

Московский государственный технический
университет имени Н. Э. Баумана.

Факультет “Информатика и системы управления”
Кафедра ИУ5 “Системы обработки информации и управления”

Курс “Парадигмы и конструкции языков
программирования”
Отчет по лабораторным работам №4-5.

Выполнила:

Студент группы ИУ5-31Б

Савельева Д.А

Подпись и дата:

Проверил:

Преподаватель кафедры ИУ5

Нардид А.Н.

Подпись и дата:

Москва, 2024 г.

Постановка задачи.

Часть 1.

Разработать программу, реализующую работу с файлами.

1. Программа должна быть разработана в виде приложения Windows Forms на языке C#. По желанию вместо Windows Forms возможно использование WPF.
2. Добавить кнопку, реализующую функцию чтения файла в список слов `List<string>`.
3. Для выбора имени файла используется класс `OpenFileDialog`, который открывает диалоговое окно с выбором файла. Ограничить выбор только файлами с расширением `«.txt»`.
4. Для чтения из файла рекомендуется использовать статический метод `ReadAllText()` класса `File` (пространство имен `System.IO`). Содержимое файла считывается методом `ReadAllText()` в виде одной строки, далее делится на слова с использованием метода `Split()` класса `string`. Слова сохраняются в список `List<string>`.
5. При сохранении слов в список `List<string>` дубликаты слов не записываются. Для проверки наличия слова в списке используется метод `Contains()`.
6. Вычислить время загрузки и сохранения в список с использованием класса `Stopwatch` (пространство имен `System.Diagnostics`). Вычисленное время вывести на форму в поле ввода (`TextBox`) или надпись (`Label`).
7. Добавить на форму поле ввода для поиска слова и кнопку поиска. При нажатии на кнопку поиска осуществлять поиск введенного слова в списке. Слово считается найденным, если оно входит в элемент списка как подстрока (метод `Contains()` класса `string`).
8. Добавить на форму список (`ListBox`). Найденные слова выводить в список с использованием метода `«название_списка.Items.Add()»`. Вызовы метода `«название_списка.Items.Add()»` должны находится

4

между вызовами методов `«название_списка.BeginUpdate()»` и `«название_списка.EndUpdate()»`.

9. Вычислить время поиска с использованием класса `Stopwatch`. Вычисленное время вывести на форму в поле ввода (`TextBox`) или надпись (`Label`).

Часть 2.

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.

Текст программы:

App.xaml

```
<Window x:Class="Pk_lab4.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"
        xmlns:local="clr-namespace:Pk_lab4"
        mc:Ignorable="d"
        Title="MainWindow" Height="450" Width="800"
        Background="Pink">
    <Grid Margin="0,0,0,-6">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="493*"/>
            <ColumnDefinition Width="307*"/>
        </Grid.ColumnDefinitions>

        <TextBox x:Name="aaa" Margin="269,295,55,0" VerticalAlignment="Top"
        TextAlignment="Center" Background="Deeppink" Foreground="LightPink"
        FontSize="80" FontWeight="Bold" Height="106">
            AAA
        </TextBox>
        <TextBox x:Name="lev"
        Margin="18,275,95,0" VerticalAlignment="Top" Foreground="Purple" FontWeight
        ="Bold" Height="40" AcceptsReturn="True" TextWrapping="Wrap" Grid.Column="1"
        TextChanged="lev_TextChanged">

        </TextBox>
        <Button x:Name="dist_lev" Content="Вычислить
        расстояние" IsCancel="True" Click="dist_Click" Margin="18,348,60,40"
```

```

Background="Deeppink" Foreground="LightPink" FontSize="20" FontWeight="Bold"
Grid.Column="1"/>
    <ListBox x:Name="l_words" Margin="282,113,55,168" FontSize="14"
FontStyle="Italic" Foreground="Purple" FontWeight="Bold">Список:</ListBox>
    <TextBox x:Name="tm" AcceptsReturn="True" TextWrapping="Wrap"
Margin="282,33,60,355" FontSize="14" FontStyle="Italic" Foreground="Purple"
FontWeight="Bold" Grid.ColumnSpan="2">Здесь могла бы быть ваша
реклама)))</TextBox>
    <TextBox x:Name="enter_w" AcceptsReturn="True" TextWrapping="Wrap"
Margin="18,113,75,275" TextChanged="enter_w_TextChanged" FontSize="14"
FontStyle="Italic" Foreground="Purple" FontWeight="Bold"
Grid.Column="1">Введите слово</TextBox>
    <Button x:Name="acceptButton" Content="Считать файл" IsDefault="True"
Click="acceptButton_Click" Margin="53,113,279,275"
    Background="Deeppink" Foreground="LightPink" FontSize="20"
FontWeight="Bold">
    <Button.Resources>
        <Style TargetType="Border">
            <Setter Property="CornerRadius" Value="20"/>
        </Style>
    </Button.Resources>
</Button>
    <Button x:Name="escButton" Content="Выход" IsCancel="True"
Click="escButton_Click" Margin="53,190,279,203" Background="Deeppink"
Foreground="LightPink" FontSize="20" FontWeight="Bold">
    <Button.Resources>
        <Style TargetType="Border">
            <Setter Property="CornerRadius" Value="20"/>
        </Style>
    </Button.Resources>
</Button>
    <Button x:Name="Button1" Content="Выбрать файл" IsCancel="True"
Click="button1_Click" Margin="53,33,279,355" Background="Deeppink"
Foreground="LightPink" FontSize="20" FontWeight="Bold">
    <Button.Resources>
        <Style TargetType="Border">
            <Setter Property="CornerRadius" Value="20"/>
        </Style>
    </Button.Resources>
</Button>
    <Button x:Name="find_w" Content="Найти слово" IsCancel="True"
Click="find_w_Click" Margin="18,194,128,194" Background="Deeppink"
Foreground="LightPink" FontSize="20" FontWeight="Bold" Grid.Column="1"/>
    <Canvas Margin="40,251,249,38">
        <Path Stroke="DeepPink" StrokeThickness="3"
Data="M 241,200
    A 20,20 0 0 0 200,240
    C 210,250 240,270 240,270

