

Министерство образования Республики Беларусь

Учреждение образования  
«Гомельский государственный технический  
университет имени П. О. Сухого»

Кафедра «Информатика»

# КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

## ПРАКТИКУМ

по выполнению лабораторных работ  
для студентов специальности 1-40 04 01 «Информатика  
и технологии программирования»  
дневной формы обучения

*Учебное электронное издание  
комбинированного распространения*

Гомель 2024

УДК 004.42(075.8)  
ББК 32.973я73  
К68

Составитель *Н. В. Ермалинская*

Рецензент: руководитель отдела по подготовке кадров ИООО «ЭПАМ Системз»  
*Е. И. Гридина*

К68

**Корпоративные** информационные системы : практикум по выполнению лаборатор. работ для студентов специальности 1-40 04 01 «Информатика и технологии программирования» днев. формы обучения / сост. Н. В. Ермалинская. – Гомель : ГГТУ им. П. О. Сухого, 2024. – 124 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; дисковод CD-ROM ; мышь ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: <https://elib.gstu.by>. – Загл. с титул. экрана.

ISBN 978-985-535-528-2.

Практикум содержит методические рекомендации по реализации способов создания и настройки объектов конфигурации для хранения и вывода данных (справочников, документов, отчетов и пр.) с применением встроенного языка программирования; средств создания и программной настройки процессов учета движения данных в системе (регистров накопления, оборотных регистров); методов настройки пользовательского интерфейса.

Для студентов специальности 1-40 04 01 «Информатика и технологии программирования» дневной формы обучения.

УДК 004.42(075.8)  
ББК 32.973я73

ISBN 978-985-535-528-2

© Ермалинская Н. В., составление, 2024  
© Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», 2024

## Содержание

Предисловие.....	4
ЛАБОРАТОРНАЯ РАБОТА 1. СОЗДАНИЕ ИНФОРМАЦИОННОЙ БАЗЫ, ПОДСИСТЕМ И СПРАВОЧНИКОВ .....	5
Задание 1.1 Знакомство с конфигуратором и создание информационной базы .....	5
Задание 1.2 Создание подсистем .....	9
Задание 1.3 Создание справочников .....	13
ЛАБОРАТОРНАЯ РАБОТА 2. СОЗДАНИЕ ДОКУМЕНТОВ И РЕГИСТРОВ НАКОПЛЕНИЯ .....	25
Задание 2.1 Создание документов .....	25
Задание 2.2 Создание регистра накопления .....	36
ЛАБОРАТОРНАЯ РАБОТА 3. ПРОСТОЙ ОТЧЕТ. МАКЕТЫ И ФОРМЫ.....	45
Задание 3.1 Создание документов .....	45
Задание 3.2 Создание и редактирование макетов и форм.....	49
ЛАБОРАТОРНАЯ РАБОТА 4. РЕГИСТРЫ СВЕДЕНИЙ. ПЕРЕЧИСЛЕНИЯ .....	56
Задание 4.1 Создание периодического регистра сведений.....	56
Задание 4.2 Создание перечислений .....	61
ЛАБОРАТОРНАЯ РАБОТА 5. ПРОВЕДЕНИЕ ДОКУМЕНТА ПО НЕСКОЛЬКИМ РЕГИСТРАМ. ОБОРОТНЫЕ РЕГИСТРЫ НАКОПЛЕНИЯ .....	65
Задание 5.1 Проведение приходной накладной по двум регистрам.....	65
Задание 5.2 Создание оборотного регистра накопления .....	72
ЛАБОРАТОРНАЯ РАБОТА 6. СОЗДАНИЕ ОТЧЕТОВ .....	76
Задание 6.1 Выбор данных из одной таблицы .....	76
Задание 6.2 Выбор данных из двух таблиц .....	81
Задание 6.3 Использование вычисляемого поля в отчете.....	92
ЛАБОРАТОРНАЯ РАБОТА 7. ОПТИМИЗАЦИЯ ПРОВЕДЕНИЯ ДОКУМЕНТОВ .....	96
Задание 7.1 Повышение скорости проведения документа .....	96
Задание 7.2 Автоматический расчет суммы .....	104
Задание 7.3 Контроль остатков .....	118
ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ ДЛЯ РАЗРАБОТКИ ПРИКЛАДНОГО РЕШЕНИЯ .....	123
Литература .....	124

## ПРЕДИСЛОВИЕ

Практикум по дисциплине «Корпоративные информационные системы» призван обеспечить формирование практических навыков использования инструментов и технологий разработки прикладных решений (конфигураций) и эксплуатационной поддержки учетных приложений на базе платформы «1С: Предприятие 8.3».

Разделы практикума содержат методические рекомендации по созданию и программной настройке объектов конфигурации общего и прикладного назначения (подсистем, констант, справочников, перечислений), объектов хранения данных (документов, форм, макетов, регистров сведений), объектов накопления и учета движения данных (регистров накопления), объектов вывода данных (отчетов), а также настройке пользовательского интерфейса конфигурации.

Предложенные к выполнению задания взаимосвязаны между собой, выстроены в порядке возрастания их сложности и ориентированы на создание завершеного прикладного решения в выбранной предметной области.

Для закрепления базовых знаний и выработки у студентов навыков их практического использования предложены индивидуальные задания для разработки прикладных решений.

Материал практикума основан на современной учебной литературе в соответствующей предметной области и базируется на работах таких авторов, как М. Г. Радченко, Е. Ю. Хрусталева, А. Н. Лыгин и др.

Практикум может быть использован как в процессе аудиторной, так и самостоятельной работы студентов.

# ЛАБОРАТОРНАЯ РАБОТА 1

## СОЗДАНИЕ ИНФОРМАЦИОННОЙ БАЗЫ, ПОДСИСТЕМ И СПРАВОЧНИКОВ

**Задание 1.1. Знакомство с конфигуратором и создание информационной базы.** В соответствии с приведенным описанием ознакомится с режимами работы системы и создать новую информационную базу.

### Теоретическая справка

1С: Предприятие – это совокупность различных программных инструментов, с которыми работают разработчики и пользователи. Логически всю систему делят на две части – *конфигурацию* и *платформу*, которая управляет работой конфигурации. То есть основа системы и различные ее настройки под конкретные условия. Сама по себе платформа не может выполнить никаких задач автоматизации, так как она создана для обеспечения работы какой-либо конфигурации. Конфигурация это синоним *прикладного решения*.

Для обеспечения таких возможностей разработки нетипового прикладного решения и внесения изменений в уже используемую конфигурацию система 1С: Предприятие имеет два режима работы: *1С: Предприятие* и *Конфигуратор*.

*Режим 1С: Предприятие* является основным и служит для работы пользователей системы. В этом режиме пользователи вносят данные, обрабатывают и получают итоговые результаты.

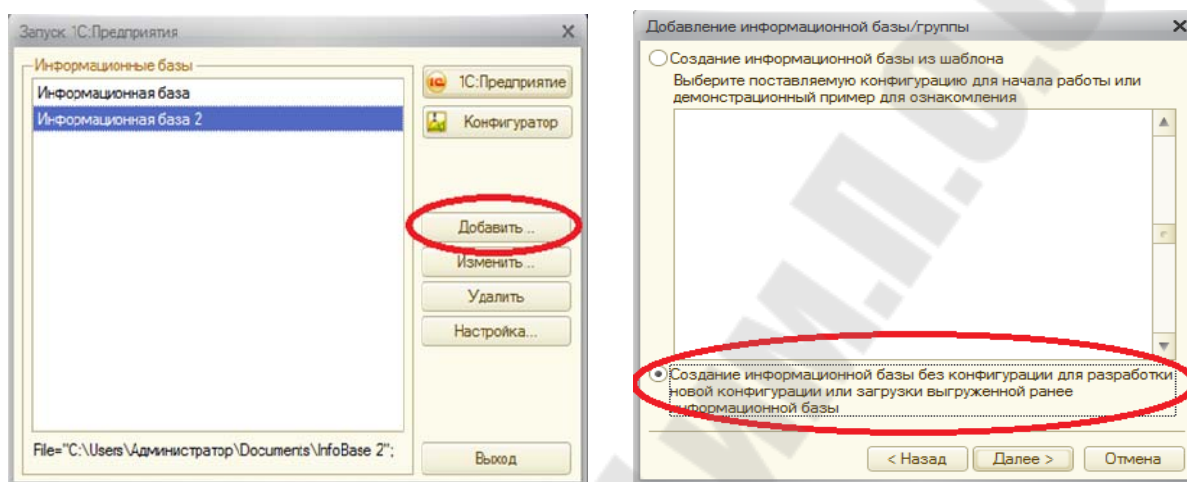
*Режим Конфигуратор* используется разработчиками и администраторами информационных баз. Именно этот режим предоставляет инструменты для изменения существующей или создания новой конфигурации.

### Методические рекомендации по выполнению задания

Шаг 1. Создание информационной базы. Необходимо запустить 1С: Предприятие. В открывшемся диалоге появится список информационных баз (рис. 1.1, а). Если список пуст, система предложит создать новую базу. В таком случае необходимо нажать кнопку *Добавить*. В открывшемся диалоге следует выбрать пункт *Создание новой информационной базы/Создание информационной базы без конфигурации* (рис. 1.1, б).

Далее необходимо задать наименование информационной базы типа «Информационная база Фамилия\_студента» и выбрать тип ее расположения *На данном компьютере или на компьютере локальной сети*. Далее указываем каталог для расположения информационной базы и нажимаем **Готово**.


В диалоге запуска 1С: Предприятия, в списке информационных баз, появится созданная пустая база.



а)

б)

Рис. 1.1. Создание новой информационной базы

Шаг 2. Открытие конфигуратора. Запускаем 1С: Предприятие в режиме **Конфигуратора**. Появится пустое окно конфигуратора. Чтобы включить нужные кнопки, следует нажать на стрелочку **Добавить** или **удалить** кнопки около кнопки .

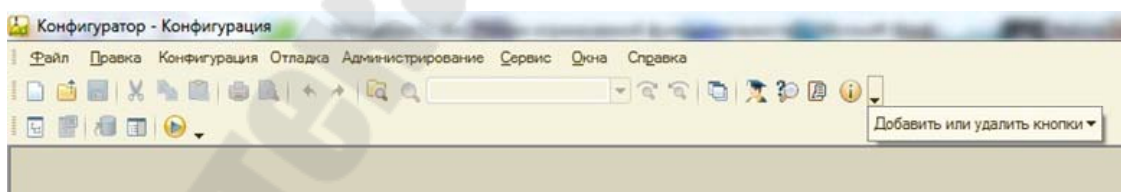


Рис. 1.2. Начальный вид пустого окна конфигуратора

Под заголовком окна находится главное меню конфигуратора (пункты **Файл**, **Правка**, **Конфигурация**, **Администрирование** и др.). Ниже находится панель инструментов конфигуратора, в которую помещены наиболее часто используемые действия, вызываемые из верхнего меню, в виде кнопок-пиктограмм. Если подвести курсор к какой-нибудь пиктограмме и задержать, появится подсказка о ее назначении.

Шаг 3. Открытие дерева объектов конфигурации. Открываем конфигурацию с помощью меню **Конфигурация/Открыть конфигурацию** или соответствующей кнопкой (рис. 1.3, а).

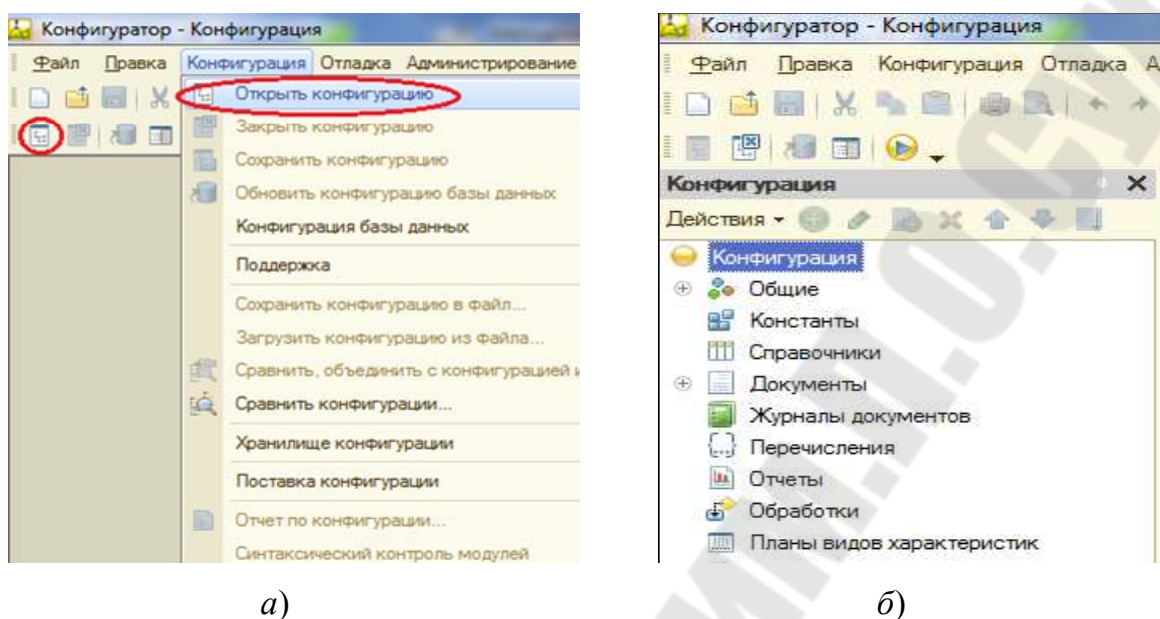


Рис. 1.3. Дерево объектов конфигурации

Дерево объектов конфигурации (рис. 1.3, б) – основной инструмент, с которым работает разработчик. Она описывает структуру данных, которые пользователь будет использовать в режиме работы 1С: Предприятие. Кроме того, конфигурация описывает алгоритмы обработки данных, содержит информацию о том, как будут выглядеть данные на экране и на принтере и т. д.

В дальнейшем платформа на основании этого описания создаст базу данных, которая будет иметь необходимую структуру, и предоставит пользователю возможность работать с этой базой данных.


Объекты конфигурации одного вида отличаются от объектов другого вида тем, что имеют разные свойства. Объекты могут взаимодействовать между собой напрямую или через другие объекты, могут быть сложные объекты, состоящие из более простых. Самое важное качество объектов конфигурации – они представляют собой аналоги реальных объектов, которыми оперирует предприятие в ходе своей работы.

В системе 1С: Предприятие разработчик может использовать только ограниченный набор объектов конфигурации, зашитый в платформе. Он только может добавлять в конфигурацию какой-либо из стандартных объектов, поставляемых системой.

Любой объект конфигурации добавляется несколькими способами, предварительно выделив нужную группу:

– устанавливается курсор на ветку того объекта, который нужно создать и нажимается кнопка *Действия/Добавить*;

– устанавливается курсор на ветку того объекта, который нужно создать и нажимается правая кнопка мыши. В появившемся контекстном меню выбирается пункт *Добавить*;

– устанавливается курсор на ветку того объекта, который нужно создать и нажимается кнопка-пиктограмма *Добавить* .

**Шаг 4.** Задаем свойства конфигурации. В дереве объектов конфигурации выбираем корневой элемент Конфигурация и двойным щелчком мыши открываем окно ее свойств. В поле *Имя* записываем без пробелов название конфигурации, как показано на рис. 1.4.

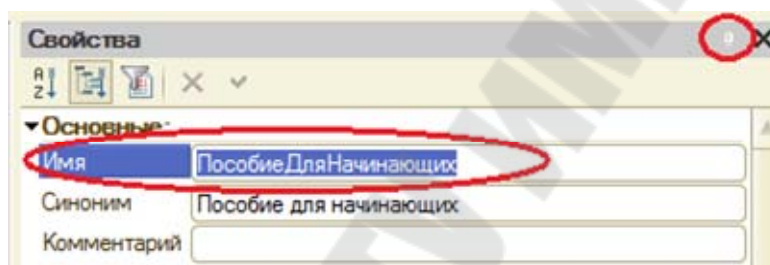


Рис. 1.4. Окно свойств конфигурации

Использование такого синтаксиса (записи без пробелов) позволяет системе лучше различать объекты. Для пользователя же будет виден синоним названия, который лучше читаем.

**Шаг 5.** Запуск режима 1С: Предприятие и отладка. Выполняем пункт меню *Отладка/Начать отладку* или можно нажать соответствующую кнопку на панели инструментов (клавиша F5).

Система проанализирует наличие изменений в конфигурации и выдаст вопрос об обновлении конфигурации базы данных. На вопрос следует ответить *Да*. На экране появится окно 1С: Предприятие.

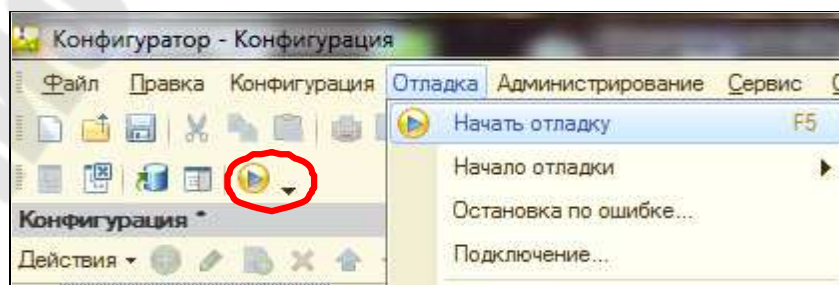


Рис. 1.5. Приемы запуска отладки



### *Контрольные вопросы к заданию 1.1*

1. Что такое конфигурируемость системы 1С: Предприятие?
2. Для чего используются разные режимы запуска системы?
3. Для чего нужно дерево объектов конфигурации?
4. Что такое объекты конфигурации?
5. Как можно добавить новый объект конфигурации?

**Задание 1.2. Создание подсистем.** Создать несколько подсистем (Бухгалтерия, Учет материалов, Оказание услуг, Расчет зарплаты), определяющих логическую структуру прикладного решения, настроить их внешний вид и порядок их следования в интерфейсе 1С: Предприятие.

### **Теоретическая справка**

Подсистемы позволяют выделить в конфигурации функциональные части, на которые логически разбивается создаваемое прикладное решение. Эти объекты располагаются в ветке объектов Общие и позволяют строить древовидную структуру, состоящую из подсистем и подчиненных подсистем. Подсистемы верхнего уровня являются основными элементами интерфейса, так как образуют *разделы прикладного решения*.

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

Таким образом, подсистемы определяют структуру прикладного решения, организуют весь пользовательский интерфейс, позволяют рассортировать различные документы, справочники и отчеты по логически связанным с ними разделам, в которых пользователю будет проще их найти и удобнее с ними работать. При этом каждому конкретному пользователю будут видны лишь те разделы, которые ему нужны в процессе работы.

### **Методические рекомендации по выполнению задания**

Шаг 1. Добавление объекта подсистемы. Производственную деятельность фирмы можно разделить на учет материалов и оказание услуг. Для выполнения специальных административных функций с базой данных потребуется подсистема для администратора.

Необходимо создать пять новых объектов конфигурации *Подсистема* с именами: *Бухгалтерия*, *РасчетЗарплаты*, *УчетМатериалов*, *ОказаниеУслуг*, *Предприятие*.

В режиме Конфигуратора необходимо раскрыть ветку **Общие** в дереве объектов конфигурации, нажав на + слева от нее. Выделить ветку **Подсистемы** и нажать **Добавить** (или кнопка Insert на клавиатуре) или воспользоваться контекстным меню.

Шаг 2. Редактирование объекта подсистемы. После добавление нового объекта откроется окно редактирования объекта конфигурации. В нем необходимо задать имя новой подсистемы – Бухгалтерия. На основании имени система сама создаст синоним – Бухгалтерия (рис. 1.6).

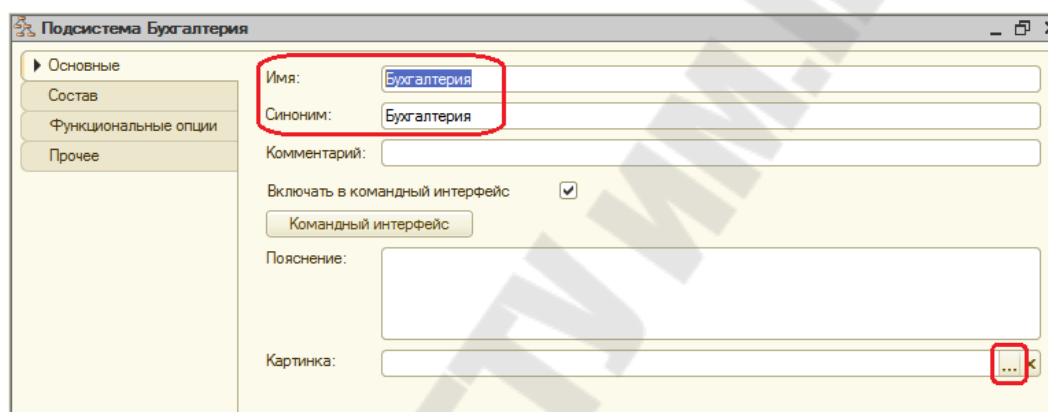
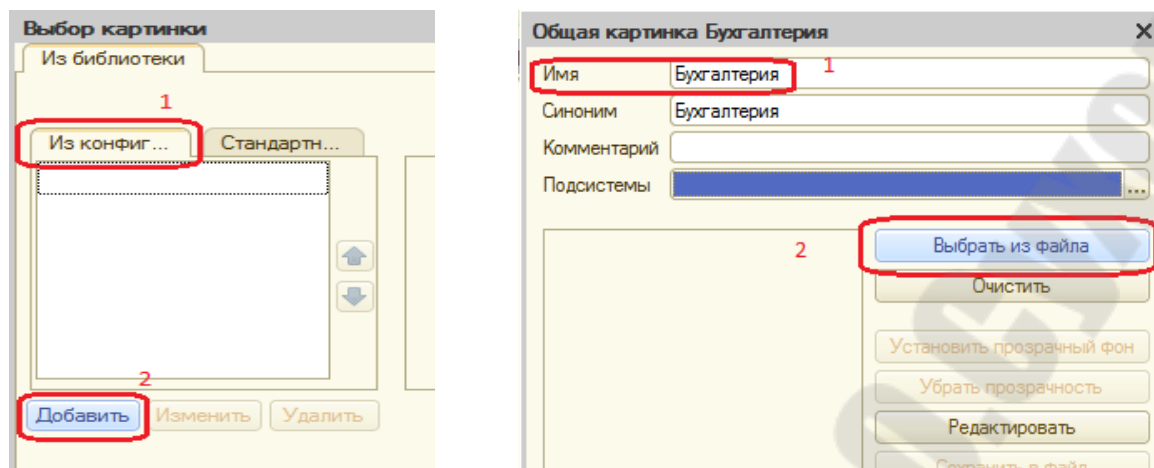


Рис. 1.6. Окно свойств подсистемы *Бухгалтерия*

*Имя* является основным свойством любого объекта конфигурации. Имя может быть любое, начинающееся с буквы и не содержащее пробела и других специальных символов. Имя объекта является уникальным и служит для обращения к свойствам и методам объекта на встроенном языке.

Свойство *Синоним* также есть у любого объекта конфигурации и предназначено для хранения альтернативного наименования объекта, которое будет видно пользователю. На Синоним нет никаких ограничений и его можно задавать в привычном виде.

Шаг 3. Установка картинки подсистемы. Можно задать также картинку (иконку) для отображения подсистемы в окне режима 1С: Предприятие. Для этого необходимо нажать кнопку выбора в поле *Картинка* (рис. 1.6). Далее добавить картинку в список на закладке **Из конфигурации** и нажать кнопку **Добавить** как показано на рис. 1.7, а. Система создаст объект *Общая картинка* и откроет окно его свойств.



а)

б)

Рис. 1.7. Установка картинки для подсистемы

Необходимо дать картинке имя *Бухгалтерия*. Для этого следует нажать **Выбрать из файла**, чтобы задать саму картинку (рис. 1.7, б). После закрытия окна выбора файла картинка появится в списке на закладке **Из конфигурации**. Следует нажать **ОК**.

В интерфейсе 1С: Предприятие в качестве названия раздела будет показан синоним подсистемы, а над ним будет показана данная картинка.

Аналогичным образом необходимо создать подсистемы с именами *УчетМатериалов*, *ОказаниеУслуг*, *Предприятие*, *РасчетЗарплаты*, установив для них соответствующие картинки с именами *Материалы*, *Услуги*, *Зарплата*, *Предприятие*. Должно получиться всего 5 подсистем (рис. 1.8).

**Шаг 4.** Установка требуемого порядка следования подсистем. Допустим, порядок расположения подсистем не соответствует требованиям. Необходимо его изменить.

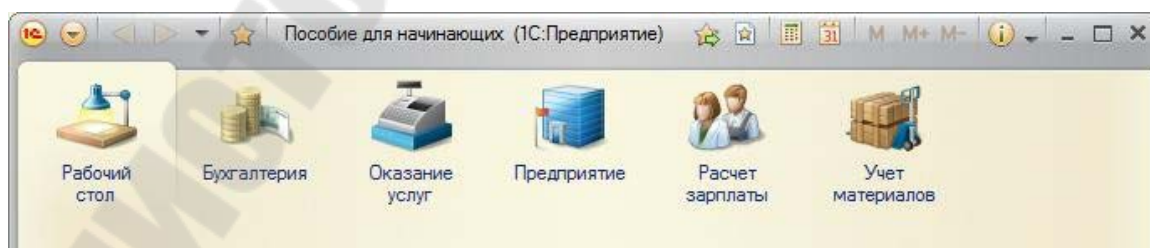


Рис. 1.8. Представление подсистем конфигурации в режиме 1С: Предприятие

Для этого необходимо вернуться в конфигуратор. Выделить корень дерева объектов конфигурации, нажатием правой кнопки мыши вызвать контекстное меню и выбрать пункт **Открыть командный интерфейс конфигурации** (рис. 1.9, а).

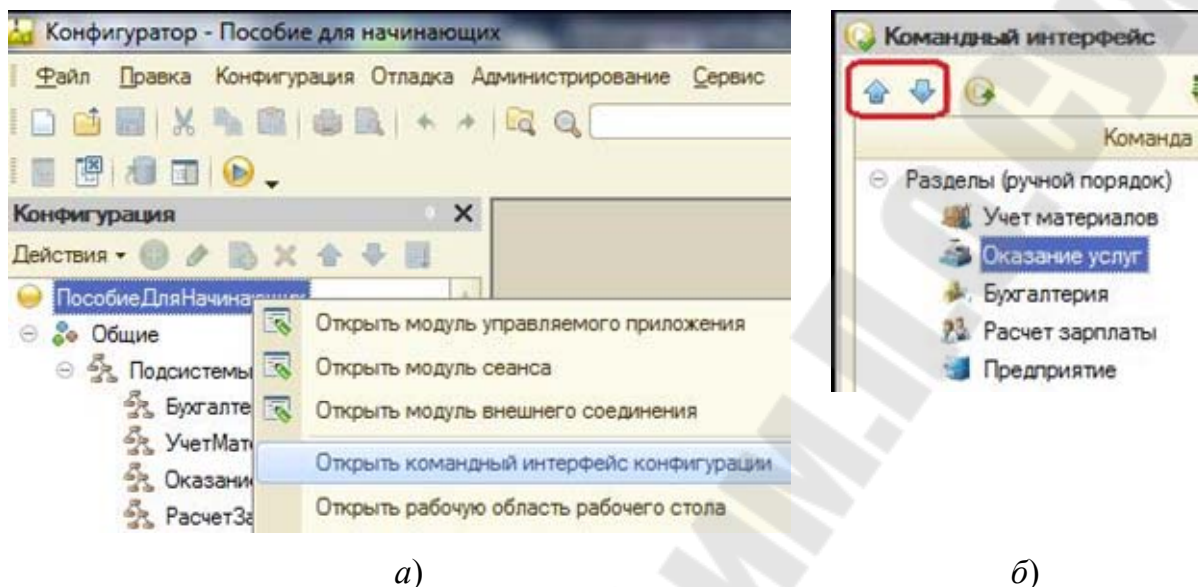


Рис. 1.9. Настройка порядка разделов

В открывшемся окне появится список созданных подсистем. С помощью кнопок *Вверх*, *Вниз* можно изменить порядок расположения разделов в этом списке. Системы необходимо расположить, как показано на рис. 1.9, б.

Далее необходимо закрыть командный интерфейс и запустить 1С: Предприятие в режиме отладки для проверки установленного порядка следования подсистем.

**Важно!** После успешного завершения каждой работы необходимо сохранять конфигурацию: **Администрирование/Выгрузить информационную базу**. Это полезно, когда необходимо вернуться к работающему варианту. Это можно сделать, выполнив команду **Администрирование/Загрузить информационную базу**.

#### Контрольные вопросы к заданию 1.2

1. Для чего используется объект конфигурации Подсистема?
2. Как управлять порядком вывода и отображения подсистем в конфигурации?

**Задание 1.3. Создание справочников.** Создать простой справочник *Клиенты*, справочник с табличной частью *Сотрудники*, справочник с предопределёнными элементами *Склады* и заполнить их данными.

### Теоретическая справка

Объект конфигурации *Справочник* предназначен для работы со списками данных. Как правило, в работе любой фирмы используются списки сотрудников, товаров, клиентов, поставщиков и т. д. Свойства и структура этих списков описываются в объектах *Справочник*, на основе которых платформа создает в базе данных таблицы для хранения информации из этих справочников.

Справочник состоит из *элементов*. Например, для справочника *сотрудники* элементом является сотрудник, для справочника *товаров* – товар. Пользователь в процессе работы может самостоятельно добавлять новые элементы в справочник. В базе данных каждый элемент справочника представляет собой отдельную запись (строчку) в основной таблице, хранящей информацию из этого справочника.

Каждый элемент справочника, как правило, содержит более подробно описывающую этот элемент информацию. Набор такой информации является одинаковым для всех элементов справочника и для описания такого набора используются *реквизиты* объекта конфигурации *Справочник*, которые также являются объектами конфигурации. Можно это представить наглядно как таблицу: Справочник *Товары* содержит список товаров, имеющих реквизиты (колонки в таблице *Товары*) *Производитель*, *Поставщик*, *Наименование*.

Можно использовать стандартные реквизиты, свойственные выбранному типу *Справочника*, а можно создать свои, более удобные. Каждый элемент справочника может содержать некоторый набор информации, одинаковой по структуре, но различной по количеству, и предназначенной для разных элементов справочника.

Например, каждый элемент справочника *Сотрудники* (каждый сотрудник) может содержать информацию о составе семьи сотрудника. Для одного сотрудника это будет только жена, для другого – жена, сын, дочь.

Для описания подобной информации могут быть использованы **табличные части** объекта конфигурации *Справочник*, являющиеся подчиненными ему объектами конфигурации. В этом случае в базе

данных будут созданы дополнительные таблицы для хранения табличных частей, подчиненных конкретному элементу справочника.

Элементы одного справочника могут быть подчинены элементам или группам другого справочника. Например, Справочник *Единицы Измерения* может быть подчинен справочнику *Товары*.

Иногда возникают ситуации, когда необходимо, чтобы в справочнике некоторые элементы существовали всегда, независимо от действий пользователя. В таком случае *Справочник* позволяет описать любое количество таких неизменяемых элементов, называемых **предопределенными элементами справочника**.

Предопределенные элементы отличаются от обычных тем, что создаются в конфигураторе и пользователь не может их удалить. Все остальные действия с ними он может делать, в том числе переименовывать. В интерфейсе предопределенные элементы помечены специальной пиктограммой.

Удобно представлять справочник в разном виде – *список, дерево, в определенной последовательности* и т. п. Для этого используется объект конфигурации **Форма**.

Система может самостоятельно создать все формы, которые нужны для представления данных Справочника.

### **Методические рекомендации по выполнению задания**

Шаг 1. Создание простого справочника. Согласно выбранной предметной области фирма, для которой разрабатывается прикладное решение, оказывает услуги по ремонту бытовой техники. Таким образом, для ведения учета потребуется хранить *список сотрудников предприятия, список клиентов, перечень услуг, оказываемых фирмой, список расходуемых материалов и список складов*, где они хранятся.

Создадим справочник, в котором будут храниться данные о клиентах. Для этого необходимо открыть конфигурацию, выделить в дереве объектов ветку **Справочники**, нажать **Добавить**. В свойствах созданного объекта указать имя справочника *Клиенты*. На основании имени система создаст синоним – *Клиенты*.

Представление объекта (название объекта в единственном числе) необходимо задать, как показано на рис. 1.10.

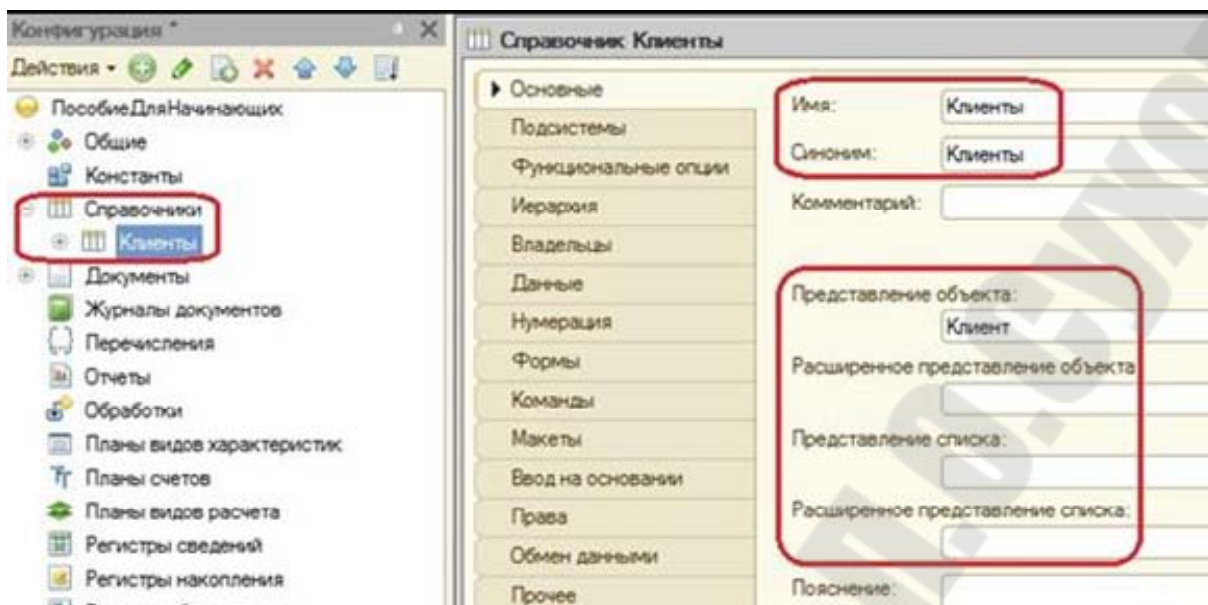


Рис. 1.10. Окно свойств объекта *Справочник*

На закладке *Подсистемы* необходимо указать, в каких подсистемах будет отображаться справочник. Логично, что список клиентов должен быть доступен в разделе *Оказание услуг* и *Бухгалтерия*. Также на закладке *Данные* следует отредактировать длину наименования (установить 50 символов).

Шаг 2. Настройка команды добавления нового элемента справочника. Команда для открытия списка справочника, как и команда для создания его новых элементов, добавляется в интерфейс тех подсистем, в которых будет отображаться справочник. Но команда создания новых элементов по умолчанию невидима в интерфейсе приложения. Это объясняется тем, что возможность просматривать списки справочника нужна почти всегда, а возможность создания новых элементов – редко. Поэтому соответствующую команду следует включать только для некоторых справочников.

Для того, чтобы сделать доступной команду создания новых клиентов на панели действий раздела *ОказаниеУслуг*, следует в дереве объектов конфигурации выделить ветку *Подсистемы*, далее вызвать ее контекстное меню и выбрать пункт ***Все подсистемы***.

В открывшемся окне *Все подсистемы* слева в списке *Подсистемы* выделим *ОказаниеУслуг*, справа в списке *Командный интерфейс* отразятся все команды выбранной подсистемы. В группу ***Панель действий.Создать*** добавилась команда ***Клиент: создать*** для

создания нового элемента справочника, но она невидима по умолчанию. Необходимо включить видимость этой команды, поставив соответствующую галочку (рис. 1.11).

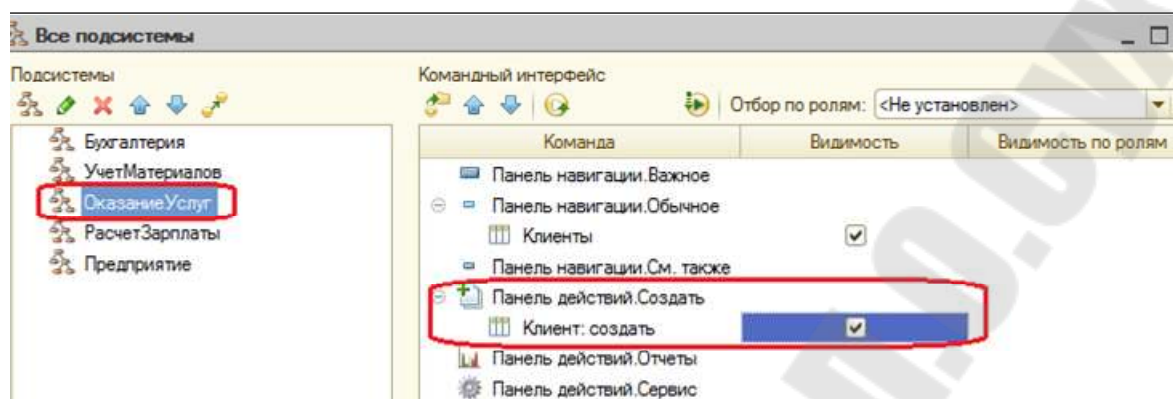


Рис. 1.11. Окно командного интерфейса

Для подсистемы *Бухгалтерия* никаких команд добавлять в панель действий не нужно. В данном случае предполагается, что основную ежедневную работу с клиентами ведет менеджер, занимающийся оказанием услуг. В том числе создает в базе новых клиентов. А бухгалтерия просто обрабатывает имеющиеся в базе данные для получения отчетности.

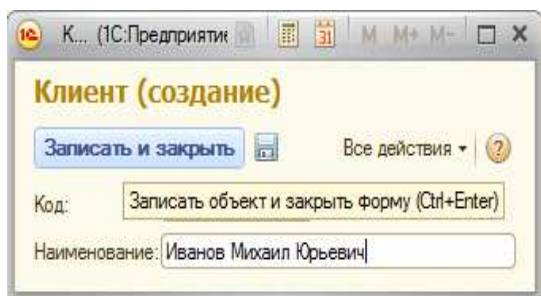
Далее следует закрыть окно редактирования справочника *Клиенты*, всегда отвечая утвердительно на вопрос обновления конфигурации и принятии изменений.

**Шаг 3.** Создание элементов справочника. Для добавления новых элементов необходимо выполнить команду *Клиент* в панели действий раздела *Оказание услуг*. Откроется форма для создания элемента справочника – основная форма объекта. Следует внести имя нового клиента – Иванов Михаил Юрьевич, код создается автоматически (рис. 1.12, а). Нажмите *Записать и закрыть*.

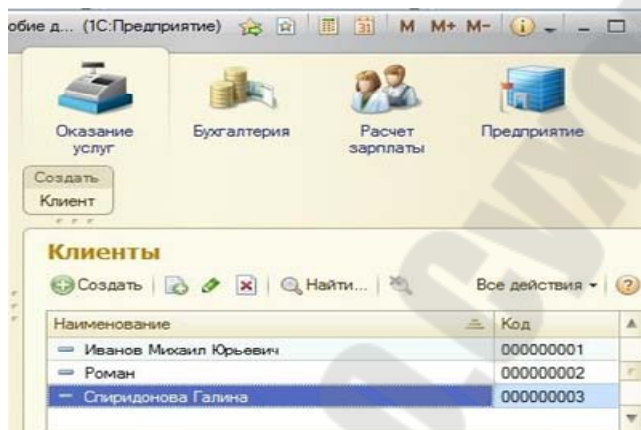
Аналогично добавляем еще одного клиента с именем Роман.

Последнего клиента необходимо добавить с использованием *Формы списка клиентов*. Для этого следует выполнить команду *Клиенты* в панели навигации раздела *Оказание услуг*. Справа от панели навигации откроется основная форма списка. Добавить новый элемент в справочник можно кнопкой *Создать* (рис. 1.12, б).





а)



б)

Рис. 1.12. Создание элементов справочника

Создадим клиента с наименованием Спиридонова Галина.

Следует обратить внимание, что поле *Наименование* подсвечивается красным пунктиром – это значит, что у него стоит свойство обязательного заполнения. Элементы справочника редактируются двойным нажатием на него.

Шаг 4. Создание справочника с табличной частью. Необходимо создать второй справочник *Сотрудники*. В нем будет храниться не только Ф.И.О. сотрудника, но и информация о его прошлых работах. Эта информация однородна по структуре, но разная по количеству, поэтому для ее хранения будем использовать табличную часть справочника.

Добавляем новый объект конфигурации *Справочник* под именем *Сотрудники*. Представление объекта – *Сотрудник*. Расширенное представление списка – *Список сотрудников*. Нажимаем *Далее* и на закладке *Подсистемы* отмечаем видимость справочника в подсистемах *ОказаниеУслуг* и *РасчетЗарплаты*.

На вкладке *данные* оставляем по умолчанию длину и тип кода, а длину наименования справочника следует задать равной 50 символов.

Шаг 5. Создание справочника с табличной частью справочника.

Необходимо создать второй справочник *Сотрудники*. В нем будет храниться не только Ф.И.О. сотрудника, но и информация о его прошлых работах. Эта информация однородна по структуре, но разная по количеству, поэтому для ее хранения будем использовать табличную часть справочника.

Для начала следует создать новый объект конфигурации Справочник под именем *Сотрудники*. Представление объекта – Сотрудник. Представление списка не заполняем, а Расширенное представление списка – Список сотрудников. Нажимаем *Далее* и на закладке *Подсистемы* отмечаем *ОказаниеУслуг* и *РасчетЗарплаты*.

При оказании услуг должен быть указан сотрудник, оказавший эти услуги, и по результатам этой работы будет начисляться зарплата каждому сотруднику. На закладке *Данные* оставляем по умолчанию длину и тип кода, а длину наименования справочника задаем равной 50 символов.

