

Сергей Кашаев

ПРОГРАММИРОВАНИЕ в 1С:Предприятие 8.3

1C
специалист

Сергей Михайлович Кашаев

Программирование в 1С:Предприятие 8.3

Серия «1Специалист»

Заведующий редакцией	<i>П. Щеголев</i>
Руководитель проекта	<i>А. Юрченко</i>
Ведущий редактор	<i>Ю. Сергинко</i>
Литературный редактор	<i>А. Жданов</i>
Художественный редактор	<i>С. Заматевская</i>
Корректоры	<i>К. Герцен, В. Листова</i>
Верстка	<i>Л. Родионова</i>

ББК 32.973.233-018.2

УДК 004.9

Кашаев С. М.

К31 Программирование в 1С:Предприятие 8.3. — СПб.: Питер, 2014. — 304 с.: ил. — (Серия «1Специалист»).

ISBN 978-5-496-01234-8

Как самостоятельно освоить приемы разработки конфигурации и модификации прикладных приложений в среде 1С:Предприятие 8.3? Каким арсеналом средств вооружает программиста новейшая версия платформы? Ответы на эти вопросы предлагает данная книга, на практике знакомящая читателя со встроенным языком программирования и системным отладчиком, редакторами и конструкторами, системой компоновки данных, инструментами анализа и построения диаграмм. Подробно изложена вся требуемая последовательность действий, приведены необходимые справочные сведения по используемым средствам и языку. Теоретические сведения подкреплены многочисленными примерами, которые помогут разработчику максимально использовать все возможности, предоставляемые версией 8.3.

6+ (Для детей старше 6 лет. В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ISBN 978-5-496-01234-8

© ООО Издательство «Питер», 2014

Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги.

ООО «Питер Пресс», 192102, Санкт-Петербург, ул. Андреевская (д. Волкова), д. 3, литер А, пом. 7Н.

Налоговая льгота — общероссийский классификатор продукции ОК 005-93, том 2: 95 3005 — литература учебная.

Подписано в печать 06.06.14. Формат 70x100/16. Усл. п. л. 24,510. Тираж 1700. Заказ

Отпечатано в полном соответствии с качеством предоставленных издательством материалов
в ГППО «Псковская областная типография». 180004, Псков, ул. Ротная, 34.

Содержание

Введение	5
Взгляд на систему 1С:Предприятие	6
Читателям второго издания	8
Краткое содержание книги	9
Благодарности	10
От издательства	10
Глава 1. Знакомство с платформой 1С:Предприятие 8.3	11
Взгляд на систему 1С:Предприятие	12
Средства разработки в системе 1С:Предприятие	14
Установка системы 1С:Предприятие 8.3	15
Постановка задачи	19
Начало работы	20
Добавление новой информационной базы	22
Подсистемы	26
Панель разделов приложения	28
Справочники	29
Перечисления	50
Документы	53
Регистры накопления	76
Перемещение товаров на выставку и обратно	84
Оборотный регистр накопления	89
Организация интерфейса приложения	94
Итоги	97
Глава 2. Язык запросов в системе 1С:Предприятие..	98
Организация выборки из таблицы	100
Сортировка в запросах	106
Отбор фрагмента данных	109
Отбор уникальных значений	111
Логические операторы	113
Агрегатные функции в запросах	116
Параметрические запросы	120
Ключевое слово МЕЖДУ	127

Ключевое слово ИМЕЮЩИЕ	128
Объединение запросов	129
Соединение таблиц при организации запроса	131
Функция ПРЕДСТАВЛЕНИЕ	137
Итоги	137
Глава 3. Разработка отчетов с использованием системы компоновки данных	138
Отчеты по остаткам товаров	139
Отчет по документам Поступление Товаров	152
Отчет по работе филиалов	155
Отчет по работе менеджеров	159
Использование конструктора настроек при создании отчетов	163
Отчеты с графиками	167
Итоги	173
Глава 4. Автоматизация при организации учета	174
Константы	174
Документ Заказ Клиента	176
Ввод на основании	177
Журналы документов	182
Коррекция документов Поступление Товаров и Продажа Товаров	186
Регистр Расчеты С Фирмами	188
Документ Выписка Банка	194
Автоматизация расчета суммы по табличной части документа	199
Отчет по взаиморасчетам с фирмами	204
Итоги	211
Глава 5. Примеры прикладных решений	212
Учет оплаты учебных курсов	212
Программирование форм	228
Заполнение формы документа данными из регистра сведений	251
Автоматизация для учета денежных средств	264
Передача данных в Microsoft Excel	281
Анализ работы менеджеров	287
Итоги	303
Литература	304

Введение

Информационная система 1С:Предприятие широко известна в нашей стране и за ее пределами. С помощью этого программного продукта практически всегда удается существенно повысить степень автоматизации работы в организациях различного профиля. За последнее десятилетие выпущено несколько версий данной программы, и появление каждой новой версии приводит к качественному повышению уровня использования информационных технологий в большинстве организаций.

В предлагаемой книге речь идет о последней на сегодняшний день версии данной программы – системе 1С:Предприятие 8.3. Заметим, что она уже весьма популярна как среди тех, кто имеет опыт использования более ранних платформ 1С:Предприятие, так и среди новых пользователей.

Для автоматизации конкретных участков работы (бухгалтерия, склад, управление персоналом и т. д.) специалистами фирмы 1С создано большое число типовых решений, позволяющих пользоваться ими без каких-либо программных доработок. Однако часто встречаются ситуации, когда требуется внести изменения или добавления в готовую конфигурацию. Дело в том, что практически в каждой организации имеются свои особенности. В других ситуациях бывает удобнее разработать собственную конфигурацию, направленную на решение той или иной прикладной задачи. В этом случае от разработчика требуется знание технологии создания прикладных решений на платформе 1С:Предприятие. Если попытаться подобрать наиболее емкое и понятное слово, характеризующее данную деятельность, то лучше всего подходит слово *программирование*. Оно как раз и вынесено в название данной книги. Однако здесь требуется немного уточнить сказанное. В традиционном представлении под программированием понимается написание программного кода — в текстовом редакторе, строка за строкой. В результате такой работы и получается исполняемая программа. Частично это относится и к системе 1С:Предприятие 8.3. Так, в программный продукт 1С:Предприятие заложен встроенный программный язык, и построение большинства прикладных решений на платформе 1С:Предприятие связано с написанием программного кода на данном языке. Однако при создании решений также активно используются визуальные средства разработки. Именно в версии 1С:Предприятие 8.3 они выходят на первый план, а доля программирования в традиционном виде существенно снижена. Так, задачу создания сложных отчетов можно полностью решить вообще без написания программного кода. Выполнение технических действий с использованием конструкторов, построителей и мастеров позволяет быстро получить необходимый результат. Фактически при этом происходит так называемое визуальное программирование,

когда формирование необходимого прикладного решения производится средствами автоматизации, предоставляемыми средой разработки.

Таким образом, наряду с традиционным программированием визуальные средства разработки (визуальное программирование) занимают важное (можно даже сказать, ключевое) место в системе 1С:Предприятие 8.3.

Понятно, что для создания новых разработок, а также для внесения даже небольших изменений в готовое решение необходимы существенные знания и практические навыки. К ним относятся: представление об объектах, составляющих систему 1С:Предприятие 8.3, знание встроенного языка программирования и языка построения запросов, представление об имеющихся визуальных средствах, опыт разработчика и многое другое. А уже после этого перед пользователями открываются широкие возможности как для внесения изменений в готовые решения, так и для создания собственных прикладных решений. Именно для такой категории читателей, которые на практических примерах хотят быстрее овладеть секретами разработки решений на платформе 1С:Предприятие 8.3, и написана эта книга.

Одним из первых возникает вопрос о том, что необходимо предварительно знать для успешного усвоения информации. Практические занятия с различными категориями учащихся показали, что освоение приведенных в книге примеров вполне по силам как тем, кто не имеет навыков в области программирования, так и тем, кто до этого не сталкивался с системой 1С:Предприятие. Большинство примеров использовались в реальном учебном процессе. На занятиях присутствовали учащиеся различных возрастных категорий и с разным уровнем начальной подготовки.

Вообще практика показала, что приобрести навыки самостоятельных разработок в системе 1С:Предприятие можно за относительно небольшой срок.

Взгляд на систему 1С:Предприятие

Программный продукт 1С:Предприятие позволяет существенно автоматизировать деятельность как определенного подразделения, так и организации в целом.

Основной отличительной чертой рассматриваемой системы является ее *способность подстройки* под конкретную область деятельности. Это достигается за счет возможности *конфигурирования*, которая заложена в данный программный продукт. Вообще функционирование системы разделяется на два процесса — конфигурирование (работа в режиме *конфигуратора*) и непосредственная работа с тем, что создано на этапе конфигурирования.

В режиме конфигуратора создаются необходимые для решения прикладной задачи объекты (точнее — объекты конфигурации). При этом определяются формы ввода информации, алгоритмы обработки данных, разрабатываются необходимые

пользователю отчеты. В целом результатом конфигурирования является конкретная **конфигурация**, которая отражает модель определенного участка деятельности (подразделения или всей организации).

Сама по себе деятельность разработчика в режиме конфигуратора может быть разделена на две составляющих. Так же, как и в любой другой современной инструментальной среде, в системе 1С:Предприятие 8.3 большое место занимают визуальные средства разработки. В этом случае практическая сторона фактически сводится к управлению объектами с помощью мыши, а также к размещению необходимой информации в полях ввода. Вторая же составляющая работы в конфигураторе касается непосредственного программирования на встроенным языке.

При создании конфигурации, направленной на решение конкретной прикладной задачи, разработчик использует такие заложенные в систему базовые объекты конфигурации, как справочник, документ, регистр накопления, форма, табличный документ и многие другие. На базе указанных объектов разработчик создает «свои» объекты конфигурации для определенной прикладной области. Например, это могут быть справочники «Номенклатура» или «Сотрудники», первый из которых предназначен для хранения названий и реквизитов товаров, второй — для хранения информации о лицах, работающих в организации. Также можно, используя *документ* в качестве базового объекта конфигурации, создать структуру прикладного документа «Приходная накладная».

Здесь следует подчеркнуть, что в режиме конфигуратора создается не конкретный документ (соответственно — не конкретный справочник или отчет), а лишь определяется его вид, структура или, можно сказать, шаблон. Например, в режиме конфигуратора можно определить, что в создаваемом документе должны присутствовать следующие элементы:

- название склада;
- название фирмы;
- список товаров с указанием количества.

После этого в работу включается пользователь, который в режиме 1С:Предприятие вносит данные в информационные структуры, созданные системой на основании разработанных шаблонов. Таким образом, сотрудникам, не знакомым с конфигуратором и технологией создания объектов конфигурации, доступна только такая, пользовательская составляющая деятельности.

Важно отметить, что система 1С:Предприятие позволяет создавать объекты, которые являются очень близкой моделью реальных документов и отчетов, используемых в конкретной сфере деятельности. В этом случае пользователи, впервые столкнувшись с этой информационной системой, отмечают, насколько удобнее работать после перехода от бумажной документации к электронной.

Фактически для последующей работы пользователя в режиме 1С:Предприятие разработчиком в режиме конфигуратора создается необходимая схема (алгоритм действий). В результате пользователь работает с информационной базой, созданной системой на основании созданной конфигурации.

В качестве одного из технологических средств разработки, как мы уже отметили, используется встроенный в рассматриваемую систему язык программирования. Это наиболее существенный ресурс разработчика, но он не единственный, поэтому следует упомянуть еще несколько наиболее популярных ресурсов. Так, в системе 1С:Предприятие присутствуют конструкторы – инструменты, которые позволяют разработчику избежать выполнения разнообразных рутинных действий. В частности, конструкторы предоставляют возможность быстро создавать необходимые формы для ввода данных, формировать печатные формы, генерировать программный код и, что важно, все это делается в результате несложного диалога.

В программный продукт 1С:Предприятие заложен также механизм запросов, который в основном используется для построения разнообразных отчетов, являющихся непременным атрибутом автоматизации в любой предметной области.

Доступный в системе текстовый редактор служит для создания программных модулей на встроенном языке системы 1С:Предприятие. Для удобства разработчика в этом редакторе имеется возможность контекстного выделения цветом синтаксических конструкций, а при работе программист может пользоваться подсказками и шаблонами.

Для создания макетов выходных форм в системе 1С:Предприятие предусмотрен редактор табличных документов. Формат табличного документа является универсальным для документов, справочников, отчетов и других объектов.

Если провести сравнение с версией 1С:Предприятие 8.0, то следует отметить, что в информационные системы 1С:Предприятие 8.1, 1С:Предприятие 8.2 и 1С:Предприятие 8.3 добавлено несколько новых ресурсов. Наиболее существенное добавление – *система компоновки данных*. Это очень эффективный ресурс для построения отчетов. Механизм компоновки данных позволяет исключительно визуальными средствами дать целостное описание отчета. Системой компоновки данных мы будем пользоваться на протяжении всей книги.

Читателям второго издания

Первое издание книги, вышедшее под названием «Программирование. 1С:Предприятие 8.2», получило много положительных отзывов читателей. Фактически это и стало поводом для подготовки второго издания, которое вы сейчас держите в руках. Хочу выразить благодарность руководителю проекта, Андрею Юрченко, за предложение подготовить второе издание данной книги.

Материал первого издания был переработан для новой платформы 1С:Предприятие 8.3. Также в новую книгу вошли несколько новых разделов и примеров разработки прикладных решений.

Краткое содержание книги

В целом книга состоит из пяти глав, наполненных разнообразными практическими примерами. Такая организация издания связана с удобством восприятия информации читателями. Необходимые справочные сведения по используемым средствам, встроенному языку программирования и языку запросов приводятся по ходу изложения примеров, что также весьма удобно для читателя. Большой эффект при работе с материалом книги будет достигаться при параллельном выполнении приведенных разработок на компьютере. Именно для этого все технические шаги в книге описаны очень подробно.

В первой главе читатели познакомятся с такими основными объектами конфигурации, как *справочник*, *документ* и *регистр накопления*. На практических примерах будет рассмотрена технология разработки электронных форм, являющихся обязательным атрибутом интерфейса пользователя. Программирование начнется с небольших процедур, которые будут понятны читателям, не встречавшимся до этого с системой 1С:Предприятие 8 либо работавшими с ней только в качестве пользователей. Вы увидите, как с помощью несложного программного кода можно легко предоставить пользователям необходимый сервис для последующей работы.

В целом стратегическая цель первой главы заключается в том, чтобы заложить у читателей базовые навыки по работе в режиме конфигуратора, который является одним из основных режимов для системы 1С:Предприятие. Результатом действий разработчика в этом режиме является *конфигурация*, применяемая пользователем в режиме работы 1С:Предприятие (это другой, не менее важный для рассматриваемой системы режим).

Вторая глава целиком отведена *языку запросов*, который позволяет обеспечить удобный доступ к имеющимся в системе данным. С помощью языка запросов можно реализовать отбор и группировку данных.

В третьей главе подробно рассматривается *система компоновки данных*. Здесь показано, каким образом без знания программного кода можно эффективно создавать необходимые отчеты. Система компоновки данных – это новое средство, предназначенное для создания отчетов на основе декларативного описания. Механизм компоновки данных позволяет дать целостное описание отчета, используя только визуальные средства.

Четвертая глава полностью посвящена приемам *автоматизации учета информации*. Мы путем совершенствования конфигурации, приведенной в первой главе, рассмотрим новые технические ресурсы, которые можно использовать при построении прикладных решений.

В пятой главе разбираются разнообразные *примеры создания конфигураций*. Эти разработки касаются учета денежных средств и товаров, а также анализа работы менеджеров в торговой организации. Большое внимание при этом уделяется созданию сервисов для пользователя и построению разнообразных отчетов.

Примеры рассматриваемых в книге информационных баз можно загрузить с сайта издательства «Питер» (www.piter.com).

Благодарности

В первую очередь мне бы хотелось выразить благодарность всем читателям, которые приобрели мою книгу. По тематике, связанной с офисным программированием, она у меня не первая. Что касается системы 1С:Предприятие, то здесь следует упомянуть издания [1–4], которые были отмечены положительными отзывами читателей. Фактически это стало для меня поводом написать еще одну книгу, связанную с системой 1С:Предприятие. И я очень благодарен издательству «Питер», которое приняло решение о выпуске этого издания.

В заключение еще раз замечу, что наибольшего эффекта вы добьетесь, если параллельно с чтением будете выполнять практические примеры непосредственно в системе 1С:Предприятие 8.3.

Я думаю, что книга не лишена неточностей; все их я отношу исключительно на свой счет и заранее приношу читателям свои извинения.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу электронной почты comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства <http://www.piter.com> вы найдете подробную информацию о наших книгах.

Глава 1

Знакомство с платформой 1С:Предприятие 8.3

Учитывая большую популярность программного продукта 1С:Предприятие 8, практически повсеместно растет потребность в специалистах, знакомых с этой информационной системой. Иногда знание платформы 1С:Предприятие 8 является определяющим фактором в организациях при подборе кандидатуры на то или иное вакантное место. В этом случае от потенциального сотрудника требуются в основном навыки уверенного пользователя одной из наиболее популярных на сегодняшний день конфигураций, разработанных для платформы 1С:Предприятие 8. Такой специалист занимается профессиональной деятельностью в своей предметной области (например, бухгалтерия или сфера складского учета), но с использованием современных технологий автоматизации. В свою очередь, большинство сложных задач, связанных с внесением изменений в имеющуюся конфигурацию (например, добавление нового объекта конфигурации — документа или отчета), решает, как правило, программист.

Распространена и другая ситуация. Так, в небольших организациях пользователям и программистам (которые не сталкивались до этого с программными продуктами фирмы 1С) часто приходится самостоятельно решать многие практические вопросы, связанные с внесением изменений в имеющуюся конфигурацию. Именно для такой категории читателей мы и попробуем на примерах разобрать основные приемы, касающиеся подготовки прикладных решений на платформе 1С:Предприятие 8.3.

Разумеется, сразу возникает вопрос — а что нужно предварительно знать для успешного восприятия изложенного в книге материала? Конечно, желательны навыки в области программирования, также приветствуется любой опыт работы с информационной системой 1С:Предприятие. Все это придаст читателю больше уверенности при чтении книги, повысит скорость рассмотрения примеров и позволит ему быстрее перейти к решению собственных задач. Однако можно обойтись и без этого.

Главное, что необходимо — это наличие на своем компьютере установленной программы 1С:Предприятие 8.3. В этом случае оптимальная технология работы с книгой на начальном этапе заключается в параллельном выполнении рассматриваемых

примеров на своем компьютере. Понятно, что объем справочной информации по разработке прикладных решений в среде 1С:Предприятие 8.3 просто огромен, и на протяжении всей книги нам потребуется знакомиться с рядом справочных сведений. По мере чтения книги мы будем приводить подобную информацию постепенно, что весьма удобно для читателей. Рассмотрение примеров не требует использования дополнительных источников информации: все справочные данные есть в книге.

В работе нам не потребуется наличие стандартных конфигураций, разработанных для платформы 1С:Предприятие 8.3, — мы будем создавать собственные конфигурации и на их примере рассматривать различные технические приемы, необходимые для выполнения прикладных разработок. Конечно, эти конфигурации не будут такими сложными, как разработки специалистов фирмы 1С, однако итогом рассмотрения изложенных примеров для читателей станут навыки, которые они смогут использовать в практической работе. Это касается и создания собственных разработок, и внесения изменений в стандартные конфигурации. Так или иначе, вы сможете автоматизировать деятельность своих организаций, используя платформу 1С:Предприятие 8.3.

Перед рассмотрением примеров весьма полезно предварительно поговорить об архитектуре системы 1С:Предприятие.

Взгляд на систему 1С:Предприятие

В системе 1С:Предприятие можно выделить две ключевые составляющие:

- технологическая платформа;
- прикладные решения, которые создаются с помощью технологической платформы и касаются автоматизации различных участков деятельности.

Такая организация обеспечивает хорошую адаптацию прикладных решений для конкретных заказчиков. В этом случае разработчик, даже не участвовавший в создании прикладного решения, располагает всеми необходимыми ресурсами для внесения в него изменений. Заметим, что в настоящее время уже имеется большое количество как стандартных, так и узкоспециализированных конфигураций. И большей части разработчиков приходится модифицировать уже имеющиеся решения. Открытость для подобной модификации делает систему 1С:Предприятие очень удобной для программистов, которые могут дорабатывать и развивать существующие типовые разработки.

Можно сказать, что каждая организация, располагая небольшим составом разработчиков (даже одним программистом), может настроить большинство стандартных прикладных решений «под себя». При этом для модификации не требуется использовать какие-либо отдельные программные продукты — все средства разработки входят в состав технологической платформы.

В технологической платформе выделяются две составляющие:

- среда исполнения;
- среда разработки.

Среда исполнения поддерживает *режим работы 1С:Предприятие*, а среда разработки открывается пользователю при запуске системы в режиме *конфигуратора*.

В среде разработки используется технология *метаданных*, которые представляют собой структуру объектов, описывающую конкретное прикладное решение. Среда исполнения обрабатывает (воспроизводит) метаданные. Здесь можно провести некую аналогию с любой системой, исполняющей программный код.

При работе с метаданными доступны визуальные средства разработки. В этом случае программисту не требуется самостоятельно писать программный код для добавления нового объекта. Все действия выполняются щелчками мыши и установкой необходимых параметров с помощью списков, переключателей и других элементов управления. Результат же этих действий автоматически трансформируется системой в программный код.

Работа программиста в среде разработки приводит к построению *конкретной конфигурации*. И такая конфигурация обрабатывается (воспроизводится) в *среде исполнения* (в режиме 1С:Предприятие). Что касается содержания конфигурации, то следует заметить, что платформа 1С:Предприятие содержит фиксированный набор прототипов (шаблонов) объектов конфигурации — это так называемые базовые объекты конфигурации. Например, имеются шаблоны справочника, документа, регистра сведений, отчета и т. д. Когда в среде разработки создается новый объект конфигурации, то он *наследует используемый при его построении прототип*. Важно отметить, что любой объект конфигурации для прикладного решения может быть создан только на основании одного из базовых объектов конфигурации.

Таким образом, несмотря на то что каждая прикладная конфигурация обладает собственной индивидуальностью, она целиком строится из базовых объектов конфигурации. Такая организация системы придает всем прикладным решениям определенную степень стандартизации (любая разработка состоит из стандартного набора «деталей»).

Встроенный язык программирования 1С:Предприятие, а также язык запросов служат для описания специфических алгоритмов прикладного решения. Встроенный язык 1С:Предприятие имеет много общих черт с другими языками программирования, такими как Visual Basic, JavaScript и др. Для построения запросов в систему 1С:Предприятие включен еще один язык — язык запросов, который похож на известный (по работе с базами данных) язык построения запросов SQL. Важной особенностью языка запросов в системе 1С:Предприятие является то, что он обеспечивает доступ к данным *только для чтения*. Для записи же информации используются конструкции на встроенном языке 1С:Предприятие.

Средства разработки в системе 1С:Предприятие

К средствам работы с метаданными относятся: *окно конфигурации*, *окно редактирования объекта конфигурации* и *панель свойств*. С помощью этих инструментов выполняется добавление объектов конфигурации, изменение их свойств, установка связей с другими объектами конфигурации.

В состав средств разработки входит большое количество конструкторов и редакторов. *Конструкторы* позволяют автоматизировать процесс создания прикладного решения. При этом они существенно облегчают данный процесс для разработчика. Например, текст запроса может быть написан самим разработчиком, и для этого необходимо использовать синтаксические конструкции языка запросов (которые, разумеется, надо хорошо знать). В качестве альтернативы текст запроса можно также создать с помощью *конструктора запросов*. При этом имеет место визуальное конструирование запроса, и все основные действия выполняются разработчиком с помощью мыши. В результате конструктор автоматически создает синтаксически верный текст запроса.

ПРИМЕЧАНИЕ

Построение синтаксически правильных текстов запросов с помощью конструктора можно использовать для изучения запросов.

В системе 1С:Предприятие 8.3 существует много различных конструкторов, и при рассмотрении примеров книги мы воспользуемся следующими:

- конструктор запросов;
- конструктор движений регистров;
- конструктор печати;
- конструктор форм объектов конфигурации.

Например, *конструктор форм объектов конфигурации* позволяет создать типовую форму объекта, указать для нее поля, назначить источники данных и т. д. Использование конструктора в данном случае связано с тем, что рассматриваемый процесс легко формализуется.

Для разработчика системой предоставляется ряд редакторов, таких как редактор форм, редактор текстов, редактор табличных документов, редактор интерфейсов и еще ряд других.

В системе имеется *помощник по синтаксису*, позволяющий получить справку по использованию конструкций встроенного языка, свойствам и методам объектов.

Если провести сравнение с версией 1С:Предприятие 8.0, то следует отметить, что в информационные системы, начиная с 1С:Предприятие 8.1, добавлено несколько новых ресурсов. На протяжении всей книги мы будем активно использовать наиболее существенное добавление — *систему компоновки данных*. Это весьма эффективный ресурс для построения отчетов. Механизм компоновки данных позволяет дать целостное описание отчета, используя только визуальные средства.

Установка системы 1С:Предприятие 8.3

Для работы с книгой вам потребуется установить на свой компьютер платформу 1С:Предприятие 8.3. Возможно, у вас эта система уже установлена; тогда данный раздел вы можете пропустить.

Загрузить дистрибутив платформы 1С:Предприятие 8.3 можно бесплатно с сайта компании 1С. Далее в этом разделе мы поясним, как это сделать.

Существует несколько сайтов компании 1С. В данном случае нас интересуют те, на которых имеется возможность бесплатной установки последней версии платформы 1С:Предприятие. Таких сайтов два:

- русскоязычный — <http://v8.1c.ru/>;
- англоязычный — <http://1c-dn.com/>.

Поговорим сначала о сайте <http://v8.1c.ru/>. На стартовой странице сайта присутствует горизонтальное меню, в котором имеется пункт **Методическая поддержка**. С помощью этого пункта следует обратиться к разделу **Учебные версии**, где после предварительной регистрации можно бесплатно загрузить последнюю версию программы 1С:Предприятие. Однако на момент написания данной книги на сайте <http://v8.1c.ru/> в качестве последней позиционировалась версия 1С:Предприятие 8.2. Поэтому для получения учебной версии 1С:Предприятие 8.3 вам, возможно, придется обратиться к сайту <http://1c-dn.com/>.

На рис. 1.1 показана англоязычная стартовая страница сайта <http://1c-dn.com/>, где в центральной части присутствует название раздела **Learn**. Щелкнув мышью на этом названии, мы перейдем к следующей странице (рис. 1.2). Здесь нас интересует раздел **Building your first application**. В результате перехода по данной гиперссылке перед нами откроется новая страница сайта (рис. 1.3), где нам нужно воспользоваться ссылкой **download the free 1C:Enterprise Platform (Training Version)**.

Далее перед нами откроется окно, показанное на рис. 1.4, где необходимо щелкнуть на ссылке **Download**. Это приведет к переходу на страницу с набором ссылок на различные версии платформы 1С:Предприятие (рис. 1.5). На момент написания книги

для загрузки последней версии платформы 1С:Предприятие нужно было щелкнуть на ссылке 1C:Enterprise 8 platform Training version v. 8.3.4.365.

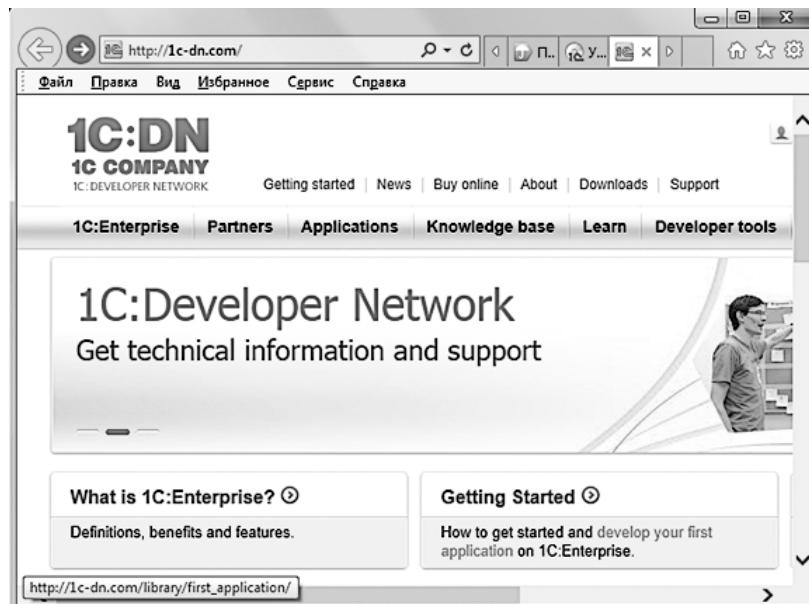


Рис. 1.1. Стартовая страница англоязычного сайта компании 1С

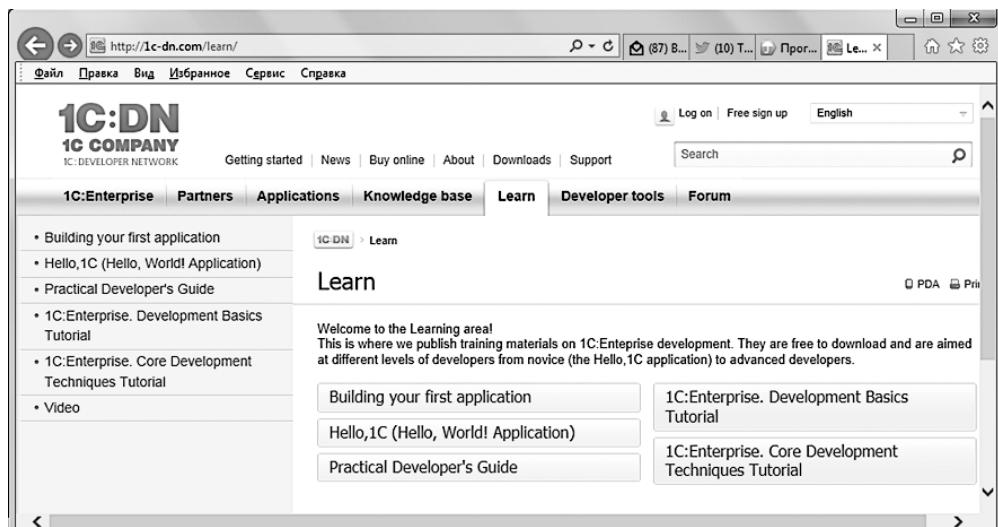


Рис. 1.2. Страница Learn англоязычного сайта компании 1С

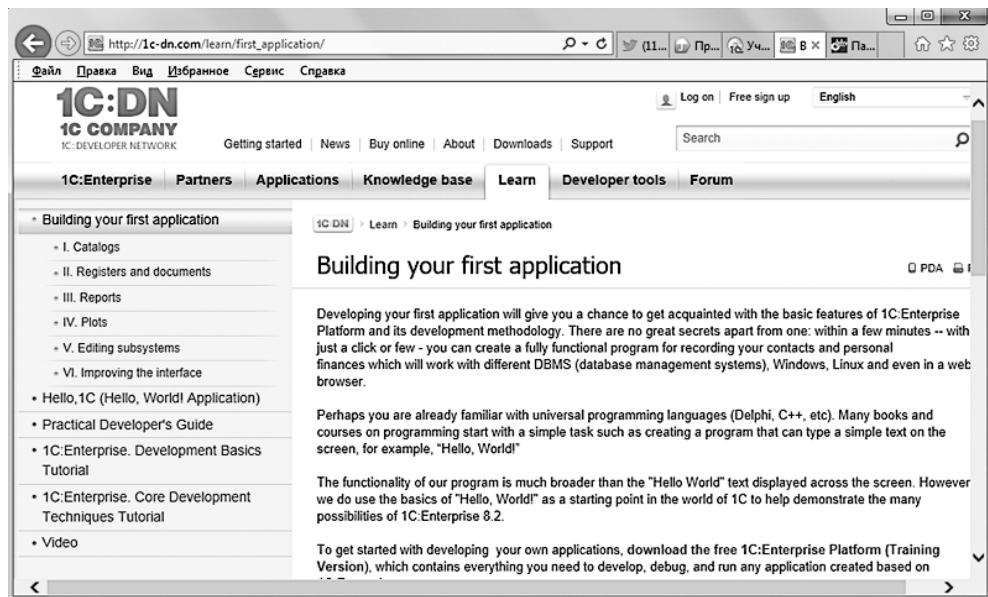


Рис. 1.3. Страница англоязычного сайта компании 1С

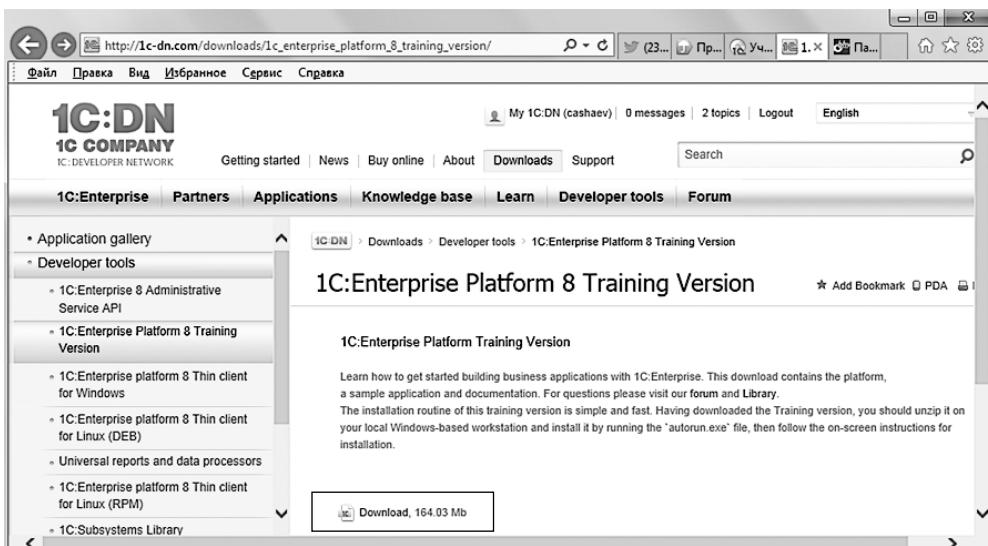


Рис. 1.4. Страница сайта компании 1С со ссылкой для загрузки программы 1С:Предприятие 8.3

ПРИМЕЧАНИЕ

Перед загрузкой программы потребуется предварительно зарегистрироваться.

The screenshot shows a web browser window for the 1C:DN website (https://1c-dn.com/user/uploads/1c_enterprise_platform_training_version/). The page displays a sidebar with navigation links like 'Файл', 'Правка', 'Вид', 'Избранное', 'Сервис', 'Справка', '1C:COMPANY', 'Getting started', 'News', 'Buy online', 'About', 'Downloads', 'Support', 'Search', and 'Logout'. The main content area is titled '1C:Enterprise 8 platform Training version' and includes a 'Product description' section and a 'Versions' table.

	Versions	Release date
1C:Enterprise 8 platform Training version v. 8.3.4.365	Dec 04, 2013	
Version 8.3.4 includes known bug fixes and some functionality modifications.		
1C:Enterprise 8 platform Training version v. 8.2.19.80	Dec 03, 2013	
Version 8.2.19 includes known bug fixes and some functionality modifications.		
1C:Enterprise 8 platform Training version v. 8.2.19.76	Nov 01, 2013	

Рис. 1.5. Страница сайта компании 1С с перечнем доступных версий платформы 1С:Предприятие

В результате загрузки мы получим архив, который сначала необходимо распаковать. В распакованном наборе файлов архива следует воспользоваться исполняемым файлом установочного пакета — autorun.exe.

Стартовое окно программы установки показано на рис. 1.6. Здесь следует щелкнуть на ссылке Custom Setup, после чего в следующем окне (рис. 1.7) выбрать вариант 1C:Enterprise 8.3. Training version. В результате начнется процесс установки системы 1С:Предприятие.

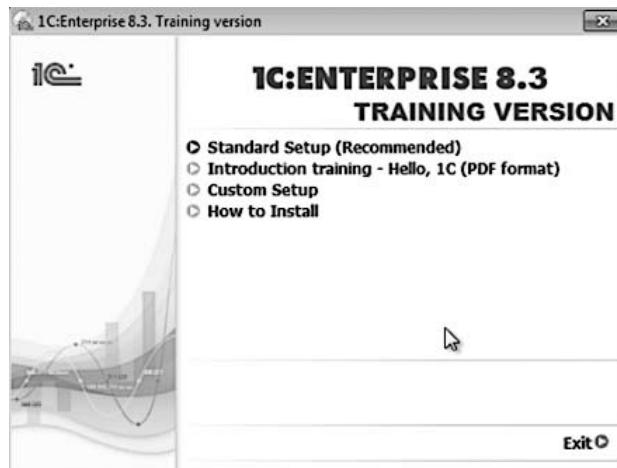


Рис. 1.6. Стартовое окно программы установки системы 1С:Предприятие 8.3

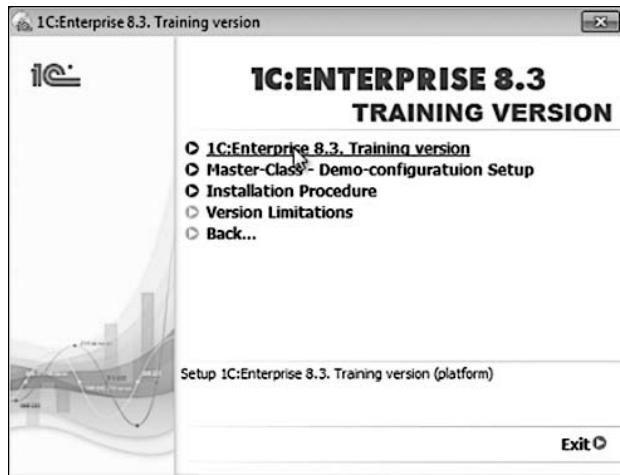


Рис. 1.7. Окно программы установки системы 1С:Предприятие 8.3

В результате установки в меню Пуск появится новое подменю 1С:Предприятие 8 (учебная версия). В этом подменю присутствует пункт 1С:Предприятие, с помощью которого и запускается приложение 1С:Предприятие 8.3 (рис. 1.8).

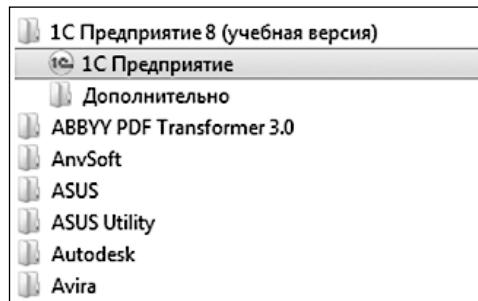


Рис. 1.8. Подменю 1С:Предприятие 8 (учебная версия) в меню Пуск

Постановка задачи

Итак, к настоящему моменту платформа 1С:Предприятие 8.3 установлена на вашем компьютере и можно приступать к практическим шагам, которые нам позволят познакомиться с реальной работой программиста по созданию прикладных решений для платформы 1С:Предприятие 8.3. Точнее, в терминологии системы 1С:Предприятие наши действия заключаются в разработке конфигурации для платформы 1С:Предприятие 8.3.

Для решения типичной задачи учета товаров мы создадим набор объектов конфигурации (это будут справочники, документы и регистры накопления), а затем поработаем в качестве пользователей с информационной базой, созданной системой на основании разработанной конфигурации. В последующих главах разработанная здесь конфигурация будет использоваться для изучения языка запросов (см. главу 2) и создания отчетов с использованием *системы компоновки данных* (см. главу 3). В этих главах разработанная конфигурация станет базой для создания других объектов конфигурации. В дальнейшем (в главе 4) мы усовершенствуем конфигурацию, разработанную в главе 1. В частности, мы создадим механизм учета денежных средств от контрагентов.

Все технические моменты, связанные с работой в конфигураторе, подробно комментируются, а также иллюстрируются копиями экрана компьютера. Таким образом, читатель легко сможет повторить описанные действия на своей вычислительной системе. При желании можно обратиться к сайту издательства «Питер» (www.piter.com), где разработанную в данной главе конфигурацию (как, впрочем, и все другие конфигурации, описываемые в книге) можно непосредственно загрузить на свой компьютер.

Для определенности дальнейших действий сразу сформулируем прикладную задачу, которую мы собираемся решить для условной фирмы. Будем считать, что наша организация ведет торговую деятельность, которая заключается в закупке товаров у поставщиков и производителей. Дальнейшие действия сводятся к продаже этих товаров через торговую сеть, состоящую из нескольких филиалов. При этом будем считать, что иногда часть товаров передается на выставку для рекламы нашего ассортимента. Соответственно, существует и обратный процесс — передача товаров с выставки в филиалы. Разумеется, разрабатываемая компьютерная система учета должна эти перемещения фиксировать. При продаже товаров необходимо также учитывать результаты работы менеджеров для их последующего денежного вознаграждения. Это описание представляет собой упрощенную модель деятельности реальной организации.

Мы, как разработчики прикладного решения на платформе 1С:Предприятие 8.3, попробуем создать конфигурацию, которая поможет автоматизировать учет поступлений, продаж и перемещений товаров. Логическая завершенность этой разработки будет реализована во второй и третьей главах книги, где рассматривается технология отбора информации из базы данных и иллюстрируется процесс создания необходимых отчетов.

Начало работы

Вся разработка выполняется в режиме *конфигуратора*, который, как уже говорилось, является одним из режимов работы информационной системы 1С:Предприятие 8.3.

По сути, в конфигураторе были созданы все известные прикладные разработки фирмы 1С. Вы их, наверняка, хорошо знаете или хотя бы неоднократно о них слышали. Конфигурации Бухгалтерский учет, Управление персоналом, Управление торговлей и многие другие помогают эффективно организовать работу в организациях. После внимательного изучения последующего описания вы будете представлять технологию разработки конфигураций для платформы 1С:Предприятие 8.3.

Приступим к практическим действиям, и после выбора в подменю 1С Предприятие 8 (учебная версия) меню Пуск пункта 1С Предприятие перед нами откроется стартовое окно системы 1С:Предприятие 8.3, которое показано на рис. 1.9. Это окно запуска, которое позволяет нам как начать новую разработку, так и продолжить работу по корректировке, доработке или совершенствованию имеющейся конфигурации.

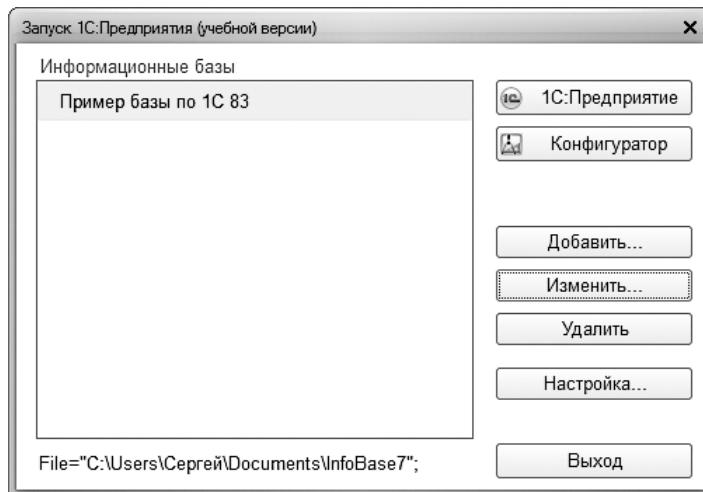


Рис. 1.9. Стартовое окно системы 1С:Предприятие 8.3

ПРИМЕЧАНИЕ

В соответствии с принятой терминологией режим 1С:Предприятие в рассматриваемой информационной системе считается основным, что связано с работой в нем пользователей. Они заполняют справочники, создают документы того или иного образца, получают необходимые отчеты и т. д. Режим конфигуратора, как правило, используется разработчиками и системными программистами. В данной книге основное внимание уделяется режиму конфигуратора, и большинство описываемых в книге действий будет производиться именно в нем.

Внимание сразу привлекает центральная часть этого окна, где перечислены уже имеющиеся информационные базы, каждая из которых представляет собой кон-

крайнюю разработку, направленную на решение определенной прикладной задачи (как правило, подобные задачи связаны с автоматизацией требуемого участка работы). Сейчас мы начинаем решение новой задачи, поэтому от нас требуется создание новой информационной базы, а уже имеющиеся разработки (одна из них представлена в окне на рис. 1.9) нам не нужны.

Добавление новой информационной базы

В правой части окна запуска (см. рис. 1.9) расположен набор кнопок, которые позволяют перейти к созданию, удалению либо корректировке информационной базы. Технические действия для того, чтобы перейти к одному из этих действий, достаточно просты.

ПРИМЕЧАНИЕ

Обратим внимание на то, что при параллельном выполнении описываемых действий на компьютере следует соблюдать аккуратность и точно воспроизводить предлагаемые шаги.

В связи с тем, что сейчас мы создаем новую информационную базу, первое действие после появления на экране стартового окна заключается в щелчке на кнопке Добавить. Это приводит к появлению на экране следующего окна (рис. 1.10), в котором необходимо выбрать один из двух сценариев дальнейшего развития событий.

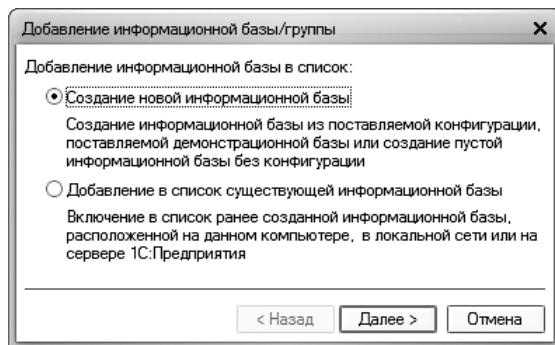


Рис. 1.10. Окно для добавления новой (выбора существующей) информационной базы

В данном случае наша цель заключается в разработке новой информационной базы, поэтому необходимо установить верхний переключатель, как показано на рис. 1.10. После этого, щелкнув на кнопке Далее, мы перейдем к окну, в котором

следует определиться со способом будущей разработки (рис. 1.11). Здесь верхний переключатель позволяет определить способ создания новой информационной базы на основе одного из имеющихся шаблонов. Например, если мы собираемся вести бухгалтерский учет, то для этого специалистами фирмы 1С разработан «специальный шаблон» (базовая конфигурация, включающая все необходимые объекты). В этом случае мы получаем готовую информационную систему для бухгалтерского учета, включающую все необходимые документы и отчеты. При подобном развитии событий для пользователя обеспечивается максимально удобный сервис ведения бухгалтерского учета в организации. Понятно, что если решаемая задача не относится к категории стандартных, то и стандартной конфигурации (шаблона) для нее нет. Сейчас наша задача заключается в самостоятельной разработке конфигурации, поэтому в окне на рис. 1.11 установим нижний переключатель, после чего щелкнем на кнопке Далее.

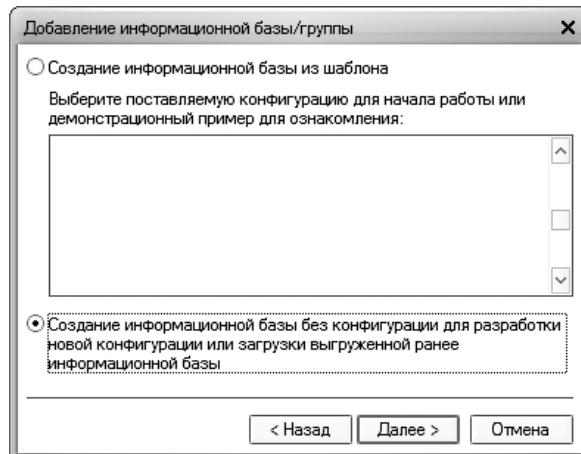


Рис. 1.11. Окно для выбора способа создания информационной базы

После выполненных действий на экране открывается очередное окно диалога с системой (рис. 1.12), где необходимо установить (выбрать) имя создаваемой информационной базы. Сейчас вместо варианта, предложенного системой по умолчанию, зададим другое название, например Глава 1. Если этот вариант вам не нравится, выберите любое другое название. После щелчка на кнопке Далее перед вами откроется следующее окно, где от нас потребуется указать каталог для создаваемой информационной базы (рис. 1.13). В этом случае система автоматически предлагает для выбора типовую папку. Однако можно поступить и по-другому — создать пустую папку в любом месте на компьютере либо в сети, а в окне диалога указать ее местоположение.

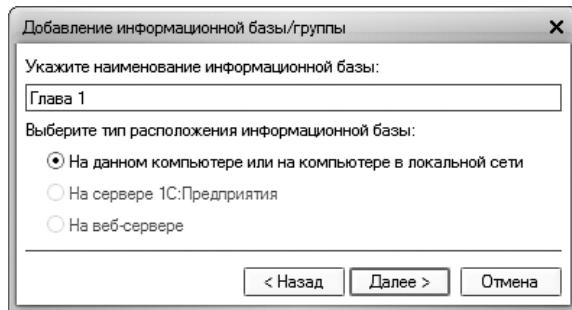


Рис. 1.12. Окно для указания наименования новой информационной базы

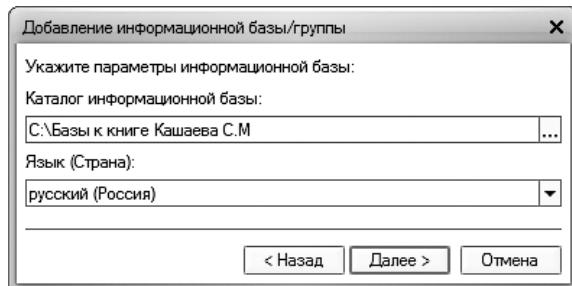


Рис. 1.13. Окно для указания каталога информационной базы

После щелчка на кнопке **Далее** перед нами откроется следующее окно, предназначенное для установки параметров запуска системы (рис. 1.14). В нашем случае изменения каких-либо установок выполнять не требуется, поэтому достаточно просто щелкнуть на кнопке **Готово**.

Последний шаг описанной последовательности действий приводит к выполнению технических процедур по созданию информационной базы, в результате на экране откроется уже знакомое окно (см. рис. 1.9), но с новой строкой, символизирующей добавление еще одной информационной базы. В дальнейшем, когда вы захотите продолжить начатую разработку и откроете стартовое окно программы, то в перечне информационных баз будет присутствовать и эта новая строка.

ПРИМЕЧАНИЕ

В нижней части стартового окна (см. рис. 1.9) при выделении информационной базы появляется каталог, где она находится.

Следующие технические действия для разработки конфигурации заключаются в том, что эту новую информационную базу необходимо выделить (щелкнуть на ее названии мышью), после чего выполнить щелчок на кнопке **Конфигуратор**.

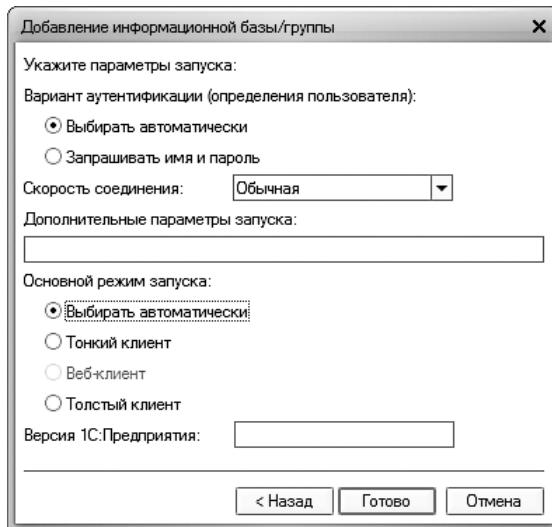


Рис. 1.14. Окно для указания параметров запуска системы

В результате мы переходим к следующему важному этапу работы, а именно — попадаем в окно конфигуратора (рис. 1.15), которое представляет собой среду для разработки прикладных решений. Фактически, это и есть тот инструмент, с помощью которого мы будем создавать конфигурацию для платформы 1С:Предприятие 8.3.

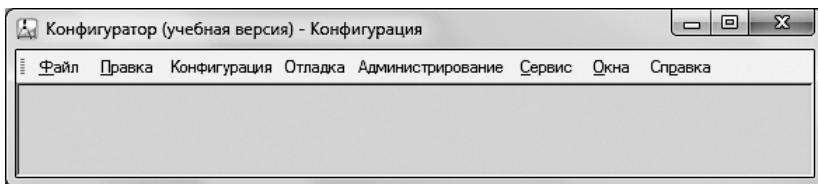


Рис. 1.15. Окно конфигуратора

Ранее уже упоминалось, что все прикладные решения для автоматизации различных участков деятельности в системе 1С:Предприятие созданы разработчиками именно с использованием конфигуратора. Заметим, что состав главного меню конфигуратора (**Файл**, **Правка**, **Сервис...**) типичен для современных программных продуктов.

После запуска конфигуратора на экране отображается только главное меню, расположено в верхней части окна. Следующее, что от нас требуется — выбрать в меню Конфигурация пункт Открыть конфигурацию. В результате данного шага на экране появится окно с деревом объектов конфигурации, которое представлено в левой части рис. 1.16. Сейчас в нем присутствуют только заголовки, но в дальнейшем в этом дереве будут автоматически появляться все создаваемые нами объекты конфигурации.

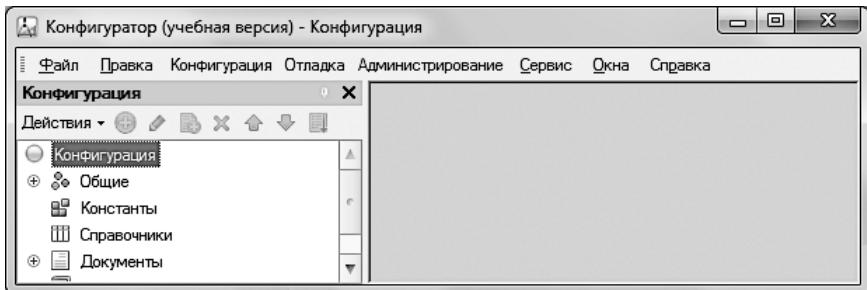


Рис. 1.16. Состав конфигурации

ПРИМЕЧАНИЕ

Каждая ветвь древовидной структуры в окне конфигурации описывает определенную тематическую группу конфигурации. Например, при создании новых справочников они будут группироваться в ветви Справочники, а в ветви Документы найдут свое место объекты конфигурации типа Документ. Технология работы с деревом конфигурации аналогична работе с подобными древовидными представлениями в операционной системе Microsoft Windows. Для открытия той или иной ветви (например, для обзора имеющихся в конфигурации справочников) достаточно щелкнуть на значке «плюс».

Подсистемы

Подсистема представляет собой основной элемент построения интерфейса информационной системы 1С:Предприятие. Как правило, любое более или менее серьезное прикладное решение подразумевает его применение различными категориями пользователей. Это могут быть менеджеры, бухгалтеры, управленческий персонал и т. д. В этом случае для каждой такой категории пользователей программа 1С:Предприятие 8.3 должна быть удобна в плане интерфейса. Удобство подразумевает доступность только тех элементов интерфейса, которые требуются конкретному пользователю. Так, менеджеру не нужны объекты, с которыми работает только бухгалтер, и наоборот. Для организации подобного сервиса и предназначены подсистемы. Разумеется, если конфигурация очень простая, то ее разбивка на подсистемы не требуется. Однако практически все реальные разработки при делении на подсистемы становятся более удобными для пользователей.

В этой главе мы рассмотрим создание конфигурации с включением в ее состав нескольких подсистем. Сейчас наша задача заключается в том, чтобы организовать три подсистемы, которые назовем:

- Управление;
- Поступление;
- Продажа.

Вернемся к практическим действиям и в окне Конфигурация обратимся к ветке объектов конфигурации Общие (рис. 1.17). Для этого достаточно раскрыть ее щелчком мыши на значке «плюс».

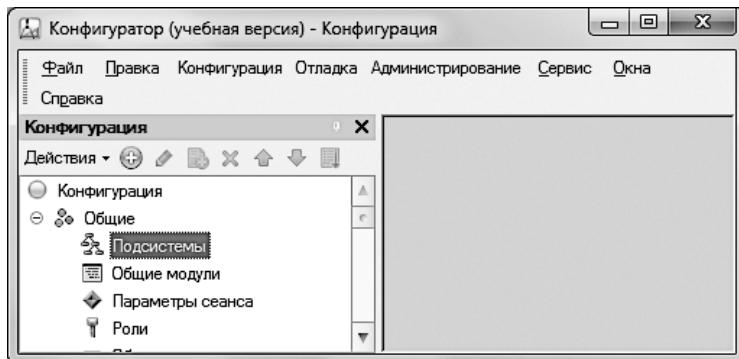


Рис. 1.17. Содержание ветки Общие в окне Конфигурация

Технические действия по созданию новой подсистемы достаточно просты. Для начала необходимо выделить ветвь Подсистемы и щелкнуть на ней правой кнопкой мыши. В результате на экране появится контекстное меню, где следует выбрать пункт Добавить. После этого вы увидите на экране новое окно — окно редактирования объекта конфигурации, которое предназначено для установки необходимых параметров объектов конфигурации (рис. 1.18).

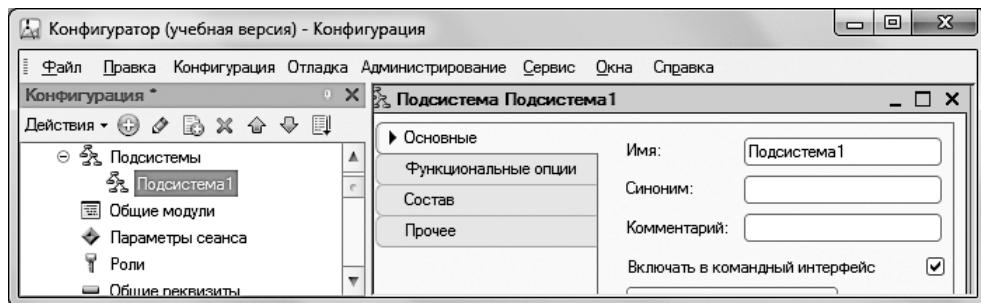


Рис. 1.18. Окно редактирования объекта конфигурации

Для навигации (перехода от вкладки к вкладке) в окне редактирования объекта конфигурации можно просто щелкнуть на названии необходимой вкладки (после некоторого опыта вы легко будете самостоятельно выбирать нужную вкладку). Сейчас от нас требуется внести слово Управление в поле Имя. В результате окно редактирования объекта конфигурации примет вид, показанный на рис. 1.19. Присутствующие в окне поля Синоним и Комментарий являются общими для боль-

шинства объектов конфигурации, и их назначение мы подробно прокомментируем далее в этой главе при рассмотрении справочников (это другой базовый объект конфигурации).

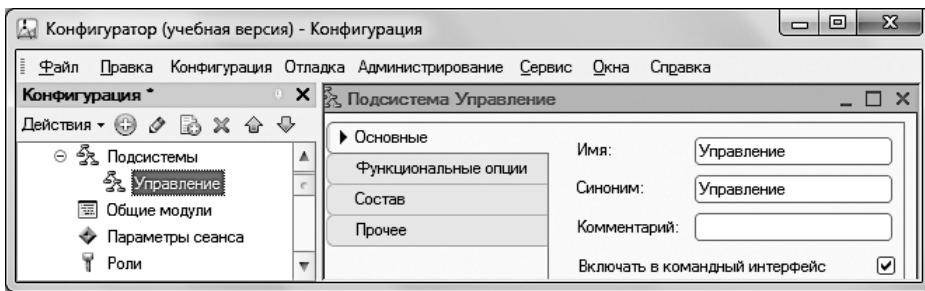


Рис. 1.19. Формирование подсистемы Управление

Аналогичным образом создадим еще две подсистемы, которые нам требуются в соответствии с поставленной задачей (рис. 1.20).

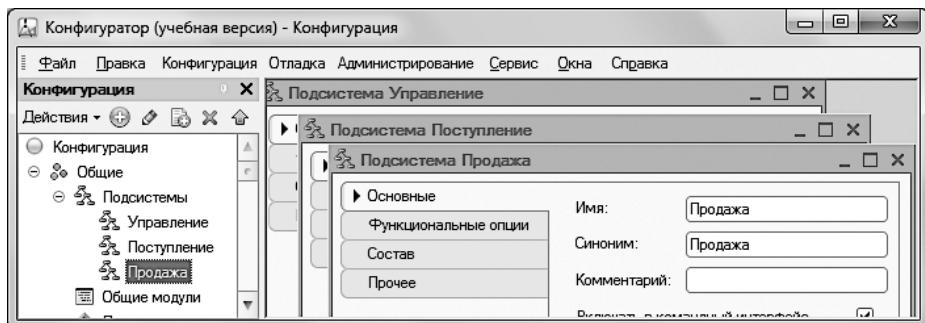


Рис. 1.20. Создание трех подсистем в конфигурации

Панель разделов приложения

Теперь после формирования в конфигураторе трех подсистем настало время посмотреть на результат выполненных действий. Для этого требуется перейти в режим 1С:Предприятие, что можно сделать с помощью команды Отладка ▶ Начать отладку. В результате после простого диалога с рядом промежуточных окон мы увидим на экране окно нашей прикладной задачи (рис. 1.21). Сразу под главным меню располагается *панель разделов приложения*, где и нашли место созданные нами подсистемы.

Разделы приложения организованы в виде ссылок, с помощью которых можно получить доступ к объектам, входящим в эти разделы. В данном случае разделы

пусты, что связано с тем, что мы еще не сформировали их состав в конфигураторе. Поэтому щелчки на ссылках любого из трех разделов не приводят каким-либо изменениям на экране.

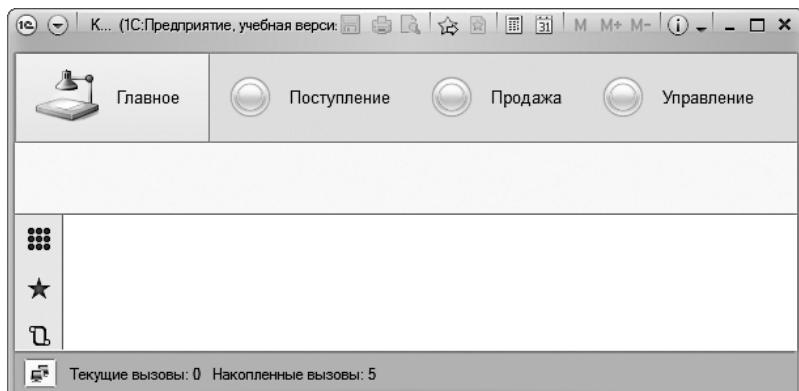


Рис. 1.21. Панель разделов приложения

ПРИМЕЧАНИЕ

Представленное на рис. 1.21 окно является основным окном приложения. С его помощью пользователь осуществляет навигацию по всему прикладному решению. Кроме основного окна при работе отображается вспомогательное окно приложения, которое обеспечивает работу с данными (просмотр и ввод документов, отчетов и т. д.).

Кроме созданных нами подсистем на рис. 1.21 присутствует также ссылка Главное. Данный раздел формируется системой автоматически и его следует использовать для размещения наиболее часто используемых команд.

Справочники

На данном этапе мы познакомились с режимом конфигуратора, и теперь наша задача заключается в создании нескольких справочников, которые понадобятся в нашей информационной базе.

В системе 1С:Предприятие имеется набор базовых объектов конфигурации, и сейчас мы поговорим об одном из этих объектов под названием *справочник*. Общепринято справочник ассоциировать с совокупностью строк, которые называются элементами справочника. При этом важно отметить, что в каждой строке справочника присутствует несколько столбцов и количество столбцов одинаково для каждой строки. Строки есть не что иное, как имеющиеся в справочнике записи (как уже говорилось

выше, более точное название — *элементы справочника*), а столбцы определяют структуру этих записей. Базовый объект конфигурации, справочник, фактически является шаблоном, на основании которого разработчики могут создавать свои объекты конфигурации — конкретные справочники, например Сотрудники, Должности, Номенклатура и т. д. Эти создаваемые справочники наследуют все характеристики *объекта конфигурации справочник*. Платформа на основании объекта конфигурации создает информационную структуру, в которой будут храниться конкретные списки (сотрудников, должностей и номенклатуры).

Использование *объекта конфигурации справочник* позволяет пользователю при работе в режиме 1С:Предприятие вносить данные в информационную структуру, созданную на основании этого объекта конфигурации. Данную информационную структуру также принято называть термином *справочник*. Далее мы так и будем поступать, хотя на самом деле *объект конфигурации справочник* и *программный объект справочник* представляют разные сущности. Из контекста в дальнейшем всегда будет понятно, о чем идет речь.

Справочники представляют собой перечни (списки) данных, которые весьма полезны в ситуациях, когда необходимо исключить неоднозначный ввод информации. Разумеется, эти перечни данных могут формироваться, корректироваться и дополняться в процессе работы в режиме 1С:Предприятие.

ПРИМЕЧАНИЕ

Система 1С:Предприятие 8.3 позволяет создать практически любое количество справочников. При этом в режиме конфигуратора создается структура любого справочника, а данными он заполняется уже в режиме 1С:Предприятие.

Формирование списка городов

В разрабатываемой в этой главе информационной базе мы будем формировать список фирм, с которыми наша организация сотрудничает (по закупке либо продаже мебели). Для отражения территориальной принадлежности фирм нам потребуется предварительно сформировать список городов. В связи с этим в качестве первого шага необходимо создать справочник со списком городов.

Ранее мы использовали слово *столбцы*, что отражает представление справочника в виде таблицы. Однако в системе 1С:Предприятие используется более общий термин *реквизиты* справочника (столбцов столько, сколько реквизитов у справочника). При этом в любом справочнике всегда присутствуют *обязательные* (в терминологии фирмы 1С — стандартные) реквизиты. Среди обязательных реквизитов важнейшими являются *код* и *наименование*. Кроме обязательных реквизитов разработчик может создавать в любом справочнике свои дополнительные реквизиты.

ВНИМАНИЕ

В каждой строке справочника всегда есть поле кода и поле наименования. Если же требуются дополнительные реквизиты, то их необходимо создать в конфигураторе в окне редактирования объекта конфигурации.

Таким образом, можно сказать, что упрощенно справочник в базе данных представляет собой таблицу, где каждому реквизиту соответствует свой столбец.

Теперь после формирования общего представления о структуре справочников продолжим практическую работу в режиме конфигуратора, а именно создадим справочник Города, в котором будут содержаться названия городов. Технические действия при разработке справочников достаточно просты. Так, для того чтобы создать новый справочник, необходимо щелкнуть правой кнопкой мыши на пункте Справочники в дереве объектов конфигурации и в появившемся контекстном меню выбрать единственный вариант – Добавить (рис. 1.22). После этого перед нами откроется уже знакомое окно редактирования объекта конфигурации (рис. 1.23). Однако учитывая, что в данном случае мы имеем дело с другим объектом конфигурации (не с подсистемой, как ранее), вид окна будет несколько другой (эти изменения касаются количества вкладок в окне). Некоторые вкладки справочника соответствуют вкладкам подсистемы, другие являются новыми.

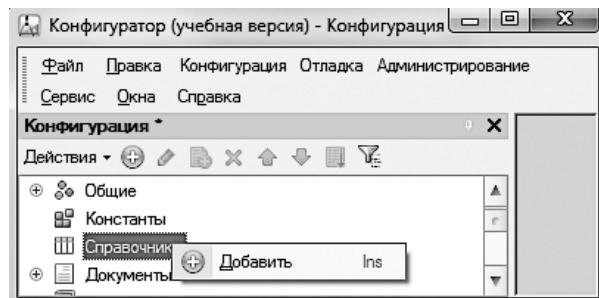


Рис. 1.22. Дерево объектов конфигурации

Структура окна редактирования объекта конфигурации в виде совокупности нескольких тематических вкладок удобна разработчику. Первый шаг заключается в том, что на вкладке Основные необходимо заполнить поле Имя, а что касается полей Синоним и Комментарий, то здесь мы их заполнять не будем, а само назначение этих полей поясняется в следующем примечании.

Итак, на вкладке Основные следует ввести имя, которое мы собираемся дать создаваемому справочнику – Города (рис. 1.24). Часть полей (Представление объекта, Расширенное представление объекта и т. д.) в окне редактирования не имеет существенного

значения при разработке учебных конфигураций (разработок, на которых мы только учимся). Разумеется, когда вы будете заниматься профессиональными разработками, то эти поля следует заполнять. Подробно их назначение описано в литературе [5].

ПРИМЕЧАНИЕ

В системе 1С:Предприятие для редактирования свойств основных объектов конфигурации (а также для настройки взаимодействия между объектами) предназначено окно редактирования объекта конфигурации. Редактируемые свойства располагаются на нескольких вкладках, а определенная вкладка предназначена для настройки свойств конкретной категории. Сами вкладки (их название и содержание) меняются в зависимости от типа объекта конфигурации (так, для документов вкладки свои, а, скажем, для справочников они несколько другие). Все основные объекты конфигурации, с которыми мы встретимся в книге, содержат вкладку Основные. Ключевое поле здесь — Имя, значение которого представляет собой идентификатор объекта конфигурации. Другое поле, Синоним, используется в различных ситуациях в режиме 1С:Предприятие как представление объекта, понятное пользователю. В частности, поле Синоним служит в качестве названия пункта меню, связанного с открытием конкретного справочника, документа, отчета и др. Дело в том, что на имя любого объекта накладывается ряд ограничений, например в нем не должно быть пробелов.

В этом случае при использовании сочетаний из нескольких слов имя объекта (например, документа ПоступлениеТоваров) смотрится несколько непривычно для нас. И для того чтобы при работе в режиме 1С:Предприятие объект ассоциировался с названием в более традиционной форме, следует использовать поле Синоним для замены имени в меню. На синоним как раз не накладывается подобных ограничений. Еще одно поле, Комментарий, носит вспомогательный характер и представляет собой произвольную строку символов, описывающих данный объект. В последующих примерах каким-то образом заполнять поле комментария мы не будем.

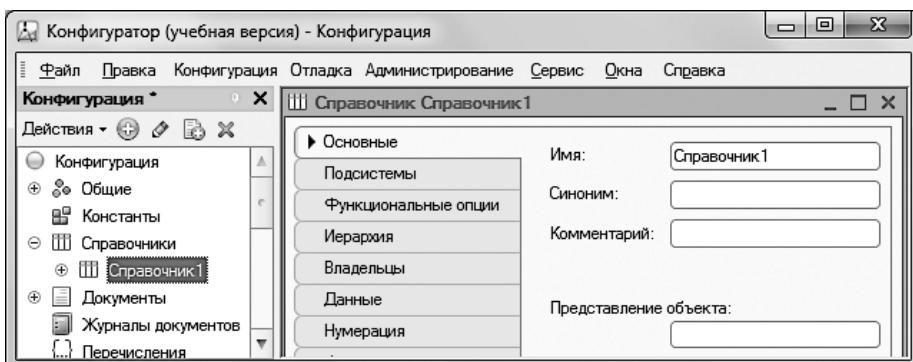


Рис. 1.23. Окно редактирования объекта конфигурации

Здесь для описания назначения задаваемых параметров прокомментируем лишь поле **Представление объекта**. Этот параметр определяет представление объекта в единственном числе и используется в названии стандартной команды. Например, для созданного нами объекта Города это может быть Город:создать. Представление объекта стоит задавать тогда, когда синоним конфигурации задан во множественном числе.

После задания имени на вкладке **Основные** перейдем на вкладку **Данные** и обратим внимание на два поля (рис. 1.25):

- Длина кода;
- Длина наименования.

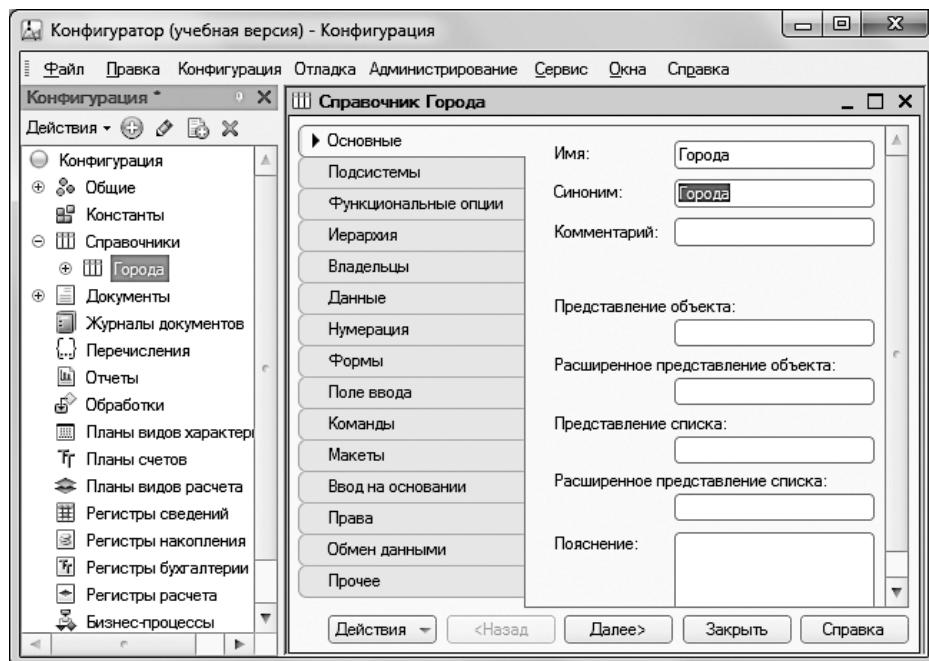


Рис. 1.24. Окно редактирования справочника Города

Длина кода определяет количество элементов, которые можно создать в рассматриваемом справочнике. В свою очередь, длина наименования определяет число символов, которые можно внести в качестве наименования элемента справочника.

Следующий важный момент заключается в отнесении созданного объекта конфигурации к одной из подсистем. Для этого в окне редактирования объекта конфигурации следует перейти на вкладку **Подсистемы** и указать, в каких подсистемах созданный справочник будет отображаться. В данном случае отнесем созданный справочник к подсистеме Управление (рис. 1.26).

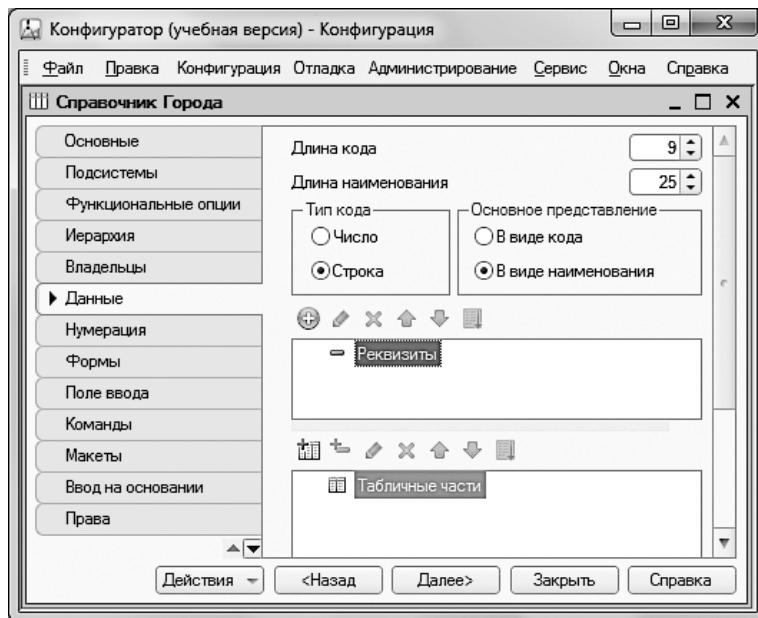


Рис. 1.25. Вкладка Данные окна редактирования справочника Города

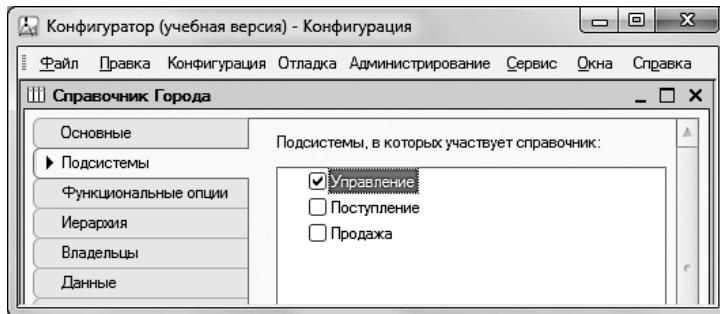


Рис. 1.26. Вкладка Подсистемы окна редактирования справочника Города

Таким образом, не выполнив никаких особо сложных действий, мы тем не менее создали новый объект конфигурации — справочник Города, с которым уже можно работать пользователю.

Следующий новый технический момент — *заполнение созданного справочника* в режиме 1С:Предприятие. Для этого в меню Отладка конфигуратора (см. рис. 1.26) следует выбрать пункт Продолжить отладку (первоначальное название Начать отладку автоматически изменяется системой), что позволит нам повторно перейти в режим 1С:Предприятие. В результате мы получим возможность работать с информационной базой, построенной системой на основании созданной конфигурации. В связи

с тем, что справочник Города был включен нами в подсистему Управление, там мы его и увидим (рис. 1.27).

После щелчка на ссылке Города откроется форма данного справочника со списком, который пока пуст (рис. 1.28).

Сейчас необходимо заполнить созданный справочник названиями нескольких городов. Это, с одной стороны, даст вам некоторый практический навык работы с системой в качестве пользователя, а с другой — заполненный информацией справочник потребуется для дальнейшей работы. Заметим, что при попытке добавления в справочник нового элемента (с помощью кнопки Создать) открывается экранная форма элемента (рис. 1.29).

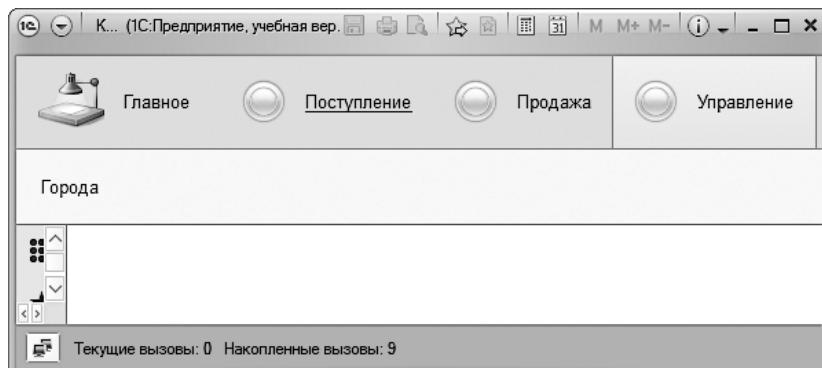


Рис. 1.27. Вид подсистемы Управление в режиме 1C:Предприятие

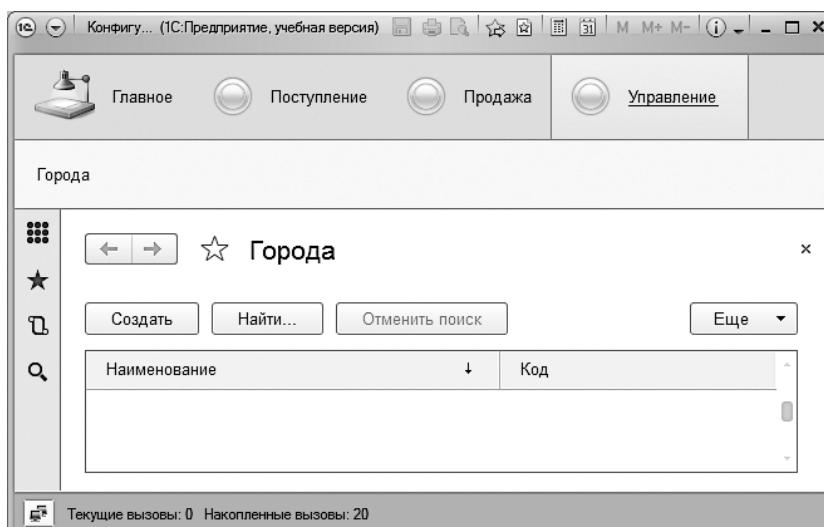


Рис. 1.28. Форма справочника Города

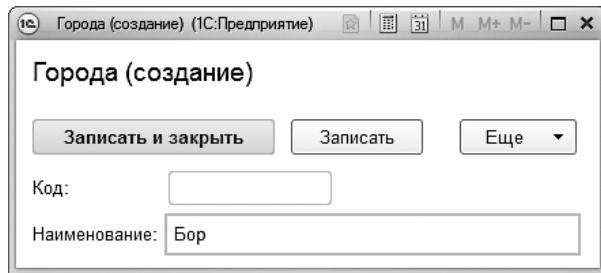


Рис. 1.29. Форма элемента справочника Города

При работе по вводу данных в справочник следует обратить внимание на один важный момент. Так, в процессе действий в режиме 1С:Предприятие пользователи просматривают и вводят информацию в справочники с помощью электронных форм (это справедливо и для других объектов). При этом, если мы в режиме конфигуратора не совершаляем никаких шагов в плане создания форм справочника (или других объектов), то система автоматически генерирует стандартные формы в режиме 1С:Предприятие, когда в этом появляется необходимость. В частности, экранная форма элемента, показанная на рис. 1.29, генерируется системой по умолчанию.

После внесения данных о нескольких городах форма справочника может выглядеть так, как показано на рис. 1.30.

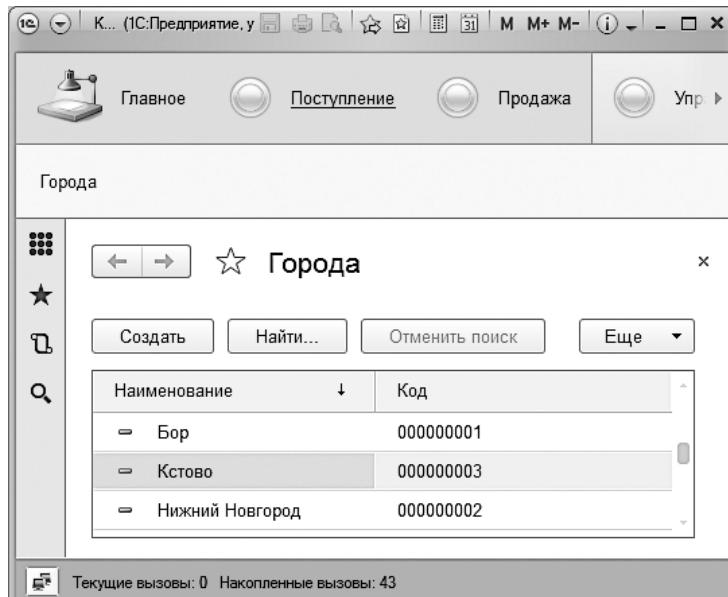


Рис. 1.30. Заполненная форма справочника Города

Таким образом, в режиме конфигуратора (в окне редактирования объекта конфигурации) для любого справочника можно ввести только имя на вкладке Основные и после этого практически сразу (разумеется, еще требуется отнести его к какой-нибудь подсистеме) начать работу по внесению информации в базу данных. Необходимые для этого программные объекты система создает автоматически. Эта возможность позволяет существенно ускорить процесс разработки, так как исключает задание дополнительных пользовательских сервисных функций помимо тех, которые автоматически создаются в формах, генерируемых системой по умолчанию.

ПРИМЕЧАНИЕ

Форма представляет собой вспомогательный объект системы 1С:Предприятие, который предназначен для ввода и просмотра информации. Форма может быть эффективно использована для организации работы с данными. При открытии определенной формы справочника, документа или регистра система формирует на ней необходимую информацию. В дальнейшем при вводе данных в элементы, расположенные на форме, они сохраняются в этих основных объектах (тех же справочниках, документах, регистрах и т. д.). Программный продукт 1С:Предприятие предоставляет пользователю возможность при желании отказаться от самостоятельного создания необходимых форм объектов, а действовать те, которые автоматически генерируются системой в режиме 1С:Предприятие. В этом случае, соответственно, и от разработчика требуется меньше ручных действий при создании прикладных решений.

Формирование списка организаций

Ранее мы уже упоминали, что справочник Города нам требуется для отражения территориальной принадлежности фирм, с которыми наша организация сотрудничает. В этом разделе наша задача заключается в разработке еще одного справочника — Фирмы (рис. 1.31).

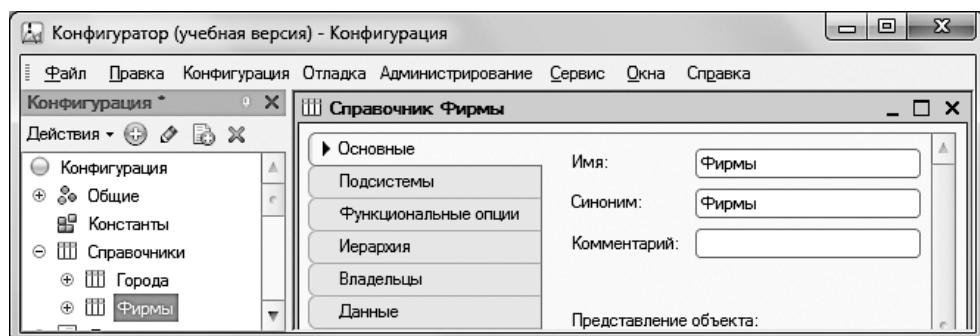


Рис. 1.31. Вкладка Основные окна редактирования справочника Фирмы

После указания имени справочника на вкладке **Основные** в окне редактирования объекта конфигурации перейдем на вкладку **Данные**, которая предназначена для определения реквизитов справочника (рис. 1.32). На открывшейся форме разработчик может установить необходимые параметры справочника. Так, в верхней части данной формы расположены элементы управления, позволяющие задать длину для кода и наименования. При создании нового справочника эти значения формируются системой автоматически, но при необходимости их можно изменить. Значение, устанавливаемое в качестве длины кода, как правило, для большинства ситуаций вполне достаточно. Что касается наименования (фактически — названия фирмы), то двадцать пять символов нас также вполне устроят. При необходимости это число можно увеличить (например, до 50 символов).

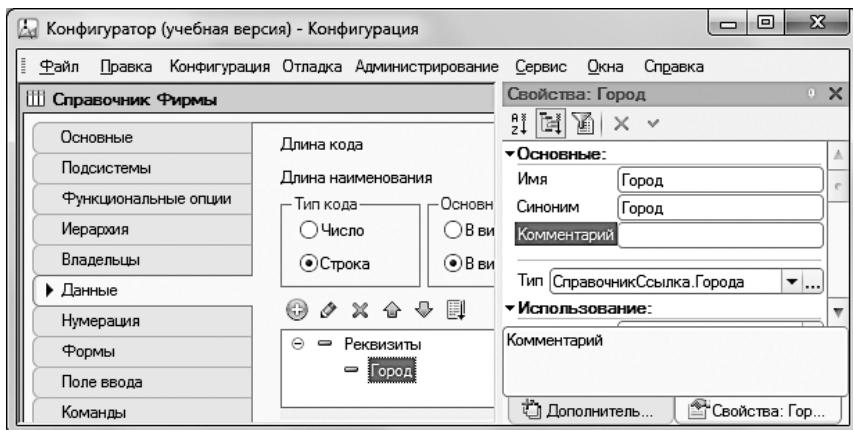


Рис. 1.32. Вкладка Данные окна редактирования справочника Фирмы

ПРИМЕЧАНИЕ

Код в справочнике служит для идентификации элементов справочника (записей справочника). Содержание поля кода должно быть уникально для каждого элемента справочника. В принципе, пользователю можно не задумываться о заполнении данного поля — в режиме 1С:Предприятие система сама обеспечивает автоматическое внесение кодов при создании новых элементов в справочнике. Однако при желании можно самостоятельно заполнять поле кода, исходя из определенных соображений. Например, содержание поля кода может помочь пользователю быстро идентифицировать записи (фактически в поле кода при желании можно ввести необходимую информацию об элементах справочника). Наименование представляет собой основную информацию об элементе справочника (например, если рассматривается справочник городов, то поле наименования предназначено для названия города). В разных конкретных ситуациях для него можно отводить разное число символов, однако существуют рекомендации компании 1С (или, можно даже сказать, норматив), касающиеся числа символов в наименовании [6].

Обратимся теперь к центральной части окна редактирования объекта конфигурации, где располагается поле **Реквизиты** (см. рис. 1.32). Данное поле предназначено для добавления необходимых реквизитов в справочник. Это действие можно выполнить с помощью самой левой кнопки (со знаком «плюс») верхнего пиктографического меню. С учетом нашей задачи в справочнике **Фирмы** создадим один дополнительный реквизит **Город** для хранения информации о городе, где расположена конкретная фирма, с которой мы ведем коммерческую деятельность.

Для каждого реквизита должен быть определен свой тип данных или, по-другому, тип значений, которые в режиме 1С:Предприятие можно для данного реквизита вносить. Для установки типа данных (а также и других параметров) предназначено **окно свойств** (показано в правой части на рис. 1.32), которое открывается двойным щелчком на имени реквизита. Другое название окна свойств — **палитра свойств**. В качестве типа данных для поля **Город** укажем вариант **СправочникСсылка.Города**. В этом случае при работе в режиме 1С:Предприятие при попытке заполнения поля **Город** будет автоматически открываться форма справочника **Города**.

ПРИМЕЧАНИЕ

При создании нового объекта конфигурации справочника Города система автоматически создает новый тип данных **СправочникСсылка.Города**. Это позволяет хранить в переменной ссылку на определенный элемент справочника.

Таким образом, мы рассмотрели техническую возможность добавления новых реквизитов в справочник. В результате этого справочники становятся более информативными. Например, в справочник сотрудников можно добавлять реквизиты для фиксирования их адресов и телефонов.

ПРИМЕЧАНИЕ

Табличные части также являются объектами конфигурации, только они относятся к категории не основных, а вспомогательных объектов.

Для каждой строки (каждого элемента) справочника можно создать одну или несколько подчиненных табличных частей (таблиц). Сейчас воспользуемся этой возможностью и добавим в справочник **Фирмы** табличную часть с именем **КонтактныеЛица** (рис. 1.33).

Ситуация, когда в разных организациях мы используем разное число лиц для коммерческих контактов, вполне реальна. Если с фирмами-партнерами ведется очень активный товарооборот, то приходится общаться с большим числом сотрудников (директор, заместитель директора, менеджеры, главный бухгалтер и т. д.). В то же время для мелких организаций можно ограничиться одним-двумя лицами. В подобной ситуации табличная часть справочника для хранения подобной информации подходит идеально, поэтому далее мы реализуем ее на практике.

Следующее действие в режиме конфигуратора заключается в добавлении в табличную часть КонтактныеЛица нескольких реквизитов (это действие выполняется с помощью второй кнопки слева в пиктографическом меню над табличной частью):

- Сотрудник — фамилия, имя и отчество конкретного сотрудника фирмы (в качестве типа данных укажем строку длиной не более 50 символов);

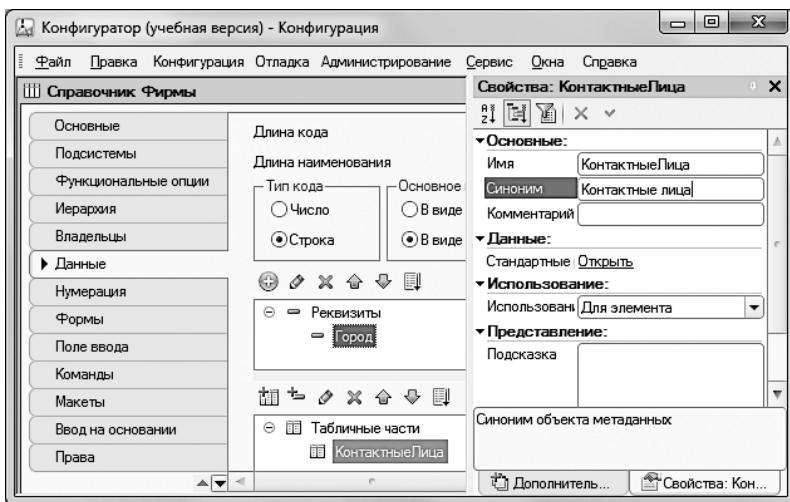


Рис. 1.33. Размещение табличной части в справочнике Фирмы

- Должность — должность специалиста (тип данных — строка длиной не более 25 символов);
- Телефон (тип данных — строка длиной не более 25 символов).

На рис. 1.34 показана вкладка Данные окна редактирования объекта конфигурации после внесения указанной информации о реквизитах.

Обратим внимание на вкладку Иерархия (рис. 1.35). Часто элементы справочника удобно классифицировать по группам (категориям), например сотрудников — по отделам, товары — по производителям, и т. д. В этом случае на вкладке Иерархия необходимо установить флагок Иерархический справочник. Выберем вариант иерархии Иерархия групп и элементов (наиболее часто используемый вариант). В результате при работе в режиме 1С:Предприятие можно создавать в справочнике категории элементов (для этого в пиктографическом меню формы списка справочника появляется соответствующий значок — Создать группу).

Итак, на вкладке Иерархия установим с помощью флагка иерархический вид данного справочника. В этом случае мы сможем создавать в рассматриваемом справочнике группы, например отдельные группы для покупателей и продавцов.

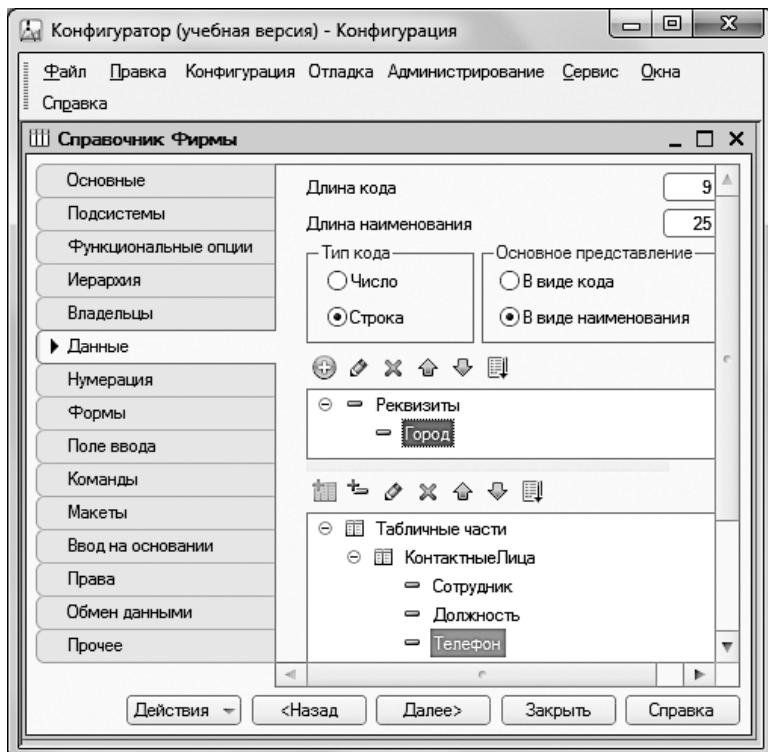


Рис. 1.34. Внесение реквизитов в табличную часть справочника Фирмы

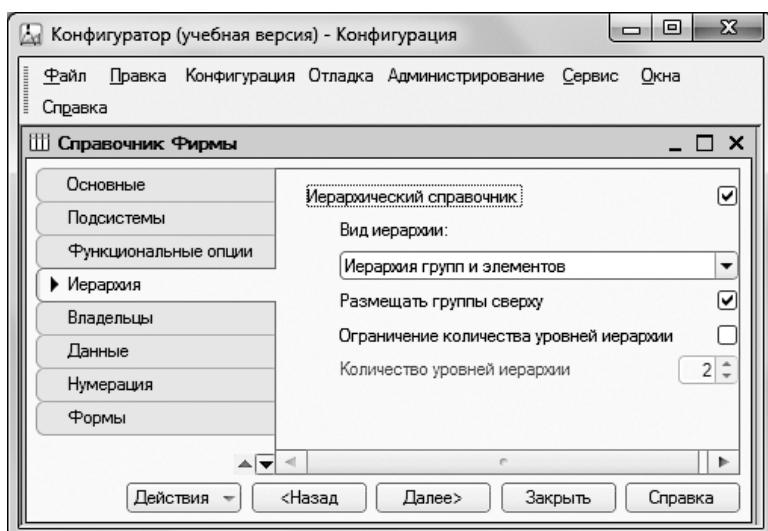


Рис. 1.35. Вкладка Иерархия окна редактирования справочника Фирмы

ПРИМЕЧАНИЕ

Количество уровней иерархии в справочнике можно ограничить, что производится установкой соответствующего флажка и выбора необходимого значения в счетчике Количество уровней иерархии. Если же данный флажок сброшен, то максимальное количество уровней вложенности справочника ничем не ограничивается.

Также на вкладке Подсистемы включим данный справочник во все три имеющиеся у нас подсистемы.

Подведем промежуточный итог выполненных действий. Итак, мы разработали структуру нового объекта конфигурации – справочника Фирмы, и теперь следует подумать о том, как пользователь будет с ним работать. Для этого в информационной системе существуют электронные формы справочников, а инструментальные средства позволяют пользователю создавать необходимые формы для работы в режиме 1С:Предприятие.

Как уже упоминалось, для ускорения процесса разработки в системе 1С:Предприятие 8.3 автоматически создаются формы, предлагаемые по умолчанию. Ранее при разработке справочника Города мы уже пользовались этой возможностью. Данный способ очень удобен, когда после разработки справочника (или какого-либо другого объекта) необходимо как можно быстрее начать с ним работу (либо продолжить работу в режиме конфигуратора по созданию других объектов). В дальнейшем (в этой главе) мы будем разрабатывать свои пользовательские формы, а сейчас еще раз воспользуемся формами, генерируемыми системой по умолчанию.

В форме справочника информация представлена в виде таблицы, где на каждую запись отводится одна строка (это обычный вид любого справочника). Основным достоинством подобной формы является то, что в этом случае на экране видны сразу несколько записей (элементов) справочника.

В то же время, недостаток такой формы становится очевидным, когда в справочнике много столбцов (большое количество реквизитов), а часто желательно, чтобы все они одновременно были бы видны пользователю. При практической работе в этом случае форму часто приходится прокручивать, чтобы увидеть необходимую информацию. Для решения проблемы применяется еще одна стандартная форма – *форма элемента*, которая позволяет отображать на экране только одну запись справочника. Но при этом, даже если реквизитов в справочнике много, все они видны на экране.

ПРИМЕЧАНИЕ

В реальной работе, как правило, используется большое количество разнообразных форм. *Объект конфигурации форма* ориентирован исключительно на пользователя, предоставляя ему удобный интерфейс при работе с основными объектами конфигурации.

У формы элемента имеется еще одно существенное достоинство. Так, если в справочнике присутствует табличная часть (как в только что созданном справочнике Фирмы), то в форме списка она не отображается, но видна в форме элемента.

На рис. 1.36 показан вариант заполнения формы элемента в справочнике Фирмы информацией по одной из организаций. Разумеется, сам процесс заполнения происходит в режиме 1С:Предприятие (в него следует перейти через меню Отладка). Видно, что в этом случае при редактировании элемента нам доступна и табличная часть справочника Фирмы.

Рис. 1.36. Заполнение формы элемента справочника Фирмы

ПРИМЕЧАНИЕ

Каждый справочник может иметь неограниченное (практически) число табличных частей.

Воспользуемся тем, что данный справочник мы определили как иерархический. В форме списка справочника Фирмы присутствует кнопка Создать группу (рис. 1.37). С помощью этого элемента управления создадим в справочнике Фирмы две группы:

- Поставщики;
- Покупатели.