```

```

        C 240,270 260,260 280,240
        A 20,20 0 0 0 239,200
        " Height="141" Stretch="Fill" Width="169"
HorizontalAlignment="Left" Canvas.Left="10" VerticalAlignment="Top"/>
    </Canvas>
</Grid>
</Window>

```

App.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Threading.Tasks;
using System.Windows;

namespace Pk_lab4
{
    /// <summary>
    /// Логика взаимодействия для App.xaml
    /// </summary>
    public partial class App : Application
    {
    }
}

```

MainWindow.xaml.cs

```

using Microsoft.Win32;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
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 System.IO;

```

```

using static System.Net.Mime.MediaTypeNames;
using System.Collections.Specialized;
using System.Diagnostics;
using System.Xml.Linq;

namespace Pk_lab4
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();

            private List<string> words = new List<string>();
            private string filePath;
            private string word;
            private int dist;

            private void acceptButton_Click(object sender, RoutedEventArgs e)
            {
                if (filePath != null)
                {
                    Stopwatch time = new Stopwatch();
                    time.Start();
                    string all_text = File.ReadAllText(filePath);
                    char[] separator = {'\n', ' ', ',', '.', ':', '\t', '!', '?',
                    '-' };

                    string[] list = all_text.Split(separator);

                    foreach (string s in list)
                    {
                        if (words.Contains(s) != true)
                        {
                            words.Add(s);
                            l_words.Items.Add(new ListBoxItem() { Content = s });
                        }
                    }
                    time.Stop();
                    tm.Text = $"Время загрузки: {time.ElapsedMilliseconds} мс";
                }
            }
        }
    }
}

```

```

private void escButton_Click(object sender, RoutedEventArgs e)
{
    this.Close(); // закрытие окна
}

private void button1_Click(object sender, EventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.DefaultExt = ".txt";
    openFileDialog.Filter = "Текстовые файлы (*.txt)|*.txt";
    Nullable<bool> result = openFileDialog.ShowDialog();

    if (result == true)
    {
        filePath = openFileDialog.FileName;
    }
}

private void find_w_Click(object sender, RoutedEventArgs e)
{
    string temp = "Слово не найдено";

    foreach (string s in words)
    {
        if (s == word)
        {
            temp = $"Слово {word} найдено";
        }
    }
    enter_w.Text = temp;
}

private void enter_w_TextChanged(object sender, TextChangedEventArgs
e)
{
    word = enter_w.Text;
}

private int Levensh(string w1, string w2)
{
    int[,] matrix = new int[w1.Length+1, w2.Length+1];

```

```

        for( int i = 0; i < w1.Length+1; ++i)
        {
            for ( int j = 0; j < w2.Length + 1; ++j)
            {
                if (i == 0)
                {
                    matrix[i, j] = j;
                }
                else if (j == 0)
                {
                    matrix[i, j] = i;
                }
                else
                {
                    matrix[i, j] = 0;
                }
            }
        }

        for (int i = 1; i < w1.Length+1; ++i)
        {
            for (int j = 1; j < w2.Length+1; ++j)
            {
                if (w1[i-1] == w2[j-1])
                {
                    int[] temp_list = { matrix[i - 1, j]+1 , matrix[i, j - 1] +1, matrix[i - 1, j - 1] };
                    matrix[i, j] = temp_list.Min();
                }
                else
                {
                    int[] temp_list = { matrix[i - 1, j] + 1, matrix[i, j - 1] + 1, matrix[i - 1, j - 1] +1};
                    matrix[i, j] = temp_list.Min();
                }
            }
        }

        return matrix[w1.Length, w2.Length];
    }

    private void dist_Click(object sender, RoutedEventArgs e)
    {
        string temp = "Слово не найдено";

        foreach (string s in words)
        {
            if (Levensh(s,word) <= dist)

```



