

Т. С. Балданова, О. А. Лобсанова

**ВВЕДЕНИЕ В 1С
ПРЕДПРИЯТИЕ 8**

Учебно-методическое пособие

Улан-Удэ
2019

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
БУРЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ ДОРЖИ БАНЗАРОВА

Т.С. Балданова, О.А. Лобсанова

ВВЕДЕНИЕ В 1С: ПРЕДПРИЯТИЕ 8

*Рекомендовано учебно-методическим советом БГУ
в качестве учебно-методического пособия для студентов,
обучающихся по направлениям подготовки
02.03.03 Математическое обеспечение и администрирование
информационных систем, 09.03.03 Прикладная информатика*

Улан-Удэ
Издательство Бурятского государственного университета
2019

УДК 004:658 (075.8)
ББК 32.8:65.29 я 73
3 715

Утверждено к печати
редакционно-издательским советом
Бурятского госуниверситета

Рецензенты

С.В. Архипов,

кандидат технических наук, директор ЦИТ и ДО,
доцент кафедры «Информационные технологии», БГУ

И.Б. Елтунова,

кандидат педагогических наук, доцент, начальник отдела
дистанционного и дополнительного профессионального
образования, СибГУТИ

Текст печатается в авторской редакции

Балданова Т.С.

Б 153 Введение в 1С: Предприятие 8 : учебно-методическое
пособие / Т.С. Балданова, О.А. Лобсанова. — Улан-Удэ: Изд-во
Бурятского госуниверситета, 2019. — 149 с.
ISBN 978-5-9793-1427-3

В пособии даются начальные представления о работе с основными объектами и механизмами платформы 1С, с целью сформировать у студентов практические навыки по работе в системе 1С Предприятие 8. Пособие предназначено для студентов направлений подготовки по математическому обеспечению и администрированию информационных системы прикладной информатике по дисциплине «Основы конфигурирования в 1С: Предприятие 8». Пособие состоит из отдельных глав. Каждая глава посвящена отдельной теме, которая рассматривается на конкретных примерах.

УДК 004:658 (075.8)
ББК 32.8:65.29 я 73 я

ISBN 978-5-9793-1427-3

©Т. С. Балданова, О. А. Лобсанова, 2019
©Бурятский госуниверситет
им. Д. Банзарова, 2019

Предисловие

1С: Предприятие – это специализированная объектно-ориентированная система управления базами данных (СУБД). 1С: Предприятие – одна из ведущих программ на рынке программного обеспечения в России, включающая в себя самые передовые технологии автоматизации разнообразных учетных задач, таких как кадровый учет на предприятии, фирме или организации, расчет заработной платы сотрудников, бухгалтерский учет, складской учет и другие задачи современной фирмы.

Помимо того, что программа 1С: Предприятие содержит большой набор функций и инструментов по обслуживанию финансовой, торговой, организационной деятельности фирмы (предприятия или др.), данная программа также предоставляет разработчику возможность создавать и внедрять новые прикладные решения, благодаря встроенному языку программирования, различным визуальным редакторам и конструкторам.

Таким образом, изучение основ конфигурирования в 1С: Предприятии студентами информационно – технических вузов и информационно – технических направлений позволит им овладеть профессиональными компетенциями по соответствующим направлениям и в дальнейшем обеспечит возможность их эффективного трудоустройства.

Цель пособия – 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.

Учебное пособие «Введение в 1С: Предприятие 8» предназначено для студентов направлений 02.03.03. «Математическое обеспечение и администрирование информационных систем» и 09.03.03. «Прикладная информатика» по дисциплине «Основы конфигурирования в 1С: Предприятие 8» и способствует формированию общепрофессиональных компетенций, установленные программой бакалавриата, в частности:

- способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом требований

информационной безопасности (ОПК-4) по направлению подготовки 09.03.03. «Прикладная информатика», профиль подготовки «Проектирование и внедрение информационных систем.

- способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности (ОПК-1) по направлению подготовки 02.03.03. «Математическое обеспечение и администрирование информационных систем», профиль подготовки «Информационные системы и базы данных»

Данное пособие состоит из отдельных глав. Каждая глава посвящена отдельной теме, которая рассматривается на конкретных примерах. В этих примерах изложена последовательность действий, выполняя которую слушатели шаг за шагом смогут освоить материал, изложенный в соответствующей главе.

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. 1.1). Назначение сотруднику оклада или перевод его в какое-либо подразделение отражаются документом "Кадровый приказ".

N	Учреждение	Дата поступле...	Дата оконча...	Специальность
1	1С: УЦ №1	03.05.2011	05.05.2011	Знакомство с платформой
2	МГУ	01.09.2002	30.06.2008	Бухгалтер

Рис. 1.1

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. 1.2), а его оплата документом "Поступление денежных средств" (Рис.1.3). Для документа "Расходная накладная" необходимо иметь возможность сформировать печатную форму.

Расходная накладная 000000003 от 19.01.2014 23:53:00

Провести и закрыть | Записать | Провести | Печать | Еще

Расходная накладная №: 000000003 от: 19.01.2014 23:53:00 Валюта: Доллар

Контрагент: MMM

Договор: ммм № 2

Сотрудник: Паниковский М.С. Управленческий

N	Товар	Количество	Цена	Ставка НДС	Всего
			Сумма	Сумма НДС	
1	Рога обыкновенные	5,000	3 000,00	18 %	17 700,00
			15 000,00	2 700,00	
2	Рога ветвистые	6,000	800,00	10 %	5 280,00
			48 000,00	4 800,00	
3	Рога обыкновенные	10,000	200,00	18 %	2 360,00
			20 000,00	3 600,00	
Итого товаров:					25 340,00
			83 000,00	11 100,00	

Рис. 1.2

Поступление денежных средств 000001 от 14.01.2014 12:00:00

Провести и закрыть | Записать | Провести | Еще

Поступление денежных средств №: 000001 от: 14.01.2014 12:00:00

Контрагент: MMM

Договор: ммм № 2

Сумма: 10 000,00 Валюта: Доллар Управленческий

Номер документа плательщика: мм0064 **Способ оплаты**

Дата документа плательщика: 08.01.2014

Наличные

Безналичные

Электронные деньги

Комментарий: важное

Рис. 1.3

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. 1.4).

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

Рис. 1.4

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 "C:\\" создадим каталог "\Знакомство с платформой", после чего запускаем систему "1С:Предприятие". На экран будет выведен диалог "Запуск 1С:Предприятия" (Рис.1.5).

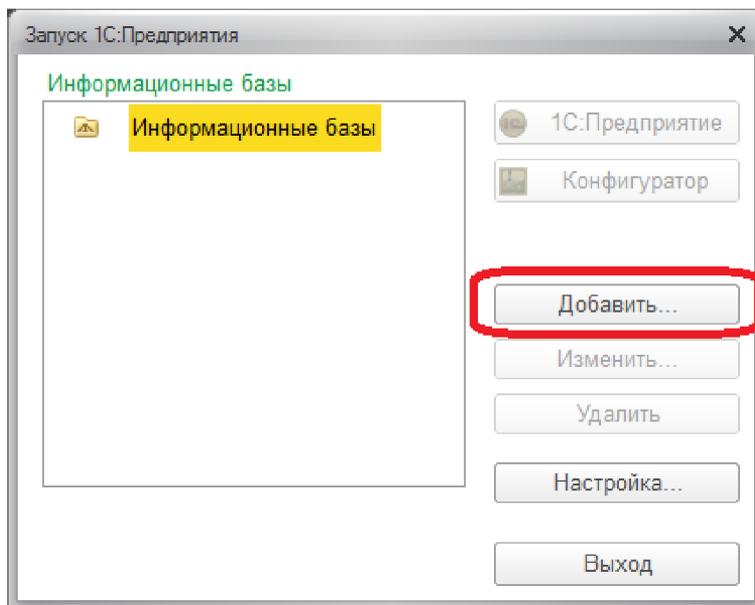


Рис. 1.5

Создадим новую информационную базу, нажав кнопку "Добавить". Далее следуем инструкциям, появляющимся на экране. Поскольку создаем новую базу, в первом окне оставляем верхний вариант (Рис. 1.6).

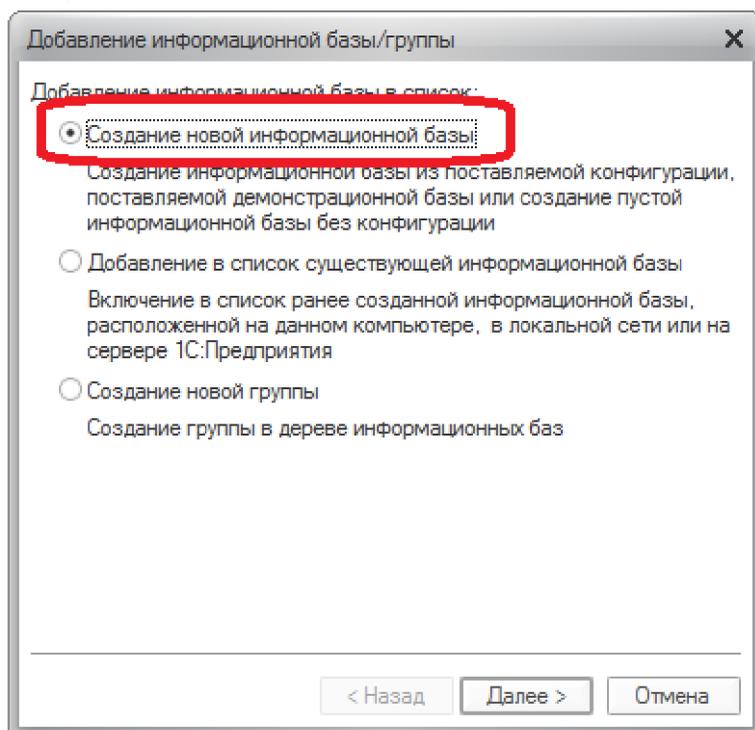


Рис. 1.6

Далее необходимо указать, что создается новая пустая база, а не типовая из шаблона, для чего выбираем нижний вариант (Рис 1.7).

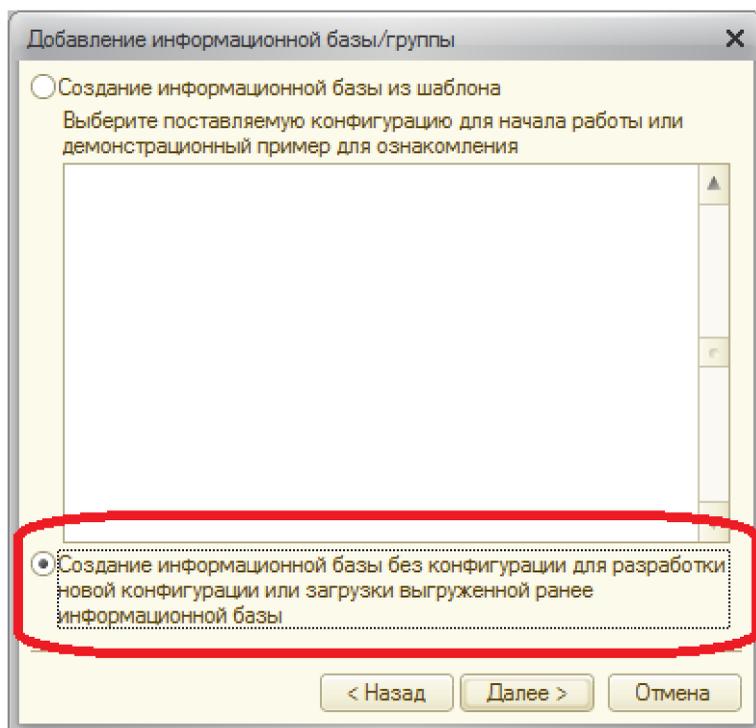


Рис. 1.7

Следующее, что необходимо сделать, это дать имя создаваемой базе. В нашем случае назовем базу "Знакомство с платформой" (Рис. 1.8).

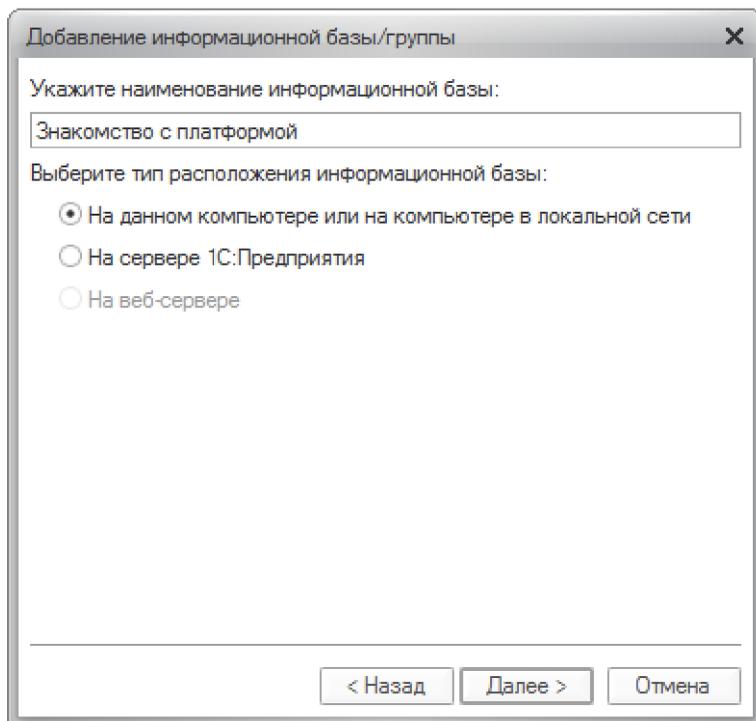


Рис. 1.8

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. 1.9).

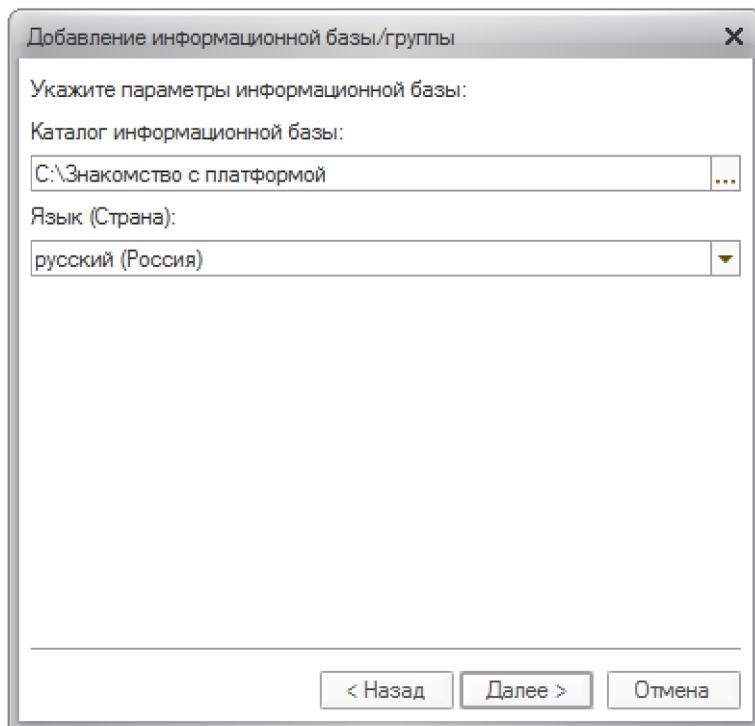


Рис. 1.9

В следующем окне оставляем параметры запуска по умолчанию (Рис. 1.10) и на этом создание новой информационной базы будет завершено.

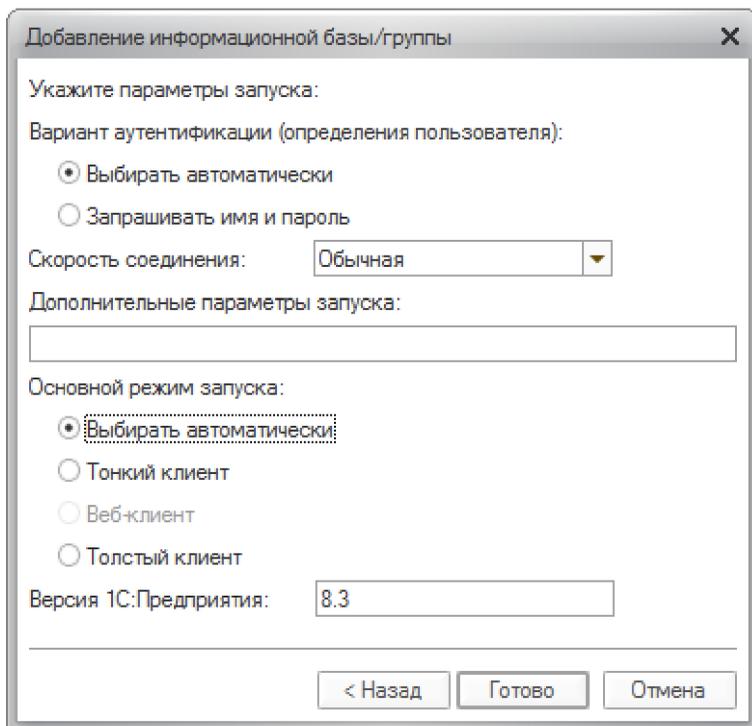


Рис. 1.10

Теперь после запуска "1С:Предприятия" у нас появится возможность работать с новой информационной базой (Рис. 1.11), причем в обоих режимах, выбор которых будет осуществляться нажатием на соответствующие кнопки "1С:Предприятие" или "Конфигуратор".

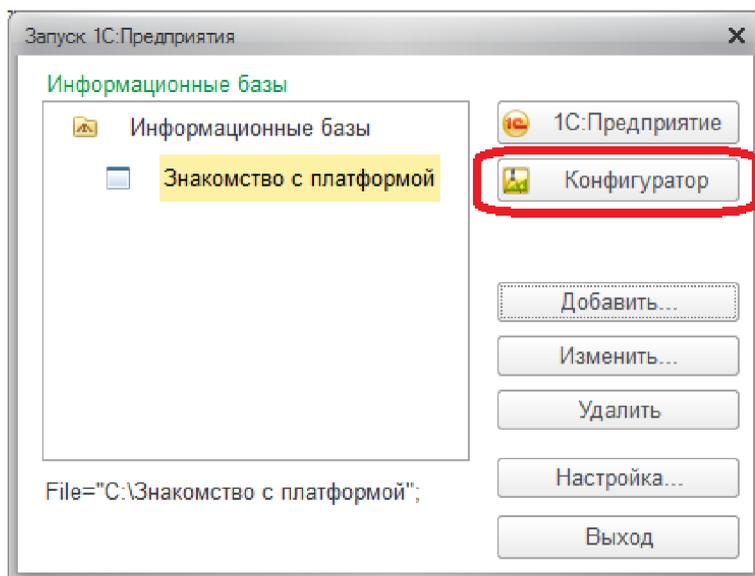


Рис. 1.11

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 configurator. В открывшемся окне откроем окно "Конфигурация". Для этого выберем пункт меню *Конфигурация-Открыть конфигурацию* (Рис. 1.12).

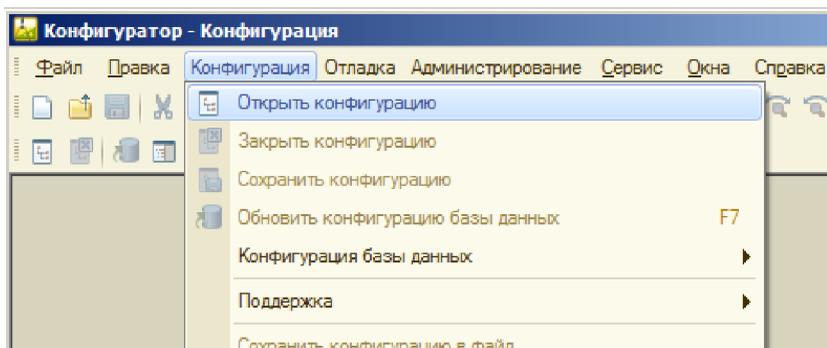


Рис. 1.12

Открывается новое окно (Рис. 1.13):

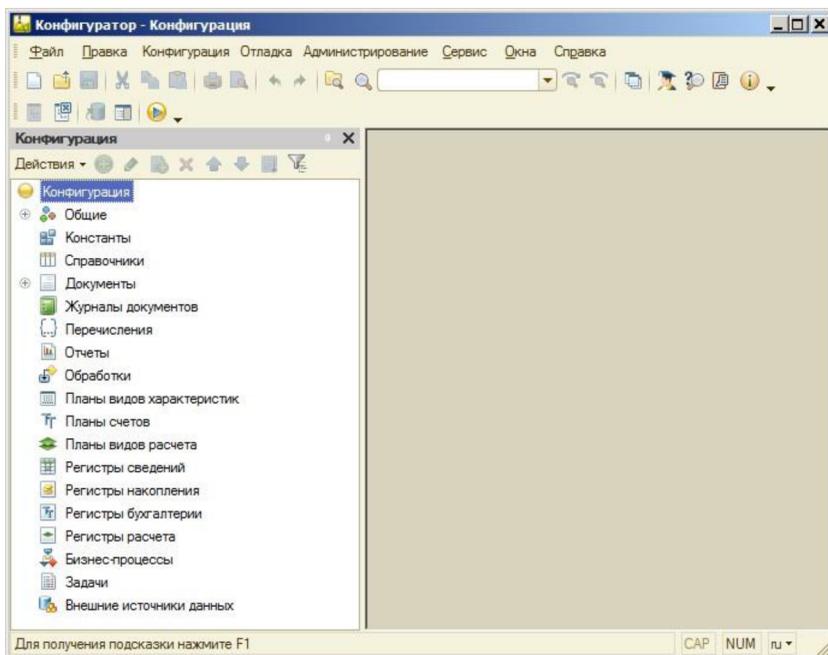


Рис. 1.13

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 2.1)

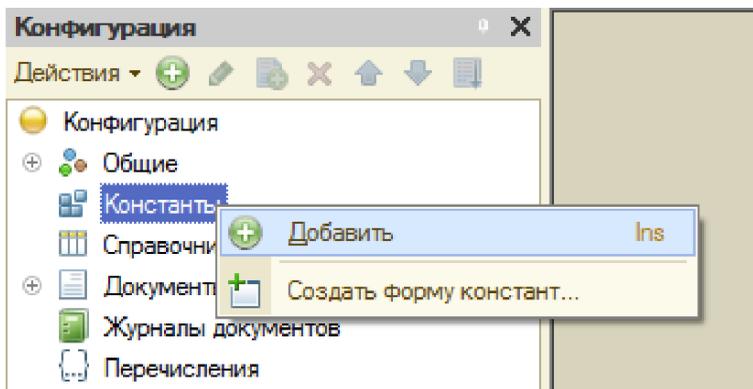


Рис. 2.1

После того, как мы выберем данный пункт, на экране откроется новое окно, называемое Палитра свойств, в которой будут отражены свойства константы. (Рис 2.2). Свойства можно сворачивать по группам, для удобства работы.

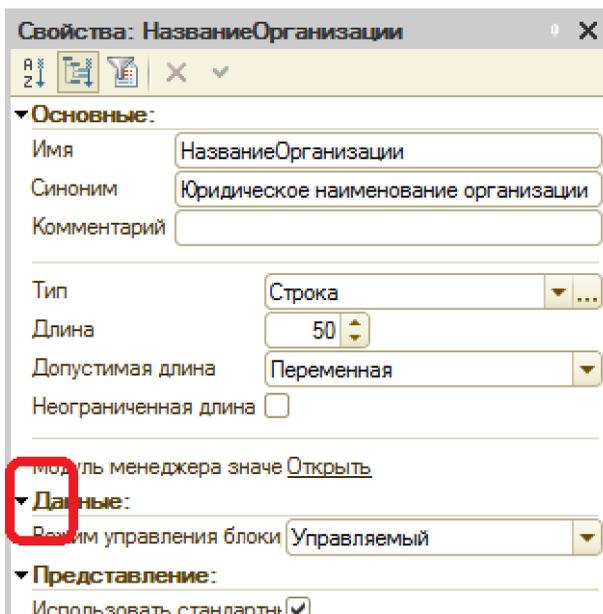


Рис. 2.2

Внешний вид палитры свойств можно изменить через контекстное меню по нажатию на правую кнопку мыши (Рис. 2.3).

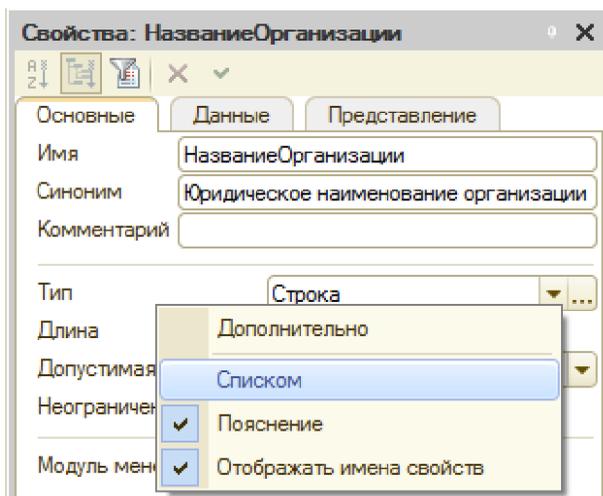


Рис. 2.3

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 мыши. (Рис 2.4)

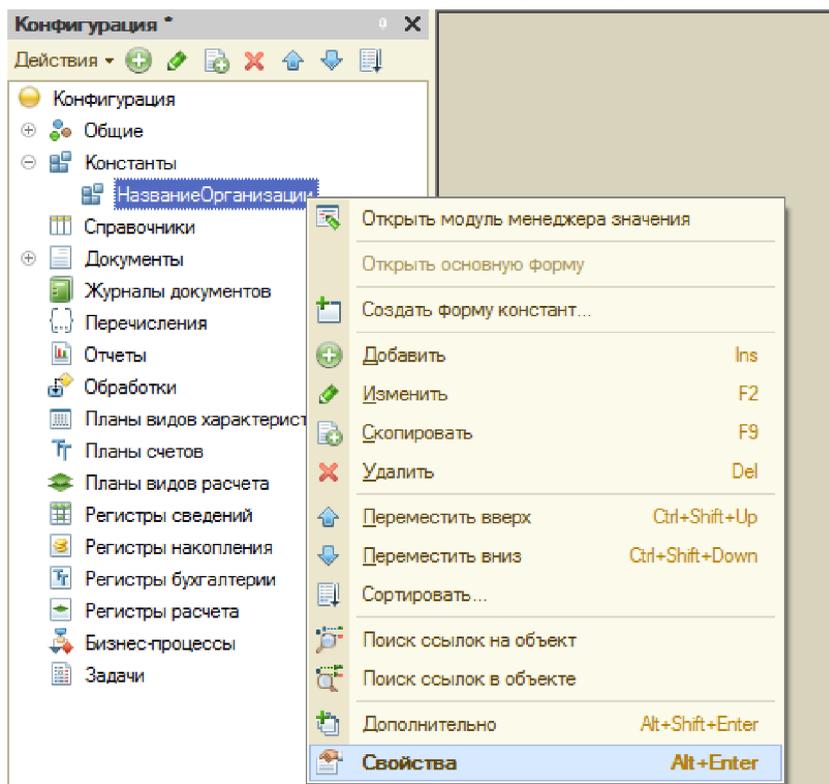


Рис. 2.4

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. 2.5) и последовательно подтверждаем необходимость обновления информационной базы (Рис. 2.6 - Рис. 2.8).

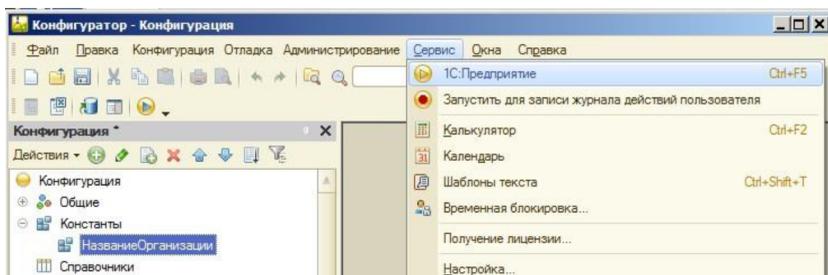


Рис. 2.5

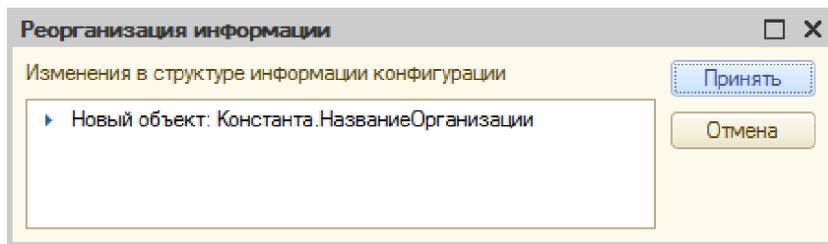


Рис. 2.6

Только после выполнения всех этих действий, в пользовательском режиме можно ввести необходимую информацию (Рис. 2.6 и Рис. 2.7).

ИС:Предприятие и последовательно подтверждаем необходимость обновления информационной базы (Рис. 2.4 и Рис. 2.5).

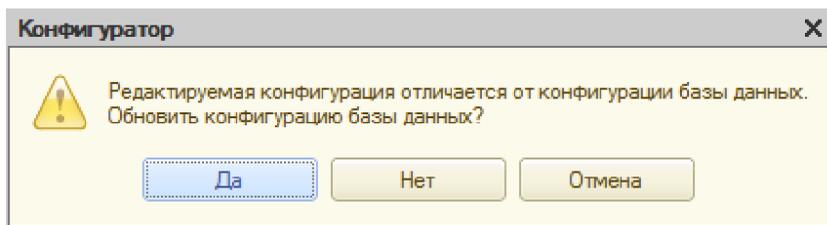


Рис. 2.7

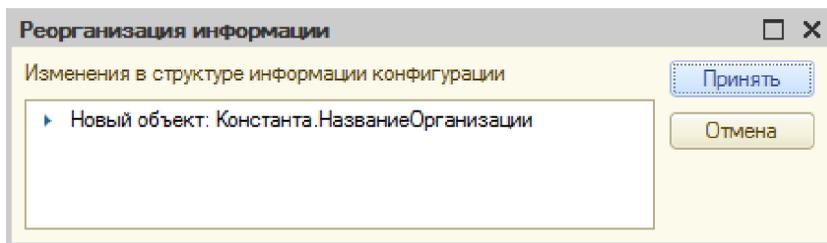


Рис. 2.8

Только после выполнения всех этих действий, в пользовательском режиме можно ввести необходимую информацию (Рис. 2.9 и Рис. 2.10).

