Лабораторная работа 1.

***Знакомство с 1С***

ПОСТАНОВКА ЗАДАЧИ:

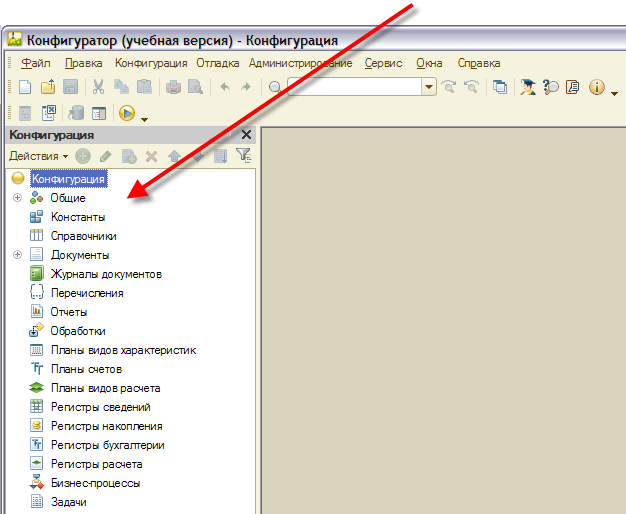
Создать простейшую систему CRM, которая, с одной стороны, позволит вести базу контактов (хранить информацию обо всех наших друзьях и знакомых). С другой стороны, будет учитывать различного рода события – как прошедшие, так и будущие. С возможностью фиксировать финансовые потоки – поступление и расход денег, связанные как с друзьями, так и с событиями, происходящими в жизни.

Любая программа 1С собирается из готовых компонент, набор которых ограничен, платформа изначально знает, для чего предназначены те или иные компоненты и как они между собой взаимодействуют.

Таким образом, разработчику достаточно просто добавить в информационную базу нужные компоненты, и платформа самостоятельно обеспечит их правильное функционирование.

Для внесения измененного поведения компонентов платформы используется встроенный язык и язык запросов. С их помощью можно описывать особенные алгоритмы взаимодействия, можно создавать собственные алгоритмы обработки данных и т. д.

В 1С:Предприятии компоненты называются *объектами конфигурации*. Все объекты конфигурации собираются в дерево.



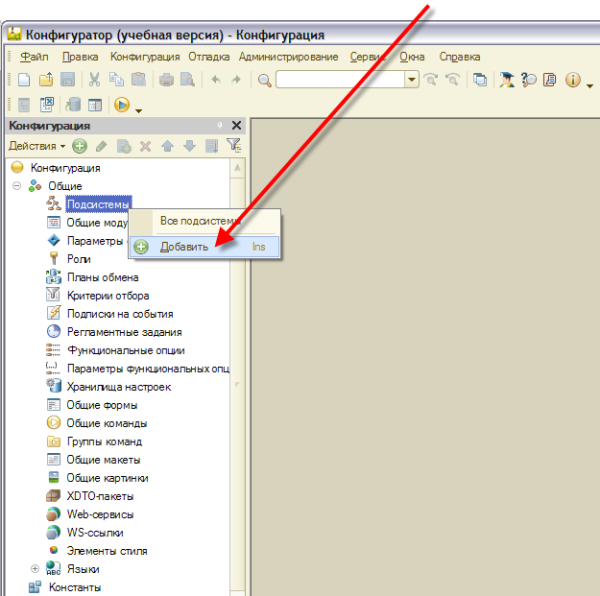
1. Необходимо определить подсистемы.

Подсистема служит для группировки объектов конфигурации по их функциональной принадлежности. Используя подсистемы, в дальнейшем легко структурировать пользовательский интерфейс.

Исходя из идеи создаваемого продукта, можно выделить три функциональные области:

* то, что относится к контактам;
* то, что относится к происходящим событиям;
* то, что связано с финансовыми потоками.

Добавим три подсистемы: *Контакты*, *События* и *Финансы*.

Добавление любых объектов конфигурации выполняется одинаковым образом. Нужно выделить ту ветку дерева конфигурации, в которой располагаются нужные объекты, и из контекстного меню выполнить команду *Добавить (Ins)*.

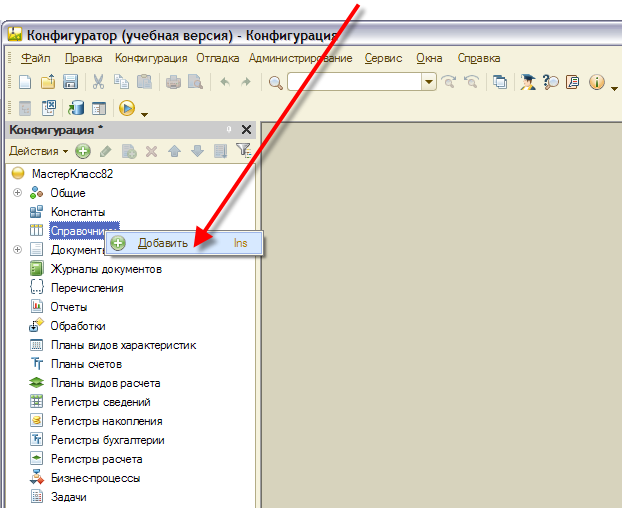
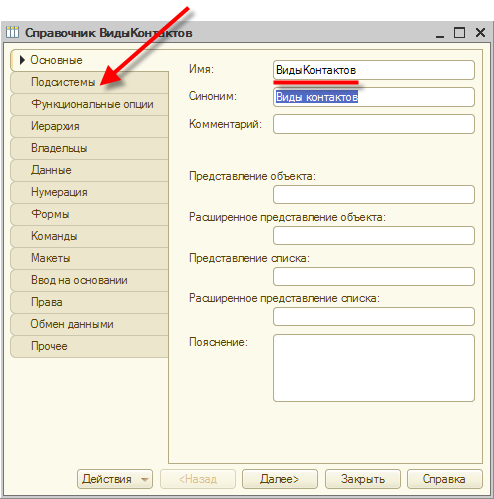
Подсистемы располагаются в ветке *Общие – Подсистемы*.

1. По условию поставленной задачи система хранит в себе списки знакомых нам людей и событий, происходящих в нашей жизни. При этом, кроме хранения собственно информации о своих знакомых, хотелось бы иметь под рукой список их контактных данных и как-то разделять (например: друг, знакомый, член семьи и т. п.). События тоже бывают разные, и нужно иметь способ разделять их по статусам.

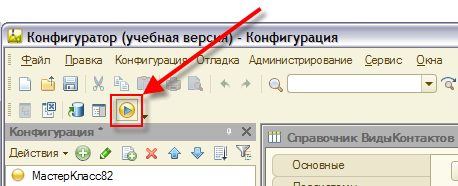
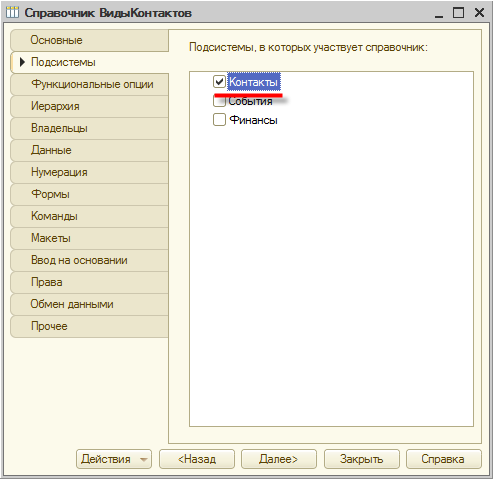
Таким образом, необходимо создать пять справочников

* *Друзья*, где будут храниться все записи о наших знакомых,
* *ВидыКонтактов*, для хранения списка различных вариантов контактных данных (например: «телефон», «адрес», «e-mail» и т. п.)
* *СтатусыДрузей* – список назначаемых нашим знакомым статусов (например, «семья», «друг», «знакомый» и т. п.).
* *События*, где, будут регистрироваться произошедшие и планируемые события
* *СтатусыСобытий*, для внесения разнообразных статусов (например, «учеба», «спорт», «отдых» и т. п.).

Справочники находятся в ветке дерева, которая так и называется –*Справочники*.

 , 

Необходимо обязательно указать к какой подсистеме относится справочник. Для проверки можно запустить систему в пользовательском режиме. Чтобы из режима конфигуратор запустить систему в пользовательском режиме (1С:Предприятие), нужно нажать кнопку *Начать отладку (F5)*.



Форму списков справочника система сгенерировала автоматически, основываясь на том, как он описан в дереве объектов. При этом доступны все команды работы со списком справочника: добавление новых элементов, удаление, поиск и пр.

Полностью аналогично справочнику *ВидыКонтактов* создаем справочник *СтатусыДрузей.* Длину поля *Наименование* увеличим до1*50* на вкладке *Данные.*



Запустите систему в режиме 1С:Предприятие и заполните справочники.

Создайте основной справочник, который будет хранить все контакты – *Друзья*. Аналогично предыдущим справочникам определите имя («Друзья»), привяжите к подсистеме *Контакты* и установите длину Наименования равной 150 (в нем мы будем хранить ФИО людей).

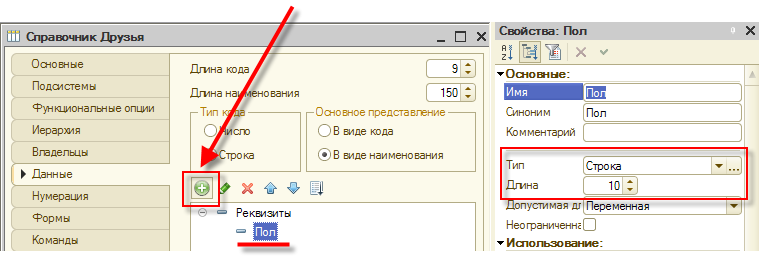
Для хранения дополнительной информации имеется возможность указать произвольное количество реквизитов в справочнике. Добавьте реквизиты *Пол, Статус и Комментарий*.

В реквизите *Пол* будем хранить пол знакомого.

В реквизите *Статус* – один из статусов ранее созданного справочника СтатусыДрузей.

В *Комментарии –* любую произвольную информацию о нашем знакомом, которую захочется сохранить.

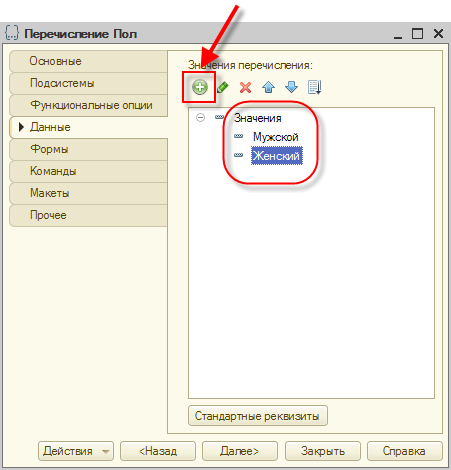
Добавить новый реквизит *Пол* кнопкой Добавить (Ins) на закладке Данные справочника Друзья.

.

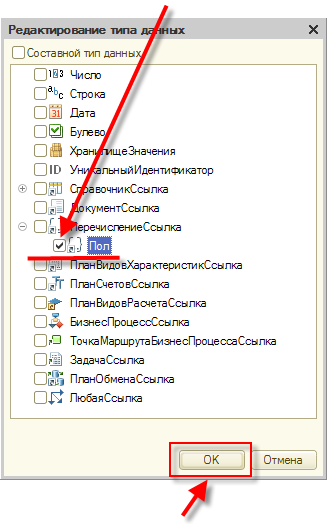
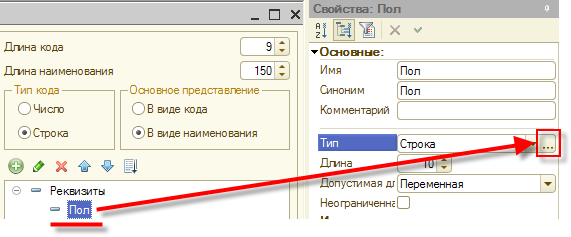
По умолчанию для нового реквизита предлагается строковый тип. Если оставить все как есть, то при заполнении справочника, каждый раз вписывать руками пол вновь добавляемых друзей. При этом высока вероятность ошибки (опечатки), что затруднит автоматический анализ этих данных в будущем, если таковой понадобится.

Создание нового объекта справочник для всего двух записей избыточная операция. Воспользуйтесь объектом конфигурации *Перечисление*.

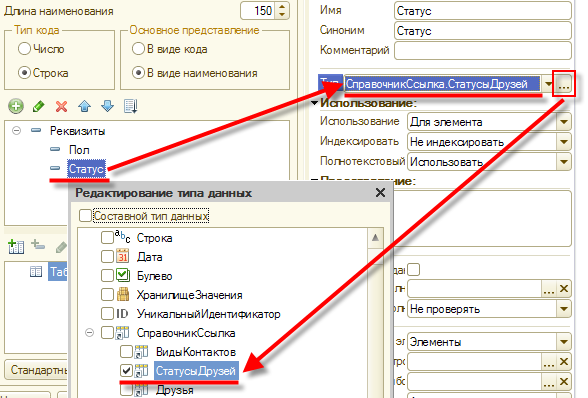
Перечисления – это объекты, которые задаются разработчиком на этапе создания конфигурации и хранят в себе ограниченный и заранее известный набор однотипной информации.



После того, как значения добавлены для реквизита *Пол* справочника *Друзья* можно изменить тип со «Строка» на тип только что созданного перечисления «Пол».



По аналогии с *Полом*, добавляем новый реквизит и выбираем для него тип «*СтатусыДрузей*», расположенный в Справочниках.

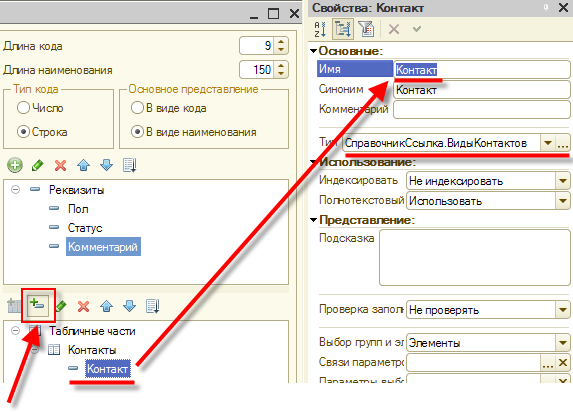
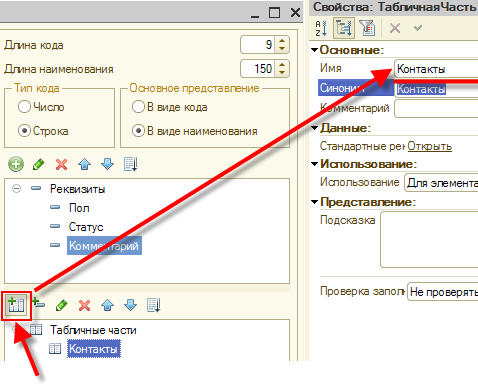


По условию задачи необходимо создать набор реквизитов, которые должны хранить в себе контактные данные *Друзей*. Добавьте реквизит *Контакт* типа *ВидКонтакта* (который будет хранить выбранный вид) и еще как минимум один реквизит, хранящий в себе значение вида контакта (пусть он так и будет называться – *Значение*). То есть для того, чтобы сохранить адрес знакомого, надо будет выбрать вид контакта «Адрес», а в реквизит значения записать собственно сам адрес.

Если создается пара «контактных» реквизитов там же, то в один момент времени для одного знакомого сможем сохранить только один контакт – ведь все реквизиты, которые мы добавляли до этого времени, относятся к одному элементу (записи) справочника.

Можно, добавить столько пар «контактных» реквизитов *Контакт-Значение,* сколько нам нужно на текущий момент времени. Но тогда не будет соблюдаться принцип , что у каждого друга свой набор данных. Кроме этого, если в будущем появится новый вид контакта, придется в конфигураторе вновь добавлять очередную пару реквизитов *Контакт-Значение* и так каждый раз, когда будет появляться что-то новое.

В справочнике *Друзья* создать табличную часть *Контакты*. Воспользуясь кнопкой *Добавить табличную часть*, после чего назовем новую табличную часть «*Контакты*».



Аналогичным образом добавим реквизит З*начение* строкового типа (длина строки – *1024*).



На этом создание справочника *Друзья* можно считать законченным. Запустите пользовательский режим и проверьте работу. (F5). Заполните список друзей.

Кроме форм отображения списков элементов, содержащихся в справочниках, система автоматически построила и форму элемента (записи) справочника.



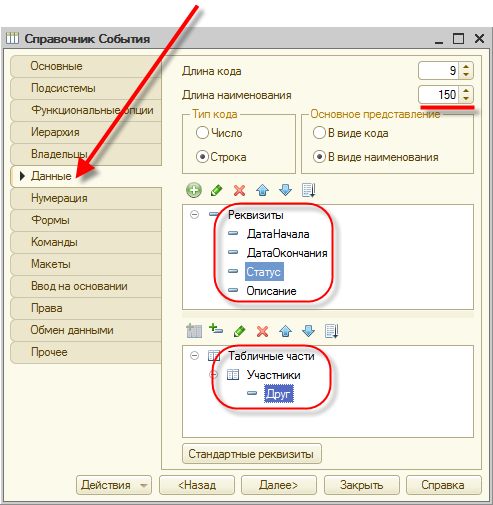
Создать справочник *СтатусыСобытий* по аналогии с *СтатусыДрузей* – длина наименования 150, принадлежит подсистеме *События*.



Создать справочник нашего примера –События. Отнести его к подсистеме События. Длина наименования 150. Набор дополнительных реквизитов:

* *ДатаНачала*, тип – Дата, состав даты – Дата и Время
* *ДатаОкончания*, тип – Дата, состав даты – Дата и Время
* *Статус,* тип – СправочникСсылка.СтатусыСобытий (воспользуйтесь кнопкой выбора типов)
* *Описание*, тип – Строка, неограниченная длина

В табличные части добавить новую табличную часть *Участники*, с единственным реквизитом –*Друг* типа СправочникСсылка.Друзья.



Проверить результат работы.

Таким образом, справочник *Друзья* может содержать записи обо всех наших родственниках, знакомых и друзьях (включая разнообразные контактные данные этих людей). Справочник *События* – информацию о произошедших и планируемых событиях (включая возможность указания друзей, участвующих в том или ином событии). При этом три «вспомогательных» справочника (*Виды контактов, Статусы друзей и Статусы событий*) помогают гибко настраивать учет различных данных, связанных с «основными» справочниками.

1. Добавить регистр накопления *Финансовые операции.*

По условию задачи система должна не просто хранить список знакомых и учитывать разнообразные события. Она должна уметь вести базовый финансовый учет – как минимум регистрировать денежные потоки и предоставлять минимальную отчетность.

Регистрация различных событий, связанных с поступлением и затратой денег, будет производиться такими объектами конфигурации, как *Документы.*

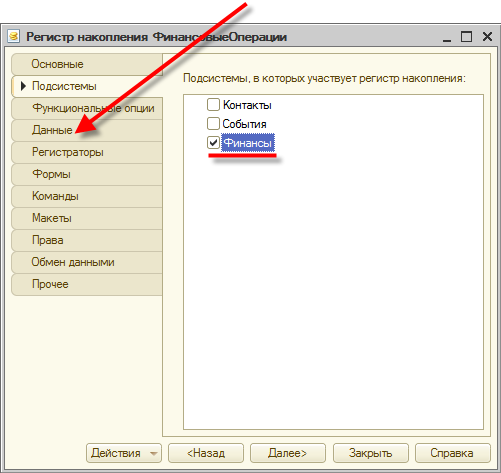
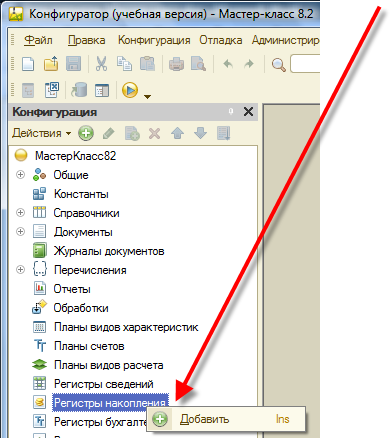
Кроме, собственно, регистрации фактов поступления и расходования денежных средств, необходимо получать отчеты, наглядно отображающие движение средств. Такие отчеты можно было построить по тем данным, которые содержатся во вводимых пользователем документах поступления и расходования денег. Однако, если через месяц (или год), вы решите дополнить свой финансовый учет новыми документами или функционально расширить уже существующие, отчеты будут работать неправильно, так как они учитывают только ту часть поступлений и расходов, которая оформляется старыми документами. Значит, придется переделывать и все отчеты. А их в реальной конфигурации может быть несколько десятков.

Чтобы избежать этой работы и большого количества ошибок, которые могут возникнуть в результате такой переделки отчетов, в системе «1С:Предприятие» применяется следующий подход.

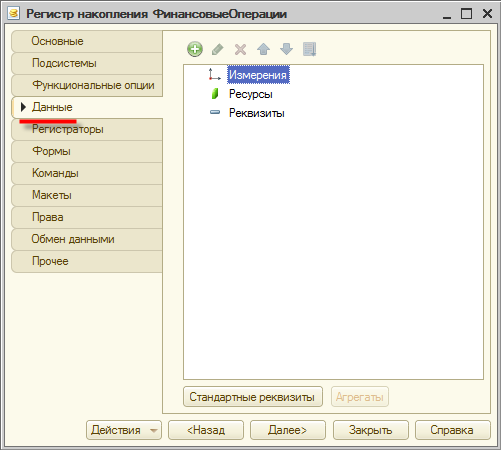
Для хранения данных о движениях каких-либо сущностей (например, финансов) используются специальные объекты конфигурации – регистры. Документы, которые отражают события финансовой деятельности, заносят свои данные в регистры. А отчеты строятся по данным этих регистров.

В результате создается независимая структура. С одной стороны, однажды создав регистры нужной структуры, можно сразу же создать нужные отчеты, которые будут красиво и понятно показывать пользователям данные, содержащиеся в этих регистрах. С другой стороны, при появлении новых видов документов в системе необходимо будет лишь правильно написать алгоритм, по которому их данные помещаются в эти регистры. При этом все созданные нами отчеты не потребуют переделки и будут показывать правильные данные.

Создать регистр накопления – *Финансовые операции* для хранения данные о том, как часто поступают деньги и как часто расходуются. Регистры накопления находятся в ветке дерева объектов конфигурации, которая так и называется: Регистры накопления

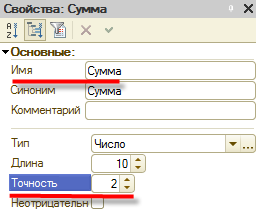
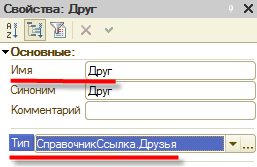
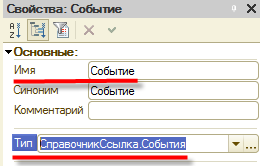
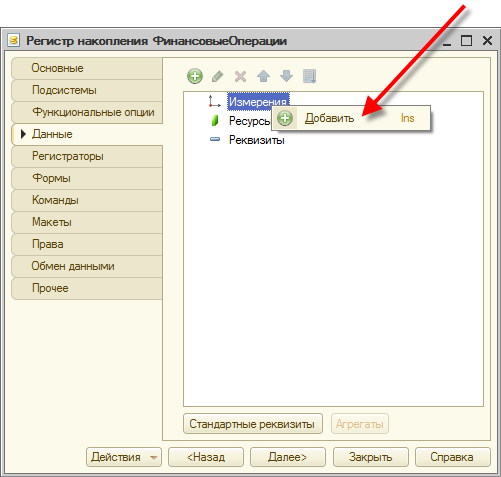


Добавить измерения, ресурсы и реквизиты регистра.

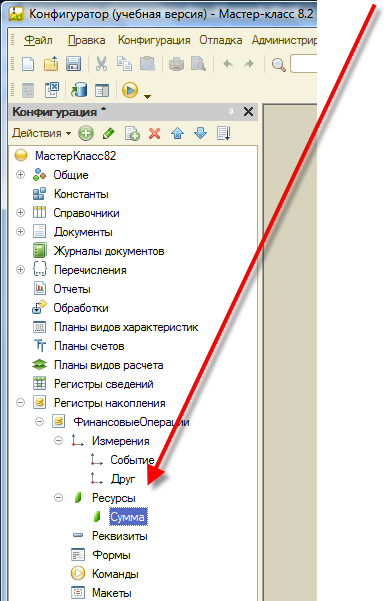


**Ресурсы** – это данные, которые получают из регистра. В нашем случае - сумма средств. Значит, нужен всего один числовой ресурс – Сумма.

**Измерения** – это то, в каких разрезах необходимо получать данные из регистра. Например, для анализа финансовых потоков может оказаться важным, какие события или знакомые «приносят» больше всего денег или, наоборот, – «потребляют». Поэтому понадобится два измерения – Событие и Друг.

