

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное бюджетное образовательное
учреждение
высшего образования**
**«Пермский государственный национальный исследовательский
университет»**

Колледж профессионального образования

1С программирование

Методические рекомендации

для практических работ по изучению дисциплины

для студентов Колледжа профессионального образования

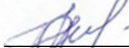
специальности

09.02.03 Программирование в компьютерных системах

Утверждено на заседании ПЦК

Информационных технологий

Протокол № 9 от 23.05.2018

председатель  Н.А. Серебрякова

Пермь 2018

Составитель:

Серебрякова Н.А., преподаватель высшей квалификационной категории, преподаватель ПГНИУ

Методические указания по организации и выполнению практических работ для студентов колледжа по дисциплине 1 С Программирование для специальностей 09.02.03 Программирование в компьютерных системах и 09.02.04 Информационные системы (по отраслям)/ сост. Н.А.Серебрякова; Колледж проф. образ. ПГНИУ. – Пермь, 2018. – 101 с.

Методические указания «1С Программирование» разработаны на основе требований Федерального государственного образовательного стандарта среднего профессионального образования по специальностям 09.02.03 Программирование в компьютерных системах и 09.02.04 Информационные системы (по отраслям) для оказания помощи студентам специальностей 09.02.03 Программирование в компьютерных системах и 09.02.04 Информационные системы (по отраслям) по дисциплине «1С программирование». Содержат типичные практические задания по всем разделам дисциплины.

Предназначены для студентов Колледжа профессионального образования ПГНИУ специальностей 09.02.03 Программирование в компьютерных системах и 09.02.04 Информационные системы (по отраслям) (СПО) всех форм обучения.

Печатается по решению педагогического совета Колледжа профессионального образования Пермского государственного национального исследовательского университета

СОДЕРЖАНИЕ

Пояснительная записка.....	4
Тематика лабораторных занятий.....	6
Содержание лабораторных работ.....	8
Список рекомендуемой литературы и источников.....	101

Пояснительная записка

Данные методические указания предназначены для изучения студентами практических основ программирования на платформе 1С: Предприятие.

Целью курса является формирования у студентов практических навыков для создания прикладных решений в системе 1С.

Каждая лабораторная работа содержит: цель, методические указания к выполнению и задание. Задания носят творческий характер и необходимы для развития интеллекта студента, его умения оригинально мыслить и реализовывать теоретические знания на практике.

При подготовке методических указаний был использован материал по программированию в среде 1С. Кроме методического материала представлена рекомендуемая литература по всему курсу.

Лабораторные занятия предназначены для реализации государственных требований к минимуму содержания и уровню подготовки выпускников по специальности 230115.51 Программирование в компьютерных системах.

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся в ходе освоения профессионального модуля должен:

уметь:

- работать с конфигуратором, глобальным модулем, общими таблицами;
- создавать справочник;
- создавать документ;
- создавать новый вид перечисления и использование его в документах;
- работать с регистрами;
- создавать запросы;
- работать с внешними приложениями.
- работать с объектами компоненты “Бухгалтерский учет”.
- Добавлять, удалять, редактировать объекты, изменять их порядок, сортировку, описание.
- Управлять положением объектов в форме;
- Запускать тестовое окна, окно редактирования слоев формы.
- Настраивать последовательность обхода элементов диалога;
- Формировать печатные формы;
- Редактировать список пользователей, устанавливать пароли, задавать интерфейс и набор прав для каждого пользователя;
- Создавать резервные копии и восстанавливать конфигурации и данные из резервной копии;

- Настроить фильтр журнала, просматривать журнал регистрации в Мониторе пользователей;
- Тестировать и исправлять информационную базу;
- Загружать измененную конфигурацию в рабочую базу, объединять конфигурации
- Разрабатывать шаблон справочника, устанавливать связь между справочниками;
- Программировать справочники;
- Создавать, сохранять и проводить документ, осуществлять поиск, удаление и перебор документов;
- Программно управлять точкой актуальности, последовательностью документов;
- Создавать, сохранять и проводить отчет;
- Использовать регистры;

- Организовывать запросы;
- Осуществлять операции в таблице значений;
- Осуществлять операции с объектом список значений;
- Создавать экранные формы.

Тематика лабораторных занятий

№ п/п	Тема занятия	Вид задания	Объем часов	Форма контроля
1	Тема 1.1. Знакомство с системой 1С:Предприятие 8.2	Лабораторная работа № 1, 2 Создание новой информационной базы. Внешний вид интерфейса прикладного решения. Запуск отладки в режиме 1С:Предприятие. Добавление подсистемы. Создание справочников. Проверка заполнения стандартных реквизитов. Создание справочников: простой справочник, справочник с predetermined элементами, иерархический справочник	6	Отчет. Проверка работы. Устный опрос
2				
3	Тема 1.2. Документы и регистры накопления	Лабораторная работа № 3, 4, 5 Справочники и документы. Автоматический пересчет суммы в строках документа. Создание документов и их заполнение.	6	Отчет. Проверка работы. Устный опрос
4	Тема 2.1. Регистры сведений	Лабораторная работа № 6, 7, 8. Добавление периодического регистра сведений. Автоматическая подстановка цены в документ при выборе номенклатур. Проведение документа «Оказание услуги» по двум регистрам.	6	Отчет. Проверка работы. Устный опрос
5	Тема 2.2. Механизм запросов	Лабораторная работа №9, 10, 11 Создание простых запросов при помощи конструктора.	6	Отчет. Проверка работы. Устный опрос
6	Тема 3.1. Отчеты и система компоновки данных	Лабораторная работа № 12, 13, 14 Отчет о начислениях сотрудникам. Перерасчет. Диаграмма Ганта.	8	Отчет. Проверка работы. Устный опрос
7	Тема 3.2. Программирова-	Лабораторная работа № 15, 16, 17 Форма выбора для	8	Отчет. Проверка ра-

	ние форм в при- кладных задачах	поля, содержащего ссылоч- ный реквизит. Проверка за- полнения реквизитов. Авто- матическая проверка запол- нения. Программная провер- ка заполнения.		боты. Устный опрос
--	--	---	--	-----------------------

Содержание лабораторных работ Лабораторная №1

Тема работы: Создание новой информационной базы. Внешний вид интерфейса прикладного решения. Запуск отладки в режиме 1С:Предприятие. Добавление подсистемы.

Цель лабораторной работы: ознакомиться с возможностью создания конфигурации в системе 1С:Предприятие и разработка конфигурации.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1С:Предприятие является универсальной системой автоматизации экономической и организационной деятельности предприятия. Поскольку такая деятельность может быть довольно разнообразной, система 1С:Предприятие может «приспосабливаться» к особенностям конкретной области деятельности, в которой она применяется. Для обозначения такой способности используется термин конфигурируемость, то есть возможность настройки системы на особенности конкретного предприятия и класса решаемых задач.

Логически всю систему можно разделить на две большие части, которые тесно взаимодействуют друг с другом: конфигурацию и платформу, которая управляет работой конфигурации.

Существует одна платформа (1С:Предприятие 8) и множество конфигураций. Для функционирования какого-либо прикладного решения всегда необходима платформа и какая-либо конфигурация



Рис. Конфигураций много, а платформа одна

Сама по себе платформа не может выполнить никаких задач автоматизации, так как она создана для обеспечения работы какой-либо конфигурации. То же самое с конфигурацией: чтобы выполнить те задачи, для которых она создана, необходимо наличие платформы, управляющей ее работой.

Итак, поскольку задачи автоматизации могут быть самыми разными, фирма «1С» и ее партнеры выпускают прикладные решения, каждое из которых предназначено для автоматизации одной определенной области человеческой деятельности. В качестве примера существующих прикладных решений можно перечислить следующие типовые решения:

- 1С:Управление небольшой фирмой,
- 1С бухгалтерия,
- 1С:Предприятие. Управление торговлей,
- 1С: Зарплата и Управление Персоналом,
- 1С:Предприятие. Управление производственным предприятием,
- 1С:Консолидация.

Режимы работы системы:

1С: Предприятие имеет различные режимы работы: 1С: Предприятие и Конфигуратор.

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

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

ПРАКТИЧЕСКАЯ ЧАСТЬ

Создание новой информационной базы

Запустите 1С Предприятие. В открывшемся диалоге вы увидите список информационных баз, с которыми вы работаете. Если этот список пуст, система сама предложит вам создать новую базу. Если же в списке информационных баз содержится какая-либо база, например, у вас установлена демонстрационная конфигурация, то для создания новой базы нажмите кнопку *Добавить* (рис. 1.2).

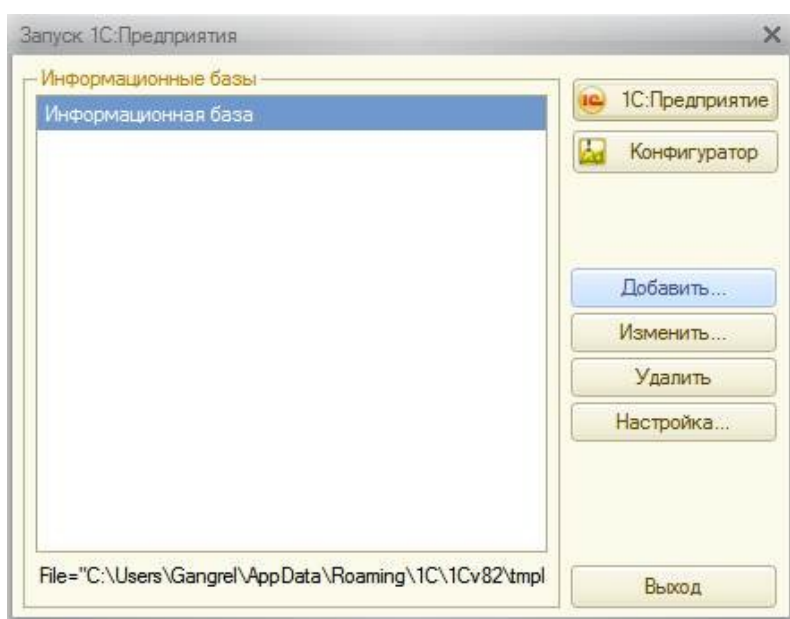


Рис. Создание новой информационной базы

- В открывшемся диалоге выберите пункт **«Создание новой информационной базы»**;
- Нажмите кнопку **«Далее»**. На следующем шаге выберите пункт **«Создание информационной базы без конфигурации»**;
- Нажмите кнопку **«Далее»**. На следующем шаге задайте наименование вашей информационной базы и выберите тип ее расположения **«На данном компьютере»**;

- Нажмите кнопку «Далее». На следующем шаге укажите каталог для расположения вашей информационной базы. Язык по умолчанию установлен в значение **Русский**;
- Нажмите кнопку «Далее». На следующем шаге нажмите кнопку «Готово»;
- В диалоге запуска 1С:Предприятия, в списке информационных баз, вы увидите созданную вами новую пустую базу.

Знакомство с конфигуратором:

Итак, запустим 1С:Предприятие в режиме Конфигуратор. Для этого нажмем кнопку «**Конфигуратор**» в диалоге запуска системы.

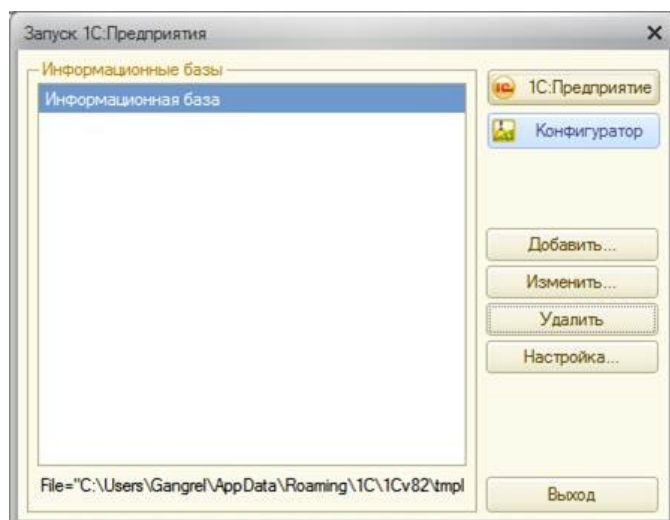


Рис. Конфигуратор

Перед вами окно конфигуратора:

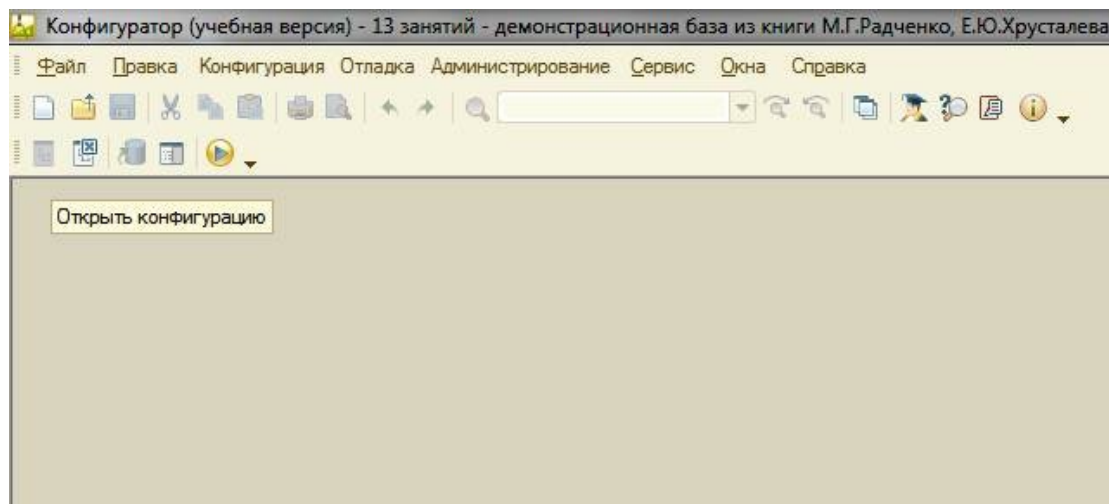


Рис. Окно конфигуратора

Именно с помощью этого инструмента мы будем создавать нашу конфигурацию. Сразу под заголовком окна находится главное меню configurатора, содержащее пункты **Файл, Правка, Конфигурация, Администрирование** и т.п. В каждом из этих пунктов содержится много подпунктов, вызов которых обеспечивает выполнение различных действий configurатора.

Ниже находится панель инструментов configurатора, в которую в виде кнопок-пиктограмм помещены наиболее часто используемые действия, вызываемые из меню.

Таким образом, одни и те же действия можно выполнить двумя разными способами: вызвав определенный пункт меню или нажав соответствующую ему кнопку на панели инструментов.

Выполним первую команду, с которой начинается работа с любой конфигурацией, - откроем конфигурацию с помощью пункта меню **Конфигурация > Открыть конфигурацию** или соответствующей кнопки на панели инструментов.

На экране *откроется дерево объектов конфигурации:*

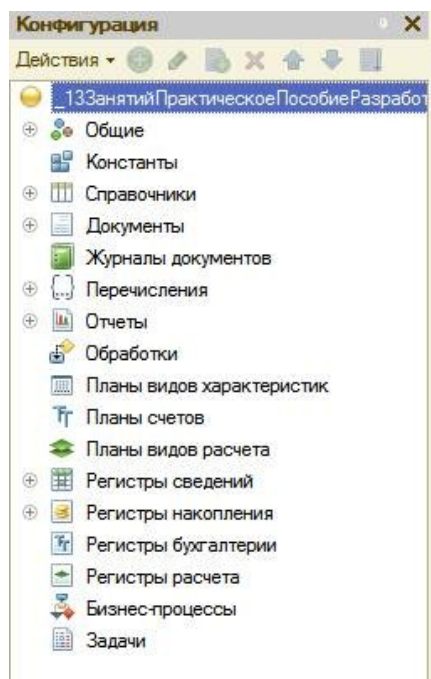


Рис. Дерево объектов конфигурации

Можно сказать, что дерево объектов конфигурации - основной инструмент, с которым работает разработчик. Дерево объектов конфигурации содержит в себе практически всю информацию о том, из чего состоит конфигурация.

Что такое объекты конфигурации:

Конфигурация представляет собой описание. Она описывает структуру данных, которые пользователь будет использовать в режиме работы 1С:Предприятие.

Кроме этого, конфигурация описывает всевозможные алгоритмы обработки этих данных, содержит информацию о том, как эти данные должны будут выглядеть на экране и на принтере, и т.д.

С одной стороны, объекты конфигурации представляют собой детали «конструктора», из которого собирается конфигурация. Обычно в конструкторе существует некоторый набор деталей. Детали могут быть разного вида. Теперь представьте, что деталей каждого вида мы можем создавать столько, сколько нам нужно. Мы можем соединять детали между собой различными способами.

То же и с объектами конфигурации. Мы можем создавать только объекты определенных видов. Но каждого вида объектов мы можем создать столько, сколько нам нужно. Объекты одного вида отличаются от объектов другого вида тем, что имеют разные свойства. Объекты могут взаимодействовать друг с другом, и мы можем описать такое взаимодействие. Объекты конфигурации также обладают различным поведением, и оно зависит от вида объекта. Одни объекты могут выполнять какие-то действия, другие этих действий выполнять не могут, зато у них есть свой собственный набор действий.

«Сложные» объекты конфигурации состоят из более «простых», и одни и те же «простые» объекты могут входить в состав сложных объектов. Такая структура позволяет упростить работу с объектами конфигурации, поскольку если мы знаем, как работать с каким-либо «простым» объектом, то в любом

«сложном» объекте, в состав которого он входит, мы будем работать с ним все тем же образом.

Самое важное качество объектов конфигурации - это их прикладная направленность. Объекты конфигурации не просто некие абстрактные конструкции, при помощи которых разработчик пытается описать поставленную перед ним задачу. Они представляют собой аналоги реальных объектов, которыми оперирует предприятие в ходе своей работы.

Как добавить объект конфигурации:

Перед началом работы следует объяснить некоторые приемы работы с конфигуратором:

Для того чтобы открыть и закрыть конфигурацию, следует использовать пункты меню **Конфигурация > Открыть конфигурацию** и **Конфигурация > Закрыть конфигурацию** или соответствующие им кнопки на панели инструментов.

После того как конфигурация открыта, ее состав появляется в окне дерева конфигурации (см. рис. 1.5). Это окно вы можете закрыть, как любое другое окно Windows, при этом конфигурация останется открытой. Чтобы снова отобразить на экране окно дерева конфигурации, следует воспользоваться командой меню **Конфигурация > Окно конфигурации**.

Добавить новый объект конфигурации можно несколькими способами, и вы можете использовать наиболее понятный и удобный для вас.

Первый способ: Необходимо установить курсор на ту ветку объектов конфигурации, которая вас интересует, и в командной панели окна конфигурации нажать кнопку **Действия > Добавить**.

Второй способ. Вы можете воспользоваться контекстным меню, которое вызывается при нажатии на правую клавишу мыши. Установите курсор на интересующую вас ветку объектов конфигурации и нажмите правую клавишу мыши. В появившемся меню выберите пункт **Добавить**.

Третий способ. Установите курсор на интересующую вас ветку объектов конфигурации и в командной панели окна конфигурации нажмите кнопку «Добавить» (с пиктограммой «+»).

Палитра

свойств:

Палитра свойств - это специальное служебное окно, которое позволяет редактировать все свойства объекта конфигурации и другую связанную с ним информацию. Поскольку разные объекты конфигурации имеют самые разные свойства, содержимое этого окна будет меняться в зависимости от того, какой объект является текущим (на каком объекте конфигурации установлен курсор).

- Выделим в дереве объектов конфигурации корневой элемент «**Конфигурация**» и двойным щелчком мыши откроем его палитру свойств.

- Зададим имя конфигурации «**Фамилия**». Соответствующий ему синоним устанавливается автоматически, но его можно изменить по своему усмотрению. В дальнейшем именно его мы будем видеть в рабочем окне 1С предприятия.

В этом случае, как и сейчас, палитра свойств откроется и будет закреплена на рабочей области конфигуратора. То есть при выделении какого-либо объекта конфигурации окно его свойств всегда будет открыто. Однако есть удобная возможность «открепить» палитру свойств, используя символ кнопки в заголовке окна палитры свойств.

Запуск отладки в режиме 1С:Предприятие:

Проверим первые изменения в режиме 1С:Предприятие. Для этого выполним пункт меню **Отладка > Начать отладку** или нажмем соответствующую кнопку > на панели инструментов конфигуратора. Система сама анализирует наличие изменений в конфигурации и выдает соответствующий вопрос об обновлении конфигурации базы данных. На вопрос конфигуратора ответим «**ДА**», и на экране появится окно 1С:Предприятия.

