

Основные объекты. Документы. Журнал документов

Цель

Ознакомиться с объектом конфигурации «Документы», журналами документов, изучить их свойства и способы создания. Научиться создавать документы и журналы документов.

Время

6 ч.

Порядок выполнения лабораторной работы

1. Изучить теоретический материал.
2. Выполнить задание согласно выданному варианту.
3. Ответить на вопросы для самоконтроля.
4. Подготовить отчет.
5. Защитить лабораторную работу.

Теоретический материал

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, вступят 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.

Каждый документ содержит информацию, которая подробнее описывает этот документ. 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 реквизиты. Два наиболее важных из них – это «Дата» и «Номер». 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.

Формы документа

Для «визуализации» документа существует несколько основных форм, которые имеют несколько вариантов названий (табл. 3.1).

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 3.1. Основные формы документа.

Типы данных. 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. *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.

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

Создание Документа «Приходная накладная»

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. Так же зададим «Представление списка», как «Приходные накладные».

- На закладке «Подсистемы»: список приходных накладных должен быть доступен в разделах «Учет материалов» и «Бухгалтерия».

- На закладке «Данные»: создадим реквизит документа с именем «Склад» (нажать на «+»); далее выберем для реквизита ссылочный тип данных «СправочникСсылка.Склады». Этот тип стал доступен в конфигурации после создания объекта конфигурации «Справочник.Склады». В качестве значения свойства «Значение заполнения» выберем predetermined элемент справочника «Склады» - «Основной».

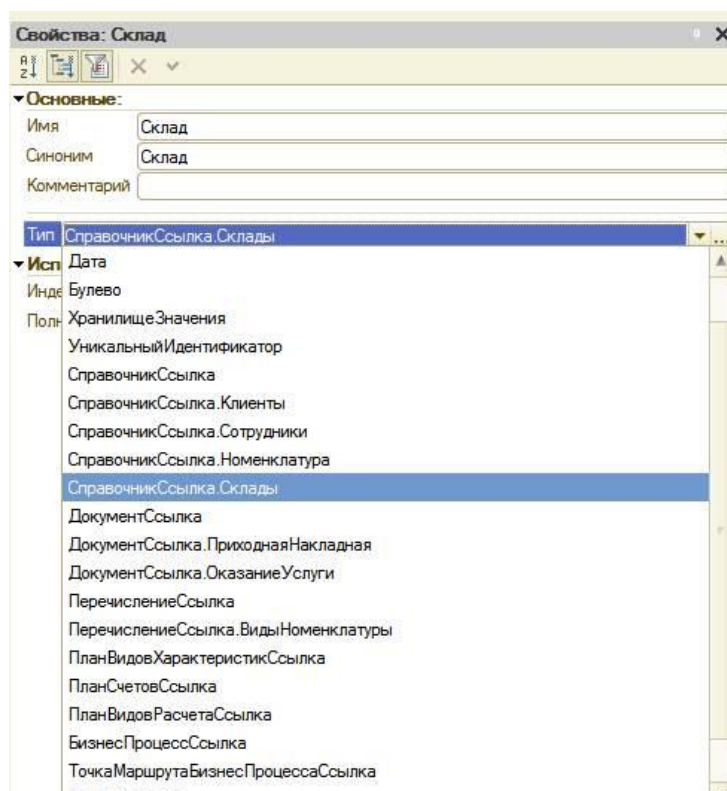


Рис. 3.1. Справочник Склады

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».

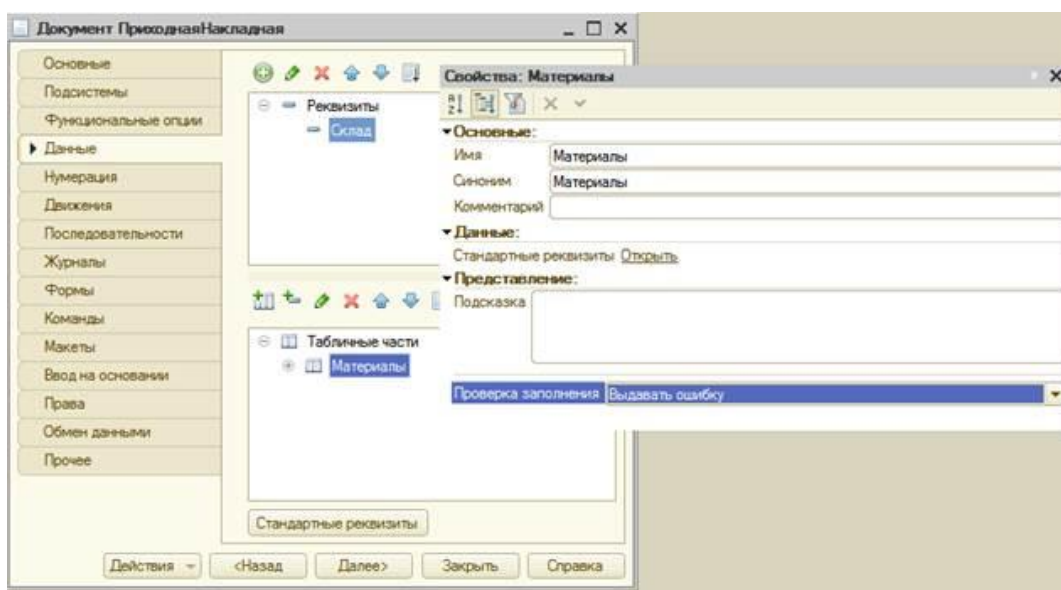


Рис. 3.2. Свойства: Материалы

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 значение «Выдавать ошибку».

- На закладке «Нумерация»: убедимся, что свойство «Авто-нумерация» включено.

- В дереве объектов конфигурации выделим ветвь «Подсистемы», вызовем ее контекстное меню и выберем пункт «Все подсистемы». В открывшемся окне в списке «Подсистемы» выделим подсистему «УчетМатериалов»: включим видимость у команды «Приходная накладная»: создать, в группе Панель действий.Создать.

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.

