

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Минцаев Магомед Шаварович

Должность: Ректор

Дата подписания: 04.10.2023 17:17:17

Уникальный программный ключ:

236bcc35c296f119d6aafdc22876b21db52dbc07971e86865e5825f9fa4704ee

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ГРОЗНЕНСКИЙ ГОСУДАРСТВЕННЫЙ НЕФТЯНОЙ**

**ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**имени академика М.Д. Миллионщикова**

Кафедра «Информационные технологии»

**Н.А. Моисеенко**

**Д.А. Вахаева**

**Методические рекомендации к лабораторным работам по дисциплине**

**«Программирование в 1С»**

для студентов, обучающихся по направлению подготовки

09.03.02 «Информационные системы и технологии»

Бакалавр

Грозный 2021

### **Составители:**

Моисеенко Н.А., к.п.н., доцент кафедры «Информационные технологии»

Вахаева Д.А., старший преподаватель кафедры «Информационные технологии»

### **Рецензент:**

Э.Д. Алисултанова, доктор педагогических наук, кандидат физико-математических наук, профессор, директор Института прикладных информационных технологий, заведующая кафедрой «Информатика и вычислительная техника»

Методические указания предназначены для бакалавров по направлению подготовки 09.03.02 Информационные системы и технологии института прикладных информационных технологий.

Методические рекомендации рассмотрены и утверждены на заседании кафедры «Информационные технологии»

Протокол № 4 от 23.11.2022г

Рекомендовано к изданию редакционно-издательским советом ГГНТУ.

© Федеральное государственное бюджетное образовательное учреждение высшего образования «Грозненский государственный нефтяной технический университет имени академика М.Д. Миллионщикова», 2022

## Оглавление

Лабораторная работа №1. Создание новой информационной базы .....	4
Лабораторная работа №2. Справочники .....	13
Лабораторная работа №3. Предопределенные элементы и Реквизиты справочника .....	18
Лабораторная работа №4. Табличная часть и Подчинение .....	25
Лабораторная работа №5. Форма. Форма списка, форма элемента.....	29
Лабораторная работа №.6 Перечисление. Элементы управления формы .....	33
Лабораторная работа №.7 Документ. Формы документа .....	40
Лабораторная работа №.8 Обработчик события. Функциональные опции ....	48
Лабораторная работа №.9 Подсистемы и интерфейс. Регистр сведений.....	53
Лабораторная работа №.10 Запрос .....	67
Лабораторная работа №.11 Отчет. компоновка данных.....	75
Библиографический список .....	85

## Лабораторная работа №1. Создание новой информационной базы

Система программ "1С:Предприятие 8.2" включает в себя платформу и прикладные решения, разработанные на ее основе, для автоматизации деятельности организаций и частных лиц. Сама платформа не является программным продуктом для использования конечными пользователями, которые работают с прикладным решением (конфигурацией), разработанным на данной платформе. Соответственно, система имеет два режима работы: 1С:Предприятие (пользовательский режим) и Конфигуратор (предназначен для разработчиков прикладных решений и администраторов информационных баз данных). Поскольку именно в режиме конфигуратора и происходит модификация существующих и создание новых конфигураций, то именно этот режим и будет использован нами в первую очередь.

### Задание:

#### 1. Создание новой информационной базы

Первое, что нам необходимо сделать для решения предлагаемой учебной задачи, это создать новую информационную базу. Запускаем систему "1С:Предприятие". На экран будет выведен диалог "Запуск 1С:Предприятия" (Рис.1.1).

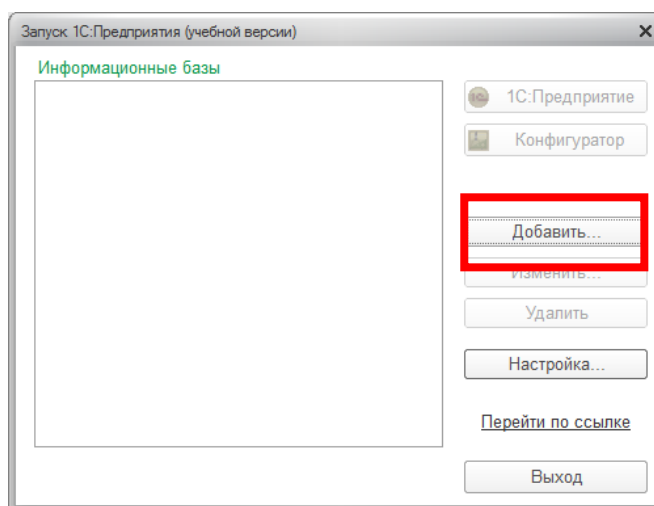


Рис.1.1

Создадим новую информационную базу, нажав кнопку "Добавить". Далее следуем инструкциям, появляющимся на экране (Рис. 1.2).

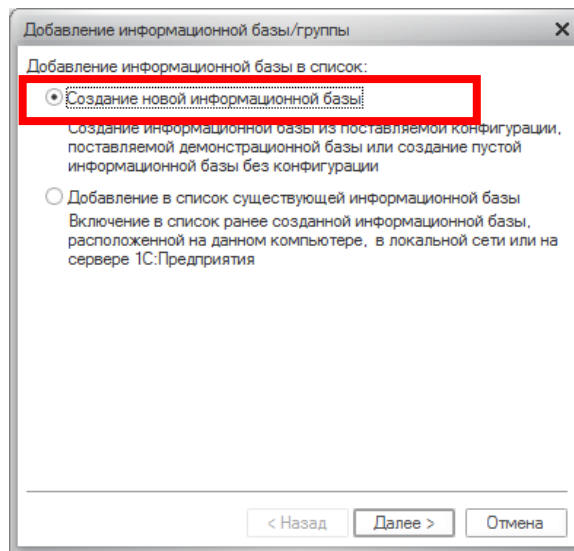


Рис.1.2

Далее необходимо указать, что создается новая пустая база, а не типовая из шаблона, для чего выбираем нижний вариант (Рис. 1.3).

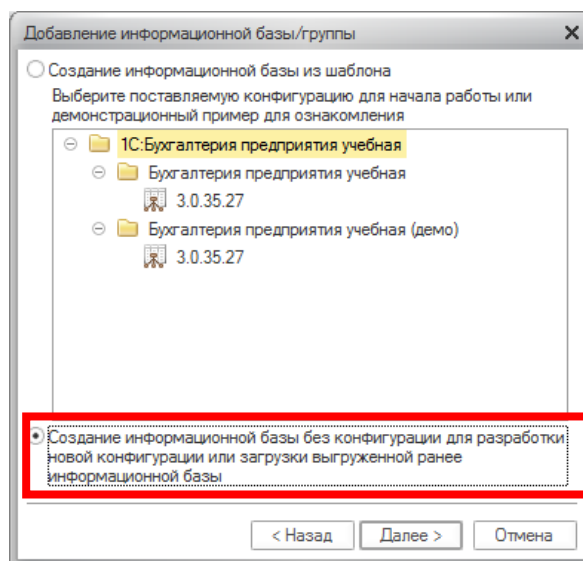


Рис.1.3

Следующее, что необходимо сделать, это дать имя создаваемой базе (на ваше усмотрение). (Рис.1.4)

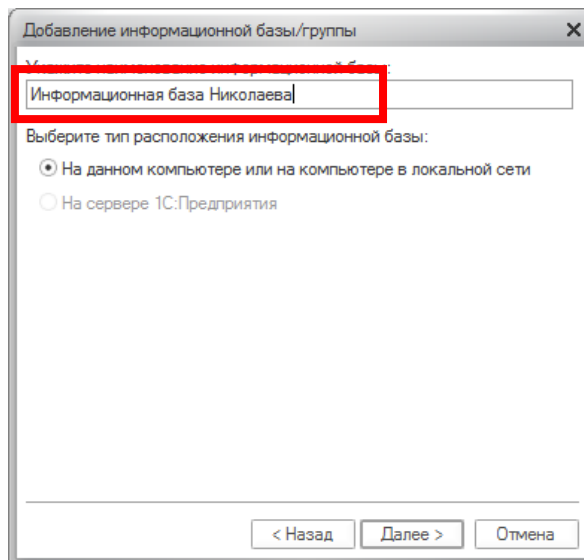


Рис.1.4

После этого необходимо указать, где будут храниться данные, содержащиеся в нашей информационной базе. (Рис.1.5)

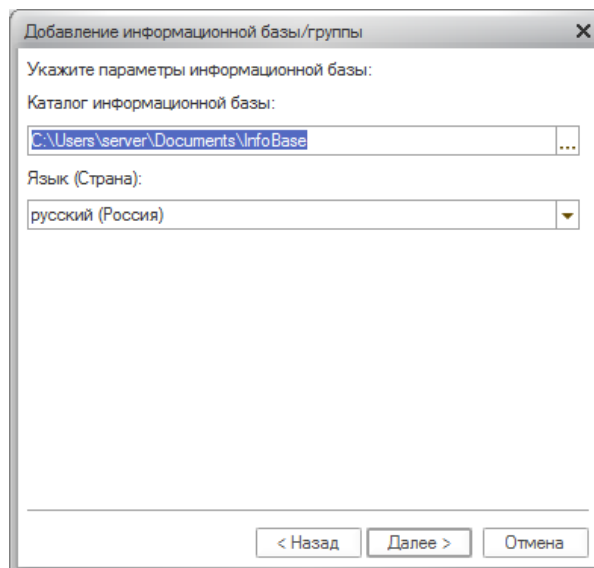


Рис.1.5

В следующем окне оставляем параметры запуска по умолчанию.

Теперь после запуска "1С:Предприятия" у нас появится возможность работать с новой информационной базой (Рис. 1.6), причем в обоих режимах, выбор которых будет осуществляться нажатием на соответствующие кнопки "1С:Предприятие" или "Конфигуратор".

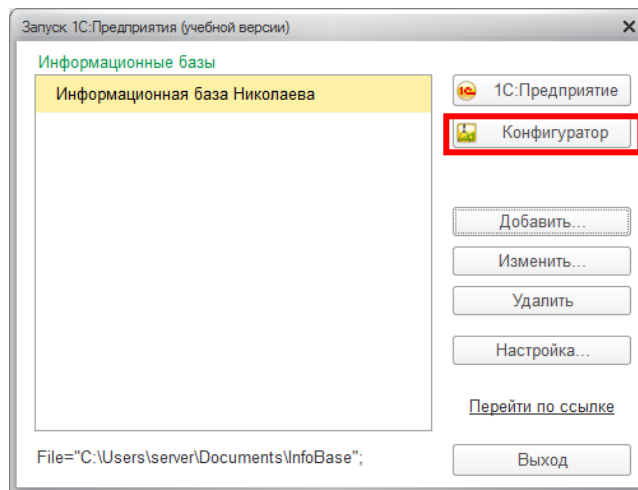


Рис.1.6

Так как в нашей базе отсутствует структура для хранения какой-либо информации, то необходимо создать такую структуру, для чего необходимо открыть информационную базу в режиме конфигуратора. В открывшемся окне откроем окно "Конфигурация". Для этого выберем пункт меню *Конфигурация-Открыть конфигурацию* или на панели инструментов нажать кнопку *Открыть конфигурацию* (Рис. 1.7).

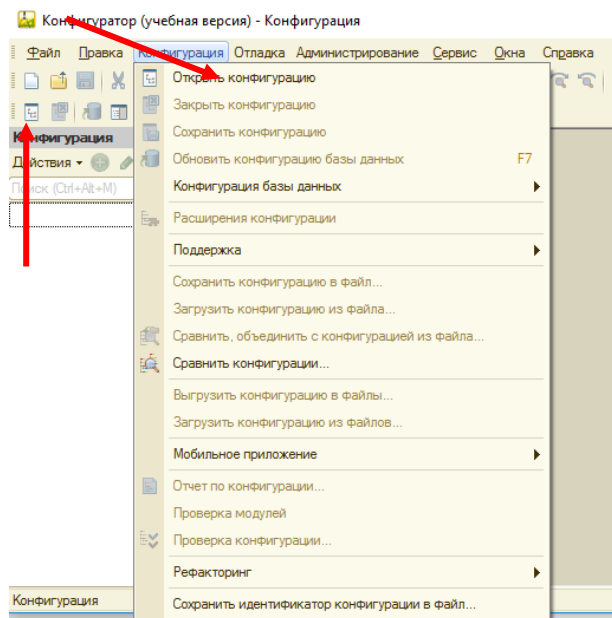


Рис.1.7

Данное окно называется окном конфигурации, а содержащаяся в нем информация – деревом конфигурации. (Рис.1.8)

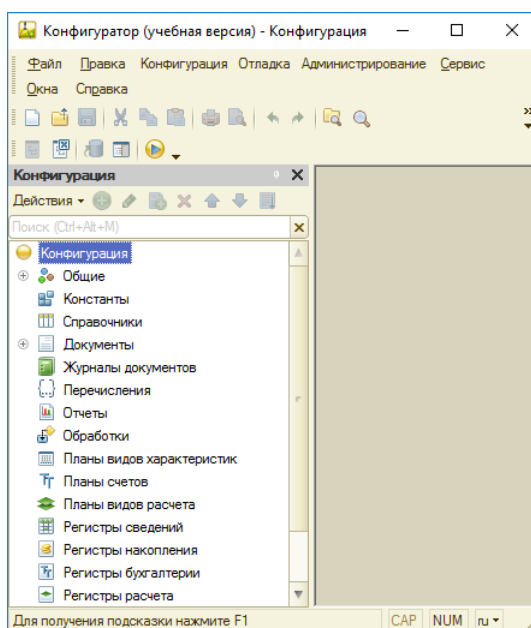


Рис.1.8

## 2. Константы

*Объект Константа предназначен для хранения постоянной или условно-постоянной информации. Важным является то, что одна константа предназначена для хранения только одного значения. В системе может быть произвольное количество констант. Константы создаются в конфигураторе, но заполняются непосредственными значениями в пользовательском режиме.*

Поскольку константа предназначена для хранения одного значения, потребуется создать три константы: с информацией о генеральном директоре нашей фирмы, ее наименовании и дате регистрации.

Для создания новой константы в окне дерева объектов конфигурации правой кнопкой мышки на ветке *Константы* вызываем контекстное меню, в котором выбираем пункт *Добавить*. (Рис . 1.9)



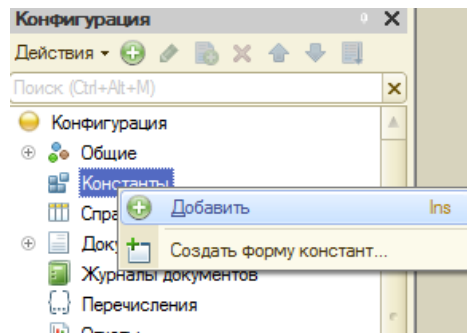


Рис.1.9

Далее откроется окно, называемое *Палитра свойств*, в которой будут отражены свойства константы. (Рис. 1.10)

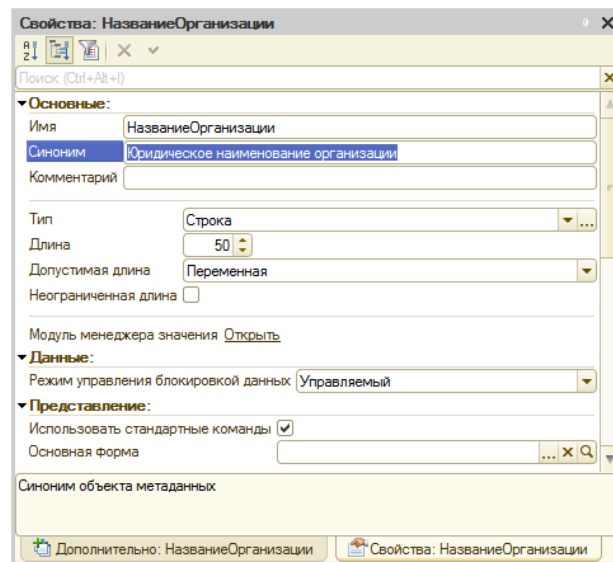


Рис.1.10

Нам потребуется заполнить следующие поля:

- **Имя** – идентификатор константы. Поскольку в системе может существовать произвольное количество констант, то необходимо однозначно идентифицировать каждую из них.
- **Синоним** – удобное для пользователя название.
- **Тип данных** – тип данных хранимых в константе. Для строкового типа данных необходимо указать предельную длину строки, а также способ ее хранения. В системе предусмотрена возможность хранить строки неограниченной длины.

- Допустимая длина отражает системный способ реализации хранения строк. В нашем случае будем использовать строку длиной 50 с переменной допустимой длиной.

Таким образом, мы создали структуру данных, которая позволит хранить необходимую нам информацию, но не определили, как именно называется наша организация. Для этого необходимо перейти в пользовательский режим (режим "1С: Предприятие"), выбираем пункт меню *Сервис - 1С: Предприятие* или на панели нажимаем на кнопку *Начать отладку* (Рис.1.11) и последовательно подтверждаем необходимость обновления информационной базы (Рис. 1.12 - Рис. 1.13).

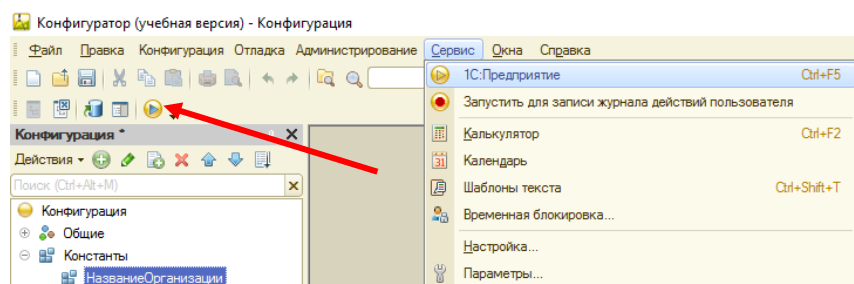


Рис.1.11

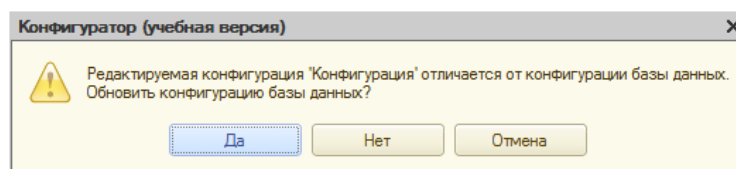


Рис.1.12

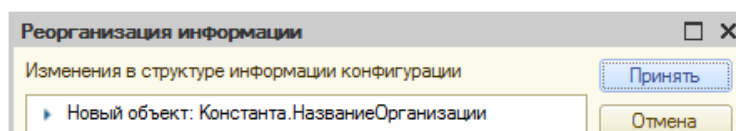


Рис.1.13

Только после выполнения всех этих действий, в пользовательском режиме можно ввести необходимую информацию (Рис 1.14 - Рис.1.15).

После запуска даем наименование нашей организации:

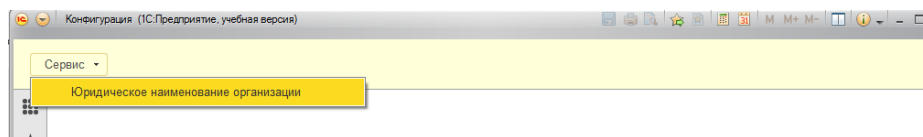


Рис.1.14

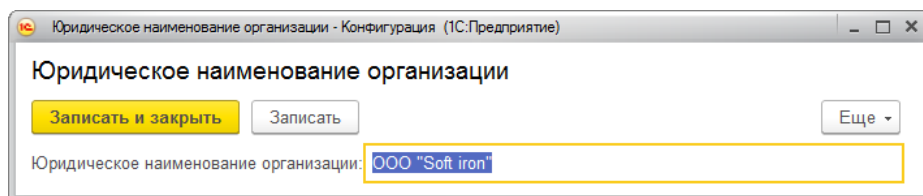


Рис.1.15

Приступим к созданию следующей константы для учета ИНН нашей фирмы. Для этого возвращаемся в режим *Конфигуратор* (т.е. закрываем окно *1С: Предприятие*) и повторим сделанные ранее действия. В окне дерева конфигурации (обратите внимание на то, как изменился вид дерева – появились новые уровни вложенности) правой кнопкой мышки на пункте *Константы* вызываем контекстное меню, в котором выбираем пункт *Добавить*.

После того, как мы выберем данный пункт, на экране откроется палитра свойств создаваемой константы. Заполняем поля *Имя* и *Синоним*, а тип данных указываем *Число*. В этом случае набор свойств изменился по сравнению с тем, какие свойства были у константы *НазваниеОрганизации* с типом данных *Строка*. (Рис.1.16)

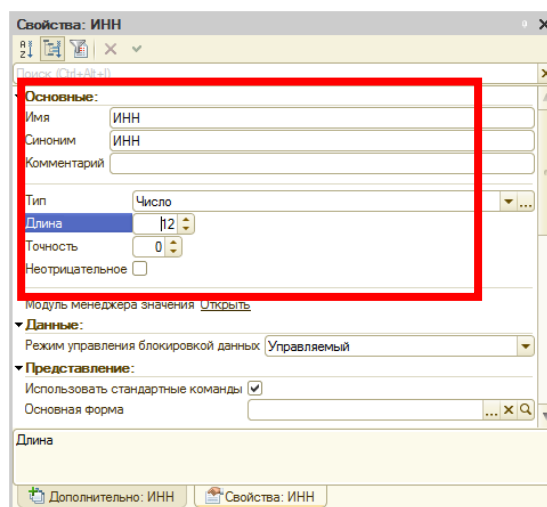


Рис.1.16

Если программа выдала данное сообщение (Рис.1.17) это указывает на то, что база открыта в пользовательском режиме, и для дальнейшей реструктуризации необходимо закрыть пользовательский режим. Только после обновления конфигурации базы данных пользователь может использовать новый объект

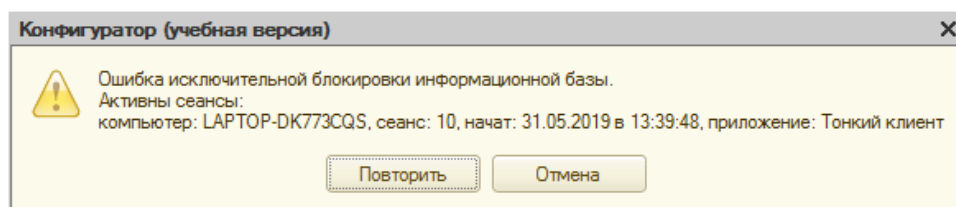


Рис.1.17

***Самостоятельно:***

***Создайте новую константу Генеральный Директор для хранения в информационной базе информацию о генеральном директоре.***

## Лабораторная работа №2. Справочники

Справочник предназначен для хранения справочной информации об однотипных объектах и представляет собой список, заполняемый пользователем на этапе исполнения. Каждый элемент этого списка может иметь сложную структуру, задаваемую на этапе конфигурирования. Количество справочников задается на этапе конфигурирования и может быть произвольным. Использование справочника позволяет избежать многократного ввода одной и той же информации, например, наименования контрагента, а так же обеспечивает сквозную идентификацию, например идентификация контрагента при отгрузке и оплате.

### Задание:

#### 1. Справочники

Для того чтобы создать новый справочник, на дереве объектов конфигурации выбираем ветку *Справочники*. Далее правой кнопкой мышки вызываем контекстное меню, в котором выбираем пункт *Добавить*, либо на панели инструментов окна дерева объектов конфигурации в меню *Действия - Добавить*, либо на панели инструментов окна дерева объектов конфигурации нажимаем кнопку *Добавить* (Рис. 2.1).

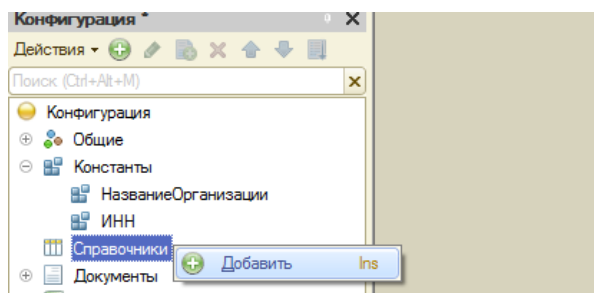


Рис.2.1

Для отличия одного справочника от другого предназначено свойство *Имя*. Именно по имени будет происходить системное обращение к этому справочнику, поэтому имя справочника формируется по тем же правилам, что и имена констант или названия переменных, т.е. нельзя использовать пробелы и ряд других служебных символов. Поле *Синоним* содержит удобное для конечного пользователя представление имени без каких-либо ограничений.

Первый справочник будем создавать для хранения информации о наших товарах и услугах. Для удобства *Имя* укажем как общее понятие "Номенклатура". (Рис.2.2)

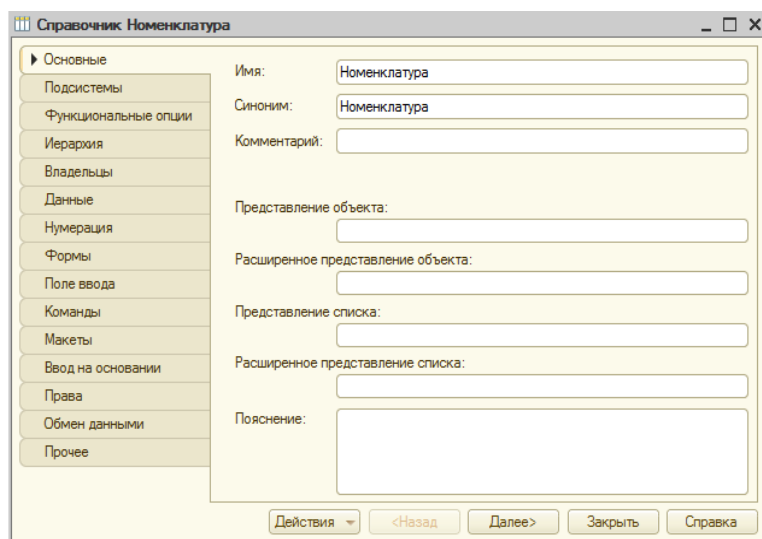


Рис.2.2

На этом создание простейшего справочника завершено. Переходим в пользовательский режим предприятия и смотрим на то, что мы только что создали (Рис. 2.3).

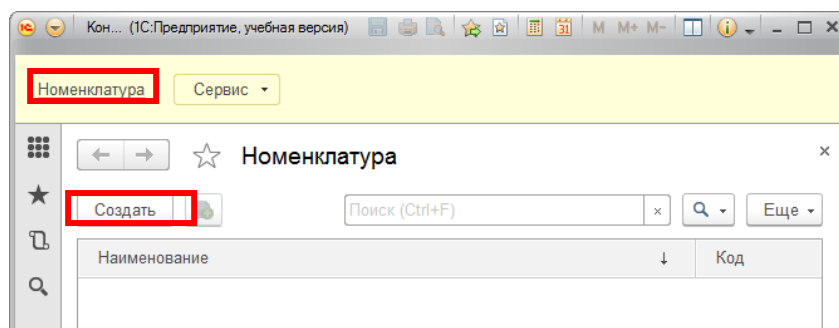


Рис.2.3

Для добавления информации о товаре нажмем на кнопку "Создать" и в открывшемся окне укажем наименование нашего товара, поле *Код* заполняется автоматически (Рис. 2.4):

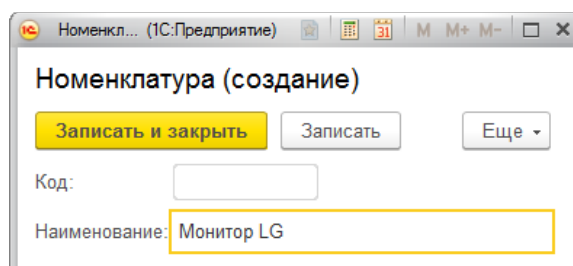


Рис.2.4

Заполним справочник данными (необходимо придумать самим, в зависимости какая у вас организация). (Рис.2.5)

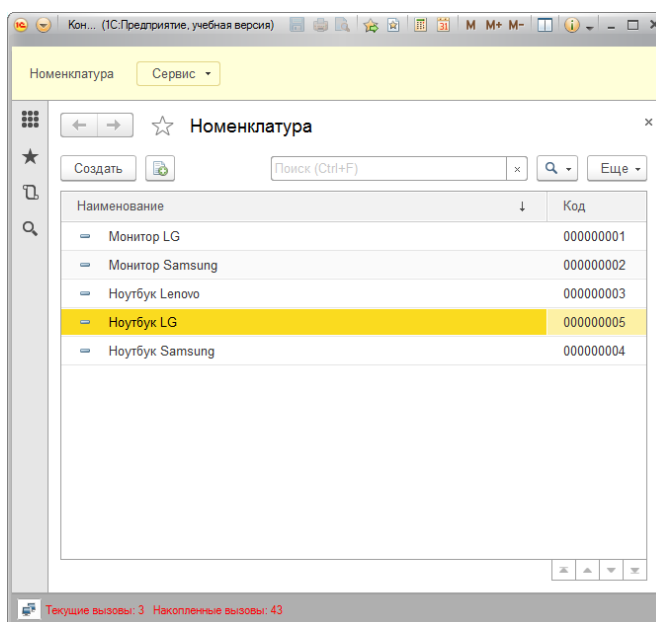


Рис.2.5

## 2. Иерархия

Созданный нами справочник позволяют комфортно работать только с небольшими наборами данных. Для удобства работы можно справочник сделать иерархическим, т.е. создать древовидную структуру.

Сделаем справочник "Номенклатура" иерархическим. Для этого в режиме конфигуратора в окне дерева объектов конфигурации двойным щелчком мышки на справочнике "Номенклатура" вызываем окно редактирования справочника и переходим на закладку *Иерархия*. Прежде всего, необходимо проставить флажок *Иерархический справочник*. Следующее поле, которое нам необходимо заполнить, это *Вид иерархии*, где возможно выбрать одно из двух значений *Иерархия групп и элементов* либо *Иерархия элементов* (Рис.2.6). Выбираем *Иерархия групп и элементов*.

Также у нас появляется еще и возможность по ограничению уровней вложенности. Для того чтобы ей воспользоваться, необходимо проставить

флажок *Ограничение количества групп иерархии* и проставить нужное количество в поле *Количество уровней иерархии*.

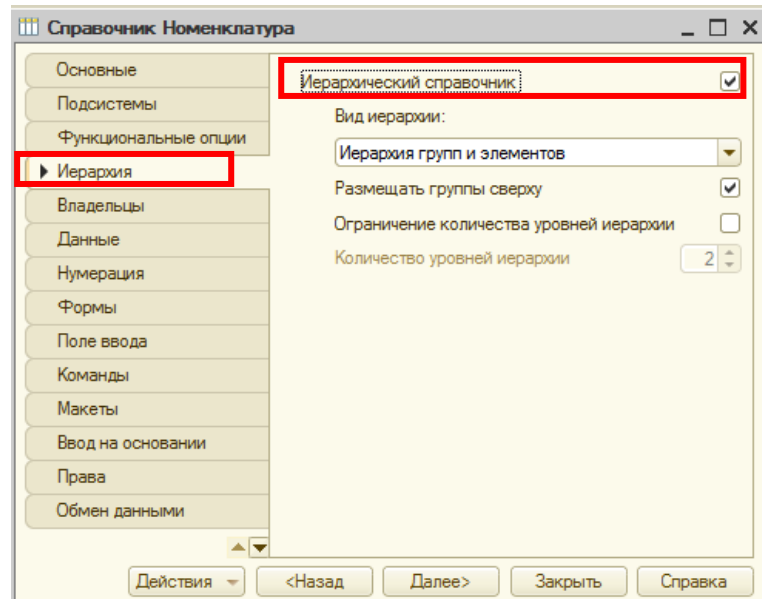


Рис.2.6

Для пользователя теперь внешний вид справочника изменится. Помимо элементов появятся еще и группы. (Рис.2.7 - 2.8)

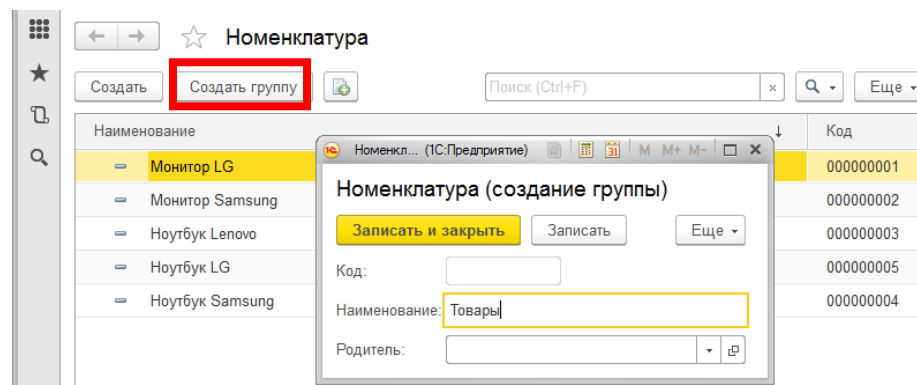


Рис.2.7

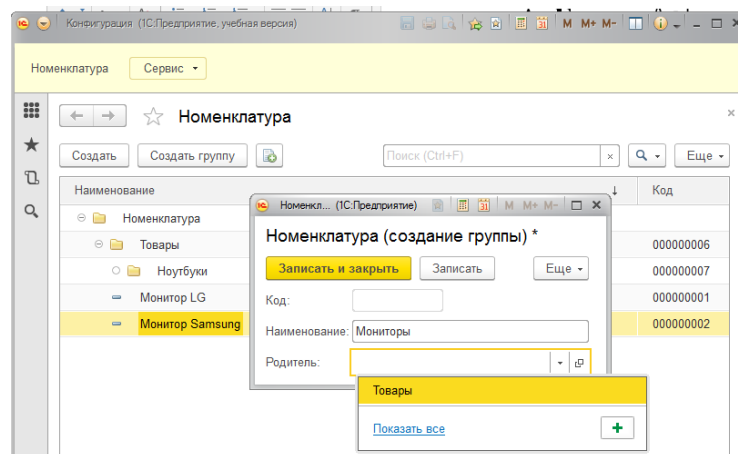


Рис.2.8



Следует обратить внимание, что с помощью кнопки "Еще" пункта меню "Режим просмотра" можно выбрать один из трех способов отображения для быстрой и легкой навигации в справочнике (Рис. 2.9). Режим просмотра выбираем *Дерево*.

