

Утверждаю
Директор ОсОО «Ална»
Нестерова А.Р.

Рабочая программа курса
**«Практика
программирования в среде
1С: Предприятие»**

Цель курса. Предлагаемый курс предназначен для обучения слушателей, стремящихся освоить программирование на встроенном языке «1С». Выпускники могут создавать простейшую конфигурацию кадрового учета, а также систематизировать свои знания в области программирования на встроенном языке «1С».

Занятие 1. Основы «1С».

1. Введение в «1С». Общие сведения о программе.
2. «Конфигуратор».
3. Практическая работа.

Занятие 2. Базовые понятия.

1. «Встроенный язык», «Модули», «Глобальный модуль», «Локальный модуль».
2. Практическая работа.

Занятие 3. Базовые понятия.

1. «Агрегатные типы данных», «Атрибуты», «Методы», «Константы», «Периодические константы».
2. Практическая работа.

Занятие 4. Объект: «Справочники».

1. «Справочники», «Реквизиты справочников», «Иерархические (многоуровневые) справочники», «Подчиненные справочники».
2. Практическая работа. Практическое задание.

Занятие 5. Объект: «Документы».

1. «Документы», «Движение документа», «Экранная форма», «Журналы документов».
2. Практическая работа. Практическое задание.

Занятие 6. Конфигурация: «Дерево конфигурации», «Синтакс-Помощник».

1. «Дерево конфигурации», «Синтакс-Помощник».
2. Практическая работа. Практическое задание.

Занятие 7. Конфигурация: «Редактор форм».

1. «Редактор форм».
2. Практическая работа. Практическое задание.

Занятие 8. Конфигурация: «Редактор программных модулей».

1. «Редактор программных модулей».
2. Практическая работа.

Занятие 9. Формат программного модуля.

1. Формат программного модуля, «Комментарий», «Оператор», «Операторы объявления переменных», «Исполняемые операторы».
2. Практическая работа.

Занятие 10. Имена переменных, процедур и функций.

1. Имена переменных, процедур и функций, «Зарезервированные слова».
2. Практическая работа. Практическое задание.

Занятие 11. Структура программного модуля.

1. Структура программного модуля, «Раздел определения переменных», «Раздел процедур и функций», «Раздел основной программы».
2. Практическая работа. Практическое задание.

Занятие 12. Специальные символы.

1. Специальные символы, используемые в исходном тексте.
2. Практическая работа. Практическое задание.

Занятие 13. Процедуры и функции.

1. Процедуры и функции программного модуля, «Русскоязычный Синтаксис описания процедуры и функции», «Англоязычный Синтаксис описания процедуры и функции», «Параметры».
2. Практическая работа. Практическое задание.

Занятие 14. Передача параметров.

1. Передача параметров.
2. Практическая работа. Практическое задание

Занятие 15. Основы программирования.

1. Практическая работа. Практическое задание.

Занятие 1. Основы «1С:Предприятия».

1. Введение в 1С:Предприятия. Общие сведения о программе.
2. «Конфигуратор».
3. Практическая работа.

Введение в 1С:Предприятия. Общие сведения о программе.

«1С:Предприятие» - это система программ, включающая в себя технологическую платформу и прикладные решения (конфигурации), созданные с ее помощью и функционирующие только при наличии платформы.

Обычно конечный пользователь работает с типовой конфигурацией от фирмы «1С» - это «1С:Бухгалтерия», «1С:Торговля и Склад», «1С:Зарплата и Кадры», созданные самой фирмой «1С». Однако любой человек или фирма может создать собственную конфигурацию для удовлетворения индивидуальных потребностей, так как средство разработки «Конфигуратор» поставляется вместе с «1С:Предприятием». Система «1С:Предприятие» предоставляет в распоряжение разработчика широкий набор объектов, на основе которых создаются все прикладные решения. К таким объектам относятся справочники, документы, регистры и т.д. Кроме того «1С:Предприятие» имеет встроенный язык программирования, механизм запросов, различные визуальные редакторы и конструкторы.

«Конфигуратор».

Разработка и корректировка любого прикладного решения начинается с запуска «Конфигуратора». Программы, разрабатываемые в «1С» называются "конфигурациями". В «Конфигураторе» программист-разработчик создает или корректирует структуру базы данных, программные модули, производит администрирование работы, включая загрузку базы данных из архивов. Пользователи программ работают в режиме «Предприятие». Вообще «1С» имеет несколько режимов запуска: Предприятие, Конфигуратор, Отладчик, Монитор.

Существует два основных правила:

- 1) в режиме «Конфигуратор» определяется структура базы данных и пишутся программы;
- 2) в режиме «Предприятие» используются написанные программы и вводятся данные.

Основное понятие «1С» - это объект экономической деятельности предприятия: документ, отчет, справочник и т.д. Конфигурация «1С» представляет собой совокупность взаимосвязанных между собой объектов.

Занятие 1. Практическая работа.

Простейший кадровый учет.

1.

Каждая конфигурация должна храниться в отдельном каталоге.

1. Создайте новую папку на любом диске, например, «D:\Basa\KadrUch».
2. Запустите «1С:Предприятие». Откроется список информационных баз.
3. Нажмите кнопку «Добавить». Укажите имя информационной базы, например, «Простейший Кадровый Учет».
4. В следующем поле укажите путь к папке, которую вы создали на 1-м шаге. Нажмите кнопку «...», войдите в папку и нажмите «Выбрать». У вас должны быть заданы полные права на эту папку (чтение, запись и т.д.).
5. Нажмите «ОК». В списке информационных баз появится новая строка.

