

Программирование на языке C#

Файлы

***Файл* –**

именованный
набор байтов, который
может быть сохранен на
некотором накопителе,
например файл.txt.

Под работой с файлами подразумевается:

1. Создание файлов;
2. Чтение данных;
3. Запись данных.

В Си-шарп есть пространство имен **System.IO**, в котором реализованы все необходимые нам классы для работы с файлами. Чтобы подключить это пространство имен, необходимо в самом начале программы добавить строку **using System.IO**.

Для использования кодировок еще добавим пространство **using System.Text**;

Режимы открытия FileMode:

- ***Append*** – открывает файл (если существует) и переводит указатель в конец файла (**данные будут дописываться в конец**), или создает новый файл.
- ***Create*** - создает новый файл(если существует – заменяет)
- ***Open*** - открывает файл

Для работы с файлами нам
понадобится класс **FileSteam** .

FileSteam - класс который
позволяет выполнять операции
чтения/записи в файл.

```
static void Main(string[] args)
{
    FileStream file1 = new FileStream("d:\\file1.txt",
    FileMode.Create); //создание нового файла

    FileStream file2 = new FileStream("d:\\file2.txt",
    FileMode.Open); //открытие существующего файла

    FileStream file3 = new FileStream("d:\\file3.txt",
    FileMode.Append); //открытие файла для дозаписи
    в конец файла
}
```

Для чтения данных нам понадобится класс **StreamReader**. В нем реализовано множество методов для удобного считывания данных.

```
static void Main(string[] args)
```

```
{
```

```
    FileStream file1 = new FileStream("d:\\test.txt", FileMode.Open);  
    //обратились к файлу
```

```
    StreamReader reader = new StreamReader(file1); //создаем объект  
    на файл
```

```
    Console.WriteLine(reader.ReadToEnd ()); //выводим на экран все  
    строки которые находятся в файле
```

```
    reader.Close(); //закрываем поток
```

```
}
```


Для записи данных используется класс **StreamWriter**.

```
static void Main(string[] args)
{
    FileStream file1 = new FileStream("d:\\test.txt", FileMode.Create);

    StreamWriter writer = new StreamWriter(file1);

    writer.WriteLine("текст"); //записываем в файл

    writer.Close(); //Не закрыв, в файл ничего не запишется
}
```

Задача. Создать файл с именем data.txt и заполнить его N случайными числами.

```
StreamWriter sw = File.CreateText("data.txt");
```

```
Console.Write("Введи n= ");  
int n = Convert.ToInt32(Console.ReadLine());  
int a;  
for (int i = 0; i < n; i++)
```

Создаем элемент класса StreamWriter и связываем его с файлом на диске

```
{
```

```
    a = Random.Next(0, 100);
```

Закрываем файл

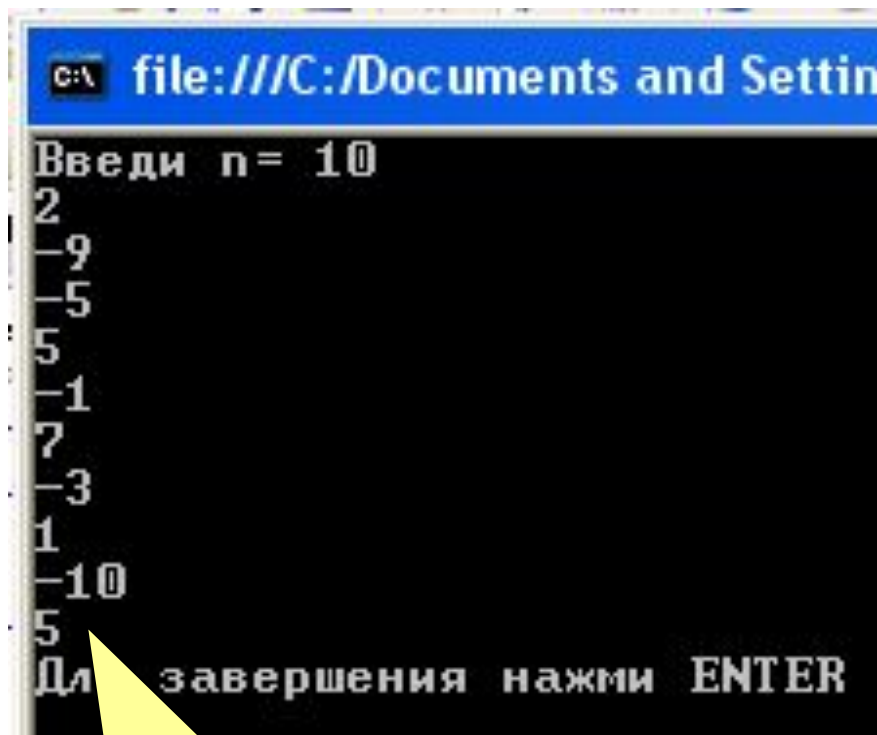
```
    sw.WriteLine(a.ToString());
```

```
    sw.WriteLine(a.ToString());
```

```
}
```

