**Работа с документами**

Объект конфигурации **Документ** предназначен для описания информации совершенных хозяйственных операций или о событиях в организации. Свойства и структура реальных документов описываются в объектах **Документ** на основе которых платформа создает в базе данных таблицы для хранения информации из этих документов.

Документ обладает способностью **проведения**. Факт проведения документа означает, что событие, которое он отражает повлияло на состав учета. До тех пор, пока документ не проведен, состояние учета неизменно, и документ — не более чем черновик, заготовка.

Документ всегда привязан к конкретному времени, что позволяет отражать в базе данных фактическую последовательность событий.

Система 1С: Предприятие умеет отслеживать правильность учета. Пользователь может сам создавать необходимые документы — приходные и расходные накладные, счета, приказы и т.п.

Каждый документ может содержать дополнительную информацию, которая подробнее описывает **реквизиты.** Всегда существуют стандартные реквизиты — **Дата** и **Номер**. Также Документ может содержать **табличную часть** для описания набора данных. Для визуализации документа существует несколько основных **форм**.

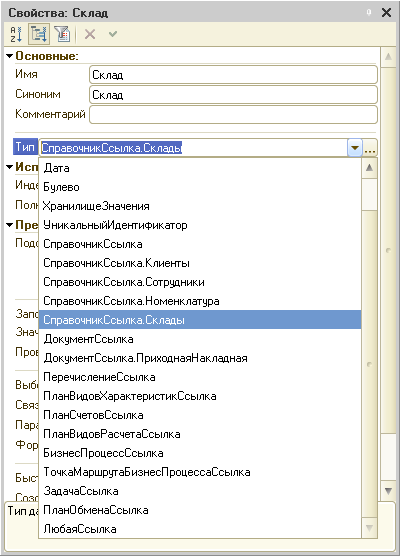
**Типообразующие объекты конфигурации**

Когда мы создавали реквизиты справочников или табличных частей, мы всегда указывали тип значения, которое может принимать этот реквизит. Это были примитивные типы данных — Число, Строка, Дата, Булево. Примитивные типы данных изначально определены в системе и их набор ограничен. В любой конфигурации могут существовать типы данных, определяемые только конкретной конфигурацией, т. е. Появляющиеся в результате добавления некоторых объектов.

### Например, после создания объект *Справочник* Склады, сразу же появились новые типы данных, связанные с этим справочником. Среди них СправочникСсылкаСклады. Если мы укажем какому — либо реквизиту этот тип данных, то сможем сохранитьв нем ссылку на конкретный объект справочника Склады.