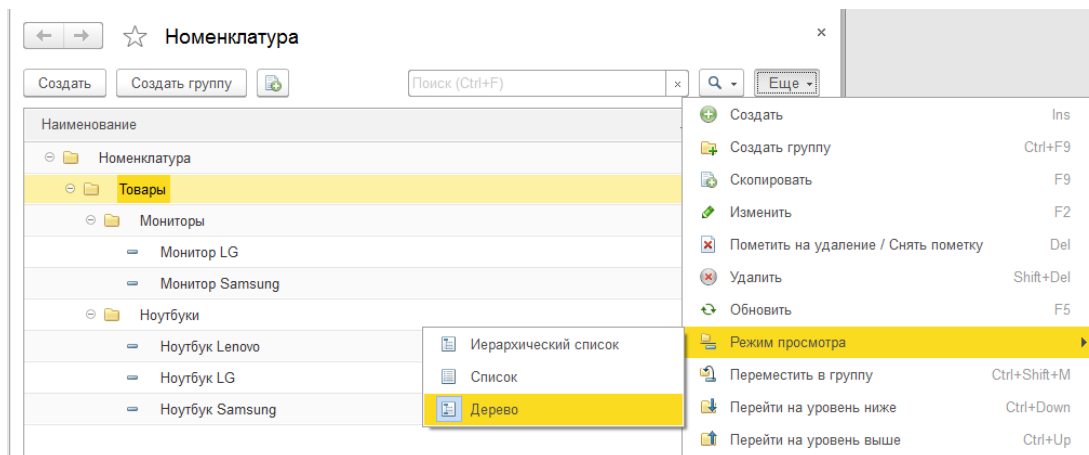


Рис.2.9

**Самостоятельно:**

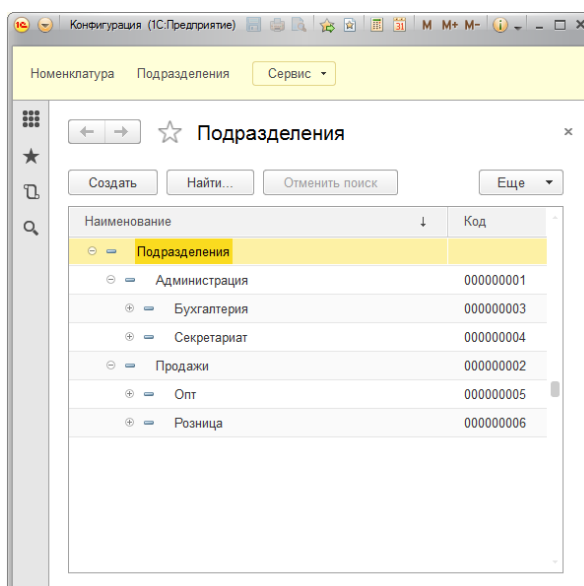


Рис.2.10

**Создайте новый справочник Подразделения для хранения данных и подразделения организации. Справочник должен иметь иерархию элементов (Рис. 2.10).**

## Лабораторная работа №3. Предопределенные элементы и Реквизиты справочника

Для удобства работы пользователя, можно заранее создать элементы справочника. В отличие от созданных пользователем интерактивно (или программно) элементов, предопределенные элементы нельзя удалить. Также каждый предопределенный элемент справочника имеет свое уникальное имя, которое позволяет более удобно (по сравнению с обычными элементами справочника) программно обратиться к этому элементу.

### Задание:

#### 1. Предопределенные элементы

Создадим новый справочник *Контрагенты*, в котором будут храниться данные о наших поставщиках и покупателях. Для удобства работы конечного пользователя, желательно разделить покупателей и поставщиков, что удобно сделать с помощью групп. Исходя из этого, справочник *Контрагенты* необходимо сделать иерархическим. Далее, либо с помощью контекстного меню в дереве конфигурации (Рис.3.1), либо по кнопке "Предопределенные" на закладке *Прочее* вызываем окно для редактирования предопределенных элементов справочника.

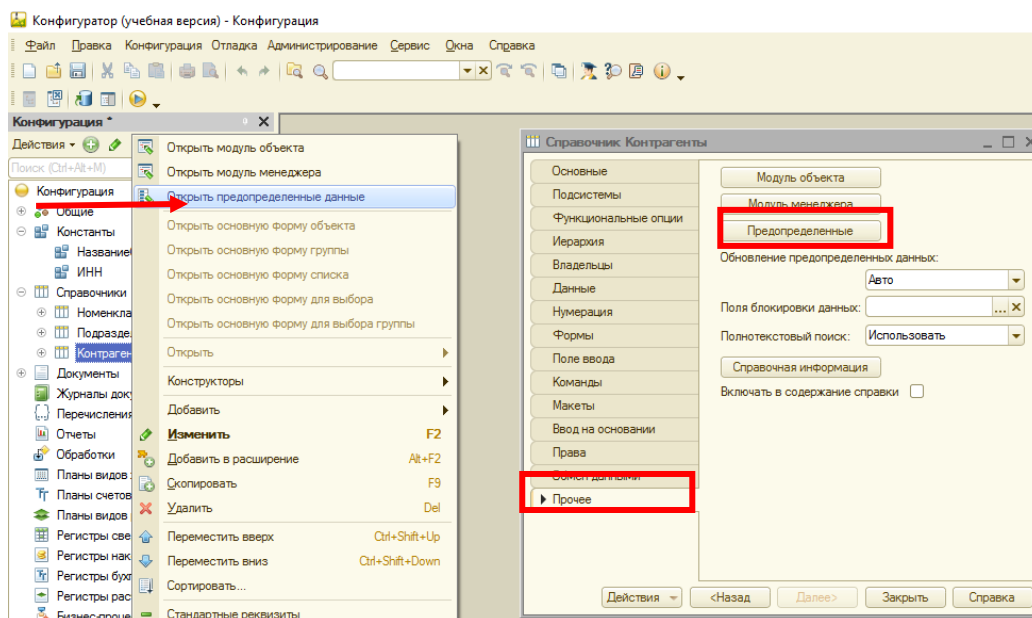


Рис.3.1

Добавление новых предопределенных групп и элементов происходит так же, как и в пользовательском режиме. Отличие будет заключаться в наличии

дополнительного системного реквизита *Имя*, на которое распространяются стандартные правила наименования, и которое не будет доступно для пользователя в режиме "1С:Предприятие". (Рис.3.2)

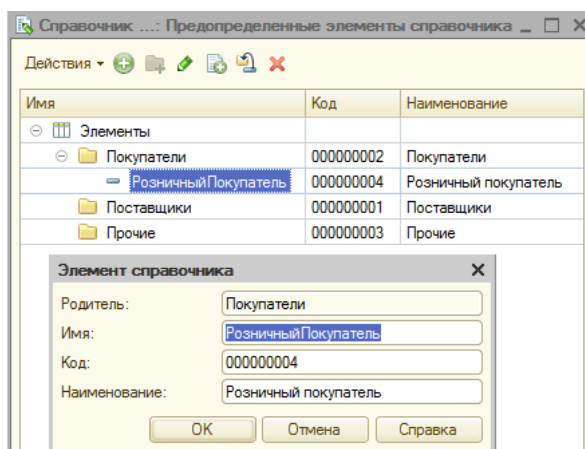


Рис.3.2

В режиме "1С:Предприятие" для пользователя будет уже готовая иерархия. Заполним справочник данными (Рис. 3.3).

Наименование	Код
Контрагенты	
Покупатели	000000002
Властелина	000000007
МММ	000000005
Розничный покупатель	000000004
Хопёр-Инвест	000000006
Поставщики	000000001
Красный богатырь	000000009
Красный пролетарий	000000008
Красный цветок	000000010
Прочие	000000003
ПФ РФ	000000011

Рис.3.3

## 2. Реквизиты справочника

Возможна ситуация, когда для элемента справочника нам необходимо хранить дополнительную информацию. Для этой цели служат реквизиты справочника.

В простейшем случае можно рассматривать справочник как таблицу, где каждому элементу справочника будет соответствовать своя строка таблицы, а колонки этой таблицы будут соответствовать его реквизитам. Для объекта

типа *Справочник* всегда существует ряд системных реквизитов, например, *Код* и *Наименование*.

На закладке *Нумерация* необходимо определить следующие свойства кода справочника (Рис.3.4):

- *Автонумерация* – позволяет системе автоматически присваивать следующий код при создании нового элемента справочника. В дальнейшем данный код можно изменить.
- *Контроль уникальности* – системная возможность отслеживать уникальность кодов, т.е. при записи нового элемента будет проводиться контроль существования элемента с таким кодом.
- *Серии кодов* – определяет способ нумерации кодов *Во всем справочнике*, *В пределах подчинения* (внутри каждой группы) или *В пределах подчинения владельцу* (имеет смысл для подчиненных справочников).

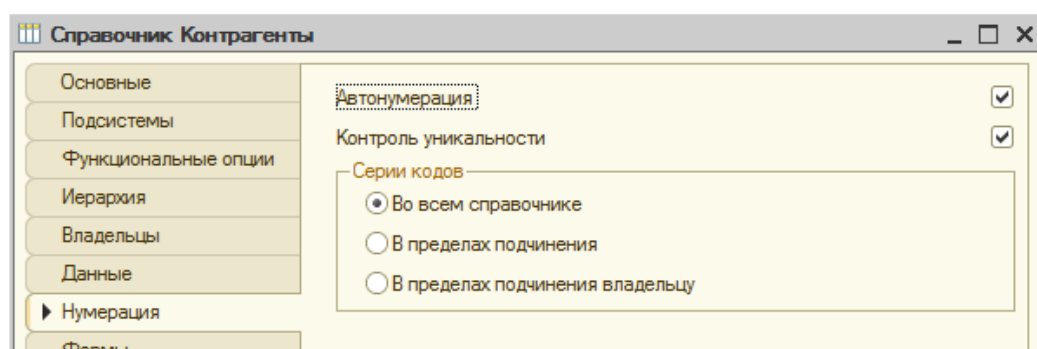


Рис. 3. 4

- *Длина наименования* – максимальное количество символов, которое может занимать наименование.
- *Длина кода* – максимальное количество символов, которое может занимать код. Например, при длине кода 10, сам код может быть равен как "1" или "100000".
- *Тип кода* – определяет будет ли код храниться как число или как строка. Следует отметить, что в случае строкового кода цифры воспринимаются в качестве символов, например код "01" и код "001" будут различаться.

Для того, чтобы добавить новый реквизит в справочник, на закладке *Данные* правой кнопкой мышки на пункте *Реквизиты* вызвать контекстное меню и выбрать пункт *Добавить* (Рис . 3.5).

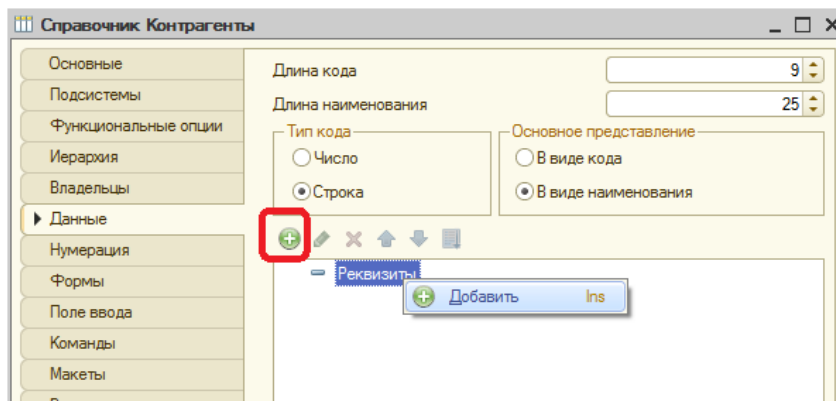


Рис. 3.5

В справочнике "Контрагенты" создадим новый реквизит *ИНН* (тип *Число*, длина 12). (Рис.3.6)

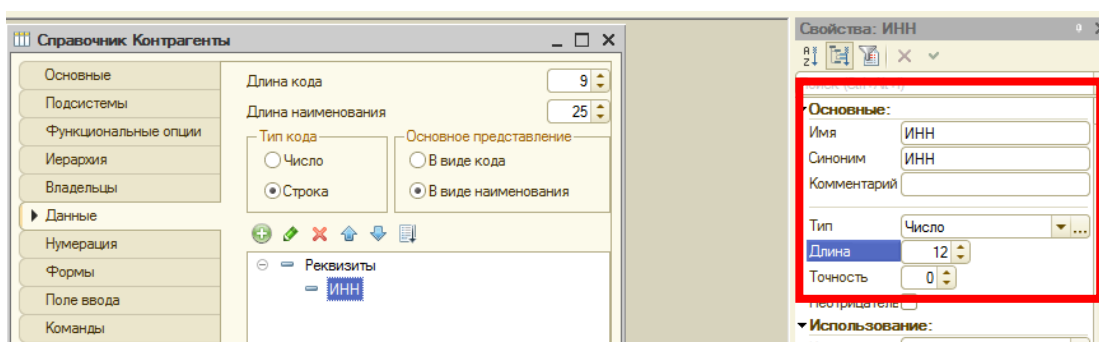


Рис.3.6

После обновления базы для пользователя появляется возможность у каждого контрагента указать ИНН (Рис. 3.7).

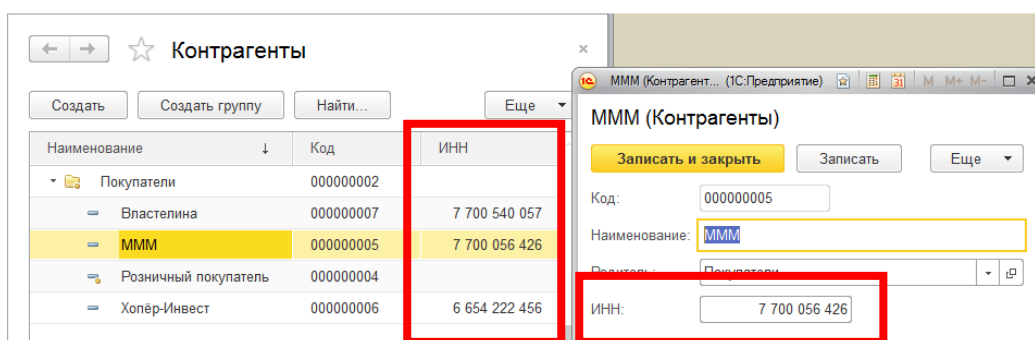


Рис. 3.7

Рассмотрим теперь следующую ситуацию: в целях ведения кадрового учета необходимо хранить информацию о сотрудниках, работающих в нашей организации. Кроме того, требуется отражать в системе информацию о подразделении, в котором работает сотрудник.

Для решения поставленной задачи добавим в *Окне конфигурации* новый справочник *Сотрудники*. Стандартные реквизиты для пользователя будут называться "Код" и "Наименование", что для сотрудника не совсем логично. Ведь наименование сотрудника это его Ф.И.О., а код, скорее всего, будет удобно использовать в качестве табельного номера. Чтобы сделать данное улучшение зададим синонимы для стандартных реквизитов (Рис. 3.8).

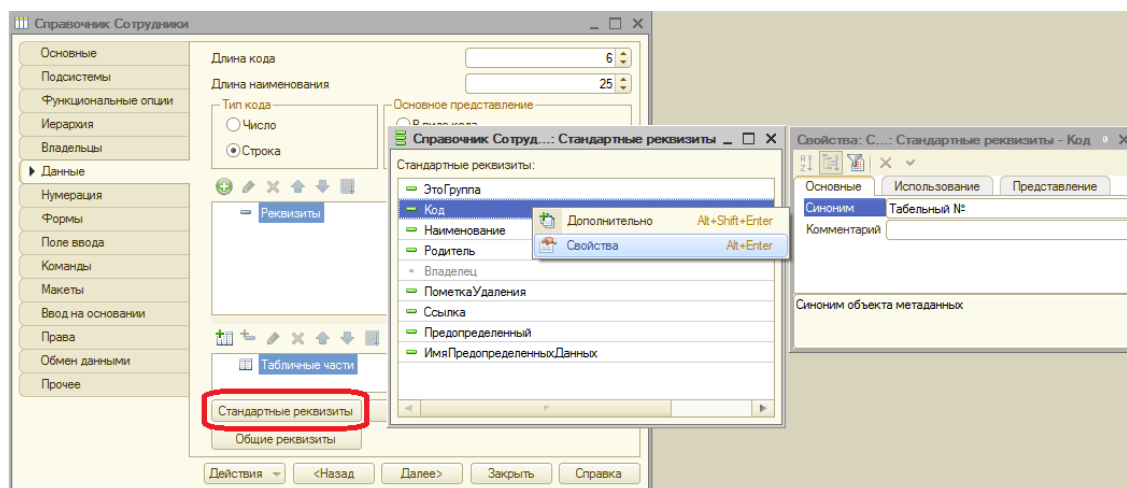


Рис. 3.8

Теперь добавим реквизит *Подразделение*, с типом данных *СправочникСсылка.Подразделения*. (Рис. 3.9)

В этом случае, вновь созданный реквизит будет заполняться значениями из справочника *Подразделения* (Рис. 3.10), причем следует иметь в виду, что в справочнике *Сотрудники* будет храниться ссылка на элемент справочника *Подразделения*, т.е. фактически, обращаясь к реквизиту *Подразделение* справочника *Сотрудники*, мы обращаемся к элементу справочника *Подразделения*.

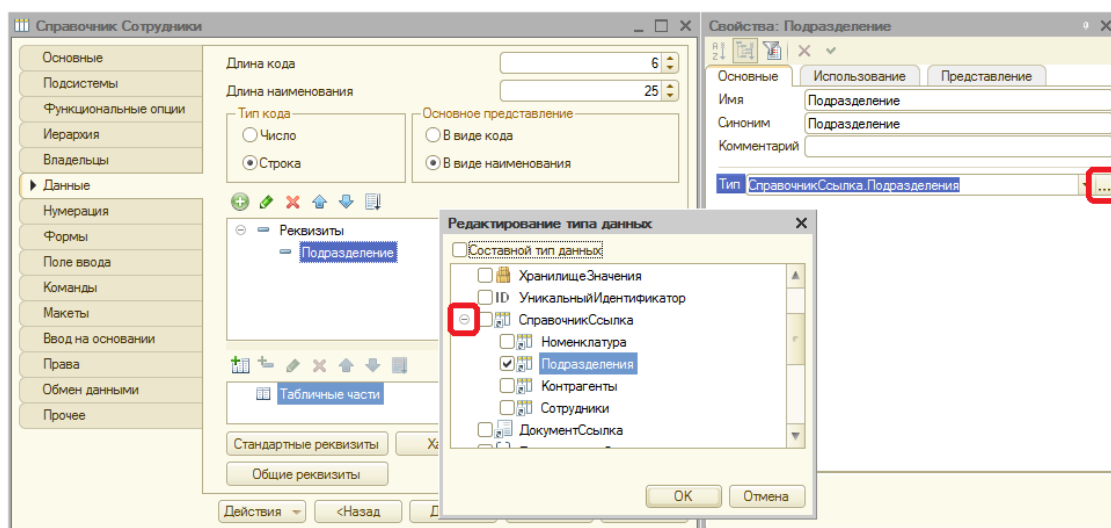


Рис. 3.9

ФИО	↓	Табельный №	Подразделение
Сотрудники			
Работающие		000001	
Балаганов А.П.		000005	Продажи
Бендер О.И.		000003	Администрация
Козлевич А.К.		000006	Секретариат
Паниковский М.С.		000004	Бухгалтерия
Уволенные		000002	
Воробьянинов И.М.		000007	
Фунт		000008	

Рис. 3.10

Например, для сотрудника *Балаганов* мы указали, что он работает в отделе *Продажи*. После этого, в результате структурной реорганизации компании название отдела изменилось, и он стал называться не просто *Продажи*, а *Отдел продаж*. Для отражения новой ситуации в справочнике *Подразделения* мы поменяем название у соответствующего элемента на *Отдел продаж*. Надо ли нам теперь изменить название отдела у сотрудника *Балаганов*? Нет, не надо, поскольку в справочнике *Сотрудники* хранится не название отдела, а ссылка на элемент справочника, со всеми его реквизитами.

***Самостоятельно:***

***Создайте в справочнике Номенклатура новый реквизит для хранения ставки НДС, по которой в дальнейшем будет осуществляться продажа данной номенклатурной позиции.***



## Лабораторная работа №4. Табличная часть и Подчинение

Табличная часть представляет собой таблицу, структура которой задается на этапе конфигурирования, а данные заносятся пользователем на этапе исполнения. Для объекта типа Справочник реализована возможность создания произвольного количества табличных частей.

### Задание:

#### 1. Табличная часть

Открываем Справочник «Сотрудники» в режиме Конфигуратор. Для того, чтобы создать табличную часть справочника, на закладке Данные правой кнопкой мышки необходимо вызвать контекстное меню, в котором выбрать пункт *Добавить*. Тот же результат можно получить, вызвав контекстное меню правой кнопкой мышки в окне дерева объектов конфигурации на пункте *Табличные части* выбранного справочника. (Рис. 4.1)

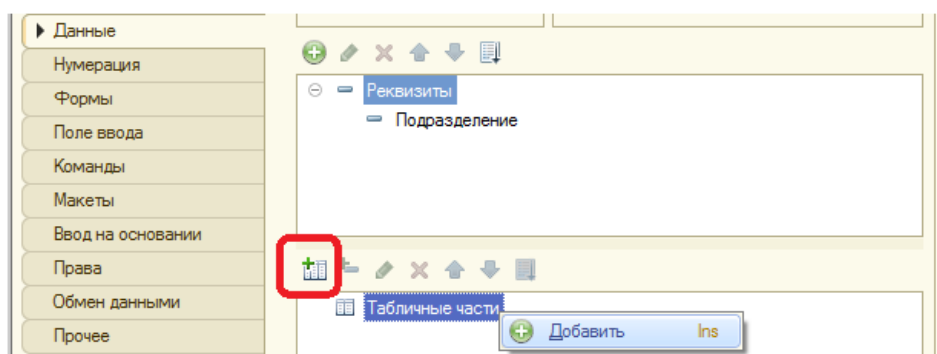


Рис. 4.1

Первую табличную часть назовем *Дети*.

Следующее что необходимо сделать, это определить структуру табличной части. В нашем случае необходимо, чтобы таблица содержала две колонки имя ребенка и дату рождения. Создадим два реквизита табличной части. Реквизиты табличной части определяются, так же, как и реквизиты справочника. Правой кнопкой мышки на соответствующей табличной части вызывается контекстное меню, в котором выбирается пункт *Добавить - Реквизит табличной части*. (Рис. 4.2)

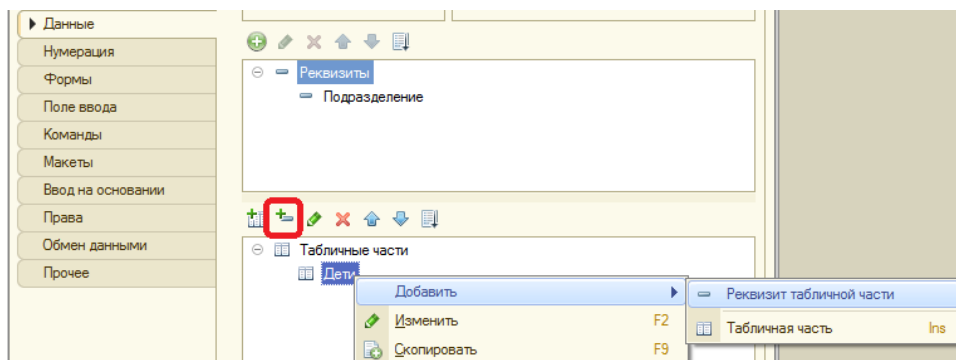


Рис. 4.2

Для хранения данных о фамилии, имени и отчестве ребенка создадим строковый реquisite с длиной 50 символов, который назовем "ФИО", а для даты рождения - соответственно реquisite *ДатаРождения* типа *Дата* с составом даты *Дата*, поскольку время рождения ребенка нас не интересует (Рис. 4.3).

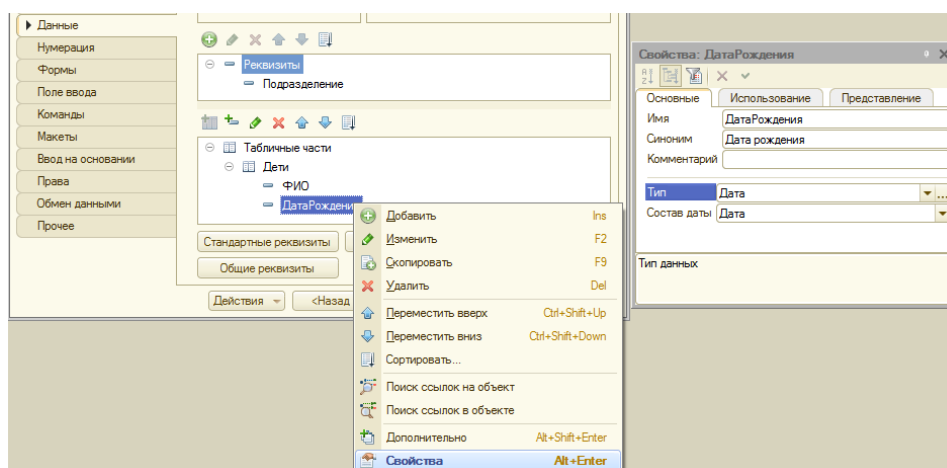


Рис. 4.3

В результате таких действий в пользовательском режиме справочник имеет вид Рис. 4.4.

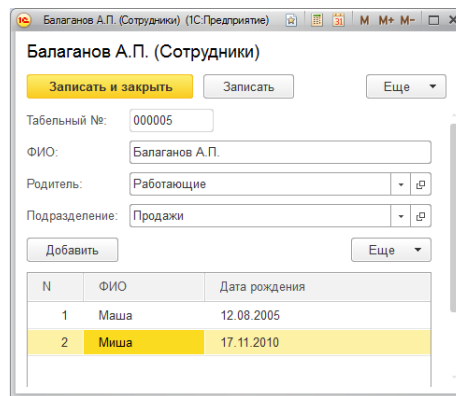


Рис. 4.4

### Самостоятельно

*Добавьте в справочнике Сотрудники табличную часть Образование (Рис. 4.5)*

N	Учреждение	Дата поступления	Дата окончания	Специальность
1	МГУ	01.09.2000	31.05.2006	Бухгалтер
2	1С: УЦ №1	03.05.2011	05.05.2011	Знакомство с платформой

Рис. 4.5

## 2. Подчинение

Создадим новый справочник *Договоры*. Предполагаем, что количество договоров у одного клиента будет не очень большим и группировать их не придется, поэтому справочник делаем неиерархическим. Для настройки механизма подчинения перейдем на закладку *Владельцы*. В панели инструментов окна *Список владельцев справочника* необходимо выбрать пункт *Редактировать элемент списка*. При выборе данного пункта у нас откроется список для выбора, в котором необходимо выбрать справочник владелец. (Рис. 4.6).

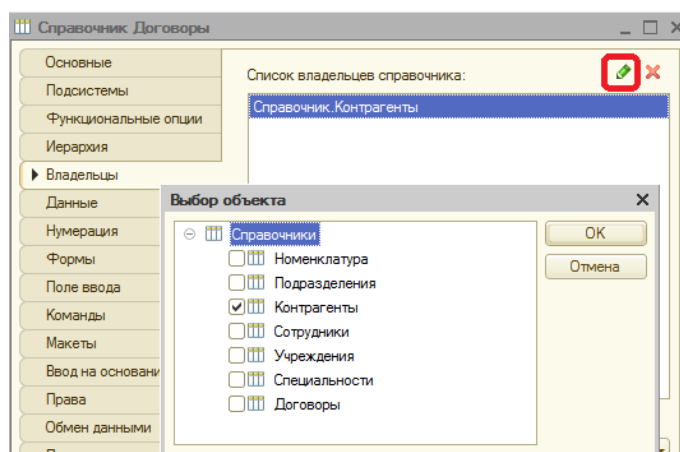


Рис. 4.6

В нашем случае справочник *Договоры* должен быть подчинен справочнику *Контрагенты*, т.е. справочник *Контрагенты* будет являться *Владельцем* справочника *Договоры* (система позволяет организовывать подчинение одновременно нескольким справочникам, однако принципиально такой случай ничем отличаться не будет и в рамках данного курса не будет рассмотрен).

В режиме "1С:Предприятие" при работе с каждым контрагентом появляется кнопка "Договоры", которая позволяет открыть список только тех договоров, которые связаны с текущим контрагентом (Рис. 4.7).

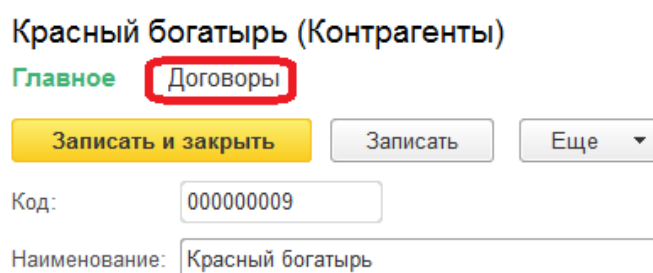


Рис. 4.7

Возврат к данным контрагента происходит через ссылку "Главное", а работа с договорами происходит стандартным образом, также как с любым другим справочником.

## Лабораторная работа №5. Форма. Форма списка, форма элемента

Работа пользователя в режиме "1С:Предприятие" происходит не напрямую с объектом, хранящим информацию, а посредством объекта Форма. Следует отметить, что в зависимости от текущей задачи пользователя работа с одним и тем же объектом может происходить с использованием различных форм. Например, для справочника при работе с ним как со списком элементов используется Форма списка, а при работе с одним элементом данного списка используется Форма элемента.

### Задание:

Создадим форму списка для справочника *Контрагенты*. Для этой цели перейдем на закладку формы и создадим новую форму списка с помощью конструктора (Рис. 5.1).

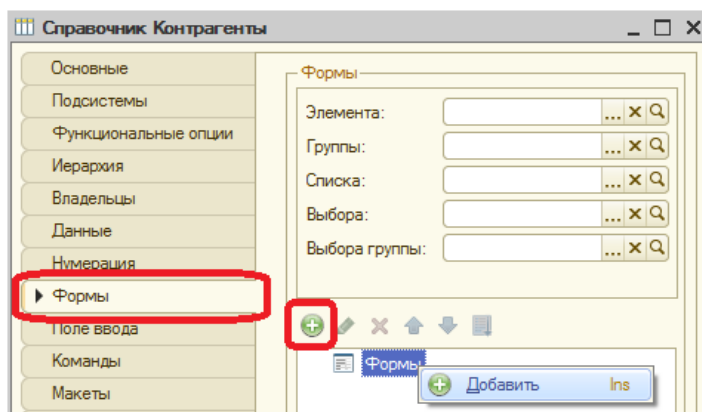


Рис. 5.1

Будем последовательно делать настройки, которые предлагает конструктор. На первой закладке укажем, что создаем форму списка (Рис. 5.1).

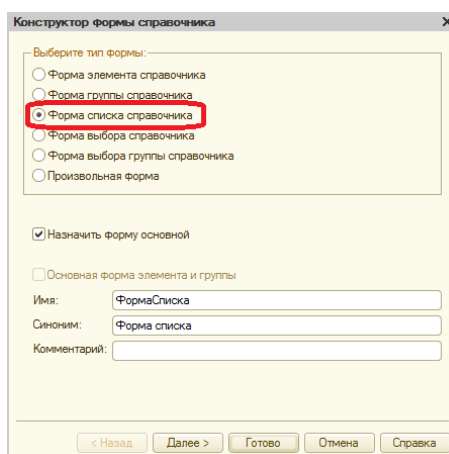


Рис. 5.1

Далее отметим те реквизиты справочника, которые мы хотим видеть (Рис. 5.2)

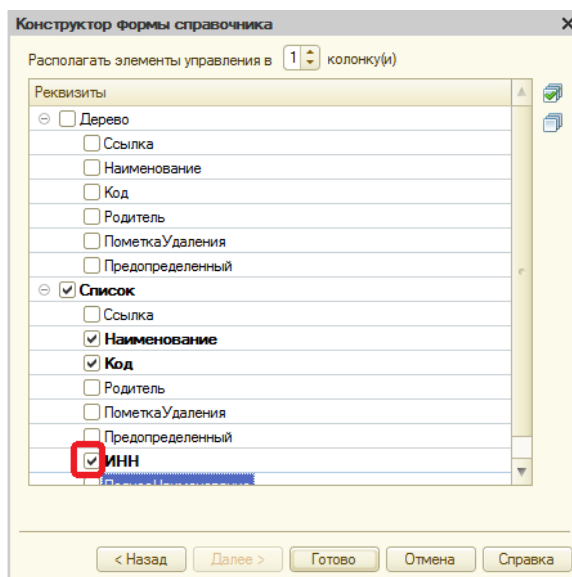


Рис. 5.2

В результате будет создана форма (Рис. 5.3)

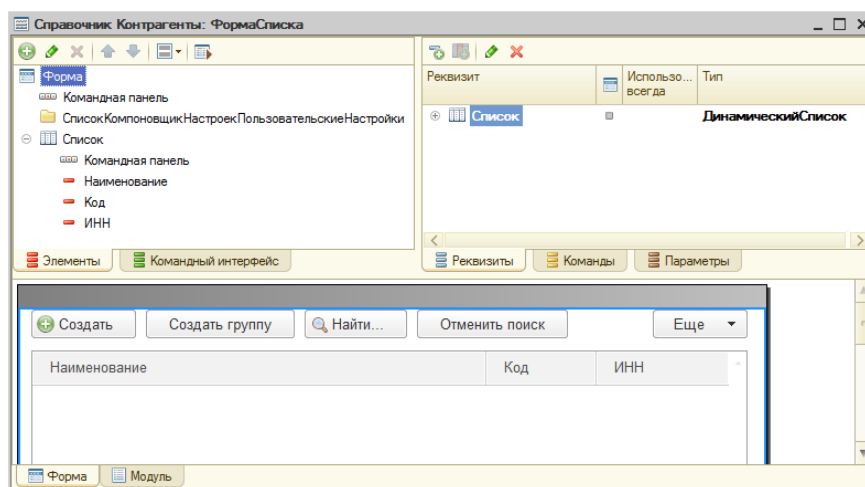


Рис. 5.3

Откроем закладку Команды – Глобальные команды и перетащим элемент "Элементы.Список.ТекущиеДанные.Ссылка" в левое окно в "Командная панель" (Рис. 5.4).

