

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Пономарева Светлана Викторовна
Должность: Проректор по УР и НО
Дата подписания: 20.09.2023 17:31:00
Уникальный программный ключ:
bb52f959411e64617366ef2977b97e87139b1a2d



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

(ДГТУ)

Авиационно-технологический колледж

УТВЕРЖДАЮ

Директор колледжа

_____ В.А. Зибров

«__» _____ 2023г

Методические рекомендации (указания)
по проведению практических занятий по дисциплине
ОП.16. Программирование в 1С
для студентов специальности очной формы обучения по специальности
09.02.07 Информационные системы и программирование

Методические рекомендации (указания) по проведению практических занятий по дисциплине ОП.16. Программирование в 1С для студентов специальности очной формы обучения разработаны на основе актуализированного Федерального государственного образовательного стандарта по специальности (специальностям) среднего профессионального образования 09.02.07 Информационные системы и программирование.

В методические рекомендации включены лабораторные работы по программированию в среде системы «1С: Предприятие», направленные на изучение основных объектов и механизмов платформы, приобретение начальных навыков конфигурирования и программирования в системе «1С: Предприятие», разработки несложных приложений.

ПРАКТИЧЕСКАЯ РАБОТА №1

Тема: Создание информационной базы. Подсистемы.

Цель работы: Научиться создавать ИБ и подсистемы.

Ход выполнения работы

1. Запустите 1С:Предприятие. В открывшемся диалоге вы увидите список информационных баз, с которыми вы работаете. Если этот список пуст, система сама предложит вам создать новую базу. Для создания новой базы нажмите кнопку **Добавить**

2. В открывшемся диалоге выберите пункт Создание новой информационной базы. Нажмите кнопку **Далее**. На следующем шаге выберите пункт **Создание информационной базы без конфигурации...**

3. Нажмите кнопку **Далее**. На следующем шаге задайте наименование вашей информационной базы (Фамилия Имя и группа) и выберите тип ее расположения **На данном компьютере...**

4. Нажмите кнопку **Далее**. На следующем шаге укажите каталог для расположения вашей информационной базы. Язык по умолчанию установлен в значение **Русский**.

5. Нажмите кнопку **Далее**. На следующем шаге нажмите кнопку **Готово**.

6. В диалоге запуска 1С:Предприятия, в списке информационных баз, вы увидите созданную вами новую пустую базу.

7. Запустить 1С: Предприятие в режиме Конфигуратор. Для этого нажмем кнопку **Конфигуратор** в диалоге запуска системы. Сразу под заголовком окна находится главное меню конфигуратора, содержащее пункты **Файл**, **Правка**, **Конфигурация**, **Администрирование** и т. п. В каждом из этих пунктов содержится много подпунктов, вызов которых обеспечивает выполнение различных действий конфигуратора.

Ниже находится панель инструментов конфигуратора, в которую в виде кнопок-пиктограмм помещены наиболее часто используемые действия, вызываемые из меню. Таким образом, одни и те же действия можно выполнить двумя разными способами: вызвав определенный пункт меню или нажав соответствующую ему Кнопку на панели инструментов.

8. Выполните первую команду, с которой начинается работа с любой конфигурацией, - откроем конфигурацию с помощью пункта меню **Конфигурация ► Открыть конфигурацию**. Для того чтобы открыть и закрыть конфигурацию, следует использовать пункты меню **Конфигурация ► Открыть конфигурацию** и **Конфигурация ► Закрыть конфигурацию** или соответствующие им кнопки на панели инструментов.

9. На экране откроется *дерево объектов конфигурации*. Это окно вы можете закрыть, как любое другое окно, при этом конфигурация останется открытой (то есть доступной для редактирования). Чтобы снова отобразить на экране окно дерева конфигурации, следует воспользоваться командой меню **Конфигурация ► Окно конфигурации**.

10. Выделите в дереве объектов конфигурации корневой элемент **Конфигурация** и двойным щелчком мыши откроем его палитру свойств. Задайте имя конфигурации **ПособиеДляНачинающих**. Соответствующий ему синоним устанавливается автоматически, но его можно изменить по своему усмотрению. В дальнейшем именно его мы будем видеть в рабочем окне 1С:Предприятия.

11. Проверим первые изменения в режиме 1С Предприятие. Для этого выполним пункт меню **Отладка ► Начать отладку**. Система сама анализирует наличие изменений в конфигурации и выдает соответствующий вопрос об обновлении конфигурации базы данных.

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

19. Первое, с чего следует начинать разработку конфигурации, - это проектирование состава подсистем.

20. Создать в конфигурации пять новых объектов конфигурации Подсистема, которые будут иметь имена: **Бухгалтерия**, **РасчетЗарплаты**, **УчётМатериалов**, **ОказаниеУслуг** и **Предприятие**. Чтобы это сделать, выполните следующие действия:

- раскрыть ветвь **Общие** в дереве объектов конфигураций, нажав на + слева от нее;
- выделить ветвь **Подсистемы**, вызвать ее контекстное меню и выбрать пункт **Добавить**.

После этого система откроет *окно редактирования объекта конфигурации*. Оно предназначено специально для сложных объектов конфигурации и позволяет путем выполнения последовательных действий быстро создавать такие объекты. При открытии окна редактирования объекта конфигурации мы попадаем на закладку **Основные**. Чтобы изменить свойства объекта в процессе разработки часто бывает нужно повторно открыть окно редактирования объекта конфигурации. Для этого следует выделить нужный элемент в дереве объектов конфигурации и нажать в командной панели окна конфигурации кнопку **Изменить текущий элемент** или дважды щелкнуть мышью по выделенному элементу.

- задайте *имя* подсистемы - **Бухгалтерия**. На основании имени платформа автоматически создаст синоним - **Бухгалтерия**. Имя объекта является уникальным и служит для обращений к свойствам и методам объекта на встроенном языке. Свойство **Синоним** также есть у любого объекта конфигурации. Оно предназначено для хранения «альтернативного» наименования объекта конфигурации в элементах интерфейса нашей программы, то есть будет показано пользователю. Поэтому на синоним практически нет никаких ограничений, и его можно задавать в привычном для человека виде.

В целях совершенствования интерфейса приложения можно также задать картинку для отображения подсистемы. Нажмем кнопку выбора в поле **Картинка**. Отсутствие картинки у подсистемы не препятствует отображению раздела в интерфейсе. В этом случае рядом с названием раздела отображается стандартная картинка по умолчанию.

21. Добавим подсистему Предприятие для доступа к административным и сервисным функциям. Выделить корень дерева объектов конфигурации ПособиеДляНачинающих, нажатию, правой кнопки мыши вызвать контекстное меню и выбрать пункт **Открыть командный интерфейс конфигурации**.

22. В открывшемся окне **Командный интерфейс** вы увидите список созданных вами подсистем (разделов приложения). С помощью кнопок **Вверх**, **Вниз** измените порядок расположения разделов в этом списке. Расположите сначала подсистемы, отражающие производственную деятельность фирмы: **Учет материалов** и **Оказание услуг** затем

бухгалтерскую деятельность и расчет зарплаты сотрудников **Бухгалтерия** и **Расчет зарплаты**, а затем подсистему **Предприятие**.

23. Запустите 1С: Предприятие в режиме отладки и увидите, что порядок расположения подсистем в панели разделов приложения изменился.

24. Закройте приложение и вернитесь в конфигуратор.

25. После успешного завершения каждого занятия рекомендуется сохранять конфигурацию, выполнив команду главного меню **Администрирование ► Выгрузить информационную базу**. Это полезно на случай, если вы запутаетесь в своих действиях и захотите вернуться к работающему варианту. Это можно сделать, выполнив команду **Администрирование ► Загрузить информационную базу**

Контрольные вопросы:

1. Что такое конфигурируемость системы 1С: Предприятие.
2. Из каких основных частей состоит система.
3. Что такое платформа, и что такое конфигурация
4. Для чего используются разные режимы запуска 1С: Предприятие.
5. Что такое дерево объектов конфигурации.
6. Что такое объекты конфигурации.
7. Что создает система на основе объектов конфигурации.
8. Какими способами можно добавить объект конфигурации.
9. Зачем нужна палитра свойств.
10. Как запустить 1С: Предприятие в режиме отладки.
11. Для чего используется объект конфигурации Подсистема.
 12. Как описать логическую структуру конфигурации при помощи объектов Подсистема.
 13. Как управлять порядком вывода и отображением подсистем в конфигурации.
 14. Что такое окно редактирования объекта конфигурации и в чем его отличие от палитры свойств.

ПРАКТИЧЕСКАЯ РАБОТА № 2

Тема: Создание Справочников.

Цель работы: Научиться создавать справочники.

Ход выполнения работы

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

Пользователь в процессе работы может самостоятельно добавлять новые элементы в Справочник: например, добавить новых сотрудников, создать новый товар или внести нового клиента. В базе данных каждый элемент справочника представляет собой отдельную запись в основной таблице, хранящей информации из этого справочника.

Каждый элемент справочника содержит некоторую дополнительную информацию, которая подробнее описывает этот элемент. Набор такой информации является одинаковым для всех элементов справочника, и для описания такого набора используются *реквизиты* объекта конфигурации Справочник, которые также, в свою очередь, являются объектами конфигурации. Поскольку эти объекты конфигурации логически связаны с объектом Справочник, они называются *подчиненными* этому объекту.

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

2. Задать представление в единственном числе - **Клиент**. Представление **объекта** определяет название объекта в единственном числе и используется в названии стандартной команды, например, команды создания объекта - **Клиент**: создать. Представление объекта нужно задавать тогда, когда, синоним объекта конфигурации задан во множественном числе, или когда он описывает множество объектов.

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

4. Нажмите кнопку **Далее** и перейдите на закладку **Подсистемы** окна редактирования объекта конфигурации Справочник. На этой закладке определяется, в каких подсистемах будет отображаться данный справочник. Список клиентов должен быть доступен в разделе **Оказание услуг**, так как оказываемые услуги относятся к определенному клиенту.

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

1. Перейдите к длине наименования и увеличьте длину наименования до 50.

7. Прежде чем запускать 1С: Предприятие, настроить интерфейс приложения, чтобы было удобнее вводить новые элементы справочника. Команда для открытия списка справочника, как и команда для создания его новых элементов, добавляется в интерфейс тех разделов (подсистем), в которых будет отображаться справочник. Но команда создания новых элементов по умолчанию невидима в интерфейсе приложения. Это объясняется тем, что возможность просматривать списки справочника нужна, как правило, всегда. А возможность создания новых элементов справочника используется не так часто.

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

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

9. Закрыть окно редактирования справочника **Клиенты** и запустите **1С: Предприятие** в режиме отладки. Ответить утвердительно на запрос конфигуратора об обновлении конфигурации и увидим окно, содержащее список изменений в структуре конфигурации, автоматически сгенерированный платформой. В данном случае мы добавили справочник **Клиенты**. Нажмите кнопку **Принять**. Перед нами откроется окно системы в режиме **1С: Предприятие**.

У раздела **Бухгалтерия** нет панели действий, так как для этой подсистемы мы не устанавливали видимость команды создания новых элементов из группы команд **Создать**. Кроме этого, нет ни одной другой видимой команды из групп **Отчеты** или **Сервис**. Соответственно, раз нет команд, входящих в группу, значит невидима вся группа, а раз нет ни *одной* группы, значит невидима вся панель действий.

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

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

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

13. Закрыть **1С: Предприятие** и вернуться в режим конфигуратор. Чтобы посмотреть состав и свойства стандартных реквизитов справочника, в окне редактирования объекта конфигурации **Справочник Клиенты** на закладке **Данные** нажать кнопку **Стандартные реквизиты**. Выделить в списке реквизит **Наименование** и вызвать его контекстное меню и выбрать пункт **Свойства**. В палитре свойств реквизита **Наименование** установлено свойство **Проверка заполнения по умолчанию** значение выдавать ошибку.

14. Перейти к созданию второго Справочника - **Сотрудники**. На основании имени платформа автоматически заполнит его синоним.

