

## Практическая работа 13

### Методы работы с файлами

#### Цель занятия

Получить практические навыки анализа предметной области

#### Перечень оборудования и программного обеспечения

Персональный компьютер  
Microsoft Office (Word, Visio)  
Microsoft Visual Studio 2010

#### Краткие теоретические сведения

Класс File пространства имен System.IO предоставляет статические методы для создания, копирования, удаления, перемещения и открытия файлов, а также помогает при создании объектов FileStream.

Для того, чтобы работать с файлами, необходимо:

- подключить пространство имен: using System.IO;
- использовать класс File с методами:

File.ReadAllLines - считывает из текстового файла все строки и закрывает файл;

File.ReadAllBytes - считывает двоичный файл и записывает результат в массив байтов и закрывает файл;

File.ReadAllText - считывает текстовый файл с заданной кодировкой;

File.ReadLines - считывает строки с заданной кодировкой.

#### Пример:

Создайте форму и четыре кнопки для создания, удаления, записи и чтения файла

```
using System.IO; //Подключаем пространство имен
namespace Files
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            FileInfo file = new FileInfo("vbbook.txt");
        }
    }
}
```

```

/**
 * Создание файлов
 */
private void button1_Click(object sender, EventArgs e)
{
    FileInfo file = new FileInfo("vbbook.txt");
    if (file.Exists == false) //Если файл не существует
    {
        file.Create(); //Создаем
    }
    else MessageBox.Show("Файл уже создан!");
}
/**
 * Удаляем файл
 */
private void button2_Click(object sender, EventArgs e)
{
    FileInfo file = new FileInfo("vbbook.txt");
    if (file.Exists == true) //Если файл существует
    {
        file.Delete(); //Удаляем
    }
    else MessageBox.Show("Файла не существует!!");
}
/**
 * Записать текст в файл
 */
private void button3_Click(object sender, EventArgs e)
{
    StreamWriter write_text; //Класс для записи в файл
    FileInfo file = new FileInfo("vbbook.txt");
    write_text = file.AppendText(); //Дописываем инфу в файл, если
файла не существует он создается
    write_text.WriteLine(textBox1.Text); //Записываем в файл текст из
текстового поля
    write_text.Close(); // Закрываем файл
}
private void button4_Click(object sender, EventArgs e)
{
    StreamReader streamReader = new StreamReader("vbbook.txt");
//Открываем файл для чтения
    string str = ""; //Объявляем переменную, в которую будем
записывать текст из файла
    while (!streamReader.EndOfStream) //Цикл длиться пока не будет
достигнут конец файла

```

```

        {
            str += streamReader.ReadLine(); //В переменную str по строчно
записываем содержимое файла
        }
        textBox1.Text = str;
    }

```

## Чтение и запись файлов

Класс File предоставляет ряд методов для чтения-записи текстовых и бинарных файлов:

- AppendAllLines(String, IEnumerable<String>) / AppendAllLinesAsync(String, IEnumerable<String>, CancellationToken) добавляют в файл набор строк. Если файл не существует, то он создается
- AppendAllText(String, String) / AppendAllTextAsync(String, String, CancellationToken) добавляют в файл строку. Если файл не существует, то он создается
- byte[] ReadAllBytes (string path) / Task<byte[]> ReadAllBytesAsync (string path, CancellationToken cancellationToken) считывают содержимое бинарного файла в массив байтов
- string[] ReadAllLines (string path) / Task<string[]> ReadAllLinesAsync (string path, CancellationToken cancellationToken) считывают содержимое текстового файла в массив строк
- string ReadAllText (string path) / Task<string> ReadAllTextAsync (string path, CancellationToken cancellationToken) считывают содержимое текстового файла в строку
- IEnumerable<string> ReadLines (string path) считывают содержимое текстового файла в коллекцию строк
- void WriteAllBytes (string path, byte[] bytes) / Task WriteAllBytesAsync (string path, byte[] bytes, CancellationToken cancellationToken) записывают массив байт в бинарный файл. Если файл не существует, он создается. Если существует, то перезаписывается
- void WriteAllLines (string path, string[] contents) / Task WriteAllLinesAsync (string path, IEnumerable<string> contents, CancellationToken cancellationToken) записывают массив строк в текстовый файл. Если файл не существует, он создается. Если существует, то перезаписывается
- WriteAllText (string path, string? contents) / Task WriteAllTextAsync (string path, string? contents, CancellationToken cancellationToken) записывают строку в текстовый файл. Если файл не существует, он создается. Если существует, то перезаписывается

Как видно, эти методы покрывают практически все основные сценарии - чтение и запись текстовых и бинарных файлов. Причем в зависимости от

задачи можно применять как синхронные методы, так и их асинхронные аналоги.

