МГТУ им. Баумана  
Факультет «Информатика и системы управления»  
Кафедра «Автоматизированные системы обработки информации и управления»  
Дисциплина «Базовые компоненты интернет-технологий»

Отчёт по лабораторной работе №5

Выполнил:  
Студент группы ИУ5-33Б  
Андреев А.А.   
Преподаватель:  
Гапанюк Ю.

1. **Описание задания**
   1. Разработать программу, реализующую вычисление расстояния Левенштейна с использованием алгоритма Вагнера-Фишера.
   2. Программа должна быть разработана в виде библиотеки классов на языке C#.
   3. Использовать самый простой вариант алгоритма без оптимизации.
   4. Дополнительно возможно реализовать вычисление расстояния Дамерау-Левенштейна (с учетом перестановок соседних символов).
   5. Модифицировать предыдущую лабораторную работу, вместо поиска подстроки используется вычисление расстояния Левенштейна.
   6. 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.
2. **Текст программы**

**Элемент Distance\_Levenstein.cs**

using System;

namespace Lab5\_1

{

public static class Distance\_Levenstein

{

public static int Distance(string str1Param, string str2Param)

{

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

int str1Len = str1Param.Length;

int str2Len = str2Param.Length;

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

if (str1Len == 0) return str2Len;

if (str2Len == 0) return str1Len;

string str1 = str1Param.ToUpper();

string str2 = str2Param.ToUpper();

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

for (int i = 0; i <= str1Len; i++)

matrix[i, 0] = i;

for (int j = 0; j <= str2Len; j++)

matrix[0, j] = j;

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

{

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

{

int symbEqual = (

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

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

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

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

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

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

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[str1Len, str2Len];

}

}

}

**Элемент MainWindow.xaml.cs**

using System;

using System.IO;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Microsoft.Win32;

using System.Diagnostics;

using ClassLibraryLab5;

namespace Lab5\_2

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

List< String> list = new List<String>();

public MainWindow()

{

InitializeComponent();

}

private void Read\_File\_Button\_Click(object sender, RoutedEventArgs e)

{

OpenFileDialog Dialog\_one = new OpenFileDialog();

Dialog\_one.Filter = "text\_files|\*.txt";

if (Dialog\_one.ShowDialog()== true)

{

Stopwatch mytimer = new Stopwatch();

mytimer.Start();

string text = File.ReadAllText(Dialog\_one.FileName);

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

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

foreach (string strTemp in textArray)

{

string str = strTemp.Trim();

if (!list.Contains(str)) list.Add(str);

}

mytimer.Stop();

this.textbox\_for\_timer.Text = mytimer.Elapsed.ToString();

this.textbox\_for\_list.Text = list.Count.ToString();

}

else

{

MessageBox.Show("выберите файл");

}

}

private void Search\_button\_Click(object sender, RoutedEventArgs e)

{

string word = this.Inputwords.Text.Trim();

if (!string.IsNullOrWhiteSpace(word) && list.Count > 0 && word != "введите слово, которое хотите найти")

{

string wordUpper = word.ToUpper();

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

Stopwatch t = new Stopwatch();

t.Start();

int maxRange = Int32.Parse(this.Max\_range.Text.Trim());

foreach (string str in list)

{

if (Distance\_Levenstein.Distance(str, wordUpper) <= maxRange)

{

tempList.Add(str);

}

}

t.Stop();

this.Anothertimer.Text = t.Elapsed.ToString();

this.found\_words.Items.Clear();

foreach (string str in tempList)

{

this.found\_words.Items.Add(str);

}

}

else

{

MessageBox.Show("введите слово, которое хотите найти");

}

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

}

}

1. **Экранные формы с примерами выполнения программы**