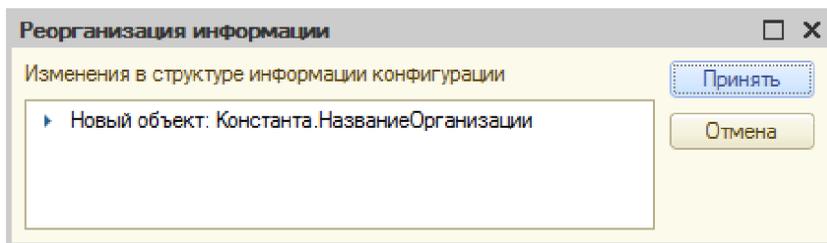


Рис. 2.9

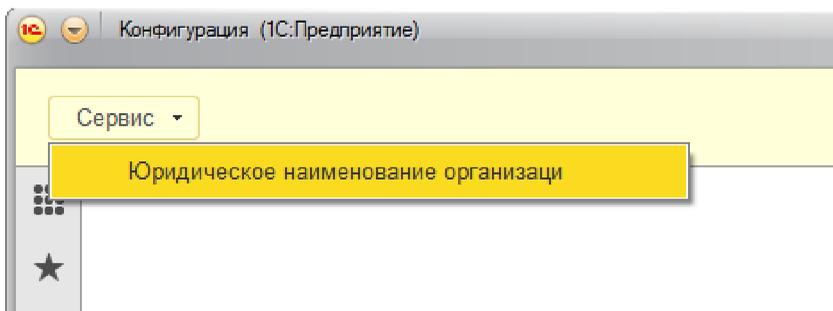


Рис. 2.10

В дальнейшем, если нет необходимости вносить изменения в структуру данных, запуск информационной базы производится стандартным образом (Рис. 1.11).

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 2.11).

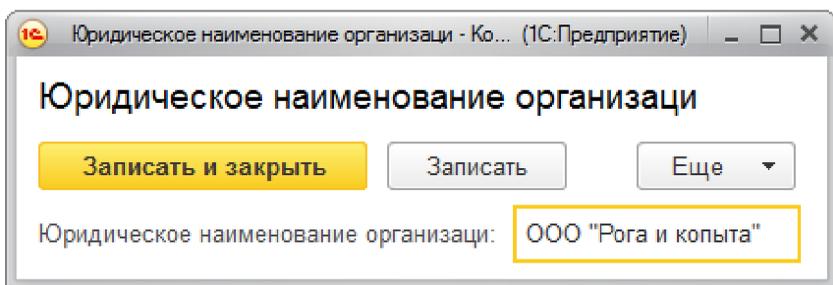


Рис. 2.11

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. 2.12).

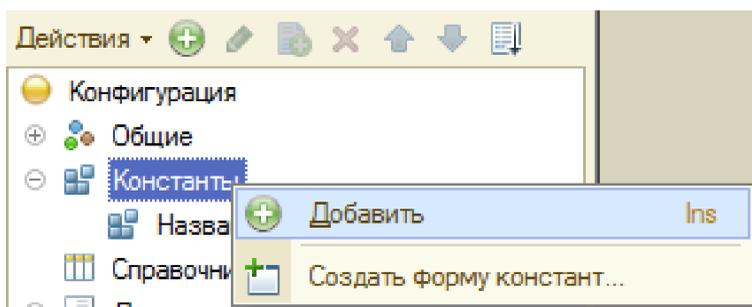


Рис. 2.12

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. 2.13):

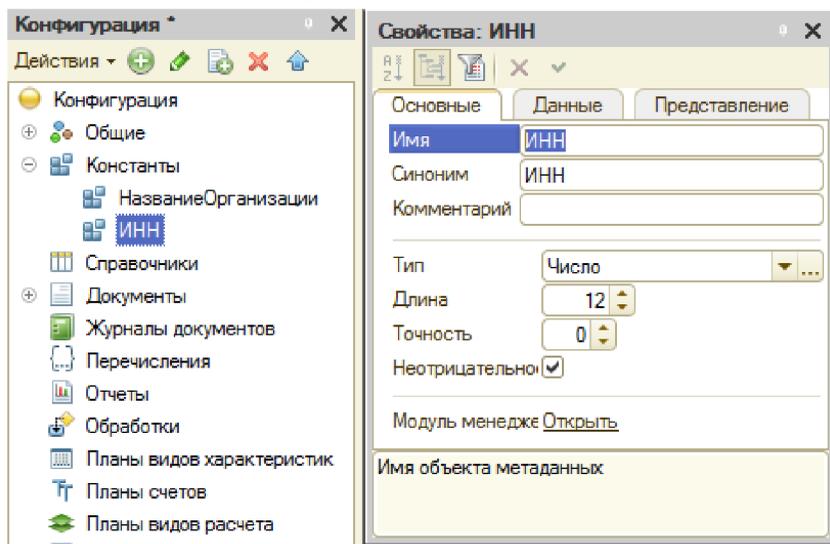


Рис. 2.13

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. 2.14).

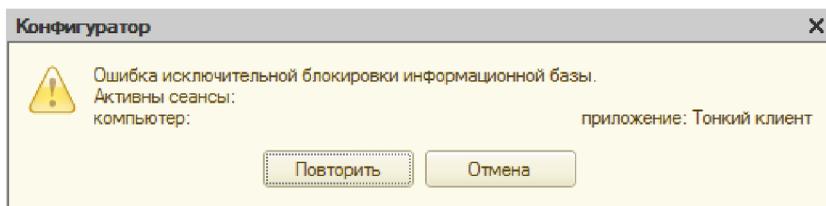
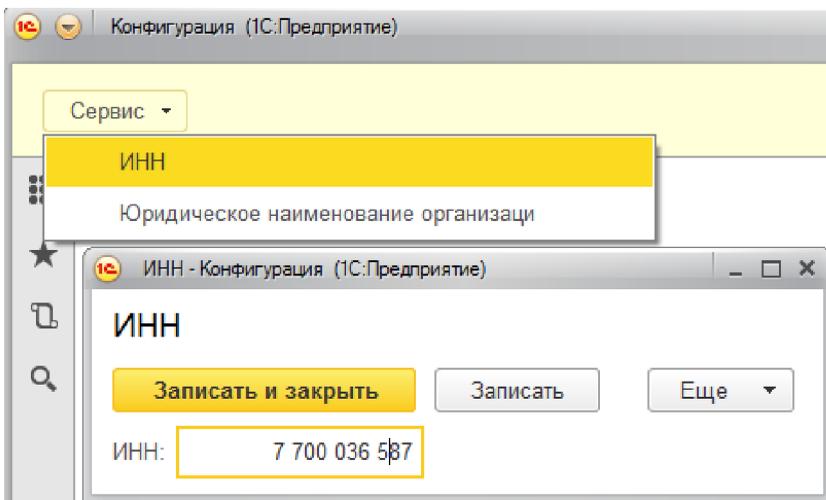


Рис. 2.14

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 Генеральный Директор 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. 3.1).

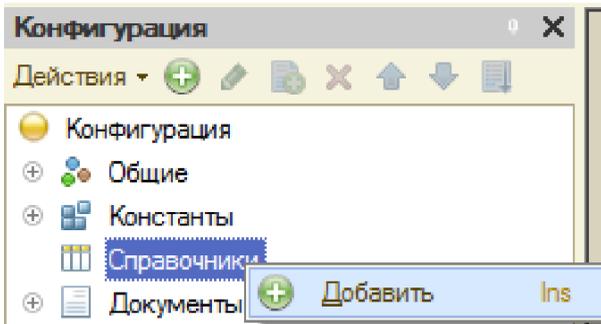


Рис. 3.1

Все эти действия одинаковые и приведут к появлению на

экране окна редактирования (Рис. 3.2) 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".

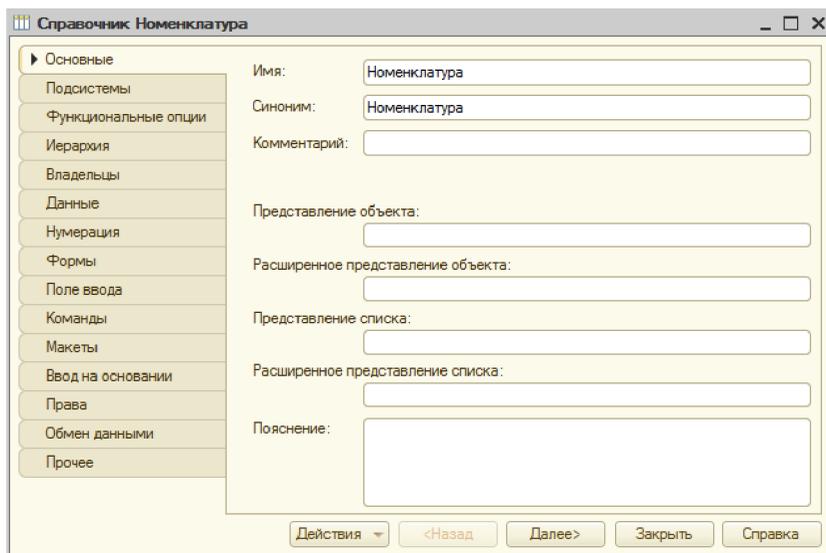


Рис. 3.2

На этом создание простейшего справочника завершено. Переходим в пользовательский режим предприятия и смотрим на то, что мы только что создали (Рис. 3.3). Все необходимые действия при работе со справочником можно прои

звестипутемнажатиянакнопку"Еще"ивыборасоответствующего пункта в выпадающем меню. Наиболее востребованные действия доступны через командную панель или контекстное меню, вызываемое правой кнопкой мышки.

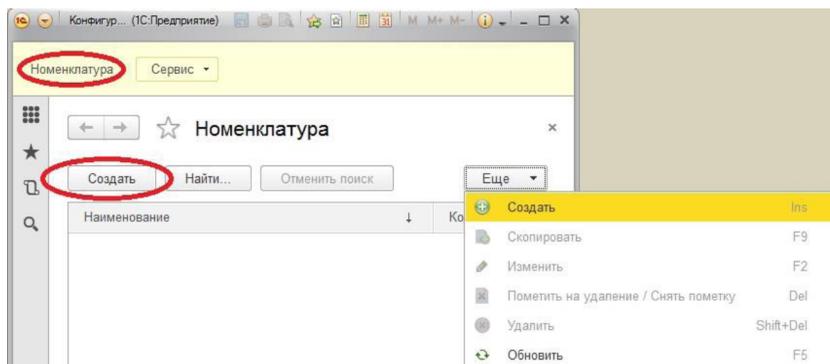


Рис. 3.3

Для добавления информации о товаре нажмем на кнопку "Создать" и в открывшемся окне укажем наименование нашего товара (Рис. 3.4):

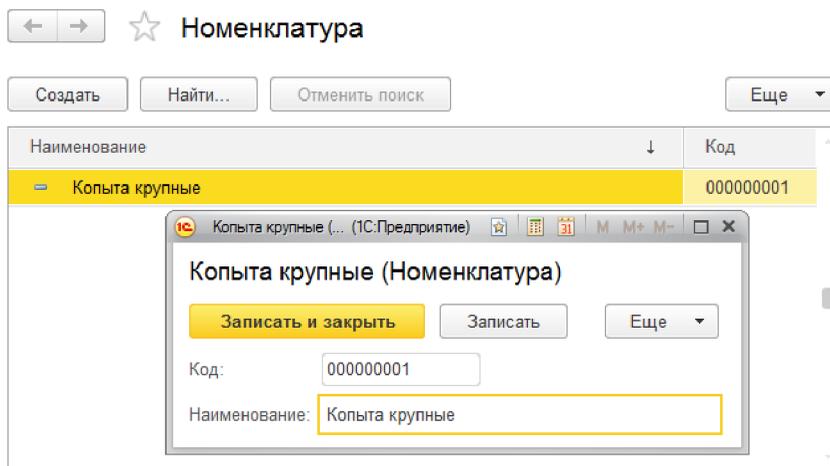


Рис. 3.4

Заполним справочник следующими данными (Рис. 3.5):

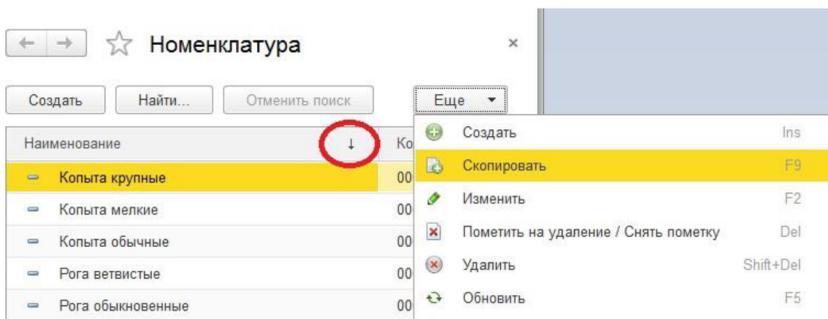


Рис. 3.5

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 наименовании сочетание букв "обы" (Рис. 3.6).

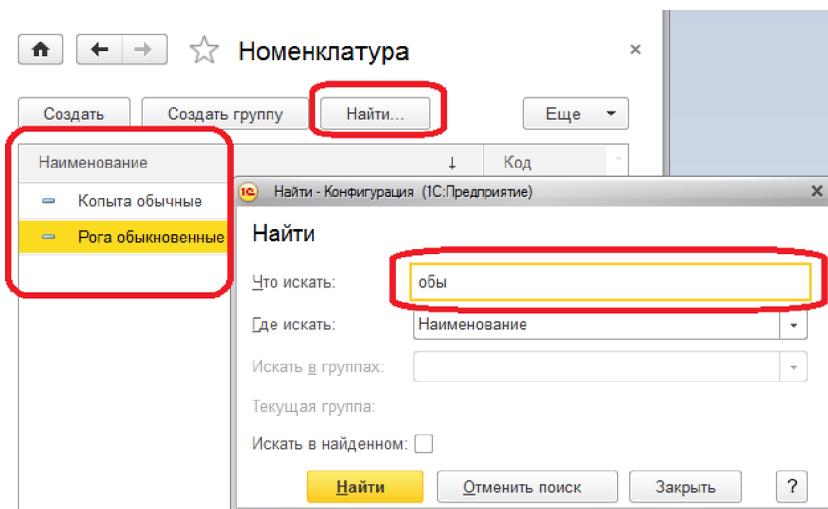


Рис. 3.6

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 конфигулятора 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 3.7) 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*.

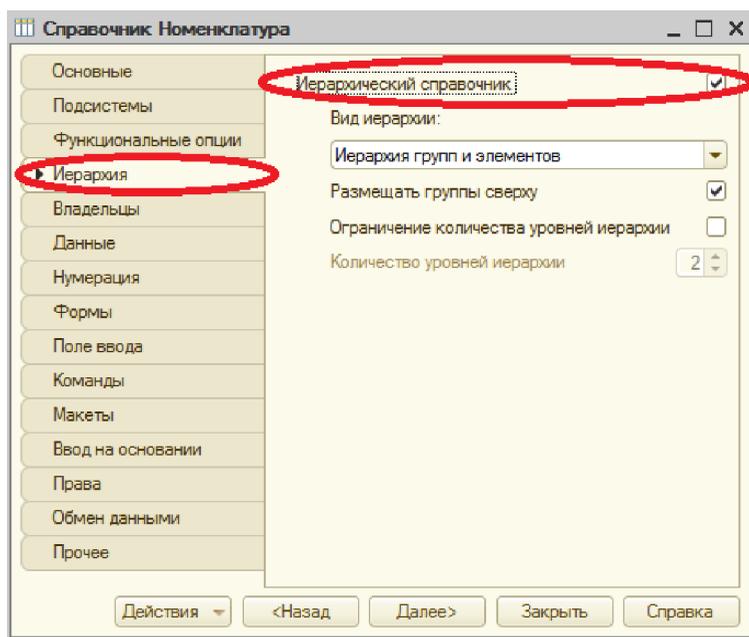


Рис. 3.7

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. 3.8).

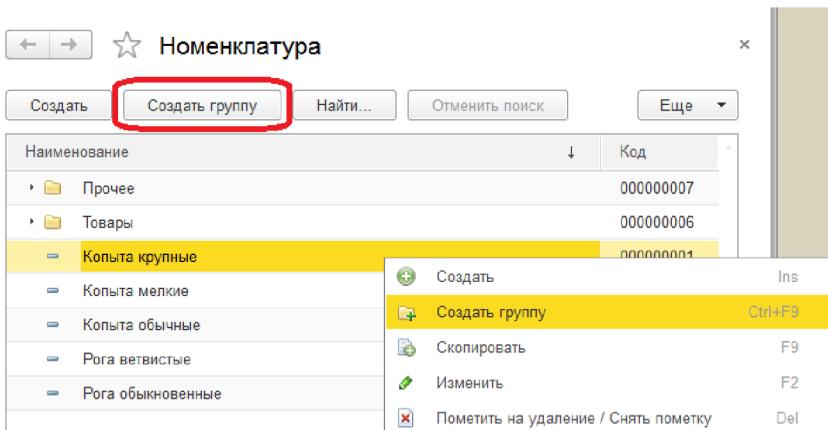


Рис. 3.8

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.

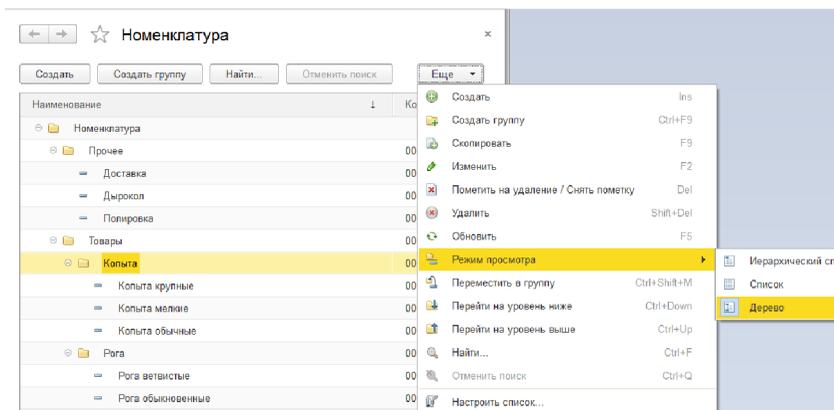


Рис. 3.9

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. 3.9).

Самостоятельно

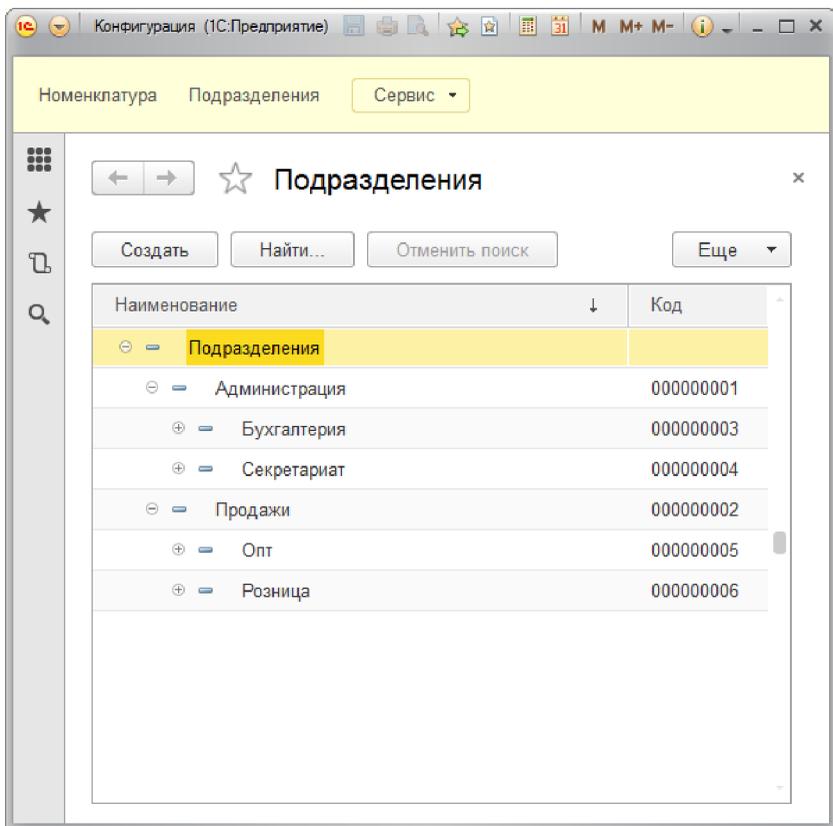


Рис. 3.10

Создайте новый справочник Подразделения для хранения данных и подразделения организации. Справочник должен иметь иерархию элементов (Рис. 3.10).

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.3.11), либо по кнопке "Предопределенные" на закладке *Прочее* вызываем окно для редактирования предопределенных элементов справочника (Рис.3.12).

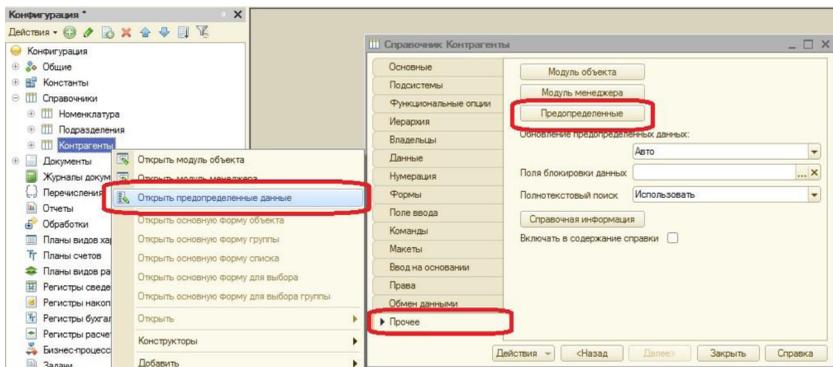


Рис. 3.11

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".

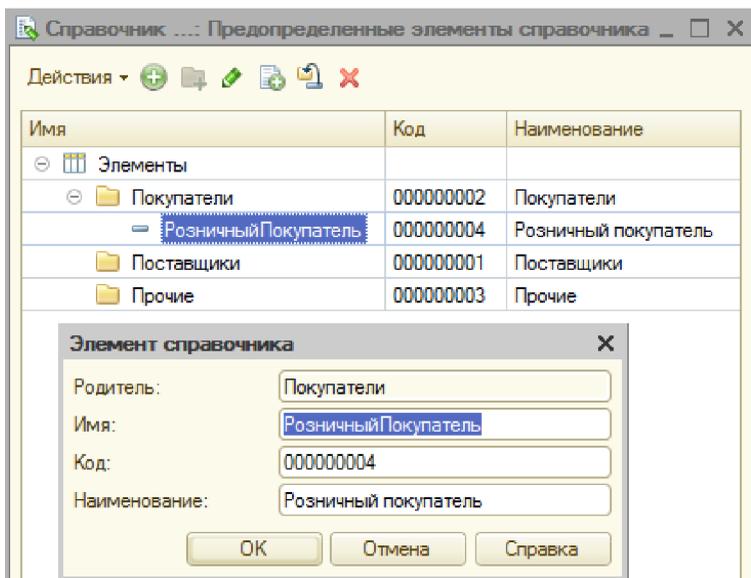


Рис. 3.12

В режиме "1С:Предприятие" для пользователя будет уже готовая иерархия. Заполним справочник данными (Рис. 3.13).

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

Рис. 3.13

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 3.14 и Рис 3.15):

- *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).

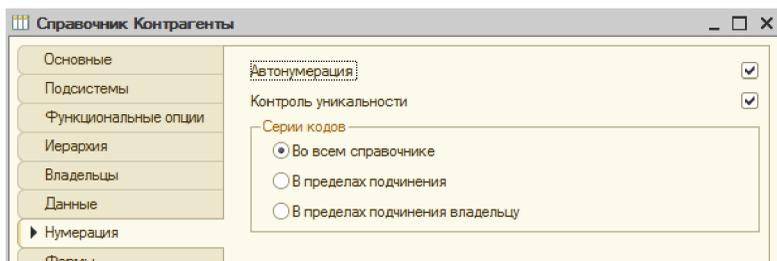


Рис. 3.14

– *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.

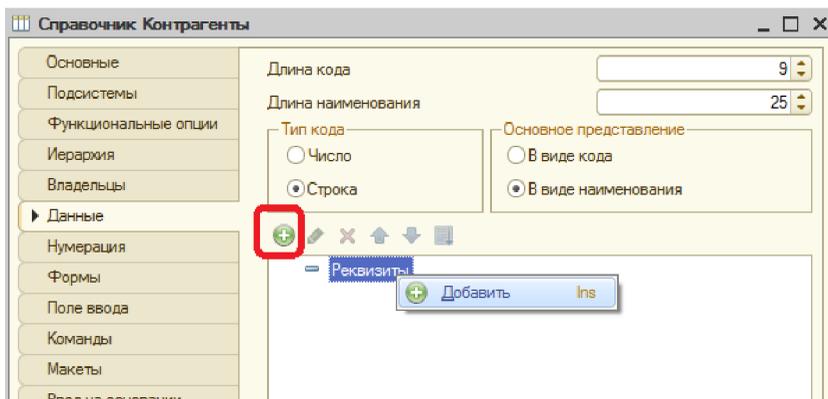


Рис. 3.15

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 3.15). Тот же результат можно достичь, выбрав пункт *Добавить* в панели инструментов этого поля.

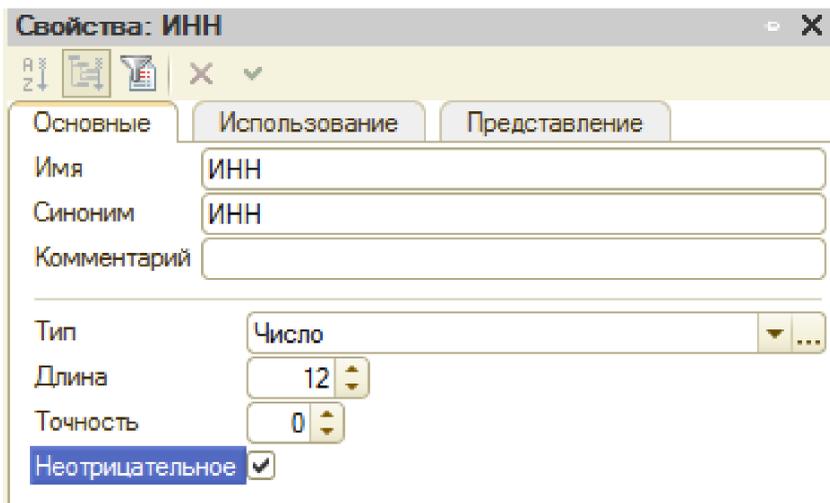


Рис. 3.16

Далее необходимо определить свойства нашего реквизита в соответствующем окне. (Рис. 3.16) 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.

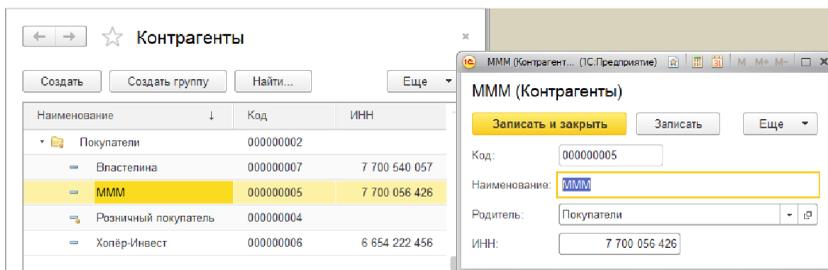


Рис. 3.17

После обновления базы для пользователя появляется возможность у каждого контрагента указать ИНН (Рис. 3.17).

Спомощьюреквизитовможетбытьрешенаиследующаязадача: необходимохранитьнаименованиенашегоконтрагента,ноэтонаименованиеочень 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. 3.18).

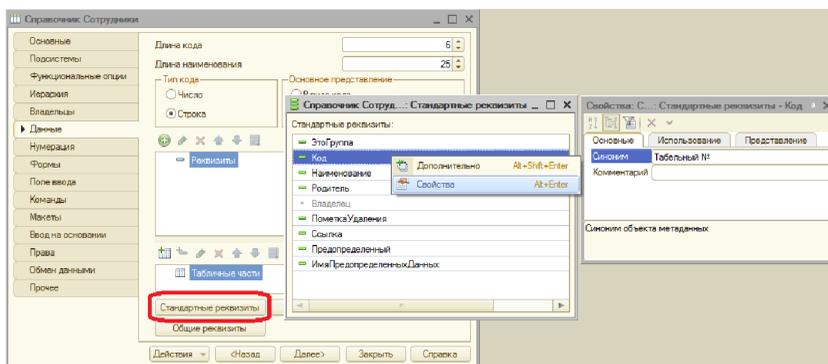


Рис. 3.18

Теперь добавим реквизит *Подразделение*, с типом данных *СправочникСсылка.Подразделения*. (Рис. 3.19)

В этом случае, вновь созданный реквизит будет заполняться значениями из справочника *Подразделения* (Рис. 3.20), 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*.