В режиме 1С:Предприятие: Внешний вид интерфейса прикладного решения

В заголовке окна видим название нашей конфигурации. Пустое пространство - это рабочая область приложения, которая пока ничем не заполнена.

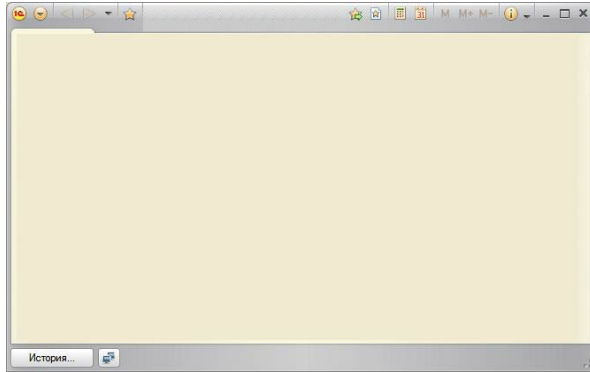


Рис. Внешний вид интерфейса прикладного решения

Кроме заголовка конфигурации в окне 1С:Предприятия ничего не появилось. И этого следовало ожидать.

Контрольные вопросы:

1. Что такое конфигурируемость системы 1С:Предприятие.
2. Из каких основных частей состоит система.
3. Что такое платформа и что такое конфигурация.
4. Для чего используются разные режимы запуска системы 1С:Предприятие.
5. Что такое дерево объектов конфигурации.
6. Что такое объекты конфигурации.
7. Что создает система на основе объектов конфигурации.
8. Какими способами можно добавить новый объект конфигурации.
9. Зачем нужна палитра свойств.
10. Как запустить 1С:Предприятие в режиме отладки.

Лабораторная работа 2

Тема: Создание справочников. Проверка заполнения стандартных реквизитов
Создание справочников: простой справочник, справочник с predetermined elements, иерархический справочник

Цель работы: Научиться создавать справочники.

Ход выполнения работы

Объект конфигурации Справочник предназначен для работы со списками данных. Как правило, в работе любой фирмы используются списки сотрудников, списки товаров, списки клиентов поставщиков и т.д. Свойства и структура этих списков описываются в объектах конфигурации Справочник, на основе которых платформа создает в базе данных таблицы для хранения информации из этих справочников. Справочник состоит из *элементов*. Пользователь в процессе работы может самостоятельно добавлять новые элементы в Справочник: например, добавить новых сотрудников, создать новый товар или внести нового клиента. В базе данных каждый элемент справочника представляет собой отдельную запись в основной таблице, хранящей информации из этого справочника. Каждый элемент справочника содержит некоторую дополнительную информацию, которая подробнее описывает этот элемент. Набор такой информации является одинаковым для всех элементов справочника, и для описания такого набора используются *реквизиты* объекта конфигурации Справочник, которые также, в свою очередь, являются объектами конфигурации. Поскольку эти объекты конфигурации логически связаны с объектом Справочник, они называются *подчиненными* этому объекту. Большинство реквизитов разработчик создает самостоятельно, однако у каждого объекта конфигурации Справочник по умолчанию существует набор стандартных реквизитов: **Код** и **Наименование**

и пр. Причем доступность стандартных реквизитов зависит от свойств справочника. Если справочник иерархический, у него будет доступен стандартный реквизит **Родитель**. Если справочник под-

чинен другому объекту конфигурации, у него будет доступен реквизит **Владелец**. Если установить длину стандартного реквизита **Код** равной нулю, то у справочника будет недоступен этот реквизит. То же самое относ к реквизиту **Наименование**. Однако как минимум либо **Код**, либо **Наименование** должны присутствовать в реквизитах справочника, иначе такой справочник не имеет смысла. Таким образом в базе данных справочник хранится в виде таблицы, в строках которой расположены элементы списка, а каждому реквизиту в этой таблице соответствует отдельный столбец. Соответственно, в ячейках этой таблицы хранится значение конкретного реквизита для конкретного элемента справочника.

Кроме этого, каждый элемент справочника может содержать некоторый набор информации, которая одинакова по своей структуре, различна по количеству и предназначена для разных элементов справочника. Для описания подобной информации могут быть использованы *табличные части* объекта конфигурации Справочник, являющиеся подчиненными ему объектами конфигурации. В этом случае в базе данных будут созданы дополнительные таблицы для хранения табличных частей, подчиненных конкретному элементу справочника.

Причем система скрывает от разработчика всю «техническую» часть, связанную с хранением данных: в базе данных для справочника создаются несколько таблиц, эти таблицы связываются по уникальному полю (Ссылка), поля таблиц имеют определенные типы и т.д. Все это система делает сама. Нам лишь нужно добавить в объект конфигурации подчиненный ему объект Табличная

часть.

Для удобства использования элементы справочника могут быть сгруппированы пользователем по какому-либо принципу. Возможность со-

здания таких групп в справочнике задается свойством **Иерархический** объекта конфигурации Справочник.

В этом случае элемент справочника, представляющий собой группу, будет являться *родителем* для всех элементов и групп, входящих в эту группу. Такой вид иерархии называется *иерархией групп и элементов*.

Возможен и другой вид иерархии - *иерархия элементов*. В этом случае в качестве родителя выступает не группа элементов справочника, а непосредственно один из его элементов. Например, такой вид иерархии можно использовать при создании справочника Подразделения, когда, одно подразделение является **родителем** для нескольких других, входящих в его состав.

Элементы одного справочника могут быть *подчинены* элементам или группам другого справочника. Например, справочник **ЕдиницыИзмерения** может быть подчинен справочнику **Товары**. Тогда для каждого элемента справочника **Товары** мы сможем указать единицы измерения, в которых этот товар поступает на склад. В системе 1С Предприятие это достигается путем указания списка *владельцев* справочника для каждого объекта конфигурации Справочник. В данном случае справочник **Товары** будет владельцем справочника **ЕдиницыИзмерения**.

Порой возникают ситуации, когда необходимо, чтобы в справочнике некоторые элементы существовали всегда, независимо от действий пользователя. Допустим, логика бизнес-процессов на предприятии такова, что все товары сначала поступают на основной склад, а затем по мере надобности перемещаются на другие склады. В этом случае в справочнике Склады всегда должен существовать склад Основной иначе приходование товаров будет выполнено неправильно. Объект конфигурации Справочник позволяет описать любое количество таких элементов справочника. Они называются *предопределенными* элементами справочника.

Предопределенные элементы отличаются от обычных тем, что они создаются в конфигураторе и что пользователь не может их удалить. Все

остальные действия с ними он делать может, в том числе и переименовывать. В интерфейсе predetermined элементы справочника помечены специальной пиктограммой.

В зависимости от того, какие действия мы хотим выполнять со справочником, нам требуется изображать справочник в «разном виде». Например, для того, чтобы выбрать некоторый элемент справочника удобнее представить справочник в виде списка; а для того, чтобы изменить какой-то элемент справочника, удобнее представить в реквизиты этого элемента справочника в одной форме. Система может самостоятельно сгенерировать все формы, которые нужны для представления данных, содержащихся в справочнике. Причем система «знает», какие именно формы нужно использовать в каких ситуациях.

Вообще говоря, для отображения справочника в различных ситуациях требуется максимум пять форм для справочника. Обратите внимание, что в различных местах конфигуратора одни и те же формы называются немного по-разному. Дело в том что в контекстном меню и палитре свойств отображаются свойства объектов конфигурации. Они одинаковые для всех объектов конфигурации. А в конструкторе форм и на закладке формы отображаются представления этих свойств в виде, более понятном, форма элемента используется для редактирования или создания элемента справочника. Форма группы используется для редактирования или создания группы справочника. Группа, как правило, содержит гораздо меньше информации, чем сам элемент справочника. Поэтому для нее нужна отдельная форма, отличная от формы элемента.

Форма списка используется для отображения списка элементов справочника. Форма выбора используется для того, чтобы в поле некоторой формы выбрать один из элементов справочника. При этом форма выбора проще, чем форма списка, так как в форме списка может показываться много реквизитов. А при выборе элемента (в документе, например), нам нужно знать только наименование. Поэтому можно для выбора использовать от-

дельную, более простую форму. Форма выбора группы используется, когда в поле некоторой формы нужно выбрать не просто элемент справочника, а одну из его групп, при этом форма выбора группы проще, чем форма выбора элемента, так как группа, как правило, содержит гораздо меньше информации, чем сам элемент справочника. При этом для всех ссылочных объектов конфигурации (справочников, документов и т.д.) будет использоваться форма объекта, но нужно понимать, что под объектом здесь понимается объект информационной базы, то есть «элемент» того, что хранит этот объект конфигурации. Для справочника это будет элемент справочника, для документа - документ, для плана счетов - счет и т. д.

Любая форма может быть описана в конфигураторе. Для создания такого описания существует подчиненный объект конфигурации *Форма*.

Объект конфигурации Форма подчинён одному из прикладных объектов, но может существовать и самостоятельно. На основании описания, содержащегося в объекте конфигурации Форма, в нужный момент работы платформа 1С:Предприятия создаст программный объект Форма, с которым и будет работать пользователь. Форма служит для «визуализации» данных, находящихся в базе данных. Она представляет эти данные в удобном для пользователя виде и позволяет описать алгоритмы, которые будут сопровождать работу пользователя с данными, показанными в форме.

Так как наше ООО «На все руки мастер» оказывает услуги по ремонту бытовой техники, очевидно,

что для ведения учета нам потребуется хранить некоторую списочную информацию.

Для начала нам понадобится список сотрудников предприятия которые будут оказывать услуги. Затем нам будет нужен список клиентов, с которыми работает ООО «На все руки мастер». После этого нам понадобится перечень услуг, которые может оказывать наше предприятие, и список материалов, которые могут быть израсходованы. Кроме этого, нам потребуется список

складов, на которых может находиться материалы ООО «На все руки мастер».

1. Создать справочник, в котором будут храниться наименования наших клиентов для этого в конфигураторе выделите в дереве объектов конфигурации ветвь **Справочники** и нажмите кнопку **Добавить** в командной панели окна конфигурации. В открывшемся окне редактирования объекта конфигурации задать имя справочника **Клиенты**. На основании имени платформа автоматически создаст синоним – **Клиенты**.

2. Представление **объекта** определяет название объекта в единственном числе и используется в названии стандартной команды, например, команды создания объекта - **Клиент: создать**. Представление объекта нужно задавать тогда, когда синоним объекта конфигурации задан во множественном числе, или когда он описывает множество объектов. Потому что в интерфейсе автоматически формируются команды открытия списка справочника и команды создания нового элемента справочника. Если синоним задан во множественном числе - то для команды открытия списка это вполне подходит **Клиенты**, то есть посмотреть всех Клиентов. Для этой команды нужно задать представление в единственном числе - Клиент. Представление объекта как раз и используется для того, чтобы описать, как будет выглядеть в интерфейсе команда представления нового клиента. Также оно будет использовано в заголовке формы клиента и в представлении ссылки на клиента. Расширенное **представление объекта** определяет заголовок формы объекта. Представление **списка** определяет название списка объектов и используется в названии стандартной команды, например, команды открытия списка объектов - **Клиенты: открыть**. Представление списка нужно задавать тогда, когда синоним задан в единственном числе. Например, это часто бывает у документов (Приходная накладная). Тогда в представлении списка нужно указывать название объекта конфигурации во множественном числе (**Приходные накладные**).

3. Задать два свойства Представление объекта -Клиент и Представление списка -Клиенты.

4. Нажмите кнопку **Далее** и перейдите на закладку **Подсистемы** окна редактирования объекта конфигурации Справочник. На этой закладке определяется, в каких подсистемах будет отображаться данный справочник. Список клиентов должен быть доступен в разделе **Оказание услуг**, так как оказываемые услуги относятся к определенному клиенту. Бухгалтерская отчетность, формируемая в разделе **Бухгалтерия**, также может быть представлена в разрезе клиентов. Обратите внимание, что на закладке Состав также можно изменить список объектов, входящих в подсистему.

5. Вернуться к окну редактирования объекта конфигурации Справочник и нажать на закладку Данные. Здесь представляют интерес длина кода и длина наименования. Длина кода - важное свойство справочника. Код справочника используется для идентификации элементов справочника и содержит уникальные для каждого элемента справочника, назначения. Платформа может сама контролировать уникальность код и поддерживать автоматическую нумерацию элементов справочника. Поэтому от длины кода будет зависеть количество элементов, содержащихся в справочнике. Длина кода - 9 символов. В результате мы сможем использовать коды от 1 до 999999999 - этого вполне достаточно для нашего небольшого ООО «На все руки мастер»

6. Перейдите к длине наименования, 25 символов для нас явно мало, увеличить длину наименования до 50. Прежде чем запускать 1С: Предприятие, настроить интерфейс приложения, чтобы было удобнее вводить новые элементы справочника. Команда для открытия списка справочника, как и команда для создания его новых элементов, добавляется в интерфейс тех разделов (подсистем), в которых будет отображаться справочник. Но команда создания новых элементов по умолчанию невидима в интерфейсе приложения. Это объясняется тем, что возможность просматривать списки справочника нужна, как правило, всегда. А возможность создания

новых элементов справочника используется не так часто. Поэтому соответствующую команду следует включать только для тех справочников (объектов конфигурации), создание новых элементов в которых является «основной» деятельностью для пользователей в данном разделе прикладного решения.

7.

команду для создания новых клиентов. Для этого в дереве объектов конфигурации выделите ветвь Подсистемы, вызовем ее контекстное меню и выберем пункт Все подсистемы. В открывшемся окне **Все подсистемы** слева в списке **Подсистемы** выделим подсистему **ОказаниеУслуг**. Справа в списке **Командный интерфейс** отразятся все команды выбранной подсистемы. При создании справочника в группу **Панель навигации Обычное** добавилась команда **Клиенты** для открытия этого списка. В группу **Панель действий Создать** добавилась команда **Клиент: создать** для создания нового элемента справочника, но она невидима по умолчанию. Включим видимость у этой команды. Команду создания нового клиента мы отражаем в подсистеме **Оказание услуг**, где «работает» менеджер, а для бухгалтерии она невидима, так как не предполагается, что бухгалтеры будут вводить новых клиентов. Однако это не лишает бухгалтера такой возможности - он может создать нового клиента, используя список клиентов открыть список **элемента** без использования списка элементов - это вопрос удобства работы, а не ограничения прав пользователя, и мы предоставляем эту удобную возможность менеджеру, а не бухгалтеру.

8. Закройте окно редактирования справочника **Клиенты** и запустите **1С: Предприятие** в режиме отладки. Ответить утвердительно на запрос конфигурирования об обновлении конфигурации и увидим окно, содержащее список изменений в структуре конфигурации, автоматически сгенерированный платформой. В данном случае мы добавили справочник **Клиенты**. Нажмите кнопку **Принять**. Перед нами откроется окно системы в режиме **1С: Предприятие**. Мы видим, что если перейти в раздел **Оказание услуг** или **Бухгалтерия**, то слева в вертикальной области окна появится *панель навига-*

ции. Панель навигации отображает структуру выбранного раздела. Как правило, панель навигации предназначена для быстрого перехода к различным спискам в пределах выбранного раздела программы. Сейчас она содержит команду для открытия нашего первого списка - **Клиенты**. Обратите внимание, что название команды **Клиенты** определяется свойством **Представление списка**, которое мы задали для этого справочника. Если это свойство не задано, то для представления списка используется значение синонима объекта конфигурации Справочник. Также в разделе **Оказание услуг** появилась панель действий. *Панель действий* содержит команды, которые соответствуют текущему разделу, выбранному в панели разделов. Эти команды объединены в стандартные группы: **Создать**, **Отчеты**, **Сервис** и группы, созданные разработчиком. Группа **Создать** включает в себя команды создания новых объектов информационной базы, например, документов или элементов справочников. Сейчас в панели действий раздела **Оказание услуг** в группе **Создать** доступна команда для создания элементов нашего первого справочника **Клиенты**, которую мы сделали видимой в интерфейсе этого раздела. Этой командой мы и воспользуемся для создания новых элементов справочника, не открывая при этом списка клиентов. Обратите внимание, что название стандартной команды создания нового элемента определяется свойством **Представление объекта**; которое мы задали для этого справочника. Если бы мы это свойство не задали, то в названии команды использовался бы синоним объекта конфигурации Справочник **Клиенты - Клиенты**. Это неудобно, так как ничем не отличается от команды открытия списка, и не совсем верно - ведь при создании элемента справочника мы создаем только одного нового клиента. У раздела **Бухгалтерия** нет панели действий, так как для этой подсистемы мы не устанавливали видимость команды создания новых элементов из группы команд **Создать**. Кроме этого, нет ни одной другой видимой команды из групп **Отчеты** или **Сервис**. Соответственно, раз нет команд, входящих в группу, значит невидима Вся группа, а раз нет ни *одной* группы, значит невидима вся панель действий.

9. В справочник добавить несколько элементов. Для этого выполните команду Клиент в панели действий раздела Оказание услуг. Перед нами откроется форма для создания элемента справочника - основная форма объекта.

10. Внести наименование нового клиента Иванов Михаил Юрьевич. Код вносить не будем, так как он генерируется автоматически. Нажмите *записать и закрыть*. При этом в правом нижнем углу появится информационное сообщение о том, что элемент был создан либо изменен.

11. Нажав на ссылку в *информационной панели* (в нижней части окна приложения), можно открыть этот элемент. В этой панели автоматически отображается информация о последних действиях, выполненных в системе. Это позволяет не пользоваться списком для того, чтобы убедиться, что нужный элемент записан.

12. Добавить еще одного клиента с наименованием Роман. Последнего клиента с наименованием Спиридонова Галина добавить, пользуясь формой списка клиентов. Для этого выполните команду **Клиенты**, расположенную в панели навигации раздела **Оказание услуг**. Справа от панели навигации в рабочей области окна приложения откроется основная форма списка. Добавить новый элемент в справочник можно при помощи кнопки **Создать** в командной панели формы или клавишей **insert**. Нажмете кнопку **Создать**. Обратите внимание, что поле **Наименование** при вводе нового клиента подсвечено красным пунктиром. Это значит, что для этого поля по умолчанию выполняется проверка заполнения. Если это поле оставить пустым и попытаться записать клиента, то будет получено сообщение об ошибке.

13. Закрыть 1С: Предприятие и вернуться в режим конфигуратор. Чтобы посмотреть состав и свойства стандартных реквизитов справочника, в окне редактирования объекта конфигурации Справочник Клиенты на закладке Данные нажать кнопку **Стандартные реквизиты**. Выделить в списке реквизит **Наименование** и вызвать его контекстное меню и выбрать пункт

Свойства. В палитре свойств реквизита Наименование установлено свойство Проверка заполнения по умолчанию в значение выдавать ошибку.

14. Перейти к созданию второго Справочника; который будет использоваться в нашей конфигурации. Добавить новый объект конфигурации Справочник. Назовите его Сотрудники. На основании имени платформа автоматически заполнит его синоним.

15. Задать Представление объекта как Сотрудник. Представление списка устанавливать не будем, а Расширенное представление списка задать как Список сотрудников. Нажмите кнопку Далее и перейдите на закладку Подсистемы. Список сотрудников должен быть доступен в разделах **Оказание услуг** и **Расчет зарплаты**.

16. Создать справочник, имеющий табличную часть. Поэтому *добавим* в справочник новую Табличную часть с именем ТрудоваяДеятельность. Для этого нажмите кнопку **Добавить табличную часть** над списком табличных частей справочника.

17. Задать имя табличной части ТрудоваяДеятельность:

18. Создать реквизиты табличной части ТрудоваяДеятельность.

Для этого нажмем кнопку **Добавить реквизит** над списком табличной частей справочника Добавим следующие реквизиты:

- Организация - тип Строка, длина 100;
- НачалоРаботы-типДата, состав даты Дата;
- ОкончаниеРаботы-тип Дата, состав даты – Дата; > .,!(,.
- Должность - тип Строка, длина 100.

19. Для реквизитов **НачалоРаботы** и **Окончание Работы** выбрать состав даты - **Дата**, поскольку в системе 1С:Предприятие 8 значения типа **Дата** содержат как дату, так и время.

20. Сделайте видимой в панели действий подсистемы **РасчетЗарплаты** стандартную команду для создания новых сотрудников. Для этого в дереве объектов конфигурации выделить ветвь Подсистемы, вызвать ее контекст-

ное меню и выберем пункт Все подсистемы. В открывшемся окне слева в списке **Подсистемы** выделить подсистему **РасчетЗарплаты**. Справа в списке **Командный интерфейс** отразятся все команды выбранной подсистемы.