**Реквизиты** – это дополнительная информация, сопровождающая каждую запись в регистре. 

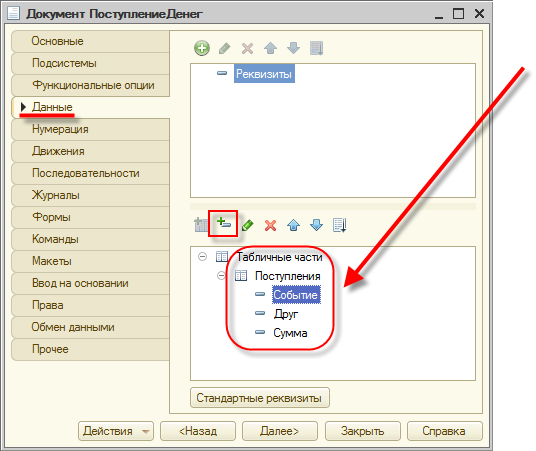
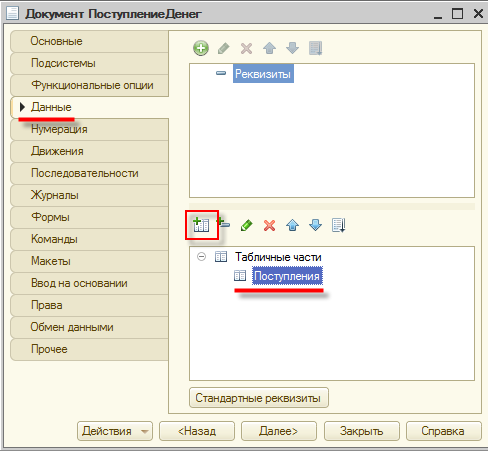
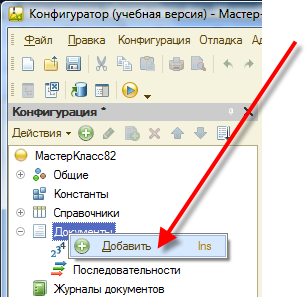
В результате регистр накопления *ФинансовыеОперации* будет выглядеть следующим образом.

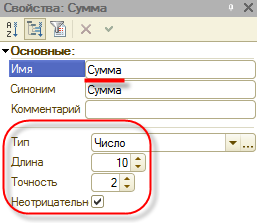
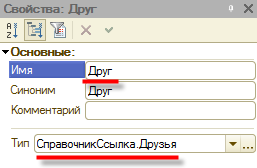
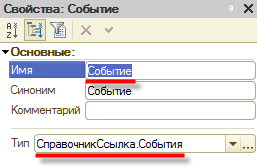


1. Добавить документы, регистрирующие поступление и расход денежных средств.

Документы находятся в ветке дерева объектов конфигурации, которая так и называется –*Документы*. Добавим новый документ. Назовем его *ПоступлениеДенег*, привяжем к подсистеме *Финансы* и перейдем на закладку *Данные*.

У каждого документа стандартно есть два реквизита – номер и дата создания. Позволим регистрировать одним документом несколько однотипных событий. Например, удобно за один день зафиксировать все поступления в одном документе, чтобы потом упростить поиск и анализ этой информации. Для этого используется **табличная часть**. Создаем табличную часть *Поступления*.

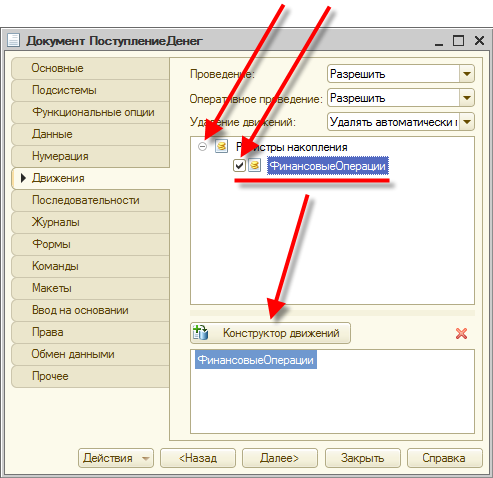




Документы записывают данные в регистры, а отчеты показывают данные из регистров в том виде, который удобен для пользователя.

**Движения документа**– это те записи, которые документ делает в регистрах.

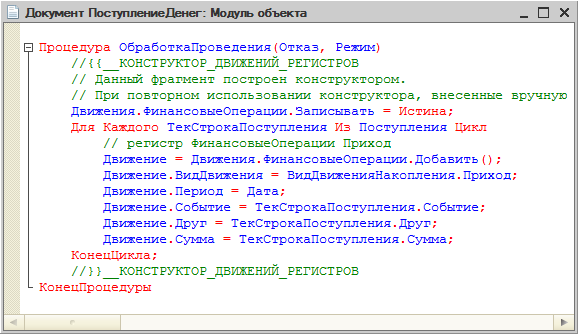
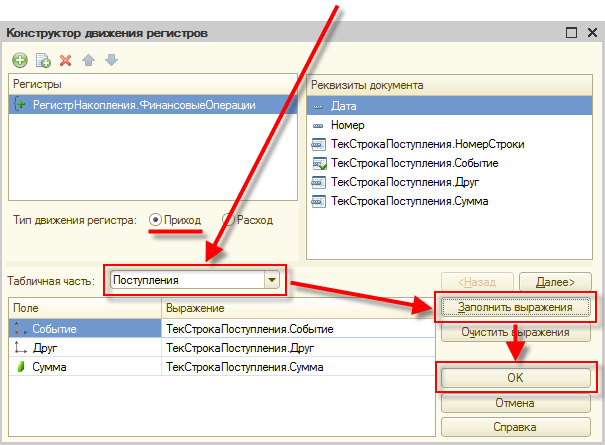
Раскройте ветку *Регистры накопления*, установите флажок, что документ *Поступление денег* будет двигаться по регистру *ПоступлениеДенег,* Запустите *Конструктор движений*, чтобы система помогла сформировать алгоритм записи данных в этот регистр.

**

Откроется конструктор движений документа. Установить тип движений «Приход», т. к. наш документ будет регистрировать поступление денег, выберите табличную часть – *Поступления* и нажмем кнопку *Заполнить выражения*.

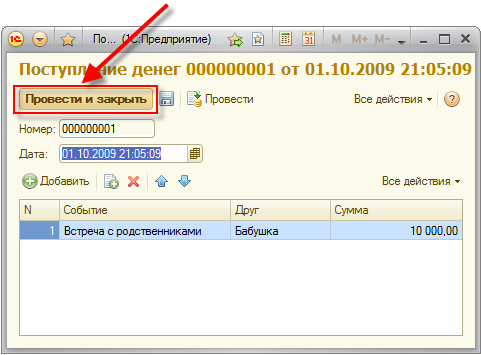
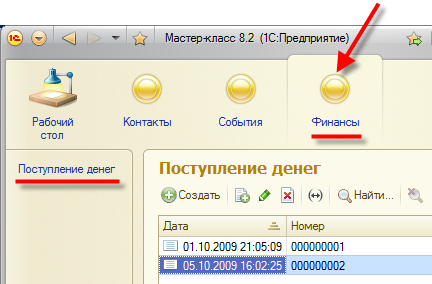
В таблице, расположенной в нижней части окна система автоматически установила соответствие между реквизитами регистра накопления и реквизитами документа. Так, в измерение регистра *Событие* будут записаны данные реквизита *Событие* документа, в измерение *Друг* – данные реквизита *Друг* табличной части документа, а в ресурс *Сумма* – данные реквизита *Сумма* табличной части документа.

Система сформирует процедуру записи данных документа в регистр и покажет ее на экране.

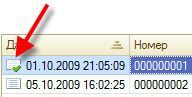
**

Запустите систему в режиме *1С:Предприятие* и заполните данные в документе *Поступление денег* Перейдите на вкладку *Финансы* и откройте список документов *Поступление денег.*

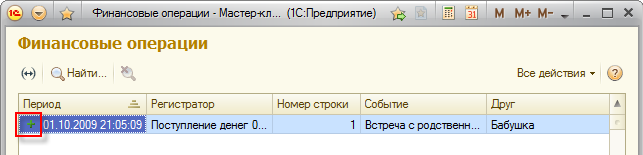
Для того чтобы добавленные данные повлияли на состояние учета, нужно записать и провести документ. В момент проведения будут добавлены записи в регистр накопления *Финансовые операции*.

**

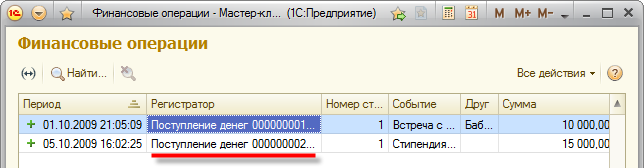
Факт успешного проведения документа в системе отражается в виде отдельного значка в списке документов.

**

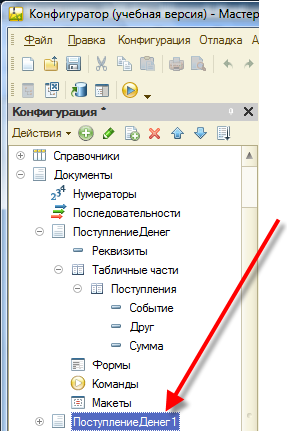
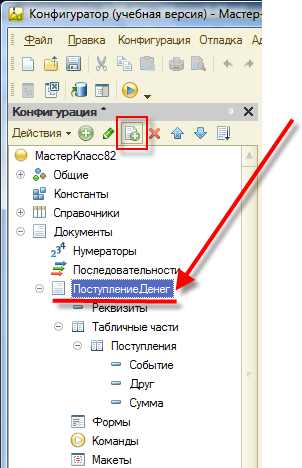
Проверить факт проведения, т. е. помещения нужной информации в записи регистра *Финансовые операции,* можно следующим способом: зайти в главное меню *> Все функции… >* раскрыть узел *Регистры накопления > Финансовые операции*. Откроется форма записей регистра накопления *Финансовые операции*. Обратите внимание на значок «+», который символизирует добавление информации к значениям ресурсов.

**

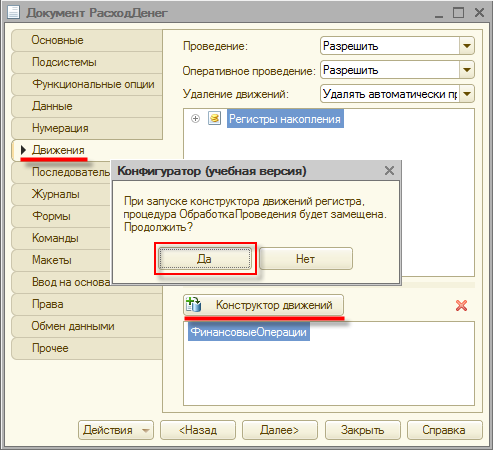
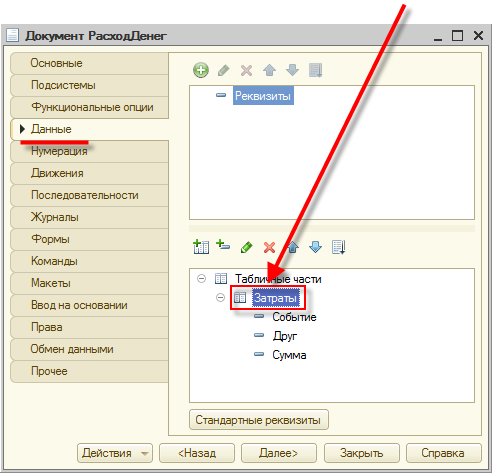
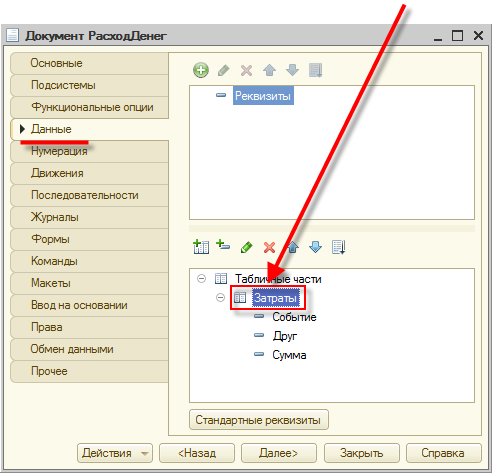
Проведите несколько из введенных доументов.

**

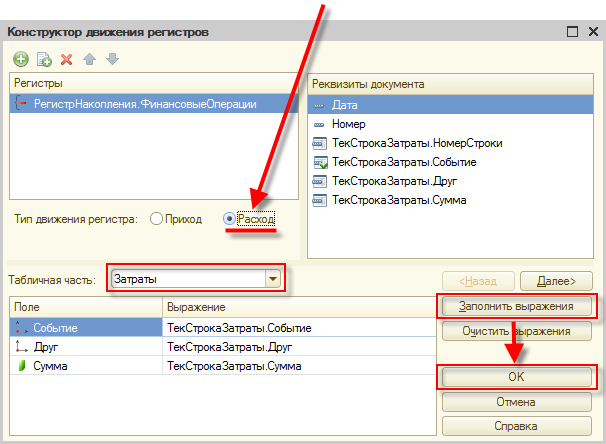
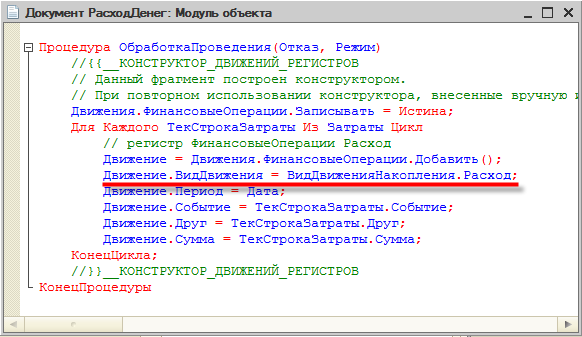
Зафиксируйте расходы. Возвращаемся в конфигуратор. Необходимо создать документ *РасходДенег*. Это можно выполнит вручную по образу существующего документа *ПоступлениеДенег.* А можно воспользоваться сервисом платформы «1С:Предприятие» – созданием новых объектов конфигурации на основании уже существующих путем их копирования. Для этого нужно установить курсор на существующий документ *ПоступлениеДенег* в списке объектов метаданных и нажать кнопку *Добавить копированием* (*F9*).

**

Переименовать этот документ и подправить его свойства. Изменить имя документа – *РасходДенег*, привязать новый документ к подсистеме *Финансы*, изменить название табличной части –*Затраты*. Исправить движения документа. Сейчас они скопированы из документа *ПоступлениеДенег*. Тот документ учитывал поступления денег. Запустим конструктор движений. Система предупредит нас, что существующая процедура проведения будет замещена.

**

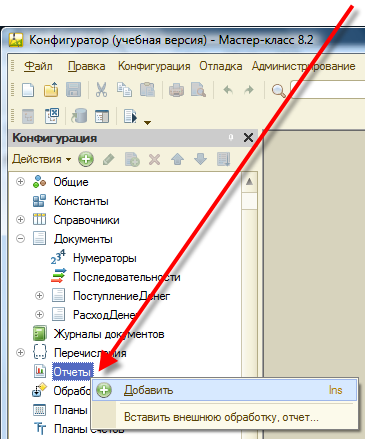
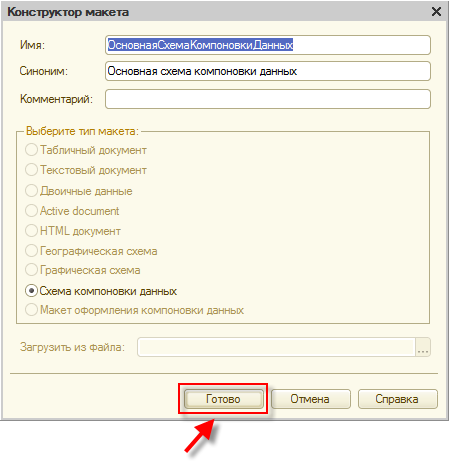
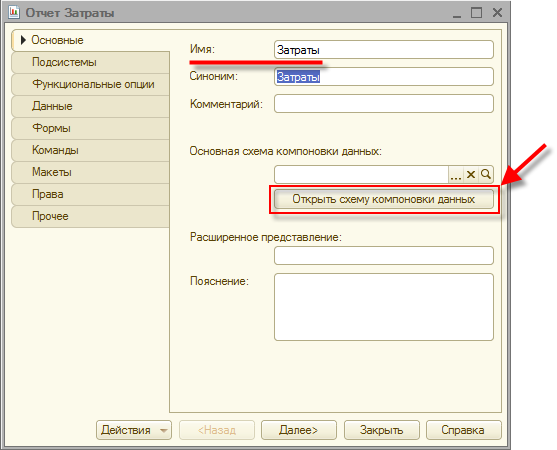
Изменить тип движения документа на *Расход* и для заполнения выбираем табличную часть *Затраты*.

**

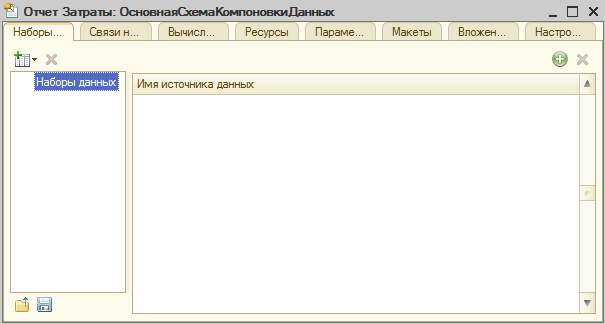
Можно заметить, что отличие от документа *ПоступлениеДенег* по сути всего в одной строке, которая задает вид движений. В нашем случае в регистр будут добавляться движения расхода.

В пользовательском режиме вводим данные по расходам. Провести три документа и проконтролировать, что записи о фактах затрат денежных средств отразились в регистре накопления *Финансовые операции*. «Расходные» записи отражаются в списке регистра значком «–». Кроме этого, видно, что несколько строк одного документа попадают в регистр в виде отдельных записей – обратите внимание на реквизит *Номер строки*.

1. Создание отчетов, отображающих в удобном виде информацию пользователя. Отчеты находятся в ветке дерева объектов конфигурации, которая так и называется – *Отчеты*. Добавим новый отчет  *Затраты*.

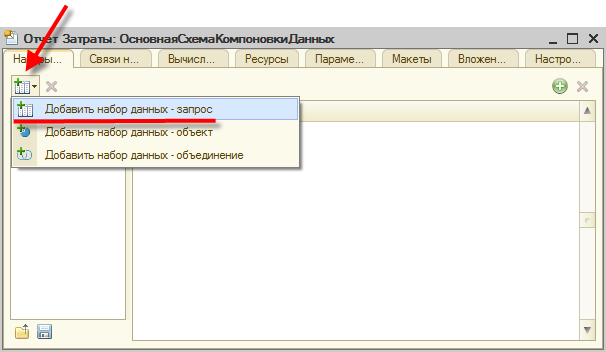
**

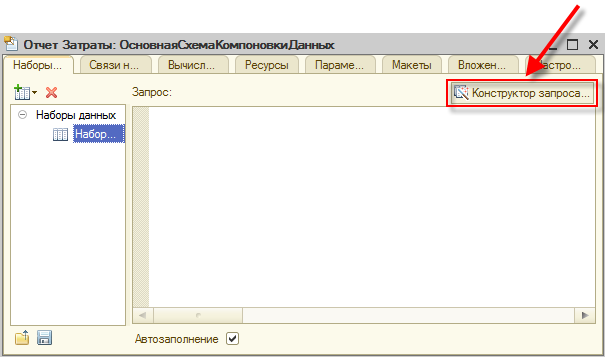
После этого система откроет конструктор схемы компоновки данных.

**

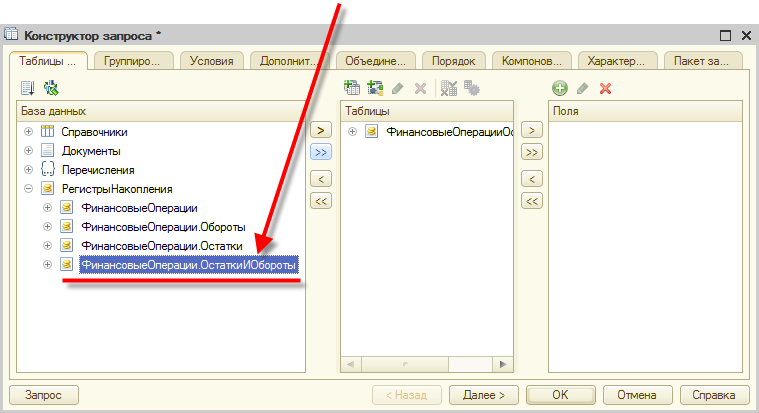
Описать набор данных, из которого отчет будет получать данные, описать структуру отчета.

Добавить набор данных – запрос. То есть данные для этого отчета мы будем получать запросом из базы данных «1С:Предприятия».

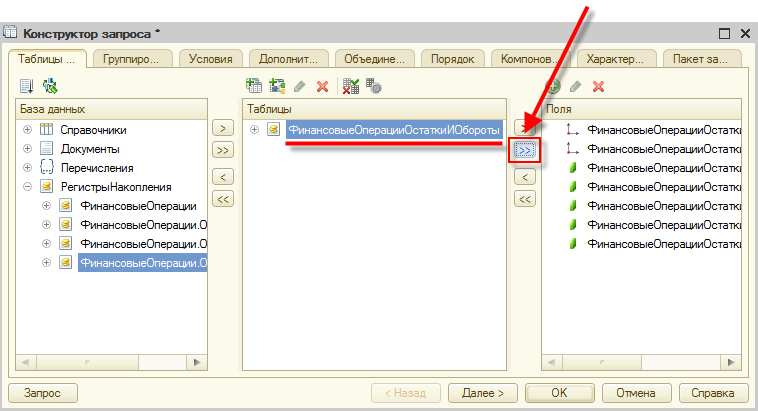
**

Можно ввести текст запроса вручную, а можно использовать конструктор запроса. 

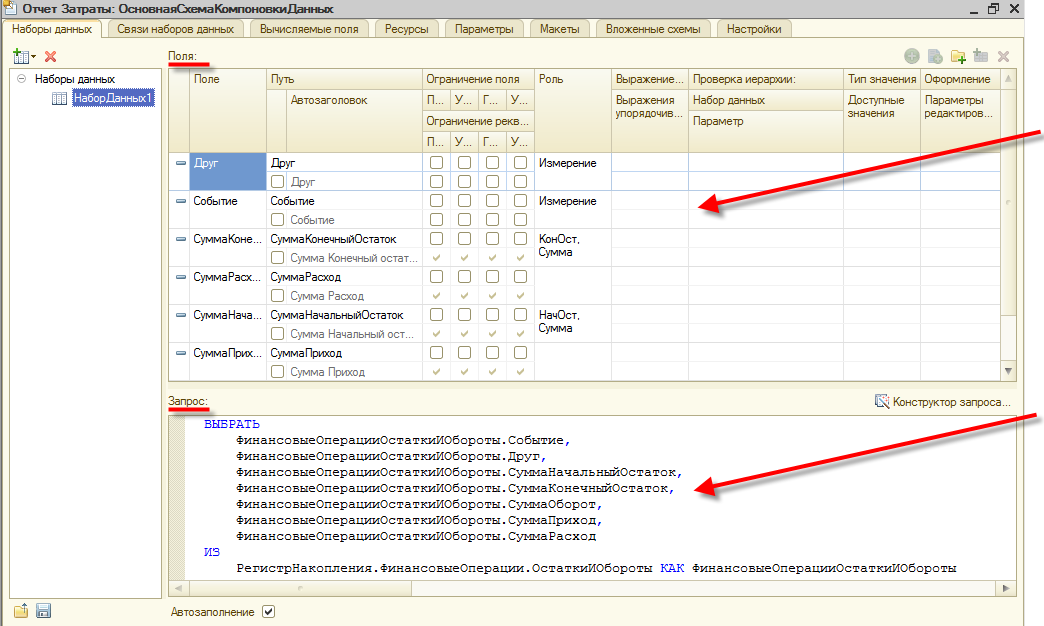
В открывшемся окне конструктора запроса слева перечислены таблицы, из которых можно получать данные.



*Добавить все поля* таблицы в качестве полей запроса.

**

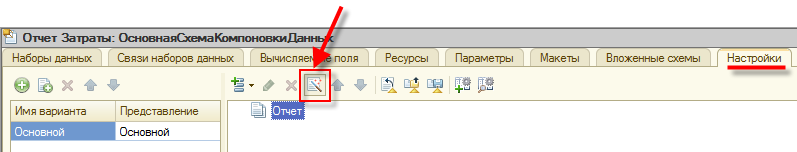
ПО нажатию Ok, система создаст текст запроса и автоматически заполнит поля системы компоновки данных.



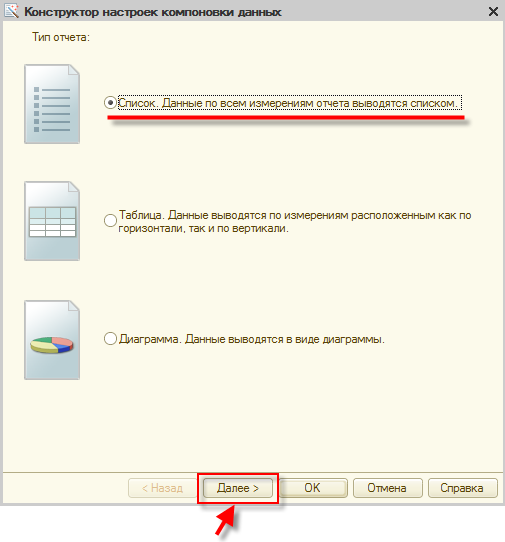
Перейдем на вкладку *Ресурсы*. Здесь двойным щелчком выберем следующие поля:

* *СуммаКонечныйОстаток,*
* *СуммаНачальныйОстаток,*
* *СуммаОборот,*
* *СуммаПриход,*
* *СуммаРасход.*

**

Перейдем на вкладку *Настройки*. Воспользуемся конструктором. **

Конструктор настроек позволяет быстро создать несколько простых вариантов настройки отчета. Для наших целей хорошо подойдет список.

