

Цикл For

Цикл For

For <параметр цикла> := <начальное значение> **to** <конечное значение> **do**
begin
 <инструкции, называемые телом цикла,>
end;

For<параметр цикла>:= начальное значение>**downto**<конечное значение> **do**
begin
 <инструкции, называемые телом цикла,>
end;

Порядок выполнения инструкции **for...to..do**

1. Вычисляются <начальное значение> и <конечное значение >. **Один лишь раз!**
2. Переменной < параметр цикла > присваивается <начальное значение>.
3. Если значение <параметра цикла> превышает <конечное значение>, то цикл заканчивается.
4. Выполняется тело цикла.
5. Переменной < параметр цикла > присваивается следующее значение.
6. Выполнение продолжается с пункта 3.



Пример использования цикла for

```
function factorial(n:integer):int64;  
    var i:integer;  
begin  
    result:=1;  
    for i:=1 to n do  
        result:=result*i  
    end;
```

Пример использования цикла for

```
function power(x:real; n:integer):real;  
    var i:integer;  
begin  
    result:=1;  
    for i:=1 to n do  
        result:=result*x;  
end;
```

Среднее арифметическое (Цикл с for)

```
function average( n : integer ) : real;  
  var sum:real; count: integer;  
begin  
  sum := 0;  
  for count :=n downTo 1 do  
  begin  
    sum := sum + random();  
  end;  
  result := sum / n;  
end;
```

Какой будет result?

```
...  
b:=10; result :=0;  
for i:=result to 2*b do  
begin  
    result := result +1;  
    b:=b-1;  
end;  
...
```


Табулирование функции

```
Procedure tab(x0, xMax, step : real;)
  // Начальное (x0), текущее (x), конечное (xMax), шаг (step)
var x,y : real; k, i : integer; s : string;
begin
  k:= round((xMax-x0)/step); // k- номер последней строки
  for i:=0 to k do // i – номер текущей строки begin
    x := x0 + step * i; // x - очередное значение x
    y := sin(x); // y - очередное значение y s :=
format('x=%5.3 y=%5.3',[x,y] );
memo1.Lines.Append(s);
  end;
end;
```

Вложенные циклы

Построение циклов

Синус | Корень | For | Табл

Максимальное число | 50

Найти подходящие квадраты чисел до

5^2	=	3^2	+	4^2
10^2	=	6^2	+	8^2
13^2	=	5^2	+	12^2
15^2	=	9^2	+	12^2
17^2	=	8^2	+	15^2
20^2	=	12^2	+	16^2
25^2	=	7^2	+	24^2
25^2	=	15^2	+	20^2
26^2	=	10^2	+	24^2

Вложенные циклы

```
procedure TfrmCharCod.Button2Click(Sender: TObject);
  var n, a1, a2, max: integer; s: string;
begin
  Mem1.Clear;
  max:=strToInt(Edit1.text);
  for n:=1 to max do
    for a1:=1 to max-1 do
      for a2:=a1 to max-1 do
        if sqr(a1)+sqr(a2)=sqr(n) then
          begin
            s:= intToStr(n)+'^2=' +
              intToStr(a1)+'^2+'+intToStr(a2)+'^2';
            Mem1.Lines.append(s);
          end;
        end;
      end;
    end;
  end;
```

Тип String (ShortString)

Строка – это тип данных, которому соответствует последовательность символов. При работе со строками оперируют понятиями:

- имя строки,
- размер строки,
- СИМВОЛ,
- номер символа (индекс).

Тип String (ShortString)

Данные типа ShortString представляют собой последовательности 8-битовых ANSI символов, количество которых может быть от 0 до 255. Длина строки в процессе работы программы может изменяться, но размер выделяемой памяти при этом не меняется и равен 256 байтов.

Начальный (нулевой байт) используется для хранения длины строки, в остальных хранятся символы строки.

Поэтому символы нумеруются с 1 и длина строки не может быть больше 255.

Типы пользователя для ShortString

type

```
Tstring10 = String[10];
```

```
var myString: Tstring10;
```

Эта запись равносильна такой:

```
var MyString: string[10];
```

Но так лучше не писать! Так как

```
procedure Check(S: string[10] ); но
```

```
procedure Check(S: Tstring10 );
```

Операции над строками

>, >=, <, <=, =, <>, +, []

'papa' < 'mama' (false)

'Papa' < 'mama' (true)

'papa' > 'pap' (true)

Var s:string[10]

s:='papa'+ 'mama' ; ('papamama')

s[1] ('p')

Length(s) равно 8

Формирование строк

```
procedure TForm1.Button1Click(Sender:  
    TObject);  
var s: String[20]; i: integer;  
begin  
    s:="";  
    for i:=1 to 20 do  
        s := s+intToStr(random(2))[1];  
    edit1.Text:=s;  
end;
```


Формирование строк

```
procedure TForm1.Button1Click(Sender:  
    TObject);  
var s: String[20]; i: integer;  
begin  
    for i:=1 to 20 do  
        s[i] := intToStr(random(2))[1];  
    edit1.Text:=s;  
end;
```

Пример обработки строки

```
procedure Cod_Decod(var str:string;  
    mask:char);  
var m, i:integer;  
Begin  
    m:=ord(mask);  
    for i:=1 to length(str) do  
        if (str[ i ]<>mask) and (str[ i ]<>chr(0))  
            then str[ i ]:=chr( ord( str[ i ] ) xor m);  
    end;
```

Стандартные процедуры и функции для строк

```
function Pos (sub: string; S: string): Integer;  
function AnsiUpperCase (const S:string): string  
function AnsiLowerCase (const S:string): string  
function Copy (S; Index, Count: Integer): string;  
procedure Delete (var s: string; ind, cnt:integer);  
procedure Insert (ins: string; var S: string; ind: Integer);  
function Trim (const S: string): string;  
function TrimLeft (const S: string): string;  
function TrimRight (const S: string): string
```

```
procedure TForm1.Button1Click(Sender: TObject);  
    var s : String;    w : String[20];    i, wordEndPos : integer;  
begin  
    s:=Edit1.Text; //прочитали строку из нескольких слов  
    while length(Trim(s))>0 do  
    begin  
        s:=Trim(s); // Удаляем пробелы  
        // Находим позицию конца очередного слова  
        wordEndPos:=Pos( ' ',s); // Ищем пробел  
        if wordEndPos = 0 // Пробела не нашли,  
        then wordEndPos := length(s) // значит конец слова это конец строки  
        else wordEndPos := wordEndPos -1; // Конец слова левее пробела  
        // Копируем слово в переменную w  
        w:= Copy(s,1,wordEndPos);  
        // Выводим слово в MEMO  
        Memo1.Lines.Append(w);  
        // Удаляем это слово из исходной строки  
        Delete(s,1,wordEndPos);  
    end; // Конец цикла поиска слов  
end;
```