Московский государственный технический университет

им. Н. Э. Баумана

**Отчет**

**по лабораторной работе №5**

Выполнил:

Студент группы ИУ5-35Б

Богданова В.В.

Проверил:

Преподаватель

Гапанюк Ю.Е.

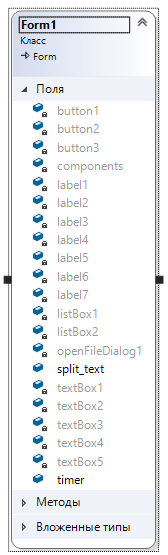
2020г.

**1). Описание 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.

**2). Диаграмма классов**



**3). Текст программы**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

using System.IO;

namespace lab4\_сишарп

{

public partial class Form1: Form

{

public static List<string> split\_text = new List<string>();

public static Stopwatch timer = new Stopwatch();

public Form1()

{

InitializeComponent();

}

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

{

OpenFileDialog ofd1 = new OpenFileDialog();

ofd1.Filter = "Text files(\*.txt)|\*.txt";

if (ofd1.ShowDialog() == DialogResult.Cancel)

{

MessageBox.Show("Необходимо выбрать файл");

return;

}

Stopwatch stopWatch = new Stopwatch();

stopWatch.Start();

// читаем файл в строку

string fileText = System.IO.File.ReadAllText(ofd1.FileName);

//добавляем разделители

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

//получаем список слов

string[] split\_text\_temp = fileText.Split(separators);

//убираем дубликаты

foreach (string str in split\_text\_temp)

{

if (!split\_text.Contains(str))

split\_text.Add(str);

}

stopWatch.Stop();

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

//this.textBoxFileReadCount.Text = list.Count.ToString();

}

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

{

//определяем слово для поиска в нижнем регистре

string word = this.textBox2.Text.Trim().ToLower();

int max\_dist = int.Parse(this.textBox4.Text);

int potok\_kol = int.Parse(this.textBox5.Text);

if (string.IsNullOrWhiteSpace(word) || split\_text.Count == 0)

{

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

return;

}

List<string> find\_list = new List<string>();

//Временные результаты поиска

Stopwatch stopWatch = new Stopwatch();

stopWatch.Start();

foreach (string str in split\_text)

{

int dist = Distance(str.ToLower(), word);

if (dist<=max\_dist)

find\_list.Add(str);

}

stopWatch.Stop();

this.textBox3.Text = stopWatch.Elapsed.ToString();

this.listBox1.BeginUpdate();

//Очистка списка

this.listBox1.Items.Clear();

//Вывод результатов поиска

foreach (string str in find\_list)

{

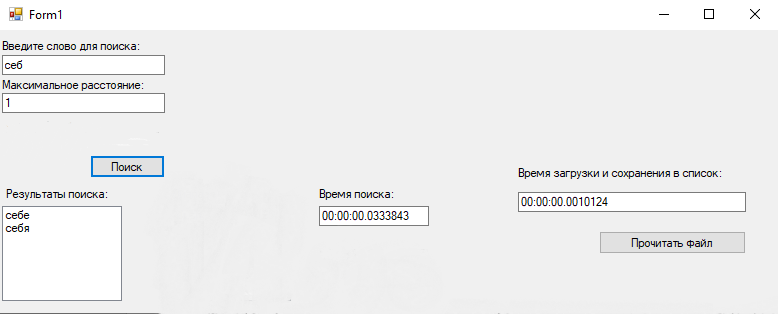
this.listBox1.Items.Add(str);

}

this.listBox1.EndUpdate();

}

**4). Пример выполнения**

****