

Министерство образования и науки Российской Федерации  
Курганский государственный университет  
Кафедра «Анализ, бухгалтерский учет и аудит»

**АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ  
БУХГАЛТЕРСКОГО И НАЛОГОВОГО  
УЧЕТА И АУДИТА**

Методические указания  
к выполнению лабораторных работ  
студентами очной формы обучения  
специальности 080109 «Бухгалтерский учет, анализ и аудит»

Курган 2012

Кафедра: «Анализ, бухгалтерский учет и аудит»

Дисциплина: «Автоматизированные системы бухгалтерского и налогового учета и аудита».

Составитель: канд. экон. наук, доцент О.А. Воробьева.

Утверждены на заседании кафедры «17» ноября 2011 г.

Рекомендованы методическим советом университета «24» ноября 2011 г.

## Содержание

### **Лабораторная работа 1.**

Создание объекта конфигурации Справочник ..... 4

### **Лабораторная работа 2.**

Создание объекта конфигурации Документ ..... 8

### **Лабораторная работа 3.**

Создание объекта конфигурации Регистр накопления (остатки)..... 13

### **Лабораторная работа 4.**

Создание объектов конфигурации Отчет и Макет. Редактирование макетов и форм ..... 15

Создание объекта конфигурации Макет ..... 17

Редактирование макетов и форм..... 18

### **Лабораторная работа 5.**

Создание объектов конфигурации Регистр сведений и Перечисление ..... 19

### **Лабораторная работа 6.**

Проведение документа по нескольким регистрам..... 24

### **Лабораторная работа 7.**

Создание объекта конфигурации Регистр накопления (обороты) ..... 28

### **Лабораторная работа 8.**

Создание отчетов ..... 30

## Лабораторная работа 1. Создание объекта конфигурации Справочник

### Объект конфигурации Справочник

Объект конфигурации *Справочник* является прикладным и предназначен для работы со списками данных. Объект конфигурации *Справочник* используется для того, чтобы на его основе платформа создала в базе данных информационную структуру, в которой будет храниться, например, список сотрудников, перечень товаров, список клиентов или поставщиков.

*Справочник* состоит из *элементов*. Характерной особенностью объекта конфигурации *Справочник* является то, что пользователь в процессе работы может самостоятельно добавлять новые элементы в *Справочник*: например, добавить новых сотрудников, создать новый товар или внести нового клиента.

### Создание справочника Клиенты

Задача – создать справочник, в котором будут храниться наименования клиентов организации. Для этого в открытом дереве конфигурации на объекте метаданных справочник нажмем *Добавить*. После этого система откроет *окно редактирования объекта конфигурации*.

Зададим *имя* справочника – *Клиенты*. Имя является основным свойством любого объекта конфигурации. При создании нового объекта система автоматически присваивает ему некоторое имя. Можно использовать имя, присвоенное системой, но лучше заменить его своим, понятным именем. Имя можно задавать любое, главное, чтобы оно начиналось с буквы и не содержало некоторых специальных символов (например, пробел). Для удобства чтения конфигурации принято составлять интуитивно понятные имена и, если они состоят из нескольких слов, удалять пробелы между словами и каждое слово начинать с большой буквы.

На основании имени платформа автоматически создаст *синоним* справочника Клиенты. Свойство *Синоним* также есть у любого объекта конфигурации. Оно предназначено для хранения «альтернативного» наименования объекта конфигурации, которое будет использовано в элементах интерфейса программы, то есть будет показано пользователю. Поэтому на синоним практически нет никаких ограничений, и его можно задавать в привычном для человека виде.

Все свойства объекта конфигурации Справочник пока встраивать не будем, поэтому три раза нажмем *Далее* и окажемся на закладке *Данные*.

Здесь представляют интерес длина кода и длина наименования. *Длина кода* – важное свойство справочника. Как правило, код справочника используется для идентификации элементов справочника и содержит уникальные для каждого элемента справочника значения. Платформа может сама контролировать уникальность кодов и поддерживать автоматическую нумерацию элементов справочника.

Уникальность кодов платформа может отслеживать сама, поэтому от длины кода будет зависеть количество элементов, содержащихся в справочнике. Длина кода – 9 символов. В результате можно использовать коды от 1 до

999 999 999.

Перейдем к *длине наименования*. 25 символов – мало, увеличим длину наименования до **50**.

Посмотрим, что создаст система на основе того Объекта конфигурации Справочник, который был только что добавлен. Для этого выполним пункт меню **Отладка ► Начать отладку**. На появившийся вопрос ответим **Да** и в следующем окне нажмем **Принять**. Все остальные свойства объекта конфигурации Справочник оставим такими, как их по умолчанию предлагает система, и нажмем **Заккрыть**.

После выполнения указанных команд откроется окно системы в *режиме 1С:Предприятие*. Поскольку мы не создавали никаких интерфейсов пользователей, для просмотра результатов нашей работы следует воспользоваться меню, которое создает система по умолчанию. Выполним пункт меню **Операции ► Справочник** и выберем справочник **Клиенты**.

Пока справочник Клиенты пуст, поэтому добавим в него несколько элементов (добавить новый элемент в справочник можно при помощи пункта меню **Действия ► Добавить**, соответствующей иконки или клавишей **Insert**):

- Гаврилов Роман Геннадьевич;
- Иванов Иван Иванович;
- Спиридонова Галина Петровна.

Теперь можно перейти к созданию второго справочника, который будет использоваться в данной конфигурации.

### Создание справочника **Сотрудники**

Справочник **Сотрудники** будет устроен несколько сложнее, чем Справочник Клиенты. В нем будет храниться не только фамилия, имя и отчество сотрудника, но и информация о прошлой трудовой деятельности. Эта информация однородна по своей структуре (организация, начало, окончание работы, занимаемая должность), но количество предыдущих мест работы у разных сотрудников может быть различным. Поэтому для хранения такой информации будем использовать табличную часть справочника.

Вернемся в *режим Конфигуратора* и создадим новый объект конфигурации **Справочник**. Назовем его **Сотрудники**. На закладке **Данные** зададим длину кода – **9**, тип кода – **Строка**, длину наименования справочника равной **50** символам и добавим в справочник новую **табличную часть** с именем **ТрудоваяДеятельность**.

Создадим **реквизиты** табличной части ТрудоваяДеятельность:

- **Организация** – тип **Строка**, длина **100**;
- **НачалоРаботы** – тип **Дата**, состав даты – **Дата**;
- **ОкончаниеРаботы** – тип **Дата**, состав даты – **Дата**;
- **Должность** – тип **Строка**, длина **100**.

Теперь следует выбрать вариант редактирования справочника. Очевидно, что редактирование в списке уже не подойдет, поскольку в списке нельзя редактировать табличную часть справочника и вводить информацию о трудовой деятельности. Поэтому в справочнике **Сотрудники** мы выберем вариант редак-

тирования справочника обоими способами – *как в списке, так и в диалоге*. Для этого перейдем на закладку **Формы** и установим соответствующий переключатель.

Создание справочника Сотрудники завершено – теперь можно запустить 1С:Предприятие в режиме отладки и заполнить справочник.

Выполнить пункт меню **Отладка** ► **Продолжить отладку**, на вопрос системы ответить *Да*, на последующие вопросы ответить так же, как и ранее. В появившемся окне программы открыть справочник **Сотрудники** и убедиться, что установлен режим редактирования справочника в диалоге (меню **Действия** ► **Редактировать в диалоге** или аналогичная иконка в командной панели).

Создать сотрудников:

- Гусаков Николай Дмитриевич (ЗАО «НТЦ», начало работы – 01.02.2000, окончание работы – 16.04.2003, должность – ведущий специалист);
- Деловой Иван Сергеевич (ООО «Автоматизация», начало работы – 22.01.1996, окончание работы – 31.12.2002, должность – инженер; НПО «СпецСвязь», начало работы – 20.06.1986, окончание работы – 21.01.1995, должность – начальник производства);
- Симонов Валерий Михайлович (ООО «СтройМастер», начало работы – 06.02.2001, окончание работы – 03.04.2004, должность – ведущий про-раб).

Обратить внимание, что строки табличной части справочника можно сортировать по содержимому любой из колонок или располагать в произвольном порядке, используя иконки командной панели.

### Создание справочника Номенклатура

Справочник **Номенклатура** будет содержать информацию об услугах, которые оказывает организация, и о тех материалах, которые при этом могут быть использованы. Этот справочник не будет сложным. Единственная особенность, которой он будет обладать, – наличие иерархической структуры. Для того чтобы справочником было удобно пользоваться, сгруппируем услуги в одну группу, а материалы – в другую. Кроме этого, поскольку организация оказывает самые разные услуги, они также будут логически собраны в несколько групп. То же самое можно сказать и про материалы. Задача будет состоять в создании иерархического справочника.

Создадим новый объект конфигурации **Справочник** и назовем его **Номенклатура**. Перейдем на закладку **Иерархия** и установим флаг **Иерархический справочник**. На закладке **Данные** зададим длину кода – **9**, тип кода – **Строка**, длину наименования справочника равной **100** символам.

Запустим 1С:Предприятие в режиме отладки и заполним справочник Номенклатура. В процессе заполнения освоим, как создавать группы и переносить элементы из одной группы в другую.

Создадим *две группы* в корне справочника: **Материалы** и **Услуги** (меню **Действия** ► **Новая группа**).

В группе **Материалы** создадим пять *элементов*:

- Строчный трансформатор Samsung;

- Строчный трансформатор LG;
- Транзистор Philips;
- Шланг резиновый;
- Кабель электрический.

В группе *Услуги* тоже создадим несколько *элементов* – услуги по ремонту телевизоров:

- Диагностика;
- Ремонт отечественного телевизора;
- Ремонт импортного телевизора.

Кроме того, в качестве услуг по установке стиральных машин определим:

- Подключение воды;
- Подключение электричества.

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

Для того чтобы переместить услуги в соответствующие группы, в окне списка установим курсор на ту услугу, которую хотим переместить, и выполним команду *Действия ► Переместить в группу*. В открывшемся окне выберем новую группу.

Аналогичным образом переместите в группу Телевизоры услуги Ремонт отечественного телевизора и Ремонт импортного телевизора. Услуги Подключение воды и Подключение электричества переместить в группу Стиральные машины.

Затем в *группе Материалы* создать *две группы: Радиодетали и Прочее*. В группу Прочее поместить Кабель электрический и Шланг резиновый. Остальные материалы переместить в группу Радиодетали.

### Создание справочника Склады

Создадим справочник *Склады*, который будет содержать информацию о складах, используемых в организации. Этот справочник будет включать в себя один *предопределенный элемент* – склад Основной, на который будут поступать все материалы. Таким образом, задача – создать справочник, содержащий предопределенные элементы.

Откроем конфигуратор и создадим новый объект конфигурации *Справочник* с именем *Склады*. На закладке *Данные* установим длину кода – *9* и тип кода – *Строка*. Перейдем на закладку *Прочее* и нажмем кнопку *Предопределенные*. Система откроет список предопределенных элементов справочника. Сейчас он пуст, поэтому выполним команду *Действия ► Добавить* и создадим предопределенный элемент с именем *Основной*.

Обратите внимание на то, что помимо наименования мы задали еще и *имя* предопределенного элемента справочника. В дальнейшем, когда возникнет потребность в использовании средств встроенного языка, будет возможно обратиться к этому элементу справочника, используя имя, присвоенное ему в конфигураторе.

Запустим 1С:Предприятие в режиме отладки и откроем справочник Склады. Добавим в справочник еще один *склад*, который назовем *Розничный*.