Записываем в файл очередную строку

```
sw.Close();
```



file:///C:/Documents and Settings

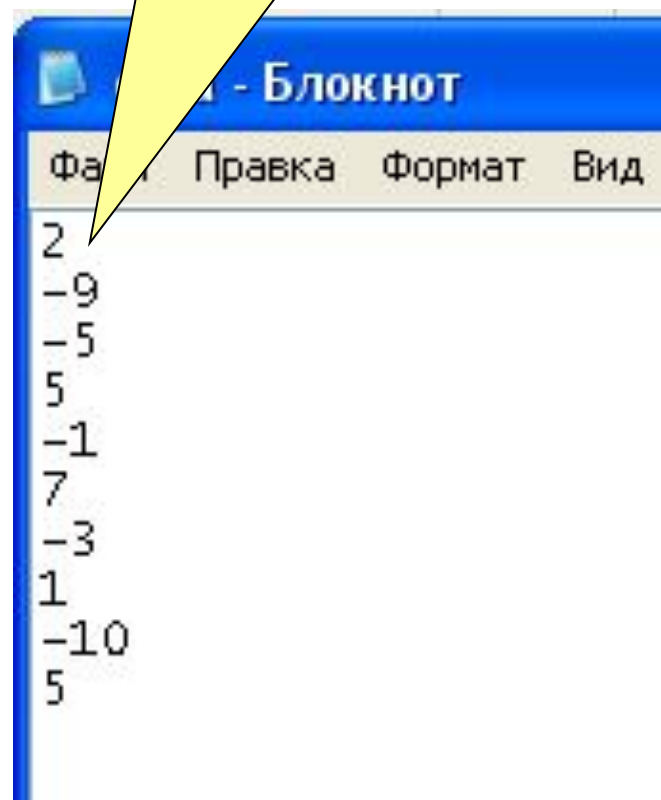
Введи n = 10

2
-9
-5
5
-1
7
-3
1
-10
5

Для завершения нажми ENTER

**Файл с данными,
выведенный на экран**

**Файл с данными,
открытый Блокнотом**

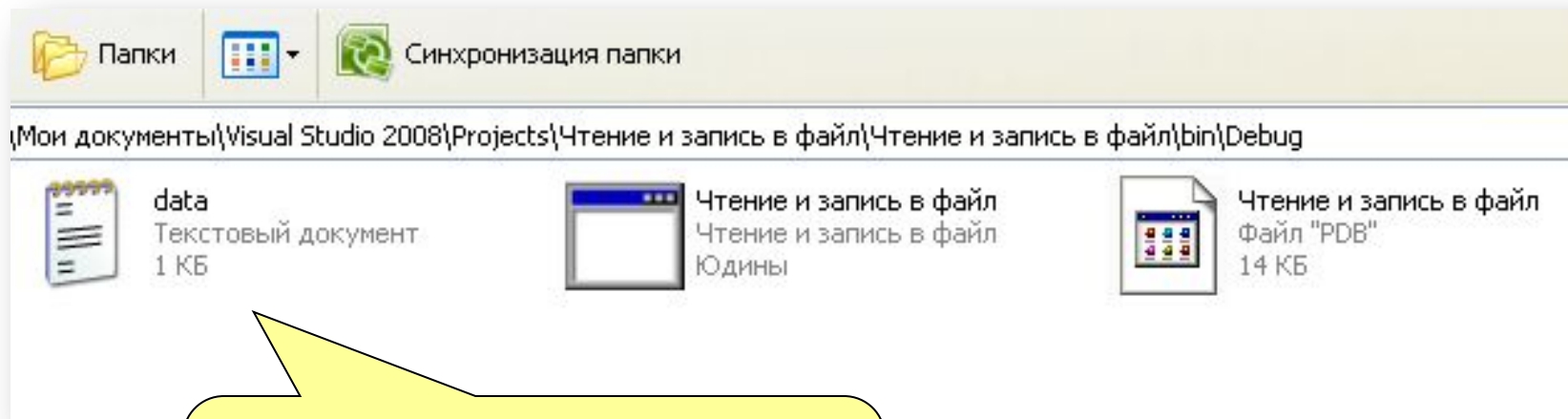


Блокнот

Файл Правка Формат Вид

2
-9
-5
5
-1
7
-3
1
-10
5

Где находится файл с данными

[Содержание](#)

**Файл данных и
исполняемый файл в
проводнике**

**\\Мои документы\\Visual Studio 2008\\
Projects\\Имя_проекта\\Имя_проекта\\bin\\Debug**

**Примерный путь к
файлу с данными**

Задача 1. Считать из файла информацию и вывести на экран.

Задача 2. Сохранить в файл любую информацию.

Задача 3. Создайте файл numbers.txt и запишите в него натуральные числа от 1 до 500 через запятую.

Задача 4. Дан массив строк: "red", "green", "black", "white", "blue". Запишите в файл элементы массива построчно (каждый элемент в новой строке).

Напишите программу, которая дописывает в находящийся на диске A: файл phone.txt имя, фамилию и номер телефона, например, вашего товарища. Если файла на диске нет, то программа должна создать его. В файле каждый элемент данных (имя, фамилия, телефон) должен находиться в отдельной строке. Рекомендуемый вид экрана во время работы программы приведен ниже.

Добавление в телефонный справочник

Фамилия -> Сидоров

Имя -> Вася

Телефон -> 234-84-37

Информация добавлена.

Для завершения работы нажмите <Enter>

Составим, по которой будет рассчитана и записана в файл таблица квадратных корней для целых чисел от 1 до 10. Для контроля эта же таблица выводится на экран.

x	sqrt (x)
1.000000	1.000000
2.000000	1.414214