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

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

17. Создать реквизиты табличной части **ТрудоваяДеятельность**. Для этого нажмем кнопку **Добавить реквизит** над списком табличной частей справочника **Добавим** следующие реквизиты:

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

Для реквизитов **НачалоРаботы** и **Окончание Работы** выбрать состав даты - **Дата**, поскольку в системе

1С:Предприятие 8 значения типа **Дата** содержат как дату, так и время.

19. Сделайте видимой в панели действий подсистемы **РасчетЗарплаты** стандартную команду для создания новых сотрудников. Для этого в дереве объектов конфигурации выделить ветвь **Подсистемы**, вызвать ее контекстное меню и выберем пункт **Все подсистемы**. В открывшемся окне слева в списке **Подсистемы** выделить подсистему **РасчетЗарплаты**. Справа в списке **Командный интерфейс** отразятся все команды выбранной подсистемы.

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

24. На этом создание справочника **Сотрудники** завершено. Закрыть окно редактирования справочника **Сотрудники** и запустить **1С: Предприятие** в режиме отладки.

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

25. Создадим следующих сотрудников:

1. **Гусаков Николай Дмитриевич**.

Трудовая деятельность: Организация - ЗАО «НТЦ», Начало работы - 01.02.2000, Окончание работы - 16.04.2003, Должность - Ведущий специалист.

2. Деловой Иван Сергеевич.

Трудовая деятельность:

а) Организация ООО «Автоматизация», Начало работы -22.01.1996, Окончание работы -31.12.2002, Должность - Инженер;

б) Организация - ЗАО «НПО СпецСвязь» Начало работы - 20.06.1986, Окончание работы - 21.01.1995, Должность - Начальник производства;

3. Симонов Валерий Михайлович.

Трудовая деятельность: Организация - ООО «СтройМастер», Начало работы - 06.02.2001, Окончание работы- 03.04.2004, Должность - Прораб.

26. Приступить к созданию следующего справочника Номенклатура. Справочник Номенклатура будет содержать информацию об оказываемых услугах и о тех материалах, которые при этом могут быть использованы. Единственная особенность которой он будет обладать, - наличие иерархической структуры. Для того чтобы справочником было удобно пользоваться, сгруппируем услуги в одну группу, а материалы - в другую.

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

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

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

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

33. В группе **Панель действий.Создать** включить видимость у команды **Номенклатура: создать**.

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

34. Закроем окно редактирования справочника **Номенклатура** и запустим 1С: Предприятие в режиме отладки. В открывшемся окне 1С Предприятия мы видим, что в панели навигации разделов **Учет материалов, Оказание услуг и Бухгалтерия** появилась команда **Номенклатура**. Название этой команды определяется синонимом объекта, так как - других представлений для этого справочника не задавали.

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

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

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

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

39. Раскроем группу **Услуги** и тоже создадим в ней несколько элементов - услуги по ремонту телевизоров.

- Диагностика,
- Ремонт отечественного телевизора,
- Ремонт импортного телевизора.

И услуги по установке стиральных Машин:

- Подключение воды,
- Подключение электричества.

40. **Перенос элементов в другие группы**. Теперь разнести услуги по двум смысловым группам: услуги по ремонту телевизоров и услуги по установке стиральных машин. Для этого в группе **Услуги** создайте еще две группы: **Телевизоры** и **Стиральные машины**. Для того чтобы переместить услуги в соответствующие группы в окне списка установим курсор на ту услугу, которую мы хотим переместить, и выполним команду **Все действия ► Переместить в группу**. В открывшемся окне выберем новую группу. Или же можно мышью перетащить выделенный элемент справочника в нужную группу. Можно, также открыть для редактирования выделенный элемент справочника и изменить поле **Родитель**.

- Переместить в группу **Телевизоры** услуги **Диагностика, Ремонт отечественного телевизора** и **Ремонт импортного телевизора**.
- Услуги **Подключение воды** и **Подключение электричества** переместить в группу **Стиральные машины**.

41. В группе **Материалы** создать две группы; **Радиодетали** и **Прочее**. В группу **Прочее** поместить **Кабель электрический** и **Шланг резиновый**. Остальные материалы переместить в группу **Радиодетали**. Переключившись в представление списка в

виде дерева по команде **Все Действия - Режим просмотра - Дерево** можно увидеть список номенклатуры в виде дерева.

42. **Справочник с предопределенными элементами.** Создать справочник Склады. Этот справочник будет включать в себя один предопределенный элемент - склад **Основной**, на который будут поступать все материалы.

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

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

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

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

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

48. В группе Панель действий.Создать включить видимость у команды **Склад: создать**. Также мы видим, что в группу Панель навигации.Обычное добавилась команда Склады для открытия этого списка. Она включена' по умолчанию. Закроем окно редактирования справочника Склады и запусти 1С Предприятие в режиме отладки. Ответим утвердительно на запрос конфигулятора об обновлении конфигурации и увидим окно, содержащее список изменений в структуре конфигурации, автоматически сгенерированный платформой. В данном случае мы добавили справочник **Склады**.

52. В открывшемся окне 1С Предприятия мы видим, что в панели действий раздела **Учет материалов** появилась команда **Склад** для создания новых складов. Название этой команды определяется свойством Представление объекта, которое мы задали для этого справочника. Также в панели навигации разделов **Оказание услуг и Учет материалов** появилась команда **Склады** для открытия списка складов. Название этой команды определяется синонимом объекта, так **Представление списка** мы для этого справочника не задавали.

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

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

Контрольные вопросы

1. Для чего предназначен объект конфигурации Справочник? Каковы характерные особенности справочника?
2. Для чего используются реквизиты и табличные части справочника?
- 3.Зачем нужны иерархические справочники и что такое родитель?
- 4.Зачем нужны подчиненные справочники и что такое владелец?
- 5.Какие основные формы существуют у справочника?
- 6.Что такое предопределенные элементы? Чем с точки зрения конфигурации отличаются обычные элементы справочника от предопределенные элементов?
- 7.Как пользователь может отличить обычные элементы справочника от предопределенных?
8. Как изменить конфигурацию базы данных?
9. Как связаны объекты конфигурации и объекты базы данных?
10. Что такое подчиненные объекты конфигурации?
11. Зачем нужна проверка заполнения у реквизитов справочника?
12. Что такое быстрый выбор и когда его использовать?

ПРАКТИЧЕСКАЯ РАБОТА № 3

Тема работы: Создание документов.

Цель работы: Научиться создавать различные документы.

Ход выполнения работы

1. Открыть конфигуратор и добавить новый объект конфигурации **Документ**. На закладке **Основные** задать имя документа - **ПриходнаяНакладная**. На основании имени платформа автоматически заполнит его синоним. Здесь же

определим, как будет представлен документ в интерфейсе 1С:Предприятия. **Представление объекта** задавать не будем, вместо него будет использоваться **Синоним** объекта. Это нам вполне подходит, так как мы задали его имя в единственном числе. **Представление списка**, задать во множественном числе как **Приходные накладные**.

2. Перейдем на закладку **Подсистемы**. По логике нашей конфигурации список приходных накладных должен быть доступен в разделах **Учет материалов** и **Бухгалтерии**. Отметьте в списке подсистем эти подсистемы.

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

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

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

6. Установите свойство **Проверка заполнения** в значение **Выдавать ошибку**. Тем самым мы задаем условие, что документ **Приходная накладная** обязательно должен содержать табличную часть, то есть список приходимых материалов. Иначе будет выдано сообщение об ошибке, и документ не будет сохранен.

7. Создайте реквизиты табличной части **Материалы**:

Материал, тип **СправочникСсылка.Номенклатура**;

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

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

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

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

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

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

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

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

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

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

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

16. Аналогичным образом создайте второй документ, который будет приходовать на **Основной** склад материалы для установки стиральных машин. Но теперь не будете использовать кнопку выбора в поле **Материал**, а просто начните вводить название материала в это поле. Платформа автоматически найдет материалы, найдет материалы название которых начинается с введенных символов, и предложит их нам для выбора. Нажмите **Провести и закрыть**.

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

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

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

20. Этот инструмент также построен по принципу «мастеров»: ввод данных в определенной последовательности и передвижение кнопками **Далее** и **Назад**. Выберем тип формы **Форма документа** и нажмем кнопку **Готово**, согласившись тем самым со всем, что нам предложила система. Обратите внимание, что в дереве объектов конфигурации у объекта

фигурации Документ ПриходнаяНакладная появилась ФормаДокумента, а на экране открылось окно редактора, содержащее эту форму.

Редактор форм объединяет несколько окон взаимосвязанных между собой редакторов. Необходимо понимать, что при разработке форм объектов конфигурации разработчик не имеет возможности «нарисовать» форму. Он может только указать, из каких элементов будет состоять форма, а система уже сама самостоятельно расположит эти элементы в форме.

Элементы формы в верхнем левом окне редактора форм образуют иерархическую структуру, из которой следует, что чем выше в списке находится элемент, тем выше и левее на форме он будет располагаться. Эта структура редактируется на закладке **Элементы** и позволяет управлять отображением и редактированием данных в форме.

На основе описания в конфигурации документа **ПриходнаяНакладная** система создала структуру элементов, которая определяет, как будет выглядеть форма. Эти элементы имеют разное назначение и разное поведение. Однако все они служат для того, чтобы отображать информацию, хранящуюся в базе данных, и организовывать интерактивную работу с этой информацией. Вы можете попробовать перетащить мышью поля в дереве элементов и поменять местами, например, реквизиты табличной части. Результат изменений сразу отразится в форме документа в нижней части редактора форм. При этом разработчику не нужно задумываться над вопросами конкретного расположения того или иного элемента формы, его размеров и привязки к другим элементам. Эту работу берет на себя система. Разработчик может через палитру свойств изменить свойства элемента, которые повлияют на его отображение в форме. Он может также изменить структуру элементов формы - создать новое поле, группу полей, добавить табличную часть, связав эти элементы данными формы.

Необходимо, чтобы каждый раз, когда меняется значение в поле **количество** или в поле **Цена**, в поле **Сумма** автоматически устанавливалось значение, равное **Количество*Цена**. Для этого нужно написать на встроенном языке команду, похожую на **Сумма = Количество*Цена**, которая будет выполняться при изменении значения поля **Количество** или **Цена**.

Обработчик события

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

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

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

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

Листинг 1. Процедура «МатериалыКоличествоПриИзменении()»

СтрокаТабличнойЧасти = Элементы.Материалы.ТекущиеДанные;

СтрокаТабличнойЧасти.Сумма = СтрокаТабличнойЧасти.Количество * СтрокаТабличнойЧасти.Цена;

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

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

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

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

Получив этот объект, мы можем обратиться к данным конкретной колонки табличной части, указав имя колонки в качестве свойства объекта. Например, используя обращение **СтрокаТабличнойЧасти.Количество** мы получаем число, которое находится в редактируемой строке в колонке **Количество**.

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

В режиме 1С:Предприятие

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

23. Поместим расчет суммы в некоторое «общедоступное» место, чтобы разные документы, имеющие аналогичные реквизиты табличной части, могли использовать этот алгоритм. Для описания таких «общедоступных» мест служат объекты конфигурации **Общий модуль**, расположенные в ветке **Общие** ► **Общие модули**. Процедуры и функции, содержащиеся в

этих модулях, могут быть доступны для любых объектов конфигурации. Поэтому создадим общий модуль и перенесем в него нашу процедуру расчета суммы. А в документе просто оставим вызовы этой процедуры из общего модуля.

24. Добавить объект конфигурации **Общий модуль**. Для этого раскрыть ветвь **Общие** в дереве объектов конфигурации, нажав + слева от нее. Затем выделить ветвь **Общие модули** и нажмите кнопку **Добавить** в командной панели окна конфигурации. Назвать его **РаботаСДокументами** и установить в его свойствах флажок **Клиент(управляемое приложение)**, а флажок **сервер** снять. Это означает что экземпляры этого модуля будут скомпилированы в контексте тонкого клиента и в контексте веб-клиента.

