

Федеральное государственное образовательное бюджетное учреждение высшего образования "Финансовый университет при Правительстве Российской Федерации"

Федеральное государственное автономное образовательное учреждение высшего образования "Московский физико-технический институт (национальный исследовательский университет)"

Федеральное государственное бюджетное образовательное учреждение высшего образования "Российский экономический университет имени Г.В. Плеханова"

Федеральное государственное бюджетное образовательное учреждение высшего образования "Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)"

Федеральное государственное бюджетное образовательное учреждение высшего образования "Московский государственный технологический университет "СТАНКИН"

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования "Московский технический университет связи и информатики"

Федеральное учебно-методическое объединение по укрупненной группе специальностей и направлений подготовки высшего образования "Информатика и вычислительная техника"

Фирма "1С"

Мероприятие проходит при поддержке Федерального агентства связи (Россвязь)

**XX международная научно-практическая конференция
"Новые информационные технологии в образовании"
(Технологии 1С: Перспективные решения для построения карьеры,
цифровизации организаций и непрерывного обучения)
4–5 февраля 2020 года**

**Мастер-класс:
Система быстрой разработки кроссплатформенных приложений или
1С – не Pascal переведенный Промтом!**

Раздаточный материал

г. Москва, 2020

ПРАВО ТИРАЖИРОВАНИЯ ДОКУМЕНТАЦИИ ПРИНАДЛЕЖИТ ФИРМЕ "1С"

Получив настоящие материалы для обучения от фирмы "1С",

Вы тем самым даете согласие не допускать

их копирования без письменного

разрешения фирмы "1С".

© ООО "1С-Софт"

Москва, ул. Селезневская, д. 21

Телефон: (495) 688-90-02

Автор методических материалов: 1С:Учебный центр №1

Оглавление

Введение	4
Постановка задачи	5
Создание новой информационной базы	6
Конфигуратор. Первый взгляд	12
Создаем справочники	14
Создаем документы	22
Работа с формой	25
Ввод на основании	33
Печать	37
Регистры	46
Отчеты.....	53
Заключение	67

Введение

Зачастую преподаватели it-специальностей в начале обучения новой для них темы "программирование в среде 1С:Предприятие (1С:Enterprise)" применяют свои заготовки и примеры из других языков программирования, которые они использовали ранее, C++, Python или Java. Это в корне неверный подход для изучения предметно-ориентированного языка программирования. Точнее даже не языка программирования, а среды разработки, в которую встроен предметно ориентированный язык программирования.

В данном мастер-классе мы попробуем продемонстрировать экспресс курс для знакомства с возможностями системы 1С:Предприятие (1С:Enterprise) на примере быстрой разработки небольшого программного продукта для ведения учета торговой компании. Покажем сильные стороны системы.

Цель – заинтересовать слушателей в изучении 1С.

Постановка задачи

Изучая domain-specific language (DSL), предметно-ориентированный язык, коим является 1С:Предприятие (1С:Enterprise), важным является как раз разобраться с каким "предметом" этот язык "заточен" работать. Для системы 1С это бизнес-приложения.

Мы поставим задачу по автоматизации торговой компании.

Необходимо создать программу, учетную систему, в которой будут автоматизированы следующие операции:

- Выставление счетов на продажу товаров: ввод, хранение, печать документов
- Ввод, хранение и печать накладных на отгрузку товаров
- Создание отчетов по продажам товаров, в табличном виде и в виде диаграмм

Естественно мы сильно упростим задачу, чтобы уложиться в отведенное нам время, будем по максимуму использовать конструкторы, которые помогут не вводить большинство кода руками. Но не стоит относиться к системе 1С как к простому языку программирования или недоязыку. В системе заложены огромные возможности, к примеру механизмы прогнозирования и анализа данных. Но цель нашего мастер-класса показать сильные стороны платформы и заинтересовать слушателей изучать дальше систему.

Создание новой информационной базы

Для начала создадим новую информационную базу. Информационная база – это база данных + специальная конфигурация в которой и описывается бизнес-логика приложения.

Добавление информационной базы/группы ✕

Добавление информационной базы в список:

- ☒ **Создание новой информационной базы**
Создание информационной базы из поставляемой конфигурации, поставляемой демонстрационной базы или создание пустой информационной базы без конфигурации
- ☐ **Добавление в список существующей информационной базы**
Включение в список ранее созданной информационной базы, расположенной на веб-сервере (в интернете), на данном компьютере, в локальной сети или на сервере 1С:Предприятия
- ☐ **Создание новой группы**
Создание группы в дереве информационных баз

< Назад Далее > Отмена

Базу будет создавать пустую, новую.

Добавление информационной базы/группы ×

☐ Создание информационной базы из шаблона

Выберите поставляемую конфигурацию для начала работы или демонстрационный пример для ознакомления:

+

1С:Бухгалтерия предприятия

+

1С:Библиотека интеграции с 1С:Документооборотом

+

1С:Документооборот 8 КОРП

+

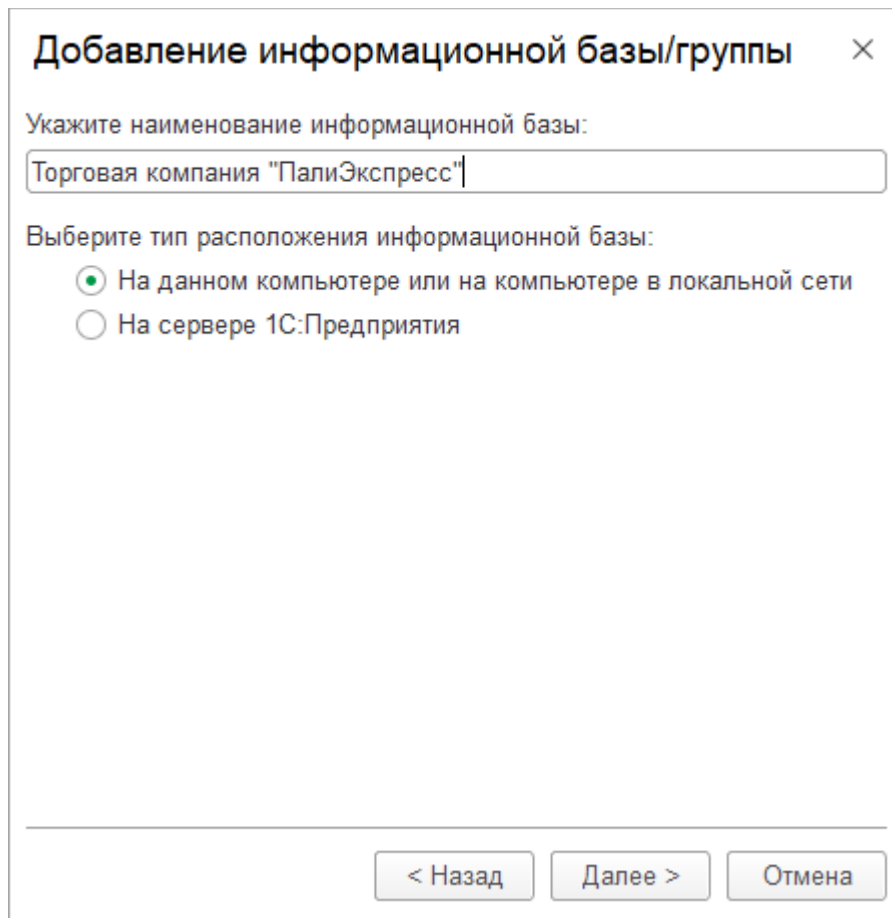
1С:Управление торговлей

☒ Создание информационной базы без конфигурации для разработки новой конфигурации или загрузки выгруженной ранее информационной базы

< Назад

Далее >

Отмена



Добавление информационной базы/группы ×

Укажите наименование информационной базы:

Торговая компания "ПалиЭкспресс"

Выберите тип расположения информационной базы:

- ☒ На данном компьютере или на компьютере в локальной сети
- ☐ На сервере 1С:Предприятия

< Назад Далее > Отмена

Система может работать в упрощенном варианте, или же можно развернуть кластер серверов для управления серьезной учетной системой в большом предприятии. 1С работает во многих крупных компаниях, к примеру, на заводе "КАМАЗ" и в Почте России.

Добавление информационной базы/группы

×

Укажите параметры информационной базы:

Каталог информационной базы:

C:\Users\Chistov_P\Documents\InfoBase1

...

Язык (Страна):

русский (Россия)

▼

< Назад

Далее >

Отмена

Добавление информационной базы/группы ×

Укажите параметры запуска:

Вариант аутентификации (определения пользователя):

- ☒ Выбирать автоматически
- ☐ Запрашивать имя и пароль

Скорость соединения: Обычная ▾

Дополнительные параметры запуска:

Основной режим запуска:

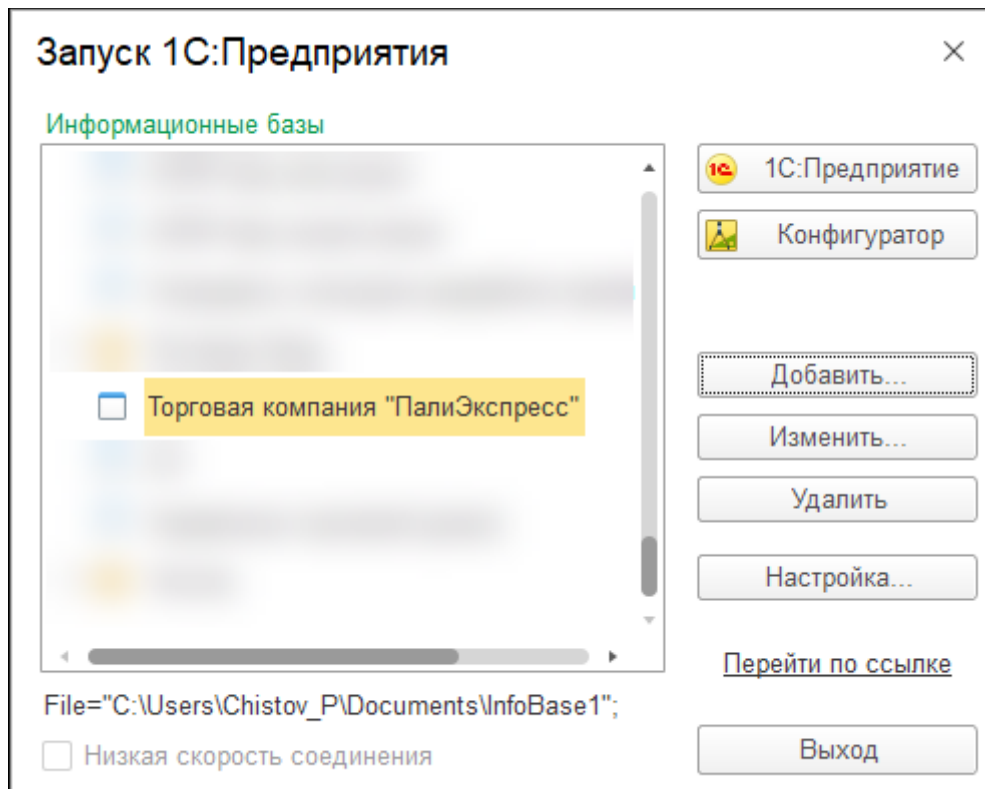
- ☒ Выбирать автоматически
- ☐ Тонкий клиент
- ☐ Веб-клиент
- ☐ Толстый клиент

Версия 1С:Предприятия: 8.3

Разрядность: ▾

< Назад Готово Отмена

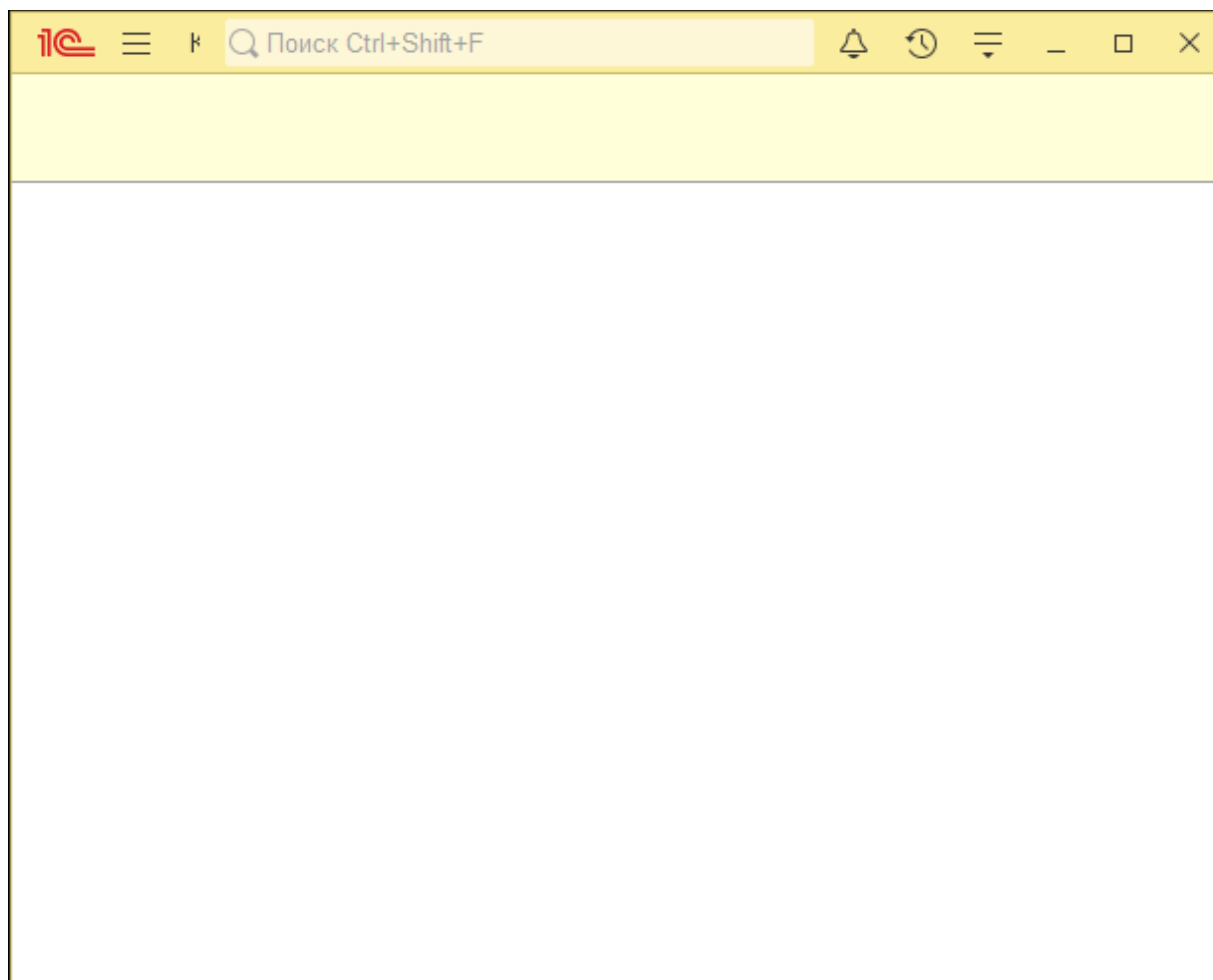
Система может работать в операционных системах MS Windows, Linux, MacOS, Android, iOS, Windows Mobile и в браузерах.



Информационная база создана.

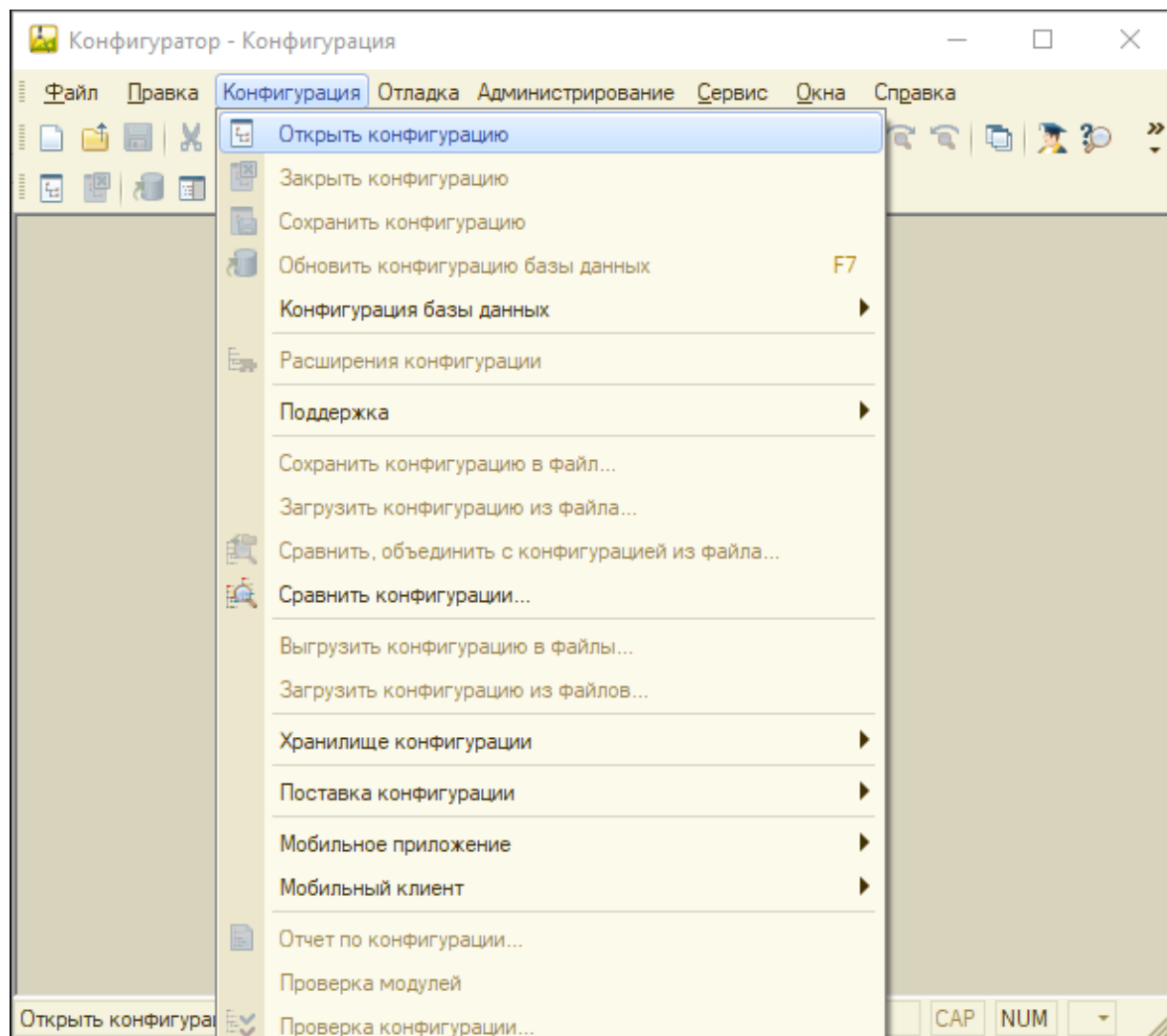
Система 1С:Предприятие поставляется сразу и как среда разработки и как среда исполнения.

Для начала запустим среду исполнения, режим 1С:Предприятие. Как и ожидается система отображает пустое окно.

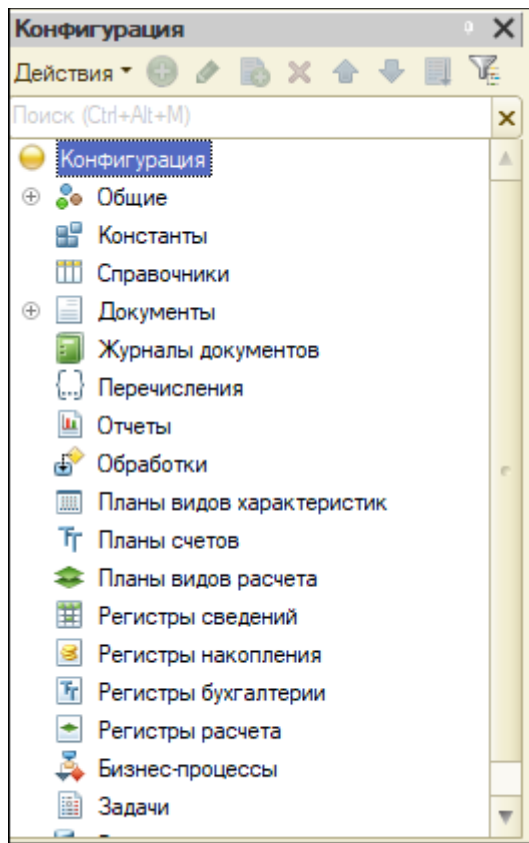


Конфигуратор. Первый взгляд

Откроем среду разработки – Конфигуратор. В конфигураторе откроем дерево метаданных – Конфигурацию.



Конфигурация отображает дерево прикладных объектов.



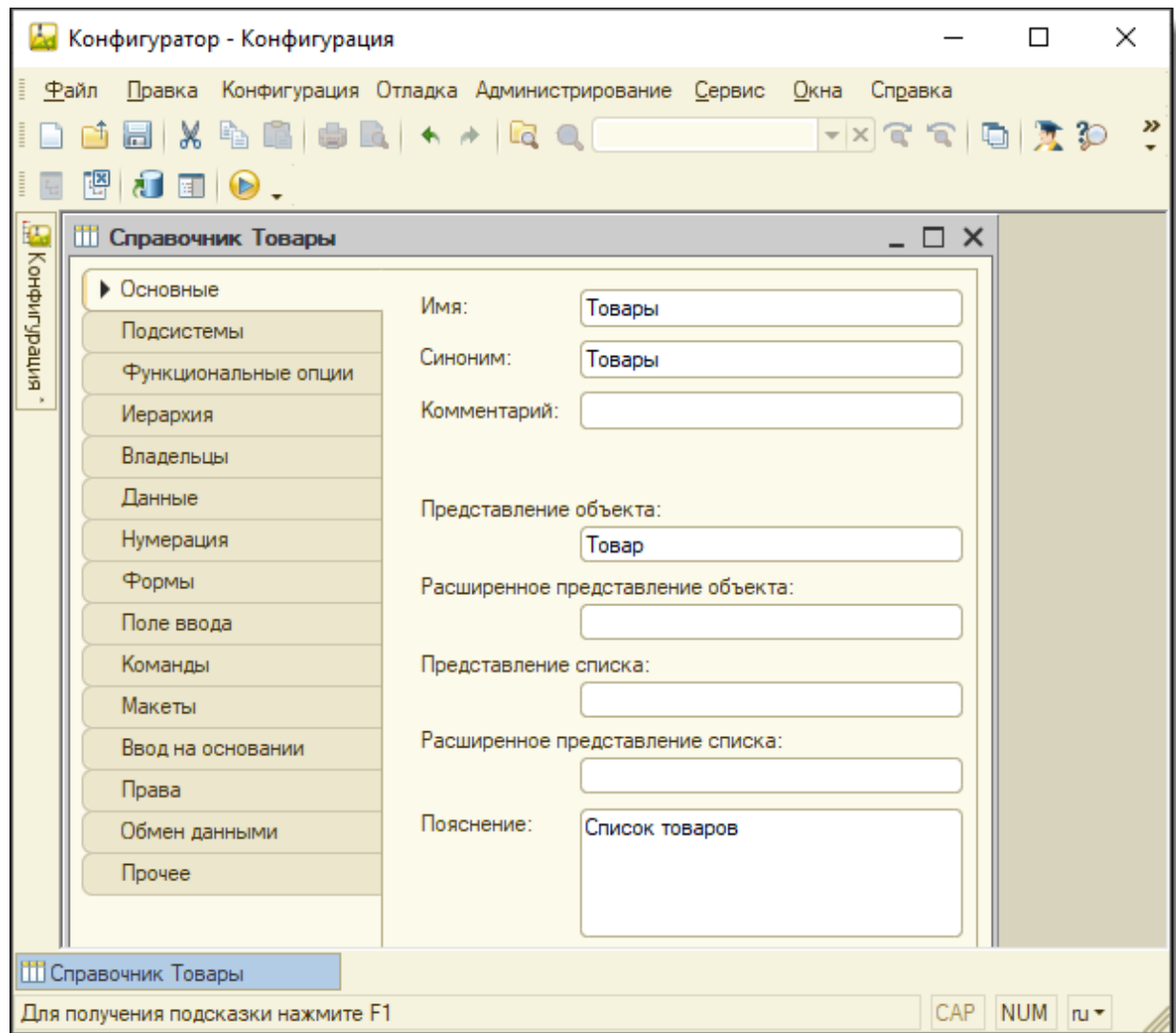
В других языках программирования их бы назвали "предопределенные классы объектов".

Большинство веток интуитивно понятны, к примеру "Отчеты", "Документы".

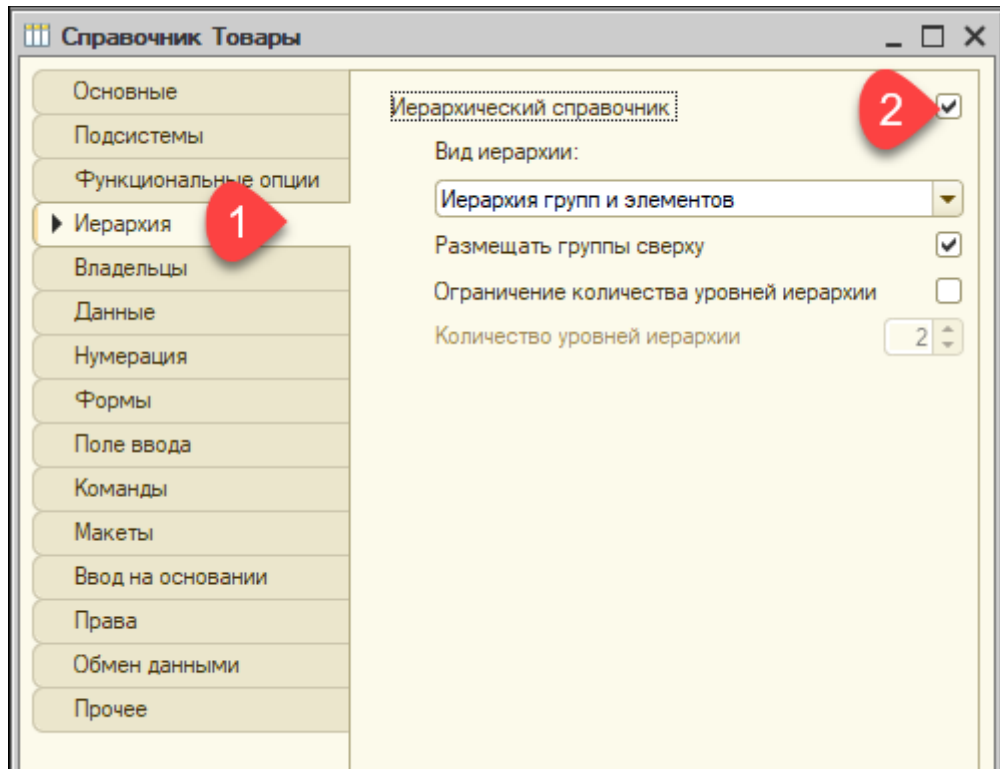
Для нашей задачи мы начнем со справочников. Нам необходимо хранить данные о товарах и покупателях.

Создаем справочники

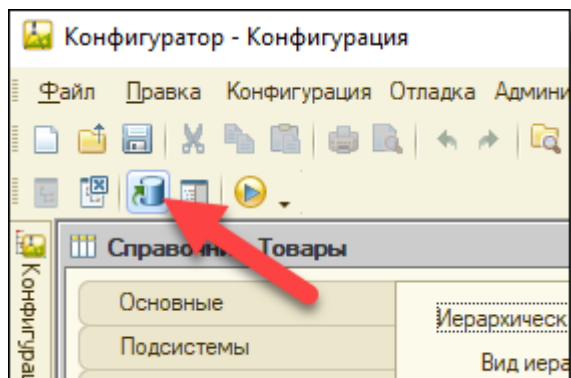
Добавим справочник "Товары".

