**ПРАКТИЧЕСКОЕ ЗАДАНИЕ 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: 4 |

Анализ результатов:



Рисунок 16.1 – Результат работы программы

Источник: собственная разработка

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:

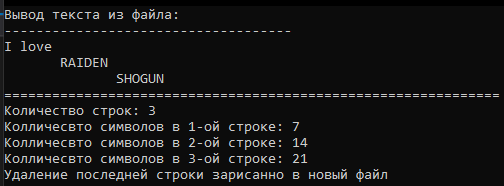


Рисунок 16.3 – Результат работы программы

Источник: собственная разработка

Задание 4. Дана последовательность из n вещественных чисел. Записать все числа в файл, затем вывести на экран все компоненты файла, меньшие заданного числа. Вывести на экран все положительные компоненты файла. Вычислить среднее арифметическое компонентов файла, стоящих на четных позициях.

Листинг программы:

class Program

{

static void ShowMenu()

{

Console.WriteLine();

Console.WriteLine("Меню:");

Console.WriteLine("0 - Выйти из программы");

Console.WriteLine("1 - Вывести на экран все компоненты файла,меньшие заданного числа.");

Console.WriteLine("2 - Вывести на экран все положительные компоненты файла");

Console.WriteLine("3 - Вычислить среднее арифметическое компонентов файла, стоящих\nна чётных позициях");

Console.WriteLine();

} // ShowMenu()

static void Task1(BinaryReader fromFile) // Вывести на экран все компоненты файла, меньшие заданного числа

{

double userNum; // Опорное число

double item; // Элемент файла

// Получаем опорное число

do // 1

{

Console.Write("Введите опорное число: ");

try

{

userNum = double.Parse(Console.ReadLine());

break;

}

catch (FormatException)

{

Console.WriteLine("Ошибка! Неверный формат данных.");

}

} while (true); // do 1

// Читаем файл, выполняем задание

Console.WriteLine("Числа, меньшие заданного вами:");

try // 1

{

fromFile.BaseStream.Seek(0, SeekOrigin.Begin); // Указатель файлового потока - в начало.

while (fromFile.BaseStream.Position < fromFile.BaseStream.Length)

{

item = fromFile.ReadDouble();

if (item < userNum) Console.WriteLine(item);

}

}

catch (IOException A)

{

Console.WriteLine("Внимание! Возникло исключение при чтении файла.\n" + A.Message);

}

// end try 1

} // Task1()

static void Task2(BinaryReader fromFile) // Вывести на экран все положительные компоненты файла

{

double item; // Элемент файла

// Читаем файл, выполняем задание

Console.WriteLine("Положительные компоненты файла:");

try // 1

{

fromFile.BaseStream.Seek(0, SeekOrigin.Begin); // Указатель файлового потока - в начало.

while (fromFile.BaseStream.Position < fromFile.BaseStream.Length)

{

item = fromFile.ReadDouble();

if (item > 0.0) Console.WriteLine(item);

}

}

catch (IOException A)

{

Console.WriteLine("Внимание! Возникло исключение при чтении файла.\n" + A.Message);

}

// end try 1

} // Task2()

static void Task3(BinaryReader fromFile) // Вычислить среднее арифметическое компонентов файла, стоящих на чётных позициях

{

int i = 0;

double summ = 0.0;

double item; // Элемент файла

// Читаем файл и, собственно, выполняем задание

Console.WriteLine("Среднее арифметическое компонентов файла, стоящих на чётных позициях:");

try // 1

{

fromFile.BaseStream.Seek(0, SeekOrigin.Begin); // Указатель файлового потока - в начало.

while (fromFile.BaseStream.Position < fromFile.BaseStream.Length)

{

item = fromFile.ReadDouble();

summ += item; // Суммируем

i++; // Подсчитываем количество нужных нам элементов

fromFile.BaseStream.Position += 8; // Одно число, типа double занимает 8 байт.

}

}

catch (IOException A)

{

Console.WriteLine("Внимание! Возникло исключение при чтении файла.\n" + A.Message);

}

// end try 1

Console.WriteLine("{0:F3}", summ / i);

} // Task3()

static void Main(string[] args)

{

int i = 0; // Переменная цикла

int seqLength = 0; // Длина последовательности

double currNum; // Очередное число

double[] arr; // Ссылка на массив вещественных чисел (пока пустая)

string fileName = "output.dat"; // Имя файла

FileStream fs; // Базовый файловый поток

BinaryWriter dataOut; // Бинарный поток на запись

BinaryReader dataIn; // Бинарный поток на чтение

Console.WriteLine("Ввод последовательности из n вещественных чисел:");

do // 1 Получение длины последовательности

{

Console.Write("Укажите длину последовательности: ");

try

{

seqLength = int.Parse(Console.ReadLine());

if (seqLength <= 0) throw new FormatException();

}

catch (FormatException)

{

Console.WriteLine("Ошибка! Неверный формат данных.");

}

} while (seqLength <= 0); // do 1

arr = new double[seqLength]; // Подключаем ссылку к массиву из заданного количества элементов

// Заполняем массив

do // 2

{

Console.Write("[{0}]: ", i);

try

{

currNum = double.Parse(Console.ReadLine());

arr[i] = currNum;

i++;

}

catch (FormatException)

{

Console.WriteLine("Ошибка! Неверный формат данных.");

}

} while (i < arr.Length); // do 2

Console.WriteLine("Полученная последовательность:");

for (i = 0; i < arr.Length; i++)

