

**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

“ Разработка программы, реализующую работу с файлами”

Выполнил:
студент группы ИУ5-
33Б
Ларионова Амина
Подпись и дата:
15.12.20

Проверил:

Подпись и дата:

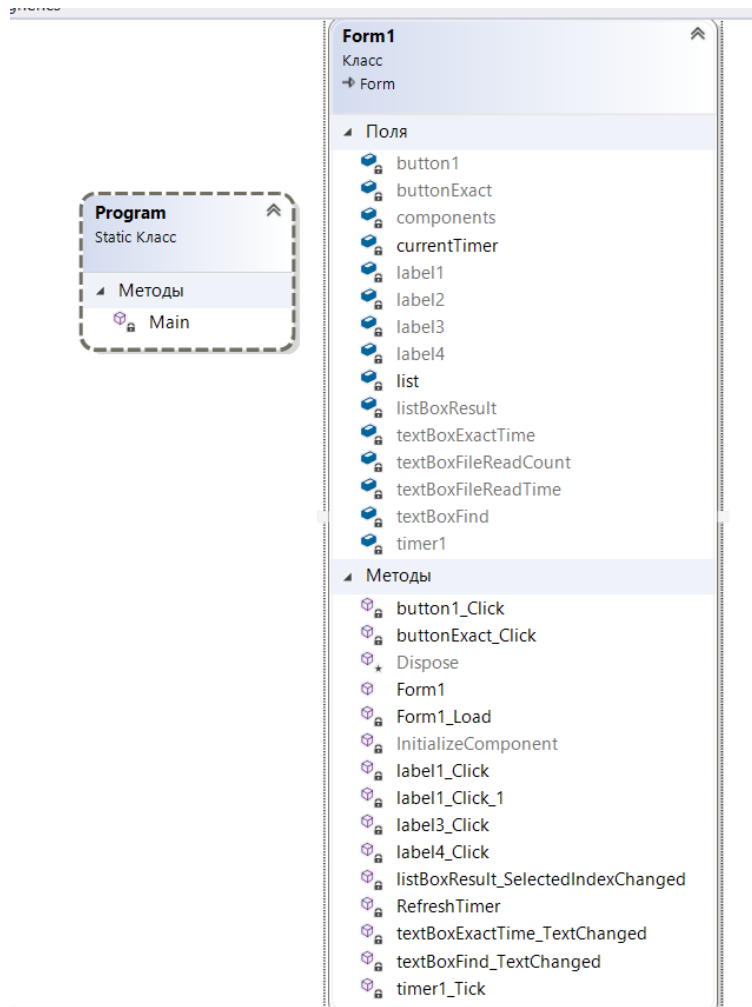
Москва, 2020 г.

1) 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`).

2) Диаграмма классов



3) Текст программы

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Лабораторная___4
{
    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.Diagnostics;
using System.IO;
using System.Windows.Forms;

namespace Лабораторная___4
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            List<string> list = new List<string>(); // Список слов

            private void button1_Click(object sender, EventArgs e)
            {
                OpenFileDialog fd = new OpenFileDialog();
                fd.Filter = "текстовые файлы|*.txt";

                if (fd.ShowDialog() == DialogResult.OK)
                {
                    Stopwatch t_load = new Stopwatch();
                    t_load.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_load.Stop();
                    this.textBoxFileReadTime.Text = t_load.Elapsed.ToString();
                    this.textBoxFileReadCount.Text = list.Count.ToString();
                    MessageBox.Show("Файл успешно прочитан");
                }
            }

            /// Текущее состояние таймера
            /// </summary>
            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 = this.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();
        this.textBoxExactTime.Text = t_search.Elapsed.ToString();
        this.listBoxResult.BeginUpdate();

        //Очистка списка
        this.listBoxResult.Items.Clear();
        //Вывод результатов поиска
        foreach (string str in tempList)
        {
            this.listBoxResult.Items.Add(str);
        }
        this.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)
{
}

```

