**СПбГТУ**

Вечерний Факультет Технологий Экономики и Менеджмента

**Реферат на тему :**

 **«Программы системы 1С: Предприятие.**

**Конфигурирование и администратирование в программах 1С: Предприятие»**

## Принял: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Группа: 5074

**г. Санкт-Петербург**

**2002**

**Содержание**

|  |  |
| --- | --- |
| * **Введение**
 | **Стр. 3** |
| **Функциональные компоненты** | **Стр. 3** |
| * **Типовые Конфигурации системы 1С: Предприятие**
 |  |
| * **Типовая конфигурация 1С: Бухгалтерия 7.7**
 | **Стр. 4** |
| * **Типовая конфигурация 1С: ТОРГОВЛЯ И СКЛАД 7.7**
 | **Стр. 5** |
| * **Типовая конфигурация 1С: Зарплата и Кадры 7.7**
 | **Стр. 6** |
| * **1С: Предприятие 7.7". Комплексная поставка**
 | **Стр. 7** |
| Версии программных продуктов | **Стр. 8** |
| * **Распределенная информационная система**
 | **Стр. 8** |
| Понятие конфигурации в системе программ 1С: Предприятие | **Стр. 9** |
| * Технологические средства конфигурирования и администрирования системы 1С: Предприятие
 | **Стр. 11** |
| * **Описание встроенного языка**
 | **Стр. 13** |
| * **Пример программного модуля**
 | **Стр. 26** |

Введение

1С: Предприятие является системой программ для автоматизации различных областей экономической деятельности. В конкретный программный продукт, входящий в систему программ 1С: Предприятие, включаются те функции и возможности, которые отвечают назначению этого продукта.

Все составляющие системы программ 1С: Предприятие можно разделить на Технологическую платформу и Конфигурации. Технологическая платформа представляет собой набор различных механизмов, используемых для автоматизации экономической деятельности и не зависящих от конкретного законодательства и методологии учета. Конфигурации являются собственно прикладными решениями. Каждая конфигурация ориентирована на автоматизацию определенной сферы экономической деятельности и, разумеется, отвечает принятому законодательству.

## **Функциональные компоненты**

Технологическая платформа, кроме механизмов, используемых во всех продуктах 1С: Предприятия, включает три функциональные основные компоненты. Функциональные компоненты включаются в состав продуктов системы 1С: Предприятие, использующих специфические возможности компонент.

**Компонента «Бухгалтерский учет»** предназначена для ведения учета на основе бухгалтерских операций. Она обеспечивает ведение планов счетов, ввод проводок, получение бухгалтерских итогов. Компонента используется для автоматизации бухгалтерского учета в соответствии с любым законодательством и методологией учета.

**Компонента «Расчет»** предназначена для выполнения сложных периодических расчетов. Она может использоваться для расчета заработной платы любой сложности, расчетов по ценным бумагам и других видов расчетов.

**Компонента «Оперативный учет»** предназначена для учета наличия и движения средств в самых различных разрезах в реальном времени. Она используется для учета запасов товарно-материальных ценностей, взаиморасчетов с контрагентами и т.д. Компонента позволяет отражать в учете операции хозяйственной жизни предприятия непосредственно в момент их совершения.

**Типовые Конфигурации системы 1С: Предприятие**

В комплект поставки программных продуктов системы программ 1С: Предприятие включаются типовые конфигурации. Они представляют собой универсальные прикладные решения для автоматизации конкретной области экономики.

Конфигурации, использующие возможности одной функциональной компоненты, предназначены для автоматизации отдельных сфер деятельности предприятий, например, бухгалтерского учета, торгового учета или расчета заработной платы. Конфигурации, использующие возможности нескольких компонент, обеспечивают комплексную автоматизацию различных направлений учета в единой информационной базе.

Особенностью системы программ 1С: Предприятие является возможность изменения конфигурации самим пользователем или организациями, специализирующимися на внедрении и поддержке программных продуктов фирмы "1С". Эта возможность позволяет обеспечить максимальное соответствие автоматизированной системы особенностям учета в конкретной организации.

Рассмотрим несколько типовых конфигураций, содержащих выше перечисленные функциональные компоненты.

###

### Типовая конфигурация 1С: Бухгалтерия 7.7 (Компонента «Бухгалтерский учет»)

### «1С: Бухгалтерия» - универсальная программа массового назначения для автоматизации бухгалтерского учета. Она может использоваться на предприятиях разных направлений деятельности и форм собственности.

### «1С: Бухгалтерия» предоставляет бухгалтеру гибкие возможности учета:

* использование нескольких планов счетов одновременно
* многоуровневые планы счетов
* многомерный аналитический учет
* многоуровневый аналитический учет
* количественный учет
* многовалютный учет по неограниченному числу валют
* ведение на одном компьютере учета для нескольких предприятий
* ведение консолидированного учета
* учет импортных товаров в разрезе ГТД (автоматический подбор необходимых номеров ГТД при выписке счетов-фактур)

### Программа позволяет автоматизировать ведение всех разделов бухгалтерского учета:

* операции по банку и кассе
* основные средства и нематериальные активы
* материалы и МБП
* товары и услуги, выполнение работ
* учет производства продукции
* учет валютных операций
* взаиморасчеты с организациями
* расчеты с подотчетными лицами
* расчеты по зарплате
* расчеты с бюджетом
* другие разделы учета.

1С: Бухгалтерия позволяет автоматизировать подготовку любых первичных документов:

* платежные поручения
* счета на оплату и счета-фактуры
* приходные и расходные кассовые ордера
* акты, накладные, требования, доверенности
* другие документы.

Средства работы с документами позволяют организовать ввод документов, их произвольное распределение по журналам и поиск любого документами по различным критериям: номеру, дате, сумме, контрагенту.

### Удобный пользовательский интерфейс

* Путеводитель по конфигурации предназначен для быстрого освоения программы бухгалтером. Он позволяет ознакомиться с порядком использования программы по конкретным разделам учета и тут же выполнить необходимые действия.
* Режим комментирования проведения документов. В процессе автоматического формирования проводок документами программа информирует пользователя о том, почему были сформированы те или иные проводки.
* "Советы дня" подскажут вам эффективные приемы работы и удобные возможности системы.
* Настраиваемые панели инструментов позволят вам быстро получать доступ к часто используемым функциям программы.

**Типовая конфигурация 1С: ТОРГОВЛЯ И СКЛАД 7.7 (Компонента «Оперативный учет»)**

"1С: Торговля и склад" предназначена для учета любых видов торговых операций. Благодаря гибкости и настраиваемости, система способна выполнять все функции учета - от ведения справочников и ввода первичных документов до получения различных ведомостей и аналитическихотчетов.