21. В группе **Панель действий. Создать** включить видимость у команды **Сотрудник: создать**. Также мы видим, что в группу **Панель навигации.Обычное** добавилась команда **Сотрудники** для открытия этого списка. Она включена по умолчанию.

22. Для подсистемы **ОказаниеУслуг** никаких команд добавлять в панель действий не будем, так **как** вряд ли понадобится пополнять список сотрудников в этом разделе. На этом создание справочника **Сотрудники** завершено.Закреть окно редактирования справочника **Сотрудники** и запустить

1С: Предприятие в режиме отладки.

23. Ответим утвердительно на запрос конфигуратора об обновлении конфигурации и увидим окно, содержащее список изменений в структуре конфигурации, автоматически сгенерированный платформой. В данном случае мы добавили справочник **Сотрудники** . Нажмите кнопку **Принять**.

24. В открывшемся окне 1С Предприятия мы видим, что в панели навигации разделов **Оказание услуг** и **Расчет зарплаты** появилась команда **Сотрудники** для открытия списка сотрудников. Название этой команды определяется синонимом объекта, так как **Представление списка** мы для этого справочника не задавали. Также в панели действий раздела **Расчет зарплаты** появилась команда **Сотрудник** для создания новых сотрудников. Название этой команды определяется свойством **Представление объекта**, которое задали для этого справочника. Этой командой и воспользуемся для создания новых элементов справочника не открывая при этом списка сотрудников:

25. Выполнить команду **Сотрудник**. Откроется форма для создания элемента справочника –основная форма объекта. Заголовок этой формы

определяется свойством **Представление объекта**. Создадим следующих сотрудников:

1. Гусаков Николай Дмитриевич. Тру-

довая деятельность:

Организация - ЗАО «НТЦ»,

Начало работы -

01.02.2000,

Окончание работы

- 16.04.2003, Должность

- Ведущий специалист.

2. Деловой Иван Сергеевич.

Трудовая деятельность:

а) Организация ООО

«Автоматизация»,

Начало работы -22.01.1996,

Окончание работы -

31.12.2002,

Должность - Инженер;

2) Организация - ЗАО «НПО СпецСвязь»

Начало работы - 20.06.1986,

Окончание работы - 21.01.1995,

Должность - Начальник производства;

3. Симонов Валерий Михайлович.

Трудовая деятельность:

Организация - ООО «СтройМастер»,

Начало работы - 06.02.2001,

Окончание работы-03.04.2004,

Должность - Прораб.

Строки табличной части справочника можно добавлять кнопкой **Добавить** и располагать в произвольном порядке используя кнопки **Вверх**, **Вниз** в командной панели табличной части.

26. Чтобы просмотреть список добавленных сотрудников, выполните команду **Сотрудники**, расположенную в панели навигации раздела **Расчет зарплаты**. Справа от панели навигации в рабочей области окна приложения откроется основная форма списка. Обратите внимание, что заголовок этой формы определяется свойством **Расширенное представление списка**, которое задали для этого справочника.

27. Приступить к созданию следующего справочника **Номенклатура**. Справочник **Номенклатура** будет содержать информацию об услугах которые оказывает ООО «На все руки мастер», и о тех материалах которые при этом могут быть использованы. 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 ООО «На все руки мастер» оказывает самые разные услуги, они также будут логически собраны в несколько групп.

28. Создать новый объект конфигурации **Справочник** и назвать ее **Номенклатура**. На основании имени платформа автоматически заполнит его синоним. Поскольку понятие **Номенклатура** не имеет единственного числа больше никаких свойств, определяющих представление объекта интерфейса приложения, задавать не будем. Вместо **Представления объекта** и **Представления списка** будет использоваться **Синоним объекта Номенклатура**. Перейдите на закладку **Подсистемы**. Список номенклатуры должен бы доступен в разделах **Учет материалов**, **Оказание услуг** и **Бухгалтерия**. Отметить в списке **подсистем эти подсистемы**.

29. Создать иерархический справочник. Для этого перейти на закладку **Иерархия** и установить флажок **Иерархический справочник**.

30. На закладке **Данные** оставим по умолчанию длину и тип кода, длину наименования справочника задать равной **100** символам.

31. Прежде чем запускать 1 С: Предприятие, настроить интерфейс приложения, чтобы нам было удобнее вводить новые элементы справочника. Сделаем доступной в панели действий разделов **УчетМатериалов** и **ОказаниеУслуг** стандартную команду создания новых элементов списка номенклатуры. Для этого в дереве объектов конфигурации выделим ветвь **Подсистемы**, вызовем ее контекстное меню и выберем пункт **Все подсистемы**.

Лабораторная работа № 3

Тема: Справочники и документы. Автоматический пересчет суммы. Автоматический пересчет суммы в строках документа. Создание документов и их заполнение.

Цель работы: Научиться создавать справочники.

Ход выполнения работы

1. В открывшемся окне слева в списке **Подсистемы** выделим подсистему **Учет Материалов**. Справа в списке Командный интерфейс отразятся все команды выбранной подсистемы.

2. В группе **Панель действий.Создать** включить видимость у команды **Номенклатура: создать**. В группу **Панель навигации.Обычное** добавилась команда **Номенклатура** для открытия этого списка. Она включена по умолчанию. Выделить в списке подсистем **ОказаниеУслуг**, проделать те же действия. А для подсистемы **Бухгалтерия** никаких команд добавлять в панели действий не будем, так как вряд ли понадобится пополнять список номенклатуры в этом разделе. Теперь заполним справочник **Номенклатура**.

3. Закроем окно редактирования справочника **Номенклатура** и запустим 1С: Предприятие в режиме отладки. Ответим утвердительно на запрос конфигуратора об обновлении конфигурации и увидим окно, содержащее список изменений в структуре конфигурации, автоматически сгенерированный платформой. В данном случае мы добавили справочник **Номенклатура**.

4. Нажмите кнопку **Принять** в открывшемся окне 1С Предприятия мы видим, что в панели навигации разделов **Учет материалов, Оказание услуг** и **Бухгалтерия** появилась команда **Номенклатура**. Название этой команды определяется синонимом объекта, так как - других представлений для этого справочника не задавали.

5. Выполним команду **Номенклатура** в панели навигации раздела **Учет материалов**. Справа от панели навигации в рабочей области окна приложения откроется основная форма списка.

6. **Создание элементов в иерархическом справочнике.** Создадите две группы в корне справочника: **Материалы** и **Услуги**. Для этого нажмите кнопку **Создать группу** в командной панели формы списка. Задайте наименование групп: **Материалы**, **Услуги**. Поля **Родитель** и **Код** не заполнять.

7. Раскройте группу **Материалы** и создайте в ней пять элементов:

- Строчный трансформатор Samsung\$
- Строчный трансформатор Gold-Star,
- Транзистор Philips 2N2369;
- Шланг резиновый,
- Кабель электрический.

8. Для добавления элемента в открытую группу, справочника нажмите кнопку **Создать** в командной панели формы списка справочника. Откроется форма для создания элемента справочника - основная форма объекта. Причем если новый элемент добавляется из формы

списка в некоторую открытую группу, то система автоматически подставляет в качестве родителя эту группу. В данном случае родителем является группа **Материалы**.

9. Для создания нового **элемента** справочника можно также воспользоваться командой **Номенклатура**, которая появилась в панели действий разделов **Учет материалов** и **Оказание услуг**. В этом случае, если элемент добавляется командой из панели действий, она никак не связана со списком номенклатуры. Поэтому система «не знает», в какую группу добавлять элемент, и родителя нужно указывать вручную. При создании новых материалов или услуг список номенклатура открывать не обязательно. Факт добавления элемента в справочник отражается в сообщении в нижнем углу приложения. Раскроем группу **Услуги** и тоже создадим в ней несколько элементов - услуги по ремонту телевизоров.

10. Диагностика,

11. Ремонт отечественного телевизора,

12. Ремонт импортного телевизора.

И услуги по установке стиральных Машин:

13. Подключение воды,

14. Подключение электричества.

15. Перенос элементов в другие группы. Теперь разнести услуги по двум смысловым группам: услуги по ремонту телевизоров и услуги по установке стиральных машин. Для этого в группе **Услуги** создайте еще две группы: **Телевизоры** и **Стиральные машины**. Для того чтобы переместить услуги в соответствующие группы в окне списка установим курсор на ту услугу, которую мы хотим переместить, и выполним команду **Все действия** ► **Переместить в группу**. В открывшемся окне выберем новую группу. Можно выделить в списке сразу несколько элементов (левой кнопкой мыши, удерживая при этом клавишу **Ctrl**) и переместить их все сразу. Или же можно мышью перетащить выделенный элемент справочника в нужную группу. Мож-

но, также открыть для редактирования выделенный элемент справочника и изменить поле Родитель.

16. Переместить в группу Телевизоры услуги Диагностика, Ремонт отечественного телевизора и Ремонт импортного телевизора.

17. Услуги Подключение воды и Подключение электричества переместить в группу Стиральные машины.

18. В группе Материалы создать две группы; Радиодетали и Прочее.

19. В группу Прочее поместить Кабель электрический и Шланг резиновый. Остальные материалы переместить в группу Радиодетали. Переключившись в представление списка в виде дерева по команде Все Действия - Режим просмотра - Дерево можно увидеть список номенклатуры в виде дерева.

20. Справочник с предопределенными элементами. Создать справочник Склады, который будет содержать информацию о складах, используемых ООО «На все руки мастер». Этот справочник будет включать в себя один предопределенный элемент - склад Основной, на который будут поступать все материалы. Наша задача - создать справочник, содержащий предопределенные элементы.

21. Откройте конфигуратор и создайте новый объект Конфигурации Справочник с именем Склады. На основании имени платформа автоматически заполнит его синоним. Задать Представление объекта как Склад. Вместо свойства Представления списка будет использоваться Синоним объекта Склады.

22. Перейдите на закладку Подсистемы. По логике нашей конфигурации список складов должен быть доступен в разделах Оказание услуг и Учет материалов, так как поступление материалов и оказание услуг, как правило, учитываются в разрезе складов. Поэтому отметьте в списке подсистем эти подсистемы.

23. Свойство «Быстрый выбор». Заполните еще одно свойство справочника Склады - Быстрый выбор. Для этого перейдите на закладку

Формы и установите соответствующий флажок. Дело в том, что по умолчанию при нажатии кнопки выбора в поле содержащем ссылку на элемент справочника, открывается основная форма выбора элемента справочника. Она может быть не всегда удобна, особенно в том случае, когда справочник неиерархический заведомо содержит небольшое количество элементов. Свойство Быстрый выбор как раз позволяет выбирать элементы не из отдельной формы, а из небольшого выпадающего списка, заполненного элементами этого справочника. Этот вариант наиболее удобен для списка складов, так как их, вероятно, будет немного. Для остальных справочников свойство Быстрый **выбор** не устанавливали, так как Номенклатура - иерархический справочник, и, следовательно, быстрый выбор для него не имеет смысла. А список сотрудников и, особенно, список клиентов может быть очень большим, и выпадающий список в этом случае будет неудобно прокручивать.

24. Предопределенные элементы. Перейдите на Закладку Прочее и нажмите кнопку Предопределенные. Система откроет список предопределенных элементов справочника и он пуст, поэтому нажмите кнопку Добавить и создайте предопределенный элемент с именем Основной. Обратить внимание на то, что помимо наименования еще задали и имя предопределенного элемента справочника. Наименование предопределенного элемента пользователь может изменить а имя пользователь не видит и изменить не может:

25. Прежде чем запустить 1С:Прейприятие настроим интерфейс приложения, чтобы нам было удобнее вводить новые элементы справочника. В панели действий разделов УчетМатериалов сделаем доступной стандартную команду для создания новых складов. Для этого в дереве объектов конфигурации выделим ветвь Подсистемы, вызовем ее контекстное меню и выберем пункт подсистемы. В открывшемся окне слева в списке Подсистемы выделим подсистему УчетМатериалов. Справа в списке Командный интерфейс отразятся все команды выбранной подсистемы.

26. В группе Панель действий.Создать включить видимость у команды **Склад: создать**. Также мы видим, что в группу Панель навигацни.Обыч-

ное добавилась команда **Склады** для открытия этого списка. Она включена по умолчанию. Закроем окно редактирования справочника **Склады** и запустим 1С Предприятие в режиме отладки. Ответим утвердительно на запрос конфигулятора об обновлении конфигурации и увидим окно, содержащее список изменений в структуре конфигурации, автоматически сгенерированный платформой. В данном случае мы добавили справочник **Склады**.

27. В открывшемся окне 1С Предприятия мы видим, что в панели действий раздела **Учет материалов** появилась команда **Склад** для создания новых складов. Название этой команды определяется свойством Представление объекта, которое мы задали для этого справочника. Также в панели

навигации разделов **Оказание услуг** и **Учет материалов** появилась команда **Склады** для открытия списка складов. Название этой команды определяется синонимом объекта, так **Представление списка** мы для этого справочника не задавали.

28. Выполним команду **Склады** в панели навигации раздела **Учет материалов**. Справа от панели навигации в рабочей области окна приложения откроется основная форма списка. В списке складов уже есть один элемент с наименованием; **Основной**. Это предопределенный элемент, который мы создали в конфигураторе. Выполнив команду **Склад** в панели действий, добавим в справочник еще один склад, который назовем **Розничный**.

29. Обратите внимание, что система отмечает различными пиктограммами обычный и предопределенный элементы справочника. Несмотря на то, что можно изменить код или наименование у обоих элементов, пометка на удаление (или удаление) возможна только для обычных элементов справочника. При попытке пометить на удаление предопределенный элемент система выдаст предупреждение.

30. Таким образом, мы можем обозначить две характерные особенности предопределенных элементов:

на predeterminedенные элементы могут опираться алгоритмы работы конфигурации (т. к. возможно обращение к ним из встроенного языка **по** имени);

предeterminedенные элементы являются объектами базы данных, которые нельзя удалить в режиме 1С Предприятия.

Из этого видно, в чем заключается принципиальная с точки зрения конфигурации разница между обычными и predeterminedенными элементами справочника.

Обычные элементы «непостоянны» для конфигурации. В процессе Работы пользователя они могут появиться, исчезнуть. Поэтому конфигурация хоть и может отличить их друг от друга, но рассчитывать на них в выполнении каких-либо алгоритмов она не может в силу их «непостоянства». Предeterminedенные элементы, напротив «постоянны». В процессе Работы пользователя они находятся всегда на своих местах и исчезнуть не могут. Поэтому конфигурация может работать с ними вполне уверенно и опираться на них при отработке различных алгоритмов. По этой причине каждый из predeterminedенных элементов имеет уникальное имя для того, чтобы к нему можно было обратиться средствами встроенного языка.

Контрольные вопросы

1. Для чего предназначен объект конфигураций Справочник.
2. Каковы характерные особенности справочника.
3. Для чего используются реквизиты и табличные части справочника.
4. Зачем нужны иерархические справочники и что такое родитель.
5. Зачем нужны подчиненные справочники и что такое владелец.
6. Какие основные формы существуют у справочника.

7. Что такое predetermined elements.
8. Чем с точки зрения конфигурации отличаются обычные элементы справочника от predetermined elements.
9. Как пользователь может отличить обычные элементы справочника от predetermined elements.
10. Как создать объект конфигурации Справочник и описать его структуру.
11. Как добавить новые элементы в справочник.
12. Как создать группу справочника.
13. Как переместить элементы из одной группы справочника в другую.
14. Зачем нужна основная конфигурация и конфигурация: базы данных.
15. Как изменить конфигурацию базы данных.
16. Как связаны объекты конфигурации и объекты базы данных.
17. Что такое подчиненные объекты конфигурации.
18. Зачем нужна проверка заполнения у реквизитов справочника.
19. Что такое быстрый выбор и когда его использовать.
20. Как отобразить справочник и определить его представление в различных разделах интерфейса приложения.

Лабораторная работа № 3 б

Тема работы: Создание документов.

Цель работы: Научиться создавать различные документы.

Ход выполнения работы.

1. Открыть конфигуратор и добавить новый объект конфигурации **Документ**. На закладке **Основные** задать имя документа - **Приходная-**

Накладная. На основании имени платформа автоматически заполнит его синоним. Здесь же определим, как будет представлен документ в интерфейсе 1С:Предприятия. **Представление объекта** задавать не будем, вместо него будет использоваться **Синоним** объекта. Это нам вполне подходит, так как мы задали его имя в единственном числе.

2. **Представление списка**, задать во множественном числе как **Приходные накладные**. Нажмем **Далее** и перейдем на закладку **Подсистемы**.

3. По логике нашей конфигурации список приходных накладных должен быть доступен в разделах **Учет материалов** и **Бухгалтерии Действительно**, к первому разделу этот документ имеет прямое отношение, а для бухгалтерского анализа всегда может понадобиться список документов, отражающих поступление материалов. Поэтому отметьте в списке подсистем эти подсистемы.

4. Перейдите на закладку **Данные** и создайте реквизит документа с именем **Склад**. Для этого нажмем кнопку **Добавить** над списком реквизитов документа. Выбрать для реквизита ссылочный тип данных **СправочникСсылка.Склады**. Этот тип стал доступен в конфигурации после создания объекта конфигурации **Справочник Склады**.

5. В палитре свойств для созданного нами реквизита **Склад** документа найти свойство **Значение заполнения и в качестве значения** этого свойства выберите predetermined элементы справочника **Склады - Основной**. Таким образом, при создании нового документа склад будет сразу заполняться значением **Основной**, и пользователю не придется делать это вручную.

6. После этого добавить в документ табличную часть с именем **Материалы**. Для этого нажмите кнопку **Добавить табличную часть** над списком табличных частей документа.

7. Кроме имени табличной части установите свойство **Проверка заполнения** в значение **Выдавать ошибку**. Тем самым мы задаем условие,

что документ **Приходная накладная** обязательно должен содержать табличную часть, то есть список приходуемых материалов. Иначе будет выдано сообщение об ошибке, и документ не будет сохранен. Создадите реквизиты табличной части **Материалы**. Для этого нажмите кнопку **Добавить реквизит** в разделе описания табличных частей документа:

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.

8. Для каждого реквизита табличной части также установите свойство **Проверка заполнения** в значение **Выдавать ошибку**. Тем самым при записи документа будет проверяться на заполнение не только табличная часть в целом, но и ее отдельные реквизиты.

9. Перейдите на закладку **Нумерация** и убедитесь, что свойство **Автонумерация** включено. Это обеспечит автоматическую генерацию уникальных номеров для создаваемых нами документов.

10. Отредактируйте командный интерфейс, чтобы в подсистеме **Учет материалов** была доступна команда создания новых документов. Для этого в дереве объектов конфигурации выделим ветвь **Подсистемы** вызовем ее контекстное меню и выберем пункт **Все подсистемы**. В открывшемся окне слева в списке **Подсистемы** выделим подсистему **УчетМатериалов**. Справа в списке **Командный интерфейс** отразятся все команды выбранной подсистемы. В группе **Панель действий. Создать** включить видимость у команды **Приходная накладная: создать**.

11. Запустите 1С:Предприятие в режиме отладки и протестируйте получившийся документ. В открывшемся окне 1С:Предприятия увидите, что в панели навигации разделов **Бухгалтерия** и **Учет материалов** появилась команда **Приходные накладные** для открытия списка приходных накладных.

12. Название этой команды определяется свойством **Представление списка**, которое мы задали для этого документа. Также в панели действий раздела **Учет материалов** появилась команда **Приходная накладная** для создания новых документов этого вида. Название этой команды определяется синонимом, так как **Представление объекта** для этого документа не задавали.

13. Выполните команду **Приходная накладная** в панели действий раздела **Учет материалов** и создайте новую приходную накладную.

14. В форме документа система автоматически подставит текущую дату создания документа нулевое время, так как документ еще не проведен. В качестве времени документа при оперативном проведении ему присваивается оперативная отметка времени. Поле **Номер** не заполнено, но система сама сгенерирует для нового документа уникальный номер, так как свойство **Автономерация** документа включено по умолчанию. Новый номер будет сохранен в момент записи документа. Обратите внимание, что склад уже заполнен значением **Основной**, т.к. мы и задали в свойствах этого реквизита.

15. Осталось только заполнить табличную часть приходной накладной материалами для ремонта телевизоров. При нажатии кнопки выбора в поле **Материал** табличной части документа открывается форма для выбора документов справочника **Номенклатура**, так как этот реквизит имеет ссылочный тип данных и ссылается на справочник **Номенклатура**.

