**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ**

***Институт Принтмедиа и информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 22**

**Дисциплина:** Введение в программирование

**Выполнил(а): студент(ка) группы \_191-726\_\_**

\_\_\_\_\_Мухина Н.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия И.О.)

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Проверил: \_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2019**

Оглавление

[Теория: 3](#_Toc28267920)

[Задания: 4](#_Toc28267921)

[Код программы: 5](#_Toc28267922)

[Результат программы: 10](#_Toc28267923)

# Теория:

Простейшей формой многомерного массива является двумерный массив. Местоположение любого элемента в двумерном массиве обозначается двумя индексами. Такой массив можно представить в виде таблицы, на строки которой указывает один индекс, а на столбцы — другой. В следующей строке кода объявляется двумерный массив integer размерами 10×20. int[,] table = new int[10, 20]; Обратите особое внимание на объявление этого массива. Как видите, оба его размера разделяются запятой. В первой части этого объявления синтаксическое обозначение [,] означает, что создается переменная ссылки на двумерный массив. Если же память распределяется для массива с помощью оператора new, то используется следующее синтаксическое обозначение. int[10, 20] В данном объявлении создается массив размерами 10×20, но и в этом случае его размеры разделяются запятой. Для доступа к элементу двумерного массива следует указать оба индекса, разделив их запятой. Например, в следующей строке кода элементу массива table с координатами местоположения (3,5) присваивается значение 10. table[3, 5] = 10; Ниже приведен более наглядный пример в виде небольшой программы, в которой двумерный массив сначала заполняется числами от 1 до 12, а затем выводится его содержимое.

В C# допускаются массивы трех и более измерений. Ниже приведена общая форма объявления многомерного массива. тип[,...,] имя\_массива = new тип[размер1, размер2, ... размерN]; Например, в приведенном ниже объявлении создается трехмерный целочисленный массив размерами 4×10×3. int[,,] multidim = new int[4, 10, 3]; А в следующем операторе элементу массива multidim с координатами местоположения (2,4,1) присваивается значение 100. multidim[2, 4, 1] = 100;

Для инициализации многомерного массива достаточно заключить в фигурные скобки список инициализаторов каждого его размера.

# Задания:

1. Дан символьный файл, содержащий по крайней мере один символ пробела. Удалить все его элементы, расположенные перед первым символом пробела, включая и этот пробел.

2. Дано имя файла и целые положительные числа N и K. Создать текстовый файл с указанным именем и записать в него N строк, каждая из которых состоит из K символов «\*» (звездочка).

3. Даны два текстовых файла. Добавить в начало первого файла содержимое второго файла

4. Дан текстовый файл. Заменить в нем все подряд идущие пробелы на один пробел.

5. Дан текстовый файл. Найти количество абзацев в тексте, если первая строка каждого абзаца начинается с 5 пробелов («красная строка»). Пустые строки между абзацами не учитывать.

# Код программы:

Листинг 1- Задание 1

|  |
| --- |
| 1. using System; 2. using System.Collections.Generic; 3. using System.Linq; 4. using System.Text; 5. using System.Threading.Tasks; 6. using System.IO; 7. namespace \_1 8. { 9. class Program 10. { 11. static void Main(string[] args) 12. { 13. string filename =("C:/Users/Надя/source/repos/Лабораторная.txt"); 14. Console.WriteLine("Вывод:"); 15. using (StreamReader sr = new StreamReader(filename)) 16. { 17. string file = sr.ReadToEnd(); 18. int whiteSpaceIndex = file.IndexOf(' '); 19. file = file.Remove(0, whiteSpaceIndex + 1); 20. Console.WriteLine(file); 21. } 22. Console.ReadKey(); 23. } 24. } 25. } |

Листинг 2- Задание 2

|  |
| --- |
| 1. using System; 2. using System.Collections.Generic; 3. using System.Linq; 4. using System.Text; 5. using System.Threading.Tasks; 6. using System.IO; 7. namespace \_2 8. { 9. class Program 10. { 11. static void Main(string[] args) 12. { 13. Console.WriteLine("Введите адрес файла: "); 14. string path = Console.ReadLine(); 15. Console.WriteLine("Введите N: "); 16. int n = Convert.ToInt32(Console.ReadLine()); 17. Console.WriteLine("Введите K=1: "); 18. int k = Convert.ToInt32(Console.ReadLine()); 19. IEnumerable<string> s = Enumerable.Range(1, k).Select(v => "\*"); 20. for (int i = 0; i < n; i++) 21. { 22. File.AppendAllText(path, String.Join("", s)); 23. File.AppendAllText(path, Environment.NewLine); 24. } 25. string s0 = File.ReadAllText(path); 26. Console.WriteLine("Вывод: \n" + s0); 27. Console.ReadKey(); 28. } 29. } 30. } |

