

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Радиотехнический»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет технологий»
Пояснительная записка по домашнему заданию

Выполнил:

студент группы РТ5-31Б
Сысоев Александр

Проверил:

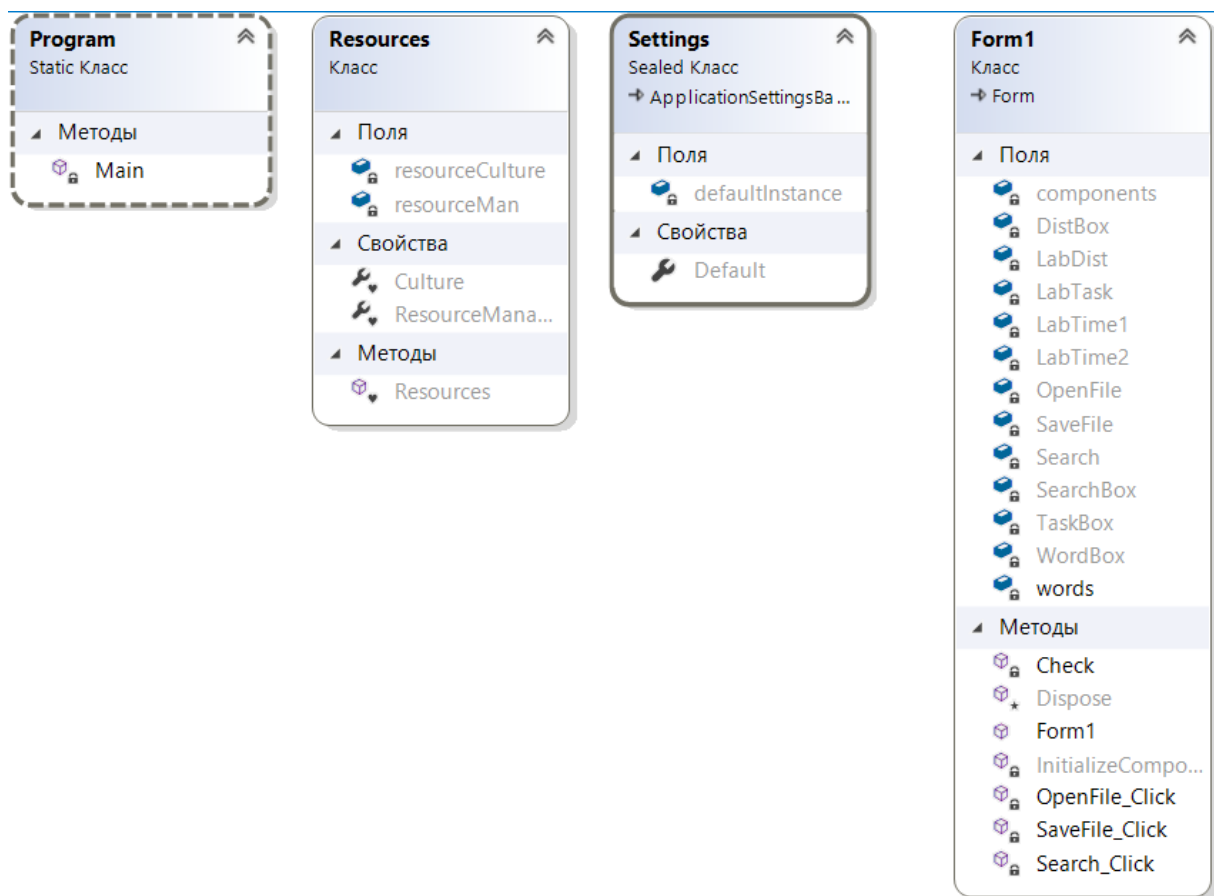
доцент каф. ИУ5
Гапанюк Ю.Е.

Описание задания

Разработать программу, реализующую многопоточный поиск в файле.

1. Программа должна быть разработана в виде приложения Windows Forms на языке C#. По желанию вместо Windows Forms возможно использование WPF.
2. В качестве основы используется макет, разработанный в лабораторных работах №4 и №5.
3. Реализуйте функцию поиска с использованием расстояния Левенштейна в многопоточном варианте. Количество потоков для запуска функции поиска вводится на форме в поле ввода (TextBox). В качестве примера используйте проект «Parallel» из примера «Введение в C#».
4. Реализуйте функцию записи результатов поиска в файл отчета. Файл отчета создается в формате .txt или .html. В качестве примера используйте проект «WindowsFormsFiles» (обработчик события кнопки «Сохранение отчета») из примера «Введение в C#».

