

# Модуль 1

## Практическое занятие 13

**Работа с файлами**  
**Чтение и запись данных для**  
**массивов из файлов**

# Задача 2

- Разработать консольное приложение для управления файловой системой компьютера
- Изучите работу предложенного кода и использованные методы
- Проверьте работоспособность кода по всем, предложенным в меню командам
- По необходимости доработайте программу (необходимо декомпонировать)
- Код  
(<https://repl.it/@Maksimenkova/ArrayFiles02>)

# Задача 2

- В предложенной далее программе разработан:
  - метод преобразования целых чисел в строку чисел, разделённых пробелами
  - метод записи в текстовый файл последовательности целых случайных чисел, разделённых пробелами
  - метод чтения из файла последовательности и преобразования ее в целочисленный массив
- К сожалению, работа предложенного кода завершается исключением
  - Внесите правки в код, завершите распределение исходного кода по методам
  - Предложите решение, не требующее выбрасывания исключений в методах для управления ходом выполнения программы

# Задача 2

```
static Random rnd = new Random();
static char[] invalidPathChars = Path.GetInvalidPathChars();
static string path = @"output.txt";
public static string TestDataToString(int nmbOfValues)
{
    string dataToFile = string.Empty;
    if (nmbOfValues < 0)
    {
        throw new ArgumentOutOfRangeException();
    }
    for (int i = 0; i < nmbOfValues; i++) // create data
    {
        dataToFile += rnd.Next(10).ToString() + " ";
    }
    return dataToFile;
}
```

# Задача 2

```
public static void PrintIntArrToConsole(int[] arr)
{
    foreach (int i in arr)
        Console.Write(i + " ");
}
//reading file with data
public static int[] GetDataFromFile(string path)
{
    int[] data = null;
    if (path.IndexOfAny(InvalidPathChars) != -1 || !File.Exists(path))
    {
        throw new FileNotFoundException();
    }
    string dataFromFile = File.ReadAllText(path);
    string[] dataFromString = dataFromFile.Split(" ", StringSplitOptions.RemoveEmptyEntries);
    data = new int[dataFromString.Length];
    for (int i = 0; i < dataFromString.Length; i++)
    {
        data[i] = int.Parse(dataFromString[i]);
    }
    return data;
}
```

# Задача 2

```
public static void Main()
{
    // TODO: Extract writing to a method
    // writing file with data
    if (path.IndexOfAny(invalidPathChars) == -1)
    {
        File.WriteAllText(path, TestDataToString(10));
    }
    int[] arr = GetDataFromFile(path);
    PrintIntArrayToConsole(arr);
}
```

# Задача 3

- В проекте консольного приложения создать файл **Data.txt** и заполнить его случайными целыми значениями из диапазона **[10;100)**
- Записать данные в файл через один пробел, по пять в строку
- Прочитать данные из этого файла и поместить их в целочисленный массив
- По массиву создать два индексных массива, содержащих индексы элементов с чётными и нечётными значениями
- После чего в основном массиве заменить нечётные числа нулями
- Дополнительно декомпозировать предложенный код заготовки программы

**Здесь и далее: не забудьте про обработку исключений, воспользуйтесь дополнительными проверками из кода задачи 2**

# Задача 3

```
static void Main(string[] args) {
    string path = @"Data.txt";
    // Создаем файл с данными
    // Сейчас данные для записи вбиты в коде
    // TODO1: сохранить в файл целые случайные значения из диапазона
    [10;100)
    string createText = "10 20 30 40 50" + Environment.NewLine +
        "60 70 80 90";
    File.WriteAllText(path, createText, Encoding.UTF8);

    // Open the file to read from
    if (File.Exists(path)) {
        string readText = File.ReadAllText(path);
        string[] stringValue = readText.Split(' ');
        int[] arr = StringArrayToIntArray(stringValue);
        foreach(int i in arr) { Console.Write(i + " "); }
    }
} // end of Main()
```



# Задача 3

```
// преобразование массива строк в массив целых чисел
public static int[] StringArrayToIntArray(string[] str) {
    int[] arr = null;
    if (str.Length != 0) {
        arr = new int[0];
        foreach (string s in str) {
            int tmp;
            if (int.TryParse(s, out tmp)) {
                Array.Resize(ref arr, arr.Length + 1);
                arr[arr.Length - 1] = tmp;
            }
        } // end of foreach (string s in str)
    }
    return arr;
} // end of StringToIntArray()
```

**// Красивая альтернатива**

```
public static int[] StringArrayToIntArray(string[] str)
=> Array.ConvertAll(str, s => {
    int.TryParse(s, out int tmp); return tmp; });
```

# Задача 4

- У многих людей развился необычный страх - пунктуофобия - боязнь точек в конце предложения.
- Ваша задача не показывать точки в конце предложений при выводе на экран.
- Написать программу, которая:
  1. Генерирует случайным образом 6 предложений длиной от 20 до 50 символов (алфавит – все строчные символы кириллицы), заканчивающихся точкой.
  2. Записывает предложения построчно (каждое предложение на отдельной строке файла) в текстовый файл **dialog.txt** в кодировке UTF-8, расположенный в папке с исполняемым файлом.
  3. Выводит на экран то, что было записано в файл изначально.
  4. Открывает полученный файл и выводит на экран построчно все предложения из него, удаляя точки в конце предложения.
- Заготовка кода (<https://repl.it/@Maksimenkova/ArrayFiles03>)

