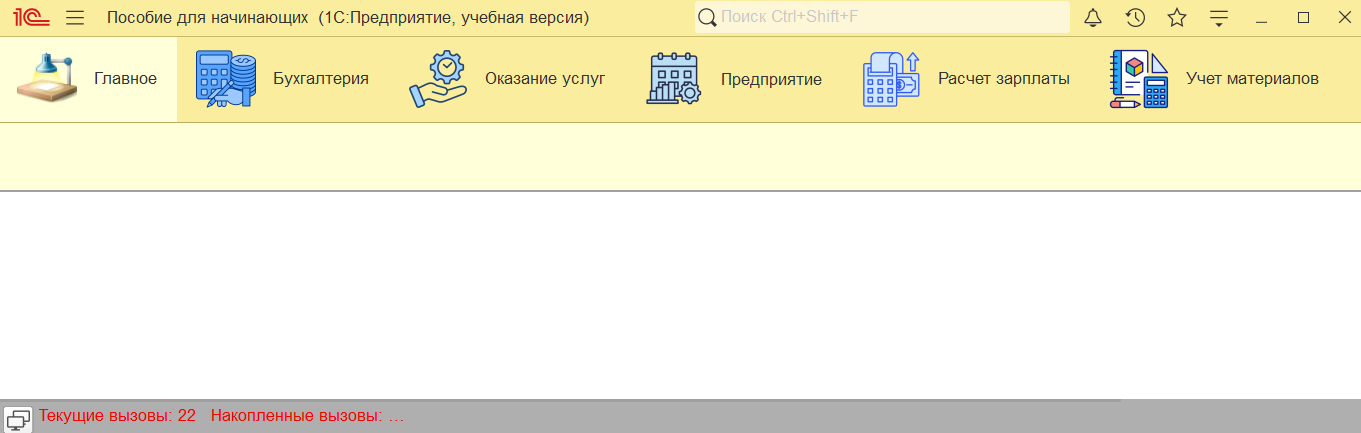
**ПРАКТИЧЕСКОЕ ЗАДАНИЕ**

**ЦЕЛЬ: Разработка прикладного решения для компании ООО «На все руки мастер»**

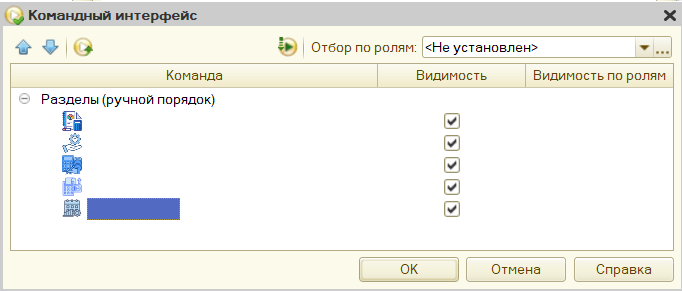
1. Создайте пустую информационную базу, в палитре свойств задайте имя конфигурации – *ПособиеДляНачинающих*, назначение использования – *Приложение для платформы*. Проверьте внешний вид интерфейса прикладного решения.

**ЧАСТЬ 1. ПОДСИСТЕМЫ**

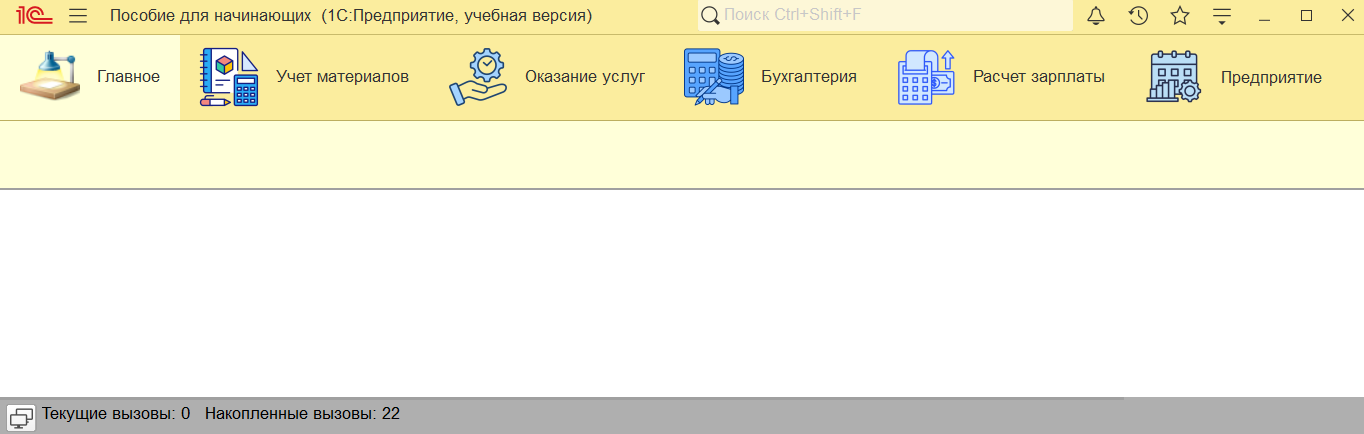
1. Создайте подсистемы *Бухгалтерия*, *УчетМатериалов*, *ОказаниеУслуг*, *РасчетЗарплаты*, *Предприятие*. Расположение: Общие –> Подсистемы. Установите в качестве картинки общую картинку, добавив ее из файла, для каждой подсистемы.



1. В режиме «Конфигуратор» откройте командный интерфейс конфигурации. Расположим сначала подсистемы, отражающие производственную деятельность нашей фирмы: *Учет материалов* и *Оказание услуг*, затем бухгалтерскую деятельность и расчет зарплаты сотрудников: *Бухгалтерия* и *Расчет зарплаты*, а затем подсистему *Предприяти*е.



Проверьте отображение в режиме «1С:Предприятие».

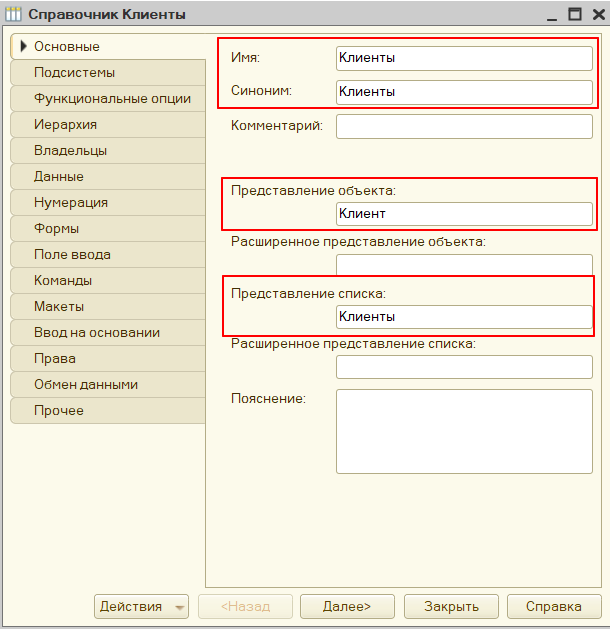
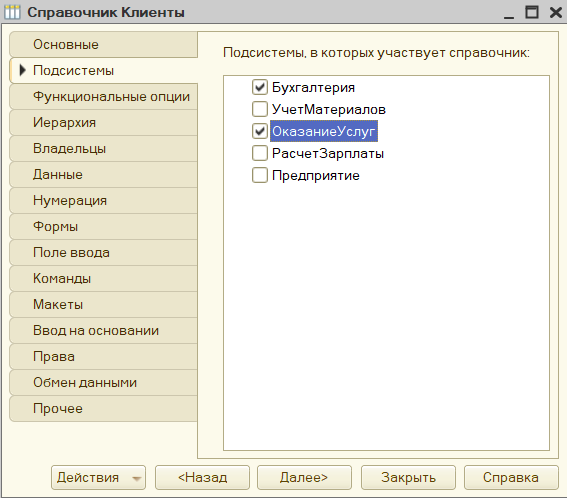


**КОНТРОЛЬНЫЕ ВОПРОСЫ**

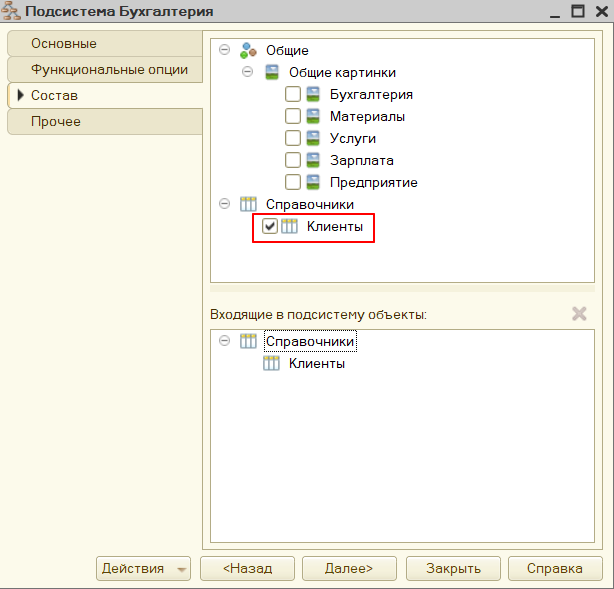
* Для чего используется объект конфигурации «Подсистема»?
* Как описать логическую структуру конфигурации при помощи объектов «Подсистема»?
* Как управлять порядком вывода и отображением подсистем в конфигурации?
* Что такое окно редактирования объекта конфигурации, и в чем его отличие от палитры свойств?

**ЧАСТЬ 2. СПРАВОЧНИКИ**

1. Создайте справочник *Клиенты*. Установите основные свойства справочника. Определите список подсистем, в которых отображается справочник.

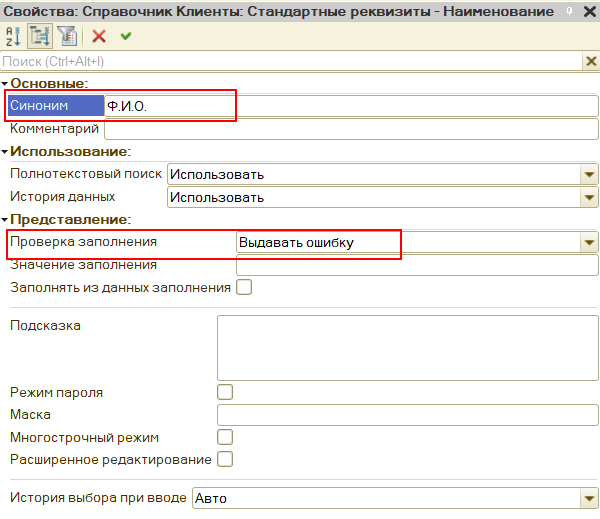
 

Перейдите в подсистему *Бухгалтерия*, на закладку *Состав*, здесь появился новый объект конфигурации. На закладке *Состав* также можно изменять список объектов, входящих в подсистему.

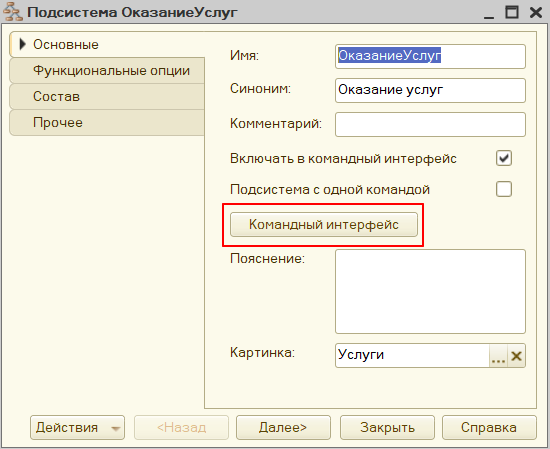
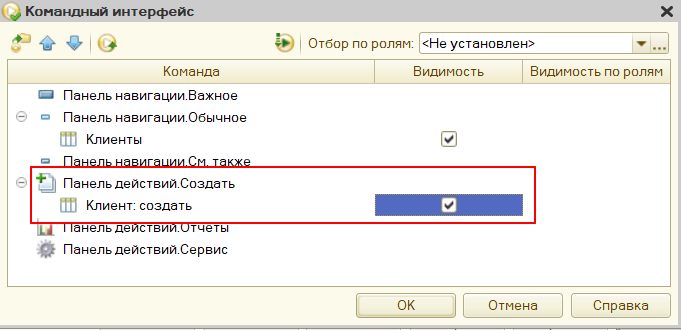


1. Вернитесь к справочнику *Клиенты*, перейдите на вкладку *Данные*, установите длину кода 9 и наименование справочника 50.