Занятие 2. Базовые понятия.

1. «Встроенный язык», «Модули», «Глобальный модуль», «Локальный модуль».
2. Практическая работа.

«Встроенный язык», «Модули», «Глобальный модуль», «Локальный модуль».

Поведение объектов конфигурации задается на «Встроенном языке». Программист может написать свои собственные модули, состоящие из процедур и функций. С помощью языка определяются алгоритмы поведения документов, алгоритмы формирования отчетов и многие другие действия программы. Встроенный язык позволяет работать со справочниками, документами и другими объектами. Особенность встроенного языка «1С» в том, что все ключевые слова можно писать по-русски.

Программный код конфигурации организован в виде «Модулей». Это могут быть модули документов, модули отчетов, или другие модули. Модуль – это часть программы. Разбиение программы на модули позволяет справиться со сложностью задачи.


В любой конфигурации «1С» существует «Глобальный модуль». Его особенность в том, что он относится к конфигурации в целом, а не конкретному объекту метаданных. Глобальный модуль нужен для того, чтобы писать процедуры и функции, доступные в любом месте программы (в любом другом модуле). Для этого необходимо просто указать ключевое слово «Экспорт» в заголовке процедуры или функции. Точно также можно создать глобальные переменные, которыми можно будет пользоваться в любом месте программы.

Помимо глобального модуля, конфигурация системы имеет «Локальные модули». «Локальный модуль» описывает поведение конкретного объекта конфигурации и использует в своей работе только те данные, включая объекты, которые доступны этому объекту.

Занятие 2. Практическая работа.

2.

Запуск «Конфигуратора».

1. Укажите режим запуска «Конфигуратор».
2. Нажмите «ОК». Для запуска в режиме «Конфигуратор».
3. Так как в созданной папке ничего нет, то система спросит, в каком формате будет храниться база данных: в файлах DBF или на SQL-Server. Укажите формат «Файлы *.DBF, *.CDX» и нажмите ОК.
4. Запустится «Конфигуратор». Откройте дерево конфигурации: «Конфигурация» => «Открыть конфигурацию» или кнопка . Появится окно дерева конфигурации. Задайте ему нормальные размеры. В этом окне представлены все объекты конфигурации, такие как справочники, документы, отчеты и т.д.

3.

Поменяем имя нашей задачи:

1. Установите курсор на строку «Задача», дважды щелкните по ней.
2. Откроется окно «Задача». Задайте идентификатор – «KadrUch».
3. Нажмите ОК.

Занятие 3. Базовые понятия.

1. «Агрегатный тип данных», «Атрибуты», «Методы», «Константы», «Периодические константы».
2. Практическая работа.

«Агрегатный тип данных», «Атрибуты», «Методы», «Константы», «Периодические константы».

«Агрегатный тип данных» («Объект») — это специализированные данные системы «1С», отражающие объекты деятельности предприятия и способы работы с ними. Каждый «Объект», как правило, имеет набор «атрибутов» и «методов».

«Атрибуты» (свойства, реквизиты) по свойствам напоминают переменные, т. е. им можно присваивать или читать из них значения. Конкретные значения «Атрибутов» могут быть определены двумя способами: 1) некоторые значения заранее известны в из его глобального и локального модуля; 2) другие значения могут быть определены с помощью системной функции «СоздатьОбъект».

«Методы» (процедуры и функции) — это те действия, которые может выполнять «Объект». Методы могут иметь или не иметь возвращаемое значение. Если метод имеет возвращаемое значение, то он может размещаться в правой части оператора присваивания, в выражениях, в описании фактических параметров других вызываемых методов, процедур или функций.

«Константы» - очень простой для понимания «Объект». Константа позволяет хранить одно-единственное значение какого-либо типа. Например, число, строку, элемент справочника, значение перечисления и т.д. Эти значения относятся в целом к конфигурации и являются общими для всех пользователей, работающих с данной информационной базой. Значение константы обычно указывается интерактивно через меню "Константы" -> "Операции", но константа может быть установлена программно. Для этого предназначено ключевое слово "Константа" во встроенном языке. Константы могут быть числового типа, строкового или типа дата.


Бывают также «Периодические константы», которые хранят историю изменения своего значения. При обращении с такой константой всегда нужно указывать дату считывания или дату установки нового значения.

Занятие 3. Практическая работа.

Простейший кадровый учет.

4.

Создание справочника.

1. Установите курсор на строку «Справочники», щелкните правой кнопкой мыши и выберите «Новый справочник». Откроется окно «Конструктор справочников». Конструкторы предназначены для помощи программисту в выполнении определенных действий.
2. Зайдите в идентификатор справочника (имя). Например, создадим справочник «Должности», это будет имя справочника.
3. Перейдите по кнопке «Далее» к следующему шагу, еще раз «Далее» и «Готово».
4. Закройте окно «Справочника Должности».
5. На всякий случай сохраните конфигурацию . Система 1С сообщит вам, что выполнит анализ изменений. Ответьте «Да». Затем система 1С обнаружит, что конфигурация абсолютно новая, и у нее нет даже словаря данных. Отвечайте «Да» – пусть создает этот словарь.
6. Так как мы редактировали структуру информационной базы, то система попросит разрешение на реорганизацию информационной базы. Нажимаем «Принять» и «ОК».

Занятие 4. Объект: «Справочники».

1. «Справочники», «Реквизиты справочников», «Иерархические (многоуровневые) справочники», «Подчиненные справочники».
2. Практическая работа. Практическое задание.

«Справочники», «Реквизиты справочников», «Иерархические (многоуровневые) справочники», «Подчиненные справочники».