{

Console.WriteLine("{0}", arr[i]);

try // 1

{

// Инициируем потоки

fs = new FileStream(fileName, FileMode.Create, FileAccess.Write);

dataOut = new BinaryWriter(fs);

}

catch (IOException A)

{

Console.WriteLine("Внимание! Возникло исключение при открытии файла {0}.", fileName);

Console.WriteLine(A.Message);

Console.WriteLine("Аварийное завершение программы.");

return;

}

// end try 1

// Пишем данные

try // 2

{

for (i = 0; i < arr.Length; i++)

{

dataOut.Write(arr[i]);

}

Console.WriteLine("Файл записан.");

}

catch (IOException B)

{

Console.WriteLine("Внимание! Возникло исключение при записи файла.\n" + B.Message);

}

finally

{

dataOut.Close(); // Закрываем пишущий поток

}

// end try 2

// Открываем файл для чтения

try // 3

{

fs = new FileStream(fileName, FileMode.Open, FileAccess.Read);

dataIn = new BinaryReader(fs);

}

catch (IOException C)

{

Console.WriteLine("Внимание! Возникло исключение при открытии файла {0}.", fileName);

Console.WriteLine(C.Message);

Console.WriteLine("Аварийное завершение программы.");

return;

}

// end try 3

ShowMenu(); // Показываем меню. Его можно вставить внуть цикла do\_3. Тогда оно будет выводиться после выполнения каждого задания.

bool IsExit; // Флаг завершения работы программы

byte itemMenu;

try // 4

{

do // 3

{

IsExit = false;

try // 5

{

Console.WriteLine("Введите номер пункта меню");

itemMenu = byte.Parse(Console.ReadLine());

switch (itemMenu) // 1

{

case 0:

IsExit = true;

break;

case 1:

Task1(dataIn);

break;

case 2:

Task2(dataIn);

break;

case 3:

Task3(dataIn);

break;

default:

Console.WriteLine("Нет такого пункта меню.");

break;

} // switch 1

}

catch (FormatException)

{

Console.WriteLine("Ошибка! Неверный формат при вводе пункта меню.");

}

// end try 5

} while (!IsExit); // do 3

}

finally

{

dataIn.Close();

}

// end try 4

Console.WriteLine("Конец программы");

} // Main()

}

}

Таблица 16.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Укажите длину последовательности: 3  [0]: 2  [1]: 3  [2]: 15 | Меню:  0 - Выйти из программы  1 - Вывести на экран все компоненты файла,меньшие заданного числа.  2 - Вывести на экран все положительные компоненты файла  3 - Вычислить среднее арифметическое компонентов файла, стоящих  на чётных позициях  Введите номер пункта меню  1  Введите опорное число: 2  Числа, меньшие заданного вами:  Введите номер пункта меню  2  Положительные компоненты файла:  2  3  15  Введите номер пункта меню  3  Среднее арифметическое компонентов файла, стоящих на чётных позициях:  8,500 |

Анализ результатов:

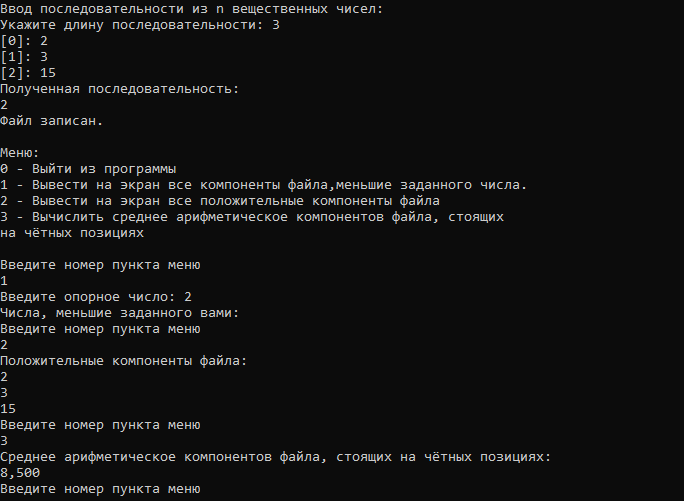


Рисунок 16.4 – Результат работы программы

Источник: собственная разработка

Задание 5. Дан текстовый файл. Переписать в новый файл все его строки, вставив в конец каждой строки количество символов в ней.

Листинг программы:

class Program

{

static void Main(string[] args)

{

using (StreamWriter sw = File.CreateText("input.txt"))

{

for (int i = 0; i < 5; i++)

{

sw.WriteLine(Console.ReadLine());

}

}

using (StreamReader sr = File.OpenText("input.txt"))

using (StreamWriter sw = new StreamWriter("output.txt"))

{

for (int i = 0; i < System.IO.File.ReadAllLines("input.txt").Length; i++)

{

int count = 0;

string s = sr.ReadLine();

char[] chars = s.ToCharArray();

foreach (char ch in chars)

{

if ((ch == '.') || (ch == '!') || (ch == '?')) { }

else

{ count++; }

}

sw.WriteLine(s + " " + count);

}

}

}

}

Таблица 16.5 – Входные и выходные данные

|  |
| --- |
| Входные данные |
| Привет, как дела?  Хы  Хы  ХЫ |

Анализ результатов:

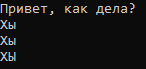


Рисунок 16.5 – Результат работы программы

Источник: собственная разработка