



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**Базовые компоненты интернет технологий
Отчет по лабораторной работе №4**

Студент: Дубянский А.И.
Группа: ИУ5Ц-51Б

Преподаватель: Гапанюк Ю. Е.

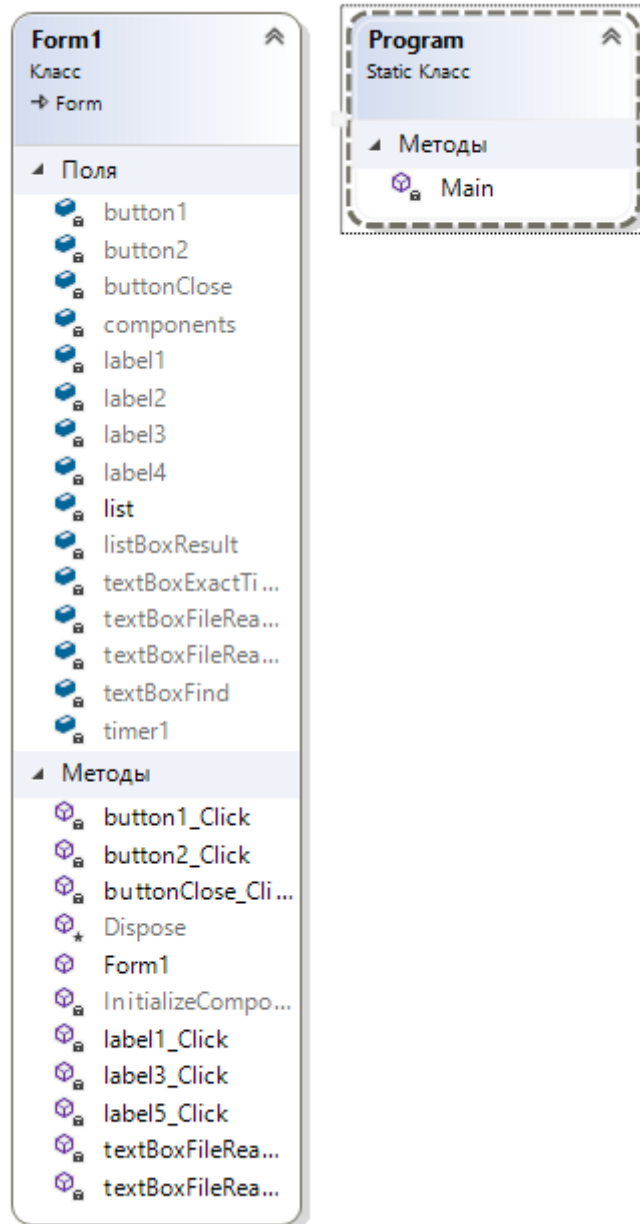
2019 г.

Лабораторная работа №4

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

namespace Dub_Lab_4
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Console.WriteLine("Лабораторная работа №4");
            Console.Title = "Дубянский А. И., ИУ5Ц-51Б";
            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.Text;
using System.Windows.Forms;
using System.IO;
using System.Diagnostics;

namespace Dub_Lab_4
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        /// <summary>
        /// Список слов
        /// </summary>
        List<string> list = new List<string>();

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void label3_Click(object sender, EventArgs e)
        {

        }
    }
}
```

```

    }

    private void label5_Click(object sender, EventArgs e)
    {

    }

    private void buttonClose_Click_1(object sender, EventArgs e)
    {
        this.Close();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        OpenFileDialog fd = new OpenFileDialog();
        fd.Filter = "текстовые файлы|.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();
            this.textBoxFileReadTime.Text = t.Elapsed.ToString();
            this.textBoxFileReadCount.Text = list.Count.ToString();
        }
        else
        {
            MessageBox.Show("Необходимо выбрать файл");
        }
    }

    private void button2_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 = new Stopwatch();
            t.Start();
            foreach (string str in list)
            {
                if (str.ToUpper().Contains(wordUpper))
                {
                    tempList.Add(str);
                }
            }
        }
    }
}

```

```

        }
    }
    t.Stop();
    this.textBoxExactTime.Text = t.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");
}
}

