

Организация программ циклической структуры. Массивы (регулярные типы)

В Паскале существует три вида операторов цикла:

- 1) с параметром (**FOR**);
- 2) с предусловием (**WHILE**);
- 3) с постусловием (**REPEAT**).

★ **Цикл с параметром FOR** служит для организации цикла с известным числом повторений: **FOR I:= m1 TO m2 Do S;**
FOR I:= m1 DOWNT0 m2 Do S;

где **I** – параметр цикла (целого типа), **m1, m2** – начальное и конечное значение параметра соответственно, **S** – тело цикла, содержит один или несколько операторов **TO** – шаг изменения параметра цикла **+1**; **DOWNT0** – шаг изменения параметра цикла **-1**;

При выполнении оператора **FOR** выполняются следующие действия:

- вычисляется **<m1>** , которое присваивается параметру цикла;
- проверяется условие окончания цикла: **<l>** больше **<m2>** при использовании конструкции **TO** и **<l>** меньше **<m2>** при использовании конструкции **DOWNTO**;
- выполняется тело цикла, если параметр цикла не вышел за границу цикла;
- наращивается (**TO**) или уменьшается (**DOWNTO**) на единицу параметр цикла;
- все этапы, кроме первого, циклически повторяются.

При использовании оператора необходимо помнить:

- внутри цикла **FOR** нельзя изменять начальное, текущее или конечное значения параметра цикла;
- если в цикле с шагом **+1** начальное значение больше конечного, то цикл не выполнится ни разу. Аналогично для шага **-1**, если начальное значение меньше конечного;
- после завершения цикла значение параметра цикла считается неопределенным, за исключением тех случаев, когда выход из цикла осуществляется оператором **GOTO** или с помощью процедуры **BREAK**;
- телом цикла может быть другой оператор цикла.

Если телом цикла является другой цикл, то циклы называются **вложенными** или **сложными**.

Цикл, содержащий в себе другой цикл, называют **внешним**.

Цикл, содержащийся внутри другого цикла, называется **внутренним**.

Внутренний и **внешний** циклы могут быть любыми из трех видов: **FOR**, **WHILE** или **REPEAT**.

При построении вложенных циклов необходимо, чтобы все операторы внутреннего цикла полностью находились в теле внешнего цикла.

★ **Оператор цикла WHILE** позволяет организовать цикл с неизвестным числом повторений, так как он зависит от вычислений в операторе: **WHILE B DO S;** где **B** – логическое выражение, **S** – тело цикла (один или несколько операторов).

Если логическое выражение имеет значение **true**, то выполняются операторы, входящие в тело цикла. Как только логическое выражение принимает значение **false**, выполнение операторов цикла прекращается. Если логическое выражение сначала имеет значение **false**, то оператор не выполняется. Значение переменных, входящих в условие, должно изменяться в теле цикла, иначе оператор никогда не завершится

★ Оператор цикла **REPEAT** позволяет организовать цикл с неизвестным числом повторений:
REPEAT A1; A2; ...; AN
UNTIL B

Здесь **A1; A2; ...; AN** – операторы тела цикла; **B** – выражение булевского типа. Данная конструкция оператора цикла используется, если число повторений цикла заранее неизвестно, но известно условие выхода из цикла. Управляющее циклом логическое выражение является условием выхода из цикла. Если оно принимает значение **TRUE**, то выполнение цикла прекращается. При использовании оператора **REPEAT** цикл выполняется хотя бы один раз. В отличие от других операторов цикла оператор данного вида не требует операторных скобок **BEGIN – END**, так как их роль выполняют **REPEAT–UNTIL**.

