Министерство образования и науки Российской Федерации

федеральное государственное автономное образовательное учреждение

высшего профессионального образования

**«Северный (Арктический) федеральный университет имени М.В. Ломоносова»**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | | | | | | | |  |
|  | Информационных систем и технологий | | | | | | | | | | | | |  |
|  | (наименование кафедры) | | | | | | | | | | | | |  |
|  |  | | | | | | | | | | | | |  |
|  | Данилов Алексей Александрович | | | | | | | | | | | | |  |
|  | (фамилия, имя, отчество студента) | | | | | | | | | | | | |  |
|  |  | | | | | | | | | | | | |  |
|  | Институт | ИМИКТ | | | курс | 2 | | | группа | 28 | | |  |  |
|  |  | | | | | | | | | | | | |  |
|  | |  | | --- | | 230400.62 Информационные системы и технологии | | (код и наименование направления подготовки/специальности) | | | | | | | | | | | | | |  |
|  | **ОТЧЁТ** | | | | | | | | | | | | |  |
|  | о лабораторном практикуме | | | | | | | | | | | | |  |
|  |  | | | | | | | | | | | | |  |
|  | По дисциплине | |  | Технология программирования | | | | | | | | | |  |
|  |  | |  |  | | |  |  | | |  |  | |  |
|  |  | |  |  | | |  |  | | |  |  | |  |
|  |  | |  |  | | |  |  | | |  |  | |  |
|  |  | |  |  | | |  |  | | |  |  | |  |
|  |  | |  |  | | |  |  | | |  |  | |  |
|  | Отметка о зачёте | |  |  | | | | | | |  |  | |  |
|  |  | |  |  | | |  |  | | |  | (дата) | |  |
|  |  | |  |  | | |  |  | | |  |  | |  |
|  | Руководитель практикума | | | | | |  |  | | |  |  | |  |
|  |  | |  |  | | | | | | |  |  | |  |
|  | Доцент | |  |  | | | | | | |  | И. С. Майоров | |  |
|  | (должность) | |  | (подпись) | | | | | | |  | (инициалы, фамилия) | |  |
|  |  | |  |  | | |  |  | | |  |  | |  |
|  |  | |  |  | | |  |  | | |  |  | |  |
|  |  | |  |  | | |  |  | | |  |  | |  |
|  |  | |  |  | | |  |  | | |  |  | |  |
|  |  | |  |  | | |  |  | | |  |  | |  |
|  |  | |  |  | | |  |  | | |  |  | |  |
|  |  | | | | | | | | | | | | |  |
|  | Архангельск  2014 | | | | | | | | | | | | |  |

ЛИСТ ДЛЯ ЗАМЕЧАНИЙ

1. Задание

Найти предложения, не содержащие запятых, посчитать количество слов в каждом предложении.

1. Планируемая реализация

Планировалась следующая логика исполнения программы.

Сначала выполнить получение и проверку на существование директории и файла, с которыми потом будет вестись работа, с помощью методов и экземпляров классов DirectoryInfo и FileInfo. В случае ошибок будут вызываться исключения с помощью оператора throw, и выполнение программы будет завершаться с сообщением об исключении.

Далее найденный .txt файл планировалось считывать из потока StreamReader построчно с помощью метода ReadLine() и заносим каждую строку в массив строк, при этом входящий файл не должен иметь предложений начинающихся на одной строке, а заканчивающихся на другой. Затем с помощью регулярного выражения каждая полученная строка разбивалась на предложения, и они заносились в List<string>. Среди всех предложений с помощью регулярного выражения искались предложения, не содержащие запятых. В этих предложениях подсчитывалось количество слов с помощью регулярного выражения, и для каждого предложения в листе в конце добавлялось количество слов в нём.

В конце, обрабатывая возможные ошибки, создавался файл для вывода результата с помощью метода File.Create(). Создавался поток, используя класс StreamWriter, и в файл отчёта с помощью метода WriteLine() выводились полученные строки.

1. Итоговая реализация

В ходе выполнения лабораторной работы были дополнительно созданы классы FileSearch и SearchInFile.

