**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Радиотехнический»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет технологий»

Отчет по лабораторной работе №4

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы РТ5-31Б |  | преподаватель каф. ИУ5 |
| Алиев Тимур |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

# Описание 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.IO;

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.Diagnostics;

using System.Threading;

namespace lab4

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private string fileName;

private string readText;

private List<string> ListWords = new List<string>();

private List<string> FindWords = new List<string>();

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

{

Microsoft.Win32.OpenFileDialog dlg = new Microsoft.Win32.OpenFileDialog();

dlg.FileName = "Document"; // Имя файла по умолчанию

dlg.DefaultExt = ".txt"; // Расширение файла по умолчанию

dlg.Filter = "Text documents (.txt)|\*.txt"; // Фильтр файлов по расширению

// Показать диалоговое окно открытия файла

bool? result = dlg.ShowDialog();

// Обработка результатов диалогового окна открытия файла

if (result == true)

{

// Open document

fileName = dlg.FileName;

path\_file.Text = fileName;

}

}

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

{

Stopwatch stopWatch = new Stopwatch();

stopWatch.Start();

// Приложение выполняется мгновенно, поэтому искусственная задержка

Thread.Sleep(1000);

if (fileName == null) {

path\_file.Text = "Укажите путь к файлу";

return;

}

else {

readText = File.ReadAllText(fileName);

}

string[] words = readText.Split(' ', ',', '.', '!', '?');

foreach (string word in words)

{

if (!ListWords.Contains(word))

{

ListWords.Add(word);

}

}

main\_content.ItemsSource = ListWords;

stopWatch.Stop();

// Получите прошедшее время как значение TimeSpan.

TimeSpan ts = stopWatch.Elapsed;

// Отформатируйте и отобразите значение TimeSpan.

string elapsedTime = String.Format("{0:00}:{1:00}:{2:00}.{3:00}",

ts.Hours, ts.Minutes, ts.Seconds,

ts.Milliseconds / 10);

Console.WriteLine("RunTime " + elapsedTime);

elapsed\_time.Text = elapsedTime;

}

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

{

if (find\_field.Text == null)

{

return;

}

if (ListWords.Contains(find\_field.Text))

{

FindWords.Add(find\_field.Text);

find\_words.ItemsSource = FindWords;

find\_field.Text = "";

}

else

{

find\_field.Text = "";

}

}

}

}

# Результат работы программы