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

Листинг 2. Процедура «**РассчитатьСумму()**»

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

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

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

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

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

Листинг 3. Процедура «**МатериалыКоличествоПриИзменении()**»

&НаКлиенте

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

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

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

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

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

Листинг 4. Процедура «**МатериалыЦенаПриИзменении()**»

&НаКлиенте

```
Процедура МатериалыЦенаПриИзменении(Элемент)  
СтрокаТабличнойЧасти=Элементы.Материалы.ТекущиеДанные;  
РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);  
КонецПроцедуры
```

27. Запустите 1С:Предприятие в режиме отладки и убедиться, что теперь сумма в строках табличной части документов **ПриходнаяНакладная** рассчитывается как при изменении количества, так и при изменении цены.

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

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

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

- Склад, тип **СправочникСсылка.Склады**. Выбрать для свойства **Значение** заполнения **предопределенный элемент** **Основной справочника** **Склады**.
- Клиент, тип **СправочникСсылка.Клиенты**. Установить свойство **Проверка** заполнения в значение **Выдавать ошибку**.
- Мастер, тип **СправочникСсылка.Сотрудники**. Установить свойство **Проверка** заполнения в значение **Выдавать ошибку**.

31. Создайте табличную часть этого документа **ПереченьНоменклатуры** с реквизитами:

- Номенклатура, тип **СправочникСсылка.Номенклатура**;
- Количество, тип **Число**, длина 15, точность 3, неотрицательное;
- Цена, тип **Число**, длина 15, точность 2, неотрицательное,
- ■ Сумма, тип **Число**, длина 15, точность 2, неотрицательное;

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

32. На закладке Формы создать основную форму документа. Для поля ПереченьНоменклатурыКоличество создать обработчик события ПриИзменении, в котором будем вызывать процедуру РассчитатьСумму из общего модуля РаботаСДокументами. При этом откроется модуль формы с шаблоном обработчика события ПереченьНоменклатурыКоличествоПриИзменении, который пока заполнять не надо, а перейти в окно элементов формы на закладку Форма и аналогичным образом создать обработчик события ПереченьНоменклатурыЦенаПриИзменении для поля ПереченьНоменклатурыЦена.

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

Листинг 5. Модуль формы документа «ОказаниеУслуги»

&НаКлиенте

Процедура ПереченьНоменклатурыКоличествоПриИзменении(Элемент)

СтрокаТабличнойЧасти=Элементы.ПереченьНоменклатуры.ТекущиеДанные;

РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);

КонецПроцедуры

&НаКлиенте

Процедура ПереченьНоменклатурыЦенаПриИзменении(Элемент)

СтрокаТабличнойЧасти=Элементы.ПереченьНоменклатуры.ТекущиеДанные;

РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);

КонецПроцедуры

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

Контрольные вопросы

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

ПРАКТИЧЕСКАЯ РАБОТА № 4

Тема работы: Регистры накопления.

Цель работы: Научиться создавать различные регистры.

Общая теория

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

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

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

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

Отличительной особенностью регистра накопления является то, что он не предназначен для интерактивного редактирования пользователем. Разработчик может при необходимости предоставить пользователю возможность редактировать регистр накопления. Но предназначение регистра накопления заключается в том, чтобы его модификация производилась на основе алгоритмов работы других объектов базы данных, а не в результате непосредственных действий пользователя. Основным назначением регистра накопления является накопление числовой информации в разрезе нескольких *измерений*, которые описываются разработчиком в соответствующем объекте конфигурации Регистр накопления и являются подчиненными объектами конфигурации.

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

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

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

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

Ход выполнения работы

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

1. Открыть в конфигураторе нашу учебную конфигурацию и добавим Объект конфигурации Регистр накопления. Для этого выделим в дереве объектов конфигурации ветвь Регистры накопления и нажмем кнопку Добавить в командной панели окна конфигурации. В открывшемся окне редактирования объекта конфигурации на закладке **Основные** зададим имя регистра - **ОстаткиМатериалов**. Также зададим и **Расширенное представление списка** как **Движения по регистру Остатки материалов**. Этот заголовок будет отображаться в окне списка записей регистра.

2. **Закладка Подсистемы**. По логике нашей конфигурации данный регистр должен быть доступен в разделах Учет материалов, Оказание услуг и Бухгалтерия.

3. **Закладка Данные** - создадим структуру регистра. Создадим измерения регистра. Выделим ветвь **Измерения** и нажмем кнопку **Добавить** в командной панели окна.

4.

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

5. Создадим ресурс **Количество** с длиной 15 и точностью 3. Для этого выделить ветвь **Ресурсы** и нажать кнопку **Добавить** в командной панели окна.

6. В результате этих действий регистр **ОстаткиМатериалов** должен иметь следующий вид:

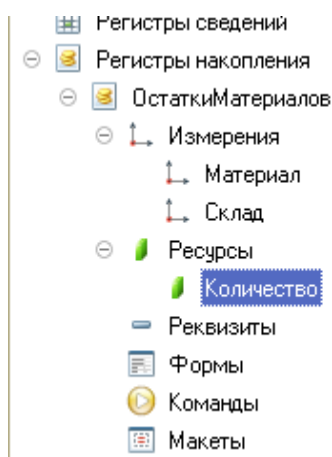


Рис.1. Регистр «ОстаткиМатериалов»

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

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

Движения документа

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

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

8. Сразу после отметки выбранного регистра становится доступной кнопка **Конструктор движений**. Нажмем ее и воспользуемся этим конструктором. В списке **Регистры** перечислены регистры, в которых документ может создавать движения. В нашем случае там пока один регистр **ОстаткиМатериалов**.

В списке **Реквизиты документа** должны находиться исходные данные для создания движений - реквизиты документа **ПриходнаяНакладная**.

В таблице **Поле - Выражение** должны быть заданы формулы, по которым будут вычисляться значения измерений и ресурсом регистра при записи движений. По умолчанию конструктор предлагает нам создавать движения прихода (**Тип движения регистра - Приход, символ + рядом с названием регистра**) по регистру **ОстаткиМатериалов**. Это нас вполне устраивает, ведь документ **ПриходнаяНакладная** и должен приходить материалы.

В поле выбора **Табличная часть** выберем табличную часть нашего документа - **Материалы**. Список реквизитов документа, который уже заполнен реквизитами шапки документа, автоматически дополнится реквизитами нашей табличной части.

9. Нажмем кнопку **Заполнить выражения**. В нижнем окне сформируется соответствие полей (измерений и ресурсов) регистра и выражений для их расчета. Конструктор движений установит соответствия подходящим образом: в качестве материала в регистр будет записан материал из табличной части документа, в качестве склада - склад, указанный в шапке документа, а в качестве количества - количество из табличной части документа.

10. Нажмем кнопку ОК и посмотрим, какой текст сформировал конструктор в модуле документа **Приходная Накладная**

Листинг 1.

Процедура **ОбработкаПроведения(Отказ, Режим)**

```
///_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

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

```
// регистр ОстаткиМатериалов Приход
```

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

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

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

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

```
Движение.Период = Дата;
```

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

```
Движение.Склад = Склад;
```

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

```
КонецЦикла;
```

```
///_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

```
КонецПроцедуры
```

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

Пояснение текста процедуры обработчика

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

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

Затем через точку можно использовать различные методы набора записей регистра, например, **Движения.ОстаткиМатериалов.Добавить()**.

*Метод **Добавить()** добавляет новую запись в набор записей.*

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

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

В цикле обращение к табличной части документа происходит по имени (**Материалы**). Переменная **ТекСтрокаМатериалы** содержит объект с данными текущей строки табличной части документа. Эта переменная создается в начале цикла и меняется по мере его прохождения.

В первой строке тела цикла, используя метод **Добавить()**, мы добавляем к набору записей, который создает наш документ в регистре, новую запись. Тем самым мы создаем объект **РегистрНакопленияЗапись** и сохраняем его в переменной **Движение**. Используя этот объект, мы можем обратиться к полям этой записи, указав имя поля через точку от этой переменной (например, **Движение.Количество**).

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

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

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

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

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

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

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

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

15. Аналогично, выделив подсистемы **ОказаниеУслуг** и **Бухгалтерия**, в панели навигации в группе **Обычное** включим видимость у команды **Остатки материалов** и перенесем ее в группу **См.также**.

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

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

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

19. Перепровести документы можно и не открывая документов. Для этого нужно выделить нужный документ в списке, нажать кнопку **Все действия** в командной панели формы списка и выбрать пункт **Провести**.

20. Выполнить команду **Остатки Материалов** и открыть список регистра накопления.

21. При проведении приходных накладных появляются соответствующие записи в регистре накопления **Остатки материалов**. Все поля регистра заполнились данными документов так, как мы задали в обработчике проведения документа **ПриходнаяНакладная**. Пиктограмма со знаком + слева от каждой записи указывает на тип движения - **Приход**.

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

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

24. В разделе **Панель навигации** раскроем группу **Перейти** и увидим команду для открытия списка регистра накопления **Остатки материалов**. Эта команда была автоматически помещена в панель навигации формы документа, так как он является регистратором, то есть создает движения в нашем регистре.

25. Установим свойство **Видимость** для этой команды.

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

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

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

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

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

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

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

32. Конструктор создал обработчик события **ОбработкаПроведения** объекта конфигурации **Документ ОказаниеУслуги** и поместил его в модуль объекта

Листинг 2.

Процедура ОбработкаПроведения(Отказ, Режим)

```
//{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

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

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

