**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

«Трек курса «Задания на основе языка C#»»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-34Б |  | преподаватель каф. ИУ5 |
| Киреев А.А. |  | Ю.Е. Гапанюк. |
| Подпись и дата:  28.12.2020 |  | Подпись и дата:  28.12.2020 |

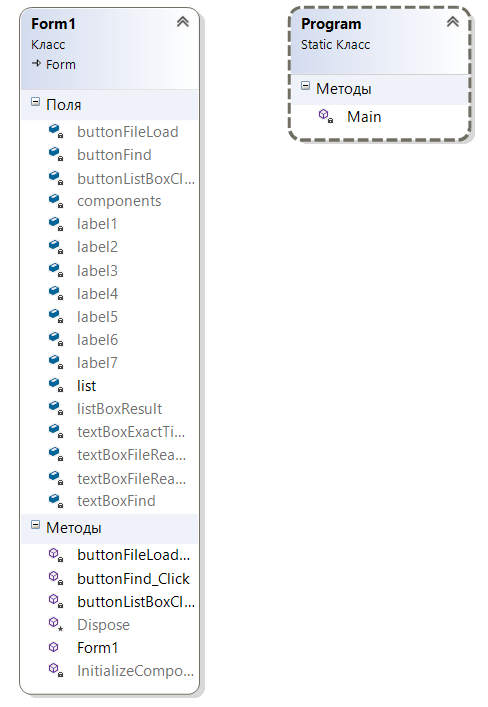
Москва, 2020 г.

**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 находится 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**

Для Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab4\_2

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

Для Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab4\_2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

List<string> list = new List<string>();

private void buttonFileLoad\_Click(object sender, EventArgs e)

{

OpenFileDialog fileDialog = new OpenFileDialog();//диалог открытия файла

fileDialog.Filter = "текстовые файлы|\*.txt";// фильтрация только по текстовым файлам

fileDialog.InitialDirectory = "C:\\Users\\111\\Documents"; //директория по умолчанию

if (fileDialog.ShowDialog() == DialogResult.OK) //если был выбран файл

{

list.Clear(); //очистка списка от предыдущих файлов

Stopwatch timeOpen = new Stopwatch();

timeOpen.Start();

string text = File.ReadAllText(fileDialog.FileName);//чтение файла

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

string[] textArray = text.Split(separators);

foreach (string strTemp in textArray)

{

string str = strTemp.Trim(separators);//Удаление разделителей в начале и в конце

if (!list.Contains(str)) list.Add(str); //добавление уникальных строк

}

if (list.Contains("") == true) //удаление "слов" при дублировании разделителей

{

list.Remove("");

}

timeOpen.Stop();

this.textBoxFileReadTime.Text = timeOpen.Elapsed.ToString();

this.textBoxFileReadCount.Text = list.Count.ToString();

}

else

{

MessageBox.Show("Выберите файл!"); //если файл не выбран

}

}

private void buttonFind\_Click(object sender, EventArgs e)

{

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

string word = this.textBoxFind.Text.Trim(separators); //Слово для поиска

if (!string.IsNullOrWhiteSpace(word) && list.Count > 0) //Проверка является ли слово пустым или разделителем

{

string wordLower = word.ToLower(); //преобразуем слово в нижний регистр

List<string> tempList = new List<string>(); //Временный список

Stopwatch timeFind = new Stopwatch();

timeFind.Start();

foreach (string str in list) //перебор слов в списке

{

if (str.ToLower().Contains(wordLower))

{

tempList.Add(str); //добавление во временный список

}

}

timeFind.Stop();

this.textBoxExactTime.Text = timeFind.Elapsed.ToString();

this.listBoxResult.BeginUpdate();

foreach (string str in tempList) //Вывод результатов поиска

{

this.listBoxResult.Items.Add(str);

}

this.listBoxResult.EndUpdate();

}

else

{

MessageBox.Show("Выберите файл и введите искомое слово!");

}

}

private void buttonListBoxClear\_Click(object sender, EventArgs e)

{

this.listBoxResult.Items.Clear();//Очистка списка

}

}

}

**Экранные формы с примерами выполнения программы**

