Федеральное государственное бюджетное образовательное учреждение высшего образования «КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

М. В.Соловей

АРХИТЕКТУРА И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ МАЛЫХ И СРЕДНИХ ПРЕДПРИЯТИЙ

Учебно-методическое пособие по выполнению лабораторных работ для студентов направления подготовки 09.03.03 Прикладная информатика

Калининград Издательство ФГБОУ ВО «КГТУ» 2022

Рецензент:

кандидат педагогических наук, доцент кафедры прикладной информатики ФГБОУ ВО «Калининградский государственный технический университет» Е. Ю. Заболотнова

Соловей, М. В.

Архитектура и разработка информационных систем малых и средних предприятий: учебно-методическое пособие по выполнению лабораторных работ для студентов направления подготовки 09.03.03 Прикладная информатика / М. В. Соловей.– Калининград: Изд-во ФГБОУ ВО «КГТУ», 2022. – 107 с.

Данное учебно-методическое пособие содержит лабораторные работы по дисциплине: задания, методические указания по выполнению работ, структуру отчета и требования к его оформлению, приведены контрольные вопросы и порядок защиты лабораторных работ.

Учебно-методическое пособие рассмотрено и одобрено в качестве локального электронного методического материала кафедрой прикладной информатики института цифровых технологий ФГБОУ ВО «Калининградский государственный технический университет» 19 сентября 2022 г., протокол № 3

Учебно-методическое пособие рекомендовано к использованию в качестве локального электронного методического материала в учебном процессе методической комиссией ИЦТ 20 сентября 2022 г., протокол № 6

УДК 004.2(075)

© Федеральное государственное бюджетное образовательное учреждение высшего образования «Калининградский государственный технический университет», 2022 г. © Соловей М. В., 2022 г.

оглавление

введение	4
ЛАБОРАТОРНАЯ РАБОТА №1.	5
СОЗДАНИЕ ИНФОРМАЦИОННОЙ БАЗЫ. РАБОТА С КОНСТАНТА КЛИЕНТ-СЕРВЕРНОГО ПРОГРАММИРОВАНИЯ	МИ. ОСНОВЫ 5
ЛАБОРАТОРНАЯ РАБОТА №2	
РАБОТА СО СПРАВОЧНИКАМИ	
ЛАБОРАТОРНАЯ РАБОТА № 3	53
О РАЗЛИЧНЫХ ВИДАХ СПРАВОЧНИКОВ	53
ЛАБОРАТОРНАЯ РАБОТА №4	69
РАБОТА С ОТЧЕТАМИ	69
ЛАБОРАТОРНАЯ РАБОТА № 5	81
ДОКУМЕНТЫ И РЕГИСТРЫ	81
ЛИТЕРАТУРА	

введение

Данное учебно-методическое пособие предназначено для студентов направления подготовки **9.03.03** Прикладная информатика, изучающих дисциплину «Архитектура и разработка информационных систем малых и средних предприятий». Материалы для лабораторных работ взяты из источника: Официальный сайт интернет-университета «Интуит» (электронный ресурс), режим доступа

http://www.intuit.ru/studies/courses/2318/618/lecture/17939.

Цель лабораторного практикума по дисциплине: изучить принципы проектирования ИС на платформе ««1С:Предприятие»».

Лабораторный практикум содержит пять лабораторных работ, предназначенных для студентов очного и заочного форм обучения.

Лабораторные работы проводятся в лабораториях 143, 256, 353 и других, где установлена платформа ««1С:Предприятие»»».

В результате выполнения лабораторных работ ожидается, что студенты будут:

знать: роль и место информационных систем малых и средних предприятий в экономике; виды и классификацию автоматизированных информационных систем; историю развития, закономерности построения и функционирования АИС; состав, структуру и архитектуры АИС (функциональные подсистемы АИС, обеспечивающие подсистемы АИС, элементы и средства АИС); методологию и технологии обследования и разработки различных типов АИС и отдельных видов обеспечения, основные этапы жизненного цикла АИС и их особенности, стандартные этапы проектирования АИС, а также технологию и методологию внедрения АИС; методологические основы создания АИС в управлении предприятием (системный, информационный, стратегический и объектно-ориентированный подходы, разработка информационной модели системы предприятием); современные средства информационных управления И коммуникационных технологий обеспечения управленческой деятельности; основы документирования проектных решений по созданию АИС;

<u>уметь</u>: обосновывать необходимость и целесообразность автоматизации ИС; составлять техническое задание на создание АИС; выбирать инструментальные средства создания АИС; осуществлять конфигурирование типовых проектных решений по созданию АИС и выполнять их адаптацию к конкретным условиям применения; документировать проектные решения по созданию АИС, готовить организационнораспорядительную документацию стадии ввода АИС в действие; осуществлять мероприятия по вводу АИС в действие;

<u>владеть:</u> способностью составлять техническую документацию проектов информационных систем различного масштаба.

ЛАБОРАТОРНАЯ РАБОТА №1

СОЗДАНИЕ ИНФОРМАЦИОННОЙ БАЗЫ. РАБОТА С КОНСТАНТАМИ. ОСНОВЫ КЛИЕНТ-СЕРВЕРНОГО ПРОГРАММИРОВАНИЯ

ОБЩИЕ СВЕДЕНИЯ

Цель: научиться информационную базу на платформе ««1С:Предприятие»»

Материалы, оборудование, программное обеспечение: 1. персональный компьютер (компьютерные классы ГУК), 2. программное обеспечение «1С:Предприятие».

Условия допуска к выполнению: умение работать на ПК и знание техники безопасности

Критерии положительной оценки: предоставление результатов работы в виде файла и прохождение защиты.

Планируемое время выполнения: Аудиторное время выполнения (под руководством преподавателя): 4 ч. Время самостоятельной подготовки: 2 ч.

Теоретическое введение

Создание новой информационной базы

Откроем окно запуска «1С:Предприятие», создадим новую пустую информационную базу. Для этого нажмем на кнопку **Добавить**, в появившемся окне выберем Создание новой *информационной базы*, в следующем окне, рисунок 1.1, выберем вариант создания*информационной базы* без конфигурации.

	юй базы из шаблона	
Выберите поставляемую демонстрационный прим	о конфигурацию для начала раб иер для ознакомления	оты или
		A
		· •
		*

Рисунок 1.1. Создание информационной базы без конфигурации

Дадим информационной базе имя Салон, управляемое приложение, зададим в качестве папки информационной базыС:\Salon1, остальные параметры оставим по умолчанию.

После того, как база будет создана, откроем ее в Конфигураторе и, для того, чтобы открыть *дерево конфигурации*, выполним команду Конфигурация > Открыть конфигурацию. Вызовем контекстное *меню корневого элемента Конфигурация*, выберем в нем *пункт* Свойства, рисунок 1.2.



Рисунок 1.2. Свойства новой информационной базы

Обратите внимание на то, что свойство **Основной режим запуска** установлено в *значение* **Управляемое приложение**, в нижней части окна свойств расположено свойство **Режим совместимости**, которое установлено в *значение* **Не использовать**. В данном случае оно может принимать значения Версия 8.1. и Версия 8.2.13.

В качестве имени конфигурации введем СалонКрасоты, *поле* Синоним будет автоматически заполнено текстом Салон красоты.

Можно заметить, что изменились изображения интерфейсных элементов в **Конфигураторе**. Все говорит нам о том, что сейчас мы занимаемся разработкой конфигурации в режиме управляемого приложения. Среди нововведений платформы 8.2. можно отметить изменение состава объектов конфигурации. В частности, появились следующие новые объекты:

Общие реквизиты: здесь содержатся реквизиты, которые могут использоваться во многих объектах конфигурации. Например, если вы планируете добавить в документы своей конфигурации одинаковый *реквизит*, содержащий наименование организации, от имени которой составлен документ, это вполне логично реализовать с помощью общего реквизита. Кроме того, общие реквизиты используются в механизме разделения данных.

Функциональные опции: их используют для того, чтобы описывать возможности, которые можно включать и отключать в процессе эксплуатации системы. Функциональные опции могут влиять на командный *интерфейс*, например, скрывая или

отображая некоторые группы команд, а так же – на алгоритмы, написанные на встроенном языке.

Параметры функциональных опций: Содержит параметры, влияющие на функциональные опции

Хранилища настроек: Используется для сохранения и загрузки настроек.

Общие команды: Позволяет создавать команды, которые можно использовать в других объектах конфигурации, вызывая их, например, с помощью кнопок на формах.

Группы команд: Позволяет создавать группы для объединения команд

Элементы стиля: Позволяет создавать элементы стиля, такие, как цвет, *шрифт*, рамка, для организации единообразного оформления других объектов.

Внешние источники данных: эти объекты используются для получения информации из внешних источников и последующего использования ее в системе, в частности, в качестве источников данных для запросов, в качестве типов реквизитовинформационной базы и так далее.

Теперь в дереве конфигурации нет следующих объектов:

Интерфейсы

Стили

Кроме того, некоторые объекты, в частности, это касается подсистем, теперь используются по-другому. Начнем разработку обновленной конфигурации с создания подсистем.

Подсистемы – основа командного интерфейса управляемого приложения

Ранее, когда понятия командного интерфейса не существовало, подсистемы при разработке *прикладных решений* для «1С:Предприятие», играли вспомогательную роль. В сущности, они были нужны лишь разработчику, помогая ему структурировать конфигурацию для своих целей и автоматизировать некоторые *операции*. С приходом командного интерфейса подсистемы приобрели новую роль – именно на их основе строится *интерфейс* приложений. Поэтому с них мы и начнем работу по созданию нашей конфигурации.

Создадим подсистемы в новой конфигурации:

- Оперативный учет материалов
- Учет работы мастеров
- Администрирование

Порядок создания новых объектов конфигурации выглядит так же, как он выглядел ранее. Для создания новой подсистемы нужно перейти в *ветвь* дерева конфигурации **Общие > Подсистемы**, после чего либо выбрать команду **Добавить** из контекстного *меню* ветви **Подсистемы**, либо выделить эту *ветвь* и нажать клавишу **Ins**на клавиатуре, либо воспользоваться кнопкой **Добавить** из командной панели дерева конфигурации. После этого появится окно редактирования объекта конфигурации, приведенное на рисунок 1.3.

	Имя:	Подсистема1
Функциональные опции	-	
Состав	Синоним:	L
Прочее	Комментарий:	
	Включать в ком	андный интерфейс
	Командный	интерфейс
	Пояснение:	
		7.0
	Картинка:	
		<u> </u>

Рисунок 1.3. Окно редактирования объекта

Здесь можно либо перемещаться по вкладкам окна в произвольном порядке, либо, используя кнопку Далее, перемещаться по ним последовательно.

Зададим следующие параметры для нашей новой подсистемы:

Имя: УчетРаботыМастеров

Синоним: Учет работы мастеров

Синоним генерируется автоматически на основе имени, при необходимости его можно отредактировать вручную.

Поле Картинка можно использовать для того, чтобы задать подсистеме заранее созданную картинку. Это позволяет сделать интерфейс пользователя более удобным.

После того, как подсистема создана, посмотрим, на что будет похожа разрабатываемая конфигурация в режиме «1С:Предприятие». Запустим ее в этом режиме из Конфигуратора, воспользовавшись комбинацией клавиш Ctrl+F5, соответствующей командой меню (Сервис >«1С:Предприятие»), или кнопкой на панели инструментов Конфигурация.

То, что мы увидим после запуска конфигурации, разительно отличается от того, что мы привыкли видеть, рисунок 1.4.





Рабочий стол нужен для ускорения доступа пользователя к наиболее часто используемым объектам системы. Это – одна из закладок командного интерфейса, которая появляется первой при открытии конфигурации в пользовательском режиме.

Наша подсистема видна в верхней части окна программы, в так называемой панели разделов. Она снабжена стандартным рисунком, назначаемым автоматически, подпись соответствует синониму. Щелчок по вкладке "Бухгалтерский учет" приведет нас к командам по работе с объектами конфигурации, которые включены в эту подсистему.

Здесь хочется обратить ваше внимание на кнопку **Главное меню**. Она открывает *меню*, содержащее стандартные для Windows-программ команды, рисунок 1.5.



Рисунок 1.5. Главное меню в режиме «1С:Предприятие»

В сравнении с «1С:Предприятие» 8.1. в составе разделов этого *меню* многое поменялось (в особенности это касается разделов **Вид,Сервис**). В частности, обратите внимание на команду **Главное меню > Все функции**. Эта *команда*, рисунок 1.6, открывает *доступ*к дереву объектов конфигурации, позволяет использовать некоторые стандартные команды.



Рисунок 1.6. Окно Все функции

Особенно это окно полезно при разработке и отладке конфигурации – для быстрого поиска необходимых объектов без использования основного пользовательского интерфейса, для выполнения административных функций (таких, как удаление помеченных объектов, просмотр журнала регистрации). В законченной конфигурации есть смысл создать отдельную подсистему, которая будет содержать набор команд для вызова административных функций.

Вернемся в *Конфигуратор*, добавим к списку подсистем, которые следует создать, оставшиеся подсистемы – «Оперативный учет материалов» и «Администрирование».

Снова откроем конфигурацию в режиме «1С:Предприятие», рисунок 1.7.



Рисунок 1.7. Панель разделов после добавления подсистем

Если порядок расположения разделов не соответствует желаниям пользователя, то, для того, чтобы изменить порядок следования подсистем в панели разделов нужно воспользоваться командой контекстного *меню* корневого объекта дерева конфигурации **Открыть командный интерфейс конфигурации**, рисунок 1.8.

Конфигурация			×
lействия - 🍈 🖉	R	x 🛧 🗣 🔲 🖫	
 СалонКрасоты Э So Общие Э So Подсис Э So Бухг 		Открыть модуль управляемого Открыть модуль сеанса Открыть модуль внешнего сое,	приложения динения
술컵 One		Открыть командный интерфей	с конфигурации
зд Адм Зд Уче Зд Рас		Открыть рабочую область рабо Открыть командный интерфей	очего стола с рабочего стола
 Общие Параме Роли Общие 	3	Справочная информация Все подсистемы Все роли Все остраничения лостипа	
👸 Планы (**	Все картинки	Alt+Shift+Enter
		Свойства	Alt+Enter

Рисунок 1.8. Открыть командный интерфейс конфигурации

В появившемся окне мы можем управлять порядком следования подсистем на панели разделов и их видимостью. Еще одной полезной возможностью настройки видимости подсистем является видимость по ролям. С помощью этого механизма можно конструировать интерфейсы для отдельных ролей, которые можно назначать пользователям, формируя, таким образом, рабочую среду, которая не содержит ничего лишнего. Настроим порядок следования подсистем с помощью кнопок **Переместить вверх** и**Переместить вниз**.

После нажатия на кнопку **ОК** и запуска конфигурации в пользовательском режиме, внесенные изменения можно будет наблюдать на панели разделов.

Теперь внесем в нашу конфигурацию еще одно изменение. Добавим в *ветвь* Справочники новый справочник, назовем егоСотрудники (*реквизит* Имя на вкладке Основные) и добавим во все подсистемы, рисунок 1.9.

Справочник Сотрудники		_ 🗆
Основные	Подсистемы, в которых участвует справочник:	
Подсистемы		
Функциональные опции	✓ Оперативный УчетМатериалов	
Иерархия	 Администрирование УчетРаботы Мастеров 	
Владельцы		
Данные		
Нумерация		
Формы		
Команды		
Макеты		
Ввод на основании		
Права		
Обмен данными		
Прочее		
	Действия – (Назад Далее) Закрыть (Справка

Рисунок 1.9. Новый справочник, добавленный во все подсистемы

Константы

Константы в «1С:Предприятие» используются для хранения информации, которая либо не меняется никогда, либо меняется – но очень редко. Константы содержатся в ветви дерева конфигурации Константы. Создадим новую константу (рисунок1.10), заполним ее параметры следующим образом:

Имя: ТекстСообщения Тип: Строка Длина: 50

Включим константу в состав подсистемы **УчетРаботыМастеров** (Свойства-Дополнительно). Предполагается, что данная константа будет использоваться для показа сообщения пользователям, входящим в систему.



Рисунок1.10. Настройка параметров новой константы

Посмотрим, как включение константы в подсистему УчетРаботы Мастеров, отразится на интерфейсе нашего приложения в режиме «1С:Предприятие». Видно, рисунок1.11., что в разделе Учет работы мастеров, под панелью разделов, появилась еще одна панель. Она называется панелью действий. В панель действий автоматически включаются команды, разбитые на группы –Сервис, Создать, Отчеты. Группы в панели действий можно создавать и самостоятельно. В нашем случае в панели

действий видна *группа* Сервис, содержащая команду для работы с только что созданной константой.



Рисунок1.11. Константа в панели действий в разделе Учет работы мастеров

В левой части окна программы можно видеть еще одну панель – она называется панелью навигации. Сейчас она отображает ссылку для доступа к справочнику Сотрудники, который мы создавали в предыдущей лекции. Свободная часть окна – это рабочая область, в которой, например, открываются списки справочников.

Щелкнем по команде **Текст сообщения** в панели действий. Отобразится окно, которое позволяет нам редактировать константу**ТекстСообщения**. Введем в *поле* **Текст сообщения** строку "Здравствуйте, уважаемый *пользователь*!", рисунок1.12, и нажмем на кнопку **Записать и закрыть**.

🖻 Текст сообщения (1С:Предприятие)	🖬 🔳 M M+ M- 🗆 🗙
Текст сообщения	
Записать и закрыть 📄	Все действия 👻 ②
T	

Рисунок1.12. Форма редактирования константы Текст сообщения

Если мы не хотим сохранять внесенные изменения, можно просто закрыть окно с помощью стандартной кнопки Закрыть, для записи изменений без закрытия формы служит кнопка Записать объект.

Для того, чтобы воспользоваться дополнительными возможностями по работе с формой, можно использовать *меню* Все действия, рисунок1.13.

1	M 🖬 M	M+ M- 🗆 🗙	
	Bce	действия	
ватель!		Записать	Ctrl+S
		Изменить форму	
	2	Справка	F1

Рисунок1.13. Меню Все действия

Отчасти оно дублирует кнопки, имеющиеся на форме, в нем так же имеется одна специфичная для платформы «1С:Предприятие» 8.2. команда. А именно, речь идет о команде Изменить форму.

Форма, которую мы видим, сформирована автоматически. Однако в режиме «1С:Предприятие» мы можем вносить в нее некоторые изменения. Выполним команду **Изменить форму**, появится окно Настройка формы, рисунок1.14.

 Свойства элемента формы Заголовок Положение заголовка Подсказка Активизировать при открытии Растягивать по горизонтали 	Все действия • Текст сообщения Авто •
Заголовок Положение заголовка Подсказка Активизировать при открытии Растягивать по горизонтали	Текст сообщения Авто
Растягивать по вертикали	Авто
	Ширина

Рисунок1.14. Окно Настройка формы

В нашем случае, рисунок1.14, в группе Элементы формы выделен элемент Текст сообщения, в группе Свойства элемента формы мы можем настраивать его свойства. Изменим свойство Заголовок, вместо "Текст сообщения" введем "Текст сообщения для пользователей", в итоге форма будет выглядеть так, как показано на рисунке1.15.



Рисунок1.15. Отредактированный заголовок объекта

Перейдем в режим конфигурирования, создадим еще одну константу (она пригодится нам позже):

Имя: ПрефиксНомера

Тип: Строка

Длина: 2

Включим эту константу в подсистему Администрирование. В режиме «1С:Предприятие» *доступ* к этой константе будет организован в группе Сервис панели действий раздела Администрирование. Кроме того, мы можем организовать *доступ* к константам из других мест нашего приложения. Мы можем самостоятельно включить команду для вызова формы просмотра и редактирования*константы*, отредактировав командный *интерфейс*, можем так же создать специальную форму, называемую формой констант.

Форма констант

Для создания формы констант нужно вызвать контекстное меню ветви Константы дерева конфигурации и выбрать В нем констант. В появившемся окне Конструктор командуСоздать форму общих форм, рисунок1.16, нужно оставить тип формы в значении Форма констант, при необходимости заполнить другие поля и нажать на кнопку Далее.

Конструктор	общих форм	×
- Выберите ти	п формы:	
ОПроизвол	льная форма	
• Форма к	онстант	
Форма о	тчета	
• Форма на	астроек отчета	
Форма в	арианта отчета	
Имя:	ФормаКонстант	
Синоним:	Форма констант	
Комментари	й:	
Расширенное	е представление:	
Пояснение:		
Использо	вать стандартные команды	
-		
Командна	ия панель формы сверку	
Командна	я панель формы снизу	
< Hasan	Далее > Готово Отмена Спр	авка
C Manual		

Рисунок1.16. Конструктор общих форм

В появившемся окне, рисунок1.17, можно настроить состав формы констант, в нашем случае нас устраивает то, что в нее включены обе созданные в конфигурации *константы*, поэтому нажмем на кнопку **Готово**.

			and a second second	199400162081	
Рекви	зиты				
 Image: A state Image: A state<td>ТекстСо</td><td>общения</td><td></td><td></td><td>_ 0</td>	ТекстСо	общения			_ 0
	Префик	сномера			_

Рисунок1.17. Конструктор общих форм, состав формы констант

В ветви Общие формы появится новая форма с именем ФормаКонстант, будет открыто окно редактирования формы, рисунок1.18.

) Ø 🗙 🛧 🗣 🔖	5 III & X			
Форма Форма Командная панель	Реквизит		Использо всегда	Тип
 ТекстСообщения ПрефиксНомера 	HatopKonctant			(КонстантыНабор)
Злементы 🗍 🧧 Командный интерфейи	с 🗧 Реквизиты 🦉 Ком	ианды	🚦 Napa	аметры
	Rea	a neder	вия + (7)	
Записать и закрыть 📄	but	e postici		
Записать и закрыть		, there		
Записать и закрыть 📄 Текст сообщения:	.004	ACTION		
Записать и закрыть 📷 Текст сообщения:				

Рисунок1.18. Окно редактирования формы

При разработке командного интерфейса в «1С:Предприятие» 8.2 используется концепция декларативного описания. Это означает, что разработчик задает схематичное описание элементов интерфейса, их группировку, свойства. При построении интерфейса для конкретного пользователя система создает его, опираясь на описание, сделанное разработчиком, учитывая различные дополнительные факторы, например, такие как *права* пользователя и пользовательские настройки интерфейса.

Форму констант также нужно включить в одну из подсистем. Включим ее в подсистему **Администрирование**, посмотрим, что у нас получилось, рисунок1.19.

Форма ко	нстант	
Записать и	закрыть	Все действия 👻 🌔
Текст сообщен	ия: Здравствуйте, уважаемый пользователь!	
Префикс номер	a:	

Рисунок1.19. Окно редактирования формы

Мы видим, что форма констант доступна в группе Сервис панели действий раздела Администрирование. В текущей ситуации наличие в той же группе команды вызова окна константы Префикс номера может показаться избыточным. Для того чтобы убрать эту команду из панели действий, нам понадобится отредактировать командный интерфейс. Для этого мы можем выполнить команду Открыть командный интерфейс подсистемы Администрирование и в появившемся окне, рисунок1.20, снять флагВидимость для команды Префикс номера группы Сервис панели действий.



Рисунок 1.20. Настройка панели действий

Теперь при запуске в режиме «1С:Предприятие» ненужная команда отображаться не будет.

Выше мы создавали константу Текст сообщения, предполагая выводить заданный в ней текст в качестве сообщения для пользователей, входящих в систему. Реализуем эту функциональность. Для этого нам понадобится написать код в модуле управляемого открыть приложения. Для того, чтобы этот модуль, нужно воспользоваться командой Конфигурация - Открыть модуль управляемого приложения корневого элемента конфигурации. Для этого модуля предусмотрено несколько стандартных обработчиков событий. которые можно найти в панели инструментов Модуль, рисунок1.21. Hac интересует

обработчикПриНачалеРаботыСистемы.



Рисунок 1.21. Выбор обработчика ПриНачалеРаботы Системы

В модуле появится пустое тело обработчика, в которое нам нужно ввести команду для вывода сообщения пользователям. Если попытаться обратиться к константе напрямую из модуля управляемого приложения – мы столкнемся с ошибкой. Дело в том, что*исполнение* модуля управляемого приложения происходит на клиенте, в контексте которого нет доступа к константам. Поэтому нам понадобится код, который выполняется на сервере и возвращает *значение константы*. В данном примере мы можем возложить на серверную часть примера и *вывод* сообщения (*функция* Сообщить, которой можно здесь воспользоваться, работает и на клиенте, и на сервере). Но гораздо полезнее, в плане перспектив повторного использования кода серверной процедуры, "вытащить" серверные данные в метод, который выполняется на клиенте.

