



**«Московский государственный технический университет
имени Н.Э. Баумана»
(национальный исследовательский университет)
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)


Отчет

по лабораторной работе № 4

Название лабораторной работы: Обработка массивов, строк и подпрограммы.

Дисциплина: Алгоритмизация и программирование

Студент гр. ИУ6-24Б

 **20.04.2024**

(Подпись, дата)

А.С. Воеводин

(И.О. Фамилия)

Преподаватель

 **20.04.2024**

(Подпись, дата)

О.А. Веселовская

(И.О. Фамилия)

Цель работы

Изучение и отработка знаний по работе с массивами, строками и подпрограммами в объектно-ориентированном языке C#, а также закрепление навыков работы в среде разработки Rider для создания консольных приложений.

Задание

Часть 1. Массивы.

Разработать программу решения системы 2-х линейных неоднородных уравнений с двумя неизвестными методом Крамера (через определители). Коэффициенты уравнений вводить с клавиатуры построчно, результаты вывести на экран. Предусмотреть соответствующее сообщение, если $= 0$.

Часть 2. Строки.

Дана последовательность строк. Каждая строка состоит из слов, разделенных пробелами. Написать программу, обеспечивающую ввод строк и их корректировку. Корректировка заключается в следующем: если среди слов строки обнаруживаются слова, заключенные в символ #, заменить их словами, введенными с клавиатуры. Вывести на экран исходную и скорректированную последовательности строк.

Ход работы:

- Написание кода для первой части
- Отладка программы
- Написание схемы алгоритма
- Повторение предыдущих пунктов для второй части задания
- Вывод

Сперва напишем код для первой части задания, которая решает неоднородную систему линейных уравнений для двух переменных. Для решения системы линейных уравнений применим алгоритм Крамера. Так как количество неизвестных равно 2, вычисление определителя системы не составит труда. Так же предусмотрены случаи, когда пользователь вводит нечисловые данные или вводит не 3 числа, также рассмотрен случай, когда система решений не имеет или

имеет их бесконечное количество:

Листинг 1

```
namespace laba_4_1;

class Program
{
    static double Determinant(int m1, int m2, double[,] arr)
    {
        return (arr[0, m1] * arr[1, m2] - arr[1, m1] * arr[0, m2]);
    }

    static void Main(string[] args)
    {
        double[,] kramer = new double[2, 3];
        Console.Write("Введите коэффициенты а и b и c через пробел: ");
        string input = Console.ReadLine();
        string[] parts = input.Split(' ');
        while (!(parts.Length == 3 && double.TryParse(parts[0], out kramer[0, 0]) &&
            double.TryParse(parts[1], out kramer[0, 1]) &&
            double.TryParse(parts[2], out kramer[0, 2])))
        {
            Console.Write("Некорректный ввод. Попробуйте еще раз: ");
            input = Console.ReadLine();
            parts = input.Split(' ');
        }

        Console.Write("Введите коэффициенты c и d через пробел: ");
        input = Console.ReadLine();
        parts = input.Split(' ');
        while (!(parts.Length == 3 && double.TryParse(parts[0], out kramer[1, 0]) &&
            double.TryParse(parts[1], out kramer[1, 1]) &&
            double.TryParse(parts[2], out kramer[1, 2])))
        {
            Console.Write("Некорректный ввод. Попробуйте еще раз: ");
            input = Console.ReadLine();
            parts = input.Split(' ');
        }

        if (Determinant(0, 1, kramer) == 0)
        {
            Console.WriteLine("Решений нет или их бесконечное количество.");
        }
        else
        {
            Console.WriteLine($"x = {Determinant(2, 1, kramer) / Determinant(0, 1, kramer)}");
            Console.WriteLine($"y = {Determinant(0, 2, kramer) / Determinant(0, 1, kramer)}");
        }
    }
}
```

```
}  
}  
}
```

Теперь введём несколько тестовых данных:

```
Введите коэффициенты a и b и c через пробел: 1 2 3  
Введите коэффициенты c и d через пробел: 3 4 5  
x = -1  
y = 2
```

