

Министерство науки и высшего образования Российской Федерации  
Севастопольский государственный университет  
Кафедра ИС

Отчет  
по лабораторной работе №1  
"Встроенные типы данных в С#. Массивы. Строки. Регулярные  
выражения"  
по дисциплине  
ПЛАТФОРМА .NET

Выполнил студент группы ИС/б-22о  
Горбенко К. Н.  
Проверил  
ст. преп. Забаштанский А.К.

Севастополь  
2019

## 1 ЦЕЛЬ РАБОТЫ

- изучить классификацию типов данных и отличительные особенности синтаксических конструкций языка C# от C++;
- изучить базовые типы: Array, String, StringBuilder, а также средства стандартного ввода/вывода и возможности форматирования вывода;
- получить понятие о регулярных выражениях и их применении для поиска, замены и разбиения текста на синтаксические лексемы.

## 2 ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ

Для **варианта № 3** заданы следующие задания:

1. Проработать примеры программ 1-8, данные в теоретических сведениях. Создать на их основе программы. Получить результаты работы программ и уметь их объяснить. Внести их в отчет по работе с комментариями.
2. Найти номер столбца двумерного массива целых чисел, для которого среднеарифметическое его элементов максимально.
3. Создать программу, которая будет вводить строку в переменную String. Найти в ней те слова, которые начинаются и заканчиваются одной и той же буквой.
4. Задан текст. Определить, является ли он текстом на английском языке.

### 3 ХОД РАБОТЫ

#### 3.1 Примеры программ на C#

##### 3.1.1 Пример № 1

Результат работы программы № 1:

```
Here is the array:
Саша
Маша
Олег
Света
Игорь

Игорь
Света
Олег
Маша
Саша

Cleared out all but one...
Игорь
```

Рисунок 1 – Результат работы программы из примера № 1

В примере № 1 демонстрируются статические методы Reverse и Clear класса Array.

##### 3.1.2 Пример № 2

Результат работы программы № 2:

```
oneDimensionalArray
1234
twoDimensionalArray
1      2      3      4
5      6      7      8
```

Рисунок 2 – Результат работы программы из примера № 2

В программе демонстрируется возможность передать любой массив в функцию с помощью абстрактного класса Array. При этом индексирование массива происходит с помощью методов GetValue.

### 3.1.3 Пример № 3

Результат работы программы № 3:

```
strM1
Здравствуй, Мир!
World
Мир
Мир
```

Рисунок 3 – Результат работы программы из примера № 3

В данной программе исследуется тип данных `char` и возможность перехода от массива символов к строке и наоборот. Для перевода строки в массив символов используется метод класса `String ToCharArray`. Для обратной операции используется цикл `for`. Однако лучше это сделать с помощью конструктора класса `String`, принимающего массив символов.

### 3.1.4 Пример № 4

Результат работы программы № 4:

```
s1=ABCCDE, s2=CDE, b1=True, ch1=A, ch2=C
Lenon
```

Рисунок 4 – Результат работы программы из примера № 4

В данной программе демонстрируется класс `StringBuilder`, который используется для увеличения производительности при работе со строками. Основные методы: `Append`, `Remove`, `Insert`.

### 3.1.5 Примеры № 5, № 6, № 7

В программах № 5, № 6 и № 7 исследуются способы использования регулярных выражений для поиска соответствий в строке. Для данной цели могут использоваться статические методы `Matches` и `Match`, принимающие как строку, так и шаблон, а также те же методы класса `Regex`, инициализированного с шаблоном в конструкторе.

### 3.1.6 Пример № 8

```
theMatch.Length: 4
theMatch: Это
theMatch.Length: 7
theMatch: строка
theMatch.Length: 4
theMatch: для
```

Рисунок 5 – Результат работы программы из примера № 4

В данной программе в шаблон регулярного выражения включаются именованные группы.

