**Короткі теоретичні відомості**

Для програмування циклів з відомим числом повторень слід користуватися **оператором for***.* Цей оператор передбачає повторне виконання деякого іншого оператора з одночасною зміною за правилом арифметичної прогресії значення, що привласнюється змінній цього циклу, що управляє.

Оператор циклу з параметром має наступний вигляд:

**FOR** *параметр циклам := поч. знач.* **ТО** *кін. знач.* **DO** *оператор*;

або

**FOR** *параметр циклу := поч. знач.* ***DOWNTO*** *кін. знач.* **DO** *оператор*;

Тут **FOR** (для), **ТО** (до), **DO** (виконати) **DOWNTO** (вниз до), - службові слова;

***параметр циклу* -** змінна будь-якого скалярного типа, окрім речовинного;

***поч. знач., кін. знач*. -** скалярні вирази того ж типа, що і параметр циклу;

***оператор* -** будь-який оператор мови.

**Крапку з комою після do не ставлять.**

При використанні в циклі службового слова ТЕ значення параметра циклу збільшується, при DOWNTO - зменшується. Крок зміни параметра циклу залежить від його типа. Найчастіше використовується змінна цілого типа, що визначає крок, рівний 1 при ТЕ, і -1 - при DOWNTO.

**Приклад:**

for i:=l to 10 do begin

label1.caption:=label1.caption + '\*'; end;

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

Якщо між begin і end знаходиться тільки одна інструкція, то слова begin і end можна не писати.

*Розглянемо наступне завдання.* Хай потрібно написати програму, яка обчислює значення функції у = 5х2 - 7 в крапках —1, -0.5, 0, 0.5 і 1 (таблиця повинна бути виведена в полі мітки форми застосування). Процедура, що забезпечує рішення поставленої задачі, може виглядати так:

procedure TForm1.ButtonlClick(Sender: TObject);

var

у: real; // значення функції

x: real; // аргумент функції

dx: real; // приріст аргументу

st: string; // зображення таблиці

i : integer; // лічильник циклів

begin

st:=''; x := -1; dx := 0.5;

for i:=l to 5 do begin

у := 5\*x\*x -7;

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

Label1.Caption := st;

end;



Рисунок 1 Алгоритм оператора for

Алгоритм, відповідний оператора for, представлений на мал. 1 Звернете увагу, що якщо початкове значення лічильника більше кінцевого значення, то послідовність операторів між begin і end не буде виконана жодного разу.

Крім того, після кожного виконання інструкцій тіла циклу лічильник циклів збільшується автоматично.

Змінну-лічильник можна використовувати усередині циклу (але у жодному випадку не змінювати). Наприклад, в результаті виконання наступних інструкцій:

tab1: = '' ;

for i:=l to 5 do

begin

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

end;

змінна tabl містить зображення таблиці квадратів чисел.

*Розглянемо програму, яка обчислює суму перших 10 елементів ряду:*

1 + + 1/3 + ... (значення i-го елементу ряду пов'язане з його номером формулою 1//). Діалогове вікно програми повинне містити, принаймні, два компоненти: поле мітки (Label1) і командну кнопку (Buttonl).

Обчислення суми ряду і виведення результату виконує процедура обробки події Onclick, текст якої приведений нижче. Після обчислення чергового елементу ряду процедура виводить в полі Labell номер елементу і його значення в поле мітки форми, призначене для виведення результату.

procedure TForm1.ButtonlClick(Sender: TObject);

var

i:integer; { номер елементу ряду }

elem:real; { значення елементу ряду }

summ:real; { сума елементів ряду )

begin

summ:=0;

label Leapt ion: = ' ' ;

for i:=l to 10 do begin

elem:=l/i;

label1.caption:=labell.caption+

IntToStr(i)+' '+FloatToStr(elem)+#13; sunrn: =summ+elem;

end;

label1.caption:=label1.caption+'Сумма ряду:'+FloatToStr(summ);

end;

end.