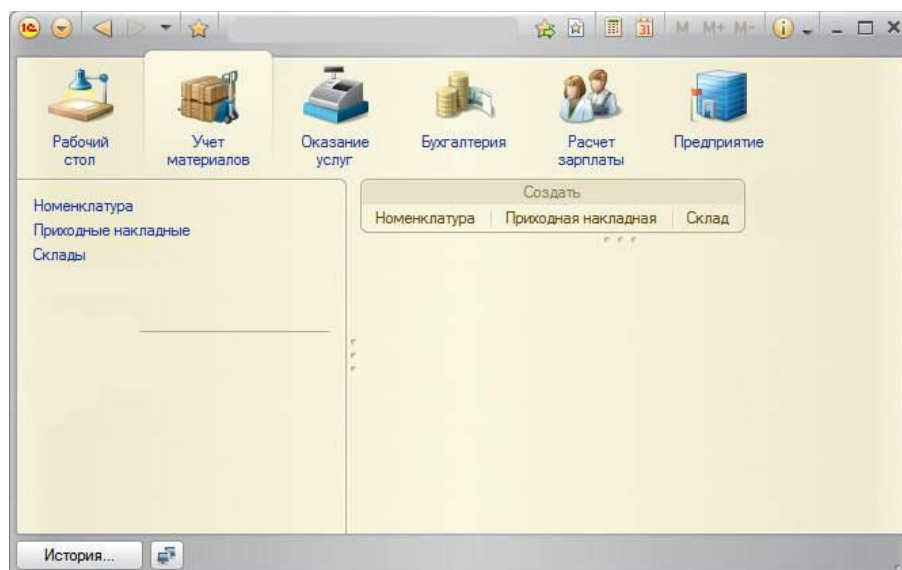


Рис. 3.3. В режиме 1С:Предприятие

Пока в БД нет ни одного документа «**Приходная накладная**», поэтому выполним команду «**Приходная накладная**» в панели действий раздела «**Учет материалов**» и создадим новую приходную накладную. 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 «**Автонумерация**» для документа включено по умолчанию. Новый номер будет сохранен в момент записи документа. *Осталось только заполнить табличную часть приходной накладной материалами (добавьте 3 любых записи в накладную). Так же система может вводить данные с клавиатуры и предлагать вам материалы по совпадающим буквам (создайте еще одну накладную таким образом)* После чего нажмем «Провести и закрыть».

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

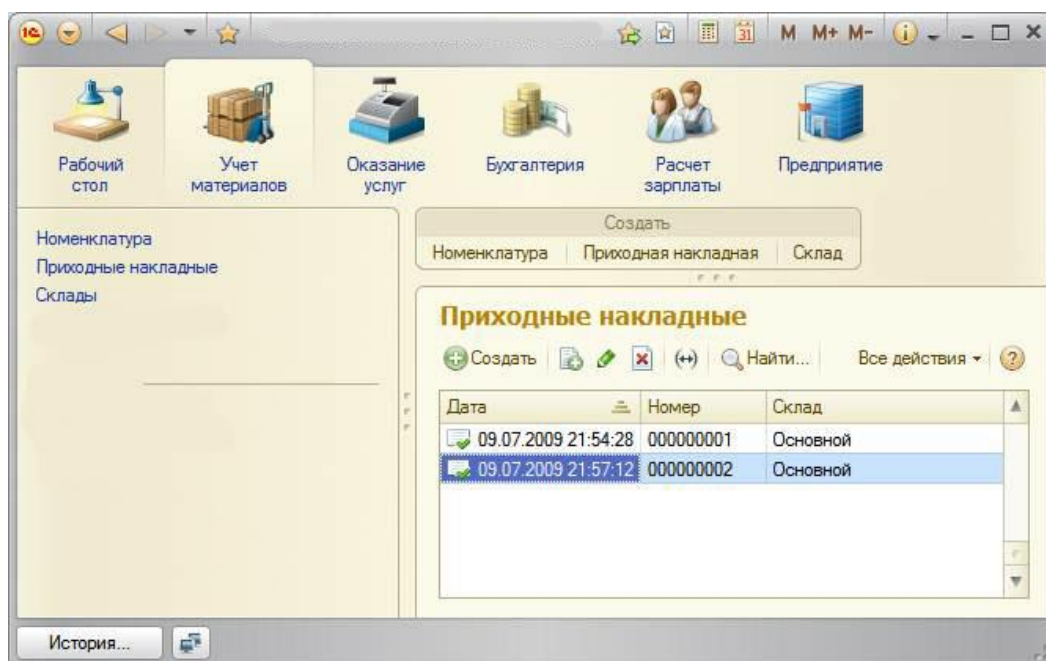


Рис. 3.4. Приходные накладные

Автоматический пересчет суммы в строках документа

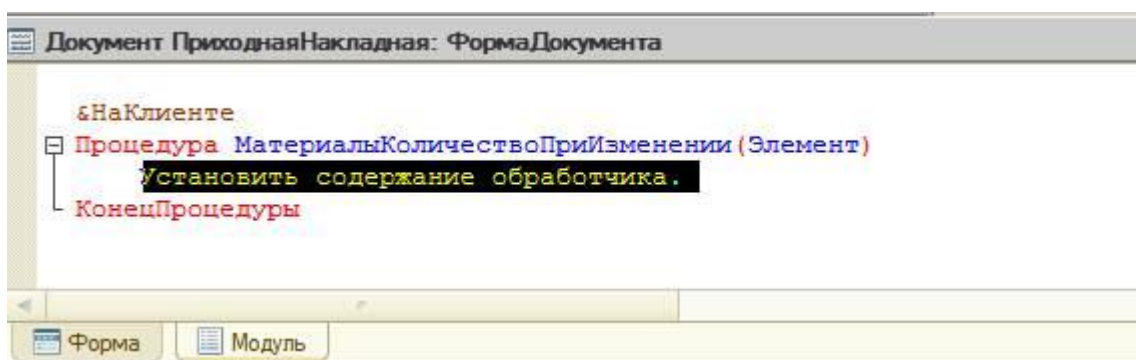
При заполнении документа приходится вводить сумму в каждой строке. Это неудобно, и возникает естественное желание автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при

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.

