**ЗНАКОМСТВО С ПЛАТФОРМОЙ 8.3**

Цель курса - дать слушателям начальное представление о работе с основными объектами и механизмами платформы, сформировать у них практические навыки по работе в системе на примере решения так называемого "сквозного" примера. При этом основной упор в процессе занятий делается на самостоятельную работу слушателей.

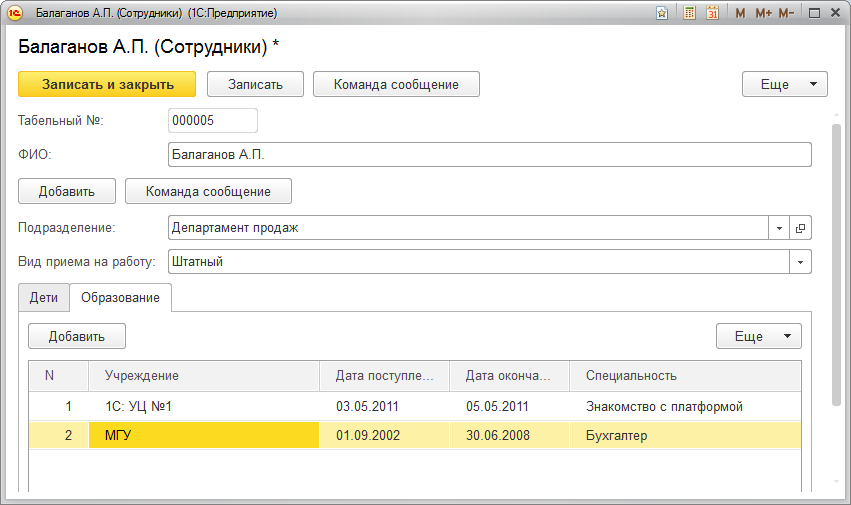
**1. Общие положения**

**1.1. Задача**

Необходимо автоматизировать торговую организацию.

Структурно компания состоит из администрации и отдела продаж, который в свою очередь разделен на оптовый и розничный отдел.

Для ведения кадрового учета необходимо хранить список сотрудников организации. Для каждого сотрудника необходимо иметь возможность указать сведения о его квалификации (где и когда он обучался), а так же сведения о его детях (Рис. 2.1). Назначение сотруднику оклада или перевод его в какое-либо подразделение отражаются документом "Кадровый приказ".

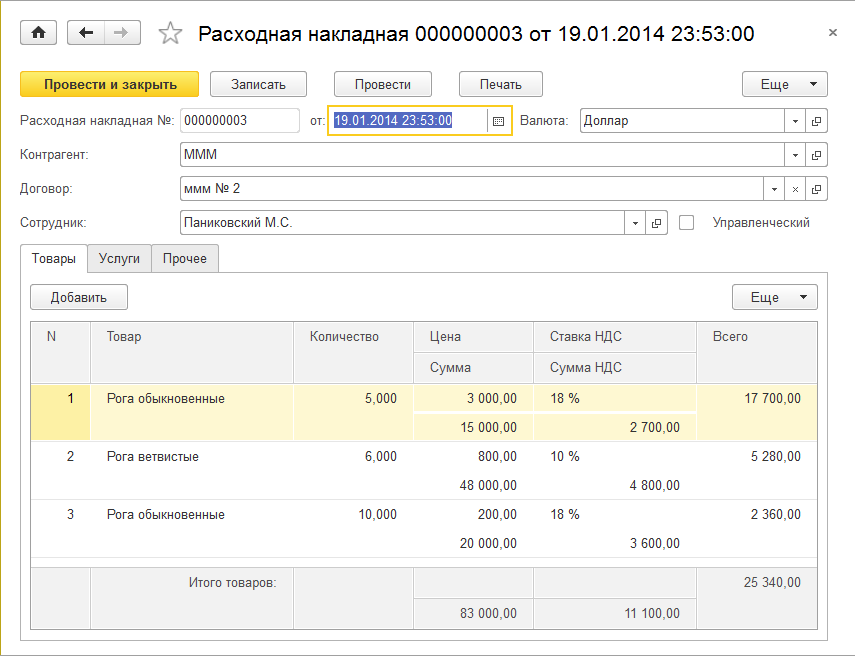


**Рис. 2.1.**

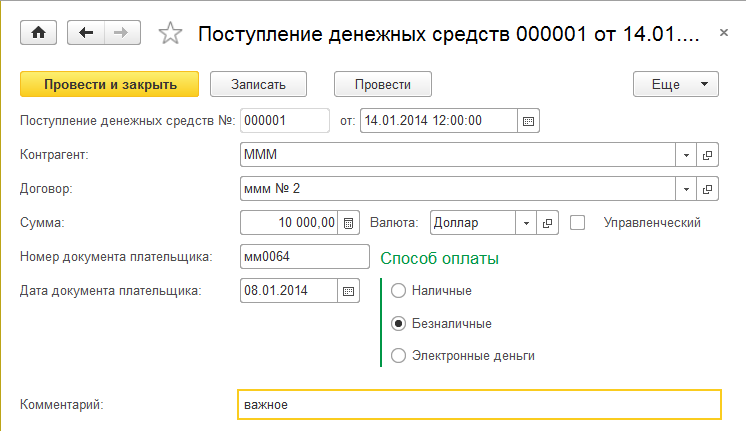
Основная деятельность компании заключается в закупке и дальнейшей продаже товаров и оказания сопутствующих услуг. Необходимо вести перечень используемых товаров и услуг; поставщиков и покупателей. Все взаиморасчеты с контрагентами ведутся в разрезе договоров.

Оплата товара поставщику отражается документом "Списание денежных средств", поступление товаров документом "Приходная накладная". Следует предусмотреть возможность выплаты денежных средств не только поставщику, но и сотруднику компании.

Отгрузка товара покупателю отражается документом "Расходная накладная" (Рис. 2.2), а его оплата документом "Поступление денежных средств" (Рис.2.3). Для документа "Расходная накладная" необходимо иметь возможность сформировать печатную форму.



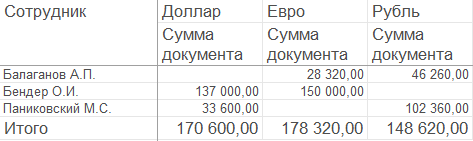
**Рис. 2.2.**



**Рис. 2.3.**

При продаже товаров в компании может возникнуть необходимость вести валютный учет. Данная возможность должна являться дополнительной, то есть подключаться или отключаться при необходимости.

Для анализа деятельности компании необходимо иметь возможность узнать в разрезе сотрудников в текущем месяце, на какую сумму было произведено продаж (Рис. 2.4).

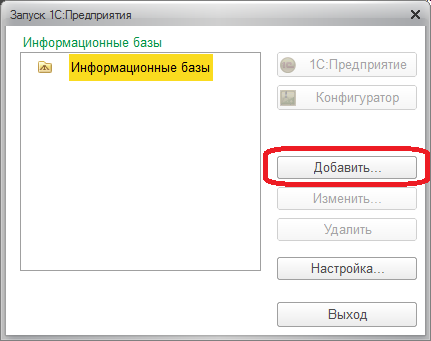


**Рис. 2.4.**

**1.2. Создание новой информационной базы**

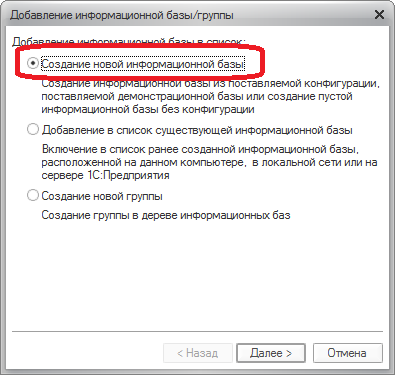
Система программ "1С:Предприятие 8.2" включает в себя платформу и прикладные решения, разработанные на ее основе, для автоматизации деятельности организаций и частных лиц. Сама платформа не является программным продуктом для использования конечными пользователями, которые работают с прикладным решением (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.

Первое, что нам необходимо сделать для решения предлагаемой учебной задачи, это создать новую информационную базу. Для этого на диске "С:\" создадим каталог "\Знакомство с платформой", после чего запускаем систему "1С:Предприятие". На экран будет выведен диалог "Запуск 1С:Предприятия" (Рис.2.5).



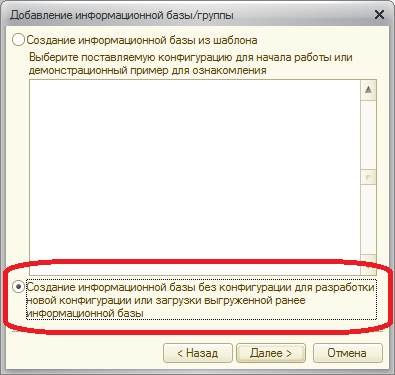
**Рис. 2.5.**

Создадим новую информационную базу, нажав кнопку "Добавить". Далее следуем инструкциям, появляющимся на экране. Поскольку создаем новую базу, в первом окне оставляем верхний вариант (Рис. 2.6).



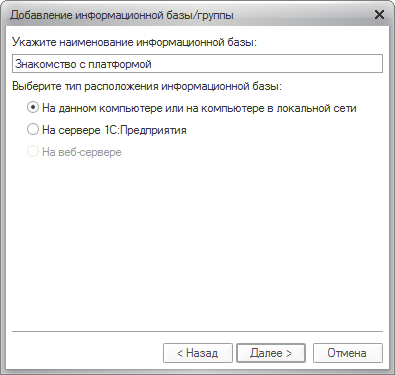
**Рис. 2.6.**

Далее необходимо указать, что создается новая пустая база, а не типовая из шаблона, для чего выбираем нижний вариант (Рис 2.7).



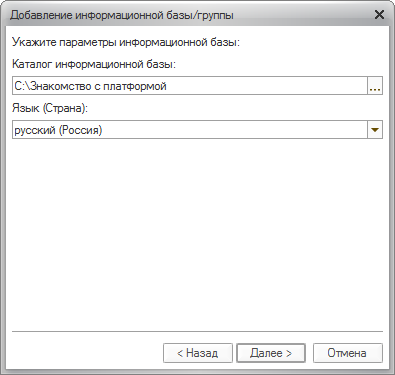
**Рис. 2.7.**

Следующее, что необходимо сделать, это дать имя создаваемой базе. В нашем случае назовем базу "Знакомство с платформой" (Рис. 2.8).



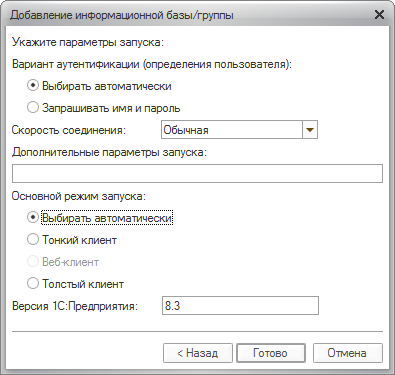
**Рис. 2.8.**

После этого необходимо указать, где будут храниться данные, содержащиеся в нашей информационной базе. Для этого ранее нами была создана специальная папка, которую теперь необходимо указать в настройках (Рис. 2.9).



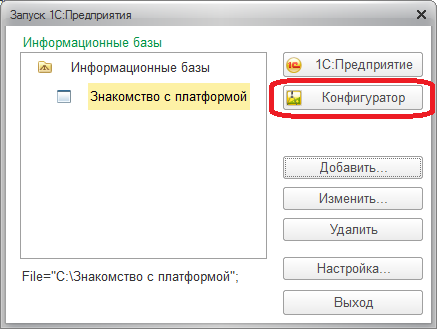
**Рис. 2.9.**

В следующем окне оставляем параметры запуска по умолчанию (Рис. 2.10) и на этом создание новой информационной базы будет завершено.



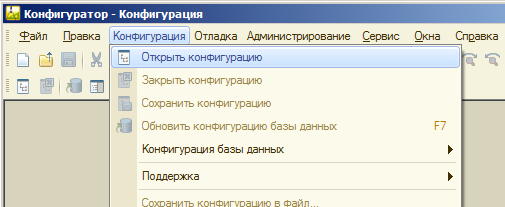
**Рис 2.10.**

Теперь после запуска "1С:Предприятия" у нас появится возможность работать с новой информационной базой (Рис. 2.11), причем в обоих режимах, выбор которых будет осуществляться нажатием на соответствующие кнопки "1С:Предприятие" или "Конфигуратор".



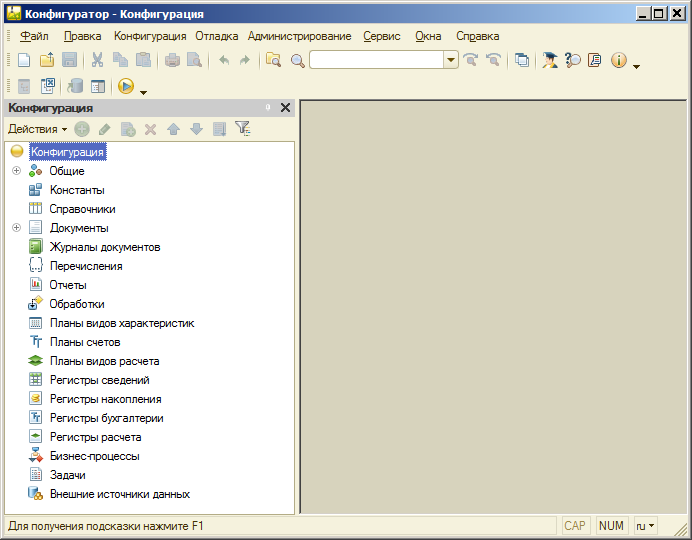
**Рис. 2.11.**

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* (Рис. 2.12).



**Рис. 2.12.**

Открывается новое окно (Рис. 2.13):



**Рис. 2.13.**

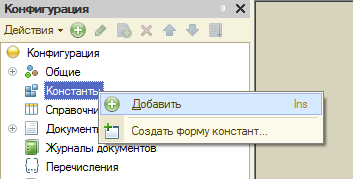
Данное окно называется окном конфигурации, а содержащаяся в нем информация – деревом конфигурации.

**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.

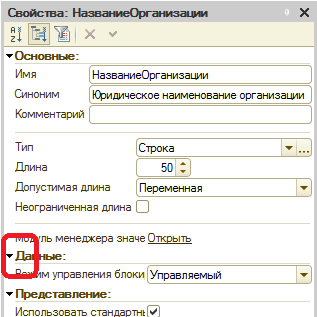
При оформлении документов нам понадобится информация о генеральном директоре нашей фирмы, ее наименовании и дате регистрации. Данная информация не будет меняться, поэтому для ее хранения можно использовать константу. Поскольку константа предназначена для хранения одного значения, потребуется создать три константы.

Для создания новой константы в окне дерева объектов конфигурации правой кнопкой мышки на ветке *Константы* вызываем контекстное меню, в котором выбираем пункт *Добавить*. (Рис 3.1)



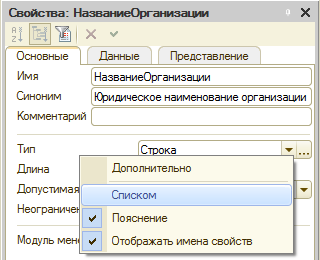
**Рис. 3.1.**

После того, как мы выберем данный пункт, на экране откроется новое окно, называемое Палитра свойств, в которой будут отражены свойства константы. (Рис 3.2). Свойства можно сворачивать по группам, для удобства работы.



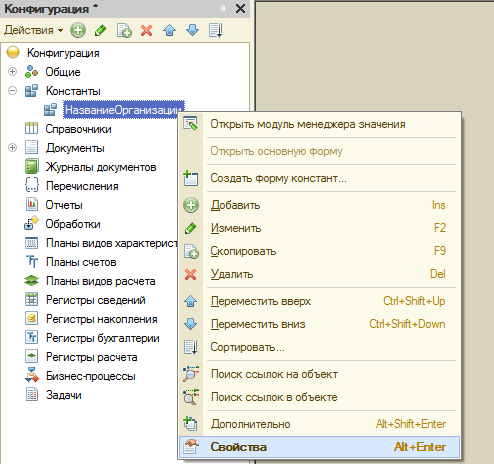
**Рис. 3.2.**

Внешний вид палитры свойств можно изменить через контекстное меню по нажатию на правую кнопку мыши (Рис. 3.3).



**Рис. 3.3.**

В дальнейшем можно вызвать это окно двойным щелчком мыши на соответствующей ветке в окне дерева объектов конфигурации или через пункт *Свойства* контекстного меню, вызванного правой кнопкой мышки. (Рис 3.4)



**Рис. 3.4.**

Нам потребуется заполнить следующие поля:

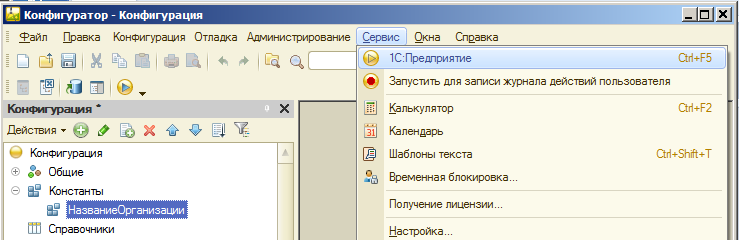
         Имя – идентификатор константы. Поскольку в системе может существовать произвольное количество констант, то необходимо однозначно идентифицировать каждую из них.

