

МГТУ имени Н. Э. Баумана

Базовые компоненты интернет технологий

Отчет по лабораторной работе №3

Мелисов Арсен Ермекович

Группа ИУ5-31Б

23 декабря 2018 г.

Москва
2018

Задание

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` (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

Код

Program.cs

```
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using Lab_3.SparseMatrix;
5  using Lab_3.Stack;
6
7  namespace Lab_3
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             // ArrayList example
14             Console.WriteLine("ArrayList:");
15             var arrayList = new ArrayList
16             {
17                 new Rectangle(5, 6.3),
18                 new Rectangle(6, 3.6),
19                 new Rectangle(9.1, 4.7),
20                 new Square(6.5),
21                 new Square(3.9),
22                 new Square(5.87),
23                 new Circle(2.3),
24                 new Circle(4.2),
25                 new Circle(6.543)
26             };
27
28             arrayList.Sort();
29
30             foreach (Figure figure in arrayList)
31             {
32                 figure.Print();
33             }
34
35             // Figure List example
36             Console.WriteLine("\nFigure List:");
37             var figureList = new List<Figure>
38             {
39                 new Rectangle(5, 6.3),
40                 new Rectangle(6, 3.6),
41                 new Rectangle(9.1, 4.7),
42                 new Square(6.5),
43                 new Square(3.9),
44                 new Square(5.87),
45                 new Circle(2.3),
46                 new Circle(4.2),
47                 new Circle(6.543)
48             };
```

```

49
50     figureList.Sort();
51
52     foreach (var figure in figureList)
53     {
54         figure.Print();
55     }
56
57     // Figure Matrix example
58     Console.WriteLine("\nMatrix:");
59     var matrix = new Matrix<Figure>(5, 5, 4, new FigureMatrixCheckEmpty());
60     matrix[0, 0, 0] = figureList[0];
61     matrix[0, 3, 2] = figureList[1];
62     matrix[3, 3, 0] = figureList[2];
63     matrix[2, 2, 2] = figureList[3];
64
65     Console.WriteLine(matrix);
66
67     // Simple Stack example
68     Console.WriteLine("\nSimple Stack:");
69     var figureStack = new SimpleStack<Figure>(); foreach
70     (var figure in figureList) {
71
72         figureStack.Push(figure);
73     }
74
75     while (!figureStack.Empty())
76     {
77         Console.WriteLine(figureStack.Pop());
78     }
79 }
80 }
81 }

```

SparseMatrix/IMatrixCheckEmpty.cs

```
1 namespace Lab_3.SparseMatrix
2 {
3     public interface IMatrixCheckEmpty<T>
4     {
5         T GetEmptyElement();
6
7         bool CheckEmptyElement(T element);
8     }
9 }
```

SparseMatrix/Matrix.cs

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel.DataAnnotations;
4  using System.Text;
5  // ReSharper disable HeapView.BoxingAllocation
6
7  namespace Lab_3.SparseMatrix
8  {
9      public class Matrix<T>
10     {
11         private Dictionary<string, T> _matrix = new Dictionary<string, T>();
12
13         private int _maxX;
14         private int _maxY;
15         private int _maxZ;
16
17         private IMatrixCheckEmpty<T> _checkEmpty;
18
19         public Matrix(int t_maxX, int t_maxY, int t_maxZ, IMatrixCheckEmpty<T> t_checkEmpty)
20         {
21             _maxX = t_maxX;
22             _maxY = t_maxY;
23             _maxZ = t_maxZ;
24
25             _checkEmpty = t_checkEmpty;
26         }
27
28         public T this[int x, int y, int z]
29         {
30             get
31             {
32                 CheckBounds(x, y, z);
33                 var key = DictKey(x, y, z);
34
35                 return _matrix.ContainsKey(key) ? _matrix[key] :
36                     !_checkEmpty.GetEmptyElement();
37             }
38             set
39             {
40                 CheckBounds(x, y, z);
41                 var key = DictKey(x, y, z);
42                 _matrix.Add(key, value);
43             }
44         }
45
46         private void CheckBounds(int x, int y, int z)
47         {
48             if (x < 0 || x > _maxX)
```

```

49         throw new ArgumentOutOfRangeException(nameof(x),
50             ,!    границу");
51     }
52     if (y < 0 || y > _maxX)
53     {
54         throw new ArgumentOutOfRangeException(nameof(y),
55             ,!    границу");
56     }
57     if (z < 0 || z > _maxX)
58     {
59         throw new ArgumentOutOfRangeException(nameof(z),
60             ,!    границу");
61     }
62 }
63
64 private string DictKey(int x, int y, int z)
65 {
66     return $"{{x}}_{{y}}_{{z}}";
67 }
68
69 public override string ToString()
70 {
71     var builder = new StringBuilder();
72
73     for (var k = 0; k < _maxZ; k++)
74     {
75         builder.Append("\n");
76         for (var j = 0; j < _maxY; j++)
77         {
78             builder.Append("\t");
79             builder.Append("[");
80             for (int i = 0; i < _maxX; i++)
81             {
82                 if (i > 0)
83                 {
84                     builder.Append("\t");
85                 }
86
87                 if (!_checkEmpty.CheckEmptyElement(this[i, j, k]))
88                 {
89                     builder.Append(this[i, j, k]);
90                 }
91                 else
92                 {
93                     builder.Append(" - ");
94                 }
95             }
96
97             builder.Append("]\n");
98         }
99     }

```

\$"x = {x} выходит за

\$"y = {y} выходит за

\$"z = {z} выходит за

```
97         builder.Append("]\n");
98     }
99
100     return builder.ToString();
101 }
102 }
103 }
```


Stack/SimpleStackElement.cs

```
1 namespace Lab_3.Stack
2 {
3     public class SimpleStackElement<T>
4     {
5         public T Data { get; set; }
6         public SimpleStackElement<T> Next { get; set; }
7
8         public SimpleStackElement(T data)
9         {
10             Data = data;
11         }
12     }
13 }
```

Stack/SimpleStack.cs

```
1  using System;
2
3  namespace Lab_3.Stack
4  {
5      public class SimpleStack<T>
6      {
7          public SimpleStackElement<T> Head { get; set; }
8
9          public SimpleStack()
10         {
11             Head = null;
12         }
13
14         public void Push(T data)
15         {
16             var element = new SimpleStackElement<T>(data);
17
18             element.Next = Head;
19             Head = element;
20         }
21
22         public T Pop()
23         {
24             if (Head == null)
25             {
26                 throw new Exception("Стек пуст");
27             }
28
29             var returnElement = Head;
30             Head = Head.Next;
31
32             return returnElement.Data;
33         }
34
35         public bool Empty()
36         {
37             return Head == null;
38         }
39     }
40 }
```

Тесты

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

Рис. 1: Проверка Array List

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

Рис. 2: Проверка Figure List

```

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    -    -    -    - ]
    [ -    -    -    -    - ]
]
[
    [ -    -    -    -    - ]
    [ -    -    -    -    - ]
    [ -    -    -    -    - ]
    [ -    -    -    -    - ]
    [ -    -    -    -    - ]
]
]

```

Рис. 3: Проверка Matrix

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

```

Рис. 4: Проверка Simple Stack