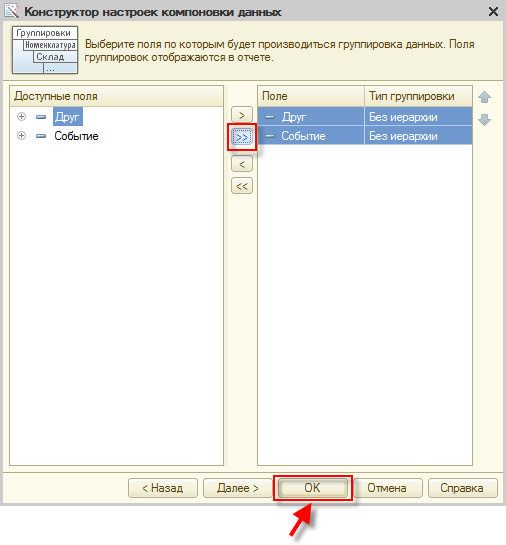
**

На следующем шаге выберем поля, которые будут отображаться в нашем отчете. Двойным щелчком в списке *Доступные поля* выберем следующие поля:

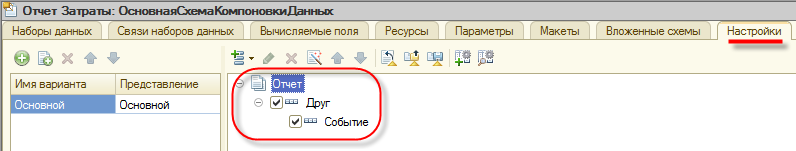
* *Друг,*
* *Событие,*
* *СуммаОборот,*
* *СуммаПриход,*
* *СуммаРасход.*

**

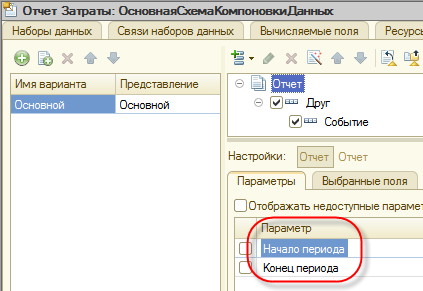
Выбрать поля, по которым данные в отчете будут сгруппированы. Требуется видеть движения по каждому знакомому в отдельности и еще по каждому событию, связанному с этим знакомым. Поэтому двойным щелчком из доступных полей выбираем оба – *Друг* и *Событие*.



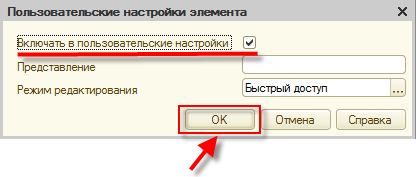
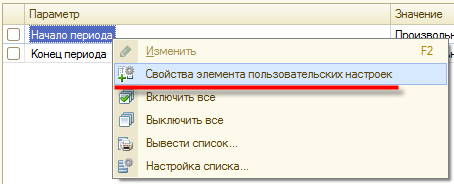
Система покажет структуру отчета.



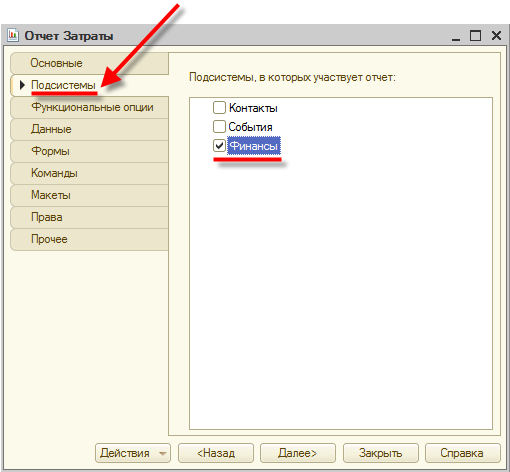
Дадим возможность пользователю отчета произвольно задавать период выборки данных –Ниже структуры отчета доступны два параметра: *Начало периода* и *Конец периода*.

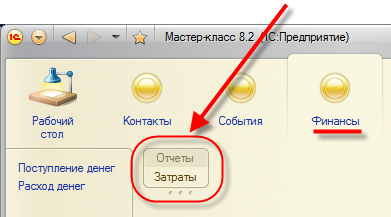
**

Достаточно выделить мышью каждый из параметров и выбрать вариант *Свойства элемента пользовательских настроек*.

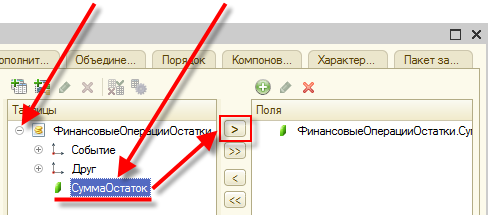
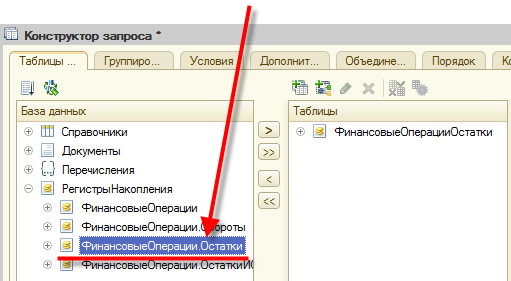
**

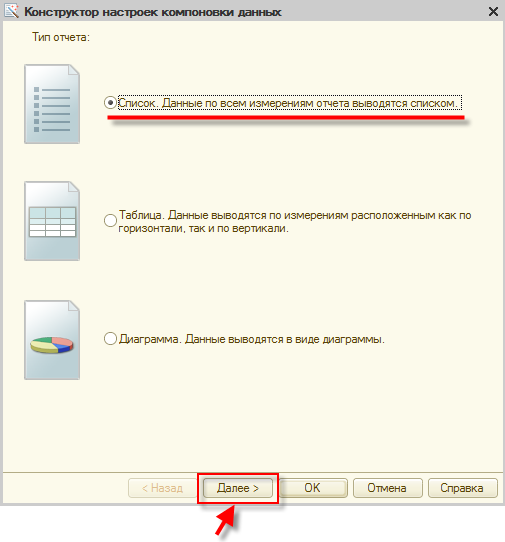
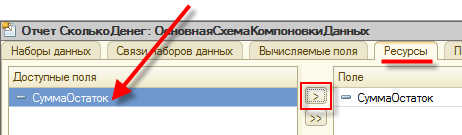
Выполнить эту операцию для параметров *Начало периода* и *Конец периода*. Привязать н отчет к подсистеме *Финансы*.

**

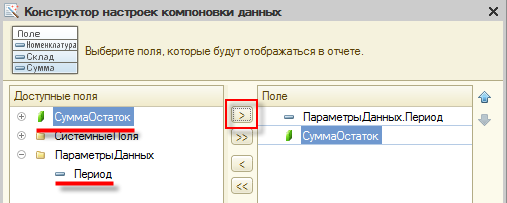
Запустить систему в режиме *1С:Предприятие* проверить отчет. Искать наш отчет следует на вкладке *Финансы*– обратите внимание, как система самостоятельно отделила его в отдельную область на экране. Открыв окно отчета, сформируем его, нажав кнопку *Сформировать*. **

Создать еще один отчет, способный быстро отобразить нам текущий остаток доступных денежных средств. Назовем его *СколькоДенег* и отнесем сразу к подсистеме *Финансы*. Аналогично отчету *Затраты* создадим схему компоновки данных. В качестве источника данных снова выберем *Запрос*.

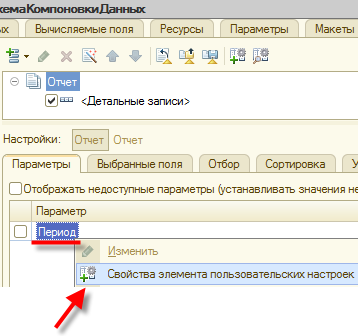
**

**

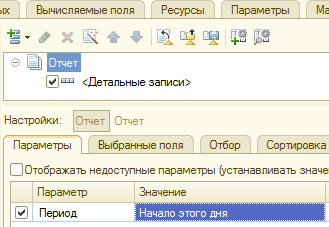
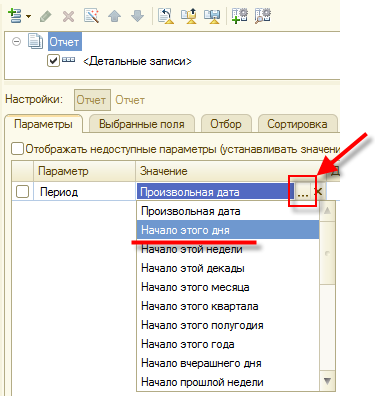
На следующей вкладке из *Параметров* данных выберем стандартное поле *Период* (чтобы иметь возможность видеть дату, на которую будет строиться отчет) и поле *СуммаОстаток*.

**

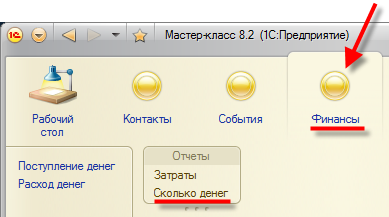
Параметр *Период*, как и в отчете *Затраты*, включим в пользовательские настройки.

**

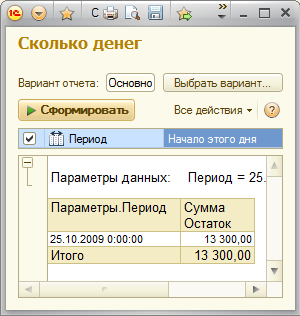
Значение параметра *Период* установим в *Начало этого дня*, чтобы отчет всегда автоматически был готов показать информацию на текущий день.

**

Запускаем пользовательский режим и проверяем работу отчета.

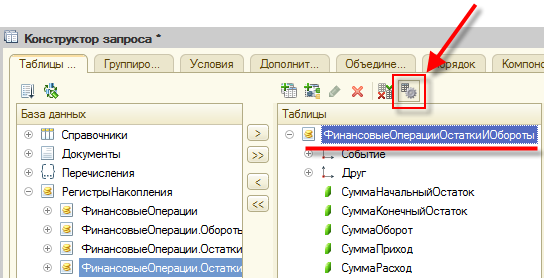
**

Информация, сообщаемая отчетом, проста, но полезна: вы всегда сможете посмотреть, сколько финансов доступно на текущий момент времени.

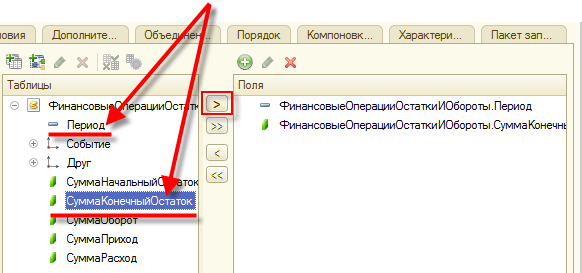
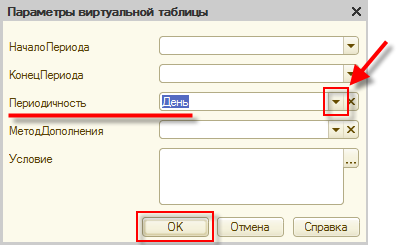
**

Создать отчет *ГрафикПоДням* в подсистеме *Финансы* В схеме источником данных вновь выберем *Запрос* и запустим *Конструктор запросов*.

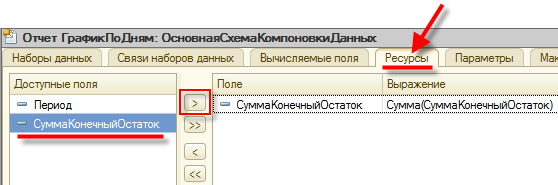
В качестве источника данных выберем таблицу *ФинансовыеОперацииОстаткиИОбороты*. Подготовим данные из регистра в нужном для отчета виде. Для этого отметим таблицу *ФинансовыеОперацииОстаткиИОбороты* в среднем окне конструктора запросов и нажмем кнопку *Параметры виртуальной таблицы*.

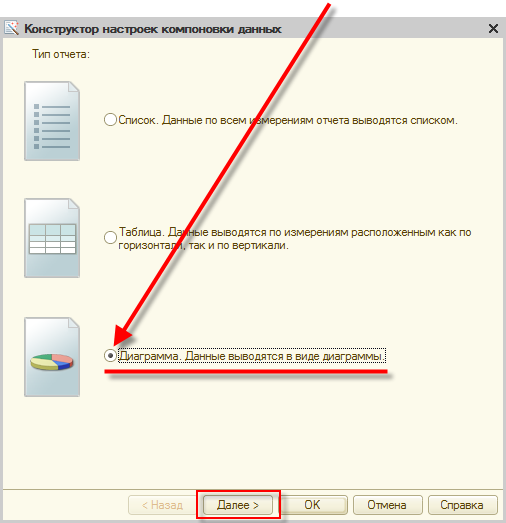
**

В открывшемся окне в поле *Периодичность* выберем *День*. Выберем два поля из нашей таблицы: *Период* и *СуммаКонечныйОстаток*.

**

На вкладке *Ресурсы* выберем поле *СуммаКонечныйОстаток*.

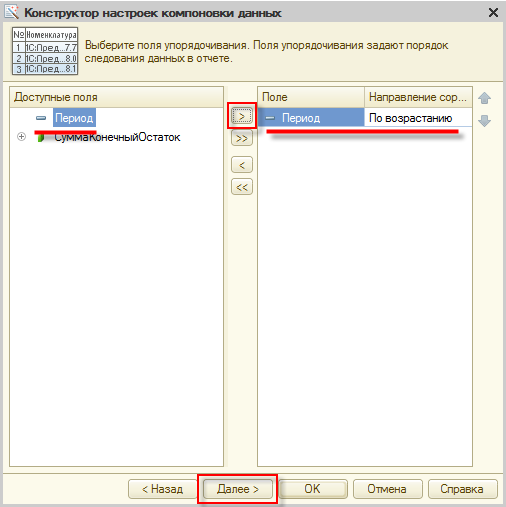
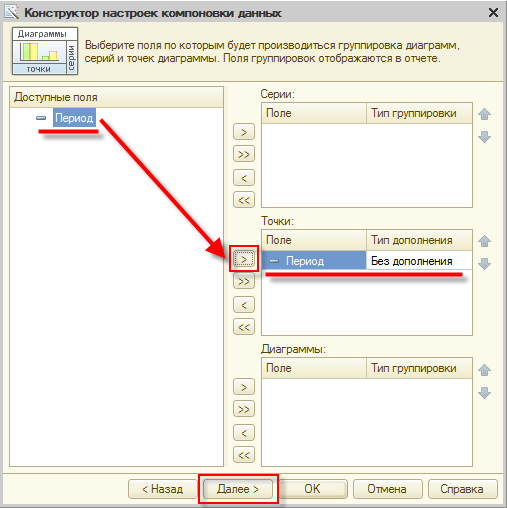
**

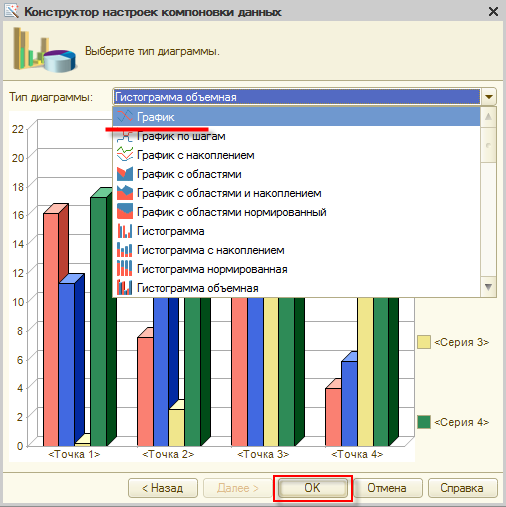
**

Выберем поля *Период* и *СуммаКонечныйОстаток*.

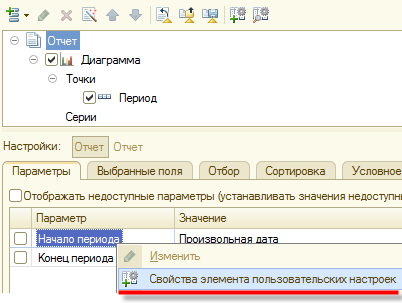
**

В следующем окне настройки диаграммы поместим поле *Период* в раздел *Точки*. В поля упорядочивания тоже добавим *Период.*

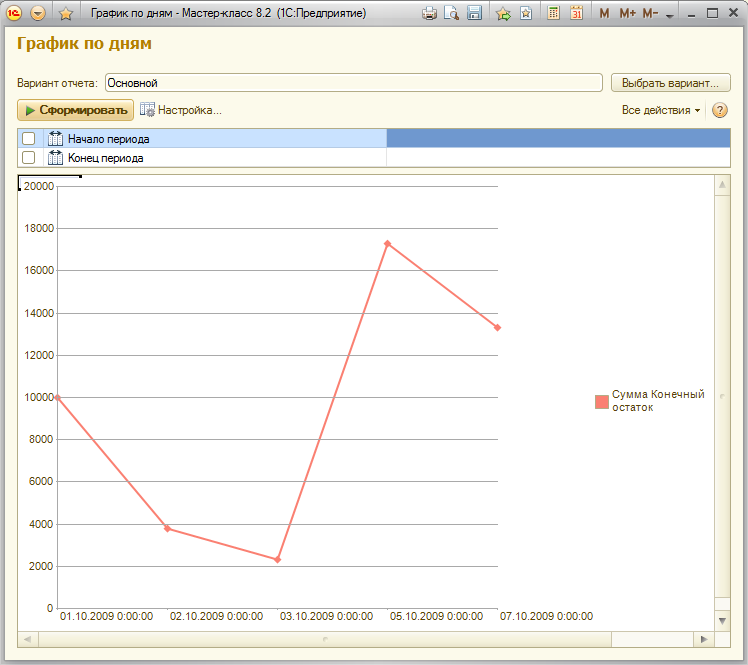
**

**

Как и в предыдущих отчетах, для параметров *Начало периода* и *Конец периода* активизируем включение в пользовательские настройки.

**

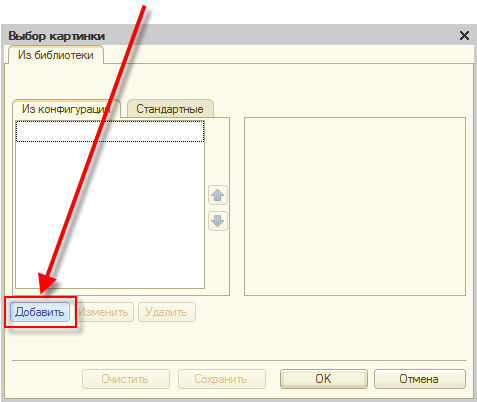
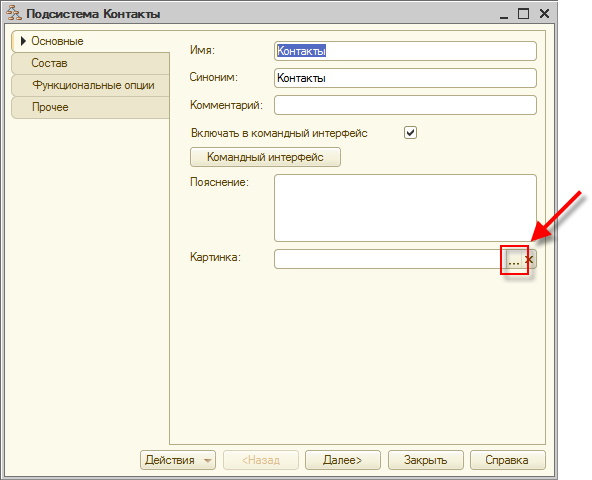
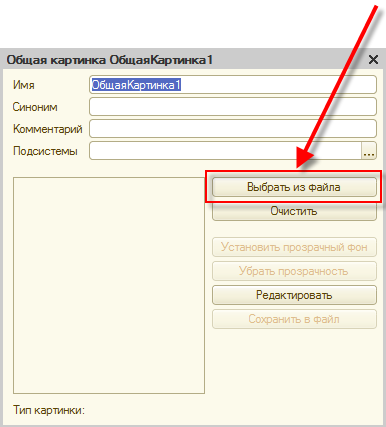
Проверим, что получилось, в пользовательском режиме. Запустим и сформируем отчет *График по дням*.

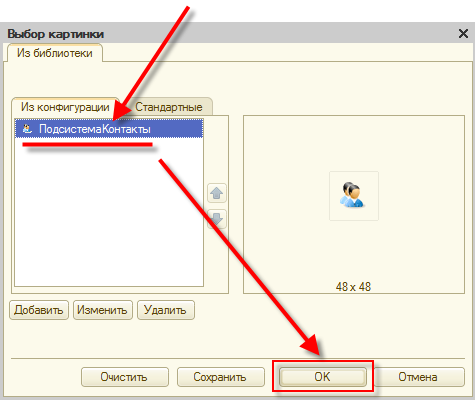
**

Лабораторная работа 2.

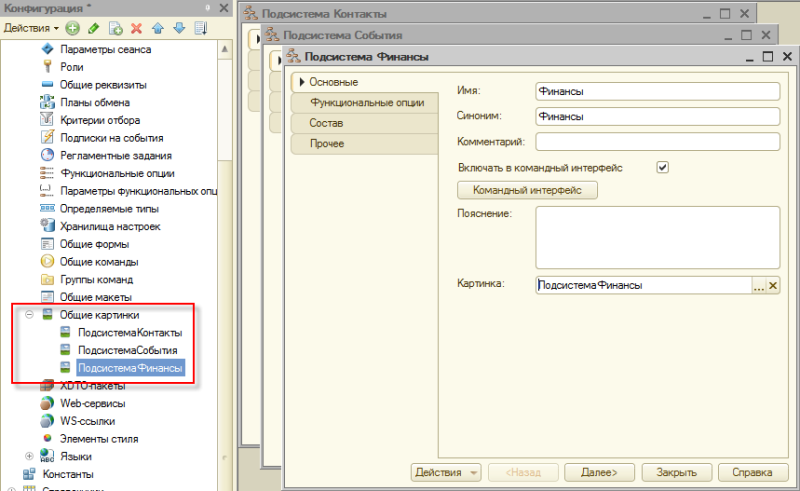
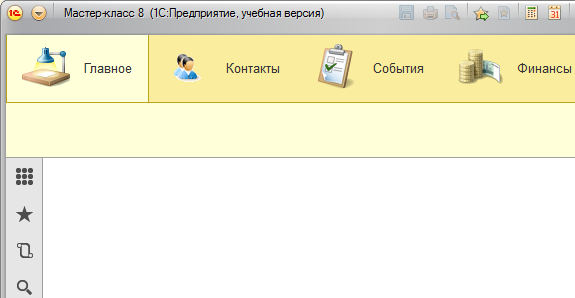
***Улучшение интерфейса и функциональные опции.***

Назначим различные картинки подсистемам. Открыть в конфигураторе окна параметров подсистем и добавить для каждой нужные картинки в поле *Картинка*. Необходим файл с расширением .png.



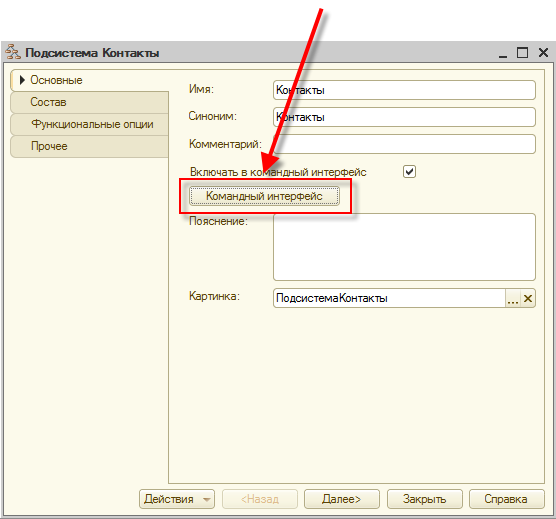
Аналогичным образом выбираем картинки для оставшихся подсистем.

