

**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:

студент группы ИУ5-33
Сергеев МЮ

Подпись и дата:
29.12.20

Проверил:

Подпись и дата:

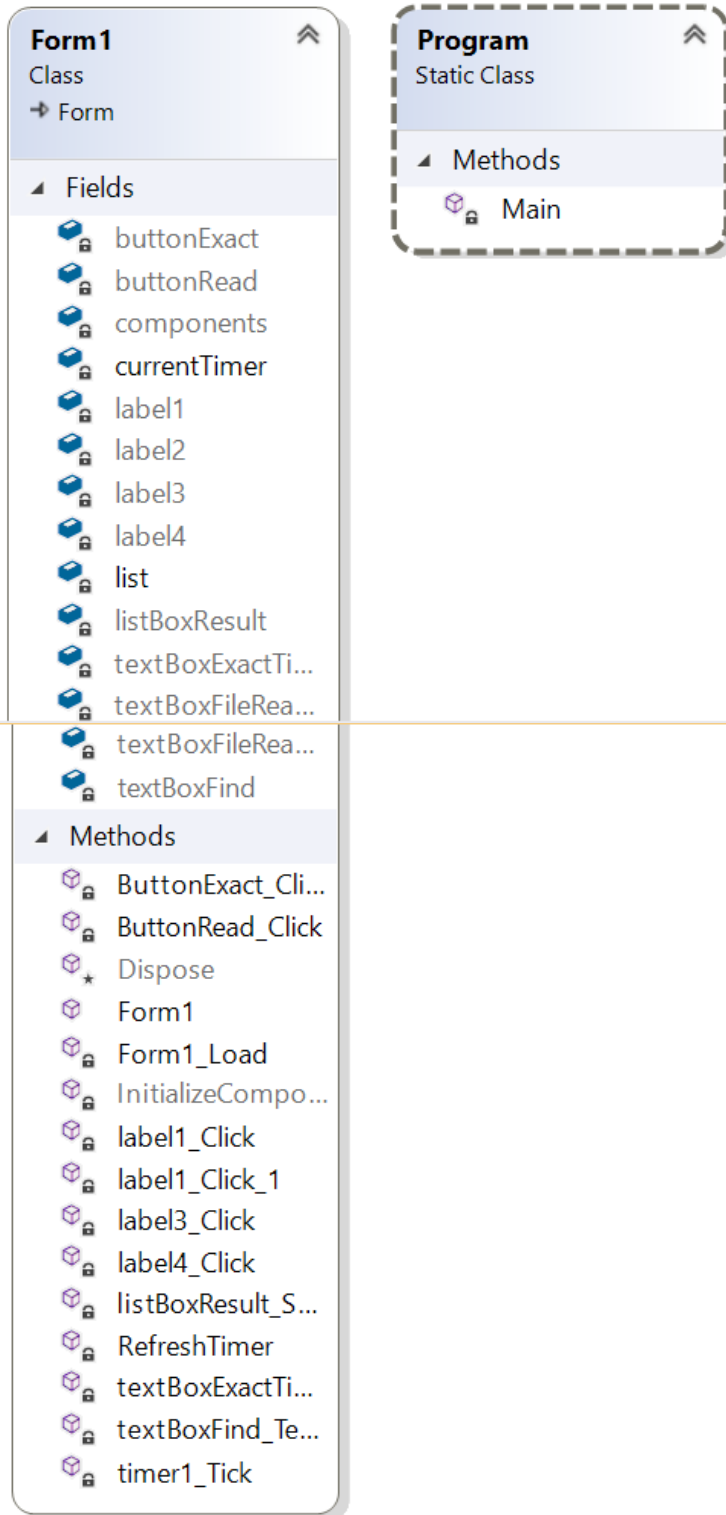
Москва, 2020 г.

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 находиться 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.Windows.Forms;

namespace Lab4
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
Form1.cs
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.IO;
using System.Windows.Forms;
namespace Lab4
{
    public partial class Form1 : Form
    {
        public Form1()
        { InitializeComponent(); }
        List<string> list = new List<string>();
        // Список слов
        private void ButtonRead_Click(object sender, EventArgs e)
        {
            OpenFileDialog fd = new OpenFileDialog();
            fd.Filter = "text files|*.txt";
            if (fd.ShowDialog() == DialogResult.OK)
            {
                Stopwatch t = new Stopwatch();
                t.Start();
                //Чтение файла в виде строки
                string text = File.ReadAllText(fd.FileName);
                //Разделительные символы для чтения из файла
                char[] separators = new char[] { ' ', '!', '?', '/', '\t', '\n' };
            }
        }
    }
}

```