# Задача 4

```
class Program {
    static Random rand = new Random();
    static void Main() {
        // ОСНОВНЫЕ НАСТРОЙКИ
        const string fileName = "dialog.txt";
        Encoding enc = Encoding.Unicode;
        const int linesCount = 6; // кол-во предложений
        // Создаем файл на диске
        File.WriteAllText(fileName, string.Empty, enc); // Создаём
пустой файл.
        Console.WriteLine("Переписка до форматирования:");
        for (int i = 0; i < linesCount; i++) {
            string message = string.Empty; // предложение
            int length = rand.Next(20, 51); // Длина сообщения
            for (int j = 0; j < length; j++) {
```

# Задача 4

```
        message += (char)rand.Next('А', 'Я' + 1); // Посимвольное
добавление букв в сообщение. "Ё" нет в диапазоне от А до Я!
    }
    message += "." + Environment.NewLine; // Добавляем в сообщение
точку и перенос на следующую строку.
    File.AppendAllText(fileName, message, enc); // Добавляем строку
в файл.
    Console.WriteLine(message);
}

// читаем сформированный файл и обрабатываем предложения
string[] messages = File.ReadAllLines(fileName, enc); // Массив
сообщений из переписки.
    Console.WriteLine(Environment.NewLine + "Переписка после
форматирования:");
    foreach (string item in messages)
        Console.WriteLine(item.Substring(0, item.Length - 1)); //
Выводим сообщения из переписки без точек на экран.
    }
}
```

# Задача 5: разделение строки ПО СИМВОЛАМ

```
string[] TestStrings = { "", "a b c", "Let it be", "All You Need Is Love"
};

foreach (string str in TestStrings)
{
    Console.WriteLine("Исходная строка: " + str + "");
    string[] tmp = str.Split(' ');
    // результат выведен без лишних пробелов
    foreach (string s in tmp)
    {
        Console.Write(s);
    }
    Console.WriteLine();
    // добавим индексы, чтобы "увидеть" элементы с пустыми строками
    for (int i = 0; i < tmp.Length; i++)
    {
        Console.WriteLine(i + ": " + tmp[i]);
    }
}
```

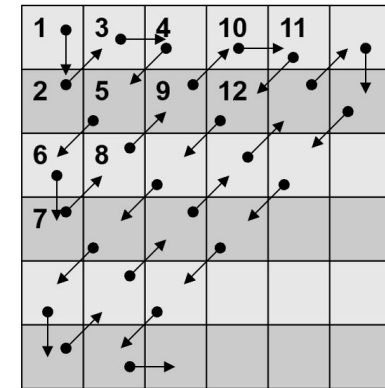
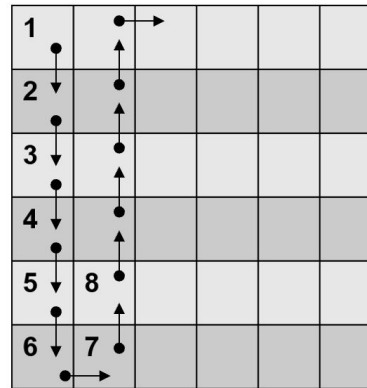
# Задача 6

На основе предложенного кода предыдущей задачи самостоятельно выполните чтение из файла строки, содержащей значения, разделённые точкой с запятой и выделите оттуда целые числа, разместив их в массиве.

- Массив целых чисел вывести на экран;
- Для массива целых чисел вычислить и вывести на экран среднее значение элементов (с точностью до трех знаков после десятичного разделителя).
- Полученное среднее значение записать в два файла:
  - текстовый (в любой кодировке),
  - двоичный (в типе float).

# Задачи для самостоятельного решения

- Разработать методы чтения и записи в текстовый файл целочисленных квадратных матриц  $N \times N$  (матрица сохраняется строкой, в которой значения разделены пробелами, а строки точкой с запятой). Метод чтения возвращает тип данных – двумерный массив (прямоугольный)
- Разработать методы заполнения матриц по следующим правилам:



Предполагается, что ваш код обрабатывает исключения, некорректные входные данные и позволяет повторно перезапускать программу.

\* Добавьте проверку существования файла на диске при повторной работе с программой и для каждого новых данных создавайте новый файл с другим именем, например, file1.txt, file2.txt

# Задачи для самостоятельного решения

Во всех задачах данные для массивов получить из текстового файла **input.txt**, результирующие массивы также сохранять в текстовый файл **output.txt**.

1. По массиву **A** целых чисел со значениями из диапазона **[-10;10]** создать массив булевских значений **L**. Количество элементов в массивах совпадает. На местах положительных элементов в массиве **A** в массиве **L** стоят значения **True**, на месте отрицательных – **False**.
2. По массиву **A** целых чисел со значениями из диапазона **[0; 10000]** создать массив целых чисел **B**, в котором на каждой позиции стоит ближайшая степень двойки меньшая значения из массива **A** у той же позиции. Например,  $A = \{30, 100, 300\}$   $B = \{16, 64, 256\}$