Создадим новый общий *модуль* (в ветви **Общие модули** *дерева конфигурации*), назовем его **СерверныеФункции**. Проследим за тем, чтобы в его свойствах были установлены флаги **Сервер** и **Вызов сервера**, рисунок1.22.

Свойства:	Серверные Функции	×
21 🖬 🖌	× ×	
Основные	ə:	
Имя	Серверные Функции	
Синоним	Серверные функции	
Комментар	ий	
Модуль Глобальны	<u>Открыть</u> й 🗌	
Клиент <mark>(у</mark> пр	равляемое приложение) 🗌	
Сервер		
Внешнее со	рединение	
Вызов сера	вера 🔽	
Привилеги	рованный	
Повторное	использование возвращаемы Не использовать	-
иноним объ	екта метаданных	



Откроем *редактор кода* для кода модуля (например, двойным щелчком по модулю в *дереве конфигурации*) и введем следующий код, рисунок1.23.:

//Экспортная функция для вызова из других модулей

Функция ПолучитьКонстанту() Экспорт

//Возвращаем полученное значение константы

Возврат(Константы. ТекстСообщения. Получить());

КонецФункции



Рисунок 1.23. Общий модуль Серверные Функции, код

Теперь нам нужно вызвать эту функцию в подходящем месте кода обработчика события **ПриНачалеРаботыСистемы** в модуле управляемого приложения. Например, это можно сделать так:

Процедура ПриНачалеРаботыСистемы()

//Выводим сообщение пользователю

Сообщить(СерверныеФункции.ПолучитьКонстанту());

КонецПроцедуры

В результате при входе в систему мы получим сообщение следующего вида, рисунок1.24.



Рисунок1.24. Вывод сообщения пользователю

Обратите внимание на то, что сообщение выводится в область **Сообщения** основного рабочего окна. Если сообщение вызвано из модуля какоголибо отдельного окна, например, из модуля формы констант, которая создана ранее, то, по умолчанию, сообщение будет выведено в этом окне.

Посмотрим на этот механизм в действии. Откроем окно редактирования формы констант (Общие формы >ФормаКонстант), перейдем на вкладку Модуль, на панели инструментов Модуль выберем стандартный обработчик события ПриОткрытии, отредактируем тело обработчика, чтобы оно приняло следующий вид, рисунок1.25:

&НаКлиенте

Процедура ПриОткрытии(Отказ)

Сообщить("Вы открыли форму констант!");

КонецПроцедуры

	Конфигурация СалонКрасоты: ФормаКонстант	_	×
	бНаКлиенте ☐ Процедура ПриОткрытии (Отказ) Сообщить ("Вы открыли форму констант!"); КонецПроцедуры		
N N	Форма Модуль		4

Рисунок1.25. Вывод сообщения пользователю из модуля формы констант

Благодаря этому коду при открытии формы констант будет появляться следующее сообщение, рисунок1.26.

Форма констант (1С:Предприятие)	18 🔟 M M+	M- 🗆 🗙
Форма констант Записать и закрыть 📄 Все действия - (Текст сообщения: Здравствуйте, уважаемый пользователь) Префикс номера:	Сообщения Вы открыли форму конста	Х

Рисунок1.26. Вывод сообщения в форму констант

Основы клиент-серверного программирования

внимание на директиву компиляции & НаКлиенте, Обратите которая, В вышеописанном участке кода. автоматически размещена перед описанием процедуры **ПриОткрытии()**. Создавая решение для «1С:Предприятие» 8.2 разработчик должен четко разграничивать код, исполняемый в клиентской и серверной частях приложения. Причем, на клиенте (в контексте клиента) и на сервере (в контексте сервера) доступны различные объекты, различные программные механизмы. Основная задача серверного кода заключается во взаимодействии с базой данных, клиентский код занимается отображением этих данных и взаимодействием с пользователем. А задача разработчика заключается в том, чтобы создать код с учетом клиент-серверного взаимодействия.

Если перед описанием процедуры, функции или переменной в модуле формы отсутствует *директива* компиляции, по умолчанию считается, что код будет исполняться на сервере. В явном виде это задается указанием директивы **&HaCepbepe**.

Попытаемся в нашем модуле формы вывести в окно сообщения значение константы. Для этого мы можем добавить в модульфункцию, возвращающую значение константы, которая должна выполняться в контексте сервера. Например, это можно сделать одним из следующих способов – ниже приведена дополненная процедура ПриОткрытии и еще пара процедур, заданных в коде модуля формы:

&НаКлиенте

Процедура ПриОткрытии(Отказ)

Сообщить("Вы открыли форму констант!");

Сообщить(ПолучитьКонстанту()+" - из функции модуля формы без директивы");

Сообщить(СерверныеФункции.ПолучитьКонстанту()+" - из общего модуля");

Сообщить(ПолучитьКонстантуНаСервере()+" - из функции модуля формы с директивой &НаСервере");

КонецПроцедуры

//По умолчанию функция считается серверной

Функция ПолучитьКонстанту()

//Возвращаем полученное значение константы

Возврат(Константы.ТекстСообщения.Получить());

КонецФункции

//Директива компиляции задана явно

&НаСервере

Функция ПолучитьКонстантуНаСервере()

//Возвращаем полученное значение константы

Возврат(Константы. ТекстСообщения. Получить());

КонецФункции

Здесь мы создали пару функций – одну назвали ПолучитьКонстанту(), при ее описании директиву компиляции ΜЫ не указывали. Вторую назвали ПолучитьКонстантуНаСервере() при ee описании была указана директива &НаСервере. Мы вызвали эти функции для вывода сообщения в клиентской процедуре ПриОткрытии(). У нас уже есть серверная функция в общем модулеСерверныеФункции – здесь показан пример ее использования, в подобном случае, возникшем при реальной разработке, если действия, которые выполняются в серверной функции модуля формы, совпадают с действиями функции, описанной в общем модуле, можно и даже нужно пользоваться функцией общего модуля.

На рисунке1.27 вы можете видеть *вывод* сообщений, выполненный вышеприведенным кодом.

🔨 Форма констант (1С:Предприятие)	Ш <mark>Ш М м+ м</mark>	×□
Форма констант	Сообщения	×
Samuran u samun	в рейстрия - 2 Вы открыли форму констант	1
Текст сообщения: Здравствуйте, уважаемый пользова	Здравствуйте, уважаемый пользователь! - из функции м формы без директивы	иодуля
Префикс номера:	Здравствуйте, уважаемый пользователь! - из общего м	одуля
	Здравствуйте, уважаемый пользователь! - из функции и формы с директивой &НаСер	иодуля овере

Рисунок1.27. Вывод сообщения в форму констант, разные варианты работы с серверными данными

Вернемся к директивам компиляции. Они могут использоваться не только в модулях форм, но и в общих модулях, и в модулях команд. Таблица 1.1. содержит описание возможности применения директив компиляции в различных модулях

	Модули			
Директивы компиляции	Модуль	Модуль	Общий	Переменные
	формы	команды	модуль	
&НаКлиенте	Да	Да	Да	Дa
&НаСервере	Да	Да	Да	Дa
&НаКлиентеНаСервере	Нет	Да	Нет	Нет
&НаСервереБезКонтекста	Да	Нет	Нет	Нет
&НаКлиентеНаСервереБезКонтекста	Да	Нет	Нет	Нет

Таблица 1.1. Применение директив компиляции в модулях и переменных

Опишем основные особенности их применения:

&НаКлиенте – эта *директива* предназначена для клиентских процедур и функций. Из такого метода могут быть вызваны любые процедуры и функции. Кроме того, с данной директивой можно объявлять переменные – их называют клиентскими. Такая*переменная* существует столько же, сколько существует клиентская часть формы. Из метода с данной директивой доступны другие клиентские переменные модуля формы.

&НаСервере – эта *директива* предназначена для серверных процедур и функций. Из такой процедуры могут быть вызваны серверные и клиент-серверные внеконтекстные методы, а так же методы неглобальных серверных общих модулей. Допустимо объявление переменных с данной директивой – такие переменные существуют во время выполнения вызова сервера. Из серверных методов доступны серверные переменные, объявленные в модуле формы.

&НаКлиентеНаСервере – предназначена для описания процедур и функций, выполняемых на клиенте и на сервере. Такие процедуры и функции могут вызывать клиентские и серверные процедуры общих модулей. Не подходит для объявления переменных.

&НаСервереБезКонтекста – такая процедура или *функция* исполняется на сервере вне контекста формы, в модуле которой она описана. Из нее можно вызывать лишь другие внеконтекстные процедуры или функции. Использование этой директивы позволяет сократить объем данных, передаваемых между сервером и клиентом. Не подходит для объявления переменных.

&НаКлиентеНаСервереБезКонтекста – такая процедура или *функция* может исполняться на клиенте и на сервере, без доступа к контексту формы. Не подходит для объявления переменных.

Помимо директив компиляции в модуле управляемой формы можно пользоваться инструкциями препроцессору. Инструкции препроцессору обрабатываются до того, как будут обработаны директивы компиляции.

Литература:

Официальный сайт интернет-университета «Интуит» (электронный ресурс), режим доступа http://www.intuit.ru/studies/courses/2318/618/lecture/17939.

Контрольные вопросы для самопроверки

- 1. Как создать информационную базу?
- 2. Как установить свойства информационной базы?
- 3. Чем отличается директива «на клиенте» от директивы «на сервере»?

Задание к лабораторной работе

Создать информационную базу в соответствии с порядком, отраженным в теоретической части

Методические указания и порядок выполнения работы

База создается в точном порядке, как это указано на рисунках и в теоретической части

Индивидуальное задание

не предусмотрено

Требования к отчету и защите

1. Результатом выполнения лабораторной работы является сформированный в программе файл, содержащий выполненные задания. В ЭИОС результаты работы не выкладываются.

2. Планируется защита работы, где студент комментирует порядок выполнения заданий, а также отвечает на вопросы, представленные выше.

ЛАБОРАТОРНАЯ РАБОТА №2

РАБОТА СО СПРАВОЧНИКАМИ

ОБЩИЕ СВЕДЕНИЯ

Цель: научиться информационную базу на платформе ««1С:Предприятие»»

Материалы, оборудование, программное обеспечение: 1. персональный компьютер (компьютерные классы ГУК), 2. программное обеспечение «1С:Предприятие».

Условия допуска к выполнению: умение работать на ПК и знание техники безопасности.

Критерии положительной оценки: предоставление результатов работы в виде файла и прохождение защиты.

Планируемое время выполнения: Аудиторное время выполнения (под руководством преподавателя): 4 ч. Время самостоятельной подготовки: 2 ч.

Теоретическое введение

Обсудив основы работы с константами и основные положения клиент-серверного программирования, перейдем к работе со справочниками. Обычно разработка системы справочников относится к начальному этапу разработки любой конфигурации, так как на типах данных, заданных справочниками, основываются другие *механизмы* системы. Но прежде чем начать разговор, о, собственно, справочниках, давайте поговорим об общих реквизитах.

Общие реквизиты

Общие реквизиты являются новой возможностью, которая была добавлена в 14й *релиз* платформы «1С:Предприятие» 8.2. Общие реквизиты можно использовать двумя способами. Первый из них заключается в использовании их, как, собственно, реквизитов, которые присутствуют у всех (или у достаточно большого количества) объектов конфигурации. Второй способ предусматривает использование общих реквизитов в механизме разделения данных в качестве разделителей данных. В базе, использующей механизм разделения данных, могут работать несколько пользователей, набор данных каждого из которых не пересекается с набором данных других пользователей, т. е. каждый из них считает, что база содержит лишь "его" данные и ничего больше.

Сейчас мы создадим общие реквизиты, которые планируется использовать именно как реквизиты для других объектов конфигурации. В частности, общий *реквизит* может быть "подключен" к следующим объектам. Правильным будет и утверждение о том, что эти объекты могут "входить в состав" общего реквизита, так как, собственно, от настройки состава общего реквизита зависит его появление в других объектах. Итак, речь идет о следующих объектах:

- планы обмена,
- справочники,
- документы,
- планы видов характеристик,
- планы счетов,
- планы видов расчета,
- регистры сведений,

- регистры накопления,
- регистры бухгалтерского учета,
- регистры расчета,
- бизнес-процессы,
- задачи.

В нашем учебном примере мы собираемся вести в базе данных учет по нескольким организациям. Для этого нам понадобится, чтобы все объекты конфигурации, для которых уместен данный *реквизит*, содержали бы *реквизит* **Организация**, который содержит ссылку на организацию. Например, каждый документ будет оформляться от лица определенной организации, каждый элемент справочника будет относиться к той или иной организации, и так далее. Для того чтобы не усложнять наши примеры, мы не будем в дальнейших лекциях курса развивать тему многофирменного учета в одной базе данных. Однако в любом случае общие реквизиты позволяют снизить трудоемкость разработки.

Второй *реквизит*, который предназначен для документов, будет использоваться для ввода комментариев к документу.

Прежде чем продолжать работу над общими реквизитами, создадим следующие объекты конфигурации, не настраивая их дополнительных свойств – справочник с именем **Организация**, и документ с именем **ПоступлениеМатериалов**. Включим их в подсистему **ОперативныйУчетМатериалов**.

Создадим новый общий реквизит со следующими параметрами, рисунок 2.1.:

Имя: Комментарий

Тип: Строка, длина 50

Свойства: Коммент	гарий	×	Общий реквизит Комментарий: Состав	_ 🗆 ×
11 国家 × ~			Determine + of of V (2)	
• Основные:		(A)		(management)
Имя Комми	внтарий		UOBEKTSI	Использование
Синонии Комм	внтарий			
Комментария			Поступление Материалов	Использовать
Tim	Строка	¥	No	
Дяина	50 \$			
Допустимая длина	Переменная	-		
Неограниченная дли-	sa 🛄			
• Использование:			1	
Состав	Открыль		Weight and the set of the set	39
Автоиспользование	Не использовать	-	используемые объекты.	•
and the second second		100	Объекты	Использование
Разделение данных	Не использовать		🛞 🔛 Документы	
Индексировать	Не индексировать			
Полнотекстовый пои	ск Использовать			
• Представление:		12		
Режим пасоля	0	Ψ.		
Длина				
1			L.	

Рисунок 2.1. Настройка общего реквизита

Обратите внимание на *параметр* Автоиспользование. В данном случае мы оставляем его в значении по умолчанию – Не использовать. То есть – состав общего реквизита мы будем настраивать вручную. Этот общий *реквизит* мы планируем добавить ко всем документам, поэтому найдем свойство Состав, нажмем на ссылку Открыть, в появившемся окне выберем вариантИспользовать для документа ПоступлениеМатериалов. При создании других документов мы сможем самостоятельно включать их в состав общего реквизита. Быстро проверить состав используемых объектов общего реквизита можно в нижней части окна настройки состава.

Преимущества использования общих реквизитов напоминают использование процедур и функций в общих модулях, к которым обращаются из многих других методов.

Если возникает необходимость в изменении общего реквизита – например – в процессе работы в конфигурации оказалось, что длину комментария нужно увеличить – достаточно изменить параметры типа общего реквизита, и это изменение затронет все объекты конфигурации, включенные в его состав.

Создадим второй общий реквизит:

Имя: Организация

Тип: СправочникСсылка.Организации

Автоиспользование: Использовать

Этот *реквизит* мы планируем добавить ко всем объектам, допускающим использование общих реквизитов, за исключением справочника **Организации** и некоторых других. Перейдем в окно настройки состава общего реквизита и установим свойство**Использование** у справочника **Организации** в *значение* **Не использовать**, рисунок 2.2.

Свойства. О	рганизация	×	Общий реквизит Организация: Состав	_ 🗆 >
11 11 1	× ×		Dedoctions + + + + 12 (2)	
•Основные:		A.		Logist Astronomy
Имя	Организация		Объекты	Использование
Синоним	Организация		😔 🖽 Справочники	
Комментария	a		Ш Сотрудники	M Автоматически
			Организации	📄 Не использовать 📃 💌
Тип Справоч	никСсылка.Организации		😑 🔜 Документы	
• Использов	ание:		Поступление Материалов	Автоматически
Состав	Открыть			
Автоиспольз	ование Использовать	*		
Разделение и	данных Не использовать	•	Не используемые объекты:	50 •
Индексирова	нь Не индексировать	-	Объекты	Использование
Полнотексто	вый поиск. Использовать		😔 🎹 Справочники	
• Представля	CHINC:		Организации	🔛 Не использовать
Подсказка				
Объекты мета	данных, включающие данный общий ре	КВИЗИТ		
Дополны	ительно: О 🛛 📑 Свойства: Органи	(3	1 ¹	

Рисунок 2.2. Настройка общего реквизита

Откроем нашу конфигурацию в режиме «1С:Предприятие» и посмотрим, как выглядит документ ПоступлениеМатериалов и

справочники Организации и Сотрудники.

Для начала перейдем на вкладку Оперативный учет материалов. Обратите внимание на то, что в панель навигации раздела были автоматически добавлены ссылки созданному справочнику Организации и лля доступа к только что К документуПоступление материалов. Щелкнем по ссылке Организации. В рабочей области окна появится список справочника. На данный момент он пуст, так как мы пока не заполняли справочник организациями, по которым будет вестись учет в базе. Щелкнем по кнопке Создать, которая расположена на командной панели списка – появится отдельное окно для заполнения свойств элемента справочника. Можно отметить, что помимо стандартных реквизитов (Наименование, Код) данный справочник не содержит ничего другого – это неудивительно, мы исключили его из состава общего реквизита Организация.

Теперь откроем *список* справочника Сотрудники и нажмем на кнопку Добавить. Общий *реквизит* Организация у данного справочника присутствует.

Откроем, наконец, окно создания документа ПоступлениеМатериалов. Здесь мы видим два общих реквизита – Комментарий иОрганизация.

Справочник «Организации»

Справочник можно сравнить с картотекой, с неким списком данных, каждая *запись* которого имеет определенную структуру. В организации – независимо от того, автоматизирован ли в ней учет или нет, присутствует множество таких списков. Это – списки сотрудников, клиентов, товаров.

В нашей конфигурации уже есть пара справочников. Один из них – это справочник **Организации**, который нужен для хранения списка организаций, *по* которым планируется вести учет. Справочник, сразу после его создания, имеет некоторые стандартные реквизиты. Это утверждение справедливо и для других объектов конфигурации. Для управления реквизитами объекта служит закладка Данные окна редактирования объекта, рисунок2.3.

Длина кода	9
Длина наименовани	19 25
– Тип кода – – – – – – – – – – – – – – – – – – –	Основное представление
O Число	🔘 В виде кода
Острока	• В виде наименования
O / X A	
😑 Реквизиты	
胡 <i>녹 / ×</i> 4	
tii 🖕 🖋 🗶 н Пії Табличные	насти
	Длина наименовани Тип кода ОЧисло ОСтрока Строка

Рисунок2.3. Настройка справочника

Ознакомиться со списком стандартных реквизитов можно, нажав на кнопку Стандартные реквизиты – появится окно, содержащее *список* таких реквизитов, рисунок2.4.

🗧 Справочник Организации: Стандартные реквизиты 💦 💶 🗙	Свойства: Сп: Стандартные реквизиты - Наименование 💦 🗙
Стандартные реквизиты:	1 1 X × -
 ЭтоГруппа. Код Наименосоние Родитель Впаделец Впаделец ПометкаУдаления Ссылка Предопределенный 	
	Маска Многострочный режим Расширенное редактирование

Рисунок2.4. Стандартные реквизиты справочника и их свойства

Стандартные реквизиты поддерживают настройку некоторых свойств – для доступа к свойствам стандартного реквизита, достаточно выделить его в окне и обратиться к палитре Свойства.

Нашему справочнику **Организации** не хватает, для полноты его использования в системе, реквизита, который содержал бы полное наименование организации. Добавим этот *реквизит* к справочнику – на вкладке **Данные** окна редактирования объекта, нажмем на кнопку **Добавить**, параметры реквизита будут следующими:

Имя: ПолноеНаименование

Тип: Строка, длина – 50.

Проверка заполнения: Выдавать ошибку

Свойство **Проверка заполнения** *по* умолчанию для новых реквизитов установлено в *значение* **Не проверять**. Оно позволяет автоматически проверять заполненность поля – если *поле* не заполнено – система выдаст ошибку (рисунок 2.5). Если нам нужны особые алгоритмы проверки содержимого поля перед записью элемента справочника, мы можем реализовать эти алгоритмы самостоятельно.

Основные.	Данна кора	
Подсистемы		25 1
Функциональные опции	- Тип кода Основное предст	Использование Для элемента
Иераркия	Очисло ОВ виде кода	Индексировать Не индексировать
Владельцы	•Строка •В виде наиме	новнотекстовыи поиск (использовать
Данные		Режим пародя
Нумерация	◎ / × * * □	Подсказка
Формы	😔 🛥 Реквизиты	
Команды	😑 Полное Наименование	
Макеты		Maoka
Ввод на основании		Расширенное релактирование
Права		
Обмен данными	出てマメキャロ	Заполнять из данных заполнения
Прочее	П Табличные части	Поолеока заполнения Выликать онибех и
		Condition of the second s
		Быстрый выбор Авто
	Стандартные реквизиты Харак те	ики Связь по типу
	Общие реквизиты	Operation and a state state state

Рисунок2.5. Настройка нового реквизита справочника

Посмотрим на наш справочник в режиме «1С:Предприятие». Создадим новый элемент, дадим ему наименование Салон красоты, а полное наименование заполнять не будем, и попытаемся записать элемент, нажав на кнопку Записать и закрыть. Элемент не будет записан, мы увидим сообщение об ошибке – в виде сообщения и в виде всплывающей подсказки, рисунок2.6.

Организации	(создание) *		Сообщения	×
Записать и закр		Все действия 👻 🧿	Поле "Полное наиме заполнено с	нование" не
Наименование: Полное наименовани	Салон красоты		æ	

Рисунок2.6. Сообщение об ошибке при попытке записи элемента справочника

Введем в *поле* **Полное наименование** текст *ООО* "Салон красоты" - после этого можно будет записать и закрыть элемент справочника. Он отобразится в списке справочника в рабочей области окна программы. В информационной панели, которая расположена в нижней части окна программы, появится *ссылка* для доступа к только что созданному элементу и будет сообщено о его создании.

Код элементу справочника будет присвоен автоматически.

Справочники в «1С:Предприятие» могут содержать предопределенные элементы. К их созданию можно перейти с вкладки **Прочее**,*no* кнопке **Предопределенные**.

Справочник ФизическиеЛица

Следующим нашим справочником будет справочник Физические. Он предназначен для хранения списка физических лиц и сведений о них. В частности, мы

хотели бы хранить данные о самом физическом лице (Фамилия, Имя, Отчество, дата рождения, пол, район проживания), а так же об истории его трудовой деятельности. Для хранения данных о физическом лице хорошо подойдут обычные реквизиты справочника, которыми мы уже занимались выше. А вот для того, чтобы хранить историю трудовой деятельности, нам понадобится другая *структура данных*, а именно – **табличная часть**.

Табличная часть – это *таблица*, состав и свойства полей (столбцов) которой мы задаем на этапе разработки. В пользовательском режиме создается необходимое количество строк. В нашем примере количество мест, в которых работало физическое лицо, заранее неизвестно.

Здесь надо отметить, что понятия "Сотрудник" и "Физическое лицо" - это разные вещи. Сотрудник – это тот, кто в настоящий момент работает в организации, и сотрудник обязательно является физическим лицом. А вот физическое лицо, сведения о котором могут храниться в базе данных организации, вполне может не являться сотрудником – например – это может быть кандидат на какую-либо должность, или, наоборот, уволенный сотрудник.

Создадим новый справочник, дадим ему имя Физические. Лица, включим его в состав подсистемы УчетРаботы Мастеров.

На вкладке Данные создадим следующие реквизиты:

Имя: Фамилия Тип: Строка, *длина* 30 Имя: Имя Тип: Строка, *длина* 30 Имя: Отчество Тип: Строка, *длина* 30 Имя: ДатаРождения Тип: Дата, состав даты – Дата

Следующие реквизиты, которые мы планируем создать – это **Пол** и **РайонПроживания**. Строковые реквизиты, которые мы создавали выше, обычно заполняют вводом данных с клавиатуры. В случае же с указанием пола и района проживания заполнение с клавиатуры непременно приведет к появлению в базе различных наименований для одних и тех же показателей при использовании текстовых полей. Для мужского пола это вполне может быть, при ограничении длины строки одним символом, "М" и "м", для районов так же возможно различное написание. Для обеспечения единообразия при вводе подобных показателей рационально использовать для их хранения отдельные справочники или перечисления. Для хранения наименований пола мы воспользуемся перечислением.