16. Нажать **Провести и закрыть**. Документ будет сохранен и проведен, ему будет присвоен автоматически сгенерированный системой номер и текущее время проведения документа.

17. Аналогичным образом создайте второй документ, который будет приходовать на **Основной** склад материалы для установки стиральных машин. Но теперь не будете использовать кнопку выбора в поле **Материал**, а просто начните вводить название материала в это поле. Платформа автоматически найдет материалы, найдет материалы название которых начинается

с введенных символов, и предложит их нам для выбора. Нажмите **Провести и закрыть**.

18. Обратите внимание, что при вводе нового документа табличная часть в целом и каждая ее колонка подсвечена красным пунктиром. Это значит, что для них выполняется проверка заполнения. Если не ввести ни одной строки в табличную часть документа или оставить незаполненной какую-либо колонку табличной части и попытаться записать документ, то будет получено сообщение об ошибке.

19. Чтобы просмотреть список созданных документов, выполните команду **Приходные накладные** в панели навигации. В форме списка, открывшейся в рабочей области окна приложения увидим два созданных документа, отмеченных пиктограммой указывающей на то, что документы проведены (зеленая галочка в пиктограмме документа)

20. В режиме конфигуратор откройте окно редактирования объекта конфигурации Документ **Приходная Накладная**. В этом окне нас интересует закладка **Формы**, мы видим, ни одна из основных форм документа пока не задана. Для того чтобы создать форму документа, нажмите кнопку открытия значком лупы в поле ввода или кнопку **Добавить** над списком форм. Система вызовет инструмент разработчика - **конструктор форм**.

21. Этот инструмент также построен по принципу «мастеров»: ввод данных в определенной последовательности и передвижение кнопками **Далее** и **Назад**. 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 Приходная Накладная появилась Форма Документа, а на экране открылось окно редактора, содержащее эту форму.

Редактор форм объединяет несколько окон взаимосвязанных между собой редакторов. За более подробной информацией о работе с редактором форм следует обратиться к документации. Необходимо понимать, что при разработке форм объектов конфигурации разработчик не имеет возможности

«нарисовать» форму. Он может только указать, из каких элементов будет состоять форма, а система уже сама самостоятельно расположит эти элементы в форме.

Элементы формы в верхнем левом окне редактора форм образуют иерархическую структуру, из которой следует, что чем выше в списке находится элемент, тем выше и левее на форме он будет располагаться. Эта структура редактируется на закладке **Элементы** и позволяет управлять отображением и редактированием данных в форме. На основе описания в конфигурации документа **ПриходнаяНакладная** система создала структуру элементов, которая определяет, как будет выглядеть форма. Эти элементы имеют разное назначение и разное поведение. Однако все они служат для того, чтобы отображать информацию, хранящуюся в базе данных, и организовывать интерактивную работу с этой информацией. Вы можете попробовать перетащить мышью поля в дереве элементов и поменять местами, например, реквизиты табличной части. Результат изменений сразу отразится в форме документа в нижней части редактора форм. При этом разработчику не нужно задумываться над вопросами конкретного расположения того или иного элемента формы, его размеров и привязки к другим элементам. Эту работу берет на себя система. Но разработчик может через палитру свойств изменить свойства элемента, которые повлияют на его отображение в форме. Он может также изменить структуру элементов формы - создать новое поле, группу полей, добавить табличную часть, связав эти элементы данными формы. Но пока нам ничего этого не нужно делать. Нас интересуют три элемента табличной части: **МатериалыКоличество**, **МатериалыЦена** и **МатериалыСумма**. Нам необходимо, чтобы каждый раз, когда меняется значение в поле **количество** или в поле **Цена**, в поле **Сумма** автоматически устанавливалось значение, равное **Количество*Цена**. Для этого нужно написать на встроенном языке команду, похожую на **Сумма = Количество*Цена**, которая будет выполняться при изменении значения поля **Количество** или **Цена**.

Обработчик события

Система сама умеет работать с теми объектами, которые описаны в дереве конфигурации. То есть у нее есть некие «стандартные представления» о том, как это все должно работать. Но как правило, разработчиков эти «стандартные представления» устраивают только в самых простых случаях. Поэтому у системы существуют *события, которые* связаны с самыми различными моментами ее «стандартного» функционирования. В том числе события, связанные с функционированием форм и элементов, расположенных в этих формах. Используя встроенный язык, разработчик может «вклиниться» в эти события и описать собственный алгоритм того, что должно происходить при наступлении этого события. Для этого дважды щелкнуть на элементе формы **МатериалыКоличество** или правой кнопкой мыши откроем для него палитру свойств (пункт контекстного меню **Свойства**).

22. Прокрутив список до конца, мы увидим перечень событий, которые быть связаны с этим полем.

23. Нам нужно событие ПриИзменении, которое возникает после изменения значения поля. Найдем его в списке событий и нажмем кнопку открытия со значком лупы в поле ввода. Система создаст шаблон процедуры *обработчика* этого события в модГле нашей формы и откроет закладку Модуль редактора формы.

Модуль - это «хранилище» для текста программы на встроенном языке. В данном случае это модуль формы, так как обработчики всех интерактивных событий, связанных с элементами формы, помещаются именно в модуль формы.

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

листинг 1 Процедура « МатериалыКоличествоПриИзменении()»

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

В первой строке мы сначала создаем переменную `СтрокаТабличнойЧасти`, в которую будет помещен объект, содержащий данные, находящиеся в строке табличной части, которую нам нужно пересчитать. Мягкая типизация данных встроенного языка позволяет сделать это, не объявляя переменную и ее тип заранее. Мы создаем переменную прямо по ходу работы, и ее тип определяется типом значения, которое она содержит. Поскольку мы находимся в модуле формы, то в нем доступны все свойства и методы объекта встроенного языка `УправляемаяФорма`. Поэтому мы можем обращаться к ним напрямую. В данном случае после знака равенства мы обращаемся к коллекции элементов формы, используя одно из свойств объекта `УправляемаяФорма` - свойство `Элементы`.

Коллекция элементов формы является объектом встроенного языка `ВсеЭлементыформы`, содержащим все элементы формы. То есть это программный аналог корня дерева элементов формы. Каждый элемент формы можно получить, указав его имя в качестве свойства этого объекта, то есть через точку от него. В данном случае мы обращаемся к табличной части документа `Материалы (Элементы.Материалы)`.

Табличная часть документа представляет собой объект встроенного языка `ТаблицаФормы`. Получить ту строку, в которой в настоящее время осуществляется редактирование, можно при помощи свойства программного объекта `ТаблицаФормы` – `ТекущиеДанные(Элементы..Материалы.ТекущиеДанные)`.

Таким образом, в результате выполнения первой строки процедуры обработчика переменная `СтрокаТабличнойЧасти` будет содержать объект `ДанныеФормыСтруктура`. Этот объект содержит данные, находящиеся в текущей строке табличной части документа (`Элементы.Материалы.ТекущиеДанные`).

Получив этот объект, мы можем обратиться к данным конкретной колонки табличной части, указав имя колонки в качестве свойства объекта. Например, используя обращение `СтрокаТабличнойЧасти.Количество` мы по-

лучаем число, которое находится в редактируемой строке в колонке Количество.

Таким образом, во второй строке процедуры обработчика вычисляется значение колонки Сумма как произведение значений колонок Количество и Цена.

В режиме 1С:Предприятие

24. Теперь посмотрим, как это работает. Запустить 1С:Предприятие в режиме отладки, открыть список документов Приходные накладные и открыть любой из двух созданных документов. Если теперь поменять количество в любой строке документа, то сумма в строке будет пересчитана автоматически.

25. Итак, убедились, что при изменении количества в любой строке документа **Приходная накладная**, сумма в этой строке пересчитывается автоматически. Замечательно. Но теперь 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 **Общий модуль**, расположенные в ветке **Общие ► Общие модули**. Процедуры и функции, содержащиеся в этих модулях, могут быть доступны для любых объектов конфигурации. Поэтому создадим общий модуль и перенесем в него нашу процедуру расчета суммы. А в документе просто оставим вызовы этой процедуры из общего модуля.

26. Добавить объект конфигурации **Общий модуль**. Для этого раскрыть ветвь **Общие** в дереве объектов конфигурации, нажав + слева от нее. Затем выделить ветвь **Общие модули** и нажмите кнопку **Добавить** в ко-

мандной панели окна конфигурации. Назвать его **РаботаСДокументами** и установить в его свойствах флажок **Клиент**(управляемое приложение), а флажок **сервер** снять. Это означает что экземпляры этого модуля будут скомпилированы в контексте тонкого клиента и в контексте веб-клиента.

27. Внести в модуль следующий текст(листинг 2):

листинг 2 Процедура «РассчитатьСумму()»

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

В процедуру **РассчитатьСумму** мы передаем переменную **СтрокаТабличнойЧасти**, которую мы определили в обработчике события **ПриИзменении** поля **Количество**. Она содержит данные редактируемой строки табличной части документа **ПриходнаяНакладная**.

Теперь, используя эту переменную, мы можем получить доступ к данным колонок табличной части и рассчитать сумму как произведение цены на количество.

Ключевое слово **Экспорт** в заголовке процедуры указывает на то, что эта процедура может быть доступна из других программных модулей.

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

листинг 3. Процедура «МатериалыКоличествоПриИзменении()»

```
&НаКлиенте
Процедура МатериалыКоличествоПриИзменении(Элемент)
СтрокаТабличнойЧасти=Элементы.Материалы.ТекущиеДанные;
РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);
КонецПроцедуры
```

Мы видим, что первая строка процедуры осталась без изменений. А во второй строке вместо непосредственного расчета суммы мы вызываем

процедуру **РассчитатьСуммуО** из общего модуля **РаботаСДокументами** и передаем ей в качестве параметра текущую строку табличной части. Проверим, как это работает, и убедимся, что ничего не изменилось.

Теперь осталось и для поля **Цена** установить такой же обработчик. Так как однажды мы уже написали в модуле формы нужную процедуру, то мы просто могли бы сопоставить ее также и другому событию другого элемента управления, расположенного в форме. Однако стандарты разработки конфигураций фирмы «1С» не допускают такого решения.

Согласно стандартам разработки фирмы «1С» у каждого события должен быть свой обработчик. Если одинаковые действия должны выполняться при изменении разных элементов управления (например, при нажатии нескольких кнопок), то в этом случае следует поступать следующим образом:

- создается отдельная процедура (функция), выполняющая необходимые действия;
- для каждого элемента управления создается отдельный обработчик с именем, назначаемым по умолчанию;
- из каждого обработчика вызывается требуемая процедура (функция).

Поэтому мы создадим обработчик события **ПриИзменении** для поля личной части **МатериалыЦена** так же, как мы делали это для **МатериалыКоличество**, и повторим в нем вызов процедуры **РассчитатьСумму** из общего модуля (листинг 4).

листинг 4. Процедура «МатериалыЦенаПриИзменении()»

```
&НаКлиенте
```

```
Процедура МатериалыЦенаПриИзменении(Элемент)
```

```
СтрокаТабличнойЧасти=Элементы.Материалы.ТекущиеДанные;
```

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

```
КонецПроцедуры
```


28. Запустите 1С:Предприятие в режиме отладки и убедитесь, что теперь сумма в строках табличной части документов **Приходная Накладная** рассчитывается как при изменении количества, так и при изменении цены.

29. Теперь аналогичным образом создать второй документ, необходимый нам, - **Оказание услуги**. Для этого потребуется выполнить уже знакомые нам действия, которые мы выполняли по созданию документа **Приходная накладная**

30. Добавить новый объект конфигурации Документ и назвать его **Оказание Услуги**. Представление объекта задавать не будем, вместо него будет использоваться Синоним объекта. Представление списка задать как **Оказание услуг**. На закладке Подсистемы отметьте, что документ будет доступен в подсистемах **Оказание услуг** и **Бухгалтерия**.

31. На закладке **Данные** создайте реквизиты документа:

- **Склад**, тип **СправочникСсылка.Склады**. Выбрать для свойства **Значение заполнения** предопределенный элемент **Основной справочника Склады**.

- **Клиент**, тип **СправочникСсылка.Клиенты**. Установить свойство **Проверка заполнения** в значение **Выдавать ошибку**.

- **Мастер**, тип **СправочникСсылка.Сотрудники**. Установить свойство **Проверка заполнения** в значение **Выдавать ошибку**.

32. Создайте 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;

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

33. На закладке **Формы** создать основную форму документа. Для поля **Перечень Номенклатуры** **Количество** создать обработчик события **При-**

Изменении, в котором будем вызывать процедуру РассчитатьСумму из общего модуля РаботаСДокументами. При этом откроется модуль формы с шаблоном обработчика события ПереченьНоменклатурыКоличествоПриИзменении, который пока заполнять не надо, а перейти в окно элементов формы на закладку Форма и аналогичным образом создать обработчик события ПереченьНоменклатурыЦенаПриИзменении для поля ПереченьНоменклатурыЦена.

34. Модуль формы документа ОказаниеУслуги заполнить следующим образом(листинг 5)

листинг 5. Модуль формы документа «ОказаниеУслуги»

```
&НаКлиенте
Процедура
ПереченьНоменклатурыКоличествоПриИзменении(Элемент)
    СтрокаТабличнойЧасти=Элементы.ПереченьНоменклатуры.ТекущиеДанные;
    РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);
    КонецПроцедуры

&НаКлиенте
Процедура ПереченьНоменклатурыЦенаПриИзменении(Элемент)
    СтрокаТабличнойЧасти=Элементы.ПереченьНоменклатуры.ТекущиеДанные;
    РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);
    КонецПроцедуры
```

35. Отредактируем командный интерфейс, чтобы в подсистеме **Оказание услуг** была доступна команда создания новых документов. Запустим 1С:Предприятие в режиме отладки. В панели действий раздела **Оказание услуги** и вызовем команду создание документа **Оказание услуги** и заполнить его.

Контрольные вопросы

1. Какими характерными особенностями обладает документ. Для чего предназначены реквизиты и табличные части документа.
2. Какие существуют основные формы документа. Что такое проведение документа.
3. Как создать объект конфигурации Документ и описать его основную структуру.
4. Как создать новый документ и заполнить его данными.
5. Как создать собственную форму документа.
6. Что такое конструктор форм.
7. Что такое редактор форм.
8. Что такое элементы формы.
9. Что такое события и с чем они связаны.
10. Что такое обработчик события и как его создать.
11. Что такое модуль и для чего он нужен.
12. Зачем нужны общие модули.
13. Что такое типобразующие объекты.

Лабораторная работа № 4

Тема работы: Добавление периодического регистра сведений. Автоматическая подстановка цены в документ при выборе номенклатур

Цель работы: Научиться создавать различные регистры.

Ход выполнения работы.

Один из главных моментов разработки любой конфигурации - создание механизма учета накопления данных.

В системе 1С:Предприятие есть несколько объектов конфигурации, которые позволяют создавать в базе данных структуры, назначенные для накопления информации в удобном для последующего анализа виде. Использование таких «хранилищ» данных дает нам, с одной стороны, накапливать в них данные, поставляемые различными документами (или другими объектами базы данных), а с другой стороны, легко создавать нужные нам отчеты или использовать эти данные в алгоритмах работы конфигурации.

В конфигурации существует несколько объектов, называемых *регистрами* для описания подобных «хранилищ». Объект конфигурации *Регистр накопления* предназначен для описания структуры накопления данных. На основе объекта конфигурации *Регистр накопления* платформа создает в базе данных таблицы, в которых будут накапливаться данные, «поставляемые» различными объектами базы данных.

Эти данные будут храниться в таблицах в виде отдельных записей, каждая из которых имеет одинаковую, заданную в конфигураторе структуру.

На основании таблицы движений регистра накопления система рассчитывает таблицу итогов регистра, которая хранит в базе данных итоги на момент времени последнего движения (актуальные итоги).

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. Основным назначением регистра накопления является накопление числовой информации в разрезе нескольких *измерений*, которые описываются разработчиком в соответ-

ствующем объекте конфигурации Регистр накопления и являются подчиненными объектами конфигурации.

Виды числовой информации, накапливаемой регистром накопления называются *ресурсами*, также являются подчиненными объектами и описываются в конфигураторе.

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

Изменение состояния регистра накопления происходит, как правило, при введении документа и заключается в том, что в регистр добавляется некоторое количество записей. Каждая запись содержит значения измерений, значения приращений ресурсов, ссылку на документ, который вызвал эти изменения (регистратор), и «направление» приращения (приход или расход). Такой набор записей называется *движениями* регистра накопления. Каждому движению регистра накопления всегда должен соответствовать *регистратор* – объект информационной базы (как правило, документ), который произвел эти движения.

Кроме этого, регистр накопления может хранить дополнительную информацию, описывающую каждое движение. Набор такой дополнительной информации задается разработчиком при помощи *реквизитов* объекта конфигурации Регистр накопления.

Теперь когда мы знаем, для чего предназначены регистры накопления, посмотрим, как можно их использовать в нашем примере.

Прежде всего, нас интересует информация о том, сколько и каких материалов есть у нас на складах. Для накопления такой информации создается регистр ОстаткиМатериалов.

1. Открыть в конфигураторе нашу учебную конфигурацию и добавим Объект конфигурации Регистр накопления. Для этого выделим в дереве объектов конфигурации ветвь Регистры накопления и нажмем кнопку Добавить в командной панели окна конфигурации. В открывшемся окне редакци-

рования объекта конфигурации на закладке Основные зададим имя регистра - ОстаткиМатериалов. Также зададим и Расширенное представление списка как Движения по регистру Остатки материалов. Этот заголовок будет отображаться в окне списка записей регистра.

2. Нажмем Далее и перейдем на закладку Подсистемы. По логике нашей конфигурации данный регистр должен быть доступен в разделах Учет материалов, Оказание услуг и Бухгалтерия. Поэтому отметим в списке подсистем эти подсистемы.

3. Выделим закладку Данные и перейдем к созданию структуры регистра. Создадим измерения регистра:

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

4. Для этого выделим ветвь Измерения и нажмем кнопку Добавить в командной панели окна.

Затем создадим ресурс Количество с длиной 15 и точностью 3. Для этого выделить ветвь Ресурсы и нажать кнопку Добавить в командной панели окна. В результате этих действий регистр ОстаткиМатериалов должен иметь следующий вид:

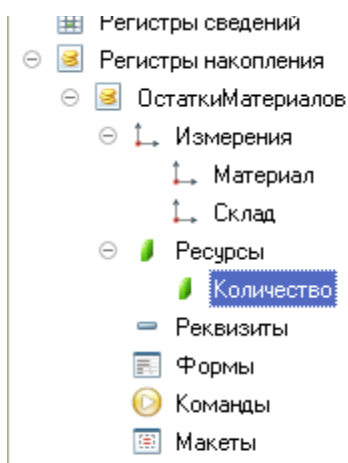


рис.1. Регистр «ОстаткиМатериалов»

Если вы сейчас попытаетесь 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.

Поэтому мы сформируем движения регистра накопления ОстаткиМатериалов в процессе проведения двух созданных нами документов ПриходнаяНакладная и ОказаниеУслуг.

Движения документа

Движения документа - это записи в регистрах, которые создаются в процессе проведения документа и отражают изменения, производимые документом.

5. Откроем окно редактирования объекта конфигурации Документ ПриходнаяНакладная. Перейдем на закладку Движения, раскроем список Регистры накопления и отметим регистр накопления ОстаткиМатериалов.

6. Обратите внимание, что сразу после отметки выбранного регистра становится доступной кнопка Конструктор движений. Нажмем ее и воспользуемся этим конструктором. Конструктор устроен просто. В списке Регистры перечислены регистры, в которых документ может создавать движения. В нашем случае там пока один регистр ОстаткиМатериалов. В списке Реквизиты документа должны находиться исходные данные для создания движений - реквизиты документа ПриходнаяНакладная. А в таблице Поле - Выражение должны быть заданы формулы, по которым будут вычисляться значения измерений и ресурсом регистра при записи движений. Обратите внимание, что по умолчанию конструктор предлагает нам 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 нашего документа - Материалы. Список реквизитов документа, который уже заполнен реквизитами шапки документа, автоматически дополнится реквизитами нашей табличной части.

7. Теперь нажмем кнопку Заполнить выражения. В нижнем окне сформируется соответствие полей (измерений и ресурсов) регистра и выражений для их расчета. Конструктор движений установит 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.

8. Нажмем кнопку ОК и посмотрим, какой текст сформировал конструктор в модуле документа ПриходнаяНакладная(Листинг 1).