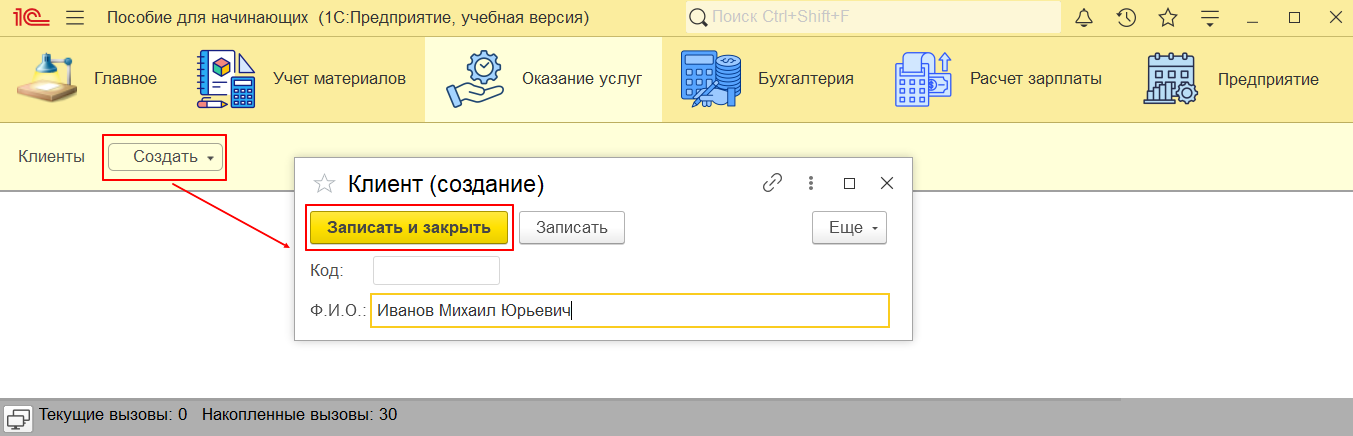
В палитре свойств стандартного реквизита *Наименование* установим свойство *Синоним* как *Ф.И.О.*



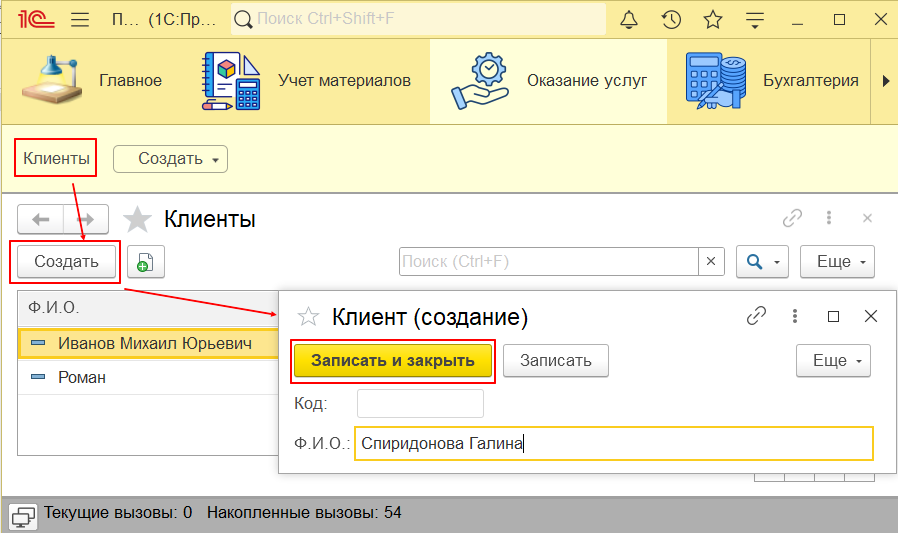
1. Сделайте доступной в панели команд раздела *ОказаниеУслуг* стандартную команду для создания новых клиентов. Для этого вызовите настройки командного интерфейса подсистемы *ОказаниеУслуг*.

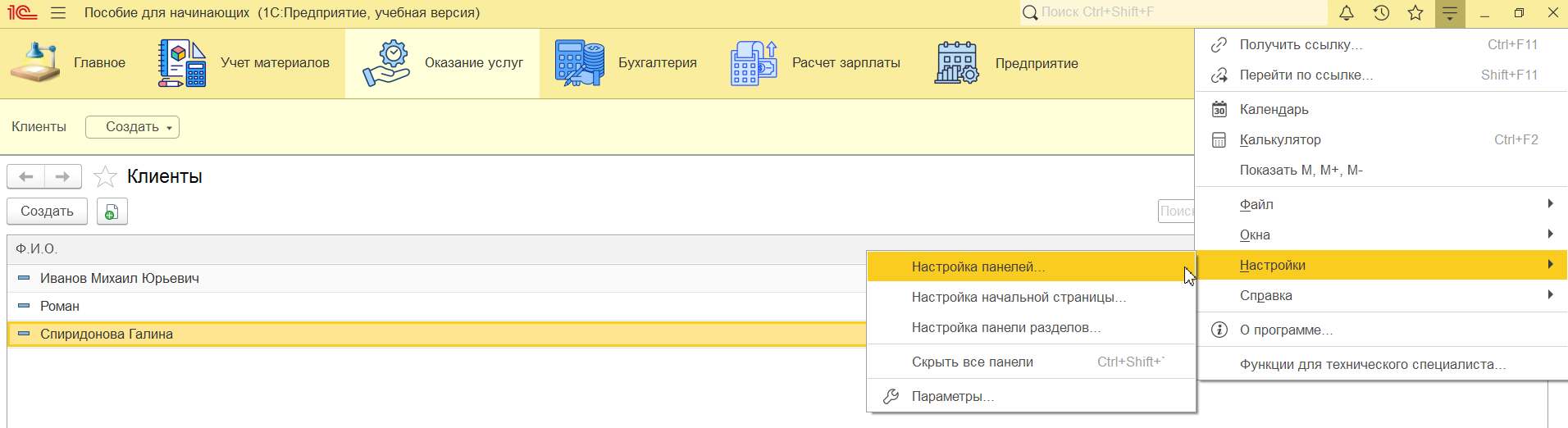
1. Сохраните список изменений в структуре конфигурации.
2. Проверьте внешний интерфейс прикладного решения. В разделах *Оказание услуг* и *Бухгалтерия* появился справочник *Клиенты*.
3. В панели команд раздела *Оказание услуг* раскройте подменю *Создать* и выполните команду *Клиент*. Внесите Ф.И.О. нового клиента *Иванов Михаил Юрьевич*.

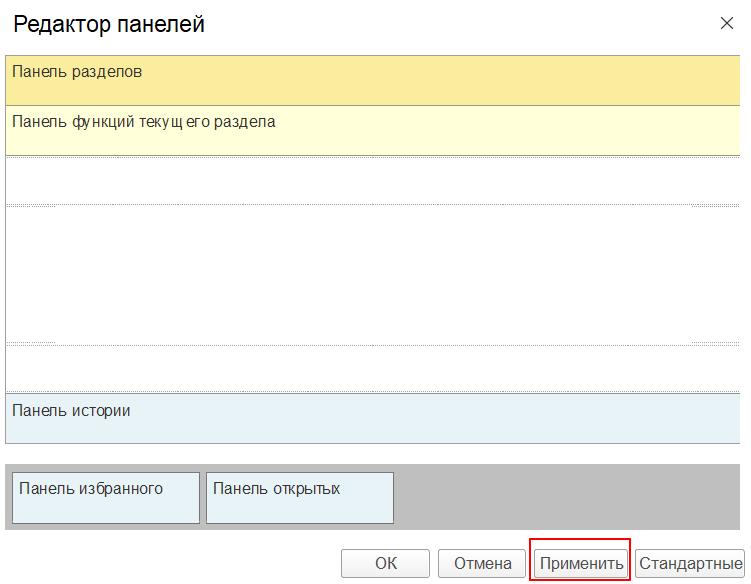


Аналогично добавьте клиента *Роман*. Последнего клиента с Ф.И.О. *Спиридонова Галина* добавьте, пользуясь формой списка клиентов.

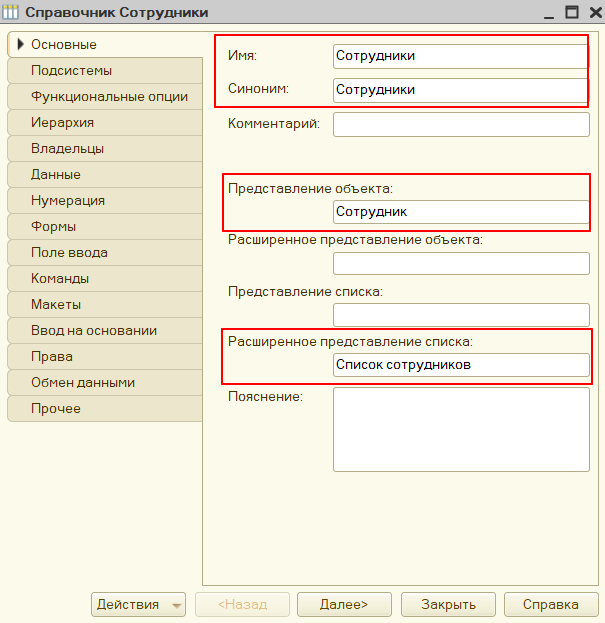
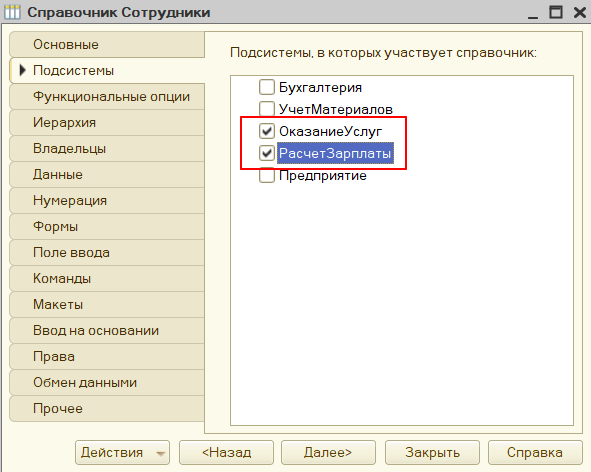


1. Настройте расположение панелей прикладного решения в режиме «*Предприятие*».

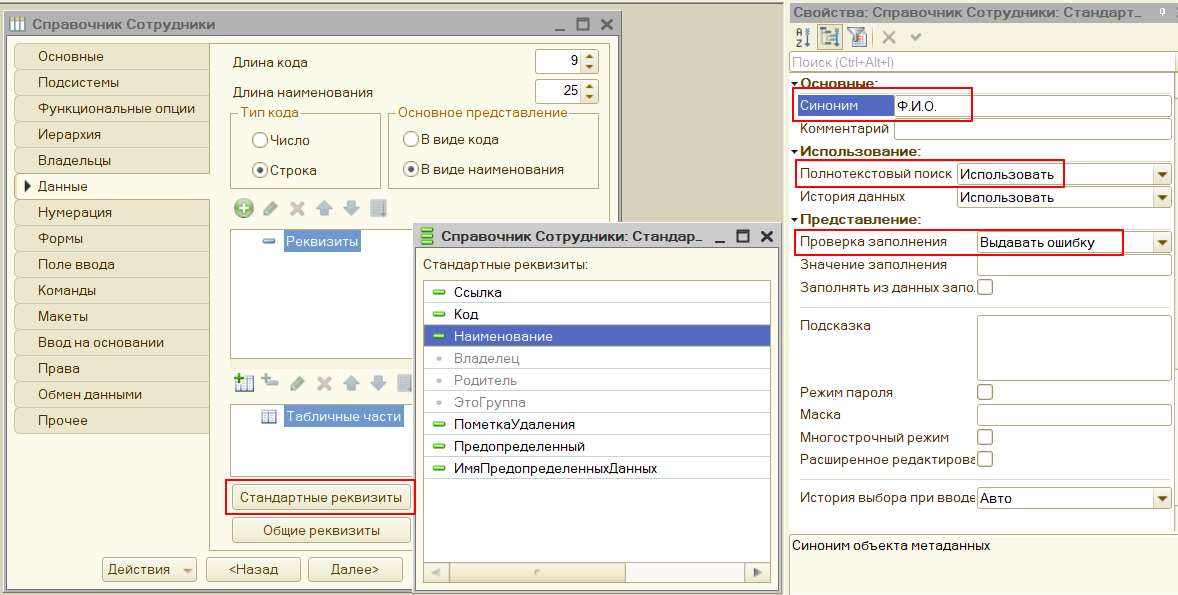




1. Создайте справочник *Сотрудники* с табличной частью. Определите список подсистем, в которых отображается справочник.

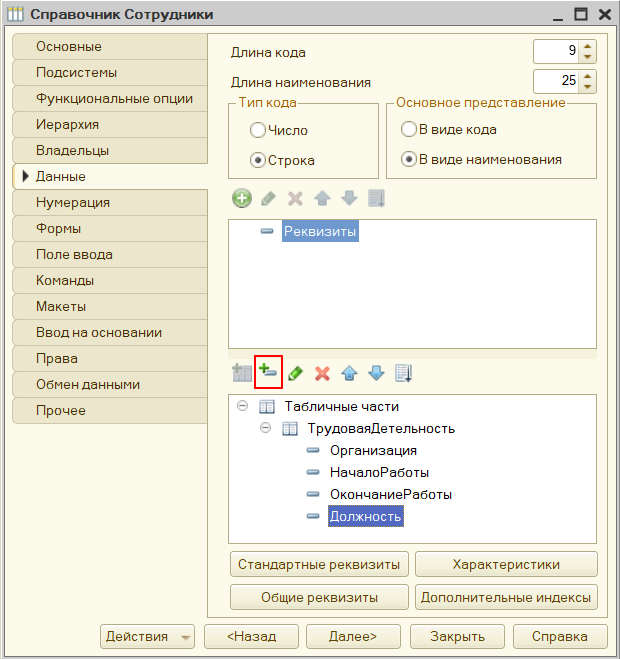
 

Установите синоним стандартного реквизита *Наименование*.



Добавьте в справочник табличную часть с именем *ТрудоваяДеятельность*. Добавьте следующие реквизиты:

* *Организация* – тип Строка, длина 100;
* *НачалоРаботы* – тип Дата, состав даты – Дата;
* *ОкончаниеРаботы* – тип Дата, состав даты – Дата;
* *Должность* – тип Строка, длина 100.



