

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
КАФЕДРА САПР

ОТЧЕТ

по лабораторной работе №5

по дисциплине «Объектно-ориентированное программирование»

Тема: Создание и использование массивов

Студентка гр. 9301

Синицкая В. А.

Преподаватель

Новакова Н.Е.

Санкт-Петербург

2021

Оглавление

1	Цель работы.....	3
2	Анализ задачи.....	3
3	Формальная постановка задачи.....	3
3.1	Исходные данные	3
3.2	Результирующие (выходные) данные	3
4	Используемые классы и методы.....	4
5	Текст программы.....	5
5.1	Текст программы 1	5
5.2	Текст программы 2	8
6	Контрольный пример.....	9
6.1	Контрольный пример для 1 программы.....	9
6.2	Контрольный пример для 2 программы.....	10
7	Полученные результаты.....	11
8	Выводы.....	11
9	Список использованных источников.....	11

1 Цель работы

Получение навыков работы с чтением из файла и с консоли, практика использования одномерных и двумерных массивов. Работа с командной строкой: вызов программы и передача аргументов, работа с методами, их вызовом и спецификаторами передаваемых параметров.

2 Анализ задачи

1. Программа:

Написать программу, запускаемую из командной строки, в качестве аргумента передать имя текстового файла, выполнить анализ текста: подсчитать количество символов, гласных и согласных букв, количество строк.

2. Программа:

Написать программу, считывающую с консоли две матрицы размера 2x2, сохранить результат произведения первой на вторую и вывести пользователю в консоль.

3 Формальная постановка задачи

3.1 Исходные данные

1. Программа:

Имя текстового файла, как параметр вызова программы из командной строки, текст, содержащийся внутри него.

2. Программа:

2 пары по 4 целых числа, записанных в виде матриц.

3.2 Результирующие (выходные) данные

1. Программа:

Текст файла, количество символов в текстовом файле, количество гласных и согласных букв, количество строк, либо сообщение об ошибке и о местоположении ее возникновения.

2. Программа:

2 пары по 4 целых числа, записанных в виде матриц.

4 Используемые классы и методы

В программах используются методы `Console.WriteLine()` и `Console.ReadLine()` — методы класса `System.Console`. `System.Console` — это класс для работы с консольным окном, определенный разработчиками стандартных библиотек для языка C#.

Метод `Console.ReadLine()` сохраняет введенную пользователем строку в заданную переменную.

Метод `Console.WriteLine()` служит для отображения в консольном окне пользователя строк и других данных, переданных в данный метод в качестве параметров.

1. Программа:

Содержит единственный класс `FileDetails`, в котором два метода: `Summarize` и `Main`. Метод `Main` в качестве параметра принимает массив строк `args`, в `args[0]` содержится название текстового файла, если он не был передан, вызывается исключение. Исключение так же вызывается, если файл не был найден, либо пуст. С помощью метода `FileStream` создается поток `stream` для чтения из указанного файла. Затем, с помощью метода `StreamReader` создается переменная для чтения, связанная с указанным ранее потоком. Используя `stream.Length`, записывается длина файла, которая совпадает с искомым количеством символов. Создается массив символов того же размера, в него копируется весь текст посимвольно, затем этот массив в качестве параметра вместе с параметрами типа `out` передается в метод `Summarize`, который, используя цикл `foreach` по символам массива и метод `IndexOf`, подсчитывает количество строк, гласных и согласных букв. Далее метод `Main` выводит необходимую информацию в консоль.

2. Программа:

Содержит единственный класс MatrixMultiply, в котором один метод — Main. В этом методе создается три массива, два из них считываются с консоли, в третий записывается результат перемножения, и далее выводится в консоль, используя два цикла for, один из которых вложен в другой, во внутреннем используется метод Console.Write(), выводящий переменную без перехода на новую строку, во внешнем цикле используется уже WriteLine, так что итоговый вывод представляется пользователю в виде матрицы. В методе используется метод string.Split(), по заданному разделителю возвращающий массив строк из одной строки, чтобы корректно считать введенные пользователем матрицы.