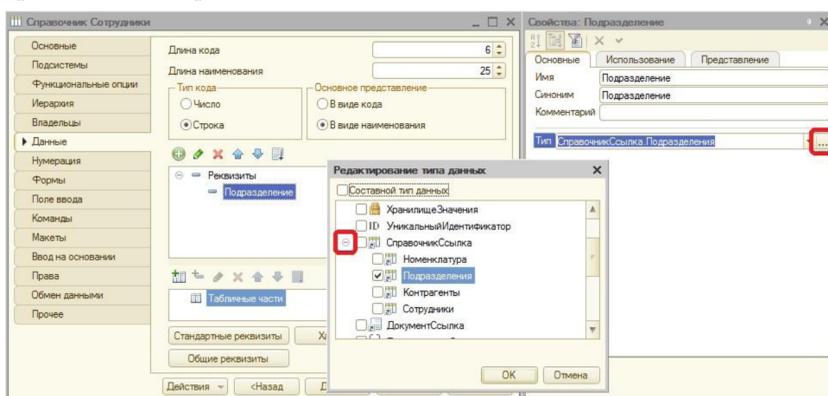


Рис. 3.19

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

Рис. 3.20

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 вносятся 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 вкладке *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. 3.21)

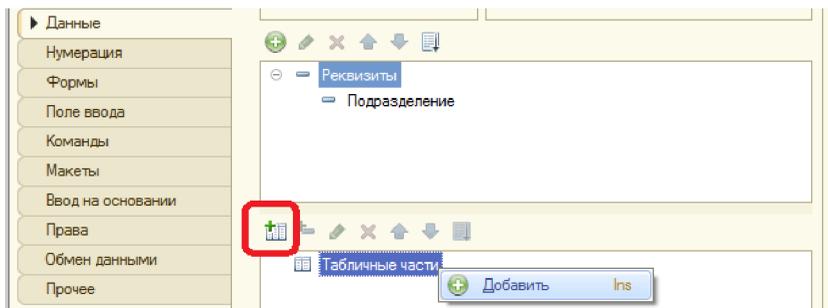


Рис. 3.21

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. 3.22)

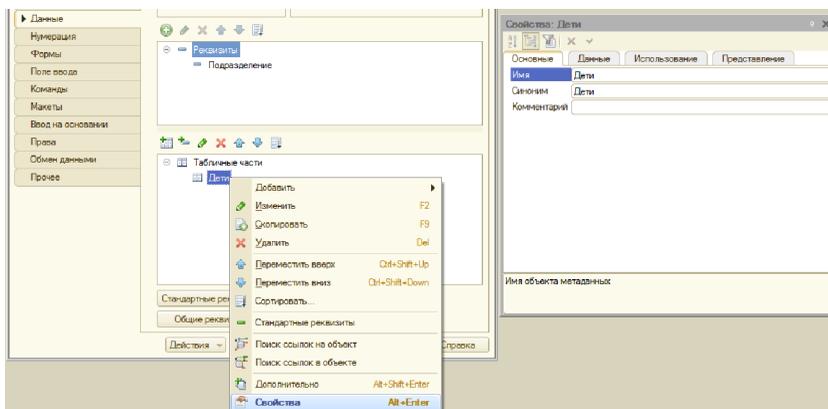


Рис. 3.22

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. 3.23)

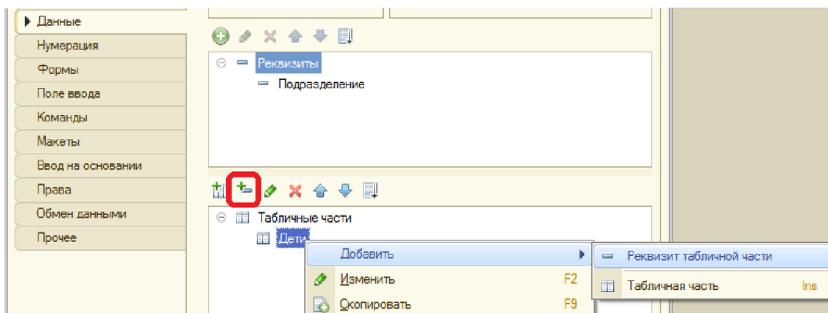


Рис. 3.23

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. 3.24).

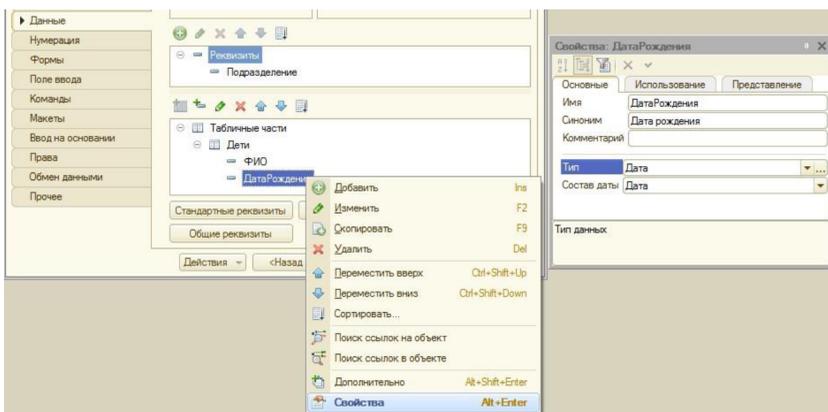


Рис. 3.24

В результате таких действий в пользовательском режиме справочник станет выглядеть так, как указано на Рис. 3.25

Балаганов А.П. (Сотрудники) (1С:Предприятие)

Балаганов А.П. (Сотрудники)

Табельный №:

ФИО:

Родитель:

Подразделение:

N	ФИО	Дата рождения
1	Маша	12.08.2005
2	Миша	17.11.2010

Рис. 3.25

Самостоятельно

Добавьте в справочнике Сотрудники табличную часть Образование (Рис. 3.26)

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. 3.26

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:Предприятие 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. 3.27).

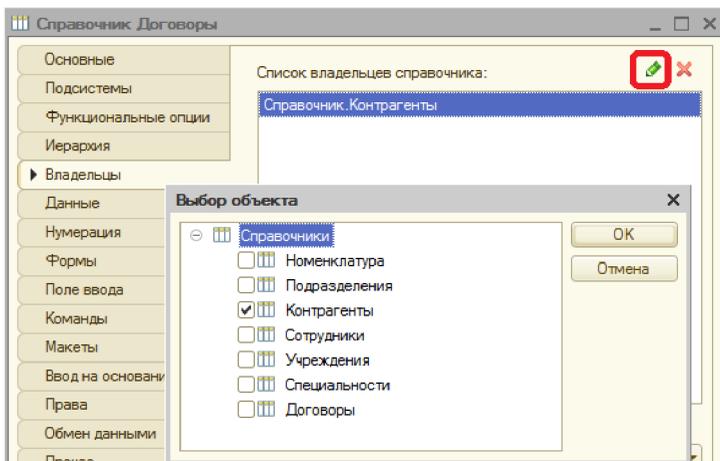


Рис. 3.27

В нашем случае справочник *Договор* 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. 3.28).

Красный богатырь (Контрагенты)

Главное

Договоры

Записать и закрыть

Записать

Еще ▾

Код:

000000009

Наименование:

Красный богатырь

Рис. 3.28

Возврат к данным контрагента происходит через ссылку "Главное", а работа с договорами происходит стандартным образом, также как с любым другим справочником (Рис. 3.29).

Красный богатырь (Контрагенты)

Главное

Договоры

Договоры

Создать

Найти...

Отменить поиск

Еще ▾

Наименование	↓	Код	Влад
— кр бг № 1		000000001	Крас
— кр бг № 2		000000002	Крас
— кр бг № 2/2		000000003	Крас

Рис. 3.29

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. 3.30).

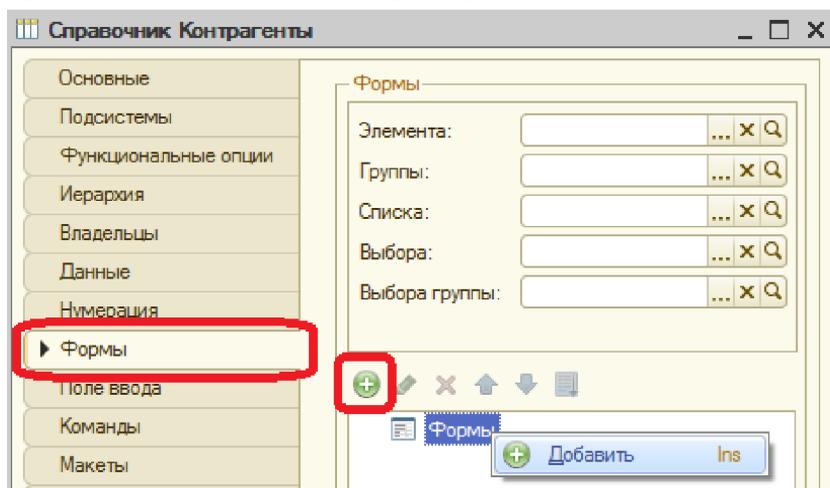


Рис. 3.30

Будем последовательно делать настройки, которые предлагает конструктор. На первой закладке укажем, что создаем форму списка (Рис. 3.31).

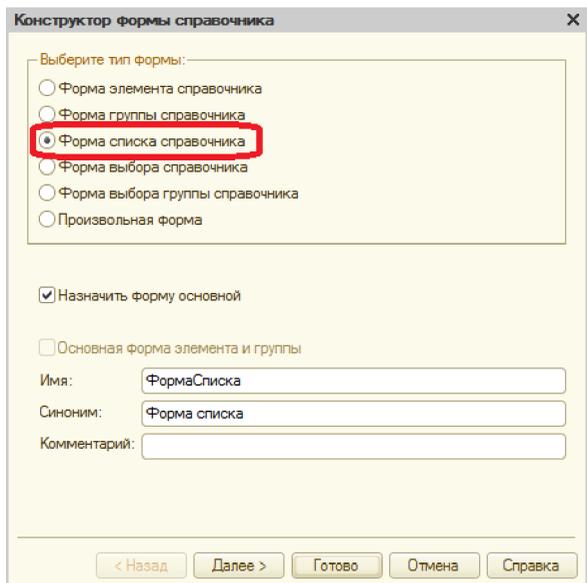


Рис. 3.31

Далее отметим те реквизиты справочника, которые мы хотим видеть (Рис. 3.32)

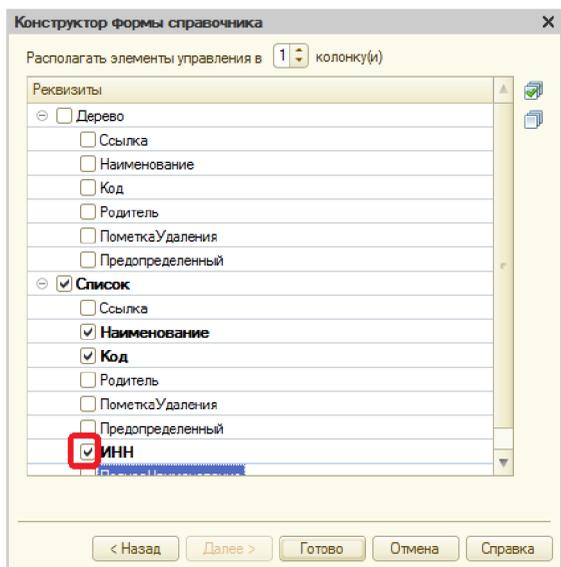


Рис. 3.32

В результате будет создана форма (Рис. 3.33)

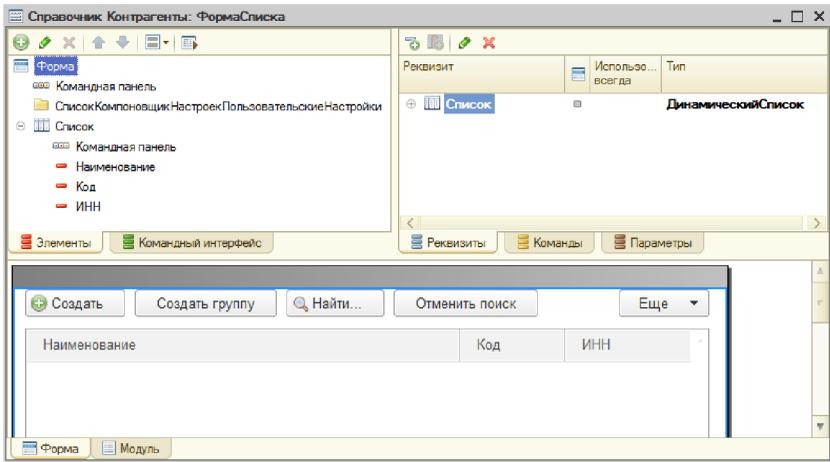


Рис. 3.33

Откроем закладку Команды – Глобальные команды и перетащим элемент "Элементы.Список.ТекущиеДанные.Ссылка" в левое окно в "Командная панель" (Рис. 3.34).

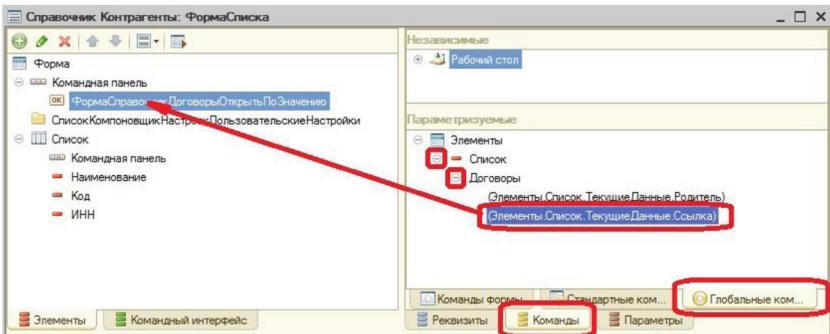


Рис. 3.34

В свойствах появившегося элемента укажем "ТолькоВоВсехДействиях" в значение "Нет" (Рис.3.35)

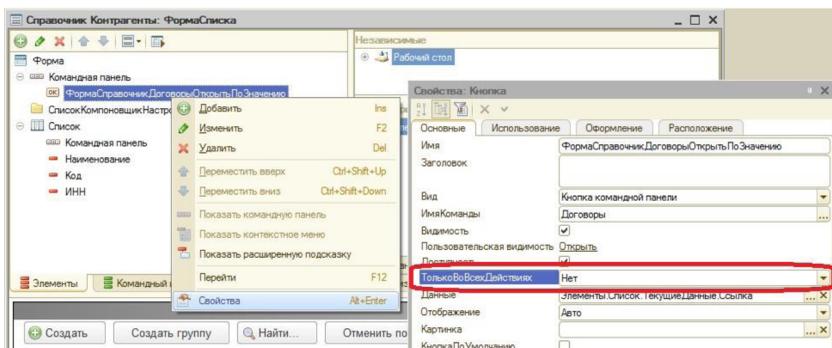


Рис. 3.35

Теперь в пользовательском режиме в форме списка справочника сотрудники и у пользователя появится возможность просматривать только те договоры, которые относятся к текущему контрагенту (Рис. 3.36).

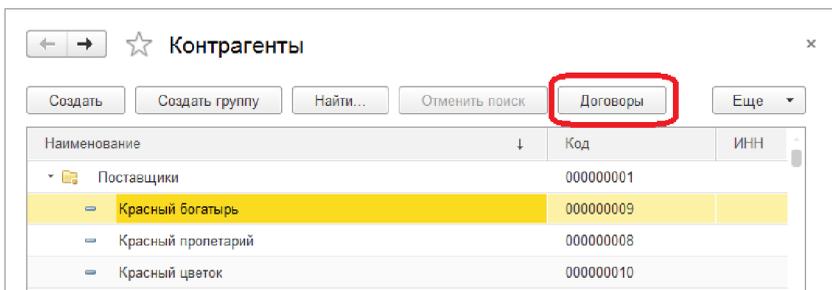


Рис. 3.36

Аналогичным образом может быть создана и форма элемента для справочника (Рис. 3.37). Главное, это указать правильный тип формы.

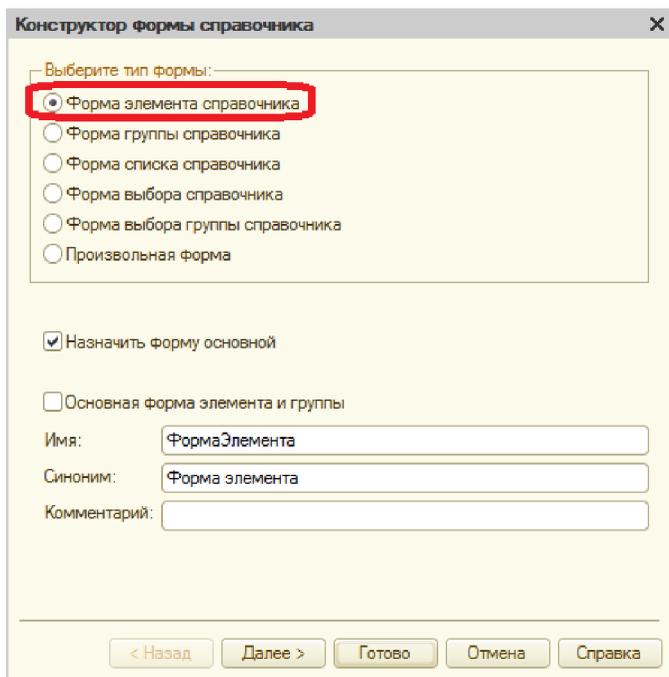


Рис. 3.37

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. 3.38)

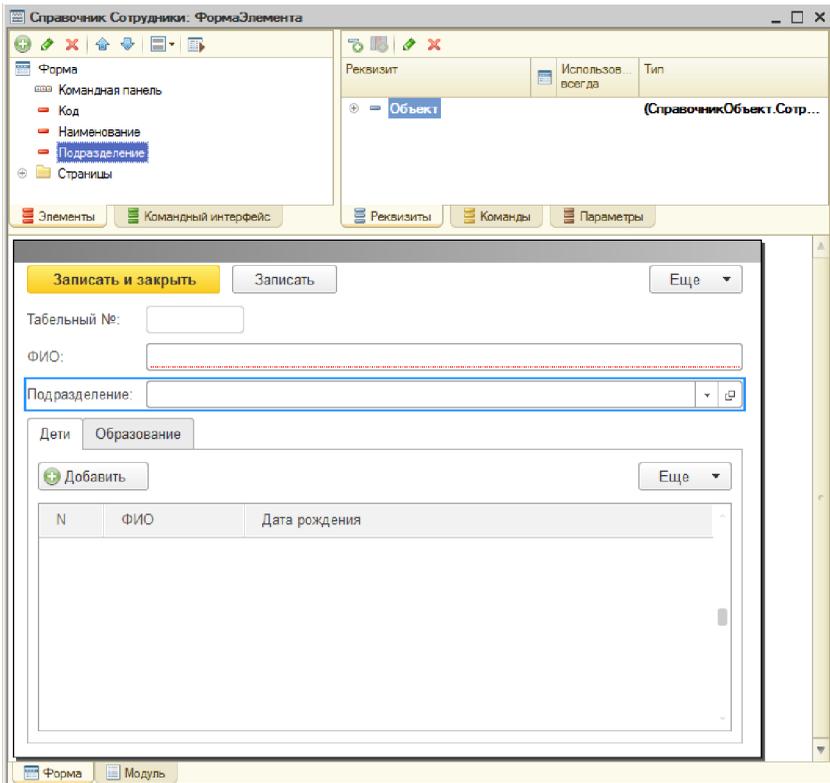


Рис. 3.38

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 predeterminedенные 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 конечногопользователя.

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. 4.1)

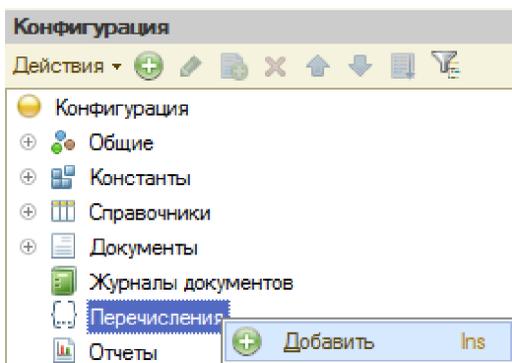


Рис. 4.1

Далее задаем имя: *ВидыПриемаНаРаботу*, при этом автоматически заполняется поле *Синоним* точно так же, как это происходило для справочников. (Рис. 4.2)

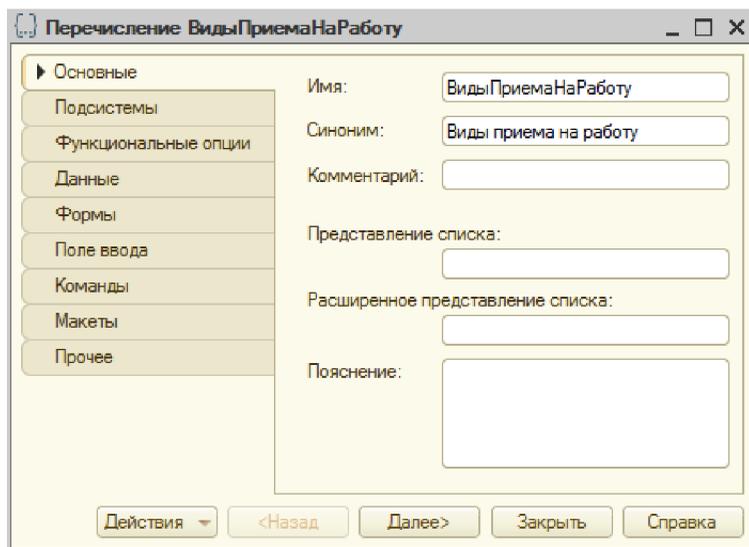


Рис. 4.2

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 нампалитра свойств (Рис. 4.3), в которой задаем *Имя* - значения перечисления. Аналогично создаем следующее значение.

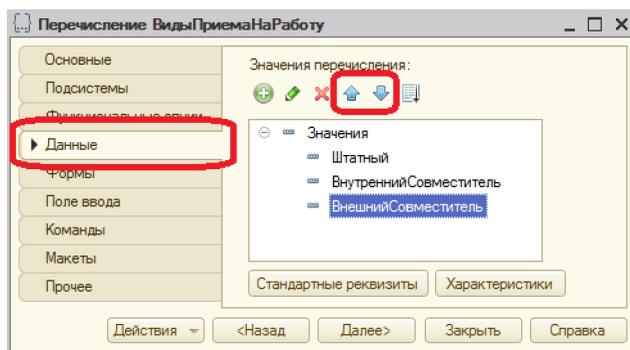


Рис. 4.3

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. 4.4).

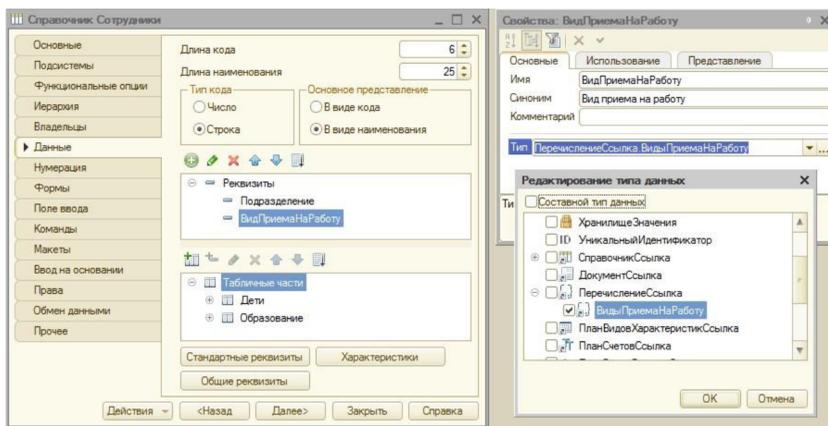


Рис. 4.4

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 будут 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 добавить используя пункт *Добавить* из контекстного меню, вызываемого правой кнопкой мышки (Рис. 5.1).



Рис. 5.1

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.

5.1.1. 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 ввода, в котором будут отражаться данные реквизита, можно воспользоваться двумя различными способами. Можно добавить новый элемент формы, указав тип *Поле* (Рис. 5.2) и указав затем в его свойствах *ПутьКДанным*, *Вид* и *Заголовок*(Рис. 5.3)

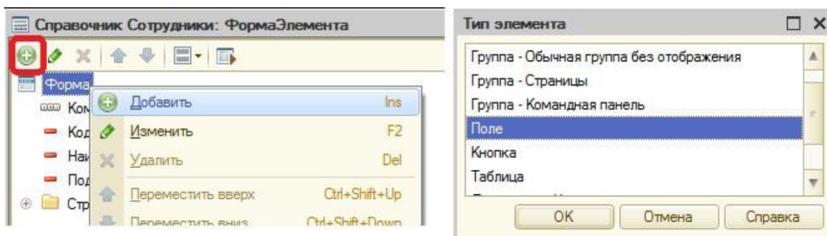


Рис. 5.2

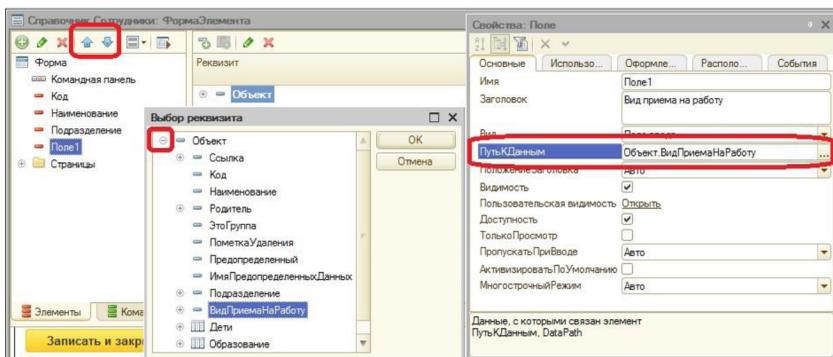


Рис. 5.3

Того же результата можно достичь, если придерживая левую кнопку мышки перетащить реквизит из окна реквизитов в окно элементов формы (Рис.5.4)

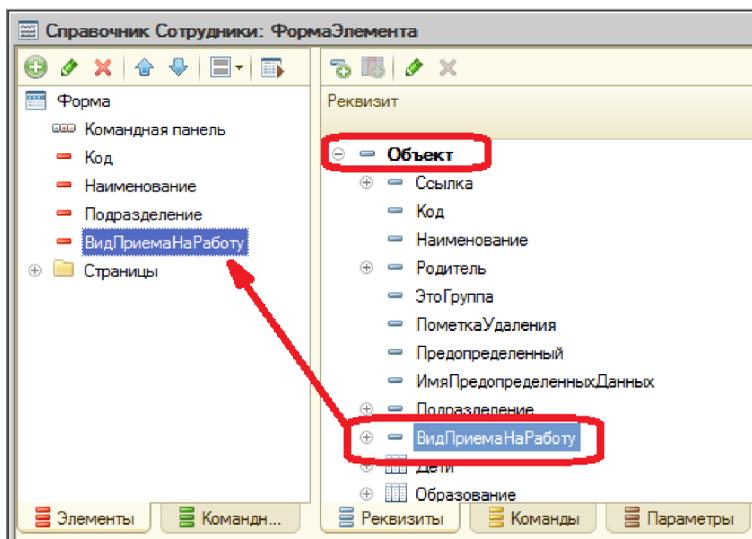


Рис. 5.4

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

Отображение Предупреждения При Редактировании

5.1.2. Поле флажка

С помощью флажка на форме могут отображаться данные с типом *Булево* или *Число*. Создадим форму элемента для справочника *Номенклатура* (Рис. 5.5).

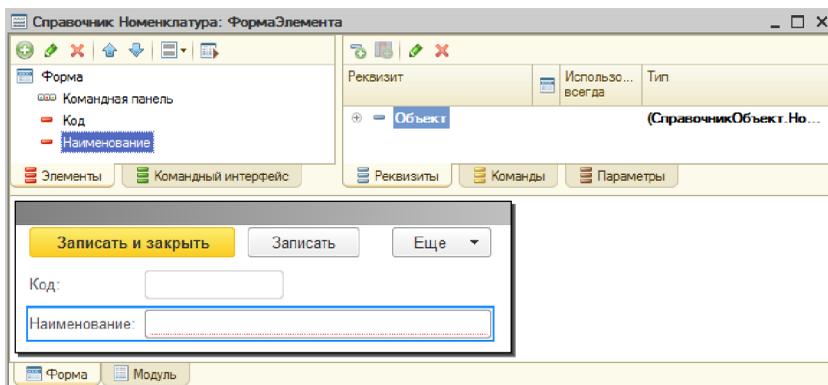


Рис. 5.5

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. 5.6)

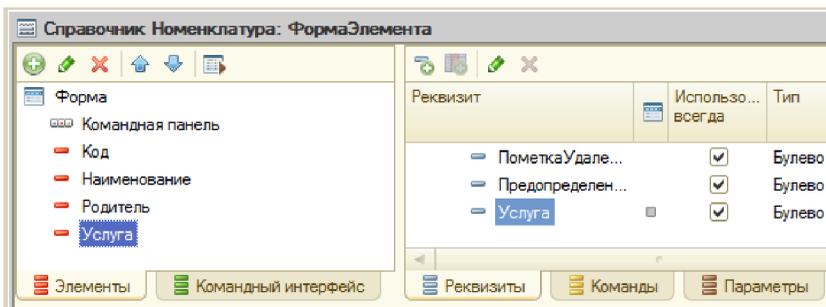


Рис. 5.6

Поскольку реквизиты с типом *Булево* удобнее всего отображать на форме в виде флажка, то система сама автоматически укажет данный вид поля (Рис. 5.7)

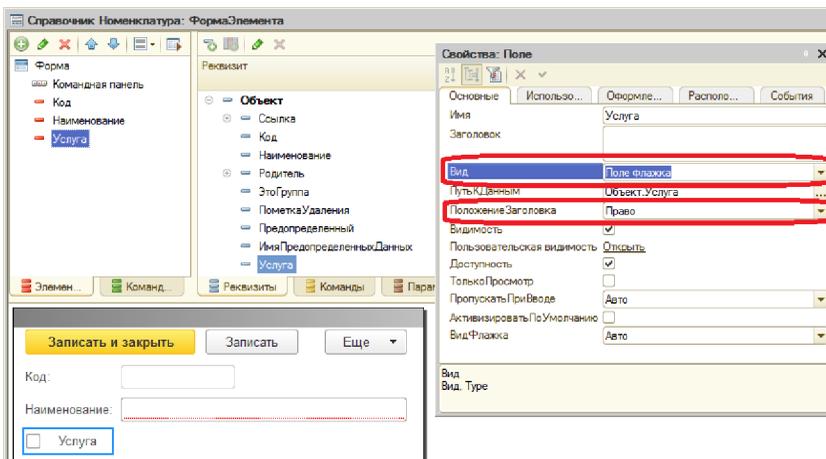


Рис. 5.7

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.