Процедура ОбработкаПроведения(Отказ, Режим)

```
//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

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

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

```
// регистр ОстаткиМатериалов Приход
```

```
Движения.ОстаткиМатериалов.Записывать = Истина;
```

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

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

```
    Движение.ВидДвижения = ВидДвиженияНакопления.При-
```

```
ход;
```

```
    Движение.Период = Дата;
```

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

```
    Движение.Склад = Склад;
```

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

```
КонецЦикла;
```

```
//}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

```
КонецПроцедуры
```


9. Конструктор создал обработчик события `ОбработкаПроведения` объекта конфигурации `Документ ПриходнаяНакладная`, поместил его в модуль объекта и открыл текст модуля. Событие `ОбработкаПроведения` является одним из важнейших событий, связанных с документом. Это событие возникает при проведении документа. Основное назначение обработчика данного события - генерация движений по документу. Выполнение различных операций с данными в процедуре обработчика влияет на состояние учета. Таким образом, именно в эту процедуру разработчик должен поместить собственные алгоритмы преобразования данных, выполняемые в момент проведения документа.

Пояснение текста процедуры обработчика. Объект встроенного языка `ДокументОбъект` имеет свойство `Движения`, возвращает объект `КоллекцияДвижений`, содержащий коллекцию наборов записей регистров, по которым этот документ может формировать движения.

К конкретному набору записей этой коллекции можно обратиться, указав через точку имя регистра, которому принадлежит этот набор записей. Например, `Движения.ОстаткиМатериалов`.

Затем через точку можно использовать различные методы набора записей регистра, например, `Движения.Остатки Материалов.Добавить()`. Метод `Добавить()` добавляет новую запись в набор записей.

В первой строке процедуры мы устанавливаем свойство `Записывать` набора записей регистра в значение `Истина`. То есть в явном виде указываем, что после завершения обработки проведения платформа должна будет записать этот набор записей в базу данных.

Внутри обработчика расположен цикл `Для Каждого ... Из ... Цикл`. Он предназначен для перебора строк табличной части нашего документа.

В цикле обращение к табличной части документа происходит по имени (`Материалы`). Переменная `ТекСтрокаМатериалы` содержит объект с данными

текущей строки табличной части документа. Эта переменная создается в начале цикла и меняется по мере его прохождения.

В первой строке тела цикла, используя метод `Добавить()`, мы добавляем к набору записей, который создает наш документ в регистре, новую запись. Тем самым мы создаем объект `РегистрНакопленияЗапись` и сохраняем его в переменной `Движение`.

Используя этот объект, мы можем обратиться к полям этой записи, указав имя поля через точку от этой переменной (например. `Движение.Количество`).

Причем `Движение.Материал`, `Движение.Склад` - это измерения регистра, `Движение.Количество` - это ресурс регистра, а `Движение.ВидДвижения` и `Движение.Период` - стандартные реквизиты регистра, которые создаются автоматически.

Чтобы присвоить полям новой записи регистра соответствующие значения полей документа, мы обращаемся к полям табличной части, указав имя поля через точку от переменной `ТекСтрокаМатериалы` (например, `ТекСтрокаМатериалы.Материал`).

Заметим, что `Склад` - это реквизит шапки документа, а `Дата` – стандартный реквизит документа, который создается автоматически, в цикле меняются только значения полей табличной части документа - `ТекСтрокаМатериалы.Материал` и `ТекСтрокаМатериалы.Количество`. Остальные поля не меняются, так как относятся к документу в целом и не зависят от текущей строки табличной части документа.

ВидДвиженияНакопления.Приход - это значение системного перечисления, которое определяет вид движения регистра накопления как `Приход`.

Таким образом, мы присваиваем нужные значения всем полям новой записи. После перебора всех строк документа (после завершения в этом наборе записей (`Движения.ОстаткиМатериалов`)) будет содержаться столько записей, сколько строк в табличной части проводимого документа.

10. Если теперь откроем окно редактирования объекта конфигурации накопления ОстаткиМатериалов и перейдем на закладку Регистраторы, то в списке документов, созданных в конфигурации, мы увидим отмеченный документ ПриходнаяНакладная, так как мы задали в модуле этого документа формирование движений в регистре ОстаткиМатериалов.

11. Отредактируем командный интерфейс, чтобы в подсистемах Бухгалтерия, Оказание услуг и Учет материалов была доступна ссылка для просмотра записей нашего регистра накопления. Дело в том, что команды открытия регистров также добавляются в панель навигации подсистем, но по умолчанию они невидимы, в отличие от команд открытия справочников и документов.

12. В дереве объектов конфигурации выделим ветвь Подсистемы, вызовем ее контекстное меню и выберем пункт Все подсистемы. В открывшемся окне слева в списке Подсистемы выделим подсистему УчетМатериалов. Справа в списке Командный интерфейс отразятся все команды выбранной подсистемы. В группе Панель навигации.Обычное включим видимость у команды Остатки материалов и мышью перетащим ее в группу Панель навигации. Действительно, команды открытия регистров накопления не так часто используются, и поэтому лучше их перенести в группу См.также панели навигации разделов интерфейса. Аналогично, выделив подсистемы ОказаниеУслуг и Бухгалтерия, в панели навигации в группе Обычное включим видимость у команды Остатки материалов и перенесем ее в группу См.также.

13. Запустим 1С:Предприятие в режиме отладки и протестируем внесенные нами изменения. В открывшемся окне 1С:Предприятия мы видим, что в панели ни в группе См.также разделов Бухгалтерия, Оказание услуг и Учет материалов появилась команда для открытия списка регистра Остатки материалов.

14. Чтобы проследить связь между проведением документа и накоплением информации в регистре, откроем список приходных накладных, выполнив команду Приходные накладные в разделе Бухгалтерия.

15. Откроем Приходную накладную № 1 и нажмем Провести и закрыть, т.е. перепроведем ее. То же самое сделаем для Приходной № 2.

16. Перепровести документы можно и не открывая документов. Для этого нужно выделить нужный документ в списке (или выделить группу документов, удерживая клавишу Ctrl), нажать кнопку Все действия в командной панели формы списка и выбрать пункт Провести.

17. Выполнить команду Остатки Материалов и открыть список регистра накопления.

18. При проведении приходных накладных появляются соответствующие записи в регистре накопления Остатки материалов. Все поля регистра заполнились данными документов так, как мы задали в обработчике проведения документа ПриходнаяНакладная. Пиктограмма со знаком + слева от каждой записи указывает на тип движения - Приход. как видим, заголовок формы списка записей регистра соответствует заданному нами в свойстве Расширенное представление для этого регистра.

19. При реальной работе записей в регистре ОстаткиМатериалов будет много и будет трудно понять, какие записи относятся к определенному документу. Поэтому наряду с общим списком регистра хотелось бы иметь возможность вызывать из формы документа список регистра, в котором показаны движения, произведенные только этим документом.

20. Чтобы реализовать такую возможность, вернемся в конфигуратор форму документа ПриходнаяНакладная. В левом верхнем окне перейдем на закладку Командный интерфейс.

21. В разделе Панель навигации раскроем группу Перейти и увидим команду для открытия списка регистра накопления Остатки материалов. Эта команда была автоматически помещена в панель навигации формы докумен-

та, так как он является регистратором, то есть создает движения в нашем регистре.

22. Установим свойство Видимость для этой команды.

23. Запустим 1С:Предприятие в режиме отладки и откроем Приходную накладную № 2. В форме документа появилась панель навигации, в которой мы можем переходить к списку записей регистра Остатки Материалов, связанному с документом, и обратно к содержимому документа.

24. Обратите внимание, что до этого панель навигации в форме приходной накладной была не видна, так как в ней не было отображено ни одной команды.

25. Теперь мы аналогичным образом создадим движения документа ОказаниеУслуги. Для этого потребуется выполнить уже знакомые нам действия.

26. Открыть окно редактирования объекта конфигурации Документ ОказаниеУслуги. перейти на закладку Движения и в списке регистров конфигурации регистр накопления ОстаткиМатериалов.

27. Нажмем кнопку Конструктор движений. В открывшемся окне конструктора изменим тип движения регистра на Расход, так как документ ОказаниеУслуги должен расходовать материалы. Пиктограмма слева от названия регистра изменится на знак -. В поле выбора Табличная часть выберем табличную часть нашего документа - ПереченьНоменклатуры. Список реквизитов документа, который уже заполнен реквизитами документа, автоматически дополнится реквизитами нашей табличной части.

28. Теперь нажмем кнопку Заполнить выражения. В нижнем окне сформируется соответствие полей (измерений и ресурсов) регистра и выражений для их расчета. Однако при автоматическом заполнении поле Материал не заполнится. Так происходит потому, что имя поля табличной части - Номенклатура не совпадает с именем измерения регистра - Материал. Если мы оставим это так, как есть, то в регистре накопления в строках с типом регистра - расход номенклатура фиксироваться не будет. Чтобы из-

бежать этого, нужно выделить поле регистра Материал и в окне Реквизиты документа дважды щелкнуть по строке ТекСтрокаПереченьНоменклатуры. Номенклатура.

29. Таким образом, номенклатура для движений регистра накопления будет выбираться из табличной части документа. Нажмем кнопку ОК.

30. Конструктор создал обработчик события ОбработкаПроведения объекта конфигурации Документ ОказаниеУслуги и поместил его в модуль объекта (листинг 2).

```
Процедура ОбработкаПроведения(Отказ, Режим)
//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную
изменения будут утеряны!!!
// регистр ОстаткиМатериалов Расход
Движения.ОстаткиМатериалов.Записывать = Истина;
Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНомен-
клатуры Цикл
    Движение = Движения.ОстаткиМатериалов.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
    Движение.Период = Дата;
    Движение.Склад = Склад;
    Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количе-
ство;
КонецЦикла;

//}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

31. Обратите внимание, что строка Движение.ВидДвижения = ВидДвиженияНакопления.Расход определяет вид движения регистра накопления, производимый этим документом как Расход, а в остальном процедура обработки документа ОказаниеУслуги идентична обработке документа ПриходнаяНакладная (см. листинг 1).

32. Отредактируем командный интерфейс формы документа, чтобы в панели навигации формы иметь возможность переходить к списку записей регистра Остатки Материалов, связанному с документом. Для этого откроем форму документа ОказаниеУслуги. В верхнем окне перейдем на закладку Командный интерфейс. В разделе панель навигации раскроем группу Перейти и установим видимость для команды открытия регистра накопления Остатки материалов.

33. Запустим 1С:Предприятие в режиме отладки и в разделе Оказание услуг откроем документ Оказание услуги № 1 и нажмем Провести и закрыть, то есть перепроведем его.

34. Выполните команду Остатки материалов и откройте список нашего регистра накопления.

35. Мы видим, что в регистре накопления Остатки материалов появилась еще одна запись, что соответствует количеству строк в табличной части проведенного документа. Все поля регистра заполнились данными документа так, как мы задали в обработчике проведения документа Оказание услуги. Пиктограмма со знаком «минус» слева от записи указывает на тип движения – Расход.

36. Сейчас мы видим весь список движений регистра. Открыв это, список из формы документа, мы можем отфильтровать движения по документу-регистратору. Для этого откроем еще раз документ Оказание услуги № 1. В форме документа появилась панель навигации, в которой мы можем переходить к списку записей регистра Остатки материалов, связанному с документом, и обратно к содержимому документа. 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 расходуемые материалы, но и услуги. Поэтому в регистр Остатки материалов будут попадать записи и о расходуемых услугах, что неправильно. Пока мы ничего не будем делать с движениями, которые сформировал конструктор.

Способы работы с коллекцией

В процессе формирования движений документов, когда в цикле и табличные части документов ПриходнаяНакладная и ОказаниеУслуги, мы столкнулись с одним из объектов встроенного языка, который является коллекцией. Многие объекты встроенного языка являются коллекциями.

Коллекция представляет собой совокупность объектов. Существуют принципы работы с любой коллекцией.

Во-первых, доступ к каждому объекту коллекции возможен путем перебора элементов коллекции в цикле. Для этого используется конструкция языка Для Каждого Из ... Цикл ... (листинг 3).

Листинг 3. Перебор элементов коллекции в цикле

