**16 Работа с файлами**

Задание 1. Имеется текстовый файл. Переписать в другой файл все его строки с заменой в них символа 0 на символ 1 и наоборот.

namespace task1

class Program

{

static void Main(string[] args)

{

string inputFile = "input.txt"; // путь к входному файлу

string outputFile = "output.txt"; // путь к выходному файлу

// Чтение строк из входного файла и замена символов

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

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

{

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

}

// Запись измененных строк в выходной файл

File.WriteAllLines(outputFile, lines);

Console.WriteLine("Готово!");

Console.ReadKey();

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Готово! |

Анализ результатов:

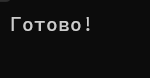


Рисунок 1.1 –работы программы

Задание 2. 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.

namespace task3

{

using System;

using System.IO;

using System.Linq;

class Program

{

static void Main(string[] args)

{

string filePath = @"input.txt";

string newFilePath = @"output.txt";

// a) вывести весь файл на экран;

Console.WriteLine(File.ReadAllText(filePath));

// b) подсчитать количество строк;

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

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

// c) подсчитать количество символов в каждой строке;

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

foreach (string line in lines)

{

Console.WriteLine($"Количество символов в строке \"{line}\": {line.Length}");

}

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

string[] newLines = lines.Take(lines.Length - 1).ToArray();

File.WriteAllLines(newFilePath, newLines);

// e) вывести на экран строки с s1 по s2;

int s1 = 2;

int s2 = 4;

for (int i = s1 - 1; i < s2; i++)

{

Console.WriteLine(lines[i]);

}

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

int maxLength = lines.Max(line => line.Length);

Console.WriteLine($"Самая длинная строка имеет длину {maxLength}");

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

char letter = 'A';

foreach (string line in lines)

{

if (line.StartsWith(letter.ToString()))

{

Console.WriteLine(line);

}

}

// h) переписать его строки в другой файл, порядок строк во втором файле должен быть обратным по отношению к порядку строк в заданном файле.

string[] reversedLines = lines.Reverse().ToArray();

File.WriteAllLines(newFilePath, reversedLines);

}

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| ghbd  dfgfdgf  fh  dfhdfhdhdd  fhh | Количество строк: 5  Количество символов в строке "ghbd": 4  Количество символов в строке "dfgfdgf": 7  Количество символов в строке "fh": 2  Количество символов в строке "dfhdfhdhdd": 10  Количество символов в строке "fhh": 3  dfgfdgf  fh  dfhdfhdhdd  Самая длинная строка имеет длину 10 |

Анализ результатов:

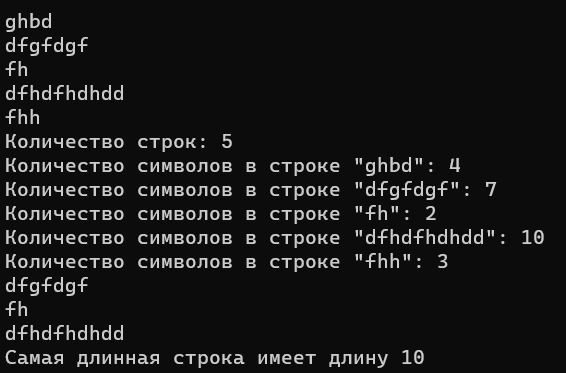
**

Рисунок 1.2 – Результат работы программы

работы программы