**Задание 1. Создание документа Приходная накладная**

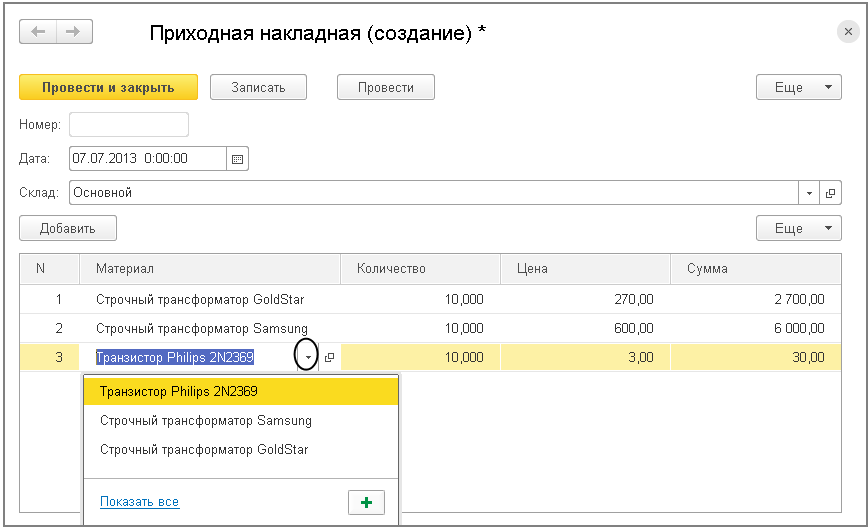
1. Одними из самых популярных услуг нашего предприятия является ремонт телевизоров и установка стиральных машин. И в том, и в другом случае требуются некоторые материалы, которые расходуются в процессе оказания этих услуг. Поэтому двумя важнейшими событиями в хозяйственной жизни нашей организации будут являться поступление материалов и оказание услуг. Для отражения этих событий в базе данных необходимо создать 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.
2. Добавьте новый объект конфигурации **Документ**. На закладке **Основные** задайте имя документа – **ПриходнаяНакладная**. На основании имени платформа автоматически заполнит его синоним. Здесь же определите, как будет представлен документ в интерфейсе **1С:Предприятия**. **Представление объекта** задавать не будем, вместо него будет использоваться **Синоним** объекта. Это нам вполне подходит, так как мы задали его имя в единственном числе. **Представление списка**, наоборот, задайте во множественном числе как **Приходные накладные**. Нажмите**Далее**.
3. Перейдите на закладку **Подсистемы**. По логике нашей конфигурации список приходных накладных должен быть доступен в разделах **Учет материалов** и **Бухгалтерия**. Действительно, к первому разделу этот документ имеет прямое отношение, а для бухгалтерского анализа всегда может понадобиться список документов, отражающих поступление материалов. Поэтому отметьте в списке подсистем эти подсистемы.
4. Перейдите на закладку **Данные** и создайте реквизит документа с именем **Склад**. Для этого нажмите кнопку **Добавить** над списком реквизитов документа. Выберите для реквизита ссылочный тип данных **СправочникСсылка.Склады**. Этот тип стал доступен в конфигурации после создания объекта конфигурации **Справочник Склады**.



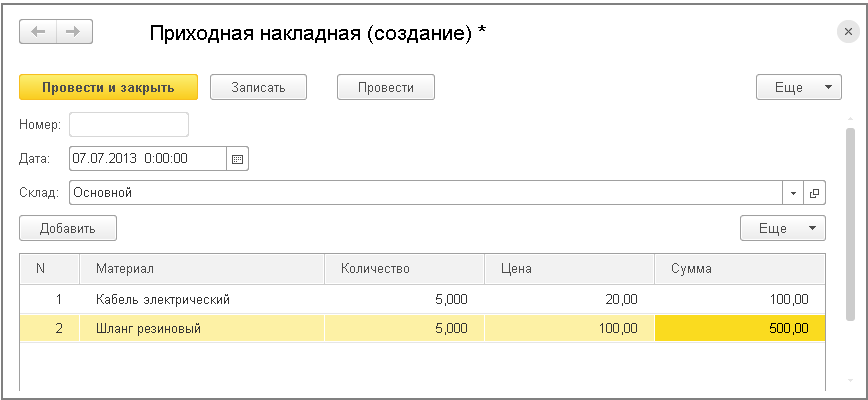
1. Работа в нашей организации построена таким образом, что, как правило, все поступающие товары приходуются на основной склад. Поэтому в палитре свойств для созданного реквизита **Склад** документа найдите свойство **Значение заполнения**. В качестве значения этого свойства выберите предопределенный элемент справочника **Склады – Основной**. Таким образом, при создании нового документа склад будет сразу заполняться значением **Основной**, и пользователю не придется делать это вручную.
2. Добавьте в документ табличную часть с именем **Материалы**. Для этого нажмите кнопку **Добавить табличную часть** над списком табличных частей документа. Кроме имени табличной части установите свойство **Проверка заполнения** в значение **Выдавать ошибку**. Тем самым мы задаем условие, что документ **Приходная накладная** обязательно должен содержать табличную часть, то есть список приходуемых материалов. Иначе будет выдано сообщение об ошибке, и документ не будет сохранен.
3. Создайте реквизиты табличной части **Материалы**. Для этого нажмите кнопку **Добавить реквизит** в разделе описания табличных частей документа:

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

1. Для каждого реквизита табличной части также установите 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.
2. Перейдите на закладку **Нумерация** и убедитесь, что свойство **Автонумерация** включено. Это обеспечит автоматическую генерацию уникальных номеров для создаваемых нами документов.
3. В заключение отредактируйте командный интерфейс, чтобы в разделе **Учет материалов** была доступна команда создания новых документов.
4. 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 **Приходная накладная: создать.**
5. Запустите**1С:Предприятие** в режиме отладки и протестируйте получившийся документ.В открывшемся окне 1С:Предприятиявы видите, что в разделах **Бухгалтерия** и **Учет материалов** появилась команда **Приходные накладные** для открытия списка приходных накладных. Название этой команды определяется свойством **Представление списка**, которое вы задали для этого документа. Кроме того, в разделе **Учет материалов** в подменю **Создать** добавилась команда **Приходная накладная** для создания новых документов этого вида. Название этой команды определяется синонимом, так как **Представление объекта** мы для этого документа не задавали.
6. Пока в нашей базе данных нет ни одного документа **Приходная накладная**, поэтому выполните команду **Приходная накладная** из подменю **Создать** в разделе **Учетматериалов** и создайте новую приходную накладную. Откроется форма документа. Заголовок этой формы **Приходная накладная** совпадает с синонимом документа.
7. Система автоматически подставит текущую дату создания документа и нулевое время, так как документ еще не проведен. В качестве времени документа при оперативном проведении ему присваивается оперативная отметка времени.
8. Поле **Номер** не заполнено, но система сама сгенерирует для нового документа уникальный номер, так как свойство **Автонумерация** для документа включено по умолчанию. Новый номер будет сохранен в момент записи документа.
9. Обратите внимание, что склад уже заполнен значением **Основной**, как было задано в свойствах этого реквизита.
10. Нам осталось только заполнить табличную часть приходной накладной материалами для ремонта телевизоров так, как показано на рисунке.