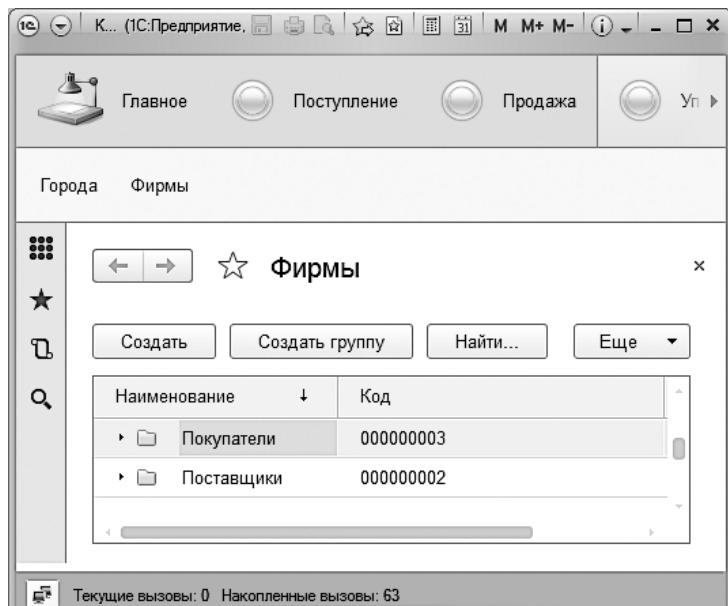


Рис. 1.37. Создание групп в справочнике Фирмы

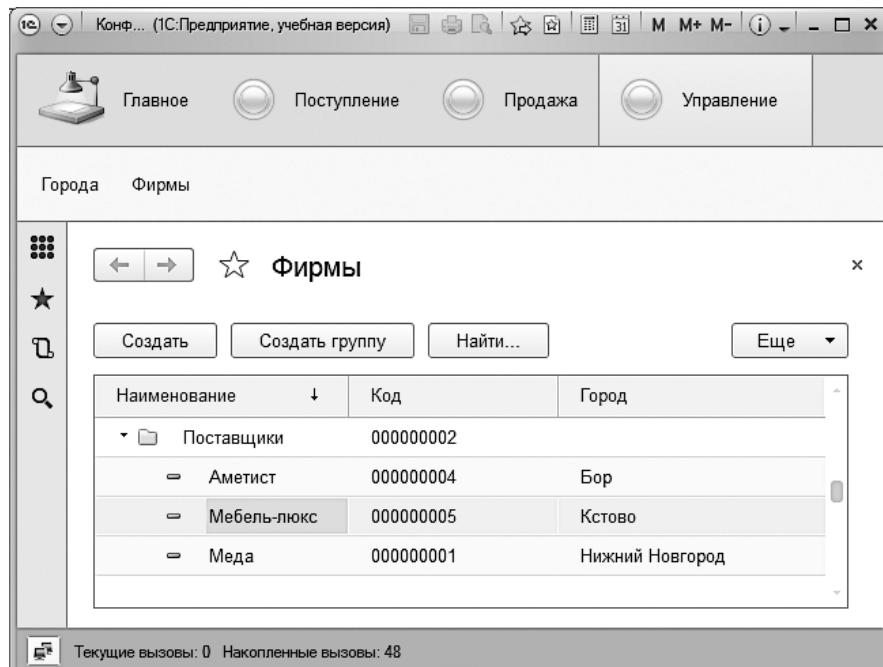


Рис. 1.38. Форма списка справочника Фирмы

Для того чтобы перевести уже имеющийся элемент справочника (показан на рис. 1.36) в категорию поставщиков, следует воспользоваться контекстным меню. В этом случае на экране открывается простое и понятное окно диалога по переносу элемента в указанную группу.

Далее необходимо внести несколько названий фирм. Один из вариантов заполнения рассматриваемого справочника **Фирмы** показан на рис. 1.38.

На данном этапе мы познакомились с технологией разработки структуры справочников (объектов конфигурации) в режиме конфигуратора, а также узнали, каким образом заполнять справочники данными с помощью электронных форм, создаваемых системой по умолчанию. На этапе разработки и отладки прикладных решений такой вариант использования автоматически генерируемых форм в режиме 1С:Предприятие применяется достаточно часто.

Размещение информации о товарах

Перейдем теперь к созданию следующего справочника, который предназначен для размещения информации о товарах. Опять обратимся к конфигуратору и приступим к разработке нового объекта конфигурации. Основная новизна в плане разработки связана здесь с созданием пользовательской формы списка справочника.

После щелчка правой кнопкой мыши на строке Справочники выберем пункт **Добавить** и на вкладке **Основные** в поле **Имя** укажем — **Товары**. В качестве дополнительных реквизитов в создаваемом справочнике укажем два:

- Описание** — краткое описание основных характеристик товара (тип данных — строка, 100 символов);
- Основной Поставщик** — основной поставщик (если их много) данного товара (тип данных — СправочникСсылка.Фирмы).

После этого на вкладке **Подсистемы** сделаем отметку о включении данного справочника в подсистему **Управление**.

Следующая вкладка, на которой понадобится сделать необходимые установки — это **Формы** (рис. 1.39). На ней для создания новой формы справочника следует использовать **конструктором форм**. С помощью конструктора можно также отредактировать уже имеющуюся форму. Это весьма полезный ресурс для автоматизации разработок прикладных решений на платформе 1С:Предприятие.

Сейчас наша задача заключается в создании пользовательской формы списка. Ранее мы пользовались формами, генерируемыми системой по умолчанию, но теперь поступим по-другому и создадим форму самостоятельно. Для этого необходимо щелкнуть на значке лупы, расположенной напротив пункта **Списка** в разделе **Формы** (см. рис. 1.39). В результате мы перейдем к работе с **конструктором формы** для справочника.

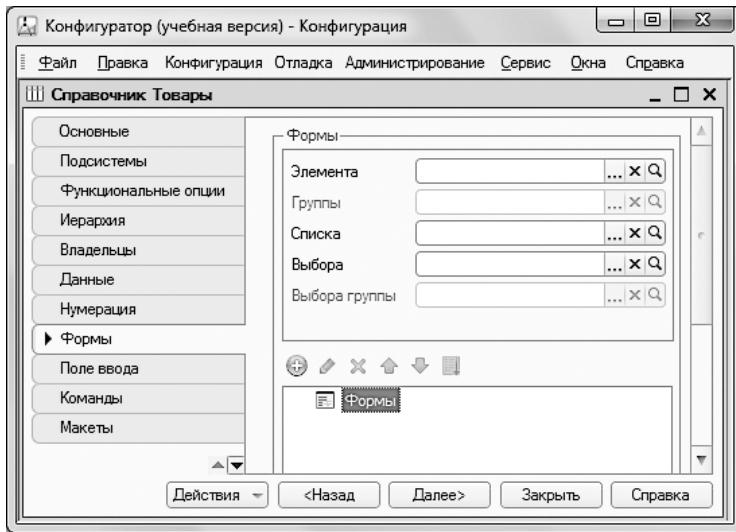


Рис. 1.39. Вкладка Формы окна редактирования объекта конфигурации

В первом окне диалога с конструктором следует установить параметры в соответствии с рис. 1.40 (если вы параллельно выполняете описанные действия на компьютере, то эти параметры устанавливаются конструктором по умолчанию).

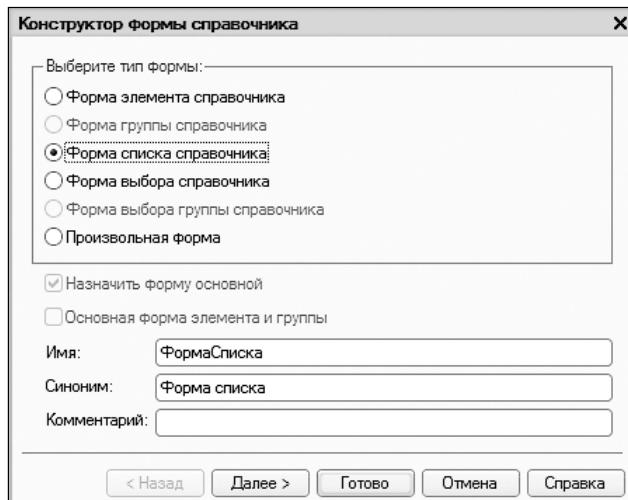


Рис. 1.40. Окно конструктора формы справочника для выбора типа формы

На рис. 1.40 переключатели в группе **Выберите тип формы** предназначены для выбора варианта электронной формы. Состав переключателей определяется видом

основного объекта конфигурации, для которого мы создаем форму. Скажем, для справочников они одни, для документов — другие. Число форм для каждого типа не ограничено, а при наличии у объекта нескольких форм одного типа одну из них следует назначить в качестве *основной*. Для этого необходимо установить флажок **Назначить форму основной**.

После щелчка на кнопке **Далее** в следующем окне нам необходимо с помощью флаjkов включить в создаваемую форму справочника код, наименование, а также дополнительные реквизиты **Описание** и **ОсновнойПоставщик** (рис. 1.41).

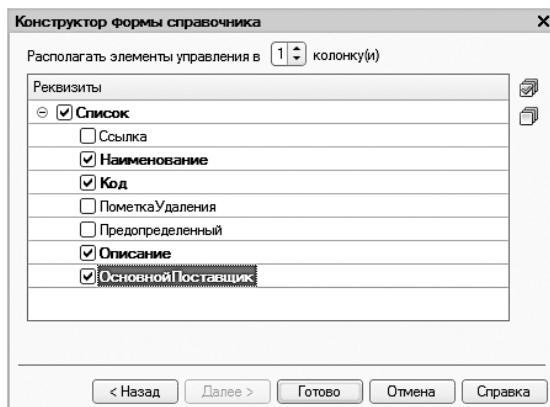


Рис. 1.41. Окно конструктора, предназначенное для включения реквизитов в форму

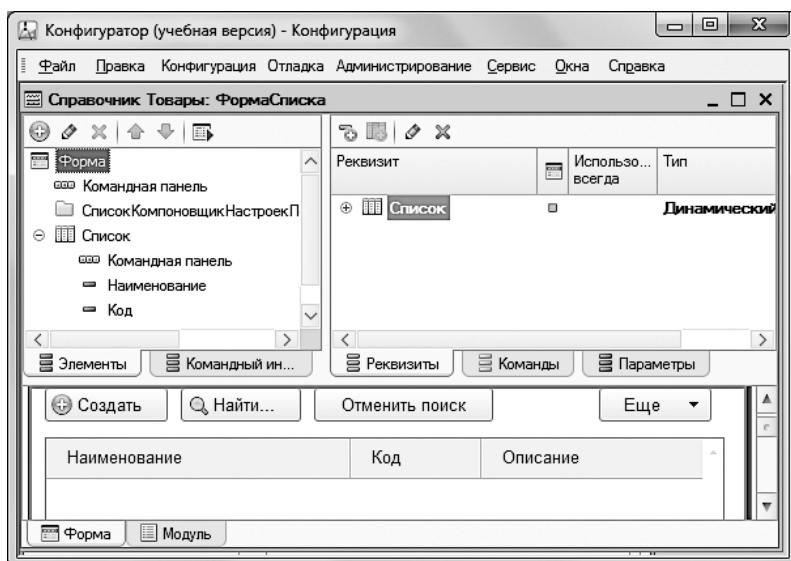


Рис. 1.42. Форма списка справочника Товары в редакторе форм

В результате после щелчка на кнопке Готово откроется окно редактора форм (рис. 1.42). Здесь в левой части окна отражены элементы, входящие в состав формы списка справочника Товары.

Сделаем небольшое изменение в созданной форме — поменяем подпись Наименование на Товар. После такого изменения вид справочника Товары в режиме 1С:Предприятие для пользователя будет более понятен. Для того чтобы выполнить данное изменение, необходимо с помощью правой кнопки мыши открыть контекстное меню поля Наименование и обратиться к окну свойств. Здесь в разделе Заголовок требуется задать новый вариант подписи для столбца товаров — Товар (рис. 1.43).

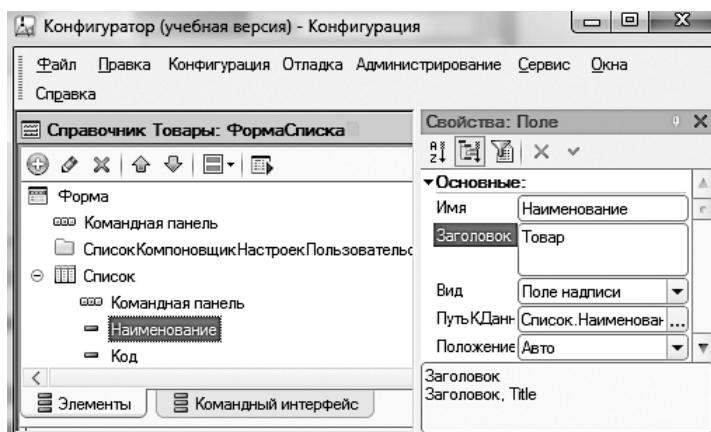


Рис. 1.43. Окно свойств поля Наименование

The screenshot shows the 'Товары' list form. At the top, there are buttons for 'Создать' (Create), 'Найти...' (Find), 'Отменить поиск' (Cancel search), and 'Еще' (More). The table below lists three items:

Товар	Код	Описание	Основной поставщик
Прихожая "Уют"	000000001	Состав: 4 элемента. Цвета: ...	Мебель-люкс
Стенка "Светлопляр"	000000002	Длина 3,5 м. Высота 2,7 м.	Меда
Стол кухонный	000000003	160 на 70 см. Цвет белый.	Меда

Рис. 1.44. Заполненная форма списка справочника Товары

Если скептически взглянуть на выполненную работу, то можно сказать: наши действия привели к созданию формы, на первый взгляд, аналогичной той, с которой

мы уже встречались (форма списка, генерируемая по умолчанию). Все изменения заключаются в другом заголовке. Однако далее в этой главе мы увидим и более существенное изменение, которое позволяет реализовать пользовательская форма. А пока перейдем в режим 1С:Предприятие и заполним справочник Товары информацией (рис. 1.44).

ПРИМЕЧАНИЕ

Плюсы пользовательской формы заключаются в возможности дополнения функциональных возможностей расположенных на ней элементов управления.

Филиалы и менеджеры

Для дальнейшей работы нам понадобится еще два справочника, один из которых назовем Филиалы. Данный справочник будет простой без дополнительных реквизитов, в связи с чем каких-то комментариев относительно его разработки в режиме конфигуратора не требуется. Включим данный справочник в подсистему Управление и затем перейдем в режим 1С:Предприятие для внесения в справочник данных. На рис. 1.45 данный справочник показан после заполнения.

Рис. 1.45. Форма списка справочника Филиалы

Последний справочник в данной конфигурации — Менеджеры, в котором также не требуется создавать дополнительные реквизиты. Его мы также включим в подсистему Управление. На рис. 1.46 приведен вариант внесения данных в справочник (информационную структуру, созданную системой на основании разработанного объекта конфигурации).

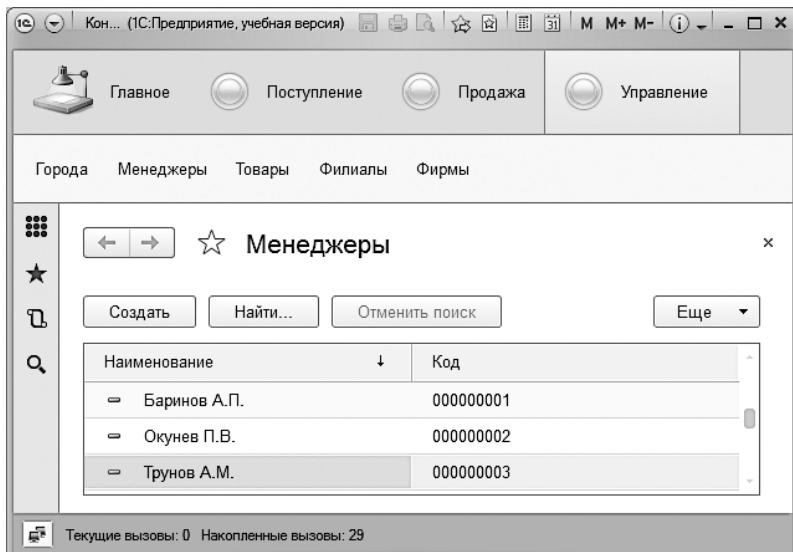


Рис. 1.46. Форма списка справочника Менеджеры

Перечисления

На данный момент мы познакомились с таким объектом конфигурации, как справочник. Этот объект относится к категории *основных* объектов конфигурации. Другим, схожим по функционалу объектом является *перечисление*. В целом перечисления представляют собой статические списки данных. Эти списки создаются на этапе разработки (в режиме конфигуратора) и в дальнейшем просто используются без возможности их изменения или дополнения в режиме 1С:Предприятие.

ПРИМЕЧАНИЕ

Основное назначение перечислений — исключить неоднозначность вводимой пользователем информации.

В конфигурации перечисления, как правило, используются не самостоятельно, а в совокупности с другими типами данных. Так, перечисления применяются для ввода ограниченного количества значений реквизитов справочников и документов.

Создадим перечисление с именем Категории, которое будет содержать категории товаров. Использование здесь перечисления связано с тем, что категории товаров не подразумевают какого-то большого разнообразия возможных значений. Кроме того, набор категорий товаров, как правило, продолжительное время не изменяется (если происходит добавление какого-то нового значения, то это случается очень редко). Фактически, основной объект конфигурации *перечисление* и был разработан для подобных ситуаций программистами фирмы «1С». В нашей ситуации в качестве значений перечисления Категории примем следующие:

- МягкаяМебель;
- Кухни;
- Прихожие;
- Стенки;
- Спальни.

Перейдем теперь к техническим действиям, позволяющим реализовать сказанное на практике. В режиме конфигуратора для создания нового перечисления необходимо в дереве объектов конфигурации сначала щелкнуть правой кнопкой мыши на элементе Перечисления и в контекстном меню выбрать пункт Добавить. После этого в окне *редактирования объекта конфигурации* следует ввести имя перечисления и указанные здесь значения (рис. 1.47).

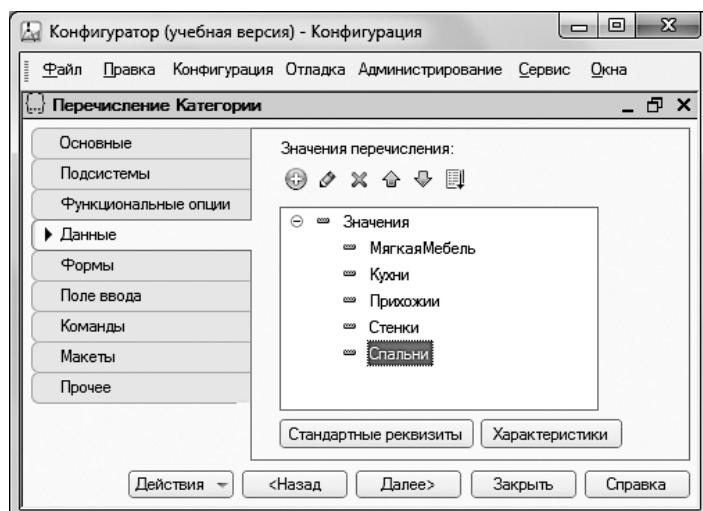


Рис. 1.47. Перечисление Категории

Теперь подкорректируем созданный ранее объект конфигурации — справочник Товары, а именно добавим в него реквизит Категории (тип данных — Перечисление-Ссылка.Категории), как показано на рис. 1.48.

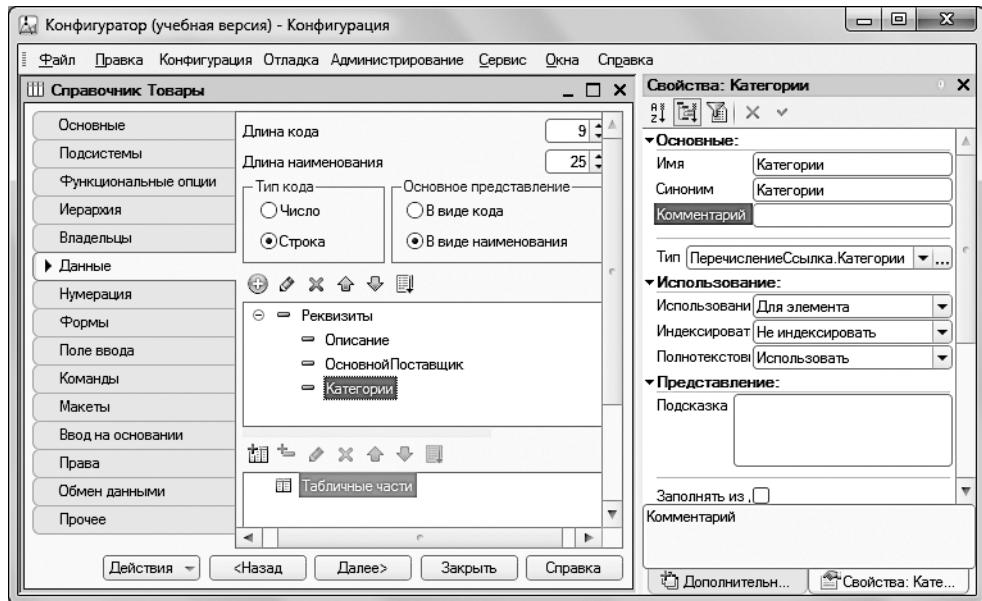


Рис. 1.48. Добавление реквизита в справочник Товары

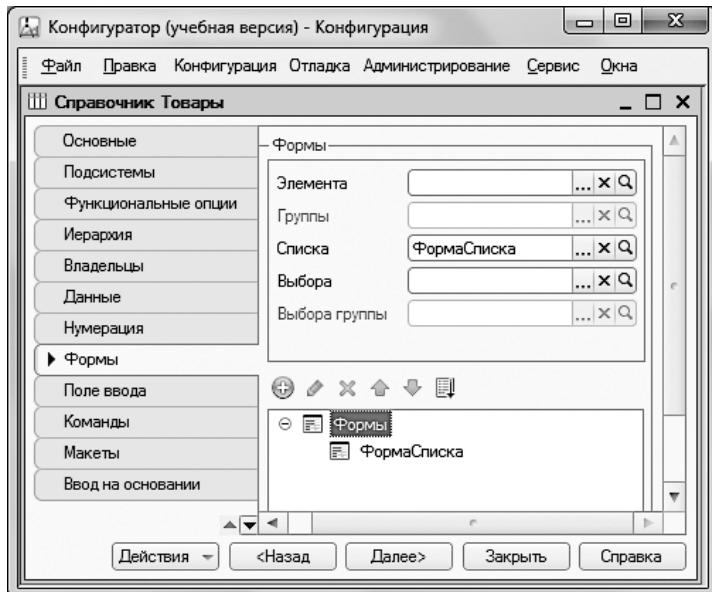


Рис. 1.49. Вкладка Формы справочника Товары

Учитывая, что форма списка справочника Товары была создана ранее без данного реквизита, то ее проще всего удалить и создать новую. Для удаления предыдущей

формы списка достаточно удалить ее в разделе **Формы** с помощью соответствующего значка (рис. 1.49). После этого уже знакомыми приемами создадим новую форму списка справочника **Товары**.

Далее следует перейти в режим 1С:Предприятие и внести сведения в поле **Категория** справочника **Товары** (рис. 1.50).

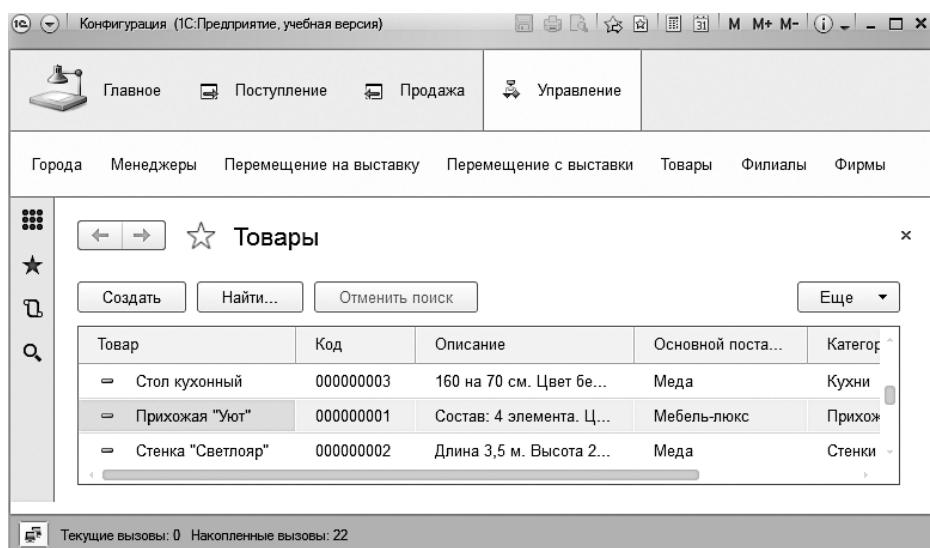


Рис. 1.50. Обновленная форма списка справочника Товары

Документы

Сразу отметим, что документы относятся к наиболее популярным объектам конфигурации. Они предназначены для фиксирования информации о происходящих событиях в подразделении или в организации в целом. На основании *документа* разработчик создает собственные объекты конфигурации – документы конкретной прикладной направленности. Разработанные в режиме конфигуратора объекты сохраняют все свойства *объекта конфигурации документ*. После этого, уже располагая новыми объектами конфигурации (например, документом **ПриходнаяНакладная**), платформа 1С:Предприятие создает в базе данных информационную структуру, в которой будут храниться компьютерные аналоги реальных документов.

Функциональность документа отличается от функциональности всех остальных объектов информационной базы. Это отличие связано с тем, что *документ обладает способностью проведения*. Факт проведения документа означает, что событие, которое он отражает, привело к изменению состояния учета.

ПРИМЕЧАНИЕ

На организацию учета в системе 1С:Предприятие 8.3 также ориентированы такие объекты конфигурации, как регистры, о которых мы поговорим в этой главе чуть позднее.

В связи с тем, что документ обладает способностью вносить изменения в состояние учета, он всегда привязан к конкретному моменту времени. В каждом документе содержится информация, которая описывает его в целом, — в любом документе автоматически создаются два реквизита:

- Дата** (для отражения даты документа);
- Номер** (для последовательной нумерации документов одного типа).

Реквизит **Дата** содержит дату и время с точностью до секунды — этот реквизит определяет положение документа на оси времени. Наша задача в этом разделе заключается в том, чтобы познакомиться с конструкцией документов и некоторыми техническими приемами работы с ними. Первый рассматриваемый далее пример будет связан с документом, который фиксирует информацию о поступлении товаров.

Учет поступлений товаров

В определенной степени действия по разработке документа в режиме конфигуратора аналогичны созданию справочников. Первое уже предсказуемое действие заключается в щелчке правой кнопкой мыши на значке **Документы** в окне Конфигурация (рис. 1.51). В результате открывается контекстное меню и после выбора пункта **Добавить** начинается техническая работа в окне редактирования объекта конфигурации.

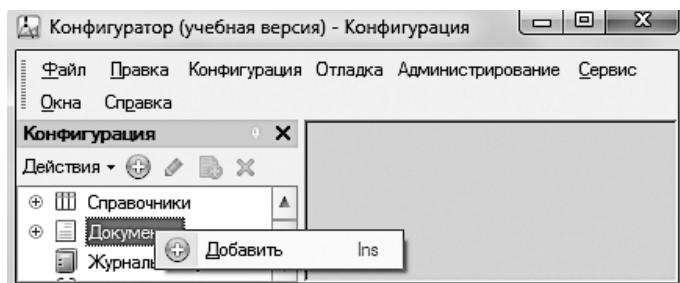


Рис. 1.51. Контекстное меню значка Документы в окне Конфигурация

В качестве имени создаваемого объекта конфигурации укажем **ПоступлениеТоваров** (рис. 1.52). Информационная структура, создаваемая системой на основании данного объекта конфигурации, будет фиксировать факт поступления товаров от одной из наших фирм-партнеров.

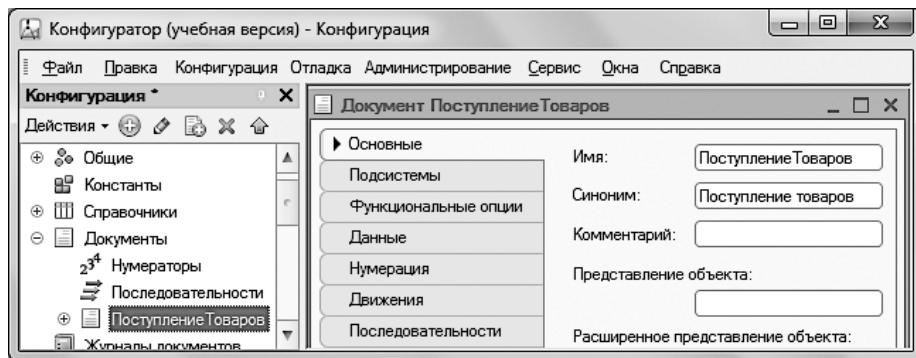


Рис. 1.52. Вкладка Основные окна редактирования документа ПоступлениеТоваров

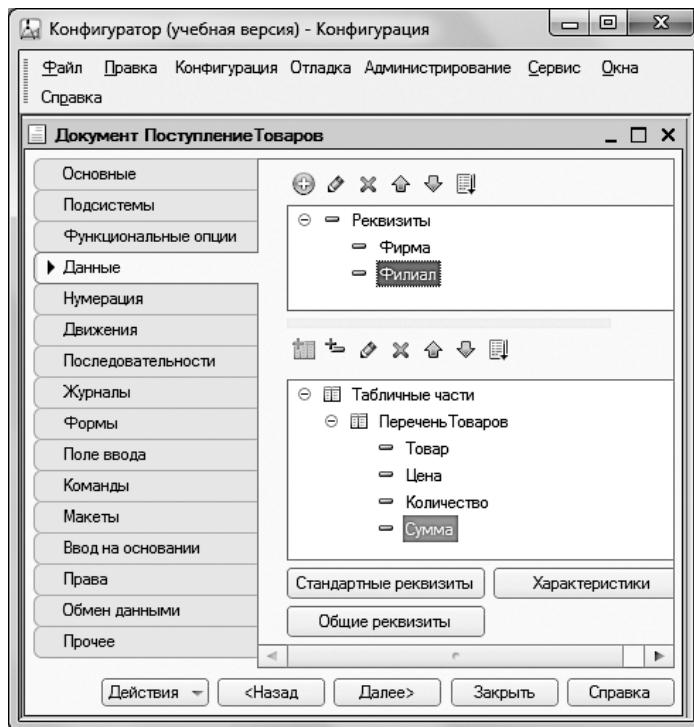


Рис. 1.53. Вкладка Данные окна редактирования документа ПоступлениеТоваров

На вкладке **Данные** следует определить поля создаваемого объекта конфигурации (рис. 1.53). Так, любой документ имеет шапку (заголовочную часть), а также одну или несколько табличных частей. В заголовочной части размещается общая информация для документа в целом, а табличная часть предназначена для перечня однотипных данных.

Для создаваемого объекта конфигурации ПоступлениеТоваров в области шапки разместим реквизит — **Фирма**, в качестве типа данных которого установим **СправочникСсылка.Фирмы**. Также в области шапки поместим еще один реквизит **Филиал** (тип данных — **СправочникСсылка.Филиалы**).

ПРИМЕЧАНИЕ

В данном примере рассматривается ситуация, в которой определенный документ фиксирует поступление товаров от одной фирмы на конкретный филиал.

Каждый реальный (бумажный или электронный) документ кроме заголовочной части, как правило, содержит фрагмент информации, которая одинакова по своей структуре. Например, в создаваемом документе ПоступлениеТоваров по логике должен содержаться перечень названий товаров с указанием цены и количества. Для подобной информации предназначен объект конфигурации *табличная часть*, поэтому далее перейдем к разработке табличной части документа (в создаваемом документе будет одна табличная часть).

ПРИМЕЧАНИЕ

Табличные части — это объекты конфигурации, которые являются подчиненными по отношению к основным объектам. С ними мы уже встречались при работе со справочниками. И в том, и в этом случае табличные части подчинены основным объектам (справочнику или документу).

На практике для документов с несложной структурой используется одна табличная часть, но в типовых конфигурациях, созданных программистами компании 1С, вы найдете немало документов с несколькими такими частями.

В разрабатываемом документе ПоступлениеТоваров создадим табличную часть, которую назовем **ПереченьТоваров**, а уже в ней разместим набор реквизитов (в скобках указаны соответствующие для них типы данных):

- Товар (тип данных — СправочникСсылка.Товары);
- Цена (тип данных — число);
- Количество (тип данных — число);
- Сумма (тип данных — число).

Форма документа

После определения структуры документа можно перейти к разработке интерфейса пользователя для работы с документом — к созданию его экранной формы. Для этого в окне редактирования объекта конфигурации перейдем на вкладку **Формы** и приступим к созданию формы документа (рис. 1.54). Исходя из этого понят-

но, что пользоваться формой, генерируемой системой по умолчанию, в режиме 1С:Предприятие мы не будем.

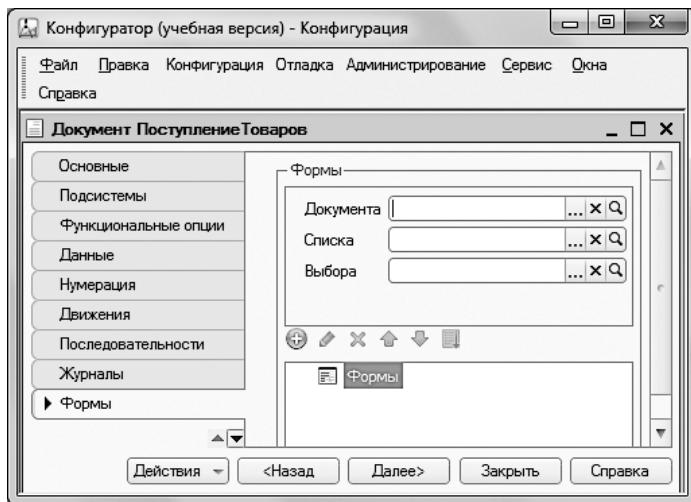


Рис. 1.54. Вкладка Формы окна редактирования документа Поступление Товаров

Создание пользовательской формы связано с тем, что далее в этом разделе нам придется добавлять в разработку собственные программные процедуры, которые будут связаны с элементами, расположенными на форме.

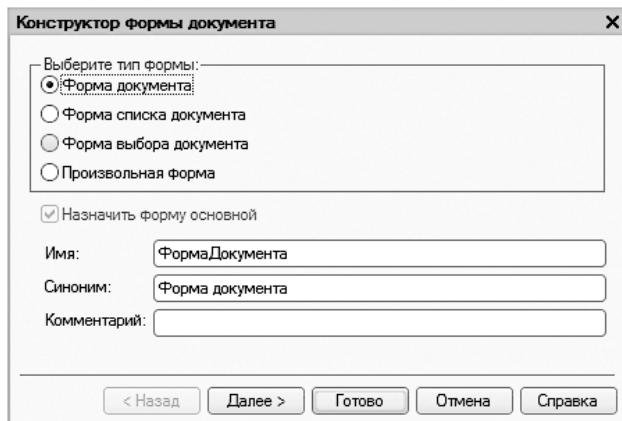


Рис. 1.55. Окно конструктора формы документа

Итак, перейдем к техническим действиям по разработке формы. Для начала требуется щелкнуть на значке лупы в поле Документа раздела Формы (см. рис. 1.54) и воспользоваться конструктором формы документа. В первом появившемся окне

диалога следует установить параметры в соответствии с рис. 1.55 (реально они должны быть установлены системой по умолчанию).

После щелчка на кнопке **Далее** мы увидим следующее окно конструктора, в котором требуется установить флажки для отображения на форме всех ранее созданных реквизитов документа (рис. 1.56).

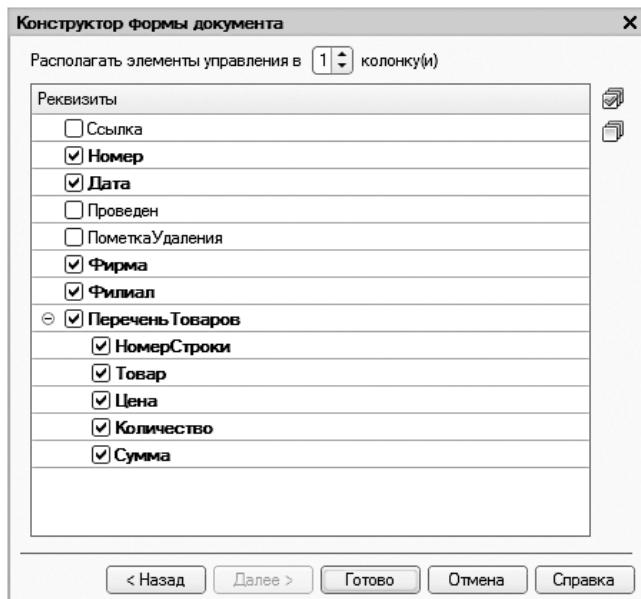


Рис. 1.56. Включение реквизитов в форму документа ПоступлениеТоваров

Работа в режиме диалога с конструктором формы документа завершается щелчком по кнопке **Готово**, что приводит к появлению на экране окна редактора форм с формой нашего документа (рис. 1.57).

Осталось на вкладке **Подсистемы** отметить флажком подсистему **Поступление**, где будут отражаться документы созданного вида.

На этом действия по разработке документа в режиме конфигуратора завершаются и далее требуется перейти в режим 1С:Предприятие, чтобы заполнить несколько документов созданного вида.

Один из вариантов заполненного документа **ПоступлениеТоваров** показан на рис. 1.58. Здесь мы установили названия фирмы-поставщика и филиала из соответствующих справочников. Также при заполнении поля **Товар** табличной части автоматически открывается форма списка справочника **Товары**. Остальные реквизиты являются числовыми и заполняются вручную с помощью клавиатуры.

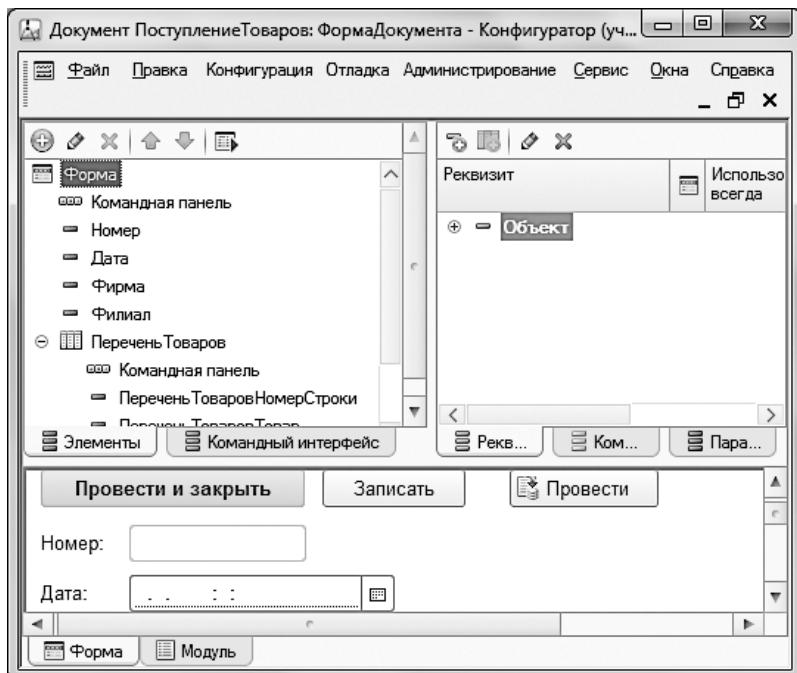


Рис. 1.57. Окно формы документа Поступление Товаров в редакторе форм

The screenshot shows the '1C:Предприятие' application window titled 'Конфигурация (1C:Предприятие, учебная версия)'. The top navigation bar includes 'Главное', 'Поступление', 'Продажа', and 'Управление'. The main area is titled 'Поступление товаров' (Goods Arrival) and 'Фирмы' (Companies). On the left is a vertical toolbar with icons for back/forward, search, and other functions. The central area contains buttons for 'Провести и закрыть' (Post and Close), 'Записать' (Save), 'Провести' (Post), and 'Еще' (More). It also includes input fields for 'Дата:' (Date) set to '20.02.2014 0:00:00' and dropdowns for 'Фирма:' (Company) set to 'Меда' and 'Филиал:' (Branch) set to 'Сормовский'. A 'Добавить' (Add) button is available. Below is a table for adding goods:

N	Товар	Цена	Количество	Сумма
1	Стенка "Светлойр"	31 000	1	31 000

At the bottom, a status bar shows 'Текущие вызовы: 0 Накопленные вызовы: 44'.

Рис. 1.58. Заполнение документа Поступление Товаров

Большим недостатком рассматриваемого документа является необходимость ручного заполнения поля **Сумма**. Так мы должны самостоятельно перемножить цену на количество и подставить результат в документ.

Несмотря на указанный недостаток, после внесения информации в поля документа его можно сохранить и закрыть (все это выполняется щелчком на кнопке **Провести и закрыть**). В дальнейшем данный документ (как и все другие, созданные таким же образом) всегда можно найти в списке, который открывается автоматически при щелчке на ссылке **Поступление товаров**, когда мы работаем с подсистемой **Поступление**.

Программирование формы документа

В связи с отмеченным недостатком (необходимостью ручного заполнения поля **Сумма** на форме документа **Поступление Товаров**) обеспечим необходимую автоматизацию работы пользователей при внесении данных в документ. В настоящий момент при заполнении строки табличной части действия пользователя достаточно рутинны, и наша цель заключается в том, чтобы обеспечить следующий функционал:

- после установки количества единиц товара должна автоматически рассчитываться сумма (как произведение цены на количество);
- после установки цены товара также должна автоматически рассчитываться сумма (как произведение цены на количество).

В этом случае при вводе в форму документа цены и количества товаров в любой последовательности будет автоматически производиться пересчет суммы по текущей строке товара в табличной части документа.

Для начала в окне редактирования формы документа необходимо двойным щелчком на пункте **Перечень Товаров Количество** открыть окно свойств этого поля (рис. 1.59).

В нижней части окна свойств располагается перечень событий, которые связаны с полем ввода **Перечень Товаров Количество** (рис. 1.60). Программируя реакцию на эти события, разработчик может влиять на действия системы в режиме 1С:Предприятие. Фактически тем самым разработчик указывает системе, что она должна делать при возникновении данных событий.

ПРИМЕЧАНИЕ

В окне редактора формы указанное нами название реквизита автоматически предваряется названием табличной части. Так, реквизит **Количество** автоматически трансформируется в название **Перечень Товаров Количество**.

В поле **При Изменении** следует щелкнуть на значке лупы, в результате на экране появится окно, которое позволяет правильно сформировать заготовку для необходимой процедуры (рис. 1.61). В данном случае следует установить переключатель **Создать на клиенте** (в дальнейшем мы разберем, зачем это нужно). После этого мы

попадем в процедуру, автоматически выполняемую при указании количества товаров в поле Количество документа ПоступлениеТоваров, когда происходит работа с документом в режиме 1С:Предприятие.

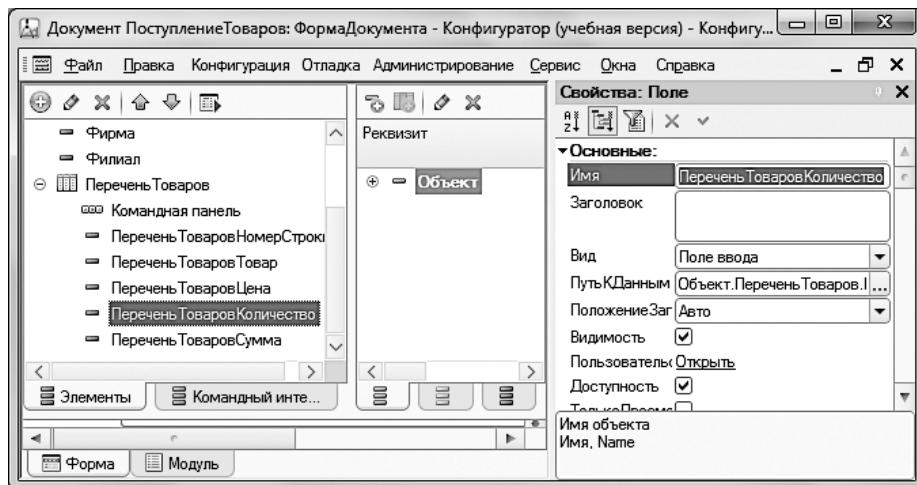


Рис. 1.59. Окно свойств поля ПереченьТоваровКоличество формы документа

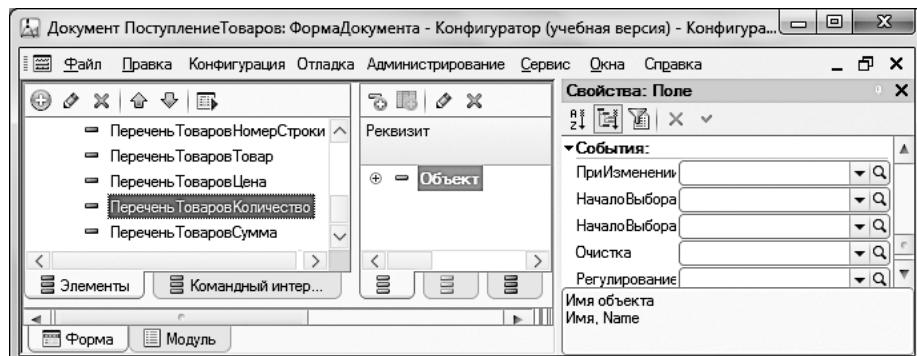


Рис. 1.60. Раздел окна свойств для обработки событий в поле ПереченьТоваровКоличество

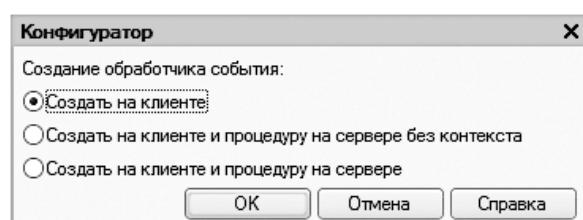


Рис. 1.61. Окно для создания обработчика событий

В результате выполненных действий активизируется редактор программного кода с процедурой, которая выполняется при вводе количества товаров в строке табличной части документа (рис. 1.62). Фактически, это еще только заготовка для будущей программы, которую нам необходимо написать.

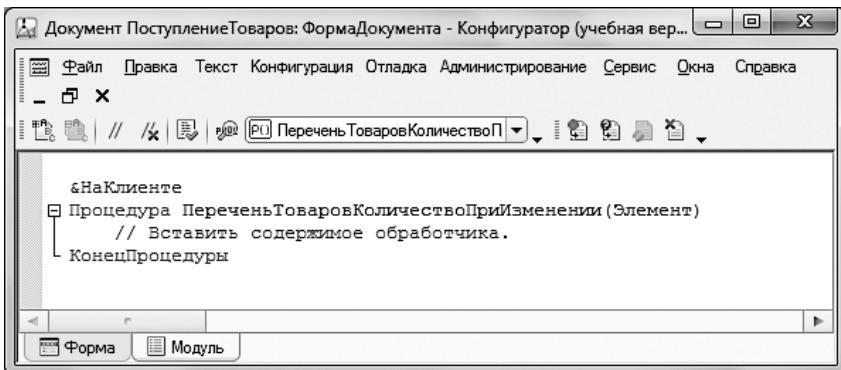


Рис. 1.62. Заготовка процедуры обработки события ПриИзменении

ПРИМЕЧАНИЕ

В дальнейшем разработанную процедуру всегда можно найти на вкладке Модуль, где размещаются программные процедуры, работающие с экранной формой документа.

Непосредственно текст процедуры, который необходимо написать для решения поставленной задачи, приведен в листинге 1.1. Модуль формы может включать набор процедур и функций. Также в модуле формы допускается размещение переменных. Каждая процедура, функция или объявление переменной модуля формы должны вначале включать одну из следующих директив компиляции:

- НаКлиенте** — выполнение на стороне клиента;
- НаСервере** — выполнение на стороне сервера;
- НаСервереБезКонтекста** — выполнение на сервере вне контекста формы;
- НаКлиентеНаСервереБезКонтекста** — выполнение как на клиенте, так и на сервере вне контекста формы.

ПРИМЕЧАНИЕ

Модуль — это контейнер для текста программ на встроенном языке. В конфигурации существует большое количество модулей, которые расположены в различных местах. Программы, содержащиеся в модуле, будут вызываться платформой в заранее определенные моменты работы системы 1С:Предприятие.

Отсутствие директивы компиляции перед процедурой означает использование директивы, предлагаемой по умолчанию, каковой является директива `НаСервере`. Подробнее об этих директивах мы поговорим далее.

Листинг 1.1. Процедура для расчета суммы при вводе количества товаров

```
&НаКлиенте
Процедура ПереченьТоваровКоличествоПриИзменении(Элемент)
    СтрTabЧасти=Элементы.ПереченьТоваров.Текущие данные;
    СтрTabЧасти.Сумма=СтрTabЧасти.Цена*СтрTabЧасти.Количество;
КонецПроцедуры
```

Ключевое слово `Процедура` (вместе с ее именем справа) определяет начало процедуры, а ключевое слово `КонецПроцедуры` — завершение ее текста. Хотя листинг 1.1 небольшой, в нем имеются важные программные конструкции. Например:

```
СтрTabЧасти=Элементы.ПереченьТоваров.Текущие данные;
```

В этой строке мы используем коллекцию `Элементы`, содержащую элементы управления, расположенные на форме. На форме документа присутствует табличная часть с именем `ПереченьТоваров`. Данный элемент включает строки табличной части документа. Получить строку, с которой в текущий момент работает пользователь, можно с помощью свойства `Текущие данные` объекта `ПереченьТоваров`.

Таким образом, в результате выполнения рассматриваемого фрагмента программного кода переменная `СтрTabЧасти` будет определять объект, включающий редактируемые в строке данные. После этого в следующей конструкции производится заполнение информацией поля `Сумма` табличной части документа:

```
СтрTabЧасти.Сумма=СтрTabЧасти.Цена*СтрTabЧасти.Количество;
```

В процессе заполнения извлекаются значения количества и цены из текущей строки табличной части документа, которые затем перемножаются.

В результате пользователь в режиме 1С:Предприятие после последовательного внесения в табличную часть документа цены товара и необходимого количества получает автоматическое заполнение поля `Сумма` (рис. 1.63).

Продолжим работу по усовершенствованию документа, наша следующая цель — обеспечить аналогичное автоматическое вычисление суммы после внесения цены товара. Для этого в окне редактора формы документа необходимо двойным щелчком на пункте `ПереченьТоваровЦена` открыть окно свойств данного поля. Далее в этом окне следует перейти к разделу `События`, где для события `ПриИзменении` щелкнуть на значке лупы. В результате проделанных действий откроется окно программного кода процедуры, которая автоматически выполняется при внесении (изменении) цены в табличной части документа. Непосредственно сам текст процедуры, которую необходимо написать для решения поставленной задачи, приведен в листинге 1.2. Данный текст аналогичен предыдущей процедуре (см. листинг 1.1) и не требует комментария.

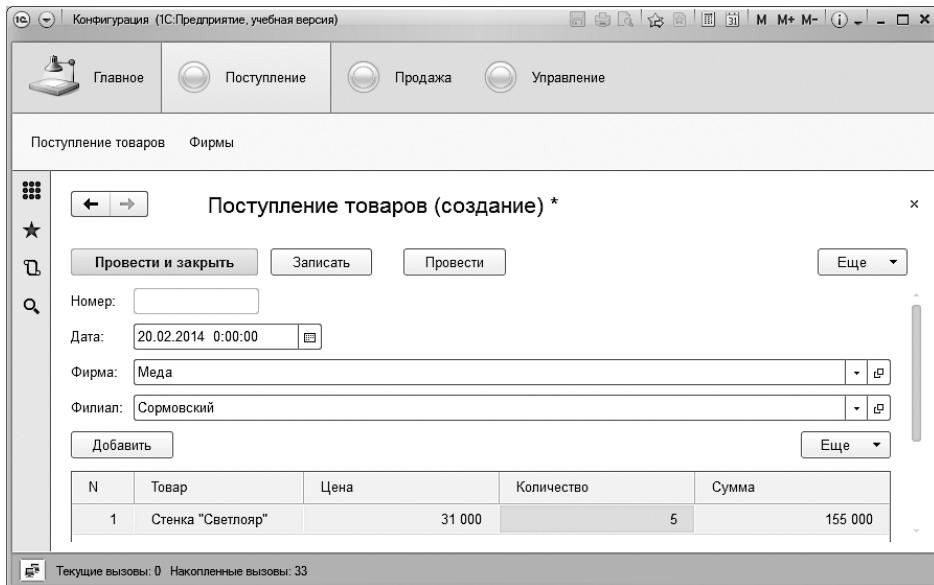


Рис. 1.63. Автоматическое заполнение поля Сумма при вводе количества товаров

Листинг 1.2. Процедура расчета суммы при вводе цены товаров

```
&НаКлиенте
Процедура ПереченьТоваровЦенаПриИзменении(Элемент)
    СтрТабЧасти=Элементы.ПереченьТоваров.ТекущиеДанные;
    СтрТабЧасти.Сумма=СтрТабЧасти.Цена*СтрТабЧасти.Количество;
КонецПроцедуры
```

После проделанных действий следует перейти в режим 1С:Предприятие, создать новый документ ПоступлениеТоваров и заполнить в нем табличную часть. Как вы можете убедиться, теперь при вводе цены и количества товаров в любой последовательности сумма рассчитывается автоматически.

В данном документе осталась недоработка, связанная с установкой цены товара, а именно при выборе товара цена должна заполняться автоматически. Эту недоработку мы устраним в последующих главах.

Печатная форма документа

Практически всегда кроме электронных форм документов пользователям в процессе работы требуются и их печатные варианты. В этом разделе мы узнаем, каким образом можно автоматизировать процесс создания печатных форм для объектов системы 1С:Предприятие 8.3. Приводимый далее пример связан с документом ПоступлениеТоваров.

Для начала откроем окно редактирования этого объекта конфигурации. Далее следует перейти на вкладку Макеты, где в нижней части окна присутствует кнопка Конструкторы (рис. 1.64). Щелкнув на ней, воспользуемся еще одним инструментом разработки, который называется *конструктором печати*.

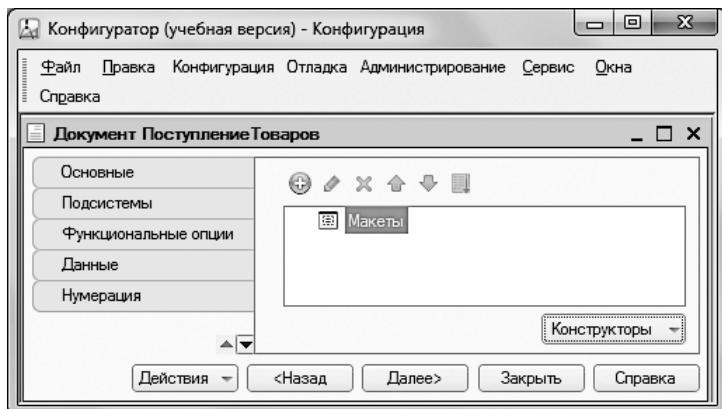


Рис. 1.64. Вкладка Макеты окна редактирования объекта конфигурации

Сам по себе диалог с конструктором печати достаточно несложный, и в первом открывшемся окне от нас потребуется только согласиться с предложением создать новую команду с именем Печать (рис. 1.65), после чего следует щелкнуть на кнопке Далее.

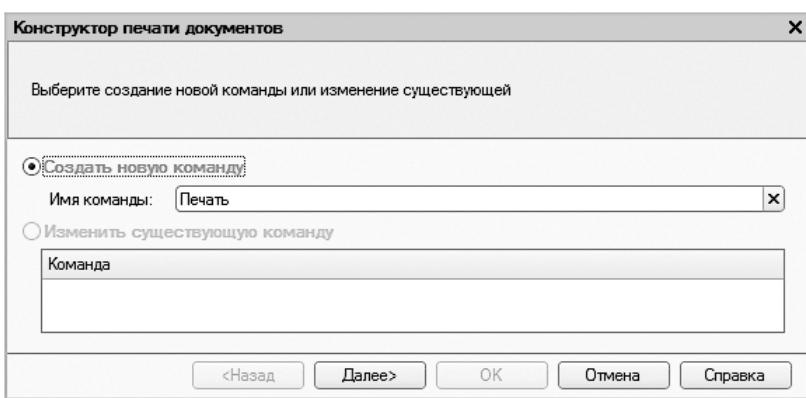


Рис. 1.65. Окно диалога с конструктором печати

Следующее окно позволяет определить, какие реквизиты шапки мы собираемся расположить на печатной форме (рис. 1.66).

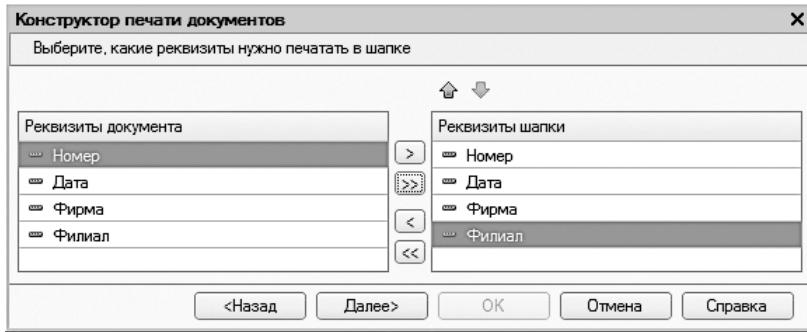


Рис. 1.66. Выбор реквизитов шапки документа в окне конструктора печати

После этого в новом окне следует задать содержание табличной части печатной формы (рис. 1.67). Включим в нее все предлагаемые поля, кроме поля НомерСтроки.

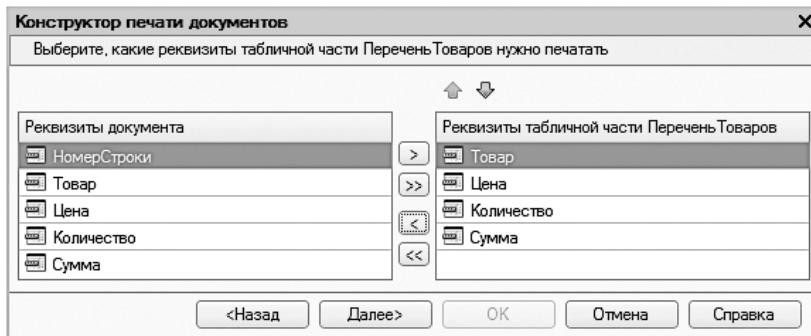


Рис. 1.67. Выбор реквизитов табличной части документа в окне конструктора печати

Заполнение нижней части документа (подвала) в рассматриваемом примере не понадобится (рис. 1.68).

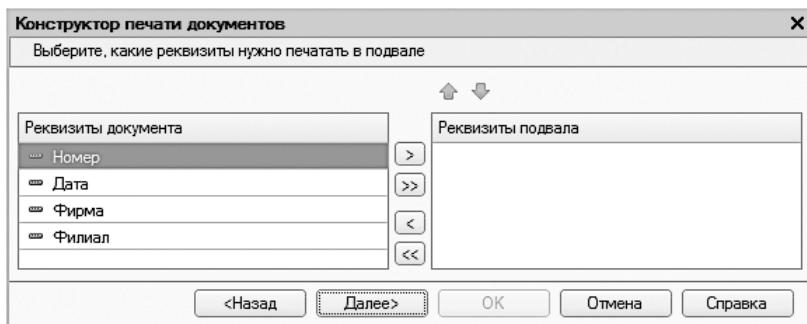


Рис. 1.68. Оформление подвала документа в окне конструктора печати

Теперь после щелчка на кнопке Далее открывается завершающее окно конструктора печати, позволяющее указать группу, куда будет входить команда формирования печатной формы (рис. 1.69).

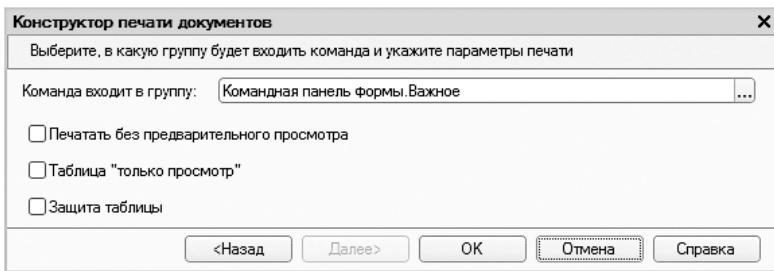


Рис. 1.69. Определение месторасположения команды печати на форме документа

В завершении работы конструктора мы увидим на экране макет печатной формы и набор программных процедур (рис. 1.70).

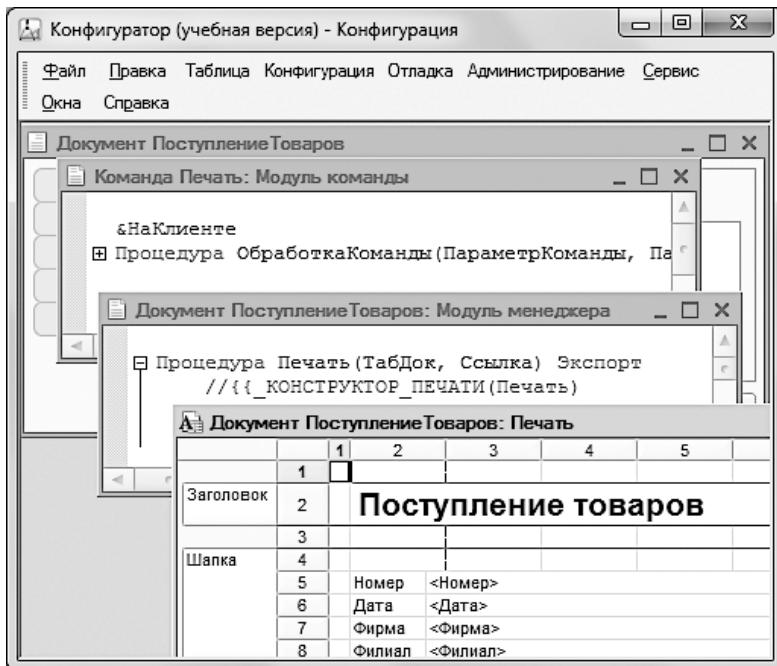


Рис. 1.70. Результат работы конструктора печати

Посмотрим теперь, как выглядит результат работы конструктора печати в режиме 1С:Предприятие. Для этого воспользуемся уже знакомым меню Отладка. После об-

ращения к форме списка документа Поступление Товаров видно, что на командной панели появилась новая кнопка Печать, щелчком на которой формируется печатная форма выделенного документа (рис. 1.71). Таким образом, действия, выполняемые в диалоге с конструктором печати, достаточно просты, а итог, как мы видим, весьма эффективен.

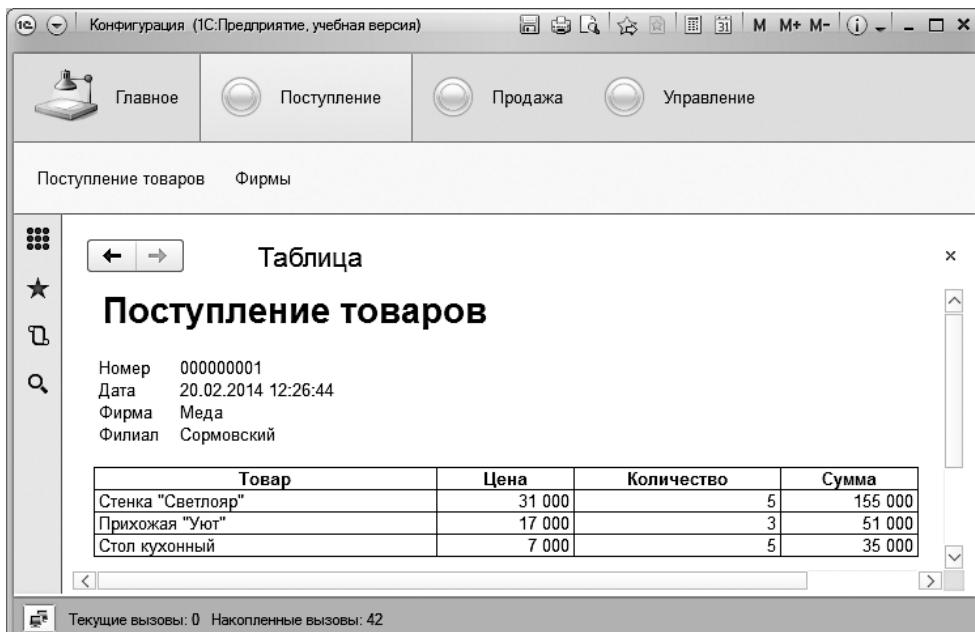


Рис. 1.71. Печатная форма документа

Вернемся теперь немного назад и обсудим результаты работы конструктора печати. На рис. 1.72 показан *макет печатной формы*, представляющий собой заготовку (*шаблон*) печатного документа, который будет формироваться системой при работе в режиме 1С:Предприятие. Макет печатной формы доступен для разработчика — его всегда можно открыть и посмотреть (а при необходимости и изменить) на вкладке *Макеты* в окне редактирования объекта конфигурации. В нашем случае на данной вкладке мы увидим значок нового объекта конфигурации — макета Печать.

Другим результатом работы конструктора печати является набор программных процедур (листинги 1.3 и 1.4), которые выполняются при работе в режиме 1С:Предприятие и взаимодействуют с макетом. А уже результатом этого взаимодействия является табличный документ, отражающий информацию из документа Поступление Товаров (см. рис. 1.71).

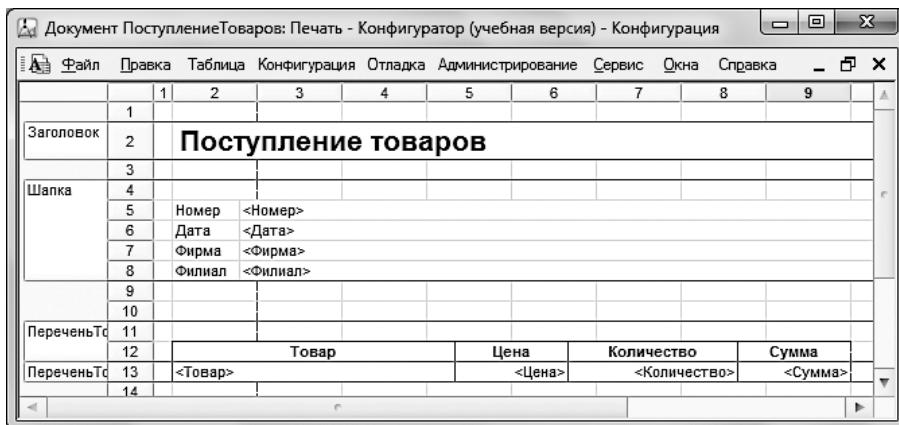


Рис. 1.72. Макет печатной формы

ПРИМЕЧАНИЕ

Для проектирования макета используется редактор табличного документа, входящий в систему 1С:Предприятие. В процессе создания макета в ячейки вводится текст (как в результате работы конструктора, так и непосредственно пользователем). При этом можно задавать различные параметры форматирования для ячеек. Печатные формы, как правило, содержат несколько типичных областей — заголовочную область, таблицу результатов, место для итоговых показателей и др. Это связано с тем, что информация, выводимая в отчеты, обычно неоднородная — есть табличные данные, заголовок, область размещения подписей. Макет представляет собой совокупность прямоугольных областей, каждая из которых служит для формирования какой-то части печатной формы. Заметим, что табличный документ (это более правильное название, чем «печатная форма», но условно их можно считать синонимами) и макет — это различные объекты. Макет представляет собой некий шаблон табличного документа в плане того, как необходимо создавать печатную форму. Фактически, в режиме конфигуратора мы определяем вид печатной формы, которую собираемся затем увидеть при работе в режиме 1С:Предприятие.

Листинг 1.3. Процедуры, формируемые конструктором печати в модуле команды &НаКлиенте

```
Процедура ОбработкаКоманды(ПараметрКоманды, ПараметрыВыполненияКоманды)
//{{_КОНСТРУКТОР_ПЕЧАТИ}(Печать)
ТабДок = Новый ТабличныйДокумент;
Печать(ТабДок, ПараметрКоманды);
ТабДок.ОтображатьСетку = Ложь;
ТабДок.Защита = Ложь;
ТабДок.ТолькоПросмотр = Ложь;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.Показать();
//}}
КонецПроцедуры
```

продолжение ⇨

Листинг 1.3 (продолжение)

```
&НаСервере
Процедура Печать(ТабДок, ПараметрКоманды)
    Документы.ПоступлениеТоваров.Печать(ТабДок, ПараметрКоманды);
КонецПроцедуры
```

Листинг 1.4. Процедура, формируемая конструктором печати в модуле менеджера

```
Процедура Печать(ТабДок, Ссылка) Экспорт
//{{_КОНСТРУКТОР_ПЕЧАТИ(Печать)}
Макет = Документы.ПоступлениеТоваров.ПолучитьМакет("Печать");
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
| ПоступлениеТоваров.Дата,
| ПоступлениеТоваров.Номер,
| ПоступлениеТоваров.Филиал,
| ПоступлениеТоваров.Фирма,
| ПоступлениеТоваров.ПереченьТоваров.(
|     Товар,
|     Цена,
|     Количество,
|     Сумма
| )
|ИЗ
| Документ.ПоступлениеТоваров КАК ПоступлениеТоваров
|ГДЕ
| ПоступлениеТоваров.Ссылка В (&Ссылка");
Запрос.Параметры.Вставить("Ссылка", Ссылка);
Выборка = Запрос.Выполнить().Выбрать();
ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");
Шапка = Макет.ПолучитьОбласть("Шапка");
ОбластьПереченьТоваровШапка =
    Макет.ПолучитьОбласть("ПереченьТоваровШапка");
ОбластьПереченьТоваров = Макет.ПолучитьОбласть("ПереченьТоваров");
ТабДок.Очистить();
ВставлятьРазделительСтраниц = Ложь;
Пока Выборка.Следующий() Цикл
    Если ВставлятьРазделительСтраниц Тогда
        ТабДок.ВывестиГоризонтальныйРазделительСтраниц();
    КонецЕсли;
    ТабДок.Вывести(ОбластьЗаголовок);
    Шапка.Параметры.Заполнить(Выборка);
    ТабДок.Вывести(Шапка, Выборка.Уровень());
    ТабДок.Вывести(ОбластьПереченьТоваровШапка);
    ВыборкаПереченьТоваров = Выборка.ПереченьТоваров.Выбрать();
    Пока ВыборкаПереченьТоваров.Следующий() Цикл
        ОбластьПереченьТоваров.Параметры.Заполнить(ВыборкаПереченьТоваров);
        ТабДок.Вывести(
            ОбластьПереченьТоваров, ВыборкаПереченьТоваров.Уровень());
    КонецЦикла;
    ВставлятьРазделительСтраниц = Истина;
КонецЦикла;
//}}
КонецПроцедуры
```

При рассмотрении листингов 1.3 и 1.4 бросается в глаза то, что несколько строк начинаются с двух наклонных черт. Это комментарий, который программно не выполняется, а предназначен лишь для пояснения разработки. В следующей строке создается новый объект типа **ТабличныйДокумент**, а переменная **ТабДок** будет с ним ассоциироваться:

```
ТабДок = Новый ТабличныйДокумент;
```

Далее в программе переменная **ТабДок** позволяет работать со свойствами и методами созданного объекта.

ПРИМЕЧАНИЕ

Оператор **Новый** создает значение указанного типа. После имени типа в скобках могут указываться параметры.

Упрощенно объект типа **ТабличныйДокумент** представляет собой таблицу, но с разнообразными свойствами и методами работы с ней. В следующей строке используется метод **ПолучитьМакет()**, в качестве входного параметра которого указывается имя макета печатной формы:

```
Макет = Документы.ПоступлениеТоваров.ПолучитьМакет("Печать");
```

Это имя создано конструктором печати. Далее с помощью переменной **Макет** производится формирование табличного документа на основании нашего макета.

ПРИМЕЧАНИЕ

Более точно табличный документ является объектом, который предназначен в разработках для формирования печатных форм основных объектов (справочников, документов, регистров и др.), а также для форм, в которых информация представляется в табличном виде. Используя свойства и методы этого объекта, можно формировать необходимый для пользователя вид таблицы в режиме 1С:Предприятие.

Автоматически созданный конструктором макет с именем **Печать** имеет несколько областей, одна из которых, содержащая заголовочную информацию, так и называется — **Заголовок** (см. рис. 1.72). В следующей строке программного кода в табличный документ выводится первая секция в соответствие с макетом:

```
ТабДок.Вывести(ОбластьЗаголовок);
```

Далее аналогичным образом производится заполнение последующих секций. Таким способом происходит *заполнение табличного документа на основании макета*. Содержание областей макета и их названий, сформированное по умолчанию, мы в принципе можем изменить, однако, как правило, в этом нет особой необходимости.

Учитывая вид макета (см. рис. 1.72), можно сказать, что основное содержание печатной формы документа представляет собой совокупность одинаковых по структуре строк, подписи для которых располагаются в секции **ПереченьТоваровШапка**. Метод **Выбрать()**, применяемый к результату выполнения запроса, касающегося документа **ПоступлениеТоваров**, позволяет подготовить дальнейшее извлечение информации.

ПРИМЕЧАНИЕ

Запросы служат для извлечения информации из базы данных. О них мы подробно поговорим во второй главе.

Далее производится последовательное извлечение всех элементов табличной части документа с помощью цикла:

Пока ВыборкаПереченьТоваров.Следующий() Цикл

На каждой итерации цикла мы получаем информацию, содержащуюся в очередной строке табличной части документа. Данный цикл перебора строк документа производит заполнение табличного документа информацией на основании области макета **ПереченьТоваров**. Для этого используется метод **Вывести()**. В результате выполнения рассмотренного цикла в табличном документе создается столько строк, сколько имеется в табличной части документа.

Часть программного кода в листинге 1.3 связана с настройкой табличного документа. Например:

ТабДок.ОтображатьСетку = Ложь;

В этой конструкции указано, что не надо выводить сетку. В качестве альтернативы для свойства табличного документа **ОтображатьСетку** можно указать значение **Истина**. Тогда для всего табличного документа (в режиме 1С:Предприятие) будет установлена сетка.

Другое свойство **Защита** позволяет при установке значения **Истина** защитить табличный документ от редактирования и копирования. В нашем примере мы этого не сделали:

ТабДок.Защита = Ложь.

Однако если для данного свойства установить значение **Истина**, то в режиме 1С:Предприятие скопировать табличный документ в буфер обмена не удастся.

Свойство **ТолькоПросмотр** позволяет обеспечить два варианта работы с табличным документом:

- только просмотр информации (при значении **Истина**) — в этом случае будут недоступны для редактирования ячейки и рисунки, у которых установлено

- свойство **Защита** (это делается в окне свойств ячейки, которое открывается с помощью контекстного меню);
- возможность изменения информации (при значении **Ложь**) в табличном документе при работе в режиме 1С:Предприятие.

Свойство **ОтображатьЗаголовки** содержит признак отображения заголовков столбцов и строк таблицы. В случае если для данного свойства установлено значение **Истина**, то в табличном документе будут выведены номера столбцов и строк соответственно в верхней и левой частях таблицы.

В заключение обратим внимание на строку первой процедуры из листинга 1.3:

```
ТабДок.Показать();
```

В ней используется метод **Показать()**, предназначенный для отображения табличного документа на экране.

Теперь нам осталось лишь перейти в режим 1С:Предприятие, открыть форму списка документа **ПоступлениеТоваров** и щелкнуть на кнопке **Печать**. На рис. 1.71 мы уже видели заполненную печатную форму одного из созданных документов.

Теперь при желании печатную форму можно скорректировать (изменить информацию, содержащуюся в ячейках), а также сохранить как рабочую книгу Microsoft Excel (для этого следует в меню **Файл** выбрать пункт **Сохранить как** и далее указать вариант рабочей книги Microsoft Excel).

Учет продажи товаров

Для фиксации факта продажи товаров необходимо разработать соответствующий документ, который так и назовем **ПродажаТоваров**.

В окне редактирования объекта конфигурации перейдем на закладку **Данные** и внесем три реквизита:

- Фирма** (тип данных — **СправочникСсылка.Фирмы**);
- Филиал** (тип данных — **СправочникСсылка.Филиалы**);
- Менеджер** (тип данных — **СправочникСсылка.Менеджеры**).

Аналогично документу **ПоступлениеТоваров** разместим в разрабатываемом документе **ПродажаТоваров** табличную часть под названием **ПереченьТоваров**, где расположим следующие реквизиты:

- Товар** (тип данных — **СправочникСсылка.Товары**);
- Цена** — цена продажи единицы указанного товара (тип данных — число);

- Количество — количество проданных единиц товара (тип данных — число);
- Сумма — сумма по данному товару (тип данных — число).

В результате вкладка **Данные** должна выглядеть так, как показано на рис. 1.73.

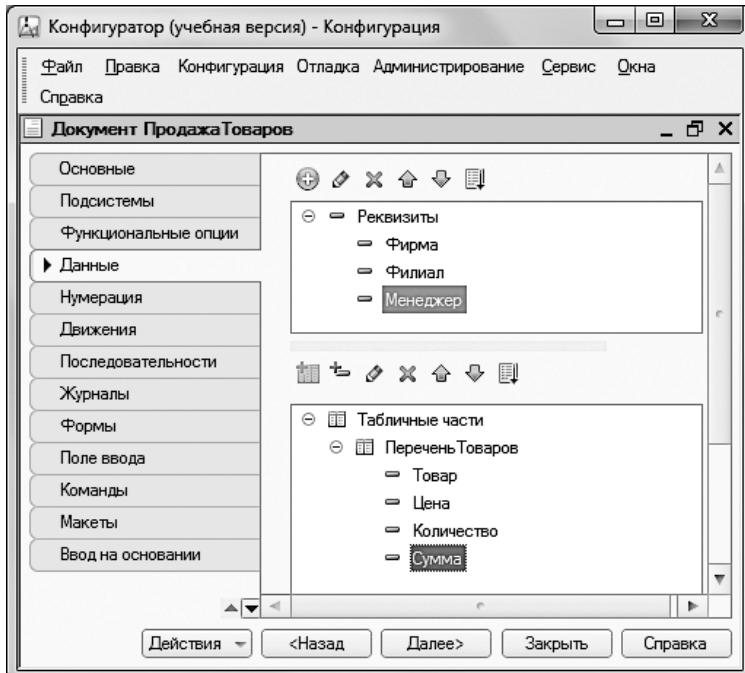


Рис. 1.73. Вкладка Данные окна редактирования документа ПродажаТоваров

В разрабатываемом документе мы должны обеспечить автоматическое вычисление суммы по текущей позиции товара после внесения цены и количества единиц проданного товара. В связи с этим нам понадобится пользовательская форма документа. Технические действия для ее создания подробно рассмотрены при разработке документа **Поступление Товаров**.

Обеспечим теперь автоматизацию расчета суммы в созданном документе, а именно после установки пользователем количества товаров или цены должна программно рассчитываться сумма по текущей позиции товара.

Для этого на вкладке **Модуль формы** документа **ПродажаТоваров** необходимо разместить процедуры, аналогичные тем, которые были созданы в документе **Поступление Товаров**. Сначала откроем окно свойств поля **Перечень Товаров Количество** и обратимся к событию **ПриИзменении**. Здесь после щелчка на значке лупы открывается окно для написания программного кода процедуры, которая автоматически выполняется при внесении

количества единиц товара в табличную часть документа. Процедуру ПереченьТоваровКоличествоПриИзменении необходимо оформить в соответствии с листингом 1.5.

Листинг 1.5. Процедура расчета суммы при вводе количества товаров

&НаКлиенте

Процедура ПереченьТоваровКоличествоПриИзменении(Элемент)

СтрТабЧасти=Элементы.ПереченьТоваров.Текущие данные;

СтрТабЧасти.Сумма=СтрТабЧасти.Цена*СтрТабЧасти.Количество;

КонецПроцедуры

Аналогичным образом следует оформить процедуру ПереченьТоваровЦенаПриИзменении, выполняемую при вводе цены в табличную часть документа (листинг 1.6).

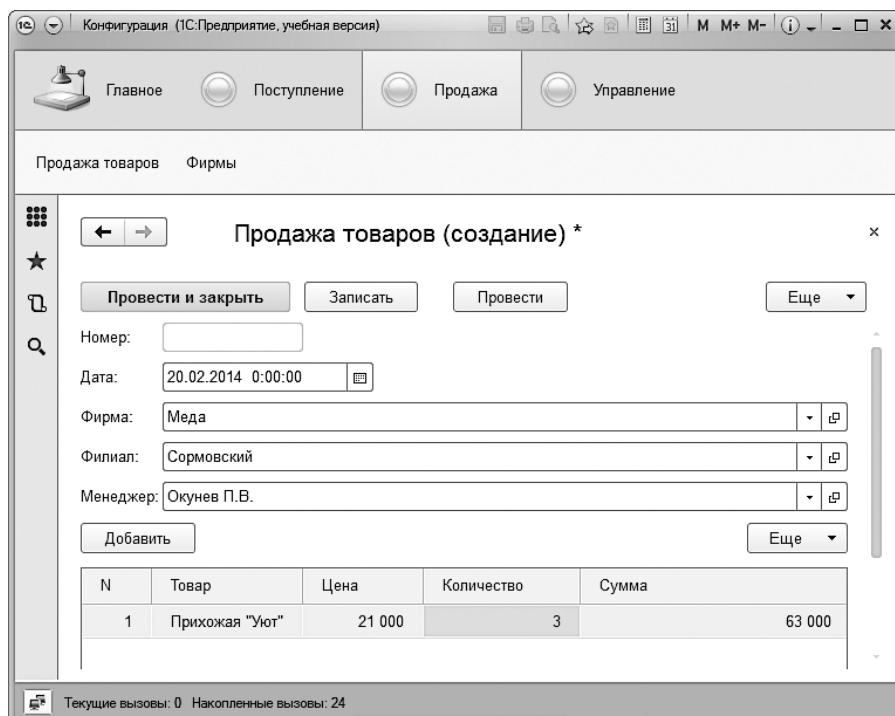


Рис. 1.74. Форма документа ПродажаТоваров в режиме 1С:Предприятие

Листинг 1.6. Процедура расчета суммы при вводе цены товара

&НаКлиенте

Процедура ПереченьТоваровЦенаПриИзменении(Элемент)

СтрТабЧасти=Элементы.ПереченьТоваров.Текущие данные;

СтрТабЧасти.Сумма=СтрТабЧасти.Цена*СтрТабЧасти.Количество;

КонецПроцедуры

Отнесем разработанный документ ПродажаТоваров к подсистеме Продажа и перейдем в режим 1С:Предприятие. На рис. 1.74 показан результат выполнения разработанных процедур в режиме 1С:Предприятие — в табличной части документа подсчитывается сумма как результат произведения цены на количество единиц товара.

Регистры накопления

В задачах учета практически ни одно прикладное решение не обходится без использования регистров. В информационной структуре, создаваемой системой на основании объекта конфигурации *регистр накопления*, производится автоматическое накопление данных (поступление и отгрузка товаров, денежные перечисления и т. д.).

Основным назначением регистра накопления является аккумулирование числовой информации в разрезе одного или нескольких *измерений*. Эти измерения описываются разработчиком в режиме конфигуратора. Например, измерениями могут быть названия товаров, склады, ответственные лица и т. д.

ПРИМЕЧАНИЕ

Измерения — это также объекты конфигурации, но они относятся к категории подчиненных.

Виды числовой информации, накапливаемой регистром, называются *ресурсами*. Они также являются подчиненными объектами и описываются в режиме конфигуратора. В нашем случае регистр накопления должен обеспечить накопление информации о количестве наших товаров как результат их поступления и продажи.

Изменение состояния регистра накопления происходит, как правило, при проведении документа. В этом случае в регистр добавляется одна или несколько записей. Каждая запись содержит значения измерений, значения приращений ресурсов, а также ссылку на документ, который вызвал эти изменения. Такая ссылка на документ называется *регистратором*, а набор записей регистра накопления, формируемых в результате проведения документа, называется *движениями регистра накопления*.

ВНИМАНИЕ

Каждому движению регистра накопления всегда должен соответствовать определенный регистратор.

Кроме того, регистр накопления может хранить дополнительную информацию, описывающую каждое движение. Набор такой дополнительной информации задается разработчиком при помощи *реквизитов* регистра накопления.

Регистр накопления КоличествоТоваров

Продолжим работу в режиме конфигуратора и создадим новый объект конфигурации — регистр накопления **КоличествоТоваров**. Для этого следует в окне Конфигурация щелчком правой кнопки мыши на значке Регистры накопления вызвать контекстное меню. Далее после выбора пункта Добавить перед нами откроется окно редактирования объекта конфигурации (рис. 1.75). Здесь на вкладке Основные необходимо дать название создаваемому регистру и указать, что он будет являться *регистром остатков* (далее мы обсудим, что это означает).

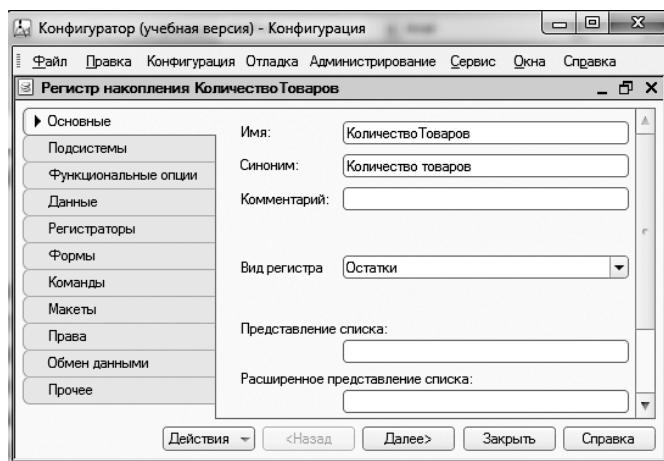


Рис. 1.75. Вкладка Основные окна редактирования регистра КоличествоТоваров

После таких начальных установок перейдем на вкладку **Данные**, где создадим два измерения регистра накопления **КоличествоТоваров** (рис. 1.76):

- Товар (тип данных — СправочникСсылка.Товары);
- Филиал (тип данных — СправочникСсылка.Филиалы).

Теперь сформируем в регистре один ресурс — **Количество** (тип данных — число).

На этом все необходимые действия по разработке нового объекта конфигурации завершаются, однако использовать созданный регистр накопления без документа, который бы *совершал движения по данному регистру*, невозможно. Более того, если вы сейчас попробуете перейти в режим 1С:Предприятие, то система напомнит о незавершенности выполненных действий.

ПРИМЕЧАНИЕ

Движения документа — это записи в регистрах, которые совершаются в процессе проведения документа.

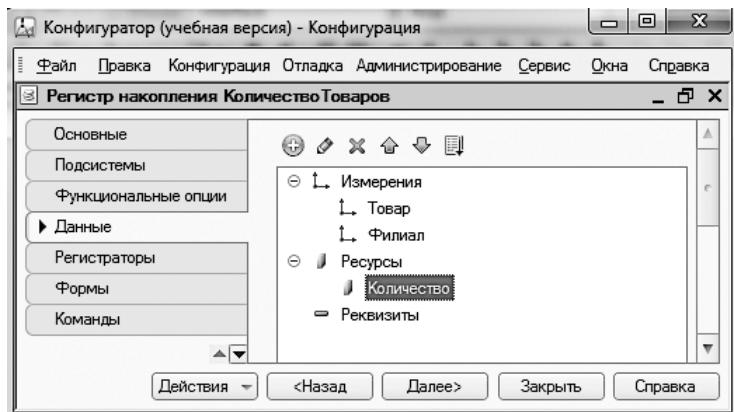


Рис. 1.76. Измерения и ресурс регистрa Количество Товаров

Итак, в окне редактирования объекта конфигурации вернемся к документу Поступление Товаров и откроем вкладку Движения (рис. 1.77).

Здесь в списке регистров конфигурации необходимо отметить флаjkом единственный регистр, который мы на данный момент создали. После установки флаjkка сразу становится доступна кнопка Конструктор движенияй, которым мы и воспользуемся. Конструктор движенияй устроен достаточно просто — в списке Регистры перечислены регистры, по которым документ может создавать движения (рис. 1.78). В данном случае это единственный регистр — Количество Товаров.

В правое верхнее поле окна конструктора автоматически внесены реквизиты шапки рассматриваемого документа. Это связано с тем, что в указанное поле включаются параметры документа, которые могут участвовать в движенияях. Однако кроме шапки в документе имеется еще и табличная часть. Для добавления реквизитов табличной части в поле Реквизиты документа следует выполнить несложное техническое действие. В центральной части рассматриваемого окна (см. рис. 1.78) присутствует раскрывающийся список Табличная часть, где перечислены табличные части, имеющиеся в документе (в нашем документе одна табличная часть — Перечень Товаров). Для перенесения в поле Реквизиты документа реквизитов табличной части ее имя необходимо просто выбрать в списке табличных частей. В результате в поле Реквизиты документа конструктора будут внесены дополнительные строки.

Рассмотрим теперь нижнюю левую часть окна конструктора движений. Это ключевой фрагмент окна, где необходимо указать соответствие полей регистра накопления и реквизитов документа. Технически это можно выполнить либо щелчком на кнопке Заполнить выражения, либо двойным щелчком на необходимом реквизите документа в правом верхнем разделе окна на рис. 1.78. В результате напротив названий полей регистра формируются соответствующие выражения (рис. 1.79).

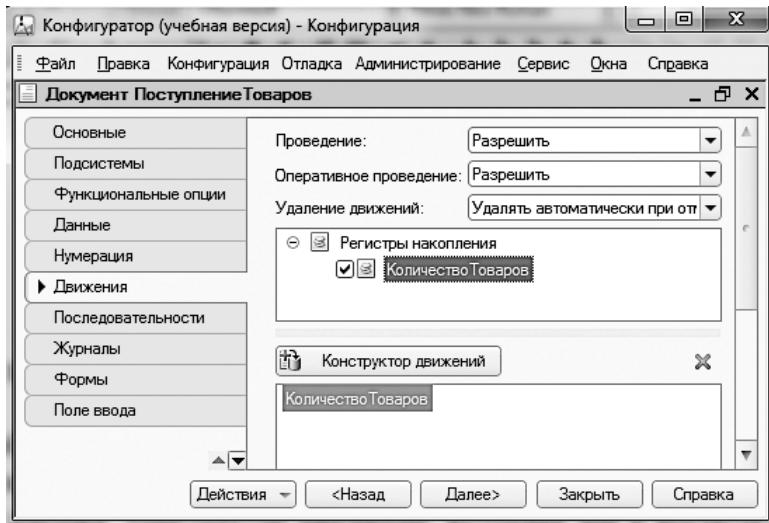


Рис. 1.77. Вкладка Движения окна редактирования документа Поступление Товаров

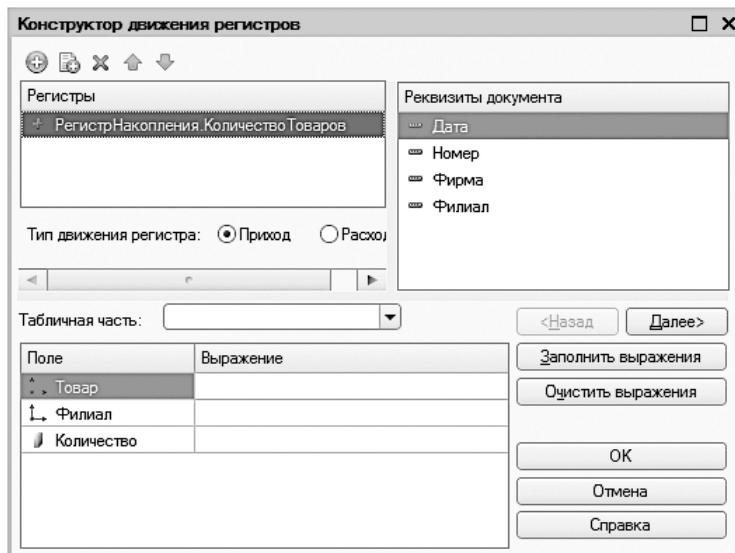


Рис. 1.78. Окно конструктора движений регистров

Теперь все действия в окне конструктора выполнены, осталось щелкнуть на кнопке OK. Это приводит к автоматической генерации программной процедуры, которая формирует движения по регистру при проведении документа Поступление Товаров (см. листинг 1.6). В дальнейшем если потребуется эту процедуру найти (скажем, для коррекции или просто просмотра), то в окне редактирования объекта конфигурации

Поступление Товаров следует перейти на вкладку Прочее. Здесь после щелчка на кнопке Модуль объекта перед нами открывается текст, представленный в листинге 1.7.