         Синоним – удобное для пользователя название.

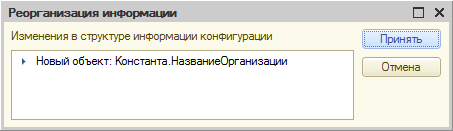
         Тип данных – тип данных хранимых в константе. Для строкового типа данных необходимо указать предельную длину строки, а также способ ее хранения. В системе предусмотрена возможность хранить строки неограниченной длины.

          Допустимая длина отражает системный способ реализации хранения строк. Для хранения строки в системе отводится определенное место, соответственно, мы можем целиком его занимать, дополняя строку пробелами, или занимать только необходимое для хранения нашей строки место. В нашем случае будем использовать строку длиной 50 с переменной допустимой длиной. В том случае, когда длина строки постоянная, то всегда храниться указанное значение символов, дополненное пробелами, если сама хранимая строка короче.

Таким образом, мы создали структуру данных, которая позволит хранить необходимую нам информацию, но не определили, как именно называется наша организация. Для того чтобы задать непосредственно нужное значение, необходимо перейти в пользовательский режим (режим "1С:Предприятие"), причем пользователь должен получить возможность использовать новый объект, который раньше отсутствовал. Для этого выбираем пункт меню *Сервис - 1С:Предприятие* (Рис. 3.5) и последовательно подтверждаем необходимость обновления информационной базы (Рис. 3.6 - Рис. 3.8).



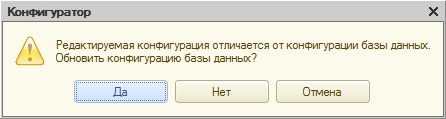
**Рис. 3.5.**



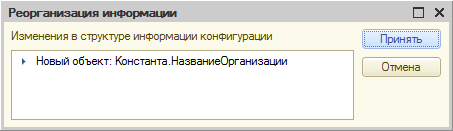
**Рис. 3.6.**

Только после выполнения всех этих действий, в пользовательском режиме можно ввести необходимую информацию (Рис. 3.6 и Рис. 3.7).

*1С:Предприятие* и последовательно подтверждаем необходимость обновления информационной базы (Рис. 3.4 и Рис. 3.5).

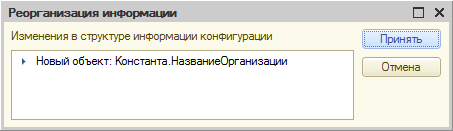


**Рис. 3.7.**

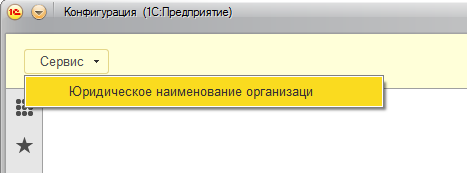


**Рис. 3.8.**

Только после выполнения всех этих действий, в пользовательском режиме можно ввести необходимую информацию (Рис. 3.9 и Рис. 3.10).



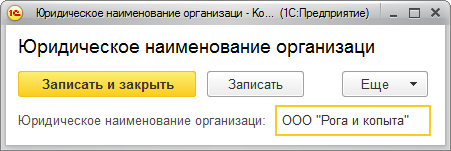
**Рис. 3.9.**



**Рис. 3.10.**

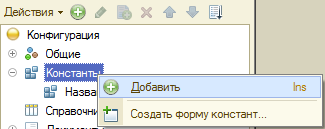
В дальнейшем, если нет необходимости вносить изменения в структуру данных, запуск информационной базы производится стандартным образом (Рис. 2.11).

Приступим к созданию следующей константы для учета ИНН нашей фирмы. 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* (Рис 3.8).



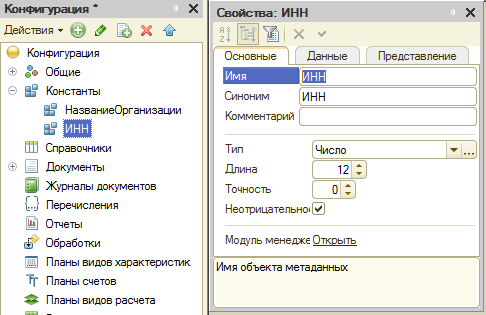
**Рис. 3.8.**

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. 3.9).



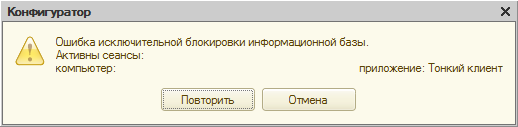
**Рис. 3.9.**

Запустим "1С:Предприятие"  в пользовательском режиме и укажем ИНН компании. Так как мы изменили структуру хранения данных в нашей базе (добавили новый объект), необходимо произвести реструктуризацию информационной базы. Процесс этот сложный и очевидно, что в момент изменения структуры объекта пользователь не может с ним работать, поэтому возможно появление следующего предупреждения (Рис. 3.10):



**Рис. 3.10.**

Данное сообщение 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. 3.11).

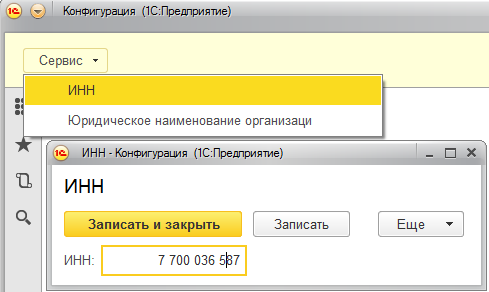


**Рис. 3.11.**

Этот пример достаточно ясно демонстрирует основы работы системы "1С:Предприятие". На этапе конфигурирования создаются необходимые структуры для хранения информации и далее на этапе исполнения эти структуры заполняются необходимыми данными.

***Самостоятельно***

***Создайте новую константу  ГенеральныйДиректор для хранения в информационной базе информации о генеральном директоре, которого зовут Остап Бендер.***



# 3. Справочники

## 3.1. Общее описание и назначение

 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.

Для реализации нашей системы учета, согласно требованиям заказчика, необходимо хранить справочную информацию о сотрудниках, отделах, товарах, клиентах и их договорах. Следовательно, необходимо создать соответствующие 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. 4.1).

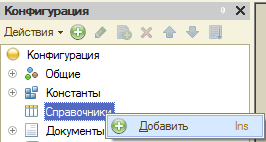


Рис. 4.1.

Все эти действия одинаковые и приведут к появлению на экране окна редактирования (Рис. 4.2) объекта конфигурации. В дальнейшем это окно можно вызвать двойным щелчком мышки на выбранном справочнике.

Поскольку в системе может быть несколько справочников, то необходимо их различать. 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. В целом, все точно так же, как и для констант.

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

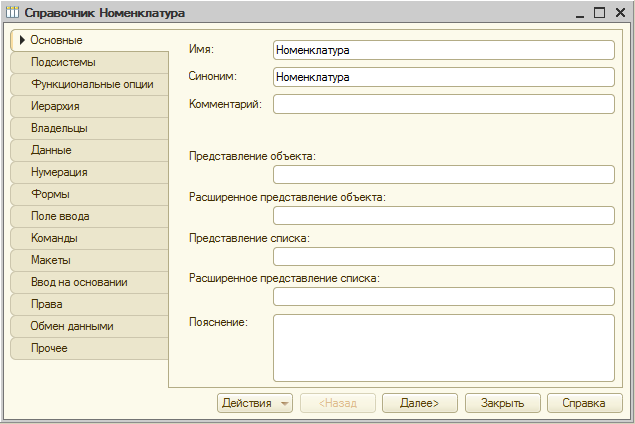


Рис. 4.2.

На этом создание простейшего справочника завершено. Переходим в пользовательский режим предприятия и смотрим на то, что мы только что создали (Рис. 4.3). Все необходимые действия при работе со справочником можно произвести путем нажатия на кнопку "Еще" и выбора соответствующего пункта в выпадающем меню. Наиболее востребованные действия доступны через командную панель или контекстное меню, вызываемое правой кнопкой мышки.

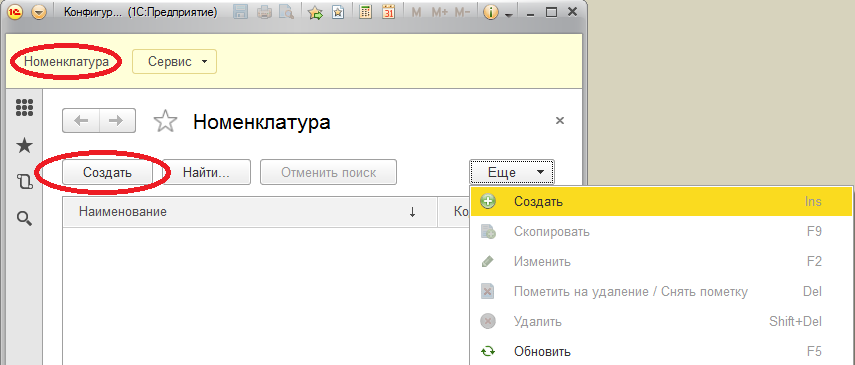


Рис. 4.3.

Для добавления информации о товаре нажмем на кнопку "Создать" и в открывшемся окне укажем наименование нашего товара (Рис. 4.4):

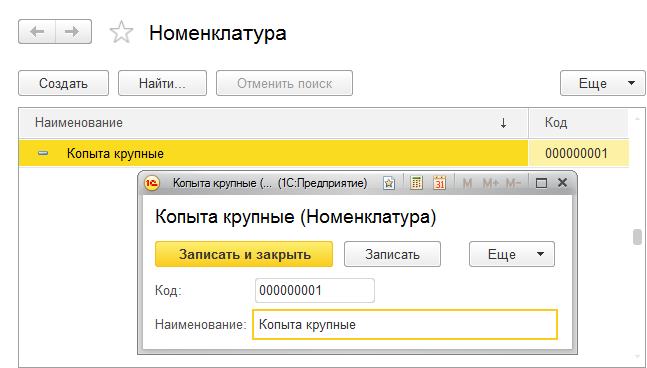


Рис. 4.4.

Заполним справочник следующими данными (Рис. 4.5):

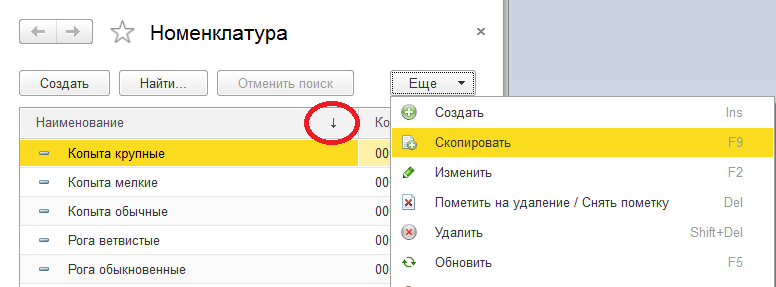


Рис. 4.5.

Обратите внимание на стрелку, которая позволяет сортировать данные справочника для более удобной работы с ними. Кроме того, для поиска нужной информации в справочнике существует механизм поиска, который позволяет найти информацию, если мы хоть как-то можем указать, что именно нас интересует. В приведенном примере используется поиск номенклатуры, содержащей в своём наименование сочетание букв "обы" (Рис. 4.6).

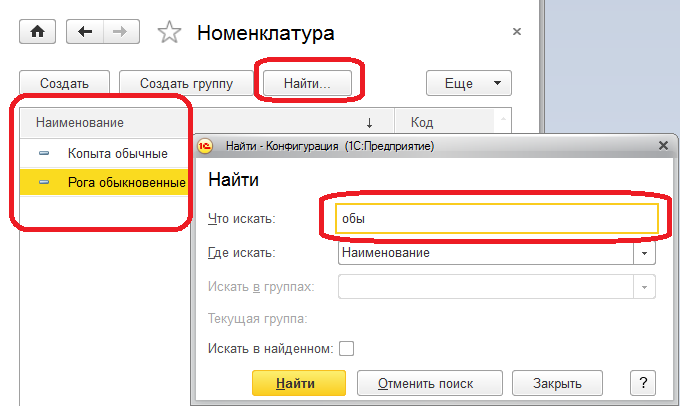


Рис. 4.6.

## 3.2. Иерархия

Созданный нами справочник позволяют комфортно работать только с небольшими наборами данных. Практически сразу возникает вопрос, как быть с большими объемами данных, удобно ли будет работать с подобным списком, состоящим из нескольких сотен элементов? Для удобства работы можно справочник сделать иерархическим, т.е. создать древовидную структуру. Самым простым примером может служить расположение файлов на компьютере, когда многие тысячи файлов располагаются в папках различной степени вложенности, что позволяет быстро в них ориентироваться.

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*. Чтобы разобраться, для чего они нужны, рассмотрим следующие ситуации.

Предположим, что мы продаем компьютер. Будем считать, что он состоит из системного блока, монитора и периферийного оборудования. Мы можем продать его как целиком, так и по частям. В этом случае, компьютер должен выступать одновременно как товар, и как "папка", как объединяющий узел дерева для комплектующих, из которых он состоит. В таком случае, когда разницы между группирующим и группируемым элементом нет, следует выбирать иерархию элементов.

Теперь рассмотрим другую ситуацию. В связи с растущим спросом, наша организация решила сосредоточить торговлю на одежде и обуви. Соответственно, в справочнике нам удобно будет сгруппировать наши товары по их виду. В то же время, сам группирующий элемент не будет нести самостоятельной ценности, а будет служить для удобства навигации  или в качестве объединяющего признака в различных отчетах, по которому можно выводить итоги.  Для этой цели в справочнике существует отдельный вид специальных элементов, которые называются *Группа справочника*, и для их использования необходимо выбрать вид иерархии *Иерархия групп и элементов*. (Рис 4.7) 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*.

Исходя из условий нашей задачи, справочник "Номенклатура" сделаем иерархическими с видом иерархии *Иерархия групп и элементов*.

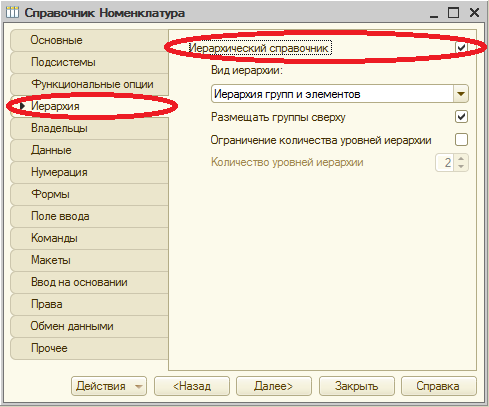


Рис. 4.7.

Для пользователя теперь внешний вид справочника изменится. Помимо элементов появятся еще и группы. Работа с ними аналогична работе с файлами и папками в проводнике (Рис. 4.8).

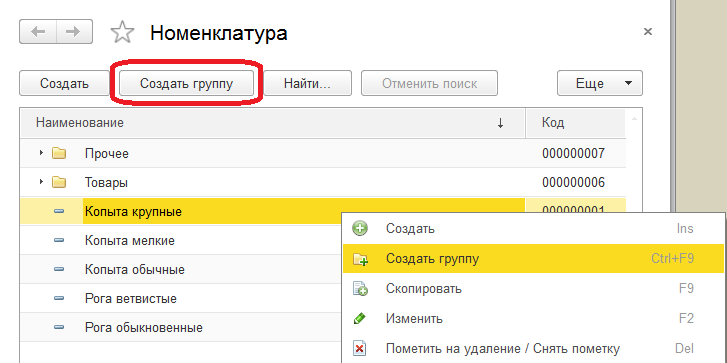


Рис. 4.8.

Новые элементы и группы можно сразу создавать в нужной группе, а для существующих элементов имеется возможность перемещения по группам. Таким образом, создав удобную иерархическую структуру, в дальнейшем пользователь получает возможность быстрой и легкой навигации в справочнике.

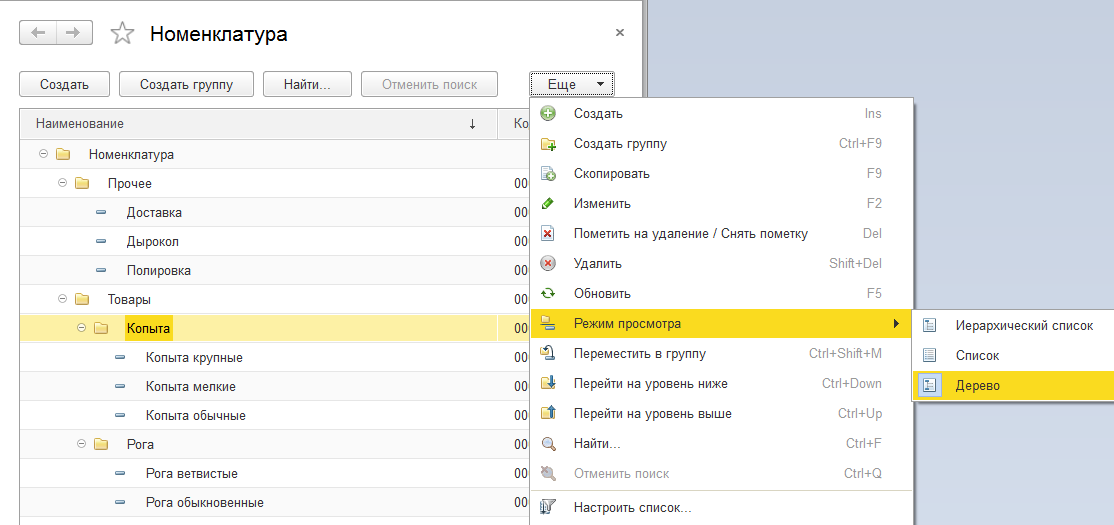


Рис. 4.9.