1. Сделайте видимой стандартную команду для создания новых сотрудников. Откройте окно редактирования объекта конфигурации *Подсистема РасчетЗарплаты* и нажмем кнопку *Командный интерфейс*. В открывшемся окне Командный интерфейс отразятся все команды этой подсистемы. В группе *Панель действий.Создать* включите видимость у команды *Сотрудник: создать*.
2. Обновите конфигурацию. В режиме «1С:Предприятие» проверьте наличие справочника *Сотрудники* в разделе *Расчет зарплаты*. Создайте следуюих сотрудников:

* *Гусаков Николай Дмитриевич*

Трудовая деятельность:

* Организация – ЗАО «НТЦ»,
* Начало работы – 01.02.2009,
* Окончание работы – 16.04.2025,
* Должность – Ведущий специалист.
* *Деловой Иван Сергеевич*

Трудовая деятельность:

1 :

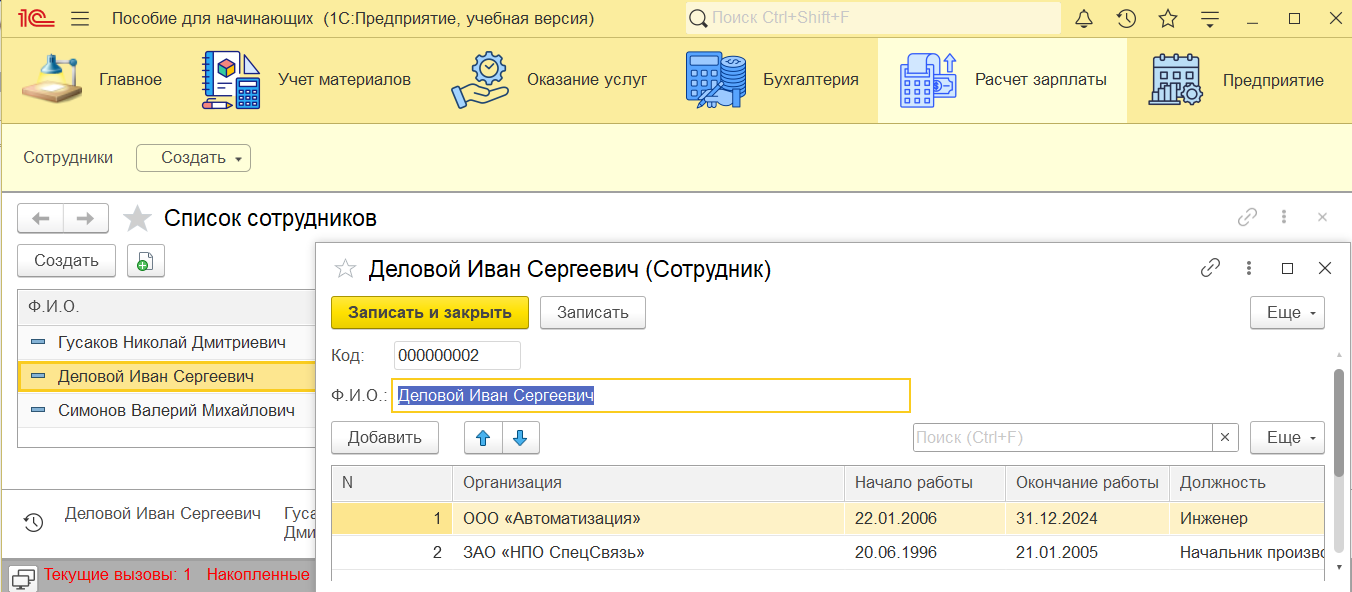
* Организация – ООО «Автоматизация»,
* Начало работы – 22.01.2006,
* Окончание работы – 31.12.2024,
* Должность – Инженер.

2 :

* Организация – ЗАО «НПО СпецСвязь»,
* Начало работы – 20.06.1996,
* Окончание работы – 21.01.2005,
* Должность – Начальник производства.
* *Симонов Валерий Михайлович*

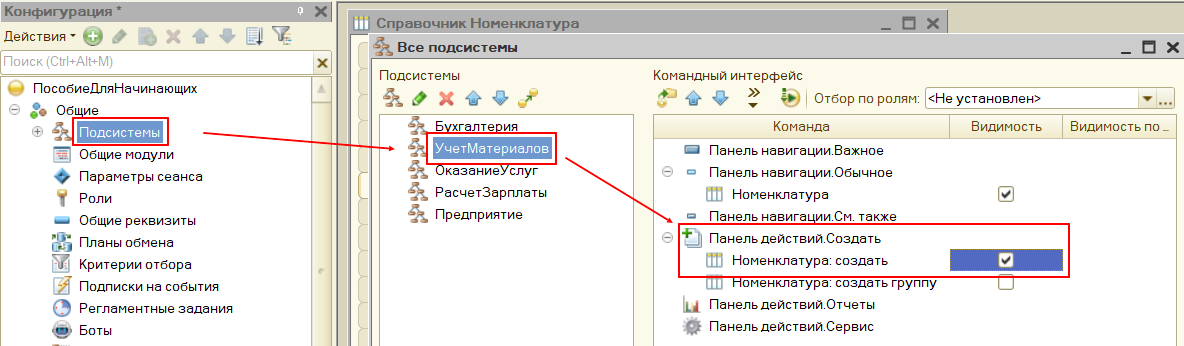
Трудовая деятельность:

* Организация – ООО «СтройМастер»,
* Начало работы – 06.02.2009,
* Окончание работы – 03.04.2025,
* Должность – Прораб.

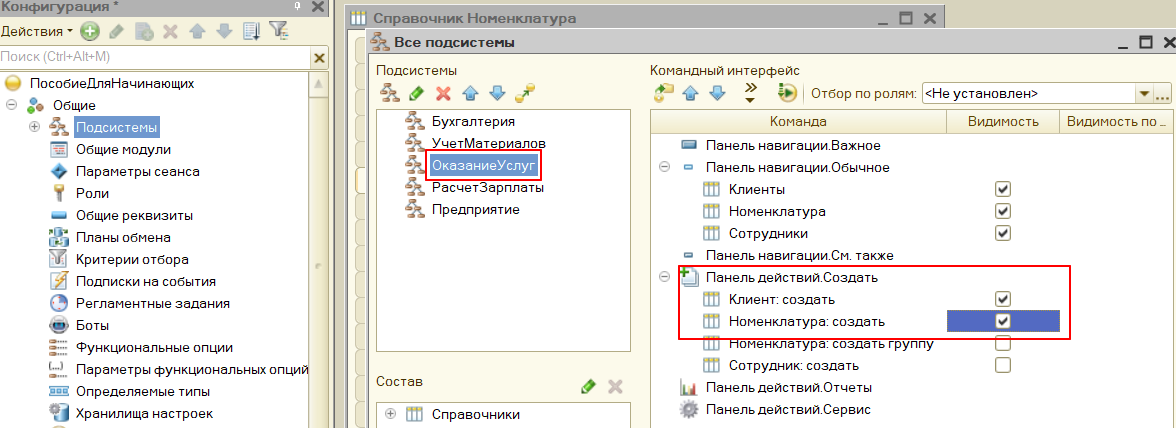


1. Создайте новый объект конфигурации *Справочник* и назовите его *Номенклатура*. Справочник Номенклатура будет содержать информацию об услугах, которые оказывает ООО «На все руки мастер», и о тех материалах, которые при этом могут быть использованы. По логике конфигурации список номенклатуры должен быть доступен в разделах (подсистемах) *Учет материалов*, *Оказание услуг* и *Бухгалтерия*. Отметьте, что справочник иерархический, вид иерархии групп и элементов. На закладке *Данные* оставьте по умолчанию длину и тип кода, длину наименования справочника задайте равной 100 символам. Здесь же измените *Синоним* стандартного реквизита *Родитель* справочника на более понятное обозначение *Группа номенклатуры*.
2. Сделайте доступной в панели команд разделов *УчетМатериалов* и *ОказаниеУслуг* стандартную команду для создания новых элементов списка номенклатуры.

Для этого в дереве объектов конфигурации выделите ветвь *Подсистемы*, вызовите ее контекстное меню и выберите пункт *Все подсистемы*. В открывшемся окне слева в списке *Подсистемы* выделите подсистему *УчетМатериалов*. Справа в списке Командный интерфейс отразятся все команды выбранной подсистемы. В группе *Панель действий.Создать* включите видимость у команды *Номенклатура: создать*.

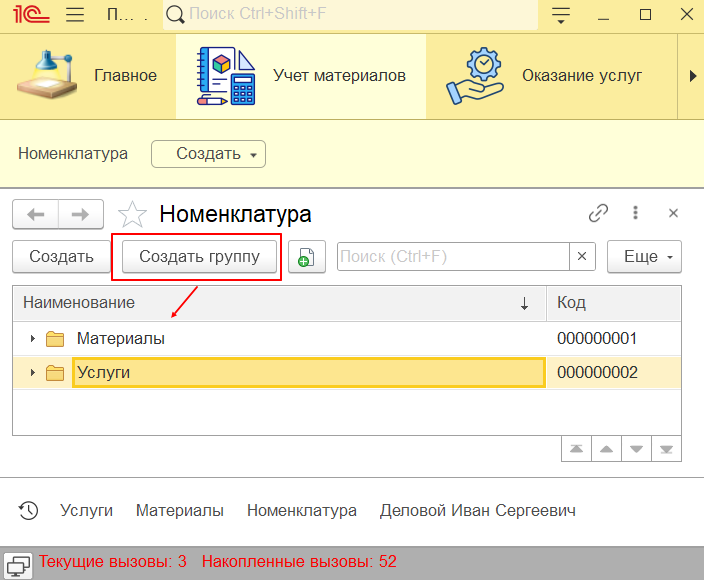


Выделив в списке подсистем *ОказаниеУслуг*, проделайте те же действия.



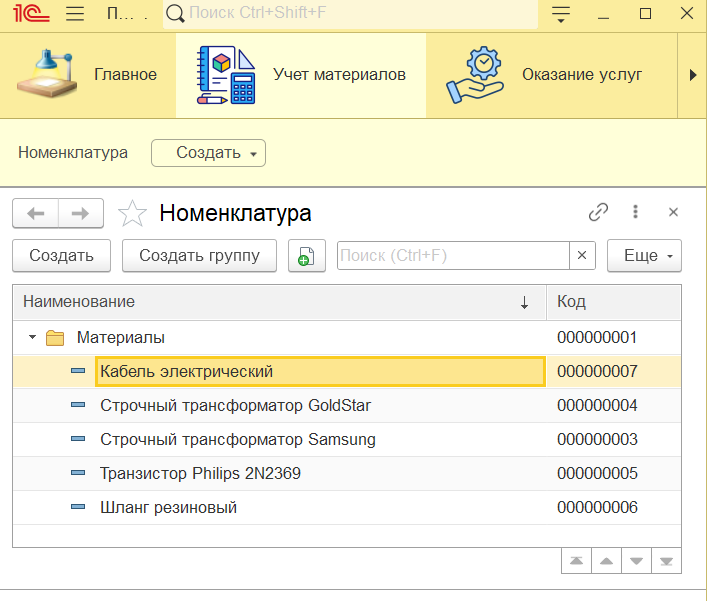
Обновите конфигурацию.

1. В открывшемся окне «1С:Предприятия» в панелях команд разделов *Учет материалов*, *Оказание услуг* и *Бухгалтерия* появилась команда *Номенклатура* для открытия списка номенклатуры. Выполните команду *Номенклатура* в разделе *Учет материалов*. Создайте две группы в корне справочника: *Материалы* и *Услуги*.



Затем двойным щелчком мыши раскройте группу *Материалы* и создайте в ней пять элементов:

* Строчный трансформатор Samsung,
* Строчный трансформатор GoldStar,
* Транзистор Philips 2N2369,
* Шланг резиновый,
* Кабель электрический.

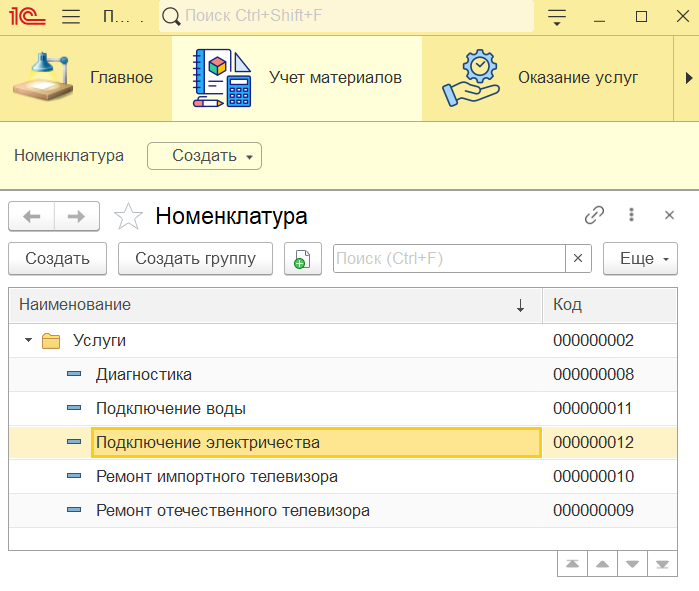


Далее раскройте группу *Услуги* и тоже создайте в ней несколько элементов – услуги по ремонту телевизоров:

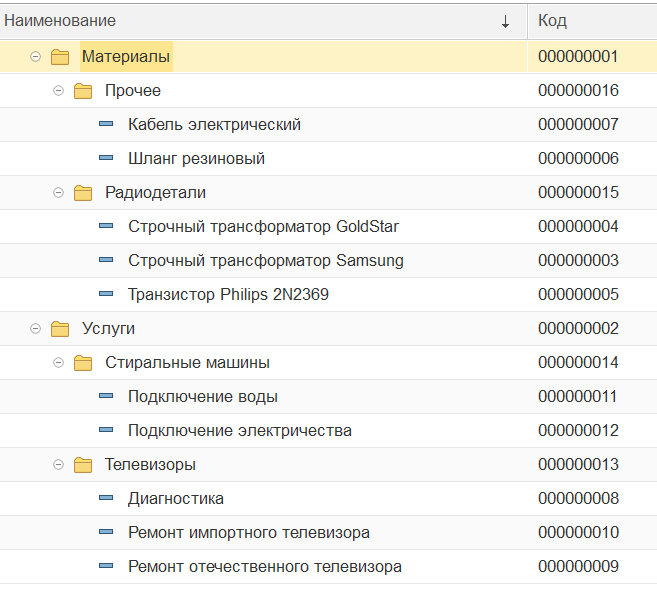
* Диагностика,
* Ремонт отечественного телевизора,
* Ремонт импортного телевизора.