Создадим новое перечисление, дадим ему имя Пол, включим В подсистему УчетРаботыМастеров. На вкладке Данные окна редактирования объекта для перечисления. Зададим перечисления задаются значения два значения Мужской и Женский, рисунок2.7.

Основные	Значения перечисления:
Подсистемы Функциональные опции Данные Формы	 Э × Э = Значения Мужской Женский
Команды Макеты	
Прочее	
	Стандартные реквизиты Характеристики

Рисунок2.7. Создание перечисления Пол

Теперь создадим новый справочник – дадим ему имя **Районы**, включим в состав подсистемы **УчетРаботыМастеров**, на вкладке **Данные** изменим длину наименования до **100** символов, этот справочник не будет иметь дополнительных реквизитов, так же мы можем исключить его из состава общего реквизита **Организация**, рисунок2.8.

	Длина кода 9
Подсистемы	Длина наименования
Функциональные опции	- Тип кода
Иерархия	ОЧисло ОВ виде кода
Владельцы	• Строка • В виде наименования
Данные	
Нумерация	
Формы	📼 Реквизиты
Команды	
Макеты	
Ввод на основании	
Права	
Обмен данными	til 🏷 / X 🛧 🐥 🛄
Прочее	🕫 Табличные части

Рисунок2.8. Создание справочника Районы

Вернемся к настройке справочника **ФизическиеЛица**. Добавим еще два реквизита: **Имя**: Пол

Тип: ПеречислениеСсылка.Пол

Имя: РайонПроживания

Тип: СправочникСсылка.Районы

Теперь займемся табличной частью справочника. При необходимости, справочники могут иметь несколько табличных частей. Сначала нажмем на кнопку Добавить табличную часть, зададим имя табличной части ТрудоваяИстория. В табличную часть добавим следующие реквизиты (поля), выделив табличную часть и нажав на кнопку Добавить реквизит:

Имя: Организация

Тип: Строка, длина 30

Имя: ДатаНачалаРаботы

Тип: Дата, состав даты – Дата

Имя: ДатаОкончанияРаботы

Тип: Дата, состав даты – Дата.

В итоге окно редактирования нашего справочника будет выглядеть так, как показано на рисунке2.9.

Основные	Длина кода	9
Подсистемы	Ллина наименования	25
Функциональные опции	– Тип кода – – – – – – – – – – – – – – – – – – –	Основное представление —
Иерархия	ОЧисло	🔘 В виде кода
Владельцы	Острока	• В виде наименования
Данные		
Нумерация	🕒 🖉 🗙 🚭 😓	
Формы	🖂 📟 Реквизиты	
Команды	— Фамилия	
Макеты		
Ввод на основании	= ДатаРож.	ления
Права	— Пол	
Обмен данными	— РайонПро	живания
Прочее		
	111 to 2 X 金	₽ 🗊
	🖂 🎹 Табличные ча	сти
	 Парличные части Трудовая История Организация 	
	😑 Датаł	НачалаРаботы
	— Дата(ОкончанияРаботы
	Стандартные реквиз	иты Характеристики
	Общие реквизить	I

Рисунок2.9. Состав справочника ФизическиеЛица

В предыдущей лекции мы создавали общий реквизит Организация, который объектам планировалось добавлять конфигурации. ко многим Справочник ФизическиеЛица имеет смысл вести по всем организациям. Как вы уже видели, настроить состав общего реквизита можно в ветви Общие реквизиты. Сделать это можно и в окне редактирования объекта, нажав кнопку Общие реквизиты на вкладке Данные. Нажмем эту вкладку установим для общего И реквизита Организация значение Не использовать, рисунок2.10.

)бщий реквизит	Использование	Условное разделение
Комментарий Организация	 Автоматически Не использовать Автоматически Использовать 	
	Не использовать	

Рисунок2.10. Настройка общих реквизитов из окна редактирования объекта

Если мы попытаемся открыть справочник в режиме «1С:Предприятие» – с ним можно будет работать, так как система автоматически сгенерирует его форму, Рисунок2.11. – с автоматически созданными формами мы уже встречались ранее. Такие формы подходят в том случае, если мы не планируем каким-либо образом вмешиваться в функционирование формы из *Конфигуратора*.

Код:)	-
аименование:		·	
Рамилия:			
1мя:			
)тчество:			
Ца <mark>та рождения</mark> :)	
Іол:			
айон проживани	ия:		C
^р айон проживани 🔁 Добавить	19: B X A V		Все действия
Район проживани ФДобавить N Организ	ия: 🚺 🗙 🛧 🐥 Вация	Дата начала работы	Все действия Дата окончания работы
^р айон проживани Ф Добавить N Организ	ия: В X 🔒 🗣 ация	Дата начала работы	Все действия Дата окончания работы
Район проживани ФДобавить N Организ	ия:	Дата начала работы	Все действия Дата окончания работы

Рисунок 2.11. Форма справочника, сгенерированная автоматически
Если же решаемая нами задача требует каких-то особенных приемов работы с формой объекта, нам понадобится собственная форма. Например, это нам понадобится, если мы хотим автоматически заполнять *поле* **Наименование** на основе полей **Фамилия**, **Имя** и **Отчество**. А именно, мы хотели бы, чтобы наименование содержало фамилию и инициалы физического лица.

Разработка формы справочника ФизическиеЛица

Откроем закладку **Формы** окна редактирования справочника **ФизическиеЛица**. Можно отметить, рисунок2.12, что ни одной формы не задано – то есть все они создаются системой автоматически. Нам же нужна собственная форма **элемента** справочника.

Основные	Формы					
Подсистемы	Элемента	×Q				
Функциональные опции	Группы	X Q				
Иерархия	Списка	× 0				
Владельцы						
Данные	Deloupa					
Нумерация	выбора группы	×Q				
Формы						
Команды	😌 / X 🛧 🗣 📕					
Макеты	📰 Формы					
Ввод на основании						
Права						
Обмен данными						
Прочее						
	Ввод по строке Наиме	енование, Код 🛄				

Рисунок 2.12. Вкладка Формы окна редактирования объекта

Нажмем на кнопку с увеличительным стеклом напротив поля Элемента в группе Формы. Появится окно Конструктора формы справочника, в его первом окне оставим все *по* умолчанию – а именно – нас интересует Форма элемента справочника, рисунок2.13.

	ормы справочника
Выберите тиг	т формы:
• Форма эл	емента справочника
Форма гру	уппы справочника
🔿 Форма сп	иска справочника
Форма вы	бора справочника
Форма вы	бора группы справочника
ОПроизволь	ьная форма
Основная Имя:	форма элемента и группы ФормаЭлемента
Основная Имя: Синоним:	Форма элемента и группы ФормаЭлемента Форма элемента
Основная Имя: Синоним: Комментарий	Форма элемента и группы ФормаЭлемента Форма элемента

Рисунок 2.13. Первое окно конструктора форм справочника

В следующем окне, рисунок2.14, мы можем указать состав реквизитов для расположения на форме, а так же указать количество колонок, которое нужно для расположения элементов управления на форме. Оставим здесь все так же *по* умолчанию и нажмем на кнопку **Готово**.

квизиты	A.
Ссылка	
🗸 Код	
И Наименование	
ПометкаУдаления	
Предопределенный	
🗹 Фамилия	
Имя	t
Отчество	
ДатаРождения	
🗸 Пол	
РайонПроживания	
У ТрудоваяИстория	
ИмерСтроки	
Организация	
🗹 ДатаНачалаРаботы	
□ □ 0	

Рисунок2.14. Второе окно конструктора форм справочника

После этого нужно открыть окно редактора форм для формы элемента справочника, рисунок2.15. Ранее мы уже сталкивались с этим окном, теперь рассмотрим его подробнее.

2 × 4 4 15		NIL A X			
		and a set	Honor some	Tues	
North Contraction Contractory	-110	monan i	BCBT DB		
- Kos		- Oferst	1 11000000	Copapos	никОбъект. Физическ
- Hannessen		Red ad a state of the state of		(produce a	
- Constant					
- Mare					
- Cruerten					
Pon	and the second se				
Pato-Dogsustature					
Точаранистория					
ш Команская панерь	-				
Sticken is		POLERSHIDE S NO	indifiation (1) and a second states		
ameritaria:			14 (M) 10 V V		
			•Осноеные:		
B:			24 (44) № А Х • Основение: Ватоловок		
ecmo:			24 (H) E A Y Ochoemae: Baronosok Aano3aronosok Dosensex		
нилия:ааа			21 (41) (40) А У • Основение: Заголовок Авто Заголовок Доступность Регламоть Логово	0.00	
нылыя: я: ество: та рождания:			21 (23) (20) А У • Основение: Ваголовок Авто Заголовок Доступность Разрешить изменять Форму Глатановика	S S S Bern	*20.435
нылыя: я: нество: та рождения:			21 (23) (23) Колонический Соновение: Заголовок Доступность Разрешить изменять Форму Групность Пастик Изменять Форму Групносека Шастик Розвение и Описат	September 2	жальная
никлина: а: нество: та рождения: л: Кон проживания:			21 (41) 10 Х Х Основные: Заголовок Доступность Разрешить изменять Форму Группировка Шахика Подиненные Элемен История, заявает	Bepty Bepty TOB Asto	жальная
нилия: а: нество: та рождения: л: Мон прохонания: Добавиль 💫 💥 🍲 🌩	іні іні Бое действия -		 Склосные: Осносные: Заголовок Доступность Разрешить изменять форму Грятировка Шахила Поденсина/Элемен Истользование: Поподжение Командор Панел 	Bepriv Bepriv	жальная
никлия: а: нество: та рождения: л: Кон прохонания: Добавиль 20 Ж 20 Ф	Boe gestioners +		21 (43) 80 Х Х Основные: Заголовок Доступность Разрешить изменять Форму Группировка Шискиз Подчинень «Элемен Моложание Командной Панег Состае командной Панег	Bepru Bepru H	жальная (Авто Открыть
милия: в ство: та рождения:	начала работы Дата о(А		21 (43) 40 Х Х Основные: Заголовок Доступность Разрешить изменять Форму Группировка Шискиа Пориниень обланиен • Использование Положение Командной Панег Состав командной Панег Состав командной	U U Bepru HTOB Astro	жальная Авто Опсрыть Боскирсевть окно впадель
ниция: в: ноство: та рождения: л: тон прохонания: Добавить 20 ж Ф Организация Дата и	ии Q Все действия + начале работы Дата о(#		21 (43) 40 Х Х Основение: Заголовок Доступность Разрешить изменять Форму Группировка Шискина Подиниенных Элемен • Истользование: Положение Командной Панег Состав команд Резон Опрытиз Оназ Поводение Клавиций Effer	С. С. Варти ПОВ. Авто И	кальная Авто Опсрыть Блокировать окно впадаль Баскировать окно впадаль
некликя: в: есство: та рождения: л: тон прокинания: Добавить 20 ж 10 Ф Организация	начала работы Дата ог 4		 21 (43) (40) А Основеные: Заголовок Доступность Резрешить изменять форму Группировка Ширина Подчиненнь «Эпемен Истользование: Попожение Командной Пенел Состав команд Резона Опрытия Пона Поводение Клавици Effer Поводение Клавици Effer 	С Вартински	кальная Авто Опорыть Блокировать окно владель Переход по элементам Ф
милия: в сство: та рождения:	начала роботы Дата о А		21 (43) 40 Х Х Основные: Заголовок Доступность Разрешить изменять Форму Группировка Шискиа Подиниенных Элемен • Истользование: Положение Кланца Резиние Кланца Резиние Кланца Повединие Кланца Етег Проверять Заполнение Аетон Использование Для ГруппИЭ	У У Верти Нова Авто М матически печчентов	кальная Авто Опсрыть Беккирсеать окно владель Переход по элементам У Для элемента
нилия: а: нество: та рожления: л: Кон прожнания: Добавиль 💫 🞉 🌚 🧇 Организация	начале работы Дата ох А		 Склюсные: Осносные: Этолосок Алто Заголовок Доступность Разрешить изменять форму Группировка Шкрана Подененьо/Элемен Истоль зование: Положное Конарной Панег Состая команарной Панег Состая команарной Панег Поведение Клавици Ептег Проверять Заполнение/етом Исполь зование Дла ГруппИЭ Автоматическое Сохранение 	У Вертински Катически маничентов Данчых8Н	кальная Авто Опсрыть Блокировать окно владель Переход по элементам ✓ Для элемента Не использовать
реклика: а: нество: та рожления: л: Кон прожонания: Добавить 🔊 🕱 🍲 🗢 Организация: Дата н	начала работы Дата о 4 начала работы Дата о 4		 Склювные: Основные: Склювные: Скловные: Скловные: Скловные: Разрешить изменять форму Группировка Шархиза Подчиненных/Элемиен Истользоваание: Положение Командера/Панег Состав командера/Панег Состав командера/Панег Состав командера/Панег Состав командера/Панег Поведение Клавици/Ептег Проверять Заполнение Астом Истользование Для/ГруппИЭ Автоматическов/Соранение, Согранение Для/ГруппИЭ 	С С С С С С С С С С С С С С С С С С С	Кальная Авто Опсрыть Блокировать окно владель Переход по элементам Для элемента Не использовать Не использовать
реклика: в: на рождення: to прохожания:) Добавиль 22 12 Ф Ф Организация Дата н	начала роботы Дата о(А начала роботы Дата о(А начала роботы Дата о(А		21 (43) Во Х Х Основные: Заголовок Доступность Разрешить изменять Форму Группировка Шисина Подиниенньо/Элемен • Использование Положение Команда Положение К	С С С С С С С С С С С С С С С С С С С	Кальная Авто Опсрыть Блокировать окно владель Переход по элементам ✓ Для элемента Не использовать Не использовать

Рисунок2.15. Окно редактирования формы элемента справочника

На самом деле, это окно объединяет в себе несколько редакторов и окон. В частности, это следующие:

Редактор элементов формы (закладка Элементы в верхней левой части окна) – с его помощью можно контролировать элементы управления, которые будут расположены на форме. Выделив элемент в данном окне, мы можем настраивать его свойства в стандартной палитре свойств. Обратите внимание на кнопку Проверить, находящуюся в правой части командной панели закладки Элементы. Нажатие на нее приводит к выводу конструируемой формы в интерактивном виде, что позволяет лучше оценить ее внешний вид в пользовательском режиме, но, конечно, не дает возможности работать с даннымиинформационной базы.

Окно просмотра формы (закладка Форма в нижней части окна) – здесь представлена форма в том виде, который она примет после настроек. Кроме того, выделяя элементы формы в данном окне, мы, не имея возможности, как это было ранее, произвольно перемещать их, можем вызывать их контекстное *меню*, рисунок2.16, с помощью которого можно перемещать элемент вверх или вниз (то же самое можно делать в окне Элементы), открывать окно его свойств, назначать обработчики событий (их можно назначать и в окне Свойства, открытом для данного элемента).

	330)					
	А Рекризит		1.073	Использовать	Two	
			-	всегда		
	😔 = Объекс	🛛 = Объект			(СправочникОбъект. Физичес.	
903/14/6		🐵 🚥 Ссылка			СправочникСсылка. Физические П.	
			.0	-	Строка	
		ленование		1	Строка	
				1	Булево	
	- Dpe	илределенный		1	Булево	
— Пол — РайонПрокизания			8		Строка	
				1	Строка	
	— Отче	C180	8		Строка	
	т — Дал	Рождения		~	Дата	
	Rentwork	Karnen	110	Department	11149/1914	
🗶 Yas	сить	Del Oxf+Shift+Up				
Ф Пер	PMACTUTE RHMS					
		Ovf+Shift+Down				
— Пок	азать конзнаную пане	Ovl+Shift+Down				
Пок	азать командную пане: азать контекстное мен	Od+Shift+Down te				
Пок	азать конандную пане: азать контекстное мен ытия	Odf+Shift+Down to c		Прийаменении		
Пок Пок Соб	азать контекстное мен азать контекстное мен ытия жёли	Oct+Shift+Down	4	ПриИзменении: Начало Быбора>		
Пок Пок Соб Пер	азать кончендную пане: азать контекстное мен ылия истра Фстра	Od+Shift+Down te 6 512 /At+Enter		При/Изменении: НечалоВыбора> НечалоВыбора//	астиска>	
Tos Tos Colo Tep Caso	азать кончендиую пане азать контекстное мен ытия жейти Аства	to to 6 F12 Att+Enter		ПриИзменении (НачалоВыбора) :НачалоВибора)/ :Очистса>	аСписка	
Tos Tos Colo Tep Caso	азать кончендиую пане азать контекстное мен ытия жейти Аства	Dd+Shift+Down to 6 F12 Alt+Ention		ПриИзменении: Начало Выбора> Начало ВыбораИ Очистка> Регулирование>	аСписка>	
Пок Пок Соб Пер	азать кончендиую пане азать контекстное мен ыпия жейти Аства	Dd+Shift+Down te 6 F12 Att+Ention		ПрыИаменения: (НачалоВыбора) (НачалоВыбора) (Очистка) (Регулирсеание) (Открытие)	аСписка>	
Code Code Rep	азать коняндную пане. казать контекстное мен ылия жейти Астра	Dd+Shift+Down te 6 F12 Att+Enter		ПрыИаменения: (НачалоВыбора) (НачалоВыбора) Очистка) (Регулирование) Опералов) Обработся Былбо	аСписка>	
Tox Tox Cod Pep Cao	азать коняндиую пане казать контекстное мен ылия мотра фотра	Dd+Shift+Down te 6 F12 Att+Enter	- - - - - -	ПриИаменении: (НачалоВыбора) (НачалоВыбора) (НачалоВыбора) (Рагулирование) (ОбработкаВыбо (ОбработкаВыбо	IsChecka>	
	wactowa -			В В	Полование Испольание • • •	

Рисунок2.16. Работа с элементами формы

Редактор реквизитов представлен вкладкой Реквизиты (рисунок2.16.). Для того, чтобы добавить *реквизит* объекта на форму (то есть – создать элемент управления, связанный с данным реквизитом), достаточно перетащить элемент из окна Реквизиты в окноЭлементы. Реквизиты, уже присутствующие на форме, отмечены серым квадратиком.

Редактор команд можно открыть, нажав на вкладку **Команды**. Здесь доступны три дополнительные вкладки. Вкладка **Команды формы** (*по* умолчанию пустая) содержит команды формы, их можно сравнить с *командными кнопками*, которые в версии «1С:Предприятие» 8.1. можно было размещать на форме. Теперь последовательность действий выглядит так – сначала создать команду формы, потом перетащить ее в окно Элементы, настроить свойства, задать обработчики событий. Вкладка **Стандартные команды** (рисунок2.17) содержит стандартный набор команд – в нашем случае – стандартный для формы и табличного поля, размещенного на форме.

🖀 Справочник ФизическиеЛица: ФормаЭлемента	_ 0 1
 Форма Код Код Код Код Фамилия Фамилия Фликов Отчество ДазаРождения Пол РабонПрокиезания ТрудоваяИстория Коланциая панель ТрудоваяИсторияЮстанизация ТрудоваяИсторияДатаНачалаРаботы ТрудоваяИсторияДатаОкончанияРаботы 	 Форма Записать и закрыть Записать Записать Скопировать Понетить на удаление / Счять пометку Удалить Соправная Изменить форму Закрыть Да Нет ОК Отмена Прервать Поргустить Повторить Восстановить параметры Восстановить параметры Восстановить параметры Дабазить Окончровать Изменить Доскопировать Изменить Дабазить Доказить Дабазить Доказить Доказить Дабазить Доказить Дакнить Законнить редактирование Набт
🗧 Элементы 🖉 Командный интерфейс	Комалды едины Стандартные команды Тлосальные команды Команды Тараметры
Harmenoanse	
чамилия: Имя: Отчество: Дете рождения:	
Pañon npoxoleanais:	

Рисунок2.17. Стандартные команды

Вкладка **Глобальные команды** содержит набор команд уровня *прикладного решения*.

Вкладка Параметры предоставляет доступ к редактору параметров.

Вкладка Командный интерфейс позволяет редактировать командный интерфейс.

Реализуем автоматическое заполнение поля **Наименование** на основе полей **Фамилия**, **Имя** и **Отчество**.

Для этого сначала настроим элемент управления, отображающий наименование на форме, таким образом, чтобы его нельзя было редактировать. Выделим элемент управления В панели Элементы, откроем окно его свойств И установим свойствоТолькоПросмотр, Рисунок2.18. Благодаря этому свойству пользователь не сможет отредактировать текст в поле ввода. Похожего эффекта можно достичь и другими например, указав свойстве Вид способами. в элемента элемента Наименование вместо Поле ввода – Поле надписи.

🖉 🗶 🚖 🚭 🖾	18 18 0 ×					
Форма Ш Командиая панель	А Реквизит	Тип	ίνη.			
- Koa (Ceoñcrea: None 21 🛐 🏵 🔀 🗡	ылка Районы бличнаяЧасть.Ф				
 Ина Отноство ДатаРождения 	Основные: Имя Заголовок	Наименование			-	
 Пол РайонПроходания Законстра Комализинай интератор 	Вид ПутьКДанным	Поле ввода 💌				
Записаль и закрыль 🔄 Все дейсті бод:	Положение Заголовка Видимость Лользовательская видимость Доступность	Авто Опкрыть		×		
Taxinehosanie	ТолькоПросмотр ПропускатьПриВаоде	Aaro				
лия	АктивизироватьПоУмолчанию Маска РекитеПарова			12		
Пата рождения:	МногострочныйРежим РасширенноеРедактирование	Авто Авто Авто		-	*	
айон прохивания:	Голько просмотр Голько Просмотр, ReadOnly					1

Рисунок2.18. Настройка элемента Наименование

Для правильного формирования наименования важно, чтобы *пользователь* ввел данные в поля **Фамилия**, **Имя** и **Отчество**.

Клиентские методы в модуле формы

Теперь перейдем к написанию кода, в котором будем формировать наименование. Для этого нам нужно понимать, что конфигурации «1С:Предприятие» управляются событиями – и сейчас нас интересуют события формы.

Выделим форму в окне Элементы, откроем окно ее свойств и рассмотрим группу свойств **События**, рисунок2.19.. Наименование должно быть сформировано до того, как данные объекта будут записаны. Для достижения нашей цели нам вполне подойдет событие **ПередЗаписью**. Здесь же можно выполнить какие-либо пользовательские проверки полей перед формированием наименования. Хотя, если говорить о производительности решения, лучше подобные проверки производить на сервере, например, с помощью обработчика события **ОбработкаПроверкиЗаполнения**, который создается в модуле объекта.

Справочник Физические/Тица: ФормаЗлемента	Свойства: Форма	×	×
0 / x + + -	HIN X V		1
Форма		12	- A
ша Командная панель	УсловноеОформление Откоълъ		
— Код	• Расположение:		кОбъект. Физич
- Наименование	Ширина 0 С Высота	0 2	Сылка. Физически
- Фамилия	• События:		
— Имя	ПриСозданииНаСервере	* Q	
- Отчество	ПриОткрытии	* Q	
- ДатаРождения	ПриПовторномОткрытии	+ 9	
— Пол	ПередЗакрытием	- Q	
РайонПроживания	ПоиЗакоылии	- 9	-
🚆 Элементы 📲 Командный интерфейс	ОбработкаВыбора	- 0 -	
Записать и закрыть 📄 Во	е дей ОбработкаОпорешения	• 0	
	Обработка и кривнова на	- 0	E E
Код	Обработка Записи Нового	- 0	
Наименование		- 0	
Фамилия:	Питина Запачсько	- 0	
14 mil	Denes 2 anun in HaCensena	- 0	
VMR.		- 0	
Отчество:	Deces 3 an under Campage	- 0	
Дата рождения:		-0	-
Bon	Official and a second a	- 0	
	Copada ikai gobecku sa taimenu inacet	-0	
Район проживания:		14	
😳 Добавить 📄 💥 🍲 🗣	Вс Присохранении/ДанныхВнастройкахиз	*4	1
Prova Doma	Гередзагрузкояданныхизнастроекн	+4	
	Призагрузкеданныхизнастроекнасе	-9	
	Вызывается перед записью объекта на клиен ПередЗаписью, BeforeWrite	пте. Можно отменить запись	

Рисунок 2.19. Выбор события для выполнения запланированных действий

Нажмем на кнопку с увеличительным стеклом в *поле* события **ПередЗаписью** – автоматически будет открыт *модуль* формы и создан пустой обработчик события **ПередЗаписью**. Он имеет следующий вид:

&НаКлиенте

Процедура ПередЗаписью(Отказ, ПараметрыЗаписи)

// Вставить содержимое обработчика.