«Справочник» – это список определенной информации, а именно список возможных значений того или иного реквизита. Справочники используются в тех случаях, когда необходимо исключить неоднозначный ввод информации. В справочник помещаются данные однородных объектов. Каждый такой объект называется элементом справочника. В конфигураторе создается не сам справочник, как список значений, а разрабатывается структура справочника, его шаблон. В процессе конфигурирования описывается структура информации, разрабатывается экранное и, если необходимо, печатное представление справочника, задаются различные особенности его «поведения».

В качестве обязательных атрибутов каждый справочник имеет "Код" и "Наименование". Это и есть «Реквизиты справочников». Код элемента справочника может быть как числовым, так и текстовым. Система «1С» предоставляет широкие возможности по работе с кодами элементов справочника: автоматическое присвоение кодов, автоматический контроль уникальности кода и другие. Разработчик может регулировать длину наименования справочника в определенных пределах. Если же наименование элементов предполагается слишком длинное, то можно не использовать наименование, а завести необязательный реквизит справочника типа "Строка" неограниченной длины. В справочнике может храниться любая дополнительная информация об элементе справочника. Для хранения такой информации создается список реквизитов. Справочник может иметь примитивные данные: число, строка, дата, но можно указывать также и сложные (агрегатные) данные. Например, реквизит "Должность" будет иметь тип данных "Справочник.Должности". Так реализуется простейшая связь между справочниками, когда значения реквизитов одного справочника выбираются из другого справочника.

Список элементов справочника в системе 1С может быть многоуровневым. В этом случае все строки справочника будут разделяться на два вида: "элементы" и "группы элементов". Группы позволяют переходить на нижележащие уровни многоуровневого справочника. И элементы, и группы элементов, в многоуровневом справочнике можно переносить из одной группы в другую. Глубина вложенности иерархического справочника не может превышать 10.

Между справочниками может существовать еще связь «Подчиненных справочников», когда элементы одного справочника принадлежат элементам другого. Например, в системе может быть справочник "Налоговые льготы". Если его сделать подчиненным справочнику "Сотрудники", то это будет означать, что "сотрудник владеет налоговыми льготами". В терминах реляционных баз данных, между таблицами устанавливается связь "один-ко-многим".

Занятие 4. Практическая работа. Практическое задание.

Простейший кадровый учет.


5.

Задание по работе со «Справочником»:

1. Создайте справочник «Сотрудники».
2. В этом справочнике будет дополнительный реквизит «Должность». Тип реквизита будет «Справочник.Должности», то есть должна быть ссылка на справочник «Должности», который мы создали раньше.
3. Создайте реквизит «Оклад» (тип «Число» и точность 2 знака после запятой).
4. Создайте реквизиты «ДатаПриема» и «ДатаУвольнения» (оба типа «Дата»).
5. Сохраните конфигурацию (на всякий случай).

6.

Теперь у нас есть два справочника и можно ввести некоторые данные.

1. Запустите конфигурацию в режиме «Предприятие». Это можно сделать прямо из «Конструктора», нажав кнопку . Появится окно «1С:Предприятие - KadrUch».
2. Откройте справочник «Должности». Для этого щелкните меню «Операции => Справочники => Должности => ОК». Откроется окно справочника.
3. Введите несколько должностей, например, «Директор», «Бухгалтер», «Программист».
4. Закройте справочник «Должности» и откройте справочник «Сотрудники».
5. Заведите несколько сотрудников. Например, «Иванов» - директор, «Петров» - бухгалтер, «Сидоров» - программист.

Занятие 5. Объект: «Документы».

1. «Документы», «Движение документа», «Экранная форма», «Журналы документов».
2. Практическая работа. Практическое задание.

«Документы», «Движение документа», «Экранная форма», «Журналы документов».

Документы — это электронные аналоги обычных бумажных документов. С помощью документов можно вводить информацию в базу данных, например, принимать нового сотрудника документом "Приказ о приеме на работу". У любого документа в системе "1С" есть уже два предопределенных атрибута: "НомерДок" и "ДатаДок". Программист может добавлять свои реквизиты документа в режиме "Конфигуратор".

Документы формируют движения. Движения документа — это порожденные им изменения в справочниках и других объектах конфигурации (регистрах, журнале проводок, журналах расчетов). Иначе движение документа называют проведением документа. С точки зрения программиста, при проведении документа произойдет вызов процедуры "ОбработкаПроведения" из модуля документа, в котором определяется движение документа с помощью встроенного языка "1С". На языке записывается, что документ делает в системе, как он формирует бумажный бланк, и как он себя ведет на экране.

У документа есть экранная форма — «диалог». Именно его видит пользователь на экране. Печатная форма документа обычно "таблица".