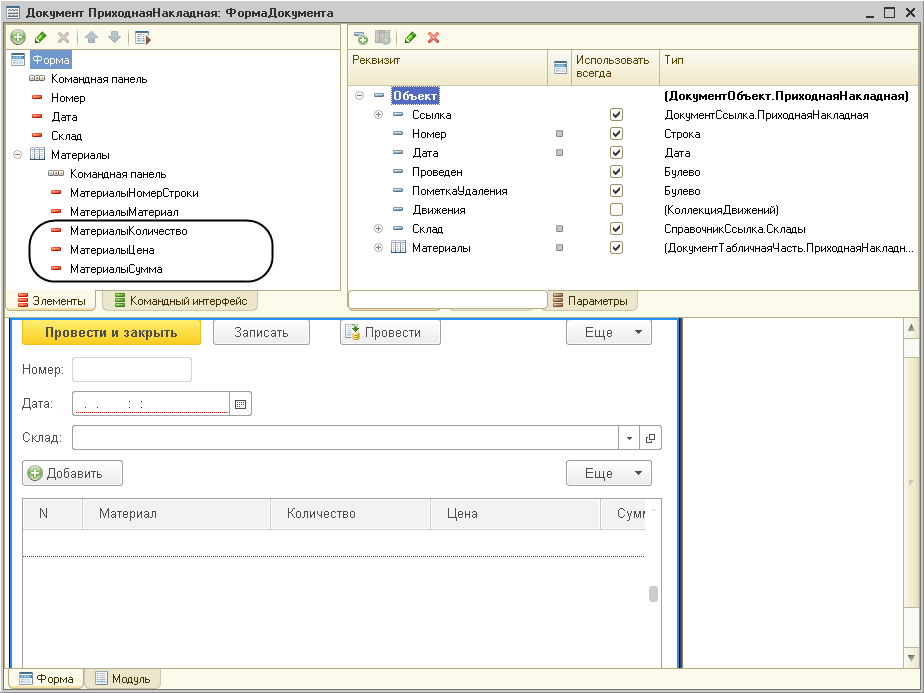


1. Обратите внимание, что при переходе к еще незаполненной колонке **Материал** (в табличной части документа) автоматически открывается окно выбора значения из справочника **Номенклатура**, так как этот реквизит имеет ссылочный тип данных и ссылается на справочник **Номенклатура**. Как только вы начнете вводить название материала в это поле, платформа автоматически найдет материалы, наименование которых начинается с введенных нами символов, и предложит их нам для выбора. А также при нажатии кнопки **Показать все** открывается форма для выбора элементов этого справочника.
2. Нажмите**Провести и закрыть**. Документ будет сохранен и проведен, ему будет присвоен автоматически сгенерированный системой номер и текущее время проведения документа.
3. Аналогичным образом создайте второй документ, который будет приходовать на **Основной** склад материалы для установки стиральных машин.

С

1. Нажмите**Провести и закрыть**. Документ будет сохранен и проведен, ему будет присвоен автоматически сгенерированный системой номер и текущее время проведения документа.
2. Чтобы просмотреть список созданных документов, выполните команду **Приходные накладные**. В форме списка вы увидите два созданных документа, отмеченных пиктограммой, указывающей на то, что документы проведены (зеленая галочка в пиктограмме документа).
3. Наверняка вы обратили внимание на то, что при заполнении документа приходится вводить сумму в каждой строке. Это неудобно, и возникает естественное желание автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества материалов в строке.Это совсем не сложно, и для этого нам потребуется сначала создать собственную форму документа, а затем воспользоваться возможностями встроенного языка. Дело в том, что до сих пор мы использовали предопределенные формы объектов, которые система **1С:Предприятие** по умолчанию создавала для нас сама. Теперь же у нас возникла необходимость слегка изменить логику работы формы документа, поэтому нам нужно создать свою собственную форму документа **ПриходнаяНакладная** для того, чтобы в ней с помощью встроенного языка описать тот алгоритм, который нам нужен. И система будет использовать нашу форму вместо формы по умолчанию.
4. Вернитесь в конфигуратор и откройте окно редактирования объекта конфигурации **Документ ПриходнаяНакладная**. В этом окне нас интересует закладка **Формы**. Как мы видим, ни одна из основных форм документа пока не задана. Для того чтобы создать форму документа, нажмите кнопку открытия http://its.1c.ru/db/content/pubdevguide83/src/pic04_01.png?_=1450873326со значком лупы в поле ввода или кнопку **Добавить** над списком форм. Система вызовет еще один полезный инструмент разработчика – конструктор формы. Этот инструмент также построен по принципу «мастеров»: ввод данных в определенной последовательности и передвижение кнопками **Далее** и **Назад**. Выберите тип формы **Форма документа** и нажмите кнопку **Готово**, согласившись тем самым со всем, что нам предложила система.

Обратите внимание, что в дереве объектов конфигурации у объекта конфигурации **Документ ПриходнаяНакладная** появилась форма **ФормаДокумента**, а на экране открылось окно редактора форм, содержащее эту форму.



1. Редактор форм объединяет несколько окон взаимосвязанных между собой редакторов. Рассмотрим только окно формы документа в режиме просмотра, расположенное внизу, и окно редактора элементов формы, расположенное слева в верхней части окна редактора форм.Прежде всего, необходимо понимать, что при разработке форм объектов конфигурации разработчик не имеет возможности нарисовать форму. Он может только указать, из каких элементов будет состоять форма, а система уже сама самостоятельно расположит эти элементы в форме.

Элементы формы в верхнем левом окне редактора форм образуют иерархическую структуру, из которой следует, что чем выше в списке находится элемент, тем выше и левее на форме он будет располагаться.