КонецПроцедуры>

Из директивы компиляции &НаКлиенте понятно, что процедура это клиентская, она имеет два параметра – нас сейчас интересует параметр Отказ – благодаря этому параметру, а именно, установив его в значение Истина, мы можем отказаться от записи объекта в том случае, если выполняется какое-либо условие, препятствующее записи. В нашем случае записи объекта могут воспрепятствовать незаполненные или неправильно поля Фамилия, Имя или Отчество. Проверку заполненные на незаполненность можем доверить и системе – для этого можно реквизита МЫ **VCTAHOBUT**Ь свойство Проверка заполнения для нужных реквизитов в значение Выдавать ошибку, делается это в списке реквизитов объекта в окне редактирования объекта или в дереве конфигурации, рисунок2.20. Не будем включать проверку заполнения, выполним ее и еще некоторые проверки самостоятельно.

Конфигурация *	×	Свойства: Фамилия		9	×
Действия • 🚱 🖉 🔂 🗶 🍲 🗣 📃 🌾		11 1 1 X V			
 Э Общие Э Константы Э Справочники Э Сотрудники Э Организации Э Физические Гица Э Реквизиты 	*	Индексировать Использование Комментарий Маска Многострочный режим Неограниченная длина Подсказка	Не индексировать Для элемента	•	
 Имя Отчество ДатаРождения Пол РайонПроживания 	•	Полнотекстовый поиск Проверка заполнения Расширенное редактирование Режим пароля Свазь по типи	Использовать Выдавать ошибку		a strange of the
		Синоним	Фамилия Строка	·······	
			(a the sum		

Рисунок 2.20. Настройка проверки заполнения

Параметры и процедуры в системе «1С:Предприятие» *по* умолчанию передаются *по* ссылке – передав в процедуру некую переменную, мы, на самом деле, передаем ссылку на нее, то есть – при модификации соответствующего этой переменной параметра внутри процедуры, фактически, происходит и модификация переменной. Вернемся к нашей процедуре**ПередЗаписью**.

В этой процедуре мы сначала проверим поля **Фамилия**, **Имя** и **Отчество** на заполненность (возможны и более сложные проверки), после чего, если хотя бы одно *поле* не заполнено – сообщим об этом пользователю и выйдем из процедуры, если все поля заполнены – сформируем наименование. Вот какой код позволяет реализовать эту задачу:

&НаКлиенте

Процедура ПередЗаписью(Отказ, ПараметрыЗаписи) //Переменная для хранения текста сообщения пользователю ПеремТекстСообщения; //Запишем пустую строку в переменную ТекстСообщения=""; //Если не введена фамилия... Если ПустаяСтрока(Объект.Фамилия) Тогда //Формируем строку сообщения ТекстСообщения=ТекстСообщения+"Не заполнено поле Фамилия;"; КонецЕсли; //Если не введено имя... Если ПустаяСтрока(Объект.Имя) Тогда ТекстСообщения=ТекстСообщения+" Не заполнено поле Имя;"; КонепЕсли. //Если не введено отчество... Если ПустаяСтрока(Объект.Отчество) Тогда ТекстСообщения=ТекстСообщения+" Не заполнено поле Отчество;"; КонепЕсли: //Если строка сообщения не пуста, то есть - содержит //сообщения о незаполненных полях

Если НЕ ПустаяСтрока(ТекстСообщения) Тогда //Выводим сообщение Сообщить(ТекстСообщения); //Отказываемся от записи объекта Отказ=Истина; //Выходим из процедуры Возврат; КонецЕсли; //Если все поля заполнены, выхода из процедуры не произошло, //формируем наименование Объект.Наименование=Объект.Фамилия+" "+ ВРег(Лев(Объект.Имя,1))+". "+ВРег(Лев(Объект.Отчество,1))+".";

КонецПроцедуры

Строковая функция Лев позволяет получить заданное количество символов из строки, начиная с самого левого. Строковая функция ВРег переводит символы в верхний perucmp – на тот случай, если пользователь случайно ввел имя, фамилию или отчество с маленькой буквы. Конечно, здесь можно предусмотреть еще множество проверок и автоматических корректировок (например, можно исправить первую букву во введенных фамилии, имени и отчестве, если она случайно введена в нижнем регистре), мы ограничимся тем, что сделано сейчас.

В итоге мы получаем следующие сообщения об ошибках при незаполненности полей, рисунок2.21.

Ризически	е лица (создание)*		Сообщения	×
Записать и за	крыть	Все дейст	твия - 🧿	😃 Не заполнено поле О	тчество;
од:					
аименование:	[
Рамилия:	Петров				
мя:	Иван				
тчество:	1				
lата рождения:					
lon:				10 m	
айон проживани	R		Q		
🕃 Добавить 📘	8 × 4 4	Boe	действия *		
N Организ	ыция	Дата начала работы	Дата окон		

Рисунок2.21. Сообщение об ошибке

После успешного выполнения процедуры **ПередЗаписью**, наименование выглядит следующим образом. Мы намеренно ввели отчество с маленькой буквы – как было пояснено выше, наш код готов к такому повороту событий, рисунок2.22.

Рабочий Булгалтерский Учи стол учет ма	ет работы астеров	Оперативный учет материалов	Pa	очет ной платы		SHURE		
Районы Сотрудники Ризические лица	(ЭСоздать 🔂 🖉	ица 😰 🍳	Наяти 🎘	1	1	Boe	действия • (
		— Петров И. А.	A	Код 000000001	Фамилия Петров	Иван	александрович	
	e.							

Рисунок2.22. Новая запись в справочнике Физические лица

Объект СообщениеПользователю

Обратите внимание на то, что здесь мы пользуемся обычным методом Сообщить – мы выводим в окно сообщения одно сообщение, содержащее необходимые сведения. В «1С:Предприятие» 8.2. мы можем поступить *по*-другому – вывести сообщения об ошибках или другие сведения, "привязав" их к полям, которые вызвали ошибки. Для этого можно воспользоваться объектом Сообщение Пользователю. Он, помимо прочих полезных возможностей, позволяет формировать сообщения и "привязывать" их к реквизитам формы. Перепишем код таким образом, чтобы сообщения об ошибках (то есть, о незаполненных полях Фамилия, Имя, или Отчество), выявленных в процедуре ПередЗаписью, выводились бы в привязке к соответствующим элементам формы. Вот какой код позволяет этого добиться:

&НаКлиенте

Процедура ПередЗаписью(Отказ, ПараметрыЗаписи)

//Если не введена фамилия...

Если ПустаяСтрока(Объект.Фамилия) Тогда

СообщитьПользователю("Объект.Фамилия", "Заполните поле Фамилия", Отказ); КонецЕсли;

//Если не введено имя...

Если ПустаяСтрока(Объект.Имя) Тогда

СообщитьПользователю("Объект.Имя", "Заполните поле Имя", Отказ);

КонецЕсли;

//Если не введено отчество...

Если ПустаяСтрока(Объект.Отчество) Тогда

СообщитьПользователю("Объект.Отчество", "Заполните поле Отчество", Отказ); КонецЕсли;

//Если флаг Отказ не был установлен - формируем наименование Если НЕ Отказ Тогда

Объект.Наименование=Объект.Фамилия+" "+ ВРег(Лев(Объект.Имя,1))+". "+ВРег(Лев(Объект.Отчество,1))+".";

КонецЕсли;

КонецПроцедуры

&НаКлиенте

//Процедура, формирующая и выводящая сообщение с переданными ей параметрами

Процедура Сообщить Пользователю (Путь КРеквизиту, Текст, Отказ)

Сообщение=Новый СообщениеПользователю;

Сообщение.Поле=ПутьКРеквизиту;

Сообщение.Текст=Текст;

Сообщение.Сообщить();

Отказ=Истина;

КонецПроцедуры

Поясним приведенный код. Для начала, мы создали новую клиентскую процедуру СообщитьПользователю. Эта процедура принимает три параметра. Первый – ПутьКРеквизиту содержит строковый путь к полю, к которому должно быть привязано сообщение. Второй – Текст – содержит текст сообщения. Третий – Отказ – используется для установки в значение Истинапараметра Отказ процедуры ПередЗаписью в том случае, если процедура СообщитьПользователю будет вызвана хотя бы один раз. А хотя бы однократный ее вызов означает, что одно из полей не заполнено, то есть наименование сформировать невозможно, соответственно, записать объект так же не получится.

Когда процедура вызывается, ΜЫ сначала создаем новый объект типа СообщениеПользователю. Затем его свойство Полеустанавливаем в значение параметра ПутьКРеквизиту. Этот *параметр* должен быть строковым и имеет, в нашем случае вид "Объект.Фамилия", "Объект.Имя", "Объект.Отчество" - это "привязать" сообщение позволяет правильно к полям формы. СвойствоТекст объекта СообщениеПользователю содержит текст для вывода.

Мы, кроме того, полностью переработали процедуру **ПередЗаписью**. А именно, если проверка на заполнение поля указывает на то, что *поле* пустое, вызывается процедура **СообщитьПользователю**. *По* окончании проверок мы проверяем, установлен ли*параметр* **Отказ** в *значение* **Истина** – если не установлен – ни одна из проверок не завершилась обнаружением пустого поля, и мы можем формировать наименование. Если установлен – наименование мы не формируем – и процедура заканчивает работу, а записи объекта, естественно, не происходит – *пользователь* видит лишь сообщения об ошибках.

Если было сформировано несколько сообщений типа СообщениеПользователю – *пользователь* видит одно окно сообщения около поля, но это окно снабжено кнопками для перемещения вперед и назад – щелчок *по* кнопке приводит к "переходу" сообщения от одного поля с ошибкой к другому, рисунок2.23, 2.24.

φν	изические лиц	ца (создание) * (1С:Предг	приятие)		M III M	M+ M- D >
Физі Запи Код: Наиме	ические кать и закр	лица (создание внь) 🔚	е) * Все дейс	твия + 🧿	Сообщения (1) Заполните поле Имя (1) Заполните поле Отчес	тво.
Фамил Имя: Отчест Дата р Пол: Район	лия: тво: рождения: проживания:	Иванов Ошибка: Заполните поле И	MR			
Ода	обавить 🔝	X & #	Bce	действия •		
Ν	Организац	ия	Дата начала работы	Дата окон		

Рисунок 2.23. Сообщение об ошибке, привязанное к полю Имя

Записать и закрыть Все действия • 2 Заполните поле Имя Код: Наименование: Фамилия: Иванов	190
Фамилия: Иванов	
/ms:	
Отчество: Дата рождения: Пол: Район проживания:	
🚱 Добавить 🐘 🗶 🍲 😻 Все действия •	
N Организация Дата начала работы Дата окон	

Рисунок2.24. Сообщение об ошибке, привязанное к полю Отчество

Доведем до логического завершения пример со справочником **ФизическиеЛица**. Для этого заполним справочник Районы и введем в *информационную базу* сведения о следующих физических лицах (таблица 2.1).

Фамилия	Имя	Отчество	Дата рождения	Пол	Район
Иванов	Иван	Иванович	27.02.1984	Мужской	Ленинский
Петров	Петр	Петрович	12.06.1985	Мужской	Ленинский
Васильев	Павел	Петрович	17.05.1985	Мужской	Ленинский
Расчетчиков	Александр	Иванович	12.03.1980	Мужской	Октябрьский
Александров	Александр	Александрович	17.09.1970	Мужской	Октябрьский
Бухгалтерова	Василиса	Владимировна	13.08.1976	Женский	Уральский

Таблица 2.1 – Физические лица

Обратите внимание на то, что справочник Физические.Лица – это пример справочника, с которым пользователям нашейинформационной базы придется работать достаточно часто. В данный момент для того, чтобы создать новый элемент справочника, нам нужно выполнить несколько действий – перейти в раздел Расчет заработной платы, открывающей список справочника, щелкнуть по ссылке, после чего нажать на кнопку Создать новый элемент списка. Для того, чтобы сократить количество действий, часто используемых необходимых лля выполнения операций. ΜЫ можем настроить интерфейс нашего прикладного соответствующим образом решения, В частности, поработать с панелью действий соответствующего раздела и сРабочим столом.

Настройка командного интерфейса для ускорения доступа к справочнику

Добавим команду создания нового элемента справочника Физические.Лица в панель действий раздела УчетРаботыМастеров. Для этого откроем окно Все подсистемы командой контекстного *меню* ветви Подсистемы *дерева конфигурации* и установим флаг Видимость напротив команды Физические лица: Создать в области Панель действий.Создать, рисунок2.25.

-		^
Подсистемы	Командный интерфейс	
32 🖉 🗙 🐨 🦊 🍼	🚰 🐨 🤟 🚱 👘 Отбор по ролям: (<Не установлен	H> 💌
💱 Оперативный Учет Материалов	Команда Видимость Видимо	сть по ролям
Администрирование	Администратор	Сотрудник
🚱 УчетРаботыМастеров	🥅 Панель навигации.Важное	
	🖂 😐 Панель навигации.Обычное	
	🚻 Подразделения 🔽	\checkmark
	🎹 Районы 🕑	\checkmark
	🎹 Сотрудники 🔽	\checkmark
	🎹 Физические лица 🕑 🕑	\checkmark
	 Панель навигации.См. также 	
	🗢 뉩 Панель действий.Создать	
Comm	Подразделения: создать	
	П Районы: создать	
🕀 🔡 Константы	Сотрудники: создать	
① Правочники	🎹 Физические лица: созд 🔽	
	🔟 Панель действий.Отчеты	
	Панель действий.Сервис	_
	🔡 Текст сообщения 🔽 🗹	\checkmark

Рисунок2.25. Настройка панели действий раздела Расчет заработной платы

Мы можем включить команду добавления нового физического лица в командный *интерфейс* Рабочего стола (рисунок 2.26).

Для этого выполним команду контекстного *меню корневого* элемента конфигурации **Открыть командный интерфейс рабочего стола**

Выделим в *поле* Доступные команды команду Физические лица: создать, в *поле* состава командного интерфейса – команду Панель действий.Создать и нажмем на кнопку со значком ">" (Добавить команду на *рабочий стол*), которая находится между полями, после чего установим флаг Видимость для добавленной команды, рисунок 3.25.



Рисунок 2.26. Настройка панели действий Рабочего стола

Теперь, рисунок2.27, *команда* для быстрого создания элементов справочника **ФизическиеЛица** добавлена в панель действий**Рабочего стола**.

TB	слияние		Составлен	ие документа и вставка по/	ей Таросмотр результатов завершить			
6		👻 😭 Салон	н красоты (1С:Пред	приятие, учебная версия)	🖧 🗋 🖩 🖬	M M+ M- (i)	□ × .	1
ć	4	\bigcirc	0	0				
Ę	Рабочий	Учет работы	Оперативный	Администрирование				
	Создать Физические л	ица	учет материалов					
						Сообщения	×	
1 C						Здравствуйте, уважаемый пользователь!	er	
- 180							<u>(</u> +	

Рисунок 2.27. Новая команда в панели действий рабочего стола

Литература:

Официальный сайт интернет-униерситета «Интуит» (электронный ресурс), режим доступа http://www.intuit.ru/studies/courses/2318/618/lecture/17939.

Контрольные вопросы для самопроверки

- 4. Как создаются справочники?
- 5. Виды справочников в «1С:Предприятие»?
- 6. Как настроить интерфейс рабочего стола?

Задание к лабораторной работе

Создать справочники в системе в соответствии с порядком, отраженным в теоретической части

Методические указания и порядок выполнения работы

Работа выполняется в точном порядке, как это указано на рисунках и в теоретической части.

Индивидуальное задание

не предусмотрено

Требования к отчету и защите

1. Результатом выполнения лабораторной работы является сформированный в программе файл, содержащий выполненные задания. В ЭИОС результаты работы не выкладываются.

2. Планируется защита работы, где студент комментирует порядок выполнения заданий, а также отвечает на вопросы, представленные выше.

ЛАБОРАТОРНАЯ РАБОТА №3

О РАЗЛИЧНЫХ ВИДАХ СПРАВОЧНИКОВ

ОБЩИЕ СВЕДЕНИЯ

Цель: научиться информационную базу на платформе ««1С:Предприятие»»

Материалы, оборудование, программное обеспечение: 1. персональный компьютер (компьютерные классы ГУК), 2. программное обеспечение «1С:Предприятие».

Условия допуска к выполнению: умение работать на ПК и знание техники безопасности.

Критерии положительной оценки: предоставление результатов работы в виде файла и прохождение защиты.

Планируемое время выполнения: Аудиторное время выполнения (под руководством преподавателя): 2 ч. Время самостоятельной подготовки: 2 ч.

Теоретическое введение

Многие справочники, используемые на практике, являются иерархическими. Это означает, что каждому из элементов справочника может быть сопоставлен другой элемент, который называется родителем.

Возможна *иерархия* различных видов, в частности, при иерархии групп и элементов в справочник, помимо обычных элементов можно включать группы, которые, в свою *очередь*, могут включать в себя другие группы и элементы.

Примером такого справочника является справочник, хранящий списки товаров, материалов, услуг. Обычно такой справочник носит название Номенклатура. Как правило, подобный справочник имеет множество групп – например – Товары – для хранения записей о товарно-материальных ценностях, и Услуги – для хранения списков услуг.

Второй вид иерархии – это *иерархия* элементов. Хорошим примером справочника, для которого естественна *иерархия* элементов, является справочник для хранения сведений о подразделениях организаций. Такой справочник обычно так и называется – *Подразделения*. Логика иерархии элементов заключается в том, что отдельные элементы справочника, описывающие отдельные*подразделения* организации, могут являться родителями для других элементов – также подразделений. Например, подразделение Администрация вполне может включать в себя *подразделения* Бухгалтерия, Отдел кадров и так далее.

Справочники, в сущности, являются хранилищами аналитических признаков учета. Например, в случае со справочником Номенклатура можно организовать учет таким образом, чтобы иметь сведения об остатках каких-либо номенклатурных позиций на складе, об их стоимости. В случае со справочником, поддерживающим иерархию элементов, любой элемент справочника можно использовать как аналитический "разделитель" учета (то есть, например, выбирать данные элементы при заполнении табличных частей других объектов, например, документов). А вот при иерархии групп и элементов использовать в качестве аналитического разделителя группу не получится.

Если родительские отношения существуют внутри справочника, такие справочники называются иерархическими. Существует еще один вид взаимоотношений между справочниками, который называется подчинением. Предположим, у нас имеется

справочник Контрагенты, содержащий *список* организаций, с которыми наша организация имеет какие-то взаимоотношения. Каждый из контрагентов, описанных в этом справочнике, имеет некоторое количество контактных лиц – представителей контрагента. Можно сказать, что каждый представитель контрагента "принадлежит" определенному контрагенту. Такие взаимоотношения между справочниками реализуются при помощи механизма указания владельцев справочника и настройки подчинения.

Рассмотрим работу с иерархическими справочниками.

Иерархические справочники

Создадим новый справочник Единицы измерения, зададим следующие его параметры:

Имя: ЕдиницыИзмерения

Длина наименования: 100 символов

Подсистемы: Оперативный Учет Материалов

Это будет очень простой справочник, стандартный *реквизит* которого **Наименование** будет использоваться для хранения информации о наименовании единицы измерения.

Теперь создадим очередной справочник – Номенклатура. Зададим следующие параметры:

Имя: Номенклатура

Подсистемы: БухгалтерскийУчет, ОперативныйУчетМатериалов

На вкладке окна редактирования объекта **Иерархия**, рисунок 4.1, установим следующие параметры:

Основные	Иерархический справочник	
Подсистемы	Вид мераронии:	C
Функциональные опции		
Иерархия		6
Владельцы	Размещать группы сверху	
Данные	Ограничение количества уровней иерархии	
Нумерация	количество уровней иерархии	2 -
Формы		
Команды		
Макеты		
Ввод на основании		
Ввод на основании Права		
Ввод на основании Права Обмен данными		
Ввод на основании Права Обмен данными Прочее		
Ввод на основании Права Обмен данными Прочее		

Рисунок 3.1. Настройка иерархического справочника

Иерархический справочник: Установлено

Вид иерархии: Иерархия групп и элементов (рисунок3.1).

Этот *параметр* может принимать *значение* **Иерархия** элементов. В нашем случае справочник сможет содержать отдельные элементы, собранные, в зависимости от их вида, в группы. Эту структуру можно сравнить с папками и файлами в файловой системе компьютера. Группы – это папки, отдельные элементы – это файлы.

На вкладке Данные, рисунок3.2, добавим следующие реквизиты:

ЕдиницаИзмерения: Тип СправочникСсылка.ЕдиницыИзмерения.

Услуга: Тип Булево, Использование: Для группы и элемента. Эта установка позволит задавать данный *реквизит* и для элементов и для групп.

Заполнять из данных заполнения: Истина

Отдельные группы нашего справочника планируется использовать для хранения исключительно услуг, и подобная установка (в частности, истинность параметра Заполнять из данных заполнения) позволит нам реализовать автоматический механизм заполнения данного реквизита для элементов, входящих в группы.

Справочник Номенклатур	a a	_ 🗆 ×	Свойства: Услуга		• >
Основные Подсистемы Функциональные опции Иерариия Владельцы Данные Цанные Нумерация Формы Команды Макеты	Длина кода Длина наименования О Число © Строка © Ф Ж 🌚 Ф О 🦇 Рекризиты — Единица — Услуга	9: 100; Овиде кода • В виде наименования	Сизовище: Имя Услуга Сизовище: Имя Услуга Сизовим Услуга Комментарий Тип Булево • Использование: Использование: Использование: Использование: Использование: Использование: Использование: Использование: Использование: Использование: Использование: Использование: Использование: Использование: Использование: Использование: Использование: Использование:	Для группы и злемента Не индексировать к Использовать	
Ввод на основании Права Обиек данными Прочее	∰ № / × е Ш Табличные ч	* E 8074	Формат Формат редактирован Подоказка	RIE	
Действия +)	Стандартные рекви: Общие реквизит «Назад Далее»	илы Характеристики н Закрыть Отравка	Заполнять из данных Эначение заполнения Проверка заполнения Использование реквизи	заполнения 🕑 Ложь Не проверять кта для элементов и прупп элемент	

Рисунок 3.2. Состав реквизитов справочника Номенклатура

Таким образом, при создании элемента справочника мы будем задавать название элемента в стандартном *реквизите***Наименование**, указывать единицу измерения, а так же, для услуг, устанавливать флаг **Услуга**, причем, установка этого флага для группы будет означать, что в ней хранятся списки услуг, а для элемента – то, что он является услугой.

Автозаполнение реквизитов

Реализуем функцию автоматического заполнения реквизита Услуга для элементов, входящих в группы. Нам нужно, чтобы элемент, создаваемый в группе с установленным флагом Услуга, при его создании, автоматически бы получал установленный флаг Услуга, соответственно, если данный флаг у группы не установлен, у элемента он так же не должен быть установлен. При этом нам нужно предусмотреть ситуацию, когда элемент создается вне группы – на верхнем уровне справочника Номенклатура. Для решения этой задачи мы можем воспользоваться обработчиком события ОбработкаЗаполнения, его процедура располагается в модуле объекта.

Перейдем в *модуль* объекта (кнопка **Модуль** объекта на закладке **Прочие** окна редактирования объекта), из списка процедур и выберем **ОбработкаЗаполнения**, рисунок3.3.



Рисунок3.3. Процедура обработки заполнения справочника

Процедура будет исполняться на стороне сервера, причем, вызываться она будет при различных способах создания элемента справочника – например, при интерактивном создании пользователем, при копировании, при программном создании. Параметрпроцедуры Стандартная Обработка позволяет включать или обработку отключать стандартную процесса заполнения реквизитов, параметр Данные Заполнения содержит данные, которые система использует для заполнения элемента.

В режиме «1С:Предприятие» откроем справочник Номенклатура, создадим две группы – Товары – флаг Услуги в этой группе не устанавливаем, и Услуги – флаг установлен, рисунок3.4.



Рисунок 3.4. Две группы в справочнике Номенклатура

Исследуем процедуру обработки заполнения, прежде чем продолжать работу над ней. Для этого вставим в нее какую-нибудь команду, например: **ТестоваяПеременная** = **0**; и установим на строку с данной командой точку останова. Для этого либо выполним *двойной щелчок* левой кнопкой мыши на сером *поле* слева от команды, либо, установив *курсор* в строку с командой, выполним команду **Отладка** > **Точка останова**, либо – установив *курсор* в нужную строку, нажмем **F9**. Для установки точки останова нужно, чтобы строка, на которую мы пытаемся ее установить, присутствовала в *конфигурации базы данных*, т е. написав код, нужно нажать на кнопку **Обновить конфигурацию** *базы данных*. В итоге у нас должно получиться следующее, рисунок3.5.