И услуги по установке стиральных машин:

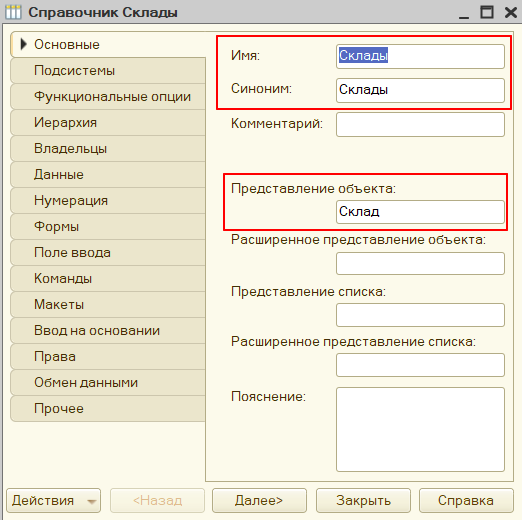
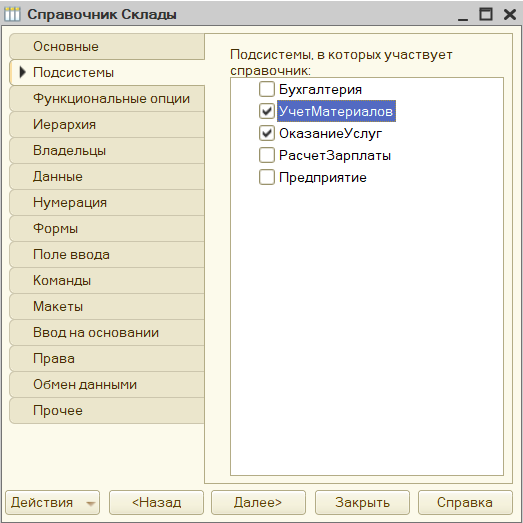
* Подключение воды,
* Подключение электричества.



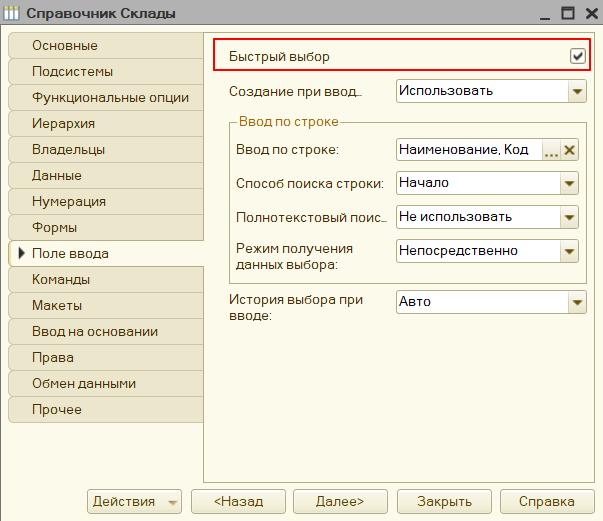
1. Разнесите услуги по двум смысловым группам: услуги по ремонту телевизоров и услуги по установке стиральных машин. Для этого в группе *Услуги* создайте еще две группы: *Телевизоры* и *Стиральные машины*, после переместите по смыслу услуги. Затем в группе *Материалы* создайте две группы: *Радиодетали* и *Прочее*. В группу *Прочее* поместим *Кабель электрический* и *Шланг резиновый*. Остальные материалы переместим в группу *Радиодетали*.



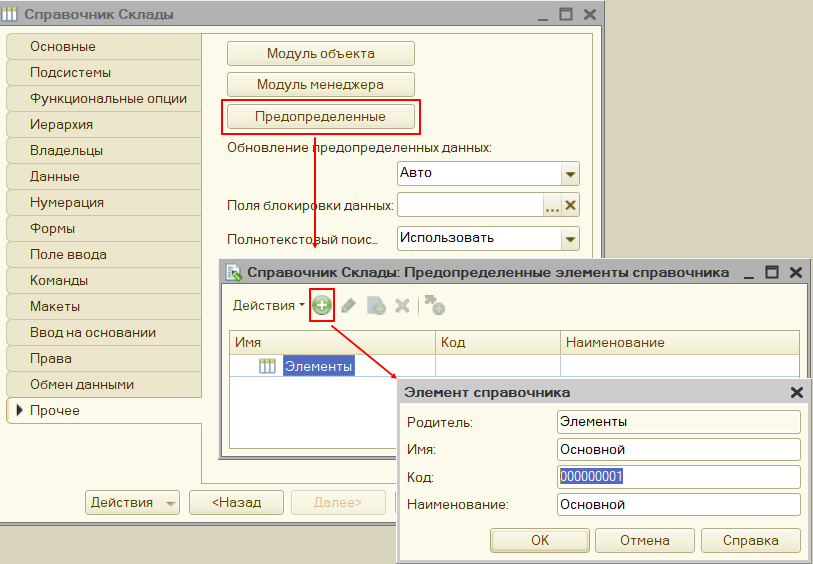
1. Создайте справочник с предопределенными элементами *Склады*, который будет содержать информацию о складах, используемых ООО «На все руки мастер». Этот справочник должен включать в себя один предопределенный элемент – склад *Основной*, на который будут поступать все материалы.

Установите свойство *Быстрый выбор*, позволяющее выбирать элементы не из отдельной формы, а из небольшого выпадающего списка, заполненного элементами этого справочника.

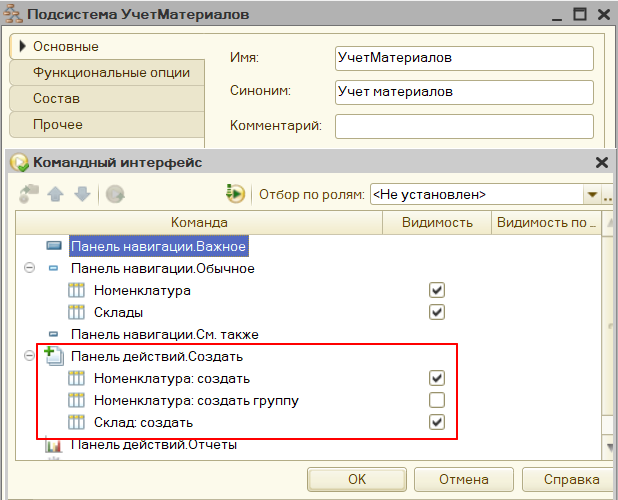


Заполните предопределенный элемента справочника *Основной*.

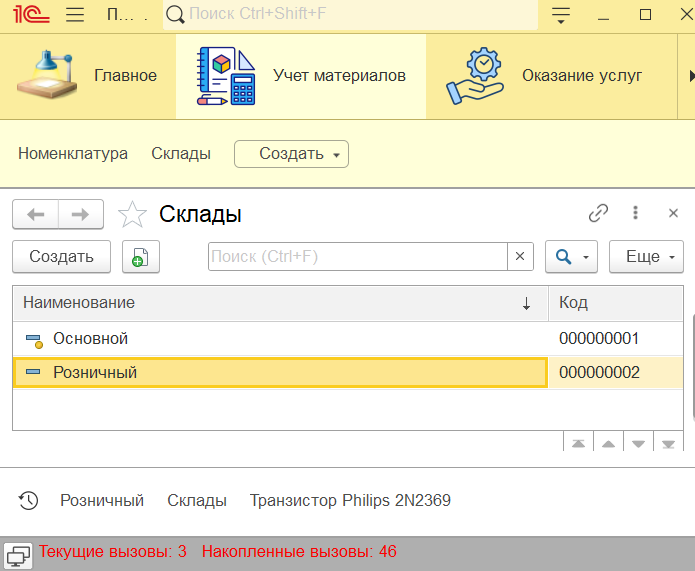


Настройте интерфейс приложения, чтобы было удобнее вводить новые элементы справочника. В разделе *Учет материалов* сделайте доступной стандартную команду для создания новых складов.

Для этого откройте окно редактирования объекта конфигурации *Подсистема УчетМатериалов* и нажмите кнопку *Командный интерфейс*. В открывшемся окне *Командный интерфейс* отразятся все команды этой подсистемы. В группе *Панель действий.Создать* включите видимость у команды *Склад: создать*.



1. Обновите конфигурацию. Изучите изменения в режиме «1С:Предприятие».
2. Выполните команду *Склады* в разделе *Учет материалов*. Откроется форма списка складов. Добавьте в справочник еще один склад *Розничный*.



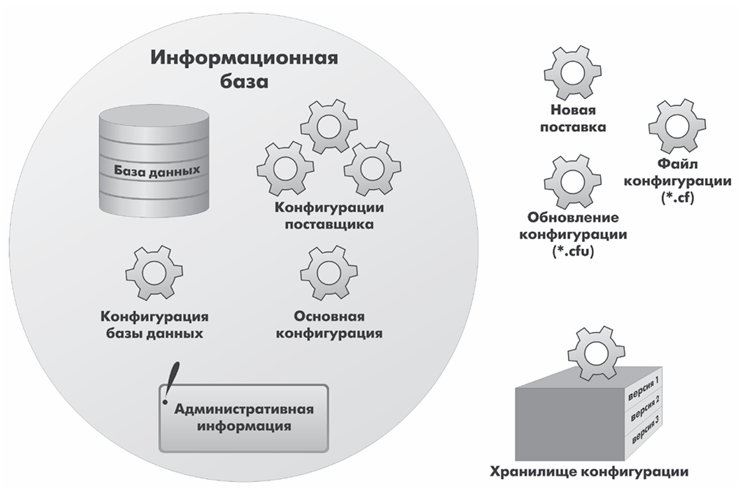
Обратите внимание, что система отмечает различными пиктограммами обычный и предопределенный элементы справочника.

1. Изучите материал.

Конфигурация, предназначенная для разработчика, называется *Основная конфигурация* (или просто Конфигурация – та, которую редактируем в конфигураторе). Конфигурация, с которой работают пользователи, называется *Конфигурация базы данных*. Основную конфигурацию можно редактировать. Конфигурацию базы данных редактировать нельзя, можно только произвести обновление конфигурации базы данных на основе основной конфигурации.

В общем случае информационная база «1С:Предприятия» может хранить более двух конфигураций: основную конфигурацию, конфигурацию базы данных и несколько конфигураций поставщиков. Кроме этого, вне информационной базы может существовать хранилище. В нем находится конфигурация, предназначенная для групповой разработки. Вне информационной базы может существовать также некоторое количество файлов конфигураций, в том числе файлы новой поставки.

Структура конфигурации.



Конфигурация поставщика, находящаяся в информационной базе, содержит предыдущее состояние поставки. Возможна ситуация, когда конфигурация находится на поддержке одновременно у нескольких поставщиков, каждый из которых поддерживает только свою часть в виде отдельной конфигурации. В этом случае информационная база будет хранить несколько конфигураций поставщиков (состояние предыдущей поставки для каждого поставщика отдельно). Файлы новой поставки могут существовать в виде файлов конфигураций (полная поставка) и файлов обновлений (поставка обновлений).

Хранилище конфигурации содержит конфигурацию, предназначенную для групповой разработки. Она хранится не в виде единой конфигурации, а в виде отдельных объектов в разрезе версий конфигурации. Таким образом, мы можем получить из хранилища конфигурацию любой версии – для этого она «собирается» из объектов нужной версии.

Теперь представьте, что между всеми этими видами конфигураций существует возможность сравнения и обновления. В этом случае очень легко запутаться, и название Основная конфигурация как нельзя лучше отражает конечную цель всех изменений.

Теперь, возвращаясь к основной конфигурации и конфигурации базы данных, нужно заметить, что внутреннее разделение на две конфигурации позволяет вносить изменения, не прерывая работы пользователей, потому что изменения вносятся в основную конфигурацию, с которой пользователи не работают.

Затем, когда разработчик будет уверен в том, что все изменения, которые он внес, верны, можно будет быстро произвести обновление конфигурации базы данных, используя основную конфигурацию.

