**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

«Расстояние Левенштейна»

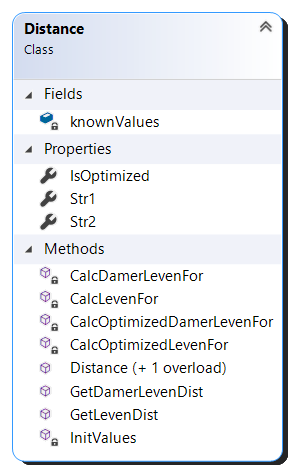
|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б |  | преподаватель каф. ИУ5 |
| Бабин Артём |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

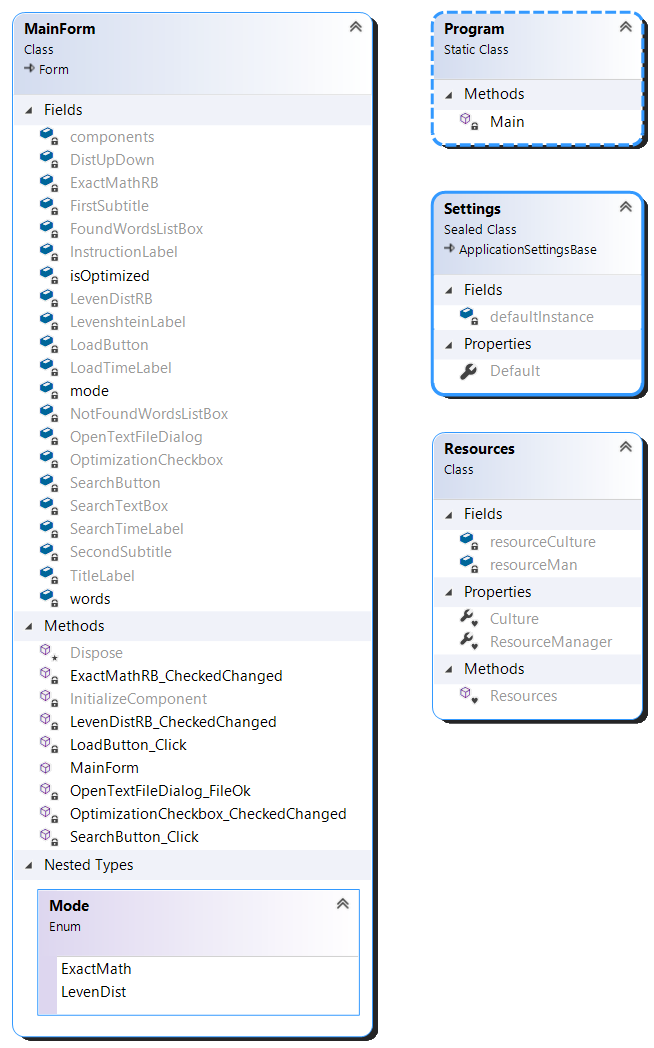
Москва, 2020 г.

**Условие задания:**

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:**

**Program.cs:**

using System;

using System.Windows.Forms;

namespace BKIT\_LAB4

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new MainForm());

}

}

}

**MainForm.cs:**

using System;

using System.IO;

using System.Collections.Generic;

using System.ComponentModel;

using System.Diagnostics;

using System.Text;

using System.Windows.Forms;

using BKIT\_LAB5;

namespace BKIT\_LAB4

{

public partial class MainForm : Form

{

enum Mode { ExactMath, LevenDist };

Mode mode;

bool isOptimized;

List<string> words;

public MainForm()

{

InitializeComponent();

words = new List<string>();

isOptimized = false;

}

private void OpenTextFileDialog\_FileOk(object sender, CancelEventArgs e)

{

Stopwatch timer = new Stopwatch();

timer.Start();

string[] text = File.ReadAllText(OpenTextFileDialog.FileName, Encoding.UTF8).Split(' ');

foreach (string word in text)

{

if (!words.Contains(word))

words.Add(word);

}

timer.Stop();

LoadTimeLabel.Text = "Load time: " + timer.Elapsed.TotalMilliseconds.ToString() + " ms";

ExactMathRB.Visible = true;

LevenDistRB.Visible = true;

OptimizationCheckbox.Visible = true;

LoadTimeLabel.Visible = true;

}

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

{

Stopwatch timer = new Stopwatch();

bool isExists = false;

timer.Start();

if (mode == Mode.ExactMath)

isExists = words.Contains(SearchTextBox.Text);

else

{

Distance dist = new Distance();

dist.IsOptimized = isOptimized;

dist.Str1 = SearchTextBox.Text;

int limitDiff = Convert.ToInt32(DistUpDown.Value);

foreach (string word in words)

{

dist.Str2 = word;

if (dist.GetLevenDist() <= limitDiff)

{

isExists = true;

break;

}

}

}

timer.Stop();

SearchTimeLabel.Text = "Search time: " + timer.Elapsed.TotalMilliseconds.ToString() + " ms";

ListBox targetListBox = (isExists) ? FoundWordsListBox : NotFoundWordsListBox;

targetListBox.BeginUpdate();

targetListBox.Items.Add(SearchTextBox.Text);

targetListBox.EndUpdate();

}

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

{

OpenTextFileDialog.ShowDialog();

}

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

{

if (ExactMathRB.Checked)

{

mode = Mode.ExactMath;

LevenshteinLabel.Visible = false;

DistUpDown.Visible = false;

InstructionLabel.Visible = true;

SearchTextBox.Visible = true;

SearchButton.Visible = true;

SearchTimeLabel.Visible = true;

FoundWordsListBox.Visible = true;

NotFoundWordsListBox.Visible = true;

LoadTimeLabel.Visible = true;

FirstSubtitle.Visible = true;

SecondSubtitle.Visible = true;

}

}

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

{

if (LevenDistRB.Checked)

{

mode = Mode.LevenDist;

LevenshteinLabel.Visible = true;

DistUpDown.Visible = true;

InstructionLabel.Visible = true;

SearchTextBox.Visible = true;

SearchButton.Visible = true;

SearchTimeLabel.Visible = true;

FoundWordsListBox.Visible = true;

NotFoundWordsListBox.Visible = true;

LoadTimeLabel.Visible = true;

FirstSubtitle.Visible = true;

SecondSubtitle.Visible = true;

}

}

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

{

isOptimized = OptimizationCheckbox.Checked;

}

}

}

**Примеры выполнения программы:**

|  |  |
| --- | --- |
| *Пример* | *Вывод программы* |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |