**Отчет по лабораторной работе № 5**

**по курсу «Базовые компоненты интернет технологий»**

Выполнил:

студент группы ИУ5-33

Дарья Павлюнина

Дата: 10.12.2017

**Описание задания лабораторной работы:**

Разработать программу, реализующую работу с файлами.

1. Программа должна быть разработана в виде приложения Windows Forms на

языке C#. По желанию вместо Windows Forms возможно использование

WPF (Windows Presentation Foundation).

2. Добавить кнопку, реализующую функцию чтения текстового файла в

список слов List<string>.

3. Для выбора имени файла используется класс OpenFileDialog, который

открывает диалоговое окно с выбором файла. Ограничить выбор только

файлами с расширением «.txt».

4. Для чтения из файла рекомендуется использовать статический метод

ReadAllText() класса File (пространство имен System.IO). Содержимое файла считывается методом ReadAllText() в виде одной строки, далее делится на слова с использованием метода Split() класса string. Слова сохраняются в список List<string>.

5. При сохранении слов в список List<string> дубликаты слов не записываются. Для проверки наличия слова в списке используется метод Contains().

6. Вычислить время загрузки и сохранения в список с использованием класса Stopwatch (пространство имен System.Diagnostics). Вычисленное время вывести на форму в поле ввода (TextBox) или надпись (Label).

7. Добавить на форму поле ввода для поиска слова и кнопку поиска. При нажатии на кнопку поиска осуществлять поиск введенного слова в списке.9

Слово считается найденным, если оно входит в элемент списка как

подстрока (метод Contains() класса string).

8. Добавить на форму список (ListBox). Найденные слова выводить в список с

использованием метода «название\_списка.Items.Add()». Вызовы метода «название\_списка.Items.Add()» должны находится между вызовами методов «название\_списка.BeginUpdate()» и «название\_списка. EndUpdate()».

9. Вычислить время поиска с использованием класса Stopwatch. Вычисленное время вывести на форму в поле ввода (TextBox) или надпись (Label)

**Текст программы на языке C#.**

*Файл Distance.cs*

using System;

namespace DistanceLibrary

{

public static class Distance

{

public static int CalculateDistance(string str1Param, string str2Param)

{

if ((str1Param == null) || (str2Param == null)) return -1;

int str1Len = str1Param.Length;

int str2Len = str2Param.Length;

//Если хотя бы одна строка пустая,

//возвращается длина другой строки

if ((str1Len == 0) && (str2Len == 0)) return 0;

if (str1Len == 0) return str2Len;

if (str2Len == 0) return str1Len;

//Приведение строк к верхнему регистру

string str1 = str1Param.ToUpper();

string str2 = str2Param.ToUpper();

//Объявление матрицы

int[,] matrix = new int[str1Len + 1, str2Len + 1];

//Инициализация нулевой строки и нулевого столбца матрицы

for (int i = 0; i <= str1Len; i++) matrix[i, 0] = i;

for (int j = 0; j <= str2Len; j++) matrix[0, j] = j;

//Вычисление расстояния Дамерау-Левенштейна

for (int i = 1; i <= str1Len; i++)

{

for (int j = 1; j <= str2Len; j++)

{

//Эквивалентность символов, переменная symbEqual

//соответствует m(s1[i],s2[j])

int symbEqual = (

(str1.Substring(i - 1, 1) ==

str2.Substring(j - 1, 1)) ? 0 : 1);

int ins = matrix[i, j - 1] + 1; //Добавление

int del = matrix[i - 1, j] + 1; //Удаление

int subst = matrix[i - 1, j - 1] + symbEqual; //Замена

//Элемент матрицы вычисляется

//как минимальный из трех случаев

matrix[i, j] = Math.Min(Math.Min(ins, del), subst);

//Дополнение Дамерау по перестановке соседних символов

if ((i > 1) && (j > 1) &&

(str1.Substring(i - 1, 1) == str2.Substring(j - 2, 1)) &&

(str1.Substring(i - 2, 1) == str2.Substring(j - 1, 1)))

{

matrix[i, j] = Math.Min(matrix[i, j], matrix[i - 2, j - 2] + symbEqual);

}

}

}

//Возвращается нижний правый элемент матрицы

return matrix[str1Len, str2Len];

}

}

}

*Файл Form1.cs*

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.IO;

using System.Diagnostics;

using DistanceLibrary;

namespace Lab\_4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

private void Form1\_Load(object sender, EventArgs e)

{

//--

}

private void label1\_Click(object sender, EventArgs e)

{

//-

}

private void button1\_Click\_1(object sender, EventArgs e)

{

OpenFileDialog file = new OpenFileDialog();

if (file.ShowDialog() == DialogResult.OK)

{

Stopwatch time = new Stopwatch();

time.Start();

string text = File.ReadAllText(file.FileName);

char[] seps = new char[] { ' ', '.', ',', '!', '?', '/', '\t', '\n', '(', ')' };

string[] textArray = text.Split(seps);

foreach (string word in textArray)

{

string trimmedWord = word.Trim();

if (!words.Contains(trimmedWord))

{

words.Add(trimmedWord);

}

}

richTextBox1.Text = text.ToString();

time.Stop();

label1.Text = "Затраченное время на загрузку файла: " + time.Elapsed.ToString();

}

else

{

MessageBox.Show("Что-то не так!");

}

}

private void button2\_Click(object sender, EventArgs e)

{

string wordToFind = textBox1.Text.Trim();

textBox1.Text = wordToFind;

int maxDistance = Convert.ToInt32(textBox2.Text);

if (!string.IsNullOrEmpty(wordToFind) && words.Count > 0)

{

string wordToFindUpper = wordToFind.ToUpper();

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

Stopwatch time = new Stopwatch();

time.Start();

foreach (string str in words)

{

if (Distance.CalculateDistance(str.ToUpper(), wordToFindUpper) <= maxDistance)

{

tempList.Add(str);

}

}

time.Stop();

label1.Text = "Затраченное время на поиск слова: " + time.Elapsed.ToString();

listBox2.BeginUpdate();

listBox2.Items.Clear();

foreach (string str in tempList)

{

listBox2.Items.Add(str);

}

listBox2.EndUpdate();

}

else

{

MessageBox.Show("Что-то не то!");

}

}

private void listBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

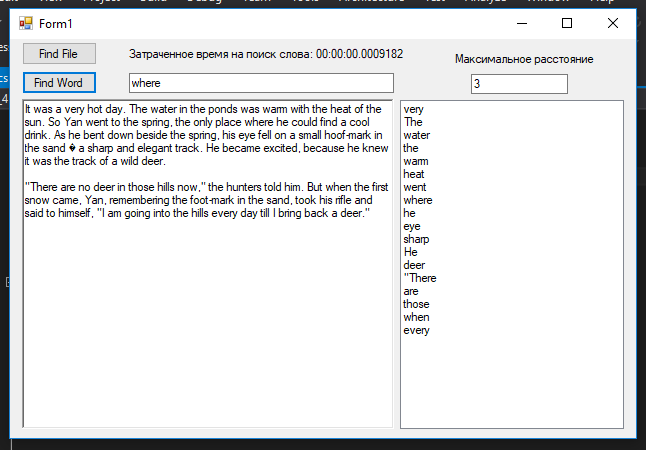
//-

}

}

}

**Результаты выполнения программы:**

****

**Диаграмма классов**

