Автономная некоммерческая организация среднего профессионального образования

«УРАЛЬСКИЙ ПРОМЫШЛЕННО-ЭКОНОМИЧЕСКИЙ ТЕХНИКУМ»

**Методические указания**

**к практическим работам**

по дисциплине

**Разработка приложений в среде 1С Предприятие**

**Укрупненная группа: 09.00.00 И**нформатика и вычислительная техника

**Специальность:** 09.02.03 Программирование в компьютерных системах

2016

|  |  |
| --- | --- |
| Одобрена цикловой комиссией  информатики и вычислительной техники  Председатель комиссии  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ О. Г. Максимова  Протокол №  от « » \_\_\_\_\_\_\_\_\_\_\_\_ 201\_\_г. | Методические указания разработаны на основе Федерального государственного образовательного стандарта по специальности среднего профессионального образования, входящей в состав укрупненной группы специальностей 09.00.00 «Информатика и вычислительная техника» 09.02.03 «Программирование в компьютерных системах»  *УТВЕРЖДАЮ*  Заместитель директора по  учебной работе АН ПОО «Уральский промышленно-экономический техникум»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.Б. Чмель  « » \_\_\_\_\_\_\_\_\_\_\_\_ 201 \_\_ г. |

Разработчик: **Максимова О.Г.** преподаватель спецдисциплин

АН ПОО «Уральский промышленно-экономический техникум»

Содержание

[Пояснительная записка 6](#_Toc480020383)

[Практическая работа № 1 Создание информационной базы 8](#_Toc480020384)

[Практическое задание № 2 Создание подсистем 13](#_Toc480020385)

[Практическое задание № 3 Создание справочников 18](#_Toc480020386)

[Практическая работа № 4 Создание документов 21](#_Toc480020387)

[Практическая работа № 5 Редактирование макетов и форм 24](#_Toc480020388)

[Практическая работа № 6 Периодические регистры сведений 28](#_Toc480020389)

[Практическая работа № 7 Перечисления 32](#_Toc480020390)

[Практическая работа № 8 Проведение приходной накладной и документа «Оказание услуги» по двум регистрам 35](#_Toc480020391)

[Практическая работа № 9 Добавление оборотного регистра накопления. Проведение документа «Оказание услуги» по трём регистрам 41](#_Toc480020392)

[Практическая работа № 10 Выбор данных из одной и двух таблиц. Вывод данных по всем дням в выбранном периоде 45](#_Toc480020393)

[Практическая работа № 11 Получение актуальных значений из периодического регистра сведений 55](#_Toc480020394)

[Практическая работа № 12 Использование вычисляемого поля в отчёте. Вывод данных в таблицу 59](#_Toc480020395)

[Практическая работа № 13 Повышение скорости проведения документа. Автоматический расчёт стоимости 61](#_Toc480020396)

[Практическая работа № 14 Контроль остатков. Блокировка данных. 71](#_Toc480020397)

[Практическая работа № 15 Доработка объектов конфигурации и учётных механизмов 77](#_Toc480020398)

[Практическая работа № 16 Создание отчёта с использованием характеристик 81](#_Toc480020399)

[Практическая работа № 17 Составление бухгалтерского баланса 86](#_Toc480020400)

[Практическая работа № 18 Корреспонденция счетов. Проводки 90](#_Toc480020401)

[Практическая работа № 19 Использование регистров расчёта 97](#_Toc480020402)

[Практическая работа № 20 Создание регламентных заданий 104](#_Toc480020403)

[Практическое задание № 21 Создание ролей. Ограничение доступа к данным на уровне записей и полей базы данных 107](#_Toc480020404)

[Практическая работа № 22 Интерактивный обмен. Программный обмен 119](#_Toc480020405)

[Практическая работа № 23 Подбор с использованием множественного выбора 134](#_Toc480020406)

[Практическая работа № 24 Получение объектов, введённых на основании 136](#_Toc480020407)

[Практическая работа № 25 Разработка форм 138](#_Toc480020408)

[Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы 145](#_Toc480020409)

# Пояснительная записка

Программа учебной дисциплины «разработка приложений в среде «1С: Предприятие»» является вариативной частью основной профессиональной образовательной программы в соответствии с ФГОС по специальности среднего профессионального образования 09.02.03 «Программирование в компьютерных системах», входящей в состав укрупненной группы специальностей «Информатика и вычислительная» техника.

Рабочая программа учебной дисциплины может быть использована в дополнительном профессиональном образовании (в программах повышении квалификации и переподготовки).

Целью изучения дисциплины является формирование у студентов представлений о численных методах решения основных математических задач на ЭВМ

Дисциплина рассчитана на студентов, освоивших курсы учебных дисциплин «Основы экономики», «Основы программирования», «Технология разработки и защиты баз данных» и «Экономика организации».

В результате освоения рабочей программы обучающийся должен

*знать:*

* как современные информационные технологии и средства автоматизации способствуют совершенствованию деятельности предприятия;
* базовые конструкции языка программирования «1С: Предприятие»;

*уметь:*

* определять степень соответствия получаемых полезных эффектов желаемым, а также уровень выполнения существующих ограничений для каждого альтернативного варианта применения ИТ на предприятии;
* работать в среде программирования;
* реализовывать построенные алгоритмы в виде программ на языке программирования «1С: Предприятие».

Содержание дисциплины должно быть ориентировано на подготовку обучающихся по базовой и углубленной подготовке к освоению профессиональных модулей ППССЗ по специальности «Программирование в компьютерных системах» и овладению профессиональными компетенциями (ПК):

* ПК 1.2 Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля;
* ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств;
* ПК 1.4 Выполнять тестирование программных модулей:
* ПК 1.5 Осуществлять оптимизацию программного кода модуля.

В результате освоения дисциплины у обучающихся по базовой подготовке формируются общие компетенции (ОК):

* ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.
* ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.
* ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.
* ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.
* ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.
* ОК 6. Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.
* ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.
* ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.
* ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.
* ОК 10. Исполнять воинскую обязанность, в том числе с применением полученных профессиональных знаний (для юношей).

Рабочей программой дисциплины предусмотрено 52 часа практических занятий

# Практическая работа № 1 Создание информационной базы

**Цель:** Создание информационной базы

**Норма времени**: 1 час

**Методическое обеспечение**: методические указания к практической работе.

**Ход работы**

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

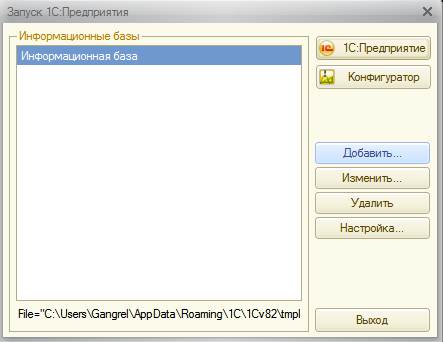
**

Рисунок – Создание новой информационной базы

В открывшемся диалоге выберите пункт **«Создание новой информаци­онной базы»;**

Нажмите кнопку **«Далее»** На следующем шаге выберите пункт **«Создание информационной базы без конфигурации»;**

Нажмите кнопку **«Далее».** На следующем шаге задайте наименование вашей информационной базы и выберите тип ее расположения **«На данном компьютере»;**

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

**Знакомство с конфигуратором:**

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

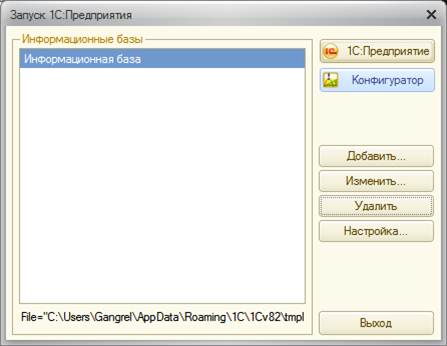


Рисунок – Конфигуратор

Перед вами окно конфигуратора:

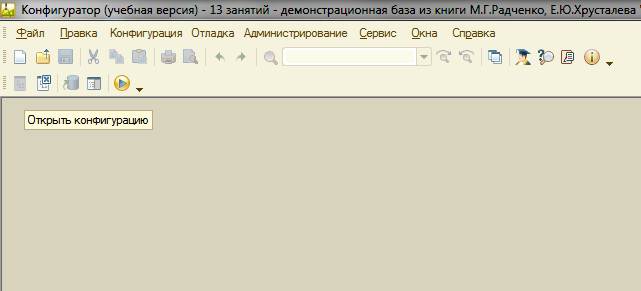


Рисунок – Окно конфигуратора

Именно с помощью этого инструмента мы будем создавать нашу конфигурацию. 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.

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

Ha экране *откроется дерево объектов конфигурации:*

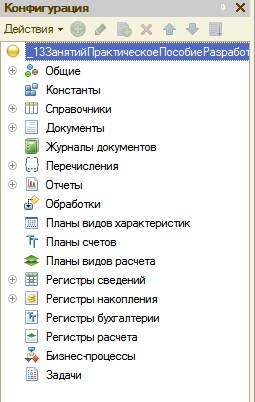


Рисунок – Дерево объектов конфигурации

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

**Что такое объекты конфигурации:**

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

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

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

То же и с объектами конфигурации. Мы можем создавать только объекты определенных видов. Но каждого вида объектов мы можем создать столько, сколько нам нужно. Объекты одного вида отлича­ются от объектов другого вида тем, что имеют разные свойства. Объекты могут взаимодейст­вовать друг с другом, и мы можем описать такое взаимодействие.

Объекты конфигурации также обладают различным поведением, и оно зависит от вида объекта. Одни объекты могут выполнять какие-то действия, другие этих действий выполнять не могут, зато у них есть свой собственный набор действий.

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

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

**Как добавить объект конфигурации:**

Перед началом работы следует объяснить некоторые приемы работы с конфигуратором:

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

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

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

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

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

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

**Палитра свойств:**

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

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

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

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

**Запуск отладки в режиме 1С: Предприятие:**

Проверим первые изменения в режиме 1С:Предприятие.

Для этого выполним пункт меню **Отладка. > Начать отладку** или нажмем соответствующую кнопку **>** на панели инструментов конфигуратора. Система сама анализирует наличие изменений в конфигурации и выдает соответствующий вопрос об обновлении конфигурации базы данных.

На вопрос конфигуратора ответим **«ДА»**, и на экране появится окно 1С:Предприятия.

В режиме 1С: Предприятие:

Внешний вид интерфейса прикладного решения

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

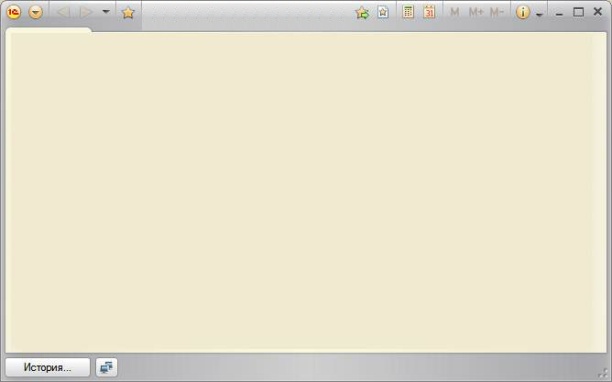


Рисунок – Внешний вид интерфейса прикладного решения

--

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

# Практическое задание № 2 Создание подсистем

**Цель:** Создание подсистем

**Норма времени**: 1 час

**Методическое обеспечение**: методические указания к практической работе.

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

Сейчас мы создадим в нашей конфигурации пять новых объектов конфигурации Подсистема, которые будут иметь имена: «**Бухгалтерия», «РасчетЗарплаты», «УчетМатериалов», «ОказаниеУслуг»**и**«Предприятие».**Чтобы это сделать, выполним следующие действия.

**Добавление подсистемы в режиме «Конфигуратор»:**

Чтобы создать новые подсистемы, раскроем ветвь «**Общие»** в дереве объектов конфи­гурации, нажав на **+** слева от нее.

Затем выделим ветвь «**Подсистемы»**, вызовем ее контекстное меню и выберем пункт «**Добавить»** или нажмем соответствующую кнопку в командной панели окна конфигурации.

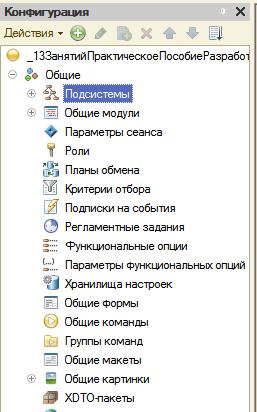


Рисунок – Дерево объектов конфигурации

После этого система откроет окно редактирования объекта конфигурации.

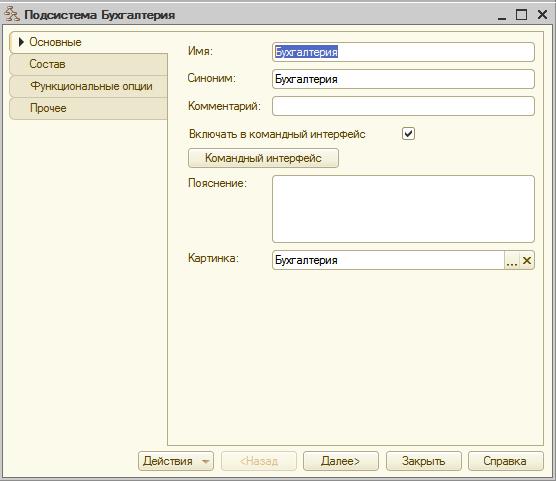


Рисунок – Подсистема Бухгалтерия

 Зададим имя подсистемы – «**Бухгалтерия»**. На основании имени платформа автоматически создаст синоним – «**Бухгалтерия»**.

**Имя и синоним объекта конфигурации:**

**Имя** является основным свойством любого объекта конфигурации. При создании нового объекта система автоматически присваивает ему некоторое имя.

Свойство **Синоним** также есть у любого объекта конфигурации. Оно предназначено для хранения «альтернативного» наименования объекта конфигурации, которое будет использовано в элементах интерфейса нашей программы, то есть будет показано пользователю. Поэтому на синоним практически нет никаких ограничений.

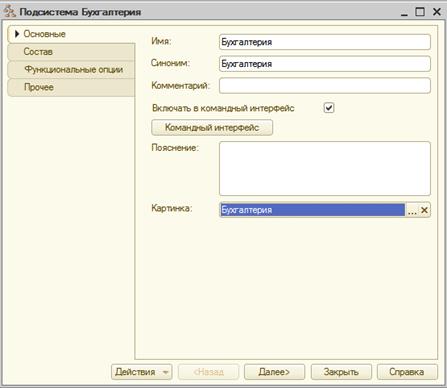
****

Рисунок – Подсистема Бухгалтерия

В целях усовершенствования интерфейса приложения мы можем также задать картинку для отображения подсистемы.

Система создаст объект конфигурации Общая картинка и откроет окно редактирования его свойств.

Картинке можно присвоить свое имя, а так же выбрать свое изображение с локального диска. Для просмотра изображений поставим флажок «**Просмотр».**Выбранная нами картинка появится в окне редактирования общей картинки. Закроем окно редактирования объекта конфигурации

Итак, мы вернулись в окно редактирования объекта конфигурации Подсистема Бухгалтерия. Мы видим, что выбранная нами одноименная картинка установилась в качестве картинки для подсистемы.

Таким образом, в интерфейсе 1С Предприятия в качестве названия раздела будет показан синоним подсистемы, и над ним будет выводиться указанная картинка.

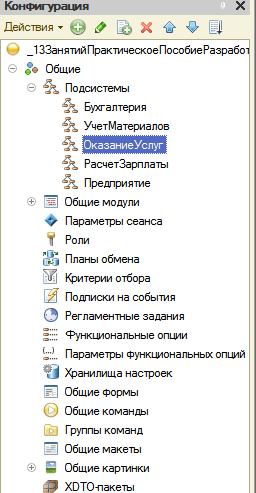


Рисунок – Дерево объектов конфигурации

Снова выделим ветвь «**Подсистемы**», нажмем кнопку «**Добавить**» в дереве объектов конфигурации и создадим подсистемы с именами «**Учет Материалов»** и «**Оказание Услуг»**. Установим для них картинки.

Теперь воспользуемся другим способом для добавления подсистем. Вызовем контекстное меню одной из созданных подсистем. Выберем в нем пункт «**Добавить»**. Он разбивается на два подпункта. Выбор подпункта «**Подсистема»** позволяет добавить подсистему того же уровня иерархии, что и выделенная. Выбор подпункта «**Подчиненная** **Подсистема»** позволяет добавить подсистему, подчиненную выделенной. Поскольку в нашей конфигурации не планируется сложной много­уровневой структуры, выберем первый вариант и добавим подсистему «**Расчет Зарплаты».** Установим для нее в качестве картинки общую картинку «**Зарплата»**, добавив ее из файла «**Зарплата»**.

В заключение добавим подсистему Предприятие для доступа к административным и сервисным функциям.

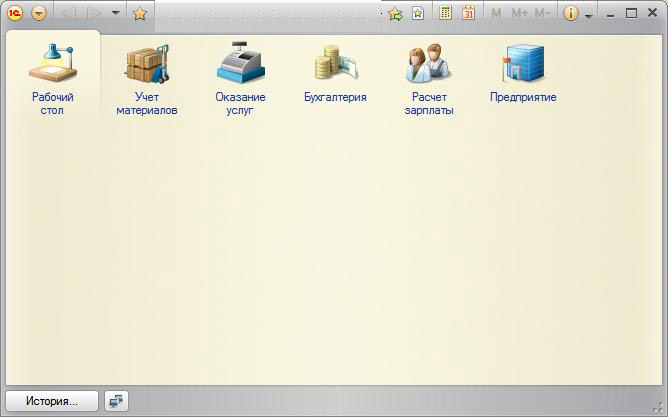


Рисунок – Подсистемы

**Панель разделов прикладного решения в режиме 1С:Предприятие:**

Запустим 1С Предприятие в режиме отладки и увидим результат наших изменений. Вид разрабатываемого нами приложения изменился. Сразу под главным меню располагается панель разделов приложения, где и отражены созданные нами подсистемы. Причем все разделы выводятся с выбранными в их свойствах картинками. Разделы представлены в форме гиперссылок, нажав на которые пользователь может открыть связанные с ними документы, справочники, отчеты и т. п.

Обратите внимание, что раздел **Рабочий стол** формируется платформой по умолчанию. Он предназначен для размещения наиболее часто используемых пользователем документов, отчетов и т. п.

**Порядок разделов в режиме Конфигуратор:**

Однако порядок расположения подсистем нас не совсем устраивает. Изменим его.

Закроем приложение и вернемся в конфигуратор. Выделим корень дерева объектов конфигурации «**Фамилия»,** нажатием правой кнопки мыши вызовем контекстное меню и выберем пункт «**Открыть командный интерфейс конфигурации»**

В открывшемся окне «**Командный интерфейс»** вы увидите список соз­данных вами подсистем (разделов приложения). Расположим сначала подсистемы, отражающие производственную деятельность нашей фирмы: «**Учет материалов**» и «**Оказание услуг**», затем бухгалтерскую деятельность и расчет зарплаты сотрудников: «**Бухгалтерия**» и «**Расчет зарплаты**», а затем подсистему «**Предприятие**».

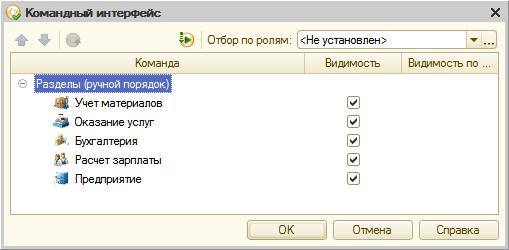


Рисунок – Командный интерфейс

**В режиме 1С:Предприятие:**

Запустим 1С Предприятие в режиме отладки и увидим, что порядок расположения подсистем в панели разделов приложения изменился так, как мы его задали.

*Примечание:*

После успешного завершения каждого занятия рекомендуется сохранять конфигурацию, выполнив команду главного меню **Адми­нистрирование > Выгрузить информационную базу.**Для загрузки БД выполните:**Администрирование > Загрузить информационную базу.**

**Окно редактирования объекта конфигурации и палитра свойств**

На первый взгляд окно редактирования объекта и палитра свойств дублируют друг друга. *Окно редактирования объекта конфигурации предназначено в первую очередь для быстрого создания новых объектов.* Окно редактирования объекта помогает быстро создать незнакомый объект конфигурации и обеспечивает удобный доступ к нужным свойствам.

Что же касается палитры свойств, то она предоставляет одну абсолютно незаменимую возможность. Дело в том, что она не привязана по своей структуре к какому-то конкретному виду объектов конфигурации. Ее содержимое меняется в зависимости от того, какой объект является текущим. За счет этого она может «запоминать», какое свойство объекта в ней выбрано, и при переходе в дереве к другому объекту будет подсвечивать у себя все то же свойство, но уже другого объекта.

# Практическое задание № 3 Создание справочников

**Цель:** Создание справочников

**Норма времени**: 2 часа

**Методическое обеспечение**: методические указания к практической работе.

В «1С: Предприятие» справочники служат для хранения условно постоянной информации, поддержки внутрифирменных или общих классификаторов. Также справочники обычно используются для организации аналитического учета.

Сведения о структуре организации заносятся в справочник «Подразделения организаций». Данный справочник поддерживает создание многоуровневой структуры организации, позволяет отразить структуру подразделений для каждой организации, входящей в компанию. Для входа в справочник «Подразделения организаций» воспользуемся меню:⎫ **Предприятие → Подразделения организаций.**

Обязательно перед вводом сведений о подразделениях **введем наименование организации**.

Наше предприятие состоит из следующих подразделений:

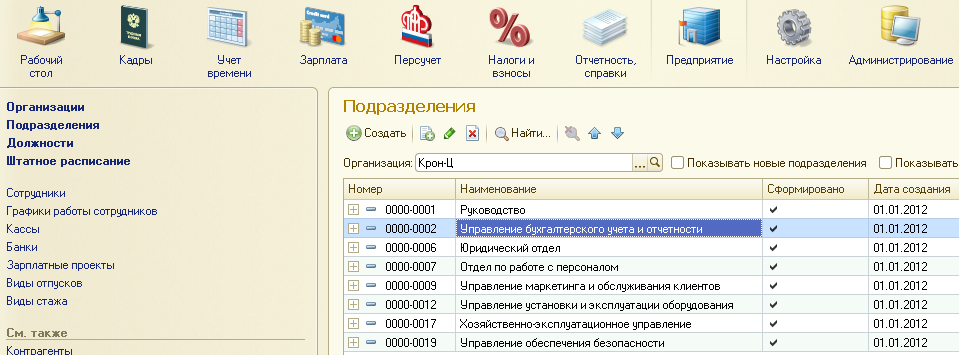


Рисунок –Предприятие

Установим курсор на строчке **«Подразделения»** и, нажав левую кнопку мыши. В полученном окне вводим только наименование подразделения, например «**Дирекция**». Нажмем кнопки «**Записать и закрыть**». Аналогичным способом можно добавить следующее подразделение.

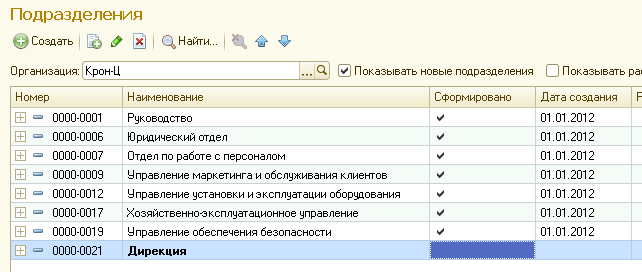


Рисунок – Подразделения

Подразделения организации

Возможности программы гораздо шире. Она позволяет учитывать территориальные особенности подразделения при начислении зарплат, например, если подразделение находится в районах Крайнего Севера и при начислении зарплаты действует иной коэффициент.

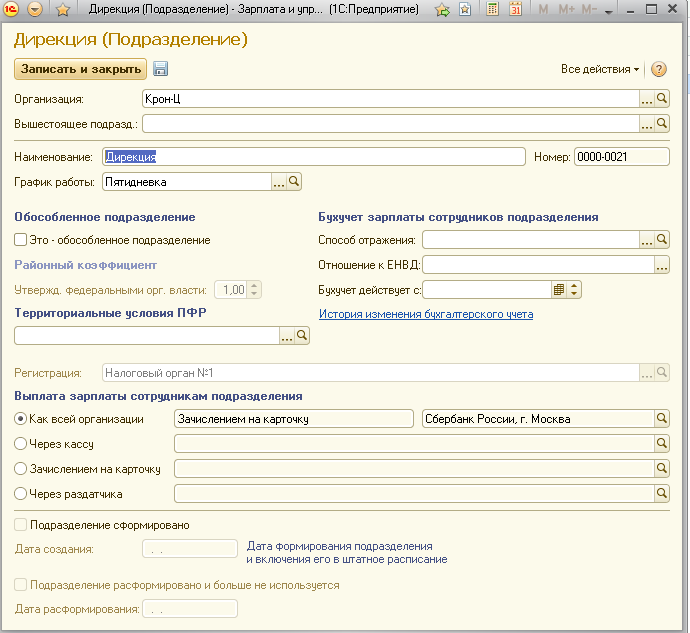


Рисунок – Дирекция

. Следующий справочник, называется⎫ **«Должности организаций»,** его можно найти, воспользовавшись меню: **Предприятие → Должности**. Его можно заполнять, используя кнопку **«Создать».**

Добавить следующие должности:

⎫ **директор акционерного общества** (Административно-управленческий персонал; Статистический учет: Руководители; Воинский учет: Прочие специалисты); менеджер (Статистический учет: Специалисты; Воинский учет: Прочие специалисты);⎫ повар (Статистический учет: Рабочие; Воинский учет: Прочие рабочие).⎫Введем их по аналогии с вводом должности «Директор акционерного общества».

Полученный список можно сортировать. Допустим, нам нужно вывести данный список на печать, для этого выберем **Все действия →Вывести список.** В открывшемся окне выберем нужные колонки и нажмем кнопку «ОК»..

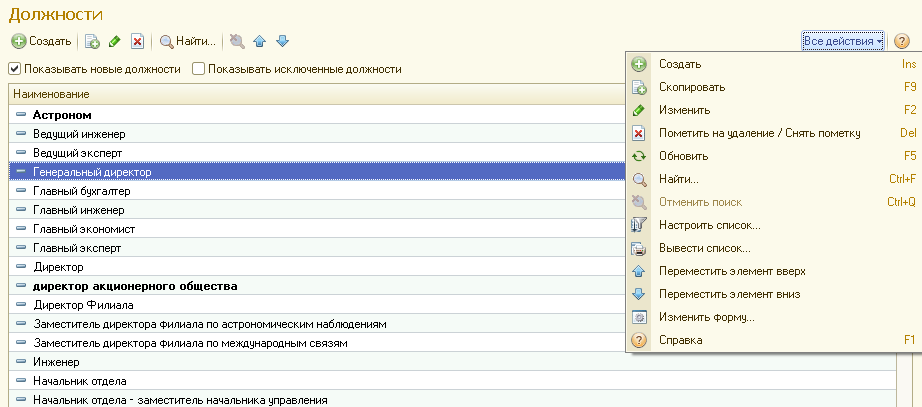


Рисунок – Должности

Этот список уже можно распечатать, воспользовавшись меню: **Файл →Печать**.

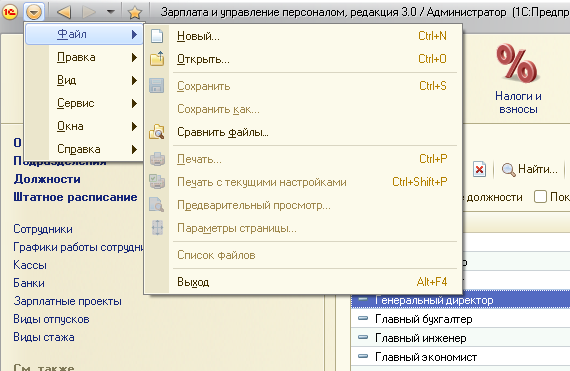


Рисунок –Список

# Практическая работа № 4 Создание документов

**Цель:** Создание документов

**Норма времени**: 2 часа

**Методическое обеспечение**: методические указания к практической работе.

Добавление документа

Откроем конфигуратор и добавим новый объект конфигурации *Документ*. На закладке *Основные* зададим имя документа – *Приходная Накладная*. На основании имени платформа автоматически заполнит его синоним. Здесь же определим, как будет представлен документ в интерфейсе «1С:Предприятия». *Представление объекта* задавать не будем, вместо него будет использоваться *Синоним* объекта. Это нам вполне подходит, так как мы задали его имя в единственном числе. *Представление списка*, наоборот, зададим во множественном числе как Приходные накладные Нажмем Далее и перейдем на закладку *Подсистемы*. По логике нашей конфигурации список приходных накладных должен быть доступен в разделах *Учет материалов* и *Бухгалтерия*. Действительно, к первому разделу этот документ имеет прямое отношение, а для бухгалтерского анализа всегда может понадобиться список документов, отражающих поступление материалов. Поэтому отметим в списке подсистем эти подсистемы. Перейдем на закладку Данные и создадим реквизит документа с именем *Склад.* Для этого нажмем кнопку *Добавить* над списком реквизитов документа Реквизиты ссылочного типа. Выберем для реквизита ссылочный тип данных *Справочник. Ссылка. Склады*. Этот тип стал доступен в конфигурации после создания объекта конфигурации *Справочник Склады* Свойство «Значение заполнения» реквизита объекта конфигурации. Теперь покажем, как можно облегчить жизнь пользователя при приходовании материалов. Работа в автоматизируемой нами фирме построена таким образом, что, как правило, все поступающие товары приходуются на основной склад. Поэтому в палитре свойств для созданного нами реквизита *Склад* документа найдем свойство *Значение заполнения.* В качестве значения этого свойства выберем предопределенный элемент справочника *Склады* – *Основной*. Таким образом, при создании нового документа склад будет сразу заполняться значением *Основной*, и пользователю не придется делать это вручную. После этого добавим в документ табличную часть с именем *Материалы*. Для этого нажмем кнопку *Добавить табличную часть* над списком табличных частей документа.

Проверка заполнения табличной части

Кроме имени табличной части установим свойство *Проверка заполнения* в значение *Выдавать ошибку*. Тем самым мы задаем условие, что документ *Приходная накладная* обязательно должен содержать табличную часть, то есть список приходуемых материалов. Иначе будет выдано сообщение об ошибке, и документ не будет сохранен. Создадим реквизиты табличной части *Материалы.* Для этого нажмем кнопку *Добавить реквизит* в разделе описания табличных частей документа Материал, тип Справочник. Ссылка. 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.

Для каждого реквизита табличной части также установим 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 Перейдем на закладку *Нумерация* и убедимся, что свойство *Автонумерация* включено. Это обеспечит автоматическую генерацию уникальных номеров для создаваемых нами документов. В заключение отредактируем 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 *Учет Материалов*. Справа в списке *Командный интерфейс* отразятся все команды выбранной подсистемы. В группе *Панель действий. Создать* включим видимость у команды *Приходная накладная: создать*

**В режиме «1С:Предприятие»**

Запустим «1С: Предприятие» в режиме отладки и протестируем получившийся документ.

В открывшемся окне «1С:Предприятия» мы видим, что в панели навигации разделов *Бухгалтерия* и *Учет материалов* появилась команда Приходные накладные для открытия списка приходных накладных. Название этой команды определяется свойством *Представление списка*, которое мы задали для этого документа. Также в панели действий раздела *Учет материалов* появилась команда *Приходная накладная* для создания новых документов этого вида. Название этой команды определяется синонимом, так как *Представление объекта* мы для этого документа не задавали Добавление приходных накладных. Пока в нашей базе данных нет ни одного документа *Приходная накладная*, поэтому выполним команду *Приходная накладная* в панели действий раздела *Учет материалов* и создадим новую приходную накладную. Перед нами откроется форма документа – основная форма объекта. Заголовок этой формы *Приходная накладная* совпадает с синонимом документа. Система автоматически подставит текущую дату создания документа и 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* для документа включено по умолчанию. Новый номер будет сохранен в момент записи документа. Обратите внимание, что склад уже заполнен значением *Основной*, как мы и задали в свойствах этого реквизита. Нам осталось только заполнить табличную часть приходной накладной материалами для ремонта телевизоров. Обратите внимание, что при нажатии кнопки выбора в поле *Материал* (в табличной части документа) открывается форма для выбора элементов справочника *Номенклатура*, так как этот реквизит имеет ссылочный тип данных и ссылается на справочник Номенклатура Нажмем Провести и закрыть. Документ будет сохранен и проведен, ему будет присвоен автоматически сгенерированный системой номер и текущее время проведения документа. Аналогичным образом мы создадим второй документ, который будет приходовать на Основной склад материалы для установки стиральных машин. Но теперь не будем использовать кнопку выбора в поле Материал, а просто начнем вводить название материала в это поле. Платформа автоматически найдет материалы, наименование которых начинается с введенных нами символов, и предложит их нам для выбора. Нажмем Провести и закрыть. Документ будет сохранен и проведен, ему будет присвоен автоматически сгенерированный системой номер и текущее время проведения документа.

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.Чтобы просмотреть список созданных документов, выполним команду *Приходные накладные* в панели навигации. В форме списка, открывшейся в рабочей области окна приложения, мы видим два созданных нами документа, отмеченных пиктограммой, указывающей на то, что документы проведены.

# Практическая работа № 5 Редактирование макетов и форм

**Цель:** Редактирование макетов и форм

**Норма времени**: 2 часа

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

Макеты. Редактирование макетов и форм.

Объект конфигурации макет предназначен для хранения различных форм представления данных.

**Макет печатной формы.**

Откроем окно редактирования документа «Оказание услуг», перейдем на закладку «Макеты», запустим конструктор печати («Конструктор – Конструктор печати»). В открывшемся окне укажем, что будет создана команда «Печать».

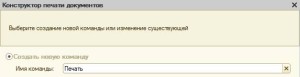


Рисунок –Конструктор печати документов

«Далее».

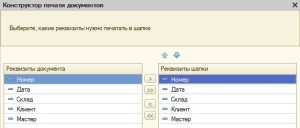
[](http://1c-ucheba.ru/wp-content/uploads/2013/05/%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80-%D0%98%D0%91-%D0%9C%D0%B0%D0%BA%D0%B5%D1%82%D1%8B002.jpg)

Рисунок –- Конструктор печати документов

«Далее».

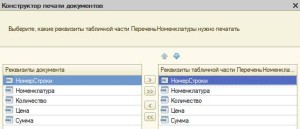
[](http://1c-ucheba.ru/wp-content/uploads/2013/05/%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80-%D0%98%D0%91-%D0%9C%D0%B0%D0%BA%D0%B5%D1%82%D1%8B003.jpg)

Рисунок – Конструктор печати документов

«Далее». Будет предложено сформировать подвал. Ничего менять не будем. «Далее», «Ок».

Конструктор создал:

Макет печатной формы документа «Оказание услуг» с именем «Печать». Команда «Печать»/

В командную панель формы документа «Оказание услуг» помещена команда «Печать».

Запустим 1С: Предприятие в режиме отладки [Пример ИБ Макеты004](http://1c-ucheba.ru/wp-content/uploads/2013/05/%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80-%D0%98%D0%91-%D0%9C%D0%B0%D0%BA%D0%B5%D1%82%D1%8B004.jpg). Откроем документ «Оказание услуг», распечатаем.

[](http://1c-ucheba.ru/wp-content/uploads/2013/05/%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80-%D0%98%D0%91-%D0%9C%D0%B0%D0%BA%D0%B5%D1%82%D1%8B005.jpg)

Рисунок –Документ «Оказание услуг»

**Редактирование макета. Добавление итогов.**

Для этого откроем макет «Печать» документа «Оказание услуг». Макет состоит из именованных областей. Добавим новую область «Всего» (выделим 2 пустые строки под табличной частью, выполним пункт главного меню «Таблица – Имена – Назначить имя»). Выровняем ширину колонок новой области с колонками табличной части. На предложение создать новый формат строк ответим утвердительно. В созданной области в колонке «Цена», напишем ВСЕГО:, а в колонке «Сумма» — Всего. По Документу. Вызовем окно свойств для последней заполненной ячейки. Свойство «Заполнение» зададим – «Параметр». Ячейки табличного документа могут содержать: текст (будет отображен на экране), параметр (параметр будет заменен некоторым значением), шаблон (текстовый шаблон в определенные места которого будут вставлены параметры).



Рисунок – Оказание услуг

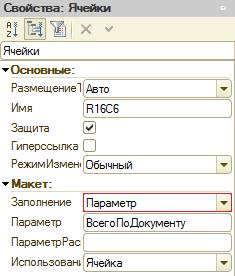


Рисунок – Свойства

Откроем модуль менеджера документа «Оказание услуг». Отредактируем процедуру печать.

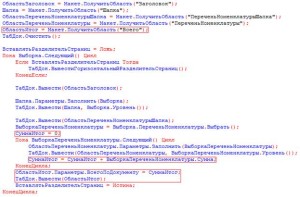
[](http://1c-ucheba.ru/wp-content/uploads/2013/05/%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80-%D0%98%D0%91-%D0%9C%D0%B0%D0%BA%D0%B5%D1%82%D1%8B008.jpg)

Рисунок – Модуль

Используя метод «Получить Область» получаем область «Всего», сохраняем в переменной «Область Итог». Вводим переменную для подсчета общей суммы. В цикле обхода строк табличной части аккумулируем общую сумму. Обращаемся к параметру «Всего По Документу» присваиваем ему «Сумма Итог». Выводим «Область Итог» в табличный документ. Запустим 1С: Предприятие в режиме отладки [Пример ИБ Макеты004](http://1c-ucheba.ru/wp-content/uploads/2013/05/%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80-%D0%98%D0%91-%D0%9C%D0%B0%D0%BA%D0%B5%D1%82%D1%8B004.jpg). Распечатаем документ.

[](http://1c-ucheba.ru/wp-content/uploads/2013/05/%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80-%D0%98%D0%91-%D0%9C%D0%B0%D0%BA%D0%B5%D1%82%D1%8B009.jpg)

Рисунок – Оказание услуг

**Редактирование формы, добавление итог в форму.**

Откроем форму документа «Оказание услуг». Вызовем окно свойств табличной части.

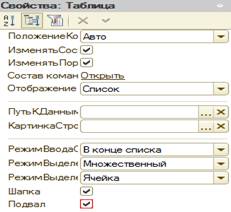
[](http://1c-ucheba.ru/wp-content/uploads/2013/05/%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80-%D0%98%D0%91-%D0%9C%D0%B0%D0%BA%D0%B5%D1%82%D1%8B010.jpg)

Рисунок – Свойства: Таблица

Откроем свойства элемента формы «Перечень Номенклатуры Цена» и установим текст подвала – Всего.

