как это указано на рис. 10.8.

Сбросьте в свойствах формы флаг «Автопорядок обхода» и, используя команду «Форма — Порядок обхода», расставьте порядок обхода согласно рис. 10.9.

N	ОС/Деталь	Количество	Цена руб	Сумма руб
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Рис. 10.8. Форма документа «Поступление»

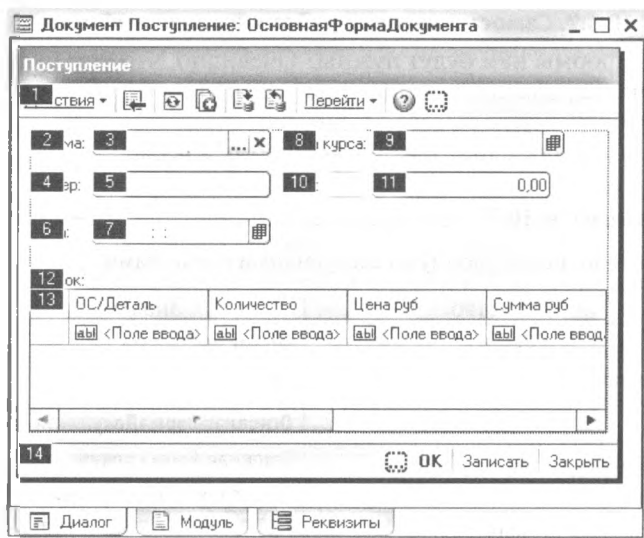


Рис. 10.9. Порядок обхода элементов в форме документа

### 10.3.8. Закладка «Интерфейсы»

Вставьте в основной интерфейс команду открытия формы списка документов «Поступление», выполнив следующее упражнение.

#### Упражнение № 10.8

На закладке «Интерфейсы» установите флажок для команды «Поступление: Открыть список документов» для объекта «ОсновнойИнтерфейс».

Проверьте работу интерфейса и, если необходимо, внесите изменения в свойства нового пункта меню.

## 10.4. РАБОТА С ДОКУМЕНТАМИ ИЗ ВСТРОЕННОГО ЯЗЫКА СИСТЕМЫ

Работа с документами на встроенном языке во многом напоминает работу со справочниками. В самом деле: для того чтобы обратиться к документам из встроенного языка системы, необходимо воспользоваться свойством **Документы** глобального контекста. Это свойство имеет тип **ДокументыМенеджер**.

В этом смысле работа со многими объектами в языке *унифицирована*.

### 10.4.1. Менеджер документа

По аналогии со справочниками: чтобы обратиться к конкретному виду документов, необходимо его имя записать «через точку», например: **Документы.Поступление**.

Такая запись имеет тип **ДокументМенеджер.<Имя документа>** — это менеджер конкретного вида документов. Объект этого типа имеет следующие группы методов:

- ✓ для организации выборки;
- ✓ для поиска отдельных записей;
- ✓ для получения макета;
- ✓ для получения формы справочника;
- ✓ для создания отдельных записей.

Для того чтобы лучше представили себе, что работа со справочниками и документами имеет много общего, приведем список методов для справочника и для документа:

СправочникМенеджер	ДокументМенеджер
Выбрать(...)	Выбрать(...)
ВыбратьИерархически(...)	—
НайтиПоКоду(...)	НайтиПоНомеру()
НайтиПоНаименованию(...)	—
НайтиПоРеквизиту(...)	НайтиПоРеквизиту(,)
ПолучитьМакет()	ПолучитьМакет()
ПолучитьФорму(...)	ПолучитьФорму(...)
ПолучитьФормуВыбора(...)	ПолучитьФормуВыбора(...)
ПолучитьФормуВыбораГруппы(...)	—
ПолучитьФормуНовогоЭлемента(...)	ПолучитьФормуНовогоДокумента(...)
ПолучитьФормуНовойГруппы(...)	—
ПолучитьФормуСписка(...)	ПолучитьФормуСписка(...)
ПустаяСсылка()	ПустаяСсылка()
СоздатьГруппу()	—
СоздатьЭлемент()	СоздатьДокумент()

Как мы видим, общего в этом списке гораздо больше, чем различий.

### 10.4.2. Самостоятельная работа: организация выборки

Этот раздел вы изучите самостоятельно.

## Упражнение № 10.9

---

Переключитесь на закладку «Макеты». Вызовите Конструктор печати и добавьте с его помощью кнопку «Печать» в основную форму документа «Поступление».

Какой метод использовал конструктор для организации выборки?

---

### 10.4.3. Работа с Документом как с объектом

У объектов типа **ДокументСсылка** и **ДокументВыборка** есть метод **ПолучитьОбъект()**, который позволяет получить доступ к объекту типа **ДокументОбъект.<Имя документа>**.

В этом и последующих разделах мы рассмотрим особенности объекта этого типа (по сравнению со справочниками).

Свойство **Проведен** позволяет определить проведен ли документ.

Метод **Записать(...)** имеет два параметра, которые определяют режим записи и режим проведения документа. Заметим, что провести можно только записанный документ:

```
НовыйДокумент=Документы.Имя.СоздатьДокумент();
НовыйДокумент.Дата = ... ;
НовыйДокумент.Номер = ... ;
НовыйДокумент.... = ... ;
.....
// Записываем и проводим документ в реальном времени
НовыйДокумент.Записать(
РежимЗаписиДокумента.Проведение,
РежимПроведенияДокумента.Оперативный);
Если Не НовыйДокумент.Проведен Тогда
Сообщить(«Об ошибке проведения»);
КонецЕсли;
```

Более подробно работу со свойствами, методами и событиями мы разберем ниже.

## 10.5. ФОРМА ДОКУМЕНТА

В этом разделе мы выполним настройку формы документа и начнем с обработки событий от элементов формы.

### 10.5.1. Обработка событий от элементов формы

В этой форме мы будем использовать в основном событие «При изменении». Поэтому выполните следующее упражнение.

## Упражнение № 10.10

---

Установите свойство «При изменении» для следующих элементов формы:



Имя	При изменении
Фирма	ФирмаПриИзменении
Дата	ДатаПриИзменении
ДатаКурса	ДатаКурсаПриИзменении
Курс	КурсПриИзменении

Вставьте для каждой процедуры текст обработчика события согласно следующему назначению:

```
// Обработчики событий от элементов формы
// ФирмаПриИзменении(Элемент) – устанавливает новый номер
// документа согласно префиксу фирмы
Процедура ФирмаПриИзменении(Элемент)
// Вставить содержимое обработчика.
КонецПроцедуры
// ДатаПриИзменении(Элемент) – изменяет дату курса, курс и
// пересчитывает содержимое колонок «ЦенаУЕ» и «СуммаУЕ» по
// новому курсу
Процедура ДатаПриИзменении(Элемент)
// Вставить содержимое обработчика.
КонецПроцедуры
// ДатаКурсаПриИзменении(Элемент) – изменяет курс и
// пересчитывает содержимое колонок «ЦенаУЕ» и «СуммаУЕ» по
// новому курсу
Процедура ДатаКурсаПриИзменении(Элемент)
// Вставить содержимое обработчика.
КонецПроцедуры
// КурсПриИзменении(Элемент) – пересчитывает содержимое
// колонок «ЦенаУЕ» и «СуммаУЕ» по новому курсу
Процедура КурсПриИзменении(Элемент)
// Вставить содержимое обработчика.
КонецПроцедуры
```

*Совет:* при написании текстов процедур можно ориентироваться на текст модуля формы элемента справочника «ОсновныеСредства».

Для надежности приводим полный текст процедур (сравните со своим вариантом).

#### ЛИСТИНГ Ю\_10.1

```
// Модуль документа «Поступление»
// Обработчики событий от элементов формы
// ФирмаПриИзменении(Элемент) – устанавливает новый номер
```

```
// документа согласно префиксу фирмы
Процедура ФирмаПриИзменении(Элемент)
УстановитьНовыйНомер(Фирма.Префикс+«-»);
КонецПроцедуры
```

```
// ДатаПриИзменении(Элемент) – изменяет дату курса, курс и
// пересчитывает содержимое колонок «ЦенаУЕ» и «СуммаУЕ» по
// новому курсу
Процедура ДатаПриИзменении(Элемент)
ДатаКурса=Дата;
Строка=РегистрыСведений.Курсы.ПолучитьПоследнее(ДатаКурса);
Курс=Строка.Курс;
ПересчетСтроки();
КонецПроцедуры
```

```
// ДатаКурсаПриИзменении(Элемент) – изменяет курс и
// пересчитывает содержимое колонок «ЦенаУЕ» и «СуммаУЕ» по
```

```
// новому курсу
Процедура ДатаКурсаПриИзменении(Элемент)
Строка=РегистрыСведений.Курсы.ПолучитьПоследнее(ДатаКурса);
Курс=Строка.Курс;
ПересчетСтроки();
КонецПроцедуры
```

```
// КурсПриИзменении(Элемент) – пересчитывает содержимое
// колонок «ЦенаУЕ» и «СуммаУЕ» по новому курсу
Процедура КурсПриИзменении(Элемент)
ПересчетСтроки();
КонецПроцедуры
```

```
Процедура ПересчетСтроки()
ТекСтрока=ЭлементыФормы.Список.ТекущаяСтрока;
Для Каждого ТекСтрока Из ЭлементыФормы.Список.Значение Цикл
ТекСтрока.ЦенаУЕ = ТекСтрока.ЦенаРуб/Курс;
ТекСтрока.СуммаУЕ = ТекСтрока.ЦенаУЕ*ТекСтрока.Количество;
КонецЦикла
КонецПроцедур
```

Перейдем теперь к обработке событий полей ввода табличного поля. Для сокращения объема кодирования мы выполним только пересчет из рублей в УЕ и пересчет рублевой суммы.

## Упражнение № 10.11

Сбросьте флаг «Редактирование текста» для полей ввода «ЦенаУЕ» и «СуммаУЕ».

Установите для полей ввода свойство «При изменении»:

Имя	При изменении
Количество	ПересчетСтроки
ЦенаРуб	ПересчетСтроки
СуммаРуб	ПересчетСтроки

Реализуйте процедуру ПересчетСтроки(...) самостоятельно.

Для надежности еще раз приводим полный текст процедуры **ПересчетСтроки** (сравните со своим вариантом).

### ЛИСТИНГ Ю\_10.2

**Процедура ПересчетСтроки()**

**ТекСтрока=ЭлементыФормы.Список.ТекущаяСтрока;**

**Для Каждого ТекСтрока Из ЭлементыФормы.Список.Значение Цикл**

**ТекСтрока.СуммаРуб=ТекСтрока.ЦенаРуб\*ТекСтрока.Количество;**

**ТекСтрока.ЦенаУЕ = ТекСтрока.ЦенаРуб/Курс;**

**ТекСтрока.СуммаУЕ = ТекСтрока.ЦенаУЕ\*ТекСтрока.Количество;**

**КонецЦикла**

**КонецПроцедуры**

### 10.5.2. Получение итогов в подвале табличного поля

Во время заполнения формы пользователь захочет иметь итоги по колонкам «Количество», «СуммаРуб» и «СуммаУЕ».

В этом разделе мы рассмотрим применение метода **Итог(...)**, который есть как у объекта «Табличное поле», так и у объекта типа «Табличная часть». Уточним, что первый из этих объектов относится к *интерфейсным*, а второй — к *прикладным объектам*.

Среди свойств объекта «Табличное поле» есть свойство «Подвал», установка которого приведет к появлению в табличном поле строки подвала.

## Упражнение № 10.12

Установите для элемента формы «Список» свойство «Подвал»:

Свойство	Значение
Подвал	<input checked="" type="checkbox"/>

Теперь для каждой отдельной колонки можно указать, «что» должно отображаться в подвале.

## Упражнение № 10.12 (продолжение)

Произведите настройку свойств для колонок табличного поля в соответствии со следующей таблицей:

Имя	Текст подвала	Горизонт. положение в подвале	Отображать итоги в подвале
ОС_Деталь	Итого	Прижать вправо	—
Количество	—	Прижать вправо	<input checked="" type="checkbox"/>
СуммаРуб	—	Прижать вправо	<input checked="" type="checkbox"/>
СуммаУЕ	—	Прижать вправо	<input checked="" type="checkbox"/>

Проверьте работу формы в режиме «1С:Предприятие».

### 10.5.3. Обработка событий формы

К этому моменту у вас уже достаточно опыта, поэтому следующее упражнение предназначено для закрепления материала из предыдущей главы.

## Упражнение № 10.13

Напишите обработку события «При открытии» и проверьте работу формы в режиме «1С:Предприятие».

Для надежности приводим текст процедуры «ПриОткрытии» (сравните со своим вариантом).

### ЛИСТИНГ Ю\_10.3

```
Процедура ПриОткрытии(Элемент)
Если ЭтоНовый() Тогда
Фирма = Константы.ОсновнаяФирма.Получить();
УстановитьНовыйНомер(Фирма.Префикс+«-»);
Строка=РегистрыСведений.Курсы.ПолучитьПоследнее(ДатаКурса);
Курс=Строка.Курс;
ДатаКурса=Дата;
КонецЕсли
КонецПроцедуры
```

Мы можем отобразить в форме некоторую *информационную строку* в виде объекта типа «Надпись».

## Упражнение № 10.14

Увеличьте высоту формы и вставьте в форму элемент управления типа «Надпись»:

Свойство	Значение
Имя	<b>ИнформационнаяСтрока</b>
Заголовок	—
Горизонт. положение	<b>Прижать влево</b>

Расположите этот элемент согласно рис. 10.10.

Заполнять эту строку мы будем при обработке события «Обновление отображения», поэтому включите его обработку в свойствах формы:

Свойство	Значение
Обновление отображения	<b>ОбновлениеОтображения</b>

Теперь мы можем заполнять заголовок этой надписи при каждом обновлении отображения в форме, реализовав следующий программный код.

### ЛИСТИНГ Ю\_10.4

```
//ОбновлениеОтображения() – отображает итоговую строку в
// «подвале» формы
Процедура ОбновлениеОтображения()
```

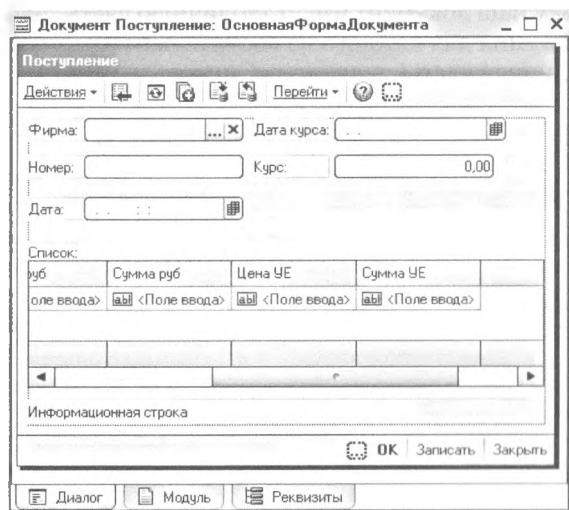


Рис. 10.10. Информационная строка в «подвале» формы

ЭлементыФормы.ИнформационнаяСтрока.Заголовок =  
«Итого: поступило средств на сумму» +  
Список.Итого(«СуммаРуб») + « руб. » + «(» +  
Список.Итого(«СуммаУЕ») + « УЕ»);  
КонецПроцедуры

## Упражнение № 10.15

Наберите текст обработчика **ОбновлениеОтображения()** и проверьте его работу в режиме «1С:Предприятие».

Важно: в свойствах объекта на вкладке «События» необходимо определить процедуру **ОбновлениеОтображения** и нажать значок лупы.

## 10.6. ОБРАБОТКА ПРОВЕДЕНИЯ ДОКУМЕНТА

Последнее, что осталось нам сделать с этим документом, — «научить» его формировать движения в регистре сведений.

К счастью, в системе имеется специальный Конструктор движения, применение которого мы рассмотрим ниже.

### 10.6.1. Конструктор движений

Для того чтобы вызвать этот конструктор, необходимо на закладке «Движения» окна редактирования документа нажать одноименную кнопку. После этого Конфигуратор откроет окно конструктора (рис. 10.11).

Поскольку наш документ имеет табличную часть, то, прежде чем писать выражения для каждого из свойств регистра, нужно выбрать в поле «Табличная часть» значение «Список».

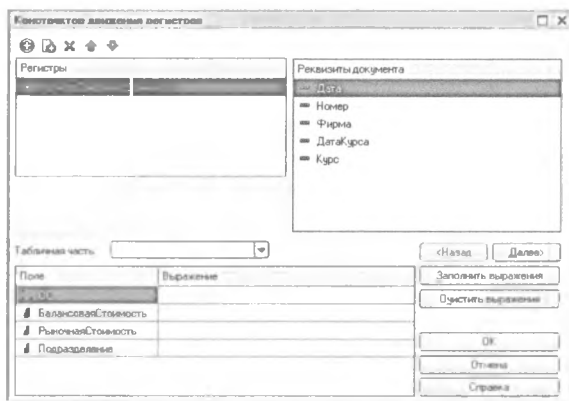


Рис. 10.11. Окно Конструктора движения

## Упражнение № 10.16

Для табличной части «Список», используя мышь, заполните окно Конструктора согласно рис. 10.12.

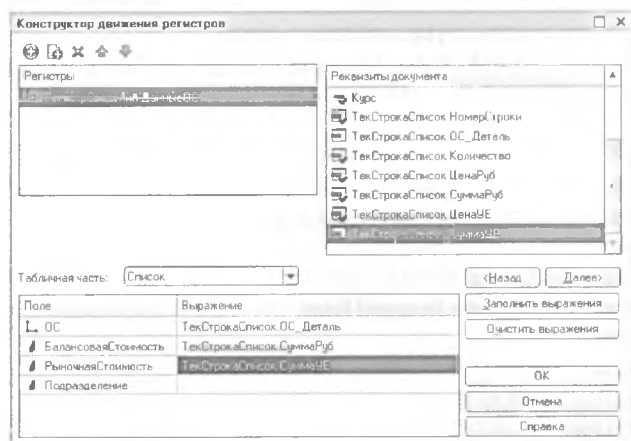


Рис. 10.12. Окно конструктора после завершения работы

Сразу после окончания своей работы в Модуле объекта появится процедура (комментарии конструктора выделены другим шрифтом).

### ЛИСТИНГ Ю\_10.5

**Процедура ОбработкаПроведения(Отказ, Режим)**

```
///__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

```
// Данный фрагмент построен конструктором.
```

```
// При повторном использовании конструктора,
```

```
// внесенные вручную изменения будут утеряны!!!
```

```
Для Каждого ТекСтрокаСписок Из Список Цикл
```

```
// регистр ДанныеОС
```

```
Движение = Движения.ДанныеОС.Добавить();
```

```
Движение.Период = Дата;
```

```
Движение.ОС = ТекСтрокаСписок.ОС_Деталь;
```

```
Движение.БалансоваяСтоимость = ТекСтрокаСписок.СуммаРуб;
```

```
Движение.РыночнаяСтоимость = ТекСтрокаСписок.СуммаУЕ;
```

```
КонецЦикла;
```

```
// записываем движения регистров
```

```
Движения.ДанныеОС.Записать();
```

```
///__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

```
КонецПроцедуры
```

Проверьте проведение документа в режиме «1С:Предприятие».

### 10.6.2. Доработка процедуры ОбработкаПроведения(...)

Сейчас наша процедура проводит документ без какой-либо обработки ошибок. Кроме того, наш регистр не может хранить детали. Поэтому мы слегка изменим текст процедуры.

#### ЛИСТИНГ Ю\_10.6

```

Процедура ОбработкаПроведения(Отказ, Режим)
Для Каждого ТекСтрокаСписок Из Список Цикл
// пропустим все детали
Если ТипЗнч(ТекСтрокаСписок.ОС_Деталь) =
Тип(«СправочникСсылка.Детали») Тогда
Продолжить;
КонецЕсли;
// Пропустим ОС не принадлежащие Фирме
Если ТекСтрокаСписок.ОС_Деталь.Фирма <> Фирма Тогда
Отказ = Истина;
Сообщить(«В документе »+ЭтотОбъект+
« в строке №»+ТекСтрокаСписок.НомерСтроки+
« ОС »+ТекСтрокаСписок.ОС_Деталь+
« не принадлежит фирме »+Фирма+«!!!»
,СтатусСообщения.ОченьВажное);
Продолжить;
КонецЕсли;
// регистр ДанныеОС
Движение = Движения.ДанныеОС.Добавить();
Движение.Период = Дата;
Движение.ОС = ТекСтрокаСписок.ОС_Деталь;
Движение.БалансоваяСтоимость = ТекСтрокаСписок.СуммаРуб;
Движение.РыночнаяСтоимость = ТекСтрокаСписок.СуммаУЕ;
КонецЦикла;
// записываем движения регистров
Движения.ДанныеОС.Записать();
КонецПроцедуры

```

### 10.6.3. Работа с движениями в форме документа

Система «1С:Предприятие 8» предоставляет возможность вставить в форму документа табличное поле, которое может быть использовано для просмотра и редактирования движений, сформированных при проведении документа.



