

## Лабораторна робота № 17

- Тема.** Одночасне використання багатьох об'єктів декількох споріднених класів в об'єктно-орієнтованому програмуванні за допомогою масивів. Поліморфні виклики методів об'єктів.
- Мета.** Формування вмінь і навиків створення та використання багатьох об'єктів декількох споріднених класів. Закріплення вмінь і навиків використання модулів, 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. У чому полягає суть властивості поліморфізму?
2. Як створити і вивести на екран форму заданого класу?
3. Як організувати доступ до багатьох об'єктів за допомогою масиву? Який тип даних його елементів? Чому?
4. Чи може одна змінна використовуватися для керування багатьма об'єктами? Чому?
5. Які способи знищення форм ви знаєте?
6. Коли і як знищуються всі виведені форми?

### Завдання.

1. **Вдоскональте головну кнопочку форму, розроблену на попередній лабораторній роботі, забезпечивши керування багатьма об'єктами декількох споріднених класів. Для цього:**

- 1.1. На початку опису класу головної кнопочкової форми оголошіть константу максимальної кількості створених форм-об'єктів та масив вказівок на об'єкти. При створенні головної кнопочкової форми занесіть у всі елементи цього масиву значення *null*. Наприклад, так:

```
public partial class Form1 : Form
{
    const int maxForm=18;
    Form[] masForm=new Form[maxForm];

    public Form1()
    {
        InitializeComponent();
        for (int i = 0; i < maxForm; i++)
            masForm[i] = null;
    }
}
```

- 1.2. Реалізуйте загальну процедуру для створення об'єктів обраного класу, наприклад, таку:

```
private void createForm(int index)
{
    int i = 0;
    while (i<maxForm)
    {
        if (masForm[i]==null)
            break;
        else
        {
            if (masForm[i].IsDisposed)
                break;
            else
                i++;
        }
        if (i==maxForm)
        {
            MessageBox.Show("Більше форм створити неможливо");
            return;
        }
        switch (index)
        {
            case 1:masForm[i]= new FormCalc();
                    break;
            case 2:masForm[i] = new FormString();
                    break;
            _ // інші об'єкти створюються аналогічно
        }
        masForm[i].Show();
    }
}
```

Забезпечте її виклики з різними значеннями параметрів для різних кнопок головної кнопочкової форми. Наприклад, в процедурі обробки натиснення кнопки завантаження калькулятора потрібно вказати *createForm(1)*;

- 1.3. Модифікуйте процедуру для знищення всіх відкритих створених раніше форм-об'єктів, наприклад, так:

```
private void button2_Click(object sender, EventArgs e)
{for (int i = 0; i < maxForm;i++ )
  if (masForm[i]!= null)
    if (!masForm[i].IsDisposed)
      {masForm[i].Close();
        masForm[i] = null;
      }
}
```

- 1.4. Створіть додаткові кнопки і відповідні процедури для переміщення і зміни розміру всіх створених форм-об'єктів;
- 1.5. Реалізуйте власні процедури додаткових групових операцій над формами-об'єктами.