

Лабораторная работа №08 по дисциплине "Программирование"

ТЕМА: Операторы (инструкции) цикла.

СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ПАПОК

1. В Вашей папке создайте папку с именем П_лр08, совпадающим с именем файла с этим заданием, и затем скопируйте файл с заданием в эту папку.
2. При выполнении задания создавайте в папке П_лр08 для каждой программы папку с именем, по смыслу соответствующим действиям, выполняемым составляемой программой, и сохраняйте все файлы, имеющие отношение к этой программе, в этой созданной папке (имена сохраняемых файлов должны отличаться от имен, присваиваемых по умолчанию, и должны по смыслу соответствовать действиям, выполняемым составляемой программой).
3. ПРИ ВЫПОЛНЕНИИ СЛЕДУЮЩИХ ЛАБОРАТОРНЫХ РАБОТ ДЕЙСТВУЙТЕ АНАЛОГИЧНО ПРЕДЫДУЩЕМУ.

ОПЕРАТОРЫ (ИНСТРУКЦИИ) ЦИКЛА

4. ОПЕРАТОРЫ ЦИКЛА служат для организации ЦИКЛОВ. ЦИКЛ представляет собой последовательность операторов, которая может выполняться более одного раза. Последовательность повторяемых операторов называется ТЕЛОМ ЦИКЛА. Для построения цикла в принципе можно использовать операторы ПЕРЕДАЧИ УПРАВЛЕНИЯ с передачей управления на начало ТЕЛА ЦИКЛА, но все-таки в большинстве случаев удобнее использовать ОПЕРАТОРЫ ЦИКЛА. Выполнение оператора цикла может быть прервано из ТЕЛА ЦИКЛА с помощью оператора перехода **goto** или специального оператора без параметров **break**, который передает управление оператору, следующему за ОПЕРАТОРОМ ЦИКЛА. С помощью специального оператора без параметров **continue** можно досрочно передать управление в конец ТЕЛА ЦИКЛА, что равносильно досрочному завершению очередного повторения ТЕЛА ЦИКЛА. Операторы цикла могут быть вложены друг в друга.
5. ОПЕРАТОР ЦИКЛА С ПРЕДУСЛОВИЕМ (с предварительным условием)
while <Условие> **do** <Оператор>;
целесообразно использовать, когда число повторений тела цикла заранее неизвестно и тело цикла может даже ни разу не выполниться.
<Оператор> тела цикла выполняется, пока значение логического выражения <Условие> равно **true** до тех пор, пока это значение не примет значение **false**, после чего происходит передача управления оператору, следующему за этим оператором цикла.
Если перед первым выполнением цикла значение <Условия> равно **false**, то тело цикла не выполняется ни разу и происходит передача управления оператору, следующему за этим оператором цикла.
6. ОПЕРАТОР ЦИКЛА С ПОСТУСЛОВИЕМ (с последующим условием)
repeat
<Оператор1>;
...
<ОператорN>;
until <Условие>;
целесообразно использовать, когда тело цикла необходимо выполнить не менее одного раза и заранее неизвестно число повторений цикла.
<Операторы> между **repeat** и **until** составляют тело цикла и выполняются, пока

значение логического выражения <Условие> не примет значение **true**, после чего происходит передача управления оператору, следующему за этим оператором цикла, (то есть тело цикла повторяется при значении <Условия>, равном **false**). Так как условие проверяется в конце цикла, <Операторы> выполняются хотя бы один раз.

<Операторы> тела цикла должны влиять на значение <Условия>, иначе произойдет заикливание (повторение тела цикла без возможности выхода из цикла).

7. ОПЕРАТОР ЦИКЛА С ПАРАМЕТРОМ имеет две разновидности:

for <Параметр>:=<Выражение1> **to** <Выражение2> **do** <Оператор>;

и

for <Параметр>:=<Выражение1> **downto** <Выражение2> **do** <Оператор>;

и применяется, если количество повторений цикла известно заранее.

<Оператор> тела цикла выполняется до полного перебора всех значений <Параметра> (параметра цикла) с шагом 1 для оператора с **to** и с шагом -1 для оператора с **downto** от значения <Выражения1> (начального значения) до значения <Выражения2> (конечного значения).

Цикл с **to** может не выполниться ни разу, если начальное значение больше конечного.

Цикл с **downto** может не выполниться ни разу, если начальное значение меньше конечного.

8. Примеры использования различных ОПЕРАТОРОВ ЦИКЛА для суммирования элементов числового ряда

$$S = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \dots + \frac{1}{n^2} \text{ с заданной точностью } e \text{ или заданным числом } n:$$

```
e:=0.001;
```

```
s:=0;
```

```
n:=1;
```

```
a:=e+1;
```

```
while a>e do begin a:=1/(n*n); s:=s+a; n:=n+1 end;
```

```
e:=0.001;
```

```
s:=0;
```

```
n:=1;
```

```
repeat
```

```
  a:=1/(n*n);
```

```
  s:=s+a;
```

```
  n:=n+1;
```

```
until a<=e;
```

```
n:=1000;
```

```
s:=0;
```

```
for i:=1 to n do begin a:=1/(i*i); s:=s+a end;
```

СОЗДАНИЕ ПРОГРАММ

ПРОГРАММА 8-1, ПРОГРАММА 8-2 и ПРОГРАММА 8-3

9. Создайте три консольных приложения (для циклов с **while**, **repeat** и **for**), каждое из которых, по аналогии с соответствующим примером из п.8, должно суммировать элементы числового ряда

$$S = \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}, \text{ где } 1!=1, \quad 2!=1 \cdot 2=1! \cdot 2, \quad 3!=1 \cdot 2 \cdot 3=2! \cdot 3, \quad \dots, \\ n!=1 \cdot 2 \cdot 3 \cdot \dots \cdot (n-1) \cdot n=(n-1)! \cdot n;$$

с введенным значением точности или,
для цикла **for**, с введенным значением числа **n**,
и выводить результат вычислений.

ОТЧЕТ О ВЫПОЛНЕНИИ РАБОТЫ

10. Продемонстрируйте тексты созданных программ, а также их выполнение в среде программирования и независимо от нее.
-