**16.** **Работа с файлами**

Задание №1

Дан файл f, компонентами которого являются натуральные числа. Получить в файле g все компоненты файла f, делящиеся на 3.

Листинг программы:

string pathF = @"D:\PracticCSharp\Prac17\z1\Files\f.txt";

List<int> nums = await GetNumsDiv3InFile(pathF);

string pathG = @"D:\PracticCSharp\Prac17\z1\Files\g.txt";

await WriteNumToFile(pathG, nums);

async Task<List<int>> GetNumsDiv3InFile(string path)

{

using (StreamReader reader = new StreamReader(path))

{

string? line;

List<int> nums = new List<int>();

while ((line = await reader.ReadLineAsync()) != null)

{

string[] elements = line.Split(' ');

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

{

if (int.TryParse(elements[i], out int num))

{

if (num % 3 == 0)

{

nums.Add(num);

}

}

}

}

return nums;

}

}

async Task WriteNumToFile(string path,List<int> nums)

{

string line = "";

foreach(int num in nums)

{

line += num + " ";

}

using (StreamWriter writer = new StreamWriter(path, false))

{

await writer.WriteLineAsync(line);

}

}

Таблица 16.1 – Входные и выходные данные программы задание №1

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| В исходном фале записано: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 | Выходном файле записано: 3 6 9 12 15 |

Анализ результатов:



Рисунок 16.1 – Результат выполнения программы задание №1

Задание №2

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

Листинг программы:

Directory.CreateDirectory(@"D:\PracticCSharp\Prac17\z2\DirectoryNew\New\_folder\");

Таблица 16.2 – Входные и выходные данные программы задание №2

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Создана папка New\_oflder |

Анализ результатов:



Рисунок 16.2 – Результат выполнения программы задание №2

Задание №3

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

Для данного файла выполнить следующие действия: вывести весь файл на экран; подсчитать количество строк; подсчитать количество символов в каждой строке; удалить последнюю строку из файла, результат записать в новый файл; вывести на экран строки с s1 по s2; найти длину самой длинной строки и вывести ее на экран; вывести на экран все строки начинающиеся с заданной буквы; переписать его строки в другой файл, порядок строк во втором файле должен быть обратным по отношению к порядку строк в заданном файле.

Листинг программы:

string pathF1 = @"D:\PracticCSharp\Prac17\z3\Files\TextFile1.txt";

List<string> linesF1 = await GetTextFile(pathF1);

Console.WriteLine("Текст из 1 файла:");

WriteList(linesF1);

Console.WriteLine();

Console.WriteLine($"Кол-во строк в 1 файле: {linesF1.Count}");

Console.WriteLine();

Console.WriteLine("Размеры строк:");

List<int> countElemtInLines = CountElemtInLines(linesF1);

WriteList(countElemtInLines);

Console.WriteLine();

linesF1.RemoveAt(linesF1.Count - 1);

Console.WriteLine("Список поле удаления последнего элемента");

WriteList(linesF1);

string pathF2 = @"D:\PracticCSharp\Prac17\z3\Files\TextFile2.txt";

await RecordInFile(pathF2, linesF1);

Console.WriteLine("Список записан в файл 2");

Console.WriteLine();

Console.WriteLine($"Самая длинна строка: {linesF1.Max()}");

Console.Write("Введите букву для поиска строки: ");

char let = Convert.ToChar(Console.ReadLine());

Console.WriteLine($"Строчки начинающиеся на {let}:");

WriteList(GetLineBeginWith(linesF1, let));

string pathF3 = @"D:\PracticCSharp\Prac17\z3\Files\TextFile3.txt";

linesF1.Reverse();

await RecordInFile(pathF3,linesF1);

Console.WriteLine("В файл 3 записан реверснутый список");

//

async Task<List<string>> GetTextFile(string path)

{

using (StreamReader reader = new StreamReader(path))

{

string? line;

List<string> lines = new List<string>();

while ((line = await reader.ReadLineAsync()) != null)

{

lines.Add(line);

}

return lines;

}

}

async Task RecordInFile(string path, List<string> lines)

{

using (StreamWriter writer = new StreamWriter(path, false))

{

foreach (string line in lines)

{

await writer.WriteLineAsync(line);

}

}

}

void WriteList<T>(List<T> list)

{

foreach (T elim in list)

{

Console.WriteLine($"{elim}");

}

}

List<int> CountElemtInLines(List<string> lines)

{

List<int> countElem = new List<int>();

foreach (string line in lines)

{

countElem.Add(line.Length);

}

return countElem;

}

List<string> GetLineBeginWith(List<string> lines, char ch) => lines.Where(l => l.StartsWith(ch)).ToList();

Таблица 16.3 – Входные и выходные данные программы задание №3

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| В исходном файле:  Первая строчка  2 строчка  Третья строчка. Перед ней 1 и 2 строчка  4 стр.  Пятая стр. | В выходном файле:  Первая строчка  2 строчка  Третья строчка. Перед ней 1 и 2 строчка  4 стр. |

Анализ результатов:

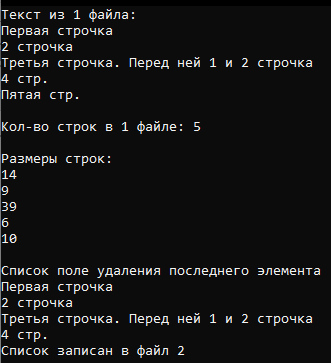


Рисунок 16.3 – Результат выполнения программы задание №3

Задание №4

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

Листинг программы:

string pathF1 = @"D:\PracticCSharp\Prac17\z4\Files\TextFile1.txt";

string pathF2 = @"D:\PracticCSharp\Prac17\z4\Files\TextFile2.txt";

List<string> linesF1 = await GetTextFile(pathF1);

RepleseList('0', '1', ref linesF1);

await RecordInFile(pathF2, linesF1);

List<string> linesF2 = await GetTextFile(pathF2);

RepleseList('1', '0', ref linesF2);

await RecordInFile(pathF1, linesF2);

async Task<List<string>> GetTextFile(string path)

{

using (StreamReader reader = new StreamReader(path))

{

string? line;

List<string> lines = new List<string>();

while ((line = await reader.ReadLineAsync()) != null)

{

lines.Add(line);

}

return lines;

}

}

async Task RecordInFile(string path, List<string> lines)

{

using (StreamWriter writer = new StreamWriter(path, false))

{

foreach (string line in lines)

{

await writer.WriteLineAsync(line);

}

}

}

void RepleseList(char oldChar, char newChar,ref List<string> list)

{

for (int i = 0; i < list.Count; i++)

{

list[i] = list[i].Replace(oldChar, newChar);

}

}

void WriteList(List<string> list)

{

foreach (string elem in list)

{

Console.WriteLine(elem);

}

}

Таблица 16.4 – Входные и выходные данные программы задание №4

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| В исходном:  0000023024  00350234502345  2340502300 000  0 3452345 23450 | В выходном:  1111123124  11351234512345  2341512311 111  1 3452345 23451 |

Анализ результатов:

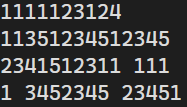


Рисунок 16.4 – Результат выполнения программы задание №4