**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №4

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-35б |  | преподаватель каф. ИУ5 |
| Зубарева Антонина |  |  |
| Подпись и дата: |  | Подпись и дата: |

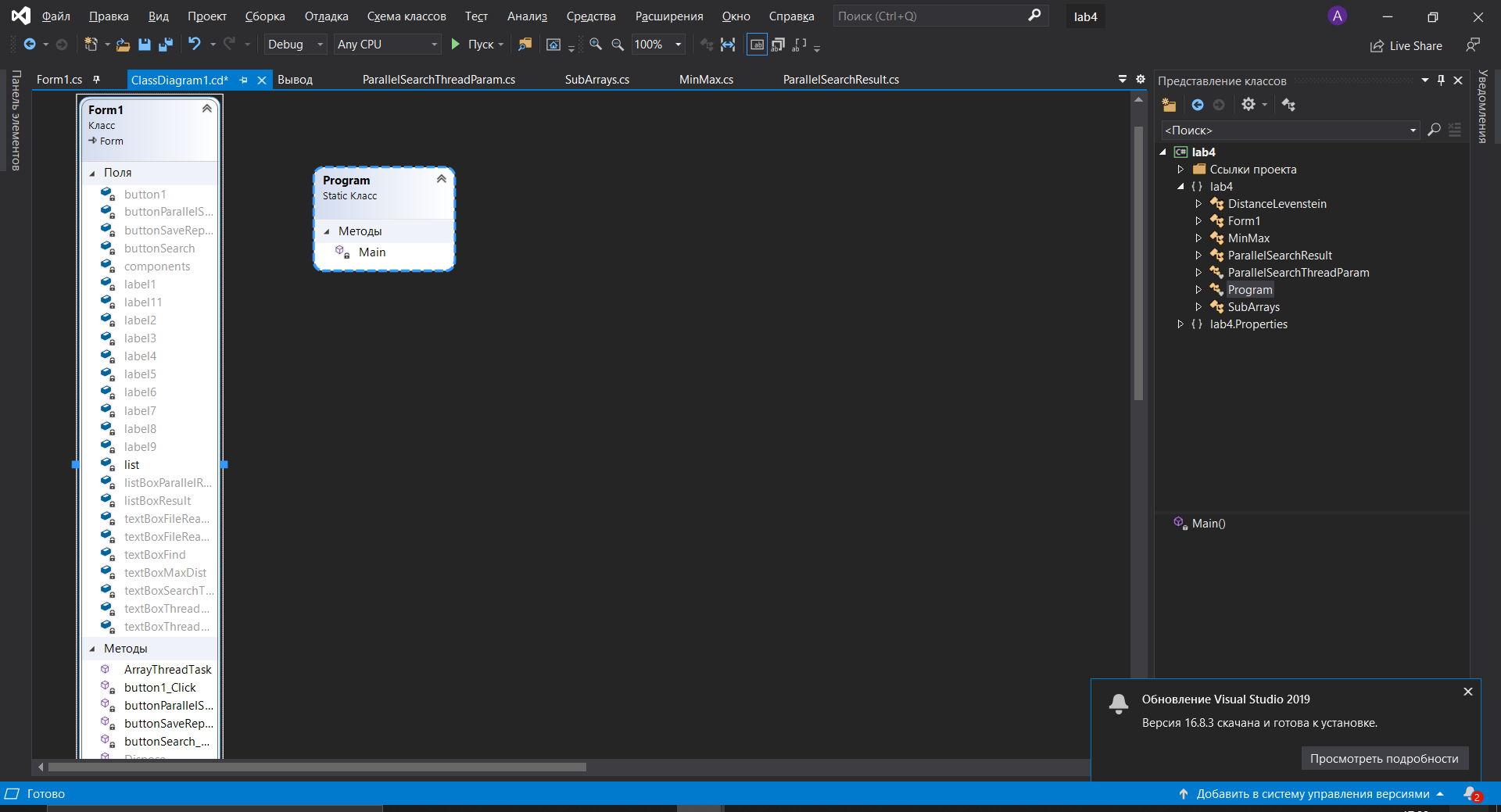
Москва, 2020 г.

# 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<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

# 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

# {

# public partial class Form1 : Form

# {

# ///<summary>

# ///Список слов

# ///</summary>

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

# public Form1()

# {

# InitializeComponent();

# }

# private void button1\_Click(object sender, EventArgs e)

# {

# OpenFileDialog fd = new OpenFileDialog();

# fd.Filter = "текстовые файлы|\*.txt";

# if (fd.ShowDialog() == DialogResult.OK)

# {

# Stopwatch t\_load = new Stopwatch();

# t\_load.Start();

# // Чтение файла в виде одной строки

# string text = File.ReadAllText(fd.FileName);

# //Разделительные символы для разбиения полученной строки

# 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 (!list.Contains(str)) list.Add(str);

# }

# t\_load.Stop();

# this.textBoxFileReadTime.Text = t\_load.Elapsed.ToString();

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

# MessageBox.Show("Файл прочитан");

# }

# else {

# MessageBox.Show("Необходимо выбрать файл");

# }

# }

# private void textBoxFileReadTime\_TextChanged(object sender, EventArgs e)

# {

# }

# private void buttonSearch\_Click(object sender, EventArgs e)

# {

# //Слово для поиска

# string word = this.textBoxFind.Text.Trim();

# // Если слово для поиска не пусто

# if (!string.IsNullOrWhiteSpace(word) && list.Count > 0)

# {

# //Слово для поиска в верхнем регистре

# string wordUpper = word.ToUpper();

# //Временные результаты поиска

# List<string> tempList = new List<string>();

# Stopwatch t\_search = new Stopwatch();

# t\_search.Start();

# foreach (string str in list)

# {

# if (str.ToUpper().Contains(wordUpper))

# {

# tempList.Add(str);

# }

# }

# t\_search.Stop();

# this.textBoxSearchTime.Text = t\_search.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();

# }

# else

# {

# MessageBox.Show("Необходимо выбрать файл и ввести слово для поиска");

# }

# }

# }

# Результаты