Например, запишем и считаем обратно в строку текстовый файл:

```
string path = @"c:\app\content.txt";

string originalText = "Hello Metanit.com";
// запись строки
await File.WriteAllTextAsync(path, originalText);
// дозапись в конец файла
await File.AppendAllTextAsync(path, "\nHello work");

// чтение файла
string fileText = await File.ReadAllTextAsync(path);
Console.WriteLine(fileText);
```

### **Консольный вывод:**

```
Hello Metanit.com
Hello work
```

Стоит отметить, что при добавлении текста добавлена последовательность "\n", которая выполняет перевод на следующую строку. Благодаря этому добавляемый текст располагается в файле на новой строке.

Если мы хотим, что в файле изначально шло добавление на новую строку, то для записи стоит использовать метод WriteAllLines/WriteAllLinesAsync, а для добавления - AppendAllLines / AppendAllLinesAsync

```
await File.WriteAllLinesAsync(path, new[] { "Hello Metanit.com", "Hello work"
});
```

Аналогично при чтении файла если мы хотим каждую строку файла считать отдельно, то вместо ReadAllText / ReadAllTextAsync применяется ReadAllLines / ReadAllTextAsync.

### **Кодировка**

В качестве дополнительного параметра методы чтения-записи текстовых файлов позволяют установить кодировку в виде объекта System.Text.Encoding:

```
using System.Text;

string path = "/Users/eugene/Documents/app/content.txt";
```

```

string originalText = "Привет Metanit.com";
// запись строки
await File.WriteAllTextAsync(path, originalText, Encoding.Unicode);
// дозапись в конец файла
await File.AppendAllTextAsync(path, "\nПривет мир", Encoding.Unicode);

// чтение файла
string fileText = await File.ReadAllTextAsync(path, Encoding.Unicode);
Console.WriteLine(fileText);

```

Для установки кодировки при записи и чтении здесь применяется встроенное значение `Encoding.Unicode`. Также можно указать название кодировки, единственное следует удостовериться, что текущая операционная система поддерживает выбранную кодировку:

```

using System.Text;

string path = @"c:\app\content.txt";

string originalText = "Hello Metanit.com";
// запись строки
await File.WriteAllTextAsync(path, originalText, Encoding.GetEncoding("iso-8859-1"));
// дозапись в конец файла
await File.AppendAllTextAsync(path, "\nHello code", Encoding.GetEncoding("iso-8859-1"));

// чтение файла
string fileText = await File.ReadAllTextAsync(path, Encoding.GetEncoding("iso-8859-1"));
Console.WriteLine(fileText);

```

## Задания

- 1 Изучить теоретические сведения и задание к работе.
- 2 Получить имена и свойства всех дисков на компьютере.
- 3 Выполнить следующие действия с файлами и папками:
  - создать текстовый файл на рабочем столе;
  - ввести данные в файл;
  - создать папку на рабочем столе;
  - скопировать созданный файл в папку;
  - дописать в файл строку;
  - удалить созданные файлы и папки.

4 В соответствии с вариантом задания составить спецификации программного продукта, разработать и отладить программный модуль Windows Form, в котором отобразить на форме содержимое исходного текстового файла, изменения данных для записи в файл результата.

5\* Выполнить дополнительные задания практической 8 и практической 9, организовав ввод данных из файла.

## Порядок выполнения работы (Пример выполнения)

### Задание 2

В список библиотек классов добавим пространство имен **System.IO**. Запишем код для просмотра имен и свойств всех дисков на компьютере, для чего создадим объект класса **DriveInfo** и в цикле отобразим параметры всех логических дисков.