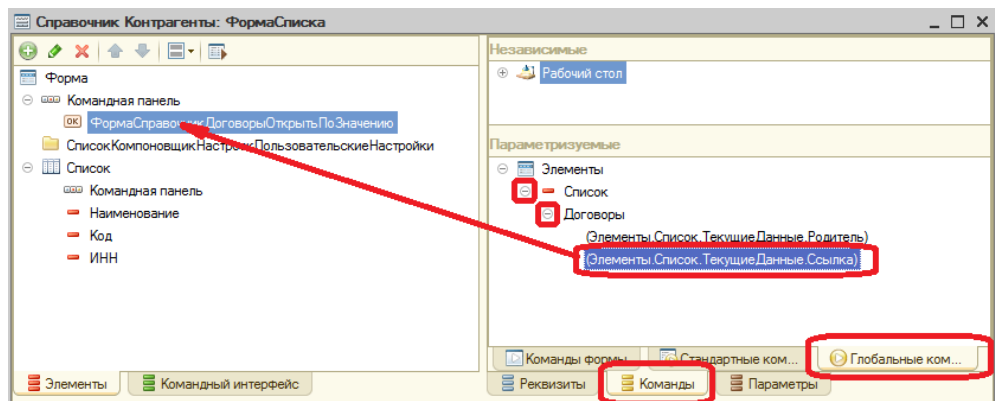


Рис. 5.4

В свойствах появившегося элемента укажем "ТолькоВоВсехДействиях" в значение "Нет" (Рис. 5.5)

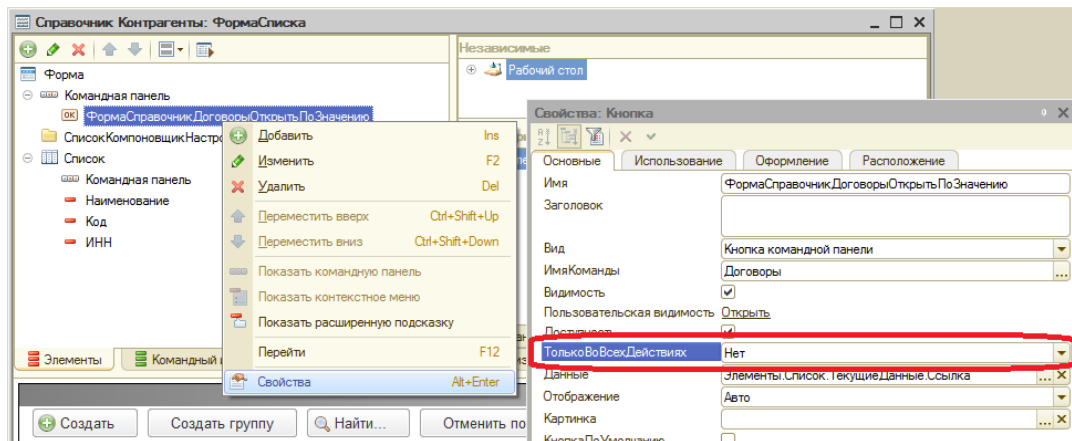


Рис. 5.5

Теперь в пользовательском режиме в форме списка справочника сотрудники и у пользователя появится возможность просматривать только те договоры, которые относятся к текущему контрагенту (Рис. 5.6).

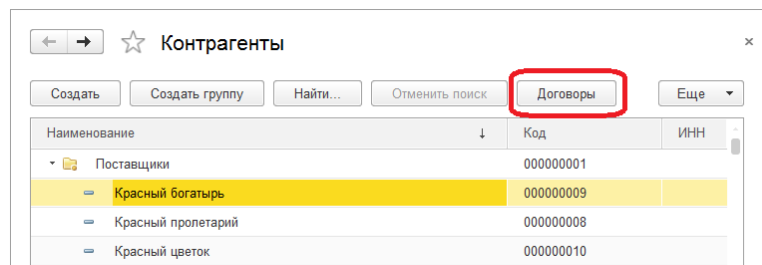


Рис. 5.6

Аналогичным образом может быть создана и форма элемента для справочника (Рис. 5.7). Главное, это указать правильный тип формы.

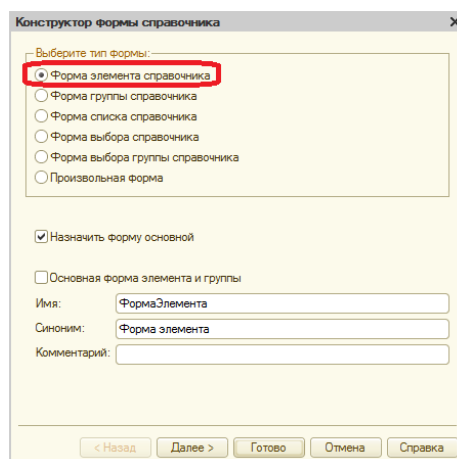


Рис. 5.7

Например, для справочника "Сотрудники" форма элемента не должна отображать реквизит "Родитель" - группу, в которой находится текущий элемент справочника. Терминология платформы в данном случае входит в противоречие с "общечеловеческим" пониманием данного термина. Чтобы не смущать пользователя, удали эту информацию с формы (Рис. 5.8)

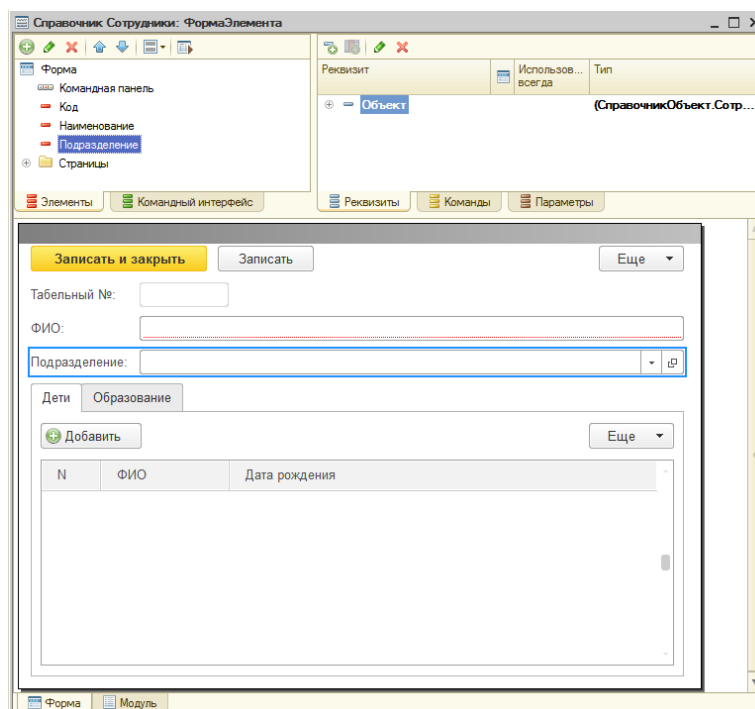


Рис. 5.8



## Лабораторная работа №.6 Перечисление. Элементы управления формы

Перечисление представляет собой неиерархический список конечной длины, заполняемый значениями только на этапе конфигурирования. Каждый элемент такого списка представляет собой простое значение (не имеет каких-либо реквизитов или атрибутов) и не может изменяться пользователем.

### Задание:

#### 1. Перечисление

Создаем новое перечисление *ВидыПриемаНаРаботу*. Для этого правой кнопкой мыши на ветке Перечисления вызываем контекстное меню и выбираем пункт *Добавить*. (Рис. 6.1)

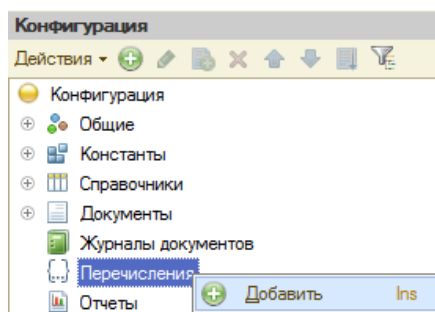


Рис. 6.1

Далее задаем имя: *ВидыПриемаНаРаботу*, при этом автоматически заполняется поле *Синоним* точно так же, как это происходило для справочников. (Рис. 6.2)

На закладке *Данные* правой кнопкой мыши на пункте *Значения* вызываем контекстное меню и выбираем пункт *Добавить*, либо выбираем соответствующую кнопку в панели инструментов. Появляется уже знакомая нам палитра свойств (Рис. 6.3), в которой задаем *Имя* - значения перечисления. Аналогично создаем следующее значение.

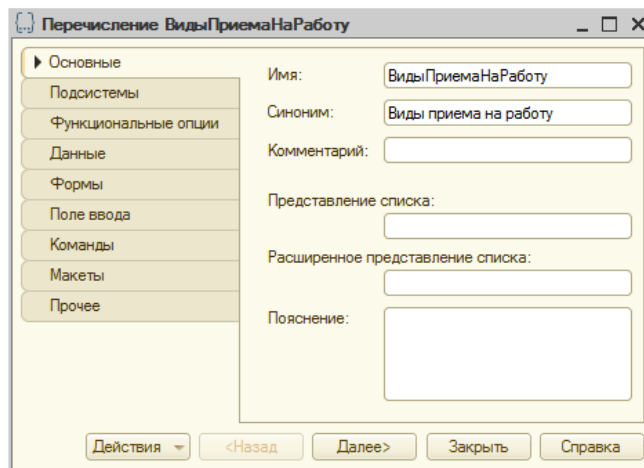


Рис. 6.2

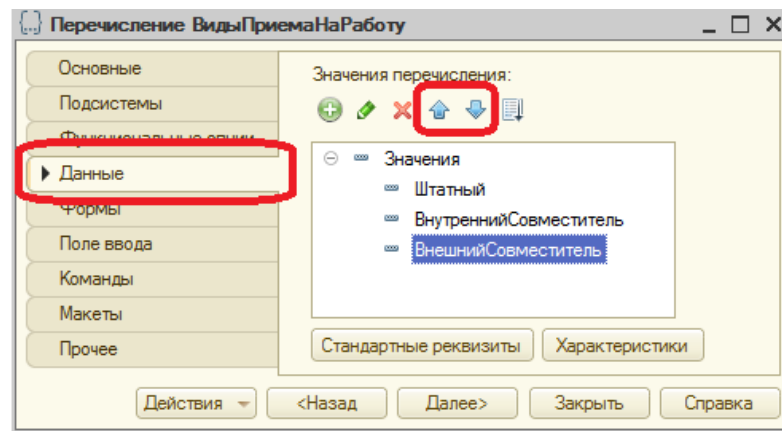


Рис. 6.3

После всех этих действий в справочник *Сотрудники* добавим новый реквизит *ВидПриемаНаРаботу* с типом данных *ПеречислениеСсылка*. *ВидыПриемаНаРаботу* (Рис. 6.4).

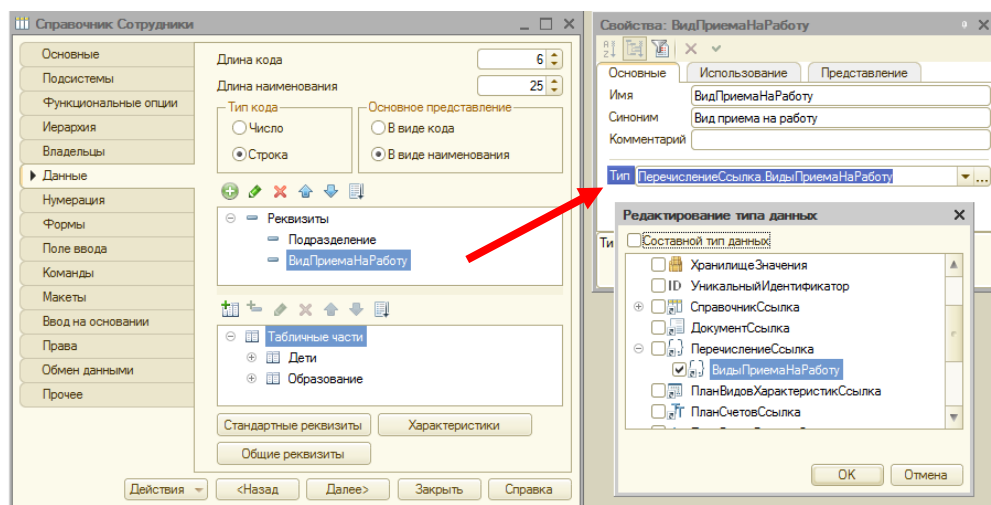


Рис. 6.4

Обратите внимание, что после обновления у пользователя всё равно отсутствует возможность указать способ приёма на работу. Это связано с тем, что форма элемента была нами создана, и платформа теперь не вмешивается в ее работу. Чтобы данная информация стала доступна пользователю, необходимо внести изменения на форму элемента справочника.

## 2. Элементы управления формы

В справочнике Сотрудники у нас есть реквизит *ВидПриемаНаРаботу*, однако пользователь не имеет возможности работать с ним. Для того, чтобы такая возможность появилась, необходимо разместить этот реквизит на форме элемента справочника. Удобно работать с этим реквизитом позволит элемент формы *Поле* с видом *Поле ввода*.

Чтобы разместить на форме поле ввода, в котором будут отражаться данные реквизита, можно воспользоваться двумя различными способами. Можно добавить новый элемент формы, указав тип *Поле* (Рис. 6.5) и указав затем в его свойствах *ПутьКДанным*, *Вид* и *Заголовок* (Рис. 6.6)

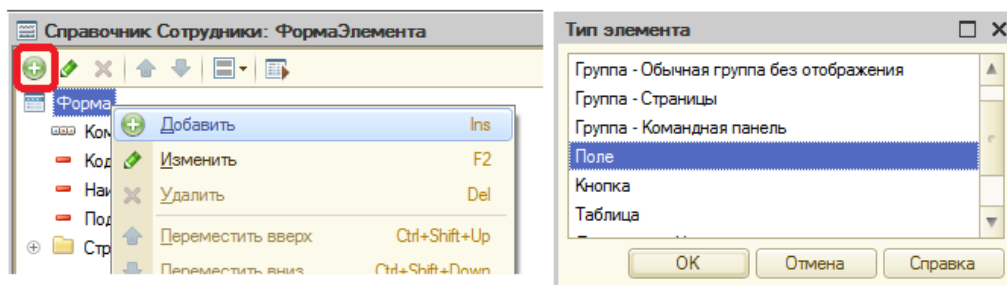


Рис. 6.5

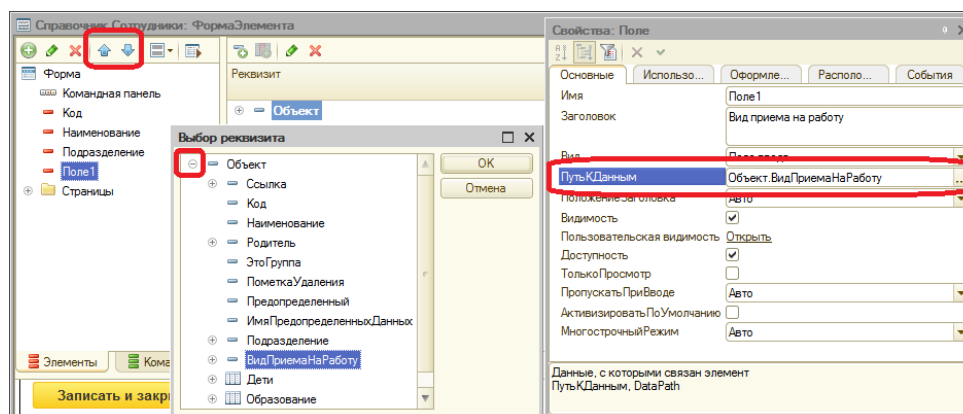


Рис. 6.6

Того же результата можно достичь если, придерживая левую кнопку мышки перетащить реквизит из окна реквизитов в окно элементов формы (Рис. 6.7)

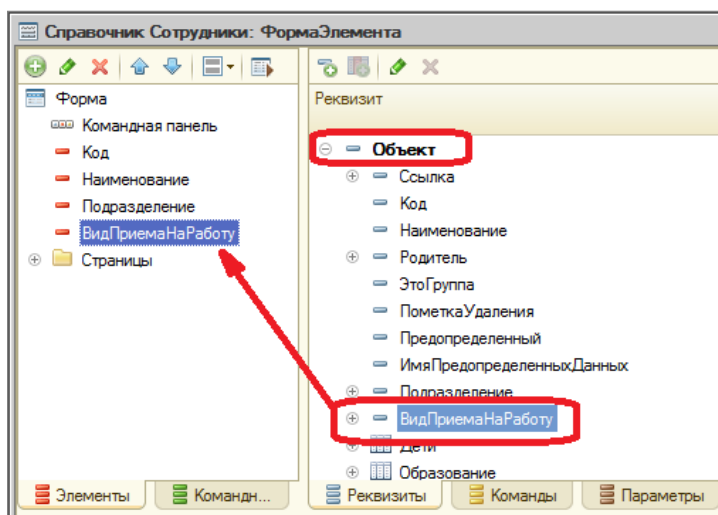


Рис. 6.7

### Поле флажка

С помощью флажка на форме могут отображаться данные с типом *Булево* или *Число*.

Создадим форму элемента для справочника *Номенклатура* (Рис. 6.8).

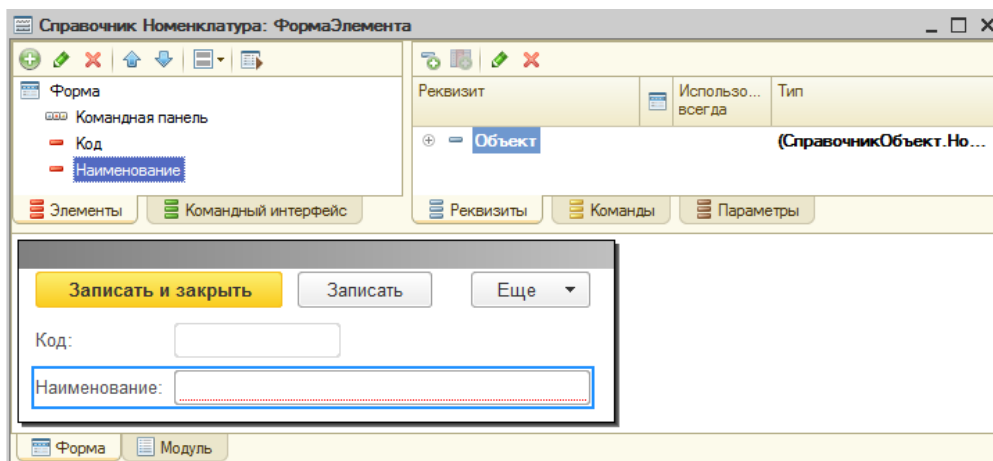


Рис. 6.8

После этого добавим в справочник реквизит *Услуга* с типом *Булево* для того, можно было указать, является ли данный элемент справочника товаром

или услугой. Теперь добавим этот реквизит на форму. Для этого перетащим реквизит *Услуга* из окна реквизитов в окно элементов формы (Рис. 6.9)

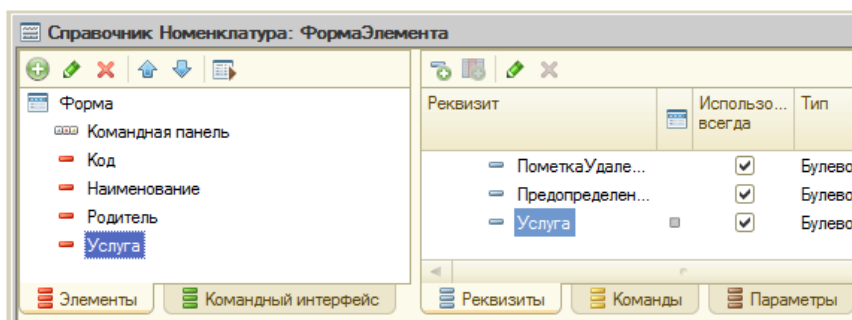


Рис. 6.9

Поскольку реквизиты с типом *Булево* удобнее всего отображать на форме в виде флажка, то система сама автоматически укажет данный вид поля (Рис. 6.10)

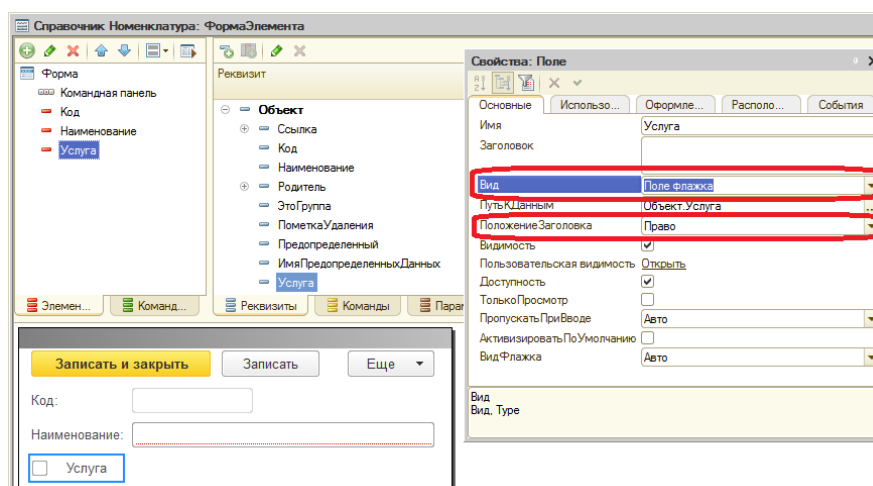


Рис. 6.10

Для флажка наиболее важными для нас будут являться следующие свойства:

*Заголовок* – текстовый заголовок, видимый пользователю

*Положение заголовка* – положение заголовка относительно флажка.

### Поле переключателя

Переключатель предназначен для того, чтобы предоставить пользователю выбор одного из нескольких заранее известных вариантов.

В справочнике *Номенклатура* для каждого товара будем указывать его качество, которое может быть одним из трех: высокое, обычное и низкое. Для этого создадим новое перечисление *КачествоТоваров* из трех значений. После этого в справочнике *Номенклатура* добавим новый реквизит *КачествоТовара* с соответствующим типом данных. В свойствах реквизита укажем значение заполнения (значение, которое будет выбираться по умолчанию) (Рис. 6.11)

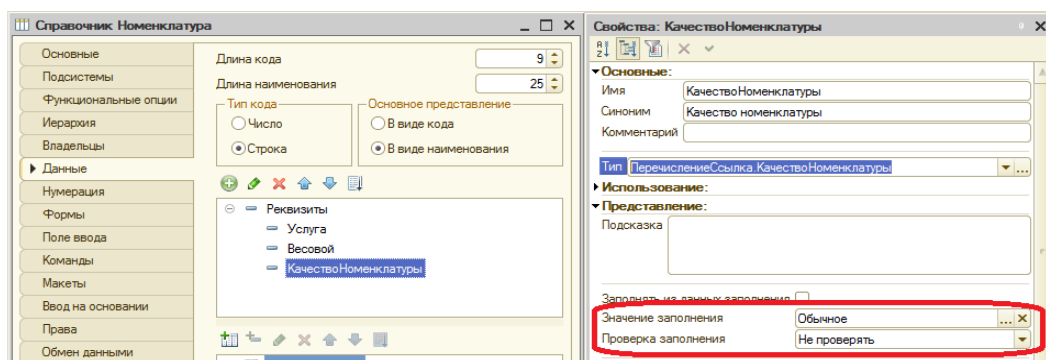


Рис. 6.11

Чтобы создать необходимое количество элементов переключателя, необходимо воспользоваться *СпискомВыбора*. Кроме того, традиционно элементы переключателя обычно располагаются друг под другом, для чего в настройках укажем количество колонок (Рис. 6.12).

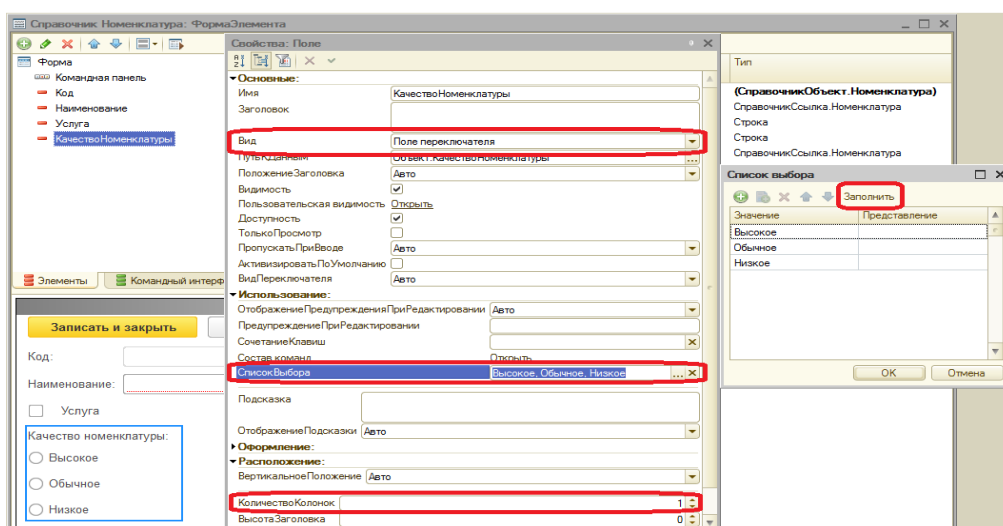


Рис. 6.12

Для переключателя наиболее важными для нас будут являться следующие свойства:

*Заголовок* – текстовый заголовок, видимый пользователю

*Положение заголовка* - определяет положение заголовка относительно элементов переключателя

*Список выбора* – определяет элементы переключателя, их количество, представление на форме и выбираемые значения

*Количество колонок* – определяет внешний вид переключателя

## Лабораторная работа №.7 Документ. Формы документа

Документ — это вид объектов конфигурации, предназначенный для отражения в системе событий, произошедших в хозяйственной жизнедеятельности предприятия, и соответствует общепринятому понятию документ. Документ однозначно определяется своим видом, номером и датой.

В окне дерева объектов конфигурации правой кнопкой мышки на ветке *Документы* вызываем контекстное меню и выбираем пункт *Добавить*. В открывшемся окне необходимо задать *Имя* документа, т.е. его вид. Назовем создаваемый документ *РасходДенег*, а в качестве *Синонима* укажем *Списание денежных средств* (Рис. 7.1). Далее необходимо определить структуру нашего документа, которая задается набором его реквизитов.

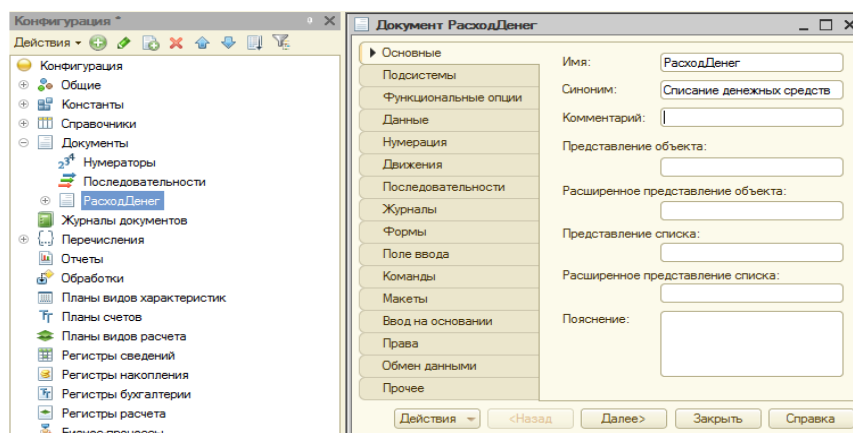


Рис. 7.1

В окне редактирования документа *РасходДенег* на закладке *Нумерация* необходимо задать следующие свойства, относящиеся к его дате и номеру (Рис. 8.2):

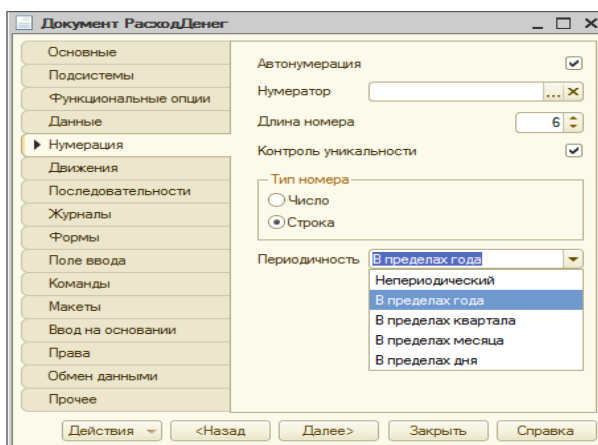


Рис. 7.2



Согласно поставленной задаче, необходимо вести учет взаиморасчетов с контрагентами в разрезе договоров. Помимо этого, документ *РасходДенег* должен позволить нам выплатить деньги не только поставщикам, но и сотрудникам. Для реализации этого требования необходимо создать новые реквизиты документа *РасходДенег*. Нам потребуется создать три реквизита:

- Контрагент,
- Договор,
- Сумма.

На закладке *Данные*, точно так же, как это делалось и для справочника, добавляем новые реквизиты. (Рис. 7.3)

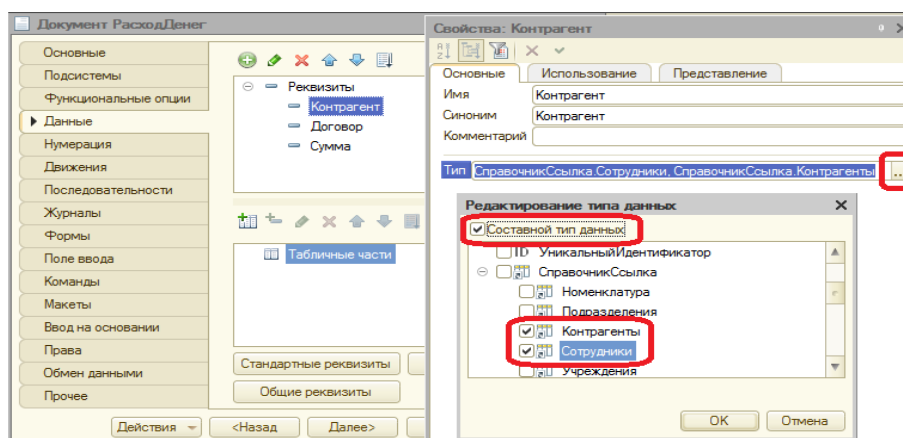


Рис. 7.3

Зададим *Имя* реквизита, его тип и, в случае необходимости, дополнительные настройки типа, например длину строки или точность числа. Поскольку в системе уже хранится информация о контрагентах и договорах, то и тип реквизитов *Контрагент* и *Договор* необходимо указать *СправочникСсылка.Контрагенты* и *СправочникСсылка.Договоры* соответственно. Для реквизита *Сумма* потребуется не только указать тип *Число*, но и *точность 2*, чтобы можно было корректно заносить суммы поступивших денег. Кроме того, чтобы иметь возможность выплатить денежные средства не только контрагентам, но и сотрудникам, можно для реквизита *Контрагент* указать составной тип данных *СправочникСсылка.Контрагенты* и *СправочникСсылка.Сотрудники*.

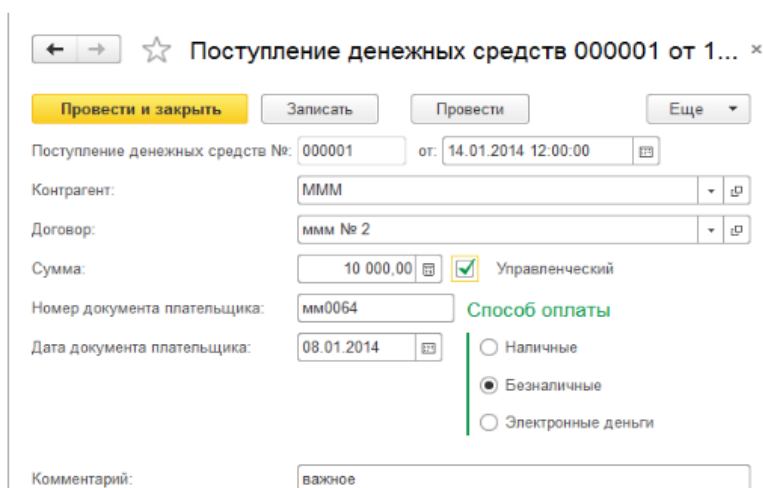
## Формы документа

Точно так же, как и при работе со справочником, пользователь работает не напрямую с объектом "Документ", а посредством экранной формы. Соответственно, при работе с одним документом используется *Форма документа*, а для работы с несколькими документами данного вида используется *Форма списка*.