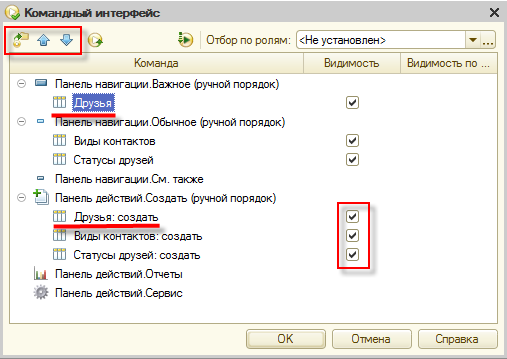
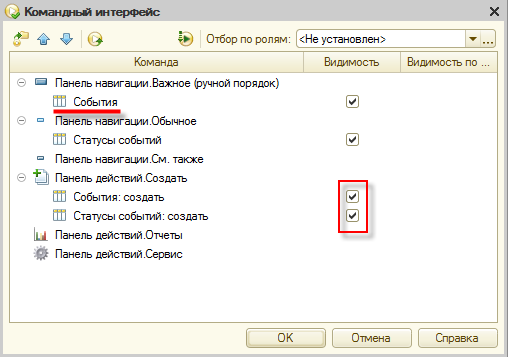
**Меняем состав подсистем**

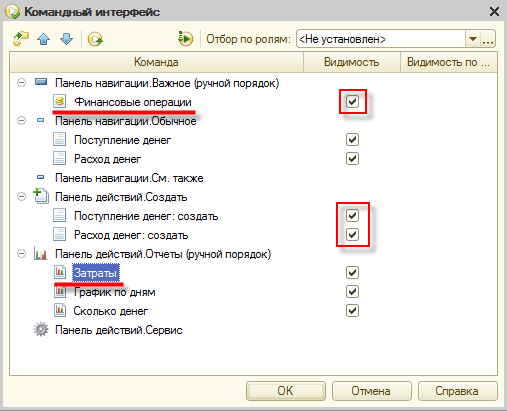
Попробуйте разнообразить и «облагородить» содержимое разделов. Откройте список доступных команд для той или иной подсистемы, посмотрите на них и включите те, которые могут быть полезны пользователю.

Настройте командный интерфейс в окне параметров подсистем.



Справочник *Друзья* поместить в раздел *Важное*. В разделе *Создать* включить возможность создания новых записей всех трех справочников и поменять порядок их расположения. В результате получится следующая настройка. Аналогично настроить *События* и *Финансы.*

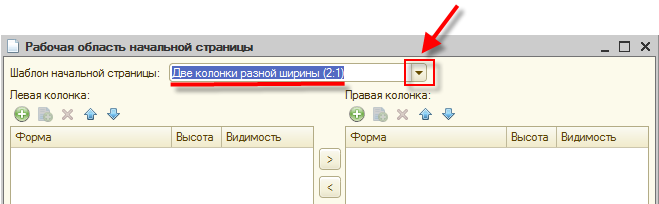
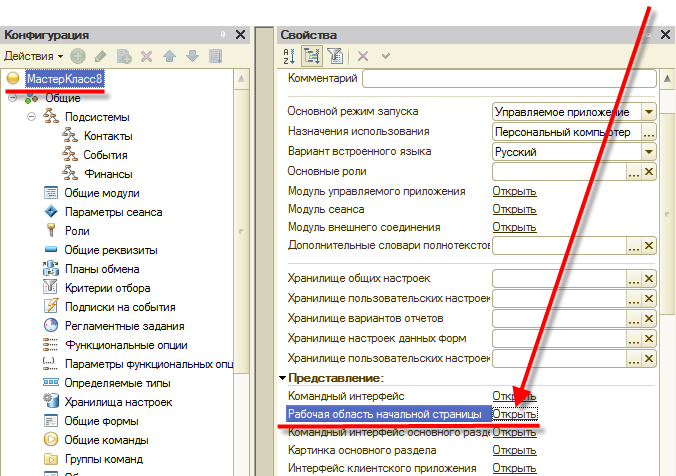
 



Проверьте настройки в пользовательском режиме.

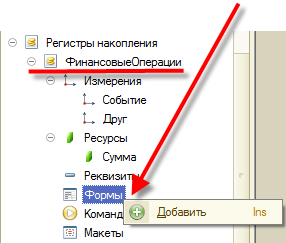
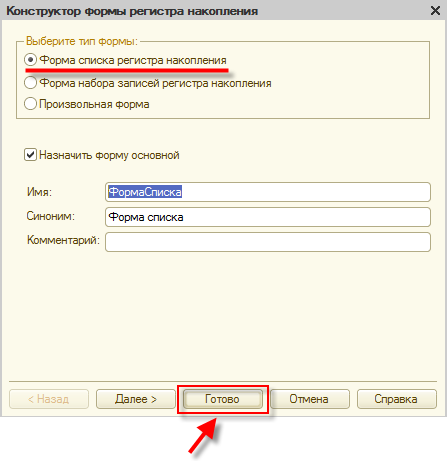
Настройте рабочий стол.

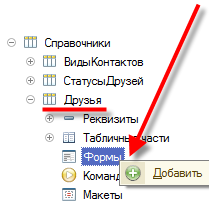
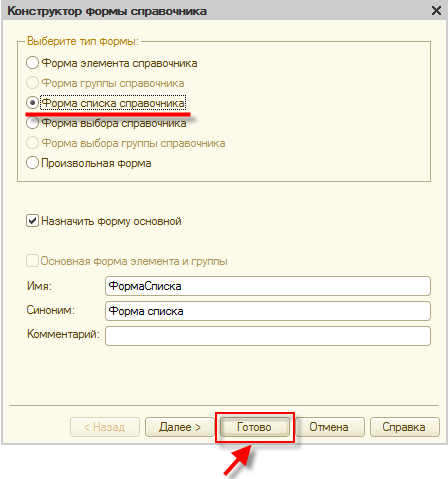
Открыть в свойствах конфигурации ссылку *Рабочая область начальной страницы*. Выбрать шаблон начальной страницы – *Две колонки разной ширины (2:1)*

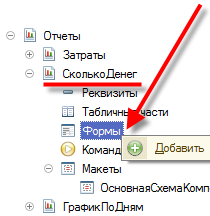
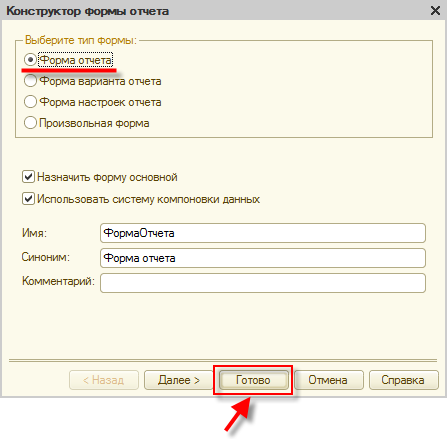


Особенность рабочего стола заключается в том, что в него можно поместить только те управляемые формы, которые в явном виде созданы в конфигурации. Создать с помощью конструктора формы списков регистра *Финансовые операции*, справочника *Друзья* и отчета *СколькоДенег*.

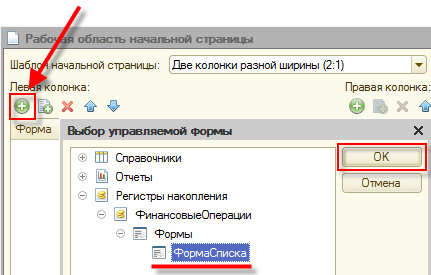
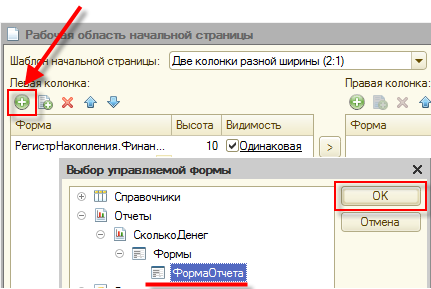
Добавить форму списка регистру и справочнику, и форму отчету.

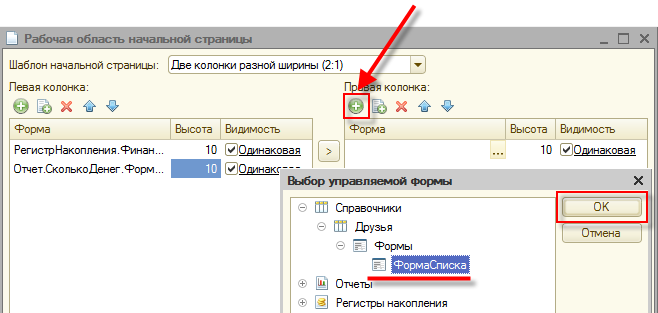
 

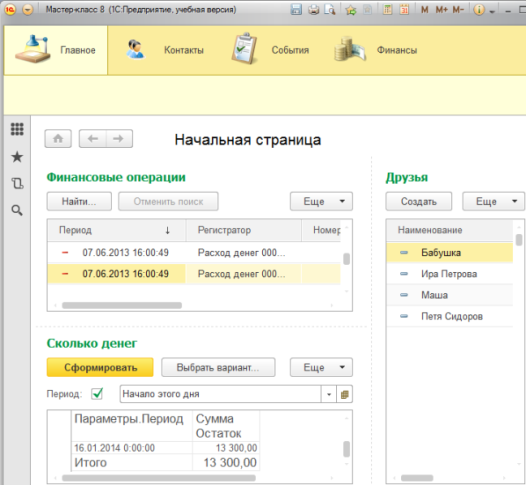
 

Настроить внешний вид рабочего стола, добавив на него только что созданные формы.



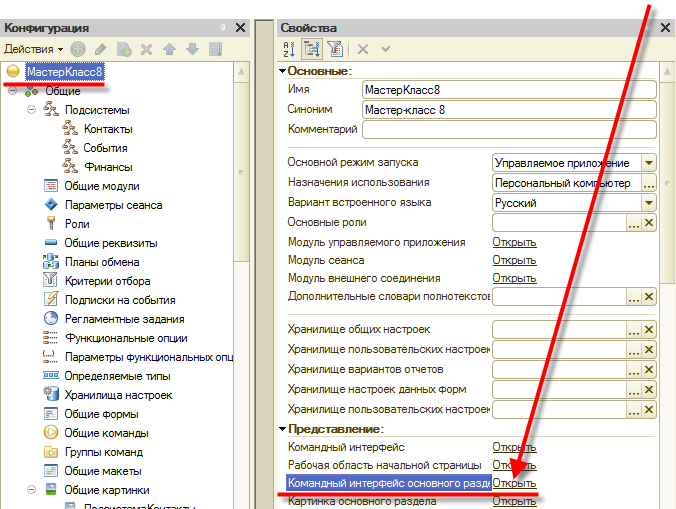
Проверить работу в пользовательском режиме.



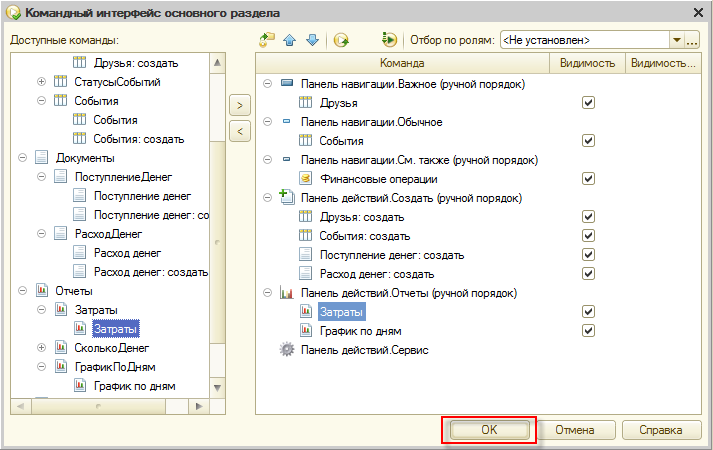
**Командный интерфейс основного раздела**

На рабочем столе доступно всего три объекта. Если необходимо больше, то для каждого объекта требуется создать формы и вытащить их на рабочий стол (что не всегда удобно), или настроить командный интерфейс рабочего стола.

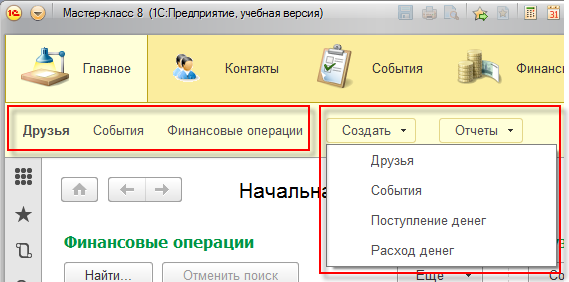
В конфигураторе открыть *Командный интерфейс основного раздела*.



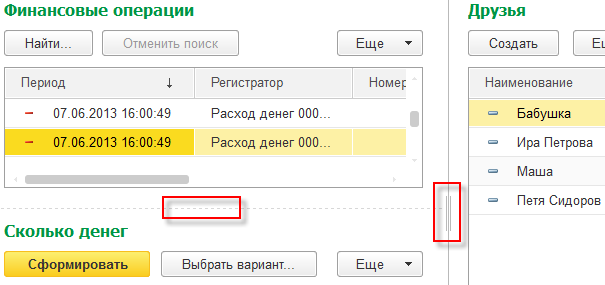
Настроить расположение объектов системы.



Проверить работу в пользовательском режиме.



Используя специальные разделители в окнах платформы «1С:Предприятие 8», можно довольно подстраивать экранные формы под себя. Например, элементы командного интерфейса можно размещать так, как удобно.

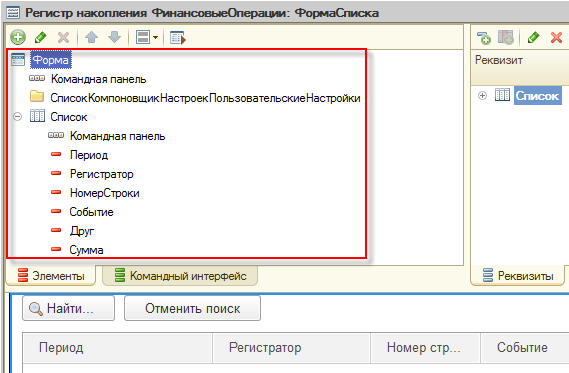


**Управляемые формы**

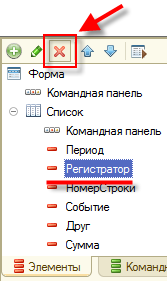
До сих пор все формы, которые генерировались автоматически – на лету (по запросу пользователя при работе в системе) либо конструктором.

В платформе «1С:Предприятие 8» нет необходимости «рисовать» форму вручную. Разработчик только описывает состав формы в виде иерархического дерева, а внешний вид получившейся формы показывается в качестве предварительного просмотра.

Система сама располагает нужные элементы в форме.

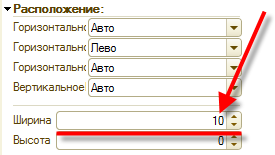
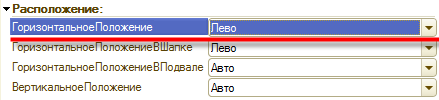


Подстроить форму регистра. Убрать реквизиты: *Регистратор*, *НомерСтроки*. После удаления ненужных элементов формы платформа «перерисует» управляемую форму так, чтобы было понятно, как будет видеть ее пользователь.

Многие свойства, влияющие на представление данных в интерфейсе, определены в свойства самих объектов конфигурации. Чтобы определить, как будет выглядеть реквизит в каждой форме достаточно один раз описать это в объекте метаданных. При показе или автоматической генерации форм, система будет использовать это описание самостоятельно, и отображать реквизит одинаково во всех формах. Если есть необходимость в какой-то одной форме изменить «стандартное» отображение – это можно сделать в самой этой форме.

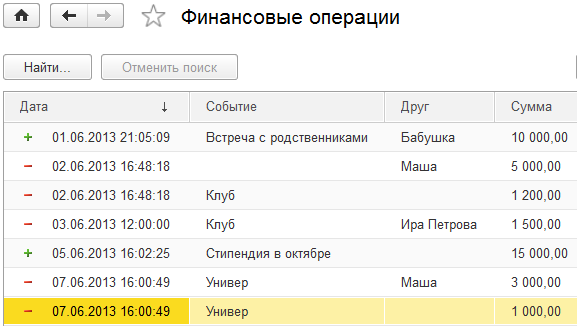
Можно задать размер и положение объекта управляемой формы, а так же его заголовок

**Р**



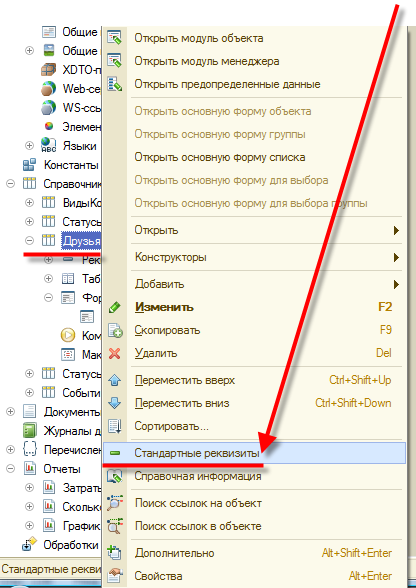
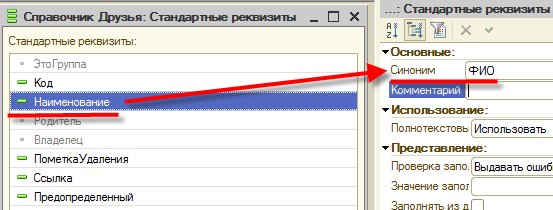
Проверить список регистра в пользовательском режиме.



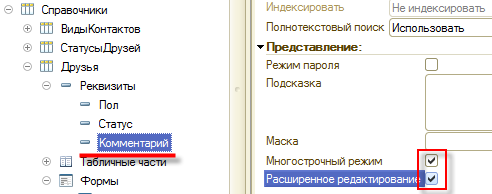
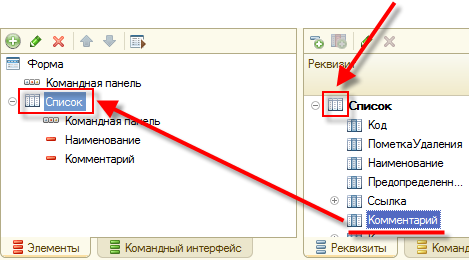
Настройка управляемой формы распространяется на все аспекты работы системы, где бы к этой форме не обращались.

Настройте форму справочника *Друзья*.

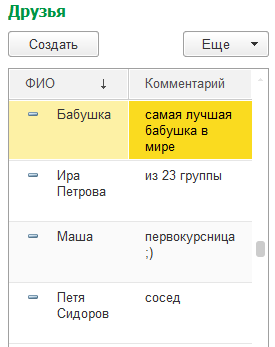
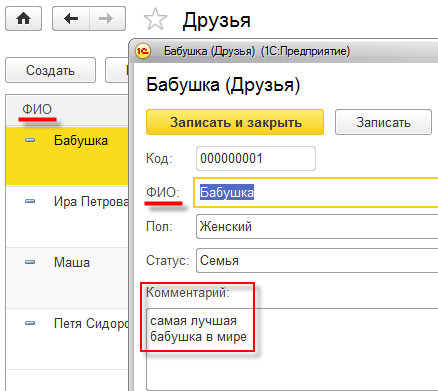
Переименовать *Наименование* в *ФИО*. У справочника *Друзья* уже существует несколько форм – формы списка и элемента справочника. Причем форма списка создана в конфигураторе, а форма элемента платформа сгенерирована на лету. Переименуйте *Наименование* в *ФИО* в одном месте – в свойствах стандартных параметров.

Включить многострочность в свойства реквизита *Комментарий* с возможностью расширенного редактирования. Добавить реквизит в управляемую форму

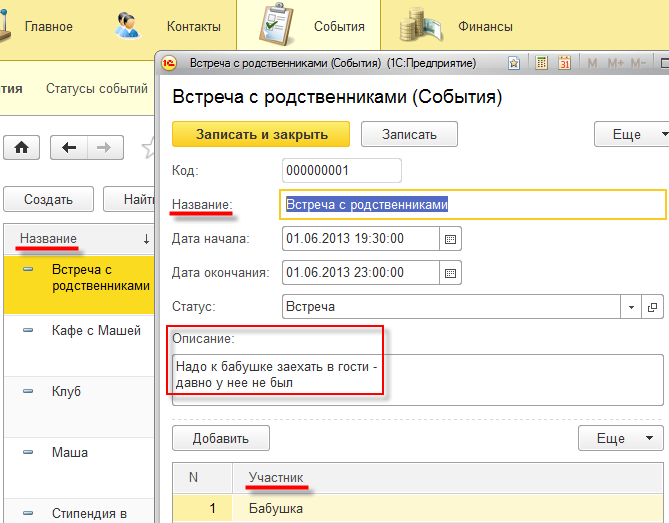
 

Получить следующие формы:

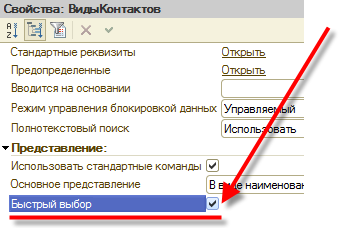
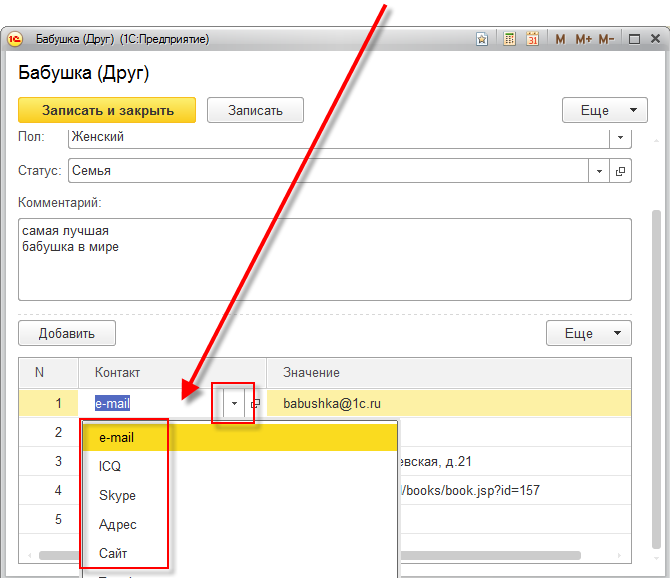
 

Настроить справочник *События*. Переименовать для пользовательского режима реквизит *Наименование* в *Название* (через *Стандартные реквизиты*), реквизит табличной части *Друг* – в *Участник* (при помощи указания синонима), а реквизит *Описание* сделать многострочным и с расширенным редактированием.

Проверить изменения.



В ряде случаев для ввода значений нет необходимости открывать справочники, достаточно использовать быстрый выбор значений, что указывается в параметрах справочника Достаточно отметить в списке объектов метаданных нужный справочник, и в палитре свойств включить опцию *Быстрый выбор*. Включить эту опцию для справочников *ВидыКонтактов*, *СтатусыДрузей* и *СтатусыСобытий*.

**Настройка отчетов**

Посмотрите отчет *ГрафикПоДням*. В целом, там все понятно, но пользователя может смущать некоторая служебная информация, форматирование дат и расшифровок.

Кроме этого, намного нагляднее видеть показатели точек графика не при наведении на них мышью, а сразу – непосредственно на диаграмме.

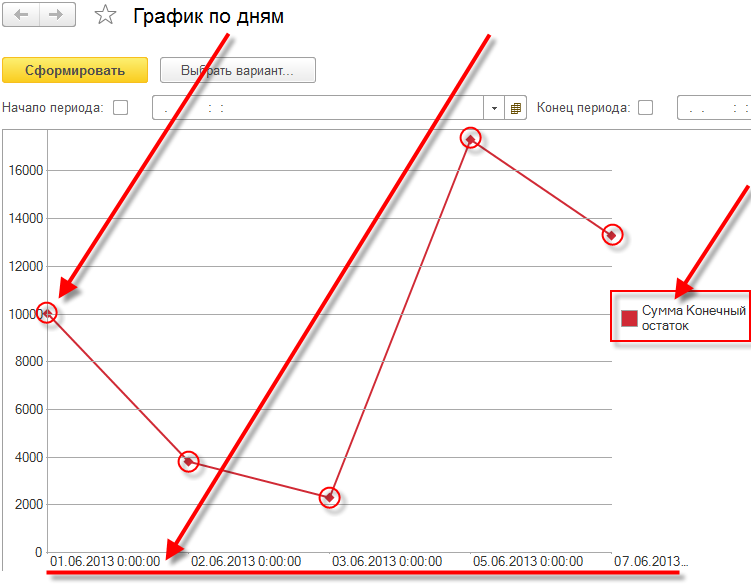
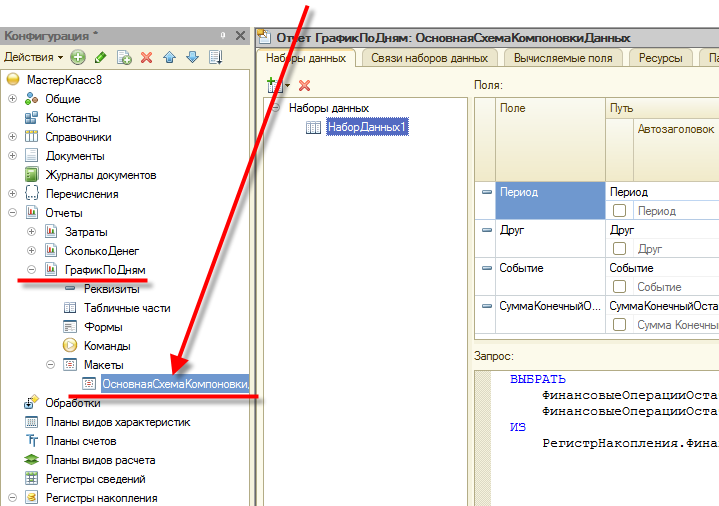
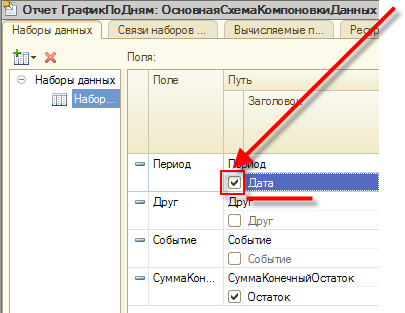
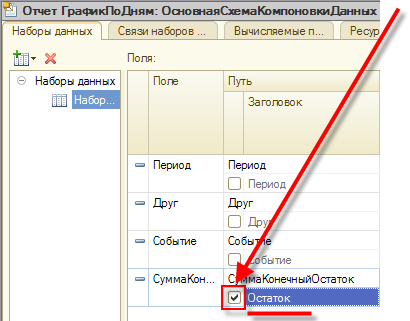


График легко привести в удобочитаемый вид. Открыть в конфигураторе схему компоновки этого отчета.