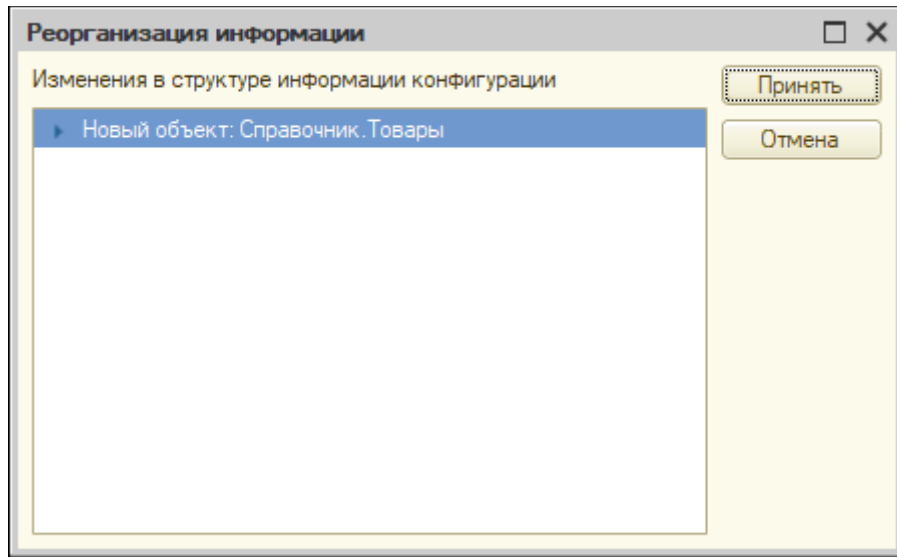


Так как товаров может быть много предусмотрим возможность создания групп, включим иерархию в справочнике.

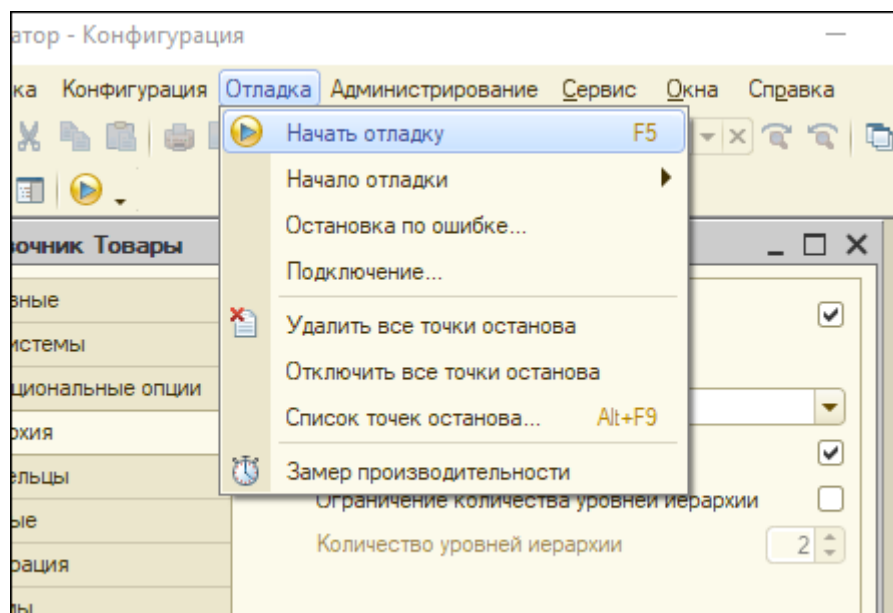


Сейчас система создаст новую таблицу в базе данных, и подключит механизмы интерфейса для работы пользователя. Сохраним конфигурацию.

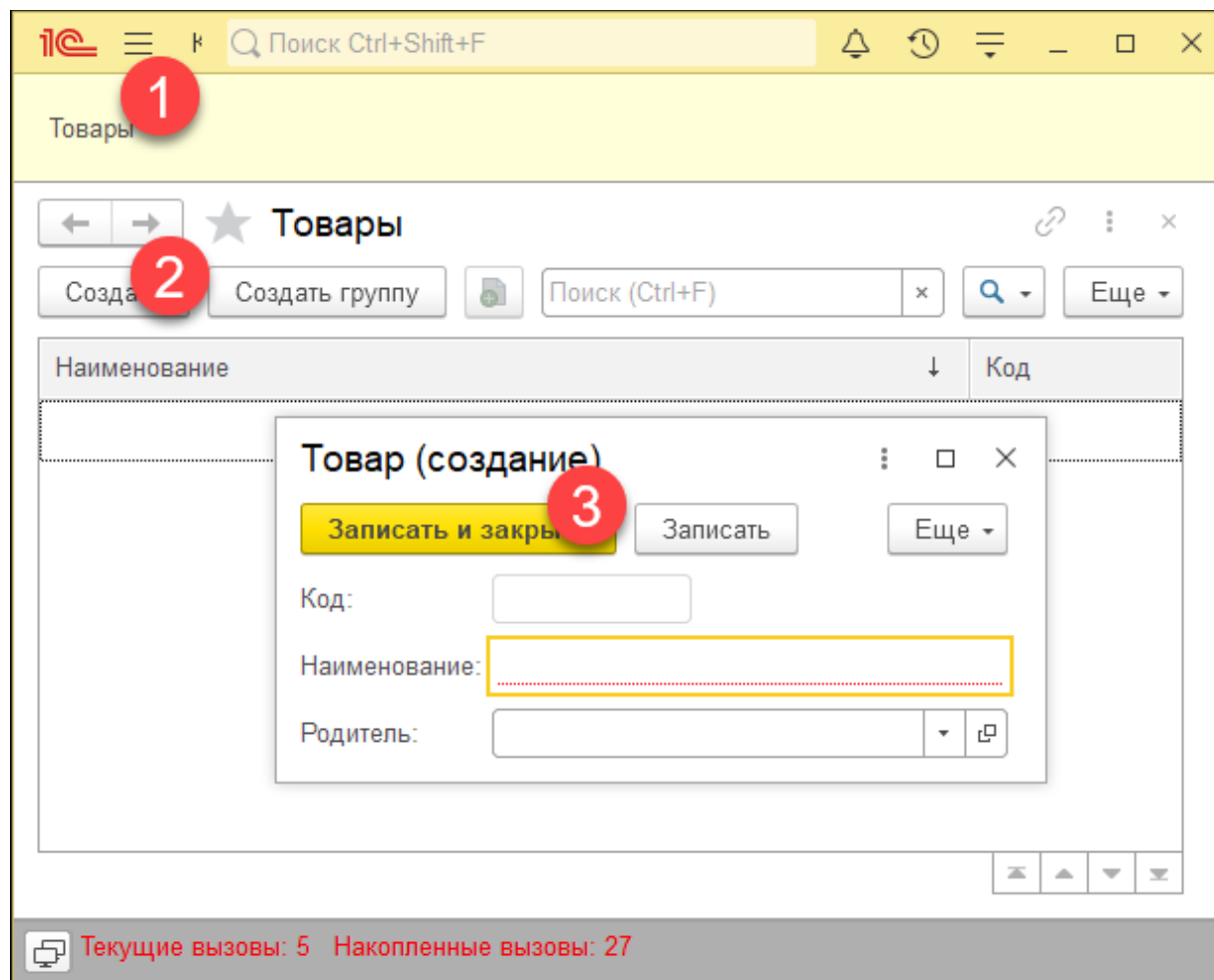




И запустим систему в режиме отладки.

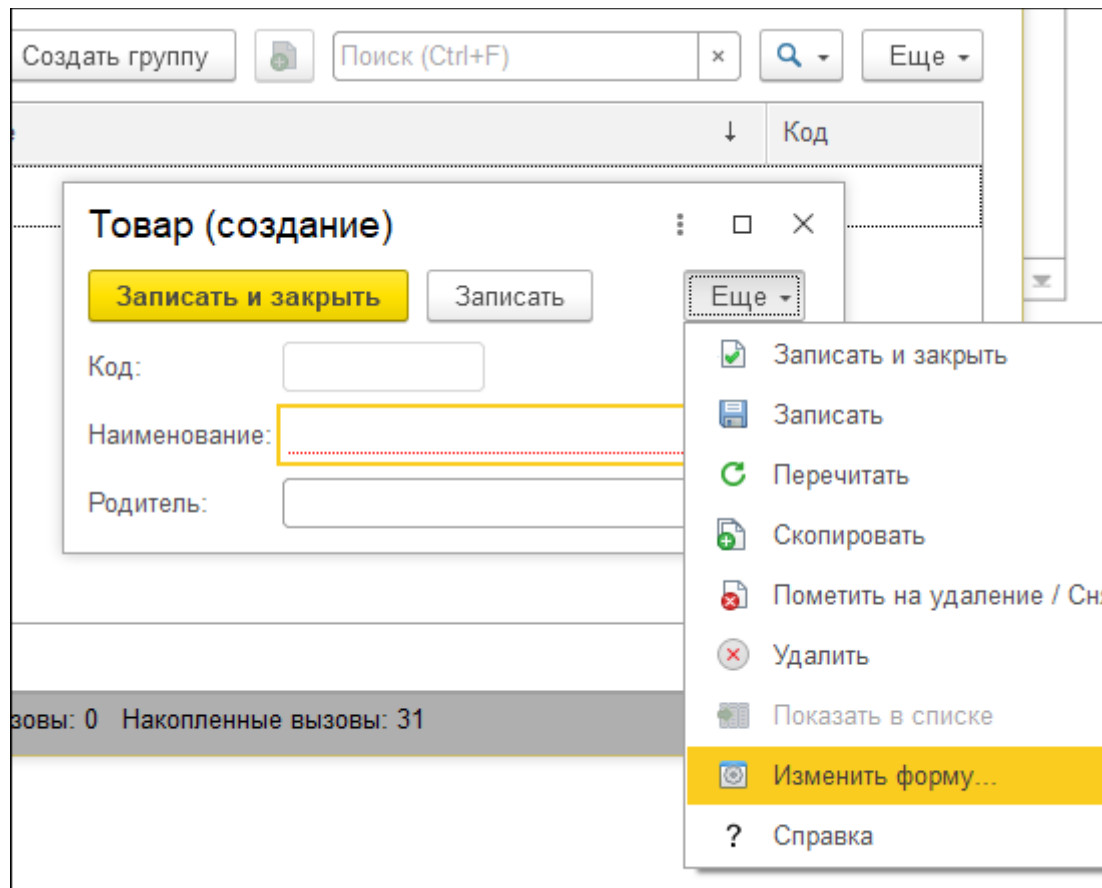


Откроем справочник. Создадим несколько элементов и групп.



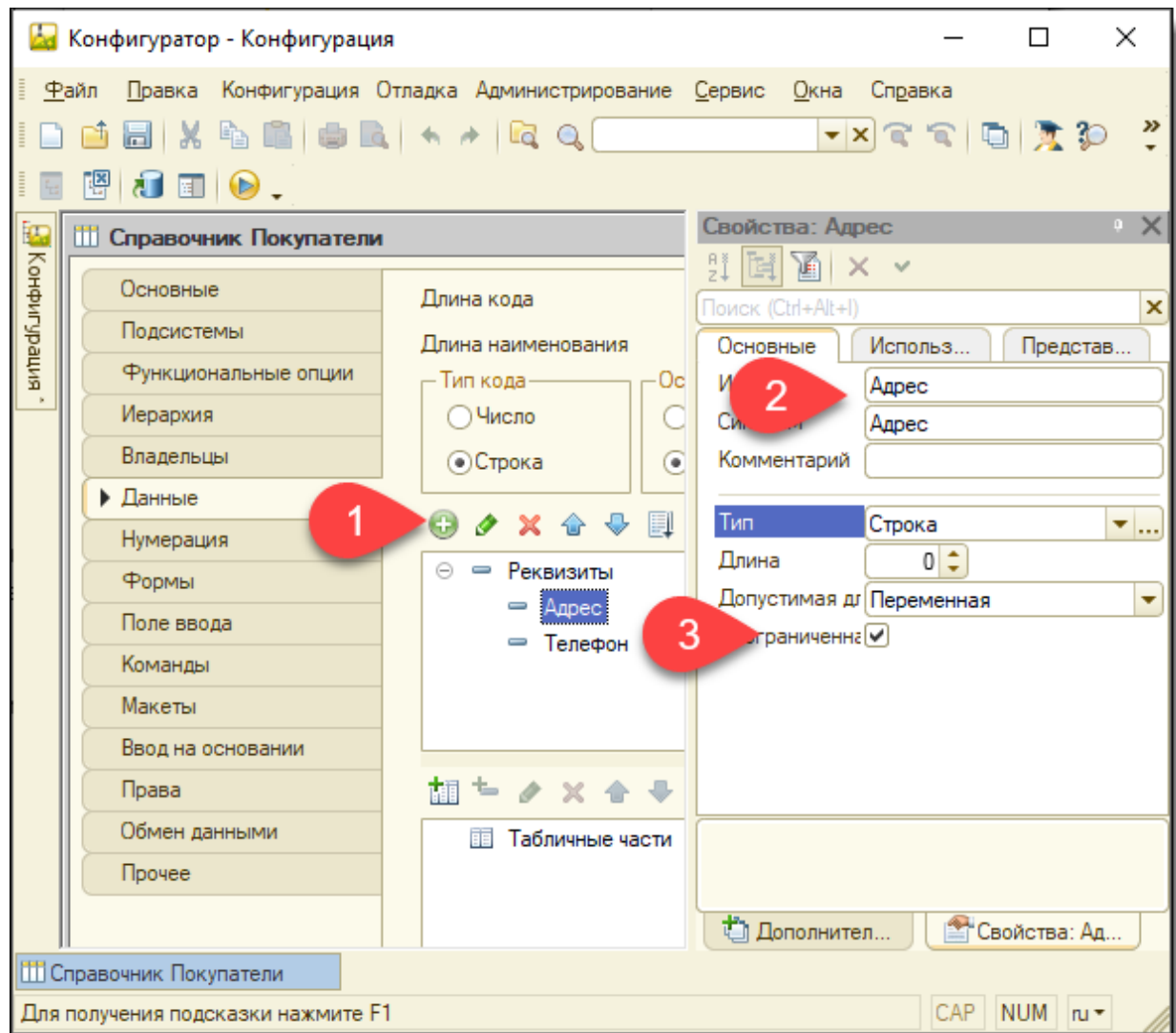
Как мы видим, формы были созданы системой автоматически. Все данные сохраняются в базе данных.

Если нас устраивает внешний вид форм, то их можно не создавать в режиме разработки. Если нужно скрыть или переместить элементы формы, то это также можно сделать самому пользователю, нет необходимости утруждать программиста.



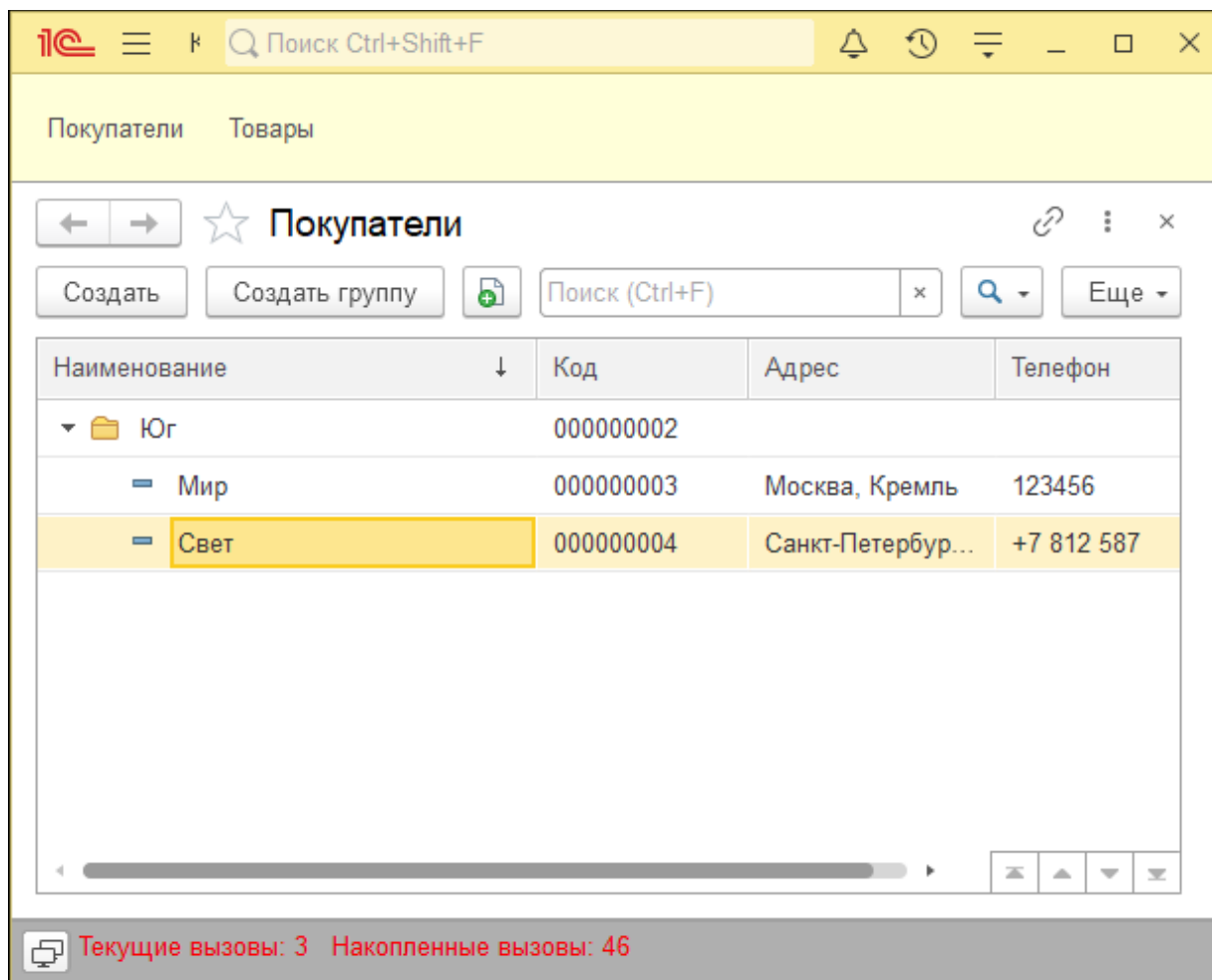
Если необходимы какие-либо специфические настройки или поведение формы, ее реакция на работу пользователя, всегда можно создать форму и описать программно все ее поведение.

Подобным образом создадим справочник "Покупатели". Но в нем кроме иерархии добавим еще несколько дополнительных реквизитов.

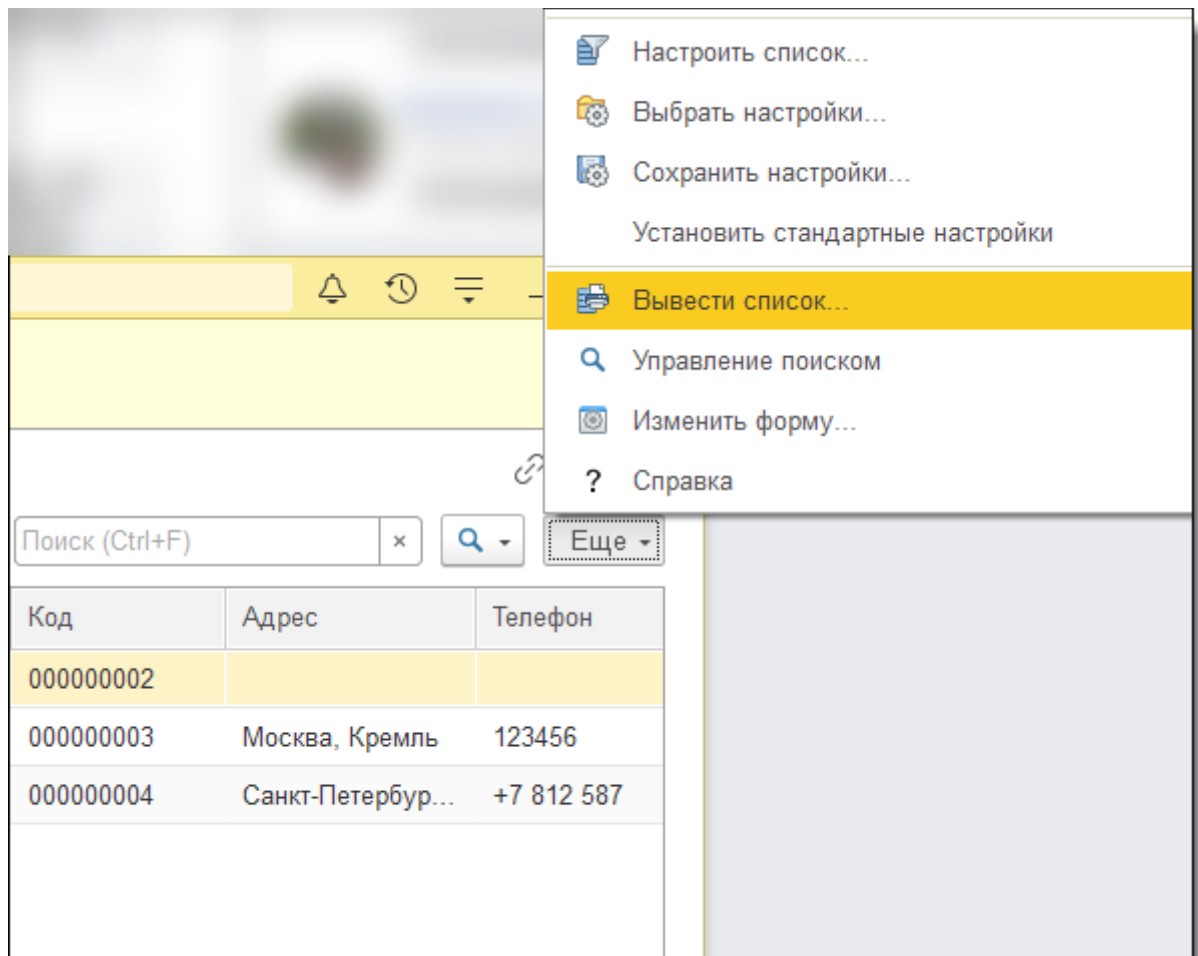


Адрес будет строкой неограниченной длины. А телефон оставим строкой длиной 10 символов. С другими типами познакомимся немного позже.

В режиме 1С:Предприятие введем несколько покупателей.

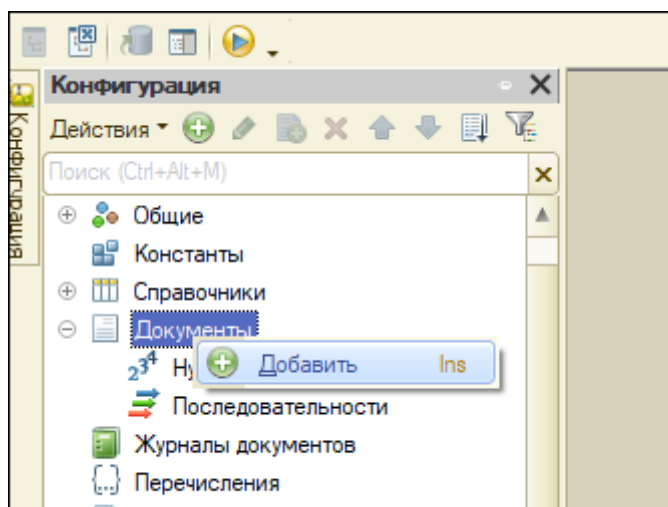


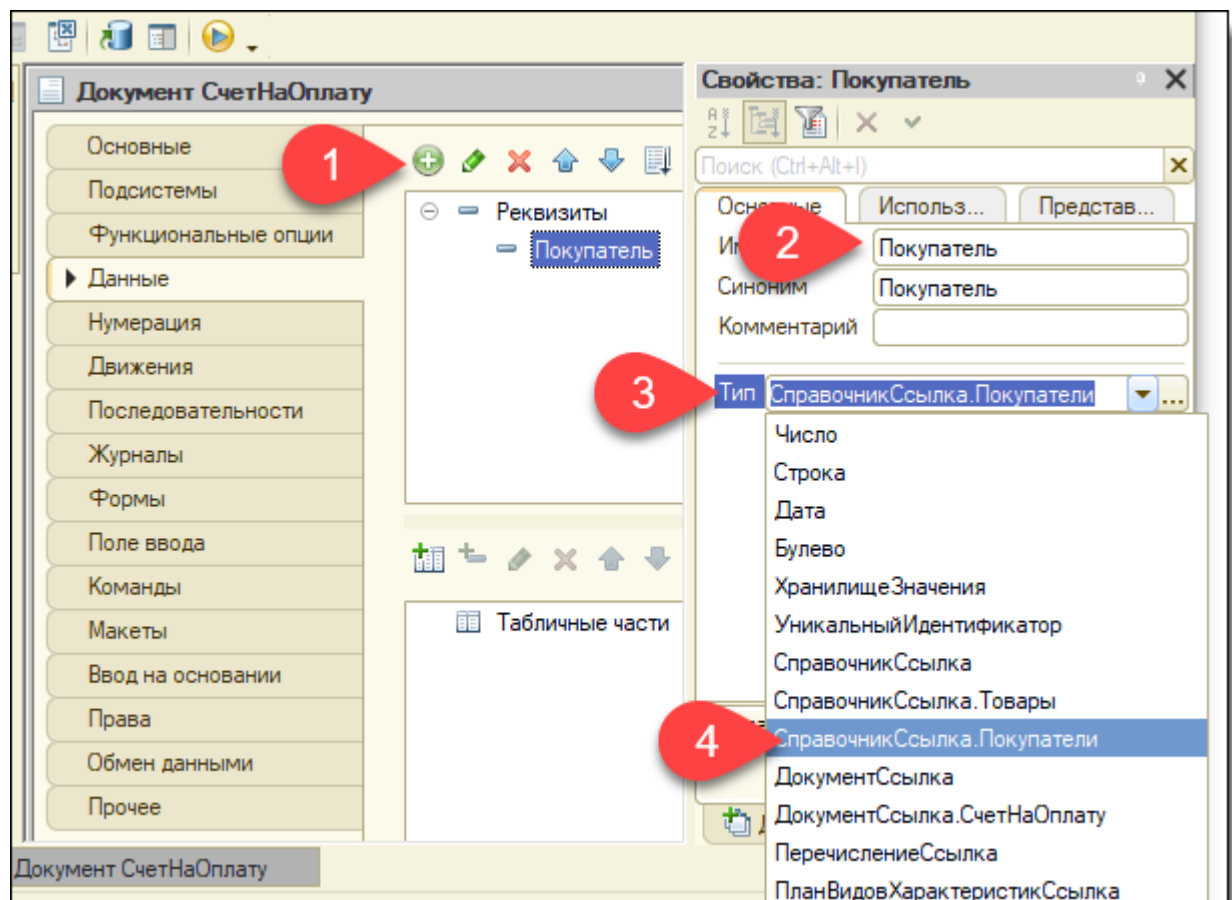
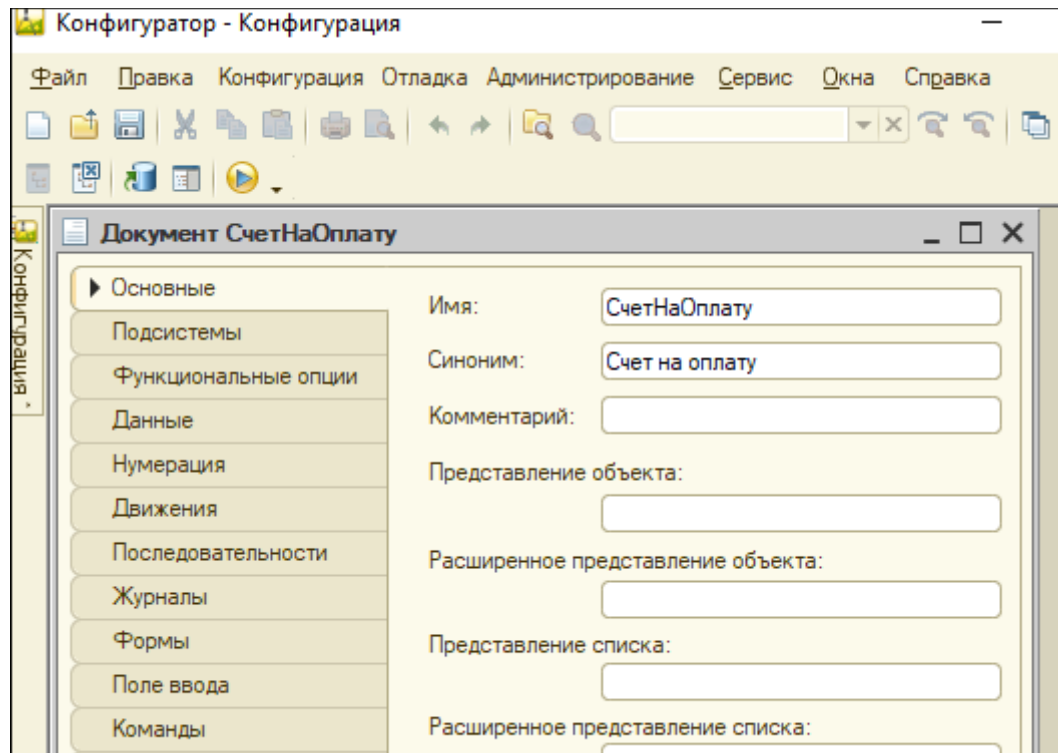
При желании, пользователь может напечатать любой список, который он видит на экране, это стандартная возможность системы. Так же можно сохранить данные списка в множество популярных форматов, к примеру, в файл MS Excel.



