

Лабораторна робота № 16

Тема. Одночасне використання кількох об'єктів декількох споріднених класів в об'єктно-орієнтованому програмуванні за допомогою вказівок.

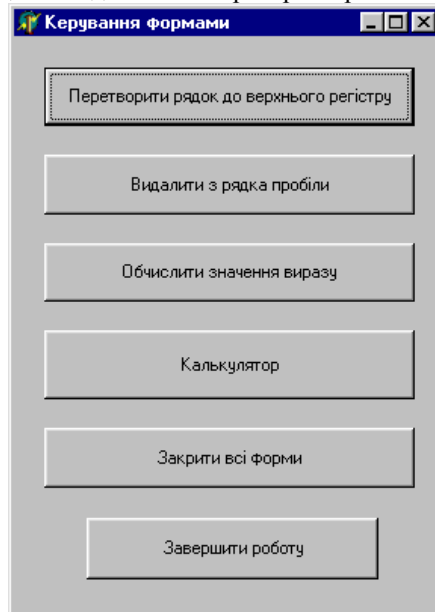
Мета. Формування вмінь і навиків використання форм та класів раніше розроблених проектів. Закріплення вмінь і навиків використання функцій стандартних бібліотек, власних підпрограм, класів, об'єктів, функцій вводу-виводу та обробки рядків. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання.

1. Що зберігається в пам'яті для класу і для об'єкта?
2. Як створити і знищити об'єкт класу?
3. Як перенести в новий проект розроблену раніше форму?
4. Як перевірити існування об'єкта класу форми?
5. Що необхідно зробити після створення об'єкта класу, породженого від форми, для забезпечення можливості використання його користувачем?

Завдання.

1. Створіть форму з кнопками для керування раніше розробленими формами. Для цього:
 - 1.1. У новому рішенні *LR13MenuForm* розробіть зовнішній вигляд форми за зразком (надписи на кнопках мають відповідати завданням лабораторних робіт №№ 9-11):



Закрийте це рішення;

- 1.2. Для уникнення конфліктів у назвах форм по чергово відкрийте рішення лабораторних робіт №№ 9-11 та задайте в панелі оглядача рішень унікальні назви файлам форм і погодьтеся з перейменуванням пов'язаних даних. З цією метою в панелі оглядача рішень відмітьте файл форми (з розширенням *.cs*), перейменуйте його з *Form1* на іншу згідно з її призначенням (наприклад, *Form1.cs* на *FormCalc.cs*, *FormStr.cs* чи *FormMath.cs*) і погодьтеся з автоматичним перейменуванням пов'язаних посилань. Як при цьому змінилася властивість *Имя файла* у вікні властивостей? Чи змінилася властивість *Name* для кожної форми? Якщо властивість *Name* для якоїсь форми не змінилася (це відбувається тоді, коли простір імен не є простором імен проекту по замовчуванню), то змініть її згідно імені файла безпосередньо у вікні властивостей в режимі конструктора або у вікні коду форми, обираючи в контекстному меню її назви пункт *Выполнить рефакторинг – Переименовать*. Чи змінилися назви просторів імен у вікнах коду форм?
- 1.3. Відкрийте одну з перейменованих форм в режимі конструктора і змініть її властивість *Name*. Чи змінилися при цьому назва файла, назва простору імен, назви пов'язаних даних, назва форми у вікні коду? Відновіть значення властивості *Name* згідно назви файла. Закрийте ці рішення;
- 1.4. Додайте перейменовані модулі форм лабораторних робіт №№ 9-11 (як з розширенням *.cs*, так і пов'язані файли) до нового рішення кнопкової форми. Для цього завантажте рішення кнопкової форми *LR13Menu*, в панелі оглядача рішень виділіть проект *LR13Menu* (з позначкою *C#*) та

почергово в його контекстному меню оберіть пункт *Добавить – Существующий элемент* і додайте три перейменовані форми попередніх лабораторних робіт з розширенням *.cs* (наприклад, *FormCalc.cs*). Які ще файли додалися при цьому автоматично?

- 1.5. У вікні програми кнопкової форми відкрийте простори імен скопійованих форм, дописуючи у розділі описів їх назви (наприклад, *using LR11Calc;*);
- 1.6. На початку опису класу кнопкової форми *FormCalc* додайте вказівки на класи скопійованих форм, вказавши у наступних рядках після рядка-заголовка класу кнопкової форми клас кожної скопійованої форми та назву відповідної вказівки. Занесіть у ці вказівки початкове значення *null*. При цьому початок коду кнопкової форми має бути подібним до такого:

```
public partial class Form1 : Form
{
    FormCalc f1=null;
    ...
    public Form1()
    {
        InitializeComponent();
    }
    ...
}
```

- 1.7. В процедурі обробки події натиснення кнопки виклику калькулятора забезпечте створення, а при існуванні – фокусування відповідної форми. При цьому код обробки цієї події має бути приблизно таким:

```
private void button4_Click(object sender, EventArgs e)
{
    if (f1==null)
    {
        f1 = new FormCalc();
        f1.Show();
    }
    else
    {
        if (f1.IsDisposed)
        {
            f1 = new FormCalc();
            f1.Show();
        }
        else
        {
            if (f1.WindowState == FormWindowState.Minimized)
                f1.WindowState = FormWindowState.Normal;
            else
            {
                if (!f1.Focused)
                    f1.Focus();
            }
        }
    }
}
```

- 1.8. Аналогічно забезпечте завантаження відповідних форм класів інших підключених модулів для інших (крім двох останніх) кнопок головної кнопкової форми;
- 1.9. Для закриття відкритих форм створіть у кнопковій формі таку загальну процедуру:

```
private void closeForm()
{
    if (f1!=null)
    {
        if (!f1.IsDisposed)
        {
            f1.Close();
            f1=null;
        }
        else
        {
            f1=null;
        }
    }
    // інші форми знищуються аналогічно
}
```

- 1.10. Забезпечте закриття відкритих форм при натисненні передостанньої і останньої кнопки та закритті кнопкової форми користувачем.