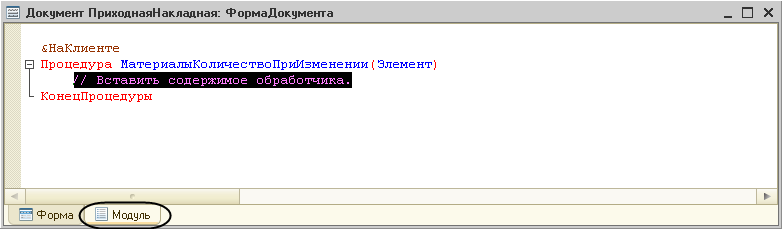
Эта структура редактируется на закладке **Элементы** и позволяет управлять отображением и редактированием данных в форме.

1. На основе описания в конфигурации документа **ПриходнаяНакладная** система создала структуру элементов, которая определяет, как будет выглядеть форма. Эти элементы имеют разное назначение и разное поведение. Однако все они служат для того, чтобы отображать информацию, хранящуюся в базе данных, и организовывать интерактивную работу с этой информацией.

Вы можете попробовать перетащить мышью поля в дереве элементов и поменять местами, например, реквизиты табличной части. Результат изменений сразу отразится в форме документа в нижней части редактора форм. При этом разработчику не нужно задумываться над вопросами конкретного (до пикселя) расположения того или иного элемента формы, его размеров и привязки к другим элементам. Эту работу берет на себя система.

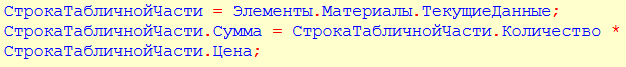
Но разработчик может через палитру свойств изменить свойства элемента, которые повлияют на его отображение в форме. Он может также изменить структуру элементов формы – создать новое поле, группу полей, добавить табличную часть, связав эти элементы с данными формы.

1. Нас интересуют три элемента табличной части: **МатериалыКоличество, МатериалыЦена** и **МатериалыСумма**. Нам нужно, чтобы каждый раз, когда меняется значение в поле **Количество** или в поле **Цена**, в поле **Сумма** автоматически устанавливалось значение, равное произведению значений полей **Количество** и **Цена**. Очевидно, что для этого нужно написать на встроенном языке команду, похожую на **Сумма = Количество\*Цена**, которая будет выполняться при изменении значения поля **Количество** или **Цена**.
2. Система сама умеет работать с теми объектами, которые описаны в дереве конфигурации: показывать их данные, добавлять новые элементы и пр. То есть у нее есть некие «стандартные представления» о том, как это все должно работать. Но, как правило, разработчиков эти «стандартные представления» устраивают только в самых простых случаях. Поэтому у системы существуют события, которые связаны с самыми различными моментами ее «стандартного» функционирования. Используя встроенный язык, разработчик может «вклиниться» в эти события и описать собственный алгоритм того, что должно происходить при наступлении этого события. Что мы сейчас и сделаем.
3. Дважды щелкните на элементе формы **МатериалыКоличество** или правой кнопкой мыши откройте для него палитру свойств (пункт контекстного меню **Свойства**). Прокрутив список до конца, вы увидите перечень событий, которые могут быть связаны с этим полем. Очевидно, что нам нужно событие **ПриИзменении**, которое возникает после изменения значения поля. Найдите его в списке событий и нажмите кнопку открытия http://its.1c.ru/db/content/pubdevguide83/src/pic04_01.png?_=1450873326со значком лупы в поле ввода.
4. Система предложит окно для выбора типа обработчика события, который вы хотите создать. Оставьте без изменения вариант **Создать на клиенте**, предложенный по умолчанию.
5. После этого система создаст шаблон процедуры обработчика этого события в модуленашей формы и откроет закладку **Модуль** редактора формы.



