**Алгоритмы с циклической структурой на языке Паскаль**

Командой повторения или *циклом*называется такая форма организации действий, при которой одна и та же последовательность действий повторяется до тех пор, пока сохраняется значение некоторого логического выражения. При изменении значения логического выражения на противоположное повторения прекращаются(цикл завершается).

Различают циклы с известным числом повторений *(цикл с параметром)*и *итерационные*циклы (с пред- и постусловием).

В цикле с известным числом повторений параметр изменяется в заданном диапазоне. Если в цикле изменяется простая переменная, то она является параметром цикла; если в цикле изменяется переменная с индексом, то индекс этой переменной является параметром цикла.

Для организации цикла с известным числом повторений в Паскале используется оператор FOR.

**

Здесь **УП** — параметр, изменяющийся в цикле; **нз, кз**— выражения порядкового типа, обозначающие начальное и конечное значение параметра цикла. Шаг изменения номера параметра цикла равен **1***.*Параметр **УП**может быть как положительным, так и отрицательным. Соответственно, цикл может работать как по возрастанию, так и по убыванию значений своего параметра.

Структура цикла, организованного с помощью этого оператора, на языке Паскаль имеет вид:

**for i := a to b do begin <оператор> end;**

или

**for i := a downto b do begin <оператор> end;**

Шаг изменения номера параметра цикла равен **1**, если в заголовке цикла стоит **to**, и **-1** при **downto**. Возможность указания произвольного шага в языке Паскаль отсутствует: в таких случаях приходится использовать циклы с условиями.

Порядок выполнения цикла с шагом **1** следующий: вычисляются значения начального и конечного значений параметра цикла; параметр **УП(i)** принимает начальное значение (**нз**)**,** если **УП(i)**  меньше или равно конечному значению (**кз**), исполняется тело цикла; значение параметра цикла увеличивается, т. е. **i=i+1**; проверяется условие **УП(i)кз** (для отрицательного шага условие **УП(i)кз**)*,*и при его выполнении цикл повторяется. Выход из цикла осуществляется, если **УП(i)<кз**(**УП(i)>кз***для*шага -1), и выполняется оператор, следующий за оператором цикла.

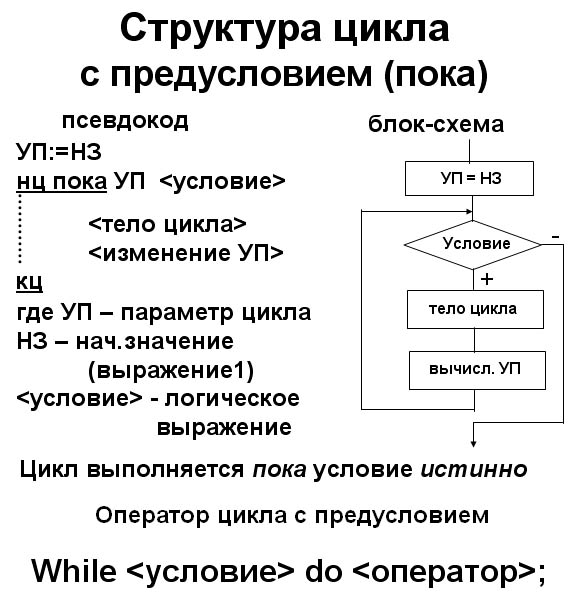
Если в операторе цикла с параметром начальное или конечное значение параметра заданы переменными или выражениями, то значения этих переменных должны быть определены в программе до оператора цикла. Не следует внутри цикла изменять параметр цикла, его начальное и конечное значения с помощью операторов присваивания или ввода.

Циклы могут быть вложенными, т. е. при выполнении одного цикла внутри него выполняется еще один или даже несколько циклов. Однако следует иметь в виду, что время выполнения вложенных циклов прямо пропорционально произведению конечных значений их параметров.

**Итерационные циклические алгоритмы**

По сравнению с циклом с параметром, рассмотренным нами выше, итерационные циклы являются универсальными. Для организации итерационных циклов используются операторы цикла с предусловием — цикл «ПОКА» и цикла с постусловием цикл «ДО».

Эти операторы не задают закон изменения параметра цикла, поэтому необходимо перед циклом задавать начальное значение параметра с помощью оператора присваивания, а внутри цикла изменять текущее значение этого параметра.

**

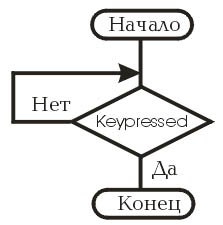
Для оператора цикла с предусловием проверяется значение логического выражения: если оно имеет значение true, то операторы, входящие в цикл, выполняются, в противном случае осуществляется выполнение оператора, следующего за циклом.

**

Цикл с постусловием выполняется хотя бы один раз. Затем проверяется значение логического выражения если оно false, то операторы, входящие в цикл, выполняются, в противном случае осуществляется выход из цикла.

Входить в цикл можно только через его начало, т. е. нельзя входить внутрь цикла с помощью управляющего оператора, так как в этом случае параметр цикла не определен.

Применение итерационного цикла можно найти при разработке интерфейса программ. Например, можно использовать функцию ожидания нажатой клавиши**keypressed**:

**

**Program Example\_ogidanie;**

**uses Crt;**

**repeat**

**until   keypressed;**

**end.**

Для использования функции **keypressed** необходимо подключить модуль Crt.