```
// регистр ОстаткиМатериалов Расход
```

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

Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл

Движение = Движения.ОстаткиМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Склад = Склад;

Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;

КонецЦикла;

//}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ

КонецПроцедуры

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

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

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

35. Выполните команду `Остатки материалов` и откройте список нашего регистра накопления.

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

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

Способы работы с коллекцией

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

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

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

Листинг 3. Перебор элементов коллекции в цикле

Для Каждого СтрокаТабличнойЧасти из ТабличнаяЧасть Цикл

Сообщить(СтрокаТабличнойЧасти.Услуга);

КонецЦикла

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

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

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

Листинг 4. Обращение к элементу коллекции

Справочники. Сотрудники;

Справочники[«Сотрудники»];

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

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

Листинг 5. Обращение к элементу коллекции по индексу

ТабличнаяЧасть[0]

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

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

Контрольные вопросы

1. Для чего предназначен объект конфигурации Регистр накопления?
2. Почему следует использовать регистры, хотя необходимая информация содержится в других объектах?
3. Для чего нужны измерения регистра, ресурсы и реквизиты?
4. Что такое движения регистра и что такое регистратор?
5. Как создать новый регистр накопления и описать его структуру?
6. Как создать движения документа с помощью конструктора движений.
7. Как средствами встроенного языка обойти табличную часть документа и обратиться к ее данным?
8. Как показать команды открытия списка регистра в интерфейсе конфигурации и в интерфейсе формы?

ПРАКТИЧЕСКАЯ РАБОТА № 5

Тема работы: Простой отчет.

Цель работы: Научиться создавать различные отчеты.

Ход выполнения работы.

Теоретическая часть

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

Добавление отчета

В режиме Конфигуратор

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

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

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

Макет

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

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

4. Добавим новый набор данных - запрос. Для этого нажмем кнопку Добавить или вызовем контекстное меню ветки Наборы данных. Для того чтобы создать текст запроса, запустим конструктор запроса - нажмем кнопку Конструктор запроса.

Конструктор запроса - инструмент, созданный для помощи разработчику, позволяющий визуально конструировать запрос, пользователь, не знакомый с языком запросов, может с помощью конструктора создать синтаксически правильный запрос.

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

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

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

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

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

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

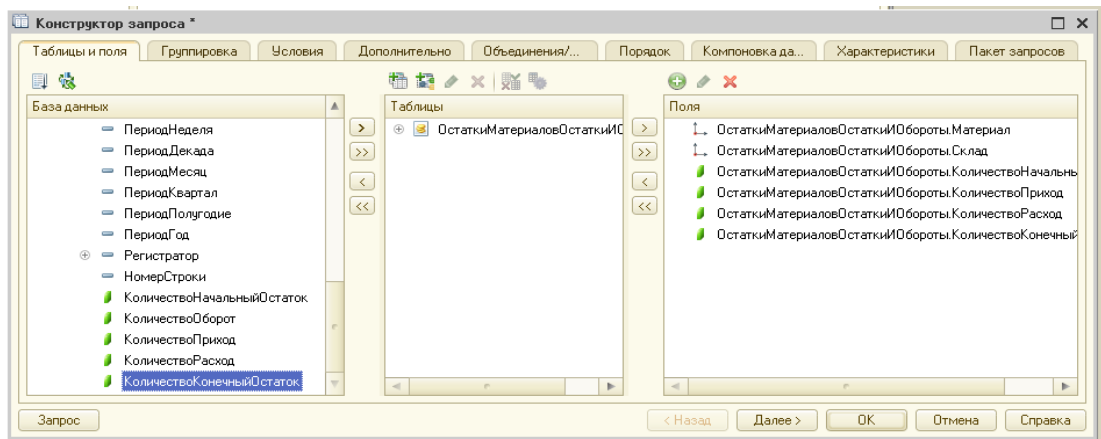


Рис. 1. Выбранные поля

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

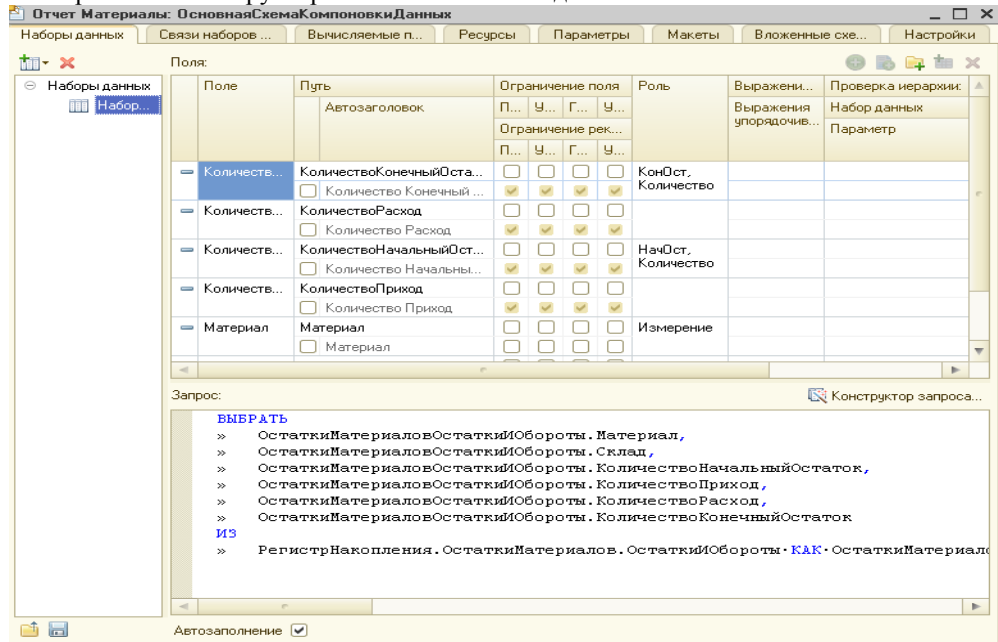


Рис. 2. Компоновка данных в отчете

11. Текст запроса, который был создан с помощью конструктора платформа поместит в поле Запрос. Это поле представляет собой текстовый редактор, в котором можно вручную отредактировать существующий запрос. Кроме того, можно снова вызвать конструктор запроса и отредактировать запрос при помощи него. Обратите внимание на список полей системы компоновки данных который платформа заполнила в верхней части конструктора. В нем отображаются поля, которые доступны у текущего набора данных. В нашем случае система 1С:Предприятие заполнила данный список автоматически, из текста запроса, и нет необходимости в его ручной настройке. Итак, мы описали, каким образом будут извлекаться данные для отчета. Но пока мы не создадим стандартных настроек нашего отчета, мы ничего не увидим в результате. Поэтому создадим самые простые настройки отчета для отображения обычных детальных записей информационной базы. В нашем случае это будут записи виртуальной Таблицы регистр накопления ОстаткиМатериалов, выбранные в линейном порядке по мере попадания их в эту таблицу.

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

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

В структуре отчета появится группировка Детальные записи (рис. 3)

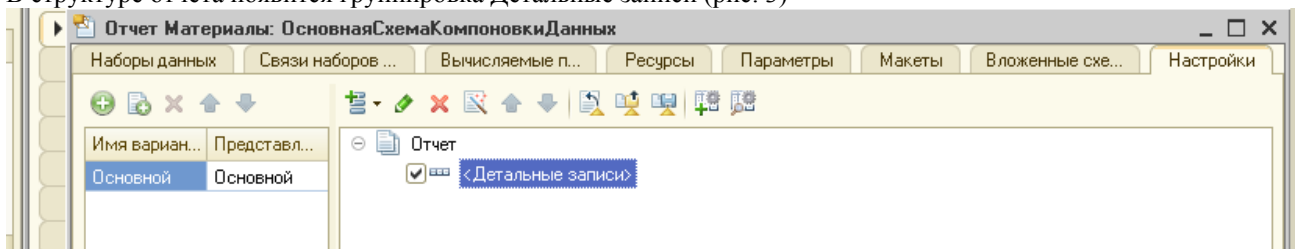


Рис. 3. Группировка данных в отчете

14. Теперь настроим поля, которые будут выводиться в результат отчета. Для этого перейдем в нижнем окне

настроек на закладку Выбранные поля и перенесем мышью из списка доступных полей:

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

ПРИМЕЧАНИЕ

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

15. В результате окно настроек отчета должно иметь вид как на Рис.4

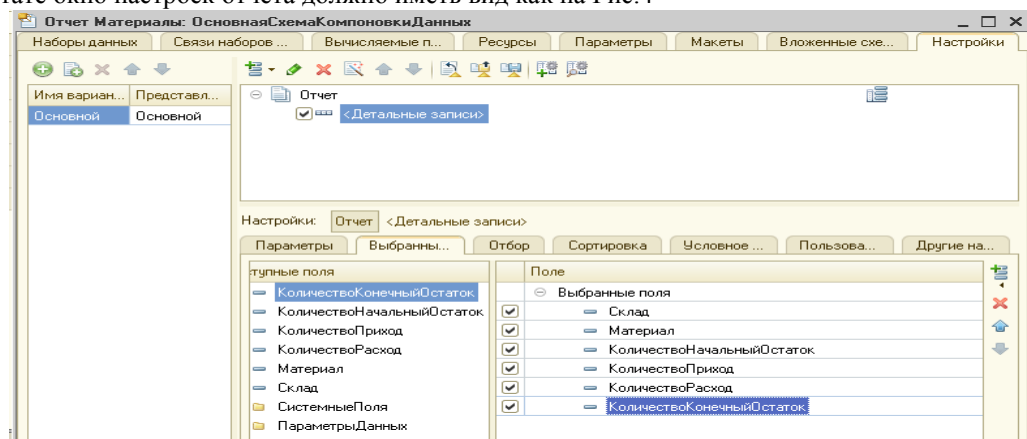


Рис. 4. Компонновка данных отчета

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

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

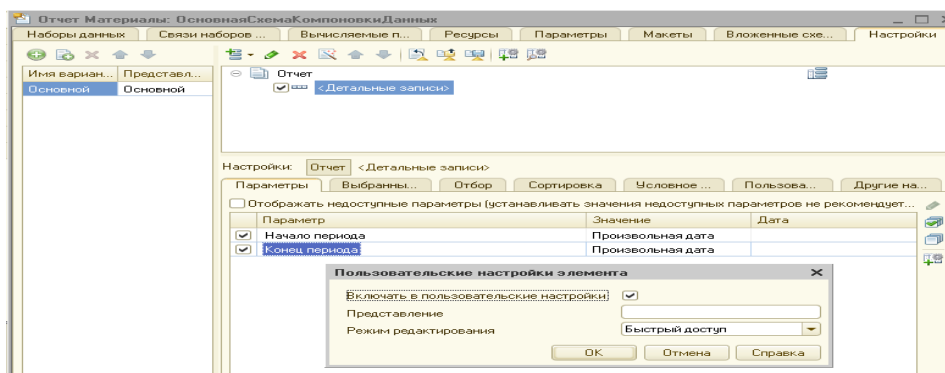


Рис. 5. Пользовательские настройки отчета

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

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

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

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

Контрольные вопросы

1. Для чего предназначен объект конфигурации Отчет.
2. Как создать отчет с помощью конструктора схемы компоновки данных.

3. Как отобразить отчет в разделах прикладного решения.

ПРАКТИЧЕСКАЯ РАБОТА № 6

Тема работы: Макет печатной формы.

Цель работы: Научиться создавать различные макеты печатной формы.

Теоретическая часть

Объект конфигурации *Макет* предназначен для хранения различных форм представления данных, различных данных, которые могут потребоваться каким-либо объектам конфигурации или всему прикладному решению в целом.

Макет может содержать табличный или текстовый документ, двоичные данные, HTML-документ или Active Document, графическую или географическую схему, схему компоновки данных или макет оформления схемы компоновки данных.

Макеты могут существовать как сами по себе (общие макеты), так и быть подчинены какому-либо объекту конфигурации.

Одно из предназначений макета, подчиненного объекту конфигурации и содержащего табличный документ, - создание печатной формы этого объекта. Создание печатной формы заключается в конструировании ее составных частей - именованных областей, из которых затем «собирается» готовая печатная форма.

Порядок заполнения областей данными и вывода их в итоговую форму описывается при помощи встроенного языка. Печатная форма может включать в себя различные графические объекты: картинки. OLE-объекты, диаграммы и т. д.

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

Ход выполнения работы

Макет печатной формы

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

1. Откроем в конфигураторе окно редактирования объекта конфигурации Документ **ОказаниеУслуги**.
2. Перейдем на закладку **Макеты** и запустим конструктор печати. В открывшемся окне на первом шаге укажем, что будет создана новая команда Печать для формирования печатной формы документа.
3. На втором шаге нажатием кнопки >> определим, что все реквизиты нашего документа будут отображены в шапке печатной формы. Нажмем **Далее**.
4. На третьем шаге точно так же определим, что все реквизиты табличной части документа будут отображены в печатной форме.
5. На четвертом шаге сформируем подвал (нижнюю часть) печатной формы. Мы не станем ничего указывать (подвал в данном случае использовать не будем), нажмем и перейдем к пятому шагу. Здесь ничего изменять не будем. Тем самым согласимся с тем, команда для вызова процедуры формирования печатной формы будет помещена в командную панель формы, в раздел Важное. Нажмем **ОК**.
6. В конфигураторе откроется модуль команды Печать, модуль менеджера документа **ОказаниеУслуги** и макет этого документа. Разработчик может создать макет печатной формы с нуля и для ее вывода создать соответствующую команду и кнопку в форме документа, но в данном случае всю работу сделал за нас конструктор печати:
 - Создан макет печатной формы документа **ОказаниеУслуги** с именем **Печать**.
 - Создана команда документа **ОказаниеУслуги** с именем **Печать**. В модуль этой команды помещен обработчик, вызывающий процедуру печати документа, выполняющуюся на сервере. Сама процедура печати помещена в модуль менеджера документа **ОказаниеУслуги**.
 - В командную панель формы документа **ОказаниеУслуги** помещена команда **Печать** для формирования печатной формы документа.
7. Поскольку команда **Печать** принадлежит документу **ОказаниеУслуги** в целом, а не конкретной его форме, эту команду можно будет помещать в любую форму, созданную для документа.
8. В будущем мы будем самостоятельно создавать процедуры обработчиков команд, и размещать соответствующие им кнопки в форме. Пока воспользуемся результатами работы конструктора печати и проверим макет в работе.
9. Запустим 1С: Предприятие в режиме отладки и откроем документ **ОказаниеУслуги № 1**. Обратите внимание, что в командной панели документа появилась кнопка **Печать**.
10. Нажмем на нее и увидим печатную форму нашего документа. Как видите, конструктор сформировал вполне подходящую печатную форму для нашего документа. Единственное, чего не хватает в данной форме, - это итоговой суммы документа.

Редактирование макета

11. В режиме Конфигуратор прежде всего добавим итоговую сумму в печатную форму документа **ОказаниеУслуги**. Откроем конфигуратор, раскроем дерево документа **Оказание Услуги** и дважды щелкнем на макете **Печать**.

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

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

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

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

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

17. Каждая ячейка редактируемого нами табличного документа может содержать либо текст, либо некоторый параметр, либо шаблон. *Текст*, содержащийся в ячейке, будет показан на экране. *Параметр* будет заменен некоторым значением, которое может быть присвоено ему средствами встроенного языка. Текст, содержащийся в ячейке, является именем этого параметра. *Шаблон* представляет собой текстовую строку, в определенные места которой будут вставлены значения параметров. Поэтому, указав для ячейки в качестве заполнения **Параметр**, мы определили параметр области с именем **ВсегоПоДокументу**, которому присвоим нужное нам значение при формировании печатной формы.

18. Откроем модуль менеджера документа **ОказаниеУслуги**. Для этого перейдем на закладку **Прочее** окна редактирования объекта конфигурации Документ **ОказаниеУслуги** и нажмем кнопку **Модуль менеджера**. Найдем в нем процедуру **Печать** и отредактируем ее следующим образом (новые строки выделены жирным шрифтом), листинг- 1.

Листинг – 1. Печать формы документа (фрагмент)