```

        {
            temp = $"Слово {s}, расстояние Левенштейна
{Levensh(s,word)}";
        }
    }
    enter_w.Text = temp;

}

private void lev_TextChanged(object sender, TextChangedEventArgs e)
{
    dist = int.Parse(lev.Text);
}
}
}

```

MainWindow.xaml

```

<Window x:Class="Pk_lab4.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"
    xmlns:local="clr-namespace:Pk_lab4"
    mc:Ignorable="d"
    Title="MainWindow" Height="450" Width="800"
    Background="Pink">
    <Grid Margin="0,0,0,-6">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="493*"/>
            <ColumnDefinition Width="307*"/>
        </Grid.ColumnDefinitions>

        <TextBox x:Name="aaa" Margin="269,295,55,0" VerticalAlignment="Top"
TextAlignment="Center" Background="Deeppink" Foreground="LightPink"
FontSize="80" FontWeight="Bold" Height="106">
            AAA
        </TextBox>
        <TextBox x:Name="lev"
Margin="18,275,95,0" VerticalAlignment="Top" Foreground="Purple" FontWeight
="Bold" Height="40" AcceptsReturn="True" TextWrapping="Wrap" Grid.Column="1"
TextChanged="lev_TextChanged">

        </TextBox>
        <Button x:Name="dist_lev" Content="Вычислить
расстояние" IsCancel="True" Click="dist_Click" Margin="18,348,60,40"

```

```

Background="Deeppink" Foreground="LightPink" FontSize="20" FontWeight="Bold"
Grid.Column="1"/>
    <ListBox x:Name="l_words" Margin="282,113,55,168" FontSize="14"
FontStyle="Italic" Foreground="Purple" FontWeight="Bold">Список:</ListBox>
    <TextBox x:Name="tm" AcceptsReturn="True" TextWrapping="Wrap"
Margin="282,33,60,355" FontSize="14" FontStyle="Italic" Foreground="Purple"
FontWeight="Bold" Grid.ColumnSpan="2">Здесь могла бы быть ваша
реклама)))</TextBox>
    <TextBox x:Name="enter_w" AcceptsReturn="True" TextWrapping="Wrap"
Margin="18,113,75,275" TextChanged="enter_w_TextChanged" FontSize="14"
FontStyle="Italic" Foreground="Purple" FontWeight="Bold"
Grid.Column="1">Введите слово</TextBox>
    <Button x:Name="acceptButton" Content="Считать файл" IsDefault="True"
Click="acceptButton_Click" Margin="53,113,279,275"
    Background="Deeppink" Foreground="LightPink" FontSize="20"
FontWeight="Bold">
    <Button.Resources>
        <Style TargetType="Border">
            <Setter Property="CornerRadius" Value="20"/>
        </Style>
    </Button.Resources>
</Button>
    <Button x:Name="escButton" Content="Выход" IsCancel="True"
Click="escButton_Click" Margin="53,190,279,203" Background="Deeppink"
Foreground="LightPink" FontSize="20" FontWeight="Bold">
    <Button.Resources>
        <Style TargetType="Border">
            <Setter Property="CornerRadius" Value="20"/>
        </Style>
    </Button.Resources>
</Button>
    <Button x:Name="Button1" Content="Выбрать файл" IsCancel="True"
Click="button1_Click" Margin="53,33,279,355" Background="Deeppink"
Foreground="LightPink" FontSize="20" FontWeight="Bold">
    <Button.Resources>
        <Style TargetType="Border">
            <Setter Property="CornerRadius" Value="20"/>
        </Style>
    </Button.Resources>
</Button>
    <Button x:Name="find_w" Content="Найти слово" IsCancel="True"
Click="find_w_Click" Margin="18,194,128,194" Background="Deeppink"
Foreground="LightPink" FontSize="20" FontWeight="Bold" Grid.Column="1"/>
    <Canvas Margin="40,251,249,38">
        <Path Stroke="DeepPink" StrokeThickness="3"
Data="M 241,200
    A 20,20 0 0 0 200,240
    C 210,250 240,270 240,270

```

```

        C 240,270 260,260 280,240
        A 20,20 0 0 0 239,200
        " Height="141" Stretch="Fill" Width="169"
HorizontalAlignment="Left" Canvas.Left="10" VerticalAlignment="Top"/>
    </Canvas>
</Grid>
</Window>

```

Результат:

