

Лекция № 4 Документы. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Документ

Объект конфигурации **Документ** предназначен для описания информации о совершенных хозяйственных операциях или о событиях, произошедших в жизни организации вообще. Как правило, в работе любой фирмы используются такие документы, как приходные накладные, приказы о приеме на работу, платежные поручения, счета и т.д. Свойства и структура этих документов описываются в объектах конфигурации **Документ**, на основе которых платформа создает в базе данных таблицы для хранения информации из этих документов.

Логика работы документов отличается от логики работы других объектов конфигурации. **Документ** обладает способностью проведения. Факт проведения документа означает, что событие, которое он отражает, повлияло на состояние учета.

До тех пор, пока документ не проведен, состояние учета неизменно, и документ - не более чем черновик, заготовка. Как только документ будет проведен, изменения, вносимые документом в учет, вступают в силу и состояние учета будет изменено.

Поскольку документ вносит изменения в состояние учета, он всегда «привязан» к конкретному моменту времени. Это позволяет отражать в базе данных **фактическую последовательность событий**.

Система «1С:Предприятие» имеет механизмы, позволяющие отслеживать правильность состояния учета. Предположим, что мы изменили один из проведенных ранее документов и снова провели его задним числом. В этом случае система «1С:Предприятие» способна отследить, повлияют ли внесенные нами изменения на последующие проведенные документы, и, если это так, система способна перепровести необходимые документы.

В процессе работы пользователь может самостоятельно создавать новые документы — приходные и расходные накладные, счета и т. п.

В базе данных каждый документ представляет собой отдельную запись в основной таблице, хранящей информацию об этом виде документов.

Конфигуратор

Конфигурация

Действия

Поиск (Ctrl+Alt+M)

- Конфигурация
- Общие
- Константы
- Справочники
- Документы
 - Нумераторы
 - Последовательности
 - ПриходнаяНакладная**
- Журналы документов
- Перечисления
- Отчеты
- Обработки
- Планы видов характеристик
- Планы счетов
- Планы видов расчета
- Регистры сведений
- Регистры накопления
- Регистры бухгалтерии
- Регистры расчета
- Бизнес-процессы
- Задачи
- Внешние источники данных

1С:Предприятие

Приходная накладная

Создать

Поиск (Ctrl+F)

Дата	Номер
03.10.2022 11:20:00	000000001
05.10.2022 15:50:00	000000002
10.10.2022 13:37:34	000000003

База данных

Дата	Номер
03.10.2022 11:20:00	000000001
05.10.2022 15:50:00	000000002
10.10.2022 13:37:34	000000003

Стандартные реквизиты документа «**Приходная накладная**» в режиме «**Конфигуратор**», в режиме «**1С:Предприятие**» и в базе данных/

Каждый документ содержит информацию, которая подробнее описывает этот документ. Например, каждый документ **Приходная накладная** может содержать информацию о поставщике товаров, складе, на который приходится товар, и т. д.

Набор такой информации является одинаковым для всех документов одного вида, и для описания такого набора используются реквизиты объекта конфигурации **Документ**, являющиеся подчиненными объектами конфигурации. Большинство реквизитов разработчик создает самостоятельно, однако у каждого объекта **Документ** по умолчанию существуют **стандартные реквизи-**

ты. Два наиболее важных из них - это **Дата** и **Номер**. Поскольку тип данных **Дата** содержит дату и время с точностью до секунды, этот реквизит и определяет в основном положение документа на оси времени.

Конфигуратор

1С:Предприятие

База данных

Дата	Номер	Поставщик
03.10.2022 11:20:00	000000001	ЗАО "Стандарт"
05.10.2022 15:50:00	000000002	ООО "Автоматика"
10.10.2022 13:37:34	000000003	ООО "Атлант"

Реквизит «**Поставщик**» документа «**Приходная накладная**» в режиме «**Конфигуратор**», в режиме «**1С:Предприятие**» и в базе данных

Кроме этого, каждый документ содержит, как правило, некоторый набор информации, которая одинакова по своей структуре, но различна по количеству и предназначена для разных документов. Так, например, каждый документ **Приходная накладная** может содержать список приходуемых товаров.

Для описания конкретной информации служат **табличные части** объекта Документ. В этом случае в базе данных будут созданы дополнительные таблицы для хранения табличных частей, подчиненных конкретному документу.

Конфигуратор

1С:Предприятие

База данных

Основная таблица

Ссылка	Дата	Номер
Ref1	03.10.2022 11:20:00	000000001
Ref2	05.10.2022 15:50:00	000000002
Ref3	10.10.2022 13:37:34	000000003

Табличная часть СписокТоваров

Ссылка	Номер строки	Товар	Количество
Ref1	1	Чайник	10
Ref1	2	Вентилятор	10
Ref2	1	Фен	50
Ref3	1	Пылесос	10
Ref3	2	Чайник	20

Табличная часть «**Список товаров**» документа «**Приходная накладная**» в режиме «**Конфигуратор**», в режиме «**1С:Предприятие**» и в базе данных

Формы документа

Для «визуализации» документа существует несколько основных форм, которые имеют несколько вариантов названий (табл. 4.1).