В режиме Конфигуратор:

Откроем окно редактирования объекта конфигурации «Документ» «ПриходнаяНакладная», далее пройдем на вкладку «Формы»: 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 форм», в нем выберем тип формы «Форма документа» и нажмем кнопку «Готово». *Обращаем внимание, что в дереве объектов конфигурации у объекта конфигурации «Документ» «ПриходнаяНакладная» появилась форма «ФормаДокумента», а на экране открылось окно редактора форм, содержащее эту форму.* 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. Используя встроенный язык, разработчик может «вклиниваться» в эти события и описать собственный алгоритм того, что должно происходить при наступлении этого события.

- Дважды щелкнем на элементе формы «Материалы» «Количество» или правой кнопкой мыши откроем для него палитру свойств: прокрутив список до конца, мы увидим перечень событий, которые могут быть связаны с этим полем. Найдем в списке событий событие «ПриИзменении», и нажмем на кнопку открытия (значок лупы), система создаст шаблон процедуры обработчика этого события в модуле нашей формы и откроет закладку «Модуль» редактора формы.



Листинг 3.1. Форма документа

Модуль – это «хранилище» для текста программы на встроенном языке. В модуль формы, в процедуру МатериалыКоличествоПриИзменении(), добавим следующий текст:

СтрокаТабличнойЧасти = Элементы.Материалы.ТекущиеДанные;

СтрокаТабличнойЧасти.Сумма =

*СтрокаТабличнойЧасти.Количество*СтрокаТабличнойЧасти.Цена;*

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.

Одна процедура для обработки нескольких событий

Теперь хотелось бы и для поля «Цена» и «Сумма» сделать то же самое. 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.

- **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:

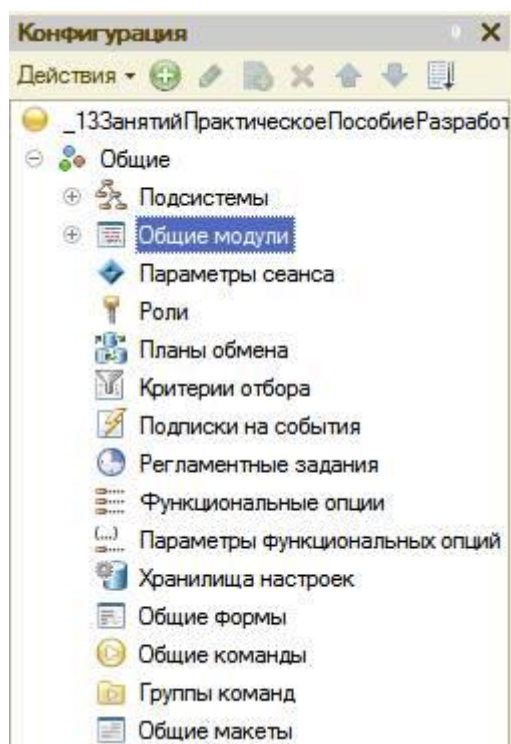


Рис. 3.5. Общие модули

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

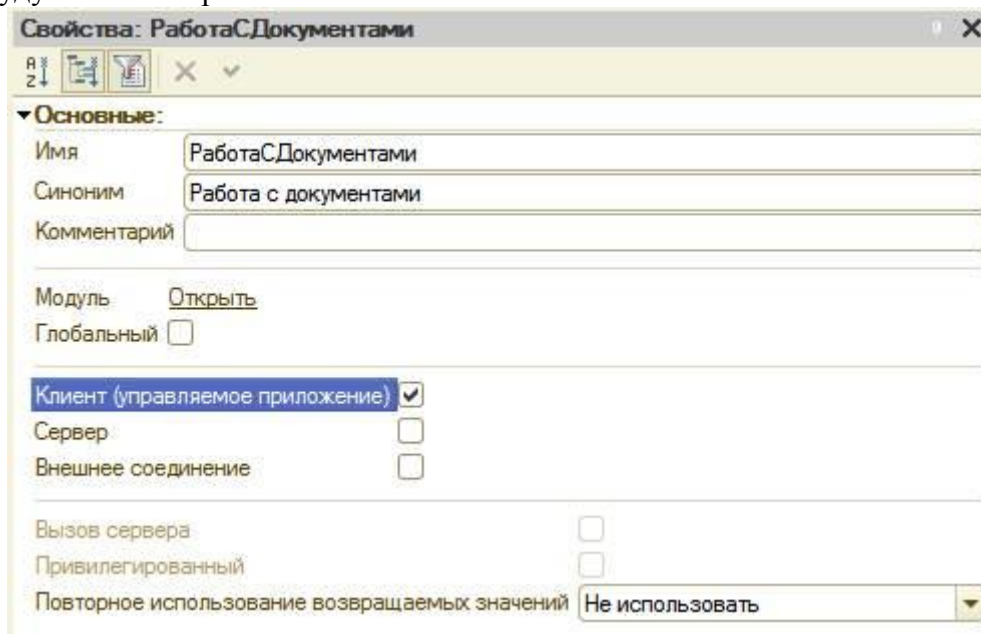


Рис. 3.6. Свойства: РаботаСДокументами

Внесем в общий модуль следующий текст:

**Процедура РассчитатьСумму(строкаТабличнойЧастии) Экспорт
СтрокаТабличнойЧастии.Сумма =
СтрокаТабличнойЧастии.Количество*СтрокаТабличнойЧастии.Цена;
КонецПроцедуры**

Изменим текст обработчика. Для этого вернитесь в модуль формы документа, выберите поле «КоличествоПриИзменении».

**&НаКлиенте
Процедура МатериалыКоличествоПриИзменении (Элемент)
СтрокаТабличнойЧастии. = Элементы.Материалы.ТекущиеДанные;
РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧастии);
КонецПроцедуры**

Создадим обработчик события «ПриИзменении» для поля табличной части «Материалы» «Цена» так же, как мы делали это для поля «Материалы» «Количество», и повторим в нем вызов процедуры «РассчитатьСумму» из общего модуля.