Обратите внимание, что система отмечает различными пиктограммами простой и предопределенный элементы справочника. Несмотря на то что можно изменить код или наименование у обоих элементов, пометка на удаление (или удаление) возможна только для *простых элементов справочника*. При попытке пометить на удаление предопределенный элемент система выдаст предупреждение: *Пометка на удаление предопределенного элемента справочника запрещена*.

Таким образом, теперь можно обозначить две характерные особенности предопределенных элементов:

- на предопределенные элементы могут опираться алгоритмы работы конфигурации (т.к. возможно обращение к ним из встроенного языка по имени);
- предопределенные элементы являются объектами базы данных, которые нельзя удалить в режиме 1С:Предприятия.

## Лабораторная работа 2. Создание объекта конфигурации Документ

### Объект конфигурации Документ

Объект конфигурации *Документ* является прикладным и предназначен для описания информации о совершенных хозяйственных операциях или о событиях, произошедших в жизни организации вообще. На основе объекта конфигурации *Документ* платформа создает в базе данных информационную структуру, в которой будут храниться, например, такие документы, как приходные накладные, или приказы о приеме на работу, или платежные поручения, или счета и т.д.

Характерной особенностью объекта конфигурации *Документ* является то, что в процессе работы пользователь может самостоятельно создавать новые объекты этой структуры – новые документы.

### Создание документа Приходная накладная

Создадим несколько объектов конфигурации *Документ*, чтобы иметь возможность фиксировать события, происходящие в организации.

Одними из самых популярных услуг данного предприятия является ремонт телевизоров и установка стиральных машин. И в том, и в другом случае требуются некоторые материалы, которые расходуются в процессе оказания этих услуг. Поэтому двумя важнейшими событиями в хозяйственной жизни организации будут поступление материалов и оказание услуг.

Для отражения этих событий в базе данных создадим *два документа: Приходная накладная* и *Оказание услуги*. Документ *Приходная накладная* будет фиксировать факт поступления в организацию необходимых материалов, а документ *Оказание услуги* – фиксировать оказание услуг и расход материа-



лов, которые используются при оказании этих услуг.

Создадим новый объект конфигурации *Документ*. Зададим имя документа – *Приходная Накладная*. Нажмем *Далее*. Создадим *реквизит* документа с именем *Склад* и типом *СправочникСсылка.Склады*.

После этого добавим *табличную* часть с именем *Материалы* и создадим у нее четыре *реквизита*:

- *Материал*, 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*.

Перейдем на закладку *Нумерация* и зададим длину номера – *9* и тип номера – *Строка*.

Запустим 1С:Предприятие в режиме отладки и протестируем получившийся результат. В открывшемся окне программы выполним команду *Операции ► Документы*, выберем документ *Приходная накладная*.

Пока в базе данных нет ни одного документа Приходная накладная, поэтому создадим *новый документ* (создать новый документ можно при помощи пункта меню *Действия ► Добавить*, соответствующей иконки или клавишей *Insert*).

Система автоматически присвоит номер новому документу, и нам останется только заполнить табличную часть. Заполним ее материалами для ремонта телевизоров и нажмем *ОК*:

- Строчный трансформатор Samsung – количество 10, цена – 270,00, сумма – 2700,00;
- Строчный трансформатор LG – количество 10, цена – 600,00, сумма – 6000,00;
- Транзистор Philips – количество 10, цена – 3,00, сумма – 30,00.

Аналогичным образом создадим *второй документ*, который будет приходить следующие материалы для установки стиральных машин:

- Шланг резиновый – количество 5, цена – 100,00, сумма – 500,00;
- Кабель электрический – количество 5, цена – 20,00, сумма – 100,00.

При заполнении документа приходится вводить сумму в каждой строке. Это неудобно, и возникает естественное желание автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества материалов в строке.

Для этого потребуется сначала создать собственную форму документа, а затем воспользоваться возможностями встроенного языка.

### Создание формы документа

Прежде всего, следует заметить, что до сих пор мы использовали predefined формы объектов, которые система 1С:Предприятие по умолчанию создавала сама. Теперь же возникла необходимость слегка изменить логику работы формы документа, а значит, нам придется создать свою собственную *форму документа Приходная Накладная* для того, чтобы в ней можно было

описать тот алгоритм, который нам нужен.

Вернемся в конфигуратор и откроем окно редактирования объекта конфигурации *Документ ПриходнаяНакладная*. В этом окне нас интересует закладка **Формы**. Как видим, ни одна из основных форм документа пока не задана. Для того чтобы создать форму документа, нажмем на *символ лупы* в поле ввода.

Выберем тип формы **Форма документа** и нажмем кнопку *Готово*, согласившись тем самым со всем, что нам предложила система.

Обратите внимание, что в дереве объектов конфигурации у объекта конфигурации *Документ ПриходнаяНакладная* появилась **форма ФормаДокумента**, а на экране открылось окно редактора форм, содержащее эту форму.

Форма документа *ПриходнаяНакладная* содержит большое количество всевозможных полей. Эти поля называются *элементами управления*. Они имеют разное назначение и разное поведение, которое соответствует их назначению. Однако все они служат для того, чтобы отображать информацию, хранящуюся в базе данных, и организовывать интерактивную работу с этой информацией.

Сейчас мы обратим свое внимание только на те элементы управления, которые нас интересуют. Это три поля ввода, расположенные в колонках *Количество*, *Цена* и *Сумма*.

Мы хотим, чтобы каждый раз, когда меняется значение в поле *Количество* или в поле *Цена*, в поле *Сумма* автоматически устанавливалось значение, равное ***Количество\*Цена***. Очевидно, что для этого нужно написать на встроенном языке команду, похожую на ***Сумма = Количество\*Цена***, которая будет выполняться при изменении значения поля *Количество* или *Цена*.

### **Создание процедуры обработки события в модуле формы**

Щелкните правой кнопкой мыши на *поле ввода* в колонке *Количество* и откройте для него палитру свойств (пункт контекстного меню *Свойства*). Прокрутите список до конца, и вы обнаружите ***перечень событий***, которые могут быть связаны с этим полем ввода.

Среди событий, связанных с полем ввода, найдите событие ***ПриИзменении***. Оно возникает после изменения значения поля ввода. Щелкните по ***кнопке с лупой*** в конце поля ввода, и система создаст заготовку процедуры ***обработчика*** этого события в ***модуле*** формы.

***Модуль*** – это «хранилище» для текста программы на встроенном языке. В конфигурации существует большое количество модулей, которые расположены в различных ее точках. Они могут принадлежать некоторым объектам конфигурации (например, формам), а могут существовать сами по себе (принадлежать всей конфигурации в целом). Текст программы, содержащийся в модулях, будет использоваться платформой в заранее известные моменты работы системы 1С:Предприятие.

В модуль формы, в процедуру ***МатериалыКоличествоПриИзменении()*** добавим следующий текст (листинг 1).

### **Листинг 1. Процедура МатериалыКоличествоПриИзменении()**

```
СтрокаТабличнойЧасти=ЭлементыФормы.Материалы.ТекущиеДанные;  
СтрокаТабличнойЧасти.Сумма=СтрокаТабличнойЧасти.Количество*  
СтрокаТабличнойЧасти.Цена;
```

Теперь посмотрим, как это работает. В окне программы откроем список документов ПриходнаяНакладная и откроем любой из двух созданных нами документов. Если теперь поменять количество в любой строке документа, то сумма в строке будет пересчитана автоматически.

Но теперь хотелось бы и для поля Цена сделать то же самое. А если заглянуть вперед, то мы увидим, что подобное автоматическое заполнение поля Сумма может нам понадобиться и в других документах. Поэтому лучше будет поместить расчет суммы в некоторое «общедоступное» место, чтобы разные документы, имеющие аналогичные реквизиты табличной части, могли использовать этот алгоритм.

Для описания таких «общедоступных» мест служат объекты конфигурации **ОбщийМодуль**, расположенные в ветке **Общие ► Общие модули**. Процедуры и функции, содержащиеся в этих модулях, могут быть доступны для любых объектов конфигурации.

### **Создание процедуры обработки события в общем модуле**

Для того чтобы алгоритм, выполняемый при обработке события, был доступен для разных документов, создадим общий модуль и перенесем в него процедуру расчета суммы. А в документе просто оставим вызовы этой процедуры из общего модуля. Создадим объект конфигурации **ОбщийМодуль** в ветке **Общие ► Общие модули** и назовем его **РаботаСДокументами**. Он будет содержать следующий текст (листинг 2).

### **Листинг 2. Процедура РассчитатьСумму()**

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

Ключевое слово Экспорт в конце оператора Процедура как раз указывает на то, что эта процедура может быть доступна из других программных модулей.

Затем в модуле нашей формы изменим текст нашего обработчика (листинг 3).

### **Листинг 3. Процедура МатериалыКоличествоПриИзменении()**

```
Процедура МатериалыКоличествоПриИзменении(Элемент)  
СтрокаТабличнойЧасти=ЭлементыФормы.Материалы.ТекущиеДанные;  
РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);  
КонецПроцедуры
```

Таким образом, теперь вместо непосредственного расчета суммы мы будем вызывать процедуру из общего модуля и передавать ей в качестве параметра нужную нам строку табличной части.

Проверим, как это работает, и убедимся, что ничего не изменилось. Теперь осталось и для поля *Цена* установить такой же обработчик. Поэтому мы создадим *обработчик события ПриИзменении* для *поля ввода*, которое расположено в колонке *Цена*, и повторим в нем вызов процедуры *РассчитатьСумму* из общего модуля (листинг 4).

#### **Листинг 4. Процедура МатериалыЦенаПриИзменении()**

Процедура МатериалыЦенаПриИзменении(Элемент)

СтрокаТабличнойЧасти=ЭлементыФормы.Материалы.ТекущиеДанные;  
РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);

КонецПроцедуры

Запустим 1С:Предприятие в режиме отладки и убедимся, что теперь сумма в строках табличной части документов ПриходнаяНакладная пересчитывается как при изменении количества, так и при Изменении цены.

#### **Создание документа Оказание услуги**

Аналогичным образом создадим второй документ *Оказание услуги*. Для этого потребуется выполнить следующие действия. Создать новый объект конфигурации *Документ Оказание услуги с реквизитами*:

- *Склад*, тип *СправочникСсылка.Склады*;
- *Клиент*, тип *СправочникСсылка.Клиенты*;
- *Мастер*, тип *СправочникСсылка.Сотрудники*.

Создать *табличную часть* этого документа *Перечень Номенклатуры 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*.

Создать основную *форму документа*: для *полей ввода* колонок *Количество* и *Цена* создать *обработчик события ПриИзменении*, в котором вызывать процедуру *РассчитатьСумму* из общего модуля *РаботаСДокументами*.

После того, как эти действия будут выполнены, запустить 1С:Предприятие в режиме отладки и убедиться, что при вводе цены и количества в табличную часть документа *Оказание услуги* сумма пересчитывается по требуемому алгоритму.

## Лабораторная работа 3. Создание объекта конфигурации Регистр накопления (остатки)

### Объект конфигурации Регистр накопления

Объект конфигурации *Регистр накопления* является прикладным и предназначен для описания структуры накопления данных. На основе объекта конфигурации Регистр накопления платформа создает в базе данных информационную структуру, в которой будут накапливаться данные, «поставляемые» различными объектами базы данных. Эти данные будут храниться в регистре в виде отдельных записей, каждая из которых имеет одинаковую, заданную в конфигураторе структуру.

Отличительной особенностью регистра накопления является то, что он не предназначен для интерактивного редактирования пользователем. Разработчик может при необходимости предоставить пользователю возможность редактировать регистр накопления. Но предназначение регистра накопления заключается в том, чтобы его модификация производилась на основе алгоритмов работы других объектов базы данных, а не в результате непосредственных действий пользователя.