В контекстном меню и в палитре свойств	В конструкторе форм	На закладке формы
Форма объекта	Форма документа	Документа
Форма списка	Форма списка документа	Списка
Форма для выбора	Форма выбора документа	Выбора

Таблица 4.1. Основные формы документа.

Типы данных. Типообразующие объекты конфигурации

На предыдущем занятии, 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**. *Примитивные типы данных изначально определены в системе, и их набор ограничен.*

Наряду с такими изначально определенными в любой конфигурации типами могут существовать **типы данных, определяемые только конкретной конфигурацией**. То есть такие типы, которые не присутствуют в конфигурации постоянно, а появляются в результате того, что добавлены некоторые объекты конфигурации.

Например, после того как мы создали объект конфигурации **Справочник Склады**, сразу же появилось несколько новых типов данных, связанных с этим справочником. Среди них, например, **СправочникСсылка.Склады**. И если теперь мы укажем какому-либо реквизиту этот тип данных, то сможем хранить в нем ссылку на конкретный объект справочника **Склады**.

Объекты конфигурации, которые могут образовывать новые типы данных, называются **типообразующими**.

Например, после создания нового справочника Номенклатура становятся доступны следующие типы данных:

- СправочникМенеджер.Номенклатура,
- СправочникСсылка.Номенклатура,
- СправочникОбъект.Номенклатура,
- СправочникВыборка.Номенклатура.

Следует отметить, эти типы данных не поддерживаются платформой изначально и существуют только в конкретном прикладном решении.

Это небольшое отступление было необходимо потому, что уже при создании первого документа мы столкнемся с использованием типов данных **СправочникСсылка.Склады** и **СправочникСсылка.Номенклатура**, которые появились в нашей конфигурации в результате создания объектов конфигурации **Справочник Склады** и **Номенклатура**.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Создание Документа «Приходная накладная»

После того как мы познакомились с объектом конфигурации **Документ**, создадим несколько таких объектов, чтобы иметь возможность фиксировать события, происходящие в нашем предприятии.

Например, популярными услугами являются 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 таких предприятий будут поступление материалов и оказание услуг.

Для отражения событий, происходящих на предприятии, в базе данных мы создадим 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.

• В режиме конфигуратор:

Откроем конфигуратор и добавим новый объект конфигурации Документ. На закладке **Основные** зададим имя документа - **ПриходнаяНакладная**. На основании имени платформа автоматически заполнит его **синоним**. 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 зададим во множественном числе, как **Приходные накладные**.

Список приходных накладных должен быть доступен в разделах **Учет материалов** и **Бухгалтерия**. Поэтому на закладке **Подсистемы** отметим в списке подсистем эти подсистемы.

Перейдем на закладку **Данные** и создадим реквизит документа с именем **Склад** (нажать на кнопку **Добавить** («+») над списком реквизитов документа). Выберем для реквизита ссылочный тип данных **СправочникСсылка.Склады**. Этот тип стал доступен в конфигурации после создания объекта конфигурации **Справочник Склады**.

Чтобы при создании нового документа реквизит **Склад** автоматически заполнялся значением склад **Основной**, мы в качестве значения свойства **Значение заполнения** выберем predetermined element of the dictionary **Склады - Основной**.

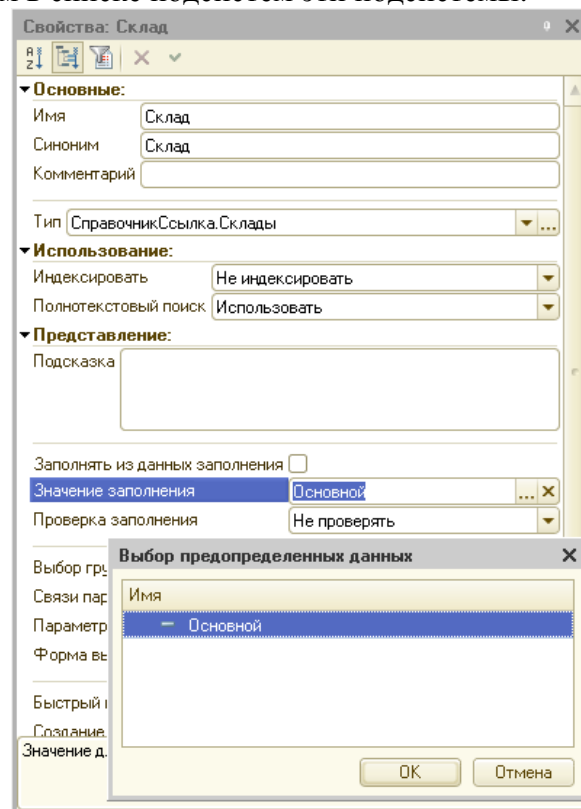


Рис. 4.1. Выбор значения заполнения по умолчанию для реквизита «Склад»

После этого добавим в документ табличную часть с именем **Материалы**. Для этого нажмем кнопку **Добавить** табличную часть над списком табличных частей документа. Кроме имени табличной части установим 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.

