**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №5

«Разработка программы, реализующей вычисление расстояния Левенштейна с использованием алгоритма Вагнера-Фишера»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-34Б |  | преподаватель каф. ИУ5 |
| Сергеев Илья |  | Гапанюк Ю. Е. |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

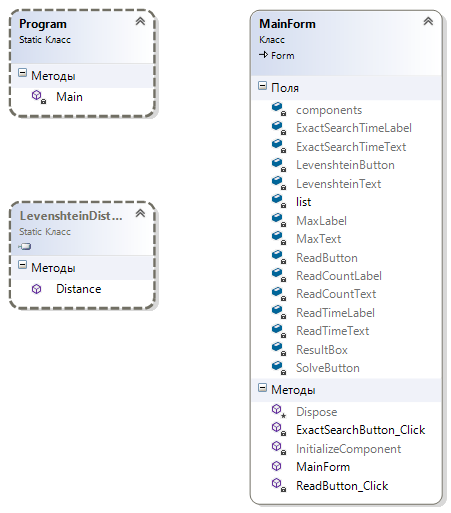
Москва, 2018 г.

**Описание задания**

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**

Диаграмма классов генерируется автоматически в среде Visual Studio:



**Текст программы (листинг)**

Листинг класса библиотеки:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace LevenshteinLibrary

{

public static class LevenshteinDistance

{

/// <summary>

/// Damerau–Levenshtein calculating class

/// </summary>

public static int Distance(string P1, string P2)

{

if ((P1 == null) || (P2 == null))

return -1; //in this case we will return -1

int L1 = P1.Length;

int L2 = P2.Length;

if ((L1 == 0) && (L2 == 0)) //if both zero-lenghted, distance is 0

return 0;

if (L1 == 0) //if first lenght equals 0 then distance equals L2

return L2;

if (L2 == 0) //if second lenght equals 0 then distance equals L1

return L1;

string UP1 = P1.ToUpper(); //upping cases of string 1

string UP2 = P2.ToUpper(); //upping cases of string 1

int[,] matrix = new int[L1 + 1, L2 + 1];

for (int i = 0; i <= L1; i++) //zero-row init

matrix[i, 0] = i;

for (int j = 0; j <= L2; j++) //zero-col init

matrix[0, j] = j;

for (int i = 1; i <= L1; i++) //encalculating the distance

for (int j = 1; j <= L2; j++)

{

int CharEqual = ((UP1.Substring(i - 1, 1) == UP2.Substring(j - 1, 1)) ? 0 : 1);

int InsertValue = matrix[i, j - 1] + 1; //adding

int DeleteValue = matrix[i - 1, j] + 1; //deleting

int subst = matrix[i - 1, j - 1] + CharEqual; //replacing

matrix[i, j] = Math.Min(Math.Min(InsertValue, DeleteValue), subst); //encalculating current item og the matrix

if ((i > 1) && (j > 1) && (UP1.Substring(i - 1, 1) == UP2.Substring(j - 2, 1)) &&

(UP1.Substring(i - 2, 1) == UP2.Substring(j - 1, 1)))

matrix[i, j] = Math.Min(matrix[i, j], matrix[i - 2, j - 2] + CharEqual); //Damerau addition

}

return matrix[L1, L2]; //result equals down-right item of the matrix

}

}

}

Листинг модуля формы:

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.Text;

using System.Windows.Forms;

using LevenshteinLibrary;

namespace files

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

/// <summary>

/// Word list type

/// </summary>

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

private void ReadButton\_Click(object sender, EventArgs e) //clicking on \*OPEN\* button

{

OpenFileDialog OD = new OpenFileDialog(); //new open dialog

OD.Filter = "Текстовые файлы|\*.txt"; //setting a filter

if (OD.ShowDialog() == DialogResult.OK) //if file has been chosen

{

Stopwatch sw = new Stopwatch(); //a new stopwatch

sw.Start(); //Чтение файла в виде строки //starting a stopwatch

string text = File.ReadAllText(OD.FileName); //reading whole text from the file

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

string[] array = text.Split(sep); //splitting a text by seps

foreach (string cur in array) //removing extra spaces

{

string str = cur.Trim();

if (!list.Contains(str)) //add to list if unique

list.Add(str);

}

sw.Stop(); //stopping the stopwatch

this.ReadTimeText.Text = sw.Elapsed.ToString(); //showing time of a search

this.ReadCountText.Text = list.Count.ToString(); //showing counter of words

}

else

MessageBox.Show("It's necessary to open a file!");

}

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

{

string word = this.LevenshteinText.Text.Trim(); //a word for a search

if (!string.IsNullOrEmpty(word) && list.Count > 0) //if a field wasn't empty

{

this.ResultBox.BeginUpdate(); //updating a listbox

this.ResultBox.Items.Clear(); //clearing a listbox

Stopwatch sw = new Stopwatch(); //new stopwatch

sw.Start(); //starting the stopwatch

foreach (string cur in list)

{

int curdis = LevenshteinDistance.Distance(word, cur); //encalculating a Levenshtein dist between word and str

if (string.IsNullOrEmpty(this.MaxText.Text.Trim()) || (curdis <= int.Parse(this.MaxText.Text.Trim())))

this.ResultBox.Items.Add(cur + "; Расстояние = " + curdis.ToString());

}

this.ResultBox.EndUpdate(); //ending of listbox updating

}

else

MessageBox.Show("It's necessary to choose a file and a word to search!");

}

}

}

Листинг основной программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace files

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new MainForm());

}

}

}

**Экранные формы с примерами выполнения программы (скриншоты)**

