

16 Работа с файлами

Задание 1. Ввести числа в файл. Найти минимум и посчитать количество положительных элементов в файле.

Листинг программы:

```
namespace Space
{
    class Program
    {
        static void Main(string[] args)
        {
            string filePath = @"E:\Учёба\Практика по пр\Les17\numbers.txt"; // путь
к файлу с числами
            int min = int.MaxValue; // начальное значение минимума -
максимальное значение типа int
            int positiveCount = 0; // счетчик положительных чисел

            // открыть файл и считать числа
            using (StreamReader reader = new StreamReader(filePath))
            {
                string line = reader.ReadLine();
                string[] numbers = line.Split(' ');

                foreach (string strNum in numbers)
                {
                    if (int.TryParse(strNum, out int num))
                    {
                        // если удалось прочесть число
                        if (num < min)
                        {
                            min = num; // обновить минимум, если найдено меньшее
число
                        }
                        if (num > 0)
                        {
                            positiveCount++; // увеличить счетчик положительных чисел
                        }
                    }
                }
            }
        }
    }
}
```

					УП 2-40 01 01.37ТП.227.23.16		
Изм.	Лист	№ док	Подпись	Дата			
Разраб.		Сорокина Е.А.			Работа с файлами	Лит	Лист
Проверил.		Новик А.И.					75
						Гродненский ГКТТид	
Н.контр.							
Утвердил.							

```

    }

    Console.WriteLine("Минимальное число: " + min);
    Console.WriteLine("Количество положительных чисел: " +
positiveCount);
    }
}
}

```

Таблица 16.1 – Входные и выходные данные

Входные данные	Выходные данные
1 8 6 4 7 2 3 12 44 0 -5 -1 8 1 4	-5 12

Анализ результатов:

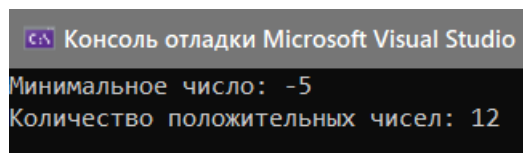


Рисунок 16.1 – Результат работы программы

Задание 2. Имеются два текстовых файла с одинаковым числом строк. Переписать с сохранением порядка следования строки первого файла во второй, а строки второго файла - в первый. Использовать вспомогательный файл.

Листинг программы:

```

namespace Space
{
    class Program
    {
        static void Main(string[] args)
        {
            string path1 = @"E:\Учёба\Практика по пр\Les17\1.txt";
            string path2 = @"E:\Учёба\Практика по пр\Les17\2.txt";
            string pathTemp1 = @"E:\Учёба\Практика по пр\Les17\temp1.txt";
            string pathTemp2 = @"E:\Учёба\Практика по пр\Les17\temp2.txt";

            // Открыть поток для чтения из первого файла
            using (StreamReader sr1 = new StreamReader(path1))
            {
                // Открыть поток для чтения из второго файла
                using (StreamReader sr2 = new StreamReader(path2))
                {

```

```

// Открыть поток для записи во временный файл 1
using (StreamWriter swTemp1 = new StreamWriter(pathTemp1))
{
    // Открыть поток для записи во временный файл 2
    using (StreamWriter swTemp2 = new StreamWriter(pathTemp2))
    {
        // Переписать строки первого файла во второй, а строки
второго файла - в первый
        string line1, line2;
        while ((line1 = sr1.ReadLine()) != null && (line2 =
sr2.ReadLine()) != null)
        {
            swTemp1.WriteLine(line2);
            swTemp2.WriteLine(line1);
        }
    }
}

// Удалить исходные файлы и переименовать временные файлы в
исходные
File.Delete(path1);
File.Delete(path2);
File.Move(pathTemp1, path1);
File.Move(pathTemp2, path2);

Console.WriteLine("Файлы успешно переписаны!");
}
}
}

```

Анализ результатов:

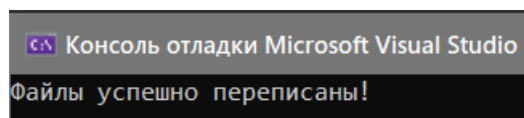


Рисунок 16.2 – Результат работы программы

Изм.	Лист	№ док	Подпись	Дата

УП 2-40 01 01.37ТП.227.23.16