**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет технологий»

Отчет по лабораторной работе №4

«Работа с файлами»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б |  | преподаватель каф. ИУ5 |
| Федосеева Е.Ю. |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

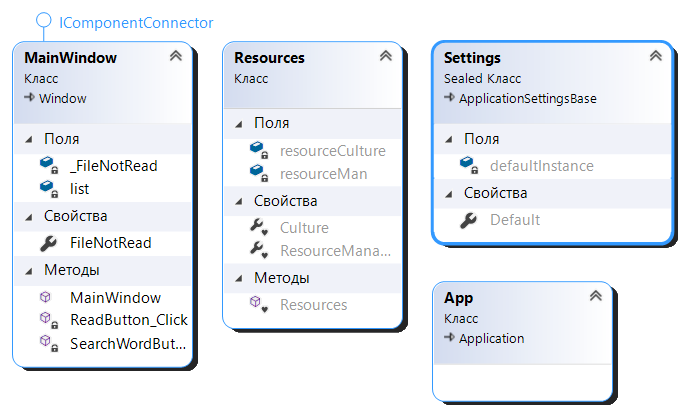
Москва, 2020 г.

**Условие задания:**

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 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).

**Диаграмма классов:**



**Текст программы:**

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;

using System.IO;

using System.Windows;

using Microsoft.Win32;

using System.Diagnostics;

using System.Threading;

namespace lab04

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

private bool \_FileNotRead = true;

public bool FileNotRead

{

get => \_FileNotRead;

set

{

\_FileNotRead = value;

}

}

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

{

OpenFileDialog fd = new OpenFileDialog();

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

if (fd.ShowDialog() == 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 strItem in textArray)

{

string str = strItem.Trim();

if (list.Contains(str) == false)

{

list.Add(str);

}

}

timer.Stop();

ReadTimeTextBox.Text = timer.Elapsed.ToString() + list.Count.ToString();

FileNotRead = false;

}

else

{

MessageBox.Show("Выберите файл!");

}

}

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

{

if (FileNotRead)

{

MessageBox.Show("Выберите файл!");

return;

}

string word = SearchWordTextBox.Text.Trim();

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

{

string wordUpper = word.ToUpper();

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

Stopwatch timer = new Stopwatch();

timer.Start();

foreach (string str in list)

{

if (str.ToUpper().Contains(word.ToUpper()))

{

tempList.Add(str);

}

}

timer.Stop();

SearchTimeTextBox.Text = timer.Elapsed.ToString();

WordList.Items.Clear();

foreach (string str in tempList)

{

WordList.Items.Add(str);

}

}

else

{

MessageBox.Show("Выберите файл и введите слово для поиска");

}

}

}

}

<Window x:Class="lab04.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:lab04"

xmlns:fa5="http://schemas.fontawesome.com/icons/"

mc:Ignorable="d"

Title="Поиск в файле" Height="500" Width="410">

<Grid Background="LightCoral">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="120"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="40"/>

<RowDefinition Height="40"/>

<RowDefinition Height="40"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<Button Width="100" Click="ReadButton\_Click" Background="MediumAquamarine" Height="20" Margin="10" VerticalAlignment="Center" HorizontalAlignment="Left">Выбор файла</Button>

<StackPanel VerticalAlignment="Center" Orientation="Horizontal" Grid.Column="1">

<Label VerticalAlignment="Center" Margin="5,0,10,0">Время чтения из файла:</Label>

<TextBox Name="ReadTimeTextBox" IsReadOnly="True" Width="110" Height="20"></TextBox>

</StackPanel>

<Label Grid.Row="1" VerticalAlignment="Center" HorizontalAlignment="Left" Margin="5">Слово для поиска:</Label>

<StackPanel Orientation="Horizontal" VerticalAlignment="Center" Grid.Row="1" Grid.Column="1">

<TextBox Name="SearchWordTextBox" HorizontalAlignment="Left" Width="110" Height="20" VerticalAlignment="Center" Margin="10"></TextBox>

