

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**ВВЕДЕНИЕ В КОНФИГУРИРОВАНИЕ
СИСТЕМЫ 1С:ПРЕДПРИЯТИЕ**

САМАРА 2015

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

**ВВЕДЕНИЕ В КОНФИГУРИРОВАНИЕ
СИСТЕМЫ 1С:ПРЕДПРИЯТИЕ**

Методические указания

САМАРА

2015

УДК 519.682.7
ББК 32.973

Составители ***И.В. Семенова***

Рецензент к.ф.-м.н., А.С. Луканов

Введение в конфигурирование системы 1С:Предприятие: метод. указания / сост. *И.В. Семенова* – Самара, 2015. –78 с.: ил.

Пособие посвящено изучению общей концепции системы «1С:Предприятие 8.3», а также вопросам ее функционирования, в том числе в режиме «Конфигуратор». Основная цель пособия – помочь студентам освоить приемы разработки и модификации прикладных решений, изучить основные элементы встроенного языка программирования в системе семейства «1С:Предприятие».

Предназначено для студентов специальности «Математическое обеспечение и администрирование информационных систем», а также для студентов и аспирантов других экономических и технических специальностей.

Подготовлено на кафедре информатики и вычислительной математики.

УДК 519.682.7
ББК 32.973

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 ОБЩИЕ СВЕДЕНИЯ О СИСТЕМЕ	4
1.1 История создания	5
1.2 Функционирование системы.....	9
1.3 Основные понятия системы	10
1.3.1 Конфигурация	10
1.3.2 Объект конфигурации	12
1.3.3 Командный интерфейс	39
1.3.4 Форма	40
1.3.5 Модуль	40
1.3.6 Макет.....	41
1.4 Варианты работы системы.....	42
1.5 Технологические средства разработки	43
1.6 Контрольные вопросы	44
1.7 Практические задания	46
2 ВСТРОЕННЫЙ ЯЗЫК.....	47
2.1 Назначение и краткая характеристика	47
2.2 Программные модули	47
2.3 Формат операторов и комментарии.....	52
2.4 Имена переменных, процедур и функций	53
2.5 Язык написания программных модулей	53
2.5.1 Специальные символы	53
2.5.2 Типы данных	55
2.5.3 Оператор присваивания и операции	57
2.5.4 Операторы и синтаксические конструкции	60
2. 6 Контрольные вопросы	75
2.7 Практические задания	76
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	78

ВВЕДЕНИЕ

Автоматизация учетных и управленческих процессов на предприятии является одной из ключевых задач, стоящих перед руководством каждого предприятия. Для решения этих трудоемких задач в настоящее время существует несколько программных продуктов. Самым популярным из них является система «1С:Предприятие», имеющая достаточно гибкий пользовательский интерфейс и открытый код.

Программирование в среде 1С имеет некоторые особенности, связанные с предметной областью. Их изучение позволяет студентам получить навыки работы в области автоматизации производственных и управленческих процессов на предприятии, понять принципы управления предприятиями, создания адекватных экономических моделей, построения сложных структур данных, реализации эффективных алгоритмов бизнес-логики, создания эргономичного интерфейса.

В данных методических указаниях приведены описание общей концепции системы «1С:Предприятие», приемы работы с объектами системы, способы создания форм, командного интерфейса.

Для освоения материала студенты должны проделать рассмотренные в разделах примеры в информационной базе, созданной на платформе «1С:Предприятие», и добиться понимания логики их выполнения. Материал построен с последовательным усложнением и поэтому, переходить к следующему разделу, не усвоив материал предыдущего, не имеет смысла. В каждом разделе приведен список вопросов, позволяющих оценить, насколько студент усвоил изложенный материал. Для формирования практических умений и навыков в конце каждого раздела приводятся практические задания для самостоятельного выполнения.

1 ОБЩИЕ СВЕДЕНИЯ О СИСТЕМЕ

«1С:Предприятие» является универсальной системой автоматизации деятельности предприятия. Система включает в себя набор инструментов, механизмов и сервисов, обеспечивающих все необходимое для разработки прикладных решений, их работы, администрирования и обслуживания. Благодаря своей универсальности система «1С:Предприятие» может быть использована для автоматизации самых разных участков экономической деятельности предприятия: учета товарных и материальных средств, взаиморасчетов с контрагентами и др.

Технологическая платформа «1С:Предприятие» представляет собой программную оболочку над базой данных. В версии системы 7.7 используются базы на основе DBF-файлов, начиная с версии 8.0 стало возможным использование собственного формата 1CD или СУБД Microsoft SQL Server, начиная с версии 8.1 хранение данных возможно в PostgreSQL и IBM DB2, а начиная с версии 8.2 появилась возможность использования Oracle. Платформа имеет свой внутренний язык программирования, обеспечивающий, помимо доступа к данным, возможность взаимодействия с другими программами посредством OLE и DDE, в версиях системы 7.7, 8.0 и 8.1 - с помощью COM-соединения.

Клиентская часть платформы функционирует в среде Microsoft Windows, начиная с версии системы 8.3, стала возможной работа в среде Linux и Mac OS X. Существуют специальные версии 1С для ноутбуков и PDA, а также программное обеспечение для создания веб-приложений, взаимодействующих с базой данных «1С:Предприятие».

Для защиты продуктов «1С:Предприятие» от несанкционированного использования компания 1С использует аппаратные ключи HASP.

В базовых конфигурациях версии 8 появилась защита через электронные ключи, которые необходимо активировать после покупки программы.

1.1 История создания

Несколько версий 1С:Бухгалтерия было выпущено для платформы DOS — это версии 3.0, 4.0, затем 5.0. Именно в версиях 5.0 и 2.0 ПРОФ был воплощен принцип работы «от документа», а также появился встроенный язык формирования бухгалтерских проводок и печатных форм.

Первой версией, выпущенной для платформы Windows 3.1, была 1С:Бухгалтерия для Windows 1.0 (позже 1.1). Следующая версия (1С:Бухгалтерия 6.0) была выпущена в 1995 году и во многом являлась повторением DOS-версии, но на новой основе. В дальнейшем под этим названием - «1С:Бухгалтерия» - стали производиться конфигурации для платформ 7.7 и 8.x.

Весной 1999 года версии «1С:Предприятие 7.0» и «1С:Предприятие 7.5» были эволюционно заменены на «1С:Предприятие 7.7».

Версия системы 7.7 состоит из программной оболочки, или «движка», который работает с одной или несколькими базами данных, определяемыми конфигурацией.

Одной из особенностей, позволяющей существенно расширить функциональность языка является использование внешних компонент. Внешняя компонента представляет собой динамически подключаемую библиотеку, которая реализует определенную функциональность. Таким образом, в 1С стало возможным использовать возможности других языков и платформ (например, таких, как .NET Framework). Разработчиками 1С были созданы различные компоненты, позволившие значительно дополнить функциональность такими средствами, как работа с FTP, почтой, HTTP-запросами. Одним из значимых проектов явилась разработка компоненты 1СРР, позволяющей работать с базой данных при помощи OLE DB- и ODBC-интерфейсов (так называемые прямые запросы), что позволило в некоторых случаях сократить время выполнения запросов к базе данных в несколько раз.

Примерами стандартных «компонент» являются:

- «Бухгалтерский учёт»;
- «Оперативный учёт»;
- «Расчёт»;

- «Управление распределенными ИБ»;
- «Web-расширение 2.0».

Система может работать в нескольких режимах:

- 1С:Предприятие - основной режим работы пользователя, предназначенный для ввода данных, получения отчетов;
- Конфигуратор - режим администрирования и изменения конфигурации;
- Отладчик - режим отладки и замера производительности конфигурации;
- Монитор - режим просмотра активных пользователей и журнала регистрации событий.

Конфигурация содержит параметры базы данных и пользовательского интерфейса. Файл конфигурации 1Сv7.MD расположен в каталоге базы данных.

Язык программирования версии 7.7 существенно отличается от языка версии 6.0 и от языка DOS-версий. Впервые в версии 7.7 появилось понятие объектов данных.

К информационной базе системы «1С:Предприятие 7.7» возможен доступ с помощью веб-интерфейса на основе расширения v7script.dll, благодаря которому можно организовать доступ к базам 1С через веб-порталы, а также возможен доступ через COM-соединение.

14 августа 2002 года была выпущена система «1С:Предприятие 8.0 ознакомительная версия». 31 июля 2003 года выпущено первое тиражное решение «1С:Предприятие 8.0. Управление торговлей», что являлось одновременно и выпуском предназначенной для широкого использования версии технологической платформы «1С:Предприятия 8.0».

Все «компоненты» были включены в базовую поставку (платформа теперь не делится на «компоненты» «бухгалтерский учет», «расчет», «оперативный учет», «управление распределенными базами данных», как это было с версией 7.7). Серверная «компонента» 1С версии 8.1 может работать в операционной системе Linux и использовать PostgreSQL.

По сравнению с версией 7.7, существенно изменился язык программирования для этой платформы. Он стал не совместим с языками версий 6.0 и 7.x. Встроенный язык

стал ближе к концепции объектно-ориентированного программирования, в нем появились коллекции и контейнеры, как структура и соответствие, существенно пересмотрена работа с коллекциями (элемент коллекции теперь представляется отдельным объектом), интерфейсные объекты (к примеру, табличные поля) стали четко разграничиваться от отображаемых в них данных (таблиц значений и табличных частей), в программировании интерфейса реализована полнофункциональная концепция обработки событий, язык запросов стал походить на язык SQL с некоторыми дополнениями.

В версиях системы 8.x существует два основных режима работы: режим предприятия и режим конфигуратора (в этом режиме также имеется режим отладки). Режим монитора исключен ввиду того, что монитор встроен в режимы и предприятия, и конфигуратора.

Платформа «1С:Предприятие 8» допускает конфигурирование, одновременное использование одной базы несколькими пользователями при наличии необходимого количества дополнительных лицензий.

Платформа «1С:Предприятие 8» при использовании со специализированными «базовыми» конфигурациями обеспечивает работу в режиме с ограничением возможностей: не допускает конфигурирования, не поддерживает режим внешнего соединения, не имеет многопользовательского режима. Для работы в «базовом» режиме используется облегченный вариант лицензирования, не требуются аппаратные ключи защиты.

Кроме того, выпускается учебная версия платформы «1С:Предприятие 8».

«1С:Предприятие 8» может использоваться в файловом режиме, с хранением базы в виде файла, или в клиент-серверном режиме с хранением баз на сервере SQL. Клиент-серверный режим требует дополнительного приобретения лицензии на «сервер 1С:Предприятия 8».

Для создания веб-порталов используется «веб-расширение», платформа ASP.NET и рекомендуемый внутренний язык C#, так как большая часть объектов доступа к данным уже предоставляется именно на нем.

Главной отличительной особенностью «1С:Предприятие 8.2» является режим «управляемое приложение», при котором пользовательский интерфейс описывается декларативно, а его вид зависит от типа клиентского программного обеспечения:

- «толстый» клиент (клиентское приложение версий 8.0 и 8.1);
- «тонкий» клиент;
- веб-клиент (включает клиентскую и серверную части: поддерживаются веб-браузеры Internet Explorer и Mozilla Firefox, а веб-сервер может работать на Apache или IIS). Клиент-серверное взаимодействие реализовано с использованием технологий AJAX (DHTML), XMLHttpRequest и JavaScript.

Также «1С:Предприятие 8.2» отличается более «строгой» реализацией клиент-серверной архитектуры, т.к. требуется перенос выполнения всей бизнес-логики в серверный код, реализована динамическая балансировка нагрузки в кластера серверов, добавлена поддержка СУБД Oracle.

В новой версии платформы реализована отказоустойчивость кластера серверов «1С:Предприятия» - основной «компоненты», обеспечивающей взаимодействие между пользователем и СУБД.

26 мая 2010 года выпущено обновление платформы 8.2.11. В этой версии проведена оптимизация внутренних механизмов и реализованы некоторые новые возможности:

- работа с криптозащитой;
- экспорт документов в форматы Microsoft Office 2007;
- отладка веб-клиента;
- поддержка браузеров Google Chrome и Safari.

В качестве значимых изменений «1С:Предприятие 8.3» можно отметить:

- предоставление пользователям нативных 64-битных клиентов под Linux и MacOS;
- 64-битный клиент и Конфигуратор для Windows;
- мобильная платформа для iOS, Android и Windows Phone;
- переработка механизма расположения элементов в формах;
- улучшение механизмов хранилища конфигурации и сравнения объектов;

- механизм автоматизированного тестирования интерфейса;
- выгрузка конфигурации в файлы текстового формата.

Вместе с продолжением развития платформы 8.3 фирма 1С представила разработчикам бета-версию нового Конфигуратора. Пакет Enterprise Development Tools базируется на основе Eclipse и является дальнейшим развитием средств разработки платформы V8.

30 декабря 2015 года была опубликована ознакомительная версия платформы 8.4.1, в которой существенно переработан механизм кластера серверов.

1.2 Функционирование системы

Функционирование системы делится на два процесса:

- разработка - описание модели предметной области средствами системы;
- исполнение - обработка данных предметной области.

На этапе *разработки* производится:

- формирование структуры обрабатываемой информации;
- создание форм для ввода исходных данных, просмотра различных списков данных;
- организация хранения введенной и итоговой информации;
- написание отчетов и обработок;
- формирование командных интерфейсов для различных групп пользователей;
- формирование списка пользователей;
- назначение пользователям определенных прав.

Результатом разработки является программный продукт (конфигурация), который представляет собой модель предметной области.

В режиме «Конфигуратор» можно создавать новые конфигурации, редактировать имеющиеся, а также производить сравнение и объединение нескольких конфигураций.

На этапе разработки система оперирует такими универсальными понятиями (объектами), как документ, журнал документов, справочник, реквизит, форма, регистр и другие. Совокупность этих понятий и определяет концепцию системы. В свою очередь, процесс конфигурирования распадается на несколько составляющих:

- «визуальное» конфигурирование - создание структуры конфигурации, форм диалогов и выходных документов, механизма работы пользователей с данными (интерфейс), указание прав доступа различных групп пользователей к различной информации;
- написание программ на встроенном языке «1С:Предприятие» для обработки входных и выходных данных.