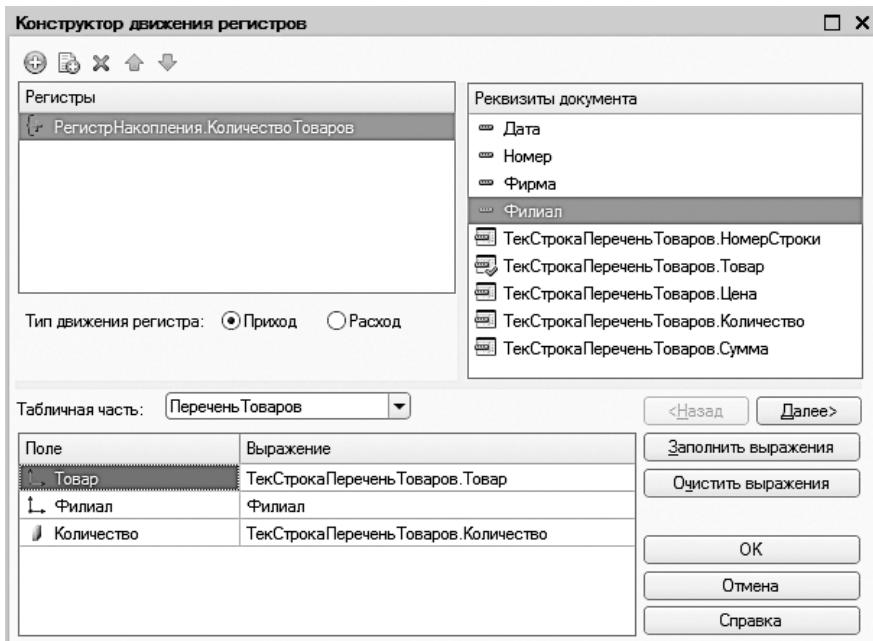


Рис. 1.79. Заполнение выражений в окне конструктора движений регистра

Листинг 1.7. Процедура проведения документа Поступление Товаров

```
Процедура ОбработкаПроведения(Отказ, Режим)
//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
Движения.КоличествоТоваров.Записывать = Истина;
Для Каждого ТекСтрокаПереченьТоваров Из ПереченьТоваров Цикл
// регистр КоличествоТоваров Приход
Движение = Движения.КоличествоТоваров.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
Движение.Период = Дата;
Движение.Товар = ТекСтрокаПереченьТоваров.Товар;
Движение.Филиал = Филиал;
Движение.Количество = ТекСтрокаПереченьТоваров.Количество;
КонецЦикла;
//}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

Вернемся к рис. 1.79 и обратим внимание на одну важную деталь в окне конструктора движений регистров. Так, в качестве значения параметра Тип движения регистра по умолчанию с помощью переключателя установлен вариант Приход, что соответствует

добавлению ресурсов регистра при проведении документа (это наиболее вероятный вариант, поэтому он и установлен по умолчанию). Альтернативный вариант **Расход** связан с уменьшением ресурсов (далее в этой главе мы разберем ситуацию, когда подобное движение по регистру реализует другой документ).

Основное содержание процедуры **ОбработкаПроведения()** составляет цикл, в котором перебираются строки табличной части **ПереченьТоваров** рассматриваемого документа **ПоступлениеТоваров**:

```
Для Каждого ТекСтрокаПереченьТоваров Из ПереченьТоваров Цикл
```

При каждом проходе цикла с помощью метода **Добавить()** создается очередная запись в регистре. В последующих строках процедуры устанавливаются значения полей регистра в соответствии с данными из очередной строки табличной части документа.

Подчеркнем главный результат работы конструктора движений. Так, вследствие несложных визуальных действий мы реализовали в документе **ПоступлениеТоваров** формирование движений по регистру накопления **КоличествоТоваров**. В дальнейшем автоматически созданную программную процедуру **ОбработкаПроведения()** можно при необходимости скорректировать либо дополнить фрагментами программного кода.

Включим регистр накопления **КоличествоТоваров** в подсистему **Управление**.

Теперь после проделанной работы перейдем в режим 1С:Предприятие и проверим результат выполненных изменений. Выберем пункт **Поступление товаров** в меню **Документы**, создадим новый документ и заполним его информацией (например, как показано на рис. 1.80).

После внесения данных в форму документа щелчок на кнопке **Провести и закрыть** приводит к записи созданного документа в информационную базу. Кроме того, осуществляется его проведение (формируются движения по регистру). На рис. 1.81 показано содержание регистра накопления **КоличествоТоваров** с внесенными записями, связанными с данным документом поступления.

Таким образом, мы рассмотрели регистрацию поступлений товаров, однако у нас имеется еще и документ **ПродажаТоваров**. Проведение этого документа должно приводить к уменьшению ресурса регистра **КоличествоТоваров** — уменьшению количества товаров. Для того чтобы это технически реализовать, необходимо в окне редактирования объекта конфигурации **ПродажаТоваров** перейти на вкладку **Движения**. Дальнейшие действия аналогичны рассмотренной ранее доработке документа **ПоступлениеТоваров**. Новых технических моментов при доработке документа **ПродажаТоваров** нет, за исключением того, что в качестве типа движения регистра следует установить вариант **Расход**.

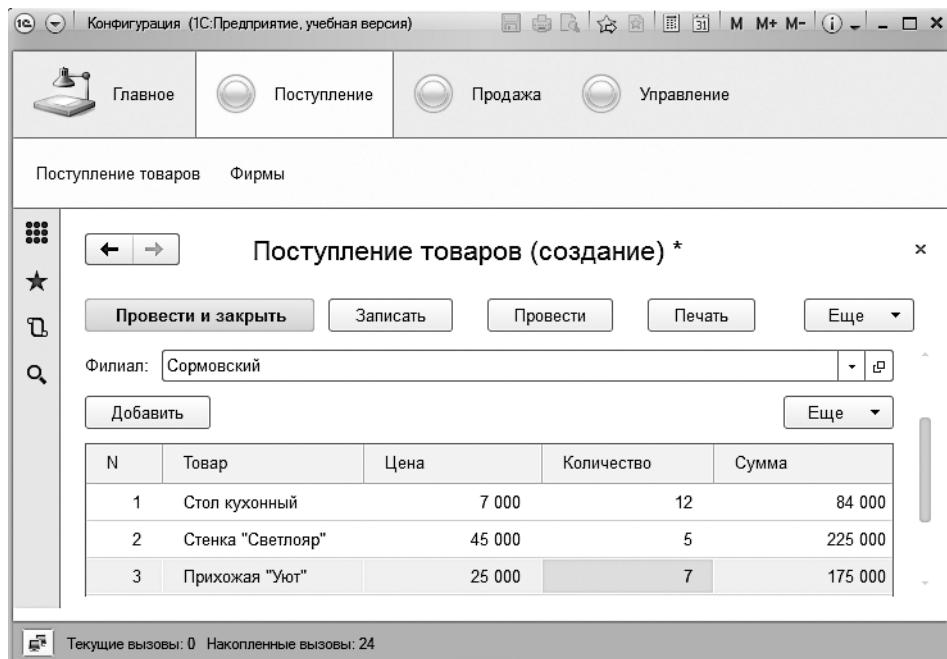


Рис. 1.80. Форма документа Поступление Товаров в режиме 1С:Предприятие

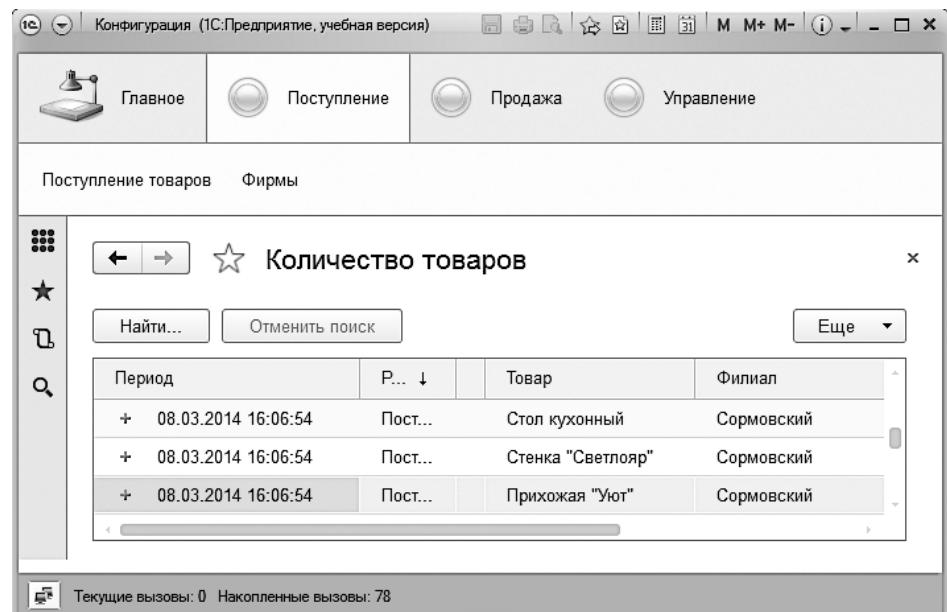


Рис. 1.81. Формирование записей в регистре Количество Товаров

После выполнения необходимых действий в окне конструктора движений автоматически генерируется программная процедура, которая формирует движения по регистру КоличествоТоваров при проведении документа ПродажаТоваров (листинг 1.8).

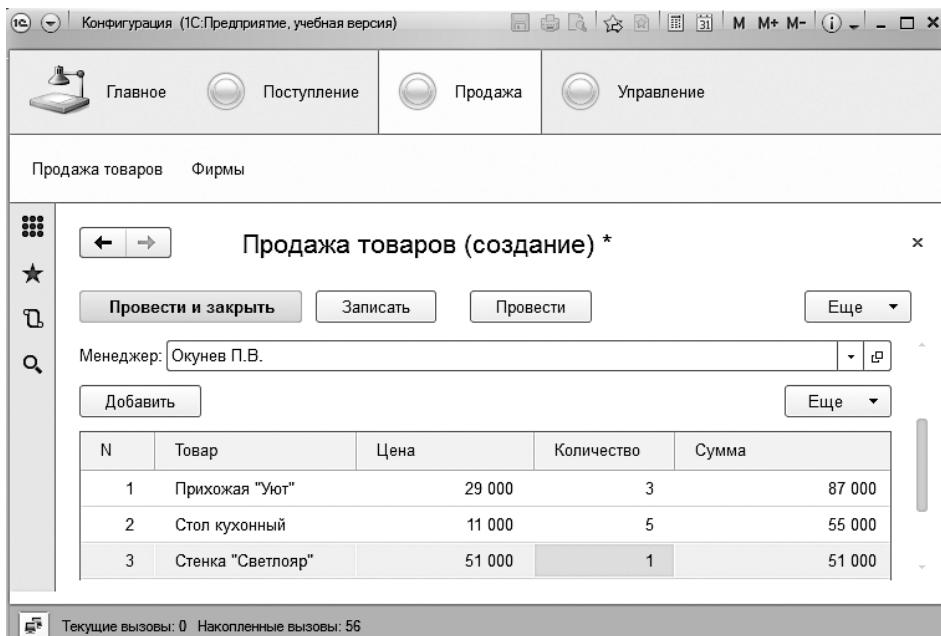


Рис. 1.82. Форма документа ПродажаТоваров в режиме 1С:Предприятие

Листинг 1.8. Процедура проведения документа ПродажаТоваров

```

Процедура ОбработкаПроведения(Отказ, Режим)
//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
Движения.КоличествоТоваров.Записывать = Истина;
Для Каждого ТекСтрокаПереченьТоваров Из ПереченьТоваров Цикл
// регистр КоличествоТоваров Расход
Движение = Движения.КоличествоТоваров.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Товар = ТекСтрокаПереченьТоваров.Товар;
Движение.Филиал = Филиал;
Движение.Количество = ТекСтрокаПереченьТоваров.Количество;
КонецЦикла;
//}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры

```

Теперь осталось перейти в режим 1С:Предприятие и ввести документ ПродажаТоваров (один варианта документа показан на рис. 1.82).

В результате проведения документа продажи товаров в регистре Количество Товаров появились новые записи (рис. 1.83).

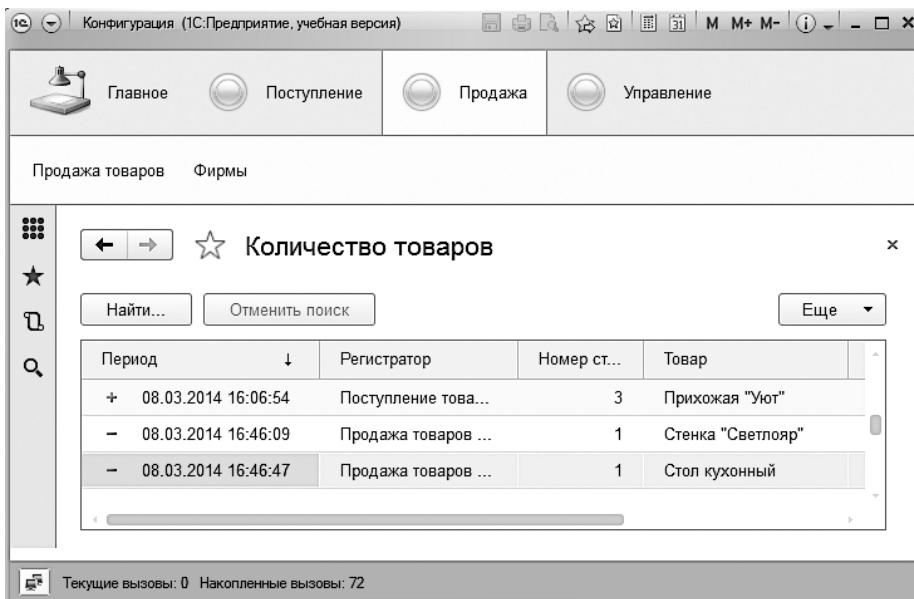


Рис. 1.83. Изменения в регистре Количество Товаров после продажи товаров

Перемещение товаров на выставку и обратно

Ситуация, когда некоторые товары перемещаются с филиала на выставку (а затем возвращаются обратно в филиал), достаточно типична. Для реализации данной функциональности создадим два документа:

- ПеремещениеНаВыставку;
- ПеремещениеСВыставки.

Начнем с документа ПеремещениеНаВыставку. После задания имени документа на вкладке Основные перейдем на вкладку Данные для формирования реквизитов. Так, для создаваемого документа в области шапки разместим реквизит Филиал, в качестве типа данных которого установим СправочникСсылка.Филиалы. После этого в документе создадим табличную часть, которую традиционно для данной главы назовем Перечень Товаров, а уже в ней разместим реквизиты (в скобках указаны соответствующие типы данных):

- Товар (тип данных — СправочникСсылка.Товары);
- Количество (тип данных — число).

В результате вкладка **Данные** должна выглядеть так, как показано на рис. 1.84.

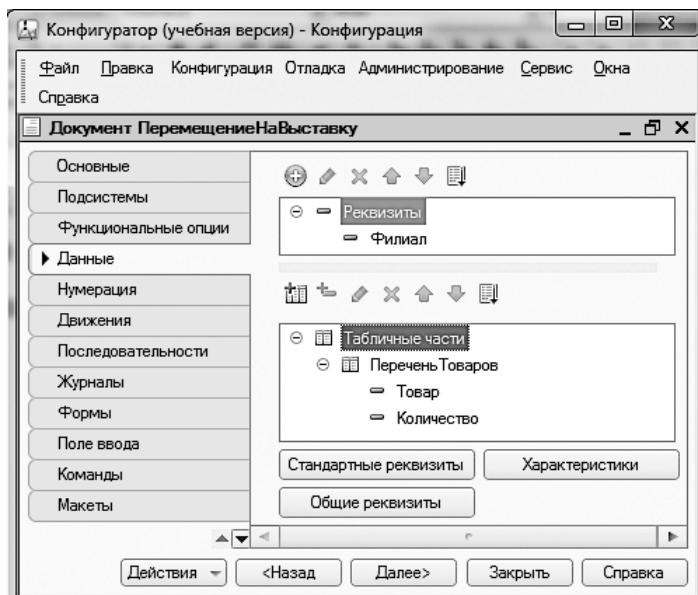


Рис. 1.84. Вкладка Данные окна редактирования документа ПеремещениеНаВыставку

Далее необходимо реализовать проводки документа **ПеремещениеНаВыставку**, поэтому с помощью конструктора движений (вкладка **Движения**) сформируем соответствие полей регистра **КоличествоТоваров** и реквизитов документа (рис. 1.85). Учитывая, что перемещения товаров на выставку является расходом ресурса, следует установить переключатель **Расход**. После этого щелчком на кнопке **OK** генерируем необходимую процедуру (листинг 1.9).

Листинг 1.9. Процедура проведения документа ПеремещениеНаВыставку

```
Процедура ОбработкаПроведения(Отказ, Режим)
//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
Движения.КоличествоТоваров.Записывать = Истина;
Для Каждого ТекСтрокаПереченьТоваров Из ПереченьТоваров Цикл
    // регистр КоличествоТоваров Расход
    Движение = Движения.КоличествоТоваров.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Товар = ТекСтрокаПереченьТоваров.Товар;
    Движение.Филиал = Филиал;
    Движение.Количество = ТекСтрокаПереченьТоваров.Количество;
КонецЦикла;
//}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

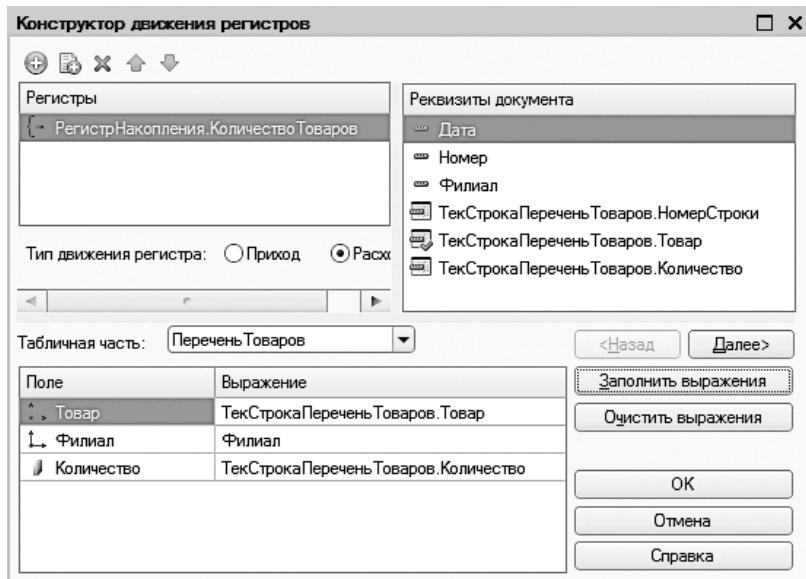


Рис. 1.85. Окно конструктора движений для документа ПеремещениеНаВыставку

Отнесем созданный документ ПеремещениеНаВыставку к подсистеме Управление (рис. 1.86).

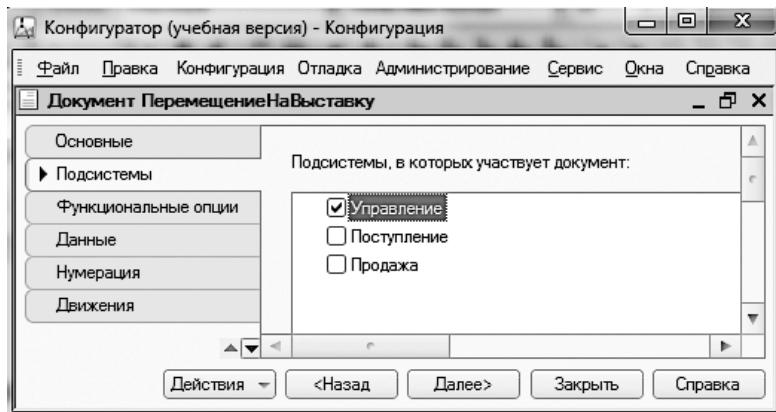


Рис. 1.86. Регистрация документа ПеремещениеНаВыставку в подсистеме Управление

В результате выполненных действий в режиме 1С:Предприятие с помощью документа ПеремещениеНаВыставку мы можем зарегистрировать перемещение товаров с наших филиалов на выставку. На рис. 1.87 приведено состояние регистра накопления КоличествоТоваров после проведения одного из документов ПеремещениеНаВыставку.

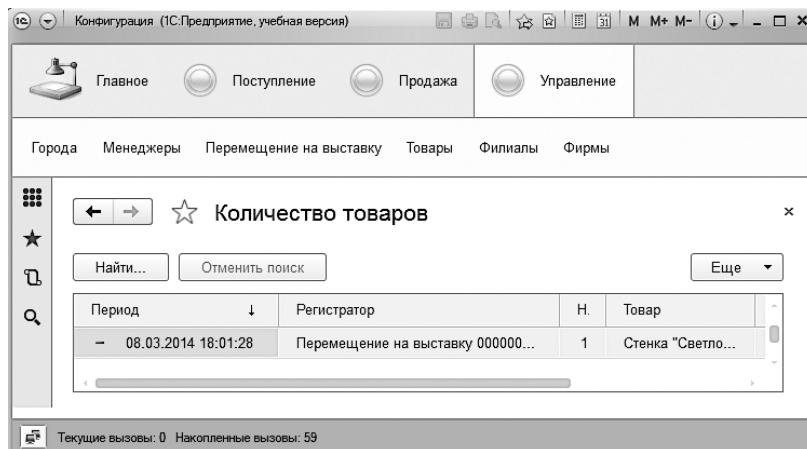


Рис. 1.87. Изменения в регистре Количество Товаров после перемещении товара на выставку

Теперь нам требуется создать еще один (аналогичный) документ, который назовем **ПеремещениеСВыставки**. Уже по названию понятно, что он будет описывать перемещение товаров в обратном направлении.

Вкладка **Данные** окна редактирования данного документа в режиме конфигуратора представлена на рис. 1.88. Как видно, реквизиты разрабатываемого документа полностью аналогичны реквизитам предыдущего документа **ПеремещениеНаВыставку**.

Для обеспечения движений по регистру **Количество Товаров** также воспользуемся конструктором движений. Рисунок 1.89 иллюстрирует соответствие полей регистра и реквизитов документа в окне конструктора движений.

В результате после щелчка на кнопке **OK** мы автоматически получим необходимую процедуру проведения документа **ПеремещениеСВыставки** (листинг 1.10).

Листинг 1.10. Процедура проведения документа ПеремещениеСВыставки

```
Процедура ОбработкаПроведения(Отказ, Режим)
//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
Движение.КоличествоТоваров.Записывать = Истина;
Для Каждого ТекСтрокаПереченьТоваров Из ПереченьТоваров Цикл
    // регистр Количество Товаров Приход
    Движение = Движение.КоличествоТоваров.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    Движение.Период = Дата;
    Движение.Товар = ТекСтрокаПереченьТоваров.Товар;
    Движение.Филиал = Филиал;
    Движение.Количество = ТекСтрокаПереченьТоваров.Количество;
КонецЦикла;
//}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

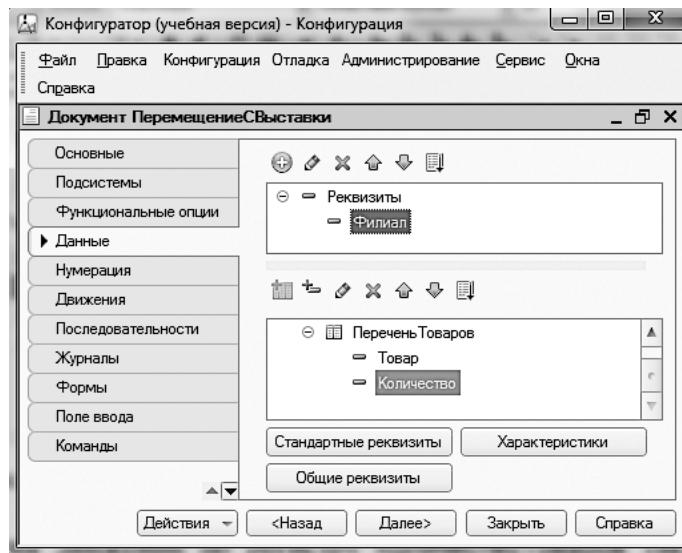


Рис. 1.88. Вкладка Данные окна редактирования документа ПеремещениеСВыставки

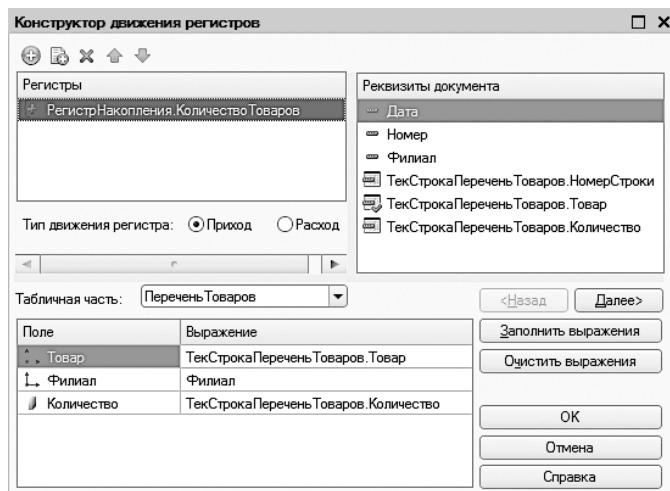


Рис. 1.89. Окно конструктора движений

Таким образом, теперь все четыре необходимых документа для регистрации товаров готовы. Мы можем осуществлять учет поступлений и продаж товаров, а также перемещений между филиалами и выставкой. На рис. 1.90 показана форма списка регистра КоличествоТоваров, отражающая проводки по разным типам созданных документов.

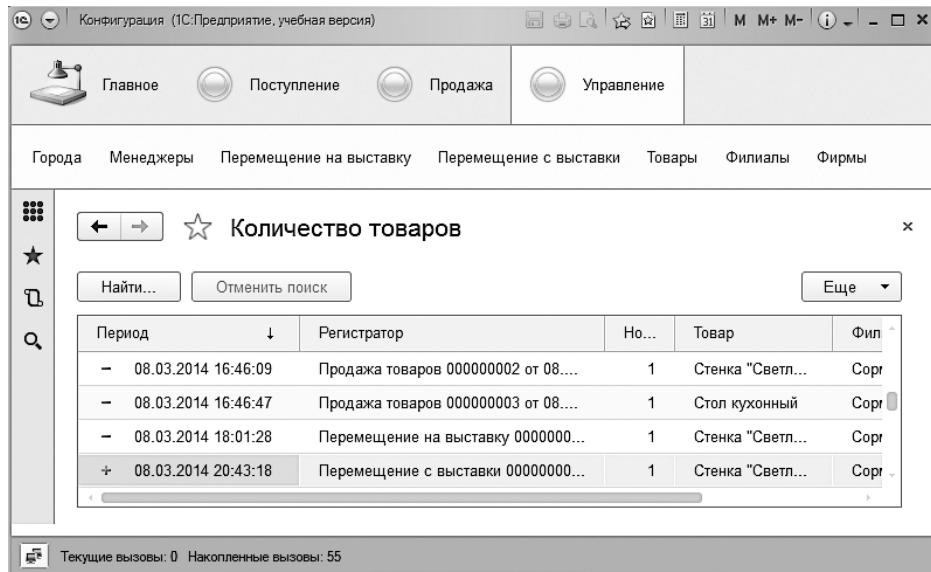


Рис. 1.90. Содержание регистра накопления Количество Товаров

Оборотный регистр накопления

Регистры накопления могут быть двух видов: *регистры остатков* и *регистры оборотов*. В предыдущих разделах этой главы для задач количественного учета мы использовали регистр накопления остатков. Для регистров подобного вида система 1С:Предприятие создает несколько виртуальных таблиц: таблицу остатков, таблицу оборотов и совместную таблицу остатков и оборотов [6]. Регистр оборотов отличается от регистра остатков тем, что для него система создает только одну виртуальную таблицу — оборотов. Об этом мы подробно поговорим в последующих главах.

Кроме того, есть еще один момент, отличающий оборотные регистры от остаточных. Если вернуться к предыдущим примерам главы, то мы увидим, что у регистра остатков присутствует параметр **Вид движения**. В этом случае для каждой записи в регистр накопления необходимо установить одно из двух значений данного параметра — **Приход** или **Расход**. Подобный параметр у оборотного регистра отсутствует.

Продолжим разработку конфигурации и для учета продаж по филиалам создадим оборотный регистр накопления ПродажиПоФилиалам. Вначале на вкладке Основные в окне редактирования объекта конфигурации необходимо задать начальную информацию, определяющую имя и вид регистра (рис. 1.91). Здесь укажем вид данного регистра — Обороты.

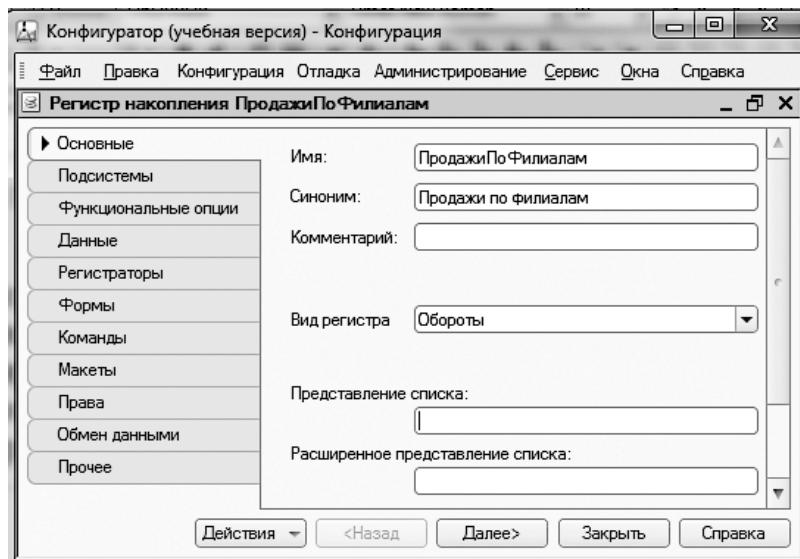


Рис. 1.91. Вкладка Основные окна редактирования регистра ПродажиПоФилиалам

Далее на вкладке **Данные** следует ввести два измерения (рис. 1.92):

- Филиал (тип данных — СправочникСсылка.Филиалы);
- Менеджер (тип данных — СправочникСсылка.Менеджеры).

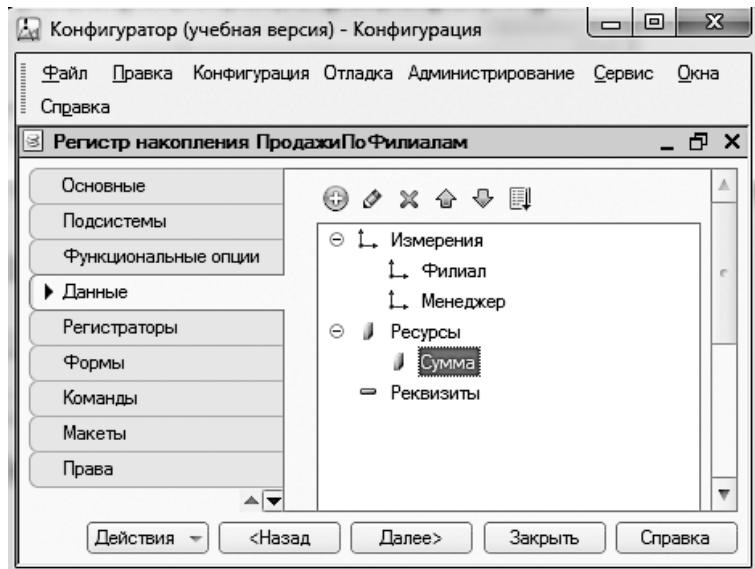


Рис. 1.92. Вкладка Данные окна редактирования регистра ПродажиПоФилиалам

В качестве ресурса укажем — Сумма (тип данных — число).

Организация движений по регистру ПродажиПоФилиалам должна быть организована с помощью документа ПродажаТоваров. В связи с этим необходимо скорректировать уже имеющуюся процедуру проведения данного документа. Следует обратить внимание на то, что здесь мы встретились с ситуацией, когда один и тот же документ осуществляет движения по нескольким регистрам.

Воспользуемся конструктором движений, где следует указать, что документ будет осуществлять движения сразу по двум регистрам (рис. 1.93).

ВНИМАНИЕ

При запуске конструктора движений следует согласиться с тем, что процедура ОбработкаПроведения будет замещена.

В окне конструктора отображаются ранее сформированные соответствия полей регистра КоличествоТоваров и реквизитов документа ПродажаТоваров. Сейчас в список регистров следует добавить еще один — ПродажиПоФилиалам, а далее в нем необходимо выбрать табличную часть и заполнить выражения для полей данного регистра (рис. 1.94).

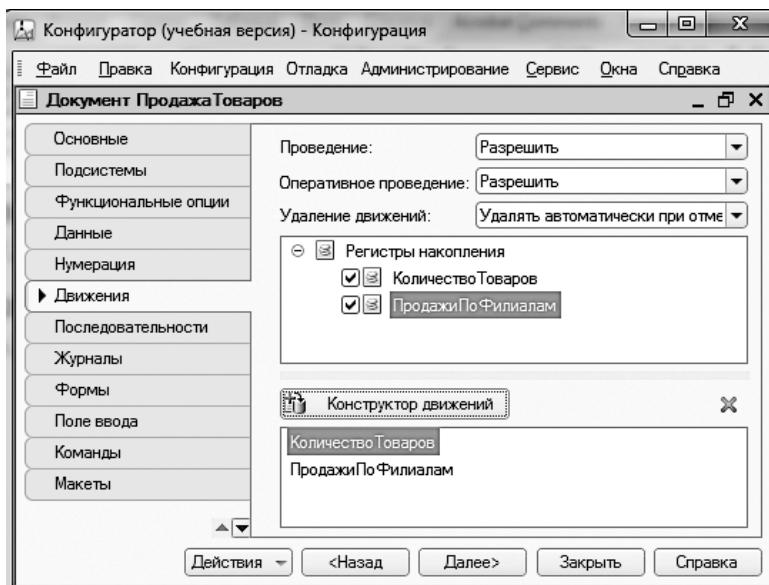


Рис. 1.93. Вкладка Движения окна редактирования документа ПродажаТоваров

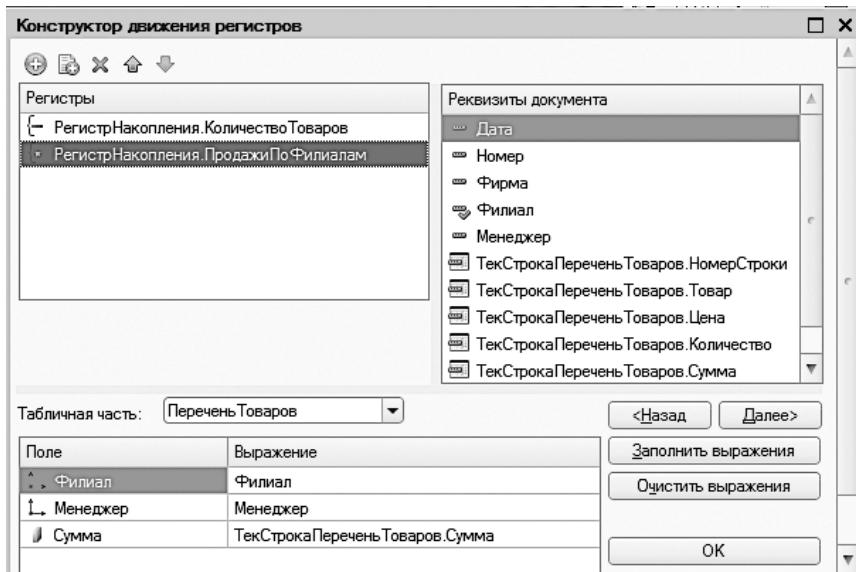


Рис. 1.94. Окно конструктора движений регистров

После щелчка на кнопке **OK** перед нами откроется текст новой процедуры проведения документа ПродажаТоваров, которую сформировал конструктор (листинг 1.11).

Листинг 1.11. Процедура проведения документа ПродажаТоваров по двум регистрам

```
Процедура ОбработкаПроведения(Отказ, Режим)
//{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
Движения.КоличествоТоваров.Записывать = Истина;
Для Каждого ТекСтрокаПереченьТоваров Из ПереченьТоваров Цикл
    // регистр КоличествоТоваров Расход
    Движение = Движения.КоличествоТоваров.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Товар = ТекСтрокаПереченьТоваров.Товар;
    Движение.Филиал = Филиал;
    Движение.Количество = ТекСтрокаПереченьТоваров.Количество;
КонецЦикла;
Движения.ПродажиПоФилиалам.Записывать = Истина;
Для Каждого ТекСтрокаПереченьТоваров Из ПереченьТоваров Цикл
    // регистр ПродажиПоФилиалам
    Движение = Движения.ПродажиПоФилиалам.Добавить();
    Движение.Период = Дата;
    Движение.Филиал = Филиал;
    Движение.Менеджер = Менеджер;
    Движение.Сумма = ТекСтрокаПереченьТоваров.Сумма;
КонецЦикла;
//}}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

Из данного текста видно, что конструктор создал два цикла обхода табличной части документа. При желании можно самостоятельно объединить эти циклы в один (листинг 1.12).

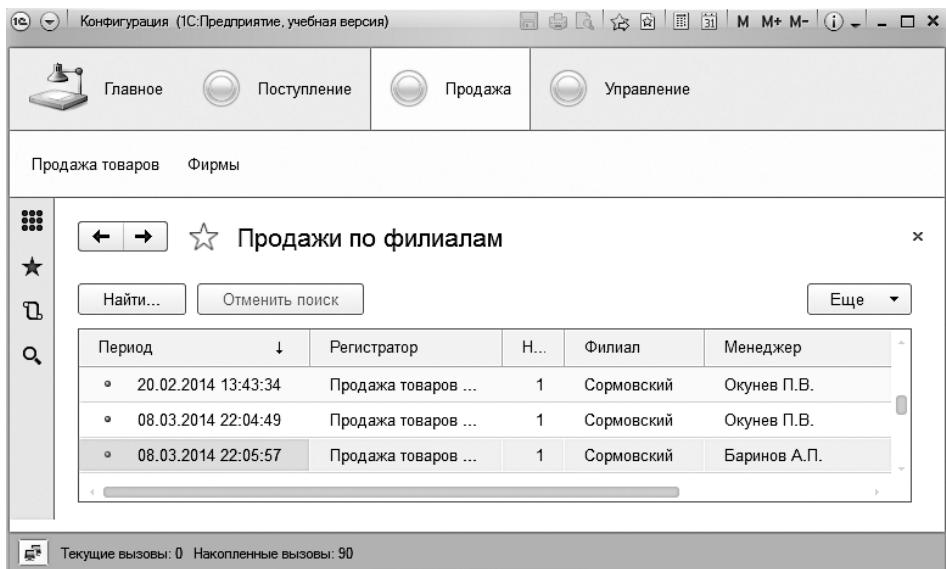


Рис. 1.95. Содержание регистра ПродажиПоФилиалам

Листинг 1.12. Измененная процедура проведения документа ПродажиПоФилиалам

```

Процедура ОбработкаПроведения(Отказ, Режим)
//{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
Движения.КоличествоТоваров.Записывать = Истина;
Движения.ПродажиПоФилиалам.Записывать = Истина;
Для Каждого ТекСтрокаПереченьТоваров Из ПереченьТоваров Цикл
// регистр КоличествоТоваров Расход
Движение = Движения.КоличествоТоваров.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Товар = ТекСтрокаПереченьТоваров.Товар;
Движение.Филиал = Филиал;
Движение.Количество = ТекСтрокаПереченьТоваров.Количество;
// регистр ПродажиПоФилиалам
Движение = Движения.ПродажиПоФилиалам.Добавить();
Движение.Период = Дата;
Движение.Филиал = Филиал;
Движение.Менеджер = Менеджер;
Движение.Сумма = ТекСтрокаПереченьТоваров.Сумма;
КонецЦикла;
//}}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры

```

Итак, все необходимые действия в режиме конфигуратора выполнены, осталось перейти в режим 1С:Предприятие, сформировать несколько документов и посмотреть результат в созданных нами регистрах накопления. На рис. 1.95 представлена форма списка оборотного регистра накопления ПродажиПоФилиалам.

Организация интерфейса приложения

В версию программы 1С:Предприятие 8.3 заложено множество ресурсов, позволяющих улучшить интерфейс прикладного решения.

Если взглянуть на созданное прикладное решение в режиме 1С:Предприятие, то мы увидим на панели разделов три однотипные по внешнему виду подсистемы (рис. 1.96). Для придания индивидуальности каждой подсистеме можно добавить уникальные значки, расположив их рядом с названиями подсистем.

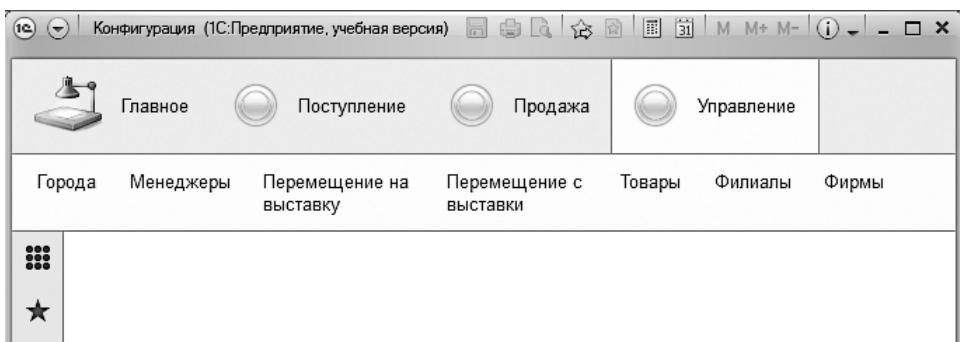


Рис. 1.96. Интерфейс прикладного решения

Рассмотрим шаги, необходимые для реализации такого оформления. В качестве первого шага откроем в конфигураторе окно редактирования сначала одной из подсистем, для определенности пусть это будет подсистема Управление (рис. 1.97).

Для установки оригинального изображения следует в этом окне щелкнуть на значке в виде трех точек в поле Картинка. После этого требуется найти на компьютере подходящую миниатюру, которая будет отображаться рядом с названием подсистемы Управление. При этом платформа 1С:Предприятие 8.3 предлагает сделать выбор из набора стандартных миниатюр либо самостоятельно подобрать необходимый рисунок из имеющихся в компьютере. И в том, и в другом случаях мы можем зафиксировать выбранное изображение в качестве значка для рассматриваемой подсистемы. Аналогичным образом следует выбрать картинки для подсистем Поступление и Продажа. После проделанных действий окно нашего прикладного решения в режиме 1С:Предприятие может выглядеть так, как показано на рис. 1.98.

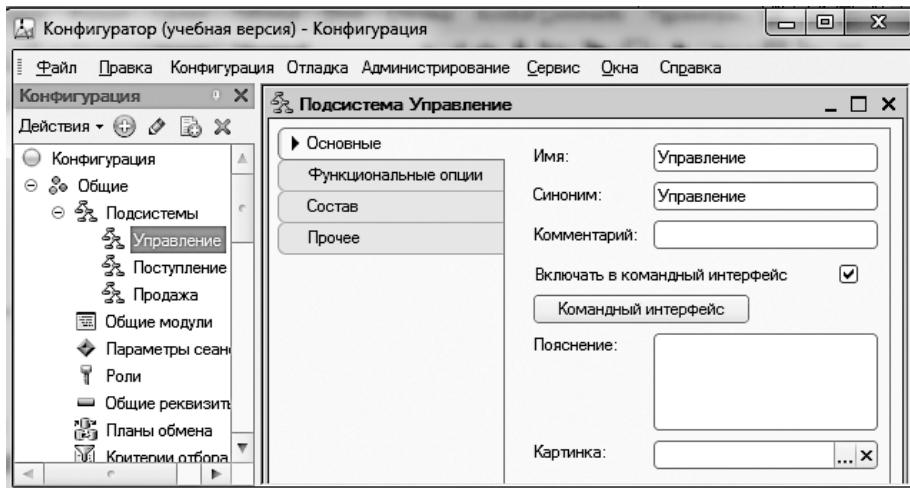


Рис. 1.97. Окно редактирования подсистемы Управление

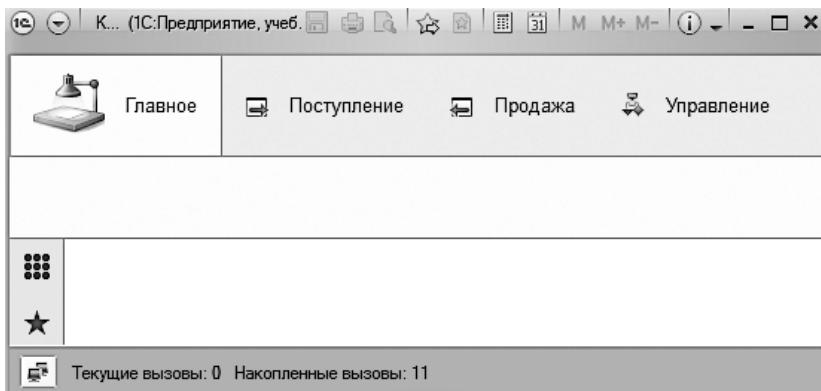


Рис. 1.98. Добавление значков подсистем в интерфейс приложения

Теперь поясним, каким образом осуществляется просмотр регистров накопления. Для большинства ситуаций лучше всего исключить возможность просмотра содержимого регистров накопления в подсистеме наряду со справочниками и документами, но при этом дать пользователю возможность доступа к их содержимому. Эту возможность можно реализовать с помощью пункта **Все функции** главного меню программы (рис. 1.99).

Если воспользоваться данным пунктом меню, то перед нами откроется окно, показанное на рис. 1.100. Здесь следует открыть узел **Регистры накопления**, после чего воспользоваться формами списков объектов для просмотра содержимого регистров **КоличествоТоваров** и **ПродажиПоФилиалам**.

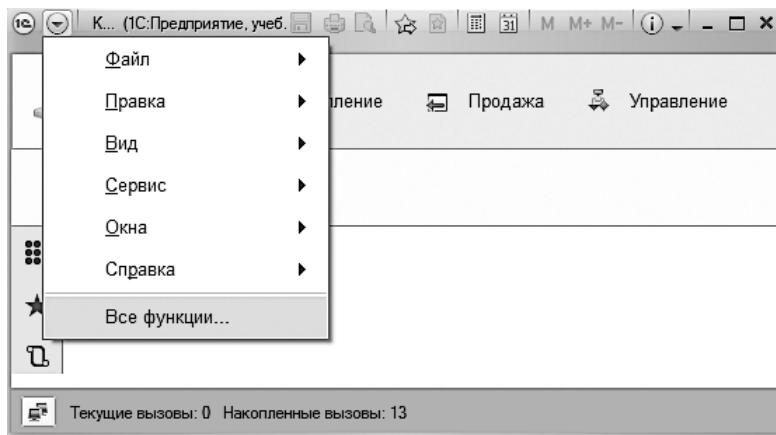


Рис. 1.99. Главное меню приложения

Важно отметить, что для включения самого пункта Все функции в меню требуется в главном меню (см. рис. 1.99) раскрыть подменю Сервис и выбрать пункт Параметры, а затем в открывшемся окне установить флажок Отображать команду «Все функции» (рис. 1.101).

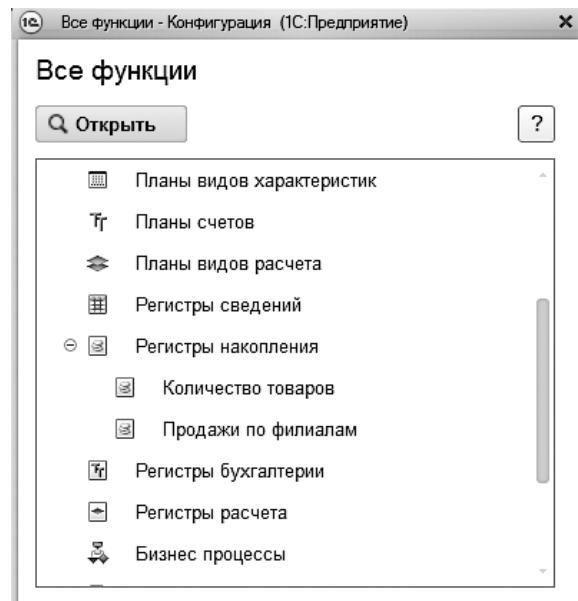


Рис. 1.100. Раздел Все функции

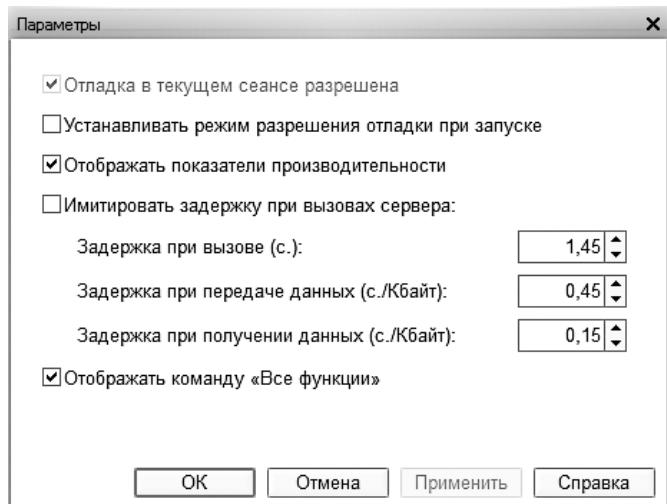


Рис. 1.101. Окно установки основных параметров системы

Итоги

Самый важный результат чтения данной главы заключается в том, что мы узнали, как выглядит технология разработки конфигураций для системы 1С:Предприятие 8.3. Также мы познакомились с основными элементами интерфейса этого программного продукта. В плане программирования были сделаны первые шаги — мы увидели, что представляют собой программы на языке 1С. Основным инструментальным средством системы 1С:Предприятие 8.3 являются конструкторы, которые значительно облегчают труд разработчика. На примерах, связанных с использованием конструкторов форм, движений и табличных документов, мы познакомились с приемами, позволяющими существенно ускорить процесс создания программного кода.

Глава 2

Язык запросов в системе 1С:Предприятие

В первой главе мы рассмотрели технологию создания объектов конфигурации, необходимых для прикладных решений, а также процесс ввода данных в информационную базу в режиме 1С:Предприятие. Для решения прикладной задачи по учету поступлений и продаж товаров было произведено заполнение ряда справочников. Необходимость учета событий, привязанных ко времени (поступление, продажа и перемещение товаров), заставила нас разработать соответствующие документы. В свою очередь, информация, содержащаяся в документах в процессе их проведения, приводила к изменениям в регистрах накопления.

Если обратиться к разработанной в первой главе информационной базе, то мы увидим списки созданных документов, перечни элементов в справочниках и записи в регистрах. Фактически эта информация представляет собой данные в исходной форме после их ввода пользователем в режиме 1С:Предприятие. Как правило, в этом случае сложно быстро получить ответ на конкретный вопрос, связанный с анализом данных (по сотрудникам, фирмам, поступлениям и продажам товаров). Разумеется, всегда можно открыть необходимый справочник, просмотреть и проанализировать содержимое регистров или список документов, однако часто такой подход отнимает слишком много времени.

Для руководящего состава любой организации больший интерес представляет информация в существенно обработанном виде (отбор по параметрам, итоговые и сводные данные). Например, поступления товаров за указанный временной период с группировкой по филиалам или списки сотрудников фирм-партнеров с указанием их телефонов.

В системе 1С:Предприятие для решения подобных задач используется специальный объект, который называется *запрос*. По существу, любой запрос представляет собой требование к системе получить из базы данных выборку с необходимой информацией. На практике данный объект наиболее часто используется при построении отчетов.

Запрос представляет собой текст на специальном языке запросов. В этом тексте описываются источник информации для запроса и условия построения запроса.

В системе 1С:Предприятие в качестве источника информации для запроса указываются таблицы информационной базы, а также поля таблиц, которые требуется обрабатывать в запросе.

Таблицы, участвующие в запросе, делятся на два основных класса: реальные и виртуальные. Их отличие заключается в том, что реальные таблицы хранятся в базе данных, а виртуальных таблиц в информационной базе нет. При обращении к информации виртуальных таблиц система самостоятельно автоматически собирает информацию из реальных таблиц базы данных для выполнения запроса. В практических задачах *виртуальные таблицы используются существенно чаще*, поскольку информация в них уже сгруппирована.

ПРИМЕЧАНИЕ

Отдельный подкласс таблиц образуют так называемые объектные таблицы. В качестве объектной таблицы всегда выступает реальная таблица базы данных. Объектные таблицы предназначены для хранения состояния объектов системы 1С:Предприятие, таких как справочники, документы, регистры и др. Каждая объектная таблица имеет один из типов объектов системы 1С:Предприятие. Например, объектам типа Справочник.Фирмы соответствует одна таблица, а объектам Справочник.Товары — другая. Каждая отдельная запись объектной таблицы хранит состояние определенного объекта соответствующего типа. Поэтому у каждой объектной таблицы определено поле ссылки на текущую запись.

Наряду с обычным отбором информации у запросов имеется еще много полезных свойств — они обеспечивают выполнение группировки отобранных данных, вычисление итогов и др. На практике при реализации сложных прикладных решений запросы используются в тех ситуациях, когда требуется сформировать сложную выборку данных, сгруппированную и отсортированную необходимым образом.

В данной главе мы рассмотрим технологию составления запросов, начиная с основных положений. После рассмотрения примеров разной степени сложности вы получите практические навыки по извлечению необходимых сведений из информационной базы. В дальнейшем, практически на протяжении всей книги, при составлении отчетов мы будем работать с запросами.

ВНИМАНИЕ

Запросы в системе 1С:Предприятие предназначены для выборки информации из базы данных. Можно считать, что запрос является требованием к системе по выборке и обработке данных. Например, с помощью запроса можно легко подсчитать внесенные суммы по конкретным контрагентам. В системе 1С:Предприятие существует специальный язык запросов, цель которого заключается в формулировании требований к системе — что нужно отобрать из имеющихся данных. А как система справляется с запросом, нас уже не интересует, главное — поставить задачу.

Далее в этой главе рассматриваются синтаксические конструкции языка запросов и приводятся примеры их использования в практических задачах.

Организация выборки из таблицы

Сейчас мы не будем создавать новую информационную базу, а воспользуемся разработкой из первой главы. Фактически уже созданные там справочники, документы и регистры станут исходным материалом для выполнения примеров данной главы. Для начала рассмотрим запрос, который будет связан с информацией, присущей в табличной части справочника Фирмы. В первой главе при его разработке мы создали в нем табличную часть КонтактныеЛица, в которую включили фамилии сотрудников, их должности и телефоны.

Вполне реальная задача может касаться необходимости составления списка всех лиц, работающих в фирмах-партнерах (точнее лиц, зафиксированных в табличной части КонтактныеЛица). В таком списке напротив фамилии каждого сотрудника должен быть указан номер его телефона. Также информацию о сотруднике желательно дополнить еще и названием организации, где он работает. Разумеется, список следует отсортировать по фамилиям сотрудников (по алфавиту в возрастающем порядке). Текст такого запроса выглядит так, как показано в листинге 2.1.

Листинг 2.1. Простейший запрос выборки

```
ВЫБРАТЬ Сотрудник, Телефон,  
        Ссылка.Наименование КАК Организация  
ИЗ Справочник.Фирмы.КонтактныеЛица  
УПОРЯДОЧИТЬ ПО Сотрудник
```

После выполнения указанного запроса мы получим таблицу из набора строк с тремя столбцами. Число строк в таблице соответствует числу имеющихся лиц в табличной части всех элементов справочника Фирмы.

В приведенном запросе использовано важнейшее ключевое слово **ВЫБРАТЬ**, позволяющее указать список полей для данного запроса. Другое ключевое слово **ИЗ** дает возможность определить таблицы, участвующие в рассматриваемом запросе. В данной ситуации используется табличная часть КонтактныеЛица справочника Фирмы.

Необходимо прокомментировать один важный момент в конструкции данного запроса. Так, при запросе к табличной части справочника обращение к обычным (вне таблицы) реквизитам справочника производится через поле **Ссылка**. В данном случае информация о фирме заключена в автоматически создаваемом в любом справочнике реквизите **Наименование**, который располагается вне табличной части.

В приведенном запросе мы использовали ключевое слово **КАК**, позволяющее определить синоним для сочетания **Ссылка.Наименование**. Вместо такой длинной конструк-

ции выбран более короткий вариант — **Организация**. Для упорядочивания данных запроса используется конструкция **УПОРЯДОЧИТЬ ПО**, в которой необходимо указать название поля (или полей), по которому будет производиться сортировка.

Важный технический момент связан с выполнением запросов в системе 1С:Предприятие. Здесь мы поступим следующим образом. Учитывая, что для формирования отчетов в программном продукте 1С:Предприятие 8.3 ключевое место занимает система компоновки данных, то и процесс выполнения запросов в данной главе мы реализуем с помощью этого ресурса. Сама система компоновки данных будет подробно рассматриваться в третьей главе книги, поэтому здесь мы не будем ее детально комментировать, а только опишем, какие технические действия требуются для того, чтобы выполнить рассматриваемые запросы.

Важно отметить, что в системе 1С:Предприятие 8.3 тексты на языке запросов можно формировать с помощью **конструктора запросов**. В этом случае от вас при составлении запроса требуются только визуальные действия (они осуществляются с помощью мыши). В третьей главе книги мы так и будем поступать, однако здесь для нас интерес представляют непосредственно сами конструкции на языке запросов.

Для работы с запросами создадим еще одну подсистему, **Аналитика**, в которой будем выводить создаваемые отчеты, построенные на основании запросов. Техническое выполнение этого действия нам хорошо знакомо из первой главы книги.

Следующий шаг заключается в создании нового объекта конфигурации — объекта, который строится на основании отчета и относится к базовым объектам конфигурации. Для этого в окне **Конфигурация** необходимо щелкнуть правой кнопкой мыши на разделе **Отчеты**. В окне редактирования объекта конфигурации на вкладке **Основные** введем имя разрабатываемого отчета — **Отчет2_1** (рис. 2.1).

ПРИМЕЧАНИЕ

Такой стиль обозначений мы будем использовать и в дальнейшем: первая цифра в названии отчета соответствует номеру главы, а вторая — порядковому номеру отчета внутри главы.

Далее на этой вкладке воспользуемся кнопкой **Открыть схему компоновки данных**. В результате перед нами откроется окно конструктора макета, в котором мы ничего менять не будем (рис. 2.2). После щелчка на кнопке **Готово** на экране появится окно конструктора схемы компоновки данных с многочисленными вкладками, среди которых сейчас нас интересует вкладка **Наборы данных** (рис. 2.3).

Здесь с помощью кнопки **Добавить набор данных** (можно также воспользоваться контекстным меню, открывающимся по щелчку правой кнопкой мыши на значке

Наборы данных) выберем вариант Добавить набор данных — запрос. В результате окно на экране изменится — в его нижней части появится поле для запроса (рис. 2.4). Внесем в это поле текст запроса в соответствии с рис. 2.5.

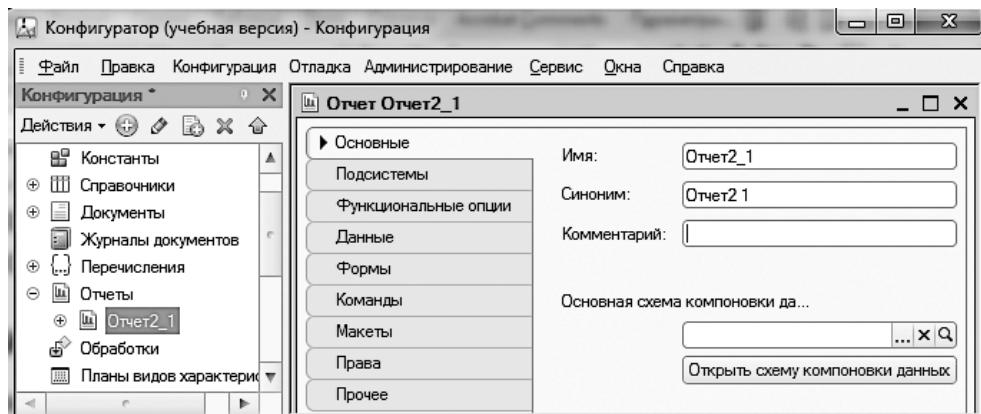


Рис. 2.1. Вкладка Основные окна редактирования объекта конфигурации Отчет2_1

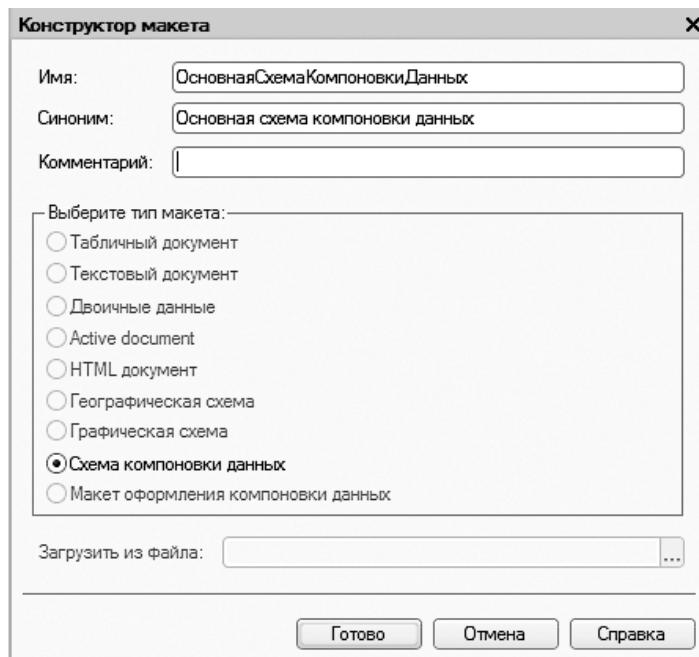


Рис. 2.2. Окно конструктора макета

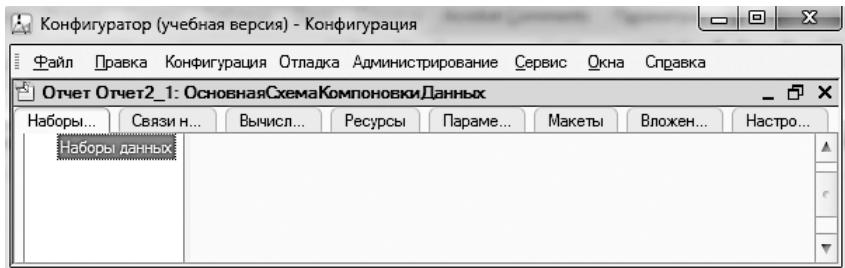


Рис. 2.3. Окно конструктора схемы компоновки данных

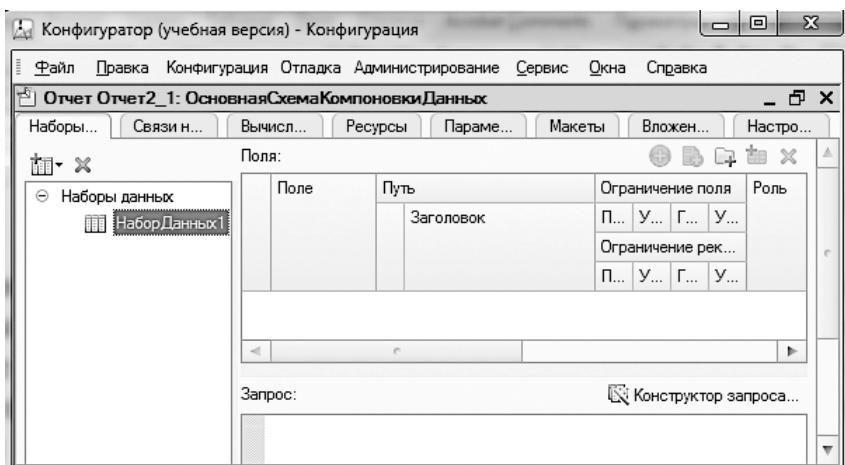


Рис. 2.4. Добавление набора данных в схему компоновки данных

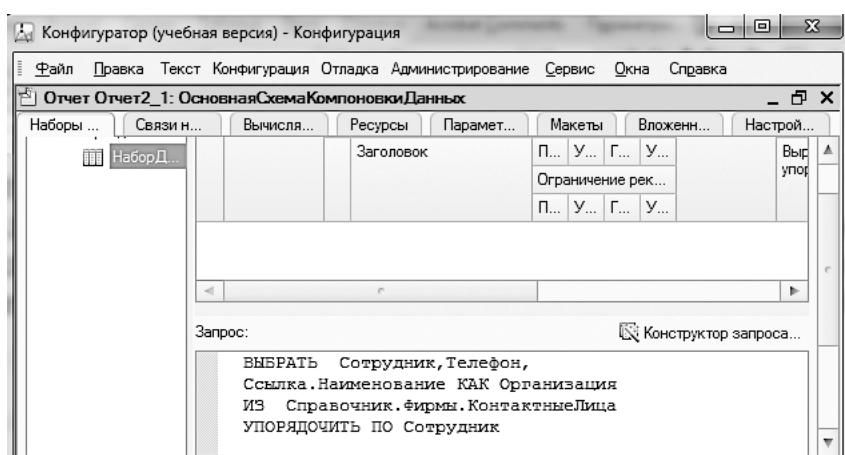


Рис. 2.5. Текст запроса в окне конструктора схемы компоновки данных

После проделанных действий перейдем на вкладку Настройки (рис. 2.6).

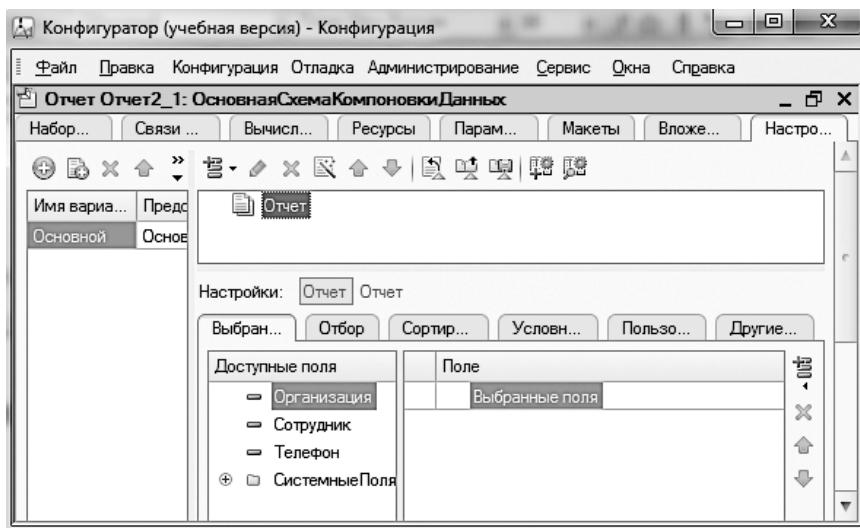


Рис. 2.6. Вкладка Настройки схемы компоновки данных

Здесь требуется перенести в раздел **Выбранные поля** все три интересующие нас поля, участвующие в запросе. Кроме того, необходимо щелчком правой кнопки на слове **Отчет** вызвать контекстное меню. В результате на экране появится окно для задания группировки в отчете (рис. 2.7). В данном случае нам никакого поля указывать не требуется, поэтому просто щелкнем на кнопке **OK**. После этого вместо вкладка **Настройки** будет выглядеть так, как показано на рис. 2.8.

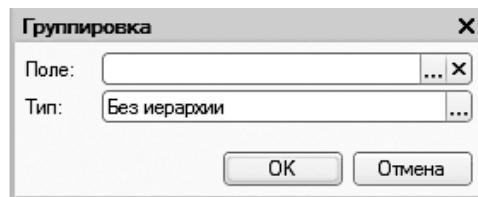


Рис. 2.7. Окно для выбора группировки в отчете

ПРИМЕЧАНИЕ

В следующих примерах этой главы описанные технические действия мы будем использовать при подготовке всех запросов. Дополнительные возможности системы компоновки данных рассматриваются в третьей главе.

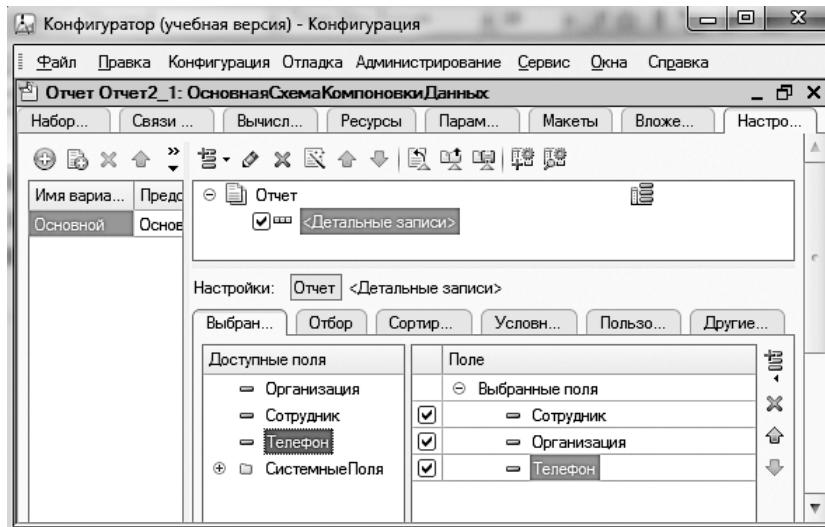


Рис. 2.8. Формирование полей для схемы компоновки данных

После выполненных действий осталось перейти в режим 1С:Предприятие и в подсистеме Аналитика воспользоваться только что созданным отчетом (рис. 2.9).

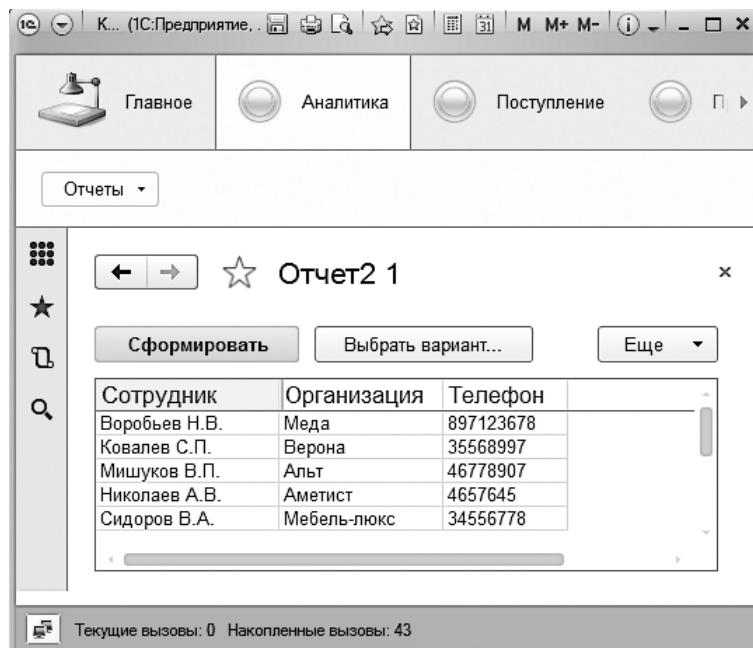


Рис. 2.9. Отчет2_1 в режиме 1С:Предприятие

Сортировка в запросах

Понятно, что любая отобранныя информация должна быть каким-то образом отсортирована, чтобы пользователь получил интересующие его данные в максимально удобном для восприятия виде.

Можно, например, изменить текст предыдущего запроса с целью расположить отобранные элементы в порядке убывания номеров телефонов (см. рис. 2.5). Для этого текст запроса должен выглядеть в соответствии с листингом 2.2. Ключевое слово УБЫВ в конструкции УПОРЯДОЧИТЬ говорит о том, что отобранныя информация должна быть отсортирована в порядке убывания значений указанного поля (в данном случае — номеров телефонов).

Листинг 2.2. Запрос с упорядочиванием данных по номерам телефонов

```
ВЫБРАТЬ Сотрудник.Телефон,
Ссылка.Наименование КАК Организация
ИЗ Справочник.Фирмы.КонтактныеЛица
УПОРЯДОЧИТЬ ПО Телефон УБЫВ
```

Перейдем теперь к техническим действиям, необходимым для выполнения рассмотренного запроса. Для этого аналогично предыдущему примеру создадим новый отчет с использованием схемы компоновки. На этот раз, в соответствии с листингом 2.2 в поле для запроса введем новую конструкцию (рис. 2.10).

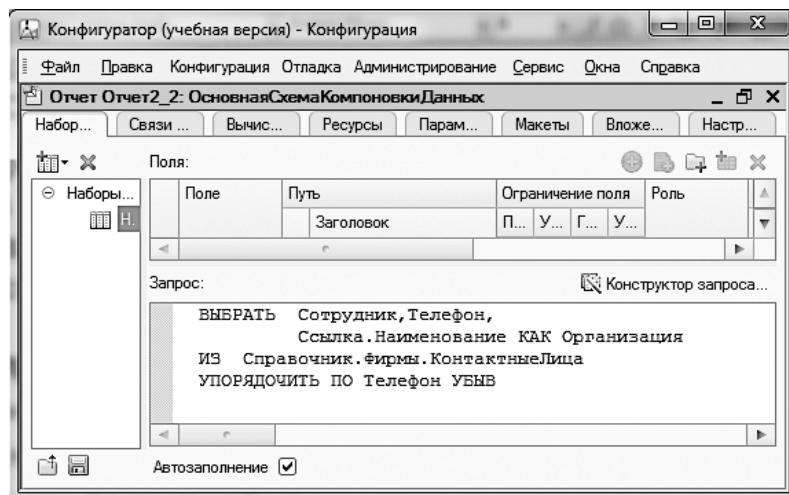


Рис. 2.10. Построение запроса с упорядочиванием по номерам телефонов

Вкладку **Настройки** следует оформить так, как мы уже видели на рис. 2.8 (параметры здесь такие же, как в предыдущей разработке).

На этом работа по созданию отчета в конфигураторе завершается, остается лишь перейти в режим 1С:Предприятие, чтобы увидеть результат выполнения данного запроса (рис. 2.11). Видно, что в отобранных данных наибольшие значения номеров телефонов оказались вверху списка.

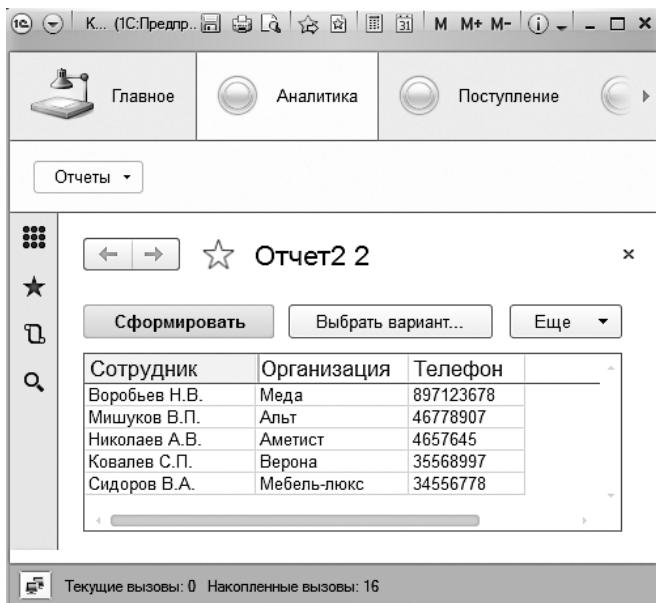


Рис. 2.11. Результат выполнения запроса с упорядочиванием по номерам телефонов

Чтобы в запросе упорядочить номера по возрастанию, следует применить конструкцию с использованием ключевого слова **ВОЗР**. Для рассмотренной ситуации это может выглядеть следующим образом:

УПОРЯДОЧИТЬ ПО Телефон ВОЗР

ПРИМЕЧАНИЕ

В предложении УПОРЯДОЧИТЬ ПО через запятую перечисляются условия, в соответствие с которыми необходимо упорядочить результат запроса. Выбранные данные упорядочиваются сначала по первому условию, затем по второму и т. д. В результате извлеченная запросом информация становится структурированной и удобной для восприятия.

В листинге 2.3 приведен запрос с последовательным упорядочиванием сначала по названиям организаций, а затем по именам сотрудников. Результат выполнения запроса для имеющейся информации в базе данных представлен на рис. 2.12.

Листинг 2.3. Запрос с упорядочиванием по двум полям

```
ВЫБРАТЬ Сотрудник, Телефон,
       Ссылка.Наименование КАК Организация
ИЗ Справочник.Фирмы.КонтактныеЛица
УПОРЯДОЧИТЬ ПО Организация ВОЗР, Сотрудник ВОЗР
```

Сотрудник	Организация	Телефон
Мишуков В.П.	Альт	46778907
Николаев А.В.	Аметист	4657645
Ковалев С.П.	Верона	35568997
Сидоров В.А.	Мебель-люкс	345567778
Афонин В.С.	Меда	897656787
Бочкирев А.С.	Меда	896544355
Воробьев Н.В.	Меда	897123678

Рис. 2.12. Результат выполнения запроса с упорядочиванием по организациям и сотрудникам

Рассмотрим еще один пример построения запроса, который будет касаться регистра накопления **КоличествоТоваров**. Будем считать, что нам необходимо получить информацию о движении товаров по нашим филиалам в следующем формате:

- Товар;
- Филиал;
- Регистратор;
- Количество.

В регистре накопления реквизит под названием *регистратор* представляет собой ссылку на документ, которому подчинена рассматриваемая запись регистра. Будем считать, что информацию, извлеченную запросом, необходимо упорядочить сначала по названиям филиалов (в возрастающем порядке), а уже внутри филиалов — по количеству товаров (в убывающем порядке). Текст запроса для реализации такого отбора данных приведен в листинге 2.4.

Листинг 2.4. Запрос для извлечения информации из регистра накопления

ВЫБРАТЬ

Количество Товаров.Товар КАК Товар,
 Количество Товаров.Филиал.Наименование КАК Филиал,
 Количество Товаров.Регистратор КАК Регистратор,
 Количество Товаров.Количество КАК Количество

ИЗ

РегистрНакопления.Количество Товаров КАК Количество Товаров

УПОРЯДОЧИТЬ ПО

Филиал, Количество УбыВ

Как и в предыдущих примерах, воспользуемся новым отчетом (назовем его Отчет2_3), при разработке которого создадим схему компоновки данных. Не будем останавливаться на уже описанных ранее технических шагах и отметим только, какие параметры необходимо установить на вкладке Настройки (рис. 2.13).

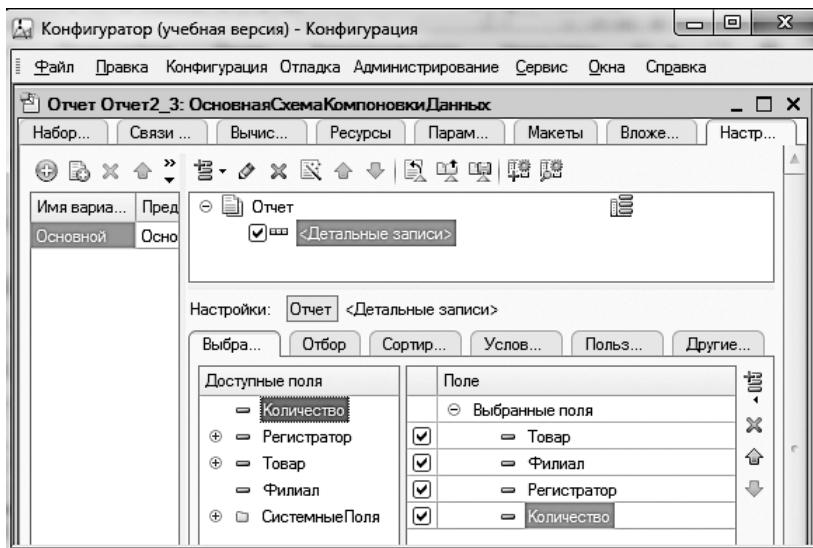


Рис. 2.13. Установки на вкладке Настройки для запроса, представленного в листинге 2.4

Теперь осталось перейти в режим 1С:Предприятие и сформировать разработанный отчет Отчет2_3 (рис. 2.14).

Отбор фрагмента данных

Часто для принятия решения не требуется просматривать все данные, а достаточно взглянуть на лучшие или, наоборот, худшие показатели среди отобранных сведений. Например, на показатели филиалов, дающих наибольшую прибыль.

Можно сказать, что при значительном объеме информации в результате запроса часто требуется просмотреть лишь несколько наибольших значений или, наоборот, несколько наименьших.

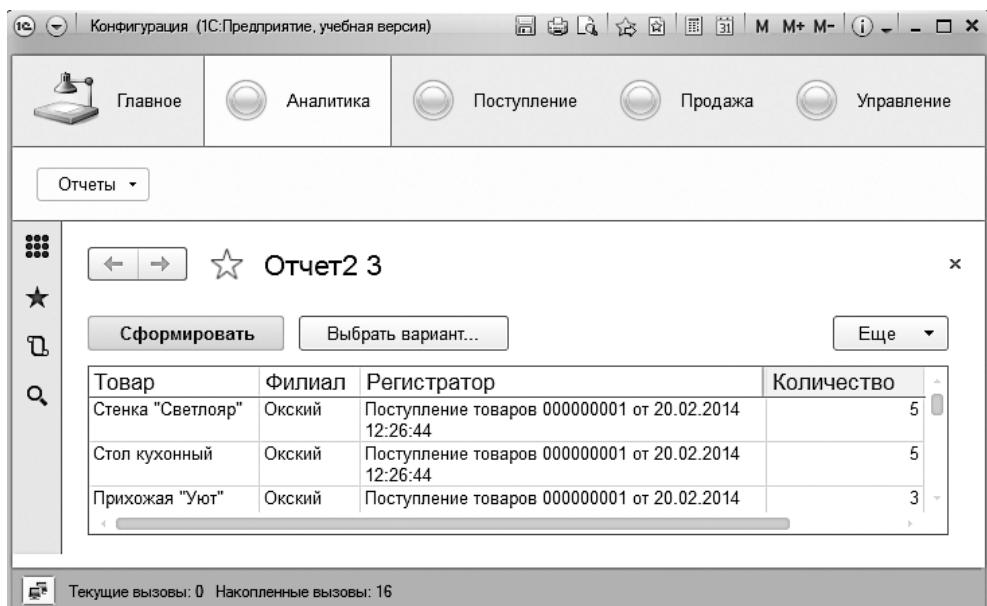


Рис. 2.14. Извлечение информации о движении товаров

При такой постановке задачи в конструкцию **ВЫБРАТЬ** добавляется ключевое слово **ПЕРВЫЕ**. Например, если нас интересуют позиции в документах **Поступление Товаров** (строки табличной части) с тремя самыми маленькими суммами поступления, то для этого можно использовать запрос из листинга 2.5.

Листинг 2.5. Запрос для выбора трех минимальных сумм при поступлении товаров

```
ВЫБРАТЬ ПЕРВЫЕ 3 Товар, Сумма,
Ссылка.Фирма КАК ФирмаПоставщик
ИЗ Документ.Поступление Товаров.Перечень Товаров
УПОРЯДОЧИТЬ ПО Сумма ВОЗР
```

Результат работы в режиме 1С:Предприятие отражен на рис. 2.15, где информация показана с учетом имеющихся данных в информационной базе.

Если бы нас наоборот интересовали три максимальных поступления по сумме, то в тексте листинга 2.5 потребовалось бы изменить только конструкцию, связанную с упорядочиванием полученных данных. Текст запроса для такого варианта приведен в листинге 2.6.

Листинг 2.6. Запрос для извлечения трех максимальных сумм поступления товаров

```
ВЫБРАТЬ ПЕРВЫЕ 3 Товар,Сумма,
Ссылка.Фирма КАК ФирмаПоставщик
ИЗ Документ.ПоступлениеТоваров.ПереченьТоваров
УПОРЯДОЧИТЬ ПО Сумма УбыВ
```

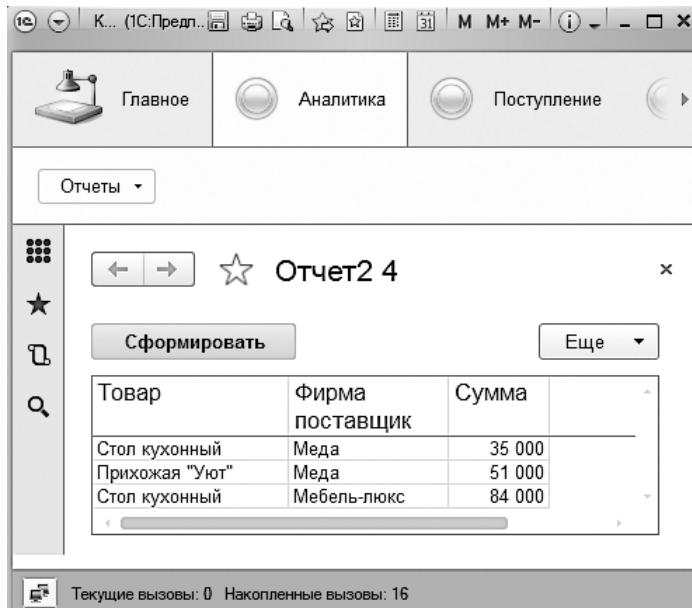


Рис. 2.15. Извлечение информации с помощью запроса, представленного в листинге 2.5

Аналогичный прием можно использовать и для ранее рассмотренного отбора по сотрудникам фирм-партнеров. Например, если нам необходимо получить только верхнюю часть списка сотрудников (упорядоченную по алфавиту), то текст запроса следует оформить так, как показано в листинге 2.7.

Листинг 2.7. Запрос для извлечения начального фрагмента списка сотрудников

```
ВЫБРАТЬ Первые 3 Сотрудник,Телефон,
Ссылка.Наименование КАК Организация
ИЗ Справочник.Фирмы.КонтактныеЛица
УПОРЯДОЧИТЬ ПО Сотрудник
```

Отбор уникальных значений

Теперь наша задача заключается в формировании перечня товаров, которые поступали от каждого поставщика по документам ПоступлениеТоваров, причем в этот список

каждое конкретное название товара в комбинации с указанием фирмы-поставщика должно попасть не более одного раза (даже если данный товар от указанного поставщика поступал несколько раз). Можно сказать, что в отчет должны попасть только уникальные комбинации поставщик-товар.

В языке запросов существует ключевое слово **РАЗЛИЧНЫЕ**, которое позволяет отобрать только различающиеся строки (в выборке, полученной в результате выполнения запроса, повторений не будет).

Возвратимся к нашему примеру и организуем выборку товаров из документов **ПоступлениеТоваров** вместе с названиями фирм без повторений данных комбинаций. Текст запроса, в котором реализуется подобный отбор, приведен в листинге 2.8. Заметим, что все встречающиеся в этом запросе синтаксические конструкции нам уже знакомы.

Листинг 2.8. Отбор уникальных комбинаций поставщик-товар

```
ВЫБРАТЬ РАЗЛИЧНЫЕ Товар.Наименование КАК Товар,
Ссылка.Фирма.Наименование КАК ФирмаПоставщик
ИЗ Документ.ПоступлениеТоваров.ПереченьТоваров КАК Перечень
УПОРЯДОЧИТЬ ПО Товар ВОЗР
```

Для проверки работоспособности данного запроса разработаем новый отчет — **Отчет2_5**. На рис. 2.16 показано ключевое (в плане задания необходимых параметров) окно схемы компоновки данных, а результат заполнения отчета в режиме 1С:Предприятие приведен на рис. 2.17.

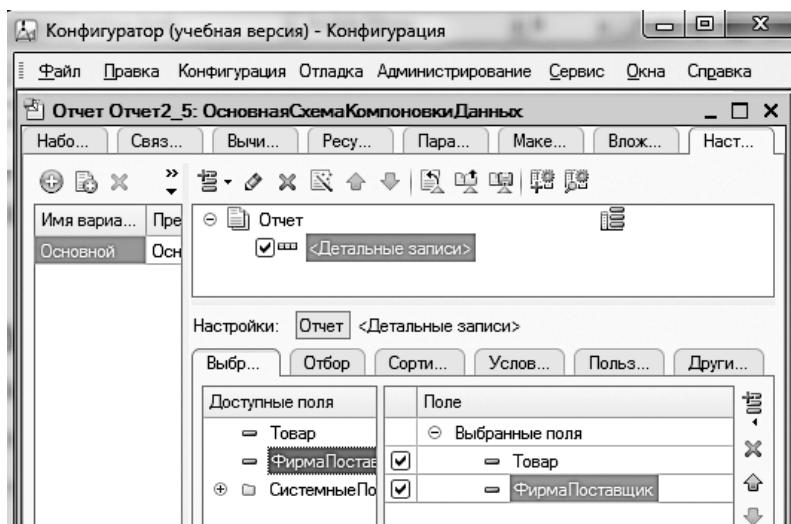


Рис. 2.16. Настройки в схеме компоновки данных

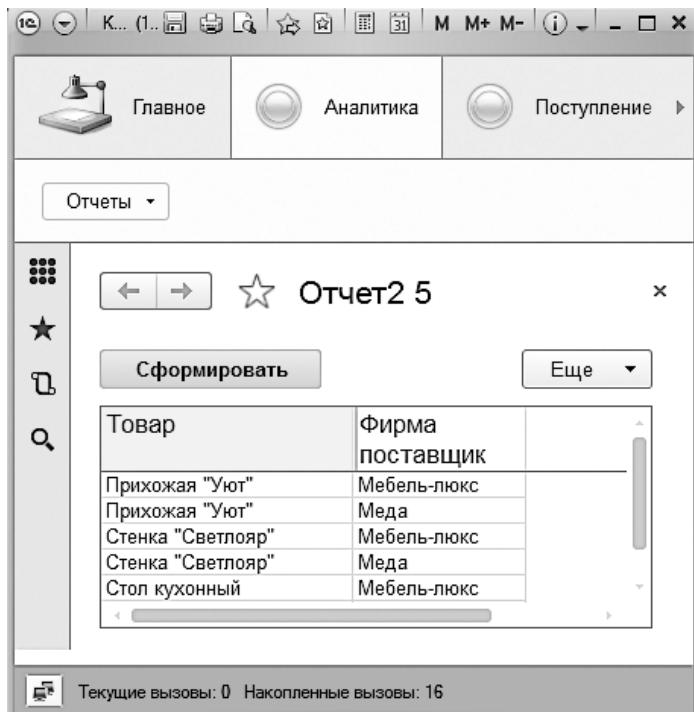


Рис. 2.17. Отчет уникальных комбинаций поставщик-товар

Таким образом, в списке на форме отчета мы получили не более одной строки с определенной комбинацией названий фирмы и товара.

Логические операторы

В предыдущих примерах нам еще не встречалось ключевое слово языка запросов ГДЕ. Оно позволяет задать условие отбора данных из исходных таблиц — источников запроса. В результате в запросе оказываются отобранными только те записи, для которых выполняется заданное условие.

В конструкции ГДЕ можно использовать логические операторы (И, ИЛИ, НЕ), которые позволяют формировать достаточно сложные логические выражения.

Применим эту возможность к задаче получения списка поступлений товаров, которые встречались в документах Поступление Товаров в количестве, большем пяти единиц, при условии их суммарной стоимости (произведение цены на количество) более 75 000 рублей. Текст запроса, в котором происходит отбор при одновременном выполнении этих двух условий, приведен в листинге 2.9.

Листинг 2.9. Запрос с включением двух условий и ключевого слова И

```
ВЫБРАТЬ Товар, Количество, Сумма,
Перечень.Ссылка.Филиал КАК Филиал
ИЗ Документ.ПоступлениеТоваров.ПереченьТоваров КАК Перечень
ГДЕ Количество > 5 И Сумма > 75000
```

Для реализации этого запроса создадим новый отчет (*Отчет2_6*) и выполним традиционные начальные действия. На рис. 2.18 показано окно схемы компоновки данных с необходимыми настройками.

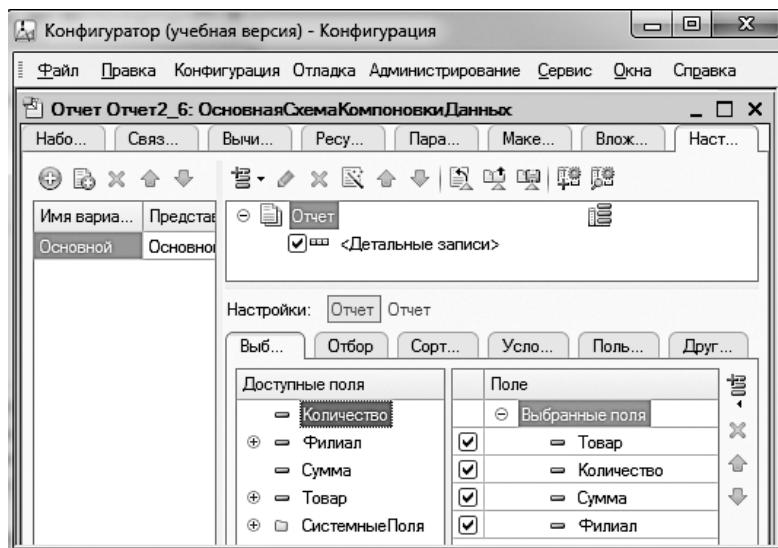


Рис. 2.18. Окно схемы компоновки данных

Результат заполнения отчета в системе 1С:Предприятие приведен на рис. 2.19. Как видно, в список отобранных данных попали только те товары, для которых были выполнены оба указанных условия.

Приведем теперь пример запроса, в котором комбинируются конструкции ИЛИ и И. Скажем, нас интересуют все товары, которые поступали либо по цене не менее 30 000 рублей в любом количестве, либо более 25 000 рублей в количестве, большем двух. В листинге 2.10 приведен запрос, реализующий необходимую выборку информации в этом случае, а на рис. 2.20 представлено окно схемы компоновки данных.

Листинг 2.10. Запрос с ключевыми словами И и ИЛИ

```
ВЫБРАТЬ Товар, Количество, Цена,
Перечень.Ссылка.Дата КАК ДатаДокумента
ИЗ Документ.ПоступлениеТоваров.ПереченьТоваров КАК Перечень
ГДЕ ( Количество > 2 И Цена > 25000 ) ИЛИ Цена >= 30000
```

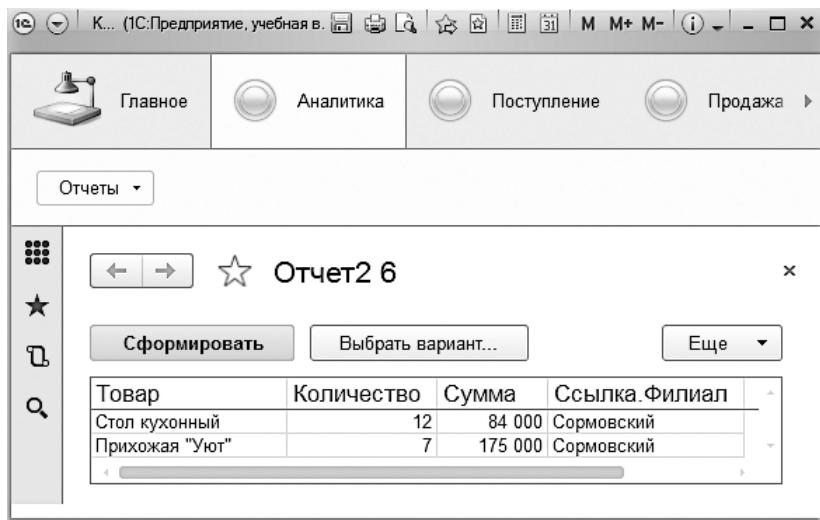


Рис. 2.19. Результат отбора товаров для запроса по двум условиям

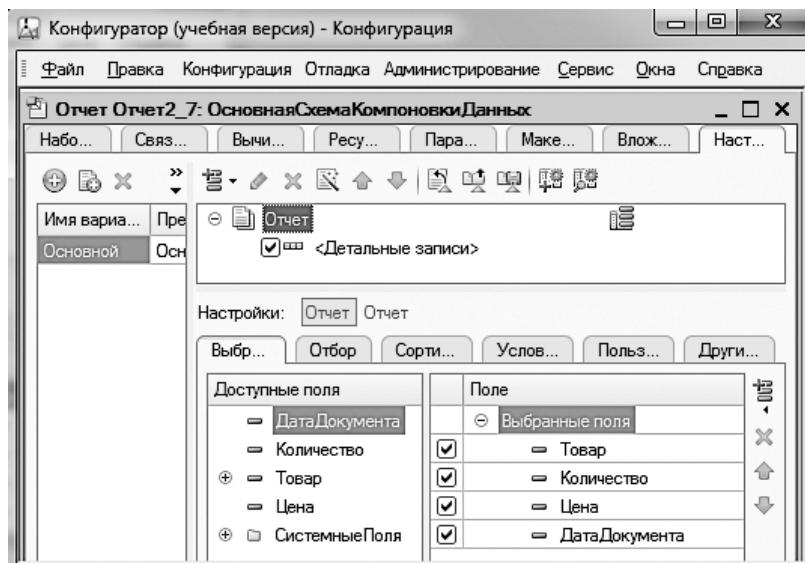


Рис. 2.20. Окно схемы компоновки данных

Результат заполнения отчета в режиме 1С:Предприятие показан на рис. 2.21.

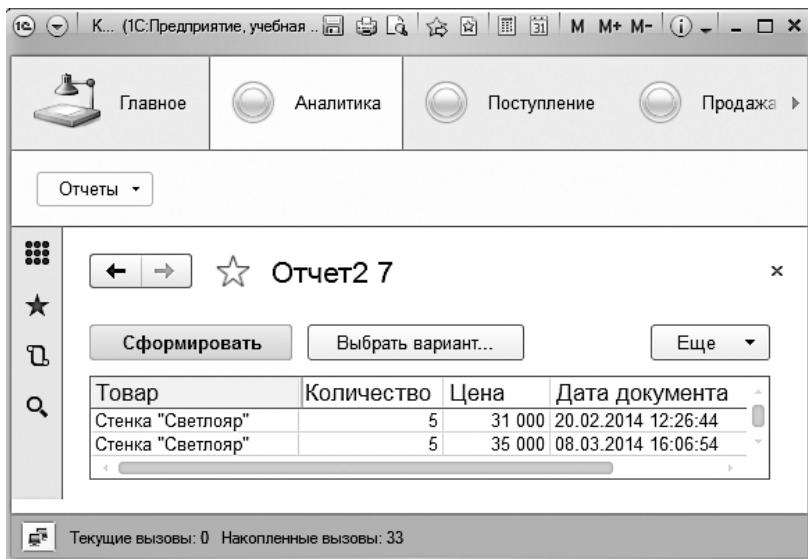


Рис. 2.21. Результат выполнения запроса, приведенного в листинге 2.10

Агрегатные функции в запросах

К настоящему моменту мы на несложных примерах познакомились с технологией выборки информации из базы данных. Однако часто на практике требуется также получать разнообразную сводную информацию. В частности, по информации из документов Поступления Товаров управленческому персоналу могут быть полезны ответы на следующие вопросы:

- Какова сумма поступлений в разрезе товаров?
- Каково среднее количество поступлений каждого товара?
- Какова наибольшая сумма поступления по каждому товару?

Для реализации подобных запросов следует воспользоваться агрегатными функциями и группировками.

ПРИМЕЧАНИЕ

В язык запросов включены агрегатные функции, которые используются при группировке результатов запросов, а также при подсчете итогов.

Любая агрегатная функция принимает в качестве аргумента какой-либо столбец табличных данных, а возвращает единственное значение. Агрегатные функции пред-

назначены для обобщения значений указанного столбца. В табл. 2.1 перечислены агрегатные функции, доступные в системе 1С:Предприятие 8.3.

Таблица 2.1. Перечень агрегатных функций

Название	Комментарий
СУММА	Вычисляет сумму всех значений, содержащихся в указанном столбце. В качестве параметра функции можно указывать только поле, содержащее числовое значение
МАКСИМУМ	Находит наибольшее значение в указанном столбце
МИНИМУМ	Находит наименьшее значение в указанном столбце
СРЕДНЕЕ	Вычисляет среднее арифметическое значение по указанному столбцу
КОЛИЧЕСТВО	Подсчитывает количество значений, содержащихся в указанном столбце

Что касается функции КОЛИЧЕСТВО, то следует отметить три способа ее применения — функция позволяет подсчитать:

- количество значений указанного поля, которые не равны NULL;
- количество различных значений указанного поля, не равных NULL (для этого необходимо указать ключевое слово РАЗЛИЧНЫЕ);
- количество строк в результате запроса (для этого в качестве параметра функции надо указать символ *).

Продемонстрируем теперь использование рассмотренных функций. В дальнейших примерах мы задействуем конструкцию языка запросов СГРУППИРОВАТЬ по. С ее помощью можно не просто выбирать записи из таблицы, а еще и группировать их определенным образом.

ПРИМЕЧАНИЕ

Под словом группировать подразумевается компоновка информации по полям группировки и вычисление агрегатных функций по каждой группе.

В листинге 2.11 представлен текст запроса, который демонстрирует применение описанных в табл. 2.1 функций применительно к документам Поступление Товаров, а окно схемы компоновки данных очередного отчета приведено на рис. 2.22.

Листинг 2.11. Запрос, демонстрирующий использование агрегатных функций

```

ВЫБРАТЬ Товар,
СУММА(Сумма) КАК СуммаПоТовару,
МАКСИМУМ(Сумма) КАК МаксимальнаяСумма,
МИНИМУМ(Сумма) КАК МинимальнаяСумма,
СРЕДНЕЕ(Сумма) КАК СредняяСумма
ИЗ Документ.ПоступлениеТоваров.ПереченьТоваров
СГРУППИРОВАТЬ ПО Товар

```

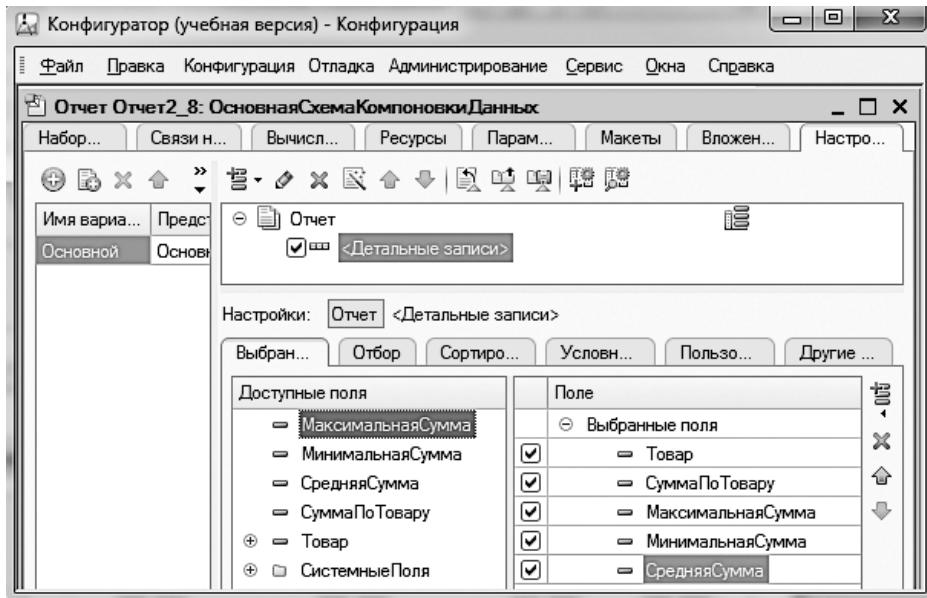


Рис. 2.22. Окно схемы компоновки данных для запроса, приведенного в листинге 2.11

В режиме 1С:Предприятие результат выполнения рассматриваемого запроса показан на рис. 2.23.

Для иллюстрации приемов применения функции КОЛИЧЕСТВО приведем запрос, который позволяет подсчитать число всех документов ПоступлениеТоваров в базе данных, а также число фирм, от которых были поступления товаров. Вариант текста запроса для решения данной задачи приведен в листинге 2.12. Окно схемы компоновки данных и результат выполнения запроса в режиме 1С:Предприятие представлены соответственно на рис. 2.24 и 2.25.

Листинг 2.12. Запрос, демонстрирующий использование функции КОЛИЧЕСТВО

```

ВЫБРАТЬ КОЛИЧЕСТВО(*) КАК Всего,
КОЛИЧЕСТВО (РАЗЛИЧНЫЕ Поступление.Фирма) КАК ЧислоFирм
ИЗ Документ.ПоступлениеТоваров КАК Поступление

```

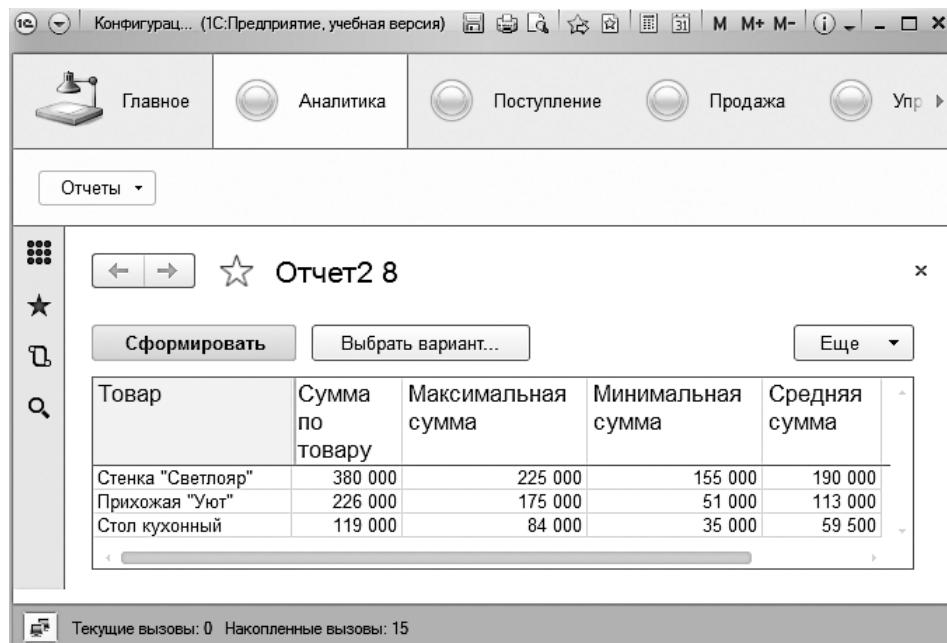


Рис. 2.23. Отчет, полученный с использованием агрегатных функций

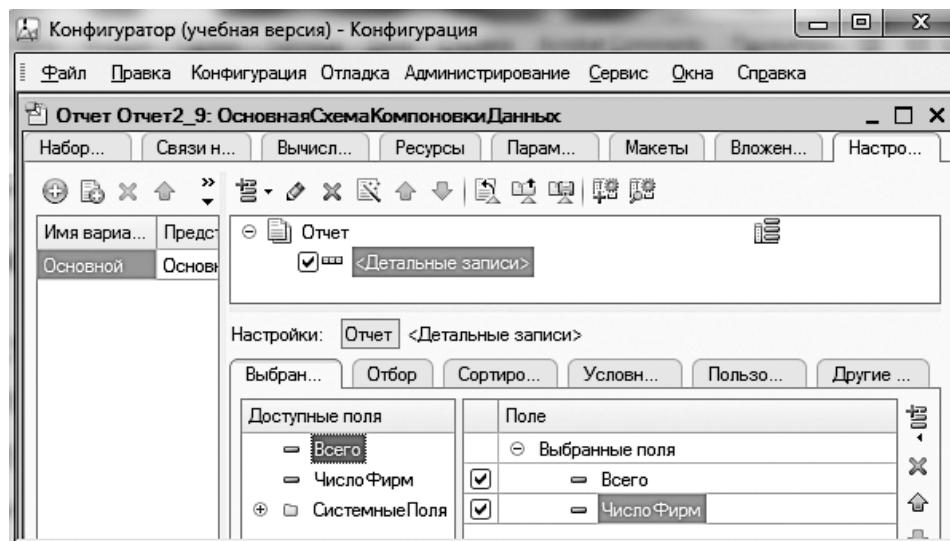


Рис. 2.24. Окно схемы компоновки данных для запроса, приведенного на листинге 2.12

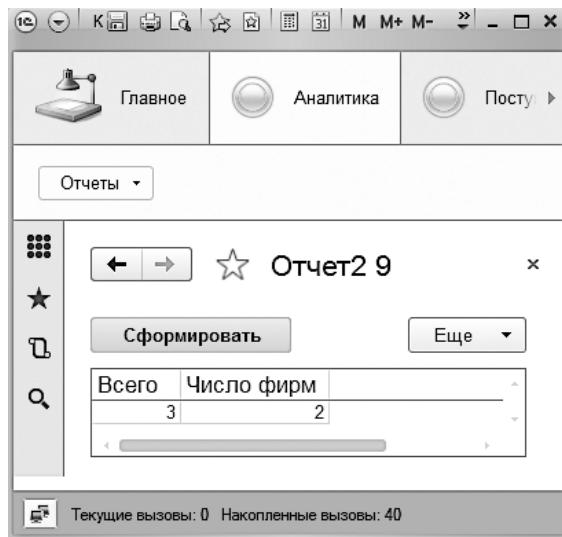


Рис. 2.25. Отчет, демонстрирующий использование функции КОЛИЧЕСТВО

Параметрические запросы

Рассмотренные к этому моменту примеры запросов не требовали установки каких-либо параметров, а какие-либо условия отбора, если таковые были необходимы, формировались непосредственно в тексте запроса. Понятно, что так выглядят не все запросы. Если посмотреть на любой из профессиональных отчетов, то мы увидим, что в запросах имеются параметры, которые пользователь может указывать в диалоге.