- Листинг процедуры «МатериалыЦенаПриИзменении()»

**&НаКлиенте
Процедура МатериалыЦенаПриИзменении (Элемент)
СтрокаТабличнойЧастии. = Элементы.Материалы.ТекущиеДанные;
РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧастии);
КонецПроцедуры**

- **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»

- **В режиме конфигуратор:**

Добавим новый объект конфигурации «Документ» и назовем его «ОказаниеУслуги».

- На закладке «Основные»: 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».

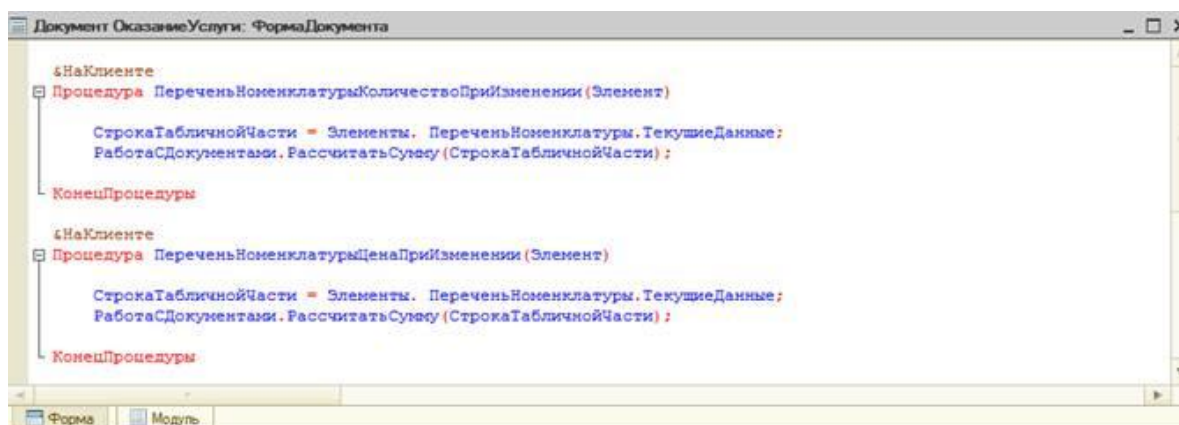
Создадим табличную часть этого документа «ПереченьНоменклатуры» с реквизитами:

- 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», 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 следующим образом:



Листинг 3.2. Модуль формы документа «ОказаниеУслуги»

В заключение отредактируем командный интерфейс, чтобы в подсистеме «Оказание услуг» была доступна команда создания новых документов. Для разнообразия воспользуемся другим способом. Откроем окно редактирования объекта конфигурации «Подсистема» «ОказаниеУслуг» и нажмем кнопку «Командный интерфейс». В открывшемся окне отразятся все команды выбранной подсистемы. В группе «Панель действий.Создать» включим видимость у команды «Оказание услуги: создать».

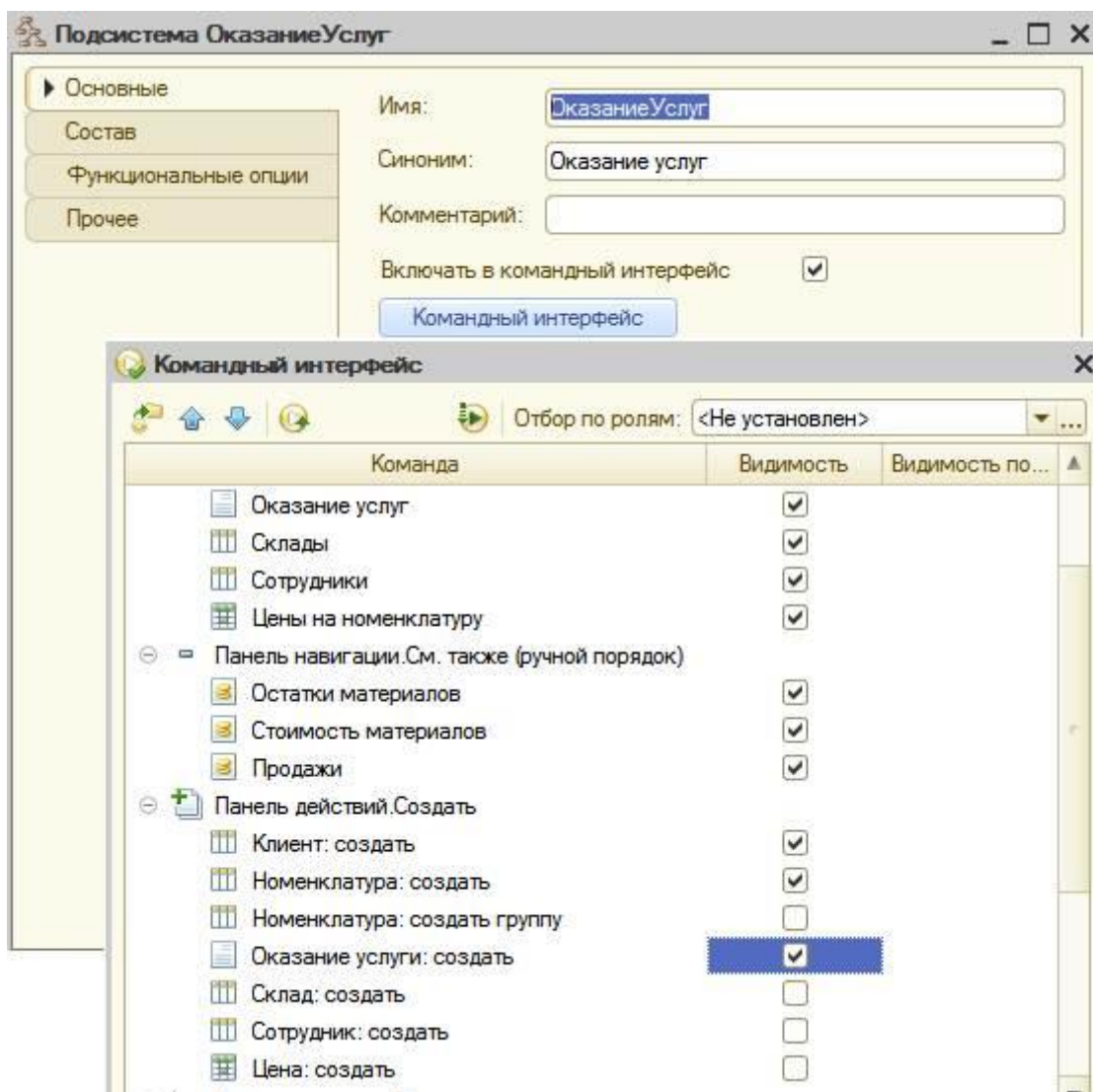


Рис. 4.7. Видимость у команды Оказание услуги: создать

В режиме 1С:Предприятие:

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

Оказание услуги 0000000001 от 13.07.2009 0:00:00

Провести и закрыть | Провести | Печать | Все действия ▾ ?