На уровне системы определены сами понятия объектов и стандартные операции по их обработке.

В процессе *исполнения* система уже оперирует конкретными понятиями, описанными на этапе конфигурирования (справочниками товаров и организаций, счетами, накладными и т. д.).

При работе пользователя в режиме «1С:Предприятие» обработка информации выполняется как штатными средствами системы, так и с использованием алгоритмов, созданных на этапе конфигурирования.

1.3 Основные понятия системы

В этом разделе рассматриваются основные понятия, которыми оперирует система «1С:Предприятие».

1.3.1 Конфигурация

Основу концепции системы «1С:Предприятие» составляет понятие «конфигурация».

Система «1С:Предприятие» представляет собой совокупность механизмов, предназначенных для манипулирования различными типами объектов предметной области. Набор объектов, структуры информационных массивов, алгоритмы обработки информации, соответствующих поставленной задаче, определяет

конкретная конфигурация. Вместе с конфигурацией система «1С:Предприятие» выступает в качестве уже готового к использованию программного продукта, ориентированного на определенные типы предприятий и классы решаемых задач.

Конфигурация создается и сопровождается (поддерживается) штатными средствами системы. Конфигурация обычно поставляется в качестве типовой для конкретной области применения, но может быть изменена, дополнена пользователем системы, а также разработана заново.

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

- подсистемы;
- структуры учетных данных, их форм ввода, выбора, печати;
- механизмы учета итоговых данных и их движений;
- отчеты и обработки;
- командный интерфейс;
- наборы ролей (прав доступа);
- наборы общих процедур и функций (модуль приложения, модуль

управляемого приложения, модуль внешнего соединения, модуль сеанса, общие модули), макетов табличных документов и др.;

- вспомогательные объекты;
- функциональные опции и их параметры;
- хранилища настроек;
- средства работы с Web (Web-сервисы, WS-ссылки);
- вспомогательная информация (картинки, шаблоны, стили).

Фактически структура конфигурации является моделью предметной области.

Роли в системе «1С:Предприятие» определяют полномочия пользователей на работу с информацией, которая обрабатывается в системе. Совокупность предоставляемых пользователю полномочий определяется, как правило, кругом его обязанностей.

Операция назначения ролей пользователю решает две основные задачи:

- ограничение круга пользователей конфиденциальной информации;

➤ запрет выполнения определенных операций (в первую очередь операций удаления и корректировки данных).

Система учитывает назначение прав на объекты при построении командного интерфейса. Если, например, пользователю запрещен просмотр какого-либо справочника, то команда открытия формы этого справочника будет удалена из командного интерфейса автоматически.

1.3.2 Объект конфигурации

Под объектом конфигурации в системе «1С:Предприятие» понимается формальное описание группы понятий (предметной области, средств взаимодействия пользователя с системой) со сходными характеристиками и одинаковым предназначением.

Примерами объектов конфигурации являются Справочники, Документы, Отчеты.

Как правило объекты конфигурации являются компьютерными аналогами реальных объектов предметной области, хотя могут использоваться и для организации списков, не имеющих явных физических аналогов.

Объект конфигурации описывает не конкретное значение, а только его вид. Например, справочник «Физические лица» описывает не конкретного человека, а содержит перечень реквизитов (набор видов характеристик о физическом лице), а также формы для ввода их значений, формы просмотра списков и макеты для печати информации.

Реализованный в системе «1С:Предприятие» компьютерный аналог конкретного понятия предметной области будем называть *объектом конфигурации*.

1.3.2.1 Свойства объекта конфигурации

Каждый объект конфигурации обладает уникальным набором свойств. Этот набор описан на уровне системы и не может быть изменен в процессе настройки конфигурации задачи. Набор свойств объекта конфигурации определяется в основном его назначением в системе «1С:Предприятие».

Некоторые свойства из всего набора свойств, присущих объекту конфигурации, доступны для редактирования. Характер изменений и их пределы также задаются на уровне системы. Специалист, осуществляющий конфигурирование системы, целенаправленным изменением свойств объекта конфигурации может добиться требуемого поведения объекта при работе системы. Однако такие изменения не затрагивают сущности объекта и не позволяют добиться от него действий, не свойственных объектам данного типа.

Практически все объекты конфигурации имеют следующие свойства, располагающиеся в категории свойств «Основные»:

- Имя – имя объекта конфигурации.

Имя должно состоять из одного слова, начинаться с буквы и не содержать специальных символов, кроме «_». По имени объекта производится доступ и управление объектом конфигурации средствами встроенного языка. Имена объектов конфигурации не должны совпадать с зарезервированными словами языка запросов. Длина имени объекта конфигурации не может превышать 80 символов. При создании нового объекта конфигурации ему автоматически присваивается условное имя, состоящее из слова, определяемого по виду объекта, и цифры (например, при создании документа – Документ1 и т. д.). Условное имя можно изменить в процессе редактирования свойств объекта конфигурации, при этом система отслеживает уникальность имен.

- Синоним.

Если конфигурация создается для использования на разных языках, то следует указать синонимы на используемых языках. При работе с системой «1С:Предприятие» синоним будет выдаваться в различных списках выбора, заголовках окон, текстах надписей, при формировании интерфейсов с учетом текущего языка. Синоним не имеет ограничений на использование символов. Если синоним не задан, выбирается имя.

Примечание. Имя или синоним, отображаемые пользователю, также называются *представлением объекта конфигурации*.

➤ Комментарий – произвольная строка символов, как правило, расшифровывающая и поясняющая имя объекта.

Для многих объектов конфигурации реализована возможность задавать различные представления объекта. Это сделано для того, чтобы разработчик имел возможность задать представление стандартных команд, их подсказок, а также заголовков форм.

Как правило, заполнение свойств, связанных с представлением объектов и списков, необходимо только в тех случаях, когда требуется несколько уточнить информацию, отображаемую для объекта по умолчанию.

Для того чтобы на уровне конфигурации переопределять некоторые интерфейсные свойства (такие как синоним, проверка заполнения и т. д.) стандартных реквизитов (например, Код, Наименование, Родитель) и стандартных табличных частей (например, ВидыСубконто, БазовыеВидыРасчета) прикладных объектов, существует возможность настройки этих свойств.

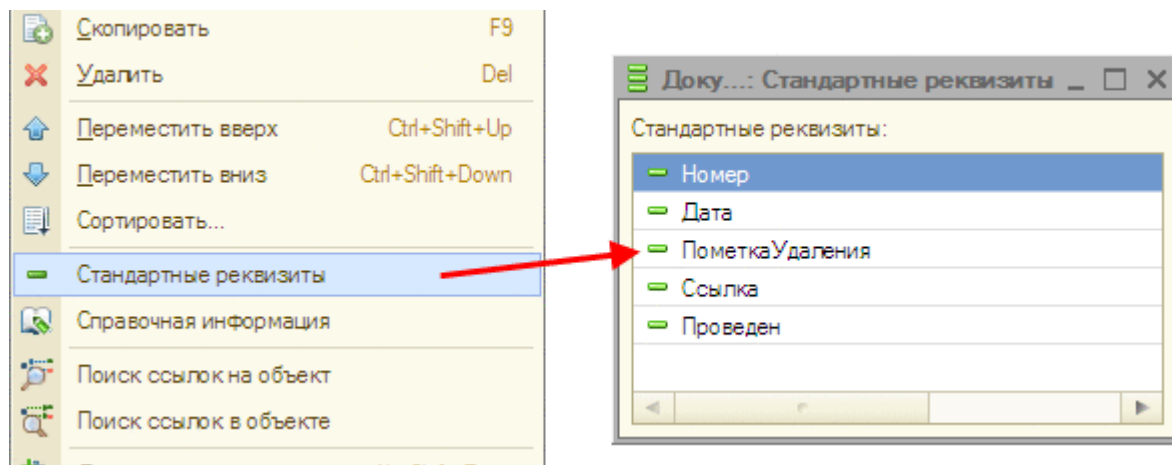


Рис. 1. Стандартные реквизиты

Для этого в палитре свойств ряда объектов существуют команды, открывающие списки стандартных реквизитов и стандартных табличных частей (рисунок 1). Такие команды доступны для тех объектов, у которых имеются стандартные реквизиты и стандартные табличные части.

С помощью палитры свойств можно переопределять некоторые свойства стандартных реквизитов таким образом, чтобы они более полно соответствовали требованиям решаемой прикладной задачи.

В случае если свойства стандартного реквизита (или стандартной табличной части) не заданы, будут использованы свойства стандартных реквизитов по умолчанию.

По набору свойств стандартный реквизит практически ничем не отличается от любого другого реквизита, за исключением того, что нельзя изменить:

- имя стандартного реквизита,
- тип стандартного реквизита,
- имя стандартной табличной части,
- индексирование стандартного реквизита.

1.3.2.2 Нумерация объектов

В системе «1С:Предприятие» существуют объекты, которые необходимо нумеровать в автоматическом или ручном режиме. Нумерация доступна для следующих объектов:

- Справочники;
- Документы;
- Планы видов характеристик;
- Бизнес-процессы;
- Задачи;
- Планы обмена.

В зависимости от объекта, доступны различные виды и режимы нумерации. Нумерация может выполняться для номера (документы, бизнес-процессы, задачи) и кода (все остальные объекты).

Код/номер объекта конфигурации может быть числовым или строковым.

Если код/номер задается как строка, то доступными становятся следующие дополнительные возможности:

- код/номер может состоять не только из цифр, а из любых доступных символов;
- возможность определить, какую длину будет иметь код/номер. Если это свойство имеет значение «Фиксированная», то длина кода/номера всегда будет равна

значению, указанному в качестве свойства «Длина кода» или «Длина номера», при этом строка будет дополняться пробелами.

Если для свойства выбрано значение «Переменная», то длина кода/номера не может быть больше значения, указанного в свойствах «Длина кода» или «Длина номера», но фактическая длина будет определяться количеством символов, участвующих в формировании номера.

- возможность задавать различные префиксы для кода/номера.

Для корректной работы автоматической нумерации строковых кодов/номеров, рекомендуется или не указывать номера вручную или задавать лидирующие нули так, чтобы любой код/номер всегда был максимально возможной длины (с учетом ведущих нулей). Кроме того, указание ведущих нулей обеспечит корректную сортировку кодов/номеров в случае, если тип кода/номера является строкой.

Свойство «Признак автонумерации» определяет, будет ли система автоматически назначать код/номер для нового объекта при его записи.

Свойство «Контроль уникальности» определяет, будет ли система контролировать уникальность кода/номера для объекта.

Для документов имеется возможность указывать нумератор, что, например, позволяет обеспечить общее пространство номеров для нескольких разных видов документов.

Для документов и бизнес-процессов имеется возможность обеспечивать периодичность номеров. Это означает, что нумерация начнется с начала, каждый раз, когда начинается новый период. Периодичность настраивается с помощью свойства «Периодичность».

Контроль уникальности при периодической нумерации будет выполняться в рамках установленного периода (за год, квартал, месяц или день). Если задана непериодическая нумерация, то уникальность будет контролироваться во всех существующих объектах данного вида.

Принадлежность к тому или иному периоду определяется по дате документа или бизнес-процесса, а не по текущей дате.

При автоматизации различных задач может потребоваться отдельная

нумерация объектов одного вида в зависимости от некоторых условий. Такая задача может быть решена с помощью префиксов кодов/номеров. Префикс кода/номера – это последовательность символов, предваряющих код/номер, в рамках которых происходит нумерация объектов. Использование префиксов нумерации имеет несколько особенностей:

- префиксы используются только со строковым типом кода/номера;
- длина кода/номера, при использовании префикса, может оказаться больше, чем без использования префикса (на размер префикса и разделителя);
- для автоматического изменения префиксов в зависимости от значений реквизитов объектов необходимо выполнять дополнительные действия на встроенном языке.

1.3.2.3 Основные виды объектов конфигурации

Все объекты конфигурации, которые существуют в системе «1С:Предприятие», образуют несколько основных видов. Каждый вид объектов конфигурации представляет собой «строительные элементы», из которых будет создаваться конфигурация.

Все объекты конфигурации объединяются в виды в дереве конфигурации. Названия видов пользователь видит на первом уровне дерева конфигурации, когда открывает окно «Конфигурация» в конфигураторе (рисунок 2).

Несмотря на отсутствие формального определения, названия видов объектов конфигурации широко используются при работе с системой «1С:Предприятие».

Например, специалист, осуществляющий конфигурирование системы «1С:Предприятие», видит свою цель в разработке необходимого набора справочников, документов, отчетов, журналов, которые будут реализовывать требуемую систему учета. Конечный пользователь системы «1С:Предприятие» – руководитель, бухгалтер, менеджер, кладовщик – также оперирует конкретными справочниками, документами и т. д. для решения стоящих перед ним задач. Общение между двумя этими категориями пользователей также будет происходить в понятиях видов объектов конфигурации.

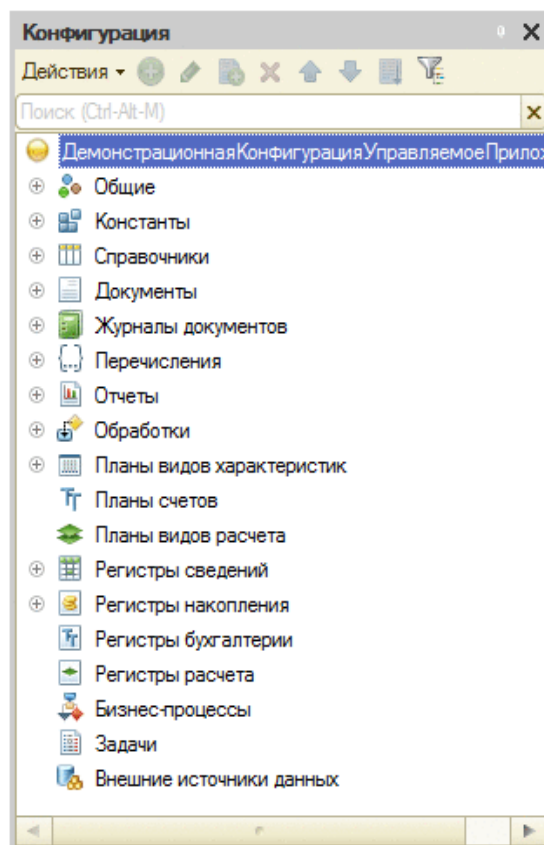


Рис. 2. Дерево метаданных

Объект данных какого-либо вида является уже конкретным документом, отчетом, журналом, константой и так далее. Как правило, каждый объект используется для работы со вполне определенной информацией предметной области.