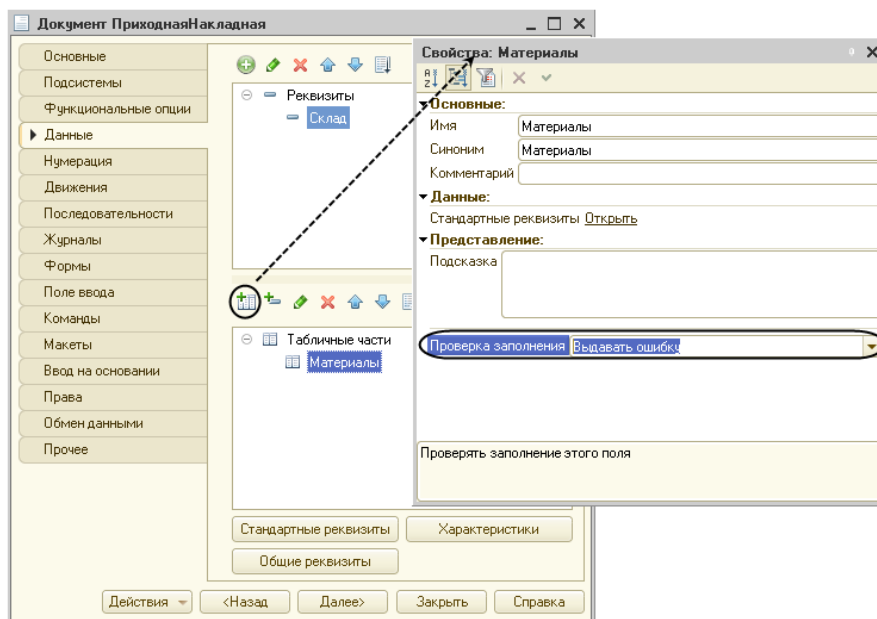


Рис. 4.2 Создание новой табличной части документа

Создадим реквизиты табличной части **Материалы**. Для этого нажмем кнопку **Добавить реквизит** в разделе описания табличных частей документа:

- **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 значении **Выдавать ошибку**.

Тем самым при записи документа будет проверяться на заполнение не только табличная часть в целом, но и ее отдельные реквизиты.

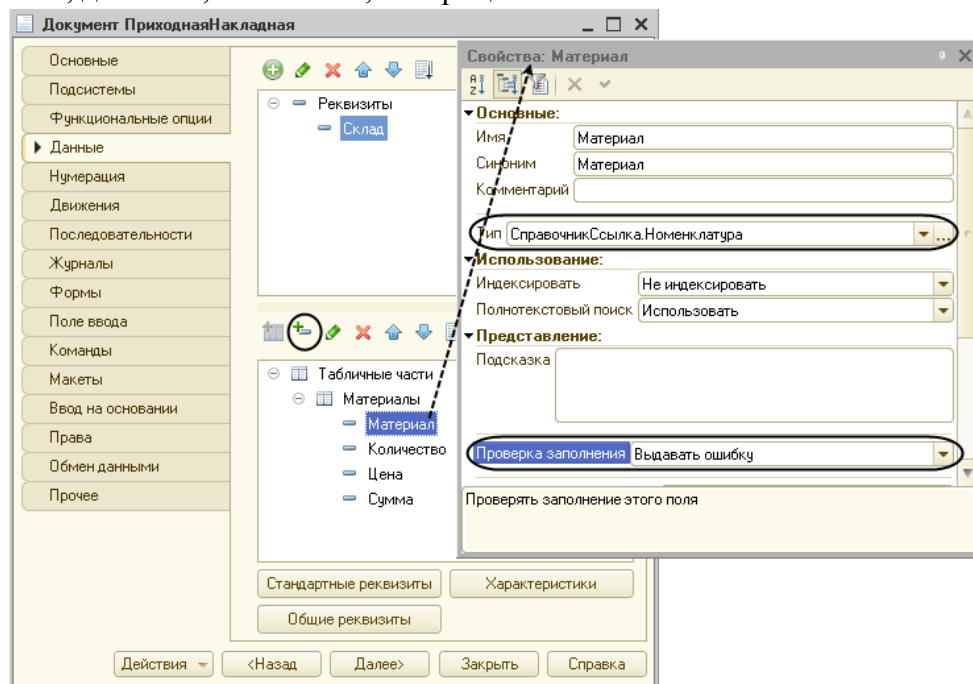


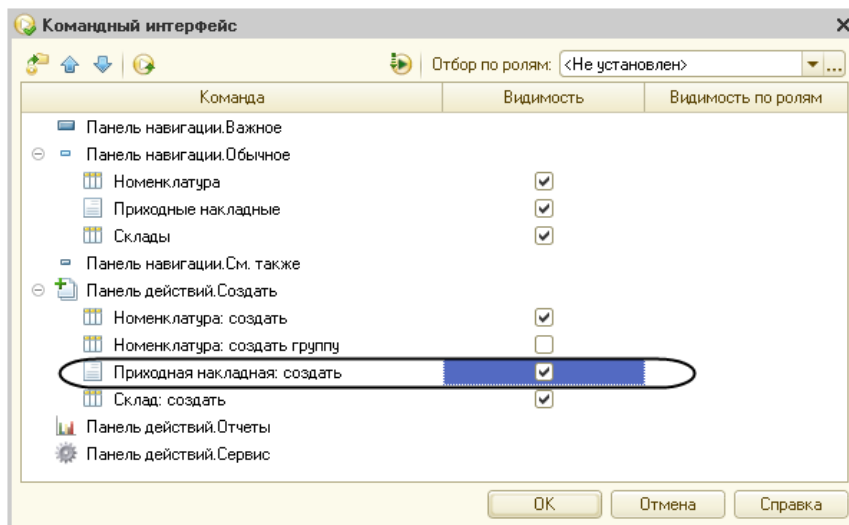
Рис. 4.2a. Создание реквизитов табличной части документа

На закладке **Нумерация**: убедимся, что свойство **Автонумерация** включено. Это обеспечит автоматическую генерацию уникальных номеров для создаваемых нами документов.