В том случае, когда автоматически создаваемая форма не устраивает, имеется возможность создать соответствующую форму в конфигураторе, используя ранее рассмотренный механизм работы с формами.

### Самостоятельно

#### Создайте документ *Поступление денежных средств* (Рис. 7.4)



The screenshot shows a web-based form titled "Поступление денежных средств 000001 от 1...". At the top, there are navigation buttons: "Провести и закрыть" (highlighted in yellow), "Записать", "Провести", and "Еще" with a dropdown arrow. Below the title, the form contains several input fields and controls:

- "Поступление денежных средств №": 000001
- "от": 14.01.2014 12:00:00
- "Контрагент": MMM
- "Договор": MMM № 2
- "Сумма": 10 000,00
- "Управленческий":
- "Номер документа плательщика": мм0064
- "Дата документа плательщика": 08.01.2014
- "Способ оплаты":
  - Наличные
  - Безналичные
  - Электронные деньги
- "Комментарий": важное

Рис. 7.4

Обратите внимание, что пользователь имеет возможность (в определенных пределах) изменить внешний вид формы самостоятельно (Рис. 7.5 и Рис. 7.6). Для этого необходимо выбрать пункт меню *Изменить форму* из выпадающего списка, сформированного при нажатии на кнопку "Еще".

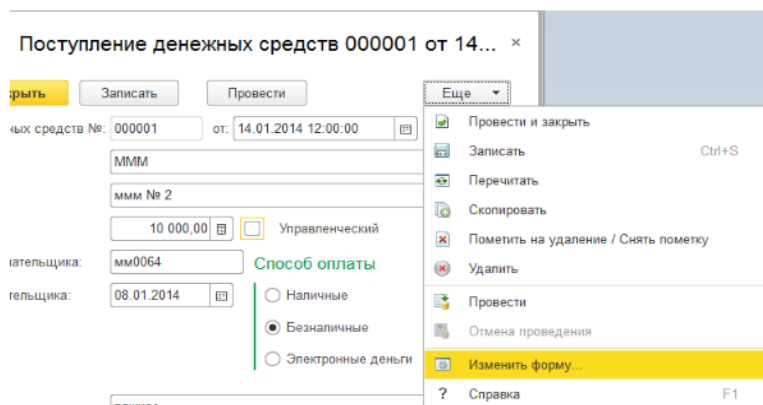


Рис. 7.5

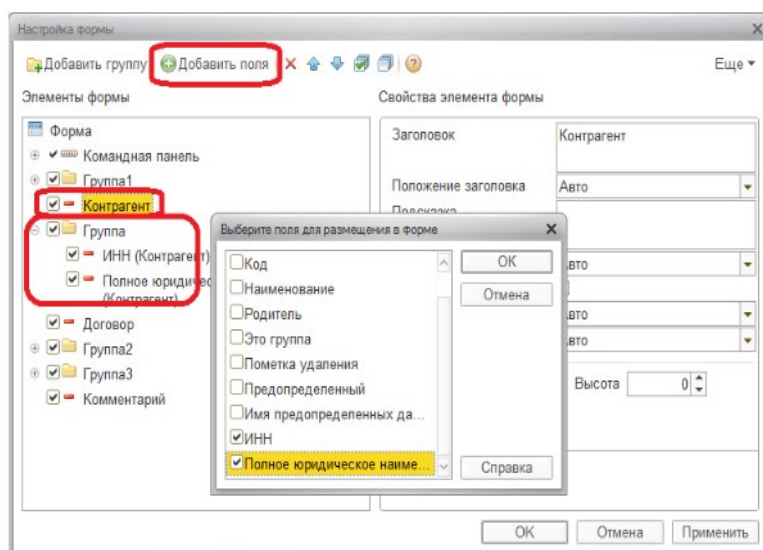


Рис. 7.6

### Реквизиты табличной части

Для документа *Приходная Накладная* необходимо создать табличную часть *Товары* с реквизитами *Товар*, *Количество*, *Цена* и *Сумма* с типами значений *СправочникСсылка.Номенклатура*, *Число* с точностью 3 и *Число* с точностью 2 соответственно. Создание табличной части документа также идентично созданию табличной части справочника. На закладке *Данные* в соответствующем окне с помощью контекстного меню создается табличная часть, а потом и ее реквизиты (Рис. 7.7).

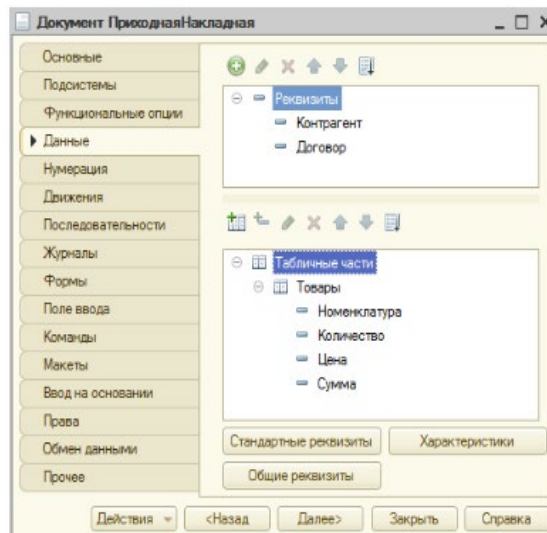


Рис. 7.7

При создании накладной необходимо обратить внимание на связь элементов *Договор* и *Контрагент*, а также производить автоматический расчет сумм. (Рис. 7.8)

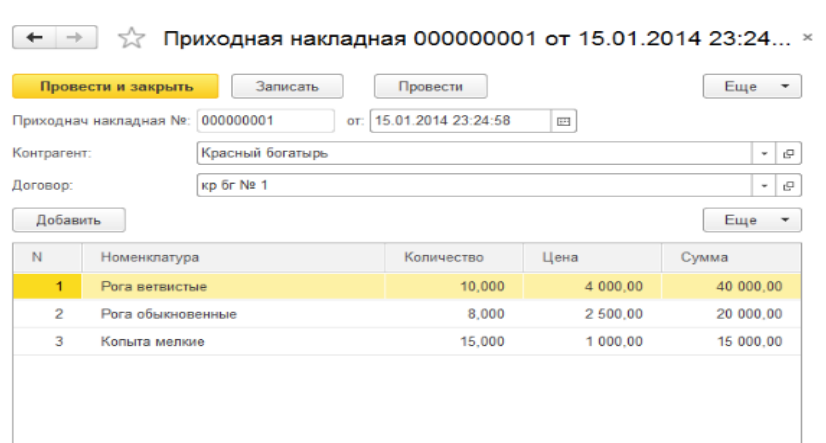


Рис. 7.8

Для расчета суммы необходимо задействовать обработчики событий *При изменении* для количества и цены (Рис. 7.9). Хотя результат этих обработчиков должен быть одинаков, для удобства приведено два разных способа написания этого действия.

&НаКлиенте

Процедура ТоварыКоличествоПриИзменении(Элемент)

Элементы.Товары.ТекущиеДанные.Сумма = Элементы.Товары.ТекущиеДанные.Цена \*

Элементы.Товары.ТекущиеДанные.Количество;

КонецПроцедуры

&НаКлиенте

Процедура ТоварыЦенаПриИзменении(Элемент)

СтрокаРасчета = Элементы.Товары.ТекущиеДанные;

СтрокаРасчета.Сумма = СтрокаРасчета.Цена \* СтрокаРасчета.Количество;

КонецПроцедуры

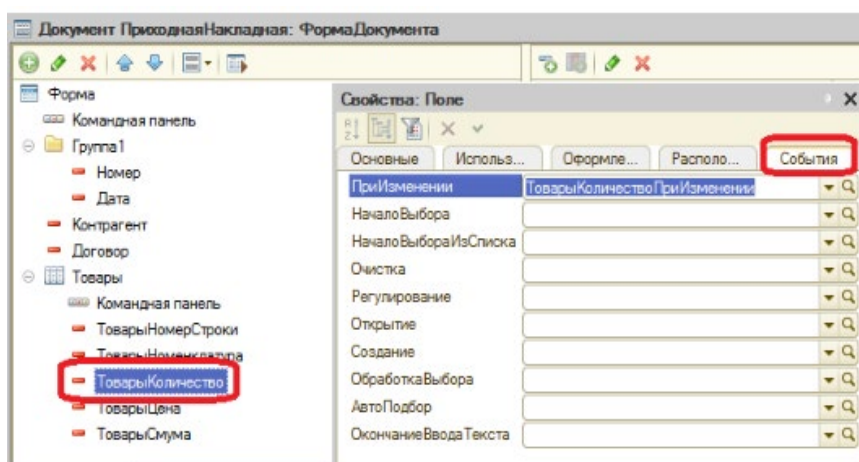


Рис. 7.9

Помимо расчета суммы по каждой строке, необходимо узнать общую сумму документа. Для чего в таблицу добавим подвал, а для колонок укажем, что именно в подвале будет отображено (Рис. 7.10)

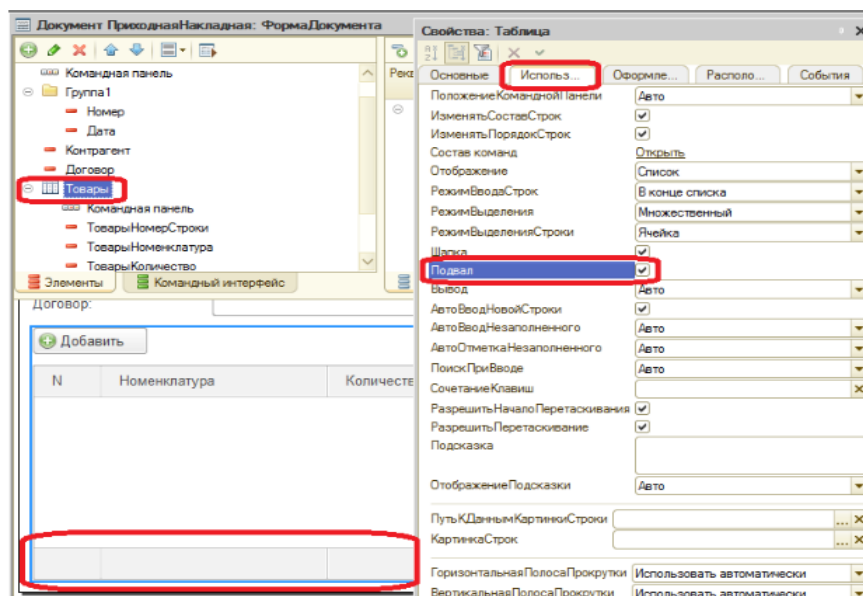


Рис. 7.10

В подвале можно отображать как заранее известный и фиксированный текст (Рис. 7.11), так и автоматически вычисляемые платформой данные. В нашем случае это будет итог по колонке сумма (Рис. 7.12).

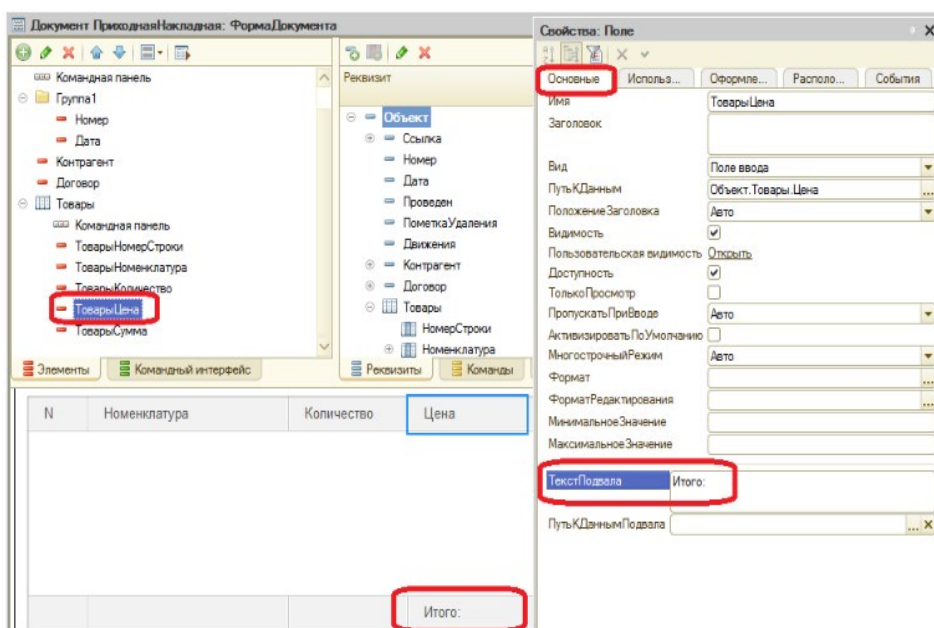


Рис. 7.11

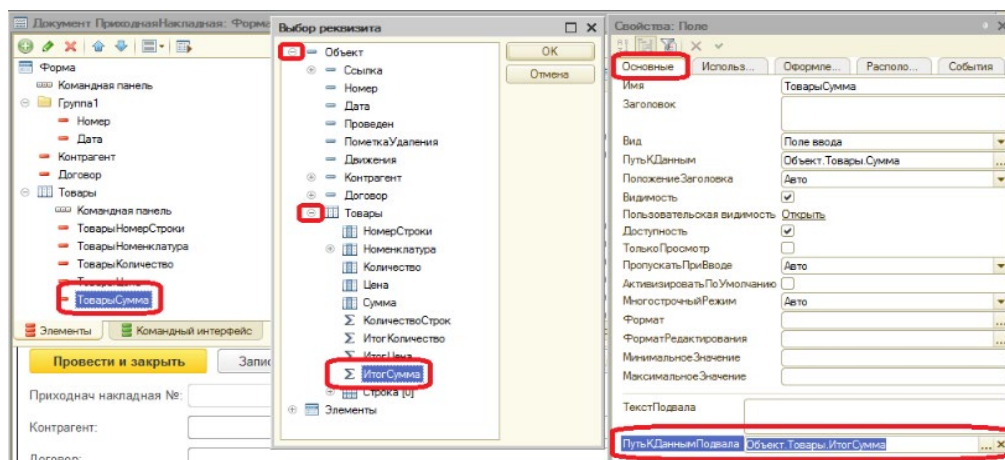


Рис. 7.12

В результате пользователь получит следующую форму (Рис. 7.13) для работы с документом приходная накладная.

← → ☆ Приходная накладная 000000001 от 15.01.2014 23:24:58 ×

Провести и закрыть    Записать    Провести    Еще ▾

Приходная накладная №: 000000001    от: 15.01.2014 23:24:58

Контрагент: Красный богатырь

Договор: кр бг № 1

Добавить    Еще ▾

N	Номенклатура	Количество	Цена	Сумма
1	Рога ветвистые	10,000	4 000,00	40 000,00
2	Рога обыкновенные	8,000	2 500,00	20 000,00
3	Копыта мелкие	15,000	1 000,00	15 000,00
			Итого:	75 000,00

Рис. 7.13

## Лабораторная работа №.8 Обработчик события. Функциональные опции

Для определения поведения прикладных объектов отличного от стандартного и создания собственных алгоритмов их поведения используется встроенный язык. Программный код выполняется в заранее известных ситуациях, называемых событиями.

В справочнике *Контрагенты* у нас храниться краткое и полное наименование контрагента. Логично предположить, что эти названия будут похожи. Следовательно, когда пользователь вводит краткое наименование желательно его же продублировать в полное, т.е. при событии изменение наименования необходимо выполнить определенные действия. Какие именно действия мы желаем выполнять при наступлении события, система сама предсказать не может. Необходимо самостоятельно написать процедуру, которая должна будет выполняться системой, и связать эту процедуру с тем событием, которое обрабатывается (Рис. 7.1).

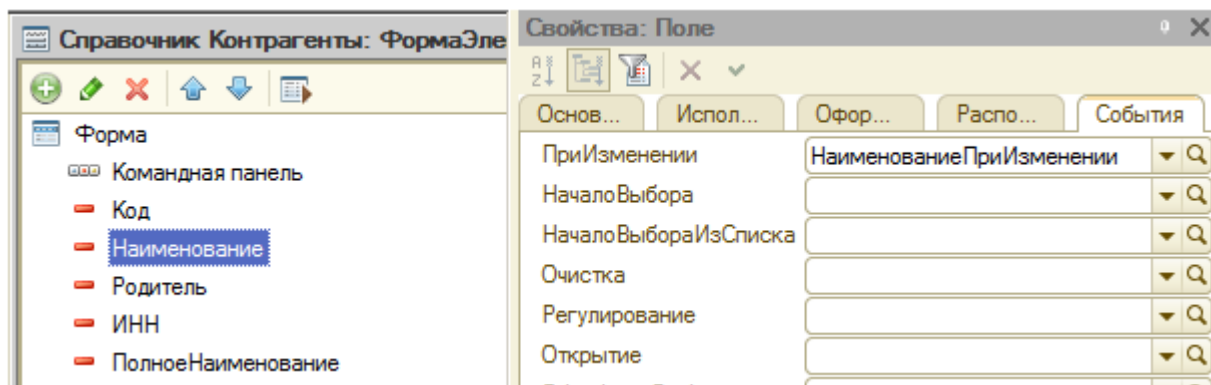


Рис. 7.1.

Текст процедуры обработчика события:

&НаКлиенте

Процедура НаименованиеПриИзменении(Элемент)

Если Объект.ПолноеНаименование = "" Тогда

Объект.ПолноеНаименование = Объект.Наименование;

КонецЕсли;

КонецПроцедуры

Помимо того, что имеется возможность выполнять какие-то действия при наступлении события, иногда можно управлять и соответствующими стандартными действиями платформы, т.е. фактически указывать, произойдет ли данное событие или нет.

Рассмотрим следующую ситуацию: необходимо для каждой номенклатурной позиции указать ставку НДС, причем у услуг ставка НДС



должна быть только 18%. В этом случае необходимо проверить при сохранении элемента, является ли данный элемент справочника услугой.

Создадим справочник, для хранения всех возможных ставок НДС, в котором ставку 18% сделаем преопределенной. В справочнике "Номенклатура" добавим соответствующий реквизит и выведем его на форму.

В том случае, когда элемент номенклатуры является услугой и ставка НДС не 18%, необходимо указать пользователю на недопустимость данной ситуации и не записывать элемент до тех пор, пока пользователь либо не укажет нужную ставку НДС, либо не укажет, что это товар.

Для решения этой задачи воспользуемся обработчиком события *ПередЗаписьюНаСервере* (Рис. 7.2)

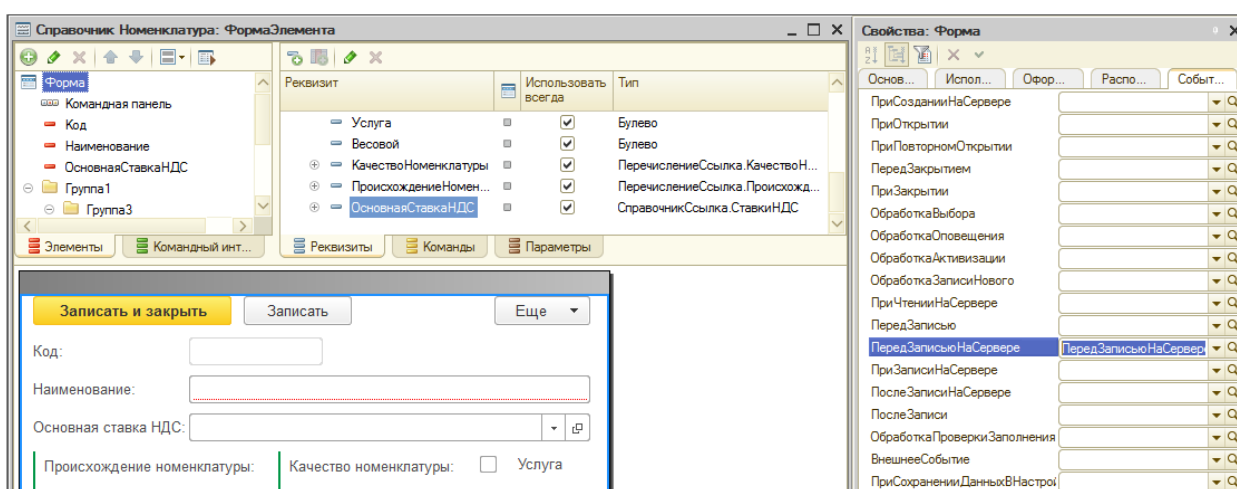


Рис. 7.2.

Текст процедуры:

&НаСервере

Процедура ПередЗаписьюНаСервере(Отказ, ТекущийОбъект, ПараметрыЗаписи)

Если ТекущийОбъект.Услуга И

(ТекущийОбъект.ОсновнаяСтавкаНДС <>

Справочники.СтавкиНДС.НДС18) Тогда

Сообщить("У услуги ставка НДС может быть только 18%");

Отказ = Истина;

КонецЕсли;

КонецПроцедуры

## Функциональные опции

Функциональные опции это механизм, который позволяет оперативно включить или отключить в пользовательском режиме выделенный функционал. Предоставим пользователю возможность самостоятельно определять необходимость использования валюты при продаже.

Для решения этой задачи начнем с добавления возможности указать валюту при продаже. Добавим справочник для хранения валют и соответствующие реквизиты в документы "РасходнаяНакладная" и "ПриходДенег". Эти реквизиты обязательно надо вывести на форму (Рис. 9.1).

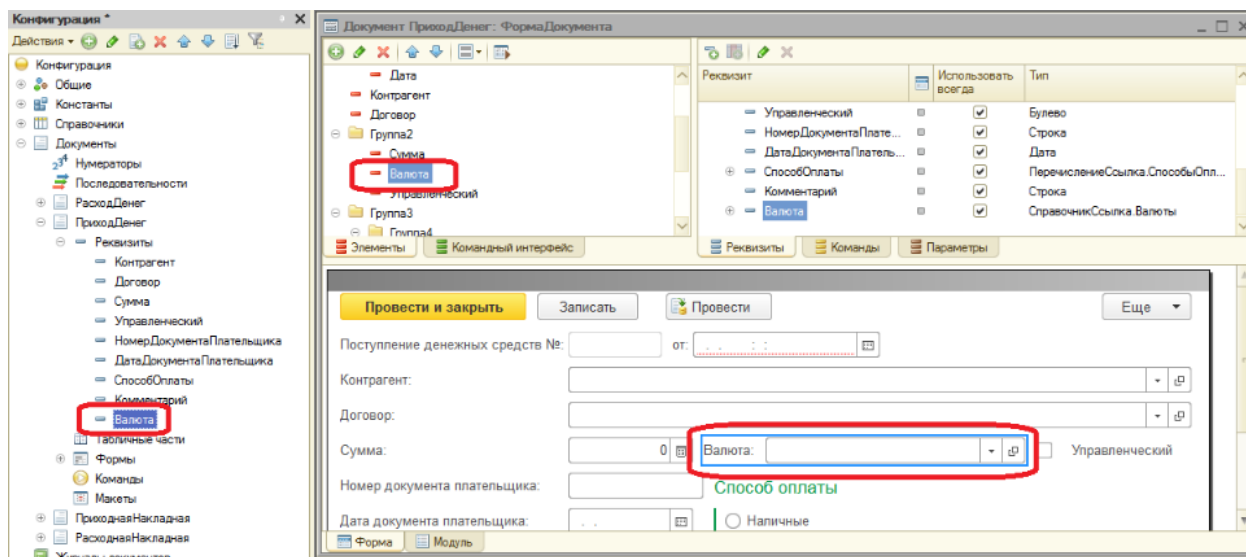


Рис. 9.1.

Для того, чтобы пользователь имел возможность указать, хочет ли он вести валютный учет или нет добавим новую константу "ВестиВалютный учетПродаж" с типом "Булево". После этого создаем саму функциональную опцию *ВалютныйУчет* (Рис. 9.2), в которой через свойство "Хранение" устанавливаем связь с нашей константой, определяющей необходимость наличия валюты.

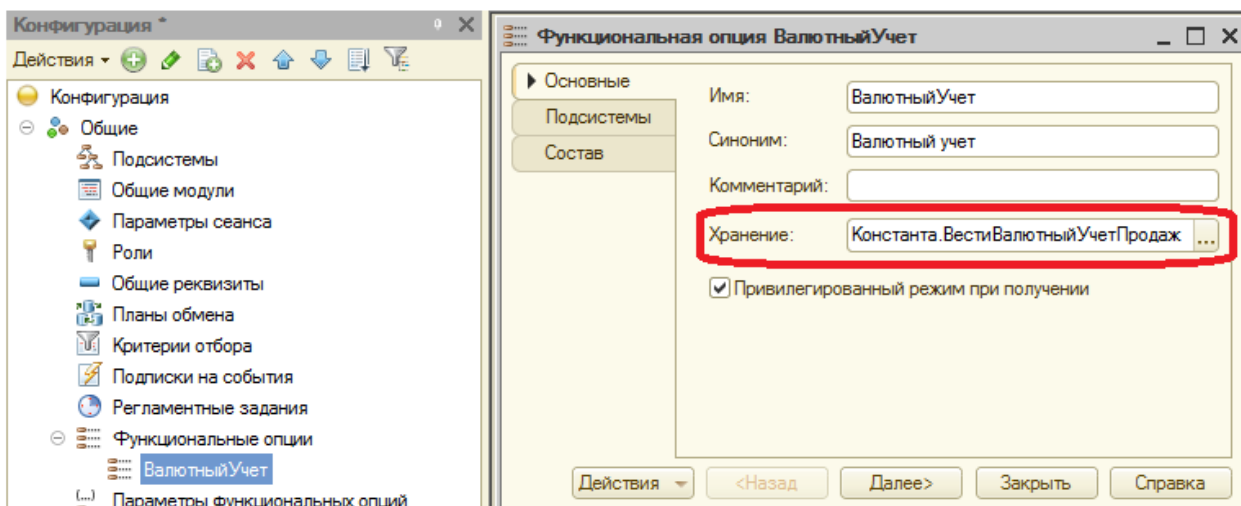


Рис. 9.2.

В свойствах функциональной опции включим в ее состав справочник и созданные реквизиты (Рис. 9.3).

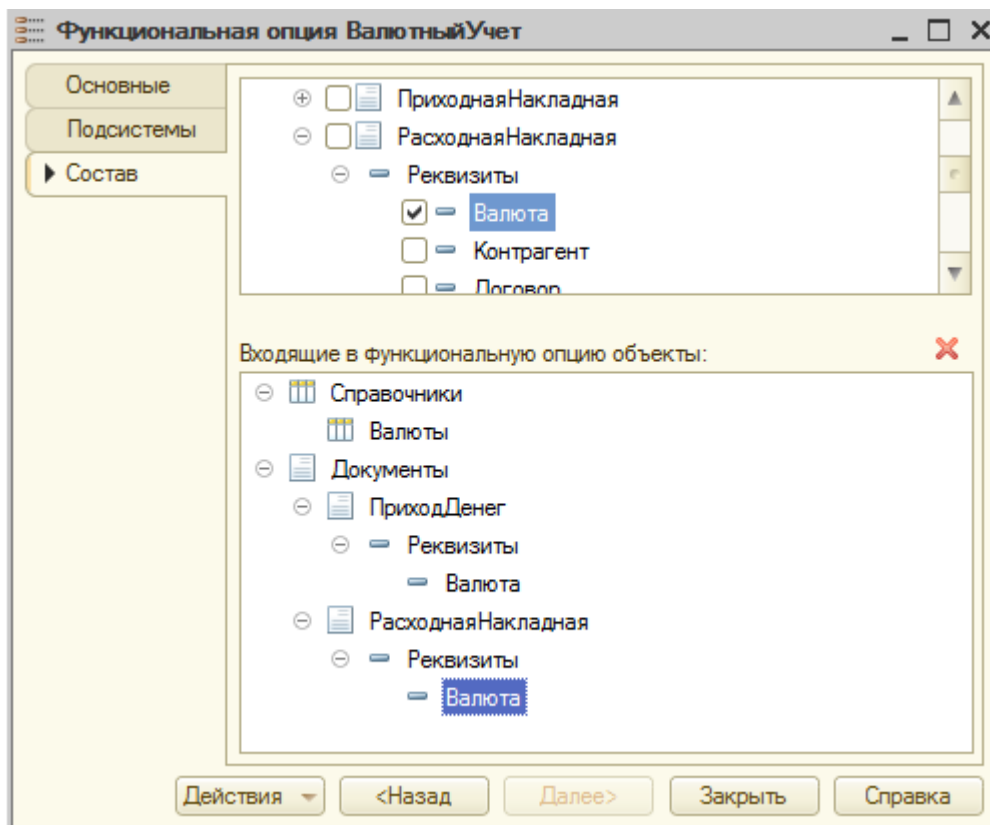


Рис. 9.3.

После этого перейдем в режим "1С:Предприятие" и проделаем следующие операции:

По умолчанию значение константы *ВалютныйУчет* будет *Ложь* (флажок снят). Откроем документ *ПриходДенег* и *РасходнаяНакладная*. В этом случае,

соответствующие реквизиты будут скрыты (Рис. 9.4). Кроме того, справочник *Валюта* будет недоступен для работы пользователя.

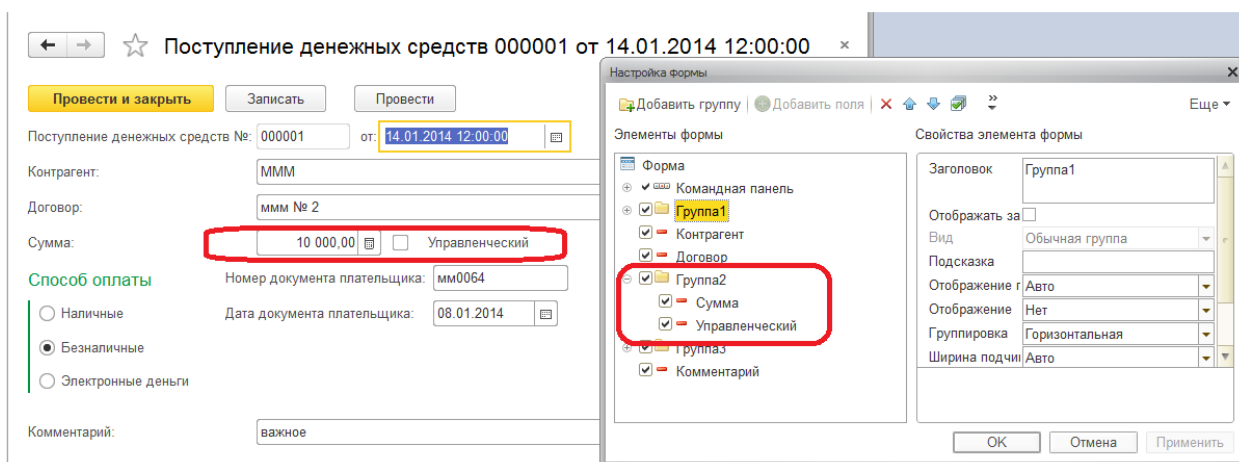


Рис. 9.4.

Теперь установим в константе значение "Истина" и убедимся, что для пользователя стали доступны объекты, отвечающие за валютный учет (Рис. 9.5).

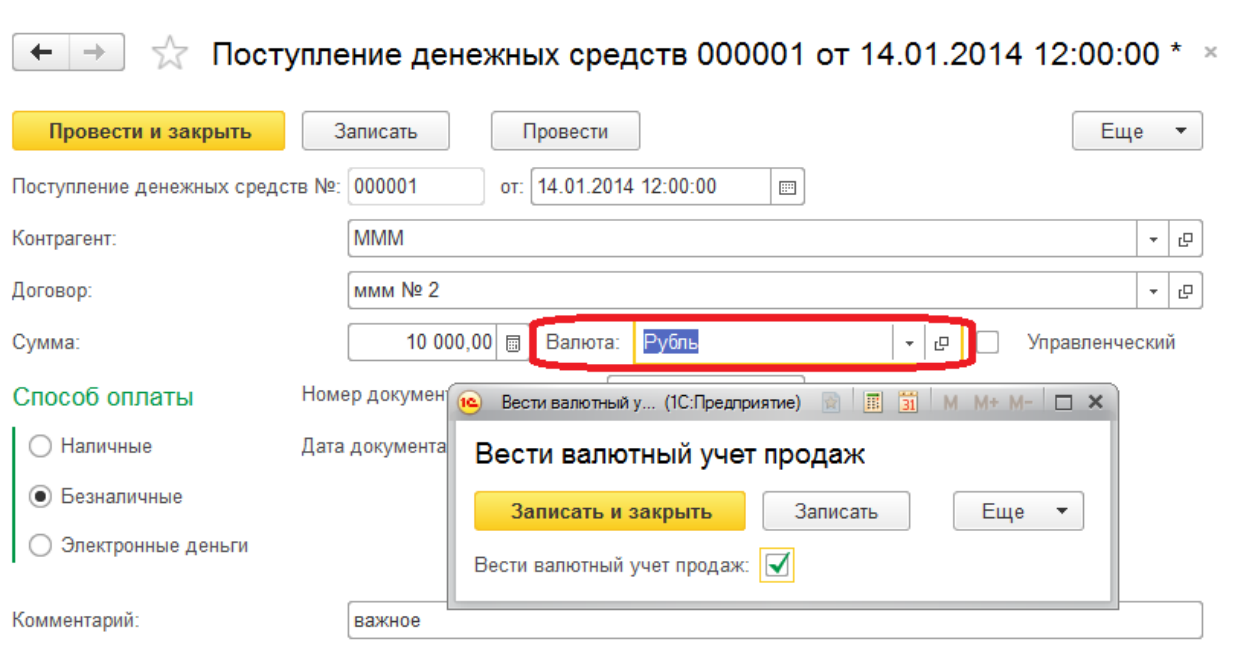


Рис. 9.5.