Допустим нам необходимо получить с помощью запроса информацию о документах по поступлениям товаров в определенный филиал не позднее указанной даты. Текст запроса для решения данной задачи приведен в листинге 2.13. В запросе мы использовали автоматически формируемые в любом документе реквизиты Номер и Дата. В текст запроса включен также символ &, обозначающий параметр (в данном случае параметров два — для определения даты и названия филиала).

Листинг 2.13. Параметрический запрос для извлечения дат и номеров документов

```
ВЫБРАТЬ Дата, Номер
ИЗ Документ.ПоступлениеТоваров
ГДЕ Дата <= &ГраницаДаты И Филиал=&УказанныйФилиал
```

Здесь технические действия для выполнения запроса и получения результата несколько сложнее, чем в предыдущих примерах, поэтому опишем необходимые шаги более подробно. Создание нового отчета (*Отчет2_10*) начнем традиционно с вкладки Основные окна редактирования объекта конфигурации.

Следующий шаг заключается в переходе к созданию схемы компоновки. На рис. 2.26 представлена вкладка Наборы данных окна конструктора схемы компоновки данных.

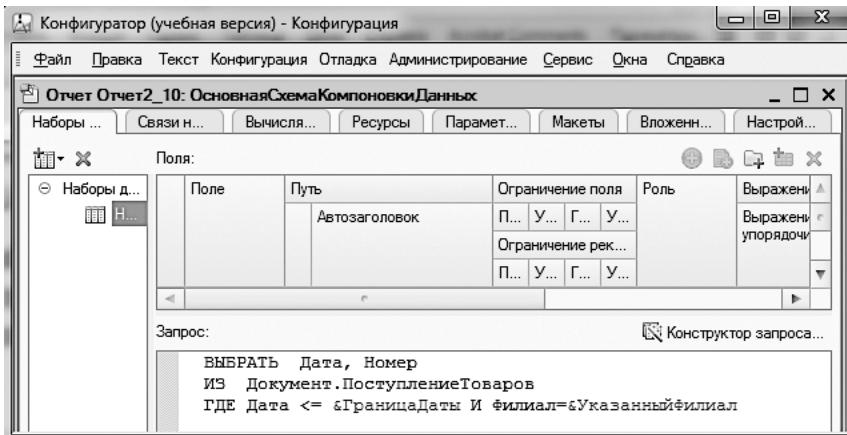


Рис. 2.26. Окно конструктора схемы компоновки данных

После этого перейдем на вкладку Параметры, где задаются условия отбора данных для отчета (рис. 2.27). Здесь мы видим, что система в соответствии с текстом запроса автоматически разместила два параметра:

- УказанныйФилиал;
- ГраницаДаты.

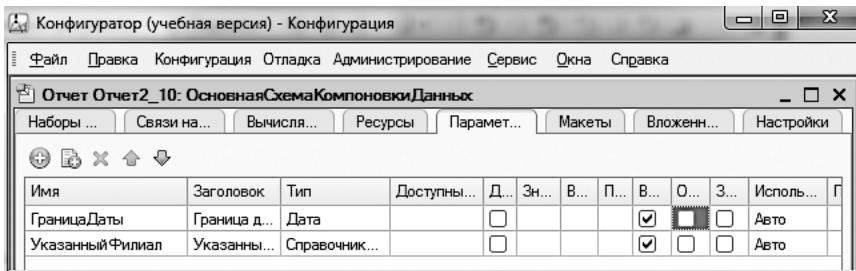


Рис. 2.27. Вкладка Параметры окна конструктора схемы компоновки данных

Обратим внимание на флажок **Ограничение доступности**. Если этот флажок не устанавливать, то рассматриваемый параметр будет доступен для пользователя в форме отчета, а если установить, то в режиме 1С:Предприятие в форме отчета параметр виден не будет.

После выполнения установок перейдем на вкладку **Настройки** и определим поля для отчета с именем **Отчет2_10** (рис. 2.28).

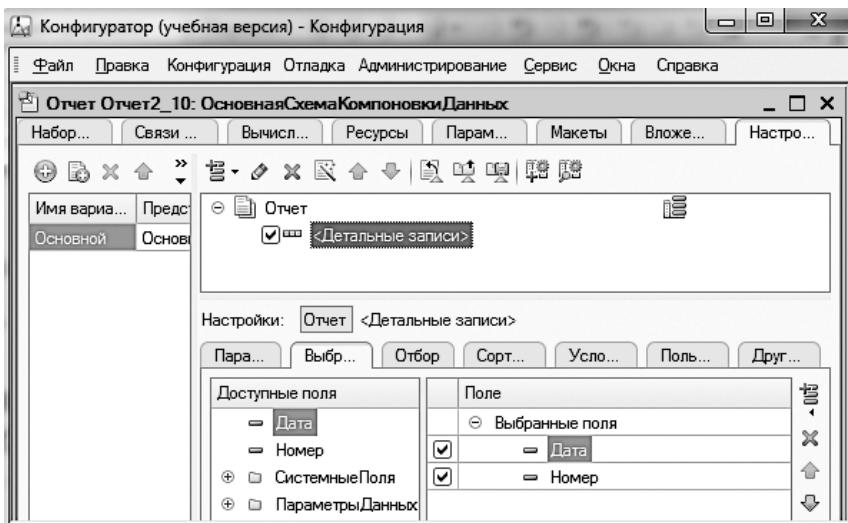


Рис. 2.28. Установки на вкладке Настройки конструктора схемы компоновки данных

Далее наша задача заключается в том, чтобы предоставить пользователю возможность устанавливать параметры *УказанныйФилиал* и *ГраницаДаты*. Для этого надо перейти на дополнительную вкладку *Параметры* (рис. 2.29), выделить параметр *Граница даты*, щелкнуть на кнопке *Свойства элемента пользовательских настроек* (самый нижний значок в вертикальном ряду значков) и в открывшемся окне установить флагок *Включать в пользовательские настройки* (рис. 2.30).

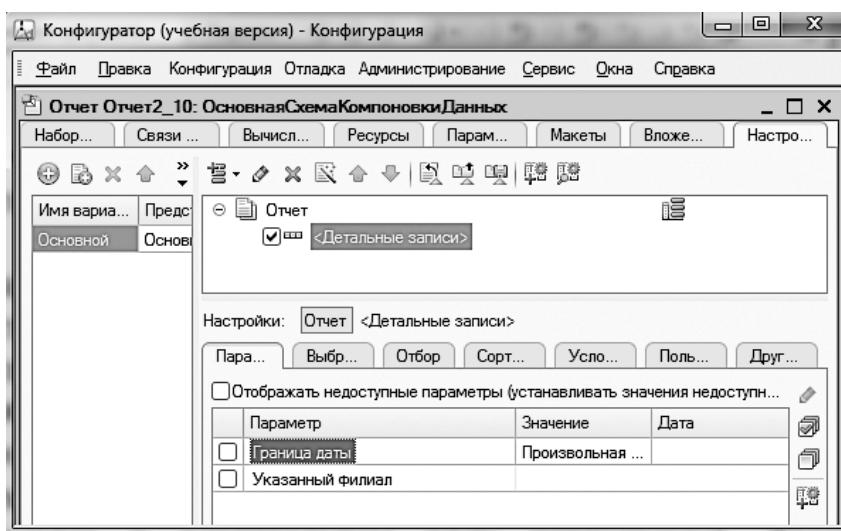


Рис. 2.29. Дополнительная вкладка Параметры в окне конструктора схемы компоновки данных

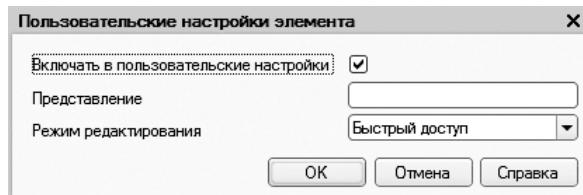


Рис. 2.30. Окно для включения параметра в пользовательские настройки

Аналогичные настройки выполним для параметра УказанныйФилиал. На этом необходимые действия в конфигураторе завершаются, теперь можно перейти в режим 1С:Предприятие и сформировать отчет (рис. 2.31). Обратите внимание на то, что перед формированием отчета необходимо установить параметры в его электронной форме.

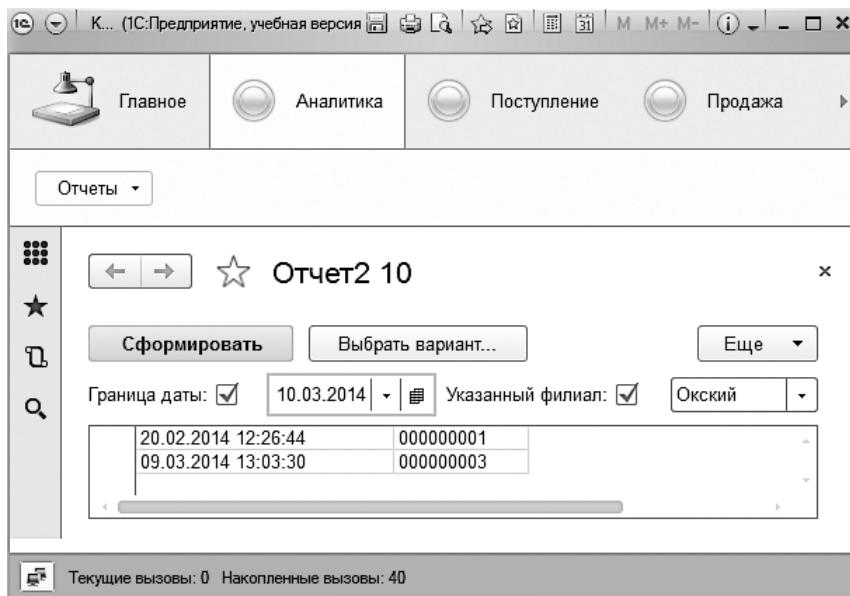


Рис. 2.31. Отчет 2_10, демонстрирующий выполнение запроса из листинга 2.13

Рассмотрим еще один пример на тему использования параметров в запросе. Будем считать, что наша задача заключается в том, чтобы прочитать движения по регистру накопления КоличествоТоваров, используя ссылку на документ. Текст запроса, который позволяет извлечь название товара, филиала, а также количество единиц поступления (либо продажи или перемещения) товара, представлен в листинге 2.14. В качестве параметра запроса выступает документ, который привел к формированию записей в регистре накопления КоличествоТоваров.

Листинг 2.14. Запрос для извлечения информации из регистра КоличествоТоваров

```
ВЫБРАТЬ Товар,Филиал,Количество
ИЗ РегистрНакопления.КоличествоТоваров КАК КоличествоТоваров
ГДЕ КоличествоТоваров.Регистратор=&Регистратор
```

Технические действия для выполнения запроса и получения результата аналогичны предыдущему примеру, поэтому некоторые уже знакомые шаги мы при комментировании опустим.

Начнем создание нового отчета и на вкладке **Основные** введем название отчета **Отчет2_11**. В поле запроса в окне схемы компоновки данных разместим текст, приведенный в листинге 2.14. После этого перейдем на вкладку **Параметры**, где определяются необходимые параметры отчета (рис. 2.32). Видно, что система компоновки данных в соответствии с текстом запроса автоматически включила параметр **Регистратор**. Обратим внимание на то, что требуетсябросить флагок **Ограничение доступности**, чтобы пользователь получил возможность установки данного параметра непосредственно в форме отчета.

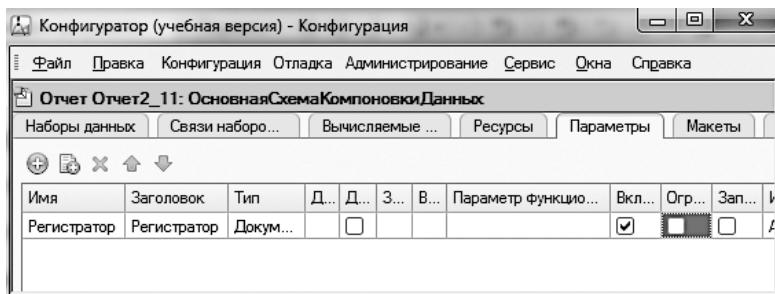


Рис. 2.32. Вкладка Параметры окна конструктора схемы компоновки данных

Перейдем далее на вкладку **Настройки** и определим поля для формируемого отчета (рис. 2.33). Для предоставления пользователю возможности устанавливать параметры в форме отчета необходимо выполнить настройки на уже знакомой дополнительной вкладке **Параметры** (рис. 2.34). В данном случае для параметра **Регистратор** следует установить флагок **Включать в пользовательские настройки**. После этого можно перейти в режим 1С:Предприятие и получить необходимый отчет (рис. 2.35).

Рассмотрим еще один пример, связанный с документом **ПоступлениеТоваров**. В табличной части этого документа присутствует реквизит **Цена**. Возможна ситуация, когда необходимо отобрать документы, в которых значение в этом поле оказалось меньше того, которое пользователь указал в форме отчета. Если документов много, то визуальный просмотр для отбора необходимых документов не очень эффективен. Для решения данной задачи оформим текст запроса в соответствии с листингом 2.15.

Листинг 2.15. Запрос для отбора документов по условию в поле Цена

```
ВЫБРАТЬ РАЗЛИЧНЫЕ ПереченьТоваров.Ссылка
КАК ОтобранныйДокумент
ИЗ Документ.ПоступлениеТоваров.ПереченьТоваров КАК ПереченьТоваров
ГДЕ ПереченьТоваров.Цена < &ГраницаЦены
```

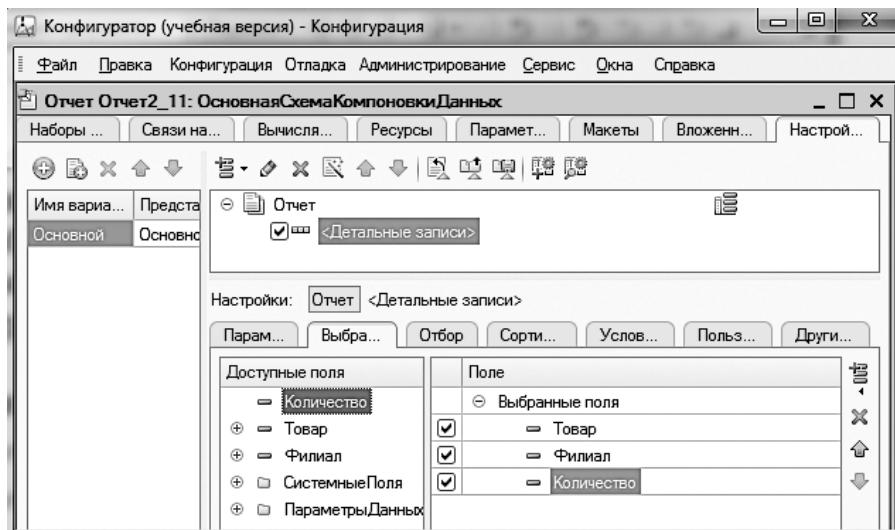


Рис. 2.33. Выбор полей для схемы компоновки данных

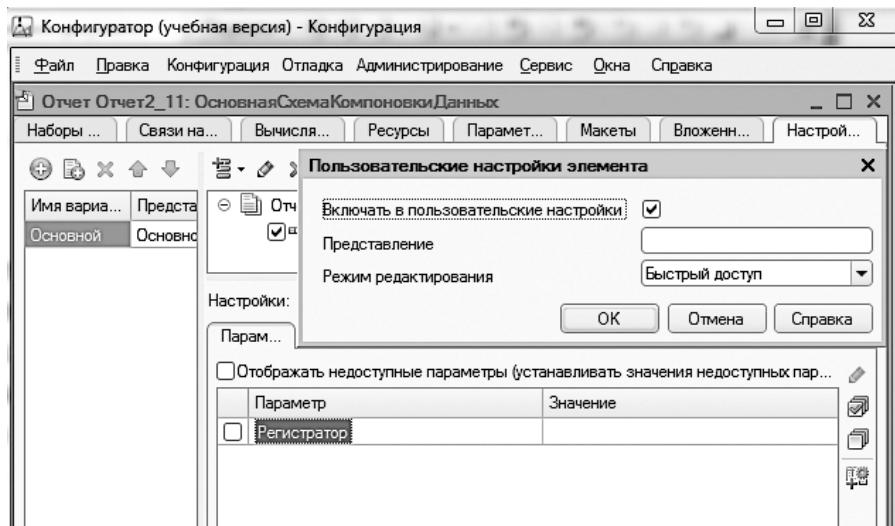


Рис. 2.34. Включение параметра Регистратор в пользовательские настройки

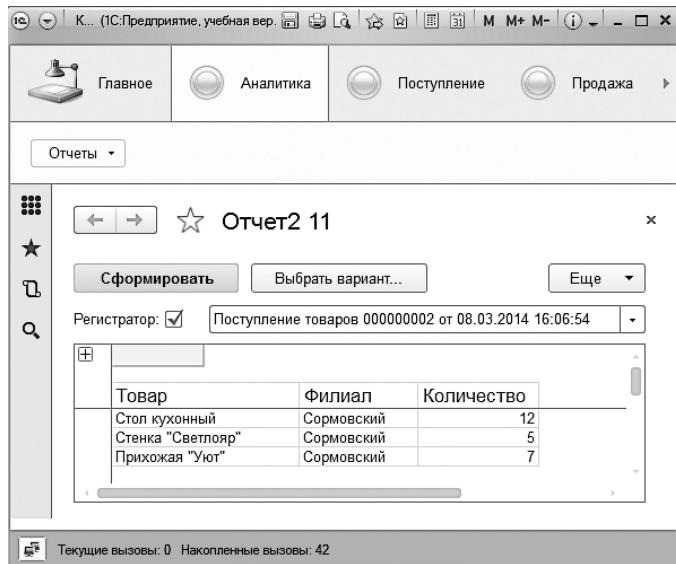


Рис. 2.35. Отчет 2_11 в режиме 1С:Предприятие

В данном запросе производится выборка неповторяющихся записей (для этого используется ключевое слово **РАЗЛИЧНЫЕ**). В этом случае при наличии в одном документе нескольких строк с ценой, меньшей, чем указанная пользователем, в результат запроса найденный документ попадает только один раз (рис. 2.36).

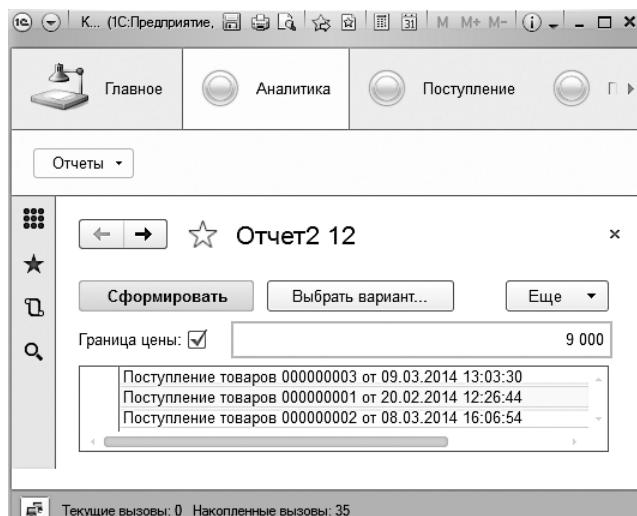


Рис. 2.36. Отбор документов по поступлению товаров с ценой меньше указанной

Ключевое слово МЕЖДУ

В языке запросов имеется вспомогательное ключевое слово **МЕЖДУ**, предназначенное для задания интервалов. Например, запрос, приведенный в листинге 2.16, перед выполнением требует установки значений параметров **Дата1** и **Дата2**. При этом в результат запроса попадут только те записи регистра накопления **КоличествоТоваров**, которые соответствуют выбранному интервалу по дате.

Листинг 2.16. Запрос с ключевым словом МЕЖДУ

```
ВЫБРАТЬ Товар,
      Филиал, Количество, Регистратор
ИЗ РегистрНакопления.КоличествоТоваров
ГДЕ РегистрНакопления.КоличествоТоваров.Период
      МЕЖДУ &дата1 И &дата2
```

Заметим, что в конструкции рассматриваемого запроса используется параметр **Период**, который отражает дату и время внесения информации. На рис. 2.37 представлен результат выполнения данного запроса.

The screenshot shows the 1C Configuration interface for '1C:Enterprise, учебная версия'. The main window displays a report titled 'Отчет2 13' (Report 2 13). The report parameters are set to 'Сформировать' (Generate) and 'Выбрать вариант...' (Select variant...). The date range is specified as 'Дата1: 05.03.2014 0:00:00' and 'Дата2: 09.03.2014 0:00:00'. The report results show a table of goods received into the 'Sormovskiy' branch. The table includes columns: Товар (Goods), Филиал (Branch), Количество (Quantity), and Регистратор (Registrar). The data shows three entries: 'Стол кухонный' (Kitchen table) received 12 times, 'Стенка "Светлояр"' (Lighting cabinet) received 5 times, and 'Прихожая "Ют"' (Entryway "Yut") received 7 times. The bottom status bar indicates 'Текущие вызовы: 0 Накопленные вызовы: 18' (Current calls: 0 Accumulated calls: 18).

Товар	Филиал	Количество	Регистратор
Стол кухонный	Сормовский	12	Поступление товаров 000000002 от I
Стенка "Светлояр"	Сормовский	5	Поступление товаров 000000002 от I
Прихожая "Ют"	Сормовский	7	Поступление товаров 000000002 от I

Рис. 2.37. Результат выполнения запроса к регистру с использованием слова МЕЖДУ

Рассмотрим теперь пример реализации аналогичного запроса к документам ПоступлениеТоваров (листинг 2.17). В этом случае происходит перебор документов, имеющихся в информационной базе (рис. 2.38). Для запроса используется параметр **дата**, характеризующий дату и время создания документа.

Листинг 2.17. Запрос к документам с использованием ключевого слова МЕЖДУ

```
ВЫБРАТЬ Товар, Количество,
Сумма
ИЗ Документ.ПоступлениеТоваров.ПереченьТоваров
ГДЕ Ссылка.Дата МЕЖДУ &дата1 И &дата2
```

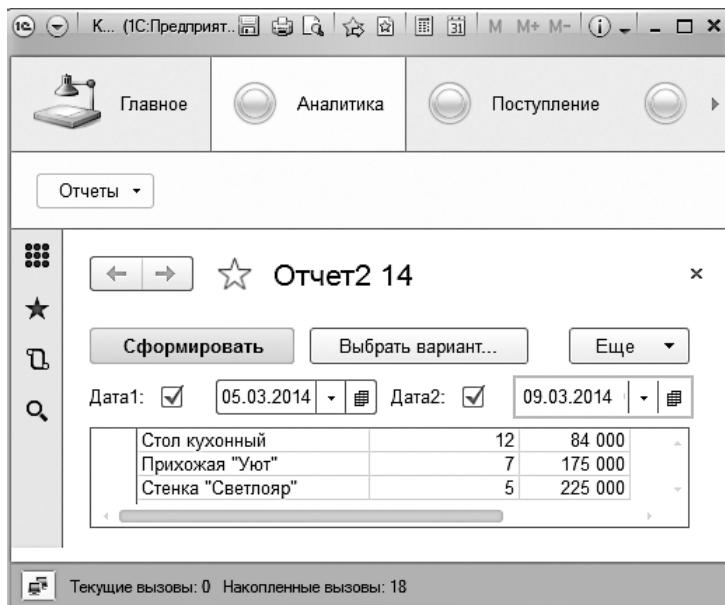


Рис. 2.38. Результат выполнения запроса к документам с использованием слова МЕЖДУ

Ключевое слово ИМЕЮЩИЕ

В языке запросов существует ключевое слово **ИМЕЮЩИЕ**, которое позволяет накладывать условия на значения агрегатных функций. При этом в условии отбора со словом **ИМЕЮЩИЕ** можно использовать агрегатные функции только для полей, по которым осуществляется группировка.

В листинге 2.18 приведен пример запроса к табличной части документа ПоступлениеТоваров, в котором используется конструкция **ИМЕЮЩИЕ**. Цель рассматриваемого запроса — отобрать товары, для которых итоговая сумма по поступлениям пре-

вышает 325 000. На рис. 2.39 показан результат указанного отбора по имеющейся информации в информационной базе.

Листинг 2.18. Запрос с ключевым словом ИМЕЮЩИЕ

```
ВЫБРАТЬ Товар, СУММА(Сумма) КАК СуммаТоваров
ИЗ Документ.ПоступлениеТоваров.ПереченьТоваров КАК Перечень
СГРУППИРОВАТЬ ПО Товар
ИМЕЮЩИЕ СУММА(Сумма) > 325000
```

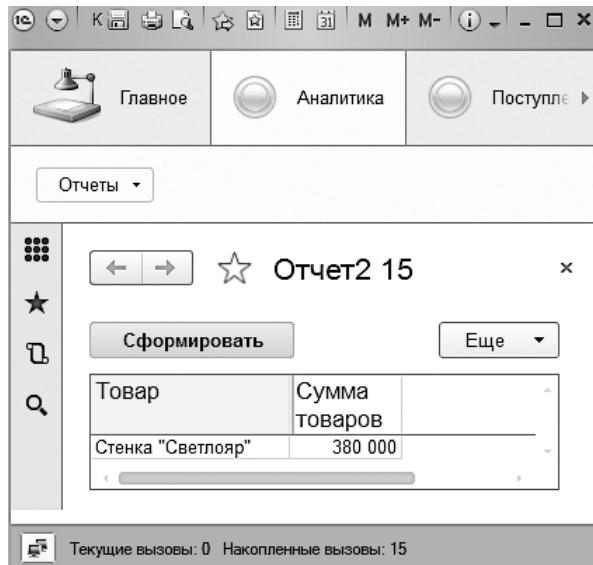


Рис. 2.39. Реализация запроса с использованием ключевого слова ИМЕЮЩИЕ

Объединение запросов

В языке запросов имеется возможность объединять результаты нескольких запросов; при этом из записей, полученных с помощью каждого из объединяемых запросов, получается общий результат. При *объединении* каждый запрос собирает данные независимо, а такие операции, как упорядочивание результатов и расчет итогов, выполняются уже над результатом объединения запросов.

Поля результата запроса будут называться так, как указано в списке полей выборки первого из объединяемых запросов. Поля выборки остальных запросов сопоставляются с полями результата в соответствии с порядком их следования в списке полей выборки. Объединяемые запросы должны иметь одинаковое количество полей в списке полей выборки.

Объединение запросов начинается с обязательного ключевого слова **ОБЪЕДИНИТЬ**, после которого следует описание присоединяемого запроса. Далее можно присоединить еще один запрос и т. д.

По умолчанию при объединении запросов полностью одинаковые строки в результате запроса (когда они формируются разными запросами) заменяются одной. Если требуется иное, необходимо указать ключевое слово **ВСЕ**.

Приведем теперь пример на тему объединения запросов. Допустим, необходимо составить список товаров в привязке к фирме, которая выступала в роли либо поставщика, либо продавца. В листинге 2.19 приведена реализация данного запроса. В результате в режиме 1С:Предприятие мы получим набор уникальных комбинаций товара в сочетании с фирмой (рис. 2.40).

Листинг 2.19. Запрос для формирования списка товаров с указанием фирмы

```

ВЫБРАТЬ
    Перечень.Ссылка.Фирма.Наименование КАК Фирма,
    Перечень.Товар
ИЗ
    Документ.ПоступлениеТоваров.ПереченьТоваров КАК Перечень
ОБЪЕДИНИТЬ
ВЫБРАТЬ
    Перечень.Ссылка.Фирма.Наименование,
    Перечень.Товар
ИЗ
    Документ.ПродажаТоваров.ПереченьТоваров КАК Перечень
УПОРЯДОЧИТЬ ПО
    Фирма

```

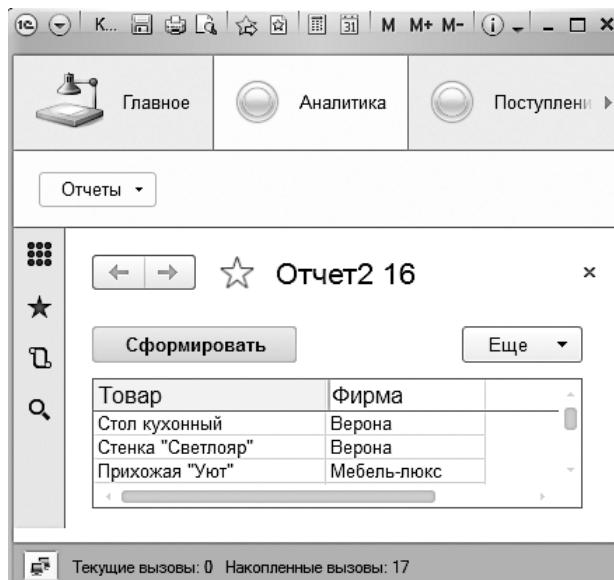


Рис. 2.40. Отбор названий товаров в сочетании с фирмой

Соединение таблиц при организации запроса

До настоящего времени практически во всех приводимых примерах запросов данные извлекались из одной таблицы. В предыдущем разделе мы рассмотрели объединение таблиц, однако этим ресурсы запросов не ограничиваются.

Важной особенностью языка запросов системы 1С:Предприятие является возможность обращения сразу к нескольким таблицам и соединения их определенным образом. Например, при работе может потребоваться выбрать названия всех товаров из справочника Товары; при этом для тех товаров, которые нами уже продавались, должно быть указано количество проданных единиц.

Внутреннее соединение

В конструкции подобного запроса используется ключевое слово СОЕДИНЕНИЕ, позволяющее отобрать из документов ПоступлениеТоваров и ПродажаТоваров только совпадающие товары (листинг 2.20). На рис. 2.41 приведен результат выполнения запроса к имеющимся в базе данным.

Листинг 2.20. Запрос для отбора совпадающих товаров

```
ВЫБРАТЬ РАЗЛИЧНЫЕ ПереченьПоступлений.Товар КАК ПоступившийТовар,
ПереченьПродаж.Товар КАК ПроданныйТовар
ИЗ Документ.ПоступлениеТоваров.ПереченьТоваров КАК ПереченьПоступлений
СОЕДИНЕНИЕ
Документ.ПродажаТоваров.ПереченьТоваров КАК ПереченьПродаж
ПО ПереченьПоступлений.Товар = ПереченьПродаж.Товар
```

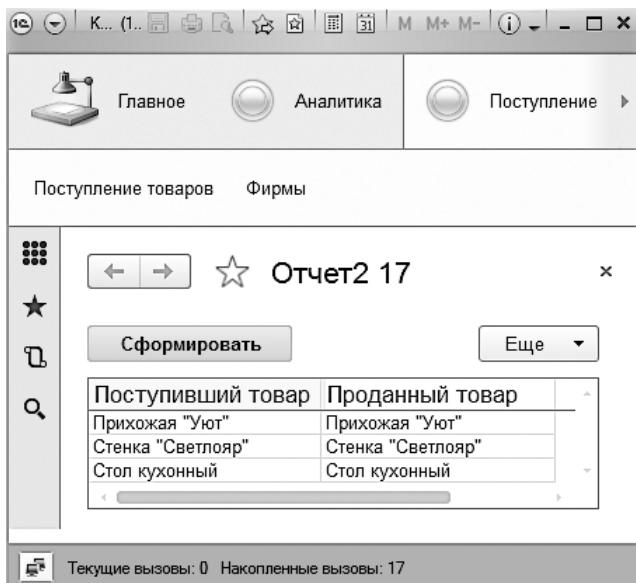


Рис. 2.41. Пример использования внутреннего соединения таблиц в запросе

Рассмотрим еще один пример на данную тему. Допустим, нам необходимо отобрать филиалы, в которые поступали товары от фирм-партнеров и одновременно были перемещения с выставок. Текст данного запроса приведен в листинге 2.21, а результат в режиме 1С:Предприятие представлен на рис. 2.42.

Листинг 2.21. Извлечение филиалов по условию поступления и перемещения товаров

```
ВЫБРАТЬ РАЗЛИЧНЫЕ Поступление.Филиал КАК НашФилиал
ИЗ Документ.ПоступлениеТоваров КАК Поступление
СОЕДИНЕНИЕ
    Документ.ПеремещениеСВыставки КАК Перемещение
ПО Поступление.Филиал=Перемещение.Филиал
```

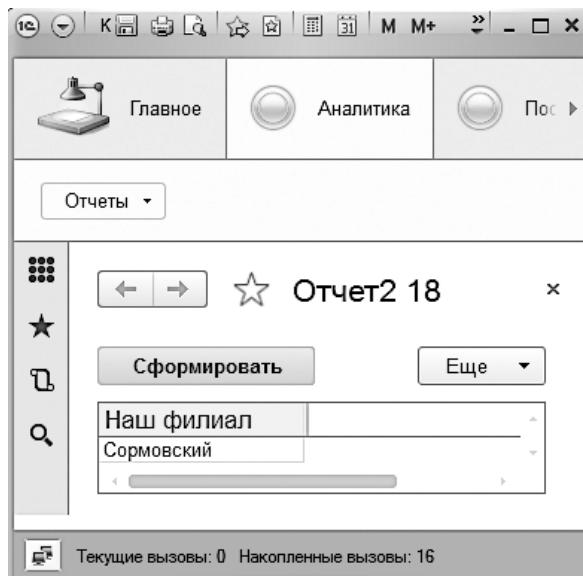


Рис. 2.42. Отбор филиалов, в которых были как поступления, так и перемещения с выставки

Левое внешнее соединение

Рассмотренные примеры соединений запросов относятся к классу *внутренних*, однако в языке запросов системы 1С:Предприятие 8.3 существует возможность еще и *внешних* соединений, которые могут быть левыми, правыми и полными.

Конструкция **ЛЕВОЕ [ВНЕШНЕЕ] СОЕДИНЕНИЕ** означает, что в результат запроса надо включить комбинации записей из обеих исходных таблиц, которые соответствуют указанному условию. Однако в отличие от внутреннего соединения в результат запроса включаются еще и записи из первого источника (указанного слева от слова **СОЕДИНЕНИЕ**), для которых не найдено соответствующих условия записей из второго источника.

Таким образом, в результат запроса включаются все записи из первого источника, которые соединяются с записями из второго источника при выполнении указанного условия. А строки результата запроса, для которых не найдено соответствующих условию записей из второго источника, будут содержать значение **NULL** в полях, формируемых на основании записей из этого источника.

ПРИМЕЧАНИЕ

Значения **NULL** являются специальными маркерами, обозначающими неуказанные (отсутствующие) значения или значения, не имеющие смысла.

Например, пусть необходимо выбрать фамилии всех менеджеров с указанием общей суммы выполненных ими продаж, однако при этом в результирующую таблицу должны также попасть фамилии менеджеров, которые еще не оформляли документов по продаже товаров. Данный запрос реализуется с помощью конструкции **ЛЕВОЕ [ВНЕШНЕЕ] СОЕДИНЕНИЕ** (листинг 2.22). Результат отбора показывает, что в этом случае (в отличие от внутреннего соединения) в полученном списке данных представлены также менеджеры, у которых еще не зарегистрировано продаж (рис. 2.43).

Листинг 2.22. Запрос для извлечения информации о менеджерах

```
ВЫБРАТЬ НашиМенеджеры.Наименование КАК НашМенеджер,
       Продажи.СуммаОборот КАК Сумма
  ИЗ Справочник.Менеджеры КАК НашиМенеджеры
ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.ПродажиПоФилиалам.Обороты КАК Продажи
ПО НашиМенеджеры.Наименование=Продажи.Менеджер.Наименование
```

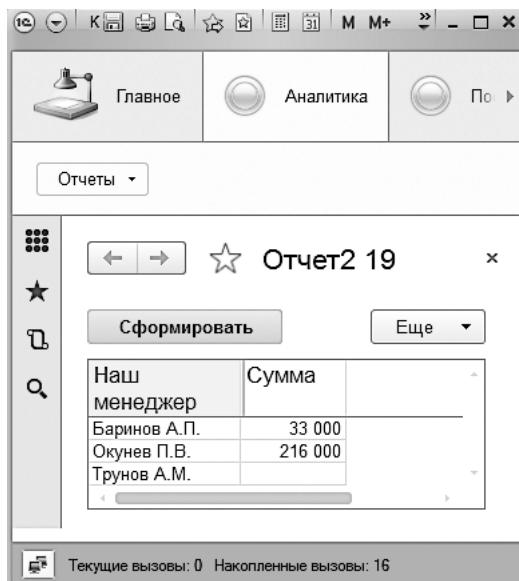


Рис. 2.43. Левое внешнее соединение таблиц в запросе при отборе менеджеров

Правое внешнее соединение

Конструкция ПРАВОЕ [ВНЕШНЕЕ] СОЕДИНЕНИЕ означает, что в результат запроса надо включить комбинации записей из обеих исходных таблиц, которые соответствуют указанному условию. Кроме того, в результат запроса надо включить еще и записи из второго источника (указанного справа от слова СОЕДИНЕНИЕ), для которых не найдено соответствующих условию записей из первого источника.

Таким образом, в результат запроса включаются все записи из второго источника, которые соединяются с записями из первого источника при выполнении указанного условия. А строки результата запроса, для которых не найдено соответствующих условию записей из первого источника, будут содержать значение NULL в полях, формируемых на основании записей из этого источника.

Правое внешнее соединение полностью аналогично левому за исключением того, что таблицы меняются местами. В качестве примера приведем формирование списка менеджеров, аналогичного рассмотренному в предыдущем случае, но полученного с использованием правого соединения таблиц. Дополнительно следует вывести информацию о филиалах, в которых были продажи. Текст запроса приведен в листинге 2.23, а результата отбора информации показан на рис. 2.44.

Листинг 2.23. Запрос с правым соединением

```
ВЫБРАТЬ  НашиМенеджеры.Наименование КАК НашМенеджер,
         Продажи.Филиал КАК НашФилиал,
         Продажи.СуммаОборот КАК Сумма
ИЗ РегистрНакопления.ПродажиПоФилиалам.Обороты КАК Продажи
ПРАВОЕ СОЕДИНЕНИЕ
    Справочник.Менеджеры КАК НашиМенеджеры
ПО НашиМенеджеры.Наименование = Продажи.Менеджер.Наименование
```

Рассмотрим еще одну ситуацию на данную тему. Будем считать, что необходимо составить общий список товаров с учетом их количества (просуммировать количество одинаковых товаров во всех филиалах). Для этого можно использовать информацию из виртуальной таблицы остатков регистра накопления:

РегистрНакопления.КоличествоТоваров.Остатки

Однако по замыслу задачи в указанный список необходимо включить также товары, по которым еще не было поступлений (это новые товары, зарегистрированные только в справочнике Товары). Фактически о подобных товарах в регистре КоличествоТоваров записей нет. Поэтому для извлечения информации мы воспользуемся еще одной таблицей — справочником Товары. В листинге 2.24 показан текст запроса, в котором используется правое соединение таблиц, а результат его выполнения в режиме 1С:Предприятие продемонстрирован на рис. 2.45. Видно, что в сформированный список вошли как товары, которые к нам уже поступали, так и товары, которые в документах по поступлению не фигурировали.

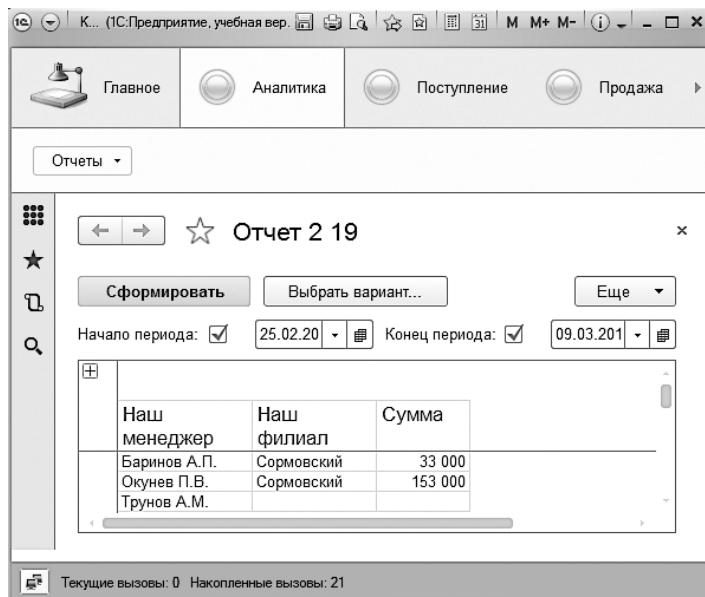


Рис. 2.44. Пример использования правого внешнего соединения таблиц в запросе

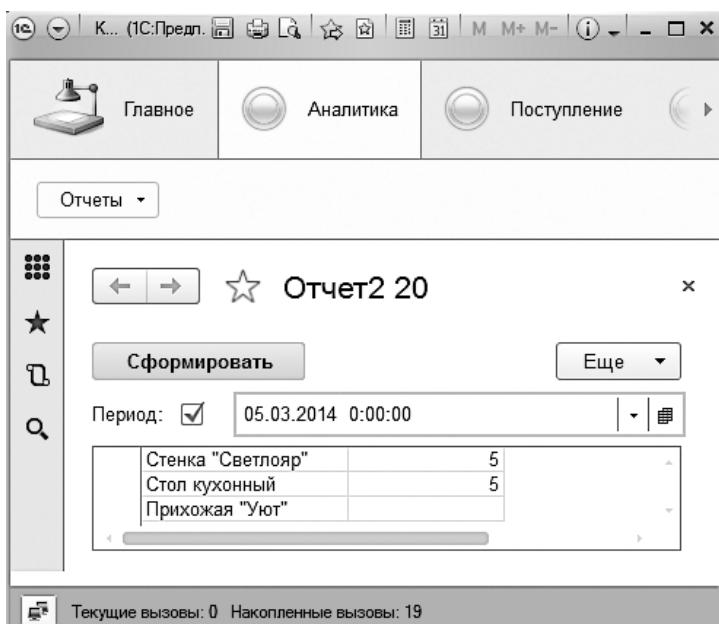


Рис. 2.45. Формирование списка товаров с использованием правого внешнего соединения

Листинг 2.24. Запрос для извлечения информации о наличии товаров

```
ВЫБРАТЬ Товары.Ссылка КАК НазваниеТовара,
    КоличествоТоваров.КоличествоОстаток КАК Количество
ИЗ РегистрНакопления.КоличествоТоваров.Остатки КАК КоличествоТоваров
ПРАВОЕ СОЕДИНЕНИЕ
    Справочник.Товары КАК Товары
ПО Товары.Ссылка=КоличествоТоваров.Товар.Ссылка
УПОРЯДОЧИТЬ ПО Количество УБЫВ
```

Полное внешнее соединение

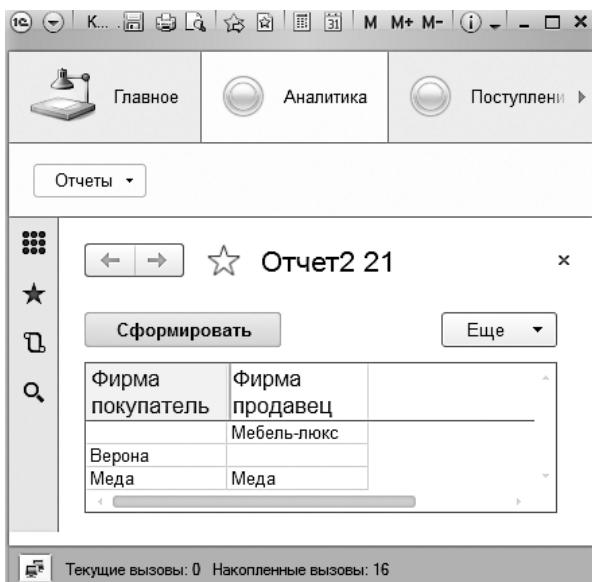


Рис. 2.46. Пример отчета, полученного с использованием полного внешнего соединения таблиц в запросе

лей. При этом необходимо в результа включить также фирмы, которые выступали только в одном качестве (в качестве только продавца или только покупателя). Текст запроса, который необходимо разработать для извлечения данной информации, представлен в листинге 2.25, а результат его выполнения в режиме 1С:Предприятие продемонстрирован на рис. 2.46. Видно, что в список попали все фирмы, по которым было движение товара.

Листинг 2.25. Запрос с использованием полного внешнего соединения таблиц

```
ВЫБРАТЬ РАЗЛИЧНЫЕ Продажа.Фирма.Наименование КАК ФирмаПокупатель,
    Поступление.Фирма.Наименование КАК ФирмаПродавец
ИЗ Документ.ПродажаТоваров КАК Продажа
ПОЛНОЕ СОЕДИНЕНИЕ
    Документ.ПоступлениеТоваров КАК Поступление
ПО Продажа.Фирма.Наименование=Поступление.Фирма.Наименование
```

Конструкция ПОЛНОЕ [ВНЕШНЕЕ] СОЕДИНЕНИЕ означает, что в результат запроса надо включить комбинации записей из обеих исходных таблиц, которые соответствуют указанному условию. Кроме того, в результат запроса включаются те записи из обоих источников, для которых не найдено соответствий. Таким образом, в результат запроса из обоих источников попадают все записи, и они соединяются друг с другом при выполнении указанного условия. Сформулируем задачу, которую нам необходимо решить. Будем считать, что необходимо отобрать фирмы, которые выступали в качестве как продавцов, так и покупателей.

Функция ПРЕДСТАВЛЕНИЕ

Функция ПРЕДСТАВЛЕНИЕ предназначена для получения строкового представления значения произвольного типа. В качестве параметра функции может выступать выражение любого типа. Например, можно привести к строковому типу информацию о датах документов. На эту тему рассмотрим пример с выводом дат из документов ПоступлениеТоваров и ПеремещениеСВыставки.

Текст запроса для извлечения данной информации приведен в листинге 2.26. В рассматриваемой процедуре производится объединение двух запросов, которые выполняют извлечение дат документов ПоступлениеТоваров и извлечение дат документов ПеремещениеСВыставки.

Листинг 2.26. Запрос с функцией ПРЕДСТАВЛЕНИЕ

```
ВЫБРАТЬ ПРЕДСТАВЛЕНИЕ(Поступление.Дата) КАК ДатыПоступленияПеремещения
    ИЗ Документ.ПоступлениеТоваров КАК Поступление
ОБЪЕДИНИТЬ
    ВЫБРАТЬ ПРЕДСТАВЛЕНИЕ(Перемещение.Дата)
        ИЗ Документ.ПеремещениеСВыставки КАК Перемещение
```

Результат выполнения рассматриваемого запроса для отбора дат документов показан на рис. 2.47.

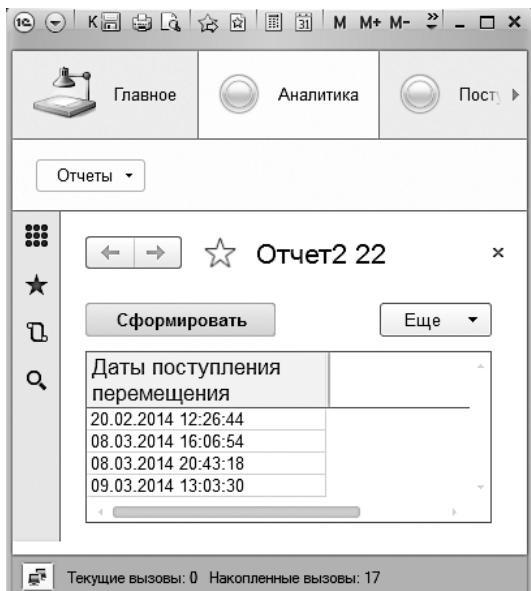


Рис. 2.47. Информация о датах из документов ПоступлениеТоваров и ПеремещениеСВыставки

Итоги

В этой главе на примерах разного уровня сложности мы познакомились с языком запросов. Практические задачи, касающиеся поступлений и продаж товаров, продемонстрировали этот важный инструмент, который разработчики используют для извлечения необходимых сведений из базы данных. Взяв в качестве исходного материала созданную ранее информационную базу, мы фактически дополнили разработанную конфигурацию средствами просмотра интересующей нас информации в удобном для пользователя виде. С языком запросов мы встретимся и в последующих главах, где много внимания уделяется построению отчетов.

Глава 3

Разработка отчетов с использованием системы компоновки данных

При решении любых задач, связанных с количественным учетом, ключевым и одновременно итоговым результатом являются *отчеты*. Одним из основных достоинств системы 1С:Предприятие 8.3 является возможность построения отчетов с минимальными усилиями со стороны разработчика. Наличие подобной автоматизации повышает скорость разработки прикладных решений и существенно облегчает задачу программиста. Возможность строить отчеты получают даже пользователи, не знакомые со встроенным языком программирования системы 1С:Предприятие.

Вся эта глава посвящена примерам использования системы компоновки данных. Указанная технология отсутствовала в версии 1С:Предприятие 8.0, а ее появление, начиная с программного продукта 1С:Предприятие 8.1, позволило разрабатывать отчеты без контакта с программным кодом на языке 1С. *Система компоновки данных* предназначена для создания отчетов на основе декларативного описания. Механизм компоновки данных позволяет дать целостное описание отчета, используя только визуальные средства. Это отличает систему компоновки данных от *построителя отчетов* (основное средство автоматизации при создании отчетов в системе 1С:Предприятие 8.0).

При работе с системой компоновки данных на этапе визуального конструирования создается основа отчета — схема компоновки данных. Также задаются параметры, применяемые по умолчанию. В дальнейшем при интерактивной работе с отчетом в режиме 1С:Предприятие пользователь может изменить имеющиеся параметры отчета, а также создать собственные.

Система компоновки данных предлагает новые возможности, недоступные построителю отчетов. Отметим основные из них:

- создание отчета без программирования;
- использование автоматически генерируемых форм просмотра и настройки отчета;

- настройка структуры отчета;
- совмещение в отчете нескольких таблиц.

Перед началом практических шагов рассмотрим систему компоновки данных в целом. Данный ресурс можно рассматривать как набор элементов, каждому из которых отводится конкретная роль при построении отчета. В итоге процесс создания отчета представляет собой переход от одного элемента к другому. Перечислим основные элементы системы компоновки данных:

- схема компоновки данных* определяет суть данных, которые используются отчетом (откуда следует получать данные и как управлять компоновкой данных);
- параметры компоновки данных* представляют собой описание всего, что может быть настроено в выбранной схеме компоновки данных (параметры отбора, упорядочивания, получения и вывода данных, а также другие параметры);
- макет компоновки данных* — готовое описание того, как должен формироваться отчет (представляет собой результат применения конкретных параметров к схеме компоновки);
- результат компоновки данных* — документ, представляемый пользователю.

Основным этапом при разработке отчета является создание *схемы компоновки данных*. На этом этапе используются визуальные средства (конструктор схемы компоновки данных). Следующие этапы процесса компоновки данных: редактирование параметров компоновки данных, подготовка к компоновке (формирование макета компоновки данных), собственно компоновка данных, вывод результата компоновки данных (отчет может быть выведен в различных форматах).

После такого взгляда на систему компоновки данных в целом рассмотрим ее на практических примерах. При этом в качестве исходной информации будем использовать две конфигурации:

- конфигурацию, разработанную в первой главе книги;
- новую конфигурацию, которую мы разработаем в этой главе (она будет связана с учетом денежных начислений).

Отчеты по остаткам товаров

Рассмотрим процесс создания ряда отчетов, которые будут относиться к информационной базе, разработанной в первой главе. Напомним, что она связана с поступлением и продажей товаров. Будем считать, что нам, как разработчикам, необходимо построить отчет, включающий в себя информацию о количестве различных товаров на наших филиалах. Другими словами, нас интересуют остатки товаров (отдельно

по каждому товару) в филиалах. В первой главе для хранения информации о количестве товаров в филиалах мы создали регистр накопления КоличествоТоваров. При работе в режиме 1С:Предприятие в информационной структуре, созданной системой на основании данного объекта конфигурации, фиксируются все движения товаров при поступлении, продаже или перемещении.

Первые технические шаги для создания отчета нам уже знакомы по второй главе книги — необходимо создать новый объект конфигурации типа отчет. На вкладке Основные окна редактирования объекта конфигурации введем имя отчета — Отчет3_1, а затем воспользуемся кнопкой Открыть схему компоновки данных. После появления окна конструктора макета, где никаких изменений делать не требуется, открывается окно конструктора схемы компоновки данных. Здесь с помощью кнопки Добавить набор данных выберем вариант Добавить набор данных — запрос. В результате на экране появится очередное окно, показанное на рис. 3.1, в правой части которого присутствует кнопка Конструктор запроса.

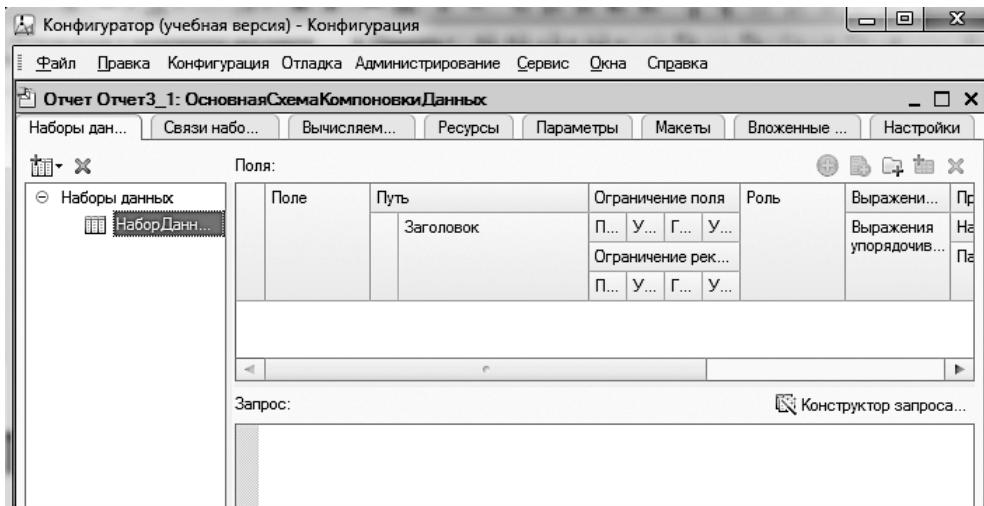


Рис. 3.1. Окно конструктора схемы компоновки данных

Если во второй главе мы самостоятельно разрабатывали необходимые тексты запросов, то теперь будем создавать запросы с помощью средства автоматизации — конструктора запроса.

В соответствие с поставленной задачей рассмотрим составление перечня имеющихся на текущий момент товаров в наших филиалах. Для этого после щелчка на кнопке Конструктор запроса перейдем к разработке запроса с помощью визуальных средств. На рис. 3.2 представлено окно конструктора запроса, в котором присутствуют три раздела:

- База данных;
- Таблицы;
- Поля.

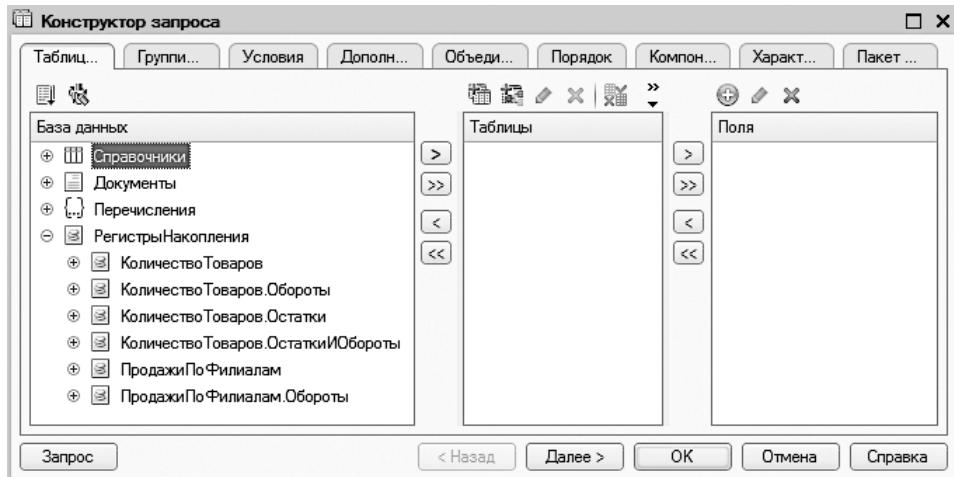


Рис. 3.2. Окно конструктора запроса

В разделе **База данных** представлены имеющиеся таблицы, в которых хранятся данные, присутствующие в информационной базе. На основании информации, содержащейся в этих таблицах, и строятся разнообразные отчеты.

В раскрытой ветке **РегистрыНакопления**, кроме созданных нами регистров **КоличествоТоваров** и **ПродажиПоФилиалам**, мы видим еще несколько виртуальных таблиц, связанных с данными регистрами. Так, три виртуальные таблицы формируются системой на основании информации, содержащейся в регистре **КоличествоТоваров**:

- КоличествоТоваров.Остатки**;
- КоличествоТоваров.Обороты**;
- КоличествоТоваров.ОстаткиИОбороты**.

Нас в данный момент интересуют остатки товаров по результатам поступления, продажи и перемещений с выставки (на выставку), поэтому перенесем в раздел **Таблицы** элемент **КоличествоТоваров.Остатки**. После этого для запроса необходимо выбрать используемые в нем поля. В рассматриваемой ситуации три поля (**Товар**, **Филиал**, **КоличествоОстаток**) следует перенести в раздел **Поля** (рис. 3.3).

Окно конструктора запроса имеет ряд вкладок, позволяющих формировать необходимые указания в запросе. Так, можно задавать группировки и условия, назначать псевдонимы полям, задавать порядок и т. д. В данном случае зададим на вкладке

Порядок упорядочивание по количеству товаров в убывающем порядке (рис. 3.4), после чего щелкнем на кнопке OK. В результате происходит формирование запроса, текст которого приведен в листинге 3.1.

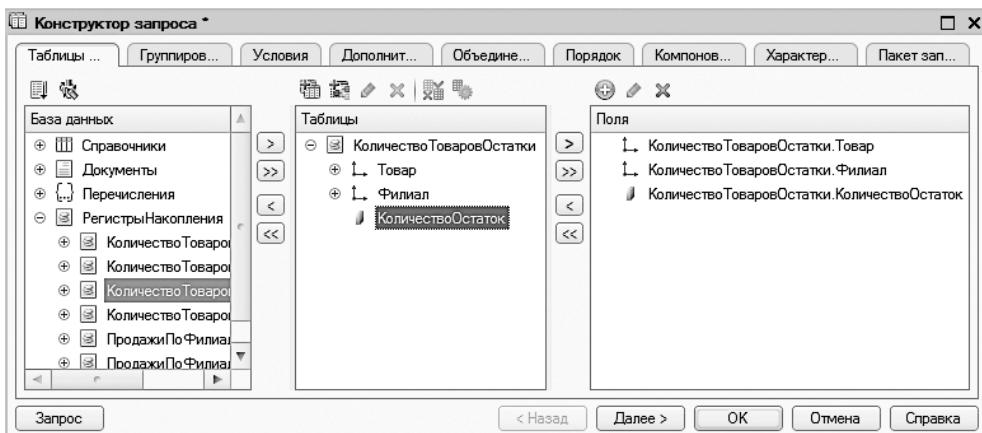


Рис. 3.3. Отбор полей для запроса по остаткам товаров

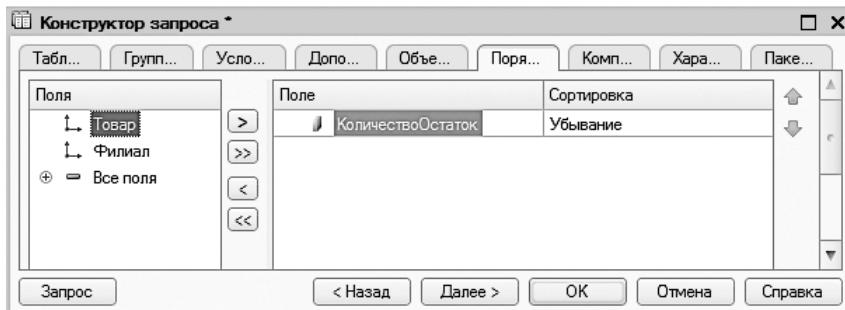


Рис. 3.4. Упорядочивание в запросе

Листинг 3.1. Запрос для извлечения информации по остаткам товаров

ВЫБРАТЬ

```
КоличествоТоваровОстатки.Товар,
КоличествоТоваровОстатки.Филиал,
КоличествоТоваровОстатки.КоличествоОстаток КАК КоличествоОстаток
```

ИЗ

```
РегистрНакопления.КоличествоТоваров.Остатки КАК КоличествоТоваровОстатки
УПОРЯДОЧИТЬ ПО
КоличествоОстаток УБЫВ
```

В результате проделанных действий на вкладке Наборы данных схемы компоновки данных происходят изменения, и на рис. 3.5 показаны поля, которые в данном случае заполняются автоматически.

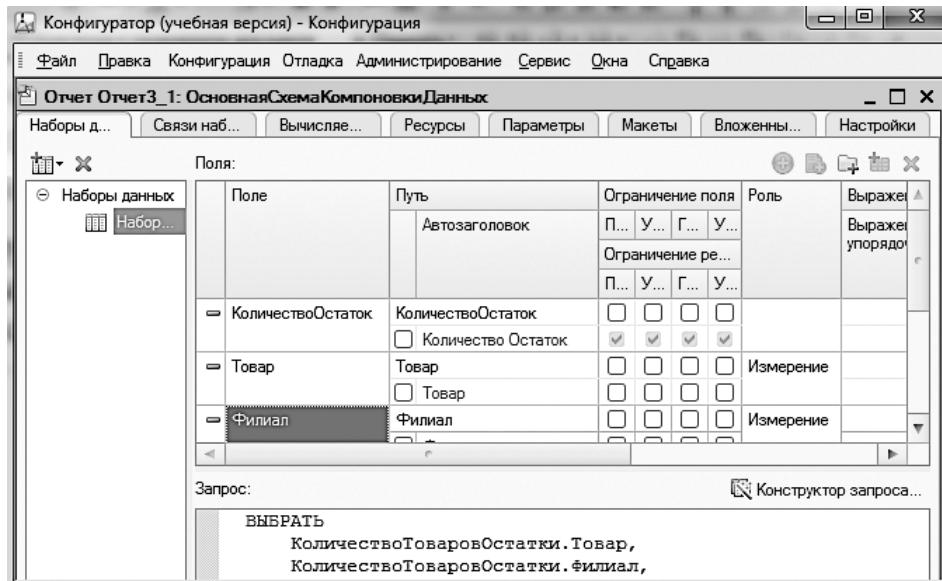


Рис. 3.5. Заполнение полей в схеме компоновки данных

ПРИМЕЧАНИЕ

Поля для текущего набора данных формируются системой автоматически только в случае, если в конструкторе схемы компоновки данных установлен флаг Автозаполнение.

Прокомментируем раздел Поля на рис. 3.5. Первый столбец с информацией называется Поля; в нем отражается имя поля набора данных, для которого формируется описание. Учитывая, что для получения набора данных мы воспользовались запросом, содержание поля определяется именем поля в тексте запроса. Важно отметить, что информация в столбце Поля не допускает редактирования.

Поле Путь содержит имя поля набора данных, под которым можно обращаться к полю в других разделах конструктора схемы компоновки данных. Значение этого поля можно устанавливать самостоятельно.

Поле Роль показывает, каким образом поле будет использовано в дальнейшем в системе компоновки данных. Информация о том, что поле используется для измерения, требуется при расчете итогов.

Сейчас наша задача заключается в том, чтобы получить информацию о количестве товаров в разрезе наших филиалов (сколько и каких товаров находится в каждом из филиалов). Для этого в окне схемы компоновки данных перейдем на вкладку Настройки (рис. 3.6), где необходимо задать параметры отчета, применяемые по умолчанию. Здесь следует щелкнуть правой кнопкой на элементе Отчет и в появившемся контекстном меню выбрать вариант Новая группировка.

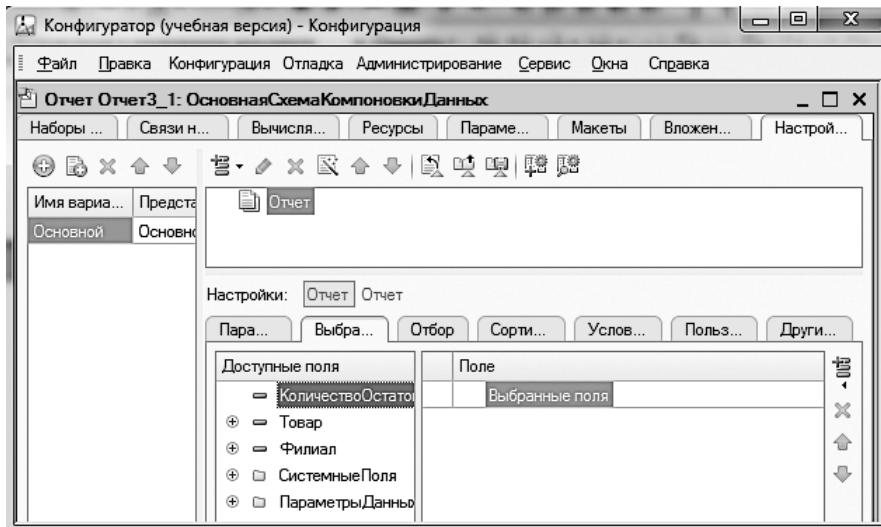


Рис. 3.6. Вкладка Настройки окна конструктора схемы компоновки

ПРИМЕЧАНИЕ

В нижней части вкладки Настройки имеется еще несколько дополнительных вкладок, позволяющих сформировать необходимый отчет.

Благодаря *группировке* в отчете можно собрать не только сгруппированную определенным образом информацию, но и детальные записи, сформированные системой в результате обработки текста запроса. И если мы хотим обеспечить вывод в отчете детальных записей, то в окне Группировка следует оставить пустой строку Поле (рис. 3.7).

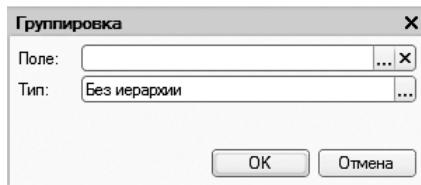


Рис. 3.7. Окно для определения группировки

Перейдем теперь на дополнительную вкладку Выбранные поля, где в число выбранных полей следует добавить три поля: Товар, Филиал, КоличествоТоваров. Добавить поле в раздел Поле проще всего двойным щелчком на элементе в разделе Доступные поля (рис. 3.8). Также элемент можно перетащить из раздела Доступные поля мышью. Еще один вариант — щелкнуть на кнопке Добавить (она находится в пиктографическом меню, расположенном в правой части окна).

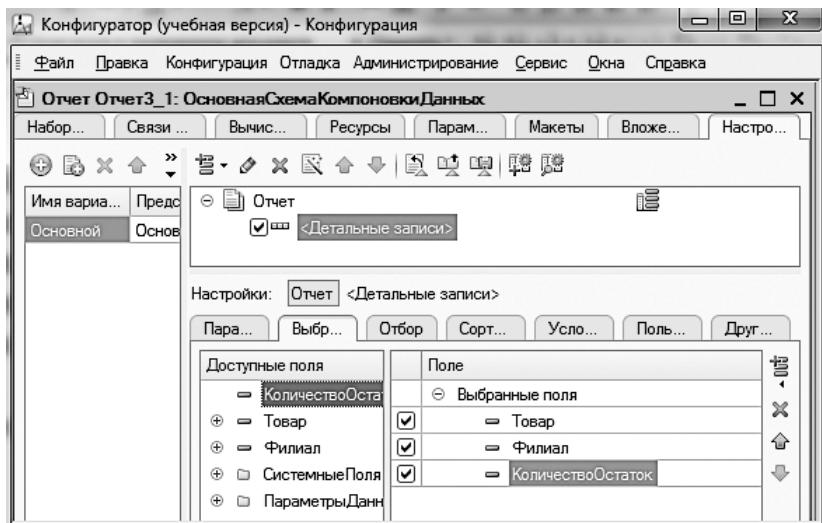


Рис. 3.8. Заполнение раздела Выбранные поля

Осталось отнести созданный отчет к одной из подсистем (для отчетов данной главы создадим подсистему с именем ОтчетыПоГлавеЗ). После выполнения этих визуальных действий можно посмотреть, как выглядит разработанный отчет в режиме 1С:Предприятие (рис. 3.9).

The screenshot shows the generated report 'Отчет3_1' in the 1C:Enterprise environment. The report displays a table with columns: Товар (Goods), Филиал (Branch), and Количество Остаток (Quantity). The data is as follows:

Товар	Филиал	Количество Остаток
Стол кухонный	Сормовский	9
Стенка "Светлопар"	Окский	5
Стол кухонный	Окский	5
Прихожая "Уют"	Сормовский	4
Прихожая "Уют"	Окский	3
Стенка "Светлопар"	Сормовский	2

Рис. 3.9. Отчет по остаткам товаров

Использование системы компоновки данных позволяет расширить возможности формирования отчетов без внесения изменений в конфигурацию. Рассмотрим, как это выглядит на практике. Будем считать, что в нашем отчете (Отчет3_1) необходимо просмотреть сводную информацию, а именно вывести данные о суммарном количестве каждого товара на всех филиалах (просуммировать каждый товар по всем филиалам). Для этого щелкнем на кнопке Еще (см. рис. 3.9), после чего раскроется новое меню, где следует выбрать пункт Изменить вариант. Далее в очередном открывшемся окне необходимо обратиться к вкладке Поля и в списке выбранных полей сбросить флажок для поля Филиал (рис. 3.10).

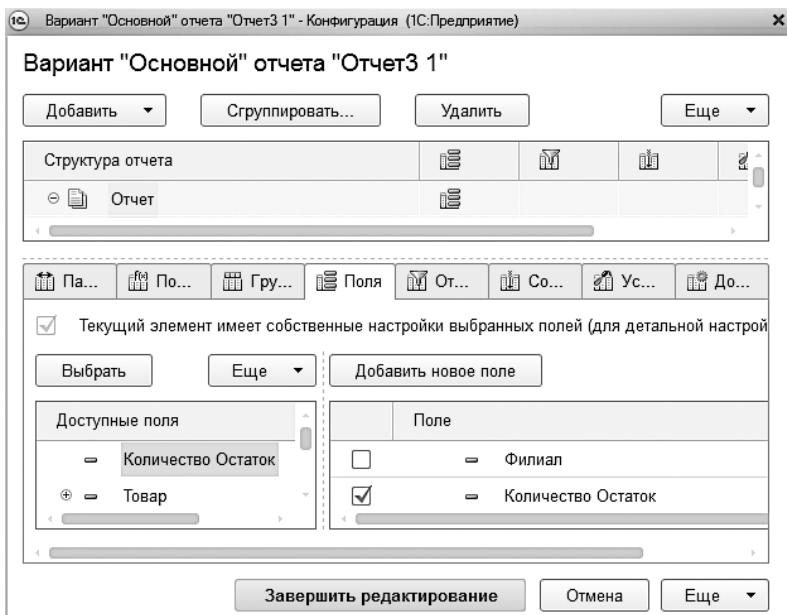


Рис. 3.10. Настройка отчета для получения сводной информации по товарам

Далее следует щелкнуть на кнопке Завершить редактирование и сформировать обычным способом уже новую версию отчета (рис. 3.11).

Обратим внимание на то, что изменение в формировании отчета (изменение вида отчета) мы выполнили *непосредственно* в режиме 1С:Предприятие. То есть подобная модификация доступна любому пользователю.

Продолжим практическую работу и рассмотрим задачу формирования в одном отчете вместе с детализированной информацией еще и итоговых сведений по каждому товару. При этом будем считать, что выводимую информацию необходимо отсортировать по товарам, а детализированные данные еще дополнительно упорядочить по филиалам. Для того, чтобы было понятно, какой отчет нам предстоит разработать, приведем его в режиме 1С:Предприятие (рис. 3.12).

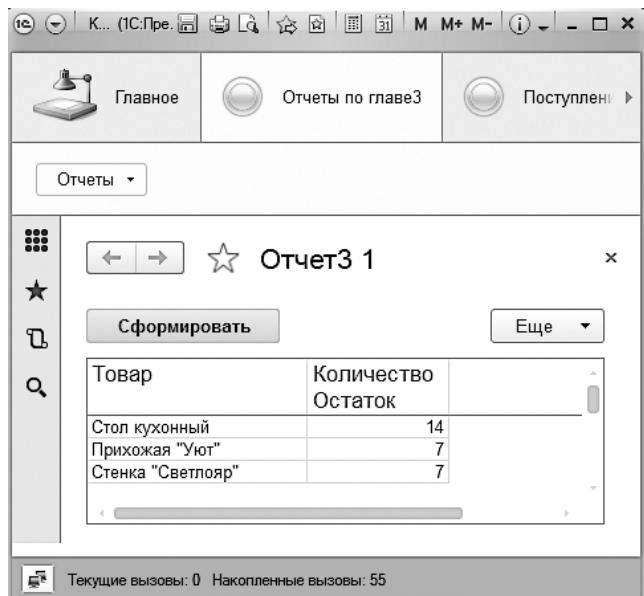


Рис. 3.11. Сводная информация по остаткам товаров

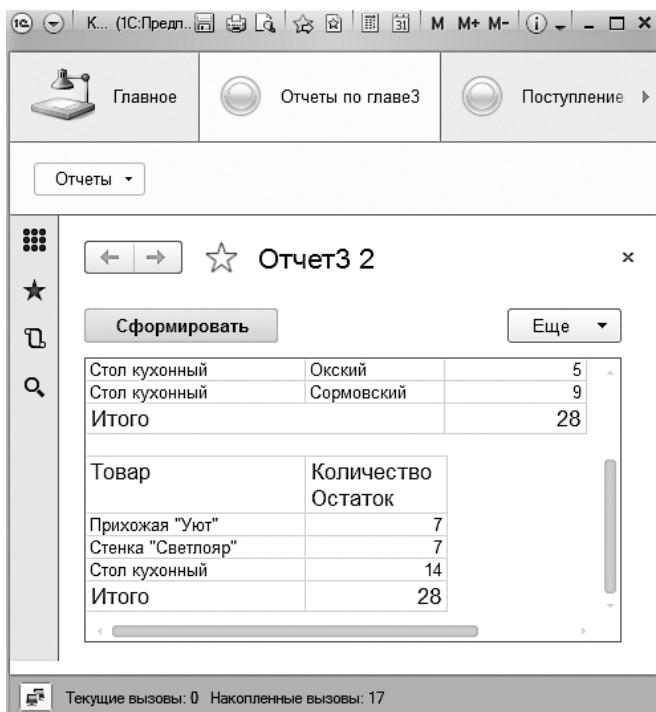


Рис. 3.12. Отчет с двумя таблицами

Вернемся в окно конструктора запросов и немного изменим вариант запроса (рис. 3.13).

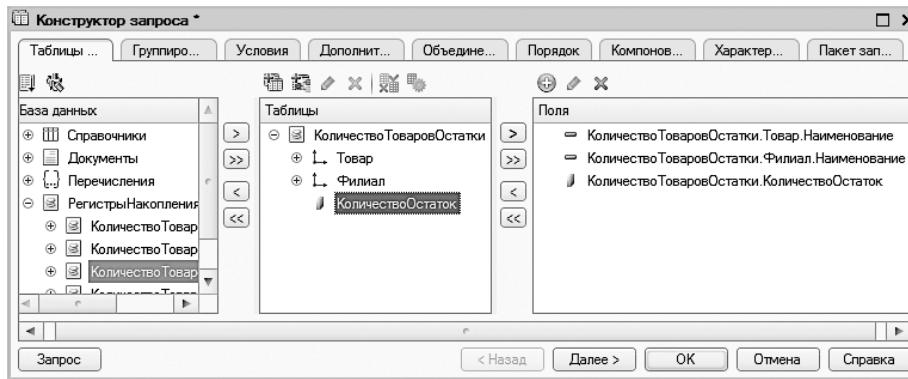


Рис. 3.13. Окно конструктора запросов

На вкладке Объединения/Псевдонимы конструктора запроса установим псевдонимы для двух полей (рис. 3.14).

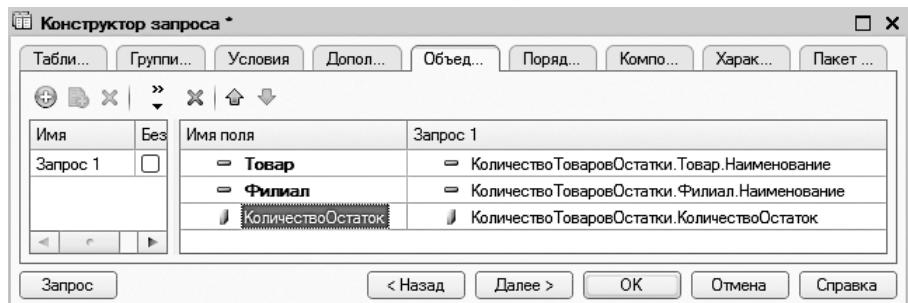


Рис. 3.14. Вкладка Объединения/Псевдонимы в окне конструктора запросов

Далее на вкладке Порядок следует задать параметры для последовательной сортировки сначала по товарам, а затем по филиалам (рис. 3.15).

После этого щелчком на кнопке OK завершим работу с конструктором запроса. В результате в окне схемы компоновки перед нами появится текст запроса (листинг 3.2).

Листинг 3.2. Запрос для извлечения информации по товарам с сортировкой

ВЫБРАТЬ

```
КоличествоТоваровОстатки.Товар.Наименование КАК Товар,
КоличествоТоваровОстатки.Филиал.Наименование КАК Филиал,
```

КоличествоТоваровОстатки.КоличествоОстаток
 из
 РегистрНакопления.КоличествоТоваров.Остатки КАК КоличествоТоваровОстатки
 УПОРЯДОЧИТЬ ПО
 Товар,
 Филиал

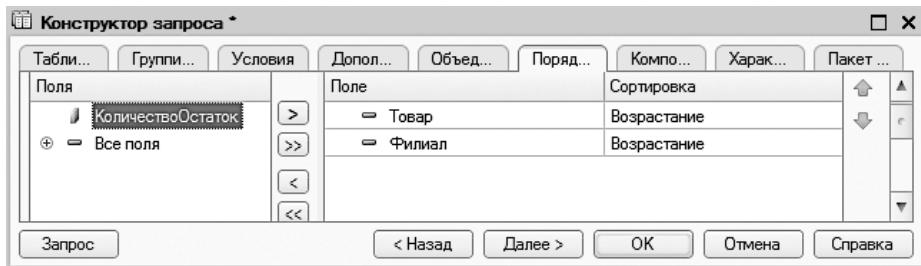


Рис. 3.15. Установки параметров сортировки данных на вкладке Порядок

Далее в окне конструктора схемы компоновки данных необходимо перейти на вкладку Ресурсы и добавить ресурс КоличествоОстаток (рис. 3.16). Это можно сделать двойным щелчком на поле в разделе Доступные поля. Также можно перетащить его из раздела Доступные поля мышью или щелкнуть на предназначенной для этого кнопке (расположена на вертикальной полосе).

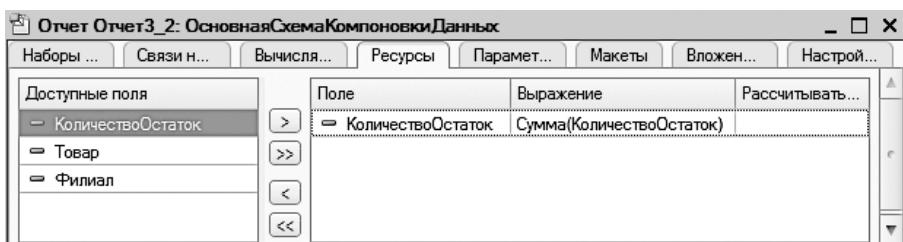


Рис. 3.16. Вкладка Ресурсы окна конструктора схемы компоновки данных

В правой части окна на рис. 3.16 присутствует несколько столбцов. Так, в столбце Поле указывается внутреннее имя поля набора данных (для данного параметра редактирование запрещено), а в столбце Выражение – выражение, по которому будет определяться значение ресурса. В качестве выражения можно выбрать одну из агрегатных функций, встречавшихся нам во второй главе книги. Значение в столбце Рассчитывать по... показывает, что данное поле ресурса применимо только для конкретной группировки.

Перейдем теперь на вкладку Настройки, где следует выбрать необходимые поля для отчета и добавить две группировки в соответствии с рис. 3.17:

- Детальные записи;
- Товар.

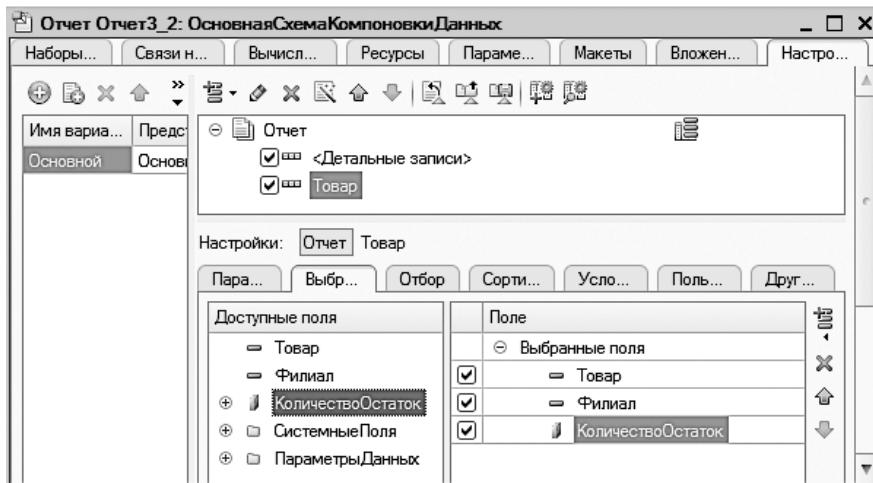


Рис. 3.17. Задание параметров группировки на вкладке Настройки

На этом этап разработки завершается; далее необходимо перейти в режим 1С:Предприятие для формирования отчета, который представлен на рис. 3.18. Как видно, отчет включает две таблицы — отдельную с детализацией товаров по филиалам и общую с итоговыми показателями.

В рассмотренных примерах отчетов мы не использовали дату, на которую производился вывод информации об остатках. Поэтому то, что мы наблюдали в электронных формах, представляет собой результат по остаткам на текущий момент. Однако часто требуется просмотреть информацию по товарам на более раннюю дату. Для этого следует использовать параметр Период, который автоматически включается в данный отчет системой на основании таблицы КоличествоТоваров.Остатки. Нам надо только дать пользователю возможность выбрать этот параметр в форме отчета.

В данном случае мы не будем создавать новый отчет, а всего лишь модифицируем предыдущий (с названием Отчет3_2). Для включения параметра Период перейдем на дополнительную вкладку Параметры вкладки Настройки, где воспользуемся соответствующей кнопкой (последняя в вертикальном пиктографическом меню) для задания требуемых параметров. В открывшемся в результате новом окне установим флажок Включать в пользовательские настройки (рис. 3.19).

На этом наши действия завершаются, и можно перейти в режим 1С:Предприятие для открытия формы отчета. Теперь перед заполнением отчета следует предварительно задать значение параметра Период (рис. 3.20).

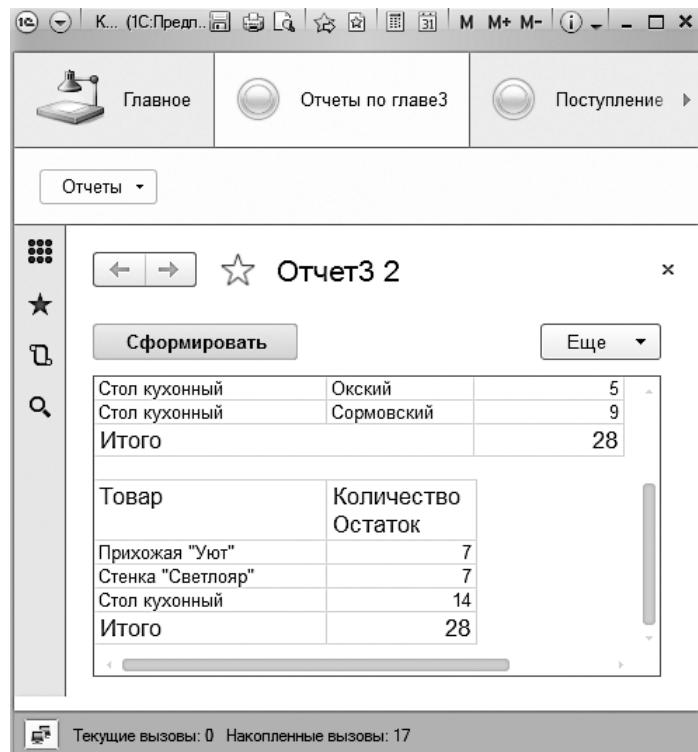


Рис. 3.18. Отчет с двумя таблицами

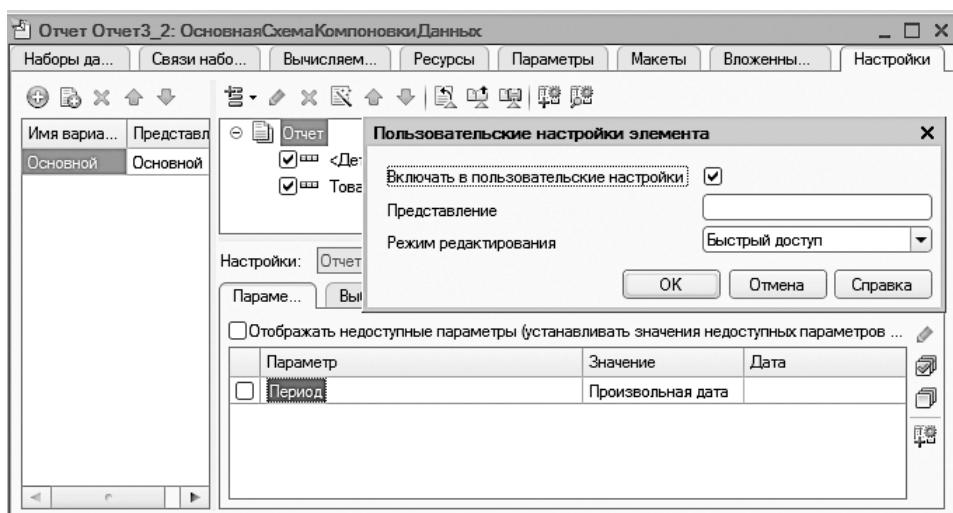


Рис. 3.19. Окно для выполнения пользовательских настроек

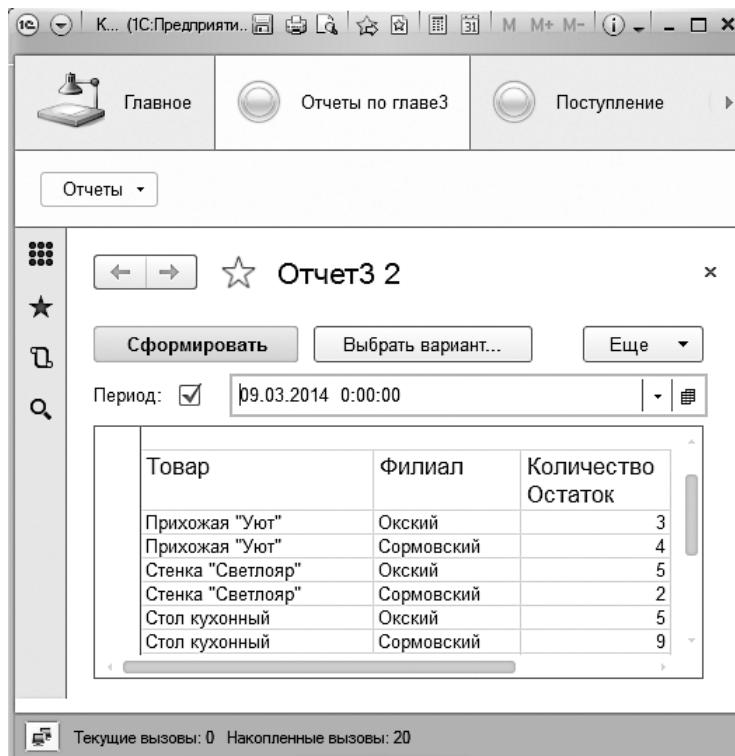


Рис. 3.20. Включение параметра Период в отчет по остаткам товаров

Отчет по документам Поступление Товаров

Разработаем отчет, который будет включать информацию по поступлениям товаров. Для этого требуется сформировать перечень документов Поступление Товаров, имеющихся в базе данных. Начнем привычные действия с создания нового объекта конфигурации — Отчет3_3. На вкладке Основные, как и ранее, воспользуемся кнопкой Открыть схему компоновки данных. В меню конструктора схемы компоновки выберем пункт Добавить набор данных — запрос и с помощью соответствующей кнопки перейдем к работе с конструктором запроса. Нам потребуется таблица Поступление Товаров, из которой для запроса необходимо отобрать следующие поля (рис. 3.21):

- Ссылка;
- Фирма;
- Филиал.

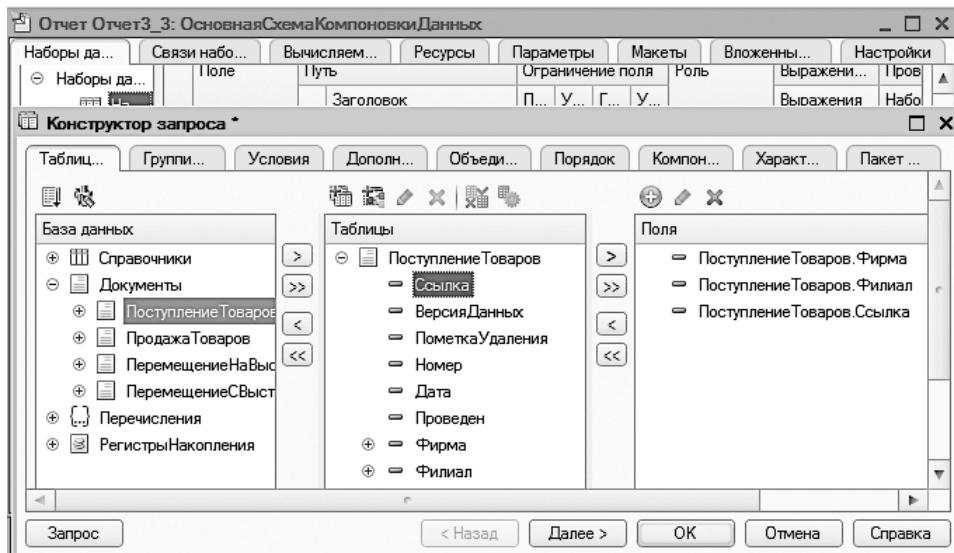


Рис. 3.21. Отбор полей для запроса при формировании отчета по документам

Сформируем псевдоним **ДокументПоступление** для поля **Ссылка** (рис. 3.22), а на вкладке **Порядок** зададим упорядочивание документов по убыванию.

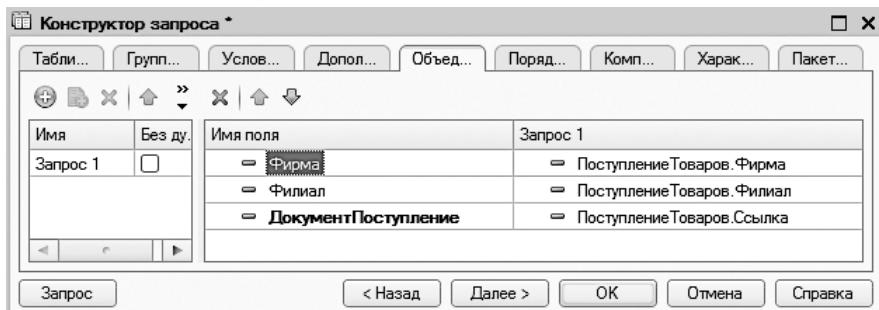


Рис. 3.22. Вкладка Объединения/Псевдонимы окна конструктора запроса

Задавать для запроса другие параметры не требуется, и щелчок на кнопке **OK** приводит к формированию текста запроса (листинг 3.3).

Листинг 3.3. Текст запроса для извлечения информации о поступлениях товаров

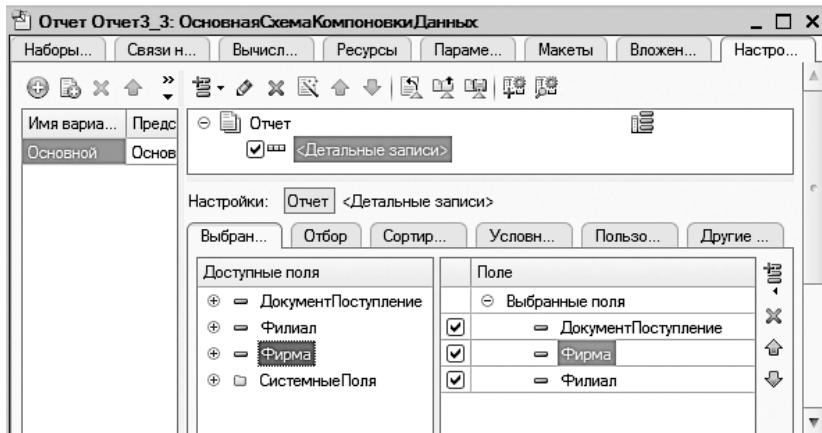
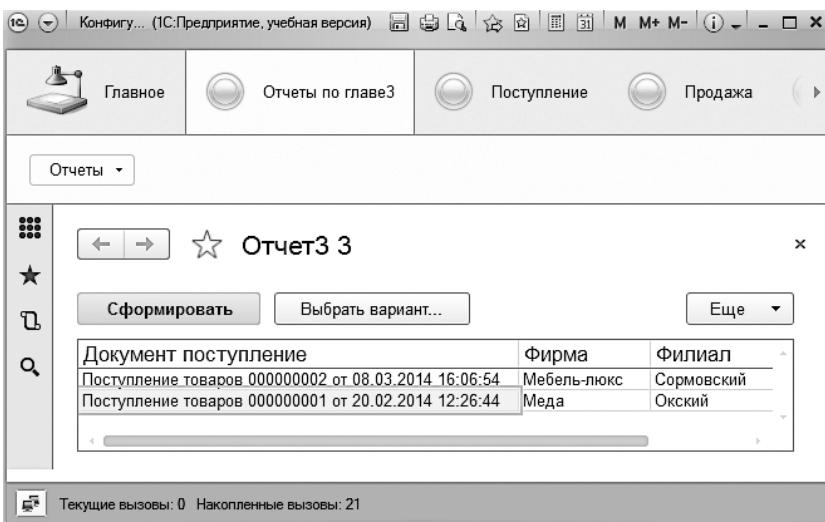
ВЫБРАТЬ

```
ПоступлениеТоваров.Фирма,
ПоступлениеТоваров.Филиал,
ПоступлениеТоваров.Ссылка КАК ДокументПоступление
Из
```

продолжение ⇨

Листинг 3.3 (продолжение)

```
Документ.ПоступлениеТоваров КАК ПоступлениеТоваров
УПОРЯДОЧИТЬ ПО
ДокументПоступление УБЫВ
```

**Рис. 3.23.** Настройка схемы компоновки данных**Рис. 3.24.** Доступ к документу из отчета по сведениям о поступлении товаров

В запросе используется объектная таблица **Документ.ПоступлениеТоваров**. После такого автоматического получения текста запроса перейдем на вкладку **Настройки**. Укажем, что в группировке будут выводиться детальные записи из информационной базы, и перенесем необходимые поля в раздел **Выбранные поля** (рис. 3.23).

Далее осталось перейти в режим 1С:Предприятие и сформировать отчет по имеющимся документам ПоступленияТоваров в информационной базе (рис. 3.24).

Важно заметить, что при работе с формой данного отчета мы можем открыть необходимый документ, а также выполнить ряд других действий, которые нам предоставляет система компоновки данных.

Отчет по работе филиалов

Рассмотрим создание отчета, который позволит получить информацию об уровне продаж товаров в наших филиалах. На примере этого отчета мы узнаем о возможности отбора информации в указываемом пользователем временном интервале.

Начнем разработку нового объекта конфигурации — отчета под именем Отчет3_4. Для этого воспользуемся схемой компоновки данных и в конструкторе схемы компоновки создадим новый набор данных — запрос. Далее используем конструктор запроса, в окне которого выберем две таблицы:

- объектную таблицу справочника Филиалы;
- виртуальную таблицу ПродажиПоФилиаламОбороты.

Из данных таблиц в раздел Поля перенесем два поля (рис. 3.25):

- Филиалы.Ссылка;
- ПродажиПоФилиаламОбороты.СуммаОборот.

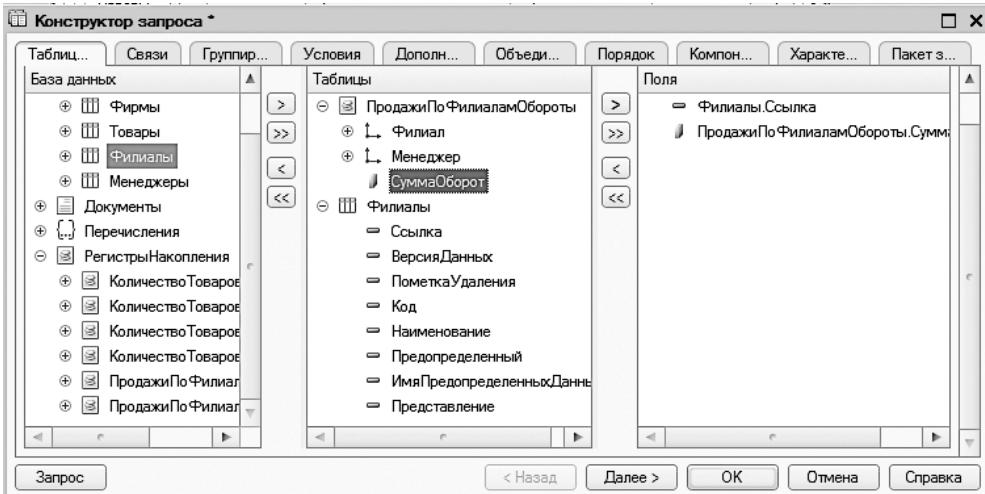


Рис. 3.25. Окно конструктора запроса для формирования отчета по работе филиалов

В отличие от предыдущих примеров этой главы, здесь в запросе мы использовали две таблицы, которые необходимо связать. Ранее во второй главе книги были рассмотрены различные варианты связывания таблиц, а также синтаксические конструкции языка запросов, позволяющие это сделать. В данном случае подобные установки мы реализуем визуальными действиями. Для начала на вкладке **Связи** следует убедиться в установке необходимой связи между двумя таблицами (рис. 3.26).