Создаем документы

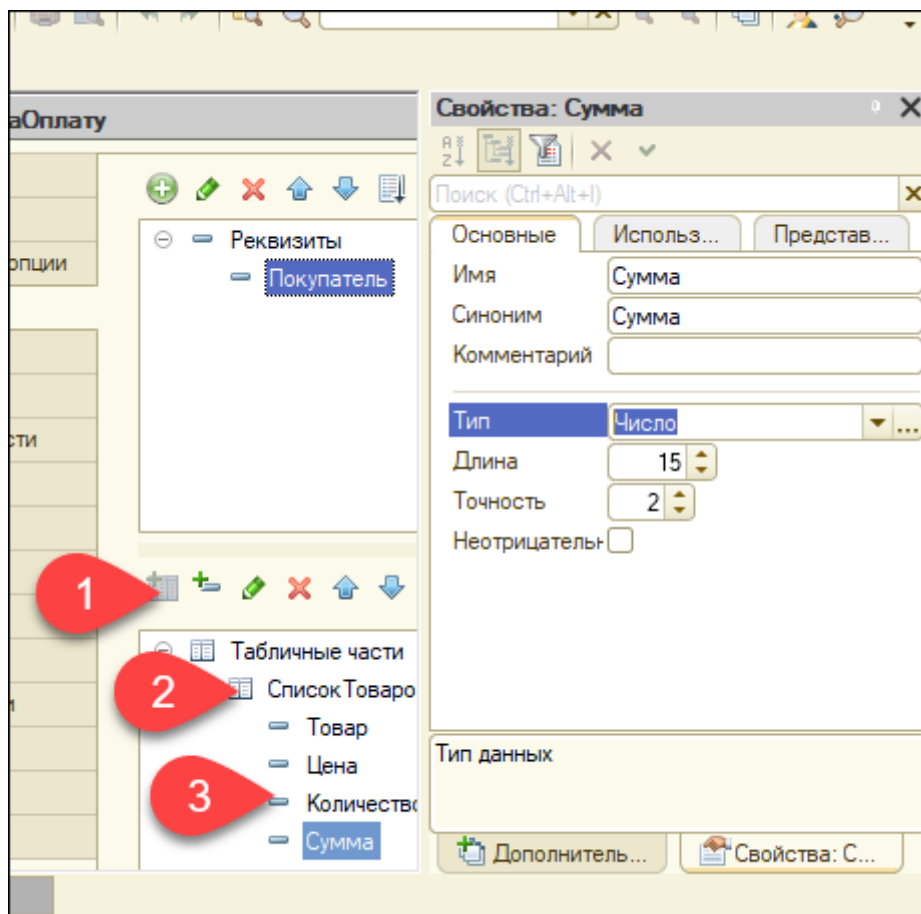
Приступим к созданию документа "Счет на оплату"





Добавим реквизит агрегатного типа "Покупатель". Пользователь будет производить выбор покупателя из соответствующего справочника. В самом реквизите будет храниться ссылка на покупателя.

Аналогичным образом добавим табличную часть "СписокТоваров". В табличной части добавим реквизит "Товар", так же с типом значения СправочникСсылка, но уже "Товары".



И реквизиты с типом "Число": Цена, Количество, Сумма.

Запустим систему и посмотрим на поведение формы, которую сгенерирует система.

Покупатели Счет на оплату Товары

← → **Счет на оплату (создание) *** ⋮ ×

Провести и закрыть Записать Провести Еще ▾

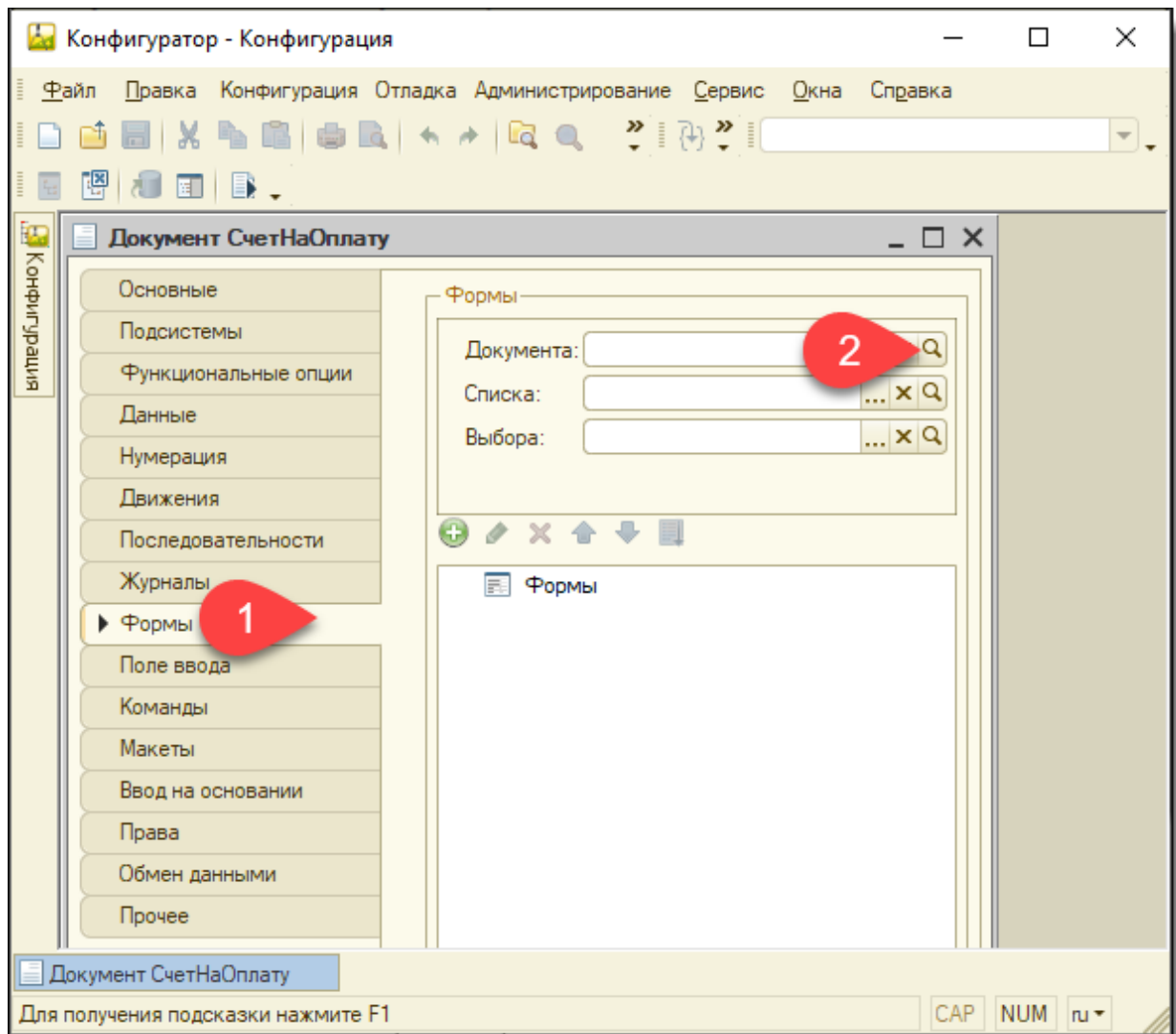
Добавить ↑ ↓ Поиск (Ctrl+F) × Еще ▾

N	Товар	Цена	Количество	Сумма
1	Ложка	10,00	10,000	100,00

Текущие вызовы: 1 Накопленные вызовы: 60

Работа с формой

Как мы убедились в форме не рассчитывается сумма по строке. Придется создавать форму самостоятельно и описывать расчет суммы.



Откроем в Конфигураторе закладку "Формы" в палитре свойств документа и создадим новую форму. При создании формы открывается конструктор форм.

Конструктор формы документа

Выберите тип формы:

- ☒ Форма документа
- ☐ Форма списка документа
- ☐ Форма выбора документа
- ☐ Произвольная форма

☒ Назначить форму основной

Имя:

Синоним:

Комментарий:

< Назад Далее > Готово Отмена Справка

Выбрав тип формы "Форма документа", мы определяем поведение формы, какие стандартные команды будут доступны. В нашем случае будут доступны команды "Записать", "Заккрыть" и прочие.

Конструктор формы документа

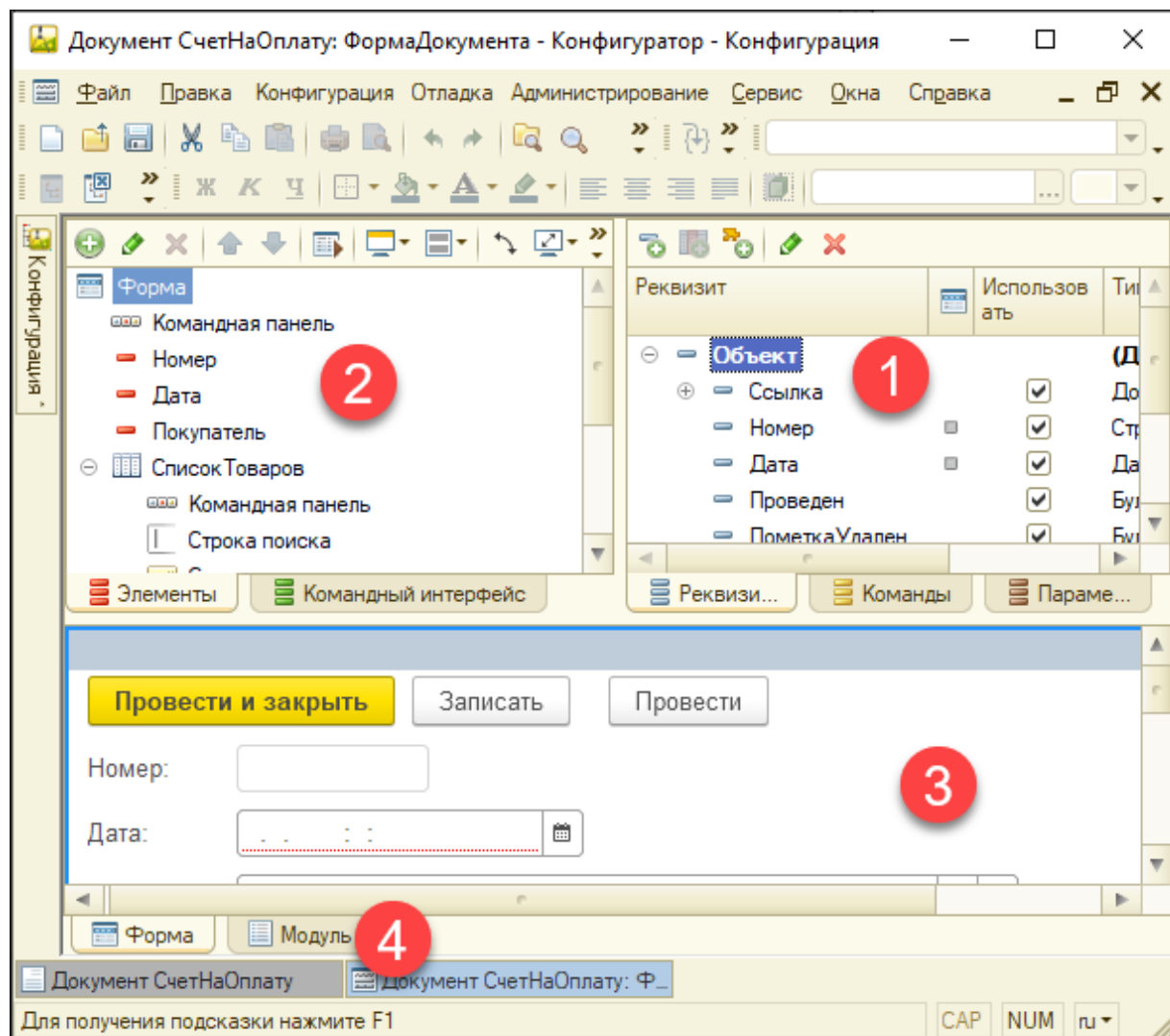
Располагать элементы управления в 1 колонку(и)

Реквизиты

- ☐ Ссылка
- ☒ **Номер**
- ☒ **Дата**
- ☐ Проведен
- ☐ ПометкаУдаления
- ☒ **Покупатель**
- ☒ **Список Товаров**
 - ☒ **НомерСтроки**
 - ☒ **Товар**
 - ☒ **Цена**
 - ☒ **Количество**
 - ☒ **Сумма**

< Назад Далее > Готово Отмена Справка

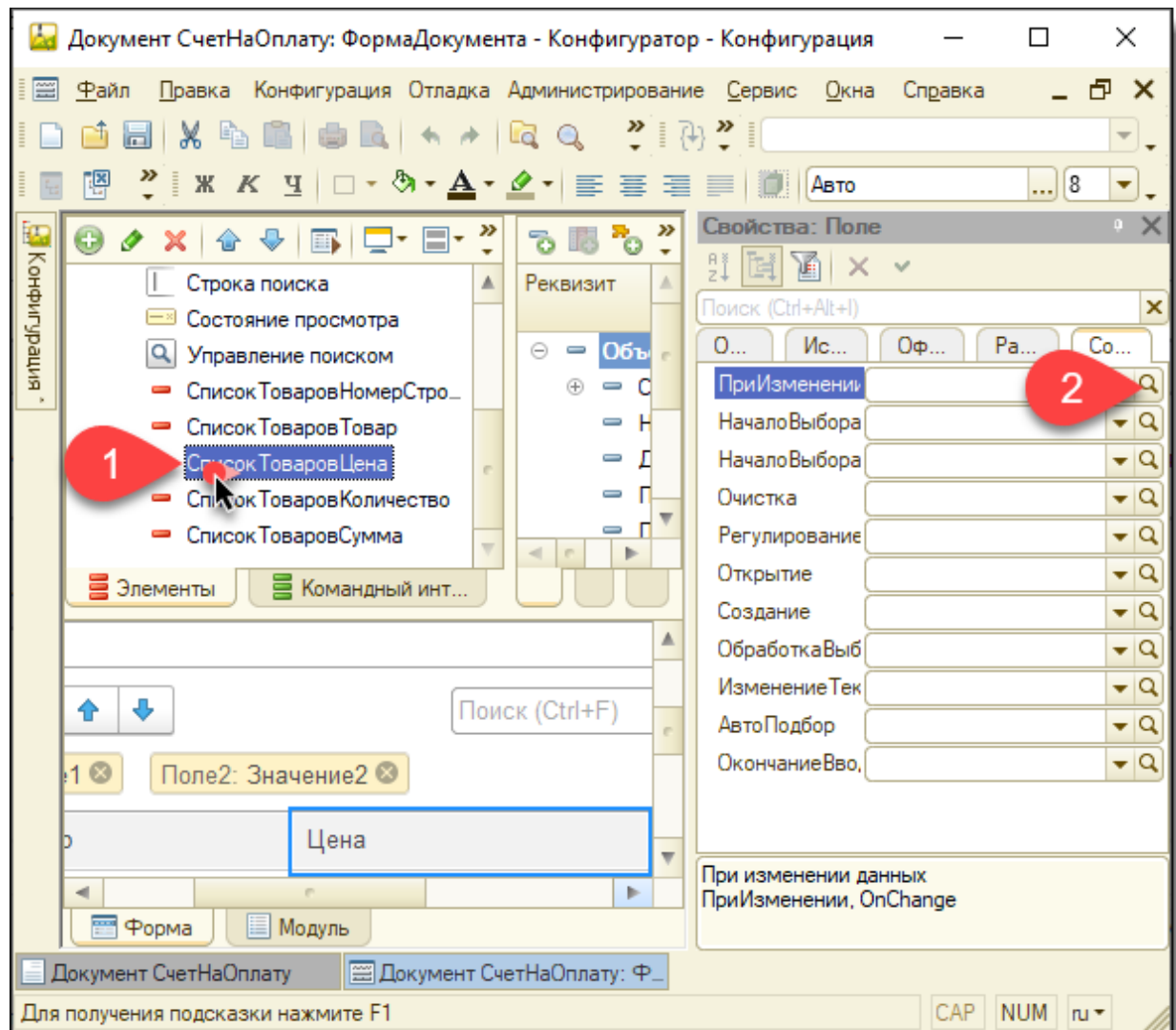
На втором шаге конструктора необходимо отметить какие реквизиты должны быть отображены на форме.



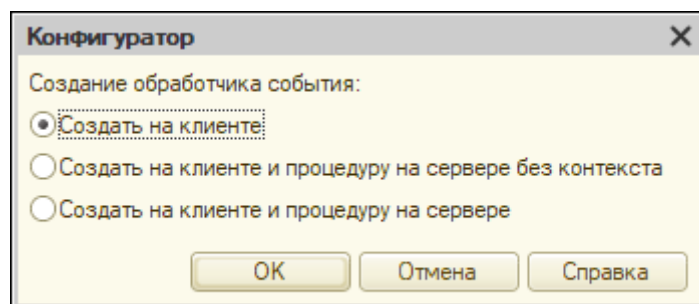
В итоге откроется редактор формы. В редакторе форм можно выделить 4 области:

1. Список реквизитов, то есть данные, которые могут быть отображены на форме
2. Элементы формы, то как данные будут отображены на форме
3. Примерный вид формы, то как пользователь будет видеть форму
4. Вкладки для перехода из графического редактора к модулю формы и обратно

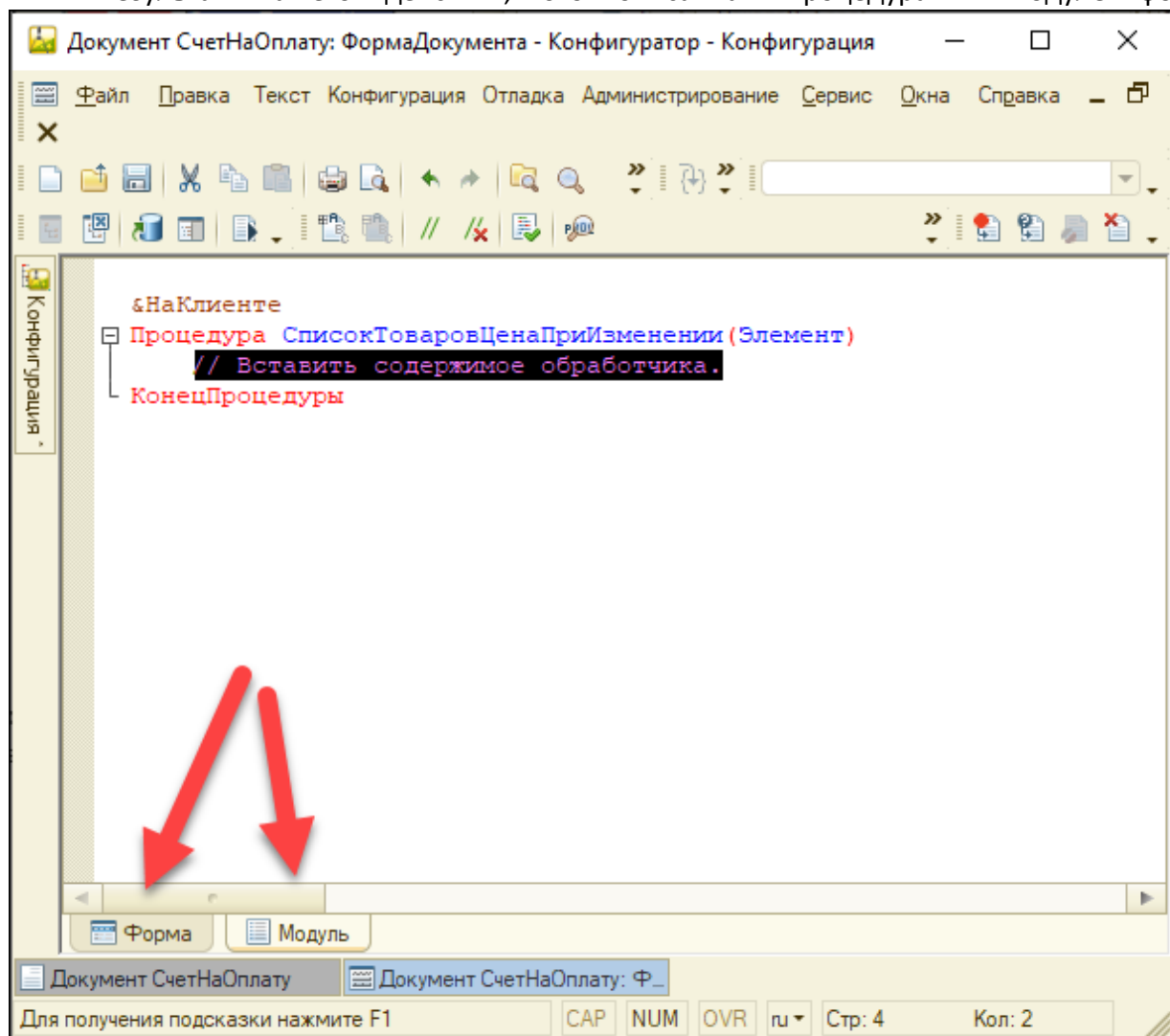
Откроем свойства элемента СписокТоваровЦена и найдем события. Нам нужно событие "ПриИзменении", так как именно при изменении полей "Цена" и "Количество" нам нужно рассчитать сумму.



При выборе обработчика события появится вопрос, где описывать исполнение обработчика событий? Так как мы просто хотим помножить две цифры и к базе данных обращаться нам не надо, мы выберем "На клиенте".



Результат нашего действия, это описанная процедура в модуле формы.

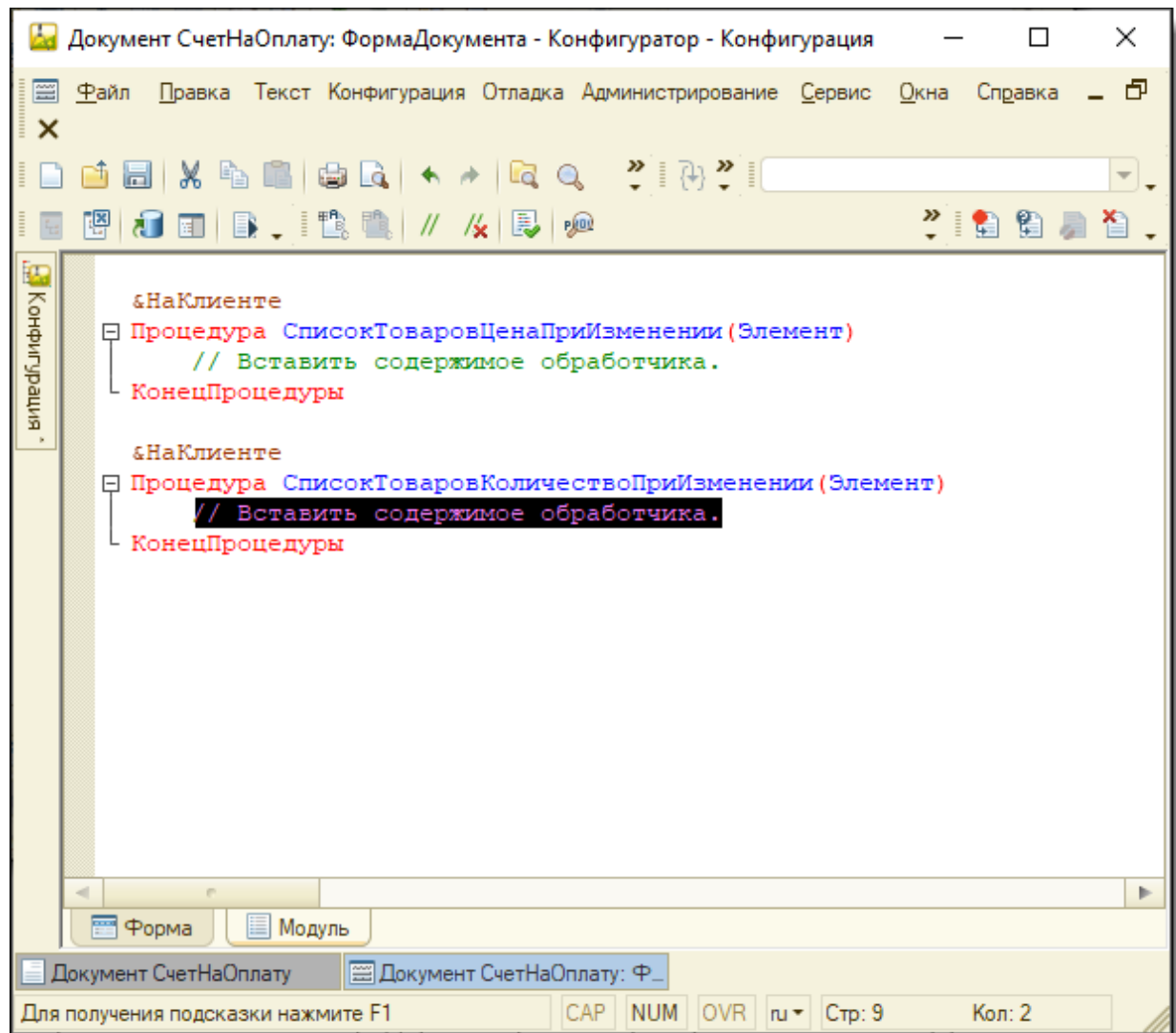


Переключаться между редактором и модулем можно вкладками.

