

Практична робота

№ 10

Тема: «Створення власних класів. Наслідування і віртуальні функції.

Інтерпретатор математичних виразів»

Виконав: Устич Максим

Група: alk43

Київ 2025р.

Мета роботи

Навчитися створювати та використовувати власні класи, нащадки, віртуальні функції та реалізовувати деревоподібні структури для обчислення математичних виразів.

Хід роботи

Елемент	Тип	Призначення
GroupBox1	TGroupBox	Об'єднує поля для введення виразів. Підпис «Вирази».
GroupBox2	TGroupBox	Об'єднує поля для відображення результатів. Підпис «Результати обчислення».
GroupBox3	TGroupBox	Об'єднує елементи керування точністю обчислення. Підпис «Точність».
Edit1	TEdit	Поле для введення значення змінної x.
Edit2	TEdit	Поле для введення виразу $f(x)$ (наприклад, $2*x+5$).
Edit3	TEdit	Поле для відображення результату обчислення x.
Edit4	TEdit	Поле для відображення результату $f(x)$.
ComboBox1	TComboBox	Вибір кількості знаків після коми (точність).
Button1	TButton	Кнопка «Обчислити» — запускає обчислення.
Button2	TButton	Кнопка «Вихід» — завершує роботу програми.

Створення головної форми

На головній формі розміщено такі компоненти:

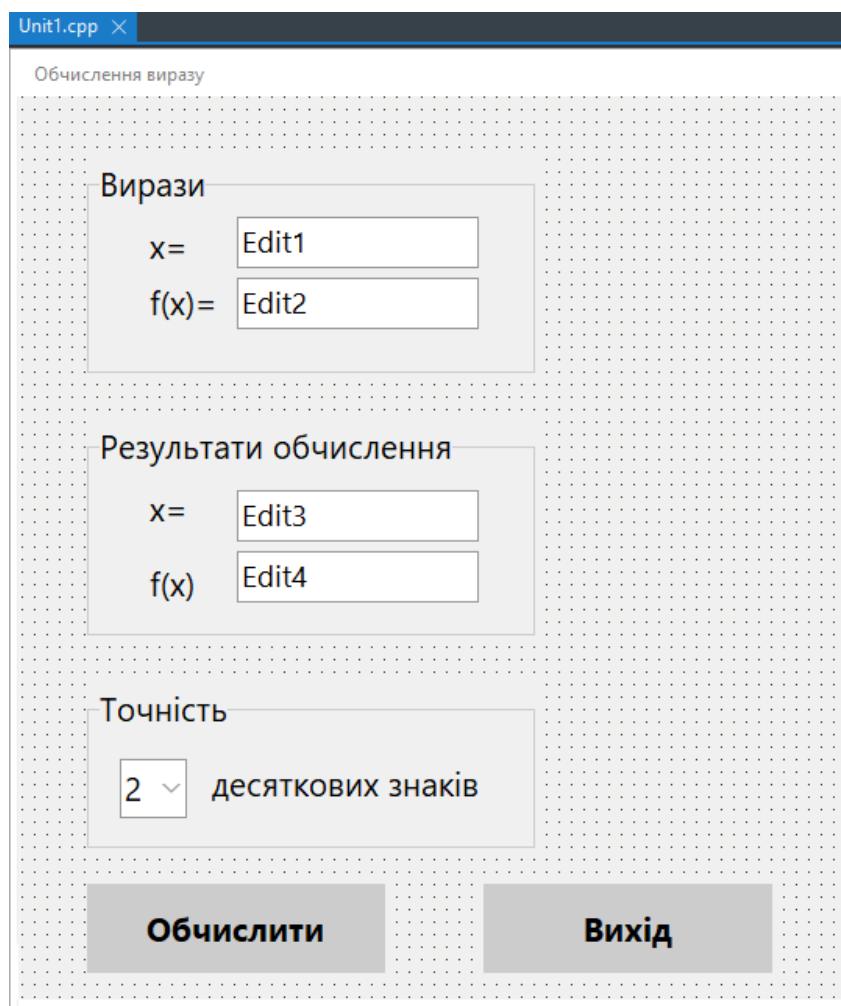


Рис. 1. Головна форма програми “Обчислення виразу” (до запуску).