Рисунок 1 – Тестовые данные 1

```
Введите коэффициенты a и b и c через пробел: 123 123 12  
Введите коэффициенты c и d через пробел: 1234 1234 ва  
Некорректный ввод. Попробуйте еще раз: 2134 124 5  
x = -0,003531124863487441  
y = 0,10109210047324353
```

Рисунок 2 – Тестовые данные 2

```
Введите коэффициенты a и b и c через пробел: 0 0 0  
Введите коэффициенты c и d через пробел: 0 0 0  
Решений нет или их бесконечное количество.
```

Рисунок 3 – Тестовые данные 3

```
Введите коэффициенты a и b и c через пробел: 1 2 3  
Введите коэффициенты c и d через пробел: 1 2 4  
Решений нет или их бесконечное количество.
```

Рисунок 4 – Тестовые данные 4

Легко проверить, что программа работает верно. Теперь сделаем схему алгоритма этой программы:

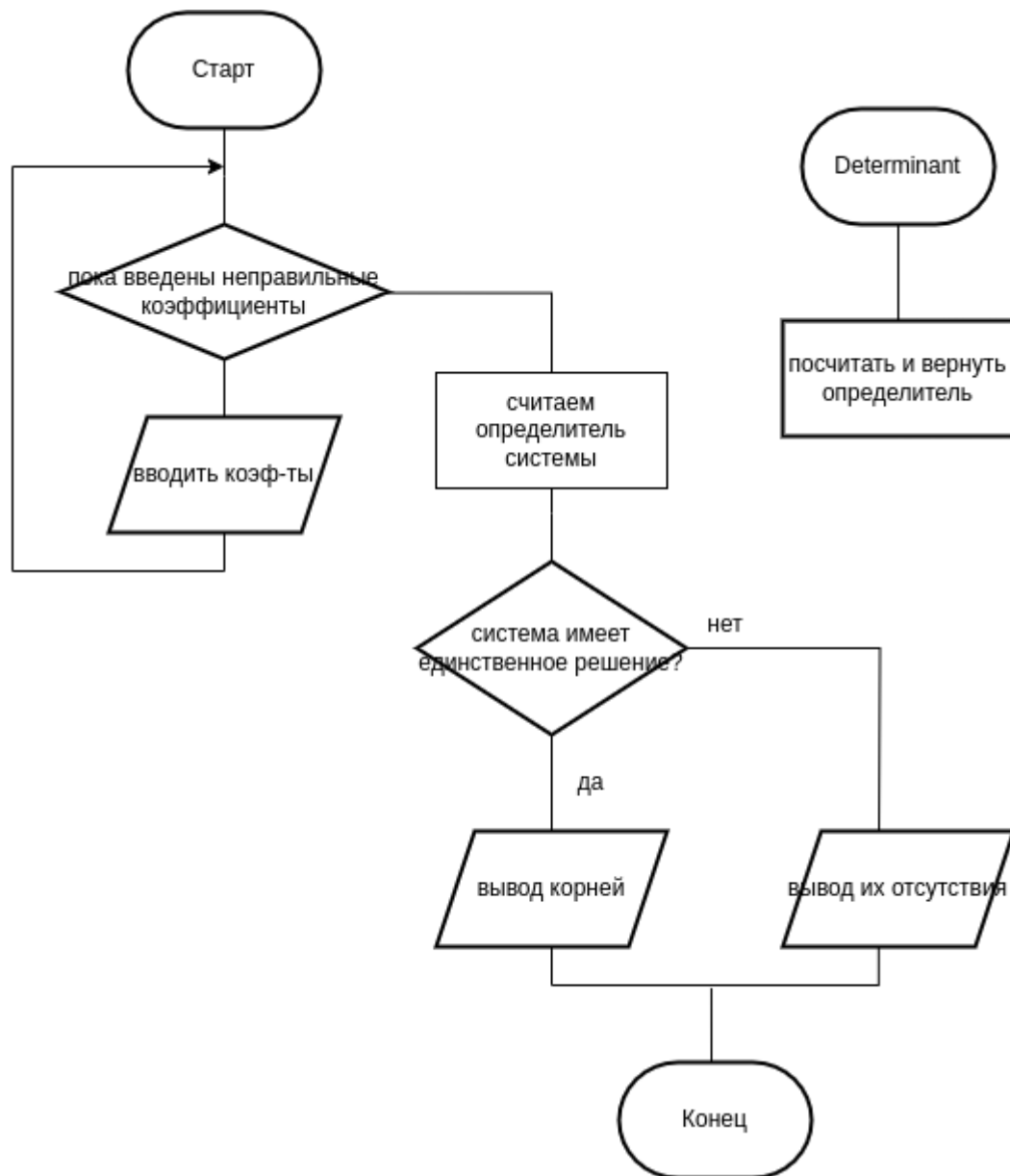


Рисунок 5 – Схема алгоритма основной программы и функции Determinant

Теперь приступим ко второй части задания и напомним для неё программу. Как и в первой части, в программе предусмотрен диалог с пользователем и обработка его вводимых данных. Также логика устроена таким образом, что слово может быть заключено в любое количество решёток с обеих сторон, так как алгоритм выполнен с помощью регулярных выражений. Ниже представлен код программы:

Листинг 2

```

using System.Text;
using System.Text.RegularExpressions;

namespace laba_4_2;

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

```

    {
        StringBuilder st;
        bool isBreak = false;
        Regex reg = new Regex(@"#+((\w)+)#+",
RegexOptions.IgnoreCase);
        while (!isBreak)
        {
            Console.WriteLine(
                "Введите строку, состоящую из слов, разделенных
пробелами или пустую строку для завершения: ");