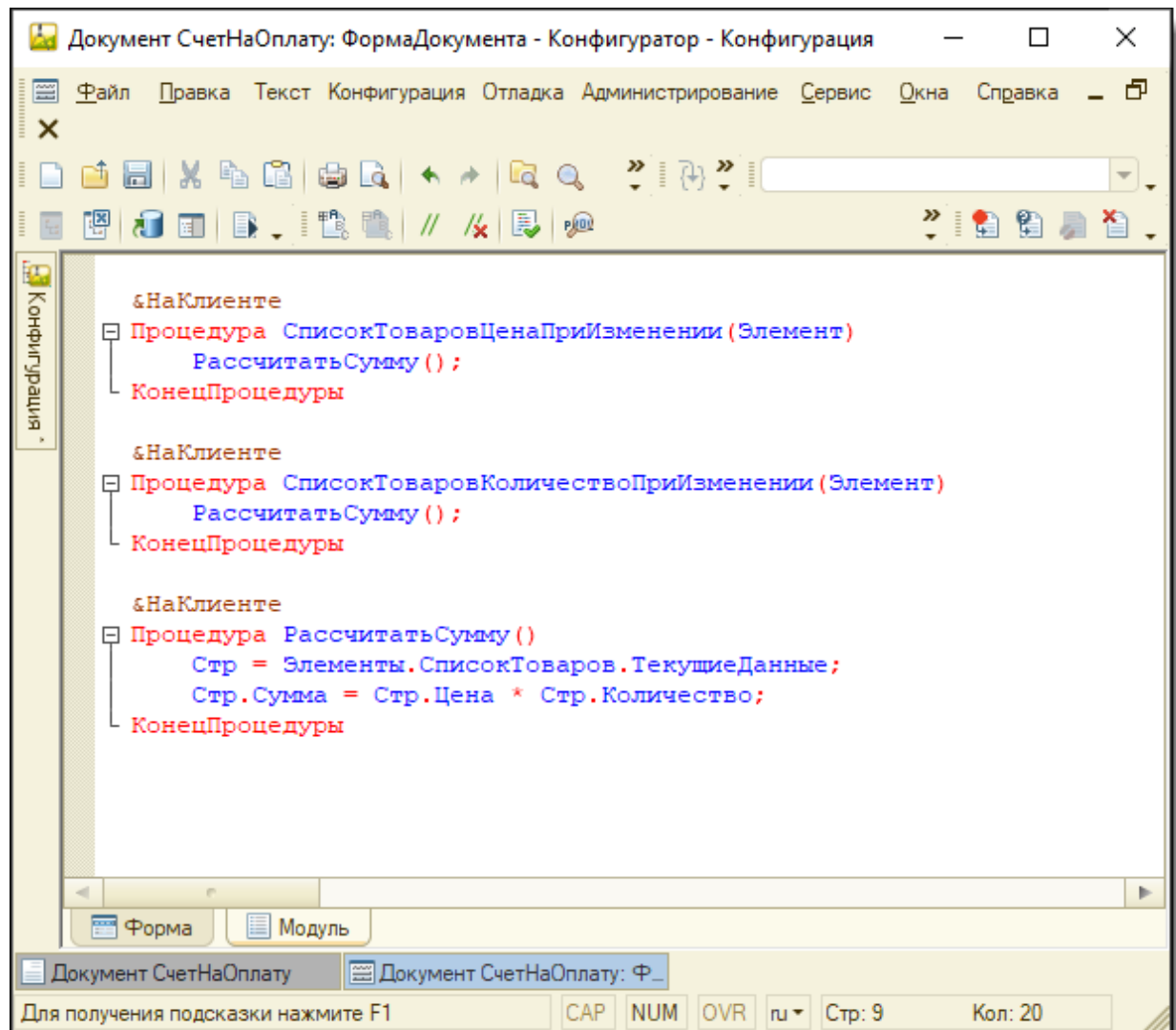
Разработка в системе 1С:Предприятие (1C:Enterprise) возможна как на русском так и на английском языках. Причем один оператор может начинаться на русском, а заканчиваться на английском. (Если ... Тогда ... EndIf)

Вернемся в форму и проделаем тоже самое с полем "Количество".

В итоге модуль примет следующий вид.



Так как логика расчета суммы у нас одинаковая опишем ее в третьей процедуре, которую наберем уже самостоятельно.



Для удобства создали переменную "Стр", которая ссылается на данные строки, которая выделена у пользователя, активна.

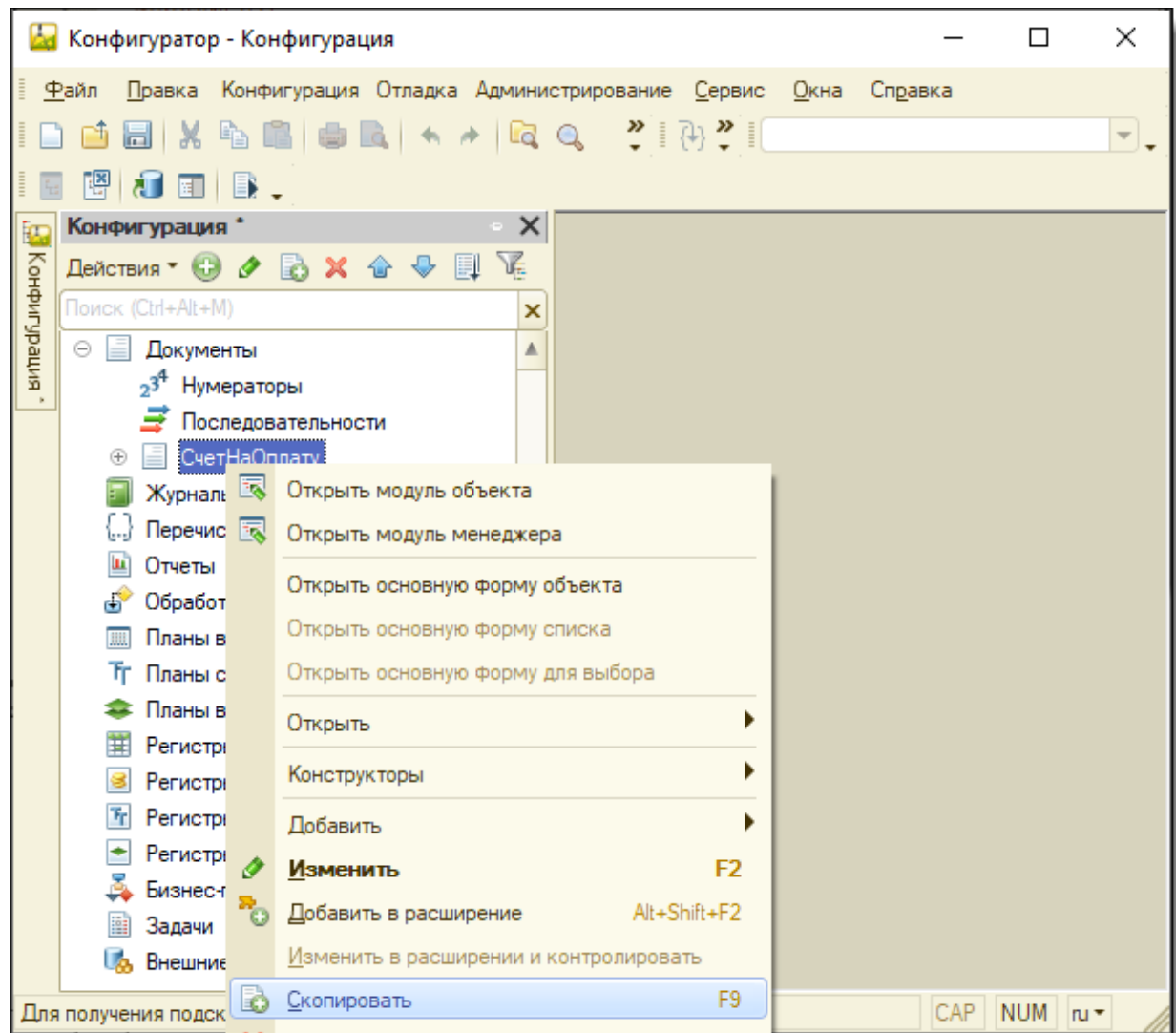
Из процедур обработчиков событий вызываем процедуру РассчитатьСумму.

Все, можно запустить систему и проверить корректность расчета суммы.

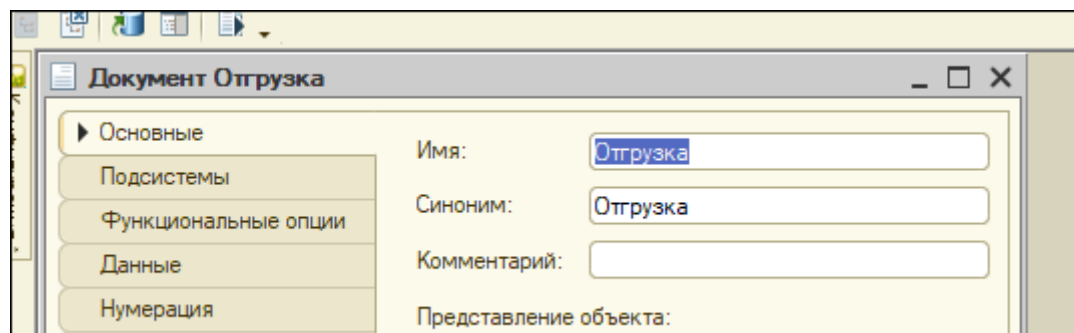
Ввод на основании

Теперь время создать документ отгрузки.

Так как по своей структуре документ отгрузки будет повторять счет, то мы создадим его копированием.



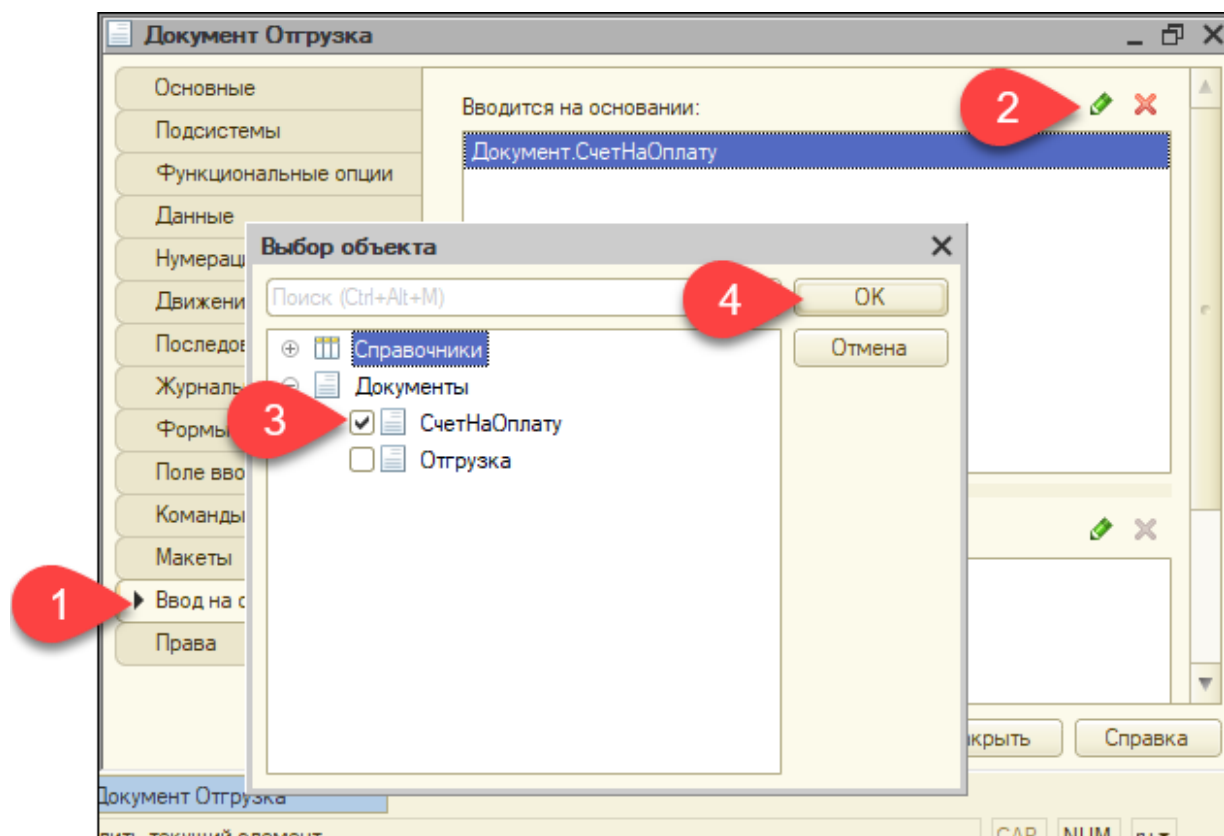
Переименуем документ.



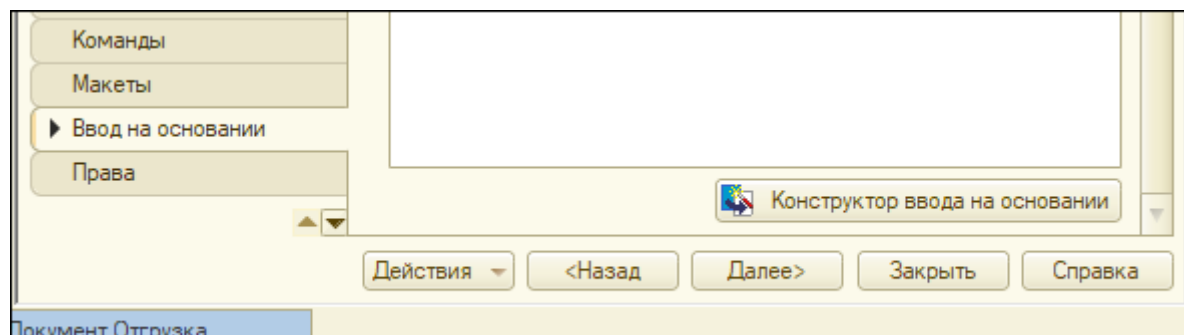
Естественно, самый популярный сценарий в компании, которая занимается продажами это "Выставили счет, подождали оплату, отгрузили все что было в счете".

Поэтому реализуем возможность копирования данных из документа Счет в отгрузку. Эта техника называется "Ввод на основании".

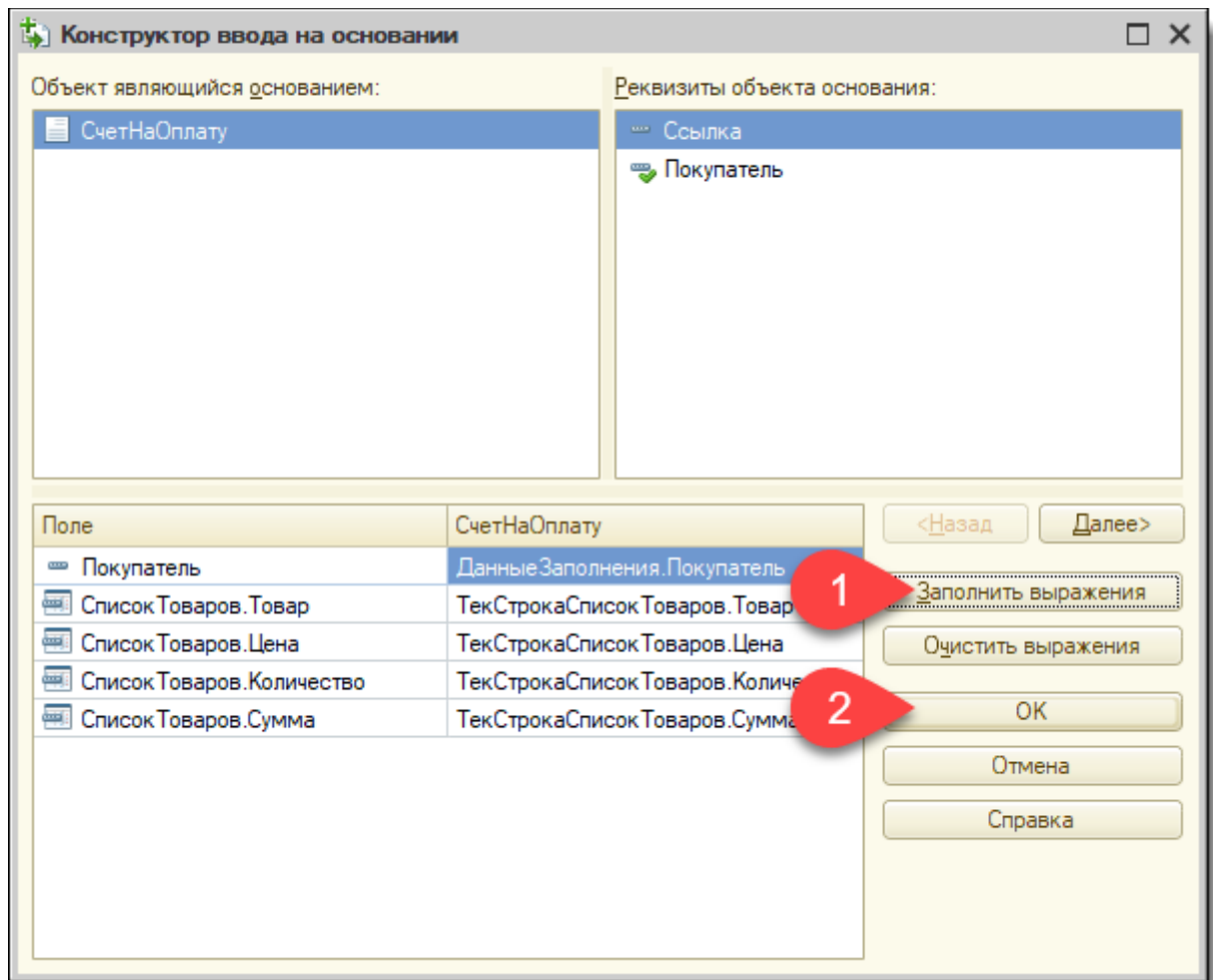
На вкладке "Ввод на основании" укажем на основании какого документа будет вводиться документ отгрузки.



После этого нужно описать программный код для заполнения нового документа отгрузки на основании счета, но у нас для этого есть конструктор, который все сделает за нас.



Запустим конструктор, заполним соответствия полей. Так как документ мы копируем, то и имена реквизитов и типы значений у нас совпадут.



В итоге работы конструктора у нас откроется модуль объекта, в котором будет описана процедура "ОбработкаЗаполнения". Это процедура-обработчик события ввода на основании.

```
Процедура ОбработкаЗаполнения(ДанныеЗаполнения, СтандартнаяОбработка)
//{{_КОНСТРУКТОР_ВВОД_НА_ОСНОВАНИИ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную и
Если ТипЗнч(ДанныеЗаполнения) = Тип("ДокументСсылка.СчетНаОплату")
// Заполнение шапки
Покупатель = ДанныеЗаполнения.Покупатель;
Для Каждого ТекСтрокаСписокТоваров Из ДанныеЗаполнения.СписокТоваров
    НоваяСтрока = СписокТоваров.Добавить();
    НоваяСтрока.Количество = ТекСтрокаСписокТоваров.Количество;
    НоваяСтрока.Сумма = ТекСтрокаСписокТоваров.Сумма;
    НоваяСтрока.Товар = ТекСтрокаСписокТоваров.Товар;
    НоваяСтрока.Цена = ТекСтрокаСписокТоваров.Цена;
КонецЦикла;
КонецЕсли;
//}}_КОНСТРУКТОР_ВВОД_НА_ОСНОВАНИИ
КонецПроцедуры
```

В процедуре проверяется на основании какого документа вводится текущий, и производится заполнение реквизитов.

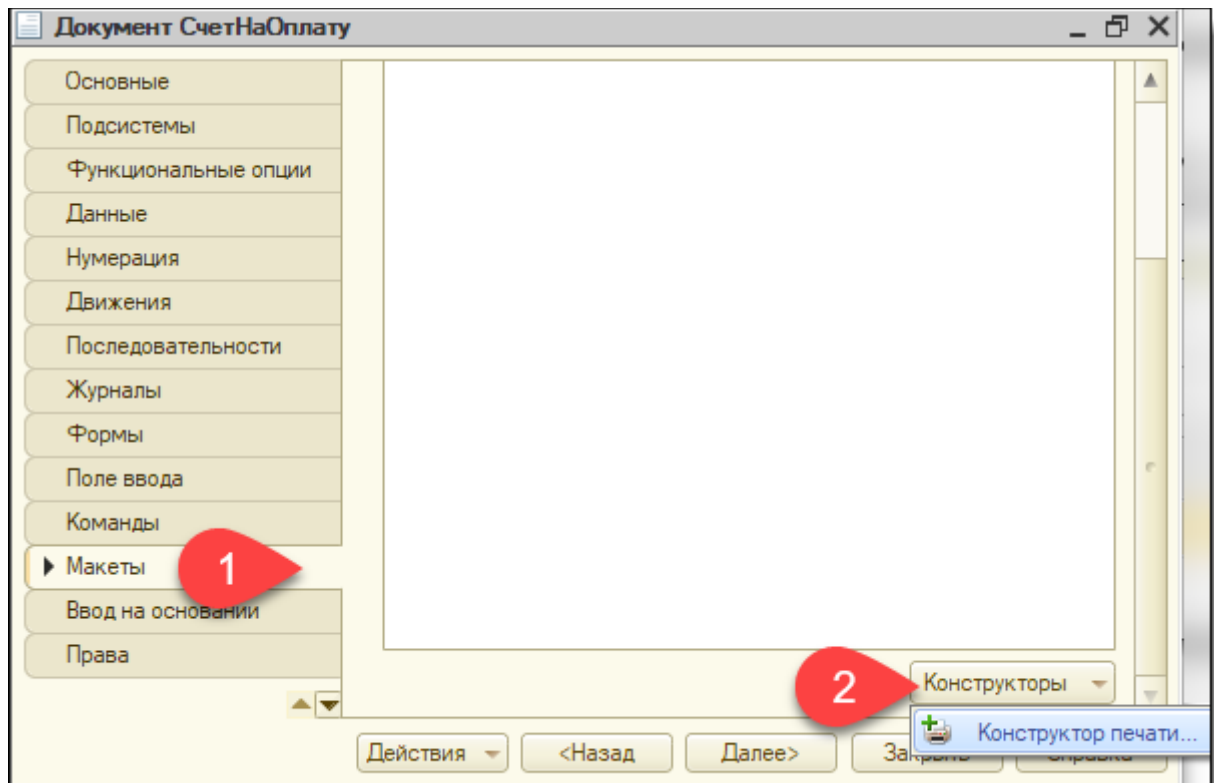
В режиме 1С:Предприятие можно воспользоваться созданным механизмом. В форме списка документов "Счет" появилась команда (кнопка) "Создать на основании".

Дата	Отгрузка	Покупатель
22.01.2020 10:06:30	000000001	Мир

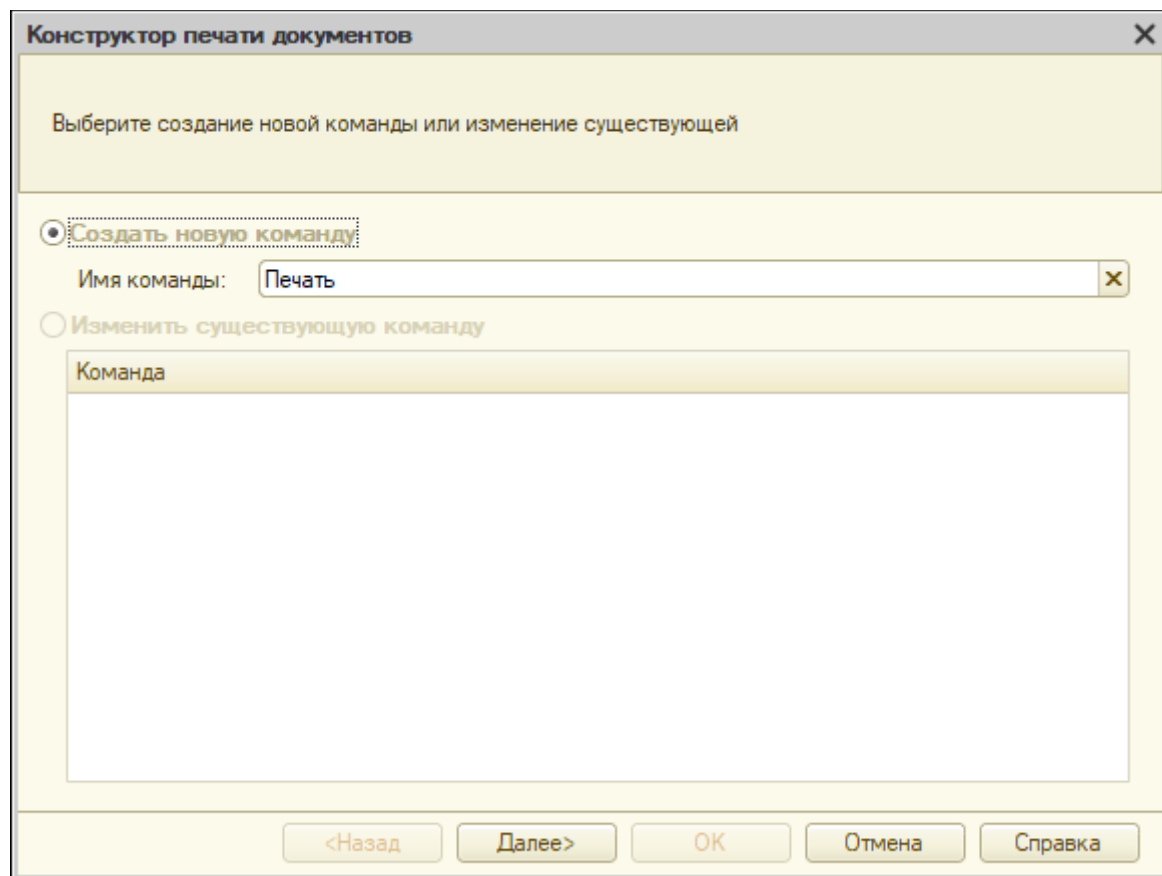
Печать

Документы созданы. Теперь нужно создать печатную форму.