```
Для Каждого СтрокаТабличнойЧасти из Таблич-
наяЧасть Цикл
Сообщить(СтрокаТабличнойЧасти.Услуга);
КонецЦикла
```

В этом примере ТабличнаяЧасть — это коллекция строк табличной объекта конфигурации. При каждом проходе цикла в переменной СтрокаТабличнойЧасти будет содержаться очередная строка коллекции.

Во-вторых, существует доступ напрямую к элементу коллекции, без а коллекции в цикле. Здесь возможны различные комбинации двух обращений.

1. Во встроенном языке бывают именованные коллекции. То есть коллекции, в которых каждый элемент имеет некоторое уникальное имя. В этом случае обращение к элементу коллекции возможно этому имени (листинг 4).

Листинг 4. Обращение к элементу коллекции

Справочники. Сотрудники;
Справочники[«Сотрудники»];

В этом примере Справочники - это коллекция менеджеров всех справочников, содержащихся в конфигурации. Так как каждый справочник конфигурации имеет свое уникальное имя, то к конкретному элементу этой коллекции (к менеджеру конкретного справочника) можно обратиться, указав имя этого справочника: Справочники[«Сотрудники»].

2. Если нет смысла в «персонификации» элементов коллекции (коллекция неименованная), тогда обращение к элементу коллекции возможно по индексу (индекс первого элемента коллекции - ноль), листинг 5.

Листинг 5. Обращение к элементу коллекции по индексу

ТабличнаяЧасть[0]

В этом примере ТабличнаяЧасть - это коллекция строк табличной части объекта конфигурации. И мы обращаемся к первому элементу этой коллекции, указывая его индекс-0

Следует отметить, что существуют коллекции, сочетающие оба вида обращений. Например, к коллекции колонок таблицы значений можно обращаться как по именам колонок, так и по индексу.

Контрольные вопросы

1. Для чего предназначен объект конфигурации Регистр накопления.
2. Почему следует использовать регистры, хотя необходимая информация содержится в других объектах.
3. Для чего нужны измерения регистра, ресурсы и реквизиты.
4. Что такое движения регистра и что такое регистратор.
5. Как создать новый регистр накопления и описать его структуру.

6. Как создать движения документа с помощью конструктора движений.
7. Как средствами встроенного языка обойти табличную часть документа и обратиться к ее данным.
8. Как показать команды открытия списка регистра в интерфейсе конфигурации и в интерфейсе формы.

Лабораторная работа № 5

Тема: Проведение документа «Оказание услуги» по двум регистрам.

Цель занятия: Научиться проводить документ по нескольким регистрам

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Зачем нужно проведение документа по нескольким регистрам

В нашей базе учитывается только количественное движение материалов, очевидно, что необходимо также знать, какие денежные средства были затрачены на приобретение тех или иных материалов и каковы материальные запасы в денежном выражении. На ряду с этим необходимо, чтобы весь суммовой учет материалов велся по средней стоимости. То есть при закупке материалов они должны учитываться в ценах приобретения, а при расходе - по средней стоимости, которая рассчитывается исходя из общей суммы закупок данного материала и общего количества этого материала.

Поскольку подобная информация имеет совершенно другую структуру, нежели количественный учет, для хранения данных об общей стоимости тех или иных материалов будет использоваться еще один регистр накопления **СтоймостьМатериалов**.

Таким образом, документы **ПриходнаяНакладная** и **ОказаниеУслу-**

ги должны будут создавать движения не только в регистре **Остатки Материалов**, но одновременно и в регистре **Стоимость Материалов**, отражая изменения суммового учета.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Добавление еще одного регистра накопления

В режиме Конфигуратор

Создадим новый объект конфигурации Регистр накопления с именем **Стоимость Материалов**.

- На закладке **Основные**: зададим **Расширенное** представление списка **Движения по регистру Стоимость материалов**.

- На закладке **Подсистемы**: отметим, что этот регистр будет отображаться в подсистемах **Бухгалтерия, Учет материалов и Оказание услуг**.

- На закладке **Данные**: создадим для регистра одно измерение - **Материал** с типом **Справочник Ссылка. Номенклатура** и один ресурс - **Стоимость** с длиной 15 и точностью 2.

- *Отредактируем командный интерфейс:*

В дереве объектов конфигурации выделим ветвь **Подсистемы**, вызовем ее контекстное меню и выберем пункт **Все подсистемы**. В открывшемся окне слева в списке **Подсистемы** выделим подсистему **Бухгалтерия**. В группе **Панель навигации. Обычное** включим видимость у команды **Стоимость материалов** и мышью перетащим ее в группу **Панель навигации. См. также**.

Аналогично, выделив подсистемы **Оказание Услуг** и **Учет Материалов**, в панели навигации в группе **Обычное** включим видимость у команды **Стоимость материалов** и перенесем ее в группу **См. также**.

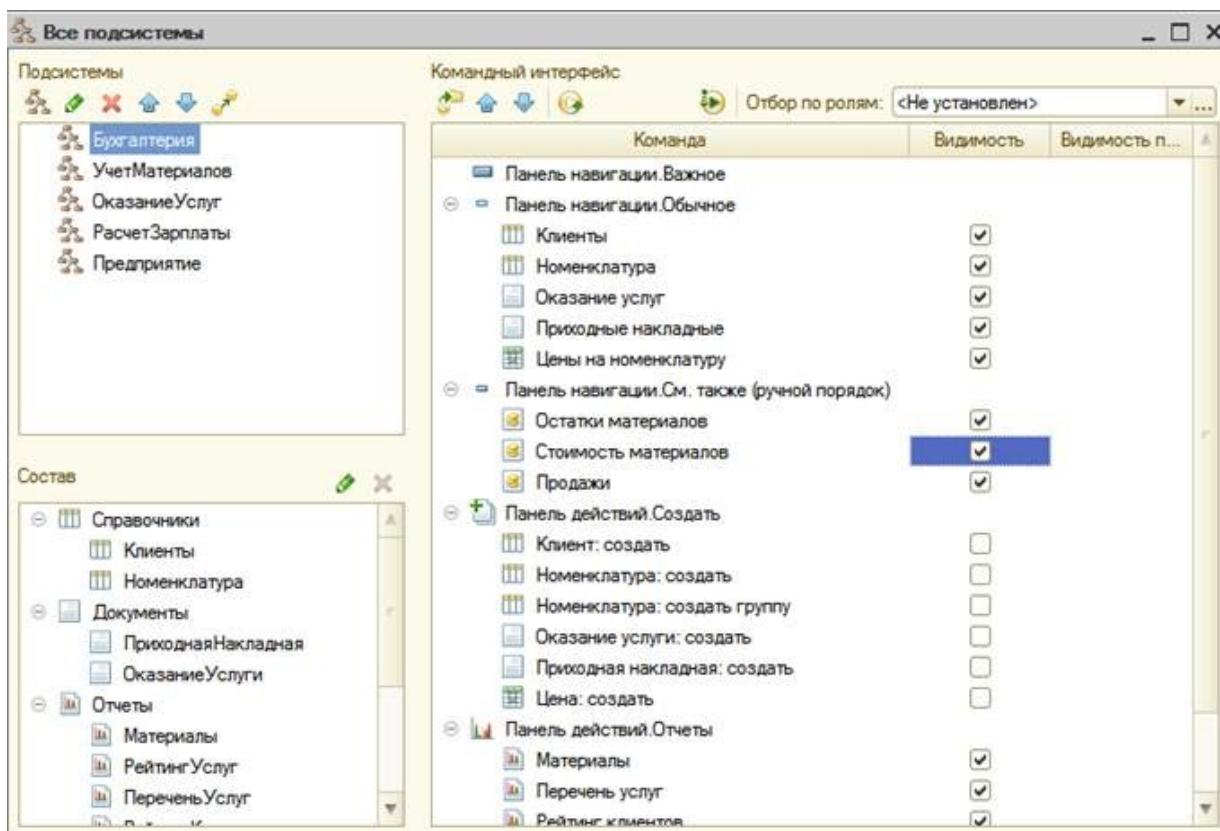


Рис. Проведение приходной накладной по двум регистрам

Откроем в конфигураторе окно редактирования объекта конфигурации Документ **Приходная Накладная**.

- На закладке **Движения**: В списке регистров отметим **Стоимость Материалов**.

- На закладке **Прочее**: нажмем кнопку **Модуль объекта**, откроем процедуру обработчика события **Обработка Проведения**. И изменим код следующим образом: (см.листинг):

Листинг 11.1

Процедура ОбработкаПроведения(Отказ, Режим)

Движения.ОстаткиМатериалов.Записывать = Истина;

Движения.СтоимостьМатериалов.Записывать = Истина;

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*:

//регистр *СтоимостьМатериалов* *Приход*
Движение = *Движения.СтоимостьМатериалов.Добавить()*;
Движение.ВидДвижения = *ВидДвиженияНакопления.Приход*;
Движение.Период = *Дата*;
Движение.Материал = *ТекСтрокаМатериалы.Материал*;
Движение.Стоимость=*ТекСтрокаМатериалы.Сумма*;

КонецЦикла;

КонецПроцедуры

Команда перехода к записям регистра

В заключение отредактируем командный интерфейс формы документа, чтобы в панели навигации формы иметь возможность переходить к списку записей регистра **СтоимостьМатериалов**, связанному с документом.

- Откроем форму документа **ПриходнаяНакладная**: перейдем на закладку **Командный интерфейс**. В разделе **Панель навигации** раскроем группу **Перейти** и увидим команду для открытия регистра накопления **Стоимость материалов**. Установим свойство **Видимость** для этой команды:

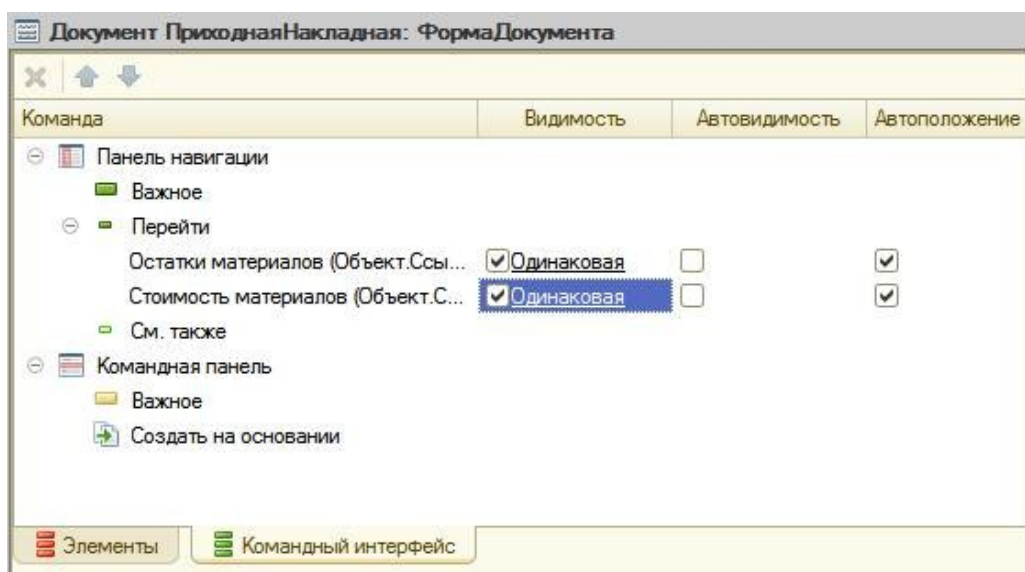


Рис. Редактируем командный интерфейс формы документа

В режиме 1С:Предприятие

В режиме 1С:Предприятие необходимо перепровести все приходные накладные. Это необходимо для того, чтобы эти документы создали новые записи в регистрах, в соответствии с алгоритмом проведения. Запустим 1С: Предприятие в режиме отладки. Откроем список документов, выполнив команду **Приходные накладные** в панели навигации раздела **Учет материалов**.

*Выделим одновременно, используя клавишу Ctrl, все приходные накладные и перепроведем их, выполнив команду **Все действия** > **Провести**.*

Проведение документа «Оказание услуги» по двум регистрам

В заключение внесем изменения в процедуру обработки проведения документа **ОказаниеУслуги**.

Это необходимо для того, чтобы 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 исходя из текущих конъюнктурных соображений. Поскольку в документе **ОказаниеУслуги** отражена только цена номенклатуры, необходимо сделать следующее:

- Добавить в табличную часть документа еще один реквизит, в котором будет указываться стоимость номенклатуры.
- После этого изменить процедуру проведения документа **ОказаниеУслуги**.
- В режиме 1С: Предприятие перепровести все эти документы.

В режиме Конфигуратор

Откроем в конфигураторе окно редактирования объекта конфигурации Документ **ОказаниеУслуги**

- На закладке **Данные**: создадим новый реквизит табличной части документа с именем **Стоимость**, типом **Число**, длиной 15 и точностью 2, и отметим флажок **неотрицательное**:

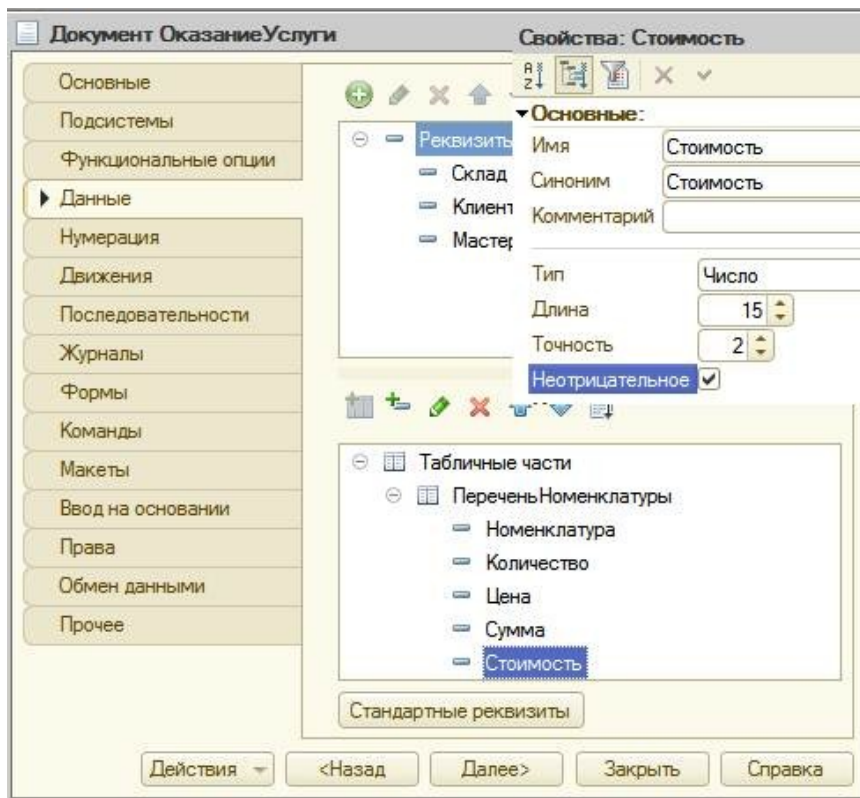


Рис. 11.3. Редактируем свойства *Стоимость*

Откроем форму **ФормаДокумента** документа **ОказаниеУслуг** и добавим табличную часть **ПереченьНоменклатуры** поле, отображающее новый реквизит **Стоимость**:

- На закладке *Реквизиты*: раскроем реквизит формы **Объект**. Найдем в табличной части реквизит **Стоимость** и с помощью мыши перетащим его в окно элементов формы, расположенное слева в верхней части редактора форм:

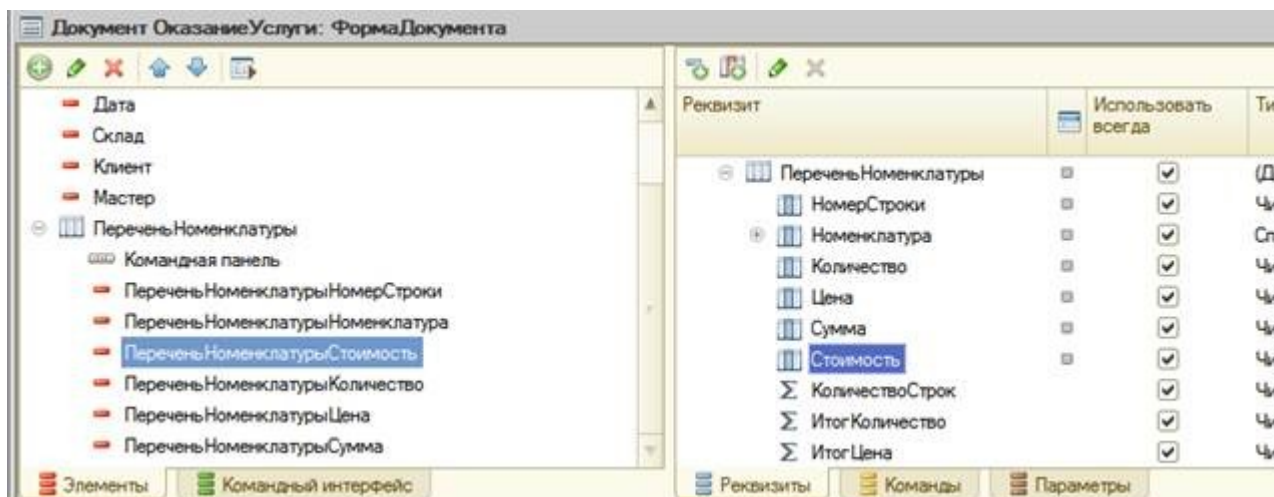


Рис. Редактируем реквизит формы Объект

Новый элемент расположим в структуре элементов формы после поля Номенклатура. Оставим свойства элемента формы, предложенные по умолчанию.

Новый реквизит сразу же отобразится в форме документа, расположенной в левом нижнем окне редактора форм.

Изменение процедуры проведения

Создадим движения документа **ОказаниеУслуги** таким же образом, как мы делали это для документа **Приходная Накладная**.

- На закладке **Движения**: в списке регистров отметим, что документ будет создавать теперь движения и по регистру **СтоимостьМатериалов**.

- На закладке **Прочее**: нажмем кнопку **Модуль объекта**. Откроем процедуру обработчика события **ОбработкаПроведения**. Отредактируем код следующим образом:

Листинг 11.2

Процедура ОбработкаПроведения(Отказ, Режим)

Движения.ОстаткиМатериалов.Записывать = Истина;

Движения.СтоимостьМатериалов.Записывать = Истина;

Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры *Цикл*

Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры = Перечисления.ВидыНоменклатуры.Материал
Тогда

//регистр ОстаткиМатериалов Расход
Движение = Движения.ОстаткиМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклату-

Создадим и проведем еще два документа **Оказание услуги**. Для этого в форме списка документов нажмем кнопку Создать. Эти документы понадобятся нам в дальнейшем.

Контрольные вопросы

- Для чего может понадобиться проведение документа по нескольким регистрам.
- Как создать движения документа по нескольким регистрам в обработчике проведения документа.
- Как создать движения документа без использования конструктора движений.
- Как средствами встроенного языка сформировать и записать движения документа в регистр накопления.
- Как добавить в форму документа новый реквизит.

Лабораторная работа № 6

Тема работы: Вывод данных по всем дням в выбранном периоде. Получение Актуальных значений из периодического регистра сведений.

Цель работы: Научиться создавать различные отчеты.

Ход выполнения работы.

Теоретическая часть.

На этом занятии мы познакомимся с объектом конфигурации *Отчет*. Вы узнаете, для чего он используется, и создадите отчет, который будет показывать движения и остатки материалов на предприятии.

Что такое отчет?

Объект конфигурации *Отчет* 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 пр.

Добавление отчета В режиме Конфигуратор

Теперь у нас все готово для того, чтобы можно было пол выходные данные. Поэтому приступим к созданию отчета, кот будет показывать нам приход, расход и остатки материалов.

Можно быстро и легко разработать отчет с использованием только визуальных средств разработки «без единой строчки кода».

1. Откроем в конфигураторе нашу учебную конфигурацию и добавим новый объект конфигурации Отчет. Для этого выделим в дереве объектов конфигурации ветвь **Отчеты** нажмем кнопку **Добавить** в командной панели окна конфигурации. В открывшемся окне редактирования объекта конфигурации на закладке **Основные** зададим имя отчета - **Материалы**. Больше никаких свойств, определяющих представление об в интерфейсе приложения, задавать не будем. Вместо них будет использоваться **Синоним** объекта, который создается автоматически на основании имени объекта.

2. Создадим основу для построения любого отчета - *схему компоновки данных*. Для этого нажмем кнопку **Открыть схему компоновки да** или кнопку открытия со значком лупы.

Макет

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

3. Нажмем кнопку Готово. Платформа создаст новый макет, содержащий схему компоновки данных, и сразу же откроет конструктор схемы компоновки данных. Конструктор обладает большим количеством возможно-

стей для визуального проектирования отчетов, но мы сейчас воспользуемся только самыми простыми его возможностями и определим те данные, которые хотим видеть в результате работы нашего отчета.

4. Добавим новый **набор данных - запрос**. Для этого нажмем кнопку **Добавить** или вызовем контекстное меню ветки **Наборы данных**. Для того чтобы создать текст запроса, запустим конструктор запроса - нажмем кнопку **Конструктор запроса**. *Конструктор запроса* - инструмент, созданный для помощи **разработчику**, позволяющий визуально конструировать запрос, пользователь, не знакомый с языком запросов, может с помощью конструктора создать синтаксически правильный запрос.

5. В списке **База данных** представлены таблицы для создания **запроса**. На основе их данных мы имеем возможность построить отчет. Если раскрыть ветку **РегистрыНакопления** то мы увидим, что кроме таблицы регистра **ОстаткиМатериалов** в этой ветке присутствуют еще несколько *виртуальных таблиц*, которые формирует **система**.

6. Эти виртуальные таблицы, создаваемые платформой для регистров, и используются в основном для построения различных отчетов. Поскольку мы хотим видеть как остатки материалов, так и информацию об их поступлении и расходовании, нас будет интересовать виртуальная таблица **ОстаткиМатериалов.ОстаткиИОбороты**. Раскроем ее.

7. Эта таблица содержит измерения регистра **ОстаткиМатериалов** – **Материал**, **Склад** и кроме этого начальные и конечные остатки, а также значения прихода, расхода и оборотов для всех ресурсов регистра **ОстаткиМатериалов**.

8. Начнем выбирать поля таблицы в нужном нам порядке двойным щелчком мыши. Сначала выберем **Склад** и **Материал**. Затем выберем **КоличествоНачальныйОстаток**, **КоличествоПриход**, **КоличествоРасход**. В заключение выберем **КоличествоКонечныйОстаток**.

Выделенные элементы можно перенести из одного списка в другой перетаскиванием мышью или двойным щелчком на них.

9. В результате окно **Поля** должно быть заполнено следующим образом (рис.1).

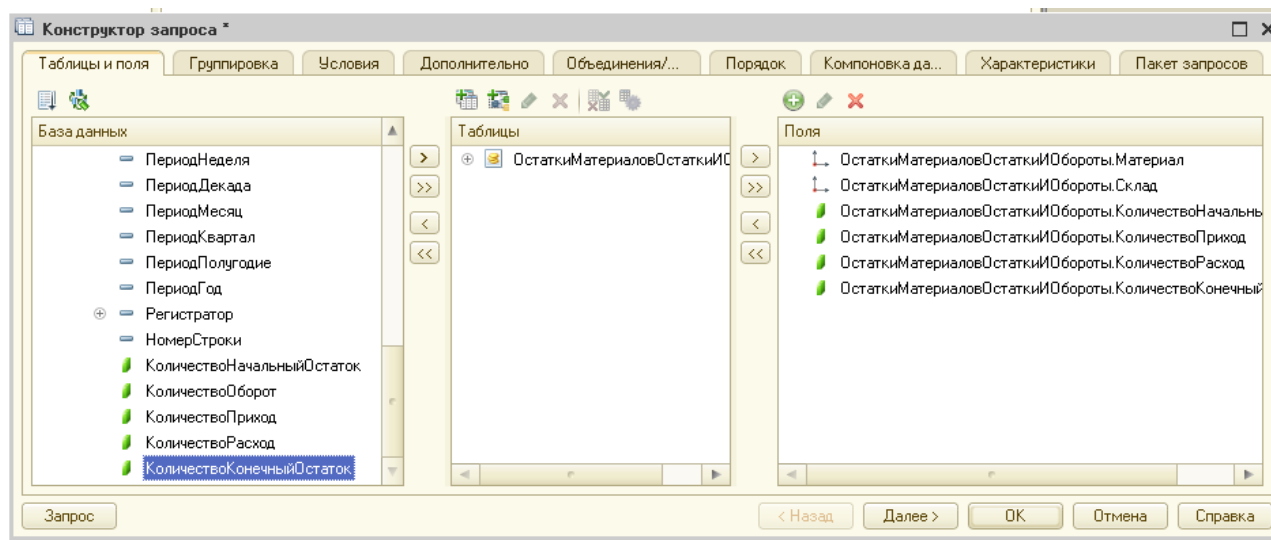
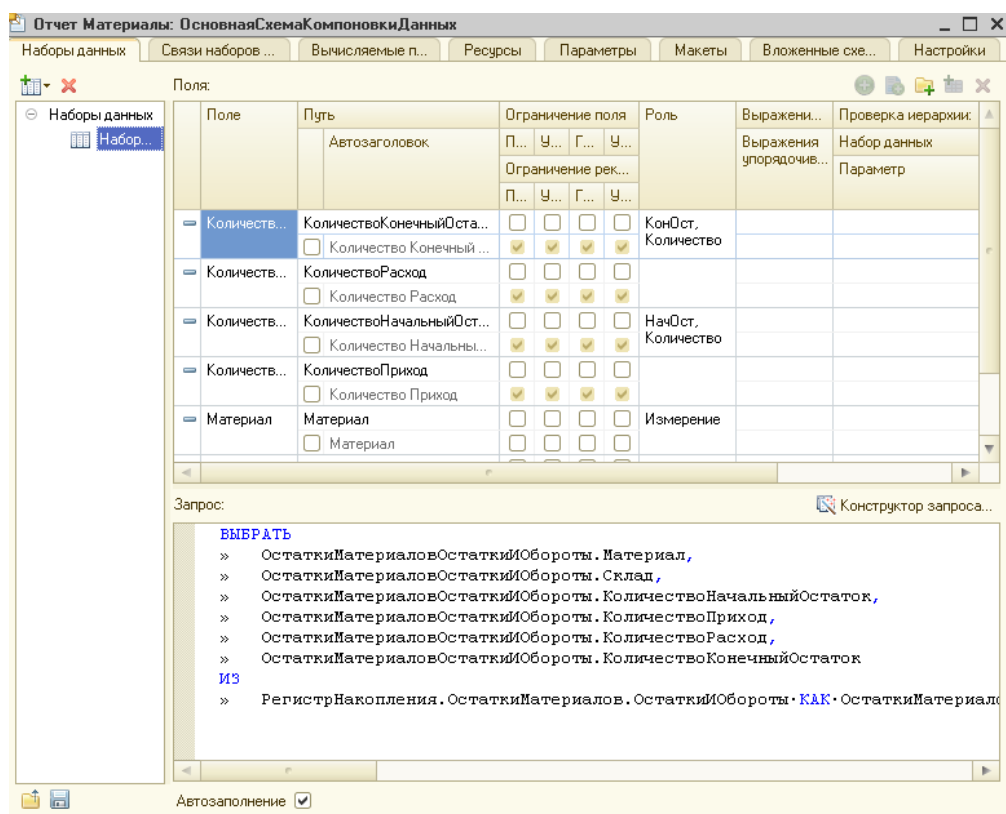


Рис. 1. Выбранные поля

10. Нажмем **ОК** и вернемся в конструктор схемы компоновки данных.



11. 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. Обратите 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 нашем случае система 1С:Предприятие заполнила данный список автоматически, из текста запроса, и нет необходимости в его ручной настройке. Итак, мы описали, каким образом будут извлекаться данные для отчета. Но пока мы не создадим стандартных настроек нашего отчета, мы ничего не увидим в результате. Поэтому создадим самые простые настройки отчета для отображения обычных детальных записей информационной базы. В нашем случае это будут записи виртуальной Таблицы регистр накопления **ОстаткиМатериалов**, выбранные в линейном порядке по мере попадания их в эту таблицу.

12. **Настройки отчета.** Перейдем на закладку **Настройки**. В верхнем правом окне будет находиться иерархическая структура нашего отчета. Для добавления нового элемента выделим в дереве структуры **отчет** корневой элемент **Отчет** и вызовем его контекстное меню. **Можно** также нажать кнопку **Добавить** в командной панели окна или нажать клавишу **1 ps**.

13. Добавим в отчет группировку (контекстное меню - **Новая группировка**). При этом не станем указывать поле группировки, а **просто** нажмем **ОК**. Тем самым мы определили, что в отчет будут выводиться *детальные записи* из информационной базы - записи, получаемые в результате выполнения запроса без итогов.

В структуре отчета появится группировка **Детальные записи**(рис. 2)

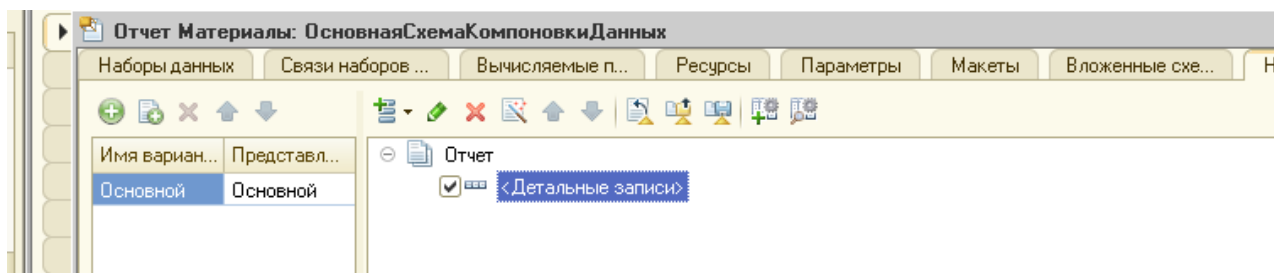


рис. 2

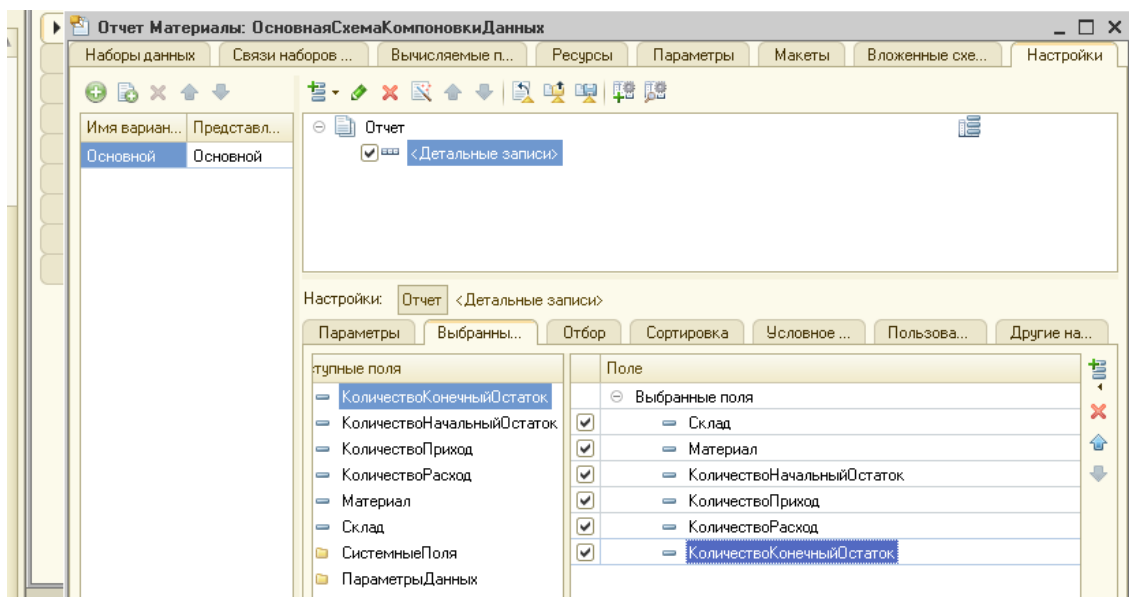
14. Теперь настроим поля, которые будут выводиться в результат отчета. Для этого перейдем в нижнем окне настроек на закладку **Выбранные поля** и перенесем мышью из списка доступных полей:

- Склад,
- Материал,
- КоличествоНачальныйОстаток,
- КоличествоПриход,
- КоличествоРасход,
- КоличествоКонечныйОстаток.

ПРИМЕЧАНИЕ

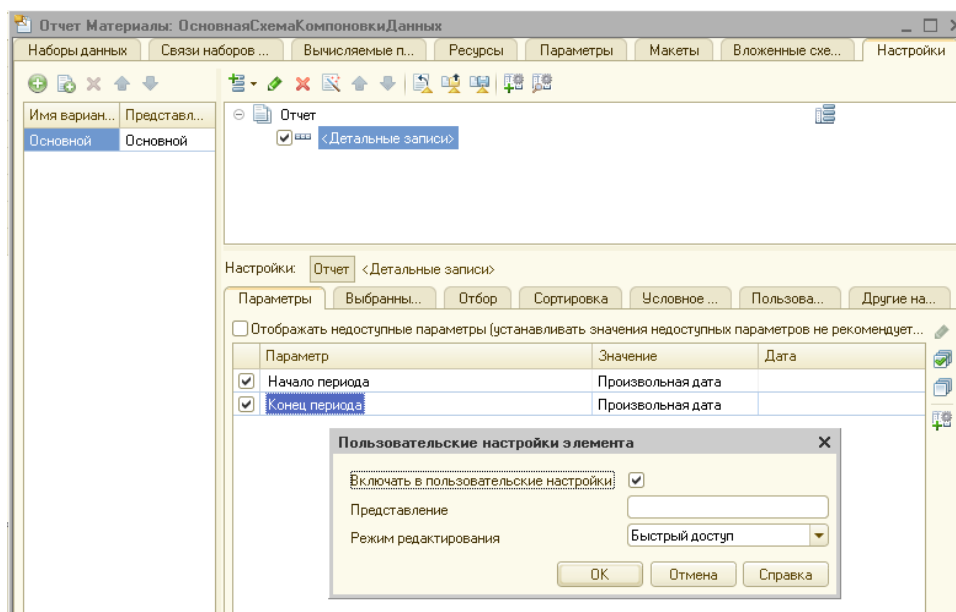
Добавление доступных полей в список выбранных полей можно осуществить перетаскиванием мышью, двойным щелчком на доступных полях либо нажатием кнопки **Добавить** справа от списка выбранных полей. Порядок выбранных полей можно изменить позже кнопками **Вверх**, **Вниз** или перетаскиванием мышью.

15. В результате окно настроек отчета должно иметь вид как на Рис.



16. Затем перейдем на закладку **Параметры** и укажем, что параметры отчета **Дата начала** и **Дата окончания** будут включены в состав пользовательских настроек, и эти настройки будут находиться непосредственно в форме отчета, то есть будут «быстрыми» настройками.

17. Сначала укажем, что оба эти параметра будут использоваться в отчете - установим флажки в первой колонке. Затем выделим каждый из параметров, нажмем кнопку **Свойство элемента пользовательских настроек** и поставим флажок **Вкл ~ в пользовательские настройки**.



18. Таким образом, перед формированием отчета пользователь сможет задать отчетный период. Определим в каких подсистемах будет отображаться наш отчет.

19. Закроем конструктор схемы компоновки данных и в окне редактирования объекта конфигурации Отчет **Материалы** перейдем на закладку Подсистемы. Отметим в списке подсистем конфигурации ветви **Учет материалов, Оказание услуг и Бухгалтерия**. Таким образом, ссылка на наш отчет автоматически попадет в панель действий этих подсистем.

20. В режиме **1С: Предприятие**. Запустим 1С:Предприятие в режиме отладки и посмотрим, как Работает отчет. В открывшемся окне 1С:Предприятия мы видим, что в панели действий разделов **Бухгалтерия, Оказание услуг и Учет материалов** появилась новая группа команд для выполнения отчетов и в ней команда для формирования отчета **Материалы**. Выполним ее.

21. Перед нами откроется автоматически сформированная системой форма отчета.

22. Зададим даты начала и окончания отчетного периода и нажмем на кнопку **Сформировать**. Как видите, наш отчет полностью отражает движение материалов, произошедшее в нашей организации.

Контрольные вопросы

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

Лабораторная работа № 7

Тема работы: Макет печатной формы.

Цель работы: Научиться создавать различные макеты печатной формы.

Ход выполнения работы.

Теоретическая часть

Что такое макет

Объект конфигурации *Макет* предназначен для хранения различных форм представления данных, различных 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. OLE-объекты, 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.

Макет печатной формы

Наша цель будет заключаться в создании печатной формы документа **Оказание услуги**.

1. Откроем в конфигураторе окно редактирования объекта конфигурации Документ **ОказаниеУслуги**.

2. Перейдем на закладку **Макеты** и запустим конструктор печати. В открывшемся окне на первом шаге укажем, что будет создана новая команда Печать для формирования печатной формы документа.

3. На втором шаге нажатием кнопки >>. определим, что все реквизиты нашего документа будут отображены в шапке печатной формы. Нажмем **Далее**.

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

5. На четвертом шаге конструктор предложит сформировать подвал (нижнюю часть) печатной формы. Мы не станем ничего указывать (подвал в данном случае использовать не будем), нажмем и перейдем к пятому шагу. Здесь ничего изменять не будем. Тем самым согласимся с тем, ко-

манда для вызова процедуры формирования печатной формы будет помещена в командную панель формы, в раздел Важное.

6. Нажмем ОК. В конфигураторе откроется модуль команды Печать, модуль менеджера документа **ОказаниеУслуги** и макет этого документа. Разработчик может создать макет печатной формы с нуля и для ее вывода создать соответствующую команду и кнопку в форме документа, но в данном случае всю работу сделал за нас конструктор печати:

- Создан макет печатной формы документа **ОказаниеУслуги** с именем **Печать**.

- Создана команда документа **ОказаниеУслуги** с именем **Печать**. В модуль этой команды помещен обработчик, вызывающий процедуру печати документа, выполняющуюся на сервере. Сама процедура печати помещена в модуль менеджера документа **ОказаниеУслуги**.

- В командную панель формы документа **ОказаниеУслуги** помещена команда **Печать** для формирования печатной формы документа.

7. Причем поскольку команда **Печать** принадлежит документу **ОказаниеУслуги** в целом, а не конкретной его форме, эту команду можно будет помещать в любую форму, созданную для документа.

8. В будущем мы будем самостоятельно создавать процедуры обработчиков команд и размещать соответствующие им кнопки в форме, «пока воспользуемся результатами работы конструктора печати и проверим макет в работе.

9. Запустим 1С: Предприятие в режиме отладки и откроем документ **ОказаниеУслуги № 1**. Обратите внимание, что в командной панели документа появилась кнопка **Печать**.

10. Нажмем на нее и увидим 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.

11. **Редактирование макета.** В режиме Конфигуратор прежде всего

добавим итоговую сумму в печатную форму документа **ОказаниеУслуги**. Откроем конфигуратор, раскроем дерево документа **Оказание Услуги** и дважды щелкнем на макете **Печать**.

12. Как видите, макет документа состоит из именованных областей, которые в определенном порядке выводятся на печать. Те именованные области, которые мы видим слева, были созданы с помощью конструктора. Но разработчик может сам создавать или удалять области, переименовывать их и т. п.

13. Добавим новую область для вывода итоговой суммы документа. Выделим мышью две пустые строки под табличной частью документа и выполним пункт главного меню **Таблица ► Имена ► Назначить имя**.

14. Назовем область **Всего**, нажмем **ОК**.

15. Чтобы формат добавленных нами строк совпадал с имеющимся форматом заголовка и табличной части документа, изменим ширину колонок. Для этого потянем мышью в заголовке таблицы за правую границу колонки 2 так, чтобы ее ширина совпала с шириной колонки № в шапке документа. Отпустим мышшь. Платформа предложит создать новый формат для выделенных строк. Согласиться с этим. Аналогичные действия выполним и для колонок 3, 4, 5 и 6.

16. В созданной области, в колонке **Цена**, напишем **ВСЕГО**, а в колонке **Сумма** напишем **ВсегоПоДокументу**. Вызвав палитру свойств для последней заполненной нами ячейки (контекстное меню - **Свойства**), в свойстве **Заполнение** укажем, что в этой ячейке будет находиться не текст, а параметр.

17. Каждая ячейка редактируемого нами 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 нам значение при формировании печатной формы.

18. Откроем модуль менеджера документа **ОказаниеУслуги**. Для этого перейдем на закладку **Прочее** окна редактирования объекта конфигурации **Документ ОказаниеУслуги** и нажмем кнопку **Модуль менеджера**.

Найдем в нем процедуру **Печать** и отредактируем ее следующим образом (новые строки выделены жирным шрифтом), листинг- 1.

Листинг – 1. Печать формы документа (фрагмент)