Последняя возможность может быть применена для ручной коррекции движений уже проведенных документов в случае изменения законодательства.

### Упражнение № 10.17

Добавьте в элементе формы «Панель1» новую страницу со следующими свойствами:

Свойство	Значение
Имя	ДанныеОС
Заголовок страницы	Данные ОС

Вызовите команду «Форма — Размещение данных» и вставьте в форму на этой странице табличное поле «ДанныеОС». После чего, сделав клик правой кнопкой мыши, на этом поле выберите пункт меню «Размещение данных» и отметьте колонки согласно рис. 10.13.

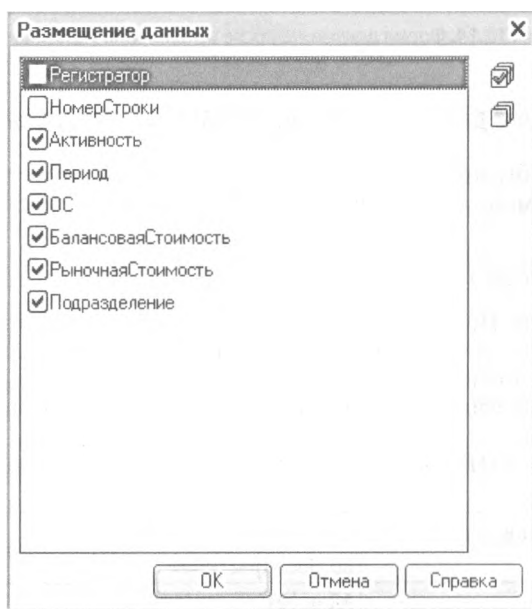


Рис. 10.13. Вставка колонок табличного поля

Добавьте к этому полю командную панель (рис. 10.14).

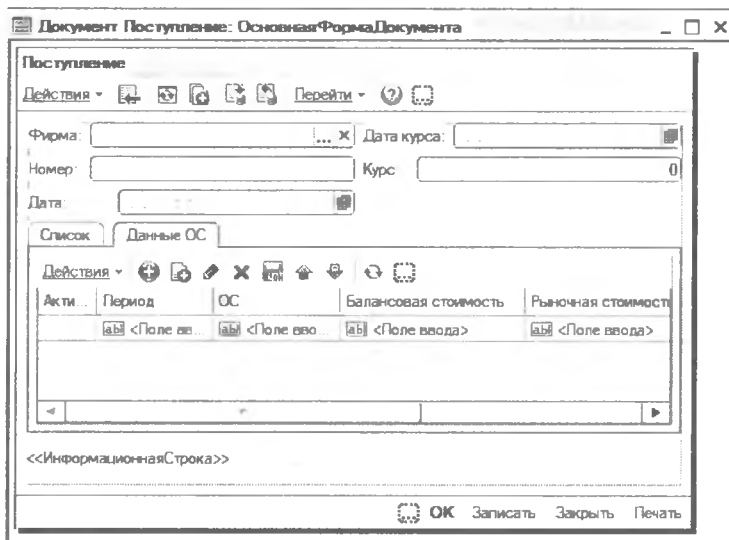


Рис. 10.14. Форма документа после выполнения упражнения

## 10.7. РАБОТА С ДОКУМЕНТОМ В РЕЖИМЕ «1С:ПРЕДПРИЯТИЕ»

Для того чтобы протестировать работу конфигурации после произведенных изменений, мы введем в ИБ пару новых документов.

### Упражнение № 10.18

**Внимание!** Предварительно отключите процедуру **ПриЗаписи()** элемента справочника «Основные средства». Проверьте, и при необходимости исправьте, точность (должна быть 2) и длину (должна быть не меньше 10) для реквизитов **ЦенаРуб.**, **СуммаРуб.**, **ЦенаУЕ** и **СуммаУЕ** документа «Поступление».

Введите в ИБ новый документ со следующими значениями реквизитов:

Реквизит	Значение
Фирма	АО «Альтернатива»
Номер	АЛ-00001
Дата	15.08.2008 00:00:00
Дата курса	15.08.2008
Курс	24,85

Добавьте в табличную часть следующие строки:

№	ОС/Деталь	Количество	Цена руб.	Сумма руб.	Цена УЕ	Сумма УЕ
1	Головной офис	1	10 000 000,00	10 000 000,00	402 414,49	402 414,49
2	Автомобиль директора	1	1 200 000,00	1 200 000,00	48 289,74	48 289,74
Итого:		2	—	11 200 000,00	—	450704,2284

При заполнении табличной части добавьте в справочник «Основные средства» элемент «Автомобиль директора».

Сохраните и проведите документ. Какие движения сформировал документ?

Добавьте в ИБ самостоятельно еще один документ.

Введите в ИБ новый документ со следующими значениями реквизитов:

Реквизит	Значение
Фирма	<b>ЗАО «Бухгалтер»</b>
Номер	<b>БХ-00001</b>
Дата	<b>14.08.2008</b>
Дата курса	<b>13.08.2008</b>
Курс	<b>24,85</b>

Добавьте в табличную часть следующие строки:

№	ОС/Деталь	Количество	Цена руб.	Сумма руб.	Цена УЕ	Сумма УЕ
1	Принтер лазерный	1	13 038,00	13 038,00	524,668	524,668
2	Картридж	8	750,00	6 000,00	30,18	241,4486
Итого:		9	—	19 038,00	—	766,1166

## 10.8. ЧТО МЫ УЗНАЛИ

В этой главе мы познакомились с конфигурированием документов.

Напомним, что документы используются для регистрации актов хозяйственной деятельности, которые возникают на предприятии. Для этого используется механизм *проведения* документа.

В процессе своего проведения документ формирует специальные записи в *регистрах* конфигурации, которые называются *Движениями*.

В качестве примера мы использовали регистр сведений «ДанныеОС». Мы внесли необходимые изменения в его настройку, изменив режим записи, подчинив его регистратору.

Мы подробно рассмотрели окно редактирования документа в Конфигураторе на примере документа «Поступление».

В этой главе мы рассмотрели особенности обращения к документам из встроенного языка системы. При этом мы сравнили список методов у справочников и документов и выяснили, что использование многих объектов в языке унифицировано.

В этой главе мы построили необходимые для работы пользователя формы, углубив наши знания в этом разделе конфигурирования.

Для формирования движений вы должны сформировать в модуле документа текст процедуры **ОбработкаПроведения(...)**. Именно эта процедура осуществляет заполнение свойства документа Движения.

Для формирования текста процедуры в Конфигураторе имеется Конструктор движений. Как и раньше, мы применили доработку текста процедуры, который был построен конструктором для достижения необходимого результата.

В завершение главы мы рассмотрели работу с документами из режима «1С:Предприятие».

Отчеты как объекты в дереве «Конфигурация» предназначены для построения *выходных печатных форм*. В качестве такой формы очень часто выступает табличный документ, который мы уже с вами подробно разбирали ранее.

Как правило, отчеты строятся с использованием объекта «Запрос». Особенностью этого объекта является то, что для извлечения данных необходимо сначала выполнить сам запрос, передав объекту текст запроса, написанный на специальном языке *запросов*.

Текст запроса, по сути, определяет структуру некоторой временной таблицы, которая будет построена в результате работы объекта. Эта таблица имеет тип данных — **РезультатЗапроса**, который обладает достаточным набором методов для организации выборки, извлечения информации из временной таблицы.

Синтаксис языка запросов достаточно подробно описан в документации, поэтому в этой главе мы остановимся на методике его применения для формирования отчетов.

### 11.1. СОЗДАНИЕ ОТЧЕТА (ЗАПРОСА) «ВРУЧНУЮ»

Как мы убедились ранее, в системе имеется необходимый набор конструкторов, которые облегчают процесс создания объектов в конфигурации. Но для лучшего понимания работы Конструктора запросов (который мы рассмотрим ниже) один из отчетов мы сделаем полностью «вручную» — практически без применения конструкторов.

В качестве примера мы используем уже известный нам отчет «Печать курса УЕ» — только реализуем мы его другими средствами.

Для создания нового отчета в дереве «Конфигурация» необходимо выделить ветвь «Отчеты» и, используя правую кнопку мыши, выбрать пункт «Добавить» (рис. 11.1).

#### 11.1.1. Окно редактирования отчета

Сразу после этого действия Конфигуратор создаст в дереве объект «Отчет1» и откроет его окно редактирования, которое нам уже знакомо (рис. 11.2).

Заполним основные свойства отчета следующими значениями.

#### Упражнение № 11.1

---

Создайте в конфигурации новый отчет.

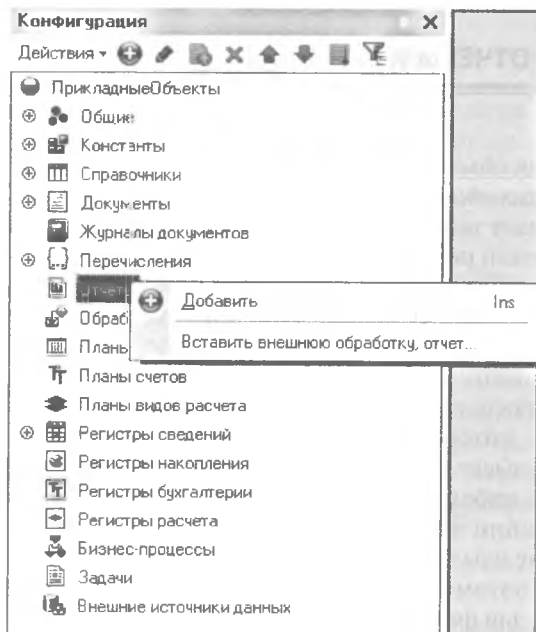


Рис. 11.1. Добавление нового отчета в конфигурацию

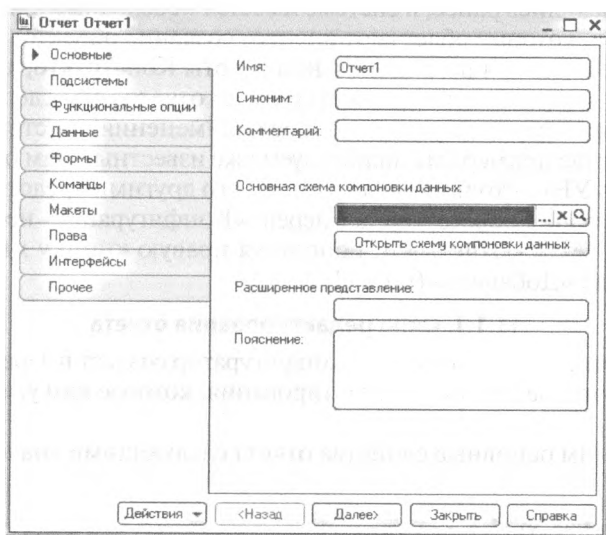


Рис. 11.2. Окно редактирования отчета

Свойство	Значение
Имя	<b>КурсыУЕ</b>
Синоним	<b>Курсы УЕ</b>
Комментарий	<b>Курсы УЕ за период</b>

Нажмите на кнопку «Далее>», для того чтобы перейти к следующей закладке.

На закладке «Данные» мы можем определить состав реквизитов, которые могут быть у отчета. Дело в том, что при построении отчета алгоритм его формирования может использовать различные параметры, влияющие на состав и структуру выходной формы.

Такие параметры пользователь может устанавливать в форме отчета в различные значения, после чего давать команду на его построение. Представляется разумным, что между вызовами одного и того же отчета пользователь захочет сохранить эти значения в информационной базе. Поэтому нам необходимо будет создать именно *реквизиты*, а не элементы формы.

Кроме того, отчет можно формировать программным путем, через обращение к соответствующему менеджеру. А раз так, то мы должны предоставить вызывающей процедуре некоторое количество *реквизитов*, чтобы отчет мог быть сформирован без открытия его формы путем установки им соответствующих значений.

Отчет, который был у нас раньше, печатал все курсы за все даты, которые есть в регистре сведений. Вполне возможно, что наш пользователь захочет ограничить отчет каким-то *периодом*.

Для того чтобы предоставить ему такую возможность, мы создадим в отчете два реквизита.

## Упражнение № 11.2

На закладке «Данные» добавьте реквизит, который будет содержать начало периода, со следующими свойствами:

Свойство	Значение
Имя	<b>НачПериода</b>
Синоним	<b>Начало периода</b>
Комментарий	<b>Начало периода</b>
Тип	<b>Дата</b>
Состав даты	<b>Дата</b>

И еще один такой же по типу данных реквизит, который будет содержать конец периода:

Свойство	Значение
Имя	КонПериода
Синоним	Конец периода
Комментарий	Конец периода
Тип	Дата
Состав даты	Дата

Нажмите на кнопку «Далее>», для того чтобы перейти к следующей закладке.

На следующем этапе мы создадим основную форму отчета (к этому моменту Вы уже должны уметь это делать хорошо).

### 11.1.2. Особенности создания формы отчета

В этом подразделе мы рассмотрим особенности создания формы отчета. И прежде чем это сделать, создайте новую форму отчета, используя для этой цели кнопку открытия на соответствующей закладке окна редактирования.

#### Упражнение № 11.3

В Конструкторе укажите следующие значения в полях его диалога:

Свойство	Значение
Выберете тип формы	Форма отчета
Назначить форму основной	<input checked="" type="checkbox"/>
Имя	ОсновнаяФорма
Синоним	Основная форма
Комментарий	Основная форма отчета
Командная панель сверху	<input checked="" type="checkbox"/>
Командная панель снизу	<input checked="" type="checkbox"/>

Нажмите на кнопку «Далее>», для того чтобы рассмотреть следующий шаг конструктора более подробно.

Поскольку мы собираемся в форме предоставить пользователю возможность указывать период, на втором этапе работы конструктора мы откажемся от автоматического размещения в форме реквизитов нашего отчета (рис. 11.3).

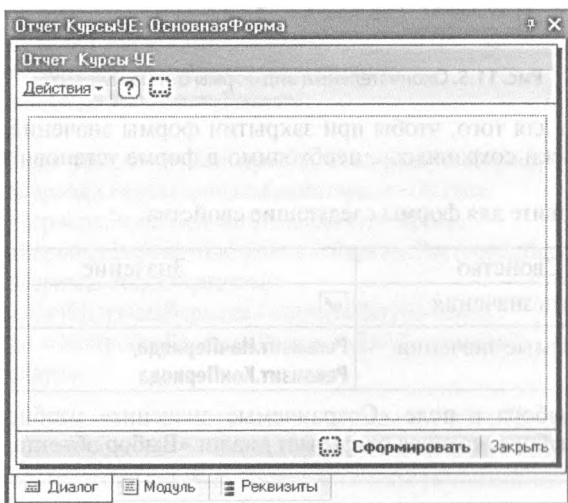




**Рис. 11.3.** Отказ от предварительного размещения в форме реквизитов отчета

Поэтому на этом этапе следует просто нажать на кнопку «Готово», что в результате приведет к созданию формы, изображенной на рис. 11.4.

Над этой формой следует еще поработать. Для начала уменьшим ее размер.



**Рис. 11.4.** Форма отчета по окончании работы конструктора

## Упражнение № 11.4

Установите в качестве размера формы следующие значения:

Свойство	Значение
Ширина	280
Высота	100

Для удобства работы пользователя нам нужно будет вставить в форму группу элементов, которые будут отвечать за выбор пользователем периода запроса.

Теперь мы можем вставить группу элементов формы, которые помогут пользователю выбрать период.

Используйте пункт меню «Форма — вставить элемент управления...». Выберите в качестве типа элемента «Выбор периода» и разместите вставляемые элементы в форме так, как это изображено на рис. 11.5.

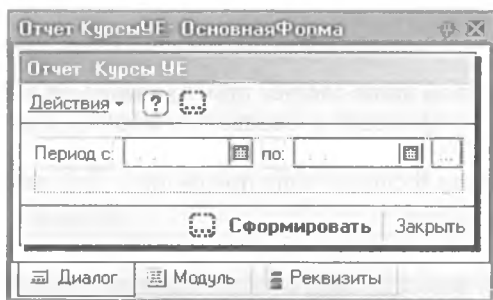


Рис. 11.5. Окончательный вид формы отчета «КурсыУЕ»

Теперь для того, чтобы при закрытии формы значение периода автоматически сохранялось, необходимо в форме установить некоторые свойства.

Установите для формы следующие свойства:

Свойство	Значение
Сохранять значения	<input checked="" type="checkbox"/>
Сохраняемые значения	Реквизит.НачПериода, Реквизит.КонПериода

Для выбора в поле «Сохраняемые значения» удобно применить кнопку выбора, которая открывает диалог «Выбор объекта» (рис. 11.6).

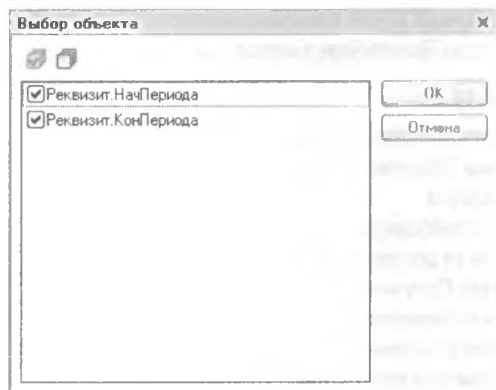


Рис. 11.6. Выбор реквизитов, которые будут сохраняться формой

### 11.1.3. Модуль формы отчета

Теперь мы можем собственно приступить к написанию процедуры, которая будет выполнять запрос к таблицам ИБ и формировать результирующий табличный документ.

Сейчас наш модуль содержит следующие процедуры.

#### ☁ ЛИСТИНГ Ю\_11.1

```

Процедура КнопкаСформироватьНажатие(Элемент)
// Вставить содержимое обработчика.
КонецПроцедуры
Процедура ВыбПериодНажатие(Элемент)
НастройкаПериода = Новый НастройкаПериода;
НастройкаПериода.УстановитьПериод(
НачПериода,
?(КонПериода='0001-01-01', КонПериода, КонецДня(КонПериода)));
НастройкаПериода.РедактироватьКакИнтервал = Истина;
НастройкаПериода.РедактироватьКакПериод = Истина;
НастройкаПериода.ВариантНастройки = ВариантНастройкиПериода.Период;
НастройкаПериода.Редактировать();
НачПериода = НастройкаПериода.ПолучитьДатуНачала();
КонПериода = НастройкаПериода.ПолучитьДатуОкончания();
КонецПроцедуры

```

Для формирования отчета мы используем обработчик нажатия кнопки «Сформировать» — **КнопкаСформироватьНажатие(Элемент)**.

Во многом наш алгоритм будет совпадать с аналогичной процедурой из модуля основной формы списка регистра сведений «Курсы».

Поэтому вставьте в нее следующий текст из модуля формы регистра из процедуры **ПечатьЧерезЗапрос(...)**.

