

**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:

студент группы РТ5-31Б
Борисочкин М.И.

Подпись и дата:

Проверил:

к.т.н., доцент
Гапанюк Ю. Е.

Подпись и дата:

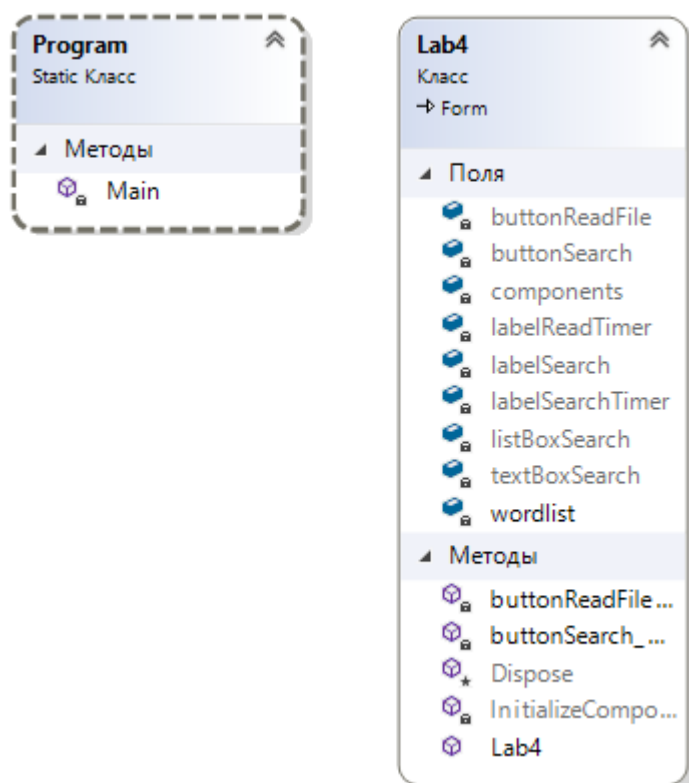
Москва, 2020 г.

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 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

Lab4.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Diagnostics;

namespace Lab4
{
    public partial class Lab4 : Form
    {
        public Lab4()
        {
            InitializeComponent();

            /// <summary>
            /// Список со словами
            /// </summary>
            List<string> wordlist = new List<string>();

            /// <summary>
            /// Чтение файла
            /// </summary>
            private void buttonReadFile_Click(object sender, EventArgs e)
            {
                OpenFileDialog fd = new OpenFileDialog();
                fd.Filter = "Текстовые файлы|*.txt";

                if (fd.ShowDialog() == DialogResult.OK)
                {
                    // Запуск таймера
                    Stopwatch tmr = new Stopwatch();
                    tmr.Start();

                    //Чтение файла в виде строки
                    string ftext = File.ReadAllText(fd.FileName);

                    // Разделители
                    char[] separators = new char[] { ' ', '.', ',', '!', '?', '/', '\\t', '\\n' };

                    string[] ftetxArray = ftext.Split(separators);

                    foreach (string strTemp in ftetxArray)
                    {
                        // Удаление пробелов в начале и в конце строки
                        string str = strTemp.Trim();

                        // Добавление строки в список, если её нет с списке
                        if (!wordlist.Contains(str))
                            wordlist.Add(str);
                    }
                }
            }
        }
    }
};
```

```

        // Остановка таймера
        tmr.Stop();
        this.labelReadTimer.Text = "Время чтения: " + tmr.Elapsed.ToString();
    }
    else
        MessageBox.Show("Необходимо выбрать файл", "Предупреждение",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }

    /// <summary>
    /// Поиск подстрок
    /// </summary>
    private void buttonSearch_Click(object sender, EventArgs e)
    {
        // Слово для поиска
        string word = this.textBoxSearch.Text.Trim();

        // Если слово для поиска не пусто
        if(!string.IsNullOrEmpty(word) && wordlist.Count > 0)
        {
            // Слово для поиска в верхнем регистре
            string wordUpper = word.ToUpper();

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

            // Запуск таймера
            Stopwatch tmr = new Stopwatch();
            tmr.Start();

            foreach(string str in wordlist)
            {
                if (str.ToUpper().Contains(wordUpper))
                    tempList.Add(str);
            }

            // Остановка таймера
            tmr.Stop();
            this.labelSearchTimer.Text = "Время чтения: " + tmr.Elapsed.ToString();

            this.listBoxSearch.BeginUpdate();

            // Очистка списка
            this.listBoxSearch.Items.Clear();

            //Вывод результатов поиска
            foreach(string str in tempList)
            {
                this.listBoxSearch.Items.Add(str);
            }

            this.listBoxSearch.EndUpdate();
        }
        else
            MessageBox.Show("Необходимо выбрать слово для поиска или считать файл",
            "Предупреждение", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

```

Примеры выполнения программы

Скриншот № 1