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. 4.9).

***Самостоятельно***

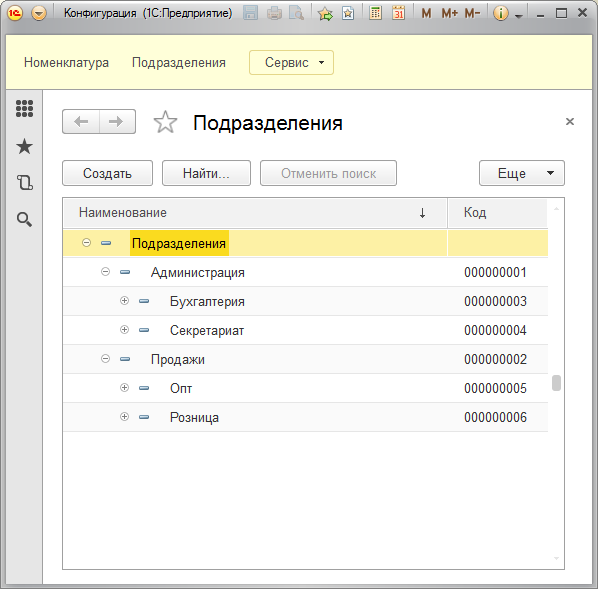


Рис. 4.10.

***Создайте новый справочник Подразделения для хранения данных и подразделения организации. Справочник должен иметь иерархию элементов (Рис. 4.10) .***

## 3.3. Предопределенные элементы

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.

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.4.11), либо по кнопке "Предопределенные" на закладке *Прочее* вызываем окно для редактирования предопределенных элементов справочника (Рис. 4.12).

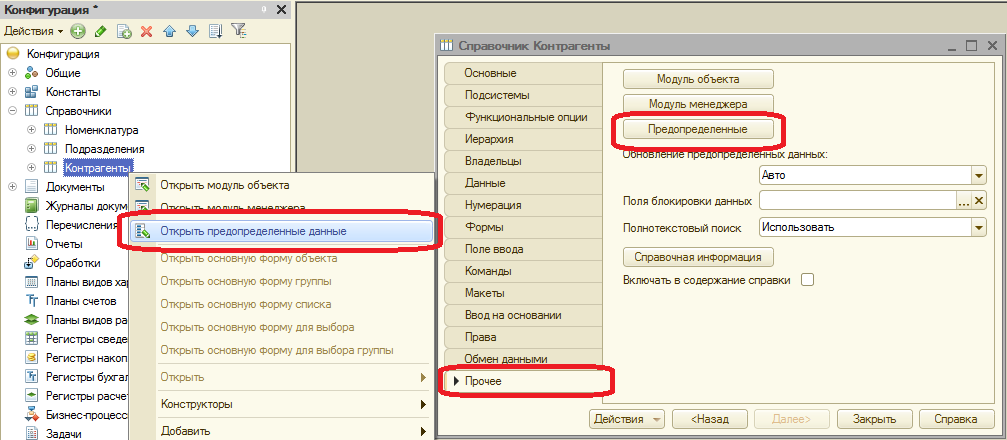


Рис. 4.11.

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".

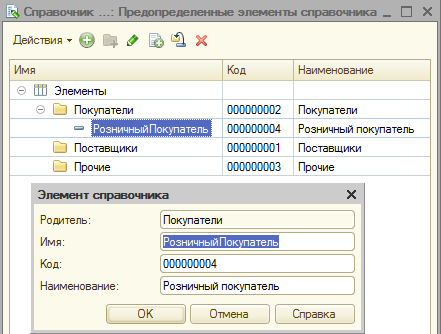


Рис. 4.12.

В режиме "1С:Предприятие" для пользователя будет уже готовая иерархия. Заполним справочник данными (Рис. 4.13).

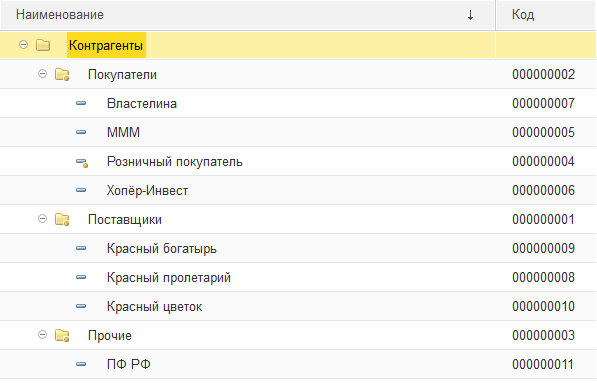


Рис. 4.13.

## 3.4. Реквизиты справочника

До сих пор мы рассматривали справочник как список элементов, а теперь обратим более подробное внимание на элементы этого списка.

Как уже упоминалось ранее, 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*.

Поскольку справочник постоянно пополняется новыми элементами, необходимо как-то идентифицировать их, и для этой цели обычно предназначен реквизит *Код*, а реквизит *Наименование* обычно служит для хранения смысловой информации об элементе.

На закладках *Данные* и *Нумерация* необходимо определить следующие свойства кода справочника (Рис 4.14 и Рис 4.15):

         *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).

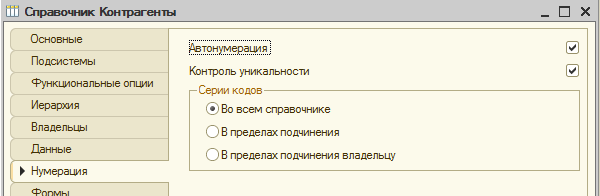


Рис. 4.14.

         *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.

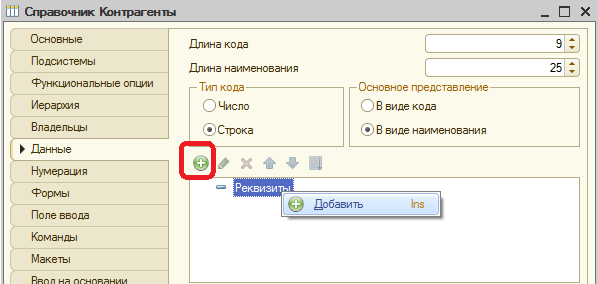


Рис. 4.15.

Помимо системных реквизитов в справочник можно добавлять свои необходимые реквизиты. 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 4.15). Тот же результат можно достичь, выбрав пункт *Добавить* в панели инструментов этого поля.

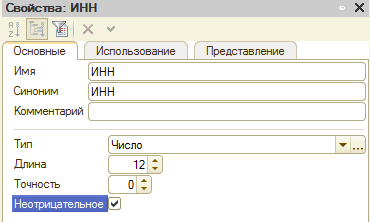


Рис. 4.16.

Далее необходимо определить свойства нашего реквизита в соответствующем окне. (Рис. 4.16) В дальнейшем палитру свойств реквизита можно вызвать двойным щелчком мышки на данном реквизите, либо на данном реквизите правым щелчком мышки вызвать контекстное меню, в котором выбрать пункт *Свойства*. В справочнике "Контрагенты" создадим новый реквизит *ИНН* (тип *Число,* длина 12).

Для нас существенными будут являться следующие свойства реквизита:

         *Имя* – идентификатор реквизита

         *Использование* – определяет, к какому элементу справочника (элементу или группе) будет относиться данный реквизит

         *Тип данных* – определяет тип данных реквизита. Данные могут быть примитивного или ссылочного типа, а также составного типа.

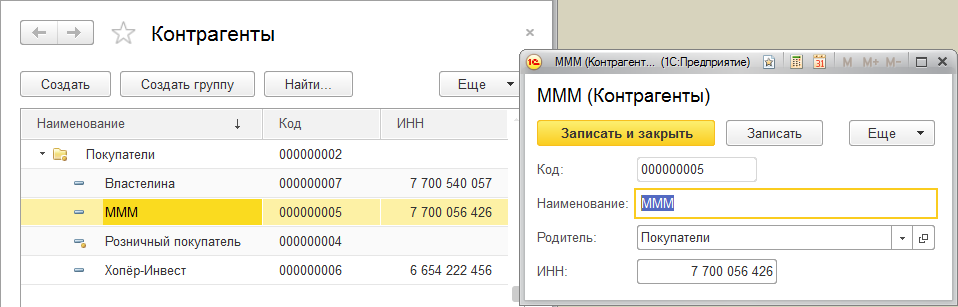


Рис. 4.17.

После обновления базы для пользователя появляется возможность у каждого контрагента указать ИНН (Рис. 4.17).

С помощью реквизитов может быть решена и следующая задача: необходимо хранить наименование нашего контрагента, но это наименование очень длинное. Поскольку реквизит *Наименование* является системным, то на него наложены определенные ограничения, в том числе и на его длину. С одной стороны, наименование нашего контрагента должно храниться в реквизите *Наименование*, с другой стороны, оно там не помещается, как быть? В этом случае можно в реквизите *Наименование* хранить краткое наименование контрагента для внутреннего использования, а полное наименование хранить в дополнительном реквизите, который уже сможет хранить информацию полностью.

***Самостоятельно***

***Создайте в справочнике Контрагенты новый реквизит для хранения полного юридического наименования контрагента.***

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. 4.18).

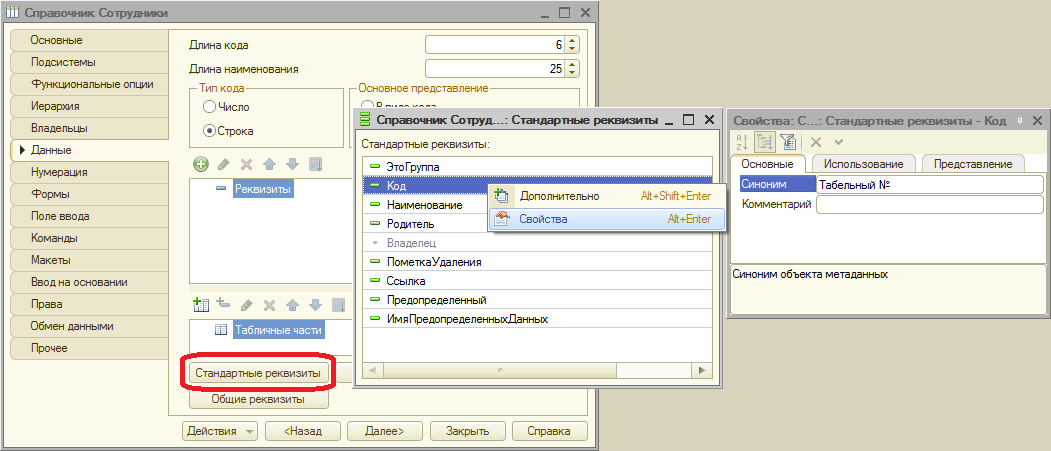


Рис. 4.18.

Теперь добавим реквизит *Подразделение*, с типом данных *СправочникСсылка.Подразделения*. (Рис. 4.19)

В этом случае, вновь созданный реквизит будет заполняться значениями из справочника *Подразделения*  (Рис. 4.20), 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*.

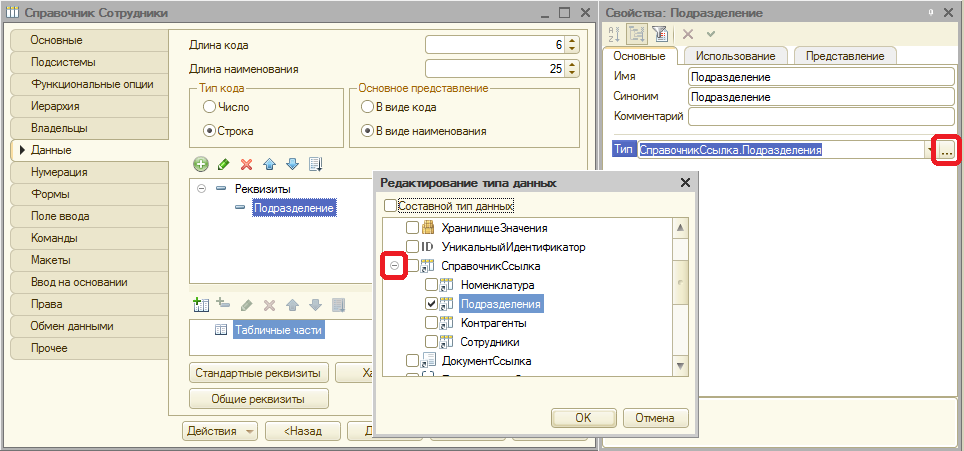


Рис. 4.19.

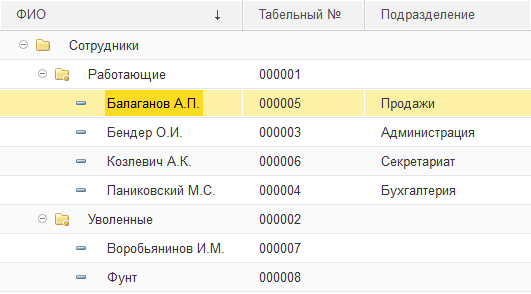


Рис. 4.20.

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.***

## 

## 3.5. Табличная часть

При решении прикладной задачи может возникнуть необходимость хранить в реквизите не одно значение, а несколько. Например, необходимо хранить информацию о детях сотрудника. Для решения такого рода задач в системе предусмотрено понятие *Табличная часть*.

*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.

Согласно пожеланию нашего заказчика, необходимо хранить информацию об учебных заведениях и детях сотрудника. Данная информация самостоятельной ценности не имеет, поэтому должна быть реквизитом справочника. Поскольку наличие и количество детей неизвестно, но может быть значительно больше одного, то необходимо создать не просто реквизит, а табличную часть.

Кроме информации о детях аналогичная ситуация возникает и для сведений об образовании, причем, поскольку между собой эти данные никак не связаны, то и хранить их следует по отдельности. Таким образом, необходимо создать две табличных части.

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. 4.21)

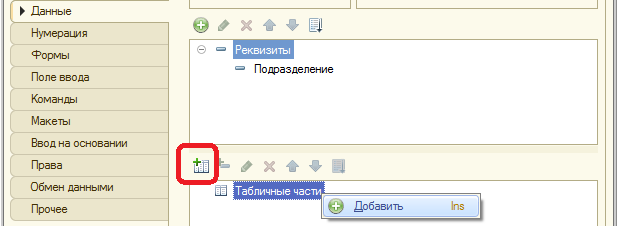


Рис. 4.21.

В дальнейшем палитру свойств можно вызвать двойным щелчком мышки на выбранной табличной части. Поскольку табличных частей может быть несколько, то для однозначного определения нужной нам табличной части необходимо задать ее имя. На имя табличной части налагаются все те же условия, как и на имена справочников или констант. Первую табличную часть назовем *Дети*. (Рис. 4.22)

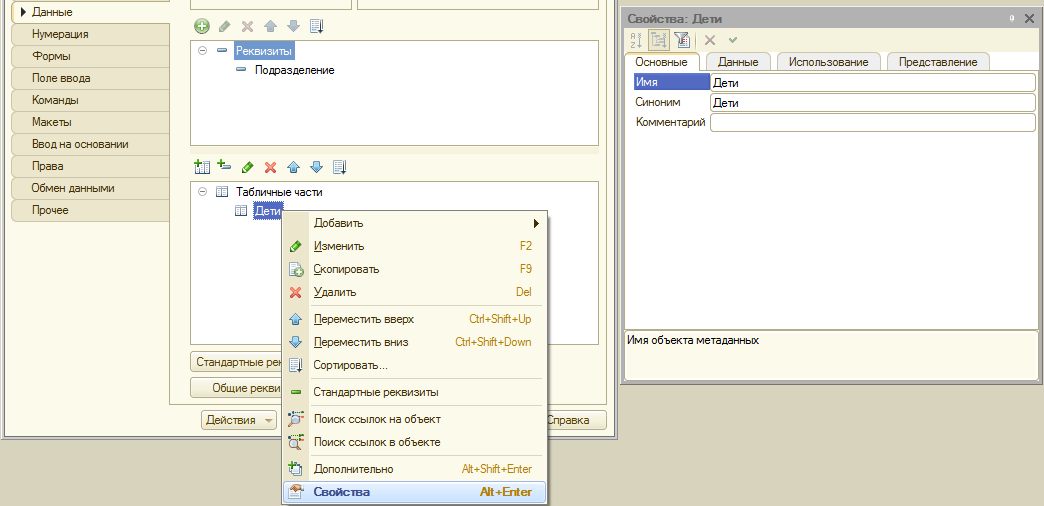


Рис. 4.22.

Определим, будет ли данная табличная часть относиться только к элементам, только группам или к ним обоим. В нашем случае информация о детях по отношению к группе справочника не имеет смысла, поэтому в поле *Использование* оставляем значение *Для элемента*.

