

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Пензенский государственный аграрный университет»

Кафедра «Прикладная информатика»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по проведению практических занятий
по дисциплине
«Разработка программных приложений»
направления подготовки
«09.03.03 "Прикладная информатика»

Курс 3
семестр 5

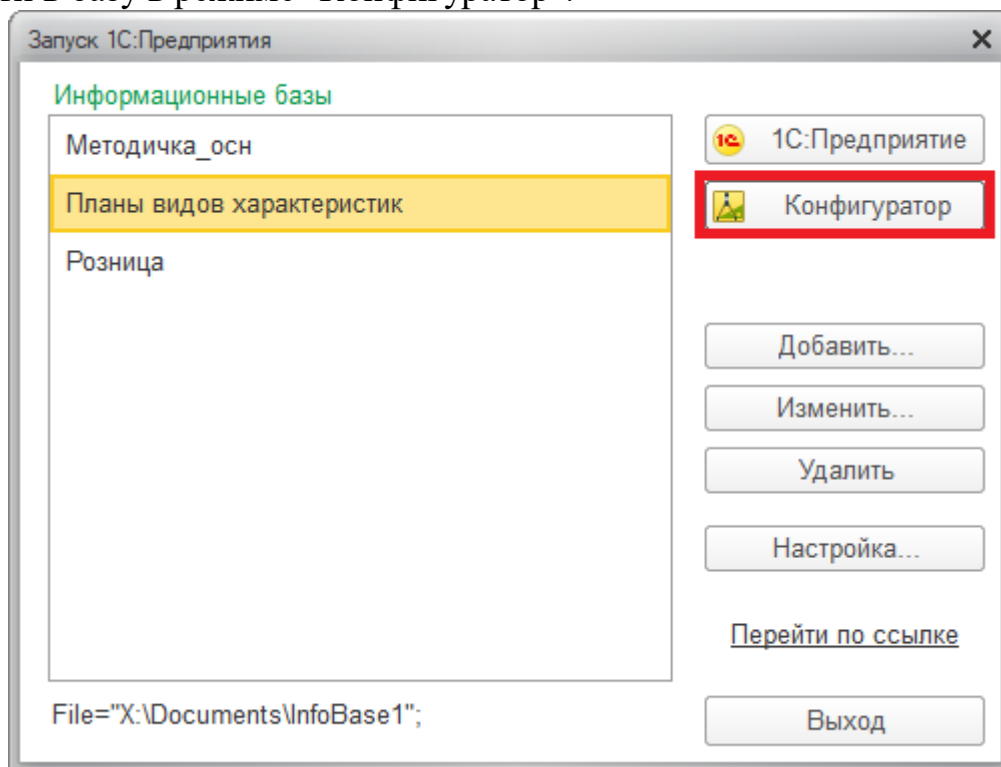
Пенза 2026 г.

ВЫГРУЗКА И ЗАГРУЗКА ИНФОРМАЦИОННОЙ БАЗЫ

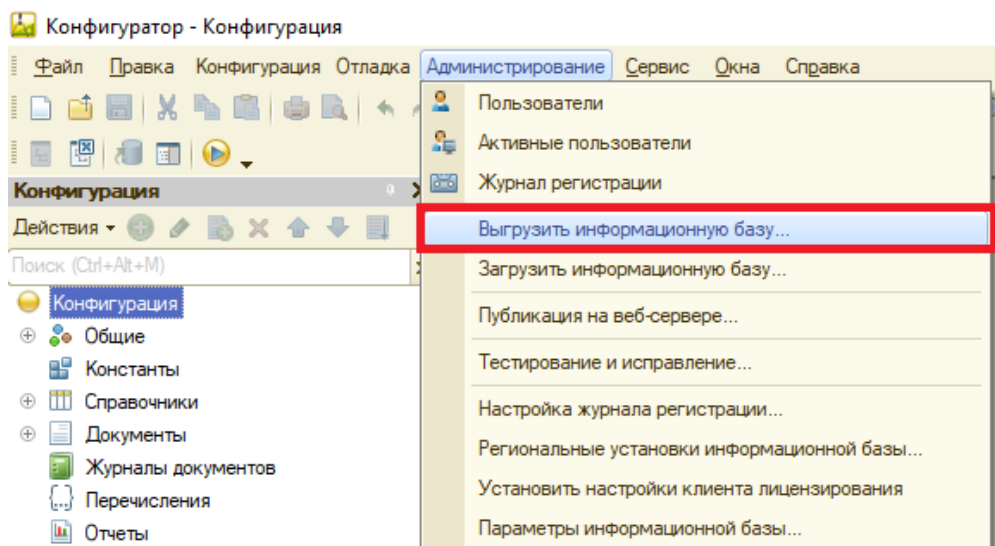
Выгрузка информационной базы

При завершении изменений и работой с информационной базой 1С ее необходимо сохранить (выгрузить), для чего при условии выхода из 1С необходимо запустить программу 1С.

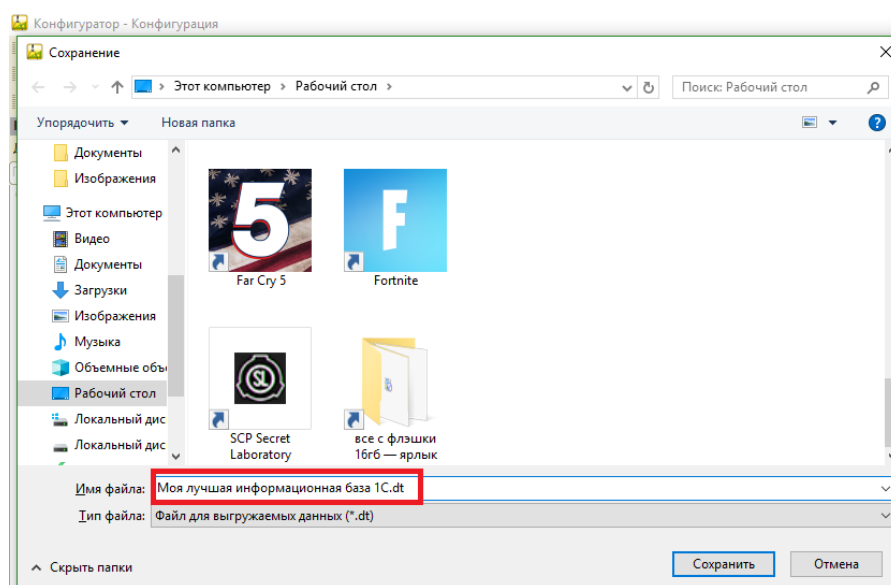
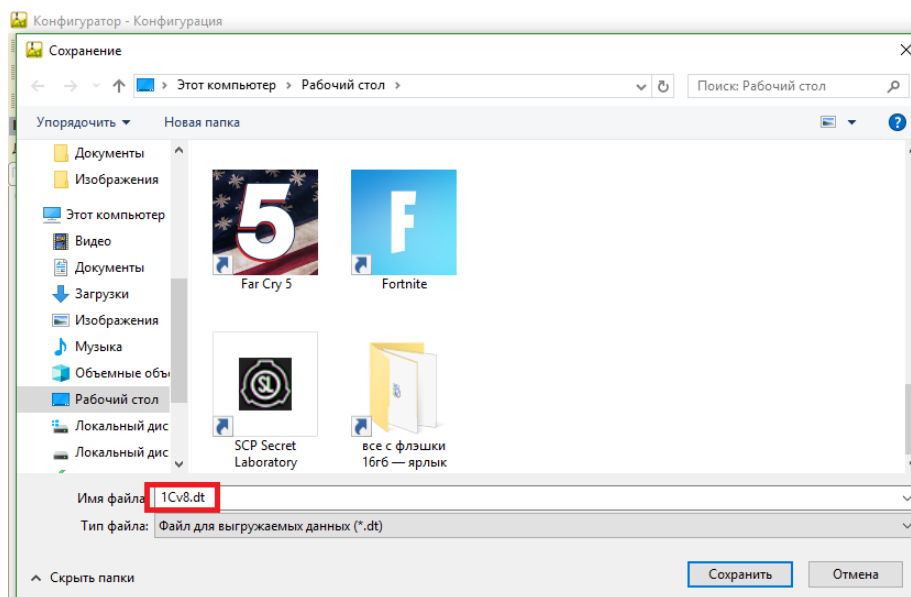
Войти в базу в режиме "Конфигуратор".



В меню программы выбрать пункт "Администрирование", подпункт "Выгрузить информационную базу".

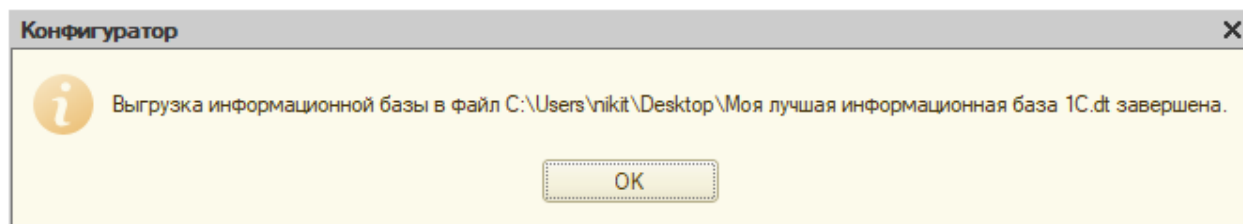


Указать место, куда произойдёт выгрузка архива. Например, "Рабочий стол". Также, можно задать своё имя для архива. По умолчанию, она задаётся как "1Cv8".



Нажимаем кнопку "Сохранить".

База сохранилась в файл на Рабочем столе.

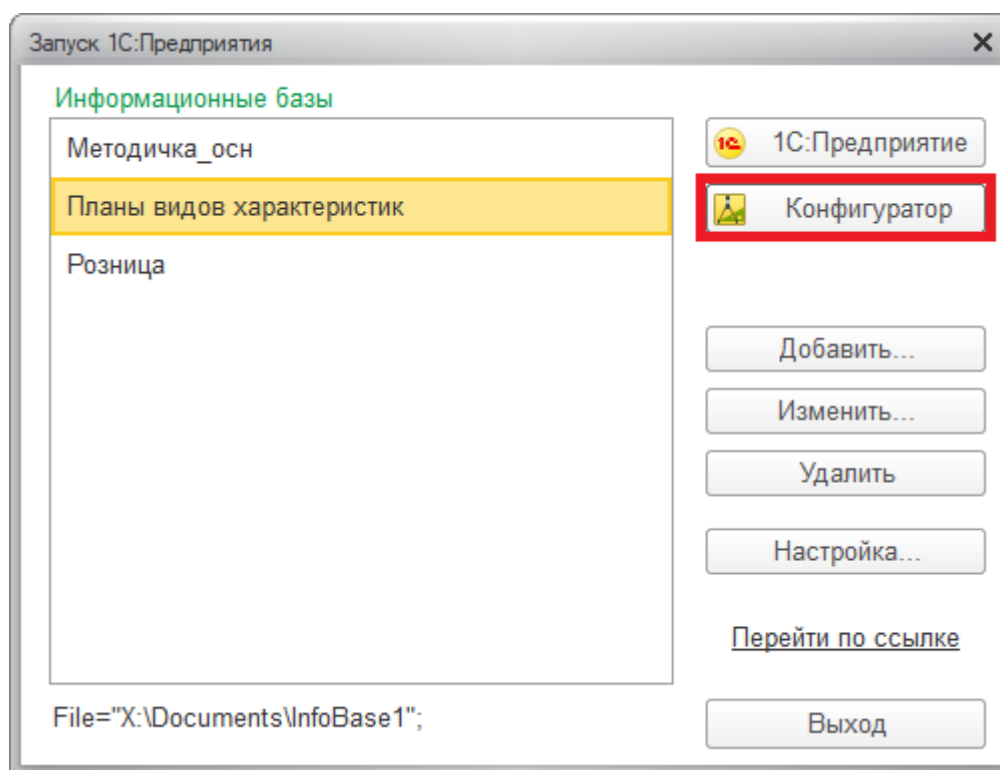


При необходимости, в будущем, Вы сможете им воспользоваться этим архивом для восстановления данных. Как это сделать - смотрите ниже.

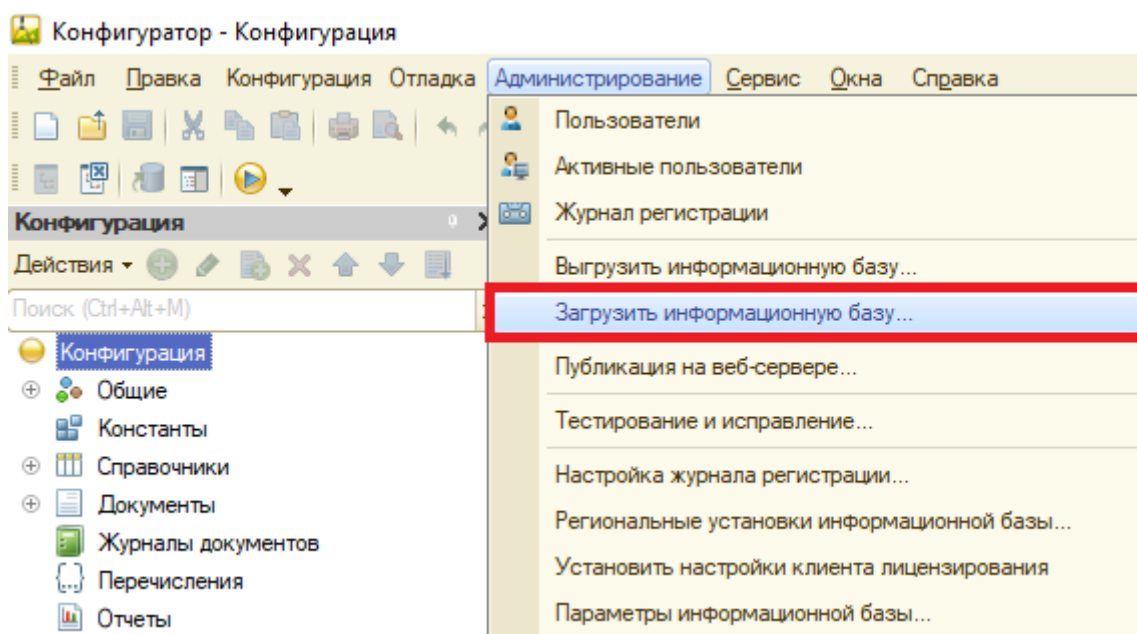
Загрузка базы

Запускаем программу 1С.

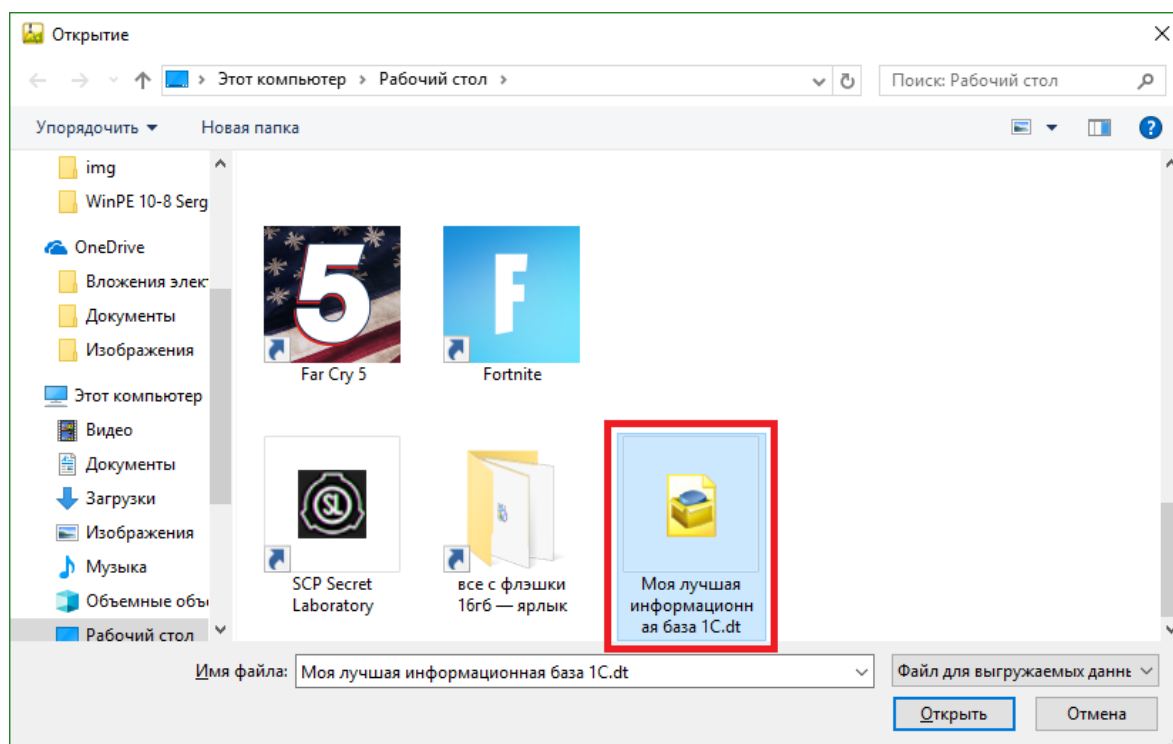
Входим в базу, в которую будем загружать наш архив, по кнопке "Конфигуратор".



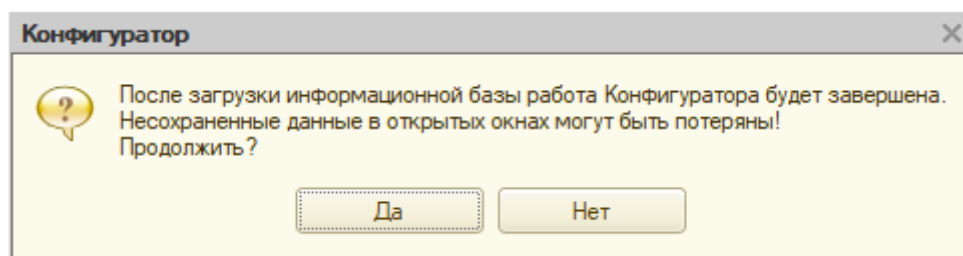
В меню выбираем — "Администрирование" — "Загрузить информационную базу".



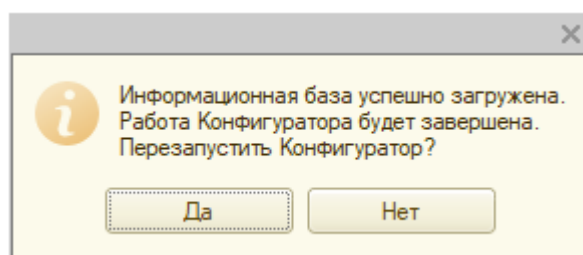
Выбираем файл с базой, который мы заблаговременно поместили на рабочий стол и нажимаем кнопку открыть.



Программа предупреждает нас, что произойдёт замена существующей базы на загружаемую. Нажимаем "Да".



Программа выдаёт сообщение, что база данных загружена и просит перезапустить конфигуратор. На этот вопрос мы отвечаем "Да" или "Нет", в зависимости от того, хотим ли мы продолжить работу в режиме Конфигуратора.



1. ПОСТАНОВКА ЗАДАЧИ ПО АВТОМАТИЗАЦИИ ДЕЯТЕЛЬНОСТИ ОРГАНИЗАЦИИ

Теория:

Рассмотрим практическую задачу по автоматизации оперативного учета хозяйствующего субъекта, относящегося к категории малого бизнеса. Сфера деятельности данного предприятия охватывает оптово-розничную торговлю и предоставление сопутствующих сервисных услуг конечным потребителям.

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 также сведения о его детях (рис. 1.1).

The screenshot shows a software application window titled "Паниковский М.С. (Сотрудники) (1С:Предприятие)". The window contains a form for employee data. At the top, there is a button "Записать и закрыть" and a "Сообщение" button. Below this, there are input fields for "Код:" (0002), "Подразделение:" (Продаж), "Наименование:" (Паниковский М.С.), and "Вид приема на работу:" (Штатный). Below these fields, there are tabs for "Образование" and "Дети". Under the "Образование" tab, there is a table with the following data:

N	Учреждение	Дата поступления	Дата окончания	Специальность
1	ПТУ	01.09.1997	20.06.1999	Наладчик СЧПУ
2	ВУЗ	01.09.1999	21.06.2004	Инженер

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.

Оказание услуг покупателю отражается документом «Оказание услуги» (рис. 1.2), а его оплата документом «Поступление денежных средств» (рис.1.3). Для документа «Расходная накладная» необходимо иметь возможность сформировать печатную форму.

Оказание услуги 000000002 от 14.07.2009 21:24:08

Провести и закрыть | Провести | Печать | Остатки по складу | Все действия ▾

Номер: 000000002

Дата: 14.07.2009 21:24:08

Склад: Основной

Клиент: Спиридонова Галина

Мастер: Гусаков Николай Дмитриевич

+ Добавить | ✕ | ↑ | ↓ | Все действия ▾

N	Номенклатура	H.	Количество	Цена	Сумма
1	Ремонт импортного телевизора		1,000	800,00	800,00
2	Строчный трансформатор Sams...		1,000	900,00	900,00
				Всего:	1 700,00

Рис. 1.2

Поступление денежных средств 000001 от ...

Провести и закрыть | Провести | Все действия ▾

Номер: 000001 от: 15.06.2014 10:53:04

Контрагент: Компьютер

Договор: ком-001

Сумма: 1 000,00 | Управленческий: ☐

Номер плательщика: Компьютер

Дата плательщика: 16.06.2014

Способ оплаты: ☐ Наличный ☒ Безналичный ☐ Ценные бумаги

Комментарий: срочно

Рис. 1.3

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

Для анализа деятельности компании необходимо иметь возможность узнать в разрезе подразделений в текущем месяце, на какую сумму было произведено продаж (рис. 1.4).

Мастер	Период	Клиент	Выручка
Итого			4 794,00
Гусаков Николай Дмитриевич			1 700,00
Деловой Иван Сергеевич			950,00
Симонов Валерий Михайлович			2 144,00

Рис. 1.4

2. КОНСТАНТЫ

Теория:

Правильно выбранный объект конфигурации – первый шаг к успешной разработке.

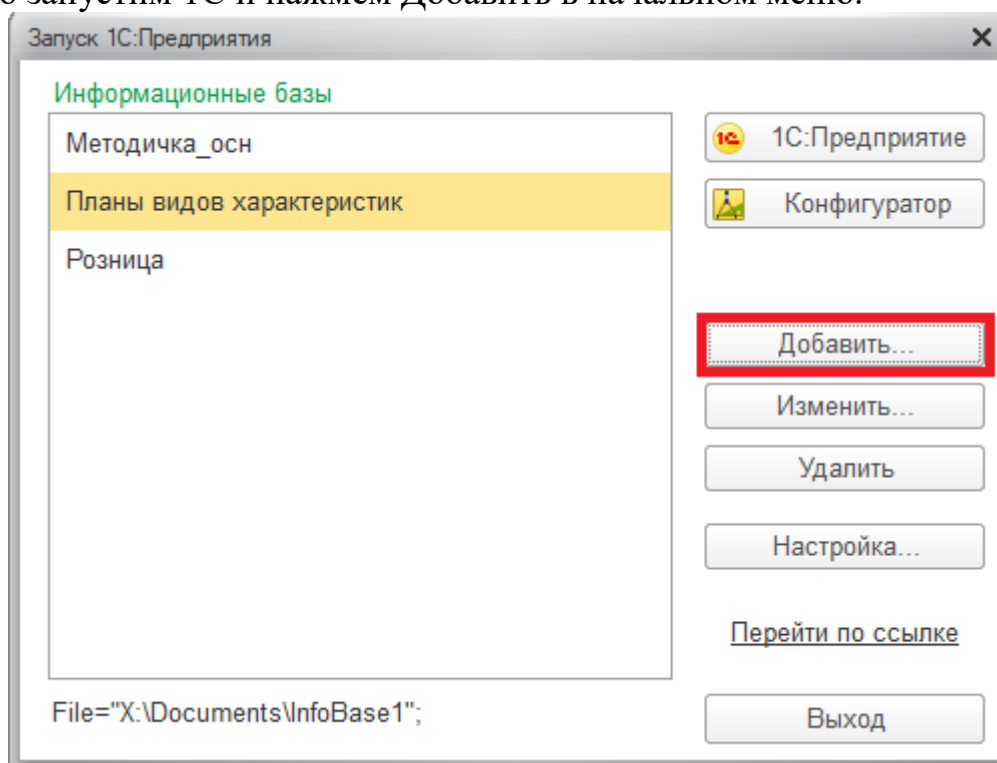
Существует множество предметных сущностей, которые могут использоваться одновременно в разных направлениях учета и документооборота. Объекты конфигурации для их реализации назовем условно прикладными объектами общего назначения, и начнем их рассмотрение по порядку, определенному в дереве конфигурации.

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.

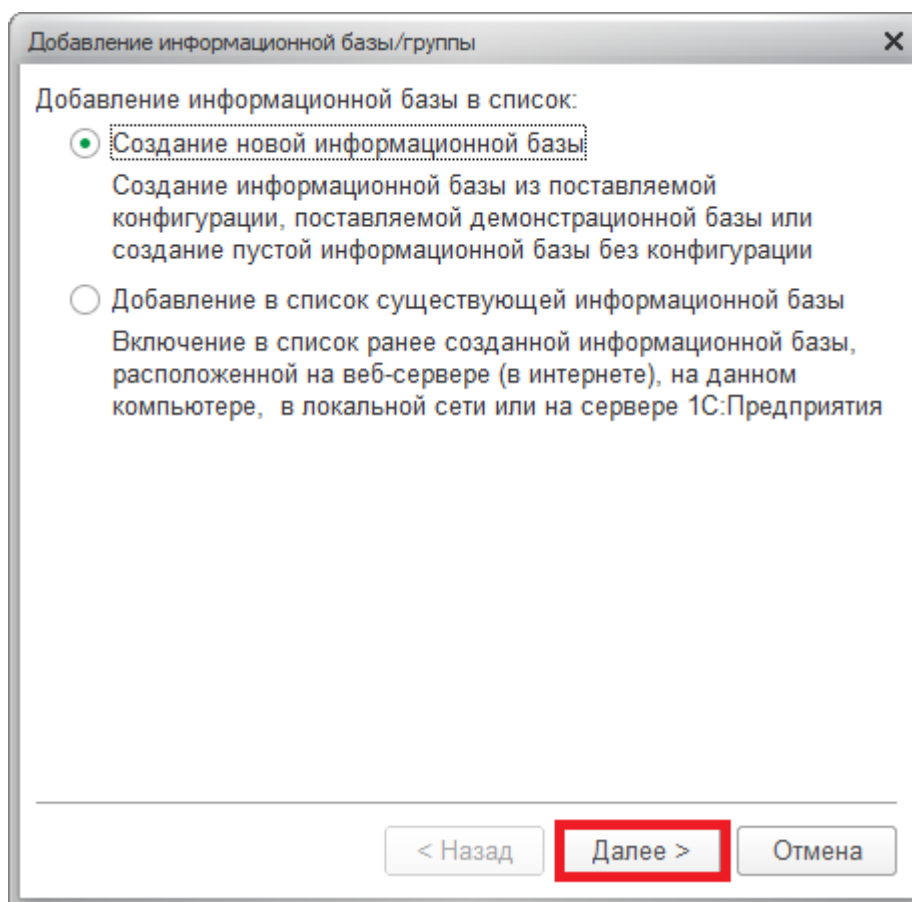
При оформлении документов нам понадобится информация о генеральном директоре нашей фирмы, ее наименовании и дате регистрации. Данная информация не будет меняться, поэтому для ее хранения можно использовать константу. Поскольку константа предназначена для хранения одного значения, потребуется создать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

Практика:

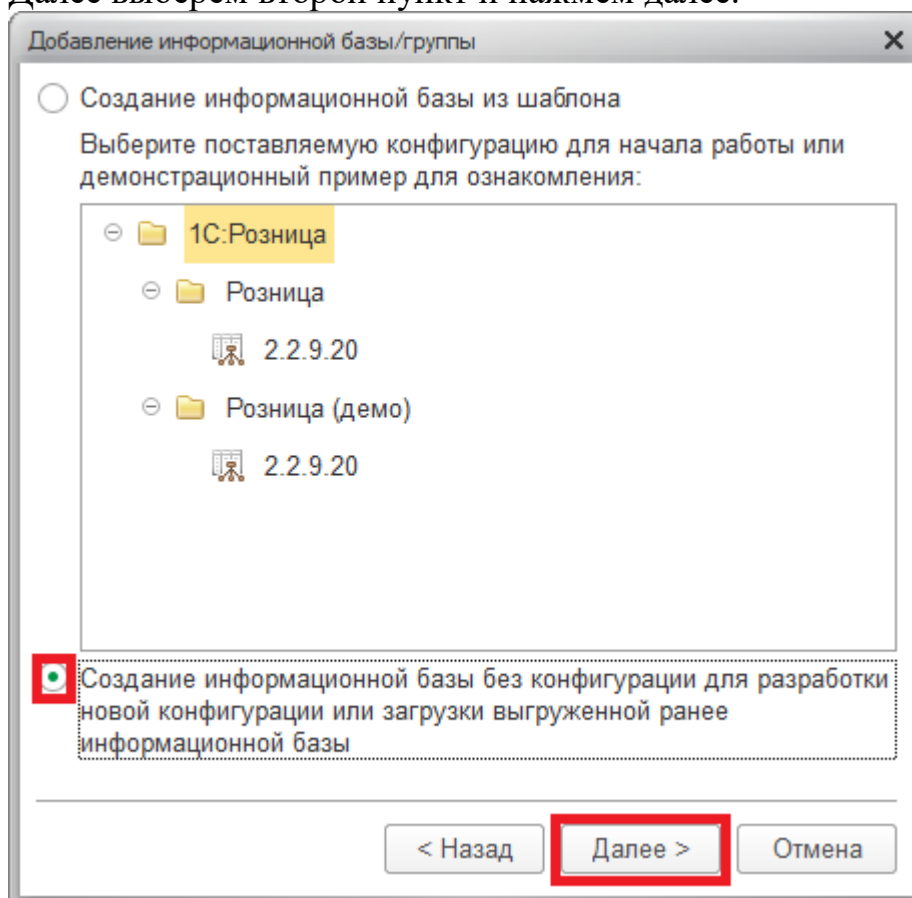
Для начала работы нам необходимо создать Информационную базу. Для этого запустим 1С и нажмем Добавить в начальном меню.



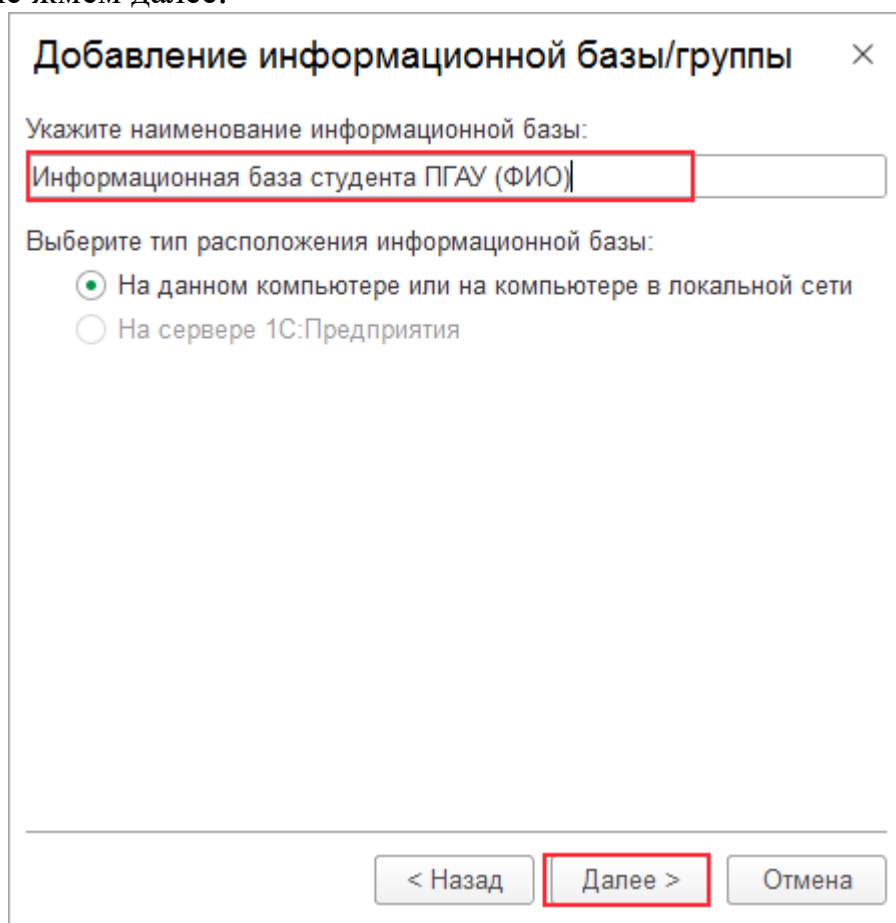
Далее выберем пункт «Создание новой информационной базы» и нажмем далее.



Далее выберем второй пункт и нажмем далее.



Нам необходимо указать наименование информационной базы и выбрать тип расположения информационной базы на данном компьютере, после жмем далее.



The screenshot shows a dialog box titled "Добавление информационной базы/группы" (Adding information base/group). It contains two main sections. The first section, "Укажите наименование информационной базы:" (Specify the name of the information base:), has a text input field containing "Информационная база студента ПГАУ (ФИО)". The second section, "Выберите тип расположения информационной базы:" (Select the type of information base location:), has two radio button options: "На данном компьютере или на компьютере в локальной сети" (On this computer or on a computer in the local network), which is selected, and "На сервере 1С:Предприятия" (On the 1C:Enterprise server). At the bottom of the dialog, there are three buttons: "< Назад" (Back), "Далее >" (Next), and "Отмена" (Cancel). The "Далее >" button is highlighted with a red rectangle.

Добавление информационной базы/группы

Укажите наименование информационной базы:

Информационная база студента ПГАУ (ФИО)

Выберите тип расположения информационной базы:

☒ На данном компьютере или на компьютере в локальной сети

☐ На сервере 1С:Предприятия

< Назад Далее > Отмена

В следующем окне выбираем где будет храниться информационная база и жмем далее.

Добавление информационной базы/группы

Укажите параметры информационной базы:

Каталог информационной базы:

X:\Documents\InfoBase3

Язык (Страна):

русский (Россия)

< Назад **Далее >** Отмена

В следующем окне оставляем все без изменений и жмем готово.

Добавление информационной базы/группы

Укажите параметры запуска:

Вариант аутентификации (определения пользователя):

☒ Выбирать автоматически

☐ Запрашивать имя и пароль

Скорость соединения: Обычная

Дополнительные параметры запуска:

Основной режим запуска:

☒ **Выбирать автоматически**

☐ Тонкий клиент

☐ Веб-клиент

☐ Толстый клиент

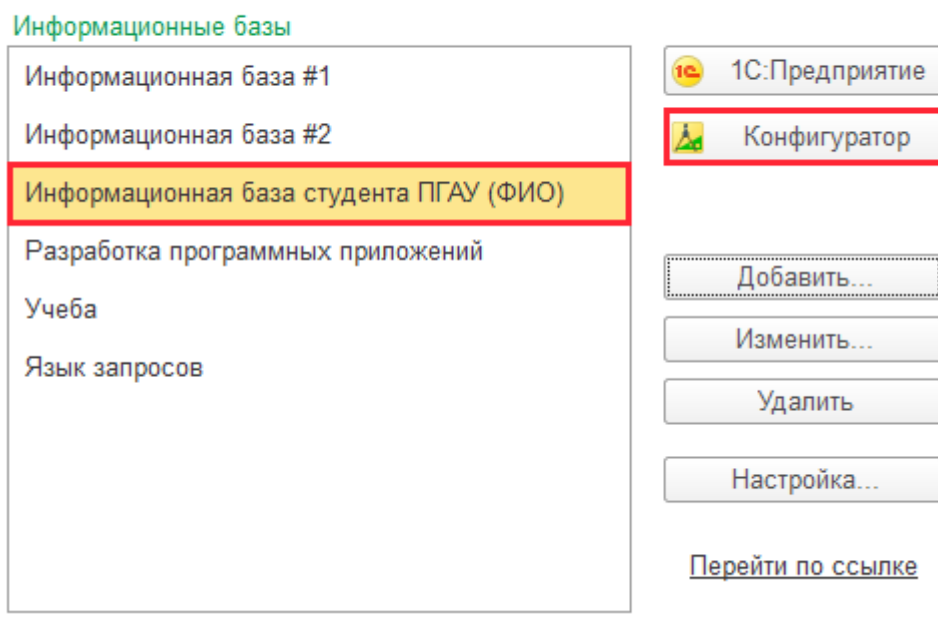
Версия 1С:Предприятия: 8.3

Разрядность:

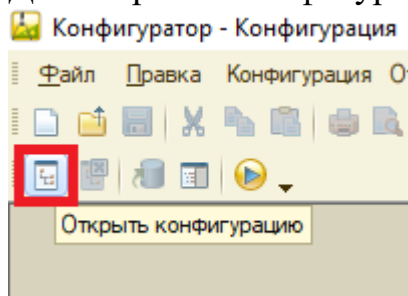
< Назад **Готово** Отмена

Далее нам необходимо открыть информационную базу в режиме Конфигуратор.

Запуск 1С:Предприятия (учебной версии)



Для открытия конфигурации нажмем соответствующую кнопку.



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).

