**Руководство пользователю, программисту.**

**ОСНОВНОЕ ЗАДАНИЕ:**

Имеется два входных файла – файл\_с\_текстом и файл\_словаря. Файл\_с\_текстом содержит текст в виде последовательностей текстовых строк, разделенных стандартным разделителем строки. Файл\_словаря содержит произвольное количество строк, каждая из которых содержит ровно одно слово.

Пример файла\_с\_текстом:

----НАЧАЛО ФАЙЛА----

Файл\_с\_текстом содержит текст

в виде последовательностей строк,

разделенных стандартным разделителем

строки. Файл\_с\_текстом содержит текст

в виде последовательностей строк,

разделенных стандартным разделителем строки.

----КОНЕЦ ФАЙЛА----

Пример файла\_словаря:

----НАЧАЛО ФАЙЛА----

текст

виде

изображение

файл

строки

----КОНЕЦ ФАЙЛА----

Необходимо создать программу, создающую корректный HTML\_файл, содержащий весь текст из файла\_с\_текстом в котором помечены жирным и наклонным шрифтом все слова из файла\_словаря. HTML\_файл должен отображаться в любом Internet-браузере (IE, Firefox,Chrome и др.) корректно, показывая весь текст, с выделением указанных слов.

Пример отображения HTML\_файла в окне произвольного браузера (на основе примера выше):

----НАЧАЛО ВЫВОДА ----

Файл\_с\_текстом содержит текст

в виде последовательностей строк,

разделенных стандартным разделителем

***строки***. Файл\_с\_текстом содержит

текст в виде последовательностей строк,

разделенных стандартным разделителем ***строки***.

----КОНЕЦ ВЫВОДА ----

**Руководство пользователю.**

Программа предназначена для разбиения исходного текста на HTML файлы размером по 10 предложений.

До запуска программы пользователь помещает обрабатываемый текст в файл «TheTextFile.txt». В файле «Dictionary.txt» пользователь помещает в слова для выделения, каждое слово в отдельной строке.

При запуске программы пользователь увидит консольное окно, в которое будет по одному предложению выводиться текст. Слова, которые есть в словаре («Dictionary.txt») будут выделены ярким цветом.

По окончании работы программы, пользователь может посмотреть HTML файлы в той же директории, что и исполняемые файлы программы. Слова, которые есть в словаре («Dictionary.txt») будут выделены жирным курсивом.

**Руководство программисту.**

Входные данные:

«TheTextFile.txt» - файл текста;

«Dictionary.txt» - файл словаря.

Выходные данные:

«HTML1.htm», «HTML2.htm»… . – HTML файлы с текстом, нарезанным по 10 предложений. Слова из словарика выделены жирным курсивом.

Программа сосоит из файлов:

«Program.cs», «DictionaryClass.cs», «HTMLConnectionClass.cs», «C\_W.cs».

«Program.cs» - основной файл программы. Он включает в себя:

/// <summary>Метод, готовый к использованию.</summary>

/// <param name="TextFilePath">Путь к тексту</param>

/// <param name="DictionaryFilePath">Путь к словарю.</param>

/// <param name="MaxLineCount">Максимум предложений в HTML файле</param>

static void DoIt(string TextFilePath, string DictionaryFilePath, int MaxLineCount)

{…}

static void Main(string[] args){

DoIt("TheTextFile.txt", "Dictionary.txt",10);

Console.Read();

}

Основная идея метода «DoIt» в том, что используется отдельная переменная «string StringOverview» для накопления текста. Если в StringOverview найдено окончание предложения, то это предложение вырезается, и централизовано выводится на экран и в HTML файл. При этом учитывается наличие или отвутствие каждого отдельного слова в словаре. Проверяется количество уже записанных предложений в текущем HTML файле. При необходимости происходит смена HTML файла. По окончании работы программы весь оставшийся текст в переменной StringOverview воспринимается как последнее предложение и тоже выводится на экран.

«DictionaryClass.cs» - содержит «public class DictionaryClass{…}»  
DictionaryClass – содержит метод загрузки словаря из файла, он вшиты в конструктор.

И метод обработки некоторых исключительных ситуаций, он вшит в DictionaryClass.Contains\_.

Нами для хранения используется стандартный List<string>, и для поиска вхождения в словарь используется стандартный метод .Contains(string).

«HTMLConnectionClass.cs» - содержит класс, в котором реализован доступ к HTML файлам.

/// <summary> Конструктор</summary>

public HTMLConnectionClass(int \_maxcount,int \_id){…}//-Конструктор

/// <summary>Запись в HTML файл.</summary>

public void WriteLine(string s){…}

/// <summary>Завершение записи в файл.</summary>

public void fin(){…}

«C\_W.cs» - содержит статичный класс, в котором реализован вывод цветного текста в консоль. Этот класс был обрезан и взят из другого приложения.