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. 4.23)

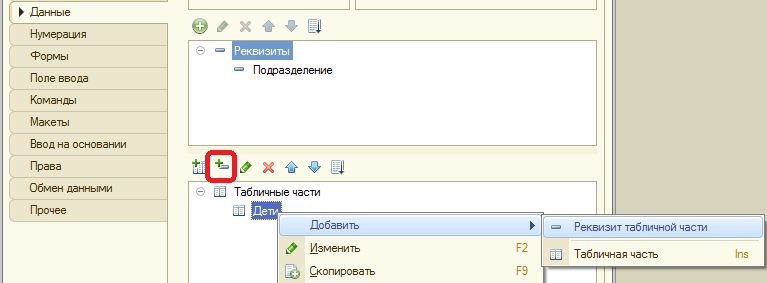


Рис. 4.23.

Для хранения данных о фамилии, имени и отчестве ребенка создадим строковый реквизит с длиной 50 символов, который назовем "ФИО", а для даты рождения - соответственно реквизит *ДатаРождения* типа *Дата* с составом даты *Дата*, поскольку время рождения ребенка нас не интересует (Рис. 4.24).

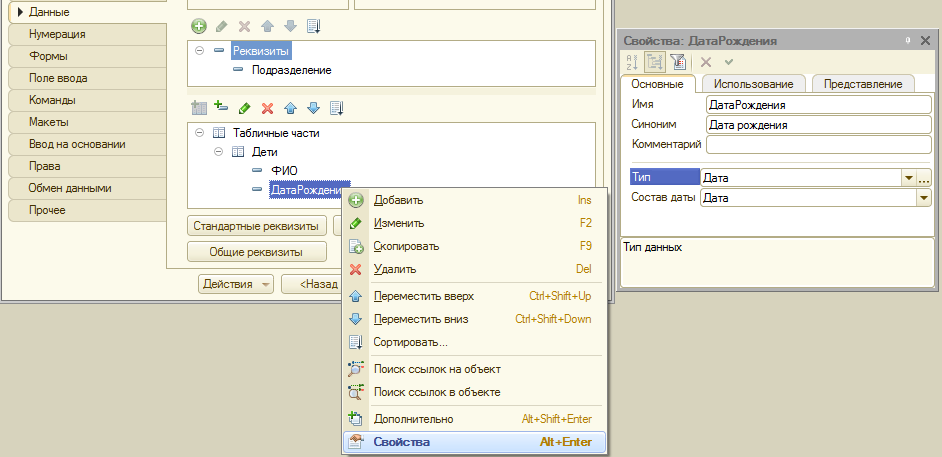


Рис. 4.24.

В результате таких действий в пользовательском режиме справочник станет выглядеть так, как указано на Рис. 4.25

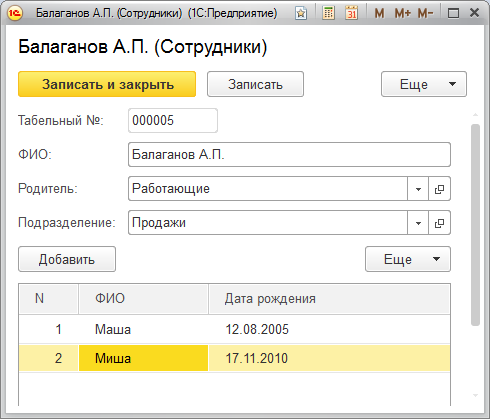
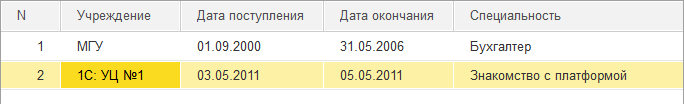


Рис. 4.25.

***Самостоятельно***

***Добавьте в справочнике Сотрудники табличную часть Образование (Рис. 4.26)***



**Рис. 4.26.**

## 3.6. Подчинение

Рассмотрим следующую ситуацию: взаиморасчеты с контрагентами ведутся в разрезе договоров, т.е. у любого контрагента может быть несколько договоров. В принципе, можно создать новый справочник, в котором договоры будут элементами, а для того, чтобы объединить договоры одного контрагента, использовать группы. Однако данный способ не совсем удобен. Во-первых, при большом количестве контрагентов, для договоров придется организовывать систему вложенных групп еще более сложного уровня, а во-вторых, нам все равно придется вручную искать нужную группу договоров, соответствующую выбранному контрагенту. Система "1С:Предпрятие 8" позволяет осуществлять подобные отборы на системном уровне. В рамках решения нашей задачи рассмотрим реализацию данного механизма более подробно.

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. 4.27).

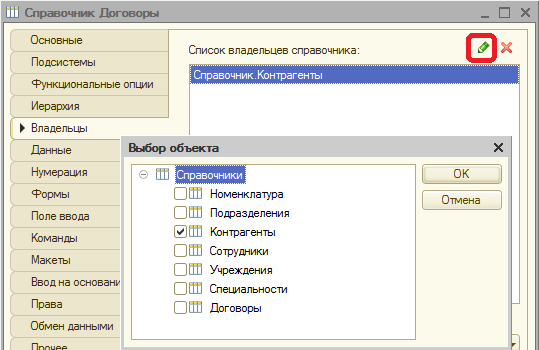


Рис. 4.27.

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. 4.28).

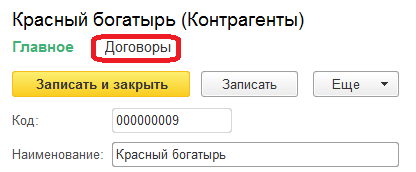


Рис. 4.28.

Возврат к данным контрагента происходит через ссылку "Главное", а работа с договорами происходит стандартным образом, также как с любым другим справочником (Рис. 4.29).

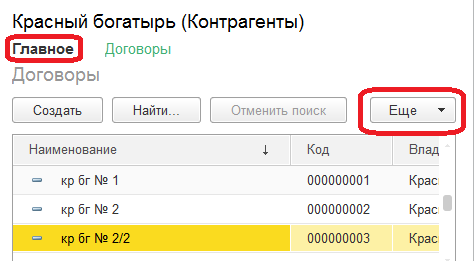


Рис. 4.29.

## 3.7. 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*. В том случае, когда разработчик не предпринимал никаких дополнительных мер по настройке объекта, системой будет автоматически создана соответствующая форма, с которой и будет работать пользователь. Если же по каким-либо причинам автоматическая форма не устраивает пользователя, то разработчик имеет возможность самостоятельно создать необходимую форму.

Создадим форму списка для справочника *Контрагенты*. Для этой цели перейдем на закладку формы и создадим новую форму списка с помощью конструктора (Рис. 4.30).

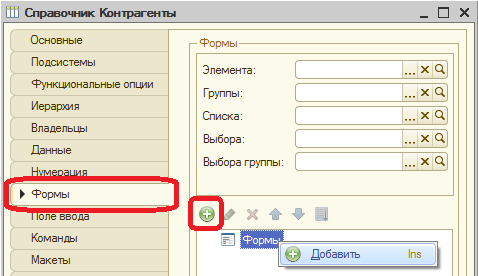


Рис. 4.30.

Будем последовательно делать настройки, которые предлагает конструктор. На первой закладке укажем, что создаем форму списка (Рис. 4.31).

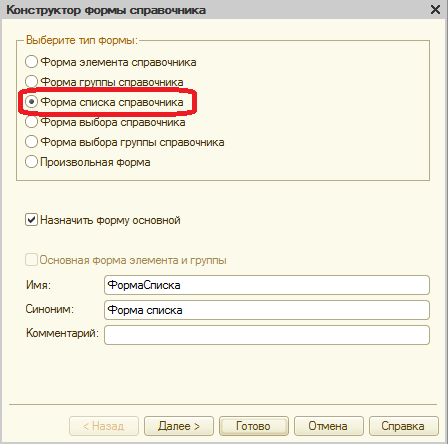


Рис. 4.31.

Далее отметим те реквизиты справочника, которые мы хотим видеть (Рис. 4.32)

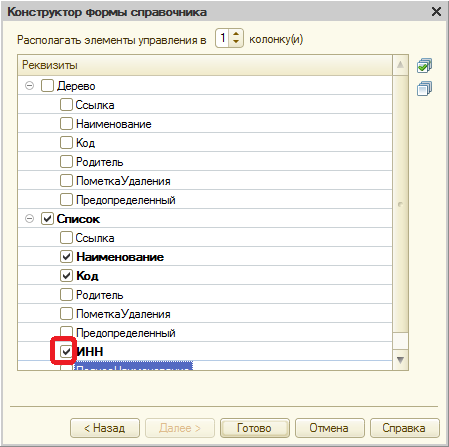


Рис. 4.32.

В результате будет создана форма (Рис. 4.33)

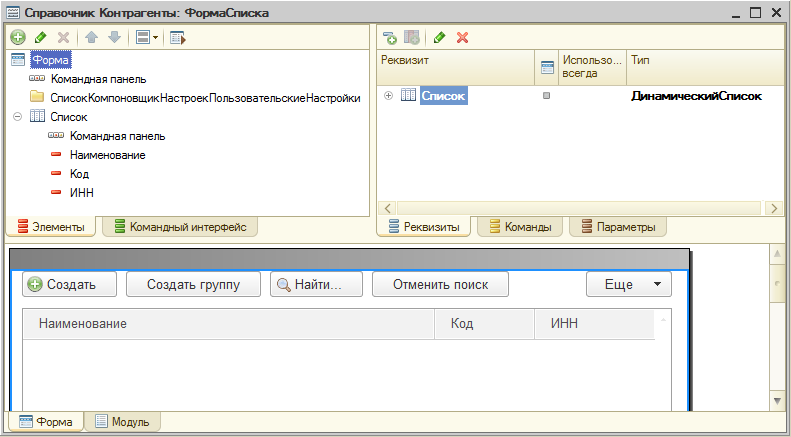


Рис. 4.33.

Откроем закладку Команды – Глобальные команды и перетащим элемент "Элементы.Список.ТекущиеДанные.Ссылка" в левое окно в "Командная панель" (Рис. 4.34).

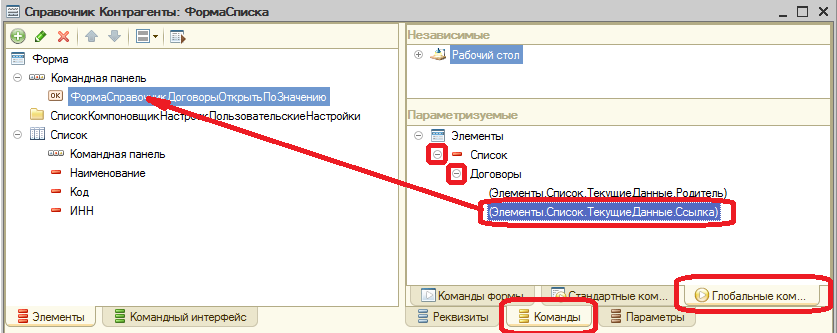


Рис. 4.34.

В свойствах появившегося элемента укажем "ТолькоВоВсехДействиях" в значение "Нет" (Рис.4.35)

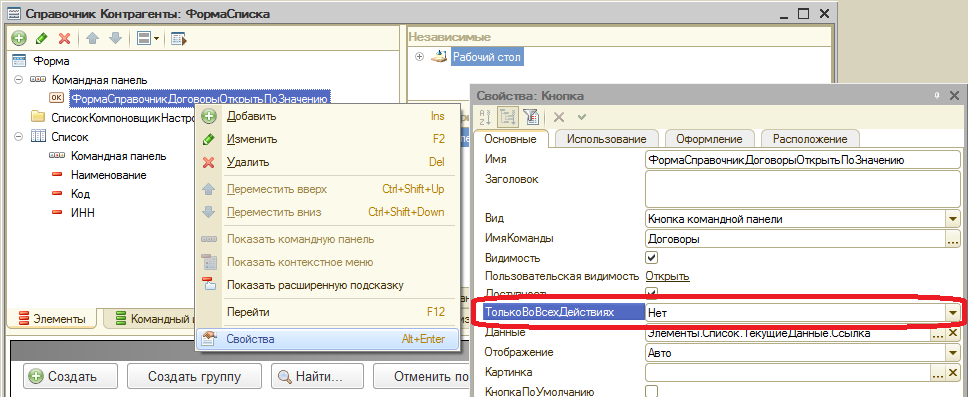


Рис. 4.35.

Теперь в пользовательском режиме в форме списка справочника сотрудники и у пользователя появится возможность просматривать только те договоры, которые относятся к текущему контрагенту (Рис. 4.36).

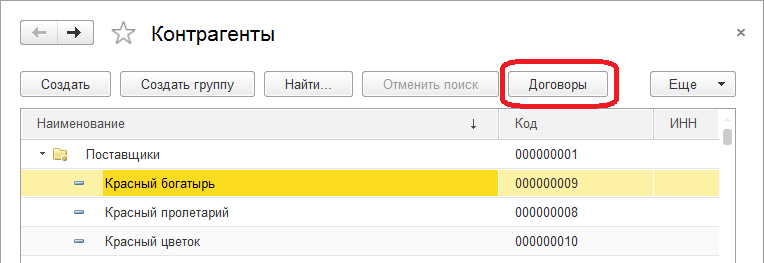


Рис. 4.36.

Аналогичным образом может быть создана и форма элемента для справочника (Рис. 4.37). Главное, это указать правильный тип формы.

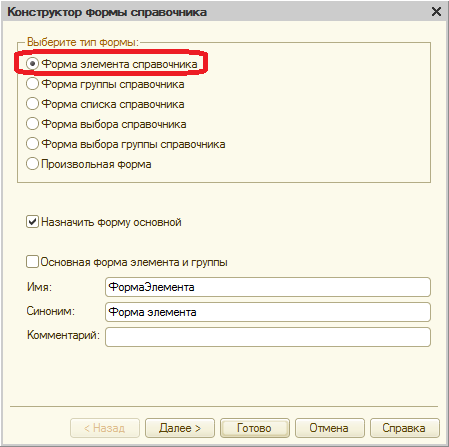


Рис. 4.37.

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. 4.38)

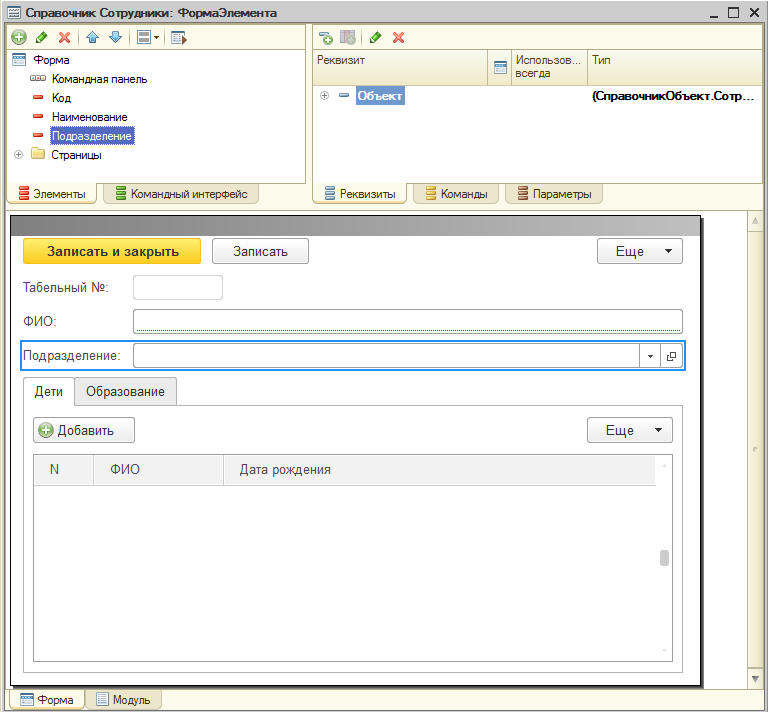


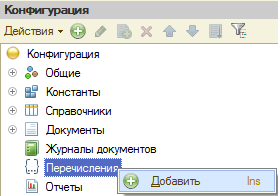
Рис. 4.38.

**4. Перечисление**

*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. Например, если нам точно известно, что сотрудник может быть принят либо совместителем, либо на основное место работы, то логично создать перечисление с соответствующими элементами. Можно конечно создать новый справочник, задать в нем предопределенные элементы, назначить роли так, чтобы пользователи не могли задать новые элементы, однако перечисление использовать проще и быстрее как для программиста и системы, так и для конечного пользователя.

Итак, в нашей базе мы хотим иметь информацию о том, является ли наш сотрудник штатным работником или совместителем. Каким образом это сделать? Ну, во-первых, эта информация должна быть указана для каждого сотрудника, и для этого необходимо создать новый реквизит справочника *Сотрудники*. Во-вторых, сотрудник может быть либо штатным, либо совместителем, поэтому логично создать новое перечисление и именно его указать в качестве типа данных нашего нового реквизита справочника.