Листинг 3- Задание 3

|  |
| --- |
| 1. using System; 2. using System.Collections.Generic; 3. using System.Linq; 4. using System.Text; 5. using System.Threading.Tasks; 6. using System.IO; 7. namespace \_3 8. { 9. class Program 10. { 11. static void Main(string[] args) 12. { 13. string path = @"C:\Users\Надя\source\repos\Лабораторная.txt"; 14. string way = @"C:\Users\Надя\source\repos\ЛЯЛЯЛЯ.txt"; 15. string s = File.ReadAllText(path); 16. string s0 = File.ReadAllText(way); 17. File.WriteAllText(path, s0+s); 18. Console.ReadKey(); 19. } 20. } 21. } |

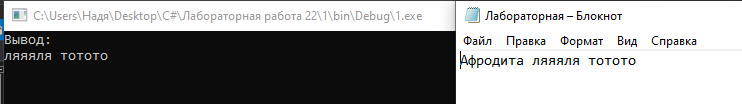
Листинг 4- Задание 4

|  |
| --- |
| 1. using System; 2. using System.Collections.Generic; 3. using System.Linq; 4. using System.Text; 5. using System.Threading.Tasks; 6. using System.IO; 7. using System.Text.RegularExpressions; 8. namespace \_22.\_4 9. { 10. class Program 11. { 12. static void Main(string[] args) 13. { 14. string path = @"C:\Users\Надя\source\repos\Лабораторная.txt"; 15. string s = File.ReadAllText(path); 16. string Str = new Regex(@"\s+").Replace(s, " "); 17. Console.WriteLine(Str); 18. File.WriteAllText(path, Str); 19. Console.ReadKey(); 20. } 21. } 22. } |

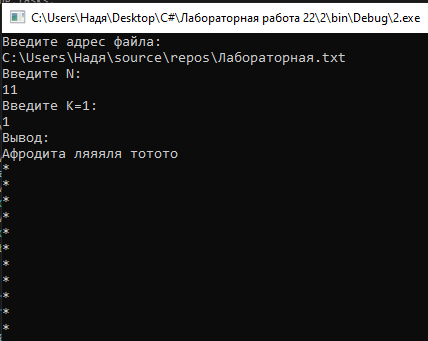
Листинг 5- Задание 5

|  |
| --- |
| 1. using System; 2. using System.Collections.Generic; 3. using System.Linq; 4. using System.Text; 5. using System.Threading.Tasks; 6. using System.IO; 7. namespace \_22.\_5 8. { 9. class Program 10. { 11. static void Main(string[] args) 12. { 13. int x = 0; 14. foreach (string path in File.ReadLines(@"C:\Users\Надя\source\repos\Лабораторная.txt")) 15. { 16. if (path.Contains(" ")) 17. { 18. x++; 19. } 20. } 21. Console.WriteLine("Количество абзацов в тексте: " + x); 22. Console.ReadKey(); 23. } 24. } 25. } |

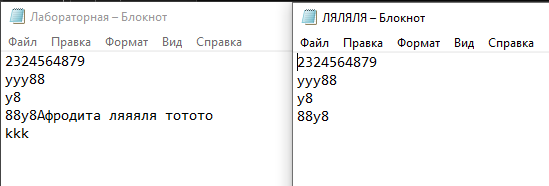
# Результат программы:



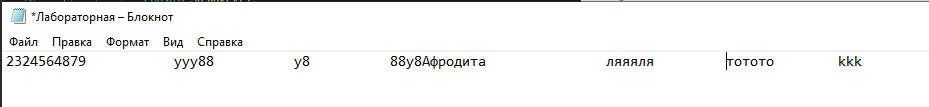
**Рис.1-Скриншот работы программы Задание №1**

****

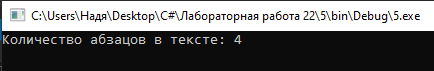
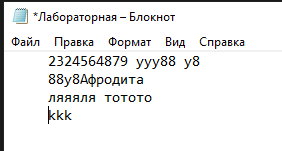
**Рис.2-Скриншот работы программы Задание №2**

****

**Рис.3-Скриншот работы программы Задание №3**

****

**Рис.4-Скриншот работы программы Задание №4**

****

**Рис.5-Скриншот работы программы Задание №5**