Откроем свойства элемента формы «Перечень Номенклатуры Сумма». Для того чтобы в подвале колонки «Сумма» отображался итог по ней, нажмем кнопку выбора в поле Путь К Данным Подвала, выберем «Итог Сумма».

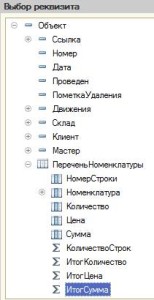
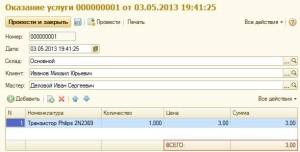
[](http://1c-ucheba.ru/wp-content/uploads/2013/05/%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80-%D0%98%D0%91-%D0%9C%D0%B0%D0%BA%D0%B5%D1%82%D1%8B011.jpg)

Рисунок – Выбор реквизита

Запустим 1С: Предприятие в режиме отладки [Пример ИБ Макеты004](http://1c-ucheba.ru/wp-content/uploads/2013/05/%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80-%D0%98%D0%91-%D0%9C%D0%B0%D0%BA%D0%B5%D1%82%D1%8B004.jpg).

[](http://1c-ucheba.ru/wp-content/uploads/2013/05/%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80-%D0%98%D0%91-%D0%9C%D0%B0%D0%BA%D0%B5%D1%82%D1%8B012.jpg)

# Практическая работа № 6 Периодические регистры сведений

**Цель:** Создание регистров сведений

**Норма времени**: 1 час

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

В режиме Конфигуратор

Откроем в конфигураторе конфигурацию и добавим новый объект конфигурации Регистр сведений.

Для этого выделим в дереве объектов конфигурации ветвь Регистры сведений и нажмем кнопку Добавить.

*- На закладке Основные:* Имя регистра - Цены. Периодичность регистра - В пределах секунды. Представление записи - Цена, а Представление списка - Цены на номенклатуру. Режим записи – Независимый.

*Нажмем «Далее»*

*- На закладку Подсистемы*: отметим в списке следующие разделы: Учет материалов, Оказание услуг и Бухгалтерия.

*Нажмем «Далее»*

*- На закладке Данные:* а) создадим измерение Номенклатура с типом Справочник Ссылка. Номенклатура. Укажем, что это измерение будет Ведущим.

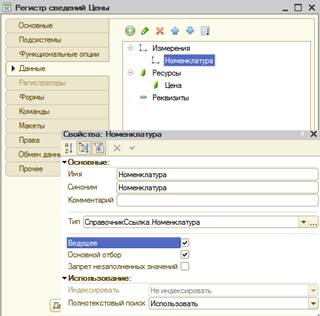


Рисунок – Измерение Номенклатура будет Ведущим

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 выбрана в качестве значения этого измерения в этой записи. При удалении объекта, все записи регистра сведений по этому объекту тоже будут автоматически удалены.

б)  создадим ресурс Цена, тип Число, длина 15, точность 2, неотрицательное. Для этого выделим ветвь Ресурсы к нажмем кнопку Добавить в командной панели окна:

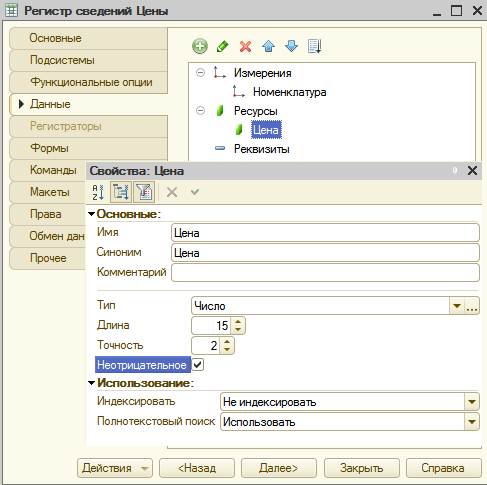


Рисунок – Свойства Цена

В режиме 1С: Предприятие

Теперь запустим 1С: Предприятие в режиме отладки и посмотрим, 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. Создание записей в регистре сведений. Чтобы добавить новую запись в регистр сведений, нажмем кнопку Создать. *Зададим стоимость услуг.*При этом период зададим прошлым числом, так как он должен быть меньше или равен дате создания документа об оказании услуг.

Автоматическая подстановка цены в документ при выборе номенклатуры

Для этого нужно сделать две вещи: Написать некую функцию, которая будет возвращать актуальную цену номенклатуры. Вызвать эту функцию в тот момент, когда в документ добавляется номенклатура, и подставить в документ цену номенклатуры, которую вернет функция. Необходимо данную функцию разместить в общедоступном, для других документов, месте. В режиме Конфигуратор *Создадим функцию Розничная Цена(),которая будет возвращать нам актуальную розничную цену номенклатуры, и поместим ее в общий модуль конфигурации:* Откроем конфигуратор, в ветке Общие > Общие модули добавим новый объект конфигурации Модуль и назовем его Работа. Со Справочниками. Установим флажок Вызов сервера для того, чтобы экспортные процедуры и функции этого модуля можно было вызывать с клиента:

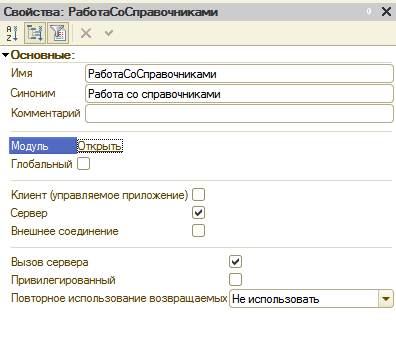


Рисунок – Свойства Работа со справочниками

Затем поместим в модуль следующий текст (см. Листинг 1):

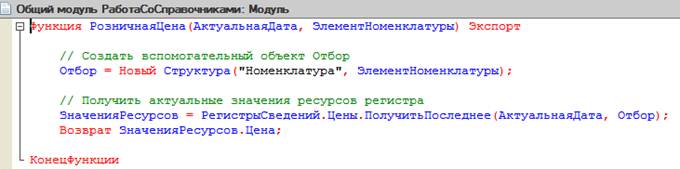


Рисунок – Редактируем Общий Модуль Работа со справочниками

Для получения розничной цены мы будем передавать в функцию два параметра: Актуальная Дата - параметр типа Дата, определяет точку на оси времени, в которой нас интересует значение розничной цены. Элемент Номенклатуры   -  ссылка   на   элемент   справочника Номенклатура, для которого мы хотим получить розничную цену.

*Теперь эту функцию нужно вызвать в некоторый момент работы документа.* Вызов функции при выборе номенклатуры и заполнение цены в документе

1 - Найдем в конфигураторе документ Оказание Услуги и откроем его форму Форма Документа.

2 - Дважды щелкнем на элементе формы Перечень Номенклатуры Номенклатура. Прокрутим список до конца, найдем событие При Изменении, которое возникает после изменения значения поля.

3 - 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. Внесем в него следующий текст (см. Листинг 2):

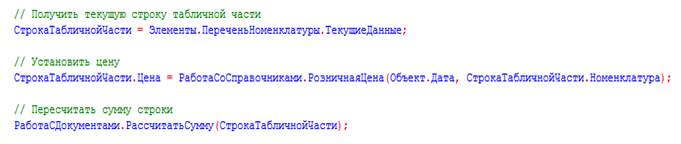


Рисунок – Модуль редактора формы

В режиме 1С: Предприятие

Запустим 1С:Предприятие в режиме отладки и откроем регистр сведений Цены. Для элемента таблицы добавим другим числом новую цену.

Теперь откроем документ Оказание услуги № 1. Оставим дату документа без изменения и повторим выбор тран­зистора в колонке Номенклатура табличной части документа. Автоматически установится последнее значение цены товара.

# Практическая работа № 7 Перечисления

**Цель:** Создание перечислений

**Норма времени**: 1 час

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

В режиме «Конфигуратор»

Откроем конфигуратор и создадим сначала новый объект конфигурации *Перечисление* с именем *Виды Номенклатуры*. На закладке *Данные* добавим два значения перечисления: *Материал* и *Услуга*. Для этого нажмем кнопку. Д*обавить* над списком значений перечисления

**Привязка номенклатуры к значениям перечисления**

**«Вид Номенклатуры»**

Чтобы привязать номенклатуру к значениям перечисления, мы сделаем следующее: в режиме *Конфигуратор* создадим у справочника *Номенклатура* реквизит, который будет хранить значение перечисления; в режиме *1С:Предприятие* проставим нужные значения этого реквизита для всех элементов справочника *Номенклатура*.

**В режиме «Конфигуратор»**

Добавим в справочник *Номенклатура* новый реквизит *Вид Номенклатуры* с типом *Перечисление Ссылка. Виды Номенклатуры*. Для этого откроем окно редактирования объекта конфигурации Справочник *Номенклатура* и на закладке *Данные* нажмем кнопку *Добавить* над списком реквизитов справочника В режиме «1С: Предприятие» После этого запустим «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. В режиме *1С: Предприятие* зададим для каждого элемента справочника *Номенклатура* соответствующее значение реквизита *Вид номенклатуры* Теперь посмотрим, как можно применить новые данные, полученные благодаря использованию перечисления *Виды Номенклатуры*.

**Регистрация расхода только той номенклатуры, которая**

**является материалом**

Если вы помните, на шестом занятии, когда создавались движения документа *Оказание Услуги* по регистру накопления *Остатки Материалов*, мы сказали, что они не совсем правильные, поскольку в регистр будут попадать не только записи об израсходованных материалах, но и записи об оказанных услугах

Процедура Обработка Проведения(Отказ, Режим)

//{{\_\_КОНСТРУКТОР\_ДВИЖЕНИЙ\_РЕГИСТРОВ

// Данный фрагмент построен конструктором.

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

Движения. Остатки Материалов. Записывать = Истина; Для Каждого Тек Строка Перечень Номенклатуры Из Перечень Номенклатуры Цикл // регистр Остатки Материалов Расход Движение = Движения. Остатки Материалов .Добавить(); Движение. Вид Движения = Вид Движения Накопления. Расход; Движение. Период = Дата; Движение. Материал = Тек Строка Перечень Номенклатуры .Номенклатура; Движение. Склад = Склад; Движение. Количество = ТекСтрокаПереченьНоменклатуры.Количество; КонецЦикла;

//}}\_\_КОНСТРУКТОР\_ДВИЖЕНИЙ\_РЕГИСТРОВ

КонецПроцедуры

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 режиме *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 производительности, зато позволит нам получить нужные данные в регистре *Остатки Материалов*. В режиме «Конфигуратор» Скорректируем движения документа, исключив из обработки те строки табличной части, в которых находятся услуги. Для этого откроем в конфигураторе модуль документа *Оказание Услуги* (контекстное меню документа – *Открыть модуль объекта*) и добавим в обработчик события *Обработка Проведения* это условие. Текст следует добавить в начало цикла обхода табличной части документа после строки *Для Каждого ТекСтрокаПереченьНоменклатуры Из Перечень Номенклатуры Цикл*. В результате процедура *Обработка Проведения* должна выглядеть следующим образом

Процедура Обработка Проведения(Отказ, Режим)

//{{\_\_КОНСТРУКТОР\_ДВИЖЕНИЙ\_РЕГИСТРОВ

// Данный фрагмент построен конструктором.

// При повторном использовании конструктора внесенные вручную изменения будут

утеряны!!!

Движения. Остатки Материалов. Записывать = Истина; Для Каждого ТекСтрокаПеречень Номенклатуры Из Перечень Номенклатуры Цикл Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры = Перечисления.ВидыНоменклатуры.Материал Тогда // регистр ОстаткиМатериалов Расход

Движение = Движения. ОстаткиМатериалов .Добавить(); Движение .Вид Движения = Вид Движения Накопления.Расход; Движение. Период = Дата; Движение. Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура; Движение. Склад = Склад; Движение. Количество = ТекСтрокаПереченьНоменклатуры.Количество; Конец Если; Конец Цикла;

//}}\_\_КОНСТРУКТОР\_ДВИЖЕНИЙ\_РЕГИСТРОВ

Конец Процедуры

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

**В режиме «1С:Предприятие»**

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 Philips*; добавим услугу – *Подключение воды*; добавим материал – *Шланг резиновый* Обратите внимание, что цены подставляются автоматически из регистра сведений *Цены*. Нажмем кнопку. П*ровести* 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.

# Практическая работа № 8 Проведение приходной накладной и документа «Оказание услуги» по двум регистрам

**Цель:** Проведение приходной накладной и документа

**Норма времени**: 2 часа

**Методическое обеспечение**: методические указания к практической работе.

В данном случае воспользоваться конструктором движения нецелесообразно, так как процедура проведения уже содержит текст программы, а использование конструктора приведет к тому, что имеющаяся процедура будет удалена.

Откроем документ ПриходнаяНакладная , перейдем на закладку Движение, отметим **РегистрНакопления** ˅СтоимостьМатериалов. Перейдем на закладку **Прочее**и откроем модуль объекта

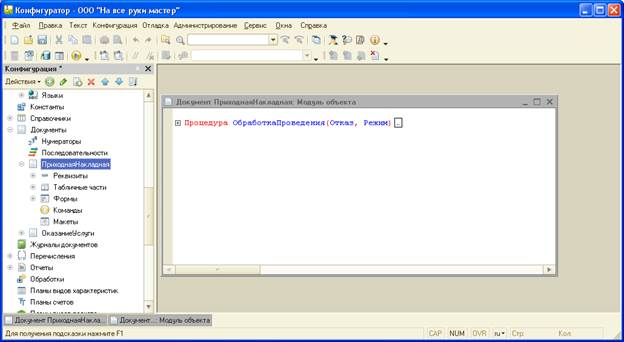


Рисунок – модуль

Процедура Обработка Проведения(Отказ, Режим)

Движения .Остатки Материалов. Записывать = Истина; **Движения. Стоимость Материалов. 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 материаловПриход **Движение = Движения.СтоимостьМатериалов.Добавить(); Движение.ВидДвижения = ВидДвиженияНакопления.Приход; Движение.Период = Дата; Движение. Материал = ТекСтрокаМатериалы.Материал; Движение. Стоимость= ТекСтрокаМатериалы.Сумма;** Конец Цикла;

//}}\_\_КОНСТРУКТОР\_ДВИЖЕНИЙ\_РЕГИСТРОВ

Конец Процедуры

Завершим редактирование модуля формы.

**Настройка командного интерфейса формы документа «Приходная Накладная»**

Перейдем к закладке ФормаДокумента. В левом верхнем окне перейдем на закладку Командный интерфейс. В разделе ПанельНавигации раскрыть группу Перейти и установить флажок для регистра СтоимостьМатериалов свойства Видимость

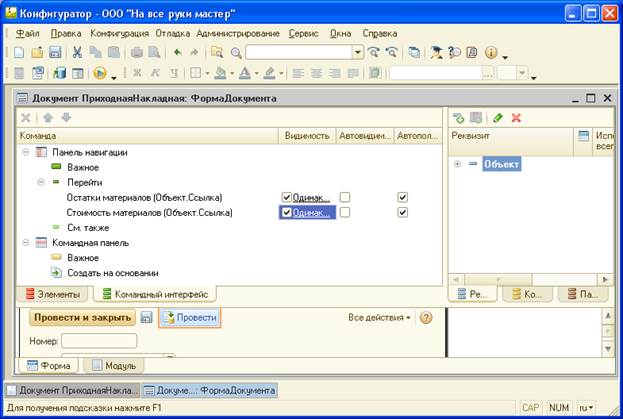


Рисунок – Командный интерфейс

Переходим в режим 1С. Открываем список документов, выполнив команду Приходные накладные в панели навигации Учет материалов. Выделим одновременно все приходные накладные (клавиша CTRL) и выполним команду Все действия-> Провести. Затем откроем первый документ и, выполнив команды перехода к регистрам Остатки Материалов и Стоимость материалов, убедимся, что документ создает движение в обоих регистрах накопления.

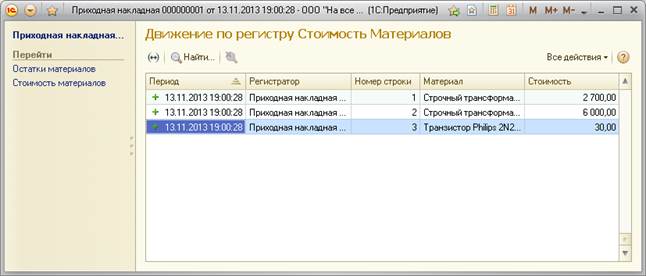


Рисунок – Регистр

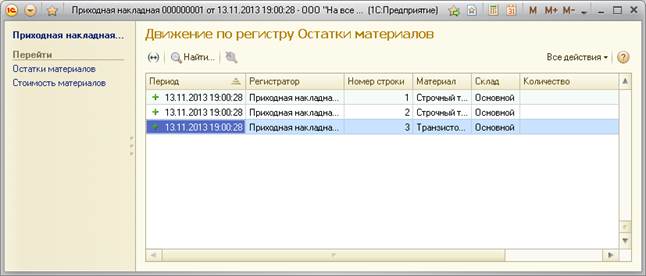


Рисунок – Регистр

3. Проведение документа «Оказание услуги» по двум регистрам

Работаем в режиме Конфигуратор. Откроем

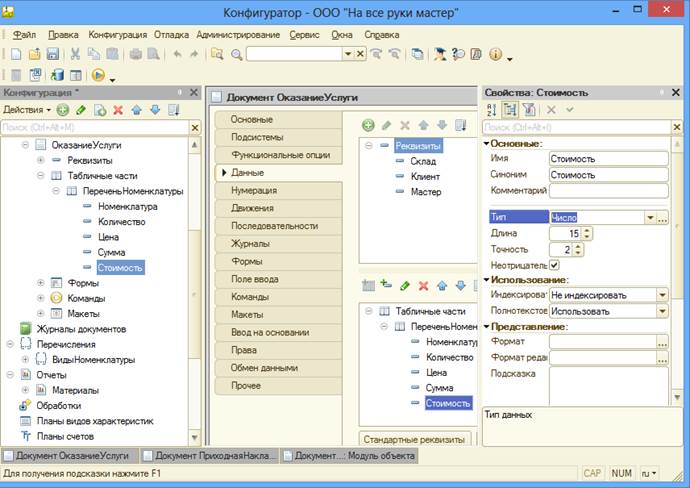


Рисунок - Конфигуратор

После этого откроем форму Форма Документа документа Оказание Услуги и добавим табличную часть Перечень Номенклатуры поле , отображающий новый реквизит Стоимость. Для этого в правом верхнем окне редактора форм на закладке РЕКВИЗИТЫ раскроем реквизит Объект

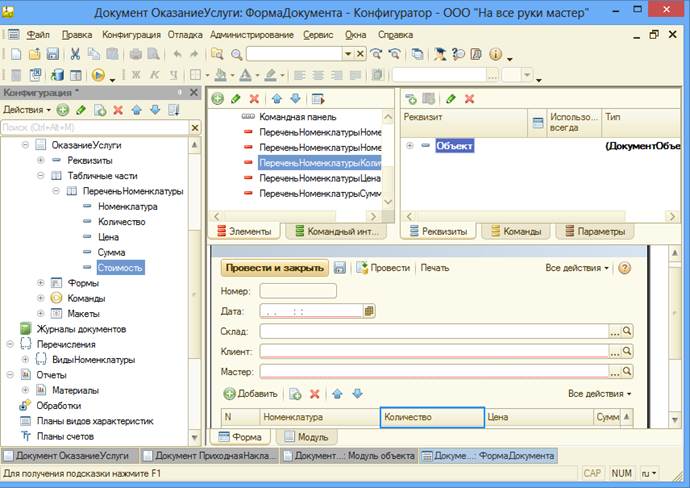


Рисунок – Объект

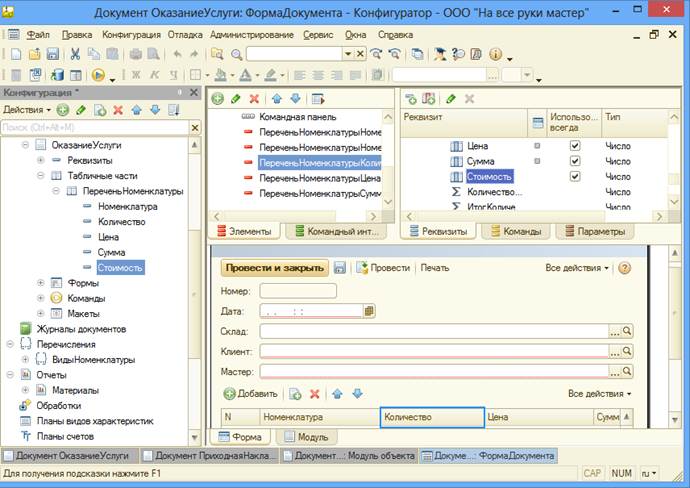


Рисунок –Объект

Найдем в табличной части реквизит Стоимость и с помощью мыши перетащим его в окно элементов формы, расположенное слева в верхней части редактора форм.

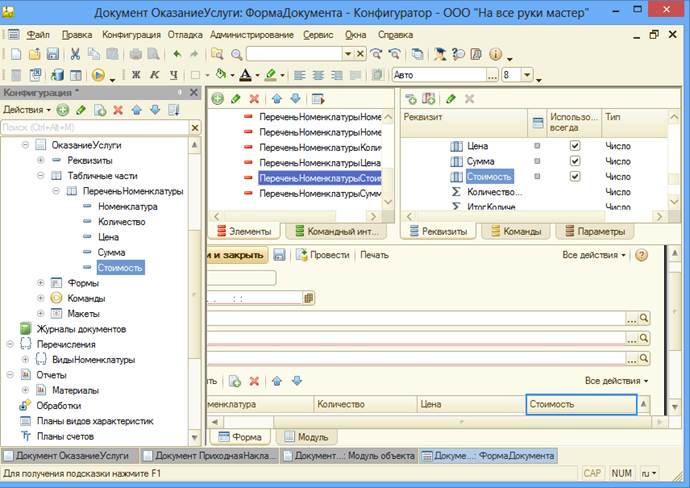


Рисунок – Реквизит Стоимости

  При помощи кнопки управления вверх(синяя стрелка ), переведем этот реквизит на два поля вверх, а на форме это будет на два поля вправо, сразу за номенклатурой

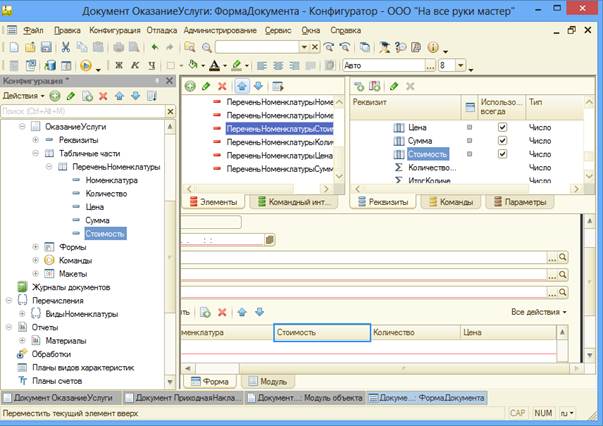


Рисунок – Реквезиты

  Теперь создадим движение документа ОКАЗАНИЕУСЛУГИ таким образом, как мы делали это для документа ПРИХОДНАЯНАКЛАДНАЯ.

В окне редактирования объекта конфигурации Документ перейдем на закладку ДВИЖЕНИЯ. В списке регистров отметим, что документ будет создавать теперь движения по регистру СТОИМОСТЬМатериалов.

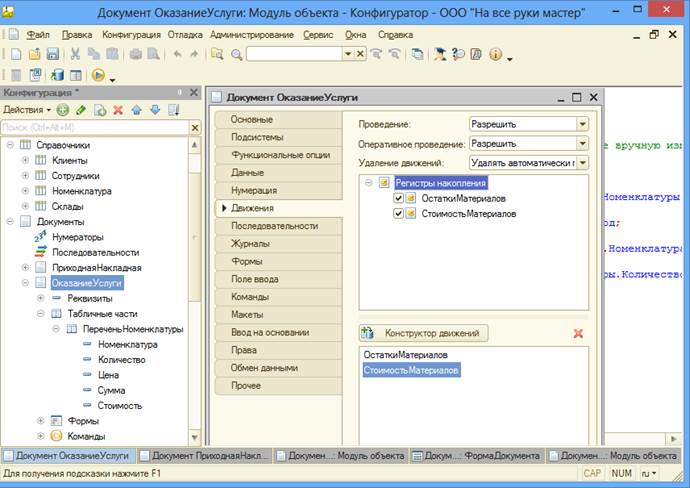


Рисунок – Движение по регистру

Перейдем на закладку ПРОЧЕЕ и откроем модуль объекта. Откроем процедуру обработчика события ОБРАБОТКАПРОВЕДЕНИЯ.

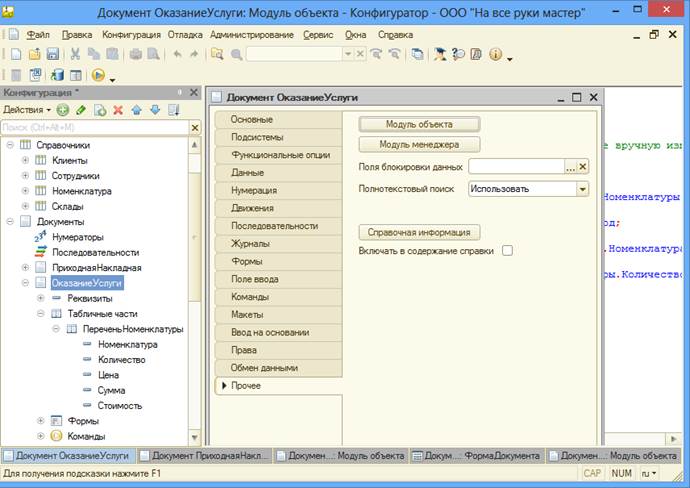


Рисунок – Процедура обработчика

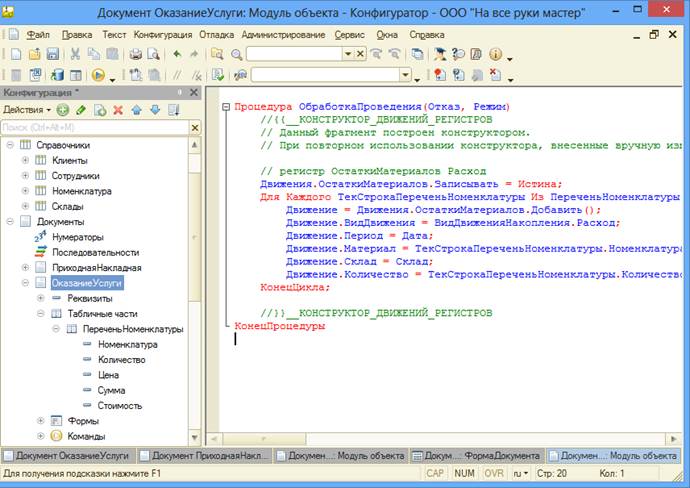


Рисунок – Модуль

Внесем изменения в код процедуры.

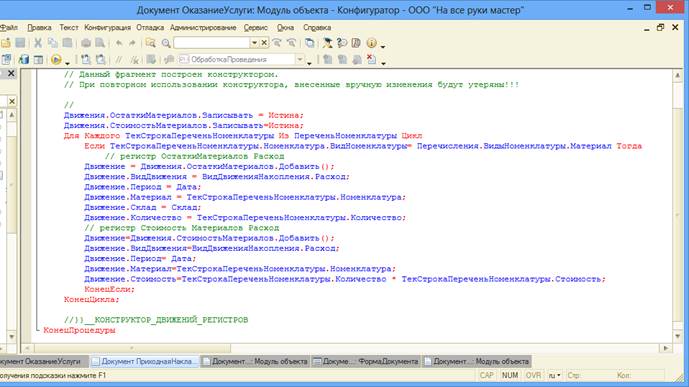


Рисунок – Модуль

Для удобства работы необходимо отредактировать интерфейс.

Перейдем на форму документа, и в левом верхнем углу выберем закладку Командный интерфейс.

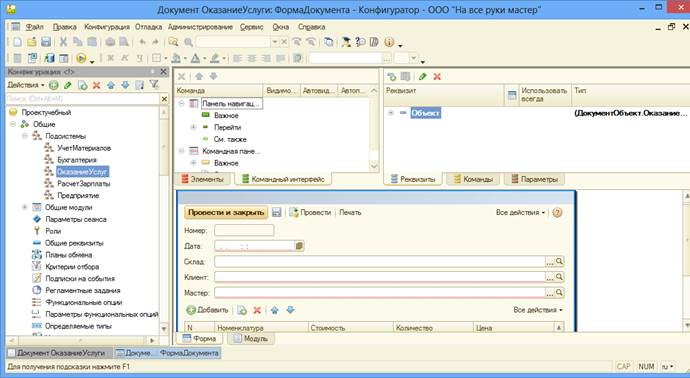


Рисунок - Командный интерфейс

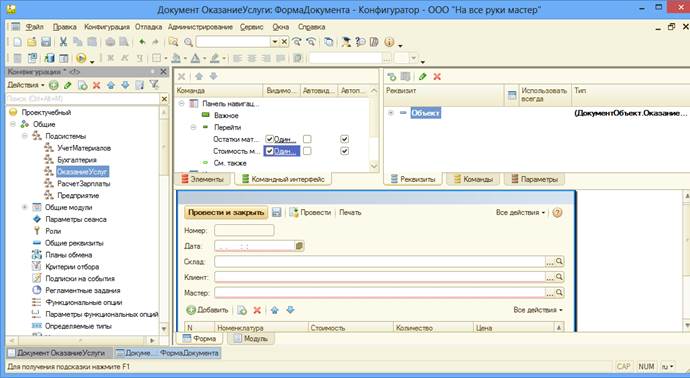


Рис 15 – Командный интерфейс

Сохранить внесенные изменения в конфигурацию. Запустить 1С. Проверить движение регистра Стоимость материалов.

# Практическая работа № 9 Добавление оборотного регистра накопления. Проведение документа «Оказание услуги» по трём регистрам

**Цель:** Добавление оборотного регистра накопления и проведение документа по трем регистрам

**Норма времени**: 10 часов

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

В режиме Конфигуратор

Откроем конфигуратор и создадим новый объект конфигурации Регистр накопления *- На закладке Основные:*имя *-* Продажи; вид регистра – Обороты; расширенное представление списка – Движения документа по регистру Продажи. *- На закладке Подсистемы:* отметим, что этот регистр будет отобра­жаться в подсистемах Бухгалтерия, Учет материалов и Оказание услуг. *- На закладке Данные:*  *создадим измерения регистра:* Номенклатура, тип СправочникСсылка.Номенклатура; Клиент, тип СправочникСсылка.Клиенты; Мастер, тип СправочникСсылка.Сотрудники. Создадим три ресурса Количество, тип Число, длина 15, точность 3; Выручка, тип Число, длина 15, точность 2; Стоимость, тип Число, длина 15, точность 2.

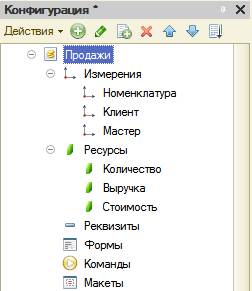


Рисунок - Создание регистров

*- В дереве объектов конфигурации выделим ветвь Подсистемы*, вызовем Все подсистемы: В открывшемся окне выделим подсистему Бухгалтерия; в группе Панель навигации. Обычное включим видимость у команды Продажи и мышью перетащим ее в группу Панель навигации.См.также.

Аналогично, выделив подсистемы Оказание Услуг и Учет Материалов, в панели навигации включим видимость у команды Продажи и перенесем ее в группу См.также. Проведение документа «Оказание услуги» по трем регистрам. Изменим процедуру проведения доку­мента Оказание Услуги, а затем в режиме 1С:Предприятие перепроведем все эти документы. В режиме Конфигуратор откроем окно редактирования объекта конфигурации Документ Оказание Услуги . *- На закладке Движения*: укажем, что этот документ будет создавать движения еще и по регистру Продажи.  *- На закладке Прочее:*  нажмем кнопку Модуль объекта, затем откроем процедуру обработчика события Обработка Проведения и изменим код следующим образом (см. Листинг 12.1): Листинг 12.1

*Процедура Обработка Проведения(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;*

*//РегистрПродажи*

*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*

Отредактируем командный интерфейс формы документа, чтобы в панели навигации формы иметь возможность переходить к списку записей регистра Продажи, связанному с документом:

*- Откроем форму документа ОказаниеУслуги:*перейдем на закладку Командный интерфейс, раскроем группу Перейти и увидим команду для открытия регистра накопления Продажи. *Установим свойство Видимость для этой команды.*

В режиме 1С: Предприятие

В режиме 1С: Предприятие  перепроведем все документы для этого запустим 1С: Предприятие в режиме отладки и откроем по очереди каждый документ Оказание услуги. Проведение документа «Оказание услуги» по трем регистрам . В этом разделе мы сначала изменим процедуру проведения документа *Оказание Услуги*, а затем в режиме *1С:Предприятие* перепроведем все эти документы, чтобы отработал новый, измененный нами алгоритм проведения документов *Оказание услуги*. В режиме «Конфигуратор» откроем окно редактирования объекта конфигурации *Документ Оказание Услуги* и на закладке *Движения* укажем, что этот документ будет создавать движения еще и по регистру *Продажи*. Перейдем на закладку *Прочее* и откроем модуль документа. Для этого нажмем кнопку *Модуль объекта*. Откроем процедуру обработчика события *Обработка Проведения*. В конце цикла после строки *Конец Если* и перед строкой *Конец Цикла* добавим строки кода, создающие движения регистра *Продажи*, производимые документом *Оказание Услуги* // 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; Перед началом цикла установим свойство *Записывать* набора записей движений по этому регистру в значение *Истина*. В результате процедура *ОбработкаПроведения* будет выглядеть следующим образом

Процедура ОбработкаПроведения(Отказ, Режим) Движения.ОстаткиМатериалов.Записывать = Истина; Движения.СтоимостьМатериалов.Записывать = Истина; Движения.Продажи.Записывать = Истина; Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры = Перечисления.ВидыНоменклатуры.Материал Тогда // Регистр ОстаткиМатериалов Расход

Движение = Движения.ОстаткиМатериалов.Добавить(); Движение.ВидДвижения = ВидДвиженияНакопления.Расход; Движение.Период = Дата; Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура; Движение.Склад = Склад; Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество; // Регистр СтоимостьМатериалов Расход Движение = Движения.СтоимостьМатериалов.Добавить(); Движение.ВидДвижения = ВидДвиженияНакопления.Расход; Движение.Период = Дата; Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура; Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Количество \* ТекСтрокаПереченьНоменклатуры.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

Все добавленные конструкции вам уже хорошо известны . Обратите внимание лишь на то, что у оборотного регистра отсутствует Свойство *ВидДвижения*, поскольку отражение вида движения (приход или расход) имеет смысл лишь при учете остатков. В случае регистра оборотов 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. В заключение 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 *Оказание Услуги*. 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.

**В режиме «1С:Предприятие»**

В режиме *1С:Предприятие* нам нужно перепровести все документы оказания услуг и проверить, что они создают правильные движения в регистре *Продажи*. Запустим «1С:Предприятие» в режиме отладки и откроем по очереди каждый документ Оказание услуги. Нажмем Провести и перейдем к списку движений этих документов по регистру Продажи.

# Практическая работа № 10 Выбор данных из одной и двух таблиц. Вывод данных по всем дням в выбранном периоде

**Цель:** Выбор данных из таблиц и вывод данных в выбранном периоде

**Норма времени**: 10 часов

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

Следующий отчет, который мы добавим, будет называться Выручка мастеров. Он будет содержать информацию о том, какая выручка была получена ООО «Фамилия курсанта» благодаря работе каждого из мастеров, с детализацией по всем дням в выбранном периоде и разворотом по клиентам, обслуженным в каждый из дней.

**В режиме Конфигуратор**

Добавим новый объект конфигурации Отчет. Назовем его ВыручкаМастеров и запустим конструктор схемы компоновки данных. Добавим новый **Набор данных - запрос** и вызовем конструктор запроса. В качестве источника данных для запроса выберем виртуальную таблицу регистра накопления **Продажи. Обороты.**

**Запрос для набора данных. Параметры виртуальной таблицы** Зададим один из параметров этой виртуальной таблицы - Периодичность. Для этого перейдем в поле **Таблицы**, выделим таблицу и нажмем кнопку **Параметры виртуальной таблицы** (рис. 15.1). В открывшемся окне параметров зададим значение параметра **Периодичность - День** (рис. 15.2)

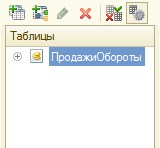


Рисунок – Изменение параметров виртуальной таблицы

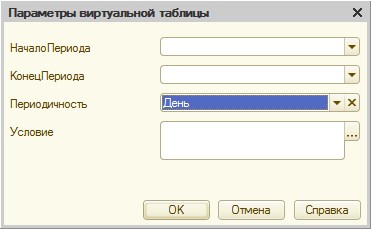


Рисунок – Параметры виртуальной таблицы

Нажмем ОК. После этого выберем из таблицы следующие поля

•Продажи Обороты.Мастер,

•Продажи Обороты.Период,

•Продажи Обороты. Клиент,

•Продажи Обороты. Выручка Оборот.

Теперь перейдем на закладку **Объединения/Псевдонимы** и зададим псевдоним **Выручка** для поля **Продажи Обороты. Выручка Оборот** (рис. 15.3)

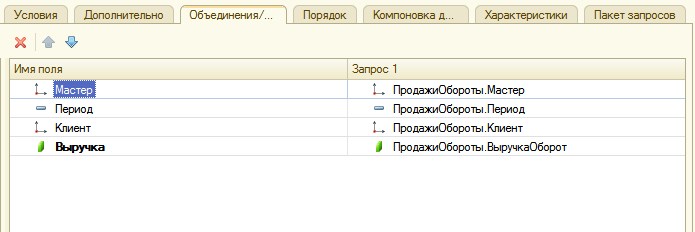


Рисунок – Объединения/Псевдонимы

**Ресурсы**

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

**Параметры**

На закладке **Параметры** выполним те же действия, что и при создании предыдущего отчета. Для параметра **Начало Периода** зададим заголовок **Дата начала**. В поле Тип зададим состав даты - **Дата**. Затем добавим еще один параметр – **Дата Окончания**, установим его тип как **Дата**, состав даты - **Дата**. Для параметра **Конец Периода** зададим выражение (листинг 15.1) и в поле **Ограничение доступности** установим флажок ограничения доступности.

Листинг 15.1. Выражение для расчета значения параметра «Конец Периода»

***Конец Периода(&Дата Окончания,"День")***

В результате перечисленных действий параметры компоновки данных будут иметь следующий вид (рис. 15.4).