<Button Name="SearchButton" Click="SearchWordButton\_Click" Background="MediumAquamarine" Height="20" Width="60" VerticalAlignment="Center">

<fa5:ImageAwesome Height="15" Icon="Solid\_Search"/>

</Button>

</StackPanel>

<Label Grid.Row="2" VerticalAlignment="Center" HorizontalAlignment="Left" Margin="5">Время поиска:</Label>

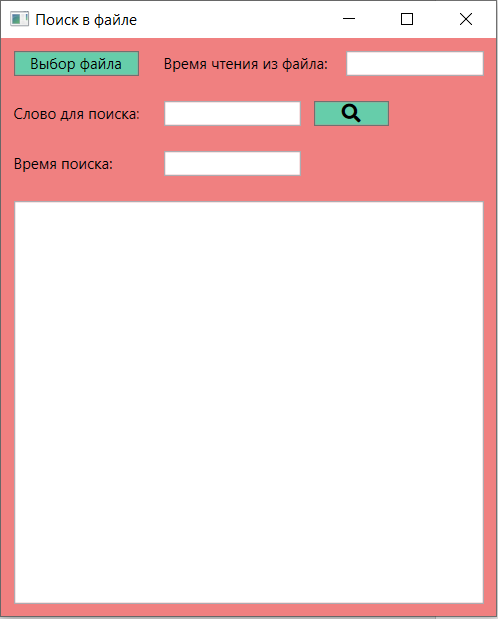
<TextBox Name="SearchTimeTextBox" IsReadOnly="True" Width="110" Height="20" Grid.Row="2" Grid.Column="1" HorizontalAlignment="Left" Margin="10"></TextBox>