## Лабораторная работа №.9 Подсистемы и интерфейс. Регистр сведений

Подсистема позволяет группировать различные объекты конфигурации по какому-либо функциональному признаку.

Создадим подсистемы и укажем их состав (Рис. 10.1 и Рис. 10.2). Следует учитывать, что все объекты должны входить в состав какой-либо подсистемы.

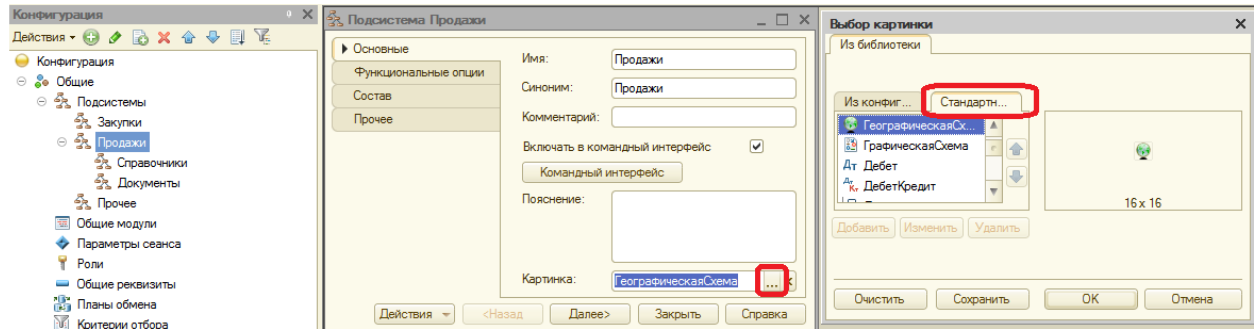


Рис. 10.1.

Один и тот же объект может входить в произвольное количество подсистем, каждая подсистема может включать в себя произвольное количество объектов. Кроме того, подсистемы можно создавать в виде иерархической древовидной структуры, используя подчиненные подсистемы. Для красоты каждой подсистеме мы сопоставим картинку, которую можно взять откуда-то из внешнего файла, либо использовать стандартные пиктограммки платформы.

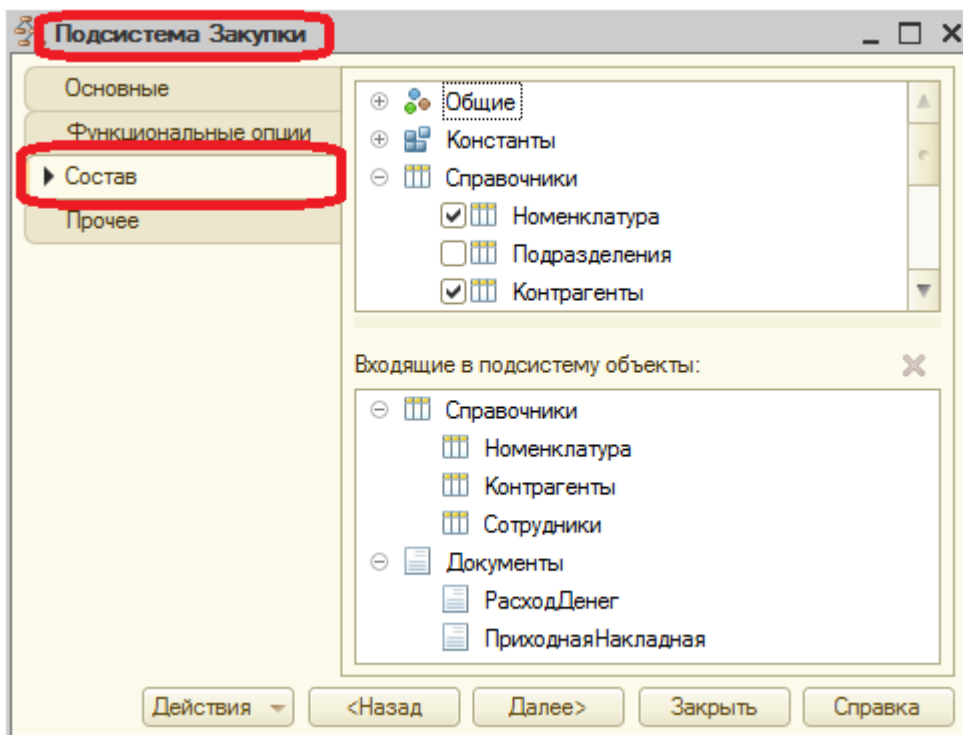


Рис. 10.2.

Для дальнейших действий нам потребуется у всех документов создать формы списка. Их внешний вид, в данном случае, значения не имеет. После чего переходим к настройкам командного интерфейса конфигурации (Рис. 10.3).

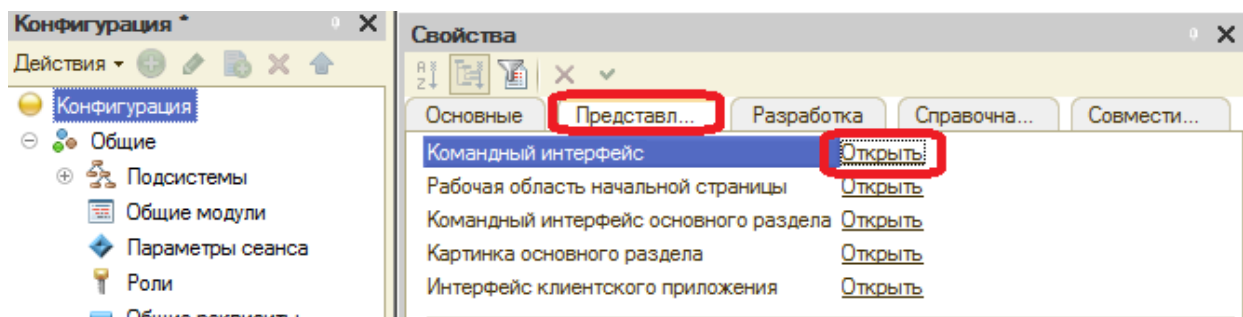


Рис. 10.3.

С помощью стандартных приёмов работы заполним рабочую область начальной страницы в соответствии с Рис. 11.4.

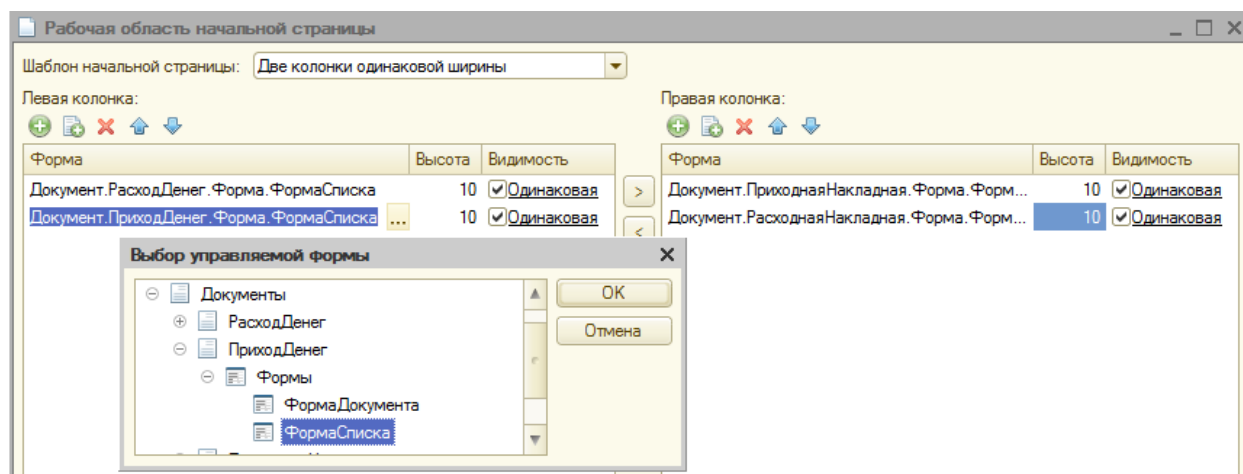
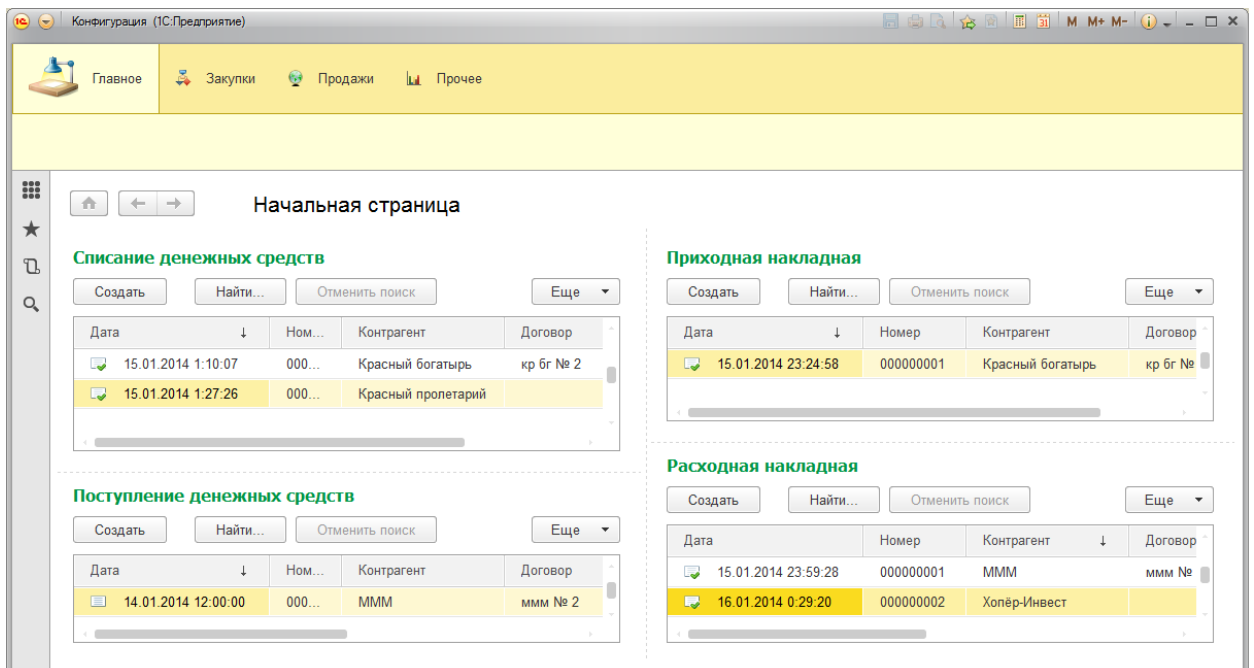


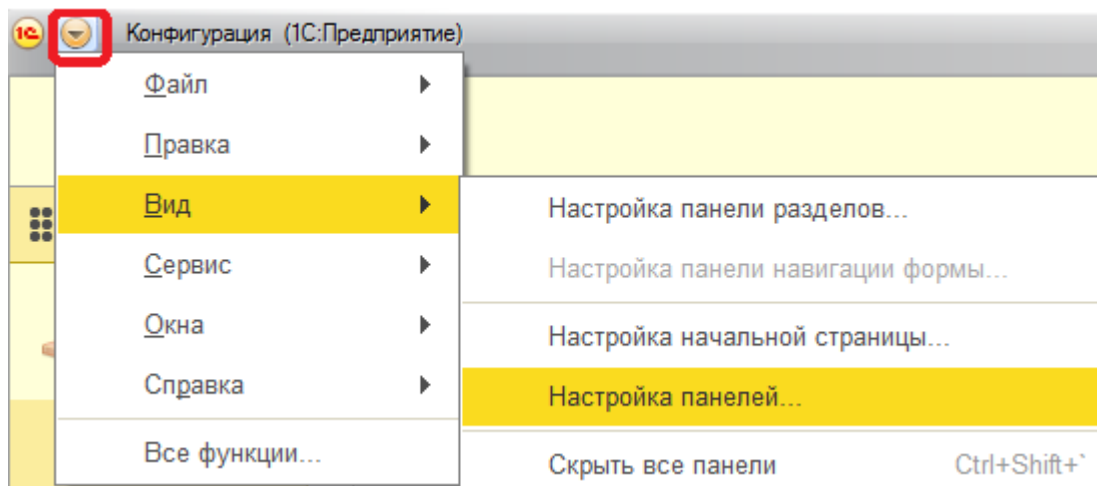
Рис. 10.4.

Теперь результат наших действий в пользовательском режиме сгруппирует объекты по разделам, в соответствии с подсистемами, а для начальной страницы позволит пользователю одновременно работать сразу с несколькими объектами (Рис.10.5).



**Рис. 10.5.**

Важной особенностью интерфейса является возможность его гибкой настройки пользователем. Причем можно настроить как правила работы интерфейса в целом (Рис. 10.6 и Рис. 10.7), так и особенности внешнего вида каждого раздела (Рис. 10.8).



**Рис. 10.6.**

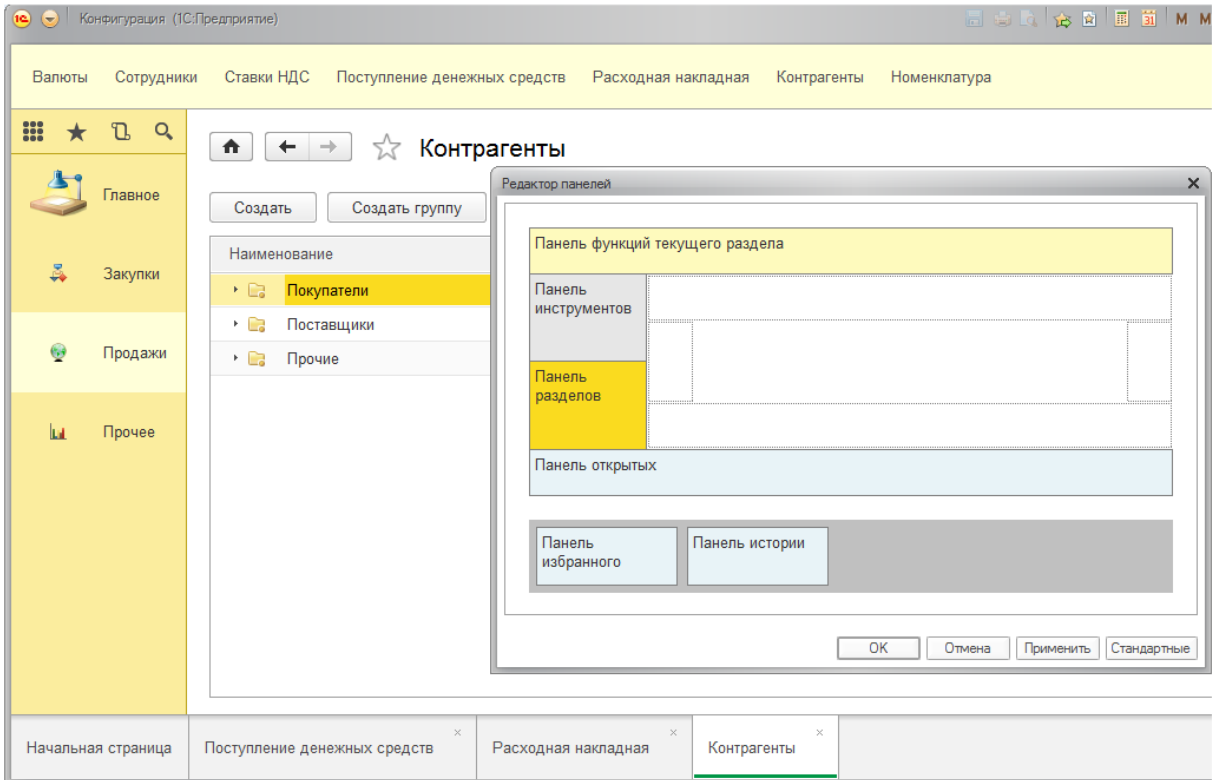


Рис. 10.7.

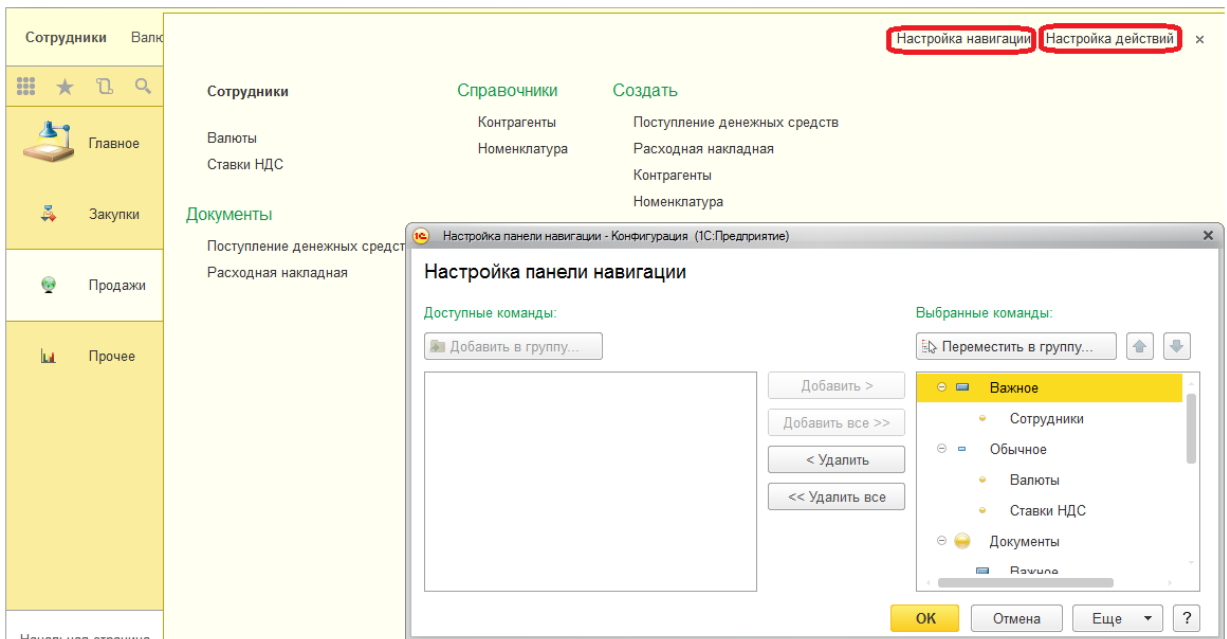


Рис. 10.8.



## Регистр сведений

Бывают ситуации, когда хранить информацию в справочнике неудобно. Например, если мы покупаем товар у одного и того же поставщика, то, скорее всего, цена на этот товар будет одной и той же (будет меняться достаточно редко). Было бы удобно, если бы имелась возможность эту цену хранить и автоматически подставлять в документы. Для решения этой задачи необходимо для каждого поставщика для каждого купленного у него товара хранить цену. Задача сильно усложнится, если мы решим хранить эту цену еще и в рамках договора. С одной стороны, это информация справочного характера, т.е. имеет смысл хранить ее в справочнике. С другой стороны, реализовать ее хранение достаточно сложно, причем далеко не все возможности, предоставляемые системой для справочника, нам в этом случае понадобятся. Кроме того, желательно иметь возможность быстро найти нужное нам значение. Для решения подобного рода задач предназначен специальный объект *Регистр сведений*.

Работа с регистром сведений похожа на работу с остальными объектами. В зависимости от того, какого рода информация будет храниться в регистре, он должен иметь соответствующую структуру. Если необходимо хранить информацию различного вида, то для этого можно создать произвольное количество регистров сведений.

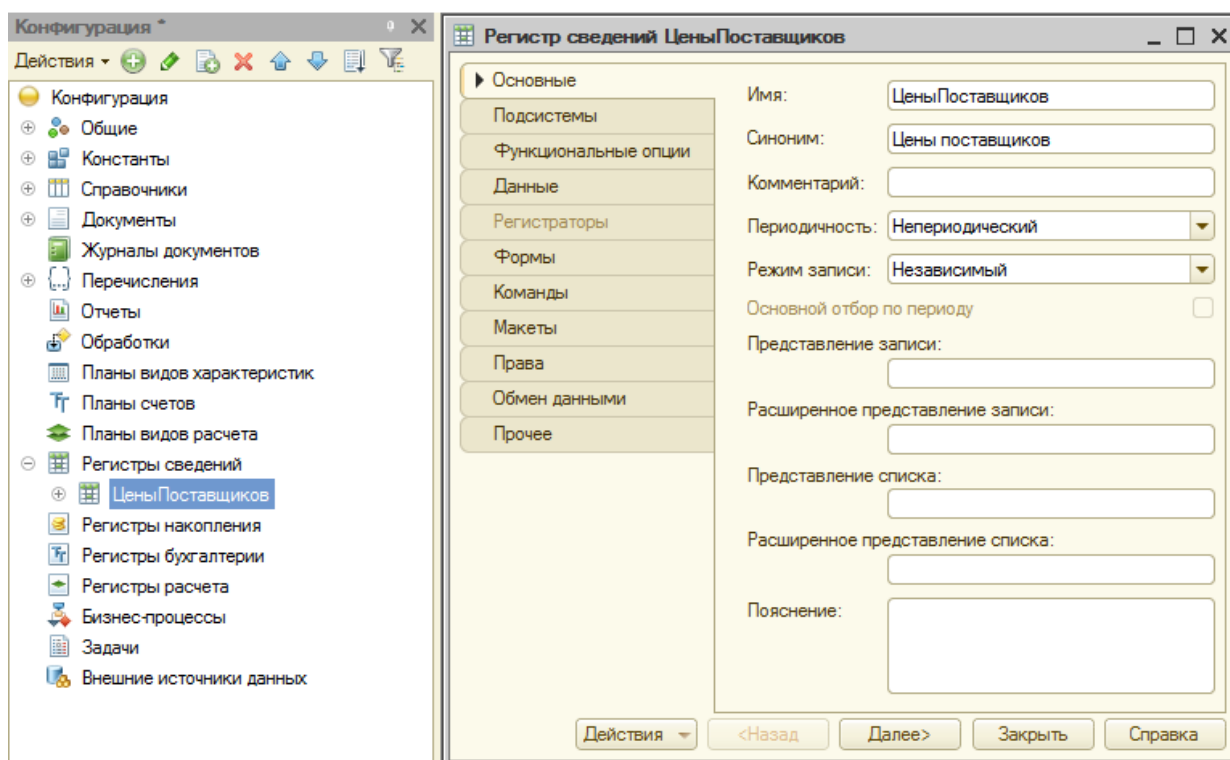


Рис. 11.1.

Создание нового регистра аналогично созданию нового документа или справочника (Рис. 11.1). Кроме того, как только мы стали использовать механизм подсистем для построения интерфейса, необходимо при создании объекта сразу же предусмотреть возможность работы пользователя с этим объектом, т.е.

отобразить его на интерфейсе, для чего включить в соответствующую подсистему (Рис. 11.2).

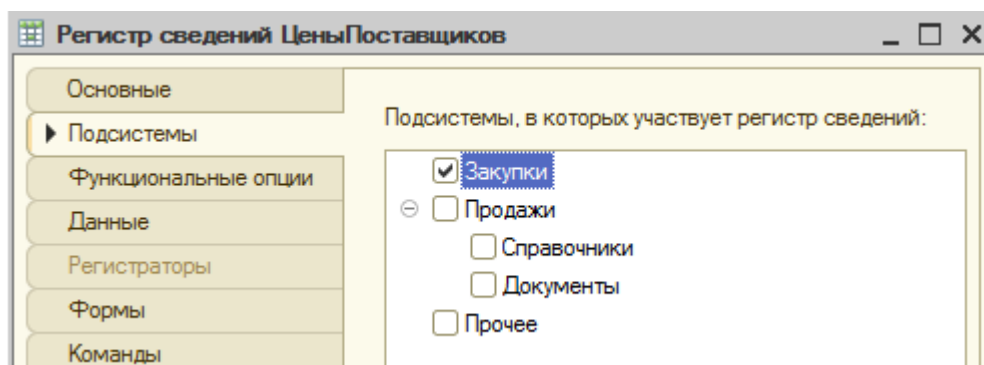


Рис. 11.2.

Структуру регистра можно задать на закладке *Данные*. (Рис. 11.3) То, что мы собственно храним в регистре, находится в ресурсе, а те разрезы, в которых нас это информация интересует, находятся в измерениях. В нашем случае необходимо хранить цену для каждого товара, купленного по каждому договору и каждого поставщика, т.е. цена в разрезе поставщиков, договоров и номенклатуры. Следовательно, цена должна быть ресурсом, а контрагент, договор и номенклатура – измерениями. В этом случае система будет контролировать, что для каждого уникального набора значений измерений может быть только одно значение ресурса, т.е. для одного и того же товара по одному договору для одного поставщика может быть только одна цена. Естественно, что необходимо и для измерений и для ресурсов указать соответствующий тип данных.

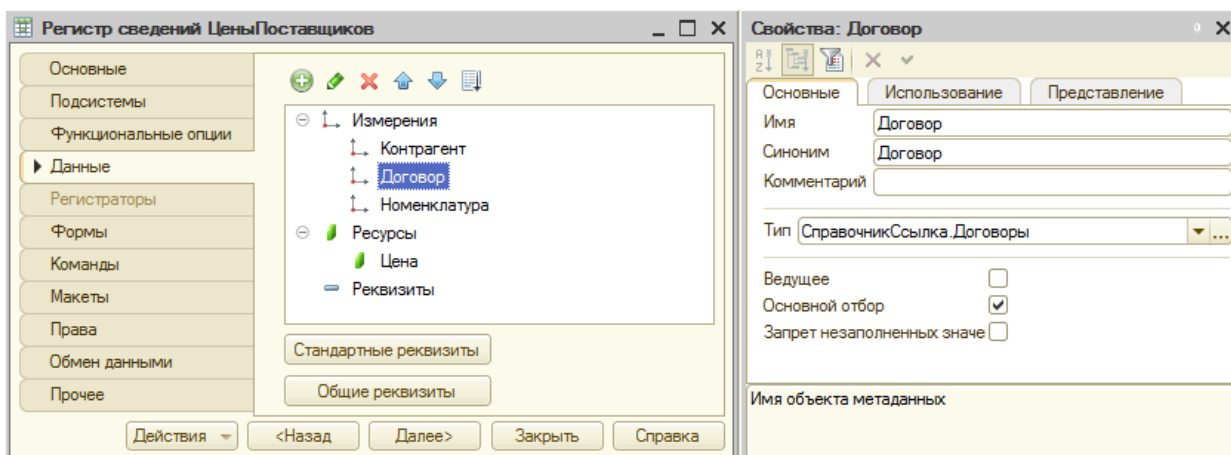










Рис. 11.3.

Информацию в регистр пользователь может внести самостоятельно в режиме "1С:Предприятие", используя соответствующую форму списка (Рис. 11.4).





**Цены поставщиков**
×

Контрагент ↓	Договор	Номенклатура	Цена
 Красный богатырь	кр бг № 1	Копыта мелкие	1 000,00
 Красный богатырь	кр бг № 1	Рога обыкновенные	2 000,00
 Красный богатырь	кр бг № 2	Рога обыкновенные	2 050,00
 Красный пролетарий	кр пр № 7	Рога обыкновенные	2 020,00

**Рис. 11.4.**

Для того, чтобы цена автоматически подставлялась в документе "Приходная накладная", необходимо задействовать обработчик *ТоварыНоменклатураПриИзменении*. Чтобы получить нужную нам цену, необходимо из всей хранящейся в регистре информации извлечь только ту, которая нам нужна в данный момент, т.е. каким-то образом ее отфильтровать. Сделать это можно с помощью специального объекта *Структура*. Далее, применив такой фильтр к нашему регистру, можно будет получить интересующую нас цену.

Ниже приведен соответствующий программный код:

&НаСервере

Функция ПолучитьЦену(НоменклатураДокумента)

СтруктураОтбора = Новый Структура;

СтруктураОтбора.Вставить("Контрагент", Объект.Контрагент);

СтруктураОтбора.Вставить("Договор", Объект.Договор);

СтруктураОтбора.Вставить("Номенклатура", НоменклатураДокумента);

РезультатОтбора =

РегистрыСведений.ЦеныПоставщиков.Получить(СтруктураОтбора);

Возврат РезультатОтбора.Цена;

КонецФункции

&НаКлиенте

Процедура ТоварыНоменклатураПриИзменении(Элемент)

СтрокаРасчета = Элементы.Товары.ТекущиеДанные;

СтрокаРасчета.Цена = ПолучитьЦену(СтрокаРасчета.Номенклатура);

ТоварыЦенаПриИзменении(Элемент)

## КонецПроцедуры

Помимо того, что регистр сведений позволяет быстро извлекать информацию в различных разрезах, он позволяет решать и еще ряд задач. В некоторых случаях, нас интересует не просто какая-то информация, но и история ее изменения, например курс валюты. В таком случае, для каждой валюты на каждый день необходимо хранить ее курс. Исходя из того, что уже рассмотрено выше, можно добавить в регистр сведений новое измерение *Дата*, однако, поскольку такая ситуация встречается достаточно часто, то эта задача решается на системном уровне. В свойствах регистра сведений можно указать, что он является периодическим (Рис. 11.5).

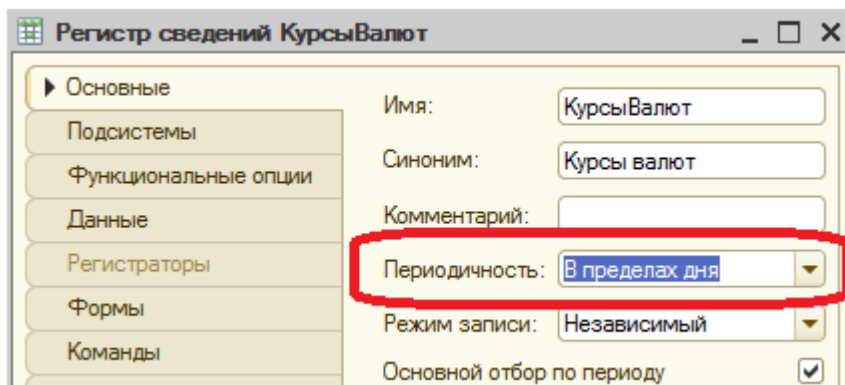


Рис. 11.5.

В зависимости от того, какая указана периодичность регистра сведений, в нем можно хранить информацию уникальную для различных периодов, например, одно значение на каждый год, или одно значение на месяц, или на день и т.д. Поскольку валюта у нас используется только в зависимости от желания пользователя, то и логично регистр, связанный с валютой также использовать в зависимости от функциональной опции (Рис. 11.6).

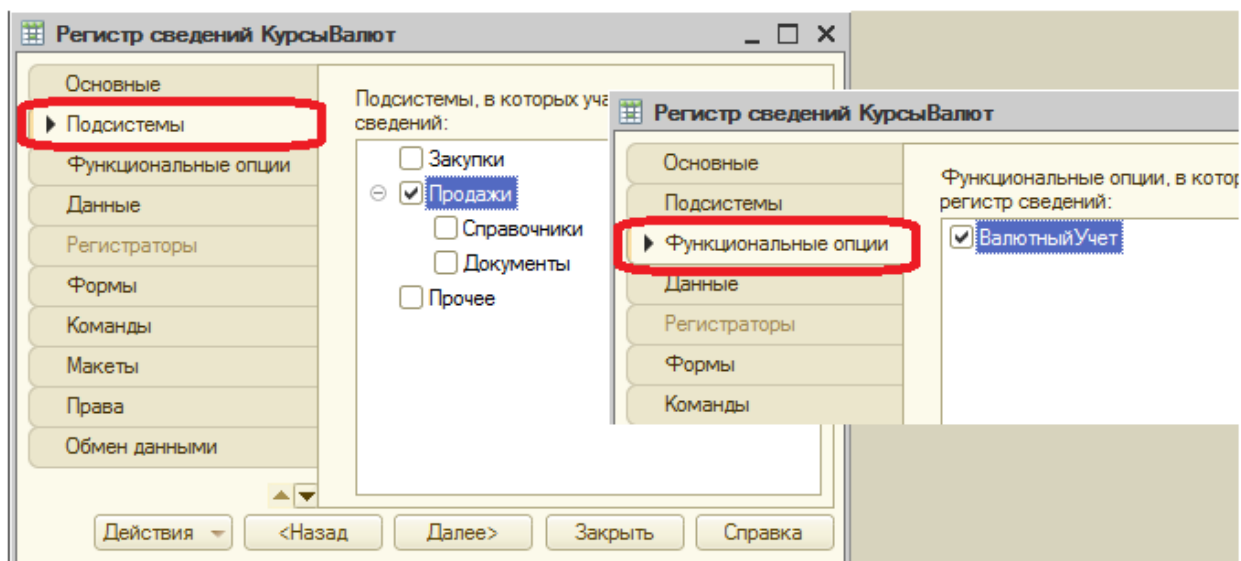


Рис. 11.6.

Сама структура регистра будет состоять из 1 измерения и 1 ресурса (Рис. 11.7). Поскольку наш регистр является периодическим, то наряду с существующими измерениями добавляется еще одно *Период* (Рис. 11.8), и система предоставляет возможность быстро извлечь интересующую нас информацию на произвольный момент времени.

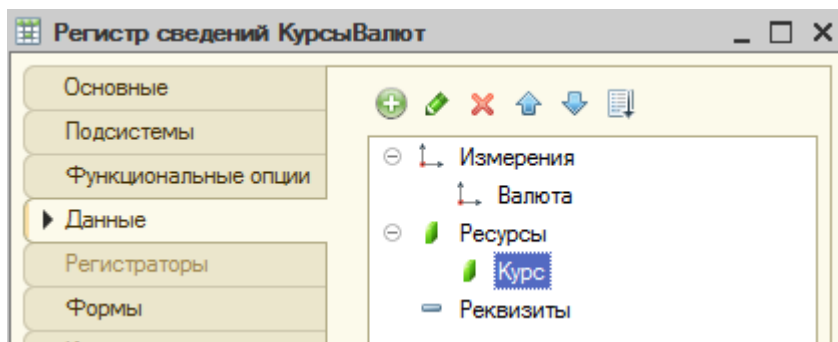


Рис. 11.7.