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

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

В группе **Панель действий.Создать** включим видимость у команды **Приходная накладная: создать**



Окно настройки командного интерфейса подсистемы

• В режиме 1С:Предприятие

Запустим **1С:Предприятие** в режиме отладки и протестируем получившийся документ. В открывшемся окне **1С:Предприятия** мы видим, что в панели навигации разделов **Бухгалтерия и**

Учет материалов появилась команда **Приходные накладные** для открытия списка приходных накладных (рис. 4.3). Название этой команды определяется свойством **Представление списка**, которое мы задали для этого документа.

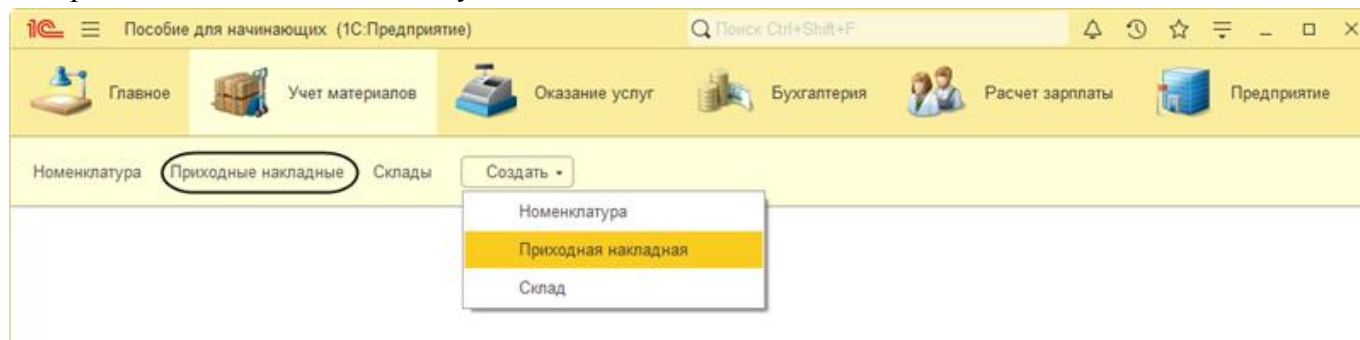


Рис. 4.3. Раздел «Учет материалов»

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

Пока в нашей базе нет ни одного документа **Приходная накладная**, поэтому выполним команду **Приходная накладная** из подменю **Создать** в разделе **Учет материалов** и создадим новую приходную накладную.

Откроется форма документа. Заголовок этой формы **Приходная накладная** совпадает с синонимом документа.

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.

Поле **Номер** не заполнено, но система сама сгенерирует для нового документа уникальный номер, так как свойство **Автоумерация** для документа включено по умолчанию. Новый номер будет сохранен в момент записи документа.

Обратите внимание, что склад уже заполнен значением **Основной**, как мы и задали в свойствах этого реквизита.

Нам осталось только заполнить табличную часть приходной накладной материалами (добавьте 3 разных записи в накладную).

N	Материал	Количество	Цена	Сумма
1	Строчный трансформатор GoldStar	10,000	270,00	2 700,00
2	Строчный трансформатор Samsung	10,000	600,00	6 000,00
3	Транзистор Philips 2N2369	10,000	3,00	30,00

Создание нового документа «Приходная накладная»

Как только мы начнем вводить название материала в поле **Материал**, 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.

Нажмем **Провести и закрыть**.

Документ будет сохранен и проведен, ему будет присвоен автоматически сгенерированный системой номер и текущее время проведения документа.

Обратите внимание, что при вводе нового документа табличная часть в целом и каждая ее колонка подсвечены красным пунктиром. Это значит, что для них выполняется проверка заполнения. Если не ввести ни одной строки в табличную часть документа или оставить незаполненной

какую-либо колонку табличной части и попытаться провести документ, то будет получено сообщение об ошибке.

The screenshot shows the 'Приходная накладная (создание)' window. At the top, there are buttons: 'Провести и закрыть', 'Записать', and 'Провести'. Below these are input fields for 'Номер:', 'Дата: 12.10.2022 0:00:00', and 'Склад: Основной'. A table with columns 'N', 'Материал', 'Количество', 'Цена', and 'Сумма' is visible. The first row contains '1', 'Кабель электрический', and '0.000'. An error message box is displayed over the table, stating: 'Ошибка: Не заполнена колонка "Количество" в строке 1 списка "Материалы"'. Below the table, a 'Сообщения:' section lists three errors: 'Не заполнена колонка "Количество" в строке 1 списка "Материалы"', 'Не заполнена колонка "Цена" в строке 1 списка "Материалы"', and 'Не заполнена колонка "Сумма" в строке 1 списка "Материалы"'. A search bar with 'Поиск (Ctrl+F)' is also present.