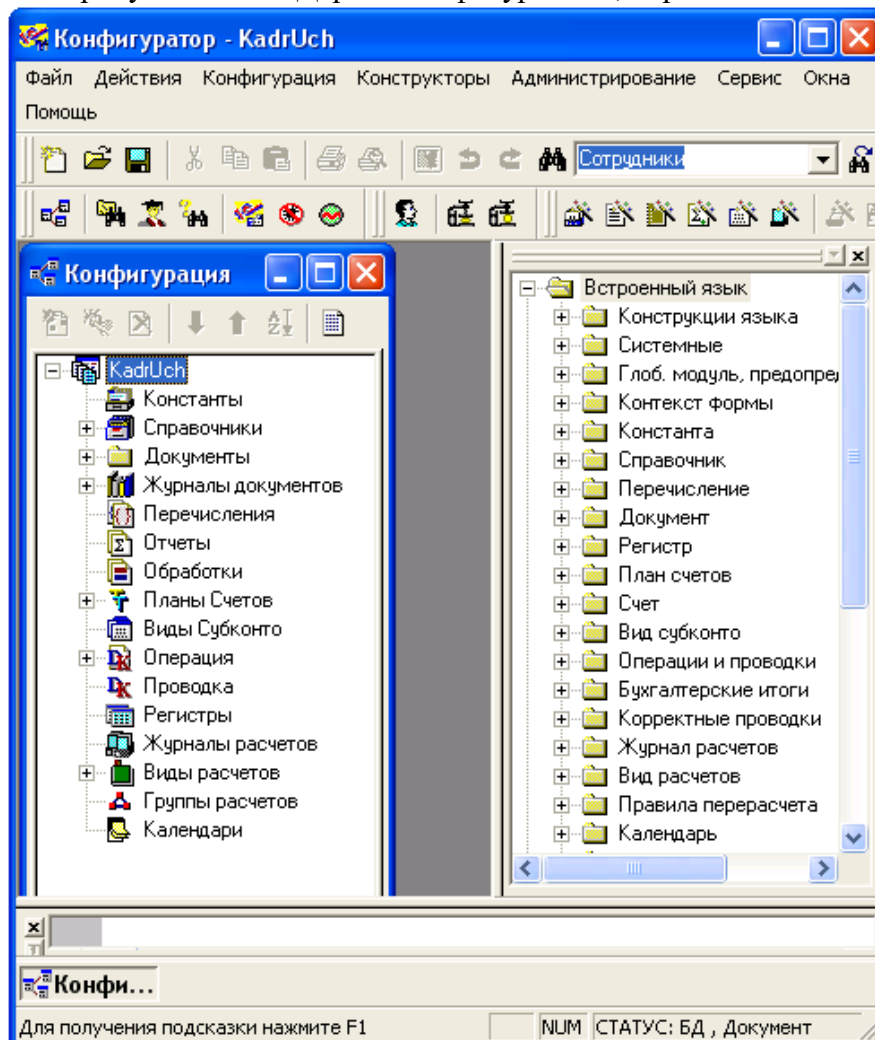
Все документы в "1С" хранятся в журналах. На экране журнал выглядит как таблица, строки которой являются документами. Журналы в системе "1С" могут быть следующих видов: Обычный, Дополнительный, (Общий) и (Прочие). У журнала документов есть экранная форма, где можно настраивать отображаемые колонки. Можно также создавать дополнительные колонки, которые будут отображать информацию из документов.


Занятие 6. Конфигурация: «Дерево конфигурации», «Синтакс-Помощник».


1. «Дерево конфигурации».
2. Практическая работа.

«Дерево конфигурации». «Синтакс-Помощник».

На рисунке слева «Дерево конфигурации», справа «Синтакс-Помощник».




Кнопка	Описание
	«Новый элемент (Ins)»
	«Редактировать»
	«Удалить (Del)»
	«Переместить вниз»
	«Переместить вверх»
	«Сортировать»
	«Редактировать описание (Ctrl+T)»

Войти в «Дерево конфигурации» можно по кнопке .

Во вкладке «Метаданные» «Дерева конфигурации» можно производить различные действия над объектами, предусмотренные кнопками слева.

Во вкладке "Интерфейсы" можно создавать и редактировать меню пользователя, а также тестировать его.

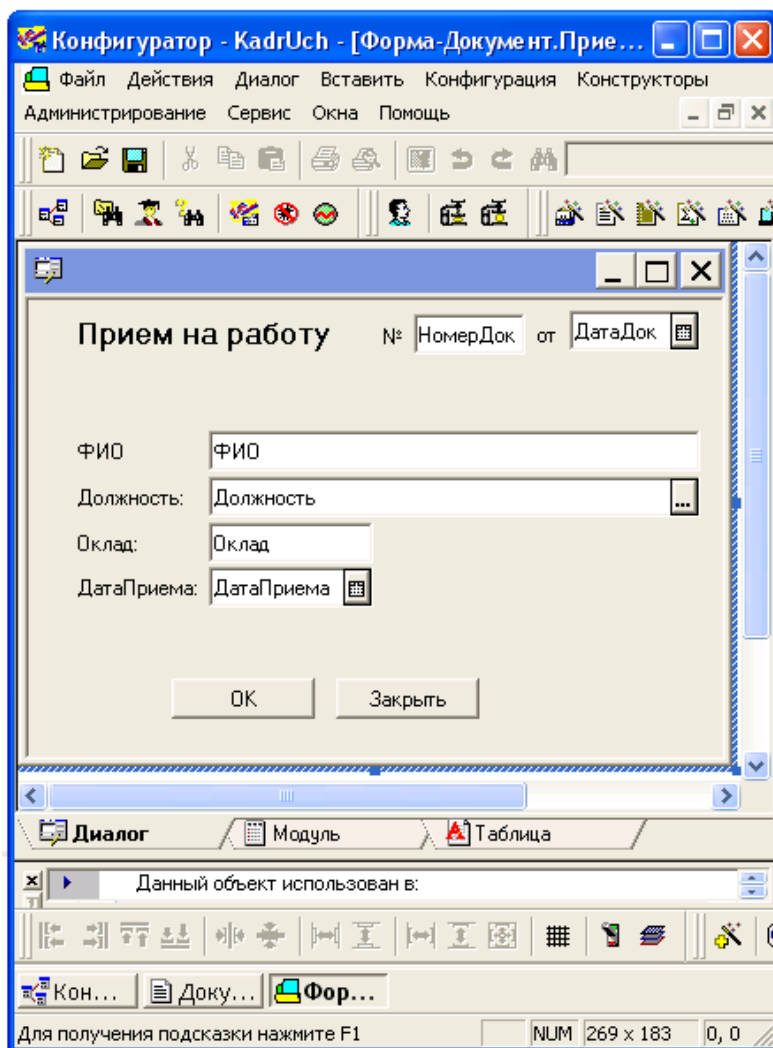
Во вкладке «Права» – определять доступ к объекту.

