**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №5

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-35б |  | преподаватель каф. ИУ5 |
| Ширшов Артём |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2020 г.

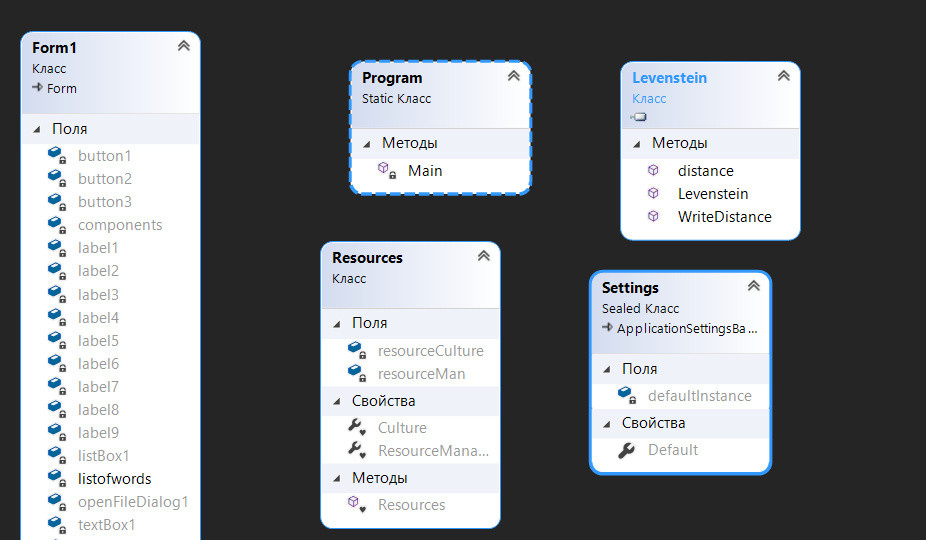
# Задание.

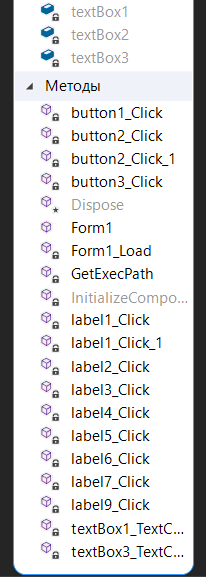
# Задание.

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.

# Диаграмма классов





# Примечание

Данная лабораторная работа была модифицирована до лабораторной работы 5, поэтому вид формы на скриншотах результатах содержит дополнительные поля. Код, приведённый ниже был упрощён до исходного состояния

# Текст программы

Levenstein.cs

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Biblioteka

{

public class Levenstein

{

public static int distance(string str1, string str2)

{

if ((str1 == null) || (str2 == null)) return -1;

int str1len = str1.Length;

int str2len = str2.Length;

if ((str1len == 0) && (str2len == 0)) return 0;

if (str1len == 0) return str2len;

if (str2len == 0) return str1len;

string strnew1 = str1.ToUpper();

string strnew2 = str2.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 = ((strnew1.Substring(i - 1, 1) == strnew2.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) &&

(strnew1.Substring(i - 1, 1) == strnew2.Substring(j - 2, 1)) &&

(strnew1.Substring(i - 2, 1) == strnew2.Substring(j - 1, 1)))

{

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

}

}

}

return matrix[str1len, str2len];

}

public static string WriteDistance(string firstparam, string secondparam)

{

int d = distance(firstparam, secondparam);

return ("'" + firstparam + "','" + secondparam + "' -> " + d.ToString());

}

}

}

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.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Diagnostics;

using Biblioteka;

namespace ButtonForm

{

public partial class Form1 : Form

{

private List<string> listofwords = new List<string>();

static string GetExecPath()

{

string exeFileName= System.Reflection.Assembly.GetExecutingAssembly().Location;

string Result = Path.GetDirectoryName(exeFileName);

return Result;

}

public Form1()

{

InitializeComponent();

}

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

{

if (openFileDialog1.ShowDialog() == DialogResult.Cancel)

return;

textBox1.Clear();

listofwords.Clear();

Stopwatch t = new Stopwatch();

Stopwatch t1 = new Stopwatch();

t.Start();

string fileName = openFileDialog1.FileName;

t.Stop();

label3.Text = t.Elapsed.ToString();

t1.Start();

string ReadText= File.ReadAllText(fileName);

char[] delimiterChars = { ' ', ',', '.', ':', '\t' };

string[] mas = ReadText.Split(delimiterChars);

foreach (string word in mas)

{

string clearword = word.Trim();

if(!listofwords.Contains(clearword))

listofwords.Add(clearword);

}

t1.Stop();

label4.Text = t1.Elapsed.ToString();

foreach (string word in listofwords)

{

textBox1.Text+= word+" ";

}

label8.Text = listofwords.Count().ToString();

}

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

{

}

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

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

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

{

}

private void label1\_Click\_1(object sender, EventArgs e)

{

}

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

{

}

private void button2\_Click\_1(object sender, EventArgs e)

{

Stopwatch t = new Stopwatch();

t.Start();

string findWord = textBox3.Text;

if (listofwords.Contains(findWord))

{

MessageBox.Show(

"Слово найдено"

);

listBox1.BeginUpdate();

listBox1.Items.Add(findWord);

listBox1.EndUpdate();

}

else

{

MessageBox.Show(

"Слово не найдено"

);

}

t.Stop();

label6.Text = t.Elapsed.ToString();

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

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

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void label6\_Click(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void label7\_Click(object sender, EventArgs e)

{

}

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

{

listBox1.Items.Clear();

string matchword = textBox3.Text;

int maxdistance;

if (!Int32.TryParse(textBox2.Text, out maxdistance))

{

MessageBox.Show(

"Введите максимальное расстояние"

);

}

else {

foreach (string word in listofwords)

{

if (Biblioteka.Levenstein.distance(matchword, word) <= maxdistance)

{

listBox1.Items.Add(Biblioteka.Levenstein.WriteDistance(matchword, word));

}

}

}

}

private void label9\_Click(object sender, EventArgs e)

{

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ButtonForm

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

# Результаты