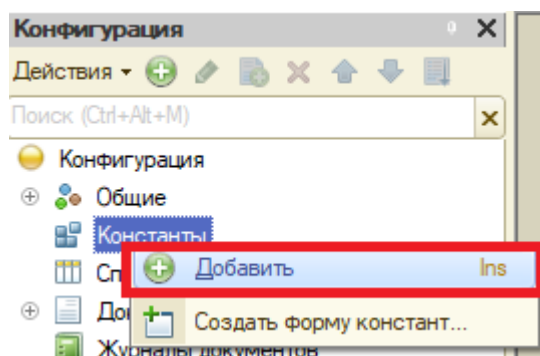


Рис. 2.1

После того как мы выберем данный пункт, на экране откроется новое окно, называемое *Палитра свойств*, в которой будут отражены свойства константы (рис 2.2). В поле имя напишем «НазваниеОрганизации» и нажмем Enter. Обратите внимание, синоним сгенерировался автоматически. Укажем длину 50.

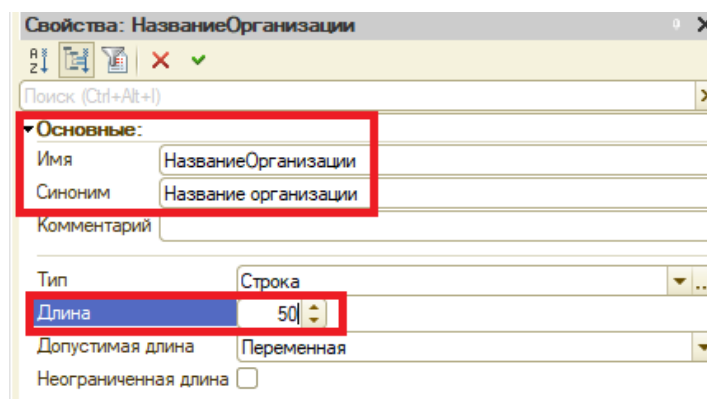


Рис. 2.2

В дальнейшем можно вызвать это окно двойным щелчком мыши на соответствующей ветке в окне дерева объектов конфигурации или через пункт *Свойства* контекстного меню, вызванного правой кнопкой мышки (рис 2.3).

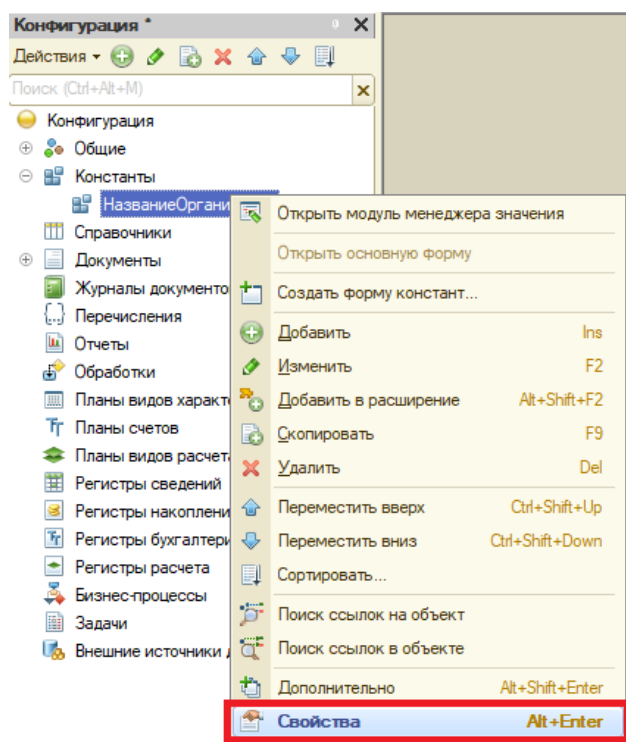


Рис. 2.3

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, всегда хранится 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 зовут директора нашей организации. 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* и последовательно подтверждаем необходимость обновления информационной базы (рис. 2.4-рис. 2.5).

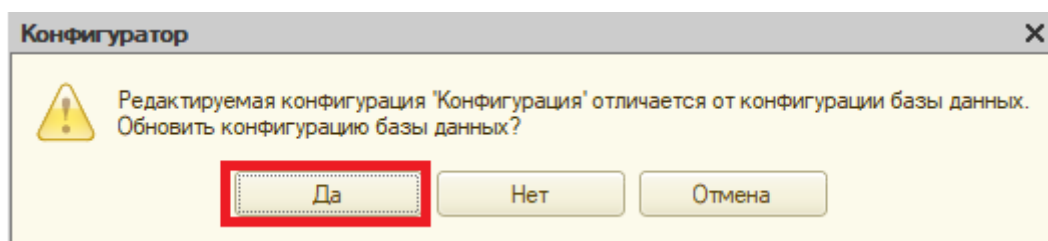


Рис. 2.4

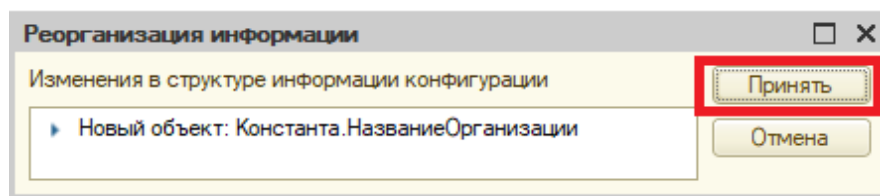


Рис. 2.5

Только после выполнения всех этих действий в пользовательском режиме можно ввести необходимую информацию (рис. 2.6-2.7).

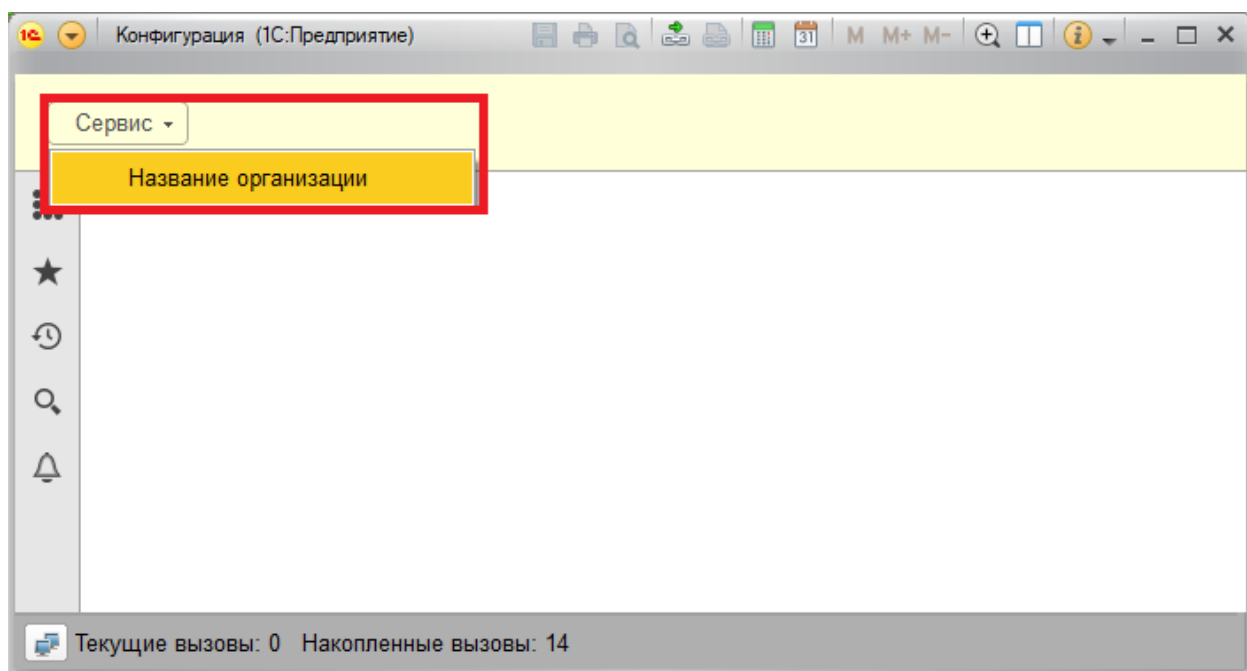


Рис. 2.6

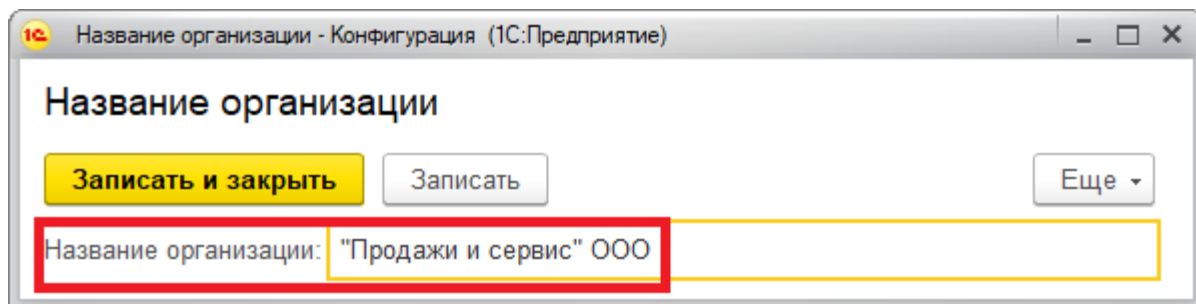


Рис. 2.7

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

Приступим к созданию следующей константы для учета даты регистрации 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 2.8).

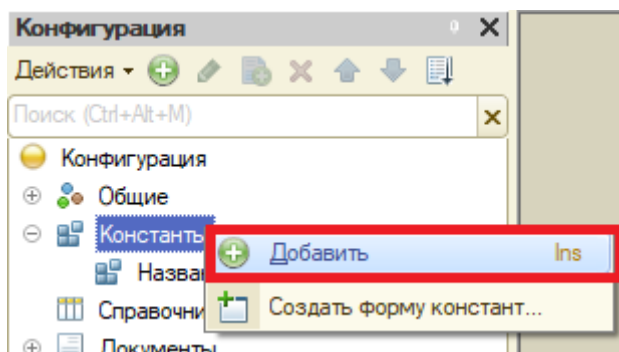


Рис. 2.8

После того как мы выберем данный пункт, на экране откроется палитра свойств создаваемой константы. Заполняем поля *Имя* значением «ДатаРегистрации» и нажмём Enter для автоматического заполнения синонима, а тип данных указываем *Дата*. В этом случае набор свойств изменился по сравнению с тем, какие свойства были у константы *НазваниеОрганизации* с типом данных *Строка* (рис. 2.9). В системе «1С:Предприятие 8» предусмотрено хранение даты с точностью до секунды. Мы имеем возможность вручную указать нужный нам состав в зависимости от того состава даты, с которым нам удобнее работать в данном случае. В поле *Состав даты* можно выбрать один из трех вариантов представления: дата, время и дата со временем. Для нашей цели необходимо выбрать вариант *Дата*, поскольку время регистрации нашего предприятия нам не важно.

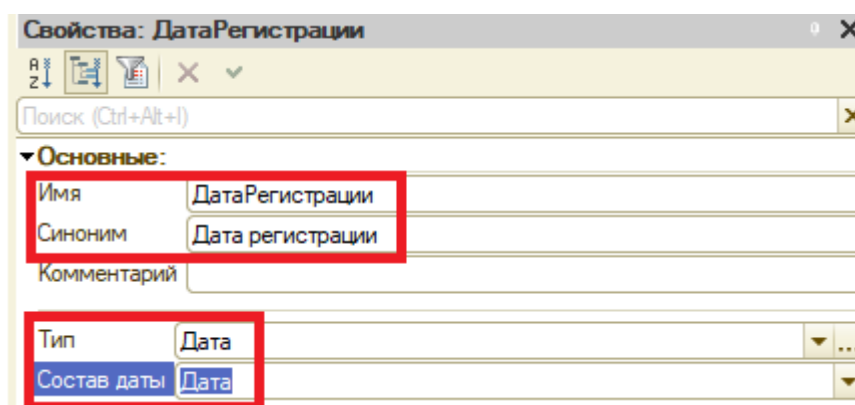


Рис. 2.9

Запустим в режиме предприятия и укажем реальную дату регистрации компании 27.05.18. (рис. 2.10)

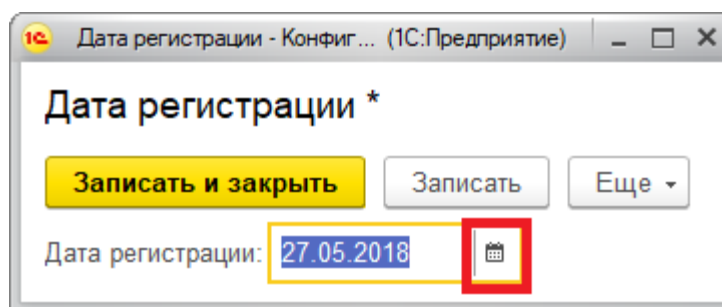


Рис. 2.10

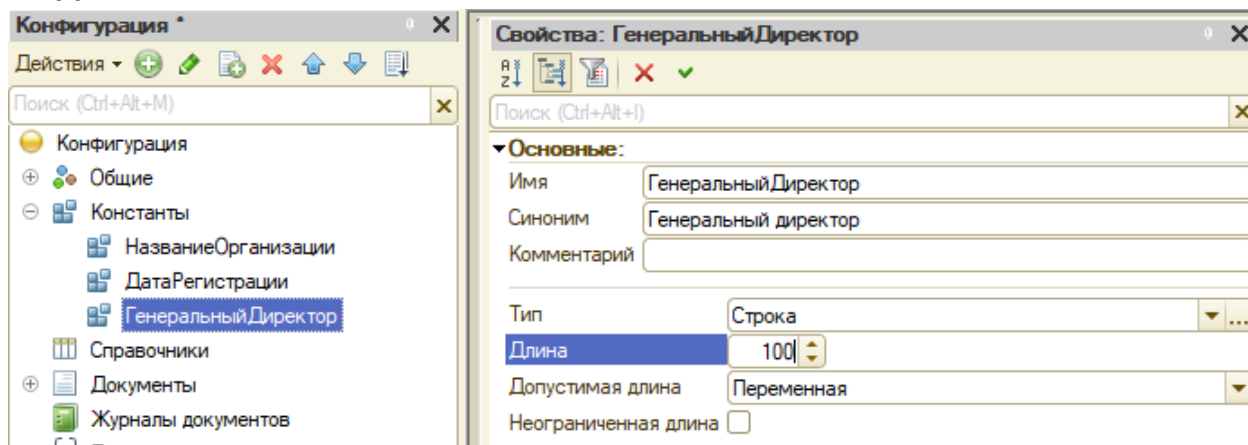
Этот пример достаточно ясно демонстрирует основы работы системы «1С:Предприятие 8.3». На этапе конфигурирования создаются необходимые структуры для хранения информации и далее на этапе исполнения эти структуры заполняются необходимыми данными.

Самостоятельная работа №1

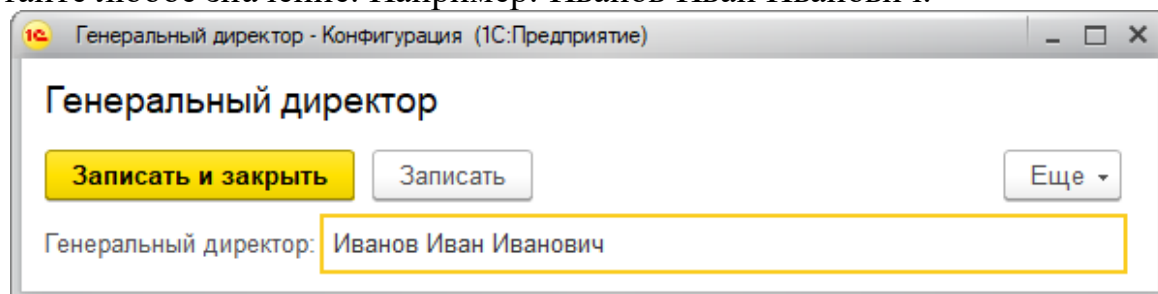
Создайте новую константу **ГенеральныйДиректор** для хранения в информационной базе информации о генеральном директоре и присвойте значение константе в режиме 1С:Предприятие.

Проверка выполнения самостоятельной работы №1.

1. Создадим новую константу и присвоим ей имя «ГенеральныйДиректор» и синоним «Генеральный директор». Длина строки будет 100.



2. Запустим конфигурацию в режиме 1С:Предприятие и присвоим константе любое значение. Например: Иванов Иван Иванович.



3. СПРАВОЧНИКИ

3.1. Общее описание и назначение

Теория:

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 также 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.

Практика:

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* (рис. 3.1).

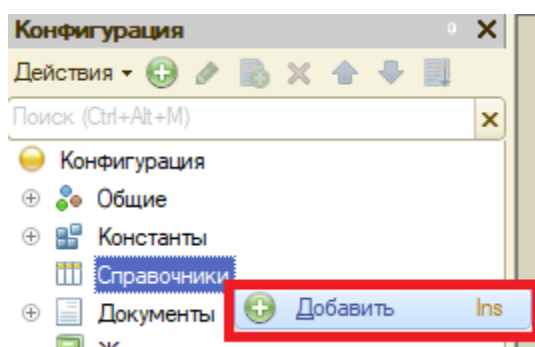


Рис. 3.1

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 нему 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 основном 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», а для пользователя дополнительно расшифруем в поле Синоним как «Товары и услуги»(рис 3.2).

Рис. 3.2

На этом создание простейшего справочника завершено. Далее запускаем конфигурацию в режиме предприятия и смотрим на то, что мы только что создали (рис. 3.3). Все необходимые действия при работе со справочником можно произвести путем нажатия на кнопку «Ещё» и выбора соответствующего пункта в выпадающем меню.

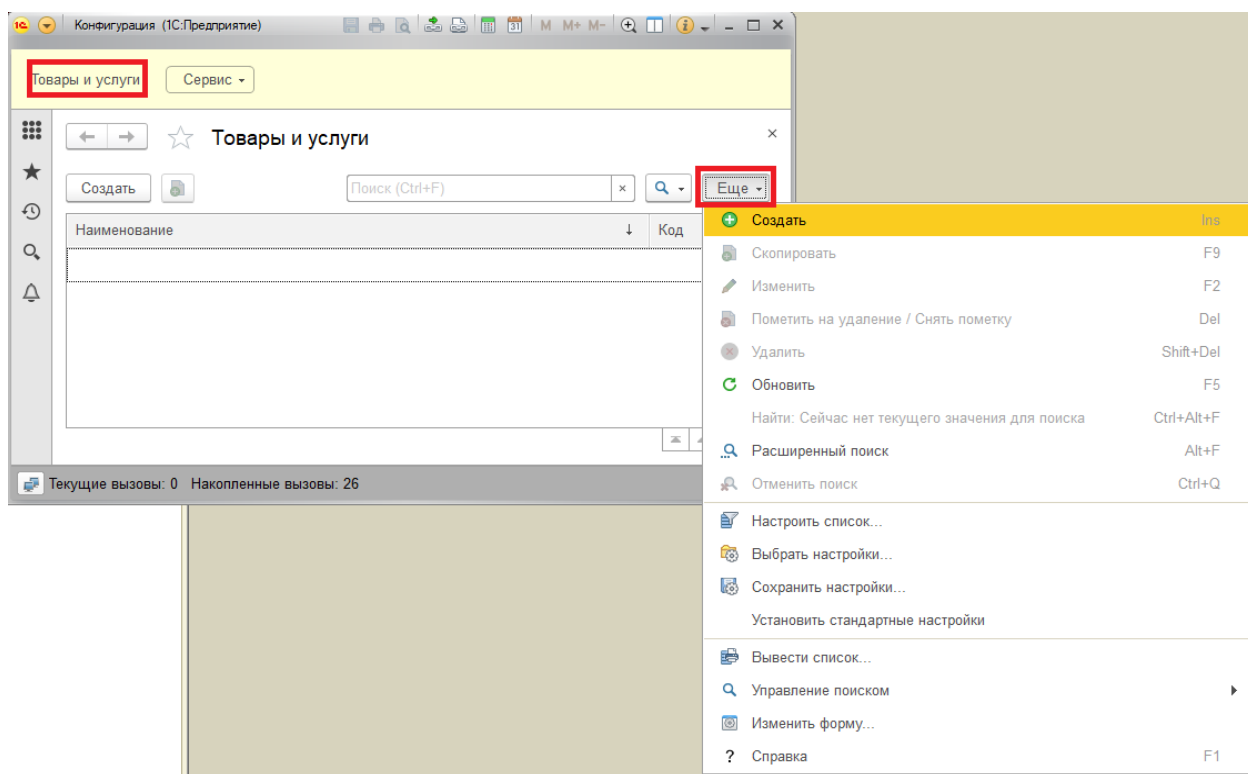


Рис. 3.3

Для заполнения справочника необходимо нажать кнопку создать и вписать Наименование, затем нажать «Записать и закрыть» (рис. 3.4.). Таким образом заполнить справочник следующими данными: Бойлер, Кондиционер бытовой, Посудомоечная машина, Стиральная машина, Холодильник бытовой (рис. 3.5).

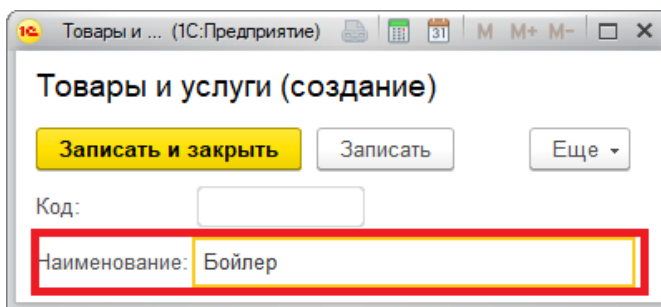


Рис. 3.4

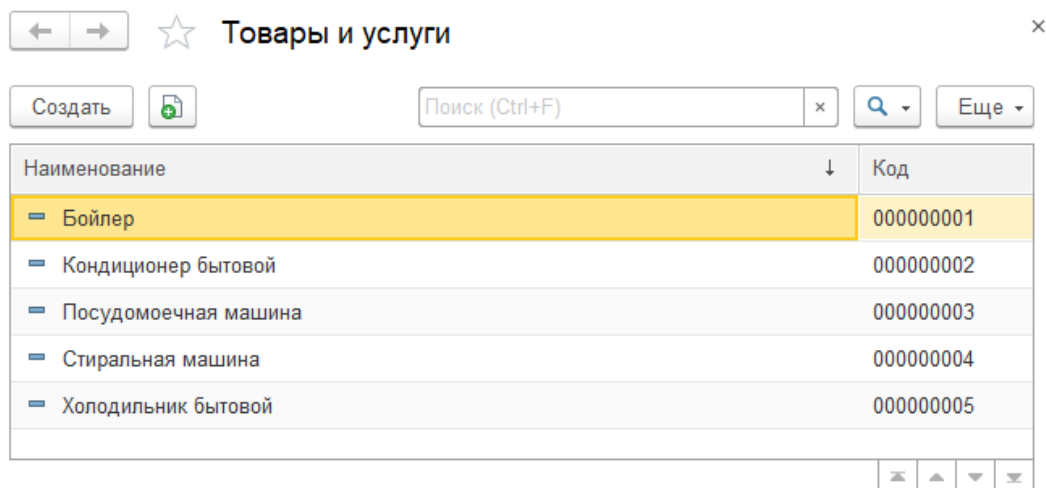


Рис. 3.5

3.2. Иерархия

Теория:

Созданный нами справочник позволяет работать только с простейшими 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.

Практика:

Сделаем справочник «Номенклатура» иерархическим. Для этого в режиме конфигуратора в окне дерева объектов конфигурации двойным щелчком мышки на справочнике «Номенклатура» вызываем окно редактирования справочника и переходим на закладку *Иерархия*. Прежде всего, необходимо проставить флажок *Иерархический справочник* (рис. 3.6).

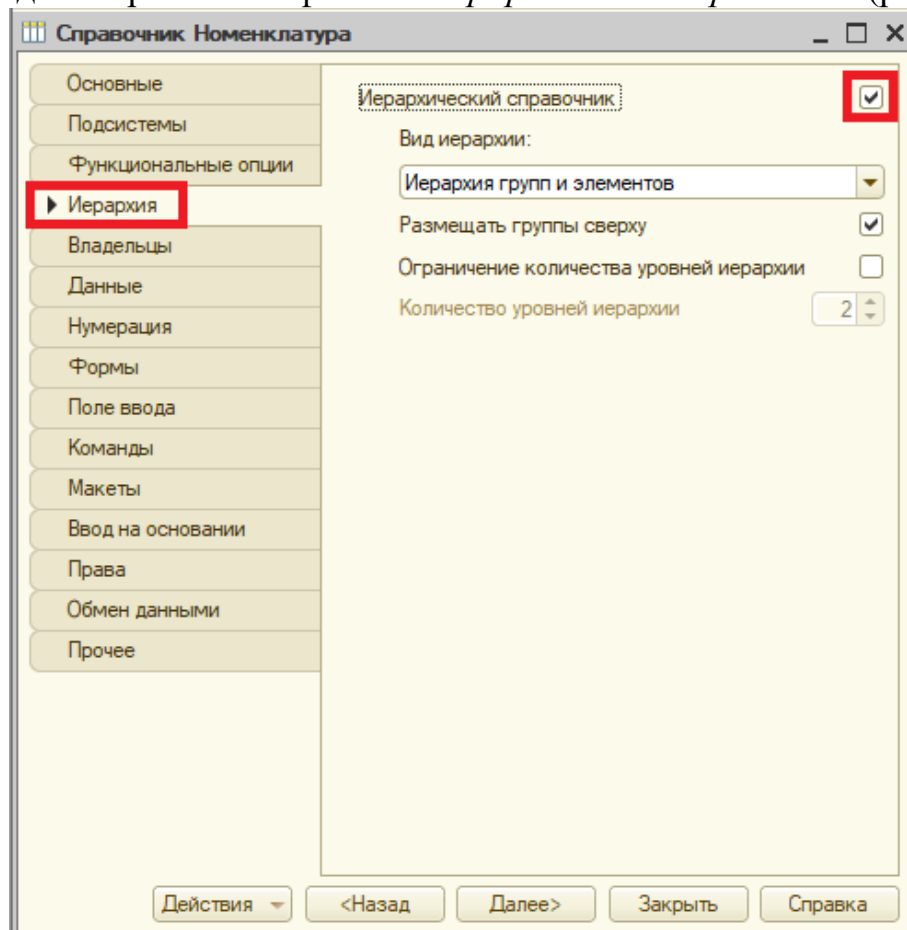


Рис. 3.6

Теория:

Следующее поле, которое нам необходимо выбрать: *Вид иерархии*, где можно 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 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*. 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*.

Практика:

Исходя из условий нашей задачи, справочник «Номенклатура» должен быть с видом иерархии *Иерархия групп и элементов*, который указан по умолчанию (рис. 3.7).

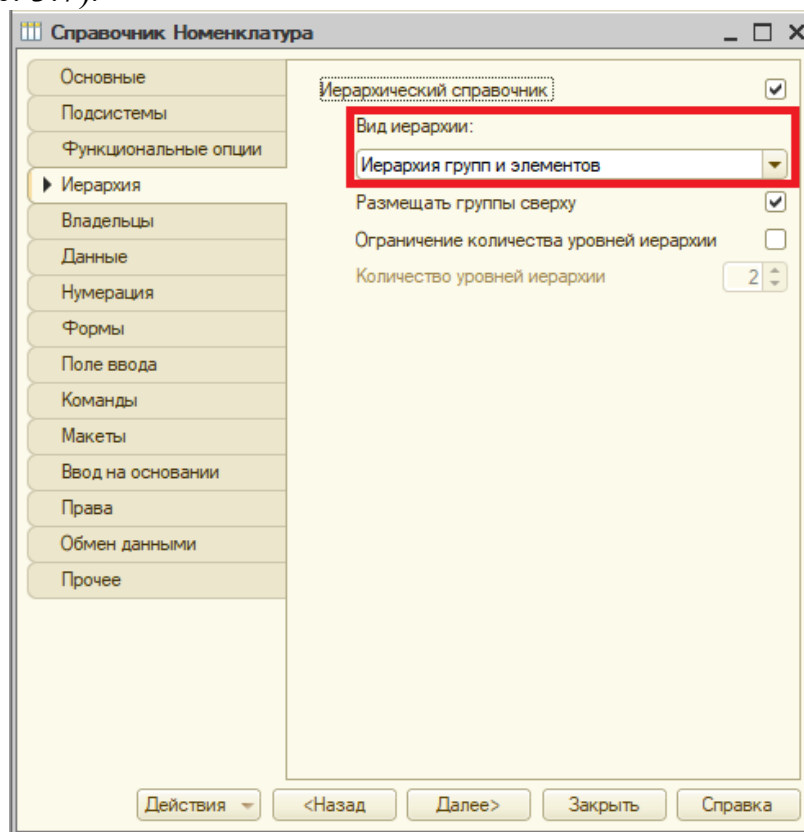


Рис. 3.7

Запустим конфигурацию в режиме предприятия. Для пользователя теперь внешний вид справочника изменится. Помимо элементов появятся еще и группы. Работа с ними аналогична работе с файлами и папками в проводнике. Новые элементы и группы можно сразу создавать в нужной группе, а для существующих имеется возможность перемещения по группам.

Создадим группу и назовём ее «Товары и услуги» с помощью кнопки «Создать группу» (рис. 3.8).

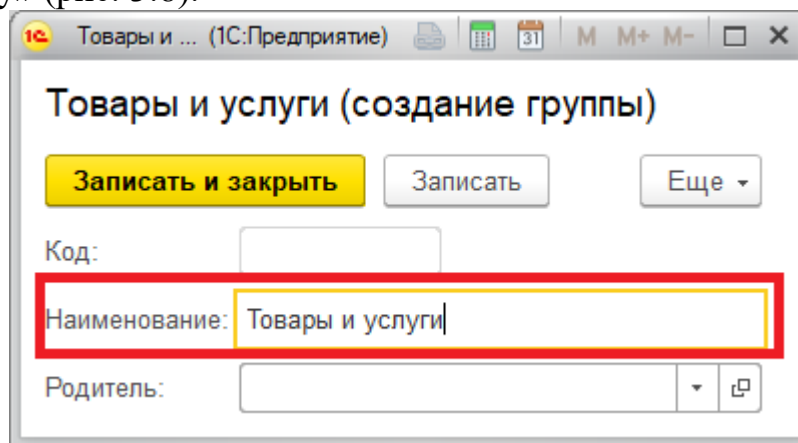


Рис. 3.8

Далее в группе «Товары и услуги» создадим новую группу и назовём её «Товары». Внутри неё создадим группу «Бойлеры» и укажем в качестве родительской группы «Товары» (рис. 3.9).

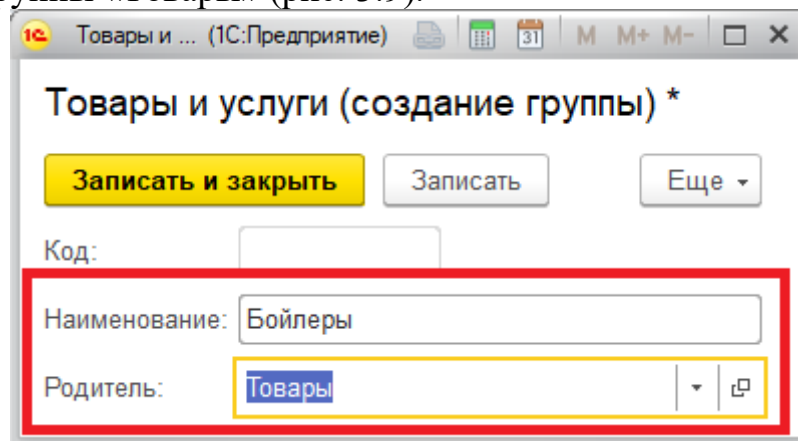


Рис. 3.9

Затем переместим Бойлер в группу Бойлеры. Для этого щелкнем по Бойлер правой кнопкой мыши и выберем пункт «Переместить в группу» (рис. 3.10). И выберем из списка группу «Бойлеры» (рис. 3.11).

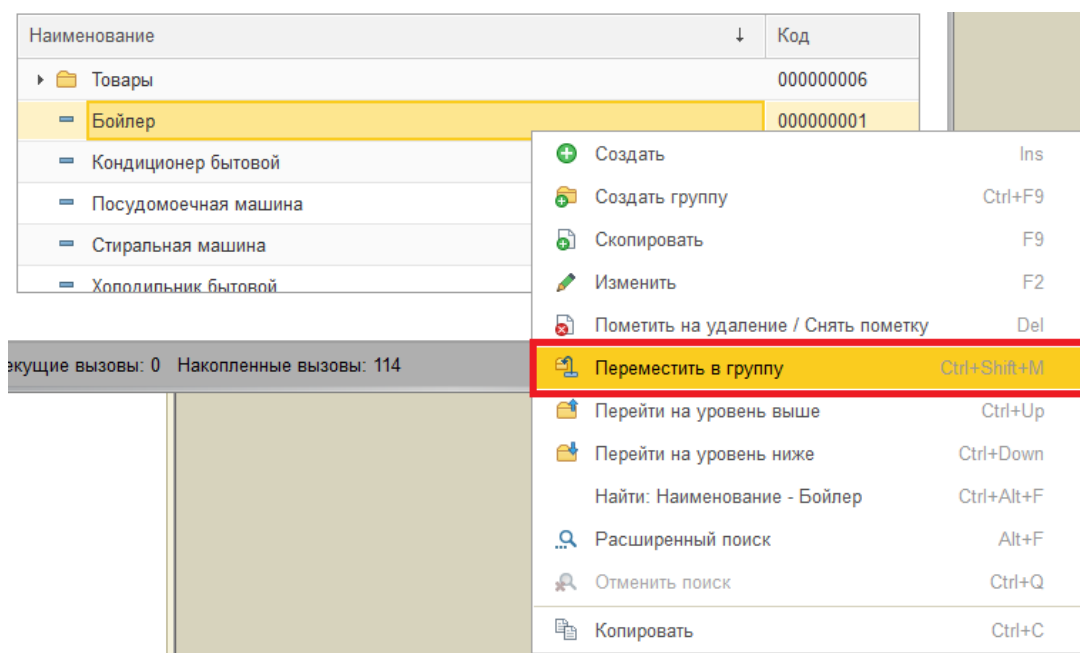


Рис. 3.10

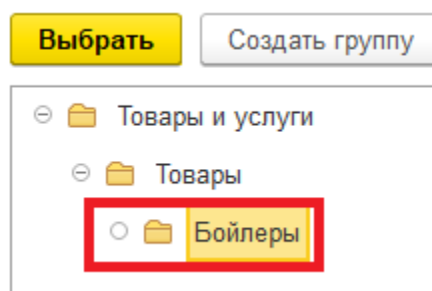


Рис. 3.11

Следует обратить внимание, что с помощью пункта меню *Еще – Режим просмотра* можно выбрать один из трех способов отображения справочника. Если выбрать режим «Дерево» (рис. 3.12), то можно перемещать элементы и группы с помощью «Drag and drop», то есть выбрать элемент, зажать левую кнопку мыши и переместить в желаемую группу избегая долгой процедуры перемещения через «Переместить в группу».