Рассмотрим характеристики основных видов объектов конфигурации системы «1С:Предприятие».

Константы

Для работы с постоянной и условно постоянной информацией в системе используются объекты типа «Константа». Информация, хранящаяся в константах, редко изменяется, но, как правило, часто используется в работе. Например, в константах может храниться наименование предприятия, его ИНН, фамилии директора и главного бухгалтера и другая подобная информация.

В системе может быть описано неограниченное количество констант.

Постоянная или условно-постоянная информация либо совсем не изменяется в процессе деятельности предприятия, либо изменяется достаточно редко.

Основная причина использования констант заключается в том, что в них один раз заносится какая-либо информация, которая затем может использоваться при формировании документов, в расчетах, при построении отчетных форм. Значение константы время от времени может редактироваться.

Пример. Наиболее часто на документах предприятия встречается подпись директора и главного бухгалтера. Помимо самой подписи в документах требуется и ее расшифровка – фамилия подписавшего документ. С одной стороны, можно ввести фамилии директора и главного бухгалтера непосредственно в формы документов. Однако, если какая-либо из фамилий изменится, то придется вновь редактировать все бланки документов, что потребует значительного количества времени. В связи с этим, гораздо удобнее создать в конфигурации две константы – для хранения фамилий директора и главного бухгалтера, ввести эти фамилии в константы один раз, а в многочисленных бланках документов использовать имена констант для получения их значений. При смене директора или главного бухгалтера достаточно внести изменения только в значения самих констант и все изменения автоматически будут отражены в тех местах, где эти константы используются.

Примечание. При изменении значения константы прежнее значение теряется. Для получения прежних значений данных вместо константы нужно использовать регистр сведений без измерений.

Для работы с константами предназначена ветвь «Константы» дерева конфигурации.

Свойства константы редактируются в палитре свойств. Необходимо обратить внимание, что константа является типизированным объектом конфигурации. Тип константы может быть любым из стандартных типов конфигурации (Дата, Число, СправочникСсылка, ДокументСсылка и т. д.) или составным (состоящим из набора различных типов). В зависимости от выбранного типа палитра свойств может содержать дополнительные свойства, уточняющие выбранный тип.

Формы для ввода констант создаются в ветви «Общие формы».

Справочники

Для работы с постоянной и условно постоянной информацией с некоторым множеством значений в системе используются объекты типа «Справочник».

Обычно справочниками являются списки материалов, товаров, организаций, валют, сотрудников и др.

Пример. При заполнении бланка какого-либо документа часто требуется указывать информацию, выбирая значение из заранее заданного списка. Например, при поступлении на работу необходимо заполнить анкету. При заполнении графы «Место рождения» необходимо указать населенный пункт. Несмотря на то, что общее количество населенных пунктов велико, список всех населенных пунктов является ограниченным. Фактически место рождения можно указать, выбрав из подобного списка нужный населенный пункт. Такой список и представляет собой справочник.

Следовательно, справочник можно использовать как список возможных значений того или иного реквизита документа (в широком смысле слова «документ»).

Также справочники используются в тех случаях, когда необходимо исключить неоднозначный ввод информации.

На этапе конфигурирования можно описать, какими свойствами обладает каждый конкретный справочник. К настраиваемым свойствам относятся, например, длина и тип кода, количество уровней иерархии, поддержка уникальности кодов, набор реквизитов справочника.

Как правило, справочники имеют стандартные реквизиты код и наименование, при этом код может иметь тип Число или Строка.

Система «1С:Предприятие» предоставляет широкие возможности по работе с кодами элементов справочника: автоматическое присвоение кодов, автоматический контроль уникальности кода и другие.

Помимо кода и наименования, механизм работы со справочниками позволяет создавать набор реквизитов для хранения любой дополнительной информации об элементе справочника, а также табличные части. В табличных частях хранится

однотипная информация, число которой может быть переменным, например, описание комплектующих изделия, состав семьи сотрудника, телефоны организации и т. д.

Для описания соподчиненных сущностей можно использовать подчиненные справочники. В этом случае в подчиненном справочнике каждый элемент «принадлежит» к определенному элементу справочника-владельца.

В конкретной конфигурации создается необходимое количество справочников для хранения данных об объектах, используемых при автоматизации данной предметной области. Например, это могут быть справочники Организации, Товары, Сотрудники и т. д.

При этом в конфигурации создается структура справочника, а собственно его содержимое – элементы справочника – вводится пользователем при работе с программой. В процессе конфигурирования описывается структура информации, которая будет храниться в справочнике, разрабатывается экранное и, если необходимо, печатное представление справочника, задаются различные особенности его «поведения».

Справочник в системе «1С:Предприятие» может быть иерархическим. Существуют два вида иерархии: иерархия групп и элементов и иерархия элементов. В первом случае все данные справочника будут разделяться на два вида: «просто» элементы справочника и группы справочника. Группы – это логическое объединение элементов справочника. Примером иерархического справочника может служить справочник товаров, где группами являются виды товаров («Сантехника», «Бытовая химия» и т. д.), а элементами – конкретные товары («Смеситель», «Зеркало», «Стиральный порошок»).

Использование иерархических справочников позволяет организовать ввод информации в справочник с нужной степенью детализации. Элементы и группы элементов в иерархическом справочнике можно переносить из одной группы в другую.

Отличительной особенностью справочников с видом иерархии «Иерархия элементов группы» является функциональность всех элементов. Примером

справочника данного вида может служить справочник подразделений. Каждое подразделение описывается одинаковым набором реквизитов и при этом логически может содержать другое подразделение или входить в него.

Для иерархических справочников конфигуратор позволяет установить ограничение числа уровней справочников, или допускается неограниченное число уровней вложенности.

Для работы с информацией, хранящейся в справочнике, можно создать экранные формы. Могут быть созданы отдельные формы для просмотра списка элементов справочника, для редактирования элемента справочника, формы для выбора из справочника нужного элемента.

Конфигуратор позволяет создать несколько форм одного типа, например, формы для выбора из справочника нужного элемента, и использовать разные формы в разных случаях.

Рекомендуется создавать разные формы для отображения списка элементов и для выбора элементов справочника.

Для работы со справочниками предназначена ветвь «Справочники» дерева конфигурации. С ее помощью можно создать и задать настройки нового справочника, изменить настройки уже существующего справочника, удалить справочник.

Перечисления

Перечисления используются в системе «1С:Предприятие» для описания постоянных наборов значений, не изменяемых в процессе работы 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), статус клиента (постоянный, разовый) и т. д.

Перечисление представляет собой служебный тип данных, который не используется самостоятельно, а применяется в основном в совокупности с другими

типами данных. Определить перечисление можно как список возможных значений реквизита.

Перечисления используются при вводе значений реквизитов документов, справочников, при вводе значений констант, в тех случаях, когда необходимо исключить неоднозначный ввод информации.

Перечисление не имеет вложенности – все его значения находятся на одном уровне.

Для работы с перечислениями предназначена ветвь «Перечисления» дерева конфигурации.

Редактирование перечисления заключается в создании списка его значений. Для редактирования перечисления используется окно редактирования объекта Перечисление.

Документы

Документ – одно из основных понятий системы «1С:Предприятие». При помощи документов организуется ввод в систему информации о совершаемых хозяйственных операциях.

В большинстве своем документы, которые создаются в процессе настройки конфигурации задачи, являются электронными аналогами стандартных бумажных документов, однако использование этого типа данных может выходить далеко за рамки простой фиксации информации о хозяйственных операциях.

Например, в конфигурации, предназначенной для учета торговых операций, могут быть такие документы, как счет, приходная накладная, расходная накладная и прочие. При помощи документов отражаются и платежи с расчетного счета, и операции по кассе, и движения по складу, и прочие подобные события, происходящие на предприятии.

В конфигурации описывается только структура документа, конкретные экземпляры документов вводятся при работе с программой пользователем.

Все документы характеризуются номером, датой и временем. При настройке для документа также задается длина номера документа, условия поддержки уникальности номеров и т.д.

Для определения положения документа на оси времени используется реквизит документа «Дата», который содержит время с точностью до секунды, что позволяет контролировать последовательность записи документов. Однако при большом объеме создаваемых документов вероятна ситуация, когда несколько документов будут иметь одинаковое значение даты (т. е. будут созданы в течение одной секунды). В такой ситуации становится невозможным определить последовательность созданных документов.

Для обработки подобных ситуаций существует понятие *момент времени*. Момент времени представляет собой совокупность даты, времени и ссылки на объект базы данных. Он позволяет однозначно идентифицировать любой объект ссылочного типа базы данных на оси событий, но имеет смысл в основном только для документов. Кроме того, момент времени позволяет идентифицировать и необъектные данные, например, записи регистров, подчиненных регистратору.

Понятие момента времени реализовано во встроенном языке при помощи универсального объекта МоментВремени.

Для нескольких документов, имеющих одинаковую дату и время, последовательность их на оси событий определяется системой исходя из ссылок на эти документы. Она может не совпадать с последовательностью создания документов, и является недоступной для изменения пользователем, то есть нельзя каким-либо образом повлиять на последовательность документов внутри одной секунды или «вычислить», что один документ создан раньше, а другой – позже.

Помимо даты, времени и номера документа можно создать набор реквизитов, позволяющих хранить дополнительную информацию.

Если объект предметной области, которой соответствует документ, имеет не только такие «простые» свойства, как, например, дату, номер и итоговую сумму, но и составные (списочные) свойства, документу может быть создан набор табличных частей. Каждый вид документа может иметь неограниченное количество реквизитов и табличных частей. Несколько табличных частей требуются в тех случаях, когда одним документом необходимо зарегистрировать разные по сути, но связанные

события, например: отразить поступление товара на склад и зарегистрировать понесенные дополнительные затраты – оплату транспорта, грузчиков и др.

В процессе конфигурирования настраивается произвольное количество видов документов. Примерами видов документов являются: «Платежное поручение», «Счет», «Приходная накладная», «Расходная накладная», «Накладная на внутреннее перемещение», «Приходный кассовый ордер» и другие. Каждый вид документа предназначен для отражения своего типа событий. Принадлежность документа к тому или иному виду определяет его структуру и свойства, которые описываются в конфигурации.

Для документа создаются формы ввода – экранные аналоги реальных документов. Если в других формах используются данные из документов, то для включения этой информации разрабатываются формы для выбора. Для просмотра списка документов одного вида создаются формы списков. Число форм неограниченно.

Документ может вводиться как на основании другого документа, так и на основании объектов другого вида (элементов справочника, плана видов характеристик, плана счетов и плана видов расчета). Также документ может являться основанием для ввода не только другого документа, но и объектов другого вида.

Каждый документ также может иметь неограниченное число печатных форм.

Документы играют центральную роль для основных механизмов, реализуемых системой. Все документы образуют единую хронологическую последовательность. Фактически она отражает реальную последовательность событий. Внутри даты последовательность документов определяется их временем, при этом время документа является не столько средством отражения реального (астрономического) времени ввода документа, сколько средством, позволяющим четко упорядочить документы внутри одной даты. Данные, вводимые в документ (в реквизиты и табличные части документа), обычно содержат информацию о происшедшем событии: например, в накладной – информацию о том, с какого склада, каких товаров и сколько отгружено, какие дополнительные затраты произведены при приобретении товаров.

При создании документа можно указать перечень журналов, в которых будет осуществляться работа с документами этого вида. Для документов разных видов можно указывать один журнал, что позволяет произвольным образом группировать документы в журналах. Назначенные документу журналы можно менять.

Информация, отражающая хозяйственную деятельность предприятия, хранится в регистрах. Документы могут изменять состояние регистров. Этот процесс называется *проведением*. Данный механизм является рекомендуемым механизмом изменения состояния регистров.

Документ может быть «проведен» или «непроведен». Если документ не является «проведенным», это значит, что событие, которое он отражает, не влияет на состояние учета, который ведется в данной конфигурации. Если документ проводится, то он изменяет состояние тех или иных учитываемых данных. При проведении документ отражает зафиксированное им событие в механизмах, реализуемых различными регистрами.

Проведение может выполняться в оперативном или неоперативном режиме (настраивается с помощью свойства «Оперативное проведение»).

Механизм оперативного проведения предназначен для того, чтобы разделить случаи, когда документ проводится в реальном времени, и случаи, когда проведение документа отражает уже свершившийся факт.

Проведение в реальном времени необходимо тогда, когда ввод и проведение документа не просто фиксируют в системе произошедшее событие, а участвуют в его формировании, помогая пользователю правильно ввести информацию. Очевидно, что это имеет смысл только в тот момент, когда данное событие происходит в реальной жизни.

Пример. Классическим примером является ввод и проведение документа, отражающего продажу товаров со склада. При вводе такого документа в задачу оператора входит не только правильный ввод списка товаров, которые приобретает покупатель, но и выполнение различных проверок. Прежде всего, необходимо проверить, что запрашиваемый товар имеется на указанном складе. При этом очень важно, чтобы проверка учитывала тот факт, что одновременно с этим оператором

работают и другие операторы, которые могут одновременно выписывать те же самые товары. Соответственно, задачей системы является не допустить продажу одного и того же товара двум покупателям. Кроме того, может потребоваться и проверка доступного покупателю размера кредита или наличие факта оплаты счета, а также другие самые разнообразные проверки.

Если же в момент ввода документа известно, что такое событие уже произошло в жизни предприятия, например, конкретный товар уже отгружен клиенту, необходимость в таких проверках отпадает, и нужно просто отразить в учете произошедшее событие. В этом случае проведение документа только фиксирует событие, а не участвует в его формировании.

Таким образом, задача механизма оперативного проведения заключается в разделении этих двух вариантов проведения и с точки зрения пользователя и с точки зрения алгоритма проведения документа.

Оперативность или неоперативность проведения документа определяется по его дате. Если дата проводимого документа совпадает с текущей датой, то система будет проводить такой документ в оперативном режиме.