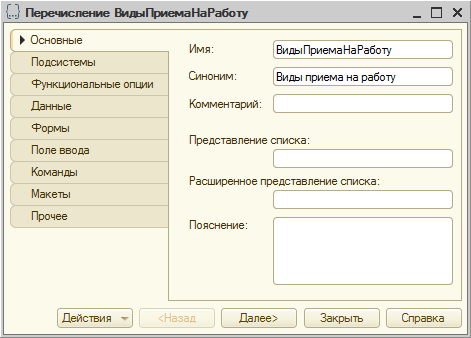
Создаем новое перечисление *ВидыПриемаНаРаботу*. Для этого правой кнопкой мыши на ветке Перечисления вызываем контекстное меню и выбираем пункт *Добавить*. (Рис. 5.1)



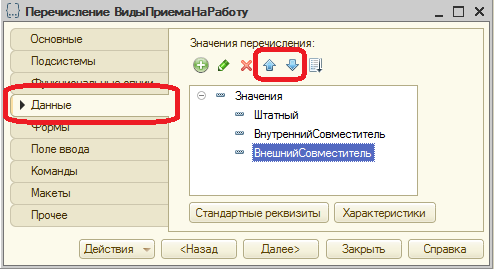
**Рис. 5.1.**

Далее задаем имя*:* *ВидыПриемаНаРаботу*, при этом автоматически заполняется поле *Синоним* точно так же, как это происходило для справочников. (Рис. 5.2)

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. 5.3), в которой задаем *Имя* - значения перечисления. Аналогично создаем следующее значение.



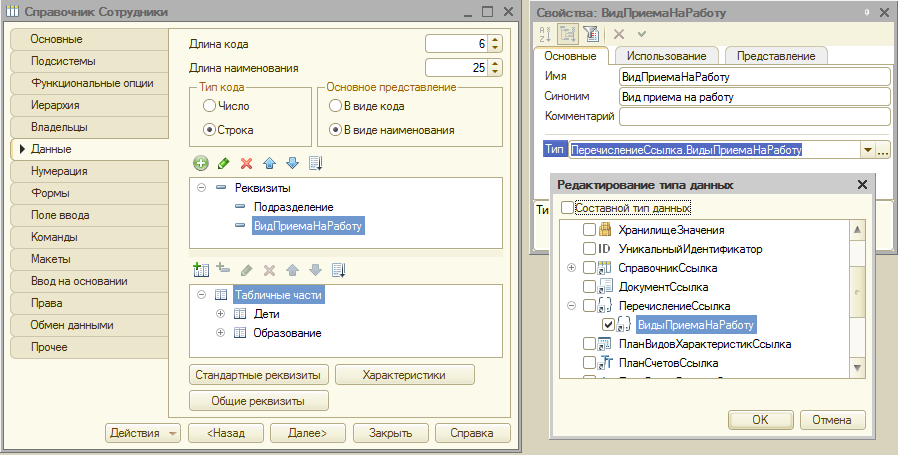
**Рис. 5.2.**



**Рис. 5.3.**

Следует обратить внимание на то, что в отличие от реквизитов справочника значения перечисления должны быть расположены именно в том порядке, в котором их желает видеть пользователь. Изменить порядок значений перечисления можно с помощью панели инструментов кнопки *Переместить вверх* и *Переместить вниз*, либо с помощью аналогичных пунктов контекстного меню, вызванного правой кнопкой мышки на выбранном значении.

После всех этих действий в справочник *Сотрудники* добавим новый реквизит *ВидПриемаНаРаботу* с типом данных *ПеречислениеСсылка.ВидыПриемаНаРаботу* (Рис. 5.4).



**Рис. 5.4.**

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.

# 5. Элементы управления формы

Как уже упоминалось выше, пользователь работает не напрямую с объектом, а с экранной формой, причем с одним и тем же объектом пользователь может работать посредством разных форм. Система предоставляет разработчику возможность использовать ряд стандартных механизмов (элементов формы), с помощью комбинаций которых можно создать практически любой функционал и внешний вид. Прямой аналогией может служить детский конструктор "Лего", когда из некоторого набора "кубиков" можно собрать что-то совершенно новое. Далее мы рассмотрим ряд таких стандартных механизмов более подробно. Причем следует иметь в виду, что данные возможности экранной формы не привязаны к какому-то конкретному виду объектов конфигурации и, соответственно, могут применяться для любой экранной формы любого объекта.

Хотя различные элементы формы предназначены для разных целей и обладают своими специфическими особенностями, тем не менее, многие свойства буду важны для всех них.

*Имя* – системное имя, уникальный идентификатор, позволяющий обратиться именно к данному элементу

*ПутьКДанным* – устанавливает связь между тем элементом формы, что отображается на форме, и тем реквизитом, где храниться информация

*Видимость* – позволяет либо показать, либо спрятать от пользователя элемент формы

*Доступность* – позволяет пользователю видеть элемент формы, но не позволяет его интерактивно изменять

Любой из элементов формы можно добавит используя пункт *Добавить* из контекстного меню, вызываемого правой кнопкой мышки (Рис. 6.1).

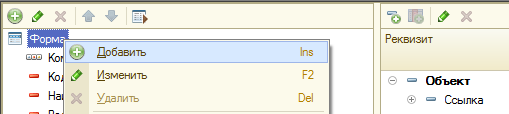


Рис. 6.1.

Правой кнопкой мышки через контекстное меню для любого элемента формы можно вызвать его палитру свойств для настройки необходимых механизмов.

## 5.1. Поле

Элемент формы *Поле* предназначен для отображения данных на форме и предоставления пользователю возможности эти данные изменить. В зависимости от свойства *Вид* может принимать различный внешний вид.

### 6.1.1. Поле ввода

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 вода, в котором будут отражаться данные реквизита, можно воспользоваться двумя различными способами. Можно добавить новый элемент формы, указав тип *Поле* (Рис. 6.2) и указав затем в его свойствах *ПутьКДанным*, *Вид* и *Заголовок* (Рис. 6.3)

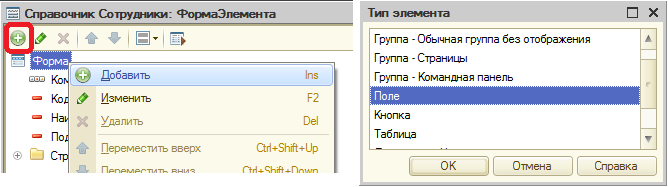


Рис. 6.2.

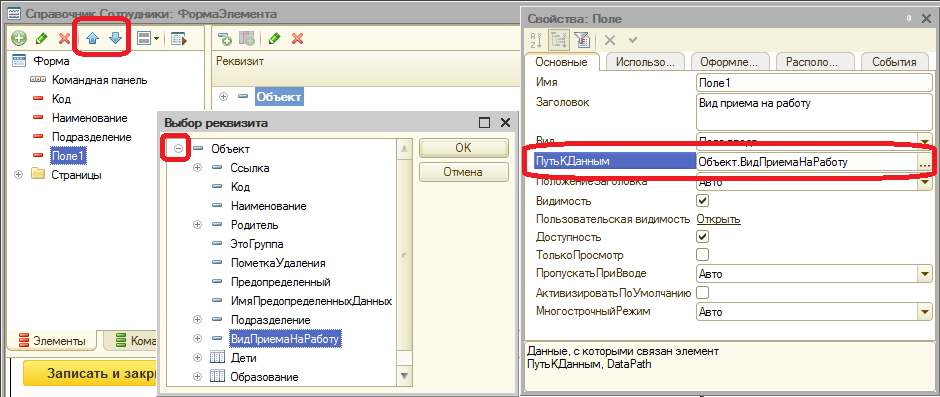


Рис. 6.3.

Того же результата можно достичь, если придерживая левую кнопку мышки перетащить реквизит из окна реквизитов в окно элементов формы (Рис. 6.4)

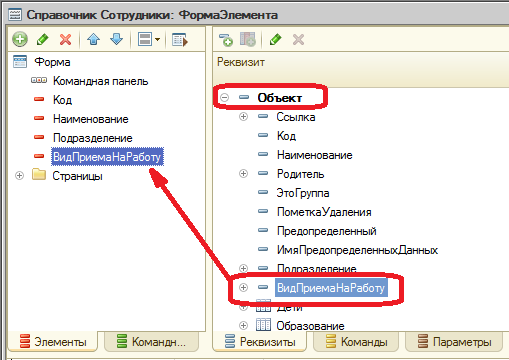


Рис. 6.4.

Для поля ввода наиболее важными свойствами будут являться:

*Заголовок* -  текстовый заголовок, видимый пользователю

*Положение заголовка* – положение заголовка относительно поля ввода

*Пропускать при вводе* – позволяет автоматически перемещать курсор на поле при редактировании элементов на форме

*Активизировать по умолчанию* – позволяет сделать поле активным при открытии формы

*Предупреждение при редактировании* – позволяет выдавать пользователю предупреждение при редактировании данных поля, в том случае, если указано *ОтображениеПредупрежденияПриРедактировании*

### 6.1.2. Поле флажка

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

Создадим форму элемента для справочника *Номенклатура* (Рис. 6.5).

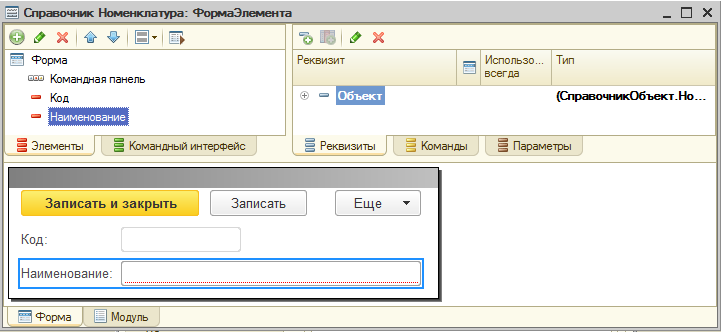


Рис. 6.5.

 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. 6.6)

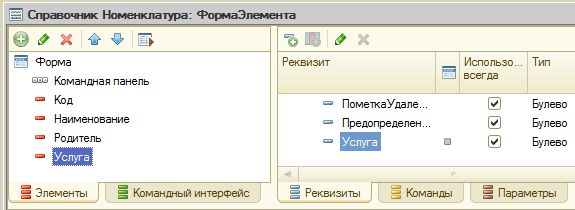


Рис. 6.6.

Поскольку реквизиты с типом *Булево* удобнее всего отображать на форме в виде флажка, то система сама автоматически укажет данный вид поля (Рис. 6.7)

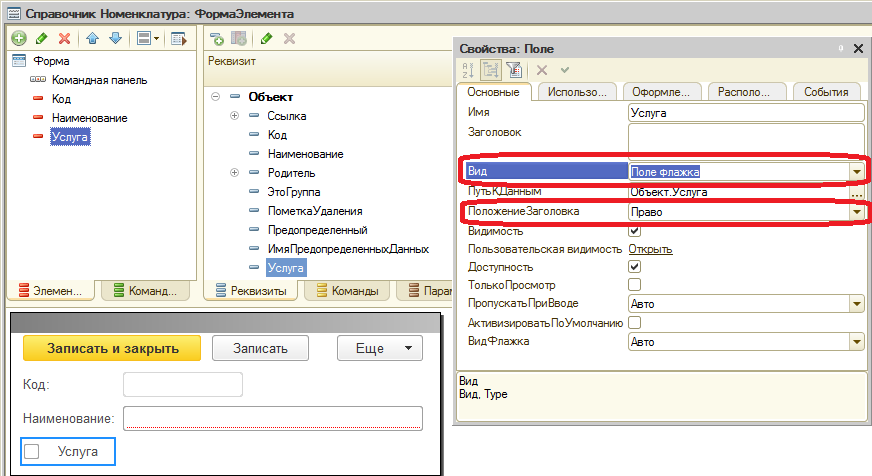


Рис. 6.7.

Для флажка наиболее важными для нас будут являться следующие свойства:

*Заголовок* – текстовый заголовок, видимый пользователю

*Положение заголовка* – положение заголовка относительно флажка

***Самостоятельно***

***Добавьте в справочнике Номенклатура новый реквизит "Весовой", который на форме должен отображаться в виде флажка.***

### 6.1.3. 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. 6.8)

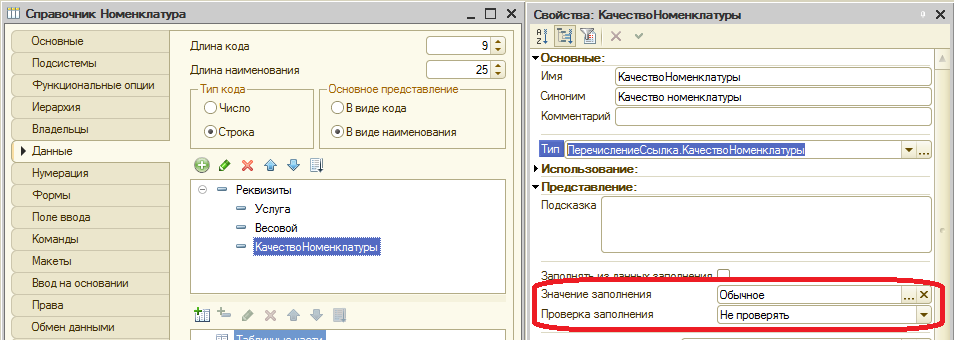


Рис. 6.8.

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. 6.9).

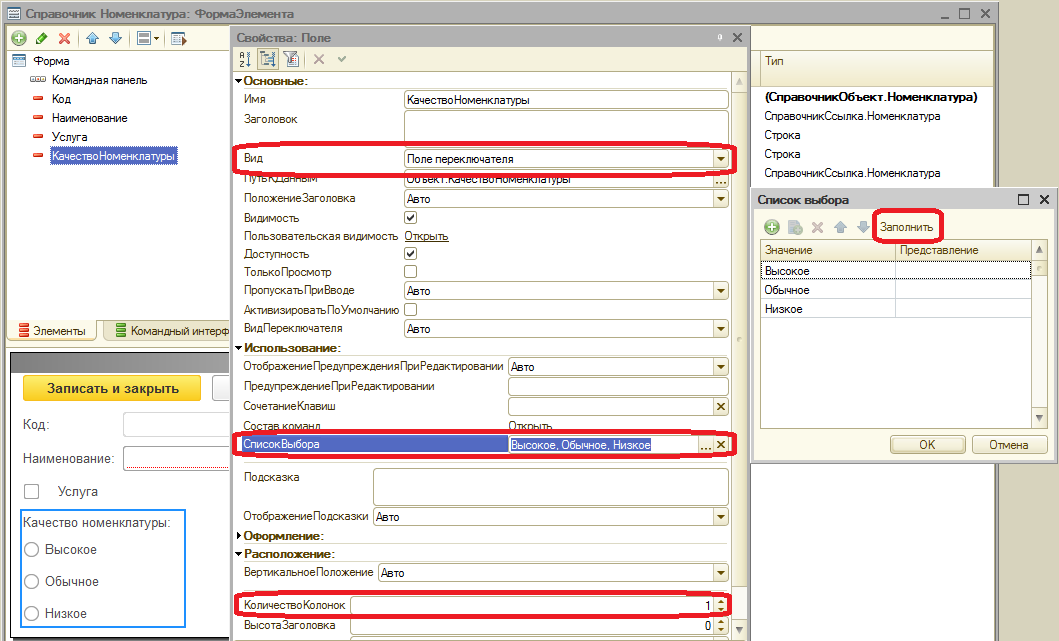


Рис. 6.9.

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

***Самостоятельно***

***Добавьте в справочнике Номенклатура новый реквизит "Происхождение номенклатуры", который на форме должен отображаться в виде переключателя из 3-х значений: "отечественный", "импортный" и "неизвестно".***

## 5.2. Команда

*Команда* позволяет пользователю возможность выполнить какие-либо из заранее определенных на этапе конфигурирования действий. Эти действия могут быть либо стандартными (Рис. 6.10), либо описаны на встроенном языке в модуле формы и затем связаны с командой (Рис. 6.11) посредством кнопки (или гиперссылки) и выполняться при нажатии пользователем на этот объект.

Фактически, команда является самостоятельным объектом, а элементом формы является только кнопка, вызывающая команду. Однако, в целях последовательности изложения материала, объект команда рассмотрен именно в текущем разделе.

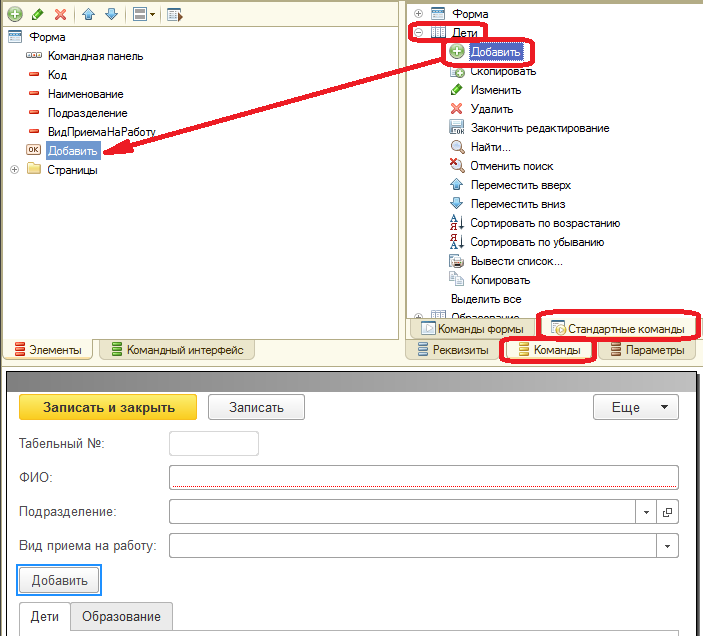


Рис. 6.10.