Номер: 0000000001

Дата: 13.07.2009 0:00:00

Склад: Основной

Клиент: Иванов Михаил Юрьевич

Мастер: Деловой Иван Сергеевич

+ Добавить | Все действия ▾

N	Номенклатура	Стоимость	Количество	Цена	Су
1	Подключение воды		1,000	800,00	
2	Шланг резиновый	100,00	1,000	150,00	
					Всего:

Рис. 3.8. 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.

Дополнительный теоретический материал

Механизм основных форм

На предыдущем занятии мы 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. 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.

Обработчики событий

При работе с событиями в платформе 1С:Предприятие 8 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, совпадающее 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. Назначение обработчика может выполняться интерактивно, при работе с формой в конфигураторе, или программно, используя методы формы и ее элементов – «УстановитьДействие()».

Модули

Виды модулей:

- *Модуль управляемого приложения.* Модуль управляемого приложения выполняется при старте системы 1С:Предприятие в режимах тонкого клиента и веб-клиента.

В нем возможно объявление переменных, а также объявление и описание процедур и функций, которые будут доступны в любом модуле конфигурации (кроме модуля внешнего соединения). Их доступность также обеспечивается для неглобальных общих модулей с установленным свойством Клиент (управляемое приложение). В контексте модуля управляемого приложения доступны экспортируемые процедуры и функции общих модулей. Чтобы открыть модуль управляемого приложения, нужно выделить корень дерева объектов конфигурации и вызвать из контекстного меню команду Открыть модуль управляемого приложения

- *Общие модули.* В общих модулях хранятся процедуры и функции, которые вызываются из других модулей системы. Сам по себе общий модуль не исполняется. Исполняются отдельные его процедуры/функции в момент их вызова из других модулей. Чтобы открыть общий модуль, нужно раскрыть ветвь Общие в дереве объектов конфигурации, затем раскрыть ветвь Общие модули и дважды щелкнуть мышью на нужном модуле.
- *Модули объектов.* Модули объектов – это, например, модуль элемента справочника или модуль документа. Эти модули вызываются тогда, когда либо программно создается этот объект средствами встроенного языка, например, методами СоздатьЭлемент() у менеджеров справочников или СоздатьДокумент() менеджеров документов, либо когда пользователь создает новый элемент справочника или документ интерактивно. Чтобы открыть модуль объекта, нужно в окне редактирования объекта конфигурации перейти на закладку «Прочее» и нажать кнопку «Модуль объекта». Или, выделив нужный объект в дереве объектов конфигурации, вызвать из контекстного меню команду «Открыть модуль объекта».

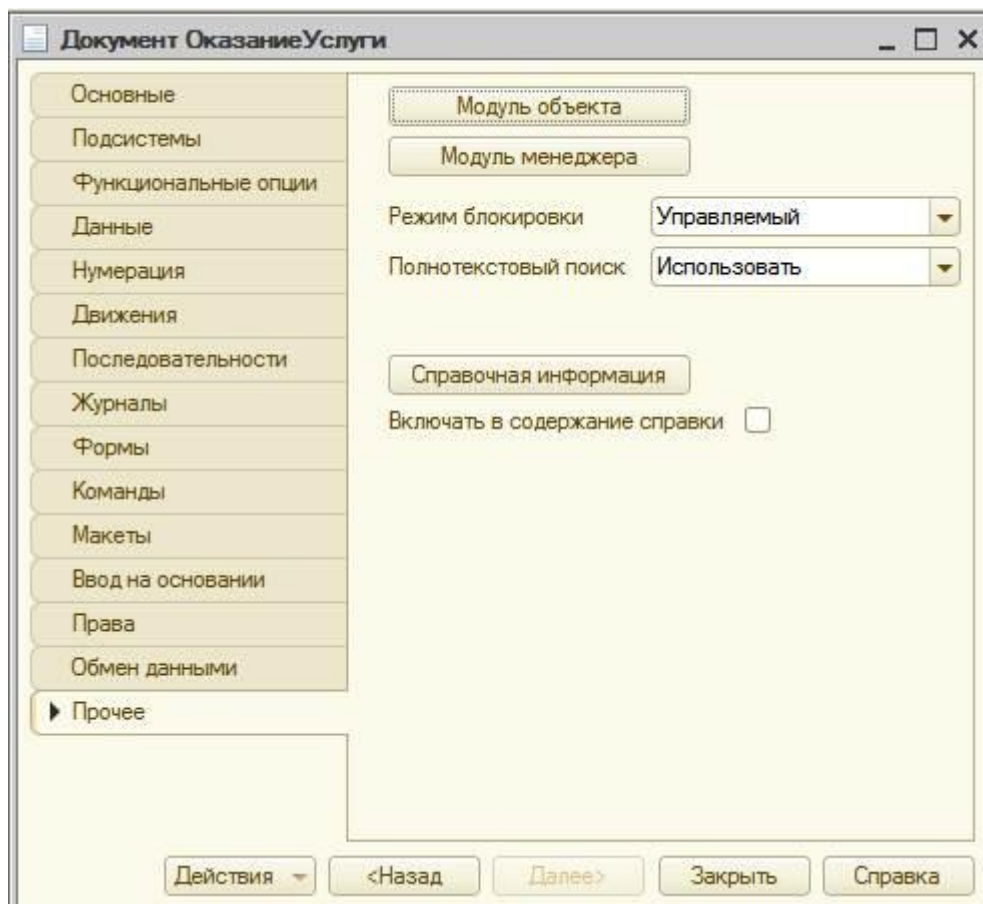
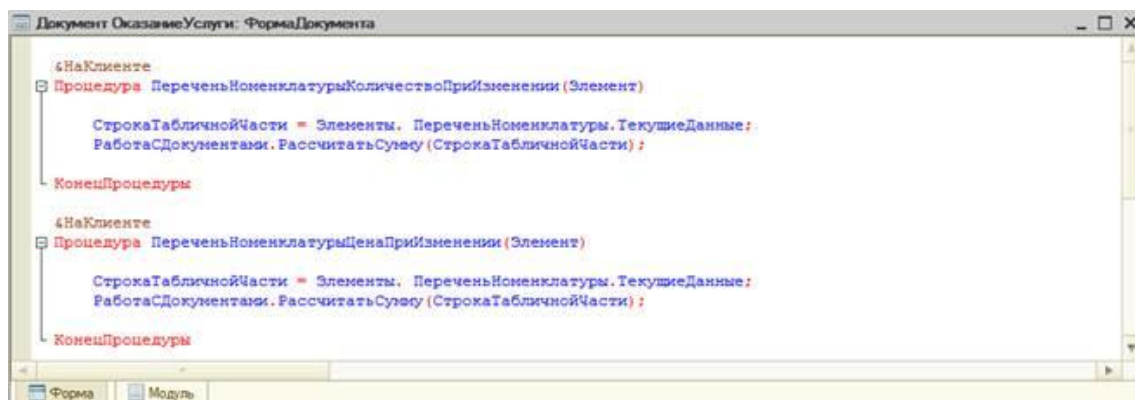