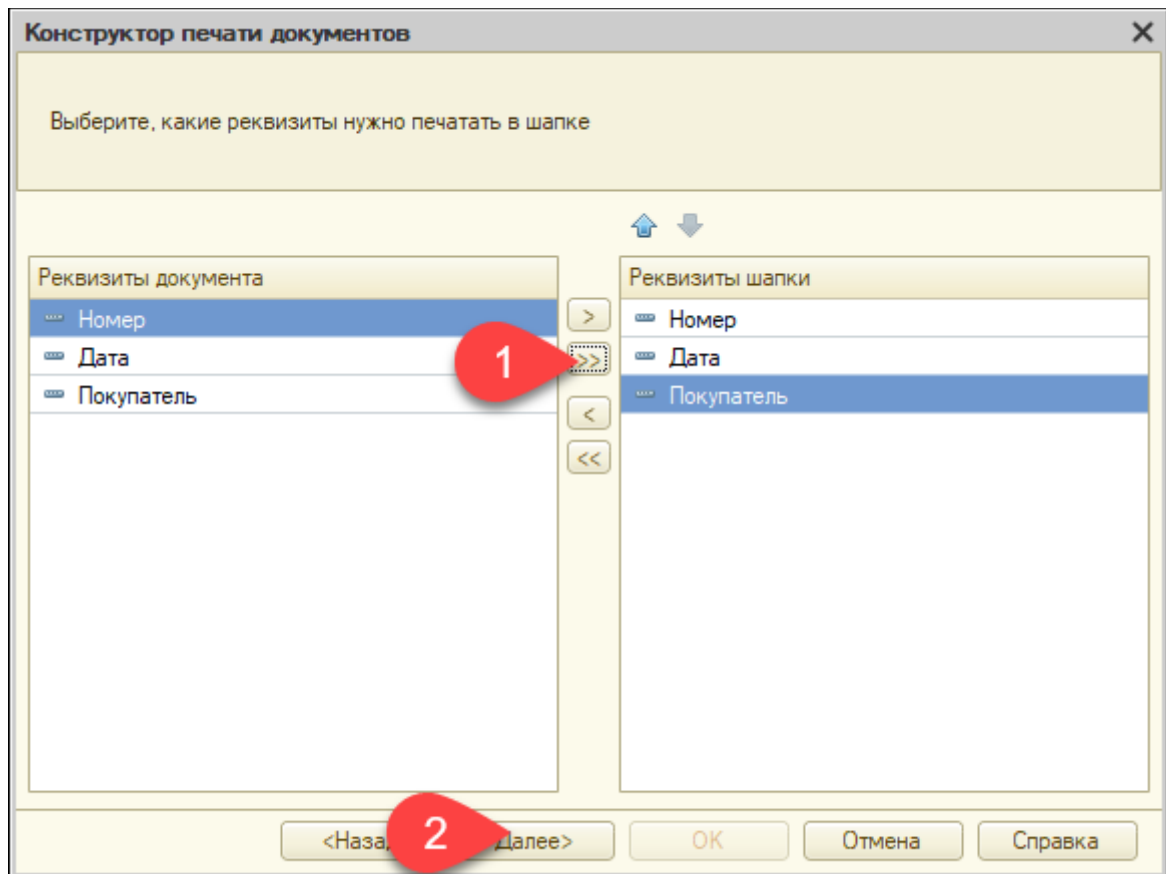
Для этого в системе также есть конструктор. Откроем свойства документа "Счет" и найдем вкладку "Макеты".



Запустим конструктор печати. Конструктор создаст макет, это заготовка табличного документа, которую можно подправить после работы конструктора тем самым приведя вид печатного документа к тому, который требуется организации. Также конструктор создаст команду печати и опишет программный модуль.

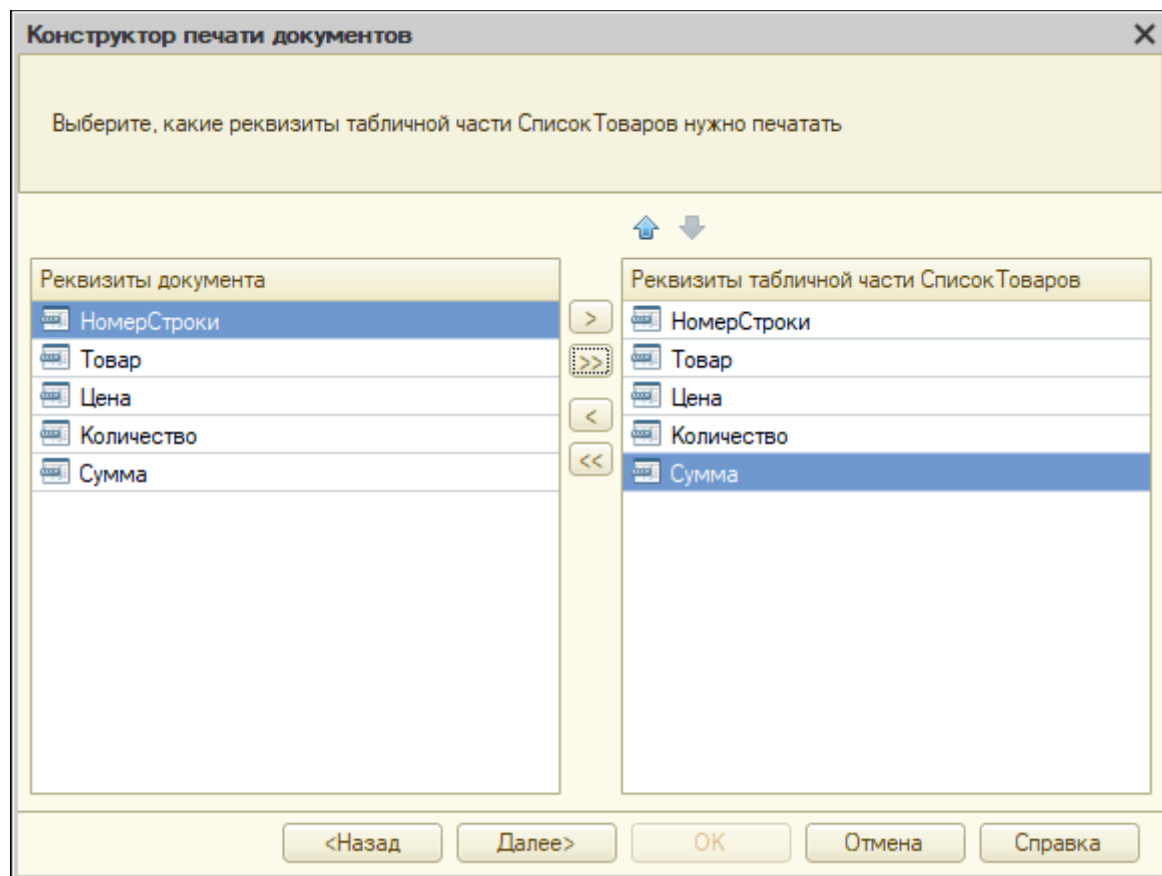


Команду можно назвать произвольно, но оставим "Печать".

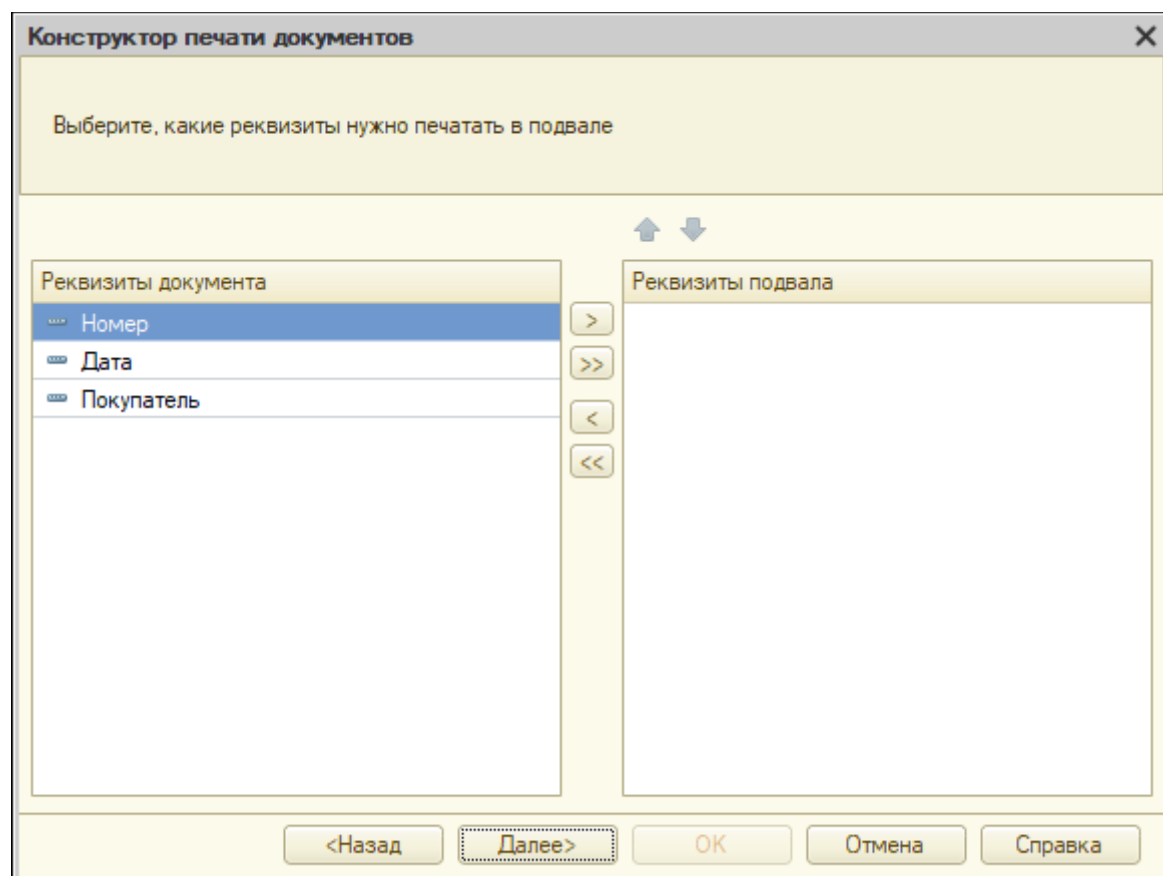


Перенесем все реквизиты для печати в шапку.

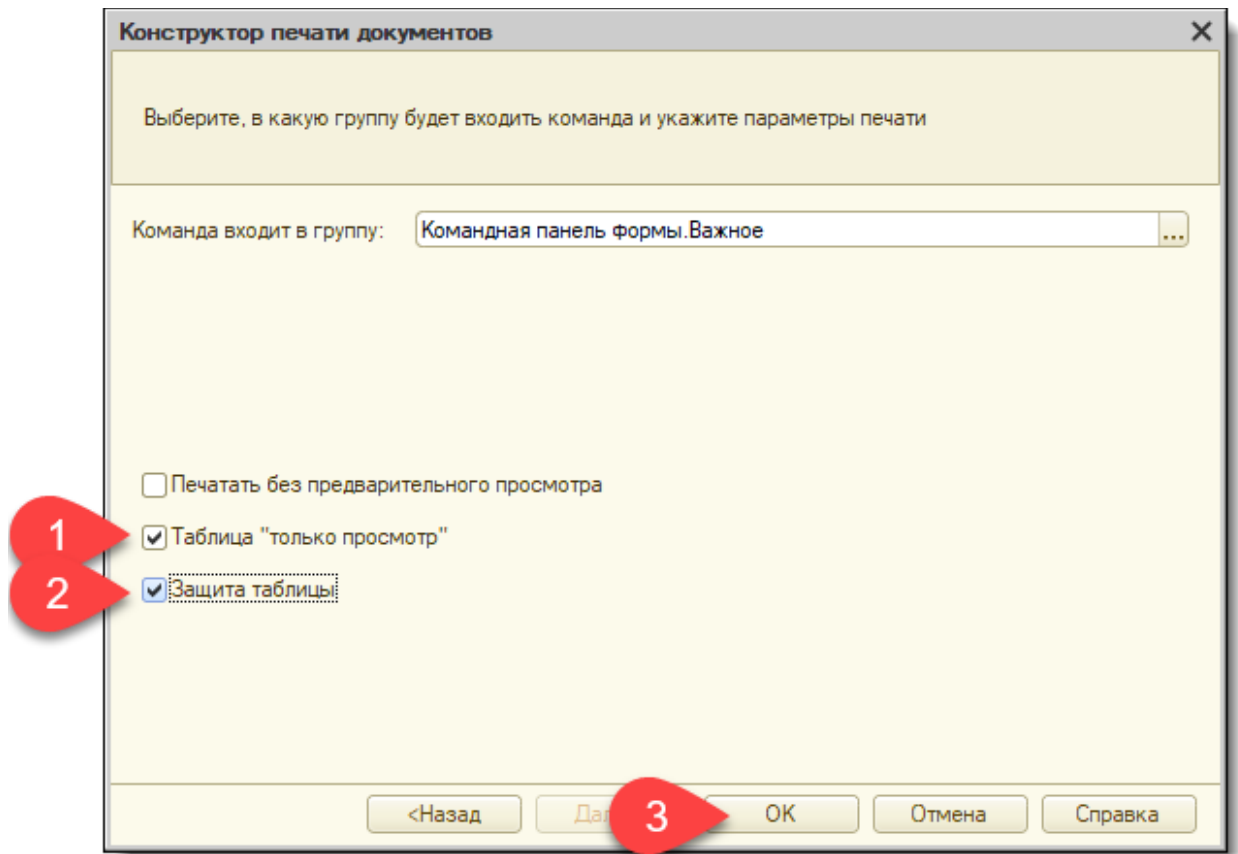
По аналогии перенесем все реквизиты табличной части...



В подвале нам печатать нечего, пропустим шаг.

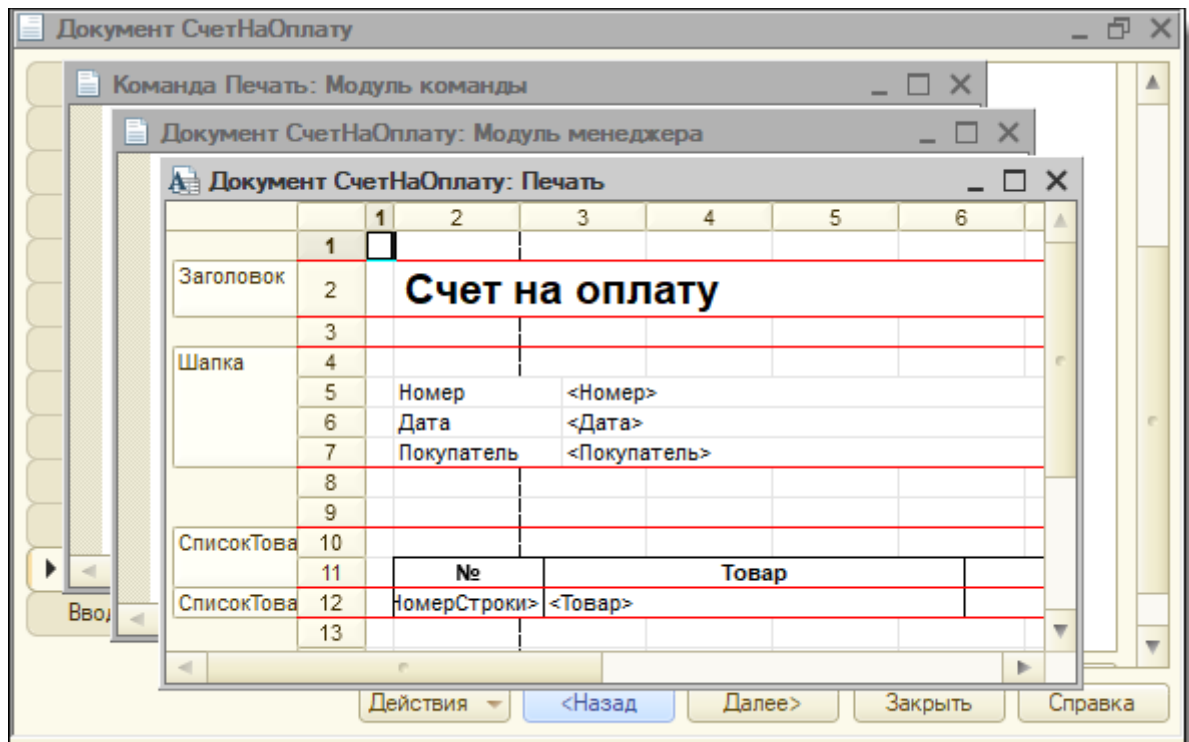


Установим защиту от изменения табличного документа.



И завершим работу конструктора.

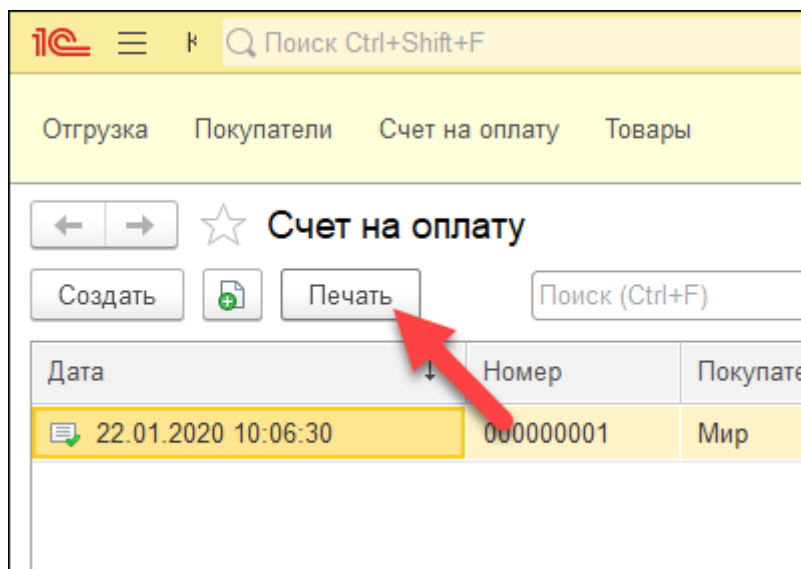
В результате откроются вновь созданные: макет, модуль менеджера и модуль команды.



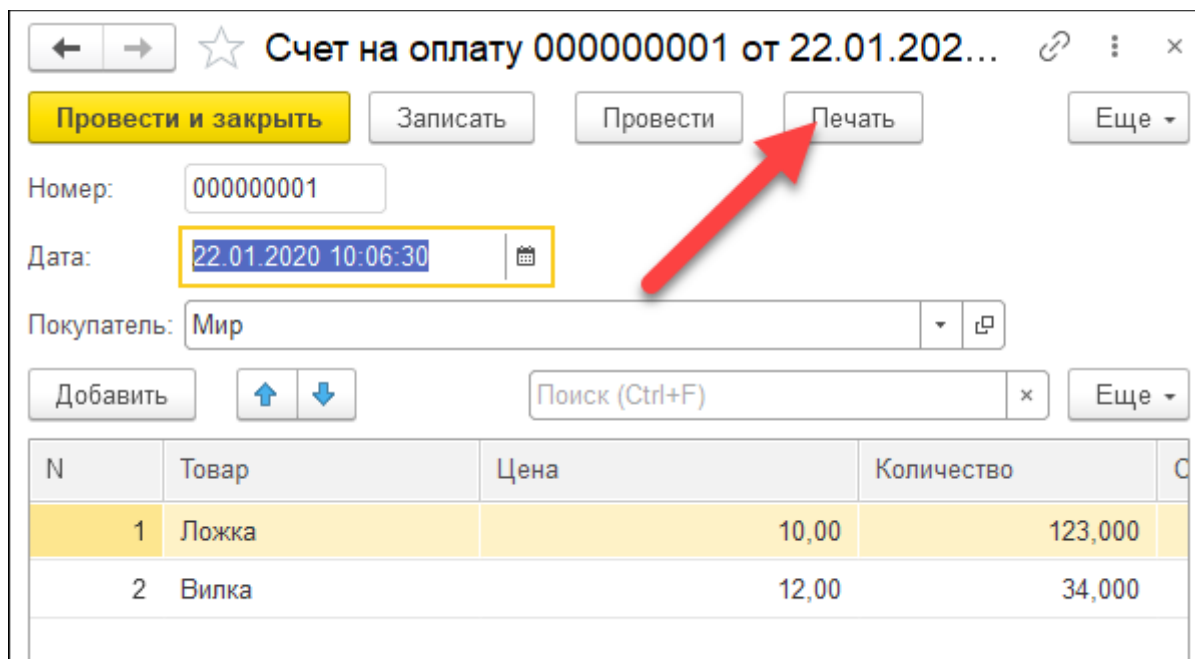
При желании можно поправить макет.

Запустим систему и попробуем напечатать документ.

Команда появится как в форме списка документа...



... так и в форме самого документа.



← → ☆ Счет на оплату 000000001 от 22.01.202... 🔗 ⋮ ×

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

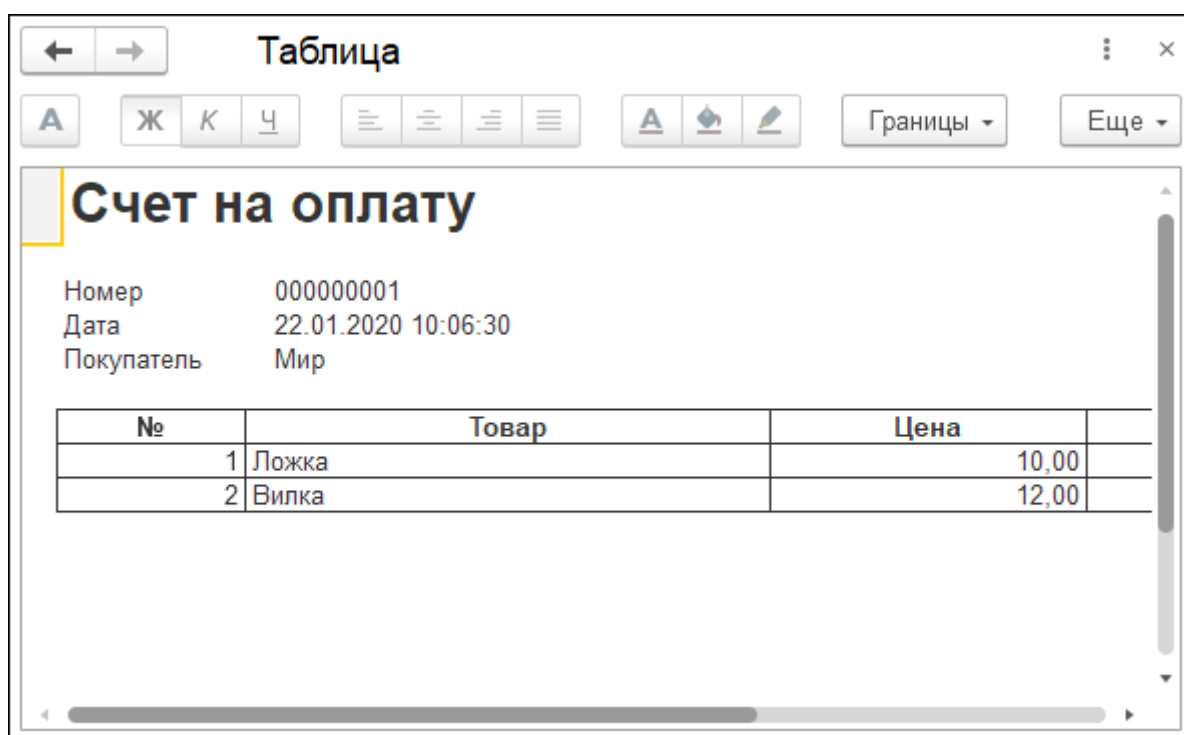
Номер: 000000001

Дата: 22.01.2020 10:06:30 📅

Покупатель: Мир ▾ 📄

Добавить ⬆ ⬇ Поиск (Ctrl+F) × Еще ▾

N	Товар	Цена	Количество	С
1	Ложка	10,00	123,000	
2	Вилка	12,00	34,000	



← → Таблица ⋮ ×

A Ж К Ч ⋮ ⋮ ⋮ ⋮ A 🖨 🖋 Границы ▾ Еще ▾

Счет на оплату

Номер 000000001
Дата 22.01.2020 10:06:30
Покупатель Мир

№	Товар	Цена	
1	Ложка	10,00	
2	Вилка	12,00	

Готовый табличный документ можно распечатать, или сохранить во множество популярных форматов, включая xlsx и pdf.

Регистры

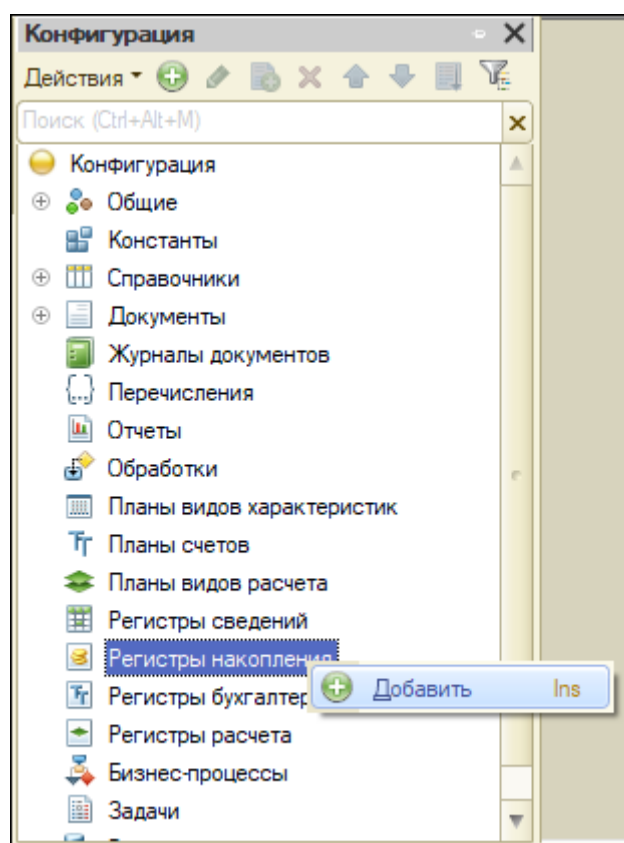
Ввод и хранение информации о продажах мы реализовали, нужно думать об отчетах.

Конечно можно и анализируя документы построить отчеты по продажам, но это не эффективно. Почему? В документах данные не агрегированы, при добавлении новых видов документов придется переписывать отчеты. Мы воспользуемся специальным механизмом, который как раз предназначается для накопления показателей и позволяет абстрагироваться от структуры и видов документов. Это регистры.

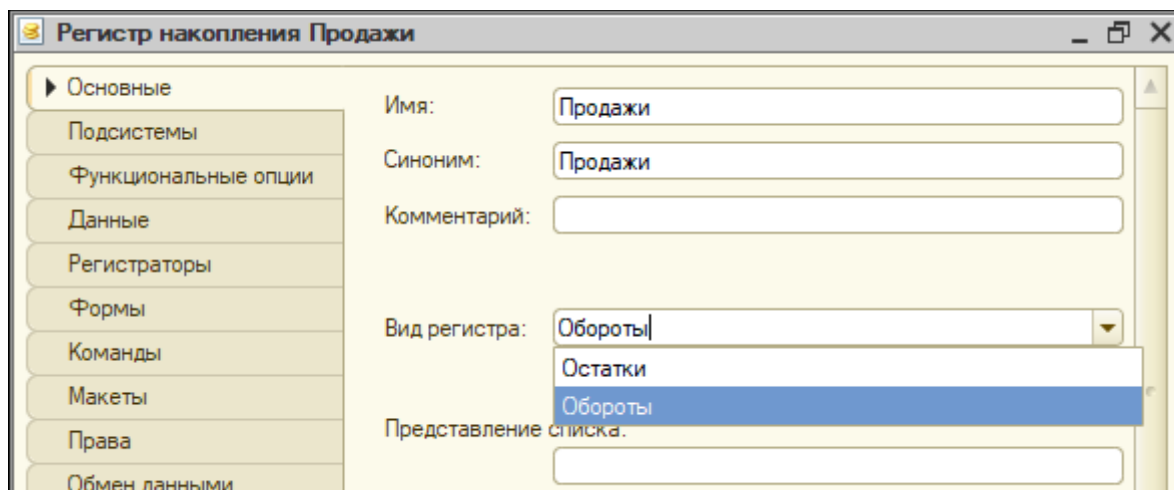
Регистры по своей природе — это таблицы которые хранят данные в максимально агрегированном виде и позволяют извлекать информацию о хранящихся в них показателях максимально эффективно.

Для нашей задачи воспользуемся регистром накопления оборотов. Будем хранить данные о продажах в суммовом и количественном вариантах в разрезе товаров и покупателей.