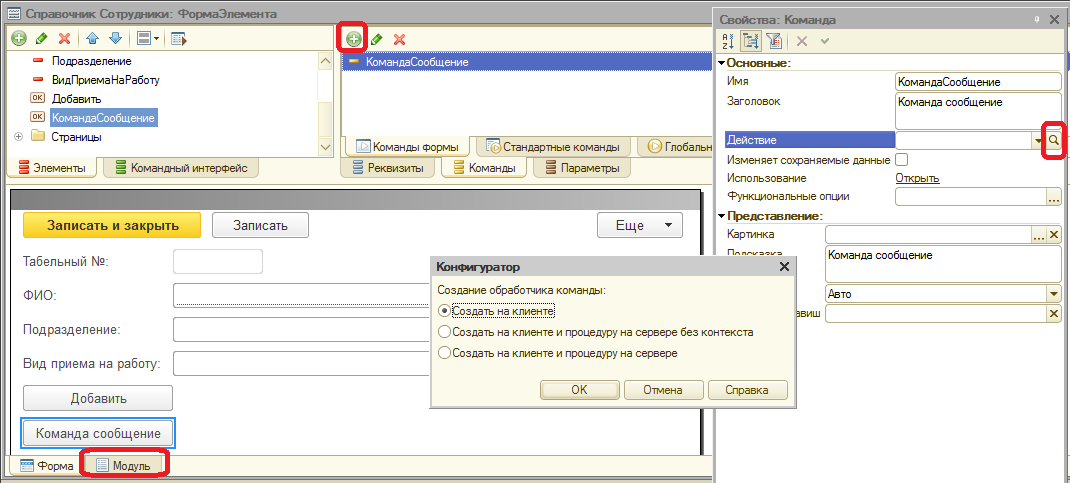


Рис. 6.11.

Написание программного кода можно упростить, если использовать конструкторы. В нашем случае удобно использовать кнопку выбора, которая создаст в модуле новую процедуру и свяжет ее с командой.

Текст процедуры, выполняемой при нажатии кнопки:

&НаКлиенте

Процедура Команда1(Команда)

Сообщить("Нажата кнопка");

КонецПроцедуры

В результате у нас используются 3 разных объекта: модуль, команда и кнопка. Каждый из этих объектов имеет свои собственные настройки.

У кнопки важными для нас будут являться следующие свойства:

*Заголовок* – текстовое представление объекта на форме

*Вид* – устанавливает внешний вид команды в виде кнопки или гиперссылки

*ИмяКоманды* – устанавливает связь между объектом формы и выполняемым действием

*Картинка* – позволяет разместить на форме картинку, отображаемую в зависимости от установок свойства *Отображение*

## 5.3. Группа

Группа позволяет объединить несколько рядом расположенных элементов на форме для дальнейших действий с ними. В зависимости от свойства *Вид* может изменять свой внешний вид.

### 6.3.1. Обычная группа

Элемент *Группа* с видом *Обычная группа* позволяет визуально объединить несколько элементов на форме, управлять их расположением на форме (Рис.6.13)

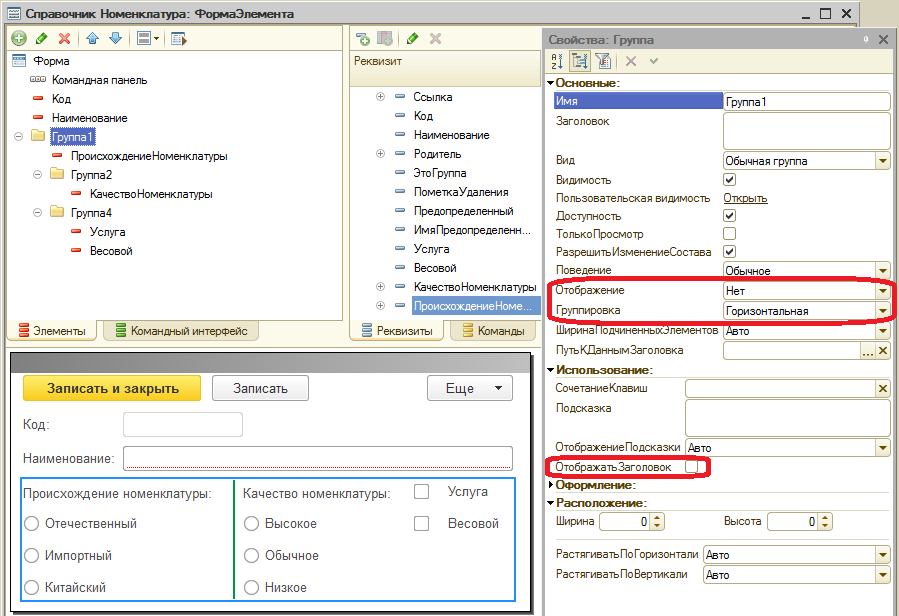


Рис. 6.13.

Для обычной группы наиболее важными будут являться свойства:

*Заголовок* – надпись на форме

*Отображение* – определяет внешний вид группы на форме

*Группировка* – определяет характер расположения на форме элементов, входящих в группу

Используя вложенность групп можно располагать элементы на форме практически любым способом.

### 6.3.2. Командная панель

Командная панель позволяет группировать только кнопки, вызывающих команды, которые могут быть как стандартными, так и созданными разработчиком (Рис. 6.14).

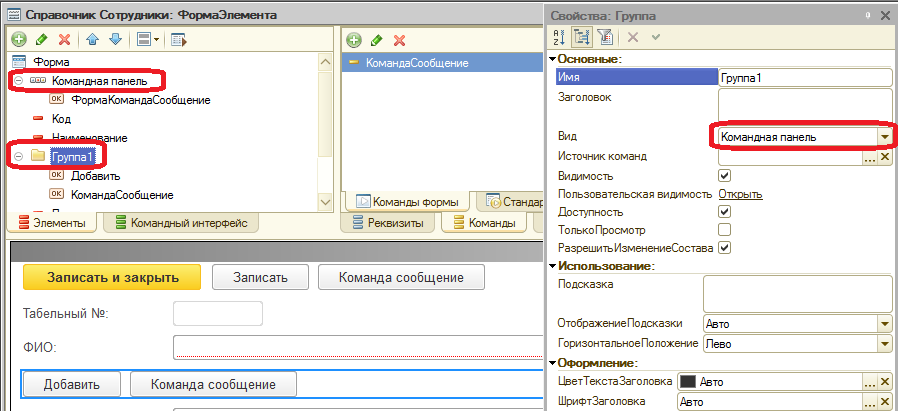


Рис. 6.14.

### 6.3.3. Группа страниц

Группа с видом *Страницы* и *Страница* позволяют разместить элементы на форме в виде нескольких слоёв-страниц. Для этого необходимо создать группу с видом *Страницы*, а затем для каждой страницы в ее составе создать группу с видом *Страница* (Рис. 6.15 и Рис. 6.16) и уже на ней разместить элементы формы обычным порядком.

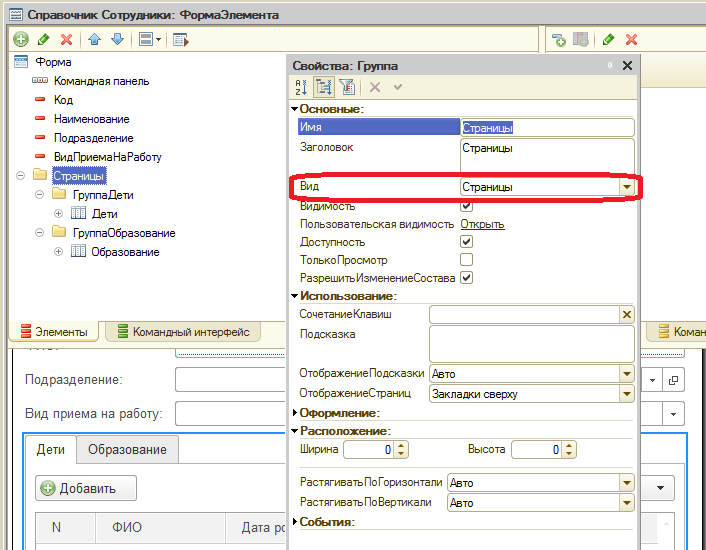


Рис. 6.15.

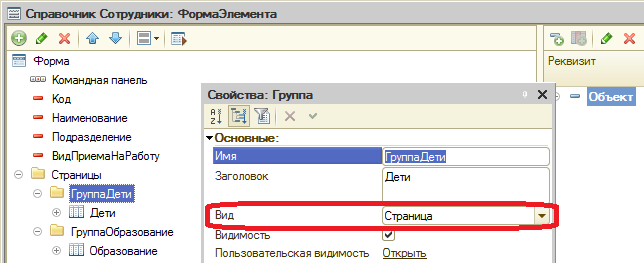


Рис. 6.16.

Для нас будут важны следующие свойства:

*ОтображениеСтраниц* – для группы *Страницы* задает вид отображения закладок на форме

*Заголовок* – для группы *Страница* задает заголовок на закладке формы

## 5.4. Таблица

*Таблица* позволяет отобразить на форме таблицу, например табличную часть объекта (Рис. 6.16)

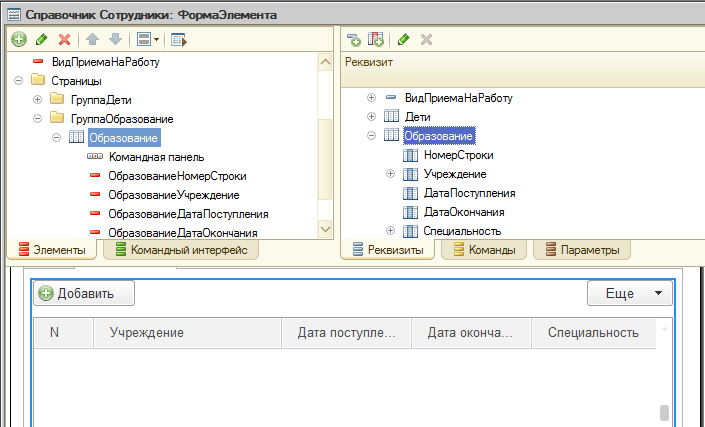


Рис. 6.16.

В состав таблицы могут быть включены различные элементы: командная панель, поле, группа – таким образом определяя внешний вид таблицы на форме.

# 6. Обработчик события

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).

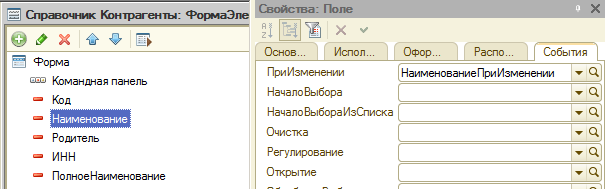


Рис. 7.1.

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)

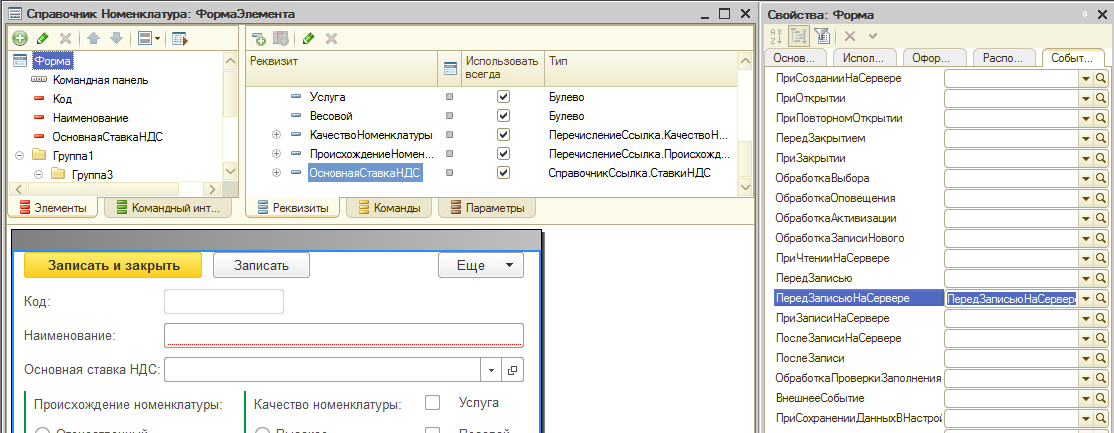


Рис. 7.2.

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

# 7. Документ

*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. Дата документа хранится в формате дата и время с точностью до секунды. В системе может быть создано произвольное количество различных видов документов.

Для отражения в системе факта выплаты денежных средств создадим новый вид документов. Операция по созданию нового вида документа происходит аналогично созданию нового справочника – 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. На это имя налагаются все те же ограничения, что и на имена остальных объектов конфигурации. Назовем создаваемый документ *РасходДенег*, а в качестве *Синонима* укажем *Списание денежных средств* (Рис. 8.1). Далее необходимо определить структуру нашего документа, которая задается набором его реквизитов.

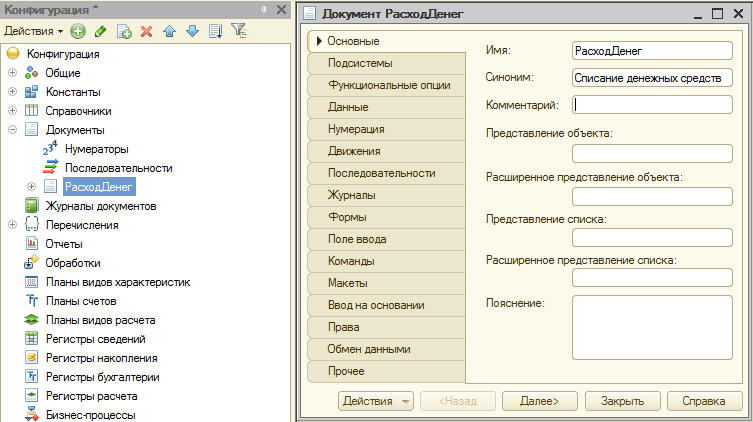


Рис. 8.1.

В зависимости от назначения все реквизиты можно разделить на те, для  которых значение единственное, и те, значения которых являются списком, соответственно, реквизиты документа делятся на реквизиты документа и реквизиты табличной части. Например, обычно в приходной накладной указывается поставщик и перечень поступивших от него товаров, и в таком случае поставщик будет являться реквизитом шапки, а товар реквизитом табличной части.

## 7.1. Реквизиты шапки

В случае списания денежных средств будем считать, что заказчику удобнее работать с каждым платежом по отдельности, следовательно, в этом случае документ будет иметь только реквизиты документа.

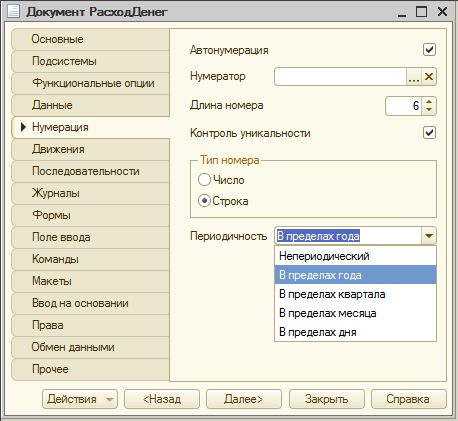


Рис. 8.2.

Как мы уже говорили, для документа всегда существуют реквизиты *Дата* и *Номер*. В окне редактирования документа *РасходДенег* на закладке *Нумерация* необходимо задать следующие свойства, относящиеся к его дате и номеру (Рис. 8.2):

         *Автонумерация* – данный флаг включает автоматическое формирование номера документа при его создании. В последствии номер может быть изменен.

         *Длина номера* – максимальное количество символов, которое может занимать номер документа.

         *Контроль уникальности* – этот флаг включает автоматический контроль системой номера документа. В случае нарушения правил нумерации системой будет выдано соответствующее предупреждение.

         *Тип номера* – определяет тип данных номера документа. Это может быть число или строка. С одной стороны, удобно, когда документы нумеруются по порядку, но бывает необходимо задавать не только числовой порядковый номер, но и буквенный префикс документа. В таком случае весь номер надо определить как строковый, т.е. практически то же самое, что и с кодом справочника.

         *Периодичность* – обычно нумерация документов идет не сквозная, а в пределах какого-то временного интервала, чаще всего года. Для реализации этого в системе предусмотрен соответствующий механизм. Предусмотрены следующие варианты периодичности номеров:

  непериодический;

  в пределах года;

  в пределах квартала;

  в пределах месяца;

  в пределах дня.

Для документа Рас*ходДенег* изменим длину номера на 6 знаков и  периодичность на значение *В пределах года*.

Согласно поставленной задачи, 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.