Добавляем в справочник табличную часть с именем *ТрудоваяДеятельность*. Для этого нажимаем кнопку *Добавить табличную часть* над списком табличных частей справочника (рис. 1.13).

Создадим **реквизиты** табличной части (колонки) *ТрудоваяДеятельность*. Для этого необходимо нажать кнопку *Добавить реквизит* над списком табличных частей справочника (рис. 1.14).

Для данного справочника необходимо создать следующие реквизиты: *Организация* – тип Строка, длина 100; *НачалоРаботы* – тип Дата, состав даты – Дата; *ОкончаниеРаботы* – тип Дата, состав даты – Дата; *Должность* – тип Строка, длина 100.

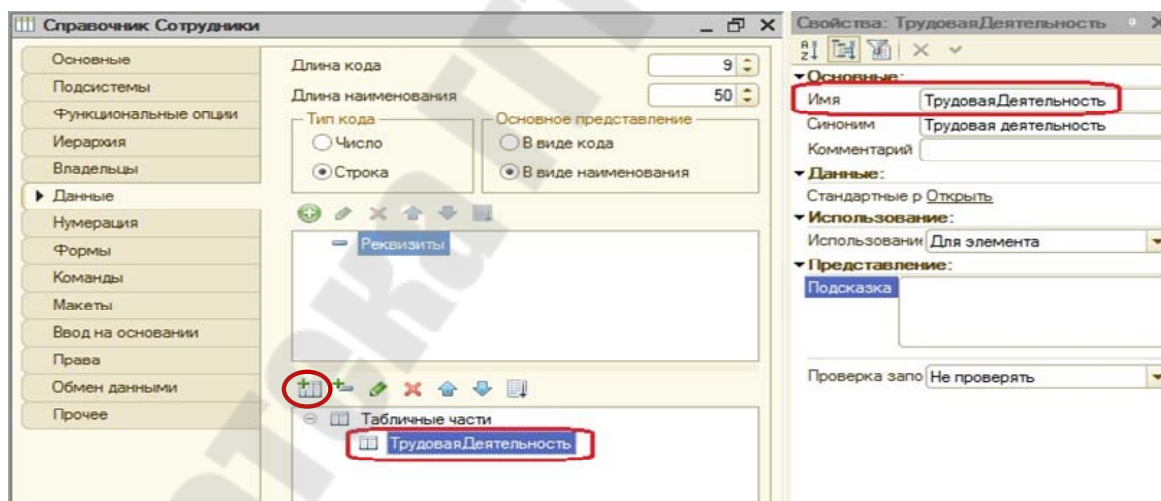


Рис. 1.13. Создание табличной части справочника

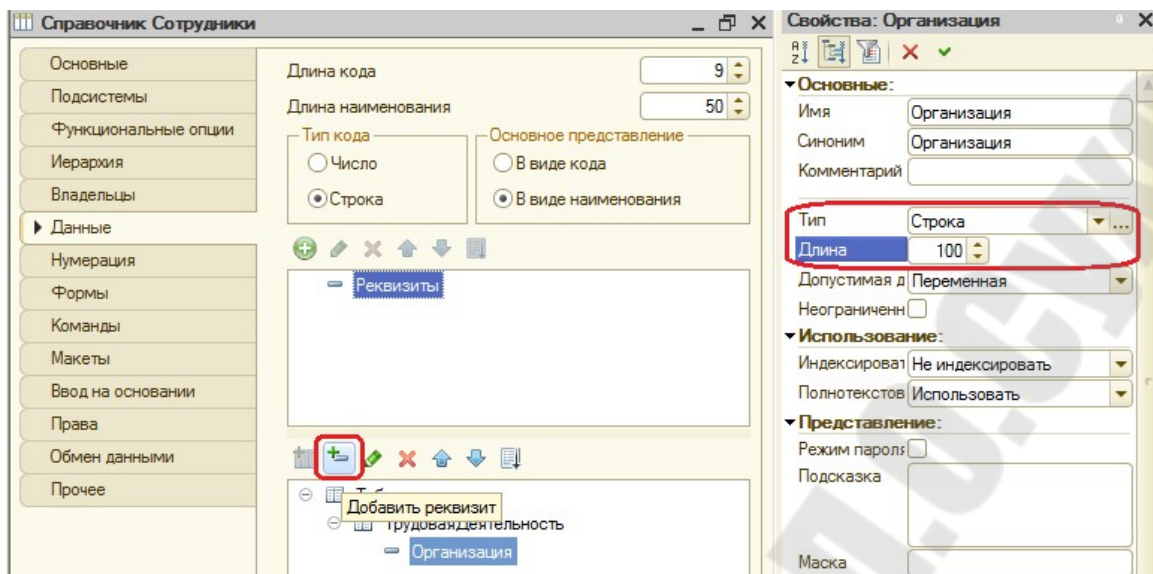
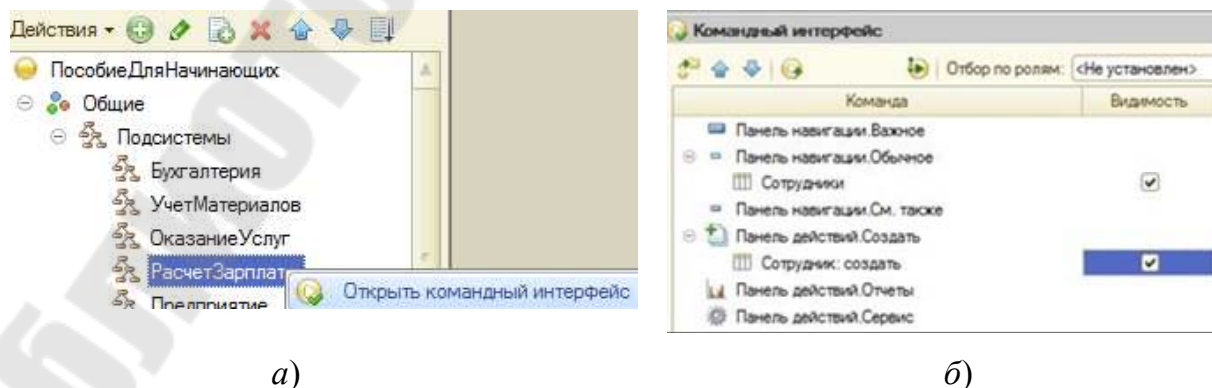


Рис. 1.14. Создание реквизита табличной части справочника

**Шаг 6.** Настройка пользовательского интерфейса. Отредактируем командный интерфейс, чтобы было удобнее вводить новые элементы справочника. Сделаем видимой в панели действий подсистемы *РасчетЗарплаты* команду для создания новых сотрудников. Для этого необходимо выделить подсистему *РасчетЗарплаты*, контекстное меню – **Открыть командный интерфейс** (рис. 1.15, а).

В появившемся окне ставим галочку напротив команды **Сотрудник: создать**, чтобы эта команда была видимой в пользовательском интерфейсе (рис. 1.15, б).

На этом создание справочника *Сотрудники* завершено. Необходимо закрыть окно редактирования справочника, запустить 1С: Предприятие в режиме отладки и проверить наличие видимой команды создания нового сотрудника в подсистеме *РасчетЗарплаты*.



а)

б)

Рис. 1.15. Настройка видимости кнопки создания сотрудника

Шаг 7. Заполнение табличной части справочника. В режиме 1С: Предприятие необходимо выполнить команду **Создать/Сотрудник**. Откроется форма для создания элемента справочника – *основная форма объекта*. Эта форма содержит табличную часть с реквизитами, которые были описаны выше. Для создания нового элемента необходимо ввести его имя (имя сотрудника), нажать кнопку **Добавить** и далее ввести его реквизиты.

Требуется создать следующих сотрудников:

Гусаков Николай Дмитриевич (организация – ЗАО «НТЦ», начало работы – 01.02.2000, окончание работы – 16.04.2003, должность – ведущий специалист);

Деловой Иван Сергеевич (организация – ООО «Автоматизация», начало работы – 22.01.1996, окончание работы – 31.12.2002, должность – инженер; организация – ЗАО «НПО СпецСвязь», начало работы – 20.06.1986, окончание работы – 21.01.1995, должность – начальник производства);

Симонов Валерий Михайлович (организация – ООО «Строй-Мастер», начало работы – 06.02.2001, окончание работы – 03.04.2004, должность – прораб).

Шаг 8. Создание иерархического справочника. Создадим справочник *Номенклатура*. Он будет содержать информацию об услугах, которые оказывает фирма и материалах, которые при этом могут быть использованы. Они будут логически собраны в группы.

В режиме конфигуратора создадим новый объект *Справочник* и назовем его *Номенклатура*. Так как понятие «номенклатура» не имеет единственного числа, никаких представлений можно не задавать.

Переходим на вкладку **Подсистемы**. Задаем доступность справочника в разделах *Бухгалтерия*, *УчетМатериалов* и *ОказаниеУслуг*.

Переходим на закладку **Иерархия** и установите флажок **Иерархический справочник** (рис. 1.16).

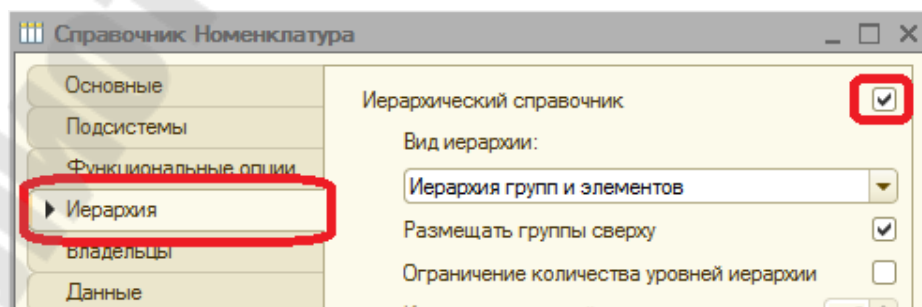


Рис. 1.16. Выбор иерархического типа справочника

На закладке *Данные* устанавливаем длину наименования 100 символов.

Настраиваем интерфейс приложения, включив команду создания новых элементов списка номенклатуры в подсистемах *УчетМатериалов* и *ОказаниеУслуг*. Для этого выделяем ветвь *Подсистемы*, через контекстное меню выбираем *Все подсистемы*. В открывшемся окне слева выделяем *УчетМатериалов*.

Справа отразятся все команды выбранной подсистемы. В группе *Панель действий.Создать* включаем видимость у команды *Номенклатура: создать*. Аналогичным образом настраиваем интерфейс в подсистеме *ОказаниеУслуг* (рис. 1.17).

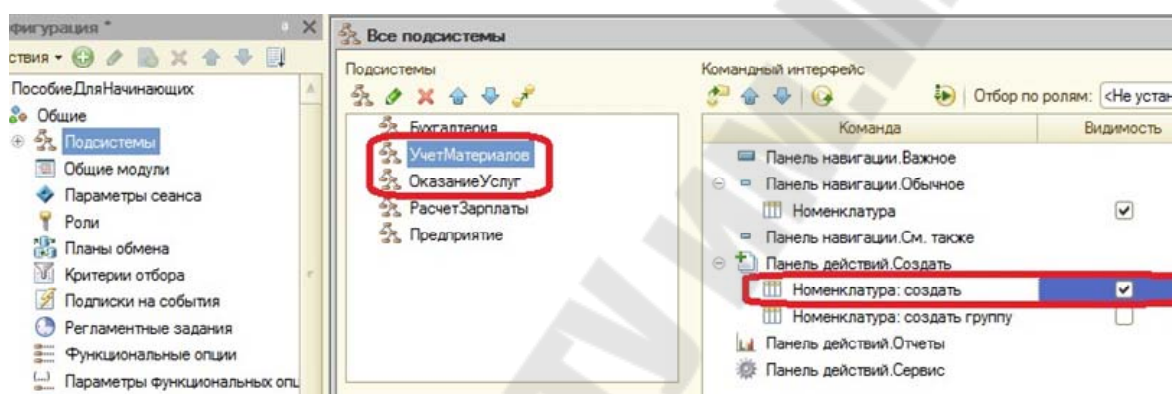


Рис. 1.17. Настройка видимости кнопки создания нового элемента справочника

Заполняем справочник *Номенклатура*. Для этого закрываем окно редактирования справочника и запускаем 1С: Предприятие в режиме отладки.

В открывшемся окне видно, что в панели навигации разделов *Учет материалов*, *Оказание услуг* и *Бухгалтерия* появилась команда *Номенклатура*.

Необходимо выполнить команду *Номенклатура* в панели навигации раздела *Учет материалов*. Справа от панели навигации откроется основная форма списка (рис. 1.18).

Необходимо создать две группы в корне справочника: *Материалы* и *Услуги* с помощью кнопки *Создать группу* в командной панели формы списка (рис. 1.18). Поля Родитель и Код заполнять не нужно.

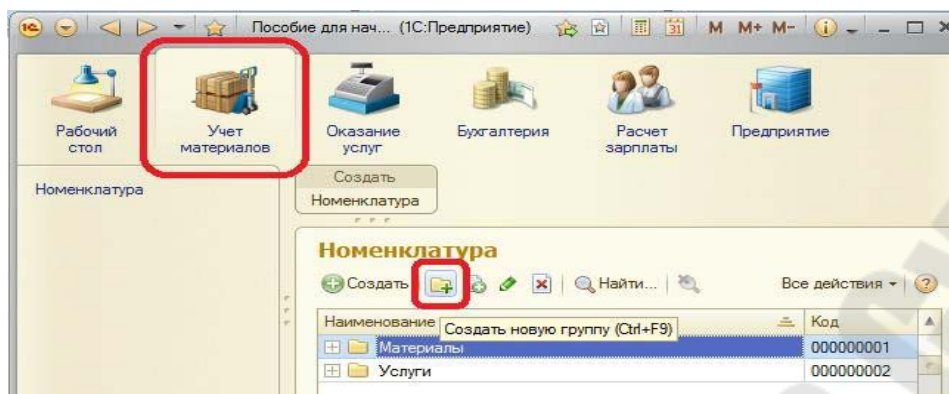


Рис. 1.18. Заполнение иерархического справочника

Затем закрываем группу *Материалы*, нажав на крестик слева от нее. Далее создаем в ней 5 элементов: строчный трансформатор Samsung, строчный трансформатор GoldStar, транзистор Philips 2N2369, шланг резиновый, кабель электрический.

Если новый элемент добавляется из формы списка в некоторую открытую группу, то система автоматически подставляет в качестве родителя эту группу. В данном случае родитель – *Материалы*.

Если элемент добавляется командой **Номенклатура: Создать**, то родителя необходимо устанавливать вручную.

Раскрываем группу *Услуги* и создаем в ней услуги: 1) по ремонту телевизоров: диагностика, ремонт отечественного телевизора, ремонт импортного телевизора; 2) по установке стиральных машин: подключение воды, подключение электричества.

Далее распределим услуги по двум смысловым группам: услуги по ремонту телевизоров и услуги по установке стиральных машин. Для этого в группе *Услуги* создаем еще две группы: *Телевизоры* и *Стиральные машины*.

Есть несколько способов распределения элементов в группы.

Первый – ставим курсор на услугу, которую необходимо перетащить в другую группу, кнопка **Все действия/Переместить в группу**. Можно переместить сразу несколько элементов, выделив их с помощью **Ctrl**.

Второй способ – выбрать услугу двойным щелчком и установить в поле *Родитель* нужную группу.

Третий способ – самый простой – выбрать услуги мышью и ею перетащить в нужную группу.

Любым из предложенных способов перемещаем услуги «диагностика», «ремонт отечественного телевизора», «ремонт импортного

телевизора» в группу *Телевизоры*, остальное – в группу *Стиральные машины*.

В группе *Материалы* создаем две группы: «радиодетали» и «прочее». «Кабель электрический» и «шланг резиновый» перемещаем в группу *Прочее*. Остальное – в группу *Радиодетали*.

Переключаем представление списка в виде дерева – *Все действия/Режим просмотра/Дерево* (рис. 1.19).

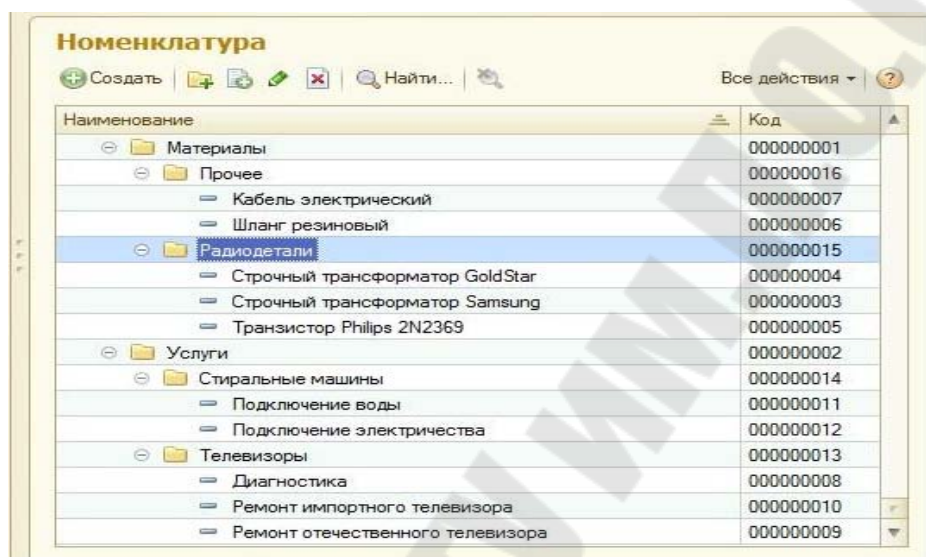
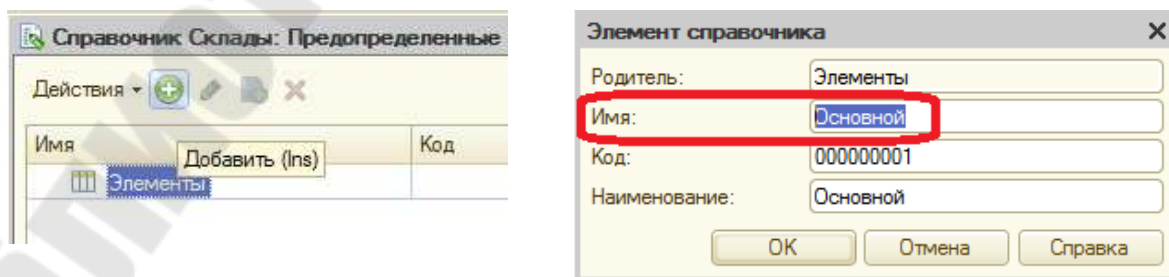


Рис. 1.19. Создание иерархического справочника

**Шаг 9.** Создание справочника с predetermined elements. Создаем справочник *Склады* и переходим в окне свойств на закладку *Прочее* и нажмите кнопку *Предопределенные*. Появится список predetermined elements of the directory. Нажимаем кнопку *Добавить* и задаем имя *Основной* (рис. 1.20, б).

Имя служит для обращения к нему напрямую через встроенный язык (рис. 1.20, а). Наименование видит и изменяет пользователь, имя – пользователь не видит и изменить не может.



а)

б)

Рис. 1.20. Создание predetermined elements of the directory

Настраиваем командный интерфейс для удобства ввода новых элементов справочника (см. выше). Выделяем *Подсистемы*, контекстное меню – *Все подсистемы*. Выделяем в списке слева *УчетМатериалов* и включите видимость команды **Склад: Создать**.

В режиме отладки можно увидеть, что в панели действий раздела *Учет материалов* появилась команда **Склад** для создания новых складов, а в панели навигации разделов *Учет материалов* и *Оказание услуг* команда **Склады**.

Выполним команду **Склады** в панели навигации раздела *Учет материалов*. Справа от панели навигации откроется основная форма списка. В списке складов уже есть один элемент с наименованием *Основной*. Это предопределенный элемент, который был недавно задан в конфигураторе, обозначается специальным значком. Добавляем еще один склад, назвав его «розничный».

#### *Контрольные вопросы к заданию 1.3*

1. Для чего предназначен объект конфигурации Справочник?
2. Для чего используются реквизиты и табличные части справочника?
3. Зачем нужны иерархические справочники?
4. Чем с точки зрения конфигурации отличаются обычные элементы справочника от предопределенных элементов?
5. Как отобразить команды создания нового элемента справочника в интерфейсе подсистем?



## ЛАБОРАТОРНАЯ РАБОТА 2 СОЗДАНИЕ ДОКУМЕНТОВ И РЕГИСТРОВ НАКОПЛЕНИЯ

**Задание 2.1. Создание документов.** В соответствии с приведенным описанием создать и заполнить документы Приходная накладная и Оказание услуги.

### Теоретическая справка

Объект конфигурации *Документ* предназначен для описания информации о совершенных хозяйственных операциях или о событиях в организации. Свойства и структура реальных документов описываются в объектах *Документ*, на основе которых платформа создает в базе данных таблицы для хранения информации из этих документов.

*Документ* обладает способностью проведения. Факт проведения документа означает, что событие, которое он отражает, повлияло на состояние учета. До тех пор, пока документ не проведен, состояние учета неизменно, и документ – не более чем черновик, заготовка.

Каждый документ может содержать дополнительную информацию, которая подробнее его описывает – *реквизиты*. Всегда существуют стандартные реквизиты – *Дата* и *Номер*. Также *Документ* может содержать *табличную часть* для описания набора данных. Для визуализации документа существует несколько основных форм.

### Методические рекомендации по выполнению задания

Шаг 1. Создание документа *Приходная накладная*. Согласно выбранной предметной области фирма, процесс которой автоматизируются, ремонтирует телевизоры и устанавливает стиральные машины. В обоих случаях требуются некоторые расходуемые в процессе оказания услуг материалы, поэтому двумя событиями в хозяйственной деятельности фирмы будут поступление материалов и оказание услуг.

Для отражения этих событий в базе данных необходимо создать два документа: *Приходная накладная* и *Оказание услуги*. Первый будет фиксировать факты поступления в фирму необходимых материалов, а второй – фиксировать оказание услуг и расход материалов, которые используются при оказании этих услуг.

Открываем конфигуратор и добавьте новый объект – *Документ*. Задаем ему имя – *Приходная Накладная*, представление списка – *Приходные накладные*.

Переходим на вкладку *Подсистемы* и отмечаем *Учет Материалов* и *Бухгалтерия*. На вкладке *Данные* создаем реквизит документа с именем *Склад* с помощью кнопки *Добавить*. Для реквизита выбираем ссылочный тип данных *СправочникСсылка. Склады*. Этот тип стал доступен в конфигурации после создания справочника *Склады*.

Работа в фирме организована так, что почти всегда поступающие товары идут на основной склад, поэтому устанавливаем для созданного реквизита свойство *автозаполнения*. В свойствах реквизита находим *Значение заполнения* и выбираем в нем *Основной*. Таким образом, при создании нового документа склад будет сразу заполняться значением *Основной* и пользователю не придется делать это вручную.

После этого добавим в документ табличную часть с именем *Материалы*. Для этого нажимаем кнопку *Добавить табличную часть* над списком табличных частей документа. Кроме имени табличной части устанавливаем свойство *Проверка заполнения* в значение *Выдавать ошибку*. Таким образом задается условие, что документ *Приходная накладная* обязательно должен содержать список приходимых материалов, иначе будет выдано сообщение об ошибке и документ не будет сохранен.

Создаем реквизиты табличной части *Материалы*: *материал*, тип *СправочникСсылка.Номенклатура*; *количество*, тип *Число*, длина 15, точность 2 знака после запятой, неотрицательное; *цена*, тип *Число*, длина 15, точность 2, неотрицательное; *сумма*, тип *Число*, длина 15, точность 2, неотрицательное (рис. 2.1).

Для каждого реквизита также задаем свойство *Проверка заполнения* в *Выдавать ошибку*. Тем самым при записи документа будет проверяться на заполнение не только табличная часть, но и ее отдельные реквизиты (колонки). Закрываем окно свойств.

Шаг 2. Настройка командного интерфейса. Чтобы в подсистеме *Учет Материалов* была доступна команда создания новых документов, необходимо выделить ветвь *Подсистемы*, в контекстном меню выбрать *Все подсистемы*, в группе *Панель действий.Создать* включить видимость у команды *Приходная накладная: создать* (рис. 2.2).

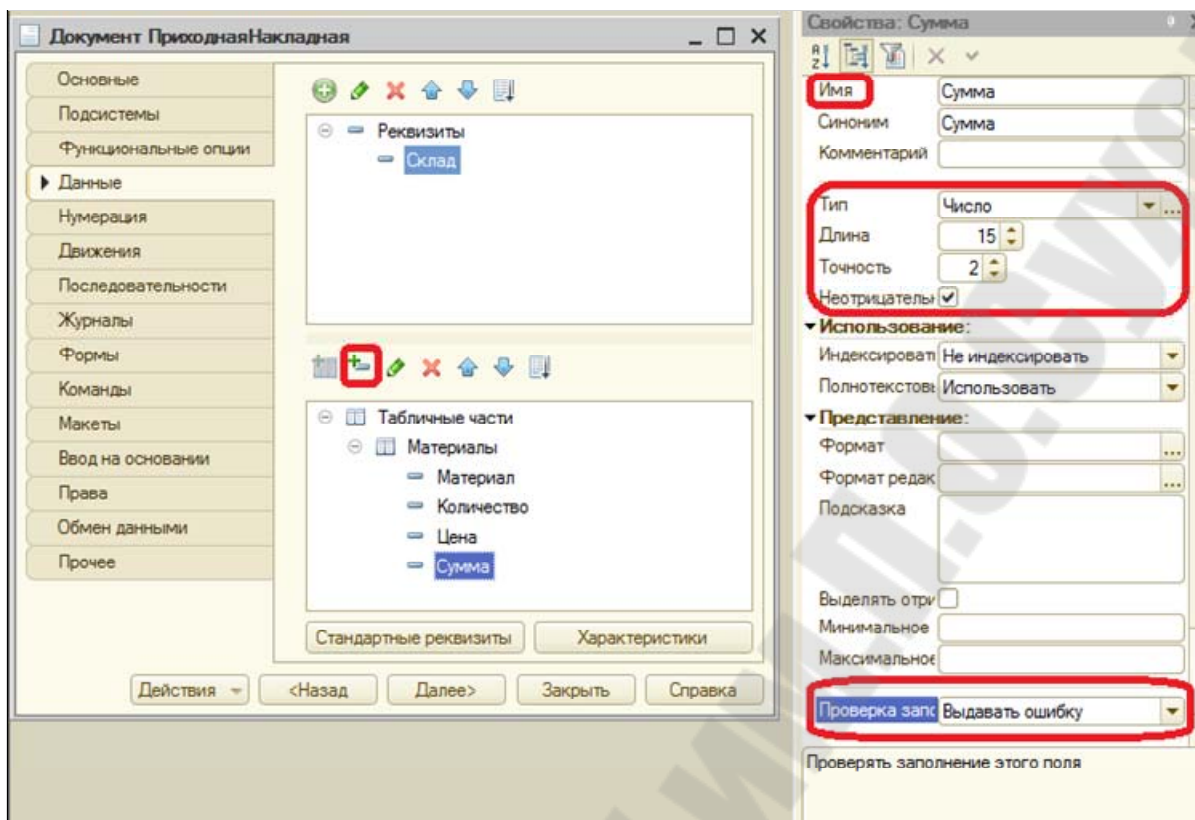


Рис. 2.1. Создание реквизитов табличной части *Материалы*

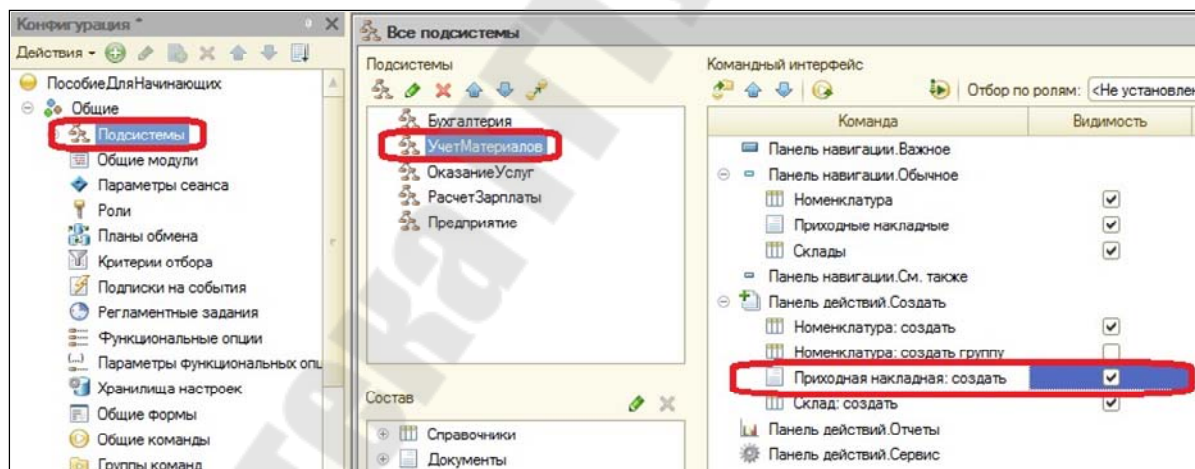


Рис. 2.2. Настройка видимости команды создания документа

**Шаг 3.** Создание документа. Запускаем 1С: Предприятие в режиме отладки. В панели навигации разделов *Бухгалтерия* и *Учет материалов* появилась команда **Приходные накладные** для открытия списка приходных накладных, а также в панели действий раздела *Учет материалов* появилась команда **Приходная накладная** для создания новых документов этого вида.

Пока в нашей базе данных нет ни одного документа *Приходная накладная*, поэтому выполняем команду *Приходная накладная* в панели действий раздела *Учет материалов* и создаем новую приходную накладную. Откроется основная форма документа.

Система автоматически подставляет текущую дату создания документа и нулевое время, так как документ еще не проведен. Поле *Номер* не заполнено, но система сама его сгенерирует, так как свойство *Автонумерация* (на вкладке *Нумерация* в свойствах документа) включено по умолчанию. Поле *Склад* уже заполнено значением *Основной*, как было задано в свойствах реквизита. Заполняем табличную часть приходной накладной материалами для ремонта телевизоров как показано на рис. 2.3.

N	Материал	Количество	Цена	Сумма
1	Строчный трансформатор GoldStar	10,00	270,00	2 700,00
2	Строчный трансформатор Samsung	10,00	600,00	6 000,00
3	Транзистор Philips 2N2369	10,00	3,00	30,00

Рис. 2.3. Заполнение табличной части *Приходной накладной*

При нажатии кнопки выбора в поле *Материал* открывается форма для выбора элементов справочника *Номенклатура*, так как этот реквизит имеет ссылочный тип данных и ссылается на справочник *Номенклатура*. Также можно начать набирать название материала в поле *Материал* и система выдаст всплывающий список подходящих под описание материалов. После заполнения необходимо нажать *Провести и закрыть*.

Аналогичным способом создаем второй документ, который будет приходовать на *Основной* склад материалы для установки стиральных машин (рис. 2.4).

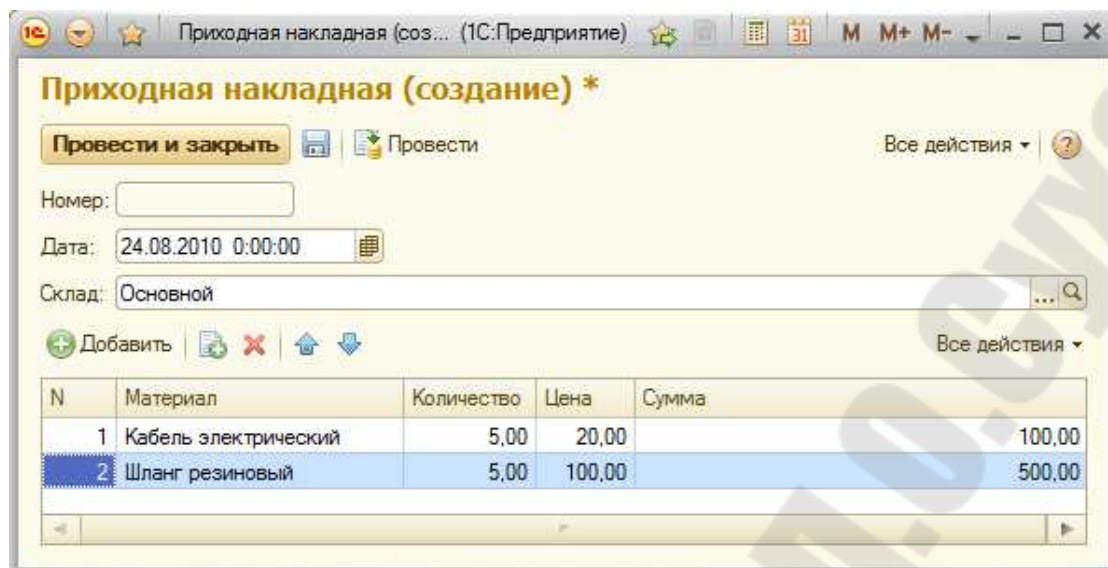


Рис. 2.4. Создание второго документа *Приходная накладная*

Посмотреть список созданных документов можно, выполнив команду **Приходные накладные** в навигационной панели.

**Шаг 4.** Настройка автоматического пересчета суммы в строках документа. Для удобства работы необходимо, чтобы сумма автоматически вычислялась каждый раз при изменении цены или количества материалов. Это можно сделать, создав свою форму документа и воспользовавшись возможностями встроенного языка.

В режиме конфигуратора открываем окно редактирования документа *Приходная Накладная*, закладка **Формы**. Для создания формы необходимо нажать кнопку **Добавить** или значок лупы около поля *Документы*. Откроется конструктор форм. Там ничего не меняем и сразу нажимаем **Готово**. Откроется окно редактора форм (рис. 2.5). Разработчик не может нарисовать форму, а только может указать из каких элементов будет состоять форма.

Элементы формы в верхнем левом окне редактора форм образуют иерархическую структуру. Чем выше в списке элемент, тем выше и левее на форме он будет располагаться (красное). От изменения порядка в дереве объектов изменится порядок следования на форме (желтое).

**Обработчик событий.** У системы существуют события, которые связаны с различными моментами ее функционирования. Используя встроенный язык, разработчик может внедриться в эти события и описать собственный алгоритм того, что должно происходить при наступлении события.

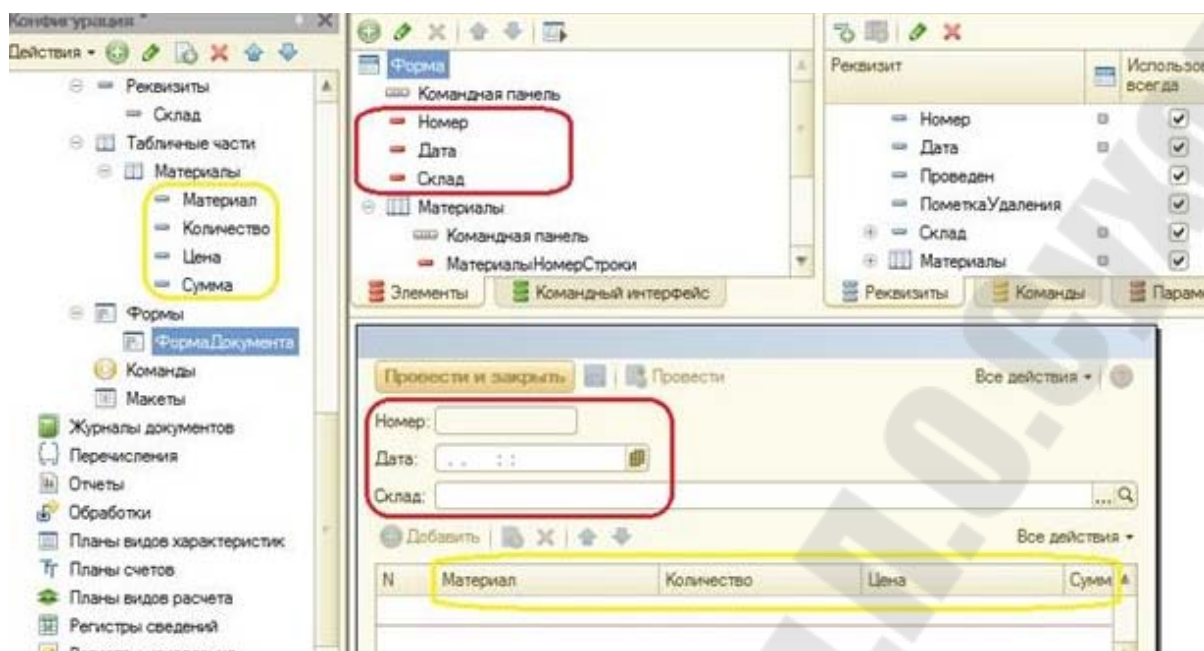


Рис. 2.5. Окно редактора форм

Дважды щелкаем на элементе формы **МатериалыКоличество** (или контекстное меню – Свойства). Справа возникнет окно свойств, прокручиваем его до конца вниз, находим поле **ПриИзменении** и нажимаем на значок лупы рядом с ним (рис. 2.6).

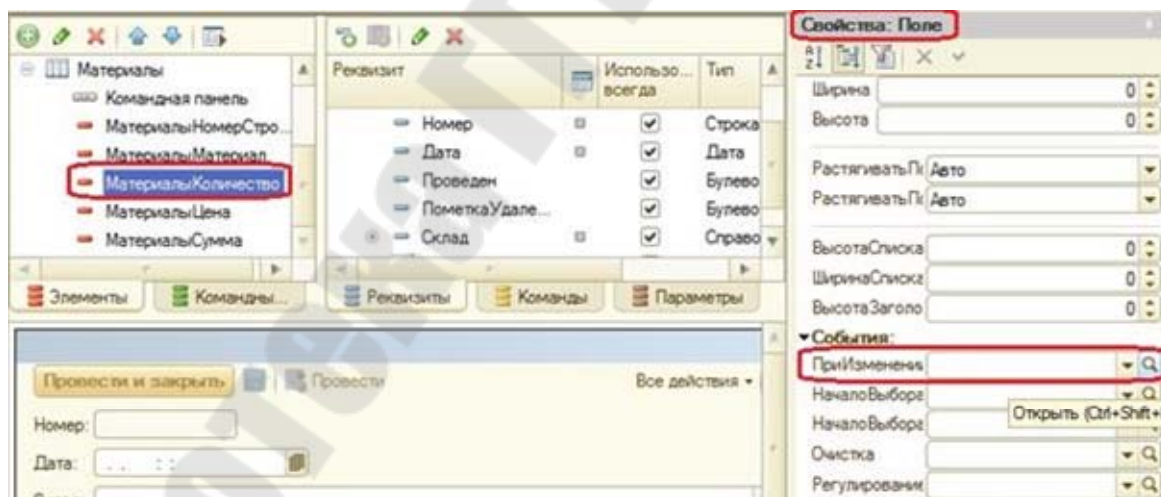


Рис. 2.6. Окно редактора форм

Система откроет закладку **Модуль редактора формы**. Модуль – это хранилище для текста программы на встроенном языке. В данном случае это модуль формы. В модуль формы в процедуру **МатериалыКоличествоПриИзменении()** необходимо добавить следующий текст:

```
СтрокаТабличнойЧасти = Элементы.Материалы.ТекущиеДанные;  
СтрокаТабличнойЧасти.Сумма = СтрокаТабличнойЧасти.Количество * СтрокаТаблич-  
нойЧасти.Цена;
```

Запускаем 1С: Предприятие в режиме отладки, заходим в любой документ *Приходная накладная* и изменяем количество товара – сумма автоматически пересчитывается. Закрываем приложение и возвращаемся в конфигурактор.

**Шаг 5.** Создание процедуры для обработки нескольких событий. Теперь при изменении количества в любой строке документа *Приходная накладная*, сумма в этой в этой строке пересчитывается автоматически. Теперь следует сделать тоже самое для поля *Цена*. Также автозаполнение может понадобиться в других документах. Поэтому лучше будет поместить расчет суммы в некоторое общедоступное место, чтобы разные документы могли использовать этот алгоритм. Для описания таких мест служат объекты *Общий модуль*, расположенные в ветке *Общие/Общие модули*. Процедуры и функции, содержащиеся в этих модулях, могут быть доступны для любых объектов конфигурации.

Создадим общий модуль *РаботаСДокументами*. В его свойствах необходимо установить флажок *Клиент*, а флажок *Сервер* необходимо снять (это означает, что модуль будет скомпилирован в тонком клиенте и в веб-клиенте, а не на сервере) (рис. 2.7).