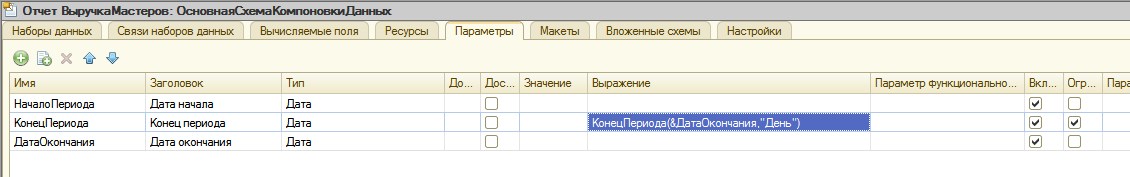


Рисунок – Параметры компоновки данных

**Настройки**

Теперь создадим структуру отчета. На закладке **Настройки** последовательно создадим две вложенные группировки:

•верхнего уровня - по полю Мастер;

•вложенная в нее - по полю Период.

Для этого сначала выделим корневой элемент **Отчет** в структуре отчета, нажмем кнопку **Добавить** в командной панели окна настроек, добавим новую группировку и укажем поле группировки **Мастер** (рис. 15.5).

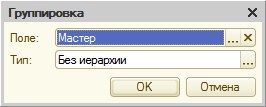


Рисунок – Поле группировки

*>* Затем добавим в группировку **Мастер** вложенную группировку по полю **Период**.  Для этого выделим группировку **Мастер**, нажмем кнопку **Добавить**, добавим новую группировку и укажем поле группировки **Период**. Затем добавим еще одну группировку, вложенную в группировку по полю **Период**, - **Детальные записи**(без указания группировочного поля).  Для этого выделим группировку **Период**, нажмем кнопку **Добавить** и добавим новую группировку без указания группировочного поля. После этого перейдем на закладку **Выбранные поля** и добавим в список выбранных полей поля **Клиент** и **Выручка**. Поля **Мастер** и **Период** мы не задаем, так как по этим полям производится группировка данных и их значение будет выведено автоматически. В результате структура отчета будет иметь вид (рис. 15.6).

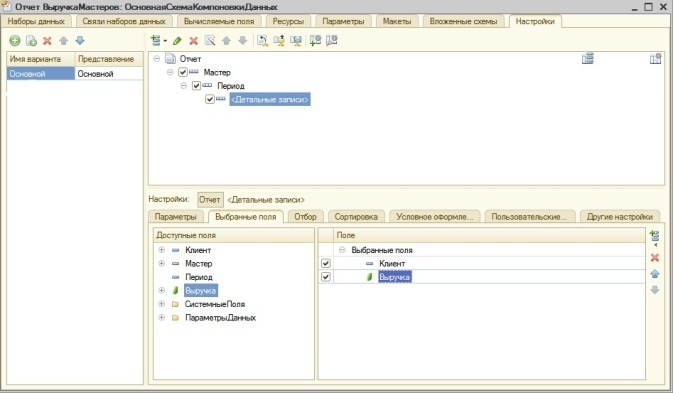


Рисунок – Структура и поля отчета

В заключение перейдем на закладку **Другие настройки** и изменим следующие параметры. Для параметра **Расположение полей группировок** установим значение **Отдельно и только в итогах**.  По умолчанию поля группировок в отчете располагаются вертикально друг под другом. Установка этого свойства в значение **Отдельно и только в итогах** означает, что каждая группировка будет располагаться в отдельной области отчета слева направо и ее наименование будет выводиться только в данной группировке.

Для параметра **Расположение общих итогов по вертикали** зададим значение **Начало**. По умолчанию итоги по вертикали располагаются в конце . Установка этого свойства означает, что общие итоги будут отображаться в начале перед строками группировки. В результате другие настройки отчета примут вид (рис. 15.7).

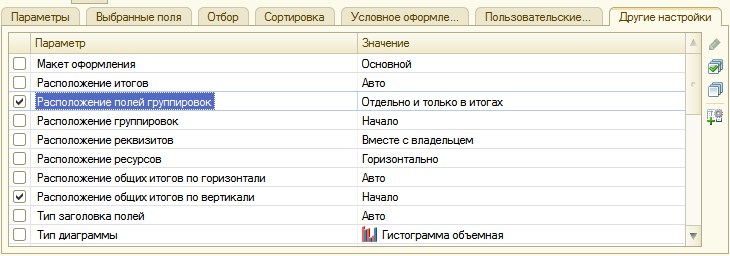


Рисунок – Параметры настроек вывода отчета

Здесь же для параметра **Заголовок** зададим значение **Выручка мастеров**. Затем укажем, что параметры **Дата начала** и **Дата окончания** будут включены в состав пользовательских настроек, и эти настройки будут находиться непосредственно в отчетной форме, то есть будут «быстрыми» настройками. Таким образом, перед формированием отчета пользователь сможет задать отчетный период (рис. 15.8).

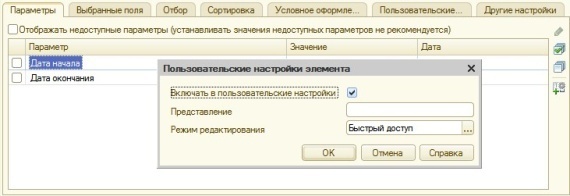


Рисунок – Создание быстрых настроек отчетного периода

В заключение определим, в каких подсистемах будет отображаться наш отчет. Закроем конструктор схемы компоновки данных и в окне редактирования объекта конфигурации Отчет **Выручка Мастеров** перейдем на закладку **Подсистемы**. Отметим в списке подсистем конфигурации подсистемы **Оказание услуг** и **Расчет зарплаты**. Таким образом, ссылка на наш отчет автоматически попадет в панель действий этих подсистем.

**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 **Оказание услуг** и **Расчет зарплаты** в группе команд для выполнения отчетов появилась команда для формирования отчета **Выручка мастеров**.  Выполним эту команду. Зададим отчетный период с 01.07.2009 по 15.07.2009 и сформируем отчет (рис. 15.9).

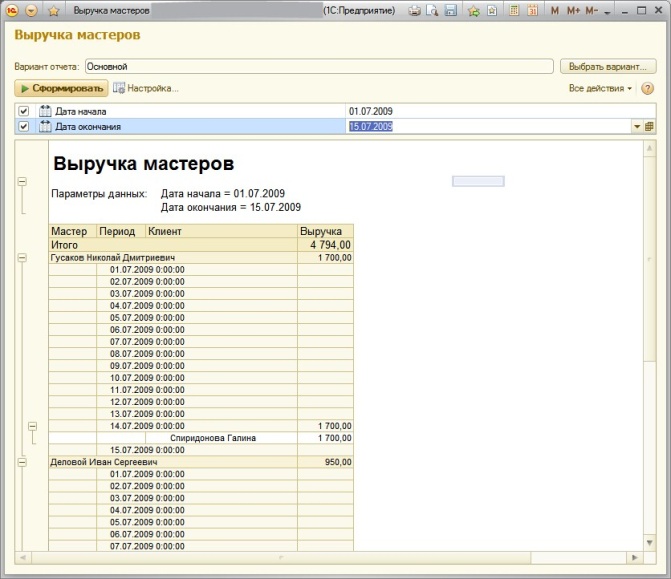


Рисунок – Результат выполнения отчета

**Вывод всех дат в выбранном периоде**

Если вы помните, в начале раздела мы говорили, что этот отчет должен показывать данные с детализацией по всем дням в выбранном периоде.

У нас же отображаются только те дни, для которых существуют ненулевые записи в таблице регистра накопления **Продажи**.

Для детализации данных в отчете система компоновки данных позволяет указывать для группировок дополнение периодов с заданной периодичностью в указанном интервале. Поэтому сейчас мы изменим настройки отчета таким образом, чтобы в отчет попадала каждая дата из периода, за который сформирован отчет.

**В режиме Конфигуратор**

Вернемся в режим Конфигуратор и выполним более тонкую настройку структуры отчета. Откроем схему компоновки данных на закладке **Настройки**. До сих пор все настройки структуры, которые мы выполняли, относились ко всему отчету в целом. Но система компоновки данных позволяет настраивать также и каждый элемент структуры в отдельности.

При установке настроек отчета в средней части окна, под деревом структуры отчета, должна быть выделена кнопка, соответствующая режиму настроек. Кнопка **Отчет** - для настройки отчета в целом или кнопка с именем группировки, например **Детальные записи**, если настройки относятся только к ней.

В нашем случае потребуется изменить настройку группировки **Период**.

Для того чтобы перейти к настройкам именно этой группировки, в поле структуры отчета установим курсор на эту группировку, а затем нажмем кнопку **Период** в командной панели окна. В нижней части окна будут отображены настройки, доступные для данной группировки.

Перейдем на закладку **Поля группировки**. Для поля **Период** установим **Тип дополнения - День** (рис. 15.10).

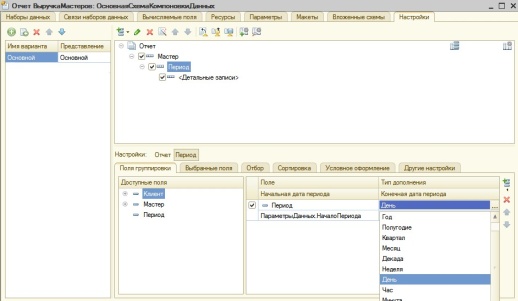


Рисунок – Установка типа дополнения периода

Тем самым мы укажем, что для этой группировки существующие записи с ненулевым значением ресурса будут дополняться записями для каждого из дней.  После этого следует указать, в каком именно периоде будет выполняться такое дополнение.  В поля, расположенные строчкой ниже, можно ввести даты начала и окончания этого периода. Но указание дат в явном виде нас не устраивает, так как пользователь может сформировать отчет за произвольный период. И нам нужно, чтобы дополнение дат выполнялось не в некотором фиксированном периоде, а именно в том периоде, который выбрал пользователь для всего отчета.  Для того чтобы обеспечить именно такую работу отчета, войдем в режим редактирования поля **Начальная дата периода**, дважды кликнув на нем, и нажмем кнопку очистки \*. После этого, нажав кнопку выбора типа данных, мы сможем выбрать тип данных, отображаемых в этом поле.

Выберем **Поле компоновки данных**

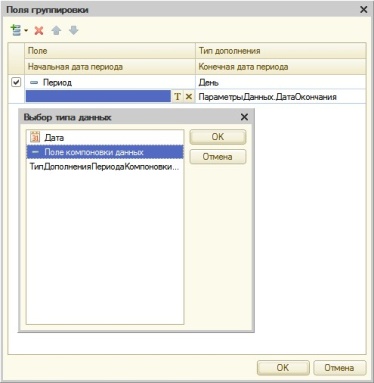


Рисунок –- Выбор типа данных

Нажмем ОК.

Теперь нажмем в поле ввода кнопку выбора и в открывшемся окне выбора поля отметим параметр **Начало Периода**Нажмем ОК.

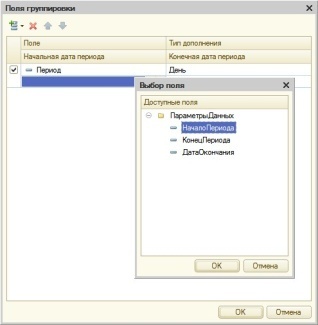


Рисунок – Выбор поля

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

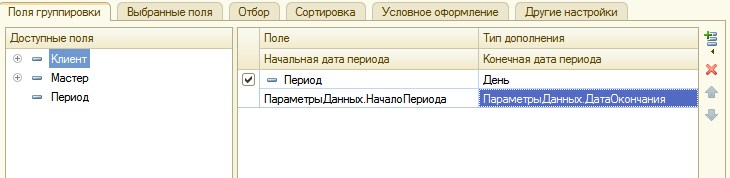


Рисунок –- Настройки группировки "Период"

**В режиме 1С Предприятие**

Запустим 1С:Предприятие в режиме отладки и выполним отчет **Выручка мастеров** за период с 10.07.2009 по 15.07.2009

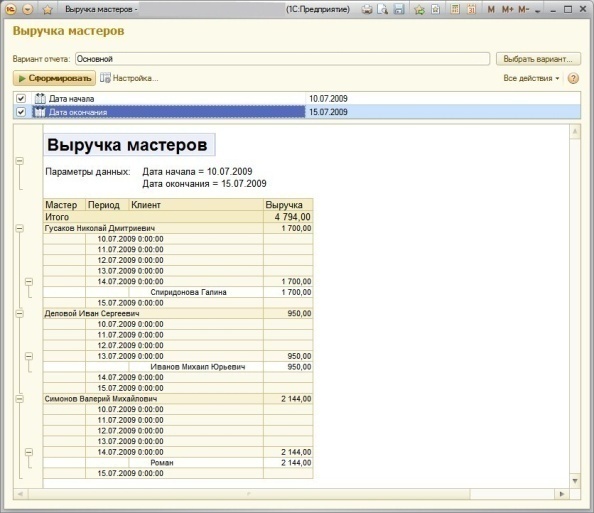


Рисунок – Результат выполнения отчета

**Новый вариант отчета**

Для анализа работы мастеров за определенный период может понадобиться представить ту же информацию в другом, более наглядном виде.

Например, директору при начислении зарплаты, чтобы понять, какой из мастеров лучше работает, вполне может понадобиться увидеть диаграмму, отражающую вклад каждого мастера в общую выручку предприятия за период.

Поэтому мы создадим другой вариант отчета **Выручка Мастеров**, представляющий данные в виде диаграммы.

**В режиме Конфигуратор**

Вернемся в конфигуратор и откроем схему компоновки данных на закладке **Настройки**.  В левой части окна находится список вариантов отчета. При создании настроек отчета в первый раз система компоновки данных по умолчанию создает **Основной** вариант настроек. И мы видим его в списке вариантов нашего отчета.

Чтобы добавить новый вариант, нажмем кнопку **Добавить** над этим списком. Зададим имя варианта – **Объем Выручки**

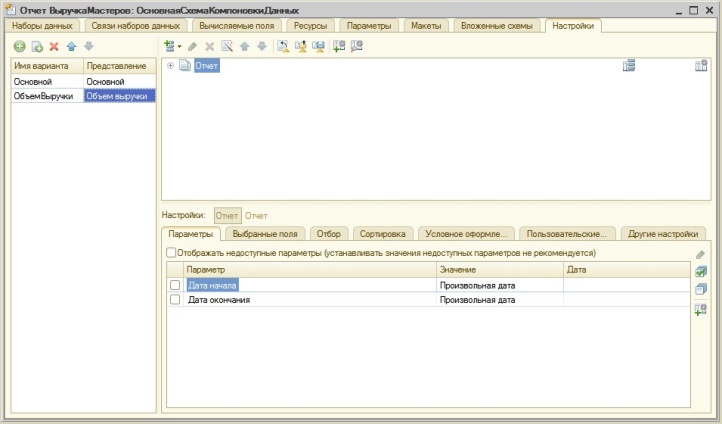


Рисунок – Добавление нового варианта настроек

Мы видим, что структура отчета и все его настройки очистились.

Но они не пропали, а стали невидимы, так как относятся к **Основному** варианту настроек.  Если у отчета есть несколько вариантов, то мы видим и можем изменять настройки того варианта, который выделен в данный момент. Причем вся остальная информация в схеме компоновке данных (ресурсы, параметры, наборы данных) осталась без изменений. Данные для отчета будут получены с помощью того же запроса к базе данных. Изменятся лишь настройки, которые определят, как будет представлен отчет.  Добавим в структуру отчета диаграмму. Для этого выделим корневой элемент **Отчет**, вызовем его контекстное меню и добавим диаграмму. Затем выделим ветку **Точки** и добавим в нее группировку по полю **Мастер**.  **Серии** диаграммы оставим без изменений. Для демонстрации вклада мастеров в общий объем выручки хорошо подойдет измерительная диаграмма, которую мы хотим показать. Для этого вида диаграммы достаточно задать только точки, поэтому серии мы не задаем.  В значения диаграммы всегда выводится один из ресурсов отчета. У нас всего один ресурс - **Выручка** (поле ресурса помечено соответствующей пиктограммой и отличается от обычных полей).  Поэтому перейдем на закладку**Выбранные поля**, перейдем на уровень настроек отчета в целом (нажав кнопку **Отчет**) и выберем поле **Выручка** для вывода в отчет.

Структура отчета должна принять следующий вид (рис. 15.16).

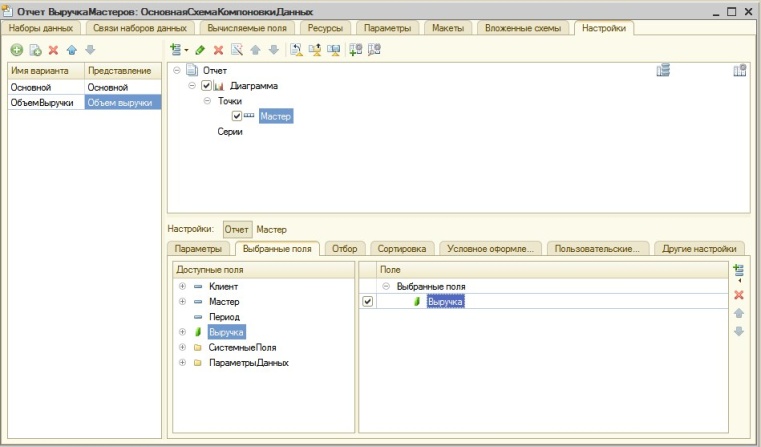


Рисунок – Структура отчета и настройки диаграммы

На закладке **Другие настройки** выберем тип диаграммы - **Измерительная** (рис. 15.17).

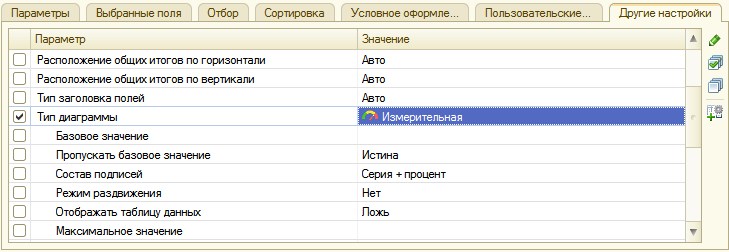


Рисунок – Настройка типа диаграммы

Прокрутив вниз список свойств измерительной диаграммы, зададим ее полосы - **Плохо**, **Хорошо** и **Отлично** (рис. 15.18).

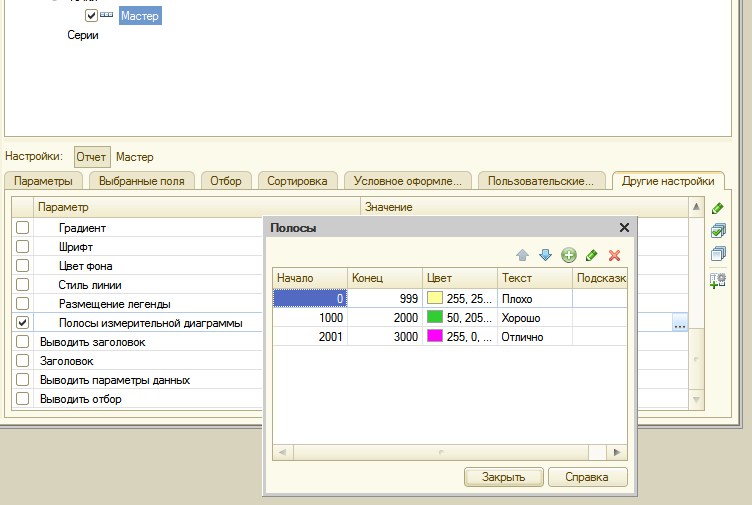


Рисунок – Настройка полос измерительной диаграммы

В заключение включим параметры **Дата начала** и **Дата окончания** в состав пользовательских настроек и установим для них **Режим редактирования** - **Быстрый доступ**.

**В режиме 1С Предприятие**

Запустим 1С:Предприятие в режиме отладки и выполним команду **Выручка мастеров** в панели действий раздела **Расчет зарплаты**. В открывшемся окне отчета нажмем кнопку **Выбрать вариант** (рис. 15.19).

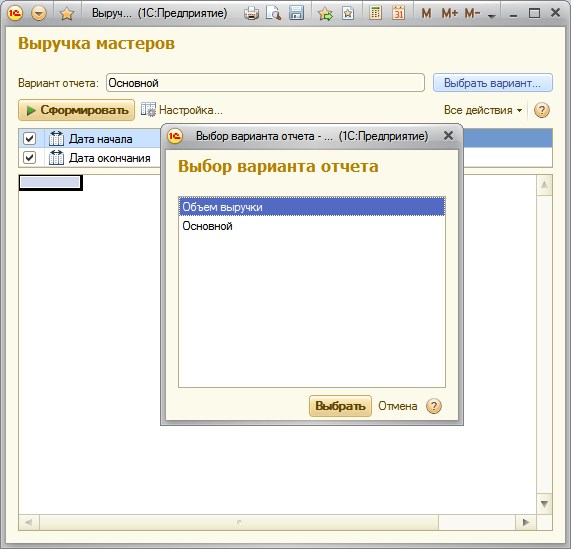


Рисунок – Выбор варианта отчета

В окне вариантов отчета мы видим теперь два варианта - **Основной** и только что созданный нами вариант **Объем выручки**. Выделим его и нажмем кнопку **Выбрать**. Зададим отчетный период с 01.07.2009 по 15.07.2009 и сформируем отчет (рис. 15.20).

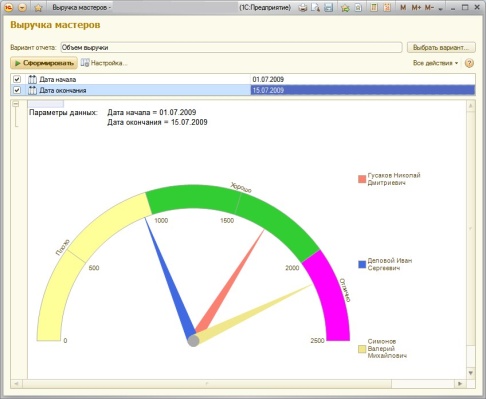


Рисунок – Результат выполнения отчета

В результате мы видим те же данные, что и в основном варианте отчета, представленные в виде измерительной диаграммы. На диаграмме хорошо видна доля каждого мастера в общем объеме выручки. Обратите внимание, что при наведении курсора на сектор диаграммы появляется подсказка.

Если же понадобится просмотреть данные о работе какого-либо мастера с разбивкой по дням и клиентам, достаточно выбрать Основной вариант отчета и переформировать отчет.

Таким образом, на примере отчета Выручка мастеров мы показали создание и использование различных вариантов отчета в целях наилучшего представления информации о работе мастеров.

# Практическая работа № 11 Получение актуальных значений из периодического регистра сведений

**Цель:** Получение актуальных значений

**Норма времени**: 10 часов

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

В режиме Конфигуратор

Создадим настройку отбора в отчете. Для этого в нижней части окна настроек перейдем на закладку Отбор.

Слева мы видим список доступных полей отчета.

Раскроем поле Услуга и двойным щелчком мыши на поле Родитель перенесем его в список условий отбора в правой части окна.

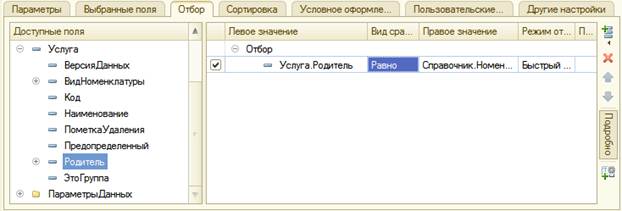


Рисунок – Настройка отбора

Тем самым была создана возможность отбора по группам услуг, которые пользователь может задать в режиме 1С:Предприятие.

В режиме 1С:Предприятие

Откроем отчет в режиме 1С: Предприятие и нажмем кнопку Настройка. В окне пользовательских настроек отчета появились настройки Отбор и Условное оформление, которые мы только что отметили. Настройку Непопулярная услуга мы заранее создали в конфигураторе. Добавив настройку условного оформления, мы предоставили пользователю возможность создавать любое количество собственных условий для условного оформления. Пользователь сможет, при наличии определенной квалификации, задавать многие настройки по своему желанию. Если же такого желания или соответствующих знаний у него нет, лучше задавать эти на стройки жестко, а пользователю останется только включать или выключать их использование. Да собственно часто достаточно только отчетного периода или еще какой-то жизненно важной настройки, и такие настройки, конечно, нужно размещать непосредственно в отчетной форме. Получение актуальных значений из периодического регистра сведений.

Изучим возможность получения последних значений из периодического регистра сведений и с возможность вывода иерархических справочников. В режиме Конфигуратор

Добавим новый объект конфигурации Отчет. Назовем его Перечень Услуг и запустим конструктор схемы компоновки данных. Добавим новый Набор данных - запрос и вызовем конструктор запроса.  Запрос для набора данных. В качестве источника данных для запроса выберем объектную  таблицу справочника Номенклатура и виртуальную таблицу регистра сведений Цены Срез Последних. Для того чтобы исключить неоднозначность имен в запросе, переименуем таблицу Номенклатура в СпрНоменклатура. Для этого выделим ее в списке Таблицы, вызовем ее контекстное меню ив выберем пункт Переименовать таблицу. Параметры виртуальной таблицы. Вызовем диалог ввода параметров виртуальной таблицы Цены Срез Последних и укажем, что период будет передан в параметре Дата Отчета. Для этого выделим эту таблицу в списке Таблицы и нажмем кнопку Параметры виртуальной таблицы. Затем выберем из таблиц следующие поля:

СпрНоменклатура. Родитель,

СпрНоменклатура.Ссылка,

ЦеныСрезПоследних.Цена.

Левое соединение таблиц

*-*На закладке *Связи*: в поле Условие связи, что значение измерения Номенклатура регистра сведений должно быть равно ссылке на элемент справочника Номенклатура. А также снимим флажок Все у таблицы регистра и установим его у таблицы справочника, тем самым установив вид связи как левое соединение для таблицы справочника:

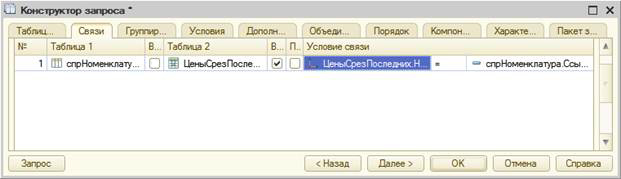


Рисунок – Связь таблиц в запросе

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

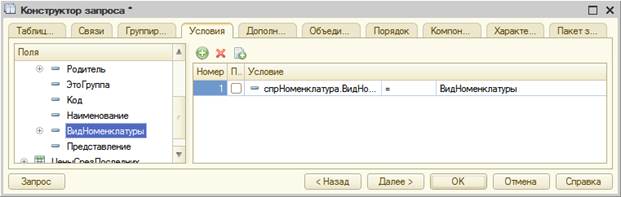


Рисунок – Условия выбора элементов

*-*На закладке *Объединения/Псевдонимы:*указать псевдоним поля Родитель = Группа Услуг, а поля Ссылка = Услуга.

*- НажмемОК–*

После этого, необходимо отредактировать схему компоновки данных, для этого на закладке Ресурсы, нажмем на кнопку добавить и выберем ресурс – Цена *-*На закладке *Параметры* зададим значение параметра Вид Номенкла­туры - Перечисление. Виды Номенклатуры. Услуга.

Кроме этого, снимем ограничение доступности для параметра Дата Отчета. В поле Тип этого параметра зададим состав даты - Дата. Для параметра Период, наоборот, установим ограничение доступ­ности:

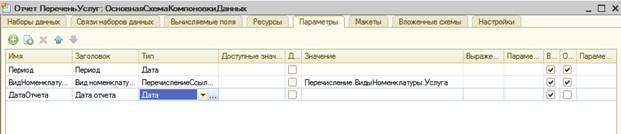


Рисунок – Параметры схемы компоновки

Настройки

*-*Перейдем на закладку *Настройки:* создадим группировку по полю Группа Услуг, указав тип группировки Иерархия. *Существуют следующие типы иерархии для группировок отчета:*

• Без иерархии - в группировке выводятся только неиерархические

записи.

• Иерархия - в группировке выводятся как неиерархические, так

и иерархические записи.

• Только иерархия - в группировке выводятся только иерархические (родительские) записи.

Внутри этой группировки создадим еще одну, без указания группового поля. На подзакладке *Выбранные поля:* укажем поля для вывода Услуга и Цена:

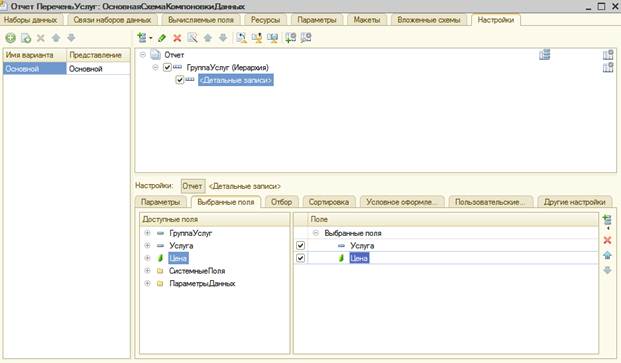


Рисунок – Структура и поля отчета

 - На подзакладке   *Другие* настройки осуществим следующие действия:

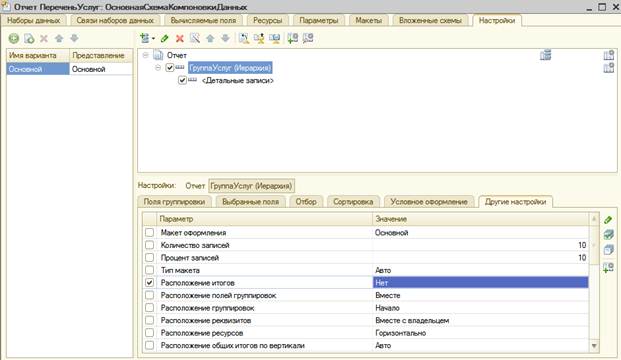


Рисунок – Настройки вывода общих итогов для группировки "Группа Услуг

 - затем:

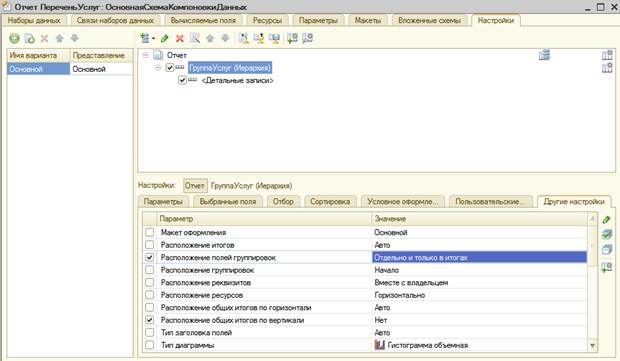
**

Рисунок – Настройки вывода итогов для глобального отчета

Рис. 7 -

В заключение включим параметр Дата отчета в состав пользовательских настроек и установим для него Режим редактирования -Быстрый доступ. Закроем, конструктор схемы компоновки данных и в окне редактирования объекта Перечень Услуг перейдем на закладку Подсистемы. Отметим в списке подсистем конфигурации подсистемы Оказание услуг и Бухгалтерия. В режиме 1С: Предприятие Запустим 1С: Предприятие в режиме отладки и прежде всего, откроем периодический регистр Цены. *После чего протестируем отчет.* На примере этого отчета было изучено, как система компоновки данных получает последние значения из периодического регистра сведений и как выводятся группировки по иерархии справочника.

# Практическая работа № 12 Использование вычисляемого поля в отчёте. Вывод данных в таблицу

**Цель:** Использование вычисляемого поля в отчёте

**Норма времени**: 10 часов

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

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

**В режиме Конфигуратор** Добавим новый объект конфигурации Отчет. Назовем его Рейтинг Клиентов и запустим конструктор схемы компоновки данных. Создадим новый Набор данных - запрос и вызовем конструктор запроса. **Запрос для набора данных в** качестве источника данных для запроса выберем виртуальную таблицу регистра накопления Продажи. Обороты. Затем выберем из нее следующие поля:

Продажи Обороты. Клиент,

Продажи Обороты. Выручка Оборот,

Продажи Обороты. Стоимость Оборот.

 - На закладке *Объединения/Псевдонимы* укажем, что поле Выручка Оборот будет иметь псевдоним Выручка, а поле Стоимость Оборот -Стоимость. *- НажмемОК–* Перейдем к редактированию схемы компоновки данных. **Вычисляемые поля** На этом этапе мы столкнулись с необходимостью отразить в отчете поле, которого нет в наборе данных. Раньше мы использовали в отчете те поля, которые описывались в наборе данных. Теперь, чтобы отобразить доход от оказания услуг в разрезе клиентов, нам необходимо дополнительное поле, рассчитанное как разница между выручкой и стоимостью оказания услуг. Для этого в системе компоновки данных есть возможность опреде­ления вычисляемого поля. **Вычисляемые поля:**  дополнительные поля схемы компоновки данных, значения которых будут вычисляться по некоторой формуле.  - На закладке *Вычисляемые* *поля* добавим вычисляемое поле, нажав кнопку Добавить. Дадим ему имя - Доход, в колонку Выражение введем следующий код: *Листинг 13.2. Выражение для расчета вычисляемого поля «Доход»* ***Выручка – Стоимость* Ресурсы**

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

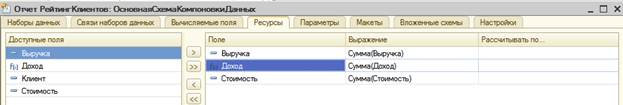


Рисунок – Ресурсы схемы компоновки данных

**Настройки**

*-*На закладке *Настройки*добавим в структуру отчета диаграмму. Для этого нажмем кнопку Добавить в командной панели окна настроек и добавим диаграмму:  Затем выделим ветку Точки и добавим в нее группировку по полю Клиент. Серии диаграммы оставим без изменений.

Дело в том, что для демонстрации рейтинга клиентов хорошо подойдет круговая диаграмма, которую мы хотим показать. Для этого вида диаграммы достаточно задать только точки, поэтому серии мы не задаем.

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

Структура отчета должна принять следующий вид:

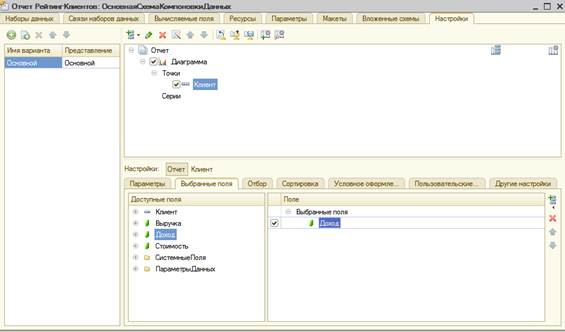


Рисунок – Структура отчета и настройки диаграммы

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

**В режиме 1С: Предприятие**

Запустим 1С:Предприятие в режиме отладки и выполним команду Рейтинг клиентов в панели действий раздела Бухгалтерия. Нажмем Сформировать. Мы видим данные о доходе от оказания услуг по каждому из клиентов, представленные в виде круговой диаграммы. Если вернуться в конфигуратор и изменить тип диаграммы на Гистограмма объемная, то наша диаграмма поменяет вид.

# Практическая работа № 13 Повышение скорости проведения документа. Автоматический расчёт стоимости

**Цель:** Повышение скорости проведение документа

**Норма времени**: 4 часа

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

Повышение скорости проведения

Первое, чем мы займемся на этом занятии, – избавимся от «вредной» конструкции

*ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры*

В режиме «Конфигуратор» откроем модуль документа *ОказаниеУслуги*.

Напомним, как выглядит сейчас процедура проведения этого документа

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;

Другими словами, все данные, необходимые для проведения документа, мы получаем из самого документа, и только для определения того, чем является номенклатура (товаром или услугой), мы обращаемся к базе данных, читая данные всего объекта *Номенклатура* Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры .Забегая вперед, скажем, что это не единственные данные, которые не содержатся в самом документе и которые в то же время будут нужны нам для правильного его проведения Поэтому поступим следующим образом: все данные, связанные с номенклатурой, которая содержится в табличной части документа, мы будем получать с помощью запроса к базе данных. А данные, связанные с самим документом (например, дата документа, склад), мы по-прежнему будем получать из документа. Такой подход позволит нам читать только нужные данные и за счет этого максимально ускорить проведение документа. Итак, запросом мы будем получать: номенклатуру, количество, сумму, стоимость.

Из документа мы возьмем следующие данные: дата, клиент, мастер, склад. Приступим к созданию запроса. Установим курсор перед циклом обхода табличной части документа и из контекстного меню выберем пункт *Конструктор запроса с обработкой результата* Подтвердим, что мы хотим создать новый запрос. В окне конструктора запросов перейдем на закладку *Таблицы и поля* и выберем таблицу *ОказаниеУслугиПереченьНоменклатуры* – это табличная часть документа *ОказаниеУслуги*. Из этой таблицы нам нужны поля – *Номенклатура*, *Номенклатура.ВидНоменклатуры, Количество, Сумма* и *Стоимость* Но нам нужны не все записи этой таблицы, а только те, которые относятся к нашему документу. Поэтому перейдем на закладку *Условия* и зададим условие отбора из таблицы документа только строк проводимого документа. Для этого перетащим поле *Ссылка* в список условий запроса . ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка Ссылка на этот документ будет передана в параметр запроса *Ссылка* . Также следует учесть, что в табличной части документа одна и та же номенклатура может встречаться несколько раз. Поэтому на закладке *Группировка* сгруппируем наши записи по полю *Номенклатура* и *НоменклатураВидНоменклатуры*, а рассчитывать будем сумму значений для полей *Количество* и *Сумма*.

Благодаря этому в результате значения номенклатуры повторяться не будут, и для каждого из них будут посчитаны суммарные значения по полям *Количество* и *Сумма*, если в табличной части документа содержится несколько строк с одинаковой номенклатурой.

Также в состав суммируемых полей включим и поле *Стоимость*. По нему будем рассчитывать, например, функцию *Максимум*. Мы подразумеваем, что для разных строк одной и той же номенклатуры стоимость будет одинаковой, поэтому функция *Максимум* нужна нам лишь для того, чтобы получить одно из имеющихся значений стоимости На закладке *Объединения/Псевдонимы* зададим псевдонимы для полей *Количество* и *Сумма* – *КоличествоВДокументе* и *СуммаВДокументе*, а для поля *НоменклатураВидНоменклатуры* зададим псевдоним *ВидНоменклатуры* просто для облегчения чтения запроса Нажмем *ОК* и посмотрим, какой текст запроса сформировал конструктор

//{{КОНСТРУКТОР\_ЗАПРОСА\_С\_ОБРАБОТКОЙ\_РЕЗУЛЬТАТА

// Данный фрагмент построен конструктором.

// При повторном использовании конструктора внесенные вручную изменения будут

утеряны!!!