Для работы с документами предназначена ветвь «Документы» дерева конфигурации. На этой же ветви располагаются служебные объекты конфигурации – нумераторы и последовательности.

Журналы документов

Журналы документов предназначены для просмотра документов разных видов. Каждый вид документа может быть показан в нескольких журналах. Журнал документов не добавляет новые данные в систему, а является средством их группировки.

Пример. В системе может быть создан журнал «Складские документы», в котором будут отображаться все приходные и расходные накладные и накладные на внутреннее перемещение.

Для журнала могут быть определены графы журнала, предназначенные для отображения реквизитов документов разного вида, отнесенных к данному журналу.

Каждый журнал может иметь неограниченное число форм визуального представления и печатных форм.

Если ни одной формы журнала не создано, в режиме «1С:Предприятие» автоматически создается форма журнала по умолчанию.

В системе «1С:Предприятие» журналы документов являются объектами, позволяющими осуществлять работу с документами разных видов. Работая с формами журнала, пользователь может вводить документы, просматривать их, редактировать и удалять.

Пользователь может осуществлять поиск любого документа в журнале по содержимому граф, выполнять поиск документов по их номерам, осуществлять отбор документов по различным признакам.

Для отражения данных документа в журнале необходимо или в журнале, или в документе указать эту принадлежность.

Конфигуратор позволяет создавать любое необходимое число журналов.

Для работы с журналами документов предназначена ветвь «Журналы документов» дерева конфигурации. Она позволяет создать и настроить свойства нового журнала, отредактировать свойства уже существующего журнала, удалить журнал.

Отчеты и обработки

Любая система автоматизации учета только тогда выполняет свои функции, когда она имеет средства обработки накопленной информации и получения сводных данных в удобном для просмотра и анализа виде. Как правило, для решения подобных задач в системе автоматизации учета существует возможность формирования отчетов. Конфигуратор позволяет формировать набор различных отчетов, достаточных для удовлетворения потребностей пользователей системы в достоверной и подробной выходной информации.

Для получения отчетной информации в системе «1С:Предприятие» используются объекты конфигурации, расположенные в ветви «Отчеты» дерева конфигурации.

Алгоритм получения отчета может описываться с использованием встроенного

языка или формироваться системой автоматически, в случае использования системы компоновки данных. Для вывода отчетов может быть использован как текстовый формат, так и специализированный табличный формат отчетов (макеты).

Для выполнения различных действий над информацией в системе «1С:Предприятие» используются объекты конфигурации, расположенные в ветви «Обработки» дерева конфигурации. С помощью обработок можно выполнять удаление из системы устаревших данных, импорт информации из других систем и многое другое.

Обработка может содержать одну или несколько форм, с помощью которых, при необходимости, можно организовать ввод каких-либо параметров, влияющих на ход алгоритма. Вывод результатов выполнения алгоритма на экран и принтер осуществляется с помощью конструктора макетов описания печатных форм.

Основное отличие отчета от обработки заключается в возможности использования схемы компоновки данных.

Для описания отчетов и процедур обработки информации на этапе конфигурирования может быть создано неограниченное число отчетов и обработок. Отчеты и обработки могут иметь несколько форм, предназначенных, например, для ввода параметров формирования отчета или параметров обработки данных.

Внешней обработкой в системе «1С:Предприятие» называется обработка, хранящаяся вне конфигурации, в отдельном файле. Внешняя обработка служит для решения тех же задач, что и объекты конфигурации типа «Отчет» или «Обработка».

Основное назначение внешней обработки (отчета) заключается в возможности реализовывать, поставлять и обновлять некоторые возможности отдельно от конфигурации.

Внешние обработки и отчеты могут храниться в двух разных форматах:

- двоичные файлы, имеющие расширение .erf и .erf . Эти файлы могут использоваться как для редактирования в конфигураторе, так и для использования в режиме «1С:Предприятие». Имеется возможность разработки и отладки внешних обработок в процессе работы системы «1С:Предприятие». В этом случае разработка и отладка обработки (отчета) значительно ускоряются: ее редактирование и

сохранение выполняются в режиме «Конфигуратор», без сохранения конфигурации в целом, а запуск – в режиме «1С:Предприятие». Для выполнения внешняя обработка (отчет) загружается при помощи пункта «Файл – Открыть» и работает так же, как и любая другая обработка (отчет) конфигурации.

➤ XML-файлы в формате, соответствующим файлам, которые используются при выгрузке/загрузке конфигурации в файлы. При использовании этого формата поддерживается только редактирование внешней обработки (отчета) в режиме «Конфигуратор». Для использования в режиме «1С:Предприятие» внешнюю обработку (отчет) необходимо сохранить в двоичном формате .erf/.epf. Для работы используется формат версии 2.0.

Любой объект конфигурации типа «Обработка» или «Отчет» может быть сохранен в файл внешней обработки (отчета), и наоборот – существующий объект конфигурации может быть заменен внешней обработкой (отчетом). Для внешней обработки (отчета) может быть создана справочная информация, как и для других объектов конфигурации.

Сохранение внешней обработки (отчета) в XML-формате можно использовать в том случае, если предполагается хранить эту обработку (отчет) во внешней системе контроля версий. XML-формат для этого более удобен, чем двоичный формат .erf (.epf). Конфигуратор позволяет открывать для редактирования внешние обработки (отчеты) в любом формате. При этом запись по умолчанию выполняется ровно в том же формате, в котором внешняя обработка (отчет) была открыта в конфигураторе. Для смены формата хранения необходимо выполнить команду «Файл – Сохранить как...» или «Файл – Сохранить копию...» и в открывшемся диалоге выбрать нужный формат сохранения.

Планы видов характеристик

В системе «1С:Предприятие» объекты «Планы видов характеристик» предназначены для описания множеств однотипных объектов аналитического учета.

При разработке прикладных решений часто возникают вопросы организации хранения некоторых свойств прикладных объектов, состав и тип которых заранее (на этапе разработки прикладного решения) не известен.

Примерами таких свойств являются:

- цвет, размер, габаритные размеры и т. д. для номенклатуры;
- федеральный округ, признак постоянного покупателя и т.д. для контрагента;
- количество и тип разрезов аналитического учета (для конкретного счета) для плана счетов.

Список таких свойств (далее будем называть их характеристиками) во время разработки не определен ни по составу, ни по типу хранимых данных. Кроме того, возможна ситуация, когда перечень таких характеристик достаточно большой и может быть не востребован каждым пользователем прикладного решения. Следует отметить, что набор характеристик может различаться для разных групп объектов, например, для одной группы элементов номенклатуры используются такие характеристики, как цвет и размер, а для другой – только характеристика, описывающая габаритные размеры. Также необходимо учесть, что характеристики могут вводиться в процессе эксплуатации самими пользователями системы.

Решить такую проблему можно двумя способами:

1) Для каждой характеристики заводится отдельный реквизит нужного типа в соответствующем объекте. Недостатками такого подхода являются:

- каждая характеристика – это отдельное поле таблицы базы данных, которое будет занимать место в базе данных вне зависимости от того, используется характеристика или нет;
- при большом количестве характеристик количество «лишних» полей будет очень большим, что приведет к неудобству работы как разработчика, так и пользователя;
- при необходимости добавить новую характеристику придется дорабатывать прикладное решение: добавлять новый реквизит, везде в прикладном решении учитывать его, что трудоемко и неудобно.

2) Использовать специальный объект, в котором описываются характеристики с указанием того, как называется характеристика, какого она типа и т. д. Этот подход лишен выше перечисленных недостатков: для добавления новой

характеристики нет необходимости изменять прикладное решение, не расходуется избыточное место в базе данных. Кроме того, имеется возможность самому пользователю добавлять новые характеристики объектов. В системе «1С:Предприятие» реализован второй способ. В качестве такого специального объекта выступает *план видов характеристик*.

К недостаткам такого подхода можно отнести более сложную логику организации и использования характеристик. Однако, затраты, связанные с усложнением прикладного решения, будут понесены только один раз – при разработке решения. По мере увеличения количества (и типов) характеристик, дополнительная модернизация прикладного решения не требуется.

План видов характеристик также можно использоваться для организации аналитического учета в бухгалтерском учете. В этом случае каждый элемент (с уникальным набором типов) будет являться видом субконто и при редактировании счета в плане счетов можно указать виды субконто, используемые для конкретного счета. При формировании проводки, для заполнения будет использоваться состав субконто (соответствующего типа), какой был задан при редактировании плана счетов.

Планы видов расчета

Объекты данного вида предназначены для создания видов расчетов, используемых в механизмах периодических расчетов.

Планы счетов

План счетов является одним из основных понятий бухгалтерского учета. Планом счетов в бухгалтерском учете называется совокупность синтетических счетов, предназначенных для группировки информации о хозяйственной деятельности предприятия. Информация, накапливаемая на таких синтетических счетах, позволяет получить полную информацию об имуществе предприятия в денежном выражении.

Объект конфигурации «План счетов» является программной реализацией соответствующего понятия бухгалтерского учета.

Планы обмена

Объекты данного вида предназначены для организации обмена данными между различными информационными базами, а также информационными базами и внешними программными системами.

Бизнес-процессы и задачи

Позволяют создавать формализованные описания типичных последовательностей работ, выполняемых в организации, и на их основе формировать списки задач, которые необходимо выполнить тому или иному сотруднику организации в данный момент.

Пример. Процесс продажи товара может быть представлен как последовательность выписки счета, его утверждения, получения наличной оплаты и отгрузки товара со склада. За выполнение каждого из этапов могут отвечать различные сотрудники. Таким образом, в любой момент времени можно определить, в каком состоянии находится процесс продажи товара и кто из сотрудников в данный момент должен выполнить какие-либо действия.

Регистры

Регистры предназначены для хранения и обработки различной информации, отражающей хозяйственную или организационную деятельность предприятия и не имеющей объектной природы.

В регистрах обычно хранится информация об изменении состояния объектов или другая информация, не отражающая непосредственно объекты предметной области. Например, в регистрах может храниться информация о курсах валют или информация о приходе и расходе товаров.

В системе «1С:Предприятие» существует 4 вида регистров:

- регистры сведений;
- регистры накопления;
- регистры расчетов;
- регистры бухгалтерии.

Единицей хранения информации в регистрах является запись. Однако, сама запись, например, о курсе валюты не представляет собой ничего существенного. Она

не соответствует никакому объекту в предметной области. Существенным является только то, что в ней содержится валюта, дата и курс валюты, установленный на эту дату. Можно удалить эту запись и внести такую же – это не повлияет на логику работы системы. Соответственно, у записей регистров не существует ссылок, и в полях базы данных нельзя хранить ссылки на записи регистров.

Основная задача *регистра сведений* – хранить существенную для прикладной задачи информацию, состав которой развернут по определенной комбинации значений и при необходимости развернут во времени. Например, если необходимо хранить информацию о ценах конкурентов на продаваемые организацией товары, то собранная информация о ценах разворачивается по товарам и конкурентам. А если необходимо отслеживать динамику изменений цен и информация будет заноситься периодически, то хранимая информация разворачивается также и во времени.

Регистр сведений фактически представляет собой в общем случае многомерный массив данных, необходимый для реализации функции, которая может выдать нужную информацию по определенному набору аргументов. Аргументы функции называются *измерениями*, а результат функции – *ресурсами*. В приведенном выше примере двумерный регистр ЦеныКонкурентов будет содержать измерения Конкурент и Товар и ресурс Цена. Ресурсов может быть больше чем один: например, можно хранить оптовую и розничную цены.

Помимо измерений и ресурсов для регистра сведений может быть создан набор реквизитов. Реквизиты позволяют включать в записи регистров различную дополнительную информацию. Реквизиты не влияют на значения ресурсов регистра и могут использоваться для анализа записей регистра.

Регистры сведений, информация в которых развернута во времени, называются *периодическими*. Для периодических регистров сведений система поддерживает такие стандартные операции, как получение наиболее позднего или наиболее раннего значения, а также получение среза наиболее поздних или ранних значений.

Уникальность записей по набору измерений принципиально отличает регистры сведений от регистров накоплений, которые позволяют вводить несколько

записей с одинаковым значением измерений и периодом.

Регистр накопления – это объект конфигурации, предназначенный для хранения движений регистра и итоговой информации.

Проблема, которая обычно возникает при создании «хранилища» сводной информации, состоит в определении его структуры: в каких разрезах следует накапливать сводные данные, чтобы затем можно было извлечь нужную информацию без долгой обработки. Система «1С:Предприятие» использует простые и в то же время гибкие средства для создания регистров накопления.

Методы встроенного языка позволяют получить остатки регистра накопления на заданный момент времени. Реализована возможность фильтрации по значениям измерений, а также получения остатков в разрезе других измерений.

Пример. Предположим, что в создаваемой программе торгово-складского учета требуется хранить сведения о количестве и стоимости каждого товара на каждом складе. В дальнейшем предполагается получать информацию такого типа: «остаток конкретного товара на конкретном складе», «остаток конкретного товара всего, на всех складах», «стоимость всех товаров на конкретном складе».

В идеологии системы «1С:Предприятие» регистр накопления такого вида представляет собой прямоугольную систему координат, на одной оси которой находятся склады, на другой – товары, а на пересечении конкретного склада и конкретного товара находятся цифры количества товара и стоимости товара.

Изменение состояния регистров накопления выполняется обычно при проведении документа.

Специалист, выполняющий конфигурирование системы, имеет возможность предоставить конечному пользователю средства просмотра движений регистров. Конфигуратор позволяет создавать экранные и печатные формы для просмотра и анализа движений регистров.

В процессе разработки конфигурации можно создать неограниченное количество регистров накопления.

Помимо измерений и ресурсов для регистра накопления может быть создан набор реквизитов. Реквизиты позволяют включать в движения регистров различную

дополнительную информацию. Реквизиты не влияют на значения ресурсов регистра и могут использоваться для анализа движений регистра.

Движения регистров оказывают влияние на его итоги. Итоги – это сводная информация регистров, которая получается путем суммирования значений, вносимых движениями регистров.

В отличие от движений регистра, нет возможности непосредственно просматривать итоги регистра накопления. Для обращения к итогам в конфигурации может быть создано необходимое число отчетов, которые будут обращаться к итогам и выдавать их в виде товарных отчетов, складских карточек и ведомостей и т. д.