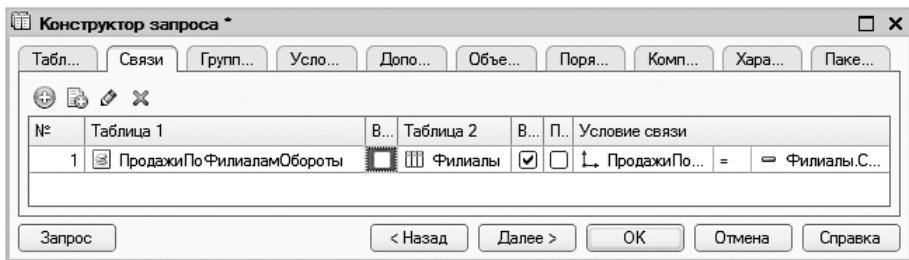


Рис. 3.26. Установка связей между таблицами в запросе

Значение измерения **Филиал** виртуальной таблицы должно быть равно ссылке на элемент справочника **Филиалы**. При этом обязательно требуется проверить, что у таблицы справочника установлен флагок **Все**. В этом случае из справочника выбираются все элементы справочника, и этим элементам ставится в соответствие значение оборота по данному филиалу из виртуальной таблицы регистра. Таким образом, в результате выполнения запроса из базы данных извлекаются все филиалы, а для ряда их (в которых были продажи) указывается значение суммы оборота продаж.

После вкладки **Связи** перейдем на вкладку **Объединения/Псевдонимы**, где установим псевдоним **НашФилиал** вместо сочетания **Филиалы.Ссылка**. Новое словосочетание намного информативнее для пользователя, чем **Филиалы.Ссылка**. На вкладке **Порядок** укажем, что результат должен быть отсортирован по значению поля **СуммаОборот** в порядке убывания. После этого щелкнем на кнопке **OK** и получим текст запроса (листинг 3.4).

Листинг 3.4. Текст запроса для формирования отчета по работе филиалов

```

ВЫБРАТЬ
    Филиалы.Ссылка КАК НашФилиал,
    ПродажиПоФилиаламОбороты.СуммаОборот КАК СуммаОборот
ИЗ
    Справочник.Филиалы КАК Филиалы
    ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ПродажиПоФилиалам.Обороты КАК
        ПродажиПоФилиаламОбороты
    ПО ПродажиПоФилиаламОбороты.Филиал = Филиалы.Ссылка
УПОРЯДОЧИТЬ ПО
    СуммаОборот УБЫВ

```

В данном случае в запросе используется уже знакомое *левое соединение* таблиц, о котором мы говорили во второй главе. Однако вместо написания программного кода необходимая конструкция текста запроса здесь получена визуальными действиями.

После формирования текста запроса система автоматически возвращает нас в окно конструктора схемы компоновки данных, где на вкладке Ресурсы следует включить в список ресурсов поле СуммаОборот. В качестве выражения установим суммирование по данному полю – Сумма(СуммаОборот).

Перейдем далее на вкладку Параметры, где уже указано два параметра (рис. 3.27). Это связано с тем, что система компоновки данных самостоятельно анализирует текст запроса и добавляет те параметры, которые, скорее всего, будут использоваться в отчете. Присутствующие здесь параметры НачалоПериода и КонецПериода являются параметрами виртуальной таблицы РегистрНакопления.ПродажиПоФилиалам.Обороты.

Отчет Отчет3_4: Основная Схема Компоновки Данных							
Набор...	Связи ...	Вычисл...	Ресурсы	Парем...	Макеты	Вложе...	Настро...
Имя	Заголовок	Тип	Доступн...	Д...	Значение	Выражение	
НачалоПериода	Начало периода	Дата	...	<input type="checkbox"/>			
КонецПериода	Конец периода	Дата		<input type="checkbox"/>			

Рис. 3.27. Вкладка Параметры схемы компоновки данных

Первый параметр (НачалоПериода) представляет собой начало периода расчета итогов, второй (КонецПериода) – конец периода. В результате результирующая таблица будет содержать обороты, рассчитанные в указанном пользователем интервале дат.

Важно отметить, что тип данных *Дата* кроме непосредственно *даты* содержит еще и *время*. Разумеется, при указании интервала дат присутствие времени является излишним. Для необходимой коррекции (исключения конкретного времени из формата даты) необходимо дважды щелкнуть в поле Тип параметра НачалоПериода и с помощью кнопки с многоточием открыть окно Редактирование типа данных (рис. 3.28). Здесь в списке Состав даты следует выбрать вариант *Дата*.

Особенностью системы является то, что по умолчанию время в дате установлено как 00:00:00. В связи с этим, если пользователь задаст период расчета, скажем, до 20.02.08, то данные на этот день (время создания которых позже 00:00:00) в расчет не войдут. Поэтому при задании интервала по дате в процессе формировании отчета это надо учитывать.

ПРИМЕЧАНИЕ

В литературе [5,6] приведен ряд примеров, в которых с помощью программных конструкций реализуется включение в результат информации по указанной дате включительно.

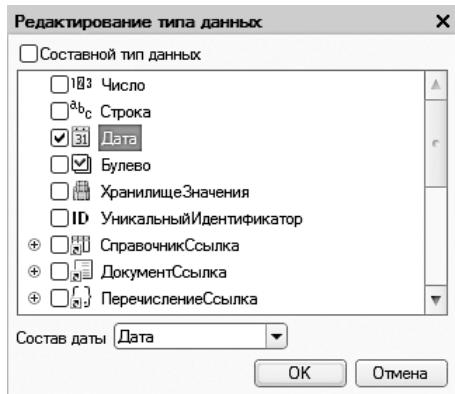


Рис. 3.28. Окно для редактирования типа данных

Перейдем теперь на вкладку Настройки, где добавим группировку без указания поля группировки. Здесь же на дополнительной вкладке Выбранные поля в качестве выбранных полей укажем поля СуммаОборот и НашФилиал (рис. 3.29).

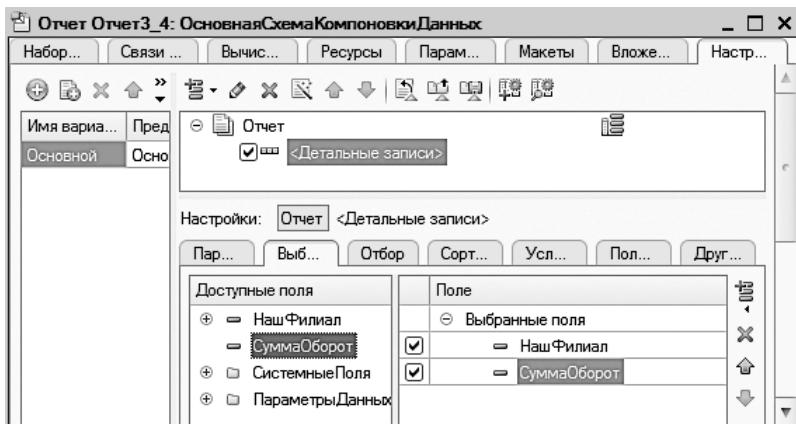


Рис. 3.29. Настройки отчета продаж по филиалам

Необходимо также включить параметры отчета (НачалоПериода и КонецПериода) в пользовательские настройки, после чего можно перейти в режим 1С:Предприятие и посмотреть на отчет. Но перед формированием отчета требуется предварительно

задать параметры — интервал по дате (рис. 3.30). В результате в полученный отчет будут включены все наши филиалы с указанием оборота продаж в указанном интервале времени для тех филиалов, где такие продажи имели место.

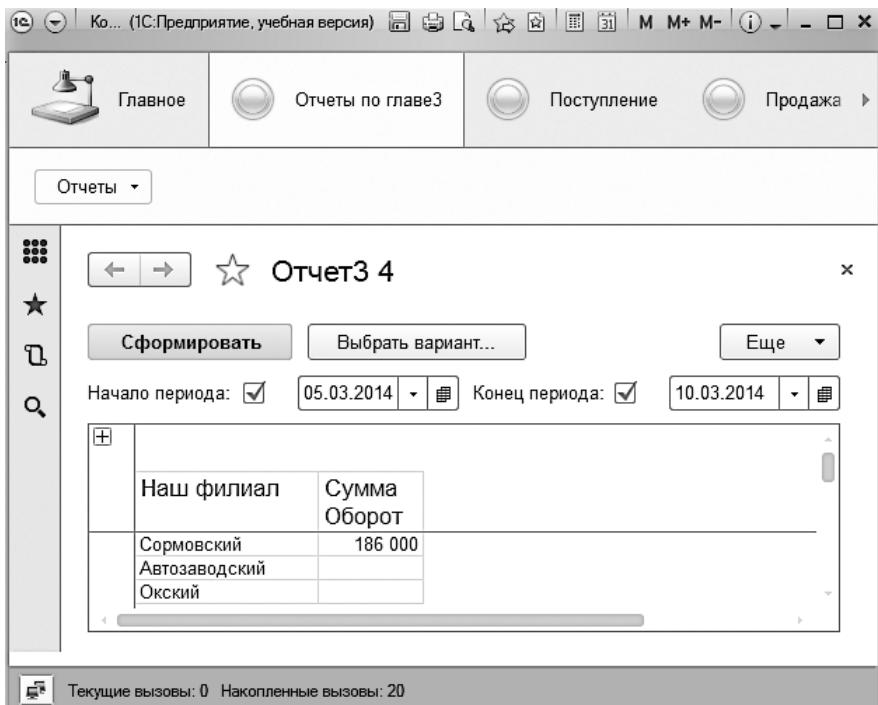


Рис. 3.30. Установка интервала дат для отчета по продажам

Отчет по работе менеджеров

В этом разделе наша задача заключается в построении отчета, в котором производится сравнение результатов работы менеджеров. Это сравнение мы проиллюстрируем с помощью диаграммы (такой вариант представления подобной информации более нагляден). В запросе мы будем использовать данные из двух таблиц. Одна из них — объектная таблица справочника Менеджеры, а другая виртуальная — ПродажиПоФилиалам.Обороты. Наша цель — получить перечень данных из набора строк и двух столбцов. Каждая строка будет содержать фамилию менеджера и сумму проданных им товаров. Если какой-то менеджер за указанный пользователем период дат ничего не продал, то его фамилия на диаграмме присутствовать все равно будет, однако соответствующий показатель окажется минимальным.

Итак, перейдем к практическим действиям и создадим в режиме конфигуратора объект конфигурации Отчет3_5. После ввода на вкладке Основные имени отчета щелкнем на кнопке Открыть схему компоновки данных. Далее в окне конструктора схемы компоновки данных выберем вариант Добавить набор данных — запрос и перейдем к работе с конструктором запроса (рис. 3.31). Здесь нам требуется выбрать две упомянутые таблицы:

- таблицу справочника Менеджеры;
- виртуальную таблицу ПродажиПоФилиалам.Обороты.

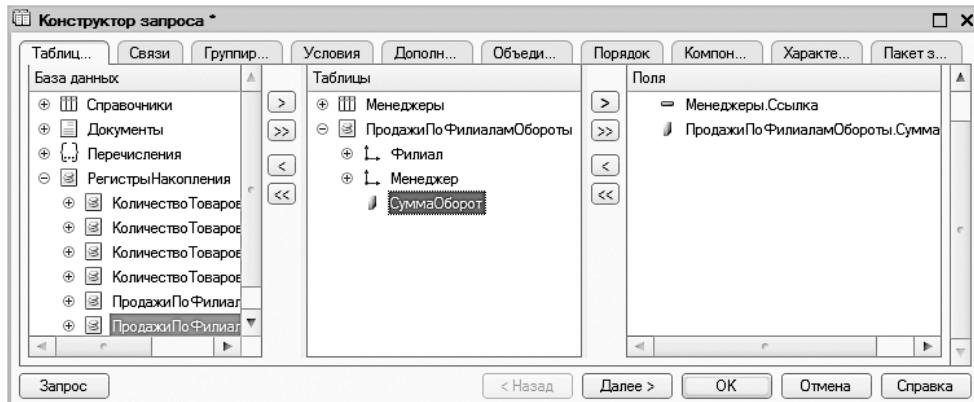


Рис. 3.31. Выбор таблиц и полей в окне конструктора запросов

В соответствии с рис. 3.31 перенесем в список полей запроса два поля: Менеджеры.Ссылка и ПродажиПоФилиаламОбороты.СуммаОборот. После этого перейдем на вкладку Связи для установки необходимой связи между таблицами (рис. 3.32). Флажок Все следует установить только для таблицы справочника. Данное действие приведет к тому, что из справочника будут выбраны все элементы и этим элементам поставлено в соответствие значение оборота из виртуальной таблицы регистра. В этом случае для тех менеджеров, у которых не зафиксировано продаж, не будет указано ничего.

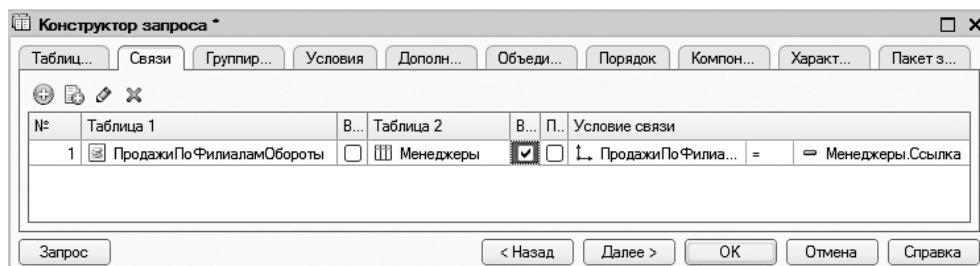


Рис. 3.32. Установка связи между таблицами в запросе

Перейдем далее на вкладку Объединения/Псевдонимы и обратим внимание на то, чтобы представление полей справочника и виртуальной таблицы оборотов имели понятные псевдонимы (рис 3.33).

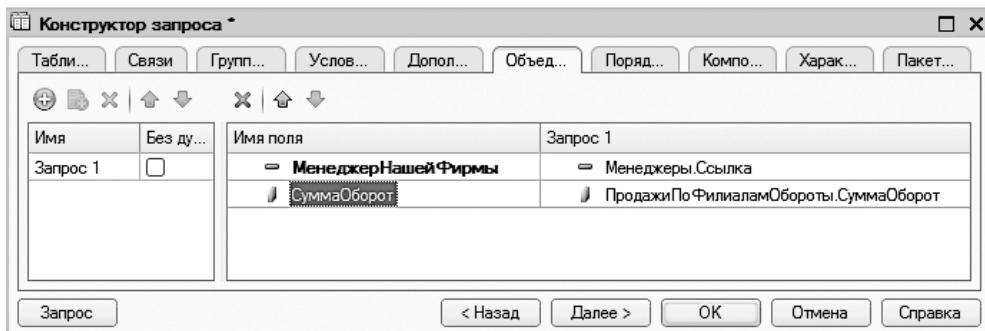


Рис. 3.33. Вкладка Объединения/Псевдонимы окна конструктора схемы компоновки данных

На этом действия по конструированию запроса завершаются, остается щелкнуть на кнопке OK, чтобы получить текст интересующего нас запроса (листинг 3.5).

Листинг 3.5. Запрос для извлечения информации о менеджерах

ВЫБРАТЬ

```
Менеджеры.Ссылка КАК МенеджерНашейФирмы,
ПродажиПоФилиаламОбороты.СуммаОборот КАК СуммаОборот
ИЗ
Справочник.Менеджеры КАК Менеджеры
ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ПродажиПоФилиалам.Обороты КАК
ПродажиПоФилиаламОбороты
ПО ПродажиПоФилиаламОбороты.Менеджер = Менеджеры.Ссылка
```

Продолжим формирование отчета в окне схемы компоновки данных, где следует перейти на вкладку Ресурсы и выбрать ресурс СуммаОборот (рис. 3.34).



Рис. 3.34. Вкладка Ресурсы окна конструктора схемы компоновки данных

Теперь следует перейти на вкладку **Настройки** и задать параметры отчета в соответствии с рис. 3.35. Здесь мы выбрали вариант **Диаграмма**, позволяющий размещать диаграммы и графики в таблицах и формах.

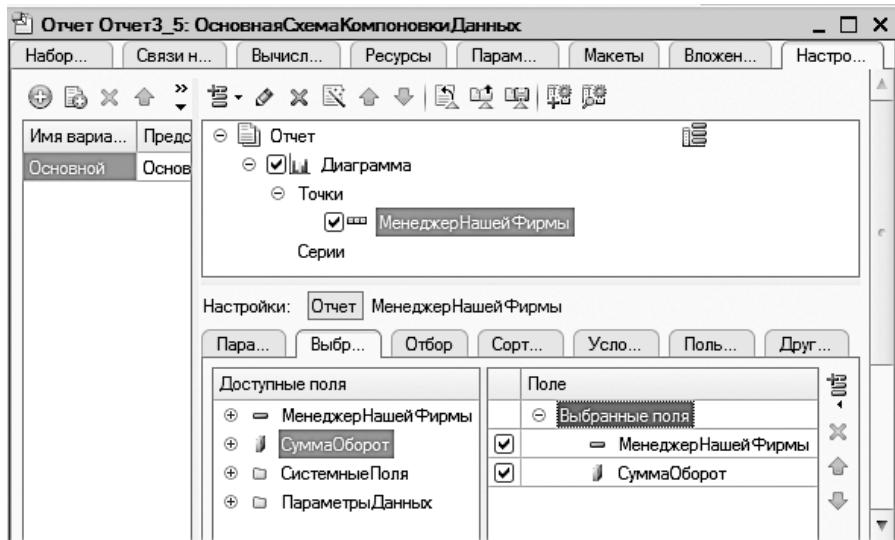


Рис. 3.35. Настройка отчета по работе менеджеров

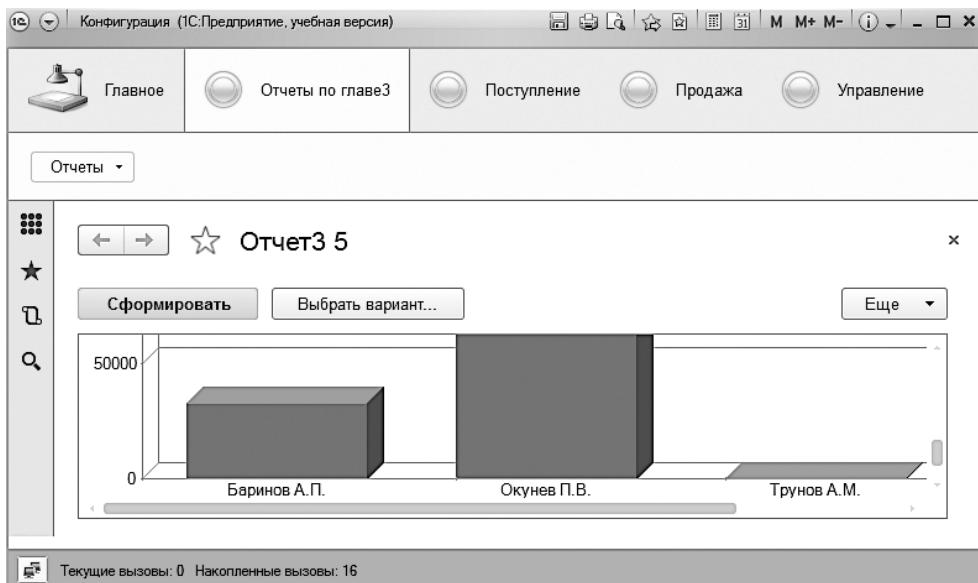


Рис. 3.36. Диаграмма показателей работы менеджеров

ПРИМЕЧАНИЕ

Диаграмма строится на основании матрицы данных и представляет собой совокупность точек, серий и значений серий в точке. Например, диаграмма поступления конкретного товара по дням будет состоять из точек — дней, серий — разных филиалов и значений — количества единиц товара.

Теперь, когда разработка готова, осталось перейти в режим 1С:Предприятие и сформировать отчет по работе менеджеров (рис. 3.36).

Использование конструктора настроек при создании отчетов

При разработке всех предыдущих отчетов мы выполняли ручную настройку в конструкторе схемы компоновки данных. В этом разделе мы рассмотрим дополнительные возможности автоматизации разработки отчетов, а именно познакомимся с *конструктором настроек*. Для определенности сразу посмотрим на отчет, который нам предстоит разработать (рис. 3.37).

The screenshot shows the 1C:Enterprise configuration interface. At the top, there is a toolbar with various icons. Below it, a navigation bar with tabs: 'Главное' (Main), 'Отчеты по главе З' (Report by Chapter 3), 'Поступление' (Arrival), 'Продажа' (Sale), and 'Управление' (Management). A dropdown menu 'Отчеты' (Reports) is open. In the main area, a report titled 'Отчет 6' is displayed. The report has two buttons: 'Сформировать' (Generate) and 'Выбрать вариант...' (Select variant...). It includes date filters: 'Начало периода:' with a checked checkbox and '01.03.2014 0:00:00', and 'Конец периода:' with a checked checkbox and '10.03.2014 0:00:00'. A 'Еще' (More) button is also present. The report content is a table showing product stock levels across different branches. The table has columns: Товар (Product), Количество Конечный остаток (Quantity End-of-period balance), Количество Приход (Quantity Arrival), and Количество Расход (Quantity Expenditure). The rows show data for three branches: Стол кухонный (Kitchen table), Окский (Oksky), Сормовский (Sormovsky), and Итого (Total). The total values are 28, 25, and 7 respectively. On the left side of the report area, there is a vertical toolbar with icons for search, filter, and other report-related functions. At the bottom of the screen, a status bar displays 'Текущие вызовы: 0 Накопленные вызовы: 23' (Current calls: 0 Accumulated calls: 23).

Рис. 3.37. Отчет по приходу, расходу и конечному остатку товаров

Начнем, как обычно, с создания нового объекта конфигурации (Отчет3_6). На рис. 3.38 показаны поля для отчета, которые нам потребуется выбрать с использованием конструктора запросов. Единственная таблица, которая в данном случае понадобится, — это виртуальная таблица КоличествоТоваров.ОстаткиИОбороты.

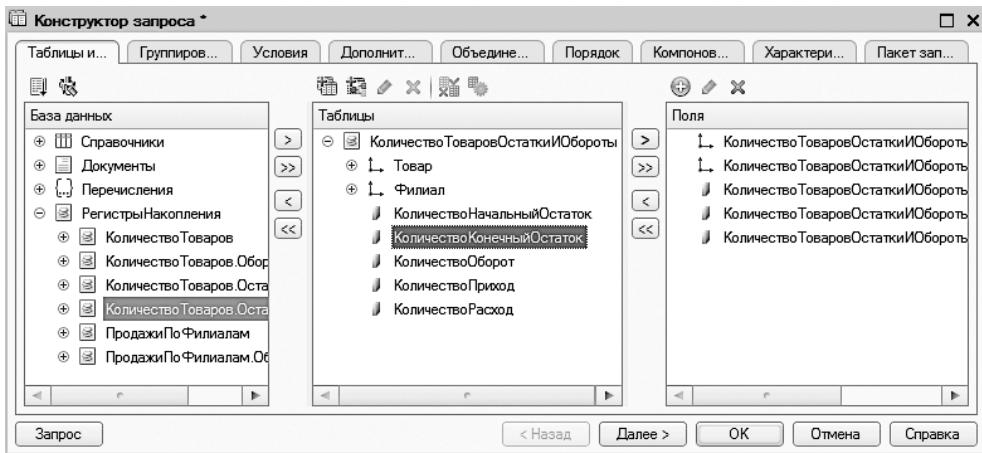


Рис. 3.38. Окно конструктора запроса

Больше никаких действий в окне конструктора запроса выполнять не требуется, и после щелчка на кнопке OK система автоматически сформирует текст запроса (листинг 3.6).

Листинг 3.6. Запрос, обеспечивающий извлечение информации для объекта Отчет3_6

```
ВЫБРАТЬ
    КоличествоТоваров.ОстаткиИОбороты.КоличествоКонечныйОстаток,
    КоличествоТоваров.ОстаткиИОбороты.КоличествоПриход,
    КоличествоТоваров.ОстаткиИОбороты.КоличествоРасход,
    КоличествоТоваров.ОстаткиИОбороты.Товар,
    КоличествоТоваров.ОстаткиИОбороты.Филиал
ИЗ
    РегистрНакопления.КоличествоТоваров.ОстаткиИОбороты КАК
        КоличествоТоваров.ОстаткиИОбороты
```

Дальнейшие действия по разработке отчета будут производиться в окне конструктора схемы компоновки данных. Переходим на вкладку Ресурсы и перенесем в правую часть окна три поля в соответствии с рис. 3.39.

Далее перейдем на вкладку Настройки и для создания структуры отчета воспользуемся конструктором настроек. Для этого следует щелкнуть по кнопке Открыть конструктор настроек (рис. 3.40).

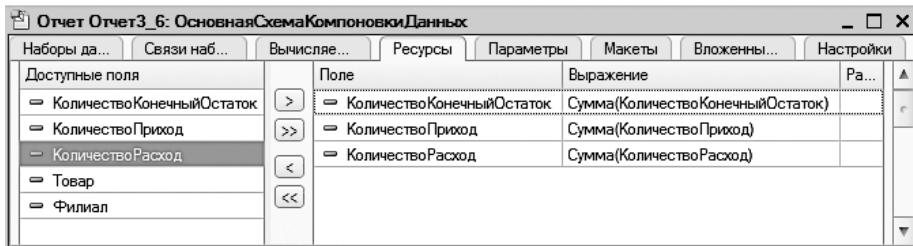


Рис. 3.39. Вкладка Ресурсы окна конструктора схемы компоновки данных

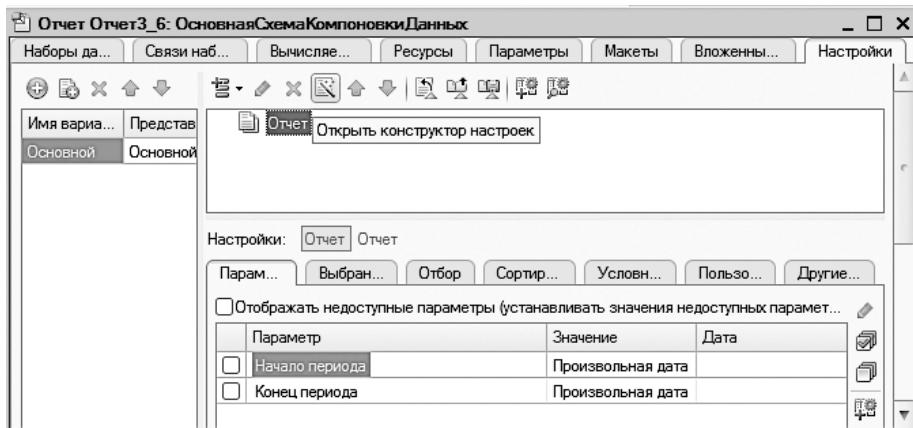


Рис. 3.40. Использование конструктора настроек для формирования отчета

Если охарактеризовать конструктор настроек в целом, то можно сказать, что данный инструмент позволяет быстро создать несколько вариантов отчетов. Сейчас нам понадобится вариант Список, поэтому отметим переключателем данный тип отчета и щелкнем на кнопке Далее (рис. 3.41).

В следующем окне выберем указанные поля для нашего отчета и также щелкнем на кнопке Далее (рис. 3.42).

Теперь следует указать поля, по которым будет сгруппирована информация в отчете. Как видно из рис. 3.43, это два поля. На этом работа с конструктором настроек завершается и можно щелкнуть на кнопке OK. В результате мы получим знакомую структуру отчета, но созданную с помощью конструктора настроек (рис. 3.44).

Осталось выполнить уже знакомую настройку (предоставление выбора значений в окне для пользователя) параметров отчета:

- Начало периода;
- Конец периода.

В результате мы получим отчет, который в режиме 1С:Предприятие мы уже видели (см. рис. 3.37).

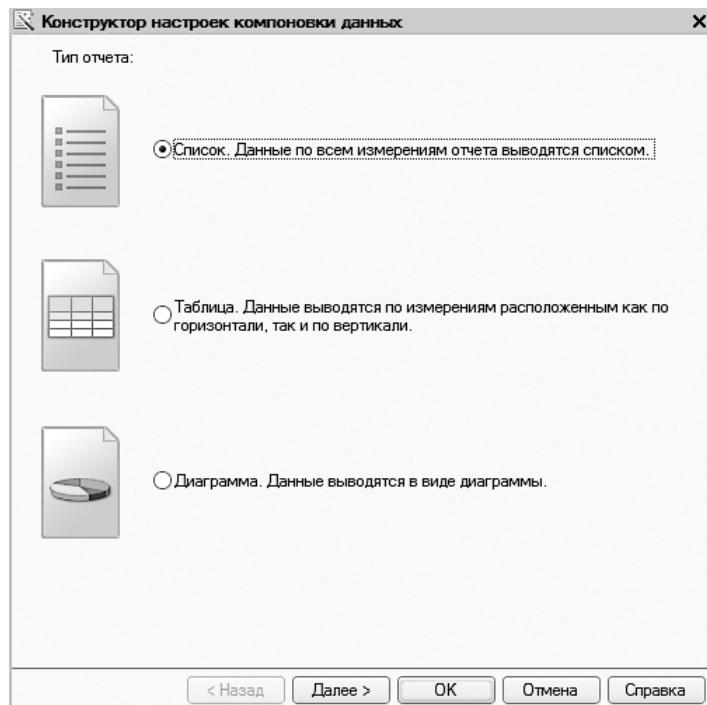


Рис. 3.41. Окно конструктора настроек компоновки данных отчета

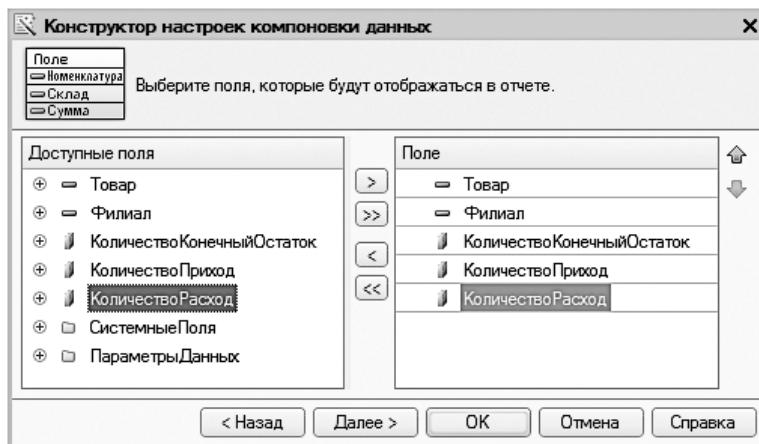


Рис. 3.42. Отбор полей для отчета в конструкторе настроек

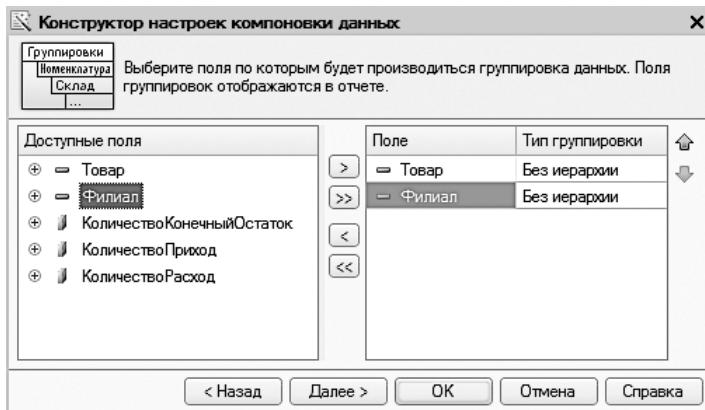


Рис. 3.43. Выбор полей для группировки в отчете

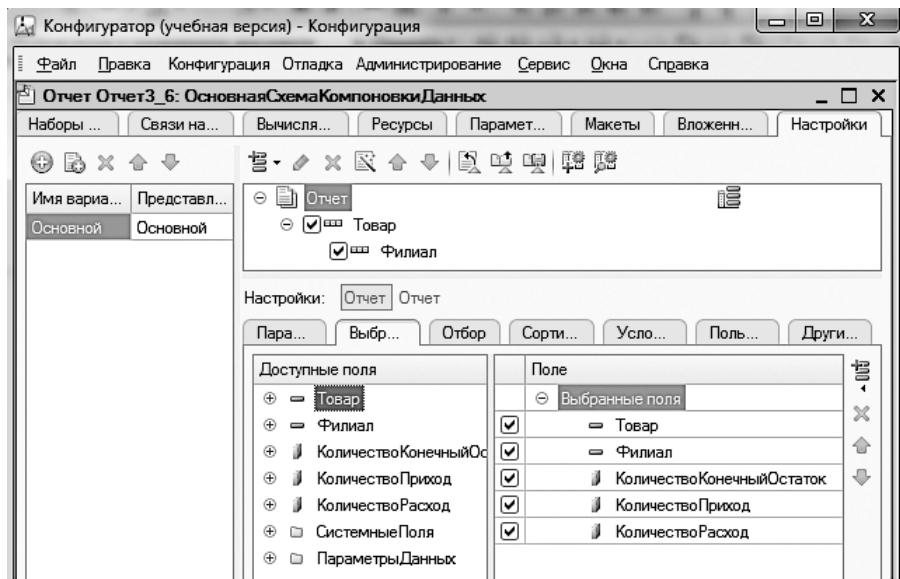


Рис. 3.44. Вкладка Настройки окна конструктора схемы компоновки данных

Отчеты с графиками

Рассмотрим пример еще одного отчета (Отчет3_7), который будет показывать результат отбора информации на графике. Для создаваемого отчета понадобится только одна виртуальная таблица — КоличествоТоваров.ОстаткиИОбороты (рис. 3.45).

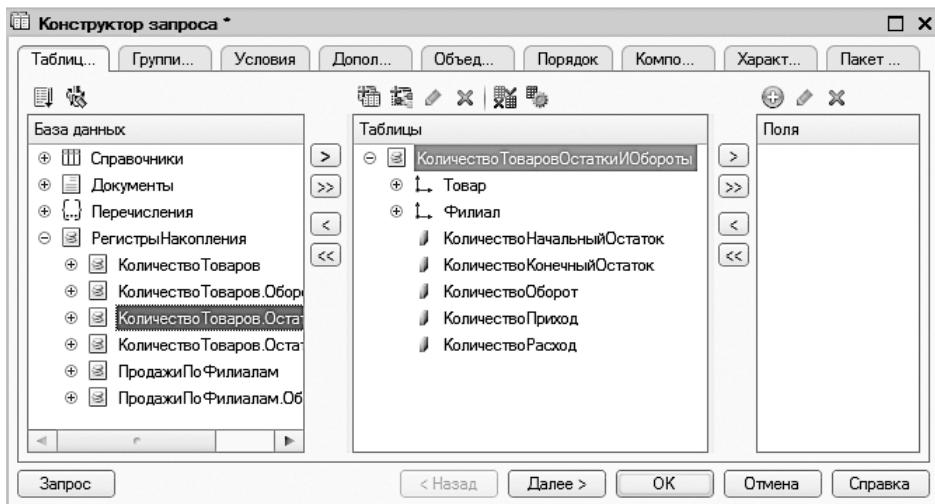


Рис. 3.45. Окно конструктора запроса

Здесь нам необходимо произвести дополнительную подготовку данных, для чего воспользуемся кнопкой Параметры виртуальной таблицы (на рис. 3.45 она располагается в пиктографическом меню над разделом Таблицы). В новом открывшемся окне в списке Периодичность нужно выбрать вариант День (рис. 3.46), после чего щелкнуть на кнопке OK.

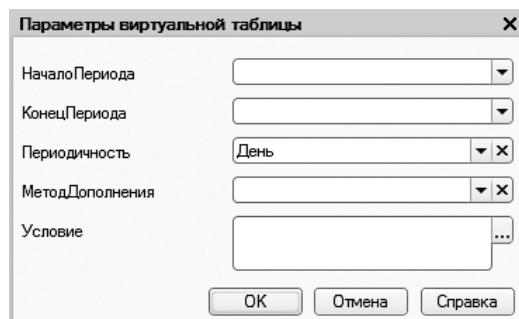


Рис. 3.46. Окно параметров виртуальной таблицы

Сформируем необходимые поля для выполняемого запроса (рис. 3.47).

После этого определим условие отбора товара при выполнении запроса (рис. 3.48).

После щелчка на кнопке OK на экране мы увидим автоматически сформированный текст запроса (листинг 3.7).

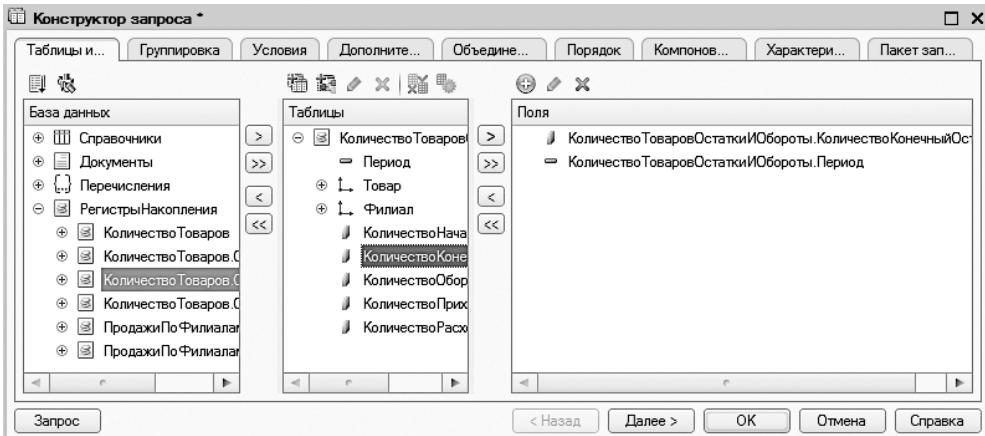


Рис. 3.47. Выбор полей для запроса

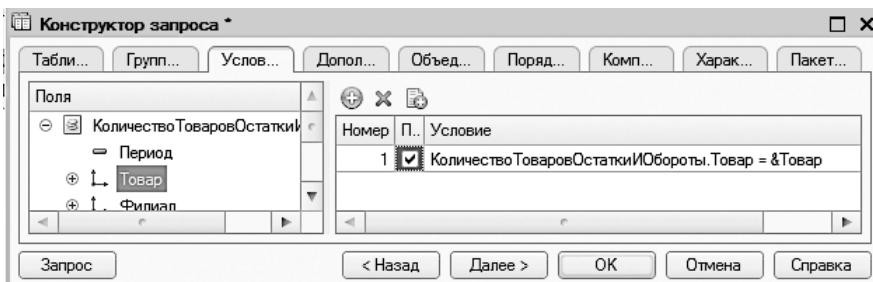


Рис. 3.48. Задание условия для запроса

Листинг 3.7. Запрос, обеспечивающий извлечение информации для объекта Отчет3_7

ВЫБРАТЬ

КоличествоТоваровОстаткиИОбороты.КоличествоКонечныйОстаток,
КоличествоТоваровОстаткиИОбороты.Период

ИЗ

РегистрНакопления.КоличествоТоваров.ОстаткиИОбороты(, , День, ,) КАК
КоличествоТоваровОстаткиИОбороты

ГДЕ

КоличествоТоваровОстаткиИОбороты.Товар = &Товар

Далее перейдем к работе с конструктором схемы компоновки данных и укажем на вкладке Ресурсы необходимый ресурс для отчета (рис. 3.49).

Воспользуемся, как и в предыдущем отчете, конструктором настроек, где укажем в качестве типа создаваемого отчета диаграмму (рис. 3.50).

В следующем окне определим поля диаграммы (рис. 3.51).

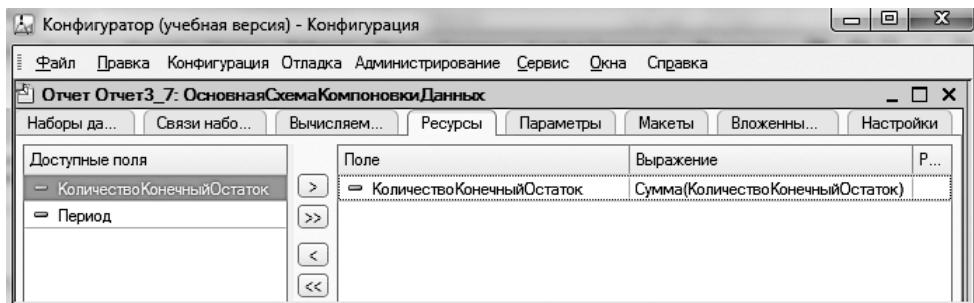


Рис. 3.49. Вкладка Ресурсы окна конструктора схемы компоновки данных

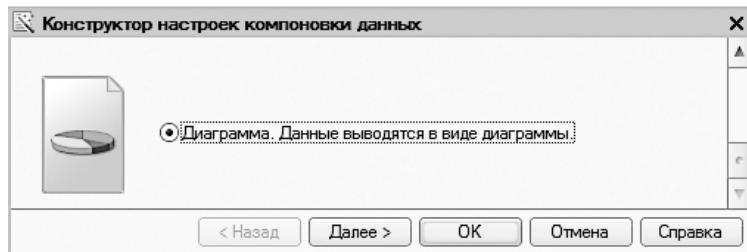


Рис. 3.50. Выбор диаграммы в окне конструктора настроек отчета

Осталось задать точки для диаграммы (рис. 3.52), параметры упорядочивания данных (рис. 3.53) и тип диаграммы (рис. 3.54).

В результате работы конструктора схемы компоновки данных параметры отчета будут соответствовать рис. 3.55.

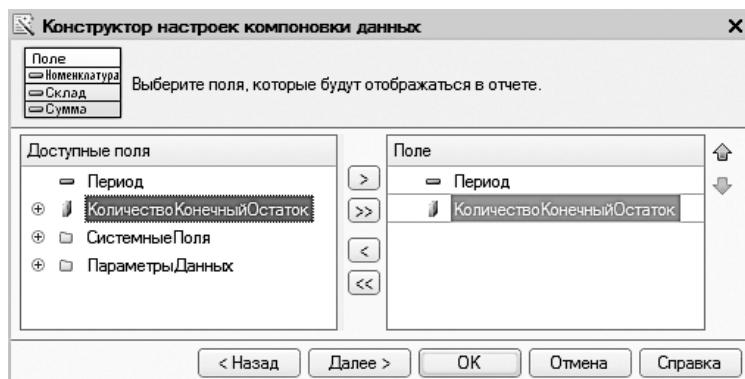


Рис. 3.51. Задание полей диаграммы

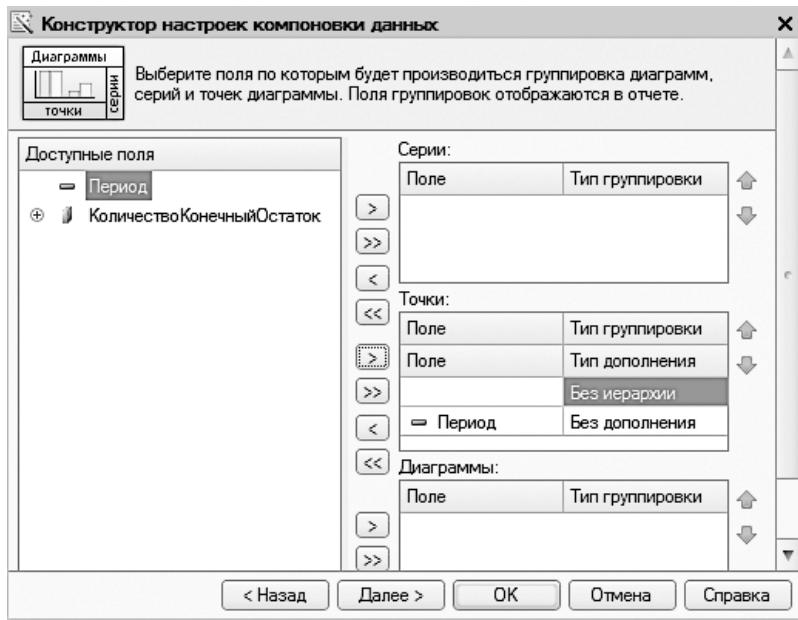


Рис. 3.52. Задание точек диаграммы

На завершающей стадии разработки необходимо включить в пользовательские настройки три параметра отчета:

- Начало периода;
- Конец периода;
- Товар.

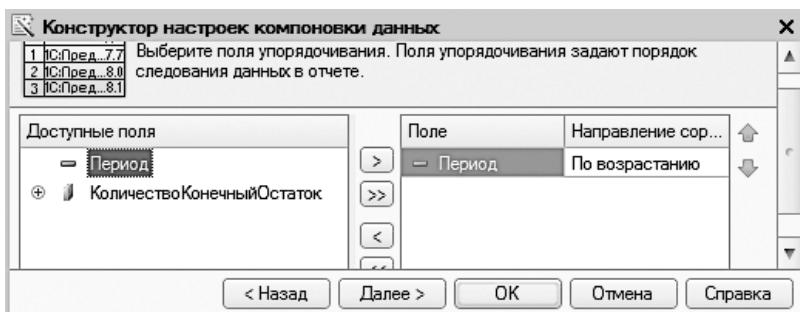


Рис. 3.53. Упорядочивание данных на графике

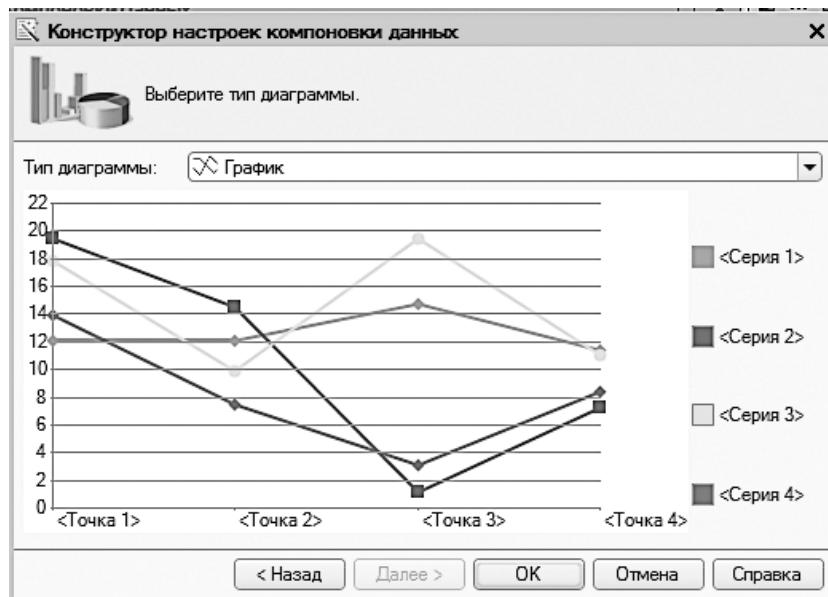


Рис. 3.54. Определение типа диаграммы

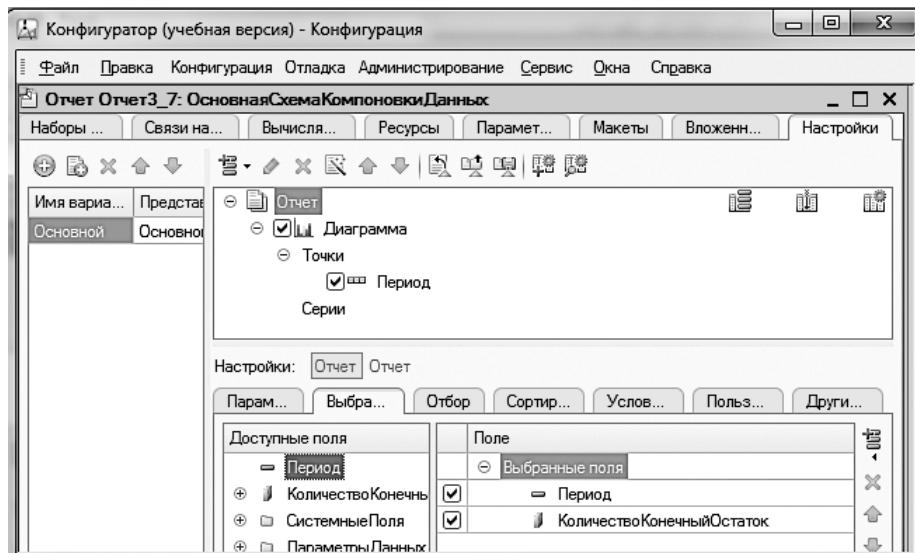


Рис. 3.55. Okno конструктора схемы компоновки данных

На этом конструкторские действия по разработке завершаются, и можно перейти в режим 1С:Предприятие для получения готового отчета (рис. 3.56).

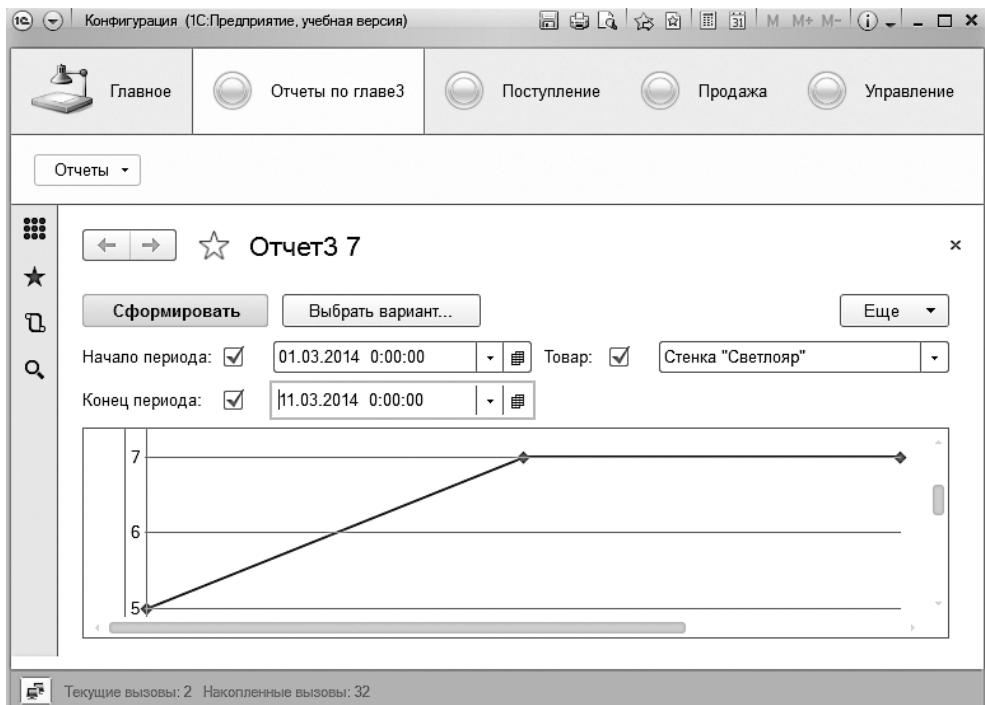


Рис. 3.56. Пример отчета с графиком

Итоги

В этой главе мы подробно на примерах познакомились с системой компоновки данных. Разнообразные задачи количественного учета продемонстрировали этот ключевой инструментальный ресурс, который используют разработчики для построения необходимых отчетов по базе данных.

Все примеры разработки описаны очень подробно, что должно быть удобно для читателей. Система компоновки данных отсутствовала в версии 1С:Предприятие 8.0, а ее появление, начиная с программного продукта 1С:Предприятие 8.1, дало пользователям возможность разрабатывать эффективные отчеты вообще без обращения к программному коду на языке 1С. В этом случае создание отчетов производится на основе декларативного описания. Механизм компоновки данных позволяет исключительно визуальными средствами получить целостное описание отчета.

Глава 4

Автоматизация при организации учета

В данной главе мы продолжим рассмотрение технических приемов учета информации в системе 1С:Предприятие 8.3. В качестве основы для информационной базы мы будем использовать конфигурацию, разработанную нами в первой главе. Фактически здесь мы доработаем эту конфигурацию в плане ее большего приближения к реальности.

Константы

Большое внимание в предыдущих главах было уделено справочникам. Эти объекты конфигурации позволяют хранить в информационной базе набор значений. Однако иногда требуется вносить в систему только одно значение. Например, это может быть название организации или процентная ставка определенного налога и т. д. Для подобных ситуаций в системе 1С:Предприятие имеется такой объект конфигурации, как **константа**.

Константа содержит только одно фиксированное значение. Считается, что это так называемая условно-постоянная информация — значение константы или совсем не меняется, или меняется крайне редко.

Перейдем к практическим действиям и посмотрим, как в системе 1С:Предприятие определять константы, присваивать им значения и задействовать эти значения в пользовательском режиме.

Итак, в режиме конфигуратора откроем конфигурацию, которая была разработана нами в первой главе. Далее в окне Конфигурация, показанном на рис. 4.1, щелкнем правой кнопкой мыши на пункте Константы и в открывшемся контекстном меню выберем пункт **Добавить**.

В результате на экране откроется уже знакомая нам палитра свойств (рис. 4.2). В этой палитре установим имя константы — **НазваниеОрганизации**. В качестве типа данных этой константы укажем вариант **Строка**, а длину символов в строке ограничим значением 50.

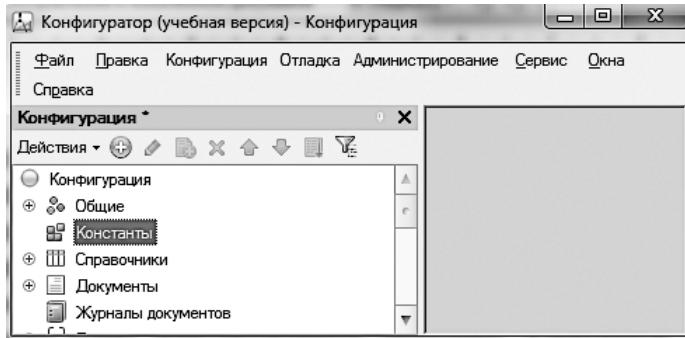


Рис. 4.1. Окно Конфигурация

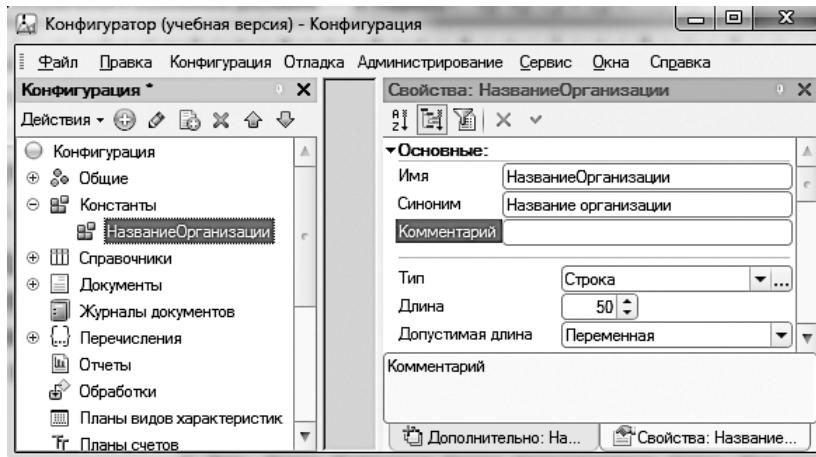


Рис. 4.2. Окно свойств константы НазваниеОрганизации

Теперь нам надо отнести созданную константу к одной из имеющихся у нас подсистем. Для этого следует обратиться к контекстному меню объекта конфигурации НазваниеОрганизации. В этом меню выберем пункт Дополнительно и путем установки соответствующего флажка отнесем созданную константу к подсистеме Управление (рис. 4.3).

После выполненных действий в режиме конфигуратора перейдем в режим 1С:Предприятие, где сначала выберем подсистему Управление, а затем обратимся к списку Сервис (рис. 4.4), где выберем конкретное значение константы НазваниеОрганизации (рис. 4.5).

Где вообще могут использоваться константы? Например, в печатных формах, с которыми мы уже познакомились. В принципе, программно можно обращаться к значению константы в любых алгоритмах.

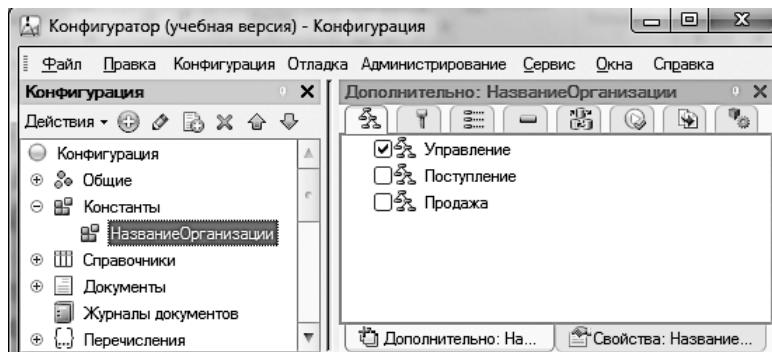


Рис. 4.3. Включение объекта конфигурации НазваниеОрганизации в подсистему Управление

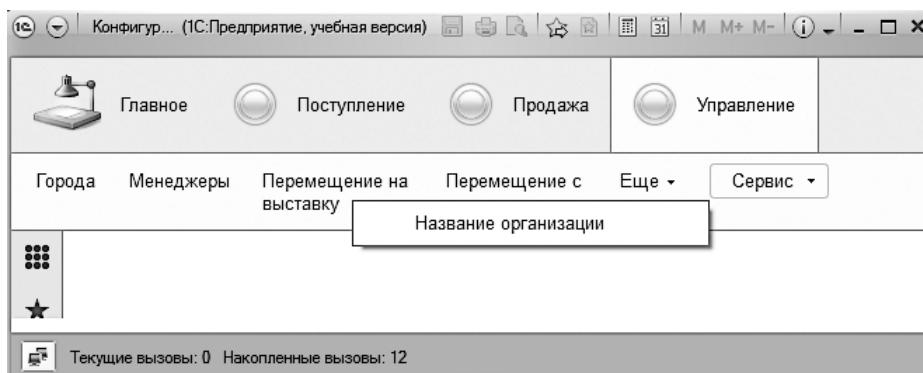


Рис. 4.4. Обращение к константе НазваниеОрганизации в режиме 1C:Предприятие

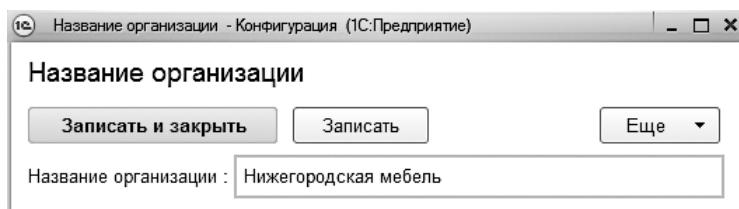


Рис. 4.5. Окно установки значения константы НазваниеОрганизации

Документ ЗаказКлиента

В предыдущих главах мы познакомились с разработкой и использованием документов. В системе 1С:Предприятие *документ* регистрирует факт некоторого события на предприятии. Самое главное свойство любого документа заключается

в том, что он имеет точное местоположение на временной оси. Рассмотрим на примере создания нового документа ряд технических приемов, которые мы до этого не использовали и которые могут помочь вам в реальной разработке прикладных решений на платформе 1С:Предприятие.

Итак, создадим новый объект конфигурации — документ, который назовем **ЗаказКлиента**. По структуре он будет похож на уже имеющийся в конфигурации документ **ПродажаТоваров**. Поэтому вместо знакомых технических действий по созданию нового документа мы поступим здесь более эффективно, а именно сделаем копию объекта конфигурации — документа **ПродажаТоваров**, после чего поменяем его название и немного подкорректируем.

Опишем теперь технические действия подробно. Для начала обратимся к знакомому окну **Конфигурация** и вызовем контекстное меню документа **ПродажаТоваров**, где просто воспользуемся пунктом **Скопировать** (рис. 4.6). В результате мы получим копию документа **ПродажаТоваров** под названием **ПродажаТоваров1** и далее внесем в него небольшие изменения. Для начала изменим название **ПродажаТоваров1** на **ЗаказКлиента**.

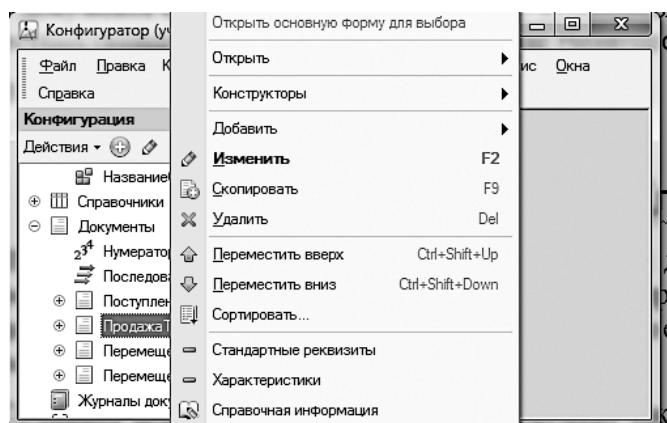


Рис. 4.6. Контекстное меню объекта конфигурации ПродажаТоваров

В качестве подсистемы для нового документа **ЗаказКлиента** укажем подсистему **Управление**, а на вкладке **Движения** очистим имеющуюся там процедуру проведения документа. В отличие от документа **ПродажаТоваров**, документ **ЗаказКлиента** не будет фиксировать факт поступления товаров (это пока только заказ).

Ввод на основании

Таким образом, в информационной базе теперь у нас есть два документа — **ЗаказКлиента** и **ПродажаТоваров**, которые по логике работы связаны между собой. Сначала

оформляется документ **ЗаказКлиента**, а затем осуществляется продажа товаров, которая оформляется документом **ПродажаТоваров**.

Чтобы упростить технические действия персонала при заполнении документов в режиме 1С:Предприятие, удобнее сделать так, чтобы документ **ПродажаТоваров** формировался автоматически на основании ранее созданного документа **ЗаказКлиента**.

Опишем технические действия для организации такого функционала. Обратимся к окну редактирования объекта конфигурации **ПродажаТоваров**, в котором перейдем на вкладку **Ввод на основании** (рис. 4.7). Здесь в верхней части окна располагается список **Вводится на основании**. В этот список можно включить объект, на основании которого будет вводиться документ **ПродажаТоваров**. С помощью кнопки справа над списком выберем в качестве основания документ **ЗаказКлиента** и щелкнем на кнопке **OK** (рис. 4.8).

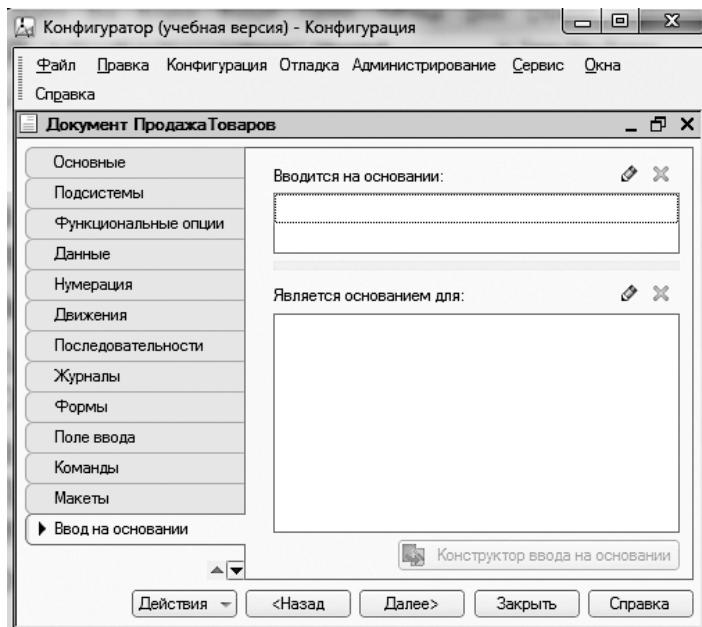


Рис. 4.7. Вкладка Ввод на основании документа ПродажаТоваров

В результате предыдущего шага мы переходим к следующему окну конструктора ввода на основании, в котором следует задать соответствие полей документа **ПродажаТоваров** и реквизитов документа-основания **ЗаказКлиента** (рис. 4.9).

Данное окно во многом напоминает окно конструктора движений, с которым мы уже сталкивались. С помощью кнопки **Заполнить выражения** оформим соответствие полей создаваемого документа реквизитам документа-основания (рис. 4.10).

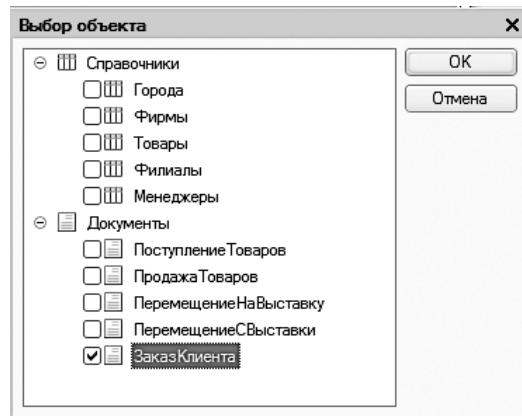


Рис. 4.8. Начальное окно конструктора ввода на основании

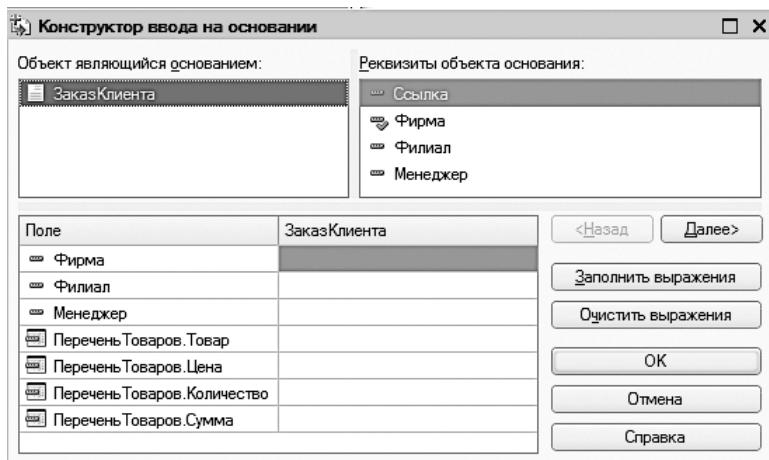


Рис. 4.9. Основное окно конструктора ввода на основании

Теперь, после щелчка по кнопке OK, можно получить процедуру заполнения полей документа ПродажаТоваров (листинг 4.1).

Листинг 4.1. Процедура заполнения полей документа ПродажаТоваров

```
Процедура ОбработкаЗаполнения(ДанныеЗаполнения, СтандартнаяОбработка)
//{{_КОНСТРУКТОР_ВВОД_НА_ОСНОВАНИИ
// Данный фрагмент построен конструктором.
Если ТипЗнч(ДанныеЗаполнения) = Тип(«ДокументСсылка.ЗаказКлиента») Тогда
    // Заполнение шапки
    Менеджер = ДанныеЗаполнения.Менеджер;
    Филиал = ДанныеЗаполнения.Филиал;
    Фирма = ДанныеЗаполнения.Фирма;
```

продолжение ⇨

Листинг 4.1 (продолжение)

```

Для Каждого ТекСтрокаПереченьТоваров Из ДанныеЗаполнения.ПереченьТоваров Цикл
    НоваяСтрока = ПереченьТоваров.Добавить();
    НоваяСтрока.Количество = ТекСтрокаПереченьТоваров.Количество;
    НоваяСтрока.Сумма = ТекСтрокаПереченьТоваров.Сумма;
    НоваяСтрока.Товар = ТекСтрокаПереченьТоваров.Товар;
    НоваяСтрока.Цена = ТекСтрокаПереченьТоваров.Цена;
КонецЦикла;
КонецЕсли;
//}}}_КОНСТРУКТОР_ВВОД_НА_ОСНОВАНИИ
КонецПроцедуры

```

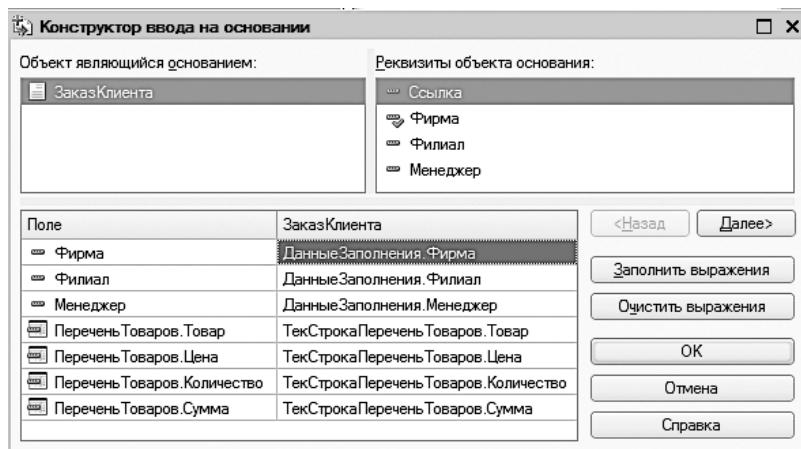


Рис. 4.10. Заполнение полей в основном окне конструктора ввода на основании

К настоящему моменту мы выполнили все технические действия по разработке документа ПродажаТоваров на основании документа ЗаказКлиента. Теперь мы можем перейти в режим 1С:Предприятие для проверки работоспособности заданного функционала. На рис. 4.11 показан созданный документ ЗаказКлиента, который фиксирует сделанный нашим клиентом заказ. После записи данного документа в форме списка мы можем воспользоваться кнопкой Создать на основании для формирования на основании заказа клиента нового документа ПродажаТоваров (рис. 4.12).

В результате мы получим автоматически заполненную форму документа ПродажаТоваров (рис. 4.13). Здесь при необходимости пользователь может подкорректировать количество единиц товаров и их цены. И далее уже документ ПродажаТоваров можно провести и тем самым зафиксировать в регистрах накопления необходимые движения.

Таким образом, мы рассмотрели технологию заполнения одних видов документов на основании других документов. Практически во всех типовых прикладных разработках такие приемы широко используются.

The screenshot shows the 1C:Enterprise application interface. At the top, there is a menu bar with icons for file operations like Open, Save, Print, and Exit. Below the menu is a toolbar with four main buttons: Главное (Main), Поступление (Arrival), Продажа (Sale), and Управление (Management). The main window title is 'Конфигурация (1С:Предприятие, учебная версия)'.

The current window is titled 'Заказ клиента 000000001 от 11.03.2014 15:53:57'. It contains several input fields: 'Фирма:' set to 'Верона', 'Филиал:' set to 'Автозаводский', and 'Менеджер:' set to 'Трунов А.М.'. There are also buttons for 'Провести и закрыть' (Post and Close), 'Записать' (Save), 'Провести' (Post), and 'Еще' (More). A 'Добавить' (Add) button is located below the input fields.

A table displays the order details:

N	Товар	Цена	Количество	Сумма
1	Прихожая "Уют"	17 000	3	51 000

At the bottom of the window, a status bar shows 'Текущие вызовы: 0 Накопленные вызовы: 52'.

Рис. 4.11. Форма документа ЗаказКлиента

This screenshot shows the same 1C:Enterprise application interface as the previous one, but it is displaying a list of client orders. The window title is 'Заказ клиента'.

The interface includes a toolbar with buttons for 'Создать' (Create), 'Найти...' (Find), 'Отменить поиск' (Cancel search), 'Создать на основании' (Create based on), and 'Еще' (More). The list view shows columns for 'Дата' (Date), 'Номер' (Number), 'Фирма' (Company), 'Филиал' (Branch), and 'Менеджер' (Manager). The first item in the list has the date '11.03.2014 15:53:57', number '000000001', company 'Верона', branch 'Автозаводский', and manager 'Трунов А.М.'.

At the bottom of the window, a status bar shows 'Текущие вызовы: 0 Накопленные вызовы: 53'.

Рис. 4.12. Форма списка документов ЗаказКлиента

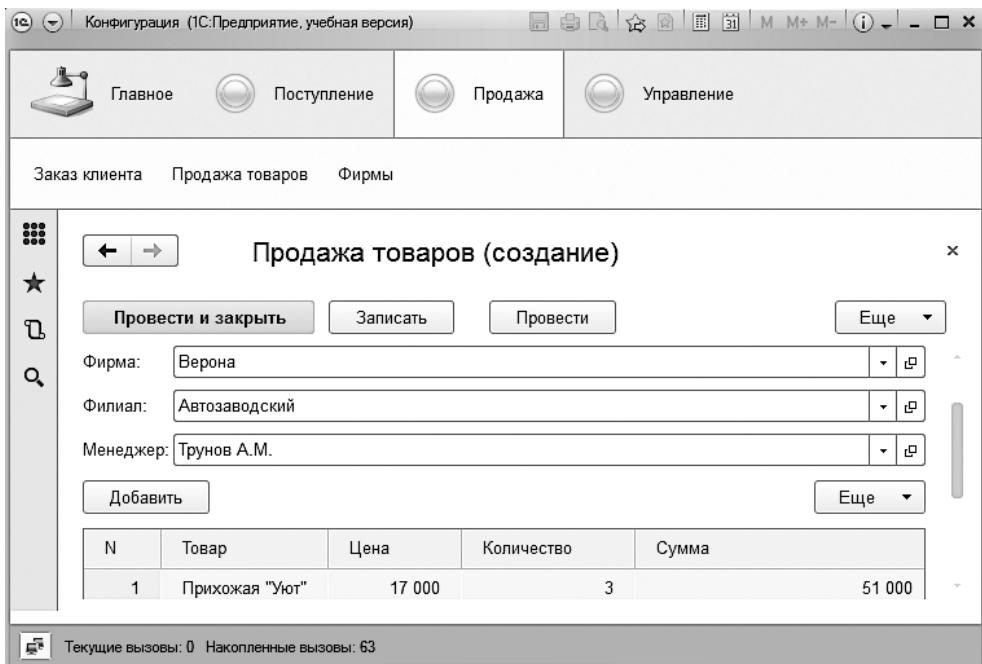


Рис. 4.13. Форма документа ПродажаТоваров

Журналы документов

К настоящему моменту мы создали несколько документов разного вида — несколько новых объектов конфигурации. В режиме 1С:Предприятие записанные документы автоматически формируются в виде списков. Так, мы видели формы списка документов Поступление Товаров, Продажа Товаров, Заказ Клиента и т. д. Однако бывают ситуации, когда документы разного вида удобнее хранить в одном месте. Для этого используются *журналы документов*.

В данном разделе мы рассмотрим, каким образом в системе можно создать новый журнал и поместить в него несколько документов разного вида, которые по логике работы относятся к одному направлению деятельности.

Начнем практические действия в окне Конфигурация, где вызовем контекстное меню раздела Журналы (рис. 4.14). Здесь мы создадим новый журнал, который назовем Торговля. Как уже стало привычным, укажем это имя журнала на вкладке Основные.

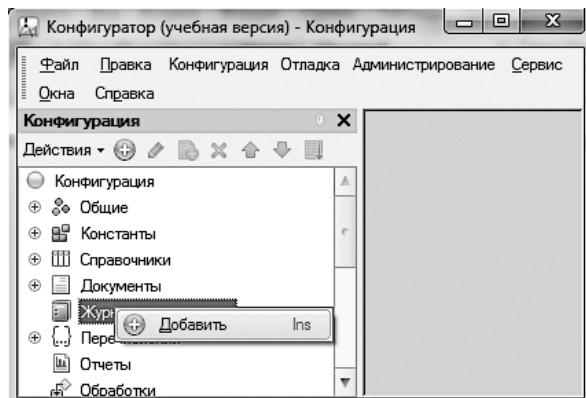


Рис. 4.14. Раздел Журналы в окне Конфигурация

Далее перейдем на вкладку **Данные**, где необходимо включить документы в создаваемый журнал (рис. 4.15). Для этого служит соответствующая пиктографическая кнопка в верхней части окна.

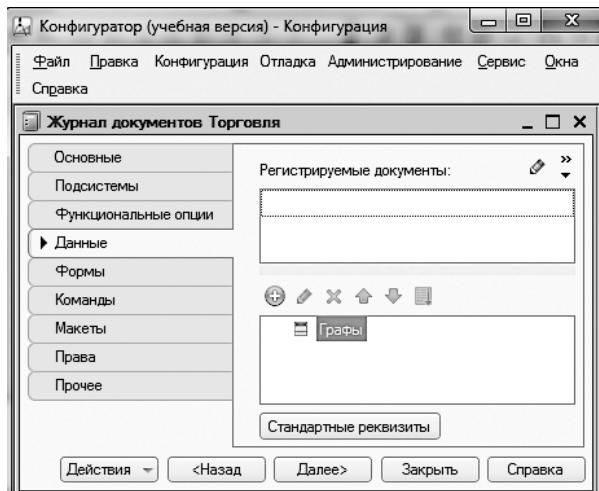


Рис. 4.15. Вкладка Данные окна редактирования журнала Торговля

В данной ситуации включим с помощью соответствующих флажков в журнал Торговля следующие наши документы (рис. 4.16):

- ЗаказКлиента;
- ПродажаТоваров.

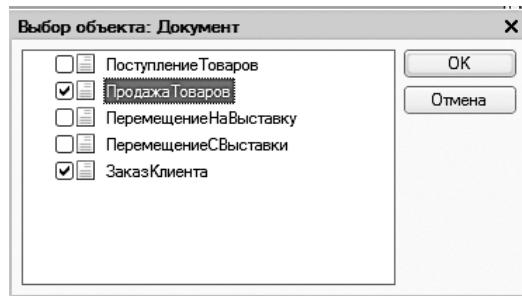


Рис. 4.16. Окно для включения документов в журнал

Отнесем созданный журнал к подсистеме Управление, после чего можно посмотреть, как выглядит этот журнал в пользовательском режиме (рис. 4.17).

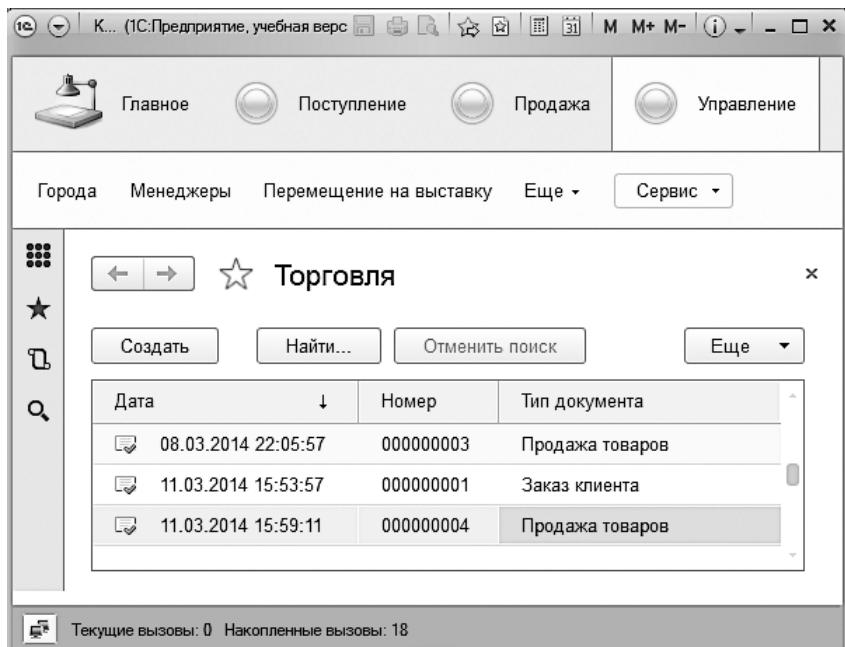


Рис. 4.17. Перечень документов в журнале Торговля

Как видно из рисунка, в созданном журнале отражаются необходимые торговые документы. Однако для информативности данного списка было бы неплохо показывать в нем и информацию о клиенте (кто данный заказ сделал и кто его оплатил в соответствии с документом продажи товаров). Для этого в журналах документов можно создавать дополнительные графы с помощью вкладки Данные (рис. 4.18).

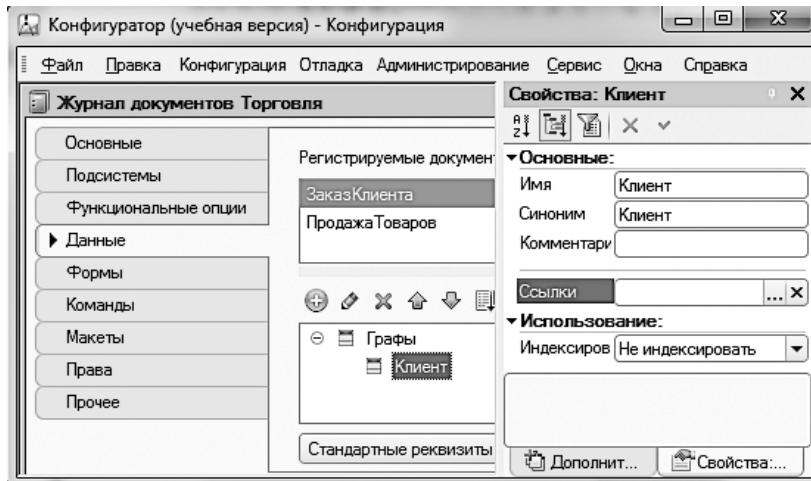


Рис. 4.18. Создание дополнительной графы в журнале Торговля

Сейчас мы создадим графу с именем Клиент. После ввода имени графы в окне панели свойств необходимо щелкнуть в поле Ссылки. В этом случае на экране появляется новое окно (рис. 4.19). В нем следует указать те реквизиты документов, которые мы собираемся отображать в созданной графе. В данном случае в обоих документах отметим реквизит Фирма.

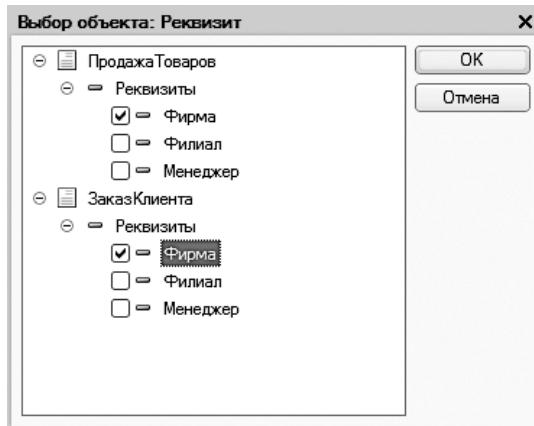


Рис. 4.19. Включения реквизитов для отображения в графе Клиент

Теперь уже можно перейти в пользовательский режим и убедиться, что в нашем журнале напротив каждого документа в соседнем поле отображается фирма-клиент, которая данный документ инициировала (рис. 4.20).

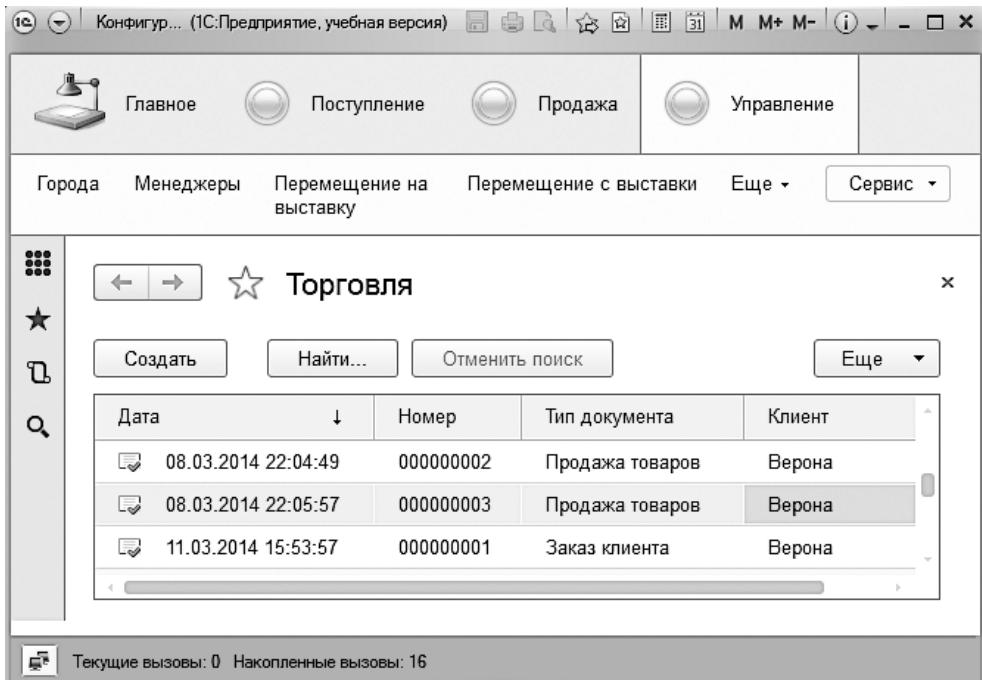


Рис. 4.20. Отображение графы Клиент в журнале

Коррекция документов ПоступлениеТоваров и ПродажаТоваров

Перейдем к следующему этапу разработки и создадим механизм взаиморасчетов с контрагентами — поставщиками и покупателями. Для этого добавим в разработку механизм учета денежных средств (оплата клиента нам и наша оплата поставщикам товаров). В этом случае у нас всегда будет возможность узнать нашу задолженность перед поставщиками и задолженность наших покупателей перед нами.

В качестве первого шага выполним небольшую доработку двух имеющихся у нас документов. В документах ПоступлениеТоваров и ПродажаТоваров отсутствует поле, информирующее сотрудника нашей организации об общей сумме по документу (отсутствует суммирование по перечню товаров с учетом их количества и цены за единицу товара). Поэтому последовательно добавим реквизит СуммаПодокументу в каждый из рассматриваемых документов.

Итак, в режиме конфигуратора обратимся к окну редактирования документа ПоступлениеТоваров и добавим реквизит СуммаПодокументу целого типа (рис. 4.21). Далее

этот реквизит необходимо добавить в уже имеющуюся в конфигурации форму документа. Технически это выполняется в окне редактирования формы перетаскиванием мышью реквизита СуммаПоДокументу на элемент Командная панель (рис. 4.22).

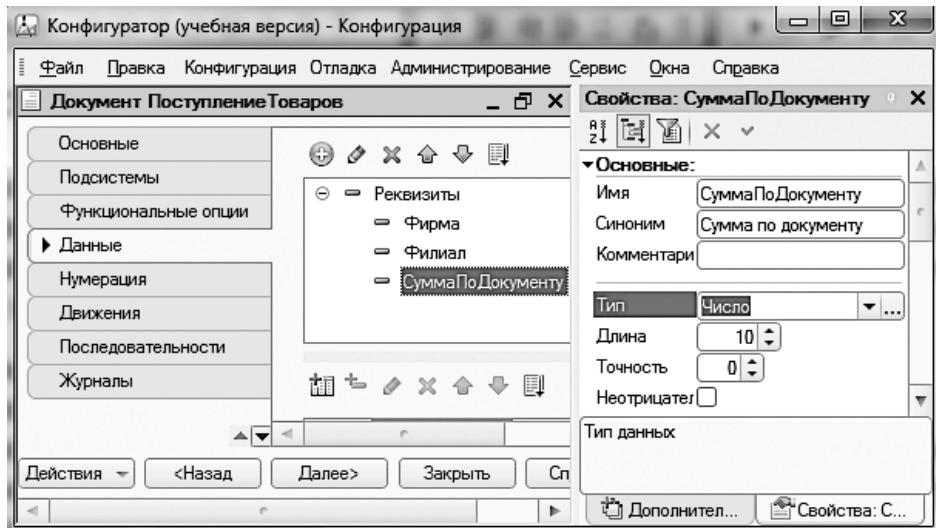


Рис. 4.21. Добавление реквизита в документ ПоступлениеТоваров

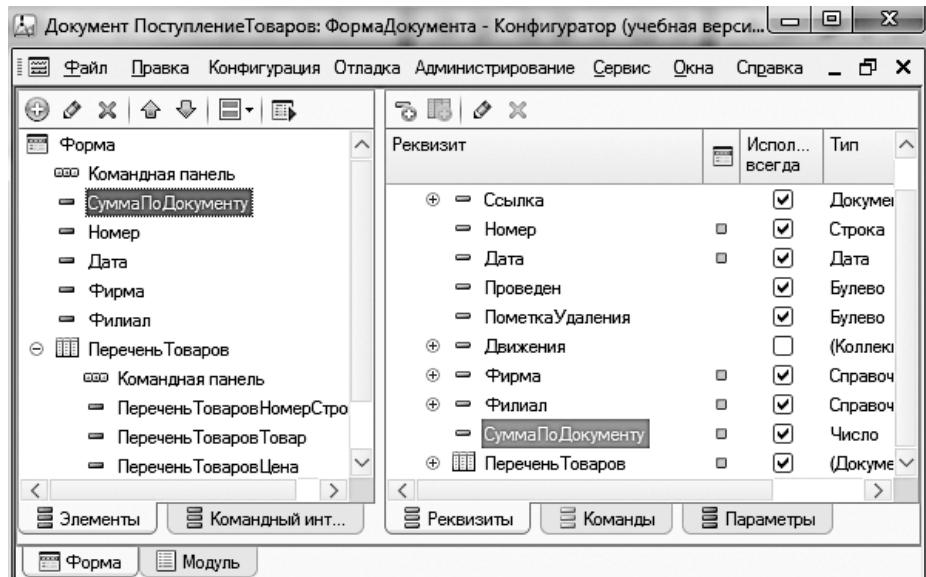


Рис. 4.22. Окно редактирования формы документа ПоступлениеТоваров

Теперь можно перейти в режим 1С:Предприятие и в созданных документах Поступление Товаров заполнить поле Сумма Подокументу. Недостатком нашей разработки на данный момент является необходимость ручного подсчета суммы. Пока оставим все как есть, а далее, в заключительном разделе этой главы, мы реализуем автоматическое заполнение рассматриваемого поля по табличной части документа.

Аналогичные действия по созданию реквизита Сумма Подокументу и включения его в форму следует проделать и с документом Продажа Товаров.

Регистр Расчеты С Firmами

Перейдем к следующему шагу, который заключается в разработке регистра Расчеты С Firmами. На вкладке Основные окна редактирования объекта конфигурации внесем имя регистра и укажем, что он будет являться регистром остатков. После этого перейдем на вкладку Данные (рис. 4.23), где определим измерение данного регистра (Фирма — тип данных СправочникСсылка . Фирмы) и его ресурс (Сумма — числовой тип данных). Что касается измерения Фирма, оно не требует особых комментариев. Это та фирма, с которой мы вычисляем взаиморасчеты.

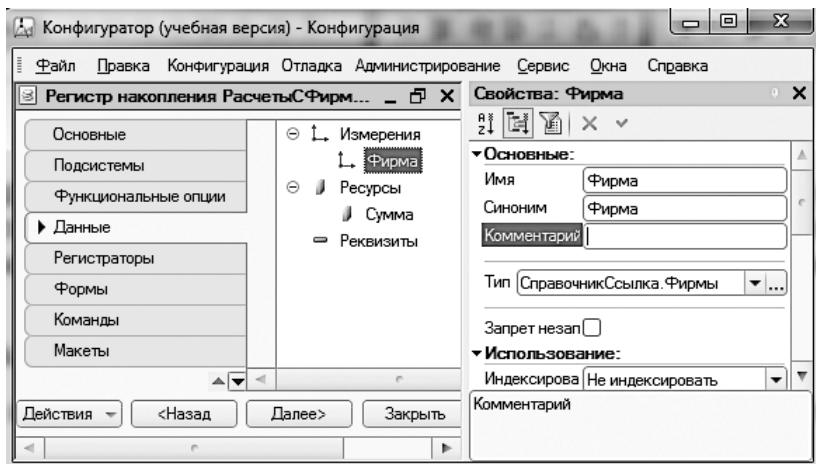


Рис. 4.23. Вкладка Данные окна редактирования регистра Расчеты С Firmами

Что же касается поля Сумма, то здесь следует дать небольшое пояснение. В этом поле будет храниться наш долг перед поставщиками (если мы получили товар, но пока не заплатили) и долг наших покупателей перед нами (если не оплачена вся сумма за купленный у нас товар). Итак, определим, что ресурс Сумма — это *долг покупателя перед нами*.

ПРИМЕЧАНИЕ

Если бы в задаче требовалось учитывать долг не только в разрезе фирм, но и с учетом конкретных договоров, заключенных с этими фирмами, то в регистре РасчетыСФирмами необходимо было бы создать два измерения — Фирма и Договор.

Поясним теперь, каким образом в регистре РасчетыСФирмами будут формироваться движения. Рассмотрим сначала документ ПродажаТоваров. Допустим, мы продаем товары определенной фирме-покупателю на сумму 100 000 рублей. В этом случае долг фирмы-покупателя перед нами увеличивается на 100 000 рублей. Фирма может сразу же погасить долг, внеся деньги в кассу или сделав необходимый перевод на наш расчетный счет. Возможна также ситуация, когда фирма-покупатель оплачивает сделанную покупку спустя некоторое время (это может быть даже прописано в договоре с данной фирмой). Но в любом случае при проведении документа ПродажаТоваров образуется долг покупателя перед нами. В этом случае по регистру РасчетыСФирмами формируется движение Приход на сумму 100 000 рублей.

Далее через некоторое время фирма-покупатель частично оплачивает сумму покупки. Скажем, пусть эта фирма оплачивает 70 000 рублей. В этом случае по регистру РасчетыСФирмами формируется движение Расход на сумму 70 000 рублей. И в системе сформируется остаток долга по данному покупателю, который в этом случае равен 30 000 рублей.

Аналогичным образом разберем ситуацию с поступлением товаров. Пусть в определенный момент форма-поставщик поставляет нам товар на 50 000 рублей. В результате увеличивается наш долг перед этим поставщиком. Таким образом, в документе ПоступлениеТоваров мы должны сделать движение Расход на 50 000 рублей. Если после этого посмотреть на остатки по данному регистру, то мы увидим в нем отрицательный остаток в -50 000 рублей. Через некоторое время мы частично гасим наш долг перед поставщиком, перечисляя ему 35 000 рублей. В этом случае будет сформировано движение Приход на 35 000 рублей. И по результатам этих двух действий по данному поставщику будет сформирован остаток в -15 000 рублей. Это означает, что данному поставщику мы должны 15 000 рублей. Если спустя еще какое-то время мы сделаем оплату данному поставщику на сумму 20 000 рублей, то остаток в регистре по данному поставщику станет равен 5000 рублей (остаток теперь положительный, так как в данном случае уже поставщик должен нам).

Вернемся к конструированию нашего регистра РасчетыСФирмами. На вкладке Регистраторы отметим флагками документы, которые будут обеспечивать движения по данному регистру (рис. 4.24). Это документы, о которых мы уже говорили.

Это был очередной шаг разработки и далее мы переходим к следующему. Нам надо обеспечить движения по регистру РасчетыСФирмами. Для этого в окне редактиро-

вания документа ПоступлениеТоваров перейдем на вкладку Движения и обеспечим движение по регистру РасчетыСФирмами.

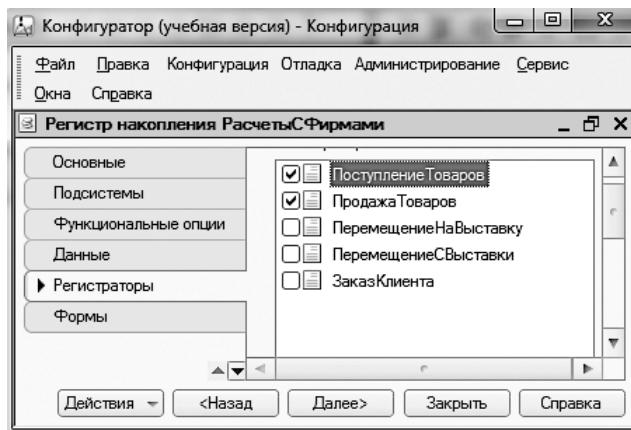


Рис. 4.24. Вкладка Регистраторы окна редактирования регистра РасчетыСФирмами

Поступление товаров приводит к увеличению нашего долга перед поставщиками. Поэтому обратим внимание на то, что в качестве типа движения в данном случае указано — Расход (рис. 4.25). Как и ранее, щелчком на кнопке OK сформируем процедуру проведения документа (листинг 4.2). Обратим внимание на то, что в данном случае проводки будут осуществляться по двум регистрам.

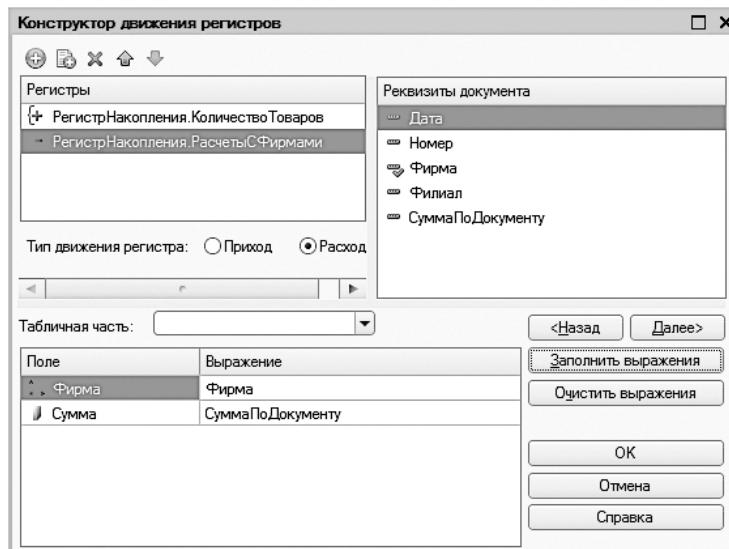


Рис. 4.25. Окно конструктора движения регистра

Листинг 4.2. Процедура проведения документа Поступление Товаров

```
Процедура ОбработкаПроведения(Отказ, Режим)
// регистр Количество Товаров Приход
Движения.Количество Товаров.Записывать = Истина;
Для Каждого ТекСтрока Перечень Товаров Из Перечень Товаров Цикл
    Движение = Движения.Количество Товаров.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    Движение.Период = Дата;
    Движение.Товар = ТекСтрока Перечень Товаров.Товар;
    Движение.Филиал = Филиал;
    Движение.Количество = ТекСтрока Перечень Товаров.Количество;
КонецЦикла;
// регистр РасчетыС фирмами Расход
Движения.РасчетыС фирмами.Записывать = Истина;
Движение = Движения.РасчетыС фирмами.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Фирма = Фирма;
Движение.Сумма = Сумма Подокументу;
КонецПроцедуры
```

После выполненных изменений перейдем в режим 1С:Предприятие и заново проведем уже созданные документы Поступление Товаров.

Чтобы убедиться в выполненных проводках по регистру РасчетыС фирмами, откроем его форму. Для этого необходимо обратиться к главному меню, которое вызывается щелчком мыши на кнопке со стрелкой, расположенной в левом верхнем углу окна приложения (рис. 4.26). Далее надо обратиться к пункту Все функции, где указать интересующий нас регистр. В результате мы увидим движения, выполненные документом Поступление Товаров.

Перейдем теперь к корректировке документа Продажа Товаров. Откроем в режиме конфигуратора окно редактирования данного документа и обратимся к вкладке Движения. Там уже определены проводки по двум регистрам и нам надо добавить проводку по регистру РасчетыС фирмами. В результате документ Продажа Товаров будет формировать проводки по трем регистрам накопления.

Продажа товаров приводит к увеличению долга покупателей перед нами (сумму, на которую куплен товар, необходимо оплатить). Поэтому обратим внимание на то, что в данном случае в качестве типа движения регистра РасчетыС фирмами указан вариант Приход (рис. 4.27).

В результате действий конструктора мы получим новую процедуру проведения документа Продажа Товаров (листинг 4.3).