```
ОбластьЗаголовков = Макет.ПолучитьОбласть("Заголовков");
    Шапка = Макет.ПолучитьОбласть("Шапка");
    ОбластьПереченьНоменклатурыШапка=
Макет.ПолучитьОбласть("ПереченьНоменклатурыШапка");
    ОбластьПереченьНоменклатуры = Макет.ПолучитьОбласть("Пе-
реченьНоменклатуры");
    ОбластьИтог=Макет.ПолучитьОбласть("Всего");
    ТабДок.Очистить();

    ВставлятьРазделительСтраниц = Ложь;
    Пока Выборка.Следующий() Цикл
        Если ВставлятьРазделительСтраниц Тогда
            ТабДок.ВывестиГоризонтальныйРазделительСтра-
ниц();
        КонецЕсли;

    ТабДок.Вывести(ОбластьЗаголовков);

    Шапка.Параметры.Заполнить(Выборка);
    ТабДок.Вывести(Шапка, Выборка.Уровень());
```

```

ТабДок.Вывести(ОбластьПереченьНоменклатурыШапка);
ВыборкаПереченьНоменклатуры = Выборка.Перечень-
Номенклатуры.Выбрать());
СуммаИтог=0;
Пока ВыборкаПереченьНоменклатуры.Следующий() Цикл
    ОбластьПереченьНоменклатуры.Параметры.Запол-
нить(ВыборкаПереченьНоменклатуры);
    ТабДок.Вывести(ОбластьПереченьНоменклатуры,
ВыборкаПереченьНоменклатуры.Уровень());
    СуммаИтог=СуммаИтог+ВыборкаПеречень-
Номенклатуры.Сумма;
    КонецЦикла;
ОбластьИтог.Параметры.ВсегоПоДокументу=СуммаИ-
тог;
ТабДок.Вывести(ОбластьИтог);

