

ОСНОВНЫЕ КОМАНДЫ

Циклические операторы

Цикл «Пока»

Выполняет команды тела цикла, пока выполняется условие, заданное в заголовке

Синтаксис

While *Условие* **do**

begin

Список команд (тело цикла)

end;

Если в теле цикла всего одна команда, то **begin** и **end** не обязательны.

While условие **do** оператор;

ОСНОВНЫЕ КОМАНДЫ

Циклические операторы

Цикл «Пока»

Примеры

```
x:=0;  
while x<=100 do  
  begin  
    x:=x+5;  
    writeln (x);  
  end;
```

*Программа будет увеличивать **x** на 5 и выводить результат на экран, пока **x** не станет равной или больше 100*

```
x:=100;  
while x>=10 do x:=x/2;
```

*Программа будет делить **x** пополам, пока результат не будет меньше 10.*

```
s:='yes';  
while s='yes' do  
  begin  
    write ('Введите число');  
    readln (x);  
    write ('Модуль числа равен ', abs(x));  
    s:='no';  
    write ('Введите yes для продолжения');  
    readln (s);  
  end;
```

*Программа будет запрашивать переменную **x** и выводить её модуль до тех пор, пока на запрос строковой переменной **s** будет вводится ответ «yes».*

ОСНОВНЫЕ КОМАНДЫ

Циклические операторы

Цикл «Повторяй»

Выполняет команды тела цикла, пока выполняется условие, заданное в заголовке

Синтаксис

Repeat

Список команд (тело цикла)

Until *Условие;*

Цикл Repeat будет выполнен минимум один раз, даже если условие выхода выполняется изначально.

Условие цикла Repeat – условие выхода из цикла, тогда как условие цикла While – условие выполнения цикла.

ОСНОВНЫЕ КОМАНДЫ

Циклические операторы

Цикл «Повторяй»

Примеры

```
x:=50;  
repeat  
    writeln (x);  
until x>=50;
```

*Программа выполнит тело цикла один раз, несмотря на то, что условие выхода из цикла выполнено и выдаст на экран значение переменной **x**.*

```
a:=10; s:=0;  
repeat  
    a:=a/2;  
    s:=s+a;  
    writeln(a:7:3);  
until a<0.01;
```

Программа вычисляет сумму ряда

$$s = \frac{10}{2} + \frac{5}{2} + \frac{2.5}{2} + \frac{1.25}{2} + \dots,$$

*до тех пор, пока точность вычислений не станет меньше 0.01 и выводит каждый член ряда **a** на экран.*

ОСНОВНЫЕ КОМАНДЫ

Циклические операторы

Цикл со счетчиком

Выполняет команды тела цикла последовательно увеличивая или уменьшая переменную счетчика, до тех пор, пока переменная счетчика не достигнет последнего значения.

Синтаксис

Цикл с увеличением счетчика

```
For  $i:=a$  to  $b$  do
```

```
begin
```

```
Список команд  
(тело цикла)
```

```
end;
```

Цикл с уменьшением счетчика

```
For  $i:=a$  downto  $b$  do
```

```
begin
```

```
Список команд  
(тело цикла)
```

```
end;
```

Где i – переменная счетчика цикла, a – начальное значение счетчика, b – конечное значение счетчика.

Переменная счетчика – целочисленная (integer) или символьная (char), изменение переменной счетчика происходит на единицу для integer и один символ для char.

Изменять переменную счетчика внутри цикла нельзя.

ОСНОВНЫЕ КОМАНДЫ

Циклические операторы

Цикл «Повторяй»

Примеры

```
For i:=10 downto 1 do  
  begin  
    write(i);  
    writeln(i*i:4);  
  end;
```

Программа выведет числа от 10 до 1 и их квадраты.

```
For i:=1 to 10 do read(x);
```

Программа запросит у пользователя 10 чисел.

```
i:=0;  
For s:='a' to 'z' do  
  begin  
    i:=i+1;  
    write(i,' буква алфавита – ');  
    writeln(s);  
  end;
```

Программа в счётчике цикла **s** (char) переберёт все буквы латинского алфавита (от **a** до **z**)

И выдаст на экран 26 сообщений:

«1 буква алфавита – a»

...

«4 буква алфавита – d»

...