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication14
{
    class Program
    {
        static void Main(string[] args)
        {
            DriveInfo[] drives = DriveInfo.GetDrives();

            foreach (DriveInfo drive in drives)
            {
                Console.WriteLine($"Название: {drive.Name}");
                Console.WriteLine($"Тип: {drive.DriveType}");
                if (drive.IsReady)
                {
                    Console.WriteLine($"Файловая система: {drive.DriveFormat}");
                    Console.WriteLine($"Объем диска: {drive.TotalSize}");
                    Console.WriteLine($"Свободное пространство: {drive.TotalFreeSpace}");
                    Console.WriteLine($"Метка: {drive.VolumeLabel}");
                    Console.WriteLine($"Корневой каталог: {drive.RootDirectory}");
                }
                Console.WriteLine();
            }
            Console.ReadKey();
        }
    }
}
```

```
file:///C:/Users/ГуныкоИА/documents/visual studio 2015/...
C:\>
g Название: C:\
g Тип: Fixed
g Файловая система: NTFS
g Объем диска: 59152265216
g Свободное пространство: 1681924096
g Метка: Windows
g Корневой каталог: C:\

D:\>
spr Название: D:\
spr Тип: Fixed
spr Файловая система: NTFS
spr Объем диска: 1000200990720
spr { Свободное пространство: 505887010816
spr Метка: My Passport
spr Корневой каталог: D:\

E:\>
spr Название: E:\
spr Тип: Removable
spr Файловая система: FAT32
spr Объем диска: 1027571712
spr Свободное пространство: 1026252800
spr Метка:
spr Корневой каталог: E:\

H:\>
spr Название: H:\
spr Тип: Network
spr Файловая система: NTFS
spr Объем диска: 685629239296
spr Свободное пространство: 50069782528
spr Метка: Новый том
spr Корневой каталог: H:\

T:\>
spr Название: T:\
spr Тип: Network
spr Файловая система: NTFS
spr Объем диска: 157526888448
spr Свободное пространство: 13944053760
spr Метка: adm
spr Корневой каталог: T:\