            st = new StringBuilder(Console.ReadLine());
            foreach (Match match in reg.Matches(st.ToString()))
            {
                Console.Write($"Введите замену для
'{match.Groups[1].Value}': ");
                string newWord = Console.ReadLine();
                st.Replace(match.Value, newWord);
            }

            Console.WriteLine($"Получившаяся строка: {st}");
            Console.WriteLine("Хотите ли вы продолжить? да/нет");
            string response = Console.ReadLine();
            while (response != "да" && response != "нет")
            {
                Console.WriteLine("Введите еще раз. Хотите ли вы
продолжить? да/нет");
                response = Console.ReadLine();
            }

            if (response == "нет")
            {
                isBreak = true;
            }

            Console.Clear();
        }
    }
}

```

Проведём тестирование получившейся программы:

```

Введите строку, состоящую из слов, разделенных пробелами или пустую строку для завершения:
привет сегодня такой ###хороший# и #прекрасный## #ночь#
Введите замену для 'хороший': плохая
Введите замену для 'прекрасный': ужасная
Введите замену для 'ночь': день
Получившаяся строка: привет сегодня такой плохая и ужасная день
Хотите ли вы продолжить? да/нет
нет

```

Рисунок 6 – Тестовые данные 1

```

Введите строку, состоящую из слов, разделенных пробелами или пустую строку для завершения:
asdfdas dsafasdf dsaf #####sfadsfsadsadf adsfadsfsadf ##### asdfdasf ## asdasfsdf dsafsadf dsaf### sfsdaf ## asdfdsaf #
Получившаяся строка: asdfdas dsafasdf dsaf #####sfadsfsadsadf adsfadsfsadf ##### asdfdasf ## asdasfsdf dsafsadf dsaf### sfsdaf ## asdfdsaf #
Хотите ли вы продолжить? да/нет
да

```

Рисунок 7 – Тестовые данные 2

```
Введите строку, состоящую из слов, разделенных пробелами или пустую строку для завершения:  
фыва ыфва  
Получившаяся строка: фыва ыфва  
Хотите ли вы продолжить? да/нет  
ыфва  
Введите еще раз. Хотите ли вы продолжить? да/нет  
ыфва  
Введите еще раз. Хотите ли вы продолжить? да/нет  
фыва  
Введите еще раз. Хотите ли вы продолжить? да/нет
```

Рисунок 8 – Тестовые данные 3

Из тестов видно, что программа работает корректно. Теперь изобразим схему алгоритма этой программы:

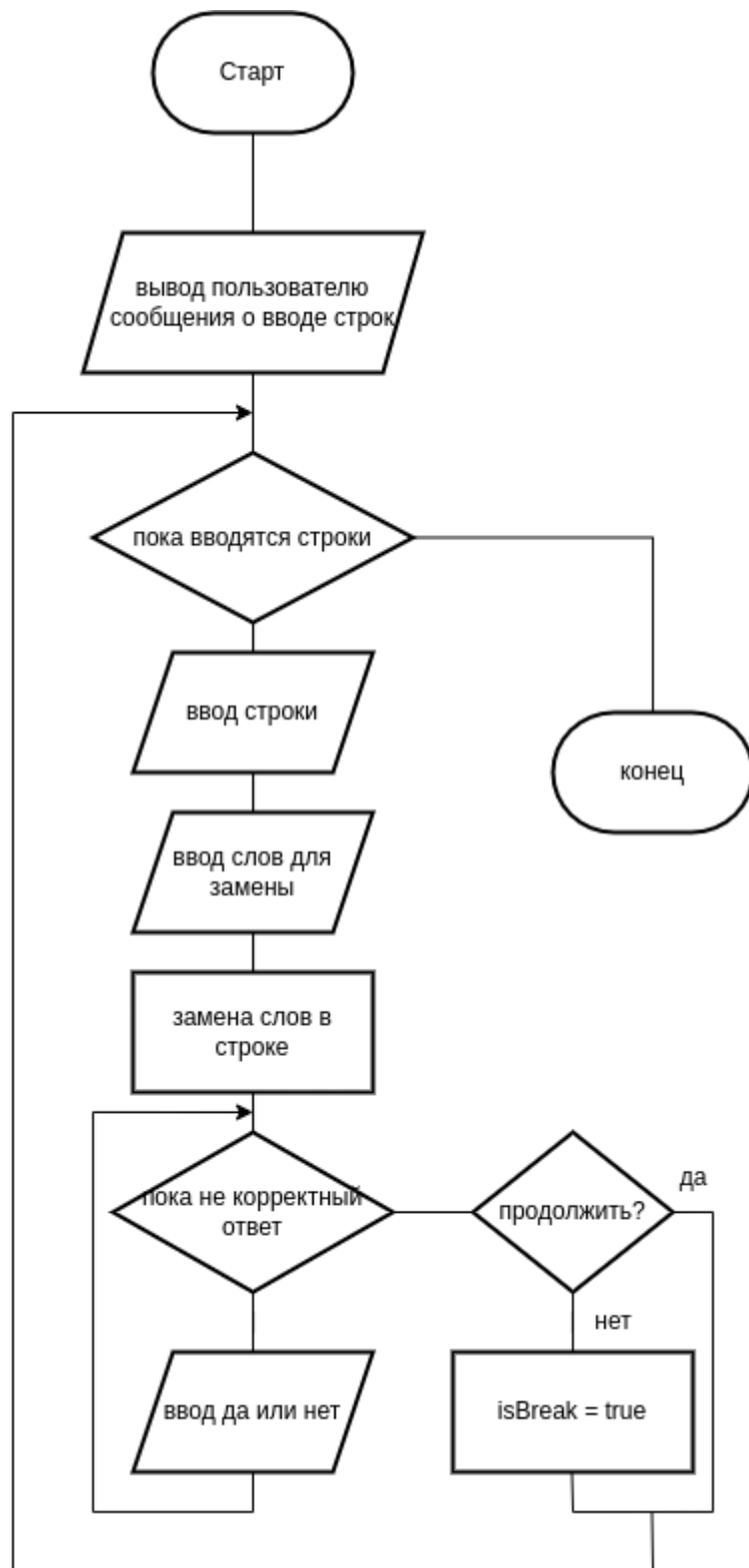


Рисунок 9 – Схема алгоритма программы

Вывод

В ходе лабораторной работы были получены навыки работы с массивами и их обработкой, включая ввод и вывод данных, со строками, их обработкой и выводом их на консоль, а также с написанием подпрограмм на языке C#.