## ЛИСТИНГ Ю\_11.2

```
Процедура КнопкаСформироватьНажатие(Элемент)
ТабДок = Новый ТабличныйДокумент();
//Получение макета
Макет = ПолучитьОбщийМакет(«КурсУЕ»);
//Получение части документа из макета
Область = Макет.ПолучитьОбласть(«Заголовок»);
// действия по заполнению параметров части документа
Область.Параметры.НаименованиеОрганизации =
Константы.НаименованиеОрганизации.Получить();
ТабДок.Вывести(Область);
Область = Макет.ПолучитьОбласть(«Шапка»);
ТабДок.Вывести(Область);
Область = Макет.ПолучитьОбласть(«Строка»);
//Создаем объект запрос
Запрос = Новый Запрос();
//Формируем текст запроса
Запрос.Текст =
«ВЫБРАТЬ Курсы.Период, Курсы.Курс
|ИЗ РегистрСведений.Курсы КАК Курсы
|УПОРЯДОЧИТЬ ПО Период УБЫВ»;
//И выполняем запрос
Результат = Запрос.Выполнить();
// Обработка результата
Выборка=Результат.Выбрать();
Пока Выборка.Следующий() Цикл
Область.Параметры.Дата = Выборка.Период;
Область.Параметры.Значение = Выборка.Курс;
ТабДок.Вывести(Область);
КонецЦикла;
// Установка свойств табличного документа в целом
// перед его отображением:
ТабДок.ИмяПараметровПечати = «ПараметрыОтчетаКурсыУЕ»;
ТабДок.ИмяСохраненияПоложенияОкна = «ОкноОтчетаКурсыУЕ»;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.ОтображатьСетку = Ложь;
ТабДок.СохранятьСвойстваОтображения= Истина;
ТабДок.ТолькоПросмотр = Истина;
ТабДок.ФиксацияСверху = 3;
ТабДок.Показать(«История курса УЕ»);
КонецПроцедуры
```

#### 11.1.4. Изменение текста запроса и его выполнение

В середине нашей процедуры есть место, которое содержит текст запроса:

```
«ВЫБРАТЬ Курсы.Период, Курсы.Курс  
|ИЗ РегистрСведений.Курсы КАК Курсы  
|УПОРЯДОЧИТЬ ПО Период УБЫВ»;
```

Но колонки в шапке нашего отчета называются «Дата» и «Значение» соответственно. Поэтому мы можем максимально приблизить структуру результата запроса к печатной форме, задав с помощью ключевого слова КАК псевдоним к каждой колонке:

```
Запрос.Текст =  
«ВЫБРАТЬ  
|Курсы.Период КАК Дата,  
|Курсы.Курс КАК Значение  
|ИЗ РегистрСведений.Курсы КАК Курсы  
|УПОРЯДОЧИТЬ ПО Дата УБЫВ»;
```

Однако пока в выборку попадают все записи регистра (за все возможные периоды). Для того чтобы записи удовлетворяли какому-либо условию, необходимо в тексте запроса после оператора **ВЫБРАТЬ** применить ключевое слово **ГДЕ**, после которого написать условие отбора записей. В нашем случае таким условием будет условие «**МЕЖДУ ... И ...**»:

```
Запрос.Текст =  
«ВЫБРАТЬ  
|Курсы.Период КАК Дата,  
|Курсы.Курс КАК Значение  
|ИЗ РегистрСведений.Курсы КАК Курсы  
|ГДЕ Курсы.Период МЕЖДУ &НачПериода И &КонПериода  
|УПОРЯДОЧИТЬ ПО Дата УБЫВ»;
```

Обратим внимание на знаки & перед именами *параметров* запроса.

Чтобы запрос работал правильно, необходимо этим параметрам присвоить значения перед его выполнением, поэтому полный текст будет выглядеть следующим образом.

#### ЛИСТИНГ Ю\_11.3

```
//Формируем текст запроса  
Запрос.Текст =  
«ВЫБРАТЬ  
|Курсы.Период КАК Дата,  
|Курсы.Курс КАК Значение
```

```

|ИЗ РегистрСведений.Курсы КАК Курсы
|ГДЕ Курсы.Период МЕЖДУ & НачПериода И & КонПериода
|УПОРЯДОЧИТЬ ПО Дата УБЫВ»;
Запрос.УстановитьПараметр(«НачПериода», НачПериода);
Запрос.УстановитьПараметр(«КонПериода», КонПериода);
//И выполняем запрос
Результат = Запрос.Выполнить();

```

### 11.1.5. Выборка данных из результата запроса

Метод **.Выполнить(...)** возвращает ссылку на объект типа **Результат-Запроса**, который среди прочих имеет метод, открывающий выборку результатов:

```

// Обработка результата
Выборка=Результат.Выбрать();
Пока Выборка.Следующий() Цикл
Область.Параметры.Дата = Выборка.Дата;
Область.Параметры.Значение = Выборка.Значение;
ТабДок.Вывести(Область);
КонецЦикла;

```

Именами свойств записи, которую мы выбрали по методу **.Следующий()**, будут те имена реквизитов источников данных (или их псевдонимов), которые мы указали в тексте запроса.

#### Упражнение № 11.5

---

Внесите соответствующие изменения в текст процедуры **КнопкаФормироватьНажатие(Элемент)** и проверьте ее работоспособность.

---

## 11.2. ВЫЗОВ ОТЧЕТА ИЗ ИНТЕРФЕЙСА КОНФИГУРАЦИИ

Сейчас наш отчет можно вызвать, только используя пункт меню «Операции — Отчеты...». Поэтому сначала мы осуществим открытие формы отчета из пункта меню.

#### Упражнение № 11.6

---

В окне редактирования отчета добавьте в основной интерфейс конфигурации пункт меню:

Свойство	Значение
Тип	Действие
Действие	Курсы УЕ: Открыть отчет

Свойство	Значение
Текст	Курсы УЕ
Подсказка	Курсы УЕ
Пояснение	Открыть отчет Курсы УЕ

Теперь в режиме «1С:Предприятие» мы можем проверить, как все это будет работать.

Выберем пункт меню «КурсыУЕ». У нас откроется форма отчета. Если в ней сразу же нажать на кнопку «Сформировать», то результат работы окажется пустым (рис. 11.7).

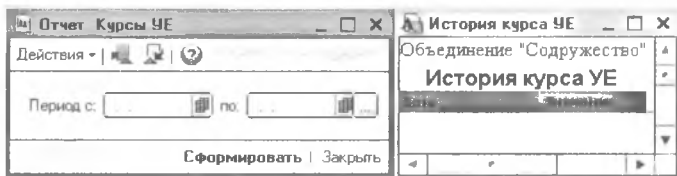


Рис. 11.7. Пустой результат запроса

Это произошло потому, что мы не указали период запроса. Хорошо, если бы при открытии формы в качестве начала и конца периода задавались бы начальные значения.

В обработчике события «При открытии» основной формы отчета «КурсыУЕ» установите значения периода, равным трем предыдущим месяцам, считая от рабочей даты, если ни дата начала, ни дата конца периода не указаны.

Проверьте работоспособность формы в режиме «1С:Предприятие».

Для надежности приводим полный текст процедуры (сравните со своим вариантом).

#### ЛИСТИНГ Ю\_11.4

##### Процедура ПриОткрытии()

Если НачПериода = '0001-01-01' и КонПериода='0001-01-01' Тогда

НачПериода = ДобавитьМесяц(РабочаяДата, -3);

КонПериода = РабочаяДата;

КонецЕсли;

КонецПроцедуры

### 11.3. ОБРАЩЕНИЕ К ОТЧЕТУ ИЗ ВСТРОЕННОГО ЯЗЫКА СИСТЕМЫ

Для того чтобы получить доступ к отчету из встроенного языка, мы должны обратиться к свойству глобального контекста Отчеты (типа ОтчетыМенеджер), после чего указать его имя — Отчеты.КурсыУЕ.

Прежде чем что-то делать с отчетом, необходимо создать его экземпляр, используя метод **.Создать()**:

```
ОтчетКурсыУЕ = Отчеты.КурсыУЕ.Создать();
```

Вот теперь переменная будет содержать ссылку на экземпляр конкретного отчета, у которого есть реквизиты. А им нужно что-то присвоить (как мы помним) перед формированием выходной формы:

```
ОтчетКурсыУЕ.КонПериода = РабочаяДата;
```

Но как запустить формирование самой формы?

Ответ тут может быть только один — разместить в *Модуле отчета* ее текст.

### 11.3.1. Модуль отчета

Чтобы открыть этот модуль, можно нажать на самом отчете в дереве «Конфигурация» правую кнопку мыши и выбрать пункт «Открыть модуль объекта».

Что мы можем разместить в этом модуле?

Особенностью его является то, что он не может содержать никаких обработчиков событий. Поэтому разместить в нем можно (и даже нужно!) процедуру, которая будет заниматься формированием отчета.

#### Упражнение № 11.7

---

Наберите в модуле отчета «КурсыУЕ» заголовок процедуры **СформироватьОтчет()** с добавочным ключевым словом «Экспорт».

Перенесете в нее все содержимое процедуры **КнопкаСформироватьНажатие(Элемент)** из модуля формы. После чего обеспечьте ее вызов из этой процедуры.

---

Если вы все сделали правильно, то текст процедуры в модуле формы будет выглядеть так:

#### ЛИСТИНГ Ю\_11.5

```
//КнопкаСформироватьНажатие(Элемент) – формирует  
//отчет с использованием запроса  
Процедура КнопкаСформироватьНажатие(Элемент)  
СформироватьОтчет();  
КонецПроцедуры
```

А модуль отчета после ваших действий должен содержать:

#### ЛИСТИНГ Ю\_11.6

```
// Модуль объекта «Отчет.КурсыУЕ»  
Процедура СформироватьОтчет() Экспорт
```



```

ТабДок = Новый ТабличныйДокумент();
//Получение макета
Макет = ПолучитьОбщийМакет(«КурсУЕ»);
//Получение части документа из макета
Область = Макет.ПолучитьОбласть(«Заголовок»);
// действия по заполнению параметров части документа
Область.Параметры.НаименованиеОрганизации =
Константы.НаименованиеОрганизации.Получить();
ТабДок.Вывести(Область);
Область = Макет.ПолучитьОбласть(«Шапка»);
ТабДок.Вывести(Область);
Область = Макет.ПолучитьОбласть(«Строка»);
//Создаем объект запрос
Запрос = Новый Запрос();
//Формируем текст запроса
Запрос.Текст =
«ВЫБРАТЬ
|Курсы.Период КАК Дата,
|Курсы.Курс КАК Значение
|ИЗ РегистрСведений.Курсы КАК Курсы
|ГДЕ Курсы.Период МЕЖДУ &НачПериода И &КонПериода
|УПОРЯДОЧИТЬ ПО Дата УБЫВ»;
Запрос.УстановитьПараметр(«НачПериода», НачПериода);
Запрос.УстановитьПараметр(«КонПериода», КонПериода);
//И выполняем запрос
Результат = Запрос.Выполнить();
// Обработка результата
Выборка = Результат.Выбрать();
Пока Выборка.Следующий() Цикл
    Область.Параметры.Дата = Выборка.Дата;
    Область.Параметры.Значение = Выборка.Значение;
    ТабДок.Вывести(Область);
КонецЦикла;
// Установка свойств табличного документа в целом
// перед его отображением:
ТабДок.ИмяПараметровПечати = «ПараметрыОтчетаКурсыУЕ»;
ТабДок.ИмяСохраненияПоложенияОкна = «ОкноОтчетаКурсыУЕ»;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.ОтображатьСетку = Ложь;
ТабДок.СохранятьСвойстваОтображения=Истина;
ТабДок.ТолькоПросмотр = Истина;
ТабДок.ФиксацияСверху = 3;
ТабДок.Показать(«История курса УЕ»);
КонецПроцедуры

```

После таких преобразований мы можем вызвать наш отчет средствами встроенного языка следующим образом:

```
// Создать объект
ОтчетКурсыУЕ = Отчеты.КурсыУЕ.Создать();
// Настроить параметры отчета
ОтчетКурсыУЕ.КонПериода = РабочаяДата;
// Сформировать сам отчет
ОтчетКурсыУЕ.СформироватьОтчет();
```

Добавим в наш основной интерфейс кнопку, аналогичную пункту меню, который был рассмотрен ранее, измените ее свойства, выполнив следующее упражнение.

#### Упражнение № 11.8

---

Добавьте в основной интерфейс кнопку на панели инструментов:

Свойство	Значение
Тип	Действие
Действие	СформироватьОтчетКурсыУЕ

Остальные свойства можно оставить без изменения.

Когда вы вводили имя действия **СформироватьОтчетКурсыУЕ**, Конфигуратор разместил в Модуле приложения заготовку.

Введите в текст процедуры следующие строки:



#### ЛИСТИНГ Ю\_11.7

```
//СформироватьОтчетКурсыУЕ() – пример формирования отчета
// программным путем
Процедура СформироватьОтчетКурсыУЕ() Экспорт
// Создать объект
ОтчетКурсыУЕ = Отчеты.КурсыУЕ.Создать();
// Настроить параметры отчета
ОтчетКурсыУЕ.КонПериода = РабочаяДата;
// Сформировать сам отчет
ОтчетКурсыУЕ.СформироватьОтчет();
КонецПроцедуры
```

Проверьте работоспособность пункта меню и кнопки на панели инструментов.

---

## 11.4. КОНСТРУКТОР ВЫХОДНЫХ ФОРМ

Мы уже использовали этот конструктор, а здесь мы применим его для создания объекта типа «Отчет». В качестве примера мы построим печатную форму из справочника «Фирмы». Сначала мы добавим в Конфигурацию новый отчет.

## Упражнение № 11.9

Добавьте в конфигурацию новый отчет со следующими свойствами:

Свойство	Значение
Имя	СписокФирм
Синоним	Список фирм
Комментарий	Список организаций объединения

Перейдите к закладке «Макеты», так как наш отчет не будет иметь ни формы, ни реквизитов. Используя кнопку «Конструкторы», выберите пункт «Конструктор выходных форм...» (рис. 11.8).

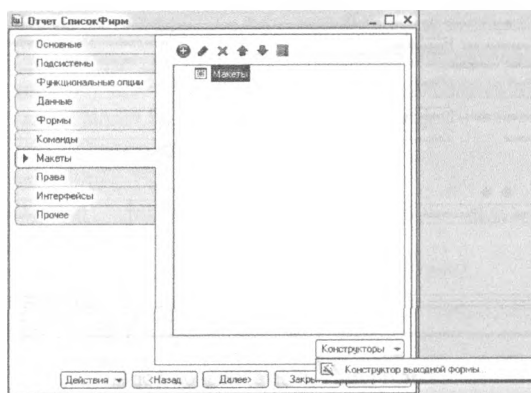


Рис. 11.8. Запуск Конструктора выходных форм

Сразу после описанных выше действий Конструктор откроет диалог по выбору места расположения выходной формы (рис. 11.9). Не меняйте предлагаемое конструктором значение и нажмите на кнопку ОК, чтобы перейти к шагам формирования выходной формы.

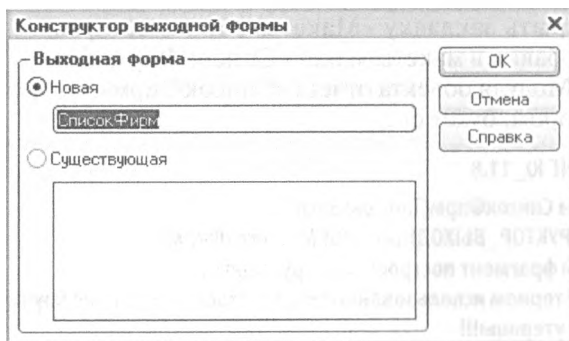


Рис. 11.9. Выбор места расположения выходной формы

## Упражнение № 11.9 (продолжение)

Используя кнопки «>» в окне диалога Конструктора, выберите в качестве полей «Код» и «Наименование» из справочника «Фирмы».

Перейдите к закладке «Порядок».

Используя кнопки «>» в окне диалога Конструктора, выберите в качестве порядка сортировки поле «Код». Перейдите к закладке «Выходная форма».

Поскольку формы у нашего отчета не будет, то переключатель «Расположение элементов управления» установите в положение «Не располагать» (рис. 11.10).

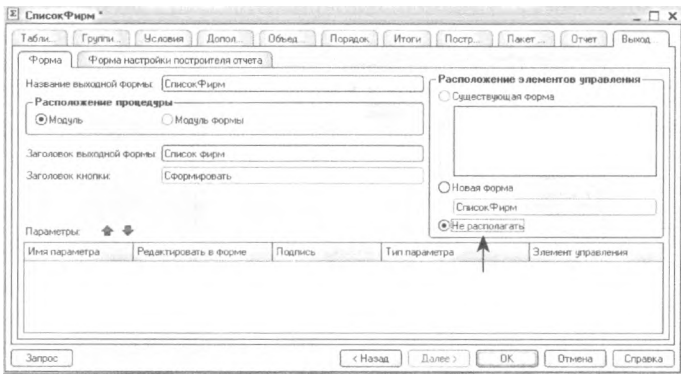


Рис. 11.10. Выбор выходной формы

*Примечание:* при выполнении последней операции в закладке «Отчет» уберите галочку из пункта «Использовать построитель отчета».

Нажмите на кнопку ОК, чтобы закончить работу с Конструктором.

Посмотрим, что у нас получилось.

Если отрыть закладку «Макеты», то можно увидеть, что конструктор добавил в макеты макет «СписокФирм».

А текст Модуля объекта отчета «СписокФирм», сформированный Конструктором, будет следующим.

### ЛИСТИНГ Ю\_11.8

**Процедура СписокФирм(ТабДок) Экспорт**

```
//{{КОНСТРУКТОР_ВЫХОДНЫХ_ФОРМ(СписокФирм)}
```

```
// Данный фрагмент построен конструктором.
```

```
// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
```

```
Макет = ПолучитьМакет(«СписокФирм»);
```

```

Запрос = Новый Запрос;
Запрос.Текст =
«ВЫБРАТЬ
| Фирмы.Код КАК Код,
| Фирмы.Наименование
| ИЗ
| Справочник.Фирмы КАК Фирмы
|
| УПОРЯДОЧИТЬ ПО
| Код»;
Результат = Запрос.Выполнить();
ОбластьЗаголовок = Макет.ПолучитьОбласть(«Заголовок»);
ОбластьПодвал = Макет.ПолучитьОбласть(«Подвал»);
ОбластьШапкаТаблицы = Макет.ПолучитьОбласть(«ШапкаТаблицы»);
ОбластьПодвалТаблицы = Макет.ПолучитьОбласть(«ПодвалТаблицы»);
ОбластьДетали = Макет.ПолучитьОбласть(«Детали»);
ТабДок.Вывести(ОбластьЗаголовок);
ТабДок.Вывести(ОбластьШапкаТаблицы);
ВыборкаДетали = Результат.Выбрать();
Пока ВыборкаДетали.Следующий() Цикл
ОбластьДетали.Параметры.Заполнить(ВыборкаДетали);
ТабДок.Вывести(ОбластьДетали);
КонецЦикла;
ТабДок.Вывести(ОбластьПодвалТаблицы);
ТабДок.Вывести(ОбластьПодвал);
/};КОНСТРУКТОР_ВЫХОДНЫХ_ФОРМ
КонецПроцедуры

```

## Упражнение № 11.10

---

Вставьте в меню «Отчеты» основного интерфейса конфигурации пункт меню «Список фирм» и обеспечьте вызов нашего отчета.

Проверьте работоспособность Конфигурации.

*Примечание:* если данный способ не сработает, то в конструкторе выходной формы на вкладке «Выходная форма» необходимо оставить галочку на поле «Новая форма» (иначе придется самим создавать форму для данного отчета).

---

В последующих разделах содержится дополнительный материал, который по тем или иным причинам не вошел в основную часть курса.

Большая часть материала предназначена для самостоятельного изучения. Однако преподаватель вправе использовать ее как в основной части курса, так и в качестве готовых ответов на дополнительные вопросы слушателей.

Часть материала предназначена полностью для самостоятельного закрепления материала основной части после окончания обучения.

### ПРИЛОЖЕНИЕ А

#### ОСНОВНЫЕ ЭЛЕМЕНТЫ ВСТРОЕННОГО ЯЗЫКА

Это приложение предназначено для тех, кто не владеет в достаточном объеме встроенным языком системы «1С:Предприятие». Читатели, имеющие опыт программирования в системе «1С:Предприятие 7.7», смогут систематизировать и углубить свои знания по данной теме.

Встроенный язык системы «1С: Предприятие 8» (далее — язык) является объектным языком программирования. Он предназначен для описания прикладных алгоритмов функционирования прикладной системы (конфигурации). При этом существенная часть разработки такого решения делается визуальным образом в Конфигураторе, а язык используется только там, где необходимо выполнить алгоритмическую настройку системы.

Для облегчения написания текстов в языке имеется практически полная двуязычность, которая выражается в том, что одни и те же структурные конструкции, зарезервированные слова, имена объектов, методов и свойств могут быть записаны как по-русски, так и по-английски в одном тексте программы.

Язык обладает достаточным набором типов данных, которые Вы можете использовать при обработке информации.

Для удобства рассмотрения дальнейшего материала, а также для закрепления основной части курса создайте новую информационную базу, выполнив упражнение.

#### Упражнение № А.1

---

Создайте новую информационную базу.

Откройте конфигурацию. В свойствах конфигурации значение параметра «Основной режим запуска» замените на «Обычное приложение» (рис. А.1).