5 Текст программы

5.1 Текст программы 1

```
1 using System;
2 using System.IO; // FileStream, FileReader
3
4 namespace OOP5LabaFileDetails
5 {
6     class FileDetails
7     {
8
9         static void Summarize(char[] char_array, out long vowels, out long consonen
10         ts, out long line_breaks)
11         {
12             vowels = consonents = line_breaks = 0;
13             foreach (char symbol in char_array)
14             {
15                 if ("AEIOUaeiou".IndexOf(symbol) != -1)
16                     // symbol is a vowel
17                     vowels++;
18
19                 else if ("BCDFGHJKLMNPQRSTVWXYZbcd fghjklmnpqrstvwxyz".IndexOf(symbol) != -
20                 1)
21                     // symbol is a consonent
22                     consonents++;
23
24                 else if (symbol == '\n') // symbol is a line break
25                     line_breaks++;
26             }
27         }
28     }
29 }
```

```

21         }
22     }
23     static void Main(string[] args)
24     {
25         //Console.WriteLine(args.Length);
26         //foreach (string string_arg in args)
27             // Console.WriteLine(string_arg);
28         try
29         {
30             try
31             {
32                 if (args.Length < 1)
33
34                     throw new IndexOutOfRangeException("start argument is missing");
35
36                     string fileName = args[0]; // we sholud start program from cmd using fileNa
me like parameter
37
38                     FileStream stream = new FileStream(fileName, FileMode.Open);
39
40                     StreamReader reader = new StreamReader(stream);
41
42                     long length_of_file = stream.Length; // length of fileName file
43
44                     if (!File.Exists(fileName)) // file doesn't exist
45
46                         throw new FileNotFoundException("file doesn't exist");
47
48                         if (length_of_file <= 0)
49
50                             throw new ArgumentOutOfRangeException("file is empty");
51
52                             char[] contents = new char[length_of_file]; // char massive, copy of fileNa
me file
53
54                             //Console.WriteLine("length = {0}", contents.Length);
55
56                             for (int i = 0; i < length_of_file; i++)
57                                 contents[i] = (char)(reader.Read());
58
59                             foreach (char symbol in contents)
60                                 Console.Write(symbol);
61
62                             reader.Close();
63
64                             long vowels_amount, consonents_amount, line_breaks_amount;

```

```

50     Summarize(contents, out vowels_amount, out consonents_amount, out line_brea
ks_amount);
51
52     Console.WriteLine("\nAmount of symbols in the file: {0}", contents.Length);
53
54     Console.WriteLine("Amount of vowels: {0}", vowels_amount);
55
56     Console.WriteLine("Amount of consonents: {0}", consonents_amount);
57
58     Console.WriteLine("Amount of lines: {0}", line_breaks_amount + 1); // last
line doesn't have a line break
59
60     }
61     catch (IndexOutOfRangeException startArgument)
62     {
63
64         Console.WriteLine("Error: {0}", startArgument.Message);
65
66         Console.WriteLine("Stack: {0}", startArgument.StackTrace);
67     }
68     catch (ArgumentOutOfRangeException argument)
69     {
70
71         Console.WriteLine("Error: {0}", argument.Message);
72
73         Console.WriteLine("Stack: {0}", argument.StackTrace);
74     }
75     catch (FileNotFoundException noFile)
76     {
77
78         Console.WriteLine("Error: {0}", noFile.Message);
79
80         Console.WriteLine("Stack: {0}", noFile.StackTrace);
81     }
82     catch (Exception some_exception)
83     {
84
85         Console.WriteLine("Error: {0}", some_exception.Message);
86
87         Console.WriteLine("Stack: {0}", some_exception.StackTrace);
88     }
89 }