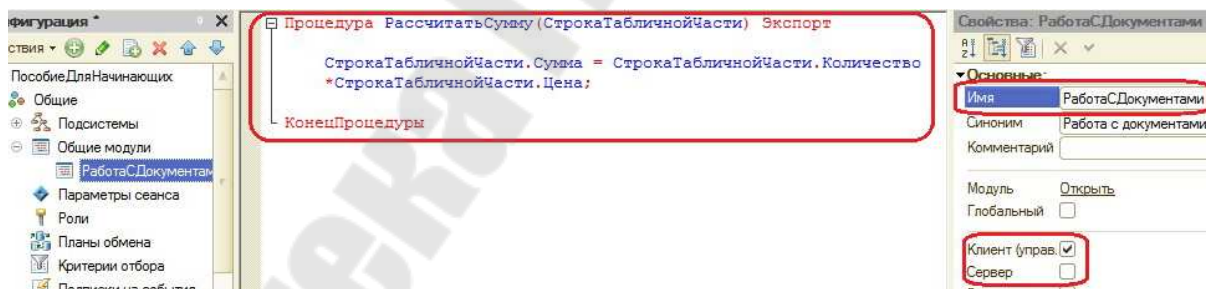


Рис. 2.7. Окно редактора форм

В созданный модуль переносим созданную процедуру расчета суммы (рис. 2.7). В документе оставляем вызов этой процедуры из общего модуля:

```
Процедура РассчитатьСумму(СтрокаТабличнойЧасти) Экспорт  
СтрокаТабличнойЧасти.Сумма = СтрокаТабличнойЧасти.Количество  
*СтрокаТабличнойЧасти.Цена;  
КонецПроцедуры
```

В модуле формы документа *Приходная накладная* проведем изменение второй строчки процедуры на следующую (рис. 2.8):

```
РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);
```

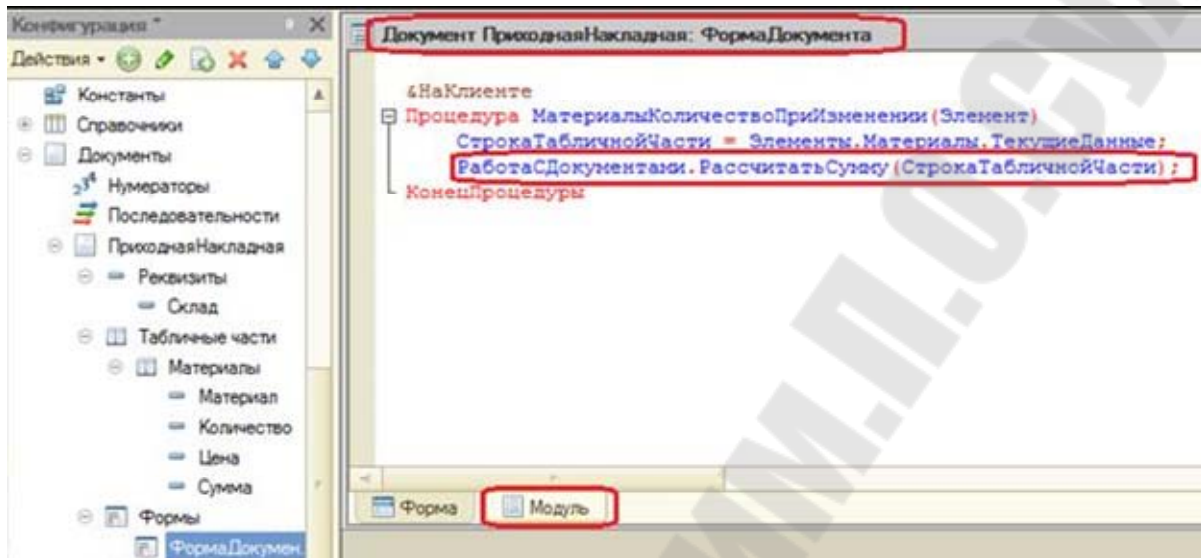


Рис. 2.8. Корректировка в модуле формы *Приходной накладной*

Таким образом будет происходить вызов процедуры *РассчитатьСумму()* из общего модуля *РаботаСДокументами* и передача ей в качестве параметра текущей строки табличной части.

Осталось для поля *Цена* установить такой же обработчик. Создайте обработчик события *ПриИзменении* для поля табличной части *МатериалыЦена* также, как это было проделано для поля *МатериалыКоличество*, повторите в нем вызов процедуры *РассчитатьСумму()* из общего модуля.

Необходимо нажать внизу вкладку *Форма* (рядом с *Модуль*), дважды щелкнуть в левом верхнем окне на *МатериалыЦена*, прокрутить окно свойств, нажать на значок лупы в поле *ПриИзменении* и в выделенную область вписать (рис. 2.9):

```
СтрокаТабличнойЧасти = Элементы.Материалы.ТекущиеДанные;  
РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);
```



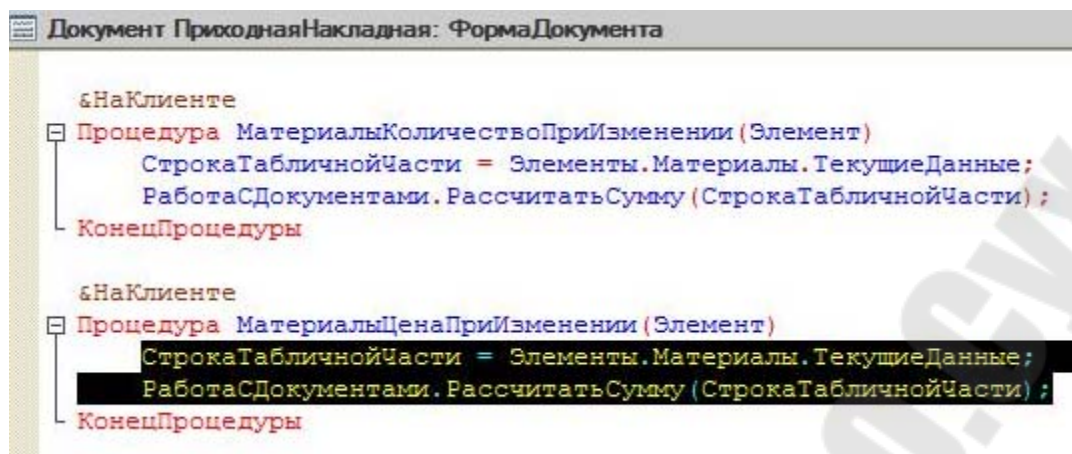


Рис. 2.9. Процедура для обработки изменения цены

**Шаг 6.** Создание документа *Оказание Услуг*. Создадим новый документ с именем *ОказаниеУслуги*. Представление объекта не задаем – будет использоваться синоним, представление списка задаем как *Оказание услуг*. На вкладке **Подсистемы** отмечаем *ОказаниеУслуг* и *Бухгалтерия*.

На вкладке **Данные** создаем реквизиты (рис. 2.10):

*склад*, тип *СправочникСсылка.Склады*, значение заполнения – *Основной*;

*клиент*, тип *СправочникСсылка.Клиенты*, проверка заполнения – *Выдавать ошибку*;

*мастер*, тип *СправочникСсылка.Сотрудники*, Проверка заполнения – *Выдавать ошибку*.

Создаем табличную часть документа с именем *ПереченьНоменклатуры* с реквизитами:

*номенклатура*, тип *СправочникСсылка.Номенклатура*;

*количество*, тип *Число*, длина 15, точность 3, неотрицательное;

*цена*, тип *Число*, длина 15, точность 2, неотрицательное;

*сумма*, тип *Число*, длина 15, точность 2, неотрицательное.

Установим для табличной части в целом и для каждого ее реквизита свойство **Проверка заполнения** в *Выдавать ошибку*.

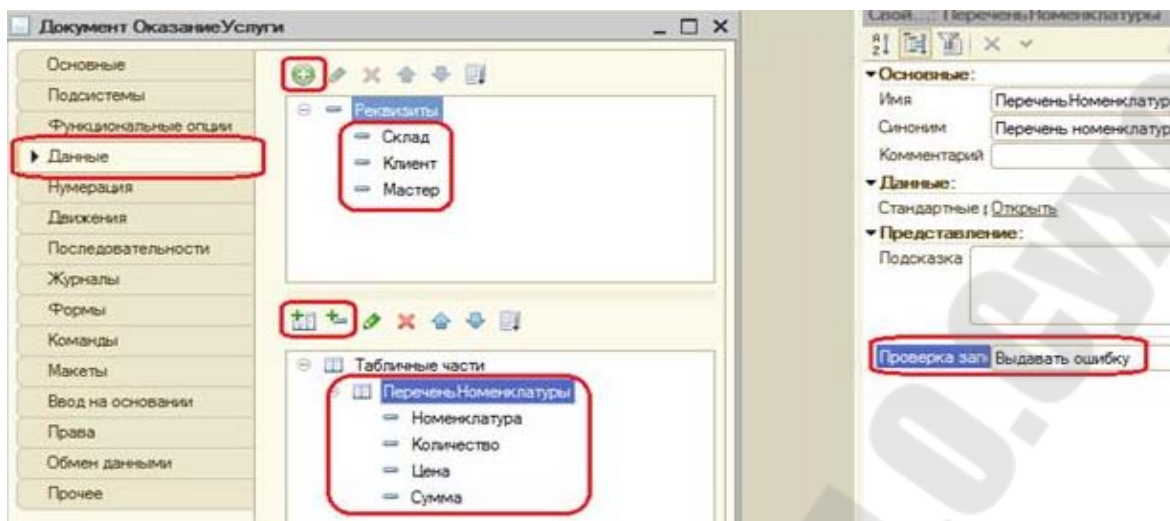


Рис. 2.10. Создание документа ОказаниеУслуги

Далее переходим на вкладку **Формы** и создаем основную форму документа. Для поля *ПереченьНоменклатурыКоличество* создайте обработчик события **ПриИзменении**, в котором будем вызывать процедуру *РассчитатьСумму()* из общего модуля **РаботаСДокументами** (рис. 2.11).

Откроется модуль формы с шаблоном обработчика события **ПереченьНоменклатурыКоличествоПриИзменении**.

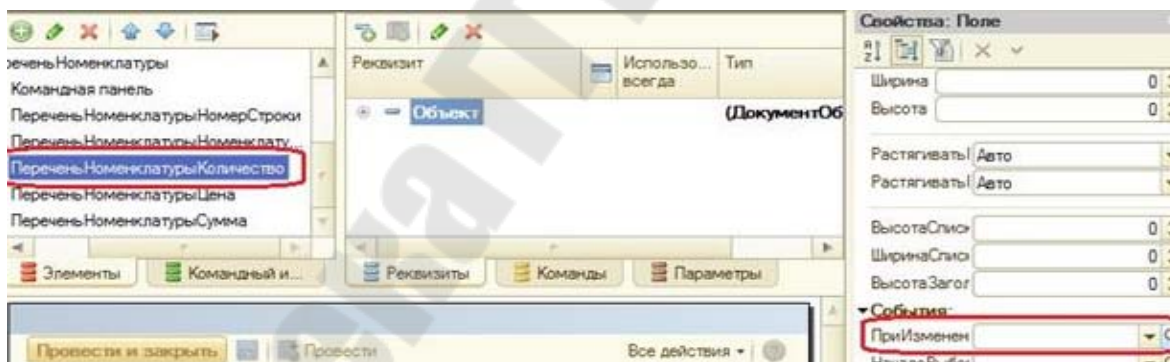


Рис. 2.11. Создание основной формы документа ОказаниеУслуги

Открытый обработчик пока не заполняем, а переходим в окно элементов формы на закладку **Форма** и аналогично создаем обработчик события **ПереченьНоменклатурыЦенаПриИзменении** для поля *ПереченьНоменклатурыЦена* (рис. 2.12).

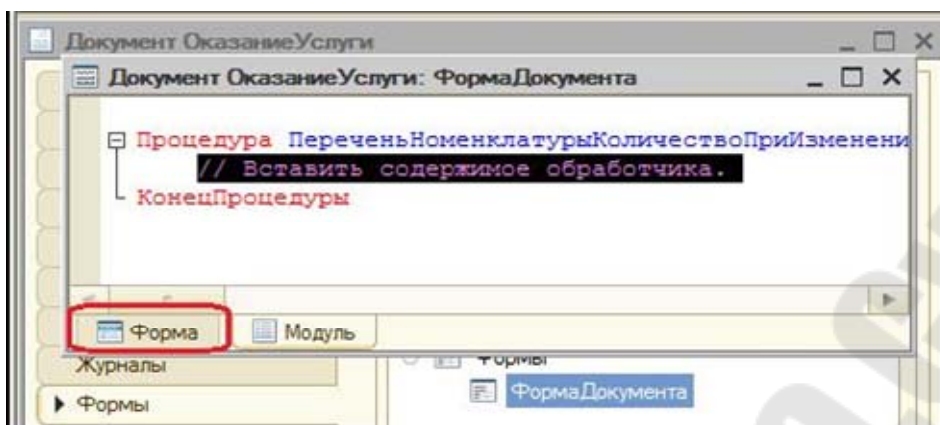


Рис. 2.12. Открытие формы документа

Далее модуль формы документа *ОказаниеУслуги* заполняем следующим образом (рис. 2.13):

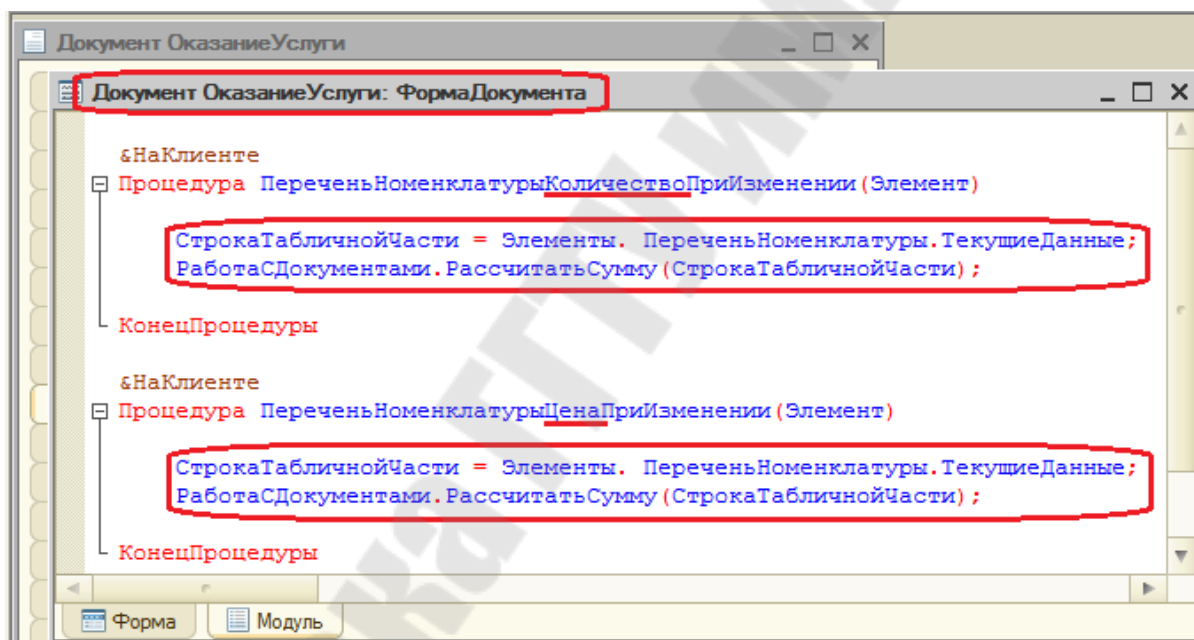


Рис. 2.13. Процедуры модуля документа *ОказаниеУслуги*

Отредактируем командный интерфейс, чтобы в подсистеме *Оказание услуг* была доступна команда создания новых документов (см. выше). Запустим 1С: Предприятие в режиме отладки.

В **панели действий** раздела *Оказание услуг* создадим документ и заполни его в соответствии с рис. 2.14.

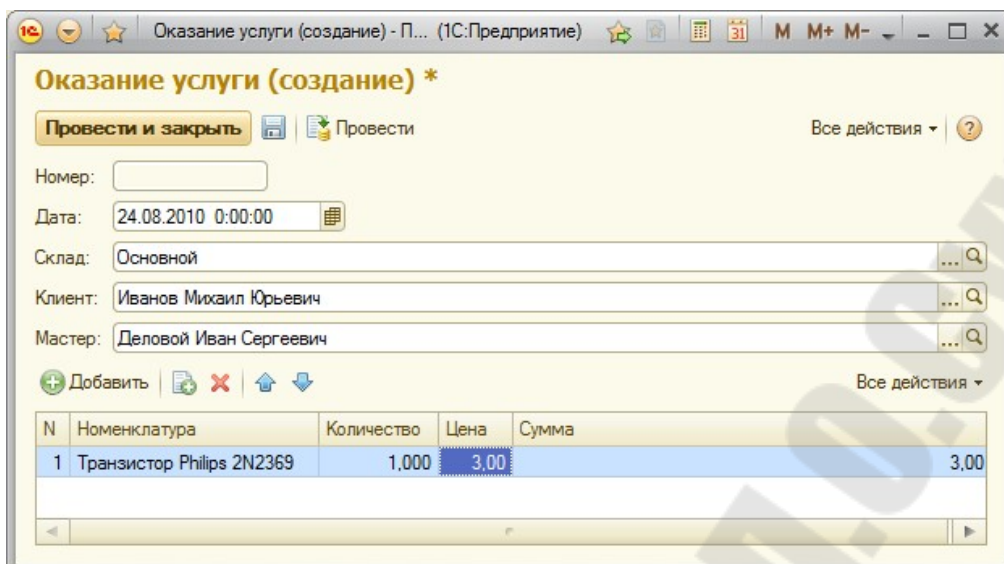


Рис. 2.14. Создание документа ОказаниеУслуги

### Контрольные вопросы к заданию 2.1

1. Какими характерными особенностями обладает документ?
2. Для чего предназначены реквизиты и табличные части документа?
3. Что такое проведение документа?
4. Как создать объект Документ и описать его структуру?
5. Как создать собственную форму документа?
6. Что такое обработчик события и как его создать?
7. Что такое модуль и для чего он нужен?
8. Зачем нужны общие модули?

**Задание 2.2. Создание регистра накопления.** В соответствии с приведенным описанием создать регистр накопления *Остатки материалов* и организовать проведение документов *Приходная накладная* и *Оказание услуги*.

### Теоретическая справка

Использование регистров в системе 1С: Предприятие позволяет накапливать в них данные, поставляемые различными документами, легко создавать нужные отчеты или использовать эти данные в алгоритмах работы конфигурации. В системе существуют несколько видов регистров.

**Регистр накопления** предназначен для описания структуры накопления данных. Эти данные будут храниться в таблице в виде отдельных записей, каждая из которых имеет одинаковую, заданную в конфигураторе структуру. Отличительная особенность регистра накопления – не предназначен для редактирования пользователем.

**Основное назначение регистра накопления** – накопление числовой информации в разрезе нескольких измерений, которые описываются разработчиком в свойствах регистра накопления. Виды числовой информации, накапливаемой регистром накопления, называются **ресурсами**.

Например, регистр накопления может накапливать информацию о количестве и сумме товаров на складах. В этом случае он будет иметь **измерения** *Товар* и *Склад* и **ресурсы** *Количество* и *Сумма*.

Изменение состояние регистра накопления происходит, как правило, при проведении документа и заключается в добавлении в него нескольких записей. Каждая запись содержит значения измерений, значения приращений ресурсов, ссылку на документ, который вызвал эти изменения (регистратор) и направление приращения (приход или расход). Такой набор записей называется **движениями регистра накопления**.

Каждому движению регистра накопления всегда должен соответствовать **регистратор** – объект информационной базы (обычно документ), который произвел эти движения.

Кроме этого регистр накопления может хранить дополнительную информацию, описывающую каждое движение – **реквизиты**.

## **Методические рекомендации по выполнению задания**

**Шаг 1.** Добавление регистра накопления. Открываем конфигуратор и добавляем новый объект конфигурации *Регистр накопления* с именем *ОстаткиМатериалов*. Расширенное представление списка – Движение по регистру Остатки материалов. Этот заголовок будет отображаться в окне списка записей регистра. На вкладке **Подсистемы** отмечаем *УчетМатериалов*, *ОказаниеУслуг* и *Бухгалтерия*. Переходим на закладку **Данные** и создаем *измерения* регистра (рис. 2.15):

- материал, тип СправочникСсылка.Номенклатура;
- склад, тип СправочникСсылка.Склады.

Затем создаем *ресурс* *Количество* с длиной 15 и точностью 3.

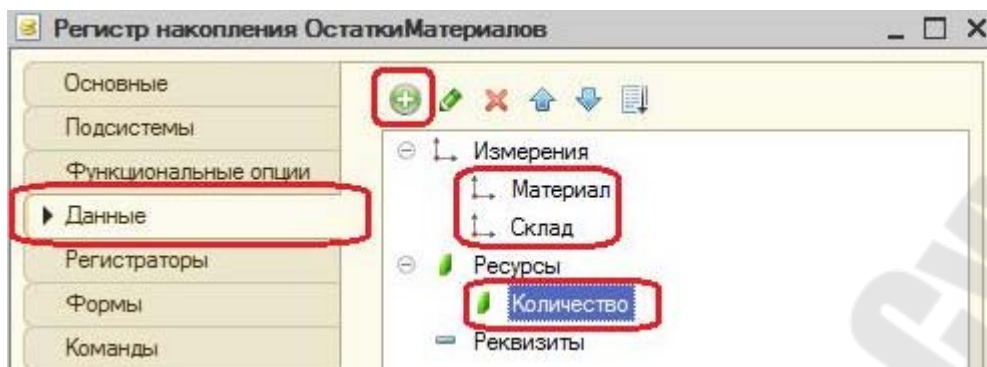


Рис. 2.15. Создание регистра накопления *ОстаткиМатериалов*

Шаг 2. Настройка движения в регистре накопления. Открываем окно редактирования объекта Документ *ПриходнаяНакладная* на закладке *Движения*. Раскройте список *Регистры накопления* и выделите *ОстаткиМатериалов*. Затем нажмите кнопку *Конструктор движений* (рис. 2.16).

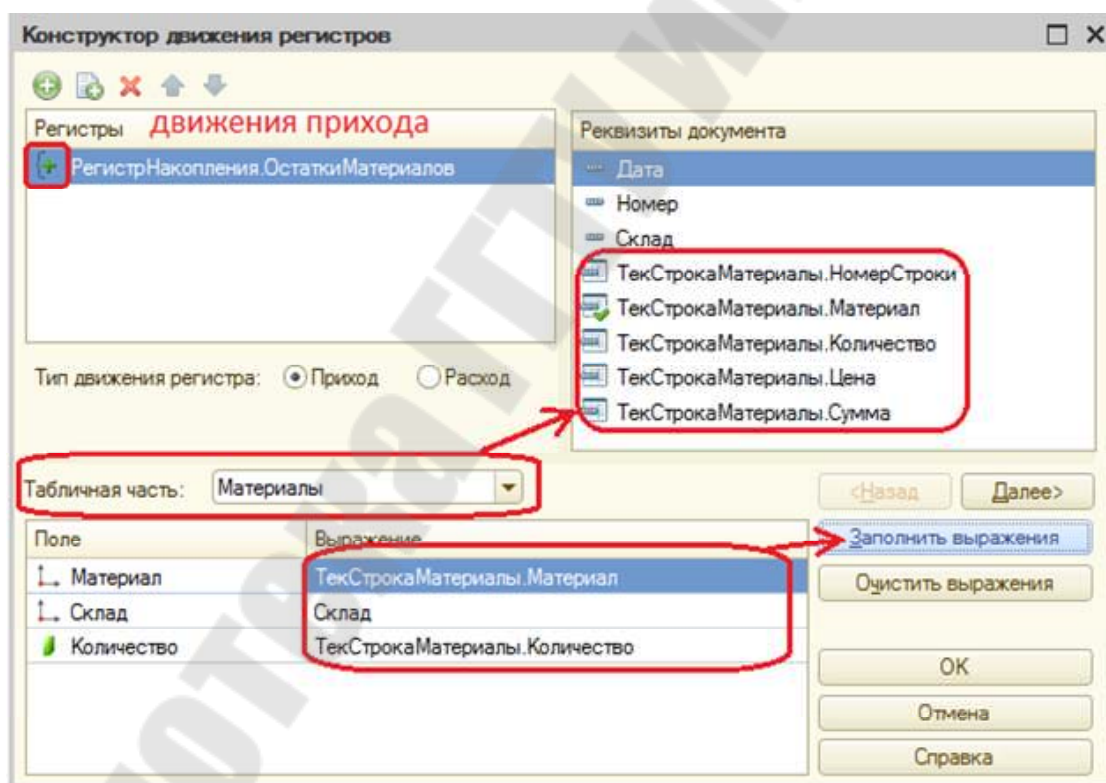


Рис. 2.16. Конструктор движения регистров

В таблице *Поле – Выражение* должны быть заданы формулы вычисления значений измерений и ресурсов регистра при записи движений.

В поле выбора **Табличная часть** выбираем табличную часть документа – **Материалы**. Нажимаем кнопку **Заполнить выражения**.

Нажимаем **ОК**. Можно просмотреть текст алгоритма, сформированного конструктором в модуле документа **Приходная Накладная**.

Открываем окно редактирования объекта Регистр накопления **Остатки Материалов**, переходим на закладку **Регистраторы**. В списке документов будет виден отмеченный документ **Приходная Накладная**, так как было задано в модуле этого документа формирование движений в регистре **Остатки Материалов** (рис. 2.17).

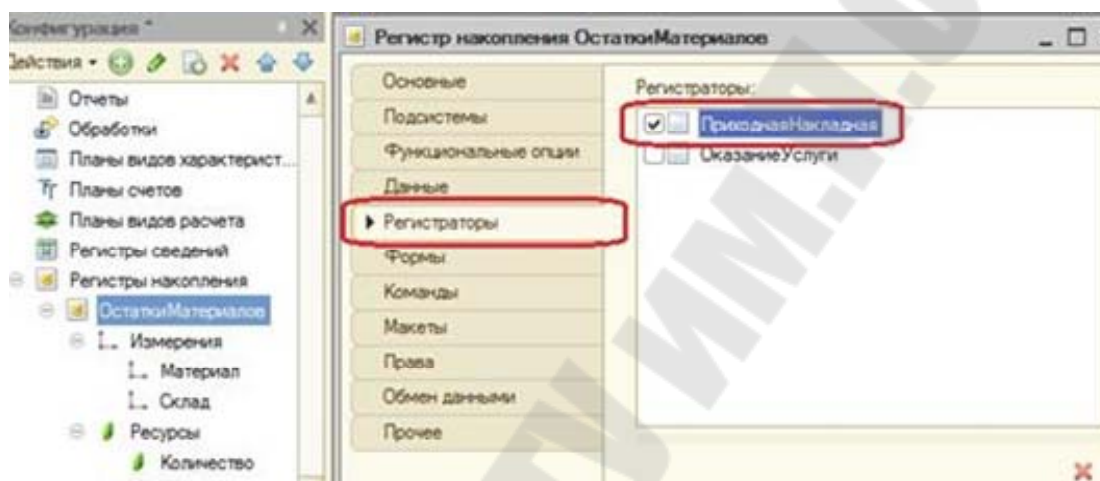


Рис. 2.17. Регистраторы регистра накопления **Остатки Материалов**

Отредактируем командный интерфейс, чтобы в подсистемах **Бухгалтерия**, **Оказание Услуг** и **Учет Материалов** была доступна ссылка для просмотра записей нашего регистра накопления.

Дело в том, что команды открытия регистров также добавляются в панель навигации подсистем, но по умолчанию невидимы, в отличие от команд открытия справочников и документов.

В группе **Панель навигации.Обычное** включаем видимость у команды **Остатки материалов** и перетаскиваем ее в группу **Панель навигации.См. также**. Аналогичные действия повторяем в указанных подсистемах (рис. 2.18).

Запускаем 1С: Предприятие в режиме отладки. В открывшемся окне видно, что в панели навигации в группе **См. также** разделов **Бухгалтерия**, **Оказание услуг** и **Учет материалов** появилась команда для открытия списка регистра **Остатки материалов** (рис. 2.19).

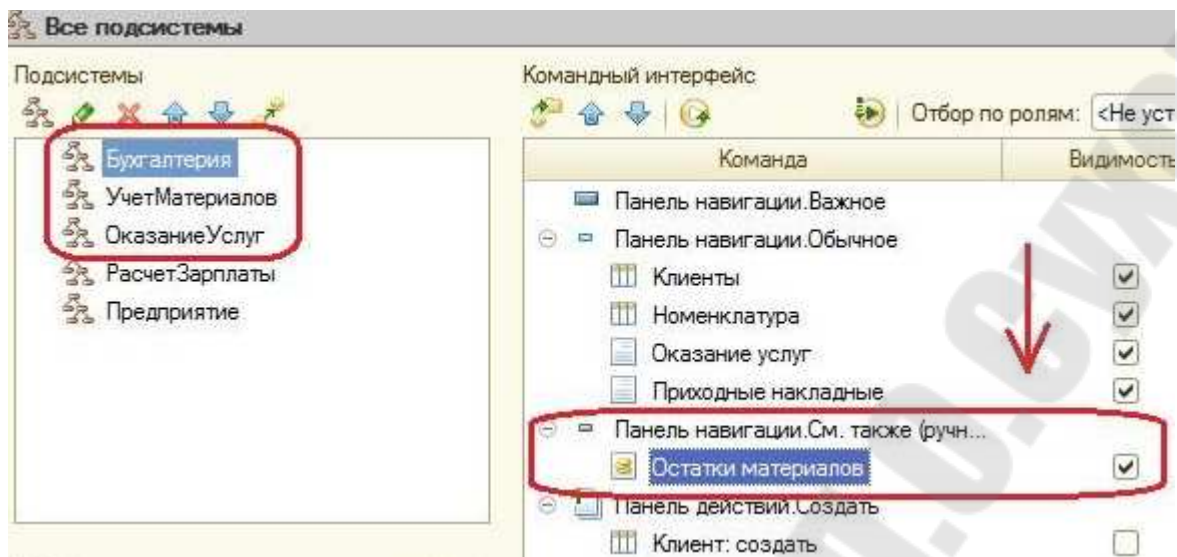


Рис. 2.18. Настройка командного интерфейса

Чтобы проследить связь между проведением документа и накоплением информации в регистре, открываем список приходных накладных. Открываем Приходную накладную № 1 и нажимаем **Провести и закрыть**, т. е. препроводим ее. То же самое повторяем для второй накладной. Перепровести можно и не открывая документа – выбрать документы, **Все действия/Провести**. У документов в списке изменится время.

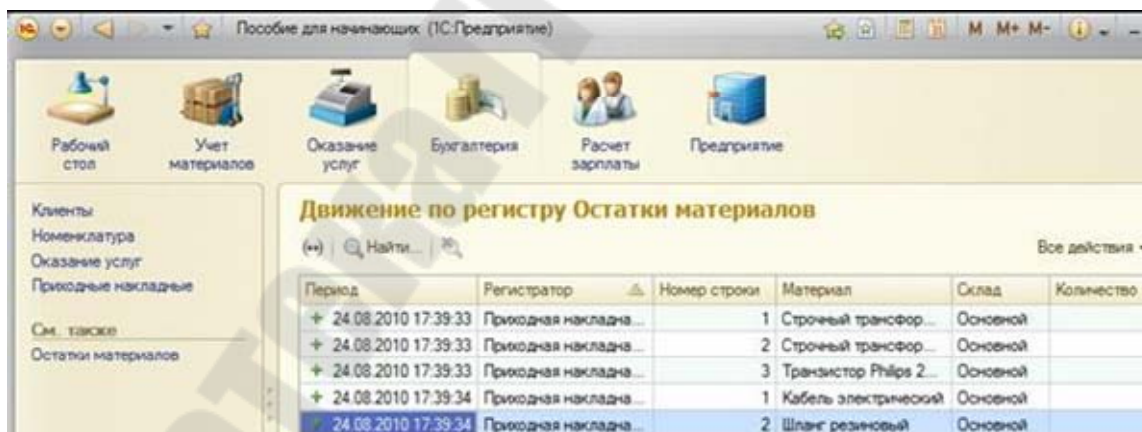


Рис. 2.19. Движение по регистру *Остатки материалов*

Открываем регистр *Остатки материалов* через навигационную панель. Видим, что при проведении приходных накладных появляются соответствующие записи в регистре накопления *Остатки материалов*. Добавилось 5 записей – первые три после проведения первого документа из трех строк, последние две после проведения второго из двух строк в табличной части.



Все поля регистра заполнились данными документов так, как было задано в обработчике проведения документа *ПриходнаяНакладная*. Пиктограмма со знаком «+» слева от каждой записи указывает на тип движения – *Приход*.

**Шаг 3.** Настройка команды перехода к движениям в форме документа. При реальной работе записей в регистре *ОстаткиМатериалов* будет много и будет трудно понять, какие записи относятся к определенному документу. Поэтому наряду с общим списком регистра хотелось бы иметь возможность вызывать из формы документа список регистра, в котором показаны движения, произведенные только конкретным документом.

Для такой настройки необходимо вернуться в Конфигуратор и открыть форму документа *ПриходнаяНакладная*. На закладке **Командный интерфейс** в разделе **Панель навигации** раскрыть группу **Перейти** и установить видимость команды **Остатки материалов (Объект.Ссылка)** (рис. 2.20).

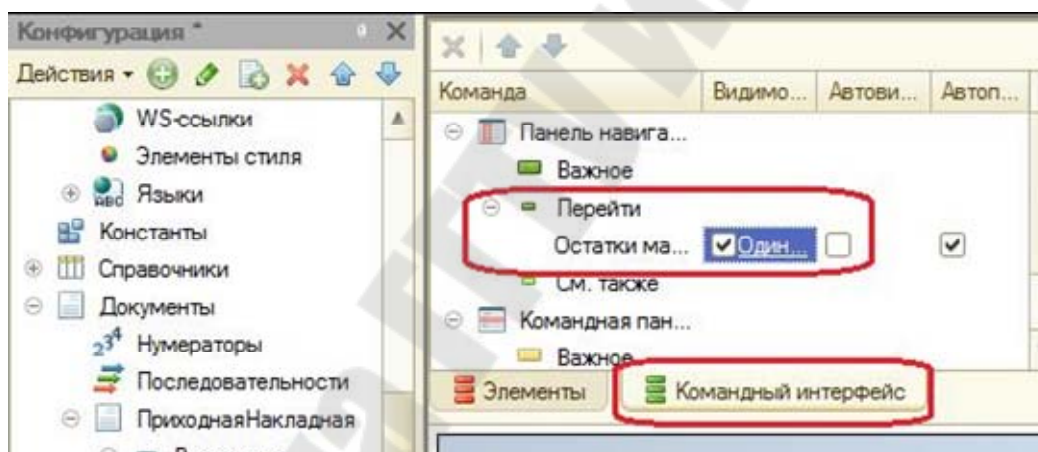


Рис. 2.20. Настройка просмотра движения документов

Запустим 1С: Предприятие в режиме отладки, откроем *Приходную накладную № 2*. В форме документа появилась панель навигации, в которой можно перейти к списку записей регистра *Остатки материалов*, связанному с документом и обратно к содержимому документа.

**Шаг 4.** Движение документа *Оказание услуги*. Аналогичным образом создаем движения документа *ОказаниеУслуги*.

Открываем окно редактирования объекта Документ *Оказание Услуги*, переходим на вкладку **Движения**, в списке регистров отмечаем *ОстаткиМатериалов*, нажимаем кнопку **Конструктор движений** (рис. 2.21).

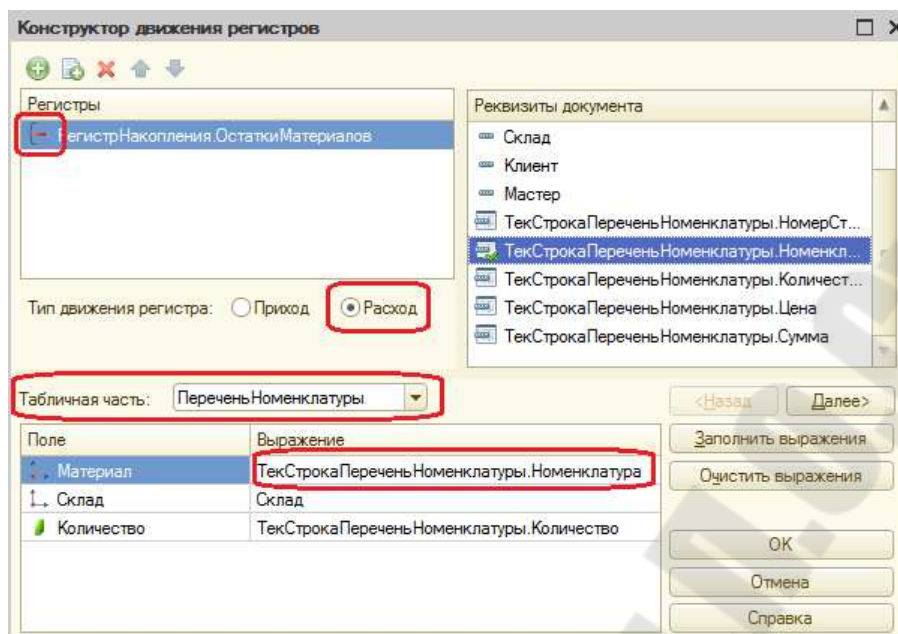


Рис. 2.21. Настройка движения документа *Оказание услуги*

Изменяем тип движения регистра на **Расход**, так как документ *ОказаниеУслуги* должен расходовать материалы в процессе оказания услуг. Пиктограмма слева от названия регистра изменится на знак «←→».

В поле выбора **Табличная часть** выбираем **ПереченьНоменклатуры**. Нажимаем кнопку **Заполнить выражения**. В нижнем окне сформируется соответствие полей (измерений и ресурсов) регистра и выражений для их расчета, но поле **Материал** не заполнится. Так происходит, потому что имя поля табличной части – **Номенклатура** не совпадает с именем измерения регистра – **Материал**. Если оставить как есть, то движение фиксироваться не будет.

Выделяем поле регистра **Материал** и в окне реквизиты документа дважды щелкаем по строке **ТекСтрокаПереченьНоменклатуры.Номенклатура**. Таким образом, номенклатура для движений регистра накопления будет выбираться из табличной части документа.

Нажимаем **ОК**. Конструктор создал обработчик события **ОбработкаПроведения** объекта Документ *ОказаниеУслуги* и поместил его в модуль объекта.

Отредактируем командный интерфейс формы документа *ОказаниеУслуги*, чтобы в панели навигации формы иметь возможность переходить к списку записей регистра *Остатки материалов*, связанному с документом.

Закрываем окно модуля, переходим на вкладку **формы документа ОказаниеУслуги**, дважды щелкните на названии формы (рис. 2.22).

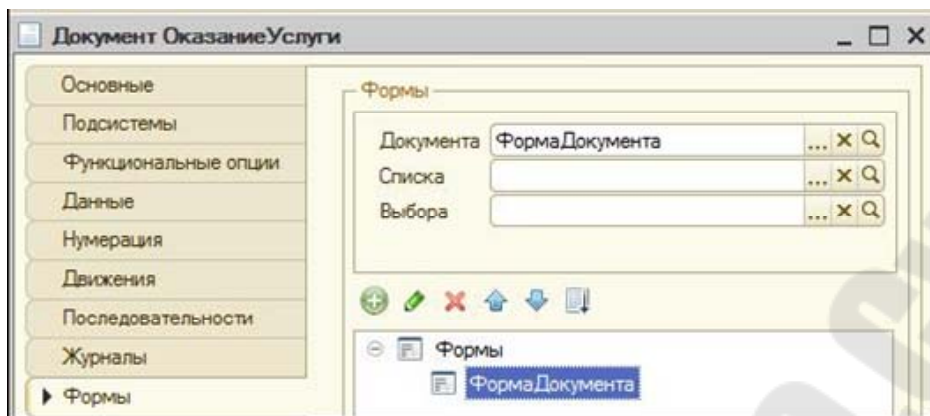


Рис. 2.22. Переход к форме документа *Оказание услуги*

Далее выберите вкладку **Командный интерфейс**, раскрываем группу **Панель навигации/Перейти** и устанавливаем видимость для команды открытия регистра накопления *Остатки материалов*.

Далее запускаем 1С: Предприятие в режиме отладки. В разделе *Оказание услуг* открываем документ *Оказание услуги № 1* и перепроводим его (рис. 2.23).

Следует отметить, что сформированные таким образом движения этого документа будут не совсем правильными. Дело в том, что в документе *Оказание услуги* могут содержаться не только расходующие материалы, но и услуги. Поэтому в регистр *Остатки материалов* будут попадать записи и о расходующих услугах. Решение данной проблемы будет описано в последующих лабораторных работах.

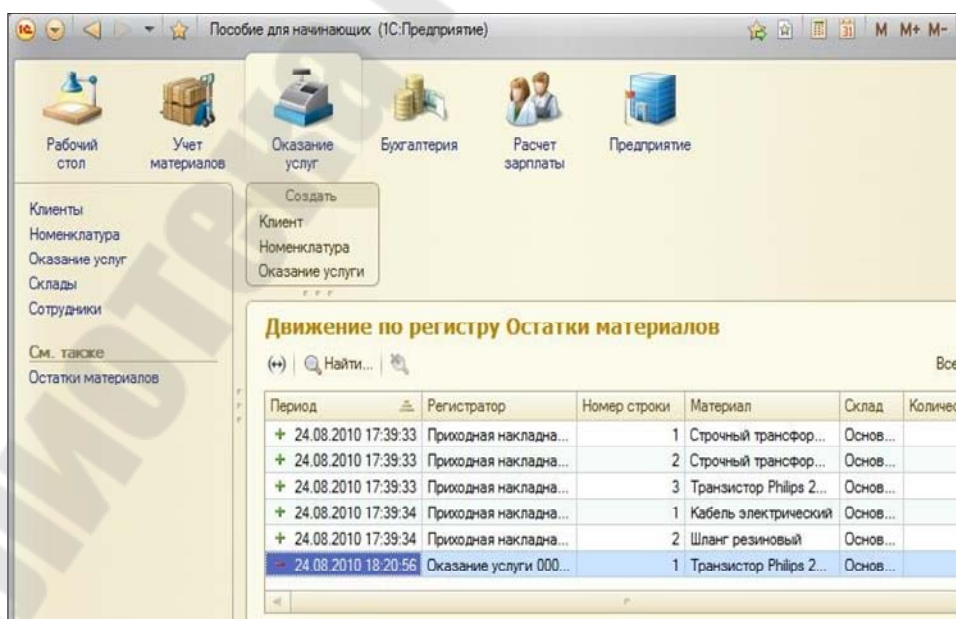


Рис. 2.23. Регистр накопления после перепроведения документа

*Контрольные вопросы к заданию 2.2*

1. Какую функцию выполняет Регистр накопления?
2. Для чего нужны измерения регистра, ресурсы и реквизиты?
3. Что такое движение регистра и что такое регистратор?
4. Как создать движения документа с помощью конструктора движений?

## ЛАБОРАТОРНАЯ РАБОТА 3 ПРОСТОЙ ОТЧЕТ. МАКЕТЫ И ФОРМЫ

**Задание 3.1. Создание документов.** В соответствии с приведенным описанием создать и настроить видимость отчета *Материалы*.

### Теоретическая справка

Объект конфигурации *Отчет* предназначен для описания алгоритмов, при помощи которых пользователь сможет получать необходимые ему данные. Алгоритм формирования выходных данных описывается при помощи визуальных средств или с использованием встроенного языка. В реальной жизни объектам *Отчет* соответствуют всевозможные таблицы выходных, сводных данных, диаграммы и т. д.

### Методические рекомендации по выполнению задания

Шаг 1. Добавление отчета. Приступим к созданию отчета, который будет показывать нам приход, расход и остатки материалов.

Добавим новый объект *Отчет* с именем *Материалы*, нажимаем **Открыть схему компоновки данных** – основу для построения любого отчета (рис. 3.1).

