

Лабораторна робота № 20

Тема. Вкладені класи. Використання інтерфейсів для порівняння та сортування об'єктів.

Мета. Формування вмінь і навиків використання інтерфейсів для порівняння та сортування об'єктів. Закріплення вмінь і навиків наслідування звичайних та абстрактних класів засобами C#, використання об'єктів, підпрограм, елементів керування. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

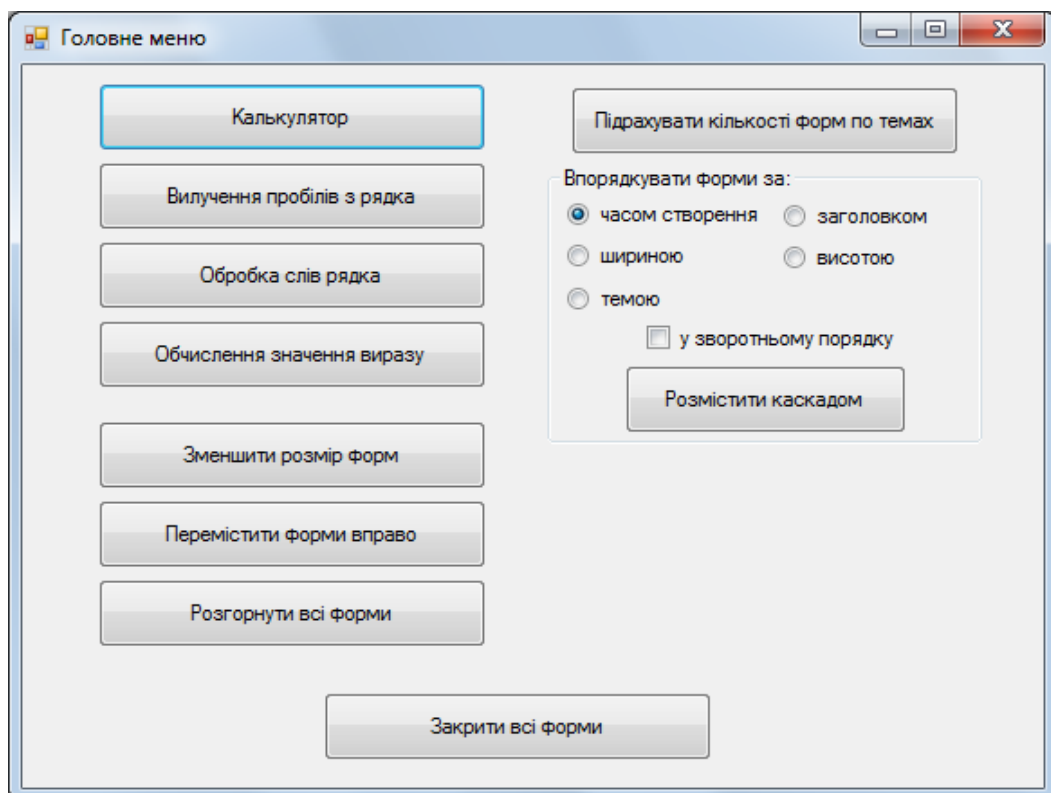
Контрольні запитання.

1. Які дві основні ролі класів? Які дві основні ролі вкладених класів?
2. Як організувати доступ до членів вкладеного класу? До яких членів класу можна звертатися без створення об'єкта класу?
3. Які основні переваги методу *Sort()* реалізовані для списків в C#? В чому між ними принципова різниця?
4. Який інтерфейс має підтримувати клас для можливості порівняння його об'єктів між собою по замовчуванню? Який метод при цьому потрібно додатково реалізувати?
5. Як реалізувати сортування об'єктів класу за різними критеріями з використанням вкладених класів? Який інтерфейс має підтримувати вкладений клас для можливості порівняння об'єктів основного класу між собою?

Завдання.

1. **Вдоскональте зовнішній вигляд головної кнопкової форми комплексу форм, розроблених на попередній лабораторній роботі, забезпечивши можливість впорядкування форм за різними критеріями під час їх розміщення каскадом.** Для цього доповніть головну кнопкову форму групою взаємозалежних перемикачів для сортування форм. З цією метою виберіть на панелі елементів відокремлену групу з надписом *GroupBox* та відмітьте у формі прямокутну область для розміщення відповідного елемента керування. Після цього виберіть на панелі елементів перемикач *RadioButton* та натягніть створеній групі декілька перемикачів для сортування зображених об'єктів за різними критеріями. Серед критеріїв сортування має обов'язково бути за *часом створення*. Встановіть властивість *Checked* цього перемикача в значення *true*, а всіх інших – в значення *false*. Для чого встановлювати цю властивість? Встановіть назви перемикачам англійською мовою згідно їх призначення. Натягніть у створеній групі також прапорець *checkBox* та дайте йому назву *sortDesc* для сортування форм за спаданням.