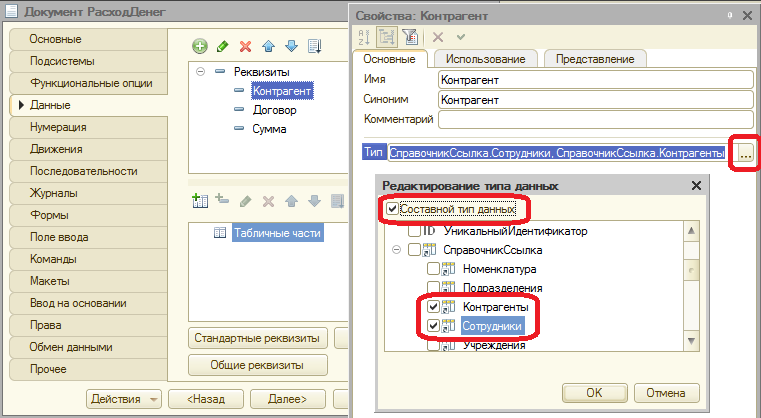


Рис. 8.3.

На закладке *Данные*, точно так же, как это делалось и для справочника, добавляем новые реквизиты. (Рис 8.3) 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*.

## 7.2. 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* (Рис. 8.4), а для работы с несколькими документами данного вида используется *Форма списка* (Рис. 8.5).

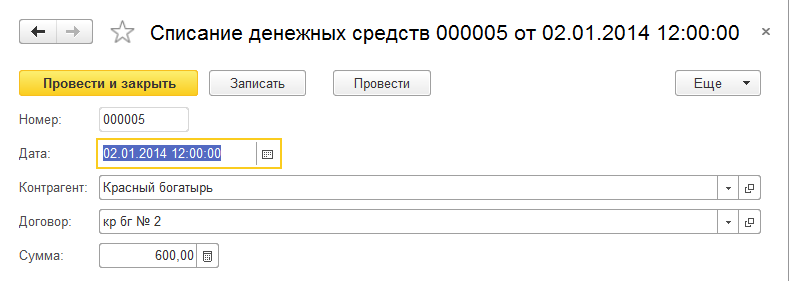


Рис. 8.4.

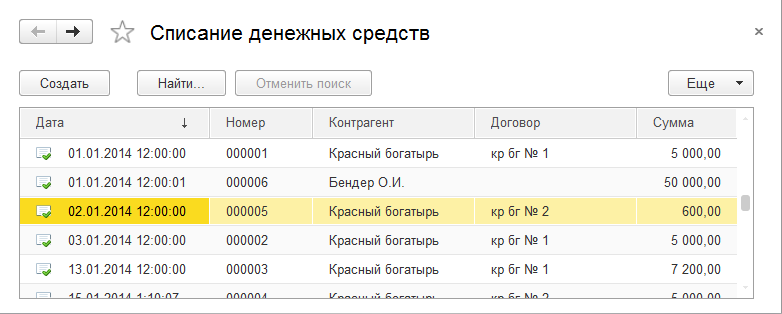


Рис. 8.5.

В том случае, когда автоматически создаваемая форма не устраивает, имеется возможность создать соответствующую форму в конфигураторе, используя ранее рассмотренный механизм работы с формами (Рис. 8.6). Однако некоторые настройки внешнего вида и поведения формы можно указать и не создавая саму форму.

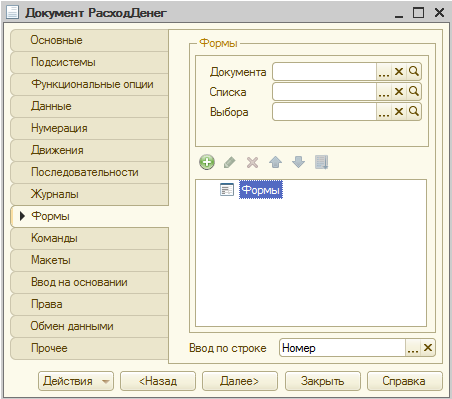


Рис. 8.6.

Для того, чтобы в форме документа осуществлялся подбор только тех договоров, которые относятся к выбранному контрагенту, необходимо сделать следующую настройку (Рис. 8.7) В свойствах реквизита *Договор* необходимо установить связь параметров выбора с *Контрагентом*, который должен являться владельцем для *Договора*.

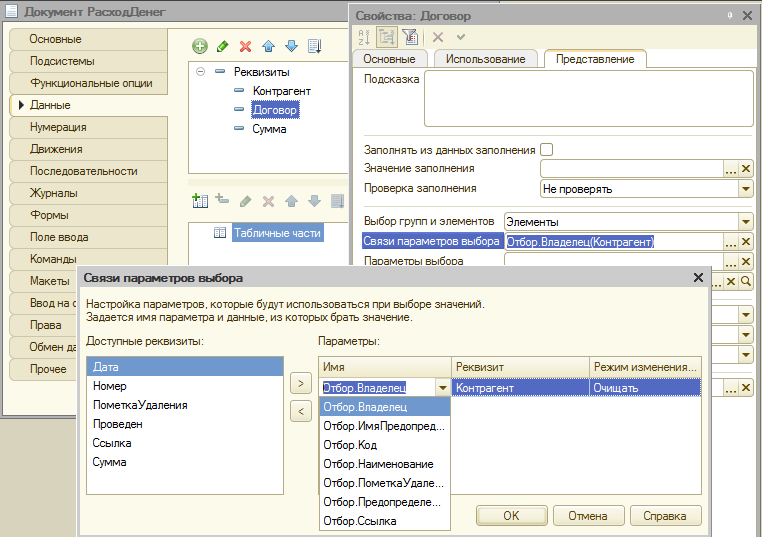


Рис. 8.7.

Широкие возможности по настройке списка предоставлены пользователю непосредственно в режиме "1С:Предприятие". Для этой цели в форме списка документа необходимо нажать кнопку *Все действия* и в открывшемся меню выбрать пункт *Настройка списка*. После этого, последовательно заполняя настройки по закладкам (Рис. 8.8 – Рис. 8.12) пользователь может добиться того, чтобы список отображалась наиболее удобным для него способом (Рис. 8.13).

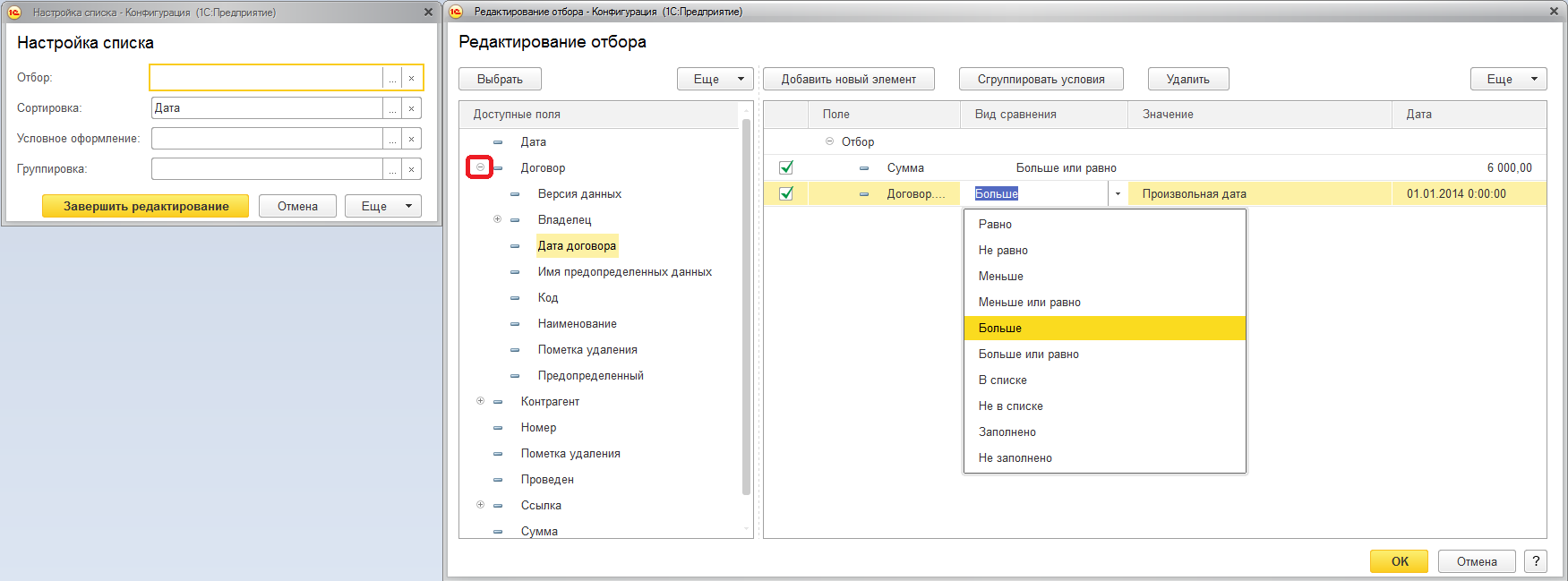


Рис. 8.8.

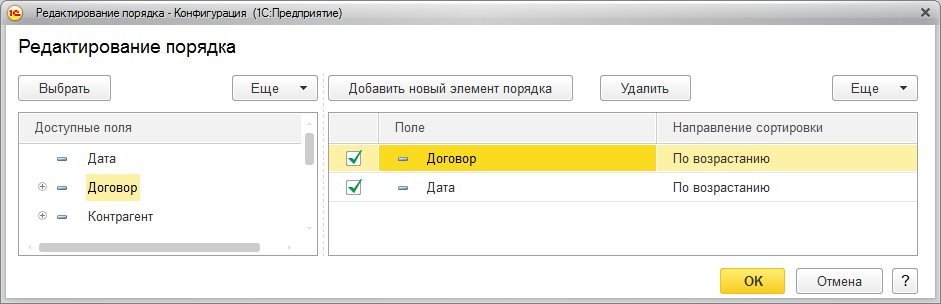


Рис. 8.8.

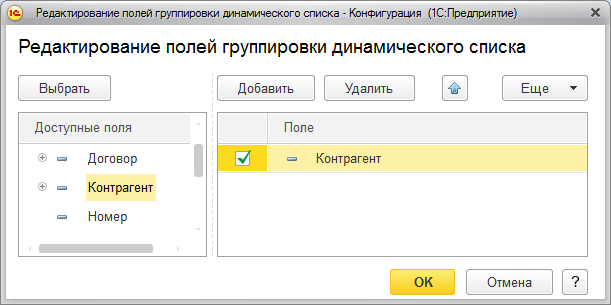


Рис. 8.9.

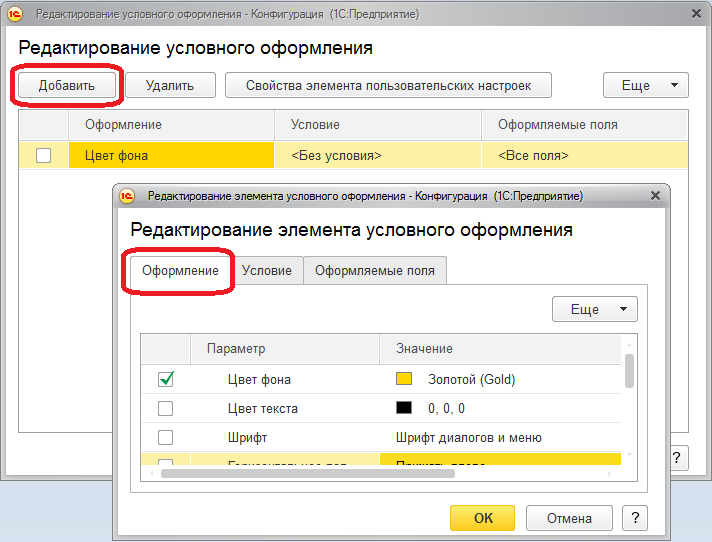


Рис. 8.10.

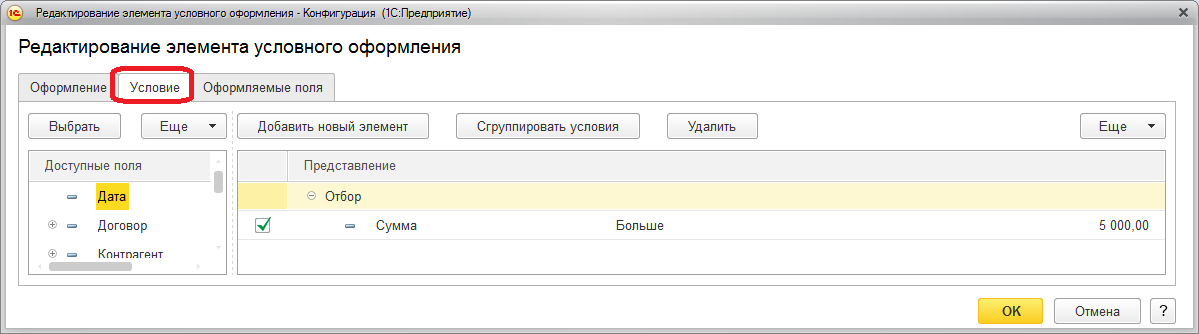


Рис. 8.11.

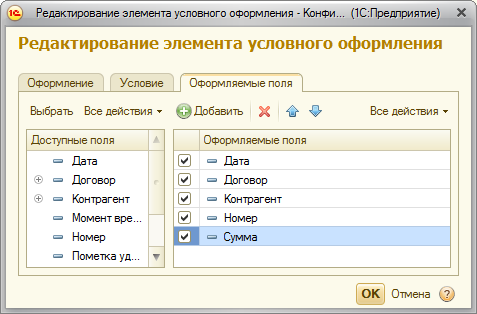


Рис. 8.12.

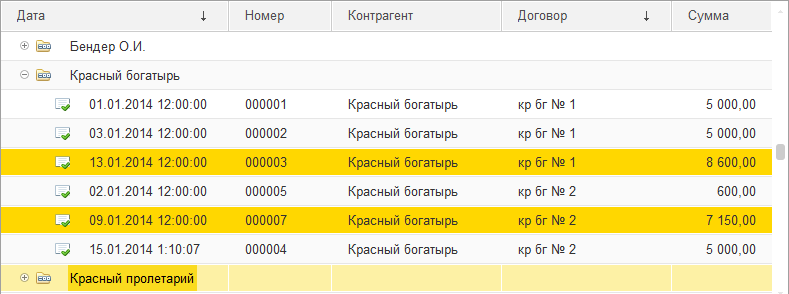
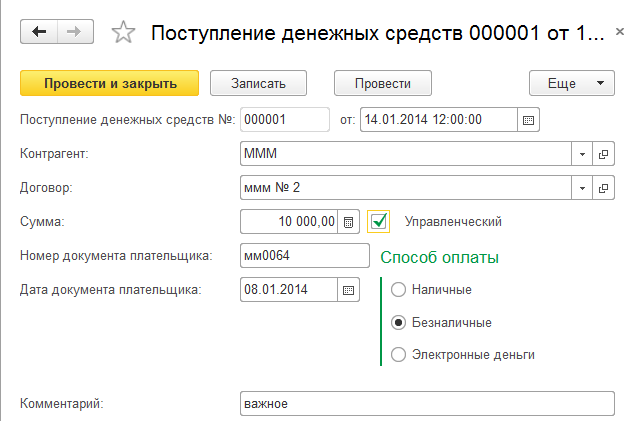


Рис. 8.13.

Следует отметить, что настройки списка автоматически сохраняются платформой для каждого пользователя.

***Самостоятельно***

***Создайте документ Поступление денежных средств (Рис. 8.14)***



**Рис. 8.14.**

|  |  |
| --- | --- |
|  |  |
|  |  |

Обратите внимание, что пользователь имеет возможность (в определенных пределах)  изменить внешний вид формы самостоятельно (Рис. 8.15 и Рис. 8.16). Для этого необходимо выбрать пункт меню *Изменить форму* из выпадающего списка, сформированного при нажатии на кнопку *"Еще"*.

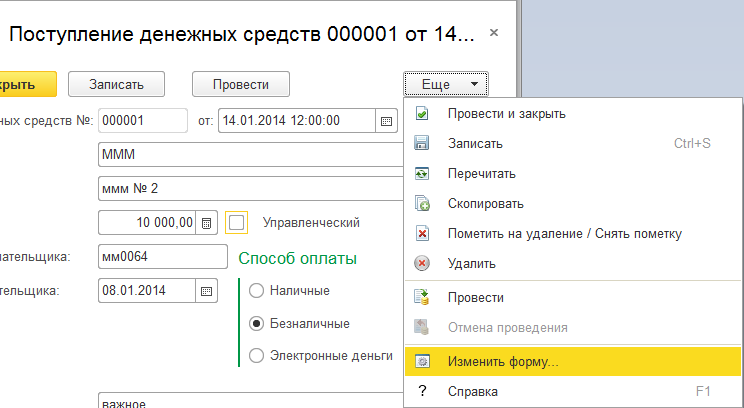


Рис. 8.15.

