Предмет: «Программирование в 1С»

Тема: «Конструкция языка 1С»

Теоретический материал

Язык 1С весьма прост в изучении по сравнению с любым другим языком программирования.

На территории России для многих большое значение имеет то, что можно писать код прямо на русском языке. «Можно» – значит не обязательно — можно писать и на английском, все операции и операторы имеют английский синоним.

Программирование – это написание программного кода на программном языке.

Программный код – это последовательность команд (слов), которые поддерживает этот язык программирования.

Команды делятся на:

* Переменные;
* Операторы;
* Функции и процедуры;
* Рассмотрим их в первом приближении.

**Переменные**

Есть цифра 12. Это значение. Сама по себе эта цифра нигде не хранится. Подули и ее уже нет. Она должна где-то храниться. Переменные – это место хранения значений. Переменная может называться любым словом.

Например:

Чтото = 12;

Чтото = Чтото + 10;

Сообщить(Чтото); //будет выведено «22»

Переменная, которая является частью объекта 1С (например поле справочника) и хранится в базе данных – называется реквизит.

В 1С переменные не типизированы, это значит, что одной и той же переменной сначала можно назначить значение одного типа (например цифру), а потом другого типа (например строку) и ошибки не будет.

Однако реквизиты объектов – типизированы. Но проверки на типы при этом не происходит. Мы можем попытаться назначить реквизиту значение не того типа. 1С попытается его преобразовать (например из цифры в строку), но если не получится, то значение реквизита останется пустым.

Длина строки в переменной – неограниченная. Длина строки реквизита – задается точно, но можно задать неограниченную (поставив длину строки 0).

По буквам к строке обращаться нельзя. Но можно искать или выделять части (Найти(), Лев(), Прав(), Сред()).

**Сложные переменные**

Если значение одно – это просто. А когда значений – много?

Массив

Это одна колонка Excel. В каждой ячейке – переменная. У нее одно название, так же как и у обычной переменной. А к ячейке можно обратиться по номеру.

Чтото = Новый Массив;

Чтото.Добавить(22); //в ячейке 1 значение 22

Чтото.Добавить(33); //в ячейке 2 значение 33

Сообщить(Чтото[0]); //выводим значение 1й ячейки

СписокЗначений

То же, что и массив, только к каждой ячейке можно подписать комментарий.

Чтото = Новый СписокЗначений;

Чтото.Добавить(22, «Это 22»); //в ячейке 1 значение 22

Чтото.Добавить(33, «Это 33»); //в ячейке 2 значение 33

Сообщить(Чтото.Получить(0).Значение); //выводим значение 1й ячейки

//а могли бы вывести и комментарий, тогда бы написали .Представление а не .Значение

ТаблицаЗначений

Это – уже полноценная страница Excel со строчками и колонками.

Как с ней работать – посмотрим в другом уроке.

ДеревоЗначений

Таблица представленная в виде дерева. Пример – окно конфигурации. Ветки, у каждой ветки может быть подчиненная ветка.

Обратите внимание – нумерация ячеек в 1С начинается не с 1, а с 0.

Строки программы

Строки заканчиваются на точку с запятой «;».

Комментарий начинается с «//» — то есть эта строчка или часть строчки не будет выполняться и будет пропущена.

Значение строк нужно указывать в кавычках «Значение». Если нужно указать в значении кавычку, то она удваивается – «Значение ««а»»!».

Значение строки может быть с включением переноса строки, тогда в начале следующей строки должен стоять «|», например:

«Это начало

| и продолжение значения»

Если нужно указать специальные символы в значении строки, для этого есть специальный объект: «Символы.», например:

«Это начало» + Символы.ПС + «и продолжение значения» //ПС – перенос строки

 Операторы

Это конструкции языка, которые заложены в нем самом и позволяют создавать простейшую последовательность выполнения программы.

Условие

Определенные строки кода будут выполнены если будет выполнено условие.

