**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет технологий»

Отчет по лабораторной работе №4

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-33Б |  | преподаватель каф. ИУ5 |
| Кузнецов Григорий |  |  |
| Подпись и дата: |  | Подпись и дата: |

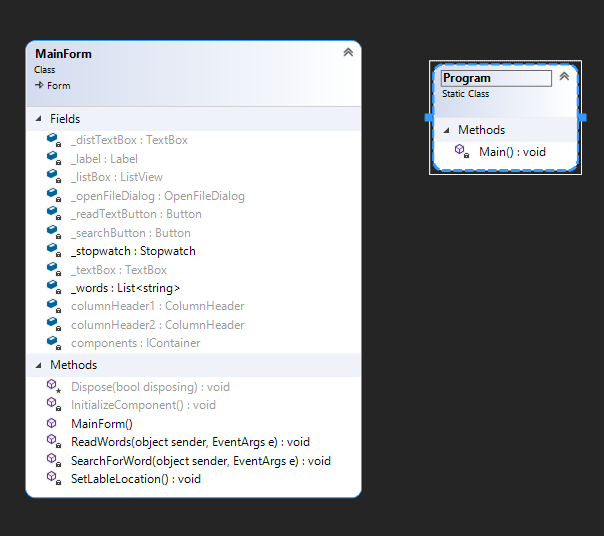
Москва, 2020 г.

**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**

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Windows.Forms;

using System.IO;

using System.Diagnostics;

namespace Lab4

{

public partial class MainForm : Form

{

private readonly List<string> \_words;

private readonly Stopwatch \_stopwatch;

public MainForm()

{

InitializeComponent();

\_words = new List<string>();

\_stopwatch = new Stopwatch();

}

private void ReadWords(object sender, EventArgs e)

{

if (\_openFileDialog.ShowDialog() == DialogResult.OK)

{

\_listBox.Items.Clear();

\_stopwatch.Start();

var text = File.ReadAllText(\_openFileDialog.FileName);

foreach (var word in text.Split(" \n\t,./\\[]{}<>@#$%^&\*();'\"|?!".ToCharArray()))

{

if (!\_words.Contains(word) && word != "")

\_words.Add(word);

}

\_stopwatch.Stop();

\_label.Text = $"Время загрузки текста:\n{\_stopwatch.Elapsed}ms";

\_label.Visible = true;

SetLableLocation();

\_searchButton.Enabled = true;

\_stopwatch.Reset();

}

}

private void SetLableLocation() =>

\_label.Location = Point.Add(\_readTextButton.Location, new Size((\_readTextButton.Size.Width - \_label.Size.Width) / 2, -\_readTextButton.Size.Height - 10));

private void SearchForWord(object sender, EventArgs e)

{

int maxDist;

try

{

maxDist = Convert.ToInt32(\_distTextBox.Text);

}

catch (FormatException)

{

MessageBox.Show("Неправильный формат ввода максимальной дистанции", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

\_listBox.BeginUpdate();

\_listBox.Items.Clear();

\_stopwatch.Start();

foreach (var word in \_words)

{

if (word.Contains(\_textBox.Text))

{

var dist = Lab5.Methods.LevenshteinDistance(word, \_textBox.Text);

if (dist <= maxDist)

\_listBox.Items.Add(new ListViewItem(new string[] { word, dist.ToString() }));

}

}

\_stopwatch.Stop();

\_listBox.EndUpdate();

\_label.Text = $"Время поиска слов:\n{\_stopwatch.ElapsedMilliseconds}ms";

SetLableLocation();

\_stopwatch.Reset();

}

}

}

Text.txt:

Уважаемые студенты. Для сдачи курса Вам необходимо выполнить 6 лабораторных работ (условия приведены по 7 лабораторным работам) и домашнее задание. Возможны следующие варианты выполнения заданий:

Выполнение 6 лабораторных работ и ДЗ на основе языка C# (стандартный трек курса).

Выполнение 6 лабораторных работ, часть из которых может быть выполнена на языке C#, а часть на языке F# (трек курса «функциональное программирование»). ДЗ может быть выполнено в виде проекта на C#, в виде проекта на F#, в виде реферата (черновика статьи) по тематике курса. В этом случае тематика проекта (реферата) должна быть согласована с преподавателями курса.

Полностью самостоятельный проект (исследование), который включает программирование на объектно-ориентированном или функциональном языке (исследовательский трек курса). В этом случае тематика проекта (исследования) должна быть согласована с преподавателями курса.

ТРЕБОВАНИЯ К ОТЧЕТАМ

Отчеты по лабораторным работам и пояснительная записка по выполнению домашнего задания оформляются в электронном и бумажном виде.

Электронные варианты отчетов размещаются в личном репозитории студентов (например, на github или аналогичном ресурсе).

Отчеты по лабораторным работам и пояснительная записка по выполнению домашнего задания должны содержать:

титульный лист;

описание задания;

диаграмму классов, которая, например, генерируется автоматически с использованием Visual Studio (если используемый Вами инструмент поддерживает такую возможность);

текст программы;

экранные формы с примерами выполнения программы.

**Пример выполнения программы**