### Создание регистра накопления Остатки материалов

Для накопления информации о том, сколько и каких материалов есть в наличии на складах, создадим регистр *ОстаткиМатериалов*.

Откроем в конфигураторе учебную конфигурацию и создадим новый объект конфигурации *Регистр накопления*. Зададим имя регистра – *Остатки-Материалов*. Нажмем *Далее* и перейдем к созданию структуры регистра.

Создадим *измерения* регистра:

- *Материал*, тип *СправочникСсылка.Номенклатура*;
- *Склад*, тип *СправочникСсылка.Склады*.

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

Если сейчас попытаться запустить 1С:Предприятие в режиме отладки, то система выдаст сообщение об ошибке: *«РегистрНакопления.ОстаткиМатериалов: Ни один из документов не является регистратором для регистра»*. Это сообщение еще раз подтверждает тот факт, что назначение регистра накопления в том, чтобы аккумулировать данные, поставляемые различными документами.

### Создание движений документа Приходная накладная

*Движения документа* – это записи в регистрах, которые создаются в процессе проведения документа и отражают изменения, производимые документом. Откроем окно редактирования объекта конфигурации *Документ ПриходнаяНакладная*. Перейдем на закладку *Движения* и в списке регистров конфигурации отметим регистр накопления *ОстаткиМатериалов*.

Обратите внимание, что сразу после отметки выбранного регистра становится доступной кнопка *Конструктор движений*. Этим конструктором мы и воспользуемся.

Конструктор устроен просто. В списке *Регистры* перечислены регистры, в которых документ может создавать движения. В данном случае там пока один регистр *ОстаткиМатериалов*.

В списке *Реквизиты* документа должны находиться исходные данные для создания движений. А в таблице *Поле – Выражение* должны быть заданы формулы, по которым будут вычисляться значения измерений и ресурсов регистра при записи движений.

Обратите внимание, что по умолчанию конструктор предлагает создавать движения *прихода* (Тип движения регистра – Приход, символ + рядом с названием регистра) по регистру *ОстаткиМатериалов*. Это нас вполне устраивает, ведь документ *ПриходнаяНакладная* и должен приходовать материалы.

В поле выбора *Табличная часть* выберем *табличную часть* данного документа – *Материалы*. Список реквизитов документа автоматически заполнится реквизитами этой табличной части. Нажмем кнопку *Заполнить выражения*. В нижнем окне сформируется соответствие полей и выражений.

Как видите, конструктор движений установил соответствия подходящим образом: в качестве материала в регистр будет записан материал из табличной части документа, в качестве склада – склад, указанный в шапке документа, а в качестве количества – количество из табличной части документа.

Нажмем кнопку *ОК* и посмотрим, какой текст сформировал конструктор в модуле объекта (листинг 5).

#### *Листинг 5. Процедура ОбработкаПроведения()*

Процедура ОбработкаПроведения(Отказ, Режим)

```
//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

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

```
// При повторном использовании конструктора
```

```
// внесенные вручную изменения будут утеряны!!!
```

```
Для каждого ТекСтрокаМатериалы Из Материалы Цикл
```

```
    // Регистр ОстаткиМатериалов Приход
```

```
    Движение = Движения.ОстаткиМатериалов.Добавить();
```

```
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
```

```
    Движение.Период = Дата;
```

```
    Движение.Материал = ТекСтрокаМатериалы.Материал;
```

```
    Движение.Склад = Склад;
```

```
    Движение.Количество = ТекСтрокаМатериалы.Количество;
```

```
КонецЦикла;
```

```
//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

```
КонецПроцедуры
```

Посмотрим, как это работает. Запустим 1С:Предприятие в режиме отладки и откроем одновременно два окна: *список документов ПриходнаяНакладная* и *список регистра накопления ОстаткиМатериалов*.

Откроем *Приходную накладную № 1* и нажмем *ОК*. Обратите внимание, что при проведении приходной накладной появляются соответствующие записи

в регистрах накопления. Аналогичные действия проделаем и с документом **Приходная накладная № 2**.

### **Создание движений документа Оказание услуги**

Аналогичным образом создадим *движения* документа **ОказаниеУслуги**.

При использовании конструктора будем внимательны и обратим внимание на то, что документ ОказаниеУслуги должен *расходовать* материалы. Поэтому перед тем, как нажать кнопку **ОК** убедимся, что выбран правильный *тип движения регистров* (нужен *Расход*). Кроме того, при автоматическом заполнении *поле Материал* не заполнится. Если оставить это так, как есть, то в регистре накопления в строках с типом *Движение регистра – расход* имя Номенклатуры фиксироваться не будет. Чтобы избежать этого, надо выбрать *поле Материал* и в *поле Реквизиты* документа дважды щелкнуть по строке **ТекСтрокаПереченьНоменклатуры.Номенклатура**. Таким образом, имя номенклатуры для строк регистра накопления будет выбираться из табличной части документа.

Запустим отладку и создадим *документ Оказание услуги*, который будет расходовать один транзистор Philips за 3 рубля. Проведем этот документ и убедимся, что в регистре накопления он создал верные движения.

Сформированные таким образом движения этого документа будут не совсем правильны. Дело в том, что в документе Оказание услуги, в отличие от документа Приходная накладная могут содержаться не только расходующие материалы, но и услуги. Поэтому в регистр Остатки материалов будут попадать записи и о расходующих услугах, что неправильно. Данный недостаток будет исправлен на одной из следующих лабораторных работ после того, как познакомимся с перечислениями.

## **Лабораторная работа 4. Создание объектов конфигурации Отчет и Макет. Редактирование макетов и форм**

### **Объект конфигурации Отчет**

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

### **Создание отчета Материалы**

Приступим к созданию отчета, который будет показывать приход, расход и остатки материалов.

Создадим новый объект конфигурации **Отчет** и назовем его **Материалы**. Откроем окно редактирования объекта конфигурации **Отчет Материалы** и создадим схему компоновки данных. Для этого нажмем кнопку **Открыть схе-**

*му компоновки данных.*

Так как у отчета, который мы создаем, еще не существует схемы компоновки данных, платформа предложит создать новую схему. Схема компоновки данных с точки зрения конфигурации является макетом, поэтому будет открыт конструктор макета, предлагающий выбрать единственный тип макета – **Схема компоновки данных**. Нажмем кнопку **Готово**.

Платформа создаст новый макет, содержащий схему компоновки данных, и сразу же откроет конструктор схемы компоновки данных. Конструктор обладает большим количеством возможностей для визуального проектирования отчетов, но сейчас воспользуемся только самыми простыми его возможностями и определим те данные, которые хотим видеть в результате работы нашего отчета. Добавим новый **набор данных – запрос**.

Для того чтобы создать текст запроса, запустим конструктор запроса – кнопка **Конструктор запроса**. В списке **База данных** представлены таблицы запросов. На основе их данных имеем возможность построить отчет. Если раскрыть ветку **РегистрыНакопления**, то увидим, что кроме таблицы регистра **ОстаткиМатериалов** в этой ветке присутствуют еще несколько виртуальных таблиц, которые формирует система.

Поскольку мы хотим видеть как остатки материалов, так и информацию об их поступлении и расходовании, нас будет интересовать виртуальная таблица **ОстаткиМатериалов.ОстаткиИОбороты**. Раскроем ее. Эта таблица содержит материал, склад и кроме этого начальные и конечные остатки, а также значения прихода, расхода и оборотов для всех ресурсов регистра **ОстаткиМатериалов**.

Начнем выбирать поля таблицы в нужном нам порядке двойным щелчком мыши:

- **Склад**;
- **Материал**;
- **КоличествоНачальныйОстаток**;
- **КоличествоПриход**;
- **КоличествоРасход**;
- **КоличествоКонечныйОстаток**.

После этого на закладке **Группировка** укажем групповое поле **Склад**. Нажмем **ОК** и вернемся в конструктор схемы компоновки данных.

Текст запроса, который был создан с помощью конструктора, платформа поместит в поле **Запрос**. Это поле представляет собой текстовый редактор, в котором можно вручную отредактировать существующий запрос. Кроме того, можно снова вызвать конструктор запроса и отредактировать запрос при помощи него.

Обратим внимание на список полей системы компоновки данных, который платформа заполнила в верхней части конструктора. В нем отображаются поля, которые доступны у текущего набора данных. В данном случае система 1С:Предприятие заполнила данный список автоматически, из текста запроса, и нет необходимости в его ручной настройке.



Перейдем на закладку **Настройки** и добавим группировку (контекстное меню – **Новая группировка**). При этом не станем указывать поле группировки, а просто нажмем **ОК**. В отчет будут выводиться детальные записи из информационной базы – записи, получаемые в результате выполнения запроса без итогов.

Теперь настроим поля, которые будут выводиться в результате отчета. Для этого перейдем на закладку **Выбранные поля** и перенесем мышью из списка доступных полей:

- **Склад**;
- **Материал**;
- **КоличествоНачальныйОстаток**;
- **КоличествоПриход**;
- **КоличествоРасход**;
- **КоличествоКонечныйОстаток**.

Запустим 1С:Предприятие в режиме отладки и посмотрим, как работает отчет. Выполним команду **Операции ► Отчет... ► Материалы**. Система автоматически сформирует форму отчета и форму настроек для него. Нажмем **Сформировать**.

## Создание объекта конфигурации **Макет**

### Объект конфигурации **Макет**

Объект конфигурации **Макет** предназначен для хранения различных форм представления данных, которые могут потребоваться каким-либо объектам конфигурации или всему прикладному решению в целом. Макет может содержать табличный или текстовый документ, двоичные данные, HTML-документ или Active Document, графическую или географическую схему, схему компоновки данных или макет оформления схемы компоновки данных. Макеты могут существовать как сами по себе (общие макеты), так и быть подчинены какому-либо объекту конфигурации.

Одно из предназначений макета, подчиненного объекту конфигурации и содержащего табличный документ, – создание печатной формы этого объекта. Создание печатной формы заключается в конструировании ее составных частей – именованных областей, из которых затем «собирается» готовая печатная форма. Порядок заполнения областей данными и вывода их в итоговую форму описывается при помощи встроенного языка. Печатная форма может включать в себя различные графические объекты: картинки, диаграммы и т.д.

Помимо создания макета «вручную» конфигуратор предоставляет разработчику возможность воспользоваться специальным инструментом – конструктором печати, который берет на себя большинство рутинной работы по созданию макета.

### Создание макета документа **Оказание услуги**

Откроем в конфигураторе окно редактирования объекта конфигурации **Документ ОказаниеУслуги**. Перейдем на закладку **Макеты** и запустим **конст-**

*руктор печати.*

*На первом шаге* укажем, что новая процедура, которая будет создана конструктором для формирования печатной формы документа, будет располагаться в модуле формы.

*На втором шаге* нажатием двойной стрелки определим, что все реквизиты нашего документа будут отображены в шапке печатной формы.

*На третьем шаге* точно так же определим, что все реквизиты табличной части документа будут отображены в печатной форме.

*На четвертом шаге* конструктор предложит сформировать нам «подвал» (нижнюю часть) печатной формы. Не станем ничего указывать («подвал» в данном случае использовать не будем).

*На пятом шаге* укажем, что конструктор должен вставить новую кнопку в форму документа для вызова процедуры формирования печатной формы, и нажмем **ОК**. В конфигураторе откроется форма документа и его макет.

Проверим макет в работе. Запустим 1С:Предприятие в режиме отладки и откроем документ *Оказание услуги № 1*. Обратите внимание, что в правом нижнем углу документа появилась новая кнопка **Печать**. Эту кнопку добавил конструктор. Нажмем на нее и увидим и печатную форму нашего документа.

Как видите, конструктор сформировал вполне подходящую печатную форму для нашего документа. Единственное, чего не хватает в данной форме, – это итоговой суммы документа. Далее познакомимся с тем, как можно редактировать макеты и формы объектов конфигурации, и добавим итоговую сумму к форме и макету документа *ОказаниеУслуги*.