Войти в «Синтакс-помощник» можно по кнопке . «Синтакс-помощник» часто используется при написании программных модулей для того, чтобы вспомнить синтаксис или параметры конкретной процедуры/функции.

Занятие 7. Конфигурация: «Редактор форм».

1. «Редактор форм».
2. Практическая работа. Практическое задание.

«Редактор форм».















У многих объектов конфигурации есть экранные формы. В частности экранные формы имеют справочники, документы, журналы документов, отчеты, обработки, журналы расчетов, журнал операций, журнал проводок, план счетов и счет. У некоторых объектов может быть несколько экранных форм. Например, у справочников есть формы списков, форма элемента и форма группы.

Экранная форма редактируется при помощи следующих кнопок «Редактора диалога»:

Кнопка	Описание
	Эти кнопки становятся доступными, когда выделено несколько элементов управления на форме. Они позволяют выравнивать их по левому, по правому, по верхнему и по нижнему краю.
	Эти кнопки центрируют выделенные элементы управления относительно всей формы по вертикали или горизонтали.
	Эти кнопки равномерно распределяют выделенные элементы управления по вертикали или горизонтали.
	Эти кнопки делают выделенные элементы управления одной ширины, одной высоты или одного размера. Определяющим является элемент управления, выделенный последним. Два элемента управления можно выделить, удерживая кнопку "Shift" и щелкая поочередно на каждом из них.
	Это вспомогательные кнопки, обеспечивающие следующие функции: <ul style="list-style-type: none"> • Включить/выключить решетку (сетку). • Запустить тестовое окно. • Вызвать окно редактирования слоев формы.

Кроме того имеются кнопки «Элементов диалога»:

Кнопка	Описание
	Эта кнопка вызывает что-то типа мастера элементов диалога, позволяющий разместить элемент управления, назначить тип, присвоить вызов процедуры и т.д.
	Эта кнопка размещает на форме обычную кнопку. В свойствах кнопки в поле "Формула" пишется имя процедуры, которая будет запущена при нажатии на кнопку.
	Эта кнопка размещает на форме флажок. В свойствах флажка в поле формула пишется имя процедуры, которая будет запущена при установке или снятии флажка.
	Эта кнопка размещает на поле переключатель. Переключатели имеют смысл, когда их несколько, в этом случае пользователь производит выбор между несколькими вариантами. Чтобы переключатели заработали, нужно сделать следующее: для первого переключателя установить в свойствах флажок "Первый в группе". Далее проверить, чтобы порядок обхода элементов был точно в том порядке, в котором следуют переключатели (меню "Действия => Настройка порядка обхода").
	Эта кнопка размещает на форме список значений. При этом по имени списка можно обращаться к нему из модуля (объект "СписокЗначений"). Можно добавлять значения в список, удалять значения и делать все другие операции, возможные для объекта "СписокЗначений".
	Эта кнопка размещает на форме поле со списком значений. При этом также создается объект "СписокЗначений". Обратите внимание, что лучше сразу задать <i>большую высоту</i> этого элемента управления, чтобы список раскрывался.
	Обычная рамка. Можно визуально объединить несколько элементов управления. Служит для удобства пользователя.
	Текстовое поле. Если в поле "Формула" вписать имя функции, то при каждом обновлении формы система будет вызывать вашу функцию, и ее результат будет отображаться в этом поле. Кстати, если щелкнуть на эту кнопку и затем щелкнуть на табличной части справочника, документа или журнала, мы создадим текстовую колонку. Она не будет сохраняться в базе данных, но у нее есть поле "Формула", куда можно написать любое выражение. Очень полезная вещь для различных информационных колонок.
	Поле для ввода данных. Для поля можно назначить любой тип данных, допустимый в "1С:Предприятии". В поле "Формула" можно написать имя процедуры, которая будет вызываться при изменении значения данного поля.
	Эта кнопка размещает на форме таблицу значений. По имени таблицы значений к ней можно обращаться из модуля. Колонки и строки таблицы создаются программным способом.
	Для иерархических справочников на форме списка можно разместить дерево групп.
	Эта кнопка размещает на форме картинку. Управление картинкой осуществляется программным способом.

Занятие 5,6,7. Практическая работа. Практическое задание.

Простейший кадровый учет.

7.

В реальной жизни принятие сотрудника на работу производится кадровым документом «Приказ о приеме на работу», чем сейчас мы и займемся.

1. В конфигураторе создайте новый документ «Прием на работу».
2. Задайте имя документа «ПриемНаРаботу».
3. Добавьте реквизит шапки «ФИО».
4. Укажите ему тип значения «Строка».
5. Задайте длину 100 символов.
6. Нажмите ОК.

8.