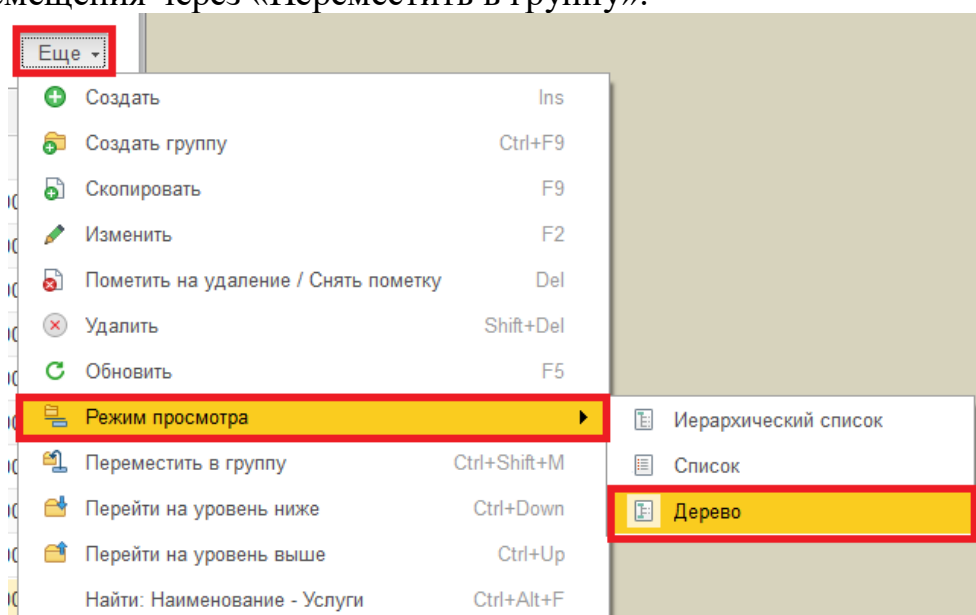
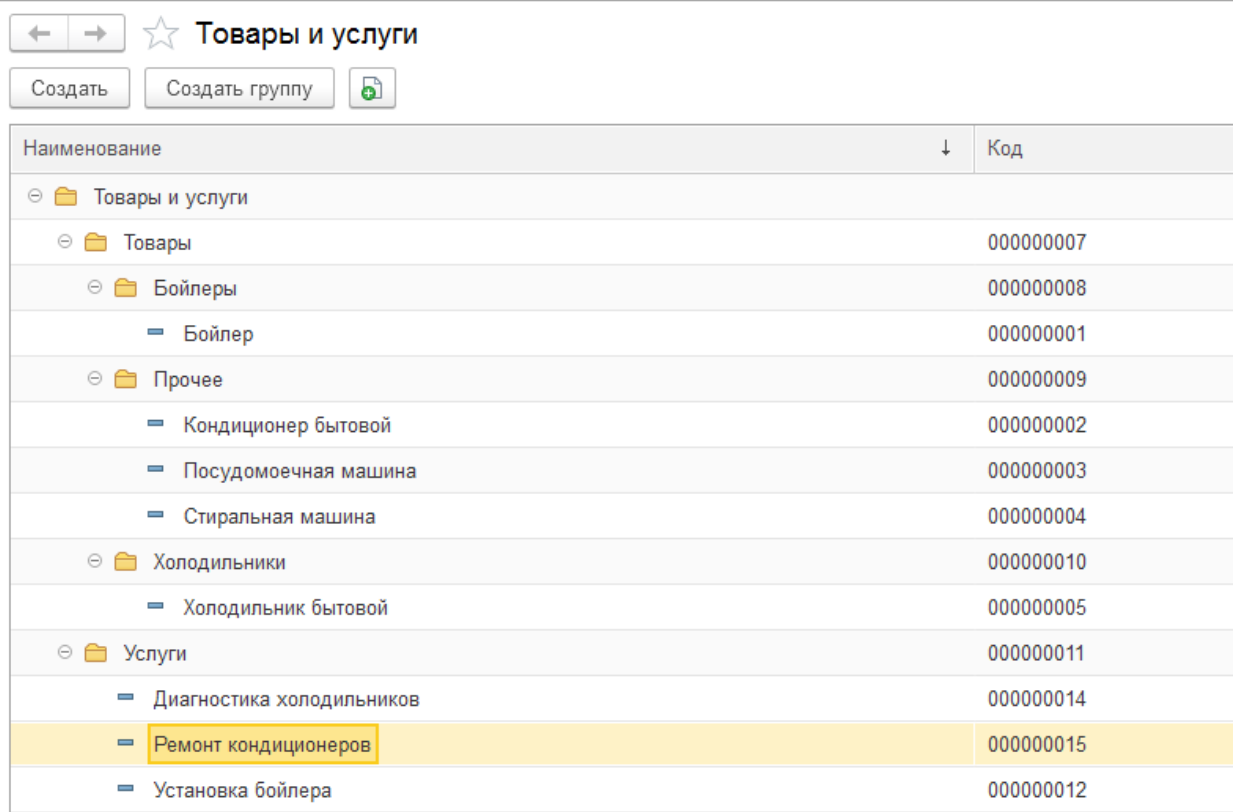


Рис. 3.12

Далее необходимо привести структуру справочника «Товары и услуги» к виду, как указано ниже (рис. 3.13)



Наименование	Код
Товары и услуги	
Товары	000000007
Бойлеры	000000008
Бойлер	000000001
Прочее	000000009
Кондиционер бытовой	000000002
Посудомоечная машина	000000003
Стиральная машина	000000004
Холодильники	000000010
Холодильник бытовой	000000005
Услуги	000000011
Диагностика холодильников	000000014
Ремонт кондиционеров	000000015
Установка бойлера	000000012

Рис. 3.13

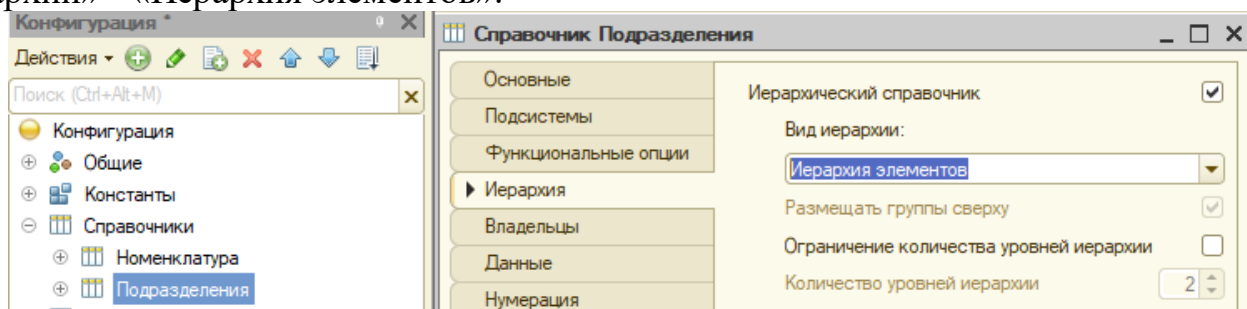
Практическая работа №2

Создайте новый справочник **Подразделения** для хранения данных и подразделений организации.

Справочник должен иметь иерархию элементов.

Проверка выполнения самостоятельной работы №2.

1. Создадим справочник «Подразделения». Во вкладке «Иерархия» отметим галку рядом с полем «Иерархический справочник» и выберем «Вид иерархии» - «Иерархия элементов».



2. Запустим конфигурацию в режиме предприятия и заполним справочник «Подразделения» как показано ниже.

← → ☆ Подразделения	
Создать Создать группу +	
Наименование ↓	Код
▼ Подразделения	000000001
▶ Подразделения ▶ Офис 1 в Москве	000000002
Подразделение 1	000000003
Подразделение 2	000000004

3.3. Предопределенные элементы

Теория:

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.

Практика:

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 (рис. 3.14).

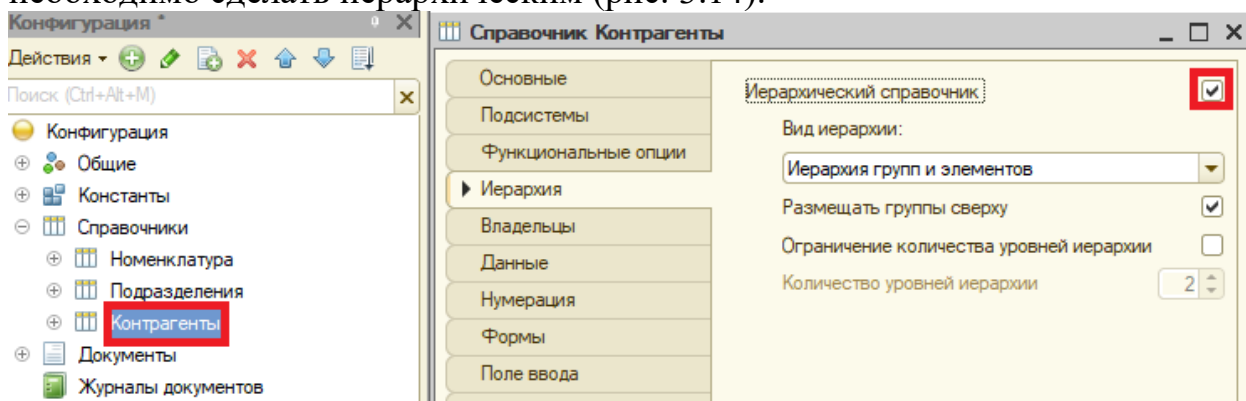


Рис. 3.14

Затем либо с помощью контекстного меню в дереве конфигурации (рис.3.15), либо по кнопке «Предопределенные» на закладке *Прочее* (рис. 3.16) вызываем окно для редактирования предопределенных элементов справочника (рис. 3.17).

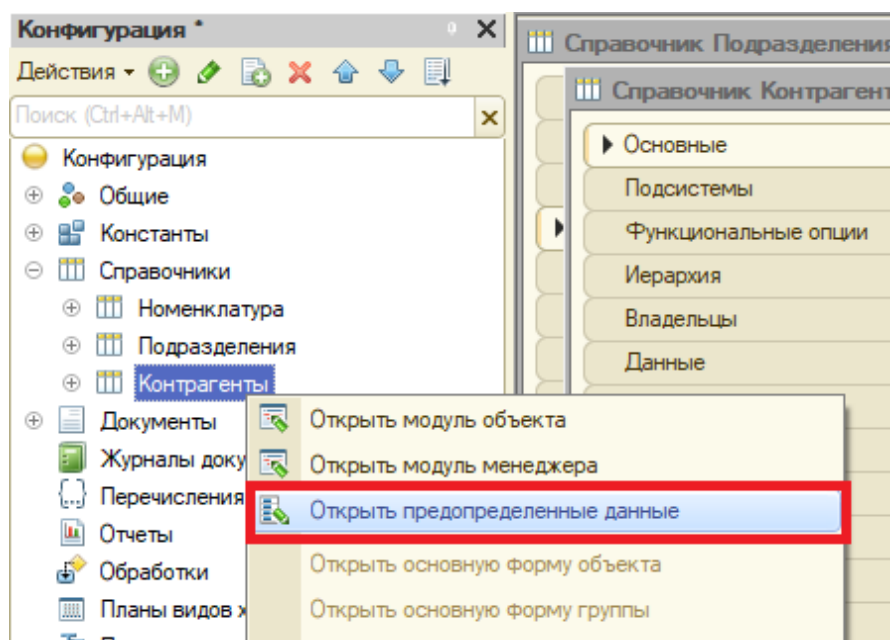


Рис. 3.15

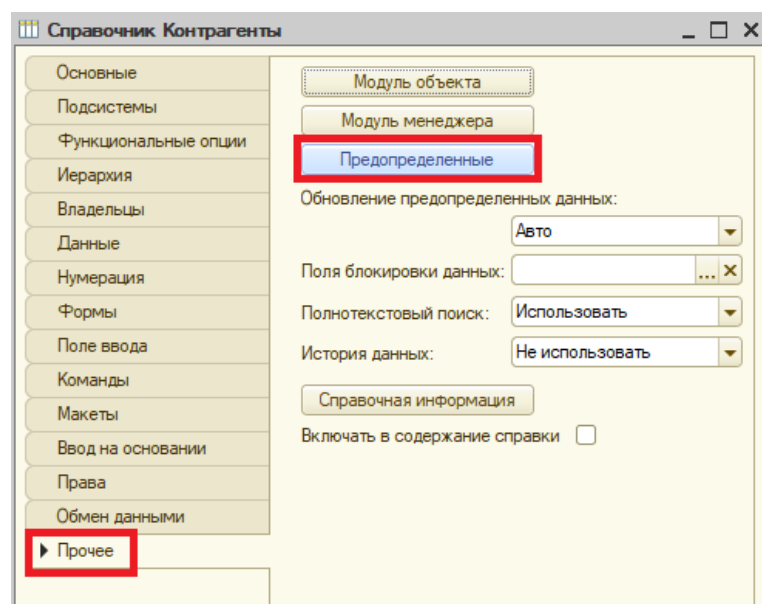


Рис. 3.16

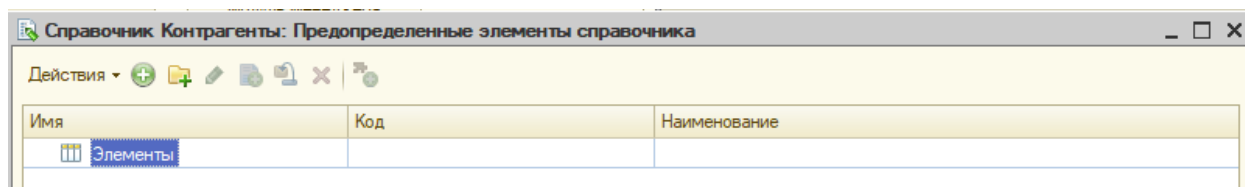
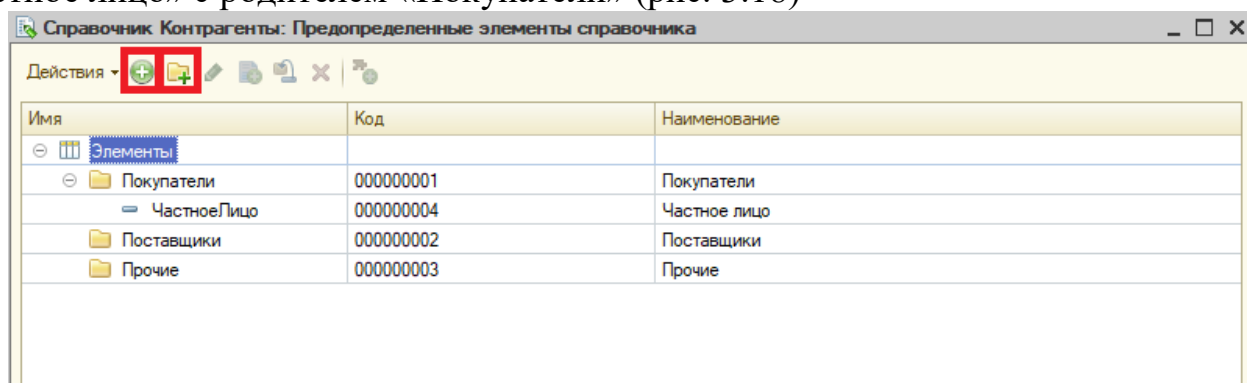


Рис. 3.17

Добавление новых predeterminedенных групп и элементов происходит так же, как и в пользовательском режиме. Разница 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». Создадим 3 группы с родителем «Элементы»: имя «Покупатели» и наименование «Покупатели», имя «Поставщики» и наименование «Поставщики», имя «Прочие» и наименование

«Прочие», а также элемент с именем «ЧастноеЛицо» и наименованием «Частное лицо» с родителем «Покупатели» (рис. 3.18)



Имя	Код	Наименование
Элементы		
Покупатели	000000001	Покупатели
ЧастноеЛицо	000000004	Частное лицо
Поставщики	000000002	Поставщики
Прочие	000000003	Прочие

Рис. 3.18

3.4. Реквизиты справочника

Теория:

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

Как уже упоминалось, 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 в элементе.

На закладках *Данные* и *Нумерация* определяются следующие свойства кода справочника (рис 3.19-3.20):

- **автономерация** – 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);

- **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.

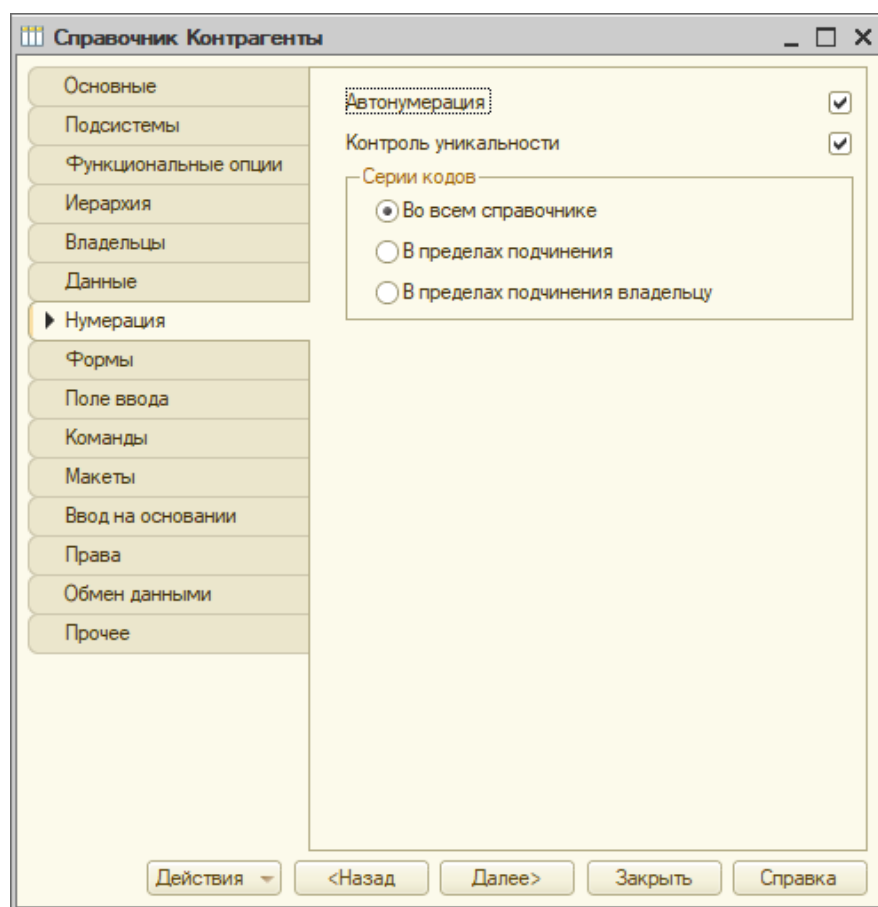


Рис. 3.19

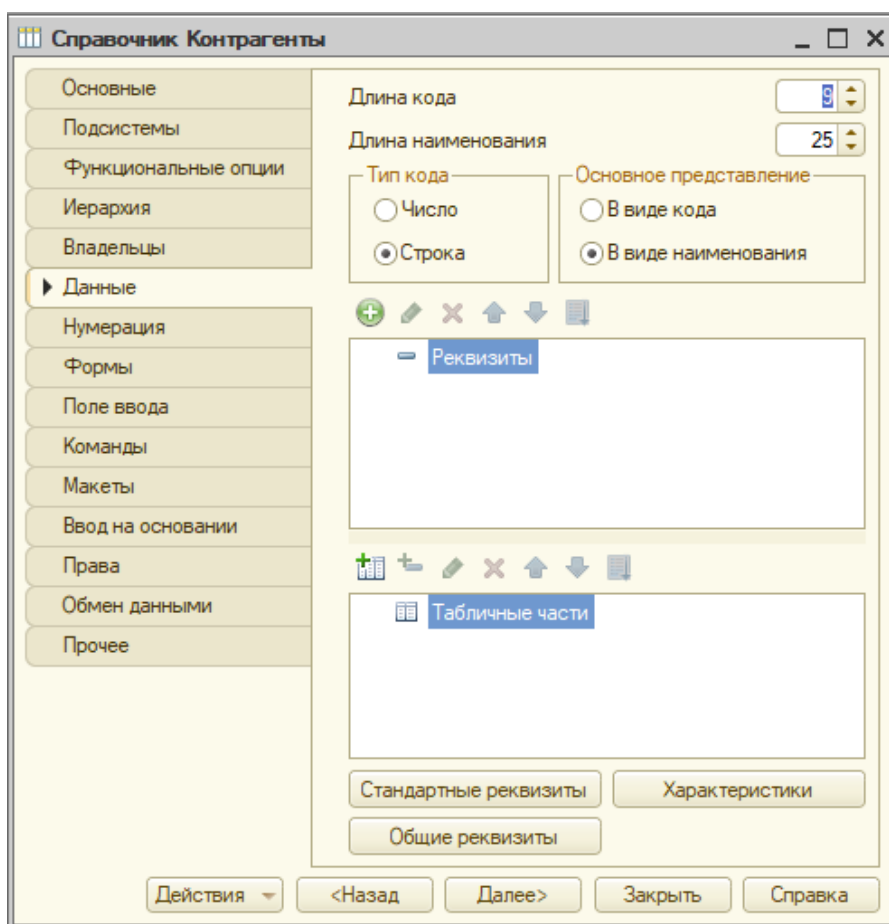


Рис. 3.20

Помимо системных, 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 3.12), или выбрав пункт *Добавить* в панели инструментов этого поля. Далее необходимо определить свойства нашего реквизита в соответствующем окне (рис. 3.13). 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*.

Существенными для нас 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. Тип данных может быть примитивного или ссылочного типа, а также составного типа.

С помощью реквизитов может быть решена и следующая задача: необходимо хранить наименование нашего контрагента, но это наименование очень длинное. Поскольку реквизит *Наименование* является системным, то на него наложены определенные ограничения, в т.ч. 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.

Практика:

В справочнике «Контрагенты» во вкладке Данные создадим новый реквизит ИНН, тип Число, длина 12 и поставим галочку неотрицательное (рис. 3.21).

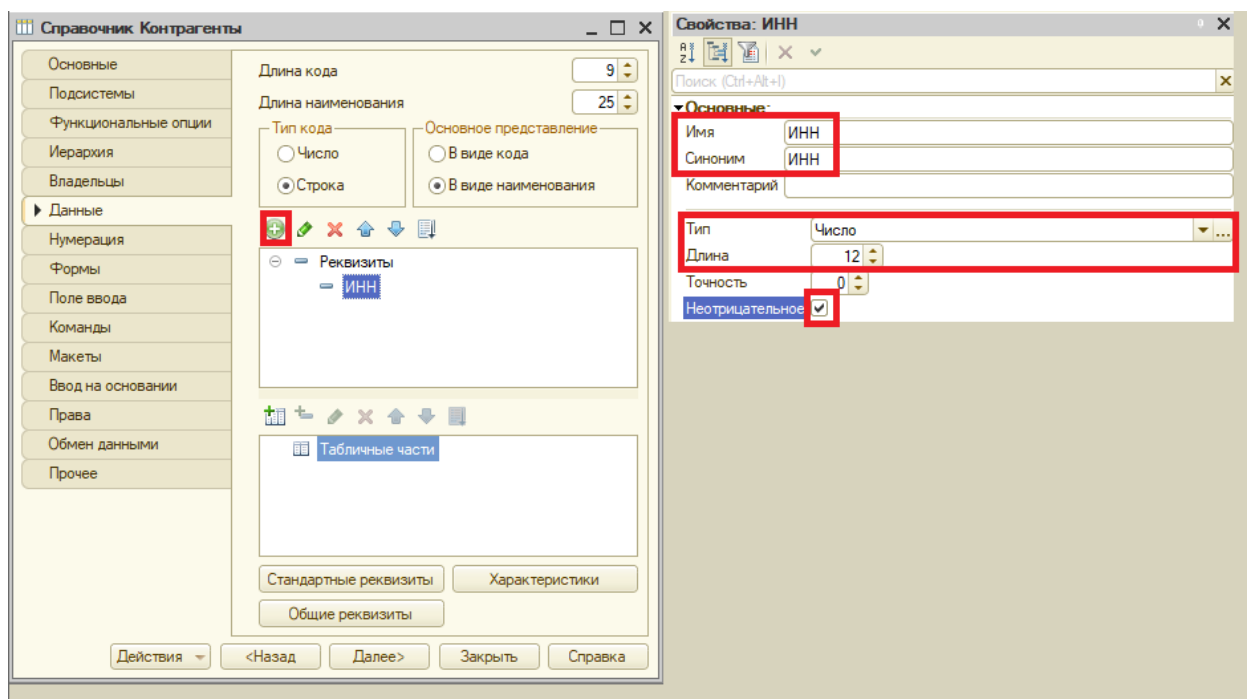


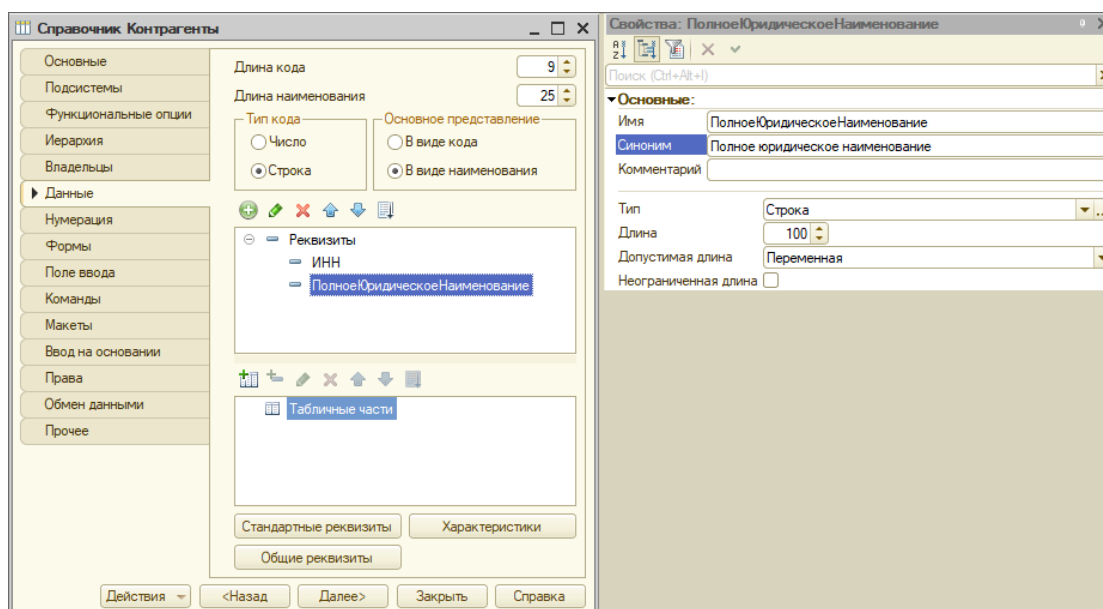
Рис. 3.21

Практическая работа №3

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

Проверка выполнения практической работы №3:

1. Создадим в справочнике Контрагенты реквизит с именем «ПолноеЮридическоеНаименование» и укажем длину 100.



Теория:

Следует отметить некоторые особенности палитры свойств. Во-первых, правой кнопкой мышки можно выбрать контекстное меню, в котором указать способ отображения списком или закладками. Так, на рис. 3.21 одна и та же палитра свойств отображается закладками и списком. Причем в случае отображения списком можно сворачивать и разворачивать соответствующие группы.

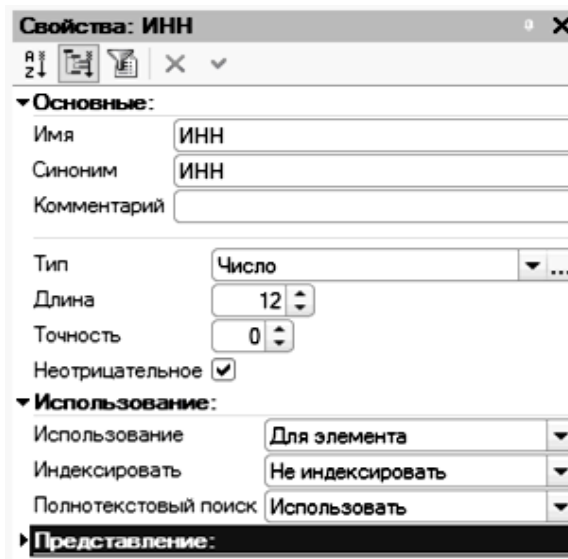


Рис. 3.21

Во-вторых, в панели инструментов существует кнопка *Показывать только важные*, предназначенная для того, чтобы не распылять внимание на заполнение не часто используемых свойств. При ее использовании всегда следует помнить о том, что в этом случае видны не все свойства.

Следует обратить внимание, что палитру свойств можно также вызвать и для всего справочника. Для этого правой кнопкой мышки на соответствующей ветке в окне дерева объектов конфигурации надо вызвать

контекстное меню и в нем выбрать пункт *Свойства* (рис. 3.22). В заголовке окна палитры свойств указывается имя объекта, чьи свойства отображаются в текущем окне.

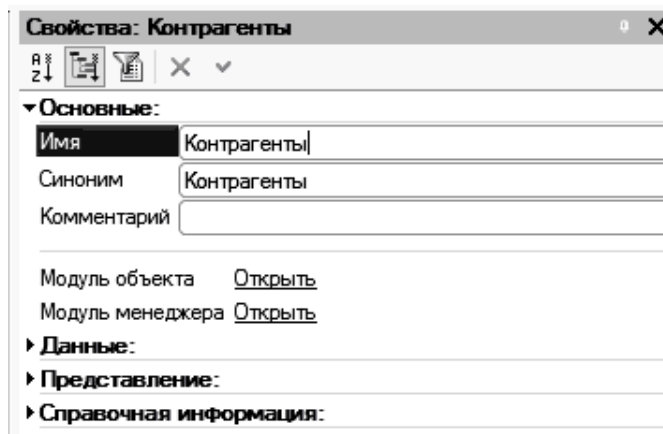


Рис. 3.22

Рассмотрим теперь следующую ситуацию. Необходимо отражать в системе информацию о подразделении, в котором работает сотрудник.

Практика:

Для решения поставленной задачи добавим в Окне конфигурации новый справочник *Сотрудники* и далее создадим у него реквизит *Подразделение* с типом данных *СправочникСсылка.Подразделения* (рис. 3.23).

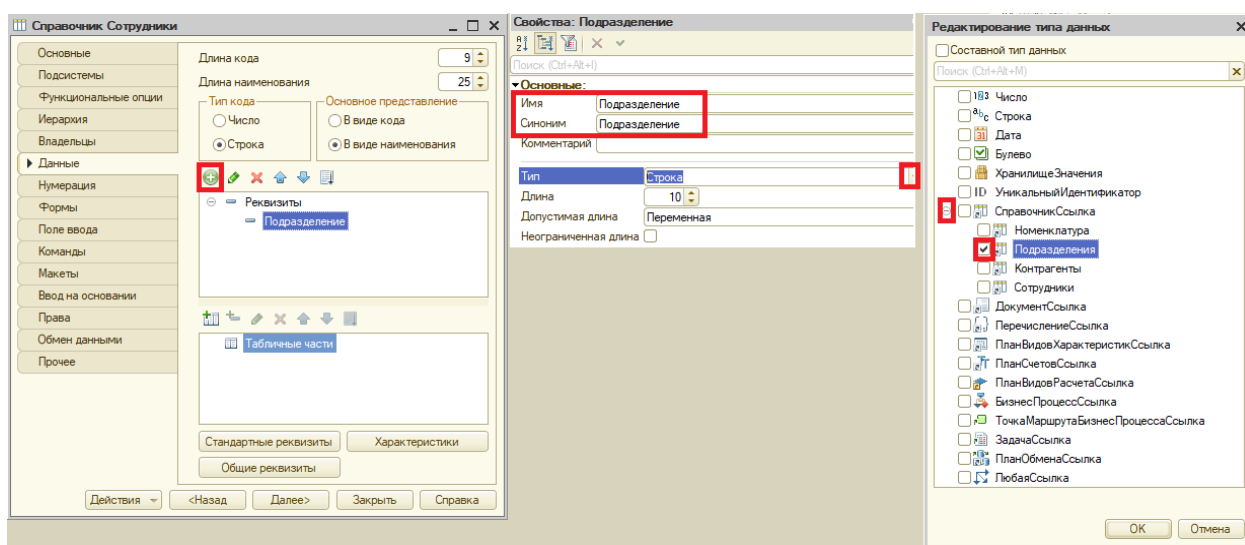


Рис. 3.23

Теория:

В этом случае вновь созданный реквизит будет заполняться значениями из справочника *Подразделения*, 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». Например, для сотрудника Иванов 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 *Продаж отдел*. 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 Иванов? Нет, не надо, поскольку в справочнике «Сотрудники» хранится не название отдела, а ссылка на элемент справочника, со всеми его реквизитами.

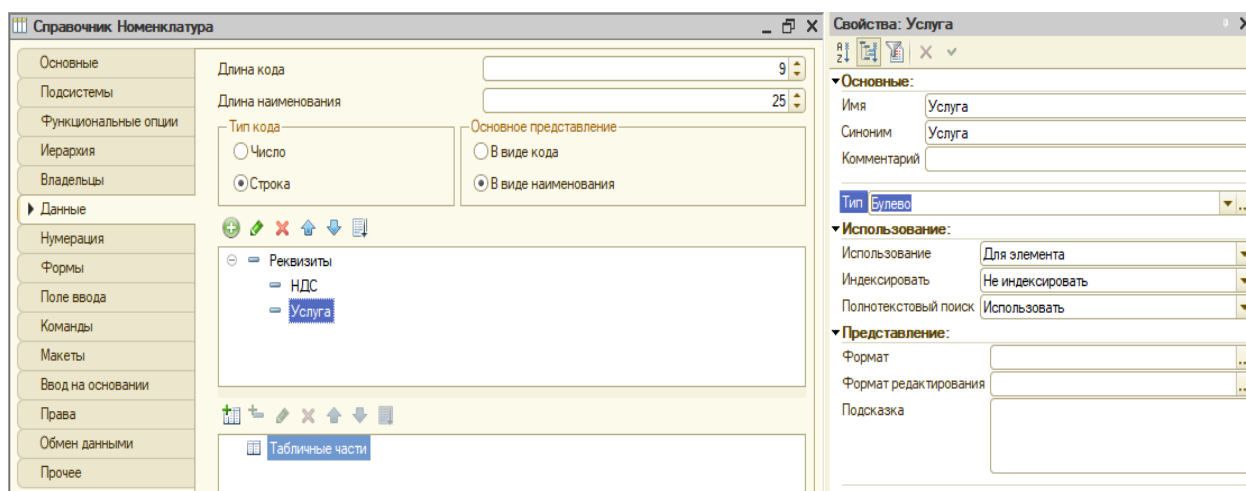
Практическая работа №4

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

Проверка выполнения практической работы №4

1. Создаем справочник Номенклатура. Добавляем реквизиты с именем НДС и типом число и Услуга с типом булево.

The screenshot shows the 'Справочник Номенклатура' (Catalog) application window. The 'Свойства: НДС' (Properties: VAT) dialog box is open, displaying the 'Основные' (Basic) tab. The 'Имя' (Name) and 'Синоним' (Synonym) fields are both set to 'НДС'. The 'Тип' (Type) is set to 'Число' (Number). The 'Длина' (Length) is 10 and 'Точность' (Precision) is 0. The 'Неотрицательное' (Non-negative) checkbox is unchecked. The 'Действия' (Actions) button is highlighted.



3.5. Табличная часть

Теория:

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

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

Согласно пожеланию нашего заказчика, необходимо хранить информацию об учебных заведениях и детях сотрудника. Данная информация самостоятельной не имеет ценности, 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.

Практика:

Для того чтобы создать табличную часть справочника «Сотрудники», на закладке *Данные* правой кнопкой мышки необходимо вызвать контекстное меню, в котором выбрать пункт *Добавить*. Тот же результат можно получить, вызвав контекстное меню правой кнопкой мышки в окне дерева объектов конфигурации на пункте *Табличные части выбранного справочника* (рис. 3.24).

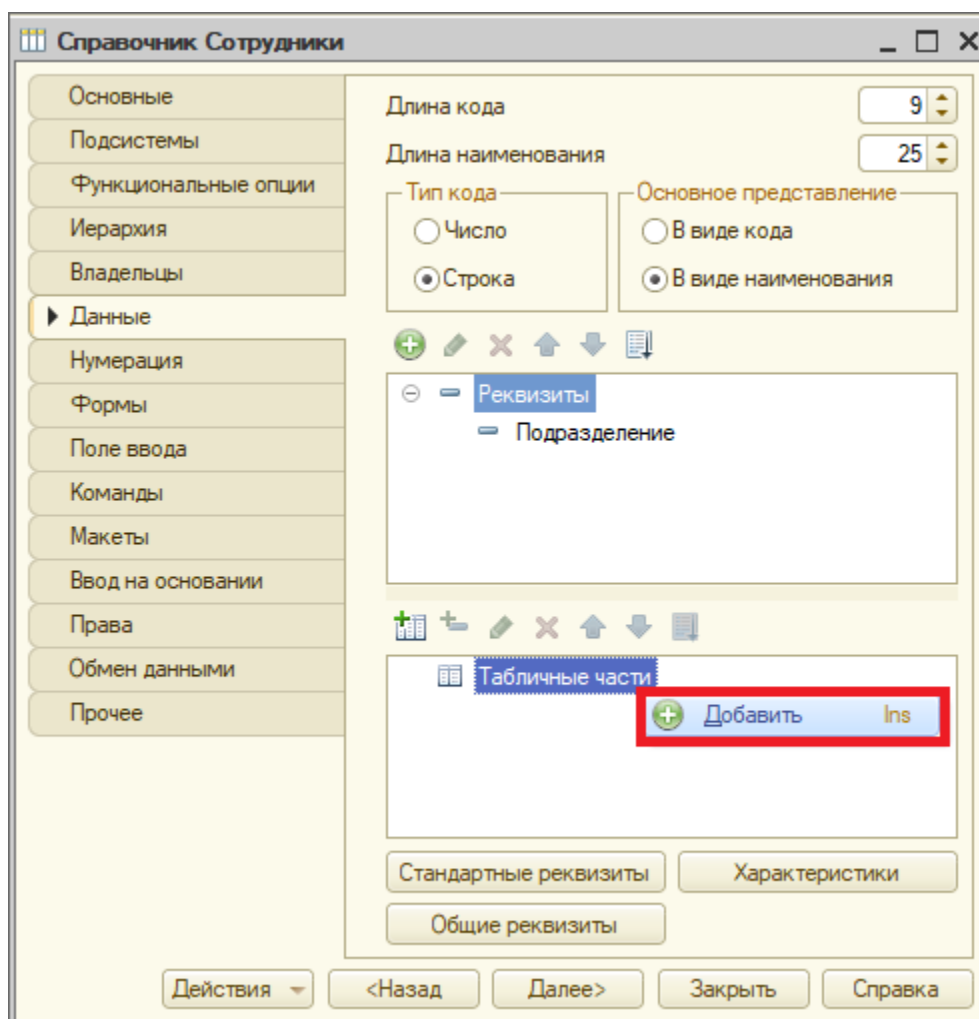


Рис. 3.24

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 те же условия, что и на имена справочников или констант.

Первую табличную часть назовем *Дети* (рис. 3.25). Определим, будет ли данная табличная часть относиться только к элементам, 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*.