**Модуль** – это «хранилище» для текста программы на встроенном языке. Модулей существует несколько, и каждый из них предназначен для описания алгоритмов, относящихся к тому или иному моменту работы программы. В данном случае это модуль формы, так как обработчики всех интерактивных событий, связанных с элементами формы, помещаются именно в модуль формы.

1. В модуль формы, в процедуру **МатериалыКоличествоПриИзменении()**добавьте следующий текст



В первой строке мы сначала создаем переменную **СтрокаТабличнойЧасти**, в которую будет помещен объект, содержащий данные, находящиеся в строке табличной части, которую нам нужно пересчитать.

Табличная часть документа представляет собой объект встроенного языка **ТаблицаФормы**. Получить ту строку, в которой в настоящее время осуществляется редактирование, можно при помощи свойства программного объекта **ТаблицаФормы – ТекущиеДанные** (**Элементы.Материалы.ТекущиеДанные**).

Таким образом, в результате выполнения первой строки процедуры обработчика переменная **СтрокаТабличнойЧасти** будет содержать объект **ДанныеФормыСтруктура**. Этот объект содержит данные, находящиеся в текущей строке табличной части документа (**Элементы.Материалы.ТекущиеДанные**).

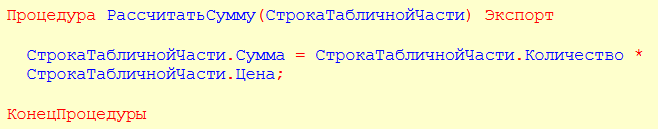
Получив этот объект, мы можем обратиться к данным конкретной колонки табличной части, указав имя колонки в качестве свойства объекта. Например, используя обращение **СтрокаТабличнойЧасти.Количество**, мы получаем число, которое находится в редактируемой строке в колонке **Количество**.

Таким образом, во второй строке процедуры обработчика вычисляется значение колонки **Сумма** как произведение значений колонок **Количество** и **Цена**.

1. Запустите **1С:Предприятие** в режиме отладки, откройте список документов **Приходныенакладные** и откроем любой из двух созданных нами документов. Если теперь вы поменяете количество в любой строке документа, то сумма в строке будет пересчитана автоматически.
2. При изменении количества в любой строке документа **Приходная накладная** сумма в этой строке пересчитывается автоматически. Теперь хотелось бы и для поля **Цена** сделать то же самое. А если заглянуть вперед, то мы увидим, что подобное автоматическое заполнение поля **Сумма** может нам понадобиться и в других документах.

Поэтому лучше будет поместить 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. Поэтому создадим общий модуль и перенесём в него нашу процедуру расчета суммы. А в документе просто оставим вызовы этой процедуры из общего модуля.

1. Добавьте объект конфигурации **Общий модуль**. Для этого раскройте ветвь **Общие** в дереве объектов конфигурации, нажмите + слева от нее. Затем выделите ветвь **Общие модули** и нажмите кнопку **Добавить** в командной панели окна конфигурации.
2. Откроется окно для ввода текста модуля и окно палитры его свойств. Дайте имя модулю – **РаботаСДокументами** и установите в его свойствах флажок **Клиент** (управляемое приложение), а флажок **Сервер** снимите. Это означает, что экземпляры этого модуля будут скомпилированы в контексте тонкого клиента и в контексте веб-клиента.
3. Внесите в модуль следующий текст:

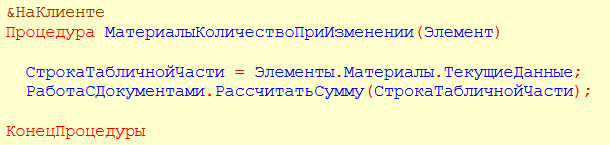


В процедуру **РассчитатьСумму()**передается переменная**СтрокаТабличнойЧасти**, которую мы определили в обработчике события **ПриИзменении** поля **Количество**. Она содержит данные редактируемой строки табличной части документа **ПриходнаяНакладная**.

Теперь, используя эту переменную, мы можем получить доступ к данным колонок табличной части и рассчитать сумму как произведение цены на количество.

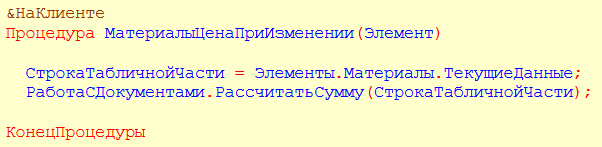
Ключевое слово **Экспорт** в заголовке процедуры указывает на то, что эта процедура может быть доступна из других программных модулей.

1. Теперь в модуле вашей формы измените текст обработчика **МатериалыКоличествоПриИзменении.**



Первая строка процедуры осталась без изменений. А во второй строке вместо непосредственного расчета суммы мы вызываем процедуру **РассчитатьСумму()** из общего модуля **РаботаСДокументами** и передаем ей в качестве параметра текущую строку табличной части.

1. Для поля **Цена** установите такой же обработчик. Так как однажды вы уже написали в модуле формы нужную вам процедуру, то вы просто могли бы сопоставить ее также и другому событию другого элемента управления, расположенного в форме. Однако стандарты разработки конфигураций фирмы **1С** не допускают такого решения. Поэтому создайте обработчик события **ПриИзменении**для поля табличной части **МатериалыЦена** так же, как вы делали это для поля **МатериалыКоличество**, и повторите в нем вызов процедуры **РассчитатьСумму**из общего модуля.



1. 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.

**Задание 2. Создание документа Оказание услуги**

1. Аналогичным образом создайте второй документ **Оказание услуги**. Для этого выполните уже знакомые вам действия, которые вы выполняли по созданию документа **Приходная накладная**.

В режиме **Конфигуратор** добавим новый объект конфигурации **Документ** и назовем его **ОказаниеУслуги**.

На закладке **Основные** определим, как будет представлен документ в интерфейсе «1С:Предприятия».

**Представление объекта** задавать не будем, вместо него будет использоваться Синоним объекта.

**Представление списка** зададим как **Оказание услуг**. На закладке **Подсистемы** отметим, что документ будет доступен в подсистемах **Оказание услуг** и **Бухгалтерия**. На закладке **Данные** создадим реквизиты документа:

**- Склад,** тип **СправочникСсылка.Склады**. Выберем для свойства **Значение заполнения** предопределенный элемент **Основной** справочника **Склады**;

**- Клиент**, тип **СправочникСсылка.Клиенты**. Установим свойство Проверка заполнения в значение Выдавать ошибку;

- **Мастер**, тип **СправочникСсылка.Сотрудники**. Установим свойство Проверка заполнения в значение Выдавать ошибку.

Создадим 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**.

На закладке **Формы** создадим основную форму документа.

Для поля **ПереченьНоменклатурыКоличество** создадим обработчик события **ПриИзменении** (выполняющийся на клиенте), в котором будем вызывать процедуру **РассчитатьСумму** из общего модуля **РаботаСДокументами**.

При этом откроется модуль формы с шаблоном обработчика события **ПереченьНоменклатурыКоличествоПриИзменении**,который мы пока заполнять не будем, а перейдем в окно элементов формы на закладку **Форма** и аналогичным образом создадим обработчик события **ПереченьНоменклатурыЦенаПриИзменении** для поля **ПереченьНоменклатурыЦена.**

Далее модуль формы документа **ОказаниеУслуги** нужно заполнить:

НаКлиенте

Процедура ПереченьНоменклатурыКоличествоПриИзменении(Элемент)

СтрокаТабличнойЧасти = Элементы.ПереченьНоменклатуры.ТекущиеДанные;

РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);

КонецПроцедуры

НаКлиенте

Процедура ПереченьНоменклатурыЦенаПриИзменении(Элемент)