Рис.3.9. Открытие модуля объекта

- *Модули форм.* Каждая форма, определенная в конфигурации, имеет свой собственный модуль. Этот модуль выполняется при создании объекта «УправляемаяФорма» встроенного языка. Этот объект создается системой в режиме 1С предприятие в тот момент, когда мы программно (методами ПолучитьФорму() или ОткрытьФорму()). Чтобы открыть модуль формы, нужно открыть нужный объект конфигурации «Форма объекта» и в окне редактора форм перейти на закладку «Модуль».



Листинг 3.4. Открытие модуля формы

- *Модуль сеанса.* Модулем сеанса называется модуль, который автоматически выполняется при старте системы 1С:Предприятие в момент загрузки конфигурации. Модуль сеанса предназначен для инициализации параметров сеанса и отработки действий, связанных с сеансом работы. Модуль сеанса не содержит экспортируемых процедур и функций и может использовать процедуры из общих модулей конфигурации. Чтобы открыть модуль сеанса, нужно выделить корень дерева объектов конфигурации (строку Фамилия) и вызвать из контекстного меню команду «Открыть модуль сеанса».
- *Модуль внешнего соединения* предназначен для размещения в нем текстов функций и процедур, которые могут вызываться в сессии внешнего соединения. Чтобы открыть модуль сеанса, нужно выделить корень дерева объектов конфигурации (строку ПособиеДляНачинающих) и вызвать из контекстного меню команду «Открыть модуль внешнего соединения».
- *Модуль менеджеров.* Для каждого прикладного объекта существует менеджер, предназначенный для управления этим объектом как объектом конфигурации. С помощью менеджера можно создавать объекты, работать с формами и макетами. Модуль менеджера позволяет расширить функциональность менеджеров, предоставляемых системой, за счет написания процедур и функций на встроенном языке. Чтобы открыть модуль менеджера, нужно в окне редактирования объекта конфигурации перейти на закладку «Прочее» и нажать кнопку «Модуль менеджера».

- *Модуль команды.* Как в самой конфигурации, так и у многих прикладных объектов могут существовать подчиненные объекты конфигурации – «Команды». У каждой команды существует модуль команды, в котором можно написать предопределенную процедуру `ОбработкаКоманды()` для выполнения этой команды. Чтобы открыть модуль команды, подчиненной некоторому объекту конфигурации, нужно в окне редактирования объекта конфигурации перейти на закладку «Команды» и дважды щелкнуть мышью на нужной команде. Или, выделив нужную команду в дереве объектов конфигурации, вызвать из контекстного меню команду «Открыть модуль команды».

Контекст модуля формы

Каждый модуль связан с остальной частью конфигурации. Эта связь называется контекстом модуля. Контекст модуля определяет набор доступных во время выполнения модуля объектов, переменных, процедур и функций. Контекст модуля формы образуется:

- локальным контекстом самого модуля формы;
- реквизитами формы, которой «принадлежит» модуль;
- свойствами и методами объекта `УправляемаяФорма` встроенного языка;
- свойствами и методами расширения формы, определяемого типом того объекта, данные которого содержатся в основном реквизите формы;
- глобальным контекстом, в том числе неглобальными общими модулями и экспортируемыми функциями и процедурами глобальных общих модулей;
- экспортируемыми переменными, процедурами и функциями модуля управляемого приложения;

Форма как программный объект

Помимо того что форма внутри своего модуля предоставляет доступ к различным частям конфигурации, она также доступна из других частей конфигурации как программный объект. При этом помимо стандартных свойств и методов объекта встроенного языка «`УправляемаяФорма`», у нее могут существовать и другие свойства и методы, определенные разработчиком.

```

«НаКлиенте
□ Процедура МатериалыКоличествоПриИзменении (Элемент)
    СтрокаТабличнойЧасти = Элементы.Материалы.ТекущиеДанные;
    РаботаСДокументами.РассчитатьСумму (СтрокаТабличнойЧасти);
КонецПроцедуры

```

Листинг 3.5.

Например, если в модуле формы «ФормаДокумента» документа «ПриходнаяНакладная» описана экспортируемая процедура, то может быть использован следующий вызов этой процедуры:

```

форма=ПолучитФорму ("Документ.ПриходнаяНакладная.форма.формаДокумента");
форма.МатериалыКоличествоПриИзменении ();|

```

Листинг 3.6.