Після цього вигляд головної кнопкової форми може бути, наприклад, таким:



2. Для забезпечення можливості сортування форм за темою лабораторної роботи скористайтесь розробленим делегатом з попередньої лабораторної роботи та врахуйте, що при сортуванні за спаданням достатньо змінити на протилежний порядок форм (реверсувати), впорядкованих за зростанням. Тоді процедура обробки натиснення кнопки для впорядкування форм каскадом та делегат набуде, наприклад, такого вигляду:

```
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 button5_Click(object sender, EventArgs e)
{
    //сортування списку форм
    if (sortTema.Checked)
        listForm.Sort(SortByTema);
    if (sortDesk.Checked)
        listForm.Reverse();
    //вивід форм каскадом
    int left = 200, top = 200;
    for (int i = 0; i < listForm.Count; i++)
        if (!listForm[i].IsDisposed)
        {
            listForm[i].Left = left;
            left += 20;
            listForm[i].Top = top;
            top += 15;
            listForm[i].Focus();
        }
}
```

3. З метою забезпечення можливості сортування форм за часом створення по замовчуванню:
- 3.1. Реалізуйте збереження моменту часу створення форми в одній з її властивостей, яка не використовується (наприклад, *Tag*) в процедурі створення форм:

```
private void createForm(int index)
{
    Form f = null;
    switch (index)
    {
        case 1: f = new FormCalc();
                break;
                ~~~~~
    }
    f.Tag = DateTime.Now.Ticks.ToString();
    listForm.Add(f);
    f.Show();
}
```

- 3.2. Для забезпечення можливості порівняння створюваних форм між собою по замовчуванню реалізуйте підтримку ними інтерфейсу *IComparable*. З цією метою створіть новий клас, породжений від класу *Form* з підтримкою цього інтерфейсу та реалізуйте в ньому метод порівняння поточного об'єкта з зовнішнім за значенням, збереженим у попередньому підпункті:

```

public class MForm : Form, IComparable
{
    int IComparable.CompareTo(object c)
    {
        long t1 = Convert.ToInt64(this.Tag);
        long t2 = Convert.ToInt64(((Form)c).Tag);
        if (t1 < t2)
            return -1;
        if (t1 > t2)
            return 1;
        return 0;
    }
}

```

- 3.3. В імпортованих формах *всіх* попередніх лабораторних робіт (*FormCalc*, *FormString* та інших) змініть батьківський клас на клас попереднього підпункту, забезпечивши цим самим підтримку інтерфейсу *IComparable*, наприклад, так:

```

public partial class FormCalc : MForm, ILabWork

```

- 3.4. Доповніть процедуру обробки натиснення кнопки для впорядкування форм каскадом обробкою перемикача з надписом за *часом створення*, викликаючи при цьому стандартний метод *Sort()* списку форм:

```

private void button5_Click(object sender, EventArgs e)
{
    //сортування списку форм
    if (sortTime.Checked)
        listForm.Sort();
    if (sortTema.Checked)
        listForm.Sort(SortByTema);
    if (sortDesk.Checked)
        listForm.Reverse();
    //вивід форм каскадом
    int left = 200, top = 200;
    for (int i = 0; i < listForm.Count; i++)
        if (!listForm[i].IsDisposed)
        {
            listForm[i].Left = left;
            left += 20;
            listForm[i].Top = top;
            top += 15;
            listForm[i].Focus();
        }
}

```

4. З метою сортування форм за висотою:

- 4.1. Опишіть у класі *головної кнопочової форми* вкладений клас з підтримкою інтерфейсу *IComparer*. Реалізуйте у цьому класі цілочисельний метод *Compare(Form f1, Form f2)* для підтримки інтерфейсу *IComparer*;
- 4.2. Доповніть процедуру обробки натиснення кнопки для впорядкування форм каскадом обробкою перемикача з надписом за *висотою*, викликаючи при цьому метод сортування списку форм з новим екземпляром вкладеного класу:

```

public class SortByHeight : IComparer<Form>
{
    public int Compare(Form f1, Form f2)
    {
        if (f1 == null || f2 == null)
            return 0;
        if (f1.Height < f2.Height)
            return -1;
        if (f1.Height > f2.Height)
            return 1;
        return 0;
    }
}

```

```

private void button5_Click(object sender, EventArgs e)
{
    //сортування списку форм
    if (sortTime.Checked)
        listForm.Sort();
    if (sortTema.Checked)
        listForm.Sort(SortByTema);
    if (sortHeight.Checked)
        listForm.Sort(new SortByHeight());
    if (sortDesk.Checked)
        listForm.Reverse();
    //вивід форм каскадом
    int left = 200, top = 200;
    for (int i = 0; i < listForm.Count; i++)
        if (!listForm[i].IsDisposed)
        {
            listForm[i].Left = left;
            left += 20;
            listForm[i].Top = top;
            top += 15;
            listForm[i].Focus();
        }
}

```

5. Самостійно забезпечте сортування форм за шириною та за заголовком (властивість *Text*) під час впорядкування каскадом.