Рисунок3.5. Точка останова в процедуре

Запустим конфигурацию в режиме отладки (кнопка Начать отладку, команда меню Отладка > Начать отладку, или клавиша F5на клавиатуре). Перейдем в группу Услуги и создадим в ней новый элемент. Когда управление будет передано в Конфигуратор, установим курсор на имя параметра ДанныеЗаполнения, вызовем контекстное меню и выберем команду Вычислить выражение. Появится окно Выражение, из которого можно понять, что переменная ДанныеЗаполнения – это структура, в которой присутствуют сведения о родителе создаваемого элемента – т. е. о группе Услуги, рисунок3.6.

выражени	e						
Выражение	c.			<u>Рассчитать</u>			
Данные За	анныеЗаполнения						
Результат:			0 🔂	DKINOWINE B TOOND			
Свойство		Значение	Тип	Закрыть			
• Даннь	еЗаполнения	Структура	Структура	Справка			
😑 Po	дитель	Услуги	СправочникСсылка.Номенклатура				
	ВерсияДанных	"AAAAAgAAAAA="	Строка				
	Владелец		Неопределено				
	ЕдиницаИзмере		Null				
	Код	"00000002"	Строка				
	Наименование	"Услуги"	Строка				
۲	Организация		СправочникСсылка.Организации				
	ПометкаУдаления	Ложь	Булево				
	Предопределенн	Ложь	Булево				
(\oplus)	Родитель		СправочникСсылка.Номенклатура				
\oplus	Ссылка	Услуги	СправочникСсылка.Номенклатура				
	Услуга	Истина	Булево				
	ЭтоГруппа	Истина	Булево				
			Subara internations				

Рисунок3.6. Структура ДанныеЗаполнения

Структура – это *таблица*, которая содержит пары вида Ключ – Значение.

В нашем случае, если процедура **ОбработкаЗаполнения** отработает – она заполнит лишь *поле* **Родитель** для создаваемого элемента. А нам хотелось бы установить и флаг **Услуга** в соответствии с данными родителя.

Рассмотрим некоторые составляющие данных, к которым мы можем получить *доступ* посредством структуры.

Родитель – здесь хранится родитель элемента – в нашем случае – *группа* Услуги типа СправочникСсылка.Номенклатура. То есть при заполнении поля Родитель создаваемого элемента окажется, что он будет хранить ссылку на другой элемент (в нашем случае – группу), входящий в справочник Номенклатура.

Владелец – данное *поле* у нашей группы, находящейся в справочнике, имеет *значение* **Неопределено**. Такое *значение* присваивается тем свойствам, которые, в принципе, могут быть установлены, но в данном случае значения не имеют.

ЕдиницаИзмерения имеет *значение* Null. При настройке состава реквизитов справочника Номенклатура, мы указали, чтоЕдиницаИзмерения может задаваться структуре справочника в информационной только для элемента. Ho В базе подобное полеприсутствует и у группы. Однако значения оно содержать не может – поэтому в качестве типа значения ΜЫ видим **Null**. Типы значений **Неопределено** и **Null** кажутся похожими, но это – разные вещи. Значение с типом **Неопределено** может быть задано, а значение **Null** не может быть задано в принципе.

Свойство **Услуга** установлено в *значение* **Истина** – этот флаг мы устанавливали при создании группы **Услуги**.

Свойство ЭтоГруппа так же истинно – оно устанавливается в истинность для групп.

Для того чтобы установить свойство **Услуга** у создаваемого элемента, мы могли бы напрямую обратиться к свойству элемента**Услуга** и установить его в *значение* флага Услуга у его родителя. Выглядеть это может, например, так:

Услуга = ДанныеЗаполнения.Родитель.Услуга.

Однако процедура ОбработкаЗаполнения предусматривает автоматический механизм заполнения реквизитов на основе переданной структуры. Так как стандартный механизм нас вполне устраивает, мы можем поступить *по*-другому. А именно, для установки свойства **Услуга** нам нужно лишь дополнить структуру необходимой записью. Сделать это можно с помощью стандартных операций *по* работе со структурой. А именно, следующим образом:

ДанныеЗаполнения.Вставить ("Услуга", ДанныеЗаполнения.Родитель.Услуга);

В итоге у нас получается такой код, рисунок3.7.



Рисунок3.7. Заполнение реквизита Услуга на основании параметров элемента родителя

Здесь мы установили точку останова для того, чтобы посмотреть, как изменится структура при выполнении данной процедуры. Опробуем решение в пользовательском режиме, можно заметить, что, во-первых, структура ДанныеЗаполнения действительно теперь содержит *ключ* Услуга со значением Истина, а также то, что элементы,

создаваемые в группе с установленным флагом **Услуга**, имеют данный *реквизит* в установленном положении, рисунок 3.8.

🕅 🔳 M M+ M- 🗆 🗙
Все действия 👻 ②
]
Q
۹
Q

Рисунок3.8. Результат заполнения реквизита Услуга на основании параметров элемента родителя

Подойдет ли *созданная процедура* для практического использования? На этот вопрос может ответить ее тестирование. А именно, попробуем создать еще одну группу в корневой части справочника. Очевидно, что у такой группы родителя не будет. То же самое касается создания элемента. Попытка приводит к появлению ошибки, рисунок3.9.

1С:Пред	приятие		×
\odot	Значение не является знач	ением объектного типа (Ро	одитель)
	OK		

Рисунок3.9. Ошибка при попытке создать элемент, у которого нет родителя

Нажав на кнопку **Подробно**, видим, что ошибка произошла при попытке добавить в структуру новую *запись*, рисунок3.10.

1C.lipearpustive	
Обнаружены ошибки: Одибка при выполнении обработчика - 'ОбработкаЗаполнения' по причине: Попратикне:	Закрыть Конфигуратор Завершить работу
(справочник.поленклатура.кодульосовская(э)): эначение не является значением осъектного типа (родите. ДанныеSаполнения.Вставить ("Услуга", ДанныеSаполнения.Родитель.Услура):	(b) (b)

Рисунок3.10. Более подробное описание ошибки

Нажав на кнопку Конфигуратор, мы попадаем в Конфигуратор. Прежде чем обращаться к элементу структуры Родитель, нужно убедиться в том, что Родитель в

структуре присутствует. Если **Родителя** нет – в структуру не нужно ничего добавлять, если есть – можно добавить. Попробуем такой код:

Если ДанныеЗаполнения.Свойство("Родитель") Тогда

ДанныеЗаполнения.Вставить("Услуга", ДанныеЗаполнения.Родитель.Услуга); КонецЕсли;

Как кажется, все должно работать правильно – если в структуре обнаружилось *поле* Родитель – мы можем обращаться к свойству Услуга. Но попытка выполнить эту процедуру снова приводит к ошибке

Эта ошибка возникает при проверке условия на наличие в структуре свойства Родитель. Здесь у нас возникает вопрос о том, чем является передаваемый *параметр* ДанныеЗаполнения при создании элемента или группы на верхнем уровне справочника. Для ответа на этот вопрос мы можем снова прибегнуть к отладке. Как видно, *значение* параметра неопределено, рисунок3.11.

Выражение			
<u>В</u> ыражение:			<u>Рассчитать</u>
Данные Заполнения			Включить в табло
Результат:			Закрыть
Свойство	Значение	Тип	
Данные Заполне	ния	Неопределено	Справка
L			

Рисунок3.11. Параметр ДанныеЗаполнения при создании элемента или группы на верхнем уровне справочника

Тип **Неопределено** говорит нам о том, что перед нами лишь *переменная*, тип которой программе не известен. Мы не можем обращаться к ней как к структуре, поэтому, прежде чем проверять, есть ли в структуре **ДанныеЗаполнения** свойство**Родитель**, нам нужно проверить, является ли передаваемый *параметр* **ДанныеЗаполнения** структурой. Мы знаем, что этот*параметр*, когда он заполнен данными, имеет тип **Структура**, следовательно, нам нужно исключить вариант, когда его тип равняется **Неопределено**.

Вышеприведенные рассуждения приводят нас к следующему коду:

Процедура ОбработкаЗаполнения (ДанныеЗаполнения, СтандартнаяОбработка)

Если ДанныеЗаполнения <> Неопределено Тогда

Если ДанныеЗаполнения.Свойство("Родитель") Тогда

ДанныеЗаполнения.Вставить("Услуга", ДанныеЗаполнения.Родитель.Услуга); КонецЕсли;

КонецЕсли;

КонецПроцедуры

В данной редакции обработчика события ОбработкаЗаполнения все работает верно.

Чисто теоретически (предположим, при изменении кем-либо нашего кода) возможна ситуация, когда ДанныеЗаполнения будут являться структурой и в этой структуре, в то же время, не будет свойства Родитель. Поэтому наряду с проверкой на неопределенность значения мы оставляем и проверку на наличие свойства Родитель.

Мы обсудили и проиллюстрировали родительские отношения в справочнике, рассмотрим теперь пример работы с *подчиненными справочниками*.

Подчиненные справочники

Создадим новый справочник, назовем его Контрагенты. Добавим его в подсистемы ОперативныйУчетМатериалов.

Справочник будет иерархическим, с иерархией групп и элементов. В состав реквизитов справочника добавим следующие (рисунок3.12.): Имя: ПолноеНаименование, тип – Строка, *длина* 100. Имя: КонтактныеСведения, тип – Строка, *длина* 100



Рисунок3.12. Справочник Контрагенты

Создадим еще один справочник. Назовем его ПредставителиКонтрагентов. Главная черта этого справочника – то, что он подчинен справочнику Контрагенты. Для настройки подчинения используется вкладка окна настройки объекта конфигурации Владельцы. Здесь мы должны добавить в Список владельцев справочника справочники-владельцы, в нашем случае – справочник Контрагенты. После того, как владелец добавлен в этот список, мы можем настроить для него параметр Использование подчинения. Здесь возможны три варианта:

Элементам – элементы *подчиненного справочника* подчинены элементам справочника-владельца. В нашем случае выбор этого параметра означает, что в качестве "владельца" представителя контрагента выступает сам *контрагент*.

Группам – подчинение группам справочника-владельца. Нас это не устроит – так как группы справочника будут содержать контрагентов, сгруппированных *по* достаточно общим показателям (Покупатели, Заказчики и т.д.), и сопоставление одного контактного лица нескольким разным, например, покупателям, смысла не имеет. В другой ситуации эта установка могла бы быть вполне оправданной.

Группам и элементам – подчинение как группам, так и элементам справочникавладельца.

Мы	укажем	В	параметре Использование
подчинения вариант	г Элементам, Рису	нок3.13.	

Подсистемы Функциональные опции Иерархия Владельцы Данные Нумерация Формы Команды Ввод на основании Права Обмен данными Прочее	Основные		1 ×
Функциональные опции Иерархия Владельцы Данные Нумерация Формы Команды Макеты Ввод на основании Права Обмен данными Прочее	Подсистемы	список владельцев справочника.	
Иерархия Владельцы Данные Нумерация Формы Команды Макеты Ввод на основании Права Обмен данными	Функциональные опции	Справочник.Контрагенты	
Владельцы Данные Нумерация Формы Команды Макеты Ввод на основании Права	Иерархия		
Данные Чумерация Рормы Соманды Соманды Макеты Ввод на основании Права Обмен данными	3ладельцы	T	
умерация Рормы іоманды Іакеты вод на основании Ірава Ібмен данными	lанные		
Рормы іоманды Іакеты вод на основании рава бмен данными рочее	умерация		
Команды Макеты Звод на основании Права Обмен данными Прочее	Формы		
Макеты Звод на основании Права Обмен данными Прочее	Команды		
вод на основании Ірава Ібмен данными Ірочее	1акеты		
Ірава Ібмен данными Ірочее	вод на основании		
бмен данными рочее	рава		
Трочее)бмен данными		
	Трочее		
		Использование подчинения:	
Использование подчинения:		Элементам	
Использование подчинения:		STEWERTAM	

Рисунок 3.13. Настройка подчинения

Добавим справочник в состав подсистемы ОперативныйУчетМатериалов.

В состав реквизитов справочника добавим следующие:

Имя: ФИО, тип – Строка, длина 100

Имя: КонтактныеСведения, тип – Строка, длина 100.

Имя: ПредставительРаботает, тип – Булево.

Посмотрим теперь, как выглядит работа с созданными справочниками в режиме »1С:Предприятие». Особенность здесь заключается в том, что, открывая карточку контрагента, в левой ее части мы видим область Перейти, где можно найти ссылку для перехода в справочникПредставителиКонтрагентов, рисунок3.14.

ОО "Полет" (Контрагент ерейти	ы) Представители ко Создать 🗟 🖉 🗵	итрагенто 🔍 Найти 🍇	DB	Все действи	ля + 🔇
редставители контрагентов	Наименование	🚊 Код	Владелец	ΦИO	
	— Казанцев П.И.	00000001	000 "Полет"	Казанцев Петр И	Іванов
представители	контрагентов (создани	e) *			
Записать и закрыть Код: Наименование: Впаделец:	Контрагентов (создания Васильев Б.П. 000 "Полет"	e) *	Все действия -	· [②	

Рисунок 3.14. Форма элемента справочника Контрагенты

При переходе в этот справочник мы будем видеть в открывшемся окне лишь тех представителей, которые относятся к контрагенту, с которым мы в данный момент работаем. При создании новой записи о представителе он автоматически будет "привязываться" к тому контрагенту (*поле*владелец будет заполнено должным образом), из формы элемента которого мы перешли в справочник **ПредставителиКонтрагентов**. В форме списка справочника будет отображаться *ссылка* для перехода к форме элемента справочника.

ОО "Полет" (Контрагент ерейти	ы) Представители кон Создать 🗟 🖉 🗵 🔍	грагенто Найти 💐	98	Все действия	•
едставители контрагентов	Наименование	Код	Владелец	ФИ0	
	— Казанцев П.И.	000000001	000 "Полет"	Казанцев Петр Ива	анов
Поолстаритори	VOUTDOFOUTOB (COODDUNO)	*			
Представители Записать и закрыть Код: Наименование:	контрагентов (создание) Васильев Б.П.	*	Все действия +		
Представители Записать и закрыть Код: Наименование: Владелец:	контрагентов (создание)	*	Все действия +		
Представители Записать и закрыть Код: Наименование: Владелец: ФИО:	контрагентов (создание) Васильев Б.П. ООО "Полет" Васильев Борис Петрович	*	Все действия +		
Представители Записать и закрыть Код: Наименование: Впаделец: ФИО: Контактные сведения:	контрагентов (создание) Васильев Б.П. 000 "Полет" Васильев Борис Петрович 3-45-43	*	Все действия +		
Представители Записать и закрыть Код: Наименование: Впаделец: ФИО: Контактные сведения: Представитель работает:	контрагентов (создание) Васильев Б.П. ООО "Полет" Васильев Борис Петрович 3-45-43	*	Все действия +		

Рисунок 4.15. Формы списка и элемента справочника ПредставителиКонтрагентов

Мы можем создавать элементы справочника **ПредставителиКонтрагентов** и непосредственно перейдя в него, тогда нам придется самостоятельно указывать его владельца – элемент справочника **Контрагенты**. При переходе в *подчиненный справочник* не из формы элемента справочника-владельца, мы можем просматривать все его элементы, рисунок 4.16.

و - < ا 🖉 🔊	😭 Салон	красоты (1С:Пр	едприятие, учебная версия)				😪 🖻 🔳 🗓	i M M+ M- (i) - □ >
Pafousia Vuez		Отератирисий						
стол мас	стеров	учет материало	18					
Документы учета матер	риалов		Отчеты					
Единицы измерения		0	статки материалов					
Контрагенты Номенклатура		G	исок контрагентов					
Организация			Полетоли кон	TROFOUTO				
Отпуск материалов мас	стеру		представители кон	грагенто	в			-
Представители контраг	ентов		🔁 Создать 🔯 🖉 🗶 🍳	Найти 🔍				Все действия • 🧐
Районы			Наименование 🚊	Код	Владелец	ФИ0	Контактные сведения	Представитель работает
Реализация материалов	6		 Васильев В.П. 	00000001	ПОЛЕТ	Васильев В.П.	123	~
Сотрудники			— Семенов В.П.	00000002	Посадка	Семенов В.П.	456	~

Рисунок 3.16. Просмотр формы списка справочника ПредставителиКонтрагентов

Перейдем в окно редактирования объекта конфигурации справочника **Контрагенты**, перейдем на его закладку **Формы**, создадим новую форму списка. При работе с конструктором форм можно заметить, что на закладке управления реквизитами присутствуют два элемента – Дерево иСписок. Список мы с вами уже видели, а элемент Дерево характерен для иерархических справочников, он позволяет облегчить навигацию побольшим справочникам, выводя их иерархическую структуру в дополнение к списку. Установим флаг в поле Дерево, из списка реквизитов, отображаемых в дереве элементов, выберем Наименование, рисунок3.17.

Реквизиты		1
🛛 🗹 Дерево		A
Ссылка		_
Иаименование		
Код		
Родитель		
ПометкаУдаления	-	
Предопределенный		
Организация		
🖯 🗹 Список		
Ссылка		
И Наименование		
✓ Код		
Родитель		
ПометкаУдаления		
Предопределенный	-	

Рисунок 3.17. Конструктор формы справочника Контрагенты

Вот как будет выглядеть форма списка справочника в режиме «1С:Предприятие», 4.18.

🤨 😔 🔍 ⊳ 👻 🏠 Can	он красоты (1С:Пр	едприятие, учебная версия)				🏠 🖻 🔳 🖼 M M+
Рабочий Учет работы мастеров	Оперативный учет материали	Администрирование				
Документы учета материалов Единицы измерения Контрагенты Номенхлатура Организация Оттуск материалов мастеру Поступление материалов Поедставители контрагентов	MOC	Отчеты атериалы статки материалов инок контрагентов Контрагенты © Создать 🙀 🐼 🖋 🔍 Найти	1.00			
Районы		Ссылка	Наименование	Код		Наименование
Реализация материалов		🖂 📄 Контрагенты				🖂 🧰 Поставщики
Сотрудники		🖂 🧰 Покупатели	Покупатели	0000	00	🖃 📴 НовыеПоставщики
		🐵 📄 ПостоянныеПокупатели	ПостоянныеПокупатели	0000	00	- ПОЛЕТ
		😑 📄 Поставщики	Поставщики	0000	00	— Посадка
		🔿 📄 Новые Поставщики	НовыеПоставщики	0000	00	

Рисунок3.18. Форма справочника со списком и деревом элементов

Перед тем, как перейти к дальнейшему рассмотрению материала – заполните справочник «Контрагенты» и подчиненный справочник «Представители контрагентов» тремя-четырьмя записями!!.

Проверка заполнения реквизита справочника, фильтрация

Расширим справочник **Контрагенты**, добавим в состав его реквизитов еще один – назовем его **ОсновноеКонтактноеЛицо**, тип – **СправочникСсылка.ПредставителиКонтрагентов**. Смысл этого поля заключается в хранении ссылки на представителя контрагента, который является "основным" для данного контрагента. Если нужно связаться с контрагентом, можно открыть его карточку и тут же увидеть, какой представитель является основным. Создадим форму элемента справочника **Контрагенты** и посмотрим на нее, попытавшись установить новое *поле* – **Основное контактное лицо**, рисунок3.19.

АО "Фантазия" (Кон ерейти редставители контраген	ЗАО "Фантазия" Записать и закрыть Код:	(Контрагенти © 00000004	ы)	Все действия + 🥝
	Наименование:	ЗАО "Фантазия"	🔞 В (1С:Предприя. 🕅	🗑 📆 M M+ M- 🗆 🗡
	Родитель:	Покупатели	Васильов Б.П. /	Превставители ко
Представите	ли контрагентов оздать 🗟 🖉 🖹 🔍 На	йти ஆ	Записать и закрыть Код: Наименование:	Все действия + (2) 000000002 Васильев Б.П.
Наименование	A	Код	Владелец	000 "Полет" Q
— Васильев Б.П.		00000002	ФИО:	Васильев Борис Петрович
Karan Did		00000001	11	(D. A.C. A.D.
— Казанцев П.И.				

Рисунок 3.19. Попытка заполнения реквизита Основное контактное лицо

Видно, что при попытке подбора элемента в данное *поле* нам показывают не только те элементы справочника**ПредставителиКонтрагентов**, владельцем которых является редактируемый элемент, но и все остальные. Так работать неудобно – это значит, что нам нужно настроить фильтрацию выводимых элементов. Для того, чтобы это сделать,

удобнее всего будет воспользоваться свойством Связи параметров выбора реквизита ОсновноеКонтактноеЛицо. Для открытия палитры свойств реквизита мы можем сделать двойной щелчок по реквизиту в окне редактирования объекта воспользоваться конфигурации, в дереве конфигурации, или командой контекстного меню Свойства.

В открывшейся палитре свойств найдем свойство Связи параметров выбора и нажмем на кнопку с тремя точками около этого поля. Появится окно Связи параметров выбора, рисунок3.20.

Код	Имя	Реквизит	Режим измене.
Наименование Организация ПолноеНаименование ПометкаУдаления Предопределенный Родитель Ссылка ЭтоГруппа	<		

Рисунок 3.20. Окно Связи параметров выбора

В левой части окна можно видеть доступные реквизиты (это реквизиты открытого элемента справочника **Контрагенты**), в правом – параметры, влияющие на отбор элементов в появляющемся окне выбора элементов при заполнении поля представителя контрагента. Выделим *реквизит* **Ссылка** и нажмем на кнопку **Добавить выбранный реквизит в параметры выбора** (она находится между полями). Окно примет следующий вид, рисунок3.21.

зязи параметров выбора				
астройка параметров, которь адается имя параметра и дан	ые будут ные, из	использоваться г которых брать зна	при выборе значений. ачение.	
оступные реквизиты:		Параметры:		
Код		Имя	Реквизит	Режим измене
КонтактныеСведения	>	Отбор.Владелеь	ц Ссылка	Очищать
Наименование	<			
Организация				
ІолноеНаименование				
ТометкаУдаления				
Предопределенный				
одитель				
ЭтоГруппа				
		1		

Рисунок 3.21. Окно Связи параметров выбора с настроенным параметром

В данном случае в строке области **Параметры** отображается как раз то, что нам нужно – нам нужно, чтобы отбор в раскрывающемся списке происходил *по* владельцу, а именно – *по* текущему открытому элементу справочника **Контрагенты**, на который и указывает *реквизит* **Ссылка**. В *поле* имя можно выбрать другие варианты отбора, рисунок3.22.

оступные реквизиты:		Параметры:		
Код		Имя	Реквизит	Режим измене
КонтактныеСведения	>	Отбор.Владелец 💌	Ссылка	Очищать
Наименование	2	Отбор.Владелец		
Организация		Отбор.Код		
ПолноеНаименование		Отбор.Контактны		
ПометкаУдаления		Отбор.Наименова		
Предопределенный		Отбор.Организация		
Родитель		Отбор.ПометкаУд		
ЭтоГруппа		Отбор.Предопред		
		Отбор.Представи		
		Отбор.Ссылка		
		Отбор.ФИО		
		Отбор.ФИО		

Рисунок 3.22. Настройки в окне Связи параметров выбора

После того, как эта настройка выполнена, мы можем нажать **ОК** в окне **Связи параметров выбора** и проверить функциональность решения – при заполнении поля **ОсновноеКонтактноеЛицо** *список* выбора ограничивается подчиненными элементами.

Литература:

Официальный сайт интернет-униерситета «Интуит» (электронный ресурс), режим доступа http://www.intuit.ru/studies/courses/2318/618/lecture/17939.

Контрольные вопросы для самопроверки

- 1. Как связать два справочника?
- 2. Виды справочников в «1С:Предприятие»?
- 3. Особенности иерархических справочников

Задание к лабораторной работе

Создать справочники в системе в соответствии с порядком, отраженным в теоретической части.

Методические указания и порядок выполнения работы

Работа выполняется в точном порядке, как это указано на рисунках и в теоретической части.

Индивидуальное задание

не предусмотрено.

Требования к отчету и защите

1. Результатом выполнения лабораторной работы является сформированный в программе файл, содержащий выполненные задания. В ЭИОС результаты работы не выкладываются.

2. Планируется защита работы, где студент комментирует порядок выполнения заданий, а также отвечает на вопросы, представленные выше.