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ОказаниеУслугиПереченьНоменклатуры.Номенклатура,

| ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры КАК

ВидНоменклатуры,

| СУММА(ОказаниеУслугиПереченьНоменклатуры.Количество) КАК

КоличествоВДокументе,

| СУММА(ОказаниеУслугиПереченьНоменклатуры.Сумма) КАК СуммаВДокументе,

| МАКСИМУМ(ОказаниеУслугиПереченьНоменклатуры.Стоимость) КАК Стоимость

|ИЗ

| Документ.ОказаниеУслуги.ПереченьНоменклатуры КАК

ОказаниеУслугиПереченьНоменклатуры

|ГДЕ

| ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка

||

СГРУППИРОВАТЬ ПО

| ОказаниеУслугиПереченьНоменклатуры.Номенклатура,

| ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры";

Запрос.УстановитьПараметр("Ссылка", Ссылка);

Результат = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = Результат.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

// Вставить обработку выборки ВыборкаДетальныеЗаписи

КонецЦикла;

//}}КОНСТРУКТОР\_ЗАПРОСА\_С\_ОБРАБОТКОЙ\_РЕЗУЛЬТАТА

Комментарии конструктора запроса в начале и конце фрагмента можно

удалить.

Поскольку для построения запроса мы использовали *Конструктор запроса с обработкой результата*, конструктор написал за нас код для выполнения и обхода записей запроса. Прокомментируем этот код. Как вы уже знаете, для работы с запросами используется объект встроенного языка *Запрос*. Вначале создается новый объект *Запрос* и помещается в

переменную *Запрос*. Затем в свойство *Текст* объекта *Запрос* помещается сам текст запроса (*Запрос.Текст = …*).

После этого устанавливается значение параметра запроса *&Ссылка* как ссылка на тот документ, в модуле которого мы сейчас находимся .

Запрос.УстановитьПараметр("Ссылка", Ссылка); Затем запрос выполняется (*Запрос.Выполнить()*), получается объект

*РезультатЗапроса,* и выполняется его метод *Выбрать()*, который формирует выборку записей из результата запроса.

Таким образом, получается объект *ВыборкаИзРезультатаЗапроса*, который помещается в переменную *ВыборкаДетальныеЗаписи*.

Далее, используя метод этого объекта *Следующий*() (*ВыборкаДетальныеЗаписи.Следующий()*), мы будем в цикле обходить выборку записей запроса. Выполняя метод выборки запроса *ВыборкаДетальныеЗаписи.Следующий()*, мы на каждом шаге цикла позиционируем указатель на следующую запись выборки, пока не будет достигнут конец выборки. Чтобы в цикле получить значение какого-либо поля выборки из результата запроса, мы будем обращаться к полям запроса через точку от переменной *ВыборкаДетальныеЗаписи*, которая содержит текущую строку выборки запроса. Например, так: *ВыборкаДетальныеЗаписи.Номенклатура*. Теперь нам осталось перенести существовавшие ранее в этом модуле строки, описывающие движения регистров, внутрь цикла обхода результата запроса.

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

// Вставить обработку выборки ВыборкаДетальныеЗаписи

КонецЦикла;

Сначала вместо комментария «// Вставить обработку выборки ВыборкаДетальныеЗаписи» перенесем условие проверки и весь код,

формирующий движения по регистрам *ОстаткиМатериалов* и *СтоимостьМатериалов* .

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры =

Перечисления.ВидыНоменклатуры.Материал Тогда

// Регистр ОстаткиМатериалов Расход

Движение = Движения.ОстаткиМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;

Движение.Склад = Склад;

Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;

// Регистр СтоимостьМатериалов Расход

Движение = Движения.СтоимостьМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;

Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Количество \*

ТекСтрокаПереченьНоменклатуры.Стоимость;

КонецЕсли;

КонецЦикла;

В условии заменим *ТекСтрокаПереченьНоменклатуры.Номенклатура* на *ВыборкаДетальныеЗаписи*, так как вид номенклатуры мы теперь получаем из запроса. В движениях также заменим *ТекСтрокаПереченьНоменклатуры* на *ВыборкаДетальныеЗаписи*

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =

Перечисления.ВидыНоменклатуры.Материал Тогда

// Регистр ОстаткиМатериалов Расход

Движение = Движения.ОстаткиМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.Склад = Склад;

Движение.Количество = ВыборкаДетальныеЗаписи.Количество;

// Регистр СтоимостьМатериалов Расход

Движение = Движения.СтоимостьМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.Стоимость = ВыборкаДетальныеЗаписи.Количество \*

ВыборкаДетальныеЗаписи.Стоимость;

КонецЕсли;

КонецЦикла;

Не забудем, что для поля *Количество* мы задали псевдоним в запросе,

поэтому заменим его на *КоличествоВДокументе*

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =

Перечисления.ВидыНоменклатуры.Материал Тогда

// Регистр ОстаткиМатериалов Расход

Движение = Движения.ОстаткиМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.Склад = Склад;

Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;

// Регистр СтоимостьМатериалов Расход

Движение = Движения.СтоимостьМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.Стоимость = ВыборкаДетальныеЗаписи.КоличествоВДокументе \*

ВыборкаДетальныеЗаписи.Стоимость;

КонецЕсли;

КонецЦикла;

Теперь перенесем формирование движений по регистру *Продажи*

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =

Перечисления.ВидыНоменклатуры.Материал Тогда

// Регистр ОстаткиМатериалов Расход

Движение = Движения.ОстаткиМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.Склад = Склад;

Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;

// Регистр СтоимостьМатериалов Расход

Движение = Движения.СтоимостьМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.Стоимость = ВыборкаДетальныеЗаписи.КоличествоВДокументе \*

ВыборкаДетальныеЗаписи.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;

Здесь произведем аналогичные замены. *ТекСтрокаПереченьНоменклатуры*

заменим на *ВыборкаДетальныеЗаписи*

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =

Перечисления.ВидыНоменклатуры.Материал Тогда

// Регистр ОстаткиМатериалов Расход

Движение = Движения.ОстаткиМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.Склад = Склад;

Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;

// Регистр СтоимостьМатериалов Расход

Движение = Движения.СтоимостьМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.Стоимость = ВыборкаДетальныеЗаписи.КоличествоВДокументе \*

ВыборкаДетальныеЗаписи.Стоимость;

КонецЕсли;

// Регистр Продажи

Движение = Движения.Продажи.Добавить();

Движение.Период = Дата;

Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.Клиент = Клиент;

Движение.Мастер = Мастер;

Движение.Количество = ВыборкаДетальныеЗаписи.Количество;

Движение.Выручка = ВыборкаДетальныеЗаписи.Сумма;

Движение.Стоимость = ВыборкаДетальныеЗаписи.Стоимость \*

ВыборкаДетальныеЗаписи.Количество;

КонецЦикла;

Поля запроса *Сумма* и *Количество* заменим на *СуммаВДокументе* и

*КоличествоВДокументе*

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =

Перечисления.ВидыНоменклатуры.Материал Тогда

// Регистр ОстаткиМатериалов Расход

Движение = Движения.ОстаткиМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.Склад = Склад;

Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;

// Регистр СтоимостьМатериалов Расход

Движение = Движения.СтоимостьМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.Стоимость = ВыборкаДетальныеЗаписи.КоличествоВДокументе \*

ВыборкаДетальныеЗаписи.Стоимость;

КонецЕсли;

// Регистр Продажи

Движение = Движения.Продажи.Добавить();

Движение.Период = Дата;

Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.Клиент = Клиент;

Движение.Мастер = Мастер;

Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;

Движение.Выручка = ВыборкаДетальныеЗаписи.СуммаВДокументе;

Движение.Стоимость = ВыборкаДетальныеЗаписи.Стоимость \*

ВыборкаДетальныеЗаписи.КоличествоВДокументе;

КонецЦикла;

Оставшийся цикл обхода табличной части можно удалить

Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл

КонецЦикла;

В результате процедура проведения примет следующий вид

Процедура ОбработкаПроведения(Отказ, Режим)

Движения.ОстаткиМатериалов.Записывать = Истина;

Движения.СтоимостьМатериалов.Записывать = Истина;

Движения.Продажи.Записывать = Истина;

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ОказаниеУслугиПереченьНоменклатуры.Номенклатура,

| ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры КАК

ВидНоменклатуры,

| СУММА(ОказаниеУслугиПереченьНоменклатуры.Количество) КАК

КоличествоВДокументе,

| СУММА(ОказаниеУслугиПереченьНоменклатуры.Сумма) КАК СуммаВДокументе,

| МАКСИМУМ(ОказаниеУслугиПереченьНоменклатуры.Стоимость) КАК Стоимость

|ИЗ

| Документ.ОказаниеУслуги.ПереченьНоменклатуры КАК

ОказаниеУслугиПереченьНоменклатуры

|ГДЕ

| ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка

| |

СГРУППИРОВАТЬ ПО

| ОказаниеУслугиПереченьНоменклатуры.Номенклатура,

| ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры";

Запрос.УстановитьПараметр("Ссылка", Ссылка);

Результат = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = Результат.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =

Перечисления.ВидыНоменклатуры.Материал Тогда

// Регистр ОстаткиМатериалов Расход

Движение = Движения.ОстаткиМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.Склад = Склад;

Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;

// Регистр СтоимостьМатериалов Расход

Движение = Движения.СтоимостьМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.Стоимость = ВыборкаДетальныеЗаписи.КоличествоВДокументе \*

ВыборкаДетальныеЗаписи.Стоимость;

КонецЕсли;

// Регистр Продажи

Движение = Движения.Продажи.Добавить();

Движение.Период = Дата;

Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;

Движение.Клиент = Клиент;

Движение.Мастер = Мастер;

Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;

Движение.Выручка = ВыборкаДетальныеЗаписи.СуммаВДокументе;

Движение.Стоимость = ВыборкаДетальныеЗаписи.Стоимость \*

ВыборкаДетальныеЗаписи.КоличествоВДокументе;

КонецЦикла;

КонецПроцедуры

**В режиме «1С:Предприятие»**

Теперь нужно запустить «1С:Предприятие» в режиме отладки, перепровести документы *Оказание услуги* и проверить, что ничего не изменилось. Таким образом, мы выполнили первый пункт нашего «плана» – избавились в процедуре проведения от считывания всех данных объекта *Номенклатура* и тем самым оптимизировали выполнение процедуры проведения

# Практическая работа № 14 Контроль остатков. Блокировка данных.

**Цель:** Контроль остатков и блокировка данных

**Норма времени**: 4 часа

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

Будем рассматривать принципы работы управляемых блокировок на некоторой тестовой конфигурации. Учет ведется в разрезе номенклатуры и свойств номенклатуры. Приходная накладная (ПН) ничего не блокирует, но пишет данные в регистр накопления "Остаток товара". На примере документа Расходная накладная (РН) изучим, как работают управляемые блокировки, а затем посмотрим на новую схему проведения документов. В ПН и РН есть одинаковые по структуре табличные части (ТЧ) СписокНоменклатуры, в которых содержится список Номенклатуры, Свойств, Количество, Сумма, Цена:

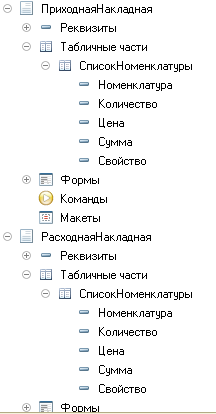


Рисунок - Номенклатура

В Регистре накопления есть соответствующие измерения: Номенклатура, Свойство и два ресурса - Сумма и Количество.

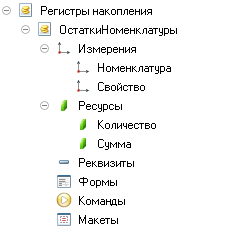


Рисунок - Регистр накопления

Предварительно для работы с управляемыми блокировками установим управляемый режим блокировок для конфигурации в целом и управляемывй режим блокировок отдельно для Приходной накладной, Расходной накладной и для всех регистров, движения по которым эти документы делают. В наших примерах - это один регистр накопления Остаток товара.

**Обратим внимание, что в файловой версии можно устанавливать блокировки только на таблицы целиком** (см. таблицу ниже), построчные (в разрезе выбранных движений из регистра накопления, например) блокировки в файловом режиме не работают. Примеры тестировались на 1С 8.2 + MS SQL 2008 R2 Express.

Примеры рассматриваются на следующих РН и ПН:

**ПН - 01**



Рисунок

**ПН - 02**

http://infostart.ru/upload/iblock/e1f/pn02.png

Рисунок

**РН - 01**

http://infostart.ru/upload/iblock/bc3/rn01.png

Рисунок - **Таблицы**

**Пример 1. Блокировка по конкретно известной номенклатурной позиции.**

**Код процедуры Обработка Проведения(Отказ, Режим) модуля объекта документа Расходна яНакладная:**

Движения.ОстаткиНоменклатуры.Записывать = Истина;

Блокировка = Новый БлокировкаДанных;

ЭлементБлокировки = Блокировка.Добавить();

ЭлементБлокировки.Область = "РегистрНакопления.ОстаткиНоменклатуры";

ЭлементБлокировки.Режим = РежимБлокировкиДанных.Исключительный;

ЭлементБлокировки.УстановитьЗначение("Номенклатура",Справочники.Номенклатура.НайтиПоНаименованию("Ботинки")); //поиск по наименованию в целях примера

Блокировка.Заблокировать(); //установили блокировку по номенклатуре с наименованием "Ботинки"

//Далее идет заполнение движений

**Как это работает:**

Если в процессе проведение РН - 01, сделать попытку провести ПН - 01, то она будет ожидать завершения блокировки регистра накопления по измерению "Номенклатура" с наименованием "Ботинки" и ПН - 02 будет ожидать завершения блокировки регистра накопления по измерению "Номенклатура" с наименованием "Ботинки".

**Пример 2. Блокировка по конкретно известной паре номенклатурная позиция - свойство.**

**Код процедуры Обработка Проведения(Отказ, Режим) модуля объекта документа Расходная Накладная:**

Движения.ОстаткиНоменклатуры.Записывать = Истина;

Блокировка = Новый БлокировкаДанных;

ЭлементБлокировки = Блокировка.Добавить();

ЭлементБлокировки.Область = "РегистрНакопления.ОстаткиНоменклатуры";

ЭлементБлокировки.Режим = РежимБлокировкиДанных.Исключительный;

ЭлементБлокировки.УстановитьЗначение("Номенклатура", Справочники.Номенклатура.НайтиПоНаименованию("Ботинки")); //поиск по наименованию в целях примера

ЭлементБлокировки.УстановитьЗначение("Свойство", Справочники.Свойства.НайтиПоНаименованию("черные 42 размера")); //поиск по наименованию в целях примера

Блокировка.Заблокировать();

//Далее идет заполнение движений

**Как это работает:**

Если в процессе проведение РН - 01, сделать попытку провести ПН - 01, то она будет ожидать завершения блокировки регистра накопления по паре измерений "Номенклатура" (с наименованием "Ботинки") -"Свойство" (с наименованием "черные 42 размера"), а ПН - 02 проведется без ожидания, так как в ее ТЧ Список Номенклатуры нет пары  "Номенклатура"-"Свойство", которая будет образовывать соответствующее движение.

**Пример 3. Блокировка на всю таблицу регистра накопления.**

**Код процедуры ОбработкаПроведения(Отказ, Режим) модуля объекта документа Расходная Накладная:**

Блокировка = Новый БлокировкаДанных;

ЭлементБлокировки = Блокировка.Добавить();

ЭлементБлокировки.Область = "РегистрНакопления.ОстаткиНоменклатуры";

ЭлементБлокировки.Режим = РежимБлокировкиДанных.Исключительный;

Блокировка.Заблокировать();

//Далее идет заполнение движений

**Как это работает:**

Если в процессе проведение РН - 01, сделать попытку провести ПН - 01, то она будет ожидать завершения блокировки таблицы регистра накопления и ПН - 02 будет ожидать завершения блокировки таблицы регистра накопления.

**Пример 4.**Блокировка по одному полю из источника данных для блокировки (это, например, список движений или список номенклатуры по которой будут движения).

**Код процедуры ОбработкаПроведения(Отказ, Режим) модуля объекта документа РасходнаяНакладная:**

Блокировка = Новый БлокировкаДанных;

ЭлементБлокировки = Блокировка.Добавить();

ЭлементБлокировки.Область = "РегистрНакопления.ОстаткиНоменклатуры";

ЭлементБлокировки.Режим = РежимБлокировкиДанных.Исключительный;

ЭлементБлокировки.ИсточникДанных = СписокНоменклатуры;

ЭлементБлокировки.ИспользоватьИзИсточникаДанных("Номенклатура", "Номенклатура");

Блокировка.Заблокировать();

//Далее идет заполнение движений

**Как это работает:**

Если в процессе проведение РН - 01, сделать попытку провести ПН - 01, то она будет ожидать завершения блокировки регистра накопления по измерению "Номенклатура" и значением "Ботинки" из источника Список Номенклатуры и ПН - 02 будет ожидать завершения блокировки регистра накопления по измерению "Номенклатура" и значением "Ботинки" из источника Список Номенклатуры (так как именно такое значение реквизита "Номенклатура" в переданной как источник ТЧ).

**Пример 5.** Блокировка по комбинации полей из источника данных для блокировки (это, например, список движений или список номенклатуры со свойствами по которой будут движения).

**Код процедуры ОбработкаПроведения(Отказ, Режим) модуля объекта документа РасходнаяНакладная:**

Блокировка = Новый БлокировкаДанных;

ЭлементБлокировки = Блокировка.Добавить();

ЭлементБлокировки.Область = "РегистрНакопления.ОстаткиНоменклатуры";

ЭлементБлокировки.Режим = РежимБлокировкиДанных.Исключительный;

ЭлементБлокировки.ИсточникДанных = СписокНоменклатуры;

ЭлементБлокировки.ИспользоватьИзИсточникаДанных("Номенклатура", "Номенклатура");

ЭлементБлокировки.ИспользоватьИзИсточникаДанных("Свойство", "Свойство");

Блокировка.Заблокировать();

 //Далее идет заполнение движений

**Как это работает:**

Если в процессе проведение РН - 01, сделать попытку провести ПН - 01, то она будет ожидать завершения блокировки регистра накопления по паре измерений "Номенклатура" значение "Ботинки" из переданного источника - "Свойство" значение "черные 42 размера" из переданного источника, а ПН - 02 проведется без ожидания, так как в ее ТЧ СписокНоменклатуры нет пары  "Номенклатура"-"Свойство", которая будет образовывать соответвствующее значениям полей источника блокировки движение.

**Пример 6.** Блокировка по двум полям из источника данных для блокировки.

**Код процедуры ОбработкаПроведения(Отказ, Режим) модуля объекта документа РасходнаяНакладная:**

//условие "И" - будут заблокированы и записи для номенклатуры из ТЧ СписокНоменклатуры

// с любыми свойствами и записи для любой номенклатуры со свойствами из ТЧ СписокНоменклатуры

Блокировка = Новый БлокировкаДанных;

ЭлементБлокировки = Блокировка.Добавить();

ЭлементБлокировки.Область = "РегистрНакопления.ОстаткиНоменклатуры";

ЭлементБлокировки.Режим = РежимБлокировкиДанных.Исключительный;

ЭлементБлокировки.ИсточникДанных = СписокНоменклатуры;

ЭлементБлокировки.ИспользоватьИзИсточникаДанных("Номенклатура", "Номенклатура");

ЭлементБлокировки = Блокировка.Добавить();

ЭлементБлокировки.Область = "РегистрНакопления.ОстаткиНоменклатуры";

ЭлементБлокировки.Режим = РежимБлокировкиДанных.Исключительный;

ЭлементБлокировки.ИсточникДанных = СписокНоменклатуры;

ЭлементБлокировки.ИспользоватьИзИсточникаДанных("Свойство", "Свойство");

Блокировка.Заблокировать();

//Далее идет заполнение движений

**Как это работает:**

Если в процессе проведение РН - 01, сделать попытку провести ПН - 01, то она будет ожидать завершения блокировки регистра накопления по измерению "Номенклатура" значение "Ботинки" с любыми свойствами и отдельно по "Свойству" со значением "черные 42 размера" для любой номенклатуры, в ПН - 01 есть и номенклатура "Ботинки" и номенклатура со свойствами "черные 42 размера". Аналогично ПН - 02 будет ожидать завершения блокировки, так как в ПН - 02 есть номенклатура "Ботинки".

**Пример 7.** Новая схема проведения (чем же отличается БлокироватьДляИзменеия = Истина от блокировок при помощи объекта БлокировкаДанных).

**Код процедуры ОбработкаПроведения(Отказ, Режим) модуля объекта документа РасходнаяНакладная:**

//Новая схема (без механизма контроля остатков)

//Разбор БлокироватьДляИзменения = Истина

Движения.ОстаткиНоменклатуры.БлокироватьДляИзменения = Истина;

Движения.ОстаткиНоменклатуры.Записать(); //Набор пустой, нет значений полей пространства блокировки. нечего блокировать.

//очень упрощенный пример создания движений расходной

Для Каждого СтрокаН Из СписокНоменклатуры Цикл

Движение = Движения.ОстаткиНоменклатуры.ДобавитьРасход();

Движение.Период = Дата;

Движение.Номенклатура = СтрокаН.Номенклатура;

Движение.Количество = СтрокаН.Количество;

Движение.Сумма = СтрокаН.Сумма;

Движение.Свойство = СтрокаН.Свойство;

КонецЦикла;

//1

Движения.ОстаткиНоменклатуры.БлокироватьДляИзменения = Истина;

Движения.ОстаткиНоменклатуры.Записать(); //Набор не пустой будет выполнена блокировка, источник - ТЧ СписокТоваров

//Далее должен идти механизм контроля остатков

**Как это работает:**

Если в процессе проведение РН - 01, сделать попытку провести ПН - 01, то она будет ожидать завершения блокировки регистра накопления по паре измерений "Номенклатура" значение "Ботинки" из переданного источника - "Свойство" значение "черные 42 размера" из переданного источника, а ПН - 02 проведется без ожидания, так как в ее ТЧ СписокНоменклатуры нет пары  "Номенклатура"-"Свойство", которая будет образовывать соответвствующее значениям полей источника блокировки движение.

При использовании команды **БлокироватьДляИзменения = Истина** на регистр накопления по значениям измерений движений установится исключительная блокировка. Никаких конструкций с использованием объекта БлокировкаДанных для этого писать не нужно.

Код под комментарием //1 выше аналогичен следующему коду:

//Работает аналогично, как и 1

Движения.ОстаткиНоменклатуры.Записать();

Блокировка = Новый БлокировкаДанных;

ЭлементБлокировки = Блокировка.Добавить();

ЭлементБлокировки.Область = "РегистрНакопления.ОстаткиНоменклатуры";

ЭлементБлокировки.Режим = РежимБлокировкиДанных.Исключительный;

ЭлементБлокировки.ИсточникДанных = Движения.ОстаткиНоменклатуры.Выгрузить(); Для Каждого Измерение Из Метаданные.РегистрыНакопления.ОстаткиНоменклатуры.Измерения Цикл

    ЭлементБлокировки.ИспользоватьИзИсточникаДанных(Измерение.Имя, Измерение.Имя);

КонецЦикла;

Блокировка.Заблокировать();

//Далее должен идти механизм контроля остатков

# Практическая работа № 15 Доработка объектов конфигурации и учётных механизмов

**Цель:** Доработка объектов и учетных механизмов

**Норма времени**: 4 часа

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

Доработка учетных механизмов

Итак, мы добавили возможность указывать произвольные характеристики для номенклатуры и создали несколько таких характеристик – вариантов номенклатуры. Но это лишь часть работы. Теперь хотелось бы иметь возможность еще и учитывать номенклатуру в разрезе этих характеристик. А именно: приходовать товар, указывая характеристики; расходовать товар, указывая характеристики; получать отчеты не просто по номенклатуре, а по номенклатуре с определенными характеристиками. Для этого потребуется доработать имеющиеся регистры и создать новый отчет, который позволит получать данные в разрезе свойств номенклатуры. Регистр «Остатки материалов» В режиме «Конфигуратор». Чтобы обеспечить учет материалов по значениям характеристик, необходимо изменить структуру регистра накопления *ОстаткиМатериалов*, чтобы хранить в нем данные еще и в разрезе наборов свойств номенклатуры. Для этого откроем окно редактирования объекта конфигурации *Регистр накопления ОстаткиМатериалов* и на закладке *Данные* добавим в него новое измерение *НаборСвойств* с типом *СправочникСсылка.ВариантыНоменклатуры* Документ «Приходная накладная» В режиме «Конфигуратор» Теперь нам нужно доработать документ *ПриходнаяНакладная,* чтобы при приходовании материалов можно было указать набор свойств и чтобы это набор свойств записывался в регистры при проведении документа. Для этого откроем окно редактирования объекта конфигурации *Документ ПриходнаяНакладная* и на закладке *Данные* добавим в табличную часть документа новый реквизит *НаборСвойств* с типом *СправочникСсылка.ВариантыНоменклатуры* У этого реквизита необходимо заполнить свойство *Связи параметров выбора,* чтобы после выбора номенклатуры в этом свойстве выбирать только среди тех наборов свойств, которые относятся к данной номенклатуре. Найдем в палитре свойств свойство *Связи параметров выбора* и нажмем кнопку выбора Перенесем из списка доступных реквизитов в список параметров реквизит *Материалы.Материал* Тем самым мы задали, что при выборе в поле *НаборСвойств* будет всегда открываться список элементов справочника *Варианты номенклатуры*, подчиненных материалу, выбранному в колонке *Материал*. После этого расположим этот реквизит в табличной части формы документа. Для этого перейдем на закладку *Формы* и двойным щелчком мыши на строке *ФормаДокумента* в списке форм откроем форму документа. Затем в правом верхнем окне редактора форм на закладке *Реквизиты* раскроем реквизит формы *Объект*. Мы видим, что он содержит все реквизиты документа *ПриходнаяНакладная*. Найдем в табличной части реквизит *НаборСвойств* и с помощью мыши перетащим его в окно элементов формы, расположенное слева в верхней части редактора форм, в таблицу *Материалы*. Новый элемент расположим в структуре элементов формы после поля *Материал* Обратите внимание, что в открывшейся палитре свойств элемента формы *НаборСвойств* в свойстве *ПутьКДанным* уже указан реквизит табличной части *НаборСвойств*, так как мы перетаскивали реквизит в форму с помощью мыши, и оно заполнилось автоматически. Свойство *ПутьКДанным* устанавливает связь элемента формы с реквизитом формы, то есть с отображаемыми данными. Это свойство обязательно должно быть заполнено, иначе элемент формы не будет показан!

В заключение в окне редактирования объекта конфигурации Документ *ПриходнаяНакладная* на закладке *Прочее* откроем модуль объекта. Откроем процедуру обработчика события *ОбработкаПроведения* и добавим к формируемым движениям присвоение значения измерению *НаборСвойств* регистра *ОстаткиМатериалов* «ОбработкаПроведения()»

… // регистр ОстаткиМатериалов Приход

… Движение.Материал =

ТекСтрокаМатериалы.Материал;

Движение.НаборСвойств = ТекСтрокаМатериалы.НаборСвойств;

Движение.Склад = Склад;

Документ «Оказание услуги»

В режиме «Конфигуратор»

Теперь аналогичным образом доработаем документ *ОказаниеУслуги*. Для того чтобы при расходовании материалов пользователь мог указывать набор свойств для каждого расходуемого материала, откроем окно редактирования объекта конфигурации *Документ ОказаниеУслуги* и на закладке *Данные* добавим в табличную часть документа новый реквизит *НаборСвойств* с типом *СправочникСсылка.ВариантыНоменклатуры*. У этого реквизита заполним свойство *Связи параметров выбора*. Перенесем из списка доступных реквизитов в список параметров реквизит *ПереченьНоменклатуры.Номенклатура*. Тем самым мы задали, что при выборе в поле *НаборСвойств* будет всегда открываться список элементов справочника *Варианты номенклатуры*, подчиненных материалу, выбранному в колонке *Номенклатура*. После этого расположим этот реквизит в табличной части формы документа Откроем форму документа и с помощью мыши перетащим его из окна реквизитов формы в окно элементов формы. Новый элемент расположим в структуре элементов формы после поля *Номенклатура*. В заключение в окне редактирования объекта конфигурации Документ *ОказаниеУслуги* на закладке *Прочее* откроем модуль объекта. Откроем процедуру обработчика события *ОбработкаПроведения* и добавим к формируемым движениям присвоение значения измерению *НаборСвойств*

регистра *ОстаткиМатериалов.*

… // регистр ОстаткиМатериалов Расход

… Движение.Материал =

ВыборкаДетальныеЗаписи.Номенклатура;

Движение.НаборСвойств = ВыборкаДетальныеЗаписи.НаборСвойств;

Движение.Склад = Склад;

…

Поскольку на предыдущем занятии мы оптимизировали процедуру проведения документа и получали все данные документа с помощью запроса, то в текст запроса нужно также добавить строки для получения нового реквизита документа.

… Запрос =

Новый Запрос;

// Укажем, какой менеджер временных таблиц использует этот запрос

Запрос.МенеджерВременныхТаблиц = МенеджерВТ;

Запрос.Текст =

"ВЫБРАТЬ

| ОказаниеУслугиПереченьНоменклатуры.Номенклатура,

| ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры КАК

ВидНоменклатуры,

| ОказаниеУслугиПереченьНоменклатуры.НаборСвойств,

| СУММА(ОказаниеУслугиПереченьНоменклатуры.Количество) КАК

КоличествоВДокументе,

| СУММА(ОказаниеУслугиПереченьНоменклатуры.Сумма) КАК СуммаВДокументе

|ПОМЕСТИТЬ НоменклатураДокумента

|ИЗ

| Документ.ОказаниеУслуги.ПереченьНоменклатуры КАК

ОказаниеУслугиПереченьНоменклатуры

|ГДЕ

| ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка

|СГРУППИРОВАТЬ ПО

| ОказаниеУслугиПереченьНоменклатуры.Номенклатура,

| ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры,

| ОказаниеУслугиПереченьНоменклатуры.НаборСвойств";

...

Запрос2 = Новый Запрос;

Запрос2.МенеджерВременныхТаблиц = МенеджерВТ;

Запрос2.Текст = "ВЫБРАТЬ

| НоменклатураДокумента.Номенклатура,

| НоменклатураДокумента.ВидНоменклатуры,

| НоменклатураДокумента.НаборСвойств,

| НоменклатураДокумента.КоличествоВДокументе,

НоменклатураДокумента.СуммаВДокументе,

| ЕСТЬNULL(СтоимостьМатериаловОстатки.СтоимостьОстаток, 0) КАК

Стоимость,

| ЕСТЬNULL(ОстаткиМатериаловОстатки.КоличествоОстаток, 0) КАК

Количество

|ИЗ

| НоменклатураДокумента КАК НоменклатураДокумента

…

Кроме этого, понадобится изменить последний запрос, который при оперативном проведении проверяет, не появились ли отрицательные остатки. Теперь мы будем получать остатки не «вообще» для номенклатуры из табличной части документа, а для номенклатуры именно с тем набором свойств, который указан в строках документа.

… Запрос3.Текст =

"ВЫБРАТЬ

| ОстаткиМатериаловОстатки.Материал,

| ОстаткиМатериаловОстатки.НаборСвойств,

| ОстаткиМатериаловОстатки.КоличествоОстаток

|ИЗ

| РегистрНакопления.ОстаткиМатериалов.Остатки( , (Материал,

НаборСвойств) В

| (ВЫБРАТЬ

| НоменклатураДокумента.Номенклатура,

| НоменклатураДокумента.НаборСвойств

| ИЗ

| НоменклатураДокумента) И Склад = &Склад)

| КАК ОстаткиМатериаловОстатки

|ГДЕ

| ОстаткиМатериаловОстатки.КоличествоОстаток ;

# Практическая работа № 16 Создание отчёта с использованием характеристик

**Цель:** Создание отчета

**Норма времени**: 4 часа

**Методическое обеспечение**: методические указания к практической работе.

**Система Компоновки Данных (СКД) 1С** позволяет создавать программисту массу необходимых отчётов «на лету». Использование этой системы, в отличие от конструктора запроса и непосредственного написания кода отчёта, является явным преимуществом. В этой статье, я хочу показать, как эффективно создавать **отчёты СКД в 1С**, не затрачивая на это много времени.

Начнём с создания внешнего отчёта. В главном меню выбираем Файл ? Новый… В списке щёлкаем на «Внешний отчет».

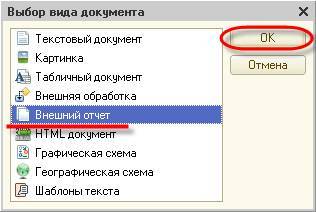
[](https://i1.wp.com/life1c.ru/wp-content/uploads/image0015.jpg)

Рисунок – Выбор документа

В открывшемся окне указываем имя отчёта и синоним.

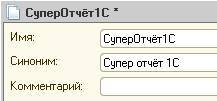
[](https://i1.wp.com/life1c.ru/wp-content/uploads/image0024.jpg)

Рисунок – Свойство

Затем в этом же окне отчёта щёлкаем на кнопку «Открыть схему компоновки данных». Откроется окно создания макета, где нам остаётся только нажать на кнопку «Готово».

[](https://i2.wp.com/life1c.ru/wp-content/uploads/image0034.jpg)

Рисунок

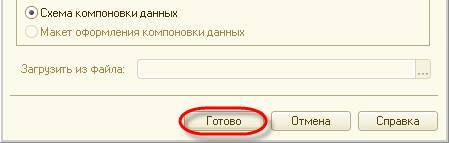
[](https://i2.wp.com/life1c.ru/wp-content/uploads/image0044.jpg)

Рисунок –Создание макета

В окне настройки СКД щёлкаем правой клавишей мыши на «Наборы данных» и выбираем пункт «Добавить наборы данных – запрос».

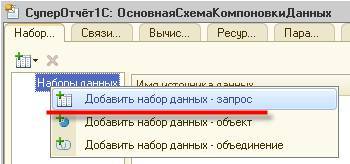
[](https://i0.wp.com/life1c.ru/wp-content/uploads/image0053.jpg)

Рисунок –СКД

Появляется новый набор данных, где нужно прописать выборку данных из текущей информационной базы с помощью запроса. Для создания нового запроса нажимаем на кнопку «Конструктор запроса…» и создаём нужный нам запрос, где выбираем все необходимые для отчёта данные. Описание языка запросов приводится в статье [«Язык запросов – быстро и легко! Конструктор запросов»](http://life1c.ru/post/378).

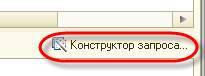
[](https://i0.wp.com/life1c.ru/wp-content/uploads/image0063.jpg)

Рисунок – Конструктор запроса

Предположим, мы решаем следующую несложную задачу. У нас есть регистр накопления «ПартииТоваровНаСкладах», он хранит информацию о партиях номенклатуры, которая находится в местах хранения (на соответствующих складах). Необходимо получить общее количество номенклатуры, которое находится на хранении у соответствующих организаций.

Для решения этой задачи в конструкторе запроса выбираем таблицу базы данных «ПартииТоваровНаСкладах.Остатки». Таблица «Остатки» является виртуальной и предназначена для получения среза на конкретную дату. Подробнее о регистрах накопления можно узнать из этой статьи: [«Регистры накопления в 1С»](http://life1c.ru/post/323).

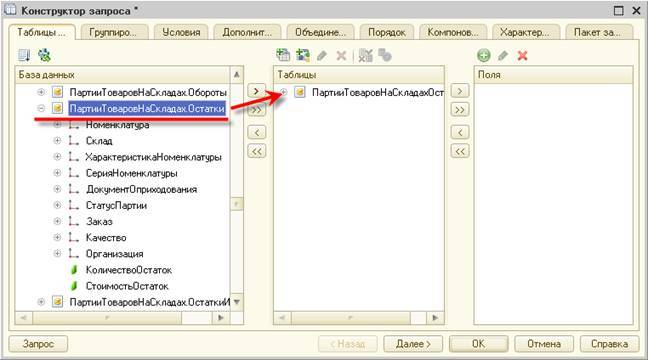
[](https://i0.wp.com/life1c.ru/wp-content/uploads/image0073.jpg)

Рисунок - – Конструктор запроса

Следующим шагом из таблицы остатков выбираем нужные нам в задаче поля, соответственно это: «Номенклатура», «Организация» и «КоличествоОстаток».

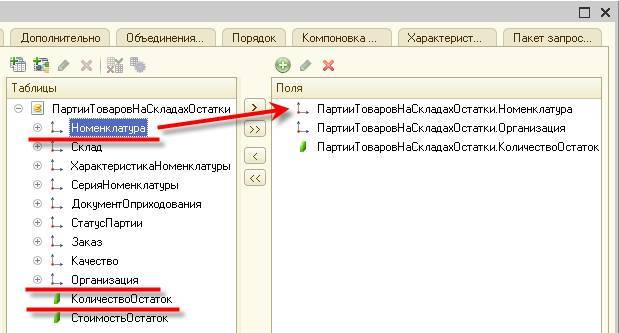
[](https://i1.wp.com/life1c.ru/wp-content/uploads/image0083.jpg)

Рисунок –- Номенклатура

Чтобы задать срез на конкретную дату, открываем параметры виртуальной таблицы. Выделяем выбранную нами таблицу в поле «Таблицы» и щёлкаем вверху на кнопку «Параметры виртуальной таблицы…». В окне параметров вводим параметр даты среза.

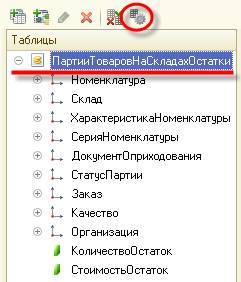
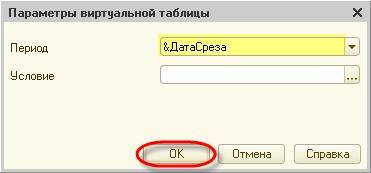
[](https://i2.wp.com/life1c.ru/wp-content/uploads/image0092.jpg)[](https://i2.wp.com/life1c.ru/wp-content/uploads/image0103.jpg)

Рисунок –- Параметры виртуальной таблицы

Всё, задача по построению запроса выполнена. Нажимаем на «ОК» в окне конструктора и переходим к построению структуры **отчёта СКД в 1С**.

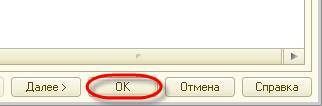
[](https://i2.wp.com/life1c.ru/wp-content/uploads/image0111.jpg)

Рисунок – Конструктор

Текст запроса при этом будет выглядеть так:

ВЫБРАТЬ

ПартииТоваровНаСкладахОстатки.Номенклатура,

ПартииТоваровНаСкладахОстатки.Организация,

ПартииТоваровНаСкладахОстатки.КоличествоОстаток

ИЗ

РегистрНакопления.ПартииТоваровНаСкладах.Остатки(&ДатаСреза,)КАК ПартииТоваровНаСкладахОстатки

Чтобы указать **отчёту СКД** какие поля нужно суммировать для получения общих остатков номенклатуры, в окне настройки СКД переходим на вкладку «Ресурсы» и выбираем поле «КоличествоОстаток» для суммирования.

