

# **Отчёт по лабораторной работе № 3**

**Волгина А. Д., ИУ5-33Б**

Разработать программу, реализующую работу с коллекциями.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Создать объекты классов «Прямоугольник», «Квадрат», «Круг».
3. Для реализации возможности сортировки геометрических фигур для класса «Геометрическая фигура» добавить реализацию интерфейса `Comparable`. Сортировка производится по площади фигуры.
4. Создать коллекцию класса `ArrayList`. Сохранить объекты в коллекцию. Отсортировать коллекцию. Вывести в цикле содержимое коллекции.
5. Создать коллекцию класса `List<Figure>`. Сохранить объекты в коллекцию. Отсортировать коллекцию. Вывести в цикле содержимое коллекции.
6. Модифицировать класс разреженной матрицы (проект `SparseMatrix`) для работы с тремя измерениями –  $x, y, z$ . Вывод элементов в методе `ToString()` осуществлять в том виде, который Вы считаете наиболее удобным. Разработать пример использования разреженной матрицы для геометрических фигур.
7. Реализовать класс «`SimpleStack`» на основе односвязного списка. Класс `SimpleStack` наследуется от класса `SimpleList` (проект `SimpleListProject`). Необходимо добавить в класс методы:
  - `public void Push(T element)` – добавление в стек;
  - `public T Pop()` – чтение с удалением из стека.
8. Пример работы класса `SimpleStack` реализовать на основе геометрических фигур.



```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Figures
{
    interface IComparable
    {
        int CompareTo(object obj);
    }

    /// <summary>
    /// Класс фигура
    /// </summary>
    abstract class Figure : IComparable
    {
        /// <summary>
        /// Тип фигуры
        /// </summary>
        public string Type
        {
            get
            {
                return this._Type;
            }
        }
    }
}
```

```

        protected set
        {
            this._Type = value;
        }
    }

    string _Type;

    /// <summary>
    /// Вычисление площади
    /// </summary>

    public abstract double Area();

    /// <summary>
    /// Приведение к строке, переопределение метода Object
    /// </summary>

    public override string ToString()
    {
        return this.Type + " площадью " +
            this.Area().ToString();
    }

    public int CompareTo(object obj)
    {
        //Приведение параметра к типу "фигура"
        Figure p = (Figure)obj;

        //Сравнение
        if (this.Area() < p.Area()) return -1;
        else if (this.Area() == p.Area()) return 0;
        else return 1; //(this.Area() > p.Area())
    }

```

```

    }
}
interface IPrint
{
    void Print();
}
class Rectangle : Figure, IPrint
{
    /// <summary>
    /// Высота
    /// </summary>
    double height;
    /// <summary>
    /// Ширина
    /// </summary>
    double width;
    /// <summary>
    /// Основной конструктор
    /// </summary>
    /// <param name="ph">Высота</param>
    /// <param name="pw">Ширина</param>
    public Rectangle(double ph, double pw)
    {
        this.height = ph;
        this.width = pw;
        this.Type = "Прямоугольник";
    }
}

```

```

    }

    /// <summary>
    /// Вычисление площади
    /// </summary>

    public override double Area()
    {
        double Result = this.width * this.height;

        return Result;
    }

    public void Print()
    {
        Console.WriteLine(this.ToString());
    }
}

class Square : Rectangle, IPrint
{
    public Square(double size)
        : base(size, size)
    {
        this.Type = "Квадрат";
    }
}

class Circle : Figure, IPrint
{
    /// <summary>
    /// Ширина

```

```

/// </summary>

    double radius;

    /// <summary>

    /// Основной конструктор

    /// </summary>

    /// <param name="ph">Высота</param>
    /// <param name="pw">Ширина</param>

    public Circle(double pr)
    {
        this.radius = pr;
        this.Type = "Круг";
    }

    public override double Area()
    {
        double Result = Math.PI * this.radius * this.radius;
        return Result;
    }

    public void Print()
    {
        Console.WriteLine(this.ToString());
    }
}

class Program
{
    static void Main(string[] args)
    {

```

```
Rectangle rect = new Rectangle(5, 4);
Square square = new Square(5);
Circle circle = new Circle(5);
ArrayList al = new ArrayList();
al.Add(rect);
al.Add(square);
al.Add(circle);
Console.WriteLine("\nПеред сортировкой:");
foreach (var f in al)
{
    Console.WriteLine(f);
}
/*Comparer comparer = new IComparable();
al.Sort();
Console.WriteLine("\nПосле сортировки:");
foreach (var f in al)
{
    Console.WriteLine(f);
}*/
List<Figure> ls = new List<Figure>();
ls.Add(rect);
ls.Add(square);
ls.Add(circle);
Console.WriteLine("\nПеред сортировкой:");
foreach (var x in ls) Console.WriteLine(x);
//сортировка
```



```
/*      ls.Sort();

        Console.WriteLine("\nПосле сортировки:");

        foreach (var x in ls) Console.WriteLine(x);*/

    }

}

}
```



Консоль отладки Microsoft Visual Studio

Перед сортировкой:  
Прямоугольник площадью 20  
Квадрат площадью 25  
Круг площадью 78,53981633974483

Перед сортировкой:  
Прямоугольник площадью 20  
Квадрат площадью 25  
Круг площадью 78,53981633974483

C:\Users\Anna\Desktop\Учёба\Labs\C#\Lab3\Lab3\bin\Debug\netcoreapp3.1\Lab3.exe (процесс 17940) завершил работу с кодом 0  
Нажмите любую клавишу, чтобы закрыть это окно...