<ListBox Name="WordList" Grid.ColumnSpan="2" Grid.Row="3" Margin="10"></ListBox>

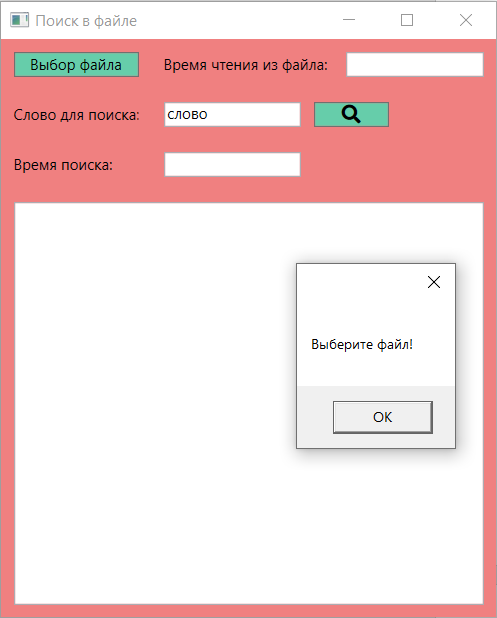
</Grid>

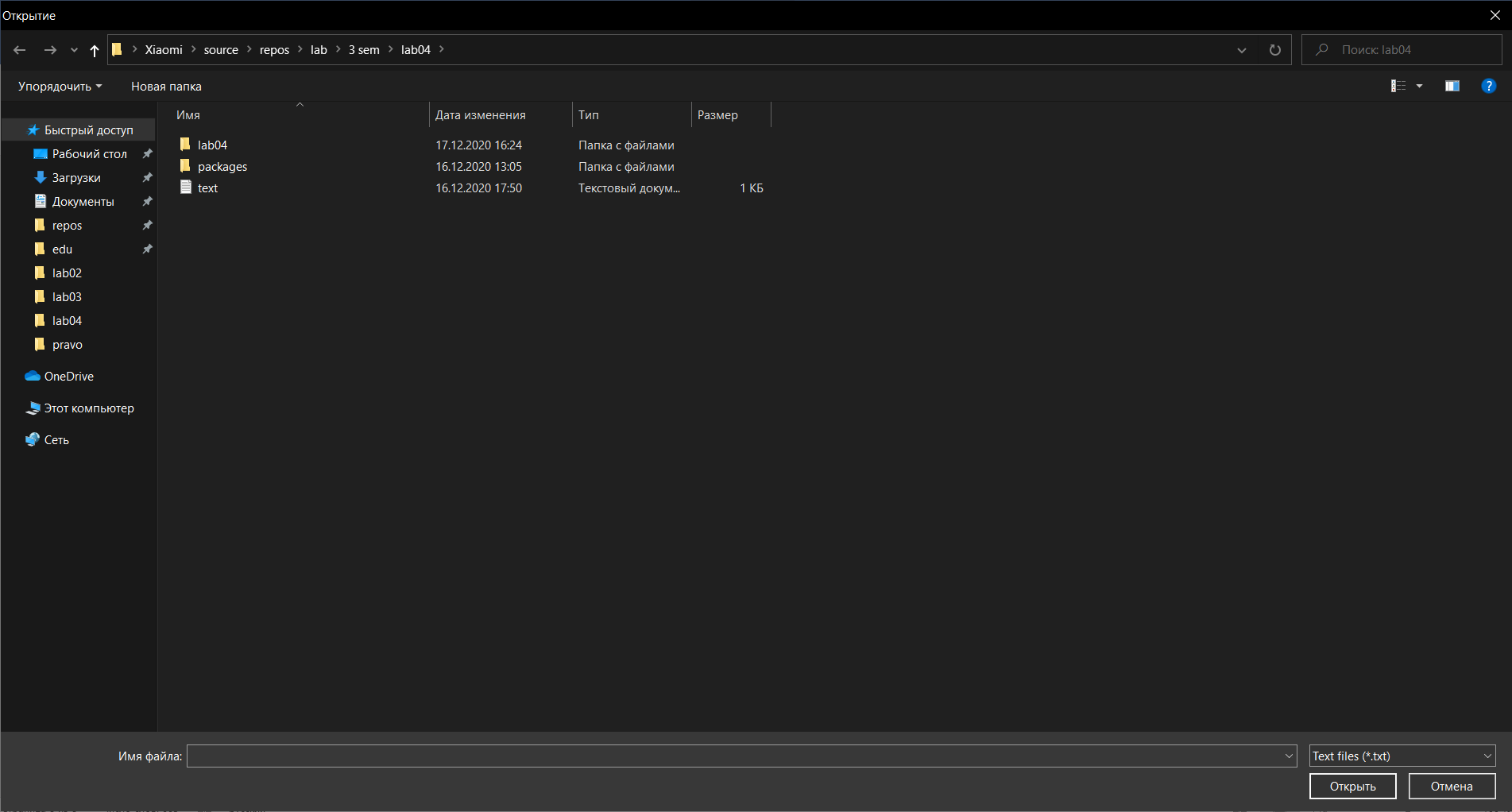
</Window>

**Результаты:**

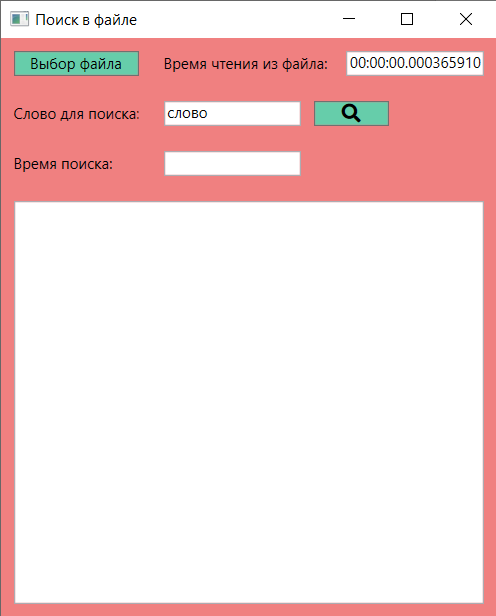


Если попытаться найти слово, не загрузив файл, возникнет предупреждение.



Нажали на кнопку «Выбор файла»: 

Успешно открыли файл.



Рабочий поиск слова:

