

## ПРАКТИЧЕСКАЯ РАБОТА №7

### «Программирование циклов операторами *do*, *do while*, *for*»

*Цель работы:* изучить операторы цикла и получить практические навыки их применения для организации циклических вычислений.

#### Задание

Изучить операторы цикла и особенности их использования.

Составить, отладить и выполнить программу по одному из вариантов, задаваемых преподавателем из таблицы 4.1. Распечатать текст программы.

Оформить отчет о лабораторной работе.

Указания по выполнению работы

1. В языке Паскаль имеется несколько вариантов оператора цикла:

а) Цикл с заданным числом повторений: *for (i=a;i<=b;i++) < тело цикла >;*

Здесь переменная цикла "i" (целого типа) увеличивается от начального значения "a" до конечного значения "b" с шагом 1 и для каждого значения "i" выполняются операторы тела цикла.

Имеется также вариант этого оператора цикла, в котором переменная цикла уменьшается от значения "b" до значения "a" с шагом 1: *for (i=a;i>b;i--) < тело цикла >;*

Если тело цикла содержит один оператор, то он записывается после ключевого слова "do", например, при выполнении цикла: *for (i=1;i<=5;i++) Console.WriteLine("{0}",i);* на экран будет выведено 12345

Если тело цикла содержит несколько операторов, то используется составной оператор. Например, требуется рассчитать и вывести на экран сумму первых двух, трех, четырех, пяти целых чисел от единицы до пяти. Тогда фрагмент соответствующей программы можно записать:

```
int i,s;  
s=1;  
for (i=2;i<=5;i++) {  
    s=s+i;  
    Console.WriteLine("Сумма целых чисел от 1 до {0} = {1}",i,s);  
}
```

В результате выполнения программы на экран будет выведено:

Сумма целых чисел от 1 до 2 = 3

Сумма целых чисел от 1 до 3 = 6

Сумма целых чисел от 1 до 4 = 10

Сумма целых чисел от 1 до 5 = 15

В этой программе реализован алгоритм расчета суммы.

Приведем пример, в котором реализуется алгоритм поиска наибольшего из 20 вводимых с клавиатуры чисел:

```
int i;  
double x=0,max=double.MinValue;  
for (i=1;i<=5;i++) {  
    Console.WriteLine("Введите значение x ");  
    x=Convert.ToDouble(Console.ReadLine());  
    if (x>max) max=x;  
}  
Console.WriteLine("Наибольшее число={0}",max);
```

б) Оператор цикла с неизвестным числом повторений и проверкой условия окончания цикла перед выполнением операторов тела цикла:

**while e < тело цикла >;**

Здесь "e" - выражение логического типа, операторы тела цикла выполняются, пока выражение "e" принимает значение "true".

Данный цикл не выполняется ни разу, если в самом начале выражение "e" принимает значение "false".

в) Оператор цикла с неизвестным числом повторений и проверкой условия окончания цикла после выполнения операторов тела цикла:

**do**  
**< тело цикла >**

**while d;**

Здесь "d" - выражение логического типа, операторы тела цикла повторяется до тех пор, пока выражение "d" имеет значение "false". Как только "d" принимает значение "true", выполнение цикла прекращается. Данный цикл обязательно выполнится, хотя бы один раз, т.к. проверка условия "d" осуществляется после выполнения операторов тела цикла.

Циклы "while" и "repeat" применяются, когда количество повторений заранее не определено, а выход из цикла происходит, например, по достижении заданной точности или заданного результата. Например, требуется определить количество первых целых слагаемых (начиная от единицы), сумма которых превышает 2000:

```
int k=1,s=1;
do {
  k=k+1;
  s=s+k;
} while (s<=2000);
Console.WriteLine("{0} - слагаемых",k);
```

Для обеспечения выхода из цикла (чтобы не произошло "зацикливание") необходимо, чтобы в теле цикла изменялась хотя бы одна из переменных, входящих в выражение "e" (см. п. б) или "d" (см. п. в). В приведенном примере в теле цикла изменяется переменная s, входящая в логическое выражение  $s \leq 2000$ .

2. Для одного из вариантов задания разработать и отладить программу, выбрав подходящий оператор цикла.

Таблица 7.1. Варианты заданий к лабораторной работе

Номер варианта	Задание
1	Рассчитать приближенное значение с точностью $\varepsilon$ $e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{3} + \dots + \frac{x^k}{k} + \dots$ (суммирование выполнять до тех пор, пока очередное слагаемое не станет меньше $\varepsilon$ )
2	Ввести N чисел и найти минимальное из них и его номер
3	Ввести N чисел и рассчитать сумму положительных из этих чисел
4	Рассчитать сумму членов последовательности: $S = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots - \frac{1}{10}$
5	Вводить произвольные числа до тех пор, пока сумма их модулей не станет больше заданного числа S. Напечатать количество введенных чисел.
6	Вводить и суммировать произвольные числа до тех пор, пока не будет введено число, целая часть которого кратна 10. Напечатать полученную сумму.

7	Вводить произвольные числа $x(i)$ и рассчитывать модуль разностей текущего и предыдущего чисел $\Delta =  x(i) - x(i-1) $ до тех пор, пока очередное значение $\Delta$ не станет меньше $\varepsilon = 0,01$ . Напечатать количество введенных чисел.
8	Рассчитывать и печатать значения функции $y = 4 - 2 \cdot \sin(x)$ для значений $x$ от 0 с шагом 0,5 до тех пор, пока очередное значение $y$ не станет больше предыдущего
9	Рассчитывать и печатать значения функции $y = 4 \cdot e^{-x}$ для значений $x$ от 0 с шагом 0,5 до тех пор, пока очередное значение $y$ не станет меньше единицы
10	Рассчитать приближенное значение: $e = 1 - \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{k!} + \dots$ , суммирование выполнять до тех пор, пока очередное слагаемое не станет меньше $\varepsilon$ )

### ВЫБОР ВАРИАНТА ЗАДАНИЯ

Вариант	Лабораторная работа 5
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	1
12	2
13	3
14	4
15	5
16	6
17	7
18	8
19	9
20	10
21	1
22	2
23	3
24	4
25	5
26	6
27	7
28	8
29	9
30	10