Если эти изменения не затрагивают структуру базы данных (например, если не нужно изменять таблицы, если поменялся только программный код в каком-то модуле), то обновить конфигурацию базы данных можно не прерывая работы пользователей. Это так называемое *динамическое обновление*.

Пользователи увидят изменения только после того, как перезапустят свое приложение. Используя метод встроенного языка *КонфигурацияБазыДанных ИзмененаДинамически()*, можно определить программно, нужно ли перезапускать приложение.

Но если изменения касаются структуры базы данных, например, добавился новый реквизит у справочника или изменился тип существующего реквизита, то тогда требуется реструктуризация базы данных и нужно завершить работу всех пользователей.

Разработчик всегда может сравнить основную конфигурацию и конфигурацию базы данных, может вернуться к исходному состоянию основной конфигурации, используя конфигурацию базы данных (если, например, совсем запутался в своих изменениях).

Таким образом, взаимодействие двух конфигураций можно представить следующим образом.



Когда разработчик работает с основной конфигурацией, система всегда подсказывает ему, отличается ли его вариант основной конфигурации от того, который сохранен, и отличается ли сохраненный вариант основной конфигурации от конфигурации базы данных.

Если разработчик редактирует основную конфигурацию и редактируемый вариант основной конфигурации отличается от того, который сохранен, в заголовке окна дерева конфигурации появляется признак модифицированности конфигурации (\*).



Если сохраненный вариант основной конфигурации отличается от конфигурации базы данных, в заголовке окна дерева конфигурации появляется знак отличия конфигураций (<!>).



Для сохранения основной конфигурации следует использовать команду *Конфигурация* → *Сохранить конфигурацию*, а для обновления конфигурации базы данных – команду *Конфигурация* → *Обновить конфигурацию базы данных*.

При выполнении команды *Отладка → Начать отладку* система сама сначала сохраняет основную конфигурацию, а затем производит ее сравнение с конфигурацией базы данных. Если конфигурации отличаются, выдается запрос на обновление конфигурации базы данных, который вы видели в предыдущих примерах.

При выполнении команды *Отладка → Продолжить* система после описанных выше действий предлагает еще и перезапустить приложение, чтобы прекратить текущий сеанс, запущенный в режиме отладки.

Таким образом, система старается облегчить жизнь разработчика и автоматизировать часто выполняемые операции.

Важным фактом является то, что именно в момент обновления конфигурации базы данных система создает (модифицирует) в базе данных те структуры хранения данных, которые описали в виде объектов конфигурации.

Таким образом, обычные элементы справочника пользователь добавляет в ту структуру базы данных, которую создала система на основе объекта конфигурации *Справочник*.

Предопределенные элементы этого справочника система добавляет в эту структуру сама, на основе того описания, которое содержится в объекте конфигурации *Справочник*.

Отсюда следует немаловажный факт (о котором говорилось в предыдущем разделе), что если простые элементы справочника «безразличны» для конфигурации, то предопределенные элементы важны для нее, поскольку на них могут быть завязаны алгоритмы работы конфигурации.

**КОНТРОЛЬНЫЕ ВОПРОСЫ**

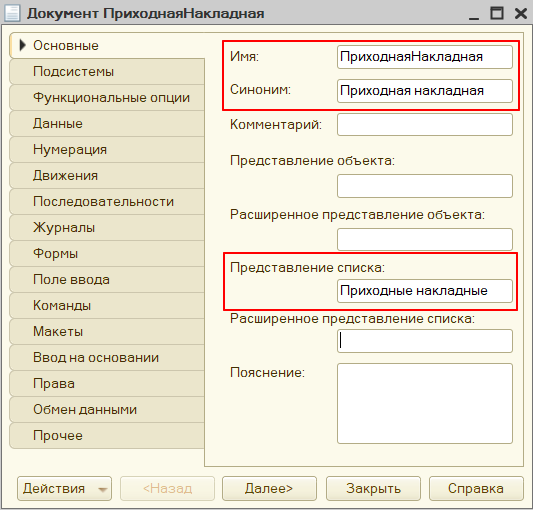
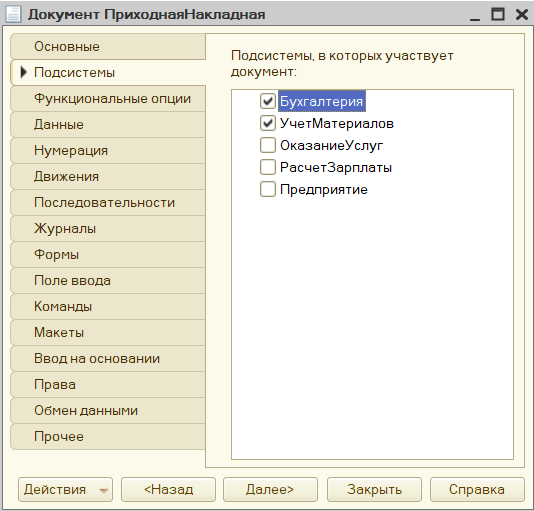
* Для чего предназначен объект конфигурации «Справочник»?
* Каковы характерные особенности справочника?
* Для чего используются реквизиты и табличные части справочника?
* Зачем нужны иерархические справочники, и что такое родитель?
* Зачем нужны подчиненные справочники, и что такое владелец?
* Какие основные формы существуют у справочника?
* Что такое предопределенные элементы?
* Чем с точки зрения конфигурации отличаются обычные элементы справочника от предопределенных элементов?
* Как пользователь может отличить обычные элементы справочника от предопределенных элементов?
* Как создать объект конфигурации «Справочник» и описать его структуру?
* Как задать синоним стандартного реквизита?
* Как добавить новые элементы в справочник?
* Как создать группу справочника?
* Как переместить элементы из одной группы справочника в другую?
* Зачем нужна основная конфигурация и конфигурация базы данных?
* Как изменить конфигурацию базы данных?
* Как связаны объекты конфигурации и объекты базы данных?
* Что такое подчиненные объекты конфигурации?
* Зачем нужна проверка заполнения у реквизитов справочника?
* Что такое быстрый выбор, и когда его использовать?
* Как отобразить справочник и определить его представление в различных разделах интерфейса приложения?
* Как отобразить команды создания нового элемента справочника в интерфейсе подсистем?
* Как редактировать командный интерфейс подсистем?
* Какие стандартные панели используются в интерфейсе приложения, и как выполнить настройку расположения этих панелей в конфигураторе и в режиме «1С:Предприятие»?

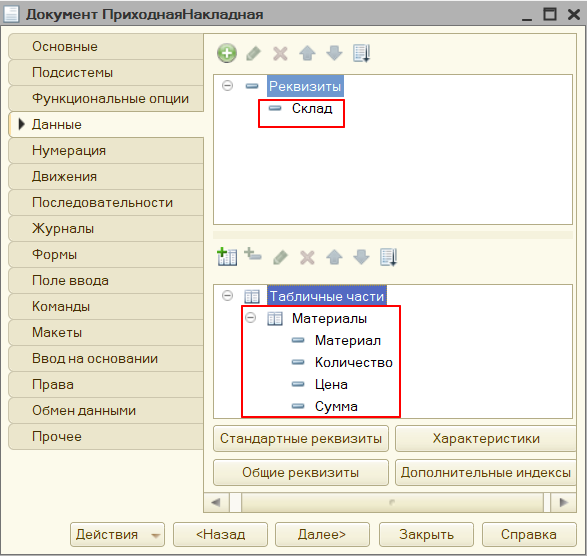
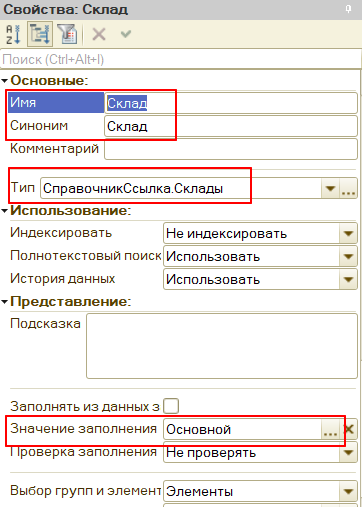
**ЧАСТЬ 3. ДОКУМЕНТЫ**

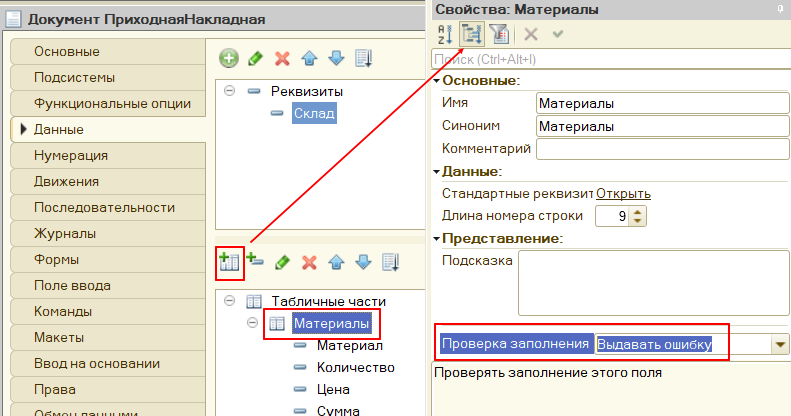
1. Создайте два документа: *Приходная накладная* и *Оказание услуги*.

Документ *Приходная накладная* будет фиксировать факт поступления в нашу организацию необходимых материалов, а документ *Оказание услуги* – фиксировать оказание услуг и расход материалов, которые используются при оказании этих услуг.

* **Приходная накладная**.

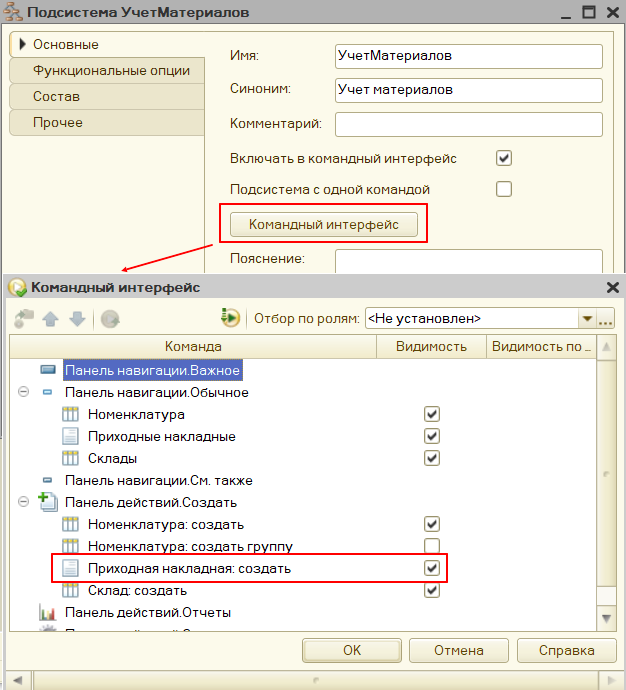


Создайте реквизиты табличной части Материалы:

* *Материал*, тип *СправочникСсылка.Номенклатура*;
* *Количество*, тип *Число*, длина 15, точность 3, неотрицательное;
* *Цена*, тип *Число*, длина 15, точность 2, неотрицательное;
* *Сумма*, тип Число, длина 15, точность 2, неотрицательное.

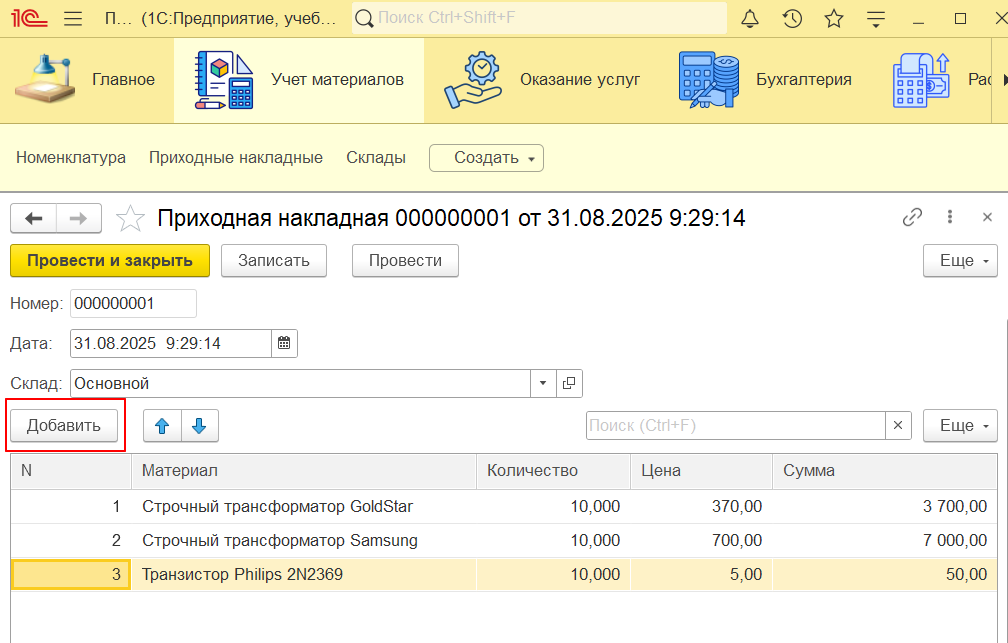
Для каждого реквизита табличной части также установим свойство *Проверка заполнения* в значение *Выдавать ошибку*. Тем самым при записи документа будет проверяться на заполнение не только табличная часть в целом, но и ее отдельные реквизиты.

