

Московский государственный технический
университет имени Н.Э. Баумана

Парадигмы и конструкции языков программирования
Отчёт по лабораторной работе №4

Работу выполнил
Студент группы ИУ5-35Б
Шиленок Д.А.

2023 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.

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.

5. При сохранении слов в список List 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 5 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).

Текст программы

Program.cs

```
namespace Lab4
{
    class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            ApplicationConfiguration.Initialize();
            Application.Run(new Form1());
        }
    }
}
```

Form1.cs

```
using System.Collections.Generic;
using System.Diagnostics;

namespace Lab4
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        // СПИСОК СЛОВ
        List<string> words = new List<string>();
        private void BLoadFile_Click(object sender, EventArgs e)
        {
            OpenFileDialog openFileDialog = new OpenFileDialog();
            openFileDialog.Filter = "текстовые файлы|*.txt";
            if (openFileDialog.ShowDialog() == DialogResult.OK)
            {
                Stopwatch t = new Stopwatch();
                t.Start();
            }
        }
    }
}
```

```

//Чтение файла в виде строки
string text = File.ReadAllText(fileDialog.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 (!words.Contains(str)) words.Add(str);
}

t.Stop();
this.timeLabel.Text = t.Elapsed.ToString();
// this.textBoxFileReadCount.Text = list.Count.ToString();
}
else MessageBox.Show("Необходимо выбрать файл");
}

private void BFindWord_Click(object sender, EventArgs e)
{
    // Слово для поиска
    string word = this.textBoxWord2Find.Text.Trim();

    // Если пустой список слов
    if (words.Count == 0)
    {
        MessageBox.Show("Необходимо выбрать файл!");
    }
    // Если слово для поиска пусто
    else if (string.IsNullOrEmpty(word))
    {
        MessageBox.Show("Необходимо ввести слово для поиска!");
    }
    else
    {
        // Переводим слово для поиска в верхний регистр

```

```

string wordUpper = word.ToUpper();

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

Stopwatch t = new Stopwatch();
t.Start();

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

t.Stop();
this.TimeFindLabel.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();
}
}
}
}
}

```

Вывод программы

Загрузить файл

Время загрузки: 00:00:00.0001723

Поиск слова:

word

Поиск

Время поиска: 00:00:00.0000053

Результаты поиска:

testword
word
wordle