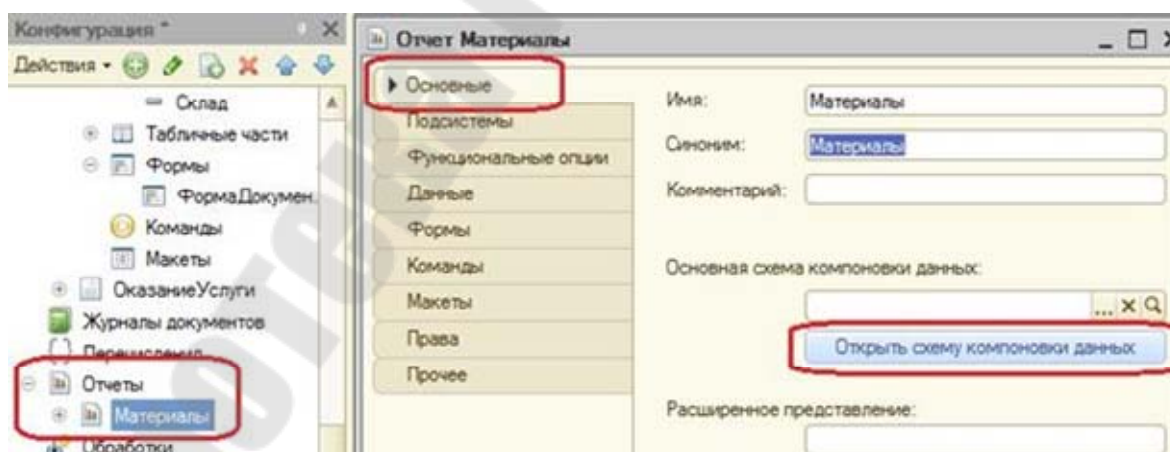


Рис. 3.1. Свойства отчета Материалы

Откроется конструктор схемы компоновки данных. Добавим новый набор данных – *запрос*. Чтобы создать текст запроса, запустим конструктор запроса с помощью одноименной кнопки (рис. 3.2).

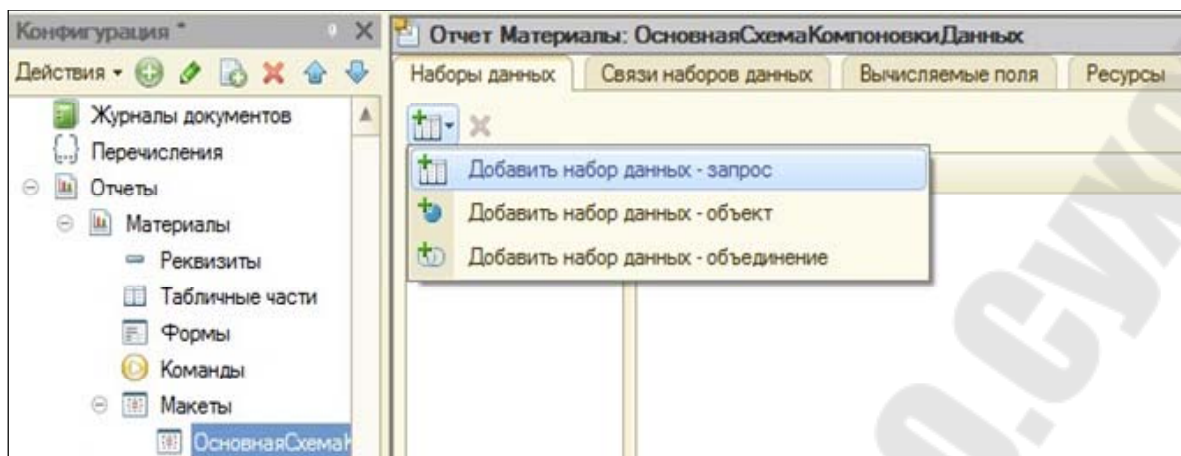


Рис. 3.2. Открытие конструктора запроса

**Конструктор запроса** – инструмент, позволяющий визуально создать запрос даже не знакомому с языком запросов пользователю.

В списке *База данных* представлены таблицы для создания запроса. На основе их данных можно построить отчет. Так как необходимо видеть остатки материалов и информацию об их поступлении и расходовании, будет использовать виртуальную таблицу **Остатки-Материалов.ОстаткиИОбороты**. Раскрываем его. Она содержит измерения регистра *ОстаткиМатериалов – Материал, Склад*, начальные и конечные остатки, значения прихода и расхода, обороты для всех ресурсов регистра. Выберите двойным щелчком из окна *База данных* поля таблицы: *Склад, Материал, КоличествоНачальныйОстаток, КоличествоПриход, КоличествоРасход, КоличествоКонечныйОстаток*. Нажимаем **ОК** и возвращаемся в конструктор компоновки схемы данных.

Текст запроса, который был создан с помощью конструктора, платформа поместит в поле *Запрос* (рис. 3.3). Это поле представляет собой текстовый редактор, в котором можно отредактировать созданный запрос.

Таким образом, будут извлекаться данные для отчета, но пока не создано стандартных настроек отображения.

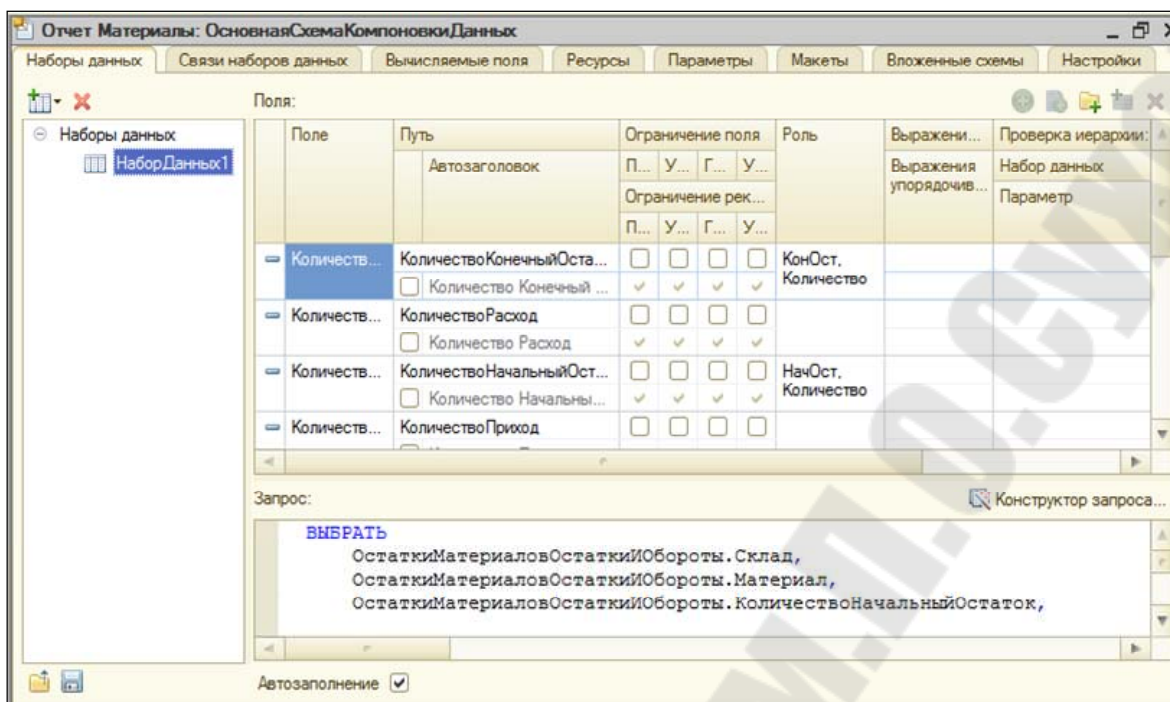


Рис. 3.3. Запрос на выборку данных для отчета

Шаг 2. Настройка отчета. Переходим на закладку **Настройки**. Выделяем элемент **Отчет** и нажимаем кнопку **Добавить новую группировку** (или через контекстное меню) (рис. 3.4).



Рис. 3.4. Добавление группировки

Таким образом определено, что в отчете будут выводиться детальные записи из информационной базы – записи, получаемые в результате запроса без итогов.

В структуре отчета появится группировка **Детальные записи**.

Теперь настроим поля, которые будут выводиться в результат отчета. Для этого необходимо перейти в нижнем окне настроек на вкладку **Выбранные поля** и перенести мышью из списка доступных полей: **Склад**, **Материал**, **КоличествоНачальныйОстаток**, **КоличествоПриход**, **КоличествоРасход**, **КоличествоКонечныйОстаток** (рис. 3.5, а).

Затем переходим на вкладку **Параметры** и указываем, что параметры отчета *Начало периода* и *Конец периода* будут включены в состав пользовательских настроек и эти настройки будут находиться в форме отчета (рис. 3.5, б).

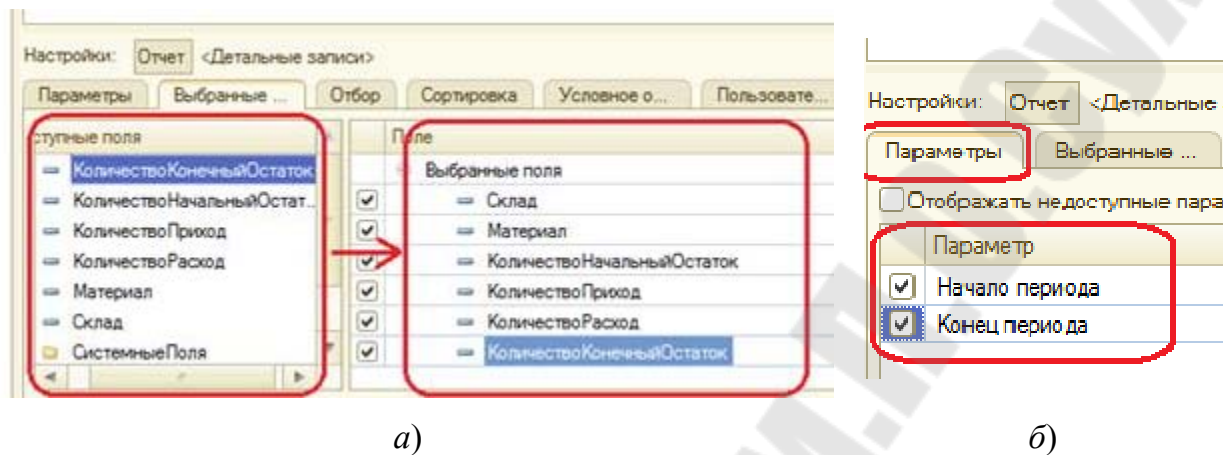



Рис. 3.5. Настройка структуры отчета

Затем выделяем каждый из параметров, нажимаем кнопку **Свойства элемента пользовательских настроек**  и отмечаем флажок **Включать в пользовательские настройки**.

Таким образом, перед формированием отчета пользователь сможет задать отчетный период.

**Шаг 3.** Настройка видимости. В завершении определим, в каких подсистемах будет отображаться отчет. Закрываем конструктор схемы компоновки данных и переходим на вкладку **Подсистемы**. Отмечаем *Бухгалтерия*, *УчетМатериалов*, *ОказаниеУслуг*. Теперь ссылка на отчет попадет в панель действий указанных разделов.

Запускаем 1С: Предприятие. В панели действий разделов *Бухгалтерия*, *Оказание услуг* и *Учет материалов* появилась новая группа команд для выполнения отчетов и в ней команда для формирования отчета *Материалы*. Выполним ее. Откроется сформированная системой форма отчета. Задаем даты начала (Начало этого месяца) и окончания (Начало этого дня) отчетного периода и нажимаем **Сформировать** (рис. 3.6).



Материал	Склад	Количество Начальный остаток	Количество Приход	Количество Расход	Количество Конечный остаток
Строчный трансформатор Samsung	Основной		10,000		10,000
Строчный трансформатор GoldStar	Основной		10,000		10,000
Транзистор Philips 2N2369	Основной		10,000		10,000

Рис. 3.6. Сформированный системой отчет *Материалы*

### Контрольные вопросы к заданию 3.1

1. Для чего предназначен объект конфигурации *Отчет*?
2. Как создать отчет с помощью конструктора схемы компоновки данных?
3. Как отобразить отчет в разделах прикладного решения?

### Задание 3.2. Создание и редактирование макетов и форм.

В соответствии с приведенным описанием отредактировать макет и форму документа *Оказание услуги*.

### Теоретическая справка

**Макет** предназначен для хранения различных форм представления данных, различных данных. Может содержать табличный или текстовый документ, двоичные данные, HTML- документ, графическую или географическую схему, схему компоновки данных или макет оформления этой схемы.

### Методические рекомендации по выполнению задания

**Шаг 1.** Создание печатной формы. Создадим печатную форму документа *ОказаниеУслуги*. Открываем в конфигураторе окно редактирования объекта *Документ ОказаниеУслуги*. Переходим на вкладку *Макеты* и запускаем *Конструктор печати* (рис. 3.7).

На первом шаге ничего менять не нужно. Нажимаем *Далее*. На втором шаге переносим все реквизиты документа из левой части в

правую – они будут отображены в шапке печатной формы. Нажимаем *Далее*. Также переносим все реквизиты в табличную часть. На четвертом шаге ничего не переносим – подвал (нижняя часть формы) не будет использоваться, нажимаем *Далее*. Нажимаем *ОК*.

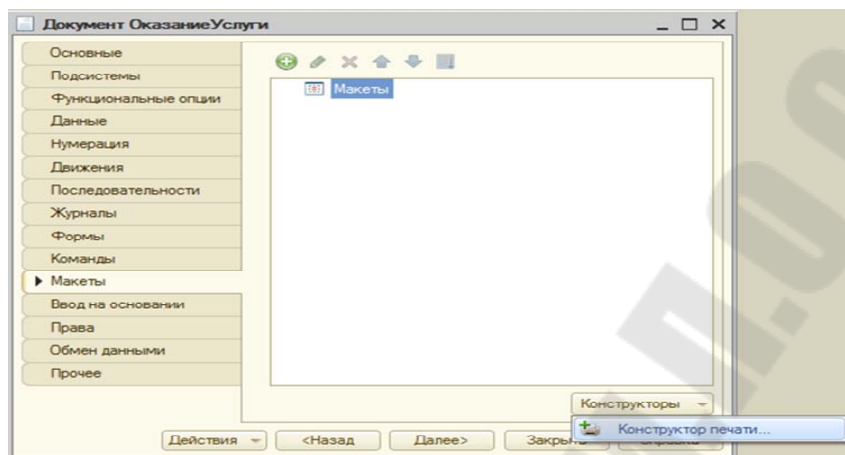


Рис. 3.7. Вкладка Макеты формы документа ОказаниеУслуг

Откроется модуль команды *Печать*, модуль менеджера документа *ОказаниеУслуги* и макет этого документа (рис. 3.8).

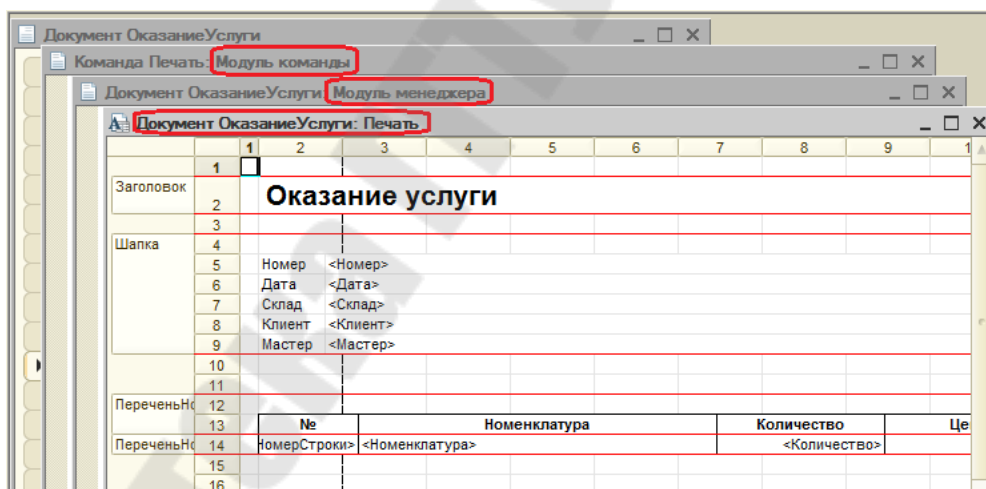


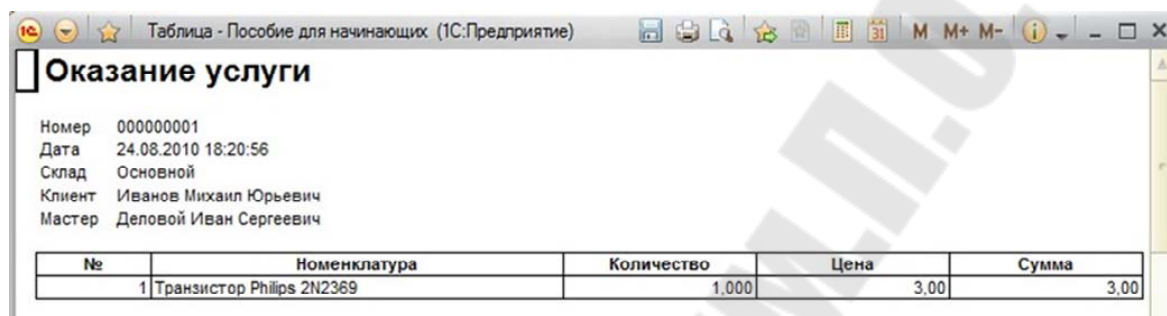
Рис. 3.8. Макет формы документа ОказаниеУслуги

Таким образом, конструктор создал:

- макет печатной формы документа *ОказаниеУслуги* с именем *Печать*;
- команду документа *ОказаниеУслуги* с именем *Печать*. В модуль этой команды помещен обработчик, вызывающий процедуру печати документа, выполняющуюся на сервере. Сама процедура печати помещена в модуль менеджера документа;

– в командную панель формы документа *ОказаниеУслуги* помещена команда **Печать** для формирования печатной формы документа. При этом команда **Печать** принадлежит всему документу, а не одной его форме, поэтому может использоваться в любой его форме.

Запустим 1С: Предприятие в режиме отладки и откроем документ *Оказание услуги № 1*. Нажимаем на появившуюся кнопку **Печать**. Откроется печатная форма документа. Единственное, чего в ней не хватает – итоговой суммы документа (рис. 3.9).



The screenshot shows a window titled 'Таблица - Пособие для начинающих (1С:Предприятие)'. The main content area is titled 'Оказание услуги' and contains the following information:

Номер 000000001  
Дата 24.08.2010 18:20:56  
Склад Основной  
Клиент Иванов Михаил Юрьевич  
Мастер Деловой Иван Сергеевич

№	Номенклатура	Количество	Цена	Сумма
1	Транзистор Philips 2N2369	1.000	3.00	3.00

Рис. 3.9. Печатная форма документа *Оказание услуги № 1*

**Шаг 2.** Редактирование макета. Возвращаемся в конфигуратор. Добавим итоговую сумму в печатную форму.

Для этого раскроем дерево документа *ОказаниеУслуги* и дважды щелкаем на макете **Печать** (Если было закрыто перед этим).

Макет документа состоит из именованных областей, которые в определенном порядке выводятся на печать. Именованные области слева созданы конструктором, однако и разработчик может сам создавать или удалять области, переименовывать их и т. д.

Добавим новую область для вывода итоговой суммы документа. Выделяем мышью две пустые строки под табличной частью документа и выполняем **Таблица/Имена/Назначить имя**. Называем область **Всего** и нажимаем **ОК** (рис. 3.10).

Слева появилась новая область *Всего*. Но ширина области *Всего* не совпадает с остальными, ее необходимо изменить. Выделяем две строчки созданной области *Всего*, тянем мышью в заголовке таблицы за правую границу колонки 2 так, чтобы ее ширина совпала с шириной колонки № в шапке документа. Требуется согласие, когда платформа предложит создать новый формат строк. Теперь сделаем такое же изменение ширины для колонок 3, 4, 5, 6, чтобы они соответствовали колонкам в шапке.

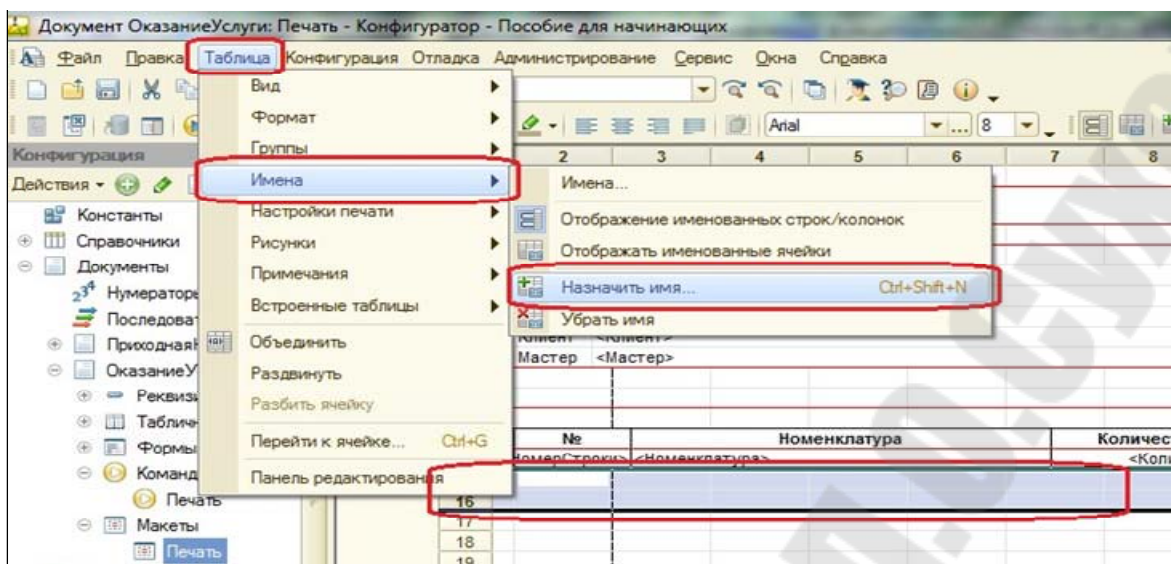



Рис. 3.10. Создание дополнительной области в форме документа

В созданной области во второй строчке колонки **Цена** необходимо написать **ВСЕГО:**, а в колонке **Сумма** запишем **ВсегоПоДокументу** (рис. 3.11).

Проконтролировать вид печатной формы можно с помощью кнопки предварительного просмотра .

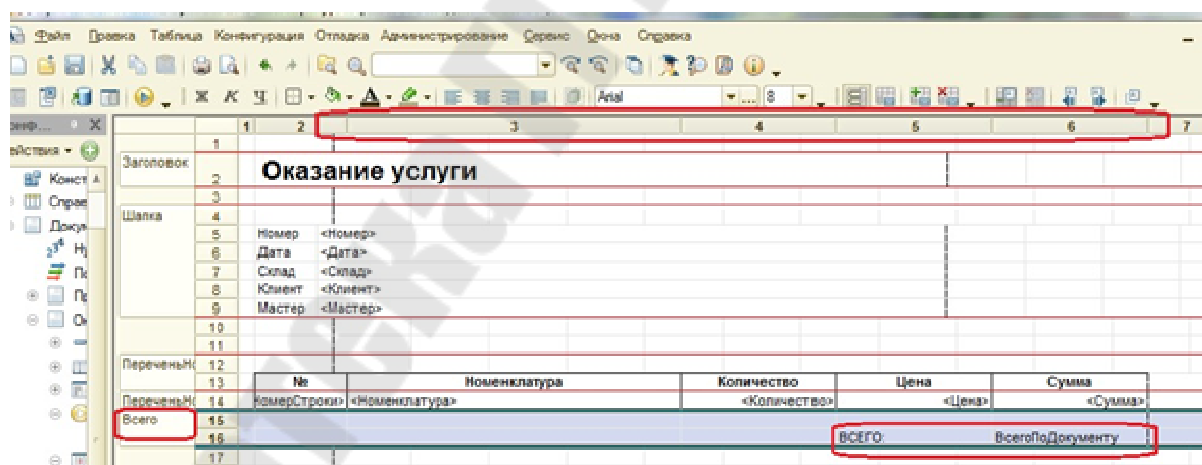
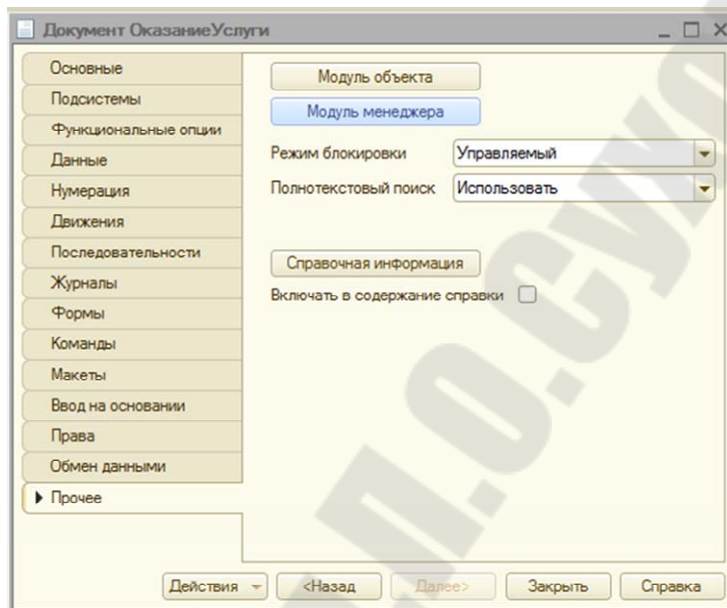
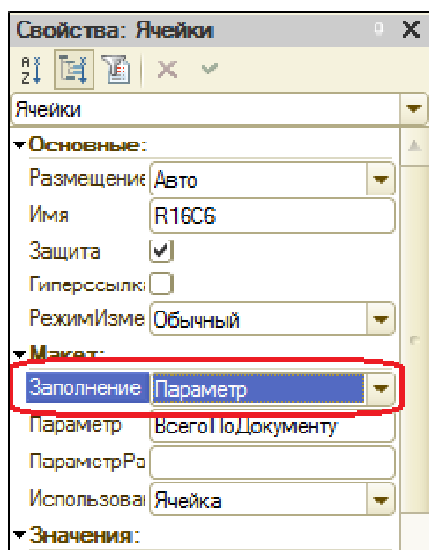


Рис. 3.11. Настройка итоговых полей в форме документа

Выделяем ячейку, где записано **ВсегоПоДокументу**, вызываем **контекстное меню/Свойства**. В свойстве **Заполнение** указываем, что в этой ячейке будет находиться не текст, а параметр (рис. 3.12, а). Подробнее об этом выборе.



а)

б)

Рис. 3.12. Настройка заполнения поля ВсегоПоДокументу

В данном случае параметр будет заменен некоторым значением, которое может быть присвоено ему средствами встроенного языка. Текст, содержащийся в ячейке, является именем этого параметра.

Откроем модуль менеджера документа *ОказаниеУслуги*. Для этого перейдем на закладку **Прочее** окна редактирования объекта *ОказаниеУслуги* и нажмем кнопку **Модуль менеджера** (или закроем макет, менеджер должен быть открыт под ним).

Появится возможность увидеть часть процедуры печати. Выделенные жирным шрифтом строчки являются новыми.

```

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");
Шапка = Макет.ПолучитьОбласть("Шапка");
ОбластьПереченьНоменклатурыШапка = Ма-
кет.ПолучитьОбласть("ПереченьНоменклатурыШапка");
ОбластьПереченьНоменклатуры = Макет.ПолучитьОбласть("ПереченьНоменклатуры");
ОбластьИтог = Макет.ПолучитьОбласть("Всего");
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(ОбластьПереченьНоменклатурыШапка);
    ВыборкаПереченьНоменклатуры = Выборка.ПереченьНоменклатуры.Выбрать();
СуммаИтог = 0;
Пока ВыборкаПереченьНоменклатуры.Следующий() Цикл
ОбластьПереченьНоменклатуры.Параметры.Заполнить(ВыборкаПереченьНоменклатуры);
    ТабДок.Вывести(ОбластьПереченьНоменклатуры,ВыборкаПереченьНоменклатуры.Уровень());
СуммаИтог = СуммаИтог + ВыборкаПереченьНоменклатуры.Сумма;
КонецЦикла;
ОбластьИтог.Параметры.ВсегоПоДокументу = СуммаИтог;
ТабДок.Вывести(ОбластьИтог);
ВставитьРазделительСтраниц = Истина;
КонецЦикла;

```

Запустите 1С: Предприятие и просмотрите форму печати документа *Оказание услуги № 1*.

Шаг 3. Редактирование формы. Возвращаемся в конфигуратор. Удобно, если в процессе создания документа можно было оперативно, не печатая его, знать итоговую сумму по документу (как в Excel). Для этого необходимо внести небольшие изменения в форму документа *ОказаниеУслуги*.

Открываем в дереве объектов форму документа *ОказаниеУслуги*, дважды щелкнув по ней. В левом верхнем окне вызываем свойства пункта *ПереченьНоменклатуры*. Установите свойство *Подвал*, которое определяет наличие подвала (нижней части) у таблицы формы (рис. 3.13).

Затем открываем свойства элемента формы *ПереченьНоменклатурыЦена* и устанавливаем (рис. 3.14):

- *Текст подвала* – Всего;
- *Горизонтальное положение* – Право;
- *Шрифт подвала* – Жирный.

Далее следует открыть свойства элемента *ПереченьНоменклатурыСумма* и установить: *Горизонтальное положение* – Право; *Шрифт* – жирный.

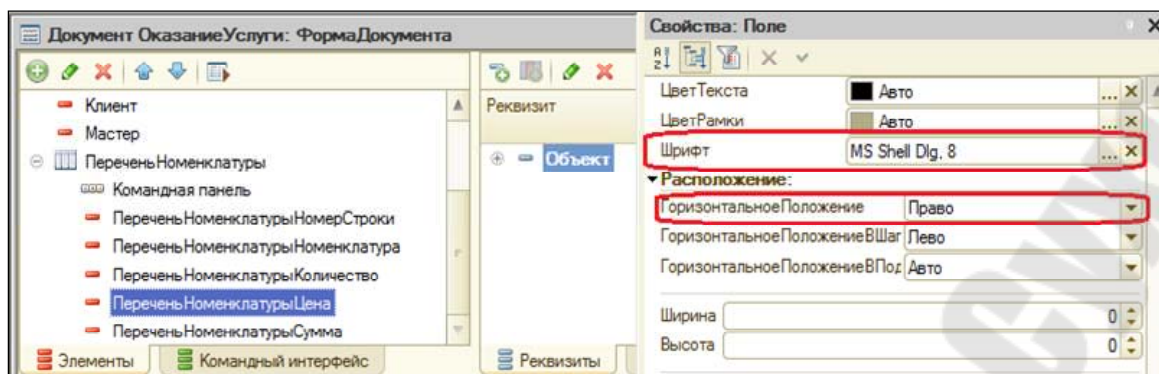


Рис. 3.13. Настройка нижней части таблицы

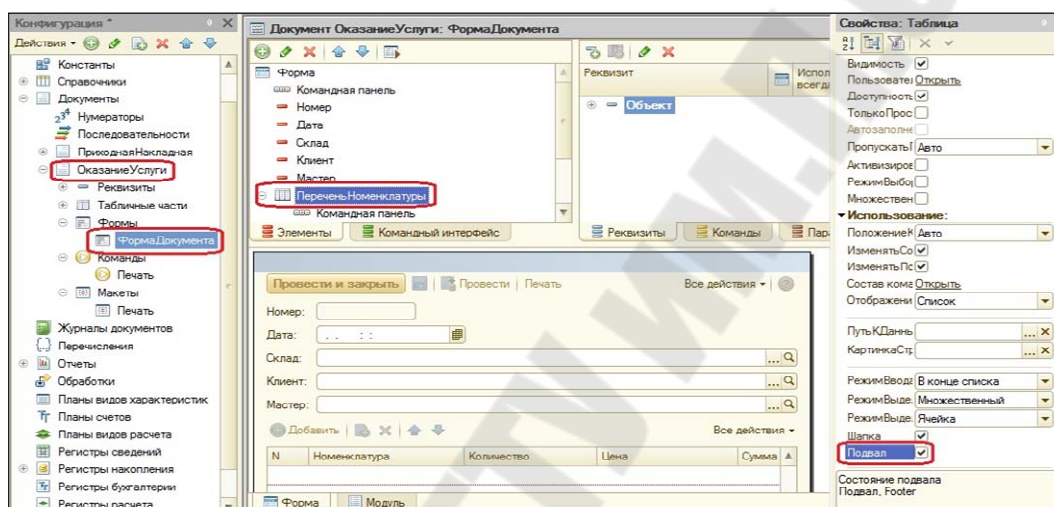


Рис. 3.14. Настройка элементов формы  
ПереченьНоменклатурыЦена

Для того, чтобы в подвале колонки *Сумма* отображался итог по ней, необходимо нажать кнопку выбора свойств в поле *ПутьКДаннымПодвала*. Раскрываем дерево реквизитов объекта и выбираем элемент *ИтогСумма*.

Запускаем 1С: Предприятие в режиме отладки. Открываем документ *Оказание услуги № 1*. По колонке *Сумма* в табличной части документа подсчитывается общий итог документа.

*Контрольные вопросы к заданию 3.2*

1. Для чего предназначен объект Макет?
2. Как создать макет с помощью конструктора печати?
3. Как изменить макет документа?
4. Как изменить внешний вид и поведение элемента формы?
5. Как отобразить сумму по колонке таблицы?

## ЛАБОРАТОРНАЯ РАБОТА 4 РЕГИСТРЫ СВЕДЕНИЙ. ПЕРЕЧИСЛЕНИЯ

**Задание 4.1. Создание периодического регистра сведений.** В соответствии с приведенным описанием создать периодический регистр *Цена на номенклатуру*.

### Теоретическая справка

В рассматриваемой фирме существует перечень услуг, который определяет стоимость каждой услуги. Но она меняется со временем, поэтому может сложиться ситуация, когда нужно будет изменить один из ранее проведенных документов *Оказание услуги*. В этом случае получить правильную стоимость услуги будет невозможно, поскольку в реквизите справочника будет храниться последнее введенное значение. Кроме этого может потребоваться видеть зависимость прибыли от изменения стоимости оказываемых услуг. И тогда необходимо будет иметь возможность анализировать изменение стоимости услуг во времени.

Поэтому для хранения стоимости услуг можно использовать объект *Регистр сведений*. Он предназначен для описания структуры хранения данных в разрезе нескольких измерений.

Принципиальное отличие *Регистра сведений* от *Регистра накопления* в том, что каждое движение регистра сведений устанавливает новое значение ресурса, в то время как движение регистра накопления изменяет существующее значение ресурса. По этой причине регистр сведений может хранить любые данные, а не только числовые как регистр накопления.

Еще одна важная особенность регистра сведений – способность хранить данные с привязкой ко времени. Благодаря этому регистр сведений может хранить историю изменения данных. Такой регистр называют *периодическим регистром сведений*.

Периодичность регистра сведений можно определить одним из значений: в пределах секунды, в пределах дня, в пределах месяца, в пределах квартала, в пределах года и в пределах регистратора (если установлен режим записи *Подчинение регистратору*).

Периодический регистр сведений всегда содержит служебное поле *Период*, добавляемое автоматически. Оно имеет тип *Дата* и служит для указания факта принадлежности записи к какому-либо



периоду. При записи данных в регистр, платформа всегда приводит значение этого поля к началу того периода, в который он попадает.

**Ключевым полем** в регистре сведений является совокупность значений измерений регистра и периода.

### Методические рекомендации по выполнению задания

**Шаг 1.** Добавление периодического регистра сведений. Добавим новый объект конфигурации *Регистр сведений* с именем *Цены*. Периодичность – в пределах секунды. Представление записи – *Цена*, Представление списка – *Цены на номенклатуру*. Следует обратить внимание на свойство *Режим записи*. По умолчанию оно имеет значение *Независимый*, т. е. создается независимый регистр сведений и в дальнейшем в него можно вводить данные без использования регистратора, вручную.

Перейдем на вкладку *Подсистемы*, отметим *УчетМатериалов*, *ОказаниеУслуг* и *Бухгалтерия*. Переходим на вкладку *Данные*. Создадим измерение *Номенклатура*, указав в качестве типа *Справочник-Ссылка.Номенклатура*.

Отмечаем, что измерение будет ведущим. Оно означает, что при удалении объекта, все записи регистра сведений по этому объекту будут автоматически удалены. Также благодаря этому свойству в панели навигации появится ссылка для перехода к записям этого регистра (рис. 4.1).

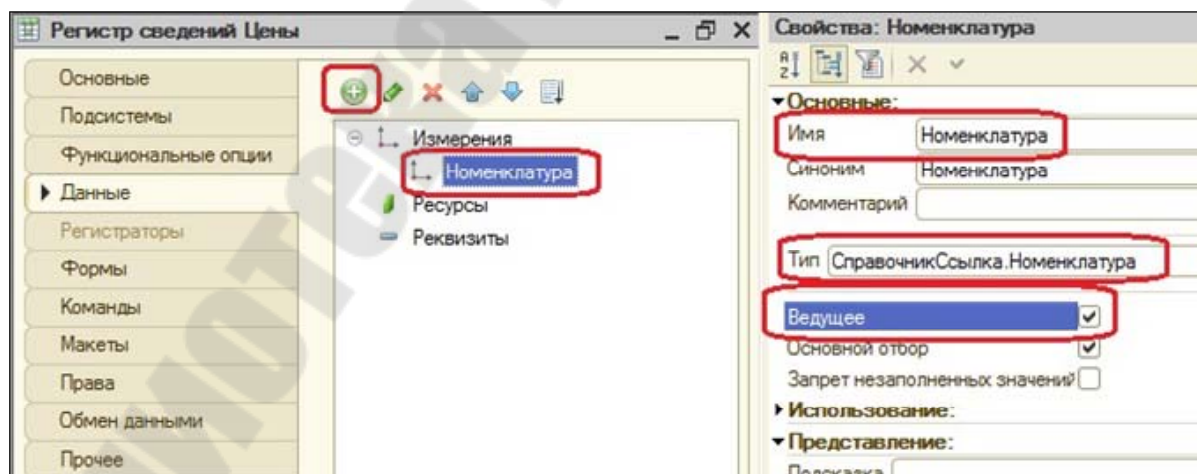


Рис. 4.1. Настройка свойств измерения *Номенклатура*

Также создаем ресурс *Цена*, тип *Число*, длина 15, точность 2, неотрицательное (рис. 4.2).

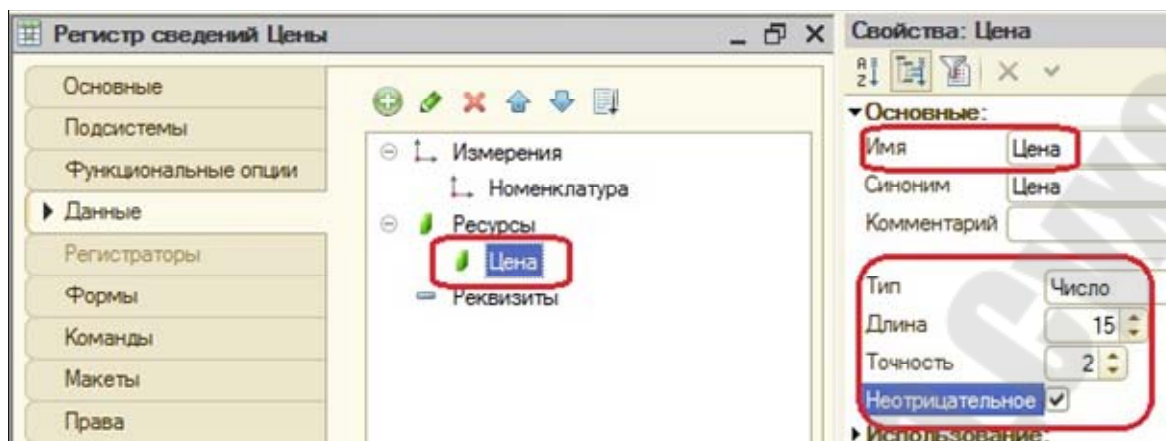


Рис. 4.2. Настройка свойств ресурса Цена

Запустим 1С: Предприятие. Видно, что в разделах *Бухгалтерия*, *Оказание услуг* и *Учет материалов* появилась команда для открытия списка регистра *Цена на номенклатуру*. Эта команда доступна по умолчанию, так как в отличие от регистров накопления предполагается изменение данных регистра пользователем.

Теперь имеется возможность заранее установить новые цены, которые вступят в действие не раньше указанного для них времени (рис. 4.3).

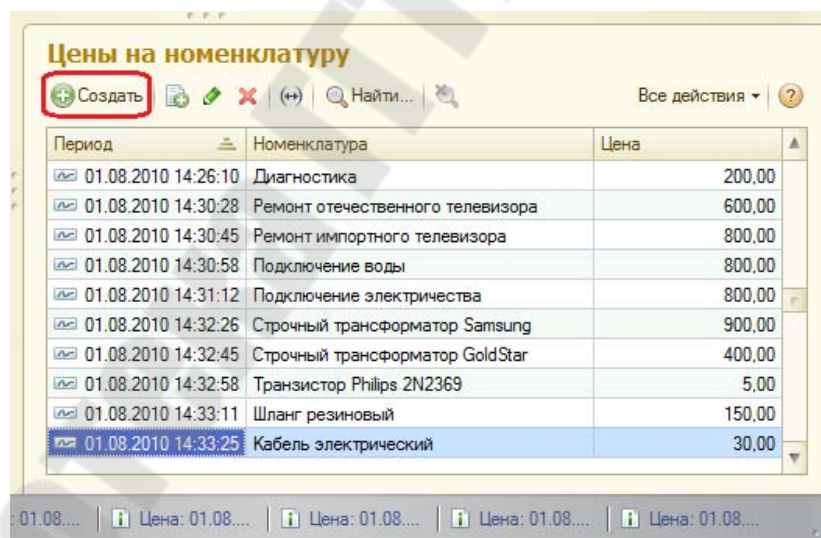


Рис. 4.3. Заполнение регистра сведений Цены

**Шаг 2.** Настройка автоматической подстановки цены в документ при выборе номенклатуры. Когда создается или изменяется документ *ОказаниеУслуги*, добавляется какую-либо номенклатура, необходимо, чтобы одновременно с этим в документ подставлялась актуальная цена этой номенклатуры, полученная из регистра сведений и соответст-

вующей дате документа. Для этого необходимо написать функцию, которая будет возвращать актуальную цену, а затем вызывать эту функцию в момент добавления номенклатуры. Поскольку такая возможность может понадобиться и в других документах, созданную функцию следует поместить в общий модуль.