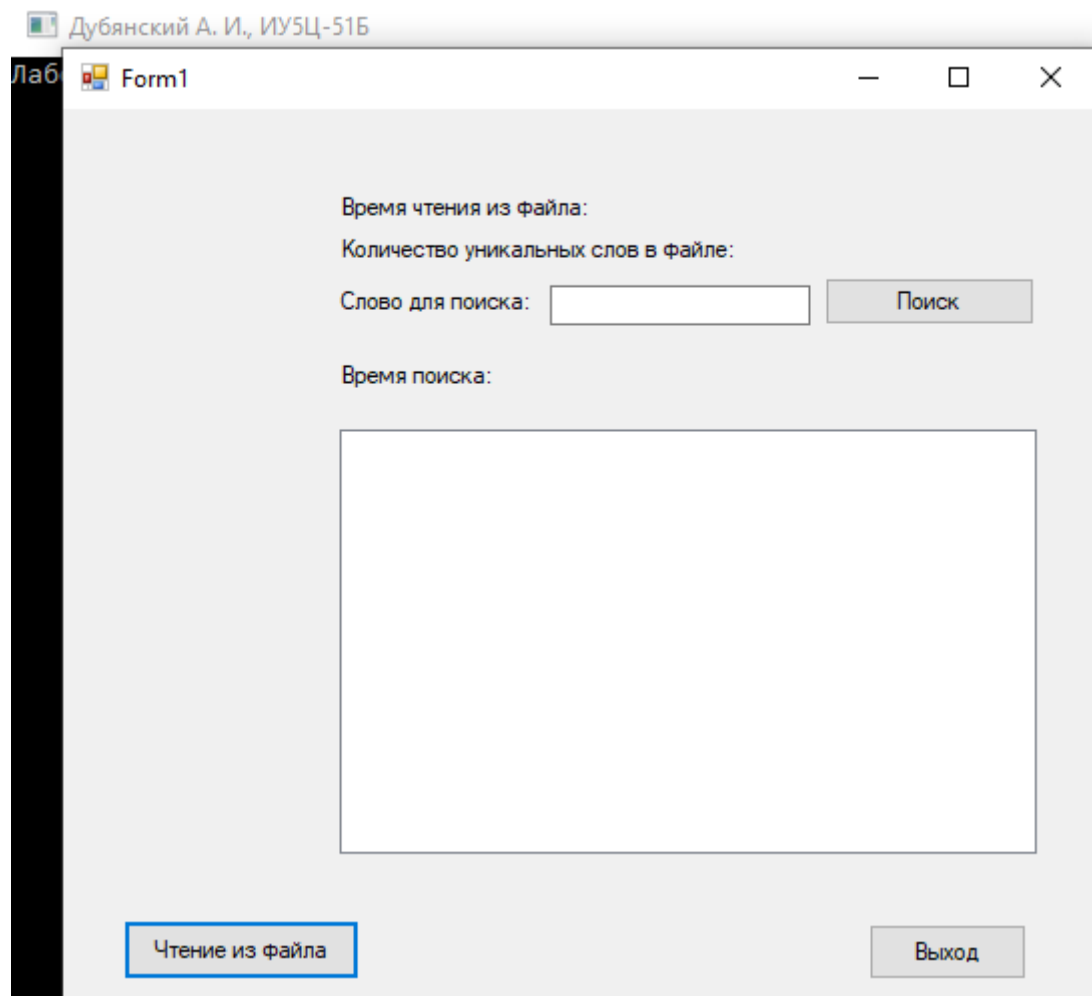
private void textBoxFileReadCount_TextChanged(object sender, EventArgs e)
{
}