Укажите для объекта «Конфигурация» имя «Прикладные Объекты». Дайте ей название: «Приложение А» и расположите ее в каталоге C:\PO\Appendix\A.

Сохраните и обновите конфигурацию.

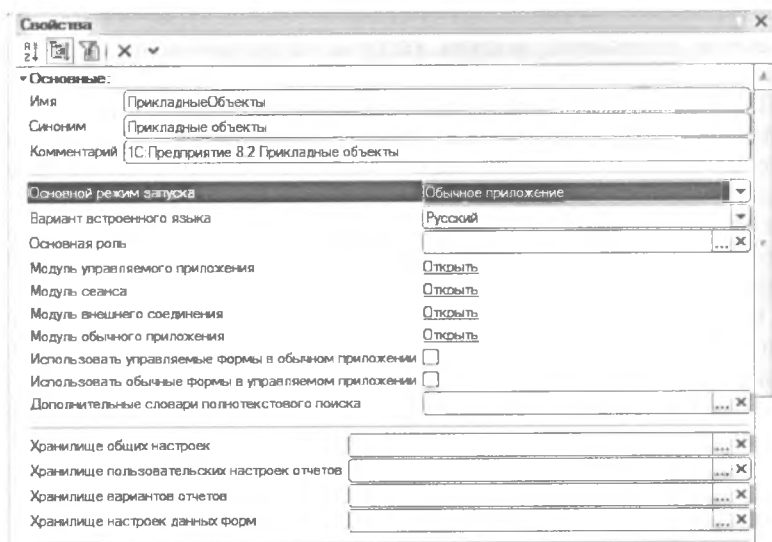


Рис. А.1. Окно свойств приложения

### А.1. Программный модуль

Текст программы, написанный на языке, размещается в программном модуле конфигурации. Модули вызываются системой в строго определенные моменты времени в тех случаях, когда необходимо реализовать специальную обработку данных. Программная часть системы производит вызов процедур и функций (размещенных разработчиком конфигурации) в модуле в строго определенные моменты времени. Всякий раз, когда мы будем говорить о вызове той или иной процедуры, мы будем упоминать тот момент времени (событие), когда это происходит. В этом смысле написание текста программы на языке системы «1С: Предприятие 8» — это ответ на вопрос: «Что нужно сделать, если произошло такое событие...?».

Так, почти каждая команда пользователя к системе в режиме «1С:Предприятие» вызывает возникновение некоторого события (или событий). Все события перечислены в документации. Каждое событие, если не предпринять каких-либо действий на встроенном языке, будет обрабатываться системой по умолчанию так, как это определено при разработке программной части.

Кроме того, события могут возникать в результате выполнения алгоритма на встроенном языке системы.

Для каждого программного модуля известен список событий, которые он может обрабатывать. Если необходимо обработать возникновение конкретного события, то текст модуля должен содержать текст, соответствующей этому событию процедуры-обработчика. Впрочем, существуют модули, которые не предназначены для обработки событий.

Вторым важным моментом для понимания логики работы модуля является его размещение в дереве объектов Конфигурации. Часто многие ошибки начинающих программистов, использующих этот язык, связаны с неверным размещением процедур и функций. В зависимости от размещения модуля его относят к одному из видов программных модулей, полный список которых имеется в описании встроенного языка системы.

Каждый модуль воспринимается системой как единое целое и, кроме того, он связан с остальной частью Конфигурации. Эта связь называется контекстом выполнения модуля. В некотором очень приблизительном смысле этого слова контекст можно себе представить как доступный в каком-либо месте набор имен свойств и методов, которые Вы можете использовать. Различают глобальный контекст и локальный контекст модуля. О том, что к какому контексту относится, мы будем говорить при рассмотрении тех или иных возможностей языка.

## **А.2. Модуль приложения**

Как уже отмечалось выше, большинство модулей предназначены для обработки определенных событий, возникающих в системе, поэтому в конфигурации может присутствовать модуль, который отвечает за сеанс работы пользователя в целом. Этот модуль носит название «Модуль приложения».

В данном случае под приложением следует понимать конкретное учетное решение, в качестве которого выступают конфигурация и система программ «1С:Предприятие», образующие в процессе своего исполнения единое целое.

Для того чтобы открыть этот модуль в Конфигураторе, необходимо в окне «Конфигурация» активизировать объект «Конфигурация» и нажать правую кнопку мыши. После чего выбрать пункт контекстного меню «Открыть модуль приложения» (рис. А.2).

Отметим, что модуль приложения инициализируется системой при ее запуске в режиме «1С:Предприятие» — первое действие, которое выполняет система сразу после открытия ИБ и считывания ее Конфигурации, но до отображения основного окна приложения.



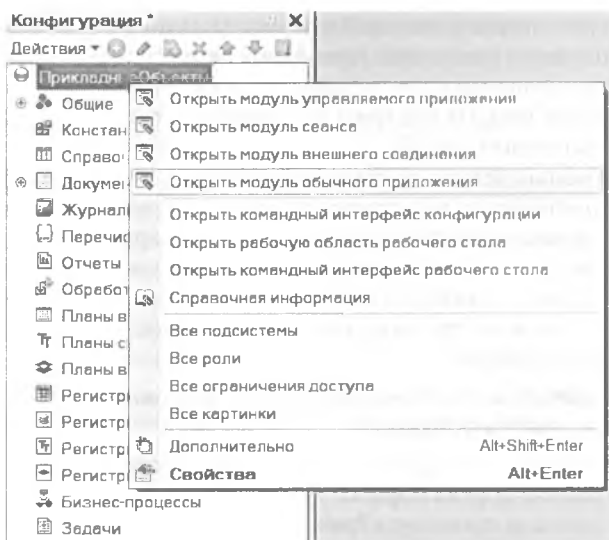


Рис. А.2. Открытие модуля обычного приложения

После того как инициализация модуля успешно завершилась, система приступает к обработке возникающих системных событий.

Поэтому в теле модуля могут содержаться тексты процедур обработчиков событий, которые призваны реализовать прикладные алгоритмы их обработки.

### А.3. Обработка событий в модуле приложения

Для Модуля приложения имена процедур — обработчиков событий строго фиксированы:

- ✓ **ПередНачаломРаботыСистемы(<Отказ>)**
- ✓ **ПриНачалеРаботыСистемы()**
- ✓ **ОбработкаВнешнегоСобытия(<Источник>, <Событие>, <Данные>)**
- ✓ **ПередЗавершениемРаботыСистемы(<Отказ>)**
- ✓ **ПриЗавершенииРаботыСистемы()**

Для каждой такой процедуры в документации описано, когда событие возникает, а также какие действия можно выполнять в этой процедуре.

Однако у начинающих программирование в этой системе часто возникает вопрос: «В каком порядке возникают события?».

Ниже мы дадим краткий ответ на этот вопрос, не повторяя описание каждой из процедур-обработчиков, так как оно имеется в документации.

Итак, сразу после успешной инициализации модуля приложения система пытается разыскать процедуру **ПередНачаломРаботыСистемы(Отказ)**, в которой вы можете выполнить подготовительные действия перед началом работы системы или вообще отменить ее запуск (до открытия основного окна).

Если отмены сделано не было, то тогда система отображает основное окно приложения и пытается найти процедуру **ПриНачалеРаботыСистемы()**. Если такая процедура есть, то система передает ей управление. В этой процедуре вы можете уже вести диалог с пользователем.

Если есть необходимость в этот момент времени завершить работу системы, то можно применить одну из встроенных процедур для управления системой:

- ✓ **ЗавершитьРаботуСистемы(<Запрашивать возможность>);**
- ✓ **ПрекратитьРаботуСистемы().**

Последняя из них принудительно прекращает работу системы, блокируя работу других обработчиков событий.

После того как процедура **ПриНачалеРаботыСистемы()** закончила свою работу, система приступает к обработке событий, возникающих в процессе работы пользователя в режиме «1С:Предприятие». При этом для их обработки будут вызываться соответствующие процедуры — обработчики событий, принадлежащие прикладным объектам в конфигурации.

Такой порядок обработки событий будет продолжаться до тех пор, пока либо пользователь не даст команду на закрытие основного окна приложения либо в каком-либо программном модуле не будет сделан вызов процедуры **ЗавершитьРаботуСистемы(<Запрашивать возможность>)**, после чего система сделает попытку передать управление в процедуру-обработчик **ПередЗавершениемРаботыСистемы(<Отказ>)** модуля приложения.

В этой процедуре вы можете запросить у пользователя подтверждение и отменить закрытие основного окна приложения.

Если отмены на этом этапе сделано не было, то система пытается найти процедуру-обработчик **ПриЗавершенииРаботыСистемы()**, в которой можно выполнить заключительные действия, перед тем как задача «1С:Предприятие» вернет управление операционной системе.

Обработку событий мы рассмотрим на примере выполнения следующего упражнения.

## Упражнение № А.2

---

Необходимо сделать так, чтобы при запуске «1С:Предприятия» в интервале с 13 до 14 часов текущего времени система запрашивала у пользователя подтверждение на продолжение его работы. И в случае

получения отрицательного ответа выполняла бы безусловное ее завершение.

Для выполнения упражнения наберите следующий текст.

#### ☁ ЛИСТИНГ П\_А.1

```
//Модуль приложения
//ПриНачалеРаботыСистемы() — проверяет время входа в систему
Процедура ПриНачалеРаботыСистемы()
ВремяЧас=Час(ТекущаяДата());
Если (ВремяЧас>=13) и (ВремяЧас<14) Тогда
Ответ=Вопрос(«Сейчас не рабочее время! Будете работать?», РежимДиалогаВопрос.ДаНет);
Если Ответ=КодВозвратаДиалога.Нет Тогда ПрекратитьРаботуСистемы();
КонецЕсли;
КонецЕсли;
КонецПроцедуры
```

Для того чтобы проверить работоспособность нашей процедуры, необходимо будет сохранить и обновить конфигурацию. После чего запустить режим «1С:Предприятие».

Если Вы все сделали правильно, то сразу после запуска конфигурации на исполнение Вы получите диалог, изображенный на рис. А.3.

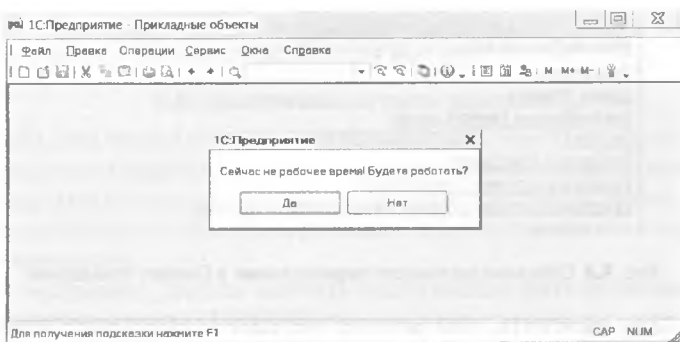


Рис. А.3. Результат обработки события в Модуле приложения


В случае положительного ответа наша конфигурация продолжает работу, а в противном случае работа будет прекращена немедленно.

#### А.4. Системные перечисления

В приведенном выше тексте процедуры мы при вызове функции **Вопрос(...)** в качестве одного из фактических значений использовали значение **РежимДиалогаВопрос.ДаНет**. Такой *тип данных* в языке «1С:Предприятия» называется *системным перечислением*.

Системные перечисления были специально введены в язык «1С: Предприятия» для того, чтобы сделать более читаемыми вызовы системных процедур и функций.

Если установить курсор в тексте процедуры на этом значении и нажать на клавиши **Ctrl+F1**, то Конфигуратор откроет специальное справочное окно «Синтакс-помощник», в котором мы сможем увидеть список всех возможных значений.

Кнопка  в окне Синтакс-помощника позволит найти в дереве соответствующий описанию объект (рис. А.4).

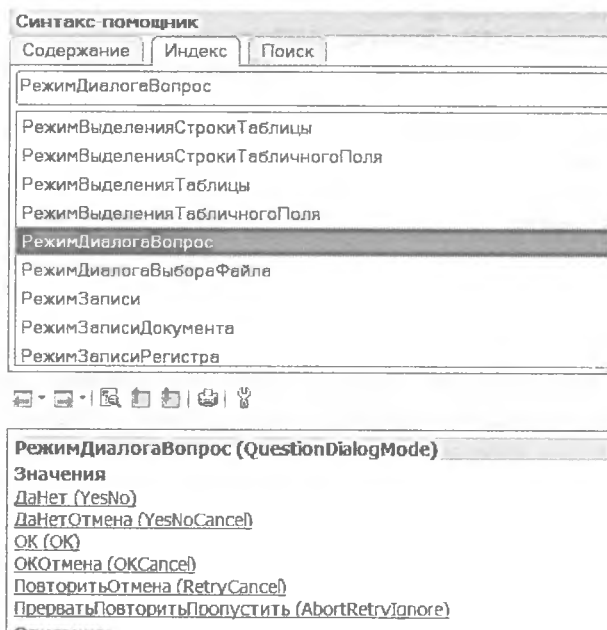


Рис. А.4. Описание системного перечисления в Синтакс-помощнике

Следует заметить, что многие методы глобального контекста (а также методы объектов) могут возвращать в качестве значения — значение системного перечисления.

Именно по этой причине мы при проверке ответа, который нам дал пользователь, в условном операторе написали следующее условие:

**Если Ответ = КодВозвратаДиалога.Нет Тогда**

#### **А.5. Системные наборы значений**

Как уже отмечалось выше, системные перечисления содержат некоторые заранее заданные наборы значений.

В этом смысле *системные наборы значений* на них очень похожи (по форме записи).

Разница между ними состоит в том, что системное перечисление представляет собой некоторый тип данных. При этом количество возможных значений и способ их записи определяются именно этим типом данных.

Системные наборы значений представляют собой просто некоторую коллекцию предопределенных значений конкретного типа данных, но множество значений соответствующего типа данных этим не ограничивается.

Приведем пример.

Предположим, что текст сообщения в функции **Вопрос(...)**, что мы использовали выше, мы могли бы между двумя предложениями вставить символ табуляции, просто набрав его на клавиатуре. Тогда соответствующий фрагмент текста процедуры будет следующим:

**Ответ = Вопрос(«Сейчас не рабочее время! Будете работать?»,  
РежимДиалогаВопрос.ДаНет);**

Но, кроме визуально большего расстояния между двумя символами, нам это ничего не даст, так как понять по тексту программы, что в этом месте стоит символ табуляции, мы не сможем.

Поэтому более читабельной будет запись с использованием системного набора значений «Символы»:

**Ответ = Вопрос(«Сейчас не рабочее время!»+Символы.Таб+  
«Будете работать?»),РежимДиалогаВопрос.ДаНет);**

## **А.6. Прimitивные типы данных**

Среди всех типов данных, что есть в языке, имеются типы, которые относятся к группе *прimitивных типов данных*. Для большинства типов из этой группы отличительной чертой является возможность записывать в тексте модуля обращения к конкретным их значениям в виде *литералов*.

Так, например, запись любого числа может быть произведена в тексте модуля явно: 1234567.89.

Строка «Сейчас нерабочее время!», что мы использовали выше по типу данных — Строка.

К примитивным типам относятся:

- ✓ Неопределено;
- ✓ Булево;
- ✓ Число;
- ✓ Строка;
- ✓ Дата;
- ✓ Null;
- ✓ Тип.

Для каждого из типов определен способ записи его конкретных значений, а также список возможных операций, которые можно выполнять над значениями данного типа. Эта информация подробно изложена в документации, а именно в книге «1С:Предприятие 8. Описание встроенного языка» (входит в поставку).

Чуть ниже мы рассмотрим особенности использования некоторых примитивных типов.

#### **А.6.1. Тип данных «Неопределено»**

Состоит из одного единственного значения — Неопределено. Его мы рассмотрим чуть ниже, когда будем говорить о переменных модуля.

#### **А.6.2. Тип данных «Булево»**

Этот тип данных имеет только два значения Истина и Ложь, которые образуются в результате вычисления логических выражений.

Например:  $1 > 2$  даст в качестве результата Ложь. Но если знак неравенства изменить:  $1 < 2$ , то в результате вычисления получится Истина.

#### **А.6.3. Тип данных «Число»**

Над числами в «1С:Предприятии» можно выполнять основные арифметические операции. Кроме того, для этого типа в языке имеются встроенные математические функции.

Для читателей, имеющих опыт программирования в «1С: Предприятии» версии 7.7, следует заметить, что точность выполнения арифметических операций — 38 знаков.

#### **А.6.4. Тип данных «Строка»**

Над строками определена только одна операция «+» — сцепление строк. Кроме этой операции в языке имеется достаточно большое количество встроенных функций обрабатывающих строковые данные.

#### **А.6.5. Тип данных «Дата»**

Первое, на что необходимо обратить внимание: данные этого типа содержат не только *дату*, но и *время*. Конкретные значения (литералы) записываются следующим образом: 'ГГГГММДДЧЧММСС'.

Так, например, если нужно указать конкретную дату, то следует записать: '20030312' — это будет означать 12 марта 2003 года 0:00:00, так как время в этой записи мы опустили. Но если нужно указать конкретный момент времени, то тогда необходимо использовать полный *канонический* способ записи: '20030312141122'. Что будет означать 12 марта 2003 г. 14 часов 11 минут 22 секунды.

Если нужно указать только время, то тогда все равно перед временем следует написать *дату отсчета* — ‘00010101’: ‘00010101141122’. Такая запись будет означать 1 января 0001 года (*точка отсчета*) 14:11:22, или просто 14 часов 11 минут 22 секунды.

Над датами определены операции «+» и «-», а также обычные операции сравнения: «<», «<=», «=», «>=», «>» и «<>».

При этом нужно помнить, что если вычесть одно значение типа «Дата» из другого значения типа «Дата», то результатом будет число секунд между этими двумя моментами времени.

Примеры:

‘20030312’-‘20030311’ даст в результате вычисления число 86 400.

К дате можно прибавлять (отнимать) необходимое число секунд: ‘200303121430’+300 даст после вычисления момент 12.03.2003 14:35:00.

#### **А.6.6. Тип данных «Null»**

Используется в языке запросов. Мы рассмотрим его позже, когда будем изучать объект «Запрос» и язык запросов.

#### **А.6.7. Тип данных «Тип»**

Тип какого-либо значения — такой же тип данных, как и число, строка или, например, Null. Единственное его отличие от типов, что были рассмотрены выше, — это отсутствие литеральной записи.

Если возникает необходимость получить конкретное значение типа «Тип», то следует применить функцию: Тип(«ИмяТипа»). Так, например, запись Тип(«Число») позволит получить значение *Тип Число*.

### **А.7. Разделы Модуля приложения**

Большинство программных модулей (но не все) имеют одинаковую структуру.

Структурно модуль может состоять из трех разделов:

- ✓ Раздел переменных;
- ✓ Раздел процедур и функций;
- ✓ Раздел основной программы.

Разделы могут размещаться только в порядке, перечисленном выше. При этом какой-либо из разделов и даже все могут отсутствовать.

*Модуль приложения* может иметь все три раздела, перечисленных выше. Сейчас этот модуль содержит только один раздел — раздел процедур и функций, в котором сейчас находится только одна процедура.

**Замечание.** Не рекомендуется в разделе основной программы выполнять какие-либо интерактивные действия, т.е. задавать пользователю

вопросы и получать от него ответы. Как правило, для этой цели можно использовать раздел процедур и функций модуля.

Использование разделов модуля мы рассмотрим на примере решения следующего упражнения.

### Упражнение № А.3

Сделайте так, чтобы по команде пользователя «Файл — Выход» или «Закрыть Alt+F4» система задавала следующий вопрос путем вывода диалога, который изображен на рис. А.5.

Причем, если ответ от пользователя так и не будет получен по истечении 30 секунд, то «1С:Предприятие» должно закончить свою работу без выдачи каких-либо дополнительных сообщений.

Если пользователь нажал на кнопку «Нет», то программа не должна быть завершена.

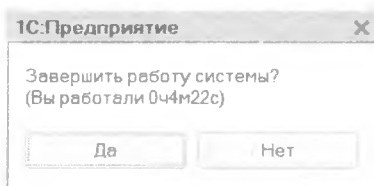


Рис. А.5. Диалог для подтверждения о выходе из системы

Для начала нам необходимо будет сохранить время начала работы. Но как это сделать? Единственный выход: объявить переменную и присвоить ей значение сразу после запуска системы. А момент этот, напомним, наступит сразу после завершения компиляции модуля.

Поэтому текст нашего модуля будет выглядеть следующим образом.

### ЛИСТИНГ П\_А.2

```
// Модуль приложения
//-----
// Раздел переменных:
Перем НачалоРаботы;
//-----
//Раздел процедур и функций:
//ПриНачалеРаботыСистемы() — проверяет время входа в систему
Процедура ПриНачалеРаботыСистемы()
    ВремяЧас=Час(ТекущаяДата());
    Если (ВремяЧас>=14) и (ВремяЧас<15) Тогда
        Ответ=Вопрос(«Сейчас не рабочее время!»+Символы.Таб+
«Будете работать?»,РежимДиалогаВопрос.ДаНет);
```



```

Если Ответ=КодВозвратаДиалога.Нет Тогда
  ПрекратитьРаботуСистемы();
КонецЕсли;
КонецЕсли;
КонецПроцедуры
//-----
// раздел основной программы:
НачалоРаботы=ТекущаяДата();

```

В этом тексте ключевое слово «Перем» объявляет переменную и ее начальное значение — типа «Неопределено».

Тут нужно отметить, что переменная в разные моменты времени может принимать значения разных типов.

Например:

```

Перем А;
А=12345; //сейчас это число
А=«абвгде»; //а теперь это строка
А=Неопределено; //а теперь это неизвестно что...
А='20030109'; //а теперь это 9 января 2003 г.

```

Можно запомнить, что тип переменной определяется тем значением, которое в нее записывается, например, с помощью оператора присваивания.

В связи с вышесказанным значение «Неопределено» используется для того, чтобы определить, было ли присвоено переменной какое-либо значение:

```

// Пытаемся восстановить значение сохраненное в предыдущем
// сеансе работы пользователя:
А=ВосстановитьЗначение(«А»);
Если А=Неопределено Тогда
// присвоим переменной начальное значение, т.к.
// предыдущее значение не было сохранено
А=1;

```

В этом примере функция глобального контекста **ВосстановитьЗначение** пытается восстановить значение «А», которое, возможно, было предварительно сохранено с помощью процедуры **СохранитьЗначение** в предыдущем сеансе работы пользователя.

Но вернемся к нашей задаче.

Для формирования строки с длительностью интервала нам придется написать текст функции **ИнтервалСтр(Начало,Конец)**, которая должна возвращать строку с интервалом между значениями **Конец** и **Начало**.

Для работы с датами (и временем!) имеется большое количество процедур и функций. Есть даже функция **ТекущаяДата()**, которая возвращает текущее значение системных часов компьютера (дату и время).

Поэтому для решения задачи необходимо из этого значения типа «Дата» извлечь часы, минуты и секунды. В этом нам могут помочь функции:

- ✓ Час(...);
- ✓ Минута(...);
- ✓ Секунда(...).

Согласно структуре нашего модуля написать эту функцию мы сможем только в разделе процедур и функций:

```
//-----  
//Раздел процедур и функций:  
//ИнтервалСтр(Начало,Конец) — возвращает продолжительность  
//интервала времени строкой:  
Функция ИнтервалСтр(Начало,Конец)  
Интервал=Конец-Начало; //вычислим интервал в секундах  
Часов=Цел(Интервал/3600); //сколько это будет часов  
Время='00010101'+Интервал; //преобразуем Интервал к дате  
Минут=Минута(Время); //получим число минут и секунд  
Секунд=Секунда(Время);  
// напишем это все строкой  
Стр=«»+Часов+«ч»+Минут+«м»+Секунд+«с»;  
Возврат Стр;  
КонецФункции
```

Последнее, что осталось, это «поймать» и обработать событие *перед завершением работы системы*:

```
//ПередЗавершениемРаботыСистемы(Отказ) — запрашивает  
//подтверждение о выходе из системы  
Процедура ПередЗавершениемРаботыСистемы(Отказ)  
КонецРаботы=ТекущаяДата();  
Ответ=Вопрос(«Завершить работу системы?»+Символы.ПС+«(Вы работали »  
+ИнтервалСтр(НачалоРаботы,КонецРаботы)+«)»,  
РежимДиалогаВопрос.ДаНет,30);  
Отказ=(Ответ=КодВозвратаДиалога.Нет);  
КонецПроцедуры
```

В этом тексте мы обратим ваше внимание на три момента.

Для того чтобы текст сообщения располагался на двух строках, мы применили значение из системного набора значений Символы (Символы.ПС), которое соответствует символу перевода строки.

Для того чтобы ожидание ответа пользователя не было бесконечно долгим, мы использовали величину 30 (секунд) в качестве продолжительности ожидания ответа (третий параметр).

Для того чтобы система могла бы принять решение об отмене команды пользователя, мы должны установить формальный параметр «Отказ» в значение Истина, если пользователь явно нажал на кнопку «Нет», что обеспечивается выражением:

**Отказ = (Ответ = КодВозвратаДиалога.Нет);**

### А.8. Преобразование типов и ошибки времени выполнения

Если внимательно разобрать текст функции **ИнтервалСтр(Начало, Конец)**, то можно обратить внимание на то, что при описании заголовка функции мы нигде не указали ни тип возвращаемого функцией значения, ни типы формальных параметров.

Поэтому запомним, что функция (процедура) может принимать и возвращать в качестве формальных параметров в разные моменты времени значения разных типов. И тип возвращаемого функцией значения определяется типом выражения, записанного в операторе «Возврат».

Очевидно, что наша функция должна возвращать строку текста, так как мы будем отображать в окне именно строку. Поэтому необходимо обеспечить, чтобы возвращаемое значение имело бы тип Строка.

Если бы мы записали в операторе «Возврат» следующее выражение:

**Стр= Часов+«ч»+Минут+«м»+Секунд+«с»;**

**Возврат Стр;**

то мы получили бы ошибку времени выполнения (рис. А.6).

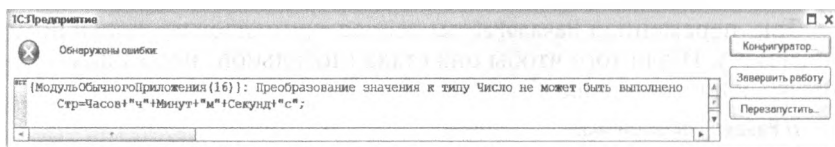


Рис. А.6. Сообщение об ошибке времени выполнения

Почему так получилось при вызове нашей функции? Потому что переменная «Часы» типа «Число», а строка «ч» — это Строка. Раз типы данных разные, то система выполняет  *неявное преобразование типа* . Но к какому типу?

Если типы данных в выражении отличаются друг от друга, то все операнды выражения  *всегда будут неявно преобразовываться к типу первого операнда* . Но строка «ч» не может быть преобразована в число, так как с точки зрения системы содержит букву, а это уже не число.

Для того чтобы выражение вычислялось правильно и операцию «+» система рассматривала как *сцепление строк*, необходимо, чтобы первым операндом была строка хотя бы и пустая:

**Стр = «» Часов + «ч» + Минут + «м» + Секунд + «с»; Возврат Стр;**

Тогда в выражении «» + Часов... переменная «Часов» будет неявно преобразована в строку.

По аналогии в тех случаях, когда необходимо обеспечить, чтобы все операнды в выражении были числовыми, выражение начинают с 0: **T=0+...**

Впрочем, для явного преобразования типа имеются встроенные функции: **Число(...)**, **Строка(...)** и **Дата(...)**.

### **А.9. Ключевое слово «Экспорт» и глобальный контекст**

Очевидно, что наша функция **ИнтервалСтр(...)** носит достаточно универсальный характер и может возникнуть необходимость применить ее вызов в других модулях конфигурации. Но сейчас эта функция принадлежит локальному контексту Модуля приложения, что делает невозможным ее использование в других модулях.

Для того чтобы эта функция попала в глобальный контекст, необходимо в ее заголовке написать добавочное ключевое слово «Экспорт»:

**Функция ИнтервалСтр(Начало,Конец) Экспорт**

**КонецФункции**

Теперь вызов этой функции можно будет сделать везде, где в системе допускается запись выражений на встроенном языке системы.

Все вышесказанное относится и к переменным.

Так, переменная **НачалоРаботы** сейчас принадлежит локальному контексту. И для того чтобы она стала глобальной, необходимо добавить ключевое слово «Экспорт»:

**// Раздел переменных-----**

**Перем НачалоРаботы Экспорт;**

### **А.10. Общие модули**

В этом разделе мы рассмотрим Общий модуль, который будет содержать примеры написания вспомогательных процедур и функций.

#### **А.10.1. Назначение Общих модулей**

Если в окне «Конфигурация» раскрыть ветвь «Общие модули», то среди общих объектов конфигурации можно увидеть отдельную ветвь для общих модулей (рис. А.7).

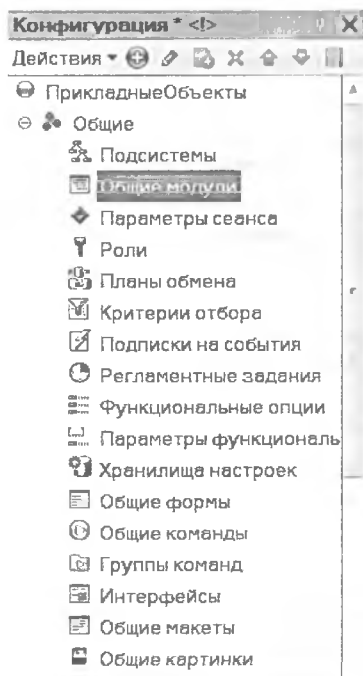


Рис. А.7. Общие модули конфигурации

Общие модули используются для размещения процедур и функций, вызов которых предполагается из других модулей конфигурации. Как было отмечено выше, такие процедуры и функции должны иметь добавочное ключевое слово «Экспорт».

Общие модули конфигурации существенно образом отличаются от большинства остальных модулей своим назначением — они предназначены для размещения общих процедур и функций, вызов которых предполагается делать из разных модулей конфигурации.

Вследствие вышеизложенного факта общий модуль не может содержать процедур обработчиков событий. И его структура допускает наличие только раздела процедур и функций. В общих модулях отсутствуют разделы переменных и раздел основной программы.

Поэтому обычно в Модуле приложения размещают только процедуры — обработчики событий, а общие процедуры и функции выносят в какой-либо общий модуль.

#### ***А.10.2. Добавление Общего модуля в Конфигурацию***

Добавим в нашу конфигурацию новый Общий модуль, выполнив следующее упражнение:

## Упражнение № А.4

Добавьте в конфигурацию новый общий модуль со следующими свойствами:

Свойство	Значение
Имя	<b>ОбщийМодульСлужебныхФункций</b>
Синоним	<b>Общий модуль служебных функций</b>
Комментарий	<b>Содержит вспомогательные процедуры и функции</b>
Глобальный	<input checked="" type="checkbox"/>

Перенесите, используя Буфер обмена, в этот модуль текст функции **ИнтервалСтр(...)** из Модуля приложения. Проверьте работу конфигурации.

Если Вы все сделали правильно, то текст этого общего модуля будет выглядеть следующим образом.

### ЛИСТИНГ П\_А.3

```
//Общий модуль служебных функций
//ИнтервалСтр(Начало,Конец) — возвращает продолжительность
//интервала времени строкой:
Функция ИнтервалСтр(Начало,Конец) Экспорт
Интервал=Конец-Начало;//вычислим интервал в секундах
Часов=Цел(Интервал/3600);//сколько это будет часов
Время='00010101'+Интервал;//преобразуем Интервал к дате
Минут=Минута(Время);//получим число минут и секунд
Секунд=Секунда(Время);
//напишем это все строкой
Стр=«»+Часов+«ч»+Минут+«м»+Секунд+«с»;
Возврат Стр;
КонецФункции
```

Обратите внимание на то, что теперь текст Модуля приложения в разделе процедур и функций содержит только процедуры обработки событий.

### ЛИСТИНГ П\_А.4

```
// Модуль приложения
//-----
// Раздел переменных:
Перем НачалоРаботы Экспорт;
//-----
//Раздел процедур и функций:
//Обработчики событий:
```

```

//ПриНачалеРаботыСистемы() — проверяет время входа в систему
Процедура ПриНачалеРаботыСистемы()
    ВремяЧас=Час(ТекущаяДата());
    Если (ВремяЧас>=13) и (ВремяЧас<14) Тогда
        Ответ=Вопрос(«Сейчас не рабочее время!»+Символы.Таб+
«Будете работать?»,РежимДиалогаВопрос.ДаНет);
        Если Ответ=КодВозвратаДиалога.Нет Тогда
            ПрекратитьРаботуСистемы();
            КонецЕсли;
        КонецЕсли;
КонецПроцедуры
//ПередЗавершениемРаботыСистемы(Отказ) — запрашивает
// подтверждение о выходе из системы
Процедура ПередЗавершениемРаботыСистемы(Отказ)
    КонецРаботы=ТекущаяДата();
    Ответ=Вопрос(«Завершить работу системы?»+Символы.ПС+«(Вы работали»
+ИнтервалСтр(НачалоРаботы,КонецРаботы)+«)»,РежимДиалогаВопрос.
ДаНет,30);
    Отказ=(Ответ=КодВозвратаДиалога.Нет);
КонецПроцедуры
//-----
// раздел основной программы:
НачалоРаботы = ТекущаяДата();

```

### А.11. Самостоятельная работа

Для закрепления материала этого раздела выполните самостоятельно следующие упражнения.

#### Упражнение № А.5

Сделайте так, чтобы конфигурация блокировала вход пользователя в режиме «1С:Предприятие» без выдачи каких-либо сообщений и открытия основного окна приложения после 17 часов текущего времени.

Какую процедуру — обработчик событий вы выберете для выполнения этого упражнения и почему?

#### Упражнение № А.6

Сделайте так, чтобы при запуске ИБ в режиме «1С:Предприятие» конфигурация выдавала в окне «Служебные сообщения» следующее сообщение типа:

Здравствуйте! Последний раз вы работали 23ч36м55с тому назад.

*Совет:* для сохранения и восстановления момента предыдущего выхода из системы между сеансами работы пользователя можно использовать методы глобального контекста **СохранитьЗначение(...)** и **ВосстановитьЗначение()** соответственно.

---

Ответьте для себя на следующие вопросы:

- ✓ Какую процедуру — обработчик события вы выберете для сохранения момента завершения работы с программой и почему?
- ✓ В какую процедуру следует поместить текст выдачи сообщения?
- ✓ Какую системную процедуру для выдачи сообщения Вы применили и почему?

### **А.12. Что мы узнали**

В этом приложении мы рассмотрели основные элементы встроенного языка.

Мы выяснили, что программные модули в конфигурации предназначены для написания алгоритмов обработки системных событий, которые возникают в системе в результате как интерактивных действий пользователя, так и выполнения алгоритмов, написанных на встроенном языке системы.

Обработку событий мы подробно рассмотрели на примере использования модуля приложения, который отвечает в конфигурации за весь сеанс работы с пользователем.

Мы также рассмотрели такие новые (по сравнению с системой «1С:Предприятие 7.7») элементы встроенного языка, как *системные перечисления* и *системные наборы значений*, и их место в системе типов данных встроенного языка системы. Мы также рассмотрели некоторые *примитивные типы данных* и преобразование типов данных при вычислении выражений.

Модуль приложения, как и любой другой модуль, имеет определенную структуру. В этом модуле вы можете использовать все три раздела (переменных, процедур и функций, основной программы).

Мы также рассмотрели разницу между Модулем приложения и Общими модулями в конфигурации их назначении и, как следствие, структуры.

Ключевое слово «Экспорт», которое можно применять в Модуле приложения и в общих модулях конфигурации, позволяет внести в глобальный контекст переменные, процедуры и функции, объявленные в этом модуле, если предполагается обращение к ним из других модулей конфигурации.

Архив ИБ, являющейся результатом выполнения упражнений Приложения А, вы можете найти в электронной части.



## ПРИЛОЖЕНИЕ В

### ПОСТРОЕНИЕ ОТЧЕТА БЕЗ ИСПОЛЬЗОВАНИЯ МАКЕТА

Этот раздел полностью предназначен для самостоятельного изучения.

Для удобства рассмотрения дальнейшего материала, а также для закрепления основной части курса создайте новую информационную базу, выполнив упражнение.

#### Упражнение № В.1

---

Создайте новую информационную базу.

Дайте ей название: «Приложение В» и расположите ее в каталоге C:\PO\Appendix\B.

Используя пункт меню «Администрирование — Загрузить информационную базу...», загрузите ИБ из файла, который расположен в электронной части.

---

Обратите внимание, что после загрузки ИБ Конфигуратор закончил свою работу — это его нормальное поведение, которое описано в документации к программе.

Формировать отчет можно и без использования макета. В этом случае необходимо, как и раньше, сначала создать объект типа **ТабличныйДокумент**, а затем обращаться к его отдельным ячейкам через метод **Область**:

```
ТабДок = Новый ТабличныйДокумент();
Область = ТабДок.Область(...);
//действия по заполнению свойств конкретной области
Область = ТабДок.Область(...);
//действия по заполнению свойств конкретной области
//Показать результат работы отчета
ТабДок.Показать(...);
```

Поскольку изначально табличный документ пуст, то в нем нет никаких именованных областей, и, следовательно, к отдельным ячейкам можно обратиться, только используя их адрес.

Обратиться к группе ячеек можно путем явного указания номеров строк и столбцов:

```
Область=ТабДок.Область(НомерПервойСтроки,НомерПервогоСтолбца,НомерПоследнейСтроки,НомерПоследнегоСтолбца);
```

Если необходимо обратиться к конкретной ячейке, то тогда указываются только номера первой строки и столбца:

```
Область=ТабДок.Область(НомерСтроки,НомерСтолбца);
```

Получив ссылку на область (тип значения — ОбластьЯчеекТабличногоДокумента), мы можем обратиться к атрибутам и методам этого объекта с целью заполнения свойств ячеек, составляющих область.

## Упражнение № В.2

Внесите изменения в текст процедуры **ПечатьКурсаУЕ()** из модуля формы списка регистра сведений «Курсы» таким образом, чтобы результирующий табличный документ формировался без использования макета.

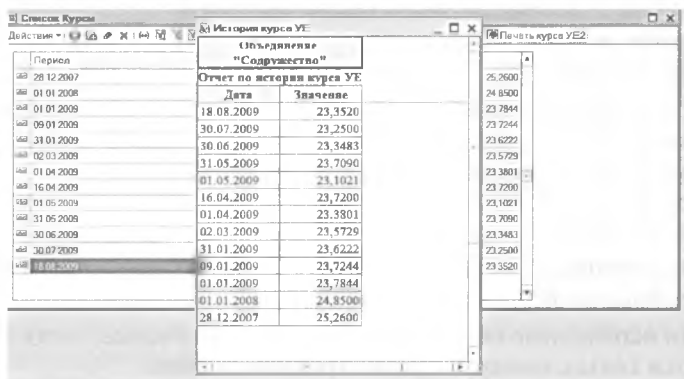
Для облегчения внесения изменений новый текст процедуры приведен чуть ниже в печатной форме. Однако мы рекомендуем все же набрать его вручную, для того чтобы получить навыки создания подобных программ и приобрести более уверенные навыки отладки модулей в системе «1С:Предприятие».

Самостоятельно изучите, используя документацию или Синтаксис-Помощник, объекты следующих типов:

- ✓ **ОбластьЯчеекТабличногоДокумента;**
- ✓ **Системное перечисление ТипРазмещенияТекстаТабличногоДокумента;**
- ✓ **Шрифт;**
- ✓ **Системные перечисления ВертикальноеПоложение и ГоризонтальноеПоложение;**
- ✓ **Линия;**
- ✓ **Цвет;**

После завершения кодирования текста обязательно протестируйте работу процедуры в режиме «1С:Предприятие».

В результате выполнения упражнения № В.2 окно табличного документа примет вид, изображенный на рис. В.1.



История курса УЕ	
Сводные данные "Содержание"	
Отчет по истории курса УЕ	
Дата	Значение
18.08.2009	23,3520
30.07.2009	23,2500
30.06.2009	23,3483
31.05.2009	23,7090
01.05.2009	23,1021
16.04.2009	23,7200
01.05.2009	23,7200
01.04.2009	23,3801
02.03.2009	23,5729
30.06.2009	23,5729
31.01.2009	23,6222
09.01.2009	23,7244
01.01.2009	23,7844
01.01.2008	24,8500
28.12.2007	25,2600

Рис. В.1. Результат выполнения упражнения № В.2

Приведем текст процедуры `ПечатьКурсаУЕ()`, которая не использует макет для построения отчета и является функциональным аналогом старого текста процедуры, который использовал макет для построения отчета.