X:\>
spr Название: X:\
spr Тип: Network
spr Файловая система: NTFS
spr Объем диска: 685629239296
spr Свободное пространство: 50069782528
spr Метка: Новый том
spr Корневой каталог: X:\
```

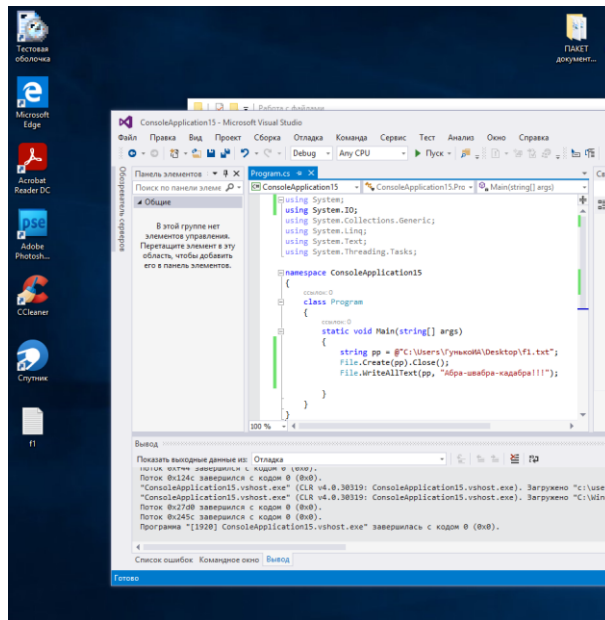
### Задание 3

Создадим файл на рабочем столе, для удобства запишем путь к файлу в строковую переменную pp. Чтобы можно было работать с этим файлом, необходимо его закрыть, иначе при следующей попытке открыть и записать в него текст, будет сообщение о том, что этот файл занят другим пользователем, для чего добавлено .Close().

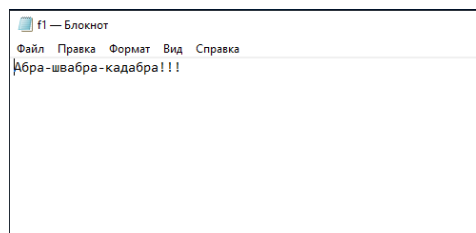
```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication15
{
    class Program
    {
        static void Main(string[] args)
        {
            string pp = @"C:\Users\ГуныкоИА\Desktop\fl.txt";
            File.Create(pp).Close();
            File.WriteAllText(pp, "Абра-швабра-кадабра!!!");
        }
    }
}
```

В результате на рабочем столе появился файл fl.



А в нем текст:



...

## Задание 4

### Исходные данные:

Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа или слова, разделённые пробелами. Создать файл, который будет содержать данные исходного файла, в котором в строке с числами добавлена сумма этих чисел, а в строке со словами добавлена сумма символов в этих словах.

### Решение:

#### 1 Спецификации

Наименование: Программа обработки данных текстовых файлов

Язык разработки: C#

Среда разработки: MS Visual Studio 2015



Цель: Создать программу, демонстрирующую способы чтения и записи данных в текстовый файл.

Входные данные: c:\\proba\\probin.txt.

Навигация:

Чтобы при считывании из текстового файла русских букв они отображались нормально, нужно при сохранении файла изменить кодировку сохраняемого .txt. По умолчанию стоит ANSI, нужно поменять ее на Юникод или UTF-8.

При нажатии на кнопку «Чтение файла» строки из файла c:\\proba\\probin.txt отображаются в ListBox1.

При нажатии на кнопку «Запись файла» строки из ListBox1 обрабатываются программой, записываются в ListBox2 и в файл c:\\proba\\probout.txt.

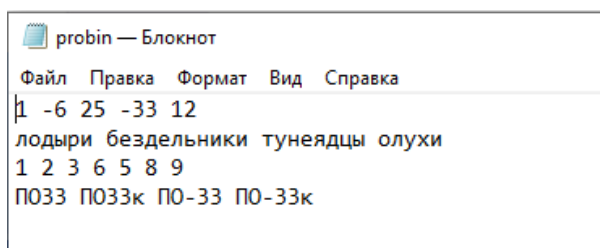
Выходные параметры: c:\\proba\\probout.txt.

Навигация внутри модуля: При нажатии на кнопку «Запись файла», выполняется метод `private void button2_Click(object sender, EventArgs e)`, в котором в операторе `use` создается файл c:\\proba\\probout.txt. После чего, считываются строки из ListBox1, в каждой подсчитывается количество пробелов `x`, затем строка преобразуется в строковый массив размером `x+1`. Далее выполняется проверка первого символа исходной строки, если этот символ число, суммируются элементы массива, и сумма приписывается к этой строке, затем полученная новая строка отображается в ListBox2 и записывается в файл c:\\proba\\probout.txt. Если первый символ исходной строки не является числом, суммируются символы элементов массива, результат приписывается к исходной строке, отображается в ListBox2 и записывается в файл результатов.

Сообщения об ошибках:

- обработка исключения отсутствия файла;
- обработка ошибок ввода данных, если введено не число;
- ошибки записей типа NULL.

## 2 Исходный файл

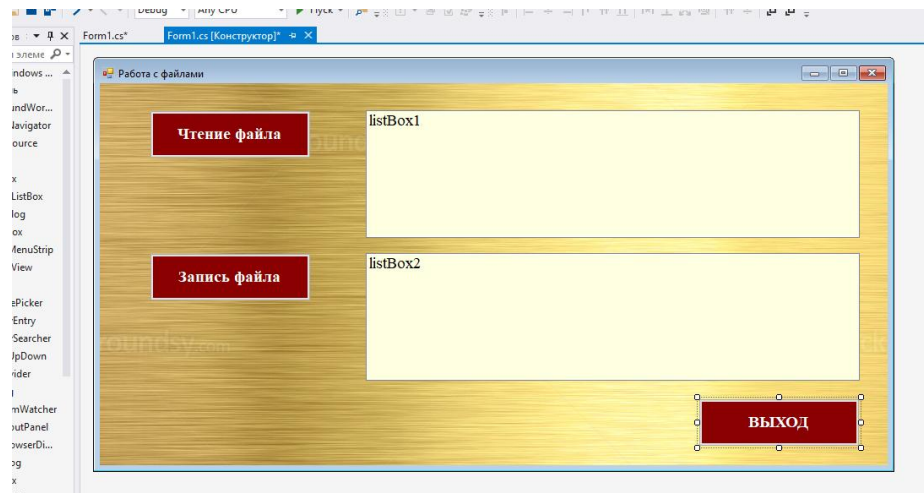


