

**Московский государственный технический
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №4

Выполнил:

студент группы ИУ5-32

Кудрявцев Сергей

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю.С.

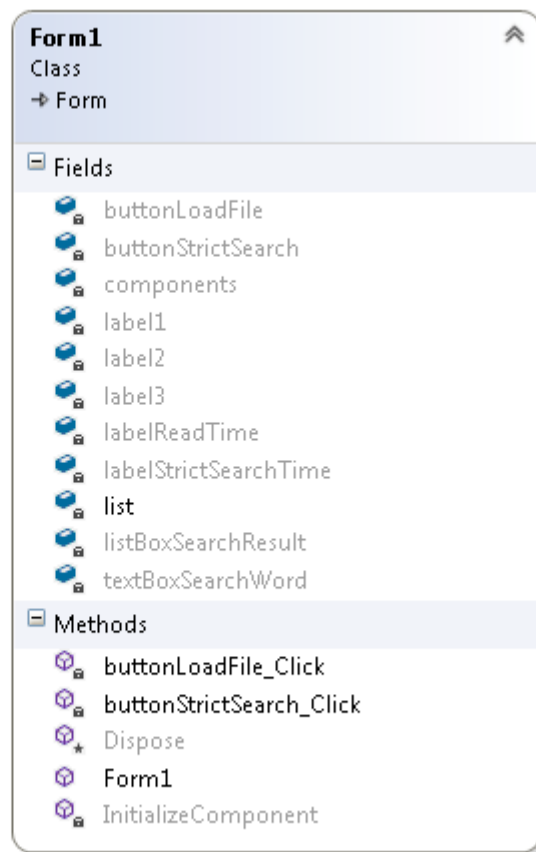
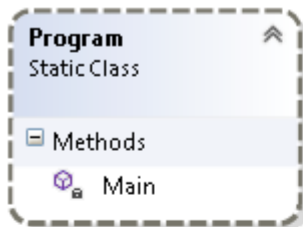
Подпись и дата:

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:

Program.cs:

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

namespace Lab4
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            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.Threading.Tasks;
```

```

using System.Windows.Forms;
using System.Diagnostics;
using System.IO;

namespace Lab4
{
    public partial class Form1 : Form
    {
        List<string> list = new List<string>();

        public Form1()
        {
            InitializeComponent();
        }

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

            if (fd.ShowDialog() == DialogResult.OK)
            {
                //XXXXXXXXXX
                list.Clear();

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

                //У меня на Windows очень странно читаются файлы. Нормально работает только с
                юникодом
                string text = File.ReadAllText(fd.FileName, Encoding.Unicode);

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

                string[] textArray = text.Split(separators);
                foreach (string strTemp in textArray)
                {
                    string str = strTemp.Trim();

                    if (!list.Contains(str))
                        list.Add(str);
                }

                t.Stop();
                this.labelReadTime.Text = t.Elapsed.ToString();
            }
            else
            {
                MessageBox.Show("Необходимо выбрать файл");
            }
        }

        private void buttonStrictSearch_Click(object sender, EventArgs e)
        {
            string word = this.textBoxSearchWord.Text.Trim();

            if (!string.IsNullOrEmpty(word) && list.Count > 0)
            {
                string wordUpper = word.ToUpper();

                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.labelStrictSearchTime.Text = t.Elapsed.ToString();

    this.listBoxSearchResult.BeginUpdate();

    this.listBoxSearchResult.Items.Clear();

    foreach (string str in tempList)
    {
        this.listBoxSearchResult.Items.Add(str);
    }

    this.listBoxSearchResult.EndUpdate();
}
else
{
    MessageBox.Show("Необходимо выбрать файл или ввести слово для поиска");
}
}
}
}

```

Примеры:

Form1

Чтение из файла Время чтения из файла: 00:00:00.0002158

Слово для поиска: и

Четкий поиск Время поиска слова: 00:00:00.0000152

привет
прив
пришел

Form1

Чтение из файла Время чтения из файла: 00:00:00.0002158

Слово для поиска:

Четкий поиск Время поиска слова: 00:00:00.0000152

Необходимо выбрать файл или ввести слово для поиска

OK