Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Московский государственный технический университет имени Н.Э. Баумана»

(МГТУ им. Н.Э. Баумана)

****

Факультет ИУ

Кафедра ИУ5

**Отчет по лабораторной работе № 5**

Студент Гаджиев К.К.

Группа ИУ5-32

Название предмета БКИТ

Руководитель

Гапанюк Ю.Е. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата подпись

Студент

Гаджиев К.К. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата подпись

Задание

Разработать программу, реализующую вычисление расстояния Левенштейна с использованием алгоритма Вагнера-Фишера.

1. Программа должна быть разработана в виде библиотеки классов на языке C#.
2. Использовать самый простой вариант алгоритма без оптимизации.
3. Дополнительно возможно реализовать вычисление расстояния Дамерау-Левенштейна (с учетом перестановок соседних символов).
4. Модифицировать предыдущую лабораторную работу, вместо поиска подстроки используется вычисление расстояния Левенштейна.
5. Предусмотреть отдельное поле ввода для максимального расстояния. Если расстояние Левенштейна между двумя строками больше максимального, то строки считаются несовпадающими и не выводятся в список результатов.

Листинг программы

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Diagnostics;

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 button1\_Click(object sender, EventArgs e)

{

OpenFileDialog chosefile = new OpenFileDialog();

chosefile.Filter = "текстовые файлы|\*.txt";

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

{

textBox1.Enabled = true;

textBox2.Enabled = true;

button2.Enabled = true;

textBox5.Enabled = true;

button3.Enabled = true;

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

char[] sep = { ' ', '.', ',', '!', '?', '/', '\t', '\n' };

string[] WordInText = text.Split(sep);

listBox1.BeginUpdate();

listBox1.Items.Clear();

foreach (string str in WordInText)

{

string temp = str.Trim();

if (!words.Contains(temp))

{

words.Add(temp);

listBox1.Items.Add(temp);

}

}

listBox1.EndUpdate();

}

else

{

MessageBox.Show("Выберите файл");

}

}

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

{

string word1 = textBox2.Text.Trim();

string word2 = textBox1.Text.Trim();

string word = textBox2.Text.Trim();

if (!string.IsNullOrWhiteSpace(word1) && !string.IsNullOrWhiteSpace(word2) && words.Count > 0)

{

Stopwatch time = new Stopwatch();

time.Start();

textBox3.Text = Distance.CountDistance(word1, word2).ToString();

time.Stop();

textBox4.Text = time.Elapsed.ToString();

}

else

{

MessageBox.Show("Введите слово для поиска");

}

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

int max = int.Parse(textBox5.Text.Trim());

if (max >= 0 && words.Count > 0)

{

Stopwatch time = new Stopwatch();

time.Start();

listBox2.BeginUpdate();

listBox2.Items.Clear();

foreach (string word1 in words)

{

foreach (string word2 in words)

{

if (word1 != word2)

{

int d = Distance.CountDistance(word1, word2);

if (d <= max)

{

string temp = word1 + " - " + word2 + " -> " + d.ToString();

listBox2.Items.Add(temp);

}

}

}

}

listBox2.EndUpdate();

time.Stop();

textBox6.Text = time.Elapsed.ToString();

}

else

{

MessageBox.Show("Введите максимальное значение");

}

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

}

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

{

}

}

}

sing System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_4

{

public static class Distance

{

public static int CountDistance(string word1, string word2)

{

if ((word1 == null) || (word2 == null)) return -1;

int len1 = word1.Length;

int len2 = word2.Length;

if ((len1 == 0) && (len2 == 0)) return 0;

if (len1 == 0) return len2;

if (len2 == 0) return len1;

string str1 = word1.ToUpper();

string str2 = word2.ToUpper();

int[,] matrix = new int[len1 + 1, len2 + 1];

for (int i = 0; i <= len1; i++) matrix[i, 0] = i;

for (int j = 0; j <= len2; j++) matrix[0, j] = j;

for (int i = 1; i <= len1; i++)

{

for (int j = 1; j <= len2; j++)

{

int symbEqual = ((str1.Substring(i - 1, 1) == str2.Substring(j - 1, 1)) ? 0 : 1);

int I = matrix[i, j - 1] + 1;

int D = matrix[i - 1, j] + 1;

int S = matrix[i - 1, j - 1] + symbEqual;

matrix[i, j] = Math.Min(Math.Min(I, D), S);

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);

}

}

}

return matrix[len1, len2];

}

}

}

Результаты работы программы

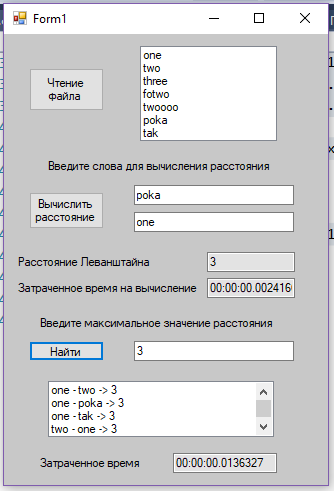


Диаграмма классов