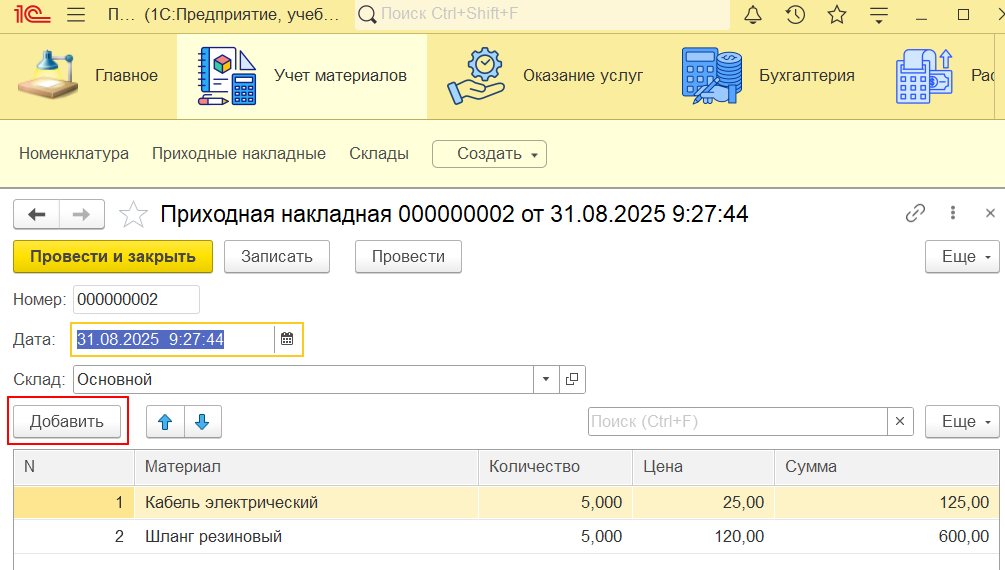
Отредактируйте командный интерфейс, чтобы в разделе *Учет материалов* была доступна команда создания новых документов.



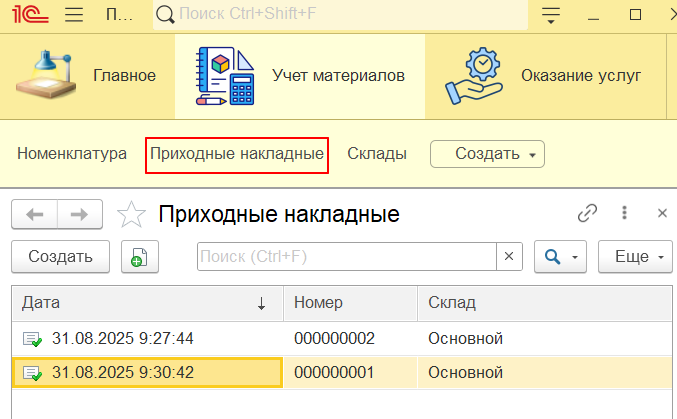
Сохраните изменения, проверьте визуализацию в режиме «1С:Предприятие».

Создайте две приходные накладные: подсистема *Учет материалов → Создать → Приходная накладная*.





Проверьте список созданных документов, выполнив команду *Приходные накладные*. Зеленая галочка в пиктограмме документа означает, что документы проведены.



ВНИМАТЕЛЬНО ИЗУЧИТЕ МАТЕРИАЛ:

**Теория: справочники и документы**

Интересно обратить внимание на разницу в употреблении единственного и множественного числа при именовании объектов вида *Справочник* и *Документ*.

Если вы откроете дерево типовой конфигурации, то увидите, что все объекты, расположенные в ветке *Справочники*, как правило, именованы во множественном числе. В ветке же *Документы*, как правило, в единственном числе.

Может сложиться впечатление, что, создавая объект конфигурации *Справочник*, мы делаем хранилище для элементов этого справочника, конкретного документа. На самом деле это не так.

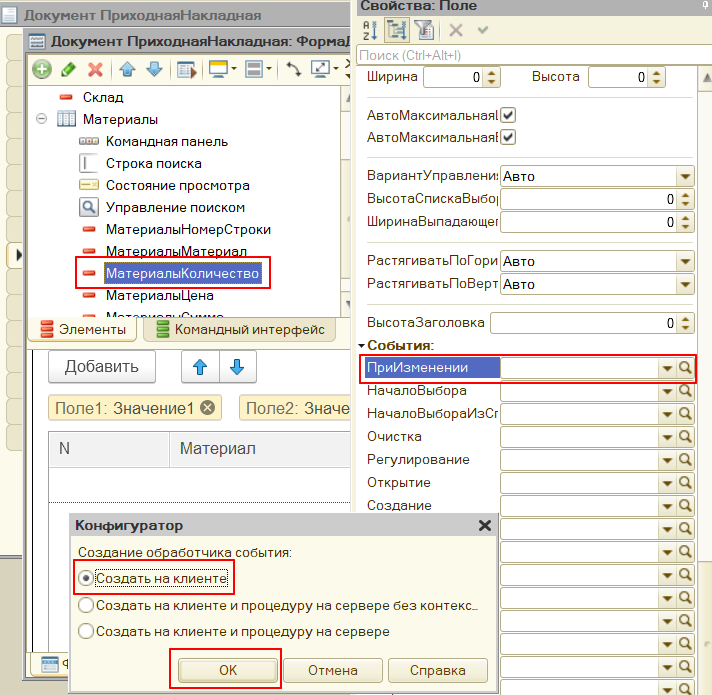
Создавая в конфигураторе объект вида *Справочник*, мы даем ему наименование во множественном числе (*Товары*). При этом подразумевается, что в базе данных этот объект будет состоять из элементов, описывающих конкретные товары (в единственном числе).

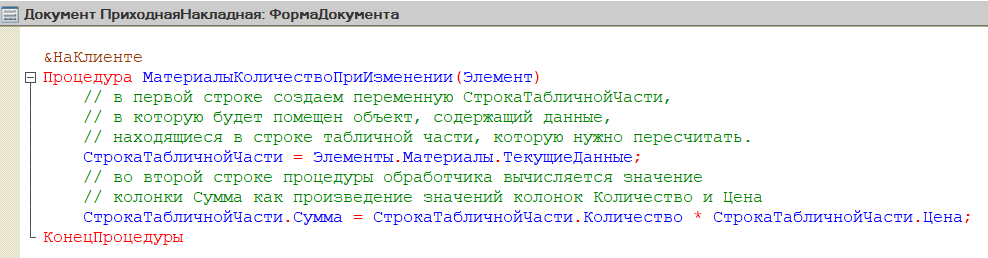
Создавая в конфигураторе объект вида *Документ*, мы даем ему наименование в единственном числе (*ПриходнаяНакладная*), однако на самом деле мы создаем такое же хранилище, как и в случае со справочником. Каждая запись этого хранилища будет описывать один документ, одну приходную накладную (в единственном числе).

Поэтому концептуально правильно было бы в конфигураторе задавать наименование объекта вида Документ во множественном числе, подчеркивая тем самым описание набора документов этого вида (например, *ПриходныеНакладные*).

Однако психология человека такова, что, открывая ветку *Документы*, он ожидает увидеть перечисление их в единственном числе, а никак не во множественном. Так происходит потому, что в реальной жизни трудно найти подходящий термин для описания совокупности документов одного вида (совокупность записей одного вида обозначить гораздо проще – справочник, план и т.д.). Поэтому соответствующая ветка объектов конфигурации имеет название *Документы*, а объекты конфигурации, создаваемые в этой ветке, именуются в единственном числе, хотя, по сути, сама ветка содержит описания хранилищ документов разных видов, а каждый элемент в этой ветке описывает набор всех документов одного вида.

Выполните автоматический пересчет суммы в строках *Документа*.



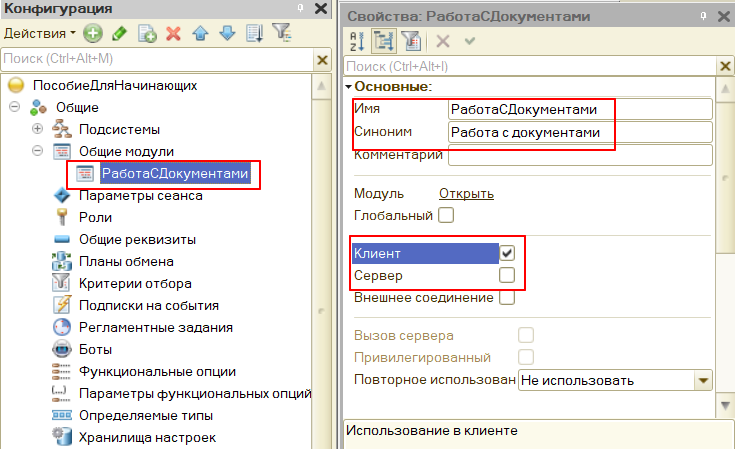


Запустите «1С:Предприятие» в режиме отладки, откройте список документов Приходные накладные и откройте любой из двух созданных нами документов. Если теперь Вы поменяете количество в любой строке документа, то сумма в строке будет пересчитана автоматически.

Но теперь хотелось бы и для поля *Цена* сделать то же самое. А если заглянуть вперед, то увидим, что подобное автоматическое заполнение поля *Сумма* может понадобиться и в других документах.

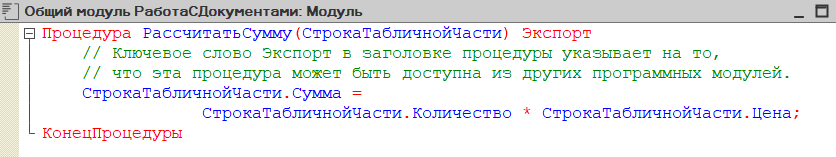
Поместите расчет суммы в некоторое «общедоступное» место, чтобы разные документы, имеющие аналогичные реквизиты табличной части, могли использовать этот алгоритм. Для описания таких «общедоступных» мест служат объекты конфигурации *Общий модуль*, расположенные в ветке *Общие → Общие модули*. Процедуры и функции, содержащиеся в этих модулях, могут быть доступны для любых объектов конфигурации.

Создайте общий модуль и перенесите в него процедуру расчета суммы. А в документе просто оставьте вызовы этой процедуры из общего модуля.

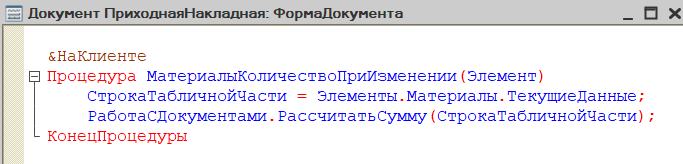


Экземпляры этого модуля будут скомпилированы в контексте тонкого клиента и в контексте веб-клиента.

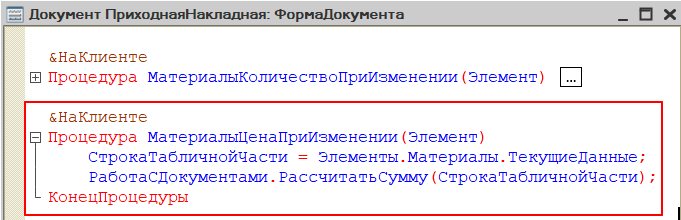
Внесите в модуль процедуру.



Измените в модуле формы текст обработчика МатериалыКоличествоПриИзменении:



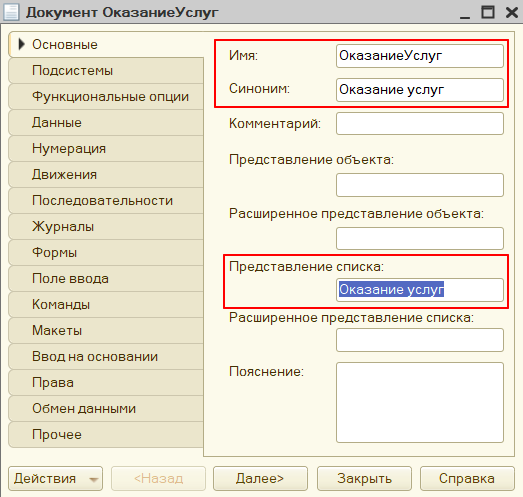
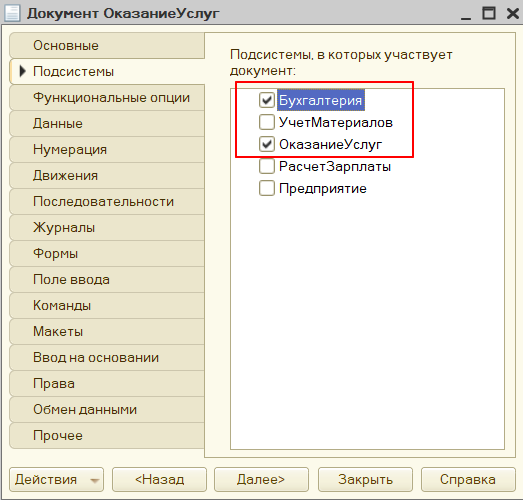
Создайте обработчик события *ПриИзменении* для поля табличной части *МатериалыЦена* так же, как делали это для поля *МатериалыКоличество*, и повторите в нем вызов процедуры *РассчитатьСумму* из общего модуля.



Запустите «1С:Предприятие» в режиме отладки и убедитесь, что теперь сумма в строках табличной части документов *ПриходнаяНакладная* пересчитывается как при изменении количества, так и при изменении цены.

* **Оказание услуг**.