## 3.2 Индивидуальное задание для работы над массивами

Была написана следующая программа:

```
1 public static class ArrayOperations
2 {
3     public static int GetMaximalAverageColumn(int[,] matrix)
4     {
5         if (matrix == null)
6             throw new ArgumentNullException($"{nameof(matrix)} instance was null");
7
8         var columnsAverages = matrix
9             .GetColumns()
10            .Select(x => Enumerable.Average(x))
11            .ToArray();
12
13         return Array.IndexOf(columnsAverages, Enumerable.Max(columnsAverages));
14     }
15
16     public static void PrintMatrix(int[,] matrix)
17     {
18         if (matrix == null)
19             throw new ArgumentNullException($"{nameof(matrix)} instance was null");
20
21         for (var i = 0; i < matrix.GetLength(0); i++)
22         {
23             for (var j = 0; j < matrix.GetLength(1); j++)
24                 Console.Write($"{matrix[i, j]} ");
25
26             Console.WriteLine();
```

```

27     }
28 }
29 }
30
31 public static class ExtensionMethods
32 {
33     public static IEnumerable<int[]> GetColumns(this int[,] array)
34     {
35         if (array == null)
36             throw new ArgumentNullException($"{nameof(array)} instance was null");
37
38         for (var j = 0; j < array.GetLength(1); j++)
39         {
40             var column = new int[array.GetLength(0)];
41
42             for (var i = 0; i < array.GetLength(0); i++)
43                 column[i] = array[i, j];
44
45             yield return column;
46         }
47     }
48 }

```

В данной программе активно используется библиотека System.Linq для операций над коллекциями. Также был написан extension-метод для извлечения столбцов матрицы. Для определения позиции с наибольшим средним столбцом в строке использовался статический метод IndexOf класса Array.

### 3.3 Индивидуальное задание для работы над строками

Была написана следующая программа:

```

1 public static class StringOperations
2 {
3     public static IEnumerable<string> GetWordsStartingAndEndingTheSameLetter(string line)
4         => line.Split(" ").Where(x => x.First() == x.Last());
5 }

```

В данной программе используется метод Split класса String для разделения строки на подстроки, а также библиотека System.Linq для фильтрации полученной коллекции. Условием для фильтра является равенство результатов выполнения методов First и Last класса String, возвращающих соответственно первый и последний символы строки.

### 3.4 Индивидуальное задание для работы с регулярными выражениями

Была написана следующая программа:

```

1 public static class RegexOperations
2 {
3     private static string englishRegexTemplate = @"^[A-Za-z0-9 .,!?;:()]+$";
4
5     public static bool IsInEnglish(string test)
6         => Regex.IsMatch(test, englishRegexTemplate);
7 }

```

В данной программе создан класс, полем которого является шаблон регулярного выражения, осуществляющего проверку того, что все встреченные в строке символы являются либо буквами английского алфавита, либо арабскими цифрами, либо пробелами, либо знаками пунктуации.

Для вызова регулярного выражения использовался статический метод `Regex.IsMatch`, который принимает как входную строку, так и сам шаблон регулярного выражения.

## 4 ВЫВОД

В ходе лабораторной работы были изучены массивы, строки и регулярные выражения языка C#, а также встроенные средства для работы с ними.

Все массивы в среде .NET являются наследниками абстрактного класса `Array`, что позволяет передавать в одну и ту же функцию любой массив. Кроме того, класс `Array` предоставляет множество статических методов и методов экземпляра для эффективной работы с массивами.

Строки в платформе .NET являются неизменяемыми, что приводит к просадкам в производительности при выполнении множества операций над ними. Для избежания проблем с производительностью в таких случаях используют класс `StringBuilder`. Класс `String` также содержит множество методов для работы со строками.

Регулярные выражения - мощный инструмент для работы со строками. Он позволяет проверять строку на соответствие шаблону и производить поиск подстрок в строке. Соответствующие операции обычными методами требуют

намного большего времени и усилий.