### Завдання для виконання

1. Написати програму, яка обчислює значення числа ПІ з точністю, програми, що задається користувачем під час роботи.

У основі алгоритму обчислення лежить той факт, що сума ряду 1 - 1/3 + 1/5 -1/7 + 1/9 + ... наближається до значення ПИ/4 при достатньо великій кількості членів ряду.

Кожен член ряду з номером n обчислюється за формулою: 1/(2\*n - 1) і умножається на мінус один, якщо n парне (визначити, чи є ПІ парним, можна перевіркою залишку від ділення ПІ на 2). Обчислення закінчується тоді, коли значення чергового члена ряду стає меншим, ніж задана точність обчислення.

Вид діалогового вікна програми під час її роботи приведений на Рис. 3. Користувач вводить точність обчислення в поле введення (Edit1). Після клацання на командній кнопці Обчислити (Buttonl) програма обчислює значення числа л і виводить результат в полі мітки (Label1).

Як і в попередніх прикладах, основну роботу виконує процедура обробки події OnClick.

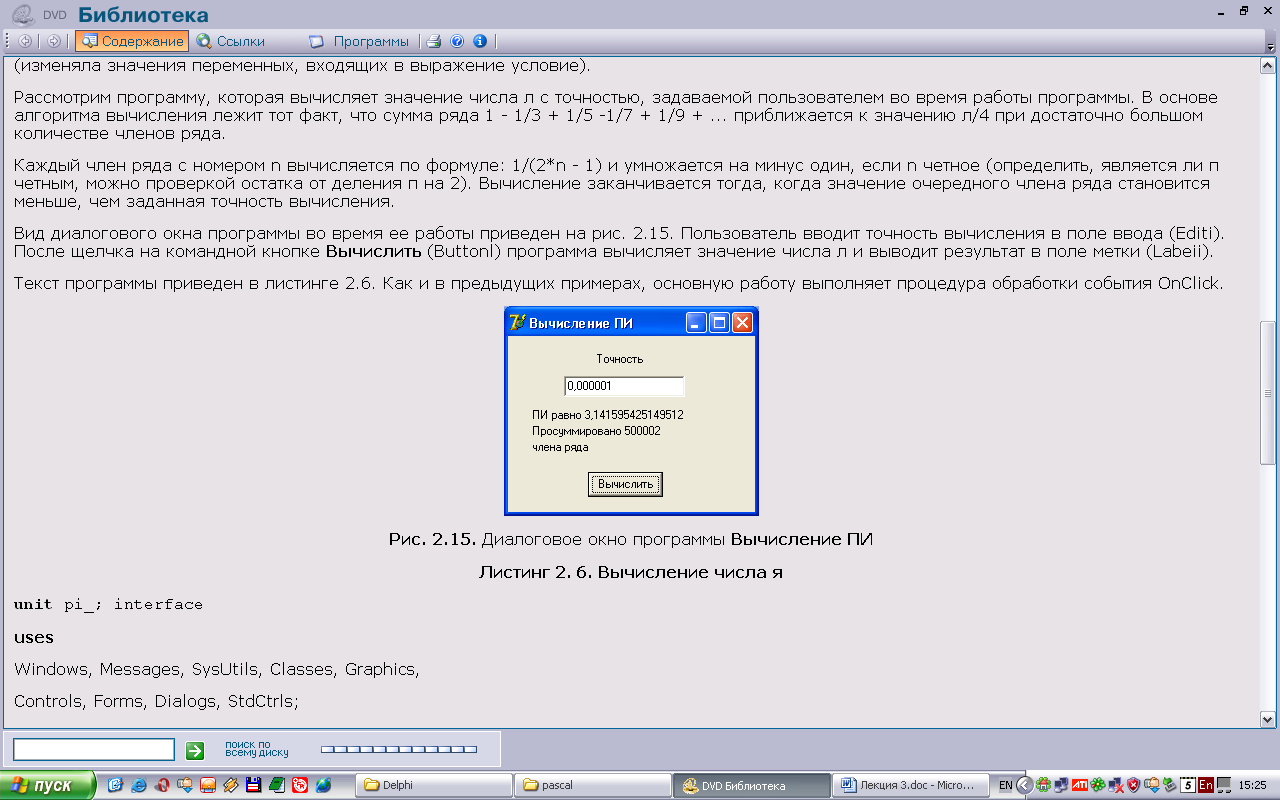


Рисунок 5 Діалогове вікно програми Обчислення ПІ

**Лістинг обчислення числа ПІ**

unit pi\_; interface

uses

Windows, Messages, SysUtils, Classes, Graphics

Controls, Forms, Dialogs, StdCtrls;

type

TForm1 = class(TForm)

Edit1: TEdit; // точність обчислення

Button1: TButton; // кнопка Обчислити

Label1: TLabel;

Label2: TLabel; // поле виведення результату

procedure ButtonlClick(Sender: TObject); private

{ Private declarations }

public

{ Public declarations )

end;

var

Form1: TForm1;

implementation

{$R \*.DFM}

procedure TForm1.Button1Click(Sender: TObject);

var

pi:real; // обчислюване значення ПІ

t:real; // точність обчислення

n:integer; // номер члена ряду

elem:real; // значення члена ряду

begin

pi := 0;

n := 1;

t := StrToFloat(editl.text);

elem := 1; // щоб почати цикл

while elem >= t do

begin

elem := 1 / (2\*n - 1) ; if n MOD 2=0

then pi := pi — elem else pi := pi + elem; n := n + 1;

end;

pi: = pi \* 4; labell.caption:= 'ПІ рівне '+ FloatToStr(pi)+ #13

+ 'Підсумовано '+IntTostr(n)+' членів ряду.'; end;

end.

1. Написати програму, яка перевіряє, чи є введене користувачем число простим (як відомо, число називається простим, якщо воно ділиться тільки на одиницю і саме на себе).

Наприклад, число 21 — звичайне (ділиться на 3), а число 17 — простої (ділиться тільки на 1 і на 17).

Перевірити, чи є число n простим, можна діленням числа n на два, на три і т.д. до n і перевірки залишку після кожного ділення. Якщо після чергового ділення залишок рівний нулю, то це означає, що знайдено число, на якому n ділиться без залишку. Порівнявши n і число, на якому n розділилося без залишку, можна визначити, чи є n простим числом.

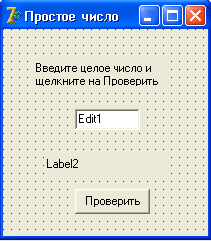


Рисунок 6 Форма застосування Просте число

**Лістинг програми**

unit simple\_;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs

StdCtrls;

type

TForm1 = class(TForm)

Button1: TButton; // кнопка Перевірити

Label1: TLabel;

Edit1: TEdit; // поле введення числа

Label2: TLabe1; // поле виведення результату

procedure ButtonlClickfSender: TObject); private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.DFM}

procedure TForm1.ButtonlClick(Sender: TObject) ;

var

n: integer; // число d, що перевіряється: integer; // дільник

r: integer; // залишок від ділення n на d

begin

n:=StrToInt(Editl.text);

d := 2; // спочатку ділитимемо на два

repeat

r := n mod d;

if r <> 0 // n не розділилося без остачі на d

then d := d + 1;

until r = 0; // знайдене число, на якому п розділилося без залишку

label2.caption:=Edit1.text;

if d = n

then Iabel2.caption:=label2.caption + ' — просте число.'

else label2.caption:=label2.caption + ' — звичайне число.';

end;

end.