В конфигураторе на ветке **Общие/Общие модули** добавим новый модуль с именем *РаботаСоСправочниками*.

У модуля по умолчанию установлен флажок **Сервер**. Это означает, что модуль будет компилироваться на сервере. Установим флажок **Вызов сервера**, чтобы процедуры и функции этого модуля можно было вызывать с клиента.

В модуль поместим следующий текст:

```
Функция РозничнаяЦена(АктуальнаяДата,ЭлементНоменклатуры) Экспорт
// Создать вспомогательный объект Отбор
Отбор = Новый Структура("Номенклатура", ЭлементНоменклатуры);
// Получить актуальные значения ресурсов регистра
ЗначенияРесурсов =РегистрыСведений.Цены.ПолучитьПоследнее(АктуальнаяДата,
Отбор);
Возврат ЗначенияРесурсов.Цена;
КонецФункции
```

Далее необходимо обеспечить автоматическое заполнение поля *Цена* после того, как пользователь выберет услугу. Причем цена услуги должна определяться исходя из даты создаваемого документа.

Найдем в конфигураторе документ *ОказаниеУслуги* и откроем его форму *ФормаДокумента*. Дважды щелкнем на элементе формы **ПереченьНоменклатурыНоменклатура**, найдем событие **ПриИзменении**, которое возникает при изменении значения поля.

Нажмем на значок *луны*. Система создаст шаблон процедуры обработчика этого события в модуле формы и откроет закладку **Модуль редактора формы**. В него необходимо внести следующий текст:

```
// Получить текущую строку табличной части
СтрокаТабличнойЧасти = Элементы.ПереченьНоменклатуры.ТекущиеДанные;
// Установить цену
СтрокаТабличнойЧасти.Цена = РаботаСоСправочниками.РозничнаяЦена(Объект.Дата,
СтрокаТабличнойЧасти.Номенклатура);
// Пересчитать сумму строки
РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);
```

Должно получиться следующее (рис. 4.4):

```

&НаКлиенте
[ Процедура ПереченьНоменклатурыКоличествоПриИзменении (Элемент) ]

&НаКлиенте
[ Процедура ПереченьНоменклатурыЦенаПриИзменении (Элемент) ]

&НаКлиенте
[ Процедура ПереченьНоменклатурыНоменклатураПриИзменении (Элемент)
  // Получить текущую строку табличной части
  СтрокаТабличнойЧасти = Элементы.ПереченьНоменклатуры.ТекущиеДанные;

  // Установить цену
  СтрокаТабличнойЧасти.Цена = РаботаСоСправочниками.РозничнаяЦена (Объект.Дата, СтрокаТабличнойЧасти.Номенклатура);

  // Пересчитать сумму строки
  РаботаСДокументами.РассчитатьСумму (СтрокаТабличнойЧасти);
]
КонецПроцедуры

```

Рис. 4.4. Программный код в Модуле редактора формы

Запустим 1С: Предприятие в режиме отладки и откроем регистр сведений *Цены на номенклатуру* (рис. 4.5).

Период	Номенклатура	Цена
01.08.2010 14:30:28	Ремонт отечественного телевизора	600,00
01.08.2010 14:30:45	Ремонт импортного телевизора	800,00
01.08.2010 14:30:58	Подключение воды	800,00
01.08.2010 14:31:12	Подключение электричества	800,00
01.08.2010 14:32:26	Строчный трансформатор Samsung	900,00
01.08.2010 14:32:45	Строчный трансформатор GoldStar	400,00
01.08.2010 14:32:58	Транзистор Philips 2N2369	5,00
01.08.2010 14:33:11	Шланг резиновый	150,00
01.08.2010 14:33:25	Кабель электрический	30,00
25.08.2010 14:58:16	Транзистор Philips 2N2369	7,00

Рис. 4.5. Установка новой цены на транзистор Philips

Теперь откроем документ *Оказание услуги № 1*, где израсходован один такой транзистор. Изменим дату документа на текущую (совпадающую с датой установки новой цены) и выберем транзистор заново. Будет установлено новое значение цены, последнее на эту дату (рис. 4.6).

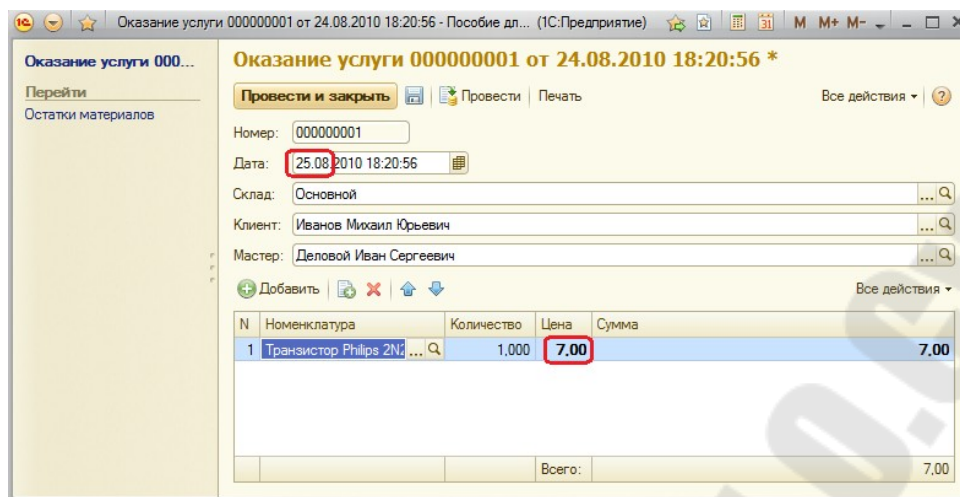


Рис. 4.6. Обновление цены при изменении даты документа

Таким образом, в документе появляется актуальная на момент создания документа цена услуги.

#### Контрольные вопросы к заданию 4.1

1. Для чего предназначен объект *Регистр сведений*?
2. В чем главные отличия регистра сведений от регистра накопления?
3. Что такое периодический регистр сведений и что такое независимый регистр сведений?
4. Как создать периодический регистр сведений?
5. Что такое ведущее измерение регистра?

**Задание 4.2. Создание перечислений.** В соответствии с приведенным описанием привязать реквизиты справочника *Номенклатура* к значениям перечисления *ВидНоменклатуры*. Настроить регистрацию расхода номенклатуры *Материал*.

#### Теоретическая справка

Объект *Перечисление* предназначен для описания структуры хранения постоянных наборов значений, неизменяемых в процессе работы конфигурации. На основе объекта *Перечисление*, платформа создает в базе данных таблицу, в которой хранится набор некоторых постоянных значений.

В реальной жизни этому объекту может соответствовать, например, перечисление вариантов цены – «включая НДС», «без НДС». Набор всех возможных значений, которые содержит перечисление, задается при конфигурировании системы, и пользователь не может изменять, удалять или добавлять новые.

### Методические рекомендации по выполнению задания

**Шаг 1.** Привязка номенклатуры к значениям перечисления *ВидНоменклатуры*. Откроем конфигуратор и создадим новый объект *Перечисление* с именем *ВидыНоменклатуры*. На закладке *Данные* добавим два значения перечисления: *Материал* и *Услуга* (рис. 4.7).

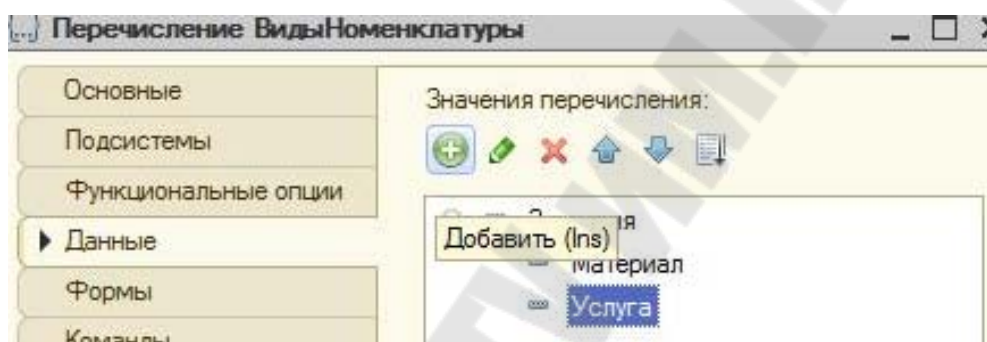


Рис. 4.6. Создание перечислений *ВидНоменклатуры*

Для привязки номенклатуры к значениям перечисления, необходимо сделать следующее:

- в режиме конфигуратора создать у справочника *Номенклатура* реквизит, который будет хранить значение перечисления;
- в режиме 1С: Предприятие проставить нужные значения этого реквизита для всех элементов справочника *Номенклатура*.

Добавим в справочник *Номенклатура* новый реквизит *ВидНоменклатуры* с типом *ПеречислениеСсылка.ВидыНоменклатуры*.

Запустим 1С: Предприятие в режиме отладки. Зайдем в *Учет материалов/Номенклатура*. Зададим каждому элементу справочника соответствующее значение реквизита *Вид номенклатуры* (рис. 4.7).

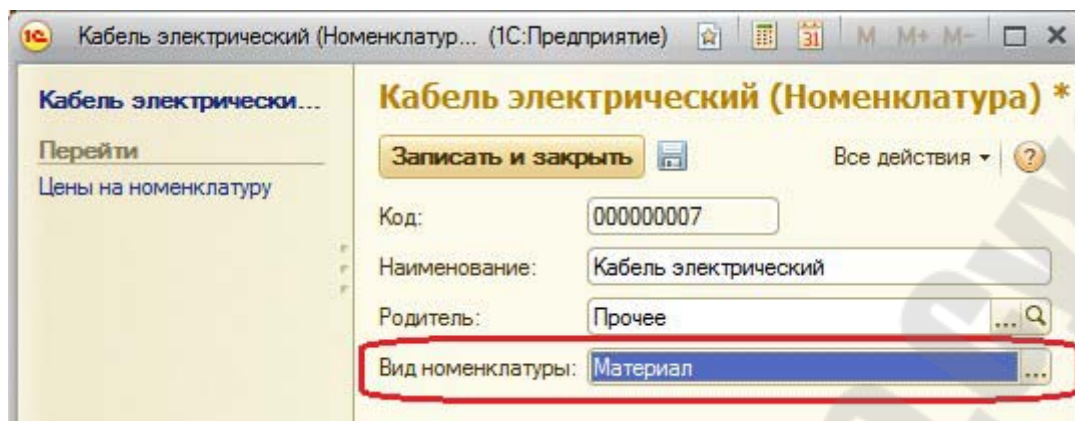


Рис. 4.7. Настройка видов для справочника *Номенклатура*

Шаг 2. Регистрация расхода номенклатуры *Материал*. На этапе настройки движения документа *ОказаниеУслуги* по регистру накопления *ОстаткиМатериалов*, было отмечено, что данные настройки не совсем правильные, поскольку в регистр будут попадать не только записи об израсходованных материалах, но и записи об оказанных услугах. Доработаем документ таким образом, чтобы в регистре появлялись только записи, относящиеся к расходу материалов.

Сначала в Конфигураторе изменим процедуру проведения документа, а потом в режиме 1С: Предприятия заново проведем все документы *Оказание услуги*, чтобы данные в регистре изменились в соответствии с новым алгоритмом проведения документа.

Откроем модуль документа *ОказаниеУслуги* (*контекстное меню документа/Открыть модуль объекта*) и добавим в обработчик события *ОбработкаПроведения* представленное ниже условие. Жирным выделены новые строки.

```

Процедура ОбработкаПроведения(Отказ, Режим)
//{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
Движения.ОстаткиМатериалов.Записывать = Истина;
Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
    Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры =
Перечисления.ВидыНоменклатуры.Материал Тогда
        // регистр ОстаткиМатериалов Расход
        Движение = Движения.ОстаткиМатериалов.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;

```

```
Движение.Склад = Склад;  
Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;  
КонецЕсли;  
КонецЦикла;  
//}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ  
КонецПроцедуры
```

Добавленный текст исключает выполнение операторов цикла для тех строк табличной части документа, в которых номенклатура не является материалом.

Запустим 1С: Предприятие в режиме отладки. Откроем список документов *Оказание услуг*. Откроем документ *Оказание услуги № 1* и внесем в него следующие изменения:

- удалим из табличной части строку, содержащую Транзистор Philips;
- добавим услугу – Подключение воды;
- добавим материал – Шланг резиновый.

Следует отметить, что цены подставляются автоматически из регистра сведений *Цены*. Проведем документ кнопкой **Провести**. Перейдем в *Остатки материалов* через **панель навигации**.

Обратите внимание, что в движения по регистру *Остатки материалов* включаются только строки, содержащие материалы. Запись про услугу *Подключение воды* в движения не попала.

#### *Контрольные вопросы к заданию 4.2*

1. Для чего предназначен объект *Перечисление*?
2. Как создать новое перечисление?
3. Как с помощью перечисления задать принадлежность элементов справочника к той или иной смысловой группе?



## ЛАБОРАТОРНАЯ РАБОТА 5 ПРОВЕДЕНИЕ ДОКУМЕНТА ПО НЕСКОЛЬКИМ РЕГИСТРАМ. ОБОРОТНЫЕ РЕГИСТРЫ НАКОПЛЕНИЯ

**Задание 5.1. Проведение приходной накладной по двум регистрам.** В соответствии с приведенным описанием организовать проведение документов *Приходная накладная* и *Оказание услуг* по двум регистрам *Остатки материалов* и *Стоимость материалов*.

### Теоретическая справка

В процесс управления фирмой необходимо знать, какие денежные средства были затрачены на приобретение материалов и каковы материальные запасы на текущий момент в денежном выражении. Так же требуется реализация задачи ведения суммового учета материалов по средней стоимости. То есть при закупке материалов они должны учитываться в ценах приобретения, а при расходе – по средней стоимости, которая рассчитывается исходя из общей суммы закупок данного материала и общего количества этого материала.

Для этих целей можно использовать регистр накопления *СтоимостьМатериалов*. Таким образом документы *ПриходнаяНакладная* и *ОказаниеУслуги* должны будут создавать движения не только в регистре *ОстаткиМатериалов*, но одновременно в регистре *СтоимостьМатериалов*, отражая изменения суммового учета.

### Методические рекомендации по выполнению задания

Шаг 1. Создание регистра накопления *СтоимостьМатериалов*. Создадим новый объект *Регистр накопления* с именем *СтоимостьМатериалов*. Расширенное представление списка зададим как движения по регистру *Стоимость материалов*. На закладке **Подсистемы** отметим *Бухгалтерия*, *УчетМатериалов* и *ОказаниеУслуг*. На закладке **Данные** создадим одно измерение – *Материал* с типом *СправочникСсылка.Номенклатура* и один ресурс – *Стоимость* с типом *Число*, длина 15, точность 2.

Далее отредактируем командный интерфейс, чтобы в подсистемах *Бухгалтерия*, *ОказаниеУслуг* и *УчетМатериалов* была доступна ссылка для просмотра созданного регистра накопления в разделе **Панель навигации**. *См.также*.

Шаг 2. Проведение приходной накладной по двум регистрам. Откроем в конфигураторе окно редактирования объекта Документ *Приходная накладная* и перейдем на вкладку *Движения*.

В списке регистров необходимо отметить, что документ будет создавать движения и по регистру *СтоимостьМатериалов* (рис. 5.1).

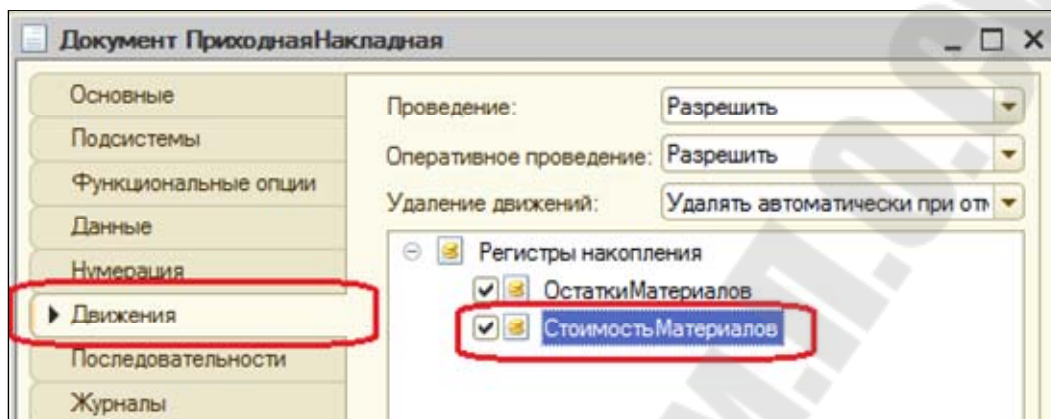


Рис. 5.1. Настройка движения документа *ПриходнаяНакладная*

На этот раз не будем использовать конструктор движений, а внесем изменения прямо в обработчик события *Обработка Проведения* документа *ПриходнаяНакладная*.

Дело в том, что с помощью конструктора можно создавать движения одновременно и в нескольких регистрах, но тогда процедура проведения, которая была написана ранее, затрется новой процедурой, созданной конструктором.

Перейдем на вкладку *Прочее* и откроем модуль объекта. В самом конце цикла перед строкой *КонецЦикла* добавим строки кода, создающие движение в регистре *СтоимостьМатериалов*.

```

Процедура ОбработкаПроведения(Отказ, Режим)
//{{ КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора,
// внесенные вручную изменения будут утеряны!!!
// регистр ОстаткиМатериалов Приход
Движения.ОстаткиМатериалов.Записывать = Истина;
Движения.СтоимостьМатериалов.Записывать = Истина;
Для Каждого ТекСтрокаМатериалы Из Материалы Цикл
    Движение = Движения.ОстаткиМатериалов.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    Движение.Период = Дата;
    Движение.Материал = ТекСтрокаМатериалы.Материал;

```

```

Движение.Склад = Склад;
Движение.Количество = ТекСтрокаМатериалы.Количество;
// регистр Стоимость Материалов Приход
Движение = Движения.СтоимостьМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаМатериалы.Материал;
Движение.Стоимость = ТекСтрокаМатериалы.Сумма;
КонецЦикла;
//}} КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры

```

Отредактируем командный интерфейс формы документа, чтобы в панели навигации формы иметь возможность переходить к списку записей регистра *Стоимость Материалов*, связанному с документом.

Для этого откроем форму документа *Приходная Накладная* и перейдем на вкладку *Командный интерфейс*. В разделе *Панель навигации* следует раскрыть группу *Перейти* и установить видимость для команды *Стоимость материалов...* (рис. 5.2).

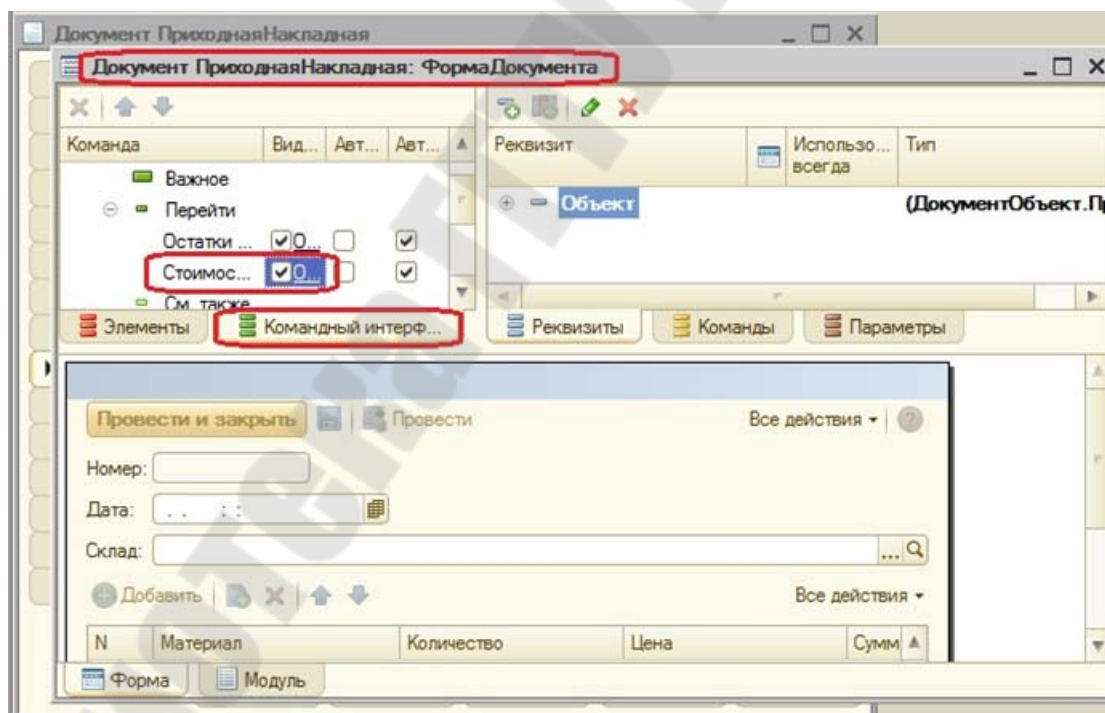


Рис. 5.2. Настройка командного интерфейса документа

Запустим 1С: Предприятие и **перепроведем** все документы *Приходная накладная*. Затем откроем первый документ и, выполнив из него переход к регистрам *Остатки материалов* и *Стоимость ма-*

териалов, убедимся, что документ создает желаемые записи в обоих регистрах.

**Шаг 3.** Проведение документа *ОказаниеУслуги* по двум регистрам. Внесем изменения в процедуру обработки проведения документа *ОказаниеУслуги*. Суть изменения – при списании материалов, израсходованных в процессе оказания услуги, должна быть возможность указывать различную стоимость для одного и того же материала, которая рассчитана руководством фирмы исходя из текущих потребностей.

Поскольку в документе отражена только цена номенклатуры, понадобится:

1. Добавить в табличную часть документа еще один реквизит, в котором будет указываться стоимость номенклатуры.

2. После этого изменить процедуру проведения документа *ОказаниеУслуги*.

3. В режиме 1С: Предприятие перепровести все эти документы, чтобы заработал новый алгоритм.

Откроем окно редактирования объекта *Документ ОказаниеУслуги* и перейдем на вкладку *Данные*. Создадим новый реквизит табличной части документа с именем *Стоимость*, типом Число, длиной 15 и точностью 2, неотрицательное (рис. 5.3).

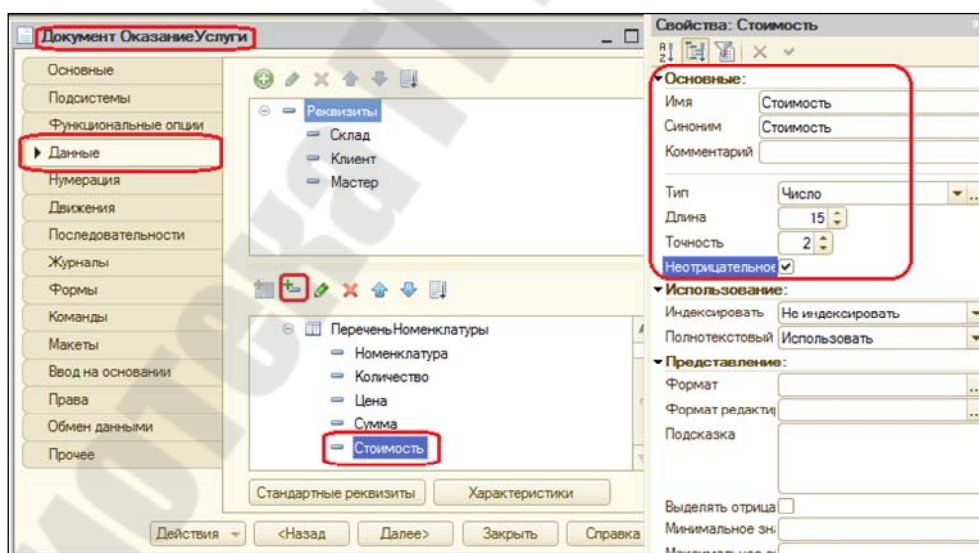


Рис. 5.3. Создание реквизита табличной части *Стоимость*

После этого откроем форму *ФормаДокумента* документа *ОказаниеУслуги* и добавим в табличную часть *ПереченьНоменклатуры* поле, отображающее новый реквизит *Стоимость*. Для этого в правом

верхнем окне редактора форм на закладке *Реквизиты* раскроем реквизит формы *Объект*. Найдем в табличной части реквизит *Стоимость* и мышью перетащим его в окно элементов формы (левый верхний угол) после поля *ПереченьНоменклатурыСтоимость*. Новый реквизит тут же отобразится внизу окна в форме документа (рис. 5.4).

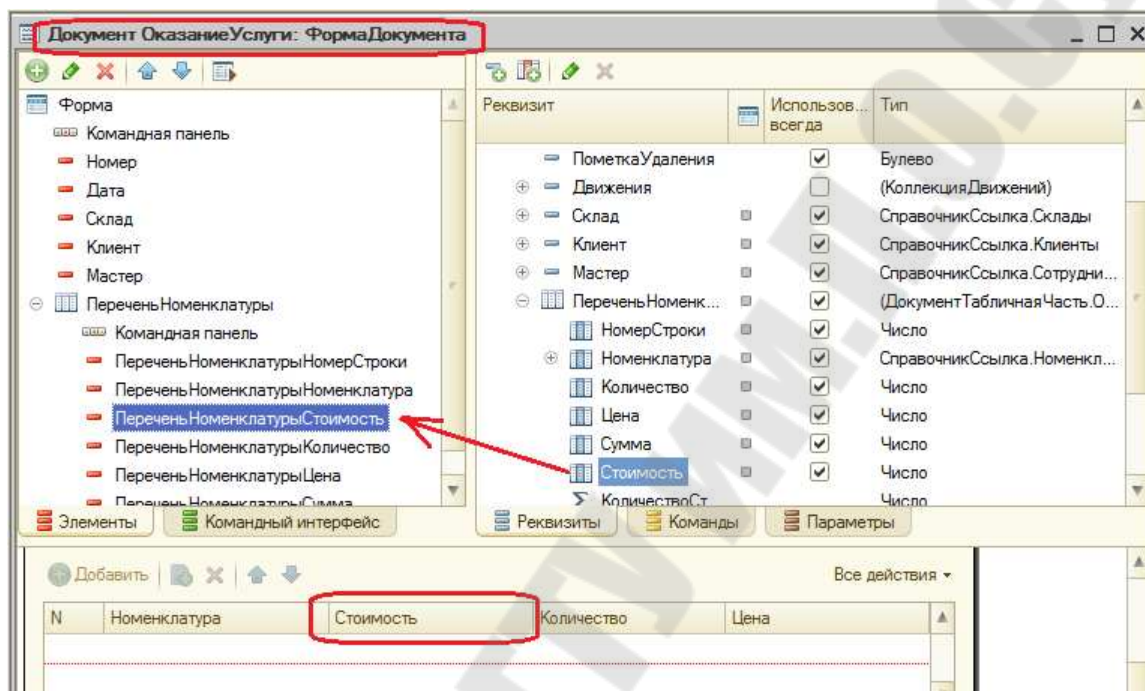


Рис. 5.4. Настройка отображения реквизита *Стоимость*

Теперь создадим движения документа *ОказаниеУслуги* таким же образом, как делали это для документа *ПриходнаяНакладная*.

В окне редактирования *Документа ОказаниеУслуги* перейдем на вкладку *Движения*. В списке регистров отметим *СтоимостьМатериалов*. Перейдем на закладку *Прочее* и откройте модуль объекта. Внесем изменения в процедуру обработки проведения, новые строки выделены жирным:

```
Процедура ОбработкаПроведения(Отказ, Режим)
Движения.ОстаткиМатериалов.Записывать = Истина;
Движения.СтоимостьМатериалов.Записывать = Истина;
Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
    Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры =
        Перечисления.ВидыНоменклатуры.Материал Тогда
        // регистр ОстаткиМатериалов Расход
            Движение = Движения.ОстаткиМатериалов.Добавить();
            Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
```

```

Движение.Период = Дата;
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Склад = Склад;
Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;
// регистр СтоимостьМатериалов Расход
Движение = Движения.СтоимостьМатериалов.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Количество*
ТекСтрокаПереченьНоменклатуры.Стоимость;
КонецЕсли;
КонецЦикла;
КонецПроцедуры

```

Отредактируем командный интерфейс формы документа, чтобы из панели навигации можно было попасть в регистр *Стоимость материалов*, связанному с документом (рис. 5.5).

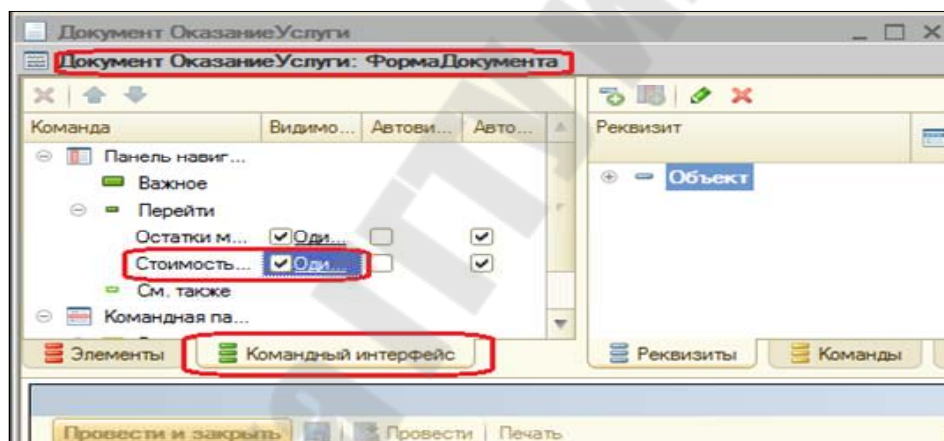


Рис. 5.5. Настройка командного интерфейса формы документа

Запустим 1С: Предприятие, откроем документ *Оказание услуги № 1* и укажем в нем стоимость резинового шланга – 100. Нажмем **Провести** и перейдем в регистр *Стоимость материалов* через панель навигации (рис. 5.6).

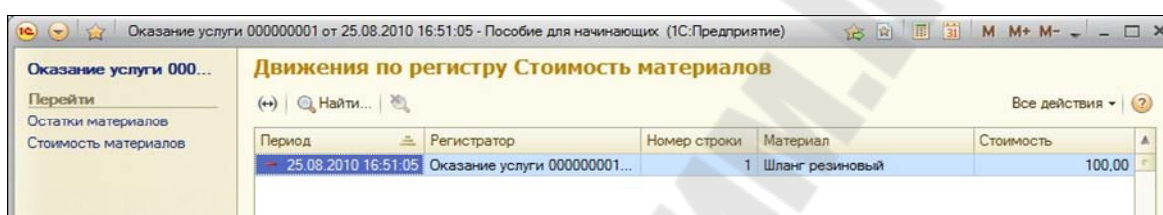
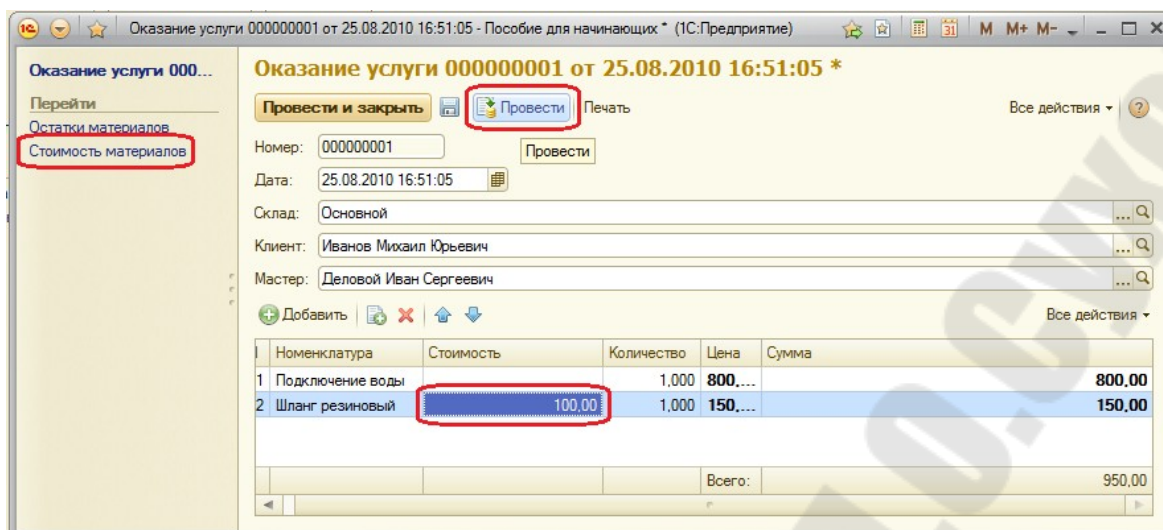
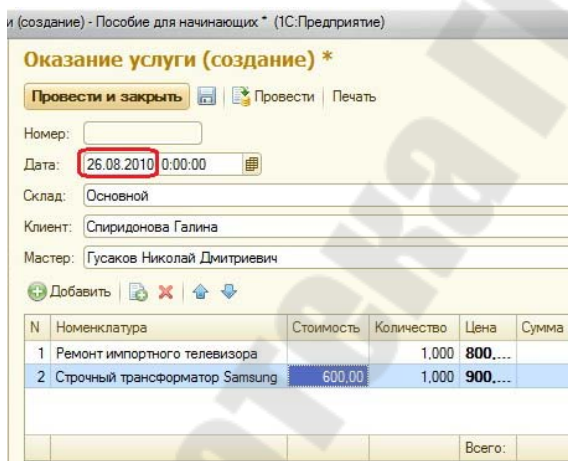
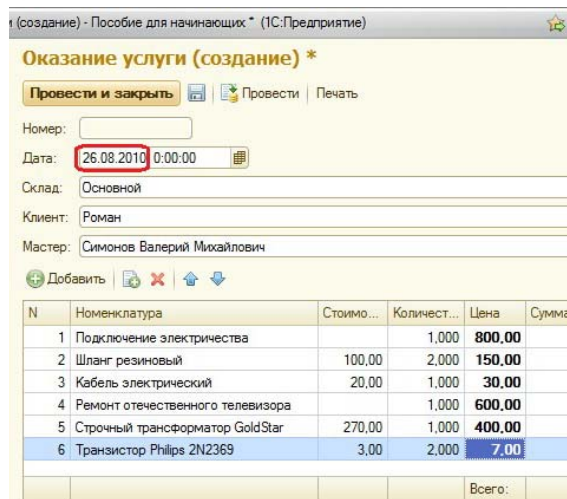


Рис. 5.6. Перепроведение документа Оказание услуги № 1

Создайте и проведите еще два документа *ОказаниеУслуги* (рис. 5.7).



а)



б)

Рис. 5.7. Создание новых документов *ОказаниеУслуги*

Проверьте результаты проведения созданных документов по регистру *Стоимость материалов*.

### *Контрольные вопросы к заданию 5.1*

1. Как создать движения документа по нескольким регистрам?
2. Как создать движения документа без использования конструктора движений?
3. Как добавить в форму документа новый реквизит?

**Задание 5.2. Создание оборотного регистра накопления.** В соответствии с приведенным описанием создать оборотный регистр накопления *Продажи*.

### **Теоретическая справка**

До сих пор создавались в регистрах накопления движения только для строк документа, которые содержат материалы. Услуги, содержащиеся в документе, никак не учитывались.

Бессмысленно говорить о том, сколько услуг было и сколько осталось, важна только сумма и количество услуг за промежуток времени. Кроме этого, интересны следующие моменты:

- Какие именно услуги оказаны (для рейтинга услуг);
- Какому именно клиенту оказывались услуги (чтобы, например, предоставить ему скидку от объема ранее оплаченных услуг);
- Какой мастер предоставлял услуги (чтобы начислить ему зарплату) и пр.

Очевидно, что существующие регистры накопления не подходят для решения этих задач. Поэтому создадим еще одно хранилище данных – оборотный регистр *Продажи*.

Регистры накопления могут быть регистрами остатков и регистрами оборотов. Существующие в конфигурации регистры накопления *ОстаткиМатериалов* и *СтоимостьМатериалов* являются регистрами остатков. Ранее, при создании отчета *Материалы* в конструкторе запроса было видно, что для таких регистров система создает **три виртуальные таблицы**: таблица остатков, оборотов и совокупная таблица остатков и оборотов.

Оборотный регистр очень похож на регистр остатков, только накапливает он обороты, остатки ему безразличны. Поэтому единственная виртуальная таблица будет таблицей оборотов.

**По каждому из измерений регистра накопления остатков ресурсы обязательно должны изменяться в обе стороны: приход**



**и расход.** Не должно существовать таких измерений, по которым осуществляется только приход или только расход. Нарушение этого принципа приведет к замедлению работы системы. Для реквизитов же регистра накопления этот принцип не важен. По реквизитам регистра ресурсы могут только приходиться или только расходоваться.

### Методические рекомендации по выполнению задания

Шаг 1. Создание оборотного регистра накопления *Продажи*. Создадим новый регистр накопления с именем **Продажи**, вид – *Обороты*. Расширенное представление списка – *Движение по регистру Продажи*. На вкладке **Подсистемы** отмечаем *Бухгалтерия*, *Учет-Материалов*, *ОказаниеУслуг*.

На вкладке **Данные** создаем измерения регистра (рис. 5.8):

- *Номенклатура*, тип СправочникСсылка.Номенклатура;
- *Клиент*, тип СправочникСсылка.Клиенты;
- *Мастер*, тип СправочникСсылка.Сотрудники.

Также создаем три ресурса:

- *Количество*, тип Число, длина 15, точность 2;
- *Выручка*, тип Число, длина 15, точность 2;
- *Стоимость*, тип Число, длина 15, точность 2.

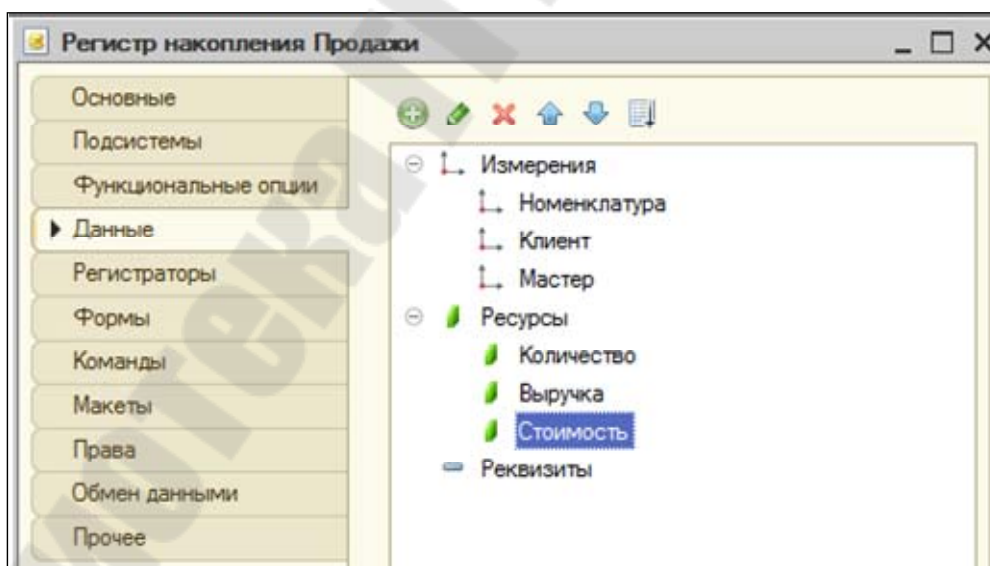


Рис. 5.8. Создание регистра накопления *Продажи*

Отредактируем командный интерфейс, чтобы в подсистемах *Бухгалтерия*, *ОказаниеУслуг* и *УчетМатериалов* была доступна ссылка для просмотра оборотного регистра накопления *Продажи*.

**(Все Подсистемы/Продажи** включить и перетащить в **Панель навигации. См. также**).

**Шаг 2.** Проведение документа *Оказание услуги* по трем регистрам. Сначала изменим процедуру проведения документа *Оказание Услуги*, а затем в режиме 1С: Предприятие перепроведем все эти документы, чтобы отработал новый алгоритм проведения.

Откроем окно редактирования объекта *Документ Оказание Услуги* и на вкладке *Движения* укажем регистр *Продажи*. Перейдем на вкладку *Прочее* и откроем модуль объекта. Изменим процедуру обработки следующим образом:

```
Процедура ОбработкаПроведения(Отказ, Режим)
Движения.ОстаткиМатериалов.Записывать = Истина;
Движения.СтоимостьМатериалов.Записывать = Истина;
Движения.Продажи.Записывать = Истина;
Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
    Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры =
Перечисления.ВидыНоменклатуры.Материал Тогда
    // регистр ОстаткиМатериалов Расход
    Движение = Движения.ОстаткиМатериалов.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
    Движение.Склад = Склад;
    Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;
    // регистр СтоимостьМатериалов Расход
    Движение = Движения.СтоимостьМатериалов.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
    Движение.Стоимость =
ТекСтрокаПереченьНоменклатуры.Количество*ТекСтрокаПереченьНоменклатуры. Стои-
мость;
КонецЕсли;
// Регистр Продажи
Движение = Движения.Продажи.Добавить();
Движение.Период = Дата;
Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Клиент = Клиент;
Движение.Мастер = Мастер;
Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;
Движение.Выручка = ТекСтрокаПереченьНоменклатуры.Сумма;
Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Стоимость * 
ТекСтрокаПереченьНоменклатуры.Количество;
КонецЦикла;
КонецПроцедуры
```

Следует обратить внимание, что у оборотного регистра отсутствует свойство *ВидДвижения*, так как отражение вида движения (приход или расход) имеет смысл только при учете остатков.

Отредактируем командный интерфейс формы документа *ОказаниеУслуги*, включив видимость команды открытия регистра *Продажи*.

Запустим 1С: Предприятие, перепроведем каждый из документов *Оказание услуги*, после переходя из каждого в регистр *Продажи* (рис. 5.9).