5.1.3. 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 *КачествоТовара* 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. 5.8)

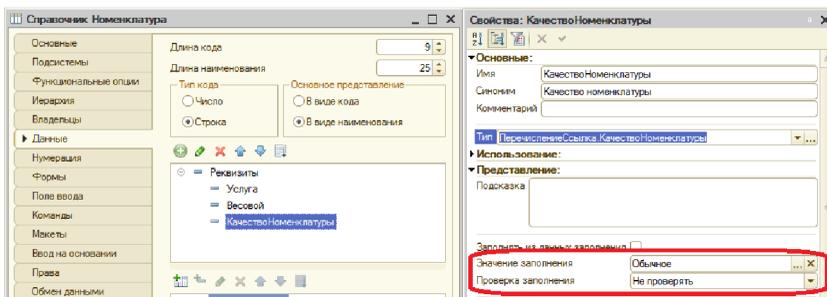


Рис. 5.8

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. 5.9).

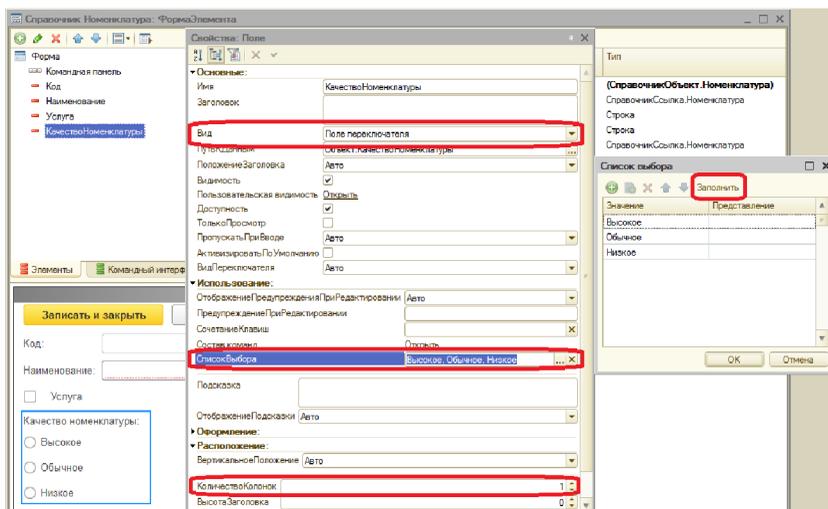


Рис. 5.9

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. 5.10), либо описаны на встроенном языке в модуле формы и затем связаны с командой (Рис. 5.11) 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.

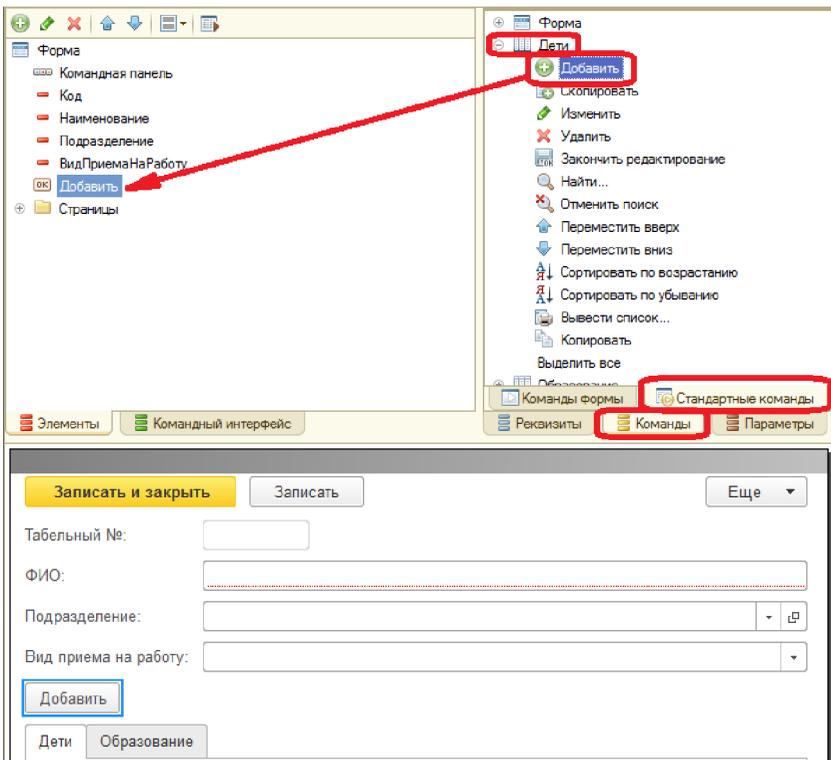


Рис. 5.10

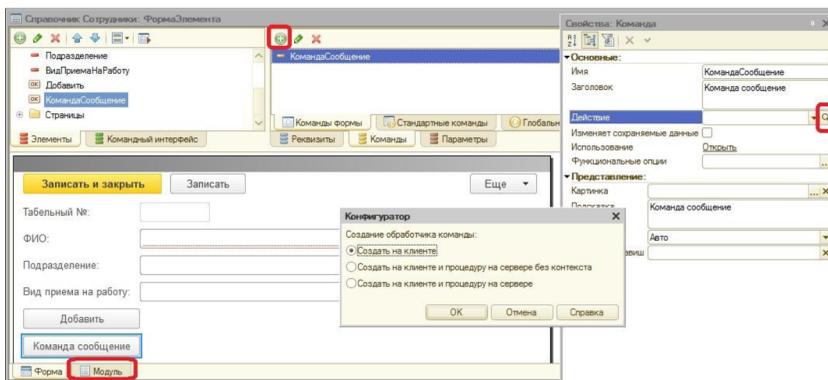


Рис. 5.11

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.

5.3.1. Обычная группа

Элемент *Группа* с видом *Обычная группа* позволяет визуально объединить несколько элементов на форме, управлять их расположением на форме (Рис.5.12)

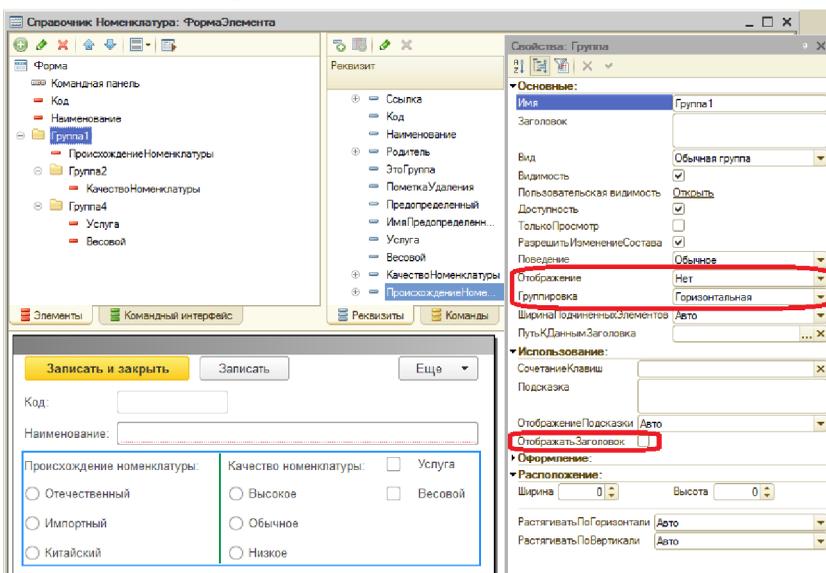


Рис. 5.12

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.

5.3.2. Командная панель

Командная панель позволяет группировать только кнопки, вызывающих команды, которые могут быть как стандартными, так и созданными разработчиком (Рис. 5.13).

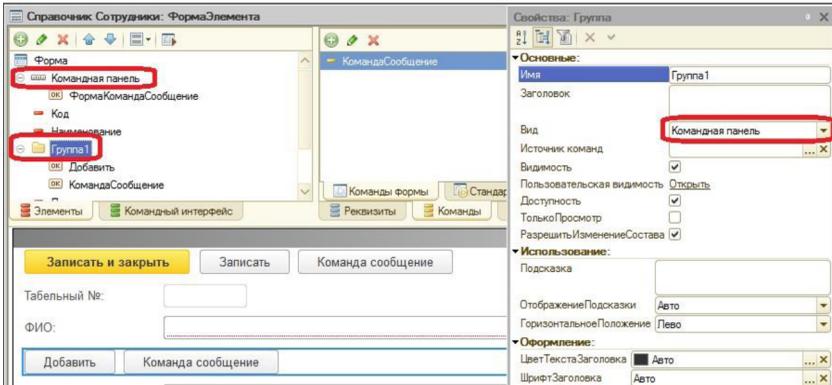


Рис. 5.13

5.3.3. 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. 5.15 и Рис. 5.16) и уже на ней разместить элементы формы обычным порядком.

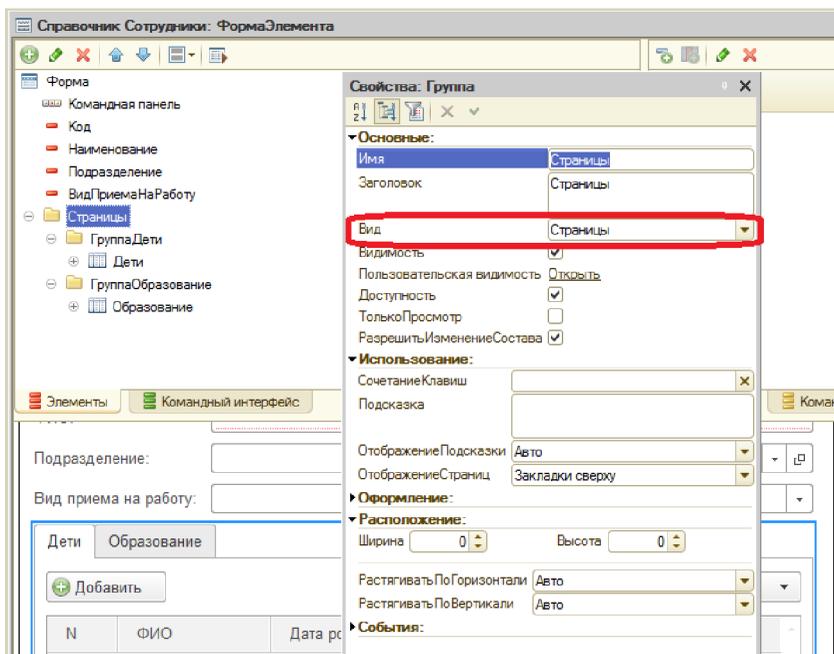


Рис. 5.14

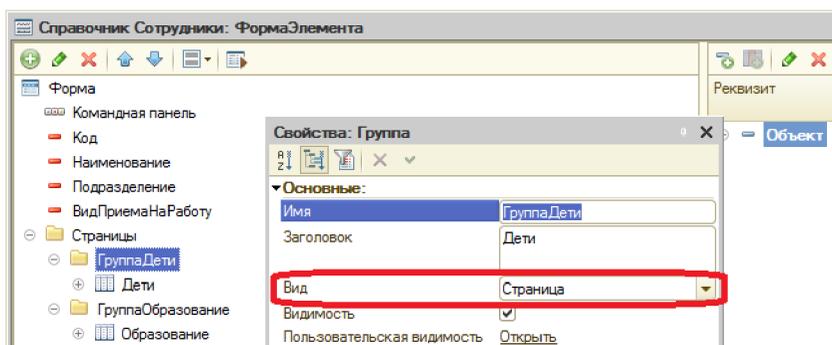


Рис. 5.15

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. 5.16)

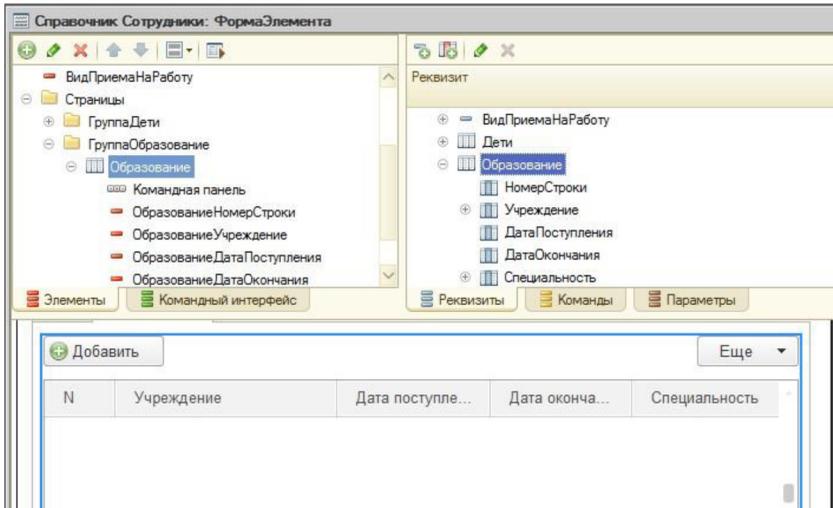


Рис. 5.16

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. 6.1).

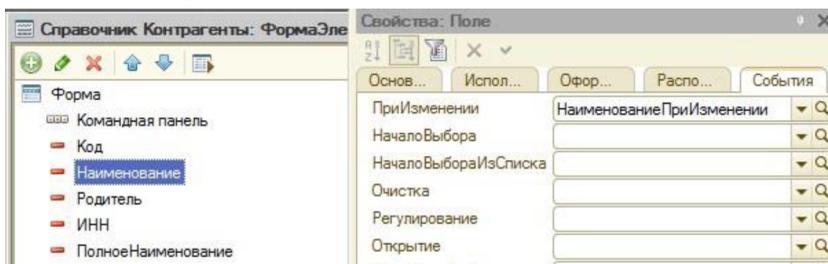


Рис. 6.1

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. 6.2)

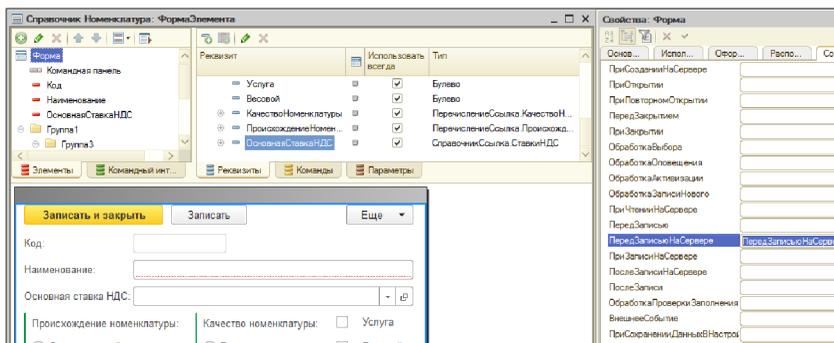


Рис. 6.2

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. 7.1). Далее необходимо определить структуру нашего документа, которая задается набором его реквизитов.

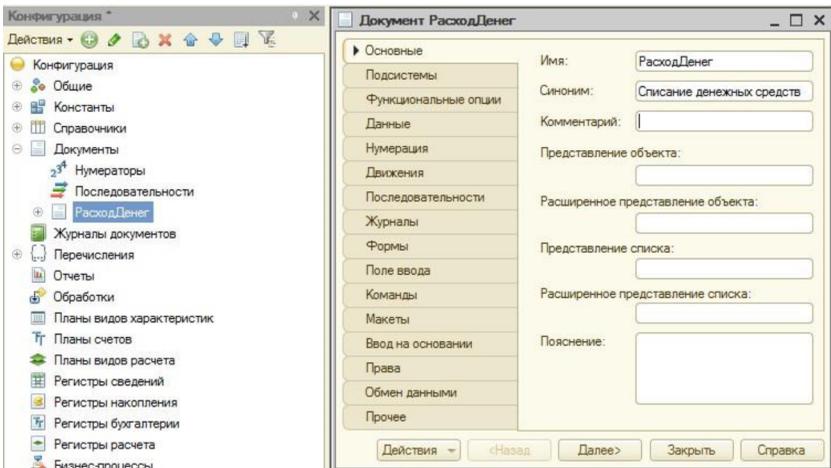


Рис. 7.1

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.

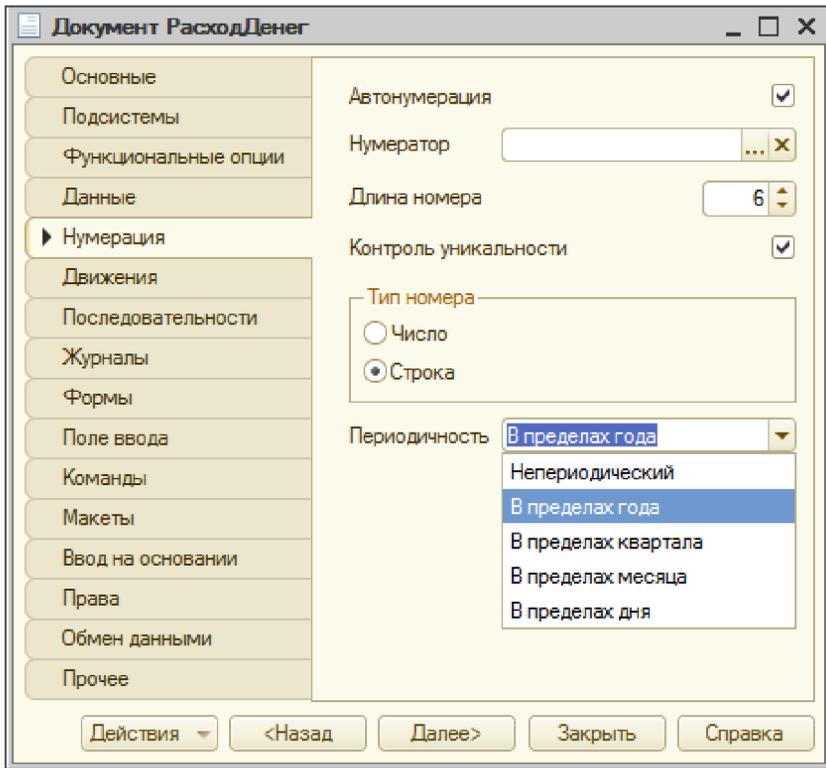


Рис. 7.2

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. 7.2):

- *Автонумерация* – данный флаг включает автоматическое формирование номера документа при его создании. Впоследствии 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:

- неперiodический;
- в пределах года;
- в пределах квартала;
- в пределах месяца;
- в пределах дня.

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

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

- Контрагент,
- Договор,
- Сумма.

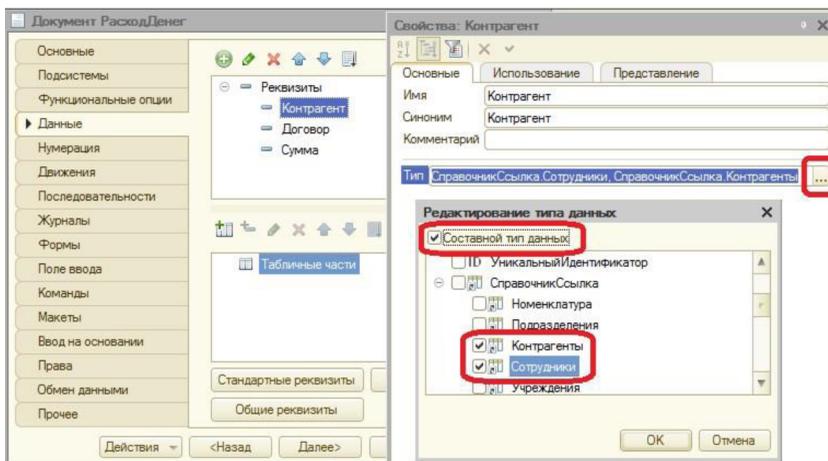


Рис. 7.3

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

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. 7.4), а для работы с несколькими документами данного вида используется *Форма списка* (Рис. 7.5).

← → ☆ Списание денежных средств 000005 от 02.01.2014 12:00:00 ×

Провести и закрыть Записать Провести Еще ▾

Номер: 000005

Дата: 02.01.2014 12:00:00

Контрагент: Красный богатырь ▾ □

Договор: кр бг № 2 ▾ □

Сумма: 600,00

Рис. 7.4

← → ☆ Списание денежных средств ×

Создать Найти... Отменить поиск Еще ▾

Дата	Номер	Контрагент	Договор	Сумма
01.01.2014 12:00:00	000001	Красный богатырь	кр бг № 1	5 000,00
01.01.2014 12:00:01	000006	Бендер О.И.		50 000,00
02.01.2014 12:00:00	000005	Красный богатырь	кр бг № 2	600,00
03.01.2014 12:00:00	000002	Красный богатырь	кр бг № 1	5 000,00
13.01.2014 12:00:00	000003	Красный богатырь	кр бг № 1	7 200,00
15.01.2014 14:40:07	000004	Красный богатырь	кр бг № 2	5 000,00

Рис. 7.5

В том случае, когда автоматически создаваемая форма не устраивает, имеется возможность создать соответствующую форму в конфигураторе, используя ранее рассмотренный механизм работы с формами (Рис. 7.6). Однако некоторые настройки внешнего вида и поведения формы можно указать и не создавая саму форму.

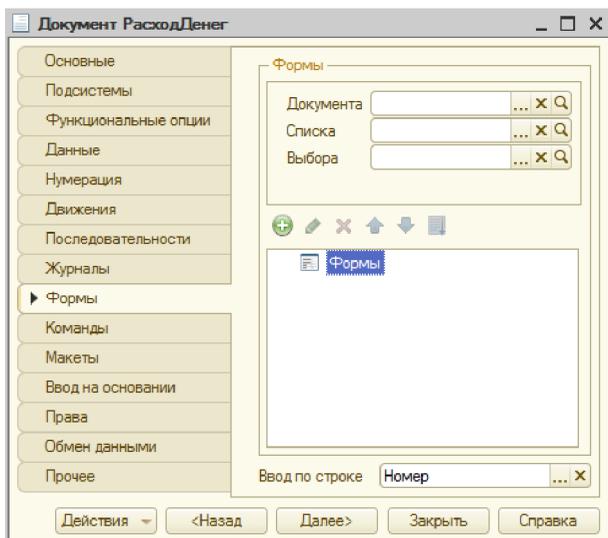


Рис. 7.6

Для того, чтобы в форме документа осуществлялся подбор только тех договоров, которые относятся к выбранному контрагенту, необходимо сделать следующую настройку (Рис. 7.7) В свойствах реквизита *Договор* необходимо установить связь параметров выбора с *Контрагентом*, который должен являться владельцем для *Договора*.

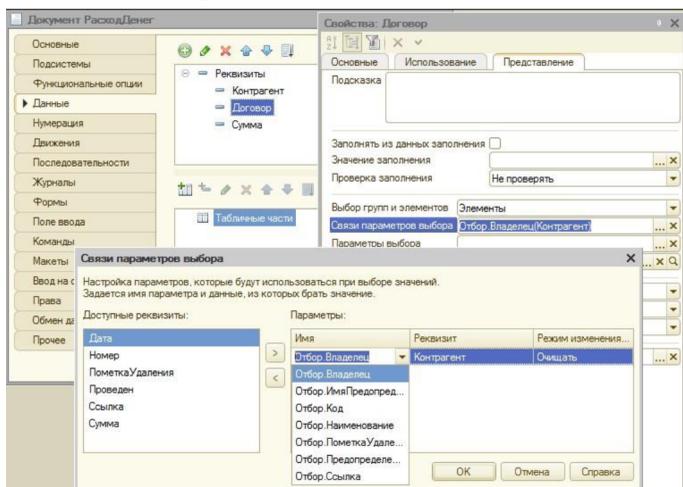


Рис. 7.7

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. 7.8 – Рис. 7.12) пользователь может добиться того, чтобы список отображалась наиболее удобным для него способом (Рис. 7.13).

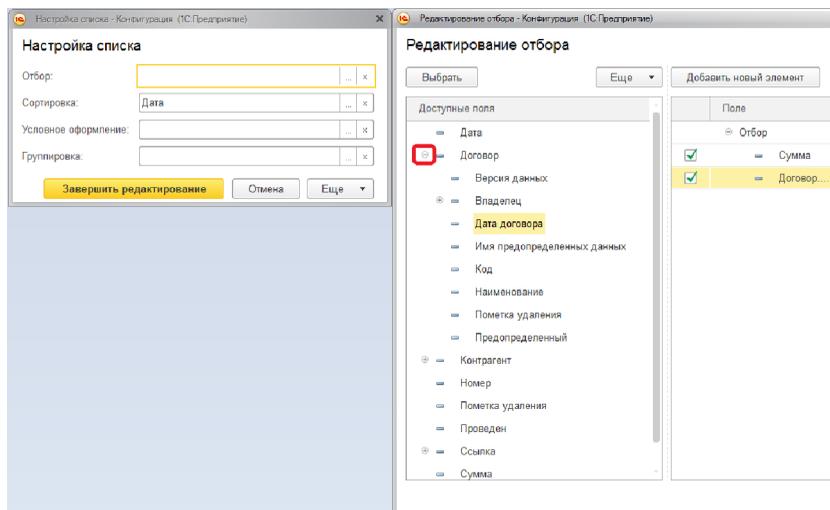
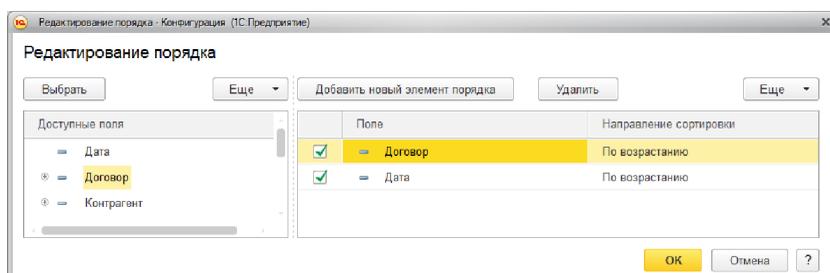


Рис. 7.8



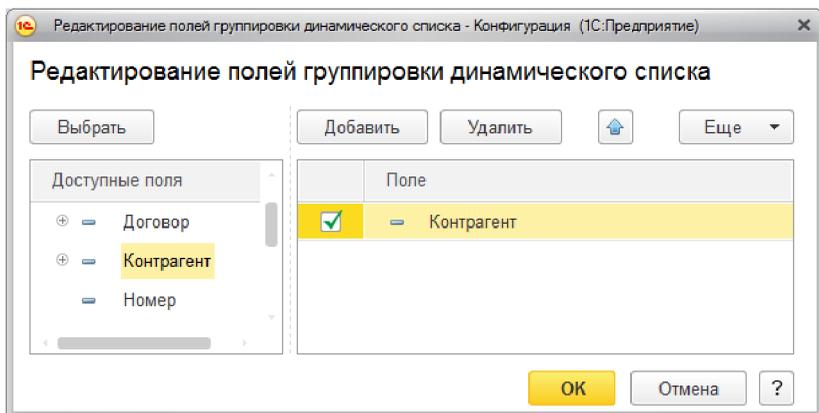


Рис. 7.9

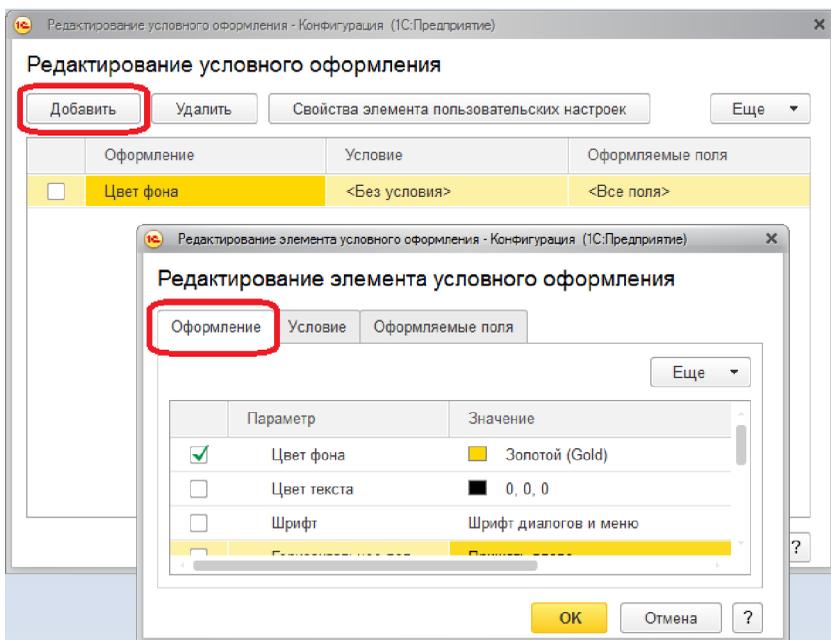


Рис. 7.10

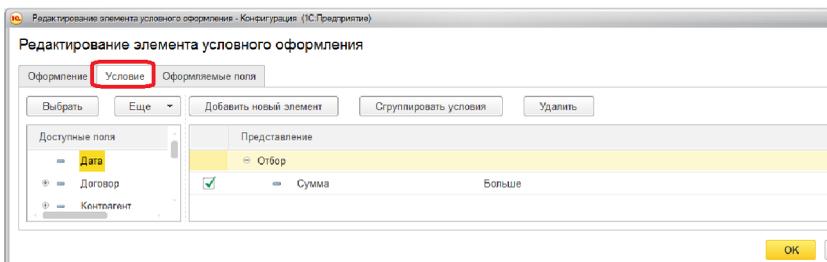


Рис. 7.11

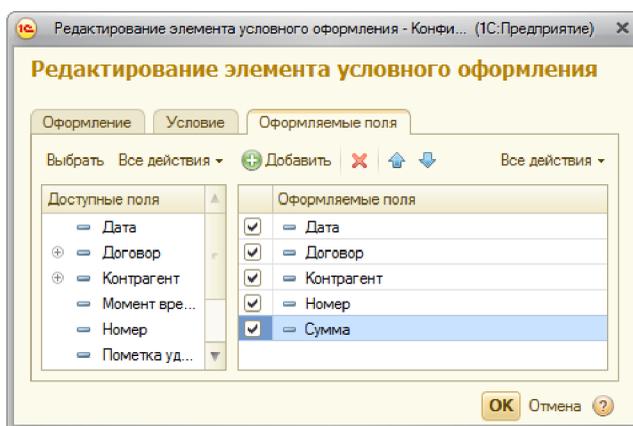


Рис. 7.12

Дата	Номер	Контрагент	Договор	Сумма
Бендер О.И.				
Красный богатырь				
01.01.2014 12:00:00	000001	Красный богатырь	кр бг № 1	5 000,00
03.01.2014 12:00:00	000002	Красный богатырь	кр бг № 1	5 000,00
13.01.2014 12:00:00	000003	Красный богатырь	кр бг № 1	8 600,00
02.01.2014 12:00:00	000005	Красный богатырь	кр бг № 2	600,00
09.01.2014 12:00:00	000007	Красный богатырь	кр бг № 2	7 150,00
15.01.2014 1:10:07	000004	Красный богатырь	кр бг № 2	5 000,00
Красный пролетарий				

Рис. 7.13

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

*Самостоятельно
Создайте документ Поступление денежных средств
(Рис. 7.14)*

← → ☆ Поступление денежных средств 000001 от 1... ×

Провести и закрыть Записать Провести Еще ▾

Поступление денежных средств №: 000001 от: 14.01.2014 12:00:00

Контрагент: MMM

Договор: ммм № 2

Сумма: 10 000,00 Управленческий

Номер документа плательщика: мм0064 **Способ оплаты**

Дата документа плательщика: 08.01.2014

Наличные
 Безналичные
 Электронные деньги

Комментарий: важное

Рис. 7.14

Поступление денежных средств 000001 от 14... ×

Провести и закрыть Записать Провести

Еще ▾

Провести и закрыть
Записать Ctrl+S
Перечитать
Скопировать
Пометить на удаление / Снять пометку
Удалить
Провести
Отмена проведения
Изменить форму...
Справка F1

Поступление денежных средств №: 000001 от: 14.01.2014 12:00:00

Контрагент: MMM

Договор: ммм № 2

Сумма: 10 000,00 Управленческий

Номер документа плательщика: мм0064 **Способ оплаты**

Дата документа плательщика: 08.01.2014

Наличные
 Безналичные
 Электронные деньги

Комментарий: важное

Рис. 7.15

Обратите внимание, что пользователь имеет возможность (в определенных пределах) изменить внешний вид формы

самостоятельно (Рис. 7.15 и Рис. 7.16). Для этого необходимо выбрать пункт меню *Изменить форму* из выпадающего списка, сформированного при нажатии на кнопку "Еще".