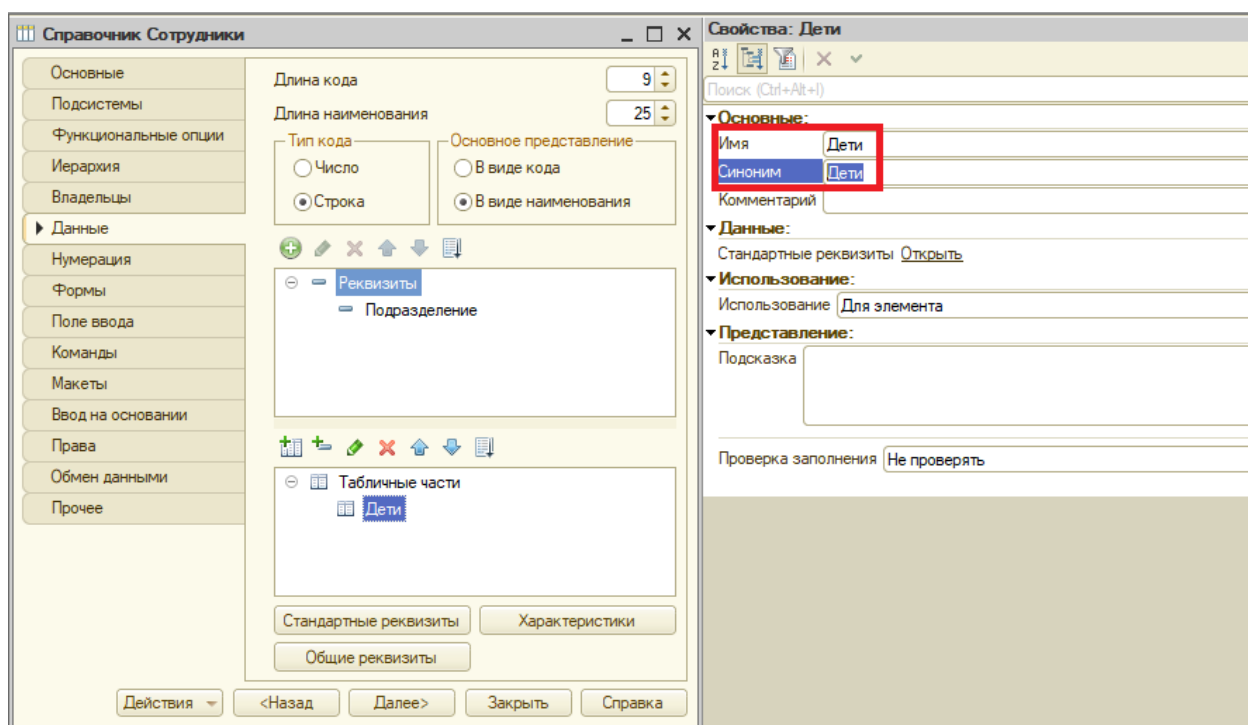


Рис. 3.25

Затем необходимо 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. 3.26).

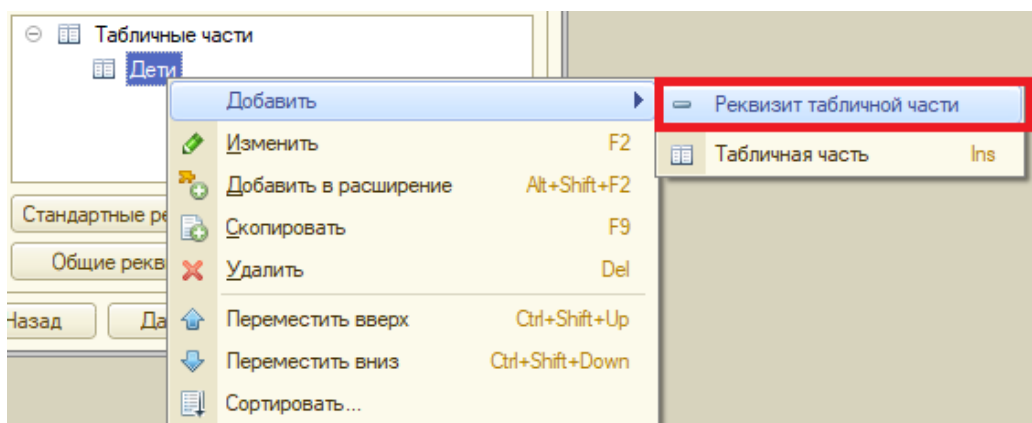


Рис. 3.26

Для хранения данных о фамилии, имени и отчестве ребенка создадим строковый реквизит с длиной 100 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. 3.27).

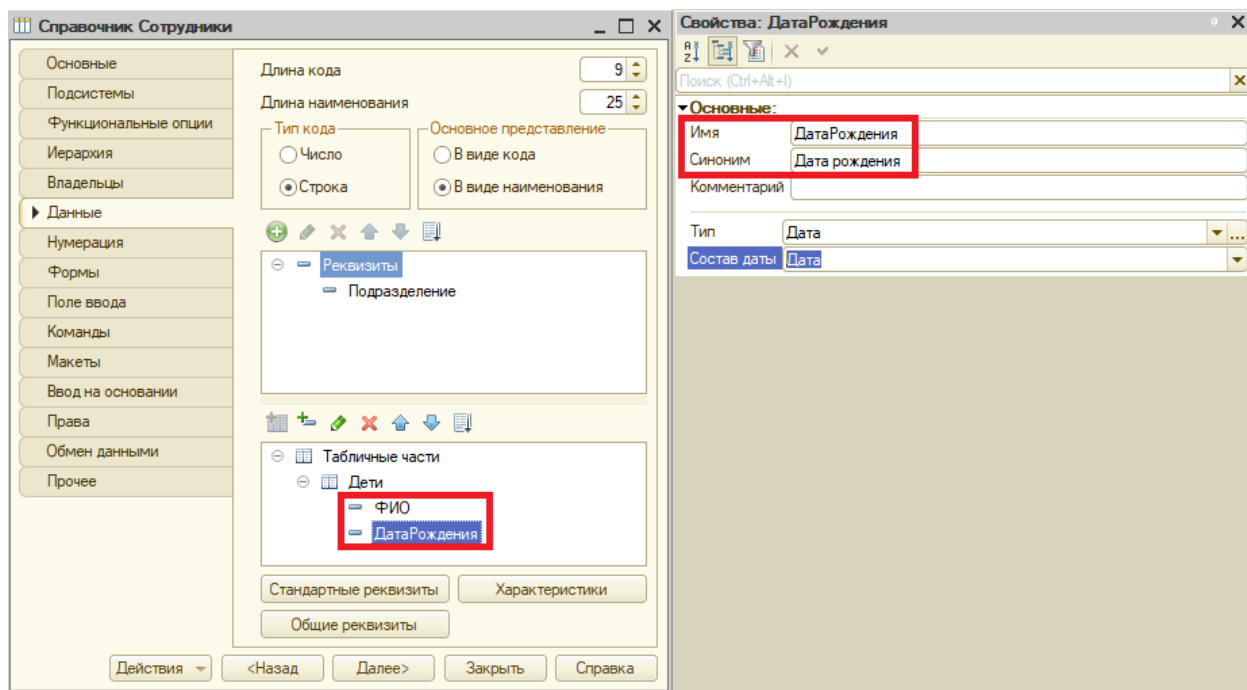


Рис. 3.27

Запустим конфигурацию в режиме предприятия и добавим сотрудника: Иванов Иван Иванович, Подразделение1, и двух детей Мария 30.05.2001, Василий 18.03.2009.

В результате таких действий в пользовательском режиме справочник станет выглядеть так, как указано на рис. 3.28.

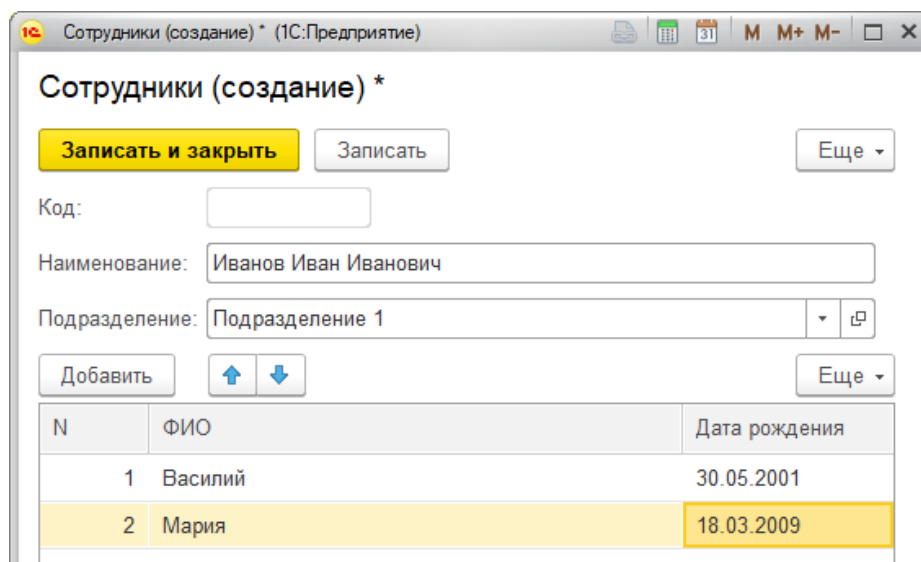


Рис. 3.28

Практическая работа №5

Добавьте в справочнике Сотрудники табличную часть Образование (рис. 3.29).

N	Учреждение	Дата поступления	Дата окончания	Специальность
1	ПТУ №11	01.09.1995	31.05.1999	Грузчик
2	РГГУ	01.09.2000	31.05.2003	Экономист

Рис. 3.29

Проверка выполнения практической работы №5

1. Создаем табличную часть в справочнике «Сотрудники» с названием «Образование». В табличную часть «Образование» добавляем следующие реквизиты: *Учреждение* (строка, длина 30), *ДатаПоступления* (тип Дата), *ДатаОкончания* (тип Дата), *Специальность*(строка, длина 20).

2. Запускаем конфигурацию в режиме предприятия, открываем справочник Сотрудники. Щелкаем два раза левой кнопкой мыши на Иванова Ивана Ивановича, добавляем ему образование.

Иванов Иван Иванович (Сотрудники) *

Записать и закрыть Записать Еще ▾

Код: 000000001

Наименование: Иванов Иван Иванович

Подразделение: Подразделение 1 ▾

Дети Образование

Добавить ↑ ↓ Еще ▾

N	Учреждение	Дата поступления	Дата окончания	Специальность
1	СГТУ им. Гагарина Ю.А.	01.09.1990	05.06.1995	Электротехника

4. ПЕРЕЧИСЛЕНИЕ

Теория:

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 двумя элементами. Можно, конечно, создать новый справочник, задать в нем predetermined элементы, назначить роли так, чтобы не могли задать новые элементы, однако перечисление использовать проще и быстрее как для программиста и системы, так и для конечного пользователя.

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

Практика:

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

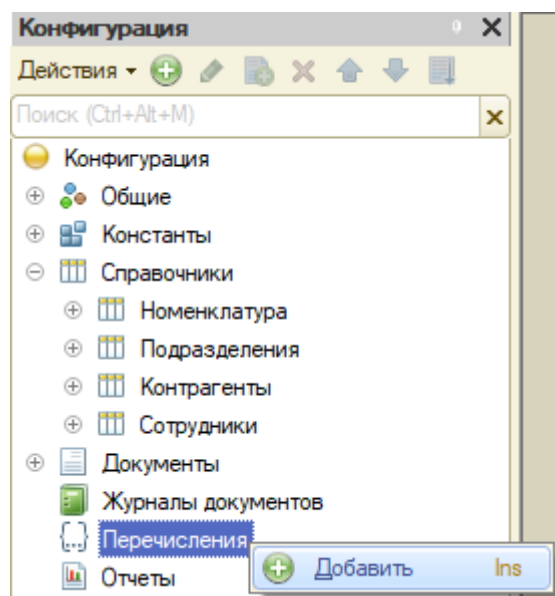


Рис. 4.1

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

На закладке *Данные* правой кнопкой мыши на пункте *Значения* вызываем контекстное меню и выбираем пункт *Добавить*, либо выбираем соответствующую кнопку в панели инструментов. Появляется уже знакомая нам палитра свойств (рис. 4.3), в которой задаем *Имя* – значения перечисления. Аналогично создаем следующее значение.

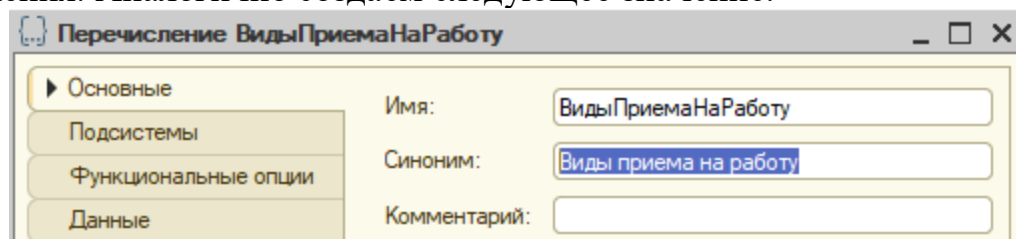


Рис. 4.2

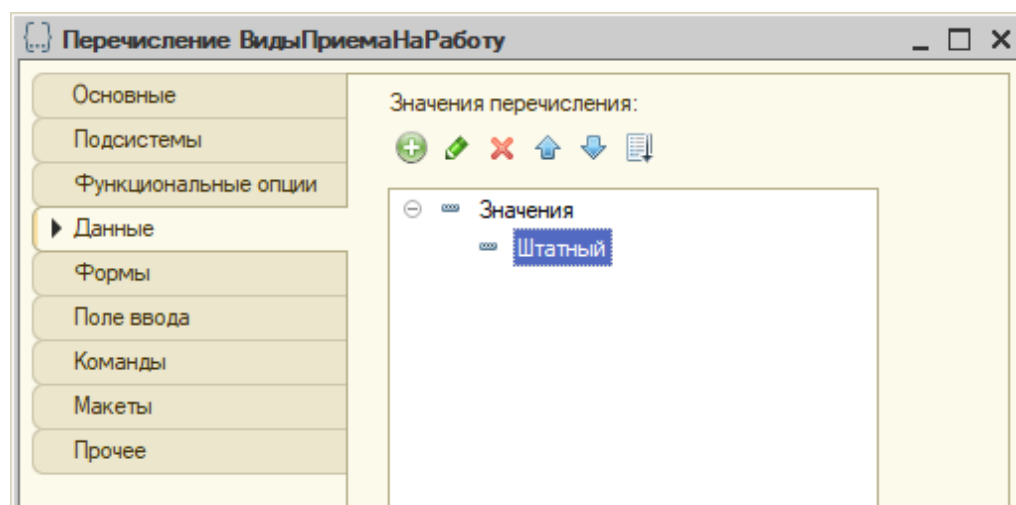


Рис. 4.3

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.Виды ПриемаНаРаботу* (рис. 4.4).

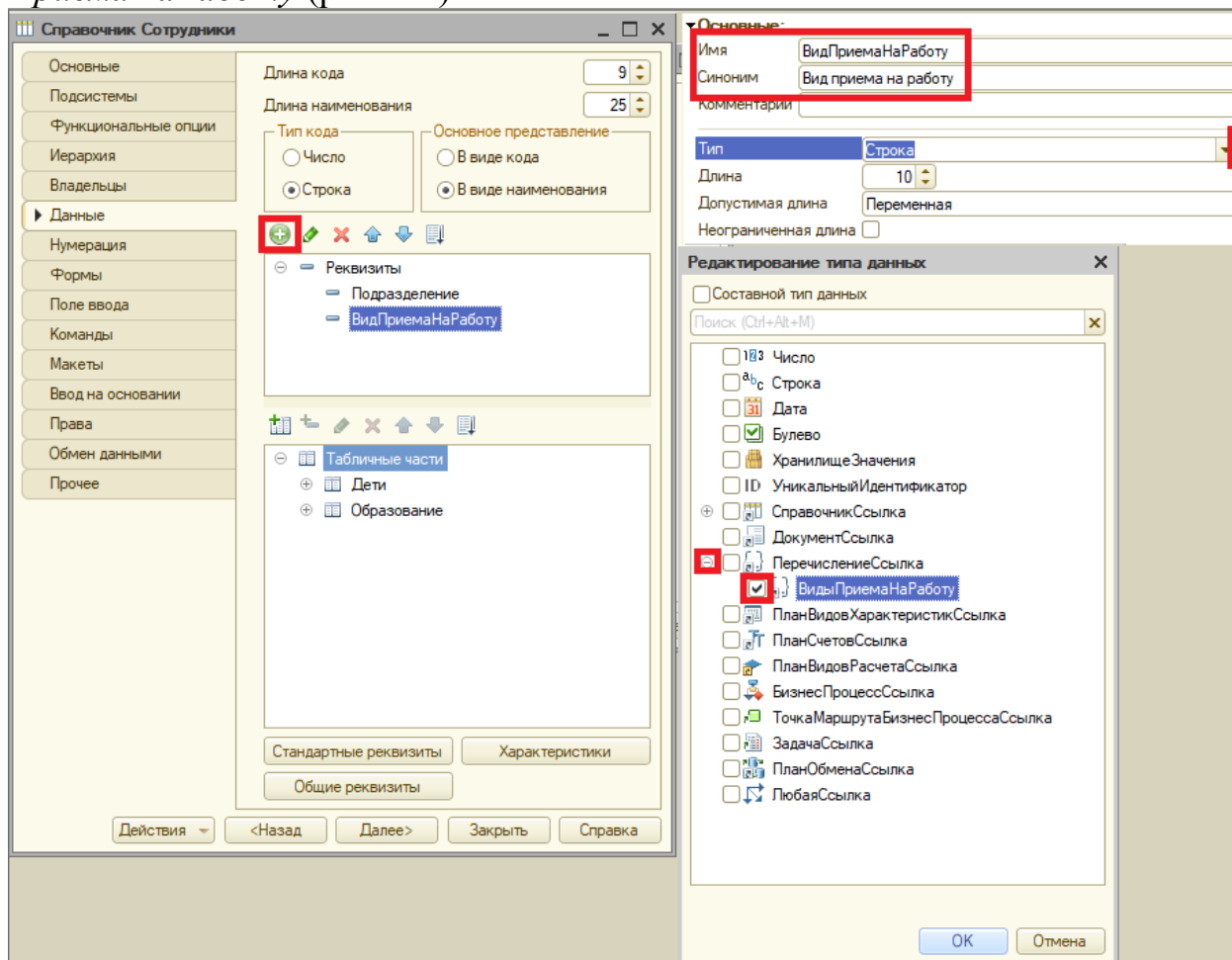


Рис 4.4

5. ЭЛЕМЕНТЫ УПРАВЛЕНИЯ ФОРМЫ

Теория:

Элементы формы предназначены для отображения и редактирования данных в форме. Также как и сама форма, элементы связаны с данными при помощи реквизитов формы (рис. 5.1).

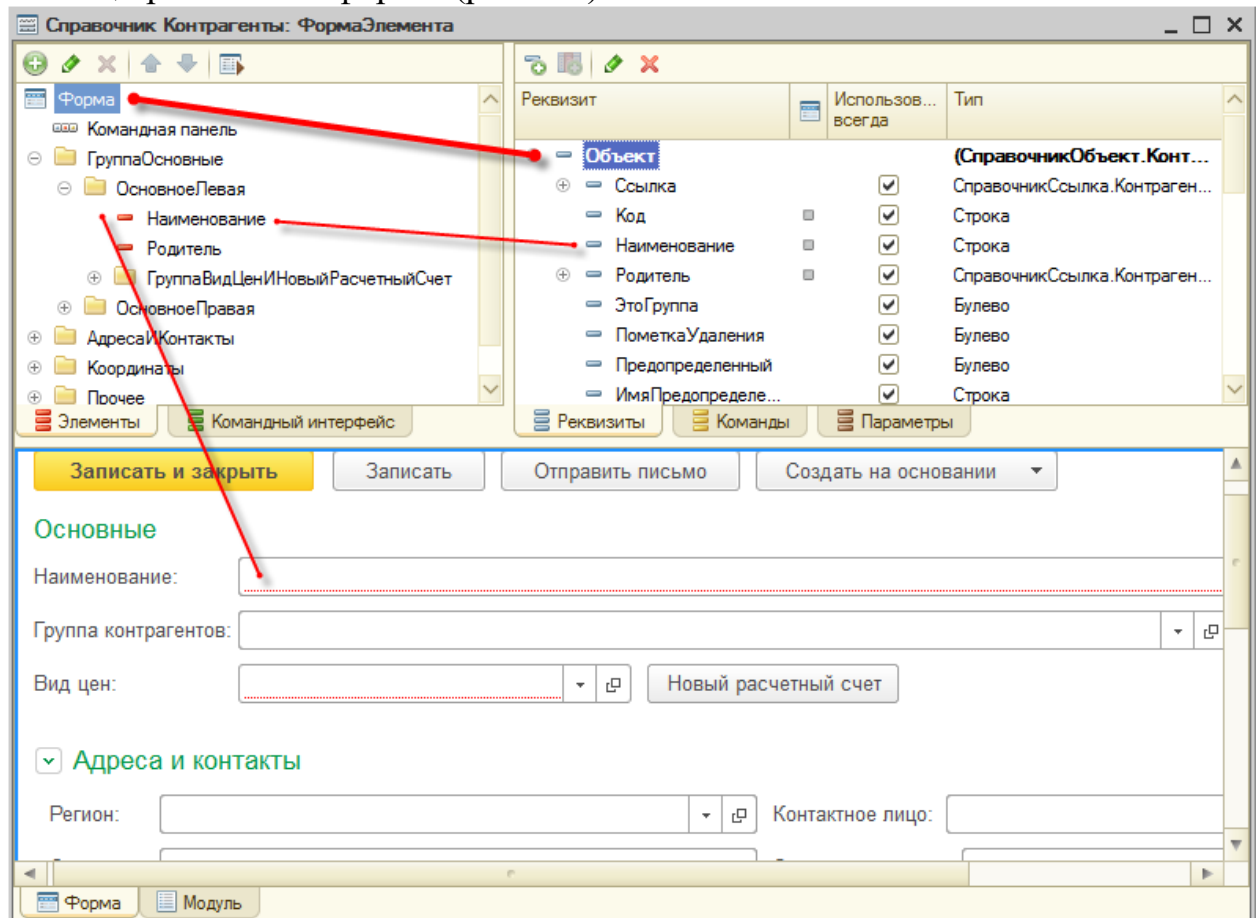


Рис. 5.1

Поведение элементов, определяемое данными

Благодаря этой связи у пользователя появляется возможность изменять данные, описанные в реквизитах формы. Кроме этого, связь элементов управления с данными определяет и поведение самих элементов управления.

Например, если элемент поле ввода, связать с данными, имеющими тип **Строка**, то оно будет иметь следующий вид (рис. 5.2):

Рис. 5.2

Если же поле ввода связать с данными, имеющими тип **Дата**, то внешний вид поля ввода изменится: появятся символы разделителей даты и дополнительная кнопка выбора (рис. 5.3):

Рис. 5.3

При нажатии на кнопку выбора будет открываться окно календаря, позволяющее выбрать нужную дату нажатием мыши (рис. 5.4):

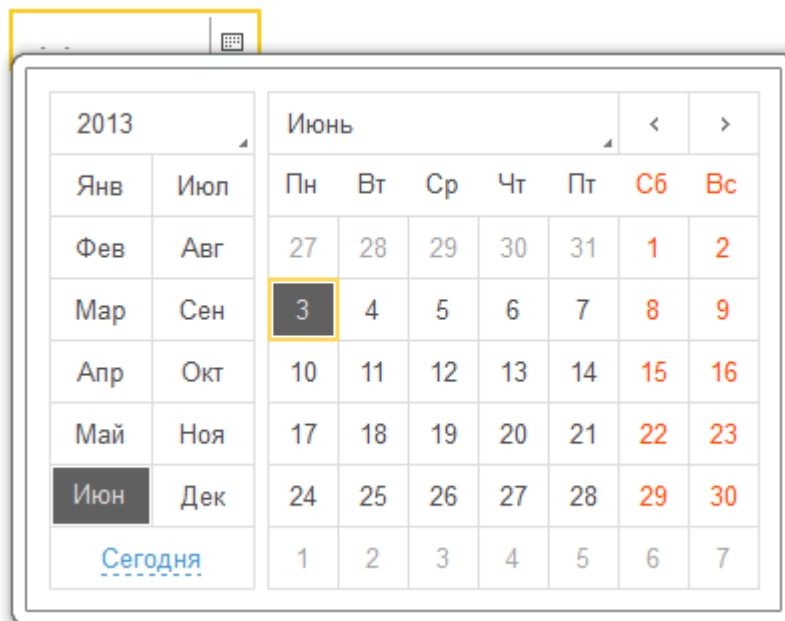


Рис. 5.4

Если же поле ввода связать с каким-либо объектом прикладного решения, то у него появятся две дополнительные кнопки: кнопка просмотра и кнопка открытия, имеющая уже другую пиктограмму, т.к. предназначена для выбора нужного объекта прикладного решения из формы списка (рис. 5.5):



Рис. 5.5

Элементы, ориентированные на бизнес-задачи

Поля ввода

Элементы, используемые в формах 1С:Предприятия, ориентированы на выполнение бизнес-задач. Например, поле ввода может иметь ряд дополнительных кнопок: выбора из списка, выбора, очистки, регулирования и открытия. Кроме этого, у поля ввода существует режим автоотметки незаполненного (подчеркивание красным пунктиром), который позволяет выделять поля, обязательные для заполнения пользователем (рис. 5.6):



Рис. 5.6

Действия, выполняемые при нажатии дополнительных кнопок поля ввода, могут быть различными, в зависимости от типа данных, которые отображаются в этом поле; при этом кнопка выбора будет иметь разные пиктограммы (рис. 5.7):

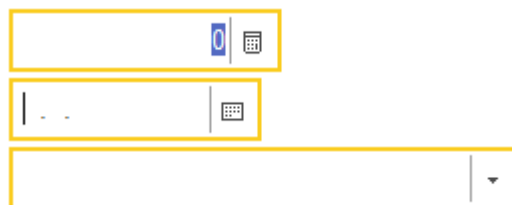


Рис. 5.7

Например, для поля ввода, содержащего число, нажатие на кнопку выбора будет приводить к открытию калькулятора (рис. 5.8):

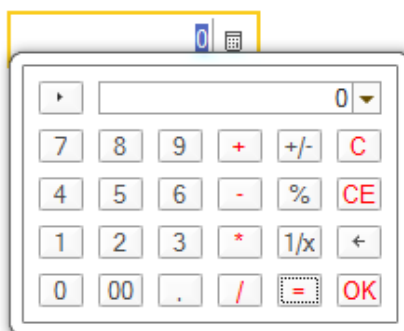


Рис. 5.8

А для поля ввода, содержащего дату, нажатие той же самой кнопки будет приводить к открытию календаря (рис. 5.9):

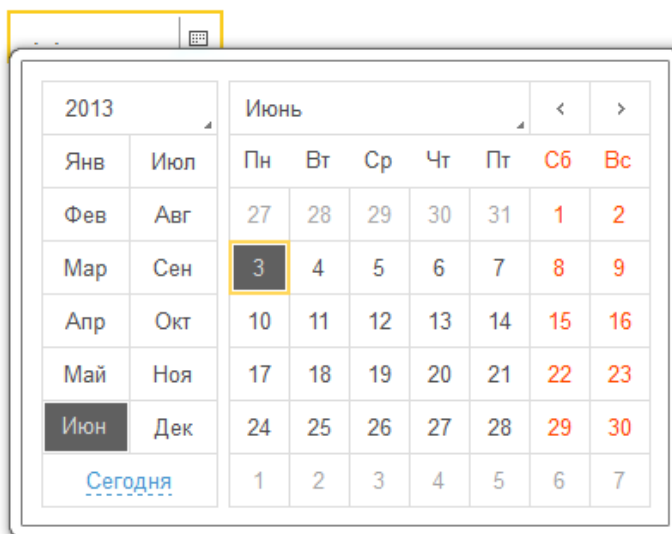


Рис. 5.9

Редактирование в одном элементе любых типов данных

Поскольку платформа 1С:Предприятия допускает хранение в полях базы данных различного типа, элементы управления также обеспечивают ввод и редактирование различных типов данных в одном элементе. Например, поле ввода, в которое могут быть введены данные различных типов, будет иметь следующий вид (рис. 5.10):



Рис. 5.10

При нажатии на кнопку выбора система откроет специальное окно для выбора типа данных, которые будут содержаться в этом поле (рис. 5.11):

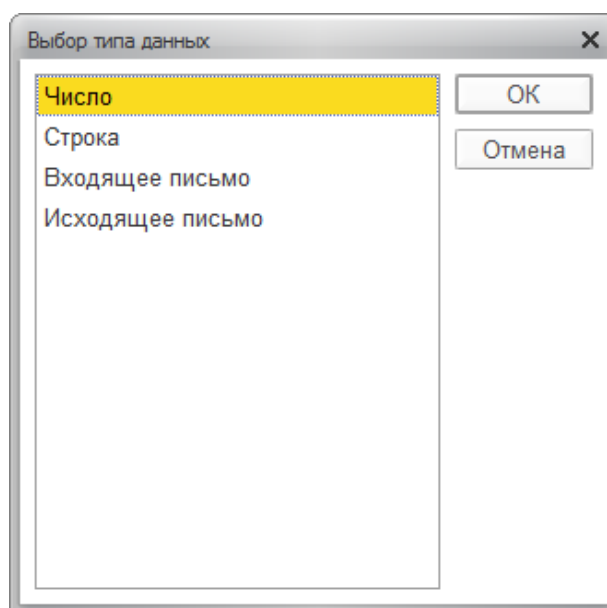


Рис. 5.11

Динамические списки

Элемент формы **Таблица** позволяет отображать и редактировать динамические списки, содержащие информацию из базы данных. Такие списки могут быть отсортированы простым нажатием мыши на заголовке нужной колонки и поддерживают различные варианты фильтрации и возможность интерактивного управления расположением колонок (рис. 5.12).

Дата	↓	Номер	Тип документа	Покупатель	Склад
 06.09.2012 9:30:12		000000015	Продажа	Шлюзовая ООО	Средний
 10.09.2012 14:21:40		000000016	Заказ	Магазин "Продукты"	Малый
 10.09.2012 14:21:55		000000016	Продажа	Магазин "Продукты"	Малый
 12.09.2012 15:30:25		000000017	Заказ	Попов Б.В. ИЧП	Малый
 12.09.2012 15:30:45		000000017	Продажа	Попов Б.В. ИЧП	Малый
 13.09.2012 10:15:00		000000018	Заказ	Магазин "Продукты"	Малый
 13.09.2012 14:39:08		000000019	Заказ	Магазин "Обувь"	Малый
 13.09.2012 15:01:22		000000020	Заказ	Магазин "Продукты"	Средний

Рис. 5.12

Практика:

Создадим Форму в справочнике «Сотрудники» (рис. 5.13).

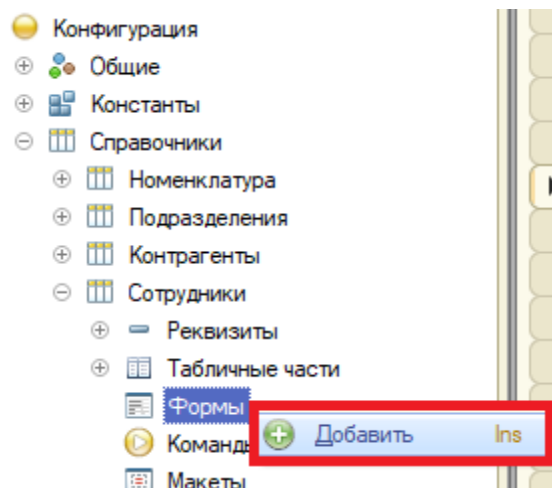


Рис. 5.13

Далее появится конструктор формы справочника, тут название оставляем без изменения и жмём сразу готово (рис. 5.14).

Рис. 5.14

После этого откроется окно, в котором мы можем изменять форму (рис. 5.15).

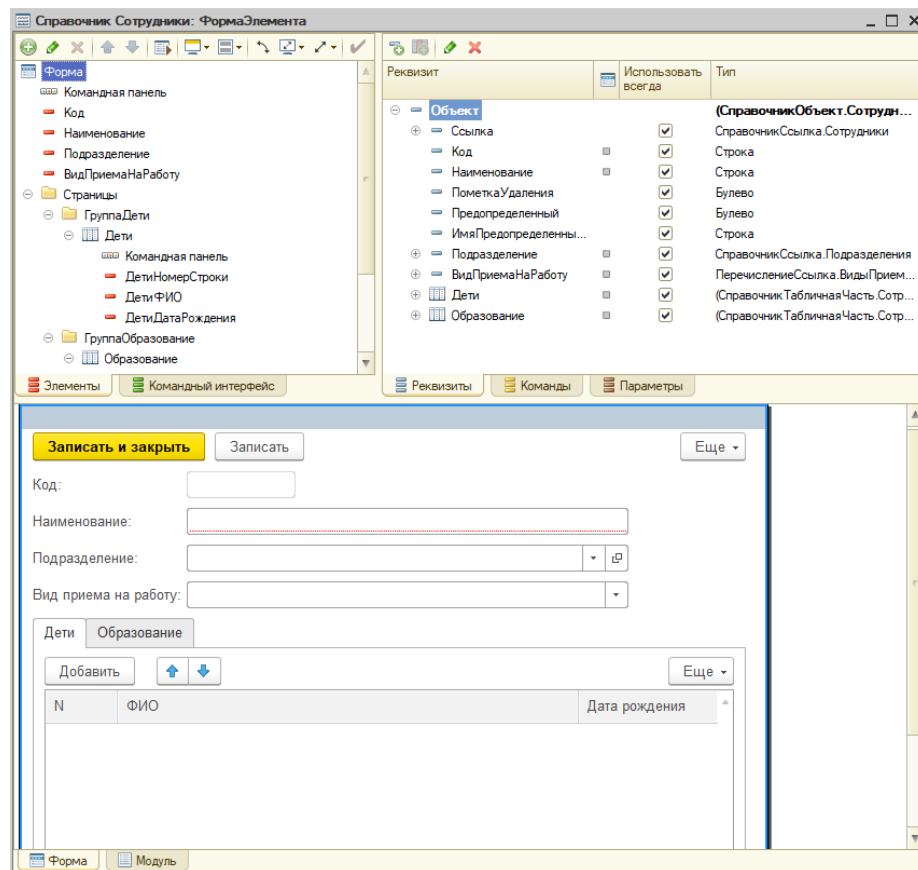


Рис. 5.15

Создадим группу «Группа - Обычная группа». Для этого выделим Форму в списке элементов и нажмём на Добавить. Откроется список элементов, выбираем нужный нам, жмём ок (рис. 5.16).

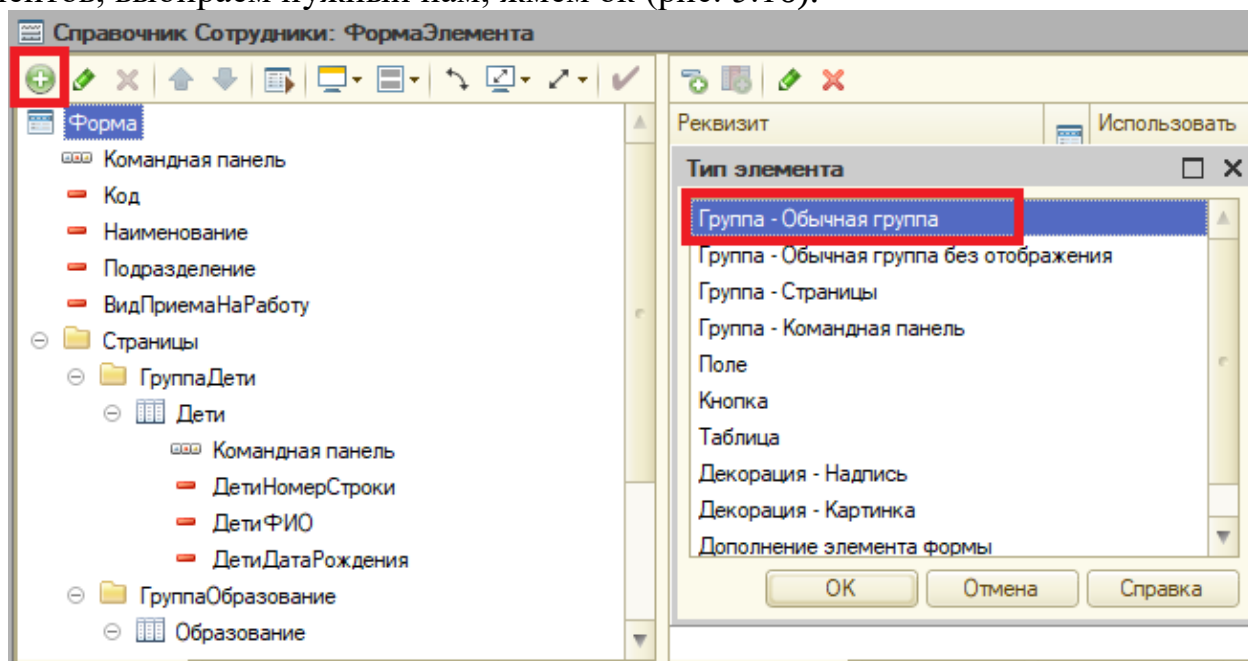


Рис. 5.16

Группа1 появится в списке. Нам нужно переместить ее вверх с помощью соответствующей кнопки и перенести с помощью «Drag and drop» в нее элементы Код и Наименование (рис. 5.17).

