

Використання циклів

Мета роботи: навчитися використовувати цикли в ході розроблення додатків.

Завдання. Ввести кількість членів ряду й розрахувати значення π .

Теоретичні відомості

На початку XVIII ст. англійський математик Брук Тейлор запропонував низку формул для розкладання функцій у степеневі ряди. Використання цього відкриття відіграло важливу роль у реалізації на обчислювальних машинах різних функцій. Замість зберігання величезних таблиць значень функцій для різних аргументів у бібліотеці мови програмування може бути записаний лише алгоритм визначення результуючого значення шляхом обчислення нескладної формули — суми обмеженого числа елементів ряду (числової послідовності).

Найвідоміша з цих формул, яка дістала назву «ряд Тейлора», призначена для обчислення функції $\arctg(x)$:

$$\arctg(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \dots$$

Якщо у формулу підставити $x = 1$, то $\arctg(1) = \frac{\pi}{4}$

Тоді формула набуває вигляду $\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$ або

$$\pi = 4 \cdot \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots \right).$$

Багато поколінь програмістів користувалися цим рядом для обчислення значення числа π із заданою точністю.

Хід роботи

1. Запустимо середовище Delphi та створимо новий додаток. На форму (рис. 1) помістимо три відеокомпоненти: кнопку, Edit та Label. В Edit вводиться кількість членів ряду, залучена до підсумовування. Натисканням кнопки в Label виводиться обчислене значення π .

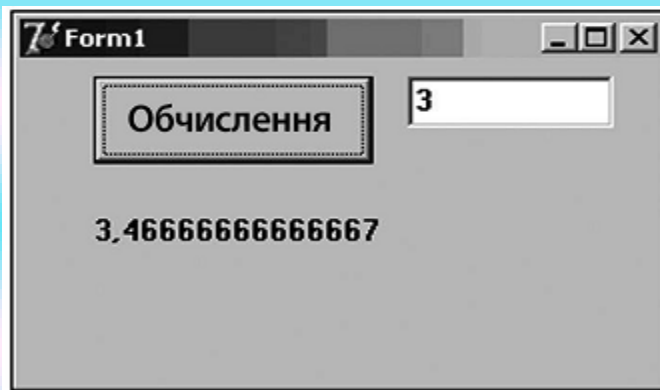


Рис. 1. Вигляд форми для обчислення числа π за допомогою ряду Тейлора

2. Уся програма для розрахунку π міститься в методі (процедурі), що викликається натисканням однієї кнопки на формі, і має такий вигляд:

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

```
var
```

```
    i,k,zn:integer;
```

```
    s:real;
```

```
begin
```

```
    k:=strtoint(Edit1.Text);
```

```
    s:=0; {сума членів ряду}
```

```
    zn:=1; {знак чергового доданка, чергується +1, -1, +1,...}
```

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

```
        begin
```

```
            s:=s + zn*(1/(2*i-1));
```

```
            zn:=zn*(-1);
```

end;

labell.caption:=floattostr(4*s);

end;

3. Наведемо деякі пояснення до даної програми. Зверніть увагу на типи даних: кількість членів ряду, параметр циклу й знак доданка — цілочислові змінні; сума ряду, що накопичує знакозмінні значення простих дробів,— дійсна величина. Знак наступного доданка визначається окремою змінною, хоча можна було б запропонувати і який-небудь арифметичний вираз, наприклад: $2 * (i \bmod 2) - 1$.

Цей вираз дає значення «-1» для парних i та «+1» — для непарних. Оскільки в циклі після оператора FOR виконуються відразу два оператори, в оформленні циклу застосовано операторні дужки `begin ... end`, що слугують для об'єднання кількох операторів (у цьому випадку їх два) в один складений оператор.

4. Якщо запустити програму і спробувати вводити різні числа, то одержимо, що ряд з одного елемента дає значення k , яке дорівнює 4, із двох елементів — 2,666..., із трьох — 3,4666... і т. д. Зі збільшенням числа доданків точність зростатиме.

5. Збережемо проект на диску.

6. Створимо новий додаток, потрібний для розв'язування наступної задачі.

Ще одним популярним алгоритмічним застосуванням циклів є перевірка на подільність: досліджуване ціле число послідовно ділять на ряд натуральних чисел (зазвичай від 2 до $n-1$, хоч у дійсності слід перевіряти на подільність від 2 до $\sqrt{n}$), після чого й перевіряють остачу від ділення — якщо ніде не вийде значення 0, то число просте. Розглянемо кілька прикладів.