Аналогичным образом создайте второй документ – *Оказание услуги*. Для этого потребуется выполнить те же действия, которые выполняли по созданию документа *Приходная накладная*.

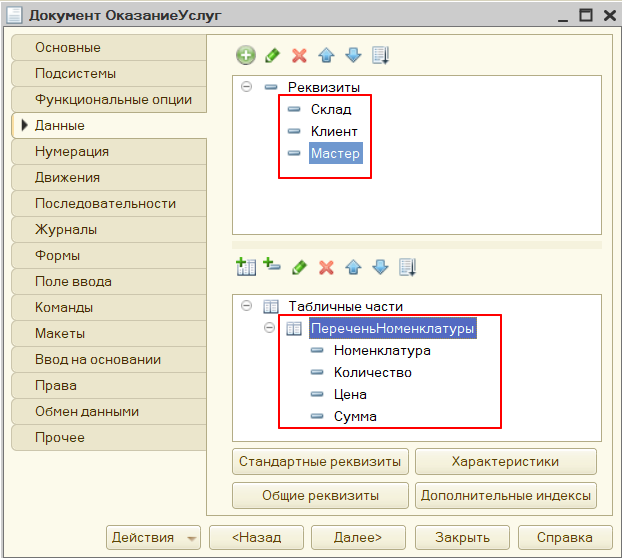
На закладке Данные создайте реквизиты документа:

* *Склад*, тип *СправочникСсылка.Склады*. Выберете для свойства *Значение заполнения* предопределенный элемент *Основной* справочника *Склады*;
* *Клиент*, тип *СправочникСсылка.Клиенты*. Установите свойство *Проверка заполнения* в значение *Выдавать ошибку*;
* *Мастер*, тип *СправочникСсылка.Сотрудники*. Установим свойство *Проверка заполнения* в значение *Выдавать ошибку*.

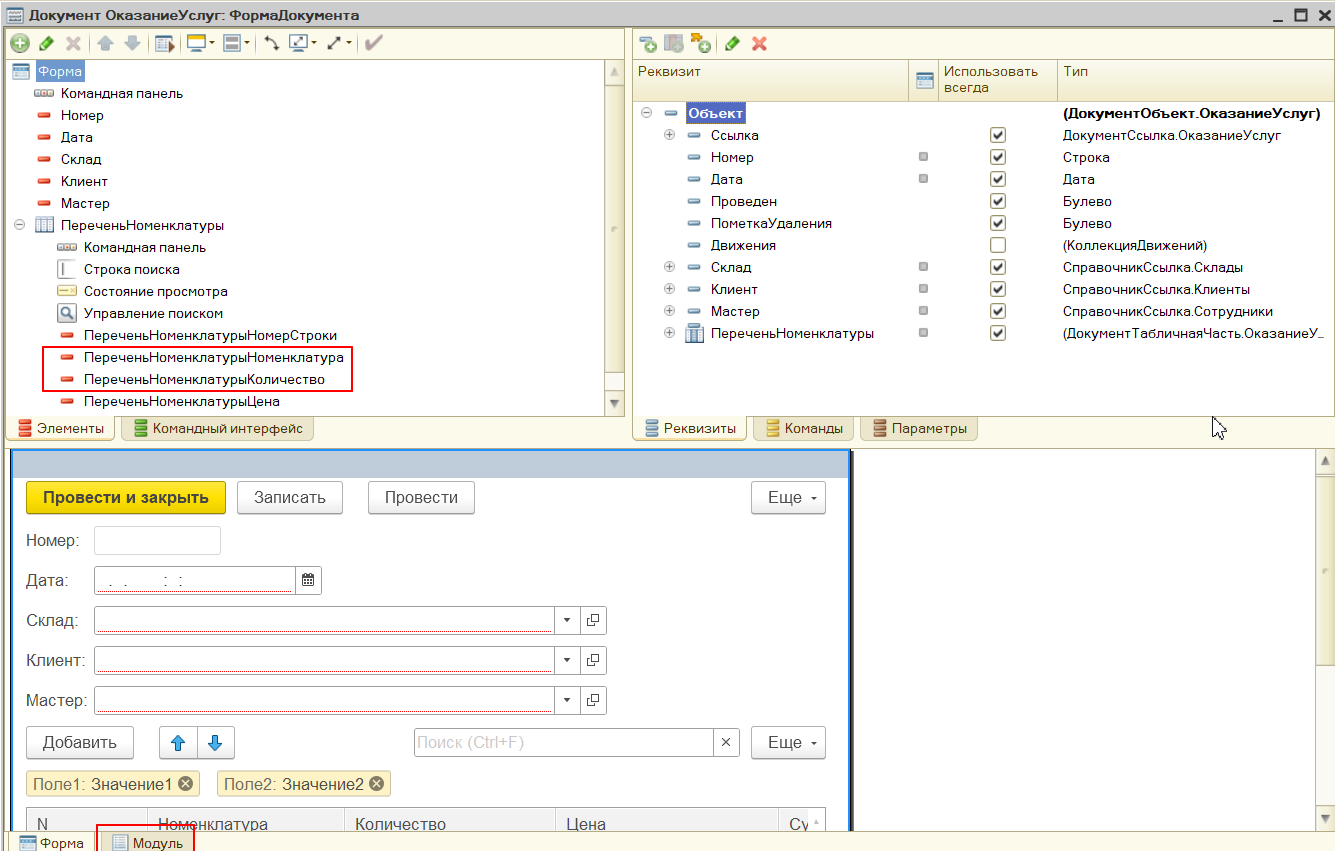
Создайте табличную часть этого документа *ПереченьНоменклатуры* с реквизитами:

* *Номенклатура*, тип *СправочникСсылка.Номенклатура*;
* *Количество*, тип *Число*, длина 15, точность 3, неотрицательное;
* *Цена*, тип *Число*, длина 15, точность 2, неотрицательное,
* *Сумма*, тип *Число*, длина 15, точность 2, неотрицательное.

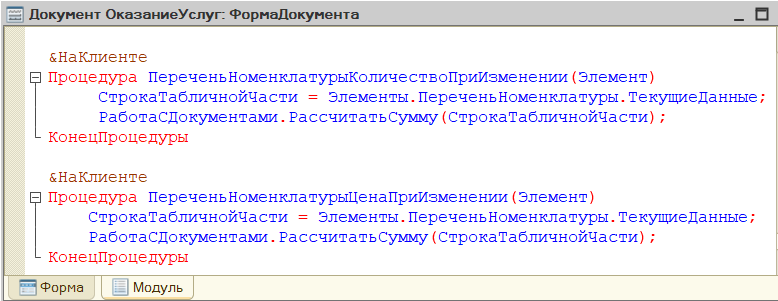
Установим для табличной части в целом и для каждого ее реквизита свойство *Проверка заполнения* в значение *Выдавать ошибку*.



Создайте основную форму документа *Оказание услуг*.

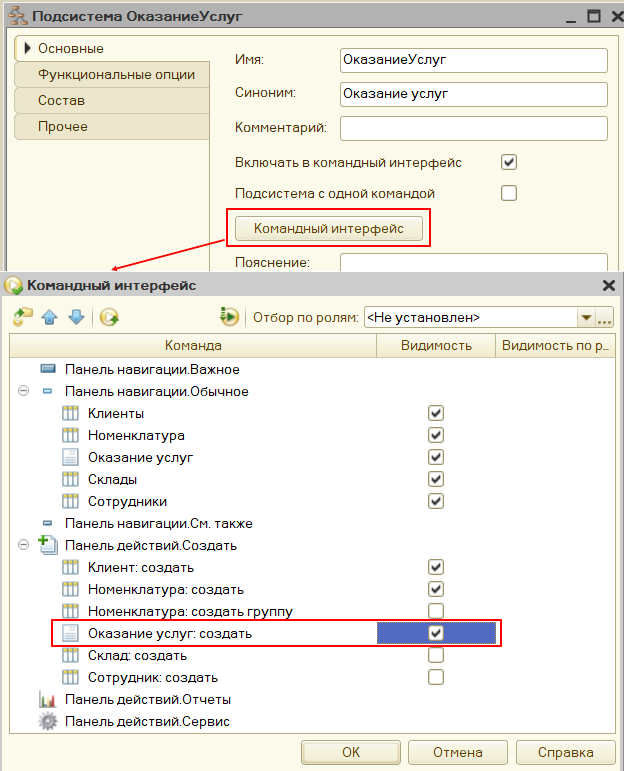


Для полей *ПереченьНоменклатурыКоличество* и *ПереченьНоменклатурыЦена* создайте обработчик события ПриИзменении (выполняющийся на клиенте), в котором будем вызывать процедуру *РассчитатьСумму* из общего модуля *РаботаСДокументами*.



Отредактируйте командный интерфейс, чтобы в разделе *Оказание услуг* была доступна команда создания новых документов.

Для этого откроем окно редактирования объекта конфигурации Подсистема *ОказаниеУслуг* и нажмем кнопку *Командный интерфейс*. В открывшемся окне отразятся все команды данной подсистемы. В группе *Панель действий.Создать* включим видимость у команды *Оказание услуги: создать*.



Запустите «1С:Предприятие» в режиме отладки. В разделе *Оказание услуг* выполните команду создания документа *Оказание услуги* из подменю *Создать* и заполните его следующим образом.

**КОНТРОЛЬНЫЕ ВОПРОСЫ**

* Для чего предназначен объект конфигурации «Документ»?
* Какими характерными особенностями обладает документ?
* Для чего предназначены реквизиты и табличные части документа?
* Какие существуют основные формы документа?
* Что такое проведение документа?
* Как создать объект конфигурации «Документ» и описать его основную структуру?
* Как создать новый документ и заполнить его данными?
* Как создать собственную форму документа?
* Что такое конструктор форм?
* Что такое редактор форм?
* Что такое элементы формы?
* Что такое события, и с чем они связаны?
* Что такое обработчик события, и как его создать?
* Что такое модуль, и для чего он нужен?
* Зачем нужны общие модули?
* Что такое типообразующие объекты?

1. Механизмы основных форм.

Вы создали форму документа *Приходная накладная* и назначили эту форму основной. Что это значит?

У всех прикладных объектов конфигурации существует некоторое количество основных форм. Они служат для отображения данных объекта в том или ином виде.

Если разработчик не назначит в качестве основных форм объекта свои собственные, система будет генерировать необходимые формы объекта самостоятельно, в те моменты, когда к ним происходит обращение. Наличие такого механизма позволяет разработчику не тратить время на создание форм для тестирования своей разработки, а воспользоваться тем, что платформа создаст по умолчанию.

Создание этих форм происходит динамически, в процессе работы системы. Форма создается в тот момент, когда к ней происходит обращение. Причем не важно, интерактивное это обращение или программное.

Так, форма списка для справочника *Клиенты* будет создана как при интерактивном выборе в меню *Все функции → Справочники → Клиенты*, так и при программном вызове глобального метода *ПолучитьФорму()*:

*ФормаСписка = ПолучитьФорму("Справочник.Клиенты.ФормаСписка");*

Также примечательным фактом является то, что состав основных форм, определенных для объекта конфигурации, может не совпадать с перечнем тех форм, которые вообще можно создать для данного объекта, используя конструктор формы.

Например, для большинства регистров в конфигураторе можно задать основную форму списка. Однако если открыть конструктор форм для регистра, вы увидите, что кроме формы списка предлагается создать и форму набора записей регистра, которая отсутствует в перечне основных форм.

Дело в том, что состав основных форм определяется исходя из того, какое представление данных может понадобиться в процессе интерактивной работы пользователя. Для этих представлений разработчик может создать свои формы и указать их в качестве основных, а может использовать те формы, которые система создаст автоматически. Конструктор форм, напротив, исходит из потребностей разработчика. Если разработчик посчитает нужным использовать для какого-либо

регистра вместо обычной формы списка форму набора записей, он сможет это сделать, воспользовавшись конструктором и определив ее в качестве основной формы регистра. Но для логики работы системы это не будет иметь принципиального значения.

1. Обработчики событий.

Вы создавали обработчики событий *ПриИзменении* у некоторых элементов формы. Что это такое?

При работе с событиями на платформе «1С:Предприятие» следует различать два типа событий: события, связанные с формой и ее элементами, и все остальные. Разница заключается в том, что обработчики событий, связанных с формой и ее элементами, – назначаемые, а обработчики всех остальных событий – фиксированные. Фиксированный обработчик события должен иметь имя, совпадающие с именем события. Только в этом случае он будет вызываться при возникновении соответствующего события.

Назначаемый обработчик может иметь произвольное имя. Если имя процедуры совпадает с именем события формы или ее элемента, этого совсем недостаточно для вызова процедуры обработки события с таким именем. Требуется явное назначение процедуры обработчиком этого события в палитре свойств, в соответствующем событии.

Таким образом, любая процедура, расположенная в модуле формы, может быть назначена обработчиком любого события (или сразу нескольких событий) формы или ее элемента, расположенного в форме. Имя процедуры в этом случае не имеет значения. Важно лишь то, что она назначена обрабатывать какое-либо событие. Назначение обработчика может выполняться интерактивно, при работе с формой в конфигураторе, или программно, используя методы формы и ее элементов – *УстановитьДействие()*.

1. **Анализ кода с помощью отладчика**.

Отладчик – вспомогательный инструмент, облегчающий разработку и отладку программных модулей системы «1С:Предприятие».

