

Лабораторна робота № 19

- Тема.** Використання інтерфейсів та делегатів.
Мета. Формування вмінь і навиків створення інтерфейсів та делегатів засобами C#. Закріплення вмінь і навиків наслідування звичайних та абстрактних класів, використання об'єктів, підпрограм, елементів керування. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання.

1. Що спільного між абстрактними класами та інтерфейсами? Чим вони відрізняються?
2. Як описуються інтерфейси?
3. Які специфікатори доступу і де можливо вказувати для інтерфейсів? Чому для членів інтерфейсів не вказуються специфікатори доступу?
4. Від скількох інтерфейсів може наслідуватися клас? Де при цьому вказується базовий клас, якщо від наявний?
5. Як використовується наслідування різних класів від одного інтерфейсу?
6. Які об'єкти називаються делегатами? Як використовуються делегати?

Завдання.

1. **Вдоскональте комплекс форм, розроблених на попередній лабораторній роботі, забезпечивши зберігання та використання тем лабораторних робіт, на яких вони були створені. Для цього:**
 - 1.1. Створіть нову папку для рішення цієї лабораторної роботи. Скопіюйте в неї рішення попередньої лабораторної роботи. Переіменуйте скопійоване рішення згідно теми цієї лабораторної роботи;
 - 1.2. Відкрийте скопійоване рішення. Змініть назву класу форми головного меню на *FormMenu*. У файлі коду головного меню під описом класу форми опишіть інтерфейс для зберігання тем лабораторних робіт всіма пов'язаними формами. Цей код може бути, наприклад, таким:

```
namespace LR19ListInterface
{
    public partial class FormMenu : Form...

    interface ILabWork
    {
        string GetTemaLabWork();
    }
}
```

Який рівень доступу описаного інтерфейсу?

- 1.3. Забезпечте підтримку розробленого інтерфейсу всіма пов'язаними формами. Для цього, по-перше, вкажіть назву цього інтерфейсу серед батьківських об'єктів цих форм і, по-друге, доповніть їх публічними методами *GetTemaLabWork()*. Тоді, наприклад, початок коду форми калькулятора може бути таким:

```
public partial class FormCalc : Form, ILabWork
{
    bool resultTablo = true;
    string operation = "";

    public FormCalc()
    {
        InitializeComponent();
    }

    public string GetTemaLabWork()
    {
        return "Організація взаємодії елементів керування у формах";
    }
}
```

...

- 1.4. Доповніть головну кнопочку форму кнопочку *Підрахувати кількості форм по темах*. Підтримка спільного інтерфейсу всіма пов'язаними формами дає можливість виконувати явне перетворення до нього всіх форм зі списку, створених користувачем. Процедура її натиснення може бути, наприклад, такою:

```

private void button8_Click(object sender, EventArgs e)
{
    List<ILabWork> ListAnaliz = new List<ILabWork>();
    foreach (Form f in listForm)
        ListAnaliz.Add((ILabWork)f);
    string res = "", tema;
    int index, count;
    while (ListAnaliz.Count > 0)
    {
        tema = ListAnaliz[0].GetTemaLabWork();
        index = 0; count = 0;
        while (index < ListAnaliz.Count)
            if (ListAnaliz[index].GetTemaLabWork() == tema)
            {
                count++;
                ListAnaliz.RemoveAt(index);
            }
            else
                index++;
        res += tema + " - " + count.ToString() + " форм\n";
    }
    if (res == "")
        res = "Форми відсутні";
    MessageBox.Show(res, "Статистика форм",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
}

```

2. Створіть в головній кнопковій формі делегат для порівняння двох форм за назвою теми. Доповніть головну кнопкову форму кнопкою *Відсортувати за темою і розмістити каскадом*. Додатковий код може бути, наприклад, таким:

```

public int SortByTema(Form f1, Form f2)
{
    if (f1 == null || f2 == null)
        throw new ArgumentException("Окремі аргументи не є формами");
    return String.Compare((f1 as ILabWork).GetTemaLabWork(),
        (f2 as ILabWork).GetTemaLabWork());
}

```

```

private void button9_Click(object sender, EventArgs e)
{
    listForm.Sort(SortByTema);
    button5_Click(this, new EventArgs());
}

```

3. Доповніть головну кнопкову форму кнопкою *Відсортувати за заголовком форми і розмістити каскадом* та самостійно забезпечте її функціонування.