ЛАБОРАТОРНАЯ РАБОТА №4

РАБОТА С ОТЧЕТАМИ

ОБЩИЕ СВЕДЕНИЯ

Цель: научиться информационную базу на платформе ««1С:Предприятие»».

Материалы, оборудование, программное обеспечение: 1. персональный компьютер (компьютерные классы ГУК), 2. программное обеспечение «1С:Предприятие».

Условия допуска к выполнению: умение работать на ПК и знание техники безопасности.

Критерии положительной оценки: предоставление результатов работы в виде файла и прохождение защиты.

Планируемое время выполнения: Аудиторное время выполнения (под руководством преподавателя): 2 ч. Время самостоятельной подготовки: 2 ч.

Теоретическое введение

Простой отчет

Конечная цель любой учетной системы – формирование отчетов. «1С:Предприятие» 8.2 предоставляет разработчику множество инструментов для *создания отчетов* – от достаточно простых механизмов, позволяющих создавать несложные отчеты, до комплексных средств, таких, как *система компоновки данных*. Сейчас мы рассмотрим пример создания простого отчета. Нам нужно вывести отчет в виде списка контрагентов *по* группам с указанием наименования контрагента, основного контактного лица и телефона этого контактного лица.

Отчет нужного нам вида можно сформировать различными способами. Так, вполне можно реализовать эту функциональность непосредственно в справочнике **Контрагенты**, добавив в него соответствующие программные *механизмы*, выведя кнопку "Сформировать список контрагентов" в форму списка справочника. Можно сделать это с помощью специализированного прикладного объекта **Отчет**. Обычно для создания подобного рода отчетов так и поступают.

Отчет, в нашем случае, будет строиться на основе макета, с областями которого работают в программном коде, формируя готовый отчет. Как правило, если речь идет о неких общих отчетах, их создают в виде отдельных объектов, если же, например, нам нужно создать печатную форму для отдельного элемента справочника или отдельного документа – вполне можно "поместить" всю функциональность требуемого отчета внутри прикладного объекта. В частности, прикладные объекты могут иметь в числе подчиненных объектов макеты, которые и используются при создании отчетов. В то же время, внешний отчет вполне можно использовать и для создания *печатной формы* отдельного элемента.

Создадим в ветви *дерева конфигурации* Отчеты новый отчет, дадим ему имя СписокКонтрагентов, рисунок 5.1.

Основные	Имя:	СписокКонтрагентов
Подсистемы	-	
Функциональные опции	Синоним:	Список контрагентов
Данные	Комментарий:	
Формы		
Команды	Основная схем	а компоновки данных:
Макеты		×
Права		Открыть схему компоновки данных
Прочее		
	Расширенное п	редставление:
		L
	Пояснение:	
		<u></u>

Рисунок 5.1. Создание нового отчета

Первым этапом работы над отчетом станет создание макета отчета. *Макет* позволяет заранее определить и оформить "блоки", из которых будет построен отчет.

Следует отметить, что во всех возможных случаях при разработке *прикладных решений* для «1С:Предприятие» 8.2. следует создавать их на основе схемы компоновки данных. Однако умение работать с макетами в форме табличных документов может пригодиться в том случае, если вам понадобится отредактировать сторонний отчет, выполненный в таком стиле.

Перейдем на закладку формы редактирования объекта **Макеты** и нажмем на кнопку **Добавить**. Появится окно конструктора макета, где нам предложат задать его имя (оставим имя *по* умолчанию – **Макет**), и тип макета – нас устроит **Табличный** документ, рисунок 5.2.

Конструктор ма	жета	×
Имя:	Макет	
Синоним:	Макет	
Комментарий:		
– Выберите тип і	макета:	
• Табличный	документ	
О Текстовый	документ	
Двоичные д	анные	
Active docur	nent	
OHTML DOKYN	иент	
ОГеографиче	ская схема	
Графическа	ия схема	
ОСхема комп	юновки данных	
Макет офор	мления компоновки данных	
Загрузить из фа	айла:	
	Готово Отмена Справ	зка

Рисунок 5.2. Создание макета для отчета

После нажатия на кнопку Готово, мы видим табличный редактор, Рисунок 5.3., очень напоминающий Microsoft *Excel*. Работая с ним, мы можем пользоваться стандартной палитрой свойств, а так же – панелями инструментов, в частности – **Форматирование, Табличный документ, Имена**. Наша задача сейчас – создать и отформатировать области, которые позже будут использованы для формирования готового отчета.

OTHET CHICK	ecKnerrpareerroes: M	SHORE .	- 101 - 1	W 815 94		ST 1 5 10	<u> </u>	×			
1	2 3	4	5	8	7 8	9	10 1	1 12 53 1 4			
1								Споясных: Нисяки			
3								11回图 × ~			
4								Sveikor			
5								· Ocnobrise:			
6	Toffeste	the press	H	-	X Messa	- ×		Текст			
7	D S	10 L L	T# 18 1	P		10 No.					
9	2 9		0.00 1.00 1.00	less i das a	Renaultantia						
10								Parameter many Tanon Ta	Anno	12	
11	and the second se	DOTUD-D-DI		_			- 21	Transie Contraction	Case of Case o		
92	XXX	D. D. A.	A . A .	105 345 321	per i 175 i And	1.1		all the second s	and a second sec		
13		5 1 LL - W -	60 (Au C).	10° 49° 40	The lift from	164	and the local	Jealurta			
15								Тимерсыяка		100	
16								Режими/зменение-азмерала	1		
17								· Maker:		174	
18								Заполнение	Texcr	*	
19								ПараметрРасшиеровки	Lai-		
24								ИспользованиеРасцияровки	Rveikce		
22								* Beariersea			
23								Viva			
24								Vies, Nane			
- 55								a second s			
4B.								2 N 19	6		

Рисунок 5.3. Средства редактирования макета отчета

При создании макета мы можем вводить в ячейки обычный текст – такой текст отображается в ячейке без каких-либо дополнительных знаков. *Ячейка* может содержать именованный *параметр*, который будет заполнен при формировании отчета. Также

ячейки могут содержать шаблоны, состоящие из обычного текста и параметров, которые так же можно заполнить.

		1	2	3	4	5				
Шапка	1						-			
	2		<Список контрагентов на [ДатаФормированияОтчета]>							
	3		11							
	4		Наименование	Контактное лицо	Телефон контактного лица	1	1			
	5					1				
Элемент	6		<Наименование>	«ОсновноеКонтакт	<ТелефонКонтактногоЛица>	-				
	7									
Группа	8		<Наименование>			10				
	9]				
	10				6. 	1				
	11						-			
	17				hard and the second		1			

На рисунке 5.4 показан готовый макет.

Рисунок 5.4. Готовый макет отчета

Ячейка 2,2 заполнена следующим образом: в нее сначала введен текст "Список контрагентов на [ДатаФормированияОтчета]", после чего вызвано окно свойств этой ячейки, в которых в свойстве Заполнение выбраноШаблон, рисунок 5.5.

						Оройства: Ячейки		•	×
А. Отчет СписокКонтрагентов: Макет						[[][][][][] × ×			
	[]	1	2	3		Ячейки			*
Wanka	1			1	in.	Имя	B2C2		
	2		<Список контрагентов на ЩатаФори	ированияОтчета]>		200000	120		
	3	I				Setteria			
	4	1	Наименование	Контактное лицо Т	'eni	Гиперссылка			
	5			100 C	-	РежимИзмененияРазмераКо;	юнки Обычный	*	1+
Элемент	6		<Наименование>	<ОсновноеКонтакт <	Ten	* Maker:			
	7				_	Заполнание	Illafinoe	1	
Группа	8		«Наименование»		_	Concern Descent of the second se			
	9				_	Гараметр асшифровки			
	10				_	ИспользованиеРасшифровки	Ямейка	-	
	11				_	• Значения:			
49						СолехжитЭначение			
_						Tier Businesen		1.1.1	
						THE LOT OF STREET		1.2.1.4.4	4
						Annanaet Merrine measure		1 100	-
						Заполнение ЕШире			
						and a second and the second se			

Рисунок 5.5. Настройка ячейки, содержащей шаблон

Параметр **ДатаФормированияОтчета** мы установим в текущую дату программно при формировании отчета.

Ячейки с 4,2 *по* 4,4 содержат обычный текст – он будет выводиться в качестве шапки таблицы.

И заголовок отчета и шапка таблицы объединены в область с именем Шапка. Для задания имени области достаточно выделить нужные ячейки (выделять нужно *по* заголовкам строк) и отредактировать в палитре свойств *параметр* Имя выделенного диапазона, или воспользоваться кнопкой Назначить имя панели инструментов Имена.

Область Элемент содержит три параметра – Наименование, ОсновноеКонтактноеЛицо и ТелефонКонтактногоЛица. После ввода в каждую из ячеек имен параметров, нужно выделить их (все вместе или *по* одной) и в окне свойств в *поле* Заполнениеуказать Параметр. К тексту в ячейках будут автоматически добавлены угловые скобки (\diamond), что позволяет визуально определить наличие в ячейке параметра.

Область Группа содержит лишь параметр Наименование.
Обратите внимание на имена параметров – они соответствуют именам реквизитов справочника, которыми мы собираемся их заполнять.

Ячейки в шаблоне можно форматировать – задавать их границы, оформление текста, выравнивание и т.д.

Теперь приступим к созданию формы отчета. Перейдем на вкладку **Формы** окна редактирования объекта, добавим новую форму отчета, оставим все настройки в состоянии *по* умолчанию и нажмем **Готово**. Добавим, на вкладке **Реквизиты** редактора форм новый *реквизит*, назовем его **Табличный Документ**, выберем для него тип **Табличный Документ**. Перетащим созданный *реквизитв поле* **Элементы**. В состав команд формы добавим новую команду, зададим ей имя **СформироватьОтчет** и так же переместим в*поле* **Элементы**. В итоге у нас получится форма, выглядящая так, как показано на рисунок 5.6.

Отчет СписокКонтрагентов: Форма	аОтчета			_ 🗆 >
0 / X 🔄 🗣 🕠	5 B 0 X			
Форма СВО Командная панель	Реквизит		Использо всегда	Тип
Табличный Документ Такі Саланта Олиона	Э Отчет Табличный Покомент			(ОтчетОбъект.СписокК
ОК СформироватьОтчет	😑 Табличный Документ	0		ТабличныйДокумент
		12070	С Пак	0140770-1
		101140		Admic (por
Табличный документ:				
	A.			
	.e.			
	*			
Сформировать отчет				
🚰 Форма 📃 Модуль				

Рисунок 5.6. Настройка формы отчета

Для построения отчета мы должны будем получать данные из справочника. Это можно сделать с помощью уже знакомого вам объекта СправочникВыборка, можно получить данные с помощью запроса. В любом случае, это предусматривает работу с базой данных, т. е. нам понадобится процедура, выполняемая на сервере. До настоящего времени мы пользовались лишь директивой компиляции &HaCepвepe – при вызове методов, объявленных с этой директивой, мы имеем *доступ* к контексту формы, при этом между клиентом и сервером происходит передача дополнительных данных – как при

вызове серверного метода с клиента на*сервер*, так и в обратном направлении. Сейчас мы воспользуемся серверным внеконтекстным методом для формирования отчета. В этом методе мы собираемся формировать табличный документ, содержащий данные отчета.

При реализации метода в виде процедуры, нам придется передать в него в качестве параметра наш *реквизит* Табличный Документ. По умолчанию параметры передаются *по* ссылке, т. е. работать процедура будет непосредственно с нашим реквизитом.

При реализации метода в виде функции мы можем ничего не передавать в него, сформировать внутри функции табличный документ и вернуть уже заполненный документ в точку вызова, присвоив его нашему реквизиту **ТабличныйДокумент**.

Реализуем метод в виде функции. Готовый код формирования отчета (рисунок 5.7) будет выглядеть следующим образом:

Список контрагентов на	a 29.09.2011 20:39:04	
Наименование	Контактное лице	о Телефон контактного лица
Покупатели Постоянные покупатели		
ЗАО "Фантазия"	Kapnoe 5.0.	3-55-44
Поставщики Новые поставщики	terrist strategy	
000 "Полет"	Казанцев П.И.	3-34-35

Рисунок 5.7. Готовый отчет

&НаКлиенте Процедура СформироватьОтчет(Команда) ТабличныйДокумент=СформироватьОтчетНаСервере(); КонецПроцедуры

&НаСервереБезКонтекста
Функция СформироватьОтчетНаСервере()
ТабличныйДокумент=Новый ТабличныйДокумент();
Макет=Отчеты.СписокКонтрагентов.ПолучитьМакет("Макет");
Шапка=Макет.ПолучитьОбласть("Шапка");
Элемент=Макет.ПолучитьОбласть("Элемент");
Группа=Макет.ПолучитьОбласть("Группа");
Шапка.Параметры.ДатаФормированияОтчета=ТекущаяДата();
ТабличныйДокумент.Вывести(Шапка);
Выборка=Справочники.Контрагенты.ВыбратьИерархически();
Пока Выборка.Следующий() Цикл
Если Выборка.ЭтоГруппа Тогда
Область=Группа;
Иначе
Область=Элемент;

КонецЕсли;

Область.Параметры.Заполнить(Выборка); ТабличныйДокумент.Вывести(Область); КонецЦикла; Возврат (ТабличныйДокумент);

КонецФункции

В клиентской процедуре СформироватьОтчет() мы вызываем серверную внеконтекстнуюфункцию СформироватьОтчетНаСервере(), присваивая возвращаемое ей значение реквизиту Табличный Документ.

В серверной функции мы создаем новую переменную с типом **ТабличныйДокумент** и именем **ТабличныйДокумент**. Именно в него мы будем выводить данные и именно его будем возвращать в точку вызова. Несмотря на то, что имя реквизита формы и имя данной переменной совпадают, между ними нет никакой связи. Это – отдельные объекты.

Далее, мы получаем *макет* из нашего отчета, пользуясь методом **ПолучитьМакет()** и задавая имя макета. Отчет может иметь несколько макетов, их выбор осуществляется *по* имени.

В переменную Шапка мы записываем область макета Шапка, соответственно поступаем с переменными Элемент и Группа.

Командой Шапка.Параметры.ДатаФормированияОтчета=ТекущаяДата(); мы заполняем ранее заданный в макете *параметр*ДатаФормированияОтчета, записав в него текущую дату. Дата, возвращаемая функцией ТекущаяДата, содержит, помимо года, месяца и дня, так же часы, минуты и секунды. При необходимости дату перед выводом можно отформатировать при помощи функции Формат().

После того, как *параметр*, находящийся в шапке, заполнен, мы можем вывести шапку в табличный документ командой **Табличный Документ.Вывести**(Шапка);

Следующим этапом нашей работы будет получение иерархической выборки справочника. Такая *выборка* позволяет получить элементы и группы справочника в иерархическом порядке, учитывая родительские отношения между элементами.

В цикле обхода выборки мы сначала проверяем, является ли текущий элемент справочника группой. Если является, присваиваем переменной **Область** ранее полученную область отчета **Группа**. Если не является – присваиваем ей область **Элемент**.

Благодаря совпадению имен параметров и имен реквизитов справочника, для помещения данных из выборки в область макета, достаточно воспользоваться конструкцией **Область.Параметры.Заполнить(Выборка);**. После заполнения параметров мы выводим сформированную область в табличный документ.

Когда цикл перебора выборки будет завершен, мы возвращаем сформированный табличный документ в точку вызова, и*пользователь* видит готовый отчет.

Для формирования простейших отчетов, *пользователь* может воспользоваться стандартной функциональностью, присутствующей в «1С:Предприятие» 8. Для этого, открыв, например, *список* справочника, он может выполнить команду Все действия > Вывести список. Появится окно Вывести список, рисунок 5.8, где в *поле* Выводить в можно выбрать либо Табличный документ (его обычно и используют), либо – Текстовый документ.

Вывести список		
Выводить в:		
Табличный документ		
Выводить колонки:	1	
И Наименование		
Код		
Фамилия		
Имя		
Отчество		
 Дата рождения 		
Пол		
Район проживания		
Только выделенные		

Рисунок 5.8. Окно Настройка списка

В *поле* Выводить колонки можно настроить состав выводимых в документ колонок (в нашем случае *команда* выполнена для справочника Физические.Лица). После нажатия на ОК выбранные данные оформляются в виде табличного документа, а с помощью команды Файл > Сохранить как, рисунок 5.9, этот документ можно сохранить в нужном формате для дальнейшей обработки в других приложениях.

Pain	*	Homese	CHAN 3	4	5
Полека		1. Common	040	Viera	Отчество
		Zubeur	1.000.000	Александр	Rootuseoone
1 a0minua		Сохранить	QifeS	Rapen	Петрович
Серенис	*	Сокранить как		Maar	Ивенович
Okna		No.		Петр	Петрович
	1	Сравнить дайлы		Александр	Иванович
Слдавка		2 Дечать	Ctri+P		
	6	Deserver and the second second second	DU.S.O.P		
	- 6	Principality of low Smithause Lago (house outs)	CONTONIET		
	1	д Предварительный просмотр			
		Параметры страницы			
		Список файлов			
			100000		

Рисунок 5.9. Вывод данных в табличный документ

Справочник с иерархией элементов

Добавим конфигурацию справочник. В нашу еще ОДИН Дадим ему имя Подразделения, добавим в состав подсистемБухгалтерскийУчет, УчетРаботыМастеров и РасчетЗаработнойПлаты. Увеличим длину наименования на закладке Данныедо 100 символов. Сделаем справочник иерархическим закладке Иерархия установим флаг Иерархический на справочник, параметр Вид иерархии установим в значение Иерархия элементов, рисунок 5.10.

Иерархия элементов вполне логична для справочника Подразделения, так как одни *подразделения* могу включать в себя другие, и, при этом, вполне самостоятельны, их

можно выбирать при заполнении, например, реквизитов других справочников, в то время, как при иерархии групп и элементов, группы играют лишь вспомогательную роль для организации информации внутри справочника.

Основные	Иерархический справочник	1
Подсистемы	Видиерархии	
Функциональные опции		1
Иерархия	иерархия элементов	
Владельцы	Размещать группы сверху	
Данные	Ограничение количества уровней иерархии	l
Нумерация	Количество уровней иерархии	2
Формы		
Команды		
Макеты		
Ввод на основании		
Права		
Обмен данными		
Прочее		

Рисунок 5.10. Настройка иерархии справочника Подразделения

Кроме того, справочник Подразделения мы добавим В несколько Эти предопределенных элементов. элементы справочника задаются в Конфигураторе, пользователь обладает лишь ограниченными возможностями по управлению ими, в частности, не может их удалить. Такие элементы обычно создают для того, чтобы ими можно было удобно и надежно оперировать в программном коде, не опасаясь того, что пользователь удалит их.

Для этого перейдем на вкладку окна редактирования объекта **Прочее** и нажмем на вкладку **Предопределенные**. В окне ввода предопределенных элементов справочника введем следующие (рисунок 5.11.):

- Администрация
- Бухгалтерия
- Парикмахерская

Гмя	Код	Наименование	
Э 🛄 Элементы			
- Администрация	00000001	Администрация	
📟 Бухгалтерия	00000002	Бухгалтерия	1.47
👄 Парикмахерская	000 Элемент справоч	ника	×
	Родитель:	Элементы	
	Имя:	Парикмахерская	B
	Код:	00000003	
	Наименование:	Парикмахерская	

Рисунок 5.11. Создание предопределенных элементов справочника Подразделения

Доработаем справочник **Сотрудники**. Снабдим его следующими реквизитами, рисунок 5.12.:

Имя: ФизическоеЛицо, Тип: СправочникСсылка.ФизическиеЛица Имя: Подразделение, Тип: СправочникСсылка.*Подразделения* Имя: Расчетчик, Тип: Булево

Имя: Пользователь, Тип: Строка, длина 50.

Увеличим длину наименования до 50 символов.

	Длина наименовани	49 50
Функциональные опции	- Тип кода	Основное представление —
Иерархия	<u>Число</u>	🔘 В виде кода
Владельцы	Острока	• В виде наименования
Данные		
Нумерация		₽
Формы	😑 👄 Реквизить	1
Команды	Физич	ескоеЛицо
Макеты	Pacuar	зделение
Ввод на основании	- Гасчет	ратель
Права		
Обмен данными	11 to 🖌 🖌 🗸	
Прочее	🗐 Табличные	части
	Стандартные рекв Общие реквизи	изиты Характеристики

Рисунок 5.12. Реквизиты справочника Сотрудники

Мы хотели бы, чтобы наименование сотрудника в данном справочнике формировалось бы автоматически и состояло бы из ФИО физического лица

и *подразделения*, в котором работает сотрудник. Создадим форму элемента справочника и, для элемента формы **Наименование**, снимем флаг **Доступность**, рисунок 5.13.

0 X 4 4 5	5 B / X				
Форма	Реквизит	-	Использовать Тип		
от Командная панель			Окойства: Поле		
- Код	🛞 🚥 Объект		AL THE VELV U		
- Наименование			24 44 80 0 1		
— ФизическоеПицо			Vice	Hammana	
 Подразделение 			Banoneny	1 Idventority bidranto	
 Расчетчик. 			Col Millions		
- Пользователь			Ben	Done secons	
- Организация			ПутьКДанным	Объект Наменованов	
			Положение Загодовка	Aero	
			Валимость	3	60
			Пользовательская видимост	ь Открыть	
	B Persented B &	Conner I	Доступность		
			ТолькоПросмотр		
		100	ПропускатьПриВеоде	Авто	
Записать и закрыть 📓	Все действи	tA = 📀	Пропускать ПриВеоде Активизировать По Умолнания	(Авто о []	
Записать и закрыть	Все действи	tA * 📀	Пропускать ПриВеоде Активнаировать ПоУмолчания Маска	Aero	1.
Записаль и закрыть 🛃 Код	Все действи	19 × @	Пропускаль При Веоде Активизировать По Умолнания Маска Режини Пароля	Авто с Авто	
Записаль и закрыть 🛃 Код	Все действи	ta = 🥘	Пропускать ПриВеоде Активизировать ПоУмолчания Маска Режим Пароля Многострочный Режим	Авто о Авто Авто Авто	
Записаль и закрыть 🔄 Код	Все действи	(D) + R*	Пропускать ПриВеоде Активизировать ПоУмолчания Маска Режим Пароля Многострочный Режим Расширенное Редак тирования	Aero Aero Aero Aero	•
Записать и закрыть Код Намисать и закрыть Намисать и пакрыть Подразделение:	Все действи		Пропускать ПриВеода Активизировать ПоУмолчания Маска Режим Пароля Многострочный Режим Расширенное Редак тирования • Использование:	Aero Aero Aero Aero	•
Записать и закрыть Код. Намоснование Физическое лицо: Подрезделение: Росчетия:	Все действи	() + R	Пропускать ПриВеода Активизировать ПоУмолчания Маска Режим Пароля Многострочный Режим Расширенное Редак тирования • Использование: Отображение Предитреждения	Авто Авто Авто Авто Авто яПриРед Авто	•
Записать и закрыть Код. Намоснование Физическое лицо: Подрезделение: Расчетия:	Все действи	(Q) + (Q)	Пропускать ПриВеода Активизировать ПоУмолчания Маска Режим Пароля Многострочный Режим Расширенное Редак тирования • Использование: Отображение При Дерактия Предитреждение При Редактия	Авто Авто Авто Авто Авто авто авто ровении	•
Записать и закрыть Код. Наменование Физическое лицо. Подрезделение Расчетник: Пользователь:	Все действи	19 × 3	Пропускать ПриВеода Активизировать ПоУмолчания Маска Режим Пароля Многострочный Режим Расширенное Редак тирования • Использование: Отображение При Дпреждения Предупреждение При Редак ти Сочетание Клавиш	Авто Авто Авто Авто Авто акто акто ровении	* * * *
Записать и закрыть Код Намоннование Физическое лицо: Подразделение: Расчетчик: Пользователь: Организация:	Все действи	() + H	Пропускать ПриВеода Активизировать ПоУмолчания Маска Режим Пароля Многострочный Режим Расширенное Редак тирования • Использование: Отображение Предпреждения Предпреждение При Редак ти Сочетание Клавици Подсказка	Авто Авто Авто Авто Авто акто акто ровании	* * * *
Записать и закрыть Код Намоснование Физическое лицо: Подразделение: Подразделение: Организация:	Все действи	9 + H 99 99	Пропускать ПриВеода Активизировать ПоУмолчания Маска Режим Пароля Многострочный Режим Расширенное Редак тирования • Использование: Отображение Придпреждения Предпреждение При Редактия Сочетание Клавиш Подсказка	Авто о Авто Авто Авто Авто авто авто ровании Отстыта	• • • •
Заликсаль и закрыль Код Накисть и закрыль Накисть на закрыль Накисть на пакрыль Физическое лицо Подрезделение Расчетчик Подрезделение Организация:	Все действи	() + H	Пропускать ПриВеода Активизировать ПоУмолчания Маска Режим Пароля Многострочный Режим Расширенное Редак тирования • Использование: Отображение Предпреждения Предпреждение При Редак ти Сочетание Клавиш Подоказка Состав жоменд Киолоз Приска Выбора	Авто Авто Авто Авто Авто Авто акто о Авто о Стврыть Дато Дато Отврыть	• • •
Заликсаль и закрыль Код Накирнование Физическое лицо Подрезделение Расчетчик: Пользователь: Организация:	Все действи	9 + H	Пропускать ПриВеода Активизировать ПоУмолчания Маска Режим Пароля Многострочный Режим Расширенное Редак тирования • Исполь зование: Отображение Предпреждения Предпреждение При Редак ти Сочетание Клавици Подоказка Состав жоменд Кнопка Списка Выбора Кнопка Списка Выбора	Авто о Авто Авто Авто Авто акто о Авто Отврыть Авто Отврыть Авто Отврыть Авто Отврыть Авто	* * * * *
Заликсаль и закрыль Код. Наморнование Физическое лицо Подрезделение Расчетник: Пользовалель: Оргенизация:	Bee gesterree	() () () () () () () () () () () () () (Пропускаты ПриВеода Активизировать По Умолчания Маска Режим Пароля Многострочный Режам Расширенное Редак тирования • Исполь зокание: Отображения Предупреждения Предупреждение ПриРедак ти Сочетание Клавиш Подоказка Состав жоменд КчопкаСпискаВыбора КчопкаВибора	Авто Авто Авто Авто Авто Авто абриРед Авто ровании <u>Открыть</u> Авто <u>Дего</u> 4вто	• • •
Заликсаль и закрыль Код Код Наиченование Физическое лицо Подразделение Расчетияк: Пользователь: Организация:	Bee assertes	() () () () () () () () () () () () () (Пропускаты ПриВеода Активизировать ПоУмолчания Маска Режим Пароля Многострочный Резаим Расширенное Редактирования • Использование: Отображение Предпрождения Предпрождение ПриРедактия Сочетание Клавики Подоказка Состав жомана Кнопка Выбора Кнопка Выбора Кнопка Выбора	Aero Aero Aero Aero Aero InTipulPezi Aero poseruni Qmspatta Aero Aero	

Рисунок 5.13. Настройка формы элемента справочника Сотрудники

Теперь подумаем над тем, как автоматически заполнить *поле* Наименование на основе данных полей Физическое лицо иПодразделение. Сделать это можно различными способами, мы реализуем следующую функциональность: перехватим события изменения полей Физическое лицо и Подразделение и вызовем в обработчике каждого из этих событий процедуру, заполняющую *поле* Наименование. Так *пользователь*, заполняющий элемент справочника, сможет сразу же увидеть результаты формирования наименования.