```

```
77         }
78     }
79 }
80 }
```

5.2 Текст программы 2

```
1  using System;
2
3  namespace OOP5LabaMatrix
4  {
5      class MatrixMultiply
6      {
7          static void Main(string[] args)
8          {
9              int[,] a = new int[2, 2];
10             int[,] b = new int[2, 2];
11             int[,] result = new int[2, 2] { { 0, 0 }, { 0, 0 } };
12
13             Console.WriteLine("Enter first matrix 2x2");
14
15             for (int i = 0; i < 2; i++)
16             {
17                 string[] data = Console.ReadLine().Split(new char[] { ' ' }
18 );
19                 for (int j = 0; j < 2; j++)
20                 {
21                     a[i, j] = int.Parse(data[j]);
22                 }
23
24                 Console.WriteLine("Enter second matrix 2x2");
25
26                 for (int i = 0; i < 2; i++)
27                 {
28                     string[] data = Console.ReadLine().Split(new char[] { ' ' }
29 );
30                     for (int j = 0; j < 2; j++)
31                     {
32                         b[i, j] = int.Parse(data[j]);
33                     }
34 }
```



```

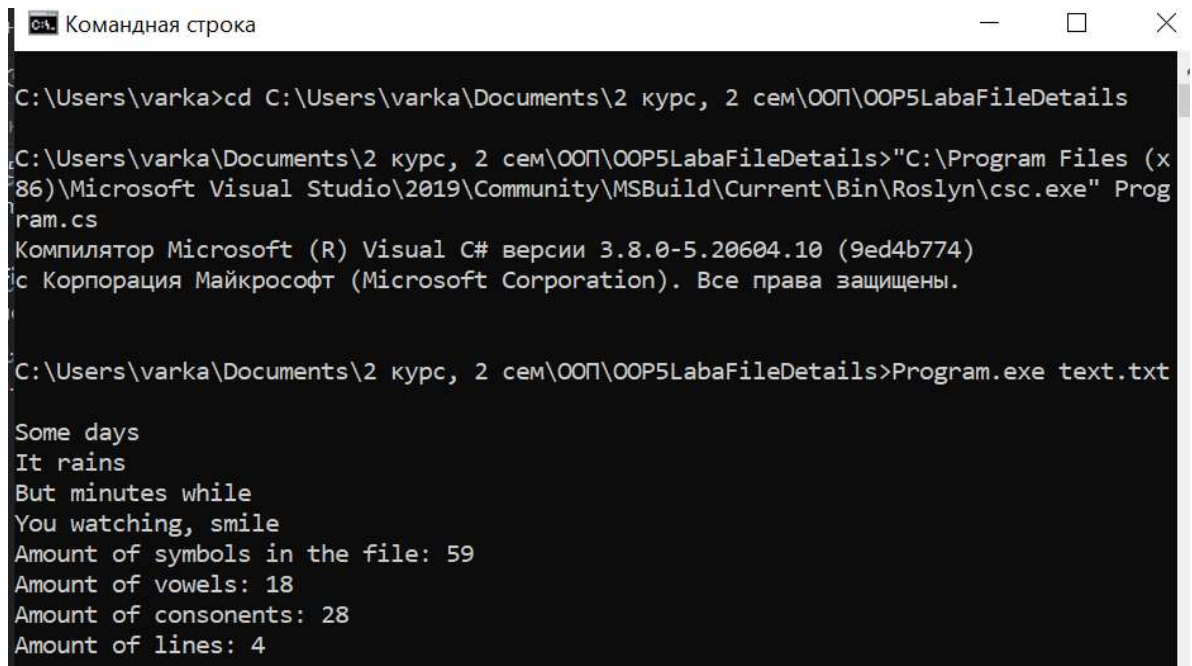
35         for (int i = 0; i < 2; i++)
36         {
37             for (int j = 0; j < 2; j++)
38             {
39                 for (int k = 0; k < 2; k++)
40                     result[i, j] += a[i, k] * b[k, j];
41             }
42         }
43
44         Console.WriteLine("the result of multiplication first on second
45         :");
46         for (int i = 0; i < 2; i++)
47         {
48             for (int j = 0; j < 2; j++)
49             {
50                 Console.Write(" {0}", result[i, j]);
51             }
52             Console.WriteLine();
53         }
54     }
55 }

