

**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:

студент группы ИУ5-31Б

Федоров Иван

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Юрий Евгеньевич

Подпись и дата:

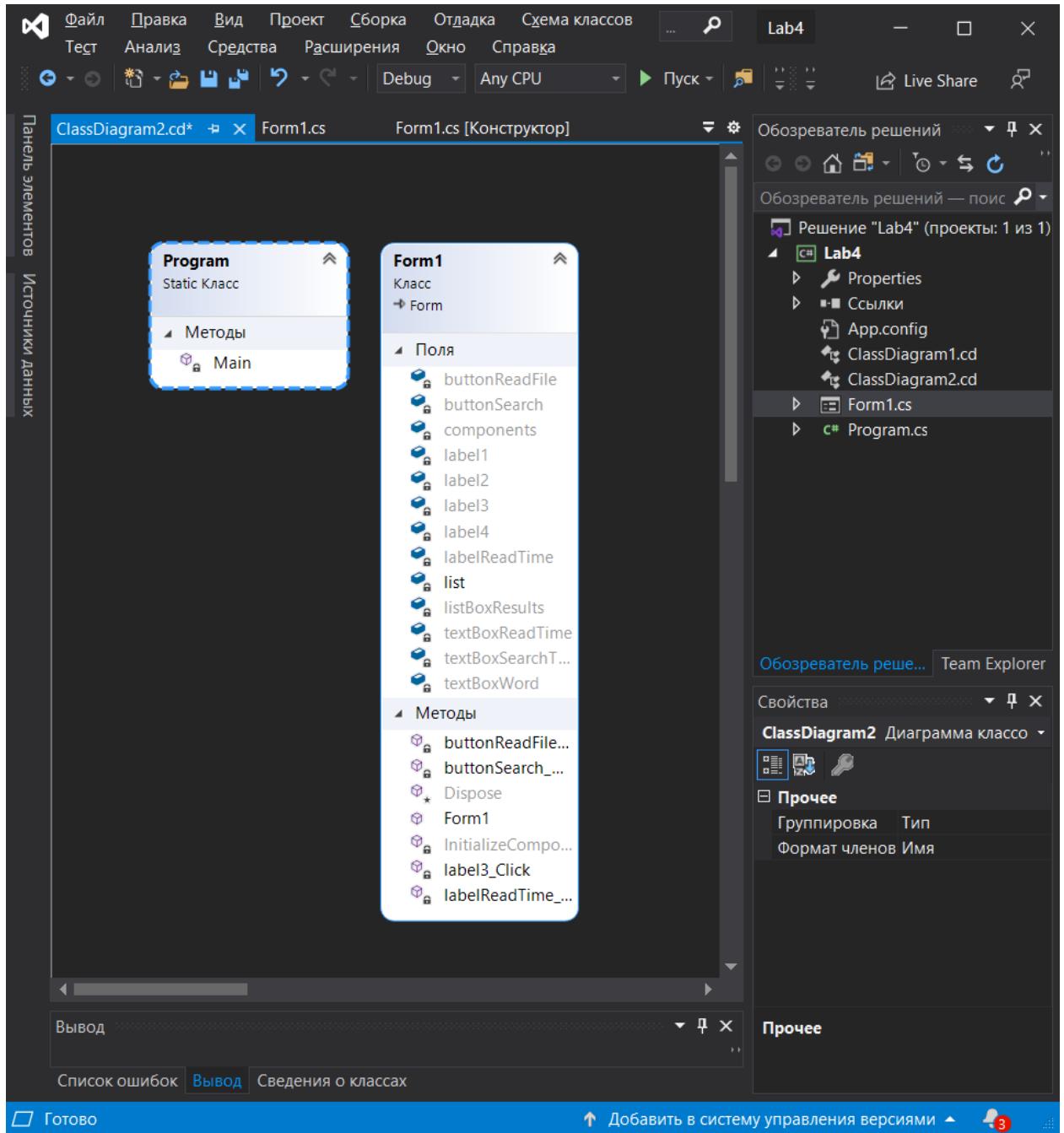
г. Москва, 2020 г.

Постановка задачи

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`).

Разработка интерфейса класса



Листинг программы

```
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.IO;
using System.Diagnostics;

namespace Lab4
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        List<string> list = new List<string>();
        private void buttonReadFile_Click(object sender, EventArgs e)
        {
            //выбираем файл через стандартное диалоговое окно
            OpenFileDialog txt = new OpenFileDialog();
            //Фильтр на TXT файлы
            txt.Filter = "Только текстовые файлы|*.txt";
            //если пользователь не выбрал файл
            if (txt.ShowDialog() != DialogResult.OK)
            {
                MessageBox.Show("Необходимо выбрать файл");
                return;
            }
            //разделители слов
            char[] separators = { '?', '.', ',', '!', '*', '/', ' ', '\t', '\n' };
            Stopwatch timer = new Stopwatch();
            timer.Start();
            //считывание текста из файла
            string text = File.ReadAllText(txt.FileName);
            //удаление ненужных символов в начале и конце текста
            text = text.Trim(separators);
            //разделение строки на слова
            string[] textArray = text.Split(separators);
            //Запись слов в список
            foreach (string word in textArray)
            {
                if (!word.Contains("-"))
                    //проверка на неповторяющиеся слова
                    if (!list.Contains(word))
                        list.Add(word);
            }
            timer.Stop();
            //запись данных из таймера
            this.textBoxReadTime.Text = timer.Elapsed.ToString();
            //this.listBoxFile.BeginUpdate();
            //this.listBoxFile.Items.Clear();
            //foreach (string word in list)
            //{
            //    this.listBoxFile.Items.Add(word);
            //}
            //this.listBoxFile.EndUpdate();
        }

        private void buttonSearch_Click(object sender, EventArgs e)
        {
            //запись слова и его очистка
            string word = this.textBoxWord.Text.Trim();
            //Подтягиваем буквы вверх, чтобы не было проблем с регистром
            word = word.ToUpper();
            //запуск листбокса
            this.listBoxResults.BeginUpdate();
            this.listBoxResults.Items.Clear();
            Stopwatch timer = new Stopwatch();

```

```

if (!string.IsNullOrWhiteSpace(word) && list.Count > 0)
{
    //Проверка на случай отсутствия совпадений
    bool NoMatches = true;
    //запуск таймера
    timer.Start();
    foreach (string w in list)
    {
        //идёт проверка слов в верхнем регистре
        if (w.ToUpper().Contains(word))
        {
            this.listBoxResults.Items.Add(w);
            NoMatches = false;
        }
    }
    //остановка таймера
    timer.Stop();
    //Если совпадений всё же не нашлось
    if (NoMatches) this.listBoxResults.Items.Add("Нет сопадений");
}
else
{
    MessageBox.Show("Необходимо открыть файл и выбрать слово для поиска");
}
this.listBoxResults.EndUpdate();
//запись данных из таймера
this.textBoxSearchTime.Text = timer.Elapsed.ToString();
}

private void label3_Click(object sender, EventArgs e)
{
}

private void labelReadTime_Click(object sender, EventArgs e)
{
}

}
}

```

Анализ результатов