Нашей задаче отвечает следующий код: &НаКлиенте Процедура ФизическоеЛицоПриИзменении(Элемент) СформироватьНаименование(); КонецПроцедуры

&НаКлиенте Процедура ПодразделениеПриИзменении(Элемент) СформироватьНаименование(); КонецПроцедуры

Процедура СформироватьНаименование() Объект.Наименование=Объект.ФизическоеЛицо.Наименование +" ("+ Объект.Подразделение.Наименование+") "; КонецПроцедуры Результаты работы созданного нами механизма показаны на рисунке 5.14.

🕒 Сотрудники (со	оздание) * (1С:Предприятие)	🕅 🗐 M M+ M- 🗆 🗙
Сотрудник	и (создание) *	
Записать и за	крыть	Все действия 👻 📀
Код:		
Наименование:	Александров А. А. (Администрация)	
Физическое лицо	Александров А. А.	Q
Подразделение:	Администрация	۹
Расчетчик:		
Пользователь:		
Организация:	(Q

Рисунок 5.14. Настройка формы элемента справочника Сотрудники

Литература:

Официальный сайт интернет-университета «Интуит» (электронный ресурс), режим доступа http://www.intuit.ru/studies/courses/2318/618/lecture/17939.

Контрольные вопросы для самопроверки:

- 1. Для чего нужны отчеты в системе?
- 2. Виды отчетов в «1С:Предприятие»?
- 3. Как настроить формы справочников и отчетов?

Задание к лабораторной работе

Создать отчеты в системе в соответствии с порядком, отраженным в теоретической

части

Методические указания и порядок выполнения работы

Работа выполняется в точном порядке, как это указано на рисунках и в теоретической части

Индивидуальное задание

не предусмотрено.

Требования к отчету и защите

1. Результатом выполнения лабораторной работы является сформированный в программе файл, содержащий выполненные задания. В ЭИОС результаты работы не выкладываются.

2. Планируется защита работы, где студент комментирует порядок выполнения заданий, а также отвечает на вопросы, представленные выше.

ЛАБОРАТОРНАЯ РАБОТА №5 Документы и регистры

ОБЩИЕ СВЕДЕНИЯ

Цель: научиться создавать документы и регистры на платформе ««1С:Предприятие»».

Материалы, оборудование, программное обеспечение: 1. персональный компьютер (компьютерные классы ГУК), 2. программное обеспечение «1С:Предприятие».

Условия допуска к выполнению: умение работать на ПК и знание техники безопасности.

Критерии положительной оценки: предоставление результатов работы в виде файла и прохождение защиты.

Планируемое время выполнения: Аудиторное время выполнения (под руководством преподавателя): 2 ч. Время самостоятельной подготовки: 2 ч.

Теоретическое введение

Любая учетная система получает исходные данные из документов. В классическом бухгалтерском учете основа всего – первичные документы, автоматизированные системы учета – не *исключение*.

Для описания документов в *дереве конфигурации* имеется отдельная *ветвь* – Документы. В одной из предыдущих лекций мы создали один документ – **ПоступлениеМатериалов**. Сейчас мы займемся работой с ним. Для начала определимся с целью использования этого документа. Мы планируем с его помощью отражать в системе поступление материалов. Исходя из этих целей, нам понадобятся следующие реквизиты документа (Рисунок 6.1.), которые мы зададим на вкладке Данные окна редактирования объекта:

Имя: Контрагент, Тип: СправочникСсылка.Контрагенты

Имя: Ответственный Сотрудник: Тип: Справочник Ссылка. Сотрудники

Добавим в состав табличных частей нашего документа новую табличную часть с именем Материалы и следующими реквизитами:

Имя: Номенклатура, Тип: СправочникСсылка.Номенклатура

Имя: Цена, Тип: Число, длина 10, точность 2

Имя: Количество, Тип: Число, длина 10, точность 3

Имя: Сумма, Тип: Число, длина 10, точность 2

Основные	
Подсистемы	
Функциональные опции	С – геквизиты
Данные	 ОтветственныйСотрудник
Нумерация	
Движения	
Последовательности	
Журналы	
Формы	den de la se a la contra
Команды	
Макеты	😔 🔟 Табличные части
Ввод на основании	🖂 🔝 Материалы
Права	📟 Номенклатура
Обмен данными	— Цена
Прочее	Сумма
	Стандартные реквизиты Характеристики
	Общие реквизиты

Рисунок 6.1. Настройка состава реквизитов документа

На закладке **Нумерация**, рисунок 6.2, можно задать параметры нумерации документов.

Основные			10
Подсистемы	нь тонумерация;		
Функциональные опции	Нумератор		
Данные	Длина номера		9
Нумерация	Контроль уникаль	ности	
Движения	Turs you oppo		
Последовательности	- Тип номера		
Журналы	• Строка		
Формы			
Команды	Периодичность (Н	lепериодический	
Макеты			
Ввод на основании			
Права			
Обмен данными			
Прочее			

Рисунок 6.2. Настройка параметров нумерации документа

В данном случае документы будут нумероваться автоматически с контролем уникальности номеров.

Нескольким различным видам документов можно назначить сквозную нумерацию с одинаковыми настройками благодаря использованию нумератора.

Если в качестве типа номера использована строка – это позволит, при возникновении необходимости, добавлять к номеру различные символьные префиксы.

Настройка периодичности в данном случае установлена в значение **Непериодический**, то есть, независимо от срока работы с информационной базой, нумерация документов будет продолжаться, а при установке некоторой периодичности (в пределах дня, месяца, квартала, года) – уникальность номеров будет соблюдаться в течение указанного периода.

Закладка Движения, рисунок 6.3., позволяет управлять проведением документа.

Основные	Проведение:	Разрешить
Подсистемы	Оперативное проведение:	Запретить
Функциональные опции	Уладение движений:	Удалять автоматически п
Данные		a denoise de lond in record in
Нумерация		
Движения		
Последовательности		
Журналы		
Формы		
Команды		
Макеты		
Ввод на основании	Конструктор движен	નાનણ
Права		
Обмен данными		
Прочее		

Рисунок 6.3. Настройка параметров проведения документа

Если документ планируется проводить, то есть – он будет формировать движения *по* регистрам, проведение следует разрешить. Оперативное проведение позволяет разработчику настроить различное реагирование кода, отвечающего за *проведение документа*, при *проведении документа*, скажем, более поздней или ранней датой, чем текущая дата.

Если мы не планируем реализацию подобной функциональности, мы можем отключить оперативное проведение.

Автоматическое удаление движений документа предусматривает, при отмене движения, проведения, автоматически удалять которые документ сформировал по регистрам. В данный момент в нашей конфигурации пока нет регистров, по которым будет проводиться документ. "Формирование движений по регистру", более простым языком, означает то, что документ при проведении делает записи в регистре. Регистры можно сравнить с таблицами, содержащими ключевые данные о документах, которые по ним проводятся. Работу с различными регистрами мы рассмотрим ниже.

Вкладка **Журналы** позволяет настраивать включение документа в так называемые **журналы** документов. Журнал позволяет организовать совместную работу с документами различных типов, которые включены в этот журнал. Такие документы отображаются в едином списке.

Если *проведение документа* запрещено – то *пользователь* сможет лишь сохранить документ в базе данных. Других воздействий на информационную базу такой документ не произведет. Например, такое поведение может быть характерно для документов, вроде выписанных счетов, которые сами *по* себе воздействия на учет не производят, но их важно

хранить в системе для того, чтобы "помнить" о том, какие счета выписаны, важно иметь возможность формировать их *печатные формы*.

Но то, что счет выписан, еще не гарантирует то, что счет будет оплачен, то, что товары, указанные в выписанном счете будут действительно отгружены покупателю. Если продолжить пример с выписанным счетом и перейти на вкладку **Ввод на основании**, рисунок 6.4., то окажется, что эта вкладка позволяет настроить ввод одного документа на основании другого.

Основные	Вволится на основании:	0 3
Подсистемы		
Функциональные опции		
Данные		
Нумерация		
Движения		
Последовательности		
Журналы		
Формы		
Команды		
Макеты	Лылиется основанием для.	9 7
Ввод на основании		
Права		
Обмен данными		
Прочее		
	Конструктор вв	ода на основани
	- 1864	

Рисунок 6.4. Настройка параметров ввода на основании

Если, например, мы имеем дело с документом наподобие "Отгрузка материалов", окажется, что такой документ вполне логично будет вводить на основании документа "Счет" - после оплаты этого счета и фактической отгрузки материалов. Документ отгрузки, в отличие от счета, фиксирует уже свершившийся факт хозяйственной жизни, который должен оказать воздействие на состояние информационной базы. Такой документ должен проводиться – то есть – делать записи в соответствующие регистры.

На данном этапе мы можем запустить систему, попытаться поработать с документом, используя автоматически сгенерированную форму, и посмотреть, все ли в данной форме нас устраивает.

Прежде чем продолжать работу с документом **ПоступлениеМатериалов**, приведите данные справочника **Номенклатура** в вашей *информационной базе* к виду, показанному в таблице 6.1.

Наименование	Единица измерения	Услуга	Группа
Парикмахерские услуги		Да	Да
Завивка	Час	Да	
Стрижка	Час	Да	
Парфюмерия		Нет	Да
Духи	Штука	Нет	
Одеколон	Штука	Нет	
Прочие материалы		Нет	Да
УФ-гель	Упаковка	Нет	
Спецодежда		Нет	Да
Одежда для парикмахера	Штука	Нет	
Уход за волосами		Нет	Да
Бальзам для волос	Штука	Нет	
Лак для волос	Упаковка	Нет	

Таблица 6.1. Данные справочника Номенклатура

На Рисунок 6.5. вы можете видеть форму документа после ввода в нее некоторых данных.

Пров	вести и закрыть	🔚 📑 Провести		1	Все действия 🔹 🥝
Номер	r:				
Дата:		02.10.2011 0:00:00			
онтра	агент:	000 "Полет"			C
) <mark>твет</mark> с	ственный сотрудник:	Иванов И. И. (Парикма)	ерская)		
Сомме	энтарий:				
)ргани	изация:	[
ОД	обавить 🗟 🗙 4	<mark>₽`</mark> ₽			Все действия
N	Номенклатура		Цена	Количество	Сумма
1	1 Духи		120,00	10,000	
	2 УФ-гель		300.00	2,000	

Рисунок 6.5. Заполнение документа Поступление товаров

Подобный документ, очевидно, заполняется в *информационной базе по* данным некоего бумажного документа (*приходной накладной*, например), поступившей от поставщика вместе с поступившими товарами. Цена и количество каждой товарной позиции вводится из документа вручную, с этим здесь ничего поделать нельзя. Но вот ввод суммы *по* каждой из строк табличной части вполне поддается автоматизированному расчету на основе данных о цене и количестве. Еще один важный момент, на который можно обратить внимание – в нашем документе не отображается итоговый показатель *по* табличной части. Как правило, в подобных случаях пользователю важно увидеть общую сумму документа. Общая сумма позволит быстро сверить данные,

введенные в электронный документ, с его бумажным аналогом. Если общая сумма совпадает – то, почти наверняка, все строки табличной части введены верно.

Реализуем эту функциональность.

Для того, чтобы автоматически заполнить *поле* сумма *по* каждой из строк табличной части, редактируемой пользователем, очевидно, что рассчитывать сумму имеет смысл либо после заполнения поля **Цена**, либо – после заполнения поля **Количество**, перехватив какие-либо события, имеющие *отношение* к редактируемой табличной части.

В нашем случае это должны быть события, генерируемые при изменении полей Цена или Количество при вводе данных в определенной строке. Для того, чтобы назначить обработчики подобных событий для определенных элементов табличной части, можно поступить так же, как мы поступали, назначая обработчики событий для любых других элементов формы (рисунок 6.6.). Для начала, конечно же, нам нужно будет создать собственную форму документа, делается это на закладке Формы окна редактирования объекта. С параметрами, предложенными конструктором форм *по* умолчанию, можно согласиться.

Документ Поступление Материалов: Фо	рмаДокумента					_ 🗆 ×
) / X 🔄 🗣 ҧ		380	ĸ			
 Контрагент ОтветственныйСотрудник 		Реканзит	-	Использ всегда	зовать	Тип
 Комментария Организация Материалы Командная панель МатериалыНомерСтроки МатериалыНоменклатура 		• = Объек	T			(ДокументОбъект.Поступлен
 МатериалыКолич Добавить МатериалыКолич // Добавить МатериалыСуми Изменить Злементы Ком // Уазакть 		F2 Del ma	и 🦉 Команды	Параме	этры	
	uaarw (Ma Shillal In		-	COLOUR	
Провести и закрыть Номер: показать ком Дата: Показать ком	аниз Сля- андную панель гекстное меню	Shift+Down	Все действия +	0		
Контралент: События			<ПриИзменении>			
Ответственный сотрудник Перейти		F12	<НачалоВыбора>			5
Комментарий: Свойства Организация: © Добавить 🔓 🗶 👁 🌳		At+Enter	<ВачалоВыбораИзС «Очистка» «Регулирование»	зиска>		
N Номенклатура	Цена	Количеств	«Открытие»			-
			<ОбработкаВыбора; <АвтоПодбор>			

Рисунок 6.6. Назначение обработчика полю табличной части

Назначим обработчики событий **ПриИзменении** для полей **МатериалыЦена** и **МатериалыКоличество**.

Теперь нам нужно реализовать следующее: при работе в определенной строке таблицы, при вводе в нее данных, получить эту строку, и, при изменении цены или количества номенклатуры рассчитать сумму.

У табличных полей есть свойство **ТекущиеДанные**, которое, как раз, позволяет обращаться к текущей редактируемой строке. Данные редактируются на клиенте, поэтому мы вполне можем обойтись здесь без вызова серверных процедур, выполнив все необходимые действия на клиенте. Если вы хотите побольше узнать о том, что можно сделать с табличным полем из кода, как и в других случаях, помочь вам в этом могут

инструменты отладки. Вот как, например, выглядит свойство **ТекущиеДанные** при срабатывании точки останова в коде модуля нашей формы при отладке кода, который будет представлен ниже, рисунок 6.7.

Выраж	ение				
Выраже	ние:				<u>Рассчитать</u>
Элемен	нты. Материалы. Текущие Данные			•	Включить в табло
Результ	rat:		1		2
Свойст	180	Значение	Тип		закрыть
۲	Родитель	Управляемая Форма	Управляемая Форма		Справка
۲	СочетаниеКлавиш	СочетаниеКлавиш	СочетаниеКлавиш		
*	ТекущаяСтрока	0	Число		
•	ТекущиеДанные	Данные Формы Элемент	Данные Формы Элемент		
	ИсходныйНомерСтроки	1	Число		
	Количество	10	Число		
	Номенклатура	Духи	СправочникСсылка.Номе		
	НомерСтроки	1	Число		
	Сумма	1 200	Число		
	Цена	129	Число		
	ТекущийРодитель		Неопределено		
æ	ТекущийЭлемент	ПолеФормы	ПолеФормы	*	

Рисунок 6.7. Просмотр свойства ТекущиеДанные в окне Выражение при отладке кода

Итак, наша задача может быть решена следующим образом: &НаКлиенте Процедура МатериалыЦенаПриИзменении(Элемент) РассчитатьСумму(); КонецПроцедуры

&НаКлиенте Процедура МатериалыКоличествоПриИзменении(Элемент) РассчитатьСумму(); КонецПроцедуры

&НаКлиенте Процедура РассчитатьСумму() ТекущаяСтрока=Элементы.Материалы.ТекущиеДанные; ТекущаяСтрока.Сумма=ТекущаяСтрока.Количество*ТекущаяСтрока.Цена; КонецПроцедуры

Из пары обработчиков событий **ПриИзменении** вызывается клиентская процедура **РассчитатьСумму()**. Здесь мы получаем данные текущей строки через свойство **ТекущиеДанные** и вычисляем *поле* **Сумма**, перемножая данные в полях **Количество** и **Цена**.

При необходимости, мы можем редактировать *поле* Сумма независимо от значений полей Цена и Количество.

Вторая задача из тех, которые мы поставили себе выше, заключается в выводе на форму итоговых сведений *по* табличному полю. Ее можно реализовать различными способами, но лучше всего воспользоваться стандартными итоговыми показателями табличного поля, которые можно найти в составе табличного поля на закладке **Реквизиты** редактора форм, рисунок 6.8.

0 × 4 4 m		8四 / ×			
 ОтветственныйСотрудник 		Реквизит		Использовать	Тип
- Комментарий				всегде	
- Организация		Home	арСтроки 🔍	~	Число
 МатериалыИтог Сумма 		Home Home	нклатура 🗆		СправочникСсылка.Номенклатура
Материалы		Ш Цена			Число
що Командная панель		П Коли	VECTBO D	•	Число
 МатериалыНомерСтроки 	+	🔝 Сумн	1a 🗉	4	Число
🥌 МатериалыНоменклатура		∑ Коли	чествоСтрок		Число
💻 МатериалыЦена		∑ Итог	Цена		Число
🗯 МатериалыКоличество		Σ Nuor	Количество		Число
 МатериалыСумма 	*	∑ Vitor	Сумма 🛛 🕮		Число
Номер:					
юмер:				Q	
Чомер: Дата: Сонтралент: Этветственный сотрудник:				Q	
юмер: Іата: Сонтрагент: Ответственный сотрудник. Сомментария:				р	
юмер: Іата: Сиптрагент: Ответственный сотрудник. Сомментарий: Сомментарий:				р	
юмер: Дата: Контрагент: Этветственный сотрудник: Комментария: Организация: Сумма (итог):				р	
Комер: Цата: Сонтрагент: Олеатственный сотрудник. Соммантарий: Организация: Организация: Организация: Организация:			Все действ	Q Q Q	
Iомер: Цата:		Количество	Все действ Сумика	P	
Номер: Дата: Контрагент: Этветственный сотрудник: Конментарий: Организация: Эрганизация: Эрганизация: Эрганизация: Эрганизация: Эрганизация: Эрганизация: Эрганизация: Эрганизация: Элеманизация:	E Constanting of the second se	Количество	Все действ Сумика	D	

Рисунок 6.8. Вывод итогового показателя для поля Сумма на форму

Этот *реквизит* – **ИтогСумма** – нужно перетащить на вкладку Элементы. Он будет отображаться на форме, изменяясь при изменениях суммы в строках табличной части, рисунок 6.9.

прове	ести и закрыть	📄 📑 Провести		1	Все действия 👻 ②
Номер:		00000001			
Дата:		02.10.2011 15:24:5	2 🗐		
Контраг	гент:	000 "Полет"			
Ответст	твенный сотрудник:	Иванов И. И. (Пари	икмахерская)		
Коммен	нтарий:	[
Органи:	зация:	1			
Сумма (Доб	(итог): бавить 🔂 🗙 1	3 590,00			Все действия
Ν	Номенклатура		Цена	Количество	Сумма
1	Духи		129,00	10,000	1 290,00
2	УФ-гель		300,00	3,000	900,00
	Onessenter		200.00	7.000	1 400.00

Рисунок 6.9. Форма после модификации

Документ **ПоступлениеМатериалов** мы разработали, настроили его форму. Документ, хотя его проведение и разрешено, пока, фактически, не проводится – мы не реализовали *механизмы* проведения, у нас нет регистров, *по* которым он будет проводиться.

Данную функциональность мы реализуем ниже, а сейчас займемся еще одним документом, который, являясь, *по* составу реквизитов и *по* особенностям устройства формы, очень похожим на документ **ПоступлениеМатериалов**, выполняет противоположную ему функцию – а именно – отвечает за списание материалов. В нашей системе материалы выбывают при передаче их в производство. Мы вполне можем создать новый документ копированием предыдущего и изменением некоторых его реквизитов. Так и поступим. Скопируем документ и приведем состав его реквизитов к показанному на рисунке 6.10.





Включим данный документ в состав подсистемы ОперативныйУчетМатериалов, вместо реквизита Контрагент у него будет*реквизит* ПолучательМатериалов с типом СправочникСсылка.Сотрудники.

В табличной части документа мы используем лишь два реквизита – это **Номенклатура** и **Количество**. Показатели стоимости списываемой номенклатуры мы будем рассчитывать автоматически. При работе с этим документом нас вполне устроит форма, генерируемая автоматически.

Форма нашего нового документа будет выглядеть так, как показано на рисунке 6.11.

Прове	ести и закрыть	🔚 📑 Провести	Все действия	. 🤅
номер:		000000001		
Дата:		02.10.2011 16:36:41		
Толучат	гель материалов:	Петров П. П. (Парикмахерская)		Q
Ответст	венный сотрудник:	Иванов И. И. (Парикмахерская)		Q
Организ	зация:			Q
🔂 Доб	бавить 🛛 🛃 🗙 🗍 -	<mark>⊪</mark>	Все дейс	твия ч
N	Номенклатура		Количество	
1	Одеколон		6	1,000
2	Духи			5,000
3	УФ-гель			7,000
	Олеколон			2 000

Рисунок 6.11. Документ ОтпускМатериаловМастеру в работе

Документы созданы, но сейчас они, во-первых, заполняются практически полностью вручную (за исключением поля Сумма в табличной части документа ПоступлениеМатериалов), а, во-вторых, нам сейчас довольно сложно будет понять, каков *остаток*материалов, числящихся за определенным сотрудником.

Единственный способ, которым можно сделать это сейчас – программно или "вручную" просмотреть все документы поступления материалов, выводя некоторые итоговые показатели. При заполнении документа отпуска материалов мы вынуждены заранее самостоятельно проверять остатки материалов *по* существующим документам, самостоятельно решать, какова цена этих материалов, контролировать остатки. Это не очень удобно, не производительно, и решением этих задач мы сейчас займемся.