**"1С: Торговля и склад"** автоматизирует работу на всех этапах деятельности предприятия и позволяет:

* вести раздельный управленческий и финансовый учет
* вести учет от имени нескольких юридических лиц
* вести партионный учет товарного запаса с возможностью выбора метода списания себестоимости (FIFO, LIFO, по средней)
* вести раздельный учет собственных товаров и товаров, взятых на реализацию
* оформлять закупку и продажу товаров
* производить автоматическое начальное заполнение документов на основе ранее введенных данных
* вести учет взаиморасчетов с покупателями и поставщиками, детализировать взаиморасчеты по отдельным договорам
* формировать необходимые первичные документы
* оформлять счета-фактуры, автоматически строить книгу продаж и книгу покупок, вести количественный учет в разрезе номеров ГТД
* выполнять резервирование товаров и контроль оплаты
* вести учет денежных средств на расчетных счетах и в кассе
* вести учет товарных кредитов и контроль их погашения
* вести учет переданных на реализацию товаров, их возврат и оплату

**В "1С: Торговля и склад" вы можете:**

* задать для каждого товара необходимое количество цен разного типа, хранить цены поставщиков, автоматически контролировать и оперативно изменять уровень цен
* работать с взаимосвязанными документами
* выполнять автоматический расчет цен списания товаров
* быстро вносить изменения с помощью групповых обработок справочников и документов
* вести учет товаров в различных единицах измерения,
* а денежных средств - в различных валютах
* получать самую разнообразную отчетную и аналитическую информацию о движении товаров и денег
* автоматически формировать бухгалтерские проводки для 1С: Бухгалтерии.

Надежность и безопасность

**"1С: Торговля и склад"** содержит средства обеспечения сохранности и непротиворечивости информации:

* возможность запрещения пользователям "прямого" удаления информации
* специальный режим удаления данных с контролем перекрестных ссылок
* возможность запрещения пользователям редактировать данные за прошлые отчетные периоды
* установка запрета на редактирование печатных форм документов
* "запирание" системы пользователем при временном прекращении работы.

### 1С: Зарплата и Кадры 7.7 (Компонента «Расчет»)

 Программа "1С: Зарплата и Кадры" предназначена для расчета заработной платы и кадрового учета, может использоваться как на хозрасчетных предприятиях Российской Федерации, так и в организациях с бюджетным финансированием. Она позволяет не только автоматизировать расчет заработной платы, но и организовать учет сотрудников, регистрировать служебные перемещения, получать статистические справки по кадровому составу. Универсальность системы позволяет реализовать любой подход к решению этих задач и получать любые отчетные документы.

## **Основные функциональные возможности:**

Первоначальные возможности

* + Разнообразные возможности, поддерживаемые конфигурацией
	+ Все необходимые отчеты и документы

Расчет заработной платы

* + Различные системы и формы оплаты труда
	+ Учет отработанного времени
	+ Учет северных и районных особенностей

Кадровый учет и штатное расписание

* + Автоматизация кадрового учета и штатного расписания

Особенности бюджетных организаций

* + Учет специфики бюджетных организаций.

Дополнительные возможности системы

* + Разнообразные возможности настройки конфигурации
	+ Работа с распределенными информационными базами
	+ Администрирование, работа в сети и права доступа
	+ Дополнительный сервис

**1С: Предприятие 7.7". Комплексная поставка "Бухгалтерия; Зарплата и Кадры;  Производство и Услуги; Торговый и складской учет"**

Система программ "1С: Предприятие" предоставляет широкие возможности ведения автоматизированного учета на предприятиях, в организациях и учреждениях, независимо от их вида деятельности и формы собственности, с различным уровнем сложности учета.

"1С: Предприятие" позволяет организовать эффективный бухгалтерский, кадровый, оперативный торговый, складской и производственный учет, а также расчет заработной платы.

В комплексную поставку входят основные компоненты системы программ "1С: Предприятие":

- "Бухгалтерский учет",

- "Оперативный учет",

- "Расчет",

а также основные конфигурации:

- "Бухгалтерский учет"

-"Торговля +Склад"

-"Зарплата + Кадры"

-"Производство + Услуги + Бухгалтерия"

-"Бухгалтерия + Торговля + Склад + Зарплата + Кадры"

Пользователи могут применять конфигурации, входящие в новую комплексную поставку, как по отдельности, так и совместно, подобрав для себя подходящий вариант работы с системой. Выбор конфигурации зависит, прежде всего, от решаемых задач, от типа деятельности и структуры конкретного предприятия, уровня сложности ведения учета и других условий.

Пользователи могут вести учет в комплексной конфигурации или решать разные задачи учета в отдельных конфигурациях, пользуясь средствами обмена данных, или же начать с автоматизации одного из направлений учета, используя отдельную конфигурацию.

**Комплексная конфигурация "Бухгалтерия + Торговля +Склад + Зарплата + Кадры"**

Сохраняя возможности основных конфигураций "Бухгалтерский учет", "Торговля + Склад" и "Зарплата + Кадры", комплексная конфигурация обеспечивает интегрированное ведение учета:

Единую систему нормативно-справочной информации;

Автоматическое отражение торгово-складских операций и расчета заработной платы в бухгалтерском учете;

Финансовый учет по нескольким юридическим лицам;

Консолидированный управленческий учет.

# Версии программных продуктов

Различные версии продуктов системы программ 1С: Предприятия могут использоваться в организациях с различными объемами информации, различным количеством пользователей автоматизированной системы и различной интенсивностью работы пользователей.

Однопользовательские версии программ обеспечивают работу с информационной базой только одного пользователя (в один момент времени). Однопользовательские версии используются в основном на небольших предприятиях.

Сетевые версии позволяют работать с одной информационной базой одновременно нескольким пользователям. Сетевые версии используются тогда, когда необходимо, чтобы с автоматизированной системой работали одновременно несколько сотрудников. Существуют версии, допускающие работу ограниченного числа пользователей, и без ограничения числа пользователей.

Версии для SQL обеспечивают возможность работы с информационной базой в режиме клиент-сервер. Это позволяет обеспечить большую устойчивость и надежность функционирования системы, а также увеличивает производительность системы, особенно при работе большого количества пользователей.

Базовые версии программ системы 1С: Предприятия рассчитаны на работу одного пользователя и, в отличие от других версий, не включают средства для изменения конфигурации. Они используются для ведения достаточно простого учета, не требующего настройки на особенности конкретной организации.

#### Распределенная информационная система

Для тех организаций, в которых необходимо поддерживать единую автоматизированную систему в нескольких территориально удаленных точках (подразделениях, филиалах, складах и т.д.), существует дополнительная компонента 1С: Предприятия "Управление распределенными информационными базами". Ввод данных в информационных базах распределенной системы ведется независимо. Компонента обеспечивает полную или выборочную синхронизацию данных всех информационных баз, входящих в распределенную систему. Компонента поставляется отдельно и устанавливается только на одной (центральной) информационной базе.