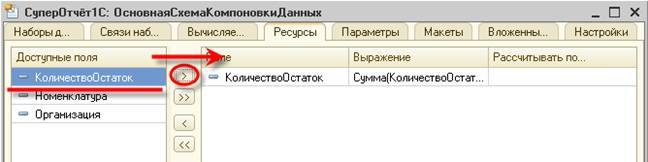
[](https://i1.wp.com/life1c.ru/wp-content/uploads/image0121.jpg)

Рисунок –- Количество остатков

Далее на вкладке «Параметры» указываются значения параметров для запроса. Но здесь мы лишь уберём галочку напротив «ДатаСреза», чтобы значение этого параметра можно было задавать в пользовательском режиме 1С:Предприятие. В колонке «Ограничение доступности» ставим галочку в параметре «Период» и убираем в «ДатаСреза».

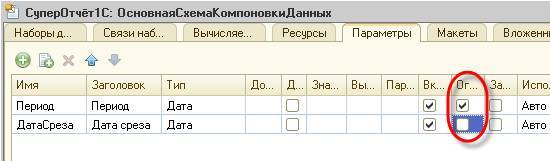


Рисунок – Дата

На вкладке «Настройки» задаём наконец структуру колонок будущего **отчёта СКД**. Для начала ставим галочку напротив параметра «Дата среза», чтобы этот параметр можно было использовать пользователю.

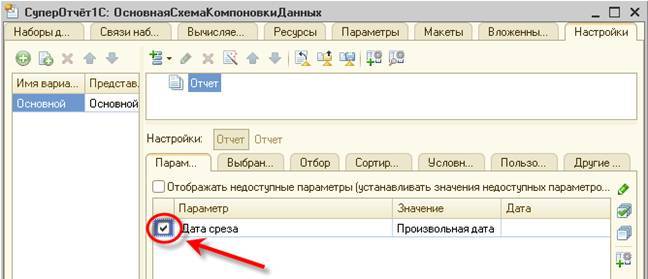
[](https://i0.wp.com/life1c.ru/wp-content/uploads/image014.jpg)

Рисунок – Дата

Затем в поле структуры отчёта СКД щёлкаем правой кнопкой мыши на элемент «Отчет» и выбираем пункт меню «Новая группировка…» (аналогично можно нажать клавишу Insert на клавиатуре).

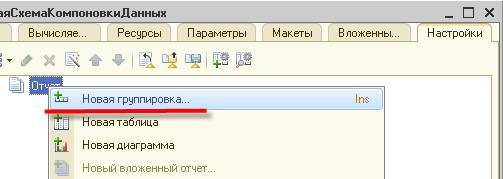
[](https://i2.wp.com/life1c.ru/wp-content/uploads/image015.jpg)

Рисунок – Дата

Нам необходимо построить группировку отчёта вначале по организации, а затем разбить каждую организацию по номенклатурам и для каждой строки отобразить соответствующий остаток. Для этого в открывшемся окне группировки выбираем поле «Организация» и нажимаем «ОК».

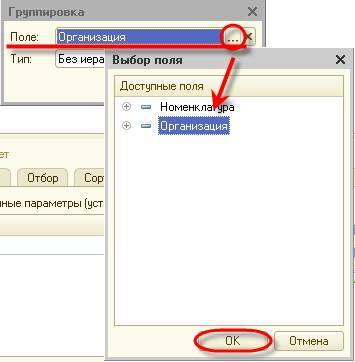
[](https://i1.wp.com/life1c.ru/wp-content/uploads/image016.jpg)

Рисунок – Группировка

Аналогично щёлкаем правой кнопкой по группировке «Организация» и добавляем группировку по полю «Номенклатура». В итоге получаем следующую структуру:

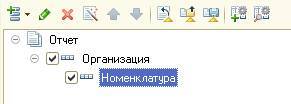
[](https://i0.wp.com/life1c.ru/wp-content/uploads/image017.jpg)

Рисунок – Номенклатура

У элемента «Отчет» выбираем снизу вкладку «Выбранные поля» и перетаскиваем из левого поля ресурс «КоличествоОстаток» в список полей.

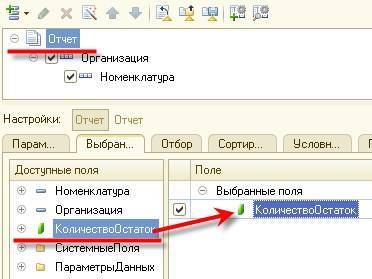
[](https://i0.wp.com/life1c.ru/wp-content/uploads/image018.jpg)

Рисунок - Количество остатков

Закрываем окно настроек **отчёта СКД в 1С.** Отчёт готов для формирования в пользовательском режиме 1С:Предприятие. Для сохранения внешнего отчёта в главном меню выбираем Файл ? Сохранить как… В окне сохранения файла вводим название отчёта и нажимаем кнопку «Сохранить».

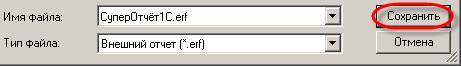
[](https://i0.wp.com/life1c.ru/wp-content/uploads/image019.jpg)

Рисунок – Файл

Теперь смотрим что получилось. Заходим в пользовательский режим 1С:Предприятие, в главном меню выбираем Файл ? Открыть…, выбираем наш созданный отчёт и щёлкаем кнопку «Открыть». Открывается стандартная для **СКД** форма **отчёта 1С**.

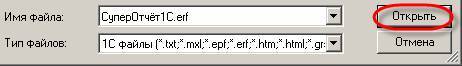
[](https://i1.wp.com/life1c.ru/wp-content/uploads/image020.jpg)

Рисунок – Файл -

В форме самого отчёта перед формированием заполняем значения параметров. Для этого нажимаем на кнопку «Настройки…» и для отчёта устанавливаем значение параметра «Дата среза». Дата среза – это как раз та дата, на которую нужно получить остатки по номенклатуре. Нажимаем «ОК».

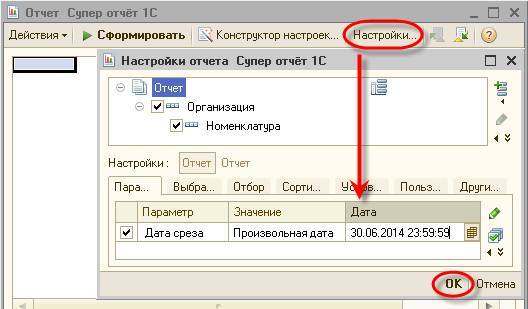
[](https://i2.wp.com/life1c.ru/wp-content/uploads/image021.jpg)

Рисунок – Отчет

В форме отчёта нажимаем кнопку «Сформировать» и получаем готовый отчёт по остаткам с группировкой по организациям и номенклатуре. Отчёт можно свернуть-развернуть по разным организациям, сохранить в формате MSExcel, а также распечатать на принтере.

# Практическая работа № 17 Составление бухгалтерского баланса

**Цель:** Создание бухгалтерского баланса

**Норма времени**: 6 часов

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

Добавление регистра бухгалтерии

В режиме «Конфигуратор» Создадим новый объект конфигурации «Регистр бухгалтерии». Зададим его имя – *Управленческий*. Свойство *Расширенное представление списка* зададим как *Движения в регистре Управленческий*. Укажем, что с ним будет связан план счетов *Основной*. Установим флажок *Корреспонденция* Флажок *Корреспонденция* будет говорить о том, что создаваемый нами регистр поддерживает корреспонденции. Это означает, что каждая запись регистра имеет дебетовую и кредитовую часть, что позволит нам получать информацию не только об остатках и оборотах по счетам, но и о корреспонденциях между счетами.

Регистры, не поддерживающие корреспонденцию, применяются тогда, когда не нужно использовать принцип двойной записи, регламентированный в бухгалтерском учете, и, соответственно, контролировать баланс хозяйственных средств и их источников. Теперь перейдем на закладку *Данные* и создадим два ресурса: *Сумма*, длина *15*, точность *2*, *Балансовый*; *Количество*, длина *15*, точность *3*, *Небалансовый*, признак учета – *Количественный*, признак учета субконто – *Количественный* На этом создание нашего регистра бухгалтерии завершено. Теперь настало время познакомиться с тем, каким образом используется созданный нами регистр бухгалтерии *Управленческий*. Доработка приходной накладной

Что нас ждет дальше?

Сначала мы доработаем оба наши документа (*ПриходнаяНакладная* и *ОказаниеУслуги)* так, чтобы они «поставляли» данные не только для регистров накопления, но и для регистра бухгалтерии. Затем мы создадим бухгалтерский отчет *Оборотно-сальдовая ведомость*, который будет показывать нам состояние товародвижения в ООО «На все руки мастер», основываясь на данных регистра бухгалтерии. При проведении наши документы будут создавать следующие бухгалтерские проводки – записи в регистре бухгалтерии Эти проводки мы отразим при создании движений документов в регистре бухгалтерии. Итак, сначала изменим процедуру проведения документа *ПриходнаяНакладная*, а затем в режиме *1С:Предприятие* перепроведем все эти документы, чтобы отработал новый, измененный нами алгоритм проведения документа *Приходная накладная*.

В режиме «Конфигуратор»

В окне редактирования объекта конфигурации *Документ ПриходнаяНакладная* перейдем на закладку *Движения*. В списке регистров отметим, что документбудет создавать теперь движения и по регистру бухгалтерии *Управленческий*. Перейдем на закладку *Прочее* и откроем модуль объекта. Откроем процедуру обработчика события *ОбработкаПроведения*. В самом конце цикла перед строкой *КонецЦикла* добавим строки кода, создающие движения регистра

*Управленческий*.

// Регистр Управленческий

Движение = Движения.Управленческий.Добавить();

Движение.СчетДт = ПланыСчетов.Основной.Товары;

Движение.СчетКт = ПланыСчетов.Основной.РасчетыСПоставщиками;

Движение.Период = Дата;

Движение.Сумма = ТекСтрокаМатериалы.Сумма;

Движение.КоличествоДт = ТекСтрокаМатериалы.Количество;

Движение.СубконтоДт[ПланыВидовХарактеристик.ВидыСубконто.Материалы] =

ТекСтрокаМатериалы.Материал;

Перед началом цикла установим свойство *Записывать* набора записей регистра *Управленческий* в значение *Истина* для записи изменений регистра в базу данных. В результате процедура *ОбработкаПроведения* будет выглядеть следующим

образом.

Процедура ОбработкаПроведения(Отказ, Режим)

Движения.ОстаткиМатериалов.Записывать = Истина;

Движения.СтоимостьМатериалов.Записывать = Истина;

Движения.Управленческий.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.НаборСвойств = ТекСтрокаМатериалы.НаборСвойств;

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;

// Регистр Управленческий

Движение = Движения.Управленческий.Добавить();

Движение.СчетДт = ПланыСчетов.Основной.Товары;

Движение.СчетКт = ПланыСчетов.Основной.РасчетыСПоставщиками;

Движение.Период = Дата;

Движение.Сумма = ТекСтрокаМатериалы.Сумма;

Движение.КоличествоДт = ТекСтрокаМатериалы.Количество;

Движение.СубконтоДт[ПланыВидовХарактеристик.ВидыСубконто.Материалы] =

ТекСтрокаМатериалы.Материал;

КонецЦикла;

КонецПроцедуры

Прокомментируем этот код.

Как видите, движения для регистра бухгалтерии формируются таким же образом, как и для регистра накопления. Но платформа (при создании таблицы хранения данных) добавила к созданным нами реквизитам регистра еще ряд полей, которые явились следствием использования плана счетов *Основной*. Прежде всего, это поля *СчетДт* и *СчетКт*. В этих полях указываются счета, дебет и кредит которых затрагивает данная проводка. Кроме этого, для измерений и ресурсов регистра, связанных с признаками учета, платформа создает пару полей для хранения значения каждого ресурса отдельно по дебету и отдельно по кредиту проводки – *КоличествоДт* и *КоличествоКт*. А также для счетов, по которым ведется учет в разрезе субконто, платформа создает коллекции *СубконтоДт* и *СубконтоКт*.

Если обратиться к таблице 16.1, то при проведении приходной накладной счетом по дебету должен быть счет *Товары* (41), а счетом по кредиту – счет *РасчетыСПоставщиками* (60). К счету мы обращаемся с помощью свойства глобального контекста

*ПланыСчетов*. Оно предоставляет доступ ко всем планам счетов, созданным в конфигурации. Через точку от него мы указываем имя нужного нам плана счетов – *Основной*. А далее, тоже через точку, указываем имя предопределенного счета в этом плане счетов – *Товары*. Этот счет (и три других) мы создали в конфигураторе. Так как количественный учет у нас ведется только для счета *Товары* (41), то поле регистра *КоличествоДт* заполняется количеством товара из табличной части документа. Поле регистра *КоличествоКт* не заполняется, так как по счету кредита проводки (*РасчетыСПоставщиками*) количественный учет не ведется. Теперь рассмотрим последнюю строку цикла, в которой присваивается значение субконто дебета. Дело в том, что количество субконто на счете дебета и на счете кредита в каждой проводке будет различное, в зависимости от того, как определены счета в используемом плане счетов. Поэтому для каждой записи движения регистра бухгалтерии платформа хранит две коллекции значений: коллекцию субконто дебета и коллекцию субконто кредита. Каждая из них содержит ровно столько элементов, сколько видов субконто указано использовать для соответствующего счета (дебета или кредита) в плане счетов. Обратиться к элементу коллекции можно, указав в квадратных скобках ссылку на соответствующий вид субконто

(*ПланыВидовХарактеристик.ВидыСубконто.Материалы*).

Другой способ – в явном виде указать имя предопределенного вида субконто через точку от коллекции субконто дебета. Другими словами, запись *Движение.СубконтоДт[ПланыВидовХарактеристик.ВидыСубконто.Материалы]* равносильна записи *Движение.СубконтоДт.Материалы*.

Коллекция регистра *СубконтоКт* не заполняется, так как по счету кредита проводки (*РасчетыСПоставщиками*) учет в разрезе субконто у нас не ведется. В заключение отредактируем командный интерфейс формы документа, чтобы в панели навигации формы иметь возможность переходить к списку записей регистра *Управленческий*, связанному с документом. Для этого откроем форму документа *ПриходнаяНакладная*. В левом верхнем окне перейдем на закладку *Командный интерфейс*. В разделе *Панель навигации* раскроем группу *Перейти* и установим видимость для команды открытия регистра бухгалтерии *Управленческий*.

В режиме «1С:Предприятие»

Запустим «1С:Предприятие» в режиме отладки.

Платформа предупредит нас, что регистр бухгалтерии *Управленческий* и справочник *Субконто* не включены ни в одну подсистему. Проигнорируем это сообщение. Откроем документ *Приходная накладная № 1* и нажмем *Провести*. Выполним команду перехода к регистру *Управленческий* и посмотрим, какие движения сформировал документ в регистре бухгалтерии. Обратите внимание: поскольку на счете 60 (*РасчетыСПоставщиками*) отсутствует аналитика и ведется только суммовой учет, в записях движений регистра СубконтоКт1, СубконтоКт2 и КоличествоКт не указаны. После этого перепроведем документ Приходная накладная № 2 и убедимся, что он тоже формирует правильные проводки по регистру бухгалтерии Управленческий. .Теперь перейдем к более сложной задаче: добавлению движений по регистру Управленческий в документ ОказаниеУслуги.

# Практическая работа № 18 Корреспонденция счетов. Проводки

**Цель:** Корреспонденция счетов

**Норма времени**: 6 часов

**Методическое обеспечение**: методические указания к практической работе.

В программе "1С Бухгалтерия 8” реализована возможность формирования списка корректных корреспонденций счетов. Данные этого регистра впоследствии можно использовать для контроля корректности корреспонденций.

Например, вы можете определить, что в вашей организации все расчеты с поставщиками ведутся исключительно на счете 60.01 "Расчеты с поставщиками и подрядчиками”. В этом случае проводка, отражающая перечисление денежных средств поставщику с расчетного счета, будет выглядеть так: Дт 60.01 "Расчеты с поставщиками и подрядчиками" Кт 51 "Расчетные счета". Эту проводку рекомендуется внести в список корректных корреспонденций: тогда при проверке бухгалтерских проводок на предмет их корректности (о том, как это делать, мы поговорим позже) программа выявит все некорректные проводки. И если у вас расчет с поставщиком отражен, например, проводкой Дт 60.02 "Расчеты по авансам выданным" Кт 51 "Расчетные счета", то программа уведомит вас о несоответствии данной проводки списку корректных корреспонденций и предложит либо отредактировать проводку, либо добавить ее в этот список.

Чтобы перейти в режим работы со списком корректных корреспонденций счетов, нужно выполнить команду главного меню Операции | Регистры сведений и в открывшемся окне выделить щелчком мыши позицию Корректные корреспонденции счетов бухгалтерского учета.

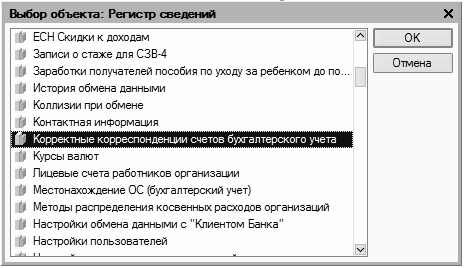


Рисунок – Регистр сведений

Выбор регистра сведений с корректными корреспонденциями счетов. После нажатия в данном окне кнопки ОК на экране отобразится окно Корректные корреспонденции счетов бухгалтерского учета.

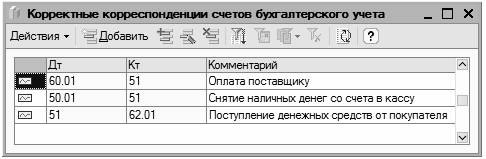


Рисунок – Корректные корреспонденции

Список корректных корреспонденций счетов

В этом окне представлен перечень сформированных ранее корректных корреспонденций (если ранее корректные корреспонденции не составлялись, данный список будет пустым). Для каждой позиции отображается счет по дебету, счет по кредиту и комментарий, кратко раскрывающий суть корреспонденции. Чтобы добавить в список корректных корреспонденций новую позицию нажмите в инструментальной панели окна кнопку Добавить или клавишу <Insert>.

Чтобы отредактировать имеющуюся запись выделите ее щелчком мыши и нажмите клавишу <F2> или выполните команду Действия | Изменить.

Ввод новых и редактирование имеющихся корреспонденций осуществляется в окне.

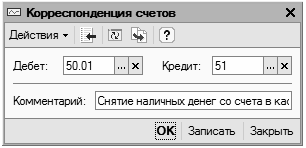


Рисунок – Корреспонденция счетов

Ввод и редактирование корректных корреспонденций

В этом окне в полях Дебет и Кредит следует указать, соответственно, счет по дебету и счет по кредиту. Для этого нужно нажать кнопку выбора и в открывшемся окне выбрать требуемый счет. В поле Комментарий с клавиатуры вводится произвольный текст к данной корреспонденции (это не обязательно, но желательно, чтобы сразу был понятен смысл корреспонденции). Как уже отмечалось ранее, в плане счетов предусмотрены некоторые ограничения по использованию счетов в проводках. В частности не допускается включать в проводки родительские счета, имеющие субсчета. Однако к регистру корректных корреспонденций это не относится: здесь вы можете формировать абсолютно любые корреспонденции безо всяких ограничений. Например, вы можете указать только счет по дебету – и в этом случае все проводки, в которых дебетуется данный счет, будут автоматически считаться корректными. Или вы можете указать только родительский счет, без конкретики по субсчету – в этом случае программа выдаст запрос.



Рисунок – Запрос

Запрос на добавление в корреспонденцию субсчетов

Если на этот запрос ответить отрицательно, то корреспонденция будет сформирована именно в таком виде – без субсчетов. Если же ответить утвердительно, то на экране отобразится окно, пример которого показан на рис. 3.7. В данном случае изначально формировалась корреспонденция Дт 60 "Расчеты с поставщиками и подрядчиками" Кт 52 "Валютные счета", но поскольку у счета 60 есть субсчета, то программа и выдала этот запрос.

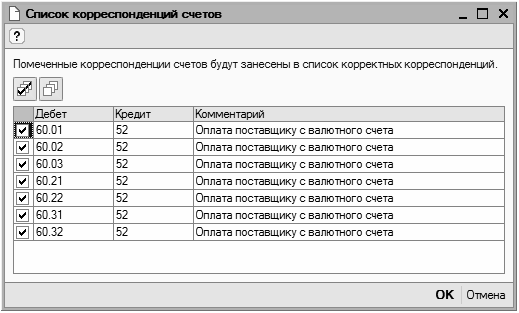


Рисунок – Список корреспонденции счетов

Выбор корректных корреспонденций для добавления в регистр

Здесь путем установки соответствующих флажков можно выбрать позиции, которые будут занесены в список корректных корреспонденций. Если установлены все флажки, то на каждый субсчет будет сформирована отдельная запись в регистре корректных корреспонденций. Если снять все флажки, то в регистр не добавится ни одна запись. А чтобы в него попала запись только с родительским счетом, нужно нажать кнопку Нет при появлении запроса. Чтобы удалить запись из регистра корректных корреспонденций, нужно выделить ее в списке щелчком мыши и выполнить команду контекстного меню Удалить или нажать клавишу <Delete>.

Использование регистра корректных корреспонденций счетов позволяет дополнительно систематизировать процесс формирования бухгалтерских проводок, что в конечном итоге позволяет избежать путаницы в учете.

Журнал проводок – это регистр, в котором содержатся все сформированные в программе бухгалтерские проводки. Он является довольно удобным механизмом для просмотра, поиска, редактирования проводок, а также для вывода списка проводок на печать.

В программе ”1С Бухгалтерия 8” ведутся отдельные журналы проводок для бухгалтерского и налогового учета. Поскольку порядок работы с ними практически идентичен, мы рассмотрим его на примере журнала проводок бухгалтерского учета.

Чтобы открыть журнал проводок, нужно выполнить команду главного меню Операции | Журнал проводок (бухгалтерский учет) либо в панели функций на вкладке Предприятие щелкнуть мышью на ссылке Журнал проводок (бухгалтерский учет). В результате на экране отобразится окно.

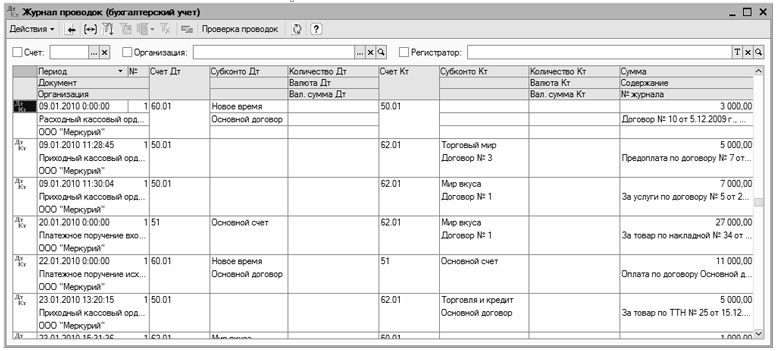


Рисунок – Журнал проводок

Журнал бухгалтерских проводок

В окне отображается перечень всех имеющихся в программе проводок независимо от того, где и как они были сформированы: в режиме ввода/редактирования документов, вручную, автоматически и т. д. Для каждой проводки показаны данные, по которым ее можно идентифицировать: дата формирования проводки, реквизиты и название документа (на основании которого она сформирована), название организации, корреспонденция счетов, аналитика, сумма и др.

В верхней части окна содержатся параметры, с помощью которых вы можете применить фильтр на имеющиеся в списке бухгалтерские проводки. Эту возможность особенно удобно использовать при работе с большими объемами данных, когда среди всего обилия проводок необходимо найти проводки только по какому?то одному счету, организации или типу данных.

Чтобы в списке отображались проводки только по какому-то одному счету, нужно в верхней части окна установить флажок возле поля Счет, после чего в этом поле нажать кнопку выбора и в открывшемся окне справочника счетов указать требуемый счет. Для применения фильтра по организации установите флажок возле поля Организация, после чего из раскрывающегося списка (его содержимое формируется в справочнике организаций) выберите нужную организацию. Чтобы применить фильтр по типу данных, нужно установить флажок возле поля Регистратор, после чего нажать кнопку выбора и указать тип данных (например, Платежное поручение), а затем – конкретный документ.

При необходимости вы можете настроить более тонкий фильтр на отображаемые данные. Для перехода в соответствующий режим используйте команду Действия | Установить отбор и сортировку списка. О том, как использовать данный механизм, было сказано на [уроке № 7](http://1cbugalteriay.ucoz.ru/index/urok_7_opisanie_osnovnykh_polzovatelskikh_interfejsov_1c/0-11) "Описание основных пользовательских интерфейсов".

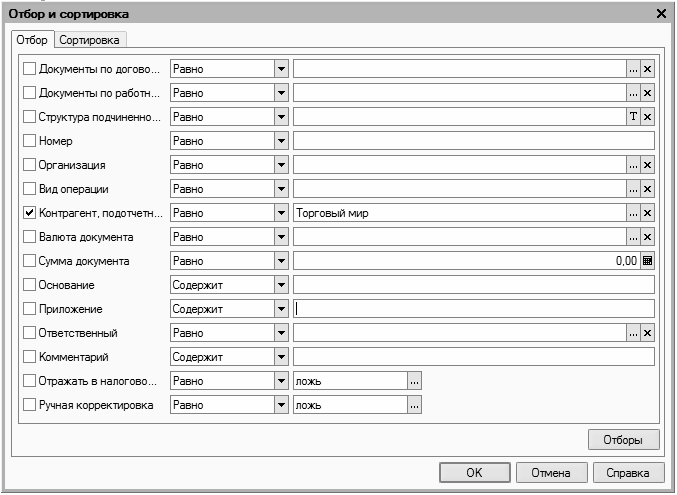


Рисунок – Отбор и сортировка

Полезный совет

Из журнала проводок вы можете быстро перейти в режим просмотра и редактирования документа, на основании которого сформирована любая бухгалтерская проводка. Для этого достаточно дважды щелкнуть мышью на этой проводке.

В журнале проводок можно выполнить быструю проверку всех бухгалтерских проводок на предмет соответствия списку корректных корреспонденций. Для этого нужно в инструментальной панели нажать кнопку Проверка проводок либо выполнить команду Действия | Проверка проводок.

Если в ходе проверки будут обнаружены проводки с некорректными корреспонденциями счетов, то на экране появится окно, а в нижней части интерфейса отобразится окно служебных сообщений с информацией об обнаружении некорректных корреспонденций.

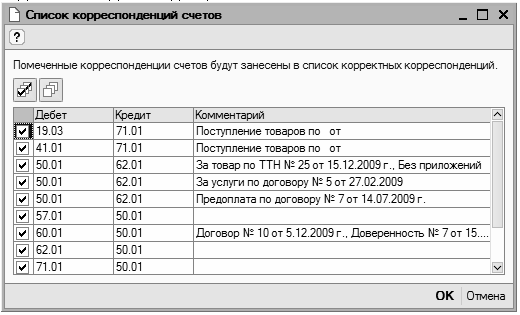


Рисунок – Список корреспонденции счетов

Список проводок, не соответствующих регистру корректных корреспонденций

На рисунке все проводки помечены флажками. В этом случае после нажатия кнопки ОК все они будут добавлены в регистр корректных корреспонденций и, соответственно, впоследствии не будут распознаваться как некорректные. Если какие? то проводки не должны добавляться в этот регистр, снимите соответствующие флажки. Если же вы не хотите добавлять в регистр корректных корреспонденций ни одну из обнаруженных проводок, просто нажмите в данном окне кнопку Отмена. Как уже отмечалось ранее, журнал бухгалтерских проводок можно вывести на печать. Для этого нужно выполнить команду Действия | Вывести список, в результате чего на экране появится окно.

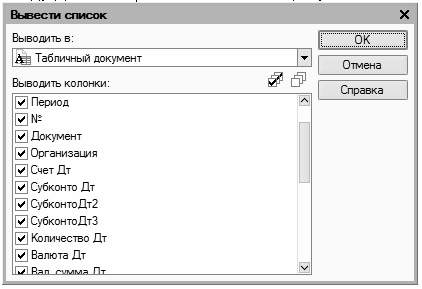


Рисунок – Настройка печатной формы документа

Здесь выполняется настройка печатной формы документа, т. е. определяется, в каком виде и какая информация должна быть выведена на печать. В верхней части окна в поле Выводить в из раскрывающегося списка выбирается формат печатного документа: Табличный документ или Текстовый документ. После этого путем установки соответствующих флажков нужно указать колонки, которые должны включаться в печатную форму документа. На рис. Показана печатная форма журнала бухгалтерских проводок, созданная в табличном формате.



Рисунок – Печатная форма журнала бухгалтерских проводок в табличном формате

Чтобы отправить документ на печать, нужно выполнить команду главного меню Файл | Печать или нажать комбинацию клавиш <Ctrl>+<P>.

# Практическая работа № 19 Использование регистров расчёта

**Цель:** Использование регистров расчёта

**Норма времени**: 4 часа

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

Прежде чем мы начнем создавать объект конфигурации **Регистр расчета Начисления**, нам потребуется создать два [дополнительных](https://stimul.kiev.ua/tags.php?t=%E4%EE%EF%EE%EB%ED%E8%F2%E5%EB%FC%ED%FB%E5&f=info) объекта конфигурации:

Регистр сведений ГрафикиРаботы;

Справочник ВидыГрафиковРаботы.

Справочник понадобится нам для хранения информации о том, какие [графики работы](https://stimul.kiev.ua/tags.php?t=%E3%F0%E0%F4%E8%EA%E8+%F0%E0%E1%EE%F2%FB&f=info) существуют на фирме, а регистр сведений → для указания того, какие дни в месяце являются рабочими, поскольку сумма оплаты по окладу будет рассчитываться исходя из того, сколько дней отработал сотрудник в расчетном месяце.

1. Создадим новый объект конфигурации **Справочник** с именем **ВидыГрафиковРаботы**.

2. В этом справочнике у нас будет два предопределенных [графика работы](https://stimul.kiev.ua/tags.php?t=%E3%F0%E0%F4%E8%EA%E8+%F0%E0%E1%EE%F2&f=info) :

ГрафикАдминистрации;

ГрафикМастеров.

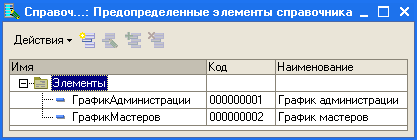


Рисунок – Справочник

3. После этого создадим объект конфигурации **Регистр сведений** с именем **ГрафикиРаботы**.

4. Этот регистр будет иметь два измерения:

**ГрафикРаботы**, тип СправочникСсылка.ВидыГрафиковРаботы;

**Дата**, тип Дата.

5. Затем создадим единственный ресурс регистра → **Значение**, с типом Число, длиной 1.

6. Запустим [1С:Предприятие в](https://stimul.kiev.ua/tags.php?t=1%F1%3A+%EF%F0%E5%E4%EF%F0%E8%FF%F2%E8%E5+8&f=info) режиме отладки и заполним регистр данными о рабочих днях сентября графика мастеров. Чтобы проще выполнить эту довольно однообразную работу, воспользуйтесь возможностью добавления элементов в справочник копированием. У вас должно получиться 22 рабочих дня в сентябре.

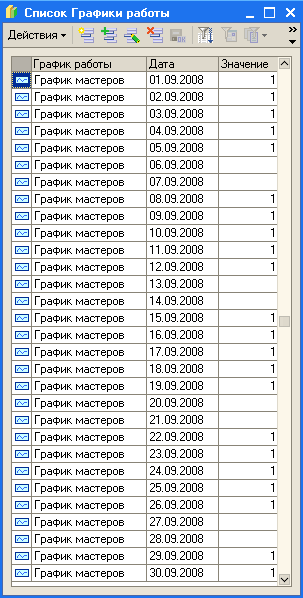


Рисунок – Графики работы

7. Теперь все готово для создания регистра расчета.

Создадим новый объект конфигурации Регистр расчета с именем **Начисления**.

8. В качестве плана видов расчета, используемого регистром, выберем **ОсновныеНачисления**.

9. Установим, что регистр будет использовать период действия, График будет задаваться в [регистре сведений](https://stimul.kiev.ua/tags.php?t=%F0%E5%E3%E8%F1%F2%F0%FB+%F1%E2%E5%E4%E5%ED%E8%E9&f=info) ГрафикиРаботы, Значение графика будет находиться в ресурсе Значение, Дата графика → в измерении Дата.

10. Укажем, что регистр расчета будет использовать базовый период и периодичность регистра будет **Месяц**.

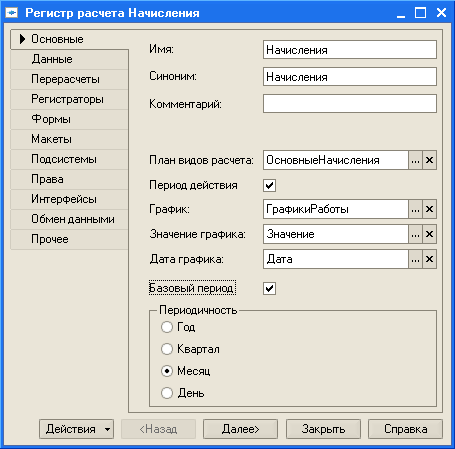


Рисунок – Регистр расчета

11. Затем перейдем на закладку **Данные** и создадим: измерение **Сотрудник**, тип СправочникСсылка.Сотрудники, базовое; ресурс **Результат**, тип Число, длина 15, точность 2; реквизит **ГрафикРаботы**, тип СправочникСсылка.ВидыГрафиковРаботы, связь с графиком по измерению ГрафикРаботы; реквизит **ИсходныеДанные**, тип Число, длина 15, точность 2.

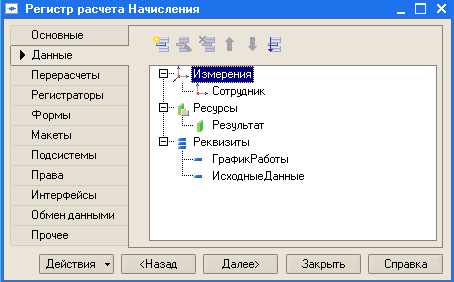


Рисунок – Начисления

Реквизит **ГрафикРаботы** мы будем использовать для того, чтобы связать запись регистра с используемым графиком работы, а реквизит **ИсходныеДанные** → чтобы хранить в нем данные, которые могут понадобиться при расчете или перерасчете (в нашем случае это будет расчет оклада).

12. Теперь перейдем на закладку **Перерасчеты**. Создадим объект конфигурации **Перерасчет**, который так и назовем → **Перерасчет**. У него будет единственное измерение → Сотрудник, для которого в установке связи мы укажем измерение регистра Сотрудник и в качестве данных ведущих регистров выберем то же самое измерение Сотрудник регистра расчета Начисления.

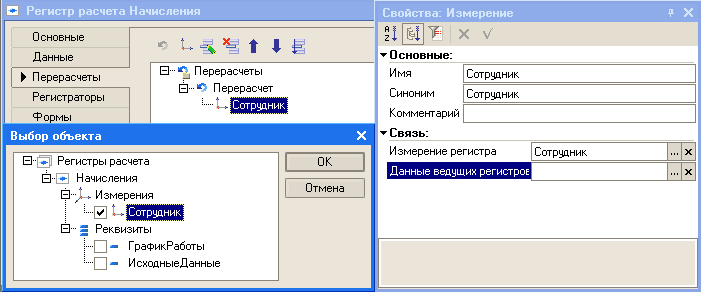


Рисунок – Регистр расчета начисления

На этом создание объекта конфигурации **Регистр расчета Начисления** завершено.

Использование регистра расчета. Создание документа Начисления Сотрудникам

Для того чтобы иметь возможность регистрировать в базе данных начисления, производимые сотрудникам фирмы, нам понадобится специальный документ.

1. Создадим новый документ с именем **Начисления Сотрудникам**.

2. Перейдем на закладку **Нумерация** и установим:

**Тип номера** документа → Число;

**Длина номера** → 5.

3. Этот документ будет иметь табличную часть **Начисления**, содержащую следующие реквизиты:

**Сотрудник**, тип СправочникСсылка.Сотрудники; **ГрафикРаботы**, тип СправочникСсылка.ВидыГрафиковРаботы; **ДатаНачала**, тип Дата; **ДатаОкончания**, тип Дата; **ВидРасчета**, тип ПланВидовРасчетаСсылка.ОсновныеНачисления; **Результат**, тип Число, длина 15, точность 2. Реквизиты **ДатаНачала** и **ДатаОкончания** понадобятся нам для того, чтобы задать период, в котором должна действовать запись расчета.

4. На закладке **Движения** запретим [оперативное проведение](https://stimul.kiev.ua/tags.php?t=%EE%EF%E5%F0%E0%F2%E8%E2%ED%EE%E5+%EF%F0%EE%E2%E5%E4%E5%ED%E8%E5&f=info) документа. Отметим, что документ будет создавать движения по регистру расчета Начисления, и запустим конструктор движения.

5. В окне конструктора выберем табличную часть **Начисления** и нажмем **Заполнить выражения**.

6. Для реквизитов **ПериодДействияКонец** и **БазовыйПериодКонец** укажем выражение **КонецДня(ТекСтрокаНачисления.ДатаОкончания)**.

7. Для поля **ПериодРегистрации** укажем выражение **Дата**.

8. Реквизиту **ИсходныеДанные** поставим в соответствие реквизит табличной части **Результат**.

9. Для реквизита **Результат** удалим выражение, присвоенное ему конструктором.

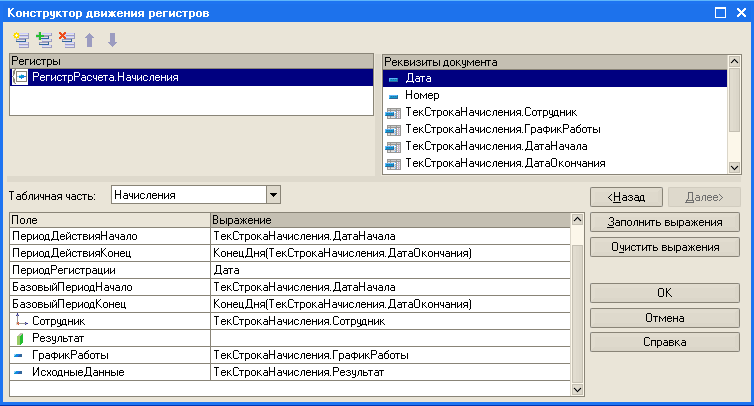


Рисунок – Результат

10. Нажмем ОК и посмотрим текст обработчика, созданный конструктором.