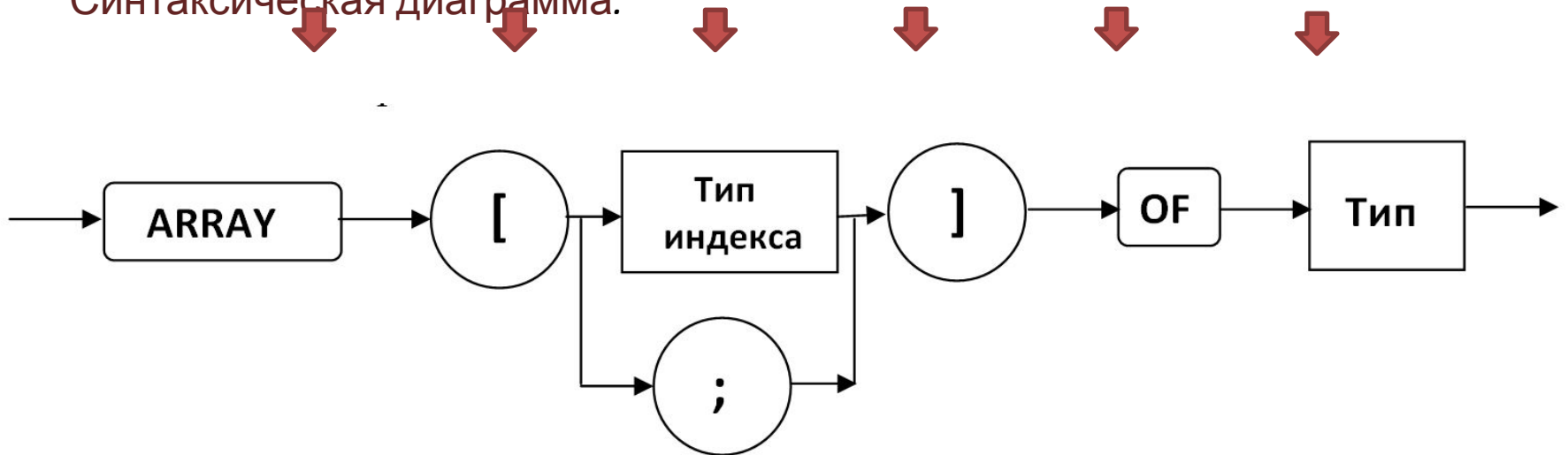
В языке Турбо Паскаль имеются процедуры **BREAK** и **CONTINUE**. Эти процедуры могут использоваться только внутри циклов **FOR**, **WHILE** или **REPEAT**. Процедура **BREAK** прерывает выполнение цикла и вызывает переход к оператору, следующему за циклом (может использоваться вместо оператора **GOTO**). Процедура **CONTINUE** осуществляет переход к следующему повторению цикла с пропуском последующих операторов тела цикла.

Массивы (регулярные типы)

Любой объект массивного типа (массив) состоит из фиксированного числа компонент (число определяется при введении этого массивного типа). Все компоненты относятся к одному типу, его называют **ТИПОМ КОМПОНЕНТ**.

Каждая компонента может быть явно обозначена с помощью имени переменной – массива, за которым в [] следует индекс. Индексы можно вычислять. Их тип называется **ТИПОМ ИНДЕКСА**.

Синтаксическая диаграмма.



Описание типа массива осуществляется следующим образом:

Type A=array [T1] of [T2];

где **A** – имя нового типа; **T1** – тип индекса, который должен быть дискретным (целый, символьный, логический); **T2** – любой тип.

Массивы позволяют группировать под одним именем несколько переменных, имеющих идентичные характеристики. Ко всей переменной – массиву применимы две операции: **присваивание** и **выборка компоненты**. Поскольку **T2** может быть любым типом, то компоненты массива могут быть и **составного типа**. В частности, если **T2** опять задаёт массив, то про исходный массив говорят, что он **многомерный**.

Если задано **n**-типов индексов, то массив называется **n-мерным**, а его компоненты обозначаются с помощью индексных выражений:

packed array [1 .. N] of char; .

В памяти ЭВМ элементы массива следуют друг за другом так, что при переходе от младших адресов к старшим наиболее быстро меняется самый правый индекс массива.