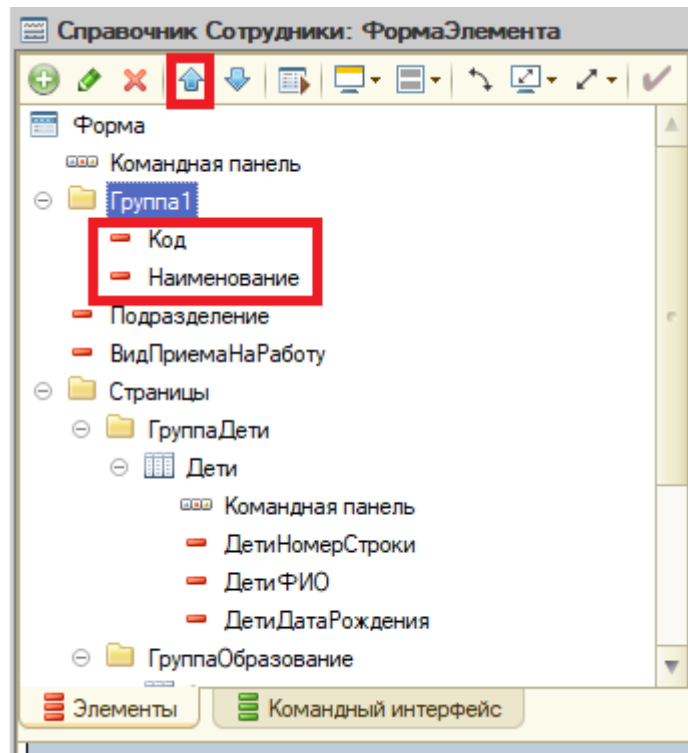


Рис. 5.17

Полученный результат можно увидеть ниже (рис. 5.18).

Записать и закрыть Записать Еще ▾

Код: Наименование:

Подразделение: ▾

Рис. 5.18

6. ОБРАБОТЧИК СОБЫТИЯ

Теория:

Если необходимо выполнять какие-либо действия автоматически при совершении пользователем каких-то действий, необходимо воспользоваться механизмом обработчика событий.

Практика:

В справочнике Контрагенты у нас хранится 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, поэтому необходимо самостоятельно написать процедуру, которая должна будет выполняться системой и связать эту процедуру с тем событием, которое обрабатывается.

Для этого создадим Форму у справочника *Контрагенты*. В конструкторе оставим все без изменения и нажмём готово. Откроется форма (рис. 6.1).

Рис. 6.1

Найдём раздел события в свойства поля «Наименование» (рис. 6.2).

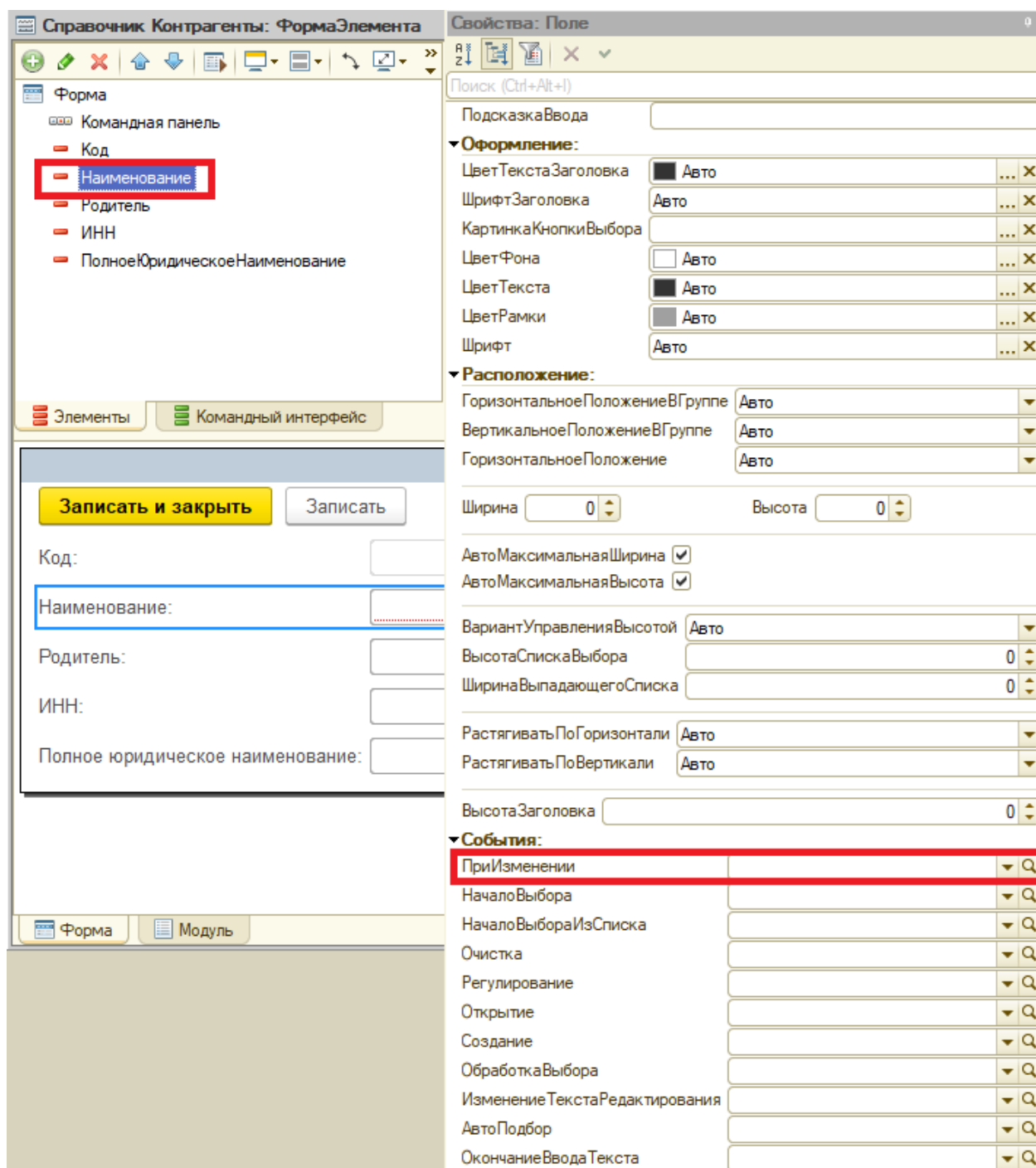


Рис. 6.2

Далее создадим обработчик события «ПриИзменении». Для этого нажмем на значок лупы и выберем в появившемся окне «Создать на клиенте» (рис. 6.3).

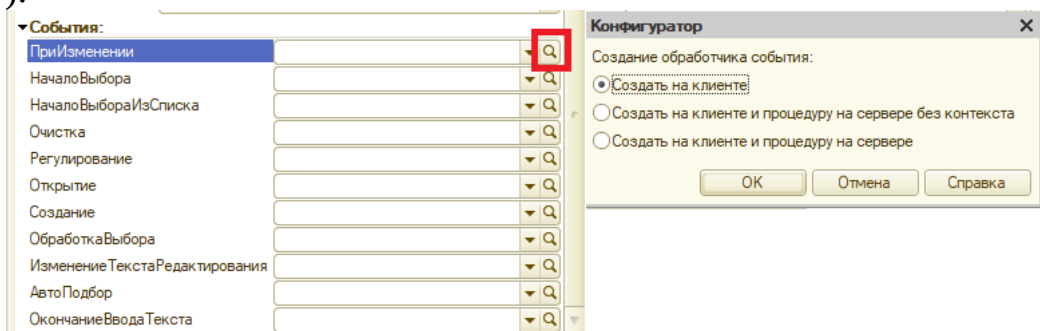


Рис. 6.3

Далее в появившемся окне пропишем следующий код.
Текст процедуры обработчика события:

Практическая работа №6

&НаКлиенте

Процедура НаименованиеПриИзменении(Элемент)

Если Объект.ПолноеЮридическоеНаименование = "" Тогда

 Объект.ПолноеЮридическоеНаименование =

Объект.Наименование;

КонецЕсли;

КонецПроцедуры

Проверка выполнения практической работы №6

1. Для проверки работы кода необходимо нажать на соответствующий значок. После этого можно закрыть данное окно (рис. 6.4).

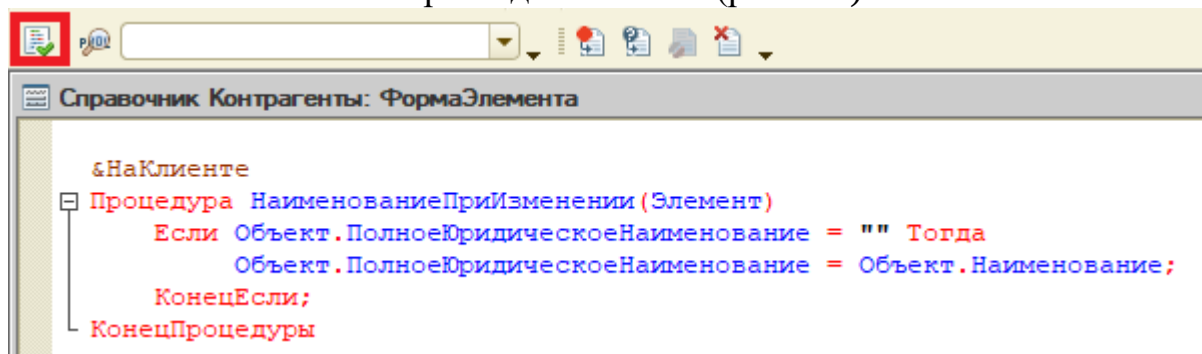


Рис. 6.4

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

Рассмотрим следующую ситуацию: необходимо, чтобы у услуг ставка НДС была только 18%. В этом случае необходимо проверить при сохранении элемента, является ли данный элемент справочника услугой. 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 для этого создадим форму элемента справочника Номенклатура (рис 6.5)

И в модуле напомним процедуру:

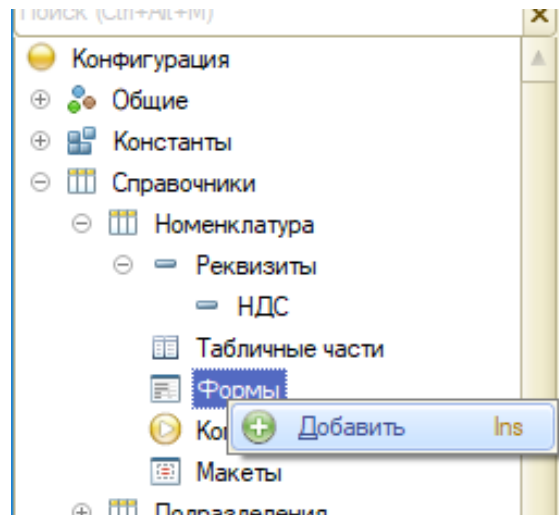


Рис. 6.5

Практическая работа №7

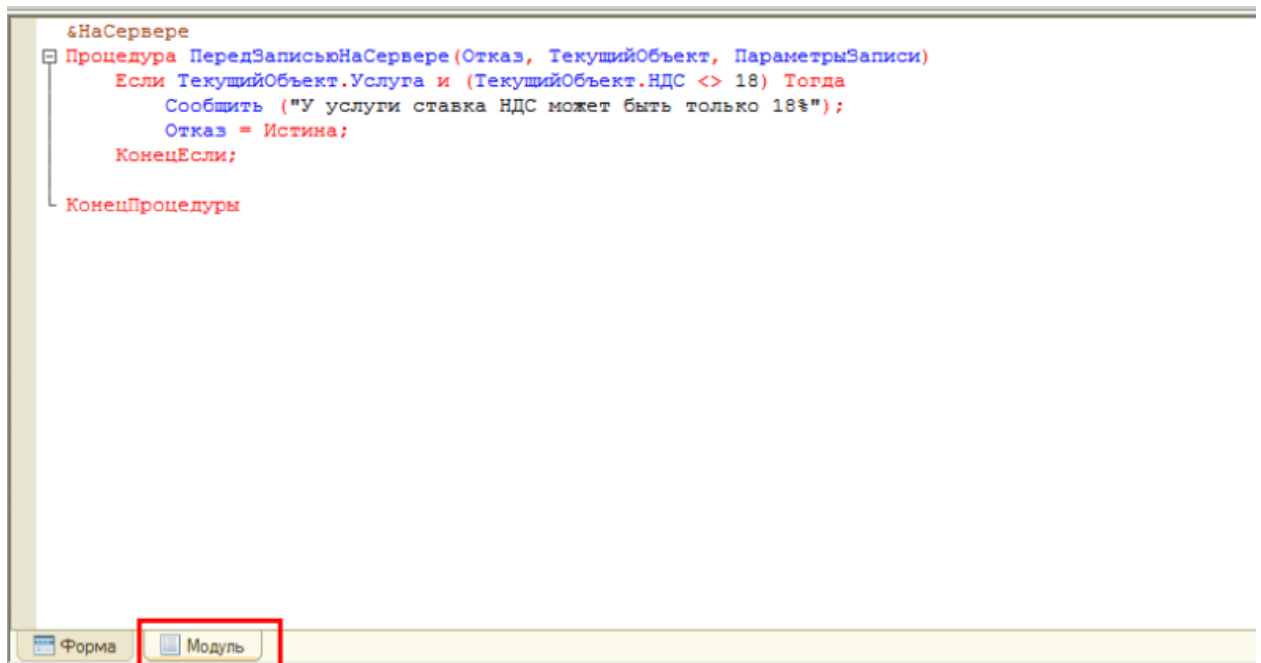
&НаСервере

Процедура ПередЗаписьюНаСервере(Отказ, ТекущийОбъект, ПараметрыЗаписи)

Если ТекущийОбъект.Услуга и (ТекущийОбъект.НДС <> 18) Тогда
 Сообщить ("У услуги ставка НДС может быть только 18%");
 Отказ = Истина;
 КонецЕсли;

КонецПроцедуры

В модуль вводим текст, что дан выше в рамочке (Рис 6.6)



(Рис 6.6)

И для того, чтобы это корректно работало не забываем в свойствах формы ПередЗаписьюНаСервере выбрать нужную процедуру(Рис. 6.7-6.8)

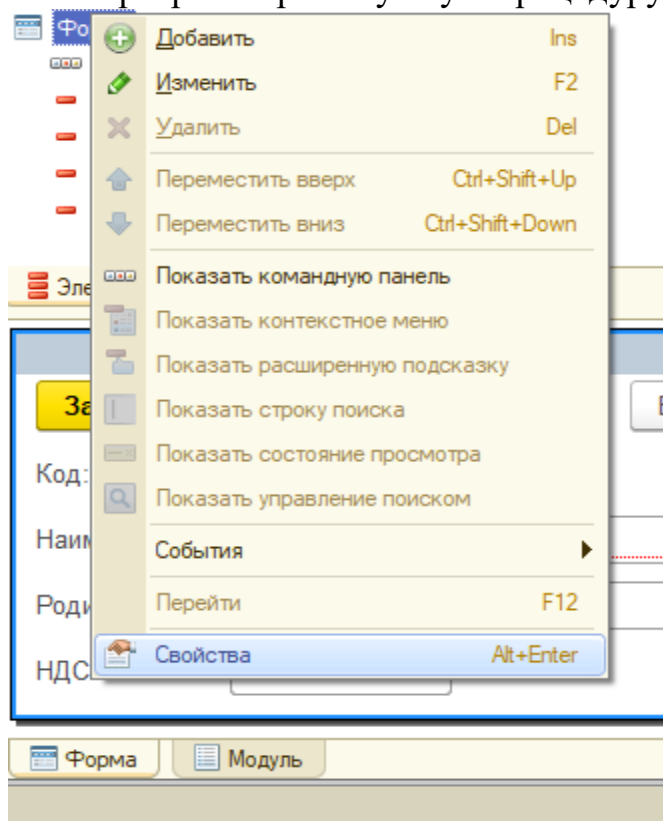


Рис. 6.7

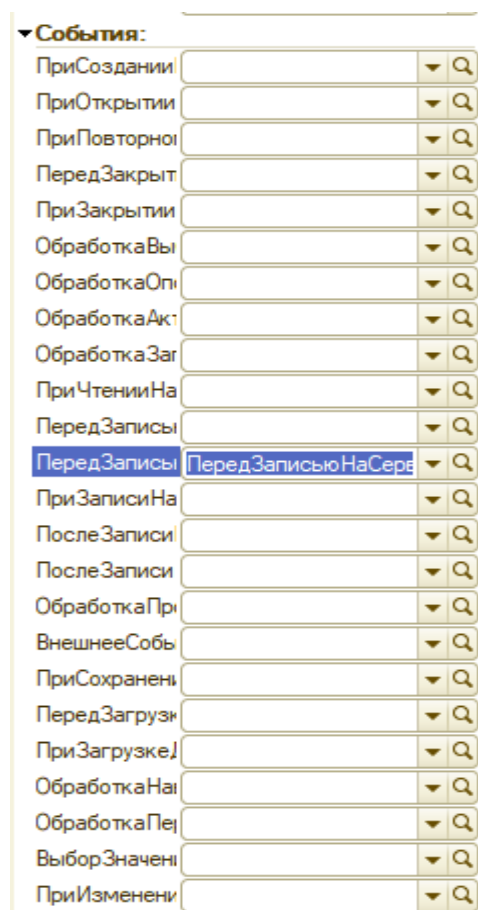


Рис. 6.8

☆ Ремонт кондиционеров (Товары и услуги) *

Записать и закрыть Записать Еще ▾

Код: 000000015

Наименование: Ремонт кондиционеров

Родитель: Услуги ▾ □

НДС: 10

Услуга: ☒

Сообщения: ×

— У услуги ставка НДС может быть только 18%

7. Документ

7.1. Реквизиты шапки

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, которые встречаются один раз, и те, которые повторяются, соответственно, реквизиты документа делятся на реквизиты шапки 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.

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 шапки.

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

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

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

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

Периодичность – обычно нумерация документов идет не сквозная, а в пределах какого-то временного интервала, чаще всего года. Для реализации

этого в системе предусмотрен соответствующий механизм. Предусмотрены следующие варианты периодичности номеров:

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

Для документа **РасходДенег** изменим длину номера на 6 знаков и периодичность на значение В пределах года (Рис 7.2)

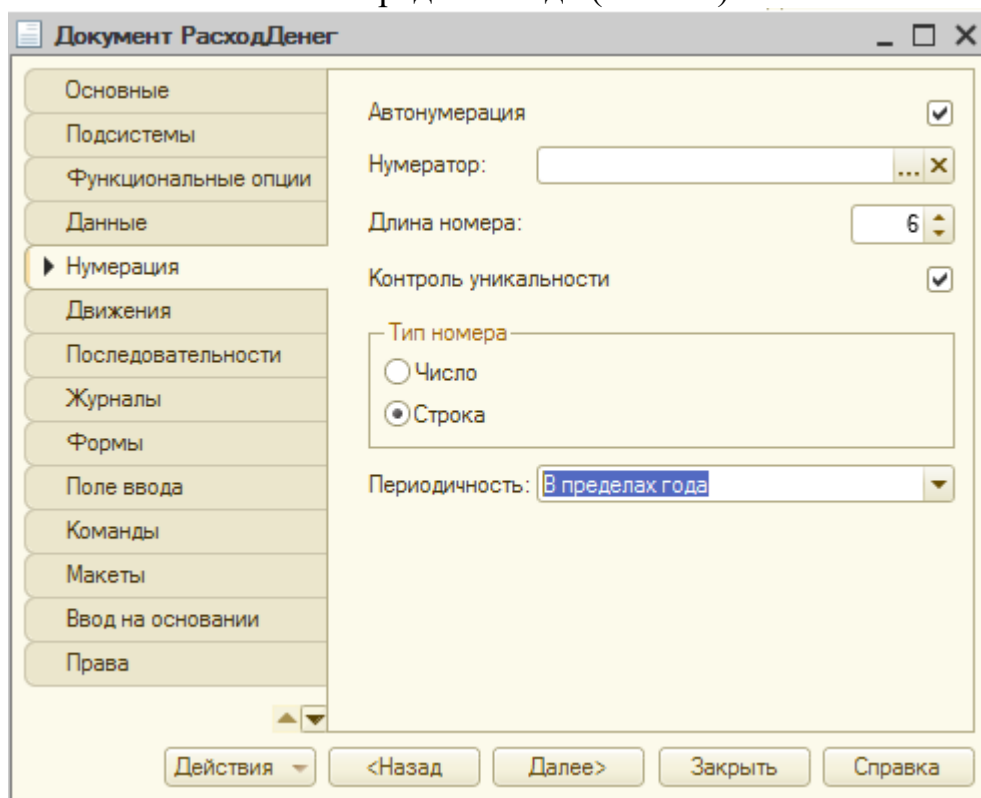


Рис. 7.2

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

Контрагент,
Договор,
Сумма.

Преджде чем продолжить, создадим справочник «Договор»(Рис.7.3), чтобы мы могли потом на него ссылаться. Нам понадобится 4 реквизита: **Котрагент**(тип: СправочникСсылка.Контрагенты), **ТоварИлиУслуга**(тип: СправочникСсылка.Номенклатура), **Дата**(тип:Дата),**Сумма**(Тип:Число, Длина:10, Точность:2, Неотрицательное).

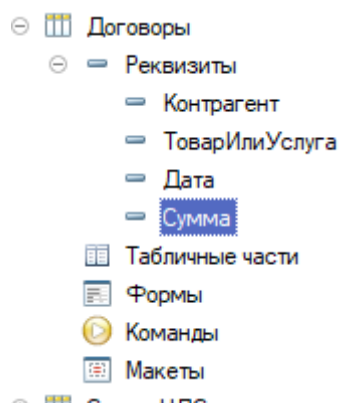


Рис. 7.3

На закладке Данные, точно так же, как это делалось и для справочника, добавляем новые реквизиты. (Рис 7.4) 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**.

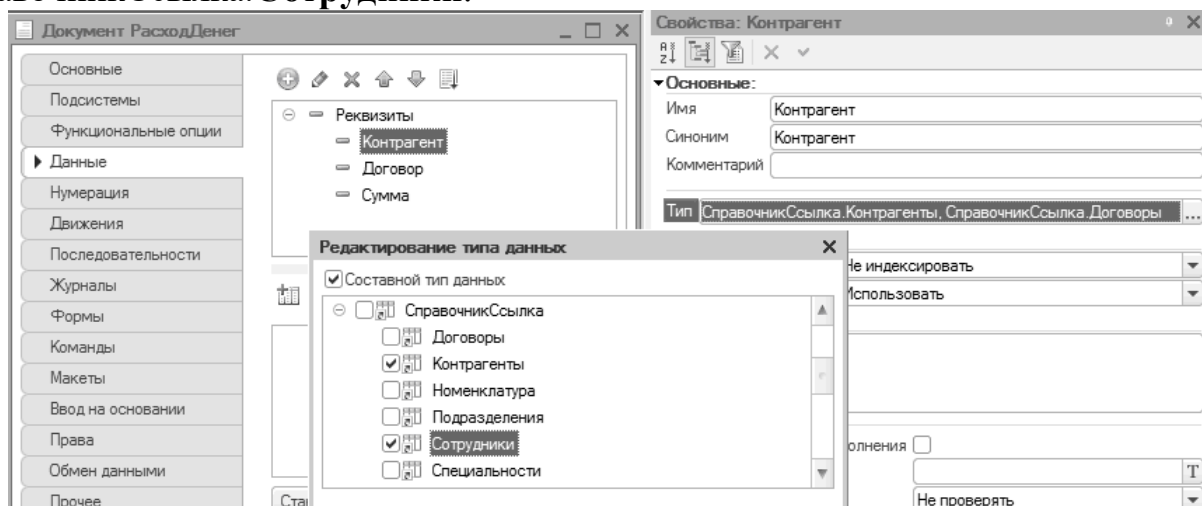


Рис. 7.3

7.2. Формы документа

Точно так же, как и при работе со справочником, пользователь работает не напрямую с объектом «Документ», а посредством экранной формы. Существует несколько форм документов (Рис. 7.4)

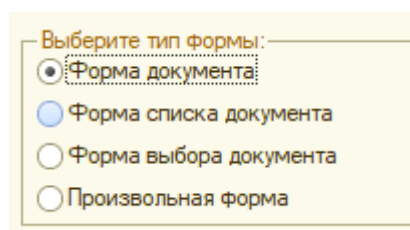


Рис. 7.4

Соответственно, при работе с одним документом используется Форма документа (Рис. 7.5), а для работы с несколькими документами данного вида используется Форма списка (Рис. 7.6).

Рис. 7.5

Дата	Номер	Контрагент	Договор	Сумма
23.12.2014 ...	000001	Красный богатырь ООО	01КрБг	10 000,00
15.01.2015 ...	000003	Красный богатырь ООО	01КрБг	9 000,00
20.01.2015 ...	000004	Добрый молодец ООО	01ДбМл	8 000,00
30.12.2014 ...	000002	Добрый молодец ООО	01ДбМл	4 000,00

Рис. 7.6

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

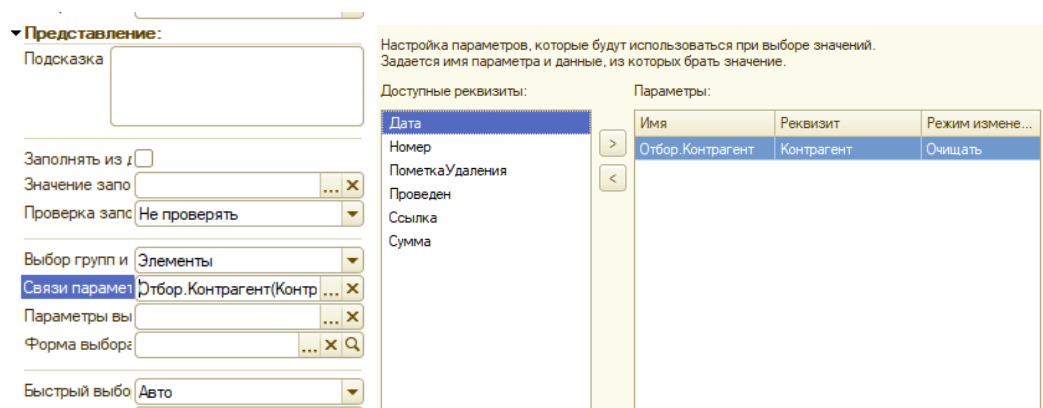


Рис. 7.7

Практическая работа №8

Создайте документ **Поступление денежных средств**

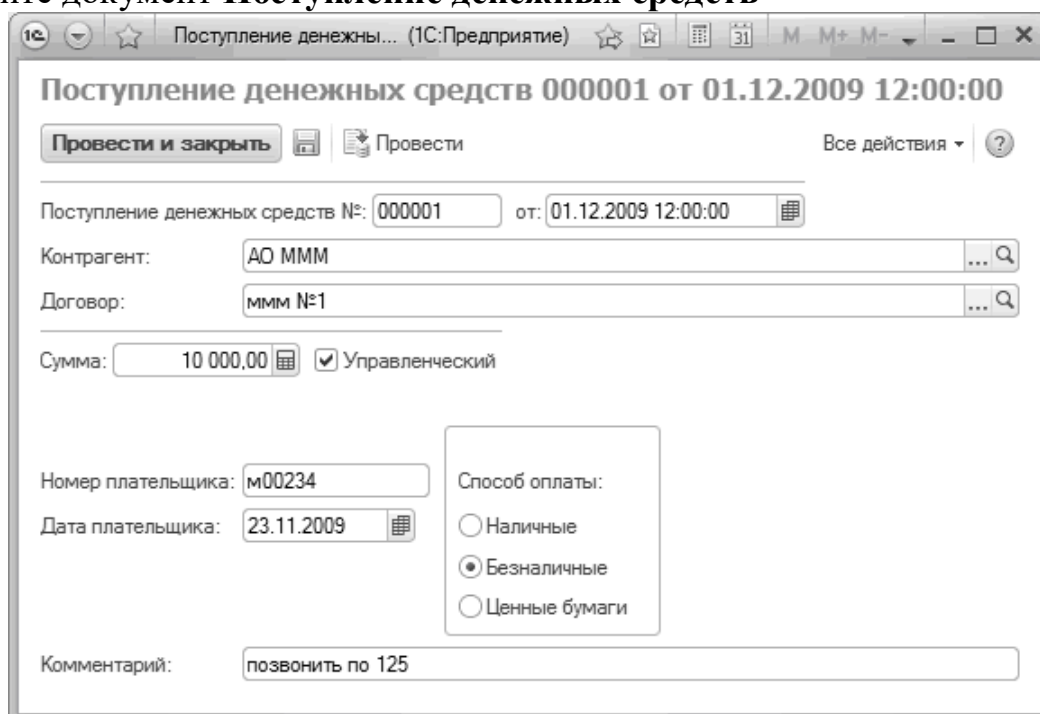
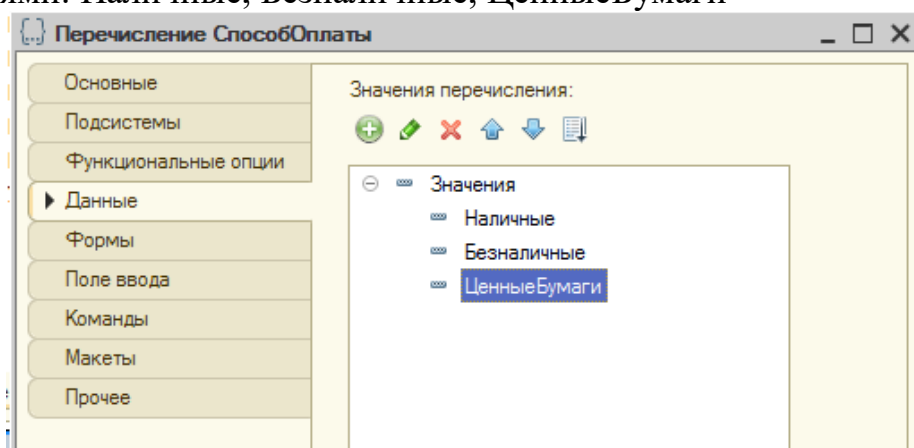


Рис 7.14

Для Способа оплаты создайте отдельное перечисление СпособОплаты с 3 значениями: Наличные, Безналичные, ЦенныеБумаги



Для документа Поступление денежных средств создайте следующие реквизиты:

Контрагент (*Справочник.Контрагенты*)

Договор (*Справочник.Договоры*)

Сумма Число (10,2)

НомерПлательщика Строка

ДатаПлательщика Дата

СпособОплаты (*Перечисление.СпособОплаты*)

Комментарий Строка (произвольная)

Управленческий Булево (флажок)

Чтобы на форме можно было выбирать «способ оплаты», создадим форму документа и в свойствах поля «Способ оплаты» в «список выбора» выбираем 3 доступных значения (Рис. 7.15)

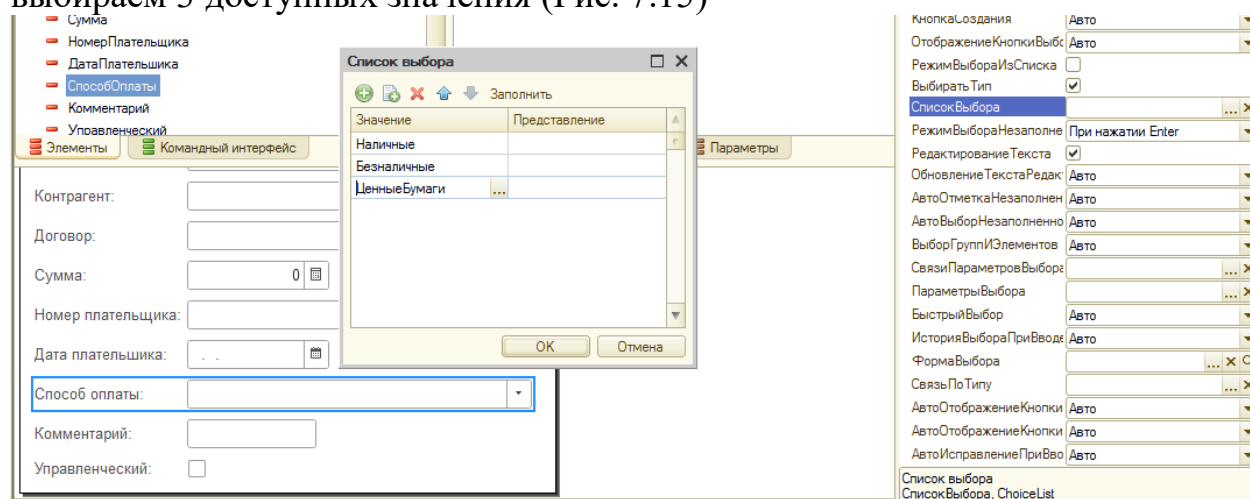


Рис 7.15

в поле «вид» выбираем Поле переключателя (Рис. 7.16)

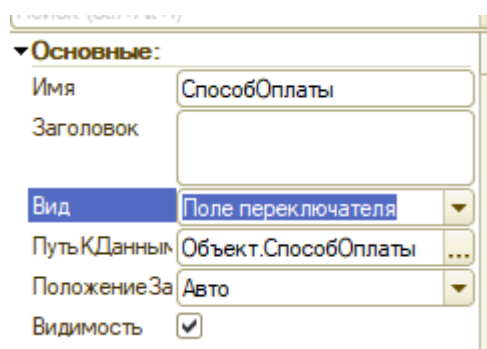


Рис 7.16

Обратите внимание, что пользователь имеет возможность (в определенных пределах) изменить внешний вид формы самостоятельно. 8.15 Для этого в форме документа Создадим «Группа-Обычная группа» и туда группируем нужные нам элементы (Рис 7.17)

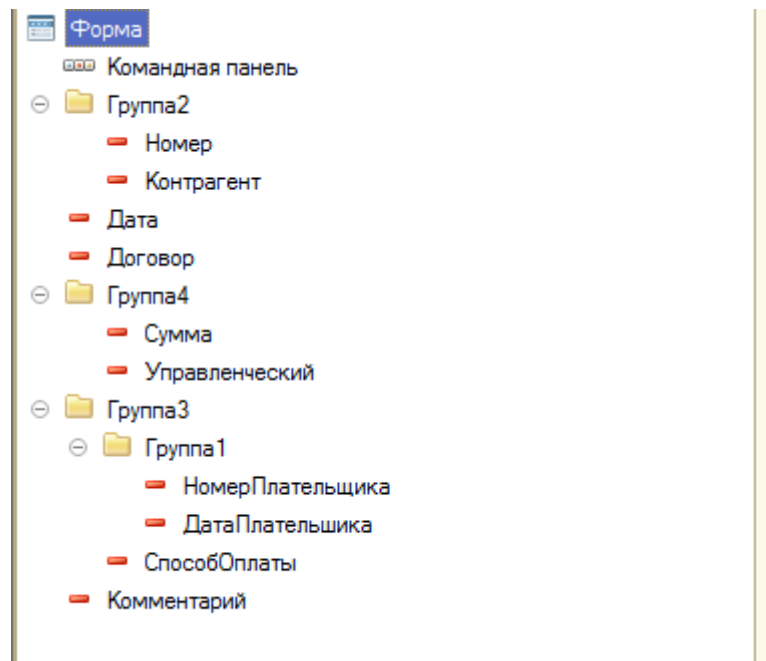


Рис. 7.17

Чтобы реквизиты исчезали, когда меняется способ оплаты, задействуем обработчик события ПриИзменении для Поля «СпособОплаты» (Рис 7.18) Кроме того, необходимо использовать обработчик события ПриОткрытии для всей формы (Рис. 7.19). Но прежде чем выбирать мы в модуле Формы Документа введем код:

&НаКлиенте

Процедура СпособОплатыПриИзменении(Элемент)

Элементы.Группа1.Видимость=ПолучитьВидимость(Объект.СпособОплаты);
КонецПроцедуры

&НаСервере

Функция ПолучитьВидимость(Способ)

Если Способ = Перечисления.СпособыОплаты.Безналичные Тогда

Воз=Истина;

Иначе

Воз=Ложь;

КонецЕсли;

Возврат Воз;

КонецФункции

&НаКлиенте

Процедура ПриОткрытии(Отказ)

Элементы.Группа1.Видимость=ПолучитьВидимость(Объект.СпособОплаты);
КонецПроцедуры

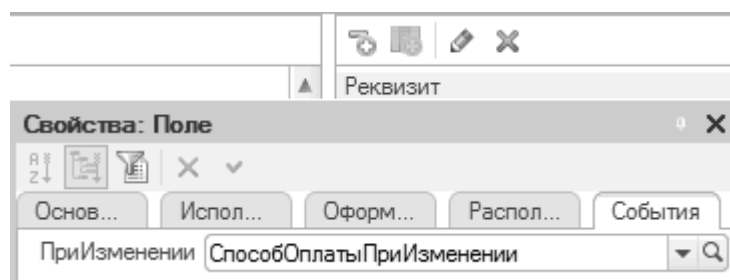


Рис 7.18

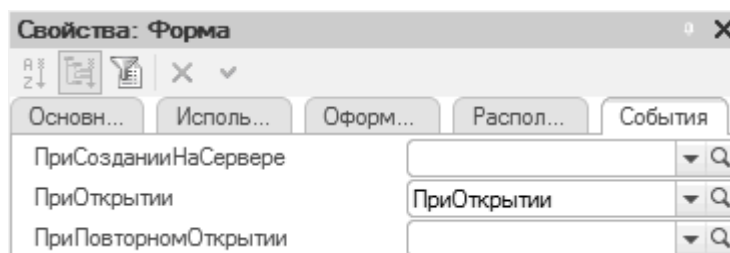


Рис.7.19

Теперь при выборе способа оплаты не безналичного реквизиты плательщика скрываются (Рис 7.20), а при выборе «безналичные» появляются.(Рис. 7.21)

← → ☆ Поступление денежных средств (создание) *

Провести и закрыть Записать Провести

Номер: Контрагент:

Дата:

Договор:

Сумма: Управленческий: ☐

Номер плательщика: Дата плательщика:

Комментарий:

Способ оплаты:

☐ Наличные

☒ Безналичные

☐ Ценные бумаги

Рис. 7.20

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.

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. 7.22).

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 вывести итог колонки Сумма.