Как понять работу кода на встроенном языке

Теперь мы покажем два способа, как самому разобраться с множеством незнакомых свойств и методов объектов конфигурации, чтобы в будущем самостоятельно изучать фрагменты кода или создавать свои собственные процедуры на встроенном языке.

Синтакс-помощник — инструмент, созданный для помощи разработчику, содержащий описание всех программных объектов, которые использует система, их методов, свойств, событий и пр. Чтобы открыть синтакс-помощник, нужно нажать соответствующую кнопку на панели инструментов конфигуратора или выполнить команду главного меню «Справка» > «Синтакс-помощник», или с помощью горячей клавиши (Ctrl + F1).

Анализ кода с помощью синтакс-помощника:

Пользоваться синтакс-помощником удобно в тех случаях, когда нужно разобраться в уже написанном незнакомом коде.

Первый способ – найти нужный раздел в содержании и спускаться вниз «по дереву», раскрывая нужные подразделы, свойства, ссылки и т.п.

Второй способ – воспользоваться контекстной помощью синтакс-помощника. Для этого нужно открыть программный модуль, установить курсор на интересующую вас конструкцию встроенного языка и нажать Ctrl+F1.

Есть также еще одна полезная возможность использования синтаксис-помощника. Можно ограничить состав объектов, которые будут отображаться в нем. Так как мы находимся на клиенте, в форме, имеет смысл ограничиться только объектами встроенного языка, доступными в режимах Тонкий клиент и Веб-клиент. Для этого нужно выполнить команду главного меню «Сервис» > «Параметры» или нажать кнопку «Открыть режим настройки параметров», находящуюся над окном описания объектов синтаксис-помощника. На закладке «Справка» окна «Параметры» можно снять или поставить отметку у нужных режимов исполнения.

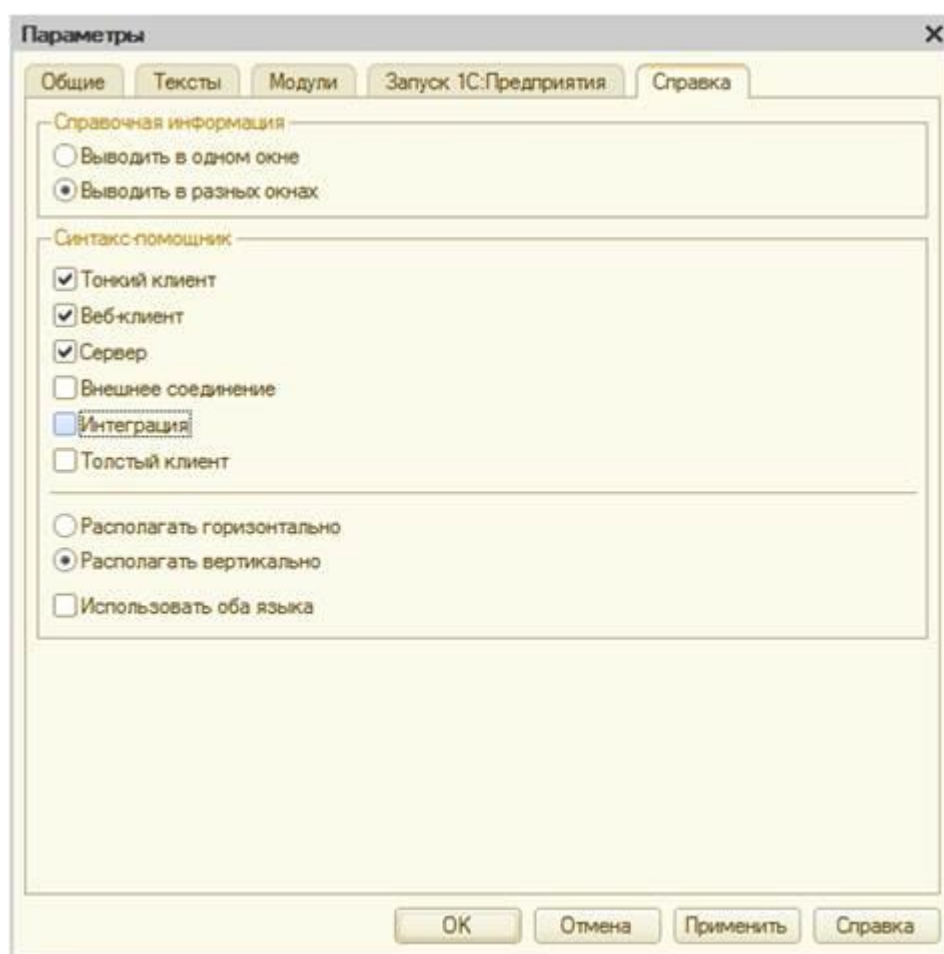


Рис. 3.10. Параметры

Анализ кода с помощью отладчика

Пользоваться отладчиком наиболее удобно в тех случаях, когда нужно написать какой-то собственный код. Можно просто остановиться в конкретном месте программы и посмотреть, какие же свойства здесь доступны или какие программные объекты здесь используются. 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.

Объекты

Как правило, термин объект употребляется в одном из трех контекстов:

- конфигурация,
- база данных,
- встроенный язык.

Говоря о конфигурации, термином объект конфигурации мы обозначаем некоторую совокупность описания данных и алгоритмов работы с этими данными. Например, в конфигурации может существовать объект «Справочник Сотрудники».

Когда мы говорим о базе данных, термином *объект* мы обозначаем всего лишь некий элемент такой информационной структуры. Характерной особенностью такого элемента является то, что на него (как на совокупность данных) существует ссылка, которая может являться значением какого-либо поля другой информационной структуры. Если же мы начинаем говорить о встроенном языке и о том, каким образом средствами встроенного языка работать со справочниками, то термином *объект* мы обозначаем тип данных, позволяющий получить доступ к данным и обладающий набором свойств и методов.

Сервер и клиенты

Система 1С:Предприятие поддерживает два варианта работы системы: файловый и клиент-серверный.

Файловый вариант работы с информационной базой рассчитан на персональную работу одного пользователя или работу небольшого количества пользователей в локальной сети. В этом варианте все данные информационной базы (конфигурация, база данных, административная информация) располагаются в одном файле.



