

Информатика и информационно-коммуникационные технологии

Сафарьян Ольга
Александровна



Лекция 1. (ч 2)

ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКАХ ВЫСОКОГО УРОВНЯ.

1. Типы данных и операторы описания переменных
2. Основные операторы

Типы данных и операторы описания переменных

Описание (объявление) переменной заключается в указании её имени и типа.

Указание типа необходимо - для определения размера памяти, выделяемой при создании переменной и в дальнейшем используемой для хранения её значения.

Описание типа каждой переменной делает программу надёжнее, убыстряет её работу, т. к. транслятору не требуется тратить время на распознавание типа неописанной переменной при каждом обращении к ней.

Некоторые языки вообще не допускают использования неописанных переменных.

Типы данных и операторы описания переменных

В таблице 1 приводятся некоторые базовые типы переменных, имеющие сходные характеристики в разных языках и используемые в операторах описания переменных.

При этом для многих версий языка Бейсик допускается использовать неявное объявление путём дописывания к имени переменной суффикса – символа, указанного в таблице в скобках после ключевого слова, определяющего тип.

Некоторые базовые типы переменных

| Бейсик (СИМВОЛ) | Паскаль | Си++ * | Тип | Размер (байт) * | Диапазон значений * | |
|--------------------|---------|--------|----------------------------------|--------------------|---------------------|----------------|
| | | | | | МИН | МАКС |
| Integer (%) | Integer | short | Целое | 2 | -2^{15} | 32767 |
| Long (&) | LongInt | int | Длинное целое | 4 | -2^{31} | 2147483647 |
| Single (!) | Single | float | Вещественное обычн. точности | 4 | $\pm 1,4E-45$ | $\pm 3.4E+38$ |
| Double (#) | Double | double | Вещественное двойной точности | 8 | $\pm 4,9E-324$ | $\pm 1.7E+308$ |
| | Char | char | СИМВОЛ | 1 | 0 | 255 |
| String (\$) | String | | Строка (n симв.) | n | "" | 2 млрд. симв. |
| Boolean | Boolean | bool | Логический | 2 | false | true |

Типы данных и операторы описания переменных

В таблице 2 приводится синтаксис оператора описания переменных и пример его использования (объявление двух переменных целого типа) в разных языках.

Таблица 2

Описание переменных в разных языках

| Вид описания | Бейсик | Паскаль | Си++ |
|------------------------------|-----------------------------------|--------------------------------|------------------------|
| Синтаксис оператора | <code>Dim Имя [As Тип]</code> | <code>Var Имя: Тип;</code> | <code>Тип Имя;</code> |
| Пример – I и J – целого типа | <code>Dim I As Integer, J%</code> | <code>Var I, J: Integer</code> | <code>Int I, J;</code> |

Типы данных и операторы описания переменных

По *области действия* переменные различаются на локальные и глобальные. Чаще всего переменные являются локальными, их область действия распространяется на текущий участок программы, в котором они описываются.

Для повышения производительности и качества работы программистов необходимо описывать данные в виде, максимально приближенном к их реальным аналогам. Тип данных, позволяющий хранить вместе под одним именем несколько переменных, называется сложным, или структурированным. Каждый язык программирования имеет свои структурированные типы. Рассмотрим структуру, объединяющую элементы одного типа данных, — массив.

Типы данных и операторы описания переменных

Массивы — последовательности однотипных элементов, число которых фиксировано и которым присвоено одно имя.

Иногда массивы называют векторами. Другим сложным типом являются структуры (записи), состоящие из полей разного типа.

Синтаксис операторов описания массивов и структур представлен в таблице 3. Можно создавать массивы, состоящие из структур, и структуры, состоящие из массивов. В результате удаётся организовать структуры данных произвольной сложности: списки, деревья и др.

Таблица 3

Синтаксис операторов описания сложных типов переменных

| Тип | Бейсик | Паскаль | Си++ |
|-------------------------|---|---|---|
| Массив | <code>Dim Имя(размер) As Тип</code> | <code>Var Имя: <u>array</u>[от..до] of Тип;</code> | <code>Тип Имя[размер];</code> |
| Структура (запись) * | <code>Type ИмяСтруктуры ИмяПоля1 As Тип ... ИмяПоляN As Тип End Type</code> | <code>record ИмяПоля1: Тип; ... ИмяПоляN: <u>Тип</u>; end;</code> | <code>Struct ИмяСтруктуры { Тип ИмяПоля1; ... Тип ИмяПоляN; };</code> |

Основные операторы

В таблице 4 приводится синтаксис наиболее часто используемых операторов, имеющих сходное назначение для разных языков.

Так, комментарии вставляются в программу (только!) для удобства её последующего чтения, а на выполнение программы комментарии никакого влияния не оказывают.

Символ разделения операторов позволяет записать несколько операторов подряд в одну строку. Блок операторов (в Бейсике отсутствует) позволяет представить последовательность из нескольких следующих друг за другом операторов единым неделимым блоком.

Таблица 4

Синтаксис некоторых операторов

| Оператор | Бейсик | Паскаль | Си++ * |
|---|--|---|--|
| Комментарий | ' текст или Rem в начале строки | // или { несколько <u>строк</u> } | // или /* несколько строк */ |
| Разделение операторов | : | ; | ; |
| Блок операторов | | begin end; | { } |
| Присваивание | Variable=expression | Variable:=expression; | Variable=expression; |
| Условный оператор (Если-То-Иначе) | If условие Then операторTo Else операторИначе End If | If условие Then операторTo Else операторИначе; | If (условие) операторTo Else операторИначе; |
| Цикл (для счётчика i от нач до кон с шагом 1) | For i=нач To кон Тело_цикла Next | For i:=нач To кон do Тело_цикла; | For (i:=нач; i<=кон; ++i) Тело_цикла; |
| Функция | Function _ Имя(параметры) As _ Тип Тело_функции End Function | Function Имя(параметры): тип; begin Тело_функции End; | Тип Имя(параметры) { Тело_функции }; |
| Процедура | Sub Имя(параметры) Тело_процедуры End Sub | procedure Имя(параметры); begin Тело_процедуры End; | void Имя(параметры) { Тело_процедуры }; |

VBA соблюдает основной синтаксис и правила программирования языков-диалектов Бейсика (BASIC, Visual Basic), что делает его достаточно простым.

VBA является общей языковой платформой для приложений MS Office, что особенно удобно при изучении – среда VBA доступна, например, из редактора Word.

В то же время VBA обладает многими возможностями современных систем программирования: позволяет создавать объекты управления графического интерфейса пользователя, задавать и изменять свойства объектов, подключать к ним необходимый для конкретного случая программный код.