

Задание 8 – Анализ программы с циклами

Алгоритм бывает:

1. **Линейный** - образуется командами, выполняемыми однократно в той последовательности, в которой они записаны.
2. **Разветвляющийся** - алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий. Во многих случаях требуется, чтобы при одних условиях выполнялась одна последовательность действий, а при других – другая.
3. **Циклический** - описание действий, которые должны повторяться указанное число раз или пока не выполнено заданное условие. Перечень повторяющихся действий называют **телом цикла**.

Циклические алгоритмы бывают двух типов:

- **Циклы со счетчиком**, в которых какие-то действия выполняются определенное число раз;
- **Циклы с условием**, в которых тело цикла выполняется, в зависимости от какого-либо условия.

Циклы со счетчиком используют, когда заранее известно, какое число повторений тела цикла необходимо выполнить.

Циклы, в которых сначала проверяется условие, а затем, возможно, выполняется тело цикла называют *циклами с предусловием*. Если условие проверяется после первого выполнения тела цикла, то циклы называются *циклами с постусловием*.

Цикл **for** - цикл со счетчиком.

Цикл **for** существует в двух формах:

```
for счетчик:=значение to конечное_значение do  
    тело_цикла;  
for счетчик:=значение downto конечное_значение do  
    тело_цикла;
```

Счетчик – это переменная любого из перечисляемых типов (целого, булевого, символьного, диапазонного, перечисления). Начальные и конечные значения могут быть представлены не только значениями, но и выражениями, возвращающими совместимые с типом счетчика типы данных. Если между начальным и конечным выражением указано служебное слово **to**, то на каждом шаге цикла значение параметра будет увеличиваться на единицу. Если же указано **downto**, то значение параметра будет уменьшаться на единицу.

Количество итераций цикла определяется разностью между вторым и первым значением плюс единица. В Pascal тело цикла не должно содержать выражений, изменяющих счетчик.

Цикл **while** является циклом с предусловием. В заголовке цикла находится логическое выражение. Если оно возвращает **true**, то тело цикла выполняется, если **false** – то нет.

```
while (<условие>) do  
begin  
    <оператор_1>;  
    ...  
    <оператор_n>  
end;
```

Когда тело цикла было выполнено, то ход программы снова возвращается в заголовок цикла. Условие выполнения тела снова проверяется (находится значение логического выражения). Тело цикла выполнится столько раз, сколько раз логическое выражение вернет **true**. Поэтому очень важно в теле цикла предусмотреть изменение переменной, фигурирующей в заголовке цикла, таким образом, чтобы когда-нибудь обязательно наступала ситуация **false**. Иначе произойдет так называемое **зацикливание (бесконечный цикл)**, одна из самых неприятных ошибок в программировании.

Цикл **while** может не выполниться ни разу, если логическое выражение в заголовке сразу вернуло **false**. Однако такая ситуация не всегда может быть приемлемой. Бывает, что тело цикла должно выполниться хотя бы один раз, не зависимо оттого, что вернет логическое выражение. В таком случае используется цикл **repeat** – цикл с постусловием.

В цикле **repeat** логическое выражение стоит после тела цикла. Причем, в отличие от цикла **while**, здесь всё наоборот: в случае **true** происходит выход из цикла, в случае **false** – его повторение.

repeat
оператор 1;
оператор 2;
оператор 3;
.....
оператор N
until условие;

Формула для вычисления n -ого элемента арифметической прогрессии:

$$a_n = a_1 + d(n - 1)$$

Формула для вычисления суммы первых n членов арифметической прогрессии:

$$S_n = \sum_{i=1}^n a_i = a_1 + a_2 + \dots + a_n = \frac{a_1 + a_n}{2} \cdot n = \frac{2a_1 + d(n-1)}{2} \cdot n$$

где a_i – i -ый элемент последовательности, d – шаг (разность) последовательности