Рис. 3.10. Файловый вариант

Основное назначение файлового варианта – быстро и легко установить систему и работать с ней. Например, чтобы что-то посмотреть или доработать дома или на ноутбуке. В файловом варианте тоже можно вести реальную учетную работу, но при этом нужно понимать, что он не предоставляет абсолютно всех тех же возможностей по масштабируемости, защите данных, какие имеет клиент-серверный вариант.

Клиент-серверный вариант предназначен для использования в рабочих группах или в масштабе предприятия - это основной вариант для работы в многопользовательской среде с большим объемом данных. Он предоставляет абсолютно все возможности по масштабируемости, администрированию и защите данных. Однако он требует значительных усилий по установке и администрированию.

Система 1С:Предприятие изначально рассчитана на клиент-серверный вариант работы. Хотя сейчас вы разрабатываете свою учебную конфигурацию в файловом варианте работы, она будет работать и в клиент-серверном варианте без ваших дополнительных усилий. Прикладные решения разрабатываются один раз и одинаково работают, что в одном, что в другом варианте. То есть переход с одного варианта на другой не требует переделки конфигурации.



Рис. 3.11. Клиент-серверный вариант

Клиент-серверная архитектура разделяет всю работающую систему на три различные части, определенным образом взаимодействующие между собой, - Клиент, Сервер 1С: Предприятия и Сервер баз данных.

Клиентское приложение – это программа, часть системы 1С: Предприятие. Основное ее назначение – организация пользовательского интерфейса, отображение данных с возможностью их изменения. Кроме этого, клиентское приложение может исполнять код на встроенном языке. Такой подход позволяет клиентскому приложению быть очень «легким», не требовать много ресурсов, «путешествовать» по Интернету и работать даже в среде веб-браузеров.

Сервер баз данных – это тоже программа. Она уже не является частью системы 1С:Предприятие, это специализированная программа, поставляемая сторонними производителями. Ее основное назначение – это организация и ведение баз данных – структурированных организованных наборов данных, описывающих характеристики каких-либо физических или виртуальных систем.

В настоящее время система 1С:Предприятие может работать со следующими серверами баз данных:

- *MS SQL Server*;
- *Posteg SQL*;
- *IBM DB2*;
- *Oracle Database*.

Вопросы для самоконтроля

1. Что такое документ в конфигурации 1С?
2. Какими характерными особенностями обладает документ?
3. Для чего предназначены реквизиты и табличные части документа?
4. Какие существуют основные формы документа?
5. Что такое проведение документа?
6. Как создать объект конфигурации «Документ» и описать его основную структуру?
7. Как создать новый документ и заполнить его данными?
8. Как создать собственную форму документа?
9. Что такое конструктор форм?
10. Что такое редактор форм?
11. Что такое элементы формы?
12. Что такое события и с чем они связаны?
13. Что такое обработчик события и как его создать?
14. Что такое модуль и для чего он нужен?
15. Зачем нужны общие модули?
16. Что такое типобразующие объекты?

Требования к содержимому отчета

1. Титульный лист
2. Задание
3. Результаты выполнения программы
4. Выводы

Теоретический материал

1. М.Г. Радченко, Е.Ю. Хрусталева. 1С: Предприятие 8.3. Практическое пособие разработчика. – М.: ООО «1С-Паблишинг», 2014.

Варианты

Вариант 1

В рамках выполнения лабораторных работ по курсу «Основы программирования и конфигурирования в 1С» разработать конфигурацию для ведения учета и оборота компьютерной техники и комплектующих.

Текущий этап: добавить в конфигурацию документы «Приходная накладная» и «Оказание услуг». В документах обязательно должна присутствовать табличная часть. Суммы и итог документа должны подсчитываться при каждом изменении элемента табличной части документа. Номер документа генерируется автоматически.

Продумать и (согласовав предварительно с преподавателем) добавить остальные документы конфигурации.

Вариант 2

В рамках выполнения лабораторных работ по курсу «Основы программирования и конфигурирования в 1С» разработать конфигурацию для ведения учета и оборота печатной продукции.

Текущий этап: добавить в конфигурацию документы «Приходная накладная» и «Оказание услуг». В документах обязательно должна присутствовать табличная часть. Суммы и итог документа должны подсчитываться при каждом изменении элемента табличной части документа. Номер документа генерируется автоматически.

Продумать и (согласовав предварительно с преподавателем) добавить остальные документы конфигурации.

Вариант 3

В рамках выполнения лабораторных работ по курсу «Основы программирования и конфигурирования в 1С» разработать конфигурацию для ведения учета и оборота продуктов питания.

Текущий этап: добавить в конфигурацию документы «Приходная накладная» и «Оказание услуг». В документах обязательно должна присутствовать табличная часть. Суммы и итог документа должны подсчитываться при каждом изменении элемента табличной части документа. Номер документа генерируется автоматически.

Продумать и (согласовав предварительно с преподавателем) добавить остальные документы конфигурации.

Вариант 4

В рамках выполнения лабораторных работ по курсу «Основы программирования и конфигурирования в 1С» разработать конфигурацию для ведения учета и оборота медикаментов в аптеке.

Текущий этап: добавить в конфигурацию документы «Приходная накладная» и «Оказание услуг». В документах обязательно должна

присутствовать табличная часть. Суммы и итог документа должны подсчитываться при каждом изменении элемента табличной части документа. Номер документа генерируется автоматически.

Продумать и (согласовав предварительно с преподавателем) добавить остальные документы конфигурации.

Вариант 5

В рамках выполнения лабораторных работ по курсу «Основы программирования и конфигурирования в 1С» разработать конфигурацию для ведения учета и оборота легковых автомобилей автосалона.

Текущий этап: добавить в конфигурацию документы «Приходная накладная» и «Оказание услуг». В документах обязательно должна присутствовать табличная часть. Суммы и итог документа должны подсчитываться при каждом изменении элемента табличной части документа. Номер документа генерируется автоматически.

Продумать и (согласовав предварительно с преподавателем) добавить остальные документы конфигурации.