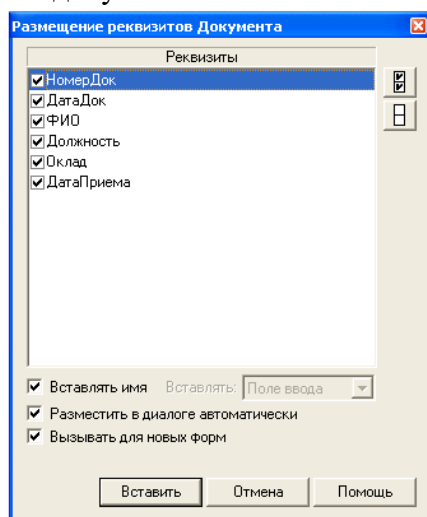
Аналогичным образом создайте остальные реквизиты шапки:

«Должность» (тип «Справочник.Должности»),
«Оклад» (тип «Число 10.2»),
«ДатаПриема», (тип «дата»).

9.

У любого документа должна быть удобная форма для ввода данных.

1. Нажмите кнопку «Форма» в нижней части окна «Документ ПриемНаРаботу». Так как мы сюда вошли в первый раз, система предлагает автоматически создать форму документа.



2. Зададим «Вставить».

10.

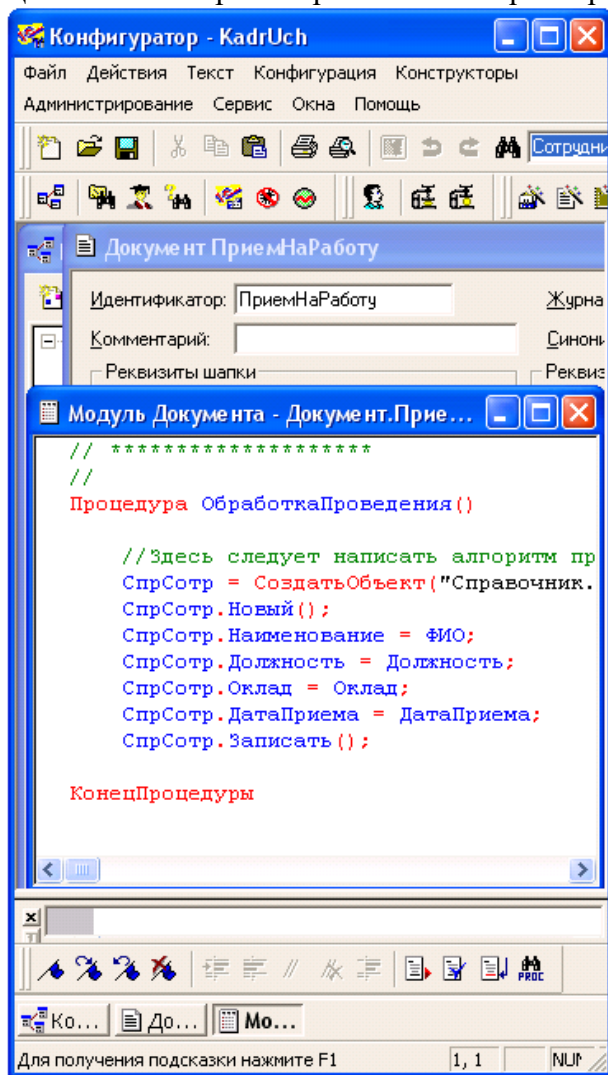
Задание: Приведите документ к следующему виду:

Занятие 8. Конфигурация: «Редактор программных модулей».

3. «Редактор программных модулей».
4. Практическая работа.






«Редактор программных модулей».

У каждой экранной формы есть программный модуль. Кроме того, есть программные модули документов и глобальный модуль. В этих модулях располагается программа на встроенном языке "1С". В программном модуле имеется синтаксическая подсветка. Разными цветами отображаются ключевые слова, переменные, комментарии и значения. Цветовая палитра настраивается в параметрах.



При редактировании модуля будут также полезны верхние кнопки. Там можно найти кнопки поиска, вызова "Синтакс-помощника" и другие.

Здесь даны значения кнопок панели «Редактора программных модулей»:

Кнопка	Описание
	По тексту программы можно расставить закладки или пометки (marks) и перемещаться по ним. Полезная вещь для больших модулей в сотни и тысячи строк.
	Кнопки сдвига блока текста вправо и влево. Становятся активными, когда выделена одна строка (или несколько строчек).
	Закомментировать / раскомментировать блок текста.
	Кнопка автоформатирования текста программы. Нажимайте ее чаще и ваши программы будут красивыми! К сожалению, это не избавит программу от всех ошибок, но их определенно станет меньше!
	Служебные функции для работы с модулем: <ul style="list-style-type: none"> • Открыть модуль в отладчике. • Проверить модуль на синтаксические ошибки. • Перейти к определенной строке модуля. • Открыть окно со списком процедур и функций модуля.

Занятие 9. Формат программного модуля.

1. Формат программного модуля, «Комментарий», «Оператор», «Операторы объявления переменных», «Исполняемые операторы».
2. Практическая работа.

Формат программного модуля, «Комментарий», «Оператор», «Операторы объявления переменных», «Исполняемые операторы».

Исходный текст программного модуля может состоять из операторов и комментариев.

«Комментарий» используется для размещения в исходном тексте пояснений к работе модуля. В режиме исполнения модуля комментарии пропускаются. В исходном тексте программного модуля комментарий начинается парой символов "//" и заканчивается концом строки.

«Операторы» имеют вид стандартного обращения к процедуре, за исключением оператора присваивания (A=B;) и управляющих конструкций (таких как Для, Пока, Если). Между собой операторы разделяются символом ";" (точкой с запятой). Операторы языка в программном модуле можно подразделить на две категории: 1) операторы объявления переменных и 2) исполняемые операторы.