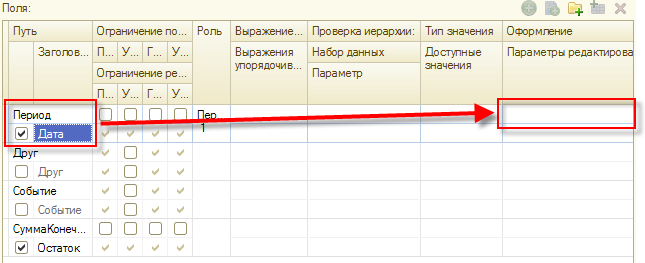


На закладке *Наборы данных* исправить подпись для полей *СуммаКонечныйОстаток* и *Период*. Кроме этого, для *Периода* задатьм более «человеческий» формат отображения дат.

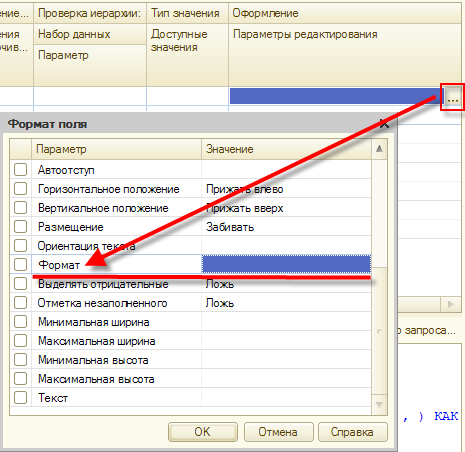
Поле *СуммаКонечныйОстаток* «переименовать» в *Остаток*. Найти поле в списке полей, включить флажок в разделе *Автозаголовок* (надпись тут же изменится на *Заголовок*) и вместо «Сумма Конечный остаток» напишем «Остаток». Тоже самое проделать с полем *Период* – «переименуем» его в *Дата*.



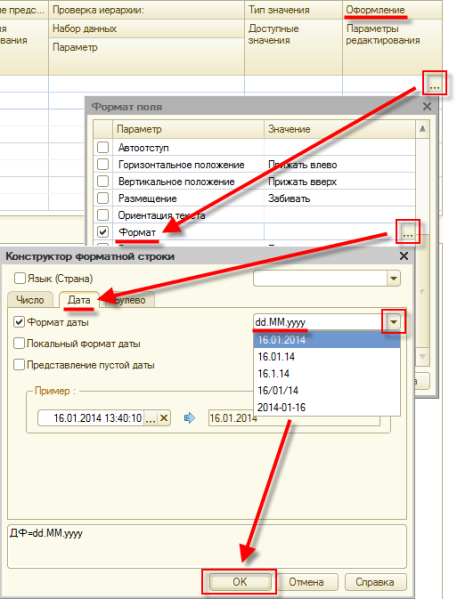
Привести в читабельный вид отображение формата дат для поля  *Период (в* параметре *Оформление)*.



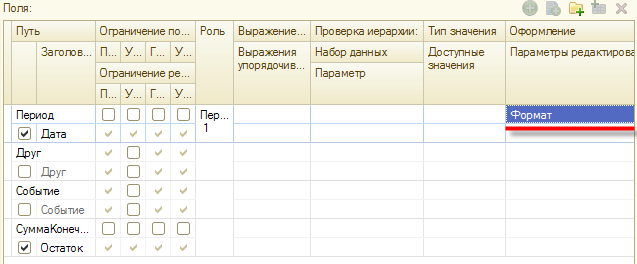
Открыть окно настроек формата поля, где найти опцию Формат.



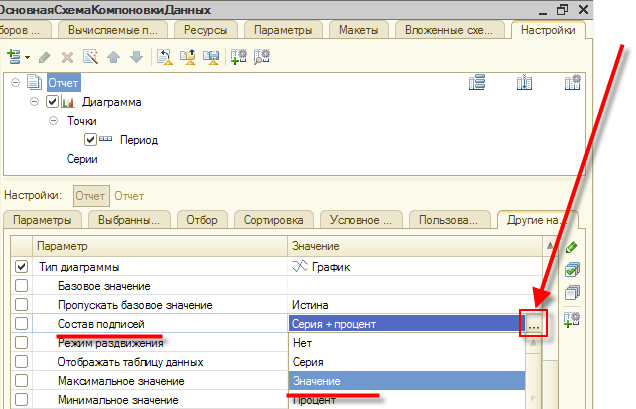
Открыть *Конструктор форматной строки*, перейти на закладку *Дата* и выбрать из списка *Формат даты* вариант «*dd.MM.yyyy*.



Далее закрываем окно настроек формата поля кнопкой *OK*.



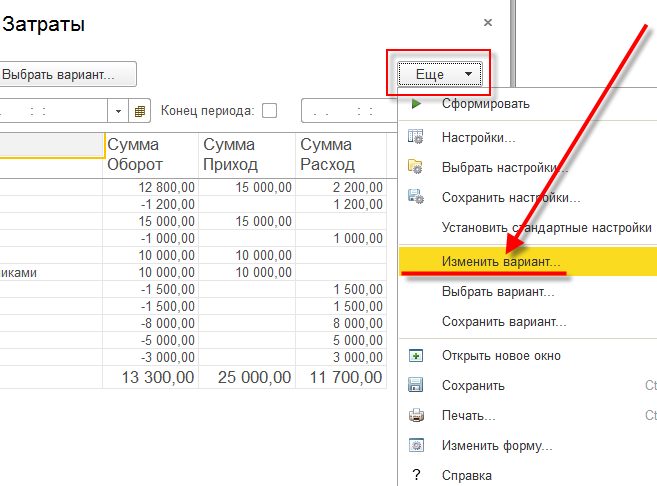
Осталось настроить вывод показателей точек на графике – перейти на закладку *Настройки*. После чего в нижней части окна активировать закладку *Другие настройки*. Найти подчиненный *Типу диаграммы* параметр *Состав подписей*, в списке которого выбрать *Значение*.



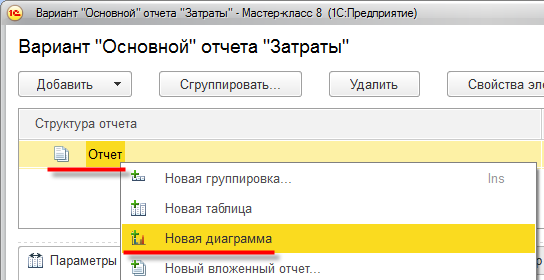
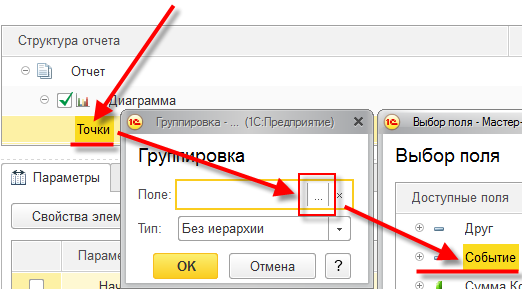
На графике представлены осмысленные подписи и организован вывод числовых показателей в точках – для более удобного анализа происходящего с нашими финансами.

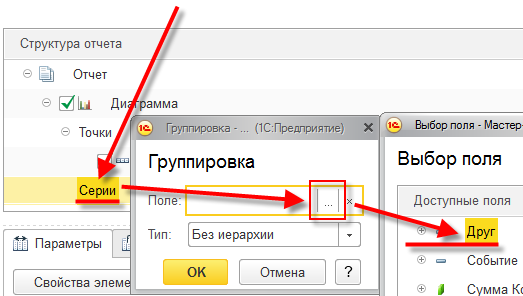
**Варианты отчетов**

Большинство настроек отчета, включая его оформление, можно изменить в пользовательском режиме, не обращаясь к разработчику системы. Измените варианты отображения отчета.

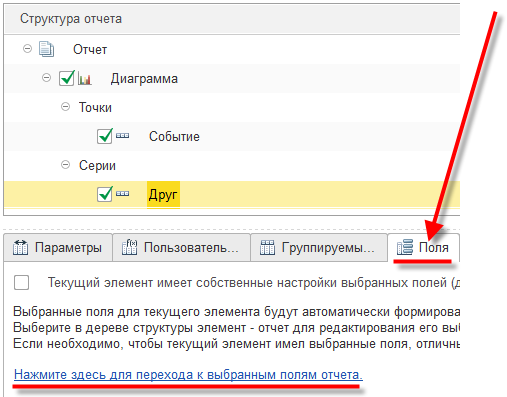
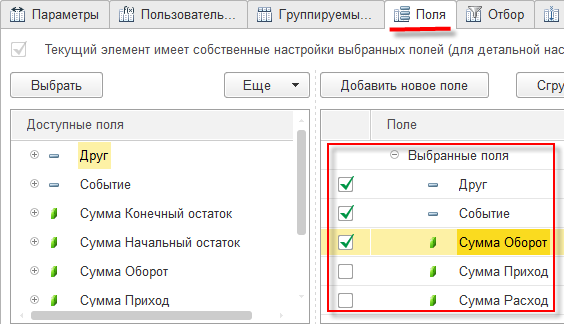
 

Создать новую диаграмму с группировкой по событию. Для *Серий* выбрать поле группировки – *Друг*.

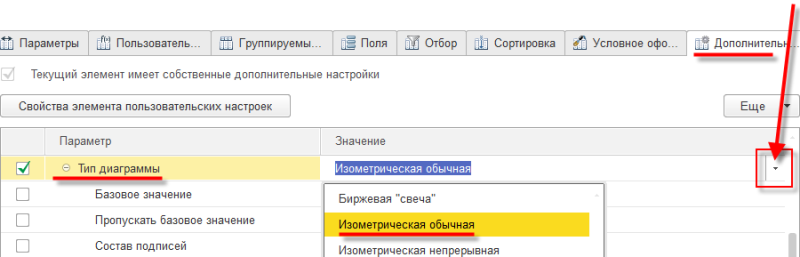
 

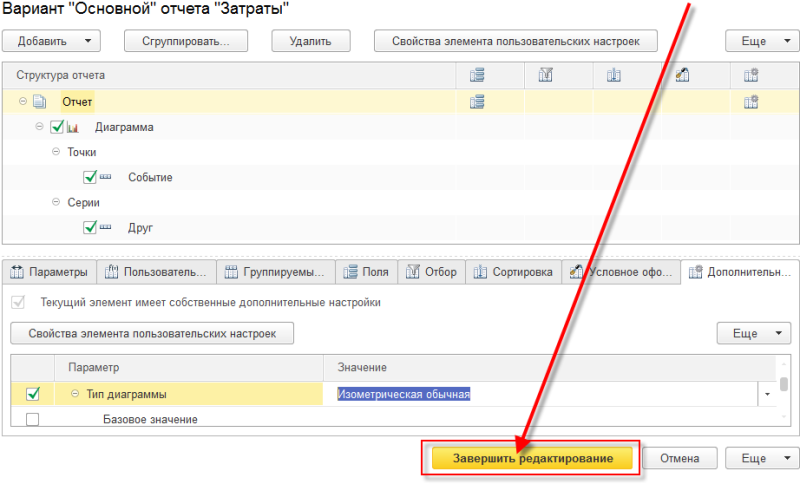


Проверить, чтобы поле *Сумма оборот* было первым после полей *Друг* и *Событие*. А у полей *Сумма Приход* и *Сумма Расход* были сняты флажки.

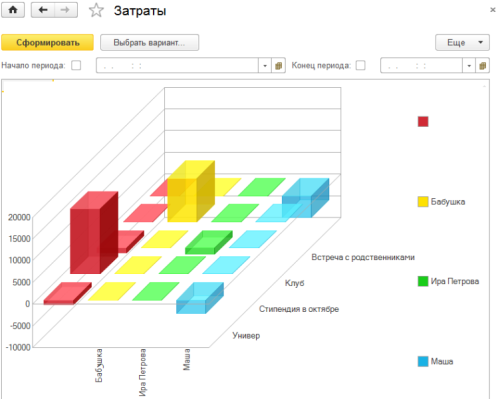
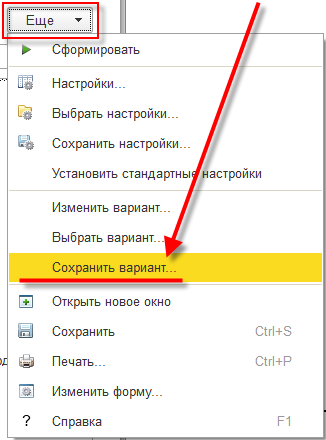
 

На вкладку *Дополнительные* настройки и выбирать тип диаграммы – *Изометрическая обычная*.

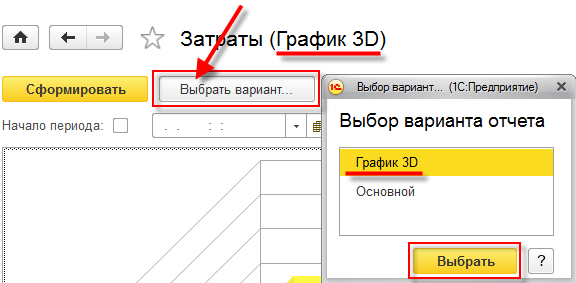




Проверь работу в пользовательском режиме

Платформа предоставляет удобный сервис сохранения и восстановления вариантов отображения отчетов. Достаточно нажать кнопку *Еще* и выбрать *Сохранить вариант*…В будущем можно использовать его столько раз, сколько потребуется, используя кнопку *Выбрать вариант…*



**Функциональные опции**

Функциональные опции позволяют выделить некоторую часть функциональности прикладного решения и быстро и просто управлять тем, будет эта часть использоваться, или не будет использоваться в конкретной информационной базе.

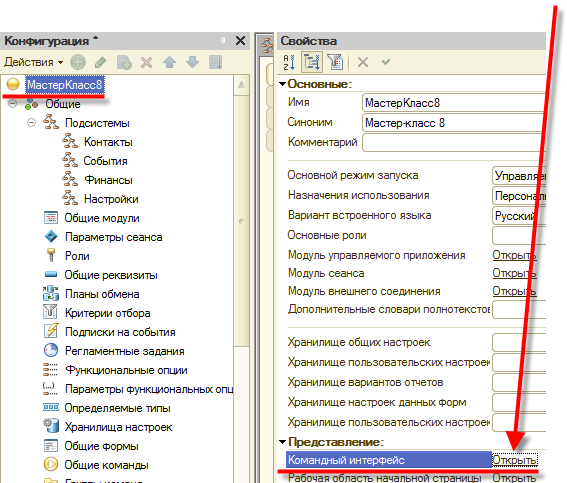
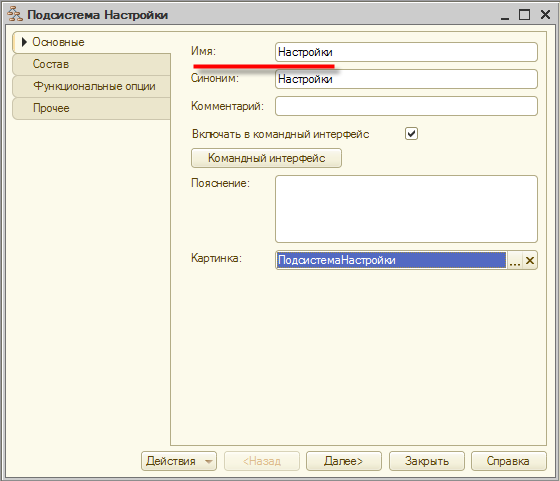
Использовать функциональные опции удобно при внедрении программ. Например, в нашей программе предусмотрено три «учетных» раздела – работа со знакомыми, событиями и финансовый учет. Если мы решим подарить нашу разработку друзьям или доработать ее функционально и продавать, всегда можно поинтересоваться, а все ли нужно потенциальным пользователям из того, что мы предлагаем. Например, далеко не все готовы набирать и отслеживать все события, которые происходят в их жизни.

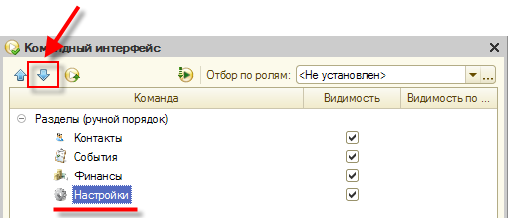
Можем выделить раздел в функциональную опцию и предоставить пользователям самим решать, нужно им это или нет. При этом платформа «1С:Предприятие 8» отследит все упоминания событий во всех разделах конфигурации и, если это пользователю не нужно, автоматически отключит их использование и отображение.

Причем все эти изменения платформа выполнит самостоятельно, так что специалисту нет необходимости как-либо изменять конфигурацию (что-либо программировать). Он просто в пользовательском режиме *1С:Предприятие* задает нужное значение функциональной опции.

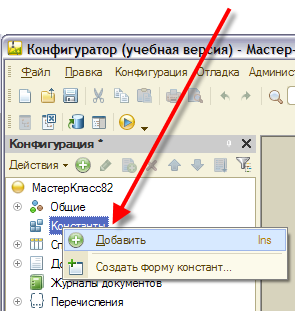
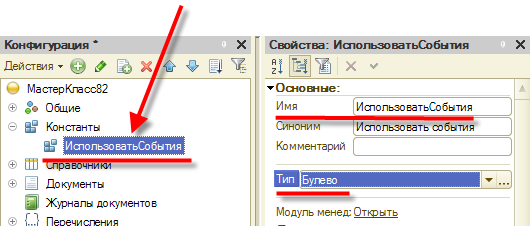
Рассмотрим простой пример – когда значение функциональной опции хранится в константе, имеющей тип *Булево*. Если там значение *Истина* – значит, функциональная опция включена. Если значение *Ложь* – функциональная опция выключена.

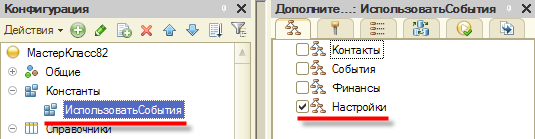
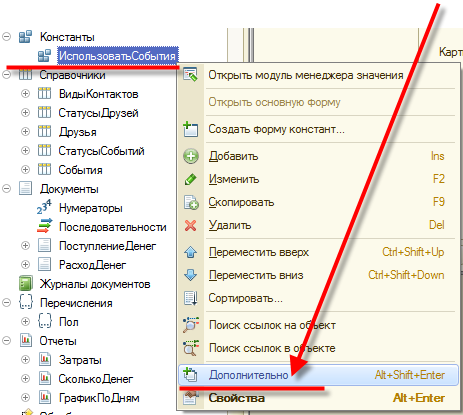
Создать подсистему Настройки. Расположить новую «настроечную» подсистему в конце «обычных» подсистем.



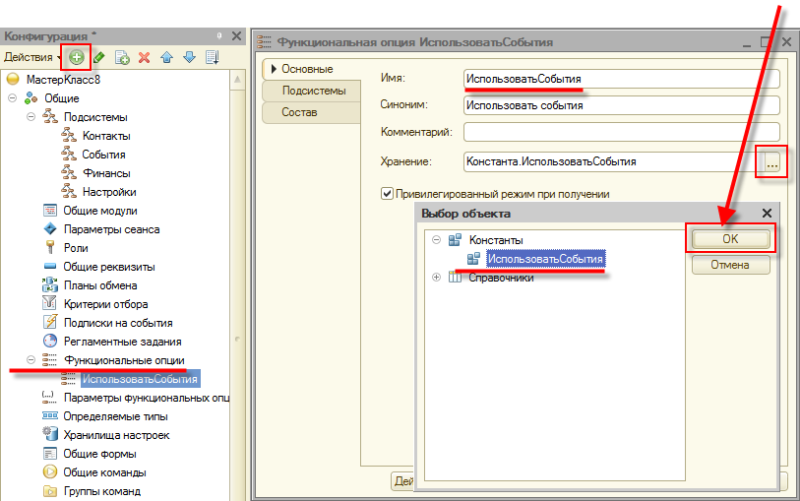


Создать константу  *ИспользоватьСобытия*, она имеет тип *Булево* и привязать ее к подсистеме *Настройки* (меню *Дополнительно Alt+Shift+Enter*).



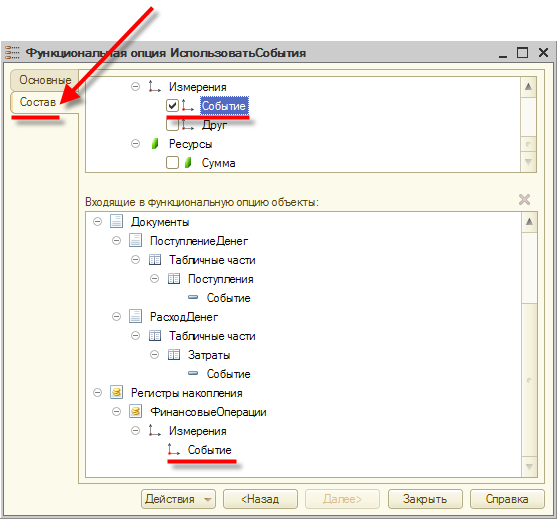
Создать функциональную опцию *ИспользоватьСобытия* и указать, что ее значение будет храниться в константе *ИспользоватьСобытия*.



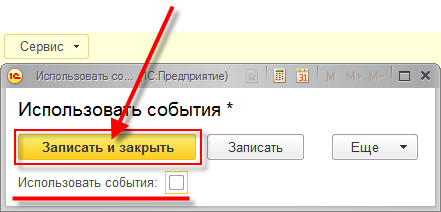
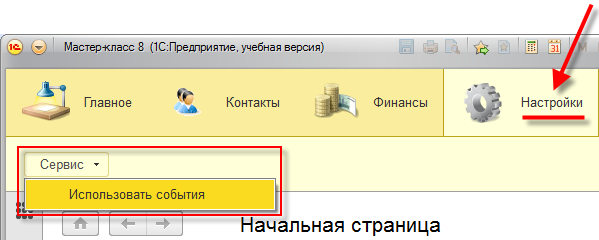
В результате этих действий в разделе *Настройки* появится команда, которая позволит в пользовательском режиме открыть форму констант и изменить включить или отключить использование функциональной опции.

Привязать объекты конфигурации и реквизиты объектов к этой функциональной опции. Функциональность работы с событиями обеспечивает: это подсистема *События*; справочник *События* и связанный с ним справочник *СтатусыСобытий*; реквизиты табличных частей документов *Событие*; и, измерение *Событие* регистра накопления.

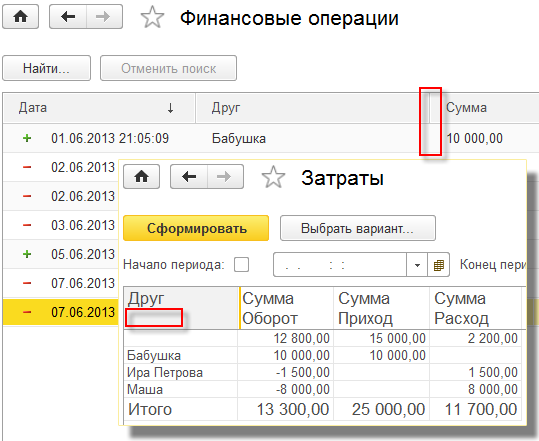
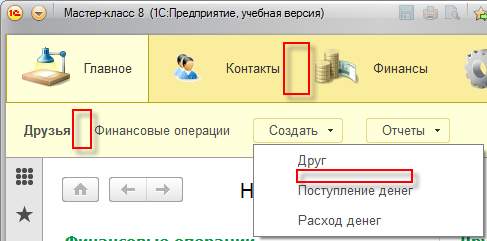
Привязать все перечисленные выше элементы конфигурации к функциональной опции *ИспользоватьСобытия*. В окне параметров функциональной опции перейти на закладку *Состав*. Последовательно отметить флажки у всех объектов, относящихся к функциональной опции. Для контроля система будет выводить в нижнем окне список того, что отмечено.



Проверить работу в пользовательском режиме.



Во-первых, «пропал» раздел *Событи*я и упоминания объектов, связанных с событиями на рабочем столе. Во-вторых, исчезли все упоминания событий во всех используемых в системе объектах, включая информацию, отображаемую отчетами.



**.**

Лабораторная работа 3

Создать собственную учетную систему. Она должна содержать несколько справочников, приходные и расходные документы, отчеты. Настроить пользовательский интерфейс системы.