```
ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");
```

```
Шапка = Макет.ПолучитьОбласть("Шапка");
```

```
ОбластьПереченьНоменклатурыШапка = Макет.ПолучитьОбласть("ПереченьНоменклатурыШапка");
```

```
ОбластьПереченьНоменклатуры = Макет.ПолучитьОбласть("ПереченьНоменклатуры");
```

```
ОбластьИтог=Макет.ПолучитьОбласть("Всего");
```

```
ТабДок.Очистить();
```

```
ВставлятьРазделительСтраниц = Ложь;
```

```
Пока Выборка.Следующий() Цикл
```

```
Если ВставлятьРазделительСтраниц Тогда
```

```
ТабДок.ВывестиГоризонтальныйРазделительСтраниц();
```

```
КонецЕсли;
```

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

```
Шапка.Параметры.Заполнить(Выборка);
```

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

```
ТабДок.Вывести(ОбластьПереченьНоменклатурыШапка);
```

```
ВыборкаПереченьНоменклатуры = Выборка.ПереченьНоменклатуры.Выбрать();
```

```
СуммаИтог=0;
```

```
Пока ВыборкаПереченьНоменклатуры.Следующий() Цикл
```

```
ОбластьПереченьНоменклатуры.Параметры.Заполнить(ВыборкаПереченьНоменклатуры);
```

```
ТабДок.Вывести(ОбластьПереченьНоменклатуры, ВыборкаПереченьНоменклатуры.Уровень());
```

```
СуммаИтог=СуммаИтог+ВыборкаПереченьНоменклатуры.Сумма;
```

```
КонецЦикла;
```

```
ОбластьИтог.Параметры.ВсегоПоДокументу=СуммаИтог;
```

```
ТабДок.Вывести(ОбластьИтог);
```

```
ВставлятьРазделительСтраниц = Истина;
```

```
КонецЦикла;
```

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

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

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

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

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

24. Запустим 1С:Предприятие в режиме отладки и проверим результат наших изменений. Подобным образом, создавая именованные области и ячейки макета, используя их свойства и управляя порядком их вывода с помощью встроенного языка, разработчик имеет возможность создать печатную форму любого дизайна. А теперь, для того чтобы наш документ **ОказаниеУслуги**, выглядел вполне законченным, добавим итоговую сумму по документу и на экранную форму, чтобы пользователь мог видеть ее в процессе заполнения табличной части документа.

Редактирование формы

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

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

27. Дважды щелкнем на элементе **ПереченьНоменклатуры** в дереве, элементов формы или правой кнопкой

мышь откроем для него палитру свойств (пункт контекстного меню **Свойства**).

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

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

- **Текст подвала - Всего:**
- **Горизонтальное положение в подвале – Право**
- свойство **Шрифт подвала** изменим начертание на **Жирный**.

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

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

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

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

Контрольные вопросы

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

ПРАКТИЧЕСКАЯ РАБОТА № 7

Тема работы: Периодические регистры сведений

Цель работы: Научиться создавать различные периодические регистры сведений.

Ход выполнения работы.

Теоретический материал.

Зачем нужен периодический регистр сведений

В ООО «На все руки мастер» существует перечень услуг, который определяет стоимость каждой услуги. Стоимость услуги является неотъемлемым свойством самой услуги, и поэтому ее следует добавить в качестве реквизита справочника **Номенклатура**. Однако стоимость услуг имеет особенность меняться со временем. И может сложиться такая ситуация, когда нам потребуется внести изменения или уточнения в один из ранее проведенных документов **Оказание услуги**. В этом случае мы не сможем получить правильную стоимость услуги, поскольку в реквизите справочника будет храниться последнее введенное значение.

Кроме этого, не исключено, что руководство пожелает видеть зависимость прибыли предприятия от изменения стоимости оказываемых услуг. И тогда просто необходимо будет иметь возможность анализировать изменение стоимости услуг во времени.

Поэтому для хранения стоимости услуг мы используем новый пока еще для нас объект - Регистр сведений.

Что такое регистр сведений.

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

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

Следующей важной особенностью регистра сведений является его способность(при необходимости) хранить данные с привязкой ко времени.

Благодаря этому регистр сведений может хранить не актуальные значения данных, но и историю их изменения во времени. Регистр сведений, использующий привязку ко времени, называют **периодическим регистром сведений**.

Периодичность регистра сведений можно определить одним из следующих значений:

- в пределах секунды,
- в пределах дня,
- в пределах месяца,
- в пределах квартала,
- в пределах года,
- в пределах регистратора (если установлен режим записи **Подчинение регистратору**).

Периодический регистр сведений всегда содержит служебное поле **Период**, добавляемое системой автоматически. Оно имеет тип **Дата** и служит для указания факта принадлежности записи к какому-либо периоду. При записи данных в регистр платформа всегда приводит значение этого поля к началу того периода, в который он попадает.

Например, если в регистр сведений с периодичностью в пределах месяца записать данные, в которых период указан как 08.04.2004, то регистр сохранит эти данные со значением периода, равным 01.04.2004.

Как и для других регистров, система контролирует уникальность записей для регистра сведений. Однако если для прочих регистров уникальным идентификатором записи является регистратор и номер строки, то для регистра сведений применяется другой принцип формирования ключевого значения.

Ключом записи, однозначно идентифицирующим запись, является в данном случае совокупность значений измерений регистра и периода (в случае если регистр сведений периодический). Регистр сведений не может содержать несколько записей с одинаковыми ключами. Если продолжать сравнение с регистром накопления, то можно сказать, что регистр сведений предоставляет больше свободы в редактировании хранимых данных. Наряду с возможностью использования в режиме подчинения регистратору (когда записи регистра сведений «привязаны» к документу- регистратору) регистр сведений может применяться и в независимом режиме, в котором пользователю предоставляется полная свобода интерактивной работы с данными регистра. Регистр сведений, не использующий подчинение регистратору, называют **независимым регистром сведений**.

Добавление периодического регистра сведений.

Создадим периодический регистр сведений, который будет хранить развернутые во времени розничные цены материалов и стоимости услуг, оказываемых нашим ООО «На все руки мастер».

1. Открыть в режиме конфигуратор учебную конфигурацию и добавить объект конфигурации Регистр сведений. Для этого выделим в дереве объектов конфигурации ветвь **Регистры сведений** и нажмем кнопку **Добавить** в командной панели окна конфигурации.

2. В открывшемся окне редактирования объекта конфигурации на закладке Основные зададим имя регистра - **Цены**. Установим свойство **Периодичность** этого регистра - В пределах секунды. Такую периодичность мы выбрали для того, чтобы иметь возможность отслеживать цены несколько раз в течение дня. Если же так часто не предполагается изменять цены, то можно выбрать, вообще, в пределах дня. Здесь же определим представление объекта в интерфейсе приложения.

3. Задать свойства **Представление записи** как **Цена**, а **Представление списка** как **Цены на номенклатуру**. Обратите внимание на свойство **Режим записи**. По умолчанию оно значение - **Независимый**, то есть мы создаем независимый сведений и сможем в дальнейшем вводить в него данные без использования регистратора, «вручную».

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

5. Нажмем на закладку **Данные** и создадим измерение **Номенклатура с типом СправочникСсылка.Номенклатура**. Для этого выделим ветвь **Измерения** и нажмем кнопку **Добавить** в командной панели и указать, что это измерение будет ведущим. Свойство ведущее имеет смысл использовать лишь тогда, когда измерение имеет тип ссылки на объект базы данных. Установка свойства **Ведущее** будет говорить о том, что запись регистра сведений представляет интерес, пока существует тот объект, ссылка на который в качестве значения этого измерения в этой записи. При удалении объекта, все записи регистра сведений по этому объекту тоже будут автоматически удалены. Также в результате того, что это измерение регистра мы сделали ведущим в форме элемента справочника **Номенклатура**, в панели навигации в группе **Перейти**, появится ссылка. По ней возможен переход к записям этого регистра, которые содержат в измерении Номенклатура ссылку на этот элемент справочника.

6. Затем создадим ресурс **Цена**, тип **Число**, длина **15**, точность **2**, неотрицательное. Для этого выделим ветвь **Ресурсы** и нажмем кнопку **Добавить** в командной панели окна.

7. Запустить **1С:Предприятие** в режиме отладки и посмотреть, как работает наш периодический регистр сведений **Цены**. В открывшемся окне **1С:Предприятия** мы видим, что в панели навигации разделов **Бухгалтерия, Оказание услуг** и **Учет материалов** появилась команда для открытия списка регистра **Цены на номенклатуру**. Команда для открытия регистра сведений по умолчанию доступна в интерфейсе разделов, в которых отображается регистр, так как в отличие от регистров накопления предполагается изменение данных регистра пользователем. Чтобы добавить новую запись в регистр сведений, нажмем кнопку **Создать**.

8. Задать стоимость услуг самостоятельно. При этом период задать задним числом, так как он должен быть меньше или равен дате создания документа Оказание услуг (в нашем случае это 10.07.2009).

9. Задать розничные цены на материалы самостоятельно. Итак, имеем очень полезную возможность в нашей программе - установка цен на услуги и материалы. Поскольку цены хранятся с привязкой к дате, мы можем заранее установить новые цены и быть уверены в том, что новые цены вступят в действие не раньше указанного для них времени.

10. **Автоматическая подстановка цены в документ при выборе номенклатуры**. Цена номенклатуры у нас теперь хранится в отдельном регистре сведений. Когда мы создаем или изменяем документ **ОказаниеУслуги** и добавляем в табличную часть какую-либо номенклатуру, нам хочется, чтобы одновременно с этим в документ подставлялась бы сразу и актуальная цена этой номенклатуры, полученная из регистра сведений и соответствующая дате документа. Для этого нам нужно сделать:

Сначала написать некую функцию, которая будет возвращать нам актуальную цену номенклатуры, а затем вызвать эту функцию в тот момент, когда в документ добавляется номенклатура, и подставить в документ цену номенклатуры, которую вернет эта функция.

Поскольку такой «сервис» понадобится нам, скорее всего, не только в этом документе, но и в других документах, которые содержат в табличной части номенклатуру, мы разместим функцию в некотором «общедоступном» месте - в общем модуле.

11. В режиме Конфигуратор создать функцию **РозничнаяЦена()**, которая будет возвращать нам актуальную розничную цену номенклатуры, и поместим ее в общий модуль конфигурации. Открыть конфигуратор, в ветке **Общие ► Общие модули** добавить объект конфигурации Модуль и назвать его **РаботаСоСправочниками**. У модуля по умолчанию установлен флажок **Сервер**. Это означает, что экземпляры этого модуля будут скомпилированы только на стороне сервера.

12. Установим флажок **Вызов сервера** для того, чтобы экспортные процедуры и функции этого модуля можно было вызывать с клиента.

13. Поместим в него текст(листинг 1.)

Листинг 1. Функция «Розничная цена()»

```
Функция РозничнаяЦена(АктуальнаяДата,ЭлементНоменклатуры)Экспорт
```

```
//Создать вспомогательный объект Отбор
```

```
Отбор=Новый Структура("Номенклатура",ЭлементНоменклатуры);
```

```
// Получить актуальные значения ресурсов регистра.
```

```
ЗначениеРесурсов= РегистрыСведений.Цены.ПолучитьПоследнее(АктуальнаяДата.Отбор);
```