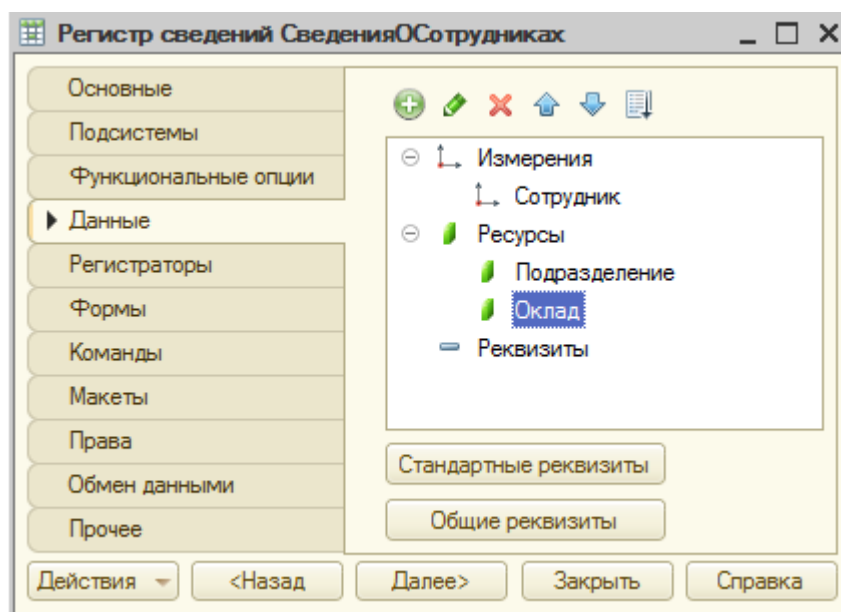
Период	Валюта	Курс
01.01.2000	Рубль	1,0000
01.01.2014	Доллар	32,6587
10.01.2014	Доллар	33,1547
15.01.2014	Евро	45,4139
18.01.2014	Евро	45,5175
18.01.2014	Доллар	33,4343

Рис. 11.8.

Еще одна задача, которую позволяет решить регистр сведений, это взаимосвязь хранимой информации с документами. Например, если мы хотим хранить в системе информацию о том, в каком подразделении работает сотрудник и какой у него оклад, причем, назначение сотрудника в подразделение и установка его ставки оформляется соответствующим кадровым приказом. С одной стороны, если мы сделаем документ и заполним реквизиты справочника, то вся эта информация в систему будет занесена. Однако, в том случае, когда мы удаляем соответствующий документ, информация в справочнике остается, что уже не соответствует положению вещей. Для решения подобного рода задач существует механизм проведения документов. Этот механизм позволяет установить взаимосвязь между данными регистра и документом. Происходит это следующим образом: для записанного документа может быть присвоен статус "Проведен" (определяется значением системного реквизита *Проведен*). Изменение статуса документа является событием, для которого есть

соответствующий обработчик проведения, где можно описать, какие записи в какие регистры формирует данный документ. В том случае, если этот статус снимается (в т.ч. при пометке на удаление), то из регистров автоматически удаляются записи, связанные с этим документом. Статус документа отображается соответствующей пиктограммой и может быть изменен пользователем.

Для решения этой задачи, создадим новую подсистему "Кадры", которая будет содержать данные, необходимые для ведения кадрового учета, после чего потребуется создать регистр сведений "СведенияОСотрудниках". Будем считать, что в нашей организации нет совместителей, т.е. сотрудник может работать только в одном подразделении и получать там какой-то оклад. Структура регистра сведений должна быть следующей: измерение *Сотрудник* и ресурсы *Подразделение* и *Оклад*. При этом, следует обратить внимание на то, что в ресурсе может храниться значение ссылочного типа, т.е. в нашем случае ссылка на справочник *Подразделения* (Рис. 11.9).



**Рис. 11.9.**

Далее, наверное, наши сотрудники со временем могут переходить из подразделения в подразделения или получать другой оклад, поэтому сделаем регистр периодическим, с периодичностью месяц. Далее для того, чтобы в наш регистр попадали данные из документа необходимо указать способ записи "Подчинение регистратору" (Рис. 11.10). В этом случае у регистра появится дополнительное поле "Регистратор", в котором будет указан документ, сформировавший эти записи.

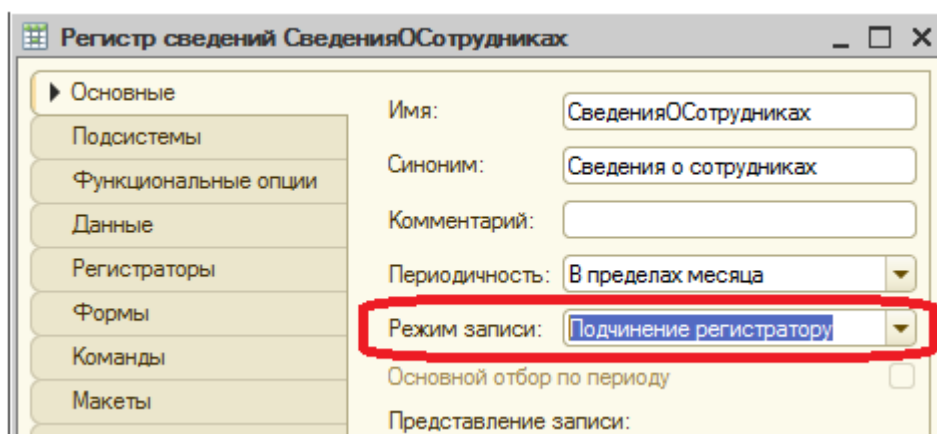


Рис. 11.10.

Теперь, когда информация о месте работы сотрудника храниться в регистре сведений, логично удалить реквизит "Подразделение" в справочнике "Сотрудники".

Создадим документ *Кадровый приказ*. Данный документ будет отражать кадровые изменения в подразделении, поэтому будет иметь реквизит "Подразделение" и табличную часть, с указанием сотрудника и его оклада (Рис. 11.11).

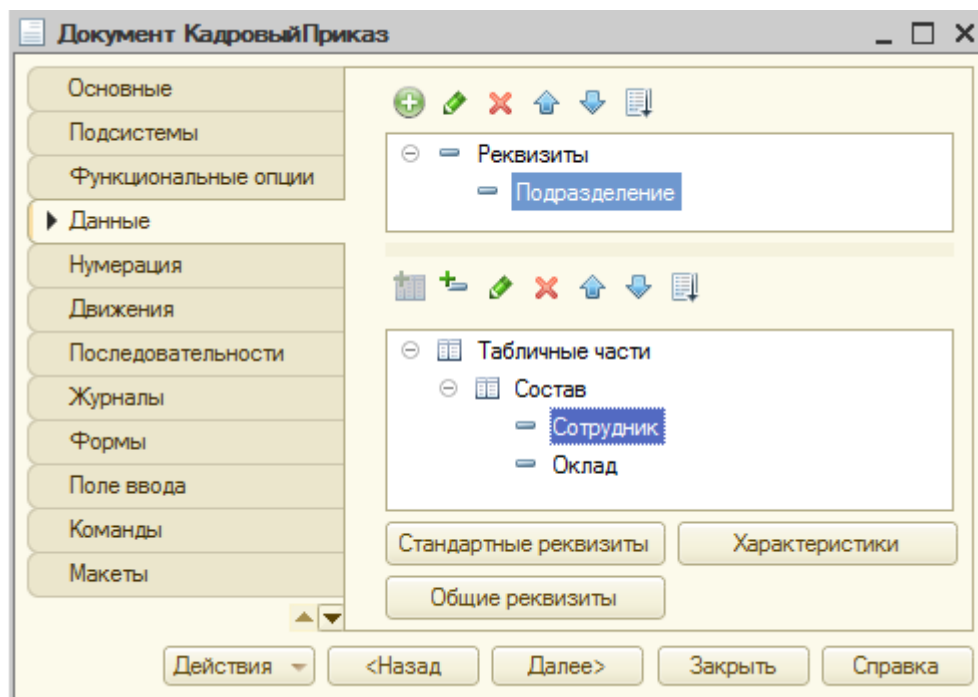


Рис. 11.11.

Чтобы документ формировал записи в регистр необходимо на закладке *Движения* настроить следующие свойства (Рис. 11.12):

*Проведение* – разрешить (для данного документа может быть статус "проведен")

*Удалять движения* – автоматически (при отмене проведения данного документа, сформированные им записи в регистрах будут удалены)

Указать те регистры, в которые будут сформированы записи.

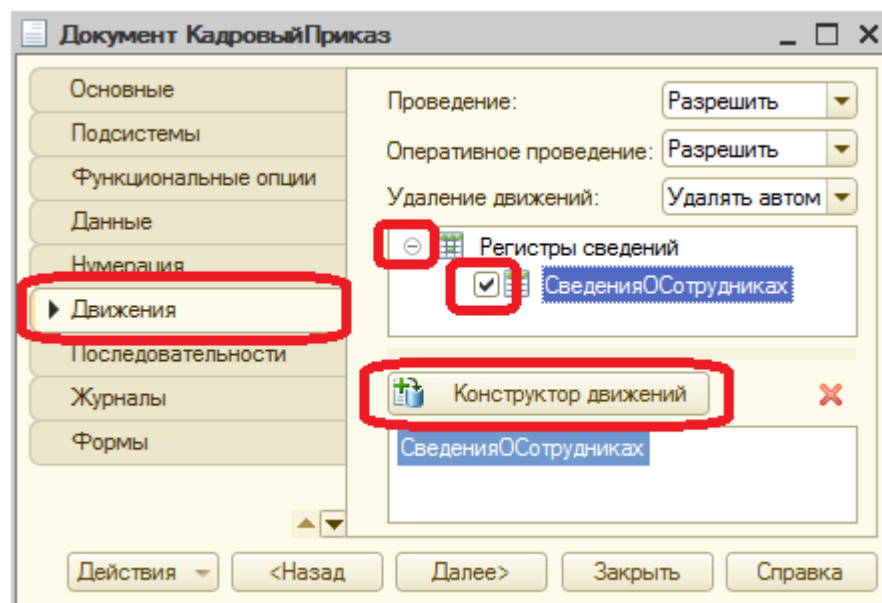


Рис. 11.12.

После этого с помощью конструктора сформируем код, который собственно и будет определять, какую именно информацию и куда мы будем записывать. Для этого нажмем кнопку *Конструктор движений*, далее можно вручную, а можно автоматически с помощью кнопки *Заполнить выражения* (Рис. 11.13) сопоставить данные документа и регистра.

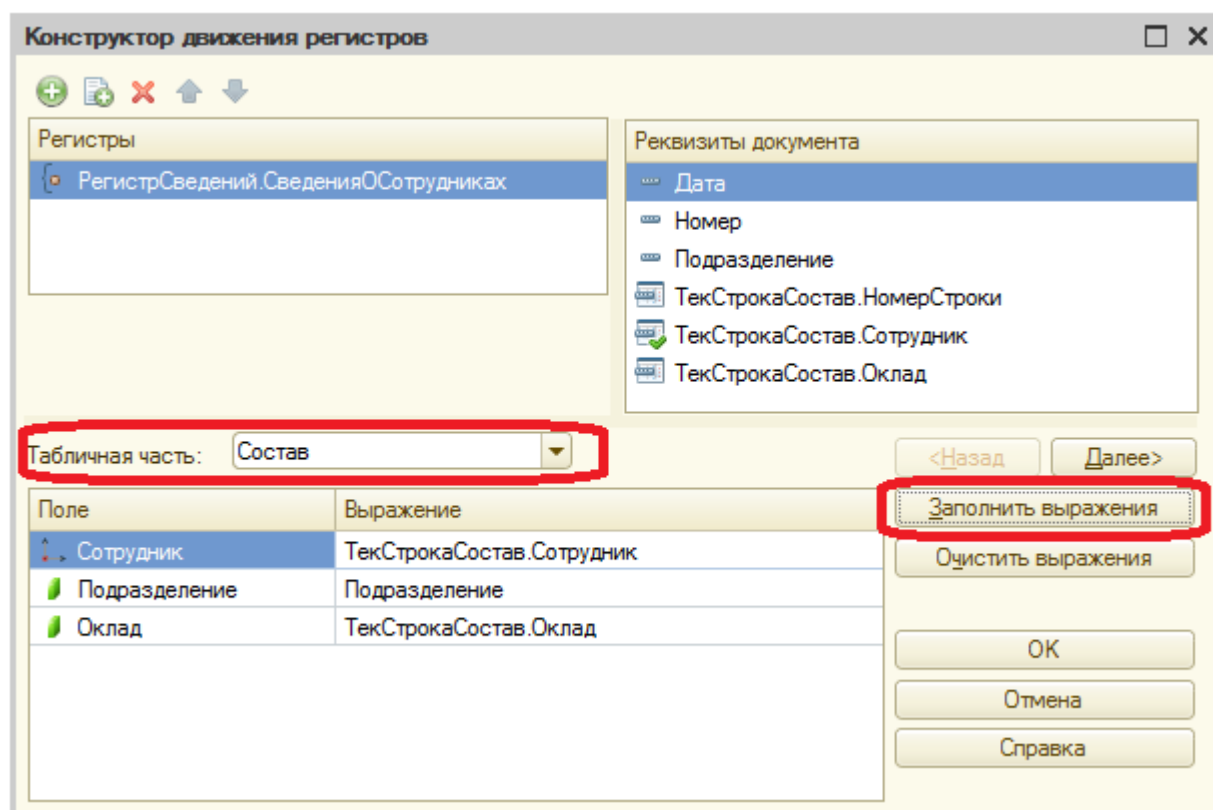


Рис. 11.13.



В результате работы конструктора в модуле документа будет создана процедура обработчик события проведение.

Процедура ОбработкаПроведения(Отказ, Режим)

```
//{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

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

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

```
Движения.СведенияОСотрудниках.Записывать = Истина;
```

```
Для Каждого ТекСтрокаСостав Из Состав Цикл
```

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

```
    Движение.Период = Дата;
```

```
    Движение.Сотрудник = ТекСтрокаСостав.Сотрудник;
```

```
    Движение.Подразделение = Подразделение;
```

```
    Движение.Оклад = ТекСтрокаСостав.Оклад;
```

```
КонецЦикла; //}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

КонецПроцедуры

В режиме "1С:Предприятие" у пользователя появляется возможность провести или снять с проведения проведенный документ (Рис. 11.14).

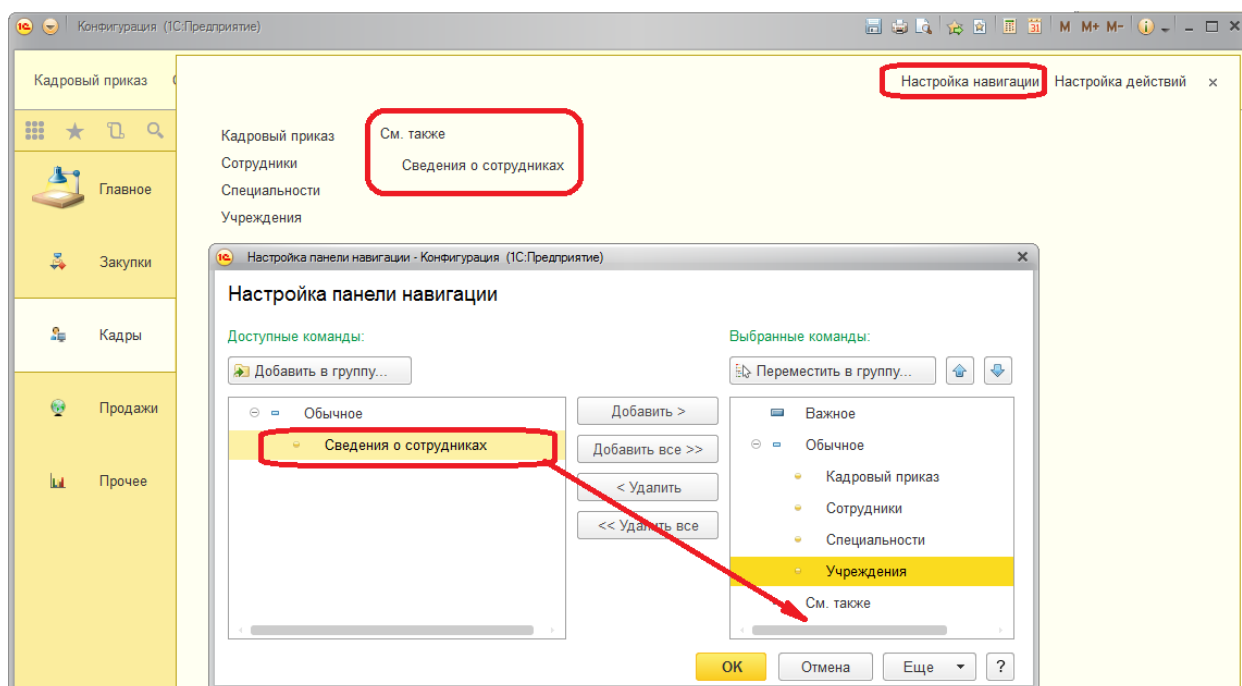













Рис. 11.14.

Обратите внимание, по умолчанию наш регистр не отобразился автоматически в интерфейсе. Чтобы посмотреть результат наших действий, самостоятельно добавьте регистр в интерфейс (Рис. 11.15).



☆ Кадровый приказ
×

Дата	↓	Номер	Подразделение
	06.11.2013 12:00:00	000000001	Администрация
	10.11.2013 12:00:00	000000002	Продажи
	19.11.2013 12:00:00	000000003	Секретариат



☆ Сведения о сотрудниках
×

Период	↓	Регистратор	Ном...	Сотрудник	Подразделение	Оклад
	01.11.2013	Кадровый приказ 000000001 от 06.11.2013 ...	1	Бендер О.И.	Администрация	100 000,00
	01.11.2013	Кадровый приказ 000000002 от 10.11.2013 ...	2	Паниковский М.С.	Продажи	50 000,00
	01.11.2013	Кадровый приказ 000000002 от 10.11.2013 ...	1	Балаганов А.П.	Продажи	45 000,00
	01.11.2013	Кадровый приказ 000000003 от 19.11.2013 ...	1	Козлевич А.К.	Секретариат	50 000,00

**Рис. 11.15.**

## Лабораторная работа №.10 Запрос

Запрос это раздел встроенного языка, который позволяет получить информацию из базы данных в виде выборки, сформированной по заданным правилам.

Чтобы познакомиться с тем, как формировать запрос, и что получается в результате его работы, воспользуемся стандартной обработкой "Консоль запросов". Данная обработка является технологической, как получить окончательный результат будет изложено в следующем разделе. Для работы обработки необходимо будет через пункт меню *Сервис-Параметры* выполнить следующие настройки (Рис. 12.1):

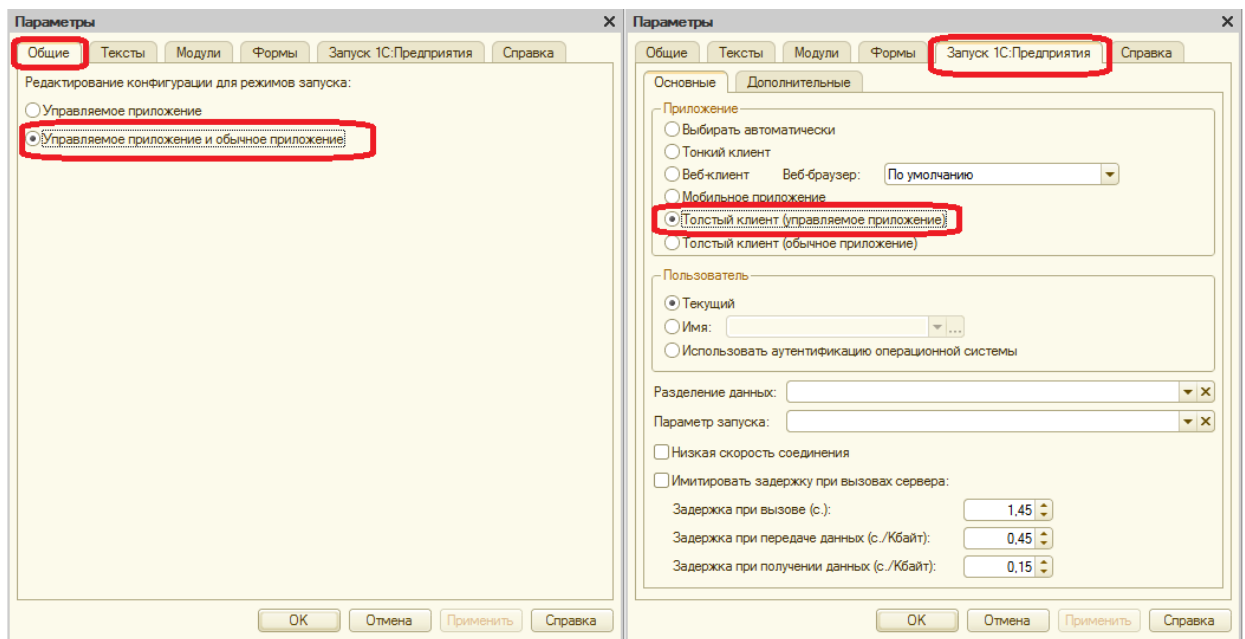


Рис. 12.1.

Только после этих настроек в свойствах конфигурации станет доступной следующая нужная нам настройка (Рис. 12.2).

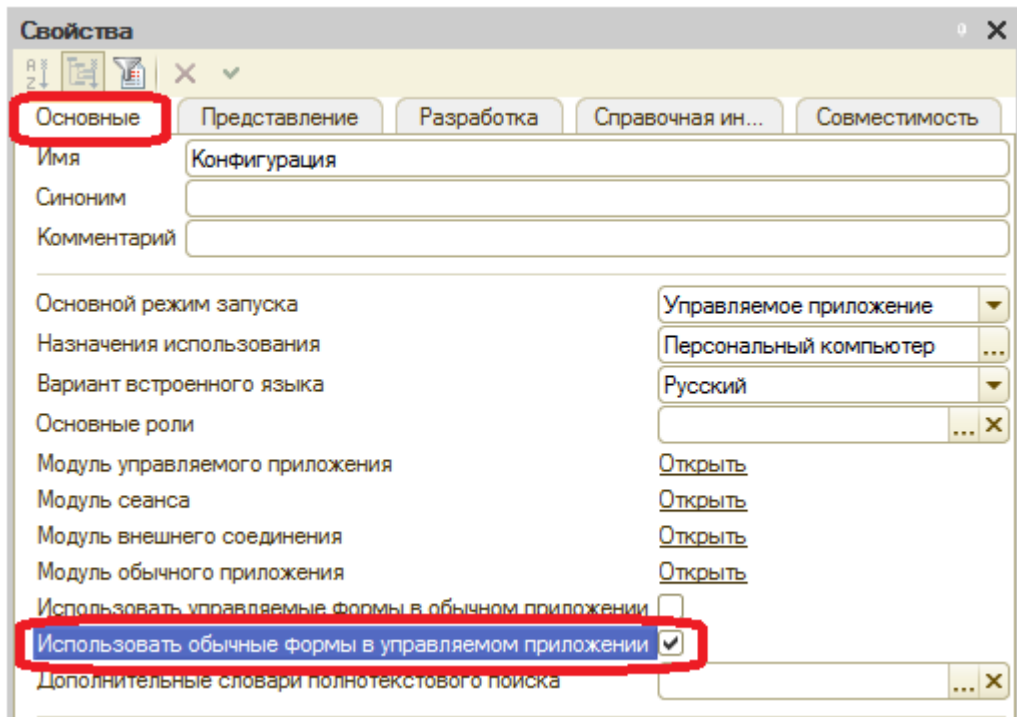


Рис. 12.2.

Откроем обработку в режиме "1С:Предприятие" и создадим новый запрос, текст которого удобнее формировать с помощью конструктора запросов (Рис. 12.3). Вызов конструктора происходит через контекстное меню, вызываемое по правой кнопке мыши. С помощью этого запроса мы попробуем узнать, какие у нас есть поставщики.

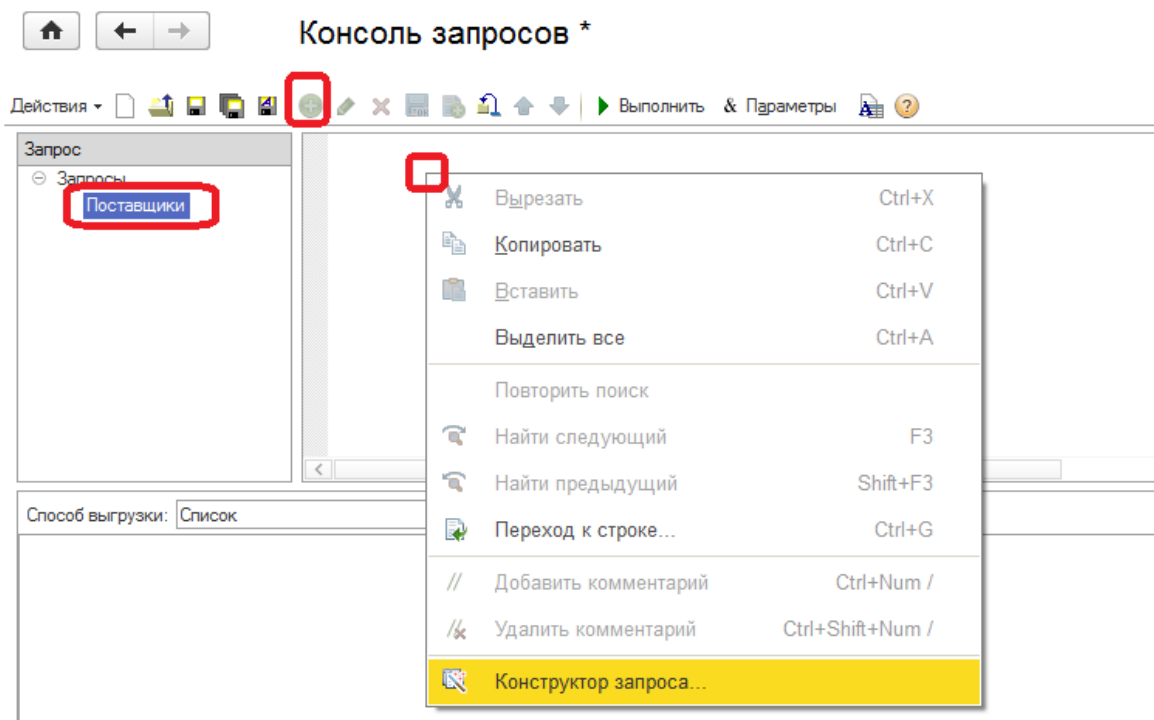


Рис. 12.3.

В конструкторе запроса мы, работая на уровне таблиц, определяем откуда и какая именно информация нас интересует (Рис. 12.4).

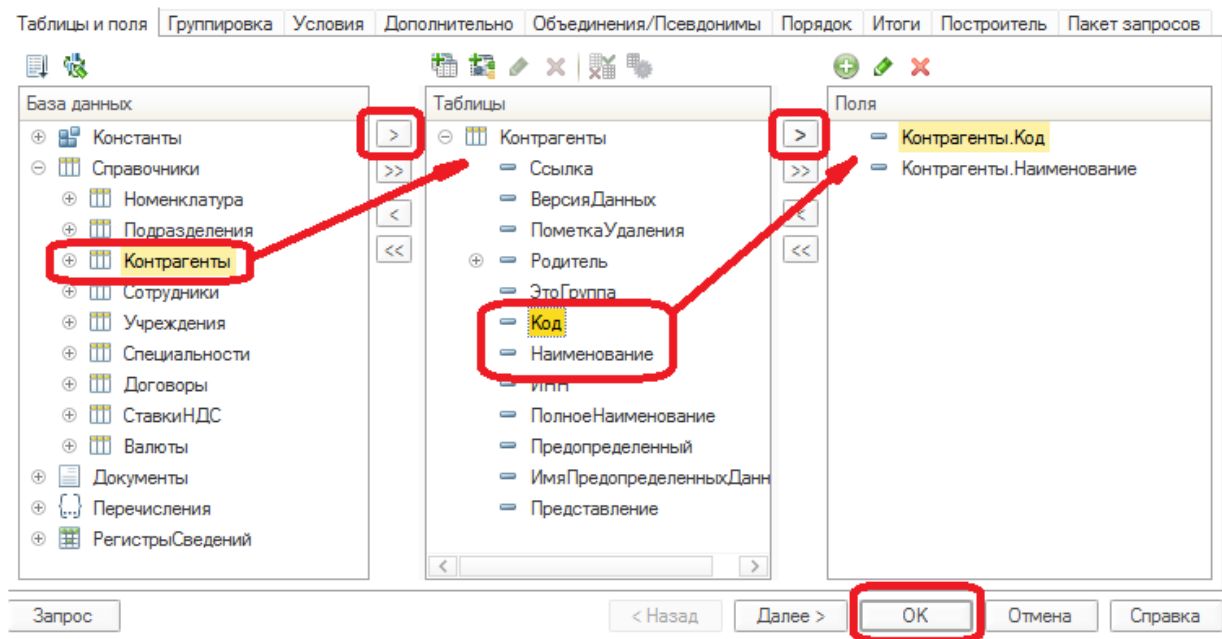


Рис. 12.4.

Результатом работы конструктора будет являться текст запроса, запустив на исполнение который с помощью кнопки "Выполнить", мы сможем увидеть полученную этим запросом информацию (Рис. 12.5)

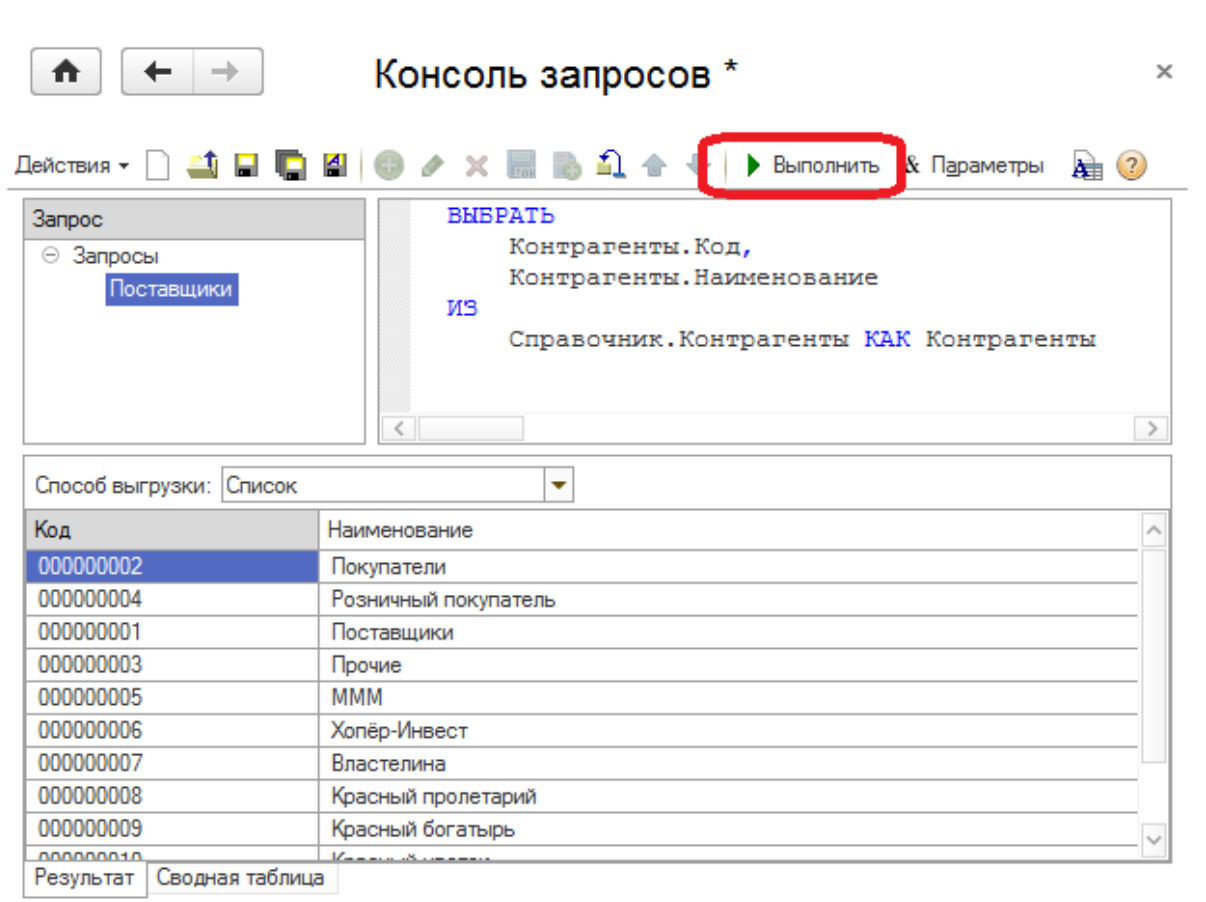


Рис. 12.5.

Результат этого запроса покажет нам не только данные о поставщиках, а вообще всё, что храниться в справочнике. Очевидно, что необходимо избавиться

от лишних данных. Для этого мы вернёмся в конструктор запроса и перейдём на закладку "Условия". Лишними для нас будут являться данные о контрагентах из всех групп, кроме группы "Поставщики" и непосредственно сами группы. Для установки данного отбора выберем поля, по которым будет ставить отбор, и сами функции, которые будут осуществлять отбор (Рис. 12.6).

