**Отчет по лабораторной работе № 5 по курсу**

**Базовые компоненты интернет технологий.**

#### "Реализация вычислений расстояния Левенштейна с использованием алгоритма Вагнера-Фишера "

Выполнил: Олюшкин М. К.

ИУ5-31

Проверил: Гапанюк Ю. Е.

Москва, 2017

**СОДЕРЖАНИЕ**

[1. Задача лабораторной работы 3](#_Toc404062393)

[2. Текст программы. 3](#_Toc404062394)

[3. Экранные формы. 6](#_Toc404062395)

# Задача лабораторной работы

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.

# Текст программы.

* **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.Windows.Forms;

using System.IO;

using System.Diagnostics;

using System.Threading.Tasks;

namespace Laba4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

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

{

OpenFileDialog fd = new OpenFileDialog();

fd.Filter = "текстовые файлы|\*.txt";

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

{

//Чтение файла в виде строки

string text = File.ReadAllText(fd.FileName);

// Засекаем время

Stopwatch t = new Stopwatch();

t.Start();

//Разделительные символы для чтения из файла

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

string[] textArray = text.Split(separators);

foreach (string strTemp in textArray)

{

//Удаление пробелов в начале и конце строки

string str = strTemp.Trim();

//Добавление строки в список, если строка не содержится в списке

if (!list.Contains(str)) list.Add(str);

}

t.Stop();

this.textBox1.Text = t.Elapsed.ToString();

this.textBox2.Text = list.Count.ToString();

}

else

{

MessageBox.Show("Необходимо выбрать файл");

}

}

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

{

//Слово для поиска

string word = this.textBox3.Text.Trim();

//Если слово для поиска не пусто

if (!string.IsNullOrWhiteSpace(word) && list.Count > 0)

{

//Слово для поиска в верхнем регистре

string wordUpper = word.ToUpper();

//Временные результаты поиска

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

Stopwatch t = new Stopwatch();

t.Start();

foreach (string str in list)

{

if (str.ToUpper().Contains(wordUpper))

{

tempList.Add(str);

}

}

t.Stop();

this.textBox4.Text = t.Elapsed.ToString();

this.listBox1.BeginUpdate();

//Очистка списка

//this.listBox1.Items.Clear();

//Вывод результатов поиска

foreach (string str in tempList)

{

this.listBox1.Items.Add(str);

}

this.listBox1.EndUpdate();

}

else

{

MessageBox.Show("Необходимо выбрать файл и ввести слово для поиска");

}

}

}

}

# Экранные формы.



