**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №5

«Работа с файлами в оконном приложении»

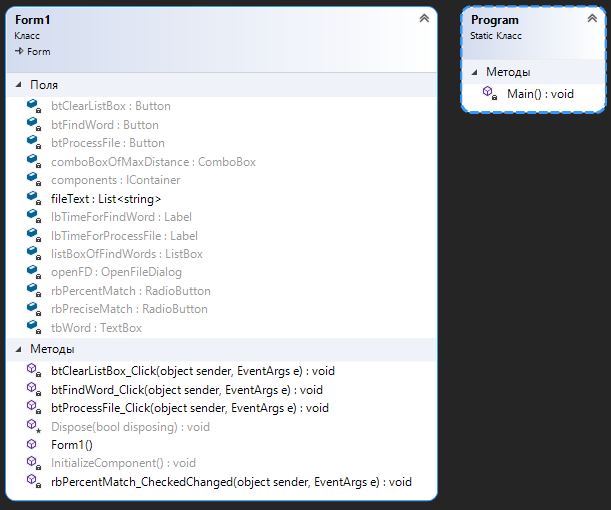
|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32 |  | преподаватель каф. ИУ5 |
| Барышников Михаил |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

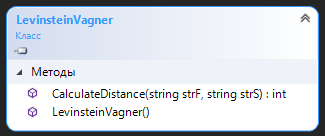
Москва, 2020 г.

**Задача**

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

* Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Diagnostics;

using VagnerFisherAlgorithmlib;

namespace FileSearch

{

public partial class Form1 : Form

{

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

public Form1()

{

InitializeComponent();

}

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

{

fileText.Clear();

Stopwatch fixedTime = new Stopwatch();

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

{

fixedTime.Start();

string[] text = File.ReadAllText(openFD.FileName).Split(' ');

foreach (string i in text)

if (!fileText.Contains(i))

fileText.Add(i);

fixedTime.Stop();

lbTimeForProcessFile.Text = fixedTime.Elapsed.ToString();

}

}

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

{

if (tbWord.Text != "" && fileText.Count != 0)

{

Stopwatch fixedTime = new Stopwatch();

int maxDistance;

if (rbPreciseMatch.Checked)

{

bool isFind = false;

fixedTime.Start();

if (fileText.Contains(tbWord.Text))

isFind = true;

fixedTime.Stop();

if (isFind)

{

listBoxOfFindWords.BeginUpdate();

listBoxOfFindWords.Items.Add(tbWord.Text);

listBoxOfFindWords.EndUpdate();

}

lbTimeForFindWord.Text = fixedTime.Elapsed.ToString();

}

else if (Int32.TryParse(comboBoxOfMaxDistance.Text, out maxDistance))

{

fixedTime.Start();

foreach (string i in fileText)

{

if (LevinsteinVagner.CalculateDistance(i, tbWord.Text) <= maxDistance)

{

listBoxOfFindWords.BeginUpdate();

listBoxOfFindWords.Items.Add(i);

listBoxOfFindWords.EndUpdate();

}

}

fixedTime.Stop();

lbTimeForFindWord.Text = fixedTime.Elapsed.ToString();

}

}

tbWord.Text = "";

}

private void rbPercentMatch\_CheckedChanged(object sender, EventArgs e)

{

comboBoxOfMaxDistance.Visible = rbPercentMatch.Checked;

}

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

{

listBoxOfFindWords.Items.Clear();

}

}

}

* Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace FileSearch

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**Результаты выполнения программы**

