**16 Работа с файлами**

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

Листинг программы:

internal class Program

{

static void Main(string[] args)

{

var directory = Directory.CreateDirectory("D:\\Folder16");

var file = new FileInfo($"{directory.FullName}\\File1.txt");

if (!file.Exists)

{

file.Create();

}

FillFile(filePath: file.FullName);

int count = default(int);

using (StreamReader reader = new StreamReader(file.FullName, Encoding.UTF8))

{

string s;

while ((s = reader.ReadLine()) != null)

{

count++;

}

}

try

{

var numbers = ReadFile(filePath: file.FullName);

var minNumber = numbers.Min();

Console.WriteLine("Минимальное число в файле: " + minNumber);

var countPositiveNumbers = numbers.Count(number => number > 0);

Console.WriteLine("Количество положительных чисел в файле: " + countPositiveNumbers);

}

catch (FormatException ex)

{

Console.WriteLine(ex.Message);

}

Console.ReadLine();

}

private static void FillFile(string filePath, int minValue = -5, int maxValue = 6, int countNumbers = 10)

{

var random = new Random();

using (var writer = new StreamWriter(filePath, append: false, encoding: Encoding.UTF8))

{

while (countNumbers > 0)

{

writer.WriteLine(random.Next(minValue, maxValue));

countNumbers--;

}

}

}

private static IEnumerable<int> ReadFile(string filePath)

{

using (var reader = new StreamReader(filePath, Encoding.UTF8))

{

var input = String.Empty;

while ((input = reader.ReadLine()) != null)

{

yield return int.Parse(input);

}

}

}

}

Таблица 16.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Минимальное число в файле: -5  Количество положительных чисел в файле: 5 |

Анализ результатов:



Рисунок 16.1 – Результат работы программы

Задание 2. С помощью метода CreateDirectory() создайте папку “New\_folder”.

Листинг программы:

class Program

{

static void Main(string[] args)

{

var directory = Directory.CreateDirectory("E:\\RaidenShogun");

}

}

}

Анализ результатов:



Рисунок 16.2 – Результат работы программы

Источник: собственная разработка

Задание 3. Создать текстовый файл, в который записать 5 строк различной длины.

Для данного файла выполнить следующие действия:

a) вывести весь файл на экран;

b) подсчитать количество строк;

c) подсчитать количество символов в каждой строке;

d) удалить последнюю строку из файла, результат записать в новый

файл;

e) вывести на экран строки с s1 по s2;

f) найти длину самой длинной строки и вывести ее на экран;

g) вывести на экран все строки начинающиеся с заданной буквы;

h) переписать его строки в другой файл, порядок строк во втором

файле должен быть обратным по отношению к порядку строк в заданном

файле.

Листинг программы:

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Вывод текста из файла: \n------------------------------------ ");

PrintTXT.Print();

Console.WriteLine("==============================================================");

var filePath = @"E:\Practice16.txt";

var filePath1 = @"E:\Practice16Dop.txt";

int count = File.ReadAllLines(filePath).Length;

Console.WriteLine($"Количество строк: {count}");

var numbers = SymbolCount.SymCount(filePath);

int k = 1;

foreach (var num in numbers)

{

Console.WriteLine($"Колличесвто символов в {k}-ой строке: { num.ToString()}");

k++;

}

Console.WriteLine("Удаление последней строки зарисанно в новый файл ");

DeletStringLast.DelStrL(filePath, filePath1);

S1toS2.GetStringsS1ToS2(filePath, 1, 3);

LongestString.GetLongestString(filePath);

GetStringFirstLetter.GetLineWithFirstLetter(filePath, 'п');

ReversOutput.PrintToFileReverseLinesInfile(filePath);

Console.ReadLine();

}

}

Анализ результатов:

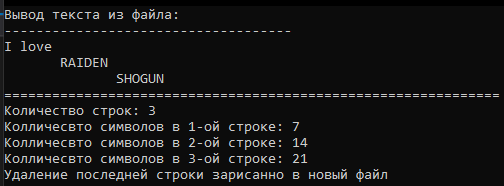


Рисунок 16.3 – Результат работы программы

Источник: собственная разработка