



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)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

О Т Ч Е Т

по лабораторной работе № 5

Название: Файлы

Дисциплина: Разработка приложений на языке C#

Студент

ИУ6-75Б

(Группа)

(Подпись, дата)

И.Ю. Жосан

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

А.М. Минитаева

(И.О. Фамилия)

Москва, 2023

Цель работы: изучить основы работы с текстовыми файлами на языке программирования C#.

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 (Windows Presentation Foundation).

2. Добавить кнопку, реализующую функцию чтения текстового 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 `List<string>`.

5. При сохранении слов в список 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 `TextBox` или `Label`.

Выполнение задания:

Для выполнения поставленной задачи была создана форма (приведена на рисунке 1), позволяющая открыть файл и найти в нем заданные слова.

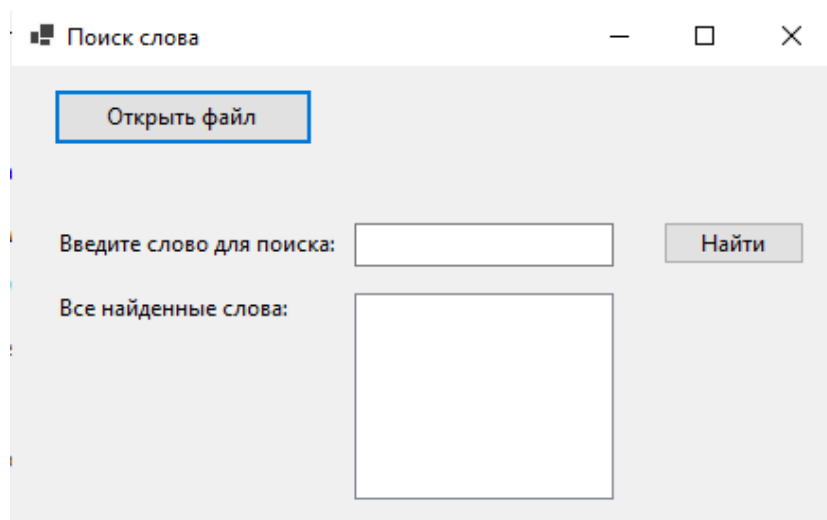


Рисунок 1 – Форма приложения

При нажатии на кнопку «Открыть файл» открывается диалоговое окно, в котором можно выбрать файл только с расширением txt (рисунок 2).

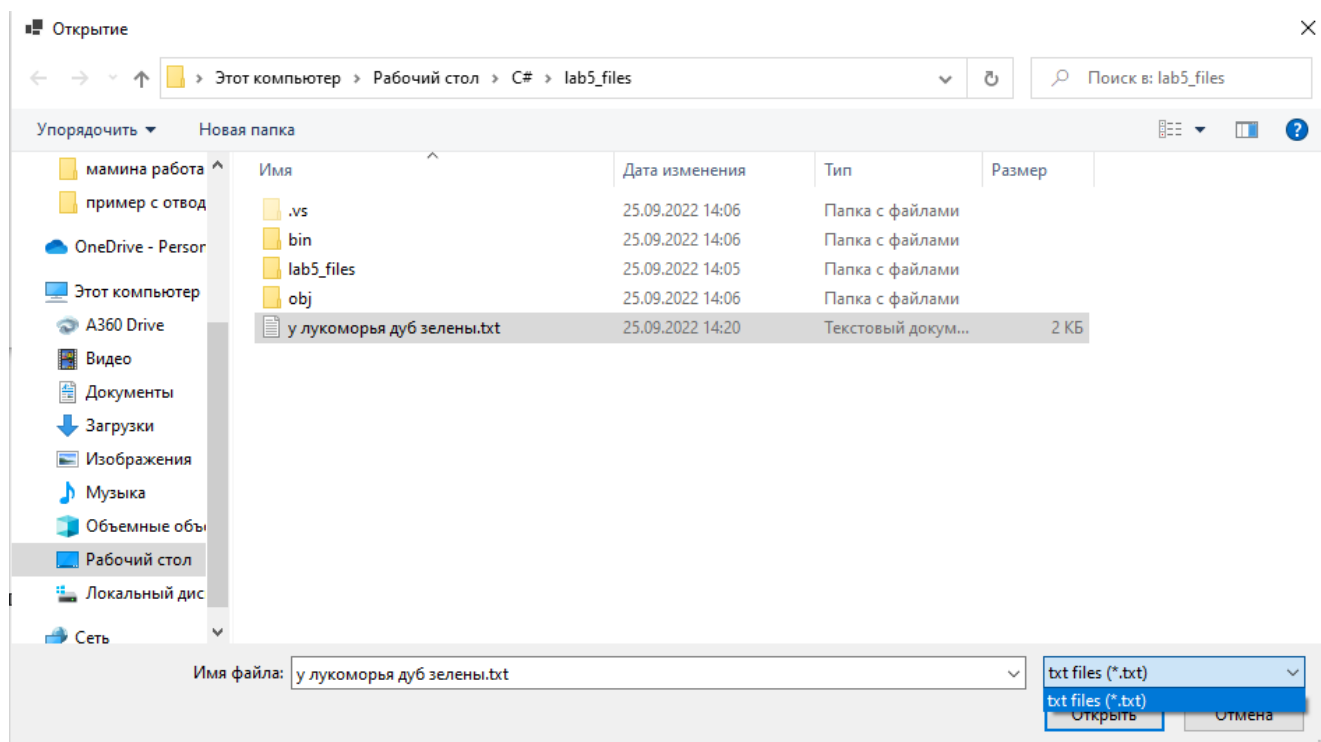


Рисунок 2 – Диалоговое окно выбора файла

При закрытии окна без выбора файла, появляется сообщение о том, что файл не выбран (рисунок 3).