```
Возврат ЗначенияРесурсов.Цена;
```

```
КонецФункции
```

Пояснение листинга: Для получения розничной цены мы будем передавать в функцию два параметра:

- **АктуальнаяДата** - параметр типа Дата, определяет точку на оси времени, в которой нас интересует значение розничной цены.

- **ЭлементНоменклатуры** - ссылка на элемент справочника Номенклатура, для которого мы хотим получить розничную цену. В теле функции мы сначала создаем вспомогательный объект Отбор. Это структура, содержащая отбор по измерениям регистра. С его помощью определяем, что нас будут интересовать записи регистра в которых измерение регистра Номенклатура равно переданной к функции ссылке на элемент справочника.

Имя ключа структуры ("Номенклатура") должно совпадать с именем измерения регистра, заданного в конфигураторе, а значение элемент структуры (ЭлементНоменклатуры) задает отбираемое по данному измерению значение.

Во второй строке мы обращаемся к менеджеру регистра сведений Цены (РегистрыСведений.Цены) и выполняем метод ПолучитьПоследнее(), который возвращает нам значения ресурсов самой поздней записи регистра, соответствующей передаваемой в функцию дате (АктуальнаяДата) и значениям измерений регистра (Отбор).

Метод ПолучитьПоследнее возвращает структуру, содержащую значения ресурсов, которая сохраняется в переменной ЗначенияРесурсов. Вообще говоря, у регистра может быть несколько ресурсов. В нашем регистре ресурс один, но все равно будет возвращена структура, содержащая единственный элемент. Поэтому в следующей строке мы получаем искомую нами розничную цену, просто указав имя нужного нам ресурса регистра через точку (ЗначенияРесурсов.Цена) и возвращаем ее при выполнении функции. Теперь эту функцию нужно вызвать в некоторый момент работы документа.

14. **Вызов функции при выборе номенклатуры и заполнение цены в документе.** При редактировании документа ОказаниеУслуги нам необходимо обеспечить автоматическое заполнение поля Цена после того, как пользователь выберет услугу. Причем цена услуги должна определяться исходя из даты создаваемого документа. Найдем в конфигураторе документ ОказаниеУслуги и откроем его форму Форма Документа.

15. Дважды щелкнуть на элементе формы ПереченьНоменклатурыНоменклатура или правой кнопкой мыши откроем для него палитру свойств (пункт контекстного меню Свойства). Прокрутив список до конца, найдем событие ПриИзменении, которое возникает после изменения значения поля.

16. Нажмем кнопку открытия со значком лупы в поле ввода. Система создаст шаблон процедуры обработчика этого события в модуле нашей формы и откроет закладку Модуль редактора формы. Внести в него следующий текст (листинг 2).

Листинг 2. Процедура «ПереченьНоменклатурыНоменклатураПриИзменении()»

```
Процедура ПереченьНоменклатурыНоменклатураПриИзменении(Элемент)
```

```
СтрокаТабличнойЧасти=Элементы.ПереченьНоменклатуры.ТекущиеДанные;
```

```
СтрокаТабличнойЧасти.Цена=РаботаСоСправочниками.РозничнаяЦена(Объект.Дата,СтрокаТабличнойЧасти.Номенклатура);
```

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

```
КонецПроцедуры
```

Пояснение к листингу:

Первая и вторая строки обработчика нам уже знакомы по процедурам ПереченьНоменклатурыКоличествоПриИзменении и ПереченьНоменклатураЦенаПриИзменении. Сначала мы получаем текущую строку табличной части документа, так как она нам понадобится в дальнейшем, и сохраняем ее в переменной СтрокаТабличнойЧасти. Затем мы вызываем нашу функцию РозничнаяЦена() из общего модуля РаботаСоСправочниками.

Первым параметром мы передаем в эту функцию дату документа, на которую необходимо получить цену. Дату документа мы получаем из основного реквизита формы - Объект.Дата.

Вторым параметром мы передаем ссылку на элемент справочника Номенклатура, который содержится в текущей строке табличной части документа (СтрокаТабличнойЧасти.Номенклатура).

Функция возвращает последнее значение цены, и это значение мы присваиваем полю **Цена** в текущей строке табличной части документа (СтрокаТабличнойЧасти.Цена).

Затем мы вызываем процедуру **РассчитатьСумму** из общего модуля **РаботаСДокументами**. Эту процедуру создали для того, чтобы при изменении цены или количества в документе пересчитывать сумму в строке документа.

Заметьте, что сама процедура **ПереченьНоменклатурыНоменклатураПриИзменении()** начинает работать в модуле формы на стороне клиента, так как это обработчик интерактивного события формы. Создавая заготовку этой процедуры, платформа автоматически разместила перед описанием процедуры директиву компиляции **&НаКлиенте**.

Затем вызывается функция **РозничнаяЦена()**. Поскольку эта функция не будет найдена на стороне клиента, то исполнение будет передано в общий модуль **РаботаСоСправочниками**, который выполняется на сервере. После завершения функции программный код продолжит исполняться на клиенте.

Это делается так, потому что любая работа с базой данных (чтение данных, запись) возможна только на сервере. В данном случае нам необходимо было прочитать последние данные из регистра сведений для некоторой номенклатуры.

Такие действия можно выполнить только на сервере, и если посмотреть в синтаксис-помощнике описание метода **ПолучитьПоследнее()** регистра сведений, то можно заметить, что этот метод доступен только на сервере, в толстом клиенте и во внешнем соединении.

Толстый клиент и внешнее соединение - это клиентские приложения прежней версии платформы, которые существуют для совместимости с прежними прикладными решениями.

Мы разрабатываем совершенно новое прикладное решение, которое работает в тонком клиенте или в веб-клиенте. Поэтому в нашем случае для получения каких-либо данных из базы данных нужно передать исполнение кода на сервер, там получить нужные данные и вернуть эти данные на клиента. Что мы и сделали.

17. Проверим, как теперь работает наш документ. Запустим 1С:Предприятие в режиме отладки и откроем регистр сведений **Цены**. Для транзистора Philips добавим другим числом новую цену.

18. Откроем документ **Оказание услуги № 1**. Оставим дату документа без изменения и повторим выбор транзистора в колонке **Номенклатура** табличной части документа. Автоматически установится значение цены транзистора от 01.07.2009. последнее значение цены на дату документа.

19. Теперь изменим дату документа на 13.07.2009 и снова повторим выбор транзистора. Будет установлено новое значение цены последнее на эту дату. Таким образом, в документе появляется актуальная на момент создания документа цена услуги.

Контрольные вопросы

1. Для чего предназначен объект конфигурации Регистр сведений.
2. Какими особенностями обладает объект конфигурации Регистр сведений.
3. В чем главные отличия регистра сведений от регистра накопления.
4. Какие поля определяют ключ уникальности регистра накопления.
5. Что такое периодический регистр сведений и что такое независимый регистр сведений.
6. Как создать периодический регистр сведений.
7. Что такое ведущее измерение регистра.
8. Как получить значения ресурсов наиболее поздних записей регистра средствами встроенного языка.

ПРАКТИЧЕСКАЯ РАБОТА № 8

Тема работы: Перечисления.

Цель работы: Научиться создавать различные перечисления.

Общая теория

Объект конфигурации **Перечисление** предназначен для описания структуры хранения постоянных наборов значений, не изменяемых в процессе работы конфигурации. На основе объекта конфигурации Перечисление платформа создает в базе данных таблицу, в которой может храниться набор некоторых постоянных значений. В реальной жизни этому объекту может соответствовать, например, перечисление вариантов указания цены («включая НДС», «Без НДС»). Набор всех возможных значений, которые содержит перечисление, задается при конфигурировании системы, и пользователь не может изменять их, удалять или добавлять новые.

Из этого следует важная особенность перечисления: значения перечисления не «обезличены» для конфигурации, на них могут опираться алгоритмы работы программы.

Ход выполнения работы

1. Открыть конфигуратор и создать сначала новый объект конфигурации **Перечисление** с именем **ВидыНоменклатуры**.

2. На закладке **Данные** добавить два значения перечисления: **Материал** и **Услуга**. Для этого нажмем кнопку **Добавить** над списком значений перечисления.

3. **Привязка номенклатуры к значениям перечисления «ВидНоменклатуры»**. Чтобы привязать номенклатуру к значениям перечисления, надо сделать следующее:

■ В режиме Конфигуратор создать у справочника **Номенклатура** реквизит, который будет хранить значение перечисления.

■ В режиме 1С:Предприятие проставить нужные значения этого реквизита для всех элементов справочника **Номенклатура**.

В режиме Конфигуратор добавим в справочник **Номенклатура** новый реквизит **ВидНоменклатуры** с типом **ПеречислениеСсылка.ВидыНоменклатуры**. Для этого откроем окно редактирования объекта конфигурации Справочник **Номенклатура** и на закладке **Данные** нажмем кнопку **Добавить** над списком реквизитов справочника.

4. Запустить 1С:Предприятие в режиме отладки. Платформа выдаст предупреждающее сообщение о том, что наше перечисление не включено ни в одну подсистему. Проигнорируем его и примем изменения конфигурации.

5. В режиме 1С:Предприятие зададим для каждого элемента справочника **Номенклатура** соответствующее значение реквизита **Вид Номенклатуры**(рис. 1).

Номенклатура		
Наименование	Код	Виды номенклатуры
Номенклатура		
Номенклатура	000000007	
Материалы	000000001	
Прочее	000000008	
Кабель электрический	000000004	Материал
шланг резиновый	000000009	Материал
Радиодетали	000000013	
Строчный трансформатор GoldStar	000000012	Материал
Строчный трансформатор Samsung	000000011	Материал
Транзистор Philips	000000003	Материал
Услуги	000000002	
Стиральные машины	000000014	
подключение воды	000000015	Услуга
подключение электричества	000000006	Услуга
Телевизоры	000000016	
диагностика	000000017	Услуга
ремонт импортного телевизора	000000018	Услуга
ремонт отечественного телевизора	000000019	Услуга

Рис. 1. Данные справочника Номенклатуры

б. **Регистрация расхода только той номенклатуры, которая является материалом.** Если вы помните, на шестом занятии, когда создавались движения документа **ОказаниеУслуги** по регистру накопления **ОстаткиМатериалов**, мы сказали, что они не совсем правильные, поскольку в регистр будут попадать не только записи об израсходованных материалах, но и записи об оказанных услугах. Теперь мы доработаем документ таким образом, чтобы в регистре появлялись только записи, относящиеся к расходу материалов. Для этого мы сначала в режиме Конфигуратор изменим процедуру проведения документа так, чтобы в регистр попадали записи только о той номенклатуре, которая является материалом, а потом в режима 1С Предприятие заново проведем (перепроведем) все документы **Оказание услуги**, чтобы данные в регистре изменились в соответствии с новым алгоритмом проведения документа. Эта доработка будет не совсем эффективна с точки зрения производственной деятельности, зато позволит нам получить нужные данные в регистре **ОстаткиМатериалов**.

Скорректируем движения документа, исключив из обработки те строки табличной части, в которых находятся услуги. Для этого откроем в конфигураторе модуль документа **ОказаниеУслуги** (контекстное меню документа - **Открыть модуль объекта**) и добавим в обработчик события **ОбработкаПроведения** это условие. Текст следует добавить в начало цикла обхода табличной части документа после строки **Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл**. В результате процедура **ОбработкаПроведения** должна выглядеть следующим образом:

Листинг Движения документа «ОказаниеУслуги»