```
probin — Блокнот
Файл  Правка  Формат  Вид  Справка
1 -6 25 -33 12
лодыри бездельники тунейдцы олухи
1 2 3 6 5 8 9
ПО33 ПО33к ПО-33 ПО-33к
```

## 2 Файл результата

```
probout — Блокнот
Файл Правка Формат Вид Справка
1 -6 25 -33 12 Сумма = -1
лодыри бездельники туняядцы олухи Количество символов = 30
1 2 3 6 5 8 9 Сумма = 34
ПОЗЗ ПОЗЗк ПО-33 ПО-33к Количество символов = 20
```

## 4 Интерфейс формы в VisualStudio



## 3 Код программы

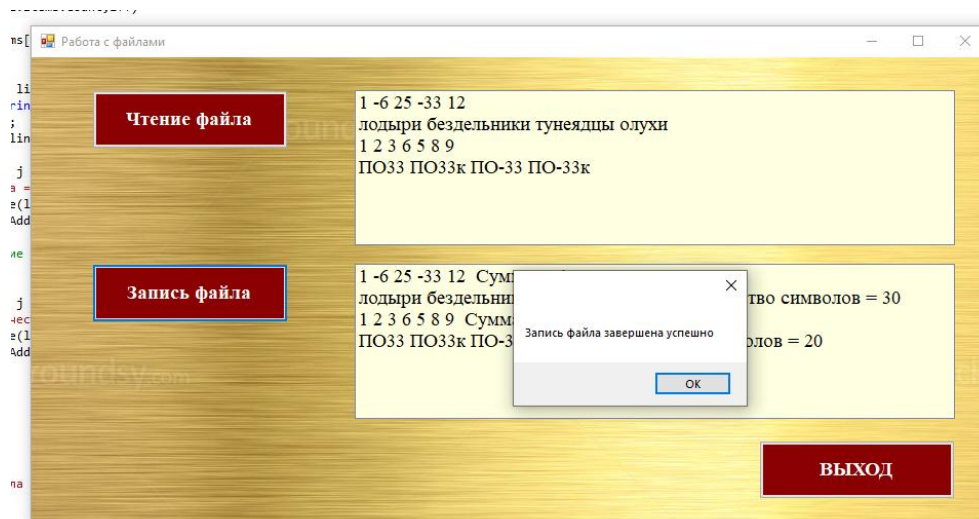
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;

namespace WindowsFormsApplication12
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