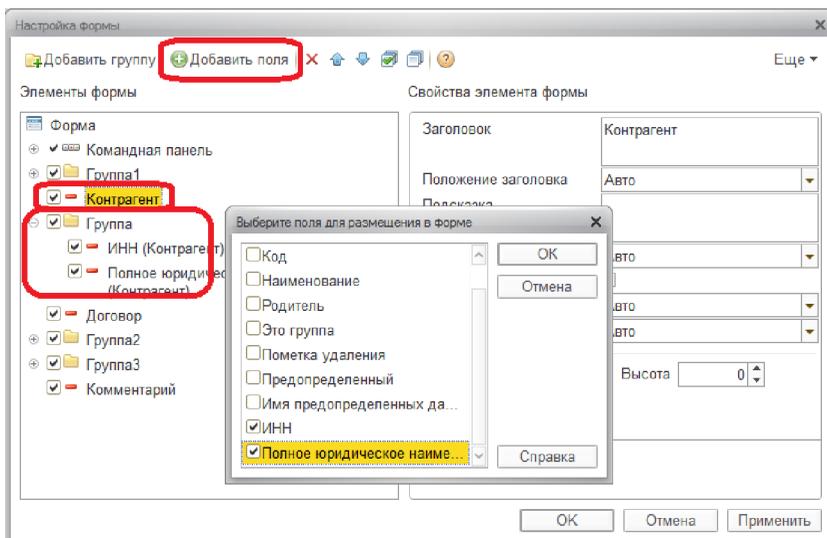


Рис. 7.16

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.7.17)

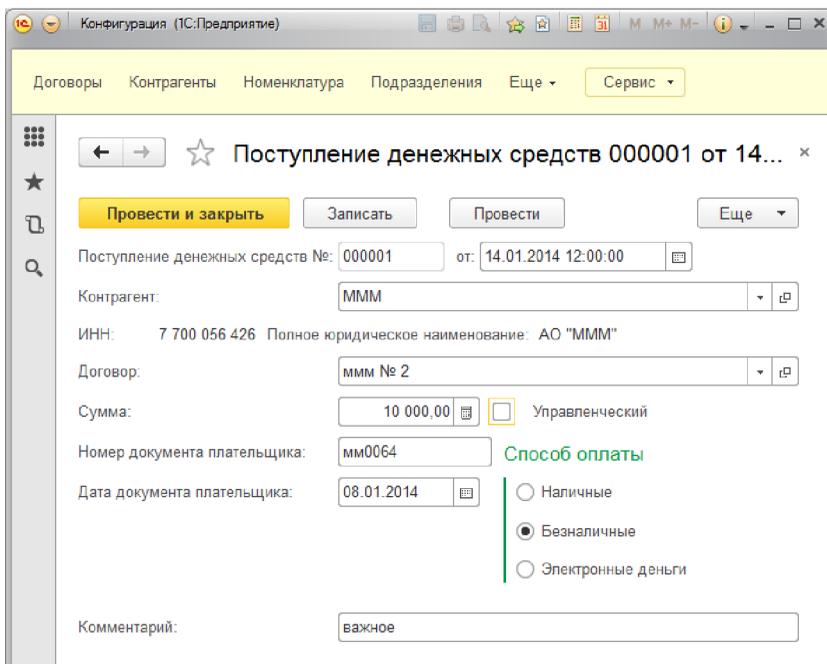


Рис. 7.17

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.7.18)

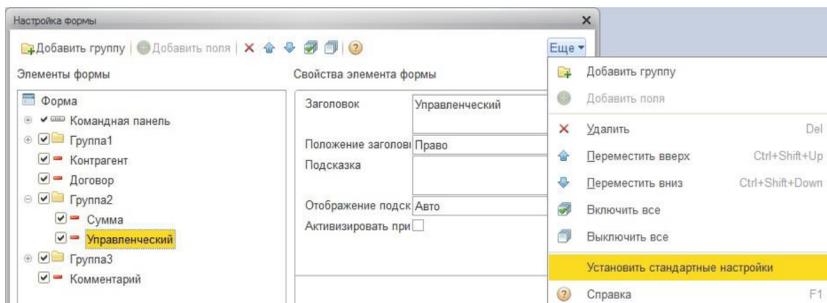


Рис. 7.18

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.7.19)

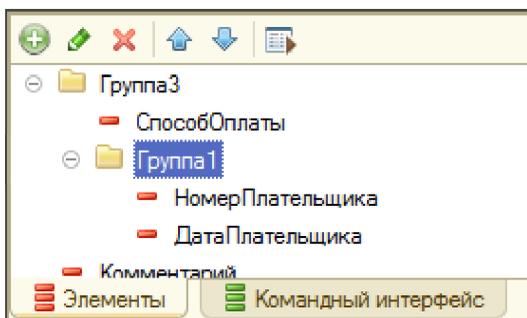


Рис. 7.19

Чтобы реквизиты исчезали, когда меняется способ оплаты, задействуем обработчик события *ПриИзменении* для способа оплаты (Рис. 7.20)

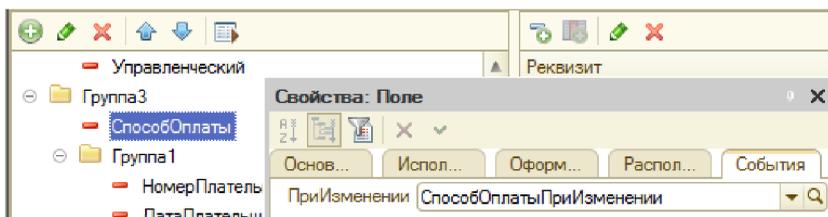


Рис. 7.20

Кроме того, необходимо использовать обработчик события *ПриОткрытии* для всей формы (Рис.7.21)

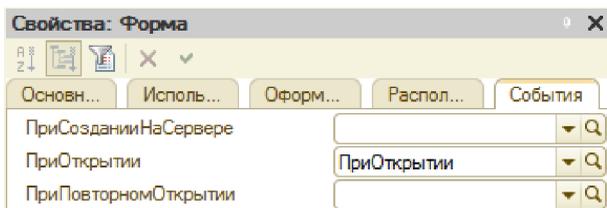


Рис. 7.21

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.7.22).

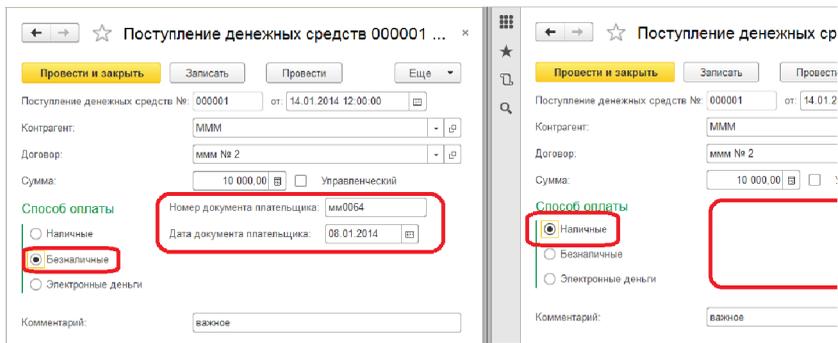


Рис. 7.22

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. 7.23).

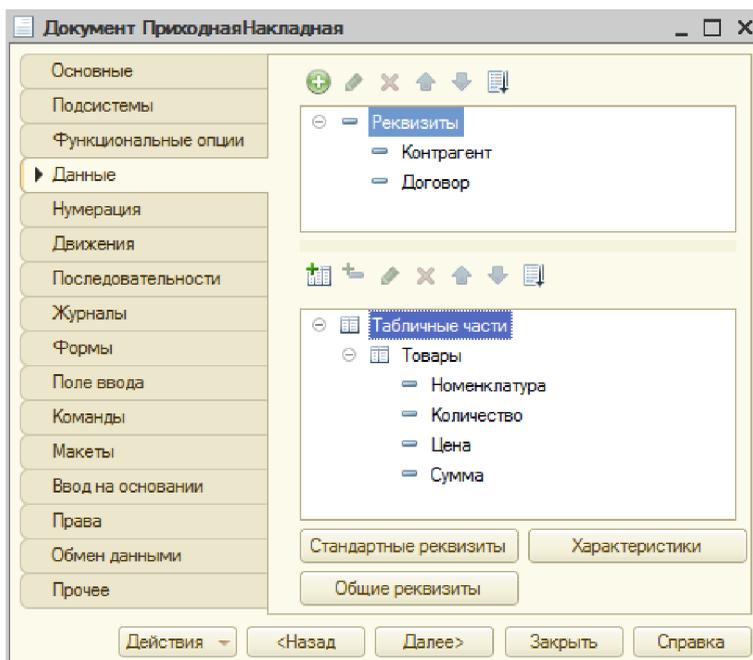


Рис. 7.23

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

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

Рис. 7.24

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.

Для расчета суммы необходимо задействовать обработчики событий *При изменении* для количества и цены (Рис. 7.24). 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

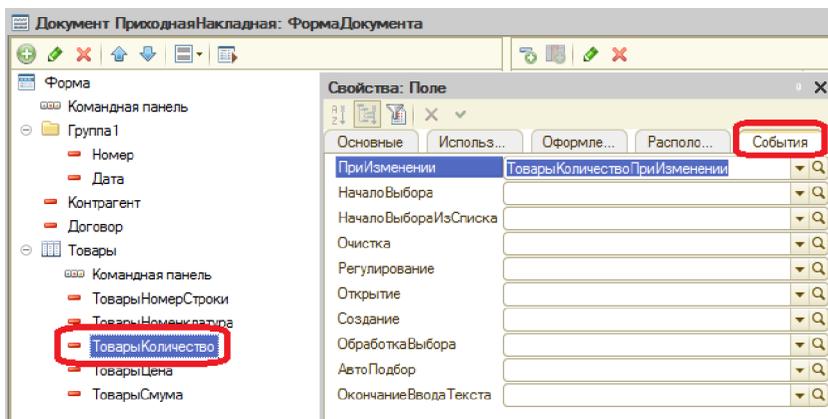


Рис. 7.25

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. 7.26)

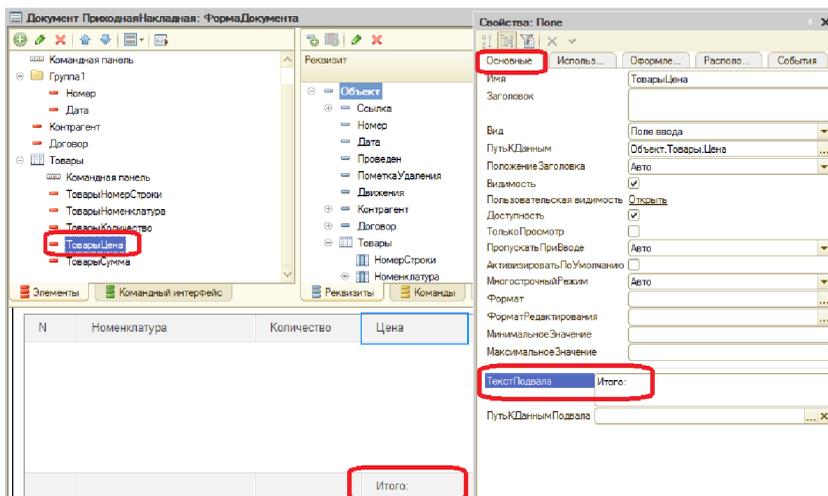


Рис. 7.26

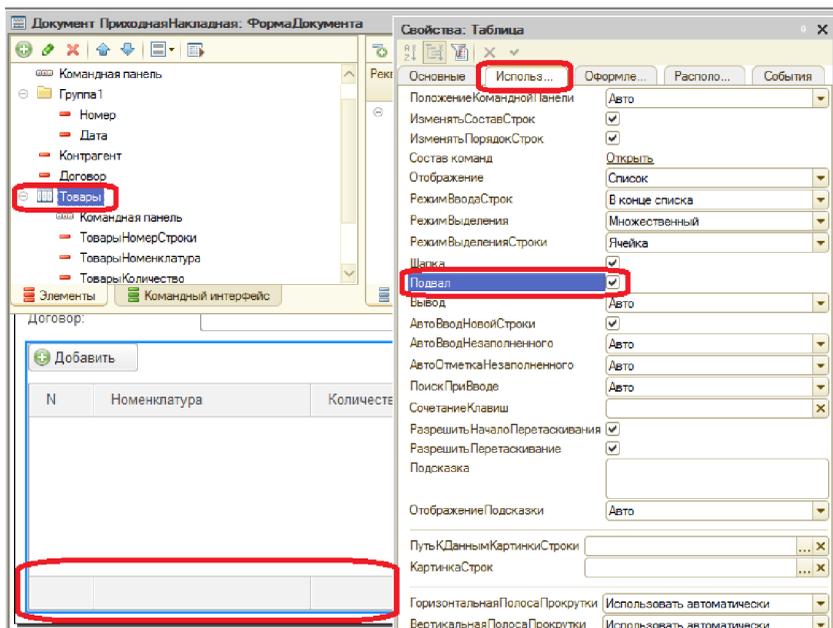


Рис. 7.27

В подвале можно отображать как заранее известный и фиксированный текст (Рис. 7.27), так и автоматически вычисляемые платформой данные. В нашем случае это будет итог по колонке сумма (Рис. 7.28).

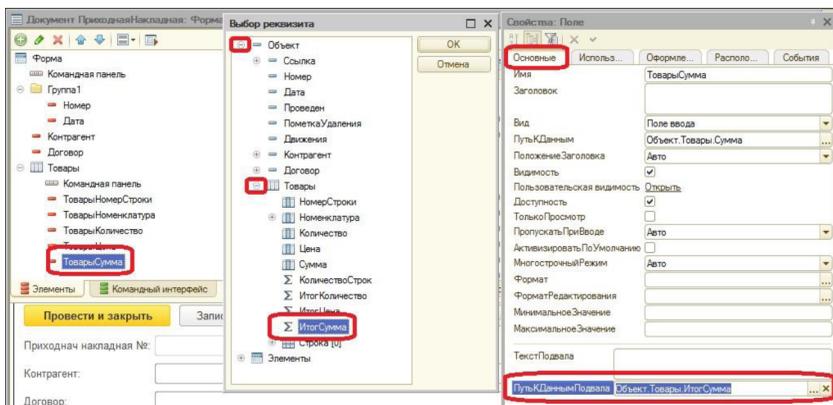


Рис. 7.28

В результате пользователь получит следующую форму

(Рис. 7.29) 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. 7.29

*Самостоятельно
Создайте документ Расходная накладная (Рис. 7.30 -
Рис. 7.32)*

← → ☆ Расходная накладная 000000001 от 15.01.2014 23:59:28

Провести и закрыть Записать Провести Еще ▾

Расходная накладная №: 000000001 от: 15.01.2014 23:59:28 📅

Контрагент: МММ ▾ 📄

Договор: лим № 2 ▾ 📄

Сотрудник: Паниковский М.С. ▾ 📄 Управленческий

Товары Услуги Прочее

Добавить Еще ▾

N	Товар	Количество	Цена	Ставка НДС	Всего
			Сумма	Сумма НДС	
1	Рога обыкновенные	5,000	3 000,00	18 %	17 700,00
			15 000,00		
2	Рога ветвистые	6,000	8 000,00	10 %	52 800,00
			48 000,00		
3	Рога обыкновенные	10,000	2 000,00	18 %	23 600,00
			20 000,00		
Итого товаров:			83 000,00		94 100,00
					11 100,00

Рис. 7.30

← → ☆ Расходная накладная 000000001 от 15.01.2014 23:59:28 ×

Провести и закрыть Записать Провести Еще ▾

Расходная накладная №: 000000001 от: 15.01.2014 23:59:28 🗄

Контрагент: MMM ▾ 🗄

Договор: ммм № 2 ▾ × 🗄

Сотрудник: Паниковский М.С. ▾ 🗄 Управленческий

Товары Услуги Прочее

Добавить Еще ▾

N	Услуга	Сумма	Сумма НДС	Всего
1	Доставка	2 000,00	360,00	2 360,00
2	Полировка	5 000,00	900,00	5 900,00
Итого услуг:		7 000,00	1 260,00	8 260,00

Рис. 7.31

← → ☆ Расходная накладная 000000001 от 15.01.2014 23:59:28 ×

Провести и закрыть Записать Провести Еще ▾

Расходная накладная №: 000000001 от: 15.01.2014 23:59:28 🗄

Контрагент: MMM ▾ 🗄

Договор: ммм № 2 ▾ × 🗄

Сотрудник: Паниковский М.С. ▾ 🗄 Управленческий

Товары Услуги Прочее

Комментарий:

Время доставки с 16:00 до 17:30

Рис. 7.32

Документ *РасходнаяНакладная* имеет более сложную структуру, чем рассмотренный ранее документ *ПриходнаяНакладная*, поэтому задействуем еще ряд механизмов для облегчения работы пользователя. Для того, чтобы в табличную часть *Товары* пользователь мог подбирать только товары, а в табличную часть *Услуги*, соответственно, услуги, в свойствах реквизитов установим параметры выбора (Рис. 7.33).

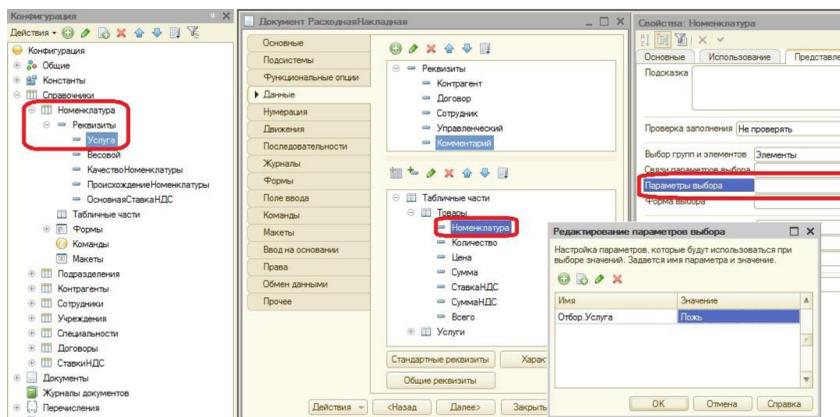


Рис. 7.33

Так же улучшим работу пользователя с сотрудником. Сделаем так, что доступными к выбору станут только не уволенные сотрудники. Нам потребуется создать новую форму выбора в справочнике "Сотрудники", причем форма должна быть неосновной, т.е. будет использовано только для нашей расходной накладной (Рис. 7.34)

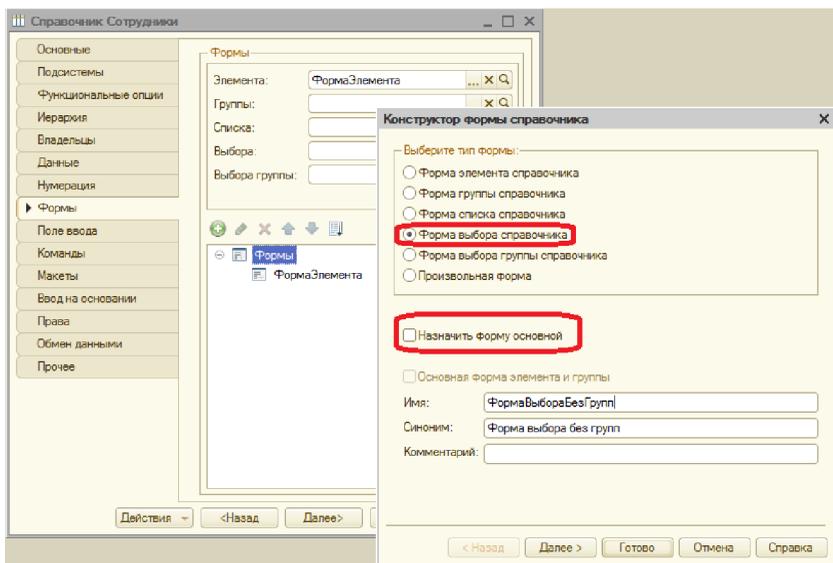


Рис. 7.34

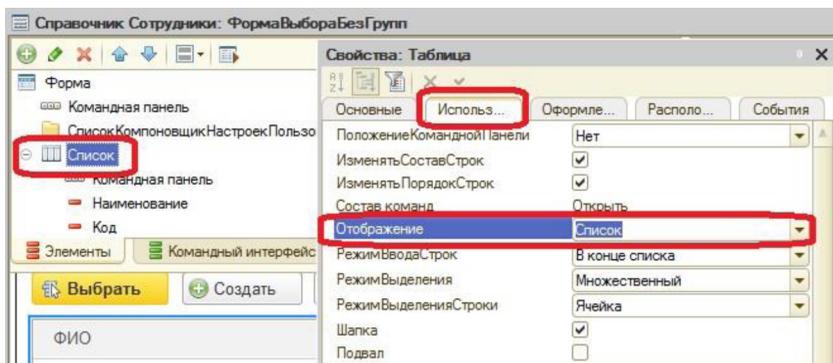


Рис. 7.35

В этой форме мы данные будем отображать без учета иерархии (Рис. 7.35)

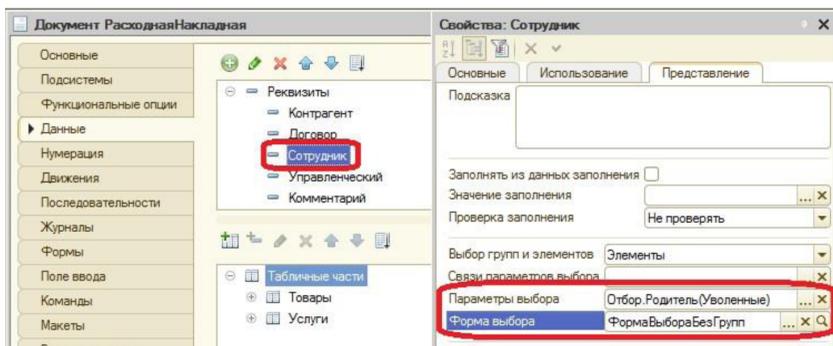


Рис. 7.36

Теперь укажем, что в расходной накладной выбор сотрудника будет происходить именно посредством новой формы (Рис. 7.36) и добавим отбор только работающих сотрудников (Рис. 7.37).

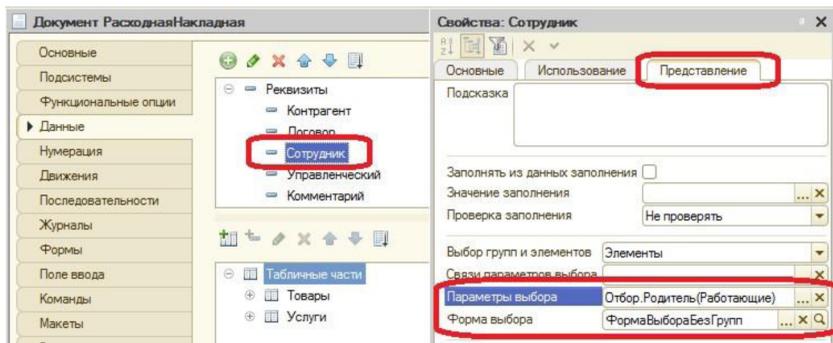


Рис. 7.37

Поскольку пользователь уже указал для каждого товара ставку НДС, по которой товар должен продаваться, то желательно эту ставку автоматически подставлять в документ. Для этого будем использовать обработчик события *ПриИзменении* для номенклатуры в таблице *Товары*.

Естественно, необходимо произвести расчет суммы НДС, согласно указанной ставки. Поскольку ставка НДС это ссылка на элемент справочника, то операция умножения над ней не выполнится. Чтобы получить для каждой ставки НДС число для вычисления суммы НДС, в справочник *СтавкиНДС* добавим новый

реквизит *Ставка* (Рис. 7.38).

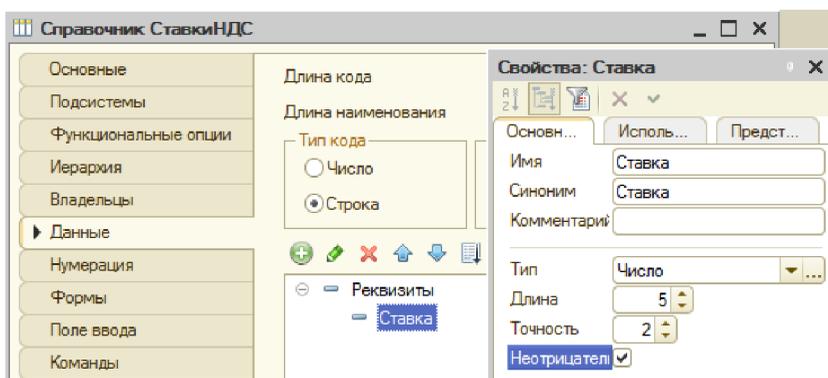


Рис. 7.38

Кроме того, чтобы в форме списка пользователь мог видеть общую сумму документа, необходимо рассчитать реквизит *СуммаДокумента*, для чего воспользуемся обработчиком события *ПередЗаписью*. Это событие будем обрабатывать не для формы, а для самого объекта, поэтому текст процедуры обработчика события будет находиться в модуле документа.

&НаКлиенте

Процедура ТоварыСуммаНДСПриИзменении(Элемент)

СтрокаРасчета = Элементы.Товары.ТекущиеДанные;

*СтрокаРасчета.Всего = СтрокаРасчета.Сумма +
СтрокаРасчета.СуммаНДС;*

КонецПроцедуры

&НаКлиенте

Процедура ТоварыСтавкаНДСПриИзменении(Элемент)

СтрокаРасчета = Элементы.Товары.ТекущиеДанные;

СтавкаЧисло = ПолучитьСтавку(СтрокаРасчета.СтавкаНДС);

*СтрокаРасчета.СуммаНДС = СтрокаРасчета.Сумма * СтавкаЧисло;*

ТоварыСуммаНДСПриИзменении(Элемент);

КонецПроцедуры

&НаСервереБезКонтекста

Функция ПолучитьСтавку(СтавкаНДС)

Возврат СтавкаНДС.Ставка / 100;

КонецФункции

&НаКлиенте

Процедура ТоварыСуммаПриИзменении(Элемент)

ТоварыСтавкаНДСПриИзменении(Элемент);

КонецПроцедуры

&НаКлиенте

Процедура ТоварыЦенаПриИзменении(Элемент)

СтрокаРасчета = Элементы.Товары.ТекущиеДанные;

*СтрокаРасчета.Сумма = СтрокаРасчета.Цена **

СтрокаРасчета.Количество; ТоварыСуммаПриИзменении(Элемент);

КонецПроцедуры

&НаКлиенте

Процедура ТоварыКоличествоПриИзменении(Элемент)

ТоварыЦенаПриИзменении(Элемент);

КонецПроцедуры

&НаКлиенте

Процедура ТоварыНоменклатураПриИзменении(Элемент)

СтрокаРасчета = Элементы.Товары.ТекущиеДанные;

СтрокаРасчета.СтавкаНДС =

ПолучитьСтавкуТовара(СтрокаРасчета.Номенклатура);

ТоварыСтавкаНДСПриИзменении(Элемент);

КонецПроцедуры

&НаСервереБезКонтекста

Функция ПолучитьСтавкуТовара(Товар)

Возврат Товар.ОсновнаяСтавкаНДС;

КонецФункции

Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

СуммаДокумента = Товары.Итог("Все") + Услуги.Итог("Все");

КонецПроцедуры

7.4. Печать документа. Конструкторпечати

Для любой таблицы в платформе предусмотрена стандартная возможность вывода этой таблицы на печать. Для чего доступна команда "Вывести список" из контекстного списка, открывающегося по кнопке "Еще" (Рис. 7.39).