Чтото = 12;

Если Чтото < 100 или Чтото > 200 Тогда

 Сообщить(«Условие выполнено»);

ИначеЕсли Чтото > 200 Тогда

 Сообщить(«Условие не выполнено»);

Иначе

 Сообщить(«Что-то еще»);

КонецЕсли;

Цикл

Определенные строки кода будут выполнены указанное количество раз.

//считаем по-одному

Для Чтото = 1 по 20 Цикл

 Сообщить(Чтото);

КонецЦикла;

//считаем по-другому

Чтото = 1;

Пока Чтото < 20 Цикл Чтото = Чтото + 1; Сообщить(Чтото); КонецЦикла;

//если у нас список значений, то можно обойти каждое из его значений

Чтото = Новый СписокЗначений;

Для каждого ЗначениеСписка из Чтото Цикл

 Сообщить(ЗначениеСписка.Значение);

КонецЦикла;

//обратите внимание, что «ЗначениеСписка» - это переменная, она может называться как угодно

Для того, чтоб прервать цикл, можно внутри написать команду «Прервать», для того чтобы прервать только одну итерацию цикла – команду «Продолжить».

Работа с объектами

Все объекты, с которыми мы работаем в языке – является полноценными объектами, то есть могут иметь свои данные и свои методы.

Объекты языка создаются с помощью команды «Новый», например:

Чтото = Новый Массив();

Чтото = Новый СписокЗначений();

И т.п.

Объекты 1С нельзя создать – доступ к ним можно получить с помощью так называемых менеджеров, названных соответственно по веткам конфигурации: Справочники, Документы и т.п. Методы менеджера можно расширять (см. урок по модулям 1С, модуль менеджера объекта).

Например:

Справочники.Номенклатура.СоздатьЭлемент();

Документы.Накладная.НайтиПоНомеру(«…»);

Свои объекты программировать нельзя, как и наследовать или переопределять типовые.

Ошибки

Программист не всегда может догадаться, будет ли ошибка при выполнении определенной строчки кода.

Например если написать в коде внешней обработки:

Справочники.Номенклатура.НайтиПоКоду(«…») и открыть эту обработку в базе, где нет справочника с названием Номенклатура, то будет ошибка.

Чтобы этого не произошло, можно использовать следующую конструкцию:

Попытка

 Справочники.Номенклатура.НайтиПоКоду(«…»);

Исключение

 Чтото = ОписаниеОшибки();

 Сообщить(«Ой-ой-ой»);

 Сообщить(Чтото);

КонецПопытки;

Если в момент выполнения строчек программы между Попытка и Исключение возникнет ошибка, то выполнение будет прервано и перейдет на строчки между Исключение и КонецПопытки. Если не будет ошибки – эти строчки не будут выполнены.

Функция ОписаниеОшибки() работает только между Исключение и КонецПопытки и возвращает описание произошедшей ошибки.

 Функции и Процедуры

Функции и Процедуры – это способ взять несколько строчек кода и назвать их каким-то словом, как переменную. Когда мы потом напишем ее название в другом месте – будет вызвана эта функция (то есть выполнены эти строки кода).

Мы неоднократно обсуждали методы объектов и обработчики событий. Функции и процедуры и являются этими методами и обработчиками.

//Программируем

Процедура КакоетоДействие()

 Сообщить(«Чтото»);

КонецПроцедуры

//Вызываем

КакоетоДействие();

Функция отличается от процедуры тем, что она может вернуть значение, которое было вычислено в результате выполнения этих строк когда.

Например:

Функция Посчитать(Переменная1, Переменная2)

 Возврат Переменная1 + Переменная2;

КонецФункции

//Вызываем

Чтото = Посчитать(12, 20);

Сообщить(Чтото); //будет сообщение «32»

Домашнее задание

Составить конспект, фотографии отправить на почту.

Результат скинуть на почту с названием по образцу — Предмет\_Тема\_датаОтправки.

В теме письма указать ФИО преподавателя.

Дата исполнения до следующей пары.