МГТУ имени Н.Э. Баумана
Факультет «Информатика и системы управления»

Базовые компоненты интернет технологий. Отчёт по лабораторной работе № 3.

Белкина Екатерина
Группа ИУ5-31Б

24 декабря 2018

Условие задания:

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 IComparable. 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 (разобранного в пособии). 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.

Текст программы

Program.cs:

```
using System;
    using System.Collections;
    using System.Collections.Generic;
    using Lab_3.SparseMatrix;
    using Lab_3.Stack;

    namespace Lab_3
    {
       class Program
       {
          static void Main(string[] args)
          {
             // ArrayList example
             Console.WriteLine("ArrayList:");
             var arrayList = new ArrayList
             {
```

```
    new Rectangle(5, 6.3),
    new Rectangle(6, 3.6),
    new Rectangle(9.1, 4.7),
    new Square(6.5),
    new Square(3.9),
    new Square(5.87),
    new Circle(2.3),
    new Circle(4.2),
    new Circle(6.543)
};

arrayList.Sort();

foreach (Figure figure in arrayList)
{
    figure.Print();
}

// Figure List example
Console.WriteLine("\nFigure List:");
var figureList = new List<Figure>
{
    new Rectangle(5, 6.3),
    new Rectangle(6, 3.6),
    new Rectangle(9.1, 4.7),
    new Square(6.5),
    new Square(3.9),
    new Square(5.87),
    new Circle(2.3),
    new Circle(4.2),
    new Circle(6.543)
};

figureList.Sort();

foreach (var figure in figureList)
{
    figure.Print();
}
```

```
            // Figure Matrix example
            Console.WriteLine("\nMatrix:");
            var matrix = new Matrix<Figure>(5, 5, 4, new FigureMatrixCheckEmpty());
            matrix[0, 0, 0] = figureList[0];
            matrix[0, 3, 2] = figureList[1];
            matrix[3, 3, 0] = figureList[2];
            matrix[2, 2, 2] = figureList[3];

            Console.WriteLine(matrix);

            // Simple Stack example
            Console.WriteLine("\nSimple Stack:");
            var figureStack = new SimpleStack<Figure>();
            foreach (var figure in figureList)
            {
                figureStack.Push(figure);
            }

            while (!figureStack.Empty())
            {
                Console.WriteLine(figureStack.Pop());
            }
        }
    }
}
```

Circle.cs:

```
using System;

namespace Lab_3
{
    public class Circle : Figure
    {
        public double Radius { get; set; }

        public Circle(double radius)
        {
            Radius = radius;
        }
```

```
        public override double Area()
        {
            return Math.PI * Radius * Radius;
        }

        public override string ToString()
        {
            return $"Круг с радиусом {Radius} имеет площадь {Area()}";
        }
    }
}
```

EmptyFigure.cs:

```
namespace Lab_3
{
    public class EmptyFigure : Figure
    {
        public override double Area()
        {
            return 0;
        }
    }
}
```

Figure.cs:

```
using System;

namespace Lab_3
{
    public abstract class Figure : IComparable, IPrint
    {
        public abstract double Area();

        public int CompareTo(object obj)
        {
            switch (obj)
            {
                case null:
                    return 1;
                case Figure figure:
```

```
                return Area().CompareTo(figure.Area());
            default:
                throw new ArgumentException("Object is not a Figure");
        }
    }

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

FigureMatrixCheckEmpty.cs:

```
using Lab_3.SparseMatrix;

namespace Lab_3
{
  public class FigureMatrixCheckEmpty : IMatrixCheckEmpty<Figure>
  {
    public Figure GetEmptyElement()
    {
      return null;
    }

    public bool CheckEmptyElement(Figure element)
    {
      return element == null;
    }
  }
}
```

IPrint.cs:

```
namespace Lab_3
{
  public interface IPrint
  {
    void Print();
  }
}
```

Rectangle.cs:

```
using System;

namespace Lab_3
{
   public class Rectangle : Figure
   {
      public double Height { get; set; }
      public double Width { get; set; }

      public Rectangle(double height, double width)
      {
         Height = height;
         Width = width;
      }

      public override double Area()
      {
         return Height * Width;
      }

      public override string ToString()
      {
            return $"Прямоугольник с высотой {Height} и шириной {Width} имеет площадь
{Area()}";
      }
   }
}
```

Square.cs:

```
using System;

namespace Lab_3
{
   public class Square : Rectangle
   {
      public Square(double side) : base(side, side) {}

      public override string ToString()
      {
```

```
            return $"Квадрат со стороной {Height} имеет площадь {Area()}";
        }
    }
}
```

IMatrixCheckEmpty.cs:

```
namespace Lab_3.SparseMatrix
{
    public interface IMatrixCheckEmpty<T>
    {
        T GetEmptyElement();

