

Лабораторна робота № 25

Тема. Ітератори. Використання делегатів для сортування об'єктів.
Мета. Формування вмінь і навиків використання ітераторів для перебору та делегатів для сортування об'єктів. Закріплення вмінь і навиків наслідування звичайних та абстрактних класів засобами С#, використання об'єктів, підпрограм, елементів керування. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання.

1. Які об'єкти називаються делегатами? Як описати делегат?
2. Як визначити змінну делегата? Чим така змінна відрізняється від звичайної змінної?
3. Як викликати функцію-делегат?
4. Що таке ітератор? Чим виконання блоку ітератора відрізняється від виконання звичайного блоку коду?
5. Як описати і як використати ітератор?

Завдання.

1. Відкрийте розроблений раніше додаток-форму з описом базового та породжених класів згідно варіанту і методами *draw3D()* для їх зображення. Використайте делегати для сортування об'єктів за різними критеріями.

Наприклад, якщо допоміжні методи та класи для сортування геометричних фігур виглядають так:

```
public static int SortByH(IDrawFigure3D p1, IDrawFigure3D p2)
{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 int SortByVolume(IDrawFigure3D p1, IDrawFigure3D 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 static class SortByPerimeterBase
{public static int Compare(IDrawFigure3D p1, IDrawFigure3D 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<IDrawFigure3D>
{public int Compare(IDrawFigure3D p1, IDrawFigure3D 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; }}
```

то узагальнена процедура сортування об'єктів може бути такою:

```
private void SortCheckedChanged(object sender, EventArgs e)
{if (radioButton1.Checked)
    ListDrawFigure3D.Sort(); //сортування об'єктів по інтерфейсу IComparable
if (radioButton2.Checked)
    ListDrawFigure3D.Sort(SortByVolume); //делегат з методу екземпляра
if (radioButton3.Checked)
    ListDrawFigure3D.Sort((new SortByAreaBase()).Compare); //делегат з методу об'єкта
if (radioButton4.Checked)
    ListDrawFigure3D.Sort(SortByPerimeterBase.Compare); //делегат з статичного об'єкта
if (radioButton5.Checked)
    ListDrawFigure3D.Sort(SortByH); //делегат з статичного методу класу
if (checkBoxDesc.Checked)
    ListDrawFigure3D.Reverse();
Refresh();
}
```

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

2. Створіть та використайте у формі ітератор для опрацювання окремих об'єктів.

Наприклад, для збільшення розміру лише паралелепіпедів серед всіх геометричних фігур доцільно створити такий ітератор:

```
public IEnumerable<IDrawFigure3D> GetParallelepiped()  
{foreach (IDrawFigure3D F in ListDrawFigure3D)  
  if (F as DrawParallelepiped !=null)  
    yield return F; }
```

Тоді процедура обробки події натиснення кнопки з надписом *Збільшити висоту паралелепіпедів* буде, наприклад, такою:

```
private void button5_Click(object sender, EventArgs e)  
{foreach (IDrawFigure3D F in GetParallelepiped())  
  F.H++; }
```