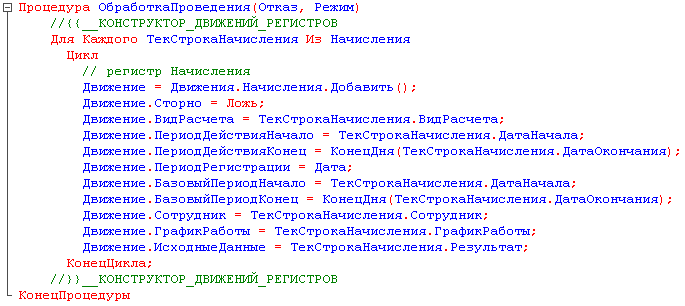


Рисунок – Модуль

11. Запустим [1С:Предприятие в](https://stimul.kiev.ua/tags.php?t=1%F1%3A%EF%F0%E5%E4%EF%F0%E8%FF%F2%E8%FF+8&f=info) режиме отладки и посмотрим, как работает наш документ.

Начислим оклад за сентябрь всем сотрудникам фирмы:

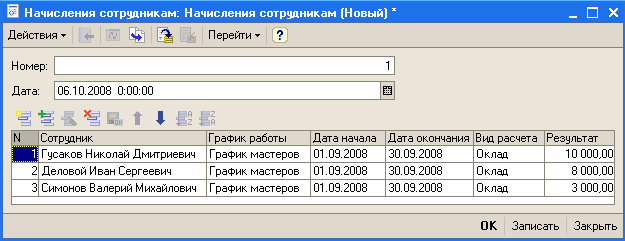
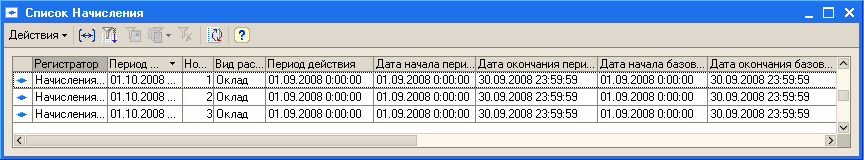


Рисунок – Начисления сотрудникам

12. Проведем документ и посмотрим, какие движения он сформировал в регистре **Начисления**:

[](https://stimul.kiev.ua/img/materialy_praktikum_11_plan_vidov_rascheta_registr_rascheta/372.png)

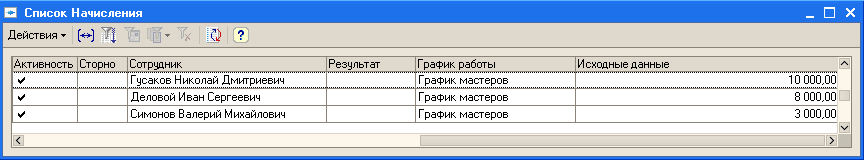
[](https://stimul.kiev.ua/img/materialy_praktikum_11_plan_vidov_rascheta_registr_rascheta/373.png)

Рисунок – Начисления

Обратите внимание на то, что платформа привела период регистрации каждой записи к началу периода регистра расчета (в обработчике проведения мы указывали значение даты документа 06.10.2008). Кроме этого заметьте, что с каждой записью мы сохранили в реквизите **ИсходныеДанные** размер оклада сотрудника, введенный в документе, чтобы в дальнейшем [рассчитать сумму](https://stimul.kiev.ua/tags.php?t=%F0%E0%F1%F1%F7%E8%F2%E0%F2%FC+%F1%F3%EC%EC%FB&f=info) оплаты по окладу.

Для дальнейшего изучения работы регистра расчета нам понадобится служебный отчет, с помощью которого мы сможем посмотреть содержимое записей перерасчета.

**Иллюстрация механизмов вытеснения и зависимости от базы**

1. Создадим новый объект конфигурации **Отчет** и назовем его **Перерасчет**.

2. Создадим основную схему компоновки данных, добавим **Набор данных → запрос** и откроем конструктор запроса.

3. Из виртуальной таблицы перерасчета **Начисления.Перерасчет** выберем все поля:

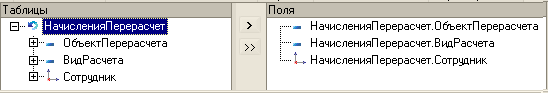


Рисунок – Таблицы

4. На этом создание запроса закончено, нажмем ОК.

5. Перейдем на закладку **Настройки** и добавим группировку детальных записей.

6. На закладке **Выбранные поля** выберем для вывода в отчет поля **ОбъектПерерасчета**, **ВидРасчета** и **Сотрудник**. На этом создание схемы компоновки данных закончено.

7. Запустим [1С:Предприятие](https://stimul.kiev.ua/tags.php?t=1%F1%3A+%EF%F0%E5%E4%EF%F0%E8%FF%F2%E8%FF&f=info) в режиме отладки, сформируем отчет **Перерасчет** и убедимся, что пока он не содержит никаких данных.

8. Создадим новый документ **Начисление сотрудникам №–2**, в котором начислим премию за сентябрь Гусакову и Деловому. Этим документом мы фиксируем тот факт, что данным сотрудникам нужно начислить премию по итогам работы за сентябрь. Поскольку размер премии нам не известен (он будет рассчитываться по некоторому алгоритму), поля **Результат** мы оставляем пустыми. ОК.

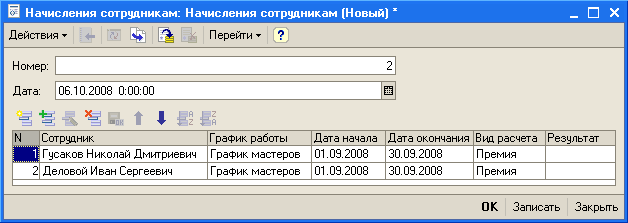


Рисунок – Начисления

9. Теперь снова откроем документ **Начисление сотрудникам №–1** и изменим оклад Гусакова с 10 000 на 7 000. Нажмем ОК. Сформируем отчет **Перерасчет**:

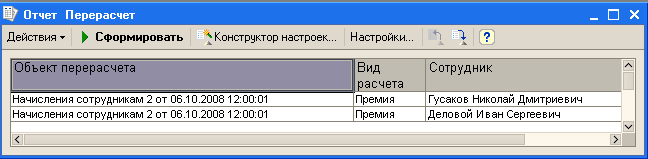


Рисунок – Перерасчет

10. Как видите, отчет теперь содержит какие-то данные. В самом деле, вид расчета **Премия** зависит у нас по базовому периоду от вида расчета **Оклад**. Как только мы изменили существовавшие в регистре записи по виду расчета **Оклад**, платформа сразу же сформировала набор записей перерасчета, которые должны быть рассчитаны заново, так как изменилась их база.

11. Перепроведем документ **Начисление сотрудникам №–2** (премия) и сформируем отчет **Перерасчет**. Он снова не содержит никаких данных → система отметила тот факт, что мы «пересчитали» зависимые записи, и очистила таблицу перерасчета.

12. На этом примере мы познакомились с работой механизма поддержки зависимости по базовому периоду у регистра расчета.

Теперь посмотрим, как работает механизм вытеснения по периоду действия. Для этого нам понадобится создать документ **Начисление сотрудникам №–3**.

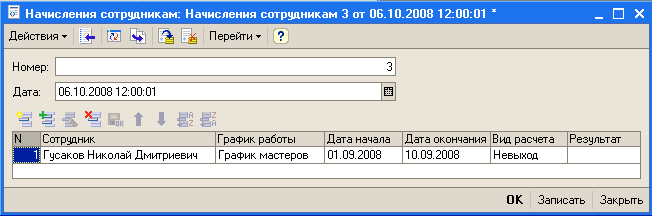


Рисунок – Начисление сотрудникам

13. Этим документом мы зафиксируем тот факт, что Гусаков не выходил на работу 1 по 10 сентября. Очевидно, что в этом случае потребуется пересчитать его оплату по окладу и, как следствие, начисленную премию. Нажмем ОК и сформируем **Перерасчет**.

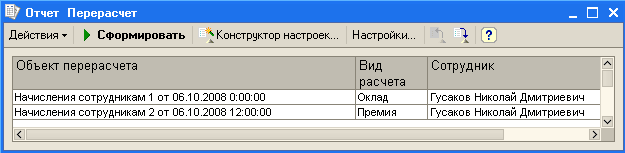


Рисунок – Перерасчет

Как вы видите, в перерасчет попала запись о начислении оклада Гусакову. Это явилось результатом работы механизма вытеснения по периоду действия, ведь вид расчета **Невыход** вытесняет у нас вид расчета **Оклад**.

Обратите внимание, что в перерасчет попала и запись о назначении премии Гусакову, поскольку при создании предопределенных видов расчета мы указали, что результат вида расчета **Премия** будет зависеть от изменения результата вида расчета **Невыход**. Эта зависимость косвенная, но поскольку мы явно указали такую зависимость, платформа ее отследила.

Перепроведем документы **Начисление сотрудникам №–1** и **№–2** и убедимся, что таблица перерасчета очистилась.

# Практическая работа № 20 Создание регламентных заданий

**Цель:** Создание регламентных заданий

**Норма времени**: 2 часа

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

Создаем справочник для регламентных заданий

Наше регламентное задание может порождать множество фоновых процессов – по одному на каждую обработку. У каждого задания в платформе 1С8 есть ключ. Но метод регламентного задания не знает ключ фонового задания, поэтому нужно использовать параметры фонового задания. В результате мы можем видеть фоновое задание в консоли заданий, но не можем вручную из этой консоли создать фоновое задание, т.к. задания с параметрами вручную не создаются.

Справочник «Параметры регламентных заданий»:

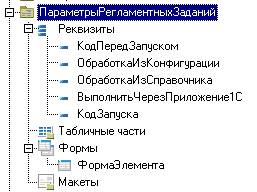
[](http://fixin.com.ru/articles/1s8_research_rtask/article__image002.jpg)

Рисунок – Справочник

Реквизиты:

·         КодПередЗапуском – неограниченная строка – код на языке 1С, который нужно выполнить перед запуском.

·         Обработка из конфигурации – строка (100)  - идентификатор обработки из конфигурации

·         Обработка из справочника – строка (100)  - ссылка на элемент справочника «Внешние обработки», если такой имеется в конфигурации

·         Выполнять через приложение 1С – булево - будет создано отдельное приложение 1С и в нем будет запущено регламентное задание. Создано для 8.1, где не все методы приложения доступны на сервере, где работает регламентное задание.

·         КодЗапуска - – неограниченная строка – код на языке 1С, который будет выполняться при запуске регламентного задания.

Создадим форму элемента:

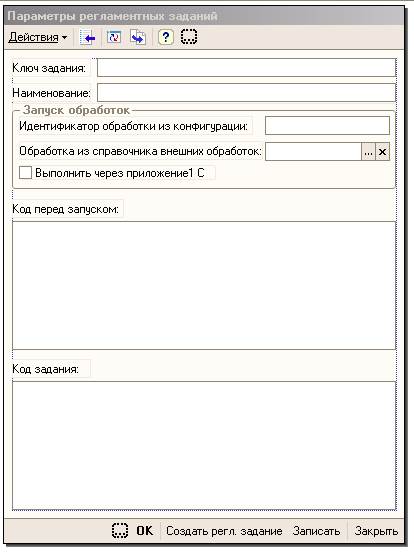
[](http://fixin.com.ru/articles/1s8_research_rtask/article__image003.jpg)

Рисунок – Форма элементов

По нажатию на кнопку «Создать регл. задание» программно создается регламентное задание с ключом-кодом:

Процедура ОсновныеДействияФормыСоздатьРеглЗадание(Кнопка)

       Перем Задание;

       Ключ = СокрЛП(Код);

       Задание = РегламентныеЗадания.СоздатьРегламентноеЗадание("ВыполнениеОбработки");

       Задание.Наименование = Ключ;

       Задание.Ключ = Ключ;

       Параметры = Новый Массив();

       Параметры.Добавить(Ключ);

       Задание.Параметры = Параметры;

       Задание.Записать();

КонецПроцедуры

Запуск регламентного задания

Каждое созданное нами регламентное задание имеет ключ:

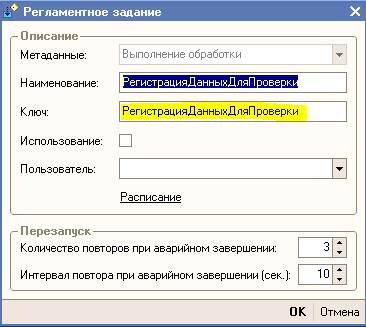
[](http://fixin.com.ru/articles/1s8_research_rtask/article__image004.jpg)

Рисунок – Регламентное задание

Этот ключ соответствует коду справочника «Параметры регламентных заданий», по нему осуществляется поиск при старте задания. Если элемент справочника не найдено, задание не выполняется. Далее, если задан код КодПередЗапуском то будет выполнен этот код. Далее, если переменная Выполнять примет значение ложь, то задание не будет выполнено. При анализе доступна переменная Параметры, где хранится ссылка на найденный элемент справочника «Параметры регламентных заданий».

В зависимости от выбранных значений реквизитов или запустится код на языке 1С, или запустится обработка из конфигурации, или обработка из типового справочника «Внешние обработки». Для 1С81 предусмотрено выполнение в новом приложении – чтобы можно было использовать код, доступный только на клиенте, в том числе и использование внешних обработок. Для этого нужно установить галочку «Выполнять через приложение 1С». Иначе регламентное задание будет выполняться на сервере.

Рекомендую устанавливать пользователя в поле «Пользователь» у вновь созданного регламентного задания, чтобы задание выполнялось под определенными правами. Рекомендую такому пользователю давать полные права. У меня используется пользователь «robot».

Расписание регламентного задания создается по гиперссылке «Расписание» из формы регламентного задания. Можно использовать обработку «Консоль регламентных заданий».

# Практическое задание № 21 Создание ролей. Ограничение доступа к данным на уровне записей и полей базы данных

**Цель:** Создание ролей

**Норма времени**: 2 часа

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

Создание ролей

В режиме «Конфигуратор»

При создании ролей исходят, как правило, из того, какие полномочия требуются различным группам пользователей на доступ к информации. Для этого мы воспользуемся подсистемами, которые значительно облегчат нашу задачу.

Администратор

Первая роль, которую мы создадим, будет *Администратор*. Она должна включать в себя полные права на работу с данными информационной базы.

Раскроем ветвь *Общие* дерева объектов конфигурации. Выделим строку *Роли* и добавим новый объект конфигурации *Роль*. Зададим его имя – *Администратор* (рис. 22.1).

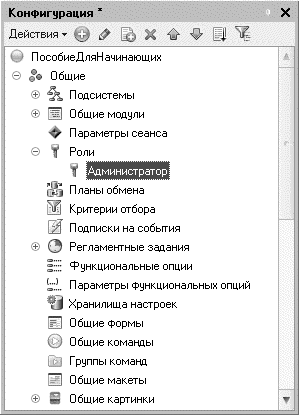


Рисунок – Создание роли

Откроется окно редактирования прав этой роли

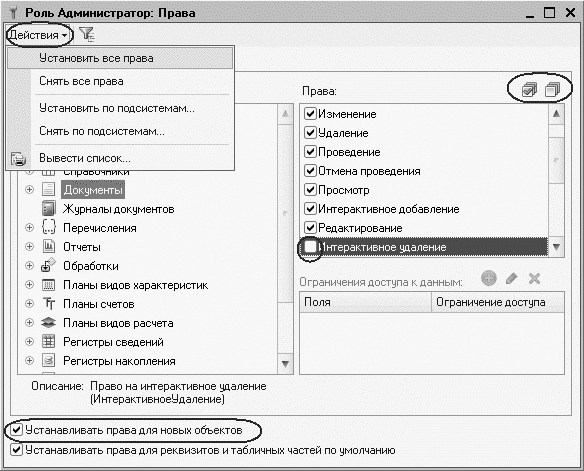


Рисунок – Окно редактирования прав для роли «Администратор»

Слева, в списке объектов, перечислены все объекты и виды объектов конфигурации, а справа, в окне прав, – доступные права для выбранного объекта или видов объектов конфигурации.

Администратор должен иметь права на все объекты и все виды объектов. Для этого выполним команду *Действия* > *Установить все права* в командной панели окна. После этого все права для всех объектов будут помечены. Однако можно поставить или снять отметку для прав конкретного объекта конфигурации, пользуясь кнопками *Отметить все элементы* и*Снять отметку со всех элементов*, расположенными над окном прав. Теперь единственное, что следует сделать, – снять разрешение на интерактивное удаление для всех объектов. Это необходимо для того, чтобы администратор случайно не мог удалить какой-либо объект базы данных. Для этого пройдемся по всем видам объектов конфигурации (*Справочники*, *Документы* и т. д.) и снимем отметку с команды *Интерактивное удаление* (см. рис. 22.2). Для того чтобы наш *Администратор* мог работать с объектами, которые мы будем создавать после расстановки прав, зададим для него параметр *Устанавливать права для новых объектов*(см. рис. 22.2). На этом создание роли *Администратор* закончено. Директор Следующей ролью, которую мы создадим, будет роль *Директор*. Создадим новый объект конфигурации *Роль* с именем *Директор*. Нас устраивает, что у новой роли нет прав на доступ ко всем объектам, за исключением тех видов объектов конфигурации, для которых не создано ни одного объекта. Для таких видов объектов конфигурации останутся установленными полные права.

Установим право *Вывод* для всей конфигурации (рис. 22.3).

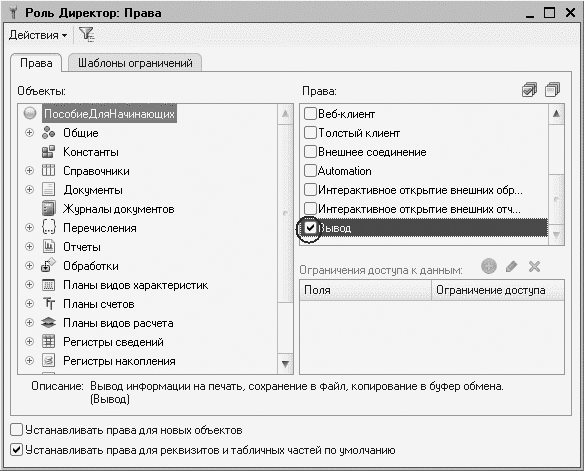


Рисунок – Окно редактирования прав для роли «Директор»

Теперь нам останется лишь пройти по видам объектов конфигурации и установить для них право *Просмотр* (права *Чтение* и *Использование* при этом установятся автоматически). Затем раскроем ветвь *Общие*, выделим ветвь *Подсистемы* и отметим право *Просмотр* у всех подсистем, кроме подсистемы *Предприятие*. Тем самым мы предоставим директору возможность просматривать все данные информационной базы, но исключим из его интерфейса все действия, которые по логике нашей конфигурации не относятся к прикладной ее части. Вторая роль нашей конфигурации готова. Мастер Следующая роль, которую мы создадим, будет роль *Мастер*. Снова добавим новый объект конфигурации Роль с именем *Мастер*. Выполним команду *Действия*> *Установить по подсистемам*и выберем подсистемы*УчетМатериалов* и *ОказаниеУслуг*. В результате будут установлены все права на объекты конфигурации, относящиеся к данным подсистемам. Если теперь установить фильтр объектов по подсистемам *УчетМатериалов* и *ОказаниеУслуг*, то можно при необходимости внести уточнения в установленные права (рис. 22.4).

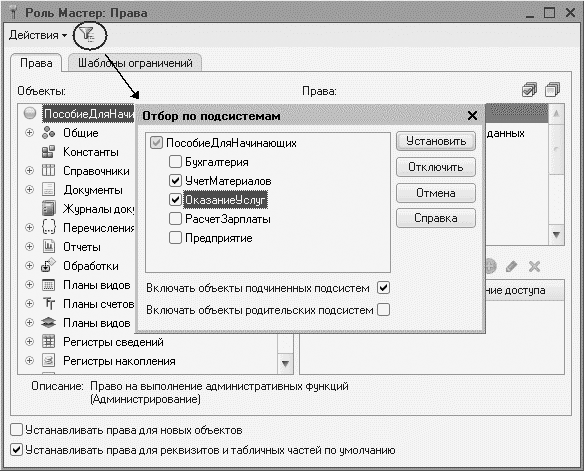


Рисунок – Установка фильтра по подсистеме

В частности, для справочника *Сотрудники* мы запретим права *Добавление*, *Изменение* и *Удаление*. Обратите внимание, что при запрете права *Добавление* исчезла отметка и у права *Интерактивное добавление*, так как оно является «уточнением» права *Добавление*. Точно так же уточненные права запрещаются и при отмене прав на изменение и удаление. Кроме этого, мы снова снимем разрешения на интерактивное удаление для всех объектов базы данных. Теперь пройдем по всем видам объектов конфигурации и снимем у всех право *Интерактивное удаление*. Затем снимем фильтр и установим все права, кроме интерактивного удаления для следующих объектов конфигурации: справочник*ВариантыНоменклатуры,* справочник*ДополнительныеСвойстваНоменклатуры,* план видов характеристик*СвойстваНоменклатуры,* регистр сведений*ЗначенияСвойствНоменклатуры.* Эти объекты мы не привязывали ни к каким подсистемам, но они будут нужны для работы с характеристиками номенклатуры. Роль *Мастер* готова.

Расчетчик

В заключение нам с вами осталось создать две роли: *Бухгалтер* и *Расчетчик*. Мы разделим права по расчету зарплаты и по ведению бухгалтерского учета. Дело в том, что в ООО «На все руки мастер» есть бухгалтер и помощник бухгалтера. Помощник бухгалтера занят в основном расчетом зарплаты, но иногда это делает и главный бухгалтер.

Поэтому главному бухгалтеру необходимо будет назначить обе роли, в то время как помощнику – только роль *Расчетчик*.

Создадим новый объект конфигурации Роль с именем *Расчетчик*. В окне редактирования прав установим их по подсистеме *РасчетЗарплаты* (и не забудем запретить интерактивное удаление). А также установим право *Просмотр* для объекта конфигурации: *Регистр накопления Продажи.* Роль *Расчетчик* готова.

Бухгалтер

В заключение создадим объект конфигурации *Роль* с именем *Бухгалтер*. В окне редактирования прав установим их по подсистеме *Бухгалтерия*. После этого отфильтруем список объектов по этой подсистеме и для справочника *Номенклатура* запретим добавление, изменение и удаление. Также запретим интерактивное удаление для всех объектов. Затем снимем фильтр и установим все права, кроме интерактивного удаления для следующих объектов конфигурации: справочник*Субконто,* регистр бухгалтерии*Управленческий.* А также установим право *Просмотр* для следующих объектов конфигурации: справочник*Склады,* справочник*ВариантыНоменклатуры,* справочник*ДополнительныеСвойстваНоменклатуры,* план видов характеристик*СвойстваНоменклатуры,* регистр сведений*ЗначенияСвойствНоменклатуры*. Права на запуск клиентских приложений В заключение установим права на запуск клиентского приложения для каждой роли. Для этого воспользуемся другим, более удобным инструментом – редактором *Все роли*. В дереве объектов конфигурации выделим ветку *Роли* и в контекстном меню выполним команду *Все роли*. В списке объектов конфигурации выделим корень, и для всех ролей установим право *Тонкий клиент*

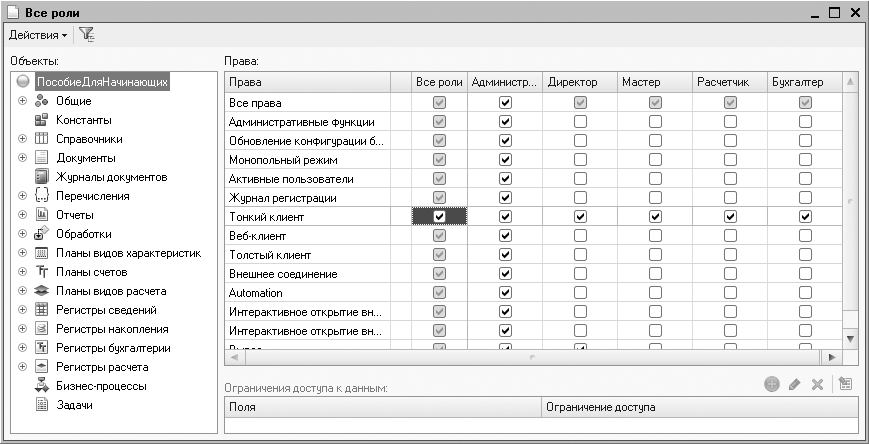


Рисунок – Список прав для роли «Бухгалтер»

Тем самым мы разрешили всем пользователям подключаться к информационной базе только с помощью тонкого клиента. Администратор же имеет возможность подключаться и с помощью других клиентских приложений. Это может понадобиться ему, например, для создания планировщика заданий, о котором шла речь в разделе [«Планировщик заданий»](https://its.1c.ru/db/content/pubdevguide/src/421.html?_=1491398150#h421). Список прав для каждой роли можно получить, выполнив в окне редактирования прав команду *Действия* > *Вывести список*

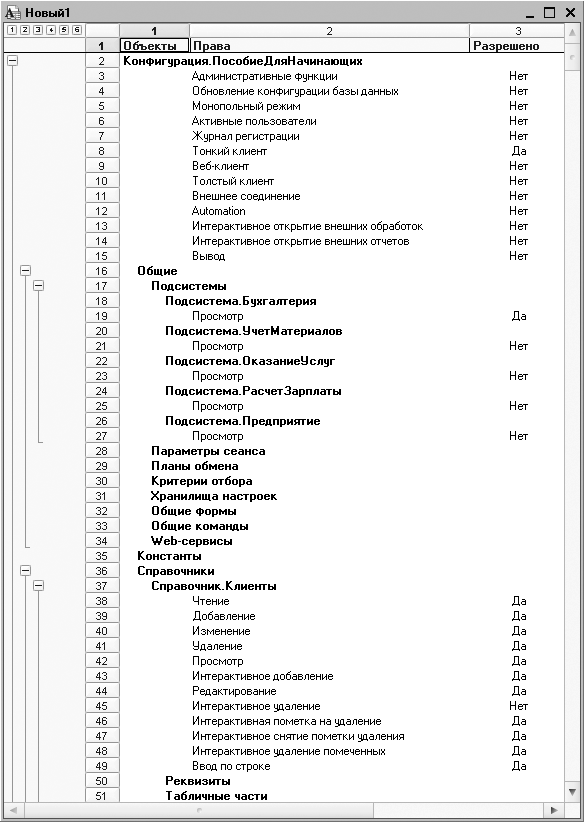


Рисунок - Список прав для роли «Бухгалтер»

Аналогичный список, но только для всех ролей, которые есть в конфигурации, можно получить из редактора *Все роли*.

Ограничение доступа к данным на уровне записей и полей базы данных

В завершение занятия мы покажем, как можно ограничить доступ к данным более точно в зависимости от самих данных, которые хранятся в информационной базе.

Для этого в системе «1С:Предприятие 8» используется *механизм ограничения доступа на уровне записей и полей базы данных*. Этот механизм позволяет для четырех основных прав (чтение, добавление, изменение и удаление) уточнить, какие же именно данные информационной базы будут доступны пользователю.

Такое уточнение записывается на специальном языке, являющемся подмножеством языка запросов.

Далее, на примере документа *Начисления сотрудникам*, мы рассмотрим небольшой пример, когда мастерам нужно дать возможность просмотреть начисленную им зарплату, но руководство запрещает им доступ к информации о начисленной премии.

Другими словами, мастерам нужно запретить просмотр тех документов *Начисления сотрудникам*, в которых есть записи о начислении премии.

режиме «Конфигуратор»

Для решения этой задачи сначала установим для роли *Мастер* право *Просмотр* для документа *НачисленияСотрудникам*.

Поскольку этот документ принадлежит подсистеме *РасчетЗарплаты*, дадим право на просмотр этой подсистемы. Также дадим права на просмотр справочника *ВидыГрафиковРаботы* и плана видов расчета *Основные начисления*, т. к. ссылки на эти объекты используются в документе *НачисленияСотрудникам*. Вернемся к редактированию прав роли *Мастер*.

Как мы видим, при установке права *Просмотр* право *Чтение* документа *НачисленияСотрудникам* установилось автоматически. Выделим его. В правой нижней части экрана находится поле *Ограничение доступа к данным*. Нажмем кнопку *Добавить*

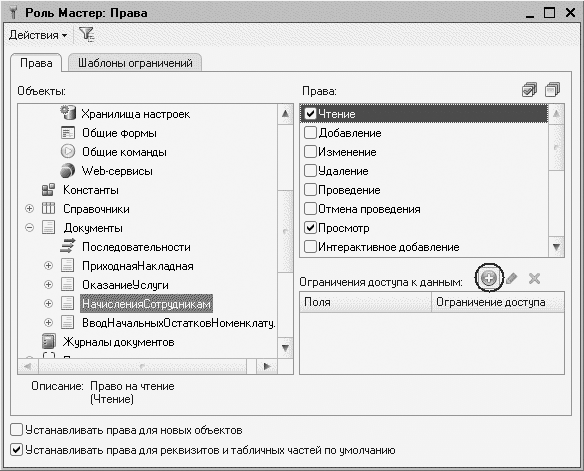


Рисунок - Добавление ограничения доступа для роли «Мастер»

Мы хотим запретить доступ ко всем полям документа *Начисление сотрудникам*.

Поэтому мы не будем выбирать поля, а нажмем кнопку выбора в поле *Ограничение доступа*

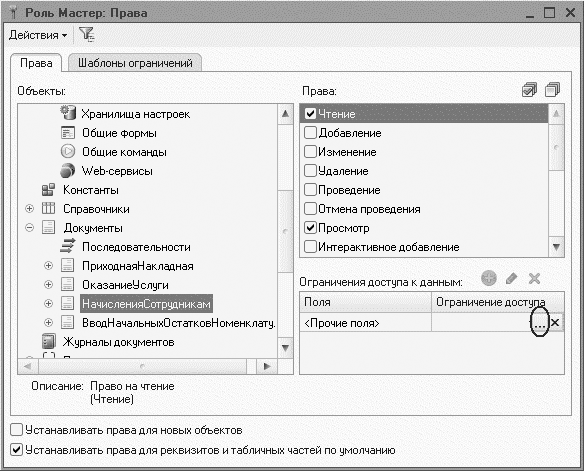


Рисунок - Вызов редактора ограничений доступа

Откроется окно *Ограничение доступа*, в котором можно задать текст на специальном языке, который является подмножеством языка запроса.

Для облегчения работы мы воспользуемся конструктором запроса. Нажмем кнопку *Конструктор запроса*.

Откроется конструктор ограничений доступа к данным. Он похож на конструктор запросов

Таблица *НачисленияСотрудникам* автоматически попала на закладку *Таблицы и поля*, а конструктор открылся на закладке *Условия*.

Перенесем в список условий поле *ВидРасчета* табличной части *Начисления*, установим флажок *Произвольное* и заполним правую часть условия, как показано в листинге 22.1 (рис. 22.13).

**Листинг 22.1.** Ограничение доступа к данным

НачисленияСотрудникам.Начисления.ВидРасчета ЗНАЧЕНИЕ(ПланВидовРасчета.ОсновныеНачисления.Премия)

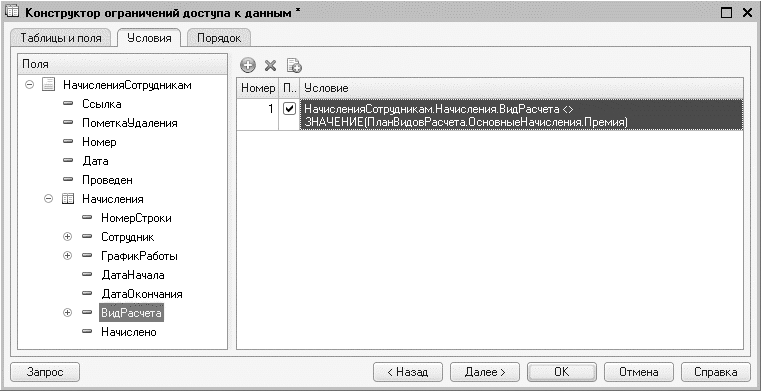


Рисунок - Конструктор ограничений доступа к данным

Нажмем *ОК*

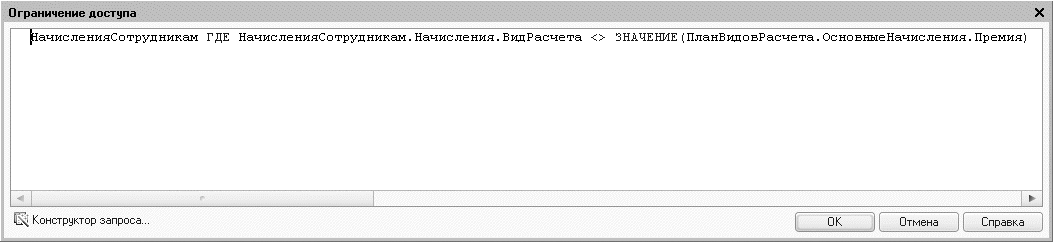


Рисунок - Текст ограничения доступа к данным

Текущий пользователь имеет право прочитать или изменить некоторый объект базы данных только в том случае, если ограничение доступа предоставляет ему такое право. То есть когда условие ограничения истинно.

В нашем случае пользователь сможет прочитать документ *Начисления сотрудникам* *НачисленияСотрудникам ГДЕ…*только в том случае, если в его табличной части *Начисления* *… ГДЕ НачисленияСотрудникам.Начисления …*есть виды расчета *… ГДЕ НачисленияСотрудникам.Начисления.ВидРасчета …,*не являющиеся видом расчета *Премия* *… <> ЗНАЧЕНИЕ(ПланВидовРасчета.ОсновныеНачисления.Премия)*

Нажмем *ОК*.

Окно ограничений доступа к данным для роли *Мастер* будет выглядеть следующим образом

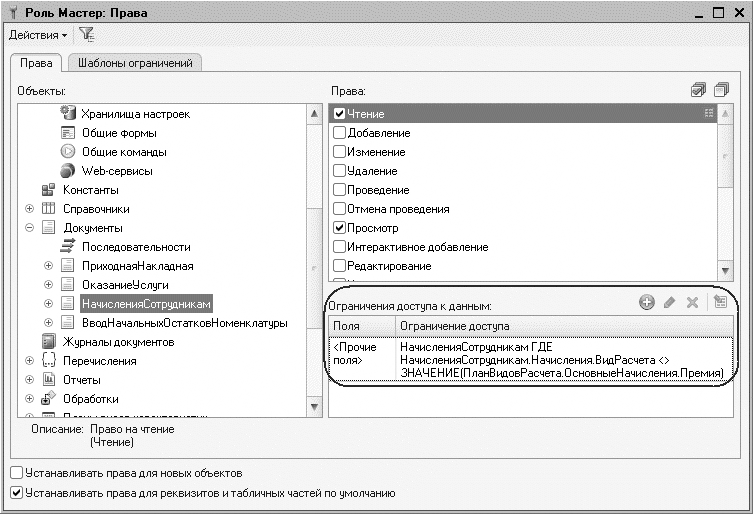


Рисунок - Ограничение доступа для роли «Мастер»

В режиме «1С:Предприятие»

Обновим информационную базу, нажав *F7*, и запустим «1С:Предприятие» для пользователя с ролью *Мастер*, например, для пользователя *Гусаков*.

В разделе *Расчет зарплаты* откроем список документов *НачисленияСотрудникам*

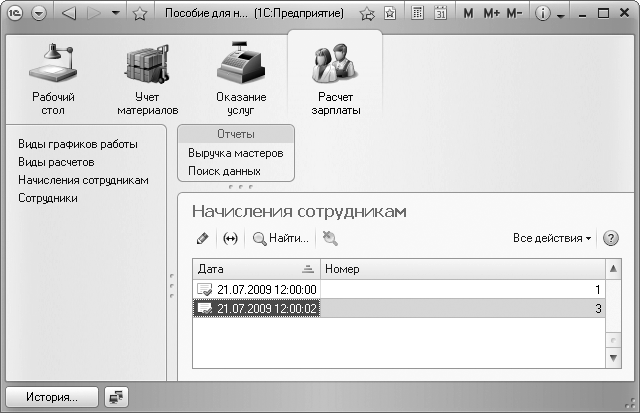


Рисунок - Список документов «Начисления сотрудникам»

Как мы видим, в списке показаны только документы № 1 и № 3, так в документе № 2 начисляется премия.

Немного усложним задачу. Мы все так же не хотим, чтобы мастер видел начисленные премии, но в то же время не хотим скрывать от него факт существования такого документа.

Другими словами, в списке документов мастер должен его видеть, но не должен иметь возможности открыть его.

В режиме «Конфигуратор»

Вернемся в конфигуратор и посмотрим на наше ограничение.

Мы не задавали никаких полей, поэтому ограничение применяется ко всем полям документа

Поэтому сейчас мы безусловно разрешим читать те поля документа, которые необходимы для отображения документа в списке.

Тем самым мы разрешим документу отображаться в списке. Но поскольку существующее условие на прочие поля мы удалять не будем, то открыть документ, как и раньше, можно будет только в том случае, если в его табличной части есть виды расчета, отличные от *Премия*.

Добавим к ограничениям доступа еще одно условие.

В списке полей выберем поля:

*Ссылка*,

*ПометкаУдаления,*

*Номер*,

*Дата*,

*Проведен.*

В ограничении доступа напишем *ГДЕ ИСТИНА*

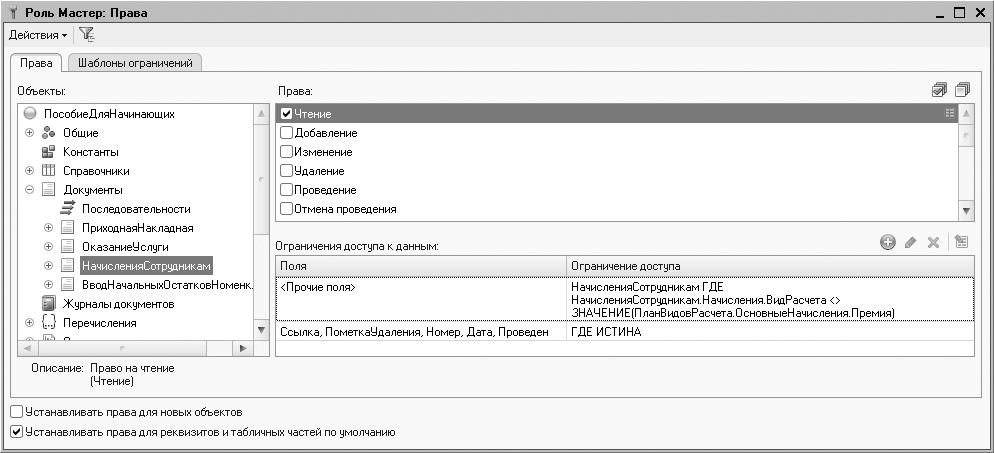


Рисунок - Установка ограничений доступа к данным для роли «Мастер»

Закроем окно редактирования прав.

