Московский Государственный Университет им. Н.Э. Баумана

Отчет по лабораторной работе №5

по курсу БКИТ

# Выполнил:

## Болотин Андрей

## ИУ5-32

## Громов Вадим

## ИУ5-31

# Условие задачи

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.

Код

## Файл: MainWindow.Xaml.cs

using System;

using System.IO;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Microsoft.Win32;

using System.Diagnostics;

namespace Lab4

{

public partial class MainWindow : Window

{

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

public MainWindow()

{

InitializeComponent();

}

private void Read\_File\_Button\_Click(object sender, RoutedEventArgs e)

{

OpenFileDialog Dialog\_one = new OpenFileDialog();

Dialog\_one.Filter = "text\_files|\*.txt";

if (Dialog\_one.ShowDialog()== true)

{

Stopwatch mytimer = new Stopwatch();

mytimer.Start();

string text = File.ReadAllText(Dialog\_one.FileName);

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

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

foreach (string strTemp in textArray)

{

string str = strTemp.Trim();

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

}

mytimer.Stop();

this.textbox\_for\_timer.Text = mytimer.Elapsed.ToString();

this.textbox\_for\_list.Text = list.Count.ToString();

}

else

{

MessageBox.Show("Please choose file");

}

}

private void Search\_button\_Click(object sender, RoutedEventArgs e)

{

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

if (!string.IsNullOrWhiteSpace(word) && list.Count > 0 && word != "Input word you'd like to find")

{

string wordUpper = word.ToUpper();

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.Anothertimer.Text = t.Elapsed.ToString();

this.found\_words.Items.Clear();

foreach (string str in tempList)

{

this.found\_words.Items.Add(str);

}

}

else

{

MessageBox.Show("Input the word you'd like to find, please");

}

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

}

}

# Примеры работающей программы