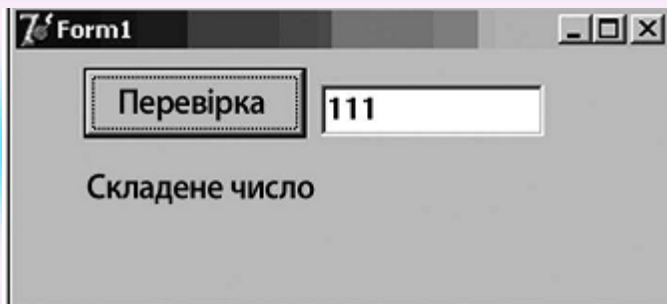


Рис. 2. Форма програми для визначення простих чисел

Завдання 1. Ввести ціле число й вивести в Label рядок «Просте число» або «Складене число» після перевірки його подільності на натуральний ряд чисел від 2 до $n-1$.

Розв'язання. Спроекуємо форму, як показано на рис. 2.

Текст програми, набраний у методі `OnClick` — єдиної кнопки на формі,— має такий вигляд:

```
procedure TForm1.Button1Click(Sender: TObject);  
  
var  
    i,к:integer;  
  
begin  
    к:=strtoint(Edit1.text);  
    Label1.caption:='Просте число';  
    for i:=2 to к - 1 do  
        if к mod i = 0 then  
            Label1.caption:='Складене число';  
  
end;
```

Зауважимо, що зазвичай цю задачу розв'язують за допомогою додаткової логічної змінної (прапорця), яка міняє своє вихідне значення, якщо число ділиться хоч на один дільник. У наведеному фрагменті роль змінної-прапорця відіграє рядок виведення в компонент `Label` (властивість `Caption`). У програмуванні прапорцем (`flag`) прийнято називати змінну, яка набуває одного з

двох значень: true або false, 0 або 1. У цьому випадку таке використання Caption цілком допустиме, бо вся задача будувалася для того, щоб вивести на екран це повідомлення. Про жодне інше використання одержаного результату не йшлося.

Завдання 2. Визначити, на який мінімальний множник потрібно помножити число 9, щоб запис результату складався тільки з цифр 2.

Розв'язання. За кодом програми задача така сама проста, як і попередня, однак алгоритм її розв'язування слід обдумати.

Розв'язувати задачу будемо методом від супротивного: припустимо, ми знайшли число, що складається тільки з цифр 2 і ділиться на 9. Тоді з усіх можливих таких чисел нам треба вибрати мінімальне. Реалізуємо це програмним шляхом. Виникає єдине питання: як зібрати числа із самих двійок (2, 22, 222, 2222 і т. д.)? Робиться це досить просто, однак необхідно стежити за межами (діапазонами) цілочислових типів даних — числа можуть сягати величезних значень. Тому скористаємося типом даних longint (cardinal).

У методі OnClick (єдиної кнопки на формі) записано такий код:

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

```
var
```

```
    chislo,stepin:longint;
```

```
begin
```

```
    chislo:=2;
```

```
    stepin:=10;
```

```
    while chislo mod 9 <>0 do
```

```
        begin
```

```
chislo:=chislo+2*stepin;
```

```
stepin:=stepin*10;
```

```
end;
```

```
Labell.caption:=inttostr(chislo div 9);
```

```
end;
```

Ця задача має елегантне аналітичне розв'язання. Ознакою подільності на 9 є той факт, що сума цифр числа ділиться на 9. Очевидно, що мінімальним із таких чисел є 222 222 222. Звідси випливає відповідь задачі: 222 222 222/9.

Примітка. Три запропоновані додатки дуже схожі один на одного й можуть бути одержані один з одного шляхом часткового змінення коду. Зберігати їх треба в різні папки.

ЗАДАЧІ ДЛЯ САМОСТІЙНОГО РОЗВ'ЯЗУВАННЯ

1. За допомогою функціонального ряду для введеного x обчислити значення $\arctg(x)$ з n першими числами послідовності:

$$\arctg(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \dots$$

Примітка. У цьому випадку на формі буде два Edit для введення p і x . Степінь x у чисельнику доведеться обчислювати за допомогою окремої змінної так само, як це робилося зі знаком доданка в розглянутому вище прикладі (див. рис. 9.4).

2. За допомогою функціонального ряду для введеного x обчислити значення $\cos(x)$ з p першими числами послідовності:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!}.$$

Примітка. У цьому випадку на формі буде два Edit для введення n і x . Степінь x у чисельнику доведеться обчислювати за допомогою окремої змінної, ще одна змінна буде потрібна для обчислення факторіала в знаменнику.

3. За допомогою функціонального ряду для введеного x обчислити значення $\sin(x)$ з n першими числами послідовності:

$$\sin(x) = 1 - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}.$$

Примітка. У цьому випадку на формі буде два Edit для введення n і x . Степінь x у чисельнику доведеться обчислювати за допомогою окремої змінної, ще одна змінна буде потрібна для обчислення факторіала в знаменнику.