        bool CheckEmptyElement(T element);
    }
}
```

Matrix.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Text;

namespace Lab_3.SparseMatrix
{
    public class Matrix<T>
    {
        private Dictionary<string, T> _matrix = new Dictionary<string, T>();

        private int _maxX;
        private int _maxY;
        private int _maxZ;

        private IMatrixCheckEmpty<T> _checkEmpty;

                public Matrix(int t_maxX, int t_maxY, int t_maxZ, IMatrixCheckEmpty<T>
t_checkEmpty)
        {
            _maxX = t_maxX;
            _maxY = t_maxY;
            _maxZ = t_maxZ;
```

```
        _checkEmpty = t_checkEmpty;
    }

    public T this[int x, int y, int z]
    {
        get
        {
            CheckBounds(x, y, z);
            var key = DictKey(x, y, z);

            return _matrix.ContainsKey(key) ? _matrix[key] : _checkEmpty.GetEmptyElement();
        }
        set
        {
            CheckBounds(x, y, z);
            var key = DictKey(x, y, z);
            _matrix.Add(key, value);
        }
    }

    private void CheckBounds(int x, int y, int z)
    {
        if (x < 0 || x > _maxX)
        {
                throw new ArgumentOutOfRangeException(nameof(x), $"x = {x} выходит за
границу");
        }
        if (y < 0 || y > _maxX)
        {
                throw new ArgumentOutOfRangeException(nameof(y), $"y = {y} выходит за
границу");
        }
        if (z < 0 || z > _maxX)
        {
                 throw new ArgumentOutOfRangeException(nameof(z), $"z = {z} выходит за
границу");
        }
    }
```

```
private string DictKey(int x, int y, int z)
{
    return $"{x}_{y}_{z}";
}

public override string ToString()
{
    var builder = new StringBuilder();

    for (var k = 0; k < _maxZ; k++)
    {
        builder.Append("[\n");
        for (var j = 0; j < _maxY; j++)
        {
            builder.Append("\t");
            builder.Append("[");
            for (int i = 0; i < _maxX; i++)
            {
                if (i > 0)
                {
                    builder.Append("\t");
                }

                if (!_checkEmpty.CheckEmptyElement(this[i, j, k]))
                {
                    builder.Append(this[i, j, k]);
                }
                else
                {
                    builder.Append(" - ");
                }
            }

            builder.Append("]\n");
        }

        builder.Append("]\n");
    }

    return builder.ToString();
```

```
        }
    }
}
```

SimpleStack.cs:

```
using System;

namespace Lab_3.Stack
{
    public class SimpleStack<T>
    {
        public SimpleStackElement<T> Head { get; set; }

        public SimpleStack()
        {
            Head = null;
        }

        public void Push(T data)
        {
            var element = new SimpleStackElement<T>(data);

            element.Next = Head;
            Head = element;
        }

        public T Pop()
        {
            if (Head == null)
            {
                throw new Exception("Стек пуст");
            }

            var returnElement = Head;
            Head = Head.Next;

            return returnElement.Data;
        }

        public bool Empty()
```

```
        {
            return Head == null;
        }
    }
}
```

SimpleStackElement.cs:

```
namespace Lab_3.Stack
{
    public class SimpleStackElement<T>
    {
        public T Data { get; set; }
        public SimpleStackElement<T> Next { get; set; }

        public SimpleStackElement(T data)
        {
            Data = data;
        }
    }
}
```

Примеры выполнения программы

```
ArrayList:
Квадрат со стороной 3,9 имеет площадь 15,21
Круг с радиусом 2,3 имеет площадь 16,61902513749
Прямоугольник с высотой 6 и шириной 3,6 имеет площадь 21,6
Прямоугольник с высотой 5 и шириной 6,3 имеет площадь 31,5
Квадрат со стороной 5,87 имеет площадь 34,4569
Квадрат со стороной 6,5 имеет площадь 42,25
Прямоугольник с высотой 9,1 и шириной 4,7 имеет площадь 42,77
Круг с радиусом 4,2 имеет площадь 55,4176944093239
Круг с радиусом 6,543 имеет площадь 134,494248712342
```

```
Figure List:
Квадрат со стороной 3,9 имеет площадь 15,21
Круг с радиусом 2,3 имеет площадь 16,61902513749
Прямоугольник с высотой 6 и шириной 3,6 имеет площадь 21,6
Прямоугольник с высотой 5 и шириной 6,3 имеет площадь 31,5
Квадрат со стороной 5,87 имеет площадь 34,4569
Квадрат со стороной 6,5 имеет площадь 42,25
Прямоугольник с высотой 9,1 и шириной 4,7 имеет площадь 42,77
Круг с радиусом 4,2 имеет площадь 55,4176944093239
Круг с радиусом 6,543 имеет площадь 134,494248712342
```

```
Matrix:
[
        [Квадрат со стороной 3,9 имеет площадь 15,21     -       -       -       - ]
        [ -      -       -       -       - ]
        [ -      -       -       -       - ]
        [ -      -       -      Прямоугольник с высотой 6 и шириной 3,6 имеет площадь 21,6       - ]
        [ -      -       -       -       - ]
]
[
        [ -      -       -       -       - ]
        [ -      -       -       -       - ]
        [ -      -       -       -       - ]
        [ -      -       -       -       - ]
        [ -      -       -       -       - ]
]
[
        [ -      -       -       -       - ]
        [ -      -       -       -       - ]
        [ -      -      Прямоугольник с высотой 5 и шириной 6,3 имеет площадь 31,5       -       - ]
        [Круг с радиусом 2,3 имеет площадь 16,61902513749        -       -       -       - ]
        [ -      -       -       -       - ]
]
[
        [ -      -       -       -       - ]
        [ -      -       -       -       - ]
        [ -      -       -       -       - ]
        [ -      -       -       -       - ]
        [ -      -       -       -       - ]
]
```

```
Simple Stack:
Круг с радиусом 6,543 имеет площадь 134,494248712342
Круг с радиусом 4,2 имеет площадь 55,4176944093239
Прямоугольник с высотой 9,1 и шириной 4,7 имеет площадь 42,77
Квадрат со стороной 6,5 имеет площадь 42,25
Квадрат со стороной 5,87 имеет площадь 34,4569
Прямоугольник с высотой 5 и шириной 6,3 имеет площадь 31,5
Прямоугольник с высотой 6 и шириной 3,6 имеет площадь 21,6
Круг с радиусом 2,3 имеет площадь 16,61902513749
Квадрат со стороной 3,9 имеет площадь 15,21
```