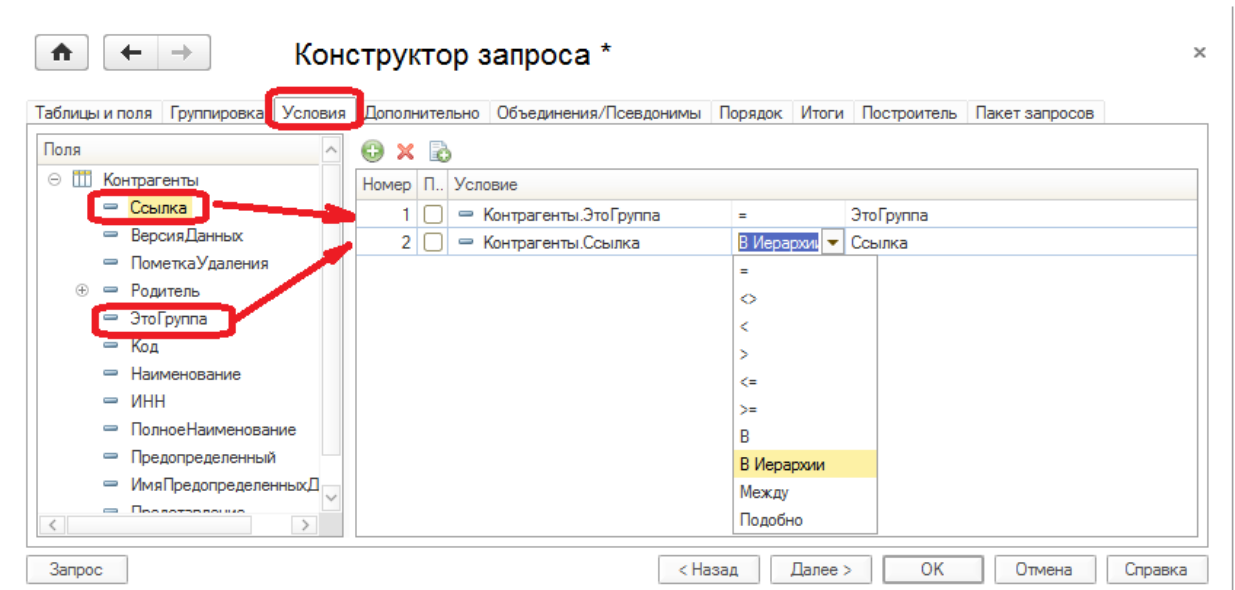


Рис. 12.6.

После добавления условий необходимо установить значения для наших параметров – по какой же собственно группе мы хотим установить отбор? Сделаем это используя кнопку "&Параметры" и в открывшемся окне кнопку "Получить из запроса". Теперь, имея список параметров, которые нам требуется определить, для каждого параметра укажем его значение (Рис. 12.7).

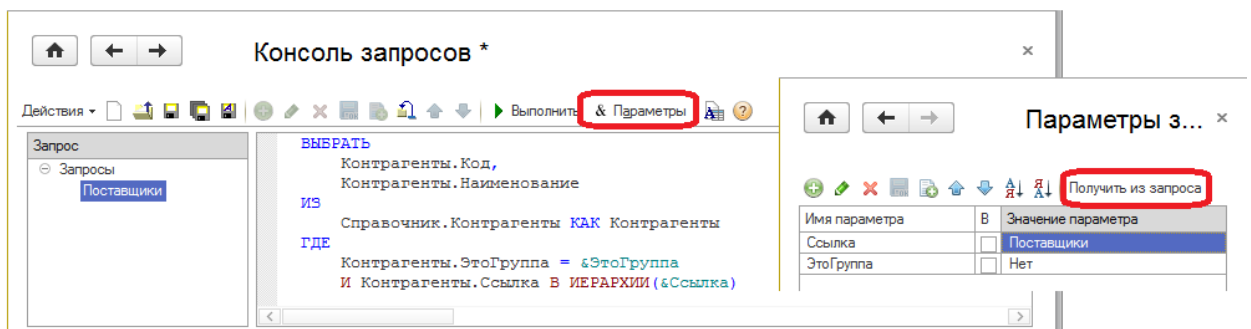
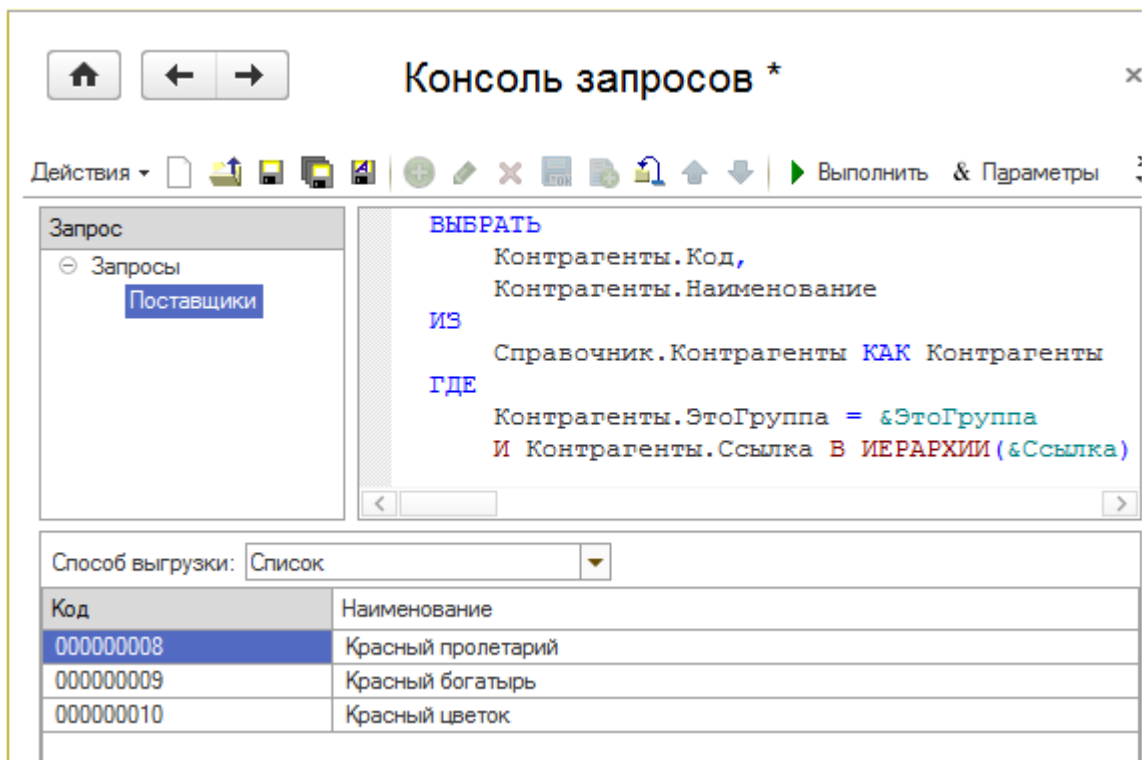


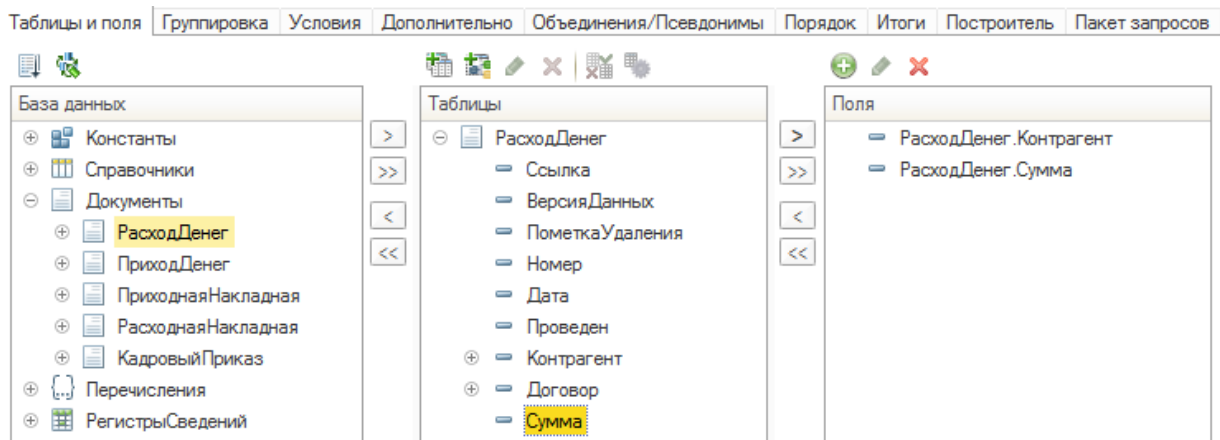
Рис. 12.7.

Установив параметры, выполним наш запрос и увидим искомый список поставщиков (Рис. 12.8). Таким образом, с помощью запроса мы можем извлечь информацию из базы, отфильтровав с помощью различных условий только то, что нам требуется.



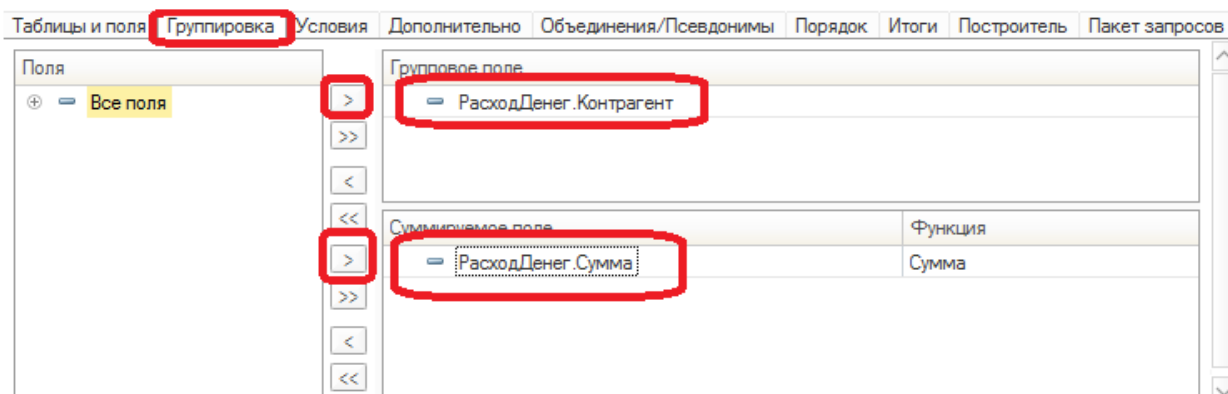
**Рис. 12.8.**

Следующее, что мы сделаем, это с помощью запроса получим данные о том, кому и сколько денег мы заплатили. Получить эту информацию мы можем из документов "РасходДенег" (Рис. 12.9),



**Рис. 12.9.**

Сложность будет заключаться в том, что платежей одному и тому же контрагенту может быть произвольное количество. Очевидно, что удобнее будет увидеть сводные цифры, т.е. по каждому получателю денег должна быть единственная запись, а не столько, сколько было платежей. Для решения этой задачи воспользуемся механизмом группировки (Рис. 12.10).



**Рис. 12.10.**

С помощью этого механизма нам стали доступны не просто исходная информация из базы, но и расчетные данные (Рис. 12.11).

**Запрос**

- Запросы
- Поставщики
- Выплаты**

**ВЫБРАТЬ**

РасходДенег.Контрагент,  
**СУММА** (РасходДенег.Сумма) **КАК** Сумма

**ИЗ**

Документ.РасходДенег **КАК** РасходДенег

**СГРУППИРОВАТЬ ПО**

РасходДенег.Контрагент

Способ выгрузки: Список

Контрагент	Сумма
Красный пролетарий	500 000
Красный богатырь	90 750
Красный цветок	250 000
Бендер О.И.	50 000
Паниковский М.С.	97 863,35
Балаганов А.П.	62 000

**Рис. 12.11.**

Говоря о расчетных данных, вспомним о периодических регистрах сведений. В отличие от справочников или документов для периодических регистров сведений в конструкторе появляются еще две дополнительных таблицы. Эти таблицы получают платформой расчетным образом в момент выполнения запроса и называются виртуальными таблицами. Начальные данные для такого расчета можно передать через параметры виртуальной таблицы. В нашем случае с помощью этого механизма мы можем получить курс валюты на любую произвольную дату (Рис. 12.12).



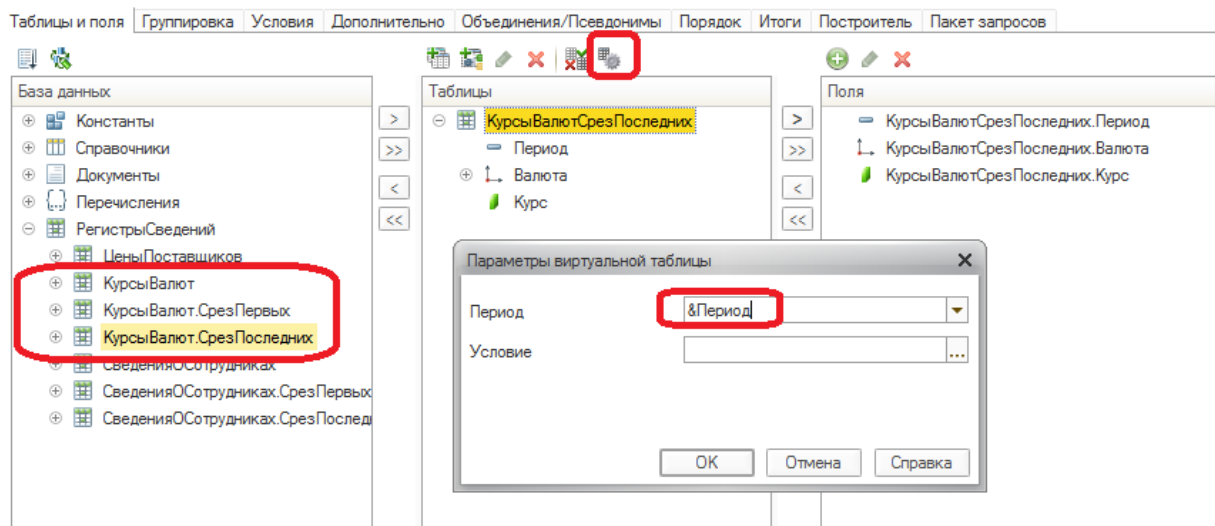


Рис. 12.12.

Результат выполнения данного запроса показывает курсы валют на 17.01.2014, хотя именно на 17 число курсы валют у нас не хранятся (Рис. 12.13).

Запрос: Курсы валют

ВЫБРАТЬ  
КурсыВалютСрезПоследних.Период,  
КурсыВалютСрезПоследних.Валюта,  
КурсыВалютСрезПоследних.Курс  
ИЗ  
РегистрСведений.КурсыВалют.СрезПоследних (&Период, ) КАК КурсыВалютСрезПоследних

Способ выгрузки: Список

Период	Валюта	Курс
10.01.2014	Доллар	33,1547
01.01.2000	Рубль	1,0000
15.01.2014	Евро	45,4139

Рис. 12.13.

Следующий пример будет более сложным. Нас будет интересовать информация о том, в какое подразделение и сколько денег мы выплатили за месяц. Сложность данной задачи заключается в том, что в отличие от предыдущих примеров у нас на первый взгляд отсутствует требуемая информация – в документе "РасходДенег" не указано подразделение. Однако, в документе указан сотрудник, а для сотрудника у нас есть кадровые данные. Таким образом, зная какому сотруднику мы заплатили и зная в каком подразделении он числится, мы можем получить требуемые данные. Для этого в запросе мы будем собирать данные сразу из двух разных источников (Рис. 12.14).

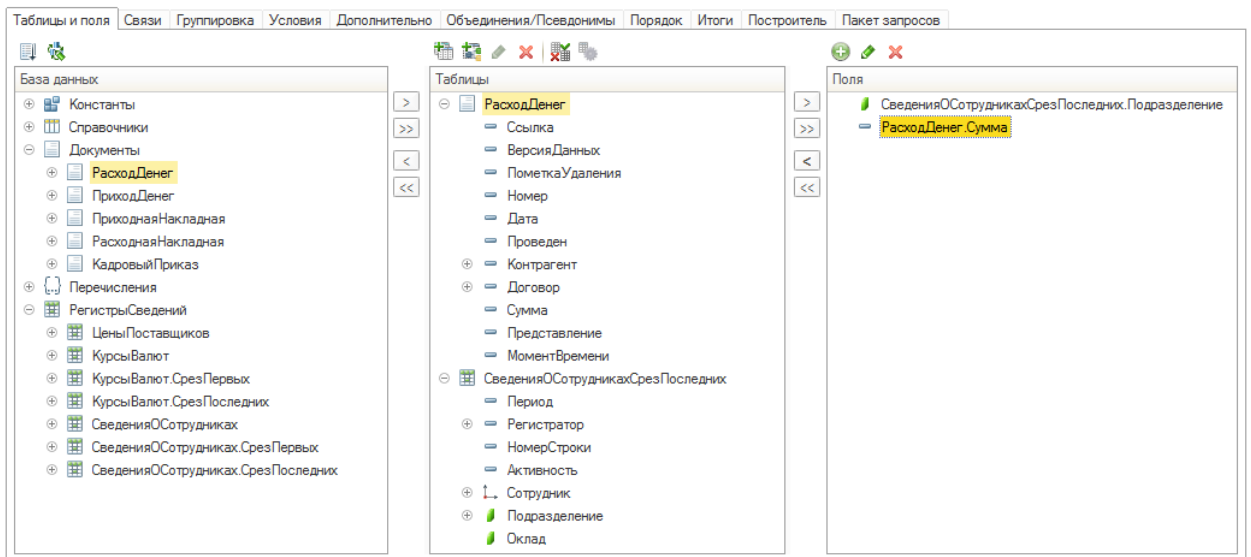


Рис. 12.14.

Сопоставление получателя денег в документе с сотрудником, работающим в подразделении, произведем через механизм соединения на закладке "Связи" (Рис. 12.15).

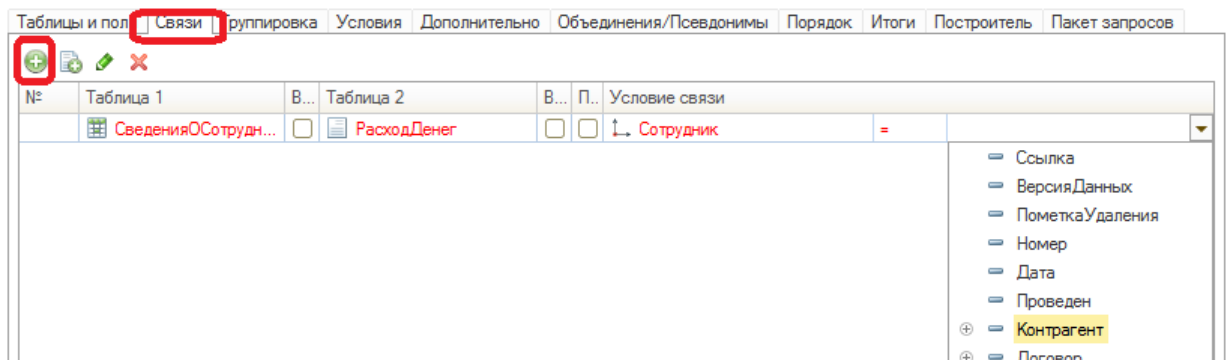


Рис. 12.15.

Применим группировку, чтобы получить общую сумму выплат (Рис. 12.16).

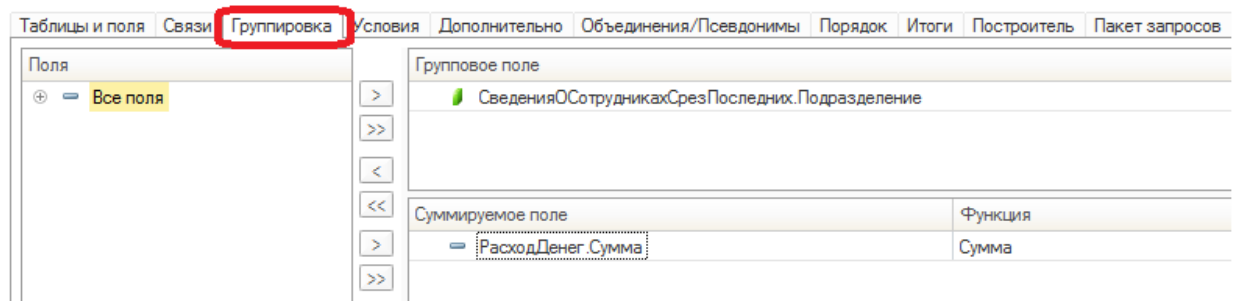


Рис. 12.16.

После чего проверяем результат (Рис. 12.17).

Подразделение	Сумма
Администрация	50 000
Продажи	159 863,35

**Рис. 12.17.**

Таким образом, с помощью запроса нам удалось собрать вместе и обработать информацию, которая хранилась в нашей информационной базе в разных местах. В результате платформа предоставляет нам инструмент для удобного получения и анализа данных. В дальнейшем результат запроса может быть применен для решения самых разных задач. Мы рассмотрим простейшее применение – отображение данных запроса в удобном для пользователя виде, для чего предназначен специальный объект "Отчет".

### **Лабораторная работа №.11 Отчет. Компоновка данных**

Отчет это объект системы "1С:Предприятие", который предназначен для обработки информации и предоставления ее пользователю в виде табличного документа, который может быть распечатан или сохранен как внешний файл. Компоновка данных является механизмом платформы, который помогает создать разработчику отчет.

Создадим стандартным образом новый объект *Отчет* (Рис. 13.1), чтобы пользователь мог удобным для себя способом получить информацию о выплатах по подразделениям. Т.е. предоставим пользователю полученную нами ранее с помощью запроса информацию, но уже в удобном для работы виде. Для формирования отчета воспользуемся системой компоновки данных.

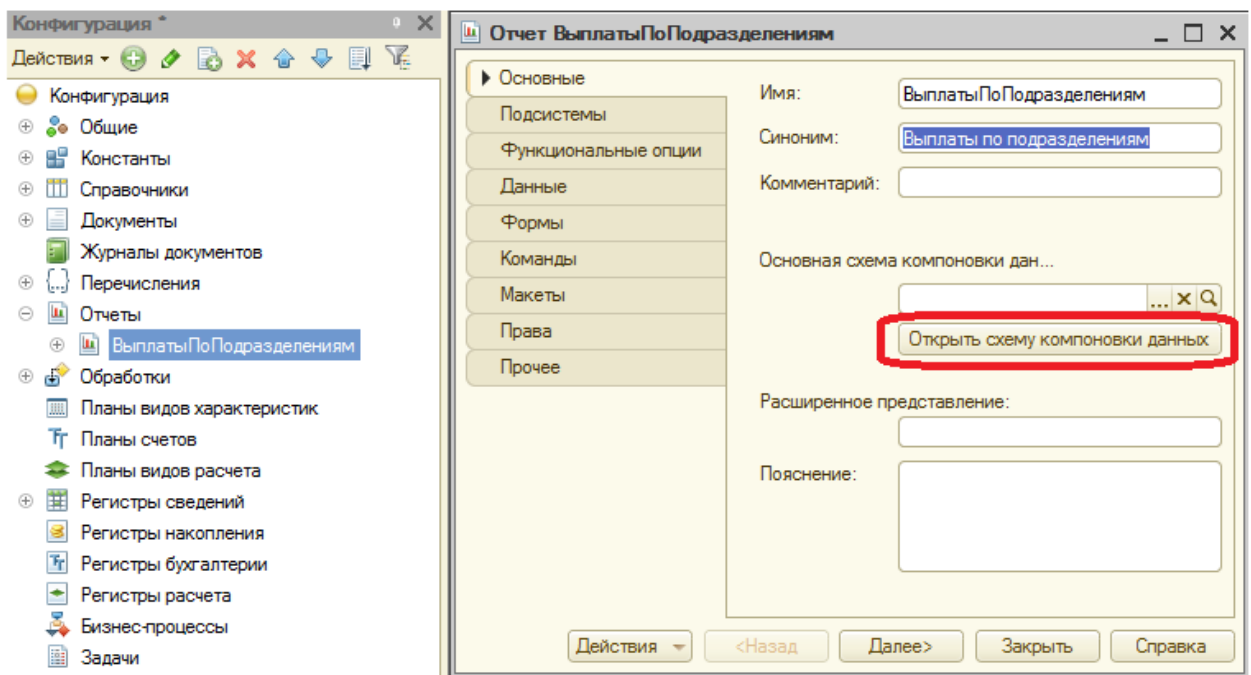


Рис. 13.1.

Создадим с помощью конструктора новую схему компоновки данных (Рис. 13.2)

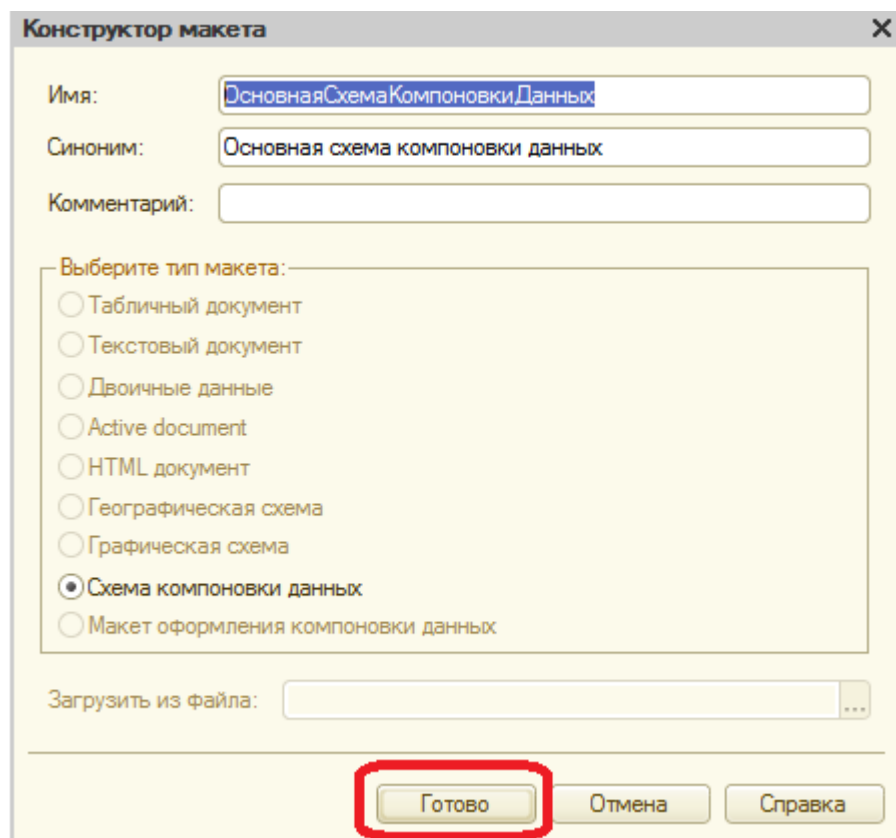


Рис. 13.2.

Добавим новый набор данных, получаемый с помощью запроса (Рис. 13.3)

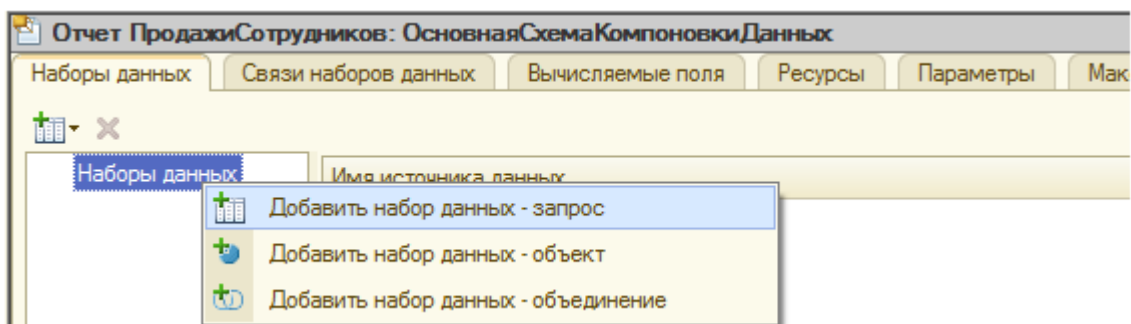


Рис. 13.3.

С помощью конструктора запросов сформируем запрос (Рис. 13.4). После этого с помощью компоновки данных настроим правила отображения этого запроса (Рис.13.5 и Рис. 13.6). Обратите внимание, что добавляя новую группировку, мы ничего не изменяем, а сразу же нажимаем кнопку "ОК" (Рис.13.5).

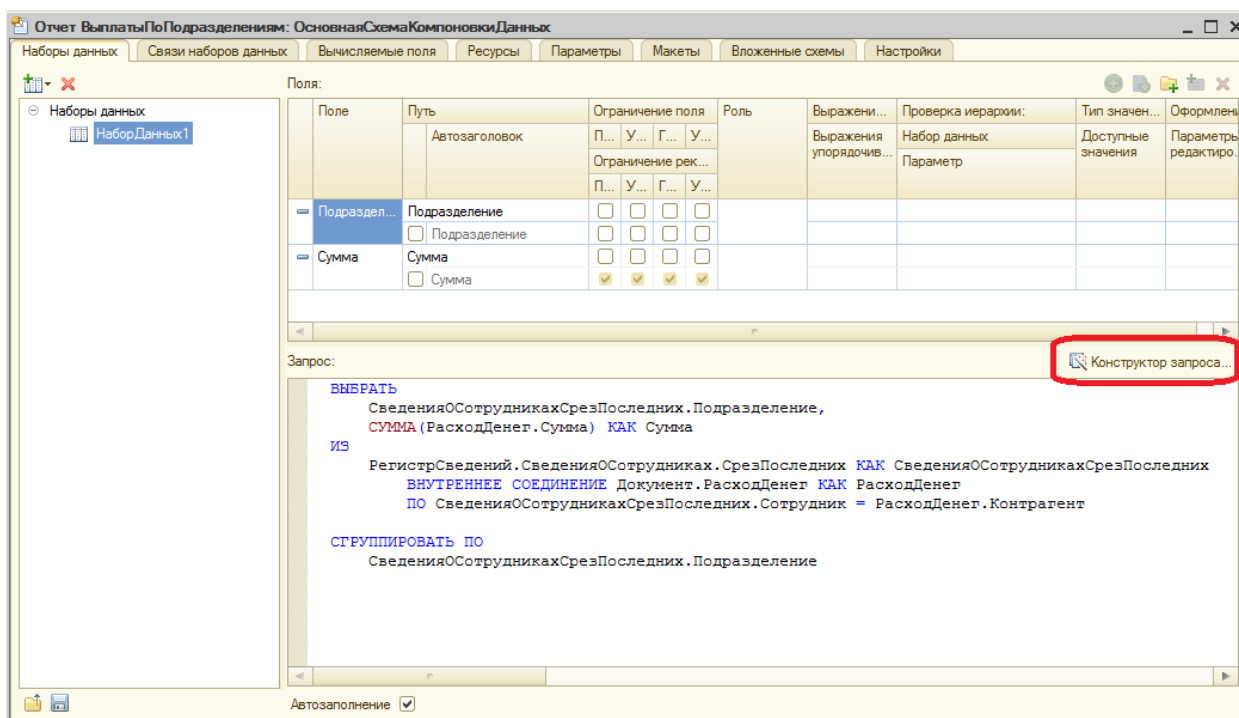


Рис. 13.4.