В режиме «1С:Предприятие»

Обновим информационную базу, нажав *F7*, и запустим «1С:Предприятие» для пользователя с ролью *Мастер*, например, для пользователя *Гусаков*.

В разделе *Расчет зарплаты* откроем список документов *НачисленияСотрудникам*

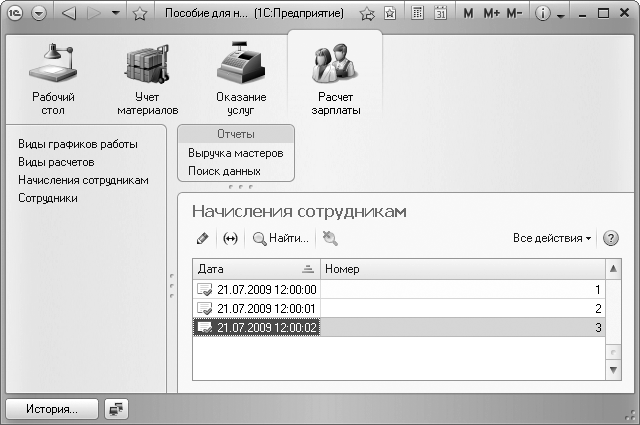


Рисунок - Установка ограничений доступа к данным для роли «Мастер»

В списке документов мы увидим все документы начислений. Документы № 1 и № 3 мы сможем открыть и просмотреть, но при попытке открыть документ № 2 мы получим сообщение о нарушении прав доступа (рис. 22.19).

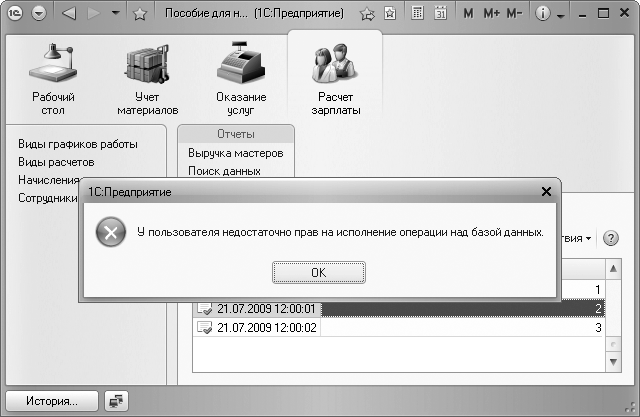


Рисунок - Установка ограничений доступа к данным для роли «Мастер»

То есть мы добились того, чего хотели.

Теперь обратим внимание на следующий момент.

Все хорошо, пока в документе № 2 содержатся записи только о расчете премии. Но вспомним как формулируется наше ограничение доступа: пользователь сможет прочитать документ *Начисления сотрудникам*только в том случае, если в его табличной части *Начисления*есть виды расчета, не являющиеся видом расчета *Премия*.

Это значит, что если в этом документе окажутся виды расчета, отличные от *Премия*, мастер сможет его открыть и просмотреть.

Убедимся в этом.

Запустим «1С:Предприятие» от имени пользователя *Администратор*.

В разделе *Расчет зарплаты* откроем список документов *НачисленияСотрудникам*. Откроем документ № 2 и скопируем любую его строку. В новой строке изменим вид расчета на *Оклад*. Проведем и закроем документ. Завершим сеанс работы.

Теперь запустим «1С:Предприятие» от имени пользователя *Гусаков*. Точно так же в разделе *Расчет зарплаты* откроем список документов *НачисленияСотрудникам*.

Откроем документ № 2. Документ откроется, и мы увидим все его строки.

В режиме «Конфигуратор»

Вернемся в конфигуратор.

Для того чтобы документ невозможно было просмотреть и в этой ситуации, нам нужно будет изменить существующее условие ограничения доступа.

Новое условие будет более сложным, поэтому заодно мы продемонстрируем использование шаблонов в ограничениях доступа.

Итак, откроем роль *Мастер* и перейдем на закладку *Шаблоны ограничений* (рис. 22.20).

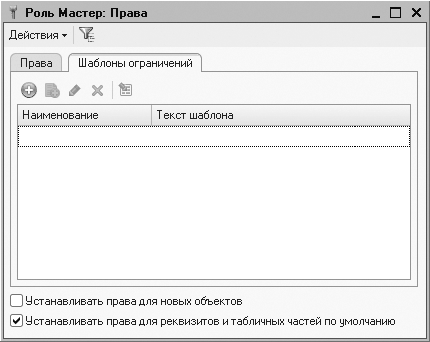


Рисунок - Установка ограничений доступа к данным для роли «Мастер»

Здесь добавим новый шаблон, назовем его *ЕстьПремия*.

Текст шаблона будет выглядеть следующим образом:

**Листинг 22.2.** Ограничение доступа к данным

ВЫБРАТЬ

1

ИЗ

Документ.НачисленияСотрудникам.Начисления

ГДЕ

Документ.НачисленияСотрудникам.Начисления.ВидРасчета = ЗНАЧЕНИЕ(ПланВидовРасчета.ОсновныеНачисления.Премия)

И Документ.НачисленияСотрудникам.Начисления.Ссылка = #Параметр(1).Ссылка

По сути это запрос к табличной части документа *НачисленияСотрудникам*, который либо не вернет нам ничего, либо вернет одну запись с одним полем, в котором будет значение 1.

Такую запись он вернет нам в том случае, если в табличной части документа есть вид расчета *Премия*.

Второе условие в этом запросе нужно нам для того, чтобы указать, табличная часть какого именно документа нас интересует. В этом условии используется возможность указания параметров в шаблоне.

**Листинг 22.3.** Ограничение доступа к данным

И Документ.НачисленияСотрудникам.Начисления.Ссылка = #Параметр(1).Ссылка

Вместо *#Параметр(1)* будет подставлена та строка, которую мы укажем при вызове этого шаблона в условии ограничения доступа.

Теперь вернемся на закладку *Права*.

В имеющемся ограничении прав доступа для прочих полей заменим старый текст новым (листинг 22.4).

**Листинг 22.4.** Ограничение доступа к данным

ДокНачисления ГДЕ НЕ 1 В (#ЕстьПремия("ДокНачисления"))

Здесь с помощью конструкции *#ЕстьПремия("ДокНачисления")* мы обращаемся к нашему шаблону. Текст шаблона просто механически будет подставлен в это место, причем строка *ДокНачисления* заменит собой первый параметр шаблона (*#Параметр(1)*).

Как мы уже говорили, если в табличной части есть начисление *Премия*, запрос в шаблоне вернет единственную запись со значением 1.

Поэтому это условие (см. листинг 22.4) разрешит нам прочитать *ДокНачисления* тогда, когда запрос из шаблона не возвращает 1:

ГДЕ НЕ 1 В (#ЕстьПремия("ДокНачисления"))

То есть тогда, когда в табличной части нет начисления *Премия*.

Можно было бы записать это условие ограничения и без использования шаблонов.

Но, во-первых, такая запись была бы менее читаемой (листинг 22.5), а во-вторых, использование шаблонов позволяет выделить и не дублировать части условий ограничений, которые могут использоваться в разных условиях.

**Листинг 22.5.** Ограничение доступа к данным

ДокНачисления ГДЕ НЕ 1 В (

ВЫБРАТЬ

1

ИЗ

Документ.НачисленияСотрудникам.Начисления

ГДЕ

Документ.НачисленияСотрудникам.Начисления.ВидРасчета = ЗНАЧЕНИЕ(ПланВидовРасчета.ОсновныеНачисления.Премия)

И Документ.НачисленияСотрудникам.Начисления.Ссылка = ДокНачисления.Ссылка)

Закроем окно редактирования прав. Проверим, как это работает.

В режиме «1С:Предприятие»

Обновим информационную базу, нажав *F7*, и запустим «1С:Предприятие» от имени пользователя *Гусаков*.

В разделе *Расчет зарплаты* откроем список документов *НачисленияСотрудникам*.

Как вы помните, в документе № 2 есть строки и с видом расчета *Премия*, и с видом расчета *Оклад*. Раньше этот документ у нас открывался.

Попробуем открыть его теперь.

Мы получим сообщение о нарушении прав доступа, что нам и требовалось (см. рис. 22.19).

В режиме «Конфигуратор»

Поскольку пример с ограничением прав доступа на уровне записей и полей базы данных мы делали скорее в демонстрационных целях, вернемся к исходному состоянию конфигурации.

Снимем для роли *Мастер* право *Чтение* для документа *НачисленияСотрудникам*. Снимем право *Просмотр* для подсистемы *РасчетЗарплаты*. Снимем право *Чтение* для справочника *ВидыГрафиковРаботы* и для плана видов расчета *Основные начисления*. Запустим «1С:Предприятие» от имени пользователя *Администратор*. В разделе *Расчет зарплаты* откроем список документов *НачисленияСотрудникам*. Откроем документ № 2 и удалим последнюю строку (которую мы добавляли). Проведем и закроем документ.

# Практическая работа № 22 Интерактивный обмен. Программный обмен

**Цель:** Интерактивный обмен

**Норма времени**: 6 часов

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

Интерактивный обмен

В режиме «Конфигуратор»

Для построения распределенной информационной базы нам понадобится создать еще один объект конфигурации *План обмена*, который мы назовем *Отделения*, представление объекта – *Отделение*.

Для этого плана обмена мы установим свойство *Распределенная информационная база*

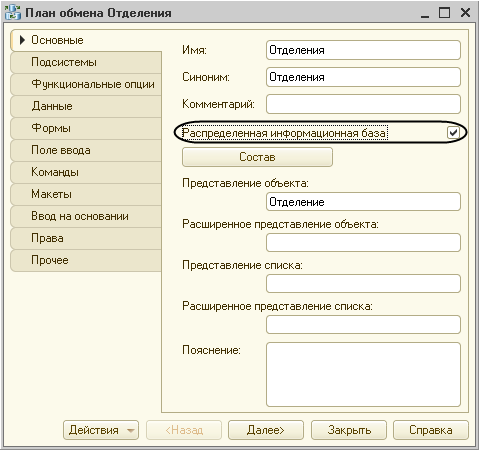


Рисунок - Установка свойства «Распределенная информационная база»

На закладке *Поле ввода* установим флажок *Быстрый выбор*, чтобы иметь возможность выбора узлов плана обмена из выпадающего списка. Затем определим состав объектов, участвующих в обмене.

На закладке *Основные* нажмем кнопку *Состав* и определим тот же состав данных для обмена, что и в плане обмена *Филиалы*. Включим в обмен все объекты, не относящиеся к ведению бухгалтерии и расчету зарплаты.

В результате состав данных обмена должен выглядеть следующим образом (рис. 24.20).

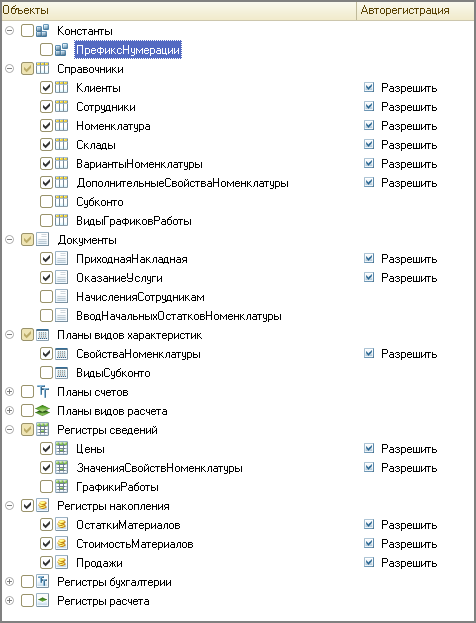


Рисунок - Состав данных обмена

На закладке *Подсистемы* укажем принадлежность плана обмена к подсистеме *Предприятие*.

В окне настройки командного интерфейса этой подсистемы перенесем команду для открытия плана обмена *Отделения* в группу *Панель навигации. Важное* (после команды *Филиалы*) и установим видимость этой команды только для роли *Администратор*. А также в группе команд *Панель действий .Создать* включим видимость у команды *Отделение: создать* только для роли *Администратор*.

Интерактивный обмен

В режиме «Конфигуратор»

Для построения распределенной информационной базы нам понадобится создать еще один объект конфигурации *План обмена*, который мы назовем *Отделения*, представление объекта – *Отделение*.

Для этого плана обмена мы установим свойство *Распределенная информационная база*

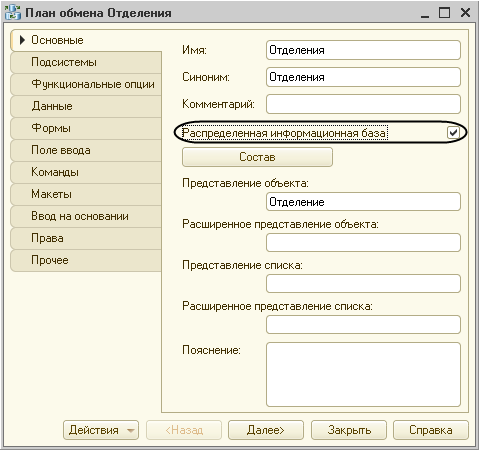


Рисунок --  Установка свойства «Распределенная информационная база»

На закладке *Поле ввода* установим флажок *Быстрый выбор*, чтобы иметь возможность выбора узлов плана обмена из выпадающего списка. Затем определим состав объектов, участвующих в обмене.

На закладке *Основные* нажмем кнопку *Состав* и определим тот же состав данных для обмена, что и в плане обмена *Филиалы*. Включим в обмен все объекты, не относящиеся к ведению бухгалтерии и расчету зарплаты.

В результате состав данных обмена должен выглядеть следующим образом (рис. 24.20).

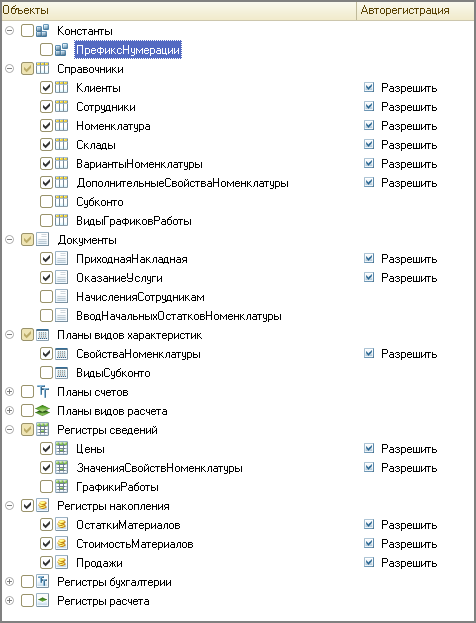


Рисунок Состав данных обмена -

На закладке *Подсистемы* укажем принадлежность плана обмена к подсистеме *Предприятие*.

В окне настройки командного интерфейса этой подсистемы перенесем команду для открытия плана обмена *Отделения* в группу *Панель навигации. Важное* (после команды *Филиалы*) и установим видимость этой команды только для роли *Администратор*. А также в группе команд *Панель действий. Создать* включим видимость у команды *Отделение: создать* только для роли *Администратор*.

В режиме «1С:Предприятие»

Запустим «1С:Предприятие» в режиме отладки. Откроем план обмена *Отделения* и зададим параметры центрального узла (предопределенный элемент плана обмена). Для этого выполним команду *Отделения*в панели команд раздела *Предприятие*.

В списке планов обмена уже присутствует одна запись. Это предопределенный узел нашей информационной базы. Откроем и отредактируем его.

Внесем код *ЦБ* и наименование *Центральная база*. После этого создадим новый узел с кодом *Отд* и наименованием *Отделение*.

Для созданного нами узла доступны три кнопки в командной панели формы плана обмена: *Записать изменения*,*Прочитать изменения* и *Создать начальный образ*

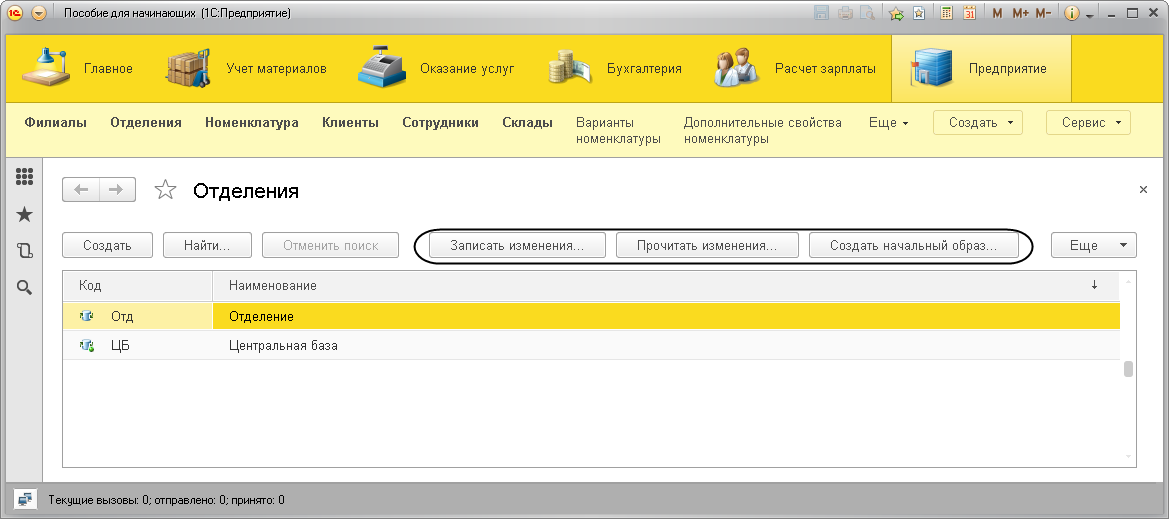


Рисунок - Команды работы с распределенной информационной базой

Не откладывая, воспользуемся одной из них, чтобы создать начальный образ информационной базы нашего отделения. Для этого нам потребуется сначала создать на диске новый каталог, в котором будет располагаться база отделения.

После этого выделим в списке узлов обмена узел *Отделение,* выполним команду *Создать начальный образ*. В появившемся окне укажем, что информационная база будет расположена на данном компьютере, и нажмем кнопку *Создать начальный образ*. На следующем шаге нажмем *Сохранить* и укажем каталог информационной базы

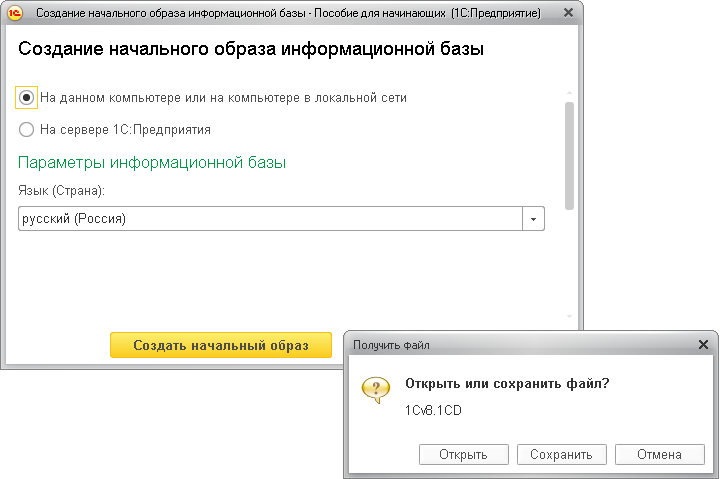


Рисунок -- Создание начального образа информационной базы

Вернемся в диалог создания начального образа информационной базы и нажмем *Закрыть*.

Система создаст в указанном каталоге начальный образ информационной базы нашего отделения.

Программный обмен

Все описанные выше действия по обмену данными в распределенной информационной базе можно выполнить программно.

Мы создадим обработку, которая будет программно выполнять для выбранного узла все те действия, которые были рассмотрены в предыдущем разделе.

В режиме «Конфигуратор»

В конфигураторе центральной базы создадим новый объект конфигурации *Обработка* с именем *ОбменСОтделениями*.

На закладке *Формы* создадим основную форму обработки. В окне редактора форм на закладке *Реквизиты* добавим реквизит формы *ПолеВводаОтделение* с типом *ПланОбменаСсылка.Отделения* и перетащим его в окно элементов формы.

В открывшейся палитре свойств этого поля зададим заголовок – *Отделение*, вид поля (*Поле ввода)* оставим по умолчанию (рис. 24.27).

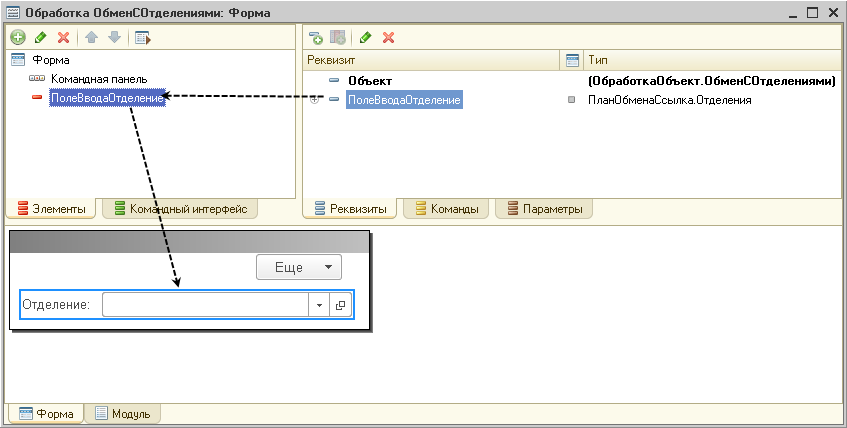


Рисунок - Редактирование формы обработки

Затем на закладке *Команды* поочередно создадим команды *СоздатьНачальныйОбраз, ЗаписатьИзменения*и *ПрочитатьИзменения*. Нажмем кнопку открытия в строке *Действие* для каждой команды.

Создадим клиентские обработчики этих команд в модуле формы, но заполнять их пока не будем, а перейдем на закладку *Форма* и поочередно перетащим эти команды в окно элементов формы.

В результате форма обработки примет вид

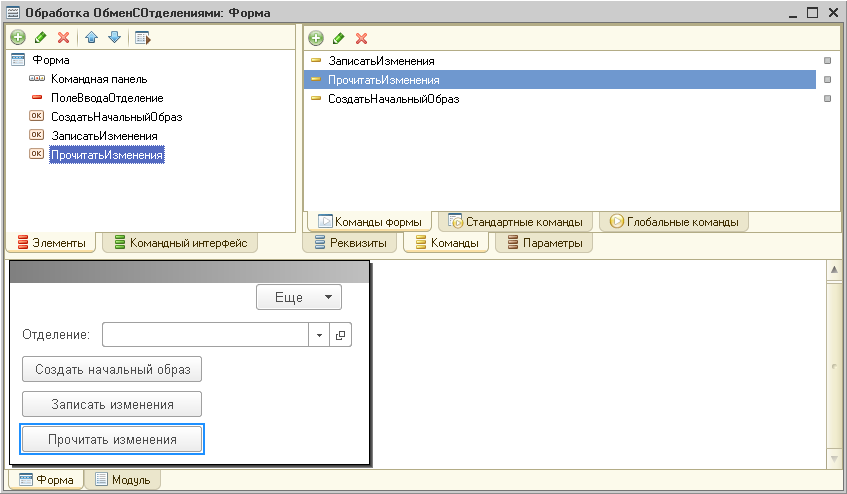


Рисунок - Форма обработки

Откроем палитру свойств кнопки *СоздатьНачальныйОбраз* и снимем флажок у свойства *Доступность*.

Таким образом, при открытии обработки кнопка будет недоступной, пока не выбран узел плана обмена в поле *ПолеВводаОтделение*. Эта кнопка также будет недоступной в случае выбора предопределенного узла нашей информационной базы, то есть создание начального образа невозможно, если выбранный узел является предопределенным.

Чтобы обеспечить такое поведение кнопки, создадим в модуле формы обработки функцию, выполняющуюся на сервере и возвращающую истину, если переданный в функцию узел является предопределенным (листинг 24.25).

**Листинг 24.25.** Функция «ПредопределенныйУзел()»

&НаСервереБезКонтекста

Функция ПредопределенныйУзел(Узел)

Возврат Узел = ПланыОбмена.Отделения.ЭтотУзел();

КонецФункции

Затем в окне элементов формы выделим элемент *ПолеВводаОтделение*, вызовем его палитру свойств и создадим клиентский обработчик события *ОбработкаВыбора*.

Заполним обработчик следующим образом (листинг 24.26).

**Листинг 24.26.** Процедура «ПолеВводаОтделениеОбработкаВыбора()»

&НаКлиенте

Процедура ПолеВводаОтделениеОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)

 Если ПредопределенныйУзел(ВыбранноеЗначение)Тогда

Элементы.СоздатьНачальныйОбраз.Доступность = Ложь;

 Иначе

Элементы.СоздатьНачальныйОбраз.Доступность = Истина;

КонецЕсли;

 КонецПроцедуры

В этой процедуре доступность кнопки *СоздатьНачальныйОбраз* определяется в зависимости от значения функции *ПредопределенныйУзел()*, в которую передается ссылка на выбранный узел (*ВыбранноеЗначение*).

Теперь заполним обработчик команды *СоздатьНачальныйОбраз* следующим образом (листинг 24.27).

**Листинг 24.27.** Обработчик нажатия кнопки «Создать начальный образ»

&НаКлиенте

Процедура СоздатьНачальныйОбраз(Команда)

 Диалог = Новый ДиалогВыбораФайла(РежимДиалогаВыбораФайла.ВыборКаталога);

Диалог.Заголовок = "Укажите каталог информационной базы:";

Если Диалог.Выбрать() Тогда

СоздатьНачальныйОбразНаСервере(ПолеВводаОтделение, Диалог.Каталог);

 Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = "Создание начального образа узла завершено.";

Сообщение.Сообщить();

 КонецЕсли;

 КонецПроцедуры

В начале процедуры мы вызываем диалог выбора каталога, в который будет помещен образ информационной базы, и затем вызываем процедуру *СоздатьНачальныйОбразНаСервере()*, исполняющуюся на сервере без контекста, в которой вызывается метод *СоздатьНачальныйОбраз()* объекта *ПланыОбменаМенеджер*.

Именно этот метод и позволяет нам создать образ для подчиненного узла распределенной информационной базы. В первом параметре метода передается ссылка на узел (реквизит формы *ПолеВводаОтделение*), для которого мы хотим создать начальный образ, а во втором – строка соединения, указывающая информационную базу (листинг 24.28).

**Листинг 24.28.** Процедура «СоздатьНачальныйОбразНаСервере»

&НаСервереБезКонтекста

Процедура СоздатьНачальныйОбразНаСервере(Узел, КаталогСоединения)

 ПланыОбмена.СоздатьНачальныйОбраз(Узел, "File =""" + КаталогСоединения + """");

 КонецПроцедуры

Теперь создадим обработчик команды *Записать изменения*.

Текст обработчика будет выглядеть следующим образом (листинг 24.29).

**Листинг 24.29.** Обработчик нажатия кнопки «Записать изменения»

&НаКлиенте

Процедура ЗаписатьИзменения(Команда)

 Диалог = Новый ДиалогВыбораФайла(РежимДиалогаВыбораФайла.Сохранение);

Диалог.Заголовок = "Укажите файл обмена:";

Если Диалог.Выбрать() Тогда

ЗаписатьИзмененияНаСервере(ПолеВводаОтделение, Диалог.ПолноеИмяФайла);

 Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = "Запись изменений завершена.";

Сообщение.Сообщить();

 КонецЕсли;

 КонецПроцедуры

В начале процедуры мы вызываем диалог ввода имени файла, в который будут записаны изменения, и затем вызываем процедуру *ЗаписатьИзмененияНаСервере()*, исполняющуюся на сервере без контекста. В первом параметре метода передается ссылка на узел (реквизит формы *ПолеВводаОтделение*), для которого будет производиться запись изменений.

В этой процедуре мы создаем объект *ЗаписьXML* для работы с этим файлом.

Затем создаем объект *ЗаписьСообщенияОбмена*, с помощью которого будем делать сообщение обмена. В методе *НачатьЗапись()*, во втором параметре, мы указываем, для какого узла обмена будет создаваться это сообщение.

После этого мы выполняем метод *ЗаписатьИзменения()*объекта *ПланыОбменаМенеджер*, который и записывает изменения, предназначенные для передачи в выбранный узел, в указанное сообщение обмена.

В заключение мы, как обычно, заканчиваем запись сообщения обмена и закрываем файл (листинг 24.30).

**Листинг 24.30.** Процедура «ЗаписатьИзмененияНаСервере»

&НаСервереБезКонтекста

Процедура ЗаписатьИзмененияНаСервере(Узел, ИмяФайла)

 // Создать и проинициализировать объект ЗаписьXML.

ЗаписьXML = Новый ЗаписьXML;

ЗаписьXML.ОткрытьФайл(ИмяФайла);

 // Создать объект ЗаписьСообщенияОбмена и начать запись сообщения.

ЗаписьСообщения = ПланыОбмена.СоздатьЗаписьСообщения();

ЗаписьСообщения.НачатьЗапись(ЗаписьXML, Узел);

 // Записать содержимое тела сообщения обмена данными распределенной ИБ.

ПланыОбмена.ЗаписатьИзменения(ЗаписьСообщения);

 // Закончить запись сообщения и запись XML.

ЗаписьСообщения.ЗакончитьЗапись();

ЗаписьXML.Закрыть();

 КонецПроцедуры

Узнай больше!

Следует отметить, что метод «ЗаписатьИзменения()» позволяет задать максимальное число элементов данных, которые помещаются в сообщение в рамках одной транзакции базы данных. По умолчанию все данные помещаются в сообщение в рамках одной транзакции.

Такой режим является рекомендуемым, так как гарантирует согласованность данных, помещаемых в сообщение.

Но при создании сообщения в многопользовательском режиме могут быть конфликты блокировок между транзакцией, в которой данные помещаются в сообщение, и транзакциями, выполняемыми другими пользователями. Для снижения вероятности возникновения таких конфликтов можно задать значение этого параметра, отличное от значения по умолчанию. Чем меньше значение параметра, тем меньше вероятность конфликта блокировок, но выше вероятность помещения в сообщение несогласованных данных.

Учитывая все вышесказанное, идеальным вариантом является выполнение обмена данными в монопольном режиме. Однако такой вариант не всегда приемлем в силу специфики организации работы конкретных информационных баз.

И последним мы создадим обработчик команды Прочитать изменения.

Текст обработчика будет выглядеть следующим образом (листинг 24.31).

**Листинг 24.31.** Обработчик нажатия кнопки «Прочитать изменения»

&НаКлиенте

Процедура ПрочитатьИзменения(Команда)

 Диалог = Новый ДиалогВыбораФайла(РежимДиалогаВыбораФайла.Открытие);

Диалог.Заголовок = "Укажите файл обмена:";

Если Диалог.Выбрать() Тогда

ПрочитатьИзмененияНаСервере(Диалог.ПолноеИмяФайла);

 Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = "Чтение изменений завершено.";

Сообщение.Сообщить();

 КонецЕсли;

 КонецПроцедуры

В начале процедуры мы снова вызываем диалог выбора имени файла, который будет прочитан, и затем вызываем процедуру *ПрочитатьИзмененияНаСервере()*, исполняющуюся на сервере без контекста.

В этой процедуре мы создаем объект *ЧтениеXML* для работы с этим файлом. Затем создаем объект *ЧтениеСообщенияОбмена* для чтения сообщения, содержащегося в указанном файле.

Затем методом *ПрочитатьИзменения()* объекта *ПланыОбменаМенеджер* мы читаем полученное сообщение.

В заключение процедуры мы завершаем чтение сообщения обмена и закрываем файл (листинг 24.32).

**Листинг 24.32.** Процедура «ПрочитатьИзмененияНаСервере»

&НаСервереБезКонтекста

Процедура ПрочитатьИзмененияНаСервере(ИмяФайла)

 // Создать и проинициализировать объект ЧтениеXML.

ЧтениеXML = Новый ЧтениеXML;

ЧтениеXML.ОткрытьФайл(ИмяФайла);

 // Создать объект ЧтениеСообщенияОбмена и начать чтение сообщения.

ЧтениеСообщения = ПланыОбмена.СоздатьЧтениеСообщения();

ЧтениеСообщения.НачатьЧтение(ЧтениеXML);

 // Прочитать содержимое тела сообщения.

ПланыОбмена.ПрочитатьИзменения(ЧтениеСообщения);

// Закончить чтение сообщения и чтение XML.

ЧтениеСообщения.ЗакончитьЧтение();

ЧтениеXML.Закрыть();

 КонецПроцедуры

В заключение на закладке *Подсистемы* укажем принадлежность обработки *ОбменСОтделениями* к подсистеме *Предприятие*.

Теперь настроим командный интерфейс основного раздела для администратора. Вызовем окно настройки командного интерфейса (*Открыть командный интерфейс основного раздела*) из контекстного меню корня дерева объектов конфигурации. Добавим команды для открытия форм *Общие настройки*, *Обмен данными* и *Обмен с отделениями* в список команд основного раздела и зададим видимость этих форм только для роли *Администратор*(рис. 24.29).

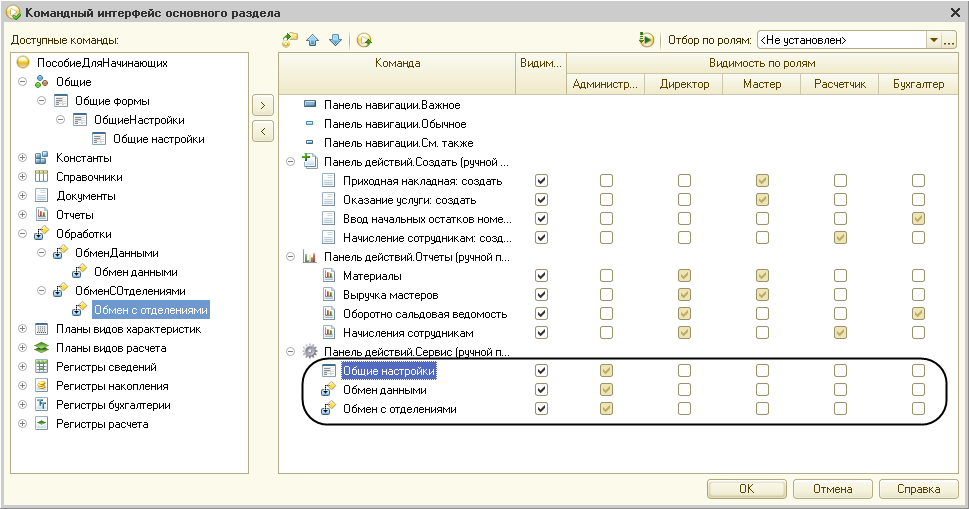


Рисунок - Настройка командного интерфейса основного раздела для администратора

А также расположим формы списка планов обмена *Филиалы* и *Отделения* на начальной странице администратора. Вызовем настройку начальной страницы (*Открыть рабочую область начальной страницы*) из контекстного меню корня дерева объектов конфигурации.

Создадим форму списка плана обмена *Отделения*. Добавим формы списка планов обмена *Филиалы* и *Отделения* в левую часть начальной страницы и зададим видимость этих форм только для роли *Администратор*

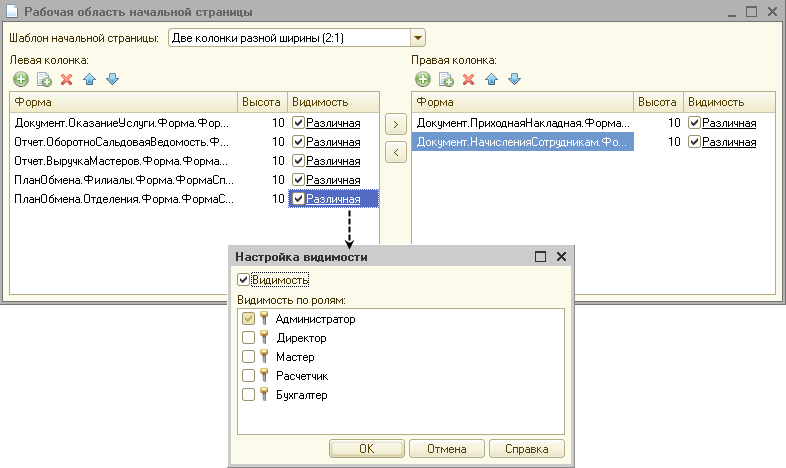


Рисунок - Настройка начальной страницы для администратора

Теперь выполним пример программного обмена данными.

**подсказка**

*При выполнении примера обмена данными нужно правильно задавать узел, в который записываются и от которого принимаются изменения. Например, после изменения данных в центральной базе в списке узлов плана обмена нужно выбрать подчиненный узел, которому будут передаваться данные (например, Отделение), и нажать кнопку Записать изменения. А в базе узла, в который будут загружаться данные, в списке узлов плана обмена нужно выбрать узел, из которого данные были выгружены (например, Центральная база), и нажать кнопку Прочитать изменения.*

Запустим «1С:Предприятие» для пользователя с ролью *Администратор*. Откроется начальная страница приложения (рис.