«26 буква алфавита – z»

ОСНОВНЫЕ КОМАНДЫ

Условный оператор

Пример программы с вложенными циклами

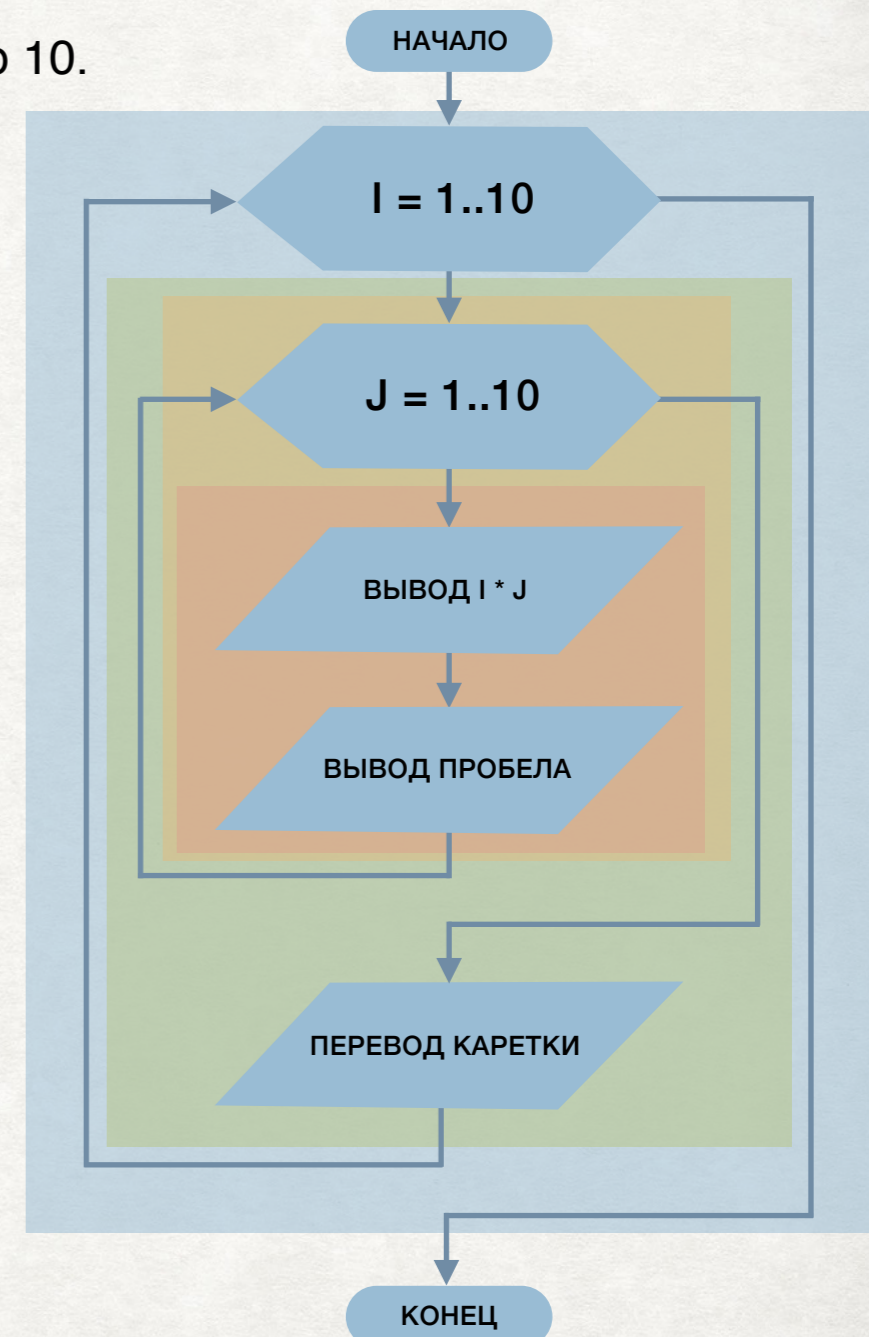
Задача

Вывести на экран таблицу умножения от 1 до 10.

```
program tabl; // Заголовок программы
// Определение используемых переменных
var
  i, j: integer;
begin
  // Цикл задающий номер строки (внешний цикл)
  for i:=1 to 10 do
    begin
      // Цикл задающий номер столбца (внутренний цикл)
      for j:=1 to 10 do
        begin
          write (i*j:4); // Вывод произведения
          write (' '); // Вывод пробела
        end;
        writeln(); // Перевод каретки на следующую строку.
      end;
    end;
  end.
```

Тело
внутреннего
цикла

Тело
внешнего
цикла



Цикл по i при первом проходе присвоит i единицу, после чего программа перейдет во внутренний цикл по j . Цикл по j выведет 10 раз произведение i на j и пробел после него, после чего вернется в цикл по i .