Сообщение об ошибке при вводе нового элемента документа

Чтобы просмотреть список созданных документов, выполним команду **Приходные накладные**.

The screenshot shows the 'Приходные накладные' window. At the top, there is a navigation bar with icons for 'Главное', 'Учет материалов', 'Оказание услуг', 'Бухгалтерия', 'Расчет зарплаты', and 'Предприятие'. Below this is a sub-navigation bar with 'Номенклатура', 'Приходные накладные', and 'Склады', along with a 'Создать' button. The main area displays a table with columns 'Дата', 'Номер', and 'Склад'. Two rows are listed: '12.10.2022 15:31:29' with number '000000001' and '12.10.2022 20:09:12' with number '000000002', both from 'Основной' склад. The second row is highlighted. A search bar with 'Поиск (Ctrl+F)' is at the top right.


Рис. 4.4. Список приходных накладных

Автоматический пересчет суммы в строках документа

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.

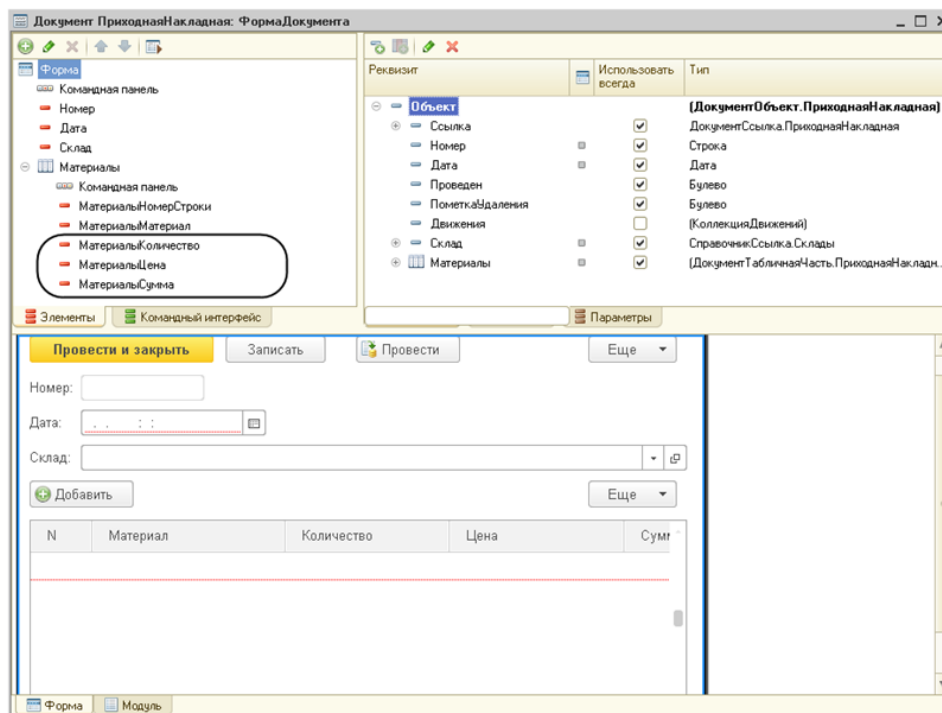
Дело в том, что до сих пор мы использовали predefined формы объектов, которые система «1С:Предприятие» по умолчанию создавала для нас сама. Теперь же у нас возникла необходимость слегка изменить логику работы формы документа, поэтому нам нужно создать свою собственную форму документа **ПриходнаяНакладная** 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.

• В режиме конфигуратор:

Откроем окно редактирования объекта конфигурации **Документ ПриходнаяНакладная**. Далее откроем закладку **Формы**. Как мы видим, ни одна из основных форм документа пока не задана. Для того чтобы создать форму документа, нажмем кнопку открытия  со значком лупы в поле ввода или кнопку **Добавить** над списком форм.

Система вызовет полезный инструмент разработчика - **конструктор формы**, в нем выберем тип формы **Форма документа** и нажмем кнопку **Готово**.

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.



Окно 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.

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.

Но пока нам ничего этого не нужно делать. Нас интересуют три элемента табличной части:

МатериалыКоличество, МатериалыЦена и МатериалыСумма.

Мы хотим, 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**. Но как «поймать» эти моменты изменения?

Обработчик события

Как мы уже видели, 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. Реальные задачи гораздо разнообразнее. Поэтому у системы существуют **события**, которые связаны с самыми различными моментами ее «стандартного» функционирования. В том числе события, связанные с функционированием форм и элементов, расположенных в этих формах.

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.