```
private void button3_Click(object sender, EventArgs e)
{
    Close();
}
...

```

#### 4 Результат работы программы:



#### Задание 5 (Пр.8)\*

##### 1 Исходные данные:

В файле содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от  $-10\,000$  до  $10\,000$  включительно. Определите и запишите в ответе сначала количество пар элементов последовательности, в которых хотя бы одно число делится на 3, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности. Например, для последовательности из пяти элементов: 6; 2; 9;  $-3$ ; 6 — ответ: 4 11.

##### Решение

...

#### Задание 5 (Пр.9)\*

##### Исходные данные:

Введите с клавиатуры строку произвольной длины, расположите символы в строке в порядке возрастания их кода и образуйте новую строку, в которой символы будут расположены в обратном порядке относительно отсортированной строки

### Решение:

На форме расположен текстовый блок для ввода адреса файла с данными, строка из блока является путем к файлу данных. Прежде, чем обрабатывать текст из файла, необходимо его прочесть.

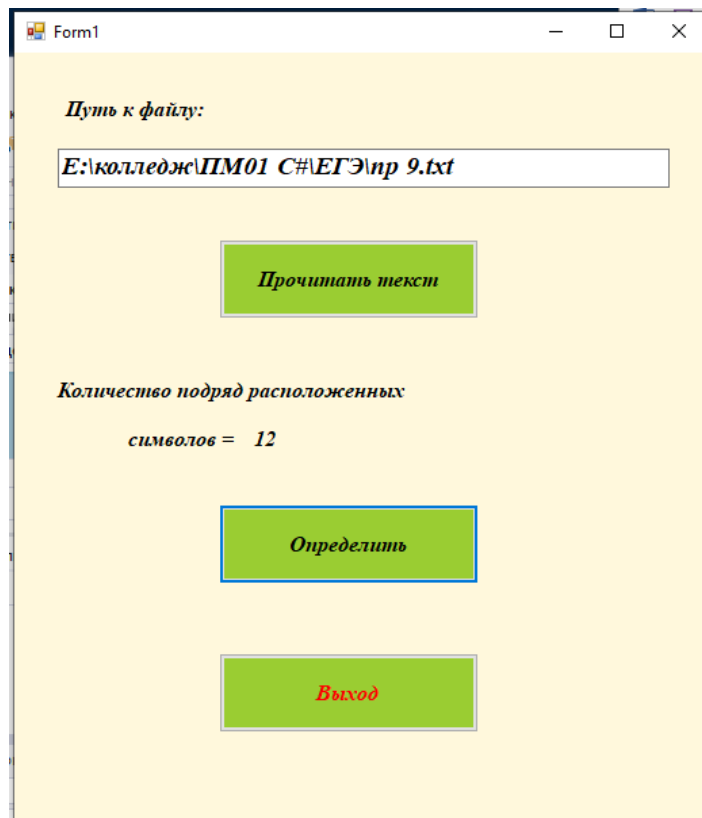
При нажатии кнопки «Прочитать текст» выполняется проверка наличия информации в TextBox1. В случае отсутствия адреса файла в диалоговом окне пользователю предлагается указать путь к файлу. При отсутствии файла обрабатывается исключение.

После открытия файла для чтения последовательность символов записывается в строковую переменную и сообщается пользователю об успешном чтении данных.

```
using System.Windows.Forms;
using System.IO;