#### ЛИСТИНГ П\_В.1

```
ПечатьКурсаУЕ(): //ПечатьКурсаУЕ()2 — печать курса УЕ
// формирует отчет без использования макета
Процедура ПечатьКурсаУЕ2() Экспорт
// Подготовительные действия:
// Получаем ссылку на необходимые шрифты
ШрифтTimes11Жирный = Новый Шрифт(«Times New Roman», 11, Истина);
ШрифтTimes11 = Новый Шрифт(«Times New Roman», 11, Ложь);
// Получаем ссылку на тип линии
ЛинияСплошная = Новый Линия(ТипЛинииЯчейкиТабличногоДокумента.
Сплошная,1);

// основной алгоритм
ТабДок = Новый ТабличныйДокумент();

//Устанавливаем ширину 1-й и 2-й колонок
ТабДок.Область(1,,1).ШиринаКолонки = 13;
ТабДок.Область(2,,2).ШиринаКолонки = 13;

//Формируем область заголовка
Область = ТабДок.Область(1,1,1,2);
Область.Объединить();
Область.РазмещениеТекста = ТипРазмещенияТекстаТабличногоДокумента.
Переносить;
Область.Текст = Константы.НаименованиеОрганизации.Получить();
Область.Шрифт = ШрифтTimes11Жирный;
Область.ВертикальноеПоложение = ВертикальноеПоложение.Центр;
Область.ГоризонтальноеПоложение = ГоризонтальноеПоложение.Центр;

//Формируем область шапки
Область = ТабДок.Область(2,1,2,2);
Область.Объединить();
Область.Текст = «Отчет по истории курса УЕ»;
Область.Шрифт = ШрифтTimes11Жирный;
Область.ВертикальноеПоложение = ВертикальноеПоложение.Центр;
Область.ГоризонтальноеПоложение = ГоризонтальноеПоложение.Центр;

ТабДок.Область(3,1).Текст = «Дата»;
```

ТабДок.Область(3,2).Текст = «Значение»;  
Область = ТабДок.Область(3,1,3,2);  
Область.ВертикальноеПоложение = ВертикальноеПоложение.Центр;  
Область.ГоризонтальноеПоложение = ГоризонтальноеПоложение.Центр;  
Область.Шрифт = ШрифтTimes11Жирный;  
Область.ГраницаСверху = ЛинияСплошная;  
Область.ГраницаСнизу = ЛинияСплошная;  
Область.ГраницаСлева = ЛинияСплошная;  
Область.ГраницаСправа = ЛинияСплошная;  
Область.ЦветФона = Новый Цвет(234,229,216);

РегСведКурсы = РегистрыСведений.Курсы;  
НабКурсы = РегСведКурсы.СоздатьНаборЗаписей();  
НабКурсы.Прочитать();  
к = НабКурсы.Количество();  
Для n = 1 по НабКурсы.Количество() Цикл  
ТекСтрока = НабКурсы[к-n];

ТабДок.Область(3+n,1).Текст = Формат(ТекСтрока.Период,«ДФ=dd.ММ.  
уууу»);

ТабДок.Область(3+n,2).Текст = Формат(ТекСтрока.Курс,«ЧЦ=10; ЧДЦ=4»);

КонецЦикла;

// Устанавливаем выравнивание для всех значений курсов  
Область = ТабДок.Область(4,2,3+НабКурсы.Количество(),2);  
Область.ГоризонтальноеПоложение = ГоризонтальноеПоложение.Право;

// Устанавливаем свойства ячеек всех строк, содержащих данные  
Область = ТабДок.Область(4,1,3+НабКурсы.Количество(),2);  
Область.Шрифт = ШрифтTimes11;  
Область.ГраницаСверху = ЛинияСплошная;  
Область.ГраницаСнизу = ЛинияСплошная;  
Область.ГраницаСлева = ЛинияСплошная;  
Область.ГраницаСправа = ЛинияСплошная;

// Установка свойств табличного документа в целом

// перед его отображением:

ТабДок.ИмяПараметровПечати = «ПараметрыКурсаУЕ2»;  
ТабДок.ИмяСохраненияПоложенияОкна = «ОкноКурсаУЕ2»;  
ТабДок.ОтображатьЗаголовки = Ложь;  
ТабДок.ОтображатьСетку = Ложь;  
ТабДок.СохранятьСвойстваОтображения = Истина;  
ТабДок.ТолькоПросмотр = Истина;

**ТабДок.ФиксацияСверху = 3;**  
**ТабДок.Показать(«История курса УЕ»);**  
**КонецПроцедуры**

## **В.1. ЧТО МЫ УЗНАЛИ**

В этом приложении мы рассмотрели основы формирования отчетов с использованием объекта «ТабличныйДокумент» без использования макета (исходного табличного документа).

Архив ИБ, являющейся результатом выполнения упражнений приложения В, вы можете найти в электронной части.

## **ПРИЛОЖЕНИЕ С**

### **МОБИЛЬНОЕ ПРИЛОЖЕНИЕ**

Одним из новшеств, представленных в «1С:Предприятие 8.3», является реализация мобильной платформы «1С:Предприятие», которая обеспечивает:

- ✓ разработку приложений (конфигураций) для мобильных устройств в среде конфигуратора с учетом ограничений мобильной платформы;
- ✓ генерацию мобильного приложения для записи на мобильное устройство;
- ✓ работу мобильных приложений в операционных системах iOS, Android OS;
- ✓ поддержку работы мобильных приложений в режиме offline и обмен данными с основным приложением (back office).

Средства разработки мобильных приложений и мобильная платформа «1С:Предприятие 8» для целей разработки включены в коммерческую поставку платформы «1С:Предприятие» и без дополнительной оплаты доступны разработчикам прикладных решений, правомерно владеющим дистрибутивом платформы «1С:Предприятие 8.3».

Документация по мобильной платформе для разработчиков включена в состав общей документации по платформе «1С:Предприятие 8.3».

Мобильные приложения разработки фирмы «1С» могут распространяться как свободно, так и на платной основе – через магазины App Store и Google Play.

Мобильные приложения, разработанные пользователями и партнерами на платформе «1С:Предприятие 8.3», в настоящее время разрешается распространять без оплаты фирме «1С», если количество распространяемых копий не превышает 50. Для распространения большего числа копий требуется согласование с фирмой «1С».

## С1. СОЗДАНИЕ ИНФОРМАЦИОННОЙ БАЗЫ

Для удобства ознакомления с мобильным приложением, а также для закрепления основной части курса создайте новую информационную базу, выполнив следующее упражнение.

### Упражнение № С.1

Создайте новую информационную базу.

Откройте конфигурацию. В свойствах конфигурации значение параметра «**Основной режим запуска**» оставьте в положении «**Управляемое приложение**» (рис. С.1). Измените «**Назначения использования**» на «**Мобильное устройство**». Сохраните и обновите конфигурацию.

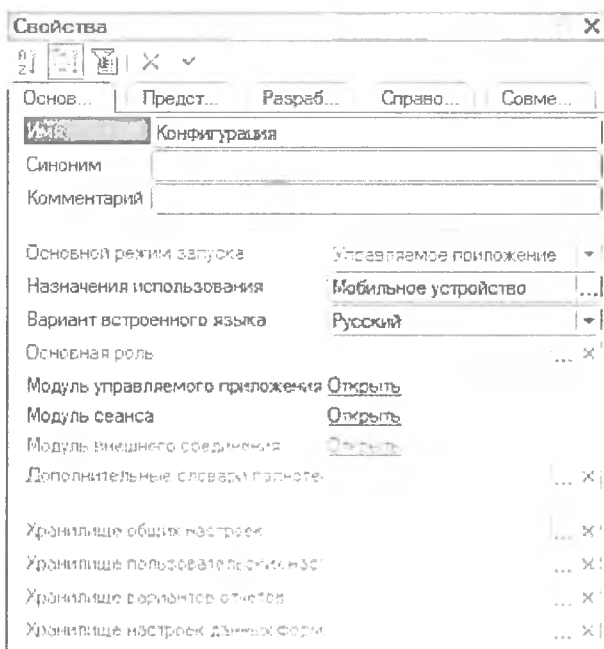


Рис. С.1. Окно свойств приложения

И здесь мы сразу обращаем внимание, что многие объекты конфигурации стали недоступны для использования. Мы уже не сможем использовать подсистемы, регламентные задания, XDTO-пакеты, Web-сервисы, отчеты, бизнес-процессы и многое другое (рис. С.2). Также многие процедуры и методы для некоторых объектов оказываются недоступны. Это стоит учитывать при разработке.

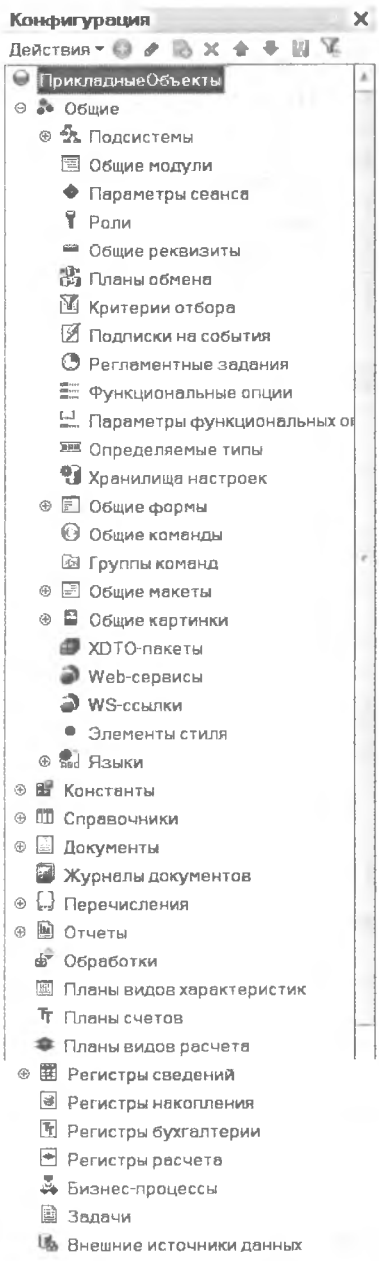


Рис. С.2. Доступные объекты конфигурации

## Упражнение № С.2

Самостоятельно создайте форму. На вкладке «Команды формы» добавьте новую команду. И перетащите эту команду на форму (рис. С.3).

Свойство	Значение
Имя	КнопкаПриветствия
Заголовок	Привет!
Подсказка	Кнопка приветствия

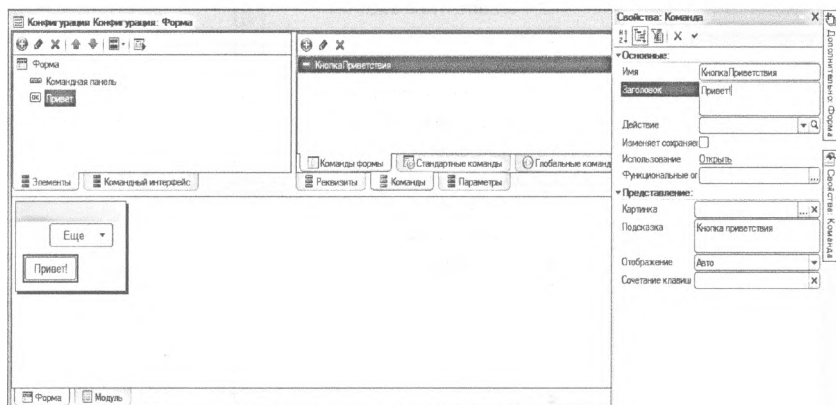


Рис. С.3. Добавление новой кнопки

## Упражнение № С.2 (продолжение)

Добавьте действие на клиенте к созданной команде, нажав на лупу в графе «Действие».

Создайте вызов диалогового окна предупреждение с текстом «Привет, Мир!».

Для надежности приводим текст процедуры **КнопкаПриветствия(Команда)**. (Сравните его со своим вариантом.)

### ЛИСТИНГ П\_С.1

```
&НаКлиенте
Процедура КнопкаПриветствия(Команда)
Предупреждение(«Привет, Мир!»)
КонецПроцедуры
```



## С.2. ВЫГРУЗКА ИНФОРМАЦИОННОЙ БАЗЫ НА СЕРВЕР

Чтобы запустить созданную информационную базу, необходимо выгрузку ИБ разместить на сервере.

Выгружаем разработанное мобильное приложение в xml-файл с помощью меню «**Конфигурация – Мобильное приложение – Записать в файл...**».

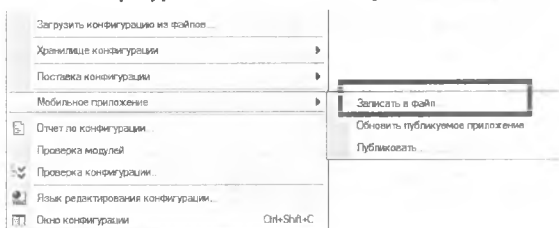


Рис. С.4. Запись выгрузки ИБ в файл

Далее необходимо создать сервер. (Для примера используется хатрр).

### Упражнение № С.3 (продолжение)

Запустите Apache и создайте на сервере папку «**mobile**» для выгрузки (C:\xampp\htdocs).

Скопируйте в созданную папку сформированный xml-файл (рис. С.5).

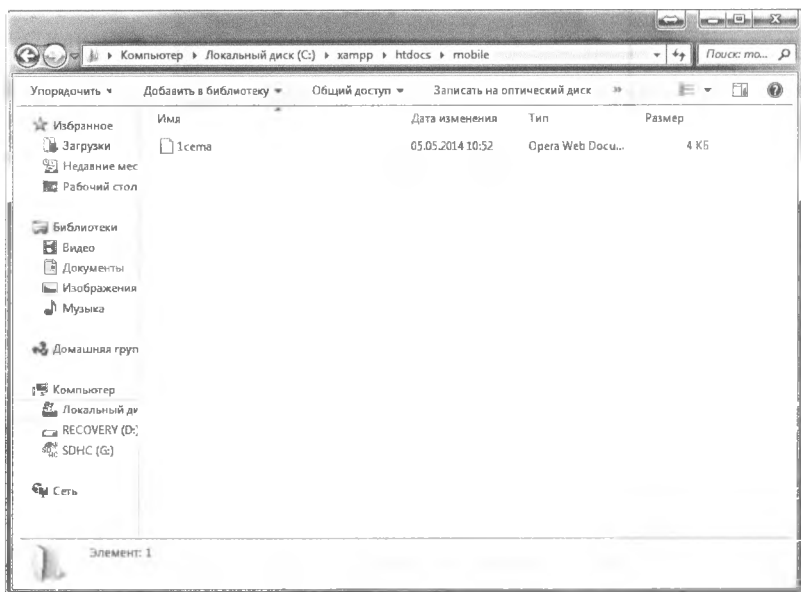


Рис. С.5. Добавление ИБ на сервер

Самостоятельно создайте сеть wi-fi с разрешением доступа к данным компьютера.

### С.3. ЗАПУСК ИБ НА СМАРТФОНЕ

Далее рассмотрен механизм запуска мобильного приложения с телефона на платформе Android. Перед началом работы с мобильным приложением необходимо подключить устройство к созданной сети. Установите на смартфон «1С:Предприятие 8». Запустите установленное приложение (рис. С.6).

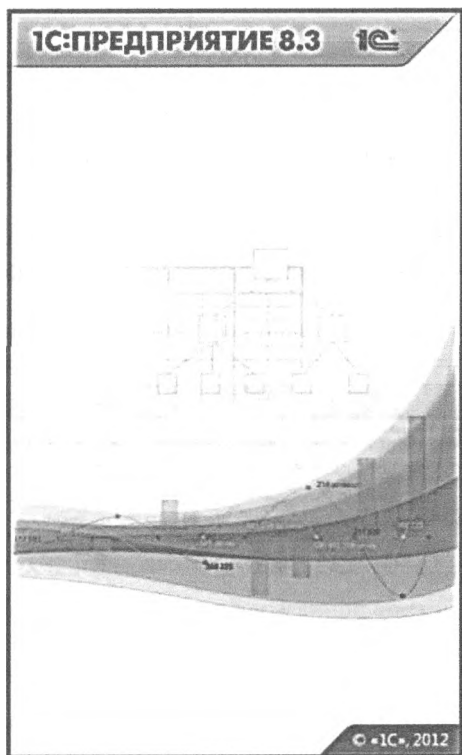


Рис. С.6. Запуск мобильного приложения

Далее необходимо добавить новую информационную базу на мобильное приложение. Для этого нажмите кнопку **«Добавить»**. На следующем шаге необходимо указать путь к xml-файлу на сервере.

Перед этим нам необходимо узнать ip-адрес компьютера. Чтобы это сделать, можно воспользоваться командой ipconfig в командной строке Windows (рис. С.7).

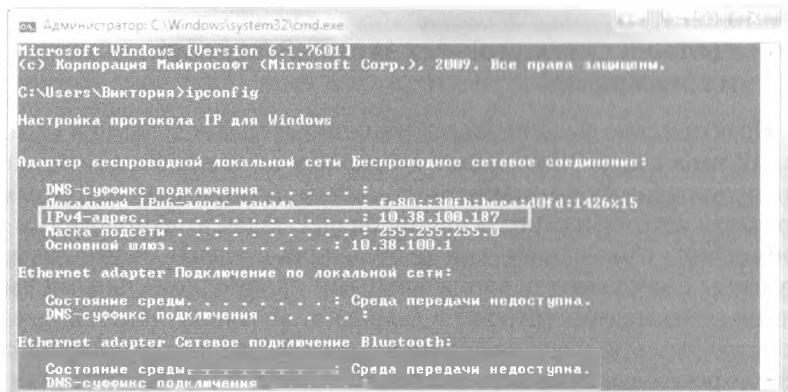


Рис. С.7. Определение ip-адреса

#### Упражнение № С.4

Самостоятельно укажите адрес к xml-файлу. В качестве примера можно использовать следующий адрес: <http://192.168.134.1/mobile/lcema.xml>.

Осталось лишь нажать кнопку «**Загрузить**».  
Запустите и проверьте работу ИБ.

## С.4. ЗАКЛЮЧЕНИЕ

На момент написания учебного пособия «мобильное приложение» является бета-версией, и вследствие этого возможны ошибки при загрузке ИБ на смартфон.

**Примечание.** Некоторые ошибки могут исчезнуть при повторном запуске приложения.

Пока новые возможности выглядят весьма «сыро»: ограничение функционала IC плюс большой размер и некоторые ошибки. Сама возможность написать программу для мобильной платформы на IC немного удивляет. С одной стороны, эта возможность пока больше похожа на игрушку, так как отсутствует возможность сделать что-то по-настоящему стоящее. Но с другой стороны, это явно большой шаг компании «iC» в сторону мобильности, и если данное направление будет активно развиваться, то это может принести немало пользы. Как пример — можно экипировать кладовщиков планшетами. Например, сейчас уже есть возможность использовать «бортовые» камеры, т.е. можно освободиться от устройств считывания и сканировать коды непосредственно с планшета. Также можно использовать геолокацию, таким образом, можно снабдить устройствами с разработанными мобильными приложениями водителей машин и посылать задания на перевозку или отслеживать маршрут автомобиля и время в движении.

## ПРИЛОЖЕНИЕ D

### СБОРНИК СИТУАЦИОННЫХ ЗАДАНИЙ (КЕЙСОВ) И УПРАЖНЕНИЙ

Практическое выполнение ситуационных заданий, задач и упражнений имеет целью овладение возможностями профессионально-ориентированных компьютерных систем, комплексов и программ в области автоматизации решения бизнес-задач и технологиями их применения в экономической деятельности студента. Решая задачи, студенты овладевают первоначальными профессиональными умениями и навыками, которые в дальнейшем закрепляются и совершенствуются при изучении специальных дисциплин, а также в процессе прохождения производственной практики. При формулировании ситуационных задач использовался так называемый метод кейсов. Метод кейсов (англ. *Case method*, кейс-метод, кейс-стади, *case-study*, метод конкретных ситуаций, метод ситуационного анализа) — техника обучения, использующая описание реальных экономических, социальных и бизнес-ситуаций. Обучающиеся должны исследовать ситуацию, разобраться в сути проблем, предложить возможные решения и выбрать лучшее из них. Кейсы основываются на реальном фактическом материале или же приближены к реальной ситуации.