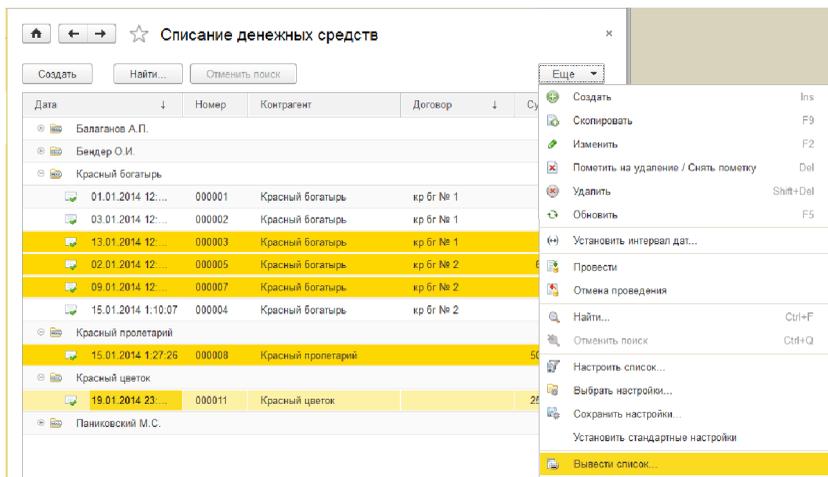


Рис. 7.39

Печатная форма будет содержать все те настройки, которые мы выполнили для формы: группировки, отборы, сортировка и т.д. Тем не менее, для печати документа такой механизм нас не устроит, поскольку для нашей накладной необходимо вывести данные из нескольких разных таблиц одновременно. Кроме того, внешний вид печатной формы должен быть не автоматическим, а таким, который будет устраивать пользователя. В том числе, внешний вид печатной формы может определяться и требованиями законодательства.

Чтобы пользователь мог формировать свою печатную форму документа, воспользуемся конструктором печати. Вызов конструктора осуществляется на вкладке *Макеты* по кнопке *Конструкторы*. Следуя инструкциям конструктора (Рис. 7.40–Рис. 7.45), заполняем все необходимые поля.

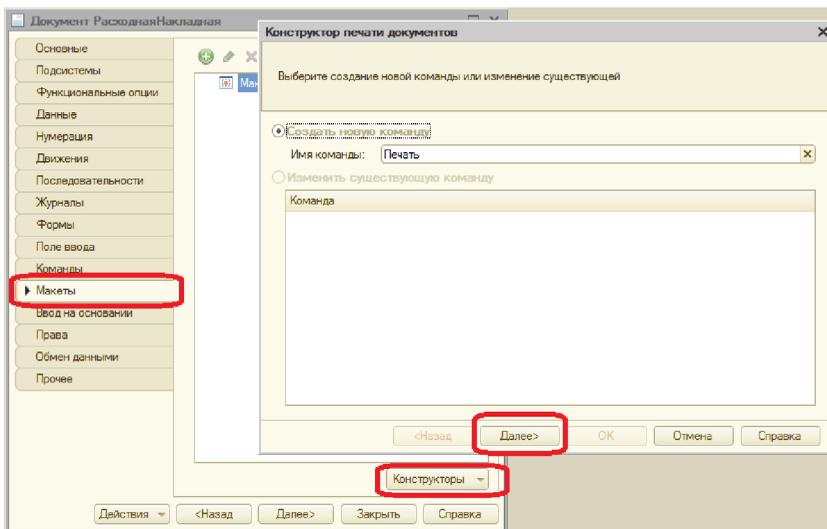


Рис. 7.40

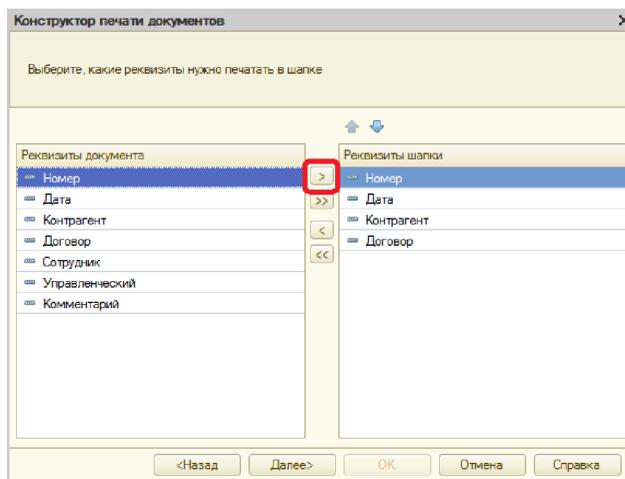


Рис. 7.41

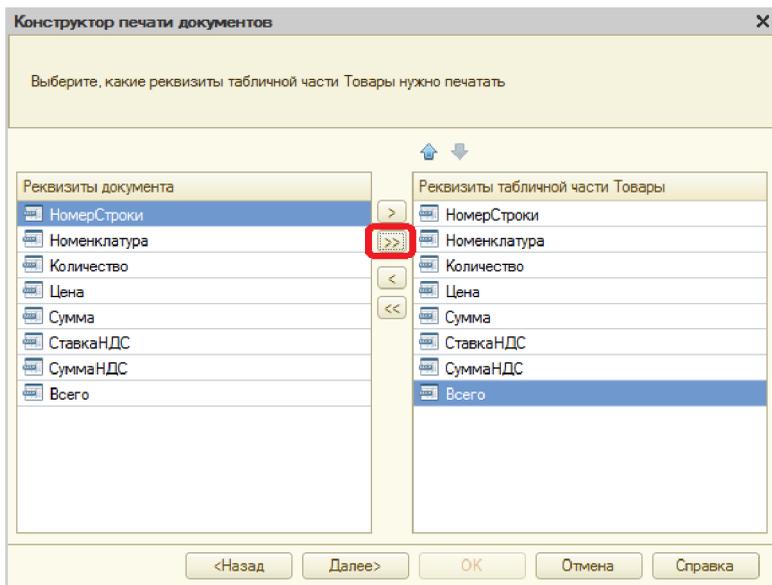


Рис. 7.42

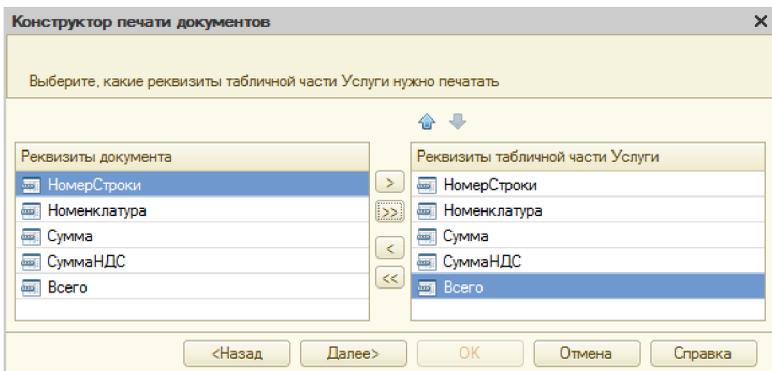


Рис. 7.43

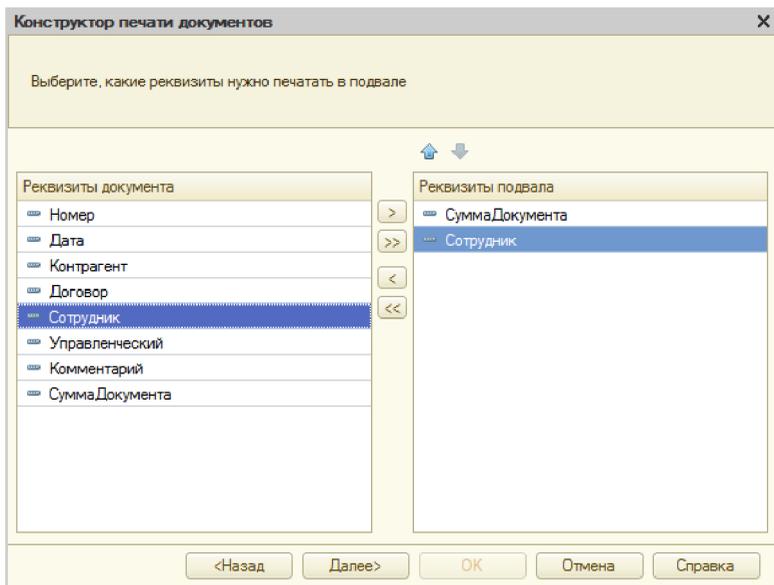


Рис. 7.44

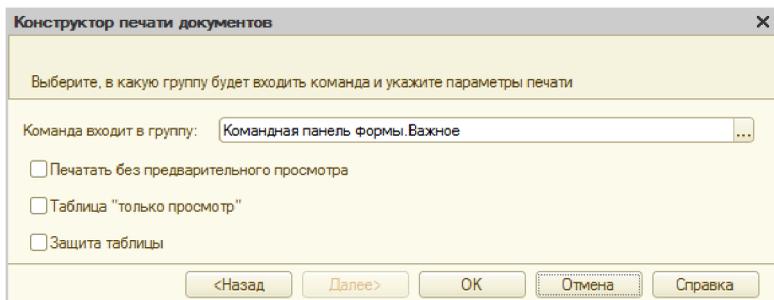


Рис. 7.45

В результате работы конструктора будет создан новый макет (Рис. 7.46), новая команда и сформируется следующий программный код в модуле команды и модуле менеджера:

&НаКлиенте

Процедура ОбработкаКоманды

(ПараметрКоманды, ПараметрыВыполненияКоманды)

*/// *_КОНСТРУКТОР_ПЕЧАТИ(Печать)**

ТабДок = Новый ТабличныйДокумент;

Печать(ТабДок, ПараметрКоманды);

ТабДок.ОтображатьСетку = Ложь;

ТабДок.Защита = Ложь;

ТабДок.ТолькоПросмотр = Ложь;

ТабДок.ОтображатьЗаголовки = Ложь;

ТабДок.Показать();

//}}

КонецПроцедуры

&НаСервере

*Процедура Печать(ТабДок, ПараметрКоманды)
Документы.РасходнаяНакладная.Печать
(ТабДок, ПараметрКоманды);*

КонецПроцедуры

Процедура Печать(ТабДок, Ссылка) Экспорт

//{{_КОНСТРУКТОР_ПЕЧАТИ(Печать)

Макет = Документы.РасходнаяНакладная.ПолучитьМакет("Печать");

Запрос = Новый Запрос;

Запрос.Текст = "ВЫБРАТЬ

*| РасходнаяНакладная.Дата,
| РасходнаяНакладная.Договор,
| РасходнаяНакладная.Контрагент,
| РасходнаяНакладная.Номер,
| РасходнаяНакладная.Сотрудник,
| РасходнаяНакладная.СуммаДокумента,
| РасходнаяНакладная.Товары.(
| НомерСтроки,
| Номенклатура,
| Количество,
| Цена,
| Сумма,*

| *СтавкаНДС,*
| *СуммаНДС,*
| *Всего*
|),
| *РасходнаяНакладная.Услуги.(*
| *НомерСтроки,*
| *Номенклатура,*
| *Сумма,*
| *СуммаНДС,*
| *Всего*
|)

| *ИЗ*

| *Документ.РасходнаяНакладная КАК РасходнаяНакладная*
| *ГДЕ*

| *РасходнаяНакладная.Ссылка В (&Ссылка)";*
Запрос.Параметры.Вставить("Ссылка", Ссылка);
Выборка = Запрос.Выполнить().Выбрать();
ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");
Шапка = Макет.ПолучитьОбласть("Шапка");
ОбластьТоварыШапка = Макет.ПолучитьОбласть("ТоварыШапка");
ОбластьТовары = Макет.ПолучитьОбласть("Товары");
ОбластьУслугиШапка = Макет.ПолучитьОбласть("УслугиШапка");
ОбластьУслуги = Макет.ПолучитьОбласть("Услуги");
Подвал = Макет.ПолучитьОбласть("Подвал");
ТабДок.Очистить();
ВставлятьРазделительСтраниц = Ложь;
Пока Выборка.Следующий() Цикл
 Если ВставлятьРазделительСтраниц Тогда
ТабДок.ВывестиГоризонтальныйРазделительСтраниц();
 КонецЕсли;
 ТабДок.Вывести(ОбластьЗаголовок);
 Шапка.Параметры.Заполнить(Выборка);

ТабДок.Вывести(Шапка, Выборка.Уровень());
ТабДок.Вывести(ОбластьТоварыШапка);
ВыборкаТовары = Выборка.Товары.Выбрать();
Пока ВыборкаТовары.Следующий() Цикл

ОбластьТовары.Параметры.Заполнить(ВыборкаТовары);
ТабДок.Вывести(ОбластьТовары,
ВыборкаТовары.Уровень());

КонецЦикла;
ТабДок.Вывести(ОбластьУслугиШапка);
ВыборкаУслуги = Выборка.Услуги.Выбрать();
Пока ВыборкаУслуги.Следующий() Цикл

ОбластьУслуги.Параметры.Заполнить(ВыборкаУслуги);
ТабДок.Вывести(ОбластьУслуги,
ВыборкаУслуги.Уровень());

КонецЦикла;
Подвал.Параметры.Заполнить(Выборка);
ТабДок.Вывести(Подвал);
ВставлятьРазделительСтраниц = Истина;

КонецЦикла;
//}

КонецПроцедуры

№	Товар	Количество	Цена	Сумма	Ставка НДС	Сумма НДС	Всего
12							
13							
14	№	Товар	Количество	Цена	Сумма	Ставка НДС	Сумма НДС
15	№	Товар	Количество	Цена	Сумма	Ставка НДС	Сумма НДС
16							
17	№	Услуга	Сумма	Сумма НДС	Всего		
18	№	Услуга	Сумма	Сумма НДС	Всего		
19							
20							
21	Сумма документа	Документа					
22							

Рис. 7.46

Кроме того, в форме списка и в форме документа (Рис. 7.45) на командной панели появилась кнопка *Печать* (Рис. 7.46), нажатие на которую позволяет получить печатную форму документа (Рис. 7.47).

← → ☆ Расходная накладная 000000001 от 15.01.2014 23:59:28

Провести и закрыть Записать Провести **Печать** Еще ▾

Расходная накладная №: 000000001 от: 15.01.2014 23:59:28

Контрагент: MMM

Договор: ммм № 2

Сотрудник: Паниковский М.С. Управленческий

Товары Услуги Прочее

Добавить Еще ▾

N	Товар	Количество	Цена	Ставка НДС	Всего
			Сумма	Сумма НДС	
1	Рога обыкновенные	5,000	3 000,00	18 %	17 700,00
			15 000,00	2 700,00	
2	Рога ветвистые	6,000	8 000,00	10 %	52 800,00

Рис. 7.47

← → ☆ Расходная накладная

Создать Найти... Отменить поиск **Печать**

Дата	Номер	Контрагент	Договор	Сотрудник	У. ↓	Сумма доку...	Ко
15.01.2014 23:59:28	000000001	MMM	mmm № 2	Паниковский М.С.		102 360,00	
16.01.2014 0:29:20	000000002	Хопёр-Инвест		Балаганов А.П.	✓	33 600,00	

Рис. 7.48

Конфигурация: ПС:Предприятие

Договоры Контрагенты Номенклатура Подразделения Поступление денежных средств Приходная накладная Расходная накладная Сотрудники Еще ▾

← → ☆ Таблица

Расходная накладная

Номер: 000000001
Дата: 15.01.2014 23:59:28
Контрагент: MMM
Договор: ммм № 2

№	Товар	Количество	Цена	Сумма	Ставка НДС	Сумма НДС
1	Рога обыкновенные	5,000	3 000,00	15 000,00	18 %	2 700,00
2	Рога ветвистые	6,000	8 000,00	48 000,00	10 %	4 800,00
3	Рога обыкновенные	10,000	2 000,00	20 000,00	18 %	3 600,00

№	Услуга	Сумма	Сумма НДС	Всего
1	Доставка	2 000,00	360,00	2 360,00
2	Папировка	5 000,00	900,00	5 900,00

Сумма документа: 102 360,00
Сотрудник: Паниковский М.С.

Рис. 7.49

Обратите внимание на то, что печатная форма может быть сохранена в виде внешнего файла, причем может использоваться разный формат, конвертация в который произойдет автоматически.

8. 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. 8.1).

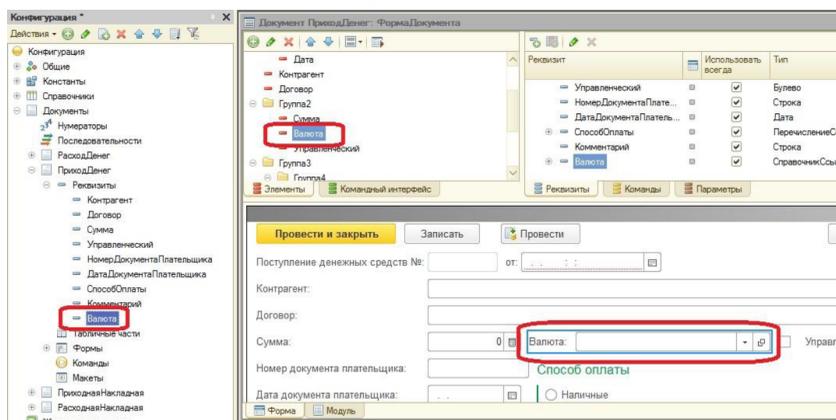


Рис. 8.1

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. 8.2), в которой через свойство "Хранение" устанавливаем связь с нашей константой, определяющей необходимость наличия валюты.

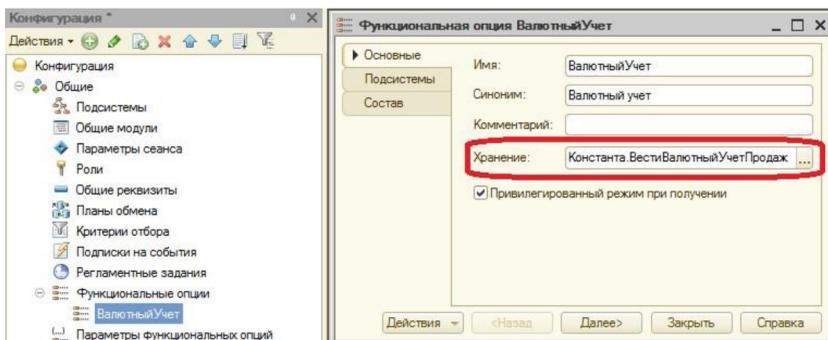


Рис. 8.2

В свойствах функциональной опции включим в ее состав справочник и созданные реквизиты (Рис. 8.3).

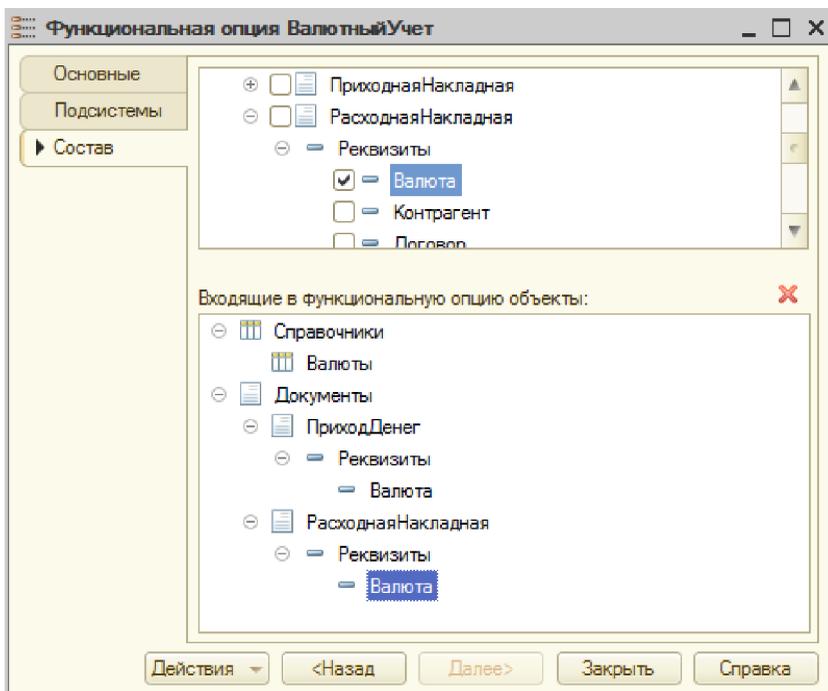


Рис. 8.3

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

По умолчанию значение константы *ВалютныйУчет* будет *Ложь* (флажок снят). Откроем документ *Приход/Деньг* и *РасходнаяНакладная*. В этом случае, соответствующие реквизиты будут скрыты (Рис. 8.4). Кроме того, справочник *Валюта* будет недоступен для работы пользователя.

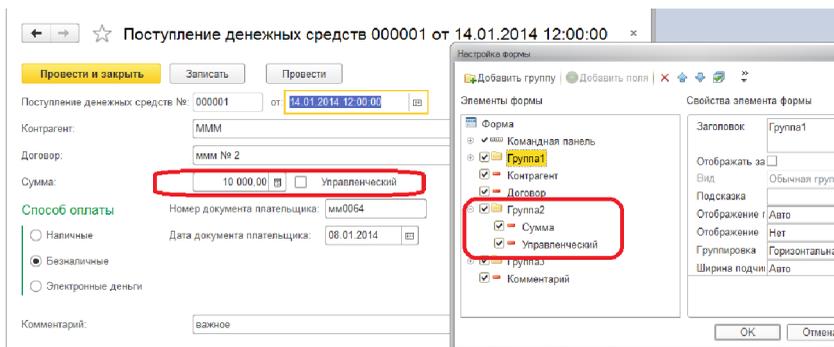


Рис. 8.4

Теперь установим в константе значение "Истина" и убедимся, что для пользователя стали доступны объекты, отвечающие за валютный учет (Рис.8.5).

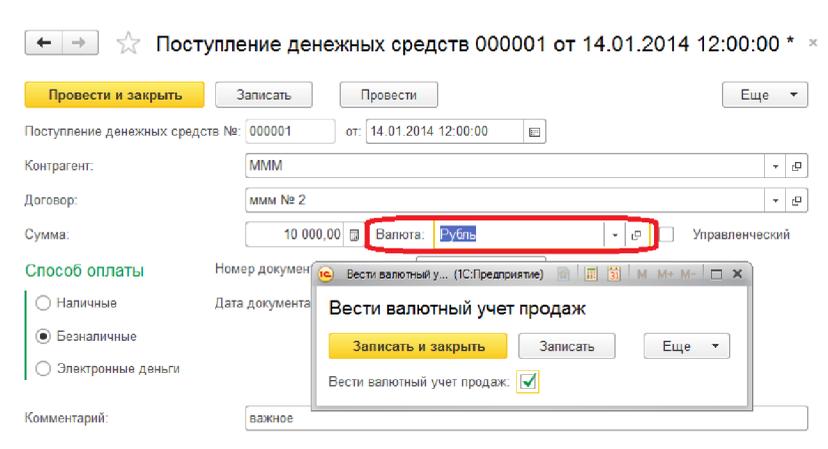


Рис. 8.5

9. Подсистемы и интерфейс

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

Создадим подсистемы и укажем их состав (Рис. 9.1 и Рис. 9.2). Следует учитывать, что все объекты должны входить в состав какой-либо подсистемы.

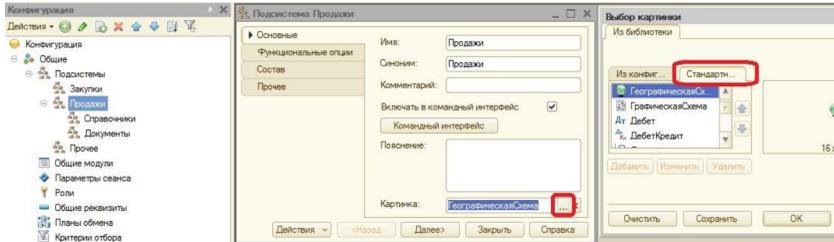


Рис. 9.1

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.

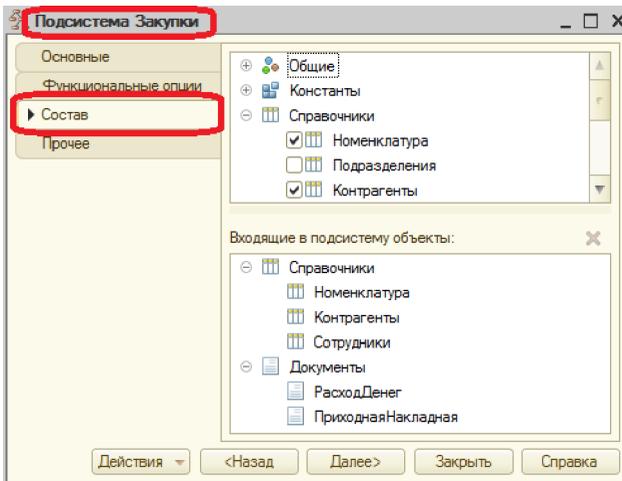


Рис. 9.2

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. 9.3).

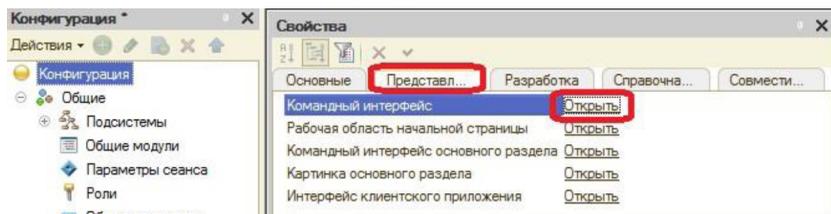


Рис. 9.3

С помощью стандартных приёмов работы заполним рабочую область начальной страницы в соответствии с Рис. 9.4.

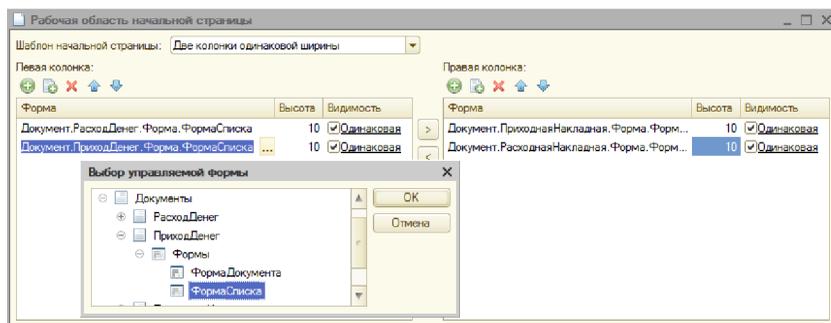


Рис. 9.4

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.9.5).

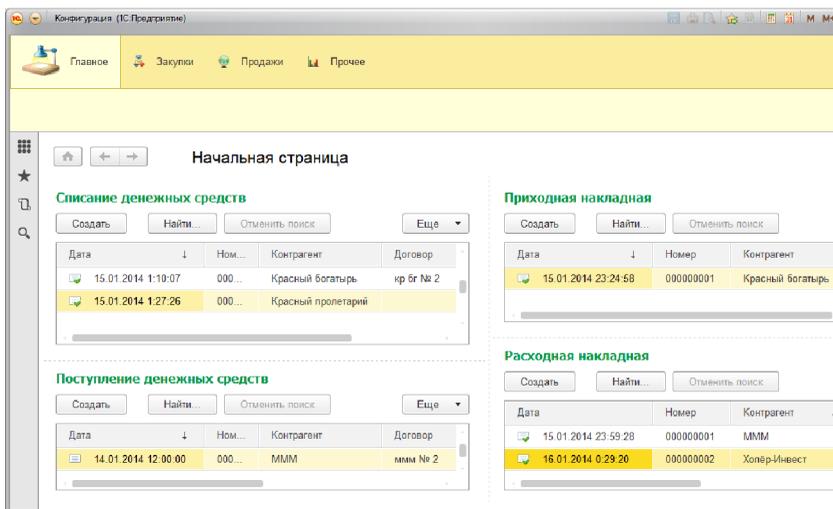


Рис. 9.5

Важной особенностью интерфейса является возможность его гибкой настройки пользователем. Причем можно настроить как правила работы интерфейса в целом (Рис. 9.6 и Рис. 9.7), так и особенности внешнего вида каждого раздела (Рис. 9.8).

