

## Лабораторна робота № 24

**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.

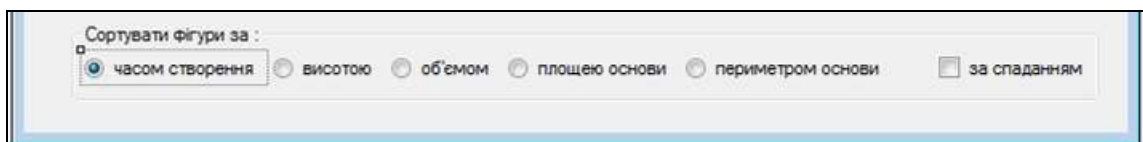
### Контрольні запитання.

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 перевантаження 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? 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. Відкрийте розроблений раніше додаток-форму з описом базового та породжених класів згідно варіанту і методами *draw3D()* для їх зображення. В режимі конструктора форми Встановіть її властивість *DoubleBuffered* рівною *true* для забезпечення швидшого виводу рухомих зображень. Доповніть її також групою взаємозалежних перемикачів для сортування зображених об'єктів. Для цього 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 у 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? Натягніть у створеній групі також прапорець *checkBox* та дайте йому назву *checkBoxDesc* для сортування за спаданням.

Вигляд цієї групи перемикачів може бути, наприклад, таким:



2. Для забезпечення можливості сортування всіх зображених об'єктів за часом створення:
  - 2.1. Задекларуйте підтримку базовим батьківським класом *Figure2D* інтерфейсу *IComparable*;
  - 2.2. Створіть у базовому батьківському класі статичне цілочисельне поле для збереження порядкового номера створеного об'єкта. Забезпечте встановлення значення цього поля в конструкторі базового батьківського класу аналогічно полю-лічильнику;
  - 2.3. Реалізуйте в базовому батьківському класі метод *CompareTo(object c)* для підтримки інтерфейсу *IComparable*;
  - 2.4. В процедурі обробки події *CheckedChanged* перемикача з надписом за *часом створення* забезпечте виклик методу *Sort()* створеного списку об'єктів та перемальовування форми;
  - 2.5. Переіменуйте цю процедуру обробки події в модулі форми та автоматично – у всіх посиланнях на неї, вказавши їй довільну назву, пов'язану з сортуванням.

Наприклад, базовий батьківський клас форми для відображення геометричних фігур набуде такого вигляду:

```

abstract class Figure2D:IComparable
{static int nextNomer=1;
  int nomer;

  public Figure2D()
  {nomer = nextNomer;
   nextNomer++;
  }

  int IComparable.CompareTo(object c)
  {Figure2D f = c as Figure2D;
   if (f != null)
    {if (this.nomer > f.nomer) return 1;
     if (this.nomer < f.nomer) return -1;
     return 0;
    }
   throw new ArgumentException("Аргумент не підтримує класу Figure2D");
  }

  public abstract double Area();
  public abstract double Perimeter();
  public abstract void Info();
}

```

...

А процедура обробки події *CheckedChanged* перемикача з надписом за часом створення може бути такою:

```

private void SortCheckedChanged(object sender, EventArgs e)
{if (radioButton1.Checked)
  {ListDrawFigure3D.Sort();
   Refresh();
  }
}

```

### 3. З метою сортування зображуваних об'єктів за різними критеріями:

- 3.1. Опишіть у класі форми вкладені класи, наслідувані від інтерфейсу *IComparer* для типу інтерфейсу *IDrawFigure3D*. Реалізуйте у цих класах цілочисельні методи *Compare(IDrawFigure3D p1, IDrawFigure3D p2)* для підтримки інтерфейсу *IComparer*;
- 3.2. Забезпечте при виникненні події *CheckedChanged* решти перемикачів виклик узагальненої процедури сортування. В цій процедурі запрограмуйте виклик методу *Sort()* для об'єкта відповідного вкладеного класу.

Наприклад, для сортування відображених геометричних фігур за висотою, об'ємом, площею чи периметром основи доцільно у класі форми створити такі вкладені класи:

```

public class SortByH : IComparer<IFigure3D>
{public int Compare(IFigure3D p1, IFigure3D p2)
//або без public з явним оголошенням перед назвою IComparer<IFigure3D>
    {if (p1 == null || p2 == null)
        throw new ArgumentException("Окремі аргументи не є тривимірними фігурами");
        if (p1.H > p2.H) return 1;
        if (p1.H < p2.H) return -1;
        return 0;
    }
}

public class SortByVolume : IComparer<IFigure3D>
{public int Compare(IFigure3D p1, IFigure3D p2)
    {if (p1 == null || p2 == null)
        throw new ArgumentException("Окремі аргументи не є тривимірними фігурами");
        if (p1.volume() > p2.volume()) return 1;
        if (p1.volume() < p2.volume()) return -1;
        return 0;
    }
}

public class SortByPerimeterBase : IComparer<IFigure3D>
{public int Compare(IFigure3D p1, IFigure3D p2)
    {if (p1 == null || p2 == null)
        throw new ArgumentException("Окремі аргументи не є тривимірними фігурами");
        if (p1.perimeterBase() > p2.perimeterBase()) return 1;
        if (p1.perimeterBase() < p2.perimeterBase()) return -1;
        return 0;
    }
}

public class SortByAreaBase : IComparer<IFigure3D>
{public int Compare(IFigure3D p1, IFigure3D p2)
    {if (p1 == null || p2 == null)
        throw new ArgumentException("Окремі аргументи не є тривимірними фігурами");
        if (p1.areaBase() > p2.areaBase()) return 1;
        if (p1.areaBase() < p2.areaBase()) return -1;
        return 0;
    }
}

```

Тоді узагальнена процедура сортування буде, наприклад, такою:

```

private void SortCheckedChanged(object sender, EventArgs e)
{ if (radioButton1.Checked)
    ListDrawFigure3D.Sort();
  if (radioButton5.Checked)
    ListDrawFigure3D.Sort(new SortByH());
  if (radioButton2.Checked)
    ListDrawFigure3D.Sort(new SortByVolume());
  if (radioButton3.Checked)
    ListDrawFigure3D.Sort(new SortByAreaBase());
  if (radioButton4.Checked)
    ListDrawFigure3D.Sort(new SortByPerimeterBase());
  if (checkBoxDesc.Checked)
    ListDrawFigure3D.Reverse();
  Refresh();
}

```

Цю процедуру слід викликати при виникненні події *CheckedChanged* для всіх перемикачів та прапорця сортування за спаданням.