#### ПЕРЕЧЕНЬ СИТУАЦИОННЫХ ЗАДАНИЙ (КЕЙСОВ)

##### Кейс 1

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации КЕЙС 1.
3. Создать подсистемы: *Основная деятельность, Зарплата*.
4. Создать объекты конфигурации *Справочники: Специалисты, Должности, Клиенты, Номенклатура услуг*.
5. Определить принадлежность справочников к подсистемам: *Клиенты* к подсистеме *Основная деятельность*; *Должности, Специалисты, Номенклатура услуг* к подсистеме *Зарплата*.
6. Добавить команду добавления нового элемента справочника в панель действий соответствующих подсистем.
7. В режиме «ИС:Предприятие» добавить в каждый справочник несколько элементов.
8. Создать объекты конфигурации *Документы: Оказание услуги*. Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Наименование, Количество, Цена, Сумма*. Определить доступность команды создания новых документов в соответствую-

ющих подсистемах. Автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества в строке. В режиме «ИС:Предприятие» создать несколько документов.

9. Добавить регистр накопления: *Оказанные услуги*. Определить доступность созданного регистра накопления из соответствующих подсистем. Создать измерения регистра. Сформировать движения регистра в процессе проведения документа. Добавить команду для открытия регистра накопления из формы документа.
10. Создать отчет: *Реестр документов, Оказание услуги*. Добавить ссылку на отчет в панель действий соответствующих подсистем.
11. Создать макет печатной формы документа. Создать команду документа с именем *Печать*. В модуль этой команды поместить обработчик, вызывающий процедуру печати документа. Команду *Печать* поместить в командную панель формы документа.

## КЕЙС 2

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации КЕЙС 2.
3. Создать подсистемы: *Продажи, Кадры*.
4. Создать объекты конфигурации *Справочники: Сотрудники, Контрагенты, Товары*.
5. Определить принадлежность справочников к подсистемам: *Сотрудники* к подсистеме *Кадры*; *Контрагенты, Товары* к подсистеме *Продажи*.
6. Добавить команду добавления нового элемента справочника в панель действий соответствующих подсистем.
7. В режиме «ИС:Предприятие» добавить в каждый справочник несколько элементов.
8. Создать объекты конфигурации *Документы: Приходная накладная, Расходная накладная*.
9. Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Наименование товара, Количество, Цена, Сумма*. Определить доступность команды создания новых документов в соответствующих подсистемах. Автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества в строке. В режиме «ИС:Предприятие» создать несколько документов.
10. Добавить регистр накопления: *Остатки товаров*. Определить доступность созданного регистра накопления из соответствующих подсистем. Создать измерения регистра. Сформировать движения регистра в процессе проведения документа. Добавить команду для открытия регистра накопления из формы документа.

11. Создать отчет: *Движение товаров*. Добавить ссылку на отчет в панель действий соответствующих подсистем.
12. Создать макет печатной формы документа. Создать команду документа с именем *Печать*. В модуль этой команды поместить обработчик, вызывающий процедуру печати документа. Команду *Печать* поместить в командную панель формы документа.

### КЕЙС 3

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации КЕЙС 3.
3. Создать подсистемы: *Оказание услуг, Бухгалтерия*.
4. Создать объекты конфигурации *Справочники: Сотрудники, Клиенты, Услуги*.
5. Определить принадлежность справочников к подсистемам: *Сотрудники* к подсистеме *Бухгалтерия*; *Клиенты, Услуги* к подсистемам *Бухгалтерия, Оказание услуг*.
6. Добавить команду добавления нового элемента справочника в панель действий соответствующих подсистем.
7. В режиме «1С:Предприятие» добавить в каждый справочник несколько элементов.
8. Создать объекты конфигурации *Документы: Оказание услуги*. Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Наименование услуги, Количество, Цена, Сумма*. Определить доступность команды создания новых документов в соответствующих подсистемах. Автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества в строке. В режиме «1С:Предприятие» создать несколько документов.
9. Добавить регистр накопления: *Оказанные услуги*. Определить доступность созданного регистра накопления из соответствующих подсистем. Создать измерения регистра. Сформировать движения регистра в процессе проведения документа. Добавить команду для открытия регистра накопления из формы документа.
10. Создать отчет: *Рейтинг услуг*. Добавить ссылку на отчет в панель действий соответствующих подсистем.
11. Создать макет печатной формы документа. Создать команду документа с именем *Печать*. В модуль этой команды поместить обработчик, вызывающий процедуру печати документа. Команду *Печать* поместить в командную панель формы документа.

### КЕЙС 4

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации КЕЙС 4.

3. Создать подсистемы: *Учет материалов, Зарплата*.
4. Создать объекты конфигурации *Справочники: Менеджеры, Покупатели, Номенклатура*.
5. Определить принадлежность справочников к подсистемам: *Менеджеры* к подсистемам *Зарплата, Учет материалов*; *Покупатели, Номенклатура* к подсистеме *Учет материалов*.
6. Добавить команду добавления нового элемента справочника в панель действий соответствующих подсистем.
7. В режиме «1С:Предприятие» добавить в каждый справочник несколько элементов.
8. Создать объекты конфигурации *Документы: Приходная накладная, Расходная накладная*. Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Наименование материала, Количество, Цена, Сумма*. Определить доступность команды создания новых документов в соответствующих подсистемах. Автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества в строке. В режиме «1С:Предприятие» создать несколько документов.
9. Добавить регистр накопления: *Остатки материалов*. Определить доступность созданного регистра накопления из соответствующих подсистем. Создать измерения регистра. Сформировать движения регистра в процессе проведения документов *Приходная накладная, Расходная накладная*. Добавить команду для открытия регистра накопления из формы документов.
10. Создать отчет: *Реестр документов, Приходная накладная, Движения материалов*. Добавить ссылку на отчет в панель действий соответствующих подсистем.
11. Создать макет печатной формы документа. Создать команду документа с именем *Печатать*. В модуль этой команды поместить обработчик, вызывающий процедуру печати документа. Команду *Печатать* поместить в командную панель формы документа.

## КЕЙС 5

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации КЕЙС 5.
3. Создать подсистемы: *Продажи, Учет материалов*.
4. Создать объекты конфигурации *Справочники: Продавцы, Контрагенты, Материалы*.
5. Определить принадлежность справочников к подсистемам: *Продавцы, Контрагенты* к подсистеме *Продажи*; *Контрагенты, Материалы* к подсистеме *Учет материалов*.
6. Добавить команду добавления нового элемента справочника в панель действий соответствующих подсистем.

7. В режиме «1С:Предприятие» добавить в каждый справочник несколько элементов.
8. Создать объекты конфигурации *Документы: Приходная накладная, Расходная накладная*. Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Наименование материала, Количество, Цена, Сумма*. Определить доступность команды создания новых документов в соответствующих подсистемах. Автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества в строке. В режиме «1С:Предприятие» создать несколько документов.
9. Добавить регистр накопления: *Остатки материалов*. Определить доступность созданного регистра накопления из соответствующих подсистем. Создать измерения регистра. Сформировать движения регистра в процессе проведения документов *Приходная накладная, Расходная накладная*. Добавить команду для открытия регистра накопления из формы документов.
10. Создать отчет: *Материалы*, показывающий приход, расход и остатки материалов. Добавить ссылку на отчет в панель действий соответствующих подсистем.
11. Создать макет печатной формы документов. Создать команду документа с именем *Печать*. В модуль этой команды поместить обработчик, вызывающий процедуру печати документа. Команду *Печать* поместить в командную панель формы документа.

## КЕЙС 6

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации КЕЙС 6.
3. Создать подсистемы: *Основная деятельность, Учет товаров*.
4. Создать объекты конфигурации *Справочники: Сотрудники, Контрагенты, Товары*.
5. Определить принадлежность справочников к подсистемам: *Сотрудники, Контрагенты, Товары* к подсистеме *Основная деятельность*; *Контрагенты, Товары* к подсистеме *Учет товаров*.
6. Добавить команду добавления нового элемента справочника в панель действий соответствующих подсистем.
7. В режиме «1С:Предприятие» добавить в каждый справочник несколько элементов.
8. Создать объекты конфигурации *Документы: Приходная накладная, Расходная накладная*. Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Товар, Количество, Цена, Сумма*. Опреде-



лить доступность команды создания новых документов в соответствующих подсистемах. Автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества в строке. В режиме «ИС:Предприятие» создать несколько документов.

9. Добавить регистр накопления: *Остатки товаров*. Определить доступность созданного регистра накопления из соответствующих подсистем. Создать измерения регистра. Сформировать движения регистра в процессе проведения документов. Добавить команду для открытия регистра накопления из формы документа.
10. Создать отчет: *Остатки товаров*. Добавить ссылку на отчет в панель действий соответствующих подсистем.
11. Создать макет печатной формы документа. Создать команду документа с именем *Печать*. В модуль этой команды поместить обработчик, вызывающий процедуру печати документа. Команду *Печать* поместить в командную панель формы документа.

## КЕЙС 7

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации КЕЙС 7.
3. Создать подсистемы: *Продажи, Кадры*.
4. Создать объекты конфигурации *Справочники: Сотрудники, Клиенты, Услуги*.
5. Определить принадлежность справочников к подсистемам: *Сотрудники* к подсистемам *Кадры, Продажи; Клиенты, Услуги* к подсистеме *Продажи*.
6. Добавить команду добавления нового элемента справочника в панель действий соответствующих подсистем.
7. В режиме «ИС:Предприятие» добавить в каждый справочник несколько элементов.
8. Создать объекты конфигурации *Документы: Оказание услуги*. Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Наименование, Количество, Цена, Сумма*. Определить доступность команды создания новых документов в соответствующих подсистемах. Автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества в строке. В режиме «ИС:Предприятие» создать несколько документов.
9. Добавить регистр накопления: *Оказанные услуги*. Определить доступность созданного регистра накопления из соответствующих подсистем. Создать измерения регистра. Сформировать движения

- регистра в процессе проведения документа. Добавить команду для открытия регистра накопления из формы документа.
10. Создать отчет: *Выручка сотрудников, Реестр документов, Оказание услуг*. Добавить ссылку на отчет в панель действий соответствующих подсистем.
  11. Создать макет печатной формы документа. Создать команду документа с именем *Печать*. В модуль этой команды поместить обработчик, вызывающий процедуру печати документа. Команду *Печать* поместить в командную панель формы документа.

## КЕЙС 8

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации КЕЙС 8.
3. Создать подсистемы: *Оказание услуг, Зарплата*.
4. Создать объекты конфигурации *Справочники: Специалисты, Должности, Клиенты, Услуги*.
5. Определить принадлежность справочников к подсистемам: *Клиенты, Услуги* к подсистеме *Оказание услуг*; *Специалисты, Должности*, к подсистеме *Зарплата*.
6. Добавить команду добавления нового элемента справочника в панель действий соответствующих подсистем.
7. В режиме «1С:Предприятие» добавить в каждый справочник несколько элементов.
8. Создать объекты конфигурации *Документы: Оказание услуги*. Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Наименование, Количество, Цена, Сумма*. Определить доступность команды создания новых документов в соответствующих подсистемах. Автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества в строке. В режиме «1С:Предприятие» создать несколько документов.
9. Добавить регистр накопления: *Оказанные услуги*. Определить доступность созданного регистра накопления из соответствующих подсистем. Создать измерения регистра. Сформировать движения регистра в процессе проведения документа. Добавить команду для открытия регистра накопления из формы документа.
10. Создать отчет: *Перечень услуг, Рейтинг услуг*. Добавить ссылку на отчет в панель действий соответствующих подсистем.
11. Создать макет печатной формы документа. Создать команду документа с именем *Печать*. В модуль этой команды поместить обработчик, вызывающий процедуру печати документа. Команду *Печать* поместить в командную панель формы документа.

## КЕЙС 9

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации КЕЙС 9.
3. Создать подсистемы: *Учет материалов, Зарплата*.
4. Создать объекты конфигурации *Справочники: Продавцы, Контрагенты, Материалы*.
5. Определить принадлежность справочников к подсистемам: *Продавцы* к подсистемам *Зарплата, Учет материалов*; *Контрагенты, Материалы* к подсистеме *Учет материалов*.
6. Добавить команду добавления нового элемента справочника в панель действий соответствующих подсистем.
7. В режиме «1С:Предприятие» добавить в каждый справочник несколько элементов.
8. Создать объекты конфигурации *Документы: Приходная накладная, Расходная накладная*. Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Материалы, Количество, Цена, Сумма*. Определить доступность команды создания новых документов в соответствующих подсистемах. Автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества в строке. В режиме «1С:Предприятие» создать несколько документов.
9. Добавить регистр накопления: *Остатки материалов*. Определить доступность созданного регистра накопления из соответствующих подсистем. Создать измерения регистра. Сформировать движения регистра в процессе проведения документа. Добавить команду для открытия регистра накопления из формы документа.
10. Создать отчет: *Материалы, Выручка продавцов*. Добавить ссылку на отчет в панель действий соответствующих подсистем.
11. Создать макет печатной формы документа. Создать команду документа с именем *Печать*. В модуль этой команды поместить обработчик, вызывающий процедуру печати документа. Команду *Печать* поместить в командную панель формы документа.

## КЕЙС 10

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации КЕЙС 10.
3. Создать подсистемы: *Продажи, Бухгалтерия*.
4. Создать объекты конфигурации *Справочники: Менеджеры, Покупатели, Номенклатура*.
5. Определить принадлежность справочников к подсистемам: *Менеджеры* к подсистемам *Продажи, Бухгалтерия*; *Покупатели, Номенклатура* к подсистеме *Продажи*.

6. Добавить команду добавления нового элемента справочника в панель действий соответствующих подсистем.
7. В режиме «1С:Предприятие» добавить в каждый справочник несколько элементов.
8. Создать объекты конфигурации *Документы: Приходная накладная, Расходная накладная*. Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Наименование, Количество, Цена, Сумма*. Определить доступность команды создания новых документов в соответствующих подсистемах. Автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества в строке. В режиме «1С:Предприятие» создать несколько документов.
9. Добавить регистр накопления: *Остатки материалов*. Определить доступность созданного регистра накопления из соответствующих подсистем. Создать измерения регистра. Сформировать движения регистра в процессе проведения документов. Добавить команду для открытия регистра накопления из формы документа.
10. Создать отчет: *Материалы*. Добавить ссылку на отчет в панель действий соответствующих подсистем.
11. Создать макет печатной формы документа. Создать команду документа с именем *Печать*. В модуль этой команды поместить обработчик, вызывающий процедуру печати документа. Команду *Печать* поместить в командную панель формы документа.

## УПРАЖНЕНИЯ

### Упражнение № D.1

Создать двухуровневый справочник «Контрагенты» следующей структуры:

Имя поля	Тип данных	Длина
<b>Код</b>	Целое число	5
<b>Наименование</b>	Символьный	25
<b>Полное наименование</b>	Символьный	30

В справочник ввести группу «Поставщики», в группу поставщики ввести элементы «Поставщик <НЗ>» и «Поставщик <НЗ>+1». Здесь и далее <НЗ> — номер задания. Например, если НЗ=1, то нужно ввести Поставщик 1 и Поставщик.

**Упражнение № D.2**

Создать линейный справочник «Договоры», подчиненный справочнику «Контрагенты» следующей структуры:

Имя поля	Тип данных	Длина
Код	Целое число	5
Наименование	Символьный	25
Полное наименование	Символьный	30

В справочник ввести элементы для контрагента «Поставщик <НЗ>» — договор <НЗ>, для контрагента «Поставщик <НЗ>+1» — договор <НЗ>.

**Упражнение № D.3**

Создать линейный справочник «Склады» следующей структуры:

Имя поля	Тип данных	Длина
Код	Целое число	5
Наименование	Символьный	25

В справочник ввести элементы «Склад материалов 1» и «Склад материалов 2».

**Упражнение № D.4**

Создать перечисление «Единицы измерения» с элементами «Кг» и «Литр».

**Упражнение № D.5**

Создать линейный справочник «Материалы» следующей структуры:

Имя поля	Тип данных	Длина
Код	Целое число	5
Наименование	Символьный	25
Полное наименование	Ссылка на перечисление «Единицы измерения»	

В справочник ввести элементы «Материал 1» (единица измерения — кг), «Материал 2» (единица измерения — кг) и «Материал 3» (единица измерения — литр).

## Упражнение № D.6

Создать линейный справочник «Статьи затрат» следующей структуры:

Имя поля	Тип данных	Длина
Код	Целое число	5
Наименование	Символьный	25

В справочник ввести элемент «Стоимость материалов».

## Упражнение № D.7

Разработать приложение согласно постановке задачи.

Разработать презентацию с демонстрацией последовательности выполняемых действий и описанием используемых объектов.

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации ВАРИАНТ 1.
3. Создать подсистемы: *Основная деятельность, Зарплата.*

4. Создать объекты конфигурации *Справочники: Специалисты, Должности, Клиенты, Номенклатура услуг.* Определить принадлежность справочников к подсистемам: *Клиенты* к подсистеме *Основная деятельность*; *Должности, Специалисты, Номенклатура услуг* к подсистеме *Зарплата.*

5. Добавить команду добавления нового элемента справочника в панель действий соответствующих подсистем. В режиме «1С:Предприятие» добавить в каждый справочник несколько элементов.

## Упражнение № D.8

Разработать приложение согласно постановке задачи.

Разработать презентацию с демонстрацией последовательности выполняемых действий и описанием используемых объектов.

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации ВАРИАНТ 2.
3. Создать подсистемы: *Основная деятельность, Зарплата.*

4. Создать объект конфигурации *Документы: Оказание услуги.* Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Наименование, Количество, Цена, Сумма.* Определить доступность команды создания новых документов в соответствующих подсистемах.

5. Автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества в строке. В режиме «1С:Предприятие» создать несколько документов.

## Упражнение № D.9

---

**Разработать приложение согласно постановке задачи.**

**Разработать презентацию с демонстрацией последовательности выполняемых действий и описанием используемых объектов**

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации ВАРИАНТ 3.
3. Создать подсистемы: *Основная деятельность, Зарплата.*
4. Создать объект конфигурации *Регистр накопления: Оказанные услуги*. Определить доступность созданного регистра накопления из соответствующих подсистем. Создать измерения регистра.
5. Сформировать движения регистра в процессе проведения документа. Добавить команду для открытия регистра накопления из формы документа.

## Упражнение № D.10

---

**Разработать приложение согласно постановке задачи.**

**Разработать презентацию с демонстрацией последовательности выполняемых действий и описанием используемых объектов.**

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации ВАРИАНТ 4.
3. Создать подсистемы: *Продажи, Кадры.*
4. Создать объекты конфигурации *справочники: Сотрудники, Контрагенты, Товары*. Определить принадлежность справочников к подсистемам: *Сотрудники* к подсистеме *Кадры*; *Контрагенты, Товары* к подсистеме *Продажи*.
5. Добавить команду добавления нового элемента справочника в панель действий соответствующих подсистем.
6. В режиме «1С:Предприятие» добавить в каждый справочник несколько элементов.

## Упражнение № D.11

---

**Разработать приложение согласно постановке задачи.**

**Разработать презентацию с демонстрацией последовательности выполняемых действий и описанием используемых объектов.**

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации ВАРИАНТ 5.
3. Создать подсистемы: *Продажи, Кадры.*
4. Создать объекты конфигурации *Документы: Приходная накладная, Расходная накладная*. Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Наименование товара, Количество, Цена, Сумма*. Определить доступность команды создания новых документов в соответствующих подсистемах.

5. Автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества в строке. В режиме «1С:Предприятие» создать несколько документов.

#### Упражнение № D.12

---

---

**Разработать приложение согласно постановке задачи.**

**Разработать презентацию с демонстрацией последовательности выполняемых действий и описанием используемых объектов.**

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации ВАРИАНТ 6.
3. Создать подсистемы: *Продажи, Кадры*.
4. Добавить регистр накопления: *Остатки товаров*. Определить доступность созданного регистра накопления из соответствующих подсистем. Создать измерения регистра. Сформировать движения регистра в процессе проведения документа. Добавить команду для открытия регистра накопления из формы документа.
5. Создать отчет: *Движение товаров*. Добавить ссылку на отчет в панель действий соответствующих подсистем.

#### Упражнение № D.13

---

---

**Разработать приложение согласно постановке задачи.**

**Разработать презентацию с демонстрацией последовательности выполняемых действий и описанием используемых объектов.**

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации ВАРИАНТ 7.
3. Создать подсистемы: *Продажи, Кадры*.
4. Добавить регистр накопления: *Остатки товаров*. Определить доступность созданного регистра накопления из соответствующих подсистем. Создать измерения регистра. Сформировать движения регистра в процессе проведения документа. Добавить команду для открытия регистра накопления из формы документа.
5. Создать макет печатной формы документа. Создать команду документа с именем *Печать*. В модуль этой команды поместить обработчик, вызывающий процедуру печати документа. Команду *Печать* поместить в командную панель формы документа.