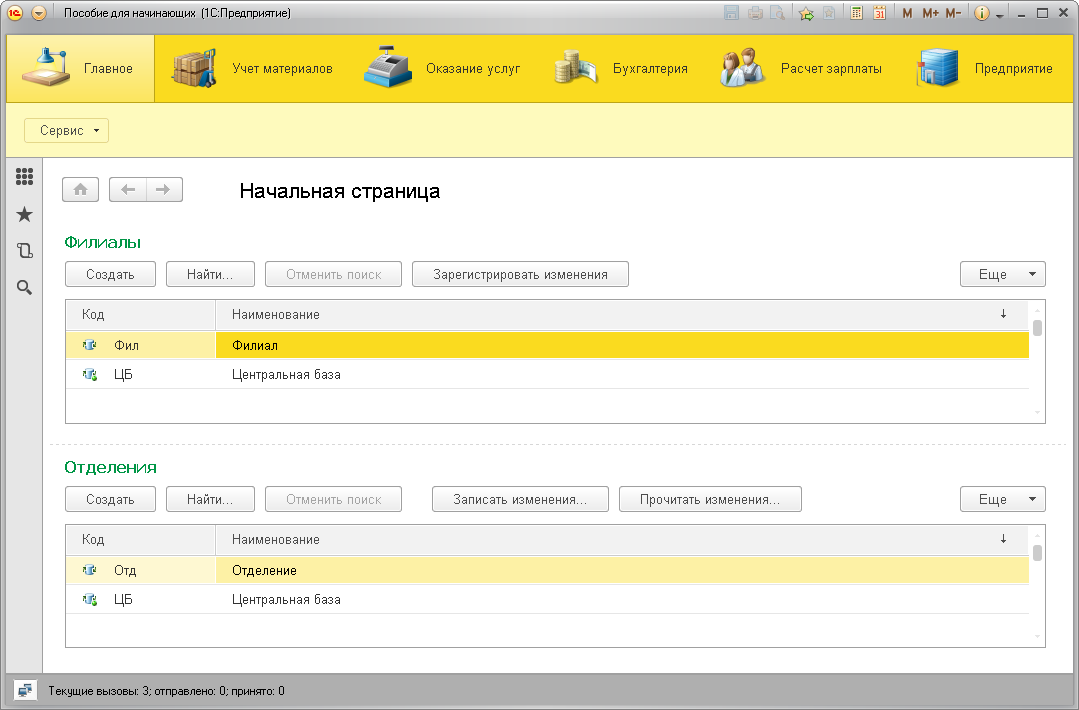


Рисунок - Начальная страница «1С:Предприятия» для администратора

Из командной панели раздела *Главное*, из подменю *Сервис* откроем форму обработки *Обмен с отделениями*. Выберем в поле ввода *Отделение* узел обмена *Отделение*, для которого мы хотим создать начальный образ нашей информационной базы

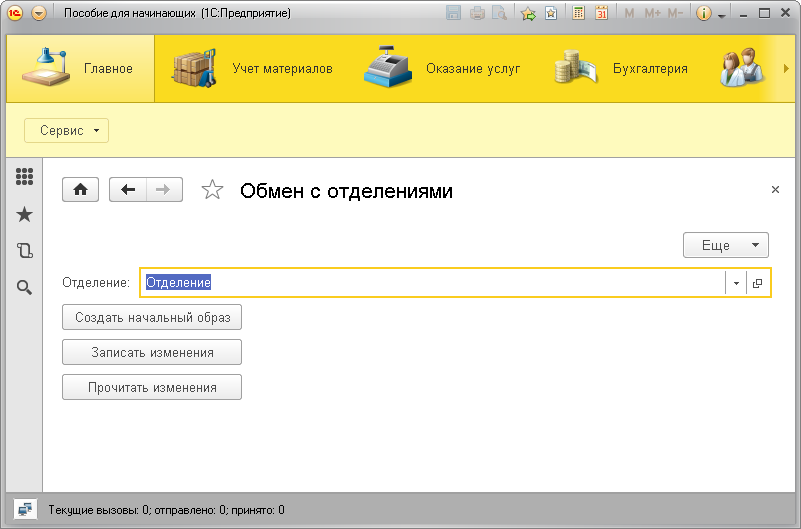


Рисунок - Пример программного обмена с отделениями

Затем выполним команду *Создать начальный образ*. Создадим каталог, куда будет выгружен начальный образ информационной базы, и укажем его в диалоге выбора каталога.

Теперь создадим в центральной базе нового сотрудника. В форме *Обмен с отделениями* выберем в поле ввода *Отделение* узел обмена *Отделение*, в который мы хотим передать изменения. Нажмем кнопку *Записать изменения* и укажем имя файла сообщения с изменениями.

Затем откроем базу отделения (куда был выгружен начальный образ центральной базы) в конфигураторе. Создадим в ней пользователя с ролью *Администратор* и запустим «1С:Предприятие». На начальной странице в форме *Обмен с отделениями* выберем в поле ввода *Отделение* узел обмена *Центральная база*, от которого мы хотим принять изменения. Нажмем кнопку *Прочитать изменения* и укажем имя файла сообщения с изменениями.

Убедимся, что сотрудник центральной базы (с префиксом кода *ЦБ*) присутствует в базе отделения.

Следует лишь сделать несколько заключительных замечаний.

При использовании механизма распределенных информационных баз становятся доступными четыре события объекта встроенного языка *ПланОбменаОбъект.<имя>*, которые позволяют управлять отправкой и приемом данных на уровне отдельных элементов данных:

*ПриОтправкеДанныхГлавному(),*

*ПриОтправкеДанныхПодчиненному(),*

*ПриПолученииДанныхОтГлавного(),*

*ПриПолученииДанныхОтПодчиненного().*

Эти события будут вызываться для каждого элемента данных, включаемого в сообщение.

Работу этих событий можно увидеть, добавив в модуль объекта *План обмена* *Отделения* следующий текст (листинг 24.33).

**Листинг 24.33.** Просмотр работы событий объекта «ПланОбменаОбъект»

Процедура ПриОтправкеДанныхГлавному(ЭлементДанных, ОтправкаЭлемента)

 Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = "ПриОтправкеДанныхГлавному " + ЭлементДанных;

Сообщение.Сообщить();

 КонецПроцедуры

 Процедура ПриОтправкеДанныхПодчиненному(ЭлементДанных, ОтправкаЭлемента)

 Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = "ПриОтправкеДанныхПодчиненному " + ЭлементДанных;

Сообщение.Сообщить();

 КонецПроцедуры

Процедура ПриПолученииДанныхОтГлавного(ЭлементДанных, ПолучениеЭлемента, ОтправкаНазад)

 Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = "ПриПолученииДанныхОтГлавного " + ЭлементДанных;

Сообщение.Сообщить();

 КонецПроцедуры

 Процедура ПриПолученииДанныхОтПодчиненного(ЭлементДанных, ПолучениеЭлемента, ОтправкаНазад)

 Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = "ПриПолученииДанныхОтПодчиненного " + ЭлементДанных;

Сообщение.Сообщить();

 КонецПроцедуры

В первом параметре всех перечисленных событий находится тот элемент данных, для которого вызвано это событие.

Параметр *ОтправкаЭлемента* позволяет управлять тем, какая информация будет помещена в сообщение. Он может принимать три значения:

*Авто* – значение по умолчанию. Указывает на то, что элемент данных будет помещен в сообщение;

*Удалить* – в сообщение будет помещено значение, предназначенное для удаления этого элемента данных;

*Игнорировать* – в сообщение не будет помещено ничего, связанного с этим элементом данных.

Параметр *ПолучениеЭлемента* позволяет указать, будет ли прочитанный элемент данных записан в базу данных или нет. Параметр также может принимать три значения:

*Авто*– значение по умолчанию. Если элемент данных получен от главного узла, он будет записан всегда. Если элемент данных получен от подчиненного узла, он будет записан, только если не зарегистрированы изменения для этого элемента данных;

*Принять* – полученный элемент данных будет записан всегда;

*Игнорировать* – проигнорировать получение элемента данных и ничего не записывать.

Также в событиях получения данных существует третий параметр – *ОтправкаНазад*, имеющий тип *Булево*.

Этот параметр позволяет выполнять принудительную регистрацию изменений для полученного элемента данных в базе-получателе.

Такая необходимость может возникнуть, например, когда при приеме данных от узла-отправителя обнаружено, что полученные данные противоречивы (например, в узле-отправителе была допущена ошибка при изменении данных).

Тогда мы можем проигнорировать присланные изменения и, установив флажок *ОтправкаНазад*, вызвать принудительную регистрацию изменений полученного элемента данных в нашей базе для узла-отправителя.

В результате последующего обмена состояние этого элемента данных в узле-отправителе будет установлено таким же, как и в нашей базе.

Изменение структуры узлов

В заключение следует сказать о том, что механизм распределенных информационных баз содержит программное средство реконфигурирования структуры узлов распределенной базы.

Для этого следует использовать метод *УстановитьГлавныйУзел()* объекта *ПланыОбменаМенеджер*.

В параметре этого метода передается ссылка на узел плана обмена распределенной информационной базы, который устанавливается главным для текущей базы. Также в этом параметре может быть передано значение *Неопределено*, и это приведет к тому, что у текущей информационной базы будет отсутствовать главный узел.

Допустим, необходимо переместить один из подчиненных узлов в корень дерева (рис. 24.33).

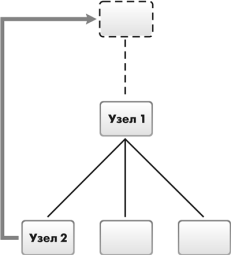


Рисунок --  Реконфигурирование структуры узлов

Для этого нужно выполнить следующие действия (листинг 24.34).

**Листинг 24.34.** Перемещение Узла2 в корень дерева

// В информационной базе Узла2.

ПланыОбменаМенеджер.УстановитьГлавныйУзел(Неопределено);

// В информационной базе Узла1.

ПланыОбменаМенеджер.УстановитьГлавныйУзел(Узел2);

При этом будут удалены все записи регистрации изменений конфигурации Узла1, относящиеся к Узлу2, т. к. передача изменений конфигурации будет возможна теперь только от Узла2 к Узлу1. Записи регистрации изменения данных удалены не будут, так как передача изменений данных будет по-прежнему возможна между этими узлами.

Таким же образом, используя значение параметра метода *Неопределено*, мы можем отключать от дерева отдельную информационную базу или целое поддерево

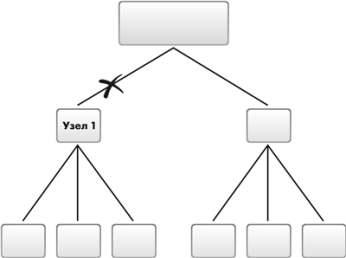


Рисунок - Отключение поддерева от распределенной информационной базы

**Листинг 24.35.** Перемещение Узла2 в корень дерева

// В информационной базе Узла1.

ПланыОбменаМенеджер.УстановитьГлавныйУзел(Неопределено);

Кроме этого, мы можем создавать распределенную информационную базу из отдельных информационных баз с идентичной конфигурацией (рис. 24.35, листинг 24.36).

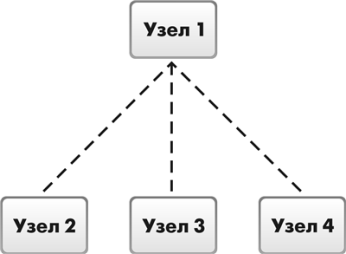


Рисунок - Информационная база из отдельных информационных баз с идентичной конфигурацией

**Листинг 24.36.** Создание распределенной информационной базы из баз с идентичной конфигурацией

// В информационных базах Узла2, Узла3 и Узла4.

ПланыОбменаМенеджер.УстановитьГлавныйУзел(Узел1);

# Практическая работа № 23 Подбор с использованием множественного выбора

**Цель:** Подбор с использованием множественного выбора

**Норма времени**: 4 часа

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

Множественный подбор

В режиме «Конфигуратор»

При множественном подборе форма справочника будет открыта до тех пор, пока пользователь не закроет ее интерактивно или не будет вызван метод формы *Закрыть()*.

В форме документа *ПриходнаяНакладная*, в обработчике команды *Подбор* заменим прежний текст новым (листинг 26.3).

**Листинг 26.3.** Обработчик нажатия кнопки «Подбор»

&НаКлиенте

Процедура Подбор(Команда)

 ПараметрыФормы = Новый Структура("ЗакрыватьПриВыборе", Ложь);

ОткрытьФорму("Справочник.Номенклатура.ФормаВыбора", ПараметрыФормы, Элементы.Материалы);

 КонецПроцедуры

При открытии формы мы используем ее *параметры*.

Параметры формы нужны для того, чтобы открыть форму в некотором нужном нам состоянии. Параметры формы представляют собой структуру. Каждый элемент этой структуры описывает один параметр формы. Ключ элемента – это имя параметра формы.

Такую структуру мы передаем в метод *ОткрытьФорму()* вторым параметром (переменная *ПараметрыФормы*).

Предварительно мы эту структуру формируем. В ней у нас всего один элемент с ключом *ЗакрыватьПриВыборе*.

Таким образом, передавая эту структуру в метод *ОткрытьФорму(),* мы устанавливаем параметр открываемой формы *ЗакрыватьПриВыборе* в значение *Ложь*.

Это значит, что созданная форма выбора после двойного щелчка мыши на номенклатуре закрываться не будет.

Множественный подбор

В режиме «Конфигуратор»

При множественном подборе форма справочника будет открыта до тех пор, пока пользователь не закроет ее интерактивно или не будет вызван метод формы *Закрыть()*.

В форме документа *ПриходнаяНакладная*, в обработчике команды *Подбор* заменим прежний текст новым (листинг 26.3).

**Листинг 26.3.** Обработчик нажатия кнопки «Подбор»

&НаКлиенте

Процедура Подбор(Команда)

 ПараметрыФормы = Новый Структура("ЗакрыватьПриВыборе", Ложь);

ОткрытьФорму("Справочник.Номенклатура.ФормаВыбора", ПараметрыФормы, Элементы.Материалы);

КонецПроцедуры

При открытии формы мы используем ее *параметры*.

Параметры формы нужны для того, чтобы открыть форму в некотором нужном нам состоянии. Параметры формы представляют собой структуру. Каждый элемент этой структуры описывает один параметр формы. Ключ элемента – это имя параметра формы.

Такую структуру мы передаем в метод *ОткрытьФорму()* вторым параметром (переменная *ПараметрыФормы*).

Предварительно мы эту структуру формируем. В ней у нас всего один элемент с ключом *ЗакрыватьПриВыборе*.

Таким образом, передавая эту структуру в метод *ОткрытьФорму(),* мы устанавливаем параметр открываемой формы *ЗакрыватьПриВыборе* в значение *Ложь*.

Это значит, что созданная форма выбора после двойного щелчка мыши на номенклатуре закрываться не будет.

В режиме «1С:Предприятие»

Запустим «1С:Предприятие» в режиме отладки.

Перейдем в раздел *Учет материалов*, откроем список приходных накладных и создадим новую приходную накладную.

В командной панели списка материалов нажмем кнопку *Подбор.*Двойным щелчком мыши подберем в накладную один материал, затем другой материал. Перейдем в группу *Услуги* и подберем несколько услуг.

Когда все желаемые товары и услуги будут выбраны, закроем окно с формой выбора.

# Практическая работа № 24 Получение объектов, введённых на основании

**Цель:** Получение объектов

**Норма времени**: 4 часа

**Методическое обеспечение**: методические указания к практической работе.

Ход работы

Получение объектов, введенных на основании

Поскольку задача получения всех объектов, введенных на основании какого-либо другого объекта, чаще всего возникает при анализе документов, мы рассмотрим применение описанной выше методики на примере получения списка документов, введенных на основании элемента справочника *Клиенты*.

В режиме «Конфигуратор»

Раскроем ветвь *Общие* и создадим новый объект конфигурации *КритерийОтбора* с именем *ОказаниеУслуги.*

На закладке *Данные* выберем тип используемого критерия – *СправочникСсылка.Клиенты*.

На закладке *Состав* в качестве объектов, входящих в критерий, выберем реквизит *ОбъектОснование* документа *ОказаниеУслуги*

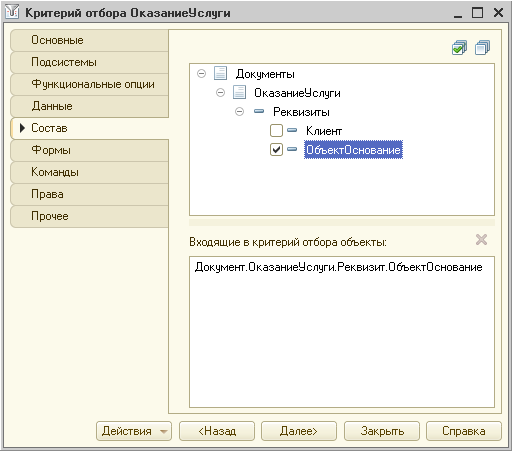


Рисунок - Состав критерия отбора «Оказание Услуги»

После этого в панели навигации формы элемента справочника *Клиенты* в группе *Перейти* появится команда для открытия критерия отбора.

Создадим эту форму и на закладке *Командный интерфейс* установим видимость команды *Оказание услуги*

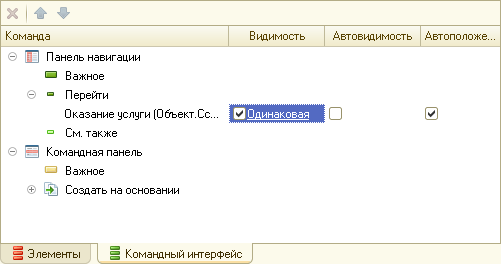


Рисунок - Команда открытия критерия отбора из формы элемента справочника «Клиенты»

В режиме «1С:Предприятие»

Запустим «1С:Предприятие» в режиме отладки и проверим работу критерия отбора.

В форме списка клиентов выделим клиента *Роман*, на основании которого мы создавали документы *Оказание услуги*.

Откроем форму этого клиента. В панели навигации появилась команда *Оказание услуги* для открытия формы списка созданного нами критерия отбора с установленным отбором по открытому элементу справочника *Клиенты*.

Выполним эту команду (рис. 26.10).

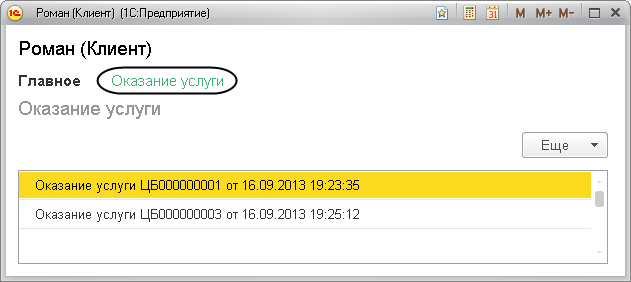


Рисунок - Открытие списка критерия отбора «Оказание Услуги» с отбором по клиенту

Мы видим в этом списке документы *Оказание услуги*, созданные на основании клиента *Роман*. К содержимому документа можно перейти, нажав соответствующую ссылку в списке документов.

# Практическая работа № 25 Разработка форм

**Цель:** Разработка форм

**Норма времени**: 2 часа

**Методическое обеспечение**: методические указания к практической работе.

Данные и элементы формы

Важной особенностью платформы «1С:Предприятие» является механизм представления данных в формах. Ключевым моментом здесь является то, что принадлежность формы к тому или иному объекту конфигурации никоим образом не определяет состав данных, которые форма будет отображать.

Например, можно создать общую форму, которая не будет подчинена ни одному из объектов конфигурации, но которая, в зависимости от содержимого, будет либо отображать список справочника, либо позволять редактировать документ и т. п. Однако такую форму уже нельзя будет назначить основной для выполнения определенных действий.

Форма сама по себе и ее элементы обособлены от объектов конфигурации. Для того чтобы форма отображала какие-либо данные, необходимо задать связь самой формы и ее элементов с данными. Если связь элементов формы с данными не задана, то элементы вообще не будут отображены в форме (кроме элементов оформления формы).

При использовании конструктора форм конфигуратор создает такие связи автоматически. Если разработчик создает форму вручную, он может определить эти связи путем задания свойств формы и ее элементов

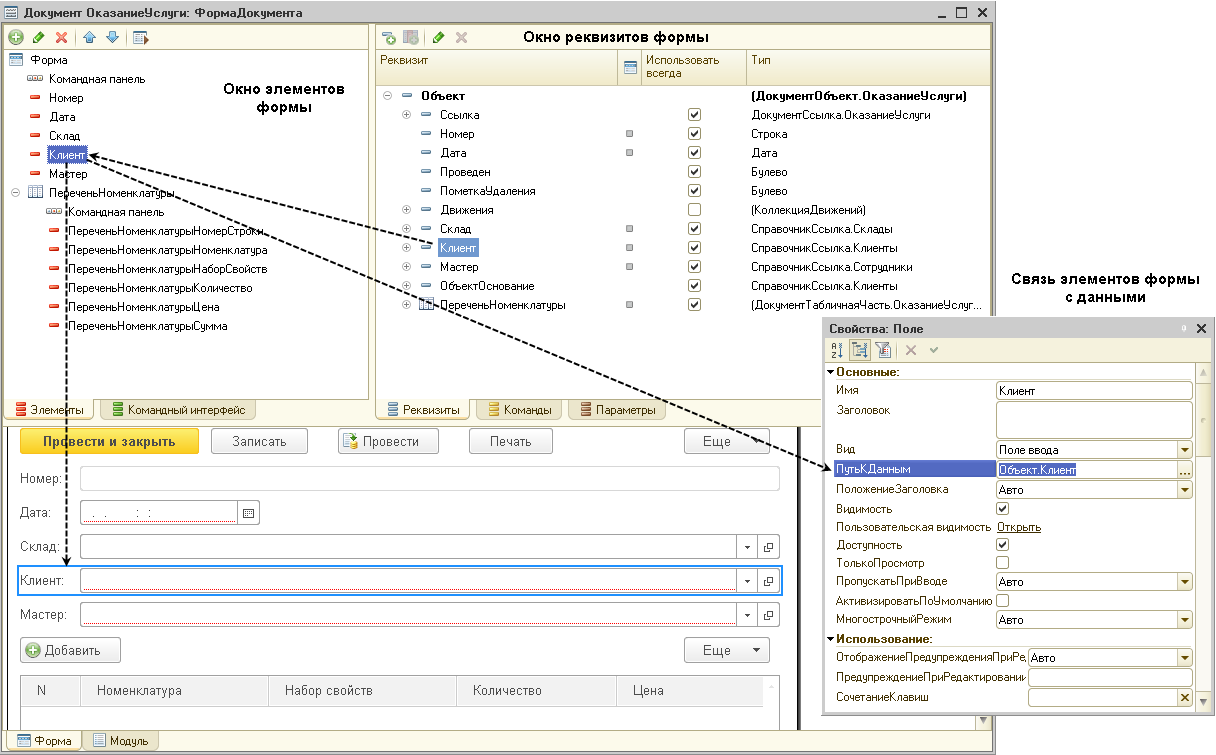


Рисунок - Связь элементов формы с данными

Связь элементов формы с данными, которые они должны отображать, задается в свойстве *ПутьКДанным*.

Связь формы и ее элементов с данными осуществляется при помощи реквизитов формы. Список существующих реквизитов формы доступен на закладке *Реквизиты* окна редактора формы.

Среди всех реквизитов формы, как правило, существует один основной реквизит (он выделен жирным шрифтом). Он определяет источник данных для формы в целом. От типа значения основного реквизита формы зависит не только то, какие данные будут отображены в элементах формы, но и поведение самой формы.

Например, если основному реквизиту формы указать тип значения *ДокументОбъект.ПриходнаяНакладная*, то при закрытии формы программа будет запрашивать подтверждение записи и проведения документа. Если же основному реквизиту формы указать тип значения *СправочникОбъект.Клиенты*, то подобного подтверждения при закрытии формы возникать не будет.

Похожее влияние источники данных оказывают и на элементы формы.

Например, состав колонок таблицы, источником данных которой является реквизит формы с типом значения *ДинамическийСписок*, будет различным в зависимости от того, какой объект используется в качестве основной таблицы этого динамического списка (например, *РегистрНакопления.ОстаткиНоменклатуры*или *Справочник.Номенклатура*).

То же самое справедливо и для элемента формы *Командная панель*. При установленном свойстве командной панели *Автозаполнение* смена источника данных (а точнее говоря, источника действий) будет приводить к изменению состава команд, которые отображает командная панель.

Возможность связать форму и ее элементы с различными данными является причиной того, что у формы и ее элементов существует несколько *расширений*.

*Расширение* представляет собой набор дополнительных свойств, методов и событий, появляющихся у формы или у элемента формы. Наличие того или иного расширения определяется либо типом данных, которые отображает форма/элемент, либо расположением элемента формы в других ее элементах.

Чтобы подробнее познакомиться с этим механизмом, создадим основную форму списка справочника *Номенклатура*. При этом в конструкторе формы мы не будем сразу нажимать кнопку *Готово*, как делали раньше.

Нажмем кнопку *Далее >* и кроме полей *Наименование* и *Код* включим в состав таблицы *Список* еще одно поле – *ВидНоменклатуры*. И затем уже нажмем *Готово*

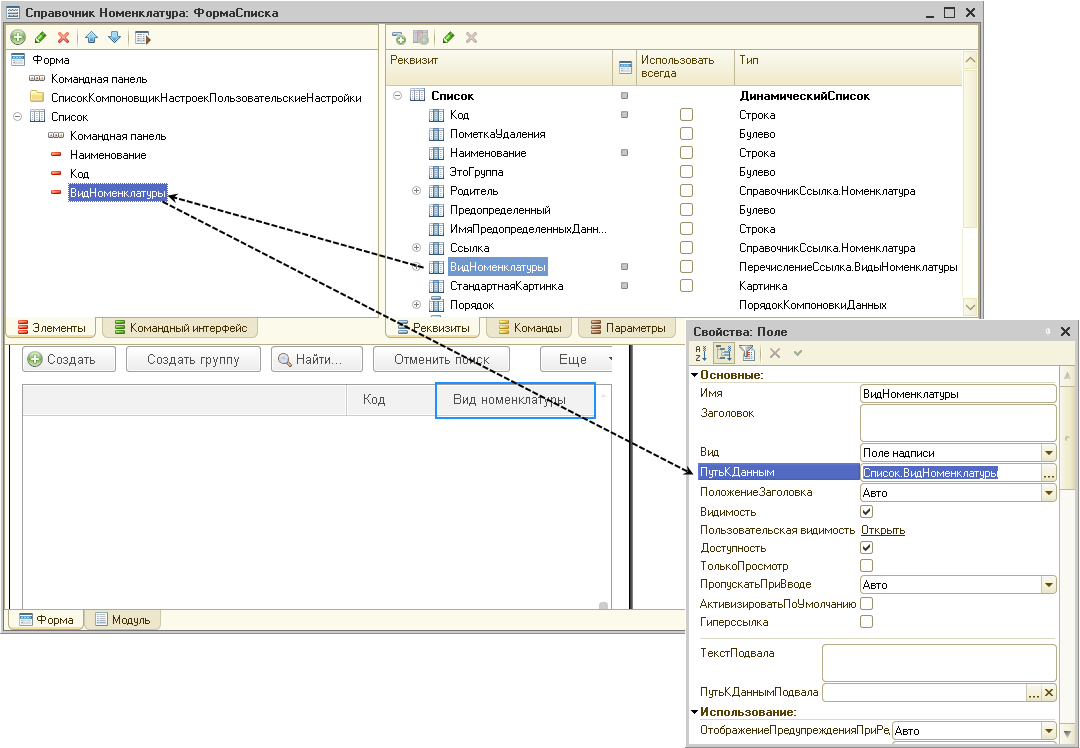


Рисунок - Форма списка справочника «Номенклатура»

Итак, с механизмом расширений мы будем знакомиться на примере поля *ВидНоменклатуры*, расположенного в таблице *Список* формы списка справочника *Номенклатура*.

Сама форма отображает данные объекта *ДинамическийСписок* (основной реквизит формы *Список* имеет тип *ДинамическийСписок*, )

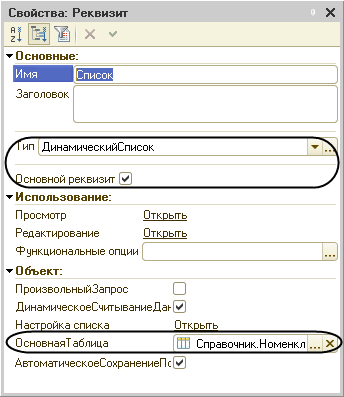


Рисунок - Основной реквизит формы списка

Поэтому к свойствам, методам и событиям объекта встроенного языка *УправляемаяФорма* добавляется *Расширение динамического списка* (рис. 27.4).

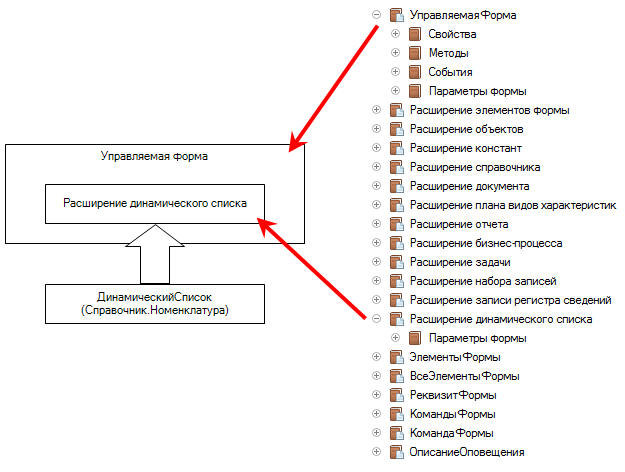


Рисунок - Контекст формы дополняется контекстом расширения динамического списка

В результате этого у формы появляются такие параметры, как *ТекущаяСтрока, Отбор* и т. п.

Теперь посмотрим на таблицу *Список*.

Поскольку в таблице отображается динамический список, то к свойствам, методам и событиям объекта встроенного языка *Таблица Формы* добавляется *Расширение динамического списка*

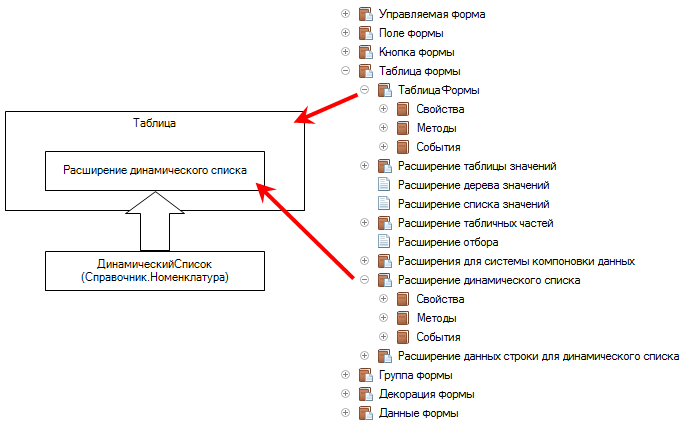


Рисунок - Контекст таблицы дополняется контекстом расширения динамического списка

В результате у таблицы *Список* появляются такие свойства, как *АвтоОбновление*, *ОтображатьКорень* и т. д.

И в заключение посмотрим на поле *ВидНоменклатуры*.

Это поле связано с реквизитом типа *ПеречислениеСсылка.ВидыНоменклатуры* и является полем ввода.

Поэтому к свойствам, методам и событиям объекта встроенного языка *ПолеФормы* добавляется *Расширение поля ввода*

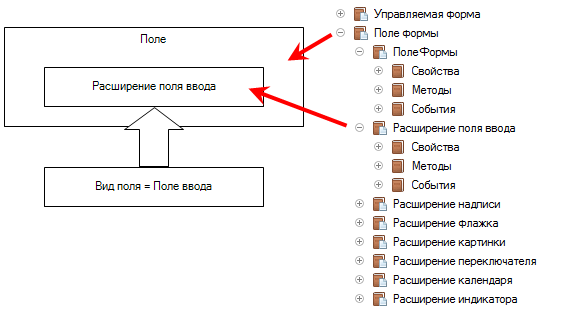


Рисунок -Контекст поля формы дополняется контекстом расширения поля ввода

В результате у поля *ВидНоменклатуры* появляются такие свойства, как *БыстрыйВыбор*, *ВыделенныйТекст* и т. д.

пы данных формы

В управляемой форме можно выделить следующие категории типов, с которыми она работает:

Типы встроенного языка, предназначенные для использования как в управляемых формах, так и вне них. Например, *Число, СправочникСсылка.<имя>, ГрафическаяСхема, ТабличныйДокумент* и т. д.

Типы встроенного языка, предназначенные исключительно для того, чтобы представить в форме данные прикладных объектов (справочников, документов и т. д.). Это такие типы, как *ДанныеФормыСтруктура*, *ДанныеФормыКоллекция* и другие.

Отдельно следует упомянуть тип *ДинамическийСписок*, который используется в управляемых формах для отображения списков прикладных объектов.

Все типы прикладных объектов (такие как *СправочникОбъект* и т. д.) не существуют на стороне тонкого и веб-клиентов, они существуют только на сервере. Однако данные этих объектов нужно отображать в управляемых формах.

Поэтому для представления в форме данных этих прикладных типов введены специальные типы данных, предназначенные для работы именно в управляемых формах. Используются следующие типы данных:

*ДанныеФормыСтруктура* – содержит набор свойств произвольного типа. Свойствами могут быть другие структуры, коллекции или структуры с коллекциями. Таким типом представляется, например, в форме *СправочникОбъект*.

*ДанныеФормыКоллекция* – это список типизированных значений, похожий на массив. Доступ к элементу коллекции осуществляется по индексу или по идентификатору. Доступ по идентификатору может отсутствовать в некоторых случаях. Это обусловлено типом прикладного объекта, который представлен этой коллекцией. Идентификатором может быть любое целое число. Таким типом представляется, например, в форме табличная часть.

*ДанныеФормыСтруктураСКоллекцией* – это объект, который представлен в виде структуры и коллекции одновременно. С ним можно обращаться как с любой из этих сущностей. Таким типом представляется, например, в форме набор записей.

*ДанныеФормыДерево* – объект предназначен для хранения иерархических данных.

Прикладной объект представлен либо одним, либо несколькими элементами данных формы. В общем виде иерархия и состав данных формы зависят от сложности и взаимосвязи прикладных объектов управляемой формы.

Например, документ, содержащий табличную часть, будет представлен объектом типа *ДанныеФормыСтруктура* (собственно документ), которому подчинен объект типа *ДанныеФормыКоллекция* (табличная часть документа).

ВНИМАНИЕ!

*Во время разработки конфигурации важно помнить, что прикладные объекты доступны только на сервере, в то время как объектами данных форм можно пользоваться и на сервере, и на клиенте.*

Фактически можно сказать, что данные формы – это унифицированное представление данных различных прикладных объектов, с которыми форма работает единообразно и которые присутствуют и на сервере, и на клиенте.

В редакторе формы (у реквизитов формы) вместо имен этих типов обычно отображаются те прикладные типы, данные которых содержит реквизит.

Например, если реквизит *Объект* содержит данные элемента справочника *Клиенты*, то в колонке *Тип* отображается ненастоящий тип этого реквизита формы – *ДанныеФормыСтруктура*, а тип прикладного объекта, данные которого содержатся в этом реквизите – *СправочникОбъект.Клиенты.*Причем чтобы было понятно, что это «ненастоящий тип» реквизита, тип прикладного объекта показывается в круглых скобках.

Таким образом форма содержит некоторую «проекцию» данных прикладных объектов в виде своих собственных типов данных и автоматически выполняет преобразование между ними при необходимости.

Однако если разработчик конфигурации реализует свой алгоритм обработки данных, то преобразование данных (из специализированных типов в прикладные и обратно) он должен выполнять самостоятельно.

Для конвертирования прикладных объектов в данные формы и обратно существует набор глобальных методов:

*ЗначениеВДанныеФормы()*– преобразует объект прикладного типа в данные формы;

*ДанныеФормыВЗначение()*– преобразует данные формы в объект прикладного типа.

Аналогичные методы, предназначенные для конвертирования значений реквизитов формы в прикладные объекты и обратно, существуют и у самой управляемой формы:

*ЗначениеВРеквизитФормы()* – преобразует объект прикладного типа в реквизит управляемой формы;

*РеквизитФормыВЗначение()* – преобразует реквизит управляемой формы в значение прикладного типа.

Методы, работающие с прикладными объектами, доступны только в серверных процедурах формы.

При выполнении стандартных действий формы с основным реквизитом (открытие формы, выполнение стандартной команды *Записать* и т. д.) преобразование выполняется автоматически.

Приведем пример преобразования данных, которое может потребоваться в собственных алгоритмах.

Например, у нас есть особенная форма, в которой в качестве одного из реквизитов (*ТоварДляМодификации)* используются данные элемента справочника *Товары*. При создании формы на сервере мы по некоторому алгоритму определяем, какой именно это товар, и читаем его данные в реквизит формы. При этом используется преобразование данных *ЗначениеВДанныеФормы(),* листинг 27.1.

**Листинг 27.1.** Пример преобразования данных прикладных объектов в данные формы

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

 ОбъектТовар = Справочники.Товары.НайтиПоНаименованию("Кофейник").ПолучитьОбъект();

ЗначениеВДанныеФормы(ОбъектТовар, ТоварДляМодификации);

 КонецПроцедуры

 &НаКлиенте

Процедура Записать()

 ЗаписатьНаСервере();

 КонецПроцедуры

 &НаСервере

Процедура ЗаписатьНаСервере()

 ОбъектТовар = ДанныеФормыВЗначение(ТоварДляМодификации, Тип("СправочникОбъект.Товары"));

ОбъектТовар.Записать();

 КонецПроцедуры

В некоторый момент работы формы мы решаем, что измененные данные нашего товара необходимо записать в базу данных, и тогда выполняем обратное преобразование данных формы в прикладной объект (*ДанныеФормыВЗначение()*) и записываем его.

Как мы уже упомянули, у формы также есть методы, позволяющие преобразовать прикладные данные в реквизит формы и наоборот.

Использование данных методов обычно удобнее, так как они имеют, например, информацию о типе реквизита формы. Кроме этого, метод *РеквизитФормыВЗначение()* выполняет установку соответствия данных формы и объекта, которая используется при формировании сообщений.

Приведем пример использования этих методов. В серверной процедуре формы мы получаем прикладной объект из реквизита формы и выполняем метод этого прикладного объекта *Пересчитать()*. Затем данные объекта, измененные в результате пересчета, преобразуем обратно в реквизит формы (листинг 27.2).

**Листинг 27.2.** Пример преобразования данных прикладных объектов в данные формы

&НаСервере

Процедура ПересчитатьНаСервере()

 // Преобразует реквизит Объект в прикладной объект.

Документ = РеквизитФормыВЗначение("Объект");

 // Выполняет пересчет методом, определенным в модуле документа.

Документ.Пересчитать();

 // Преобразует прикладной объект обратно в реквизит.

ЗначениеВРеквизитФормы(Документ, "Объект");

 КонецПроцедуры

# Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

1. Радченко, М. Г. 1С: Предприятие 8.2. Практическое пособие разработчика. Примеры и типовые приёмы [Текст]/ М. Г. Радченко, Е. Ю. Хрусталёва. – М.: ООО «1С: Паблишинг», 2009. – 874 с.

**Интернет-ресурсы:**

1. Википедия – свободная энциклопедия [Электронный ресурс] – Режим доступа:[http://ru.wikipedia.org](http://ru.wikipedia.org/) **–**;
2. ИНТУИТ. Национальный открытый университет. Проект [Издательства «Открытые Системы](http://www.osp.ru/)«. [Электронный ресурс] – Режим доступа: http://Intuit.ru
3. Научная электронная библиотека; [Электронный ресурс] – Режим доступа: [www.elibrary.ru](http://www.elibrary.ru/) –
4. Новая электронная библиотека[Электронный ресурс] – Режим доступа: [www.newlibrary.ru](http://www.newlibrary.ru/) -;
5. Общероссийский математический портал[Электронный ресурс] – Режим доступа: [www.mathnet.ru](http://www.mathnet.ru/) –;
6. Федеральный портал российского образования[Электронный ресурс] – Режим доступа: [www.edu.ru](http://www.edu.ru/) –;
7. Электронная библиотека учебных материалов[Электронный ресурс] – Режим доступа: [www.nehudlit.ru](http://www.nehudlit.ru/) –.