**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 |
| Хижняков Вадим |  |  |
| Подпись и дата: |  | Подпись и дата: |

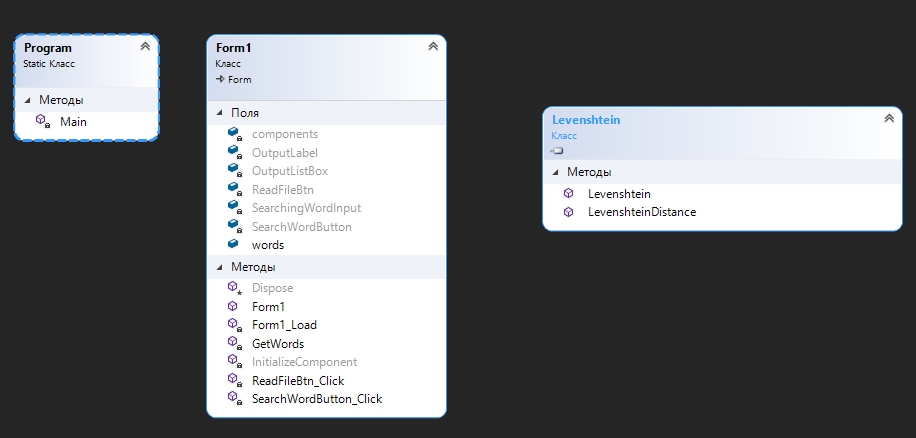
Москва, 2020 г.

# 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



# Текст программы

Levenstein.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Sem3Lab5\_Levenshteyn

{

public class Levenshtein

{

public static int LevenshteinDistance(string string1, string string2)

{

if (string1 == null) throw new ArgumentNullException("string1");

if (string2 == null) throw new ArgumentNullException("string2");

int diff;

int[,] m = new int[string1.Length + 1, string2.Length + 1];

for (int i = 0; i <= string1.Length; i++) { m[i, 0] = i; }

for (int j = 0; j <= string2.Length; j++) { m[0, j] = j; }

for (int i = 1; i <= string1.Length; i++)

{

for (int j = 1; j <= string2.Length; j++)

{

diff = (string1[i - 1] == string2[j - 1]) ? 0 : 1;

m[i, j] = Math.Min(Math.Min(m[i - 1, j] + 1,

m[i, j - 1] + 1),

m[i - 1, j - 1] + diff);

}

}

return m[string1.Length, string2.Length];

}

}

}

Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Net.NetworkInformation;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

//Custom

using Sem3Lab5\_Levenshteyn;

namespace Sem3Lab4\_WF\_Files

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

public static List<string> words;

List<string> GetWords(string path)

{

string fileData = File.ReadAllText(path, Encoding.UTF8);

List<string> words = fileData.Split().ToList();

words = words.Distinct().ToList();

return words;

}

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

{

using (OpenFileDialog openFileDialog = new OpenFileDialog())

{

openFileDialog.InitialDirectory = "c:\\";

openFileDialog.Filter = "txt files (\*.txt)|\*.txt";

openFileDialog.FilterIndex = 2;

openFileDialog.RestoreDirectory = true;

string path = @"c:\file.txt";

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

{

path = openFileDialog.FileName;

}

if (!File.Exists(path))

{

string textSeed = "Zdarova che kavo?";

File.WriteAllText(path, textSeed, Encoding.UTF8);

}

words = GetWords(path);

OutputListBox.BeginUpdate();

OutputListBox.Items.Clear();

OutputListBox.EndUpdate();

}

}

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

{

Stopwatch stopwatch = new Stopwatch();

stopwatch.Start();

string SearchingWord = SearchingWordInput.Text;

OutputListBox.BeginUpdate();

var LevensteinDistances = words.Select(word => new { LevD = Levenshtein.LevenshteinDistance(word, SearchingWord), Word = word }).ToList();

var minLevDist = LevensteinDistances.Min(i => i.LevD);

var res = LevensteinDistances.Where(i => i.LevD == minLevDist);

foreach (var item in res)

{

OutputListBox.Items.Add(item.Word);

}

OutputListBox.EndUpdate();

stopwatch.Stop();

TimeSpan ts = stopwatch.Elapsed;

string output = String.Format("{0}s {1}ms", ts.Seconds, ts.Milliseconds);

OutputLabel.Text = output + "\t Всего слов " + words.Count() + " штук";