Листинг 4.3. Процедура проведения документа Продажа Товаров

```
Процедура ОбработкаПроведения(Отказ, Режим)
// регистр Количество Товаров Расход
```

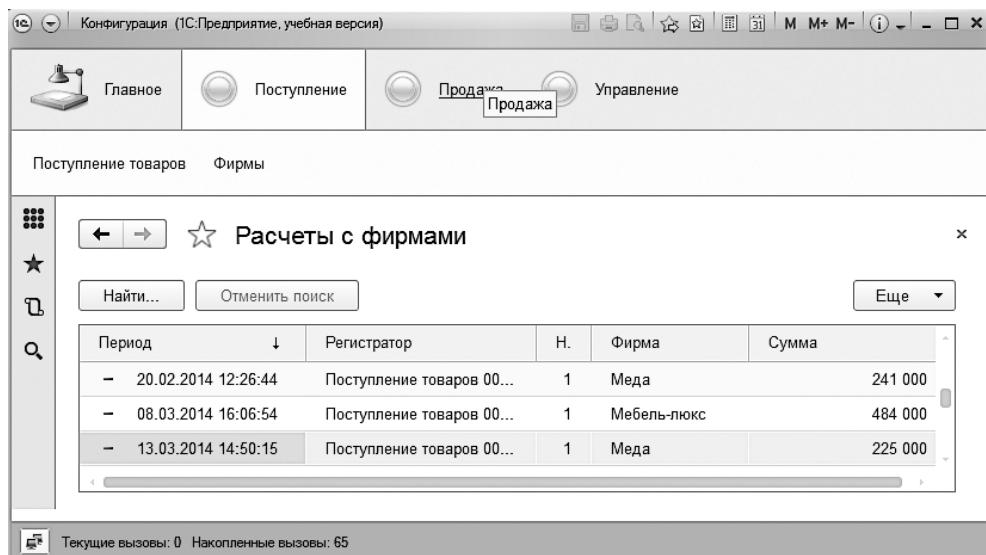
продолжение ⇨

Листинг 4.3 (продолжение)

```

Движения.КоличествоТоваров.Записывать = Истина;
Для Каждого ТекСтрокаПереченьТоваров Из ПереченьТоваров Цикл
    Движение = Движения.КоличествоТоваров.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Товар = ТекСтрокаПереченьТоваров.Товар;
    Движение.Филиал = Филиал;
    Движение.Количество = ТекСтрокаПереченьТоваров.Количество;
КонецЦикла;
// регистр ПродажиПоФилиалам
Движения.ПродажиПоФилиалам.Записывать = Истина;
Для Каждого ТекСтрокаПереченьТоваров Из ПереченьТоваров Цикл
    Движение = Движения.ПродажиПоФилиалам.Добавить();
    Движение.Период = Дата;
    Движение.Филиал = Филиал;
    Движение.Менеджер = Менеджер;
    Движение.Сумма = ТекСтрокаПереченьТоваров.Сумма;
КонецЦикла;
// регистр РасчетыСФирмами Приход
Движения.РасчетыСФирмами.Записывать = Истина;
Движение = Движения.РасчетыСФирмами.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
Движение.Период = Дата;
Движение.Фирма = Фирма;
Движение.Сумма = СуммаПодокументу;
КонецПроцедуры

```

**Рис. 4.26.** Форма списка регистра РасчетыСФирмами

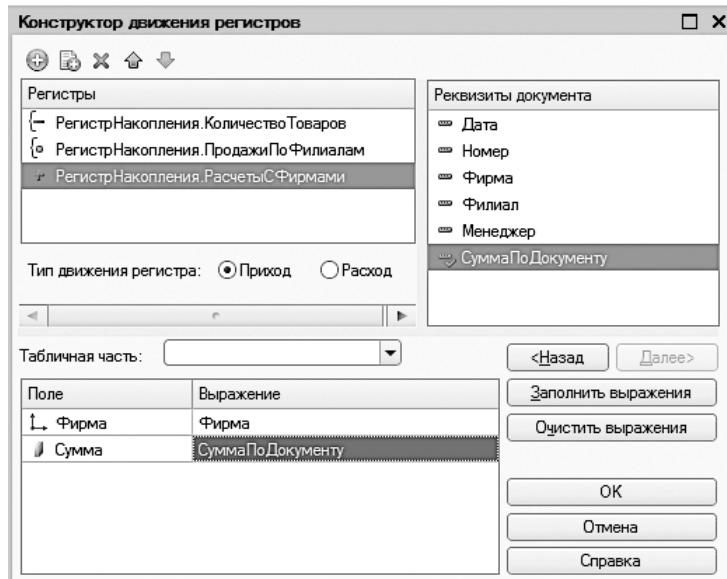


Рис. 4.27. Окно конструктора движения регистра

Теперь можно перейти в режим 1С:Предприятие и заново провести имеющиеся документы по продаже товаров. На рис. 4.28 показана форма списка регистра РасчетыСФирмами с проводками по различным документам.

Период	Registrator	Ном...	Фирма	Сумма
- 08.03.2014 16:06:54	Поступление товаров 00000...	1	Мебель-люкс	484 000
+ 08.03.2014 22:04:49	Продажа товаров 0000000...	1	Верона	153 000
+ 08.03.2014 22:06:57	Продажа товаров 0000000...	1	Верона	33 000

Рис. 4.28. Форма списка регистра РасчетыСФирмами

Сделаем еще небольшую доработку. Так, обеспечим сервис для просмотра движений по регистру **РасчетыС фирмами** из формы документа **ПродажаТоваров**. Для этого в окне редактирования объекта **ПродажиТоваров** перейдем на вкладку **Формы** и откроем форму документа. Далее обратимся к дополнительной вкладке **Командный интерфейс**. Здесь нас интересует пункт **Перейти** в разделе **Панель навигации** — для регистра **РасчетыС фирмами** необходимо установить флажок **Видимость**, как показано на рис. 4.29.

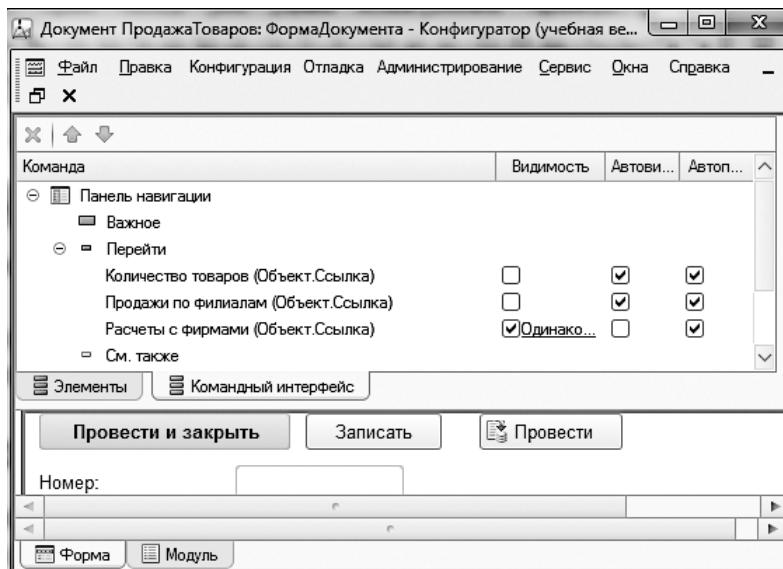


Рис. 4.29. Вкладка Командный интерфейс формы документа ПродажаТоваров

Теперь перейдем в режим 1С:Предприятие и посмотрим, к чему привели наши действия. Так, если в списке документов выбрать и открыть конкретный документ **ПродажаТоваров**, то мы увидим ссылку **Расчеты с фирмами**, позволяющую просмотреть движения по регистру **РасчетыС фирмами** (рис. 4.30).

Документ ВыпискаБанка

К настоящему моменту в нашей информационной системе организована регистрация документов по поступлению товаров и их продаже. Документ **ПродажаТоваров** добавляет ресурс регистра **РасчетыС фирмами**. Фактически задолженность организаций перед нами фиксируется со знаком «плюс». А при поставке (документ **Поступление Товаров**) наша задолженность уменьшает ресурс регистра и фактически фиксируется со знаком «минус».

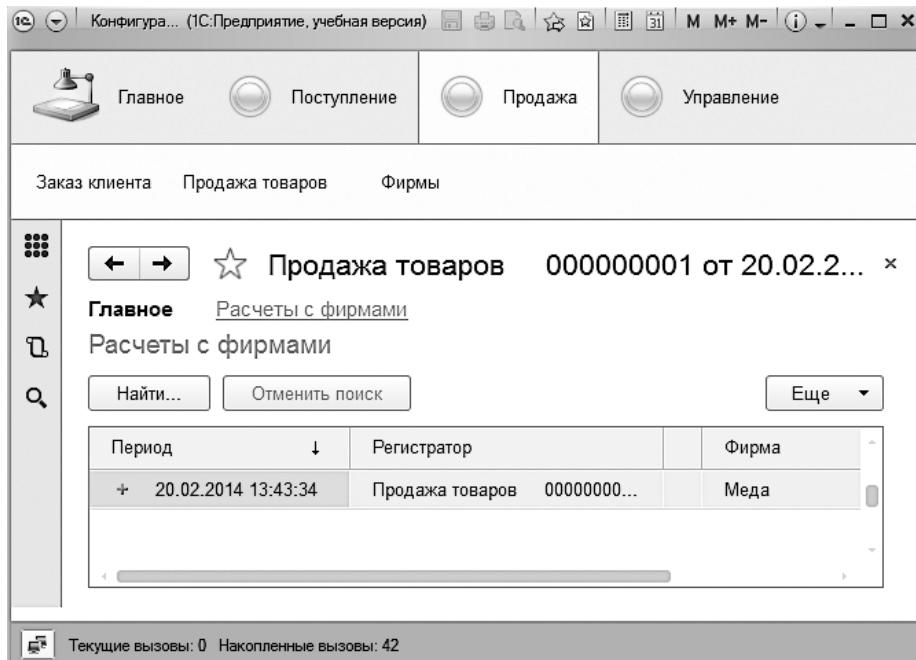


Рис. 4.30. Добавление в форму документа ссылки для перехода к регистру

Теперь перейдем к процессу погашения этих задолженностей.

В хозяйственной деятельности существует реальный документ «Выписка банка». В нем фиксируются платежи и поступления по банку за определенный день. Это могут быть входящие платежи, которыми наши покупатели оплачивают сделанные у нас покупки, а также исходящие платежи, когда уже мы платим поставщикам за поставленный от них товар.

Итак, создадим документ ВыпискаБанка. На рис. 4.31 показана вкладка **Данные**, где нам нужно создать несколько элементов. Так, в данном документе разместим две табличные части:

- ПриходСредств** — регистрация поступающих денежных средств на наш счет в банке;
- РасходСредств** — регистрация наших денежных платежей поставщикам с нашего счета в банке.

В каждой из этих табличных частей будут одинаковые реквизиты:

- Фирма** — для фиксирования фирмы-контрагента (тип данных — Справочник-Ссылка.Фирмы);
- Сумма** — для внесения суммы платежа (тип данных — число).

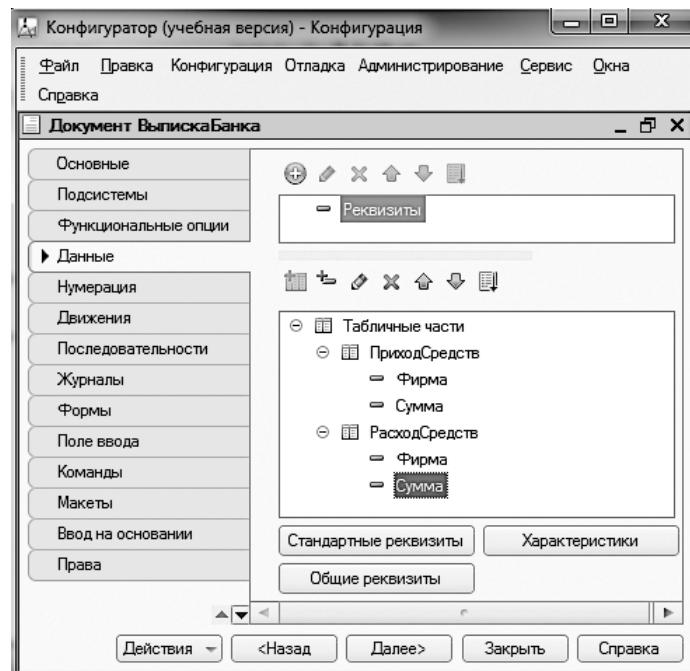


Рис. 4.31. Вкладка Данные окна редактирования документа ВыпискаБанка

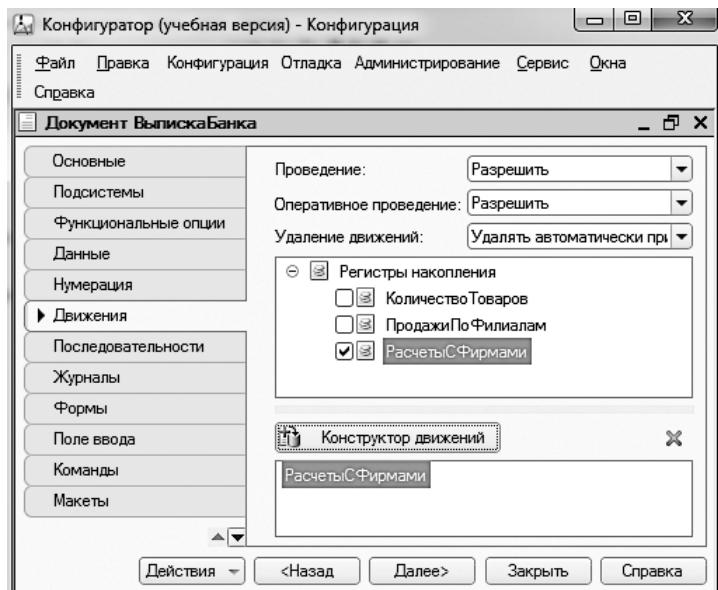


Рис. 4.32. Вкладка Движения окна редактирования документа ВыпискаБанка

После внесения этих реквизитов перейдем на вкладку Движения для формирования движений по регистру РасчетыСФирмами (рис. 4.32). Здесь воспользуемся кнопкой Конструктор движений.

В нашем документе две табличные части, поэтому мы должны организовать движения по регистру в каждой табличной части.

На рис. 4.33 показано формирование полей табличной части ПриходСредств. В данном случае нам поступили денежные средства, например, фирма-покупатель перечислила деньги за покупку. В результате долг покупателя перед нами уменьшается. В связи с этим должен быть установлен переключатель Расход.

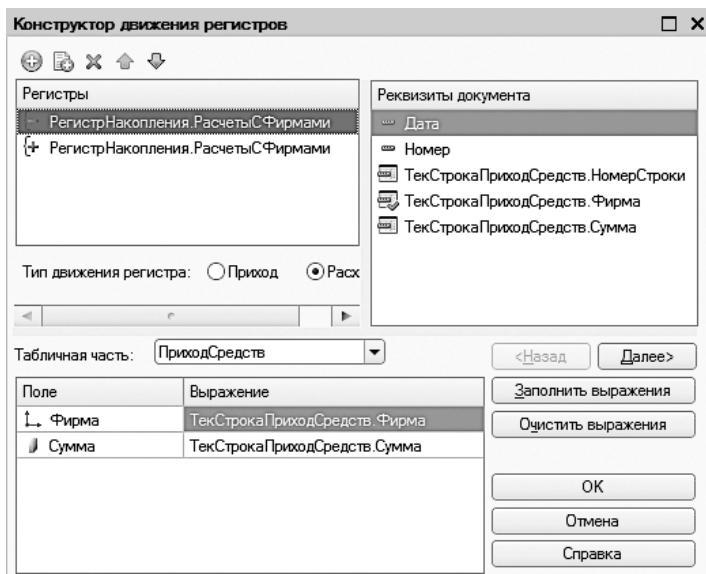


Рис. 4.33. Формирование полей для таблицы ПриходСредств

Добавим еще одно движение по тому же регистру. На рис. 4.34 показано формирование полей табличной части РасходСредств. Теперь денежные средства поступают от нас. В результате долг клиента перед нами увеличивается. В связи с этим должен быть установлен переключатель Приход.

Теперь щелчком на кнопке OK можно создать процедуру, которая и будет формировать необходимые проводки (листинг 4.4).

Листинг 4.4. Процедура проведения документа ВыпискаБанка

```
Процедура ОбработкаПроведения(Отказ, Режим)
// регистр РасчетыСФирмами Расход
Движения.РасчетыСФирмами.Записывать = Истина;
```

продолжение ↗

Листинг 4.4 (продолжение)

```

Для Каждого ТекСтрокаПриходСредств Из ПриходСредств Цикл
    Движение = Движения.РасчетыСФирмами.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Фирма = ТекСтрокаПриходСредств.Фирма;
    Движение.Сумма = ТекСтрокаПриходСредств.Сумма;
КонецЦикла;
// регистр РасчетыСФирмами Приход
Движение.РасчетыСФирмами.Записывать = Истина;
Для Каждого ТекСтрокаРасходСредств Из РасходСредств Цикл
    Движение = Движения.РасчетыСФирмами.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    Движение.Период = Дата;
    Движение.Фирма = ТекСтрокаРасходСредств.Фирма;
    Движение.Сумма = ТекСтрокаРасходСредств.Сумма;
КонецЦикла;
КонецПроцедуры

```

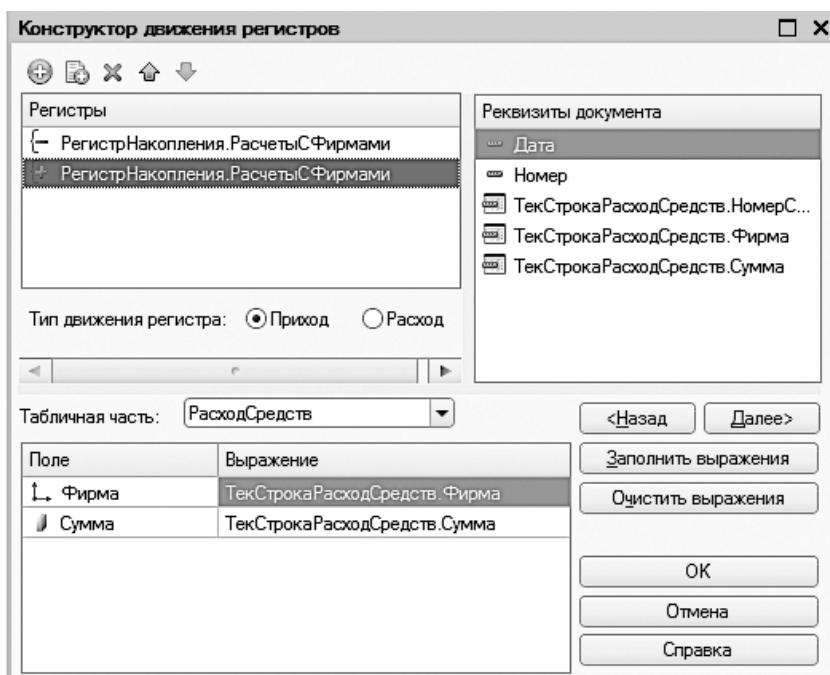


Рис. 4.34. Формирование полей для таблицы РасходСредств

Таким образом, все необходимые технические действия по созданию нового документа выполнены, и можно перейти в режим 1С:Предприятие для проверки проведения документа ВыпискаБанка. Как видно из рис. 4.35, в этом случае в регистре накопления формируются необходимые записи.

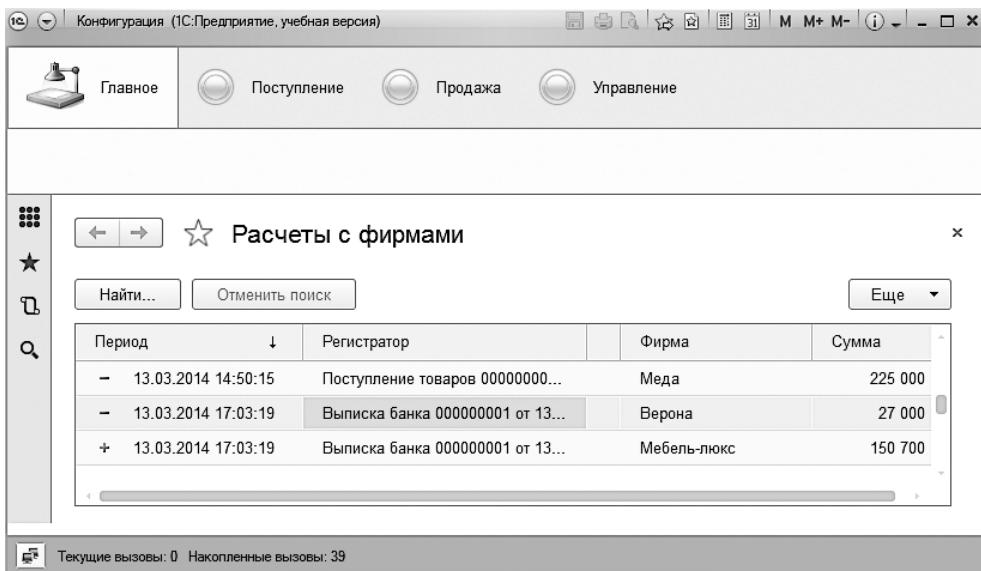


Рис. 4.35. Форма списка регистра Расчеты с фирмами

Автоматизация расчета суммы по табличной части документа

В данном разделе мы поговорим о событиях, которые можно программно обрабатывать для получения необходимого эффекта в определенном объекте системы 1С:Предприятие. Рассматриваемые события касаются работы с документами, с чем мы уже встречались в первой главе, где происходило заполнение строк табличной части документов. Здесь попробуем рассмотреть данный вопрос более подробно, а также решим отложенную задачу на тему вычисления общей суммы по документу (о чём говорилось ранее в этой главе).

Вспомним, что в начале этой главы мы ввели в документы по продажам и поставкам товаров новый реквизит **СуммаПодокументу**, в котором по замыслу должна формироваться сумма по полю табличной части документа (сумма по всем позициям табличной части). Ранее нам приходилось выполнять вычисления по табличной части вручную, а здесь мы совершенствуем данную разработку и обеспечим в реквизите **СуммаПодокументу** перед записью документа автоматическое вычисление суммы по позициям товаров из табличной части.

Начнем с того, что откроем окно редактирования документа ПродажаТоваров и обратимся к вкладке Прочее (рис. 4.36). Здесь щелчком на кнопке Модуль объекта откроем окно с программным кодом данного модуля (рис. 4.37). Здесь присутству-

ют две процедуры обработки событий (какие это события, понятно из названий процедур).

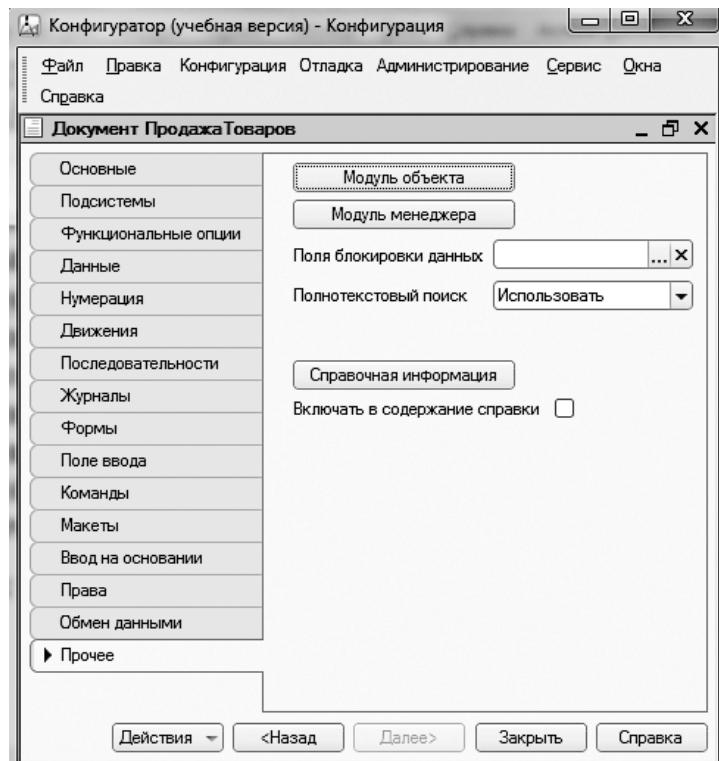


Рис. 4.36. Вкладка Прочее окна редактирования документа ПродажаТоваров

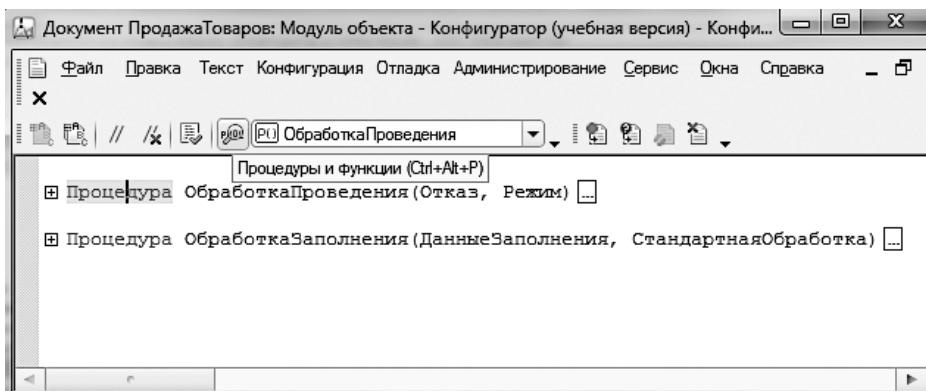


Рис. 4.37. Окно Модуль объекта документа ПродажаТоваров

Для того чтобы узнать, какие еще события могут обрабатываться в данном модуле, следует щелкнуть на значке Процедуры и функции (см. рис. 4.37). В результате на экране мы увидим окно со списком процедур и функций, предназначенных для обработки событий документа (рис. 4.38).

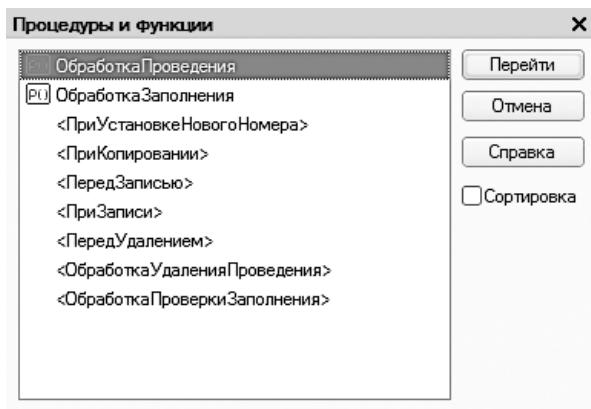


Рис. 4.38. Окно со списком процедур и функций обработки событий

Наиболее часто используется связанное с документом событие ОбработкаПроверкиЗаполнения. В этом случае мы можем проверить корректность заполнения реквизитов документа и, если они заполнены неправильно, выдать об этом сообщение пользователю. Кроме того, в этом случае значению переменной Отказ в процедуре следует присвоить значение Истина.

Сейчас нас интересует событие ПередЗаписью, которое позволяет произвести вычисления и дополнительно заполнить какие-либо реквизиты документа непосредственно перед записью его в информационную базу.

Обратимся к ранее созданному документу ПродажаТоваров и попробуем устраниТЬ один его недостаток, о котором мы упоминали (рис. 4.39). А именно — обеспечим автоматизацию подсчета суммы по табличной части документа.

При этом мы поработаем с коллекцией объектов. *Универсальная коллекция* — это упорядоченная последовательность элементов одинаковой структуры. Количество элементов в универсальной коллекции произвольно. Наиболее распространенный пример универсальной коллекции — это табличная часть, строки которой и являются элементами универсальной коллекции.

В данном случае наша задача сводится к тому, чтобы перед записью документа перебрать строки документа, просуммировать в них значения поля Сумма, а полученный результат внести в поле Сумма по документу. В результате в записанном документе будет заполнен реквизит СуммаПоДокументу.

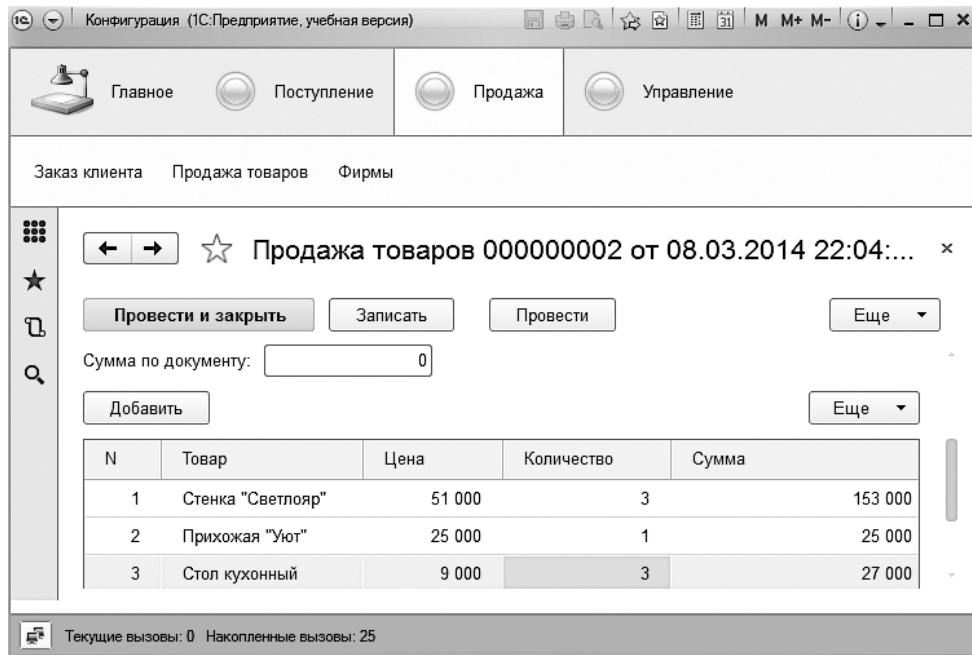


Рис. 4.39. Документ ПродажаТоваров

Для решения поставленной задачи обратимся к модулю документа ПродажаТоваров (см. рис. 4.37). Определимся с событием, обработка которого позволит нам внести в информационную базу документ с заполненным полем Сумма по документу. В качестве такового подходит событие ПередЗаписью (рис. 4.40). Необходимо оформить процедуру обработки этого события в соответствии с листингом 4.5.

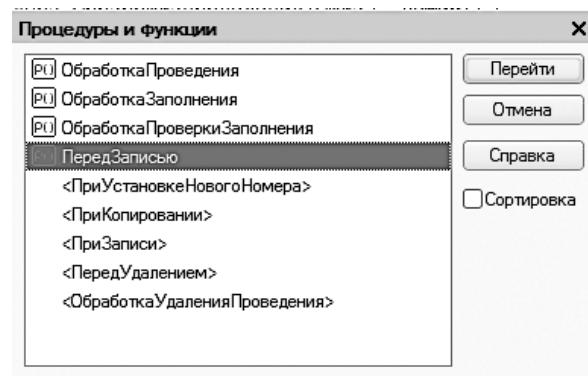


Рис. 4.40. Выбор процедуры обработки события ПередЗаписью

Листинг 4.5. Процедура обработки события ПередЗаписью

```
Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)
    СуммаТЧ=0;
    Для Каждого СтрТабЧасти Из ПереченьТоваров Цикл
        СуммаТЧ+= СуммаТЧ+ СтрТабЧасти.Сумма;
    КонецЦикла;
    СуммаПодДокументу= СуммаТЧ;
КонецПроцедуры
```

В листинге 4.5 мы использовали цикл:

Для Каждого СтрТабЧасти Из ПереченьТоваров Цикл

В этом случае ключевое слово **Каждого** говорит о том, что цикл будет перебирать каждый элемент из коллекции **ПереченьТоваров**. В данном цикле выполняется суммирование в переменной **СуммаТЧ** всех полей **Сумма** из табличной части документа. В заключительной части процедуры производится присваивание реквизиту **СуммаПодДокументу** значения **СуммаТЧ**.

Теперь можно перейти в режим 1С:Предприятие и убедиться, что данная доработка автоматически фиксирует общую сумму по всему документу (рис. 4.41).

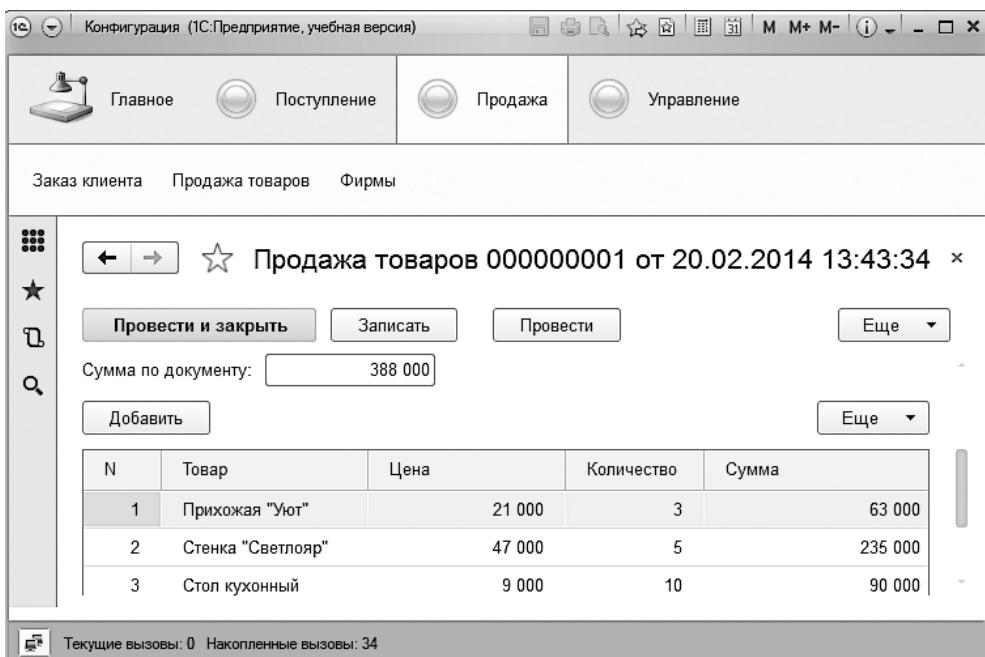


Рис. 4.41. Автоматизация заполнения поля Сумма по документу

Отчет по взаиморасчетам с фирмами

К настоящему времени мы располагаем регистром **РасчетыСФирмами**, где автоматически фиксируются (в результате проведения документов) наши долги перед фирмами и задолженность организаций перед нами. Теперь мы попробуем оформить эту информацию в виде отчета.

Итак, в окне Конфигурация создадим новый отчет **РасчетыСФирмами**, который включим в подсистему Управление. Создадим схему компоновки данных и воспользуемся конструктором запросов (рис. 4.42). Как видно, мы использовали одну виртуальную таблицу **РасчетыСФирмамиОстатки** и в этой таблице выбрали два поля.

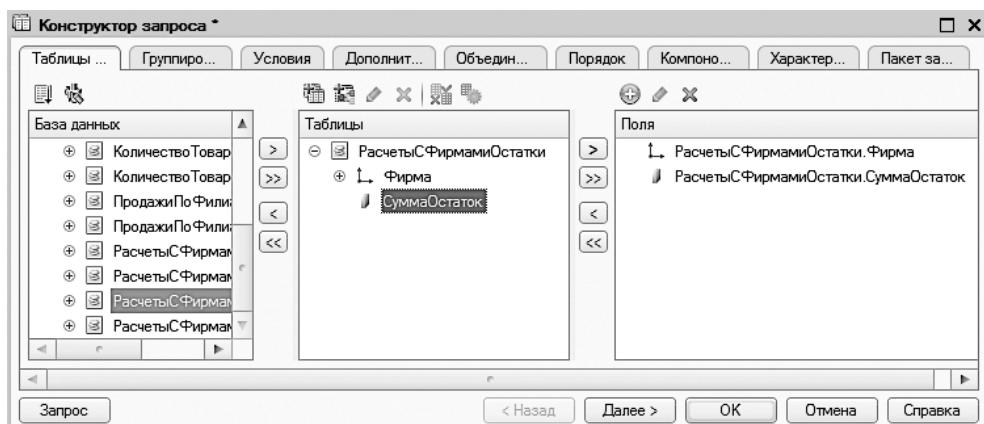


Рис. 4.42. Окно конструктора запросов

На вкладке Объединения/Псевдонимы зададим более понятные псевдонимы полей (рис. 4.43), после чего сформируем текст запроса (листинг 4.6).

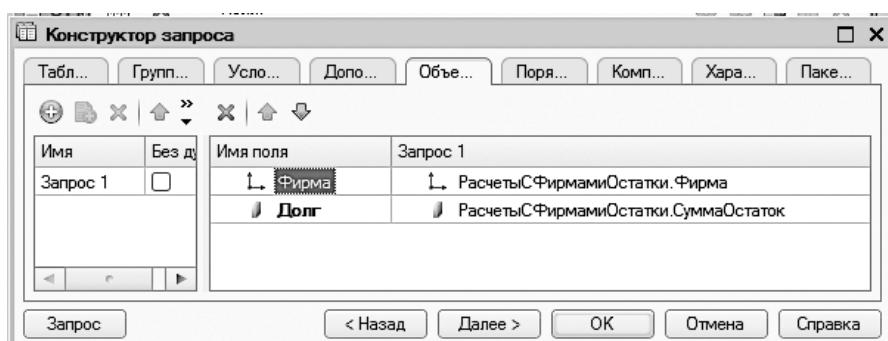


Рис. 4.43. Вкладка Объединения/Псевдонимы

Листинг 4.6. Запрос для извлечения информации из регистра РасчетыСФирмами

ВЫБРАТЬ

РасчетыСФирмамиОстатки.Фирма,
РасчетыСФирмамиОстатки.СуммаОстаток КАК Долг

ИЗ

РегистрНакопления.РасчетыСФирмами.Остатки
КАК РасчетыСФирмамиОстатки

Продолжим формирование отчета и воспользуемся конструктором настроек, где в стартовом окне выберем вариант Список (рис. 4.44).

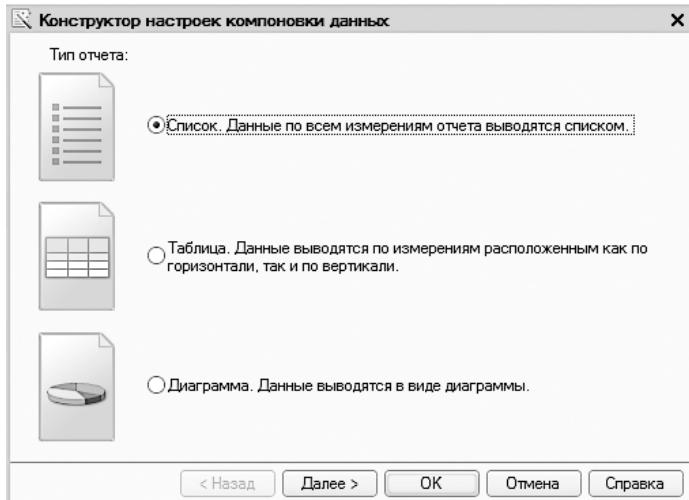


Рис. 4.44. Стартовое окно конструктора настроек отчета

В следующем окне необходимо указать поля для отчета; здесь мы включим в отчет оба имеющихся поля (рис. 4.45).

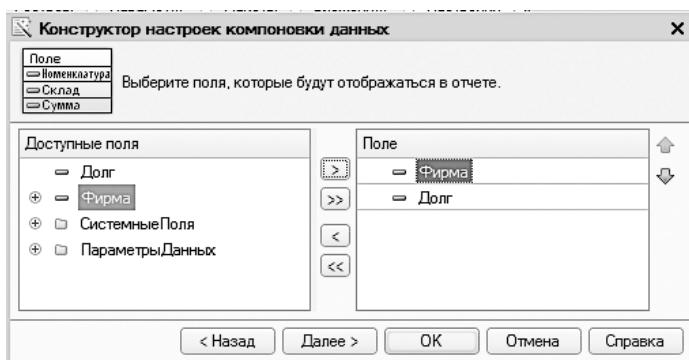


Рис. 4.45. Окно выбора полей для отчета

На этом разработка отчета завершается. Теперь можно перейти в режим 1С:Предприятие для просмотра данного отчета по уже имеющимся в системе данным (рис. 4.46).

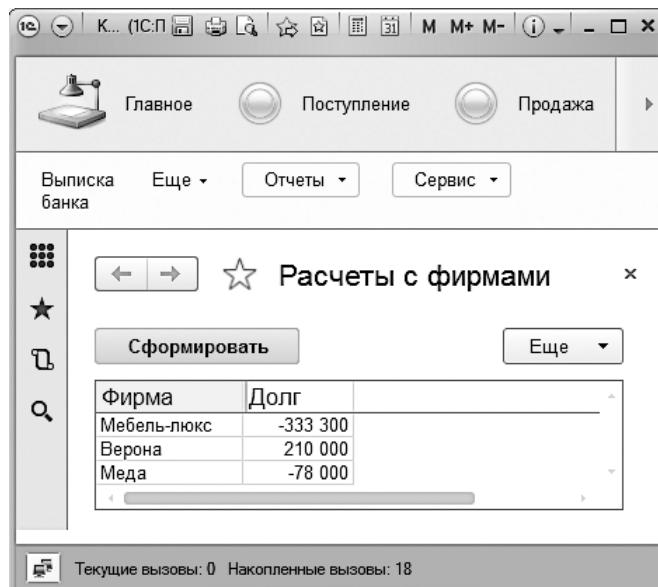


Рис. 4.46. Отчет по взаиморасчетам с контрагентами

Положительные числа в отчете интерпретируются как дебиторская задолженность (долг клиентов перед нашей организацией). Отрицательные числа в отчете говорят о том, что это наш долг перед поставщиком (кредиторская задолженность).

Данный отчет имеет ряд недостатков, которые мы попробуем устраниить. Так, заметным недостатком отчета является отсутствие итоговой суммы. Для того, чтобы ее сформировать, вернемся к конструктору схемы компоновки данных и укажем ресурс Долг (рис. 4.47). После этого обновим конфигурацию и перейдем в режим 1С:Предприятие. В результате в модифицированном отчете мы увидим необходимый итог по имеющимся позициям (рис. 4.48).

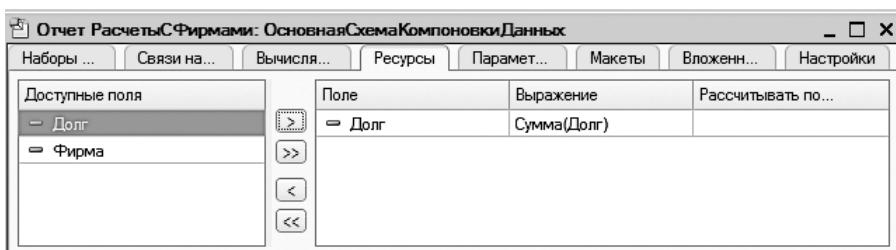


Рис. 4.47. Вкладка Ресурсы конструктора схемы компоновки данных

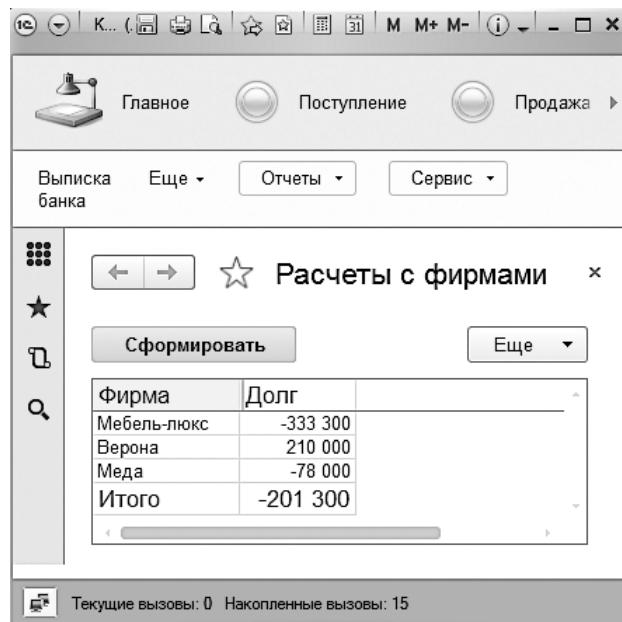


Рис. 4.48. Отчет по взаиморасчетам с контрагентами с итоговой суммой

Еще немного изменим отчет. Допустим, нам нужно иметь возможность строить отчет как по всем контрагентам, так и по выбранным, причем выбор производить непосредственно в форме отчета. Это даст пользователю возможность выбора фирмы, для которой он собирается посмотреть взаиморасчеты.

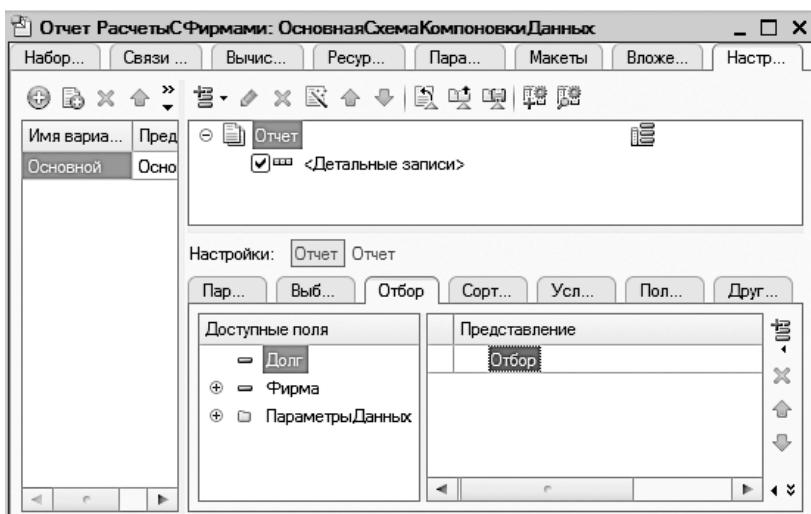


Рис. 4.49. Вкладка Отбор конструктора схемы компоновки данных

Изменим схемы компоновки данных нашего отчета с целью формирования вариантов отчета. Перейдем на вкладку Настройки, а затем — на дополнительную вкладку Отбор (рис. 4.49).

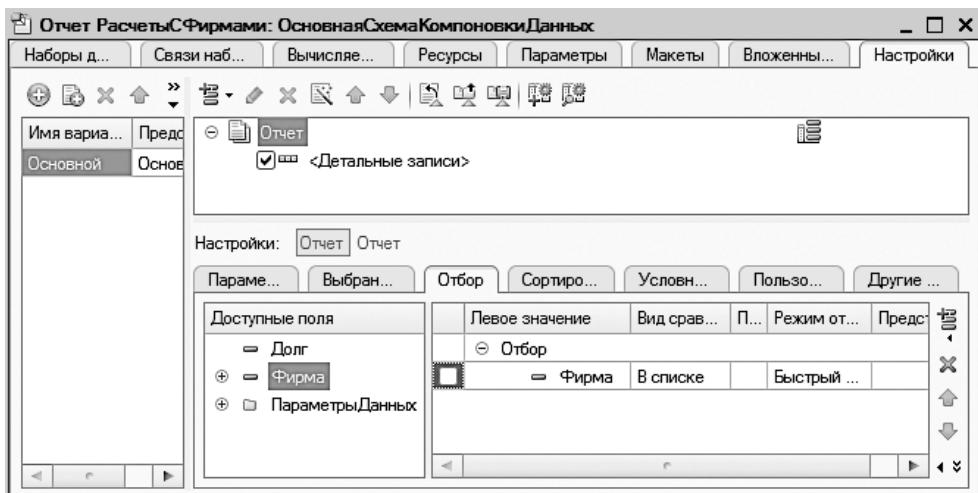


Рис. 4.50. Включение в отбор поля Фирма

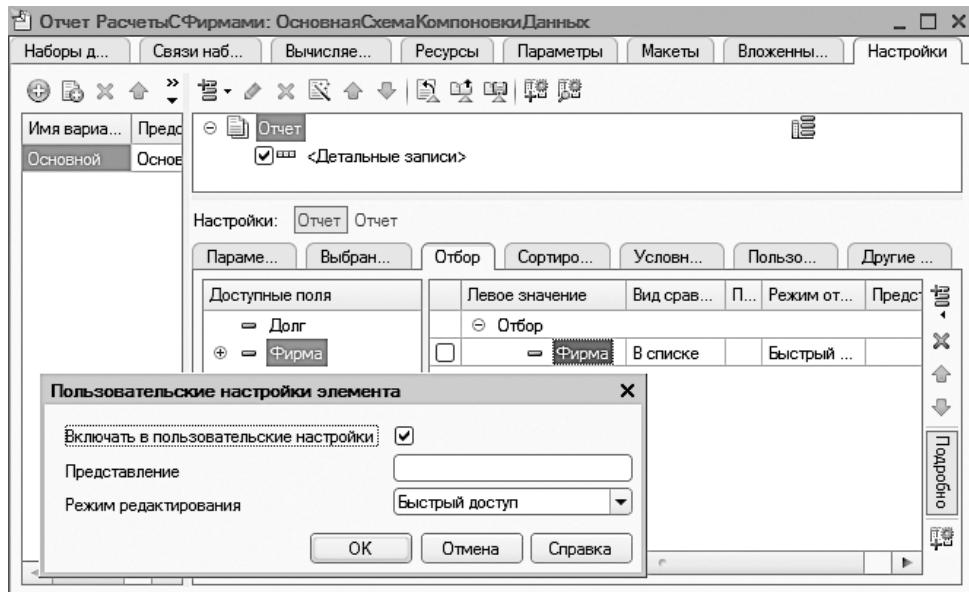


Рис. 4.51. Формирование пользовательских настроек

Перенесем в раздел Отбор поле Фирма и в качестве вида сравнения укажем вариант В списке (рис. 4.50). При этом данный вариант отбора мы отключим, то есть по умолчанию будет выводиться отчет с перечнем взаиморасчетов по всем фирмам. Для этого следует сбросить флажок в строке отбора. Кроме того, необходимо включить поле Фирма в пользовательские настройки отчета (рис. 4.51).

Теперь необходимо обновить конфигурацию, перейти в режим 1С:Предприятие и открыть форму отчета. После модернизации в форме отчета появился список для выбора фирмы, причем если мы такой выбор не делаем, то отчет будет сформирован в прежнем варианте с перечнем всех взаиморасчетов. Однако теперь мы можем сформировать список полей для отбора. Например, на рис. 4.52 показан отчет с отбором по двум определенным фирмам.

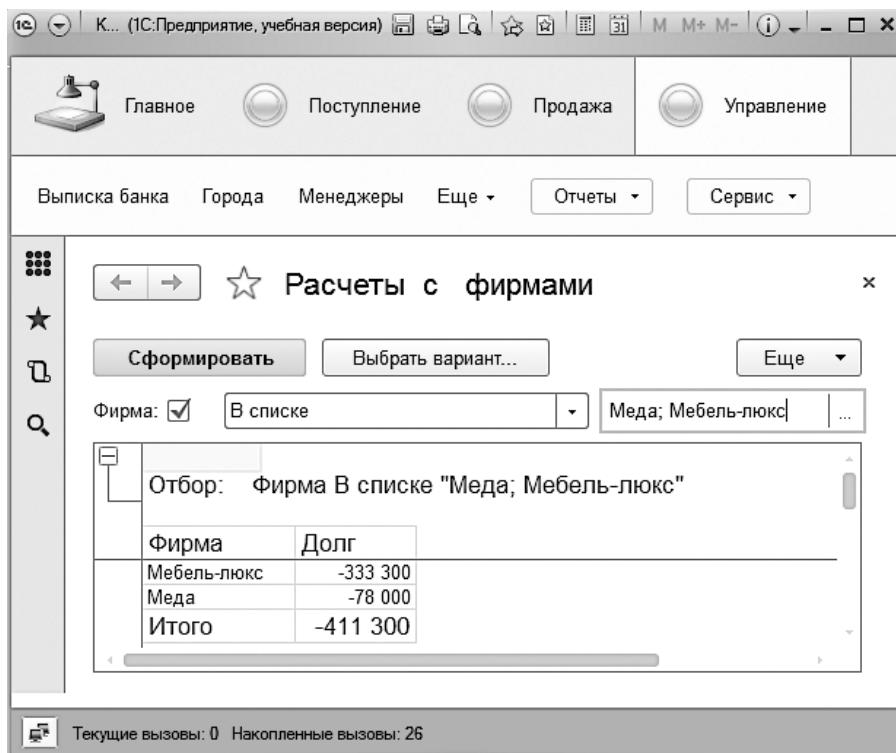


Рис. 4.52. Отчет по выбранным организациям

Хотя поставленная задача в принципе решена, мы выполним еще небольшую модификацию отчета, а именно — предоставим пользователю в форме отчета возможность указывать, на какую дату он собирается посмотреть взаиморасчеты.

В данный момент дата не задается, что означает формирование остатков на текущий момент. Для определения даты в отчете следует воспользоваться параметром Период (рис. 4.53). Этот параметр автоматически добавляется в схему компоновки данных в связи с тем, что при построении запроса производится обращение к виртуальной таблице. Осталось включить параметр Период в пользовательские настройки (рис. 4.54) и получить планируемый отчет (рис. 4.55).

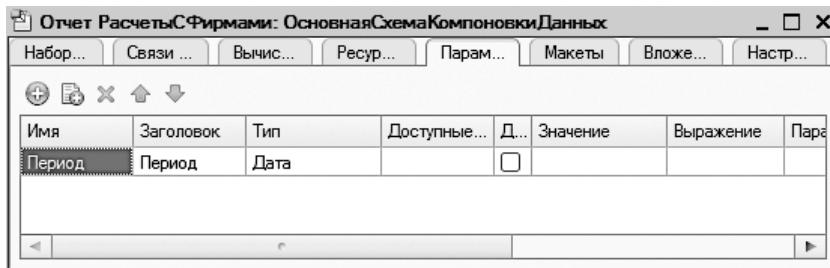


Рис. 4.53. Вкладка Параметры окна конструктора схемы компоновки данных

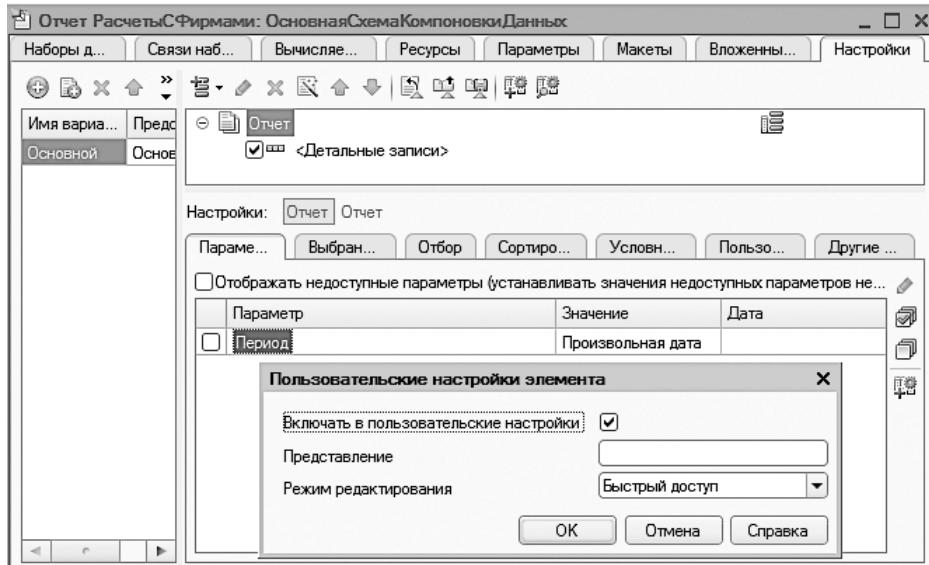


Рис. 4.54. Включение параметра Период в пользовательские настройки

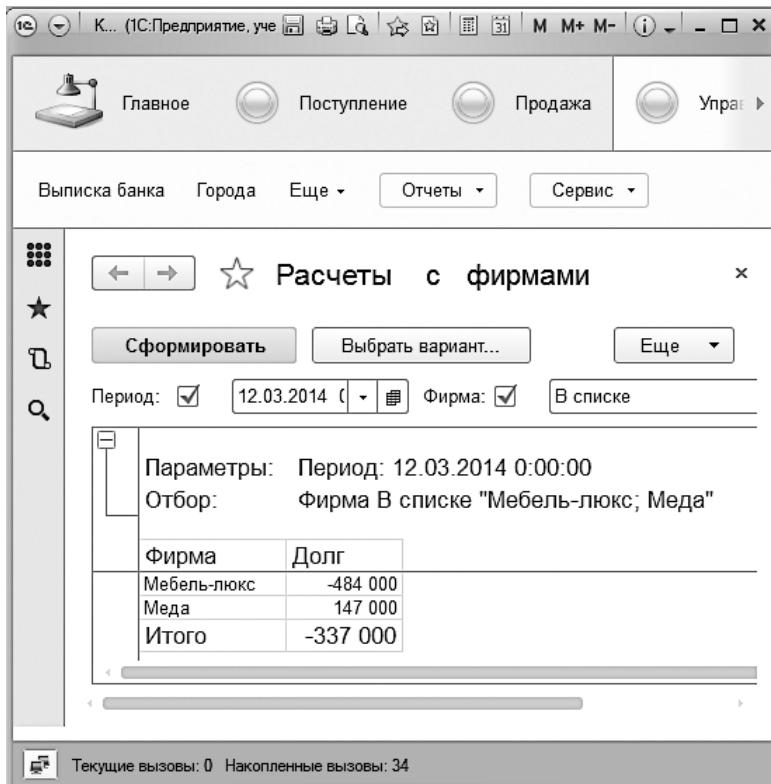


Рис. 4.55. Отчет с указанием даты формирования остатков

Итоги

В данной главе мы рассмотрели несколько приемов организации учета данных. При этом исходным материалом явилась конфигурация, разработанная нами в первой главе. Здесь мы добавили в нее несколько усовершенствований, чтобы максимально приблизить информационную модель учета к реальной ситуации.

Глава 5

Примеры прикладных решений

В данной главе мы рассмотрим несколько независимых примеров (в каждом примере будем создавать отдельную конфигурацию), используя как уже знакомые нам по предыдущим главам приемы, так и новые ресурсы разработки.

Учет оплаты учебных курсов

В этом разделе мы создадим новую конфигурацию, на примере которой продемонстрируем возможности системы компоновки данных. Тема прикладной задачи будет связана с организацией учета оплаты за учебные курсы. В данной ситуации условно будем считать, что наша организация оказывает платные образовательные услуги.

Создание справочников

Учитывая уже накопленный опыт в плане разработки конфигураций, комментарий к техническим действиям при создании объектов конфигурации для данной задачи будет менее подробный, чем в предыдущих главах.

Начнем разработку с создания справочника **Клиенты**, в котором дополнительных реквизитов добавлять не потребуется. После формирования справочника в режиме конфигуратора можно сразу перейти в режим 1С:Предприятие и внести в информационную структуру, созданную на основании разработанного объекта конфигурации, несколько фамилий учащихся (рис. 5.1).

Аналогичным образом создадим справочник **Преподаватели**. Заполненный справочник в режиме 1С:Предприятие показан на рис. 5.2.

Еще один справочник для нашей конфигурации будет называться **Курсы**. В отличие от двух предыдущих, добавим в него дополнительный реквизит **Стоимость** (тип данных — число). Тем самым мы обеспечим возможность задания в созданном справочнике стоимости конкретного курса. Для полноты картины на рис. 5.3 показан заполненный справочник **Курсы** в режиме 1С:Предприятие.

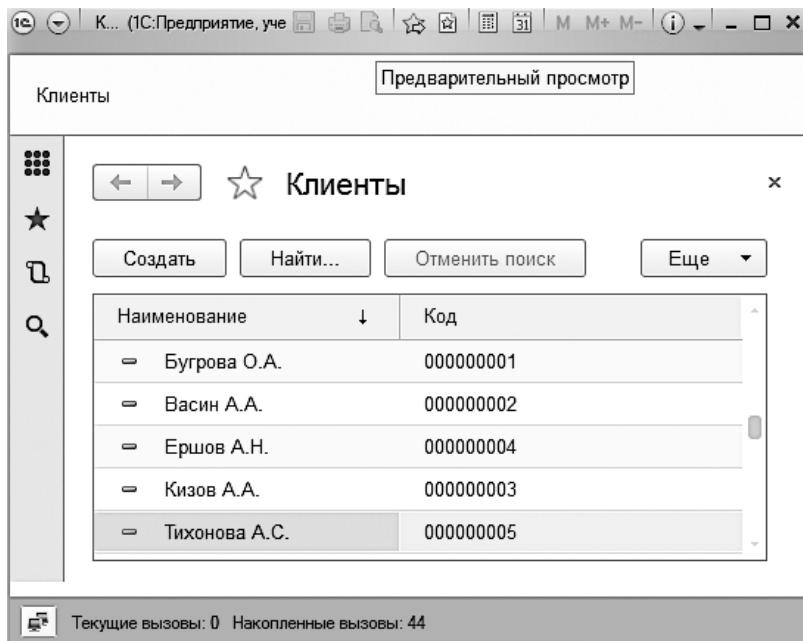


Рис. 5.1. Заполненный справочник Клиенты в режиме 1С:Предприятие

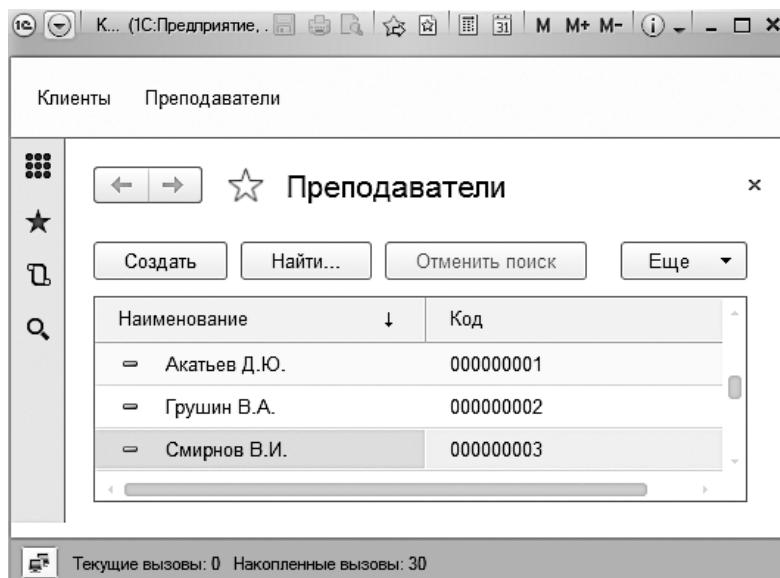


Рис. 5.2. Заполненный справочник Преподаватели в режиме 1С:Предприятие

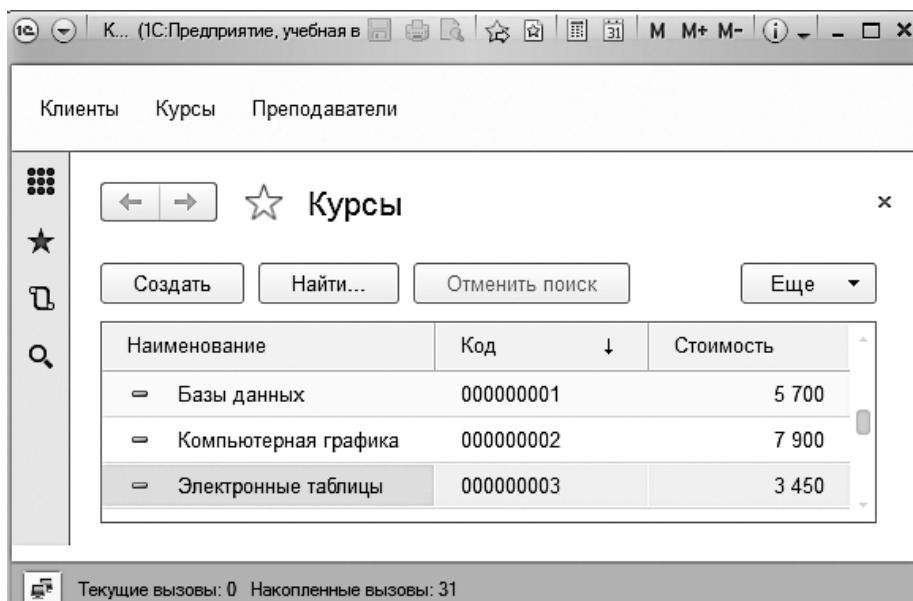


Рис. 5.3. Заполненный справочник Курсы в режиме 1С:Предприятие

Регистрация внесения денежных средств

Для организации работы сотрудников учебного отдела необходим документ, в котором должна фиксироваться оплата, внесенная за учебные курсы. При этом внесение оплаты может быть поэтапным (не сразу за весь курс), поэтому в режиме конфигуратора создадим новый объект конфигурации — документ ВнесениеОплаты. В нем кроме традиционных реквизитов (номера документа и его даты) сформируем табличную часть с названием ПереченьОплат, в которой установим следующие реквизиты (рис. 5.4):

- Учащийся (тип данных — СправочникСсылка.Клиенты);
- Курс (тип данных — СправочникСсылка.Курсы);
- Преподаватель (тип данных — СправочникСсылка.Преподаватели);
- Сумма (тип данных — число).

В табличной части документа ВнесениеОплаты напротив каждого курса указывается фамилия преподавателя, который его будет вести.

Кроме документа ВнесениеОплаты нам необходимо создать регистр накопления Оплаты. В окне редактирования этого объекта конфигурации на вкладке Основные следует указать, что создаваемый регистр будет регистром *оборотов*. После этого на вкладке Данные укажем два измерения регистра:

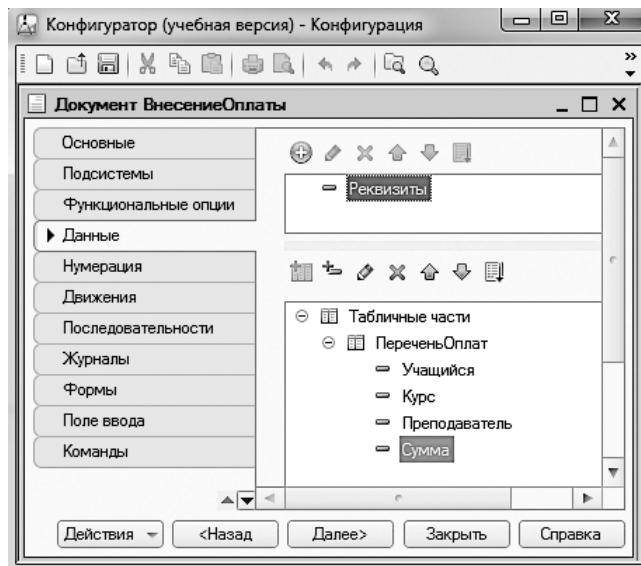


Рис. 5.4. Структура документа ВнесениеОплаты

- Клиент (тип данных — СправочникСсылка.Клиенты);
- Курс (тип данных — СправочникСсылка.Курсы).

В качестве единственного ресурса укажем ресурс ВнесеннаяСумма.

Теперь перед переходом к работе в режиме 1С:Предприятие необходимо обеспечить формирование движений по регистру (иначе система сообщит нам о незавершенности разработки). Для этого опять вернемся к окну редактирования документа ВнесениеОплаты, где на вкладке Движения с помощью конструктора обеспечим соответствие между полями регистра и реквизитами данного документа (рис. 5.5). В листинге 5.1 приведен текст процедуры, формируемой конструктором движений.

Листинг 5.1. Процедура обработки проведения документа Оплата

```
Процедура ОбработкаПроведения(Отказ, Режим)
//{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
Движения.Оплаты.Записывать = Истина;
Для Каждого ТекСтрокаПереченьОплат Из ПереченьОплат Цикл
Движение = Движения.Оплаты.Добавить();
Движение.Период = Дата;
Движение.Клиент = ТекСтрокаПереченьОплат.Учащийся;
Движение.Курс = ТекСтрокаПереченьОплат.Курс;
Движение.ВнесеннаяСумма = ТекСтрокаПереченьОплат.Сумма;
КонецЦикла;
//}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

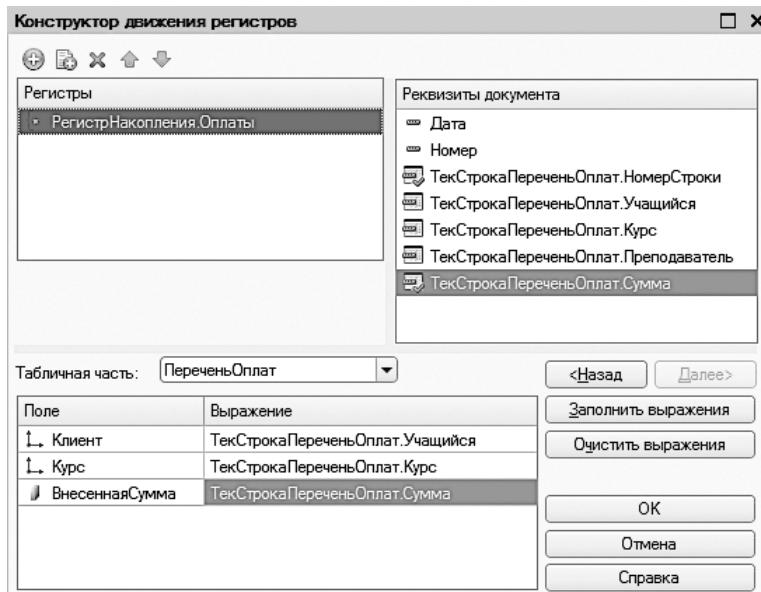


Рис. 5.5. Окно конструктора движений

Теперь все для работы пользователя в режиме 1С:Предприятие подготовлено: имеется регистр накопления и документ, выполняющий движения по данному регистру. Вводимые документы ВнесениеОплаты после их проведения будут вносить изменения в состояние учета. Поэтому перейдем в режим 1С:Предприятие и введем несколько документов по внесению оплаты за курсы (один из вариантов показан на рис. 5.6).

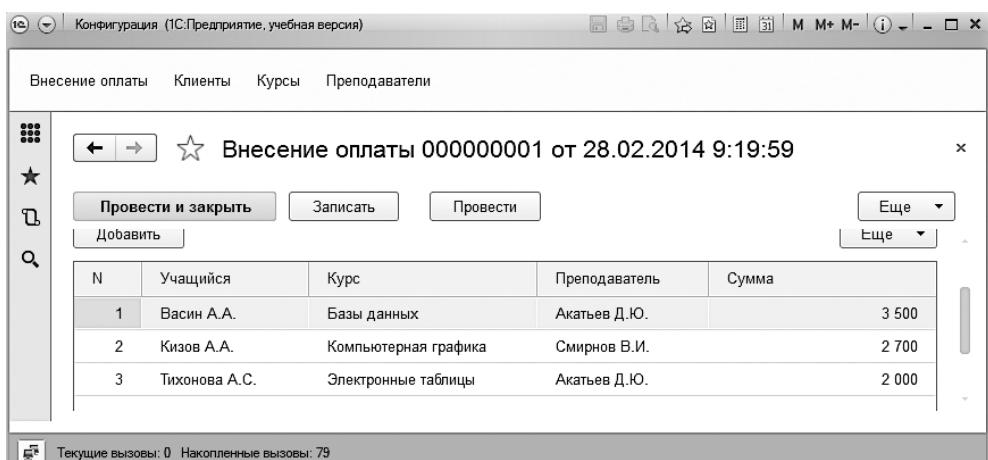


Рис. 5.6. Документ учета внесения денежных средств

Отчет по курсам

Таким образом, подготовка информационной базы завершена и можно перейти к разработке отчетов с использованием системы компоновки данных. Первый отчет будет формировать сведения о том, какой доход получен по каждому учебному курсу с детализацией по месяцам и разворотом по клиентам.

В режиме конфигуратора создадим новый отчет (**Отчет5_1**), после чего в окне конструктора запросов выберем виртуальную таблицу регистра накопления **Оплаты.Обороты** (рис. 5.7).

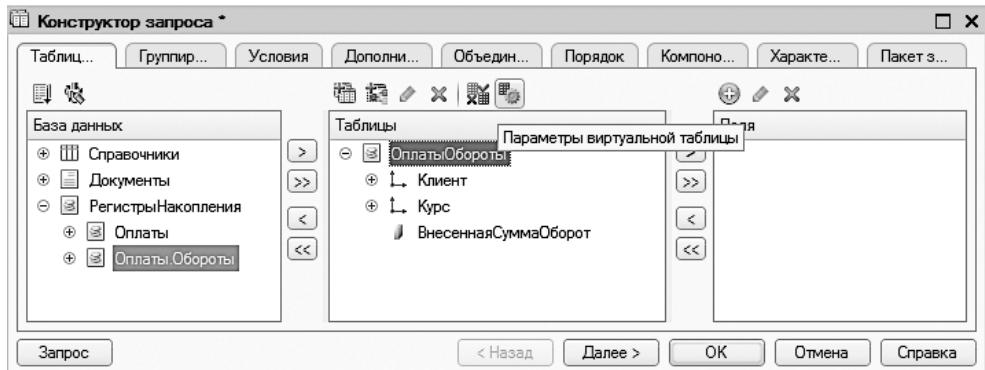


Рис. 5.7. Окно конструктора запроса

Для виртуальной таблицы следует задать параметр периодичности, поэтому воспользуемся уже знакомой кнопкой **Параметры виртуальной таблицы** (самая правая кнопка в пиктографическом меню). В результате перед нами откроется окно, показанное на рис. 5.8, где необходимо в списке **Периодичность** выбрать вариант **Месяц**.

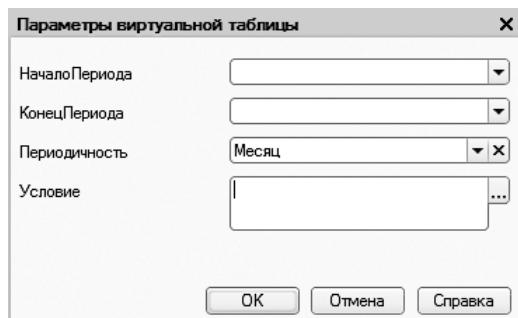


Рис. 5.8. Окно для задания параметров виртуальной таблицы

После щелчка на кнопке **OK** в число полей таблицы будет добавлено поле **Период**. Необходимые для формируемого запроса поля указаны на рис. 5.9.

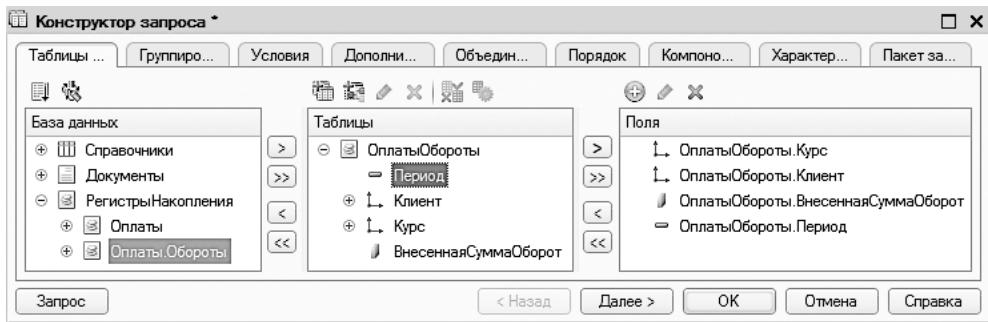


Рис. 5.9. Используемые поля в окне конструктора запроса

На рис. 5.10 показана вкладка Объединения/Псевдонимы, где следует задать для поля ВнесеннаяСуммаОборот псевдоним Сумма.

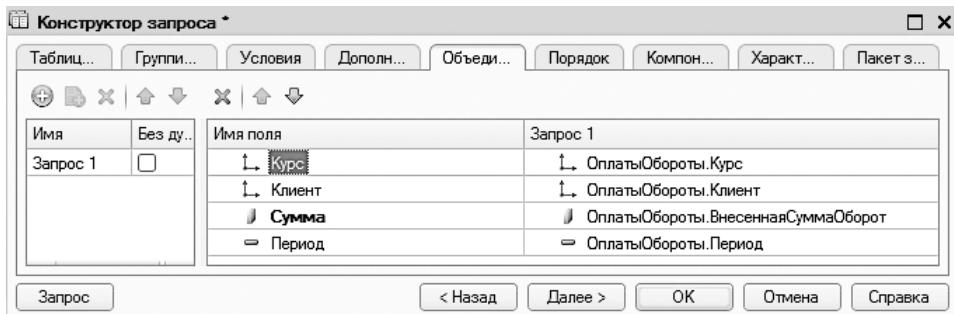


Рис. 5.10. Установка псевдонима поля

Далее на вкладке Группировка укажем три групповых поля и одно суммируемое поле (рис. 5.11).

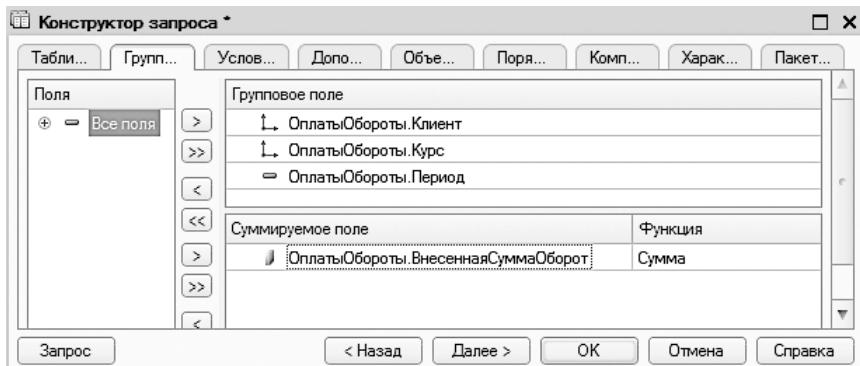


Рис. 5.11. Вкладка схемы компоновки данных для группировки полей

После выполненных действий зададим упорядочивание в запросе по полю Период в порядке убывания (рис. 5.12).

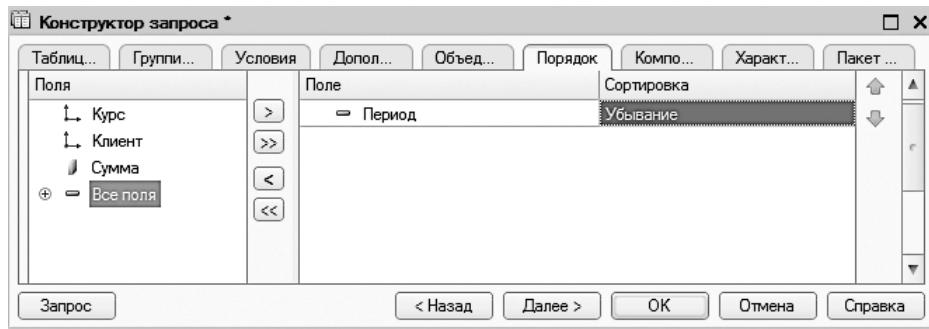


Рис. 5.12. Вкладка Порядок окна конструктора запроса

На этом работа с конструктором запроса завершается. Теперь следует щелкнуть на кнопке OK, чтобы получить текст запроса (листинг 5.2).

Листинг 5.2. Текст запроса для извлечения информации об оплате за курсы

```

ВЫБРАТЬ
    ОплатыОбороты.Курс,
    ОплатыОбороты.Клиент,
    СУММА(ОплатыОбороты.ВнесеннаяСуммаОборот) КАК Сумма,
    ОплатыОбороты.Период КАК Период
ИЗ
    РегистрНакопления.Оплаты.Обороты(, , Месяц, ) КАК ОплатыОбороты
СГРУППИРОВАТЬ ПО
    ОплатыОбороты.Клиент,
    ОплатыОбороты.Курс,
    ОплатыОбороты.Период
УПОРЯДОЧИТЬ ПО
    Период УБЫВ
  
```

В следующей конструкции запроса задается периодичность выбираемых данных:

ИЗ РегистрНакопления.Оплаты.Обороты(, , Месяц,)

Как видно, в качестве периодичности мы указали параметр Месяц.

Продолжим формирование нашего отчета с помощью конструктора схемы компоновки данных и на вкладке Ресурсы укажем необходимый ресурс (рис. 5.13).

Перейдем теперь на вкладку Настройки, где установим группировки в соответствии с рис. 5.14. Здесь кроме группировки верхнего уровня по полю Курс добавлена еще одна — по полю Период. В дополнение к ним включим группировку, вложенную в группировку по полю Период — Детальные записи, а в список выбранных полей добавим поля Клиент и Сумма.

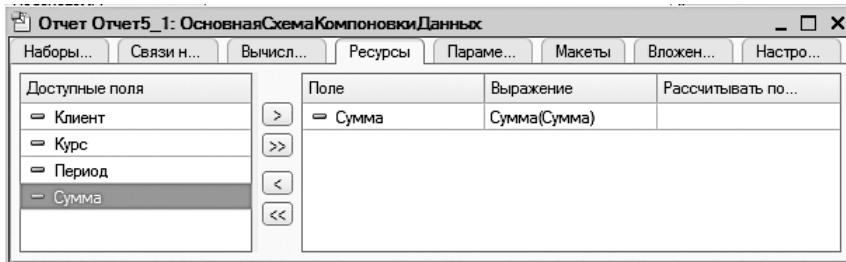


Рис. 5.13. Вкладка Ресурсы окна конструктора схемы компоновки данных

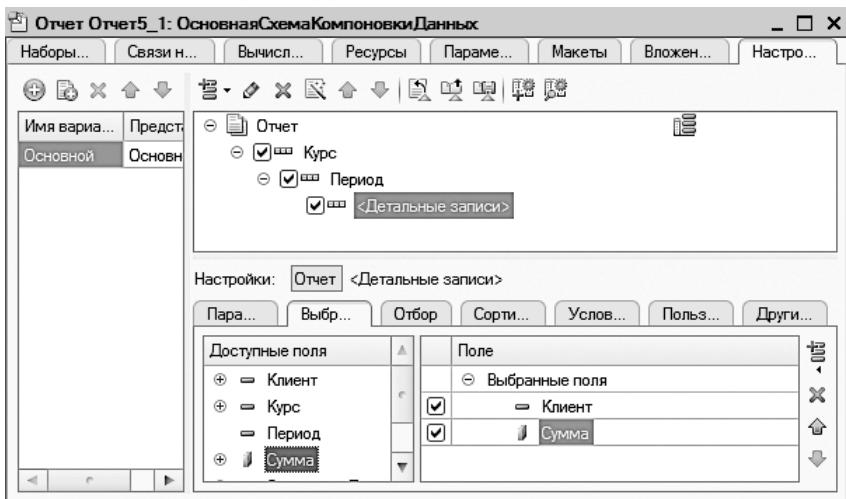


Рис. 5.14. Вкладка Настройки окна конструктора схемы компоновки данных

На вкладке Другие настройки установим значения двух параметров (рис. 5.15). Так, для параметра Расположение полей группировок установим значение Отдельно и только в итогах, а для параметра Расположение общих итогов по вертикали — значение Начала. После этого уже можно перейти в режим 1С:Предприятие и сформировать отчет по имеющимся данным (рис. 5.16).

Полученный отчет достаточно удобен. Так, в нем доступны наряду с итоговыми показателями еще и детальные данные по оплате каждого клиента за каждый месяц.

Отчет об оплате

С точки зрения практической целесообразности весьма полезен отчет, в котором напротив каждого учащегося указывается название учебного курса, уже внесенная им оплата, а также стоимость курса. Создадим такой отчет в нашей конфигурации (Отчет5_2).

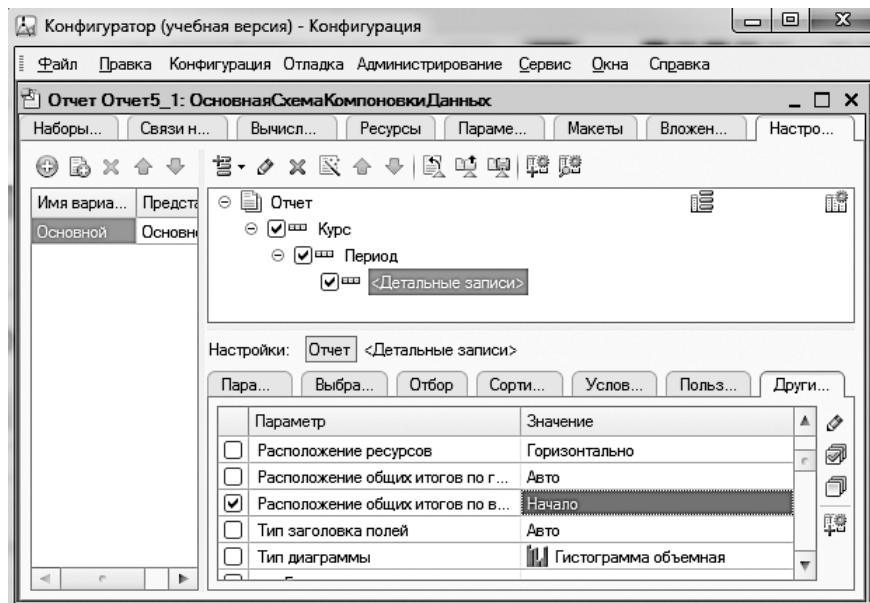


Рис. 5.15. Вкладка ДругиеНастройки окна конструктора схемы компоновки данных

Внесение оплаты Клиенты Курсы Преподаватели Отчеты Сервис

Отчет5_1

Сформировать Выбрать вариант... Еще

Курс	Период	Клиент	Сумма
Итого			22 600
Базы данных			4 200
Компьютерная графика	01.01.2014 0:00:00		8 850
	01.02.2014 0:00:00	Тихонова А.С.	1 200
		Васин А.А.	7 650
		Кизов А.А.	1 500
		Тихонова А.С.	4 400
Электронные таблицы	01.12.2013 0:00:00		1 750
			9 550
		Васин А.А.	1 550
		Кизов А.А.	500
		Тихонова А.С.	750
			300
	01.01.2014 0:00:00		2 900

Текущие вызовы: 0 Накопленные вызовы: 43

Рис. 5.16. Отчет об оплате курсов

Технология его разработки, как и ранее, связана со схемой компоновки данных. После создания нового набора данных (используется знакомая команда Добавить набор данных — запрос) первое техническое действие, которое нуждается в описании, относится к формированию запроса с помощью конструктора запроса (рис. 5.17). В качестве таблиц выберем две:

- таблица справочника Курсы;
- виртуальная таблица Оплаты.Обороты.

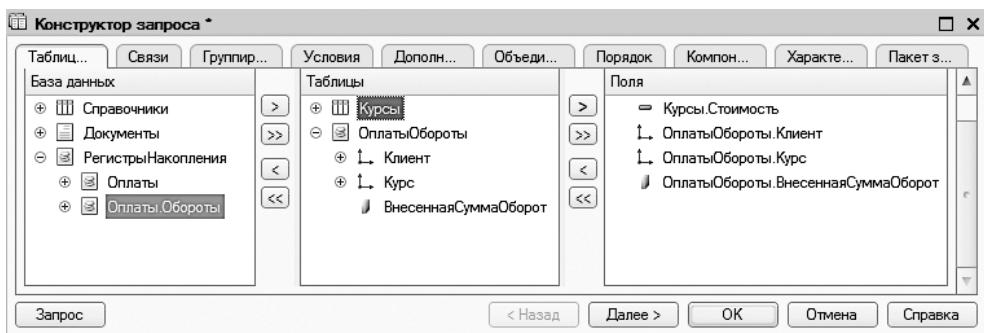


Рис. 5.17. Окно конструктора запроса

Из этих таблиц отберем четыре поля, представленные на рис. 5.17.

На вкладке Связи организуем внутреннее соединение таблиц (рис. 5.18).

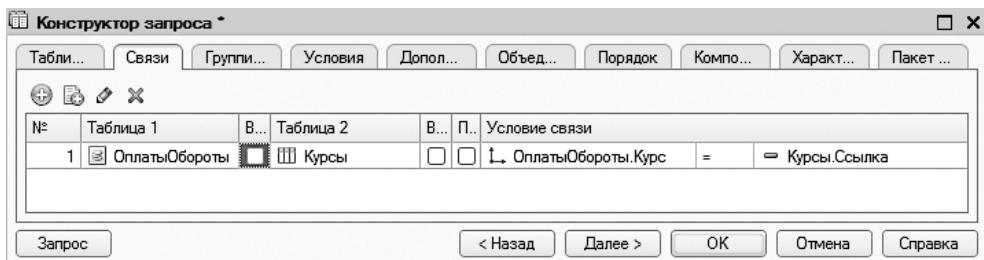


Рис. 5.18. Вкладка Связи окна конструктора запроса

Еще установим псевдоним для поля ВнесеннаяСуммаОборот на вкладке Объединения/Псевдонимы (рис. 5.19).

После этого зададим упорядочивание данных оплаченной суммы в убывающем порядке (рис. 5.20).

На этом действия по разработке запроса завершаются. Остается щелкнуть на кнопке OK для получения текста запроса (листинг 5.3).

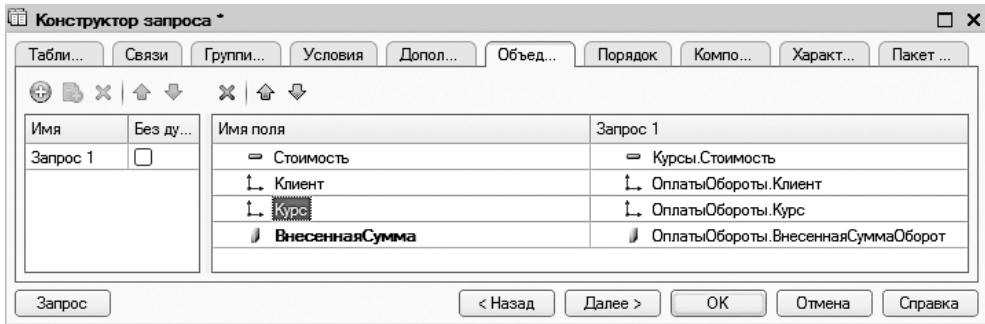


Рис. 5.19. Установка псевдонима поля для отчета

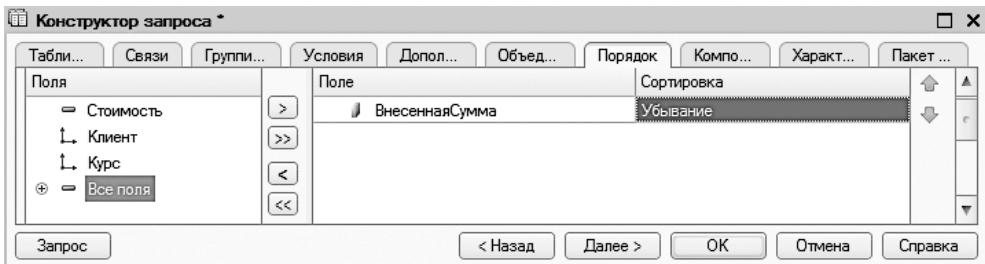


Рис. 5.20. Установка сортировки на вкладке Порядок

Листинг 5.3. Текст запроса для извлечения информации об оплате

```

ВЫБРАТЬ
    Курсы.Стоимость,
    ОплатыОбороты.Клиент,
    ОплатыОбороты.Курс,
    ОплатыОбороты.ВнесеннаяСуммаОборот КАК ВнесеннаяСумма
ИЗ
    РегистрНакопления.Оплаты.Обороты КАК ОплатыОбороты
        ВНУТРЕННЕЕ СОЕДИНЕНИЕ Справочник.Курсы КАК Курсы
            ПО ОплатыОбороты.Курс = Курсы.Ссылка
УПОРЯДОЧИТЬ ПО
    ВнесеннаяСумма УБЫВ

```

На вкладке **Вычисляемые поля** добавим к имеющимся полям еще одно, отражающее недоплату за курс (рис. 5.21).

После этого перейдем на вкладку **Настройки** и выполним типичные установки (рис. 5.22).

Теперь уже можно перейти в режим 1С:Предприятие и получить готовый отчет о реальной оплате в сочетании с необходимыми суммами, которые учащиеся должны оплатить (рис. 5.23). По тому, как составлено выражение (см. рис. 5.21), понятно,

что недоплата отражается в виде отрицательного значения. Если же имеется переплата по курсу, то она отражается в виде положительного значения.

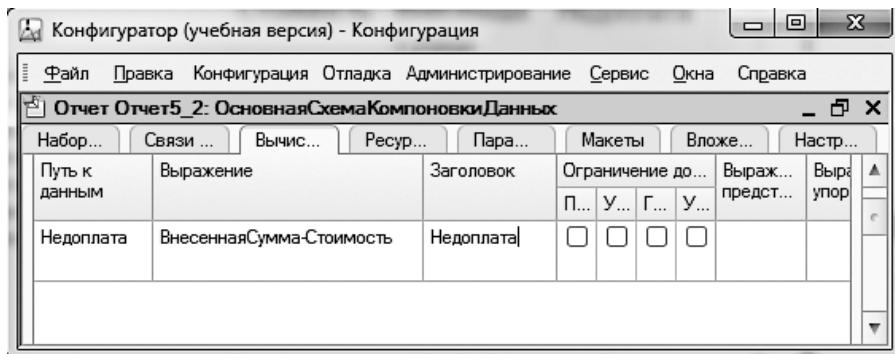


Рис. 5.21. Добавление поля Недоплата на вкладке Вычисляемые поля

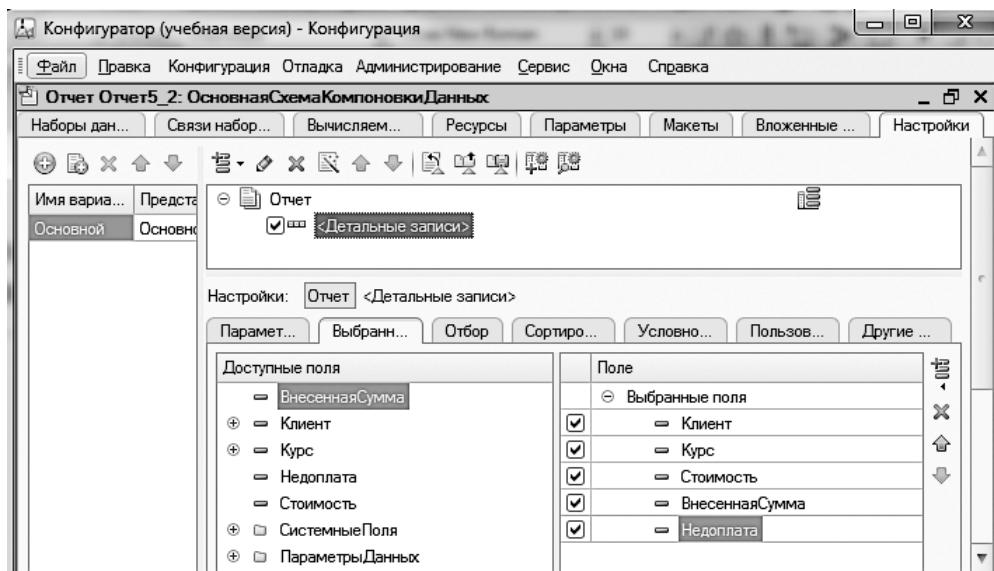


Рис. 5.22. Вкладка Настройки схемы компоновки данных

Отчет с диаграммой

Рассмотрим теперь процедуру создания отчета об оплате с диаграммой. Для иллюстрации сразу приведем внешний вид отчета, который нам предстоит разработать (рис. 5.24). Здесь в виде столбчатой диаграммы отражена оплата учащихся.

Внесение оплаты Клиенты Курсы Преподаватели Отчеты

◀ ▶ ☆ Отчет 5 2

Сформировать Выбрать вариант... Еще

Клиент	Курс	Стоимость	Внесенная сумма	Недоплата
Кизов А.А.	Компьютерная графика	7 900	4 400	-3 500
Тихонова А.С.	Электронные таблицы	3 450	4 400	950
Васин А.А.	Базы данных	5 700	4 200	-1 500
Кизов А.А.	Электронные таблицы	3 450	3 150	-300
Тихонова А.С.	Компьютерная графика	7 900	2 950	-4 950
Васин А.А.	Электронные таблицы	3 450	2 000	-1 450
Васин А.А.	Компьютерная графика	7 900	1 500	-6 400

Текущие вызовы: 0 Накопленные вызовы: 5

Рис. 5.23. Отчет об оплате за курсы

Внесение оплаты Клиенты Курсы Преподаватели Отчеты

◀ ▶ ☆ Отчет 5 3

Сформировать Выбрать вариант... Еще

Преподаватель	Сумма
Васин А.А.	1500
Кизов А.А.	1200
Тихонова А.С.	1800

Рис. 5.24. Отчет по оплатам с использованием диаграммы

Начнем разработку данного отчета с использованием схемы компоновки данных и в окне конструктора запроса отберем необходимые для него поля (рис. 5.25).

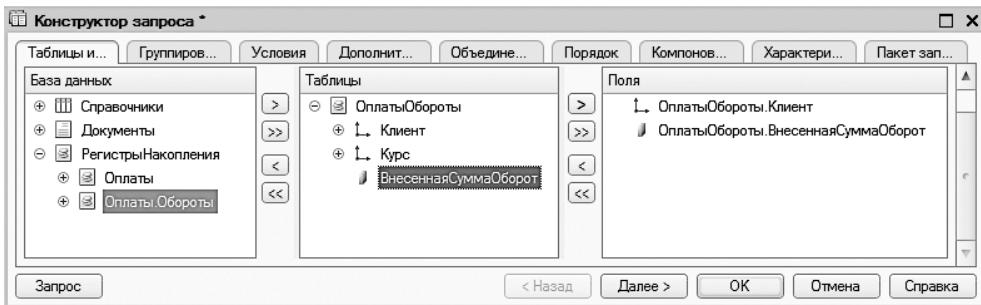


Рис. 5.25. Окно конструктора запроса

Так, для отчета мы использовали одну виртуальную таблицу Оплаты.Обороты. На этом работа с конструктором запроса завершается. Щелчком на кнопке OK для будущего отчета можно получить текст запроса (листинг 5.4).

Листинг 5.4. Текст запроса для отчета об оплате учащихся