##### Понятие конфигурации в системе программ 1С: Предприятие

Функционирование системы делится на два процесса – конфигурирование (описание модели предметной области средствами систем) и исполнение (обработку данных предметной области).

Результатом конфигурирования является Конфигурация, которая представляет собой модель предметной области.

Конфигурацией в системе 1С: Предприятие называется совокупность трех взаимосвязанных составных частей:

* Структуры метаданных
* Набора пользовательских интерфейсов
* Набора прав

**Метаданными** в системе 1С: Предприятие называется совокупность объектов метаданных, настроенных на хранение и обработку информации о хозяйственной деятельность конкретного предприятия.

**Под объектами метаданных** понимается формальное описание группы понятий предметной область со сходными характеристиками и одинаковым предназначением.

Рассмотрим объекты метаданных:

* Константы

Предназначены для работы с постоянной и условно постоянной информацией

* Справочники

Предназначены для работы с постоянной и условна постоянной информацией. Обычно справочниками являются списки материалов, товаров, организаций, валют, сотрудников и др. Справочники могут быть многоуровневыми

* Перечисления

Перечисления используются для описания постоянных наборов значений, не изменяемых для конкретной конфигурации. На этапе конфигурирования можно описать практически неограниченное количество видов перечислений. В отличие от справочника, значения перечислений задаются на этапе конфигурирования, и не могут быть изменены на этапе исполнения.

Типичными примерами перечислений являются виды оплаты (наличная, безналичная, бартер), статусу клиента (постоянный, разовый).

* Документы

Документы предназначены для хранения основной информации обо всех событиях, происходящих на предприятии, и, разумеется, имеющих смысл с точки зрения экономики. При помощи документов отражаются и платежи с расчетного счета, и операции по кассе, и кадровые перемещения, и движения по складу, прочие подобные события. Примерами документов могут служить такие документы как «Платежное поручение», «Расходная накладная», «Приходно-кассовый ордер» и т.д.

* Журналы документов

Журналы документов предназначены для просмотра документов. Каждый вид документа может быть отнесен к определенному журналу. Сам журнал документов не добавляет новых данных в систему, а служит только как средство просмотра списка документов одного или нескольких видов.

* Отчеты и обработки

Для описания отчетов и процедур произвольной обработки на этапе конфигурирования может быть создано неограниченное количество форм отчетов. Каждая форма имеет свой диалог настройки, который позволяет определить набор параметров формирования отчетов. Например, для выдачи складской справки – выбрать конкретный склад.

* Бухгалтерские счета (компонента «Бухгалтерский учет»)

Предназначены для хранения планов счетов бухгалтерского учета, то есть объектов синтетического учета средств предприятия. Суть данных объектов вполне соответствует общепринятому пониманию бухгалтерских счетов. Может одновременно поддерживаться несколько планов счетов, каждый из которых может иметь специфические свойства.

* Виды субконто

Это специальные объекты, для ведения аналитического учет по счетам бухгалтерского учета. В качестве субконто могут выступать основные средств, организации, товары и другие наборы объектов, как реально существующих, так и абстрактных. Например, субконто могут быть виды бюджетов – федеральный, территориальный, местный.

* Операции и проводки

Отражение движений средств в бухгалтерском учете записывается в виде операций и проводок. Операция является полным отражением в бухгалтерском учете хозяйственной операции, происшедшей на предприятии. Операция может содержать несколько проводок. Проводки не существуют отдельно от операций. Каждая проводка принадлежит одной и только одной операции. Операция в свою очередь, всегда принадлежит документу, причем у документа может быть только одна операция. Формирование операции и проводок происходит при проведении документа.

* Регистры (компонента «Оперативный учет»)

Предназначены для анализа остатков и движений средств в системе 1С: Предприятие.

Регистр представляет собой многомерную систему хранения остатков или оборотов. Каждый регистр на этапе конфигурации описывается набором измерений и ресурсов.

Под измерением понимается набор значений, которые детализируют движения средств, и в разрезе которых хранятся остатки; под ресурсом – числовая величина, которая является количественным или суммовым значением, отражающих размер движение (остатка).

* Журналы расчетов (компонента «Расчет»)

Журнал расчетов предназначен для хранения данных сложных периодических расчетов, поддерживаемых компонентой «Расчет», и их предыстории. Основным свойством журнала расчетов является их принадлежность тому или иному справочнику системы. Такой справочник называется справочником объектов расчета. Фактически справочник определяет вид объектов, по которым ведется расчет. Например, это может быть: журнал расчетов зарплаты, причем списком объектов расчета будет выступать справочник сотрудников.

* Виды расчетов

Используются для описания алгоритмов, по которым выполняются те или иные вычисления.

* Группы расчетов

Служат для того, чтобы при тех или иных расчетах можно было оперировать не только результатами расчетов по конкретным видам, но и результатами по нескольким видам расчетов, объединенных по определенному принципу.

* Календари

Для ведения рабочего графика предприятия или сторонних организаций, для учета рабочего времени сотрудников разных категорий служат календари.

**Под пользовательским интерфейсом** в системе1С: Предприятие понимается совокупность команд главного меню и панелей инструментов, настроенных на работу с конкретными объектами данных – документами, справочниками, журналами и т.д. Как правило, пользовательский интерфейс создается для конкретной категории пользователей. Цель создания интерфейса – обеспечить быстрый доступ пользователей к той информации, которая необходима им в соответствии с их обязанностями.

**Под набором прав** в системе1С: Предприятие понимается определение полномочий пользователей на работу с информацией, которая обрабатывается в системе.

Большинство объектов метаданных в системе1С: Предприятие могут иметь визуальное представление. В самом лучшем случае визуальное представление состоит из следующих частей:

* Экранный диалог, используемый для ввода и редактирования информации, хранящейся в объекте метаданных;
* Печатная форма объекта метаданных (печатных форм может быть несколько)
* Модуль формы - программа на встроенном языке системы1С: Предприятие. Как правило, модуль формы содержит алгоритм построения печатной формы объекта метаданных, а также может выполнять обработку вводимой в диалог информации для целей входного контроля, выполнения расчетов и т.п.

Технологические средства конфигурирования и администрирования системы 1С: Предприятие

###### Встроенный программный язык

 Необходимость наличия встроенного языка определена концепцией настраиваемости системы. Синтаксис встроенного языка вполне отвечает стандартам высокоуровневых языков.

 Язык является предметно-ореиентированным. Он поддерживает специализированные типы данных предметной области, определяемые конфигурацией системы. Работа с этими типами данных в языке организована с использованием объектной техники. Язык ориентирован на пользователей различной квалификации. В частности, его отличает мягкая типизация данных (обеспечивающая быстрое написание программных модулей) и жесткий контроль синтаксических конструкций, уменьшающий вероятность ошибок.