}

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

{

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Sem3Lab4\_WF\_Files

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

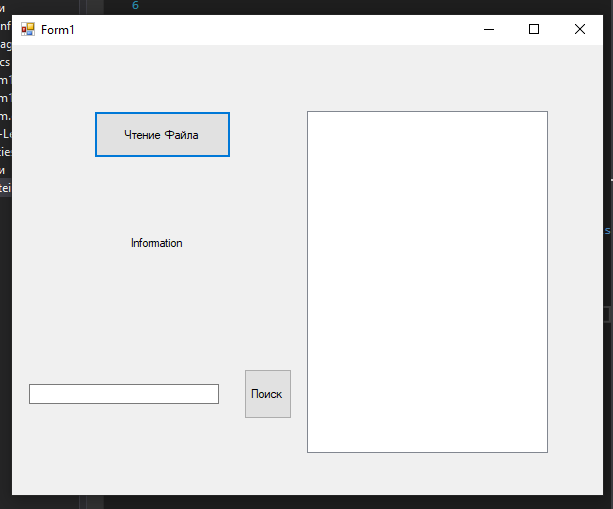
Application.Run(new Form1());

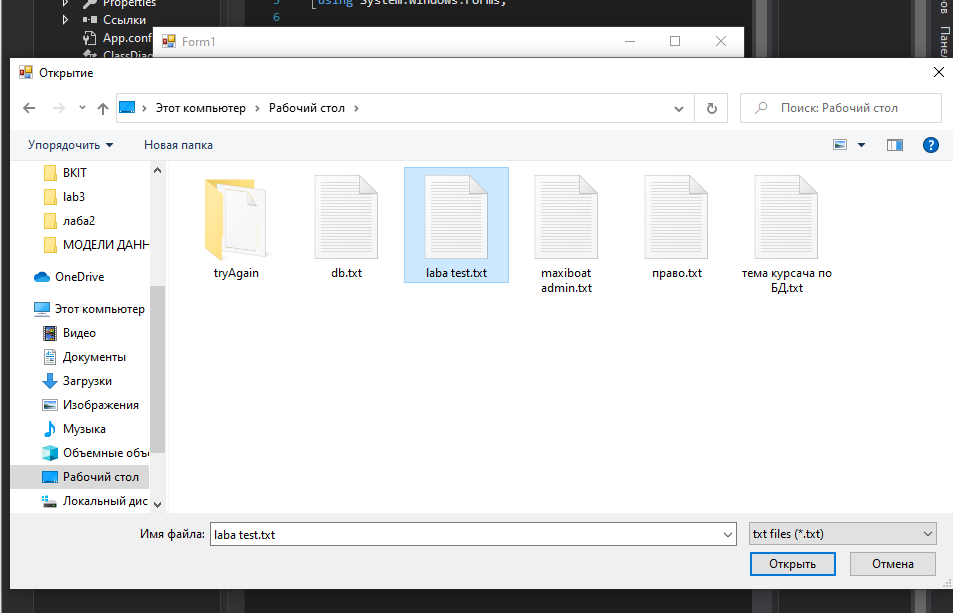
}

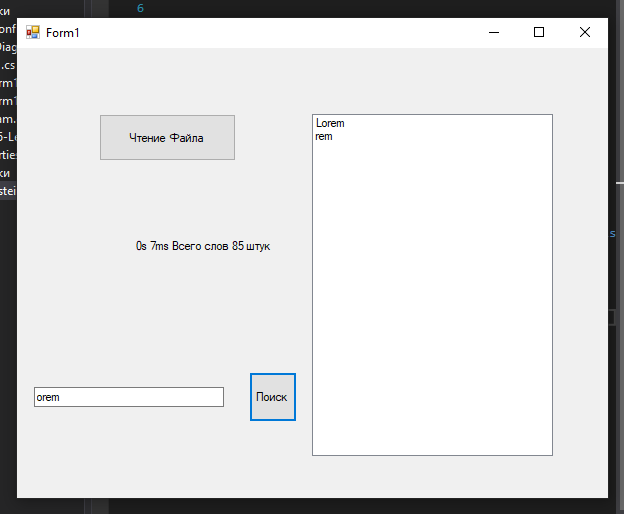
}

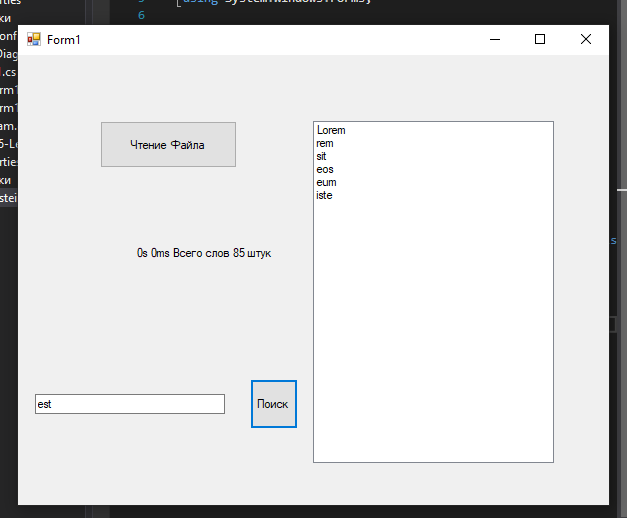
}

# Результаты









Test txt file:

