**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №4

«Разработка программы, реализующей работу с файлами»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-34Б |  | преподаватель каф. ИУ5 |
| Сергеев Илья |  | Гапанюк Ю. Е. |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

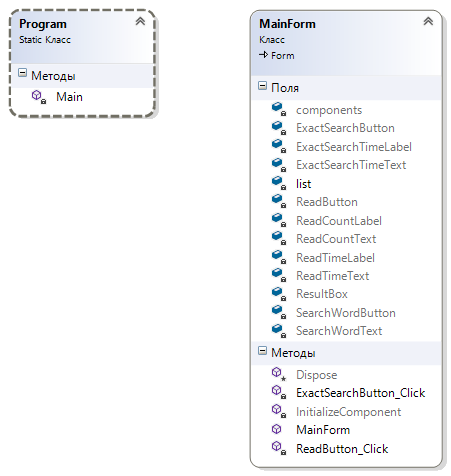
Москва, 2018 г.

**Описание задания**

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**

Диаграмма классов генерируется автоматически в среде Visual Studio:



**Текст программы (листинг)**

Листинг модуля формы:

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.Windows.Forms;

namespace files

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

/// <summary>

/// Word list type

/// </summary>

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

private void ReadButton\_Click(object sender, EventArgs e) //clicking on \*OPEN\* button

{

OpenFileDialog OD = new OpenFileDialog(); //new open dialog

OD.Filter = "Текстовые файлы|\*.txt"; //setting a filter

if (OD.ShowDialog() == DialogResult.OK) //if file has been chosen

{

Stopwatch sw = new Stopwatch(); //a new stopwatch

sw.Start(); //Чтение файла в виде строки //starting a stopwatch

string text = File.ReadAllText(OD.FileName); //reading whole text from the file

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

string[] array = text.Split(sep); //splitting a text by seps

foreach (string cur in array) //removing extra spaces

{

string str = cur.Trim();

if (!list.Contains(str)) //add to list if unique

list.Add(str);

}

sw.Stop(); //stopping the stopwatch

this.ReadTimeText.Text = sw.Elapsed.ToString(); //showing time of a search

this.ReadCountText.Text = list.Count.ToString(); //showing counter of words

}

else

MessageBox.Show("It's necessary to open a file!");

}

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

{

string word = this.SearchWordText.Text.Trim(); //a word for a search

if (!string.IsNullOrEmpty(word) && list.Count > 0) //if a field wasn't empty

{

string Upper = word.ToUpper(); //upping a case

List<string> TmpList = new List<string>(); //temporaty results list

Stopwatch sw = new Stopwatch(); //new stopwatch

sw.Start(); //starting the stopwatch

foreach (string str in list) //searching

if (str.ToUpper().Contains(Upper))

TmpList.Add(str); //adding a word

sw.Stop(); //stopping a stopwatch

this.ExactSearchTimeText.Text = sw.Elapsed.ToString(); //showing a time

this.ResultBox.BeginUpdate(); //updating a listbox

this.ResultBox.Items.Clear(); //clearing a listbox

foreach (string str in TmpList) //swowing found words to listbox

this.ResultBox.Items.Add(str);

this.ResultBox.EndUpdate(); //ending of listbox updating

}

else

MessageBox.Show("It's necessary to choose a file and a word to search!");

}

}

}

Листинг основной программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace files

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new MainForm());

}

}

}

**Экранные формы с примерами выполнения программы (скриншоты)**