```

6 Контрольный пример

6.1 Контрольный пример для 1 программы

На Рис. 6.1.1 показан вызов программы 1 из командной строки и результат работы программы: вывод текста файла и его анализа.



```
C:\Users\varka>cd C:\Users\varka\Documents\2 курс, 2 сем\ООП\ООП5LabaFileDetails

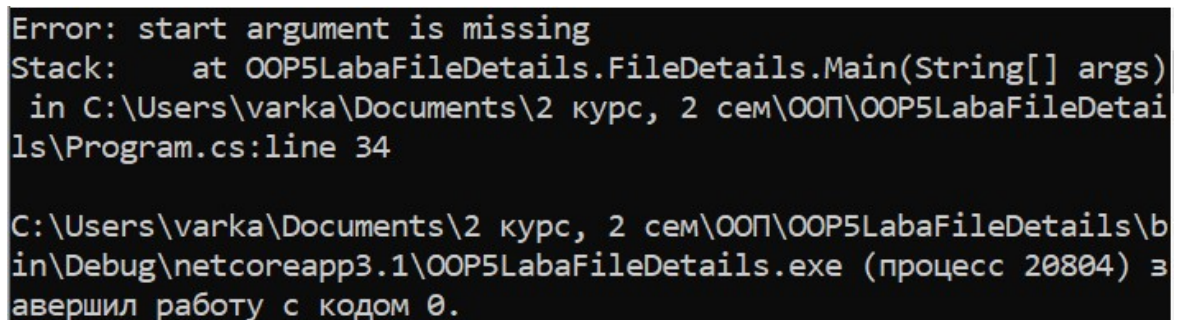
C:\Users\varka\Documents\2 курс, 2 сем\ООП\ООП5LabaFileDetails>"C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\MSBuild\Current\Bin\Roslyn\csc.exe" Program.cs
Компилятор Microsoft (R) Visual C# версии 3.8.0-5.20604.10 (9ed4b774)
© Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\varka\Documents\2 курс, 2 сем\ООП\ООП5LabaFileDetails>Program.exe text.txt

Some days
It rains
But minutes while
You watching, smile
Amount of symbols in the file: 59
Amount of vowels: 18
Amount of consonents: 28
Amount of lines: 4
```

Рис. 6.1.1 — Вызов программы 1 из командной строки

На Рис. 6.1.2 показана обработка исключения при вызове программы без указания параметра — названия текстового файла.



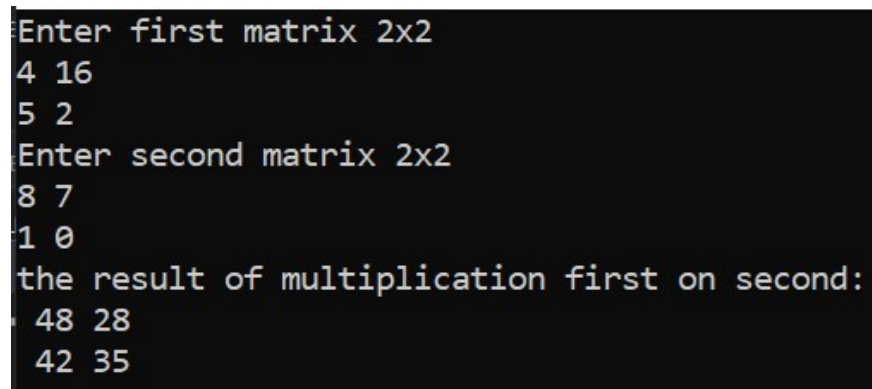
```
Error: start argument is missing
Stack:   at OOP5LabaFileDetails.FileDetails.Main(String[] args)
         in C:\Users\varka\Documents\2 курс, 2 сем\ООП\ООП5LabaFileDetails\Program.cs:line 34

C:\Users\varka\Documents\2 курс, 2 сем\ООП\ООП5LabaFileDetails\bin\Debug\netcoreapp3.1\ООП5LabaFileDetails.exe (процесс 20804) завершил работу с кодом 0.
```

Рис. 6.1.2 — Обработка исключения программой 1

6.2 Контрольный пример для 2 программы

На Рис. 6.2.1 показана работа программы 2.



```
Enter first matrix 2x2
4 16
5 2
Enter second matrix 2x2
8 7
1 0
the result of multiplication first on second:
48 28
42 35
```

Рис. 6.2.1 — перемножение двух матриц программой 2.

7 Полученные результаты

1. По результатам работы программы 1 был выведен текст указанного пользователем файла, количество символов в этом файле, количество гласных и согласных букв, количество строк.
2. По результатам работы программы 2 в консоль была выведена матрица, равная произведению двух данных.

8 Выводы

В ходе выполнения данной лабораторной работы:

1. Были получены навыки использования одномерных и двумерных массивов.
2. Были изучены методы чтения из файла, способы преобразования полученных данных с консоли.
3. Практиковались навыки обращения со ссылочными параметрами при создании и вызове метода.
4. Были получены знания о работе с командной строкой.

9 Список использованных источников

1. Руководство по языку C#: URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения: 01.03.2021)
2. Материалы учебного курса по ООП: URL: <https://vec.etu.ru/moodle/course/view.php?id=5512> (дата обращения: 01.03.2021)