```
ВЫБРАТЬ
    ОплатыОбороты.Клиент,
    ОплатыОбороты.ВнесеннаяСуммаОборот
ИЗ
    РегистрНакопления.Оплаты.Обороты КАК ОплатыОбороты
```

Продолжим работу по конструированию отчета с помощью схемы компоновки данных. На вкладке Ресурсы включим ресурс ВнесеннаяСуммаОборот (рис. 5.26), после чего перейдем на вкладку Настройки и зададим группировки для создаваемого отчета (рис. 5.27).

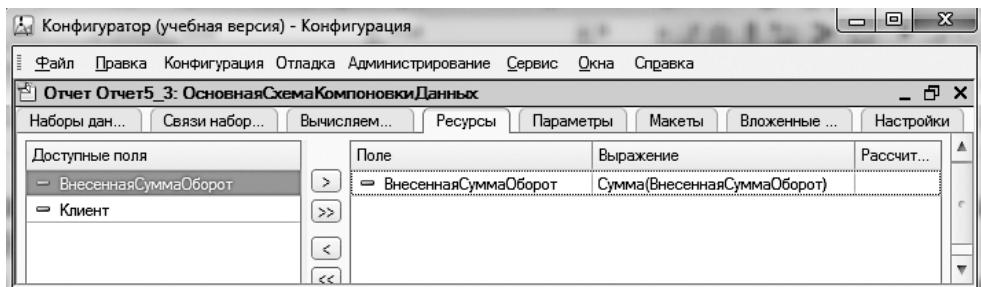


Рис. 5.26. Вкладка Ресурсы окна конструктора схемы компоновки данных

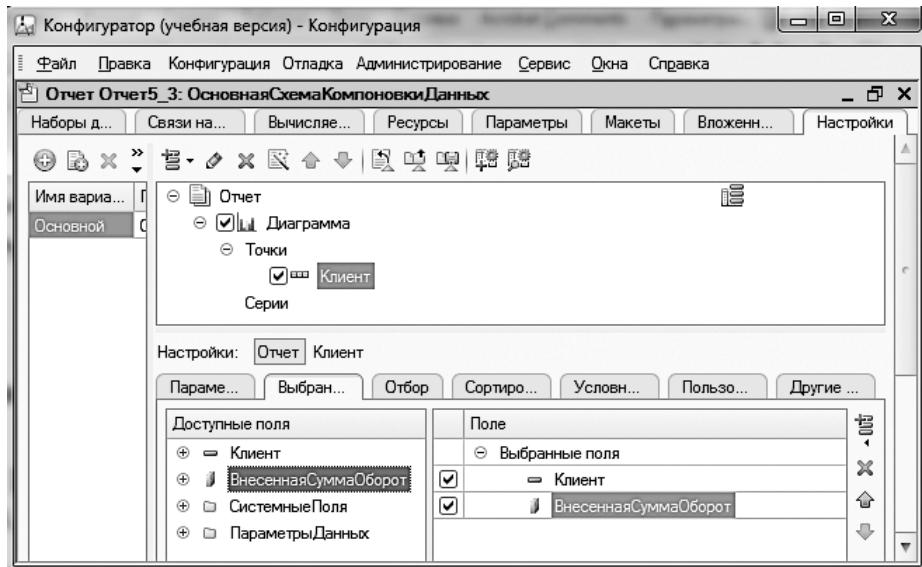


Рис. 5.27. Вкладка Настройки окна конструктора схемы компоновки данных

На этом разработка отчета в том варианте, который мы видели на рис. 5.24, завершается. То есть уже можно перейти в режим 1С:Предприятие для формирования отчета, однако сначала мы доработаем отчет с целью включения в него возможности задания интервала дат. Для этого вернемся в режим конфигуратора и на дополнительной вкладке Параметры вкладки Настройки включим присутствующие там параметры в пользовательские настройки (рис. 5.28).

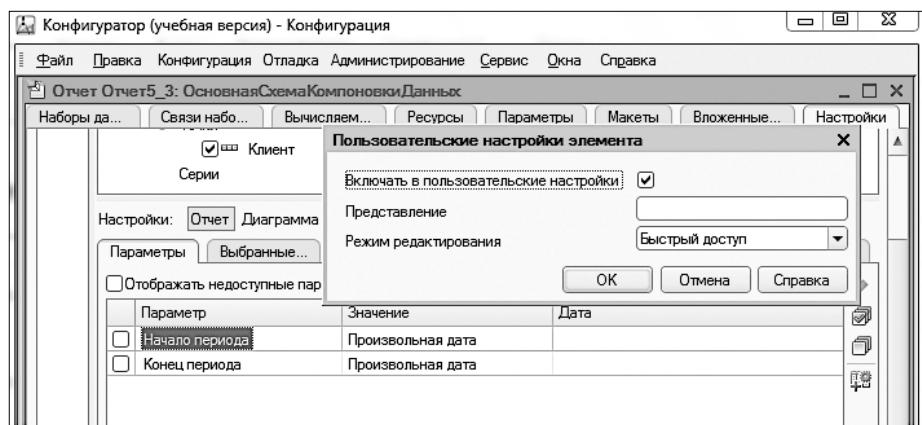


Рис. 5.28. Окно пользовательских настроек элемента

Теперь в режиме 1С:Предприятие мы увидим новый вариант отчета с возможностью выбора интервала дат (рис. 5.29).

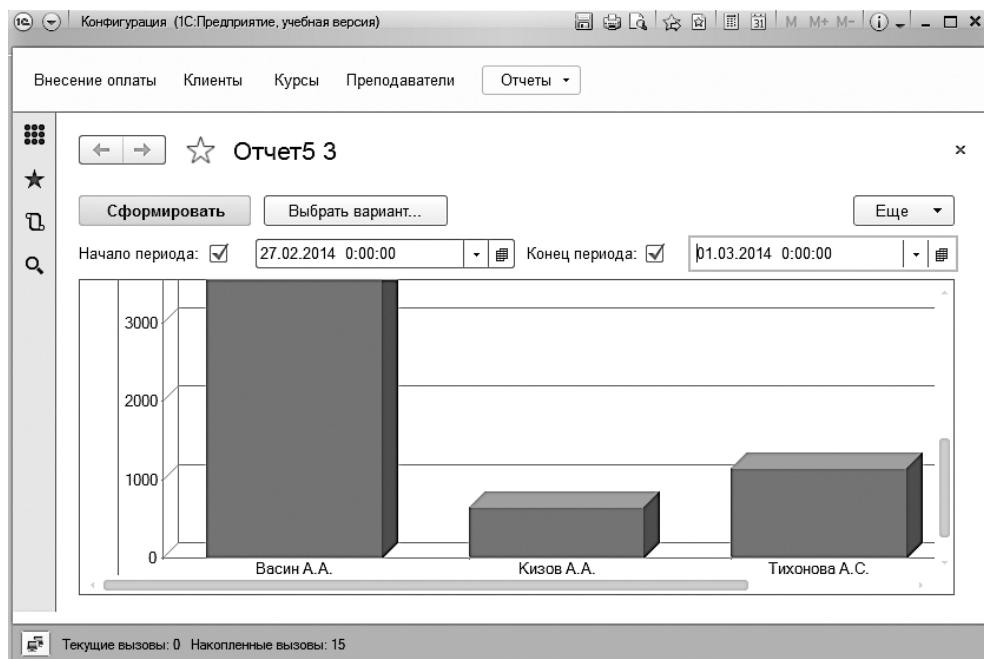


Рис. 5.29. Отчет с возможностью выбора интервала дат

Программирование форм

На платформе 1С:Предприятие 8.3 непосредственно само программирование на встроенным языке занимает далеко не самое важное место при создании прикладных решений. На первый план в этой новой версии программного продукта выходят разнообразные визуальные ресурсы. Тем не менее на платформе 1С:Предприятие 8.3 остаются участки, где программирование остается весьма востребованным, и одним из таких участков являются формы и элементы управления, расположенные на них. Так, для создания необходимых пользовательских сервисов в формах объектов (справочников, документов, регистров и т. д.) знание приемов программирования выходит на первый план.

В этом разделе мы на примере рассмотрим, как самостоятельно разработанные процедуры и функции можно использовать для обработки событий, связанных с формами и элементами управления, расположенными на них.

Сервер и клиентское приложение

Перед решением прикладных задач рассмотрим систему 1С:Предприятие 8.3 в структурном плане. Если в предыдущих главах это было не столь существенно, то сейчас такой обзор важен в плане концепции программирования на платформе 1С:Предприятие 8.3.

В новой платформе можно выделить два программных компонента:

- клиентское приложение;
- сервер 1С:Предприятие.

Клиентское приложение представляет собой программу, которая обеспечивает интерактивное взаимодействие системы с пользователем. Данная программа может выполнять программный код на встроенным языке, и в этом случае принято говорить, что код исполняется *в контексте клиентского приложения*.

Сервер 1С:Предприятие представляет собой один из рабочих процессов системы. Он обеспечивает взаимодействие клиентского приложения с хранилищем данных (базой данных). Сервер 1С:Предприятие также может выполнять программный код, и в этом случае можно сказать, что *код исполняется на сервере*.

Важно отметить, что на сервере и на клиенте доступны разные свойства, методы и объекты встроенного языка. Если речь идет о визуальном (интерактивном) взаимодействии пользователя с формой одного из объектов (справочника, документа, регистра и т. д.), то такие программные разработки осуществляются на клиенте. Другая ситуация касается извлечения информации из базы данных. Это действие уже осуществляется на сервере.

Рассмотрим, как выглядит взаимодействие клиентской и серверной частей на платформе 1С:Предприятие 8.3. Так, начало работы пользователя всегда связано с клиентской частью — пользователь запускает программное приложение.

На следующем этапе производится подключение к базе данных — здесь уже в работу включается сервер 1С:Предприятие. После этого управление опять передается клиенту. Подобная передача управления от клиента серверу и обратно происходит постоянно на протяжении всего периода работы пользователя.

Программирование при наличии двух компонентов (сервера 1С:Предприятие и клиентского приложения) уже нельзя рассматривать как единое целое. Кроме того, у сервера и клиента доступны разные объекты, свойства и методы встроенного языка.

Данные особенности организации платформы 1С:Предприятие 8.3 имеют большое значение особенно при программировании форм. Дело в том, что форма как программный объект существует одновременно и на сервере, и на клиенте.

ПРИМЕЧАНИЕ

Рассмотренные программные составляющие системы (сервер 1С:Предприятие и клиентское приложение) могут размещаться как на одном компьютере, так и на разных.

Таким образом, в новой версии платформы имеются программные компоненты форм, которые присутствуют на сервере, а есть компоненты, которые расположены в клиентской части приложения. В связи с этим при создании программной процедуры или функции, работающей с формой, мы должны предварительно указать системе, для какой из двух частей она предназначается. В терминологии разработчиков системы 1С:Предприятие можно сказать, что все процедуры и функции в модуле формы должны иметь явное указание на то, в каком контексте они будут исполняться [7]. Для этого все процедуры и функции в модуле формы должны иметь в своем программном коде директивы компиляции. Таких директив в платформе 1С:Предприятие 8.3 несколько:

- &НаКлиенте;
- &НаСервере;
- &НаСервереБезКонтекста;
- &НаКлиентеНаСервереБезКонтекста.

Если процедура или функция начинается с директивы **&НаКлиенте**, это говорит о том, что ее выполнение должно производиться в контексте клиентского приложения. Данная директива используется для всех обработчиков клиентских событий формы (например, события изменения значения в поле выбора на форме).

Директива **&НаСервере** говорит о том, что выполнение последующего программного кода происходит на сервере. Данная директива используется для всех обработчиков серверных событий формы. Также эту директиву программисты могут применять в собственных процедурах и функциях, чтобы передать исполнение программного кода на сервер.

Директива **&НаСервереБезКонтекста** говорит о том, что выполнение последующего программного кода происходит на сервере. Однако при этом контекст формы (реквизиты, элементы и параметры формы) для следующей после директивы части программы оказывается недоступен. Данная директива также используется программистами в собственных процедурах, чтобы передать исполнение программного кода на сервер.

Если процедура или функция начинается с директивы **&НаКлиентеНаСервереБезКонтекста**, это говорит о том, что ее выполнение может производиться как в контексте клиентского приложения, так и на сервере. Заметим, что данная директива используется очень редко.

Формы

Форма создается платформой, исходя из заданного нами в режиме конфигуратора описания (исходя из установок в окне редактирования объекта конфигурации). Для каждой формы на вкладке **Модуль** разработчик может располагать свои программы на встроенном языке. Эти программы система вызывает в определенные заранее известные моменты. Например, при изменении данных в определенном поле формы автоматически вызывается соответствующая процедура. Подобные процедуры и функции придают форме уникальные черты, свойственные именно ей.

С точки зрения пользователя (при работе в режиме 1С:Предприятие) форма представляет собой важный компонент интерфейса всего прикладного решения на платформе 1С:Предприятие 8.3, который позволяет просматривать имеющуюся информацию в информационной базе, а также вводить новые данные.

Если же взглянуть на форму глазами разработчика, можно выделить в ней несколько составляющих:

- элементы экранного диалога (элементы управления, расположенные на форме);
- модуль (место для размещения программ);
- реквизиты;
- параметры;
- команды.

Очень часто от форм не требуется какой-то особенной функциональности — вполне достаточно набора типовых действий при работе с формой. Для этого случая система 1С:Предприятие обеспечивает создание форм по умолчанию. В данной ситуации о формах можно даже не задумываться — система самостоятельно создаст ту форму, которая требуется в данный момент. Соответственно, стандартные действия, которые можно выполнить на форме в этом режиме, уже запрограммированы. Можно сказать, что необходимый стандартный «функционал» для работы пользователя создается системой автоматически. Так, имеются кнопки для создания нового элемента, записи и проведения документа и т. д.

Однако для реализации дополнительных сервисных возможностей требуется разработка собственных процедур и функций (о чем и рассказывается в данном разделе).

В отличие от версии 8.1 (и, разумеется, более ранних) в версиях 8.2 и 8.3 используется механизм управляемых форм. Его особенностью является передача задачи размещения элементов управления на форме от разработчика непосредственно самой системе. Заметим, что в предыдущих версиях системы программисту были доступны широкие возможности в плане расположения и редактирования элемен-

тов формы. Например, можно было с помощью визуального редактора изменить размеры элемента.

В следующем разделе мы рассмотрим пример разработки конфигурации, где нам потребуется реализовать дополнительный сервис для документа, связанного с поступлением товаров.

Разработка справочников и документа

В первой главе мы рассмотрели конфигурацию, в которой был реализован определенный «функционал» для заполнения табличной части документов Поступление Товаров и Продажа Товаров. А именно, при вводе (изменении) цены товара или количества единиц товара в табличной части этих документов производился автоматический пересчет суммы по текущей строке товара. Однако при этом саму цену товара (и поступления, и продажи) в документах мы вводили самостоятельно. Разумеется, в каких-то ситуациях это вполне подходит. Однако чаще всего цены по поступлению (продаже) товаров предварительно формируются в определенной структуре данных. Например, эти цены могут быть включены в справочник Товары в качестве дополнительных реквизитов. Кроме того, товары не всегда являются штучными, и в общем случае у каждого товара должен быть еще один реквизит – единица измерения (упаковки, килограммы, метры и т. д.). Соответственно, когда речь идет о цене в форме документа о поступлении (продажи) товаров, то цена понимается как цена за одну единицу товара.

Таким образом, при внесении в табличную часть документа очередной позиции товара должно производиться автоматическое заполнение (данными из справочника Товары) следующих полей документа:

- Цена;
- Единица Измерения.

Далее реализуем такой сервис для документа, связанного с учетом поступления товаров.

Для работы нам потребуется создать новую конфигурацию, которая будет частично похожа на уже знакомую разработку из первой главы. Так, в работе понадобятся следующие справочники:

- Организации;
- Единицы Измерения;
- Товары.

В справочнике Организации укажем один дополнительный реквизит – Адрес. В качестве типа данных данного реквизита определим строку (100 символов).

Справочник Единицы Измерения оформим без дополнительных реквизитов, а в справочнике Товары таких реквизитов будет три:

- ЕдиницаИзмерения (тип данных — СправочникСсылка.ЕдиницыИзмерения);
- ЦенаПоступления (тип данных — число целого типа);
- ЦенаПродажи (тип данных — число целого типа).

ПРИМЕЧАНИЕ

Для определения количества знаков числа после запятой в окне Свойства реквизита следует воспользоваться разделом Точность.

После создания справочников перейдем к разработке документа Поступление. На рис. 5.30 представлена информация, которую необходимо внести на вкладке Данные в окне редактирования объекта конфигурации. Здесь в разделе Реквизиты мы добавили один реквизит Организация (тип данных — СправочникСсылка.Организации), кроме того, в создаваемом документе Поступление разместили табличную часть Перечень. Состав этой табличной части выглядит так:

- Товар (тип данных — СправочникСсылка.Товары);
- ЕдиницаИзмерения (тип данных — СправочникСсылка.ЕдиницыИзмерения);
- Цена (тип данных — число целого типа);
- Количество (тип данных — число целого типа);
- Сумма (тип данных — число целого типа).

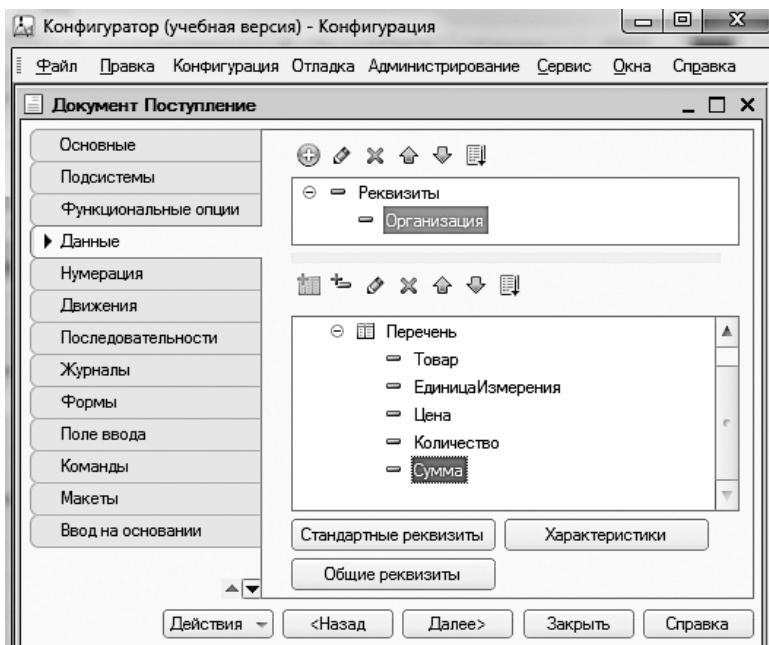


Рис. 5.30. Окно редактирования документа Поступление

В связи с тем, что далее мы собираемся программировать форму документа Поступление, следующий шаг заключается в создании пользовательской формы данного объекта конфигурации. Для этого в окне редактирования документа Поступление перейдем на вкладку Формы, где воспользуемся конструктором форм для создания пользовательской формы документа (рис. 5.31).

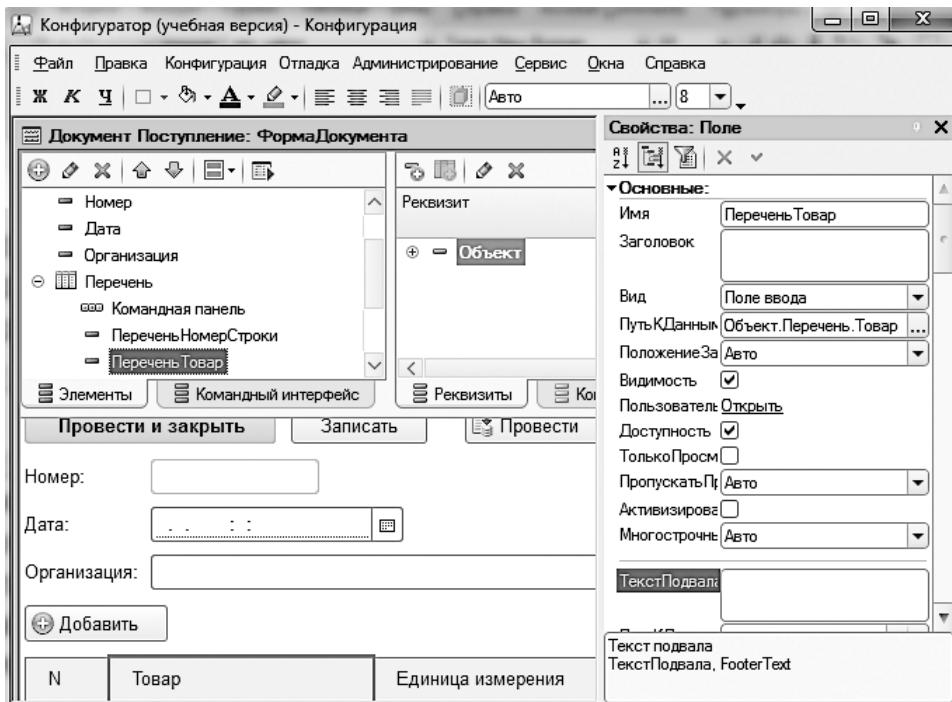


Рис. 5.31. Форма документа Поступление в режиме конфигуратора

Программирование табличной части в форме документа

Теперь наша задача заключается в том, чтобы обеспечить заполнение полей ПереченьЕдиницаИзмерения и ПереченьЦена при изменении в поле ПереченьТовар. Другими словами, когда пользователь в режиме 1С:Предприятие выбирает товар в соответствующем справочнике, то данные в поля ПереченьЕдиницаИзмерения и ПереченьЦена должны подставляться автоматически, исходя из информации, присутствующей в справочнике Товары.

ПРИМЕЧАНИЕ

Предфикс Перечень (название табличной части) к реквизитам табличной части формы документа Поступление добавляется системой автоматически.

Для реализации такого сервиса необходимо сначала открыть окно свойств поля ПереченьТовар (см. рис. 5.31). Это проще всего выполнить двойным щелчком на данном поле. После этого в окне свойств следует перейти к разделу События и щелкнуть на значке лупы для события ПриИзменении (рис. 5.32).

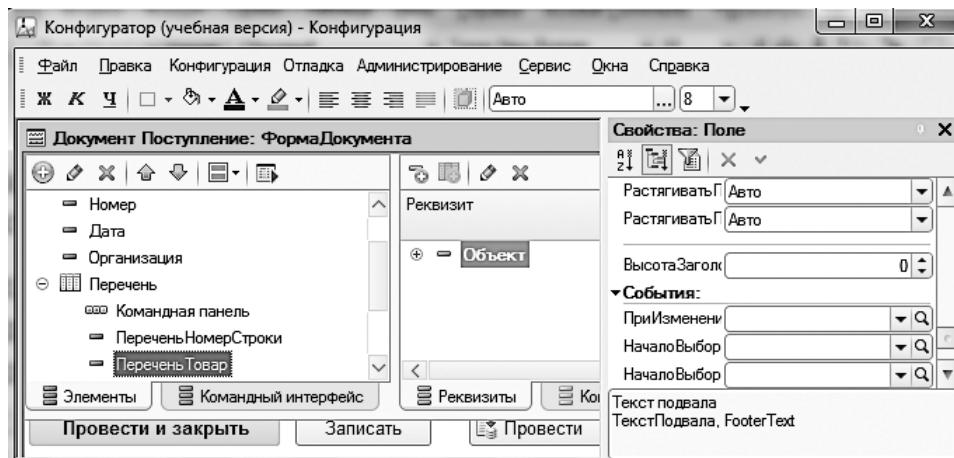


Рис. 5.32. Раздел События поля ПереченьТоваров

В результате этого действия на экране появляется окно с группой переключателей (рис. 5.33). Выбор того или иного переключателя позволяет далее сформировать необходимую заготовку программной процедуры.

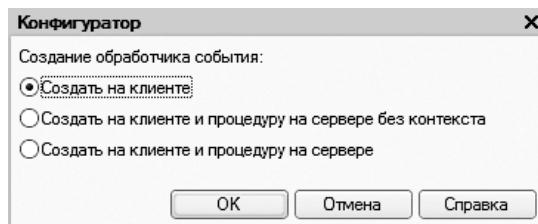


Рис. 5.33. Окно для формирования обработчика события

В результате этих действий на экране появится окно для написания программного кода обработчика выбранного события. Здесь следует разместить текст процедуры, которая представлена в листинге 5.5.

Листинг 5.5. Обработчик события ПриИзменении для поля ПереченьТовар

```
&НаКлиенте
Процедура ПереченьТоварПриИзменении(Элемент)
Стр=Элементы.Перечень.Текущие данные;
```

продолжение ↗

Листинг 5.5 (продолжение)

```
Стр.ЕдиницаИзмерения= ИзвлечьЕдиницуИзмерения(Стр.Товар);
Стр.Цена= ИзвлечьЦену(Стр.Товар);
КонецПроцедуры
```

Процедура начинается директивой **&НаКлиенте**, указывающей на то, что последующий программный код необходимо выполнять на стороне клиента. В тексте процедуры мы воспользовались функциями **ИзвлечьЕдиницуИзмерения** и **ИзвлечьЦену**. Это наши пользовательские функции, которые нам потребуется создать. Так, первая из них извлекает значение реквизита **ЕдиницаИзмерения** из справочника **Товары** для указанного товара в строке документа. Другая функция извлекает значение реквизита **ЦенаПоступления** из справочника **Товары**, которое относится к указанному товару. В эти функции в качестве параметра передается ссылка на элемент справочника **Товары**, который выбран пользователем в текущей строке документа (при заполнении поля **Товар** в табличной части документа).

Тексты функций **ИзвлечьЕдиницуИзмерения** и **ИзвлечьЦену** приведены соответственно на листингах 5.6 и 5.7. Эти функции, так же как и процедуру **Перечень-ТоварПриИзменении**, следует ввести на вкладке **Модуль** окна формы документа. Обе функции выполняются на сервере, о чем говорит предваряющая их директива **&НаСервере**.

Листинг 5.6. Функция для получения единицы измерения выбранного товара

```
&НаСервере
Функция ИзвлечьЕдиницуИзмерения(Товар)
    Возврат Товар.ЕдиницаИзмерения;
КонецФункции
```

Листинг 5.7. Функция для получения цены поступления выбранного товара

```
&НаСервере
Функция ИзвлечьЦену(Товар)
    Возврат Товар.ЦенаПоступления;
КонецФункции
```

Если теперь перейти в режим 1С:Предприятие и открыть форму документа **Поступление** для заполнения его табличной части данными о товарах, то мы увидим, что при выборе товара происходит автоматическое заполнение полей **ЕдиницаИзмерения** и **Цена** (рис. 5.34).

В двух серверных функциях, **ИзвлечьЕдиницуИзмерения** и **ИзвлечьЦену**, мы задействовали директиву **&НаСервере**. Однако в данном случае контекст формы не используется на сервере, поэтому передавать лишние данные на сервер нет смысла. В этом случае эффективнее применить директиву **&НаСервереБезКонтекста**. С учетом этого рассмотренные процедуры правильнее оформить в соответствии с листингами 5.8 и 5.9.

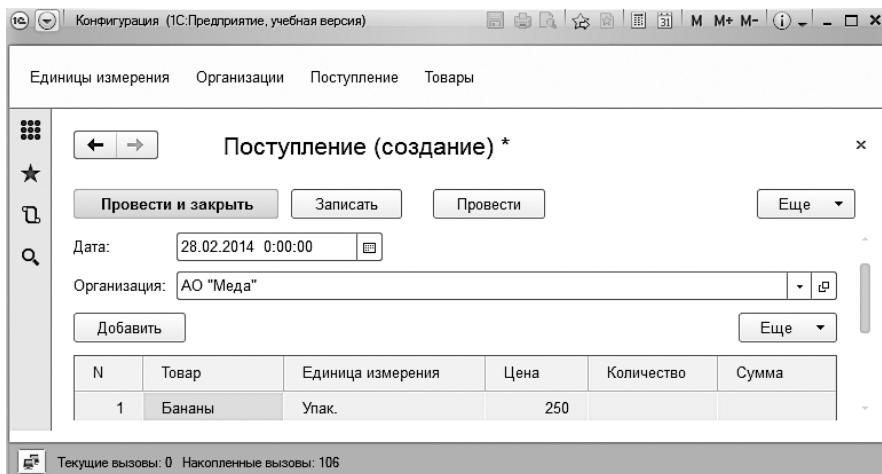


Рис. 5.34. Заполнение полей документа Поступление после выбора товара

Листинг 5.8. Использование директивы &НаСервереБезКонтекста для извлечения единицы измерения

```
&НаСервереБезКонтекста
Функция ИзвлечьЕдиницуИзмерения(Товар)
    Возврат Товар.ЕдиницаИзмерения;
КонецФункции
```

Листинг 5.9. Использование директивы &НаСервереБезКонтекста для извлечения цены поступления

```
&НаСервереБезКонтекста
Функция ИзвлечьЦену(Товар)
    Возврат Товар.ЦенаПоступления;
КонецФункции
```

Кроме разработанного сервиса извлечения информации из справочника Товары требуется обеспечить также автоматический расчет в поле Сумма после ввода количества единиц товара. Для этого вернемся к редактированию формы документа Поступление в режиме конфигуратора. Здесь необходимо сначала открыть окно свойств поля ПереченьКоличество и перейти к разделу События, где щелкнуть на значке лупы для события ПриИзменении. В результате этих действий на экране откроется окно для написания программного кода обработчика данного события. Здесь следует разместить текст процедуры, которая представлена в листинге 5.10.

Листинг 5.10. Обработчик события ПриИзменении для поля Количество

```
&НаКлиенте
Процедура ПереченьКоличествоПриИзменении(Элемент)
    Стр=Элементы.Перечень.ТекущиеДанные;
    Стр.Сумма= Стр.Количество*Стр.Цена;
КонецПроцедуры
```

Если теперь перейти в режим 1С:Предприятие, открыть форму документа Поступление, выбрать товар и ввести количество единиц товара, то мы увидим, что сумма вычисляется автоматически (рис. 5.35).

N	Товар	Единица измерения	Цена	Количество	Сумма
1	Бананы	Упак.	250	10	2 500

Рис. 5.35. Вычисление поля Сумма для документа Поступление после выбора товара

Существует еще несколько вариантов решения данной задачи. Так, можно разместить на вкладке модуля формы документа универсальную функцию, которая будет выдавать значение реквизита по его имени и ссылке на товар в текущей строке табличной части документа. Текст такой функции под названием *ИзвлечьРеквизит* приведен в листинге 5.11.

Листинг 5.11. Универсальная функция для извлечения реквизита

```
&НаСервереБезКонтекста
Функция ИзвлечьРеквизит(Ссылка,ИмяРеквизита)
    Возврат Ссылка[ИмяРеквизита];
КонецФункции
```

С учетом этой новой функции необходимо по-другому оформить обработку события *ПриИзменении* для поля *Товар*. Новый вариант процедуры заполнения единицы измерения и цены после указания товара в табличной части формы документа приведен в листинге 5.12.

Листинг 5.12. Измененный обработчик события ПриИзменении для поля Товар

```
&НаКлиенте
Процедура ПереченьТоварПриИзменении(Элемент)
    Стр=Элементы.Перечень.Текущие данные;
    Стр.ЕдиницаИзмерения=ИзвлечьРеквизит(Стр.Товар, "ЕдиницаИзмерения");
    Стр.Цена=ИзвлечьРеквизит(Стр.Товар, "ЦенаПоступления");
КонецПроцедуры
```

Разработка команды для формирования скидки

Рассмотрим еще одну задачу, связанную с автоматизацией работы пользователя с документом Поступление. Допустим, что фирма-поставщик может предоставлять нам скидку при поставке товаров по определенному документу. Причин для скидок существует много, и их не имеет смысла обсуждать. Величина скидки также может быть различной.

Реализуем следующий функционал в форме документа Поступление: создадим на форме кнопку СкидкаНаОдинПроцент, щелчком на которой все цены и суммы в табличной части документа должны автоматически уменьшаться на один процент. Соответственно, при каждом последующем щелчке на данной кнопке цены и суммы по товарам также дополнительно уменьшаются еще на один процент. Фактически, в этом случае будет происходить автоматический пересчет табличной части документа.

Перейдем к практическим действиям. Для начала необходимо в форме документа Поступление создать команду СкидкаНаОдинПроцент. Для этого на дополнительной вкладке Команды в окне редактирования формы воспользуемся значком Добавить (рис. 5.36).

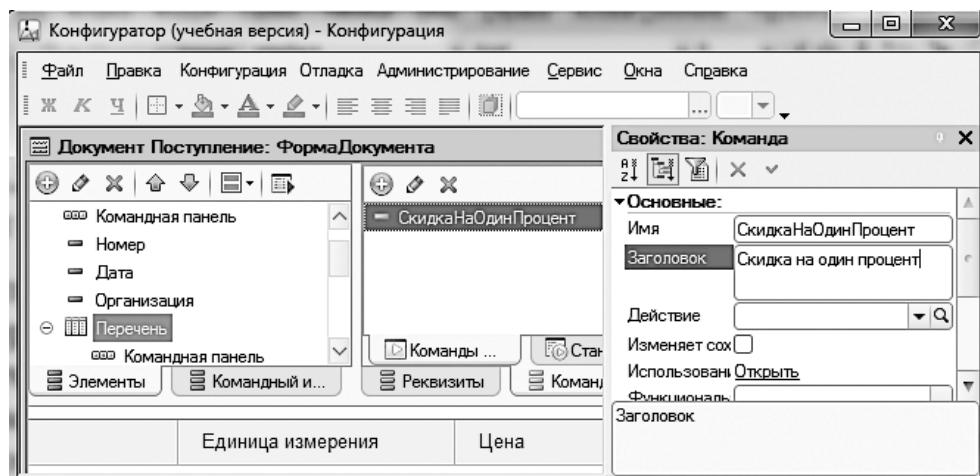


Рис. 5.36. Создание новой команды

После указания имени команды в окне свойств перейдем к разделу Действие, где щелчком на значке лупы перейдем в окно редактора программного кода команды. Текст процедуры для данной команды представлен в листинге 5.13. Здесь в цикле производится последовательный перебор строк табличной части документа и при каждом проходе цикла цена в очередной строке уменьшается на один процент. По-

сле этого с учетом новой цены пересчитывается сумма по товару в текущей строке табличной части документа.

Листинг 5.13. Процедура для команды СкидкаНаОдинПроцент

```
&НаКлиенте
Процедура СкидкаНаОдинПроцент(Команда)
    Для Каждого ТекСтрокаПеречень Из Объект.Перечень Цикл
        ТекСтрокаПеречень.Цена=ТекСтрокаПеречень.Цена*0.99;
        ТекСтрокаПеречень.Сумма=
            ТекСтрокаПеречень.Цена*ТекСтрокаПеречень.Количество;
    КонецЦикла
КонецПроцедуры
```

Таким образом, мы создали команду, которая выполняет пересчет всех строк в табличной части документа. Теперь необходимо разместить на форме элемент управления, с помощью которого пользователь данную команду сможет выполнять. По логике для этого лучше всего подходит кнопка (в режиме 1С:Предприятие на ней достаточно щелкнуть мышью).

Технические действия для создания кнопки на форме документа, связанной с разработанной командой, достаточно просты. Для этого в окне редактирования формы документа нужно перетащить мышью созданную команду на элемент Командная панель. В результате система самостоятельно добавит на форму элемент управления для выполнения разработанной команды (рис. 5.37).

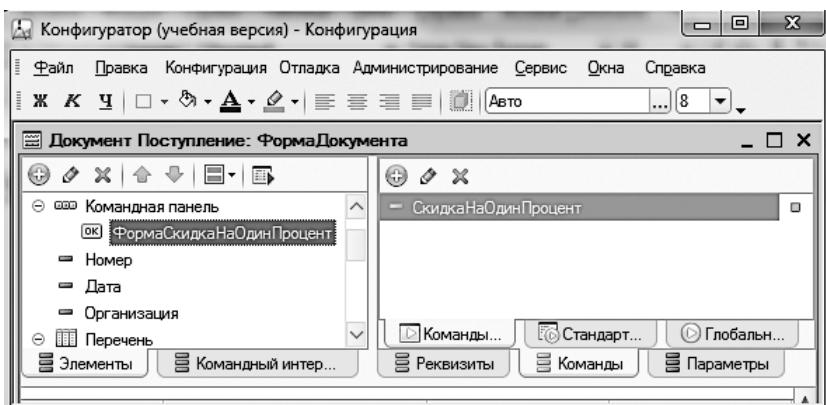


Рис. 5.37. Добавление кнопки на форму документа

Теперь все готово и можно проверить разработку на практике. Для этого перейдем в режим 1С:Предприятие. Каждый щелчок на новой кнопке приводит к уменьшению всех цен на один процент и соответствующему пересчету сумм в табличной части (рис. 5.38).

Поступление (создание) *

Номер:

Дата: 28.02.2014 0:00:00

Организация: АО "Меда"

N	Товар	Единица измерения	Цена	Количество	Сумма
1	Бананы	Упак.	248	10	2 480
2	Яблоки	Кг.	35	5	175
3	Кофе	Кг.	297	10	2 970

Текущие вызовы: 0 Накопленные вызовы: 45

Рис. 5.38. Формирование скидки в документе по поступлению товаров

Программное открытие формы

В предыдущих разделах мы на примерах познакомились с программированием элементов управления, расположенных на формах объектов. В этом разделе разберем прикладную задачу другого плана. Тема будет касаться программного открытия форм, но перед практическими действиями следует немного поговорить об организации форм в системе 1С:Предприятие 8.3.

Формы представляют собой специфические объекты конфигурации и содержат реквизиты, параметры и элементы. Для них в редакторе форм имеются соответствующие вкладки. Реквизиты формы предназначены для хранения данных, с которыми работает форма. Часть данных можно вводить и редактировать в элементах формы. К другой категории относятся данные, которые не отображаются, но используются формой в процессе работы.

Каждый элемент формы, который может изменять данные, связан с одним из реквизитов формы. В случае изменения данных в элементе формы соответствующие изменения происходят в реквизитах формы.

Параметры формы предназначены для управления функциональностью формы при ее открытии. Фактически они используются только в момент открытия формы.

ПРИМЕЧАНИЕ

Существует особый вид параметров — ключевые параметры. Такие параметры могут применяться уже после того, как форма создана.

В процессе открытия формы для нового (создание нового элемента справочника или нового документа) или существующего объекта происходит определенная последовательность программных вызовов. Так, при открытии формы нового объекта происходит следующая последовательность событий:

1. Обработка заполнения (выполняется в модуле объекта, форма которого открывается).
2. Создание на сервере (обрабатывается на сервере).
3. Открытие (обрабатывается на стороне клиента).

В случае открытия формы существующего объекта последовательность событий несколько другая:

1. Чтение на сервере.
2. Создание на сервере.
3. Открытие.

Для программного открытия формы используется метод `ОткрытьФорму`. Данный метод доступен только в контексте клиента и содержит ряд параметров, которые позволяют задать форму для открытия и определить свойства открываемой формы. Значения параметров, передаваемых форме, программист может установить самостоятельно в обработчике события формы `ПриСоздании` на сервере.

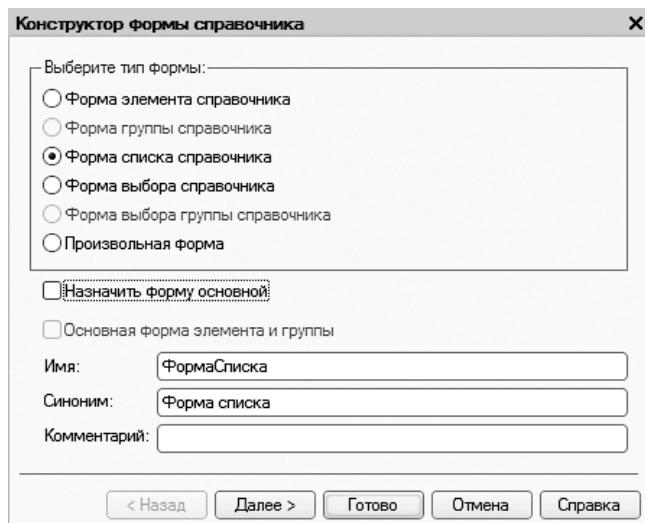


Рис. 5.39. Окно конструктора формы справочника

Теперь перейдем к примеру на данную тему. Скажем, в форме документа `Поступление` необходимо обеспечить удобный просмотр и редактирование цен по-

ступления товаров. Для этого мы создадим дополнительную кнопку на форме данного документа, щелчком на которой будет открываться форма справочника Товары, но без цен продажи. Для начала такую форму справочника Товары необходимо создать.

Технические действия по созданию новой формы начинаются с того, что в окне редактирования объекта конфигурации Товары следует перейти на вкладку Формы и с помощью конструктора форм создать форму списка. Обратим внимание на то, что создаваемую форму не надо назначать в качестве основной для данного справочника (рис. 5.39). В качестве имени новой формы укажем ФормаДляУстановкиЦенПоступления.

В процессе работы с конструктором формы следует включить в форму не все реквизиты, а только следующие: Код, Наименование, ЕдиницаИзмерения и ЦенаПоступления (рис. 5.40). В результате в режиме конфигуратора форма должна выглядеть так, как показано на рис. 5.41.

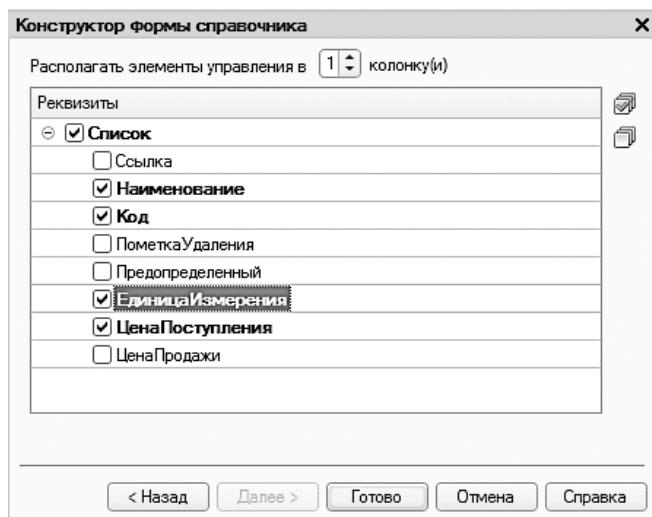


Рис. 5.40. Окно конструктора формы справочника

Теперь после разработки формы списка необходимо обеспечить ее вызов из формы документа Поступление. Для этого в режиме конфигуратора перейдем в окно редактирования формы данного документа. Здесь для начала нам необходимо создать новую команду, для чего на вкладке Команды следует воспользоваться кнопкой пиктографического меню для добавления новой команды, которую назовем ПросмотрТоваров (рис. 5.42). После этого щелкнем на значке лупы в разделе Действие и введем программную процедуру, соответствующую данной команде (листинг 5.14).

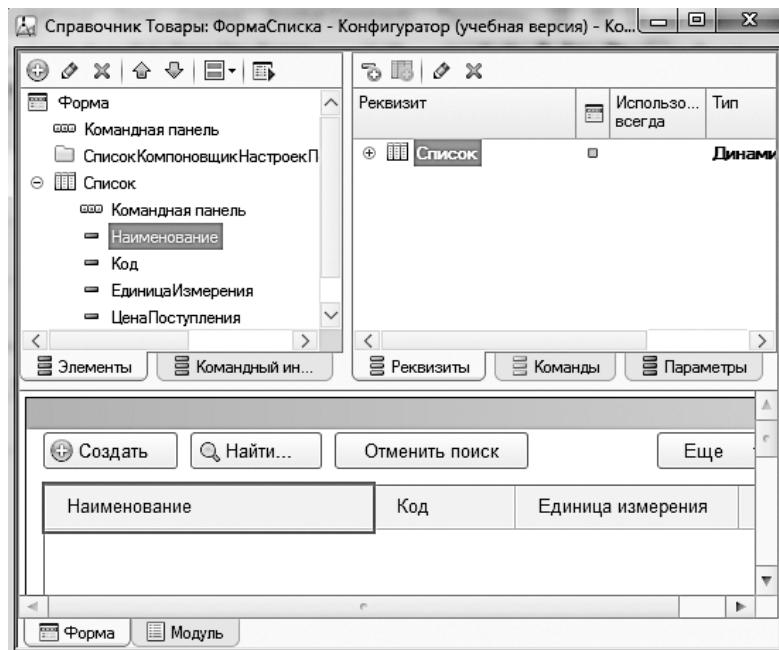


Рис. 5.41. Форма списка справочника Товары

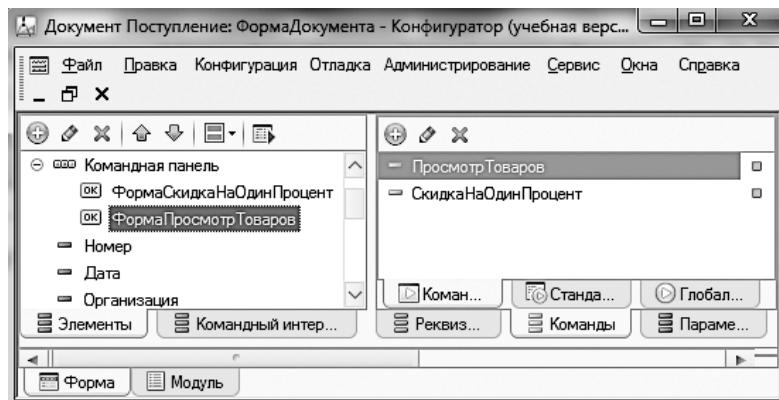


Рис. 5.42. Добавление команды в окне формы документа

Листинг 5.14. Процедура для команды для открытия формы справочника Товары

&НаКлиенте

Процедура ПросмотрТоваров(Команда)

ОткрытьФорму("Справочник.Товары.Форма.ФормаДляУстановкиЦенПоступления");
КонецПроцедуры

Теперь необходимо разместить на форме документа кнопку (элемент управления), чтобы обеспечить выполнение команды ПросмотрТоваров щелчком на кнопке. Для этого достаточно в окне редактора формы (см. рис. 5.42) перетащить мышью данную команду на элемент Командная панель. В результате на форме появится новый элемент управления, а его функциональность будет связана с разработанной командой ПросмотрТоваров.

Далее осталось перейти в режим 1С:Предприятие и открыть форму документа Поступление (рис. 5.43). На этой форме мы увидим новую кнопку Просмотр товаров, щелчком на которой открывается созданная форма списка справочника Товары.

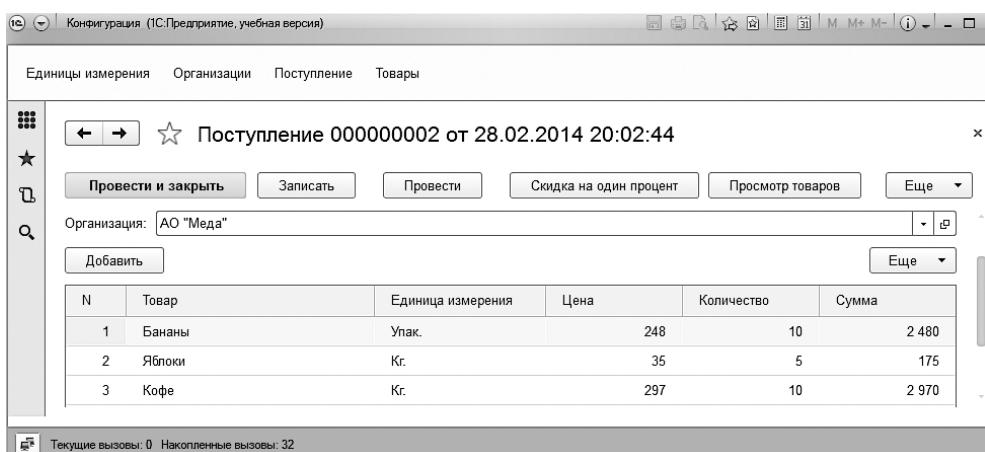


Рис. 5.43. Кнопка на форме документа Поступление

Использование динамического списка

Рассмотрим теперь более сложную задачу. Часто бывает так, что при работе с документами по поступлению товаров требуется обращать внимание не на весь перечень позиций, а только на определенную категорию, отличающуюся по цене или по каким-то другим параметрам. Для определенности будем считать, что нам необходимо в документе Поступление часто открывать список «дорогих» товаров (форму списка справочника Товары), цена которых за единицу измерения превышает 100 рублей. Как и в предыдущем примере, для решения задачи разместим на форме документа кнопку с соответствующим названием, обеспечивающую вывод списка условно дорогих товаров. На примере этой задачи мы продемонстрируем работу с динамическим списком.

Разработку можно разделить на несколько этапов. Для начала нам необходимо создать форму списка для справочника Товары. В настоящий момент кроме форм,

предлагаемых системой по умолчанию, существует одна пользовательская форма — **ФормаДляУстановкиЦенПоступления**. В режиме конфигуратора перейдем в окно редактирования справочника Товары. Здесь необходимо на вкладке Формы создать новую форму списка, которую назовем **ФормаДляДокумента** (смысл такого имени для формы справочника обусловлен тем, что далее мы ее будем открывать из документа). Обратим внимание на то, что данную форму не надо делать основной. В результате при работе со справочником обычным способом будет открываться форма списка, предлагаемая по умолчанию (другая здесь и не требуется), а из документа Поступление щелчком на нужной кнопке мы будем открывать форму с именем **ФормаДляДокумента**.

В процессе работы с конструктором формы в списке реквизитов укажем все имеющиеся реквизиты справочника, за исключением цены продажи. После несложных технических действий с конструктором мы получим результат, представленный на рис. 5.44.

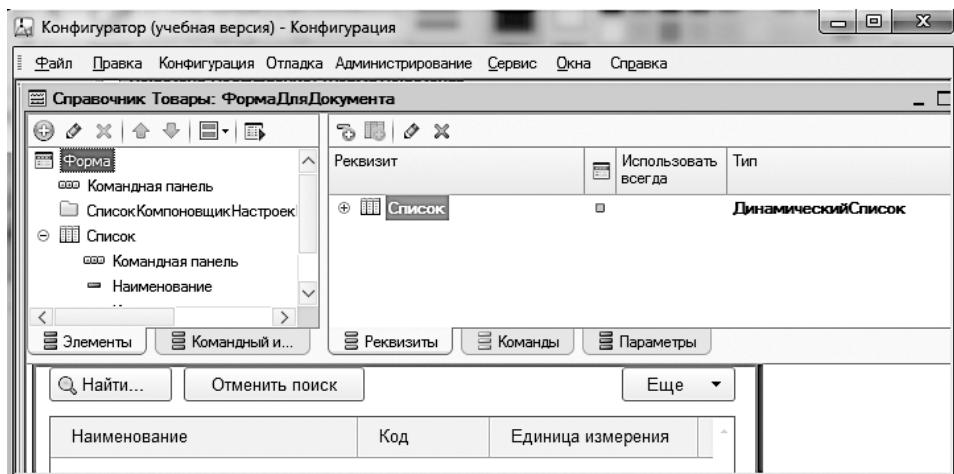


Рис. 5.44. Форма списка справочника Товары с именем ФормаДляДокумента

В правой верхней части формы, показанной на рисунке, присутствует реквизит **Список**. Щелкнув правой кнопкой мыши на этом реквизите, вызовем контекстное меню и выберем пункт **Свойства**, что позволит открыть на экране окно с аналогичным названием (рис. 5.45).

В данном окне свойств установим флајжок **Произвольный запрос**, после чего щелкнем на ссылке **Открыть** в строке **Настройка списка**. В результате на экране откроется окно **Динамический список**, в котором требуется сформировать новый текст запроса, представленный на рис. 5.46. После этого в данном окне остается щелкнуть на кнопке **OK**.

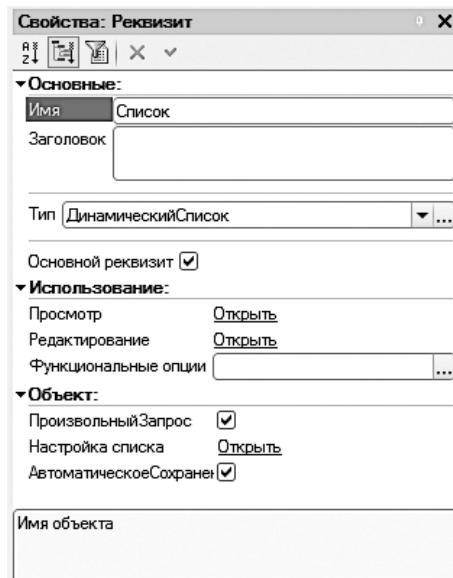


Рис. 5.45. Окно свойств реквизита Список

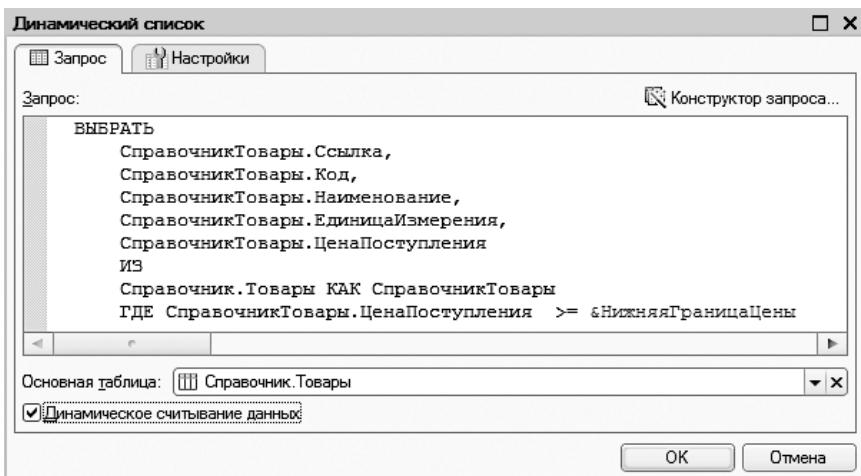


Рис. 5.46. Запрос для динамического списка

В данном запросе мы использовали параметр `НижняяГраницаЦены`. Для работы с рассматриваемой формой этот параметр должен быть установлен при открытии формы. Процесс передачи значений параметров в форму требует, чтобы сами параметры существовали в форме. Для создания параметра `НижняяГраницаЦены` перейдем на вкладку `Параметры` в окне редактирования формы (рис. 5.47) и сформируем этот

параметр с помощью кнопки Добавить. В качестве типа данных создаваемого параметра следует указать число целого типа.

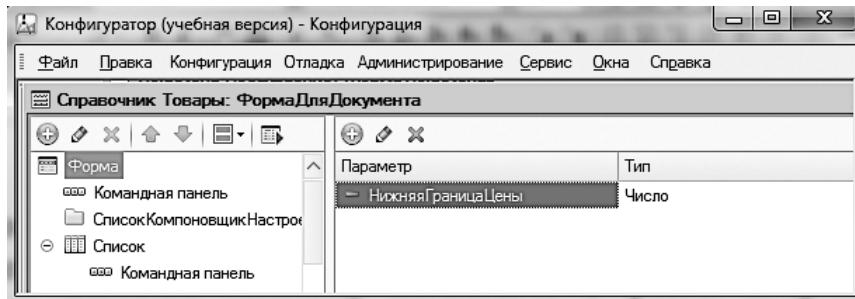


Рис. 5.47. Добавление параметра в форму списка с именем ФормаДляДокумента

На этом действия с формой справочника завершаются. Переидем теперь к доработке формы документа Поступление. Для начала нам потребуется создать новую команду с именем ДорогиеТовары (рис. 5.48), которая будет использована для открытия формы ФормаДляДокумента справочника Товары. Программный код данной команды приведен в листинге 5.15, в тексте которого мы установили нижнюю границу цены в 100 рублей. После создания команды требуется разместить на форме документа Поступление кнопку, с помощью которой пользователь ее сможет выполнить. Для этого достаточно перетащить команду ДорогиеТовары на элемент Командная панель.

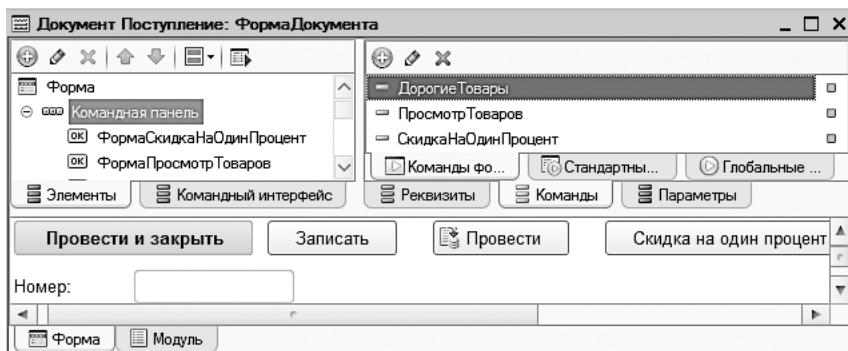


Рис. 5.48. Добавление команды ДорогиеТовары

Листинг 5.15. Процедура для команды ДорогиеТовары

```
&НаКлиенте
Процедура ДорогиеТовары(Команда)
ПараметрыФормы=Новый Структура("НижняяГраницаЦены", 30);
ОткрытьФорму("Справочник.Товары.Форма.ФормаДляДокумента", ПараметрыФормы);
КонецПроцедуры
```

Однако на этом разработка не заканчивается. Требуется опять вернуться к форме справочника **ФормаДляДокумента**, открыть окно ее свойств и задать обработчик события **ПриСозданииНаСервере** в соответствии с листингом 5.16.

Листинг 5.16. Обработчик события ПриСозданииНаСервере

```
&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
Список.Параметры.УстановитьЗначениеПараметра("НижняяГраницаЦены",
    Параметры.НижняяГраницаЦены);
КонецПроцедуры
```

На этом разработка заканчивается. Теперь можно перейти в режим 1С:Предприятие и убедиться в том, что после щелчка на кнопке **ДорогиеТовары** в форме документа **Поступление** открывается форма **ФормаДляДокумента** (рис. 5.49).

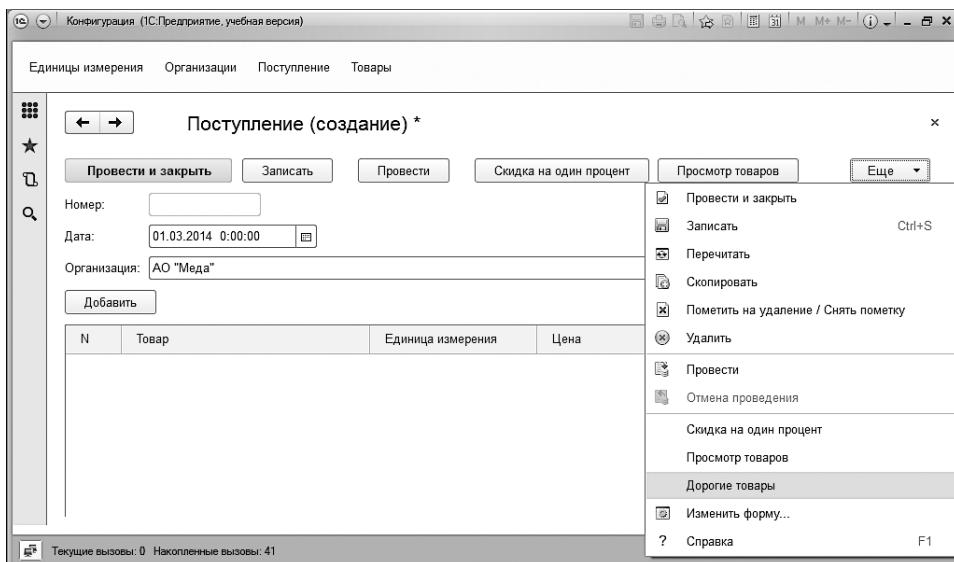


Рис. 5.49. Открытие перечня дорогих товаров в форме списка

Формирование сообщения для пользователя

В процессе создания справочника Организации мы сформировали дополнительный реквизит Адрес. Также в табличной части документа Поступления был размещен реквизит Организация с типом данных СправочникСсылка.Организации. Наша задача в данном разделе заключается в небольшой доработке документа Поступление, а именно — нам необходимо прямо из формы документа обеспечить просмотр адреса поставщика, указанного в поле Организация. В плане интерфейса такое ус-

вершенствование будет выглядеть так: после выбора поставщика его адрес должен появиться в окне сообщений. В результате и документ не будет перегружен дополнительными полями, и пользователь все увидит.

Приступим к разработке, и для начала нам потребуется в режиме конфигуратора открыть форму документа Поступление. Здесь мы создадим новую команду АдресФирмы, которая будет выводить информацию об адресе в окне сообщений. Текст данной команды представлен в листинге 5.17.

Листинг 5.17. Программный код команды АдресФирмы

```
&НаКлиенте
Процедура АдресФирмы(Команда)
Сообщение= Новый СообщениеПользователю;
Сообщение.Текст=ПолучитьАдрес(Объект.Организация);
Сообщение.Сообщить();
КонецПроцедуры
```

В тексте команды АдресФирмы нам необходимо воспользоваться серверной функцией ПолучитьАдрес, которую мы должны создать (листинг 5.18).

Листинг 5.18. Функция ПолучитьАдрес, выполняемая на сервере

```
&НаСервереБезКонтекста
Функция ПолучитьАдрес(Фирма)
Возврат Фирма.Адрес
КонецФункции
```

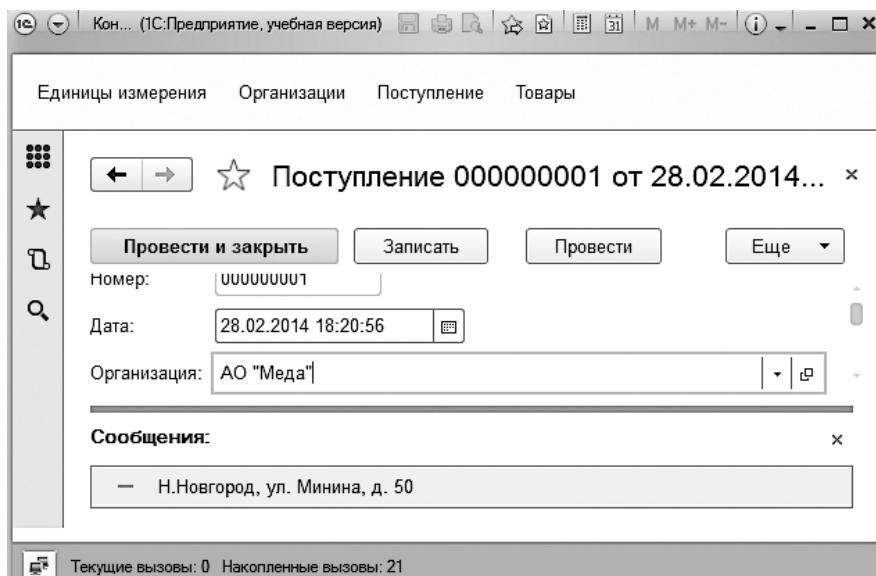


Рис. 5.50. Использование окна сообщений в форме документа

Далее необходимо создать на форме кнопку, щелчком на которой выполняется команда АдресФирмы. Технически это делается перетаскиванием данной команды на командную панель формы документа. После этого можно перейти в режим 1С:Предприятие и посмотреть, как работает созданная кнопка (рис. 5.50).

ПРИМЕЧАНИЕ

В процессе разработки справочника Товары был создан реквизит ЦенаПродажи. В качестве самостоятельного задания требуется создать соответствующий документ о продаже товаров и формы объектов, аналогичные тем, которые были разработаны для случая поступления товаров.

Заполнение формы документа данными из регистра сведений

В этом разделе мы рассмотрим еще одну задачу, связанную с программированием формы документа. Ее особенностью является использование регистра сведений, который является базовым объектом конфигурации платформы. Отметим, что регистр сведений появился в системе 1С:Предприятие, начиная с версии 8.0.

Еще один новый объект конфигурации, с которым мы встретимся в ходе решения прикладной задачи, — это перечисление. Наряду со знакомством с новыми базовыми объектами конфигурации в этом разделе мы продолжим изучение приемов программирования форм объектов в системе 1С:Предприятие 8.3.

Перечисления

Среди большого количества объектов конфигурации системы 1С:Предприятие наиболее простыми с точки зрения их организации являются *перечисления*, представляющие собой статические списки данных, которые создаются на этапе разработки и в дальнейшем просто используются (без возможности их изменения или дополнения в режиме 1С:Предприятие).

ПРИМЕЧАНИЕ

Перечисление предназначено для описания структуры постоянных наборов значений, не изменяемых в процессе работы конфигурации. На основе перечисления платформа создает в базе данных информационную структуру, в которой может храниться набор некоторых постоянных значений.

Перечисления в основном используются не самостоятельно, а в совокупности с другими типами объектов. Так, перечисления применяются для ввода значений

реквизитов справочников и документов. Основная цель при использовании перечислений — исключить неоднозначность вводимой информации.

Итак, первым объектом, который мы создадим в новой конфигурации, будет перечисление с именем **ВидыТарифов**, отражающее различные виды тарифов для образовательных программ. В практической работе варианты образовательных программ не подразумевают особого многообразия. Кроме того, для образовательных программ и, соответственно, видов тарифов характерна неизменность во времени (если и происходит добавление какого-то нового значения, то это случается очень редко).

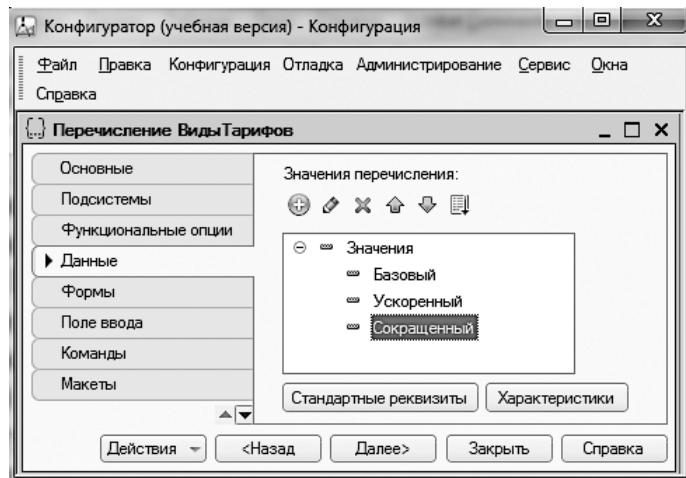


Рис. 5.51. Внесение значений перечисления ВидыТарифов

Для подобных ситуаций и был разработан объект конфигурации под названием **перечисление**. В нашей прикладной задаче в качестве значений перечисления **ВидыТарифов** определим следующие:

- Базовый;
- Ускоренный;
- Сокращенный.

Рассмотрим теперь технические действия, позволяющие реализовать сказанное на практике. Для начала требуется, находясь в режиме конфигуратора, открыть окно **Конфигурация**. В дереве объектов конфигурации необходимо щелкнуть правой кнопкой мыши на пункте **Перечисления** и воспользоваться уже знакомым нам контекстным меню. Здесь следует с помощью соответствующего пункта добавить новый объект, после чего перед нами откроется окно редактирования объекта конфигурации. Задание имени, как обычно, производится на вкладке **Основные**, а для того, чтобы внести в перечисление упомянутые значения, следует перейти на вкладку **Данные** (рис. 5.51).

Сформированные значения перечисления ВидыТарифов мы будем использовать далее в справочнике.

Справочник Учащиеся и документ Оплата

Для работы нам понадобится справочник Учащиеся, который также необходимо разработать. Кроме обязательных реквизитов любого справочника создадим еще один под названием ВидТарифа (тип значения — ПеречислениеСсылка.ВидыТарифов). Таким образом напротив фамилии учащегося мы будем отмечать вид тарифа по оплате обучения. На рис. 5.52 показана вкладка Данные нового объекта конфигурации, где и требуется добавить новый реквизит.

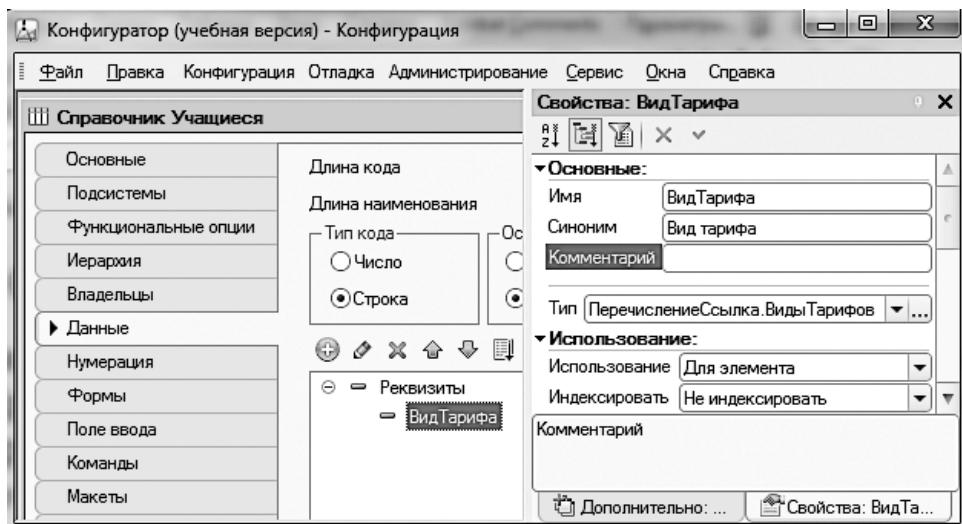


Рис. 5.52. Окно редактирования справочника Учащиеся

Еще в процессе работы нам понадобится документ Оплата. На вкладке Данные окна редактирования данного объекта конфигурации разместим табличную часть Перечень, включающую четыре реквизита (рис. 5.53):

- Учащийся (тип данных — СправочникСсылка.Учащиеся);
- БазоваяСумма (тип данных — число целого типа);
- Скидка (тип данных — число целого типа);
- Сумма (тип данных — число целого типа).

В дальнейшем наша задача заключается в программировании элементов, расположенных на форме документа, поэтому форма, генерируемая системой по умолчанию, нам не подойдет. Таким образом, на следующем шаге мы создадим пользовательскую форму документа. Для этого в окне редактирования документа Оплата перейдем

на вкладку Формы, где воспользуемся конструктором форм. Результат работы конструктора представлен на рис. 5.54.

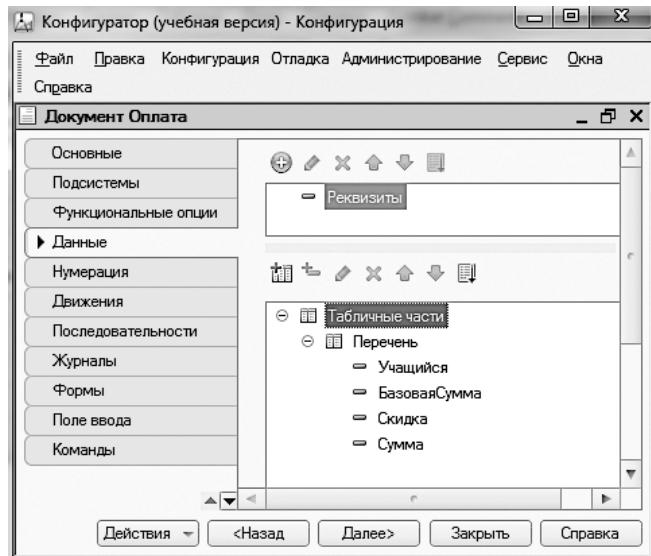


Рис. 5.53. Вкладка Данные окна редактирования документа Оплата

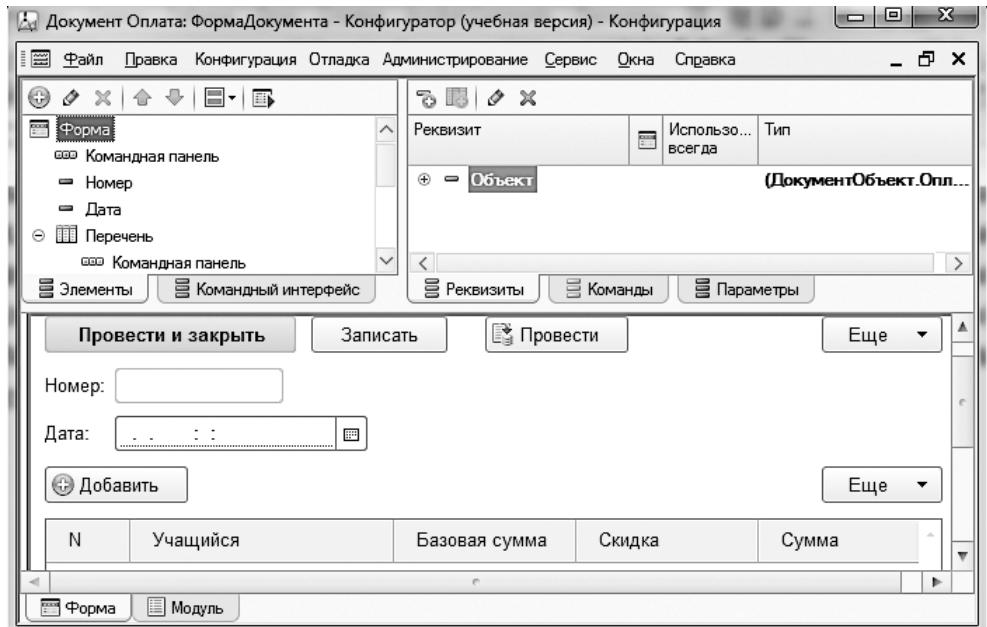


Рис. 5.54. Форма документа Оплата в режиме конфигуратора

Периодический регистр сведений

Далее нам потребуется не встречавшийся ранее в книге базовый объект конфигурации, который называется *регистр сведений*. Этот объект появился в системе 1С:Предприятие, начиная с версии 8 (в широко известной версии 7.7 его не было).

Ранее в справочнике Учащиеся мы разместили реквизит ВидТарифа, в котором фиксируется вариант тарифа для конкретного учащегося. Однако значение тарифа может меняться с течением времени, при этом в реальной работе может потребоваться ввод документа об оплате со старой датой, когда тариф отличался от текущего.

С точки зрения математических представлений можно трактовать значение тарифа как функцию от двух переменных, в качестве которых выступают вид тарифа и время. Для каждого уникального набора этих параметров должно существовать не более одного значения функции — значения тарифа. В этом и заключается предназначение регистров сведений — хранение информации по набору переменных (в системе 1С:Предприятие они называются *измерениями*), причем каждый набор не должен повторяться. При попытке повторного создания значения функции для уже использованного набора переменных (измерений) система обратит на это внимание.

Если взглянуть на регистр сведений с точки зрения его структуры, то мы увидим в нем три компонента:

- измерения;
- ресурсы;
- реквизиты.

Измерения — это как раз и есть аргументы функции, а *ресурсы* можно считать ее значениями. Несколько вспомогательную роль играют *реквизиты*, которые содержат дополнительную поясняющую информацию о рассматриваемом наборе данных. Например, в нашей ситуации реквизитом может быть фамилия ответственного лица, которое заносит информацию о значении тарифа.

Регистр сведений может быть непериодическим, в этом случае параметр (измерение) время отсутствует. В результате измерениями будут являться только те параметры, которые создаются непосредственно разработчиком в режиме конфигуратора.

ВНИМАНИЕ

Общий принцип организации регистра сведений — для каждого набора измерений должно существовать единственное значение ресурса.

В случае периодического регистра сведений его «периодичность» можно определить с помощью одного из следующих значений:

- в пределах секунды;
- в пределах дня;

- в пределах месяца;
- в пределах квартала;
- в пределах года.

Для определения подобной «периодичности» периодический регистр сведений всегда содержит поле Период, которое формируется системой автоматически. Оно имеет тип дата, и в нем фиксируется факт принадлежности записи регистра к какому-то периоду. Отметим, что в процессе записи данных в регистр сведений система 1С:Предприятие всегда приводит значение этого поля к началу того периода, в который он попадает. Например, если в регистр сведений с периодичностью в пределах месяца записать данные, в которых период указан как 07.05.14, то регистр сохранит эти данные со значением периода, равным 01.05.14.

В периодическом регистре сведений может быть только одна запись с определенным набором измерений и периодом. Система сама контролирует уникальность записей для регистра сведений. Ключом, однозначно идентифицирующим запись, является совокупность значений измерений регистра и периода.

ВНИМАНИЕ

В регистре сведений не может быть двух и более записей с одинаковыми ключами.

Таким образом, далее для решения поставленной задачи мы должны создать регистр сведений Тарифы, в котором будут храниться тарифы, развернутые во времени.

Приступим к техническим действиям по созданию периодического регистра сведений. Начало не вызывает вопросов — щелчок правой кнопкой мыши на элементе Регистры сведений в дереве объектов конфигурации с последующим созданием нового объекта конфигурации. После этого перед нами откроется вкладка Основные окна редактирования регистра сведений Тарифы (рис. 5.55).

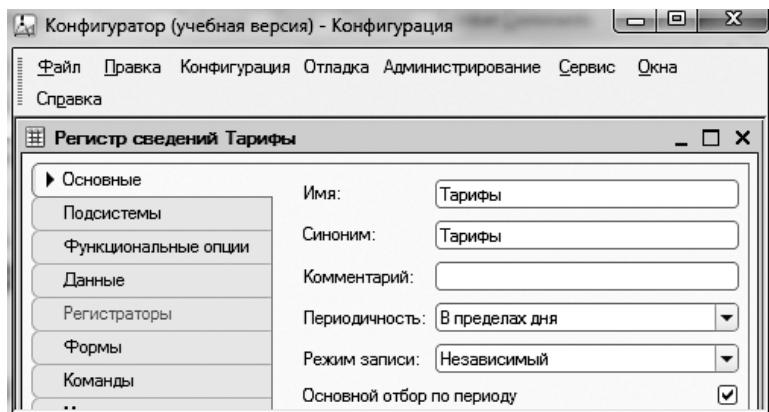


Рис. 5.55. Вкладка Основные окна редактирования регистра Тарифы

ПРИМЕЧАНИЕ

Для измерений регистра можно задать параметр Ведущее. Его смысл связан с назначением в качестве типа данных рассматриваемого измерения ссылки на объект базы данных. В этом случае запись в регистре сведений имеет смысл, только когда существует объект. И при удалении объекта записи с соответствующим значением измерения удаляются автоматически. Если запись в регистр сведений производится регистратором, то это означает, что записи регистра жестко подчиняются регистраторам (фактически — документам). В этом случае записи, как правило, порождаются при проведении документов. Регистратор же представляет собой ссылку на документ, который внес данную запись в регистр сведений. В случае, если для регистра сведений в режиме конфигуратора установлен параметр Независимый, то понятие регистратора смысла не имеет.

Периодичность регистра следует отметить на вкладке Основные в поле Периодичность. Для решаемой задачи разумно выбрать вариант В пределах дня, что позволит хранить изменения тарифов, даже если эти изменения будут происходить каждый день. Также на этой вкладке в качестве режима записи в рассматриваемый регистр выберем вариант Независимый. Данный параметр связан с тем, что информация в регистр сведений может вноситься двумя способами: вручную или из документов. Регистры сведений, которые заполняются независимо, могут редактироваться вручную (в режиме 1С:Предприятие).

После установки всех указанных параметров перейдем на вкладку Данные и создадим необходимые поля регистра (рис. 5.56):

- измерение ВидТарифа (тип данных — ПеречислениеСсылка.ВидыТарифов);
- ресурс ЗначениеТарифа (тип данных — число).

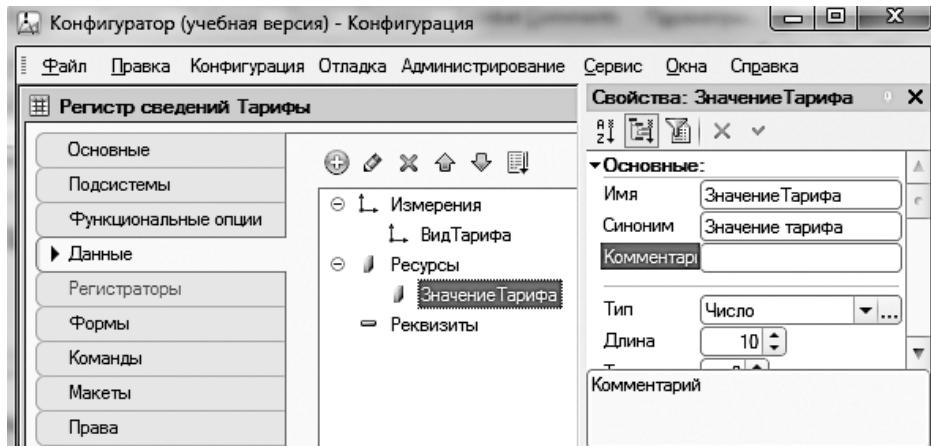


Рис. 5.56. Вкладка Данные окна редактирования регистра Тарифы

На этом разработка регистра сведений Тарифы в режиме конфигуратора завершается. Теперь можно перейти в режим 1С:Предприятие, где и внести несколько записей о значениях тарифов (рис. 5.57).

Период	Вид тарифа	Значение тарифа
03.01.2014	Базовый	9 800
07.02.2014	Базовый	10 000
03.03.2014	Базовый	10 200

Рис. 5.57. Форма списка регистра сведений Тарифы

Таким образом, мы установили значения конкретных видов тарифов на определенную дату.

Программирование документа

Дальнейшие наши действия будут связаны с программированием формы документа Оплата. Для этого откроем в режиме конфигуратора созданную ранее форму нашего документа (см. рис. 5.54). Сейчас задача заключается в разработке следующего сервиса при заполнении табличной части документа пользователем: после выбора учащегося в поле Учащийся должно автоматически заполняться поле БазоваяСумма. В данное поле должно вноситься актуальное значение из регистра сведений Тарифы. Для реализации данного функционала откроем окно свойств поля ПереченьУчащийся, где в разделе События следует обратиться к событию ПриИзменении (рис. 5.58). Далее щелчком на значке лупы откроем окно редактора программного кода, где требуется задать обработчик рассматриваемого события в соответствии с листингом 5.19.

Листинг 5.19. Обработка события ПриИзменении для поля ПереченьУчащийся

```
&НаКлиенте
Процедура ПереченьУчащийсяПриИзменении(Элемент)
    Стр=Элементы.Перечень.ТекущиеДанные;
    Стр.БазоваяСумма=ОбщиеФункции.ТарифНадень(Объект.Дата,Стр.Учащийся);
КонецПроцедуры
```

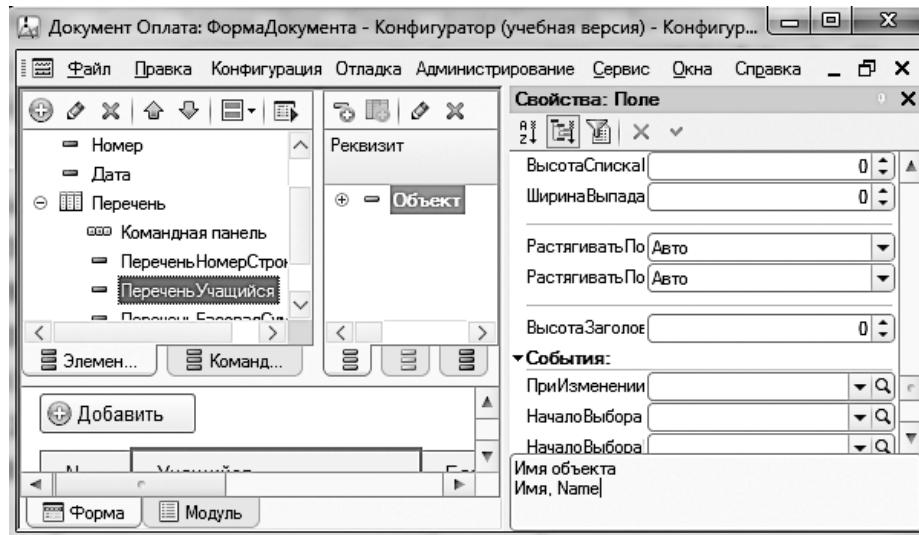


Рис. 5.58. Окно свойств поля ПереченьУчащийся

В данную процедуру заложено использование сервисной функции **ТарифНадень** из общего модуля, которую нам также нужно разработать. Для начала необходимо создать сам общий модуль под названием **ОбщиеФункции**. Для этого в окне Конфигурация откроем ветку **Общие** и обратимся к разделу **ОбщиеМодули**. Здесь с помощью правой кнопки мыши вызовем контекстное меню и добавим новый общий модуль, которому дадим название **ОбщиеФункции**. Обратим внимание на флажки, которые необходимо установить в процессе диалога с системой при создании модуля (рис. 5.59). Во-первых, должен быть установлен флажок **Сервер**, обеспечивающий компиляцию данного модуля на стороне сервера. Во-вторых, необходимо установить флажок **Вызов сервера**, чтобы далее создаваемую экспортную функцию этого модуля можно было вызывать с клиента.

В окне редактора программного кода поместим входящую в созданный модуль функцию **ТарифНадень** (листинг 5.20), к которой мы уже обращались в процедуре **ПереченьУчащийсяПриИзменении** (см. листинг 5.19).

Листинг 5.20. Общая функция ТарифНадень из модуля ОбщиеФункции

```
Функция ТарифНадень(ДатаПлаты, Учащийся) Экспорт
    ТарифУчащегося=Учащийся.ВидТарифа;
    Отбор=Новый Структура("ВидТарифа", ТарифУчащегося);
    ИнформацияИзРегистра=
        РегистрыСведений.Тарифы.ПолучитьПоследнее(ДатаПлаты, Отбор);
    Возврат ИнформацияИзРегистра.ЗначениеТарифа;
КонецФункции
```

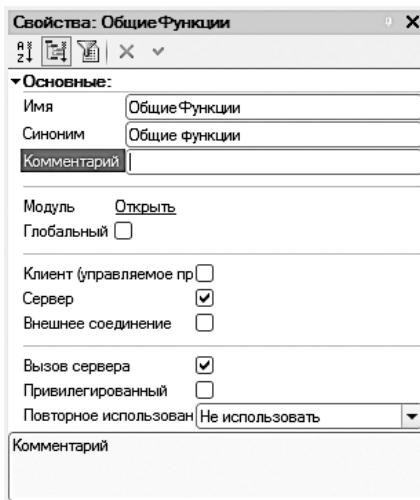


Рис. 5.59. Окно свойств модуля ОбщиеФункции

Для получения значения тарифа в функцию передается два параметра:

- ДатаПлаты** — определяет момент времени, на который нас интересует значение тарифа (тип данных — **Дата**);
- Учащийся** — ссылка на элемент справочника Учащиеся, для которого мы собираемся определить значение тарифа за обучение.

Получение вида тарифа выбранного учащегося производится с помощью имени интересующего реквизита **ВидТарифа** в справочнике Учащиеся:

ТарифУчащегося=Учащийся.ВидТарифа

Далее в тексте разработанной функции мы использовали объект **Отбор** типа **Структура**, которая представляет собой коллекцию пар. Каждая пара состоит из имени переменной (ключа) и значения переменной, которое нас устраивает при дальнейшем отборе. С помощью объекта **Отбор** мы показываем, что нас интересуют записи регистра **Тарифы** с определенным видом тарифа. Имя ключа структуры (**ВидТарифа**) должно совпадать с именем измерения регистра **Тарифы**, а значение элемента структуры **ТарифУчащегося** задает отбираемое по данному измерению значение.

После этого производится обращение к менеджеру регистра сведений **Тарифы — РегистрыСведений.Тарифы**. Здесь используется метод **ПолучитьПоследнее()**, который возвращает значение ресурса самой поздней записи регистра на указанную дату (**ДатаПлаты**) для интересующего вида тарифа (**ТарифУчащегося**).

Метод **ПолучитьПоследнее()** возвращает в переменной **ИнформацияИзРегистра** структуру, содержащую значения ресурсов регистра (в нашем случае ресурс у регистра один — **ЗначениеТарифа**).

В результате созданная функция занимает свое место в модуле ОбщиеФункции (рис. 5.60).

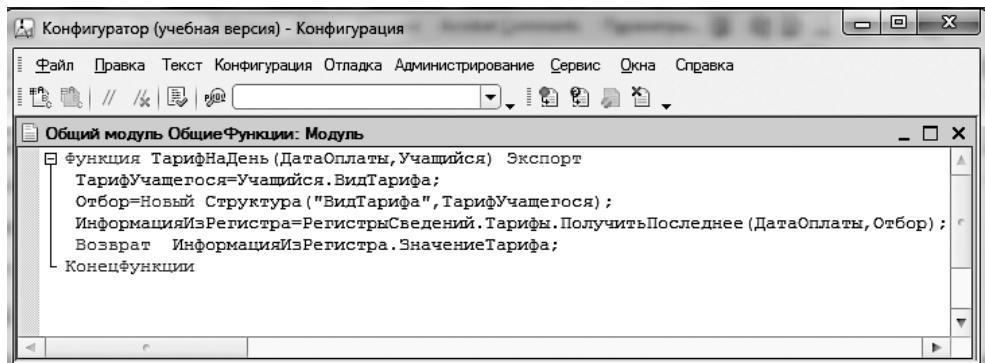


Рис. 5.60. Модуль ОбщиеФункции

Теперь, когда программные конструкции для автоматической подстановки актуального тарифа в документ Оплата готовы, можно перейти в режим 1С:Предприятие для проверки работоспособности сделанных изменений. Как видно, после внесения данных об учащемся при заполнении документа Оплата происходит автоматическая подстановка актуального значения тарифа (рис. 5.61).

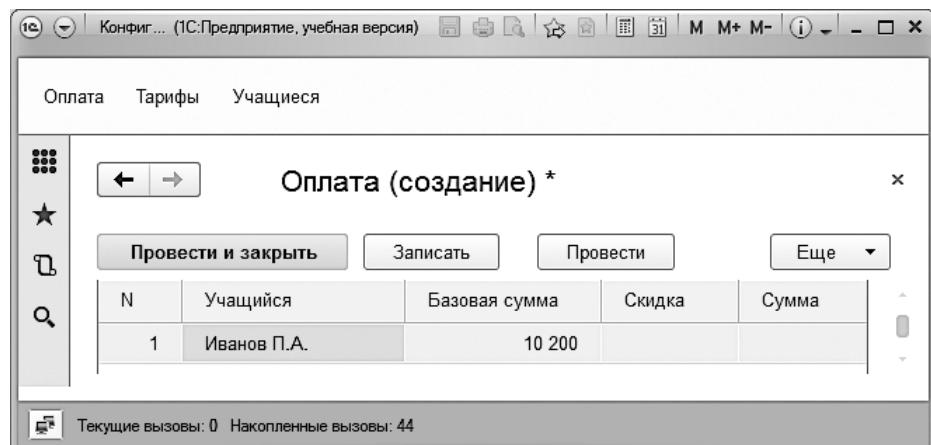


Рис. 5.61. Автоматизация заполнения поля Базовая сумма в документе Оплата

Однако на этом доработка рассматриваемого документа не завершается. Нам требуется обеспечить автоматическое вычисление суммы после ввода скидки для конкретного учащегося. Для этого необходимо запрограммировать событие При-Изменении для поля ПереченьСкидка. Технические действия для этого аналогичны

рассмотренным ранее для поля ПереченьУчащиеся. Необходимая процедура автоматического заполнения поля Сумма после ввода скидки приведена в листинге 5.21.

Листинг 5.21. Автоматизация вычисления суммы после ввода скидки

```
НаКлиенте
Процедура ПереченьСкидкаПриИзменении(Элемент)
    Стр=Элементы.Перечень.ТекущиеДанные;
    Стр.Сумма=Стр.БазоваяСумма-Стр.Скидка;
КонецПроцедуры
```

Результат проведенной автоматизации документа можно проверить, перейдя в режим 1С:Предприятие (рис. 5.62).

Просмотр тарифов в форме документа Оплата

Еще немного доработаем документ Оплата. Так, по логике работы для пользователя неплохо бы реализовать возможность просмотра списка тарифов прямо в документе при внесении оплаты. Как показывает практика, в реальной работе это требуется достаточно часто. Перед разработкой приведем результат, который собираемся получить (рис. 5.63). Здесь мы видим на форме документа Оплата кнопку Просмотр тарифов, щелчком на которой открывается форма регистра Тарифы.

Вернемся в режим конфигуратора и приступим к необходимой доработке документа Оплата. Сейчас следует открыть уже имеющуюся форму данного документа в редакторе форм и создать новую команду, которую назовем ПросмотрТарифов (рис. 5.64).

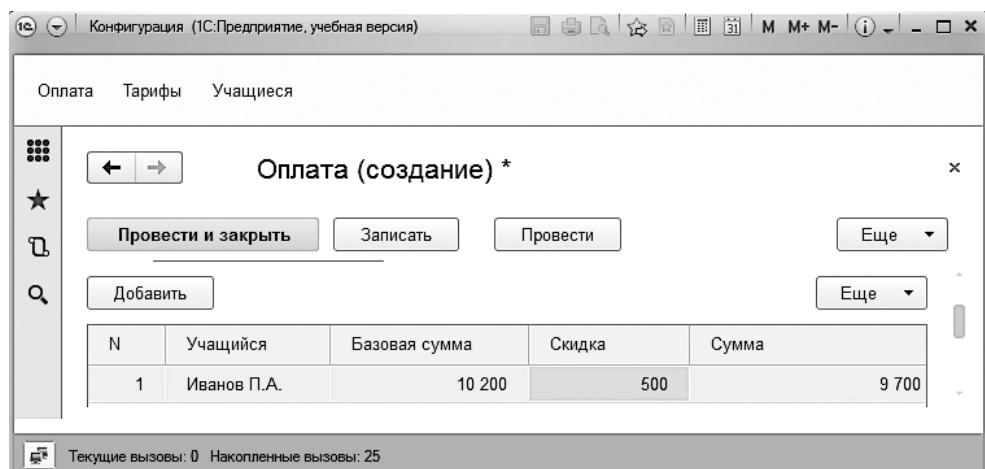


Рис. 5.62. Результат автоматизации в форме документа Оплата

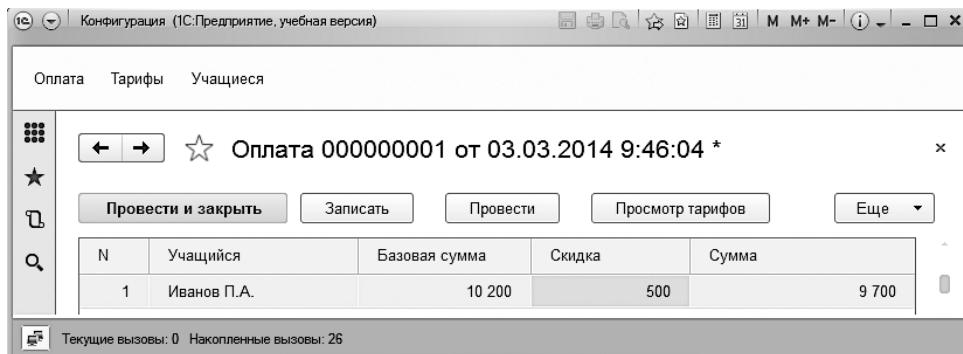


Рис. 5.63. Кнопка для просмотра тарифов в форме документа Оплата

В окне свойств данной команды обратимся к разделу **Действие**, где, щелкнув на значке лупы, перейдем в окно написания программного кода для данной команды. Здесь требуется ввести программу открытия формы регистра сведений (листинг 5.22).

Листинг 5.22. Программный код команды ПросмотрТарифов

```
&НаКлиенте
Процедура ПросмотрТарифов(Команда)
    ОткрытьФорму(«РегистрСведений.Тарифы.ФормаСписка»);
КонецПроцедуры
```

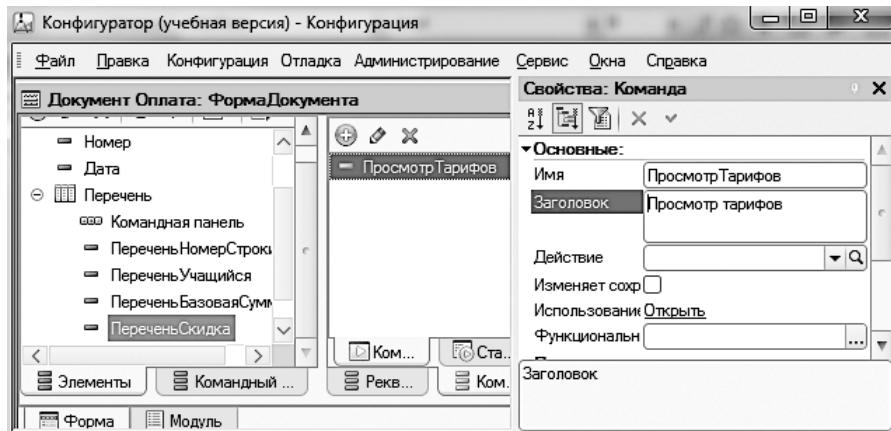


Рис. 5.64. Добавление команды в форму документа Оплата

Осталось разместить кнопку на форме документа для выполнения разработанной команды. Для этого проще всего мышью перетащить команду на элемент **Командная панель** (рис. 5.65). Теперь, когда все готово, можно перейти в режим 1С:Предприятие и проверить работоспособность сделанных изменений.

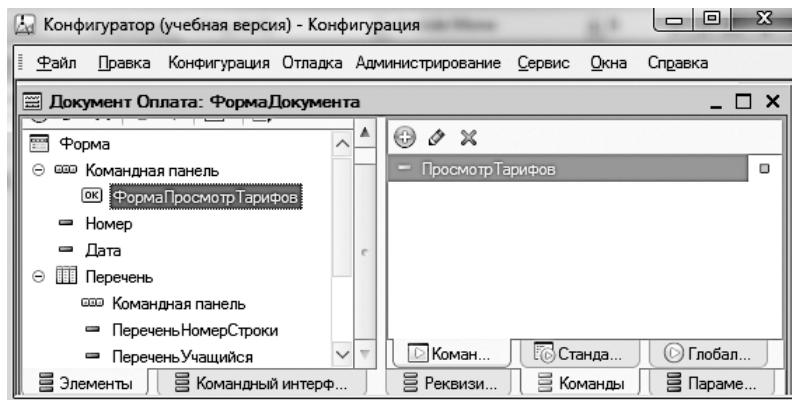


Рис. 5.65. Создание кнопки для выполнения команды в форме документа

Автоматизация для учета денежных средств

Рассмотрим процедуру создания конфигурации, воспользовавшись для этого приемами программирования и средствами визуальной разработки, с которыми мы познакомились в предыдущих разделах книги. В частности, мы рассмотрим такой важный механизм, предлагаемый системой 1С:Предприятие, как *ввод на основании*. Он позволяет значительно упростить работу с документами, так как новые документы можно создавать на основании уже имеющейся информации в базе данных. В результате за счет исключения повторения одних и тех же действий сокращается время, затрачиваемое пользователями на ввод данных в информационную базу.

Практическая задача будет касаться построения информационной системы для учета внесения денежных средств фирмами-покупателями. Будем считать, что с каждой фирмой у нас заключен договор и по договору нам поэтапно перечисляются некие суммы. Прикладное решение должно обеспечивать учет внесенных по каждому договору сумм.

Как и ранее, начнем работу в режиме конфигуратора с создания необходимых справочников. После этого нам понадобятся документы, а основным результатом разработки будут отчеты, которые мы разработаем с помощью системы компоновки данных.

Разработка справочников

Создадим новую конфигурацию. В процессе работы нам потребуются справочники Клиенты и Договоры, поэтому следующим объектом конфигурации будет справочник Клиенты. В нем, кроме автоматически формируемых кода и наименования, дополнительных реквизитов не потребуется. После включения данного справочника в созданную подсистему перейдем в режим 1С:Предприятие и заполним справочник Клиенты названиями нескольких фирм (рис. 5.66).

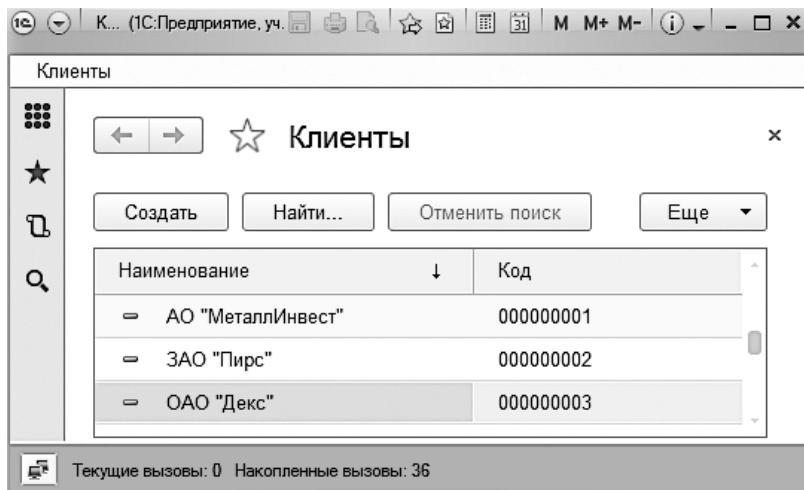


Рис. 5.66. Форма списка справочника Клиенты

Другой справочник, который нам понадобится, – это Договоры (рис. 5.67). Создадим в нем два дополнительных реквизита:

- Клиент (тип данных – СправочникСсылка.Клиенты);
- СуммаПодДоговору (тип данных – число целого типа).

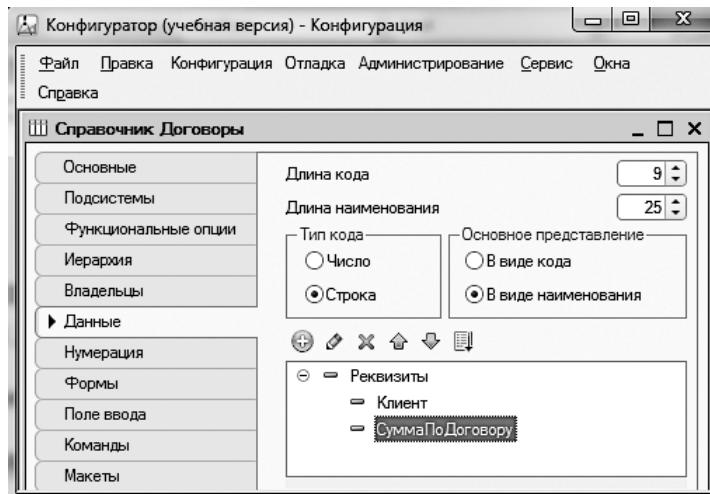


Рис. 5.67. Вкладка Данные окна редактирования справочника Договоры

Теперь можно перейти в режим 1С:Предприятие и заполнить справочник Договоры информацией по нескольким договорам, которые нам потребуются в дальнейшей работе (рис. 5.68).

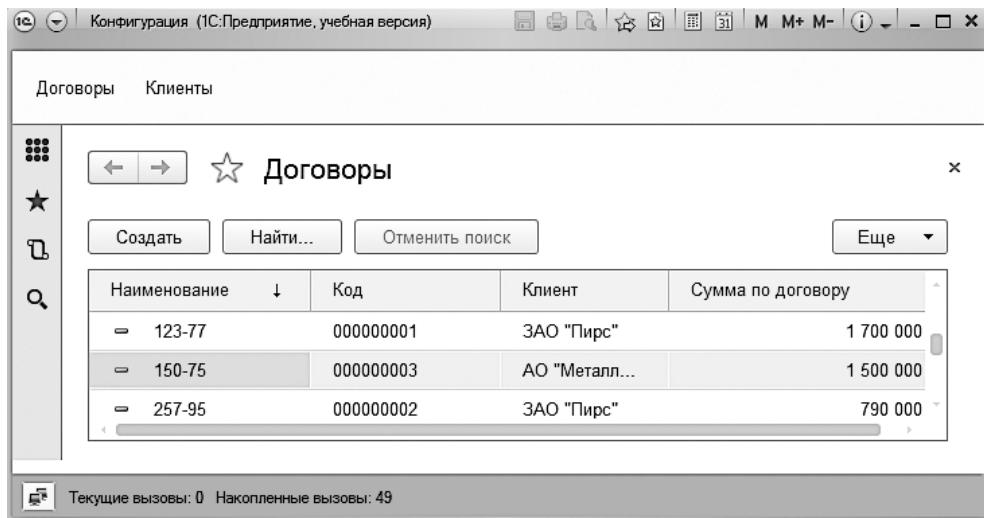


Рис. 5.68. Заполнение справочника Договоры

Создание документа и регистрация накопления

Предполагается, что наши клиенты (зарегистрированные в справочнике Клиенты) являются покупателями и перечисляют нам денежные суммы по каждому договору поэтапно. В связи с этим требуется разработать документ, который будет фиксировать факты этих оплат. Технические действия для создания нового документа (объекта конфигурации) нам хорошо знакомы по предыдущим главам, поэтому проектировщиком только основные моменты разработки документа ОплатаПодДоговору. Так, на вкладке Данные в окне редактирования объекта конфигурации создадим два реквизита в основной части документа (рис. 5.69):

- Договор (тип данных — СправочникСсылка.Договоры);
- СуммаОплаты (тип данных — число).

Табличная часть в документе ОплатаПодДоговору нам не понадобится, поэтому после работы в режиме конфигуратора (разумеется, еще необходимо включить созданный документ в имеющуюся в нашей конфигурации подсистему) перейдем в режим 1С:Предприятие для заполнения нескольких документов по оплатам. В результате в форме списка данного документа будет присутствовать некий набор записей (рис. 5.70).

Документ ОплатаПодДоговору фиксирует факт оплаты по конкретному договору с клиентом, а для регистрации и учета оплаченных сумм создадим регистр накопления ПеречисленныеСуммы. На вкладке Основные этого нового объекта конфигурации укажем, что данный регистр будет являться регистром оборотов. После

этого перейдем на вкладку **Данные**, где определим одно измерение и один ресурс регистра (рис. 5.71).

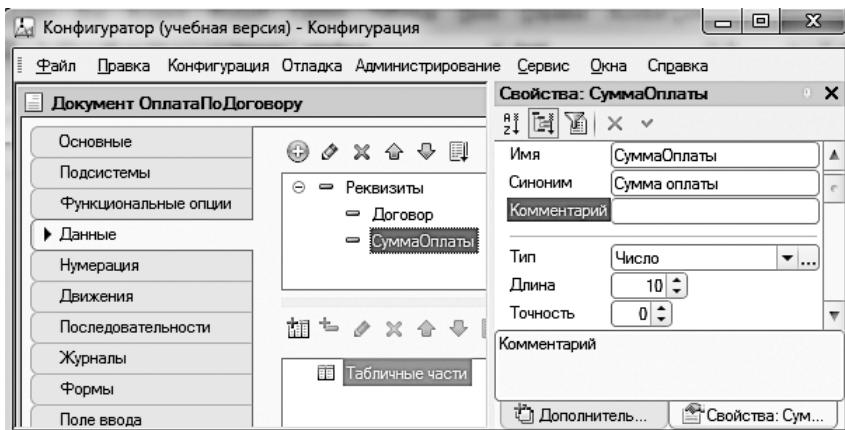


Рис. 5.69. Вкладка Данные документа ОплатаПодоговору

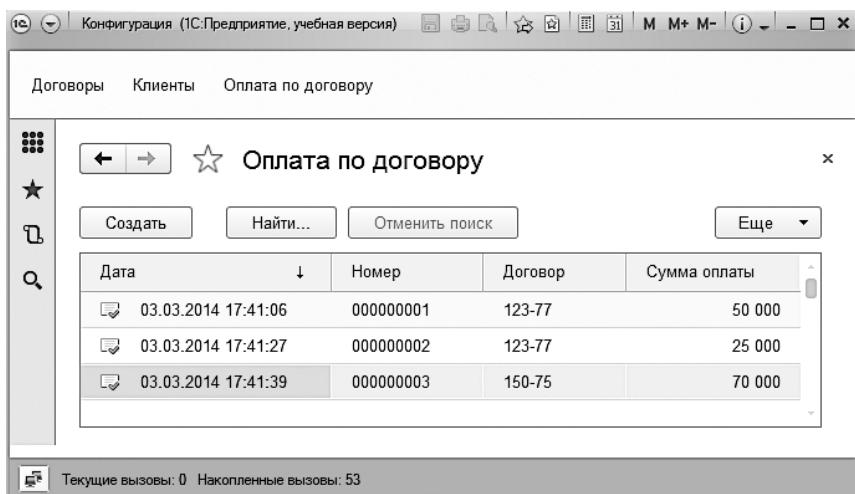


Рис. 5.70. Форма списка документа ОплатаПодоговору

Для измерения ДоговорПокупателя укажем тип данных — СправочникСсылка.Договоры, а ресурс СуммаОплаты будет иметь числовой тип данных.

Теперь необходимо обеспечить движения по регистру накопления ПеречисленныеСуммы. Для этого вернемся в окно редактирования документа ОплатаПодоговору и перейдем на вкладку **Движения**, где воспользуемся конструктором движений (рис. 5.72).

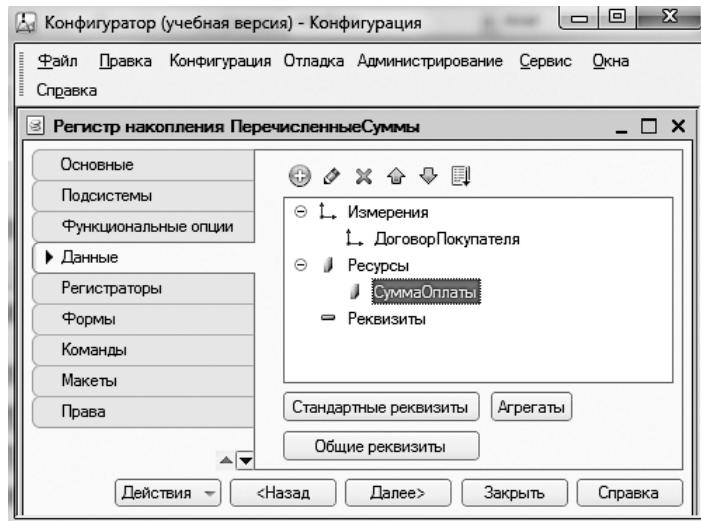


Рис. 5.71. Вкладка Данные окна редактирования регистра ПеречисленныеСуммы

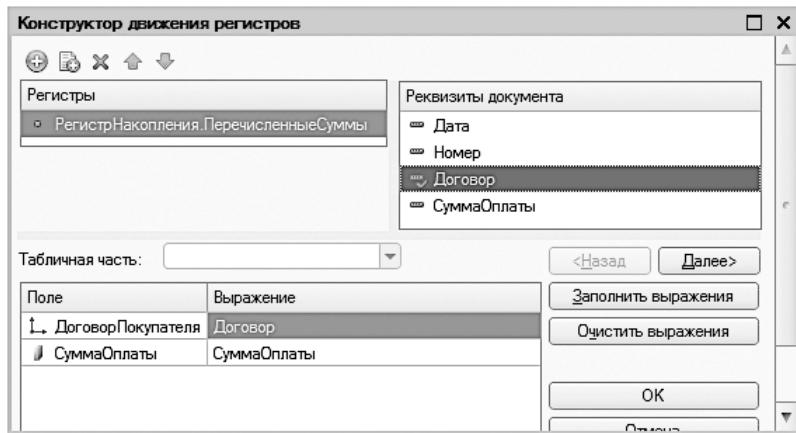


Рис. 5.72. Окно конструктора движений регистра ПеречисленныеСуммы

После щелчка на кнопке OK конструктор движений обеспечит автоматическое формирование процедуры, осуществляющей изменения в регистре ПеречисленныеСуммы при проведении документа ОплатаПодоговору (листинг 5.23).

Листинг 5.23. Процедура формирования движений по регистру ПеречисленныеСуммы