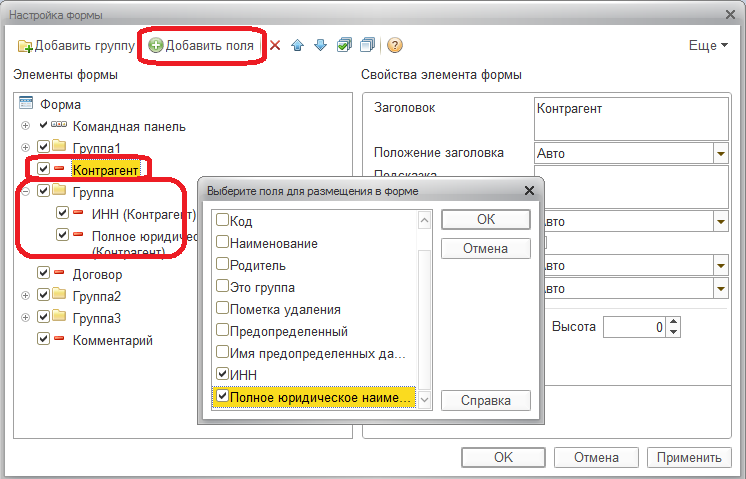


Рис. 8.16.

Редактор формы в пользовательском режиме работает по тем же принципам, что мы  рассматривали для работы в режиме конфигуратора – группы, положение, заголовки и т.д. Причем для реквизитов ссылочного типа пользователь может дополнительно вывести на форму информацию (Рис.8.17)

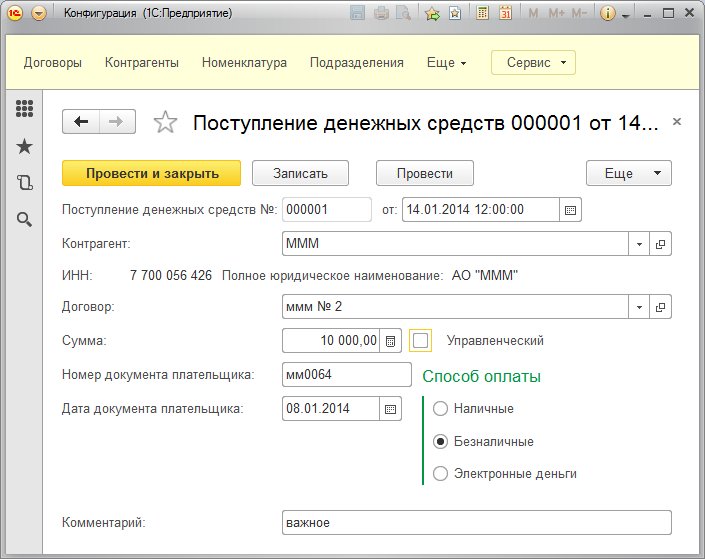


Рис. 8.17.

Естественно, что иногда пользователь может сделать некорректные настройки и испортить форму. В таком случае форма может быть легко возвращена к виду, заданному в конфигураторе. Для этого по кнопке "Еще" необходимо выбрать пункт меню "Установить стандартные настройки"  (Рис.8.18)

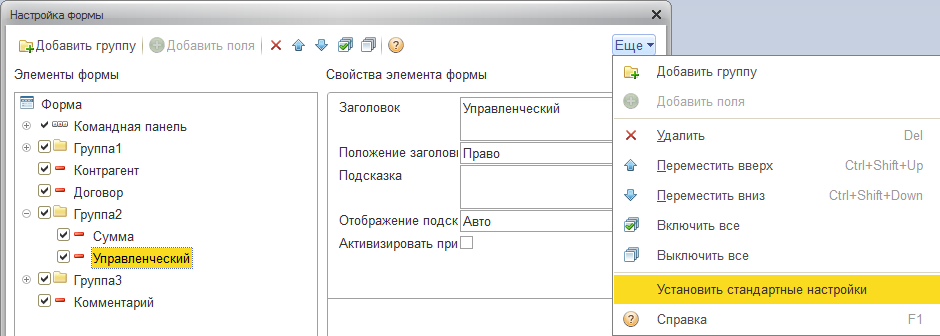


Рис. 8.18.

До сих пор все формы, которые были рассмотрены, имели статичный внешний вид, который был задан в конфигураторе, однако внешний вид формы может изменяться в зависимости от ситуации.

Так как реквизиты плательщика имеют смысл только в случае безналичной оплаты, то сделаем так, чтобы они были не видны во всех остальных случаях. Для того, чтобы удобнее было прятать реквизиты, сгруппируем их (Рис.8.19)

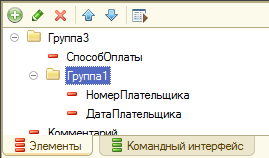


Рис. 8.19.

Чтобы реквизиты исчезали, когда меняется способ оплаты, задействуем обработчик события *ПриИзменении* для способа оплаты (Рис. 8.20)

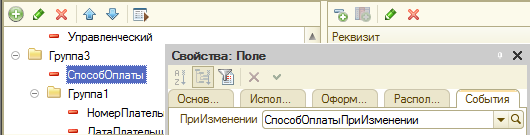


Рис. 8.20.

Кроме того, необходимо использовать обработчик события *ПриОткрытии* для всей формы (Рис.8.21)

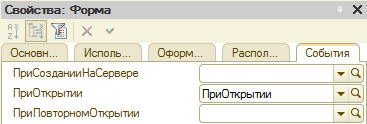


Рис. 8.21.

Текст процедур обработчиков событий в модуле формы документа *ПриходДенег* (данный программный код может быть оптимизирован, что будет гораздо менее удобно для восприятия, поэтому рассмотрим упрощенный вариант):

&НаКлиенте

Процедура СпособОплатыПриИзменении(Элемент)

            Элементы.Группа1.Видимость= НоваяВидимость (Объект.СпособОплаты);

КонецПроцедуры

&НаСервере

Функция НоваяВидимость (Способ)

            Если Способ = Перечисления.СпособыОплаты.Безналичные Тогда

                        Воз=Истина;

            Иначе

                        Воз=Ложь;

            КонецЕсли;

            Возврат Воз;

КонецФункции

&НаКлиенте

Процедура ПриОткрытии(Отказ)

            Элементы.Группа1.Видимость=НоваяВидимость(Объект.СпособОплаты);

КонецПроцедуры

В результате для пользователя в режиме "1С:Предприятие" внешний вид формы будет изменяться динамически, в зависимости от действий пользователя (Рис.8.22).

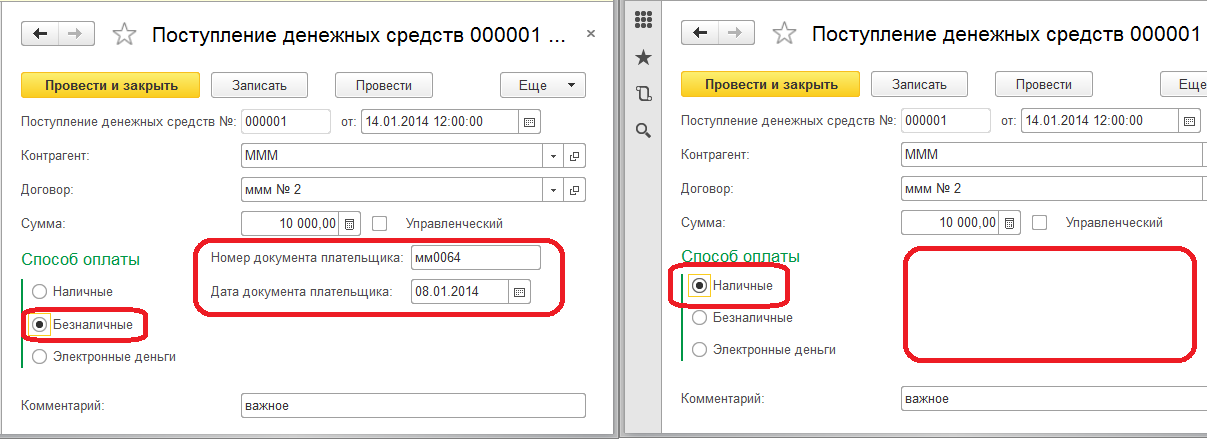


Рис. 8.22.

## 7.3. Реквизиты табличной части

Для того чтобы хранить не одно значение, а произвольное количество однотипных значений, используем табличную часть документа. Для каждого документа может быть определено произвольное количество табличных частей, каждая из которых может содержать произвольное количество реквизитов.

В нашем случае для документа *ПриходнаяНакладная* 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. 8.21).

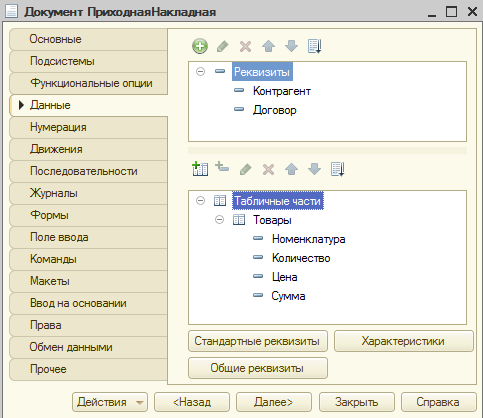


Рис. 8.21.

Для документа *Приходная накладная* форма будет отличаться от формы документа *Приход денег* и *Расход денег* наличием табличной части. При создании накладной также необходимо обратить внимание на связь элементов *Договор* и *Контрагент*, а также производить автоматический расчет сумм.

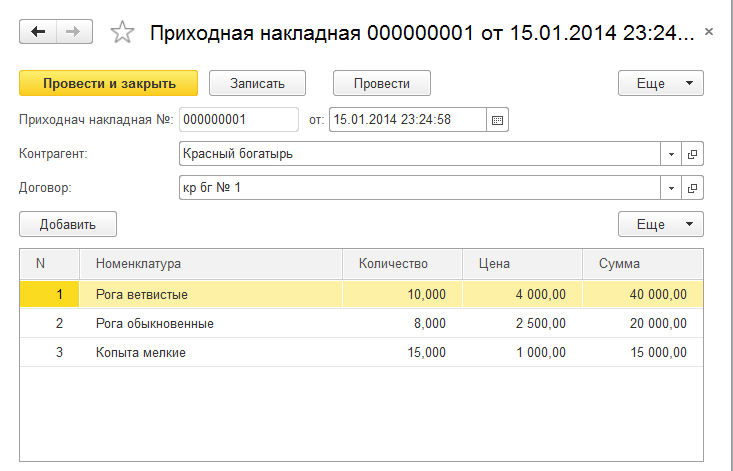


Рис. 8.22.

Для расчета суммы необходимо задействовать обработчики событий *При изменении* для количества и цены (Рис. 8.22). 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

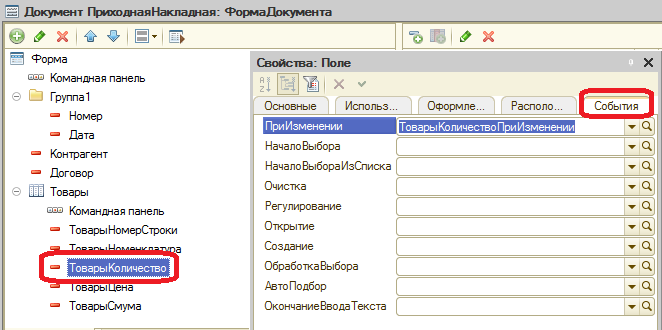


Рис. 8.23.

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. 8.24)

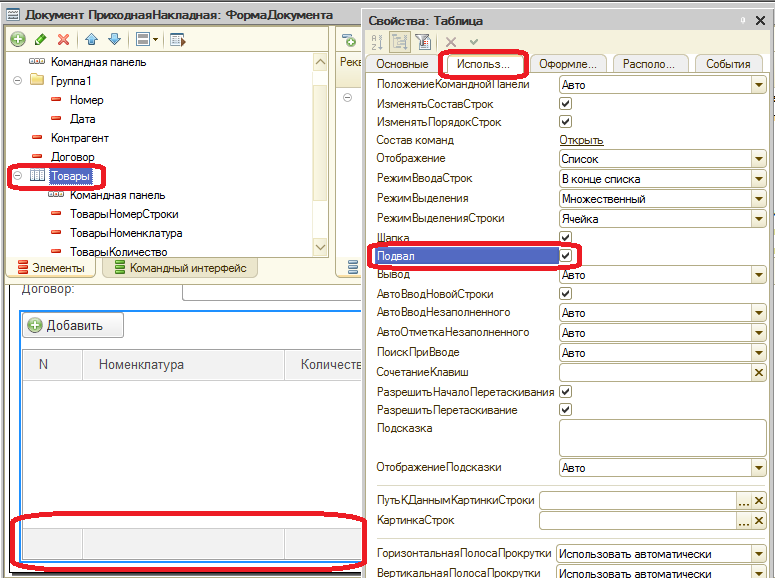


Рис. 8.24.

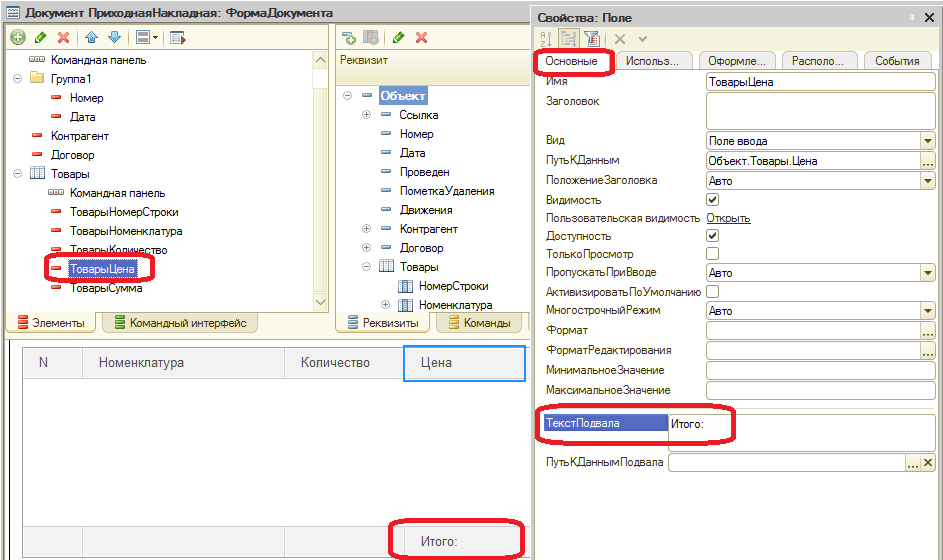


Рис. 8.25.

В подвале можно отображать как заранее известный и фиксированный текст (Рис. 8.25), так и автоматически вычисляемые платформой данные. В нашем случае это будет итог по колонке сумма (Рис. 8.26).

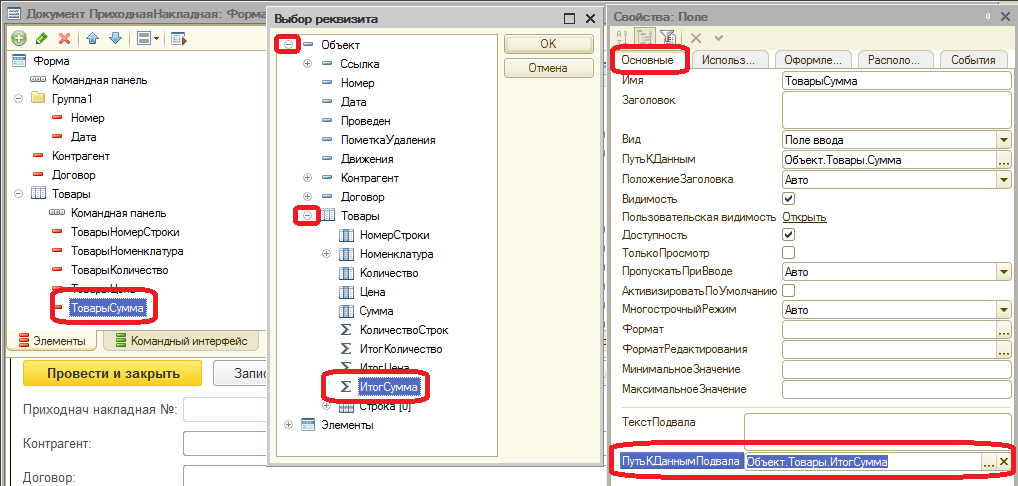


Рис. 8.26.

В результате пользователь получит следующую форму (Рис. 8.27) для работы с документом приходная накладная.

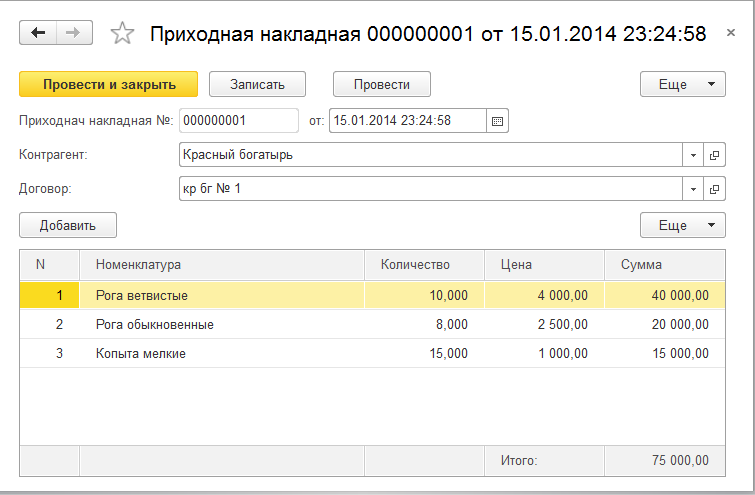


Рис. 8.27.

**СТОП ДЛЯ ЗАОЧНИКОВ!**