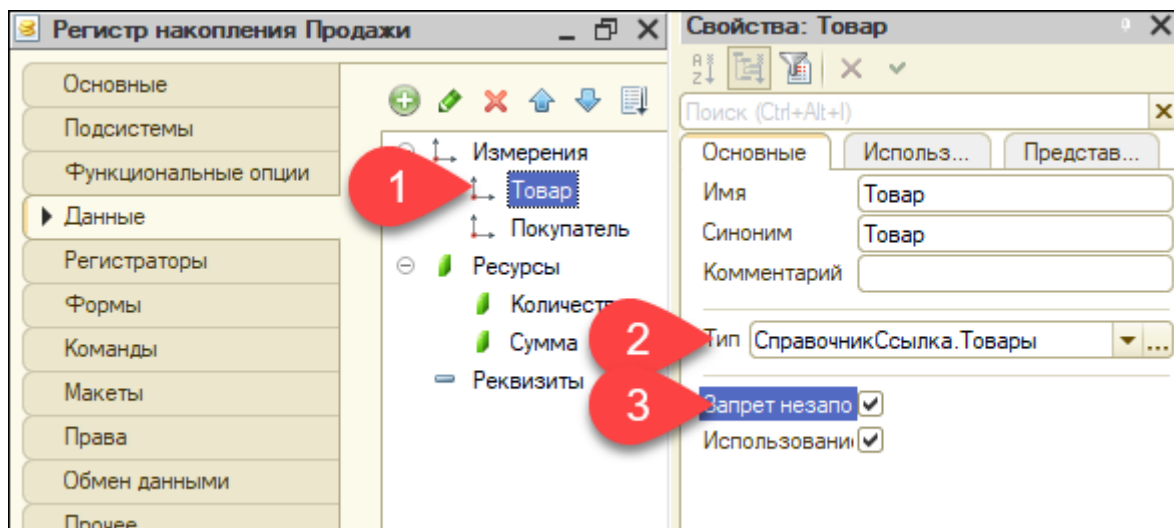
Создадим регистр накопления "Продажи".



Вид регистра укажем "Обороты".

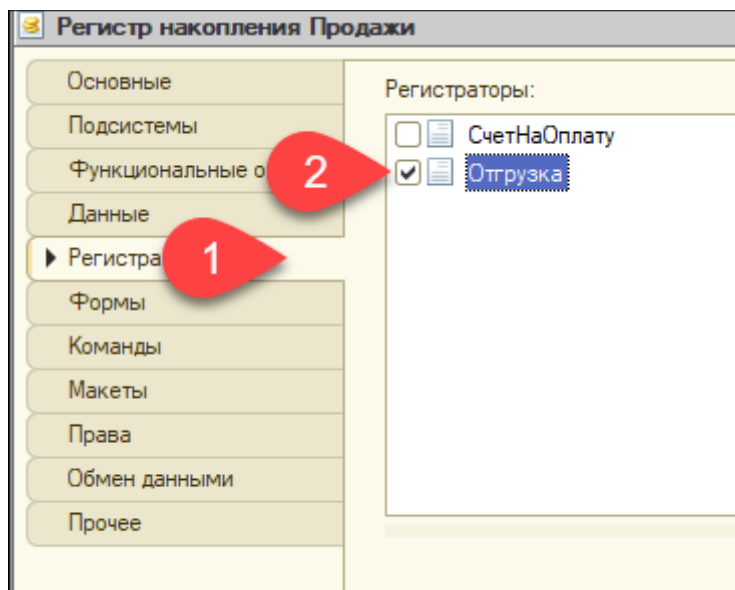


На закладке "Данные" определим, что будет хранить регистр (ресурсы) и в каком разрезе (измерения).



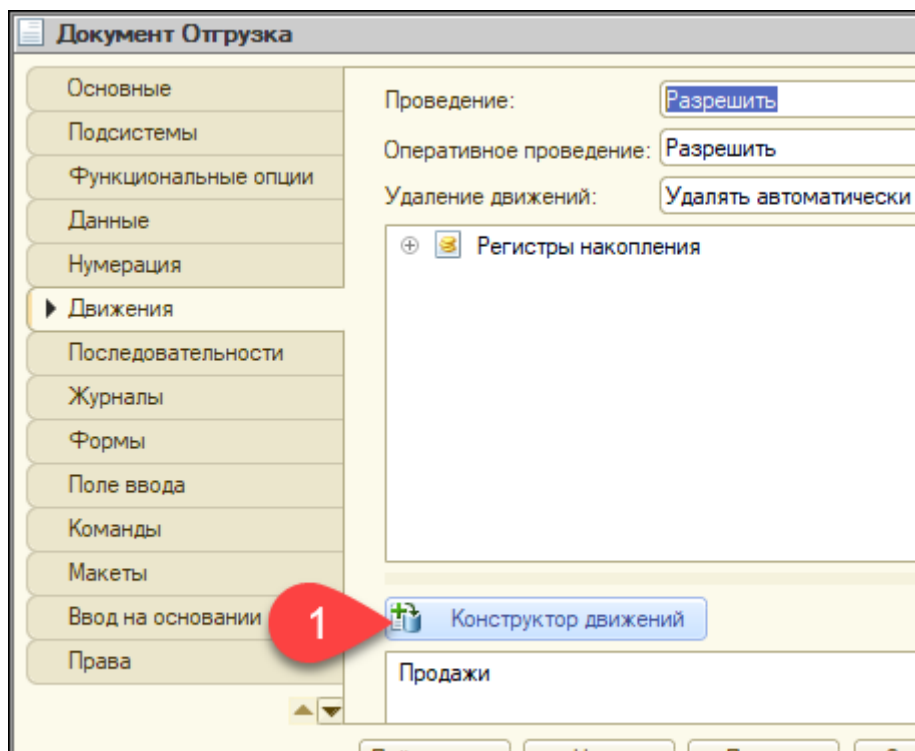
Измерения "Товар" и "Покупатель" имеют ссылочный тип, СправочникСсылка.Товары и СправочникСсылка.Покупатели соответственно. Количество и Сумма – это числа.

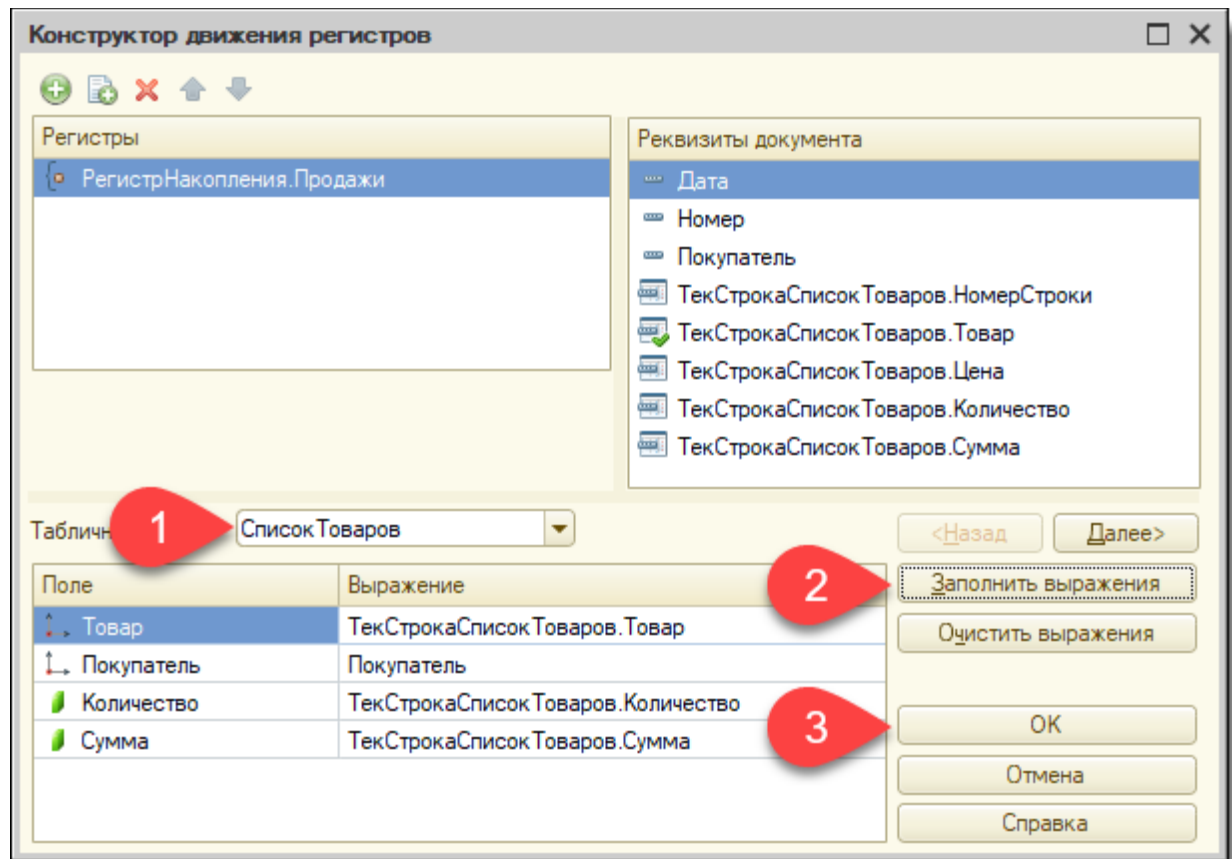
Для формирования записей в регистре укажем какие документы имеют на это право. На вкладке "Регистраторы" отметим документ "Отгрузка".



Для формирования записей в регистре необходимо описать событие "Проведение" в модуле документа. Это событие наступает после записи документа в транзакции записи. Для формирования программного кода также воспользуемся конструктором.

В свойствах документа "Отгрузка" на вкладке "Движения" запустим конструктор движений.





Заполним соответствия полей и завершим работу конструктора. Перед нами откроется модуль документа с описанной процедурой, которая сформирует движения в регистре.

```
⊕ Процедура ОбработкаЗаполнения(ДанныеЗаполнения, СтандартнаяОбработ
⊖ Процедура ОбработкаПроведения(Отказ, Режим)
    ///{} _КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
    // Данный фрагмент построен конструктором.
    // При повторном использовании конструктора, внесенные вручную

    // регистр Продажи
    Движения.Продажи.Записывать = Истина;
    Для Каждого ТекСтрокаСписокТоваров Из СписокТоваров Цикл
        Движение = Движения.Продажи.Добавить();
        Движение.Период = Дата;
        Движение.Товар = ТекСтрокаСписокТоваров.Товар;
        Движение.Покупатель = Покупатель;
        Движение.Количество = ТекСтрокаСписокТоваров.Количество;
        Движение.Сумма = ТекСтрокаСписокТоваров.Сумма;
    КонечЦикла;

    ///{} _КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
    КонечПроцедуры
```

В режиме 1С:Предприятие провести документ, чтобы он сформировал записи в регистре, можно кнопкой "Провести" или "Провести и закрыть".

1C Поиск Ctrl+Shift+F

Отгрузка Покупатели Счет на оплату Товары

Отгрузка (создание)

Провести и закрыть Записать Провести Еще ▾

Номер:

Дата: 22.01.2020 0:00:00

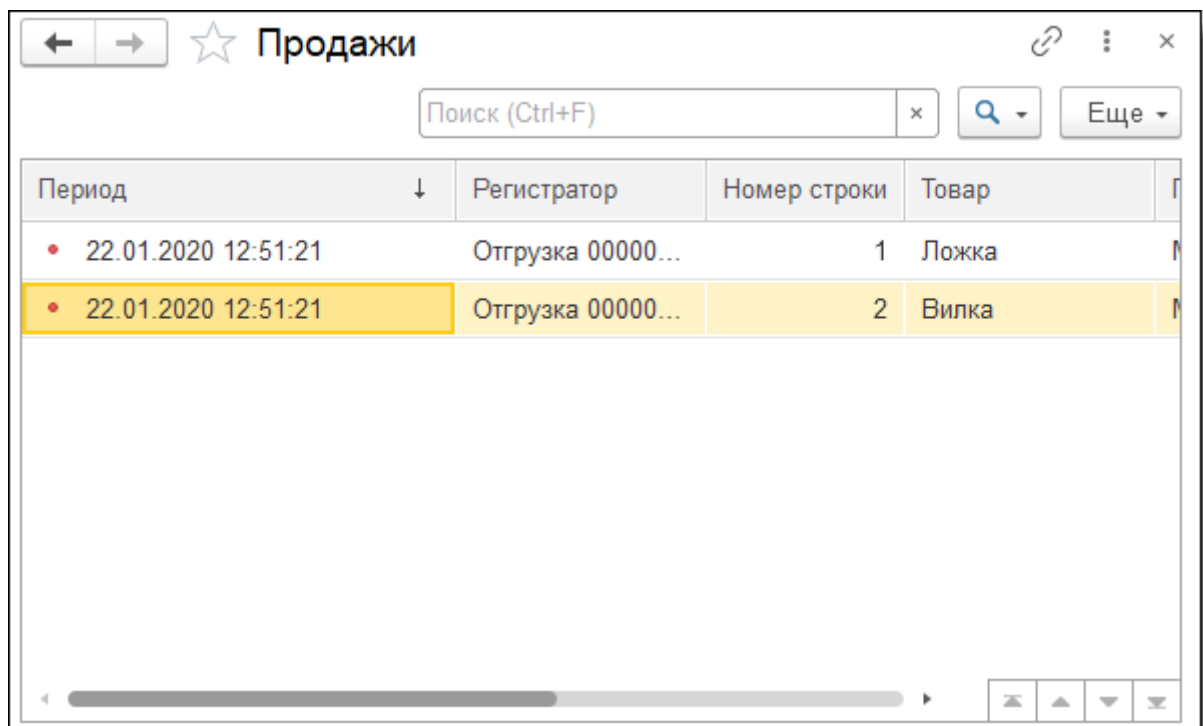
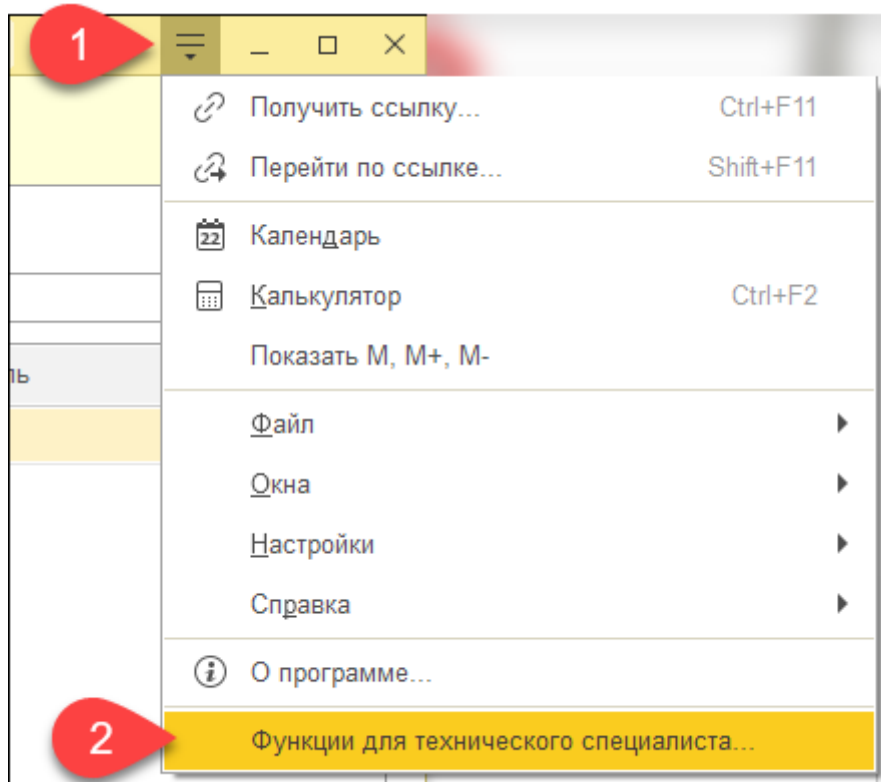
Покупатель: Мир ▾

Добавить ↑ ↓ Поиск (Ctrl+F) × Еще ▾

N	Товар	Цена	Количество	C
1	Ложка	10,00	123,000	
2	Вилка	12,00	34,000	

Текущие вызовы: 11 Накопленные вызовы: 33

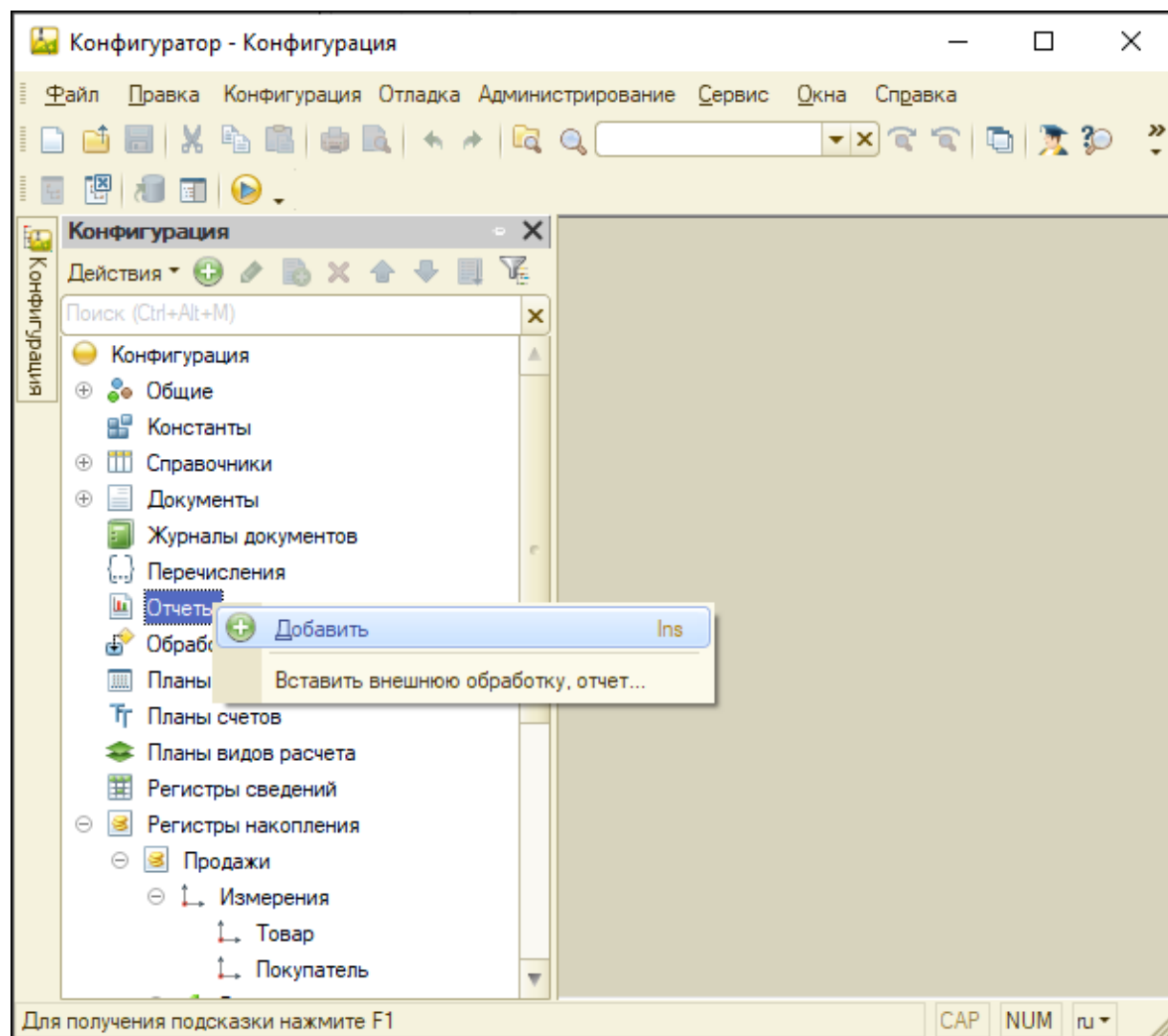
Посмотреть созданные записи в регистре можно через "функции для технического специалиста".

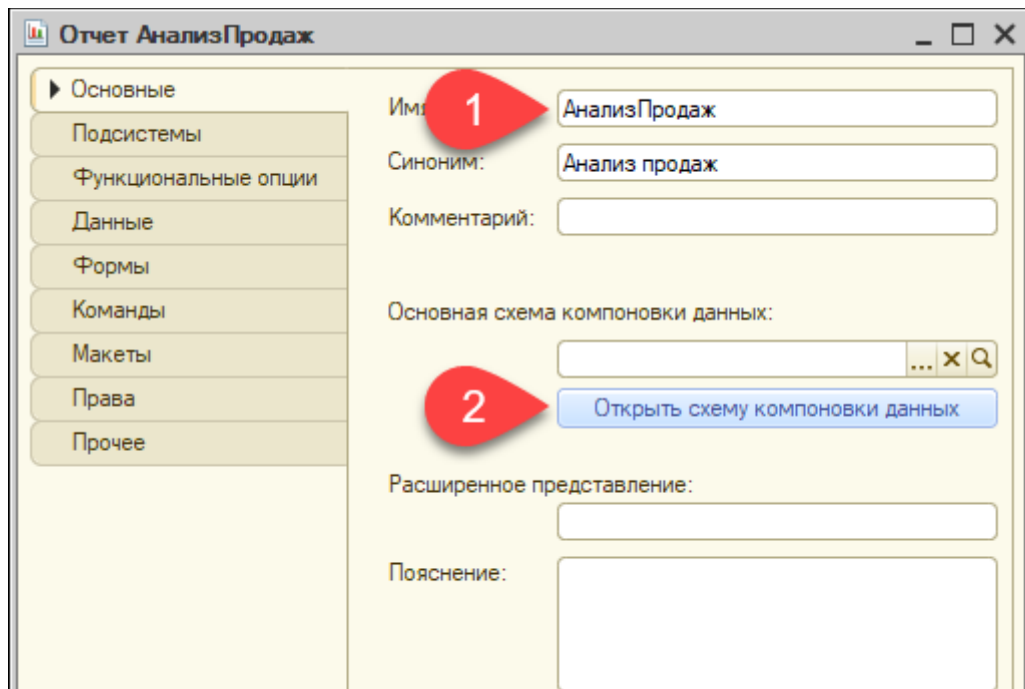


Отчеты

Когда данные для анализа мы собрали в регистре можно приступить к созданию отчетов.

Создадим отчет, показывающий продажи в разрезе покупателей.



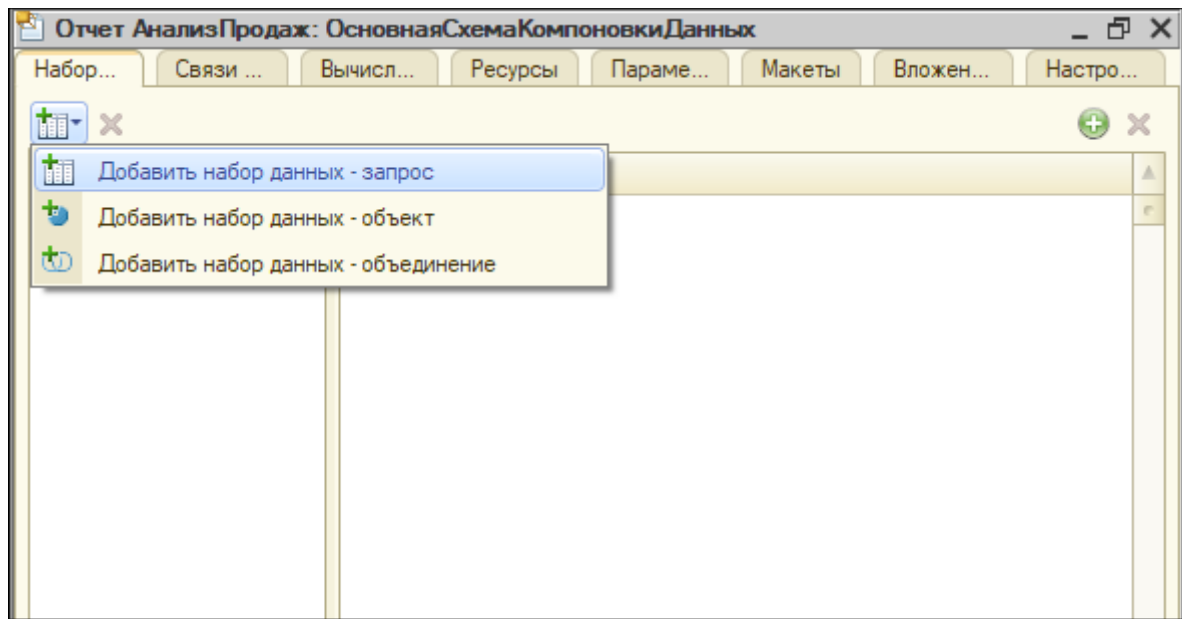


Воспользуемся механизмом декларативного описания отчетов "Схема компоновки данных".

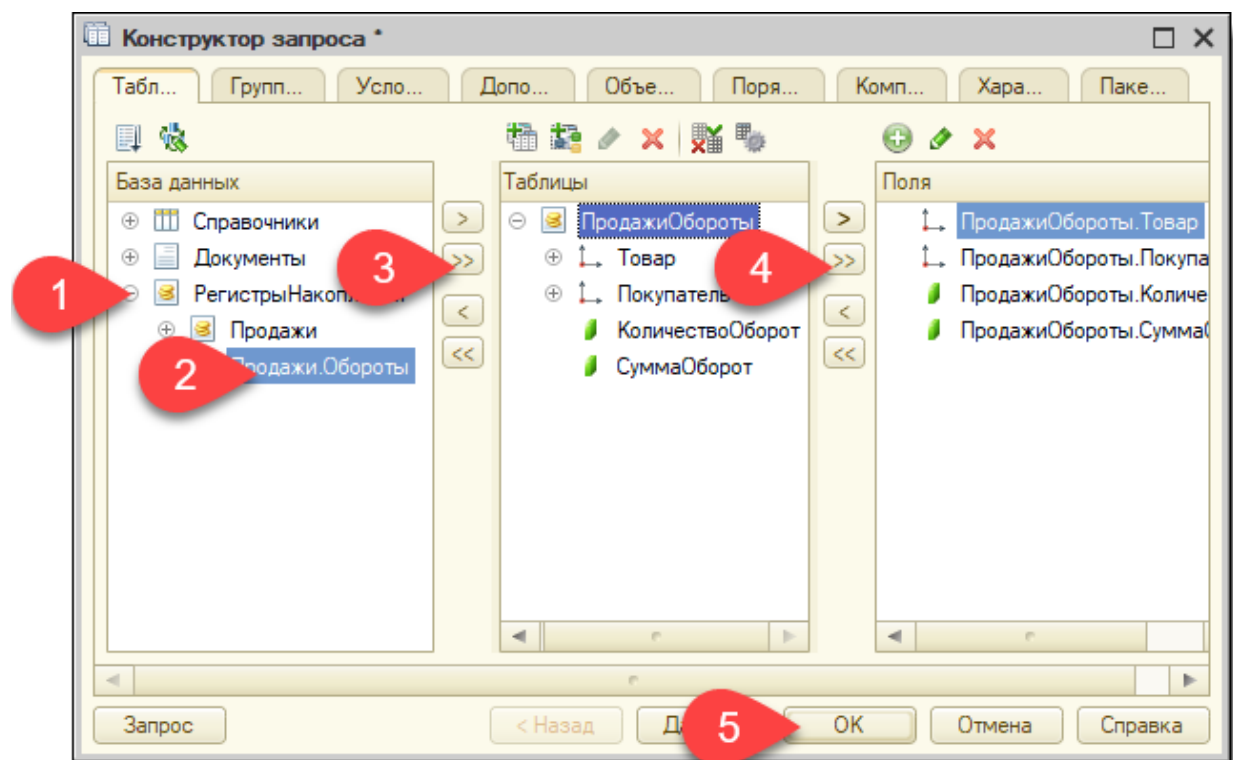
Кнопкой "Открыть схему компоновки данных" создадим новый макет схемы компоновки и откроем его.

В макете мы должны указать какие данные отображать и как их нужно отображать.

Добавим набор данных с типом "Запрос".