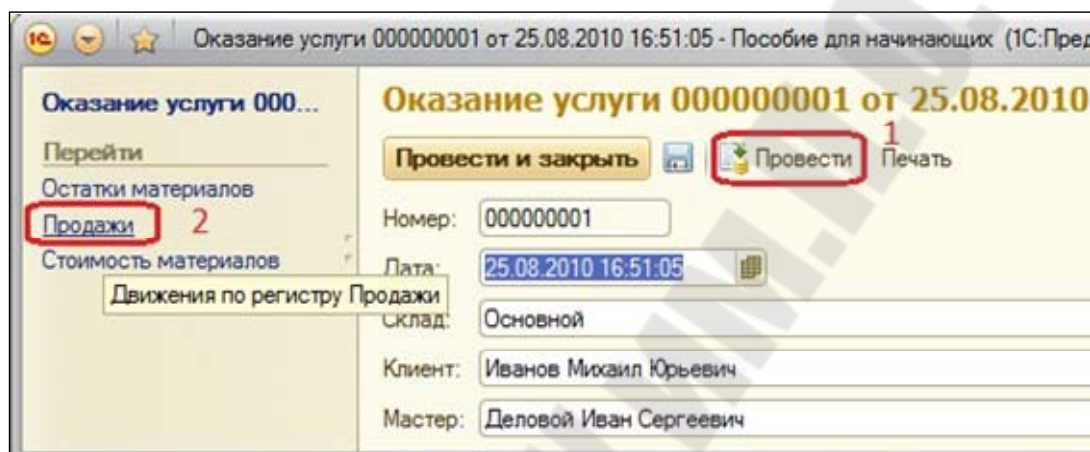


Рис. 5.9. Перепроведение документов *Оказание услуги*

### *Контрольные вопросы к заданию 5.2*

1. Что такое оборотный регистр накопления?
2. В чем отличие между регистром накопления остатков и оборотным регистром накопления?
3. Как выбирать реквизиты и измерения при создании регистров накопления?
4. Как создать оборотный регистр накопления?

## ЛАБОРАТОРНАЯ РАБОТА 6 СОЗДАНИЕ ОТЧЕТОВ

**Задание 6.1 Выбор данных из одной таблицы.** В соответствии с приведенным описанием создать и настроить отчет *Оказание услуг*.

### Теоретическая справка

Система поддерживает два способа доступа к данным, хранящимся в базе данных: **объектный** (для чтения и записи) и **табличный** (для чтения).

*Объектный* способ реализован через использование объектов встроенного языка. Важная особенность этого способа – при обращении к объекту встроенного языка обращение осуществляется к совокупности данных в базе данных как к единому целому. Такой способ обеспечивает сохранение целостности объектов, их кэширование, вызов соответствующих обработчиков событий и т. д.

*Табличный* способ доступа к данным реализован с помощью запросов к базе данных на языке запросов. В данном способе разработчик получает возможность оперировать отдельными полями таблиц. Этот способ предназначен для отбора, группировки, сортировки, объединения выборок, расчета итогов и т. д.

Для работы с запросами используется объект встроенного языка **Запрос**. Он позволяет получать информацию, хранящуюся в полях базы данных в виде выборки. Исходную информацию запрос получает из *набора таблиц*. Все таблицы, которыми оперирует язык запросов, можно разделить на группы.

*Реальные* таблицы содержат данные одной реальной таблицы в базе данных. *Виртуальные* таблицы формируются в основном из данных нескольких таблиц базы данных. Общим для виртуальных таблиц является то, что им можно задать ряд параметров, определяющих, какие данные будут в них включены.

*Объектные (ссылочные)* таблицы представляют информацию ссылочных типов данных (справочники, документы, планы видов характеристик и т. д.). А *необъектные (несссылочные)* – все остальные типы данных (константы, регистры и т. д.).

Отличительной особенностью объектных таблиц является то, что они включают в себя поле *Ссылка*, содержащее ссылку на теку-

щую запись. Для таких таблиц возможно получение пользовательского представления объекта.

Текст запроса может состоять из нескольких *частей*: описание запроса, объединение результатов, упорядочение результатов, автоупорядочение, описание итогов. Обязательная часть запроса – первая. Остальные присутствуют по необходимости.

Большинство видов отчетов можно создать с помощью системы компоновки данных, при этом формирование запроса можно осуществлять при помощи конструктора запросов.

**Система компоновки данных** предназначена для создания произвольных отчетов в системе 1С: Предприятие и состоит из нескольких основных частей.

Исходные данные для компоновки отчета содержит в себе схема компоновки данных. Это наборы данных и методы работы с ними.

Отчет системы компоновки имеет сложную иерархическую структуру и может состоять из различных элементов (группировки, таблицы, диаграммы).

Пользователь может изменить существующую структуру отчета или создать новую, может настроить необходимый ему отбор, оформление элементов структуры, получить расшифровку по каждому элементу и т. д.

Последовательность работы системы компоновки можно представить так: разработчик создает *схему компоновки данных* и *настройки по умолчанию*. На основе этого компоновщик макета создает *макет*. Процессор компоновки *выбирает данные* из информационной базы согласно макету, собирает и оформляет эти данные. *Результат компоновки* обрабатывается процессором вывода и пользователь получает результирующий табличный документ.

## Методические рекомендации по выполнению задания

Шаг 1. Создание объекта *Отчет*. Создадим отчет *Реестр документов оказание услуги*, используя систему компоновки данных. Этот отчет будет выводить список существующих в базе данных документов *ОказаниеУслуги* в порядке их дат и номеров.

Добавим в **Конфигураторе** объект конфигурации **Отчет**. Повторим первые шаги по созданию отчета, описанные в задании 3.1.

На закладке **Основные** зададим имя отчета *РеестрДокументов ОказаниеУслуги*.

Установим свойство *Расширенное представление* как *Список оказанных услуг* для представления отчета в интерфейсе программы.

Создадим схему компоновки данных для отчета. Для этого нажмем кнопку *Открыть схему компоновки данных* или кнопку со значком лупы (рис. 6.1).

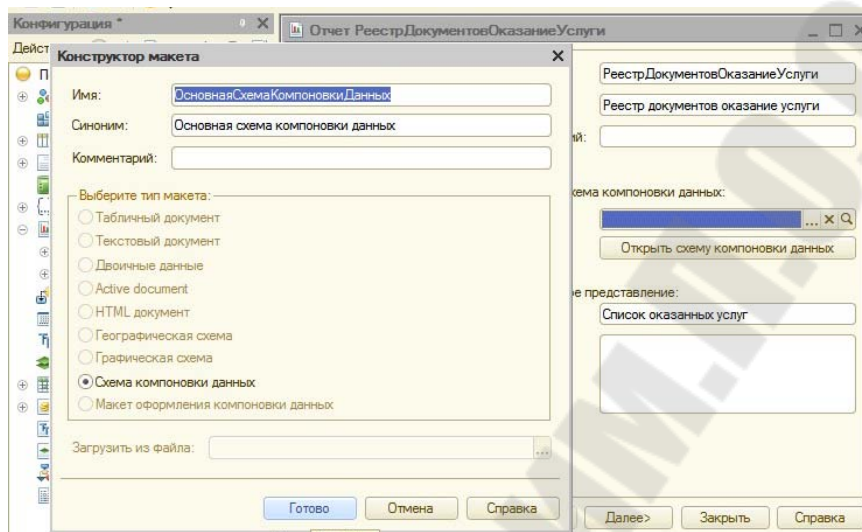


Рис. 6.1. Открытие схемы компоновки данных для отчета

Шаг 2. Создание запроса для отбора данных. В открывшемся окне нажмите *Готово*. В конструкторе схемы компоновки создайте *Набор данных – запрос*.

Далее нажмем кнопку *Конструктор запроса*. В качестве источника данных для запроса выберем объектную (ссылочную) таблицу документа *ОказаниеУслуги*. Из этой таблицы выберем следующие поля: *Склад*, *Мастер*, *Клиент*, *Ссылка* (рис. 6.2).

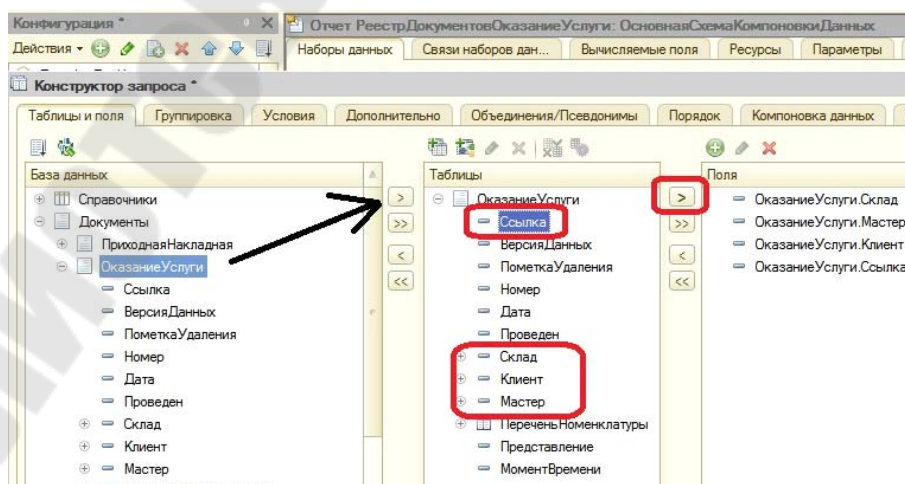


Рис. 6.2. Выбор источников запроса

Перейдем на закладку **Объединения/Псевдонимы** и укажем, что поле **Ссылка** будет иметь псевдоним *Документ*.

Имена полей лучше изменять в запросе, так как в этом случае в схему компоновки данных они перенесутся сразу в три колонки: *Поле*, *Путь* и *Заголовок*, и не нужно будет лишний раз их менять.

Перейдем в закладку **Порядок** и укажем, что результат запроса должен быть упорядочен по значению поля *Документ*.

Нажмем **ОК** и посмотрим, какой запрос сформировал конструктор:

ВЫБРАТЬ	ОказаниеУслуги.Склад, ОказаниеУслуги.Мастер, ОказаниеУслуги.Клиент, ОказаниеУслуги.Ссылка КАК Документ
ИЗ	Документ.ОказаниеУслуги КАК ОказаниеУслуги
УПОРЯДОЧИТЬ ПО	Документ

Обратите внимание, что запрос представляет собой русскоязычный аналог запроса на языке SQL.

**Шаг 3.** Стандартные настройки. Далее перейдем на закладку **Настройки** и создадим стандартные настройки, определяющие вывод информации в отчете.

Иерархическая структура отчета может содержать в различных сочетания три основных элемента: *Группировка* (для вывода информации в виде обычного линейного отчета); *Таблица* (для вывода информации в виде таблицы); *Диаграмма* (для вывода информации в виде диаграммы).

Для добавления нового элемента, в случае группировки, необходимо выделить в дереве структуры отчета корневой элемент **Отчет** и вызвать его контекстное меню. Можно также нажать кнопку **Добавить**, расположенную в командной панели окна или нажать клавишу **Insert**.

В окне выбора поля группировки нажимаем **ОК** (указываем, что будут выводиться детальные записи из информационной базы).

В структуре отчета появится *группировка Детальные записи*.

На закладке **Выбранные поля** перенесем мышью из списка доступных полей те, которые будут выводиться в отчет: *Документ*, *Склад*, *Мастер*, *Клиент* (рис. 6.3).

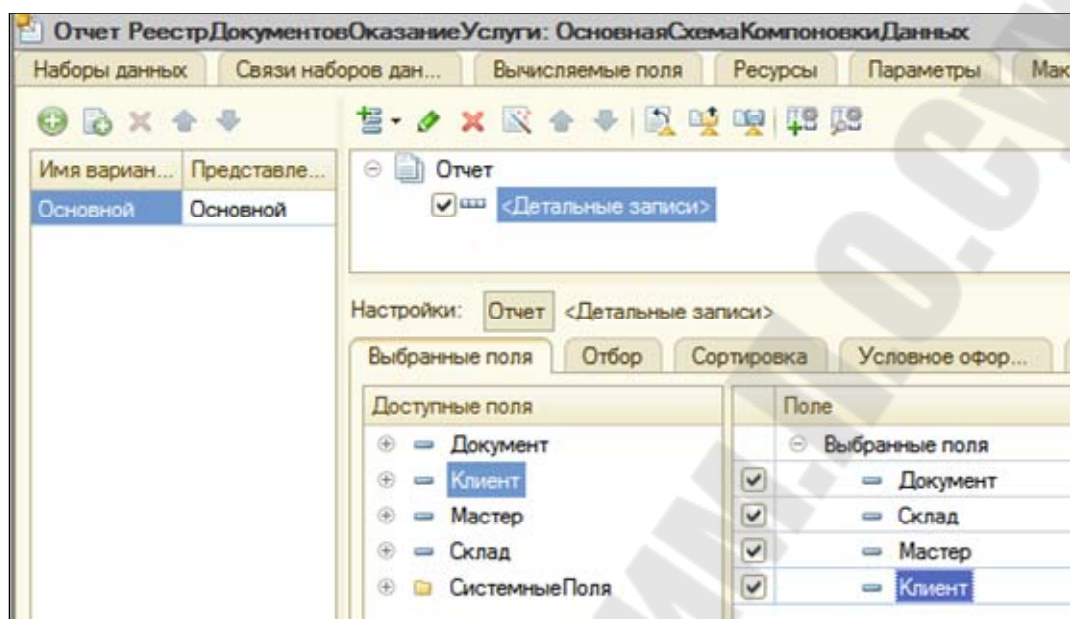


Рис. 6.3. Добавление полей для отчета

На этом создание отчета закончено. В заключение определим, в каких подсистемах будет отображаться наш отчет.

Закрываем конструктор схемы компоновки данных и в окне редактирования объекта конфигурации *Отчет РеестрДокументовОказаниеУслуги* и переходим на закладку **Подсистемы**.

Отметим в списке подсистему *ОказаниеУслуг*. Так отчет попадет в панель действий этой подсистемы.

Запустите 1С: Предприятие в режиме отладки и посмотрите, как работает отчет.

В открывшемся окне можно увидеть, что в панели действий раздела *Оказание услуг* в группе команд для выполнения отчетов появилась команда для формирования отчета *Реестр документов оказание услуги*. Причем если подвести к ней мышь, то появится всплывающая подсказка *Список оказанных услуг*, которая определяется свойством *Расширенное представление*, заданное для отчета.

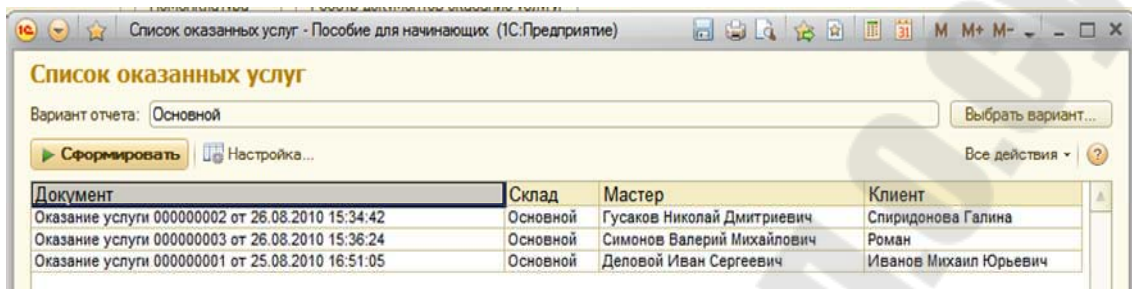
Выполним эту команду и посмотрим форму отчета, автоматически сформированную системой.

Отчет содержит реестр документов *Оказание услуги*. Двойным щелчком на поле *Документ* можно открыть исходный документ.



**Задание 6.2. Выбор данных из двух таблиц.** В соответствии с приведенным описанием создать и настроить отчет Рейтинг услуг.

**Шаг 1.** Левое соединение таблиц. Отчет *Рейтинг* услуг будет содержать информацию о том, выполнение каких услуг принесло фирме наибольшую прибыль в указанном периоде.



Документ	Склад	Мастер	Клиент
Оказание услуги 000000002 от 26.08.2010 15:34:42	Основной	Гусаков Николай Дмитриевич	Спиридонова Галина
Оказание услуги 000000003 от 26.08.2010 15:36:24	Основной	Симонов Валерий Михайлович	Роман
Оказание услуги 000000001 от 25.08.2010 16:51:05	Основной	Деловой Иван Сергеевич	Иванов Михаил Юрьевич

Рис. 6.4. Отчет *Список оказанных услуг*

Добавим новый объект конфигурации *Отчет*. Назовем его *РейтингУслуг* и запустим *конструктор схемы компоновки данных*. Добавим новый *Набор данных – запрос* и вызовем *конструктор запроса*.

Создадим запрос для набора данных, используя левое соединение двух таблиц.

В качестве источника данных для запроса выберем объектную (ссылочную) таблицу *Номенклатура* и виртуальную таблицу регистра накопления *Продажи.Обороты*.

Чтобы исключить неоднозначность имен в запросе, переименуем таблицу *Номенклатура* в *спрНоменклатура*. Для этого выделим ее в списке *Таблицы*, в контекстном меню выберем *Переименовать таблицу* (рис. 6.5).

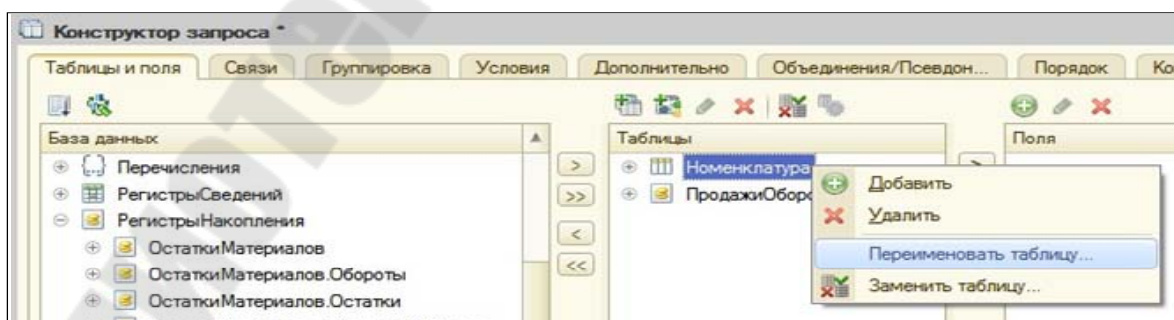


Рис. 6.5. Переименование таблицы *Номенклатура*

В список полей перенесем поля *спрНоменклатура.Ссылка* и *ПродажиОбороты.ВыручкаОборот* из этих таблиц.

Перейдем на закладку *Связи*. Так как в запросе участвует несколько таблиц, необходимо определить связи между ними.

По умолчанию уже будет создана связь по полю *Номенклатура*. То есть значение измерения *Номенклатура* регистра *Продажи* должно быть равно ссылке на элемент справочника *Номенклатура*.

Необходимо снять флажок *Все* у таблицы *ПродажиОбороты* и установить его там же у таблицы *спрНоменклатура*. Тем самым будет задан тип связи **Левое соединение**, т. е. в результате запроса будут включены все записи справочника *Номенклатура* и те записи регистра *Продажи*, которые удовлетворяют условию связи по полю *Номенклатура* (рис. 6.6).

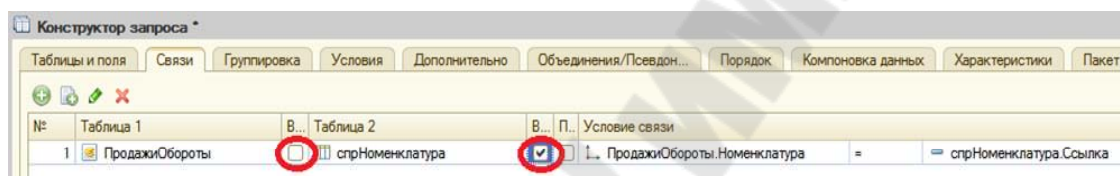


Рис. 6.6. Установка левого соединения таблиц

**Шаг 2.** Определение условия отбора записей. Перейдем на закладку *Условия* и установим отбор, чтобы группы справочника *Номенклатура* не попадали в отчет. Для этого раскроем таблицу *спрНоменклатура*, перетянем мышью поле *ЭтоГруппа* в список условий, установим флажок *Произвольное* и запишем в поле *Условие* следующий текст:

спрНоменклатура.ЭтоГруппа = ЛОЖЬ
----------------------------------

Тем самым было указано, что из базы данных нужно выбрать только те записи справочника *Номенклатура*, которые не являются группами.

Создадим второе условие, что выбранный элемент является **услугой**. Это – Простое условие. Для его создания перетянем поле *ВидНоменклатуры* в список условий. Сформируется условие, когда вид номенклатуры должен быть равен значению параметра *ВидНоменклатуры*. В дальнейшем перед выполнением запроса будет осуществляться передача в параметр *ВидНоменклатуры* значения перечисления – *Услуга*.

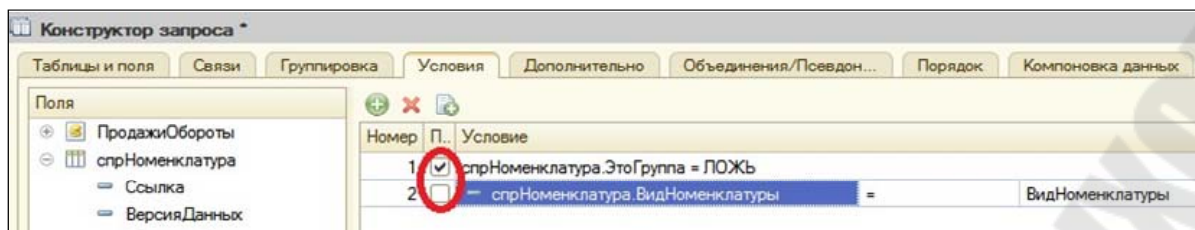


Рис. 6.7. Настройка условия отбора услуг

**Шаг 3.** Определение псевдонима полей и порядка вывода записей. Перейдем на закладку **Объединения/Псевдонимы** и укажем, что поле **Ссылка** будет иметь псевдоним **Услуга**, а поле регистра будет иметь псевдоним **Выручка** (рис. 6.8).

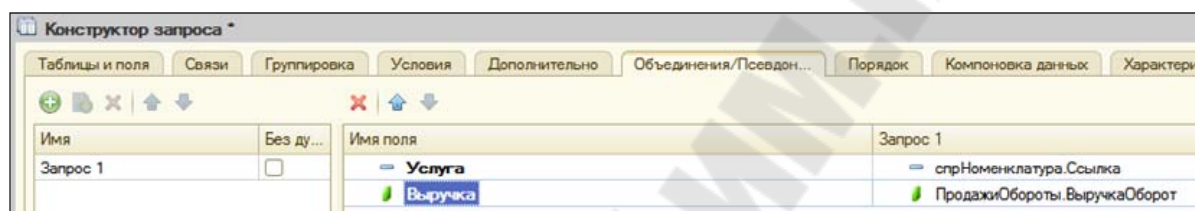


Рис. 6.8. Настройка псевдонимов полей

Перейдем на закладку **Порядок** и укажем, что результат запроса должен быть отсортирован по убыванию значения поля **Выручка**.

Создание запроса закончено, нажимаем **ОК** и возвращаемся в **конструктор схемы компоновки данных**.

Текст запроса, сформированный платформой, примет вид:

```

ВЫБРАТЬ
    спрНоменклатура.Ссылка КАК Услуга,
    ПродажиОбороты.ВыручкаОборот КАК Выручка
ИЗ
    Справочник.Номенклатура КАК спрНоменклатура
        ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты
        ПО ПродажиОбороты.Номенклатура = спрНоменклатура.Ссылка
ГДЕ
    спрНоменклатура.ЭтоГруппа = ЛОЖЬ
И спрНоменклатура.ВидНоменклатуры = &ВидНоменклатуры
УПОРЯДОЧИТЬ ПО
    Выручка УБЫВ
    
```

**Шаг 4.** Настройка вычисления итогов. Также в отчете требуется выводить итоговые значения выручки для каждой услуги. Для этого необходимо определить поля ресурсов отчета.

Ресурсы в системе компоновки – *поля*, значения которых рассчитываются на основании детальных записей, входящих в группировку. По сути, ресурсы являются групповыми или общими итогами отчета.

Итоговые данные формируются на закладке **Ресурсы**. Перейдем туда и укажем все доступные ресурсы, по которым можно вычислять итоги. Для отчета нам требуется единственный ресурс **Выручка**. По умолчанию рассчитывается сумма этого поля (рис. 6.9).

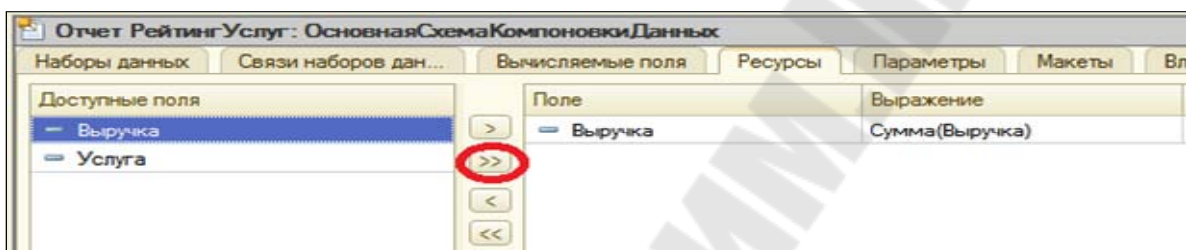


Рис. 6.9. Настройка вычисления итогов по выручке

**Шаг 5.** Настройка даты. Параметры отчета задают условия отбора записей в отчет. Перейдем на закладку **Параметры**.

Здесь еще не заданы два параметра – *начало периода* и *конец периода расчета итогов*. Они задаются с точностью до секунды.

Изменим описание типа для параметра *НачалоПериода*.

Вернемся на закладку **Параметры схемы компоновки данных** и дважды щелкнем на ячейке **Тип параметра НачалоПериода**. Нажмем кнопку выбора и в нижней части окна редактирования типа данных установим **Состав даты** в значение **Дата**.

По умолчанию время в дате установлено 00:00:00, поэтому если пользователь задаст период отчета с 01.07.2011 по 14.07.2011, итоги регистра будут рассчитаны с начала дня 01.07 00:00:00 по начало дня 14.07 00:00:00. Таким образом данные за 14-е число, отличные от начала дня, в расчет не войдут.

Чтобы исключить эту ситуацию, добавим еще один параметр, в который пользователь будет вводить дату окончания. А значение параметра *КонецПериода* будем рассчитывать автоматически так, чтобы оно указывало на конец дня даты, введенной пользователем.

Для параметра *КонецПериода* установим флажок **Ограничение доступности**.

Если этот флажок не установлен, то параметр будет доступен для настройки пользователем. Если же установить, то пользователь не увидит этот параметр.

Нажимаем кнопку *Добавить*/параметр с именем *Дата Окончания*. Автоматически сформируется заголовок *Дата окончания*. Оставим его без изменений. Тип значения – *Дата*. При этом, как и для параметра *НачалоПериода*, укажем состав даты – *Дата*. Для параметра *НачалоПериода* зададим заголовок, который будет отображаться пользователю – *Дата начала*.

Перейдем к параметру *КонецПериода*. Зададим формулу вычисления значения этого параметра через **язык выражений системы компоновки**. В нем есть функция **КонецПериода()**, которая позволяет получить дату, соответствующую концу какого-либо периода (рис. 6.10).

В ячейке *Выражение* запишем следующее выражение:

КонецПериода(&ДатаОкончания,"день")

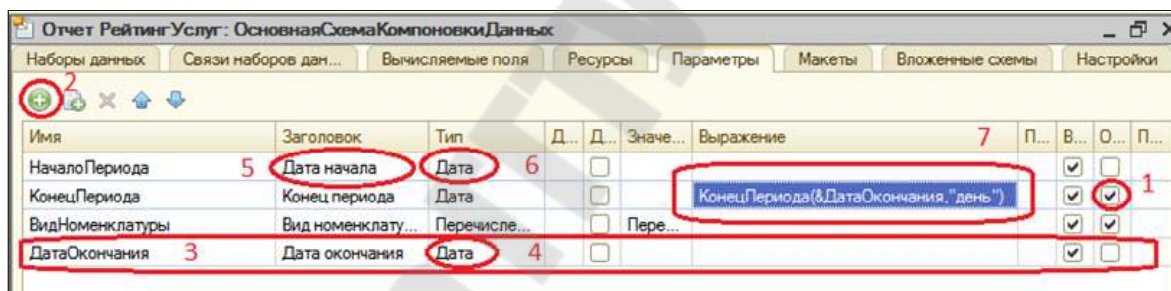


Рис. 6.10. Настройка параметров отображения даты

Настроим параметр *ВидНоменклатуры*.

Поскольку отчет должен отображать выручку, полученную только от реализации услуг, значение параметра *ВидНоменклатуры* пользователь изменять не должен. Оно должно быть задано непосредственно в схеме в компоновки как *Перечисление. ВидыНоменклатуры.Услуга*.

Воспользуемся кнопкой выбора значения и выберем из списка *Услуга* (рис. 6.11).

**Шаг 6.** Создание быстрых пользовательских настроек даты. Перейдем к формированию структуры отчета. На вкладке *Настройки* добавим группировку без указания поля группировки.

На закладке *Выбранные поля* укажем поля *Услуга* и *Выручка*.

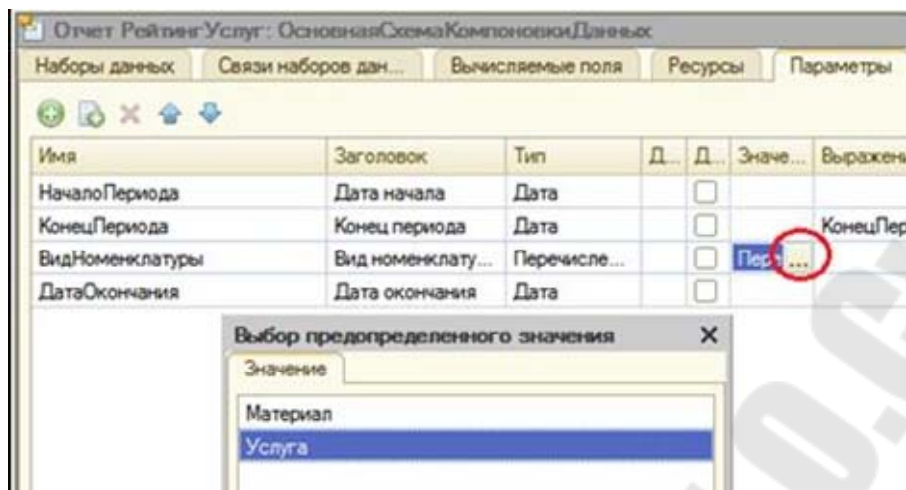


Рис. 6.11. Настройка параметра *ВидНоменклатуры*

Необходимо предоставить пользователю возможность задавать отчетный период перед формированием отчета. То есть параметры *ДатаНачала* и *ДатаОкончания* должны быть включены в состав пользовательских настроек, отображаемых в форме отчета.

На вкладке **Параметры** выделим по очереди каждый из параметров и нажмем кнопку **Свойства** элемента пользовательских настроек в правом нижнем углу окна настроек (рис. 6.12).

Установим флажок **Включать в пользовательские настройки** и оставим предложенное по умолчанию значение **Быстрый доступ**.

При этом пользователь будет иметь быстрый доступ к настройке даты начала и окончания прямо на форме отчета.

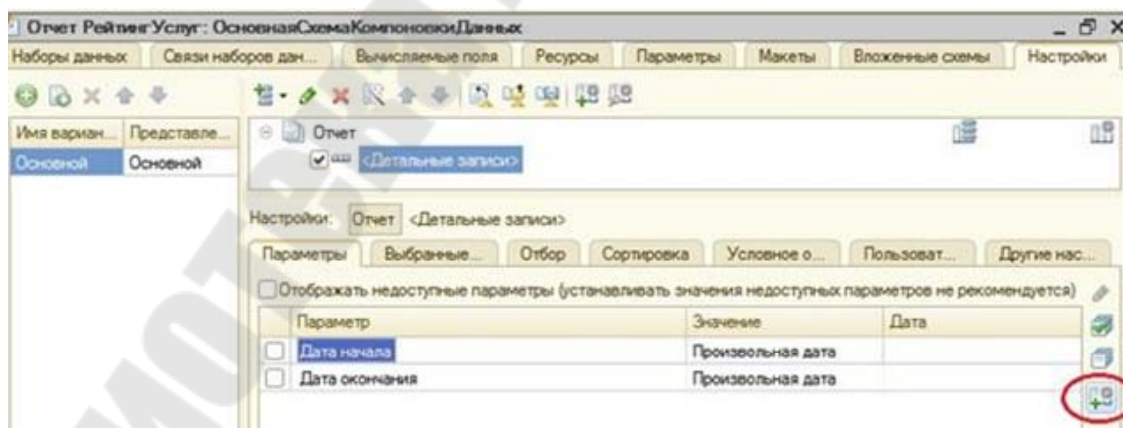


Рис. 6.12. Редактирование пользовательских настроек даты

Для улучшения интерфейса пользователя зададим для параметров *Дата Начала* и *Дата окончания* в качестве начальных значений соответственно *Начало этого месяца* и *Начало этого дня* (рис. 6.13).

Таким образом, при выполнении отчета, даты начала и окончания отчетного периода будут динамически меняться и показывать период с начала текущего месяца по сегодняшнее число и пользователю, возможно, не придется менять их вручную.

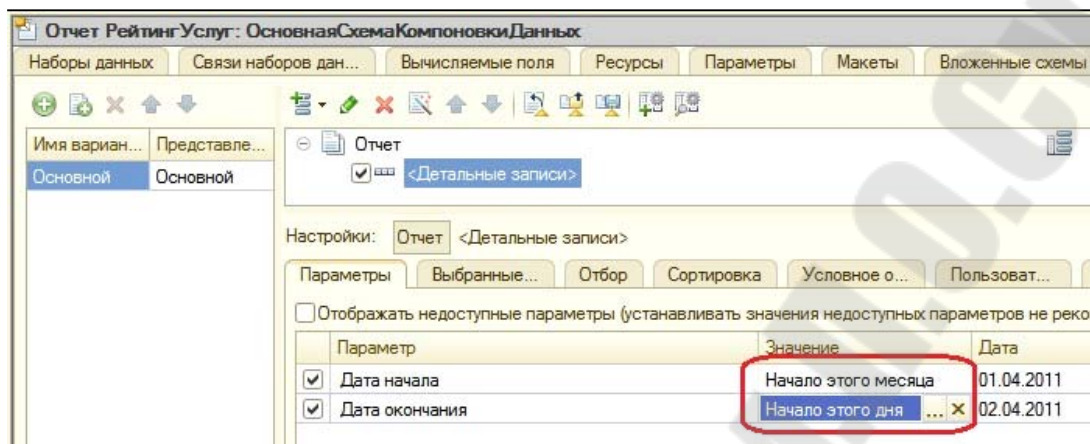


Рис. 6.13 Установка начальных значений даты

Определим подсистемы для отображения отчета. Закроем конструктор схемы компоновки и в окне редактирования объекта *Отчет РейтингУслуг* перейдем на вкладку *Подсистемы*, отметим *ОказаниеУслуг* и *Бухгалтерия*.

Запустим 1С: Предприятие в режиме отладки, вкладка *Оказание услуг*. Нажмем на *Рейтинг услуг*, *Сформировать* (рис. 6.14).

Следует учесть, что диапазон дат для отображения выручки должен попадать в диапазон заполнения документов в предыдущих работах.

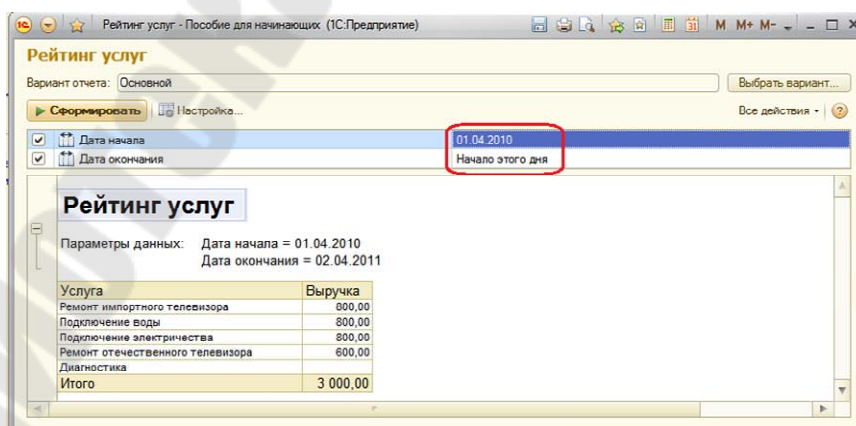


Рис. 6.14. Отчет Рейтинг услуг

Шаг 7. Определение параметров условного форматирования. Теперь на примере этого отчета покажем создание и использование других настроек отчета – *Условное оформление* и *Отбор*. В процессе настройки потребуется периодически переходить из Конфигуратора в Предприятие и обратно.

### Условное оформление

В отчете *Рейтинг услуг* удобно было бы выделять цветом записи отчета, содержащие услуги с наименьшей или с наибольшей выручкой или другим условием.

В конфигураторе откроем *схему компоновки данных* на закладке *Настройки*. В нижней части окна перейдем на закладку *Условное оформление* и нажмем кнопку *Добавить* в верхнем углу окна настроек. Укажем *Оформление – красный цвет* текста (рис. 6.15).

Затем укажем *Условие*, при наступлении которого будет применяться оформление (красный текст).

Нажмем кнопку выбора в поле *Условие* и в появившемся окне добавим *Новый элемент отбора*. Каждый элемент задает одно условие. Условий может быть несколько. Нажмем кнопку *Добавить* и укажем в графе *Левое значение* – поле *Выручка*, в графе *Вид сравнения* – *Меньше*, в *Правое значение* – *700*. Нажмите ОК.

То есть когда в поле *Выручка* окажется значение меньше 700, что-то будет выделено красным цветом текста (рис. 6.16).

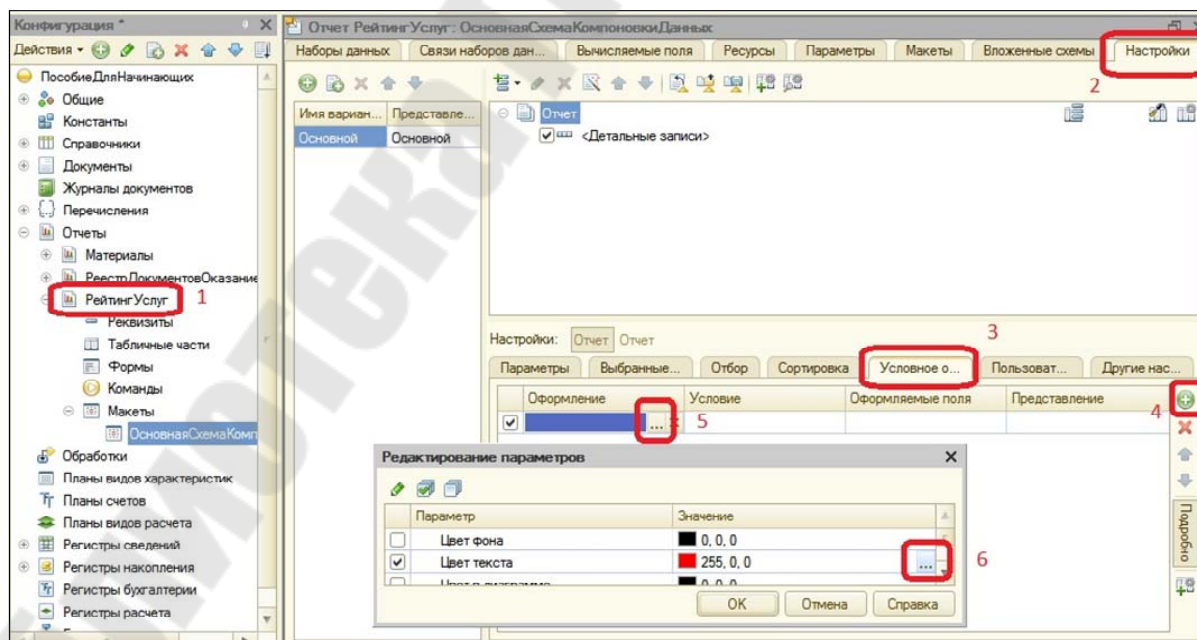


Рис. 6.15. Настройка условного форматирования



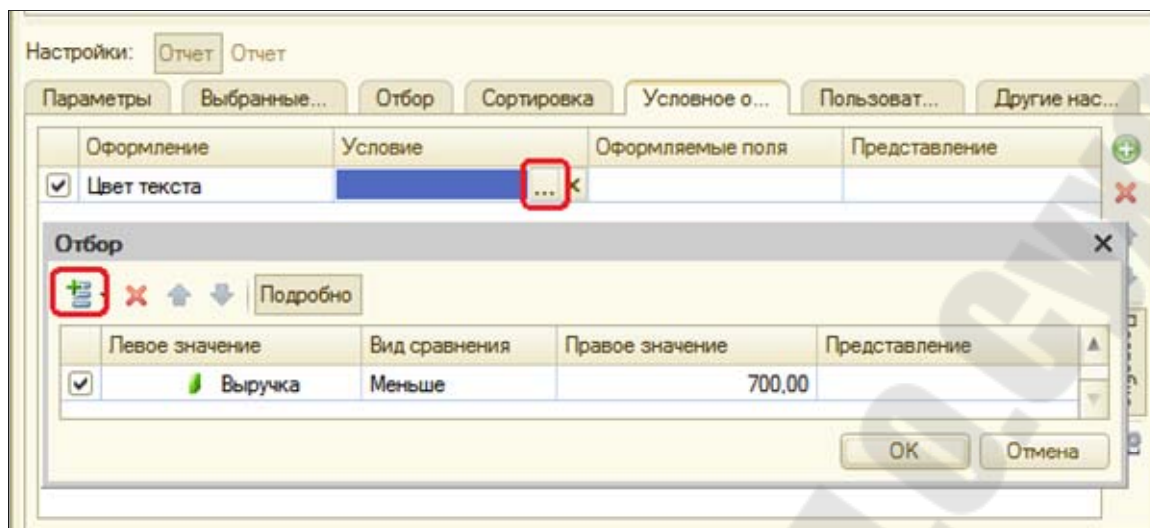


Рис. 6.16. Настройка условия

Теперь укажем это «что-то», т. е. зададим список оформляемых полей. Если требуется выделять всю строку отчета, то можно оставить этот список пустым. Или же нажать кнопку выбора в поле **Оформляемые поля** и в появившемся окне **Добавить** поля *Услуга* и *Выручка*.

