

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
КАФЕДРА ИИТ

**Лабораторная работа №2**

По дисциплине: «Современные платформы программирования»

**Выполнил:**

Студент 3 курса

группы ПО-8:

Печко В.И.

**Проверил:**

Крощенко А.А.

**Цель работы:** приобрести базовые навыки работы с файловой системой в C#.

## Вариант 18

**Задание 1.** Напишите программу, выполняющую чтение текстовых данных из файла и их последующую обработку:

**4)** Напишите программу, которая выводит слова, располагая их в порядке убывания частоты их появления. Перед каждым словом должно быть число его появлений.

### Код программы:

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;

class Program
{
    static void Main()
    {
        string text = File.ReadAllText("test.txt");

        Dictionary<string, int> wordDict = new Dictionary<string, int>();

        string[] words = text.Split(new char[] { ' ', '!', ',', ';', ':', '!', '?' },
StringSplitOptions.RemoveEmptyEntries);

        foreach (string word in words)
        {
            if (wordDict.ContainsKey(word))
            {
                wordDict[word]++;
            }
            else
            {
                wordDict[word] = 1;
            }
        }

        var sortedWords = wordDict.OrderByDescending(pair => pair.Value);

        foreach (var pair in sortedWords)
        {
            Console.WriteLine($"Word: {pair.Key}, Frequency: {pair.Value}");
        }
    }
}
```

## Спецификация ввода

test.txt

This is a test text. This text contains some words, and we will see how many times each word appears in this text. This is just a test, but it should provide enough data for the program to work properly.

## Спецификация вывода

Word: <слово из текста>, Frequency: <частота появления в тексте>

## Результат программы:

```
Word: This, Frequency: 3
Word: text, Frequency: 3
Word: is, Frequency: 2
Word: a, Frequency: 2
Word: test, Frequency: 2
Word: contains, Frequency: 1
Word: some, Frequency: 1
Word: words, Frequency: 1
Word: and, Frequency: 1
Word: we, Frequency: 1
Word: will, Frequency: 1
Word: see, Frequency: 1
Word: how, Frequency: 1
Word: many, Frequency: 1
Word: times, Frequency: 1
Word: each, Frequency: 1
Word: word, Frequency: 1
Word: appears, Frequency: 1
Word: in, Frequency: 1
Word: this, Frequency: 1
Word: just, Frequency: 1
Word: but, Frequency: 1
Word: it, Frequency: 1
Word: should, Frequency: 1
Word: provide, Frequency: 1
Word: enough, Frequency: 1
Word: data, Frequency: 1
Word: for, Frequency: 1
Word: the, Frequency: 1
Word: program, Frequency: 1
```

**Задание 2.** Написать консольную утилиту, обрабатывающую ввод пользователя и дополнительные ключи.

9) Утилита join объединяет строки двух упорядоченных текстовых файлов на основе наличия общего поля. По своему функционалу схоже с оператором JOIN, используемого в языке SQL для реляционных баз данных, но оперирует с текстовыми файлами.

Команда join принимает на входе два текстовых файла и некоторое число аргументов. Если не передаются никакие аргументы командной строки, то данная команда ищет пары строк в двух файлах, обладающие совпадающим первым полем

(последовательностью символов, отличных от пробела), и выводит строку, состоящую из первого поля и содержимого обоих строк.

Ключами -1 или -2 задаются номера сравниваемых полей для первого и второго файла, соответственно. Если в качестве одного из файлов указано – (но не обоих сразу!), то в этом случае вместо файла считывается стандартный ввод.

Формат использования:

`join [-1 номер_поля] [-2 номер_поля] файл1 файл2 [файл3]`

Параметры:

- - 1 `field_num` Задаёт номер поля в строке для первого файла, по которому будет выполняться соединение.
- - 2 `field_num` Задаёт номер поля в строке для второго файла, по которому будет выполняться соединение.

Аргументы:

- `файл1`, `файл2` – входные файлы
- `файл3` – выходной файл, куда записывается результат работы программы.

Примеры использования:

Пусть задан файл `1.txt` со следующим содержимым:

```
1 abc
2 lmn
3 pqr
```

и файл `2.txt` со следующим содержимым:

```
1 abc
3 lmn
9 orq
```

Тогда, выполнение команды `join 1.txt 2.txt` даст следующий результат:

```
1 abc abc
3 pqr lmn
```

Поскольку в обоих файлах есть строки, чьё первое поле совпадает (1, 3), выполнение команды

`join -1 2 -2 2 1.txt 2.txt` даст результат

```
abc 1 1
lmn 2 3
```

поскольку теперь сравнение выполняется по 2-му полю для первого и второго файла соответственно.

**Код программы:**

```
int col1 = 1;
int col2 = 1;
string fileName1, fileName2;
string? fileName3 = null;
```

```

int numOfArgs = 0;
if (args.Contains("-1"))
{
    int iIndex = Array.IndexOf(args, "-1");
    col1 = Convert.ToInt32(args[iIndex + 1]);
    numOfArgs += 2;
}

if (args.Contains("-2"))
{
    int lIndex = Array.IndexOf(args, "-2");
    col2 = Convert.ToInt32(args[lIndex + 1]);
    numOfArgs += 2;
}

if ((args.Length - numOfArgs) % 2 == 0)
{
    fileName1 = args[^2];
    fileName2 = args[^1];
}
else
{
    fileName1 = args[^3];
    fileName2 = args[^2];
    fileName3 = args[^1];
}

List<string[]> fileData1 = ReadData(fileName1);
List<string[]> fileData2 = ReadData(fileName2);
List<string[]> fileData3 = new List<string[]>();

foreach (var row1 in fileData1)
{
    foreach (var row2 in fileData2)
    {
        if (row1[col1] == row2[col2])
        {
            fileData3.Add(JoinData(row1, row2, col1, col2));
        }
    }
}

if (fileName3 != null)
{
    using (StreamWriter writer = new StreamWriter(fileName3))
    {
        foreach (var str in fileData3)
        {
            foreach (var s in str)
            {
                writer.Write($"{s} ");
            }
        }
    }
}

```

```

        writer.WriteLine();
    }
}
else
{
    foreach (var str in fileData3)
    {
        foreach (var s in str)
        {
            Console.Write($"{s} ");
        }
        Console.WriteLine();
    }
}

```

```

string[] JoinData(string[] str1, string[] str2, int col1, int col2)
{
    string[] strings = new string[str1.Length + str2.Length - 1];
    strings[0] = str1[col1];
    int index = 1;
    for (int i = 0; i < str1.Length; i++)
    {
        if (i == col1)
            continue;
        strings[index] = str1[i];
        index++;
    }

    for (int i = 0; i < str2.Length; i++)
    {
        if (i == col2)
            continue;
        strings[index] = str2[i];
        index++;
    }

    return strings;
}

```

```

List<string[]> ReadData(string fileName)
{
    List<string[]> fileData = new List<string[]>();
    using (StreamReader reader = new StreamReader(fileName))
    {
        string[] str = reader.ReadToEnd().Trim().Split("\n");
        foreach (var s in str)
        {
            string[] strings = s.Trim().Split(" ");
            fileData.Add(strings);
        }
    }
}

```

```
    }  
}  
  
return fileData;  
}
```

### Результат программы:

```
join -1 1 -2 1 f1.txt f2.txt  
abc 1 1  
lmn 2 3
```

**Вывод:** приобрели базовые навыки работы с файловой системой в C#.