 Так как система сочетает в себе визуальные и языковые средства конфигурирования, использование встроенного языка в системе имеет событийно-зависимую ориентацию, то есть языковые модули используются в конкретных местах для отработки отдельных алгоритмов, настраиваемых в процессе конфигурации. Так, например, для документа можно описать алгоритм автоматического заполнения реквизитов при вводе нового документа. Данная процедура будет вызвана системой в нужный момент.

**Механизм запросов**

Для получения произвольных отчетов сложной структуры в системе предусмотрен предметно-ориентированный механизм запросов. Данное средство опирается на существующую условно-переменную структуру информационной базы системы, что позволяет сравнительно просто описывать достаточно сложные запросы.

**Встроенный текстовый редактор**

Используется системой для создания программных модулей на встроенном ящике и для редактирования документов в текстовом виде.

 Одной из особенностей редактора является возможность контекстного выделения цветом синтаксических конструкций встроенного языка.

**Встроенный редактор диалогов**

Используется для произвольной настройки форм ввода и редактирования информации. Позволяет оформить большинство окон, которые используются в системе для ввода и просмотра предметной информации.

**Встроенный редактор табличных документов**

Для всех выходных документов (первичных документов и отчетов) в системе предусмотрен единый формат – формат табличных документов. Это мощное средство, сочетающее в себе оформительские возможности табличной структуры и векторной графики. Таким образом, он может быть использован как для создания небольших документов с очень сложной структурой линий, так и для объемных ведомостей, журналов и других подобных документов.

**Конструкторы**

Конструкторы – вспомогательные инструменты, облегчающие разработку стандартных элементов системы 1С: Предприятие. В системе имеются конструкторы справочника, документа, журнала документов, отчета и вида субконто. Еще пять конструкторов облегчают разработку программных модулей в стандартных случаях.

**Система настройки пользовательских интерфейсов**

Для того чтобы интерфейс конкретной конфигурации системы полностью отражал настроенные структуры данных и алгоритмы, в системе 1С: Предприятие, помимо редактора диалоговых форм и табличных документов, предусмотрена возможность настройки общих интерфейсных компонент системы: меню, панелей инструментов, комбинаций клавиш. На этапе конфигурирования может быть создано несколько пользовательских интерфейсов для разных категорий пользователей (руководителей, менеджеров, кладовщиков и др.)

**Система настройки прав пользователей и авторизации доступа**

Данная система позволяет описывать наборы прав, соответствующие должности пользователей. Структура прав определяется конкретной конфигурацией системы. Например, могут быть введены такие наборы прав, как «Главный бухгалтер», «Кладовщик», «Менеджер» и т.д.

**Отладчик**

Для удобства разработки конфигурации в системе предусмотрен отладчик. Отладчик позволяет прослеживать исполнение программных модулей конфигурации, замерять сравнительное время исполнения, просматривать содержимое переменных.

**Администрирование роботы пользователей**

Для отслеживания текущего состояния работы системы используется монитор пользователей. Он позволяет просмотреть, кто из пользователей в настоящий момент работает с конкретной информационной базой, и в каком режиме.

**Журнал регистрации изменений**

Журнал ведется системой автоматически. В нем отражаются все факты изменений данных пользователями.

**Описание встроенного языка**

## Назначение и краткая характеристика встроенного языка

Встроенный язык системы 1С: Предприятие предназначен для описания (на стадии разработки конфигурации) алгоритмов функционирования прикладной задачи.

Встроенный язык (далее по тексту — язык) представляет собой предметно-ориентированный язык программирования, специально разработанный с уче­том возможности его применения не только профессиональными программи­стами. В частности, все операторы языка имеют как русское, так и англоязыч­ное написание, которые можно использовать одновременно в одном исходном тексте. Основной язык, описываемый в данной книге — русский, однако для каждого оператора языка приводится его англоязычный синоним.

При своей относительной простоте язык обладает некоторыми объектно-ориентированными возможностями, например, правила доступа к атрибутам и методам специализированных типов данных (документам, справочникам и т. п.) подобны свойствам и методам объектов, используемых в других объектно-ориентированных языках. Однако специализированные типы данных не могут определяться средствами самого языка, а задаются в визуальном режиме кон­фигуратора.

Типизация переменных в языке не жесткая, т. е. тип переменной определя­ется ее значением. Переменные не обязательно объявлять в явном виде. Неяв­ным определением переменной является ее первое упоминание в левой части оператора присваивания. Возможно также явное объявление переменных при помощи соответствующего оператора. Допускается применение массивов.

## Что такое программный модуль?

Программные модули в конфигурации системы 1С: Предприятие не являют­ся самостоятельными программами в общепринятом понимании этого слова, поскольку они являются только частью всей *конфигурации задачи*. Программ­ный модуль — это своего рода «контейнер» для размещения текстов процедур и функций, вызываемых системой во время исполнения задачи в определенные моменты работы. Поэтому программный модуль не имеет формальных границ своего описания типа: «Начало модуля» — «Конец модуля».

Место размещения конкретного программного модуля (тот самый «контей­нер») предоставляется конфигуратором в тех точках конфигурации задачи, ко­торые требуют описания специфических алгоритмов функционирования. Эти алгоритмы следует оформлять в виде процедур или функций, которые будут вызваны самой системой в заранее предусмотренных ситуациях (например, при нажатии кнопки в диалоговом окне).

Каждый отдельный программный модуль воспринимается системой как единое целое, поэтому все процедуры и функции программного модуля вы­полняются в едином контексте.

### Контекст выполнения программного модуля

Каждый программный модуль связан с остальной частью конфигурации за­дачи. Эта связь называется контекстом выполнения модуля. Следует различать два вида контекста:

·         глобальный контекст задачи;

·         локальный контекст выполнения конкретного модуля.

***Глобальный контекст*** образуется:

·         значениями системных атрибутов, системными процедурами и функ­циями;

·         значениями заданных в конфигураторе *констант, перечислений, регист­ров, видов расчета, групп видов расчета*;

·         переменными, процедурами и функциями глобального программного модуля, объявленными с ключевым словом Экспорт.

Глобальный контекст виден всем программным модулям и определяет об­щую языковую среду конфигурации.

***Локальный контекст*** модуля образуется тем конкретным местом конфигу­рации задачи, для которого использован программный модуль. Локальный кон­текст виден только конкретному программному модулю и определяет для мо­дуля набор непосредственно доступных модулю значений агрегатных типов данных, их атрибутов и методов (см. «Виды программных модулей»). Однако, контекст модуля можно передать как объект в виде параметра при вызове про­цедур и функций (см. «Передача локального контекста программного модуля в качестве параметра»). Кроме того, контекст модуля определяет тот набор мето­дов, которые доступны только в данном контексте (см. «Атрибуты и методы контекста Модуля формы», «Методы контекста Модуля формы элемента спра­вочника» и т. п.). Локальный контекст предназначен для того, чтобы дать воз­можность управлять частными аспектами поведения задачи, присущими дан­ному модулю.