3.000000	1.732051
4.000000	2.000000
5.000000	2.236068
6.000000	2.449490
7.000000	2.645751
8.000000	2.828427
9.000000	3.000000
10.000000	3.162278

Создать файл, с произвольным текстом, который будет обязательно содержать: буквы, цифры символы. Посчитать сколько цифр от 0 до 9 встречается в файле. На экран вывести сообщение о количество цифр и сам текст файла.

В некотором вузе абитуриенты проходят предварительное тестирование, по результатам которого могут быть допущены к сдаче вступительных экзаменов в первом потоке. Тестирование проводится по двум предметам, по каждому предмету абитуриент может набрать от 0 до 100 баллов. При этом к сдаче экзаменов в первом потоке допускаются абитуриенты, набравшие по результатам тестирования не менее 30 баллов по каждому из двух предметов. На вход программы подаются сведения о результатах предварительного тестирования.

В первой строке вводится количество абитуриентов, принимавших участие в тестировании, N.

Далее следуют N строк, имеющих следующий формат: <Фамилия> <Имя> <Баллы>

Здесь:

- <Фамилия> – строка, состоящая не более чем из 20 символов;
- <Имя> – строка, состоящая не более чем из 15 символов;
- <Баллы> – строка, содержащая два целых числа, соответствующих баллам, полученным на тестировании по каждому из двух предметов.

При этом <Фамилия> и <Имя>, <Имя> и <Баллы> могут быть разделены пробелами и символами табуляции (разным количеством).

Примеры входных строк:

Ветров Роман	68	59
Анисимова Екатерина	64	88

Напишите программу, которая будет выводить на экран и в файл фамилии и имена абитуриентов, также их баллы, потерпевших неудачу, то есть не допущенных к сдаче экзаменов в первом потоке.