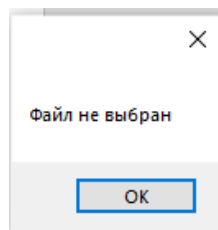


Рисунок 3 – Сообщении об ошибке, что файл не выбран

При успешном выборе файла появляется строка, сообщающая за какое время (в миллисекундах) был обработан текст из файла - разделен на неповторяющиеся слова (рисунок 4).

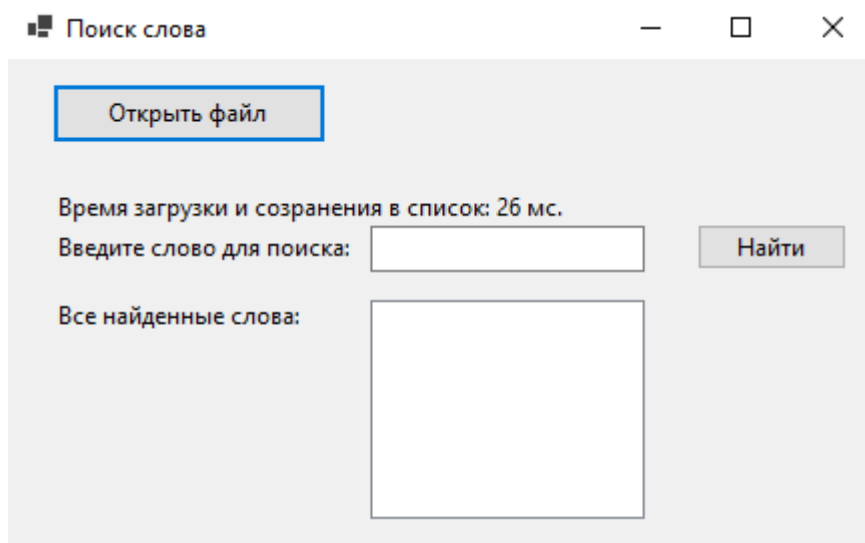


Рисунок 4 – Время, за которое был обработан текст в файле

Затем можно задавать слова для поиска. При их нахождении они добавляются в список и выводится время, затраченное на поиск (в микросекундах) – рисунок 5.

При отсутствии слова в тексте выводится соответствующее сообщение.

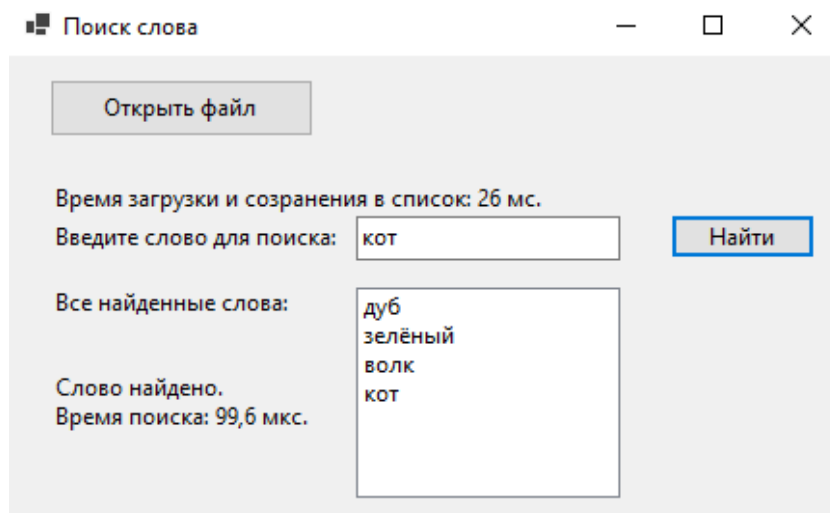


Рисунок 5 – Нахождение слов

Код формы:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;
using System.IO;
using System.Diagnostics;

namespace lab5_files
{
    public partial class Form1 : Form
    {
        List<string> words = new List<string>();
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            char[] separators = new char[] { ' ', '.', ',', '!',
            '...', ';', ':', '-', '—', '«', '»', '(', ')', '\"', '\\'
            };
            OpenFileDialog openFileDialog1 = new OpenFileDialog();
            openFileDialog1.Filter = "txt files (*.txt)|*.txt";
            if (openFileDialog1.ShowDialog() == DialogResult.OK)
            {
                Stopwatch stopWatch = new Stopwatch();
                stopWatch.Start();

                string all_text =
                File.ReadAllText(openFileDialog1.FileName);
                words = all_text.Split(separators,
                StringSplitOptions.RemoveEmptyEntries).ToList();

                stopWatch.Stop();
            }
        }
    }
}
```

```

        label1.Text = "Время загрузки и сохранения в список: " + Convert.ToString(stopWatch.ElapsedMilliseconds) + "
мс.";
    }
    else
        MessageBox.Show("Файл не выбран");
}

private void button1_Click_1(object sender, EventArgs e)
{
    Stopwatch stopWatch = new Stopwatch();
    stopWatch.Start();

    string find_word = textBox1.Text;
    if (words.Contains(find_word))
    {
        stopWatch.Stop();
        label4.Text = "Слово найдено. \nВремя поиска: " +
Convert.ToString(stopWatch.Elapsed.TotalMilliseconds *
1000) + " мкс.";
        listBox1.Items.Add(find_word);
    }
    else
        label4.Text = "Слово не найдено";
}
}
}

```

Вывод: в процессе выполнения лабораторной работы были изучены средства работы с текстовыми файлами и строками: способ открытия файла определенного расширения, чтение файла в одну строку, дробление строки в слова по разделителям. А также был использован инструмент stopWatch, позволяющий засекаать время выполнения кода.