```

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

```

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

В переменной **ТекСтрокаПереченьНоменклатуры** содержатся на каждом шаге цикла данные текущей строки табличной части **ПереченьНоменклатуры**.

Указывая через точку имя колонки Номенклатура (**ТекСтрокаПереченьНоменклатуры.Номенклатура**), мы обращаемся к ссылке на элемент номенклатуры, которая содержится в этой строке табличной части.

Затем, указывая через точку **ВидыНоменклатуры** (**ТекСтрокаПереченьНоменклатуры.ВидыНоменклатуры**), мы обращаемся к реквизиту **ВидыНоменклатуры** этого элемента справочника **Номенклатура**.

Полученное значение с помощью оператора сравнения (=) мы сравниваем со значением **Материал** перечисления **ВидыНоменклатуры** (**Перечисления.ВидыНоменклатуры.Материал**). Если значения совпадают, операторы цикла выполняются. Если нет, мы переходим к следующей итерации цикла, к следующей строке табличной части.

7. Запустим 1С: Предприятие в режиме отладки и проверим работу процедуры проведения документа **Оказание услуги**. Откроем список документов, выполнив команду **Оказание услуг** в панели навигации раздела **Оказание услуг**.

Откроем документ **Оказание услуги № 1** и внесем в него следующие изменения:

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

Обратите внимание, что цены подставляются автоматически из регистра сведений **Цены**.

Нажмем кнопку **Провести** в командной панели формы документа.

Затем выполним команду **Остатки материалов** в панели навигации формы, чтобы перейти к записям регистра **Остатки материалов** связанным с данным документом. Видим, что в движения по регистру **Остатки материалов** включаются только строки, содержащие материалы. Запись по услуге **Подключение воды** в движении не попала.

Контрольные вопросы

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

ПРАКТИЧЕСКАЯ РАБОТА № 9

Тема работы: Проведение документа «Приходная накладная» по нескольким регистрам.

Цель работы: Научиться проводить различные документы по нескольким регистрам.

Ход выполнения работы.

Зачем нужно проведение документа по нескольким регистрам

Для количественного учета движения материалов был создан регистр накопления **ОстаткиМатериалов**, только количественного учета совершенно недостаточно для нужд предприятия. Необходимо также знать, какие денежные средства были затрачены на приобретение тех или иных материалов и каковы материальные запасы в денежном выражении.

Предположим, что требуется чтобы весь суммовый учет материалов велся бы теперь по средней стоимости. То есть при закупке материалов они должны учитываться в ценах приобретения, а при расходе - по средней стоимости, которая рассчитывается исходя из общей суммы закупок данного материала и общего количества этого материала, находящегося в организации. Поскольку подобная информация имеет совершенно другую структуру, нежели количественный учет, для хранения данных об общей стоимости тех или иных материалов будем использовать еще один регистр накопления **СтоимостьМатериалов**.

Таким образом, документы **ПриходнаяНакладная** и **ОказаниеУслуг** должны будут создавать движения не только в регистре **ОстаткиМатериалов**, но одновременно и в регистре **СтоимостьМатериалов**, отражая изменения суммового учета.

1. Создадим новый объект конфигурации **Регистр накопления** с именем **СтоимостьМатериалов**. **Расширенное представление списка** зададим как **Движения по регистру Стоимость материалов**.

2. В закладке **Подсистемы** отметим, что этот регистр будет отображаться в подсистемах **Бухгалтерия**, **Учет материалов** и **ОказаниеУслуг**.

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

4. В дереве объектов конфигурации выделим ветвь **Подсистемы**, в ее контекстном меню и выберем пункт **Все подсистемы**. В открывшемся окне слева в списке **Подсистемы** выделим подсистему **Бухгалтерия**. Справа в списке **Командный интерфейс** отразятся все команды выбранной подсистемы. В группе **Панель навигации.Обычное** включим видимость у команд **Стоимость материалов** и мышью перетащим ее в группу **Панель навигации.См.также**.

5. Аналогично, выделив подсистемы **ОказаниеУслуг** и **УчетМатериалов**, в панели навигации в группе **Обычное** включим видимость у команд **Стоимость материалов** и перенесем ее в группу **См.также**.

Теперь можем приступить к внесению изменений в процедуру проведения наших документов. Начнем с самого простого - документа **ПриходнаяНакладная**.

6. Откроем в конфигураторе окно редактирования объекта конфигурации **Документ ПриходнаяНакладная** и перейдем на закладку **Движения**. В списке регистров отметим, что документ будет создавать теперь и по регистру **СтоимостьМатериалов**.

7. Не будем использовать конструктор движений документа, а внесем необходимые дополнения прямо в обработчик **ОбработкаПроведения** документа **ПриходнаяНакладная**. С помощью конструктора можно создавать движения одновременно и в нескольких регистрах (в конструкторе движений можно добавлять регистры). Но когда процедура проведения документа уже написана, использование конструктора приведет к тому, что имеющаяся процедура сохранена не будет. То есть сейчас при использовании конструктора пришлось бы заново описывать движения как для одного, так и для другого регистра. Поэтому проще внести изменения в существующую процедуру «вручную».

8. Перейдем на закладку **Прочее** и откроем модуль объекта - нажмем кнопку **Модуль объекта**. Откроем процедуру обработчика события **ОбработкаПроведения**. В самом конце цикла перед строкой **КонецЦикла** добавим строки кода, создающие движения в регистре **СтоимостьМатериалов**

Листинг 1. Движения документа «ПриходнаяНакладная» (фрагмент)

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

Перед началом цикла установим для набора записей движений по этому регистру свойство **Записывать** в значение **Истина**, чтобы платформа автоматически записала созданные нами движения после выхода из процедуры проведения документа. И еще удалим комментарии, внесенные конструктором.

В результате процедура **ОбработкаПроведения** будет **выглядеть** следующим образом (листинг 2). Движения документа «Приходная накладная»

Процедура ОбработкаПроведения(Отказ, Режим)

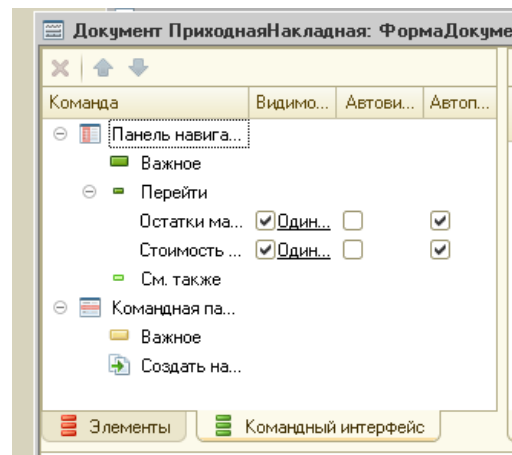
```
//{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
// регистр ОстаткиМатериалов Приход
Движения.ОстаткиМатериалов.Записывать = Истина;
Для Каждого ТекСтрокаМатериалы Из Материалы Цикл
    //регистр ОстаткиМатериалов Приход
    Движение = Движения.ОстаткиМатериалов.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    Движение.Период = Дата;
    Движение.Материал = ТекСтрокаМатериалы.Материал;
    Движение.Склад = Склад;
    Движение.Количество = ТекСтрокаМатериалы.Количество;
//регистр стоимость материалов Приход
    Движение=Движения.СтоимостьМатериалов.Добавить();
    Движение.ВидДвижения=ВидДвиженияНакопления.Приход;
    Движение.Период=Дата;
    Движение.Материал=ТекСтрокаМатериалы.Материал;
    Движение.Стоимость=ТекСтрокаМатериалы.Сумма;
КонецЦикла;
КонецПроцедуры;
```

Команда перехода к записям регистра

Отредактируем командный интерфейс формы документа, чтобы в панели навигации формы иметь возможность переходить к списку записей регистра **СтоимостьМатериалов**, связанному с документом. Для этого открыть форму документа **ПриходнаяНакладная**.

9. В левом верхнем окне перейдем на закладку **Командный интерфейс**. В разделе **Панель навигации** раскроем группу **Перейти** и увидим команду для открытия регистра накопления **Стоимость материалов**. Установим свойство **Видимость** для этой команды.

10. В режиме 1С:Предприятие проведем еще раз (перепроведем) все приходные накладные. Это необходимо для того, чтобы эти документы создали новые записи в регистрах, в соответствии с алгоритмом проведения, который мы только что изменили. Откройте список документов, выполнив команду **Приходные накладные** в панели навигации раздела **Учет материалов**. Выделите одновременно, используя клавишу **Ctrl**, все приходные накладные и перепроведем их, выполнив команду **Все действия** ►



Провести.

11. Затем откроем первый документ и, выполнив команды перехода к регистрам **Остатки материалов** и **Стоимость материалов**, убедимся, что документ создает желаемые записи как в одном, так и в другом регистре накопления.

Проведение документа «Оказание услуги» по двум регистрам

Внесем изменения в процедуру обработки проведения документа **ОказаниеУслуги**.

При списании материалов, израсходованных в процессе Оказания услуги, должна быть возможность указывать различную стоимость для одного и того же материала, которая рассчитана руководством исходя из текущих конъюнктурных соотношений.

Так как в документе **ОказаниеУслуги** отражена только цена номенклатуры, то нам понадобится:

- добавить в табличную часть документа еще один реквизит, в котором будет указываться стоимость номенклатуры;
- после этого изменить процедуру проведения документа **ОказаниеУслуги**.

В режиме 1С:Предприятие перепроведем все эти документы, чтобы отработал новый, измененный алгоритм проведения документов Оказание услуги.

12. Откроем в конфигураторе окно редактирования объекта конфигурации Документ **ОказаниеУслуги** и перейдем на закладку **Данные**.

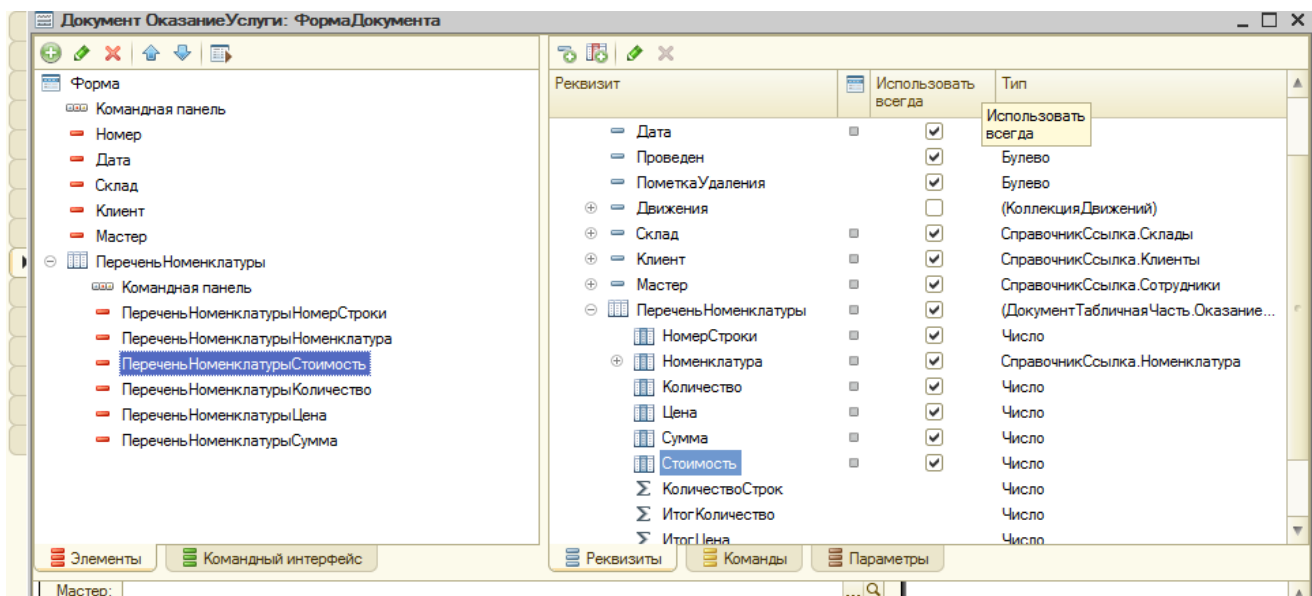
13. Создать новый реквизит табличной части документа с именем **Стоимость**, типом **Число**, длиной **15** и точностью **2**, неотрицательное.

14. Откроем форму **ФормаДокумента** документа **ОказаниеУслуги** и добавим в табличную часть **ПереченьНоменклатуры** поле отображающее новый реквизит **Стоимость**.

Для этого в правом верхнем окне редактора форм на закладке **Реквизиты** раскроем реквизит формы **Объект** и как видим он содержит все реквизиты документа ОказаниеУслуги.

Найдем в табличной части реквизит **Стоимость** и с помощью мыши перетащим его в окно элементов формы, расположенное слева в верхней части редактора форм. Новый элемент расположим в структуре элементов формы после **Номенклатура**. Оставим свойства элемента формы, предложенные по умолчанию.

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



Изменение процедуры проведения

15. Создадим движения документа **ОказаниеУслуги** таким же образом, как это делали для документа **ПриходнаяНакладная**. В окне редактирования объекта конфигурации **Документ Оказание Услуги** перейдем на закладку **Движения**. В списке регистров отметим, что документ будет создавать теперь движения по регистру **СтоимостьМатериалов**.

16. Перейдите на закладку **Прочее** и откройте модуль объекта, нажав на кнопку **Модуль объекта**.

17. Открыть процедуру обработчика события **ОбработкаПроведения**. В самом конце цикла перед строкой **КонецЕсли** добавить строки кода, создающие движения регистра **СтоимостьМатериалов**, производимые документом **ОказаниеУслуги**.

Листинг 2. Движения документа Оказание Услуги (фрагмент)

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

18. Перед началом цикла установим свойство **Записывать** набора записей движений по этому регистру в значение **Истина**. Удалим комментарии, внесенные конструктором. В результате процедура **ОбработкаПроведения** будет выглядеть следующим образом (листинг).

Листинг 3. Движения документа Оказание Услуги
Процедура ОбработкаПроведения(Отказ, Режим)