### Виды программных модулей

В системе 1С: Предприятие существуют следующие виды программных мо­дулей (места размещения программных модулей в конфигурации задачи), дос­тупных в конфигураторе:

|  |  |  |
| --- | --- | --- |
| **Размещение** | **Момент запуска** | **Контекст выполнения** |
| Глобальный модуль |
| Размещается в корне­вом разделе конфигу­рации: Метаданные. | Запускается при стар­те всей прикладной задачи. | Глобальный модуль оп­ределяет глобальный контекст всей задачи. |
| Модуль Формы списка справочника |
| Размещается в разделе конфигурации: Мета­данные — Справочник — Форма списка. | Запускается при вызо­ве формы списка справочника. | В модуле доступны: глобальный контекст, контекст *Модуля формы списка справочника*, в котором непосредствен­но доступен выбранный в списке элемент спра­вочника и реквизиты формы списка справоч­ника. |
| Модуль Формы группы справочника |
| Размещается в разделе конфигурации: Мета­данные — Справочник — Форма группы. | Запускается при от­крытии формы группы справочника. | В модуле доступны: глобальный контекст, контекст *Модуля формы группы справочника*, в котором непосредствен­но доступны реквизиты текущей группы спра­вочника и реквизиты формы. |
| Модуль Формы элемента справочника |
| Размещается в разделе конфигурации: Мета­данные — Справочник — Форма элемента. | Запускается при от­крытии формы эле­мента справочника. | В модуле доступны: глобальный контекст, контекст *Модуля формы элемента справочника*, в котором непосредствен­но доступны реквизиты текущего элемента спра­вочника и реквизиты формы. |
|  Модуль Формы документа |
| Размещается в разделе конфигурации: Мета­данные — Документ — Форма. | Запускается при от­крытии формы доку­мента. | В модуле доступны: глобальный контекст, контекст *Модуля формы документа*, в котором непосредственно дос­тупны реквизиты теку­щего документа и рек­визиты формы докумен­та. |
| Модуль документа |
| Размещается в разделеконфигурации: Мета­данные — Документ — Модуль документа. | Запускается при про­ведении документа, при удалении прове­денного документа, при снятии проведе­ния, при выполнении архивации записей журнала расчетов, по­рожденных докумен­том. | В модуле доступны: глобальный контекст, контекст *Модуля доку­мента*, в котором непо­средственно доступны реквизиты текущего документа. |
| Модуль Формы журнала документов |
| Размещается в разделе конфигурации: Мета­данные — Журнал — Форма. | Запускается при вызо­ве формы журнала документов. | В модуле доступны: глобальный контекст, контекст *Модуля формы журнала документов*, в котором непосредствен­но доступен выбранный в журнале документ и реквизиты формы жур­нала. |
| Модуль Формы журнала расчетов |
| Размещается в разделе конфигурации: Мета­данные — Журнал расчетов — Форма. | Запускается при вызо­ве формы журнала расчетов. | В модуле доступны: глобальный контекст, контекст *Модуля формы журнала расчетов*, в котором непосредствен­но доступны реквизиты журнала расчетов и рек­визиты формы. |
| Модуль Формы списка счетов |
| Размещается в разделе конфигурации: Мета­данные — План сче­тов. | Запускается при вызо­ве формы списка сче­тов. | В модуле доступны: глобальный контекст, контекст *Модуля формы списка счетов*, в котором непосредственно досту­пен выбранный в списке счет и реквизиты формы списка счетов. |
| Модуль Формы счета |
| Размещается в разделе конфигурации: Мета­данные — Справочник— счет. | Запускается при от­крытии формы счета. | В модуле доступны: глобальный контекст, контекст *Модуля формы счета*, в котором непо­средственно доступны реквизиты текущего счета и реквизиты фор­мы. |
| Модуль Формы журнала операций |
| Размещается в разделе конфигурации: Мета­данные — Журнал операций — Форма. | Запускается при вызо­ве формы журнала операций. | В модуле доступны: глобальный контекст, контекст *Модуля формы журнала операций*, в котором непосредствен­но доступны реквизиты журнала операций и реквизиты формы. |
| Модуль Формы операции |
| Размещается в разделе конфигурации: Мета­данные — Операция. | Запускается при от­крытии формы опера­ции. | В модуле доступны: глобальный контекст, контекст *Модуля формы операции*, в котором непосредственно дос­тупны реквизиты теку­щей операции и рекви­зиты формы операции. |
|  Модуль Формы журнала проводок |
| Размещается в разделе конфигурации: Мета­данные — Журнал проводок — Форма. | Запускается при вызо­ве формы журнала проводок. | В модуле доступны: глобальный контекст, контекст *Модуля формы журнала проводок*, в котором непосредствен­но доступны реквизиты журнала проводок и реквизиты формы. |
| Модуль Формы отчета |
| Размещается в разделе конфигуратора: Мета­данные — Отчет — Форма. | Запускается при от­крытии диалоговой формы подготовки отчета. | В модуле доступны: глобальный контекст, контекст *Модуля формы отчета*, в котором непо­средственно доступны реквизиты формы. |
| Модуль Формы обработки |
| Размещается в разделе конфигуратора: Мета­данные — Обработка — Форма. | Запускается при от­крытии диалоговой формы обработки. | В модуле доступны: глобальный контекст, контекст *Модуля формы обработки*, в котором непосредственно дос­тупны реквизиты фор­мы. |
| Модуль вида расчета |
| Размещается в разделе конфигуратора: Метаданные — Вид расчета — Модуль вида расчета. | Запускается при рас­чете соответствующих записей журнала рас­четов. | В модуле доступны: глобальный контекст, контекст *Модуля вида расчета*, в котором дос­тупны реквизиты  |

## Описания элементов языка

В программах 1С:Предприятие существующие элементы (конструкции) языка можно разделить на:

* Структурные конструкции
* Системные конструкции
* Управляющие операторы
* Выражения и операторы присваивания
* Системные константы
* Системные процедуры и функции
* Системные предопределенные процедуры

Мы рассмотрим «структурные конструкции и управляющие операторы»

**Структурные конструкции**

### Процедура

Секция описания процедуры.

**Синтаксис:**

Процедура <Имя\_проц>([[Знач] <Парам1>[=<ДефЗнач>],

... , [Знач] <ПapaмN>[=<ДeфЗнaч>]]) [Экспорт]

//Объявления локальных переменных;

//Операторы;

...

[Возврат;]

//Операторы;

...