```
Процедура ОбработкаПроведения(Отказ, Режим)
//{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
//  данный фрагмент построен конструктором.
Движения.ПеречисленныеСуммы.Записывать = Истина;
//  регистр ПеречисленныеСуммы
```

```

Движение = Движения.ПеречисленныеСуммы.Добавить();
Движение.Период = Дата;
Движение.ДоговорПокупателя = Договор;
Движение.СуммаПлаты = СуммаПлаты;
//}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры

```

Теперь все готово для осуществления проводок в режиме 1С:Предприятие, и после проведения нескольких документов ОплатаПоДоговору в регистре накопления появятся соответствующие записи (рис. 5.73).

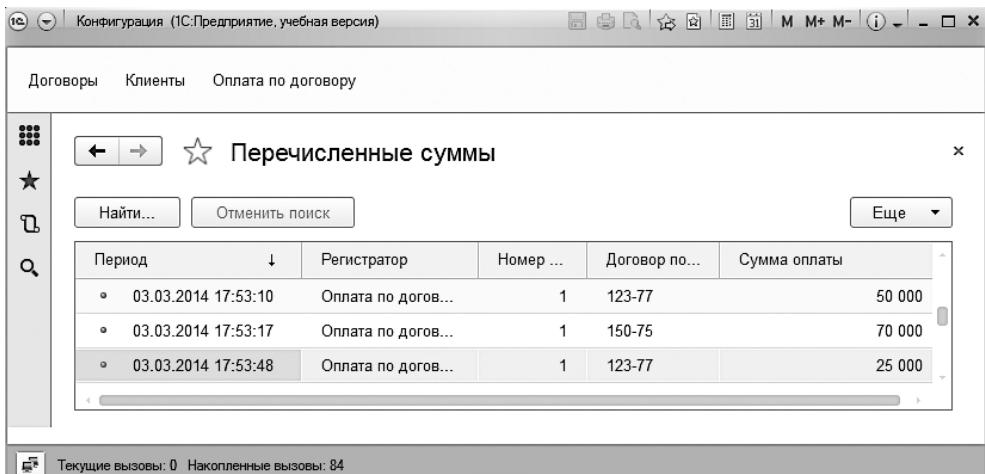


Рис. 5.73. Форма списка регистра ПеречисленныеСуммы

Разработка отчета по детальным оплатам

Разрабатываемый отчет должен позволять легко получить важную информацию по оплатам (как суммарную по каждому договору, так и детальную по каждой внесенной сумме). В качестве имени отчета укажем Отчет5_4, а в окне конструктора схемы компоновки воспользуемся знакомым пунктом меню Добавить набор данных — запрос. Далее перейдем к работе с конструктором запроса, в окне которого выберем две таблицы (рис. 5.74):

- таблица справочника Договоры;
- таблица регистра ПеречисленныеСуммы.

Из данных таблиц в раздел Поля перенесем три поля:

- Договоры.Ссылка;
- ПеречисленныеСуммы.СуммаПлаты
- ПеречисленныеСуммы.Период.

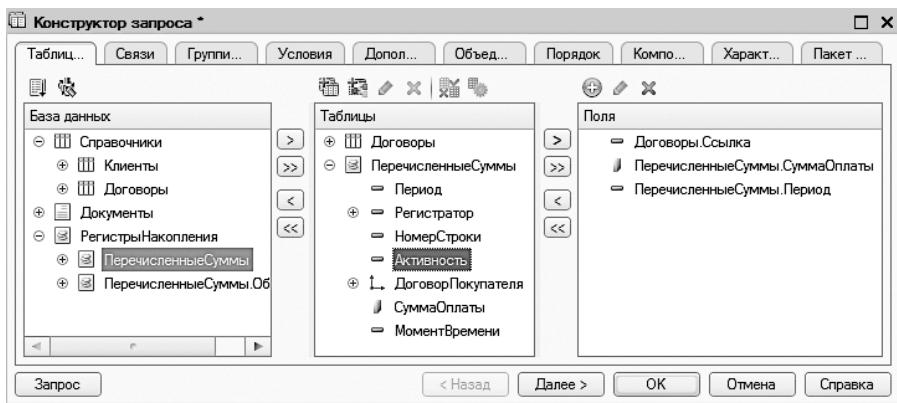


Рис. 5.74. Окно конструктора запроса

На вкладке **Связи** следует убедиться в установке необходимой связи между двумя таблицами (рис. 5.75). Значение измерения **ДоговорПокупателя** регистра накопления должно быть равно ссылке на элемент справочника **Договоры**. И обязательно требуется проверить, что установлен флагок **Все** у таблицы справочника. В этом случае из справочника будут выбраны все элементы, и этим элементам будет поставлено в соответствие значение из регистра.

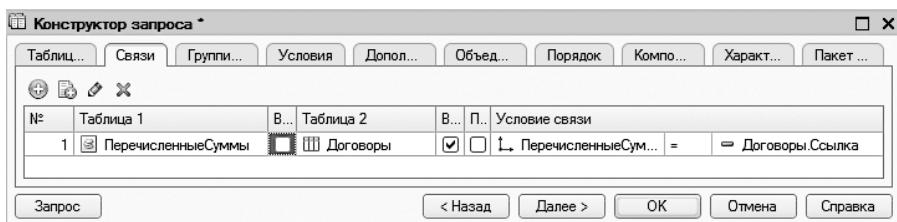


Рис. 5.75. Вкладка Связи окна конструктора запроса

Перейдем далее на вкладку **Объединения/Псевдонимы**, где сформируем более понятные для пользователя псевдонимы полей (рис. 5.76).

На этом действия по конструированию запроса завершаются. После щелчка на кнопке **OK** сформируем текст запроса (листинг 5.24). В данном случае в запросе используется левое соединение таблиц, о котором мы подробно говорили ранее (во второй главе).

Листинг 5.24. Запрос для извлечения информации об оплате по договорам

```
ВЫБРАТЬ
Договоры.Ссылка КАК НазваниеДоговора,
ПеречисленныеСуммы.СуммаОплаты,
```

ПеречисленныеСуммы.Период
 из
 Справочник.Договоры КАК Договоры
 ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ПеречисленныеСуммы КАК
 ПеречисленныеСуммы
 ПО (ПеречисленныеСуммы.ДоговорПокупателя = Договоры.Ссылка)



Рис. 5.76. Вкладка Объединения/Псевдонимы окна конструктора запроса

После получения текста запроса продолжим работу со схемой компоновки данных и на вкладке Ресурсы укажем поле СуммаОплаты. Теперь осталось на вкладке Настройки добавить группировку и выбрать необходимые поля для отчета (рис. 5.77).

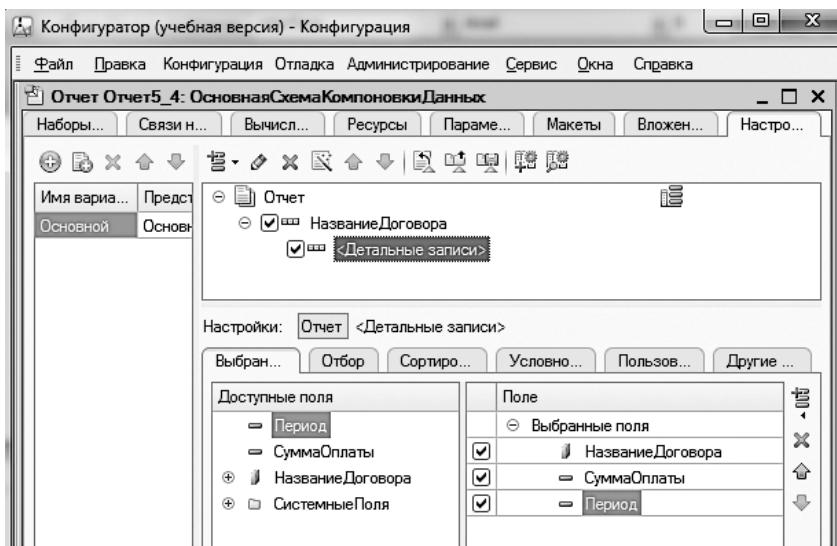


Рис. 5.77. Вкладка Настройки окна схемы компоновки данных

На этом работа в режиме конфигуратора по созданию отчета завершается. Далее можно перейти в режим 1С:Предприятие, где и сформировать отчет по имеющимся в базе данных сведениям (рис. 5.78).

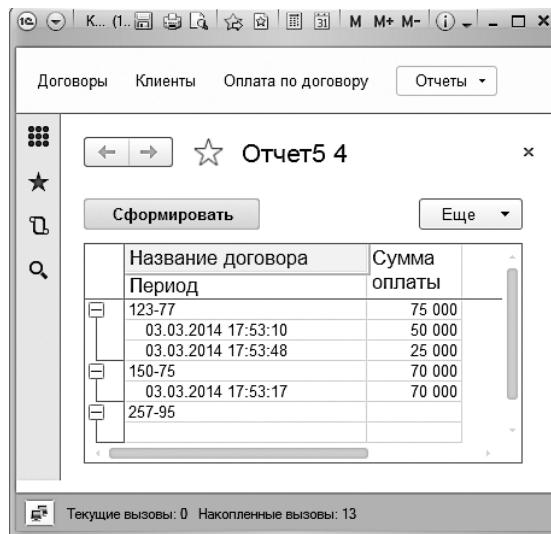


Рис. 5.78. Отчет по договорам с детализацией внесенных оплат

Разработка отчета с итогами оплаты

Создадим теперь отчет, который позволит видеть сводные результаты по оплате и по стоимости договоров клиентов. Назовем отчет *Отчет5_5* и при его создании используем систему компоновки данных. В окне конструктора запросов выберем таблицы в соответствии с рис. 5.79. Далее в таблице *Договоры* необходимо выбрать поля *Клиент*, *Ссылка* и *СуммаПодДоговору*, а в таблице *ПеречисленныеСуммыОбороты* — поле *СуммаОплатыОборот*.

Теперь на вкладке *Связи* следует указать связи между таблицами, а именно — включить *все* записи таблицы справочника (рис. 5.80).

Для большей информативности на вкладке *Объединения/Псевдонимы* укажем псевдоним для поля *Ссылка* (рис. 5.81).

После этого, щелкнув на кнопке *OK*, мы получим текст запроса (листинг 5.25).

Листинг 5.25. Запрос для извлечения информации об оплате

```

ВЫБРАТЬ
    Договоры.Клиент КАК Клиент,
    Договоры.Ссылка КАК ДоговорКлиента,
    Договоры.СуммаПодДоговору,
    ПеречисленныеСуммыОбороты.СуммаОплатыОборот
ИЗ
    Справочник.Договоры КАК Договоры
        ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ПеречисленныеСуммы.Обороты КАК
            ПеречисленныеСуммыОбороты
ПО (ПеречисленныеСуммыОбороты.ДоговорПокупателя = Договоры.Ссылка)

```

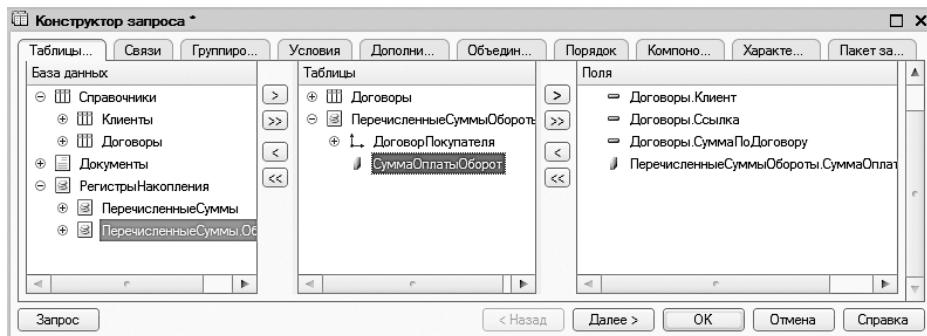


Рис. 5.79. Окно конструктора запросов

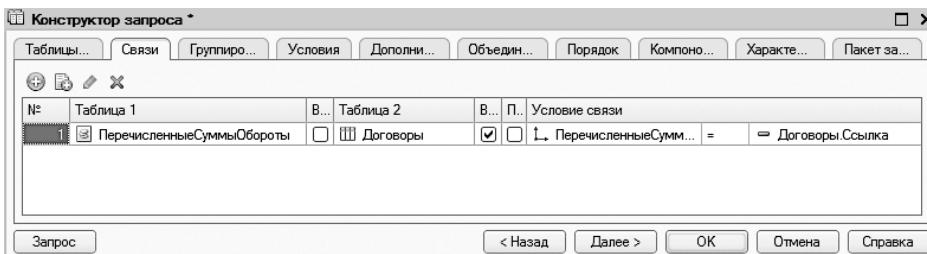


Рис. 5.80. Вкладка Связи окна конструктора запросов

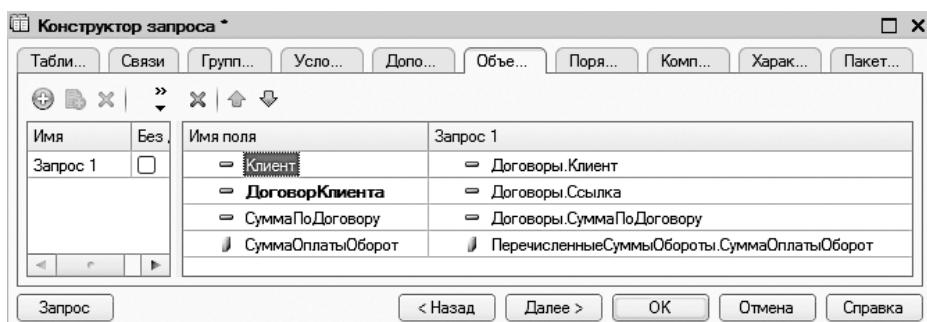


Рис. 5.81. Вкладка Объединения/Псевдонимы окна конструктора запросов

Далее в окне схемы компоновки данных на вкладке Ресурсы с помощью соответствующей кнопки включим ресурс СуммаОплатыОборот.

На вкладке Настройки добавим группировку и включим необходимые поля (рис. 5.82).

На этом работа по созданию отчета в режиме конфигуратора завершается. Далее можно перейти в режим 1С:Предприятие и получить необходимый отчет (рис. 5.83).

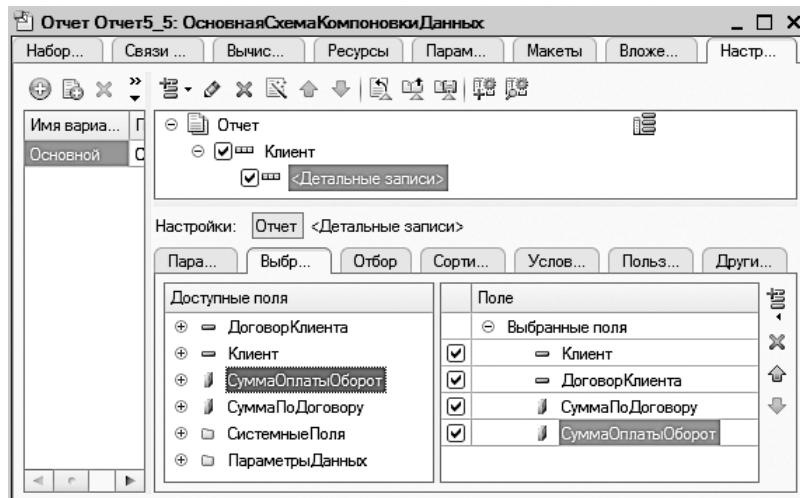


Рис. 5.82. Вкладка Настройки окна схемы компоновки данных

Клиент	Договор	Сумма по договору	Сумма оплаты Оборот
АО "МеталлИнвест"			190 000
150-75		1 500 000	190 000
ЗАО "Пирс"			175 000
123-77		1 700 000	75 000
257-95		790 000	100 000
Итого			365 000

Рис. 5.83. Сводный отчет по оплате

Разработка итогового отчета по оплате

Разрабатываемый в этом разделе отчет назовем Отчет5_6. Как и ранее, воспользуемся схемой компоновки данных и в окне конструктора запроса сформируем выборку полей, представленную на рис. 5.84.

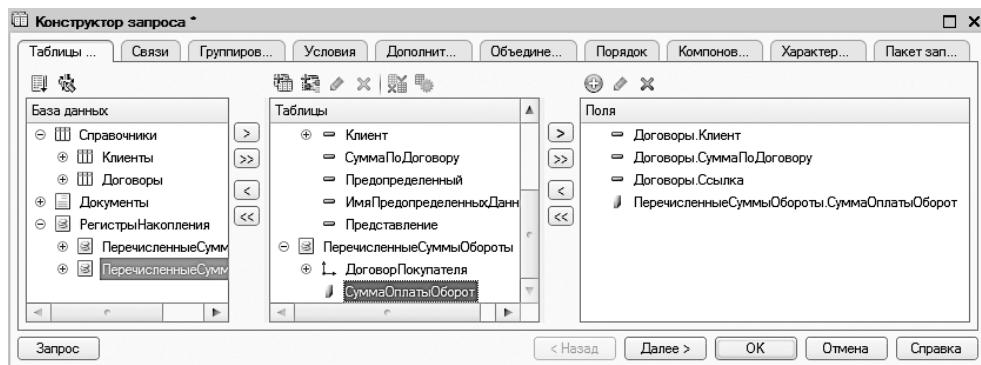


Рис. 5.84. Окно конструктора запроса

В запросе используем полное соединение таблиц при построении запроса (рис. 5.85). В этом случае в результат запроса будут включены комбинации записей из обеих исходных таблиц, которые соответствуют указанному условию. Кроме того, в результат запроса включаются также те записи из обоих источников, для которых не найдено соответствий.

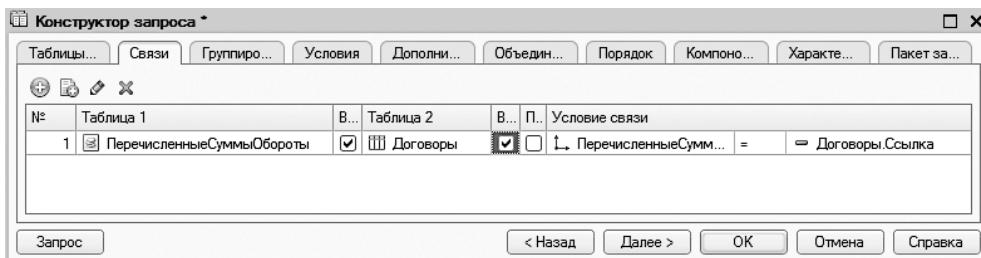


Рис. 5.85. Полное соединение таблиц в запросе

Установим псевдоним для поля **Договоры.Ссылка** (выберем короткий вариант — **Договор**) и укажем упорядочивание по сумме договора в порядке убывания. В листинге 5.26 представлен текст данного запроса, который автоматически формируется конструктором запросов после щелчка на кнопке OK.

Листинг 5.26. Запрос для извлечения полной информации об оплате

```

ВЫБРАТЬ
Договоры.Клиент,
Договоры.СуммаПодОговору КАК СуммаПодОговору,
Договоры.Ссылка КАК Договор,
ПеречисленныеСуммыОбороты.СуммаОплатыОборот
ИЗ
РегистрНакопления.ПеречисленныеСуммы.Обороты КАК ПеречисленныеСуммыОбороты
ПОЛНОЕ СОЕДИНЕНИЕ Справочник.Договоры КАК Договоры
ПО ПеречисленныеСуммыОбороты.ДоговорПокупателя = Договоры.Ссылка

УПОРЯДОЧИТЬ ПО
СуммаПодОговору УБЫВ

```

Осталось сделать несложные установки на вкладке Настройки в окне конструктора схемы компоновки данных (рис. 5.86).

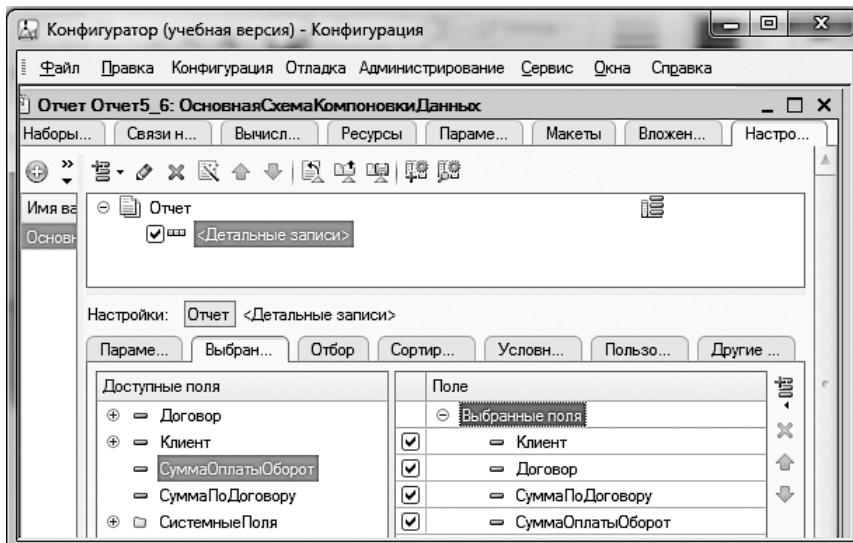


Рис. 5.86. Вкладка Настройки окна конструктора системы компоновки данных

На этом работа по созданию отчета в режиме конфигуратора завершается. Далее можно перейти в режим 1С:Предприятие для получения необходимого отчета (рис. 5.87).

Видно, что в отчет попали как договоры, по которым не было оплат, так и оплаты без указания договоров.

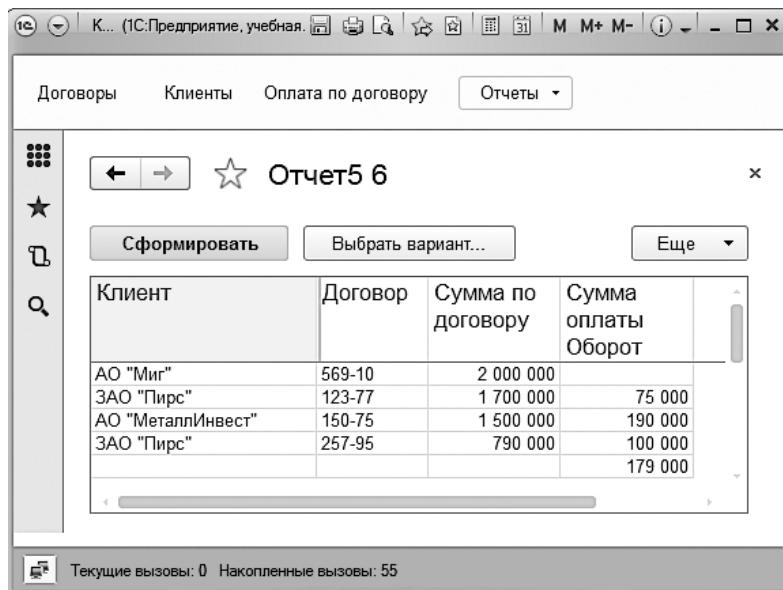


Рис. 5.87. Итоговый отчет по оплате

Ввод документа на основании справочника

Механизм ввода на основании может быть использован для создания объектов разного типа (документы, справочники и т. д.). Наиболее часто этот механизм используется при вводе новых документов. Так, для каждого объекта конфигурации *документ* можно разрешить его ввод на основании других объектов базы данных, а также сделать основанием для других документов.

Рассмотрим ситуацию, когда документ *ЗаявкаНаСклад* будет вводиться на основании справочника *Договоры*. При этом нам потребуется небольшая доработка созданного ранее справочника *Договоры*. А именно, в этот справочник мы добавим табличную часть *Перечень* (рис. 5.88). Смысл изменений заключается во внесении в табличную часть элементов (и их количества), которые мы должны поставить в соответствие с данным договором. В качестве типа данных реквизита *Название* укажем строку длиной 50 символов, а для данных поля *Количество* выберем численный тип.

После этого необходимо разработать документ *ЗаявкаНаСклад*. В окне редактирования этого объекта конфигурации добавим реквизит *Договор* (тип данных — *Справочник-Ссылка.Договоры*), а затем создадим табличную часть документа, которую назовем *Перечень*. В табличной части документа разместим два реквизита (рис. 5.89):

- Название (тип данных — строка);
- Количество (тип данных — число).

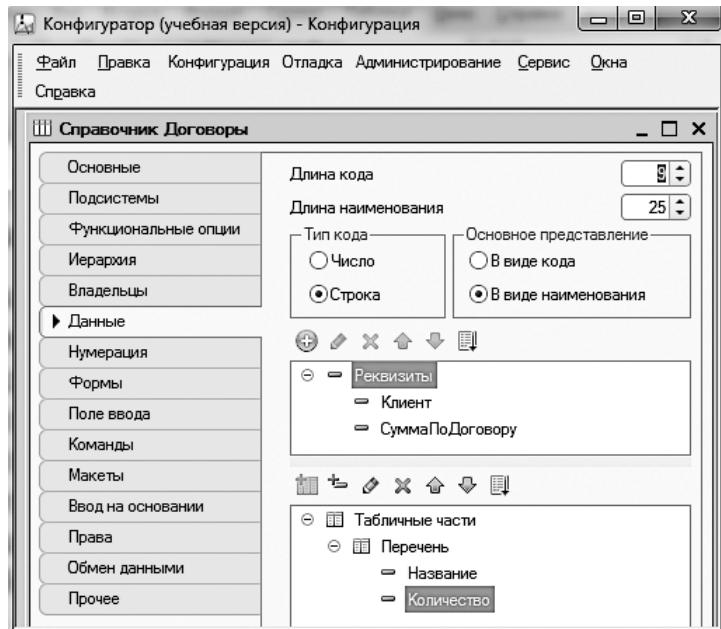


Рис. 5.88. Коррекция справочника Договоры

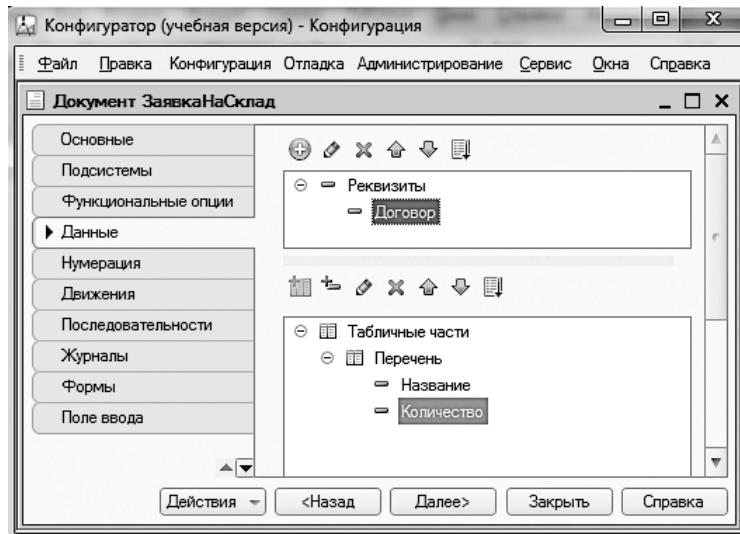


Рис. 5.89. Структура документа ЗаявкаНаСклад

Теперь в окне редактирования объекта конфигурации ЗаявкаНаСклад перейдем на вкладку Ввод на основании, где можно указать объекты, на основании которых может вводиться рассматриваемый документ. Определим в качестве основания справочник Договоры (рис. 5.90). После этого с помощью соответствующей кнопки вызовем конструктор ввода на основании. В открывшемся окне (рис. 5.91) сформируем значения реквизитов документа, создаваемого на основании справочника Договоры. Для этого следует воспользоваться кнопкой Заполнить выражения.

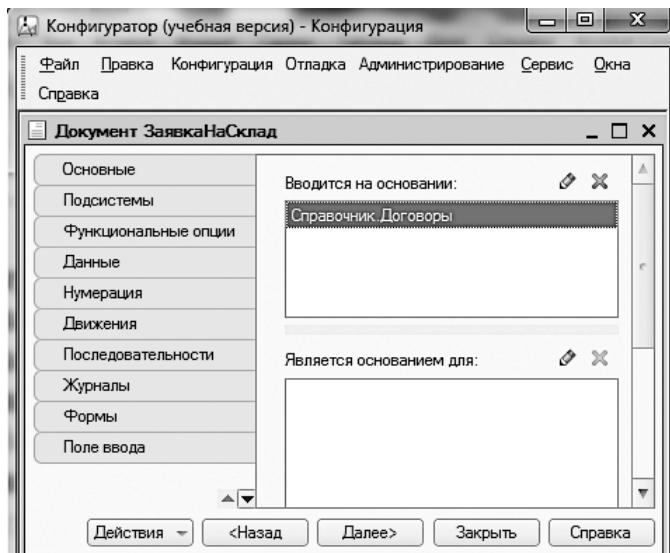


Рис. 5.90. Вкладка Ввод на основании

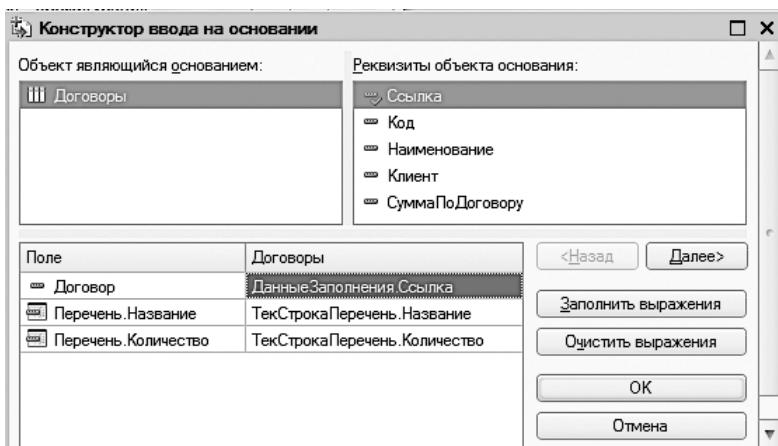


Рис. 5.91. Окно конструктора ввода на основании

В результате этих действий автоматически сформируется программная процедура, позволяющая создать новый документ на основании указанного пользователем элемента справочника (листинг 5.27).

Листинг 5.27. Процедура ввода на основании

```
Процедура ОбработкаЗаполнения(ДанныеЗаполнения, СтандартнаяОбработка)
//{{__КОНСТРУКТОР_ВВОД_НА_ОСНОВАНИИ
// Данный фрагмент построен конструктором.
Если ТипЭнч(ДанныеЗаполнения) = Тип("СправочникСсылка.Договоры") Тогда
    // Заполнение шапки
    Договор = ДанныеЗаполнения.Ссылка;
    Для Каждого ТекСтрокаПеречень Из ДанныеЗаполнения.Перечень Цикл
        НоваяСтрока = Перечень.Добавить();
        НоваяСтрока.Количество = ТекСтрокаПеречень.Количество;
        НоваяСтрока.Название = ТекСтрокаПеречень.Название;
    КонецЦикла;
КонецЕсли;
//}}__КОНСТРУКТОР_ВВОД_НА_ОСНОВАНИИ
КонецПроцедуры
```

Теперь следует проверить разработку. Для этого перейдем в режим 1С:Предприятие и для начала сформируем табличную часть справочника Договоры (рис. 5.92).

	Название	Количество
1	Блок 23-7	5
2	Модуль 12-1	10
3	Модуль 17-7	15

Рис. 5.92. Элемент справочника Договоры

В форме списка справочника Договоры присутствует кнопка Создать на основании. Если ею воспользоваться, то в вертикальном меню откроется список документов, которые можно создать на основании справочника Договоры. Используя эту возможность, сформируем новый документ ЗаявкаНаСклад, который можно при необходимости скорректировать (рис. 5.93).

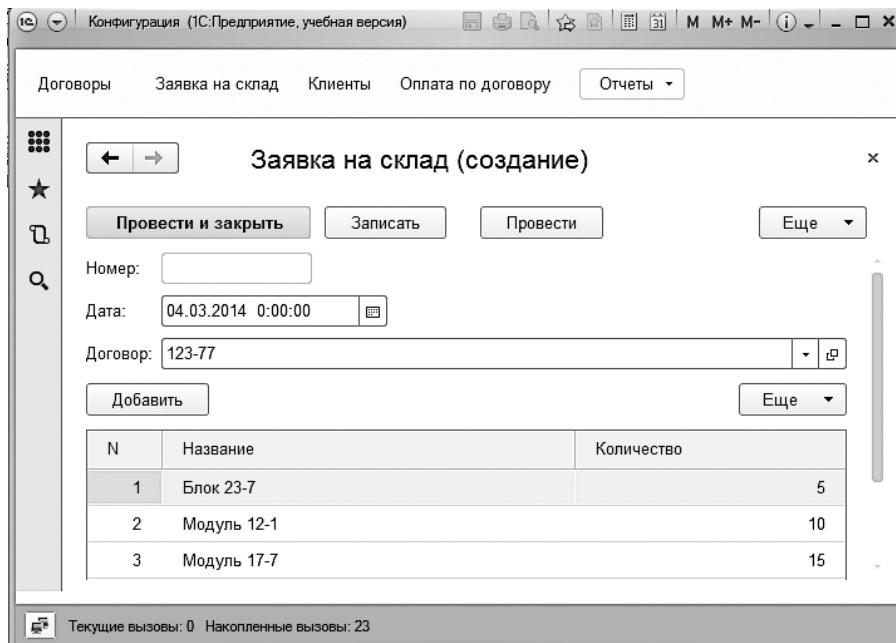


Рис. 5.93. Заполненная форма документа ЗаявкаНаСклад на основании справочника

Передача данных в Microsoft Excel

В практической работе интеграция различных приложений требуется очень часто. Это может быть связано с рядом причин. Так, разные сотрудники крупных и не очень крупных организаций, как правило, работают с разным набором программных продуктов. Это и зависит от уровня занимаемых ими должностей, и может быть связано с различными профессиональными обязанностями, а иногда и со знанием только конкретных программ. Но даже при хорошей информатизации предприятия все равно часто требуется передавать данные, созданные в одних приложениях, в другие программы.

Наверное, все согласятся, что для большинства фирм характерно использование в офисной работе приложения Microsoft Excel (электронные таблицы). В то же время статистика говорит о том, что и программный продукт 1С:Предприятие также в той или иной степени используется практически в каждой организации. В этом случае сотрудникам часто приходится каким-то образом организовывать передачу текущей или отчетной документации из одних программ в другие. Организация подобного обмена может возлагаться на системных программистов либо на обычных пользователей, но в обоих случаях она требует заметных усилий.

В данном разделе рассматривается организация электронного обмена данными между приложением 1С:Предприятие 8 и Microsoft Excel. Приводимый далее прием можно использовать и при решении других офисных задач, требующих подобной отчетной информации.

Приложения, входящие в пакет Microsoft Office, и система 1С:Предприятие 8 открыты для взаимного обмена информацией.

Разработчиками в систему 1С:Предприятие 8 включен набор технологий, ориентированных на решение разнообразных задач интеграции. Одной из них является технология *COM* (Component Object Model — компонентная объектная модель), поддерживающая организацию программ из отдельных компонентов, которые, в свою очередь, состоят из объектов. Данные компоненты представляют собой исполняемый двоичный код, и для использования того или иного компонента достаточно просто зарегистрировать его в операционной системе.

ПРИМЕЧАНИЕ

Фактически при использовании технологии COM осуществлять компоновку компонентов с основным модулем не требуется.

Другая не менее известная технология — *Automation Client/Server*. Еще одно ее название — *OLE Automation*, где OLE расшифровывается как *Object Linking and Embedding* (связывание и внедрение объектов). В этом случае пользователям предоставляется возможность работы с составными объектами, созданными в других приложениях (документы, таблицы, рисунки и др.). Основные понятия, с которыми оперирует данная технология, — это OLE-объект, сервер приложения и контейнер приложения.

OLE-объектом называют объект, созданный в другом приложении, но при этом сохранивший связь с текущим приложением. Для этого он должен быть вставлен в интересующий нас документ соответствующим способом.

OLE-контейнером является приложение, в котором создается составной документ, содержащий OLE-объект. В этом случае при необходимости работы с OLE-объектом открывается исходное приложение (то, в котором рассматриваемый объект был создан). Это исходное приложение называется OLE-сервером. Офисные приложения Microsoft Word, Microsoft Excel и система 1С:Предприятие 8 могут выступать и как OLE-сервер, и как OLE-контейнер.

В целом работа системы 1С:Предприятие через СОМ-соединение похожа на работу в режиме Automation Client/Server. Существенное отличие заключается в том, что при СОМ-соединении запускается вариант информационной системы

1С:Предприятие без интерфейсной части. Работа в этом случае производится только программными средствами. Для пользователя интерактивные средства отсутствуют. Основное назначение сервера автоматизации системы 1С:Предприятие – это управление системой 1С:Предприятие 8 из других приложений и выполнение действий, аналогичных интерактивной работе.

В целом основной целью использования СОМ-соединения для системы 1С:Предприятие 8 является обеспечение надежного программного доступа к объектам системы из внешних приложений.

Использование СОМ-соединения в разработках обеспечивает:

- снижение затрат системных ресурсов;
- быструю установку соединения с базой данных;
- более быстрое (по сравнению с технологией OLE-автоматизации) обращение к свойствам и методам объектов.

Технология СОМ не обеспечивает использования визуальных ресурсов приложения сервера, что отличает ее от технологии OLE-автоматизации, которая такую возможность предоставляет.

Здесь мы в качестве примера рассмотрим передачу информации из справочника Клиенты в рабочую книгу Microsoft Excel. При этом воспользуемся технологией OLE-автоматизации. В нашей разработке справочник Клиенты на данный момент является малоинформационным (только обязательные поля, код и наименование), поэтому добавим в него еще два реквизита (рис. 5.94):

- Адрес (тип данных – строка длиной не более 100 символов);
- Телефоны (тип данных – строка длиной не более 50 символов).

Первое практическое действие заключается в том, что мы должны открыть приложение Microsoft Excel и создать на диске C: в папке 1 новую книгу, в качестве названия которой укажем Клиенты.xls. Содержание книги представляет собой просто первый рабочий лист с определенным оформлением (рис. 5.95). В дальнейшем на данный лист будет автоматически переноситься информация из нашего справочника Клиенты.

Следующий практический шаг заключается в размещении на форме списка справочника Клиенты кнопки. По щелчку на ней будет выполняться процедура, реализующая передачу данных из данного справочника на упомянутый лист книги Клиенты.xls. Это действие разбивается на несколько этапов, которые мы далее и рассмотрим.

Для программирования нам потребуется пользовательская форма списка справочника Клиенты, создаваемая с помощью конструктора форм (рис. 5.96).

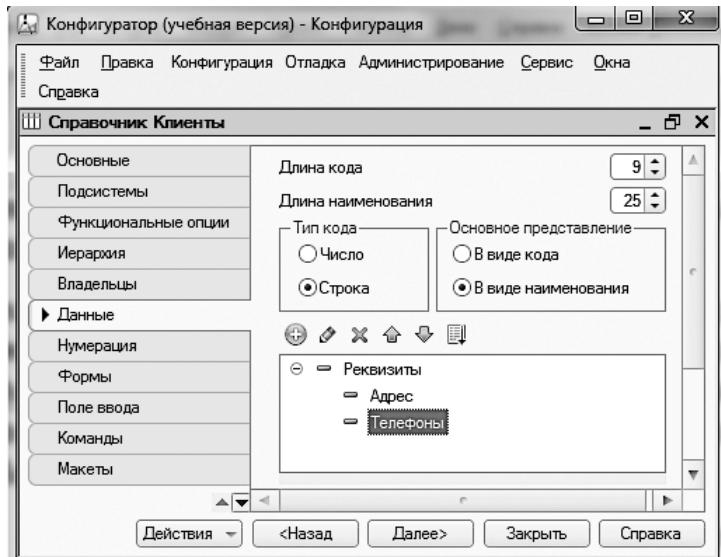


Рис. 5.94. Внесение реквизитов в справочник Клиенты

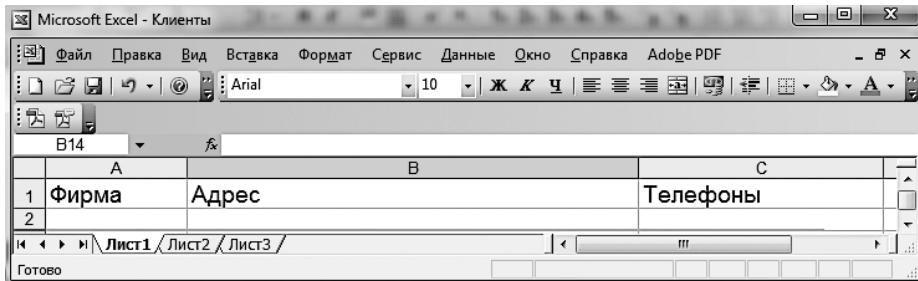


Рис. 5.95. Книга Microsoft Excel Клиенты.xls

Следующий шаг заключается в добавлении новой команды в форму, что реализуется на дополнительной вкладке **Команды** (ярлычок этой вкладки виден в правой части рис. 5.96). Используя пункт **Добавить** пиктографического меню, разместим новую команду **ПередачаВExcel** (рис. 5.97).

Далее в окне свойств (см. рис. 5.97) новой команды воспользуемся разделом **Действие** и добавим программный код для команды **ПередачаВExcel** (листинг 5.28).

Листинг 5.28. Код команды ПередачаВExcel

```
&НаКлиенте
Процедура ПередачаВExcel(Команда)
    Передача();
КонецПроцедуры
```

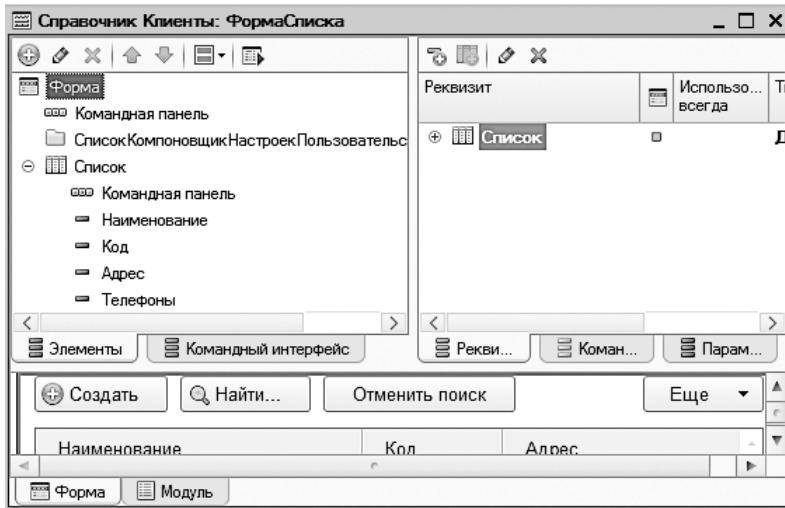


Рис. 5.96. Форма списка справочника Клиенты в окне конструктора форм

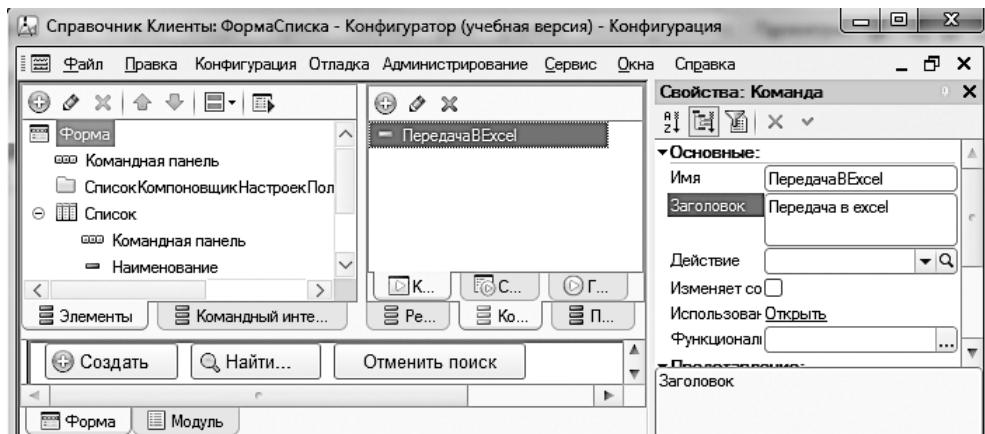


Рис. 5.97. Добавление команды в форму списка

Как видно, программный код команды ПередачаВExcel обеспечивает только вызов процедуры Передача, которая выполняется на сервере (листинг 5.29). Ее также следует расположить в закладке Модуль рядом с процедурой ПередачаВExcel.

Листинг 5.29. Код серверной процедуры Передача

```
&НаСервереБезКонтекста
Процедура Передача()
ApplicationExcel= Новый СОМОбъект("Excel.Application");
ApplicationExcel.Visible = Истина;
ApplicationExcel.Workbooks.Open ("C:\1\Клиенты.xls");
```

продолжение ↗

Листинг 5.29 (продолжение)

```

Спр=Справочники.Клиенты;
i=2;
Выборка=Спр.Выбрать();
Пока Выборка.Следующий() > 0 Цикл
    ApplicationExcel.Sheets(1).Cells(i, 1).Value=Выборка.Наименование;
    ApplicationExcel.Sheets(1).Cells(i, 2).Value=Выборка.Адрес;
    ApplicationExcel.Sheets(1).Cells(i, 3).Value=Выборка.Телефоны;
    i=i+1;
КонецЦикла;
КонецПроцедуры

```

Характерной особенностью выполняемого обмена данными является открытие окна рабочей книги Microsoft Excel в процессе передачи данных.

Программные процедуры теперь готовы, осталось разместить на форме списка справочника Клиенты кнопку, запускающую разработанную команду ПередачаВExcel. Для этого достаточно в окне редактора форм перетащить созданную команду ПередачаВExcel на элемент Командная панель (см. рис. 5.97).

Теперь, когда разработка готова, перейдем в режим 1С:Предприятие и откроем форму списка справочника Клиенты (рис. 5.98). Как видно, на командной панели присутствует новая кнопка Передача в excel. После щелчка на ней происходит запуск приложения Microsoft Excel и передача информации из справочника Клиенты в рабочую книгу Клиенты.xls (рис. 5.99).

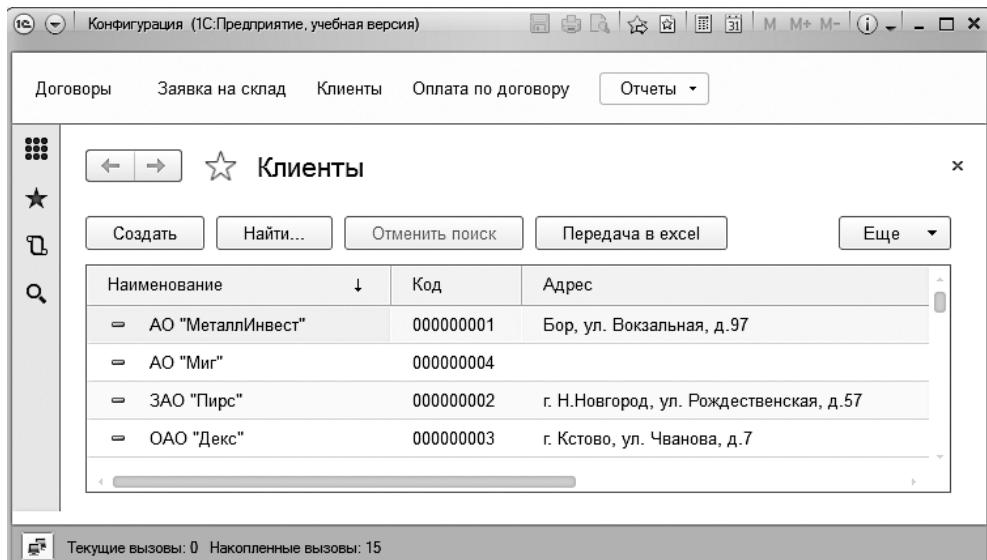


Рис. 5.98. Форма списка справочника Клиенты с дополнительной кнопкой Передача в excel

The screenshot shows a Microsoft Excel window titled "Microsoft Excel - Клиенты". The spreadsheet contains three columns: "Фирма" (Company), "Адрес" (Address), and "Телефоны" (Phones). The data is as follows:

	Фирма	Адрес	Телефоны
1	АО "МеталлИнвест"	Бор, ул. Вокзальная, д.97	674-78-90
2	АО "Миг"		
3	ЗАО "Пирс"	г. Н.Новгород, ул. Рождественская, д.57	123-67-90
4	ОАО "Декс"	г. Кстово, ул. Чванова, д.7	78-67-12
5			
6			

Рис. 5.99. Окно книги Microsoft Excel с полученной информацией из справочника Клиенты

Анализ работы менеджеров

Рассмотрим еще одну задачу учета, касающуюся торговой деятельности. При этом основное содержание прикладной разработки будет связано с анализом работы менеджеров нашей условной организации. Подчеркнем практическую значимость содержания данной главы следующим примером. Будем считать, что менеджеры условной фирмы имеют разное образование (техническое, экономическое, гуманистичное), которое влияет на то, какие товары они предлагают клиентам. В результате детального анализа может оказаться, что наиболее высокотехнологичные модели продаются в основном менеджерами с техническим образованием. Подобный результат, несомненно, важен для руководства компании. Он позволит вовремя внести необходимые корректизы в работу организации, например, переобучить часть персонала.

Создание перечисления

Для выполнения практических действий создадим новую конфигурацию, основным назначением которой будет разработка ряда аналитических отчетов. Начнем построение информационной базы с создания перечисления. Этот объект конфигурации нам уже встречался в четвертой главе книги. Перечисления представляют собой статические списки данных, создаваемые на этапе разработки. В дальнейшем в режиме 1С:Предприятие эти списки просто используются в справочниках, документах и других объектах (без возможности изменения или дополнения).

Итак, первым объектом, который мы создадим в новой конфигурации, будет перечисление **Образование**, отражающее разное образование менеджеров нашей фирмы. Напомним, что для создания нового перечисления следует щелкнуть правой кнопкой мыши на строке **Перечисления** в окне Конфигурация. Далее в открывшемся

контекстном меню необходимо воспользоваться пунктом Добавить, после чего на экране откроется окно редактирования объекта конфигурации. Здесь на вкладке Основные следует только задать имя перечисления.

В практической работе варианты образования не подразумевают какого-то большого многообразия, и в нашей разработке в качестве значений перечисления Образование учащегося мы укажем следующие:

- Экономическое;
- Техническое;
- Гуманитарное.

Рассматриваемые параметры следует задать на вкладке Данные (рис. 5.100). Эти значения перечисления Образование мы будем использовать в создаваемых далее справочниках.

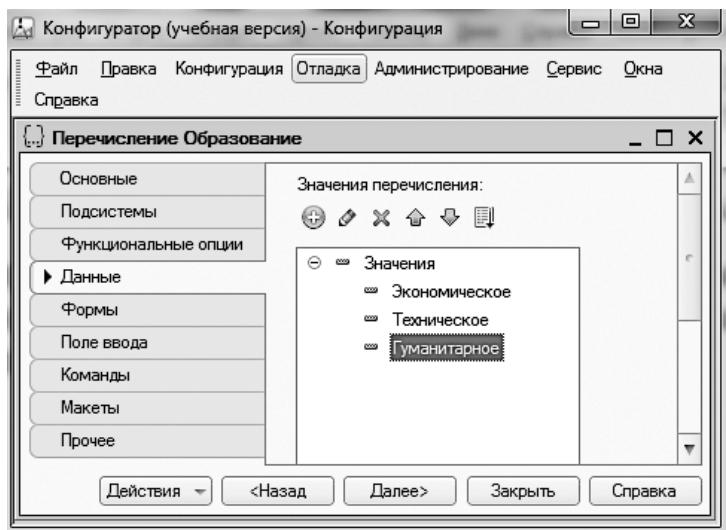


Рис. 5.100. Вкладка Данные окна редактирования перечисления Образование

Справочники Менеджеры и УчасткиРаботы

Создадим еще один объект нашей конфигурации — справочник Менеджеры. Кроме обязательных реквизитов (кода и наименования) добавим в него следующие реквизиты:

- Образование (тип данных — ПеречислениеСсылка.Образование);
- Стаж (тип данных — число).

Эти параметры необходимо задать на вкладке **Данные** в окне редактирования объекта конфигурации (рис. 5.101).

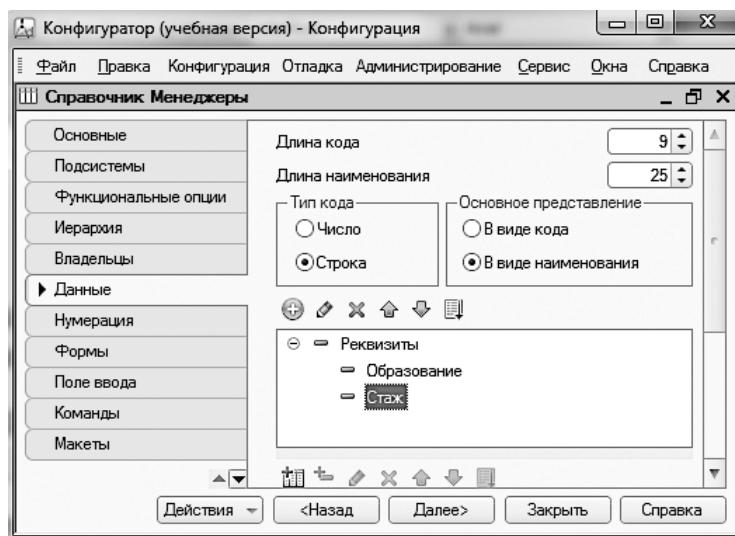


Рис. 5.101. Вкладка Данные окна редактирования справочника Менеджеры

После выполнения описанных действий в режиме конфигуратора перейдем в режим 1С:Предприятие и внесем в справочник Менеджеры несколько фамилий менеджеров (рис. 5.102).

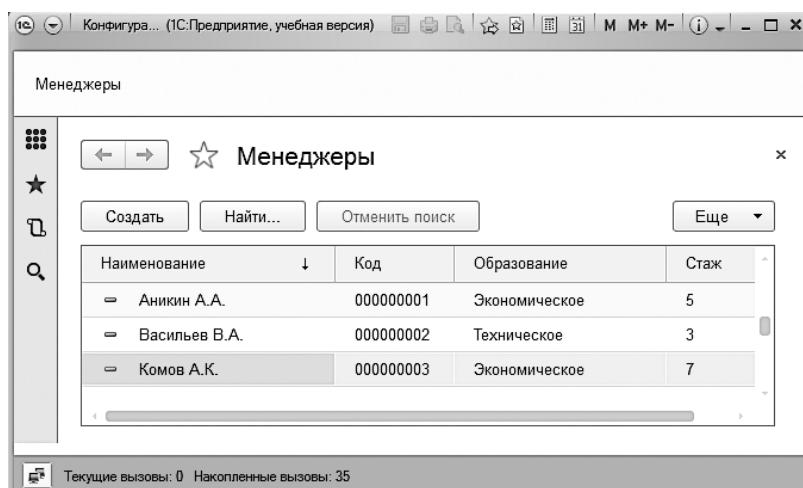


Рис. 5.102. Форма списка справочника Менеджеры в режиме 1С:Предприятие

Еще один справочник, который нам понадобится, — УчасткиРаботы. Данный справочник не будет иметь дополнительных реквизитов (кода и наименования здесь вполне достаточно). После создания данного объекта конфигурации перейдем в режим 1С:Предприятие и внесем в справочник УчасткиРаботы несколько записей, которые характеризуют основные направления деятельности нашей организации по торговой деятельности (рис. 5.103).

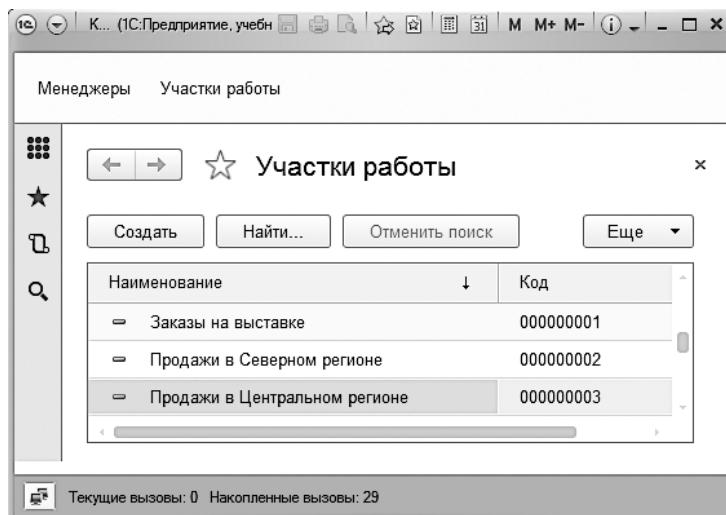


Рис. 5.103. Заполнение справочника УчасткиРаботы в режиме 1С:Предприятие

В следующем разделе мы создадим два объекта конфигурации, которые будут фиксировать факты продажи товаров и позволят провести анализ деятельности менеджеров.

Регистр накопления РаботаМенеджеров и документ ПродажиМенеджеров

Для накопления информации о продажах товаров в качестве объекта конфигурации создадим регистр накопления с именем РаботаМенеджеров. На вкладке Основные окна редактирования объекта конфигурации следует указать имя объекта и вид регистра накопления — Обороты.

После этого на вкладке Данные необходимо создать измерения регистра (рис. 5.104):

- УчастокРаботы (тип данных — СправочникСсылка.УчасткиРаботы);
- Менеджер (тип данных — СправочникСсылка.Менеджеры).

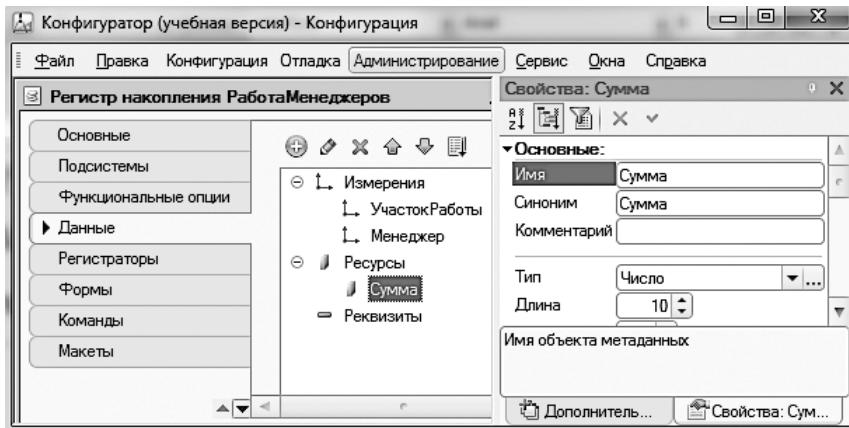


Рис. 5.104. Вкладка Данные окна редактирования регистра РаботаМенеджеров

В качестве ресурса регистра накопления укажем ресурс **Сумма** (тип данных – число). Таким образом, мы подготовили возможность проведения документа, информирующего о продажах, — создали регистр накопления для фиксирования продаж.

Перейдем теперь к разработке документа **ПродажиМенеджеров**, который будет отражать совершенные продажи конкретными менеджерами. На вкладке **Данные** разместим реквизит **Менеджер** (тип данных – **СправочникСсылка.Менеджеры**) для указания в документе фамилии менеджера, выполнившего конкретную продажу, и создадим табличную часть **Перечень**, в которую поместим два реквизита:

- УчастокРаботы (тип данных – **СправочникСсылка.УчасткиРаботы**);
- Сумма (тип данных – число).

На рис. 5.105 представлена вкладка **Данные** окна редактирования документа **ПродажиМенеджеров**.

Осталось обеспечить проведение документа – формирование движений по регистру **РаботаМенеджеров**. Для этого в окне редактирования объекта конфигурации **ПродажиМенеджеров** перейдем на вкладку **Движения** (рис. 5.106). Здесь необходимо в списке регистров конфигурации отметить единственный регистр накопления, которым мы располагаем.

После установки флажка для регистра становится доступной кнопка **Конструктор движений**, которой мы и воспользуемся. В результате на экране откроется окно конструктора движений регистра (рис. 5.107). Здесь с помощью кнопки **Заполнить выражения** формируется соответствие названий полей регистра и реквизитов документа.

Теперь в окне конструктора движений осталось щелкнуть на кнопке **OK**, что приводит к генерации программной процедуры, которая формирует движения

по регистру РаботаМенеджеров при проведении документа ПродажиМенеджеров (листинг 5.30).

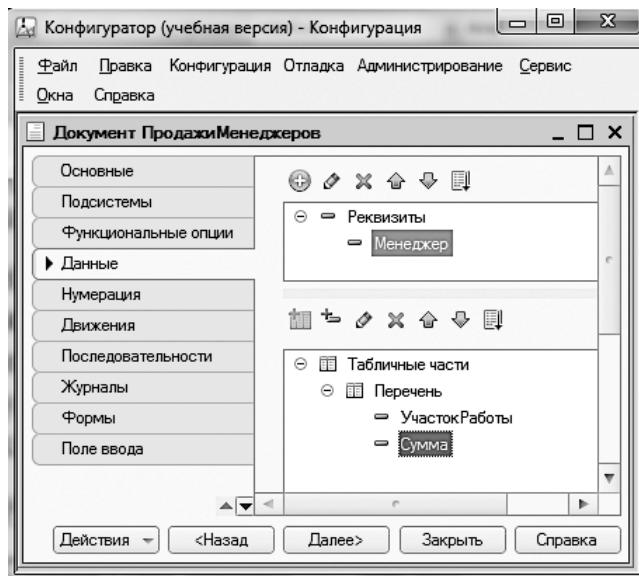


Рис. 5.105. Вкладка Данные окна редактирования документа ПродажиМенеджеров

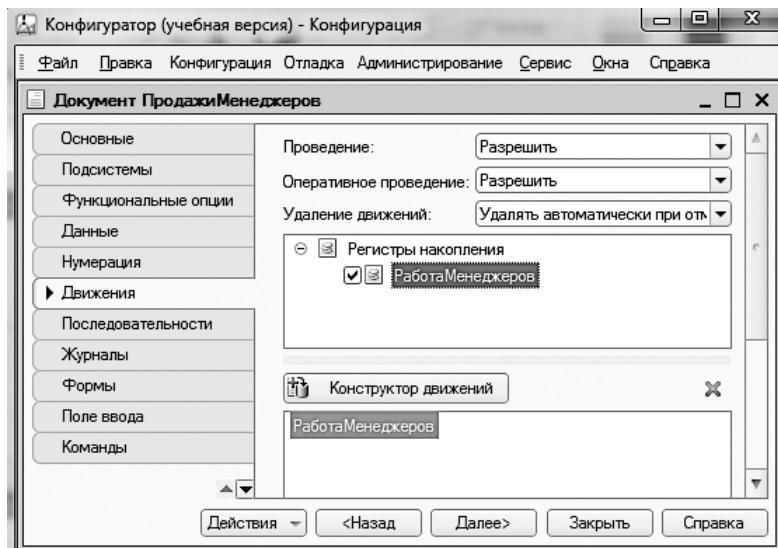


Рис. 5.106. Вкладка Движения окна редактирования документа ПродажиМенеджеров

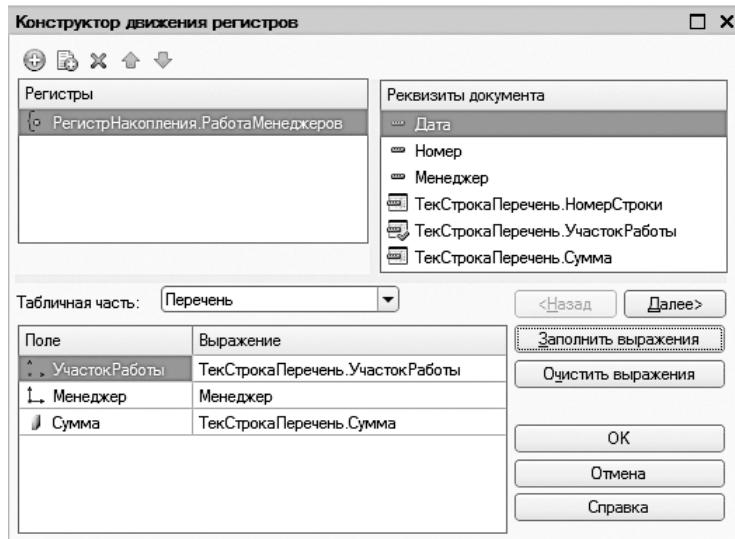


Рис. 5.107. Окно конструктора движений регистра

Листинг 5.30. Процедура проведения документа ПродажиМенеджеров

```

Процедура ОбработкаПроведения(Отказ, Режим)
//{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
Движения.РаботаМенеджеров.Записывать = Истина;
Для Каждого ТекСтроКаПеречень Из Перечень Цикл
    // регистр РаботаМенеджеров
    Движение = Движения.РаботаМенеджеров.Добавить();
    Движение.Период = Дата;
    Движение.УчастокРаботы = ТекСтроКаПеречень.УчастокРаботы;
    Движение.Менеджер = Менеджер;
    Движение.Сумма = ТекСтроКаПеречень.Сумма;
КонецЦикла;
//}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры

```

Основной программный код процедуры **ОбработкаПроведения** составляет цикл, в котором перебираются строки табличной части **Перечень** рассматриваемого документа:

Для Каждого ТекСтроКаПеречень Из Перечень Цикл

При каждом проходе цикла с помощью метода **Добавить()** создается новая запись в регистре накопления. В последующих строках данной процедуры устанавливаются значения полей регистра в соответствии с данными из очередной строки табличной части документа.

Теперь следует перейти в режим 1С:Предприятие и проверить результат выполненных действий. На рис. 5.108 представлен один из заполненных документов ПродажиМенеджеров.

N	Участок работы	Сумма
1	Заказы на выставке	120 000
2	Продажи в Северном регионе	500 000
3	Продажи в Центральном регионе	700 000

Рис. 5.108. Форма документа ПродажиМенеджеров в режиме 1С:Предприятие

После внесения данных щелчок на кнопке **Провести и закрыть** в форме документа приводит к записи созданного документа в информационную базу. Кроме того, осуществляется и его проведение (формируются движения по регистру накопления).

Отчет с группировкой по заказам

Итак, мы завершили первую фазу разработки конфигурации — разработали необходимые объекты для фиксирования фактов продаж товаров. Дальнейшая цель заключается в разработке ряда отчетов, связанных с деятельностью менеджеров. Первый из них представлен на рис. 5.109, где мы видим группировку по каждому виду деятельности с отражением участия менеджеров в денежном выражении.

Разработку данного отчета начнем с создания нового объекта конфигурации — отчета **Отчет5_7**. И как уже мы неоднократно делали в предыдущих главах, воспользуемся системой компоновки данных, а для построения запроса обратимся к конструктору запросов (рис. 5.110).

Для отчета нам понадобится виртуальная таблица регистра накопления **Работа-Менеджеров.Обороты**. В окне конструктора запросов больше никаких действий

выполнять не требуется, и щелчок на кнопке **OK** приведет к формированию текста запроса (листиング 5.31).

Листинг 5.31. Текст запроса для отчета Отчет5_7

ВЫБРАТЬ

РаботаМенеджеровОбороты.УчастокРаботы,

РаботаМенеджеровОбороты.Менеджер,

РаботаМенеджеровОбороты.СуммаОборот

ИЗ

РегистрНакопления.РаботаМенеджеров.Обороты КАК РаботаМенеджеровОбороты

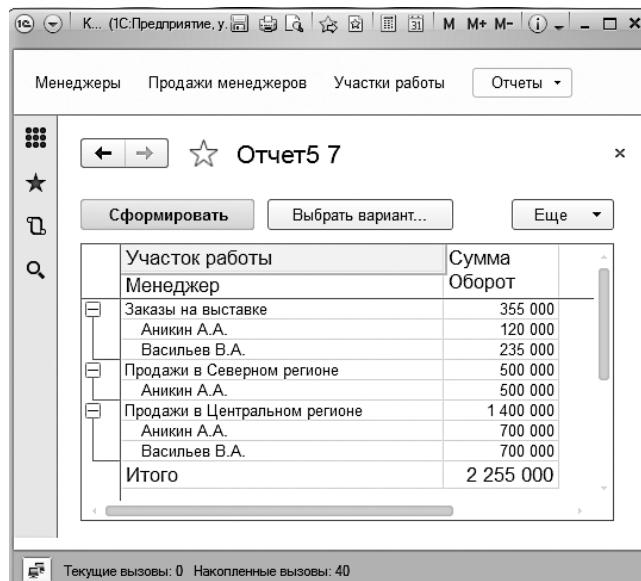


Рис. 5.109. Отчет в режиме 1С:Предприятие

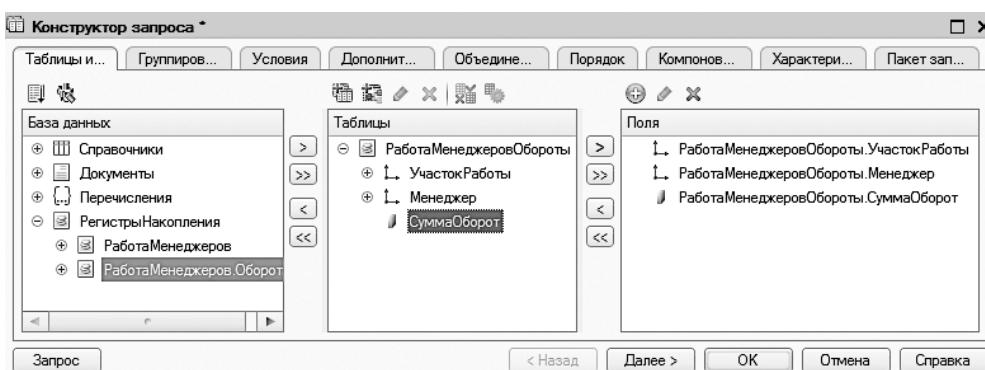


Рис. 5.110. Окно конструктора запроса

Теперь необходимо выполнить ряд установок в окне схемы компоновки данных. Так, на вкладке Ресурсы добавим ресурс СуммаОборот (рис. 5.111).

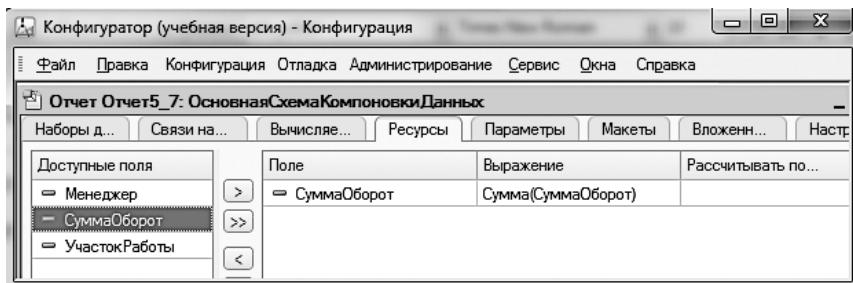


Рис. 5.111. Вкладка Ресурсы окна конструктора схемы компоновки данных

Осталось задать группировку и выбрать поля для отчета на вкладке Настройки (рис. 5.112), после чего можно перейти в режим 1С:Предприятие и получить отчет в том виде, в каком мы его уже видели (см. рис. 5.109).

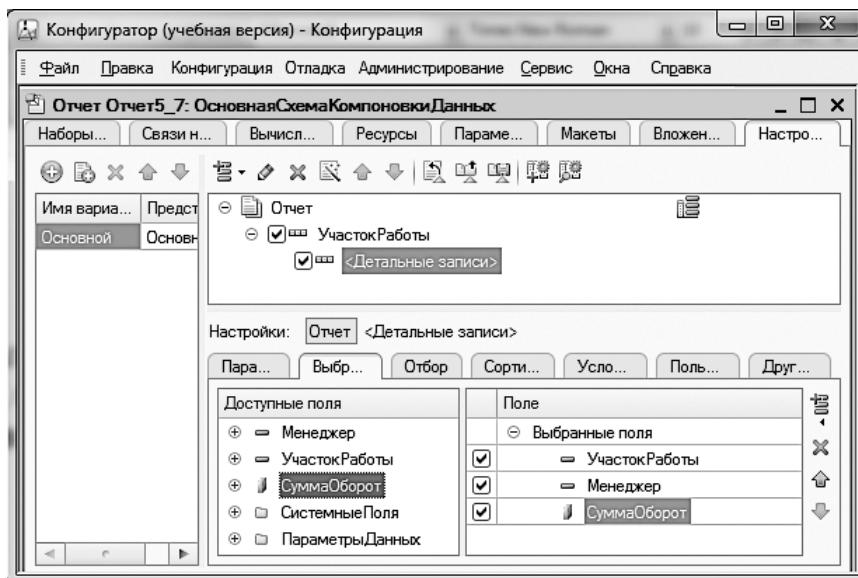


Рис. 5.112. Вкладка Настройки окна конструктора схемы компоновки данных

Отчет по интервалу дат

Теперь немного изменим предыдущий отчет, чтобы предусмотреть возможность указания интервала дат при выборе сведений из виртуальной таблицы регистра. В обновленном виде отчет должен выглядеть так, как показано на рис. 5.113.

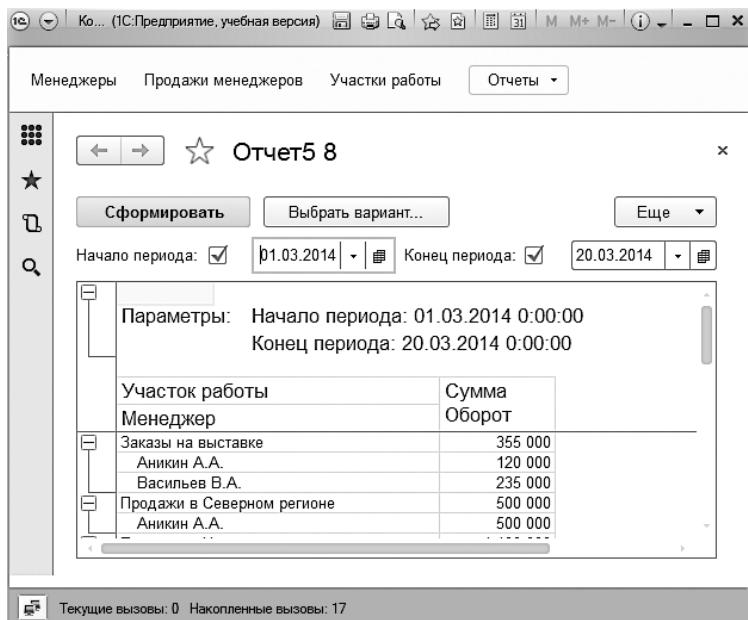


Рис. 5.113. Отчет по работе менеджеров с отбором информации по дате

Для выполнения данной разработки создадим новый объект конфигурации с именем Отчет5_8. После этого воспользуемся системой компоновки данных и конструктором запроса. Здесь, как и в предыдущем отчете, мы задействуем виртуальную таблицу РаботаМенеджеров.Обороты (рис. 5.114).

На вкладке Ресурсы укажем необходимый ресурс для отчета — СуммаОборт (рис. 5.115).

Далее на вкладке Настройки обратимся к дополнительной вкладке Параметры, где следует выполнить пользовательские настройки элементов:

- Начало периода;
- Конец периода.

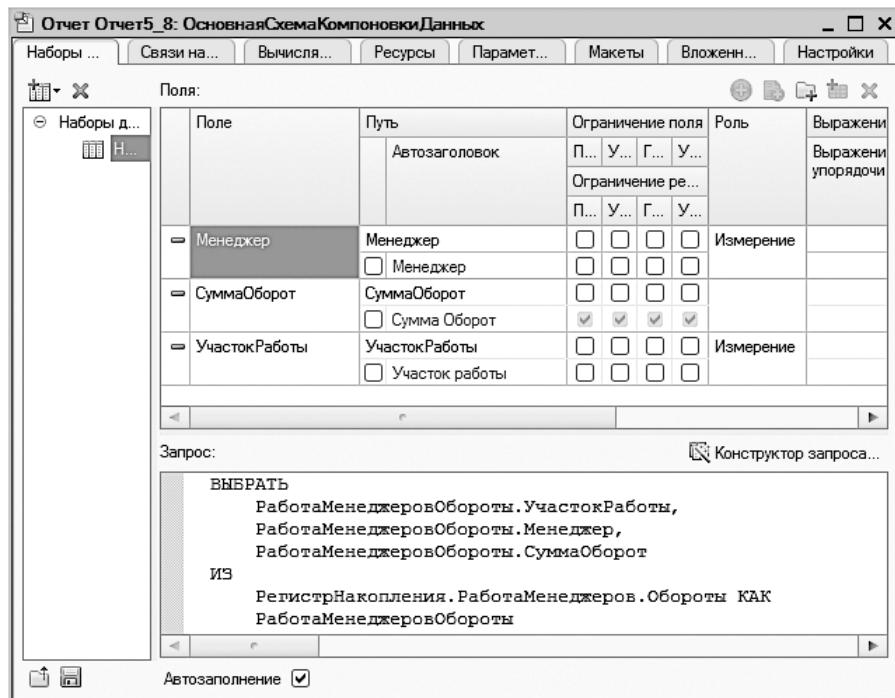


Рис. 5.114. Вкладка НаборыДанных окна конструктора схемы компоновки данных

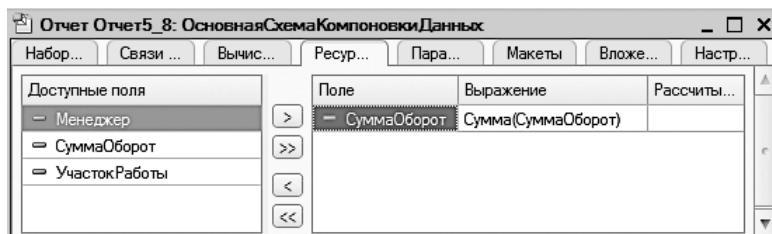


Рис. 5.115. Вкладка Ресурсы схемы компоновки данных

А именно, требуется установить флажок **Включать в пользовательские настройки** (рис. 5.116). После этого остается на той же вкладке **Настройки** создать группировки для отчета и выбрать необходимые поля (рис. 5.117).

На этом разработка завершается, а сам отчет мы уже видели (см. рис. 5.113).

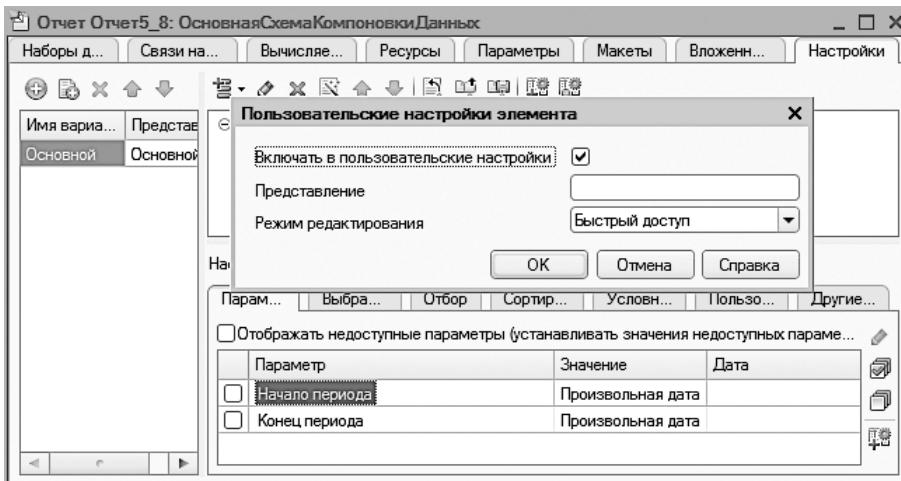


Рис. 5.116. Окно для быстрой настройки

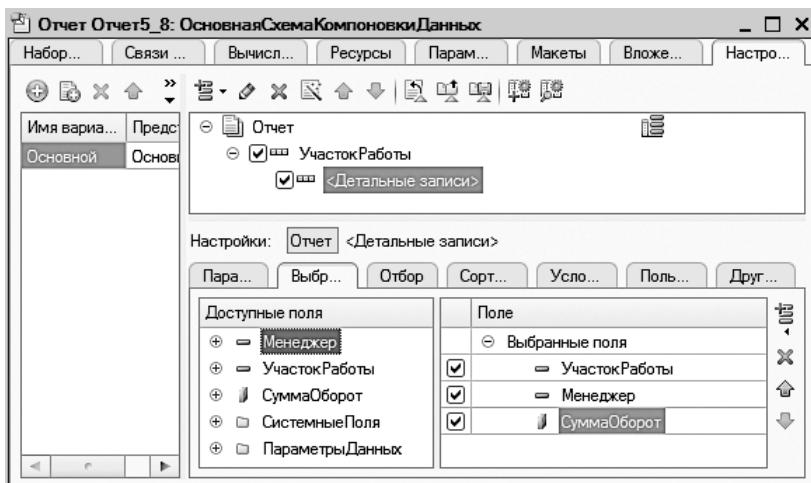


Рис. 5.117. Создание группировок и включение полей для отчета

Отчет по зависимости результатов от стажа

Рассмотрим еще один интересный отчет, который представлен на рис. 5.118. Здесь для руководящего состава организации приводится весьма полезная сводная информация о том, как результаты работы менеджеров зависят от их стажа работы. Так, в первом столбце указывается срок работы в фирме в годах, во втором — оборот продаж менеджеров, имеющих указанный стаж, в третьем — число сотрудников фирмы с указанным стажем работы.

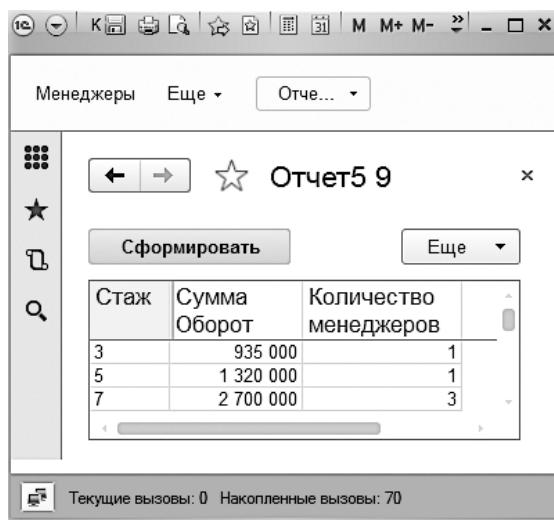


Рис. 5.118. Зависимость суммы продаж от стажа работы

На этот раз после создания нового объекта конфигурации и использования схемы компоновки данных непосредственно введем текст запроса (листинг 5.32). Именно так мы поступали во второй главе книги.

Листинг 5.32. Запрос для извлечения сводной информации

```

ВЫБРАТЬ
    СУММА(РаботаМенеджеровОбороты.СуммаОборот) КАК СуммаОборот,
    Менеджеры.Стаж,
    КОЛИЧЕСТВО(*) КАК КоличествоМенеджеров
ИЗ
    Справочник.Менеджеры КАК Менеджеры
        ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.РаботаМенеджеров.Обороты КАК
    РаботаМенеджеровОбороты
        ПО (РаботаМенеджеровОбороты.Менеджер = Менеджеры.Ссылка)
СГРУППИРОВАТЬ ПО
    Менеджеры.Стаж

```

Здесь в запросе мы использовали знакомые (по второй главе) агрегатные функции — *Сумма* и *Количество*. Технически данный запрос вводится на вкладке *Наборы данных* (рис. 5.119).

Теперь в окне конструктора схемы компоновки данных следует включить в отчет только детальные записи и выбрать следующие поля:

- Стаж;
- СуммаОборот;
- КоличествоМенеджеров.

```

ВЫБРАТЬ
    СУММА(РаботаМенеджеровОбороты.СуммаОборот) КАК СуммаОборот,
    Менеджеры.Стаж,
    КОЛИЧЕСТВО(*) КАК КоличествоМенеджеров
ИЗ
    Справочник.Менеджеры КАК Менеджеры
        ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.РаботаМенеджеров.Обороты
            КАК РаботаМенеджеровОбороты
            ПО (РаботаМенеджеровОбороты.Менеджер = Менеджеры.Ссылка)
СГРУППИРОВАТЬ ПО
    Менеджеры.Стаж

```

Рис. 5.119. Текст запроса на вкладке Наборы данных

Осталось перейти в режим 1С:Предприятие и сформировать разработанный отчет по той информации, которая присутствует в базе данных.

Отчет по зависимости продаж от образования менеджеров

Последний отчет для данной конфигурации представлен на рис. 5.120. В нем указаны объемы сделок, проведенных менеджерами с разным базовым образованием.

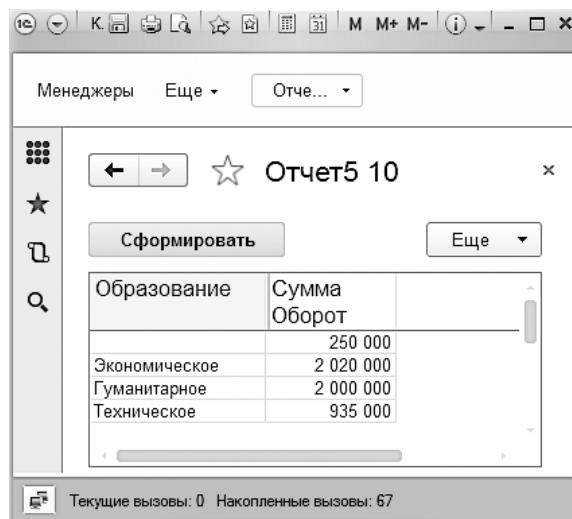


Рис. 5.120. Отчет, отражающий влияние базового образования на эффективность работы

После начальных шагов по созданию нового отчета (новый объект конфигурации — Отчет5_10) в окне конструктора запроса выберем две таблицы — Менеджеры и РаботаМенеджеров.Обороты — и укажем необходимые поля (рис. 5.121).

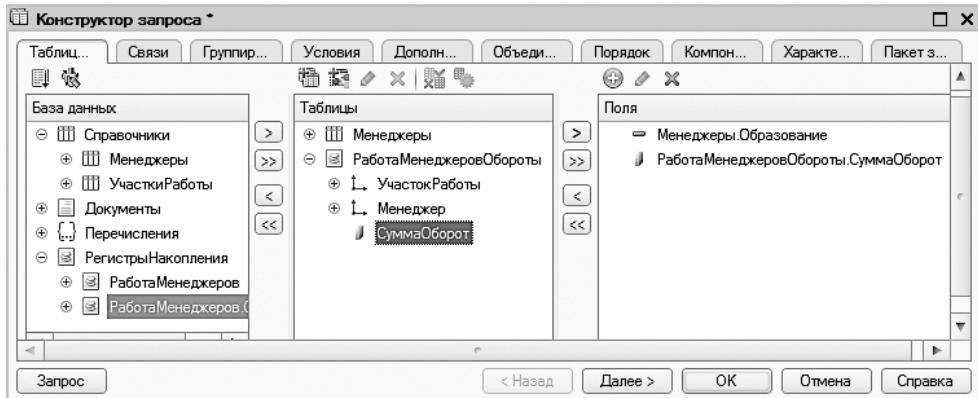


Рис. 5.121. Окно конструктора запроса

Далее в окне конструктора запросов на вкладке **Связи** следует установить связь между таблицами (рис. 5.122).

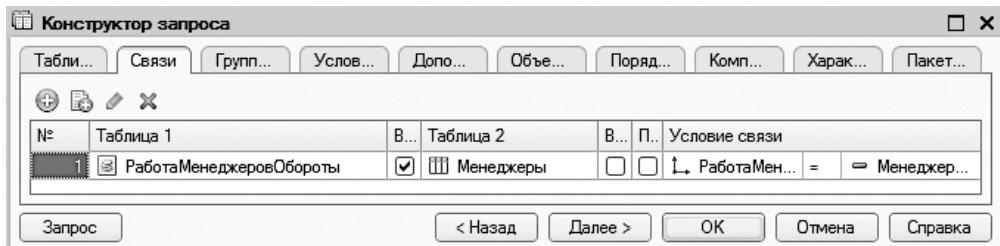


Рис. 5.122. Закладка Связи окна конструктора запросов

После этого на вкладке **Группировка** следует задать группировку по полю **Образование** (рис. 5.123).

На этом действия по работе с конструктором запроса завершаются. Остается щелчком на кнопке **OK** сформировать текст запроса (листиング 5.33).

Листинг 5.33. Запрос для отчета о роли образования

```
ВЫБРАТЬ
    Менеджеры.Образование,
    СУММА(РаботаМенеджеровОбороты.СуммаОборот) КАК СуммаОборот
ИЗ
    РегистрНакопления.РаботаМенеджеров.Обороты КАК РаботаМенеджеровОбороты
        ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Менеджеры КАК Менеджеры
        ПО (РаботаМенеджеровОбороты.Менеджер = Менеджеры.Ссылка)
СГРУППИРОВАТЬ ПО
    Менеджеры.Образование
```

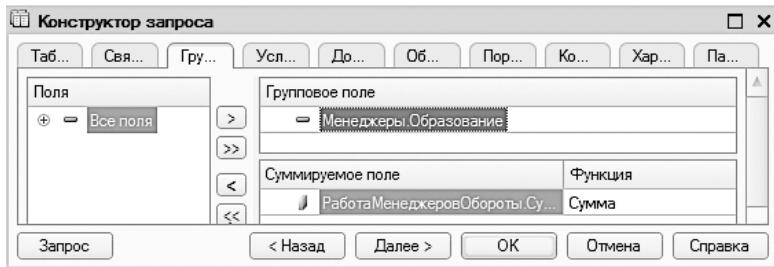


Рис. 5.123. Создание поля группировки в запросе

Теперь перейдем к схеме компоновки данных и на вкладке **Настройки** зададим режим включения детальных записей и выберем два поля для отчета (рис. 5.124). Сам отчет в режиме 1С:Предприятие мы уже видели на рис. 5.120.

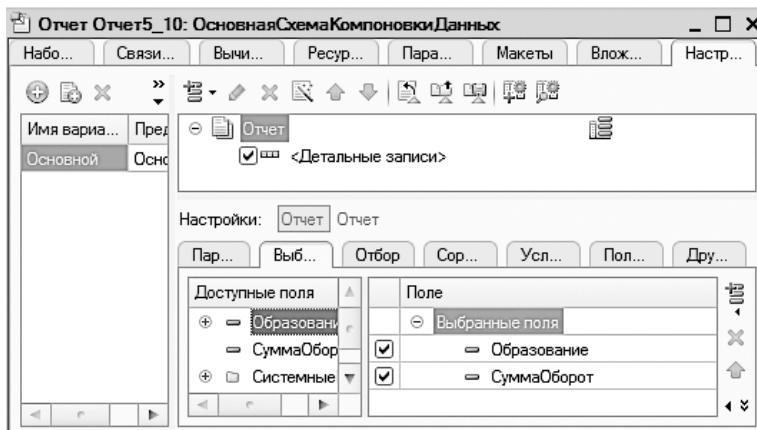


Рис. 5.124. Установки для отчета на вкладке Настройки

Итоги

В данной главе на разных примерах мы продемонстрировали приемы работы на платформе 1С:Предприятия 8.3. Ряд примеров касался программирования форм, что позволило предоставить пользователям необходимый сервис для работы. В других примерах рассматривались вопросы создания конфигурации и формирования отчетов, позволяющих руководству организации оценить эффективность работы менеджеров. Разработанные отчеты дают возможность понять, как стаж работы и базовое образование влияют на объем продаж менеджеров. Полученный результат анализа можно эффективно использовать при принятии решений управленческим составом организации.

Литература

1. *Кашаев С. М.* 1С:Предприятие 8. Учимся программировать на примерах. — СПб.: БХВ-Петербург, 2008. — 336 с.: ил. + CD-ROM.
2. *Кашаев С. М.* 1С:Предприятие 8.1. Учимся программировать на примерах. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2009. — 400 с.: ил. + CD-ROM.
3. *Кашаев С. М.* 1С:Предприятие 8.1. Учимся программировать на примерах. — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2010. — 384 с.: ил. + CD-ROM.
4. *Кашаев С. М.* 1С:Предприятие 8.1. Разработка прикладных решений. — М.: ООО «И.Д. Вильямс», 2009. — 368 с.: ил.
5. *Радченко М. Г., Хрусталева Е. Ю.* 1С:Предприятие 8.2. Практическое пособие разработчика. Примеры и типовые приемы. — М.: 1С-Паблишинг. — 874 с.: ил.
6. *Радченко М. Г.* 1С:Предприятие 8.1. Практическое пособие разработчика. Примеры и типовые приемы. — М.: 1С-Паблишинг; СПб.: Питер, 2007. — 512 с.: ил.
7. Разработка управляемого интерфейса / Ажеронок В. А., Островерх А. В., Радченко М. Г., Хрусталева Е. Ю. — М.: 1С-Паблишинг; 2010. — 723 с.: ил.