Прежде чем приступить к созданию табличной части, необходимо создать сам документ. Для этого в конфигураторе в дереве метаданных находим ветвь *Документы* и с помощью контекстного меню добавляем новый документ, которому присваиваем имя *ПриходнаяНакладная*. После этого в этом же документе создаём реквизиты шапки *Контрагент* с типом *СправочникСсылка.Контрагенты* и *Договор* с типом *СправочникСсылка.Договоры*.

Создайте табличную часть *Товары* со следующими реквизитами: *Товар* (*СправочникСсылка.Номенклатура*), *Количество* (*число, 10*), *Цена* (*число, 2*), *Сумма* (*число, 2, неотрицательное*).

Для расчета суммы необходимо задействовать обработчики событий *ПриИзменении* для количества и цены (Рис. 7.23). Хотя результат этих обработчиков должен быть одинаков, для удобства приведено два разных способа написания этого действия.

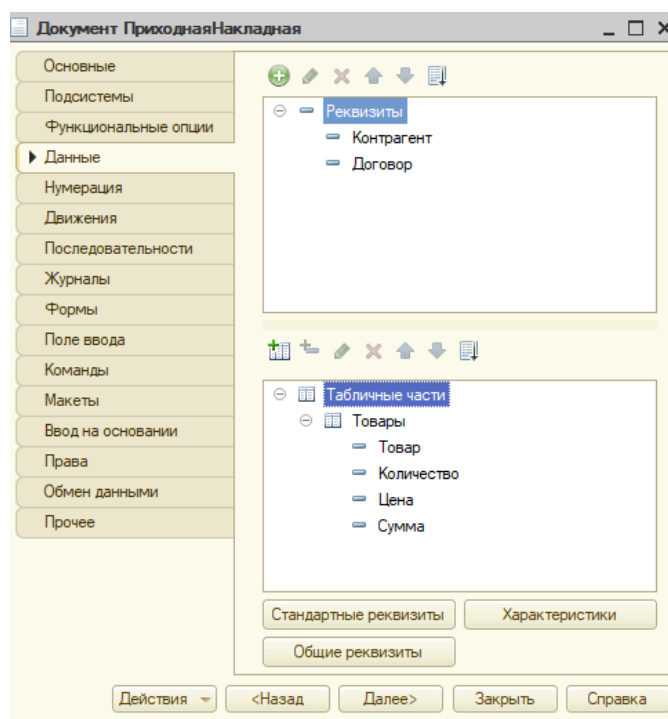


Рис 7.22

В модуле формы документа вставляем код:

&НаКлиенте

Процедура ТоварыКоличествоПриИзменении(Элемент)

Элементы.Товары.ТекущиеДанные.Сумма =

Элементы.Товары.ТекущиеДанные.Количество*Элементы.Товары.ТекущиеДанные.Цена;

КонецПроцедуры

&НаКлиенте
 Процедура ТоварыЦенаПриИзменении(Элемент)
 СтрокаРасчета = Элементы.Товары.ТекущиеДанные;
 СтрокаРасчета.Сумма =
 СтрокаРасчета.Количество*СтрокаРасчета.Цена;
 КонецПроцедуры

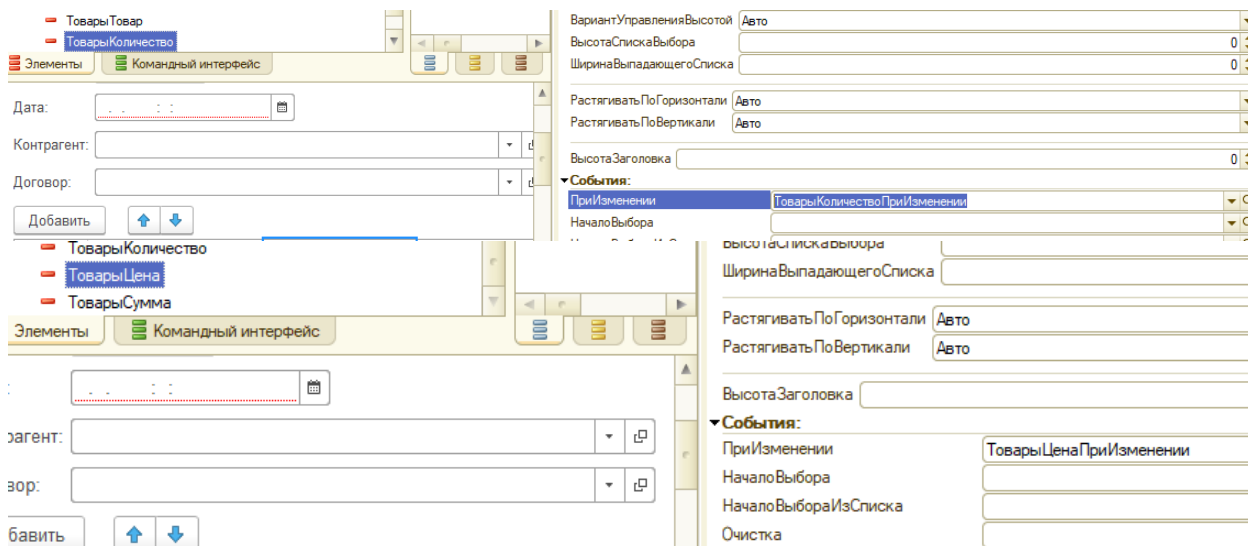


Рис. 7.23

Для того чтобы пользователь мог видеть не только сумму по каждой строке, но и общую сумму документа, в свойствах таблицы Товары укажем, что необходимо выводить *Подвал* (Рис. 7.24). После чего в свойствах поля Сумма укажем Путь к данным подвала (Рис. 7.25).

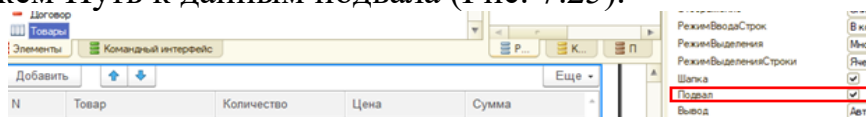


Рис. 7.24

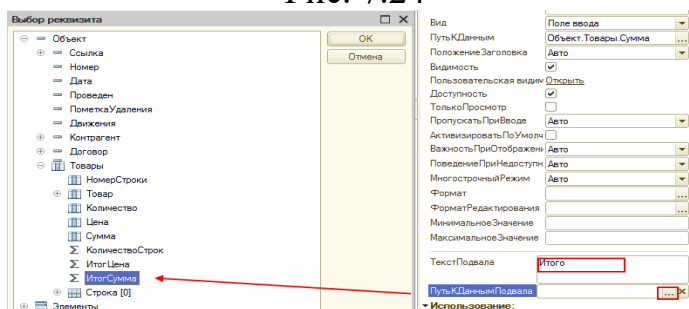


Рис. 7.25

В результате выполненных действий пользователь получит форму документа ПриходнаяНакладная, представленную на рисунке 7.26. В данной форме необходимо ввести количество и цену по каждой позиции, при этом сумма по строке должна рассчитываться автоматически путем умножения *количества на цену*. В нижней части формы автоматически подсчитывается

итоговая сумма по всем строкам табличной части.

Договоры Контрагенты Подразделения Поступление денежных средств Приходная накладная Сотрудники СписаниеДенежныхСредств Товары и услуги

← → ☆ Приходная накладная (создание) *

Провести и закрыть Записать Провести Еще ▾

Номер:

Дата: 08.09.2026 0:00:00 📅

Контрагент: Частное лицо ▾ ⚙

Договор: 0/86y ▾ ⚙

Добавить ⬆ ⬇ Поиск (Ctrl+F) × Еще ▾

N	Товар	Количество	Цена	Сумма
1	Посудомоечная машина	2	25 000,00	50 000,00
2	Холодильник бытовой	5	50 000,00	250 000,00
				Итого 300 000,00

Рис 7.26

Практическая работа №9

Создайте документ **Расходная накладная** (Рис 7.27, 7,28)

Расходная накладная 000000001 от 16.07.2... (1С:Предприятие) 🌟 📅 31 M M+ M- - □ ×

Расходная накладная 000000001 от 16.07.2014 0:00:00

Провести и закрыть 📄 💰 Провести | Печать | Остатки по складу Все действия ▾ ?

Номер: 000000001 Дата: 16.07.2014 0:00:00 📅

Контрагент: Спиридонова Галина ... 🔍

Договор: Основной ... 🔍

Сотрудник: Симонов Валерий Михайлович ... 🔍

Товары Услуги

+ Добавить 📄 ✖ ⬆ ⬇ Все действия ▾

N	Номенклатура	Набор свойств	Количество	Цена	Сумма
1	Шланг резиновый	Польша	1,000	150,00	150,00
2	Кабель электрический	Черные кабели	10,000	30,00	300,00
				Всего:	450,00

Комментарий: с 10-00 |

Рис 7.27

Расходная накладная 000000001 от 16.07.2014 0:00:00

Провести и закрыть | Провести | Печать | Остатки по складу | Все действия ▾

Номер: 000000001 | Дата: 16.07.2014 0:00:00

Контрагент: Спиридонова Галина

Договор: Основной

Сотрудник: Симонов Валерий Михайлович

Товары | Услуги

+ Добавить | Все действия ▾

N	Услуга	Сумма	Сумма НДС	Всего
1	Подключение воды	100,00	18,00	118,00
2	Подключение электричества	100,00	18,00	118,00
				236,00

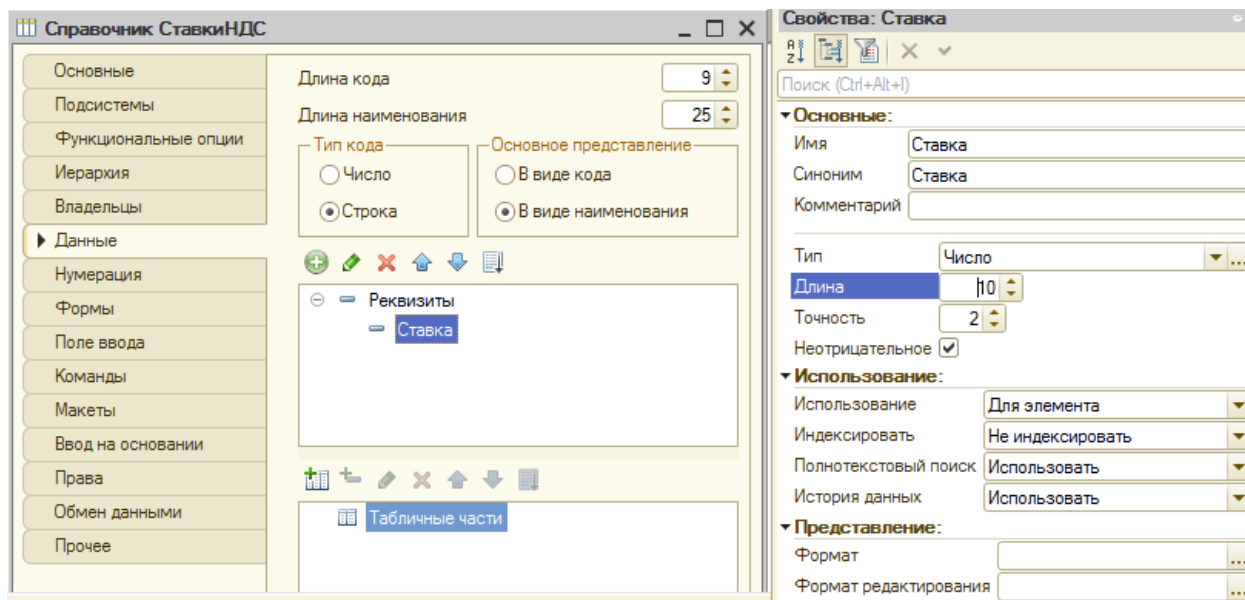
Комментарий: с 10-00

Рис 7.28

Документ РасходнаяНакладная имеет более сложную структуру, чем рассмотренный ранее документ ПриходнаяНакладная,(Рис 7.29) поэтому задействуем еще ряд механизмов для облегчения работы пользователя. Во-первых, 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. 7.30).

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*

В-третьих, 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. 7.31).



Реквизит НДС в справочнике **Номенклатура** должен ссылаться на справочник **СтавкиНДС** (тип *СправочникСсылка.СтавкиНДС*)

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.

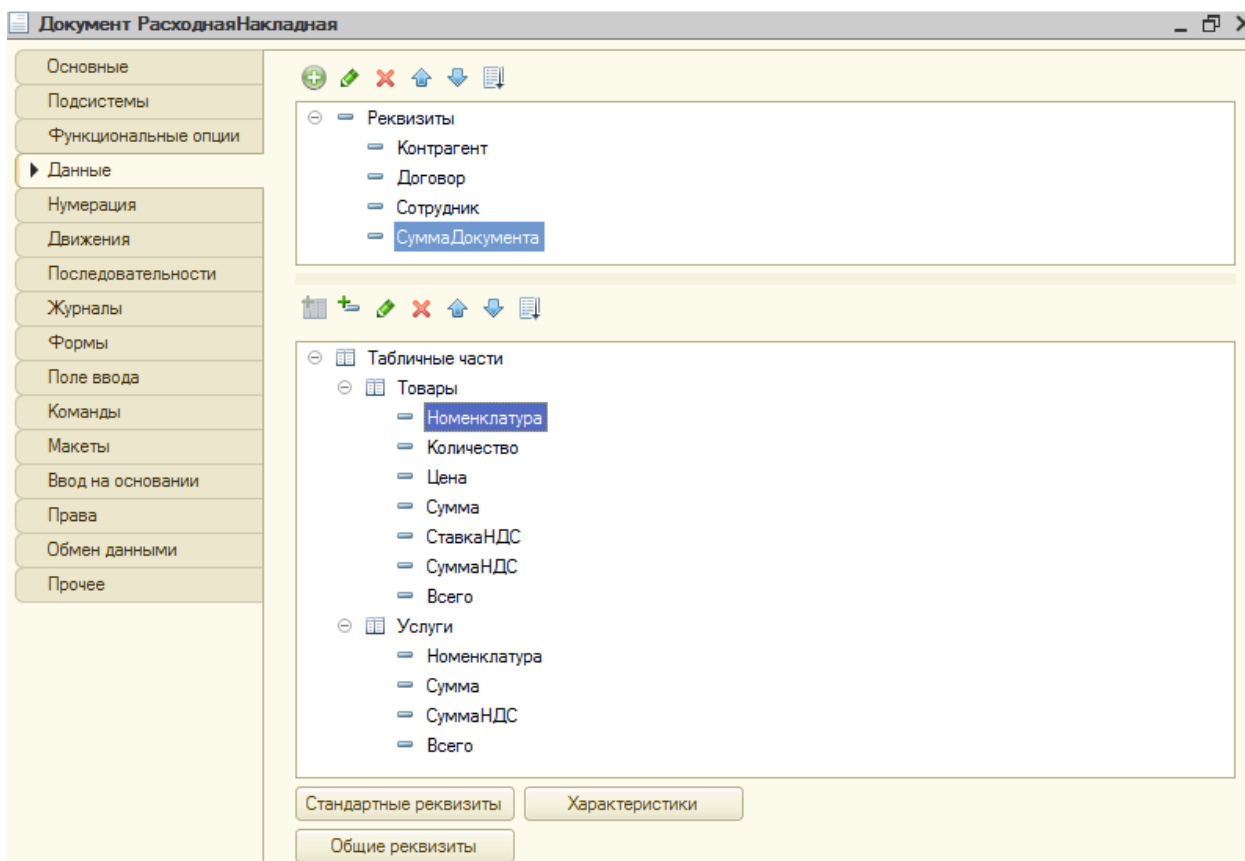
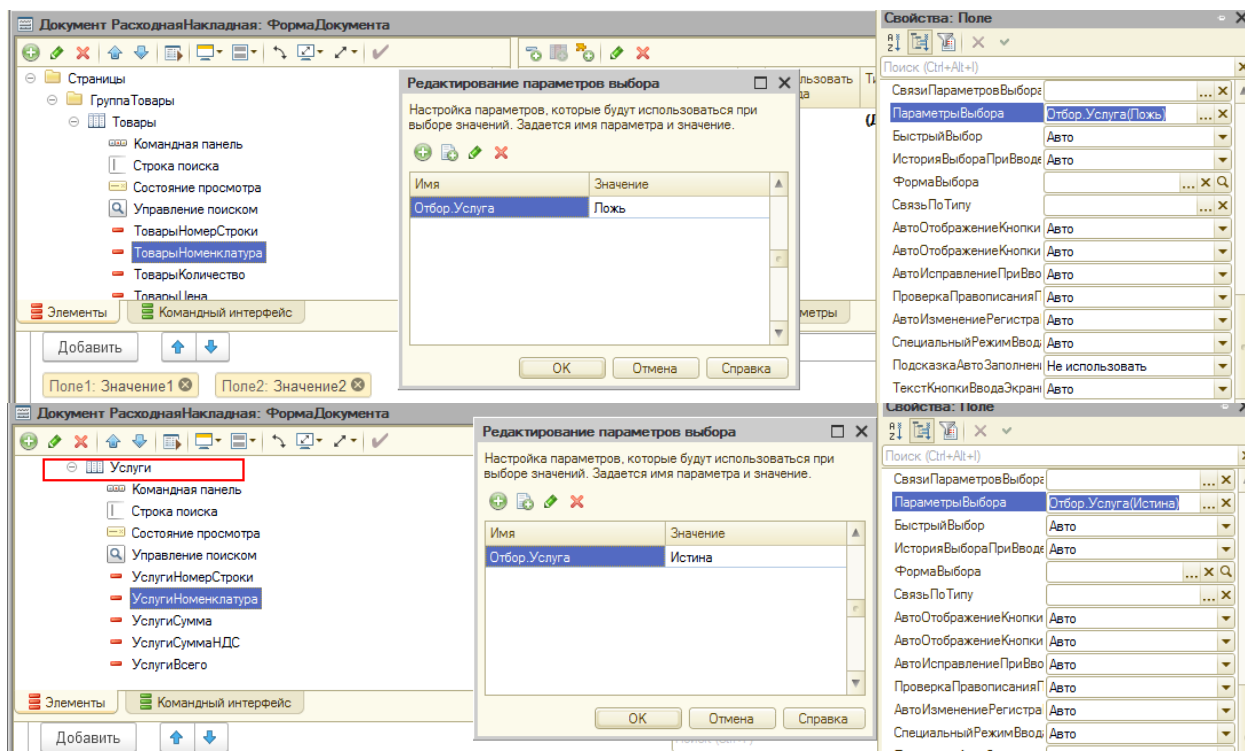


Рис 7.29



В модуль объекта расходной накладной вставляем код (Рис 7.32):

Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)
 СуммаДокумента = Товары.Итог("Всего") + Услуги.Итог("Всего");
 КонецПроцедуры

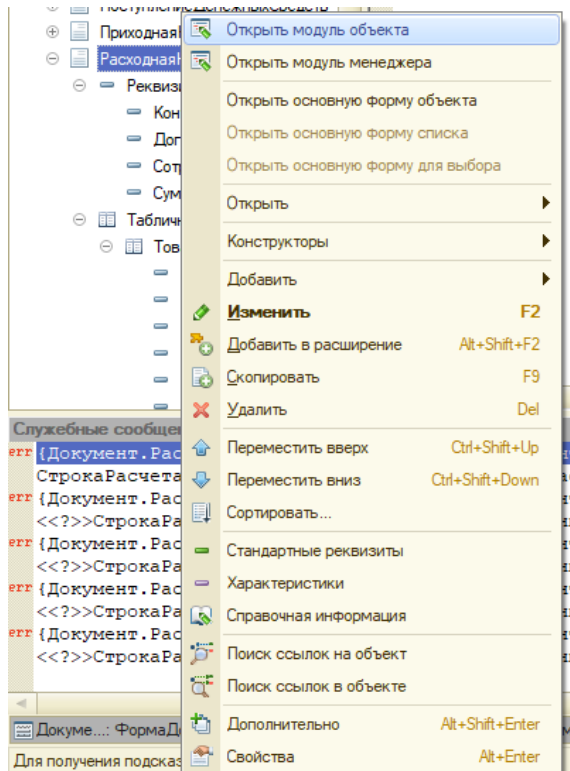


Рис 7.32

А в модуль формы документа вставляем :

&НаКлиенте

Процедура ТоварыСтавкаНДСПриИзменении(Элемент)

ТекСтрок = Элементы.Товары.ТекущиеДанные;

СтавкаЧисло = ПолучитьСтавку(ТекСтрок.СтавкаНДС);

ТекСтрок.СуммаНДС = ТекСтрок.Сумма * СтавкаЧисло;

ТоварыСуммаНДСПриИзменении(Элемент);

КонецПроцедуры

&НаСервереБезКонтекста

Функция ПолучитьСтавку(СтавкаНДС)

Если Не ЗначениеЗаполнено(СтавкаНДС) Тогда

Возврат 0;

КонецЕсли;

Возврат СтавкаНДС.Ставка / 100;

КонецФункции

&НаКлиенте

Процедура ТоварыСуммаНДСПриИзменении(Элемент)

ТекСтрок = Элементы.Товары.ТекущиеДанные;

ТекСтрок.Всего = ТекСтрок.Сумма + ТекСтрок.СуммаНДС;

КонецПроцедуры

&НаКлиенте

```
Процедура ТоварыЦенаПриИзменении(Элемент)  
ТекСтрок = Элементы.Товары.ТекущиеДанные;  
ТекСтрок.Сумма = ТекСтрок.Цена * ТекСтрок.Количество;  
ТоварыСтавкаНДСПриИзменении(Элемент); // пересчитать НДС от  
новой суммы
```

КонецПроцедуры

&НаКлиенте

```
Процедура ТоварыКоличествоПриИзменении(Элемент)
```

```
ТоварыЦенаПриИзменении(Элемент);
```

КонецПроцедуры

&НаКлиенте

```
Процедура ТоварыСуммаПриИзменении(Элемент)
```

```
ТоварыСтавкаНДСПриИзменении(Элемент);
```

КонецПроцедуры

&НаКлиенте

```
Процедура УслугиСуммаНДСПриИзменении(Элемент)
```

```
ТекСтрок = Элементы.Услуги.ТекущиеДанные;
```

```
ТекСтрок.Всего = ТекСтрок.Сумма + ТекСтрок.СуммаНДС;
```

КонецПроцедуры

Не забудьте для нужных полей указать функции и у «товары» и «услуги» включить подвал и у полей «всего» выбрать путь к подвалу

7.4. Печать документа. Конструктор печати

Чтобы пользователь мог формировать печатную форму документа, воспользуемся конструктором печати. Вызов конструктора осуществляется на закладке *Макеты* по кнопке *Конструкторы* (Рис. 7.33)

Далее, следуя инструкциям конструктора (Рис. 7.34 – Рис. 8.39), необходимо заполнить все необходимые поля.

