

Московский государственный технический

университет им. Н.Э. Баумана.

Факультет «Информатика и системы управления»

Кафедра ИУ5. Курс «Базовые компоненты интернет технологий»

Отчет по лабораторной работе № 5

«Вычисление расстояния Левенштейна с использованием алгоритма
Вагнера-Фишера»

Выполнил:

студент группы ИУ5-33

Терентьев Владислав

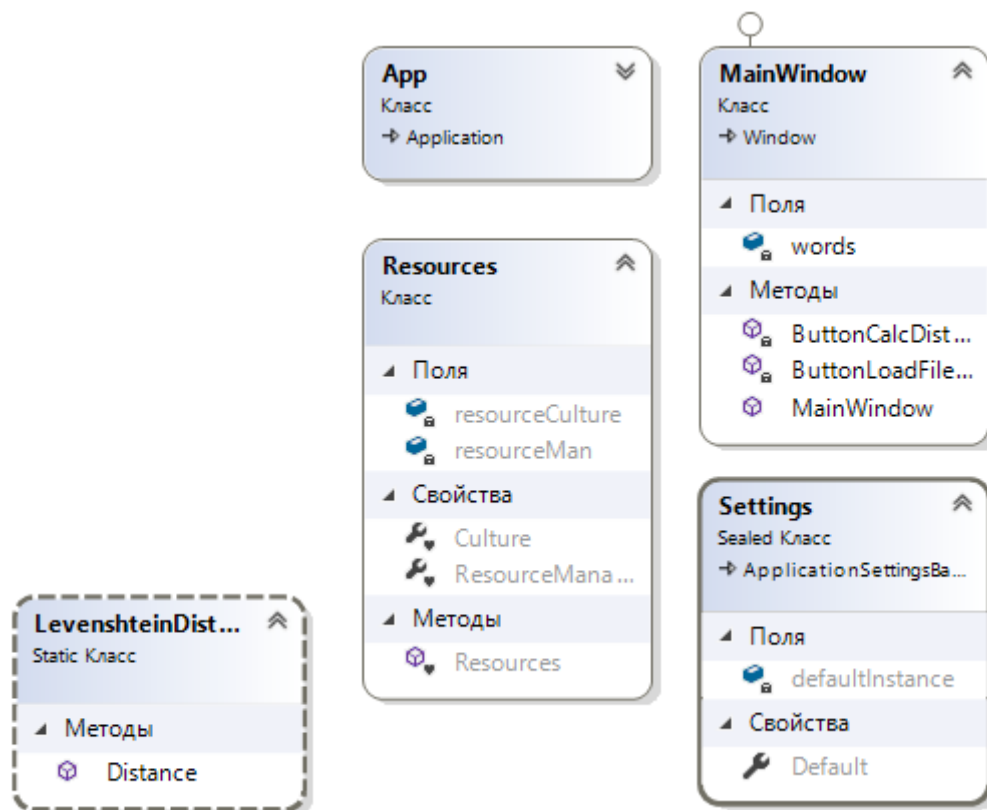
Москва, 2019 г.

1. Постановка 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 Дamerau-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

3.1. LevenshteinDistance.cs

```
using System;

namespace laba5_lib
{
    public static class LevenshteinDistance
    {
        public static int Distance(string strFirst, string strSecond)
        {

```

```

        if ((strFirst == null) || (strSecond == null)) return -1;
        int str1Len = strFirst.Length;
        int str2Len = strSecond.Length;
        if ((str1Len == 0) && (str2Len == 0)) return 0;
        if (str1Len == 0) return str2Len;
        if (str2Len == 0) return str1Len;
        string str1 = strFirst.ToUpper();
        string str2 = strSecond.ToUpper();
        if (str1 == str2) return 0;
        int[,] matrix = new int[str1Len + 1, str2Len + 1];
        for (int i = 0; i <= str1Len; i++) matrix[i, 0] = i;
        for (int j = 0; j <= str2Len; j++) matrix[0, j] = j;
        for (int i = 1; i <= str1Len; i++)
        {
            for (int j = 1; j <= str2Len; j++)
            {
                int cost = ((str1.Substring(i - 1, 1) == str2.Substring(j - 1, 1)) ?
0 : 1);

                int ins = matrix[i, j - 1] + 1;
                int del = matrix[i - 1, j] + 1;
                int subst = matrix[i - 1, j - 1] + cost;
                matrix[i, j] = Math.Min(Math.Min(ins, del), subst);
                if ((i > 1) && (j > 1) &&
                    (str1.Substring(i - 1, 1) == str2.Substring(j - 2, 1)) &&
                    (str1.Substring(i - 2, 1) == str2.Substring(j - 1, 1)))
                {
                    matrix[i, j] = Math.Min(matrix[i, j], matrix[i - 2, j - 2] +
cost);
                }
            }
        }
        return matrix[str1Len, str2Len];
    }
}
}

```

3.2. MainWindow.xaml

```

<Window x:Class="laba5_wpf.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        mc:Ignorable="d"
        Title="Терентьев Владислав ИУ5-33" Height="450" Width="800">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="25"></RowDefinition>
            <RowDefinition Height="25"></RowDefinition>
            <RowDefinition Height="35"></RowDefinition>
            <RowDefinition Height="25"></RowDefinition>
            <RowDefinition Height="25"></RowDefinition>
            <RowDefinition></RowDefinition>
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="200"></ColumnDefinition>
            <ColumnDefinition Width="Auto"></ColumnDefinition>
            <ColumnDefinition></ColumnDefinition>
        </Grid.ColumnDefinitions>
        <Button x:Name="ButtonLoadFile" Content="Чтение из файла" Grid.Column="0"
Grid.Row="0" Grid.RowSpan="2" HorizontalAlignment="Center" VerticalAlignment="Center"
Width="120" Height="30" Click="ButtonLoadFile_Click"/>
        <Label x:Name="LabelCountWords" Grid.Column="2" Grid.Row="1"
HorizontalAlignment="Left" VerticalAlignment="Center"/>
    </Grid>