## Редактирование макетов и форм

### Редактирование макета документа *Оказание услуги*

Откроем конфигуратор, раскроем дерево документа *Оказание Услуги* и дважды щелкнем на макете **Печать**. Как видите, макет документа состоит из именованных областей, которые в определенном порядке выводятся на печать.

Добавим новую область для вывода итоговой суммы документа. Выделим мышью две пустые строки и выполним команду **Таблица ► Имена ► Назначить имя...**

Назовем область **Всего**. В созданной области, в колонке **Цена** напишем **Всего**, а в колонке **Сумма** напишем **ВсегоПоДокументу**.

Вызвав палитру свойств для последней заполненной ячейки, укажем, что в этой ячейке будет находиться не текст, а **параметр**. Здесь следует сказать о том, что каждая ячейка редактируемого табличного документа может содержать либо текст, либо некоторый параметр, либо шаблон.

**Текст**, содержащийся в ячейке, будет показан на экране.

**Параметр** будет заменен некоторым значением, которое может быть присвоено ему средствами встроенного языка. Текст, содержащийся в ячейке, является именем этого параметра.

**Шаблон** представляет собой текстовую строку, в определенные места которой будут вставлены значения параметров.

Поэтому, указав для ячейки в качестве заполнения *Параметр*, мы определили параметр области с именем *ВсегоПоДокументу*, которому присвоим нужное значение при формировании печатной формы.

Теперь откроем модуль формы документа *ОказаниеУслуги – ФормаДокумента*. Найдем в нем процедуру *Печать* и после цикла *Для Каждого Текст-рокатчНоменклатура Из тчНоменклатура Цикл* добавим в нее следующие строки (листинг 6).

#### **Листинг 6. Печать формы документа**

```
Область = Макет.ПолучитьОбласть("Всего");  
Область.Параметры.ВсегоПоДокументу=ПереченьНоменклатуры.Итог("Сумма");  
ТабДок.Вывести(Область);
```

Для того чтобы наш документ *ОказаниеУслуги*, выглядел вполне законченным, добавим итоговую сумму по документу и на экранную форму, чтобы пользователь мог видеть ее в процессе заполнения табличной части документа.

#### **Редактирование формы документа Оказание услуги**

Откроем в конфигураторе форму документа *ОказаниеУслуги – ФормаДокумента*. Откроем палитру свойств для табличного поля, расположенного в форме, и установим свойство *Подвал*, которое определяет наличие подвала у элемента управления *Табличное поле*.

Затем откроем свойства колонки *Цена* и установим текст подвала – *Всего:, горизонтальное положение в подвале – Прижать вправо* и в шрифте подвала изменим начертание на *Жирный*.

После этого откроем свойства колонки *Сумма*, установим *горизонтальное положение в подвале – Прижать вправо*, установим флаг *Показывать итог* в подвале и в шрифте подвала тоже изменим начертание на *Жирный*.

Запустим 1С:Предприятие в режиме отладки и посмотрим, как теперь выглядит форма документа *Оказание услуги № 1*.

## **Лабораторная работа 5. Создание объектов конфигурации Регистр сведений и Перечисление**

### **Объект конфигурации Регистр сведений**

Объект конфигурации *Регистр сведений* является прикладным и предназначен для описания структуры хранения данных в разрезе нескольких измерений. На основе объекта конфигурации Регистр сведений платформа создает в базе данных информационную структуру, в которой может храниться произвольная информация, «привязанная» к набору измерений.

Принципиальное отличие регистра сведений от регистра накопления заключается 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 хра-

нить любые данные (а не только числовые, как регистр накопления).

Следующей важной особенностью регистра сведений является его способность (при необходимости) хранить данные с привязкой ко времени. Благодаря этому регистр сведений может хранить не только актуальные значения данных, но и историю их изменения во времени. Регистр сведений, использующий привязку ко времени, называют обычно *периодическим регистром сведений*.

### Создание периодического регистра сведений Цены

Приступим к созданию периодического регистра сведений, который будет хранить развернутые во времени розничные цены материалов и стоимости услуг. Откроем конфигуратор и создадим новый объект конфигурации *Регистр сведений*. Назовем его *Цены*. Установим периодичность этого регистра *в пределах секунды*.

Перейдем на закладку *Данные* и создадим *измерение регистра Номенклатура* с типом *СправочникСсылка.Номенклатура*. Укажем, что это измерение будет *ведущим*.

Свойство *Ведущее* имеет смысл использовать лишь тогда, когда измерение имеет тип ссылки на объект базы данных. Установка свойства *Ведущее* будет говорить о том, что запись регистра сведений представляет интерес, пока существует этот объект. При удалении объекта, все записи регистра сведений по этому объекту тоже будут автоматически удалены. Кроме того, в форме списка справочника появляется кнопка командной панели *Перейти*. По ней возможен переход к записям регистра, отобранным по значению выбранного элемента справочника.

После этого создадим новый *ресурс Цена*, тип *Число*, длина *15*, точность *2*, *неотрицательное*. Теперь запустим 1С:Предприятие в режиме отладки и посмотрим, как работает наш периодический регистр сведений Цены.

Зададим стоимость услуг организации следующим образом (таблица 1).

**Таблица 1 – Стоимость услуг, оказываемых предприятием**

31.03.2010 0:00:00	Диагностика	200,00
31.03.2010 0:00:00	Ремонт отечественного телевизора	600,00
31.03.2010 0:00:00	Ремонт импортного телевизора	800,00
31.03.2010 0:00:00	Подключение воды	800,00
31.03.2010 0:00:00	Подключение электричества	800,00

После этого зададим розничные цены на материалы (таблица 2).

**Таблица 2 – Розничные цены на материальные ценности предприятия**

31.03.2010 0:00:00	Строчный трансформатор Samsung	900,00
31.03.2010 0:00:00	Строчный трансформатор LG	400,00
31.03.2010 0:00:00	Транзистор Philips	5,00
31.03.2010 0:00:00	Шланг резиновый	150,00
31.03.2010 0:00:00	Кабель электрический	30,00

Поскольку цены хранятся с привязкой к дате, можно заранее установить новые цены и быть уверенными в том, что новые цены вступят в действие не раньше указанной для них даты. Теперь посмотрим, как можно использовать заданные нами цены в документе *ОказаниеУслуги*.

### **Создание функции Розничная цена**

Сначала создадим функцию, которая будет возвращать актуальную розничную цену номенклатуры.

Откроем конфигуратор, в ветке *Общие* ► *Общие модули* создадим новый объект конфигурации *Модуль* и назовем его *РаботаСоСправочниками*. Поместим в нем следующий текст (листинг 7).

#### ***Листинг 7. Функция РозничнаяЦена()***

```
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
```

### **Автоматическое заполнение цены в документе Оказание услуги**

Задача, которая перед нами стоит, заключается в следующем. При создании *документа ОказаниеУслуги* 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 *форму ФормаДокумента*. Откроем *свойства поля ввода*, расположенного в *колонке Номенклатура*, и внизу списка найдем *событие ПриИзменении*. Нажмем на кнопку с лупой и в открывшейся заготовке *Обработчика события* напишем следующий текст (листинг 8).

#### ***Листинг 8. Процедура***

##### ***ПереченьНоменклатурыНоменклатура ПриИзменении()***

```
// Получить текущую строку табличной части
СтрокаТабличнойЧасти = ЭлементыФормы.ПереченьНоменклатуры.Текущие
Данные;

//Установить цену
СтрокаТабличнойЧасти.Цена = РаботаСоСправочниками.РозничнаяЦена(Дата,
Элемент.Значение);

// Пересчитать сумму строки
РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);
```

Проверим, как теперь работает данный документ. Запустим 1С:Предприятие в режиме отладки и откроем *регистр сведений Цены*. Для транзистора Philips добавим следующим числом новую цену (таблица 3).

**Таблица 3 – Изменение цены Транзистора Philips**

31.03.2010 0:00:00	Транзистор Philips	5,00
01.04.2010 0:00:00	Транзистор Philips	7,00

Теперь откроем *документ Оказание услуги № 1*. Этим документом мы как раз «израсходовали» один такой транзистор. Установим дату документа равной дате, когда было задано первое значение цены транзистора, и повторим выбор транзистора в колонке Номенклатура табличной части документа. Автоматические установится первое значение цены.

Теперь изменим дату документа на следующий день и снова повторим выбор транзистора. Будет установлено новое значение цены.

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

### **Объект конфигурации Перечисление**

До сих пор не обращалось внимание на то, что у нас нет никакого признака, по которому можно было бы сказать, чем является конкретный элемент справочника Номенклатура: *материалом* или *услугой*. То, что все элементы справочника разложены по некоторым группам, не может являться надежным критерием оценки: группы можно удалить, переименовать, сгруппировать элементы по другим принципам. Поэтому нам требуется некоторый признак, позволяющий однозначно определять принадлежность элемента справочника к материалам или услугам, независимо от изменения иерархической структуры справочника.

Создадим у *справочника Номенклатура* специальный реквизит, тип значения которого образуется новым объектом конфигурации *Перечисление*. Это поможет в дальнейшем легко определять, чем является элемент справочника Номенклатура: услугой или материалом. Кроме этого, скорректируем процедуру проведения *документа Оказание* услуги и покажем, как работать с перечислением средствами встроеного языка.

Объект конфигурации *Перечисление* является прикладным и предназначен для описания структуры хранения постоянных наборов значений, не изменяемых в процессе работы конфигурации. На основе объекта конфигурации Перечисление платформа создает в базе данных информационную структуру, в которой может храниться набор некоторых постоянных значений.

Набор всех возможных значений, которые содержит перечисление, задается при конфигурировании системы, и пользователь не может изменять их, удалять или добавлять новые. Из этого следует важная особенность перечисления: значения перечисления не «обезличены» для конфигурации, на них могут опираться алгоритмы работы программы.

## Реорганизация справочника Номенклатура

Откроем конфигуратор и создадим сначала новый объект конфигурации **Перечисление** с именем **ВидыНоменклатуры**. На закладке **Данные** добавим два значения перечисления: **Материал** и **Услуга**.

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

После этого запустим 1С:Предприятие в режиме отладки и зададим для каждого элемента справочника Номенклатура соответствующее значение реквизита **Вид номенклатуры** (Материал или Услуга).

Теперь посмотрим, как можно применить новые данные, полученные благодаря использованию перечисления **ВидыНоменклатуры**.

## Изменение процедуры проведения документа Оказание услуги

Доработаем документ Оказанием услуги таким образом, чтобы в регистре появлялись только записи, относящиеся к расходу материалов. Эта доработка позволит получить нужные данные в **регистре ОстаткиМатериалов**.

Скорректируем движения документа, исключив из обработки те строки табличной части, в которых находятся услуги. Для этого **обработчик события ОбработкаПроведения**, расположенный в **Модуле** документа **ОказаниеУслуги** на вкладке **Прочее**, добавим следующий текст (текст следует добавить в начало цикла обхода табличной части документа после строки **Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл** (листинг 9)).

### Листинг 9. Движения документа ОказаниеУслуги

```
Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры <>
Перечисления.ВидыНоменклатуры.Материал Тогда Продолжить;
КонецЕсли;
```

Запустим 1С:Предприятие в режиме отладки и проверим работу процедуры проведения документа Оказание услуги. Откроем **документ Оказание услуги № 1** и внесем в него следующие изменения (обратите внимание, что изменен не только состав номенклатуры в табличной части, но и время документа):

- номер – 1;
- дата – 31.03.2010;
- склад – основной;
- клиент – Иванов Михаил Юрьевич;
- мастер – Деловой Иван Сергеевич;
- подключение воды – 1 ед.; цена – 800,00; сумма – 800,00;
- шланг резиновый – 1 ед.; цена – 150,00; сумма – 150,00.

Перед тем как провести документ, откроем список **регистра Остатки материалов**, содержащий движения этого документа. Для этого выполним команду **Перейти ► Остатки материалов** из командной панели списка документа. Проведем документ и убедимся, что в движения по регистру Остатки материалов включаются только строки, содержащие материалы.

## Лабораторная работа 6. Проведение документа по нескольким регистрам

### Создание регистра Стоимость материалов

Регистр *СтоимостьМатериалов* будет иметь всего одно *измерение* – *Материал* с типом *СправочникСсылка.Номенклатура* и один *ресурс* – *Стоимость* с длиной *15* и точностью *2*.

Теперь можно приступить к внесению изменений в процедуры проведения документов. Начнем с самого простого документа *Приходная Накладная*.

### Изменение процедуры проведения документа Приходная накладная

Откроем в конфигураторе окно редактирования объекта конфигурации *Документ ПриходнаяНакладная* и перейдем на закладку *Движения*. В списке регистров отметим, что документ будет создавать теперь движения и по *регистру СтоимостьМатериалов*. Запустим конструктор движений и согласимся с тем, что существующая процедура *ОбработкаПроведения* будет замещена.

Перед нами откроется окно конструктора движений, которое будет содержать созданные нами ранее движения документа по регистру *ОстаткиМатериалов*. Добавим в список регистров, по которым формируются движения, еще один – *СтоимостьМатериалов*. Выберем для него ту же *табличную часть Материалы* и заполним выражения.

Для *ресурса Стоимость* выберем значения реквизита *табличной части Сумма*:

- регистры – РегистрНакопления.Стоимость Материалов;
- реквизиты документа:
  - дата;
  - склад;
  - ТекСтрокаМатериалы.Материал;
  - ТекСтрокаМатериалы.Количество;
  - ТекСтрокаМатериалы.Цена;
  - ТекСтрокаМатериалы.Сумма;
- тип движения регистра – Приход;
- табличная часть – Материалы;
- поле Материал – ТекСтрокаМатериалы.Материал;
- поле Стоимость – ТекСтрокаМатериалы.Сумма

Нажмем **ОК** и посмотрим на текст, который сформировал конструктор (листинг 10).

### Листинг 10. Процедура ОбработкаПроведения()

```
ПроцедураОбработкаПроведения(Отказ,Режим)
```

```
//{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

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

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

```
Для Каждого ТекСтрокаМатериалы Из Материалы Цикл
```



```
// Регистр ОстаткиМатериалов Приход
Движение = Движения.ОстаткиМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопленияПриход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаМатериалы.Материал;
Движение.Склад = Склад;
Движение.Количество = ТекСтрокаМатериалы.Количество;
КонецЦикла;
```

**Для Каждого ТекСтрокаМатериалы Из Материалы Цикл**

```
// Регистр СтоимостьМатериалов 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**

Как видите, конструктор создал два цикла обхода табличной части документа – отдельно для каждого регистра. Так происходит потому, что в общем случае документ может иметь несколько табличных частей, и информация, содержащаяся в каждой из них, может предназначаться для своего отдельного регистра.

В нашем случае табличная часть всего одна, поэтому можно объединить эти два цикла в один, закомментировав строки, выделенные жирным в листинге 10 (листинг 11).

### ***Листинг 11. Объединение двух циклов в один***

```
// КонецЦикла;
// Для Каждого ТекСтрокаМатериалы Из Материалы Цикл
```

Запустим 1С:Предприятие в режиме отладки и перепроведем документы ***Приходная накладная № 1*** и ***№ 2***.

Затем откроем первый документ и убедимся, что документ создает желаемые записи в регистрах накопления.

### **Изменение процедуры проведения документа *Оказание услуги***

Внесем изменения в процедуру обработки проведения ***документа Оказание Услуги***. На данном этапе будем исходить из пожелания, высказанного руководством организации, учет которой мы автоматизируем. Суть его заключается в том, что на первом этапе, при списании материалов, израсходованных в процессе оказания услуги, должна быть возможность указывать различную стоимость для одного и того же материала, которая рассчитана руководством исхо-

дя из текущих конъюнктурных соображений.

Поскольку в документе *ОказаниеУслуги* отражена только цена номенклатуры, нам понадобится добавить в табличную часть документа еще одно поле, в котором будет указываться стоимость номенклатуры.

Откроем в конфигураторе окно редактирования объекта конфигурации *Документ ОказаниеУслуги*, перейдем на закладку *Данные* и создадим новый *реквизит табличной части* документа с именем *Стоимость*, типом *Число*, длиной *15* и точностью *2*.

После этого откроем *форму ФормаДокумента* документа *ОказаниеУслуги* и добавим в табличное поле колонку, отображающую новый реквизит *Стоимость*, расположив ее после колонки *Номенклатура*. Для этого выделим табличное поле и выполним команду контекстного меню *Размещение данных*. В окне *Размещение данных* отметим *реквизит Стоимость* и нажмем *ОК*. После этого с помощью мыши перетащим *колонку Стоимость* после колонки *Номенклатура*.

Теперь создадим *движения документа Оказание Услуги* таким же образом, как это было сделано для документа *Приходная Накладная*.

Откроем в конфигураторе окно редактирования объекта конфигурации *Документ ОказаниеУслуги* и укажем, что он будет создавать *движения по регистру накопления Стоимость Материалов*.

Запустим конструктор движений документа и добавим в список регистров *регистр СтоимостьМатериалов*. Опишем движения документа следующим образом (обратите внимание, что стоимость вычисляется как произведение стоимости и количества, указанных в табличной части).

Нажмем *ОК* и в тексте, сформированном конструктором, восстановим изменения, внесенные нами ранее (не записывать движения, если номенклатура – не материал). Также объединим два цикла обхода табличной части документа в один (листинг 12).

### *Листинг 12. Процедура ОбработкаПроведения()*

Процедура ОбработкаПроведения(Отказ, Режим)

```
//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

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

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

```
Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
```

```
    Если ТекСтрокаПереченьНоменклатуры.Номенклатура. ВидНоменклатуры <> Перечисления.ВидыНоменклатуры.Материал Тогда  
        Продолжить;  
    КонецЕсли;
```

```
// Регистр ОстаткиМатериалов Расход  
    Движение = Движения.ОстаткиМатериалов.Добавить();
```

```
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;  
Движение.Период = Дата;  
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенкла-  
тура;  
Движение.Склад = Склад;  
Движение.Количество = ТекСтрокаПереченьНоменклатуры.Коли-  
чество;
```

```
// Регистр СтоимостьМатериалов Расход  
Движение = Движения.СтоимостьМатериалов.Добавить();  
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;  
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенкла-  
тура;  
Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.  
Стоимость*ТекСтрокаПереченьНоменклатуры.Количество;  
КонецЦикла;
```

```
// Записываем движения регистров  
Движения.ОстаткиМатериалов.Записать();  
Движения.СтоимостьМатериалов.Записать();  
//}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

КонецПроцедуры

Проверим, как теперь работает проведение документа ОказаниеУслуги.

Запустим 1С:Предприятие в режиме отладки и укажем стоимость выбран-  
ных материалов в документе **Оказание услуги № 1**:

- номер – 1;
- дата – 31.03.2010;
- склад – основной;
- клиент – Иванов Михаил Юрьевич;
- мастер – Деловой Иван Сергеевич;
- подключение воды – 1 ед.; цена – 800,00; сумма – 800,00;
- шланг резиновый – **стоимость – 100,00**; 1 ед.; цена – 150,00; сумма – 150,00.

Проведем документ Оказание услуги № 1 и посмотрим на движения этого документа по регистру Стоимость материалов.

Теперь создадим и проведем еще **два документа Оказание услуги**. Эти до-  
кументы понадобятся в дальнейшем, поэтому будьте внимательны и обратите  
внимание на то, что эти документы созданы другими датами.

**Оказание услуги № 2:**

- номер – 2;
- дата – 14.04.2010;
- склад – основной;
- клиент – Спиридонова Галина Петровна;
- мастер – Гусаков Николай Дмитриевич;

- ремонт импортного телевизора – 1 ед.; цена – 800,00; сумма – 800,00;
- строчный трансформатор Samsung – стоимость – 600,00; 1 ед.; цена – 900,00; сумма – 900,00.

### **Оказание услуги № 3:**

- номер – 3;
- дата – 13.04.2010;
- склад – основной;
- клиент – Гаврилов Роман Геннадьевич;
- мастер – Симонов Валерий Михайлович;
- подключение электричества – 1 ед.; цена – 800,00; сумма – 800,00;
- шланг резиновый – стоимость – 100,00; 2 ед.; цена – 150,00; сумма – 300,00;
- кабель электрический – стоимость – 20,00; 1 ед.; цена – 30,00; сумма – 30,00;
- ремонт отечественного телевизора – 1 ед.; цена – 600,00; сумма – 600,00;
- строчный трансформатор LG – стоимость – 270,00; 1 ед.; цена – 400,00; сумма – 400,00;
- транзистор Philips – стоимость – 3,00; 2 ед.; цена – 7,00; сумма – 14,00.

## **Лабораторная работа 7. Создание объекта конфигурации Регистр накопления (обороты)**

### **Создание оборотного регистра накопления Продажи**

Регистры накопления могут быть двух видов: *регистрами остатков* и *регистрами оборотов*.

Существующие в нашей учебной конфигурации регистры *ОстаткиМатериалов* и *СтоимостьМатериалов* являются регистрами остатков. Если помните, при создании отчета Материалы в конструкторе запроса мы видели, что для таких регистров система создает три виртуальные таблицы: таблица остатков, оборотов и совокупная таблица остатков и оборотов.

Оборотный регистр накопления очень похож на уже знакомый регистр остатков, для которого понятие «остаток» не имеет смысла. Оборотный регистр накапливает только обороты, остатки ему безразличны. Поэтому единственной виртуальной таблицей, которую будет создавать система для такого регистра, будет таблица оборотов. В остальном оборотный регистр ничем не отличается от регистра остатков.

Следует сказать об одной особенности конструирования регистров накопления, напрямую связанной с возможностью получения остатков. При создании *оборотного регистра накопления* нет особой сложности в определении того, какие именно параметры должны являться измерениями регистра – мы можем назначить в качестве его измерений любые нужные нам параметры.

Совсем иная ситуация в случае *регистра накопления, поддерживающего накопление остатков*. Для него выбор измерений должен выполняться исходя

из того, что движения регистра могут быть осуществлены в две стороны: приход и расход. Таким образом, в качестве измерений нужно выбирать те параметры, по которым движения точно будут осуществляться как в одну, так и в другую сторону.

Откроем конфигуратор и создадим новый объект конфигурации **Регистр накопления**. Назовем его **Продажи** и определим **вид регистра Обороты**.

На закладке **Данные** создадим **измерения** регистра:

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

У регистра будет **три ресурса**:

- **Количество**, тип **Число**, длина **15**, точность **3**;
- **Выручка**, тип **Число**, длина **15**, точность **2**;
- **Стоимость**, тип **Число**, длина **15**, точность **2**.

Откроем окно редактирования объекта конфигурации **Документ ОказаниеУслуги** и на закладке **Движения** укажем, что этот документ будет создавать **движения по регистру Продажи**.

Запустим 1С:Предприятие в режиме отладки и откроем формы списка регистров накопления Продажи и Остатки материалов. Обратите внимание, что формы практически одинаковы, за исключением состава измерений и ресурсов.

### **Изменение процедуры проведения документа Оказание услуги**

На этот раз не будем использовать конструктор движения документа, а внесем необходимые дополнения прямо в обработчик события **ОбработкаПроведения документа ОказаниеУслуги**.