Код програми Unit1.h

```
1 //-----
2 .
3 . #ifndef Unit1H
4 . #define Unit1H
5 //-----
6 .
7 . #include <System.Classes.hpp>
8 . #include <Vcl.Controls.hpp>
9 . #include <Vcl.StdCtrls.hpp>
10 #include <Vcl.Forms.hpp>
11 .
12 //-----
```

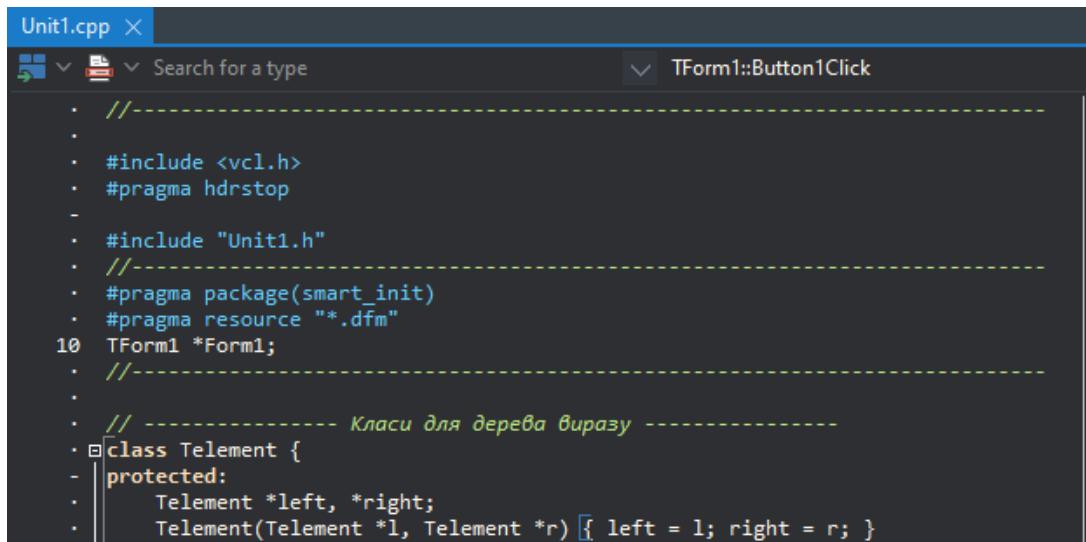
```

·   class TForm1 : public TForm
·   {
·       __published:    // IDE-managed Components
·           TGroupBox *GroupBox1;
·           TGroupBox *GroupBox2;
·           TEdit *Edit1;
·           TLabel *Label1;
·           TEdit *Edit2;
·           TLabel *Label2;
·           TLabel *Label3;
·           TLabel *Label4;
·           TEdit *Edit3;
·           TEdit *Edit4;
·           TGroupBox *GroupBox3;
·           TComboBox *ComboBox1;
·           TLabel *Label5;
·           TButton *Button1;
·           TButton *Button2;
·
·           // Події кнопок і зміни точності
·           void __fastcall Button1Click(TObject *Sender);
·           void __fastcall Button2Click(TObject *Sender);
·           void __fastcall ComboBox1Change(TObject *Sender);
·
·       private:     // User declarations
·       public:      // User declarations
40        __fastcall TForm1(TComponent* Owner);
·   };
·
·   //-----
·
·   extern PACKAGE TForm1 *Form1;
·
·   //-----
·   #endif

```

Рис. 2. Код заголовкового файлу Unit1.h.