КонецПроцедуры

**Параметры:**

|  |  |
| --- | --- |
| <Имя\_проц> | Назначает имя процедуры. |
| Знач | Необязательное ключевое слово, которое указывает на то, что следующий за ним параметр передается по значению, т. е. изменение значения формального параметра при выполнении процедуры никак не повлияет на фактический параметр, переданный при вызове процедуры. Если это ключевое слово не указано, то параметр процедуры передается по ссылке, то есть изменение внутри процедуры зна­чения формального параметра приведет к измене­нию значения соответствующего фактического па­раметра. |
| <Парам1>, ..., <ПарамN> | Необязательный список формальных параметров, разделяемых запятыми. Значения формальных па­раметров должны соответствовать значениям пере­даваемых при вызове процедуры фактических па­раметров. В этом списке определяются имена каж­дого из параметров так, как они используются в тексте процедуры. Список формальных параметров может быть пуст. **См. также:** «Передача парамет­ров». |
| =<ДефЗнач> | Необязательная установка значения параметра по умолчанию. Параметры с установленными значе­ниями по умолчанию можно располагать в любом месте списка формальных параметров.Если параметр при вызове процедуры опущен, то он принимает либо установленное по умолчанию значение (если оно есть) либо принимает «пустое» значение (значение неопределенного типа).Если параметру не задано значение по умолчанию, и он является последним в списке передавае­мых параметров, то при вызове процедуры его нельзя опускать.Если параметру задано значение по умолчанию, и он является последним в списке, то при вызове процедуры его можно опускать в списке передавае­мых фактических параметров и не ставить запятую перед опущенным параметром.Если параметру не задано значения по умолча­нию, то при вызове процедуры его можно опускать в списке передаваемых фактических параметров, но разделительную запятую надо ставить. |
| Экспорт | Необязательное ключевое слово, которое указывает на то, что данная процедура является доступной из других программных модулей. Имеет смысл только в *глобальном программном модуле*. |
| //Объявления локальных переменных | Объявляются локальные переменные, на которые можно ссылаться только в рамках этой процедуры (см. оператор Перем). |
| //Операторы | Исполняемые операторы процедуры. |
| Возврат | Необязательное ключевое слово, которое завершает выполнение процедуры и осуществляет возврат в точку программы, из которой было обращение к процедуре. Использование данного оператора в процедуре не обязательно. |
| КонецПроцедуры | Обязательное ключевое слово, обозначающее конец исходного текста процедуры, завершение выполне­ния процедуры. Возврат в точку, из которой было обращение к процедуре. |

**Описание:**

Ключевое слово Процедура начинает секцию исходного текста, выполнение которого можно инициировать из любой точки программного модуля, просто указав Имя\_процедуры со списком параметров (если параметры не передают­ся, то круглые скобки, тем не менее, обязательны). Если в *глобальном про­граммном модуле* в теле описания функции использовано ключевое слово Экспорт, то это означает, что данная процедура является доступной из всех других программных модулей конфигурации. Ключевое слово Экспорт имеет смысл использовать только в *глобальном программном модуле*.

### Функция

Секция описания функции.

**Синтаксис:**

Функция<Имя\_функции>([[Знач]<Парам1>[=<ДефЗнач>], ...

,[Знач]<ПарамN>[=<ДефЗнач>]])[Экспорт]

//Объявлениялокальных переменных;

//Операторы;

...

Возврат<ВозвращаемоеЗначение>;

//Операторы;

...

КонецФункции

**Параметры:**

|  |  |
| --- | --- |
| <Имя\_функции> | Назначает имя функции. |
| Знач | Необязательное ключевое слово, которое указывает на то, что следующий за ним параметр передается по значению, т.е. изменение значения формального па­раметра при выполнении функции никак не повлия­ет на значение фактического параметра, переданного при вызове функции. Если это ключевое слово не указано, то параметр функции передается по ссылке, то есть изменение внутри функции значения фор­мального параметра приведет к изменению значения соответствующего фактического параметра. |
| <Парам1>,...,<ПарамN> | Необязательный список формальных параметров, разделяемых запятыми. Значения формальных пара­метров должны соответствовать значениям передавае­мых при вызове функции фактических параметров. В этом списке определяются имена каждого из пара­метров так, как они используются в тексте функции. Список формальных параметров может быть пуст.  |
| =<ДефЗнач> | Необязательная установка значения параметра по умолчанию. Параметры с установленными значения­ми по умолчанию можно располагать в любом месте списка формальных параметров.Если параметр при вызове процедуры опущен, то он принимает либо установленное по умолчанию значение (если оно есть) либо принимает «пустое»значение (значение неопределенного типа).Если параметру не задано значение по умолчанию, и он является последним в списке передаваемых па­раметров, то при вызове процедуры его нельзя опус­кать.Если параметру задано значение по умолчанию, и он является последним в списке, то при вызове про­цедуры его можно опускать в списке передаваемых фактических параметров и не ставить запятую перед опущенным параметром.Если параметру не задано значения по умолчанию, то при вызове процедуры его можно опускать в спи­ске передаваемых фактических параметров, но разде­лительную запятую надо ставить. |
| Экспорт | Необязательное ключевое слово, которое указывает на то, что данная функция является доступной из других программных модулей. Данное ключевое сло­во имеет смысл использовать только в *глобальном программном модуле*. |
| //Объявлениялокальных переменных | Объявляются локальные переменные, на которые можно ссылаться только в рамках этой функции. |
| //Операторы | Исполняемые операторы функции. |
| Возврат | Ключевое слово, которое завершает выполнение функции и возвращает Возвращаемое Значение в выражение, в котором используется функция. Ис­пользование данного ключевого слова в функции обязательно. |
| <ВозвращаемоеЗначение> | Выражение, значение которого содержит результат обращения к функции. |
| КонецФункции | Обязательное ключевое слово, обозначающее конец исходного текста функции. |

**Описание:**

Ключевое слово Функция начинает секцию исходного текста функции, вы­полнение которой можно инициировать из любой точки программного модуля, просто указав Имя\_функции со списком параметров (если параметры не пере­даются, то круглые скобки, тем не менее, обязательны). Если в *глобальном про­граммном модуле* в теле описания функции использовано ключевое слово Экспорт, то это означает, что данная функция является доступной из всех дру­гих программных модулей конфигурации. Ключевое слово Экспорт имеет смысл использовать только в *глобальном программном модуле*.

Выполнение функции заканчивается обязательным оператором Возврат.Функции отличаются от процедур только тем, что возвращают Возвращаемое Значение. Конец программной секции функции определяется по оператору КонецФункции.

Вызов любой функции в тексте программного модуля можно записывать как вызов процедуры, т. е. в языке допускается не принимать от функции воз­вращаемое значение.

Переменные, объявленные в теле функции в разделе Объявления\_локальных\_переменных, являются локальными переменными данной функции, поэтому доступны только в этой функции (за исключением случая передачи их как параметров при вызове других процедур, функций или методов).

# Управляющие операторы

#### Если

Оператор условного выполнения.