Примеры:

- «Домашняя библиотека» (учет книг)

- «Техносервис» (ремонт и модернизация компьютеров)

- «Общественное питание» (производство блюд)

- «Студенческая кадровая служба» (учет вакансий по различным специальностям)

Лабораторная работа 4

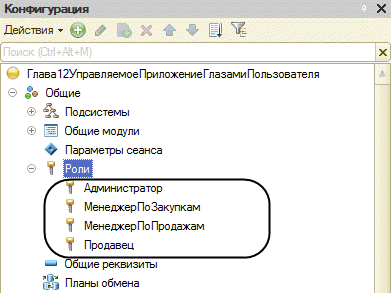
**Система прав доступа**

Основным механизмом настройки доступности команд в командном интерфейсе является *система прав доступа*.

Операция назначения роли пользователю решает две основные задачи:

* Во-первых, ограничивает состав пользователей, имеющих доступ к конфиденциальной информации.
* Во-вторых, предотвращает возможные потери информации путем запрета выполнения пользователем определенных операций (в первую очередь операций удаления и корректировки данных).

При проектировании прикладного решения разработчик должен выделить группы пользователей, выполняющих одинаковые и/или логически связанные операции. Для каждой из выделенных групп разработчик добавляет собственную роль.

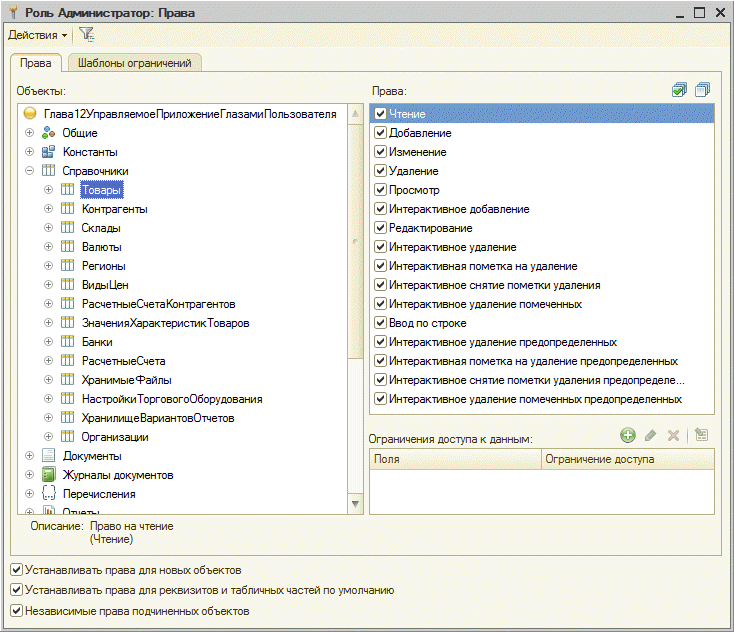


Каждая из ролей определяет независимый (от других ролей) набор прав доступа к обрабатываемой в прикладном решении информации и действий, которые пользователь может выполнять с этой информацией.

Разработчик имеет возможность создать любое необходимое количество ролей, обеспечивающих предоставление разным пользователям различных полномочий.

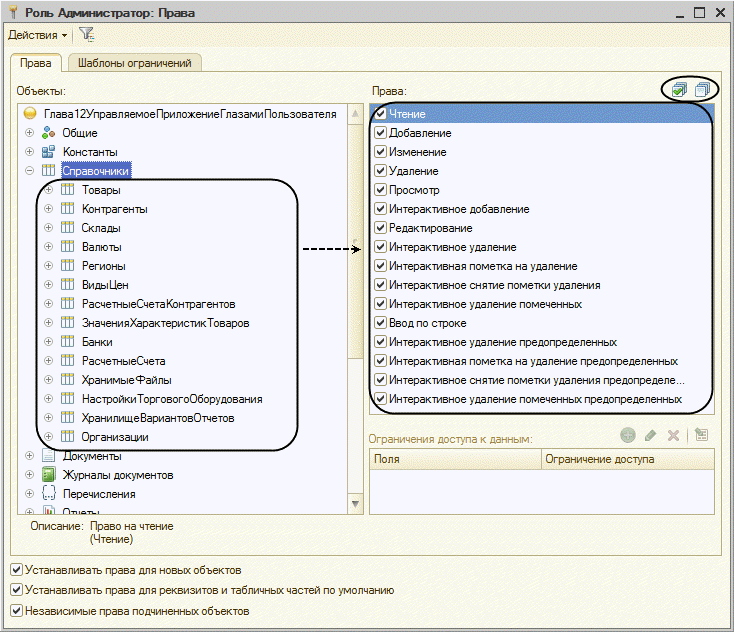
Используя механизмы управления пользователями, администратор системы регистрирует пользователей прикладного решения. Каждому пользователю назначается роль (или несколько ролей), которая определяет права пользователя на доступ к данным.

Настройка прав выполняется в окне редактирования роли, которое открывается двойным щелчком мыши на выбранной роли.



Для настройки прав необходимо в дереве конфигурации выбрать объект (левое поле) и в списке прав (правое поле) установить или снять отметку рядом с настраиваемым правом. С помощью кнопок в правом верхнем углу над списком прав можно установить или снять сразу все права

Права можно установить на уровне классов объектов конфигурации. Для этого в дереве конфигурации необходимо выбрать узел, представляющий соответствующий класс.



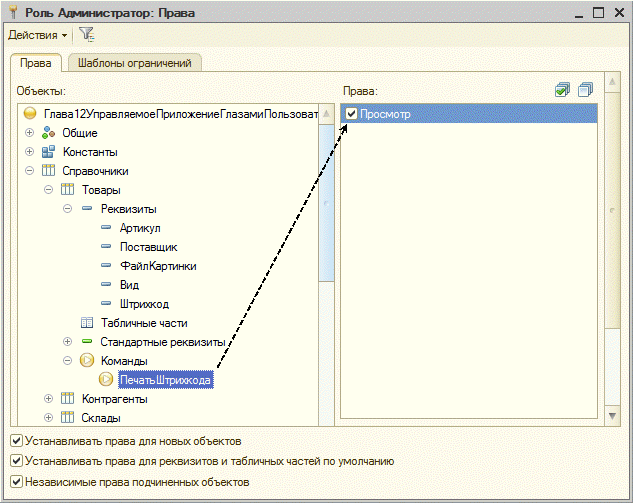
В этом случае назначаются одинаковые права всем объектам конфигурации этого класса. Например, полные права будут установлены сразу для всех справочников.

Права могут быть установлены и на отдельный объект конфигурации



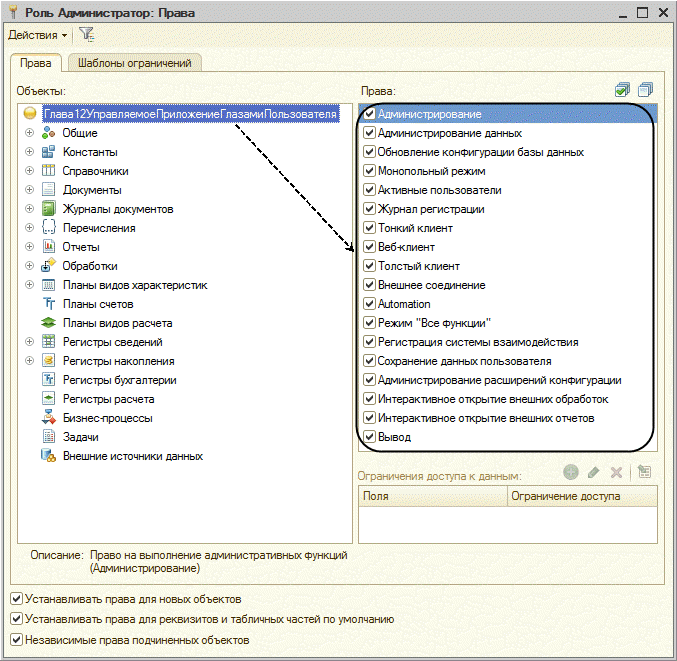
В этом случае права применяются к объектам данных информационной базы, которые описаны выбранным объектом конфигурации.

Права также могут быть установлены на объекты, подчиненные выбранному (как в целом для всех подчиненных объектов одного типа, так и индивидуально на каждый из подчиненных объектов) – реквизиты, табличные части и команды .



В этом случае права применяются только к командам, подчиненным объекту, или к данным, хранящимся в подчиненных информационных структурах.

Кроме прав, применяемых для объектов конфигурации, существует набор прав, применяемый к конфигурации в целом. Список этих прав отображается при выборе корневого узла дерева конфигурации



Эти права определяют возможность выполнения административных функций, использования внешних отчетов и обработок, доступа к журналу регистрации, сохранения данных и настроек пользователя и др.

Задание

Создать не менее 2х ролей в разрабатываемой вами конфигурации.

Для добавления роли используем команду *Добавить* контекстного меню узла *Роли* дерева конфигурации. В результате будут добавлены объекты конфигурации и откроется палитра свойств новой роли. Для каждой из ролей установить пава доступа к подсистемам и справочникам.

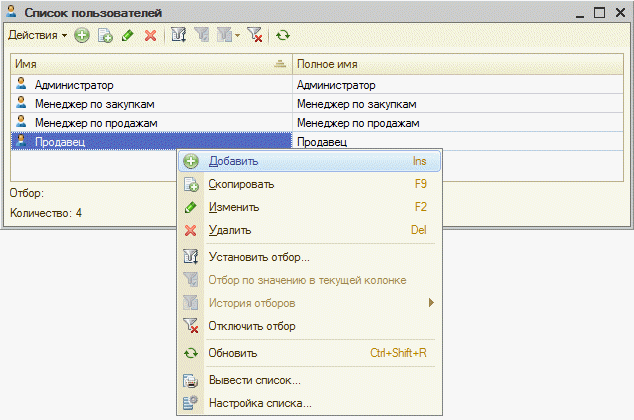
ПРИМЕЧАНИЕ

*При установке интерактивного права Просмотр автоматически устанавливается основное право Чтение.*

*Перед созданием пользователя необходимо обновить конфигурацию базы данных. Это можно сделать, выбрав команду Конфигурация – Обновить конфигурацию базы данных.*

Вы определили набор прав для пользователей. Теперь необходимо создать пользователей и назначить им вновь созданную роль. Для регистрации пользователей и назначения для них ролей используется система управления пользователями.

Командой Администрирование – Пользователи  открывается список пользователей и можно воспользоваться командой Добавить контекстного меню этого списка. В открывшемся окне определим значения свойств учетной записи добавляемого пользователя.



ПРИМЕЧАНИЕ

*Права складываются по правилам логического ИЛИ – если хотя бы в одной из доступных ролей право на выполнение операции установлено, то пользователь может выполнить операцию*.

Сохранить сведения о пользователе, нажав кнопку OK, обновить конфигурацию базы данных и запустить приложение от имени созданного пользователя и проверьте работу.

**РАБОТА С ФОРМАМИ**

Можно отметить следующие ключевые особенности форм.

Во-первых, форма не прорисовывается детально разработчиком, а строится системой автоматически. Разработчик же в режиме конфигурирования:

* определяет состав формы в виде дерева элементов;
* описывает поведение формы, задавая значения для ее свойств и/или реализуя процедуры на встроенном языке.

При построении интерфейса для того или иного пользователя система использует это декларативное описание для создания формы и размещения ее элементов.

Во-вторых, при создании формы используется модель управления доступностью и видимостью элементов формы. При этом учитываются:

* настройки прав в разрезе ролей пользователей;
* зависимость элементов формы от функциональных опций;
* настройка формы, выполненная разработчиком на этапе конфигурирования прикладного решения;
* настройка формы, выполненная пользователем на этапе эксплуатации прикладного решения

Форма является важнейшим связующим звеном в цепи «пользователь – данные». Именно в формах мы редактируем данные, вводим новую информацию, видим результаты работы.

Формы «1С:Предприятия» наделены различными возможностями как построения, так и отображения данных. В большинстве случаев от разработчика не требуется заботиться о том, как на экране будет выглядеть тот или иной элемент, система сама сможет позаботиться об этом. От разработчика требуется правильно настроить интерфейсные свойства объекта конфигурации, на основании которых и будет построено нужное представление.

Управляя размещением элементов в форме, разработчик должен «посоветовать» системе те или иные способы группировки элементов, порядок их размещения. Естественно, платформа предоставляет возможности взять под контроль некоторые этапы разработки формы, но данное действие не является приоритетным. Задача разработчика состоит не в детальном, «попиксельном», размещении элементов на форме, не в описании сложных привязок, а в логическом описании состава формы.

Необходимо описать состав формы в виде дерева элементов, которые будут отображать данные, добавить в описание необходимые реквизиты и команды. Скомпоновать элементы в логические группы, определить порядок обхода элементов формы.

Полученное от разработчика описание формы, другие факторы, влияющие на внешний вид и функциональность формы, помогут системе построить форму на экране и тем самым освободить ресурсы разработчика не для «рисования», а для разработки функциональности конкретного прикладного решения.

Разработчик может влиять на расположение и внешний вид элементов формы. Для этого у него в руках инструмент свойств элементов формы. Существует возможность перенастройки командных панелей формы, дополнительных кнопок у элементов формы, объединения элементов в группы, распределения их по страницам, настройки колонок списков. Однако все эти возможности призваны лишь помочь системе в построении формы, а не полностью заменить это построение ручным способом.

Помимо простого открытия форм для просмотра или редактирования данных существует возможность открытия с установленным отбором, с выделением каких-либо конкретных данных из общего числа. Помочь в этом могут, например, программная установка отборов и параметризуемые команды.

При построении форм системой учитываются не только настройки самой формы, сделанные разработчиком. Влияние на поведение формы и ее элементов оказывают настройки прав пользователей, применяемые к сеансу работы приложения функциональные опции, настройки, которые сделал сам пользователь в сеансе своей работы.

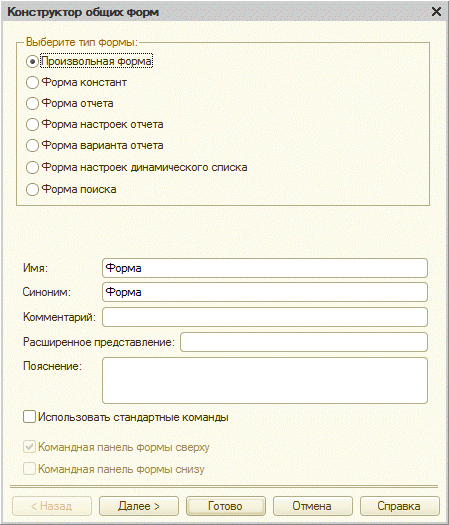
Отдельно стоит упомянуть о возможностях форм сообщать пользователю об ошибках, возникающих в процессе работы. Например, это может быть сообщение о незаполненном поле, данные в которое должны быть внесены обязательно. Система сообщит о таком типе ошибки, выделив и активизировав именно этот элемент.

Создание новой формы начинается с конструктора форм..

В конфигурации могут существовать формы, подчиненные объектам конфигурации, и общие формы. Общие формы располагаются в дереве конфигурации, в ветке Общие – Общие формы. Конструктор форм различает, какая форма будет создаваться: подчиненная объекту или общая.

При создании общей формы конструктор предлагает выбрать тип формы из следующих типов:

* Произвольная форма;
* Форма констант;
* Форма отчета;
* Форма настроек отчета;
* Форма варианта отчета;
* Форма настроек динамического списка;
* Форма поиска.



Выбор типа общей формы

При выборе произвольной формы разработчик самостоятельно в процессе разработки будет определять данные, с которыми будет взаимодействовать форма.

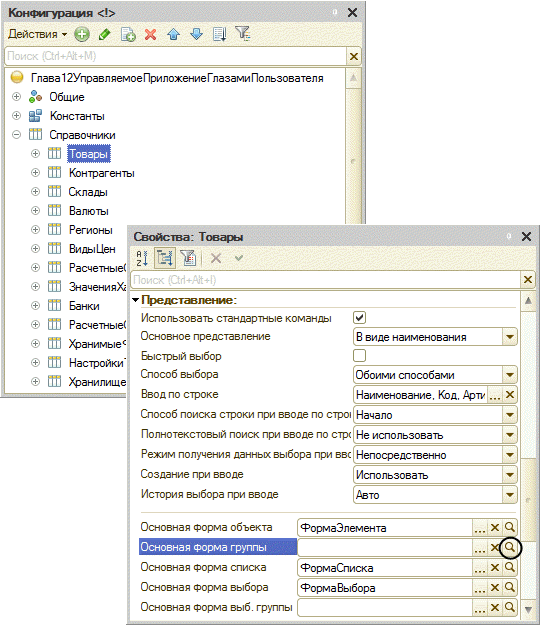
При выборе остальных типов форм разработчик указывает тип данных, с которыми в дальнейшем будет взаимодействовать форма. В зависимости от этого конструктор автоматически добавляет в форму соответствующие данные и интерфейсные элементы, сокращая дальнейшие действия разработчика.

По умолчанию конструктор общих форм предлагает создать произвольную форму. Связано это с тем, что в подавляющем большинстве случаев разработчик будет создавать именно произвольные общие формы.

Флажок Использовать стандартные команды в конструкторе общих форм позволяет автоматически добавить команды для открытия общей формы в интерфейс той подсистемы, в состав которой она будет входить.

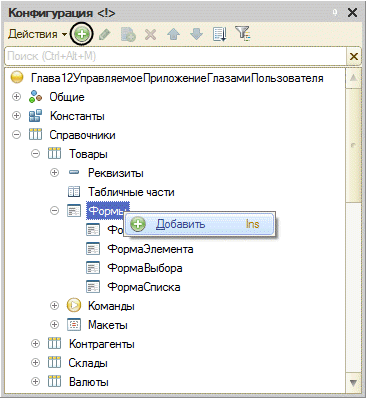
Существует несколько способов создать форму, подчиненную какому-либо объекту конфигурации:

Воспользоваться кнопкой Открыть в палитре свойств объекта конфигурации у нужного типа формы.



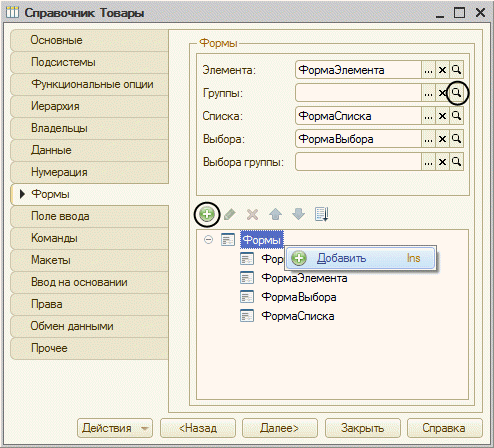
Создание формы кнопкой «Открыть»

Выделить в дереве конфигурации ветвь Формы объекта, для которого будет создаваться форма, и воспользоваться либо контекстным меню, либо кнопкой Добавить на панели дерева конфигурации .



Создание формы кнопкой «Добавить»

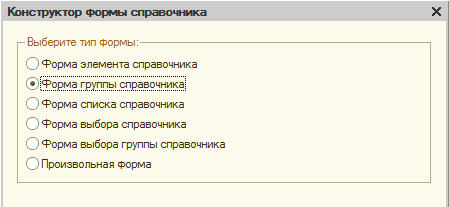
Открыть окно редактирования объекта конфигурации, перейти на закладку Формы и воспользоваться либо кнопкой Открыть у необходимого типа формы, либо кнопкой командной панели Добавить, либо контекстным меню в списке форм.



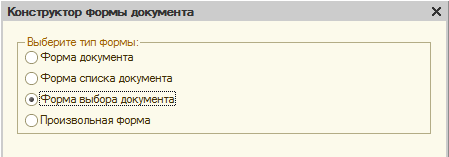
Создание формы из окна редактирования объекта

При использовании любого из вышеперечисленных способов будет открыто окно конструктора формы объекта конфигурации. В зависимости от вида объекта, его свойств конструктором будут предложены характерные для текущего режима создания типы форм:

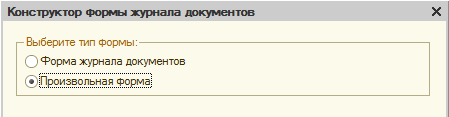
Для объекта Справочник это будет форма списка, форма выбора, форма элемента. Если справочник иерархический, то возможно создание формы группы и формы выбора группы.



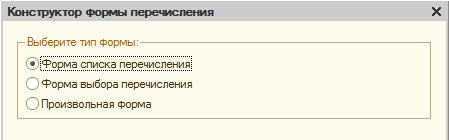
Для объекта Документ – это форма списка, форма выбора, форма документа.



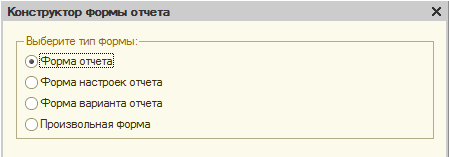
Для объекта Журнал документов – это форма журнала



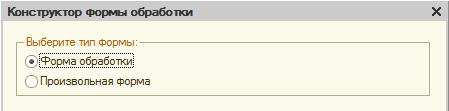
Для объекта Перечисление – это форма списка и форма выбора



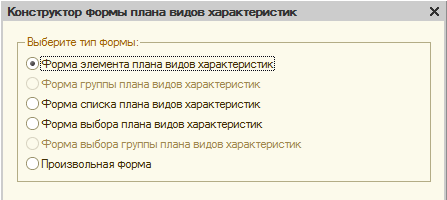
Для объекта Отчет – это форма отчета, форма варианта, форма настроек



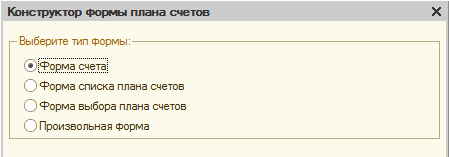
Для объекта Обработка это форма обработки



Для объекта План видов характеристик – это форма списка, форма выбора, форма элемента. Если план видов характеристик иерархический, то возможно создание формы группы и форма выбора группы. Если объект неиерархический, то создание форм группы недоступно.



Для объекта План счетов – это форма списка, форма выбора, форма счета (рис. 2.14).



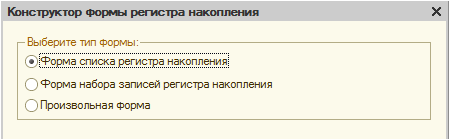
Для объекта План видов расчета – это форма списка, форма выбора, форма вида (рис. 2.15).



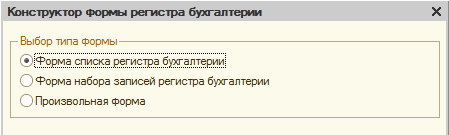
Для объекта Регистр сведений – это форма списка, форма записи, форма набора записей.



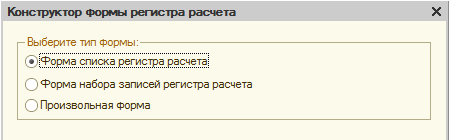
Для объекта Регистр накопления – это форма списка и форма набора записей.



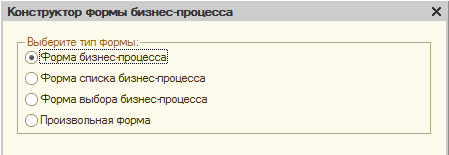
Для объекта Регистр бухгалтерии – это форма списка и форма набора записей



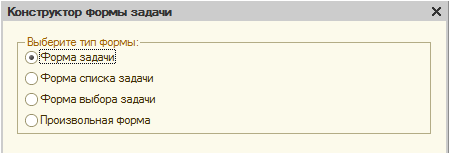
Для объекта Регистр расчета – это форма списка и форма набора записей



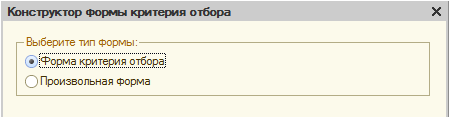
Для объекта Бизнес-процесс – это форма списка, форма выбора, форма бизнес-процесса



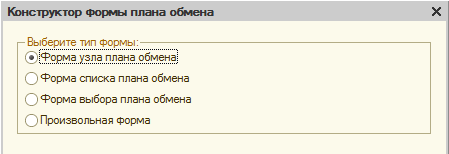
Для объекта Задача – это форма списка, форма выбора, форма задачи



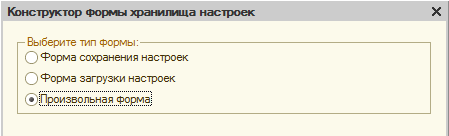
Для объекта Критерий отбора – это форма критерия отбора (рис. 2.22).



Для объекта План обмена – это форма списка, форма выбора и форма узла



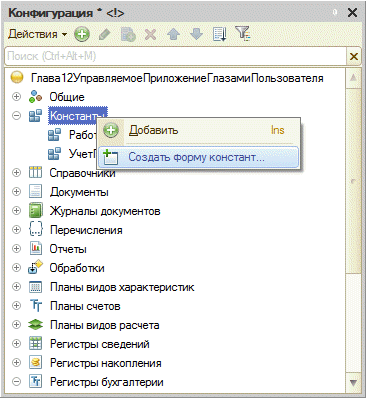
Для объекта Хранилище настроек – это форма сохранения и форма загрузки настроек



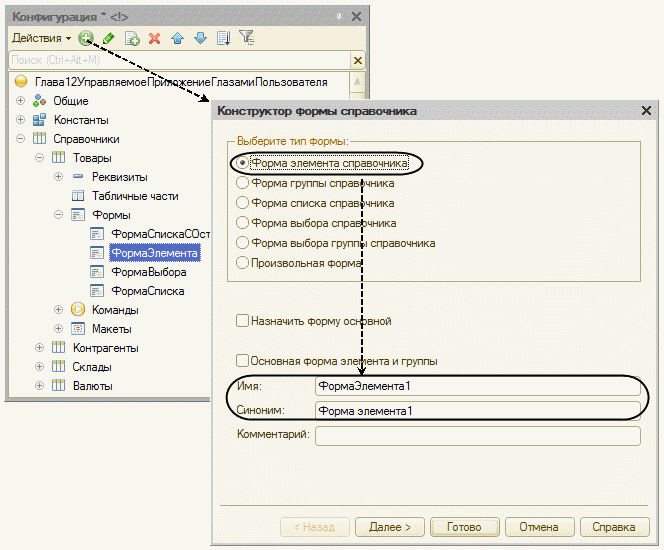
Кроме всех перечисленных форм для любого из объектов конфигурации можно создать тип формы Произвольная. Такая форма после создания не связана ни с какими данными конфигурации. Действия, которые можно будет выполнять в такой форме, целиком и полностью будут зависеть от желания и способностей разработчика формы.