Код програми Unit1.cpp



```

Unit1.cpp X
File View Search for a type
TForm1::Button1Click
·   //-
·
·   #include <vcl.h>
·   #pragma hdrstop
·
·   #include "Unit1.h"
·   //-
·   #pragma package(smart_init)
·   #pragma resource "*.dfm"
10  TForm1 *Form1;
·   //-
·
·   // ----- Класи для дерева виразу -----
·   class Telement {
·       protected:
·           Telement *left, *right;
·           Telement(Telement *l, Telement *r) { left = l; right = r; }

```

```

.     public:
.         virtual ~Telement() { delete left; delete right; }
20     virtual float result(void) { return 0; }
.     };
.
.     class Number : public Telement {
.         float f;
.     public:
.         Number(float F) : Telement(NULL, NULL) { f = F; }
.         float result(void) { return f; }
.     };
.
30     class Plus : public Telement {
.     public:
.         Plus(Telement *l, Telement *r) : Telement(l, r) {}
.         float result(void) { return left->result() + right->result(); }
.     };
.
.     class Mult : public Telement {
.     public:
.         Mult(Telement *l, Telement *r) : Telement(l, r) {}
.         float result(void) { return left->result() * right->result(); }
40     };
.
.
. // ----- функція побудови дерева -----
. Telement* form(AnsiString s)
. {
.     Telement *h;
.     int p;
.     int l = s.Length();
.     AnsiString s1, s2;
.
50     if ((p = s.Pos("+")) > 1) {
.         s1 = s.SubString(1, p - 1);
.         s2 = s.SubString(p + 1, l - p);
.         h = new Plus(form(s1), form(s2));
.     }
.     else if ((p = s.Pos("*")) > 1) {
.         s1 = s.SubString(1, p - 1);
.         s2 = s.SubString(p + 1, l - p);
.         h = new Mult(form(s1), form(s2));
.     }
.     else if (s == "x") {
.         h = new Number(StrToInt(Edit1->Text));
.     }
.     else {
.         h = new Number(StrToInt(s));
.     }
.     return h;
. }
.
. // ----- Реалізація форми -----
70     __fastcall TForm1::TForm1(TComponent* Owner)
.     : TForm(Owner)
.     {};
.
.     // Кнопка "Обчислити"
.     void __fastcall TForm1::Button1Click(TObject *Sender)
.     {
.         int dec = StrToInt(ComboBox1->Text);
.
80         Telement *x = form(Edit1->Text);
.         Edit3->Text = FloatToStrF(x->result(), fffFixed, 25, dec);
.
.         Telement *fx = form(Edit2->Text);
.         Edit4->Text = FloatToStrF(fx->result(), fffFixed, 25, dec);
84     };