Откроем в конфигураторе модуль объекта конфигурации **Документ ОказаниеУслуги** и найдем в нем процедуру обработчика события **ОбработкаПроведения**. Сразу после окончания первого цикла создадим еще один цикл обхода табличной части и команду записи **движений регистра Продажи** (листинг 13).

#### **Листинг 13. Добавление цикла обхода табличной части и записи движений регистра Продажи**

```
...
КонецЦикла;
Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры
Цикл
    // Регистр Продажи

КонецЦикла;
//}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

Теперь в тело созданного цикла вставим команды создания движений регистра Продажи (листинг 14).

## *Листинг 14. Команды создания движений регистра Продажи*

Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры  
Цикл

    //Регистр Продажи

    Движение = Движения.Продажи.Добавить();

    Движение.Период = Дата;

    Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенкла-  
тура;

    Движение.Клиент = Клиент;

    Движение.Мастер = Мастер;

    Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;

    Движение.Выручка = ТекСтрокаПереченьНоменклатуры.Сумма;

    Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.  
Стоимость\*ТекСтрокаПереченьНоменклатуры.Количество;

КонецЦикла;

Обратите внимание на то, что у оборотного регистра отсутствует свойство ВидДвижения, поскольку отражение вида движения (приход или расход) имеет смысл лишь при учете остатков. В случае регистра оборотов интерес представляет только значение, которое должно быть записано в ресурс регистра.

Запустим 1С:Предприятие в режиме отладки и перепроведем все *документы Оказание услуги*.

## **Лабораторная работа 8. Создание отчетов**

### **Отчет Реестр документов Оказание услуги**

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.

Создадим в конфигураторе новый *объект конфигурации Отчет*. Присвоим ему имя *РеестрДокументовОказаниеУслуги*. На закладке *Основные* нажмем кнопку *Открыть схему компоновки данных*. В открывшемся диалоговом окне конструктора макета нажмем *Готово*. В конструкторе схемы компоновки данных создадим *Набор данных – запрос* и запустим конструктор запроса.

В качестве источника данных для запроса выберем объектную (ссылочную) *таблицу* документов *ОказаниеУслуги*. Из этой таблицы выберем следующие поля:

- *Дата*;
- *Номер*;
- *Склад*;
- *Мастер*;
- *Клиент*.

После этого перейдем на закладку *Порядок* и укажем, что результат запроса должен быть сначала упорядочен по значению *поля Дата*, а затем – по значению *поля ОказаниеУслуги.Ссылка*. Нажмем *ОК* и посмотрим, какой запрос сформировал конструктор запроса (листинг 15).

### *Листинг 15. Текст запроса*

ВЫБРАТЬ

ОказаниеУслуги.Дата КАК Дата,  
ОказаниеУслуги.Номер,  
ОказаниеУслуги.Склад,  
ОказаниеУслуги.Мастер,  
ОказаниеУслуги.Клиент

ИЗ

Документ.ОказаниеУслуги КАК ОказаниеУслуги

УПОРЯДОЧИТЬ ПО

Дата,  
ОказаниеУслуги.Ссылка

Перейдем на закладку *Настройки* и добавим *новую группировку* в структуру отчета. В окне выбора поля группировки просто нажмем *ОК* (тем самым укажем, что в группировке будут выводиться детальные записи из информационной базы) и на закладке *Выбранные* зададим поля, которые будут выводиться в отчет:

- *Дата*;
- *Номер*;
- *Склад*;
- *Мастер*;
- *Клиент*.

Теперь запустим 1С:Предприятие в режиме отладки и откроем *отчет Реестр документов оказание услуги*. Нажмем *Сформировать* и посмотрим на результат работы отчета.

### **Отчет Рейтинг услуг**

*Отчет Рейтинг услуг* будет содержать информацию о том, выполнение каких услуг принесло организации наибольшую прибыль в указанном периоде. На примере отчета Рейтинг услуг проиллюстрируем, как отбирать данные в некотором периоде, как задавать параметры запроса, как использовать в запросе данные из нескольких таблиц и как включать в результат запроса все данные одного из источников.

Создадим новый объект конфигурации *Отчет*. Назовем его *РейтингУслуг*. На закладке *Основные* нажмем кнопку *Открыть схему компоновки данных*. В открывшемся диалоговом окне конструктора макета нажмем *Готово*.

В конструкторе схемы компоновки данных создадим *Набор данных – за-*

*прос* и запустим конструктор запроса. Выберем объектную (ссылочную) *таблицу справочника Номенклатура* и виртуальную *таблицу регистра накопления Продажи.Обороты*. Для того чтобы исключить неоднозначность имен в запросе, переименуем *таблицу Номенклатура* в *СпрНоменклатура* (контекстное меню правой кнопки мыши – *Переименовать таблицу*).

Затем выберем из таблиц *поля СпрНоменклатура.Ссылка* и *Продажи.Обороты.ВыручкаОборот*.

Перейдем на закладку *Связи* и увидим, что конструктор уже создал *связь* между двумя выбранными таблицами – *значение изменения регистра Номенклатура должно быть равно ссылке на элемент справочника Номенклатура*.

Единственное, что остается сделать, это сбросить *флаг Все* у *таблицы регистра* и установить его у *таблицы справочника*.

Установка флага *Все* у *таблицы справочника* будет означать, что из справочника будут выбраны все элементы и этим элементам будет поставлено в соответствие значение оборота выручки из регистра. Таким образом, в результате запроса будут присутствовать все услуги, и для некоторых из них будут указаны обороты выручки. Для тех услуг, которые не оказывались в выбранном периоде, не указано будет ничего.

Перейдем на закладку *Условия* и зададим условия выбора элементов из справочника *Номенклатура*. При задании условий выбора будем использовать параметры запроса. *Первым условием* должно быть то, что выбранный элемент не является группой (для этого следует переключиться в режим *Произвольное условие* – установить флаг *Произвольное*). Затем ввести в поле *Условие* следующий текст (листинг 16).

#### *Листинг 16. Условие запроса*

СпрНоменклатура.ЭтоГруппа = ЛОЖЬ

*Вторым условием* должно быть то, что выбранный элемент является услугой (это – Простое условие).

В дальнейшем, перед выполнением запроса, необходимо будет передать в *параметр ВидНоменклатуры* соответствующее значение *перечисления*.

Перейдем на закладку *Объединения/Псевдонимы* и укажем, что представление элемента справочника будет иметь *псевдоним Услуга*, а поле регистра будет иметь *псевдоним Выручка*.

Перейдем на закладку *Порядок* и укажем, что результат запроса должен быть отсортирован *по убыванию значения поля Выручка*. Создание запроса закончено, нажмем кнопку *ОК*. Рассмотрим запрос, сформированный конструктором (листинг 17).

#### *Листинг 17. Текст запроса*

ВЫБРАТЬ

СпрНоменклатура.Ссылка КАК Услуга,  
ПродажиОбороты.ВыручкаОборот КАК Выручка



ИЗ

Справочник.Номенклатура КАК СпрНоменклатура  
ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.Продажи.Обороты КАК Про-  
дажиОбороты

ПО ПродажиОбороты.Номенклатура = СпрНоменклатура,Ссылка

ГДЕ

СпрНоменклатура.ЭтоГруппа = ЛОЖЬ

И СпрНоменклатура.ВидНоменклатуры = &ВидНоменклатуры

УПОРЯДОЧИТЬ ПО

Выручка УБЫВ

Перейдем на закладку **Ресурсы** и нажмем кнопку >>, чтобы конструктор выбрал все доступные ресурсы, по которым можно вычислять итоги. В данном случае – это единственный **ресурс Выручка**.

Перейдем на закладку **Параметры**. На этой закладке присутствуют три параметра: **НачалоПериода**, **КонецПериода** и **ВидНоменклатуры**. Первым параметром передается начало периода расчета итогов, вторым – конец периода. В результате исходная таблица будет содержать только обороты, рассчитанные в переданном периоде.

Для **параметра НачалоПериода** зададим заголовок, который будет отображаться пользователю – **Дата начала**.

Здесь всегда следует помнить, что если мы передаем в качестве этих параметров дату (а в нашем случае так и будет), то дата содержит и время с точностью до секунды.

Допустим, заранее известно, что пользователя не будут интересовать результаты работы отчета в периодах, указанных с точностью до секунды. В этом случае следует учесть две особенности. Во-первых, пользователя надо избавить от необходимости указывать время при вводе даты. Для этого изменим существующее описание типа для **параметра НачалоПериода**. Дважды щелкнем в ячейке **Доступные типы**, соответствующей **параметру НачалоПериода**, нажмем кнопку выбора (...) и в нижней части окна редактирования типа данных, установим **Состав даты** в значении **Дата**. Нажмем **ОК**.

Вторая особенность заключается в том, что по умолчанию время в дате установлено 00:00:00. Поэтому, если пользователь задаст период отчета с 01.03.2012 по 31.03.2012, итоги регистра будут рассчитаны с начала дня 01.03.2012 00:00:00 по начало дня 31.03.2012 00:00:00. Таким образом, данные за 31 число, отличные от начала дня, в расчет не войдут.

Для того чтобы исключить эту ситуацию, добавим еще один параметр, в который пользователь будет вводить дату окончания. Значение **параметра КонецПериода** будем рассчитывать автоматически таким образом, чтобы оно указывало на конец дня даты, введенной пользователем. Поэтому для **параметра КонецПериода** установим **флаг Ограничение доступности**.

С помощью кнопки командной панели добавим новый **параметр** с именем **ДатаОкончания**. Для этого параметра платформа автоматически сформирует

заголовок *Дата окончания*. Оставим его без изменений. Зададим *тип параметра* – *Дата*. При этом, как и для параметра НачалоПериода, укажем *состав даты* – *Дата*. По умолчанию добавленный параметр доступен для пользователя (ограничение доступности в последней колонке снято). Перейдем к *параметру КонецПериода*. Для него система установила флаг ограничения доступности. Это правильно, поскольку значение этого параметра мы собираемся вычислять на основании значения, установленного пользователем для параметра ДатаОкончания.

Чтобы задать формулу, по которой будет вычисляться значение параметра КонецПериода, воспользуемся языком выражений системы компоновки данных. Он содержит *функцию КонецПериода()*, которая позволяет получить дату, соответствующую концу какого-либо периода, например, указанного дня.

В *ячейке Выражение* зададим для *параметра КонецПериода* следующее выражение (листинг 18).

### *Листинг 18. Выражение для расчета значения параметра КонецПериода*

КонецПериода(&ДатаОкончания, "День")

И в заключение 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 компоновки как *Перечисление.ВидыНоменклатуры.Услуга*. Флаг ограничения использования у параметра ВидНоменклатуры платформа установила по умолчанию, поэтому осталось только указать нужное значение *перечисления ВидыНоменклатуры* в ячейке *Значение*, соответствующей *параметру ВидНоменклатуры*.

Перейдем к формированию структуры отчета. На закладке *Настройки* добавим *группировку* и снова не укажем поле группировки. На закладке *Выбранные поля* укажем *поля Услуга* и *Выручка*.

В заключение перейдем на закладку *Другие настройки* и зададим *заголовок отчета* – *РейтингУслуг*. Запустим 1С:Предприятие в режиме отладки.

Запустим отчет Рейтинг услуг. Но перед тем, как нажать кнопку *Сформировать*, откроем *Настройки* и на закладке *Параметры* данных зададим период отчета с 01.03.2012 по 30.04.2012. Нажмем *ОК* и сформируем отчет. Теперь изменим дату окончания на 31.03.2012. Данные марта попадают в отчет.

### **Отчет Выручка мастеров**

*Отчет Выручка мастеров* будет содержать информацию о том, какая выручка была получена организацией благодаря работе каждого из мастеров, с детализацией по всем дням в выбранном периоде и разворотом по клиентам, обслуженным в каждый из дней. На примере этого отчета проиллюстрируем, как строить многоуровневые группировки в запросе и как обходить все даты в выбранном периоде.

Создадим новый объект конфигурации *Отчет*. Назовем его *ВыручкаМа-*

*стеров* и запустим конструктор основной схемы компоновки данных.

Создадим *Набор данных – запрос* и запустим конструктор запроса. Выберем виртуальную таблицу *регистра накопления Продажи.Обороты*.

Зададим один из параметров этой виртуальной таблицы – *Периодичность*. Для этого перейдем в *поле Таблицы* и нажмем кнопку *Параметры виртуальной таблицы*.

В открывшемся окне параметров зададим значение *параметра Периодичность – День*. Нажмем *ОК*. После этого выберем из таблицы следующие поля:

- *ПродажиОбороты.Мастер*;
- *ПродажиОбороты.Период*;
- *ПродажиОбороты.Клиент*;
- *ПродажиОбороты.ВыручкаОборот*.

Теперь перейдем на закладку *Объединения/Псевдонимы* и зададим *псевдоним Выручка* для поля *ПродажиОбороты.ВыручкаОборот*.

На закладке *Группировка* определим, что группировка будет проводиться по *полям Мастер, Период и Клиент*, а *значения поля ВыручкаОборот* будут суммироваться.

На закладке *Порядок* определим, что результат запроса будет отсортирован *по возрастанию значения поля Период*. Нажмем *ОК*. Рассмотрим текст запроса, сформированный конструктором (листинг 19).

#### *Листинг 19. Текст запроса*

ВЫБРАТЬ

ПродажиОбороты.Мастер,  
ПродажиОбороты.Период КАК Период,  
