

Министерство науки и высшего образования Российской Федерации
Рязанский государственный радиотехнический университет
имени В.Ф. Уткина

Кафедра «Электронные вычислительные машины»

Отчёт
о лабораторной работе №1
«Создание простейшего консольного приложения на языке C#»
по дисциплине:
«Визуальное программирование»

Выполнили:
Студенты группы 140
Бригады №1
Волкова Е.В.
Егорова Е.М.
Проверил:
ст. преп. Бастрычкин А.С.
ст. преп. Хизриева Н. И.

Рязань 2024

Цель работы: получить навыки написания простейших приложений на языке C#.

Задание:

Задача 1. Введите с клавиатуры строку произвольной длины и подсчитайте процент вхождения заданного символа в строку.

Задача 2. Задан массив действительных чисел размерности 10×10 . Найти сумму элементов каждой строки, произведение элементов каждого столбца, и максимальный элемент главной диагонали (подсказка: все элементы, для которых номер строки совпадают с номером столбца).

Практическая часть.

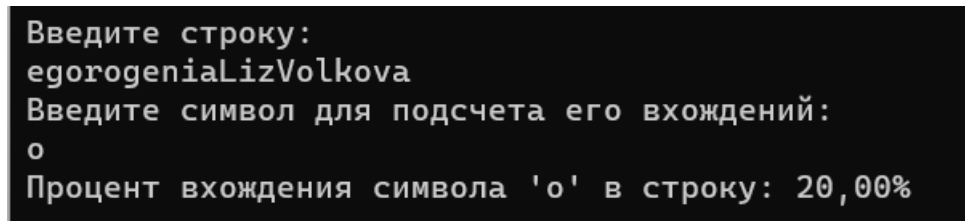
1. Код программы для задачи 1 с описанием библиотеки классов:

```
namespace VP_Lab1
{
    public class Task1
    {
        public static double calculatePercentage(string inputString, char
targetChar)
        {
            int charCount = 0; foreach (char c in inputString)
            {
                if (c == targetChar)
                {
                    charCount++;
                }
            }
            double percentage = (double)charCount / inputString.Length * 100;
            return percentage;
        }
    }
}
```

Код описания интерфейса:

```
namespace VP_Lab1
{
    class Task1UI
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Введите строку:");
            string InputString = Console.ReadLine();
            Console.WriteLine("Введите символ для подсчета его вхождений:"); char
targetChar = Console.ReadKey().KeyChar;
            Console.WriteLine();
            double p = Task1. calculatePercentage (InputString, targetChar);
            Console.WriteLine($"Процент вхождения символа '{targetChar}' в строку: {p:F2}%");
        }
    }
}
```

Результат работы программы при выполнении задачи1 представлен на рисунке 1.



```
Введите строку:
egorogeniaLizVolkova
Введите символ для подсчета его вхождений:
o
Процент вхождения символа 'o' в строку: 20,00%
```

Рисунок 1 – Результат работы программы для задачи 1

2. Код программы для задачи 2 с описанием библиотеки классов:

```
namespace VP_lab1._2
{
    public class Task2
    {
        public static double[] calculate_row_sum(double[,] Matrix)
```

```

    {
        double[] rowSums = new double[Matrix.GetLength(0)];

        for (int i = 0; i < Matrix.GetLength(0); i++)
        {
            double sum = 0;
            for (int j = 0; j < Matrix.GetLength(1); j++)
            {
                sum += Matrix[i, j];
            }
            rowSums[i] = sum;
        }

        return rowSums;
    }

    public static double[] calculate_column_products(double[,] matrix)
    {
        double[] colProducts = new double[matrix.GetLength(1)];

        for (int j = 0; j < matrix.GetLength(1); j++)
        {
            double product = 1;
            for (int i = 0; i < matrix.GetLength(0); i++)
            {
                product *= matrix[i, j];
            }
            colProducts[j] = product;
        }

        return colProducts;
    }

    public static double find_maxDiagonal_element(double[,] matrix)
    {
        double maxDiagonalElement = int.MinValue;
        for (int i = 0; i < matrix.GetLength(0); i++)
        {
            if (matrix[i, i] > maxDiagonalElement)
            {
                maxDiagonalElement = matrix[i, i];
            }
        }
        return maxDiagonalElement;
    }
}
}

```

Код описания интерфейса:

```

namespace VP_lab1._2
{
    using System;
    class Task2UI
    {
        static void Main(string[] args)
        {
            double[,] matrix = new double[10, 10];
            Random rand = new Random();
            for (int i = 0; i < 10; i++)
            {
                for (int j = 0; j < 10; j++)
                {
                    matrix[i, j] = rand.NextDouble()*10;
                }
            }
        }
    }
}

```

```

    }
}
Console.WriteLine("Сгенерированная матрица:");
for (int i = 0; i < matrix.GetLength(0); i++)
{
    for (int j = 0; j < matrix.GetLength(1); j++)
    {
        Console.Write("{0: 0.00}", matrix[i, j]);
    }
    Console.WriteLine();
}

double[] rowSum = Task2.calculate_row_sum(matrix);
double[] colProduct = Task2.calculate_column_products(matrix);
double maxDiagonal = Task2.find_maxDiagonal_element(matrix);

Console.WriteLine("Суммы элементов каждой строки:");
foreach (double sum in rowSum)
{
    Console.WriteLine("{0: 0.00}", sum);
}

Console.WriteLine("\nПроизведения элементов каждого столбца:");
foreach (double product in colProduct)
{
    Console.WriteLine("{0: 0.00}", product);
}

Console.WriteLine("\nМаксимальный элемент главной диагонали:");
Console.WriteLine("{0: 0.00}", maxDiagonal);
}
}
}

```

Результат работы программы при выполнении задачи 2 представлен на рисунке 2.

```

Сгенерированная матрица:
0,21 6,70 2,58 1,44 8,50 5,23 2,16 2,66 5,07 8,89
1,89 0,29 0,48 1,71 7,96 7,87 2,46 5,52 4,52 9,23
7,48 4,12 2,80 3,08 1,02 6,84 1,94 3,96 4,37 8,97
2,49 9,01 4,07 0,29 5,98 7,37 4,43 5,97 7,24 5,31
9,77 1,36 1,75 9,02 4,04 1,21 0,26 8,96 2,44 2,22
7,69 1,86 4,52 6,36 9,06 2,68 2,08 3,39 1,40 9,39
5,23 0,41 5,65 5,37 6,63 6,02 5,82 4,85 0,07 9,75
5,87 2,65 0,71 0,10 9,51 3,65 1,33 1,42 7,13 6,67
4,60 9,47 7,59 9,22 1,39 9,59 9,66 5,79 0,35 8,81
2,60 5,99 1,55 8,57 3,50 1,05 3,92 4,62 1,69 2,45
Суммы элементов каждой строки:
43,44
41,93
44,58
52,16
41,02
48,44
49,81
39,05
66,47
35,93

Произведения элементов каждого столбца:
209133,81
11111,14
5290,41
5246,75
4631979,16
1483613,17
7343,83
1954671,62
721,66
114148259,48

Максимальный элемент главной диагонали:
5,82

```

Рисунок 2 – Результат работы программы для задачи 2