```

```

    :
    delete x;
    delete fx;
    }

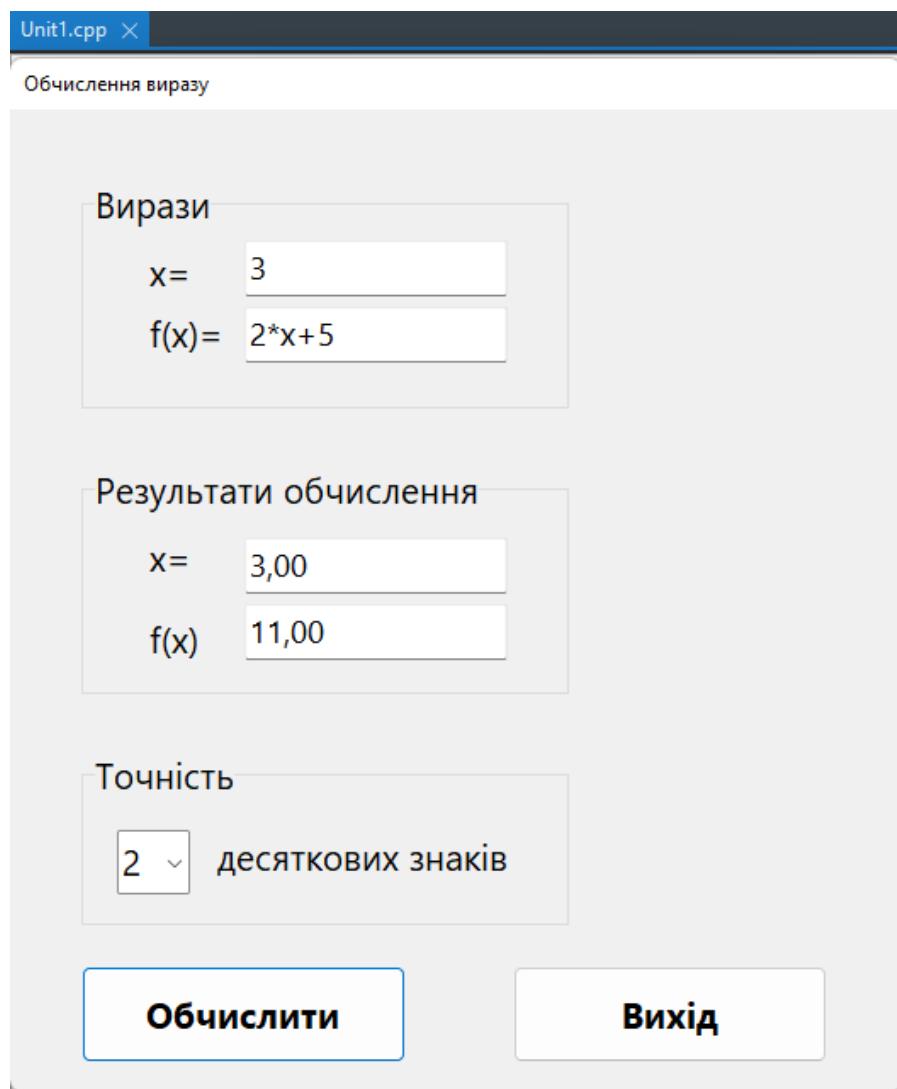
90 // Кнопка "Вихід"
91 void __fastcall TForm1::Button2Click(TObject *Sender)
92 {
93     Close();
94 }

// Зміна точності
95 void __fastcall TForm1::ComboBox1Change(TObject *Sender)
96 {
97     Button1Click(Button1);
98 }
99 //-----

```

Рис. 3. Основний код програми Unit1.cpp.

Результати виконання програми



*Рис. 4. Результат роботи програми – обчислення виразу $f(x)=2*x+5$ при $x=3$.*

Відповіді на контрольні запитання

1. Дайте визначення основним поняттям: клас, інкапсуляція, поліморфізм, успадкування.

- **Клас** — це визначений користувачем тип даних, який об'єднує в собі дані (поля) та методи (функції), що з ними працюють.
- **Інкапсуляція** — це принцип об'єктно-орієнтованого програмування, який приховує внутрішню реалізацію об'єкта та надає доступ лише через публічні методи.
- **Поліморфізм** — здатність об'єктів різних класів по-різному реагувати на один і той самий виклик функції або методу.
- **Успадкування** — це механізм створення нових класів на основі вже існуючих із можливістю розширення або зміни їхньої функціональності.

2. Що називається елементами класу?

Елементами класу називають складові частини його опису:

- **поля (дані)** — змінні, що зберігають стан об'єкта;
- **методи (функції)** — операції, що виконують дії над даними класу.

3. Надайте загальний вид оголошення класу.

```
class ClassName {  
    private:  
        // поля (дані)
```

```
public:  
    // методи (функції)  
};
```

4. З яких частин складається специфікація класу?

Специфікація класу складається з таких частин:

- опис області видимості (private, protected, public);
- оголошення полів (даних);
- оголошення методів (функцій);
- конструкторів і деструкторів;
- (за потреби) оголошення друзів класу (friend).

5. Які існують специфікатори доступу елементів класу?

- **private** — доступ лише зсередини самого класу;
- **protected** — доступ із класу та його нащадків;
- **public** — відкритий доступ з будь-якої частини програми.