```
Движения.ОстаткиМатериалов.Записывать = Истина;
Движения.СтоимостьМатериалов.Записывать = Истина;
Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры =
Перечисления.ВидыНоменклатуры.Материал Тогда
// регистр ОстаткиМатериалов Расход
Движение = Движения.ОстаткиМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Склад = Склад;
Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;
// регистр СтоимостьМатериалов Расход
Движение = Движения.СтоимостьМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Количество * ТекСтрокаПереченьНоменклатуры.
Стоимость;
КонецЕсли;
КонецЦикла;
КонецПроцедуры;
```

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

документа **ОказаниеУслуги**. В верхнем окне перейдем на закладку **Командный интерфейс**, в разделе **Панель навигации** раскроем группу **Перейти** и увидим команду для открытия регистра накопления **Стоимость материалов**, установим свойство **Видимость** для этой команды.

20. Запустим 1С:Предприятие в режиме отладки и откроем список документов. Проведем документ **Оказание услуги № 1** и посмотрим на движения этого документа по регистру **Стоимость материалов**. Для этого нажмем кнопку **Провести** и выполним команду перехода к регистру **Стоимость материалов**.

21. Создать и провести еще два документа **Оказание услуги**.

N	Номенклатура	Стоимость	Коли...	Цена	Сумма
1	Строчный трансформатор...	800,00	1,000	800,00	800,00
2	Ремонт импортного теле...	600,00	1,000	800,00	800,00
Всего:					1 700,00

Номенклатура	Стоимость	Коли...	Цена	Сумма
1 Подключение воды		1,000	800,00	800,00
2 Шланг резиновый	100,00	2,000	150,00	300,00
3 Кабель электрический	20,00	1,000	30,00	30,00
4 Ремонт отечественного т...		1,000	600,00	600,00
5 Транзистор Philips 2N2369	3,00	2,000	7,00	14,00
6 Строчный трансформатор...	270,00	1,000	400,00	400,00
Всего:				2 144,00

Движения документов **Оказание услуги** по регистру **Стоимость материалов** должны выглядеть следующим образом:

Период	Регистратор	Номер строки	Материал	Стоимость
30.03.2014 14:14:01	Оказание услуги 000000002 от ...	1	Строчный трансформатор Sam...	800,00

Период	Регистратор	Номер строки	Материал	Стоимость
30.03.2014 13:26:59	Оказание услуги 000000003 от ...	1	Шланг резиновый	200,00
30.03.2014 13:26:59	Оказание услуги 000000003 от ...	2	Кабель электрический	20,00
30.03.2014 13:26:59	Оказание услуги 000000003 от ...	3	Транзистор Philips 2N2369	6,00
30.03.2014 13:26:59	Оказание услуги 000000003 от ...	4	Строчный трансформатор Gold...	270,00

Контрольные вопросы

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

ПРАКТИЧЕСКАЯ РАБОТА № 10

Тема работы: Оборотные регистры накопления

Цель работы: Научиться создавать различные оборотные регистры накопления.

Ход работы

Продолжим рассматривать работу нашего документа **ОказаниеУслуги**.

До сих пор мы создавали в регистрах накопления движения только для строк документа, которые содержат материалы. Услуги, содержащиеся в документе никак не учитывали. При учете услуг важны совершенно другие критерии, нежели при учете материалов.

Прежде всего, бессмысленно говорить о том, сколько услуг было и сколько их осталось, важна только сумма и количество услуг, которые были оказаны за определенный промежуток времени. Кроме этого, имеет значение:

- какие именно услуги были оказаны (чтобы составить рейтинг услуг);
- какому именно клиенту оказывались услуги (чтобы, например, предоставить ему скидку от объема оплаченных ранее услуг);
- какой мастер предоставлял услуги (чтобы начислить ему заработную плату).

Так как существующие регистры накопления не подходят для решения таких задач надо создать еще одно «хранилище» данных, которое будет использоваться в нашей программе, - оборотный регистр накопления **Продажи**.

Регистры накопления могут быть *регистрами остатков* и *регистрами оборотов*. Существующие в учебной конфигурации регистры **ОстаткиМатериалов** и **СтоимостьМатериалов** являются регистрами остатков.

При создании отчета **Материалы** в конструкторе запроса мы видели, что для таких регистров система создает три виртуальные таблицы: таблица остатков, оборотов и совокупная таблица остатков и оборотов.

Оборотный регистр накопления очень похож на регистр остатков, но для него понятие «остаток» не имеет смысла, оборотный регистр накапливает только обороты, остатки ему безразличны. Поэтому единственной виртуальной таблицей, которую будет создавать система для такого регистра, будет таблица оборотов. В остальном оборотный регистр ничем не отличается от регистра остатков. При создании оборотного регистра накопления нет особой сложности в определении того, какие именно данные должны являться измерениями регистра - мы можем назначить в качестве его измерений любые нужные нам данные.

Выбор измерений для регистра накопления должен выполняться исходя из того, что движения регистра могут быть осуществляться в две стороны: приход и расход. Таким образом, в качестве измерений нужно выбирать те данные, по которым движения точно осуществляться как в одну, так и в другую сторону.

Например, если ведется учет материалов в разрезе номенклатуры и склада, очевидно, что и номенклатура, и склад могут быть измерениями, поскольку как приход, так и расход материалов всегда будут осуществляться с указанием конкретной номенклатуры и конкретного склада. Если же в этой ситуации появляется желание отразить учет материалов еще и в разрезе поставщика, то здесь уже нужно исходить из конкретной схемы учета, принятой на предприятии.

При поступлении материалов поставщик будет указан, а вот при расходовании материалов, это совершенно лишняя информация. Значит, поставщика следует добавить не как измерение, а как реквизит регистра накопления. Если же при расходе материалов поставщик будет указываться наверняка, имеет смысл добавить поставщика в измерения регистра.

Иными словами, по каждому из измерений регистра накопления остатков ресурсы обязательно должны изменяться в обе стороны: приход и расход. Не должно существовать таких измерений, по которым осуществляется только приход или только расход. Нарушение этого принципа построения регистров накопления будет вести к непроизводительному использованию ресурсов системы и как следствие к замедлению работы и падению производительности.

Для реквизитов же регистра этот принцип не важен. По реквизитам регистра ресурсы могут только приходоваться или только расходоваться.

Добавление оборотного регистра накопления.

1. В режиме Конфигуратор создадим новый объект конфигурации **Регистр накопления** и назовем его **Продажи** и определим вид регистра - **Обороты**.

2. Зададим **Расширенное представление списка** как **Движения по регистру Продажи**. Этот заголовок будет отображаться в окне списка записей регистра. На закладке Подсистемы отметим, что этот регистр будет отображаться в подсистемах **Бухгалтерия**, **Учет материалов** и **Оказание услуг**.

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

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

У регистра будет три ресурса:

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

4. Отредактируем командный интерфейс, чтобы в подсистемах **Бухгалтерия**, **Оказание услуг** и **Учет материалов** была доступна ссылка осмотра нашего оборотного регистра накопления. В дереве объектов конфигурации выделим ветвь Подсистемы, вызовем ее контекстное меню и выберем пункт **Все подсистемы**. В открывшемся окне слева в списке **Подсистемы** выделим подсистему **Бухгалтерия**. Справа в списке **Командный интерфейс** отразятся все команды выбранной подсистемы. В группе **Панель навигации.Обычное** включим видимость у команды **Продажи** и мышью перетащим ее в группу **Панель навигации.См.также**.

5. Аналогично, выделить подсистемы **ОказаниеУслуг** и **УчетМатериалов**. в панели навигации включить видимость у команды **Продажи** и перенести ее в группу **См.также**.

Проведение документа «Оказание услуги» по трем регистрам.

6. В режиме Конфигуратор откроем окно редактирования объекта конфигурации **Документ ОказаниеУслуги** и на закладке **Движения** укажем, что этот документ будет создавать движения еще и по регистру **Продажи**.

7. Перейдем на закладку **Прочее** и откроем модуль документа, для этого нажмем кнопку **Модуль объекта**.

8. Открыть процедуру обработчика события **ОбработкаПроведения**. В конце цикла после строки **КонецЕсли** и перед строкой **КонецЦикла** добавим строки кода, создающие движения регистра **Продажи**, производимые документом **ОказаниеУслуги** (листинг 1).

Перед началом цикла установим свойство **Записывать набора записей движений** по этому регистру в значение **Истина**. В результате процедура **ОбработкаПроведения** будет выглядеть следующим образом:

Листинг 1. Движения документа «ОказаниеУслуги»

Процедура ОбработкаПроведения(Отказ, Режим)

Движения.ОстаткиМатериалов.Записывать = Истина;

Движения.СтоимостьМатериалов.Записывать = Истина;

Движения.Продажи.Записывать = Истина;

Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл

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

Тогда

// регистр ОстаткиМатериалов Расход

Движение = Движения.ОстаткиМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;

Движение.Склад = Склад;

Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;

// регистр СтоимостьМатериалов Расход

Движение = Движения.СтоимостьМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;

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

КонецЕсли;

// Регистр Продажи

Движение = Движения.Продажи.Добавить();

Движение.Период = Дата;

Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенклатура;

Движение.Клиент = Клиент;

Движение.Мастер = Мастер;

Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;

Движение.Выручка = ТекСтрокаПереченьНоменклатуры.Сумма;

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

КонецЦикла;

КонецПроцедуры

У оборотного регистра отсутствует свойство ВидДвижения, поскольку отражение вида движения (приход или расход) имеет смысл лишь при учете остатков. В случае регистра оборотов нас интересует только значение, которое должно быть записано в ресурс регистра. Также заметьте, что мы разместили команды, создающие движения в регистре Продажи, в конце цикла обхода строк табличной части документа, после условия выполнения цикла только для материалов. Это важно, так как движения в этом регистре создаются как для материалов, так и для услуг.

9. Отредактируем командный интерфейс формы документа, чтобы в панели навигации формы иметь возможность переходить к списку записей регистра Продажи, связанному с документом. Для этого откроем форму документа **ОказаниеУслуги**. В левом верхнем окне перейдем на закладку **Командный интерфейс**. В разделе **Панель навигации** раскроем группу **Перейти** и увидим команду для открытия регистра накопления **Продажи**. Установим свойство **Видимость** для этой команды.

10. В режиме 1С:Предприятие нужно перепровести все документы оказания услуг и проверить, что они создают правильные движения в регистре **Продажи**.

11. Запустим 1С:Предприятие в режиме отладки и откроем по очереди каждый документ **Оказание услуги**. Нажмем **Провести** и перейдите к списку движений этих документов по регистру **Продажи** и просмотрите их.

Контрольные вопросы

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