```

string[] textArray = text.Split(separators);
foreach (string strTemp in textArray)
{
    //Удаление пробелов в начале и конце строки
    string str = strTemp.Trim();
    //Добавление строки в список, если строка не содержится в списке
    if (!list.Contains(str)) list.Add(str);
}
t.Stop();
textBoxFileReadTime.Text = t.Elapsed.ToString();
textBoxFileReadCount.Text = list.Count.ToString();
}
else { MessageBox.Show("Необходимо выбрать файл"); }
}
///Текущее состояние таймера
TimeSpan currentTimer = new TimeSpan();
///<summary>
///Обновление текущего состояния таймера
///</summary>
private void RefreshTimer()
{
    //Обновление поля таймера в форме
    textBoxFileReadTime.Text = currentTimer.ToString();
}
private void label1_Click(object sender, EventArgs e) { }
private void label3_Click(object sender, EventArgs e) { }
private void ButtonExact_Click(object sender, EventArgs e)
{
    //Слово для поиска
    string word = textBoxFind.Text.Trim();
    //Если слово для поиска не пусто
    if (!string.IsNullOrEmpty(word) && list.Count > 0)
    {
        //Слово для поиска в верхнем регистре
        string wordUpper = word.ToUpper();
        //Временные результаты поиска
        List<string> tempList = new List<string>();
        Stopwatch t_search = new Stopwatch();
        t_search.Start();
        foreach (string str in list) { if (str.ToUpper().Contains(wordUpper)) {
tempList.Add(str); } }
        t_search.Stop();
        textBoxExactTime.Text = t_search.Elapsed.ToString();
        listBoxResult.BeginUpdate();
        //Очистка списка
        listBoxResult.Items.Clear();
    }
}

```

```

        //Вывод результатов поиска
        foreach (string str in tempList)
        {
            listBoxResult.Items.Add(str);
        }
        listBoxResult.EndUpdate();
    }
    else
    {
        MessageBox.Show("Необходимо выбрать файл и ввести слово для поиска");
    }
}

private void listBoxResult_SelectedIndexChanged(object sender, EventArgs e) { }
private void label4_Click(object sender, EventArgs e) { }
private void timer1_Tick(object sender, EventArgs e)
{
    //Добавление к текущему состоянию таймера интервала в одну секунду
    currentTimer = currentTimer.Add(new TimeSpan(0, 0, 1));
    //Обновление текущего состояния таймера
    RefreshTimer();
}
private void textBoxFind_TextChanged(object sender, EventArgs e) { }
private void textBoxExactTime_TextChanged(object sender, EventArgs e)
{
}
private void label1_Click_1(object sender, EventArgs e)
{
}
private void Form1_Load(object sender, EventArgs e)
{
    //Обновление текущего состояния таймера
    RefreshTimer();
}
}
}

```

Экранные формы с примерами выполнения программы:

The image displays two screenshots of a Windows application window titled "Form1".

Top Screenshot (Initial State):

- Buttons: "Чтение из файла", "Поиск".
- Labels and Input Fields:
 - "Время чтения из файла" with an empty text box.
 - "Количество разных слов" with an empty text box.
 - "Слово для поиска" with an empty text box.
 - "Время поиска" with an empty text box.
- List Box: Labeled "listBoxResult", currently empty.

Bottom Screenshot (After Execution):

- Buttons: "Чтение из файла", "Поиск" (highlighted with a blue border).
- Labels and Input Fields:
 - "Время чтения из файла" with the value "00:00:00.0004409".
 - "Количество разных слов" with the value "94".
 - "Слово для поиска" with the value "море".
 - "Время поиска" with the value "00:00:00.0000601".
- List Box: Labeled "listBoxResult", containing the word "морей".

Lab4.txt - Notepad

File Edit Format View Help

Из тумана проступают очертания, корабль принимает форму
И тишина моря вот-вот обернётся в шторм
Знак мощи, демонстрация силы
Поднимите якорь, линкор прокладывает свой курс

Гордость нации, стальной зверь
Бисмарк в движении, король океана

Он был создан править волнами семи морей
Вести машину войны
Править волнами и возглавить Кригсмарине
Ужас морей,
Бисмарк и Кригсмарине

Две тысячи человек и пятьдесят тысяч тонн стали
Установить курс на Атлантику с союзниками на хвосте,
Огневая мощь, перестрелка
Боевые станции, держите цели в поле зрения!

В строй, охота началась,
Смерть и проклятие, флот на подходе

Он был создан править волнами семи морей
Вести машину войны
Править волнами и возглавить Кригсмарине
Ужас морей,
Бисмарк и Кригсмарине

На дне океана, в глубинах бездны,
Они связаны железом и кровью,
Флагман флота, ужас морей,
Его орудия наконец умолкли.

Гордость нации, стальной зверь,

Ln 1, Col 1 100% Windows (CRLF) UTF-8