Для повышения производительности системы в случае использования оборотных регистров накопления предназначен механизм агрегатов. Агрегат – физическая таблица базы данных, хранящая сводные обороты всех ресурсов регистра по выбранным измерениям с выбранной периодичностью и за определенный период. В регистре, для которого формируются агрегаты, не может быть более 30 измерений.

Специализированные объекты конфигурации

Помимо объектов, описывающих предметную область учета, конфигурация содержит ряд вспомогательных объектов, не относящихся непосредственно к деятельности предприятия, однако тесно связанных с функционированием самой системы. Это механизмы взаимодействия пользователей с системой «1С:Предприятие» (командный интерфейс, критерии отбора, права доступа различных групп пользователей к различной информации); вспомогательные объекты оформительского назначения, позволяющие производить конфигурирование на основе сформированных стилей; библиотеки картинок с учетом национального языка; модуль приложения и общие модули, в которых располагаются процедуры и функции, доступные из прочих модулей конфигурации; общие макеты печатных форм и многое другое.

Все эти объекты доступны в ветке «Общие» дерева конфигурации.

1.3.2.4 Подчинение группы объектов

В зависимости от вида объекта конфигурации он может иметь различные подчиненные группы объектов. Например, реквизиты, измерения, формы, табличные части и т. д. Состав подчиненных объектов зависит от типа объекта.

Реквизиты – дополнительная информация об объекте, доступная только в пределах этого объекта.

Табличные части – наборы дополнительной информации об объекте, представленной в виде таблицы (число строк одной табличной части не может быть более 100 000).

Реквизиты табличных частей – состав табличной части объекта, доступный только в пределах табличной части объекта.

Формы – форма используется для ввода, просмотра и редактирования информации, хранящейся в объекте конфигурации, содержит модуль формы – программу на встроенном языке системы «1С:Предприятие». Способность иметь визуальное представление позволяет объекту конфигурации организовать интерактивное взаимодействие с пользователем. Характер такого взаимодействия разрабатывается специалистом, осуществляющим конфигурирование системы «1С:Предприятие». Каждый объект может иметь несколько форм.

Команды – используются для выполнения различных операций с объектом. Команды могут быть независимыми и параметризуемыми.

Макеты – табличные, HTML или текстовые документы (также могут использоваться двоичные и Active-документы), предназначенные для формирования печатных форм объекта.

Графы – графы журнала документов.

Измерения – для регистров это объекты конфигурации, данные о которых учитываются в регистре.

Ресурсы – данные, учитываемые в регистре.

Группы подчиненных объектов не удаляются и не имеют редактируемых свойств.

1.3.2.5 Типизированные и типобразующие объекты

Одним из основных свойств некоторых объектов конфигурации является *Тип данных*. Это свойство определяет, какого рода информацию может содержать объект конфигурации. Тип данных объекта конфигурации назначается при создании или редактировании свойств объекта в процессе настройки конфигурации.

Объекты конфигурации, для которых может быть указан тип информации, содержащейся в объекте, в системе «1С:Предприятие» называются *типизированными* объектами конфигурации.

Такие объекты конфигурации, как Справочник, Документ, Обработка, не являются типизированными объектами, так как содержат «комплексную» информацию и, в свою очередь, включают в себя типизированные объекты конфигурации.

Типы данных, которые может принимать объект конфигурации, можно разделить на две группы.

Первую группу составляют *примитивные типы данных*: Число, Строка, Дата и Булево. Следовательно, информация, хранящаяся в объекте конфигурации, может быть числом, произвольной строкой символов, датой или логической величиной.

Кроме этого, некоторые объекты конфигурации системы «1С:Предприятие» также могут сами образовывать типы данных. Например, константе может быть назначен тип данных ДокументСсылка. В этом случае значение константы будет представлять собой ссылку на один из существующих в системе «1С:Предприятие» документов.

Объекты конфигурации, которые могут образовывать типы значений конфигурации, в системе «1С:Предприятие» называются *типобразующими* объектами конфигурации. К таким объектами в системе «1С:Предприятие» относятся:

- справочники,
- документы,
- планы видов характеристик,
- планы счетов,

- планы видов расчета,
- планы обмена,
- бизнес-процессы,
- задачи,
- перечисления.

Типообразующие объекты конфигурации образуют тип данных сразу после создания в конфигураторе объекта любого из таких типов. При этом появляются сразу три новых вида типов: Ссылка, Объект и Список. Например, когда в конфигураторе создается новый справочник, то в списке типов данных появляются новые типы данных: СправочникСсылка.<ИмяСправочника>, СправочникОбъект.<ИмяСправочника> и СправочникСписок.<ИмяСправочника>. Такие типы данных могут быть присвоены любому из типизированных объектов конфигурации

Некоторые данные могут иметь составной тип. Также допускается выбор специального типа ЛюбаяСсылка.

1.3.3 Командный интерфейс

Командный интерфейс – это основное средство навигации пользователя по функциональности конфигурации. Командный интерфейс строится на основе подсистем. Разработчик конфигурации включает прикладные объекты в соответствующие подсистемы.

На основе этой информации (структуры подсистем и привязки объектов к подсистемам) система автоматически строит командный интерфейс для пользователя. Пользователю отображается структура прикладного решения (иерархия подсистем) и предоставляются стандартные команды доступа к функциональности прикладных объектов (вызов списков справочников, документов, открытие отчетов, обработок и т. д.). Однако разработчик, разумеется, может отредактировать предлагаемое системой построение командного интерфейса (изменить порядок, видимость команд).

Сами команды, включаемые в командный интерфейс (открытие списков, ввод новых объектов, открытие отчетов и т. д.), предоставляются системой

автоматически. Но разработчик может создать свои команды, которые будут включаться в командный интерфейс.

Цель создания интерфейса – обеспечить структурированный доступ пользователей к той информации, которая необходима им в соответствии с их обязанностями.

1.3.4 Форма

Совокупность экранного диалога, модуля, реквизитов и команд называется *формой*.

Большинство объектов конфигурации в системе «1С:Предприятие» могут иметь визуальную форму. В самом общем случае форма как объект конфигурации состоит из следующих частей:

- экранный диалог, используемый для ввода и редактирования информации;
- модуль формы – программа на встроенном языке системы «1С:Предприятие»;
- список реквизитов;
- команды, используемые в форме.

Любая из этих составных частей формы может отсутствовать, то есть не содержать информации.

С помощью формы можно реализовать интерактивное взаимодействие прикладного объекта с пользователем.

1.3.5 Модуль

Модулем называется программа на встроенном языке системы «1С:Предприятие». Модули располагаются в заданных точках структуры конфигурации и вызываются для выполнения в заранее известные моменты работы системы «1С:Предприятие».

В конфигурации существует несколько видов модулей:

- управляемого приложения;
- модуль обычного приложения;
- модуль внешнего соединения;

- модуль сеанса;
- общие модули;
- модули форм;
- модули объектов конфигурации (справочников, документов, бизнес-процессов, задач, отчетов, обработок и т.д). В таких модулях возможно объявление переменных, процедур и функций, которые будут доступны при работе с объектом извне во встроенном языке, дополняя контекст объекта. В этих модулях располагают процедуры обработки различных событий, а также процедуры, с помощью которых выполняются действия над объектом, инициированные вне данного объекта (например, выполнение печати).

- модули менеджеров объектов конфигурации (справочников, документов, планов счетов, задач, отчетов, обработок, регистров сведений, регистров накопления, регистров бухгалтерии, регистров расчета, перечислений, журналов документов, хранилищ настроек);

- модули наборов записей (регистров сведений, регистров накопления, регистров бухгалтерии, регистров расчета),

- модули команд.

Для доступа к модулю необходимо в контекстном меню объекта конфигурации выбрать пункт «Открыть модуль...».

1.3.6 Макет

Макетом в системе программ «1С:Предприятие» называется объект конфигурации, предназначенный для формирования печатных форм.

Макеты могут быть следующих типов:

- табличный документ - подготовка макета производится с помощью табличного редактора;

- текстовый документ – подготовка текстового макета производится с помощью редактора текстовых макетов;

- двоичные данные – используются двоичные данные;

- ActiveDocument – предполагает использование технологии OLE Active document (документы Microsoft Office OpenXML не могут быть использованы в качестве макета ActiveDocument);
- HTML-документ – предполагает использование редактора HTML-документа;
- географическая схема – предполагает использование географической схемы, подготовленной в редакторе географических схем, в качестве макета;
- графическая схема – предполагает использование подготовленной в редакторе графической схемы;
- схема компоновки данных – предполагает использование схемы компоновки данных, подготовленной в конструкторе.

1.4 Варианты работы системы

«1С:Предприятие» поддерживает два варианта работы:

- файловый;
- клиент-серверный.

Все прикладные решения в обоих случаях работают полностью идентично.

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.

Такой вариант работы обеспечивает легкость установки и эксплуатации автоматизированной системы. При этом для работы с информационной базой не требуются дополнительные программные средства, достаточно иметь операционную систему и «1С:Предприятие».

Файловый вариант «1С:Предприятия» обеспечивает высокую целостность информационной базы и простое создание резервных копий. Однако использование файлового варианта с подключением через веб-сервер рекомендуется использовать для работы небольшого количества пользователей, при условии отсутствия

длительных операций.

Клиент-серверный вариант предназначен для использования в рабочих группах или в масштабе предприятия. Он реализован на основе трехуровневой архитектуры «клиент-сервер».

Программа, работающая у пользователя (клиентское приложение), взаимодействует с кластером серверов «1С:Предприятия», а кластер, при необходимости, обращается к серверу баз данных (Microsoft SQL Server, PostgreSQL, IBM DB2 или Oracle Database). При этом физически кластер серверов «1С:Предприятия» и сервер баз данных могут располагаться как на одном компьютере, так и на разных. Это позволяет администратору при необходимости распределять нагрузку между серверами.

Использование кластера серверов «1С:Предприятия» позволяет сосредоточить на нем выполнение наиболее объемных операций по обработке данных. Например, при выполнении даже весьма сложных запросов программа, работающая у пользователя, будет получать только необходимую ей выборку, а вся промежуточная обработка будет выполняться на сервере.

Другим важным аспектом использования 3-уровневой архитектуры является удобство администрирования и упорядочивание доступа пользователей к информационной базе. В этом варианте пользователь не должен знать о физическом расположении конфигурации или базы данных. Весь доступ осуществляется через кластер серверов «1С:Предприятия».

Важной особенностью работы в клиент-серверном варианте является возможность работы сервера «1С:Предприятия» и серверов баз данных на различных операционных системах (семейство Windows и различные дистрибутивы Linux).

1.5 Технологические средства разработки

Для описания специфических алгоритмов обработки информации и создания интерфейса, ориентированного на удобное представление описанных в

конфигурации данных, в системе «1С:Предприятие» используются несколько технологических механизмов:

- встроенный программный язык.;
- механизм запросов;
- встроенный текстовый редактор;
- встроенный редактор форм;
- встроенный редактор табличных документов;
- встроенный редактор картинок;
- встроенный редактор HTML-документов;
- конструкторы:
- система настройки пользовательского интерфейса;
- подсистемы;
- система настройки прав доступа (роли);
- отладчик;
- хранилище конфигураций;
- поддержка конфигураций.

1.6 Контрольные вопросы

1. Каково предназначение кнопок «Конфигуратор» и «1С:Предприятие» на форме запуска 1С?
2. Как добавить новую информационную базу в список?
3. Как узнать в какой папке хранится информационная база из списка?
4. Как открыть информационную базу для редактирования?
5. Для чего служит режим «1С: Предприятие»?
6. Что такое конфигурация?
7. Как создать новый объект конфигурации стандартного типа?
8. Для чего нужны объекты типа подсистема?
9. Что такое справочник в конфигурации?
10. Для чего используются объекты типа «Справочник»?

11. Как отнести объект к подсистеме?
12. Как создать реквизит справочника?
13. Что может быть реквизитом справочника?
14. Для чего нужны таблицы в справочнике?
15. Как создать или изменить форму элемента справочника?
16. Что такое документ и для чего он предназначен?
17. Что означает проведение документа?
18. Для чего предназначены реквизиты документа?
19. Как создать документ?
20. Что такое табличная часть документа?
21. Как создать форму документа?
22. Что такое конструктор форм?
23. Как создать элементы формы?
24. Что такое события?
25. Что такое обработчик событий?
26. Как создать обработчик событий?
27. Что такое модуль и для чего он нужен?
28. Для чего предназначен объект конфигурации «Отчет»?
29. Как пользоваться конструктором отчетов?
30. Для чего нужен объект конфигурации макет?
31. Что такое конструктор печати?
32. Как создать макет?
33. Как редактировать макет?
34. Как создавать и выводить на печать новые области?
35. Что такое регистры накопления?
36. Что такое измерения регистра накопления?
37. В чем преимущество использования регистров накопления?
38. Как создать движение документа?
39. Как создать ресурс регистра?
40. Что такое регистр сведений?

41. Чем регистр сведений отличается от регистра накопления?
42. Для чего предназначены регистры сведений?
43. Что значит периодический регистр сведений?
44. Как создать периодический регистр?

1.7 Практические задания

1. Создайте средствами «1С:Предприятие» новую информационную базу.
2. Перепишите папку с созданной базой данных на другой диск, компьютер и свяжите с ней интерфейс 1С.
3. Измените «Основной режим запуска», посмотрите результат.
4. Создайте объект конфигурации «Константа».
5. Создайте объект конфигурации справочник «Сотрудники».

2 ВСТРОЕННЫЙ ЯЗЫК

2.1 Назначение и краткая характеристика

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

Встроенный язык системы «1С:Предприятие» 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, которое можно использовать одновременно в одном исходном тексте.

Язык «1С:Предприятие» обладает некоторыми объектно-ориентированными возможностями, например, правила доступа к свойствам и методам специализированных типов данных (документам, справочникам и т. п.) подобны свойствам и методам объектов, используемых в других объектно-ориентированных языках. Однако при этом специализированные типы данных не могут определяться средствами самого языка, а задаются только в визуальном режиме.

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, структур, соответствий и других универсальных коллекций значений.

2.2 Программные модули

Программные модули в конфигурации системы «1С:Предприятие» не являются самостоятельными программами в общепринятом понимании этого слова, поскольку они являются только частью всей конфигурации. *Программный модуль* – это текст на встроенном языке, в котором размещены тексты процедур и функций с необходимыми алгоритмами, вызываемые системой в определенные моменты

работы. Поэтому программный модуль не имеет традиционных границ: «Начало модуля» – «Конец модуля».

Место размещения конкретного программного модуля предоставляется конфигуратором в тех точках конфигурации, 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 программных модулей

Каждый программный модуль связан с остальной частью конфигурации. Эта связь называется ***контекстом*** выполнения модуля.

Выделяют два вида контекста:

- глобальный контекст;
- локальный контекст выполнения конкретного модуля.

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 объектов, их свойств и методов.

Виды программных модулей

В системе «1С:Предприятие» существуют несколько видов программных модулей. Они различаются по месту размещения и доступному контексту.

Модулем управляемого приложения называется модуль, который автоматически выполняется в момент загрузки конфигурации, при старте системы «1С:Предприятие» в следующих режимах:

- тонкого клиента;
- веб-клиента;

- толстого клиента в режиме управляемого приложения.

Модуль управляемого приложения предназначен для отработки действий, связанных с сеансом работы конечного пользователя (прежде всего обработки начала и окончания сеанса работы). Этот модуль недоступен для процедур, работающих на сервере. В нем рекомендуется реализовывать только обработчики соответствующих событий. Следует помнить, что необработанное исключение в каком-либо обработчике события модуля управляемого приложения (ПередНачаломРаботыСистемы(), ПриНачалеРаботыСистемы(), ПередЗавершениемРаботыСистемы(), ПриЗавершенииРаботыСистемы()) приведет к аварийному завершению работы всей системы.

Процедуры и функции модуля управляемого приложения, а также переменные, для которых в заголовке указано ключевое слово *Экспорт*, являются доступными:

- в неглобальных клиентских общих модулях;
- клиентских процедурах и функциях модуля команды;
- клиентских процедурах и функциях модуля управляемой формы.

В контексте модуля управляемого приложения доступны:

- часть глобального контекста, которая может исполняться в управляемом приложении;
- экспортируемые процедуры и функции любых клиентских общих модулей;
- экспортируемые процедуры и функции серверных неглобальных общих модулей, у которых установлено свойство *Вызов сервера*.

Модуль внешнего соединения расположен, как и модуль приложения, в корневом разделе конфигурации. В нем располагаются процедуры-обработчики событий, которые инициализируются при старте и окончании работы системы в режиме внешнего соединения (СОМ-соединения).

В модуле внешнего соединения возможно объявление переменных, а также объявление и описание процедур и функций, которые будут доступны для внешнего приложения.

Этот модуль присутствует только в сессии внешнего соединения, при котором полностью отсутствует пользовательский интерфейс.

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. Этот общий модуль всегда исполняется в привилегированном режиме сервера «1С:Предприятия». Установка параметров сеанса выполняется в обработчике события УстановкаПараметровСеанса.

Модуль сеанса может содержать только определения процедур и функций, может использовать процедуры из общих модулей конфигурации и не содержит экспортируемых процедур и функций.

Вызов обработчика УстановкаПараметровСеанса() производится до вызова обработчика события ПередНачаломРаботыСистемы() (ПриНачалеРаботыСистемы()) в случае модуля внешнего соединения).

Общие модули располагаются в отдельной ветке дерева метаданных. Основным назначением общих модулей является содержание общих алгоритмов конфигурации, доступных из разных модулей. В общих модулях отсутствует раздел определения переменных и раздел основной программы, то есть они содержат только раздел процедур и функций.

Набор прикладных объектов имеет собственные модули — *модули прикладных объектов*. К таким объектам относятся: менеджеры значения константы, справочники, документы, отчеты, обработки, планы видов характеристик, планы счетов, планы видов расчетов, планы обмена, бизнес-процессы, задачи, регистры.

Модули располагаются в ветках конфигурации, в которых содержатся сами объекты, и являются свойствами объектов. Каждый объект имеет свой индивидуальный модуль. В этих модулях возможно объявление переменных, процедур и функций, которые будут доступны при работе с объектом извне во встроенном языке, дополняя контекст объекта.

Каждый прикладной объект имеет 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 за счет введения процедур и функций на встроенном языке. Фактически это позволяет описать методы для объекта конфигурации (например, справочника), которые относятся не к конкретному экземпляру объекта базы данных, а к самому объекту конфигурации.

Модуль менеджера не может иметь переменных и тела модуля.

Модули форм содержатся в формах конфигурации. Каждая форма имеет свой индивидуальный модуль. В этих модулях возможно объявление переменных, процедур и функций, которые будут доступны при работе с формой извне во встроенном языке, дополняя контекст формы.

Модуль команды предназначен для того, чтобы описать в нем на встроенном языке те действия, которые должна выполнить система при вызове команды. Модуль команды может содержать только описание процедур и функций и не может иметь переменных и тела модуля.

Формат программного модуля

Структура программного модуля включает следующие разделы:

- раздел определения переменных;
- раздел процедур и функций;
- раздел основной программы.

В конкретном программном модуле любой из разделов может отсутствовать.

Раздел определения переменных размещается от начала текста модуля до первого оператора *Процедура*, *Функция*, или любого исполняемого оператора. В этом разделе могут находиться только операторы объявления переменных *Перем*.

Раздел процедур и функций размещается от первого оператора *Процедура* или *Функция* до любого исполняемого оператора вне тела описания процедур или функций.

Раздел основной программы размещается от первого исполняемого оператора вне тела последней процедуры или функции до конца модуля. В этом разделе могут

находиться только исполняемые операторы. Раздел основной программы исполняется в момент инициализации модуля. Обычно в разделе основной программы имеет смысл размещать операторы инициализации переменных какими-либо конкретными значениями, которые необходимо провести до первого вызова любой из процедур или функций модуля.

Исходный текст программного модуля может состоять из операторов и комментариев.

2.3 Формат операторов и комментарии

Операторы имеют вид стандартного обращения к процедуре, за исключением оператора присваивания ($A = B;$) и синтаксических конструкций встроенного языка (например, таких как *Для*, *Пока*, *Если*). Между собой операторы обязательно следует разделять символом точка с запятой. Конец строки не является признаком конца оператора, т. е. операторы могут свободно переходить через строки и продолжаться на другой строке. Можно располагать произвольное число операторов в одной строке, разделяя их символом точка с запятой.

Операторы языка в программном модуле можно подразделить на две категории: операторы объявления переменных и исполняемые операторы.

Любой исполняемый оператор может иметь метку, используемую в качестве точки перехода в операторе *Перейти*.

В общем случае формат оператора языка следующий:

~метка:Оператор[(параметры)] [ДобКлючевоеСлово];

В качестве меток используются специальные идентификаторы, начинающиеся с символа тильда и состоящие из последовательности букв, цифр и символов подчеркивание. Чтобы пометить оператор, нужно поместить перед ним метку и следующий за ней символ двоеточие.

Пример. ~метка : A=B;

Комментарий используется для размещения в исходном тексте программного модуля всякого рода пояснений к работе модуля. Хорошим тоном программирования

считается, когда исходный текст содержит исчерпывающий комментарий с описанием алгоритма. В режиме исполнения программы комментарии пропускаются. В тексте программного модуля комментарий начинается парой символов «//» и заканчивается концом строки.

Пример.

```
A=B; // Это - комментарий  
// Это тоже комментарий
```

Комментарий, предшествующий процедуре или функции, может использоваться контекстной подсказкой конфигулятора.

2.4 Имена переменных, процедур и функций

Именем переменной, объявленной процедуры или функции может быть любая последовательность букв, цифр и знаков подчеркивания, начинающаяся с буквы или знака подчеркивания. Вновь создаваемые имена не должны совпадать с зарезервированными словами языка или именами свойств, непосредственно доступных в текущем контексте. Распознавание имен переменных, процедур и функций ведется без учета регистра букв.

2.5 Язык написания программных модулей

Встроенный язык системы «1С:Предприятие» является двуязычным. Почти все зарезервированные слова, имена типов значений, свойств, методов, событий имеют два имени: русское и английское. Исключение составляют слова, не имеющие аналогов в русском языке. В тексте программных модулей эти имена можно свободно смешивать, используя то русские, то английские имена без каких-либо ограничений.

2.5.1 Специальные символы

Во встроенном языке «1С:Предприятие» некоторые символы могут использоваться только для обозначения закрепленного за ними в системе действия. Такие символы называются специальными. Их перечень приведен в таблице 1.

Таблица 1. Специальные символы встроенного языка «1С:Предприятие»

Символ	Описание
//	комментарий
	символ переноса строки (используется только в начале строковых константах и означает, что данная строка является продолжением предыдущей)
~	начало метки оператора
:	окончание метки оператора
;	символ разделения операторов
()	используются в выражениях встроенного языка, также в них заключается список параметров методов, процедур, функций и конструкторов
[]	используются для обращения к элементу коллекции, к свойству объекта по строковому представлению имени свойства
,	используется для разделения элементов в списке параметров методов, процедур, функций и конструкторов
" "	используются для обозначения строковых литералов
' '	используются для обозначения литералов даты
.	десятичная точка в числовых литералах разделитель, используемый для обращения к свойствам и методам объектов
+	оператор сложения оператор конкатенации строк
-	оператор вычитания
*	оператор умножения
/	оператор деления
%	остаток от деления (допускается использование дробных значений делимого и делителя)
>	логическая операция больше
>=	логическая операция больше или равно
<	логическая операция меньше
<=	логическая операция меньше или равно
=	оператор присваивания
	логическая операция
<>	логическая операция не равно

2.5.2 Типы данных

Во встроенном языке системы «1С:Предприятие» поддерживается набор **примитивных типов данных**. Для большинства примитивных типов данных предусмотрена возможность использования в тексте модуля литералов, то есть указание значения соответствующего типа непосредственно в модуле.

Пример.

```
// Пример использования литерала типа Строка
А = "Моя строка";
// Пример использования литерала типа Булево
В = Истина;
// Пример использования литерала типа Число
В = 12345.6789;
```

Булево (Boolean)

Значения данного типа имеют два значения – Истина (True) и Ложь (False). Значения данного типа возвращаются в качестве результата вычисления логических выражений.

Дата (Date)

Значения данного типа содержат дату от Рождества Христова (с 01 января 0001 года) и время с точностью до секунды и представляются строку цифр, заключенную в одинарные кавычки вида 'ГГГГММДДччммсс', где:

- ГГГГ – четыре цифры года (включая тысячелетие и век);
- ММ – две цифры месяца;
- ДД – две цифры даты;
- чч – две цифры часа (в 24-часовом формате);
- мм – две цифры минут;
- сс – две цифры секунд.

Во встроенном языке в литерале типа Дата обязательно должно задаваться значение года, месяца и дня. Для задания даты, соответствующей началу отсчета, достаточно указать '00010101'. Допускается при указании литералов типа *Дата* опускать последние символы (секунды, минуты, часы и т. д.). Это означает, что данные параметры будут равны нулю (для времени) или единице (для даты).

В литерале даты допускается использование различных разделителей.

Пример.

```
Дата ('2008.03.23 10:45:23') = "23.03.2008 10:45:23"
```

Число (Number)

Числовым типом может быть представлено любое десятичное число. Определены основные арифметические операции над данными числового типа: сложение, вычитание, умножение и деление. Максимально допустимая разрядность числа – 32 знака.

В качестве разделителя целой и дробной части используется точка.

Пример.

```
A = 15;  
B = -968.612;
```

Строка (String)

Значения данного типа содержат строку произвольной длины в формате Unicode. Строка закодирована в формате UTF-16.

Литералы строкового типа представляют собой набор символов, заключенных в кавычки. Для задания в строке символа " (кавычка) необходимо записать две кавычки подряд ("").

Допускаются «многострочные» строковые константы. В исходном тексте многострочные константы могут задаваться двумя способами:

➤ между фрагментами, представляющими отдельные строки многострочной строки, не должно встречаться никаких символов, за исключением пробелов, переводов строки и строк комментариев;

➤ каждая отдельная составляющая не замыкается кавычками, а на каждой последующей строке помещен символ переноса строки «|» (вертикальная черта). В этом варианте комментарии допускаются, если строка начинается с символа комментария «|».

Пример.

```
// Пример строки  
МояСтрока = "Это правильная строка";  
// Пример 1 многострочной строки  
МояМногострочнаяСтрока = "Это  
|правильная  
|многострочная  
|строка";  
// Пример 2 многострочной строки  
МояМногострочнаяСтрока = "Это тоже" //Это комментарий  
"правильная"
```

```
        "многострочная"  
        "строка";  
// Пример 3 строки с кавычками  
НазваниеФирмы = "ООО ""Василек""";
```

Неопределено (Undefined)

Значение данного типа применяется, когда необходимо использовать пустое значение, не принадлежащее ни к одному другому типу. Например, такое значение изначально имеют реквизиты с составным типом значения. Существует единственное значение данного типа, задаваемое литералом *Неопределено (Undefined)*.

Тип (Type)

Значения данного типа используются для идентификации типов значений. Это необходимо для определения и сравнения типов. Данный тип не имеет литералов и возвращается функциями встроенного языка *ТипЗнач* и *Тип*.

NULL

Значения данного типа используются исключительно для определения отсутствующего значения при работе с базой данных, например, при соединении таблиц.

2.5.3 Оператор присваивания и операции

Оператор присваивания (символ «=») означает назначение нового значения <Источник> переменной, обозначенной как <Назначение>:

```
<Назначение> = <Источник>;
```

В качестве назначения может выступать переменная или свойство объекта встроенного языка, которое допускает запись.

Пример.

```
А = В;  
Стр1 = "777";  
ДатаДокумента = '20020717';
```

Выражение – это математическая, логическая или строковая формула, состоящая из соответствующих операций, по которой вычисляется значение. Математическое и логическое выражение может стоять справа от оператора присваивания, быть параметром процедур или функций. Логическое выражение также может быть условием в управляющих конструкциях *Если*, *Пока*, *Для*.

Выражения состоят из констант, переменных и функций, связанных символами логических и/или арифметических операций.