Формы, связанные через свой основной реквизит с объектами конфигурации, уже наделены определенными свойствами, характеристиками, методами. Состав таких свойств, характеристик, методов зависит от объекта конфигурации.

Например, форму редактирования констант можно создать с помощью контекстного меню ветви конфигурации Константы Созданная таким образом форма будет размещаться в ветви конфигурации Общие формы.



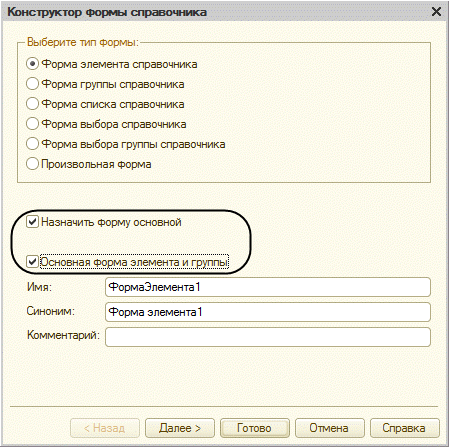
При выборе того или иного типа создаваемой формы конструктор форм автоматически формирует Имя и Синоним формы. При формировании учитывается наличие у объекта конфигурации формы с таким именем. Если такая форма существует, то к имени создаваемой формы добавится числовой показатель



Если разработчика не устраивает имя формы, которое предлагает конструктор, то его можно изменить. При этом возможно автоматическое формирование и нового синонима формы. Синоним формы – имя, под которым форма может фигурировать в интерфейсе пользователей прикладного решения. Синоним формы также можно изменить.

При необходимости можно определить создаваемую форму в качестве основной формы для объекта конфигурации. Основная форма – это та форма, которая будет открываться стандартными командами «1С:Предприятия», размещенными в интерфейсе. Если разработчик хочет открыть форму, не назначенную основной, то для этого ему нужно создать собственную команду, которая будет открывать эту форму.

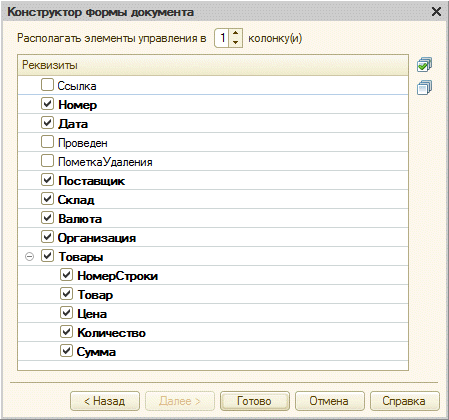
Чтобы назначить форму основной, в конструкторе форм необходимо установить соответствующий флажок. Если для объекта конфигурации создаваемый тип форм еще не создавался (нет формы создаваемого типа), то флажок Назначить форму основной устанавливается автоматически.



В дальнейшем основную форму объекта можно переопределить. Как это сделать, будет рассказано в главе, посвященной редактору формы.

Для некоторых типов форм конструктор предоставляет возможность использовать их одновременно в качестве формы элемента и формы группы. Это удобно в тех случаях, когда и элемент, и группа имеют одинаковый состав реквизитов и одинаковые алгоритмы их заполнения; не нужно создавать две разные формы.

На этом работу с конструктором форм можно закончить (кнопка Готово) или перейти к следующему шагу (кнопка Далее), который позволит продолжить конструирование формы. На этом шаге разработчик дает указания конструктору форм на необходимость использования реквизитов объекта (или состава констант для формы констант) в создаваемой форме. Делается это с помощью установки соответствующего флажка. Здесь же можно определить, сколько колонок с элементами формы будет применяться для отображения данных



ВНИМАНИЕ!

Управлять порядком следования элементов разрабатываемой формы в данном окне конструктора формы разработчик не может. Это можно сделать либо предварительно (определив порядок реквизитов объекта), либо после окончания работы конструктора (в окне редактора формы).

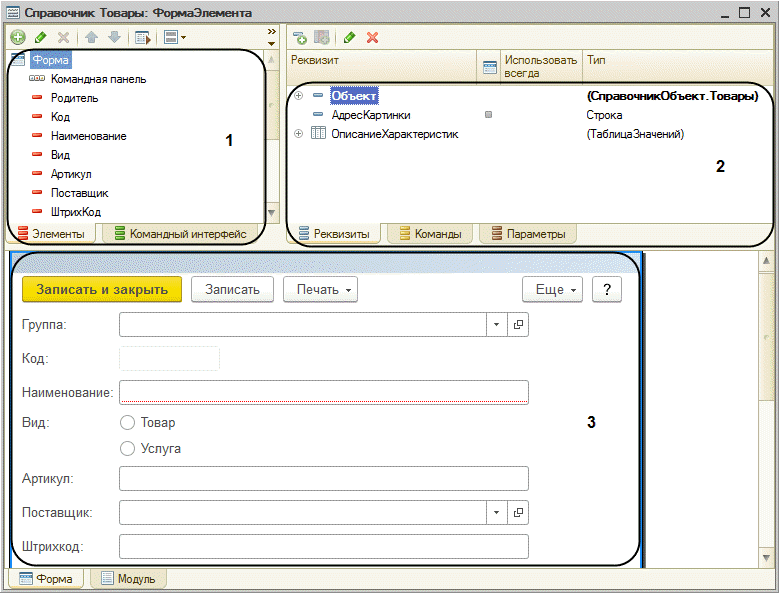
ПРИМЕЧАНИЕ

*Все настройки, сделанные на втором шаге в конструкторе форм, можно потом изменить, поэтому сильно задумываться, например, о количестве колонок элементов, если вы еще не представляете в голове внешнего вида формы, не следует.*

*Отказаться от создания новой формы можно на любом этапе работы конструктора форм. Для этого необходимо или закрыть окно стандартным для всех окон операционной системы способом, или нажать кнопку Отмена.*

Описание редактора формы

Для редактирования формы в конфигураторе используется специализированный редактор форм. Окно редактора форм разбито на несколько областей, каждая из которых отвечает за ту или иную функциональность будущей формы.



1 Описываются состав и порядок элементов, из которых состоит форма, команды интерфейса, которые могут выполняться в форме.

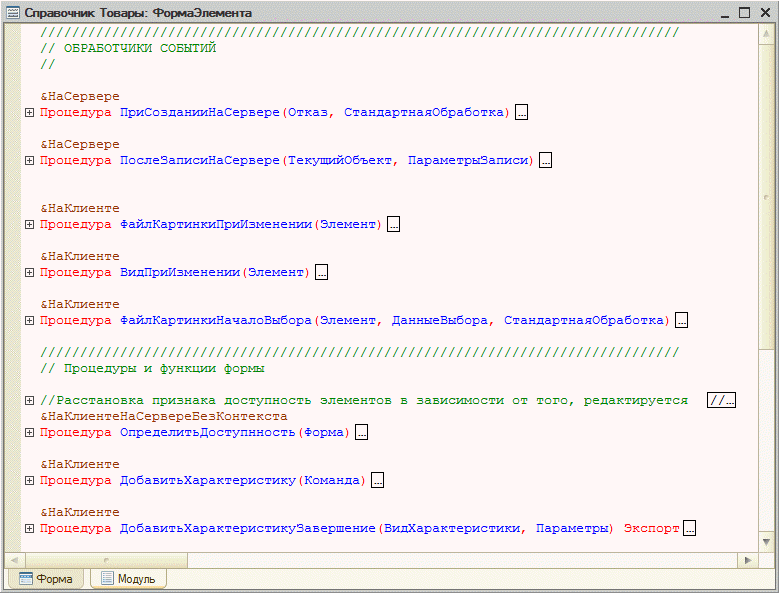
2 Описывается состав реквизитов и параметров формы, а также команды, выполняемые внутри формы.

3 Представлен внешний вид формы, как она может выглядеть на экране пользователя, с учетом описанных реквизитов, элементов, команд формы.

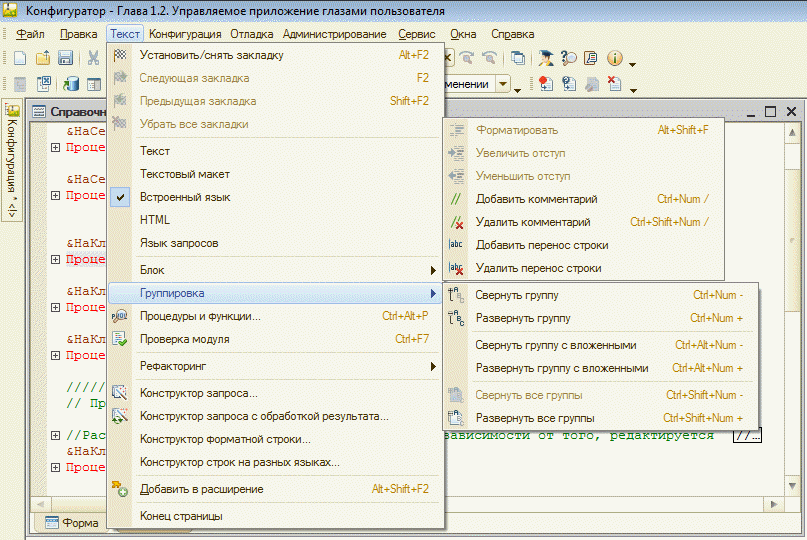
При изменении каких-либо настроек в окнах редактора они тут же применяются и изменяют вид формы. При выделении какого-либо элемента в окне элементов формы (1) он сразу же выделяется и показывается в окне предварительного просмотра формы (3), и наоборот.

Программный модуль описывает работу формы на встроенном языке. Для редактирования модуля формы можно воспользоваться закладкой Модуль, которая расположена внизу редактора форм.

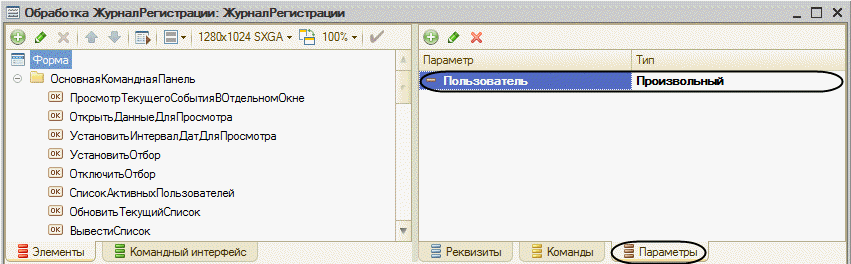
В модуле формы располагаются обработчики событий формы, элементов формы, команд формы. Помимо предопределенных обработчиков событий разработчик прикладного решения может создавать в модуле формы свои процедуры и функции.



Модуль формы, так же как и другие модули конфигурации, редактируется в специализированном редакторе, в котором для удобства разработчика предусмотрены различные интерфейсные команды и меню



С помощью закладки Параметры редактора формы можно попасть в окно редактирования параметров формы

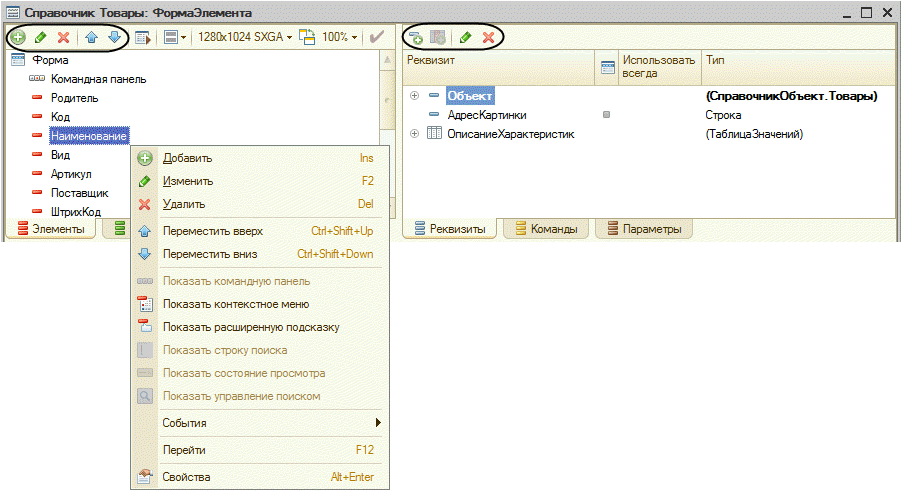


С помощью параметров формы организовано управление функциональностью формы при ее открытии в пользовательском режиме работы приложения. Если это необходимо, разработчик может изменить параметры формы уже в процессе ее работы.

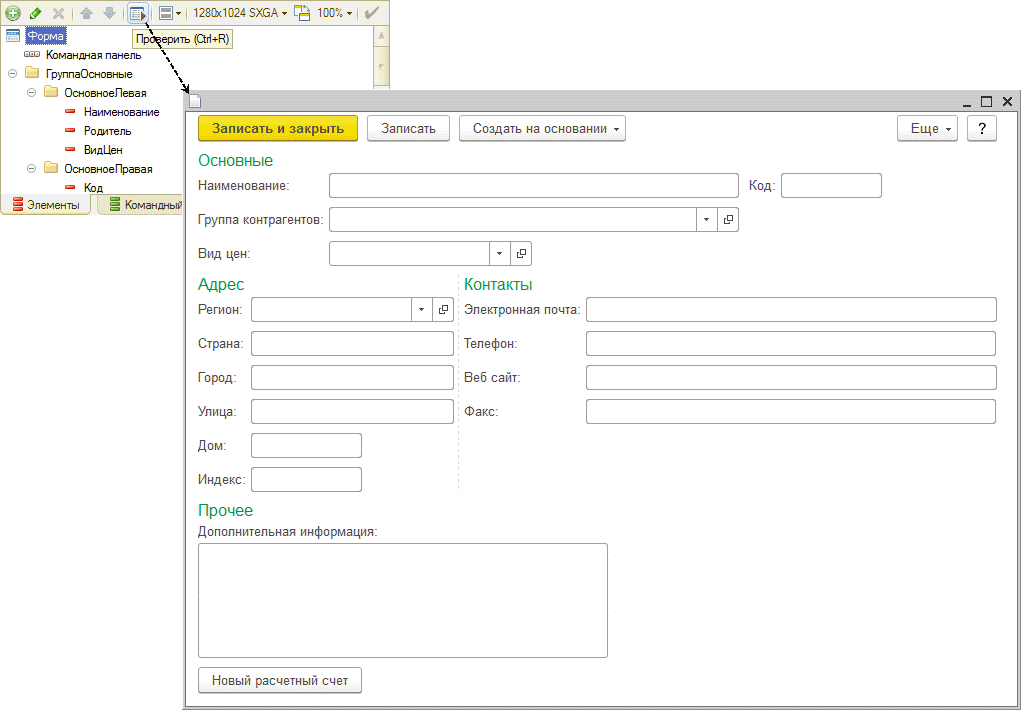
Кроме параметров, описанных разработчиком формы, при ее вызове и работе доступны параметры, автоматически предоставляемые расширением формы. Для разных объектов конфигурации, определяющих основной реквизит формы, набор параметров различается.

На состав параметров формы, предоставляемых основным реквизитом формы, влияет окружение объекта конфигурации, его свойства. Примером может служить параметр Основание, если это форма документа и этот документ может вводиться на основании какого-либо объекта.

Создание, изменение и удаление элементов формы, команд, реквизитов и прочее осуществляются с помощью кнопок командных панелей соответствующих областей редактора формы. Кроме того, для осуществления ряда действий доступны контекстные меню и перетаскивание с помощью мыши

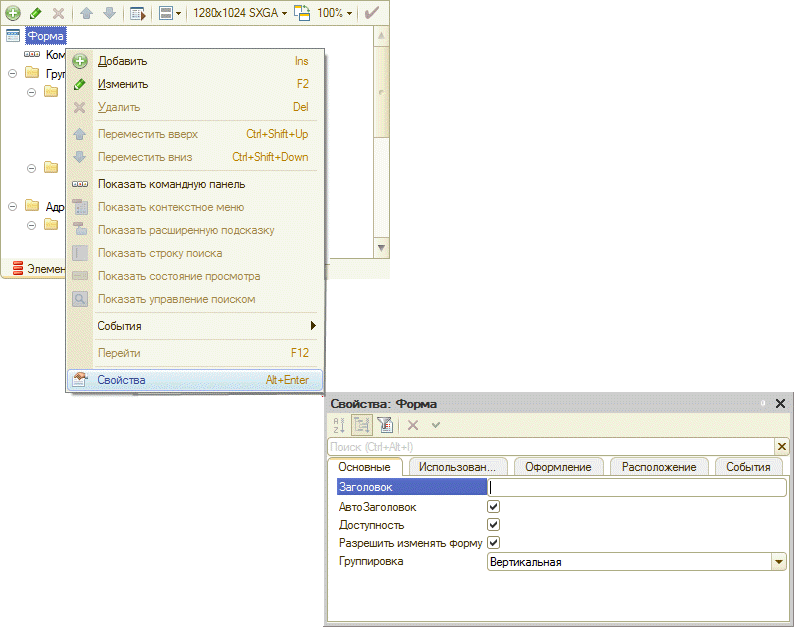


С помощью кнопки Проверить можно открыть форму не в окне предварительного просмотра, а в отдельном окне и посмотреть, как она будет выглядеть в интерфейсе приложения. После этого можно максимизировать окно формы или изменить размеры окна, как требуется (



Форма обладает рядом свойств, которые присущи ей всегда. Кроме того, ряд свойств формы определяется ее основным реквизитом. Именно основной реквизит формы определяет ее поведение, вид, состав команд, другие дополнительные возможности, предоставляемые разработчику формы и ее пользователю.

Для редактирования свойств формы необходимо воспользоваться контекстным меню (пункт меню Свойства) ветви Форма в окне элементов формы или просто дважды щелкнуть мышью на корне дерева элементов



Палитра свойств формы содержит множество свойств, собранных в группы Основные, Использование, Оформление и т. п. Рассмотрим наиболее важные из этих свойств.

Свойство Заголовок определяет заголовок формы, как его будут видеть разработчик в окне предварительного просмотра формы и пользователь во время работы с формой. Заголовок формы может быть дополнен системной информацией, полученной из свойств объекта конфигурации, связанных с расширенным представлением объектов, списков. Такое влияние на свойство Заголовок возможно, если установлено свойство АвтоЗаголово

Если свойство Заголовок не заполнено, а свойство АвтоЗаголовок включено, то заголовок формы товара будет сформирован полностью автоматически. При снятом автозаголовке и пустом заголовке заголовок формы будет отсутствовать.

С помощью свойства Отображать Заголовок можно управлять показом заголовка формы (включая автозаголовок и признак модификации). Это особенно актуально для специализированных режимов основного окна приложения Рабочее место, Полноэкранное рабочее место и Киоск.

*Режимы основного окна приложения*

*Установка режима основного окна – это комплексное решение, отражающее суть прикладного решения. Изменить режим основного окна можно из конфигуратора с помощью свойства Режим основного окна клиентского приложения или из встроенного языка при старте прикладного решения.*

*В платформе реализованы четыре режима основного окна:*

*Обычный,*

*Рабочее место,*

*Полноэкранное рабочее место,*

*Киоск.*

**Виды элементов формы**

В форме используются следующие элементы:

Группа:

* Обычная группа;
* Группа – Страницы;
* Группа – Командная панель;
* Группа колонок.

Поле.

Кнопка.

Таблица.

Декорация:

* Декорация – Надпись;
* Декорация – Картинка.
* Дополнение элемента формы.

Для того чтобы добавить в состав формы новый элемент определенного вида, необходимо воспользоваться одним из этих элементов. Для отображения различных данных прикладного решения необходимо изменять свойство элемента формы Вид.

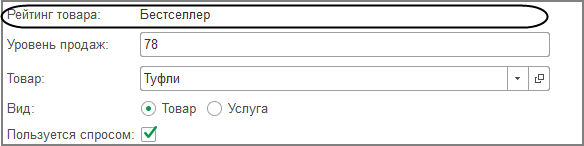
При добавлении нового элемента редактором формы будет проанализировано свойство Данные (с каким реквизитом будет связан элемент формы). В зависимости от этого платформа позволит выбрать один из нескольких возможных видов этого элемента.

Открытие свойств элементов формы ничем не отличается от рассматривавшегомя ранее открытия свойств формы.

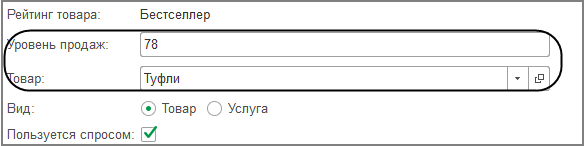
**Поле**

Элемент формы Поле предназначен для отображения примитивных типов данных, текстовых, табличных, HTML-документов, диаграмм, календарей, индикаторов и др. Тип данных, которые отображает элемент Поле, влияет на то, какие значения может принимать его свойство Вид.

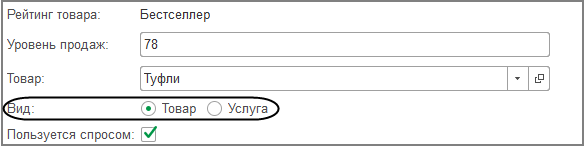
Поле надписи. Поле, недоступное для непосредственного редактирования в пользовательском режиме работы. Надпись изменяется из встроенного языка. Для отображения надписи в форме нужно создать реквизит типа Строка и связать его с полем формы вида Поле надписи.



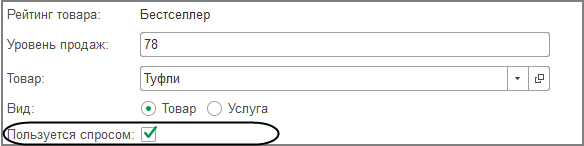
Поле ввода. Поле, допускающее редактирование данных. Это могут быть как примитивные типы данных (например, число), так и ссылочные данные (например, ссылки на элементы справочника). Для отображения поля ввода в форме нужно создать реквизит требуемого типа и связать его с полем формы вида Поле ввода.



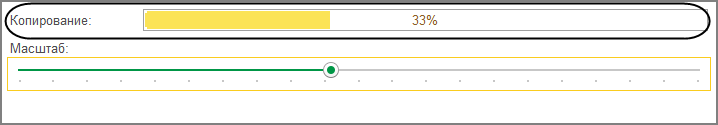
Поле переключателя. Поле, позволяющее выбрать один из нескольких вариантов значения с помощью отметки нужного. Для отображения переключателя в форме нужно создать реквизит типа Число или Строка, связать его с полем формы вида Поле переключателяи задать у этого поля список значений для выбора в свойстве СписокВыбора.



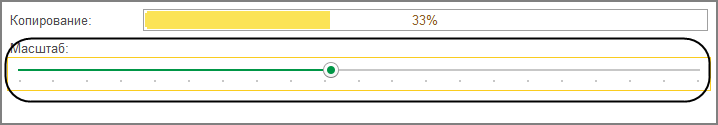
Поле флажка. Поле, предназначенное для отображения или установки одного из значений. Для отображения флажка в форме нужно создать реквизит типа Булево или Число и связать его с полем формы вида Поле флажка.



Поле индикатора. Поле, предназначенное для графического отображения текущего состояния реквизита формы. Для отображения индикатора в форме нужно создать реквизит типа Число и связать его с полем формы вида Поле индикатора.



Поле полосы регулирования. Поле, предназначенное для ввода числовых данных с помощью шкалы. Для отображения полосы регулирования в форме нужно создать реквизит типа Число и связать его с полем формы вида Поле полосы регулирования.