«Операторы объявления переменных» создают имена переменных, которыми манипулируют 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];

Особенность метки в том, что она начинается с символа '~' (тильда) и отделяется от оператора символом ':' (двоеточие).

Занятие 9. Практическая работа. Практическое задание.

Занятие 8,9. Практическая работа.

Простейший кадровый учет.

11.

Требуется настроить этот документ на выполнение действий по принятию сотрудника на работу. Действия документа закладываются в predetermined procedure «ОбработкаПроведения», которая хранится в «модуле документа».

1. Закройте экранную форму документа. На экране останется окно структуры документа.
2. Внизу есть кнопка «Модуль документа». Нажмите ее.
3. Откроется окно для редактирования текста программы, которая пока еще пустая. Напишите следующий текст:
СпрСотр = СоздатьОбъект("Справочник.Сотрудники");
СпрСотр.Новый();
СпрСотр.Наименование = ФИО;
СпрСотр.Должность = Должность;
СпрСотр.Оклад = Оклад;
СпрСотр.ДатаПриема = ДатаПриема;
СпрСотр.Записать();
4. Закройте окно модуля и окно структуры документа.
5. Сохраните конфигурацию.

12.

Все документы хранятся в журналах документов. Для документа «ПринятиеНаРаботу» был задан журнал по умолчанию «(Прочие)».

1. Запустите «1С:Предприятие» в режиме «Предприятие».
2. Для открытия журнала документов «(Прочие)» щелкните меню «Операции => Журналы документов => Прочие».
3. Нажмите Insert. Появится окошко «Выбор вида документа» с пока еще единственным документом «ПриемНаРаботу».
4. Жмем «ОК». Откроется форма документов «Прием на работу» для ввода данных. Номер документа присвоится автоматически, дата документа – сегодняшнее число.
5. Введите все свои данные:
ФИО – Петровский Иван Александрович
Должность – Программист
Оклад – 1000.00
Дата приема – 28.08.09
6. Нажмите кнопку «ОК».
7. Сначала документ будет записан, для чего потребуется ответить на вопрос «Сохранить документ?» – «Да»
8. А потом проведен при ответе на вопрос «Провести документ?» – «Да». При проведении сработает процедура «ОбработкаПроведения».

В журнале документов «Прочие» появиться новая строка, извещающая о приеме на работу сотрудника, а в справочнике «Сотрудники» добавиться новый сотрудник. Проверьте, так ли это?

Занятие 10. Имена переменных, процедур и функций.

1. Имена переменных, процедур и функций, «Зарезервированные слова».
2. Практическая работа. Практическое задание.

Имена переменных, процедур и функций, «Зарезервированные слова».

Именем переменной, объявленной процедуры или функции, может быть любая последовательность букв, цифр и знаков подчеркивания "_", начинающаяся с буквы или знака подчеркивания "_". Создаваемые имена не должны совпадать с зарезервированными словами языка или именами существующих процедур и функций.

«Зарезервированными слова» не могут использоваться в качестве создаваемых имен переменных и объявляемых процедур и функций. В языке «1С» каждое из ключевых слов имеет два представления — русское и английское. Ключевые слова в русском и английском представлении могут свободно смешиваться в одном исходном тексте. Регистр букв ключевых слов не имеет значения. Ниже приведен список ключевых слов в обоих вариантах представления.

Если	If	Не	Not	Дата	Date
Тогда	Then	Знач	Val	Формат	Format
ИначеЕсли	Elsif	СтрДлина	StrLen	Разм	Dim
Иначе	Else	СокрЛ	TrimL	Вопрос	Do Query Box
КонецЕсли	EndIf	СокрП	TrimR	Контекст	Context
Цикл	Do	Лев	Left	Перем	Var
Для	For	Прав	Right	Перейти	Goto
По	To	Сред	Mid	Возврат	Return
Пока	While	Цел	Int	Продолжить	Continue
Функция	Function	Окр	Round	Прервать	Break
КонецПроцедуры	EndProcedure	Число	Number	И	And
КонецФункции	EndFunction	Строка	String	Или	Or
Предупреждение	DoMessageBox	КонецЦикла	EndDo	Процедура	Procedure
Если	If	Не	Not	Дата	Date

Занятие 10. Практическая работа. Практическое задание.

Занятие 11. Структура программного модуля.

1. Структура программного модуля, «Раздел определения переменных», «Раздел процедур и функций», «Раздел основной программы».
2. Практическая работа. Практическое задание.

Структура программного модуля, «Раздел определения переменных», «Раздел процедур и функций», «Раздел основной программы».

Структуру программного модуля можно подразделить на следующие разделы: 1) раздел определения переменных; 2) раздел процедур и функций; 3) раздел основной программы. В программном модуле любой из разделов может отсутствовать.

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

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

«Раздел основной программы» размещается от первого исполняемого оператора вне тела процедур или функций до конца модуля. В этом разделе могут находиться только исполняемые операторы. Раздел основной программы выполняется в момент запуска модуля на выполнение. 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.

Занятие 11. Практическая работа. Практическое задание.

Занятие 12. Специальные символы.

1. Специальные символы, используемые в исходном тексте.
2. Практическая работа. Практическое задание.

Специальные символы, используемые в исходном тексте.

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

Занятие 12. Практическая работа. Практическое задание.

Занятие 13. Процедуры и функции.

1. Процедуры и функции программного модуля, «Русскоязычный Синтаксис описания процедуры и функции», «Англоязычный Синтаксис описания процедуры и функции», «Параметры».
2. Практическая работа. Практическое задание.

