**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

«Чтение и запись текстовых файлов. Основы разработки пользовательского интерфейса и поиска в текстовом файле с использованием технологии Windows Forms.»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-33 |  |  |
| Миронова Александра |  |  |
| Подпись и дата:  18.11.20 |  | Подпись и дата: |

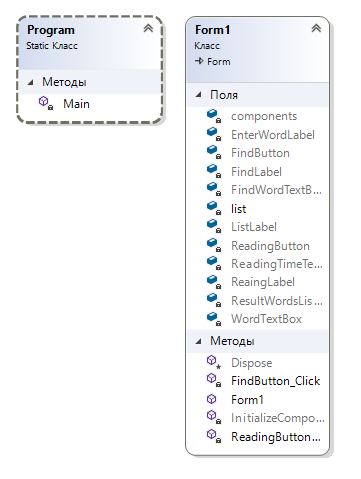
Москва, 2020 г.

**Задание:**

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:**

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.IO;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Work\_with\_files2

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

///

}

}

}

Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.IO;

using System.Diagnostics;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Work\_with\_files2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

List<string> list = new List<string>(); // Лист для записи слов в из файла

private void ReadingButton\_Click(object sender, EventArgs e)

{

OpenFileDialog fd = new OpenFileDialog(); fd.Filter = "текстовые файлы|\*.txt";

if (fd.ShowDialog() == DialogResult.OK)

{

Stopwatch t = new Stopwatch(); // Таймер

t.Start();

string text = File.ReadAllText(fd.FileName); // Чтение файла в строку

string[] textWords = text.Split(new char[] { ' ', ',', '.', '"', ':', '[', ']', '!', '?', '/', '\t','\n' }, StringSplitOptions.RemoveEmptyEntries); //Разделительные символы чтения файла

foreach (string strTemp in textWords) // Удаление пробелов в начале и конце строки

{

string str = strTemp.Trim();

if (!list.Contains(str)) // Добавление строки в список, если строка не содержится в списке

{

list.Add(str);

}

}

t.Stop();

this.ReadingTimeTextBox.Text = t.Elapsed.ToString();

}

else

{

MessageBox.Show("Необходимо выбрать файл");

}

}

private void FindButton\_Click(object sender, EventArgs e)

{

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

if (!string.IsNullOrWhiteSpace(word) && list.Count > 0) // Проверка пустого слова

{

string wordUpper = word.ToUpper(); // Поиск в верхнем регистре

List<string> tempList = new List<string>(); // Временные результаты поиска

Stopwatch t = new Stopwatch();

t.Start();

foreach (string str in list)

{

if (str.ToUpper().Contains(wordUpper))

{

tempList.Add(str);

}

}

t.Stop();

// Заполнение списка

this.FindWordTextBox.Text = t.Elapsed.ToString();

this.ResultWordsListBox.BeginUpdate();

this.ResultWordsListBox.Items.Clear();

foreach (string str in tempList) // Вывод результатов поиска

{

this.ResultWordsListBox.Items.Add(str);

}

this.ResultWordsListBox.EndUpdate();

}

else

{

MessageBox.Show("Необходимо выбрать файл и ввести слово для поиска");

}

}

}

}

**Экранные формы с примерами выполнения программы:**