Выполнение программы начинается с метода класса FileSearch CheckPath(), этот метод проверяет существование директории, с которой планируется вестись работа.

Фрагмент программного кода

public static string CheckPath()

{

DirectoryInfo Path = new DirectoryInfo(@"C:\Users\1\Documents\2 курс 3 семестр - Технология программирования\Лаб5\");

if (Path.ToString().Length <= 32767){

if (Path.Exists){

return Path.ToString();

}

else{

throw new DirectoryNotFoundException();

}

}

else{

throw new PathTooLongException();

}

}

После этого идёт проверка на наличие в выбранной директории файла с текстом.

Фрагмент программного кода

public static string CheckFile(string Path)

{

string FileName = "text.txt";

FileInfo File = new FileInfo(Path + FileName);

if (File.Exists){

return File.ToString();

}

else{

throw new FileNotFoundException();

}

}

Далее выполняются методы класса SearchInFile, у этого класса есть два поля типа List<string>: ListText и LisrWithoutCommas. Первый метод CreateStream считывает текст из файла одной строкой с помощью метода ReadToEnd() класса StreamReader и разбивает текст на предложения, используя регуляроное выражение.

Фрагмент программного кода

internal List<string> CreateStream(string FilePath)

{

StreamReader IncomingFile = new StreamReader(FilePath, Encoding.Unicode);

string line;

string pattern = @".+?[.!?]";

ListText = new List<string>();

line = IncomingFile.ReadToEnd();

foreach (Match sentence in Regex.Matches(line, pattern, RegexOptions.IgnoreCase)){

ListText.Add(sentence.Value.Trim());

}

return ListText;

}

В следующем методе CommasExclusion() с помощью регулярного выражения отбираются без запятых, они записываются в ListWithoutCommas<string>.

Фрагмент программного кода

internal List<string> CommasExclusion()

{

ListWithoutCommas = new List<string>();

foreach (string line in ListText){

if (!line.Contains(",")){

ListWithoutCommas.Add(line);

}

}

return ListWithoutCommas;

}

Используя полученный лист, метод WordCount() подсчитывает количество слов в предложениях, используя регулярное выражение.

Фрагмент программного кода

private static int WordCount(string sentence)

{

string pattern = @"\w+(?>[\s\.])";

return Regex.Matches(sentence, pattern).Count;

}

В конце с помощью метода OutputResult() создаётся файл, в который выводится в столбик количество слов с помощью метода WriteLine() класса StreamWriter.

Фрагмент программного кода

private static void OutputResult(List<string> ListWithoutCommas)

{

StreamWriter OutputText = new StreamWriter(File.Create(@"C:\Users\1\Documents\2 курс 3 семестр - Технология программирования\Лаб5\Result.txt"));

foreach (string sentence in ListWithoutCommas){

OutputText.WriteLine(WordCount(sentence));

}

OutputText.Close();

Console.WriteLine("Программа подсчитывает количество слов в предложениях без запятых и выводит результат работы в файл.");

Console.WriteLine(@"Результат работы программы вы можете найти в папке C:\Users\1\Documents\2 курс 3 семестр - Технология программирования\Лаб5");

}

Ниже приведён общий порядок вызова методов в методе Main().

Фрагмент программного кода

static void Main()

{

SearchInFile Text = new SearchInFile();

try

{

string Path = FileSearch.CheckPath();

string FilePath = FileSearch.CheckFile(Path);

Text.CreateStream(FilePath);

}

catch (Exception e){

Console.WriteLine(e.Message);

}

Text.CommasExclusion();

try

{

OutputResult(Text.ListWithoutCommas);

}

catch (OverflowException e)

{

Console.WriteLine(e.Message);

}

Console.ReadKey();

}

ЗАКЛЮЧЕНИЕ

В рамках данной лабораторной работы была разработана программа, позволяющая найти в тексте предложения без запятых и подсчитать количество слов них. Входные данные получаются из файла, результат работы программы так же записывается в файл.