ПродажиОбороты.Клиент,  
СУММА(ПродажиОбороты.ВыручкаОборот) КАК Выручка

ИЗ

РегистрНакопления.ПродажиОбороты (, , День,) КАК ПродажиОбороты

СГРУППИРОВАТЬ ПО

ПродажиОбороты.Мастер,  
ПродажиОбороты.Период,  
ПродажиОбороты.Клиент

УПОРЯДОЧИТЬ ПО

Период

Перейдем к редактированию схемы компоновки данных. На закладке *Ресурсы* нажмем >> и убедимся, что конструктор выбрал единственный имеющийся у нас *ресурс – Выручка*.

На закладке *Параметры* выполним те же действия, что и при создании предыдущего отчета.

Для параметра *НачалоПериода* зададим заголовок *Дата начала*. В поле *Доступные типы* зададим состав даты – *Дата*.

Для параметра *КонецПериода* зададим выражение (листинг 20) и в поле *Ограничение доступности* установим *флаг ограничения доступности*.

## **Листинг 20. Выражение для расчета параметра КонецПериода** КонецПериода(&ДатаОкончания, "День")

В заключение добавим еще один параметр – *ДатаОкончания*. Установим его *тип* как *Дата, состав даты – Дата*.

Теперь создадим структуру отчета. На закладке *Настройки* последовательно создадим *две вложенные группировки*: верхнего уровня – *по полю Мастер*, вложенная в нее – *по полю Период*. Затем добавим *еще одну группировку*, вложенную в группировку *по полю Период*, – *Детальные записи* (без указания группировочного поля).

Теперь, находясь *на уровне Глобального отчета*, перейдем на закладку *Выбранные поля* и добавим в список выбранных полей *поля Клиент* и *Выручка*.

В заключение с уровня Глобального отчета перейдем на закладку *Другие настройки* и изменим следующие параметры. Для параметра *Расположение полей группировок* установим значение *Отдельно и только в итогах*; для параметра *Расположение общих итогов по вертикали* зададим значение *Начало* и для параметра *Заголовок* зададим значение *Выручка мастеров*.

Запустим 1С:Предприятие в режиме отладки и посмотрим результат работы *отчета Выручка мастеров* за период с 01.03.2012 по 30.04.2012.

В начале раздела мы говорили, что этот отчет должен показывать данные с детализацией по всем дням в выбранном периоде. У нас же отображаются только те дни, для которых существуют ненулевые записи в таблице регистра накопления. Поэтому необходимо изменить настройки отчета таким образом, чтобы в отчет попадала каждая дата из периода, за который сформирован отчет.

Для этого вернемся в режим конфигуратора и выполним дополнительную настройку структуры отчета. До сих пор все настройки структуры, которые мы выполняли, относились ко всему отчету в целом. Но система компоновки данных позволяет настраивать также и каждый элемент структуры в отдельности. В данном случае потребуется изменить настройку *группировки Период*. Для того чтобы перейти к настройкам именно этой группировки, в поле структуры *установим курсор на эту группировку*, а затем нажмем *кнопку Период* в командной панели.

В нижней части формы будут отображены настройки, доступные для данной группировки. Перейдем на закладку *Поля группировки*. Для *поля Период* установим *Тип дополнения – День*. Тем самым укажем, что для этой группировки существующие записи с ненулевым значением ресурса будут дополняться записями для каждого из дней. После этого следует указать, в каком именно периоде будет выполняться такое дополнение. В поля, расположенные строчкой ниже, можно ввести даты начала и окончания этого периода. Но указание дат в явном виде нас не устраивает, так как пользователь может сформировать отчет за произвольный период. И нам нужно, чтобы дополнение дат выполнялось не в некотором фиксированном периоде, а именно в том периоде, который выбрал пользователь для всего отчета. Для того чтобы обеспечить именно такую работу отчета, войдем в режим редактирования первого поля (например,

дважды щелкнув на нем) и нажмем на кнопку очистки (**X**). После этого, нажав на кнопку **T**, мы сможем выбрать тип данных, отображаемых в этом поле. Выберем **Поле компоновки данных**. Нажмем **ОК**.

Теперь нажмем в поле ввода кнопку выбора (...) и в открывшемся окне выбора отметим **параметр НачалоПериода**. Нажмем **ОК**.

Для второго поля ввода аналогичным образом укажем, что дата окончания периода будет получена из **параметра ДатаОкончания**.

Запустим 1С:Предприятие в режиме отладки и выполним отчет Выручка мастеров за период с 20.03.2012 по 20.04.2012.

### **Отчет Перечень услуг**

**Отчет Перечень услуг** будет содержать информацию о том, какие услуги и по какой цене оказывает организация. На его примере познакомимся с возможностью получения последних значений из периодического регистра сведений и вывода иерархий справочников.

Создадим новый объект конфигурации **Отчет**. Назовем его **ПереченьУслуг** и запустим конструктор схемы компоновки данных. Создадим новый **Набор данных – запрос** и вызовем конструктор запроса. Выберем объектную (ссылочную) **таблицу справочника Номенклатура** и виртуальную **таблицу регистра сведений ЦеныСрезПоследних**. Для того чтобы исключить неоднозначность имен в запросе, переименуем таблицу **Номенклатура** в **СпрНоменклатура**.

Вызовем диалог ввода параметров виртуальной **таблицы ЦеныСрезПоследних** и укажем, что период будет передан в **параметре ДатаОтчета**. Затем выберем из таблиц следующие поля:

- **СпрНоменклатура.Родитель**;
- **СпрНоменклатура.Ссылка**;
- **ЦеныСрезПоследних.Цена**.

Перейдем на закладку **Связи**, сбросим **флаг Все** у таблицы **регистра** и установим его у таблицы **справочника**.

На закладке **Условия** зададим условие выбора элементов **справочника Номенклатура** – выбираемые элементы должны соответствовать виду номенклатуры, переданному в параметре запроса Вид Номенклатуры.

На закладке **Объединения/Псевдонимы** укажем, что **поле Родитель** будет иметь **псевдоним ГруппаУслуг**, а **поле Ссылка** – **Услуга**.

Перейдем на закладку **Группировка** и укажем, что группировка будет производиться **по полю СпрНоменклатура.Родитель**. Значения суммируемых полей задавать не станем. На этом создание запроса завершено, нажмем **ОК** и рассмотрим текст запроса, сформированный конструктором (листинг 21).

### **Листинг 21. Текст запроса**

**ВЫБРАТЬ**

СпрНоменклатура.Родитель КАК ГруппаУслуг,  
СпрНоменклатура.Ссылка КАК Услуга,  
ЦеныСрезПоследних.Цена

**ИЗ**

Справочник.Номенклатура КАК СпрНоменклатура  
ЛЕВОЕ СОЕДИНЕНИЕ  
РегистрСведений.Цены.СрезПоследних(&ДатаОтчета,) КАК ЦеныСрез-  
Последних  
ПО ЦеныСрезПоследнихНоменклатура = СпрНоменклатура.Ссылка  
ГДЕ  
СпрНоменклатура.ВидНоменклатуры = &ВидНоменклатуры  
СГРУППИРОВАТЬ ПО  
СпрНоменклатура.Родитель,  
СпрНоменклатура.Ссылка,  
ЦеныСрезПоследних.Цена

Перейдем к редактированию схемы компоновки данных. На закладке *Ресурсы* нажатием кнопки >> выберем единственный доступный *ресурс Цена*.

На закладке *Параметры* зададим значение *параметра ВидНоменклатуры* как *Перечисление.ВидыНоменклатуры.Услуга*. Кроме этого, снимем ограничение доступности для *параметра ДатаОтчета* и зададим ему *заголовок – Дата отчета*. В поле *Доступные типы* зададим *состав даты – Дата*. Для параметра *Период*, наоборот, установим ограничение доступности.

Приступим к созданию структуры отчета. Перейдем на закладку *Настройки* и создадим группировку *по полю ГруппаУслуг*, указав тип группировки *Иерархия*. Внутри этой группировки создадим еще одну группировку без указания группового поля. Она будет содержать детальные записи отчета.

Перейдем на закладку *Выбранные поля* и укажем, что в отчет будут выводиться *поля Услуга и Цена*.

И в заключение настроим внешний вид отчета на закладке *Другие настройки*. Так как наш отчет будет представлять собой просто список оказываемых услуг, в котором интересны цены на конкретные услуги, выводить значения ресурса *Цена* для каждой из группировок и для всего отчета в целом не имеет смысла. Чтобы запретить вывод общих итогов в отчете, установим *параметр Расположение общих итогов по вертикали* в значение *Нет*.

Затем перейдем к настройкам конкретной *группировки – ГруппаУслуг*. Для параметра *Расположение итогов* этой группировки укажем значение *Нет*. Вернемся к настройкам всего отчета в целом.

Для параметра *Расположение полей группировок* укажем значение *Отдельно и только в итогах* (так наш отчет будет лучше «читаться»). Напоследок зададим *заголовок отчета – Перечень услуг*.

Теперь запустим 1С:Предприятие в режиме отладки и прежде всего откроем периодический регистр *Цены*.

Добавим в него еще одно значение для услуги *Диагностика*: новая цена услуги на 01.04.2012 – 350. Это позволит протестировать отчет.

Теперь выполним *отчет Перечень услуг* по состоянию на 31.03.2012. Отчет должен отражать цену услуги *Диагностика* на 31.03.2012 – **200 руб.** Еще раз выполним отчет, но теперь уже на другую дату – 01.04.2012. Должна быть показана новая цена услуги *Диагностика* – **350 руб.**

## Отчет Рейтинг клиентов

*Отчет Рейтинг клиентов* будет показывать, каков доход от указания услуг каждому из клиентов за все время работы организации. На его примере продемонстрируем возможность использования диаграммы для отображения результата запроса.

Диаграмма является элементом управления, предназначенным для размещения в таблицах и формах диаграмм и графиков различного вида системы 1С:Предприятие. Логически диаграмма является совокупностью точек, серий и значений серий в точке. Как правило, в качестве точек используются моменты или объекты, для которых мы получаем значения характеристик, а в качестве серий – характеристики, значения которых нас интересуют. Например, диаграмма продаж видов номенклатуры по месяцам будет состоять из точек – месяцев, серий – видов номенклатур и значений – оборотов продаж.

Диаграмма может быть вставлена в структуру отчета как отдельный элемент. В следующем отчете будем использовать диаграмму в структуре настроек схемы компоновки данных.

Создадим в конфигураторе новый объект конфигурации *Отчет*. Назовем его *Рейтинг клиентов*, откроем его основную схему компоновки данных.

Создадим *Набор данных – запрос*, вызовем конструктор запроса и выберем виртуальную таблицу *регистра накопления Продажи.Обороты* и из нее одно поле – *ПродажиОбороты.Клиент*.

Затем добавим новое поле (значок *Добавить* в командной панели над списком полей) и при помощи *построителя выражений* определим его как *разность между выручкой и стоимостью*.

На закладке *Объединения/Псевдонимы* укажем, что вычисляемое поле будет иметь псевдоним *Доход*.

На закладке *Порядок* укажем, что строки результата нужно упорядочивать *по убыванию значения поля Доход*. Нажмем *ОК* и посмотрим, какой текст сформировал конструктор запроса (листинг 22).

### Листинг 22. Текст запроса

ВЫБРАТЬ

ПродажиОбороты.Клиент,

ПродажиОбороты.ВыручкаОборот – ПродажиОбороты.СтоимостьОборот  
КАК Доход

ИЗ

РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты

УПОРЯДОЧИТЬ ПО

Доход УБЫВ

Добавим поле *Доход* в ресурсы схемы компоновки и перейдем на закладку *Настройки* для того, чтобы создать структуру отчета.

В отличие от всех предыдущих отчетов, структура которых содержала группировки, в этот раз добавим в структуру отчета диаграмму. В точки диаграммы добавим *группировку по полю Клиент*. Серии диаграммы оставим без

изменений.

Перейдем на закладку **Выбранные поля** и выберем **поле Доход** для вывода в отчет.

На закладке **Другие настройки** зададим **заголовок отчета – РейтингКлиентов**, а также выберем **тип диаграммы Круговая объемная**. Запустим 1С:Предприятие в режиме отладки и откроем **отчет Рейтинг клиентов**. Обратите внимание, что при наведении курсора на сектор диаграммы появляется подсказка.

Откроем настройки отчета и изменим **тип диаграммы** на **Измерительная**. Заново сформируем отчет. Таким образом, можно продемонстрировать, как можно использовать различные виды диаграмм для визуализации данных отчетов.

### **Универсальный отчет**

На примере создания универсального отчета узнаем, как система компоновки данных может взаимодействовать со сводной таблицей.

Создадим в конфигураторе новый объект конфигурации Отчет. Присвоим ему имя **Универсальный**. Откроем его основную схему компоновки данных и создадим новый **Набор данных – запрос**.

Вызовем конструктор запроса и выберем виртуальную таблицу регистра накопления **Продажи.Обороты**. Из этой таблицы выберем все поля:

- ПродажиОбороты.Номенклатура;
- ПродажиОбороты.Клиент;
- ПродажиОбороты.Мастер;
- ПродажиОбороты.КоличествоОборот;
- ПродажиОбороты.ВыручкаОборот;
- ПродажиОбороты.СтоимостьОборот.

На закладке **Группировка** с помощью кнопок >> выберем все поля, доступные для группировки (Клиент, Мастер, Номенклатура), а все ресурсы добавим в **Суммируемые поля** (ВыручкаОборот, КоличествоОборот, СтоимостьОборот).

Нажмем **ОК** и посмотрим на текст, сформированный конструктором запроса (листинг 23).

### **Листинг 23. Текст запроса**

**ВЫБРАТЬ**