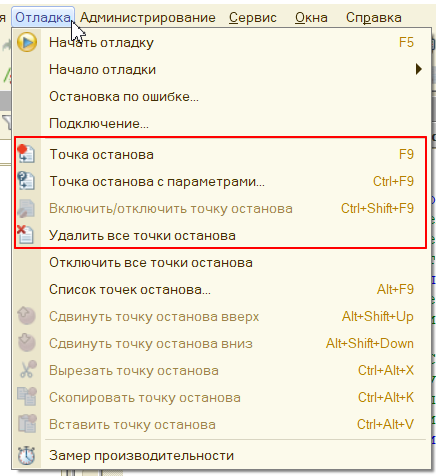
Отладчик предоставляет следующие возможности:

* пошаговое выполнение модуля,
* расстановка точек останова,
* прерывание и продолжение выполнения модуля,
* возможность отладки нескольких модулей одновременно,
* вычисление выражений для анализа состояния переменных,
* просмотр стека вызовов процедур и функций,
* возможность остановки по возникновению ошибки,
* возможность редактирования модуля в процессе отладки.

Рассмотрим использование отладчика для того, чтобы разобраться с обработчиком события *МатериалыКоличествоПриИзменении*.

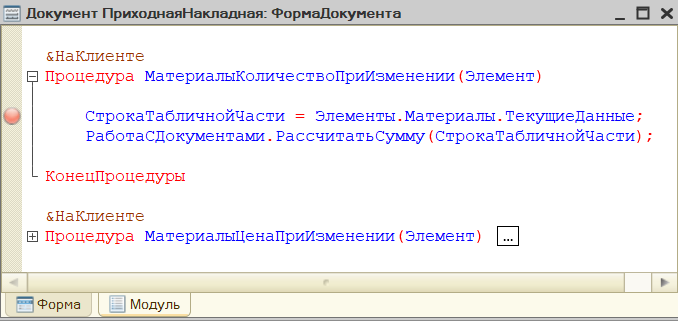
Если в режиме Конфигуратор редактируется текст модуля, то становятся доступными команды пункта главного меню Отладка, позволяющие расставлять и убирать точки останова. Точки останова позволяют прерывать выполнение программы в тех местах, где они установлены. Затем разработчик может проанализировать значение и тип выражений и переменных модуля в момент остановки и продолжить выполнение программы до следующей точки останова и т.д.

Откройте форму документа *ПриходнаяНакладная*, перейдите на закладку *Модуль*, откройте текст процедуры *МатериалыКоличествоПриИзменении*. В пункте главного меню *Отладка* и на панели инструментов конфигуратора стали доступны команды для работы с точками останова.



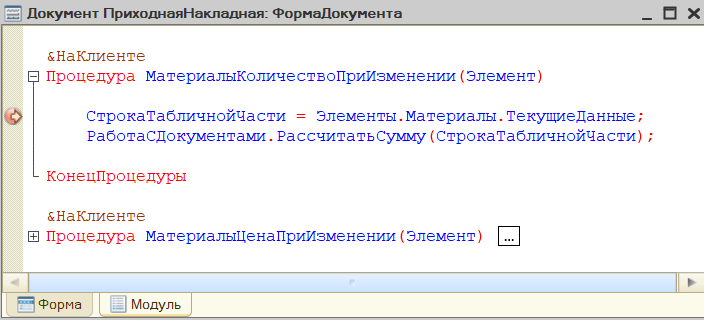
Установить точку останова можно двойным щелчком мыши в служебной области слева от нужной строки. При этом в служебной области, располагающейся вдоль левой границы, появится значок точки останова. Снять точку останова можно двойным щелчком мыши на этой точке в служебной области. Для снятия всех точек останова нужно выполнить команду *Отладка → Отключить все точки останова* или нажать соответствующую кнопку на панели инструментов *Точки останова*.

Дважды щелкните в служебной области слева от первой строки процедуры *МатериалыКоличествоПриИзменении*.

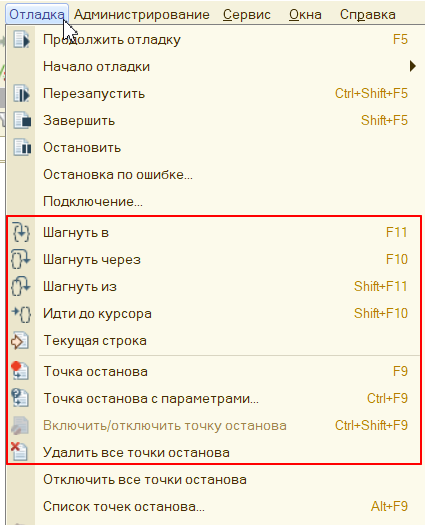


Чтобы иметь возможность отлаживать код на встроенном языке, нужно обеспечить запуск приложения, в котором исполняется код, в отладочном режиме. Для начала отладки выполним команду *Отладка → Начать отладку* или нажмем соответствующую кнопку на панели инструментов конфигуратора. Конфигуратор запустит «1С:Предприятие» в отладочном режиме. На самом деле мы так делали и раньше, но не устанавливали точек останова, и программа не прерывалась.

Откройте список документов *Приходные накладные* и откройте любой из двух созданных нами документов. Измените поле Количество в любой строке документа. Выполнение программы прервется, и в конфигураторе будет открыта процедура *МатериалыКоличествоПриИзменении* в точке останова. Рядом с ней появится стрелка, указывающая на текущую исполняемую строку модуля.



Видим, что в пункте главного меню *Отладка* и на панели инструментов конфигуратора стали доступны команды для работы с конфигурацией в процессе отладки.

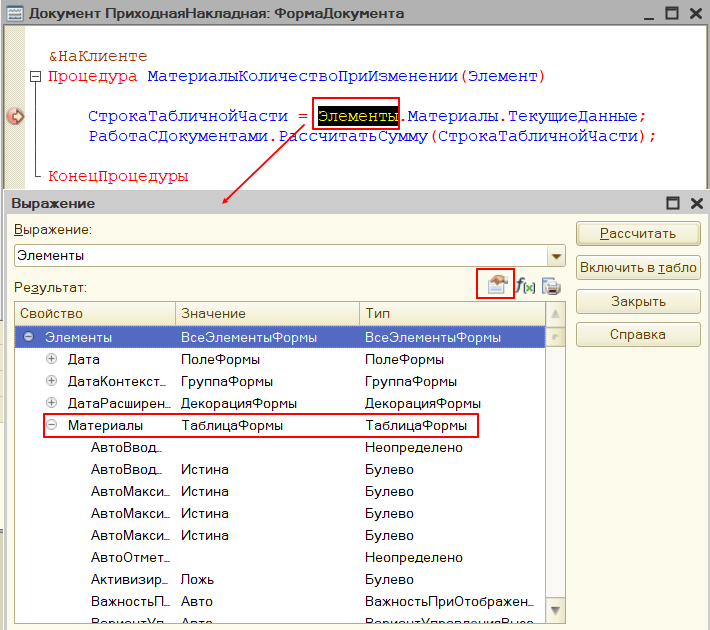


С помощью этих команд и кнопок на панели инструментов *Отладка конфигурации* можно пошагово выполнять программу (кнопки *Шагнуть через*, *Шагнуть в*, *Шагнуть из*) или продолжить отладку (кнопка *Продолжить отладку*) до следующей точки останова. С помощью кнопок *Табло* и *Вычислить* *выражение* можно получить значения интересующих вас выражений в каждый момент остановки программы. С помощью кнопки *Стек вызовов* можно проследить последовательность вызова процедур и функций.

Вы остановились в интересующей нас строке процедуры *МатериалыКоличество ПриИзменении*. Заметим, что эта строка еще не выполнялась, поэтому значения переменных будут еще не заполнены. Чтобы увидеть их после выполнения строки, можно нажать кнопку *Шагнуть через*.

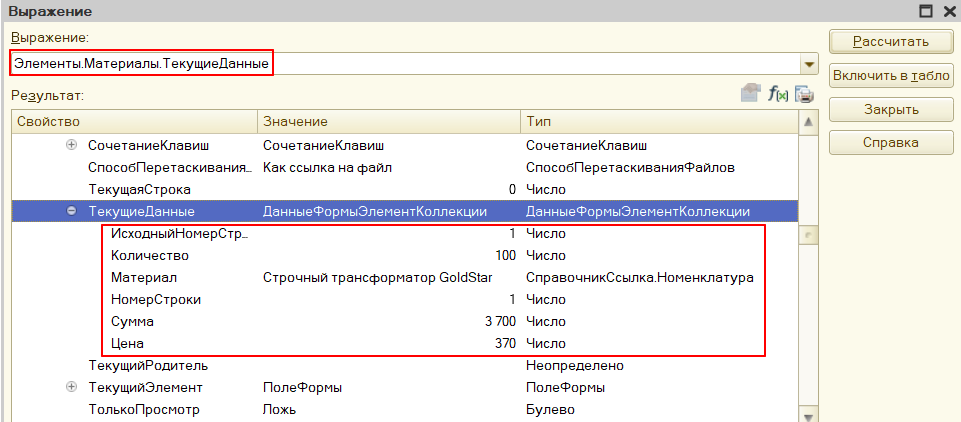
Итак, мы хотим понять, что означает выражение *Элементы.Материалы. ТекущиеДанные* и что содержит этот объект в момент остановки программы. Двойным щелчком выделим слово *Элементы* и нажмем кнопку *Вычислить выражение* (Shift + F9) на панели инструментов *Отладка конфигурации*.

В поле *Выражение* попадет выделенное нами слово *Элементы*. В соответствующих колонках мы увидим значение и тип этого объекта. Мы видим, что объект *Элементы* является коллекцией значений ВсеЭлементыФормы, содержащей все элементы формы. Раскроем этот объект.

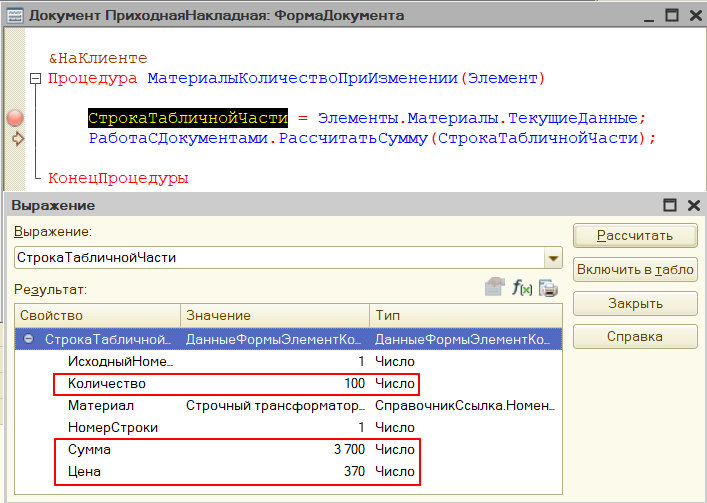


Над окном *Результат* находится кнопка *Показать значения в отдельном окне*, по нажатию которой (или F2) можно будет просматривать содержимое коллекций.

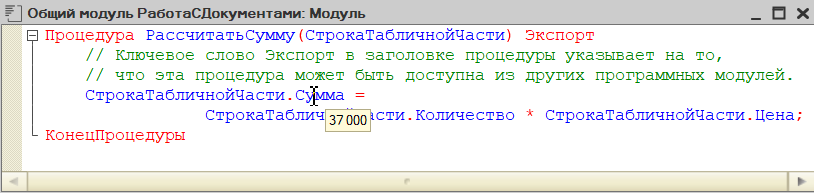
Найдите *Материалы* в списке элементов формы. Это объект *ТаблицаФормы*. Раскройте этот объект и увидите его свойства. Нас интересует свойство *ТекущиеДанные*. Найдите в списке и раскройте этот объект. Видим данные текущей строки нашей табличной части, их значение и тип. Причем *Количество* содержит только что измененное нами значение. Закройте окно *Выражение*.



Теперь нам нужно шагнуть в процедуру общего модуля *РаботаСДокументами*, где вычисляется значение переменной *СтрокаТабличнойЧасти*. Для этого нажмите кнопку *Шагнуть в* на панели инструментов *Отладка конфигурации*. Программа перейдет в процедуру *РассчитатьСумму* общего модуля *РаботаСДокументами*. Двойным щелчком выделите выражение *СтрокаТабличнойЧасти* и нажмите кнопку *Вычислить выражение*. Раскройте объект *СтрокаТабличнойЧасти*.



Видим, что теперь переменная *СтрокаТабличнойЧасти* содержит объект *ДанныеФормыЭлементКоллекции*. Но значения колонки *Сумма* еще не пересчитаны, так как вторая строка кода еще не исполнялась. Закройте окно *Выражение* и еще раз нажмите кнопку *Шагнуть через* на панели инструментов *Отладка конфигурации*. Программа выполнит процедуру Р*ассчитатьСумму* общего модуля *РаботаСДокументами* и остановится в ее конце. Теперь можно подвести курсор к колонке *Количество* или *Сумма*, и система во всплывающей подсказке покажет текущее значение.



Видим, что значения колонки *Сумма* пересчитались по заданному алгоритму, и, значит, объект *Элементы.Материалы.ТекущиеДанные* содержит новые значения.

Таким образом, можно поставить точку останова в тело цикла и смотреть значения переменных по мере его выполнения и т.д.

После анализа переменных и выражений в момент остановки можно продолжить отладку, нажав кнопку *Продолжить отладку*. Если необходимо прервать отладку, нужно выполнить команду *Отладка → Завершить*. В процессе отладки допускается редактирование текущей конфигурации и сохранение изменений.