Процедуры и функции программного модуля, «Русскоязычный Синтаксис описания процедуры и функции», «Англоязычный Синтаксис описания процедуры и функции», «Параметры», «Описание».

«Русскоязычный Синтаксис описания процедуры и функции»:
{Процедура|Функция}<Имя_конс>([[Знач] <Парам1>[=<ДефЗнач>],
... , [Знач] <ПарамN>[=<ДефЗнач>]]) [Экспорт]
//Объявления локальных переменных;
//Операторы;
...
[Возврат [<ВозвращаемоеЗначение >];]
//Операторы;
...

{КонецПроцедуры|КонецФункции}

Внимание! Здесь строчка **[Возврат [<ВозвращаемоеЗначение >];]** является необязательной для процедуры и обязательной для функции.

«Англоязычный Синтаксис описания процедуры и функции»:
{Procedure|Function}<Имя_конс>([[Val] <Парам1>[=<ДефЗнач>],
... , [Val] <ПарамN>[=<ДефЗнач>]]) [Export]
//Объявления локальных переменных;
//Операторы;
...
[Return [<ВозвращаемоеЗначение >];]
//Операторы;
...

{EndProcedure|EndFunction}

Внимание! Здесь строчка **[Return [<ВозвращаемоеЗначение >];]** является необязательной для процедуры и обязательной для функции.

«Параметры»:

<Имя_конс>
Знач

Назначает имя конструкции (процедуры или функции)
Необязательное ключевое слово, указывающее на то, что следующий за ним параметр передается по значению. Если это ключевое слово не указано, то параметр конструкции передается по ссылке.

<Парам1>, . . . ,
<ПарамN>

Необязательный список формальных параметров, разделяемых запятыми. Значения формальных параметров должны соответствовать значениям передаваемых при вызове конструкции фактических параметров. Здесь определяется имя каждого из параметров. Список формальных параметров может быть пуст.

=<ДефЗнач>

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

Если параметр при вызове конструкции опущен, то он

принимает, либо установленное по умолчанию значение (если оно есть), либо принимает «пустое» значение (значение неопределенного типа).

Если параметру не задано значение по умолчанию и, он является последним в списке передаваемых параметров, то при вызове конструкции его нельзя опускать.

Если параметру задано значение по умолчанию и, он является последним в списке, то при вызове конструкции его можно опускать в списке передаваемых фактических параметров и не ставить запятую перед опущенным параметром.

Если параметру не задано значения по умолчанию, то при вызове конструкции его можно опускать в списке передаваемых фактических параметров, но разделительную запятую надо ставить.

Экспорт

Необязательное ключевое слово, указывающее на то, что данная конструкция является доступной из других программных модулей. Имеет смысл только в *глобальном программном модуле*.

//Объявления локальных переменных

Объявляются локальные переменные, на которые можно сослаться только в рамках этой конструкции.

//Операторы

Исполняемые операторы конструкции.

Возврат

Необязательное ключевое слово для процедуры, которое завершает выполнение процедуры и осуществляет возврат в точку программы, из которой было обращение к процедуре.

Обязательное ключевое слово для функции, которое завершает выполнение функции и возвращает ВозвращаемоеЗначение в выражение, в котором используется функция.

<ВозвращаемоеЗначение>

Отсутствует для процедуры.

Необязательное выражение для функции, значение которого содержит результат обращения к функции.

EndProcedure/EndFunction

Обязательное ключевое слово, обозначающее конец исходного текста конструкции, завершение выполнения конструкции. Возврат в точку, из которой было обращение к конструкции.

Ключевое слово "*Процедура*" или "*Функция*" начинает секцию исходного текста, выполнение которого можно инициировать из любой точки программного модуля, просто указав *Имя_конс* со списком параметров (если параметры не передаются, то круглые скобки, тем не менее, обязательны). Конец программной секции процедуры определяется по ключевым словам "*EndProcedure*" или "*EndFunction*". Переменные, объявленные в теле конструкции в разделе "*Объявления локальных переменных*", являются локальными переменными данной конструкции, поэтому доступны только в этой конструкции.

Замечание: ключевые слова "*Процедура*" – "*КонецПроцедуры*" и "*Функция*" – "*КонецФункции*" являются не операторами, а операторными скобками, поэтому не должны заканчиваться точкой с запятой.

Занятие 13. Практическая работа. Практическое задание.

Занятие 14. Передача параметров.

1. Передача параметров.
2. Практическая работа. Практическое задание

Передача параметров

По умолчанию параметры методов, процедур и функций передаются по ссылке, то есть изменение внутри процедуры или функции значения формального параметра ведет к изменению значения соответствующего фактического параметра. При передаче параметра по значению изменение значения формального параметра никак не влияет на фактический параметр вызова процедуры. Для указания того, что тот или иной параметр следует передавать по значению, следует в исходном тексте процедуры или функции перед именем параметра записать ключевое слово `Знач` (англоязычный синоним — `Val`).

Если параметру задано значение по умолчанию и он является последним в списке, то при вызове процедуры его можно опускать в списке передаваемых фактических параметров и не ставить запятую перед опущенным параметром.

Если параметру не задано значения по умолчанию, то при вызове процедуры его можно опускать в списке передаваемых фактических параметров, но разделительную запятую надо ставить.

Если параметр при вызове процедуры опущен, то он принимает либо установленное по умолчанию значение (если оно есть) либо принимает «пустое» значение (значение неопределенного типа).

Если при вызове метода, процедуры или функции параметры не передаются (пустой список параметров), то, тем не менее, круглые скобки обязательно требуется ставить.

Занятие 14. Практическая работа. Практическое задание.