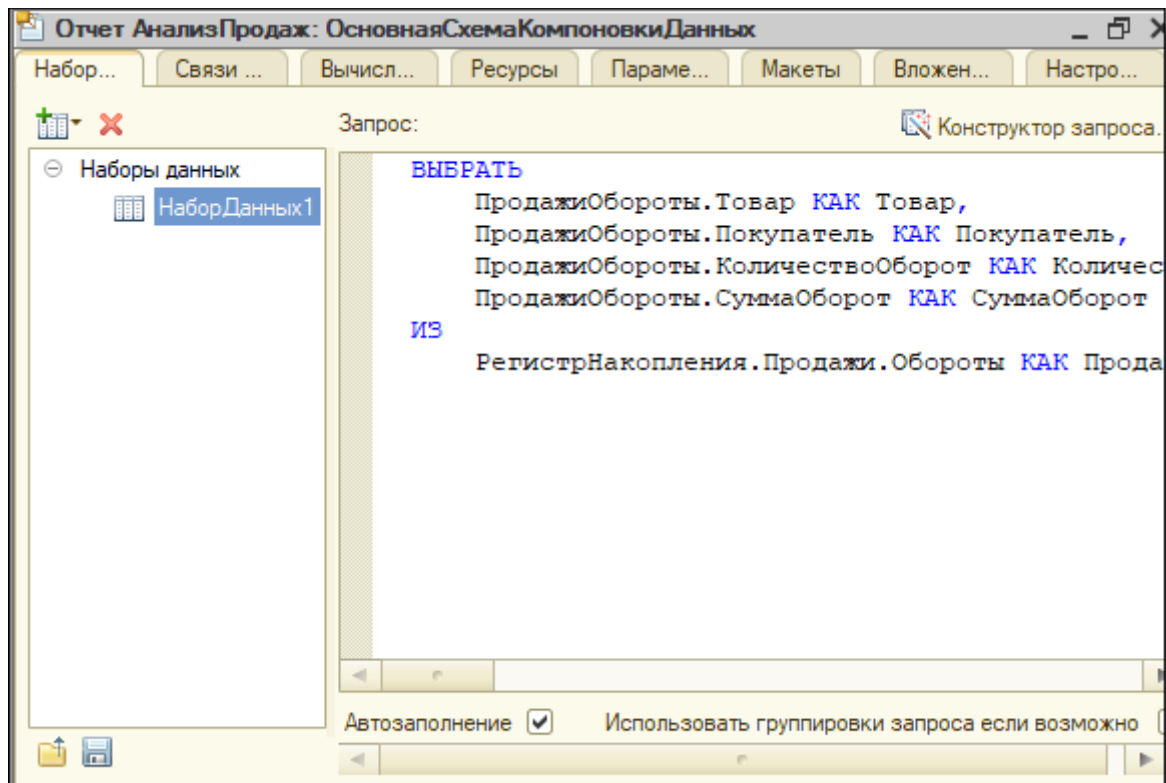


Сам запрос построим конструктором.



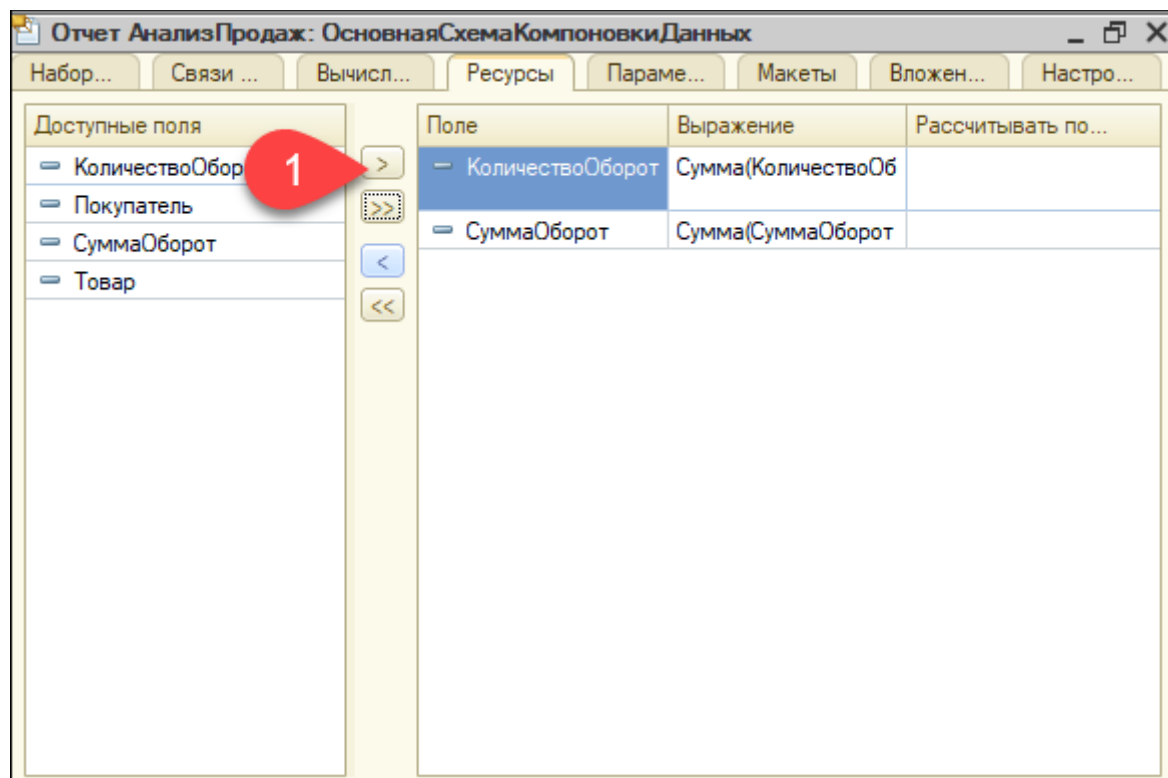
Выберем таблицу Продажи.Обороты, и перенесем все ее поля в выбранные.

Система сформирует текст запроса, при необходимости его можно отредактировать вручную.



Как видите текст запроса очень похож на язык Transact-SQL. Мы только что описали источник данных. Теперь нужно определить, как эти данные показать пользователю.

Опишем ресурсы. Собственно ресурсы – это и есть то, что мы хотим показать пользователю, то есть числа по продажам.

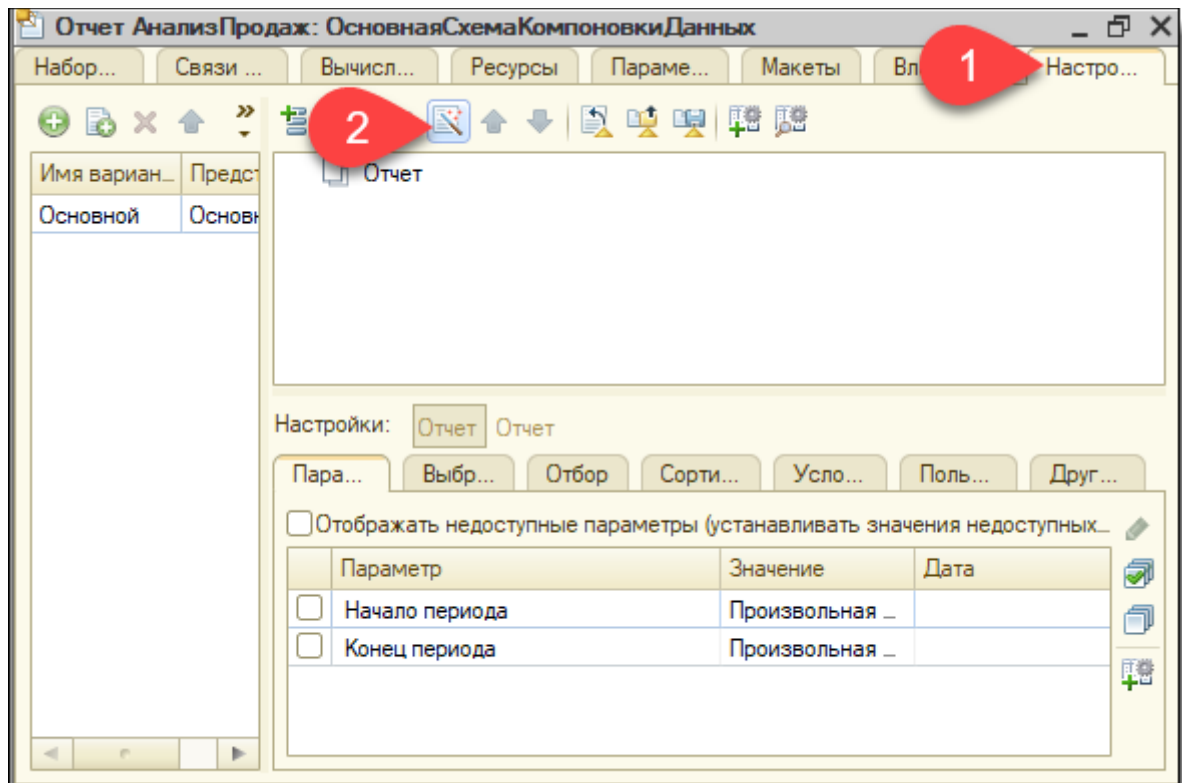


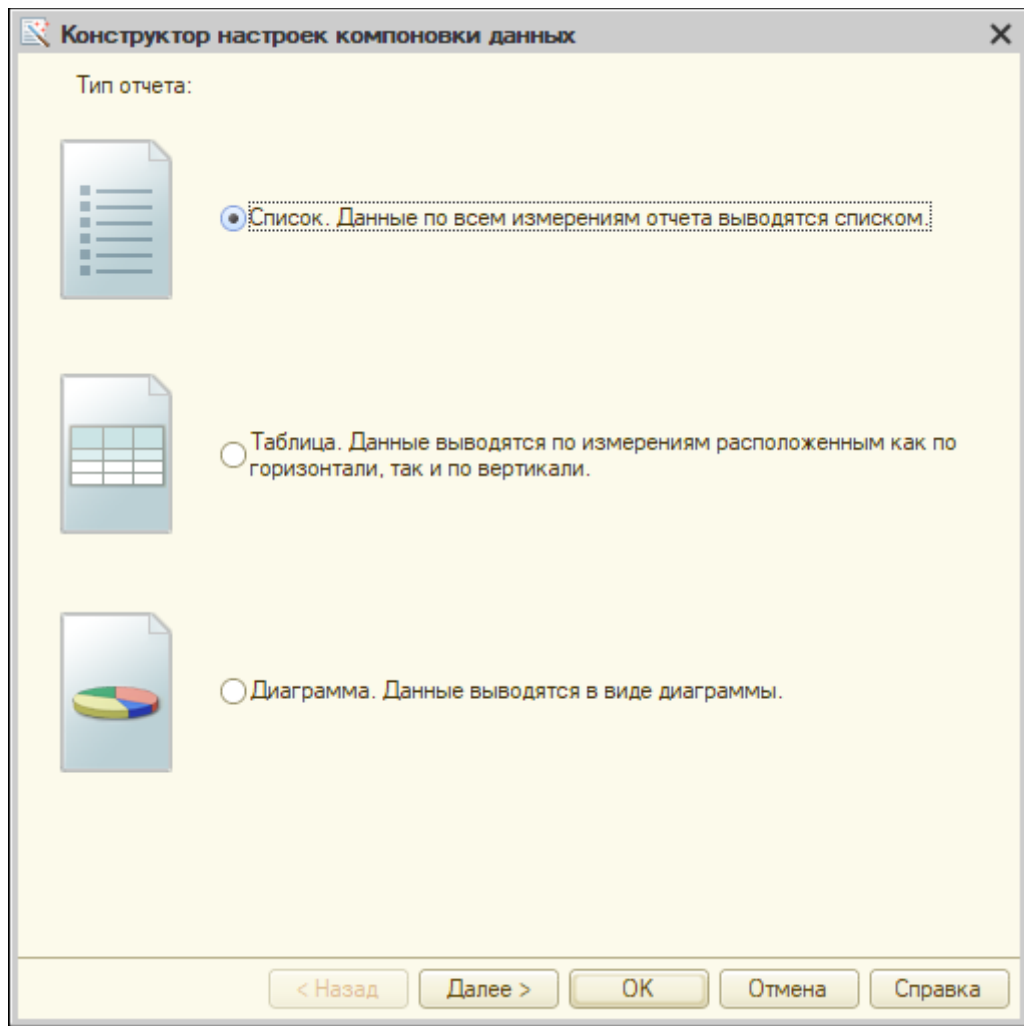
Кнопкой с двойной стрелкой перенесем все числовые поля в выбранные.

В колонке "Выражение" мы видим функцию "Сумма" (ее можно поменять), функция определяет, что делать с ресурсами, как считать итоги.

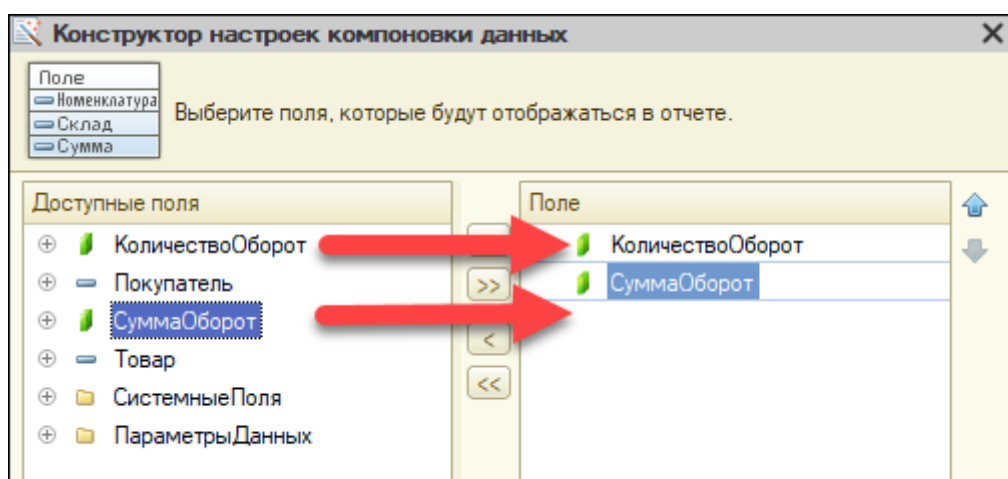
Теперь настроим отображение данных так, чтобы в итоге получилась простая таблица.

Перейдем на вкладку "Настройки" и запустим конструктор настроек.

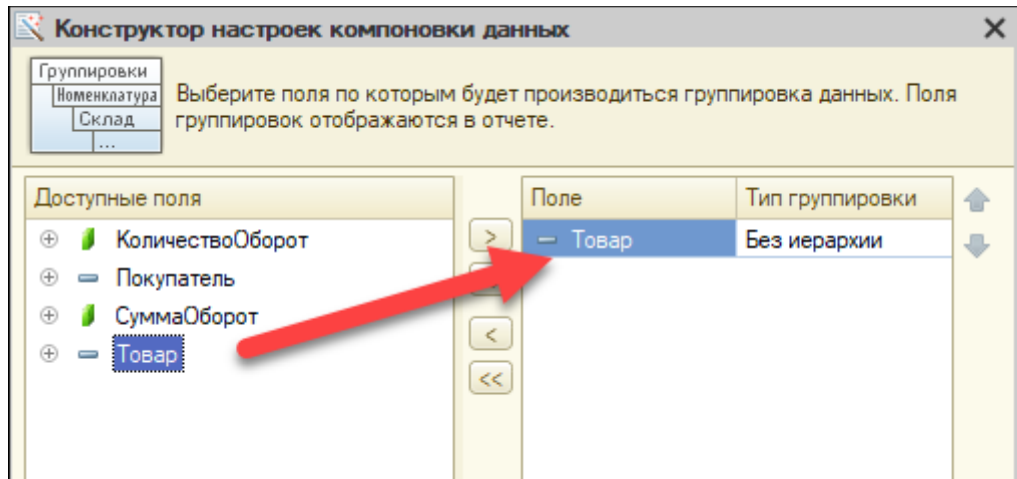




Тип отчета менять пока не будем.

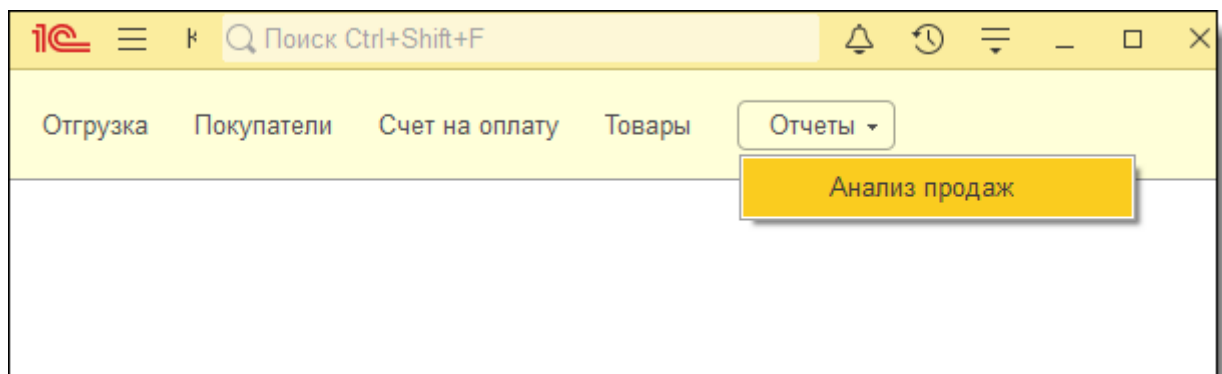


КоличествоОборот и СуммаОборот выберем как поля для отображения в отчете.

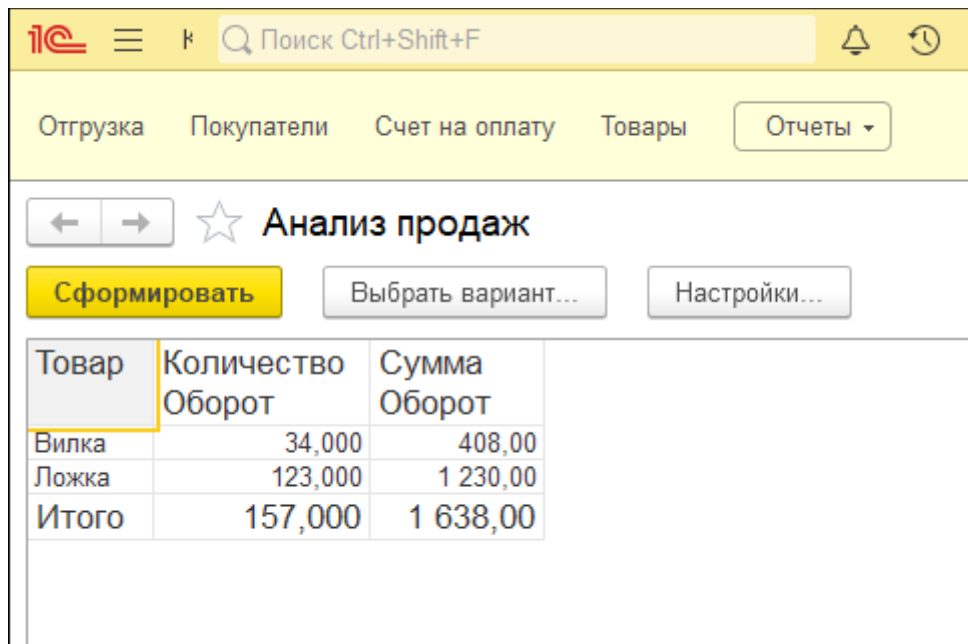


На следующем шаге выберем поле "Товар", именно в разрезе него и будет строиться наш отчет.

Отчет готов. Запустим систему и проверим его в действии.



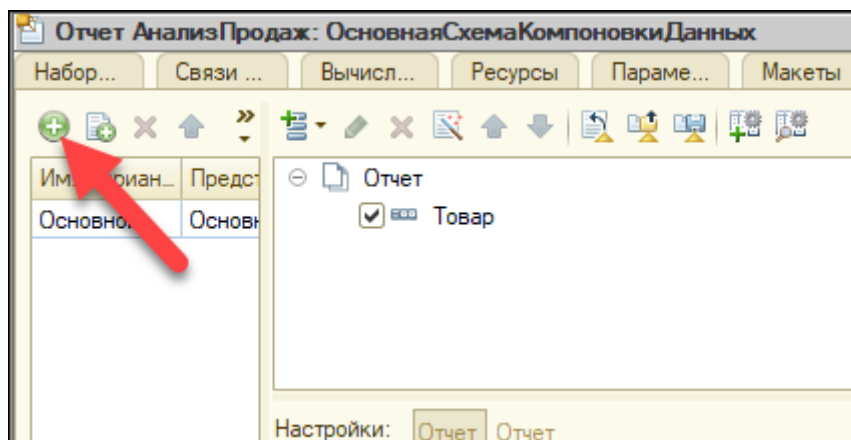
Сформируем отчет.



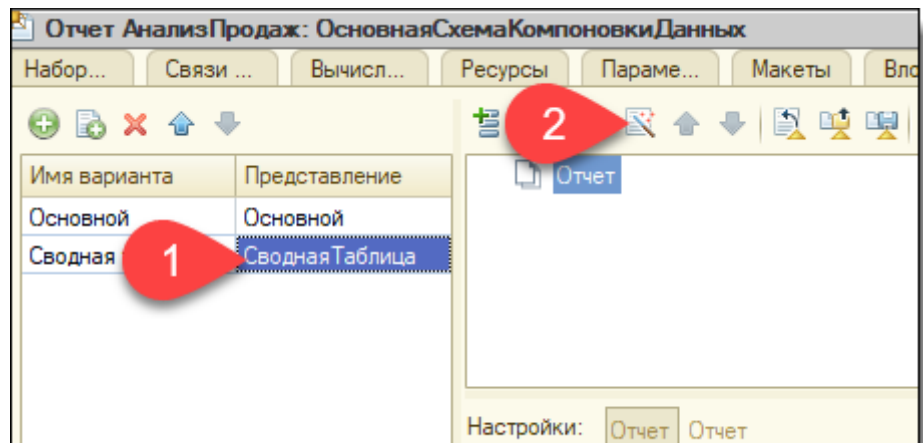
Товар	Количество	Сумма	Оборот
Вилка	34,000	408,00	
Ложка	123,000	1 230,00	
Итого	157,000	1 638,00	

Сделаем более сложный отчет, а точнее вариант этого. Вернемся в Конфигуратор.

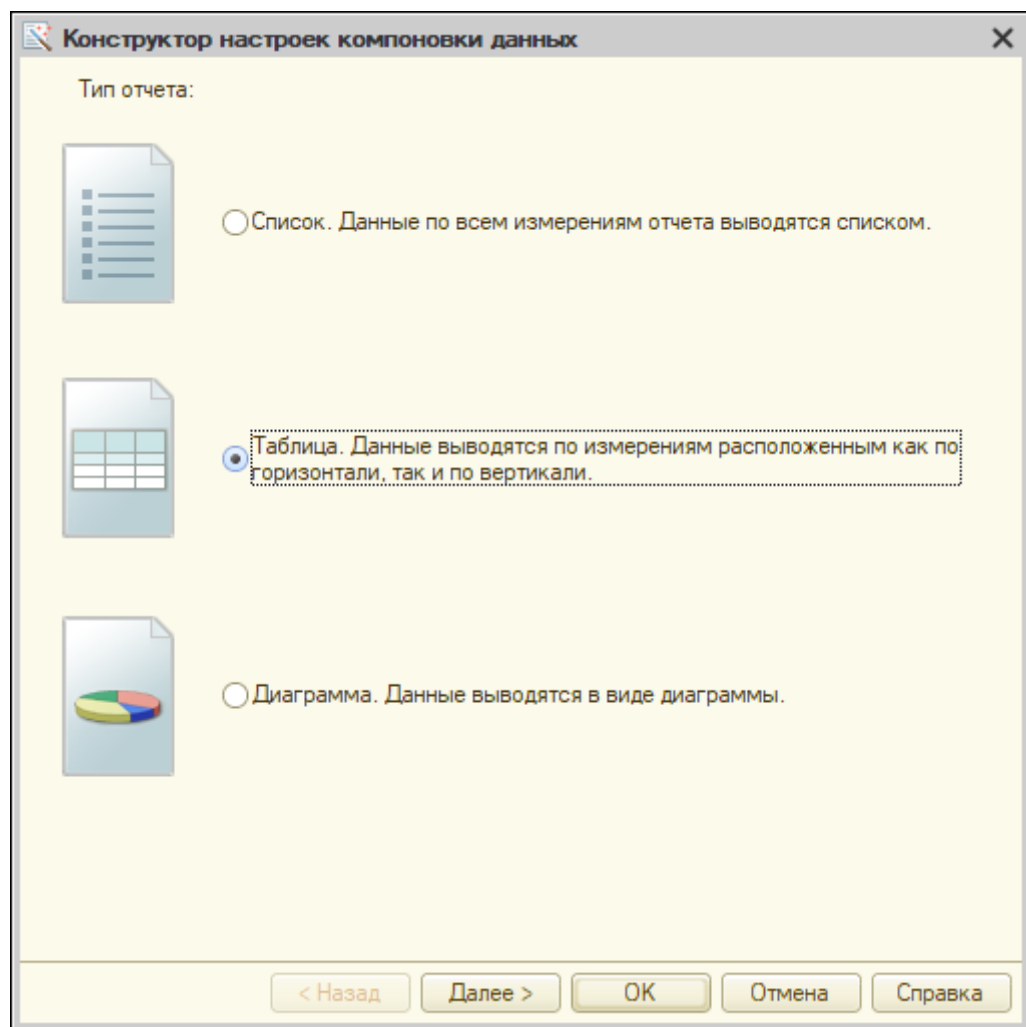
Добавим новый вариант отчета.



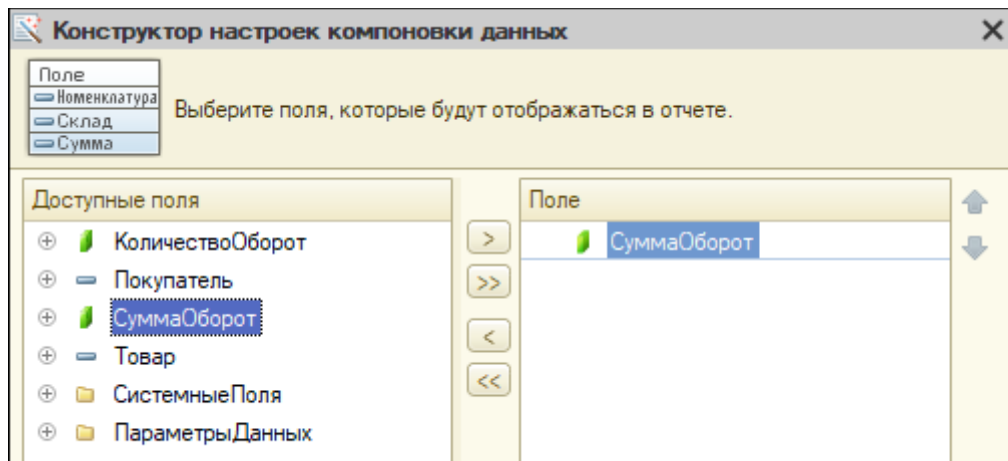
Назовем вариант отчета "Сводная таблица" и вновь запустим конструктор настроек.