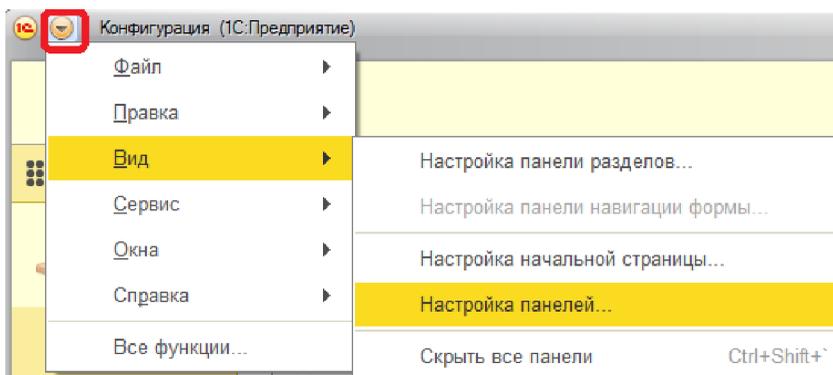


Рис. 9.6

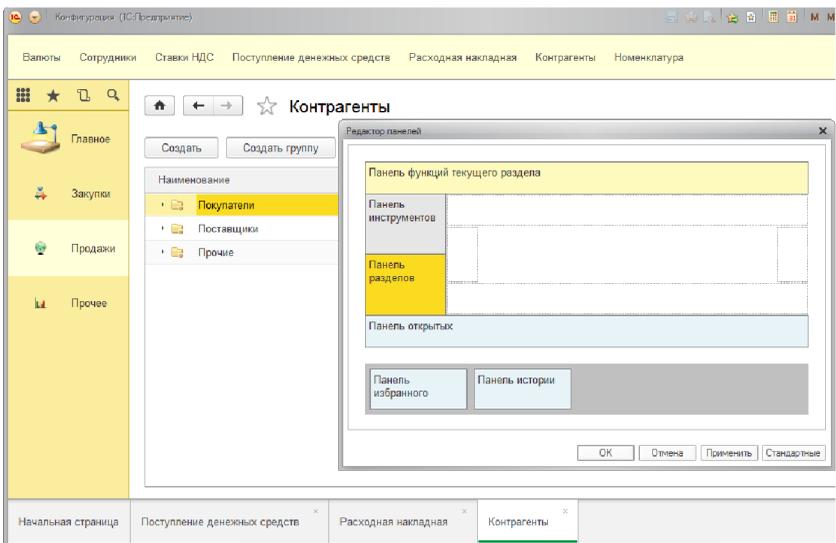


Рис. 9.7

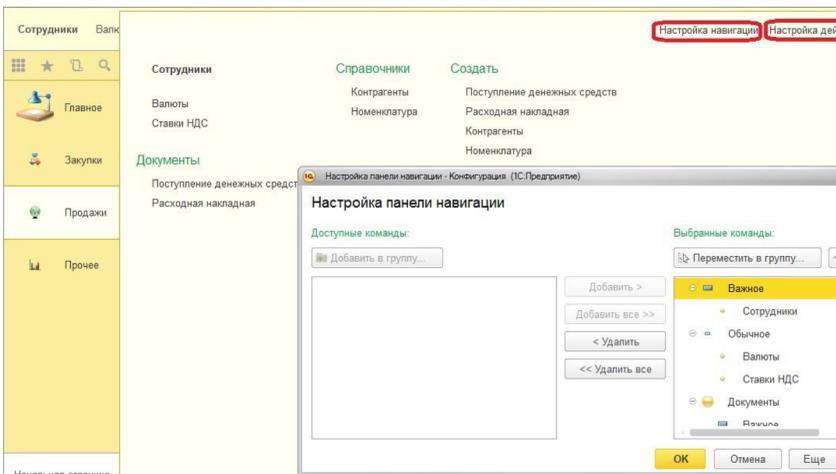


Рис. 9.8

10. 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.

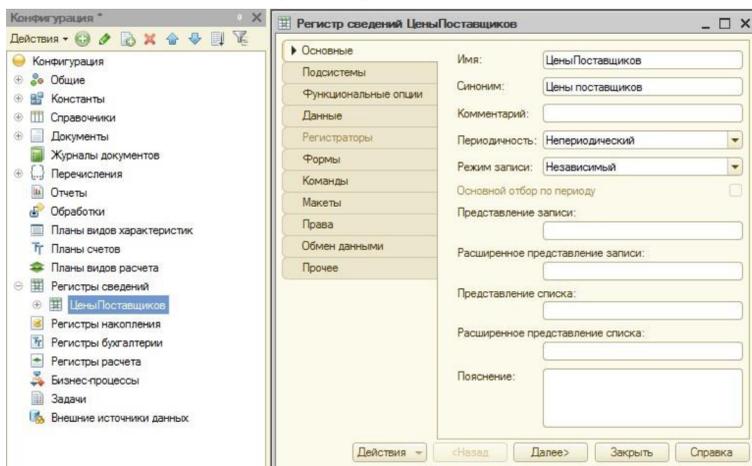


Рис. 10.1

Создание нового регистра аналогично созданию нового документа или справочника (Рис. 10.1). 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. 10.2).

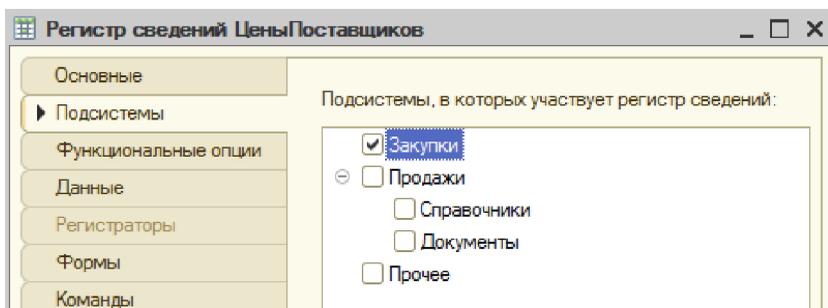


Рис. 10.2

Структуру регистра можно задать на закладке *Данные*. (Рис. 10.3) 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.

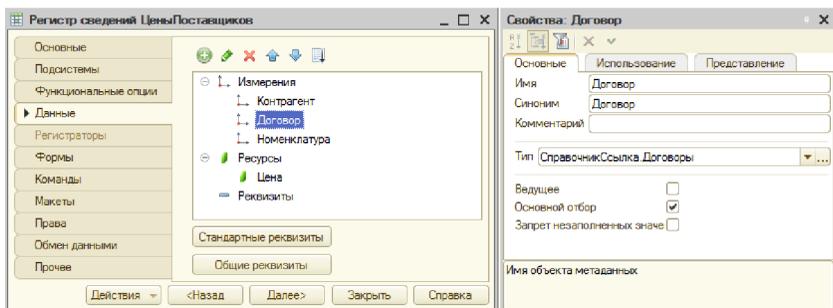


Рис. 10.3

Информацию в регистр пользователь может внести самостоятельно в режиме "1С:Предприятие", используя соответствующую форму списка (Рис.10.4).

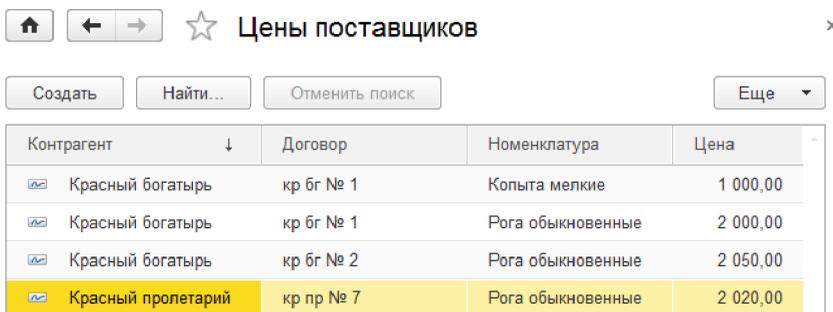


Рис. 10.4

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.10.5).

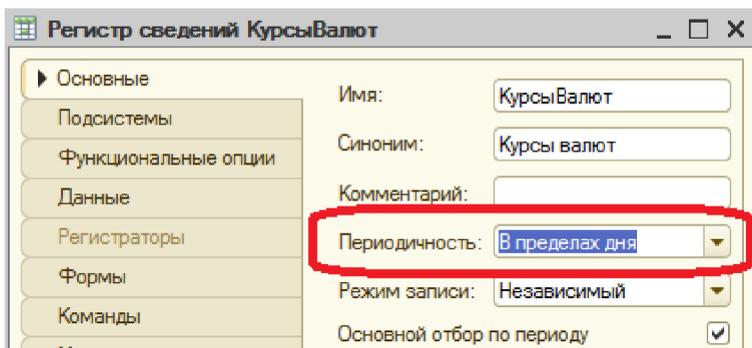


Рис. 10.5

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. 10.6).

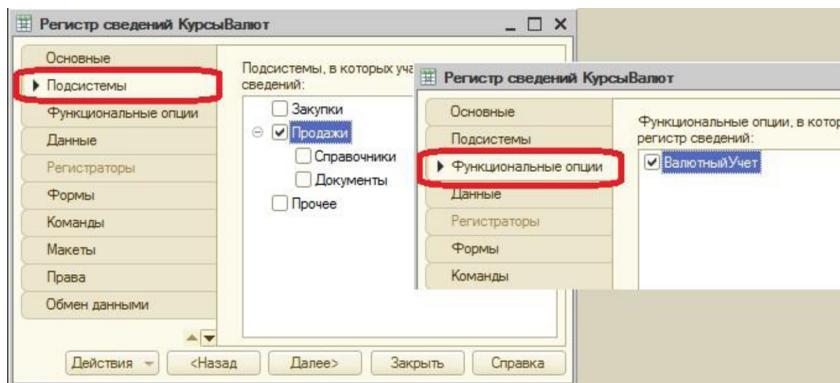


Рис. 10.6

Сама структура регистра будет состоять из 1 измерения и 1 ресурса (Рис. 10.7). Поскольку наш регистр является периодическим, то наряду с существующими измерениями добавляется еще одно *Период* (Рис. 10.8), и система предоставляет возможность быстро извлечь интересующую нас информацию на произвольный момент времени.

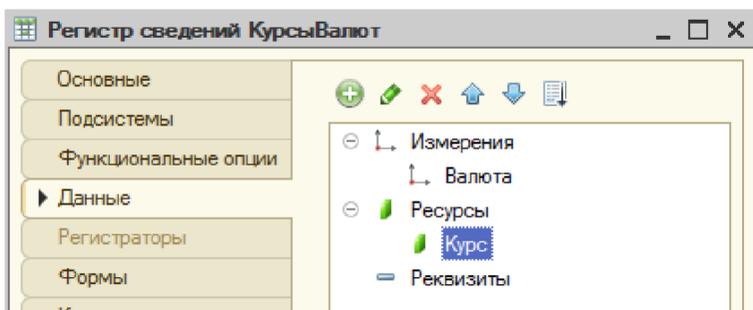


Рис. 10.7

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. 10.8

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 уж не соответствует положению вещей. Для

ешения подобно городам 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 "Сведения О Сотрудниках". 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. 10.9).

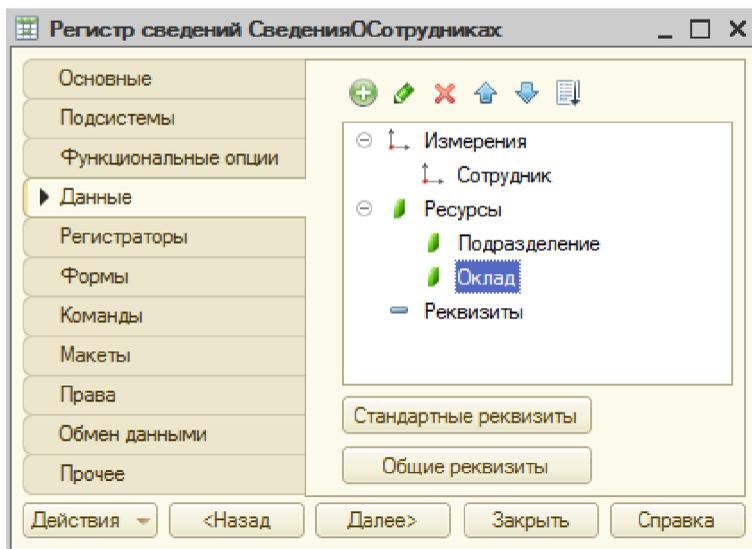


Рис. 10.9

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. 10.10). В этом случае у регистра появится дополнительное поле "Регистратор", в котором будет указан документ, сформировавший эти записи.

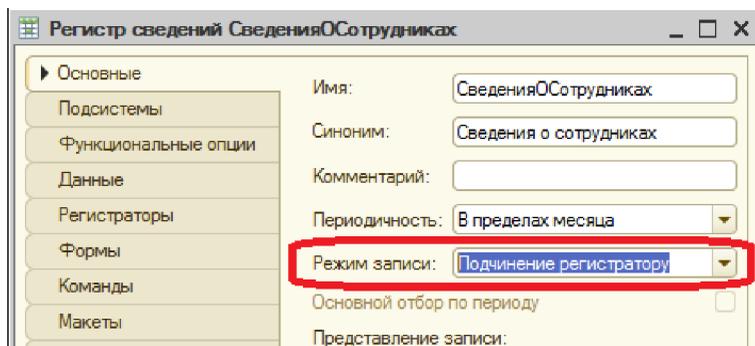


Рис. 10.10

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. 10.11).

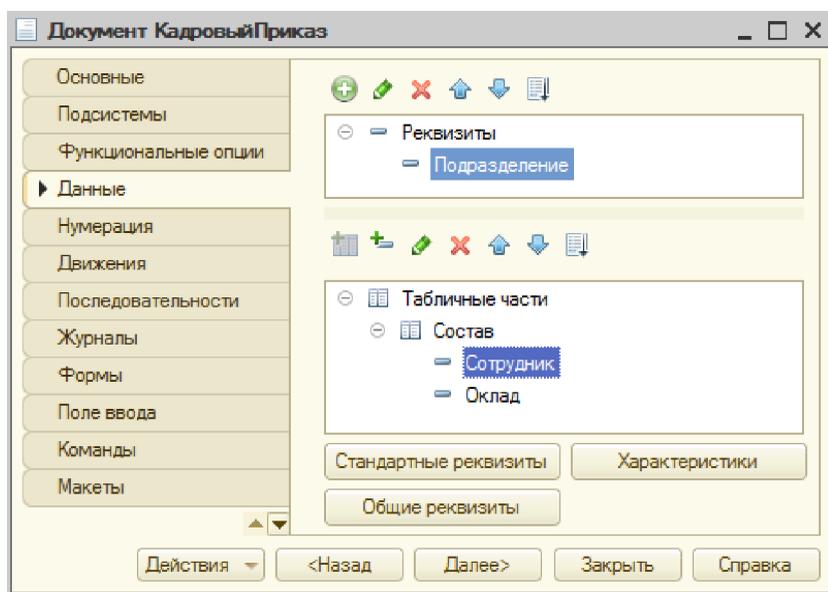


Рис. 10.11

Чтобы документ формировал записи в регистр необходимо на закладке *Движения* настроить следующие свойства (Рис. 10.12):

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.

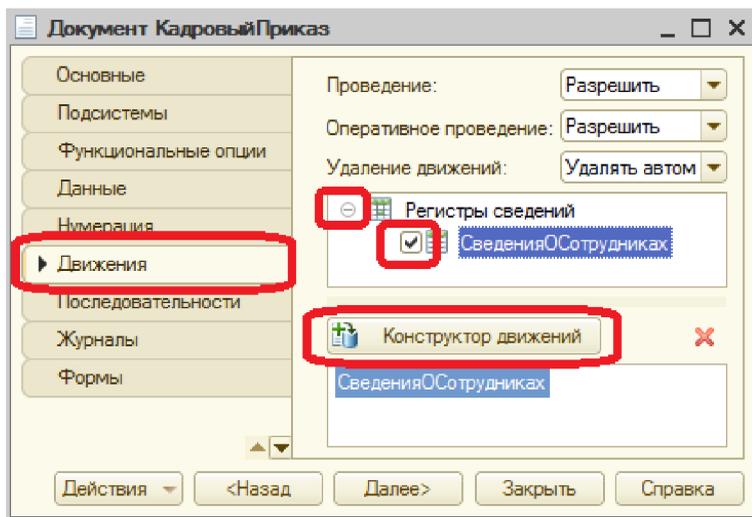


Рис. 10.12

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. 10.13) сопоставить данные документа регистра.

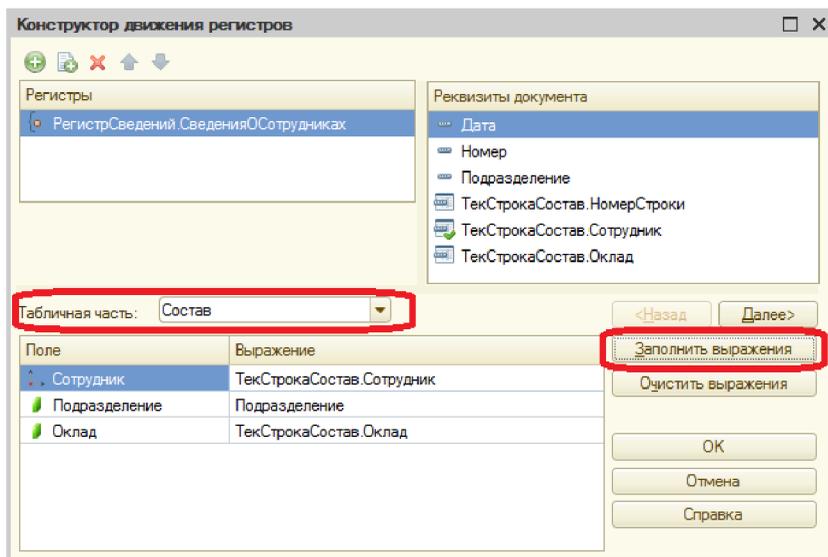


Рис. 10.13

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

В режиме "1С:Предприятие" у пользователя появляется

возможность провести или снять с проведения проведенный документ (Рис. 10.14).

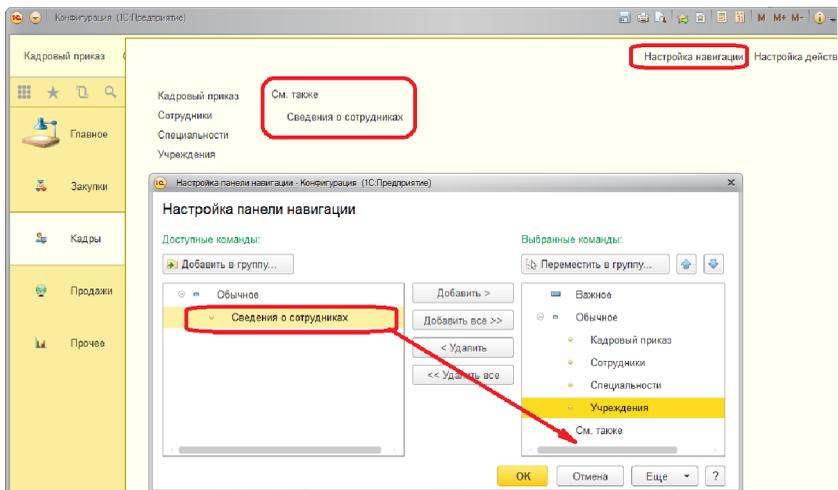


Рис. 10.14

Обратите внимание, по умолчанию наш регистр не отобразился автоматически в интерфейсе. Чтобы посмотреть результат наших действий, самостоятельно добавьте регистр в интерфейс (Рис. 10.15).

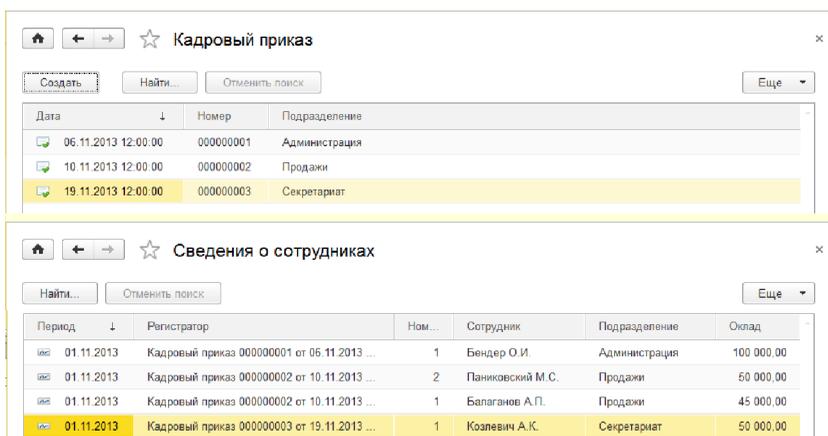


Рис. 10.15

11. 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. 11.1):

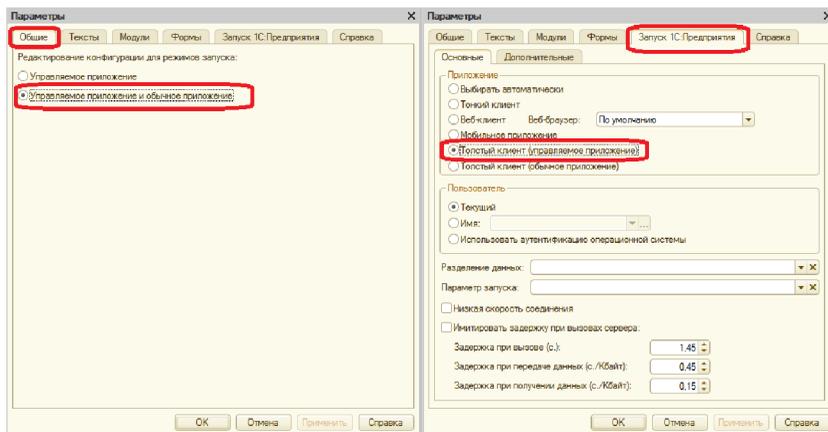


Рис. 11.1

Только после этих настроек в свойствах конфигурации станет доступной следующая нужная нам настройка (Рис. 11.2).

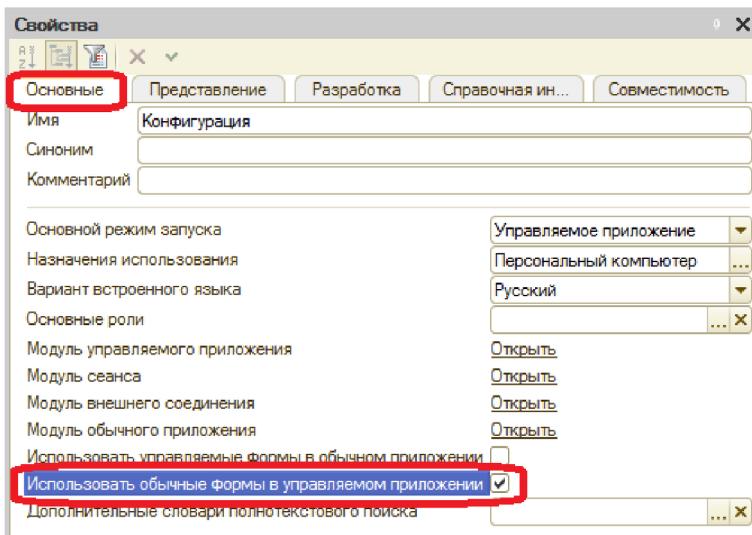


Рис. 11.2

Откроем обработку в режиме "1С:Предприятие" и создадим новый запрос, текст которого удобнее формировать с помощью конструктора запросов (Рис. 11.3). Вызов конструктора происходит через контекстное меню, вызываемое по правой кнопке мыши. С помощью этого запроса мы попробуем узнать, какие у нас есть поставщики.

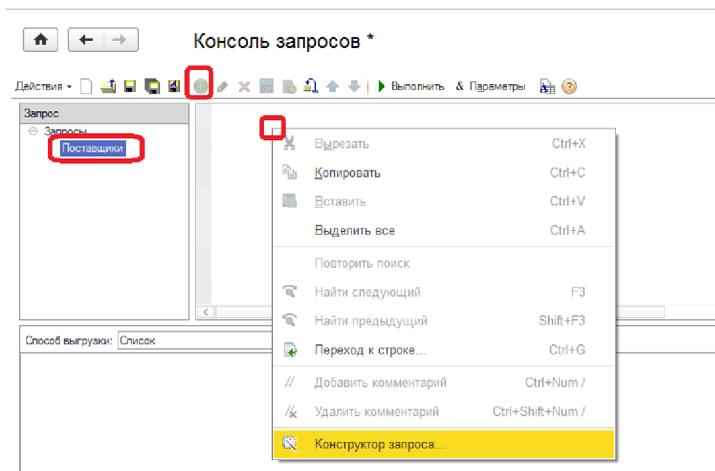


Рис. 11.3

В конструкторе запроса мы, работая на уровне таблиц, определяем откуда и какая именно информация нас интересует (Рис. 11.4).

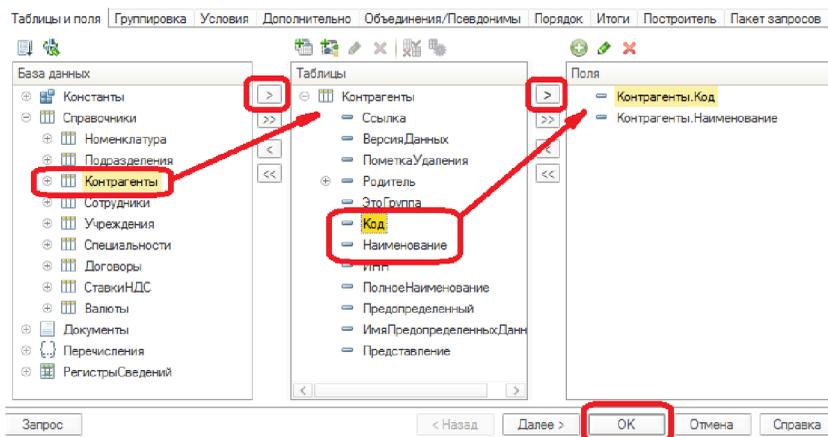


Рис. 11.4

Результатом работы конструктора будет являться текст запроса, запустив на исполнение который с помощью кнопки "Выполнить", мы сможем увидеть полученную этим запросом информацию (Рис. 11.5)

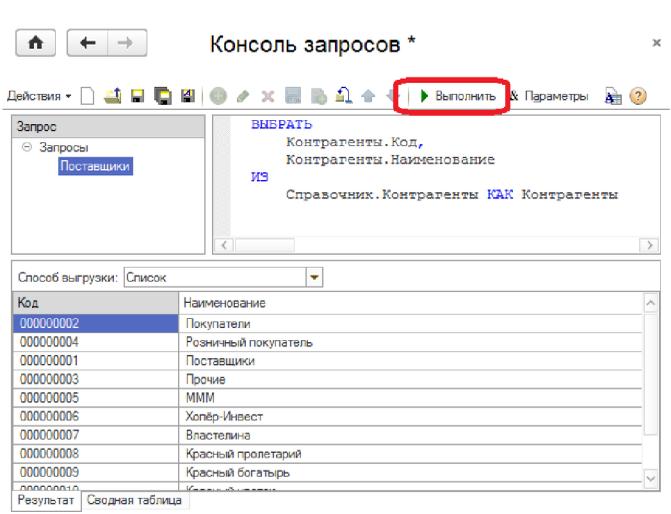


Рис. 11.5

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.11.6).

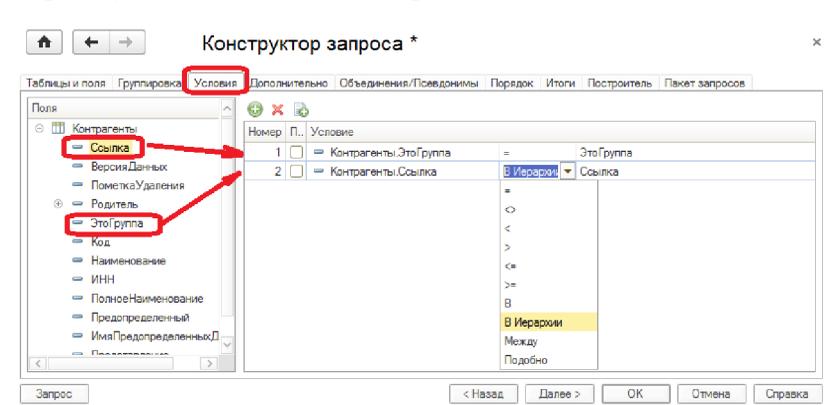


Рис. 11.6

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.11.7).

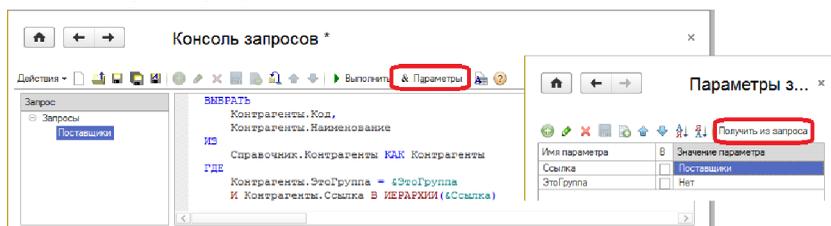


Рис. 11.7

Установив параметры, выполним наш запрос и увидим

искомый список поставщиков (Рис. 11.8). Таким образом, с помощью запроса мы можем извлечь информацию из базы, отфильтровав с помощью различных условий только то, что нам требуется.

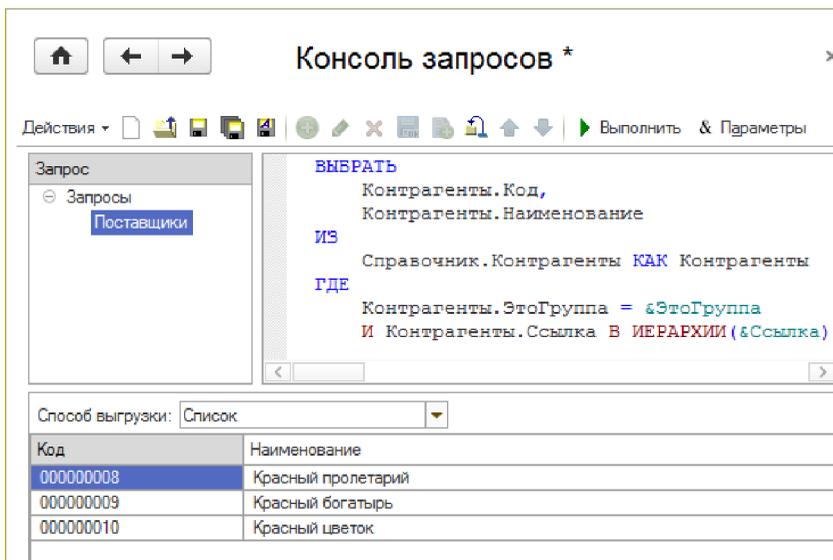


Рис. 11.8

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. 11.9).

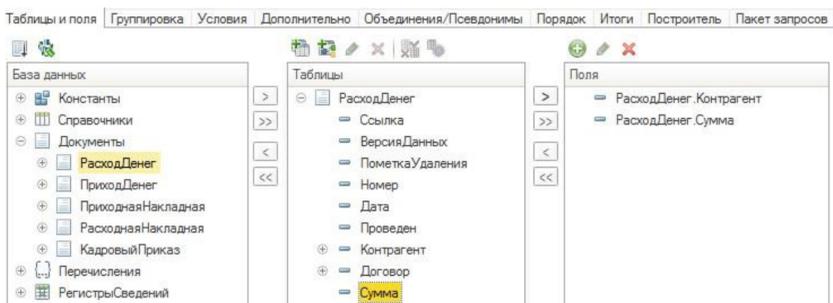


Рис. 11.9

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. 11.10).