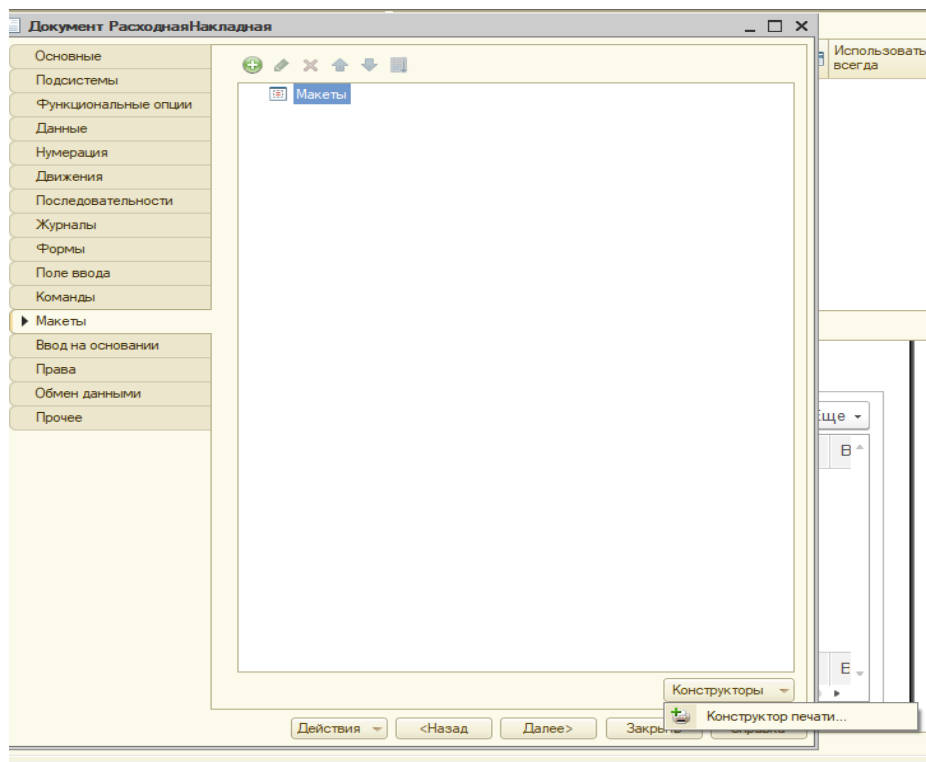


Рис. 7.33

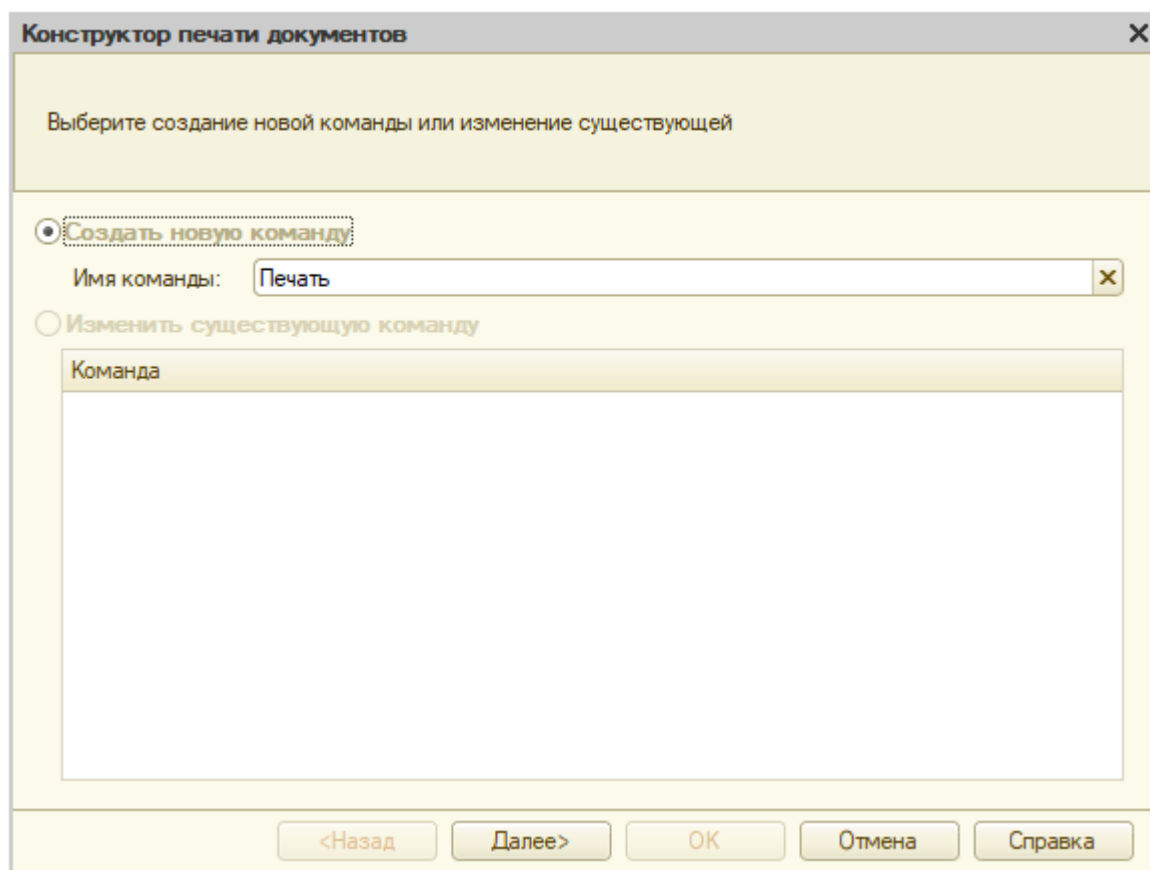


Рис. 7.34

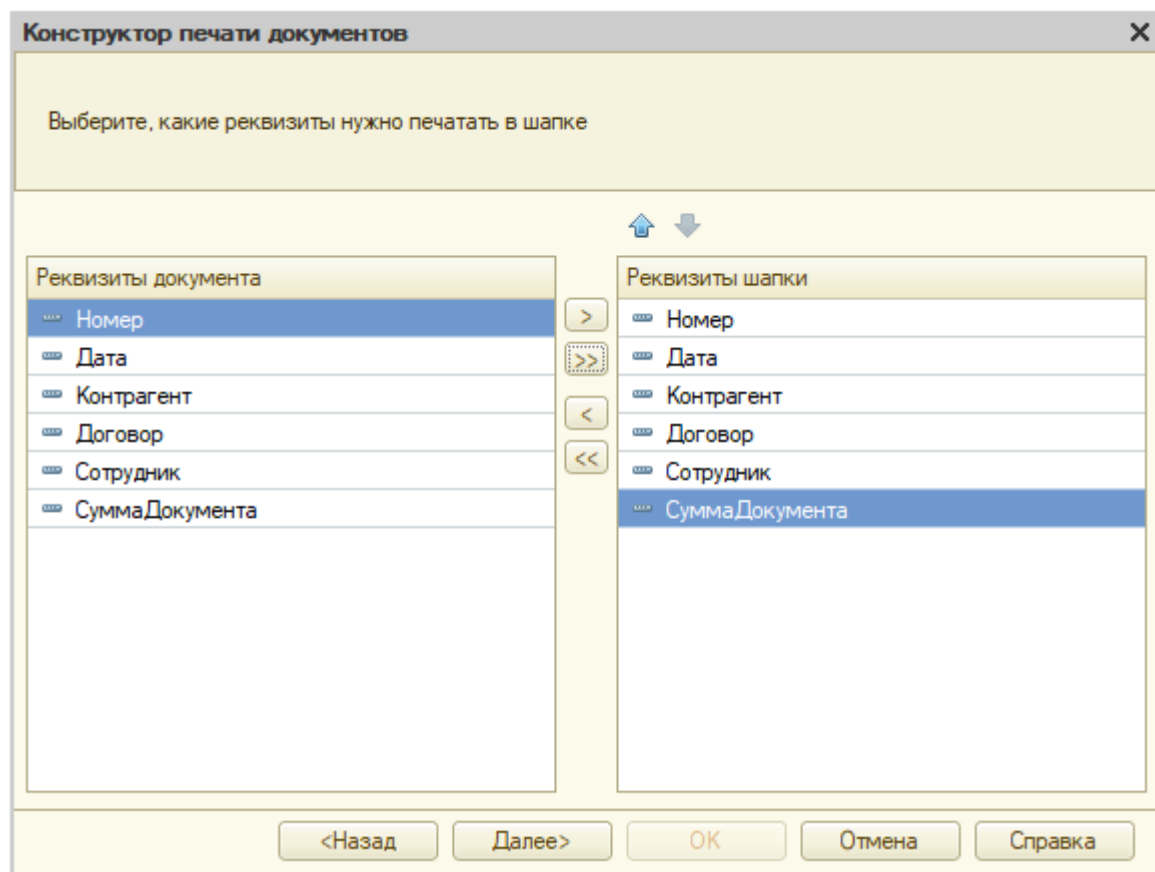


Рис 7.35

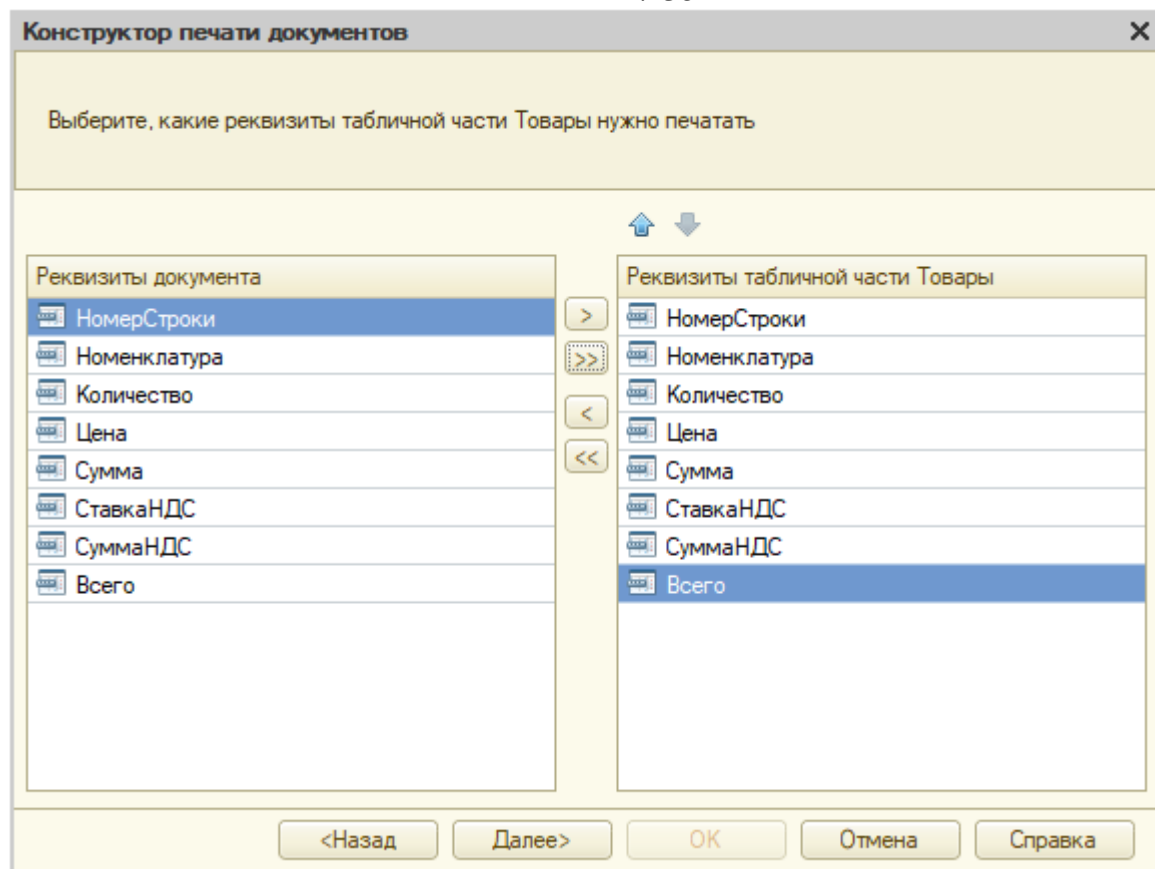


Рис. 7.36

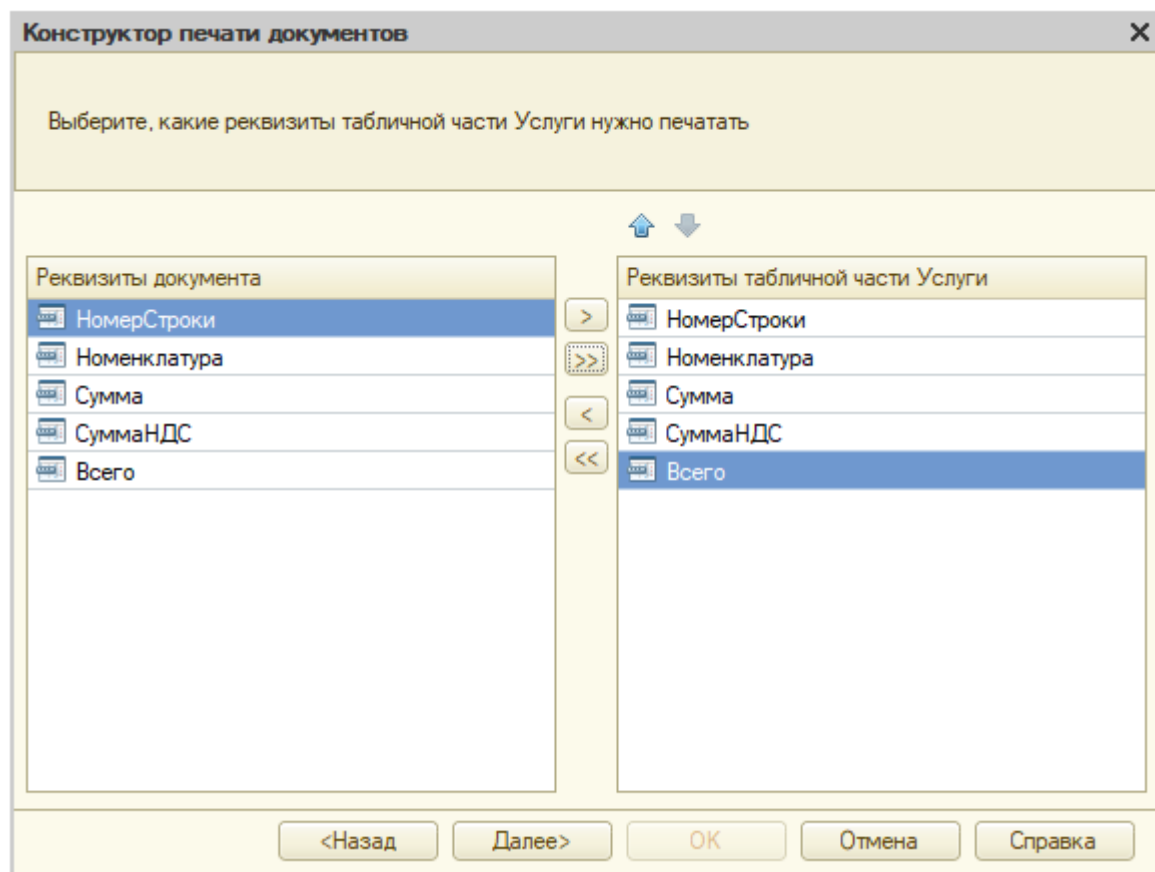


Рис 7.37

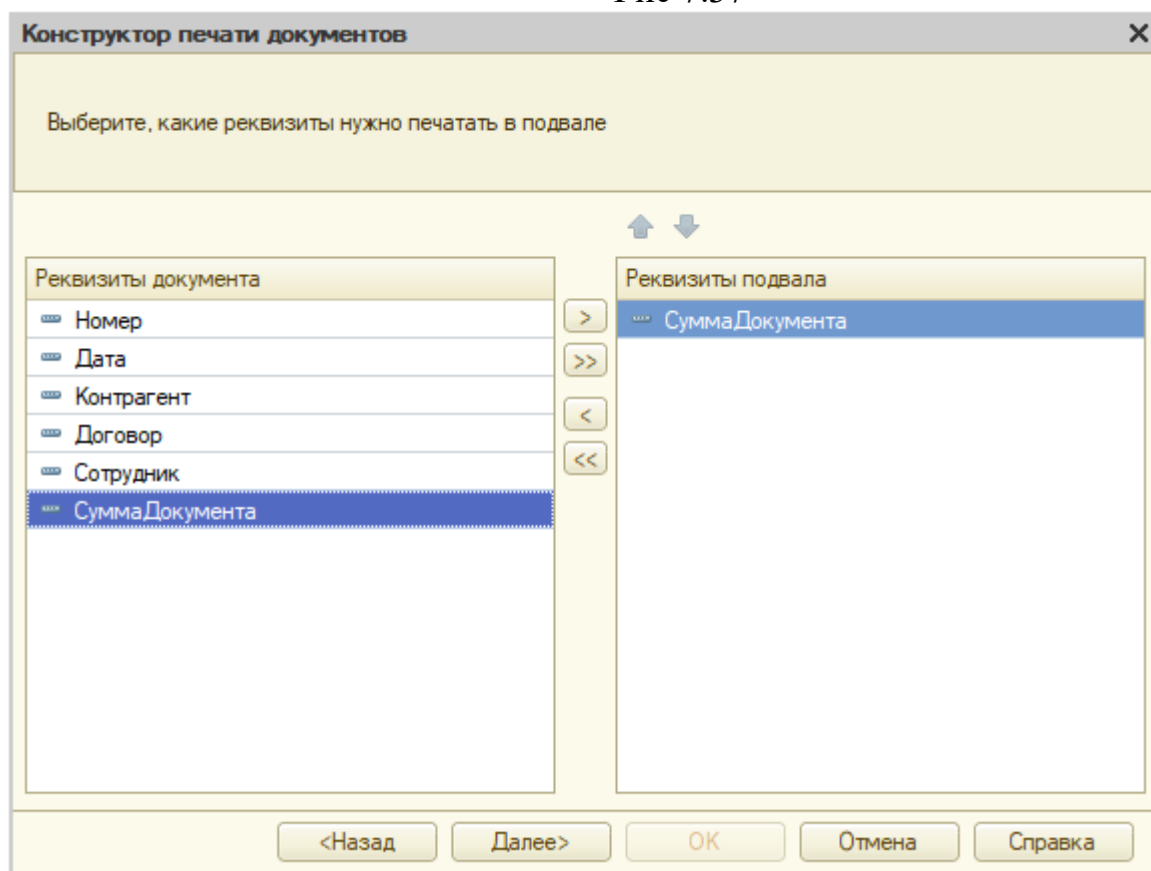


Рис. 7.38

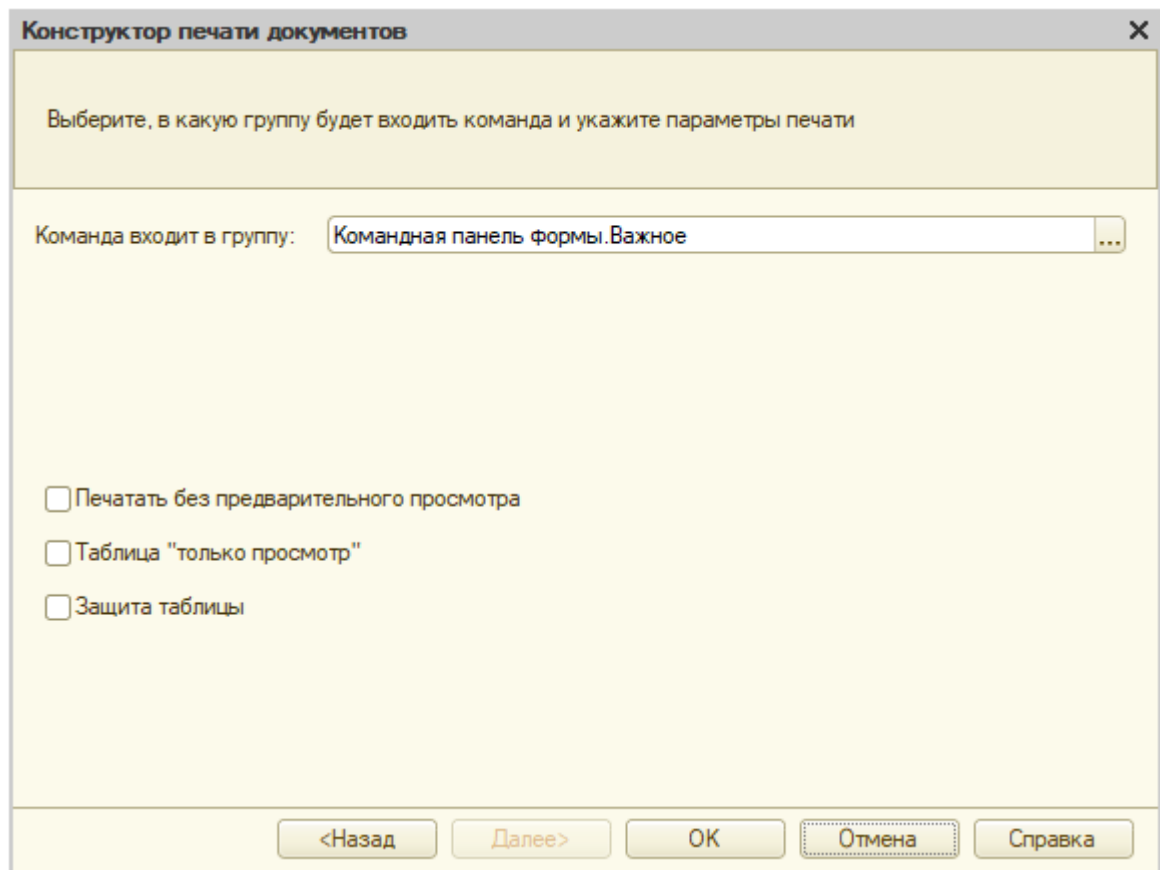


Рис. 7.39

В результате работы конструктора будет создан новый макет (Рис. 7.40), новая команда и *сформируется следующий* программный код в модуле команды и модуле менеджера:

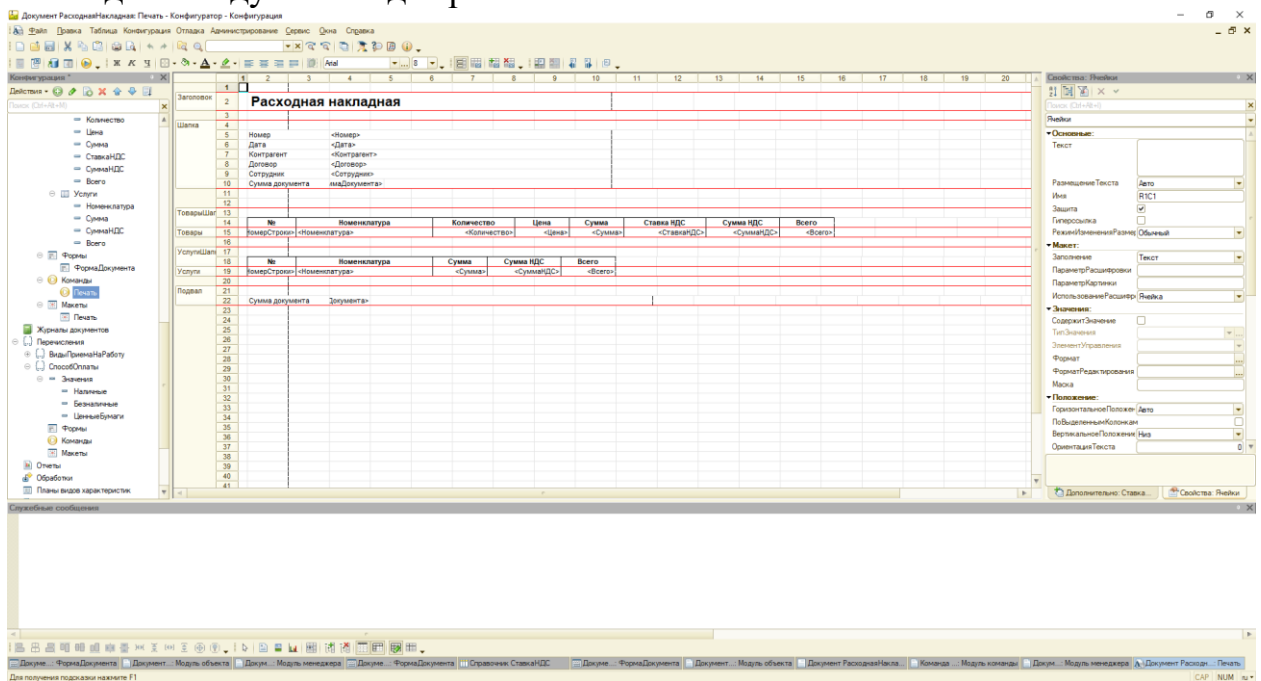


Рис 7.40

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();

ВставляяРазделительСтраниц = Ложь;

Пока Выборка.Следующий() Цикл

```

Если ВставитьРазделительСтраниц Тогда
    ТабДок.ВывестиГоризонтальныйРазделительСтраниц();
КонецЕсли;

ТабДок.Вывести(ОбластьЗаголовков);

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);

ВставитьРазделительСтраниц = Истина;
КонецЦикла;
//}}
КонецПроцедуры

```

Кроме того, в форме списка и в форме документа на командной панели появилась кнопка Печать (Рис. 7.41), нажатие на которую позволяет получить печатную форму документа

← →

☆ **Расходная накладная 000000001 от 09.09.2026 23:04:57**

Провести и закрыть
Записать
Провести
Печать

Номер:

Дата: 09.09.2026 23:04:57 📅

Контрагент: ▼ 📄

Договор: ▼ 📄

Сотрудник: ▼ 📄

Сумма документа: 📅

Товары
Услуги

Рис 7.41

8. ПОДСИСТЕМЫ И ИНТЕРФЕЙС

Подсистема позволяет группировать различные объекты системы по какой-либо подсистеме. какому-либо функциональному признаку.

Создадим подсистемы и укажем их состав (рис. 8.1). Следует учитывать, что все объекты должны входить в состав

Чтобы добавить в подсистему *Закупки* подчиненную подсистему, используйте команду контекстного меню *Добавить | Подчиненная подсистема*

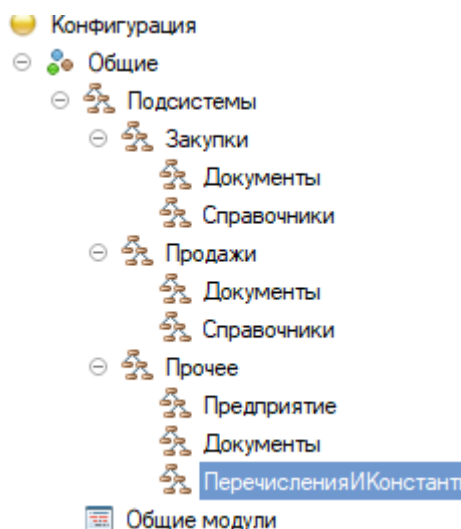


Рис 8.1

В подсистему «Закупки» добавляем подчинённые подсистемы «Документы» и «Справочники». В подсистему «Документы» добавляем документы «Расход денег» и «Расходная накладная». В подсистему «Справочники» добавляем справочники на свой выбор.

В подсистему «Продажи» добавляем подчинённые подсистемы «Документы» и «Справочники». В подсистему «Документы» добавляем документы «Приходная накладная» и «Поступление денежных средств». В

подсистему «Справочники» добавляем справочники на свой выбор.

Добавляем подсистему «Прочее» с подчинёнными подсистемами «Предприятие», «Документы» и «ПеречисленияИКонстанты». В результате этих действий для пользователя в режиме «1С:Предприятие» автоматически формируется командный интерфейс с разделами, соответствующими созданной структуре подсистем. сформируется интерфейс, разбитый согласно подсистемам (рис. 8.2).

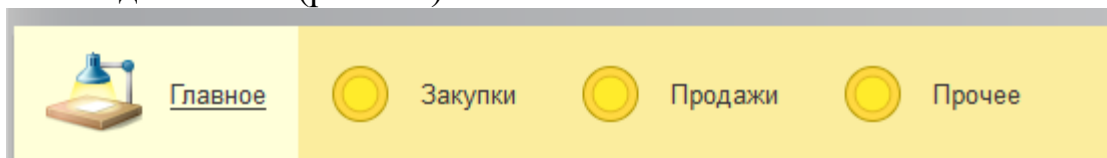


Рис. 8.2

Помимо появившихся разделов для каждой подсистемы на панели разделов появился еще и дополнительный раздел *Рабочий стол*.

9. Функциональные опции

Функциональные опции — это механизм, который позволяет оперативно включить или отключить в пользовательском режиме выделенный функционал.

Создадим следующие объекты:

1. Константу *ВалютныйУчет* с типом *булево*
2. Справочник *Валюты*

Добавим справочник «Валюты» для хранения списка валют и соответствующие реквизиты в документы «РасходнаяНакладная» и «ПоступлениеДенежныхСредств». При создании реквизита «Валюта» обязательно свяжите его с созданным справочником «Валюты» — для этого в свойстве «Тип» реквизита укажите ссылку на справочник «Валюты». Эти реквизиты обязательно надо вывести на форму документов, иначе пользователь не сможет указать валюту при заполнении документа (Рис. 9.1).

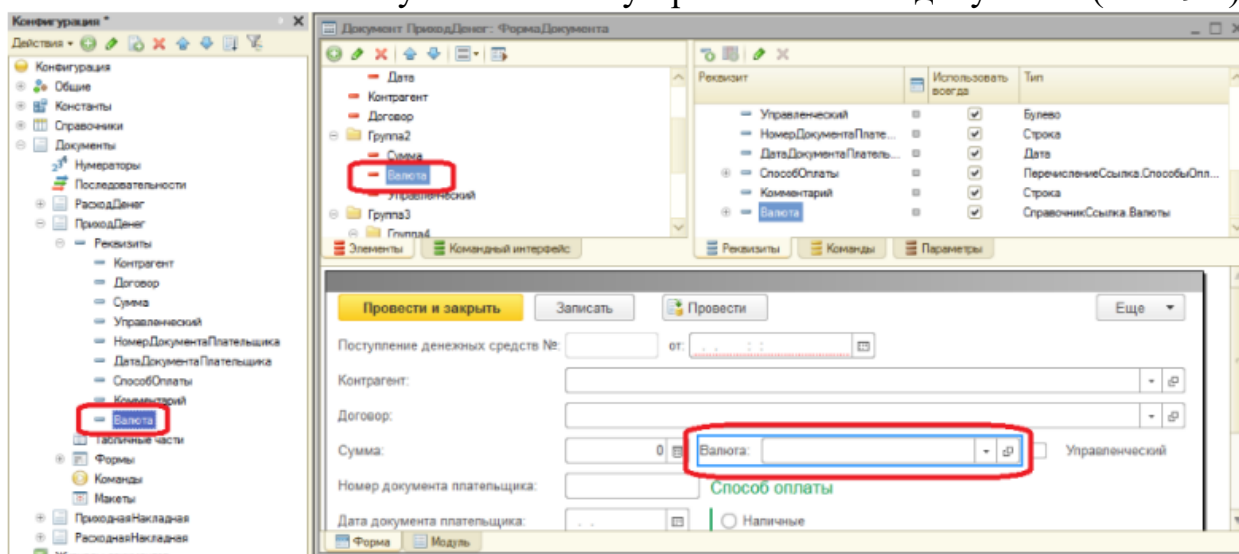


Рис. 9.1

После этого создаем саму функциональную опцию ВалютныйУчет (Рис. 9.2), в которой через свойство "Хранение" устанавливаем связь с нашей константой, определяющей необходимость наличия валюты.

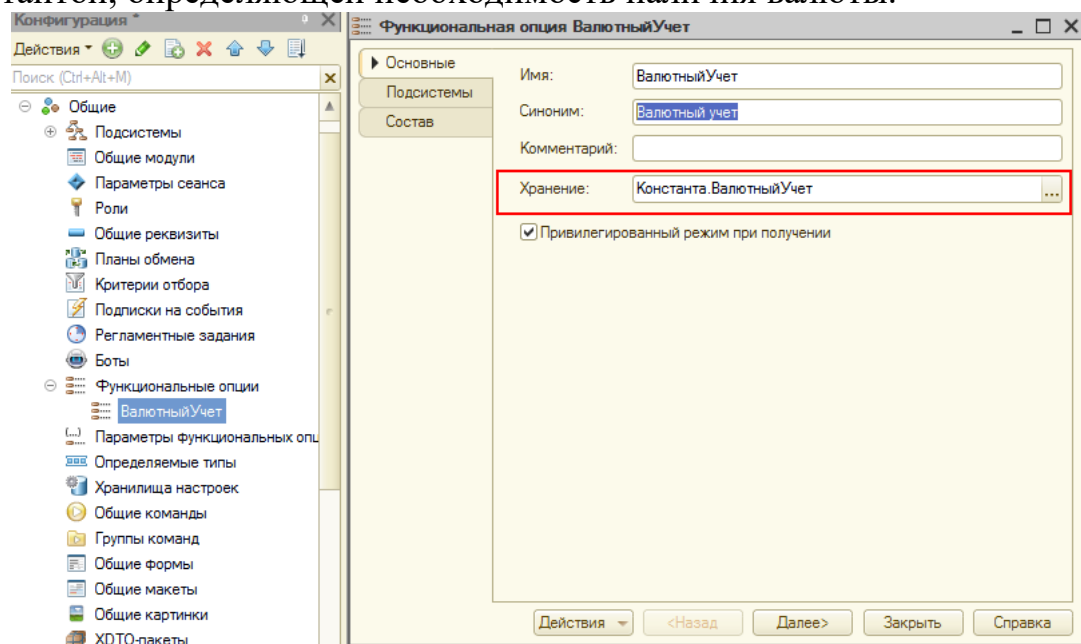


Рис 9.2

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

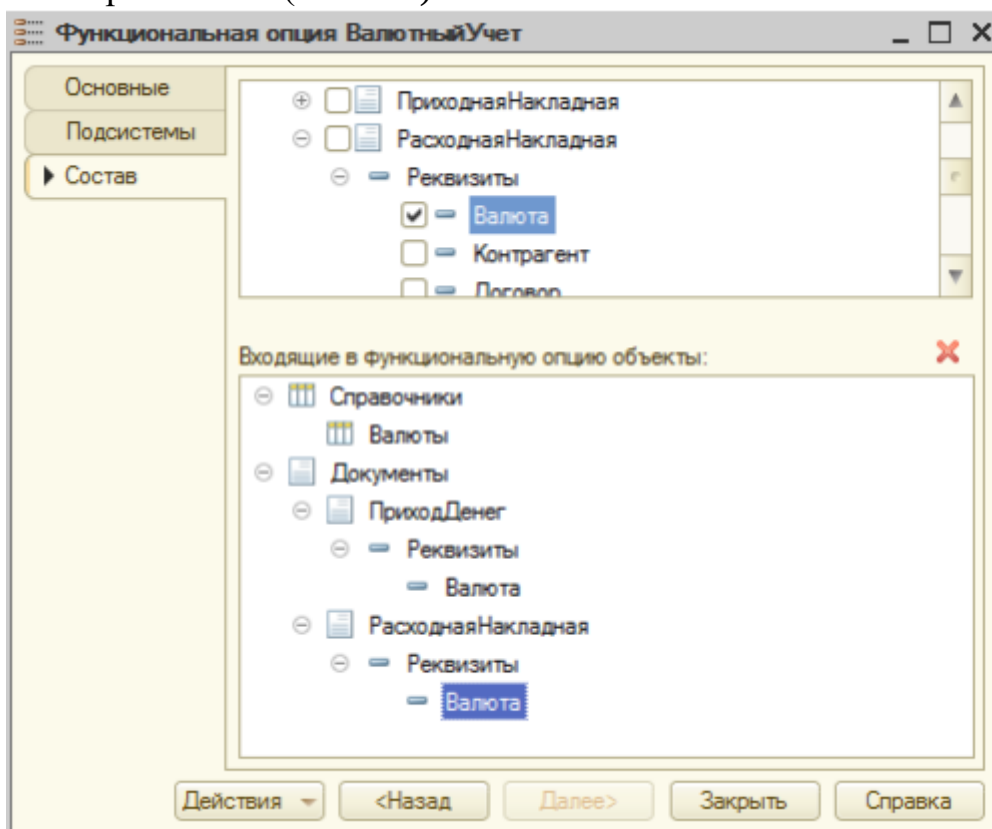


Рис. 9.3

После этого перейдем в режим "1С:Предприятие" и сделаем следующие операции: По умолчанию значение константы ВалютныйУчет

будет Ложь (флажок снят). Откроем документ ПриходДенег и РасходнаяНакладная. В этом случае, соответствующие реквизиты будут скрыты (Рис. 9.4). Кроме того, справочник Валюта будет недоступен для работы пользователя.

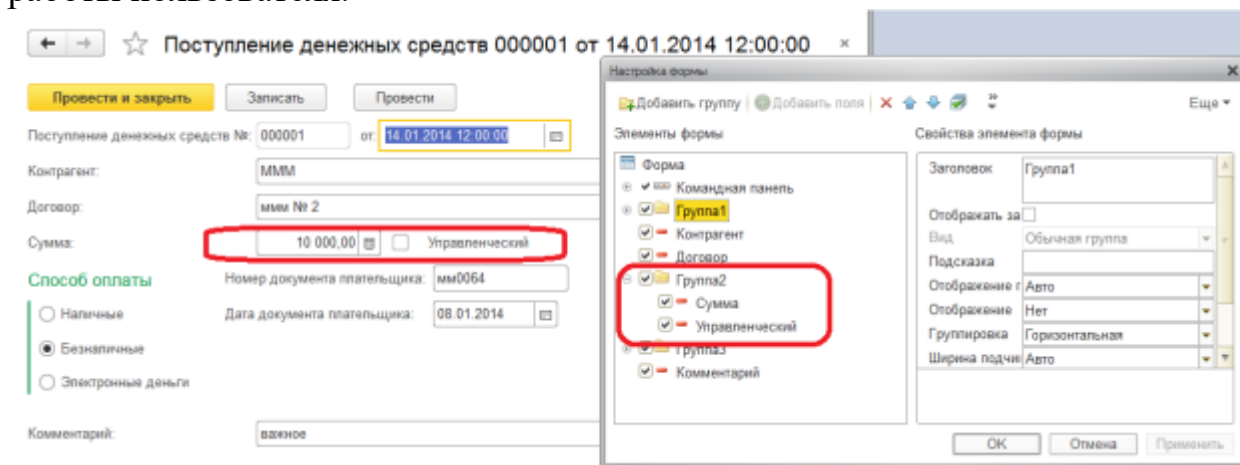
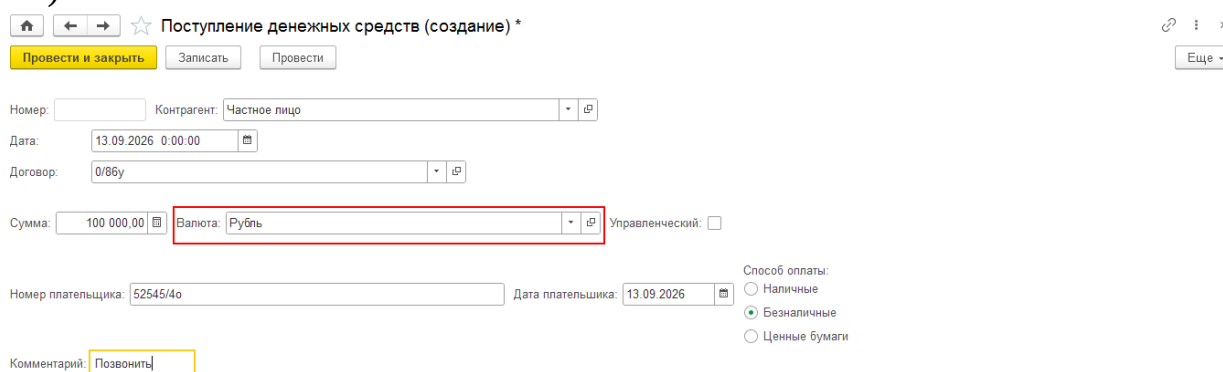


Рис. 9.4

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



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 усложниться, 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.

Рис 10.1

Создание нового регистра аналогично созданию нового документа или справочника (Рис. 10.1). Структуру регистра можно задать на закладке Данные.(Рис. 11.3) То, что мы собственно храним в регистре, находится в ресурсе, а те разрезы, в которых нас это информация интересует, находятся в измерениях. В нашем случае необходимо хранить цену для каждого товара, купленного по каждому договору и каждого поставщика, т.е. цена в разрезе поставщиков, договоров и номенклатуры. Следовательно, цена должна быть ресурсом, а контрагент, договор и номенклатура – измерениями. В этом случае система будет контролировать, что для каждого уникального набора значений измерений может быть только одно значение ресурса, т.е. для одного и того же товара по одному договору для одного поставщика может быть только одна цена. Естественно, что необходимо и для измерений и для ресурсов указать соответствующий тип данных.

Регистр сведений «Цены поставщиков»

Создайте регистр сведений «ЦеныПоставщиков» с измерениями «Контрагент», «Договоры» и «Номенклатура». В качестве типа измерений укажите ссылки на соответствующие справочники: «Контрагенты», «Договоры» и «Номенклатура». Добавьте ресурс «Цена» с типом «Число» для хранения значения цены поставщика по данной номенклатуре в разрезе контрагента и договора.

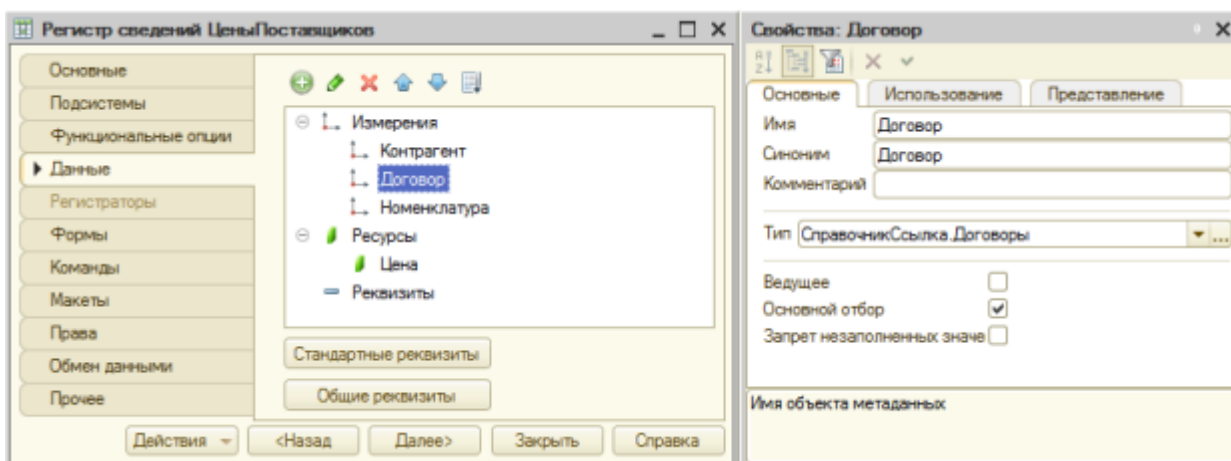


Рис. 10.2

Информацию в регистр пользователь может внести самостоятельно в режиме «1С:Предприятие», используя соответствующую форму списка (Рис. 10.3). Самостоятельно заполните регистр сведений «Цены поставщиков».

☆ **Цены поставщиков**

Создать [Иконка добавления] Поиск (Ctrl+F) [Поиск] [Еще]

Контрагент ↓	Договор	Номенклатура	Цена
☑ Частное лицо	0/86у	Бойлер	10 000,00
☑ Частное лицо	0/86у	Кондиционер бытовой	25 000,00
☑ Частное лицо	0/86у	Посудомоечная машина	50 000,00
☑ Частное лицо	0/86у	Стиральная машина	45 000,00
☑ Частное лицо	0/86у	Холодильник бытовой	35 000,00

Рис. 10.3

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(Номенклатура)

СтруктураОтбора = Новый Структура;
СтруктураОтбора.Вставить("Контрагент", Объект.Контрагент);
СтруктураОтбора.Вставить("Договор", Объект.Договор);
СтруктураОтбора.Вставить("Номенклатура", Номенклатура);

РезультатОтбора =
РегистрыСведений.ЦеныПоставщиков.Получить(СтруктураОтбора);

Если РезультатОтбора = Неопределено Тогда
Возврат 0;
Иначе
Возврат РезультатОтбора.Цена;
КонецЕсли;
КонецФункции

&НаКлиенте
Процедура ТоварыНоменклатураПриИзменении(Элемент)
СтрокаРасчета = Элементы.Товары.ТекущиеДанные;
СтрокаРасчета.Цена = ПолучитьЦену(СтрокаРасчета.Номенклатура);
ТоварыЦенаПриИзменении(Элемент);
КонецПроцедуры

Обратите внимание, как у Вас в приходной накладной называется реквизит — «Номенклатура» или «Товар». Если реквизит называется «Товар», то измените код или переименуйте реквизит «Товар» в «Номенклатура». Это важно, потому что в коде функции ПолучитьЦену и в процедуре ТоварыНоменклатураПриИзменении используется имя "Номенклатура" — как при обращении к реквизиту табличной части (СтрокаРасчета.Номенклатура), так и в структуре отбора регистра сведений ЦеныПоставщиков.

Заполните документ «Приходная накладная» в следующем порядке: сначала заполните контрагента, затем договор, а после выберите номенклатуру — цена должна автоматически подставиться в документ. Такой порядок важен, потому что функция ПолучитьЦену использует значения реквизитов «Контрагент» и «Договор» из шапки документа для отбора записи в регистре сведений «ЦеныПоставщиков». Если контрагент или договор не заполнены, цена не подставится.

Рис 10.4

Рис 10.5

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. В том случае, если регистр является периодическим, то наряду с существующими измерениями добавляется еще одно Период, (Рис. 10.5 и Рис. 10.6) и система предоставляет возможность быстро извлечь интересующую нас информацию на выбранный момент времени.

Период	Валюта	Курс
03.04.2007	Биткоин	25
25.12.2018	Евро	75.6
25.12.2018	Рубль	1
25.12.2018	Юань	7.73
25.12.2018	Биткоин	250

Рис 10.6

Еще одна задача, которую позволяет решить регистр сведений, это взаимосвязь хранимой информации с документами. Например, если мы хотим хранить в системе информацию о том, в каком подразделении работает сотрудник и какой у него оклад, причем, назначение сотрудника в подразделение и установка его ставки оформляется соответствующим кадровым приказом. С одной стороны, если мы сделаем документ и заполним реквизиты справочника, то вся эта информация в систему будет занесена. Однако, в том случае, когда мы удаляем соответствующий документ, информация в справочнике остается, что уже не соответствует положению вещей. Для решения подобного рода задач существует механизм проведения документов. Этот механизм позволяет установить взаимосвязь между данными регистра и документом. Происходит это следующим образом: для записанного документа может быть присвоен статус «Проведен» (определяется значением системного реквизита Проведен). В тот момент, когда документу присваивается этот статус, срабатывает обработчик проведения, в котором 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, сформированные этим документом. Статус документа отображается соответствующей пиктограммой и может быть изменен пользователем.

Для решения этой задачи, 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. 10.7).

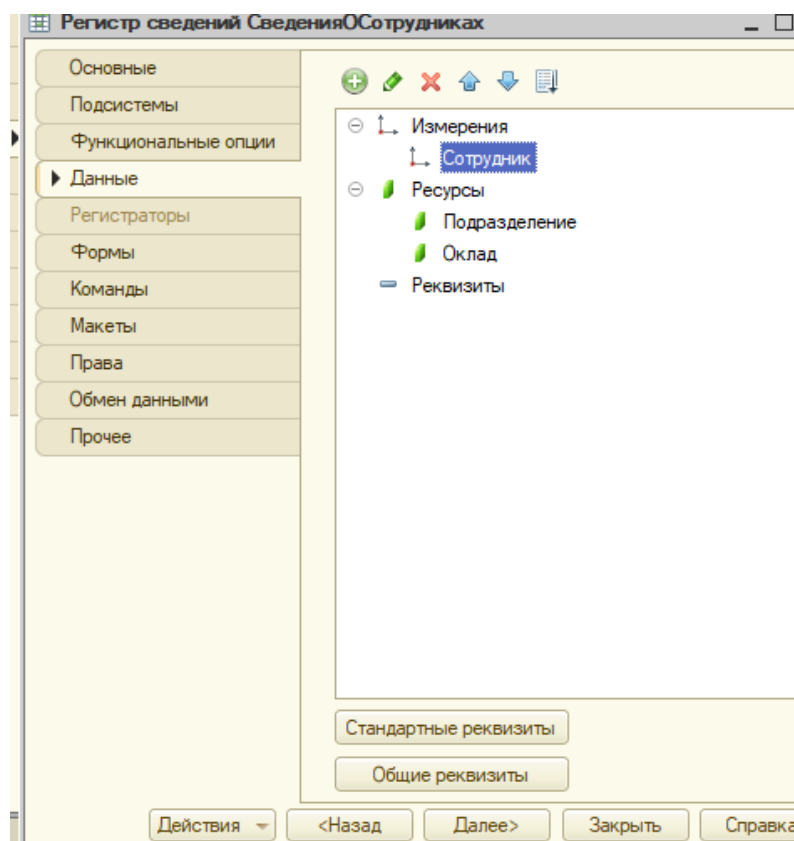


Рис 10.7

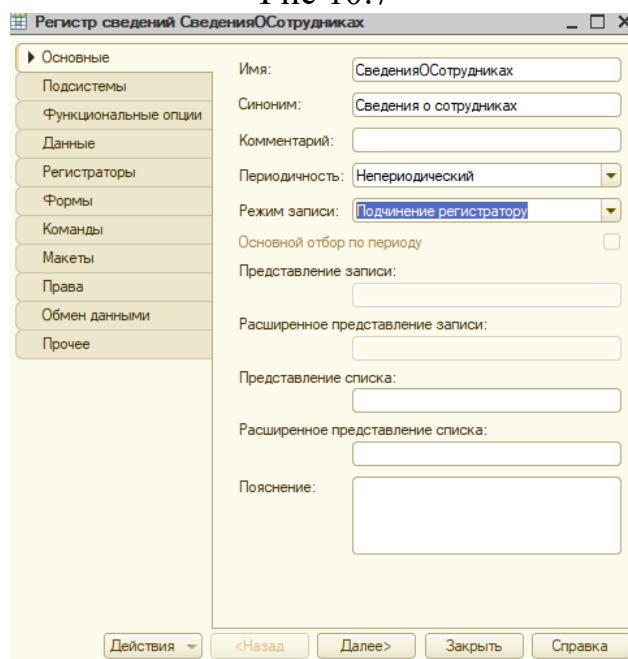


Рис. 10.8

Чтобы документ формировал записи в регистр необходимо на закладке Движения настроить следующие свойства (Рис. 10.10):

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.

После этого создадим документ Кадровый приказ. Реквизиты у документа будут Сотрудник, Подразделение и Оклад (Рис. 10.9).