namespace Pr9_4
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        public string line = null;
        private void button3_Click(object sender, EventArgs e)
        {
            try
            {
                if (textBox1.Text == "") MessageBox.Show("Укажите путь к файлу");
                else using (StreamReader r = File.OpenText(textBox1.Text))
                {
                    line = r.ReadToEnd();
                    MessageBox.Show("Файл прочитан успешно");
                }
            }
            catch (FileNotFoundException)
            {
                MessageBox.Show("Файл не найден");
            }
        }
        private void button2_Click(object sender, EventArgs e)
    }
}
```

Результат:



## Содержание отчета

- 1 Название работы
- 2 Цель работы
- 3 Технические средства обучения
- 4 Задания (условия задач)
- 5 Порядок выполнения работы
- 6 Ответы на контрольные вопросы
- 7 Вывод

## Варианты к практической работе:

1 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Проверить, содержится ли в файле заданное число. Найти сумму и количество чётных чисел файла. Результаты с соответствующими комментариями дописать в исходный файл.

2 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Извлечь квадратный корень из каждого числа в тех строках, все элементы которых имеют положительные значения. Подсчитать количество изменённых строк и результат с соответствующими комментариями поместить в конец исходного файла.

3 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Заменить в каждой строке элементы с наибольшим значением нулём. Создать файл, который будет содержать данные исходного файла, а затем данные преобразованного файла.

4 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Найти максимальные элементы в каждой строке исходного файла и поместить их в новый текстовый файл. Выдать оба файла.

5 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Подсчитать количество чисел в каждой строке. Найти номера строк, содержащих заданное число. Результаты с соответствующими комментариями поместить в новый текстовый файл.

6 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Найти среднее арифметическое чисел файла. Затем пустой строкой заменить строку с максимальной количеством чисел и удалить строку с минимальным количеством чисел в строке.

7 Даны два текстовых файла, состоящих из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Проверить поэлементно файлы на равенство. Если строки файлов не содержат поэлементно одинаковые числа, выдать номер строки и номер числа в строке, указывающих на первое несовпадение.

8 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Удалить из файла все числа, меньшие заданного числа. Если в текстовом файле таких чисел нет, оставить его без изменений.

9 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Из каждой строки файла удалить числа, предшествующие первому положительному числу. Если в строке нет положительных чисел, оставить её без изменений.

10 Дан текстовый файл, состоящий из нескольких строк. Строки содержат слова, разделённые пробелами. В каждой строке файла найти максимальную длину слова и подсчитать количество слов максимальной длины. За каждой строкой поместить строку с результатами обработки строки.

11 Дан текстовый файл, состоящий из нескольких строк. Строки содержат слова, разделённые пробелами. Подсчитать количество слов в каждой строке. Результаты с соответствующими комментариями поместить в виде строк в конце исходного файла.

12 Дан текстовый файл, состоящий из нескольких строк. Строки содержат слова, разделённые пробелами. В начало каждой строки, содержащей заданное слово, добавить слово 'ДОБАВЛЕНО'.

13 Дан текстовый файл, состоящий из нескольких строк. Строки содержат слова, разделённые пробелами. Удалить первое слово из каждой строки файла.

14 Дан текстовый файл, состоящий из нескольких строк. Строки содержат слова, разделённые пробелами. Перед каждым словом поместить номер слова в строке.

15 Дан текстовый файл. Структура строки текстового файла: фамилия и инициалы автора (20 позиций), название книги (30 позиций), издательство (10 позиций), год издания (4 позиции). Создать новый текстовый файл из строк, содержащих сведения о книгах, изданных за период с 1960 – 2000г.

16 Дан текстовый файл, состоящий из нескольких строк. Строки содержат слова, разделённые пробелами. В каждой строке файла найти минимальную длину слова и подсчитать количество слов минимальной длины. За каждой строкой поместить строку с результатами обработки строки.

17 Дан текстовый файл. Структура строки текстового файла: фамилия режиссера (20 позиций), название фильма (30 позиций), страна (10 позиций), год создания (4 позиции). Создать новый текстовый файл из строк, содержащих сведения о фильмах, созданных за период с 2018 – 2019г.

18 Дан текстовый файл, состоящий из нескольких строк. Строки содержат слова, разделённые пробелами. В каждой строке файла найти максимальную длину слова и в словах максимальной длины заменить все буквы на заглавные.

19 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Удалить из файла все числа, меньшие среднего арифметического этих чисел.

20 Дан текстовый файл, состоящий из нескольких строк. Строки содержат слова, разделённые пробелами. В каждой строке файла найти минимальную длину слова и в словах минимальной длины заменить все буквы на заглавные.

21 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Заменить в каждой строке элементы с наименьшим значением символом Z. Создать файл, который будет содержать данные исходного файла, а затем данные преобразованного файла.

22 Дан текстовый файл, состоящий из нескольких строк. Строки содержат слова, разделённые пробелами. В каждой строке файла найти минимальную длину слова и в словах минимальной длины заменить все буквы на 0.

23 Дан текстовый файл, состоящий из нескольких строк. Строки содержат слова, разделённые пробелами. В каждой строке файла найти минимальную длину слова и в словах минимальной длины заменить все буквы на ы.

## Контрольные вопросы

- 1 Что такое пространство имен System.IO?
- 2 Для чего используется класс File?
- 3 Для чего служит класс DriveInfo?
- 4 Для чего используется класс FileStream?
- 5 Для чего служит класс Directory, DirectoryInfo?
- 6 Какие пространства имен необходимо подключить в программе для работы с файлами?
- 7 Что такое поток данных?
- 8 Режимы открытия файлов и их особенности.
- 9 Для чего используется объект System.Text.Encoding?

## Используемая литература

1. Гниденко, И. Г. Технология разработки программного обеспечения: учеб. пособие для СПО / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — М.: Издательство Юрайт, 2017.
2. Шарп Джон Ш26 Microsoft Visual C#. Подробное руководство. 8-е изд. — СПб.: Питер, 2017.
3. Васильев А.Н. Программирование на C# для начинающих. Основные сведения. — Москва: Эксмо, 2018.
4. Васильев А.Н. Программирование на C# для начинающих. Особенности языка. — Москва: Эксмо, 2019.
5. <http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx>.