```

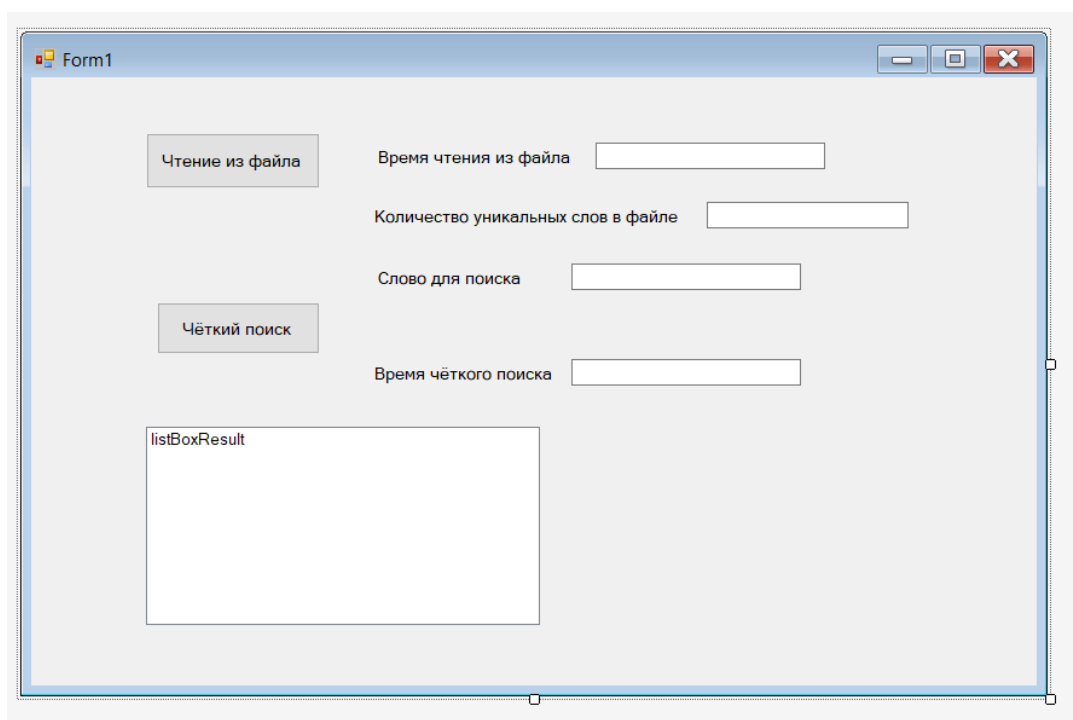
    }

    private void label1_Click_1(object sender, EventArgs e)
    {
    }

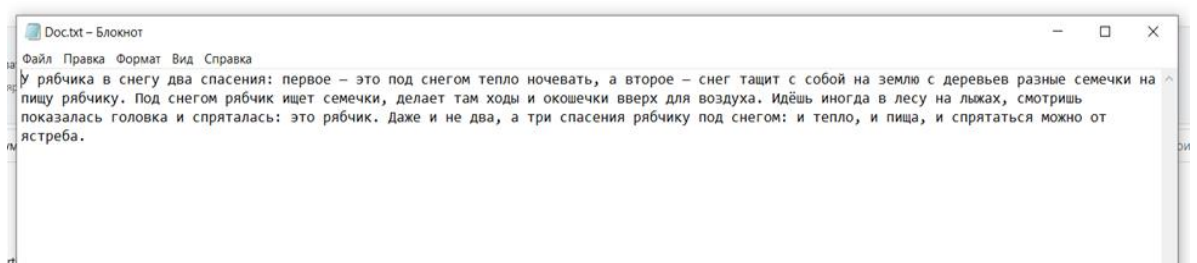
    private void Form1_Load(object sender, EventArgs e)
    {
        //Обновление текущего состояния таймера
        RefreshTimer();
    }
}
}

```

Form1.cs[Конструктор]



4) Экранные формы с примерами выполнения программы



Form1

Чтение из файла

Время чтения из файла 00:00:00.0021882

Количество уникальных слов в файле 56

Слово для поиска

Чёткий поиск

Время чёткого поиска

Файл успешно прочитан

OK

Form1

Чтение из файла

Время чтения из файла 00:00:00.0021882

Количество уникальных слов в файле 56

Слово для поиска рябчик

Чёткий поиск

Время чёткого поиска 00:00:00.0000412

рябчика
рябчику
рябчик