Арифметические операции

Виды арифметических операций, определенных во встроенном языке «1С:Предприятие» приведены в таблице 2.

Таблица 2. Арифметические операции

Операция	Описание действия
Сложение	операнд1 + операнд2
Вычитание	операнд1 - операнд2
Умножение	операнд1 * операнд2
Деление	операнд1 / операнд2
Остаток от деления	операнд1 % операнд2
Унарный минус	- операнд1

Арифметические операции имеют один или два операнда, в зависимости от типа которых операция имеет ту или иную семантику. Тот или иной семантический вариант операции определяется по первому операнду. В случае несовпадения типа второго операнда с требуемым значение преобразуется к требуемому типу в соответствии с правилами преобразования типов. Если тип первого операнда не соответствует ни одному из допустимых типов, то в зависимости от ситуации может производиться преобразование типов или возбуждаться состояние ошибки выполнения. Допустимые типы операндов приведены в таблице 3.

При выполнении арифметических операций с типом *Дата* следует помнить, что результат может иметь точность до 4 знаков после запятой (точность времени – сотни микросекунд). Однако, при выполнении операции записи такого значения в базу данных (в реквизите соответствующего типа), дробная часть числа будет отброшена. В связи с этим рекомендуется округлять до целого значение типа *Число* при выполнении арифметических операций со значением типа *Дата*.

Таблица 3. Допустимые типы операндов в арифметических операциях

Операция	Возможные типы операндов
Сложение	число1 + число2 дата + число (к дате прибавляется число секунд)
Вычитание	число1 - число2 дата - число (от даты отнимается число секунд) дата - дата (результатом является разница между двумя датами, измеренная в секундах)
Умножение	число1 * число2
Деление	число1 / число2
Остаток от деления	число1 % число2

Операция конкатенации

Операция конкатенации («+») используется для того, чтобы присоединить одну строку к другой. Длина результирующей строки равна сумме длин соединяемых строк. В случае несовпадения типа данных второго или последующих операндов со строковым типом их значение преобразуется к строковому типу в соответствии с правилами преобразования типов.

Пример.

ФИО = Фамилия + " " + Имя + " " + Отчество;

Логические операции

Логическая операция сравнивает операнды и вырабатывает значение типа *Булево*: Истина или Ложь. Существует два вида логических операций: операции сравнения и булевы операции. В операциях сравнения сравниваются два значения. Булевы операции выполняются над значениями типа *Булево*, реализуя булеву алгебру. Символы булевых операций могут комбинироваться, образуя составные операции.

В языке определены следующие виды операций сравнения: больше, больше или равно, равно, меньше, меньше или равно, не равно.

Операции сравнения больше, больше или равно, меньше, меньше или равно определены для следующих типов операндов: число, строка, дата.

Операции сравнения равно и не равно определены для любых типов операндов.

В языке определены следующие виды булевых операций:

- и (and) – конъюнкция;
- или (or) — дизъюнкция;
- не (not) — логическое отрицание.

Логические выражения вычисляются слева направо. Для того чтобы избежать неоднозначности и управлять последовательностью операндов, следует применять круглые скобки.

Приоритеты вычисления выражений встроенного языка

В таблице 4 представлен список операторов встроенного языка в порядке увеличения приоритета. Операторы с одинаковым приоритетом вычисляются слева направо.

Таблица 4. Операторы встроенного языка в порядке увеличения их приоритета

Операция	Описание
или (or)	дизъюнкция
и (and)	конъюнкция
не (not)	логическое отрицание
<, <=, >, >=, <>, =	сравнение
+, -	сложение, вычитание
*, /, %	умножение, деление
-, +	унарные минус, плюс
., ()	разыменованье, вызов метода и т.д.

2.5.4 Операторы и синтаксические конструкции

? (вычислить выражение по условию)

Позволяет вычислить одно из двух заданных выражений в зависимости от результата вычисления логического выражения.

Синтаксис:

? (<Логическое выражение>, <Выражение1>, <Выражение2>)

Если результат вычисления логического выражения Истина, то будет вычисляться <Выражение 1>, если - Ложь, то <Выражение 2>.

Пример:

```
Статус = ? (ПолучитьСкидку() > 10, "Особый клиент", "Обычный клиент");  
Предупреждение (Статус);
```

ВызватьИсключение (Raise)

При использовании данной формы оператора вызывается новое исключение.

Синтаксис:

ВызватьИсключение <Выражение>

Англоязычный синтаксис:

Raise <Expression>

Результат вычисления выражения преобразуется к строке, и данная строка используется в качестве описания исключения.

Пример.

```
ВызватьИсключение "Документ не может быть проведен";
```

Выполнить (Execute)

Позволяет выполнить фрагмент кода, который передается ему в качестве строкового значения.

Примечание. Не рекомендуется реализовывать с помощью этого метода существенную часть функциональности прикладных решений. Исполняемый код не может содержать в себе отдельных процедур или функций, а также явного объявления переменных, т. к. исполнение кода само по себе идет в рамках процедуры или функции, в которой использован этот оператор.

Синтаксис:

Выполнить(<Строка>)

Англоязычный синтаксис:

Execute(<Строка>)

Пример.

```
// Выводит в окно сообщений текущую дату  
Выполнить ("Сообщить (ТекущаяДата ())");
```

ДобавитьОбработчик (AddHandler)

При добавлении обработчика события производится проверка соответствия числа параметров события числу параметров метода, назначаемого в качестве обработчика.

Синтаксис:

ДобавитьОбработчик <Событие>, <ОбработчикСобытия>;

Англоязычный синтаксис:

AddHandler <Событие>, <ОбработчикСобытия>;

Для (For)

Оператор цикла *Для* предназначен для циклического повторения операторов, находящихся внутри конструкции *Цикл – КонецЦикла*. Перед началом выполнения цикла значение <Выражение1> присваивается переменной <Имя переменной>. Значение <Имя переменной> автоматически увеличивается при каждом проходе цикла. Величина приращения счетчика при каждом выполнении цикла равна 1. Цикл выполняется, пока значение переменной <Имя переменной> меньше или равно значению <Выражение 2>. Условие выполнения цикла всегда проверяется вначале, перед выполнением цикла.

Синтаксис:

Для <Имя переменной> = <Выражение 1> По <Выражение 2> Цикл

// Операторы

[Прервать;]

// Операторы

[Продолжить;]

// Операторы

КонецЦикла;

Англоязычный синтаксис:

For <Имя переменной> = <Выражение 1> To <Выражение 2> Do

// Операторы

[Break;]

// Операторы

[Continue;]

// Операторы

EndDo;

Прервать

Позволяет прервать выполнение цикла в любой точке. После выполнения этого оператора управление передается оператору, следующему за ключевым словом **КонецЦикла**.

Продолжить

Немедленно передает управление в начало цикла, где производится вычисление и проверка условий выполнения цикла. Операторы, следующие в теле цикла за ним, на данной итерации обхода не выполняются.

Пример.

```
// Перебор дней текущего месяца
ПоследнийДеньМесяца = День(КонецМесяца(ТекущаяДата()));
Для ТекДень = 1 по ПоследнийДеньМесяца Цикл
    Состояние("Обрабатывается день: "+ ТекДень);
    // Операторы обработки очередного дня месяца
    . . .
КонецЦикла;
```

Для каждого (For each)

Оператор цикла *Для каждого* предназначен для циклического обхода коллекций значений. При каждой итерации цикла возвращается новый элемент коллекции. Обход осуществляется до тех пор, пока не будут перебраны все элементы коллекции, или может быть завершён досрочно при выполнении оператора

Прервать.

Синтаксис:

Для каждого <Имя переменной 1> Из <Имя переменной 2> Цикл

```
// Операторы
[Прервать;]
// Операторы
[Продолжить;]
// Операторы
КонецЦикла
```

Англоязычный синтаксис:

For each <Имя переменной 1> In <Имя переменной 2> Do

```
// Операторы
[Break;]
```

```
// Операторы
[Continue;]
// Операторы
```

```
EndDo;
```

Прервать

Позволяет прервать выполнение цикла в любой точке. После выполнения этого оператора управление передается оператору, следующему за ключевым словом *КонецЦикла*.

Продолжить

Немедленно передает управление в начало цикла, где производится вычисление и проверка условий выполнения цикла. Операторы, следующие в теле цикла за ним, на данной итерации обхода не выполняются.

Пример.

```
// Перебор строк табличной части документа.
Документ = Документы.РасходнаяНакладная.НайтиПоКоду(12345);
// Проверим, найден нужный нам документ или нет
Если Не Документ.Пустая() Тогда
    Для каждого СтрокаСостава из Документ.Состав Цикл
        Состояние ("Строка: " + Документ.Состав.Индекс(СтрокаСостава)+1);
        // Операторы обработки очередной строки табличной части
        ...
    КонецЦикла;
КонецЕсли;
```

Если (If)

Оператор *Если* управляет выполнением программы, основываясь на результаты одного или более логических выражений. Оператор может содержать любое количество групп операторов, возглавляемых конструкциями *ИначеЕсли* – *Тогда*.

Синтаксис:

```
Если <Логическое выражение> Тогда
    // Операторы
[ИначеЕсли <Логическое выражение> Тогда]
    // Операторы
```

```
[Иначе]
    // Операторы
КонецЕсли;
```

Англоязычный синтаксис:

```
If <Логическое выражение> Then
    // Операторы
[ElseIf <Логическое выражение> Then]
    // Операторы
[Else]
    // Операторы
EndIf;
```

ИначеЕсли

Логическое выражение, следующее за ключевым словом *ИначеЕсли*, вычисляется только тогда, когда условия в *Если* и всех предшествующих *ИначеЕсли* оказались равны Ложь. Операторы, следующие за конструкцией *ИначеЕсли – Тогда*, выполняются, если результат логического выражения в ветке *ИначеЕсли* равен Истина.

Иначе

Операторы, следующие за ключевым словом *Иначе*, выполняются, если результаты логических выражений в конструкции *Если* и всех предшествующих конструкциях *ИначеЕсли* оказались равны Ложь.

Пример.

```
Если ДеньНедели(ТекущаяДата()) = 6 Тогда
    Сообщить("Сегодня суббота.");
ИначеЕсли ДеньНедели(ТекущаяДата()) = 7 Тогда
    Сообщить("Сегодня воскресенье.");
Иначе
    Сообщить("Сегодня рабочий день.");
КонецЕсли;
```

Новый (New)

Оператор позволяет создать значение указанного типа. Допустим только для тех типов, для которых разрешено создание новых значений. Для прикладных

объектов необходимо использовать функциональную форму оператора *Новый* (вариант 2), так как при проверке модулей в конфигураторе не определены типы для прикладных объектов.

Синтаксис (вариант 1):

Новый <Имя типа>[(<Парам 1>, ..., <Парам N>)]

После имени типа в скобках могут указываться параметры, если они определены в конструкторах для данного типа. Допустимое количество параметров и их назначение указываются в описании конструкторов объекта.

Пример.

```
// Пример создания массива из трех элементов.  
Массив = Новый Массив(3);
```

Синтаксис (вариант 2):

Новый (<Тип>[, <Параметры конструктора>])

Пример.

```
ТипЗначения = Тип("КвалификаторыСтроки");  
Параметры = Новый Массив(2);  
Параметры[0] = 20;  
Параметры[1] = ДопустимаяДлина.Переменная;  
КвалифСтр = Новый(ТипЗначения, Параметры);
```

Перейти (Goto)

Безусловная передача управления на другой оператор программы. Передает управление от одного оператора к другому.

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

Примечание. Метка в этом операторе не должна быть меткой перехода на оператор *Процедура* или *Функция*. Оператор безусловного перехода не может быть использован для передачи управления на операторы, находящиеся внутри конструкций: *Пока – КонецЦикла*, *Для – КонецЦикла*, *Для каждого – КонецЦикла*, *Если – КонецЕсли*, *Попытка – Исключение – КонецПопытки* извне этих конструкций.

Синтаксис:

Перейти <Метка>;

Пример.

```
Перейти ~Метка1;  
...  
~ Метка1: Сообщить ("Осуществлен переход по метке.");
```

Перем (Var)

Позволяет в явном виде объявить переменную.

Синтаксис:

Перем <Имя переменной 1> [Экспорт] [, <Имя переменной 2>, ...];

Англоязычный синтаксис:

Var <Имя переменной 1> [Export] [, <Имя переменной 2>, ...];

Экспорт

Необязательное ключевое слово. Указывает, что данная переменная доступна при обращении к контексту этого модуля из других модулей. Данное ключевое слово необходимо указывать для каждой объявляемой переменной отдельно. Не имеет смысла при объявлении переменных отдельных процедур или функций.

Пример.

```
// Пример объявления одной переменной  
Перем А Экспорт;  
Перем В;  
// Пример объявления нескольких переменных одним оператором  
Перем А, В Экспорт;
```

В языке необязательно объявлять переменные в явном виде. Неявным определением переменной является первое ее появление в левой части оператора присваивания. Тип переменной определяется типом присвоенного ей значения. Не допускается использование в выражениях переменных, не объявленных ранее в явном или неявном виде.

Область использования переменных зависит от места их определения в конфигурации. Существует три области, в которых можно объявить переменные:

- раздел определения переменных программного модуля управляемого приложения - глобальные переменные;
- раздел определения переменных модуля - переменные модуля;

- процедура или функция - локальные переменные.

Глобальные переменные, объявленные с ключевым словом *Экспорт*, доступны для использования в исполняемых операторах, выражениях, в любой процедуре и функции любого клиентского программного модуля конфигурации.

Таким образом, если две переменные с одинаковыми именами используются в двух различных процедурах модуля и имя этой переменной не упоминается как глобальное, то это две различные переменные, локальные для процедур. Если же переменная определена как глобальная переменная, то любое использование имени этой переменной будет приводить к обращению к одной и той же переменной.

Единственный способ создать для процедуры локальную переменную с именем, совпадающим с именем переменной, определенной как глобальная, – это объявить ее явно при помощи оператора *Перем*.

Пока (While)

Оператор цикла *Пока* предназначен для циклического повторения операторов, находящихся внутри конструкции *Цикл – КонецЦикла*. Цикл выполняется, пока логическое выражение равно Истина. Условие выполнения цикла всегда проверяется вначале, перед выполнением цикла.