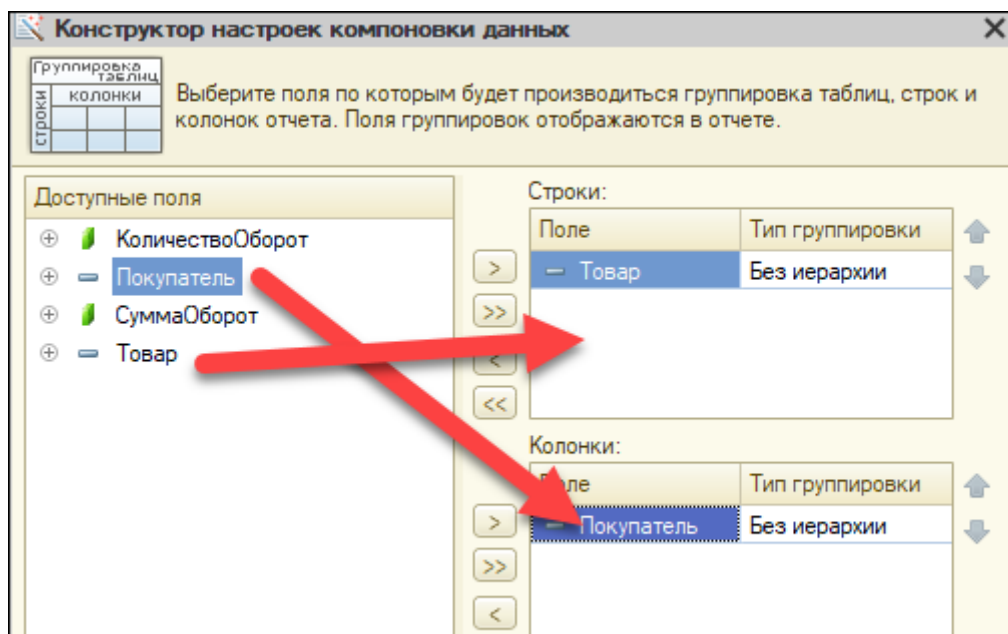
Теперь выберем "Таблица".



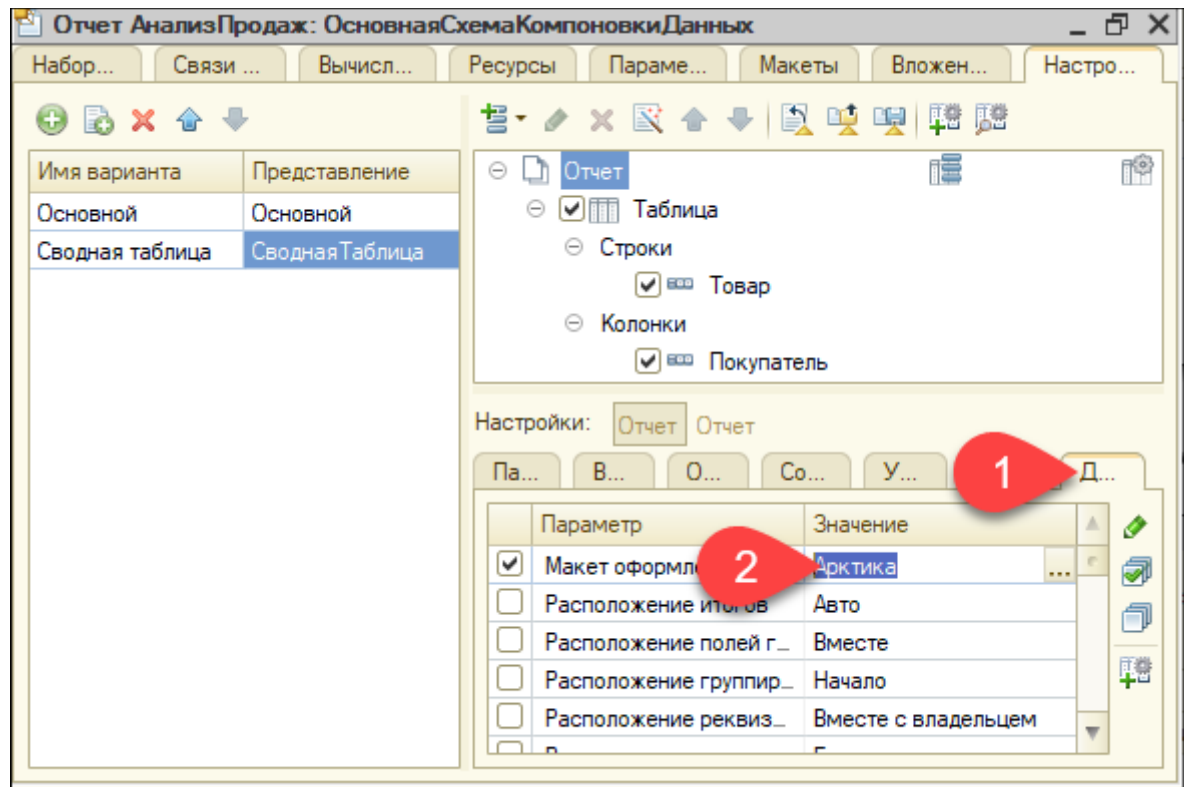
Выбирать теперь будем только СуммаОборот.



На следующей вкладке укажем что выводить в строках, а что в колонках.

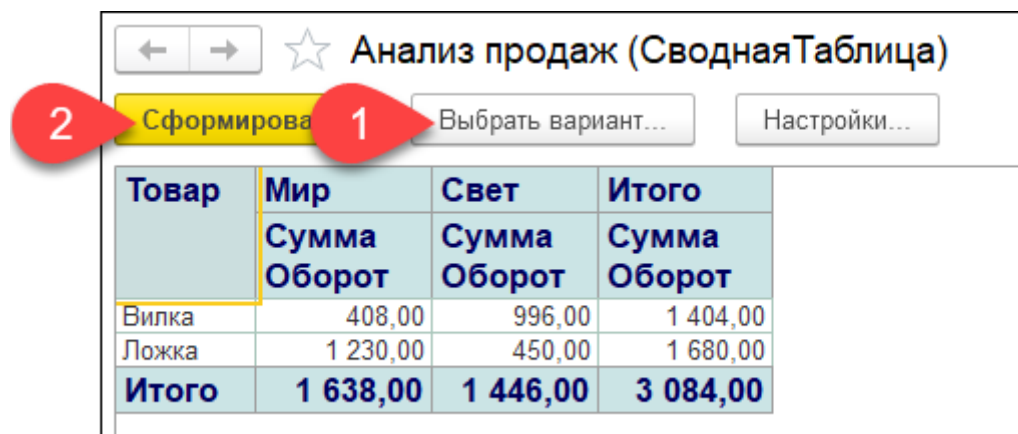


Конструктор можно закрыть кнопкой "Ок". Немного украсим отчет.



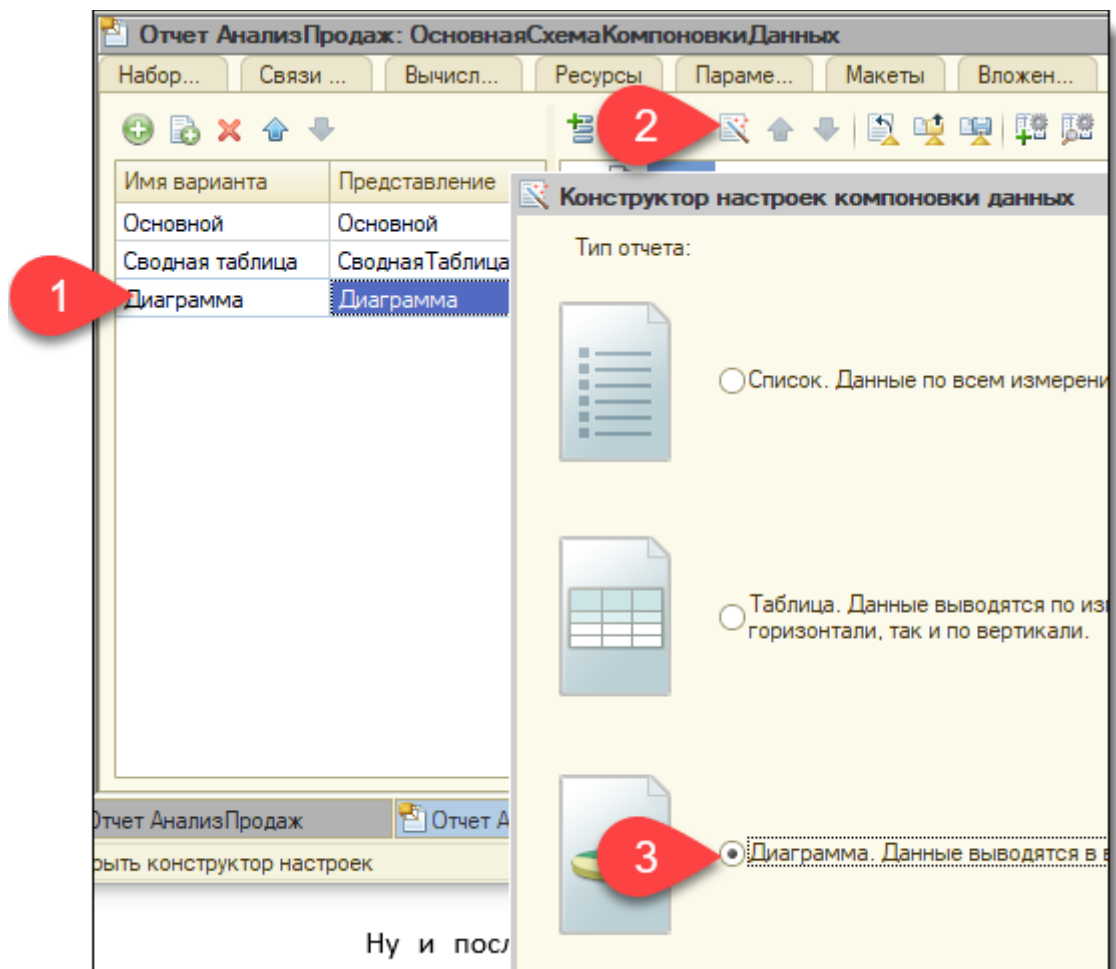
Найдем вкладку "Дополнительно" и выберите макет оформления.

Запускаем систему и строим отчет!

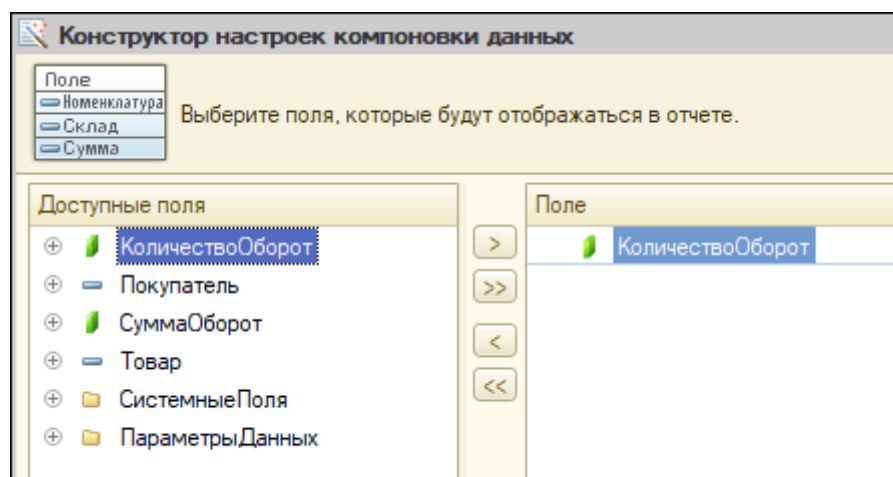


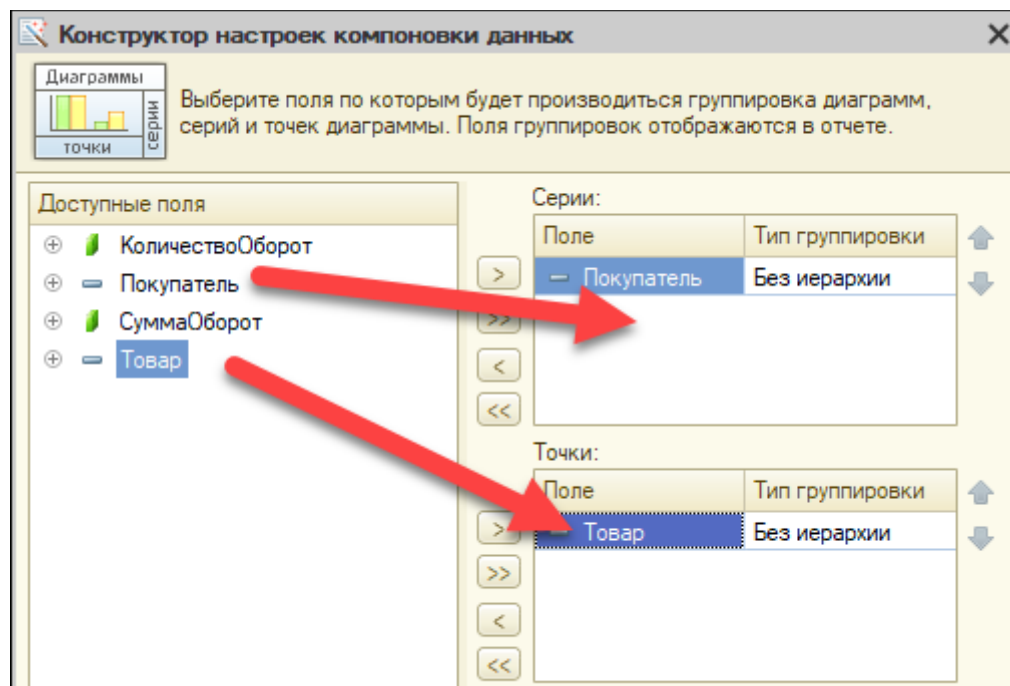
Не забудьте выбрать нужный вариант.

Ну и последний вариант, сделаем диаграмму. Добавим еще один вариант и запустим конструктор настроек.



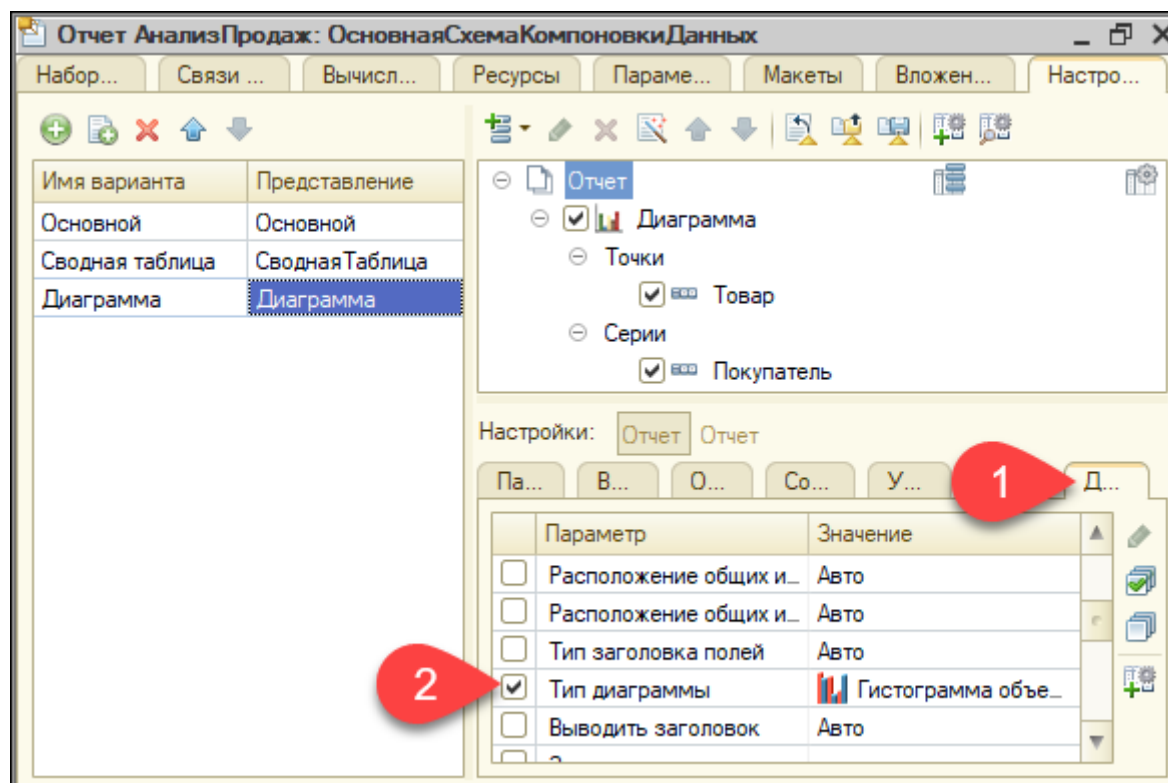
Теперь в выбранные поля добавим только КоличествоОборот.



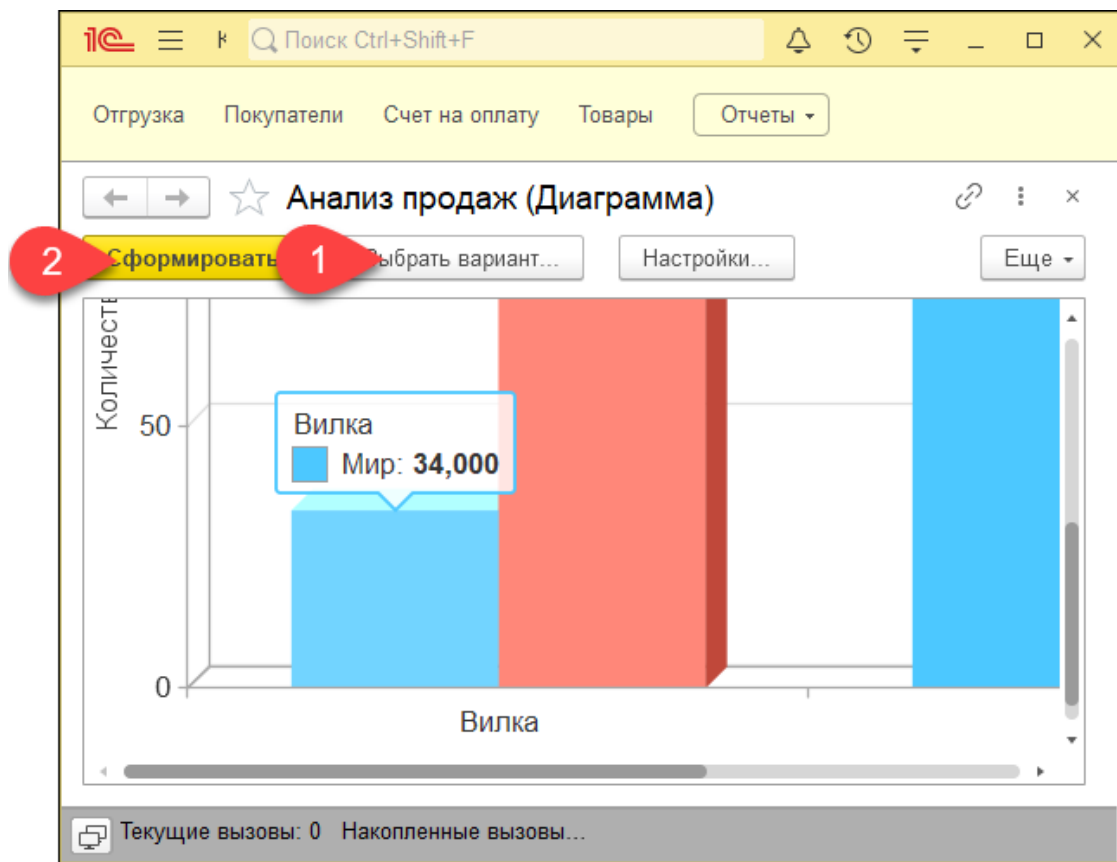


Выберем что будет представлять серии, а что точки в диаграмме.

Диаграмма на этом готова. Вид диаграммы можно выбрать на вкладке "Дополнительно".

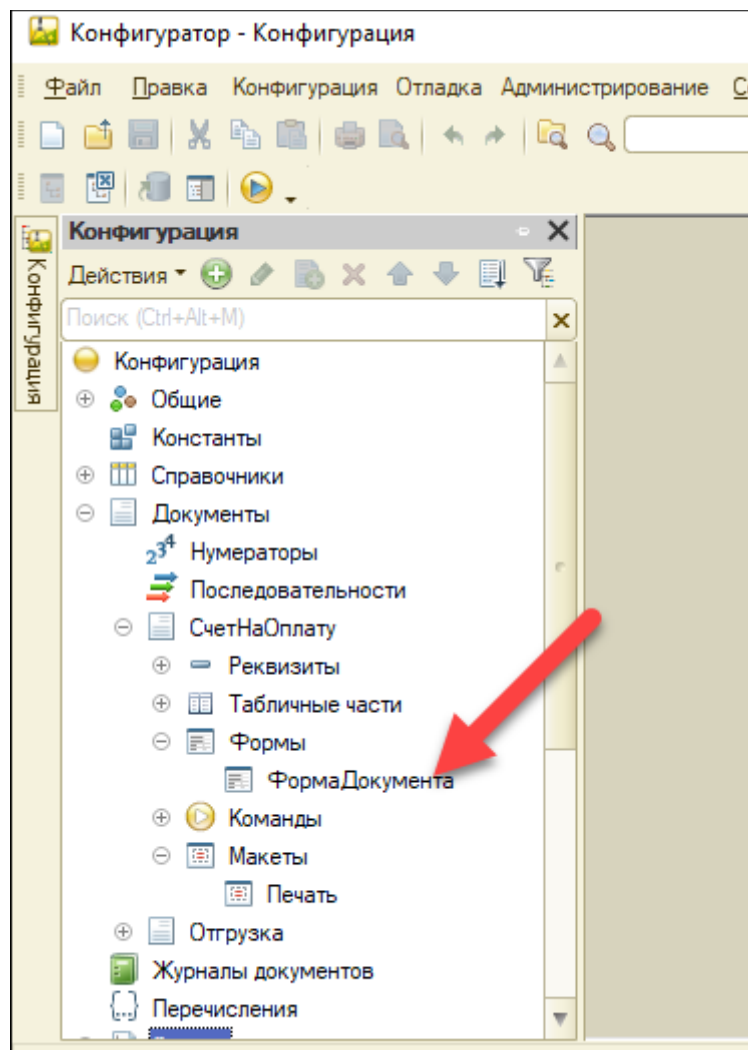


Готово!



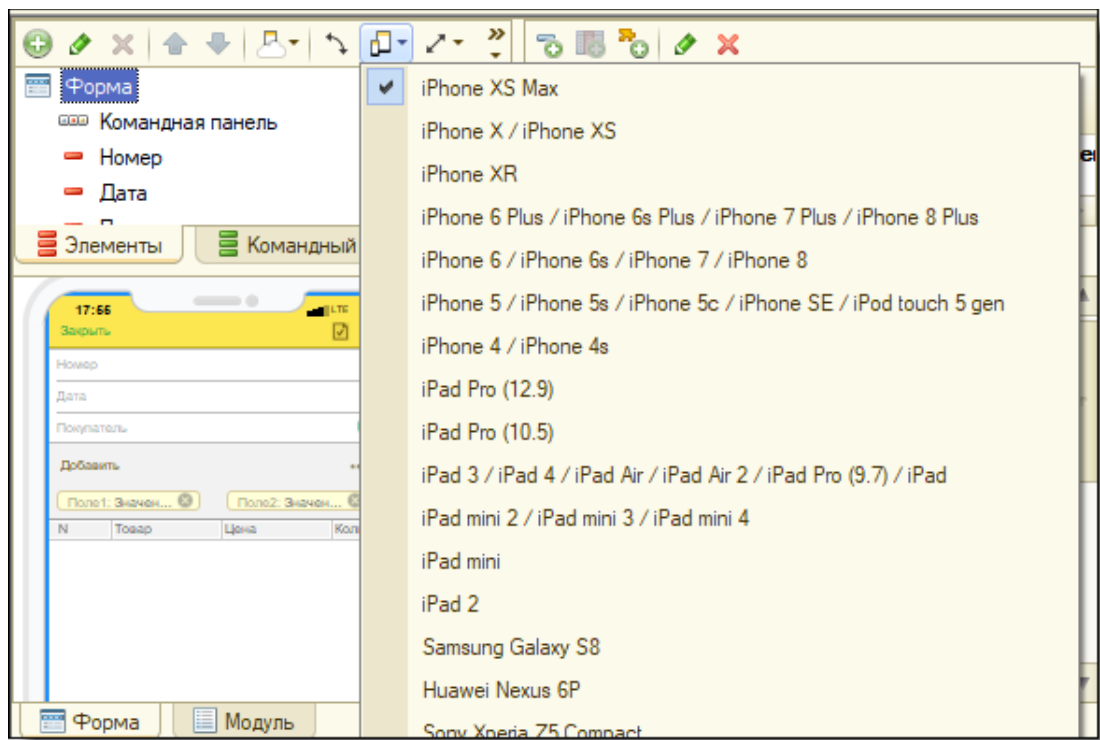
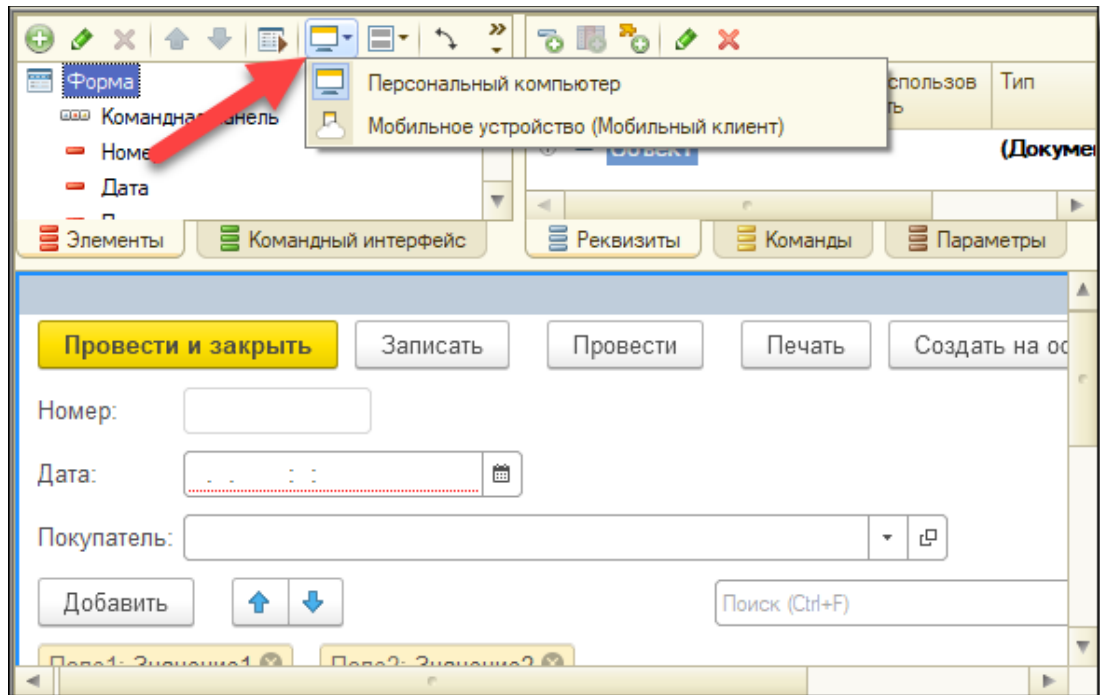
Заключение

Стоит отметить, что только что созданное приложение УЖЕ может функционировать на мобильных устройствах. Как пример можно продемонстрировать как будет выглядеть форма документа "Счет" на мобильном устройстве.



Откройте форму документа "Счет на оплату".

И поменяйте вид отображения формы.



Форма поменялась на мобильный телефон, при желании можно выбрать конкретную модель телефона или планшета.

Вот такой не большой мастер-класс, как заинтересовать слушателей системой 1С:Предприятие.