#### Упражнение № D.14

---

---

**Разработать приложение согласно постановке задачи.**

**Разработать презентацию с демонстрацией последовательности выполняемых действий и описанием используемых объектов.**

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации ВАРИАНТ 8.



3. Создать подсистемы: *Учет материалов, Зарплата.*

4. Создать объекты конфигурации *Справочники: Менеджеры, Покупатели, Номенклатура.* Определить принадлежность справочников к подсистемам: *Менеджеры* к подсистемам *Зарплата, Учет материалов; Покупатели, Номенклатура* к подсистеме *Учет материалов.*

5. Добавить команду добавления нового элемента справочника в панель действий соответствующих подсистем.

6. В режиме «1С:Предприятие» добавить в каждый справочник несколько элементов.

#### Упражнение № D.15

---

**Разработать приложение согласно постановке задачи.**

**Разработать презентацию с демонстрацией последовательности выполняемых действий и описанием используемых объектов.**

1. Создать новую информационную базу.

2. В режиме Конфигуратор задать имя конфигурации ВАРИАНТ 9.

3. Создать подсистемы: *Учет материалов, Зарплата.*

4. Создать объекты конфигурации *Документы: Приходная накладная, Расходная накладная.* Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Наименование материала, Количество, Цена, Сумма.* Определить доступность команды создания новых документов в соответствующих подсистемах.

5. Автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества в строке. В режиме «1С:Предприятие» создать несколько документов.

#### Упражнение № D.16

---

**Разработать приложение согласно постановке задачи.**

**Разработать презентацию с демонстрацией последовательности выполняемых действий и описанием используемых объектов.**

1. Создать новую информационную базу.

2. В режиме Конфигуратор задать имя конфигурации ВАРИАНТ 10.

3. Создать подсистемы: *Учет материалов, Зарплата.*

4. Создать объекты конфигурации *Документы: Приходная накладная, Расходная накладная.* Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Наименование материала, Количество, Цена, Сумма.* Определить доступность команды создания новых документов в соответствующих подсистемах.

5. Создать отчет: *Реестр документов, Приходная накладная, Движения материалов.* Добавить ссылку на отчет в панель действий соответствующих подсистем.

## Упражнение № D.17

---

**Разработать приложение согласно постановке задачи.**

**Разработать презентацию с демонстрацией последовательности выполняемых действий и описанием используемых объектов.**

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации ВАРИАНТ 11.
3. Создать подсистемы: *Продажи, Учет материалов.*
4. Создать объекты конфигурации *Документы: Приходная накладная, Расходная накладная.* Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Наименование материала, Количество, Цена, Сумма.*

Определить доступность команды создания новых документов в соответствующих подсистемах. В режиме «1С:Предприятие» создать несколько документов.

5. Создать макет печатной формы документов. Создать команду документа с именем *Печать.* В модуль этой команды поместить обработчик, вызывающий процедуру печати документа. Команду *Печать* поместить в командную панель формы документа.

## Упражнение № D.18

---

**Разработать приложение согласно постановке задачи.**

**Разработать презентацию с демонстрацией последовательности выполняемых действий и описанием используемых объектов.**

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации ВАРИАНТ 12.
3. Создать подсистемы: *Продажи, Учет материалов.*
4. Создать объекты конфигурации *Документы: Приходная накладная, Расходная накладная.* Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Наименование материала, Количество, Цена, Сумма.*

Определить доступность команды создания новых документов в соответствующих подсистемах.

5. Автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества в строке. В режиме «1С:Предприятие» создать несколько документов.

## Упражнение № D.19

---

**Разработать приложение согласно постановке задачи.**

**Разработать презентацию с демонстрацией последовательности выполняемых действий и описанием используемых объектов.**

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации ВАРИАНТ 13.
3. Создать подсистемы: *Продажи, Учет материалов.*
4. Создать объект конфигурации *Регистр накопления: Остатки товаров.* Определить доступность созданного регистра накопления из со-

ответствующих подсистем. Создать измерения регистра. Сформировать движения регистра в процессе проведения документов. Добавить команду для открытия регистра накопления из формы документа.

5. Создать отчет: *Остатки товаров*. Добавить ссылку на отчет в панель действий соответствующих подсистем.

#### Упражнение № D.20

---

**Разработать приложение согласно постановке задачи.**

**Разработать презентацию с демонстрацией последовательности выполняемых действий и описанием используемых объектов.**

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации ВАРИАНТ 14.
3. Создать подсистемы: *Продажи, Кадры*.
4. Создать объекты конфигурации *Справочники: Сотрудники, Клиенты, Услуги*. Определить принадлежность справочников к подсистемам: *Сотрудники* к подсистемам *Кадры, Продажи; Клиенты, Услуги* к подсистеме *Продажи*. Добавить команду добавления нового элемента справочника в панель действий соответствующих подсистем. В режиме «1С:Предприятие» добавить в каждый справочник несколько элементов.
5. Создать объекты конфигурации *Документы: Оказание услуги*. Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Наименование, Количество, Цена, Сумма*. Определить доступность команды создания новых документов в соответствующих подсистемах.
6. Создать отчет: *Выручка сотрудников, Реестр документов, Оказание услуг*. Добавить ссылку на отчет в панель действий соответствующих подсистем.

#### Упражнение № D.21

---

**Разработать приложение согласно постановке задачи.**

**Разработать презентацию с демонстрацией последовательности выполняемых действий и описанием используемых объектов.**

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации ВАРИАНТ 15.
3. Создать подсистемы: *Продажи, Кадры*.
4. Создать объекты конфигурации *Справочники: Сотрудники, Клиенты, Услуги*. Определить принадлежность справочников к подсистемам: *Сотрудники* к подсистемам *Кадры, Продажи; Клиенты, Услуги* к подсистеме *Продажи*. Добавить команду добавления нового элемента справочника в панель действий соответствующих подсистем. В режиме «1С:Предприятие» добавить в каждый справочник несколько элементов.
5. Создать объекты конфигурации *Документы: Оказание услуги*. Определить доступность документа из соответствующих подсистем.

В табличной части документа добавить следующие реквизиты: *Наименование, Количество, Цена, Сумма*. Определить доступность команды создания новых документов в соответствующих подсистемах.

6. Создать макет печатной формы документа. Создать команду документа с именем *Печать*. В модуль этой команды поместить обработчик, вызывающий процедуру печати документа. Команду *Печать* поместить в командную панель формы документа.

## Упражнение № D.22

---

**Разработать приложение согласно постановке задачи.**

**Разработать презентацию с демонстрацией последовательности выполняемых действий и описанием используемых объектов.**

1. Создать новую информационную базу.
2. В режиме Конфигуратор задать имя конфигурации ВАРИАНТ 16.
3. Создать подсистемы: *Продажи, Кадры*.
4. Создать объекты конфигурации *Справочники: Сотрудники, Клиенты, Услуги*. Определить принадлежность справочников к подсистемам: *Сотрудники* к подсистемам *Кадры, Продажи*; *Клиенты, Услуги* к подсистеме *Продажи*. Добавить команду добавления нового элемента справочника в панель действий соответствующих подсистем. В режиме «1С:Предприятие» добавить в каждый справочник несколько элементов.

5. Создать объекты конфигурации *Документы: Оказание услуги*. Определить доступность документа из соответствующих подсистем. В табличной части документа добавить следующие реквизиты: *Наименование, Количество, Цена, Сумма*. Определить доступность команды создания новых документов в соответствующих подсистемах. Автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества в строке. В режиме «1С:Предприятие» создать несколько документов.

# ОГЛАВЛЕНИЕ

Предисловие .....	3
<b>ВВЕДЕНИЕ</b> .....	5
Постановка задачи сквозного примера .....	6
Некоторые замечания.....	8
<b>ЮНИТ 1</b>	
<b>С САМОГО НАЧАЛА</b> .....	9
1.1. Концепции системы «1С: Предприятие».....	9
1.1.1. Концепции системы «1С: Предприятие»: Программная часть, Информационная база и Конфигурация .....	9
1.1.2. Режимы запуска системы «1С:Предприятие» .....	10
1.2. Создание новой информационной базы.....	11
1.3. Окно «Конфигурация» .....	16
1.4. Свойства объекта Конфигурации .....	16
1.4.1. Основные свойства.....	19
1.4.2. Группа свойств «Представление».....	20
1.4.3. Группа свойств «Разработка» .....	21
1.5. Пункт меню «Конфигурация» .....	22
1.5.1. Сохранение Конфигурации.....	22
1.5.2. Обновление Конфигурации.....	22
1.5.3. Сохранение Конфигурации в файл.....	23
1.6. Администрирование.....	23
1.7. Запуск в режиме «1С: Предприятие» .....	23
1.8. Что мы узнали .....	25
<b>ЮНИТ 2</b>	
<b>РАБОТА С КОНСТАНТАМИ</b> .....	26
2.1. Создание константы.....	26
2.2. Свойства константы .....	27
2.3. Подсистемы.....	30
2.3.1. Добавление подсистемы в режиме Конфигуратор.....	31
2.3.2. Имя и синоним объекта конфигурации .....	32
2.4. Создание основной формы констант.....	34
2.5. Обращение к значениям констант из встроенного языка .....	38
2.5.1. Менеджер константы.....	38
2.5.2. Набор констант .....	40
2.6. Самостоятельная работа.....	40
2.7. Что мы узнали .....	41
<b>ЮНИТ 3</b>	
<b>РЕГИСТР СВЕДЕНИЙ «КУРСЫ»</b> .....	42
3.1. Окно редактирования регистра сведений.....	43
3.1.1. Закладка «Основные» .....	44
3.1.2. Закладка «Данные» .....	45
3.1.3. Закладка «Формы».....	46
3.2. Ввод данных в регистр сведений.....	48

3.3.	Работа с регистром сведений из встроенного языка системы .....	50
3.3.1.	Получение отдельных значений.....	50
3.3.2.	Менеджер записи.....	51
3.3.3.	Использование выборки.....	53
3.4.	Самостоятельная работа.....	53
3.5.	Использование набора записей .....	56
3.5.1.	Обработка событий .....	57
3.6.	Что мы узнали .....	60
<b>ЮНИТ 4</b>		
<b>ОБЪЕКТ «ЗАПРОС».....</b>		<b>61</b>
4.1.	Создание запроса «вручную» .....	61
4.1.1.	Написание текста запроса и его выполнение .....	62
4.1.2.	Выборка данных из результата запроса.....	64
4.2.	Что мы узнали .....	65
<b>ЮНИТ 5</b>		
<b>РАЗРАБОТКА ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ.....</b>		<b>66</b>
5.1.	Создание нового интерфейса.....	66
5.2.	Свойства интерфейса .....	68
5.3.	Редактирование интерфейса .....	69
5.3.1.	Редактирование свойств пункта меню.....	69
5.3.2.	Свойство «Действие» .....	71
5.3.3.	Создание новых пунктов меню .....	72
5.3.4.	Добавление новой панели.....	74
5.4.	Самостоятельная работа.....	75
5.5.	Тестирование интерфейса.....	77
5.6.	Переключение интерфейсов .....	78
5.6.1.	Добавление команды переключения интерфейса .....	78
5.6.2.	Перенос подменю из одного интерфейса в другой.....	81
5.6.3.	Переключение интерфейсов в режиме «1С: Предприятие».....	83
5.7.	Что мы узнали .....	83
<b>ЮНИТ 6</b>		
<b>ТАБЛИЧНЫЙ ДОКУМЕНТ.....</b>		<b>85</b>
6.1.	Команда «Вывести список» режима «1С: Предприятие» .....	85
6.2.	Объект «ТабличныйДокумент» .....	87
6.3.	Формирование отчета с использованием макета.....	88
6.3.1.	Создание общего макета.....	88
6.3.2.	Конструктор макета для создания табличного документа.....	89
6.3.3.	Свойства ячейки табличного документа .....	91
6.3.4.	Именованные области .....	93
6.4.	Улучшение вида результирующего табличного документа.....	97
6.4.1.	Объединение ячеек.....	98
6.4.2.	Изменение ширины столбцов .....	99
6.4.3.	Форматирование содержимого ячейки.....	100
6.4.4.	Установка свойств табличного документа .....	101
6.5.	Самостоятельная работа.....	102
6.6.	Формирование отчета без использования макета.....	104
6.7.	Что мы узнали .....	104

## ЮНИТ 7

<b>СПРАВОЧНИКИ И ПЕРЕЧИСЛЕНИЯ</b> .....	106
7.1. Справочник «Фирмы» .....	107
7.1.1. Закладка «Данные» .....	107
7.1.2. Закладка «Нумерация» .....	108
7.1.3. Закладка «Формы» .....	109
7.1.4. Закладка «Макеты» .....	110
7.1.5. Закладка «Интерфейсы» .....	112
7.2. Справочник «Подразделения» .....	113
7.2.1. Закладка «Иерархия» .....	114
7.2.2. Закладка «Владельцы» .....	115
7.2.3. Закладка «Нумерация» .....	116
7.2.4. Закладка «Формы» .....	117
7.2.5. Закладка «Интерфейсы» .....	120
7.3. Самостоятельная работа: справочник «Сотрудники» .....	120
7.4. Справочник «Основные Средства» .....	122
7.4.1. Реквизит «Фирма» .....	123
7.4.2. Перечисление «ГруппыОС» и реквизит «ГруппаОС» .....	124
7.4.3. Реквизит «ЕстьДетали» .....	126
7.4.4. Табличная часть «СписокДеталей» .....	127
7.4.5. Реквизиты «ДатаВвода» и «ДатаСписания» .....	128
7.4.6. Реквизит «Изображение» .....	128
7.4.7. Закладка «Нумерация» .....	129
7.4.8. Закладка «Формы» .....	129
7.5. Самостоятельная работа: справочник «Детали» .....	130
7.6. Работа со справочником из встроенного языка системы .....	131
7.6.1. Менеджер справочника .....	131
7.6.2. Организация выборки .....	131
7.6.3. Получение ссылки на элемент справочника .....	134
7.6.4. Работа с отдельными записями как с объектами .....	135
7.6.5. Закладка «Прочие» и Модуль объекта .....	136
7.7. Предопределенные записи в справочнике .....	137
7.8. Работа со справочниками в режиме «1С:Предприятие» .....	138
7.8.1. Ввод элементов в справочник «Фирмы» .....	139
7.8.2. Ввод элементов в справочник «Подразделения» .....	139
7.8.3. Ввод записей в справочник «Сотрудники» .....	141
7.9. Использование языка запросов .....	143
7.9.1. Конструктор запросов .....	143
7.9.2. Конструктор выходных форм .....	147
7.10. Что мы узнали .....	150

## ЮНИТ 8

<b>РАБОТА С ФОРМОЙ</b> .....	152
8.1. Из чего состоит форма .....	152
8.2. Реквизиты формы .....	153
8.2.1. Два объекта (реквизита) в одной форме и связь между ними .....	154
8.3. Использование объекта типа «ДеревоЗначений» .....	156
8.4. Привязка элементов формы .....	164
8.5. Что мы узнали .....	166

<b>ЮНИТ 9</b>	
<b>СПРАВОЧНИК И РЕГИСТР СВЕДЕНИЙ .....</b>	<b>168</b>
9.1. Регистр сведений «ДанныеОС» .....	168
9.2. Форма элемента справочника «ОсновныеСредства» .....	171
9.2.1. Разработка диалога .....	171
9.2.2. Обработка событий от элементов формы.....	177
9.2.3. Самостоятельная работа.....	182
9.2.4. Обработка событий табличного поля .....	182
9.2.5. Обработка событий от формы в целом .....	185
9.2.6. Изменение привязок элементов .....	192
9.2.7. Отображение картинки .....	193
9.3. Самостоятельная работа.....	196
9.4. Метод СрезПоследних(...) менеджера регистра сведений.....	199
9.5. Соединение в языке запросов .....	203
9.6. Что мы узнали .....	208
<b>ЮНИТ 10</b>	
<b>ДОКУМЕНТ И РЕГИСТР СВЕДЕНИЙ .....</b>	<b>210</b>
10.1. Основные понятия.....	210
10.2. Изменение настройки регистра сведений «ДанныеОС» .....	212
10.2.1. Изменение формы списка.....	213
10.3. Документ «Поступление».....	214
10.3.1. Закладка «Основные».....	215
10.3.2. Закладка «Данные».....	215
10.3.3. Закладка «Нумерация».....	217
10.3.4. Закладка «Движения».....	218
10.3.5. Закладка «Последовательности».....	220
10.3.6. Закладка «Журналы».....	220
10.3.7. Самостоятельная работа: закладка «Формы» .....	221
10.3.8. Закладка «Интерфейсы».....	222
10.4. Работа с документами из встроеного языка системы.....	222
10.4.1. Менеджер документа .....	223
10.4.2. Самостоятельная работа: организация выборки.....	223
10.4.3. Работа с Документом как с объектом .....	224
10.5. Форма документа .....	224
10.5.1. Обработка событий от элементов формы .....	224
10.5.2. Получение итогов в подвале табличного поля .....	227
10.5.3. Обработка событий формы .....	228
10.6. Обработка проведения документа .....	230
10.6.1. Конструктор движения.....	230
10.6.2. Доработка процедуры ОбработкаПроведения(...).....	232
10.6.3. Работа с движениями в форме документа.....	232
10.7. Работа с документом в режиме «1С:Предприятие» .....	234
10.8. Что мы узнали .....	235
<b>ЮНИТ 11</b>	
<b>ОТЧЕТЫ И ЗАПРОСЫ .....</b>	<b>237</b>
11.1. Создание отчета (запроса) «вручную» .....	237
11.1.1. Окно редактирования отчета.....	237



11.1.2. Особенности создания формы отчета.....	240
11.1.3. Модуль формы отчета.....	243
11.1.4. Изменение текста запроса и его выполнение .....	245
11.1.5. Выборка данных из результата запроса .....	246
11.2. Вызов отчета из интерфейса конфигурации .....	246
11.3. Обращение к отчету из встроенного языка системы .....	247
11.3.1. Модуль отчета.....	248
11.4. Конструктор выходных форм .....	250

## ПРИЛОЖЕНИЯ

### Приложение А

Основные элементы встроенного языка.....	254
A.1. Программный модуль .....	255
A.2. Модуль приложения .....	256
A.3. Обработка событий в модуле приложения .....	257
A.4. Системные перечисления .....	259
A.5. Системные наборы значений.....	260
A.6. Прimitivesкие типы данных .....	261
A.7. Разделы Модуля приложения .....	263
A.8. Преобразование типов и ошибки времени выполнения .....	267
A.9. Ключевое слово «Экспорт» и глобальный контекст .....	268
A.10. Общие модули .....	268
A.10.1. Назначение Общих модулей .....	268
A.10.2. Добавление Общего модуля в Конфигурацию.....	269
A.11. Самостоятельная работа.....	271
A.12. Что мы узнали .....	272

### Приложение В

Построение отчета без использования макета.....	273
V.1. Что мы узнали .....	277

### Приложение С

Мобильное приложение .....	277
C.1. Создание информационной базы .....	278
C.2. Выгрузка информационной базы на сервер .....	281
C.3. Запуск ИБ на смартфоне .....	282
C.4. Заключение.....	283

### Приложение D

Сборник ситуационных заданий (кейсов) и упражнений.....	284
Перечень ситуационных заданий (кейсов) .....	284
Упражнения .....	292



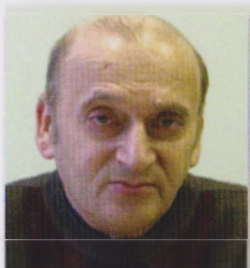
Словарь



Фрагменты листингов



www.infra-m.ru



**ДАДЯН**  
**Эдуард Григорьевич**

Кандидат технических наук, доцент. Доцент Департамента анализа данных и машинного обучения Финансового университета при Правительстве Российской Федерации.

Сфера научных интересов: нейросетевые технологии, методы и средства анализа финансового рынка и образовательной деятельности.

Автор более 300 учебных, методических и научных работ, в том числе монографий, учебников и учебных пособий.

ISBN: 978-5-16-016648-3



9 785160 166483



Уважаемый читатель!  
Вы держите в руках книгу, дополнительные материалы которой доступны Вам **БЕСПЛАТНО** в интернете на [www.znaniium.com](http://www.znaniium.com)  
Специального программного обеспечения не требуется