Синтаксис:

Пока <Логическое выражение> Цикл

// Операторы

[Прервать;]

// Операторы

[Продолжить;]

// Операторы

КонецЦикла

Англоязычный синтаксис:

While <Логическое выражение> Do

// Операторы

[Break;]

// Операторы

[Continue;]

// Операторы

EndDo;

Прервать

Позволяет прервать выполнение цикла в любой точке. После выполнения этого оператора управление передается оператору, следующему за ключевым словом *КонецЦикла*.

Продолжить

Немедленно передает управление в начало цикла, где производится вычисление и проверка условий выполнения цикла. Операторы, следующие в теле цикла за ним, на данной итерации обхода не выполняются.

Пример.

```
ВыборкаДок = Документы.РасходнаяНакладная.Выбрать ();  
    // Цикл по всем документам  
Пока ВыборкаДок.Следующий () Цикл  
    // Отобразим Документ в панели состояния  
    Состояние ("Обрабатывается документ №" + ВыборкаДок.Номер) ;  
    // Операторы выполнения действий над документом  
КонецЦикла;
```

Попытка (Try)

Оператор *Попытка* управляет выполнением программы, основываясь на возникающих при выполнении модуля ошибочных (исключительных) ситуациях, 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 в конструкции *Попытка* произошла ошибка времени выполнения, то выполнение оператора, вызвавшего ошибку, прерывается и управление передается на первый оператор последовательности операторов исключения. При этом управление будет передано даже в том случае, если ошибку вызвал оператор, находящийся в процедуре или функции, вызванной из операторов попытки. 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 исключения управление передается на следующий за ключевым словом *КонецПопытки* оператор. Если же последовательность операторов попытки выполнялась без ошибок, то последовательность операторов исключения будет пропущена и управление также будет продолжено с оператора, следующего за ключевым словом *КонецПопытки*.

Синтаксис:

```
Попытка
    // Операторы попытки
Исключение
    // Операторы исключения
[ВызватьИсключение;]
    // Операторы исключения
КонецПопытки;
```

Англоязычный синтаксис:

```
Try
    // Операторы попытки
Except
    // Операторы исключения
[Raise;]
    // Операторы исключения
EndTry;
```

ВызватьИсключение

Оператор позволяет вызвать исключение в тех случаях, когда, несмотря на отработку исключительной ситуации, необходимо прервать выполнение модуля с ошибкой времени выполнения. Оператор допустим только внутри операторных скобок *Исключение – КонецПопытки*.

Пример.

```
Процедура СформироватьВExcel()
    Попытка
        // Пытаемся обратиться к программе MS Excel
        Табл = Новый ComObject("Excel.Application");
```



```
Исключение
    Предупреждение (ОписаниеОшибки ( ) ) ;
Возврат;
КонецПопытки;
// Операторы формирования отчета
...
КонецПроцедуры
```

Процедура (Procedure)

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

При выполнении оператора *Возврат* процедура заканчивается и возвращает управление в точку вызова. Если в тексте процедуры не встретился оператор *Возврат*, то после выполнения последнего исполняемого оператора происходит выполнение неявного оператора *Возврат*. Конец программной секции процедуры определяется по оператору *КонецПроцедуры*.

Примечание. Ключевые слова *Процедура*, *КонецПроцедуры* являются не операторами, а операторными скобками, поэтому не должны заканчиваться точкой с запятой (это может приводить к ошибкам выполнения модуля).

Синтаксис:

```
Процедура <ИмяПроцедуры>([[Знач] <Парам 1> [=<ДефЗнач>], ... ,[Знач] <Парам N> [=<ДефЗнач>]])[Экспорт]
```

```
    // Объявления локальных переменных;
```

```
    // Операторы;
```

```
    ...
```

```
    [Возврат;]
```

```
    // Операторы;
```

```
    ...
```

```
КонецПроцедуры
```

Англоязычный синтаксис:

```
Procedure <ИмяПроцедуры>([[Val] <Парам 1> [=<ДефЗнач>], ... ,[Val] <Парам N>[=<ДефЗнач>]])[Export]
```

```
    // Объявления локальных переменных;
```

```
    // Операторы;
```

```
    ...
```

```
    [Return;]
```

```
    // Операторы;
```

```
    ...
```

```
EndProcedure
```

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.

=<ДефЗнач>

Необязательная установка значения параметра по умолчанию. Параметры с установленными значениями по умолчанию можно располагать в любом месте списка формальных параметров.

Пример.

```
Перем Глоб;  
// Описание процедуры  
Процедура МояПроцедура(Пар1, Пар2, Пар3) Экспорт  
    Глоб = Глоб + Пар1 + Пар2 + Пар3;  
    Возврат;  
КонецПроцедуры  
Глоб = 123;  
МояПроцедура(5, 6, 7); // Вызов процедуры
```

УдалитьОбработчик (RemoveHandler)

При удалении обработчика события производится проверка соответствия числа параметров события числу параметров метода, назначенного в качестве обработчика.

Синтаксис:

УдалитьОбработчик <Событие>, <ОбработчикСобытия>;

Англоязычный синтаксис:

RemoveHandler <Событие>, <ОбработчикСобытия>;

Событие задается в форме <Выражение>.<ИмяСобытия>, где:

➤ <Выражение> – произвольное выражение на встроенном языке. Его результатом должен быть объект, обработчик события которого удаляется.

➤ <ИмяСобытия> – идентификатор (имя) события.

Обработчиком события может являться метод объекта встроенного языка. Тогда <ОбработчикСобытия> задается как <Выражение>.<ИмяОбработчика>, где:

➤ <Выражение> – произвольное выражение на встроенном языке. Его результатом должен быть объект, метод которого служит обработчиком события.

➤ <ИмяОбработчика> – имя метода обработчика события.

Также в качестве обработчика события может быть задана процедура/функция, находящаяся в области видимости. В этом случае обработчик события задается как имя процедуры/функции.

Пример.

УдалитьОбработчик Накладная.ПриЗаписи, Обработка.ПриЗаписиДокумента;

Функция (Function)

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

Выполнение функции заканчивается оператором *Возврат*. Функции отличаются от процедур только тем, что возвращают *ВозвращаемоеЗначение*. 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.

Если ключевое слово *Возврат* в теле функции не указано или строка модуля, его содержащая, не выполнена, то функция возвращает значение типа *Неопределено*.

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).

Примечание. Ключевые слова *Функция*, *КонецФункции* являются не операторами, а операторными скобками, поэтому не должны заканчиваться точкой с запятой (это может приводить к ошибкам выполнения модуля).

Синтаксис:

```
Функция <ИмяФункции>([[Знач] <Парам 1>[=<ДефЗнач>], ... ,[Знач] <Парам N>[=<ДефЗнач>]])[Экспорт]
```

```
    // Объявления локальных переменных;
```

```
    // Операторы;
```

```
    ...
```

```
    Возврат <Возвращаемое значение>;
```

```
    // Операторы;
```

```
    ...
```

```
КонецФункции
```

Англоязычный синтаксис:

```
Function <ИмяФункции>([[Val] <Парам 1>[=<ДефЗнач>], ... ,[Val] <Парам N>[=<ДефЗнач>]])[Export]
```

```
    // Объявления локальных переменных;
```

```
    // Операторы;
```

```
...  
Return <Возвращаемое значение>;  
// Операторы;
```

```
...  
EndFunction
```

Пример.

```
Перем Глоб;  
// Описание функции  
Функция МояФункция (Пар1, Пар2, Пар3) Экспорт  
    Глоб = Глоб + Пар1 + Пар2 + Пар3;  
    Возврат Глоб;  
КонецФункции  
Глоб = 123;  
Рез = МояФункция (5, 6, 7); // Вызов функции
```

2. 6 Контрольные вопросы

1. Что такое контекст программного модуля?
2. Что такое модуль менеджера?
3. Какие разделы включает структура программного модуля?
4. Какие типы данных относятся к простейшим?
5. Как описать переменную во встроенном языке «1С:Предприятие»?
6. С помощью какого оператора выполняется присваивание переменной нового значения?
7. Какие арифметические операции определены во встроенном языке «1С:Предприятие»?
8. Какие логические операции определены во встроенном языке «1С:Предприятие»?
9. Каков приоритет выполнения операторов встроенного языка «1С:Предприятие»?
10. С помощью какого оператора можно организовать ветвление с использованием встроенного языка «1С:Предприятие»?
11. С помощью каких операторов можно реализовать циклический алгоритм во встроенном языке «1С:Предприятие»?

2.7 Практические задания

1. *Постановка задачи:* создать конфигурацию «Простейший кадровый учет».

В конфигурации потребуется создать следующие объекты:

Справочники:

➤ Должности

Реквизиты: Код (число), Наименование (строка 30 символов)

➤ Сотрудники

Реквизиты: Код (число), Наименование (строка 100 символов), Должность (справочник Должности), Оклад (число, 2 знака после запятой), ДатаПриема (дата), ДатаУвольнения (дата).

Документы:

➤ ПриемНаРаботу

Реквизиты шапки: НомерДок, ДатаДок, ФИО (строка), Должность (справочник Должности), Оклад (число), ДатаПриема (дата).

В результате проведения документа (ОбработкаПроведения()) в справочник Сотрудники должна добавляться запись с информацией о принятом человеке.

```
Процедура ОбработкаПроведения()  
    СпрСотр = СоздатьОбъект("Справочник.Сотрудники");  
    СпрСотр.Новый();  
    СпрСотр.Наименование = ФИО;  
    СпрСотр.Должность = Должность;  
    СпрСотр.Оклад = Оклад;  
    СпрСотр.ДатаПриема = ДатаПриема;  
    СпрСотр.Записать();  
КонецПроцедуры
```

➤ Увольнение

Реквизиты шапки: НомерДок, ДатаДок, Сотрудник (справочник Сотрудники), ДатаУвольнения (дата).

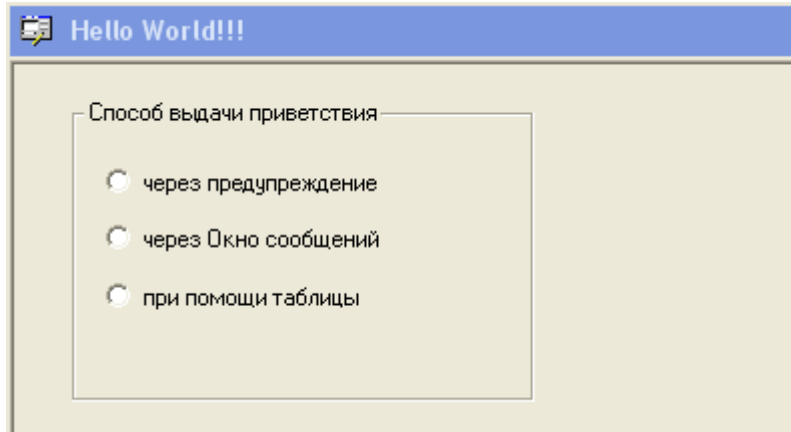
В результате проведения документа в справочнике Сотрудники должен осуществляться поиск человека с соответствующими ФИО и проставляться у него Дата Увольнения.

```
Процедура ОбработкаПроведения()  
    СпрСотр = СоздатьОбъект("Справочник.Сотрудники");  
    СпрСотр.НайтиЭлемент(Сотрудник);  
    СпрСотр.ДатаУвольнения = ДатаУвольнения;  
    СпрСотр.Записать();  
КонецПроцедуры
```

Журналы

➤ Кадровые документы

2. Создать внешнюю обработку «Hello, World», с помощью которой система выводит сообщение одним из выбранных пользователем способов.



Процедура Вывести()

```
Если Способ = 1 Тогда
    Предупреждение("Hello World");
ИначеЕсли способ = 3 Тогда
    Сообщить("Hello World!", "!!!");
ИначеЕсли Способ = 3 Тогда
    Т = СоздатьОбъект("Таблица");
    Т.ИсходнаяТаблица("Таблица");
    Т.ВывестиСекцию("Приветствие");
    Т.ТолькоПросмотр(1);
    Т.Показать("Приветствие при помощи таблицы", "");
Иначе
    Предупреждение("Выберите, пожалуйста, способ "+
        "выдачи приветствия.");
КонецЕсли;
КонецПроцедуры
```

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Гончаров, Д.И. Решение специальных прикладных задач в 1С:Предприятие 8.2 [Текст] / Д.И. Гончаров, Е.Ю. Хрусталева. – М.: 1С-Паблишинг, 2013. - 300с.
2. 1С:Предприятие 8.2. Коротко о главном. Новые возможности версии 8.2 [Текст] / под ред. М.Г. Радченко - СПб.: Питер, 2009. - 416с.
3. Профессиональная разработка в системе 1С:Предприятие 8 [Текст] / [А.П. Габец и др.] – 2-е изд. – М.: 1С-Паблишинг, 2007. - 808с.
4. Радченко, М.Г. 1С:Предприятие 8.3 практическое пособие разработчика. Примеры и типовые решения [Текст] / М.Г. Радченко, Е.Ю. Хрусталева - М.:1С-Паблибшинг», 2013. - 963с.
5. Рыбалка, В.В. HELLO-1С. Пример быстрой разработки приложений на платформе 1С:Предприятие 8.3. [Текст] / В.В. Рыбалка - М.: 1С-Паблибшинг», 2012. - 233с.
6. Хрусталева, Е.Ю. Разработка сложных отчетов в 1С:Предприятии 8. Система компоновки данных [Текст] / Е.Ю. Хрусталева - М.: 1С-Паблишинг, 2012. - 458с.
7. 1С:Предприятие 8. URL: <http://v8.1c.ru> (дата обращения: 11.05.2015)
8. Поддержка пользователей системы 1С:Предприятие 8. URL: <http://users.v8.1c.ru> (дата обращения: 18.05.2015)
9. Информационно-технологическое сопровождение пользователей 1С:Предприятия. URL: <https://its.1c.ru> (дата обращения: 25.05.2015)

Методические материалы

Семенова Ирина Владимировна

**ВВЕДЕНИЕ В КОНФИГУРИРОВАНИЕ
СИСТЕМЫ 1С:ПРЕДПРИЯТИЕ**

Методические указания