СтрокаТабличнойЧасти = Элементы.ПереченьНоменклатуры.ТекущиеДанные;

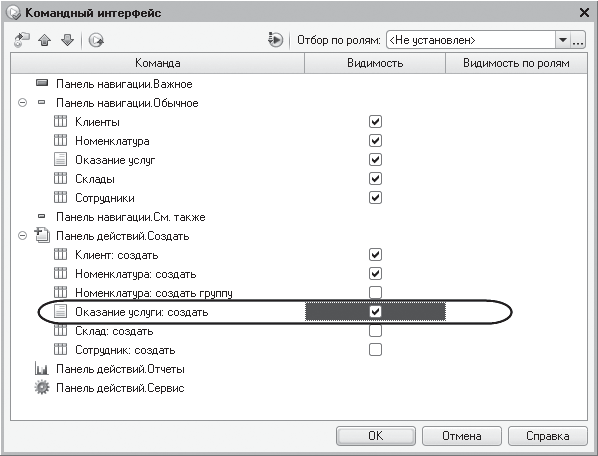
РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);

КонецПроцедуры

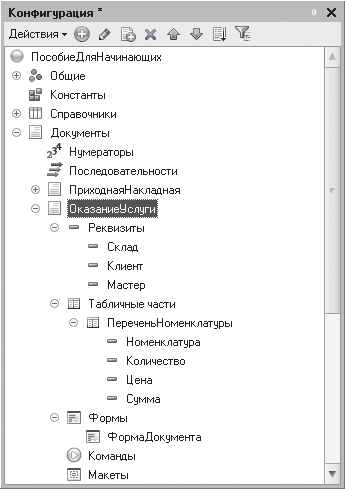
В Заключение отредактируем командный интерфейс, чтобы в разделе **ОказаниеУслуг** была доступна команда создания новых документов.

Для этого откроем окно редактирования объекта конфигурации **Подсистема ОказаниеУслуг** и нажмем кнопку **Командный интерфейс**. В открывшемся окне отразятся все команды данной подсистемы.

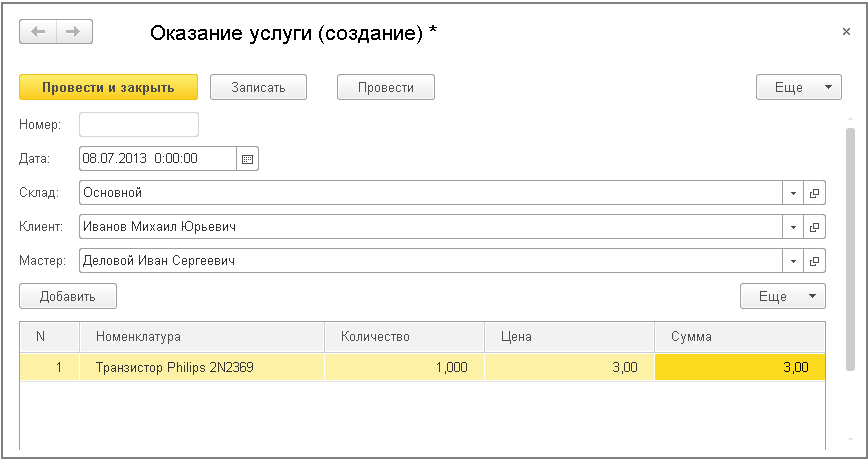
В группе **Панель действий.Создать** включим видимость у команды **Оказание услуги:** создать



В результате наших действий в дереве объектов конфигурации документ **ОказаниеУслуги** будет выглядеть следующим образом:



1. Запустите **1С:Предприятие** в режиме отладки и протестируйте работу созданного документа.



Обратите внимание, что склад **Основной** подставляется по умолчанию, а для полей **Мастер** и **Клиент** выполняется проверка заполнения. А также при вводе цены и количества в табличную часть документа **Оказание услуги** сумма пересчитывается по нашему алгоритму.