Работа с файлами

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:

using System;

using System.IO;

using System.Linq;

namespace FileOperations

{

class Program

{

static void Main(string[] args)

{

string fileName = "file.txt";

string newFileName = "newFile.txt";

string letter = "a";

int s1 = 2;

int s2 = 4;

// Create a file and write 5 lines of different lengths

using (StreamWriter sw = new StreamWriter(fileName))

{

sw.WriteLine("This is the first line");

sw.WriteLine("This is the second line");

sw.WriteLine("This is the third line with more characters");

sw.WriteLine("This is the fourth line with even more characters");

sw.WriteLine("This is the fifth line with the most characters of all");

}

// Read the file and perform the required actions

using (StreamReader sr = new StreamReader(fileName))

{

// a) Print the entire file to the console

Console.WriteLine(sr.ReadToEnd());

// b) Count the number of lines

sr.BaseStream.Seek(0, SeekOrigin.Begin);

int lineCount = 0;

while (sr.ReadLine() != null)

{

lineCount++;

}

Console.WriteLine($"Number of lines: {lineCount}");

// c) Count the number of characters in each line

sr.BaseStream.Seek(0, SeekOrigin.Begin);

string line;

while ((line = sr.ReadLine()) != null)

{

Console.WriteLine($"Number of characters in line {lineCount}: {line.Length}");

lineCount--;

}

// d) Delete the last line and write it to a new file

sr.BaseStream.Seek(0, SeekOrigin.Begin);

using (StreamWriter sw = new StreamWriter(newFileName))

{

string lastLine = "";

while ((line = sr.ReadLine()) != null)

{

if (sr.Peek() == -1)

{

lastLine = line;

break;

}

sw.WriteLine(line);

}

Console.WriteLine($"Deleted line: {lastLine}");

}

// e) Print lines s1 to s2 to the console

sr.BaseStream.Seek(0, SeekOrigin.Begin);

lineCount = 0;

while ((line = sr.ReadLine()) != null)

{

lineCount++;

if (lineCount >= s1 && lineCount <= s2)

{

Console.WriteLine(line);

}

}

// f) Find the length of the longest line and print it to the console

sr.BaseStream.Seek(0, SeekOrigin.Begin);

int longestLineLength = 0;

while ((line = sr.ReadLine()) != null)

{

if (line.Length > longestLineLength)

{

longestLineLength = line.Length;

}

}

Console.WriteLine($"Length of longest line: {longestLineLength}");

// g) Print all lines starting with the specified letter to the console

sr.BaseStream.Seek(0, SeekOrigin.Begin);

while ((line = sr.ReadLine()) != null)

{

if (line.StartsWith(letter))

{

Console.WriteLine(line);

}

}

// h) Write the lines to a new file in reverse order

sr.BaseStream.Seek(0, SeekOrigin.Begin);

using (StreamWriter sw = new StreamWriter(newFileName))

{

var lines = sr.ReadToEnd().Split(Environment.NewLine).Reverse();

foreach (var l in lines)

{

sw.WriteLine(l);

}

}

}

}

}

}

Анализ программы:

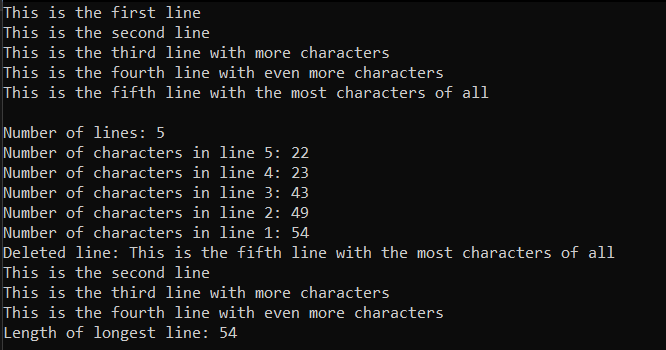


Рисунок 1.3 - Анализ программы

Источник: собственная разработка

Задание №4. Имеются два текстовых файла с одинаковым числом строк. Выяснить совпадают ли их строки. Если нет, то получить номер первой строки, в которой эти файлы отличаются друг от друга.

Листинг программы:

using System;

using System.IO;

namespace FileOperations

{

class Program

{

static void Main(string[] args)

{

string sourceFilePath = @"D:\PRAKTIKA\PRAKTIKA\_CSharp\16\ZADACHA1.txt";

string destinationFilePath = @"D:\PRAKTIKA\PRAKTIKA\_CSharp\16\ZADACHA1.txt";

try

{

string[] lines = File.ReadAllLines(sourceFilePath);

for (int i = 0; i < lines.Length; i++)

{

lines[i] = lines[i].Replace('0', '2').Replace('1', '0').Replace('2', '1');

}

File.WriteAllLines(destinationFilePath, lines);

Console.WriteLine("Good");

}

catch (Exception e)

{

Console.WriteLine($"Error: {e.Message}");

}

}

}

}

Анализ программы:



Рисунок 1.4 - Анализ программы

Источник: собственная разработка