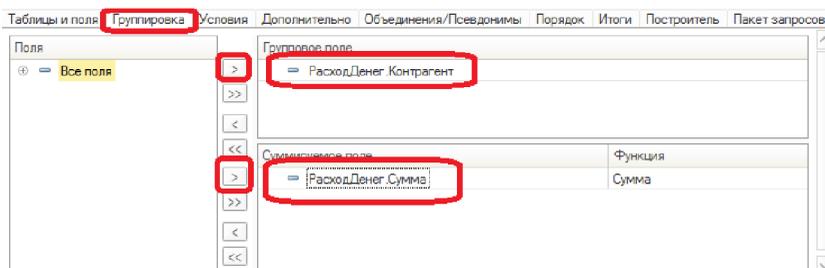


Рис. 11.10

С помощью этого механизма нам стали доступны не просто исходная информация из базы, но и расчетные данные (Рис. 11.11).

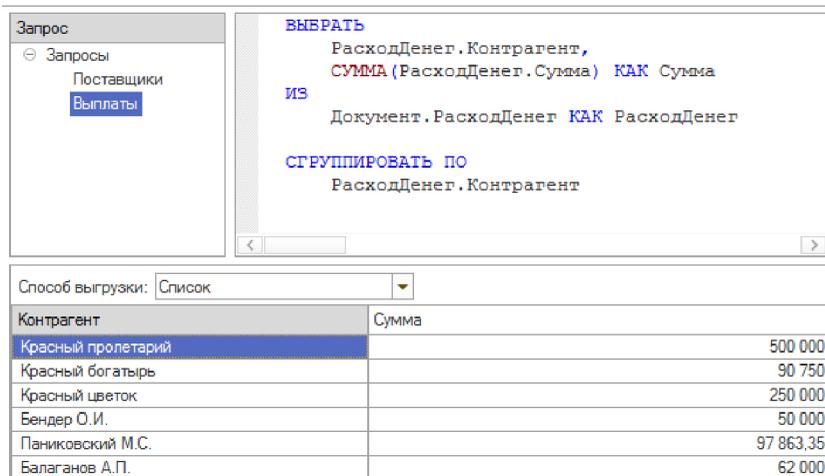


Рис. 11.11

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 получаются 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. 11.12).

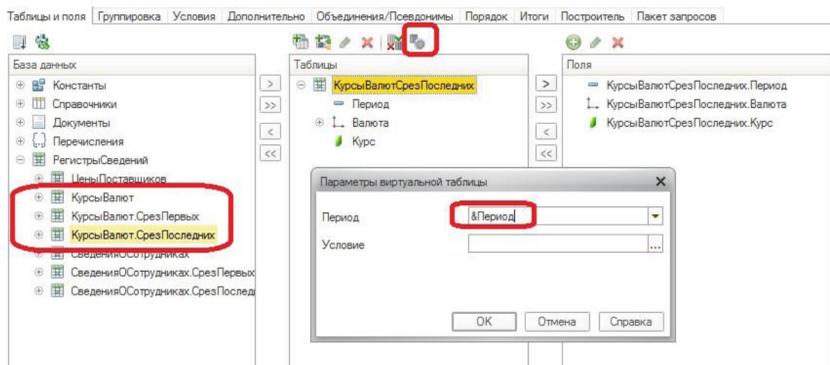


Рис. 11.12

Результат выполнения данного запроса показывает курсы валют на 17.01.2014, хотя именно на 17 число курсы валют у нас не хранятся (Рис.11.13).

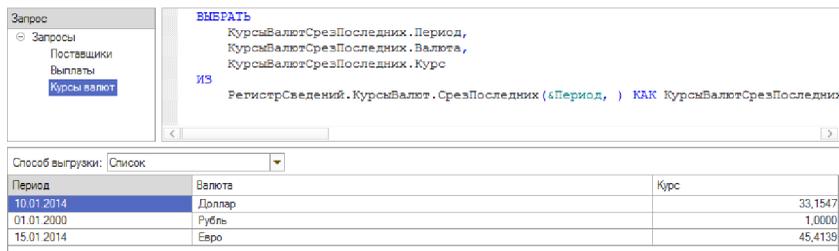


Рис. 11.13

Следующий пример будет более сложным. Нам 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. 11.14).

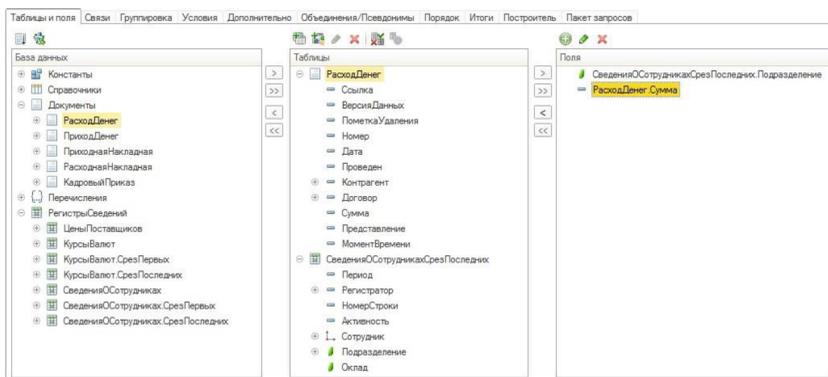


Рис. 11.14

Сопоставление получателя денег в документе с сотрудником, работающим в подразделении, произведем через механизм соединения на закладке "Связи" (Рис. 11.15).

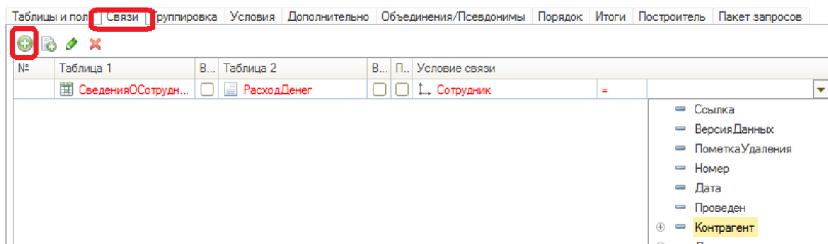


Рис. 11.15

Применим группировку, чтобы получить общую сумму выплат (Рис. 11.16).

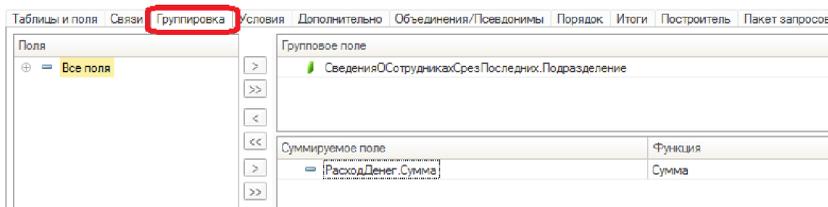


Рис. 11.16

После чего проверяем результат (Рис. 11.17).

Скриншот интерфейса программы, отображающий результат запроса. В левой панели меню выбран пункт «Выплаты подраздел». В основной области отображены SQL-подобные команды: «ВЫБРАТЬ», «СУММА (РасходДенег.Сумма) КАК Сумма», «ИЗ», «РегистрСведений.СведенияОСотрудникахСрезПоследних КАК СведенияОСотрудникахСрезПоследних», «ВНУТРЕННЕ СОЕДИНИЛИ Документ.РасходДенег КАК РасходДенег», «ПО СведенияОСотрудникахСрезПоследних.Сотрудник = РасходДенег.Контрагент» и «ГРУППИРОВАТЬ ПО СведенияОСотрудникахСрезПоследних.Подразделение». В нижней части экрана отображена таблица с двумя столбцами: «Подразделение» и «Сумма».

Подразделение	Сумма
Администрация	50 000
Продажи	159 863,95

Рис. 11.17

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".

12. 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. 12.1), 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.

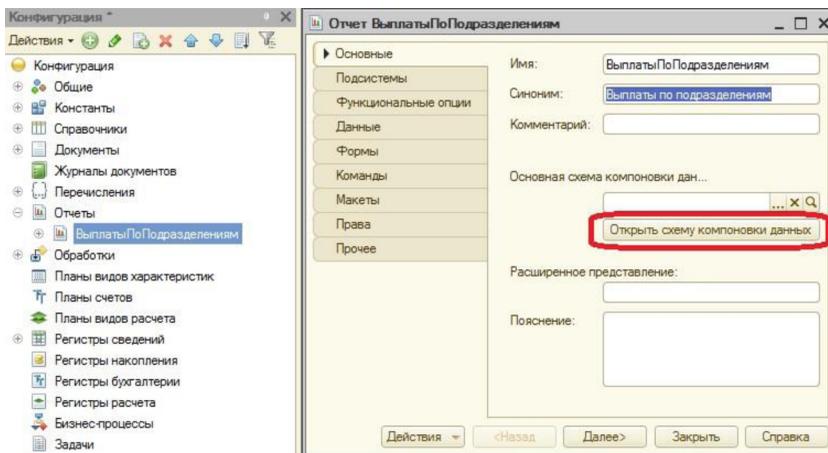


Рис. 12.1

Создадим с помощью конструктора новую схему компоновки данных (Рис. 12.2)

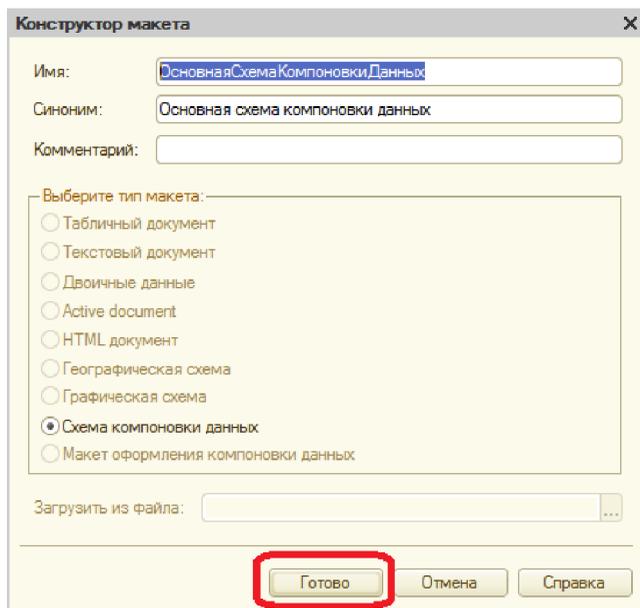


Рис. 12.2

Добавим новый набор данных, получаемый помощью запроса (Рис.12.3)

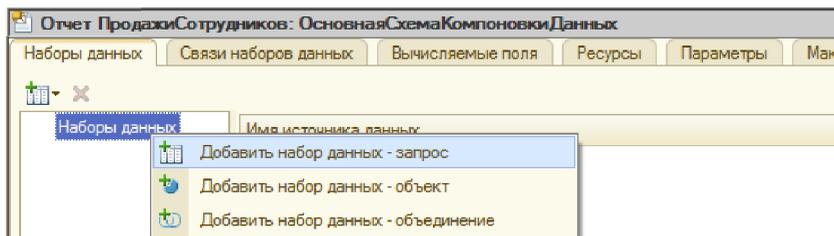


Рис. 12.3

С помощью конструктора запросов сформируем запрос (Рис. 12.4). После этого с помощью компоновки данных настроим правила отображения этого запроса (Рис.12.5 и Рис. 12.6). Обратите внимание, что добавляя новую группировку, мы ничего не изменяем, а сразу же нажимаем кнопку "ОК" (Рис.12.5).

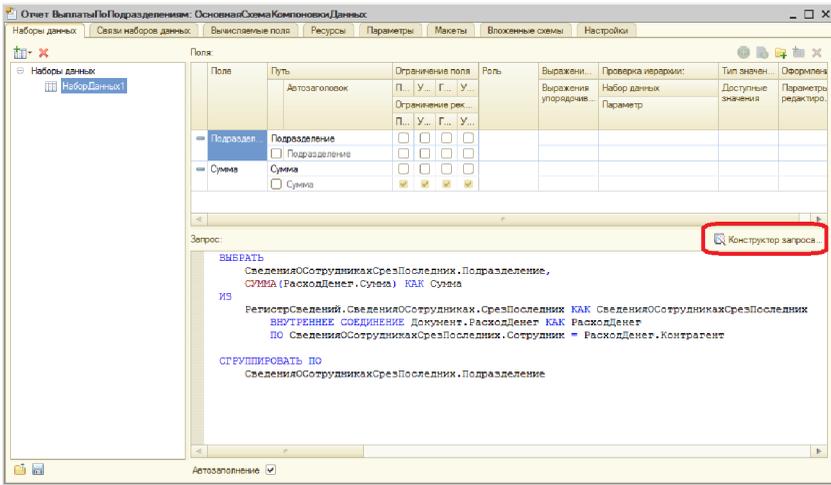


Рис. 12.4

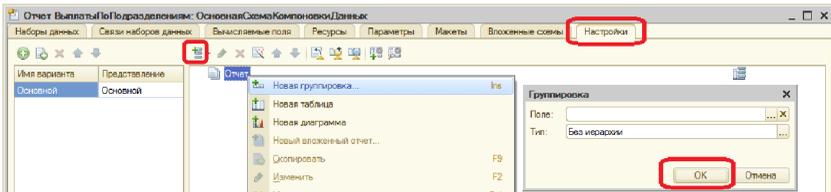


Рис. 12.5

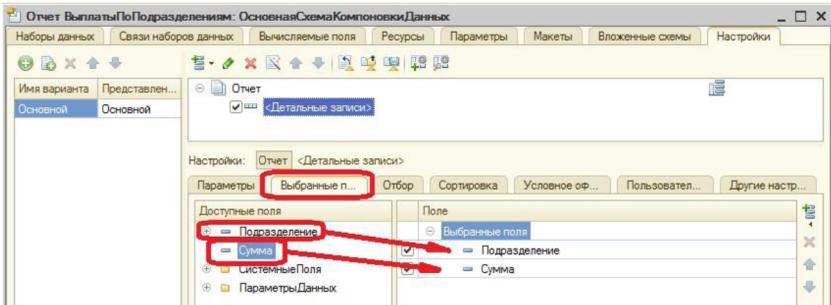


Рис. 12.6

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. 12.7).

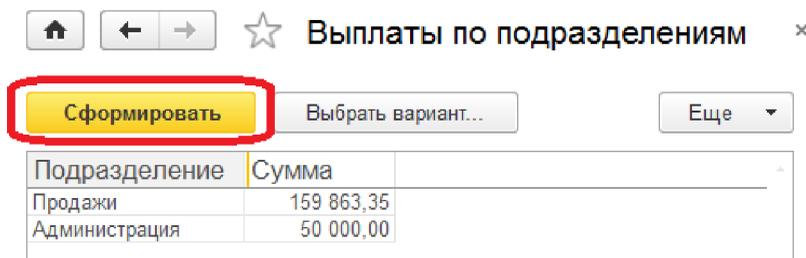


Рис. 12.7

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 "ПродажиСотрудников". Описанным выше способом воспользуемся механизмом компоновки и запросом соберём данные о продажах (Рис.12.8).

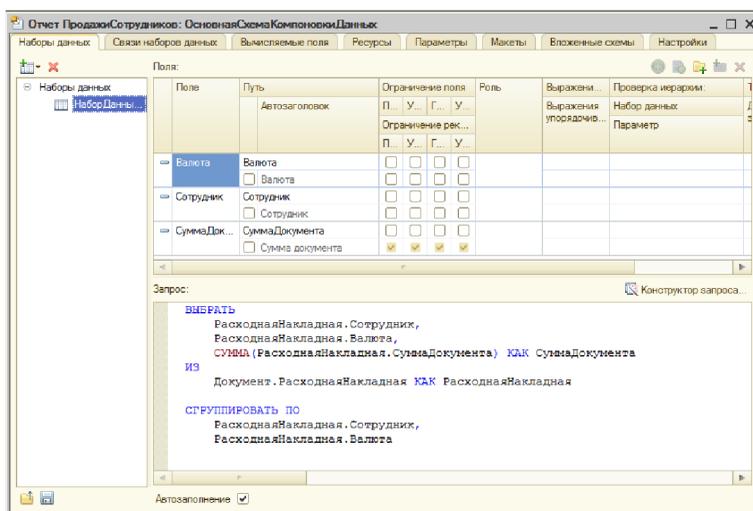


Рис. 12.8

На закладке "Ресурсы" определим поле "СуммаДокумента" как ресурс (Рис. 12.9)

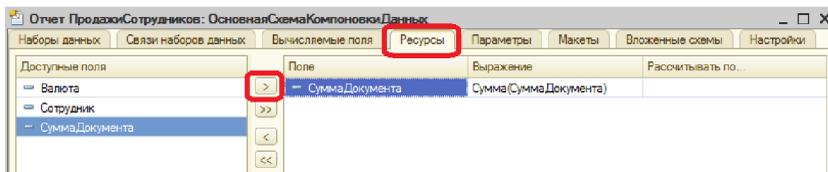


Рис. 12.9

После этого перейдем на закладку "Настройки", где определим группировки и выбранные поля (Рис. 12.10). В отличие от предыдущего отчета, здесь нам необходимо будет явным образом указать значения группировок.

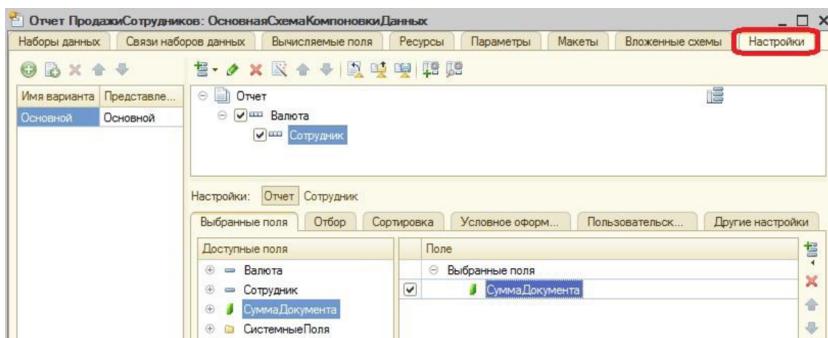


Рис. 12.10

Результат наших действий позволит получить пользователю сводные данные о продажах сотрудников в разрезе валют (Рис. 12.11). 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.



Сформировать

Выбрать вариант...

	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. 12.11

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.12.12).

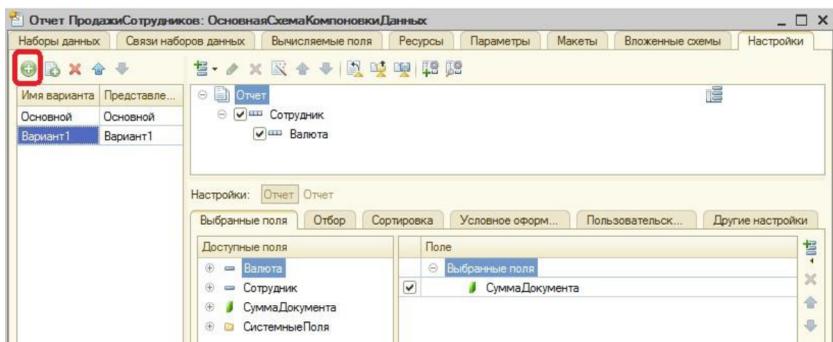


Рис. 12.12

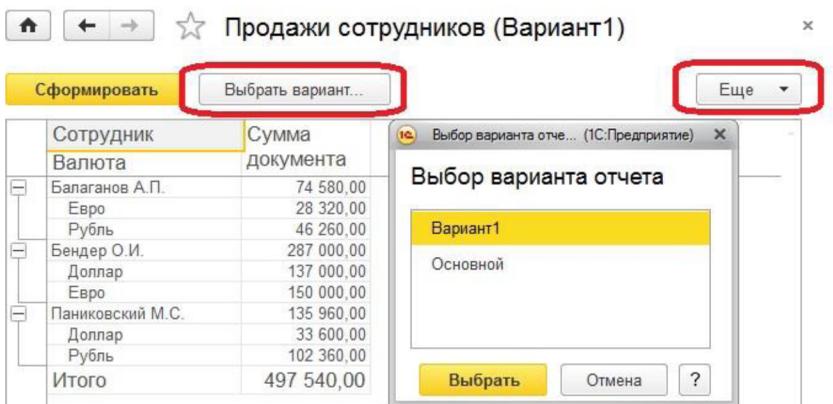


Рис. 12.13

Теперь в пользовательском режиме для нашего отчета появляется новая возможность выбора варианта (Рис. 12.13) 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. 12.14).

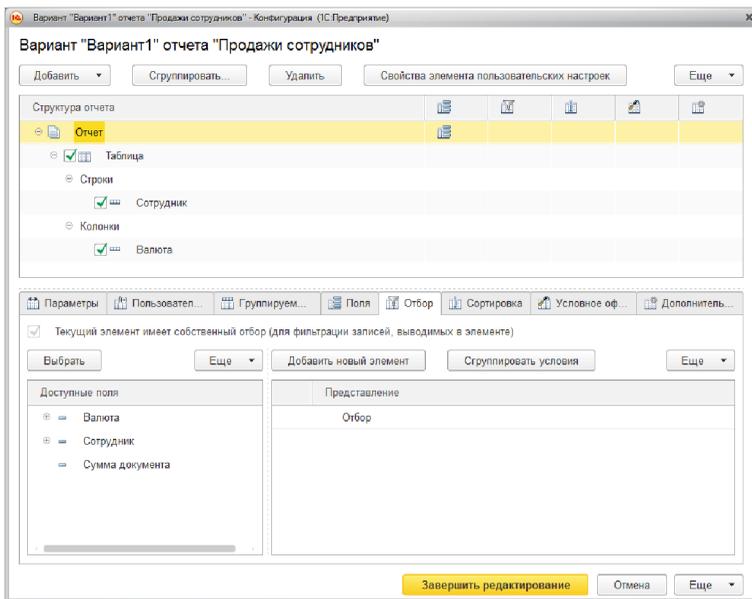


Рис. 12.14

Эти варианты могут быть позднее могут быть сохранены для дальнейшего использования (Рис. 12.15). 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. 12.14, можно отобразить те же самые данные о продажах в виде кросс-таблицы, что может иногда быть очень удобно.



Рис. 12.15

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. 12.16).

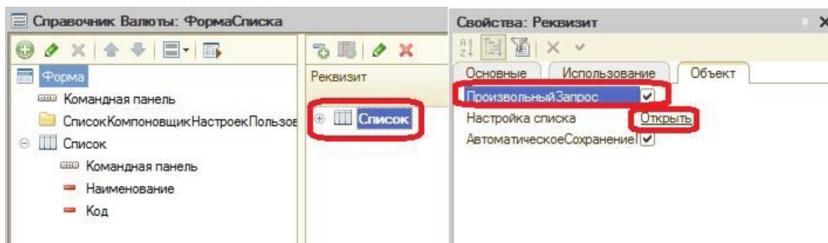


Рис. 12.16

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. 12.17) мы можем решить эту задачу.

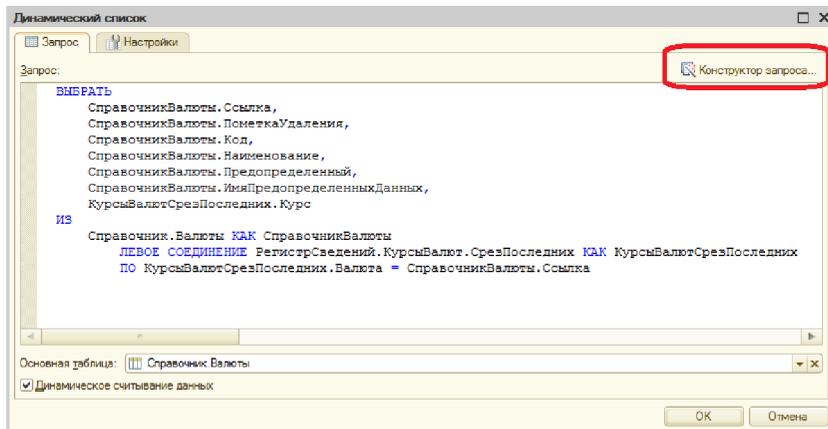


Рис. 12.17

Получив данные запросом, изменяем саму форму, добавив туда новые данные (Рис. 12.18)

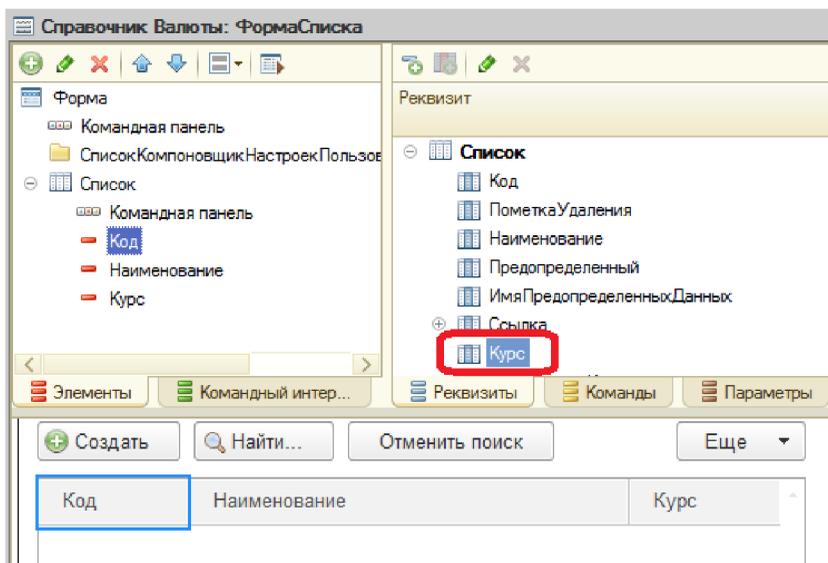


Рис. 12.18

Осталось проверить результат в пользовательском режиме (рис. 12.19)

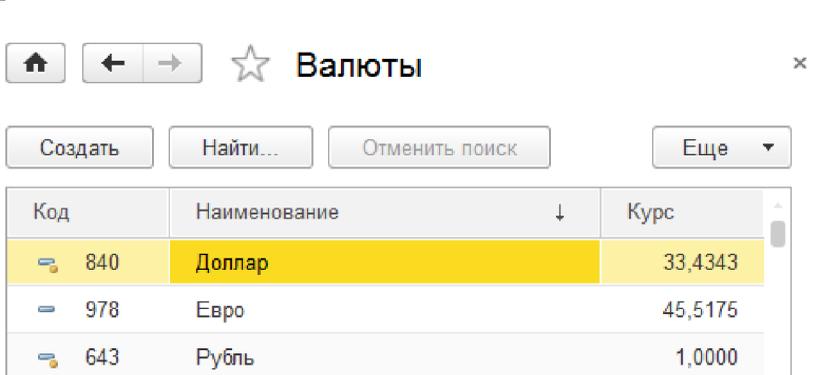


Рис. 12.19

Библиографический список

1. Официальный сайт продукта 1С:Предприятие 8 и прикладных решений на его основе [Электронный ресурс]. Режим доступа: <https://v8.1c.ru/>, свободный. — Загл. с экрана
2. Официальный сайт информационно-технологического сопровождения продукта 1С:Предприятия [Электронный ресурс]. Режим доступа: <https://portal.1c.ru/>, свободный. — Загл. с экрана
3. Официальный сайт информационной системы 1С:ИТС [Электронный ресурс]. Режим доступа: <https://its.1c.ru/>, свободный. — Загл. с экрана
4. Официальный сайт фирмы 1С [Электронный ресурс]. Режим доступа: <http://1c.ru/>, свободный. — Загл. с экрана
5. Группа компаний «Стэк» [Электронный ресурс]. Режим доступа: <http://www.stack03.ru/>, свободный. — Загл. с экрана
6. Гладкий А. 1С 8 с нуля: комплексное руководство для начинающих / А.Гладкий, А. Спивак. — Москва : Феникс, 2011. — 478 с.
7. Гладкий А. 1С:Управление торговлей 8.2 с нуля. Конфигурирование и настройка. 75 уроков для начинающих / А.Гладкий, Е. Кондукова. — Санкт-Петербург : БХВ, 2012. — 240 с.
8. Филатова В.О. 1С: Предприятие 8.1 Бухгалтерия предприятия. Управление торговлей. Управление персоналом / А.А.Гладкий, Е. Кондукова. — Санкт-Петербург : БХВ, 2010. — 288 с.

Оглавление

Предисловие	3
1. Общие положения	5
1.1. Задача	5
1.2. Создание новой информационной базы	8
2. Константы	16
3. Справочники	24
3.1. Общее описание и назначение	24
3.2. Иерархия	28
3.3. Предопределенные элементы	32
3.4. Реквизиты справочника	34
3.5. Табличная часть	40
3.6. Подчинение	44
3.7. Форма. Форма списка, форма элемента	46
4. Перечисление	52
5. Элементы управления формы	55
5.1. Поле	57
5.1.1. Поле ввода	57
5.1.2. Поле флажка	59
5.1.3. Поле переключателя	61
5.2. Команда	63
5.3. Группа	65
5.3.1. Обычная группа	65
5.3.2. Командная панель	66
5.3.3. Группа страниц	66
5.4. Таблица	68
6. Обработчик события	69
7. Документ	72

7.1.	Реквизитышапки	74
7.2.	Формыдокумента.....	77
7.3.	Реквизиты табличнойчасти.....	88
7.4.	Печать документа. Конструкторпечати.....	99
8.	Функциональные опции.....	108
9.	Подсистемы и интерфейс.....	111
10.	Регистр сведений	115
11.	Запрос	127
12.	Отчет. Компоновка данных	136
	Библиографический список.....	146

Учебное издание

Татьяна Саяновна Балданова
Оюна Анатольевна Лобсанова

**ЗНАКОМСТВО С ПЛАТФОРМОЙ
1С: ПРЕДПРИЯТИЕ 8**

Учебно-методическое пособие

Компьютерная верстка О. А. Лобсанова

Свидетельство о государственной аккредитации
№ 2670 от 11 августа 2017 г.

Подписано в печать 18.11.19. Формат 60x84 1/16.
Усл. печ. л. 8,66. Уч.-изд. л. 5,32. Заказ 195

Издательство Бурятского госуниверситета
670000, г. Улан-Удэ, ул. Смолина, 24а
E-mail: riobsu@gmail.com