Диаграмма классов



Текст программы

Код библиотеки классов

```
using System;

namespace Lab5.Lib
{
    public abstract class Levenshtein
    {
        private static int Min3(int a, int b, int c)
        {
            return Math.Min(Math.Min(a, b), c);
        }

        public static int Calculate(string s1, string s2)
        {
            int n = s1.Length + 1;
            int m = s2.Length + 1;
            int[,] D = new int[n, m];
            for (int i = 0; i < n; i++)
            {
                D[i, 0] = i;
            }
            for (int i = 0; i < m; i++)
            {
                D[0, i] = i;
            }
            for (int i = 1; i < n; i++)
```

```

        {
            for (int j = 1; j < m; j++)
            {
                int cost = s1[i - 1] == s2[j - 1] ? 0 : 1;
                D[i, j] = Min3(D[i - 1, j] + 1, D[i, j - 1] + 1, D[i - 1, j - 1] +
cost);
                if (i > 1 && j > 1 && s1[i - 1] == s2[j - 2] && s1[i - 2] == s2[j -
1])
                {
                    D[i, j] = Math.Min(D[i, j], D[i - 2, j - 2] + cost);
                }
            }
        }
        return D[n - 1, m - 1];
    }
}
}

```

Код программы

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Text.RegularExpressions;
using System.Diagnostics;
using System.Windows.Forms;
using System.Threading.Tasks;
using System.Text;

namespace Lab4
{
    public partial class Form1 : Form
    {
        private List<string> words;

        private static List<string> Check(object paramObj)
        {
            List<string> wordlist = new List<string>();
            Tuple<int, List<string>, string> param = (Tuple<int, List<string>,
string>)paramObj;
            int dist = param.Item1;
            List<string> temp = param.Item2;
            string search = param.Item3;
            foreach (string s in temp)
                if (Lab5.Lib.Levenshtein.Calculate(s, search) <= dist)
                    wordlist.Add(s);
            return wordlist;
        }

        public Form1()
        {
            InitializeComponent();
            LabTime1.Text = "";
            LabTime2.Text = "";
            LabDist.Text = "Введите минимальное расстояние Левенштейна:";
            LabTask.Text = "Введите кол-во потоков (> 0):";
        }

        private void OpenFile_Click(object sender, EventArgs e)
        {
            OpenFileDialog ofd = new OpenFileDialog();
            ofd.Filter = "Текстовые файлы (*.txt)|*.txt";
            if (ofd.ShowDialog() == DialogResult.OK)
            {
                string text = "";
            }
        }
    }
}

```

```

        Stopwatch time = Stopwatch.StartNew();
        text = File.ReadAllText(ofd.FileName).ToLower();
        text = Regex.Replace(text, @"\W", " ");
        List<string> splitText = new List<string>(text.Split(new char[] { ' ' },
StringSplitOptions.RemoveEmptyEntries));
        words = new List<string>();
        foreach (string s in splitText)
            if (!words.Contains(s))
                words.Add(s);
        time.Stop();
        LabTime1.Text = "Время загрузки: " + time.Elapsed.ToString();
    }
}

private void Search_Click(object sender, EventArgs e)
{
    int dist = 0, taskcount = 0;
    if (SearchBox.Text != "" && words != null && int.TryParse(DistBox.Text, out
dist) && int.TryParse(TaskBox.Text, out taskcount) && taskcount > 0)
    {
        int wordcount = words.Count / taskcount, min = 0;
        string search = SearchBox.Text;
        WordBox.Items.Clear();
        Task<List<string>>[] tasks = new Task<List<string>>[taskcount];
        Stopwatch time = Stopwatch.StartNew();
        for (int i = 0; i < taskcount; i++)
        {
            List<string> temp = new List<string>();
            if (i == taskcount - 1)
                for (int j = min; j < words.Count; j++)
                    temp.Add(words[j]);
            else
                for (int j = min; j < min + wordcount; j++)
                {
                    temp.Add(words[j]);
                }
            tasks[i] = new Task<List<string>>(Check, new Tuple<int, List<string>,
string>(dist, temp, search));
            tasks[i].Start();
            min += wordcount;
        }
        Task.WaitAll(tasks);
        time.Stop();
        WordBox.BeginUpdate();
        for (int i = 0; i < taskcount; i++)
            foreach (var v in tasks[i].Result)
                WordBox.Items.Add(v.ToString());
        WordBox.EndUpdate();
        LabTime2.Text = "Время загрузки: " + time.Elapsed.ToString();
    }
}