```
ПродажиОбороты.Номенклатура,  
ПродажиОбороты.Клиент,  
ПродажиОбороты.Мастер,  
СУММА(ПродажиОбороты.КоличествоОборот) КАК КоличествоОборот,  
СУММА(ПродажиОбороты.ВыручкаОборот) КАК ВыручкаОборот,  
СУММА(ПродажиОбороты.СтоимостьОборот) КАК СтоимостьОборот
```

**ИЗ**

```
РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты
```

**СГРУППИРОВАТЬ ПО**



ПродажиОбороты.Клиент,  
ПродажиОбороты.Мастер,  
ПродажиОбороты.Номенклатура

Закончим создание схемы компоновки данных тем, что на закладке конструктора схемы компоновки **Ресурсы** выберем все доступные ресурсы (нажмем >>). На этом работа со схемой компоновки данных завершена, закроем ее и вернемся к окну редактирования объекта конфигурации **Отчет Универсальный**.

Перейдем на закладку **Формы** и создадим основную форму отчета, нажав на кнопку просмотра у табличного поля **Основная форма отчета**. Так как основная форма пока отсутствует, система вызовет конструктор форм. Согласимся с тем, что система предлагает по умолчанию, и сразу нажмем **Готово**.

На экране откроется основная форма отчета. Система уже вставила поле табличного документа, присвоив ему имя – **Результат**.

Необходимо вставить в это поле сводную таблицу, но ячейки поля табличного документа Результат находятся в режиме **Только просмотр**. Для того чтобы вставить сводную таблицу, нужно снять этот режим, поэтому щелкнем на поле табличного документа и выполним **Таблица ► Вид ► Только просмотр**. Теперь становится возможным редактировать табличное поле.

Установим курсор в верхнюю левую ячейку поля табличного документа и выполним **Таблица ► Встроенные таблицы ► Вставить сводную таблицу**. Система добавит в табличное поле сводную таблицу и откроет окно полей сводной таблицы.

Для того чтобы в сводную таблицу вывести данные, потребуется написать несколько строк в модуле формы. Откроем модуль формы отчета и добавим в текст переменную (листинг 24).

#### **Листинг 24. Объявление переменной**

Перем ИсточникСводнойТаблицы;

После этого создадим обработчик события формы отчета ПриОткрытии и добавим в него следующий текст (листинг 25).

#### **Листинг 25. Обработчик события формы отчета ПриОткрытии**

Процедура ПриОткрытии()

ИсточникСводнойТаблицы = Новый ИсточникДанныхСводнойТаблицыКомпоновкиДанных;

ИсточникСводнойТаблицы.УстановитьСхему(СхемаКомпоновкиДанных);

ЭлементыФормы.Результат.ВстроенныеТаблицы.СводнаяТаблица1.ИсточникДанных = ИсточникСводнойТаблицы;

ИсточникСводнойТаблицы.УстановитьНастройки(КомпоновщикНастроек.-Настройки);

КонецПроцедуры

Этим текстом сводной таблице в качестве источника данных присваивается схема компоновки данных и устанавливаются для нее настройки компоновщика настроек.

Запустим 1С:Предприятие в режиме отладки и откроем *отчет Универсальный*. На экране появится форма отчета и окно выбора полей сводной таблицы.

Поместим значение *ресурса ВыручкаОборот* в данные, *измерение Номенклатура* – в строки, а *измерение Мастер* – в колонки. Теперь в окне выбора полей сводной таблицы раскроем *группу Номенклатура* и добавим значение *(Без иерархии)* в строки, а *измерение Клиент* добавим в колонки.

Таким образом, используя эти схемы компоновки данных, пользователю предоставляется возможность самостоятельно формировать отчет по регистру *Оказанные Услуги*.

## Список литературы

- 1 **Васина, Е. Н.** Автоматизированные информационные системы бухгалтерского учета / Е. Н. Васина, Т. Л. Партыка, И. И. Попов. – М. : Форум, 2006. – 432 с.
- 2 **Власова, М. А.** 1С:Бухгалтерия 8 КОРП. Учет в головной организации / М. А. Власова, С. А. Харитонов. – М. : ООО «1С-Пабблишинг» ; СПб. : Питер, 2009. – 528 с.
- 3 **Гридасов, А. Ю.** Бухгалтерский учет в программе 1С:Бухгалтерия 8.0. Лабораторный практикум / А. Ю. Гридасов, А. Г. Чурин, Л. И. Чурина. – М.: КНОРУС, 2009. – 216 с.
- 4 **Радченко, М. Г.** 1С:Предприятие 8.1. Практическое пособие для разработчика. Примеры и типовые приемы / М. Г. Радченко. – М. : ООО «1С-Пабблишинг» ; СПб. : Питер, 2007. – 512 с.
- 5 **Самарина, Е. В.** Секреты профессиональной работы с «1С:Бухгалтерией 8». Учет производственных операций / Е. В. Самарина, С. А. Харитонов, Д. В. Чистов. – М. : ООО «1С-Пабблишинг» ; СПб. : Питер, 2009. – 384 с.
- 6 **Соколова, Г. Н.** Информационные технологии экономического анализа / Г. Н Соколова. – М. : Экзамен, 2002. – 320 с.
- 7 **Федорова, Г. В.** Информационные технологии бухгалтерского учета, анализа и аудита / Г. В. Федорова. – М. : Омега-Л, 2009. – 296 с.
- 8 **Харитонов, С. А.** Бухгалтерский и налоговый учет в 1С:Бухгалтерии 8 / С. А. Харитонов. – СПб. : Питер, 2009. – 596 с.
- 9 **Харитонов, С. А.** Настольная книга по оплате труда и ее расчету в программе «1С:Зарплата и управление персоналом 8» / С. А. Харитонов. – М. : ООО «1С-Пабблишинг» ; СПб. : Питер, 2009. – 616 с.
- 10 **Харитонов, С. А.** Секреты профессиональной работы с «1С:Бухгалтерией 8». Банк и касса / С. А. Харитонов, Д. В. Чистов. – М. : ООО «1С-Пабблишинг» ; СПб. : Питер, 2008. – 324 с.
- 11 **Харитонов, С. А.** Секреты профессиональной работы с «1С:Бухгалтерией 8». Учет основных средств / С. А. Харитонов, Д. В. Чистов – М. : ООО «1С-Пабблишинг» ; СПб. : Питер, 2008. – 368 с.
- 12 **Чистов, Д. В.** Хозяйственные операции в 1С:Бухгалтерии 8. Задачи, решения, результаты / Д. В. Чистов, С. А. Харитонов. – М. : ООО «1С-Пабблишинг» ; СПб. : Питер, 2009. – 336 с.
- 13 **Шуремов, Е. Л.** Автоматизированные информационные системы бухгалтерского учета, анализа, аудита / Е. Л. Шуремов, Э. А. Умнова, Т. В. Ворopaева. – М. : Перспектива, 2005. – 363 с.
- 14 **Шуремов, Е. Л.** Информационные системы управления предприятиями / Е. Л. Шуремов, Д. В. Чистов, Г. В. Лямова. – М. : Бухгалтерский учет, 2006. – 112 с.
- 15 **1С:Бухгалтерия 8.** Учебная версия. – М. : ООО «1С-Пабблишинг», 2009. – 685 с.

Воробьева Ольга Алимовна

**Автоматизированные системы  
бухгалтерского и налогового учета и аудита**

Методические указания  
к выполнению лабораторных работ  
студентами очной формы обучения  
специальности 080109 «Бухгалтерский учет, анализ и аудит»

Редактор Е.А. Устюгова

Подписано к печати	Формат 60 x 84 <sup>1</sup> / <sub>16</sub>	Бумага тип. № 1
Печать трафаретная	Усл. печ.л. 2,75	Уч. изд.л. 2,75
Заказ	Тираж 30	Цена свободная

РИЦ Курганского государственного университета.  
640669, г. Курган, ул. Гоголя, 25.  
Курганский государственный университет.