```

```

        <TextBox x:Name="TextBoxCurrentWord" TextWrapping="Wrap" Grid.Column="1"
Grid.Row="2" Grid.ColumnSpan="2" Width="215" Height="20" HorizontalAlignment="Left"
VerticalAlignment="Center"/>
        <Button x:Name="ButtonCalcDistance" Content="Вычислить расстояния"
Grid.Column="0" Grid.Row="3" Grid.RowSpan="2" HorizontalAlignment="Center"
VerticalAlignment="Center" Width="140" Height="30" Click="ButtonCalcDistance_Click"/>
        <ListBox x:Name="ListBoxResult" Grid.Column="0" Grid.Row="5" Grid.ColumnSpan="3"
Margin="10"/>
        <Label x:Name="LabelTimeCalc" Grid.Column="2" Grid.Row="4"
HorizontalAlignment="Left" VerticalAlignment="Center"/>
        <Label x:Name="LabelTimer" Grid.Column="2" Grid.Row="0"
HorizontalAlignment="Left" VerticalAlignment="Center"/>
        <TextBox x:Name="TextBoxMaxDistance" TextWrapping="Wrap" Grid.Column="2"
Grid.Row="3" Width="25" HorizontalAlignment="Left" VerticalAlignment="Center"/>
        <Label Content="Время чтения из файла: " Grid.Column="1" Grid.Row="0"
HorizontalAlignment="Center" VerticalAlignment="Center"/>
        <Label Content="Количество уникальных слов в файле: " Grid.Column="1"
Grid.Row="1" HorizontalAlignment="Center" VerticalAlignment="Center"/>
        <Label Content="Слово для вычисления: " Grid.Column="0" Grid.Row="2"
HorizontalAlignment="Center" VerticalAlignment="Center"/>
        <Label Content="Время вычисления: " Grid.Column="1" Grid.Row="4"
HorizontalAlignment="Center" VerticalAlignment="Center"/>
        <Label Content="Максимальное расстояние: " Grid.Column="1" Grid.Row="3"
HorizontalAlignment="Center" VerticalAlignment="Center"/>
    </Grid>
</Window>

```

3.3. MainWindow.xaml.cs

```

using System.Collections.Generic;
using System.Windows;
using Microsoft.Win32;
using System.Diagnostics;
using System.IO;
using laba5_lib;

namespace laba5_wpf
{
    /// <summary>
    /// Логика взаимодействия для MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();

            readonly List<string> words = new List<string>();

            private void ButtonLoadFile_Click(object sender, RoutedEventArgs e)
            {
                OpenFileDialog fd = new OpenFileDialog
                {
                    Filter = "Текстовые Файлы|*.txt"
                };
                bool? result = fd.ShowDialog();
                if (result == true)
                {
                    Stopwatch timer = new Stopwatch();
                    timer.Start();
                    string text = File.ReadAllText(fd.FileName);
                    char[] separators = new char[] { ' ', '.', ',', '?', '!', '\\', '<', '>',
                    '/', '\t', '\n' };

```

```

        string[] textArray = text.Split(separators);
        foreach (string strTemp in textArray)
        {
            string str = strTemp.Trim();
            if (str != "")
            {
                if (!words.Contains(str)) words.Add(str);
            }
        }
        timer.Stop();
        LabelTimer.Content = timer.Elapsed.ToString();
        LabelCountWords.Content = words.Count.ToString();
    }
    else
    {
        MessageBox.Show("Необходимо выбрать файл");
    }
}

private void ButtonCalcDistance_Click(object sender, RoutedEventArgs e)
{
    string currentWord = TextBoxCurrentWord.Text.Trim();
    if (!string.IsNullOrEmpty(currentWord) && words.Count > 0)
    {
        if (!int.TryParse(TextBoxMaxDistance.Text.Trim(), out int maxDistance))
        {
            MessageBox.Show("Введите максимальное расстояние");
            return;
        }
        if (maxDistance < 1 || maxDistance > 5)
        {
            MessageBox.Show("Максимальное расстояние должно быть в диапазоне от 1
до 5");
            return;
        }
        List<string> tempList = new List<string>();
        Stopwatch timer = new Stopwatch();
        timer.Start();
        foreach (string str in words)
        {
            int distance = LevenshteinDistance.Distance(str, currentWord);
            if (distance <= maxDistance)
            {
                tempList.Add(str + " (расстояние: " + distance + ")");
            }
        }
        timer.Stop();
        LabelTimeCalc.Content = timer.Elapsed.ToString();
        ListBoxResult.ItemsSource = tempList;
    }
    else
    {
        MessageBox.Show("Выберите файл и введите слово для вычисления
расстояния");
    }
}
}
}
}

```

4. Анализ результатов

Терентьев Владислав ИУ5-33

Чтение из файла

Время чтения из файла: 00:00:00.0009634

Количество уникальных слов в файле: 334

Слово для вычисления: was

Вычислить расстояния

Максимальное расстояние: 5

Время вычисления: 00:00:00.0013132

Mr (расстояние: 3)
was (расстояние: 0)
the (расстояние: 3)
of (расстояние: 3)
a (расстояние: 2)
firm (расстояние: 4)
called (расстояние: 5)
which (расстояние: 4)
made (расстояние: 3)
drills (расстояние: 5)
He (расстояние: 3)
big (расстояние: 3)
heafy (расстояние: 5)

5. Ссылка на репозиторий

<https://github.com/iYroglif/newlabs>