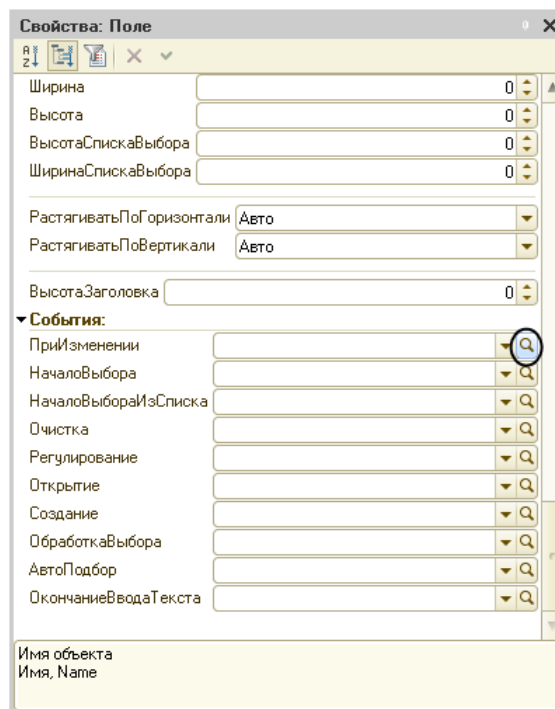
Дважды щелкнем на элементе формы **МатериалыКоличество** или правой кнопкой мыши откроем для него палитру свойств (пункт контекстного меню **Свойства**).

Прокрутив список до конца, мы увидим перечень событий, которые могут быть связаны с этим полем.

Очевидно, что нам нужно событие **ПриИзменении**, которое возникает после изменения значения поля. Найдем его в списке событий и нажмем кнопку открытия со значком лупы в поле ввода.

Система предложит окно для выбора типа обработчика события, который мы хотим создать.

Оставим без изменения вариант **Создать на клиенте**, предложенный по умолчанию.



Создание обработчика события «ПриИзменении» поля табличной части «Количество»

После этого система создаст шаблон процедуры **обработчика** этого события в модуле нашей формы и откроет закладку **Модуль редактора формы** (рис. 4.5).

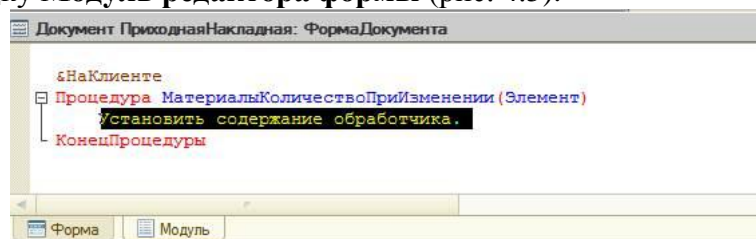


Рис. 4.5. Шаблон обработчика события «ПриИзменении» поля табличной части «Количество»

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**.

В модуль формы, в процедуру **МатериалыКоличествоПриИзменении()**, добавим следующий текст:

```

&НаКлиенте
Процедура МатериалыКоличествоПриИзменении (Элемент)
    // Вставить содержимое обработчика.
    СтрокаТабличнойЧасти = Элементы.Материалы.ТекущиеДанные;
    СтрокаТабличнойЧасти.Сумма = СтрокаТабличнойЧасти.Количество * СтрокаТабличнойЧасти.Цена;
КонечПроцедуры

```

В первой строке мы сначала создаем переменную **СтрокаТабличнойЧасти**, в которую будут помещены данные текущей строки табличной части. В этой строке нам надо пересчитать **Сумму**.

Во встроенном языке 1С можно не объявлять переменную и ее тип заранее. Мы создаем переменную прямо по ходу работы, и ее тип определяется типом значения, которое она содержит.

Поскольку мы находимся в модуле формы, то в нем доступны все свойства и методы объекта встроенного языка **УправляемаяФорма**. Поэтому мы можем обращаться к коллекции элементов формы, используя свойство **Элементы**, которое является одним из свойств объекта **УправляемаяФорма**.

Каждый элемент формы можно получить, указав его имя в качестве свойства этого объекта,

то есть через точку от него. В данном случае мы обращаемся к табличной части документа **Материалы (Элементы.Материалы)**.

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

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

И 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**.

В режиме «1С:Предприятие»

Теперь посмотрим, как это работает. Запустим «1С:Предприятие» в режиме отладки, откроем список документов **Приходные накладные** и откроем любой из двух созданных нами документов. Если теперь вы поменяете количество в любой строке документа, то сумма в строке будет пересчитана автоматически.

Одна процедура для обработки нескольких событий

Итак, мы убедились, что при изменении количества в любой строке документа **Приходная накладная** сумма в этой строке пересчитывается автоматически.

Замечательно. Но 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 перенесем в него нашу процедуру расчета суммы. А в документе просто оставим вызовы этой процедуры из общего модуля.

• В режиме конфигуратор:

Добавим объект конфигурации **Общий модуль**. Для этого раскроем ветвь **Общие** в дереве объектов конфигурации, нажав на + слева от нее. Затем выделим ветвь **Общие модули** и нажмем кнопку **Добавить** в командной панели окна конфигурации.

