

Гомельский государственный университет  
имени Франциска Скорины

## **Циклы**

Составил:

Ассистент кафедры общей физики  
Соколов С.И.

Гомель, 2015

## Инструкция for

Рассмотрим следующую задачу. Пусть нужно написать программу, которая вычисляет значение функции  $y = 5x^2 - 7$  в точках  $-1$ ,  $-0.5$ ,  $0$ ,  $0.5$  и  $1$

(таблица должна быть выведена в поле метки формы приложения). Процедура, обеспечивающая решение поставленной задачи, может выглядеть так:

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
y: real; // значение функции  
x: real; // аргумент функции  
dx: real; // приращение аргумента  
st: string; // изображение таблицы  
begin  
st:="";  
x := -1; dx := 0.5;  
y := 5*x*x -7;  
st := st+ FloatToStr(x)+' '+ FloatToStr(y)+chr(13);  
x :=x + dx;  
y := 5*x*x -7;
```

```
st := st+ FloatToStr(x)+' '+ FloatToStr(y)+chr(13);  
x :=x + dx;  
y := 5*x*x -7;  
st := st+ FloatToStr(x)+' '+ FloatToStr(y)+chr(13);  
x :=x + dx;  
y := 5*x*x -7;  
st := st+ FloatToStr(x)+' '+ FloatToStr(y)+chr(13);  
x :=x + dx;  
y := 5*x*x -7;  
st := st+ FloatToStr(x)+' '+ FloatToStr(y)+chr(13);  
x :=x + dx;  
Label1.Caption := st;  
end;
```

Из текста процедуры видно, что группа инструкций

```
y := 5*x*x - 7;
```

```
st := st + FloatToStr(x) + ' ' + FloatToStr(y) + chr(13);
```

```
x := x + dx;
```

обеспечивающая вычисление значения функции, формирование строки таблицы и увеличение аргумента, выполняется 5 раз.

```
procedure TForm1.Button1Click(Sender:  
TObject);
```

```
var
```

```
y: real; // значение функции
```

```
x: real; // аргумент функции
```

```
dx: real; // приращение аргумента
```

```
st: string; // изображение таблицы
```

```
i : integer; // счетчик циклов
```

```
begin
```

```
st:=""; x := -1; dx := 0.5;
```

```
for i:=1 to 5 do begin
```

```
y := 5*x*x -7;
```

```
st := st+ FloatToStr(x)+' '+ FloatToStr(y)+chr(13);
```

```
x :=x + dx;
```

```
end;
```

```
Label1.Caption := st;
```

```
end;
```

В общем виде инструкция for записывается следующим образом:

```
for счетчик := нач_знач to кон_знач do begin
```

```
// здесь инструкции, которые надо выполнить несколько раз
```

```
end;
```

где:

- счетчик — переменная-счетчик числа повторений инструкций цикла;
- нач\_знач-- выражение, определяющее начальное значение счетчика циклов;
- кон\_знач — выражение, определяющее конечное значение счетчика циклов.

Количество повторений инструкций цикла можно вычислить по формуле  
(кон\_знач — нач\_знач + 1).

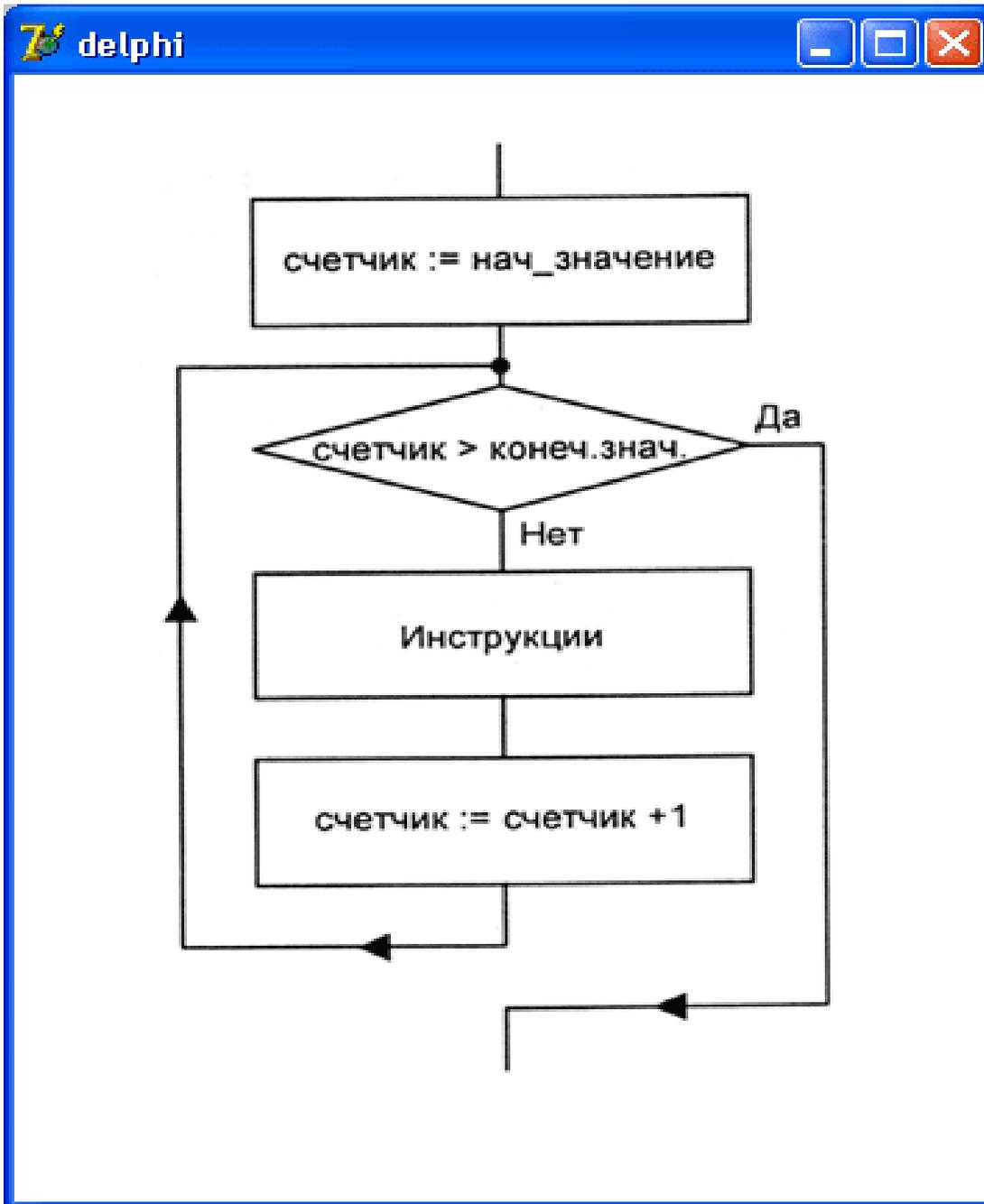
Примеры:

```
for i:=1 to 10 do begin
```

```
label1.caption:=label1.caption + '*';
```

```
end;
```

```
for i: =1 to n do s := s+i;
```



Переменную-счетчик можно использовать внутри цикла (но ни в коем случае не изменять).  
Например, в результате выполнения следующих инструкций:

```
tab1 := '' ;
```

```
for i:=1 to 5 do
```

```
begin
```

```
tab1:=tab1+IntToStr(i)+' '+IntToStr(i*i)+chr(13);
```

```
end;
```

переменная `tab1` будет содержать изображения таблицы квадратов чисел.

## Инструкция `while`

Инструкция (цикл) `while` используется в том случае, если некоторую последовательность действий (инструкций программы) надо выполнить несколько раз, причем необходимое число повторений во время разработки программы неизвестно и может быть определено только во время работы программы.

Типичными примерами использования цикла `while` являются вычисления с заданной точностью, поиск в массиве или в файле.

В общем виде инструкция `while` записывается следующим образом:

**`while`** условие **`do begin`**

`//` здесь инструкции, которые надо выполнить  
несколько раз

**`end;`**



## Инструкция repeat

Инструкция repeat, как и инструкция while, используется в программе в том случае, если необходимо выполнить повторные вычисления (организовать цикл), но число повторений во время разработки программы неизвестно и может быть определено только во время работы программы, т. е. определяется ходом вычислений.

В общем виде инструкция repeat записывается следующим образом:

**repeat**

// инструкции

**until** условие