Регистры накопления

Представим себе организацию, в которой все сведения о приходе материалов хранятся лишь в виде приходных документов. Для того, чтобы узнать количество и *стоимость* имеющихся в организации материалов, нам понадобится обращаться к документам, просматривать каждый из них, выписывая нужные данные, после чего суммировать их, получая нужные данные. Такой подход неудобен – он слишком медленный как для нашего воображаемого "ручного" случая, так и для автоматизированного учета.

Логичнее было бы, в *дополнение* к документам, вести специальные таблицы, в которые, при приеме материалов и при их списании, вносить краткие сведения об этом. Если, скажем, ежедневно, подводить итоги *по* этим таблицам и выводить остатки материалов, то, для того, чтобы сказать, сколько и каких материалов имеется в организации на определенную дату, достаточно обратиться к соответствующей графе таблицы. В системе «1С:Предприятие» такими таблицами являются *регистры накопления*. Как следует из названия, они предназначены для отражения, накопления, неких

показателей. И отражение в регистрах прихода и расхода материалов – один из типичных примеров их использования.

В нашей организации ведется учет материалов в привязке к ответственным лицам, которые их получают и с которых эти материалы списывают при отпуске в производство. Нам нужно хранить информацию о количестве материалов и об их стоимости, а так же, при списании в производство, иметь сведения о том, какому именно мастеру эти материалы переданы. Эти соображения позволяют нам спроектировать структуру *регистра накопления*, который мы сейчас будем создавать.

При планировании состава *регистра накопления* нужно понять, какие именно данные мы собираемся в нем хранить, после чего "разложить" эти данные *по* измерениям, ресурсам и реквизитам регистра.

Итак, нам нужно хранить следующие данные:

- Номенклатурная позиция
- Ответственный сотрудник, на котором числится данная позиция
- Количество номенклатуры
- Стоимость номенклатуры
- Данные о мастере, которому переданы материалы для использования.

Измерения регистра, или разрезы, в которых хранятся данные, позволяют нам ответить на вопросы о том, какие именно данные хранятся в регистре. В нашем случае нам нужно знать две основных характеристики – это, за каким ответственным лицомзакреплена та или иная номенклатурная позиция. Очевидно, измерениями из нашего списка данных будут номенклатурная позиция и ответственный сотрудник.

Ресурсы регистра – это всегда числовые значения, характеризующие хранимые данные. Числовые значения – это количество и сумма, и именно они будут ресурсами нашего регистра.

Реквизиты регистра играют вспомогательную роль, и, в нашем случае, логично будет в *реквизите* регистра хранить сведения о мастере, получившем материалы для работы – на тот случай, если нам понадобится узнать – кто именно эти материалы использовал.

Еще один важный вопрос, который нужно решить, проектируя *регистр*, заключается в том, будет ли этот *регистр* **регистром остатков** или **регистром оборотов**. Нас интересуют и сведения об остатках материалов, и сведения об оборотах, поэтому при настройке регистра следует указать вид регистра – Остатки. *Регистр* с видом **Остатки** позволяет нам работать и с остатками и с оборотами

Предложенная здесь структура *регистра накопления* – это лишь один из вариантов того, как можно организовать хранение описываемых данных. Подобную схему учета можно реализовать, скажем, с помощью пары регистров, один из которых используется исключительно для целей хранения суммовых остатков материалов – то есть, те данные, которые нужны для финансовых отчетов, другой – для хранения данных *по* центрам ответственности. В любом случае, каждая конкретная схема учета может потребовать и собственной структуры регистров, и наш пример – лишь демонстрация одного из возможных вариантов.

Обсудив теоретическую часть вопроса, перейдем к практике. Создадим новый *регистрр* накопления, назовем его**ОстаткиМатериалов**, *параметр* **Вид регистра** оставим в значении **Остатки**, Рисунок 6.12.

Основные	Имя	Остатки Материалов
Подсистемы	-	
Функциональные опции	Синоним:	Остатки материалов
Данные	Комментарий:	
Регистраторы		
Формы	Pue portuomo	Octoped
Команды	оид регистра	
Макеты		
Права	Представление	списка:
Обмен данными	2	
Прочее	Расширенное п	редставление списка:
	Пояснение:	

Рисунок 6.12. Регистр накопления Остатки Материалов

Включим регистр

накопления в

состав

подсистемы ОперативныйУчетМатериалов.

На вкладке Данные создадим следующие измерения, ресурсы и реквизиты: Измерения:

Имя: Номенклатура, Тип: СправочникСсылка.Номенклатура, Запрет незаполненных значений – установлено.

Имя: Ответственный Сотрудник, Тип: Справочник Ссылка. Сотрудники, Запрет незаполненных значений – установлено.

Ресурсы

Имя: Количество, Тип: число, *длина* 10, *точность* 3 Имя: Сумма, Тип: число, *длина* 10, *точность* 2 Рокрызиты:

Реквизиты:

Имя: ПолучательМатериалов, Тип: СправочникСсылка.Сотрудники

Обратите внимание на имена этих реквизитов, на их типы, а так же – на стандартные реквизиты регистра (рисунок 6.13.) – эти данные пригодятся нам при работе над процедурой *проведения документа*.

Исключим из состава реквизитов регистра общий *реквизит* **Организация**. Сейчас в нем нет необходимости. Для организации хранения данных в регистре в разрезе различных организаций нам понадобилось бы новое измерение – Организация, благодаря наличию которого мы смогли бы работать с материалами различных организаций.

Основные	◎ / × · · · □	
Функциональные опции	⊖ ↓. Измерения ↓ Номенклатира	
Данные	1. ОтветственныйСотрудник	
Регистраторы	😑 🧯 Ресурсы	
Формы	🕖 Количество	
Команды	🖉 Сумма	
Макеты	😌 🧁 Реквизиты	
Права	ПолучательМатериалов	
C Strange		
Обмен данными		_
Обмен данными Прочее	Регистр нак: Стандартные реквизиты	
Обмен данными Прочее	 Регистр нак: Стандартные реквизиты Стандартные реквизиты: Период 	
Обмен данными Прочее	 Регистр нак: Стандартные реквизиты Стандартные реквизиты: Период Регистратор 	
Обмен данными Прочее	 Регистр нак: Стандартные реквизиты Стандартные реквизиты: Период Регистратор НомерСтроки 	
Обмен данными Прочее	 Регистр нак: Стандартные реквизиты Стандартные реквизиты: Период Регистратор НомерСтроки Активность 	
Обмен данными Прочее	Регистр нак: Стандартные реквизиты Стандартные реквизиты: Период Регистратор НомерСтроки Активность ВидДвихения 	
Обмен данными Прочее	Стандартные реквизиты Период Период Э Регистратор НомерСтроки Э Активность ВидЛвижения	

Рисунок 6.13. Регистр накопления ОстаткиМатериалов, состав данных

Перейдем на вкладку Регистраторы окна редактирования объекта и выберем в качестве документов-регистраторов документы – ПоступлениеМатериалов и ОтпускМатериаловМастеру.

На данном этапе настройка *регистра накопления* окончена, перейдем к настройкам документов. Начнем с документа**ПоступлениеМатериалов**.

Откроем окно редактирования объекта для этого документа, перейдем на вкладку Движения (рисунок 6.14.) и нажмем на кнопкуКонструктор движений

Основные	Проведение:	Разрешить	
Подсистемы	Оперативное проведение:	Запретить	
Функциональные опции	Улаление лвижений:	Улалять автоматически	
Данные			
Нумерация	🙂 🙍 Регистры накопл	ения	
Движения			
Последовательности			
Журналы			
Формы			
Команды			
Макеты	Конструктор движен	иий	2
Ввод на основании			×
Права	Остаткиматериалов		
Обмен данными			
Прочее			
Прочее			

Рисунок 6.14. Документ Поступление Материалов, вкладка Движения

В конструкторе, выберем тип движения регистра – Приход, в *поле* Табличная часть укажем табличную часть документаМатериалы, нажмем на кнопку Заполнить выражения. Автоматический механизм установления соответствия между данными документа и регистра не всегда работает правильно (в том случае, если не может однозначно определить соответствия, или тогда, когда соответствие, определенное им *по* его логике, отличается от желаемого), поэтому проверим правильность установленных соответствий. В итоге окно Конструктора движения регистра должно выглядеть так, как показано на рисунок 6.15.

Конструктор движения ре	гистров			
😡 🔂 🗙 🋧 🗣				
Регистры		Реквизиты документа		
🕞 РегистрНакопления.Ост	аткиМатериалов	Дата		
Тип движения регистра: ()) Приход () Расход	 Номер Контрагент ОтветственныйСотру Комментарий Организация ТекСтрокаМатериал ТекСтрокаМатериал ТекСтрокаМатериал 	идник ы.НомерСтроки ы.Номенклатура	e •
Табличная часть: Материал	ы		<hasad< th=""><th>Далее></th></hasad<>	Далее>
Поле	Выражение		Заполнить вы	ражения
💪 Номенклатура	ТекСтрокаМатериалы.Но	оменклатура	Очистить вы	ражения
🛴 ОтветственныйСотруд	ОтветственныйСотрудни	ĸ		
🥖 Количество	ТекСтрокаМатериалы.Ко	оличество	01	
🥖 Сумма	ТекСтрокаМатериалы.Су	мма	UK	
😑 Получатель Материалов			Отмен	la
			Справ	ка

Рисунок 6.15. Конструктор движений

После нажатия на кнопку ОК, в модуле объекта документа будет сформирована такая процедура обработки проведения (так она выглядит после удаления комментариев о том, что код построен конструктором движений):

Процедура ОбработкаПроведения(Отказ, Режим) // регистр ОстаткиМатериалов Приход Движения.ОстаткиМатериалов.Записывать = Истина; Для Каждого ТекСтрокаМатериалы Из Материалы Цикл Движение = Движения.ОстаткиМатериалов.Добавить(); Движение.ВидДвижения = ВидДвиженияНакопления.Приход; Движение.Период = Дата; Движение.Поменклатура = ТекСтрокаМатериалы.Номенклатура; Движение.ОтветственныйСотрудник = ОтветственныйСотрудник; Движение.Количество = ТекСтрокаМатериалы.Количество; Движение.Сумма = ТекСтрокаМатериалы.Сумма; КонецЦикла; КонецПроцедуры Эта процедура объявлена в модуле объекта, она исполняется на *сервере*.

Установка свойства Движения. Остатки Материалов. Записывать в значение Истина говорит системе о том, что она должна записать в *регистр*движения, сформированные в процедуре. Движения заполняются в цикле обхода табличной части документа. Они будут физически записаны в *регистр* после того, как будет сформирован полный набор записей. Процедура проведения вполне может использовать и другие *механизмы* обращения к данным. Например, нужные данные могут быть получены с помощью запроса.

Так как мы проводим документ, отвечающий за приход товаров, *параметр* Движение.ВидДвижения устанавливается

в значениеВидДвиженияНакопления.Приход. Период устанавливается равным дате

документа (мы напрямую работаем с реквизитами документа). Остальные данные заполняются, опять же, либо из реквизитов документа, либо из реквизитов текущей строки табличной части.

Запустим конфигурацию в режиме «1С:Предприятие», проверим работу механизма на практике. Для этого перепроведем существующие документы **ПоступлениеМатериалов**, можем ввести и новые документы этого вида. При проведении или перепроведении данные из документов попадают в *регистр накопления* **ОстаткиМатериалов**, рисунок 6.16.

e) Q, Наяти &	nanti b						Все действия -	1
Териод 🎄	Репистратор	Номер стр.	Номенклатура	Ответственный сотрудник	Количество	Сумма	Получатель материалов	
+ 02.10.2011 15:24:52	Поступление материа	1	Духон	Иванов И. И. (Парикмах	10,000	1 290.00		
+ 02.10.2011 15:24:52	Поступление материа	2	УФгель	Иванов И. И. (Парикиах	3,000	900,00		
+ 02.10.2011 15:24:52	Поступление материа	3	Одеколон	Иванов И. И. (Парикмах	7,000	1 400,00		
+ 06.10.2011 0.17.29	Поступление материа	1	Духи	Васильев П. П. (Адианы.	15,000	1 860,00		
+ 06.10.2011 0.17:29	Поступление материа	2	УФ-гель	Васильев П. П. (Адиини_	3.000	900.00		
+ 06.10.2011 0.17:29	Поступление материа	3	Одеколон	Васильев П. П. (Админи	2,000	400,00		
+ 06.10.2011 0.23:43	Поступление материа	1	Духон	Иванов И. И. (Парикиах	11,000	1 749,00		
+ 06.10.2011 0.23:43	Поступление материа	2	УФгель	Иванов И. И. (Парикмах	5,000	1 550,00		
+ 06 10 2011 0 22 42	Поступление матегиа	3	Олеколон	Measure M. M. (Tansensor	9.000	1 710 00		

Рисунок 6.16. Данные в регистре накопления

Обратите внимание на то, что *регистры накопления*, как объекты, которые не предназначены изначально для "ручной" работы пользователя, не выводятся в командном интерфейсе даже при указании подсистем, в которые они входят. Нам, при разработке, понадобится просматривать регистры.

Для того, чтобы открыть окно регистра можно либо воспользоваться командой **Главное меню > Все функции** и в появившемся окне **Все функции** найти нужный *регистр*, либо открывать его с помощью команды в интерфейсе, предварительно самостоятельно добавив эту команду в нужный раздел интерфейса.

При записи данных о приходе материалов нам, в нашем случае, нет нужды в какихлибо дополнительных проверках вводимых данных, поэтому нас вполне устроит стандартная процедура проведения.

Следующим документом, проведение которого мы хотим организовать, является документ, списывающий материалы – Отпуск Материалов Мастеру. Им мы займемся в следующей лекции, а сейчас построим отчет Остатки Материалов.

Отчет ОстаткиМатериалов, введение в СКД

Ранее мы рассматривали построение простейшего отчета самостоятельно формируя *макет* отчета, код вывода данных. Такой подход имеет право на жизнь, им нужно владеть, *по* крайней мере, на тот случай, если вам придется редактировать отчет в какой-либо конфигурации, где применяется такой подход. Однако, наиболее правильным способом построения отчетов в системе «1С:Предприятие» 8.2. является использование *системы компоновки данных*, или, сокращенно, СКД. Начиная с этой лекции мы будем строить отчеты именно с использованием СКД.

Итак, прежде чем строить отчет, поймем, чего мы ждем от этого отчета.

Нам хотелось бы получать сведения о количественных и суммовых остатках номенклатуры *по* ответственным лицам на заданную дату. Эти данные мы можем найти в регистре накопления **ОстаткиМатериалов**.

Добавим в *дереве конфигурации* новый отчет, назовем его ОстаткиМатериалов. Включим его в состав подсистемыОперативныйУчетМатериалов.

Ha закладке Основные нажмем кнопку увеличительным на с стеклом в поле Основная схема компоновки данных. Появится окно конструктора макета, где имя по умолчанию можем задать имя (нас устроит ΜЫ ОсновнаяСхемаКомпоновкиДанных), тип макета ограничен единственным – Схема компоновки данных. Нажмем в этом окне Готово и попадем в окноконструктора СКД. Здесь нам, в первую очередь, нужно добавить новый источник данных, в нашем случае это будет Запрос.рисунок 6.17.

Отчет Отчет 1: ОсновнаяСкемаКомпоно	еки Данных					_ 🗆	3
Наборы данных Связи наборов данных	Вычисляемые поля	Ресурсы	Параметры	Макеты	Вложенные схемы	Настройки	
<u>н- ж</u>	18					0	×
Добавить набор данных - запрос	ax						
🖢 Добавить набор данных - объект							
Добавить набор данных - объединение							
b 820							i

Рисунок 6.17. Добавление нового набора данных – запроса

Когда набор данных, названный **НаборДанных1**, будет добавлен, мы можем нажать на кнопку **КонструкторЗапроса**, находящуюся над полем **Запрос** в нижней части окна. Это приведет к открытию окна конструктора запроса.

Из виртуальной таблицы *регистра накопления* ОстаткиМатериалов выберем следующие поля, рисунок 6.18.

- Номенклатура
- ОтвественныйСотрудник
- КоличествоОстаток
- СуммаОстаток Конструктор запроса * Таблицы и поля Группировка Условия Дополнительно Объединения/Пс... Порядок Компонсека данн... Характеристики 物語クス語う 0 0 O / X Поля База данн Таблицы > X Константы 😑 🚮 Остатки МатериаловОстатки 1. Остатки Материалов Остатки. Номенклатура Э П Справочники 1. Остатки/МатериаловОстатки.ОтветственныйСотрудник Э. 1. Номенистатура 22 22 Документы В 1. ОтветственныйСотрудник Остатки Материалов Остатки. Количество Остаток. ¢. < Перечисления КоличествоОстаток Остатки/МатериаловОстатки.СуммаОстаток. CC. 10 СумиаОстаток РегистрыНакопления 🗄 🔳 Остатки Материалов 🗄 🧃 Остатки Материалов Обороты I Потатки Материалов Остат Э Стато/Материалов Остато/ИОбороты с назва Далее > ОК Отмена Справка 3anpoc

Рисунок 6.18. Создание запроса

На этом работа с конструктором запроса завершена – остальные настройки мы будем делать в конструкторе СКД. Благодаря установленному *по* умолчанию флагу **Автозаполнение**, на вкладке **Наборы данных** после создания запроса мы можем видеть заполненный *список* полей, рисунок 6.19. – с этими полями мы сможем работать при создании отчета.



Рисунок 6.19. Автозаполнение списка полей на закладке Наборы данных

Переместимся в окне редактора СКД на вкладку Ресурсы, из списка Доступные поля перенесем в список, находящийся в правой части окна, поля, по которым можно вычислять итоги. В нашем случае это поля КоличествоОстаток и СуммаОстаток. Поумолчанию этим полям в поле Выражение будет назначена агрегатная функция Сумма, нас устроит такое положение дел, рисунок 6.20.

аооры данных	Связи наборов данных	Ber	иисляемые поля	Ресурсы	Параметры	Макеты	Bnox	сенные скемы	Настройки	1
оступные поля		1	Flone		Выражение			Рассчитывать	NO	
- КоличествоОстаток		2	- КоличествоОс	таток.	Сумма(КоличествоОстаток)		k)			
— Номенклатура		22	= СуммаОстато	ĸ	Сумма(СуммаОстаток)					
 Ответственный Сотрудник. 		1								
= СумиаОстаток										
		(22)								

Рисунок 6.20. Настройка состава ресурсов отчета

Теперь займемся настройкой внешнего вида отчета.

Перейдем на закладку **Настройки**, вызовем кнопкой с соответствующим названием **Конструктор настроек** и выберем на его первой странице тип отчета – **таблицу**. Нажмем на кнопку **Далее** и в следующем окне выберем поля, которые будут отображаться в отчете в следующем порядке (рисунок 6.21.):

- Номенклатура
- ОтветственыйСотрудник
- КоличествоОстаток
- СуммаОстаток

Конструктор настроек компоно Поле Выберите поля, которые	овки данных е будут отображаться в отчете.	×
Оклад Доступные поля </th <th>Image: Point of the image: Pointof the image: Point of the image: Point of</th> <th></th>	Image: Point of the image: Pointof the image: Point of the image: Point of	
< Назад	Далее > ОК Отмена Спр	авка

Рисунок 6.21. Выбор полей, которые будут отображаться в отчете

Нажмем кнопку Далее, в следующем окне конструктора, служащим для настройки группировки таблиц, в группу Строки добавим*поле* Номенклатура, в *поле* Колонки – ОтветственныйСотрудник. Тип группировки оставим в состоянии Без иерархии. В следующем окне конструктора, который позволяет задать упорядочение отчета, зададим упорядочивание *по* полюНоменклатура, *по* возрастанию, рисунок 6.22.



Рисунок 6.22. Настройка упорядочивания отчета

Нажмем **ОК**, в отчет будет добавлена новая *таблица*. В нижней части формы конструктора СКД, на закладке **Параметры**, выделим*параметр* **Период** и нажмем на кнопку Свойства элемента пользовательских настроек. В появившемся окне установим флагВключать в пользовательские настройки, режим редактирования оставим в значении Быстрый доступ, Рисунок 6.23. Это позволит нам вывести данный *параметр* в форму отчета, позволит пользователю выбирать нужный период перед построением отчета

Отчет Остатки Материалов:	ОсновнаяСхемаКомпоновкиДанных		- 0
нароры данных 🔢 Связи нарор	defaulter defaulter for a second laboration and the second labor	етры Макеты вложенные схемы	Настроики
Имя варианта Представление Основной Основной	 Строки Строки Строки Строки Колонки Колонки ОтветственныйСотрудник 		
	Настройки: Отчет Параметры Выбранные п Отбор Сортиров Отображать недоступные параметры (устанавливать знач	ка Условное оф Пользовател	Другие настр ся) //
	Параметр	Значение Дата	3
	Пермод	Произвольная дата	6
	Пользовательские настройки эле Включать в пользовательские настро Представление Режим редактирования	мента ×	ţe

Рисунок 6.23. Настройка вывода параметра

Запустим систему в режиме «1С:Предприятие» и построим отчет, рисунок 6.24.

	Сформировать	Настройка						Все действия •
•	1 Период				07.10.2011 0:00	:00		
Ĩ	Параметры: Г	Териод: 07.10.2	011 0:00:00					
	Номенклатура	Васильев П. І (Администрац	T. มหล)	Иванов И. И. Итого (Парикмахерская)				
		Количество Остаток	Сумма Остаток	Количество Остаток	Сумма Остаток	Количество Остаток	Сумма Остаток	
	Духи	15,000	1 860,00	21,000	3 039,00	36,000	4 899,00	
	Одеколон	2,000	400,00	16,000	3 110,00	18,000	3 510,00	
	УФ-гель	3,000	900,00	8,000	2 450,00	11,000	3 350,00	
	Итого	20,000	3 160,00	45,000	8 599,00	65,000	11 759,00	

Рисунок 6.24. Готовый отчет по количественным и суммовым остаткам материалов

Перед построением отчета, если мы хотим задать *параметр* **Перио**д, установим флаг перед этим параметром и выберем нужную дату.

Литература:

Официальный сайт интернет-университета «Интуит» (электронный ресурс), режим доступа http://www.intuit.ru/studies/courses/2318/618/lecture/17939.

Контрольные вопросы для самопроверки

- 1. Какие виды регистров вы знаете?
- 2. Для чего нужны документы в «1С:Предприятие»?
- 3. Как автоматизировать работу с документами?

Задание к лабораторной работе

Создать документы и регистры в системе в соответствии с порядком, отраженным в теоретической части.

Методические указания и порядок выполнения работы

Работа выполняется в точном порядке, как это указано на рисунках и в теоретической части.

Индивидуальное задание

не предусмотрено.

Требования к отчету и защите

1. Результатом выполнения лабораторной работы является сформированный в программе файл, содержащий выполненные задания. В ЭИОС результаты работы не выкладываются.

2. Планируется защита работы, где студент комментирует порядок выполнения заданий, а также отвечает на вопросы, представленные выше.

ЛИТЕРАТУРА

1. Рудинский, И.Д. Технология проектирования автоматизированных систем обработки информации и управления: учеб. пособие / И. Д. Рудинский . - Москва: Горячая линия, 2011. - 303 с.

2. Советов, Б.Я. Теоретические основы автоматизированного управления: учеб. / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. - Москва: Высшая школа, 2006. - 462 с.

3. Хетагуров, Я.А. Проектирование автоматизированных систем обработки информации и управления (АСОИ и У): учеб. / Я. А. Хетагуров. - Москва: Высшая школа, 2006. - 223 с.

4. Малюк, А.А. Введение в защитуинформации в автоматизированныхсистемах: учеб. пособие / А. А. Малюк, С. В. Пазизин, Н. С. Погожин. - 3-е изд., стер. - Москва : Горячая линия-Телеком, 2005. - 146 с.

5. Интеллектуальные системы управления организационно-техническими системами / А. Н. Антамошин [и др.]. - Москва: Горячая линия-Телеком, 2006. - 160 с.

Локальный электронный методический материал

Марина Викторовна Соловей

АРХИТЕКТУРА И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ МАЛЫХ И СРЕДНИХ ПРЕДПРИЯТИЙ

Редактор Г. А. Смирнова

Уч.-изд. л. 6,7. Печ. л. 6,7

Издательство федерального государственного бюджетного образовательного учреждения высшего образования «Калининградский государственный технический университет». 236022, Калининград, Советский проспект, 1