Назовем его **РаботаСДокументами** и установим в его свойствах флажок **Клиент** (управляемое приложение), а флажок **Сервер** снимем. Это означает, что экземпляры этого модуля будут скомпилированы в контексте тонкого клиента и в контексте веб-клиента

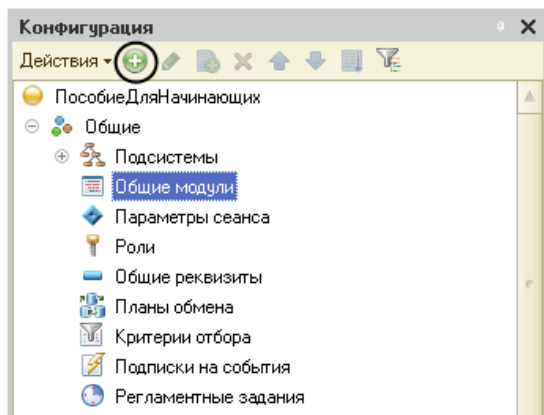


Рис. 4.5. Создание общего модуля в дереве объектов конфигурации

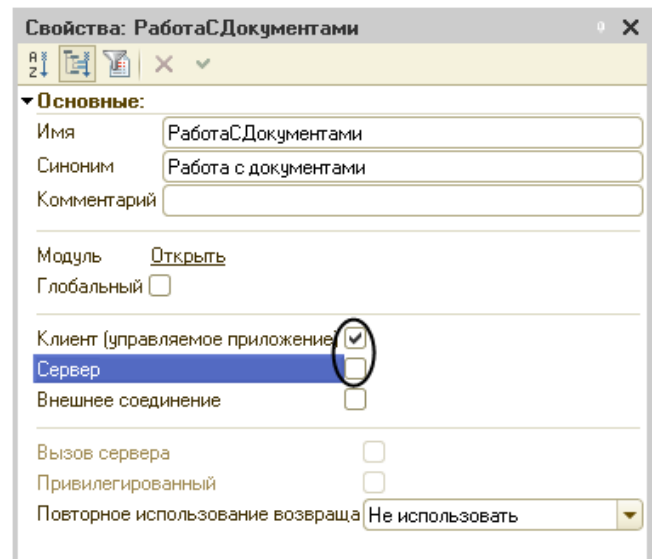
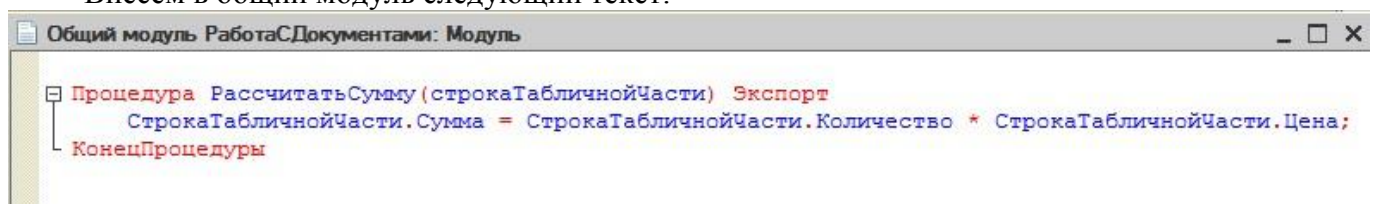


Рис. 4.6. Свойства общего модуля

Внесем в общий модуль следующий текст:



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.

Теперь надо открыть модуль формы нашего документа, чтобы в нем изменить текст процедуры **МатериалыКоличествоПриИзменении**. Форму документа можно открыть двойным щелчком мыши в дереве конфигурации и переходим на закладку **Модуль**, которая находится в нижней части формы документа.

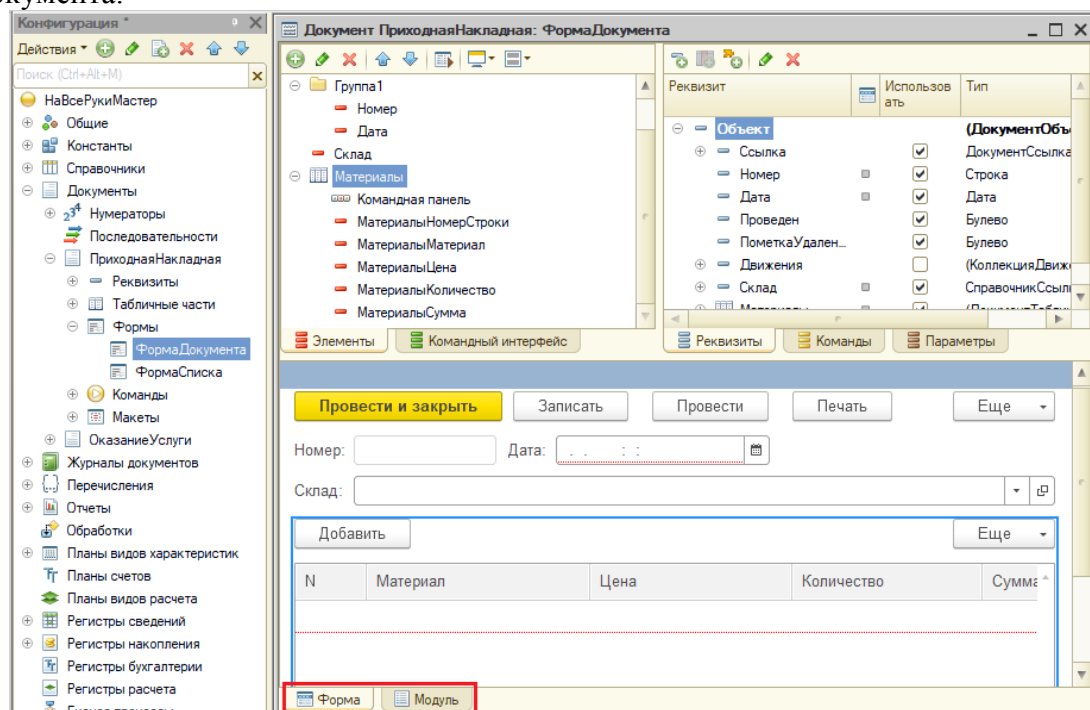


Рис. Переход между закладками **Форма** и **Модуль** формы документа.

В модуле формы документа изменим текст обработчика **МатериалыКоличествоПриИзменении**.

```

        &НаКлиенте
    □ Процедура МатериалыКоличествоПриИзменении (Элемент)
        // Вставить содержимое обработчика.
        СтрокаТабличнойЧасти = Элементы.Материалы.ТекущиеДанные;
        РаботаСДокументами.РассчитатьСумму (СтрокаТабличнойЧасти) ;
    □ КонецПроцедуры

```

Мы видим, что 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.

В режиме **1С:Предприятие** проверьте, как это работает, и убедитесь, что ничего не изменилось.

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

• В режиме 1С: Предприятие:

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

Документ «Оказание услуги»

Теперь мы аналогичным образом создадим второй документ, необходимый нам, — **Оказание услуги**. Для этого потребуется выполнить уже знакомые нам действия, которые мы выполняли по созданию документа **«Приходная накладная»**.

• В режиме конфигуратор:

Добавьте новый объект конфигурации **Документ** и назовите его **ОказаниеУслуги**.

- Представление объекта задавать не будем, вместо него будет использоваться **Синоним объекта**. Представление списка задайте как **Оказание услуг**.

- Сделайте соответствующие настройки, чтобы документ был доступен в подсистемах **Оказание услуг** и **Бухгалтерия**.

- Создайте 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**.

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;

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

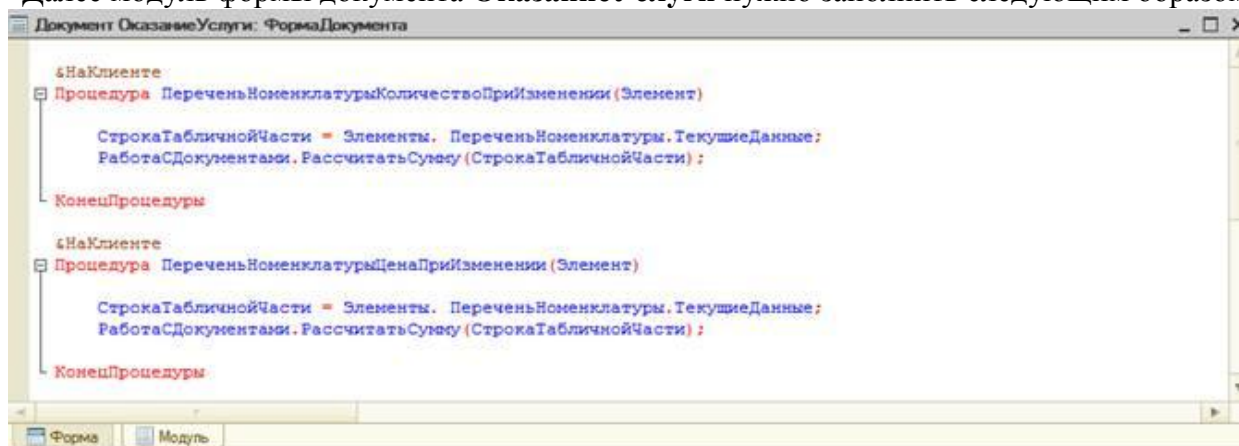
- На закладке **Формы**: создадим основную форму документа:

Для поля **ПереченьНоменклатурыКоличество** создайте обработчик события **ПриИзменении**, в котором можно будет вызывать процедуру **РассчитатьСумму** из общего модуля **РаботаСДокументами**.

При этом откроется модуль формы с шаблоном обработчика события **ПереченьНоменклатурыКоличествоПриИзменении**, который мы пока заполнять не будем, а перейдите в окно элементов формы на закладку **Форма** и аналогичным образом создайте обработ-

чик события **ПереченьНоменклатурыЦенаПриИзменении** для поля **ПереченьНоменклатурыЦена**.

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



Листинг 4.2. Модуль формы документа «ОказаниеУслуги»

В заключение отредактируйте командный интерфейс, чтобы в подсистеме **Оказание услуг** была доступна команда создания новых документов.

В режиме 1С:Предприятие:

В панели действий раздела **Оказание услуг** вызовем команду создания документа **Оказание услуги** и заполним его следующим образом:

N	Номенклатура	Количество	Цена	Сумма
1	Транзистор Philips 2N2369	1,000	3,00	3,00

Рис. 4.8. Создание документа «Оказание услуги»

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.

Так как в модуле формы документа **Оказание услуги** процедуры **ПереченьНоменклатурыКоличествоПриИзменении** и **ПереченьНоменклатурыЦенаПриИзменении** одинаковы, то можно оставить одну из этих процедур (другую удалить) и сделать настройки таким образом, чтобы оставшаяся процедура работала так же, как исходные две процедуры. Т.е. надо оставшуюся процедуру подключить в качестве обработчика события **ПриИзменении** вместо удаленной процедуры. Прodelайте эти действия и проверьте, как работает документ в режиме 1С:Предприятие.

Контрольные вопросы:

- 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 типобразующие объекты.*