В нашем случае этого можно было и не делать, так как *Услуга* и *Выручка* и есть все поля отчета. Нажимаем **ОК**.

Зададим **Представление** условного оформления как *Непопулярная услуга*. Это то, что увидит пользователь в своих настройках. То есть вместо строки «Выручка меньше 700» он увидит осмысленное выражение, которое задано в поле **Представление** (рис. 6.17).

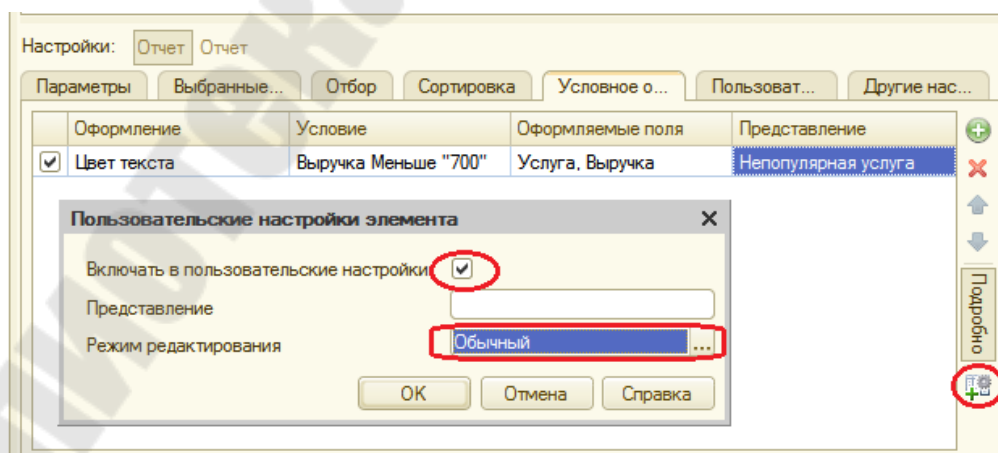


Рис. 6.17. Настройка представления

Далее следует добавить это условие в пользовательские настройки. Эти настройки расположены не в форме отчета, а вызываются нажатием кнопки **Настройка** и появляются в отдельном окне, так как эти настройки используются реже, чем настройки отчетного периода.

Перейдем в режим отладки и убедимся в выполнении условия. Зададим дату и сформируем отчет.

Данная настройка условного оформления задана жестко и пользователь может лишь включить или выключить ее. Но для более продвинутых пользователей можно предоставить более широкую свободу в использовании настроек, т. е. возможность самостоятельно задавать настройки отчета: отбор, порядок, оформление и прочие.

**Шаг 8.** Настройка отбора записей. Вернемся в конфигуратор. На закладке **Настройки схемы компоновки** нажмите кнопку **Свойства элемента пользовательских настроек** (рис. 6.18), расположенную сверху в командной панели окна настроек. Установим галочку использования для настроек **Отбор** и **Условное оформление** и свойство **Режим редактирования** – Обычный.

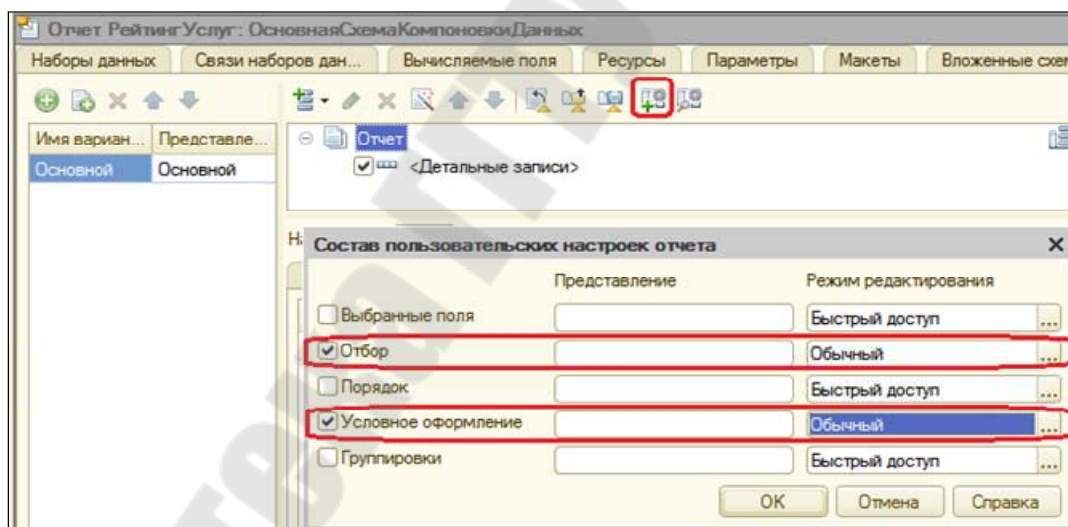


Рис. 6.18. Настройка отбора данных

Таким образом, мы включили настройки отбора и оформления в состав пользовательских настроек и предоставили пользователю возможность задавать их в отдельном окне, вызываемом кнопкой **Настройка**.

Далее следует создать настройку отбора в отчете. Для этого в нижней части окна настроек перейдем на закладку **Отбор** (рис. 6.19). Слева увидим список доступных полей отчета. Раскроем поле **Услуга**

и перенесем поле *Родитель* в список условий отбора в правой части окна. Таким образом, создается возможность отбора по группам услуг, которые пользователь может задать в режиме 1С: Предприятие.

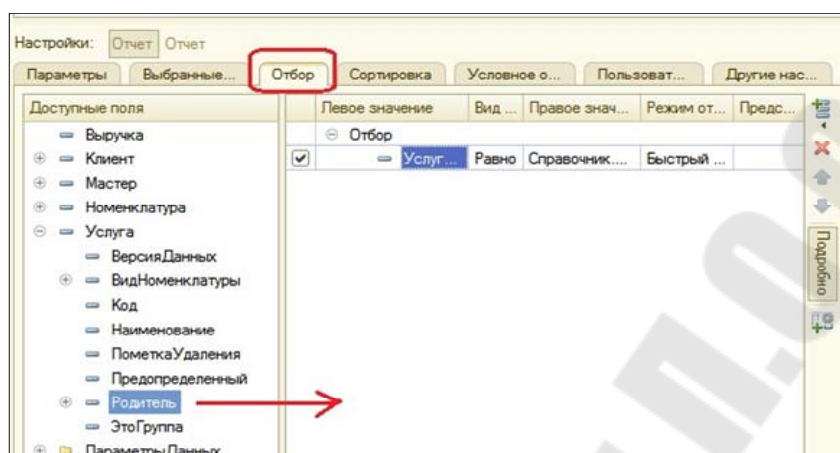


Рис. 6.19. Настройка отбора в отчете

Откроем отчет в режиме 1С: Предприятие и нажмем кнопку **Настройка**. Появились настройки *Отбор* и *Условное форматирование*.

Зададим отбор в отчете так, чтобы в него попадали только услуги, относящиеся к установке стиральных машин. Для этого следует нажать кнопку выбора в *окне пользовательских настроек* в строке *Отбор* (рис. 6.20).

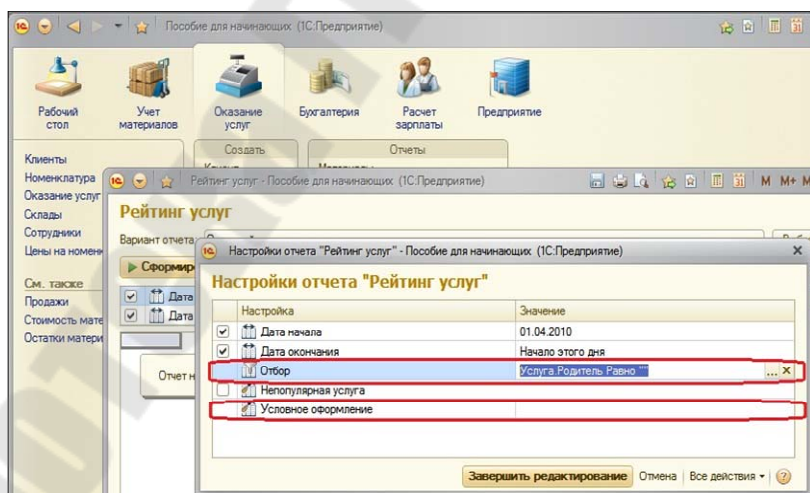



Рис. 6.20. Настройка отбора записей в отчете

В открывшемся окне **Редактирование отбора** видно созданное ранее в конфигураторе условие отбора. Нажмем в строке **Значение**, раскроем группу *Услуги*, выберем группу *Стиральные машины* из справочника *Номенклатура*.

Нажмем *Завершить редактирование* и сформируем отчет.

Вызвав окно настроек, можно очистить настройку отбора, нажав кнопку очистки  или создать ее по другому критерию, нажав кнопку выбора в строке *Отбор*.

Таким образом, пользователь может задавать многие настройки по своему желанию. Если же нет такого желания или способностей, лучше задавать эти настройки жестко, а пользователю останется включать или выключать их использование

### Задание 6.3. Использование вычисляемого поля в отчете.

В соответствии с приведенным описанием создать и настроить отчет *Рейтинг клиентов*.

Отчет *Рейтинг клиентов* будет показывать в графическом виде доход от оказания услуг каждому из клиентов за все время работы фирмы. На его примере продемонстрируем возможность использования вычисляемого поля и вывод результата в виде круговой диаграммы и гистограммы.

Шаг 1. Создание запроса для набора данных. Добавим новый отчет *РейтингКлиентов* и запустим *конструктор схемы компоновки*. Создадим новый *Набор данных – запрос* и вызовем конструктор запроса.

В качестве источника данных для запроса следует выбрать виртуальную таблицу регистра накопления *Продажи.Обороты*. Затем выберем из нее следующие поля: *ПродажиОбороты.Клиент*, *ПродажиОбороты.ВыручкаОборот*, *ПродажиОбороты.СтоимостьОборот* (рис. 6.21).

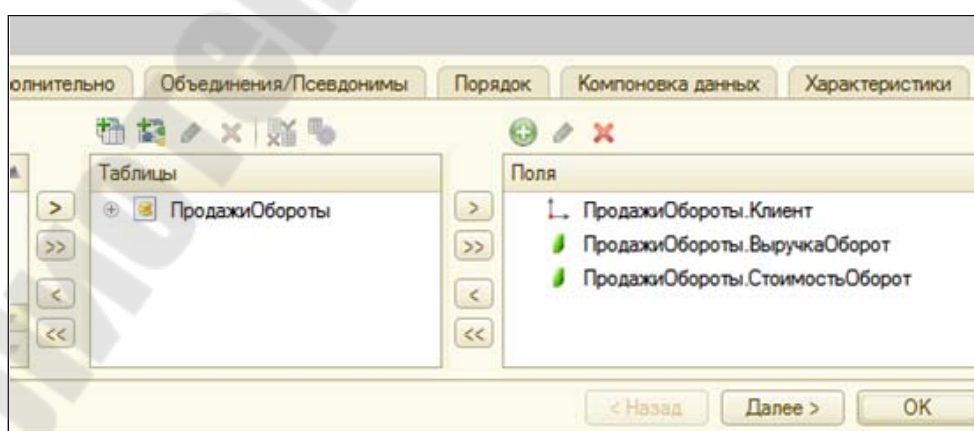


Рис. 6.21. Определение источников данных для отчета

На закладке **Объединения/Псевдонимы** укажем, что поле **ВыручкаОборот** будет иметь псевдоним **Выручка**, а поле **СтоимостьОборот** – **Стоимость**. Нажмем **ОК**.

Ничего нового в данном запросе нет, поэтому перейдем к редактированию схемы компоновки данных.

**Шаг 2.** Создание вычисляемого поля. Теперь необходимо создать дополнительное поле для отображения дохода от оказания услуг в разрезе клиентов. Для этого в системе компоновки есть возможность определения вычисляемого поля.

**Вычисляемое поле** – дополнительное поле схемы компоновки, значения которых будут вычисляться по некоторой формуле.

Перейдем на закладку **Вычисляемые поля** и нажмем **Добавить**. Зададим имя в колонке **Путь к данным** – **Доход**. В колонку **Выражение** введем следующее выражение (рис. 6.22).

Выручка – Стоимость

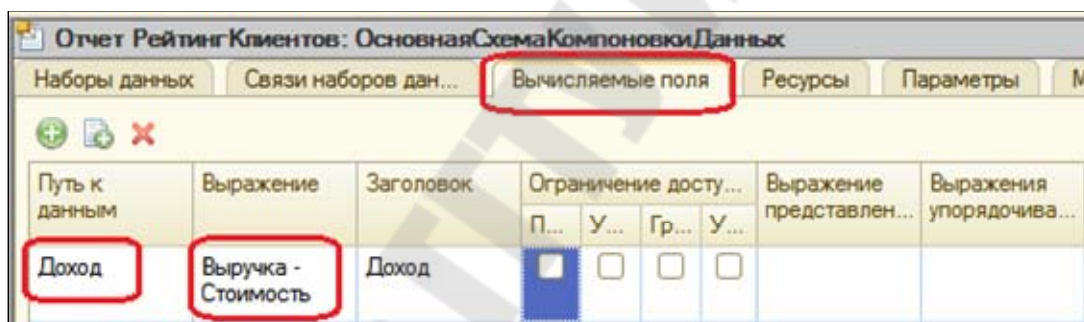


Рис. 6.22. Создание вычисляемого поля

На закладке **Ресурсы** выберем все доступные ресурсы отчета (рис. 6.23).

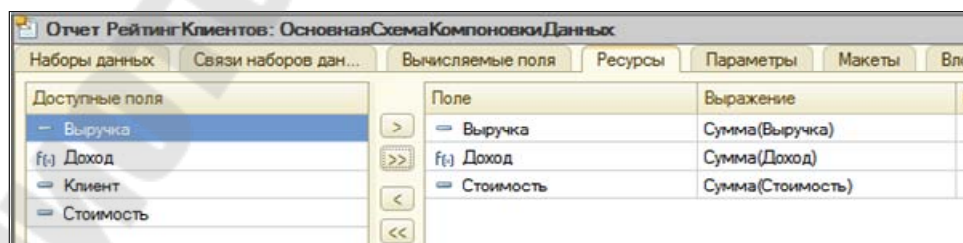


Рис. 6.23. Определение ресурсов отчета

**Шаг 3.** Добавление в структуру отчета диаграммы. На закладке **Настройки** добавим в структуру отчета диаграмму. Затем выделим

ветку **Точки** и добавим в нее группировку по полю **Клиент** (рис. 6.24, а). В значения диаграммы всегда выводится один из ресурсов отчета.

Перейдем на закладку **Выбранные поля** и выберем поле **Доход** для вывода в отчет (рис. 6.24, б).

На закладке **Другие настройки** выберем тип диаграммы – **Круговая объемная** и зададим заголовок отчета – **Рейтинг клиентов**.

Отметим в списке подсистем **Оказание услуг** и **Бухгалтерия**.

Запустим режим отладки и выполним **Рейтинг клиентов** в панели действий **Бухгалтерии**. Нажмем **Сформировать**.

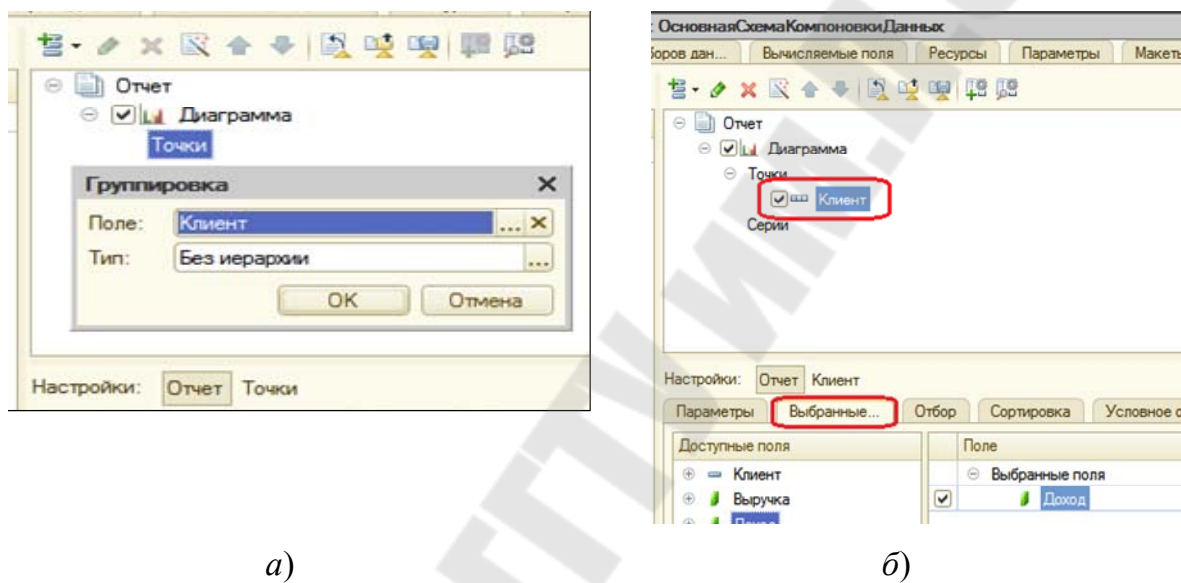


Рис. 6.24. Выбор ресурса для построения диаграммы

Можно вернуться в конфигуратор и изменить тип диаграммы на **Гистограмма объемная**. Заново сформируем отчет (рис. 6.25).

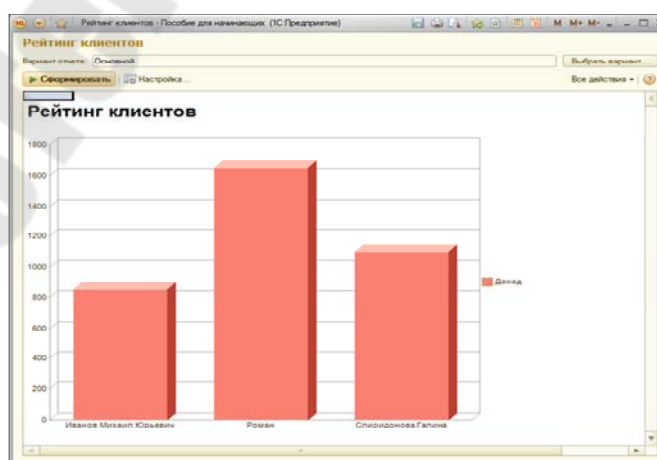


Рис. 6.25. Гистограмма Рейтинг клиентов

*Контрольные вопросы к заданиям 6.1–6.3*

1. Для чего предназначена система компоновки данных?
2. Для чего предназначены настройки компоновки данных?
3. В чем отличие между реальными и виртуальными таблицами?
4. Что является источником данных запроса?
5. Что такое псевдонимы в языке запросов?
6. Что такое параметры запроса?
7. Что такое параметры виртуальной таблицы?
8. Что такое левое соединение?
9. Как использовать конструктор запроса?
10. Как упорядочить данные в отчете?
11. Как использовать в отчете данные нескольких таблиц?
12. Как использовать группировки в структуре отчета?
13. Как создать отчет с диаграммой?
14. Что такое ресурсы в системе компоновки данных?
15. Что такое вычисляемые поля в системе компоновки данных?
16. Как создать пользовательские настройки отчета?

## ЛАБОРАТОРНАЯ РАБОТА 7 ОПТИМИЗАЦИЯ ПРОВЕДЕНИЯ ДОКУМЕНТОВ

**Задание 7.1. Повышение скорости проведения документа.**  
В соответствии с приведенным описанием скорректировать процедуру проведения документа *Оказание услуг*.

### Теоретическая справка

Есть несколько важных причин, требующих изменения обработчика события **ОбработкаПроведения** для оптимизации проведения документа *ОказаниеУслуги*, созданного в предыдущих лабораторных работах, а именно:

*во-первых*, в обращении к событию *ОбработкаПроведения* используется обращение через точку, что может сильно замедлить скорость проведения при больших объемах табличной части документа;

*во-вторых*, руководство фирмы может принять решение прекратить ручной ввод стоимости расходуемых материалов и перейти на автоматический расчет «по среднему»;

*в-третьих*, при проведении документа необходимо контролировать остатки расходуемых товаров на складе. Если товаров не хватает, то выдавать предупреждение и не проводить документ.

Поэтому в данной работе будет поставлено три цели:

- 1) повышение скорости выполнения процедуры;
- 2) автоматическое определение стоимости расходуемых материалов при проведении документа;
- 3) разделение алгоритма проведения документа на оперативный и неоперативный режимы и контроль остатков в оперативном проведении документа.

Если алгоритм проведения документа использует только те данные, которые присутствуют в реквизитах документа (и его табличных частях), вполне достаточно использовать *конструктор движений документа*.

Если же в алгоритме проведения необходимо анализировать дополнительные реквизиты объектов, ссылки на которые содержатся в документе, а также использовать результаты расчета итогов регистров, следует использовать *запросы* для более быстрой выборки.

Механизм запросов лучше «читает» информационную базу и может за один раз выбрать только нужные данные. Поэтому в типо-



вых решениях вместо объекта встроенного языка *СправочникВыборка*. *<имя>* чаще используют запросы к базе данных.

### Методические рекомендации по выполнению задания

Шаг 1. Построение запроса. Исключим конструкцию *ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры*.

В режиме *Конфигуратор* откроем модуль документа *ОказаниеУслуги*. Из процедуры обработки проведения видно, что все данные, необходимые для проведения документа, получаются из самого документа и только для определения типа номенклатуры (товар или услуга) – читаются данные всего объекта *Номенклатура*.

Обращение к объекту *Номенклатура*:

Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры
---

Это не единственные данные, которые содержатся не в самом документе и которые будут нужны для его правильного проведения.

Для оптимизации поступим следующим образом: все данные, связанные с номенклатурой, которая содержится в табличной части документа, будем получать с помощью запроса к базе данных. А данные, связанные с самим документом (дата, склад), будем по-прежнему получать из документа. Такой подход позволит читать только нужные данные и максимально ускорить проведение документа.

Запросом будем получать: *Номенклатура, Количество, Сумму, Стоимость*.

Из документа возьмем: *Дата, Клиент, Мастер, Склад*.

Установим курсор перед циклом *Если...* и из контекстного меню выберем *Конструктор запроса с обработкой результата*. Подтвердим создание нового запроса.

В окне конструктора перейдем на вкладку *Таблицы и поля* и выберем таблицу *ОказаниеУслугиПереченьНоменклатуры* – это табличная часть документа *ОказаниеУслуги*.

Из этой таблицы нужны поля – *Номенклатура, Номенклатура.ВидНоменклатуры, Количество, Сумма и Стоимость* (рис. 7.1).

Но нам нужны не все записи этой таблицы, а только те, которые относятся к документу. Поэтому перейдем на вкладку *Условия* и зададим условие отбора из таблицы только строк проводимого документа. Для этого перетянем поле *Ссылка* в список условий запроса.

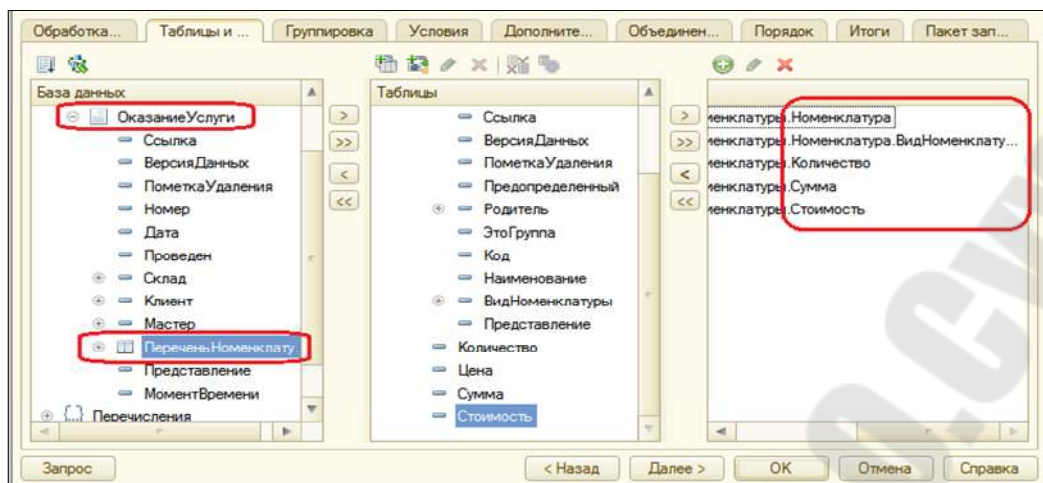


Рис. 7.1. Выбор полей из табличной части документа

В обработчике появится условие (рис. 7.2):

ОказаниеУслугиПереченьНоменклатуры.Ссылка=&Ссылка

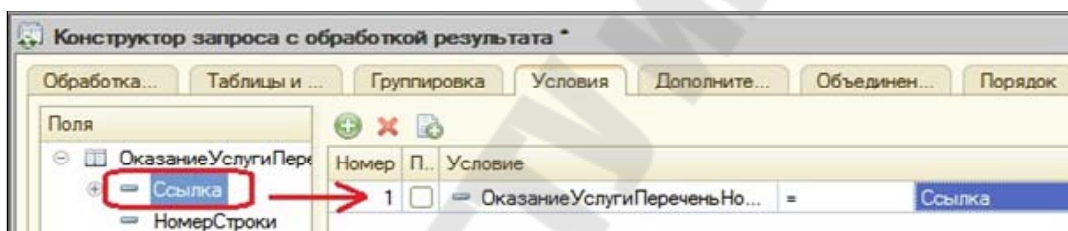


Рис. 7.2. Условие в обработчике конструктора запроса

Следует учесть, что в табличной части документа одна и та же номенклатура может встречаться несколько раз. Поэтому на закладке **Группировка** сгруппируем записи по полю *Номенклатура* и *НоменклатураВидНоменклатуры*, а рассчитывать будем сумму значений для полей *Количество* и *Сумма*. Также в состав суммируемых полей включим поле *Стоимость*. По нему будем рассчитывать, например, функцию **Максимум**.

На закладке **Объединения/Псевдонимы** зададим псевдонимы для полей *Количество* и *Сумма* – *КоличествоВДокументе* и *СуммаВДокументе*, а для поля *НоменклатураВидНоменклатуры* – *ВидНоменклатуры*.

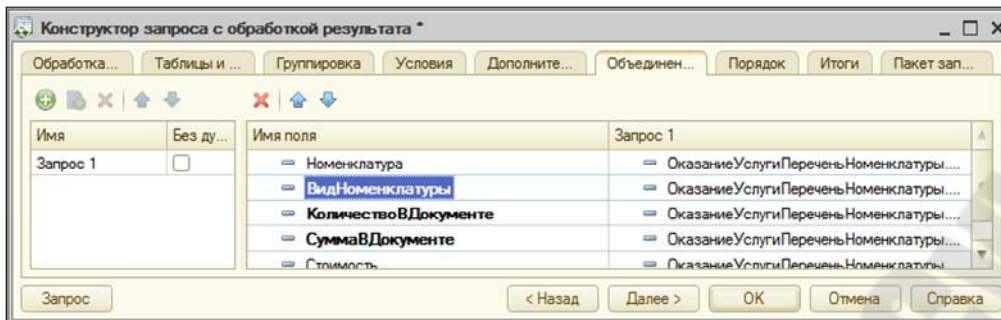


Рис. 7.3. Определение псевдонимов полей

Нажмем **ОК** и посмотрим, какой текст запроса сформирован.

```

//{{КОНСТРУКТОР_ЗАПРОСА_С_ОБРАБОТКОЙ_РЕЗУЛЬТАТА
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения будут
утрачены!!!
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
| ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
| ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры
  КАК ВидНоменклатуры,
| СУММА (ОказаниеУслугиПереченьНоменклатуры.Количество)
  КАК КоличествоВДокументе,
| МАКСИМУМ (ОказаниеУслугиПереченьНоменклатуры.Сумма)
  КАК СуммаВДокументе,
| СУММА (ОказаниеУслугиПереченьНоменклатуры.Стоимость)
  КАК Стоимость
| ИЗ
| Документ.ОказаниеУслуги.ПереченьНоменклатуры
  КАК ОказаниеУслугиПереченьНоменклатуры
| ГДЕ
| ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка
|
| СГРУППИРОВАТЬ ПО
| ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
|
ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры";
Запрос.УстановитьПараметр("Ссылка", Ссылка);
Результат = Запрос.Выполнить();
ВыборкаДетальныеЗаписи = Результат.Выбрать();
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
// Вставить обработку выборки ВыборкаДетальныеЗаписи
КонецЦикла;
//}}КОНСТРУКТОР_ЗАПРОСА_С_ОБРАБОТКОЙ_РЕЗУЛЬТАТА

```

Проанализируем текст. Для работы с запросами используется объект встроенного языка *Запрос*. Вначале создается новый объект *Запрос* и помещается в переменную *Запрос*. Затем в свойство *Текст* объекта *Запрос* помещается сам текст запроса (*Запрос.Текст=...*). После этого устанавливается значение параметра запроса *&Ссылка* как ссылка на тот документ, в модуле которого мы сейчас находимся.

```
Запрос.УстановитьПараметр("Ссылка", Ссылка);
```

Затем запрос выполняется (*Запрос.Выполнить()*), получается объект *РезультатЗапроса*, и выполняется его метод *Выбрать()*, который формирует выборку записей из результата запроса.

Получается объект *ВыборкаИзРезультатаЗапроса*, который помещается в переменную *ВыборкаДетальныеЗаписи*.

Далее, используя метод этого объекта *Следующий()* (*ВыборкаДетальныеЗаписи.Следующий()*), будем в цикле обходить выборку записей запроса. Выполняя метод выборки запроса *ВыборкаДетальныеЗаписи.Следующий()*, на каждом шаге цикла позиционируем указатель на следующую запись выборки, пока не будет достигнут конец выборки.

Чтобы в цикле получить значение какого-либо поля выборки из результата запроса, будем обращаться к полям запроса через точку от переменной *ВыборкаДетальныеЗаписи*, которая содержит текущую строку выборки запроса. Например, так: *ВыборкаДетальныеЗаписи.Номенклатура*.

Шаг 2. Корректировка движения регистров. Теперь переносим существовавшие в модуле строки, описывающие движения регистров, внутрь цикла обхода результата запроса.

```
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
// Вставить обработку выборки ВыборкаДетальныеЗаписи
КонецЦикла;
```

Новый текст в листинге будет выделен жирным текстом для удобства восприятия.

Сначала вместо комментария «//Вставить обработку выборки *ВыборкаДетальныеЗаписи*» перенесем условие проверки и весь код, формирующий движения по регистрам *ОстаткиМатериалов* и *СтоимостьМатериалов*.

```

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
Если
ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры =
Перечисления.ВидыНоменклатуры.Материал Тогда
// регистр ОстаткиМатериалов Расход
Движение = Движения.ОстаткиМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Склад = Склад;
Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;
// регистр СтоимостьМатериалов Расход
Движение = Движения.СтоимостьМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Количество* ТекСтрока-
ПереченьНоменклатуры.Стоимость;
КонецЕсли;
КонецЦикла;

```

В условии заменим *ТекСтрокаПереченьНоменклатуры.Номенклатура* на *ВыборкаДетальныеЗаписи*, так как вид номенклатуры теперь будет получаться из запроса. В движениях также заменим *ТекСтрокаПереченьНоменклатуры* на *ВыборкаДетальныеЗаписи*. Для поля *Количество* заменим его псевдоним в запросе на *КоличествоВДокументе*.

```

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =
Перечисления.ВидыНоменклатуры.Материал Тогда
// регистр ОстаткиМатериалов Расход
Движение = Движения.ОстаткиМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
Движение.Склад = Склад;
Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;
// регистр СтоимостьМатериалов Расход
Движение = Движения.СтоимостьМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
Движение.Стоимость = ВыборкаДетальныеЗаписи.КоличествоВДокументе* Выбор-
каДетальныеЗаписи.Стоимость;
КонецЕсли;
КонецЦикла;

```

```
КонецЕсли;  
КонецЦикла;
```

Теперь перенесем формирование движений по регистру *Продажи*:

```
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл  
Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =  
Перечисления.ВидыНоменклатуры.Материал Тогда  
// регистр ОстаткиМатериалов Расход  
Движение = Движения.ОстаткиМатериалов.Добавить();  
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;  
Движение.Период = Дата;  
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;  
Движение.Склад = Склад;  
Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;  
// регистр СтоимостьМатериалов Расход  
Движение = Движения.СтоимостьМатериалов.Добавить();  
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;  
Движение.Период = Дата;  
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;  
Движение.Стоимость = ВыборкаДетальныеЗаписи.КоличествоВДокументе* Выбор-  
каДетальныеЗаписи.Стоимость;  
КонецЕсли;  
// Регистр Продажи  
Движение = Движения.Продажи.Добавить();  
Движение.Период = Дата;  
Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенклатура;  
Движение.Клиент = Клиент;  
Движение.Мастер = Мастер;  
Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;  
Движение.Выручка = ТекСтрокаПереченьНоменклатуры.Сумма;  
Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Стоимость* ТекСтрока-  
ПереченьНоменклатуры.Количество;  
КонецЦикла;
```

Здесь произведем аналогичные замены. *ТекСтрокаПеречень-Номенклатуры* заменим на *ВыборкаДетальныеЗаписи*. А также поля *Сумма* и *Количество* заменим на их псевдонимы *СуммаВДокументе* и *КоличествоВДокументе*.

```
// Регистр Продажи
Движение = Движения.Продажи.Добавить();
Движение.Период = Дата;
Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;
Движение.Клиент = Клиент;
Движение.Мастер = Мастер;
Движение.Количество= ВыборкаДетальныеЗаписи.КоличествоВДокументе;
Движение.Выручка = ВыборкаДетальныеЗаписи.СуммаВДокументе;
Движение.Стоимость= ВыборкаДетальныеЗаписи.Стоимость * ВыборкаДетальныеЗаписи.КоличествоВДокументе;
КонецЦикла;
```

Оставшийся цикл обхода табличной части можно удалить:

```
Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
КонецЦикла;
```

Процедура проведения примет следующий вид:

```
Процедура ОбработкаПроведения(Отказ, Режим)
Движения.ОстаткиМатериалов.Записывать = Истина;
Движения.СтоимостьМатериалов.Записывать = Истина;
Движения.Продажи.Записывать = Истина;
//{{КОНСТРУКТОР_ЗАПРОСА_С_ОБРАБОТКОЙ_РЕЗУЛЬТАТА
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения будут
утрачены!!!
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
| ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
| ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры КАК ВидНоменклатуры,
| СУММА(ОказаниеУслугиПереченьНоменклатуры.Количество) КАК
КоличествоВДокументе,
| СУММА(ОказаниеУслугиПереченьНоменклатуры.Сумма) КАК
СуммаВДокументе,
| МАКСИМУМ(ОказаниеУслугиПереченьНоменклатуры.Стоимость) КАК Стоимость
| ИЗ
| Документ.ОказаниеУслуги.ПереченьНоменклатуры КАК
ОказаниеУслугиПереченьНоменклатуры
| ГДЕ
| ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка
| | СГРУППИРОВАТЬ ПО
| ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
```

```
| ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры";
```

```
Запрос.УстановитьПараметр("Ссылка", Ссылка);  
Результат = Запрос.Выполнить();  
ВыборкаДетальныеЗаписи = Результат.Выбрать();  
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл  
Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =  
Перечисления.ВидыНоменклатуры.Материал Тогда
```

```
// регистр ОстаткиМатериалов Расход  
Движение = Движения.ОстаткиМатериалов.Добавить();  
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;  
Движение.Период = Дата;  
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;  
Движение.Склад = Склад;  
Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;
```

```
// регистр СтоимостьМатериалов Расход  
Движение = Движения.СтоимостьМатериалов.Добавить();  
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;  
Движение.Период = Дата;  
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;  
Движение.Стоимость = ВыборкаДетальныеЗаписи.КоличествоВДокументе*  
ВыборкаДетальныеЗаписи.Стоимость;  
КонецЕсли;
```

```
// Регистр Продажи  
Движение = Движения.Продажи.Добавить(); Движение.Период = Дата;  
Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;  
Движение.Клиент = Клиент;  
Движение.Мастер = Мастер;  
Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;  
Движение.Выручка = ВыборкаДетальныеЗаписи.СуммаВДокументе;  
Движение.Стоимость = ВыборкаДетальныеЗаписи.Стоимость *  
ВыборкаДетальныеЗаписи.КоличествоВДокументе;  
КонецЦикла;  
КонецПроцедуры
```

Запустим 1С: Предприятие в режиме отладки и перепроведем документы *ОказаниеУслуги*, проверим, что ничего не изменилось.

**Задание 7.2. Автоматический расчет суммы.** В соответствии с приведенным описанием провести доработку запроса для обеспечения автоматического расчета суммы по имеющейся номенклатуре.



## Теоретическая справка

До сих пор стоимость расходуемых материалов списывалась в документ *Оказание услуги* вручную при его создании.

Теперь же будем определять стоимость номенклатуры «по среднему»: для каждой номенклатуры делить ее общую, суммарную стоимость на имеющееся количество номенклатуры, таким образом получая среднюю стоимость единицы номенклатуры.

Чтобы выполнить такой расчет, понадобятся дополнительные данные, которых сейчас нет. Для каждой номенклатуры из табличной части необходимы:

- ее стоимость, хранящаяся в регистре *СтоимостьМатериалов*;
- общее ее количество на всех складах, хранящееся в регистре *ОстаткиМатериалов*.

Поэтому необходимо доработать запрос, чтобы он получал из базы данных и эти данные тоже. Таким образом, необходимо, чтобы запрос возвращал следующие поля для каждой номенклатуры, которая есть в документе (табл. 7.1).

Таблица 7.1

### Результат выполнения запроса

Табличная часть документа				Регистр <i>Стоимость материалов</i>	Регистр <i>Остатки материалов</i>
Номенклатура	Количество в документе	Сумма в документе	Вид номенклатуры	Стоимость	Количество на всех складах

Первые четыре поля уже получают из табличной части самого документа, а последние два нужно будет получить из других таблиц.

Это значит, что запрос должен содержать два левых соединения таблицы документа с другими таблицами: **одно** – с таблицей *РегистрНакопления.СтоимостьМатериалов.Остатки*; **другое** – с таблицей *РегистрНакопления.ОстаткиМатериалов.Остатки*.

Важная деталь: в предложенной схеме виртуальные таблицы будут возвращать стоимость и остатки абсолютно для всей номенклатуры, когда нас интересует только та, которая указана в документе.

Для маленькой базы это не важно, но для реальной базы данных, где 15 000 наименований и только 5 используются в документе – это непозволительная расточительность вычислительных ресурсов.

Поэтому во все виртуальные таблицы, которые будут использоваться, нужно добавить условие отбора только номенклатуры из табличной части нашего документа. В этом случае стоимость и остатки будут рассчитаны не для всей номенклатуры вообще, а только для нужной нам.

Чтобы не получать список номенклатуры три раза (для документа и в каждой виртуальной таблице заново), можно сформировать его заранее и затем уже использовать в нужных условиях запроса.

Выполнить эту задачу помогут временные таблицы – программные объекты, которые разработчик может создать и заполнить данными, а запросы могут использовать данные временных таблиц для своих нужд.

Таким образом, схема запроса приобретает следующий вид (рис. 7.4).

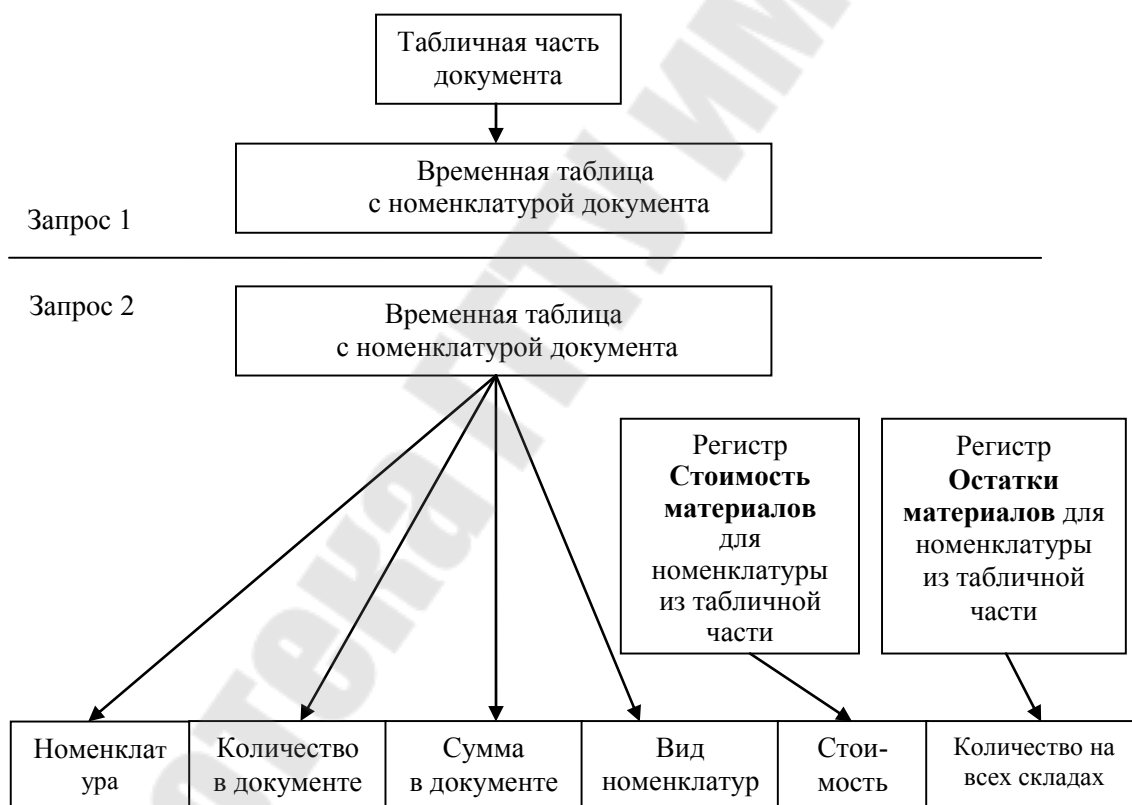


Рис. 7.4. Схема оптимизированного запроса

### Методические рекомендации по выполнению задания

Шаг 1. Удаление реквизита табличной части **Стоимость документа** *ОказаниеУслуги*. В режиме **Конфигуратор** откроем окно редактирования объекта *Документа ОказаниеУслуги*, перейдем на за-

кладку *Данные*, раскроем список реквизитов табличной части документа, выделим реквизит *Стоимость* и нажмем кнопку *Удалить* в командной панели. Хотя проще это сделать из окна конфигурации.