ВставлятьРазделительСтраниц = Истина;
КонецЦикла;

```

19. Смысл добавленного фрагмента прост. Мы обращаемся к макету **дата ОказаниеУслуги** по его имени - **Макет**. Используя его метод **ПолучитьОбласть()**, получаем область **Всего** (ту, которую мы только что добавили к макету) и сохраняем ее в переменной **ОбластьИтог**.

20. В цикле обхода строк табличной части документа, полученных в результате выполнения запроса, мы накапливаем в переменной **СуммаИтог** значение суммы табличной части документа по колонке **Сумма**.

21. Затем мы обращаемся к параметру **ВсегоПоДокументу** (**ОбластьИтог. Параметры. ВсегоПоДокументу**), находящемуся в области **Всего**, и присваиваем ему значение переменной **СуммаИтог**.

22. В заключение мы выводим итоговую область в табличный документ, который будет показан на экране и распечатан пользователем - **ТабДок.Вывести(ОбластьИтог)**.

23. Отображение табличного документа на экране выполняется в обработчике команды **Печать**, в модуле этой команды на клиенте, в то время как сама процедура печати, описанная в модуле менеджера документа, выполняется на сервере.

24. Запустим 1С:Предприятие в режиме отладки и проверим результат наших изменений. Подобным образом, создавая именованные области и ячейки макета, используя их свойства и управляя порядком их вывода с помощью встроенного языка, разработчик имеет возможность создать печатную форму любого дизайна. А теперь, 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.

25. **Редактирование формы.** После того как мы вывели итоговую сумму по табличной части в печатную форму документа, мы можем видеть такую же итоговую сумму и в форме документа. Чтобы в процессе его создания можно было оперативно, не печатая документ, знать итоговую сумму по документу. Для этого мы внесем небольшие изменения в форму документа **Оказание Услуги**.

26. Для редактирования формы документа откроем конфигуратор, раскроем дерево документа **Оказание Услуги** и дважды щелкнем на форме **ФормаДокумента**.

27. Дважды щелкнем на элементе **ПереченьНоменклатуры** в дереве, элементов формы или правой кнопкой мыши откроем для него палитру свойств (пункт контекстного меню **Свойства**).

28. Установим свойство **Подвал**, которое определяет наличие подвала у таблицы формы.

29. Затем откроем свойства элемента формы **ПереченьНоменклатурыЦена** и установим:

- **Текст подвала - Всего:** .
- **Горизонтальное положение в подвале - Право**, а в свойстве **Шрифт подвала** изменим начертание на **Жирный**. После этого откроем свойства элемента **ПереченьНоменклатурыСумма**, установим: **Горизонтальное положение в подвале – Право**. В свойстве **Шрифт подвала** тоже изменим начертание на **Жирный**.

30. Для того чтобы в подвале колонки **Сумма** отображался итог по ней, нажмем кнопку выбора в поле **ПутьКДаннымПодвала**.

31. Раскроем дерево реквизитов объекта и выберем элемент **Итог-Сумма**. Мы видим, что по колонке **Сумма** в табличной части документу подсчитывается общий итог документа.

32. Это небольшое изменение, которое мы сделали в форме, на самом деле сильно улучшает ее пользовательский интерфейс и делает ее более удобной в применении. Подобным образом, используя свойства элементов формы, задавая их значения и изменяя командный интерфейс формы, разработчик имеет возможность создать экранную форму нужного дизайна и поведения.

. . .

Контрольные вопросы

1. Для чего предназначен объект конфигурации **Макет**.
2. Что такое конструктор печати.
3. Как создать макет с помощью конструктора печати.
4. Как изменить табличный документ.
5. Какая разница в заполнении ячейки табличного документа текстом, параметром и шаблоном.
6. Как с помощью встроенного языка вывести в табличный документ новую область.
7. Как изменить внешний вид и поведение элемента формы.

8. Как отобразить сумму по колонке таблицы

Лабораторная работа № 8

Тема: Отчеты. Вывод данных в таблицу

Цель занятия: Научиться создавать отчеты. Вывод данных в таблицу

Теоретическая часть

Виртуальные таблицы запросов

При создании запроса платформа предоставляет нам в качестве источников данных некоторое количество виртуальных таблиц. Название «виртуальные» полностью соответствует их сути, поскольку эти таблицы, в свою очередь, также являются результатом запроса, который система формирует в момент выполнения соответствующего участка кода. По большому счету разработчик может самостоятельно получить те же самые данные, которые система предоставляет ему в качестве виртуальных таблиц, однако алгоритм получения этих данных не будет оптимизирован в силу следующих двух причин. Во-первых, все виртуальные таблицы параметризованы, то есть разработчику предоставляется возможность задать некоторые параметры, которые система будет использовать при формировании запроса создания виртуальной таблицы.

Примечательным здесь является то, что задание параметров виртуальной таблицы далеко не всегда приводит к простой подстановке указанных разработчиком значений в текст запроса. В зависимости от того, какие параметры виртуальной таблицы указаны разработчиком, система может формировать РАЗЛИЧНЫЕ запросы для получения одной и той же виртуальной таблицы, причем они будут оптимизированы с точки зрения переданных параметров. Во-вторых, не всегда разработчик имеет возможность получить доступ к тем данным, к которым имеет доступ система. Например, при использовании виртуальных таблиц регистров сведений разработчику доступна по боль-

шому счету вся та же информация о данных регистров, которую использует система при формировании запроса виртуальной таблицы. Совсем иная картина с виртуальными таблицами регистров накопления. Здесь система динамически формирует запрос в зависимости не только от переданных параметров, но и от периода рассчитанных итогов регистра, причем в запросе она использует данные рассчитанных итогов, которые просто недоступны для разработчика при создании запроса. Конечно, разработчик может самостоятельно перебрать все записи регистра накопления и в итоге получить те же самые данные, которые система предоставляет в виде виртуальной таблицы. Однако очевидно, что такой запрос будет менее эффективным и потребует от разработчика гораздо больше трудозатрат.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Вывод данных в таблицу

Это наиболее гибкий и универсальный отчет, который позволяет пользователю изменять его структуру и внешний вид самостоятельно.

В режиме Конфигуратор

Добавим новый объект конфигурации *Отчет*. Назовем его *Универсальный* и запустим конструктор схемы компоновки данных. Создадим новый Набор данных - запрос и вызовем конструктор запроса.

Запрос для набора данных

В качестве источника данных для запроса выберем виртуальную таблицу регистра накопления Продажи.Обороты. Затем выберем из нее все поля:

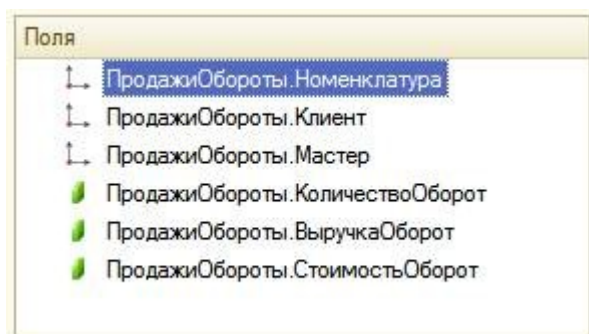


Рис.. Выбранные поля

- Нажмем *ОК*–

- На закладке *Ресурсы* нажатием кнопки » выберем все доступные ресурсы отчета.

- На закладке *Настройки*: нажмем кнопку *Добавить* в командной панели окна настроек и добавим таблицу.

Мы не будем здесь задавать строки и колонки этой таблицы, а также список выбранных полей, так как хотим предоставить полную свободу пользователю в этих действиях.

Для этого выделим в структуре элементов отчета элемент *Таблица* и нажмем кнопку *Свойства* элемента пользовательских настроек, расположенную сверху в командной панели окна настроек.

В появившемся окне сделаем следующие изменения:

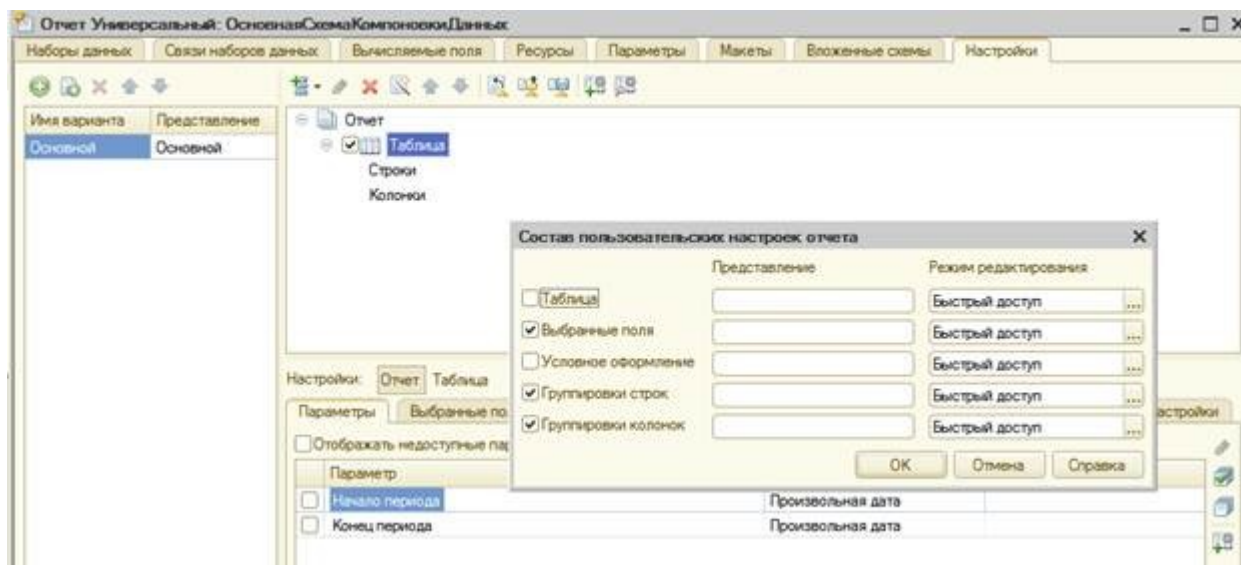


Рис. . Состав пользовательских настроек

В заключение определим, в каких подсистемах будет отображаться наш отчет.

Закроем конструктор схемы компоновки данных и в окне редактирования объекта *Универсальный* перейдем на закладку *Подсистемы*. Отметим в списке подсистем конфигурации подсистему *Оказание услуг*.

В режиме 1С: Предприятие

Запустим 1С: Предприятие в режиме отладки и выполним команду Универсальный в панели действий раздела ОказаниеУслуг. Если мы сейчас нажмем Сформировать, то ничего не увидим в результате, так список выбранных полей, группировок строк и колонок таблицы пуст. Пользователь имеет возможность самостоятельно формировать требуемый отчет по регистру Продажи.

Контрольные вопросы

- Как создать пользовательские настройки отчета.
- В чем отличие «быстрых» настроек от остальных пользовательских настроек.
- Как определить состав пользовательских настроек отчета.
- Как вывести данные в виде таблицы.
- Как сделать отчет универсальным.

Лабораторная работа № 17

Тема: Отчеты. Использование вычисляемого поля в отчете

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Система компоновки данных

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

Исходные данные для компоновки отчета содержит в себе схема компоновки данных.

Разработчик создает схему компоновки данных, в которой описывает текст запроса, наборы данных, связи между ними, доступные поля, параметры получения данных, и задает первоначальные настройки компоновки - структуру отчета, макет оформления данных и др. Отчет системы компоновки имеет сложную иерархическую структуру и может состоять из различных элементов, таких как группировки, таблицы и диаграммы.

При этом пользователь может изменить существующую структуру отчета или вообще создать совершенно новую структуру отчета. Может настроить необходимый ему отбор, оформление элементов структуры отчета, получить расшифровку по каждому элементу и т. д. Например, может быть задана такая структура отчета, состоящая из одной таблицы и одной диаграммы.

В этом случае сформированный отчет будет иметь следующий вид. В представленном отчете таблица будет состоять из записей регистра накопления ПродажиОборота о клиентах и оказанных им услугах. Эти записи сгруппированы по мастерам, которые выполняли заказы. А в группировке будет выведен список услуг, оказанных данным мастером, и затраченных на это материалов.

Как мы уже говорили в начале раздела, система компоновки данных представляет собой совокупность нескольких объектов. При формировании и исполнении отчета происходит последовательная передача данных от одного объекта системы компоновки данных к другому, до получения конечного результата - документа, показанного пользователю.

Алгоритм взаимодействия этих объектов выглядит следующим образом:

Разработчик создает схему компоновки данных и настройки по умолчанию. В общем случае на основе одной схемы компоновки данных может быть создано большое количество различных отчетов. Настройки компоновки данных, создаваемые разработчиком или изменяемые пользователем, определяют, какой именно отчет будет получен в конкретном случае.

- На основе схемы компоновки и имеющихся настроек компоновщик макета создает макет. Это этап подготовки к исполнению отчета. Макет компоновки данных является уже готовым заданием для выполнения процессором компоновки. Он содержит необходимые запросы, макеты областей отчета и др.

- Процессор компоновки данных выбирает данные из информационной базы согласно макету компоновки, агрегирует и оформляет эти данные.

Результат компоновки обрабатывается процессором вывода, и в итоге пользователь получает результирующий табличный документ.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Использование вычисляемого поля в отчете

Следующий отчет - Рейтинг клиентов - будет показывать в графическом виде, каков доход от оказания услуг каждому из клиентов за все время работы предприятия.

- **В режиме Конфигуратор**

Добавим новый объект конфигурации Отчет. Назовем его Рейтинг-Клиентов и запустим конструктор схемы компоновки данных. Создадим новый Набор данных - запрос и вызовем конструктор запроса.

Запрос для набора данных

В качестве источника данных для запроса выберем виртуальную таблицу регистра накопления Продажи.Обороты. Затем выберем из нее следующие поля:

- ПродажиОбороты. Клиент,
- ПродажиОбороты.ВыручкаОборот,
- ПродажиОбороты.СтоимостьОборот.

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

- *Нажмем ОК*–

Перейдем к редактированию схемы компоновки данных.

Вычисляемые поля

На этом этапе мы столкнулись с необходимостью отразить в отчете поле, которого нет в наборе данных. Раньше мы использовали в отчете те поля, которые описывались в наборе данных. Теперь, чтобы отобразить доход от оказания услуг в разрезе клиентов, нам необходимо дополнительное поле, рассчитанное как разница между выручкой и стоимостью оказания услуг.

Для этого в системе компоновки данных есть возможность определения вычисляемого поля.

Вычисляемые поля: дополнительные поля схемы компоновки данных, значения которых будут вычисляться по некоторой формуле.

- На закладке *Вычисляемые поля* добавим вычисляемое поле, нажав кнопку *Добавить*. Дадим ему имя - *Доход*, в колонку *Выражение* введем следующий код:

Листинг 13.2. Выражение для расчета вычисляемого поля «Доход»

Выручка – Стоимость

Ресурсы

- На закладке *Ресурсы* нажмем кнопки *»* выберем все доступные ресурсы отчета:



Рис.. Ресурсы схемы компоновки данных

Настройки

- На закладке *Настройки* добавим в структуру отчета диаграмму. Для этого нажмем кнопку *Добавить* в командной панели окна настроек и добавим диаграмму:

Затем выделим ветку *Точки* и добавим в нее группировку по полю *Клиент*.

Серии диаграммы оставим без изменений.

Дело в том, что для демонстрации рейтинга клиентов хорошо подойдет

круговая диаграмма, которую мы хотим показать. Для этого вида диаграммы достаточно задать только точки, поэтому серии мы не задаем.

- На закладке *Выбранные поля* выберем поле Доход для вывода в отчет.

Структура отчета должна принять следующий вид:

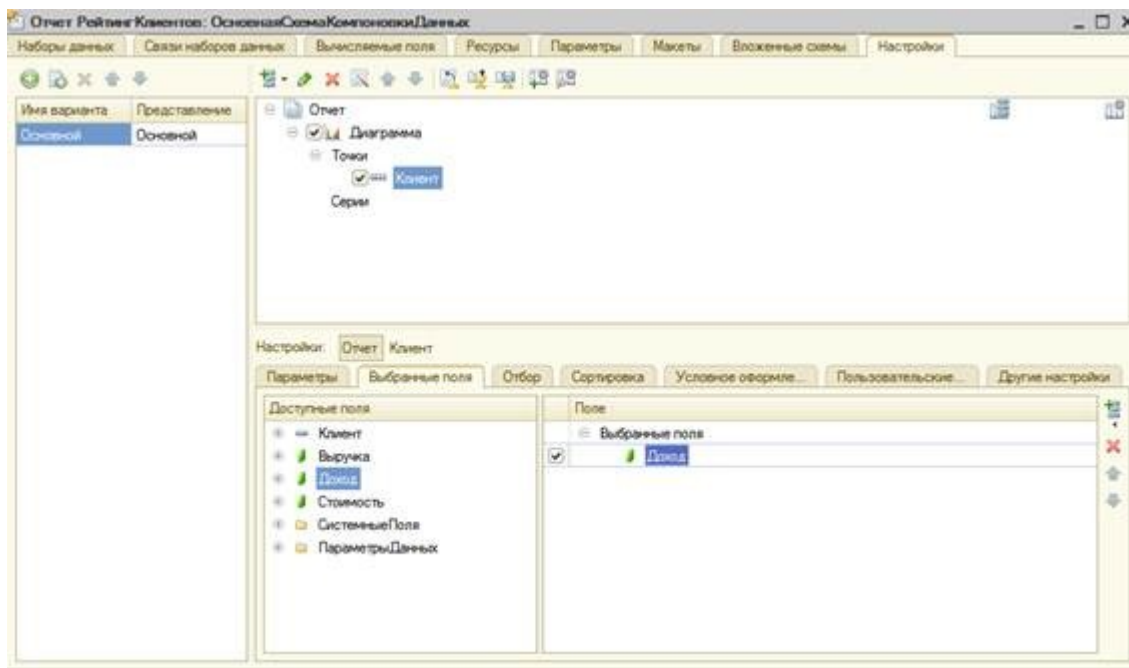


Рис. Структура отчета и настройки диаграммы

- На закладке *Другие настройки* выберем тип диаграммы *Круговая объемная* и зададим заголовок отчета - *Рейтинг клиентов*. После этого в окне редактирования объекта *РейтингКлиентов* перейдем на закладку *Подсистемы*. Отметим в списке подсистем конфигурации подсистемы *Оказание услуг* и *Бухгалтерия*.

- **В режиме 1С: Предприятие**

Запустим 1С:Предприятие в режиме отладки и выполним команду *Рейтинг клиентов* в панели действий раздела *Бухгалтерия*. Нажмем *Сформировать*.

Мы видим данные о доходе от оказания услуг по каждому из клиентов, представленные в виде круговой диаграммы. Если вернуться в конфигуратор и изменить тип диаграммы на *Гистограмма объемная*, то наша диаграмма поменяет вид.

Контрольные вопросы

- Как создать отчет, содержащий диаграмму.
- Как использовать параметры в системе компоновки данных.
- Что такое ресурсы в системе компоновки данных.
- Что такое вычисляемые поля в системе компоновки данных.
- Как дополнить данные отчета всеми датами в группировке по периоду.

Лабораторная работа № 10

Тема: Отчеты. Вывод данных в таблицу

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Виртуальные таблицы запросов

При создании запроса платформа предоставляет нам в качестве источников данных некоторое количество виртуальных таблиц. Название «виртуальные» полностью соответствует их сути, поскольку эти таблицы, в свою очередь, также являются результатом запроса, который система формирует в момент выполнения соответствующего участка кода. По большому счету разработчик может самостоятельно получить те же самые данные, которые система предоставляет ему в качестве виртуальных таблиц, однако алгоритм получения этих данных не будет оптимизирован в силу следующих двух причин. Во-первых, все виртуальные таблицы параметризованы, то есть разработчику предоставляется возможность задать некоторые параметры, которые система будет использовать при формировании запроса создания виртуальной таблицы.

Примечательным здесь является то, что задание параметров виртуальной таб-

лицы далеко не всегда приводит к простой подстановке указанных разработчиком значений в текст запроса. В зависимости от того, какие параметры виртуальной таблицы указаны разработчиком, система может формировать РАЗЛИЧНЫЕ запросы для получения одной и той же виртуальной таблицы, причем они будут оптимизированы с точки зрения переданных параметров. Во-вторых, не всегда разработчик имеет возможность получить доступ к тем данным, к которым имеет доступ система. Например, при использовании виртуальных таблиц регистров сведений разработчику доступна по большому счету вся та же информация о данных регистров, которую использует система при формировании запроса виртуальной таблицы. Совсем иная картина с виртуальными таблицами регистров накопления. Здесь система динамически формирует запрос в зависимости не только от переданных параметров, но и от периода рассчитанных итогов регистра, причем в запросе она использует данные рассчитанных итогов, которые просто недоступны для разработчика при создании запроса. Конечно, разработчик может самостоятельно перебрать все записи регистра накопления и в итоге получить те же самые данные, которые система предоставляет в виде виртуальной таблицы. Однако очевидно, что такой запрос будет менее эффективным и потребует от разработчика гораздо больше затрат.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Вывод данных в таблицу

Это наиболее гибкий и универсальный отчет, который позволяет пользователю изменять его структуру и внешний вид самостоятельно.

- **В режиме Конфигуратор**

Добавим новый объект конфигурации *Отчет*. Назовем его *Универсальный* и запустим конструктор схемы компоновки данных. Создадим новый Набор данных - запрос и вызовем конструктор запроса.

Запрос для набора данных

В качестве источника данных для запроса выберем виртуальную таблицу регистра накопления Продажи.Обороты. Затем выберем из нее все поля:

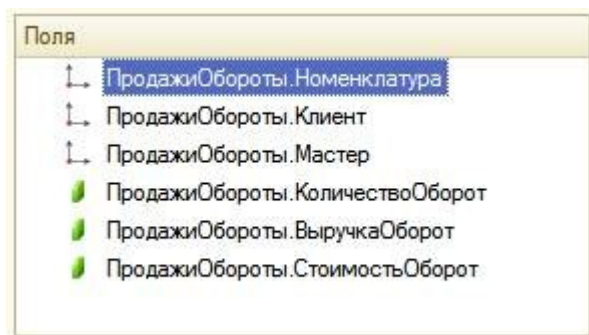


Рис. Выбранные поля

- Нажмем *ОК*–

- На закладке *Ресурсы* нажатию кнопки » выберем все доступные ресурсы отчета.

- На закладке *Настройки*: нажмем кнопку *Добавить* в командной панели окна настроек и добавим таблицу.

Мы не будем здесь задавать строки и колонки этой таблицы, а также список выбранных полей, так как хотим предоставить полную свободу пользователю в этих действиях.

Для этого выделим в структуре элементов отчета элемент *Таблица* и нажмем кнопку *Свойства элемента пользовательских настроек*, расположенную вверху в командной панели окна настроек.

В появившемся окне сделаем следующие изменения:

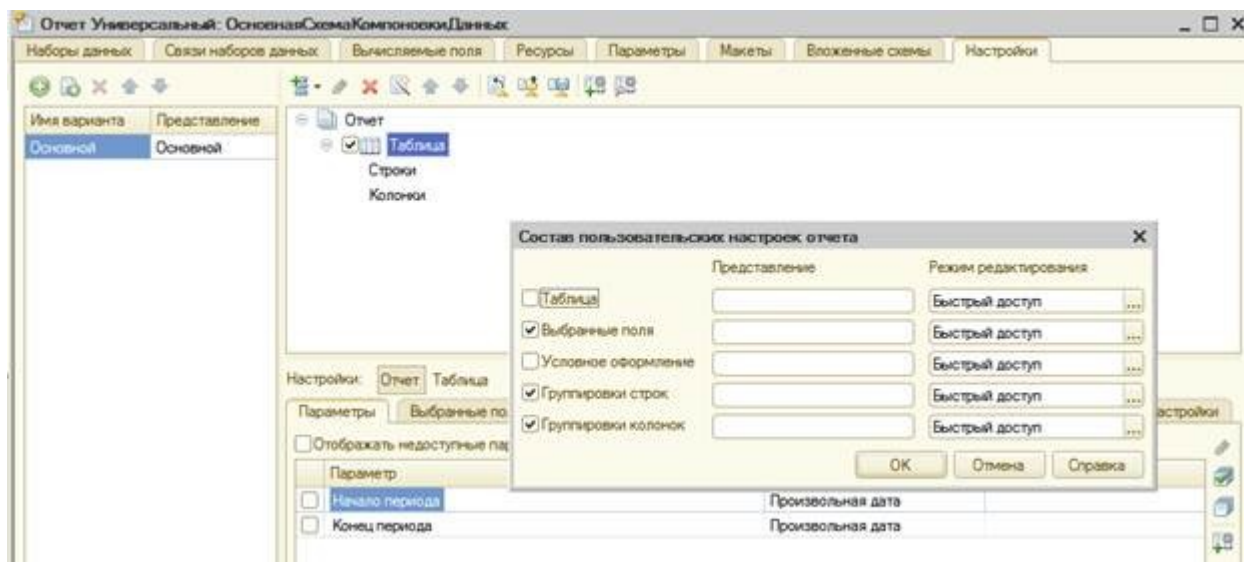


Рис. . Состав пользовательских настроек

В заключение определим, в каких подсистемах будет отображаться наш отчет.

Закроем конструктор схемы компоновки данных и в окне редактирования объекта Универсальный перейдем на закладку Подсистемы. Отметим в списке подсистем конфигурации подсистему Оказание услуг.

- **В режиме 1С: Предприятие**

Запустим 1С: Предприятие в режиме отладки и выполним команду Универсальный в панели действий раздела ОказаниеУслуг. Если мы сейчас нажмем Сформировать, то ничего не увидим в результате, так список выбранных полей, группировок строк и колонок таблицы пуст. Пользователь имеет возможность самостоятельно формировать требуемый отчет по регистру Продажи.

Контрольные вопросы

- Как создать пользовательские настройки отчета.
- В чем отличие «быстрых» настроек от остальных пользовательских настроек.
- Как определить состав пользовательских настроек отчета.
- Как вывести данные в виде таблицы.
- Как сделать отчет универсальным.

Список рекомендуемой литературы и источников

- 1 Гладких Т. В. Разработка прикладных решений для информационной системы 1С: Предприятие 8.2: Учебное пособие/Гладких Т. В..-Воронеж: Воронежский государственный университет инженерных технологий, 2016, ISBN 978-5-00032-182-9.-56.
- 2 Заика А. А. Разработка прикладных решений для платформы 1С. Предприятие 8.2 в режиме «Управляемое приложение»: учебное пособие/Заика А. А..-Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.-238.
- 3 Основы конфигурирования в системе «1С. Предприятие 8.0»: учебное пособие.-Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.-222.
- 4 Заика А. А. Практика бухгалтерского учета в 1С: Бухгалтерии 8: учебное пособие/Заика А. А..-Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.-526.
- 5 Сайт 1С.ru