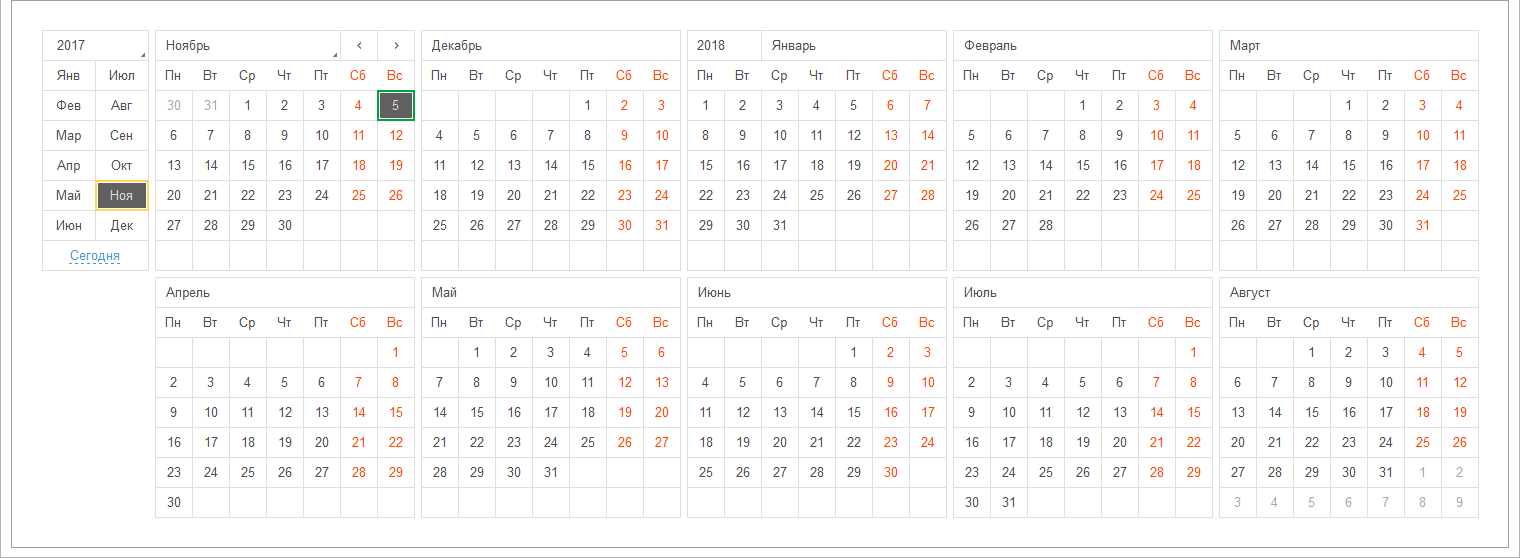


Элемент формы «Поле полосы регулирования»

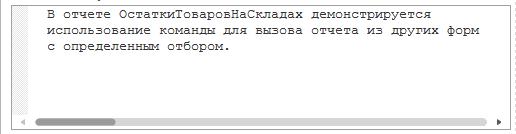
Поле картинки. Поле, отображающее картинку. Для отображения картинки в форме нужно создать реквизит типа Картинка и связать его с полем вида Поле картинки.



Поле календаря. Поле, предназначенное для выбора и отображения даты в виде календаря. Для отображения календаря в форме нужно создать реквизит типа Дата и связать его с полем вида Поле календаря.



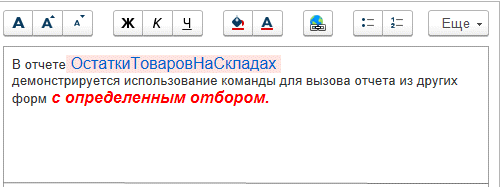
Поле текстового документа. Поле для редактирования и просмотра текстовых данных. Для отображения текстового документа в форме нужно создать реквизит типа ТекстовыйДокумент или Строка и связать его с полем вида Поле текстового документа.



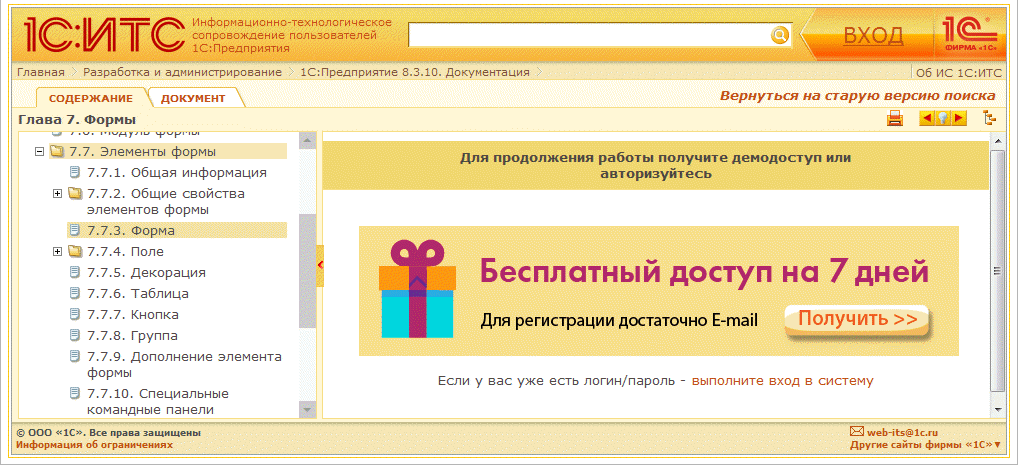
Поле табличного документа. Поле табличного вида (совокупность строк и столбцов), в котором отображаются и редактируются данные. Как правило, используется для отображения результатов работы отчетов и печатных форм. Может использоваться для ввода данных. Для отображения табличного документа в форме нужно создать реквизит типа ТабличныйДокумент и связать его с полем вида Поле табличного документа.



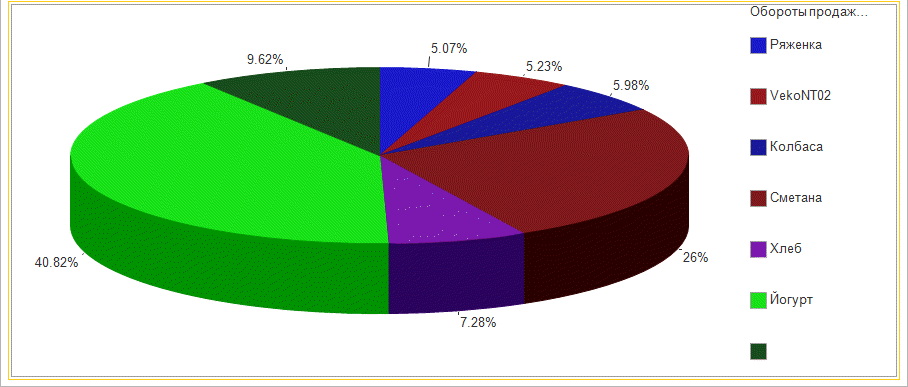
Поле форматированного документа. Поле для редактирования и просмотра текстовых данных различных форматов с возможностью их оформления. Для отображения форматированного документа в форме нужно создать реквизит типа ФорматированныйДокумент и связать его с полем вида Поле форматированного документа.



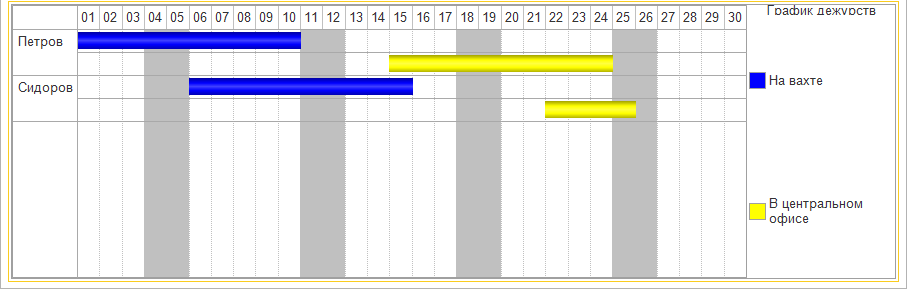
Поле HTML-документа. Поле для редактирования и просмотра HTML-ресурсов (рис. 2.62). Для отображения HTML-документа в форме нужно создать реквизит типа Строка и связать его с полем вида Поле HTML-документа. Строковый реквизит может содержать адрес для доступа к веб-контенту (URL интернет-ресурса) или сам HTML-код.



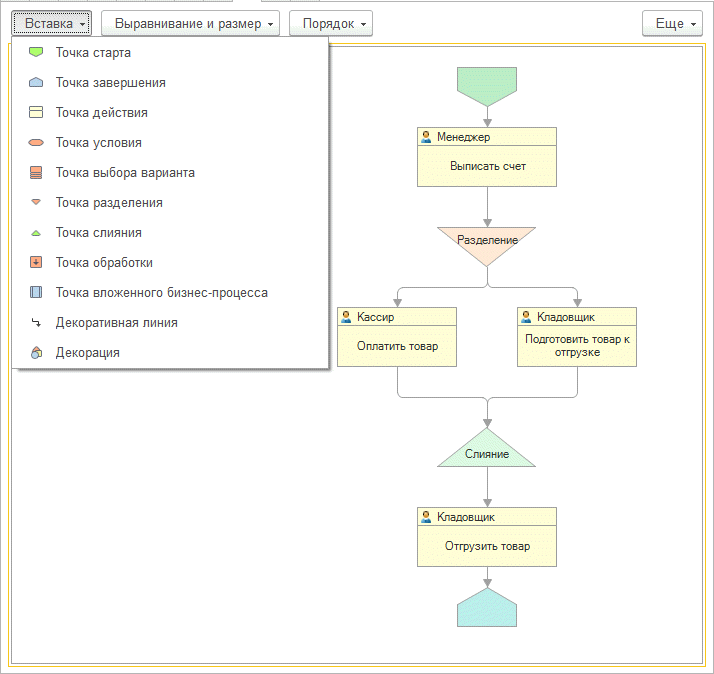
Поле диаграммы. Поле, отображающее различные виды диаграмм, представляющих данные в графическом виде. Для отображения диаграммы в форме нужно создать реквизит типа Диаграмма и связать его с полем вида Поле диаграммы.



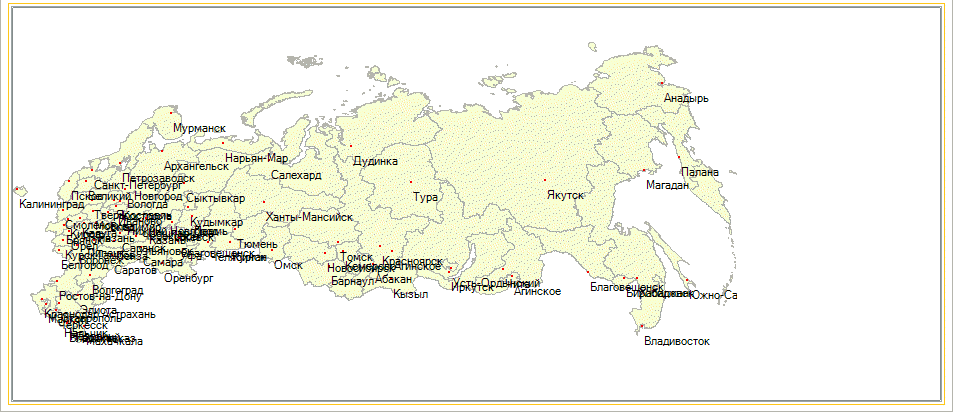
Поле диаграммы Ганта. Поле, отображающее диаграмму Ганта. Диаграмма Ганта является способом отражения длительности и последовательности процессов, которые показываются в виде полос, расположенных вдоль оси времени (рис. 2.64). Для отображения диаграммы Ганта в форме нужно создать реквизит типа ДиаграммаГанта и связать его с полем вида Поле диаграммы Ганта.



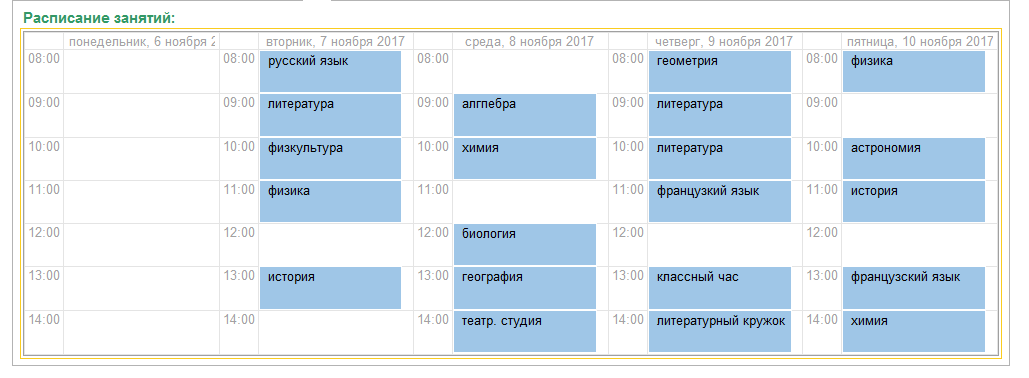
Поле графической схемы. Поле для просмотра и редактирования графических схем, созданных в специализированном редакторе, входящем в состав «1С:Предприятия» .Для отображения графической схемы в форме нужно создать реквизит типа ГрафическаяСхема и связать его с полем вида Поле графической схемы.



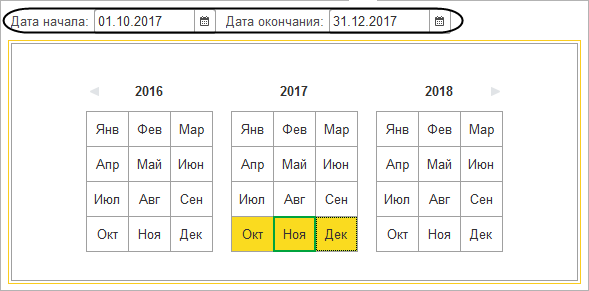
Поле географической схемы. Поле для просмотра итоговых данных в разрезе их географического положения. Для отображения географической схемы в форме нужно создать реквизит типа Географическая Схема и связать его с полем вида Поле географической схемы.



Поле планировщика. Поле для просмотра и редактирования запланированных событий, привязанных к дате и времени, с возможностью повторения этих событий с определенной периодичностью. Для отображения планировщика в форме нужно создать реквизит типа Планировщик и связать его с полем вида Поле планировщика.



Поле периода. Поле для выбора периода дат, кратных месяцу. Для отображения поля периода в форме нужно создать реквизит типа СтандартныйПериод и связать его с полем вида Поле периода.

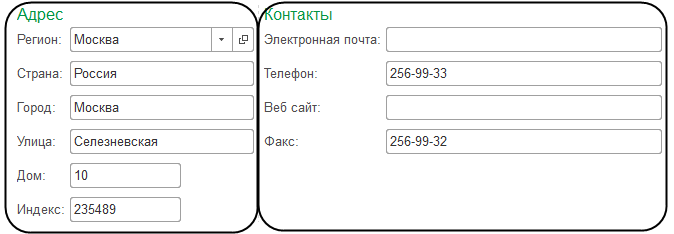


**Группа**

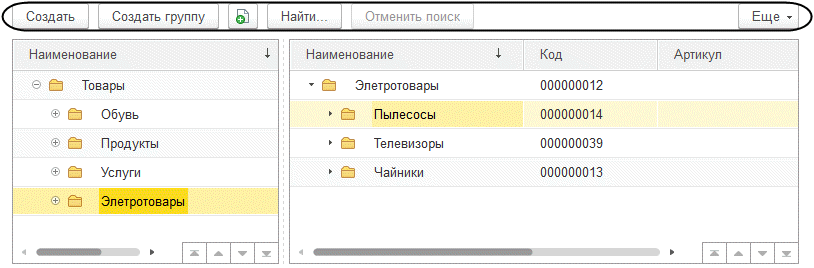
Элемент формы Группа предназначен для группировки других элементов формы. Это могут быть группы полей, группы страниц, группы команд. Также для элементов типа Таблица можно создавать группы колонок. Группы могут быть вложенными друг в друга.

Свойство Вид элемента формы Группа может принимать значения:

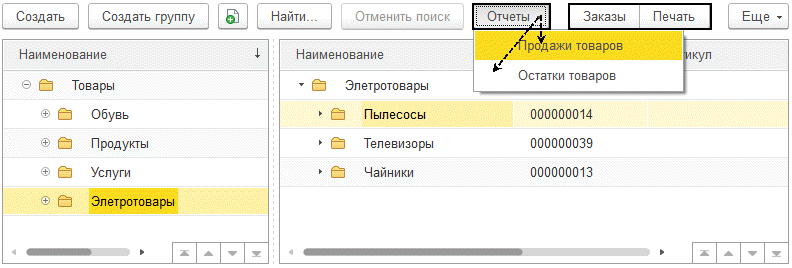
Обычная группа. Элемент формы, предназначенный для логической группировки других элементов формы. Элемент Обычная группа может не выделяться в форме, выполняя при этом функции группировки других элементов. Выделением группы управляет свойство Отображение.



Командная панель. Элемент формы, предназначенный для группировки кнопок и групп команд. Наполнение командной панели конкретными командами определяется свойством Источник команд. Источником команд является таблица формы, отображающая динамический список.



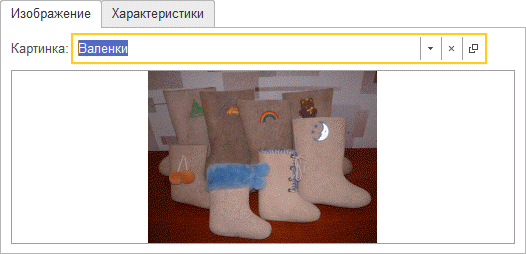
Помимо отдельных кнопок в командную панель можно добавлять подчиненные группы следующего вида:

**Подменю.** Элемент формы, представляющий собой выпадающее меню.

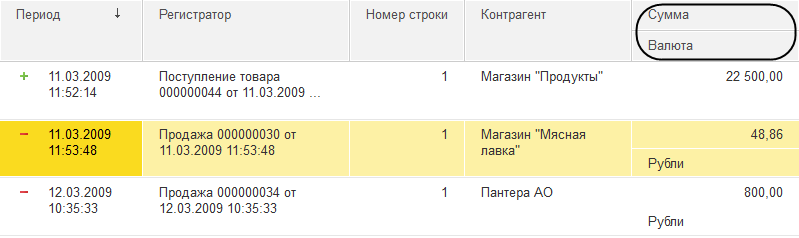
**Группа кнопок.** Элемент формы, позволяющий сгруппировать кнопки в логические группы. Сам элемент не отображается на форме, но кнопки в группе можно расположить компактно (без отступов друг от друга).

**Страницы**. Элемент формы, предназначенный для организации панели с закладками. Для того чтобы добавить на такую панель страницы, необходимо добавить столько вложенных групп вида Страница, сколько страниц должно быть у панели

Страница. Элемент формы, который может быть создан только как подчиненный элементу Страницы. Представляет собой подчиненную группу-страницу с данными (в нашем случае открыта страница «Изображение»). В пользовательском режиме работы страницы, на которых отсутствуют данные, не отображаются.

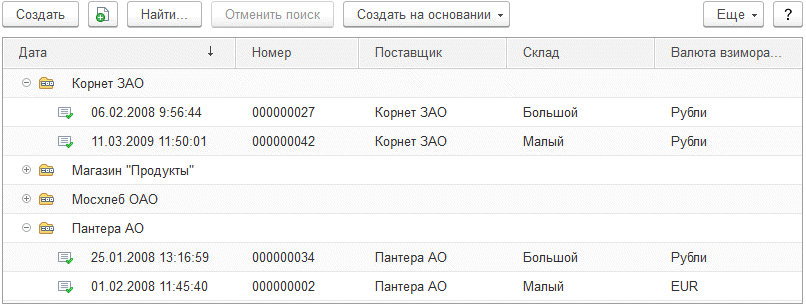


Группа колонок – позволяет объединять колонки в таблице. С помощью группы этого вида можно изменять правило группировки колонок.



**Таблица**

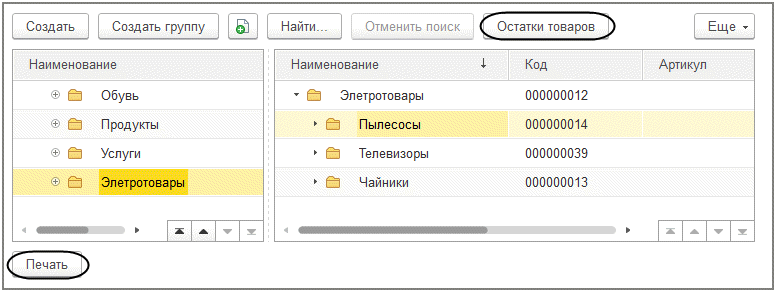
Элемент формы Таблица предназначен для отображения и редактирования различных табличных данных . Это может быть динамический список, табличная часть, список значений и т. д. У таблицы формы могут быть свои командные панели, контекстные меню. Поля колонок элемента Таблица могут быть сгруппированы



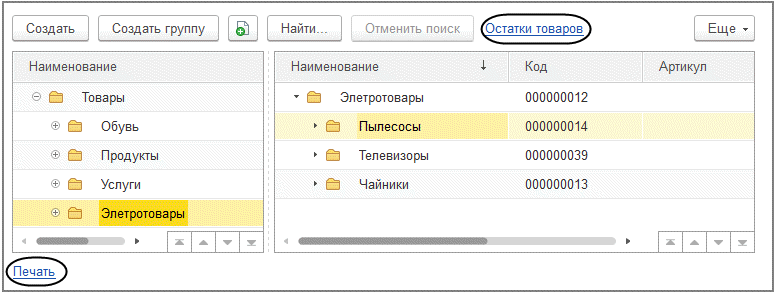
**Кнопка**

Элемент формы Кнопка предназначен для отображения кнопок и гиперссылок, при нажатии на которые выполняются связанные с ними команды. Кнопки могут быть подчинены командной панели. В зависимости от этого свойство Вид элемента формы Кнопка может принимать значения:

Кнопка, Кнопка командной панели. Элемент представляет собой обыкновенную кнопку, расположенную непосредственно в форме или в составе командной панели. Если для кнопки не назначена команда, которая будет выполняться при нажатии, то кнопка в пользовательском режиме не отображается.



**Гиперссылка,** Гиперссылка командной панели. Кнопка формы или кнопка командной панели, отображающаяся в виде гиперссылки. Если для гиперссылки не назначена команда, которая будет выполняться при нажатии, то гиперссылка в пользовательском режиме не отображается.



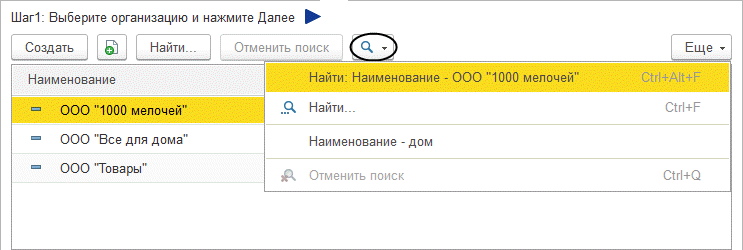
Дополнение элемента формы

Дополнение элемента формы предназначено для отображения дополнительных свойств элемента формы (на момент написания книги это дополнения таблицы, отображающей данные динамического списка) и управления этими элементами. Существуют следующие дополнения:

Отображение строки поиска. Позволяет настроить отображение строки поиска.

Состояние просмотра. Позволяет настроить отображение перечня примененных поисковых запросов. Не может располагаться в командной панели.

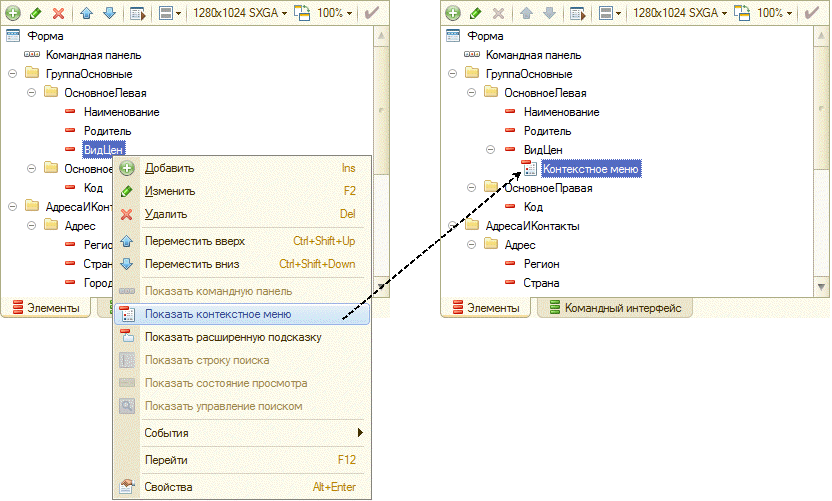
Управление поиском. Позволяет настроить внешний вид кнопки управления



**Контекстное меню элементов формы**

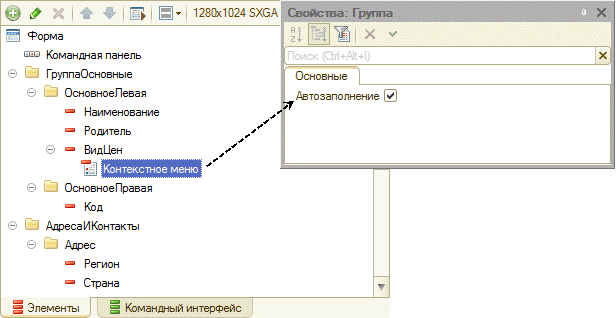
Для всех элементов формы характерно наличие контекстного меню, вызываемого с помощью мыши или клавиатуры. В большинстве случаев во время работы приложения контекстные меню формируются платформой автоматически. Однако разработчик может вмешаться в этот процесс и повлиять на состав контекстного меню.

Чтобы получить доступ к контекстному меню элемента формы, нужно из контекстного меню элемента формы в редакторе формы выполнить команду Показать контекстное меню.



После этого появится возможность добавлять в контекстное меню элемента собственные команды.

Если разработчик желает модифицировать автоматически формируемое контекстное меню элемента формы, то флажок Автозаполнение в свойствах Контекстное меню элемента снимать не следует. В случае необходимости формирования полностью своего контекстного меню флажок следует снять.



Задание

Разработать набор форм для вашей собственной конфигурации