Также удалим поле *Стоимость* из таблицы *ПереченьНоменклатуры*, расположенной в форме.

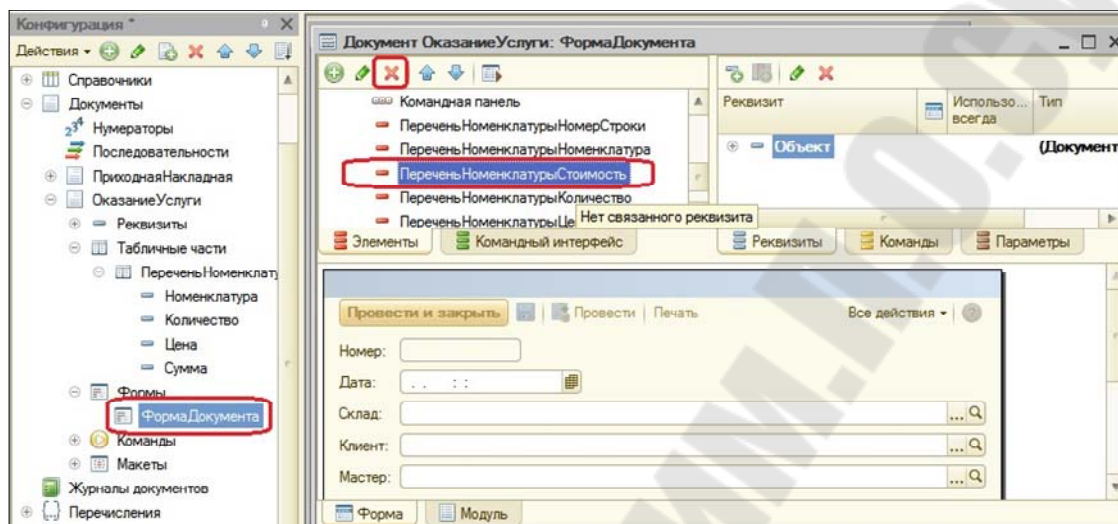


Рис. 7.5. Редактирование формы документа *ОказаниеУслуги*

Шаг 2. Временную таблицу сформируем с помощью запроса, который был написан ранее. Откроем модуль документа *ОказаниеУслуги*.

В процедуре ***ОбработкаПроведения()*** перед созданием запроса создадим менеджер временных таблиц и укажем, что этот запрос будет использовать созданный менеджер:

Процедура *ОбработкаПроведения(Отказ, Режим)*

*Движения.ОстаткиМатериалов.Записывать* = Истина;

*Движения.СтоимостьМатериалов.Записывать* = Истина;

*Движения.Продажи.Записывать* = Истина;

**//Создать менеджер временных таблиц.**

**МенеджерВТ = Новый МенеджерВременныхТаблиц;**

**Запрос = Новый Запрос;**

**//Укажем, какой менеджер временных таблиц использует этот запрос**

**Запрос.МенеджерВременныхТаблиц = МенеджерВТ;**

Запрос.Текст =

"ВЫБРАТЬ

|           *ОказаниеУслугиПереченьНоменклатуры.Номенклатура,*

Шаг 3. Теперь изменим запрос, чтобы он создавал временную таблицу, которая будет храниться в менеджере временных таблиц **МенеджерВТ**.

Чтобы конструктор запроса смог открыть запрос, необходимо удалить из него строку (поля Стоимость больше нет) и запятую в строке выше этой:

```
МАКСИМУМ(ОказаниеУслугиПереченьНоменклатуры.Стоимость) КАК Стоимость
```

Получится следующее:

```
Запрос.Текст =  
"ВЫБРАТЬ  
| ОказаниеУслугиПереченьНоменклатуры.Номенклатура,  
| ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры КАК Вид-  
Номенклатуры,  
| СУММА(ОказаниеУслугиПереченьНоменклатуры.Количество) КАК КоличествоВДо-  
кументе,  
| СУММА(ОказаниеУслугиПереченьНоменклатуры.Сумма) КАК СуммаВДокументе  
|  
|ИЗ
```

Установим курсор внутрь текста запроса (например, на слове **ВЫБРАТЬ**) и выполним команду контекстного меню **Конструктор запроса**.

Чтобы результат запроса поместить во временную таблицу, перейдем на закладку **Дополнительно** и отметим пункт **Создание временной таблицы**. Зададим ей имя – **НоменклатураДокумента**. Нажмите ОК (рис. 7.6).

Конструктор сформировал текст запроса с одной новой строкой.

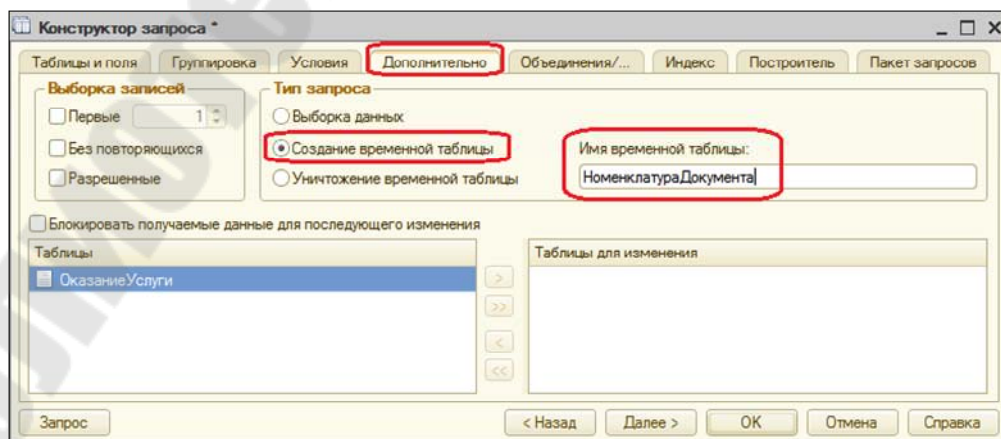


Рис. 7.6. Помещение результата запроса во временную таблицу

```
"ВЫБРАТЬ
| ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
| ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры КАК
ВидНоменклатуры,
| СУММА(ОказаниеУслугиПереченьНоменклатуры.Количество) КАК КоличествоВДо-
кументе,
| СУММА(ОказаниеУслугиПереченьНоменклатуры.Сумма) КАК СуммаВДокументе
| ПОМЕСТИТЬ НоменклатураДокумента
| ИЗ
| Документ.ОказаниеУслуги.ПереченьНоменклатуры КАК ОказаниеУслугиПеречень-
Номенклатуры
| ГДЕ
| ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка
|
| СГРУППИРОВАТЬ ПО
| ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры";
```

Результат запроса будет сохранен во временной таблице *НоменклатураДокумента*. Это был Запрос 1 на рис 7.4.

Шаг 4. Теперь если мы для другого запроса указать этот же самый менеджер временных таблиц *МенеджерВТ*, то в другом запросе можно будет обратиться к данным этой временной таблицы.

Сформируем второй запрос.

Установим курсор на следующую строку после оператора *Результат = Запрос.Выполнить()*; (именно здесь выполняется создание временной таблицы) и напишем заготовку запроса:

```
Запрос2 = Новый Запрос;
Запрос2.МенеджерВременныхТаблиц = МенеджерВТ;
Запрос2.Текст = "";
```

Таким образом создан новый объект *Запрос*, которому назначен тот же самый менеджер временных таблиц, чтобы иметь возможность обращаться к созданной ранее временной таблице.

Установим курсор **внутри кавычек** и выполним контекстную команду *Конструктор запроса*. Согласимся на создание нового запроса.

Создадим описание временной таблицы в запросе. Для этого над списком *Таблицы* нажмем **Создать описание временной таблицы**.

Введем имя временной таблицы *НоменклатураДокумента* и добавим поля (рис. 7.7):

- *Номенклатура*, тип СправочникСсылка.Номенклатура;
- *ВидНоменклатуры*, тип ПеречислениеСсылка. ВидыНоменклатуры;
- *КоличествоВДокументе*, тип Число, 15, 3;
- *СуммаВДокументе*, тип Число, 15, 2. Нажмите ОК.

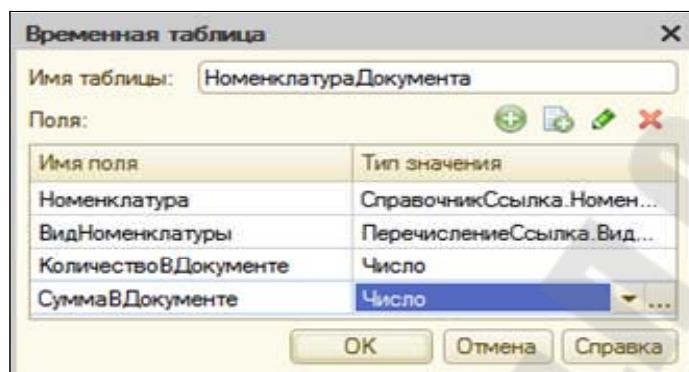


Рис. 7.7. Создание временной таблицы *НоменклатураДокумента*

Выберем из *НоменклатураДокумента* все поля и нажмем кнопку *Запрос*.

ВЫБРАТЬ  
 НоменклатураДокумента.Номенклатура,  
 НоменклатураДокумента.ВидНоменклатуры,  
 НоменклатураДокумента.КоличествоВДокументе,  
 НоменклатураДокумента.СуммаВДокументе  
 ИЗ  
 НоменклатураДокумента КАК НоменклатураДокумента

Таким образом, создана первая часть второго запроса – выбрана информация из временной таблицы. Теперь будем соединять эту конструкцию левыми соединениями с таблицами остатков.

Шаг 5. Добавим в список таблиц запроса виртуальную таблицу *РегистрНакопления.СтоимостьМатериалов.Остатки*. Из нее выберем поле *СтоимостьОстаток*. Перейдем на закладку *Связи* и зададим связь между таблицами.

Из временной таблицы будем выбирать **все** записи, и поле *Номенклатура* временной таблицы должно быть равно полю *Материал* таблицы остатков (рис. 7.8).

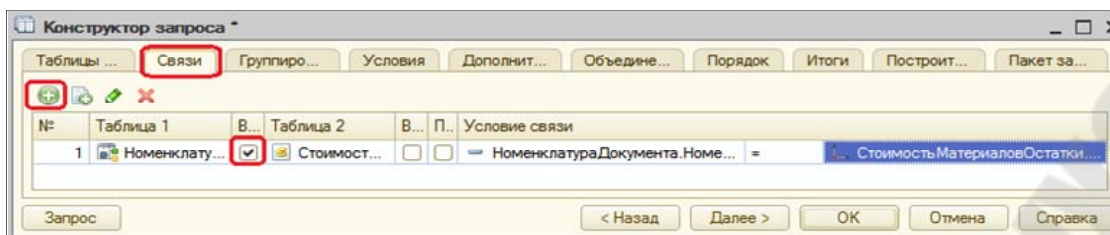


Рис. 7.8. Создание запроса на построение связи между таблицами

Ограничим виртуальную таблицу только нужной номенклатурой. Для этого вернемся на вкладку **Таблицы и поля**, выделим в списке таблицу *СтоимостьМатериаловОстатки* и нажмем кнопку **Параметры виртуальной таблицы** (рис. 7.9).

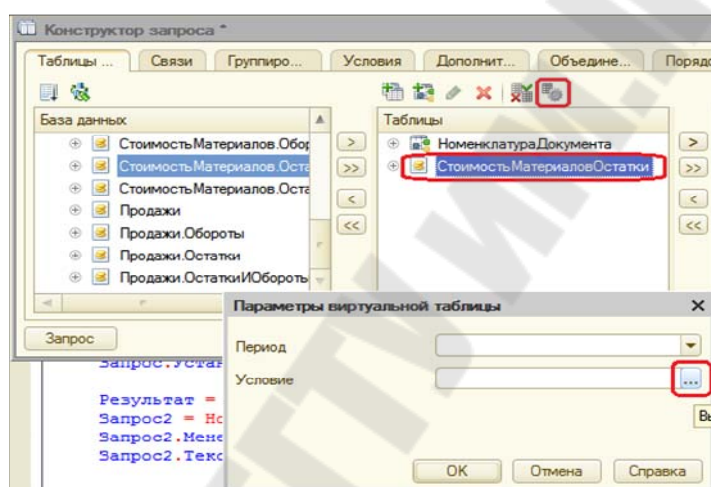


Рис. 7.9. Настройка параметров виртуальной таблицы

Зададим условие:

Материал В (ВЫБРАТЬ НоменклатураДокумента.Номенклатура ИЗ НоменклатураДокумента)

Это значит, что материал должен быть среди номенклатуры, выбранной из временной таблицы. Нажмем **ОК**.

Теперь нажмем кнопку **Запрос** и посмотрим на текст, сформированный конструктором.

ВЫБРАТЬ  
 НоменклатураДокумента.Номенклатура,  
 НоменклатураДокумента.ВидНоменклатуры,  
 НоменклатураДокумента.КоличествоВДокументе,  
 НоменклатураДокумента.СуммаВДокументе,  
 СтоимостьМатериаловОстатки.СтоимостьОстаток

ИЗ

НоменклатураДокумента КАК НоменклатураДокумента

ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.СтоимостьМатериалов.Остатки(,

Материал В

(ВЫБРАТЬ

НоменклатураДокумента.Номенклатура

ИЗ

НоменклатураДокумента)) КАК СтоимостьМатериаловОстатки

ПО НоменклатураДокумента.Номенклатура = СтоимостьМатериаловОстатки.Материал

Тем самым мы добавили к выбранным ранее полям стоимость номенклатуры.

Шаг 6. Теперь добавим виртуальную таблицу остатков регистра *ОстаткиМатериалов.Остатки*, из которой выберем поле *Количество.Остаток* (рис. 7.10).

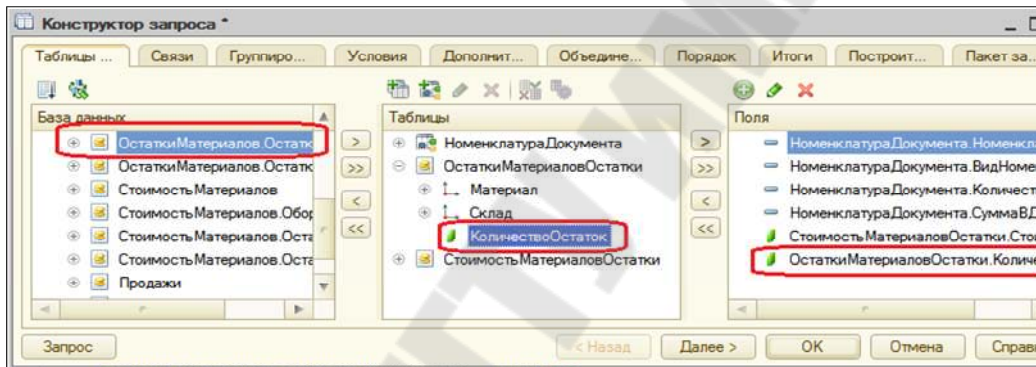


Рис. 7.10. Выбор виртуальной таблицы для создания запроса

Перейдем на закладку **Связи** и зададим связь между таблицами.

Из временной таблицы будем выбирать все записи, поле *Номенклатура* временной таблицы должно быть равно полю *Материал* таблицы остатков (рис. 7.11).

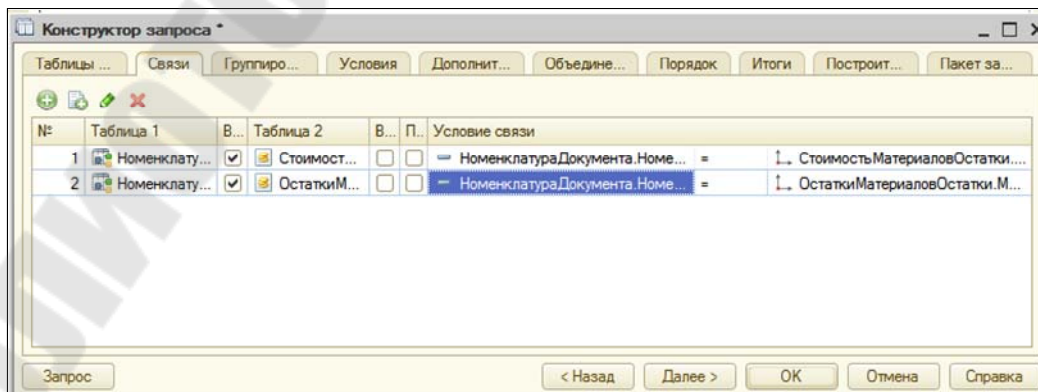


Рис. 7.11. Настройка условий связи таблиц



Также зададим параметры виртуальной таблицы *ОстаткиМатериаловОстатки*. В параметр *Условие* введем:

Материал В (ВЫБРАТЬ НоменклатураДокумента.Номенклатура ИЗ НоменклатураДокумента)

Тем самым мы добавили к выбранным ранее полям остатки номенклатуры на всех складах.

В заключение перейдем на закладку *Объединения/Псевдонимы* и зададим следующие псевдонимы полей:

*СтоимостьОстаток* – Стоимость;

*КоличествоОстаток* – Количество.

В результате получим следующий текст запроса:

```
ВЫБРАТЬ
НоменклатураДокумента.Номенклатура,
НоменклатураДокумента.ВидНоменклатуры,
НоменклатураДокумента.КоличествоВДокументе,
НоменклатураДокумента.СуммаВДокументе,
СтоимостьМатериаловОстатки.СтоимостьОстаток КАК Стоимость,
ОстаткиМатериаловОстатки.КоличествоОстаток КАК Количество
ИЗ
НоменклатураДокумента КАК НоменклатураДокумента
    ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.СтоимостьМатериалов.Остатки(
        ,
        Материал В
            (ВЫБРАТЬ
                НоменклатураДокумента.Номенклатура
                ИЗ
                НоменклатураДокумента)) КАК
    СтоимостьМатериаловОстатки
        ПО НоменклатураДокумента.Номенклатура =
    СтоимостьМатериаловОстатки.Материал
        ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ОстаткиМатериалов.Остатки(
            ,
            Материал В
                (ВЫБРАТЬ
                    НоменклатураДокумента.Номенклатура
                    ИЗ
                    НоменклатураДокумента)) КАК
        ОстаткиМатериаловОстатки
            ПО НоменклатураДокумента.Номенклатура =
    ОстаткиМатериаловОстатки.Материал
```

Шаг 7. Нужно предусмотреть тот случай, когда номенклатура в справочнике есть, но у нее нет ни остатков, ни стоимости. Это может быть, например, когда номенклатуру создали в справочнике, но она еще не поступала на фирму.

В такой ситуации левые соединения с виртуальными таблицами не вернут ничего. На языке запросов это значит, что в полях *Стоимость* и *Количество* будут значения *NULL*.

Избавимся в самом запросе от этих значений. Для этого применим функцию **ЕСТЬNULL()** к полям *Стоимость* и *Количество*. Если значение этого поля будет *NULL*, функция вернет 0. В остальных случаях функция вернет само значение этого поля.

Перейдем на закладку **Таблицы и поля**, выделим *СтоимостьМатериаловОстатки.СтоимостьОстаток* и нажмем кнопку **Изменить текущий элемент**.

Отредактируем значение поля (рис. 7.12):

```
ЕСТЬNULL(СтоимостьМатериаловОстатки.СтоимостьОстаток,0)
```

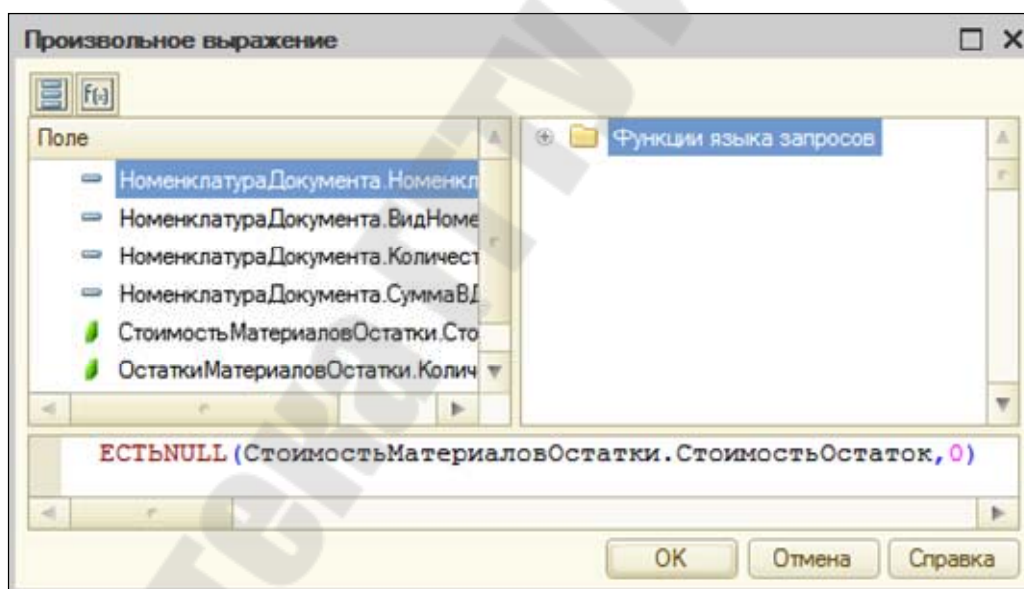


Рис. 7.12. Редактирование значения поля

Аналогично с другим полем – *ОстаткиМатериаловОстатки.КоличествоОстаток*.

```
ЕСТЬNULL(ОстаткиМатериаловОстатки.КоличествоОстаток,0)
```

Нажмем **OK** – текст запроса будет вставлен в модуль. Останется всего лишь дописать после него оператор выполнения запроса:

```

Запрос2 = Новый Запрос;
Запрос2.МенеджерВременныхТаблиц = МенеджерВТ;
Запрос2.Текст = "ВЫБРАТЬ
| НоменклатураДокумента.Номенклатура,
| НоменклатураДокумента.ВидНоменклатуры,
| НоменклатураДокумента.КоличествоВДокументе,
| НоменклатураДокумента.СуммаВДокументе,
| ЕСТЬNULL(СтоимостьМатериаловОстатки.СтоимостьОстаток, 0) КАК Стоимость,
| ЕСТЬNULL(ОстаткиМатериаловОстатки.КоличествоОстаток, 0) КАК Количество
| ИЗ
| НоменклатураДокумента КАК НоменклатураДокумента
|     ЛЕВОЕ СОЕДИНЕНИЕ
|         РегистрНакопления.СтоимостьМатериалов.Остатки(
|
|             ,
|             Материал В
|             (ВЫБРАТЬ
|             НоменклатураДокумента.Номенклатура
|             ИЗ
|                 НоменклатураДокумента)) КАК СтоимостьМатериаловОстатки
|     ПО НоменклатураДокумента.Номенклатура = СтоимостьМатериаловОстат-
ки.Материал
|     ЛЕВОЕ СОЕДИНЕНИЕ
|         РегистрНакопления.ОстаткиМатериалов.Остатки(
|
|             ,
|             Материал В
|             (ВЫБРАТЬ
|             |             НоменклатураДокумента.Номенклатура
|             ИЗ
|                 НоменклатураДокумента)) КАК ОстаткиМатериаловОстатки
|     ПО НоменклатураДокумента.Номенклатура = ОстаткиМатериаловОстат-
ки.Материал";
Результат = Запрос2.Выполнить();
ВыборкаДетальныеЗаписи = Результат.Выбрать();

```

Шаг 8. Теперь разберемся с записью движений. Все операторы, которые были написаны ранее, будут работать без изменений.

Единственное, что потребуется изменить, – способ получения стоимости.

Раньше мы просто брали ее из документа, теперь же нужно ее рассчитать на основании данных запроса. Стоимость материала равна частному от деления всей стоимости, полученной запросом (*Стоимость*) на общее количество материала на всех складах (*Количество*).

Но, как изложено выше, *Количество* может быть равно 0, а на ноль делить нельзя. Поэтому сразу после начала цикла обхода результата запроса рассчитаем стоимость для текущей номенклатуры.

```
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
Если ВыборкаДетальныеЗаписи.Количество = 0 Тогда
    СтоимостьМатериала = 0;
Иначе
    СтоимостьМатериала =
        ВыборкаДетальныеЗаписи.Стоимость/ВыборкаДетальныеЗаписи.Количество;
КонецЕсли;
Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =
    Перечисления.ВидыНоменклатуры.Материал Тогда
```

Теперь заменим расчет стоимости в движениях регистров *СтоимостьМатериалов* и *Продажи*.

```
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
Если ВыборкаДетальныеЗаписи.Количество = 0 Тогда
    СтоимостьМатериала = 0;
Иначе
    СтоимостьМатериала =
        ВыборкаДетальныеЗаписи.Стоимость/ВыборкаДетальныеЗаписи.Количество;
КонецЕсли;
Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =
    Перечисления.ВидыНоменклатуры.Материал Тогда

// регистр ОстаткиМатериалов Расход
    Движение = Движения.ОстаткиМатериалов.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
    Движение.Склад = Склад;
    Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;

// регистр СтоимостьМатериалов Расход
    Движение = Движения.СтоимостьМатериалов.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
    Движение.Стоимость =
        ВыборкаДетальныеЗаписи.КоличествоВДокументе*СтоимостьМатериала;
КонецЕсли;

// Регистр Продажи
```

```
Движение = Движения.Продажи.Добавить();
Движение.Период = Дата;
Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;
Движение.Клиент = Клиент;
Движение.Мастер = Мастер;
Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;
Движение.Выручка = ВыборкаДетальныеЗаписи.СуммаВДокументе;
Движение.Стоимость =
СтоимостьМатериала * ВыборкаДетальныеЗаписи.КоличествоВДокументе;
КонецЦикла;
```

Шаг 9. Теперь, если проводить этот документ в первый раз, результат получится правильный. Но, если документ был проведен ранее и мы заново решим его провести, получим неправильный результат. Дело в том, что когда мы находимся в обработчике проведения и этот документ был проведен ранее, то в базе данных существуют движения этого документа.

Таким образом, читая из базы стоимость и остатки материалов, они будут прочитаны с учетом движений документа. А это неправильно.

Чтобы в обработчике проведения документа прочитать данные без учета предыдущих движений, которые мог выполнять документ, нужно перед чтением записать пустые наборы записей в те регистры, из которых будет проводиться чтение. В нашем случае это *СтоимостьМатериалов* и *ОстаткиМатериалов*.

Перед выполнением второго запроса добавим строки:

```
| ПО НоменклатураДокумента.Номенклатура =
ОстаткиМатериаловОстатки.Материал";
//Запишем пустые наборы записей, чтобы читать остатки без учета данных
в документе.
Движения.СтоимостьМатериалов.Записать();
Движения.ОстаткиМатериалов.Записать();

Результат = Запрос2.Выполнить();
```

Запустите режим отладки и перепроведите все документы *Оказание услуги*, проверьте правильность занесения данных в регистры.

**Задание 7.3. Контроль остатков.** В соответствии с приведенным описанием скорректировать процедуру контроля остатков материалов на складах при проведении документа *Оказание услуги*.

### Методические рекомендации по выполнению задания

Общая методика контроля остатков при проведении документа заключается в записи движения документа, а затем чтения из базы остатков. Если появились отрицательные остатки, значит, такой документ проводить нельзя. Нужно сообщить пользователю, каких материалов не хватает и отменить проведение документа. Осталось проконтролировать это.

Шаг 1. В режиме *Конфигуратор* сделаем заготовку. После цикла обхода результата запроса и перед концом процедуры напишем:

```
КонецЦикла;  
Движения.Записать();  
Если Режим = РежимПроведенияДокумента.Оперативный Тогда  
//проверить отрицательные остатки.  
КонецЕсли;  
КонецПроцедуры
```

Сначала записываем движения в регистры. Затем определяем режим проведения документа. При выполнении процедуры *ОбработкаПроведения()* вторым параметром (*Режим*) в нее передается режим проведения документа и значение этой переменной сравнивается со значением системного перечисления *РежимПроведенияДокумента*. В случае оперативного проведения будем выполнять контроль остатков.

Шаг 2. Теперь сделаем заготовку запроса для проверки отрицательных остатков. Используем тот же менеджер виртуальных таблиц.

```
Если Режим = РежимПроведенияДокумента.Оперативный Тогда  
//проверить отрицательные остатки.  
Запрос3 = Новый Запрос;  
Запрос3.МенеджерВременныхТаблиц = МенеджерВТ;  
Запрос3.Текст = "";  
КонецЕсли;
```

Установим курсор внутрь кавычек и вызовем конструктор запроса. Подтвердим создание нового запроса.

Выберем таблицу *ОстаткиМатериалов.Остатки* и из нее два поля: *Материал* и *КоличествоОстаток*. Зададим параметры этой таблице. В параметре *Условие* напишем:

Материал В (ВЫБРАТЬ НоменклатураДокумента.Номенклатура ИЗ НоменклатураДокумента) И Склад = &Склад

Таким образом, получаем итоги только для той номенклатуры, которая содержится во временной таблице и только по складу, который указан в документе.

Шаг 3. Затем на вкладке *Условия* перенесем в список условий поле *ОстаткиМатериалов.Остатки.КоличествоОстаток*, поставим флажок *Произвольное* и укажем, что нас интересуют только отрицательные остатки (рис. 7.13).

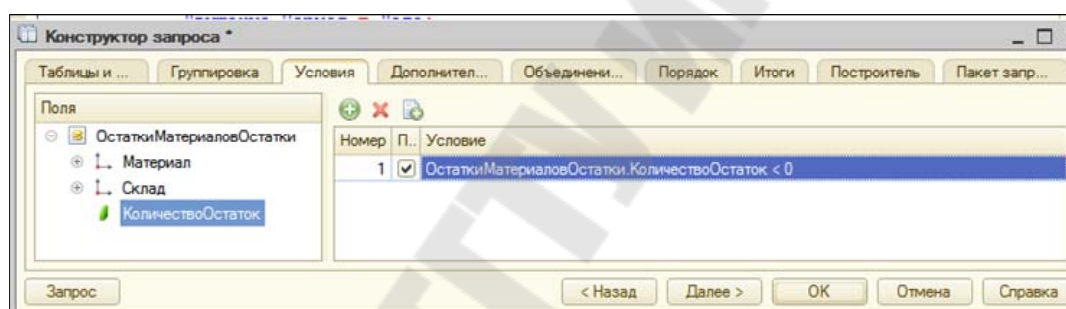


Рис. 7.13. Условие на отбор отрицательных остатков

Нажмем **ОК**.

Шаг 4. Теперь осталось только установить параметр запроса, обойти результат запроса и вывести сообщения об отрицательных остатках.

```
//проверить отрицательные остатки.  
Запрос3 = Новый Запрос;  
    Запрос3.МенеджерВременныхТаблиц = МенеджерВТ;  
    Запрос3.Текст = "ВЫБРАТЬ  
    |     ОстаткиМатериалов.Остатки.Материал,  
    |     ОстаткиМатериалов.Остатки.КоличествоОстаток  
    |ИЗ  
    |     РегистрНакопления.ОстаткиМатериалов.Остатки(  
    |     ,  
    |     Материал В  
    |     (ВЫБРАТЬ
```

```

| НоменклатураДокумента.Номенклатура
|   ИЗ
|   НоменклатураДокумента)
|   И   Склад =   &Склад)   КАК ОстаткиМатериаловОстатки
|ГДЕ
|   ОстаткиМатериаловОстатки.КоличествоОстаток < 0";

```

```

Запрос3.УстановитьПараметр("Склад", Склад);
Результат = Запрос3.Выполнить();
ВыборкаДетальныеЗаписи = Результат.Выбрать();
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    Сообщение = Новый СообщениеПользователю();
Сообщение.Текст = "Не хватает" + Строка(- ВыборкаДетальныеЗаписи.КоличествоОстаток) + "единиц материала" + ВыборкаДетальныеЗаписи.Материал + "";
    Сообщение.Сообщить();
    Отказ = Истина;
КонецЦикла;
КонецЕсли;
КонецПроцедуры

```

При выполнении проверки в запрос в параметре *Склад* передается склад, указанный в документе. Затем выполняется запрос для получения отрицательных остатков номенклатуры, содержащейся во временной таблице и на складе, указанном в параметре *Склад*.

После этого выборка записей запроса обходится в цикле, и если есть такие записи, они выводятся в сообщениях пользователю.

При этом параметру *Отказ* присваивается значение *Истина*, т. е. документ не проводится, начатая транзакция отменяется, состояние данных, измененных в процессе проведения, возвращается в исходное, как до начала проведения документа.

Шаг 5. Блокировка данных, которые читаются и изменяются при проведении. Сейчас схема нашей процедуры такова:

1. Выполняем первый запрос с именем *Запрос*. В результате формируется временная таблица из перечня номенклатуры документа.

2. Выполняем второй запрос с именем *Запрос2*. В результате читаем стоимость и остатки для номенклатуры, содержащейся в табличной части документа.

3. Записываем движения регистров (*Движения.Записать()*)

4. Выполняем третий запрос *Запрос3*. Тем самым проверяем наличие отрицательных остатков.

Обратите внимание, что, начиная с выполнения второго запроса и до конца процедуры, нам необходимо обеспечить неизменность стои-



мости и остатков номенклатуры, с которой мы работаем, и запретить другим транзакциям даже читать эти данные. Сама система заблокирует изменение этих данных, но лишь начиная с момента записи движения.

Однако может возникнуть следующая ситуация. Выполняя второй запрос, мы прочитали, что есть 2 шт. некоторого материала. И другая транзакция (другой пользователь), которая собирается списывать материалы, тоже прочитала, что есть 2 шт. этого материала. После этого мы записали движения, система заблокировала эти данные. Другая транзакция ждет, когда мы освободим данные. Мы провели документ, списали 2 шт. материала и освободили данные. Другая транзакция пытается тоже списать 2 шт. материала, но его уже нет!

Аналогичная ситуация может возникнуть и между пунктами 3 и 4, в результате чего контроль остатков будет работать неверно.

Поэтому необходимо заблокировать остатки от чтения другими транзакциями еще до выполнения второго запроса. То есть прежде чем читать что-то, что мы собираемся изменять, нужно запретить чтение этих данных другими транзакциями до окончания введения изменений нами.

В режиме *Конфигуратор* посмотрим на свойство конфигурации **Режим управления блокировкой данных**. Он установлен в значение *Управляемый* (рис. 7.14).

Это значит, что нам нужно использовать управляемые блокировки, которые устанавливаются средствами встроенного языка.

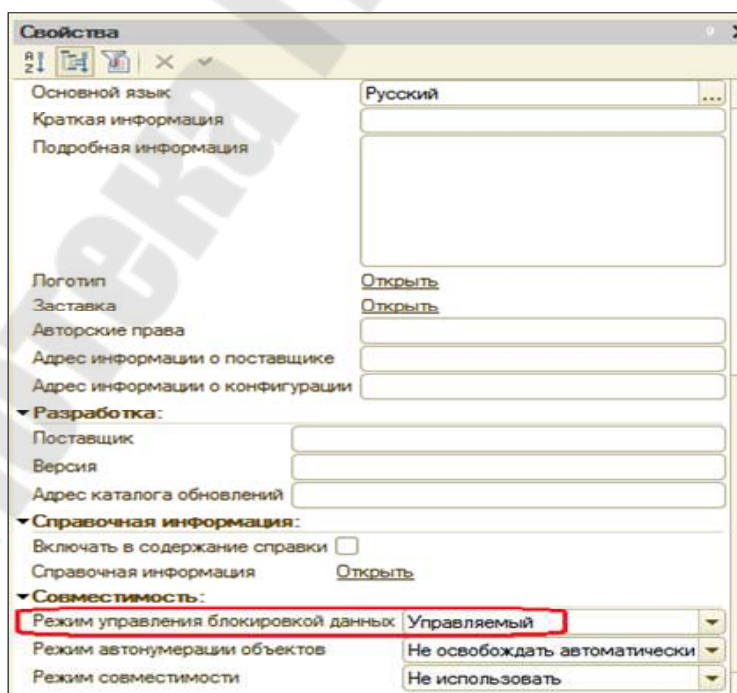


Рис. 7.14. Проверка режима управления блокировкой данных

Необходимо заблокировать те данные, которые мы собираемся читать и впоследствии изменять. Для этого у наборов записей регистров есть свойство **БлокироватьДляИзменения**. Вставим этот код перед записью пустых наборов записей.

```
ПО НоменклатураДокумента.Номенклатура = ОстаткиМатериаловОстатки.Материал";  
  
//Установим необходимость блокировки данных в регистрах СтоимостьМатериалов и ОстаткиМатериалов.  
Движения.СтоимостьМатериалов.БлокироватьДляИзменения = Истина;  
Движения.ОстаткиМатериалов.БлокироватьДляИзменения = Истина;  
  
//Запишем пустые наборы записей, чтобы читать остатки без учета данных в документе.  
Движения.СтоимостьМатериалов.Записать(); Движения.ОстаткиМатериалов.Записать();  
  
Результат = Запрос2.Выполнить();
```

Управляемая блокировка будет установлена в момент записи этих наборов записей, т. е. как раз перед выполнением второго запроса. Что и требовалось.

Запустите режим отладки и проверьте работу нового обработчика события *ОбработкаПроведения*, перепроведя все документы *Оказание услуги*.

### *Контрольные вопросы к заданиям 7.1–7.3*

1. Почему для доступа к массивам данных информационной базы предпочтительнее использовать запросы?
2. Почему при неоперативном проведении документов не нужно контролировать остатки?
3. Что такое временные таблицы и зачем их использовать?
4. Что такое менеджер запросов?
5. Как программно заблокировать данные?

## **ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ ДЛЯ РАЗРАБОТКИ ПРИКЛАДНОГО РЕШЕНИЯ**

1. Информационная система (далее – ИС) регистрации купли-продажи валют.
2. ИС учета текущей успеваемости студентов.
3. ИС учета проката автомобилей.
4. ИС обслуживания абонентов библиотеки.
5. ИС учета доходов от продаж товаров.
6. ИС расчета с абонентами за электроэнергию.
7. ИС расчета с абонентами за услуги ЖКХ.
8. ИС расчета стоимости коммунальных услуг.
9. ИС расчета с абонентами за услуги мобильной связи.
10. ИС страхования автотранспорта.
11. ИС страхования жилья.
12. ИС расчета за проживание в общежитии.
13. ИС учета кадров предприятия.
14. ИС учета движения товаров на складе.
15. ИС расчета заработной платы работников.
16. ИС учета выполнения производственных работ.
17. ИС записи на прием в медицинское учреждение.
18. ИС учета результатов сдачи сессии студентами.
19. ИС расчета стоимости услуг по ремонту автомобилей.
20. ИС управления складом.
21. ИС учета продаж в магазине.
22. ИС учета грузоперевозок компании.
23. ИС учета материально-технических ценностей на кафедре.
24. ИС учета студентов факультета.
25. ИС учета движения топлива на АЗС.
26. ИС кадастрового учета земельных участков.
27. ИС учета легковых автомобилей в ГАИ.
28. ИС учета заказов в интернет-магазине.
29. ИС учета заказов в химчистке одежды.
30. ИС учета книг в хранилище библиотеки.
31. ИС учета сделок по купле-продаже недвижимости.
32. ИС учета банковских переводов.
33. ИС учета основных средств предприятия.
34. ИС учета операций со счетами организации.
35. ИС учета прохождения техосмотра автотранспорта.

## ЛИТЕРАТУРА

1. Кашаев, С. М. Программирование в 1С: Предприятие 8.3 / С. М. Кашаев. – СПб. : Питер, 2014. – 304 с. – Режим доступа: [https://www.prorobot.ru/load/Kashaev\\_S\\_Programmirovanie\\_v\\_1S\\_Predpriatie\\_8\\_3.pdf](https://www.prorobot.ru/load/Kashaev_S_Programmirovanie_v_1S_Predpriatie_8_3.pdf).

2. Лыгин, А. Н. Практические работы по 1С «Предприятие 8.2» / А. Н. Лыгин. – М. : МИИГАиК, 2011. – 508 с. – Режим доступа: [https://www.studmed.ru/lygin-a-n-prakticheskie-raboty-po-1s-predpriatie-8-2\\_ef17d9904be.html](https://www.studmed.ru/lygin-a-n-prakticheskie-raboty-po-1s-predpriatie-8-2_ef17d9904be.html).

3. Радченко, М. Г. 1С: Предприятие 8.2. Практическое руководство разработчика. Примеры и типовые приемы / М. Г. Радченко, Е. Ю. Хрусталева. – М. : 1С-Паблишинг, 2009. – Режим доступа: <https://www.booktech.ru/books/sapr-i-drugie/11051-1spredpriatie-8-2-prakticheskoe-rukovodstvo-razrabotchika-2009-m-g-radchenko.html>.

4. Радченко, М. Г. 1С: Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы / М. Г. Радченко, Е. Ю. Хрусталева. – М. : 1С-Паблишинг, 2013. – Режим доступа: <https://www.booktech.ru/books/sapr-i-drugie/11052-1spredpriatie-8-3-prakticheskoe-posobie-razrabotchika-2013-m-g-radchenko.html>.

5. 1С: Предприятие 8.3. Руководство разработчика: издание 83.002.05. – М. : ООО «1С», 2013 г. – Режим доступа: <https://master1c8.ru/wp-content/uploads/2017/01/Руководство-разработчика-83.002.05.pdf>.

6. 1С: Предприятие 8.3. Руководство пользователя: издание 83.004.04. – М. : ООО «1С», 2013 г. – Режим доступа: <https://master1c8.ru/wp-content/uploads/2017/01/Руководство-пользователя-83.004.04.pdf>.

Учебное электронное издание комбинированного распространения

# **КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ**

**Практикум  
по выполнению лабораторных работ  
для студентов специальности 1-40 04 01 «Информатика  
и технологии программирования»  
дневной формы обучения**

Составитель **Ермалинская Наталья Васильевна**

Редактор  
Компьютерная верстка

*Н. Г. Мансурова  
Н. Б. Козловская*

Свидетельство о гос. регистрации в качестве издателя  
печатных изданий за № 1/273 от 04.04.2014 г.  
пр. Октября, 48, 246746, г. Гомель