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

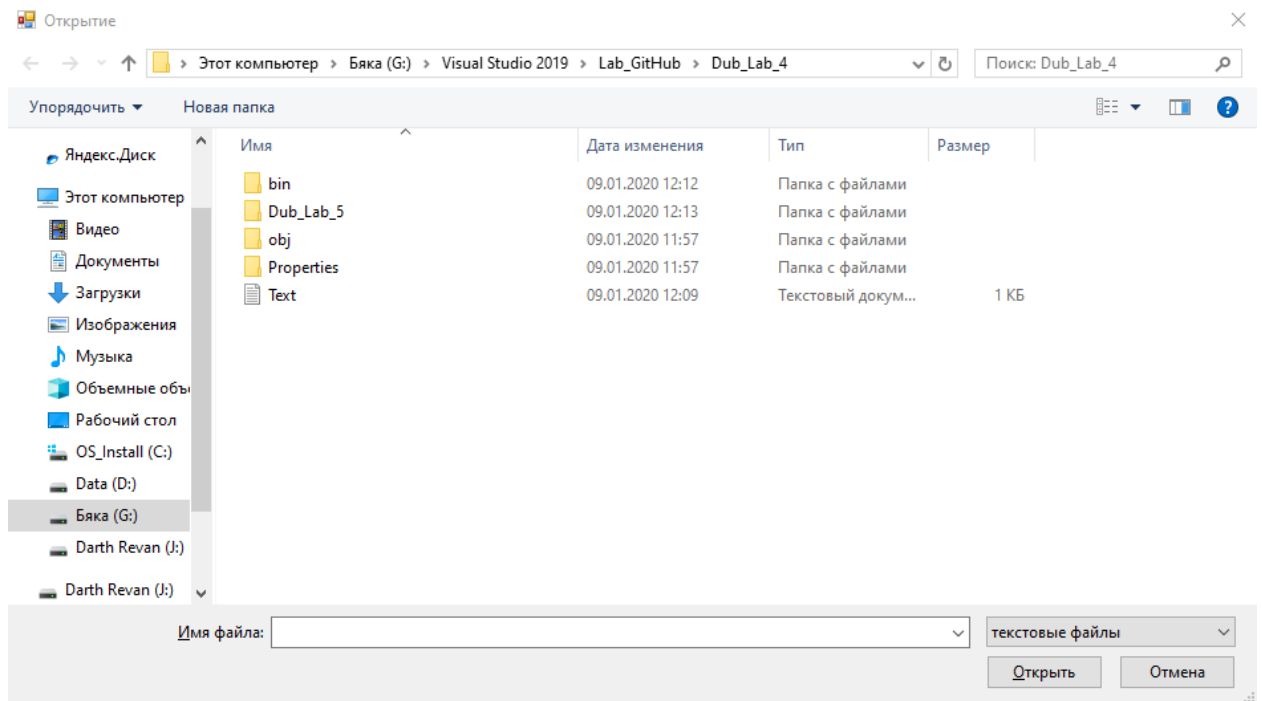
```

Тест программы

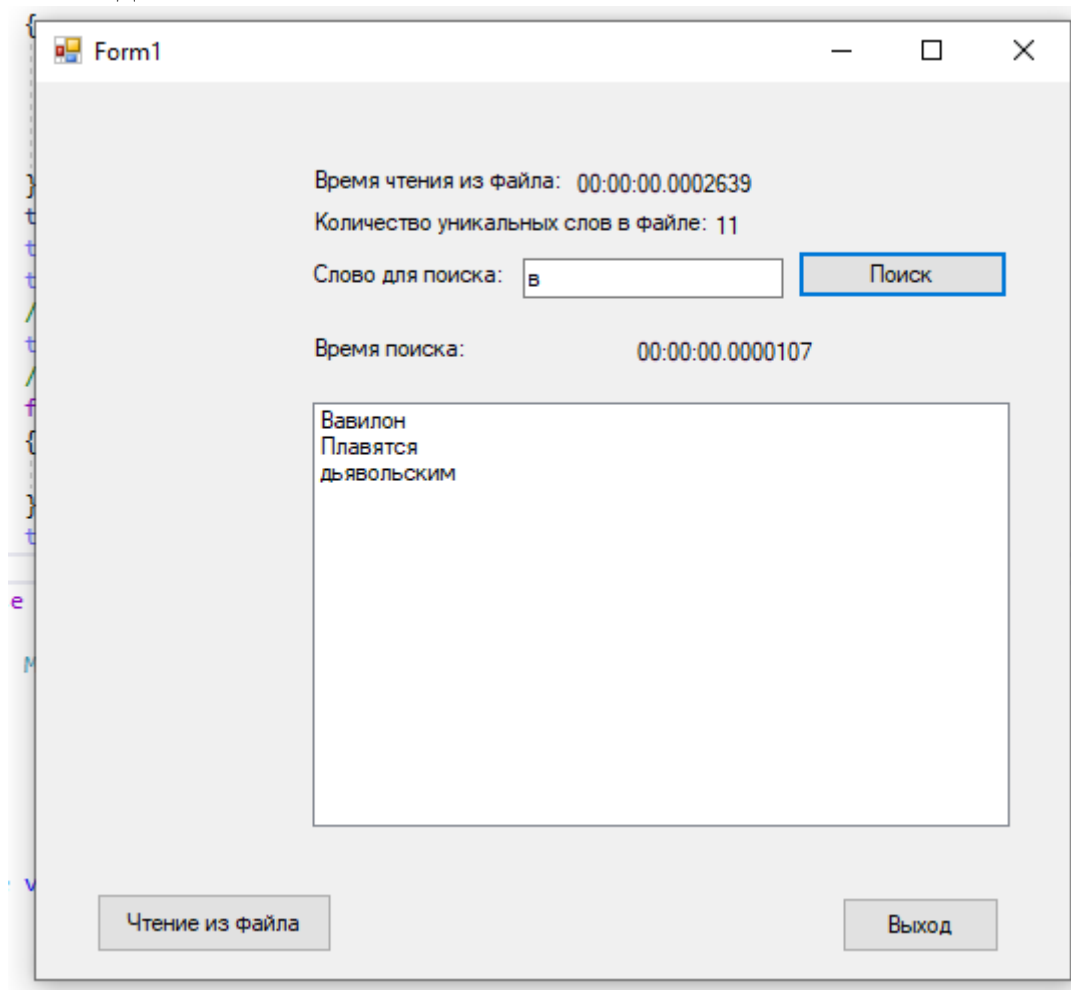
Окно приложения при первоначальном запуске



Диалоговое окно открытия файла для чтения



Слово для поиска



Пустой ввод:

The screenshot shows a Windows application window titled "Form1". Inside the window, there is a search interface with the following elements:

- Labels: "Время чтения из файла:", "Количество уникальных слов в файле:", "Слово для поиска:", and "Время поиска:".
- Input fields: A text box for "Слово для поиска:" and a larger text box for "Время поиска:".
- Buttons: A blue "Поиск" button next to the search word input, and "Чтение из файла" and "Выход" buttons at the bottom.

A modal dialog box is displayed in the center, containing the message "Необходимо выбрать файл и ввести слово для поиска" and an "ОК" button. The dialog box has a close button (X) in the top right corner.

Ссылка на репозиторий исходных кодов GitHub

https://github.com/VolandAID/Dub_Lab_4