private void SaveFile_Click(object sender, EventArgs e)
{
    string TempReportFileName = "Report_" +
DateTime.Now.ToString("dd_MM_yyyy_hhmmss");
    SaveFileDialog fd = new SaveFileDialog();
    fd.FileName = TempReportFileName;
    fd.DefaultExt = ".html";
    fd.Filter = "HTML Reports|*.html";
    if (fd.ShowDialog() == DialogResult.OK)
    {
        string ReportFileName = fd.FileName;
        StringBuilder b = new StringBuilder();
        b.AppendLine("<html>");
    }
}

```

```

        b.AppendLine("<head>");
        b.AppendLine("<meta http-equiv='Content-Type' content='text/html; charset=UTF-8' />");
        b.AppendLine("<title>" + "Отчет: " + ReportFileName + "</title>");
        b.AppendLine("</head>");
        b.AppendLine("<body>");
        b.AppendLine("<h1>" + "Отчет: " + ReportFileName + "</h1>");
        b.AppendLine("<table border='1'>");
        b.AppendLine("<tr>");
        b.AppendLine("<td>Время чтения из файла</td>");
        b.AppendLine("<td>" + LabTime1.Text + "</td>");
        b.AppendLine("</tr>");
        b.AppendLine("<tr>");
        b.AppendLine("<td>Количество уникальных слов в файле</td>");
        b.AppendLine("<td>" + WordBox.Items.Count.ToString() + "</td>");
        b.AppendLine("</tr>");
        b.AppendLine("<tr>");
        b.AppendLine("<td>Слово для поиска</td>");
        b.AppendLine("<td>" + SearchBox.Text + "</td>");
        b.AppendLine("</tr>");
        b.AppendLine("<tr>");
        b.AppendLine("<td>Максимальное расстояние для нечеткого поиска</td>");
        b.AppendLine("<td>" + DistBox.Text + "</td>");
        b.AppendLine("</tr>");
        b.AppendLine("<tr>");
        b.AppendLine("<td>Время нечеткого поиска</td>");
        b.AppendLine("<td>" + LabTime2.Text + "</td>");
        b.AppendLine("</tr>");
        b.AppendLine("<tr valign='top'>");
        b.AppendLine("<td>Результаты поиска</td>");
        b.AppendLine("<td>");
        b.AppendLine("<ul>");
        foreach (var x in WordBox.Items)
        {
            b.AppendLine("<li>" + x.ToString() + "</li>");
        }
        b.AppendLine("</ul>");
        b.AppendLine("</td>");
        b.AppendLine("</tr>");
        b.AppendLine("</table>");
        b.AppendLine("</body>");
        b.AppendLine("</html>");
        File.AppendAllText(ReportFileName, b.ToString());
        MessageBox.Show("Отчет сформирован. Файл: " + ReportFileName);
    }
}
}
}

```

Результат работы программы

Form1

Открыть файл...

Время загрузки: 00:00:00.0496615

Введите кол-во потоков (> 0):

3

Введите минимальное расстояние Левенштейна:

3

алиса

алиса
ли
слова
алису

Найти...

Время загрузки: 00:00:00.2098107

Сохранить отчёт

×

Отчет сформирован. Файл:
C:\Users\alexx\Desktop\Report_29_12_2020_094222.html

OK

Отчет: C:\Users\alexx\Desktop\Report_29_12_2020_094222.html

Время чтения из файла	Время загрузки: 00:00:00.0496615
Количество уникальных слов в файле	4
Слово для поиска	алиса
Максимальное расстояние для нечеткого поиска	3
Время нечеткого поиска	Время загрузки: 00:00:00.2098107
Результаты поиска	<ul style="list-style-type: none">• алиса• ли• слова• алису