Синтаксис:

Если <Логическое\_выражение> Тогда

 //операторы;

[ИначеЕсли <Логическое\_выражение> Тогда]

 //операторы;

[Иначе]

 //операторы;

КонецЕсли;

|  |  |
| --- | --- |
| <Логическое\_выражение> | Логическое выражение. |
| Тогда | *Операторы* следующие за Тогда вы­полняются, если результатом *логиче­ского выражения* в конструкции Если является истина. |
| //операторы | Исполняемый оператор или последова­тельность таких операторов. |
| ИначеЕсли | Логическое выражение, следующее за ключевым словом ИначеЕсли вычис­ляется только тогда, когда условия в Если и всех предшествующих ИначеЕсли оказались ложными. Опе­раторы следующие за конструкцией ИначеЕсли — Тогда выполняются, если результатом *логического выраже­ния* в данном ИначеЕсли является ис­тина. |
| Иначе | Операторы, следующие за ключевым словом Иначе выполняются, если ре­зультаты логических выражений в кон­струкции Если и всех предшествующих конструкциях ИначеЕсли оказались ложными. |
| КонецЕсли | Ключевое слово, которое завершает структуру оператора условного выпол­нения. |

 Описание:

Оператор Если управляет выполнением программы, основываясь на ре­зультате одного или более логических выражений. Оператор может содер­жать любое количество групп операторов, возглавляемых конструкциями

ИначеЕсли — Тогда.

#### Пока

Оператор выполнения цикла.

Синтаксис:

Пока <Логическое\_выражение> Цикл

 //операторы;

КонецЦикла;

**Параметры:**

|  |  |
| --- | --- |
| <Логическое\_выражение> | Логическое выражение. |
| Цикл | Операторы, следующие за ключевым сло­вом Цикл, выполняются, пока результа­том *логического выражения* является истина. |
| //операторы; | Исполняемый оператор или последова­тельность таких операторов. |
| КонецЦикла | Ключевое слово, которое завершает структуру оператора цикла. |

Описание:

Оператор цикла Пока предназначен для циклического повторения операто­ров, находящиеся внутри конструкции Цикл ... КонецЦикла. Цикл выполняет­ся пока *логическое выражение* — истина. Условие выполнения цикла все­гда проверяется вначале, перед выполнением цикла.

#### Для

Оператор выполнения цикла.

Синтаксис:

Для <Имя\_переменной> = <Выражение1> По <Выражение2> Цикл

 //Операторы;

КонецЦикла;

**Параметры:**

|  |  |
| --- | --- |
| <Имя\_переменной> | Идентификатор переменной (счетчика цикла), значение которой автоматически увеличивается при каждом повторении цикла. |
| <Выражение 1> | Числовое выражение, которое задает на­чальное значение, присваиваемое счетчику цикла при первом проходе цикла. |
| По | Синтаксическая связка для параметра <Выражение2>. |
| <Выражение 2> | Максимальное значение счетчика цикла. Когда переменная Имя\_переменной стано­вится больше чем <Выражение2>, выпол­нение оператора цикла Для прекращается. Величина приращения счетчика при каж­дом выполнении цикла равна 1. |
| Цикл | Операторы следующие за ключевым сло­вом Цикл выполняются, пока значение переменной Имя\_переменной меньше зна­чения <Выражение2>. |
| //Операторы | Исполняемый оператор или последова­тельность таких операторов. |
| КонецЦикла | Ключевое слово, которое завершает струк­туру оператора цикла. |

Описание:

Оператор цикла Для предназначен для циклического повторения операто­ров, находящиеся внутри конструкции Цикл ... КонецЦикла. Перед началом выполнения цикла значение <Выражение1> присваивается переменной <Имя\_переменной>. Значение <Имя\_переменной> автоматически увеличи­вается при каждом проходе цикла. Цикл выполняется пока значение перемен­ной <Имя\_переменной> меньше или равно значению <Выражение2>. Вели­чина приращения счетчика при каждом выполнении цикла равна 1. Условие выполнения цикла всегда проверяется вначале, перед выполнением цикла.

#### Попытка

Обработка исключительных ситуаций.

Синтаксис:

Попытка

 // Некоторые действия;

Исключение

 // Обработка исключительной ситуации;

КонецПопытки;

**Параметры:**

|  |  |
| --- | --- |
| // Некоторые действия | Исполняемый оператор или последова­тельность таких операторов. |
| // Обработка исключительной ситуации | Исполняемый оператор или последовательность операторов, которые обраба­тывают исключительную ситуацию. |
| Исключение | Операторы, следующие за ключевым словом Исключение выполняются, если при выполнении последовательно­сти операторов <// Некоторые действия> произошла ошибка времени вы­полнения. |
| КонецПопытки | Ключевое слово, которое завершает структуру оператора обработки исклю­чительных ситуаций. |

Описание:

Оператор Попытка управляет выполнением программы, основываясь на возникающие при выполнении модуля ошибочные (исключительные) ситуации и определяет обработку этих ситуаций.

В качестве ошибочных (исключительных) ситуаций воспринимаются ошиб­ки времени выполнения модуля. Не предусмотрено определяемых пользовате­лем исключений.

Если при выполнении последовательности операторов <// Некоторые действия> произошла ошибка времени выполнения, то выполнение оператора, вызвавшего ошибку прерывается и управление передается на первый оператор последовательности операторов <// Обработка исключительной ситуации>. При этом управление будет передано даже в том случае, если ошибку вызвал оператор, находящийся в процедуре или функции, вызванной из <// Некото­рые действия>. Если ошибка произошла в вызванной процедуре или функции, то ее выполнение будет прервано, а локальные переменные уничтожены. Это справедливо для любой вложенности вызовов. После выполнения последова­тельности операторов <// Обработка исключительной ситуации> управление передается на следующий за ключевым словом КонецПопытки оператор. Если же последовательность <// Некоторые действия> выполнилась без ошибок, то последовательность операторов <// Обработка исключительной ситуации> будет пропущена и управление также будет продолжено с оператора, следующе­го за ключевым словом КонецПопытки.

Конструкции Попытка-Исключение-КонецПопытки могут быть вложен­ными. При этом, при возникновении исключительной ситуации управление будет передано на самый «внутренний» обработчик, начинающийся с ключево­го слова Исключение. Последовательность операторов <// Обработка ис­ключительной ситуации> может содержать оператор ВызватьИсключение (англоязычный синоним Raise). Выполнение данного оператора прекращает выполнение последовательности <// Обработка исключительной ситуации> и производится поиск более «внешнего» обработчика. Если таковой есть, то управление передается на его первый оператор. Если нет, то выполнение моду­ля прекращается с выдачей сообщения о первоначально возникшей ошибке. Оператор ВызватьИсключение может встречаться только внутри операторных скобок Исключение ... КонецПопытки.

В выдаче диагностики помощь может оказать встроенная функция ОписаниеОшибки() (англоязычный синоним GetErrorDescription). Эта функция не имеет параметров, а в качестве значения возвращает описание ошибки, такое же, как было бы выдано в окне сообщений, в случае отсутствия обработчика исключительной ситуации, за исключением описания модуля и строки, в которой ошибка возникла. Применять данную функцию имеет смысл только при выполнении последовательности операторов <// Обработка исклю­чительной ситуации>, так как в ином случае она вернет пустую строку.