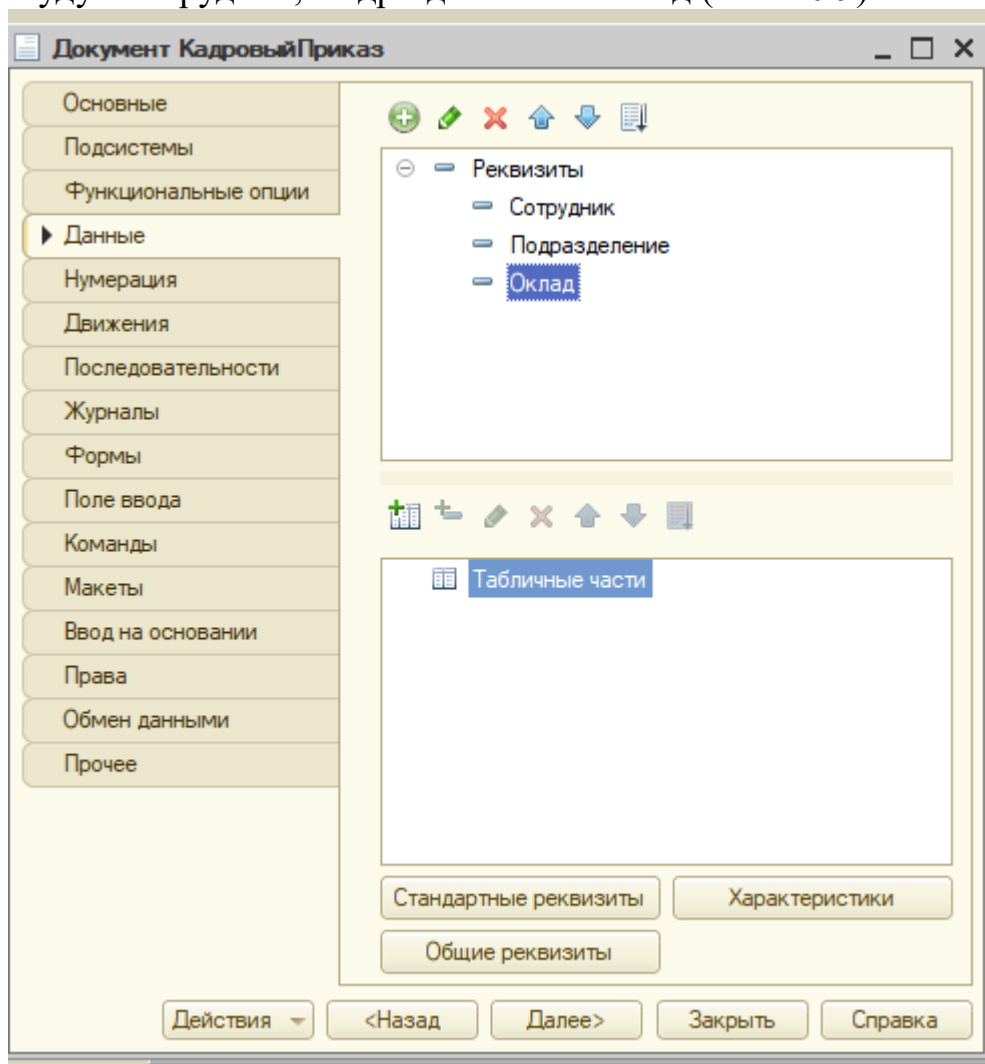


Рис 10.9

Чтобы документ формировал записи в регистр необходимо на закладке Движения настроить следующие свойства (Рис. 10.10):

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.

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. 10.11).

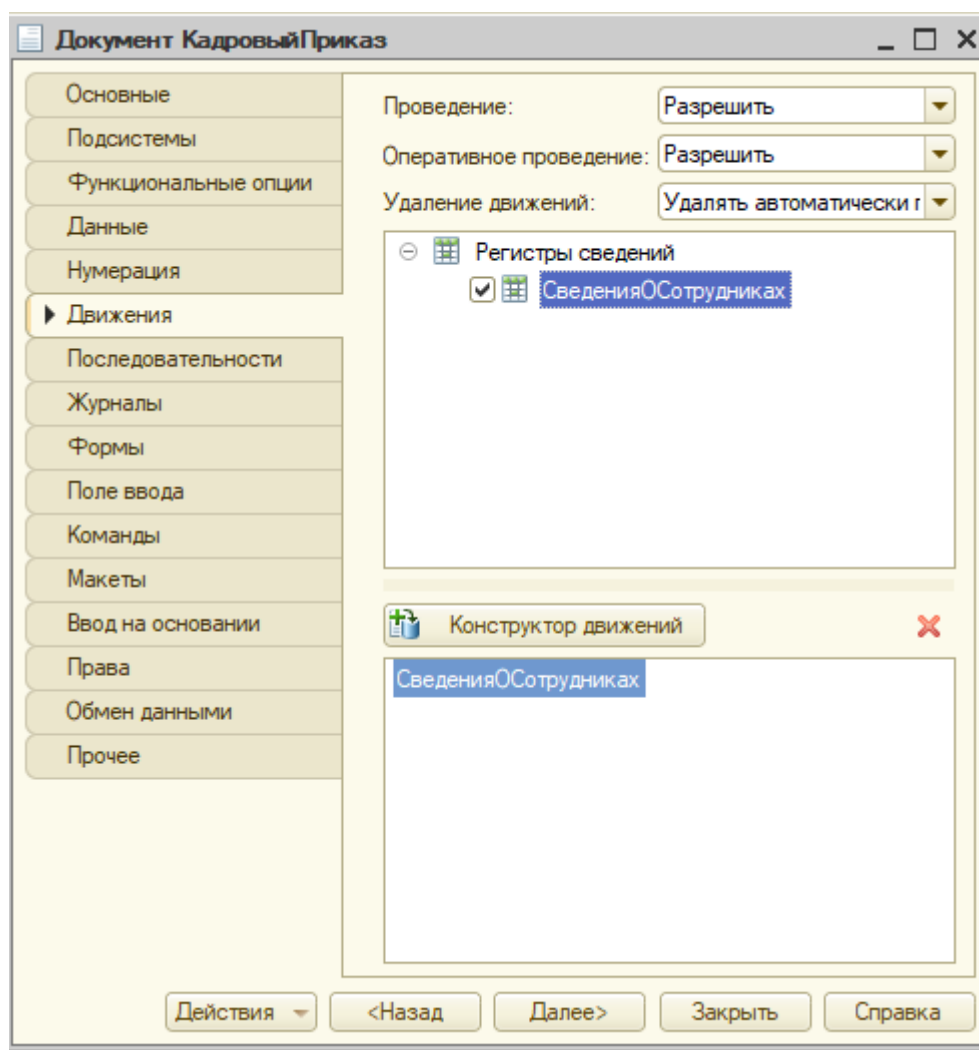


Рис 10.10

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. 10.11).

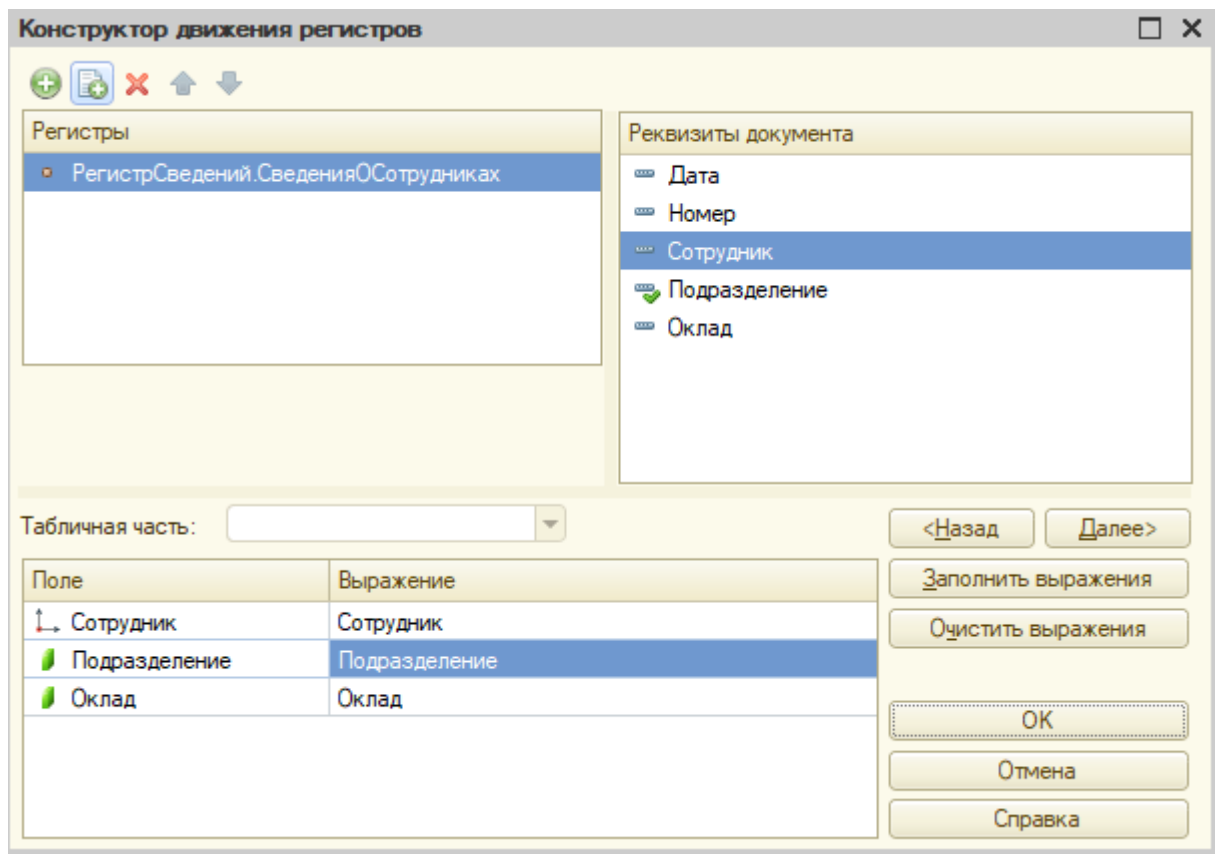


Рис 10.11

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

Процедура ОбработкаПроведения(Отказ, Режим)
 //{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
 // Данный фрагмент построен конструктором.
 // При повторном использовании конструктора, внесенные вручную
 изменения будут утеряны!!!

```
// регистр СведенияОСотрудниках
Движения.СведенияОСотрудниках.Записывать = Истина;
Движение = Движения.СведенияОСотрудниках.Добавить();
Движение.Сотрудник = Сотрудник;
Движение.Подразделение = Подразделение;
Движение.Оклад = Оклад;
```

```
//}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

В режиме «1С:Предприятие» у пользователя появляется возможность провести или снять с проведения проведенный документ (Рис. 10.12).

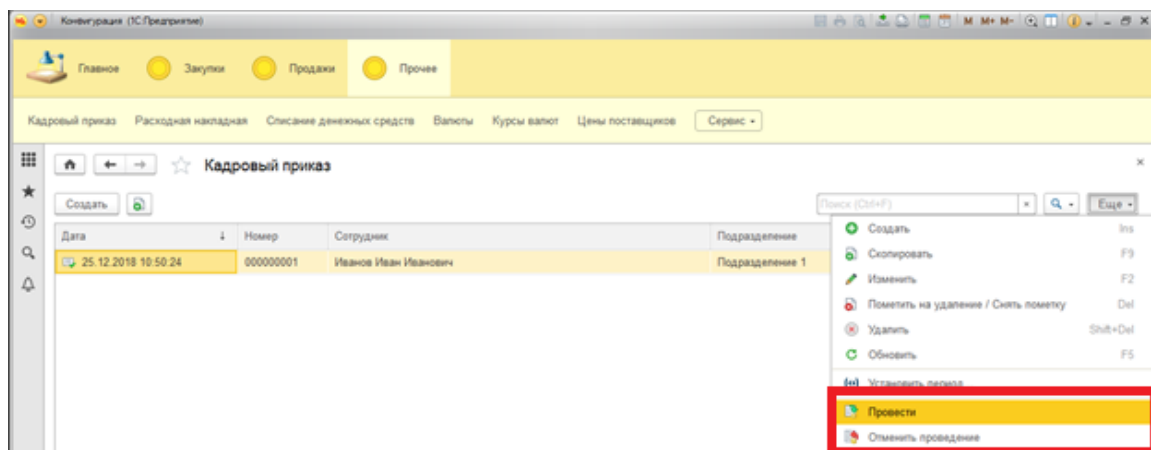


Рис 10.12

11. Отчет. компоновка данных

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

Создадим стандартным образом новый объект Отчет и воспользуемся компоновкой данных для его формирования (Рис 11.1)

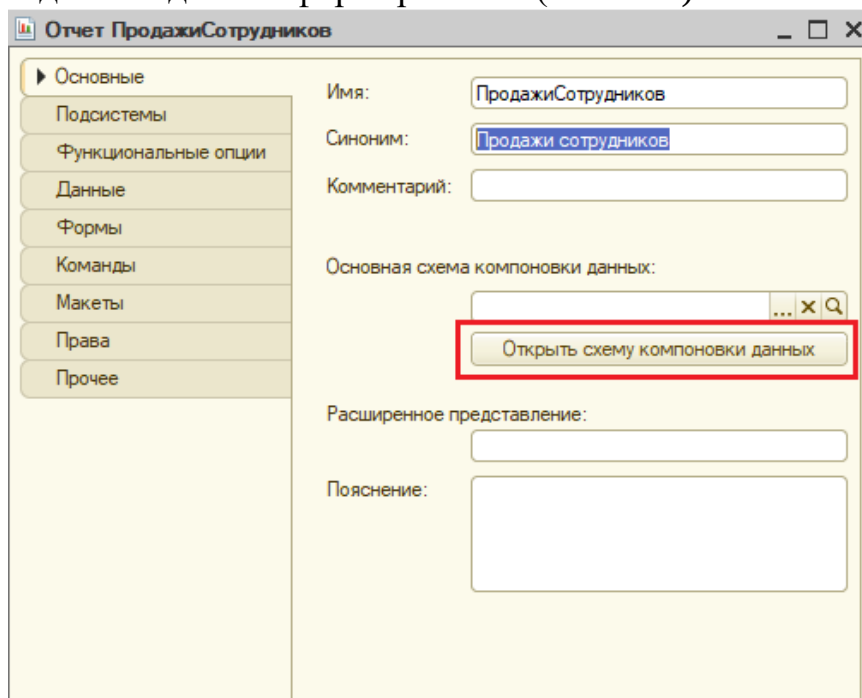


Рис 11.1

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

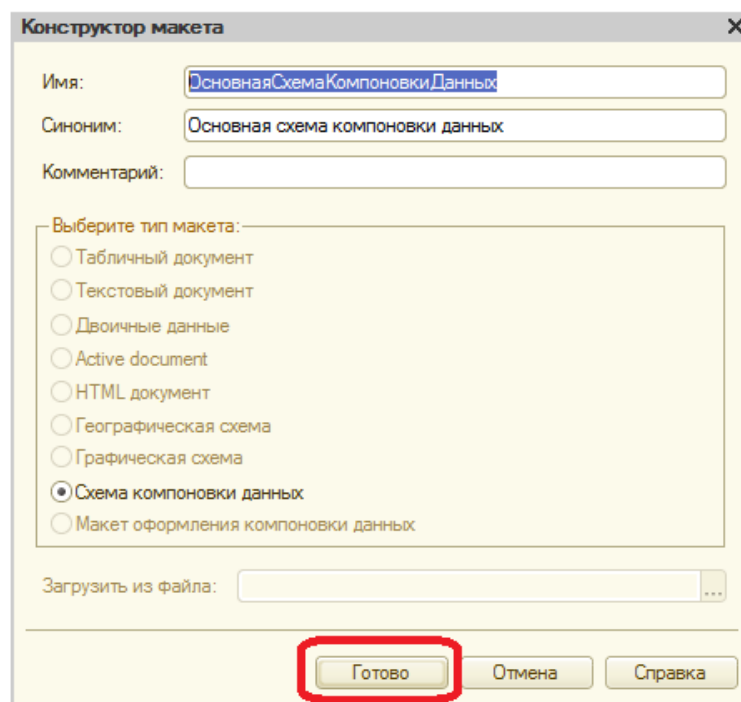


Рис 11.2

Добавим новый набор данных, которые будем получать с помощью запроса (Рис. 11.3)

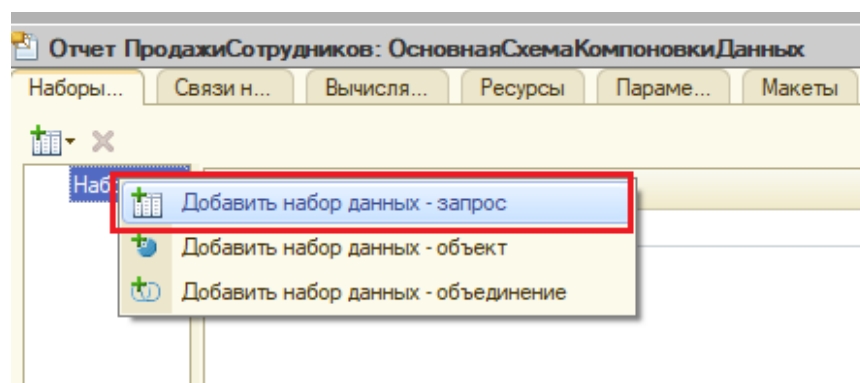


Рис 11.3

С помощью конструктора запросов сформируем запрос (Рис. 11.4 –Рис. 11.7), который соберет данные о продажах сотрудников в разрезе валют за интересующий пользователя период.

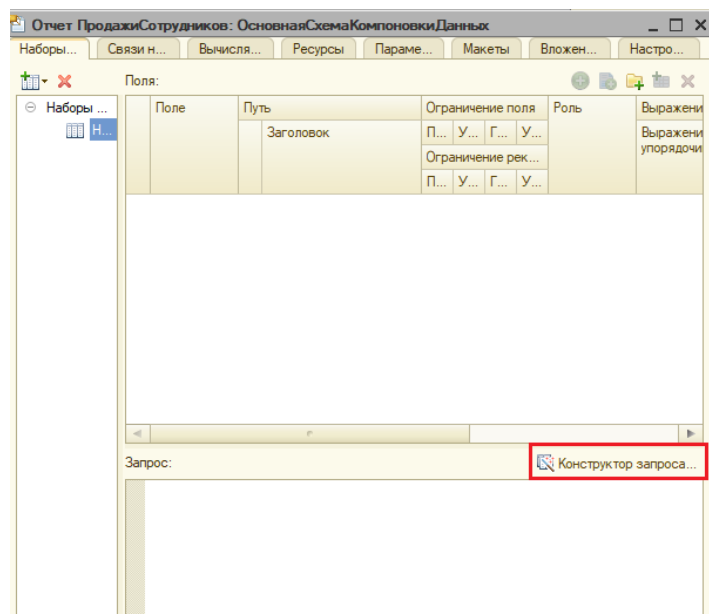


Рис 11.4

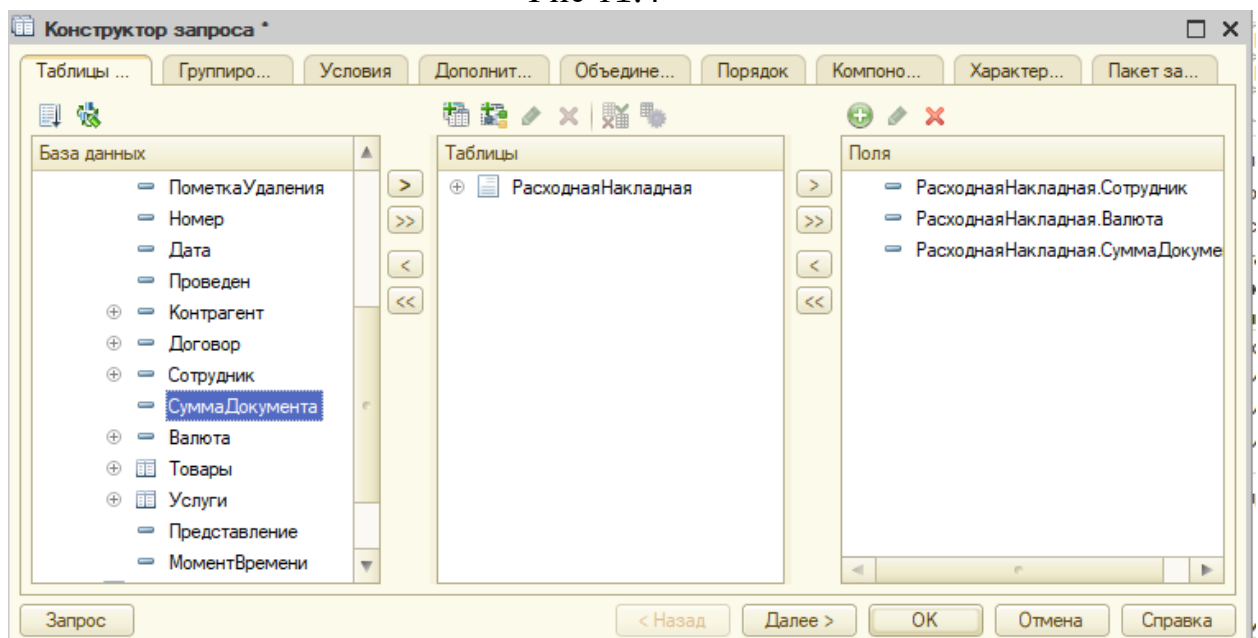


Рис 11.5

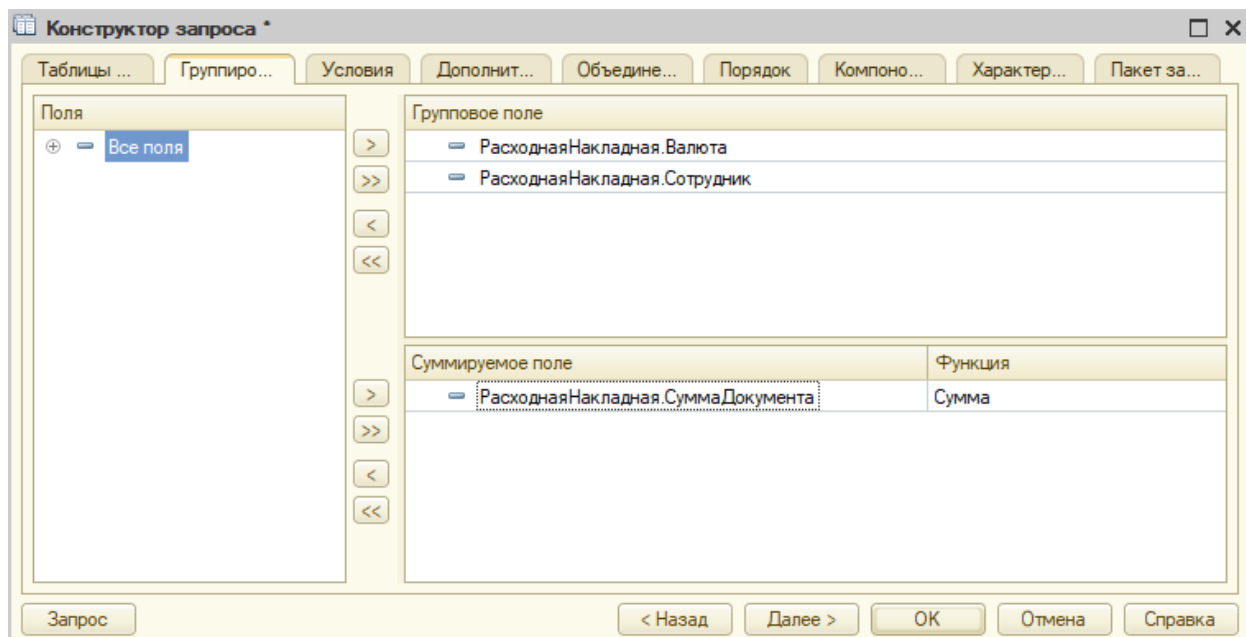


Рис 11.6

В настройке схемы компоновки укажем на закладке Ресурсы в качестве ресурса СуммаДокумента (Рис. 11.7).

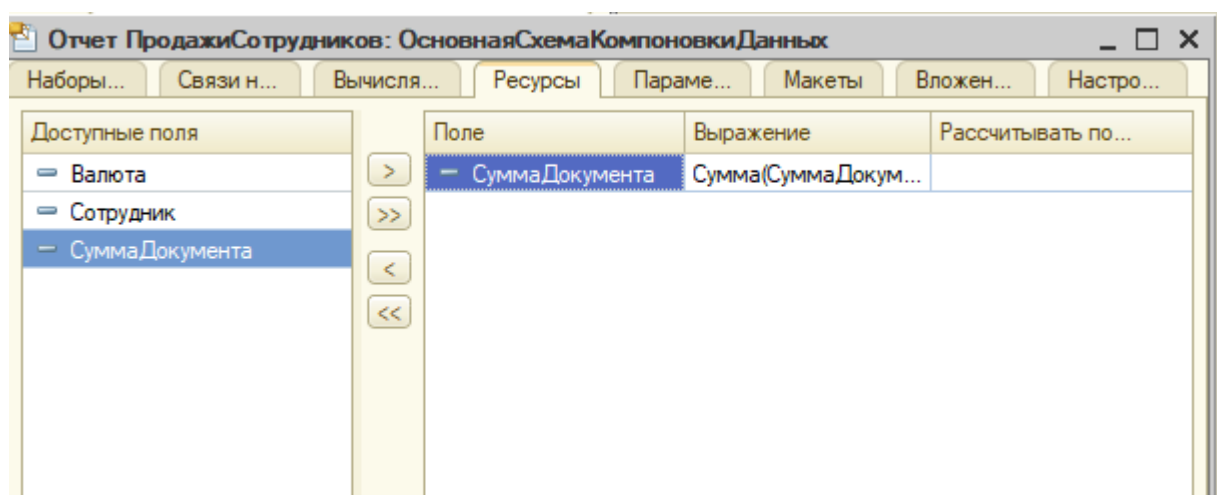


Рис 11.7

На закладке Параметры снимем ограничение доступности с параметров даты начала и конца периода отчета (Рис. 12.8)

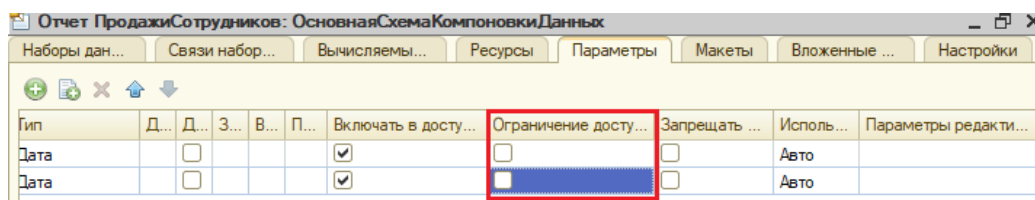


Рис 11.8

На закладке Настройки определим внешний вид нашего отчета. Для этого добавим новую *таблицу* и зададим у нее группировки строк (Рис. 12.9). Кроме того, в свойствах параметров для обоих дат укажем, что они

включаются в пользовательские настройки и установим для них обычный режим редактирования.

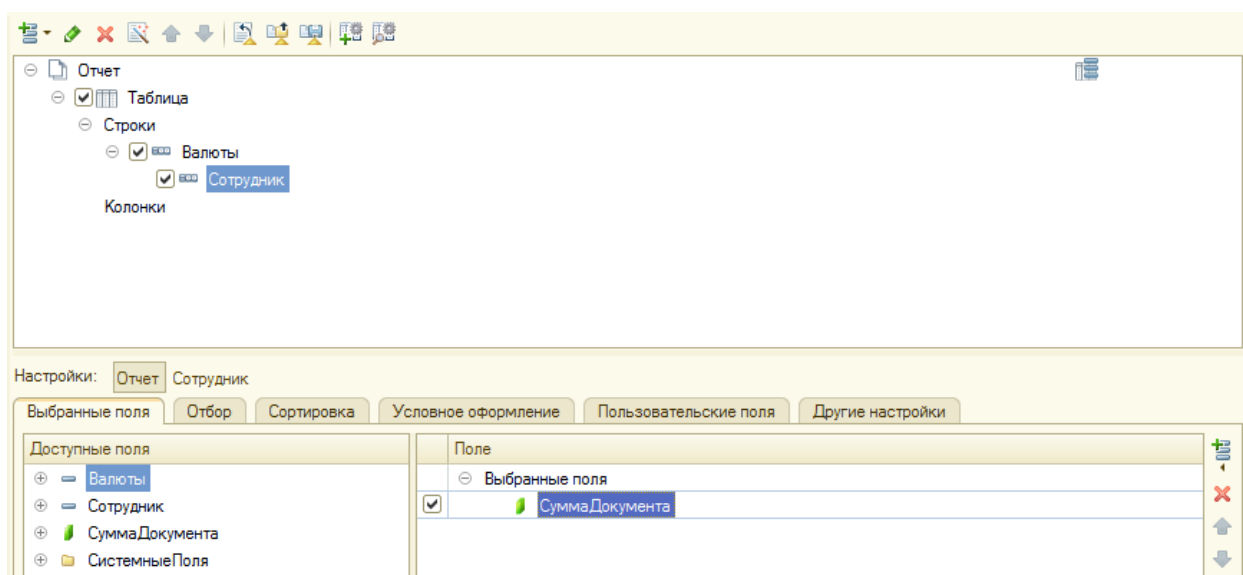


Рис 11.9

Там же, на закладке Выбранные поля выберем поле СуммаДокумента (Рис. 11.10)

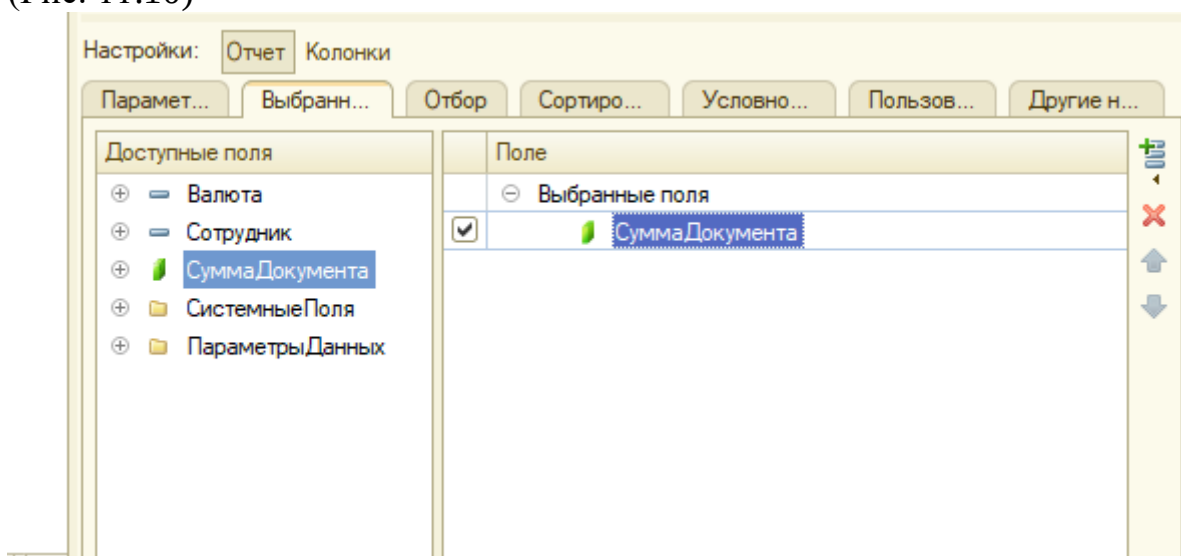
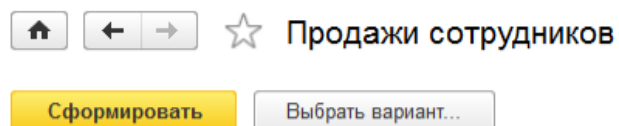


Рис 11.10



	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 11.11

С одной стороны, мы решили поставленную задачу и предоставили пользователю нужные данные в красивом и удобном виде. Однако, с другой стороны, как быть, если теперь пользователя заинтересовала та же самая информация, но в другом разрезе? Если теперь нас в первую очередь интересует не валюта, а сотрудник? Получается, что все необходимые данные в отчете присутствуют, но пользоваться ими неудобно. Конечно, можно сделать еще один отчет для решения этой новой задачи, но вместо этого механизм компоновки данных предлагает нам новый механизм – варианты отчета. Вернемся в режим "Конфигуратор" в настройки компоновки нашего отчета. Все необходимые данные уже получены нами с помощью запроса, а вот их отображение определяется нами на закладке "Настройки", где в левом окне мы добавляем новый вариант, для которого можем совершенно новым образом определить группировки и выбранные поля (Рис. 13.12)

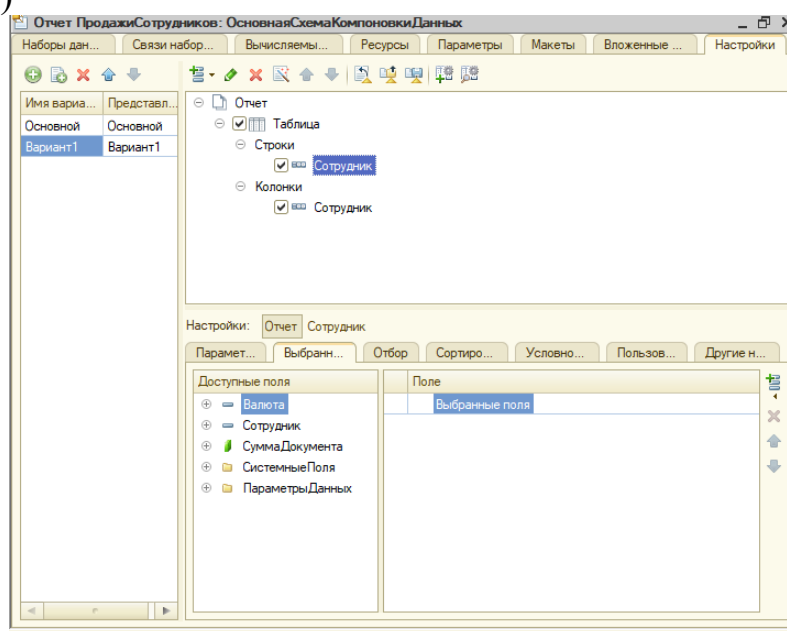


Рис 12.13

Не забудем проделать для варианта1 проделать действия, что отображены на Рис.11.14-11.15

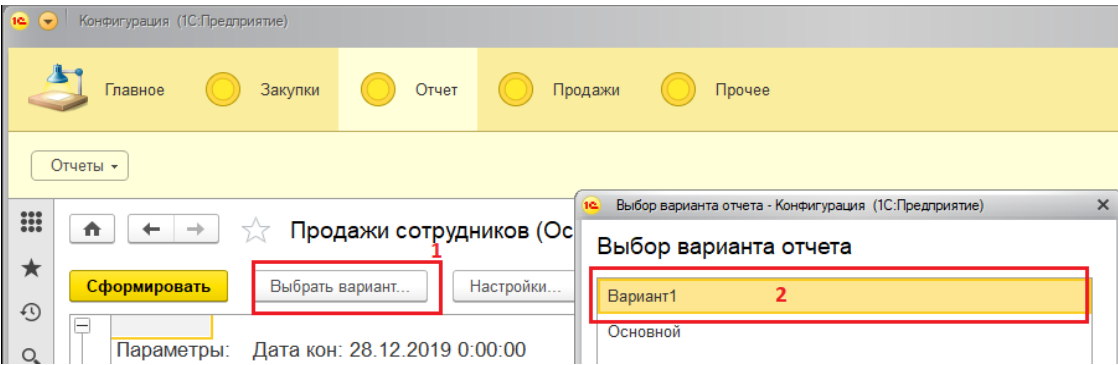


Рис 11.14

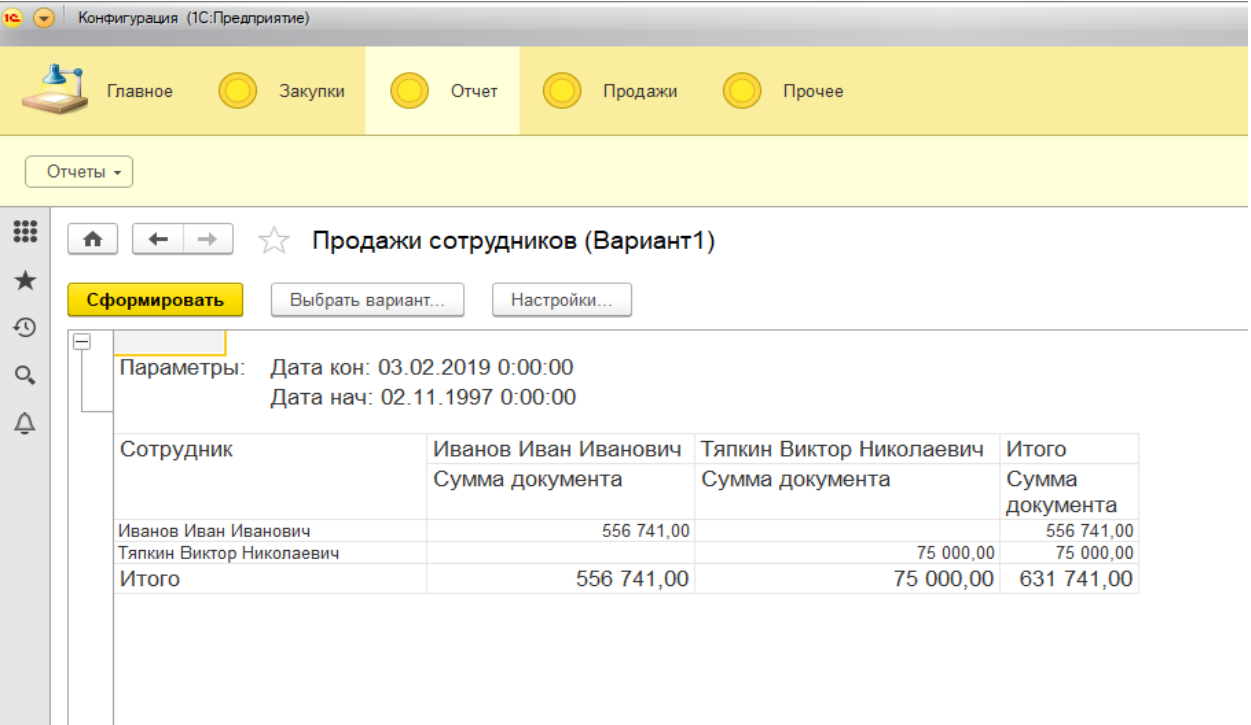


Рис 11.15

Практическая работа №10

Создайте отчет о продажах подразделений (Рис. 12.15)