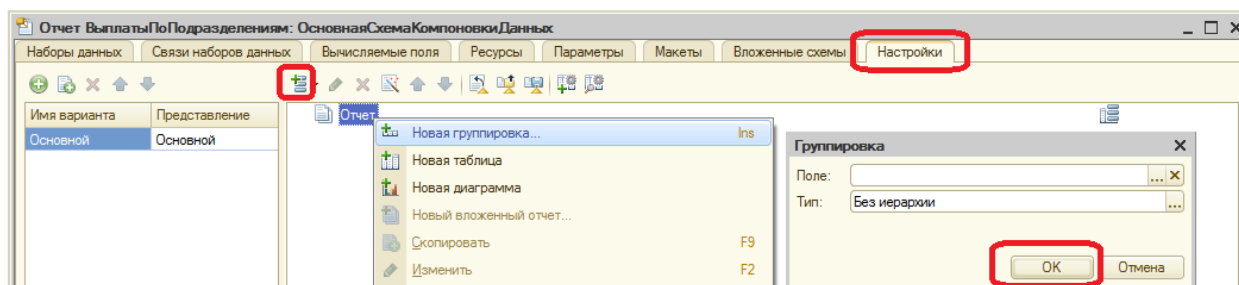
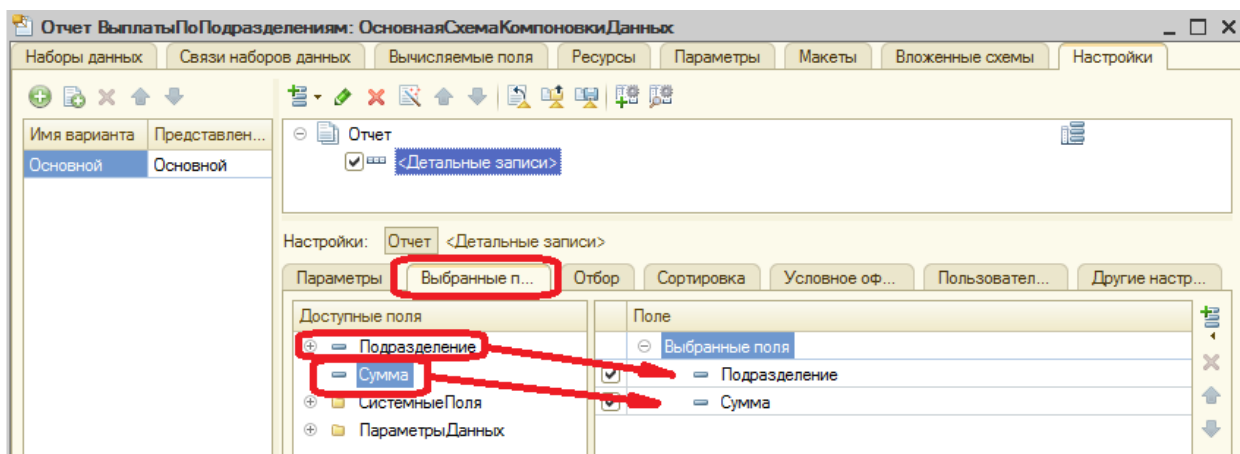
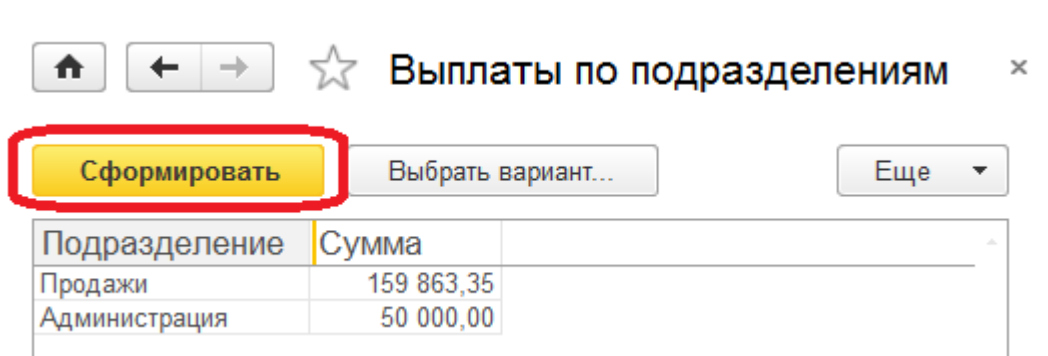


Рис. 13.5.



**Рис. 13.6.**

Не забудем добавить отчет в подсистему, после чего переходим в пользовательский режим, где уже без всяких конструкторов запроса у конечного пользователя появляется возможность получить табличный документ, содержащий требуемую информацию (Рис. 13.7).



**Рис. 13.7.**

Таким образом, сумев создать запрос, получающий нужную информацию, предоставление этой информации конечному пользователю с помощью механизма компоновки данных является достаточно несложной задачей. Однако, механизм компоновки открывает перед нами гораздо больше возможностей, чтобы ознакомиться с ними, мы создадим новый отчет. На этот раз нас будет интересовать информация о продажах – на какую сумму и в какой валюте произвёл продаж каждый сотрудник.

Создадим новый отчет "Продажи Сотрудников". Описанным выше способом воспользуемся механизмом компоновки и запросом соберём данные о продажах (Рис. 13.8).

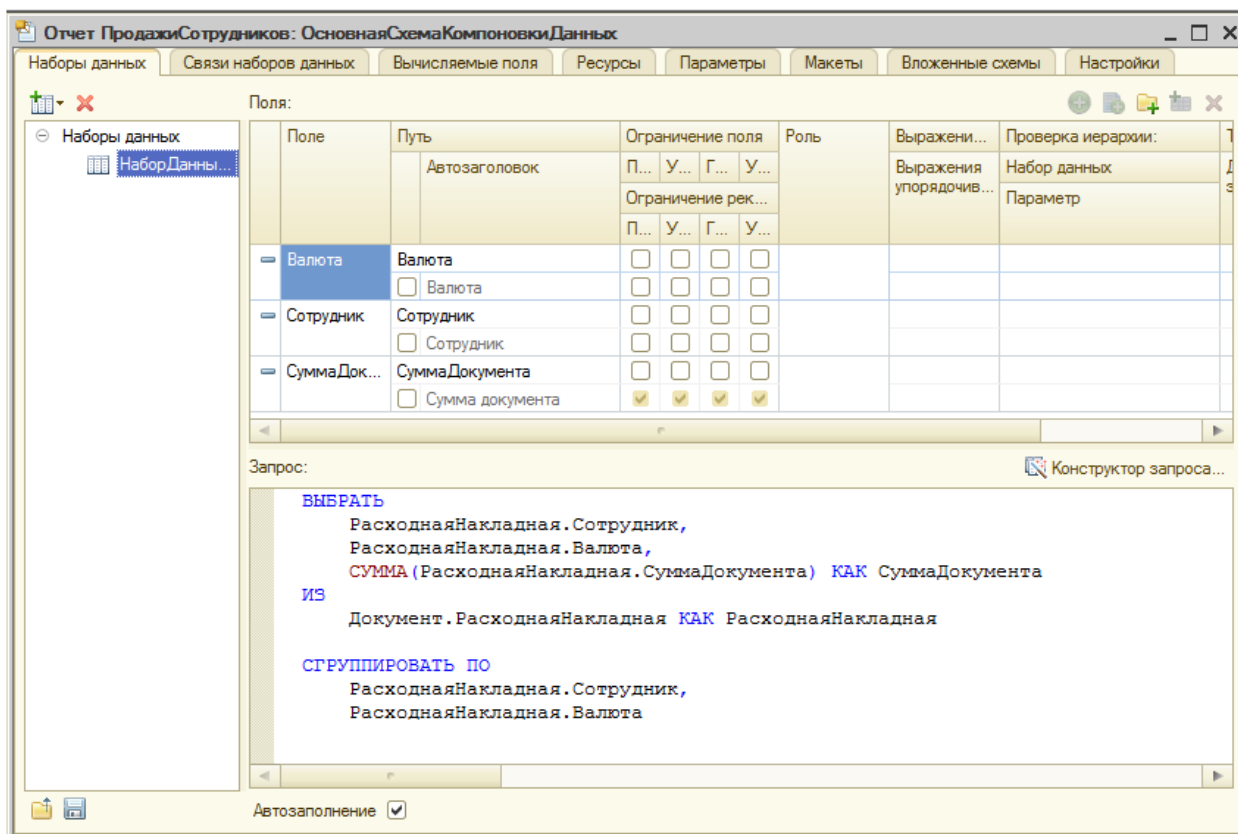


Рис. 13.8.

На закладке "Ресурсы" определим поле "СуммаДокумента" как ресурс (Рис. 13.9). После этого перейдем на закладку "Настройки", где определим группировки и выбранные поля (Рис. 13.10). В отличие от предыдущего отчета, здесь нам необходимо будет явным образом указать значения группировок.

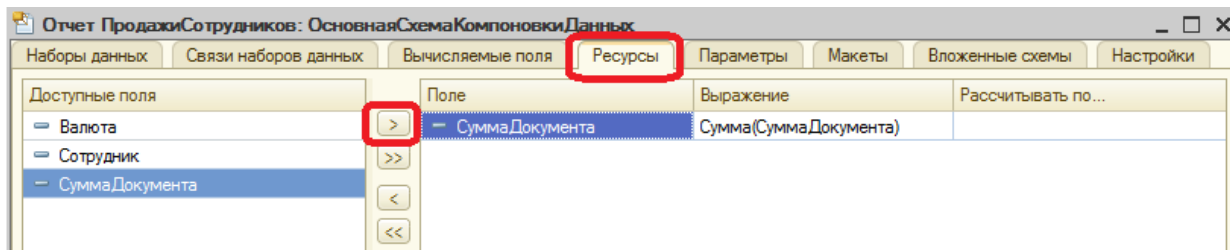
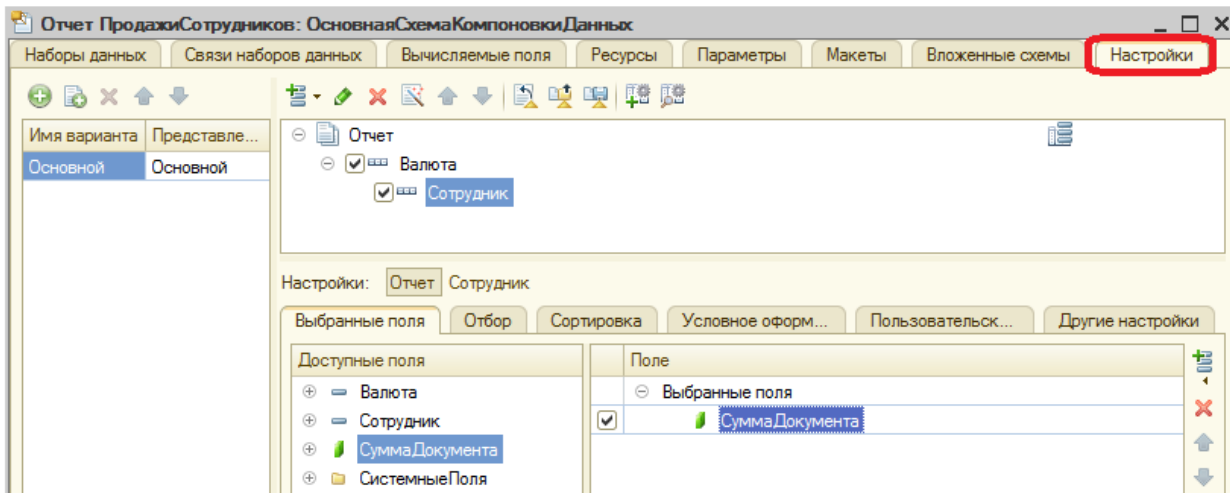
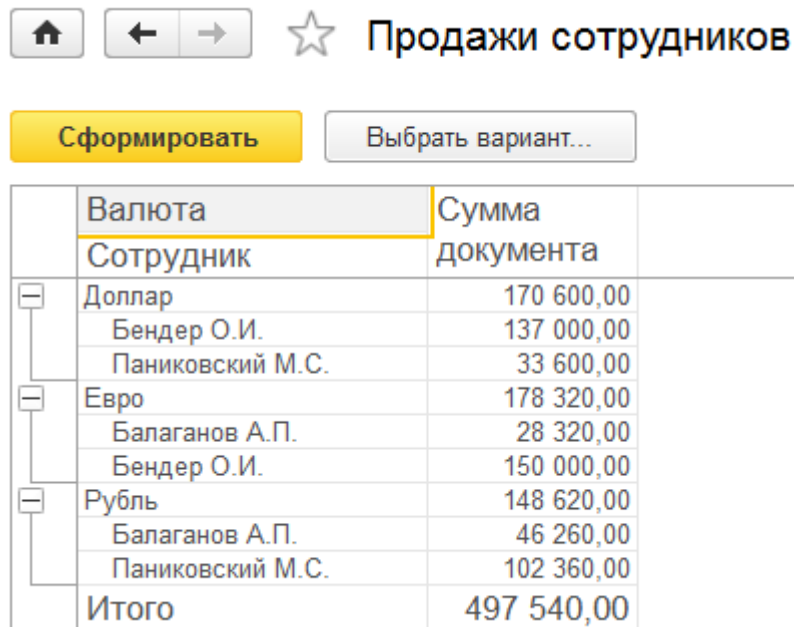


Рис. 13.9.



**Рис. 13.10.**

Результат наших действий позволит получить пользователю сводные данные о продажах сотрудников в разрезе валют (Рис. 13.11). Естественно, что для проверки правильности работы нашего отчета перед его формированием надо внести сами данные о продажах. Чем больше и разнообразнее данных будет содержать база, тем более наглядным будет наш отчет.



	Валюта	Сумма
	Сотрудник	документа
[-]	Доллар	170 600,00
	Бендер О.И.	137 000,00
	Паниковский М.С.	33 600,00
[-]	Евро	178 320,00
	Балаганов А.П.	28 320,00
	Бендер О.И.	150 000,00
[-]	Рубль	148 620,00
	Балаганов А.П.	46 260,00
	Паниковский М.С.	102 360,00
	Итого	497 540,00

**Рис. 13.11.**

С одной стороны, мы решили поставленную задачу и предоставили пользователю нужные данные в красивом и удобном виде. Однако, с другой стороны, как быть, если теперь пользователя заинтересовала та же самая информация, но в другом разрезе? Если теперь нас в первую очередь интересует не валюта, а сотрудник? Получается, что все необходимые данные в отчете присутствуют, но пользоваться ими неудобно. Конечно, можно сделать еще один отчет для решения этой новой задачи, но вместо этого механизм компоновки данных предлагает нам новый механизм – варианты отчета.

Вернемся в режим "Конфигуратор" в настройки компоновки нашего отчета. Все необходимые данные уже получены нами с помощью запроса, а вот их отображение определяется нами на закладке "Настройки", где в левом окне мы добавляем новый вариант, для которого можем совершенно новым образом определить группировки и выбранные поля (Рис. 13.12).



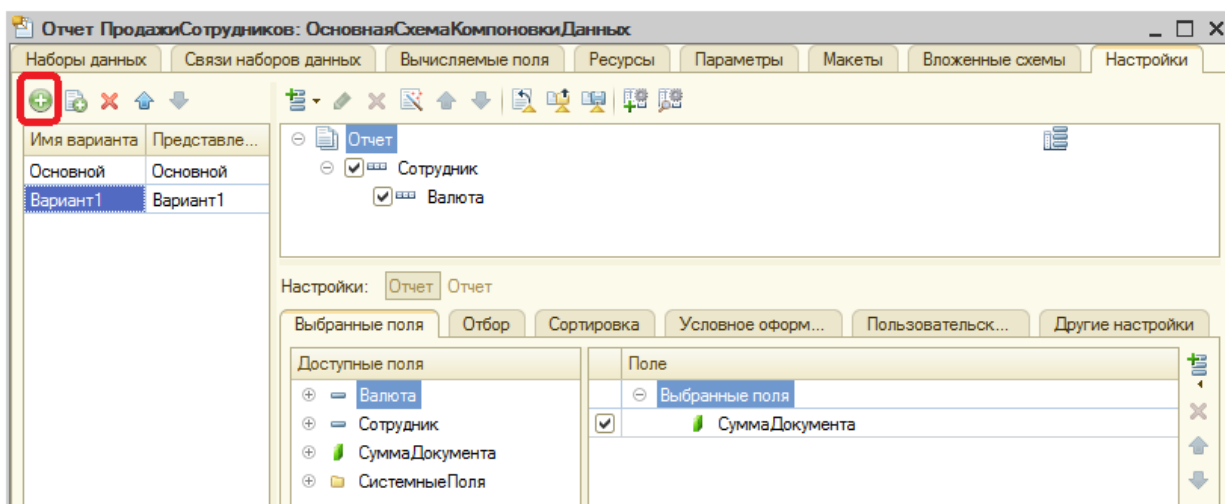


Рис. 13.12.

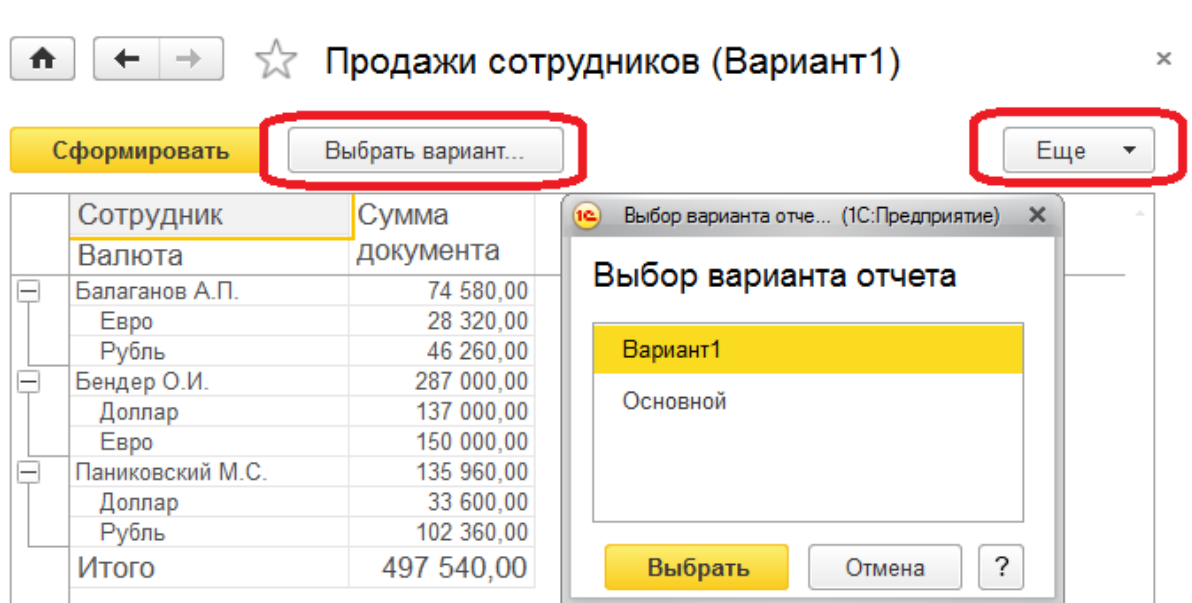


Рис. 13.13.

Теперь в пользовательском режиме для нашего отчета появляется новая возможность выбора варианта (Рис. 13.13) и, таким образом, отображать одни и те же данные разными способами в зависимости от своих текущих предпочтений.

Кроме того, по кнопке "Еще" у пользователя появляется крайне важная возможность добавить свои собственные варианты отчета (Рис. 13.14).

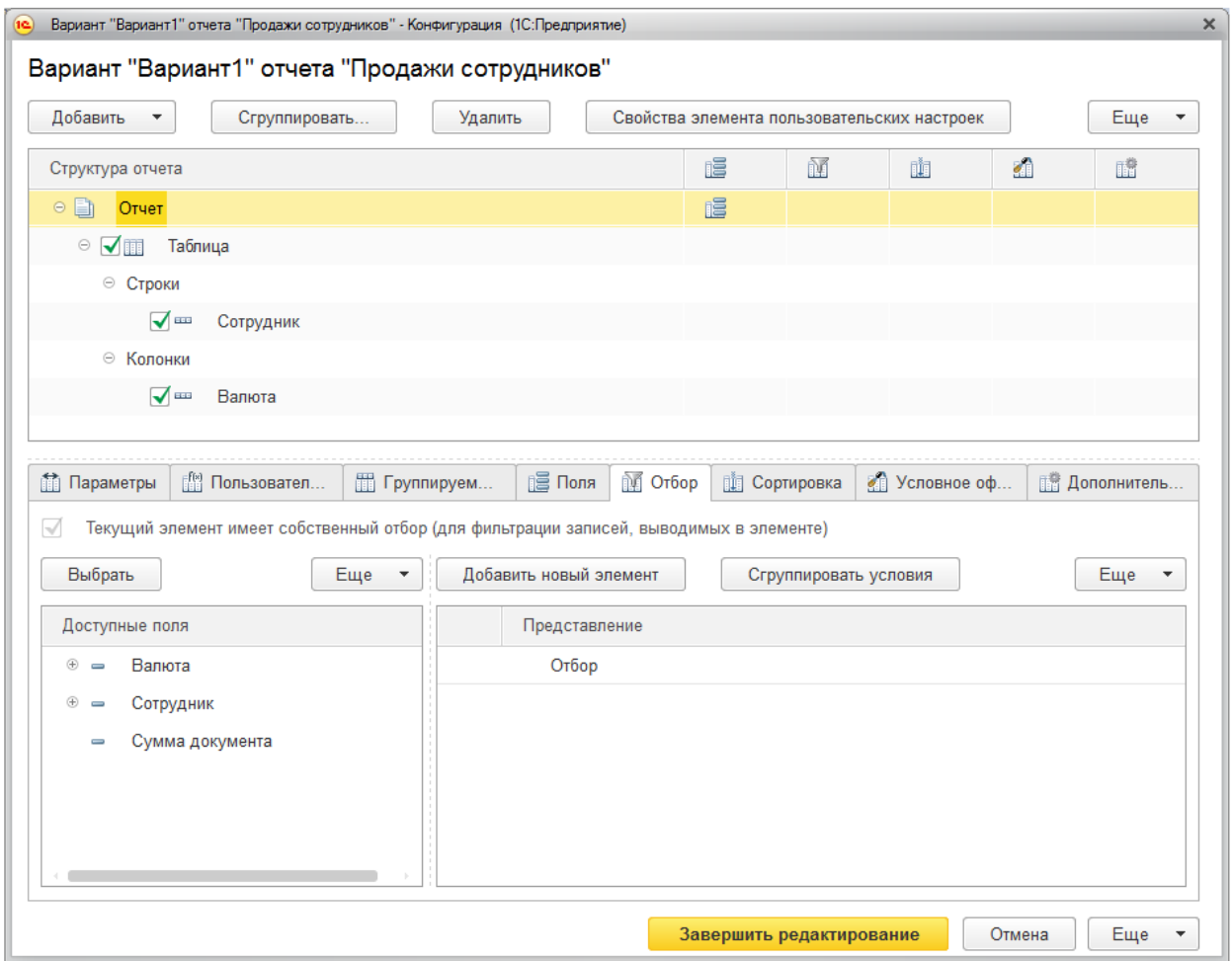


Рис. 13.14.

Эти варианты могут быть позднее могут быть сохранены для дальнейшего использования (Рис. 13.15). Обратите внимание, что при настройке варианта у пользователя появляется еще множество дополнительных возможностей по установке различных отборов, сортировок и прочих вещей, которые и определяют окончательный внешний вид нашего отчета. В частности, используя настройки, приведенные на Рис. 13.14, можно отобразить те же самые данные о продажах в виде кросс-таблицы, что может иногда быть очень удобно.

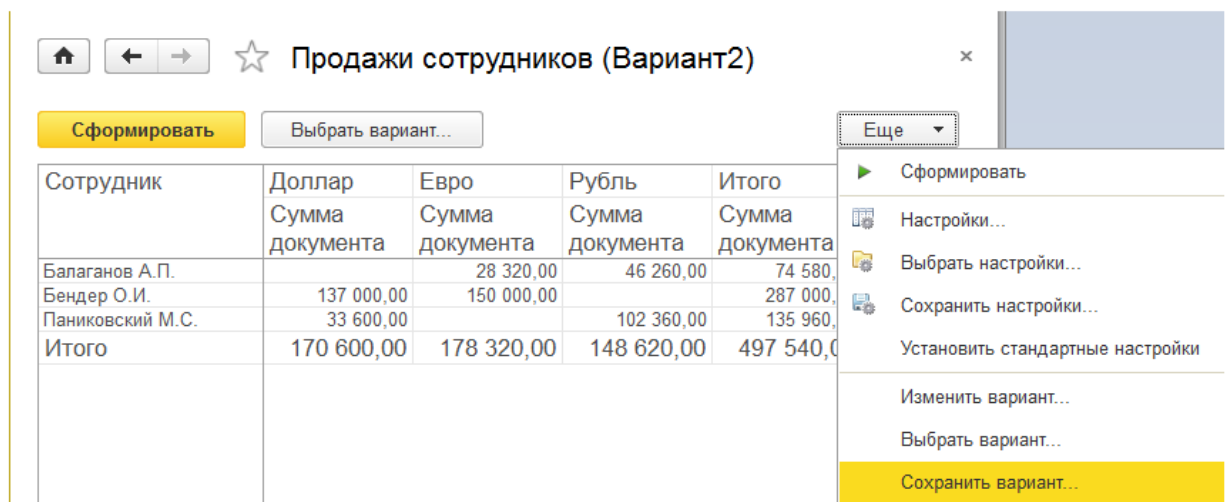


Рис. 13.15.

Однако и это еще не все возможности, которые предоставляет нам система компоновки данных. Создадим основную форму списка для справочника "Валюты". В свойствах основного реквизита формы "Список" установим флаг "ПроизвольныйЗапрос" и нажмём на гиперссылку "Открыть" (Рис. 13.16).

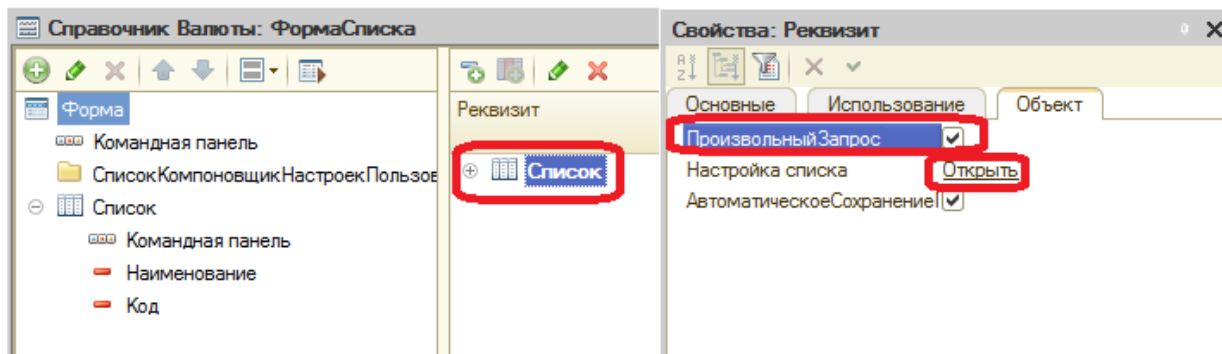


Рис. 13.16.

Для отображения данных справочника платформа получает данные с помощью того же самого механизма запросов, а отражает его результат с помощью компоновки данных. В результате мы, как разработчик, получаем возможность изменить текст запроса для отображения данных на форме каким-то иным образом. Например, при работе с валютой крайне удобно видеть её текущий курс, который храниться не в справочнике, а в регистре сведений. Добавив в запрос соединение (Рис. 13.7) мы можем решить эту задачу.

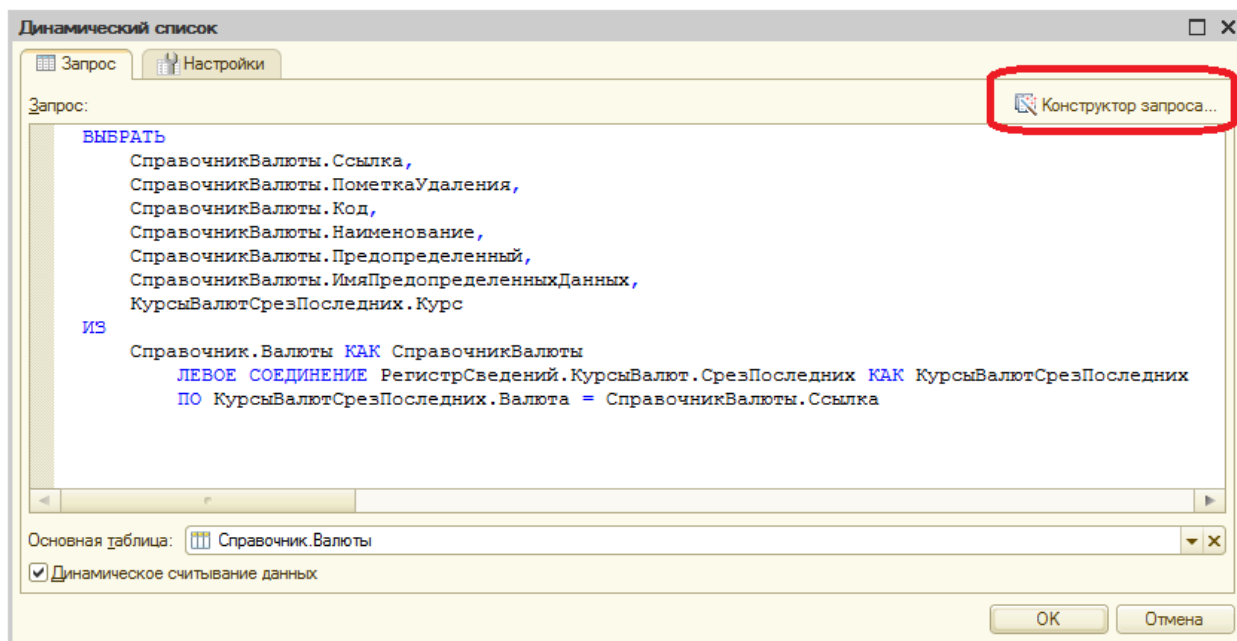


Рис. 13.17.

Получив данные запросом, изменяем саму форму, добавив туда новые данные (Рис. 13.18)

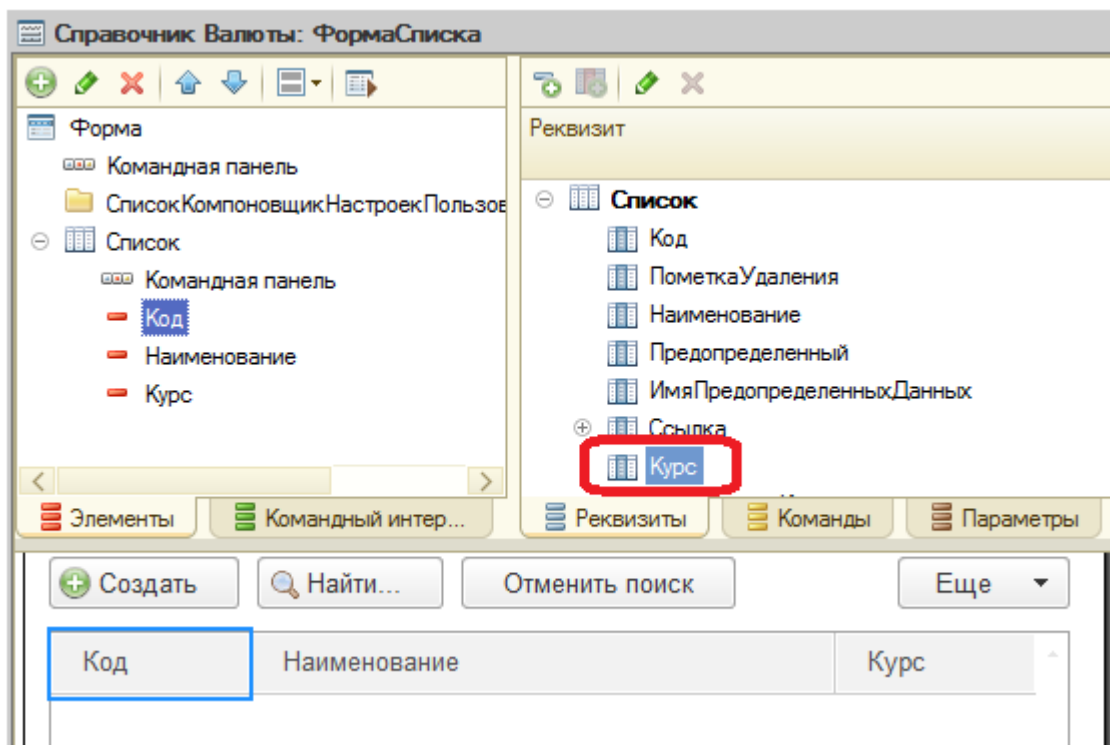


Рис. 13.18.

Осталось проверить результат в пользовательском режиме (рис. 13.19)

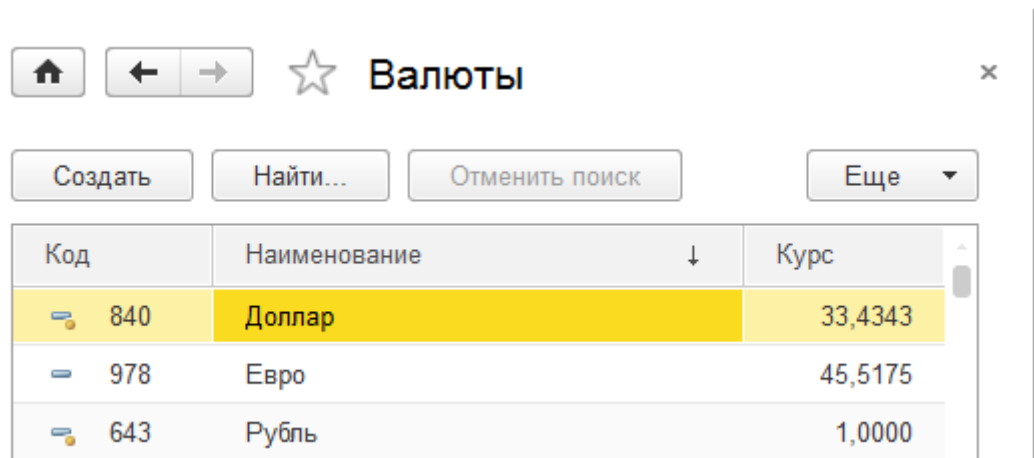


Рис. 13.19.

## Библиографический список

1. Создание знания и информационной инфраструктуры субъектов предпринимательства [Электронный ресурс]/ А.Н. Асаул [и др.].— Электрон. текстовые данные.— СПб.: Институт проблем экономического возрождения, 2010.— 196 с.— Режим доступа: <http://www.iprbookshop.ru/18213.html>.— ЭБС «IPRbooks»

2. Трутнев Д.Р. Инфраструктура доверия в государственных информационных системах [Электронный ресурс]: учебное пособие/ Трутнев Д.Р.— Электрон. текстовые данные.— СПб.: Университет ИТМО, 2012.— 94 с.— Режим доступа: <http://www.iprbookshop.ru/68655.html>.— ЭБС «IPRbooks»

3. Савельев А.О. Решения Microsoft для виртуализации ИТ-инфраструктуры предприятий [Электронный ресурс]/ Савельев А.О.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 284 с.— Режим доступа: <http://www.iprbookshop.ru/52175.html>.— ЭБС «IPRbooks»

4. Олейник А.И., ИТ-инфраструктура [Электронный ресурс]: учеб. метод. пособие / Олейник А.И., Сизов А.В. - М. : ИД Высшей школы экономики, 2012. - 134 с. - ISBN 978-5-7598-0958-6 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785759809586.html>