#### Перейти

Безусловная передача управления на другой оператор программы.

Синтаксис:

Перейти <Метка>;

**Параметры:**

|  |  |
| --- | --- |
| <Метка> | Метка другого исполняемого оператора в программе. |

Описание:

Безусловная передача управления на другой оператор программы. Оператор Перейти передает управление от одного оператора к другому. <Метка> в этом операторе не должна быть меткой перехода на оператор Процедура или Функция.

Область действия оператора Перейти ограничивается программным моду­лем, процедурой или функцией; он не может передать управление за пределы программного модуля, процедуры или функции.

Оператор безусловного перехода не может быть использован для передачи управления на операторы, находящиеся внутри конструкций: Пока ... КонецЦикла, Для ... КонецЦикла, Если ... ИначеЕсли ... Иначе ... КонецЕсли, Попытка ... Исключение ... КонецПопытки извне этих конструкций.

#### Продолжить

Передача управления в начало цикла.

Описание:

Передает управление в начало цикла. Оператор Продолжить немедленно передает управление в начало цикла, где производится вычисление и проверка условий выполнения цикла. Оператор Продолжить может использоваться только внутри конструкций операторов цикла Пока и Для.

#### Прервать

Прерывает выполнение цикла.

Синтаксис:

Прервать

Описание:

Прерывает выполнение цикла. Оператор Прервать приводит к немедлен­ному прекращению выполнения цикла и передает управление первому опера­тору, следующему за конструкцией оператора Пока или Для. Оператор Прервать может использоваться только внутри конструкций операторов цик­ла Пока и Для.

#### Возврат

Завершение процедуры или функции.

Синтаксис:

Возврат [<Выражение>]

**Параметры:**

|  |  |
| --- | --- |
| <Выражение> | Выражение, значение которого содержит результат обращения к функции. Этот параметр обязателен для оператора Возврат в теле описания функции, но не может использоваться в процедуре. |

Оператор Возврат завершает выполнение процедуры или функции и пере­дает управление в вызвавшую точку программы. Оператор Возврат обязателен в функции и необязателен в процедуре. Если в процедуре не используется опе­ратор Возврат, то его неявное выполнение происходит вслед за последним исполняемым оператором процедуры. Данный оператор может применяться только в теле описания процедуры или функции, причем <Выражение> может задаваться только в случае использования оператора в контексте функции.

**Символы**

|  |  |
| --- | --- |
| **Символ** | **Значение** |
| [ ] | В квадратных скобках заключаются необязательные син­таксические элементы. |
| ( ) | Круглые скобки заключают в себе список параметров. |
| | | Вертикальной линией разделяются синтаксические эле­менты, среди которых нужно выбрать только один. |

## **Специальные символы, используемые в исходном тексте**

|  |  |
| --- | --- |
| // | Двумя знаками «косая черта» начинается комментарий. Коммен­тарием считается весь текст от знака "//" до конца текущей стро­ки. |
| | | Вертикальная черта в начале строки используется только в стро­ковых константах и означает, что данная строка является про­должением предыдущей (перенос строки), (см. «Строковые кон­станты»). |
| ~ | Знаком тильда начинается метка оператора. |
| : | Двоеточием заканчивается метка оператора. |
| ; | Точка с запятой является символом разделения операторов. |
| ( ) | В круглые скобки заключается список параметров методов, про­цедур и функций. |
| [ ] | В квадратные скобки заключается размерность массивов. |
| ,  | Запятая разделяет параметры в списке параметров методов, про­цедур и функций. |
| " " | В двойные кавычки заключаются строковые константы. |
| ' ' | В одинарные кавычки заключаются константы даты. |
| . | Десятичная точка в числовых константах. Разделитель, используемый в описаниях агрегатных типов данных. |
| + | Символ «плюс» обозначает операцию сложения. |
| - | Символ «минус» обозначает операцию вычитания. |
| \* | Символ «звездочка» обозначает операцию умножения. |
| / | Символ «косая черта» обозначает операцию деления. |
| > | Правая угловая скобка обозначает логическую операцию «боль­ше». |
| >= | Логическая операция «больше или равно». |
| < | Левая угловая скобка обозначает логическую операцию «мень­ше». |
| <= | Логическая операция «меньше или равно». |
| = | Знак равенства обозначает присвоение или логическую операцию «равно». |
| <> | Две угловые скобки обозначают логическую операцию «не рав­но». |

**Пример**

Рассмотрим часть модуля документа «Счет» в программе «1С:Бухгалтерия»

**Процедура «Печать» - создание печатной формы документа.**

Процедура Печать()

 Таб = СоздатьОбъект("Таблица");

 ИмяФайлаПечатнойФормы = КаталогИБ()+"ExtForms\PrnForms\1cbs.mxl";

 Если ФС.СуществуетФайл(ИмяФайлаПечатнойФормы) = 1 Тогда

 Таб.ИсходнаяТаблица(ИмяФайлаПечатнойФормы);

 Иначе

 Таб.ИсходнаяТаблица("Таблица");

 КонецЕсли;

 Таб.ВывестиСекцию("Шапка");

 ВыбратьСтроки();

 Ном = 0;

 Пока ПолучитьСтроку()=1 Цикл

 Ном = Ном + 1;

 Таб.ВывестиСекцию("Строка");

 КонецЦикла;

 Таб.ВывестиСекцию("Итого");

 Таб.ВывестиСекцию("НДС");

 Если УчитыватьНП = 1 Тогда

 Таб.ВывестиСекцию("НП");

 КонецЕсли;

 Таб.ВывестиСекцию("Подвал");

КонецПроцедуры

**Процедура «Ввод на основании» - документ будет выводиться на основании другого документа.**

Процедура ВводНаОсновании(ДокОсн) //предопределенная

 Новый = 1;

 Если ДокОсн.Вид() = "РасходнаяНакладная" Тогда

 РасчетныйСчет = ДокОсн.РасчетныйСчет;

 Контрагент = ДокОсн.Контрагент;

 Договор = ДокОсн.Договор;

 Плательщик = ДокОсн.Контрагент;

 ВариантРасчетаНалогов = ДокОсн.ВариантРасчетаНалогов;

 ДокОсн.ВыбратьСтроки();

 Пока ДокОсн.ПолучитьСтроку() = 1 Цикл

 НоваяСтрока();

 Товар = ДокОсн.Товар;

 Количество = ДокОсн.Количество;

 Цена = ДокОсн.Цена;

 Сумма = ДокОсн.Сумма;

 НДС = ДокОсн.НДС;

 НП = ДокОсн.НП;

 Всего = ДокОсн.Всего;

 КонецЦикла;

 КонецЕсли;

КонецПроцедуры //ВводНаОсновании