**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет**

**имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

Факультет «Информатика и вычислительная техника»

Кафедра ИУ5 «Системы обработки информации и управления»

**Отчет по лабораторной работе № 4**

Задание: разработать программу, реализующую работу с файлами

Студент Левин Михаил Александрович

*фамилия, имя, отчество*

Группа ИУ5-32Б

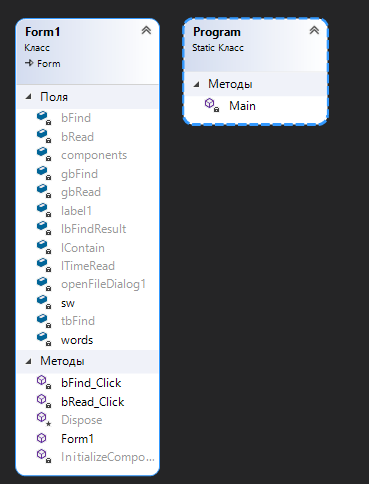
Москва, 2020 г.

# 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

**Form1.cs**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.IO;

using System.Windows.Forms;

namespace Lab4

{

public partial class Form1 : Form

{

private List<string> words;

private Stopwatch sw = new Stopwatch();

public Form1()

{

InitializeComponent();

}

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

{

openFileDialog1.InitialDirectory = Directory.GetCurrentDirectory();

openFileDialog1.Filter = "txt files (\*.txt)|\*.txt";

if (openFileDialog1.ShowDialog()==DialogResult.OK)

{

//Get the path of specified file

var filePath = openFileDialog1.FileName;

words = new List<string>();

sw.Reset();

sw.Start();

words.AddRange(File.ReadAllText(filePath).Split(" ".ToCharArray(),StringSplitOptions.RemoveEmptyEntries));

sw.Stop();

lTimeRead.Text = $"Время считывания файла: {sw.Elapsed}";

}

}

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

{

sw.Reset();

sw.Start();

lbFindResult.Items.Clear();

lbFindResult.BeginUpdate();

foreach (var word in words)

{

if (word.Contains(tbFind.Text))

{

lbFindResult.Items.Add(word);

}

}

lbFindResult.EndUpdate();

sw.Stop();

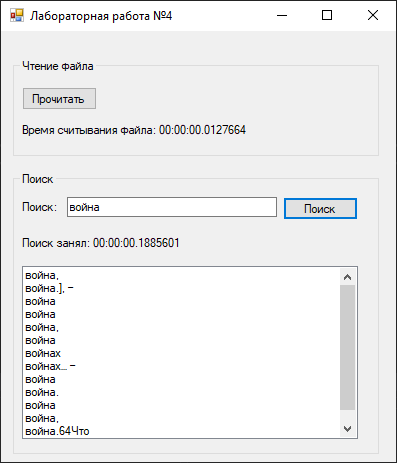
lContain.Text = $"Поиск занял: {sw.Elapsed}";

}

}

}

# Экранные формы



# Контрольные вопросы:

1. Как создать приложение Windows Forms?

Для создания приложения Windows Forms необходимо в Visual Studio выбрать проект «Приложение Windows Forms (.NET Framework)», после создания проекта будет доступна форма, созданная по-умолчанию Form1.

1. Как реализовать измерение времени выполнения программы с использованием класса Stopwatch?

Класс Stopwatch предоставляет набор методов и свойств, которые можно использовать для точного измерения затраченного времени. Для запуска отчёта времени необходимо вызвать метод Start() перед тем кодом, время выполнения которого необходимо замерять, а после измеряемого когда необходимо вызвать метод объекта Stopwatch – Stop(). Время выполнения можно узнать через обращение к свойству Elsapsed.

1. Для чего используются события FormClosed и FormClosing?

Событие FormClosed происходит после закрытия формы. Это событие возникает после закрытия формы пользователем или Close методом или Exit методом Application класса. Чтобы предотвратить закрытие формы, необходимо обработать событие и задать Cancel для свойства объекта, CancelEventArgs переданного обработчику событий, значение true.

Это событие можно использовать для выполнения таких задач, как освобождение ресурсов, используемых формой, и сохранения информации, введенной в форму, или для обновления ее родительской формы.

Событие FormClosing происходит перед закрытием формы. Когда форма закрывается, она удаляется, освобождая все ресурсы, связанные с формой. Если отменить это событие, форма останется открытой. Для отмены закрытия формы, задаётся свойства Cancel объекта, FormClosingEventArgs переданного обработчику событий, значение true.

1. Как используется класс OpenFileDialog?

Класс OpenFileDialog отображает диалоговое окно, позволяющее пользователю открыть файл. Этот класс позволяет проверить, существует ли файл, и открыть его. ShowReadOnlyСвойство определяет, отображается ли флажок только для чтения в диалоговом окне. ReadOnlyCheckedСвойство указывает, установлен ли флажок только для чтения. Для вызова отображения диалога используется метод ShowDialog(), возвращаемое значение типа DialogResult позволяющее определить результат вызова диалога. Большая часть основных функций для этого класса находится в FileDialog классе.

1. Как используется класс SaveFileDialog?

Класс SaveFileDialog запрашивает у пользователя местоположение для сохранения файла. Этот класс может либо открыть, либо перезаписать существующий файл, либо создать новый файл. Вызов диалога выбора файла осуществляется аналогично OpenFileDialog.

1. Как используется элемент TextBox?

Класс TextBox Предоставляет элемент управления "текстовое поле" Windows. С помощью TextBox элемента управления пользователь может вводить текст в приложении. Этот элемент управления обладает дополнительными функциональными возможностями, которые не найдены в стандартном элементе управления "текстовое поле Windows", включая многострочное редактирование и маскирование символов пароля.

Как правило, TextBox элемент управления используется для вывода или приема в качестве входных данных, одной строки текста. Можно использовать Multiline Свойства и, ScrollBars чтобы разрешить отображение или ввод нескольких строк текста.

1. Как осуществить чтение текстового файла в виде единой строки?

Для осуществления чтения текстового файла в виде единой строки используется метод File.ReadAllText. Открывает текстовый файл, считывает весь текст файла в строку и затем закрывает файл.

1. Как осуществить запись текстового файла в виде единой строки?

Для осуществления записи текстового файла в виде единой строки используется метод File.WriteAllText, который создает новый файл, записывает в него содержимое и затем закрывает файл. Если целевой файл уже существует, он будет переопределен.

1. Как осуществить чтение текстового файла в виде массива строк?

Для чтение текстового файла в виде массива строк используется метод File.ReadAllLines, который открывает текстовый файл, считывает все строки файла в массив строк и затем закрывает файл.

1. Как осуществить запись текстового файла в виде массива строк?

Для осуществления записи текстового файла в виде массива строк используется метод File.WriteAllLines, который создает новый файл, записывает в него одну или несколько строк, затем закрывает файл.