

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

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по домашнему заданию

Выполнил:

студент группы ИУ5-31

Головацкий Андрей

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю. Е.

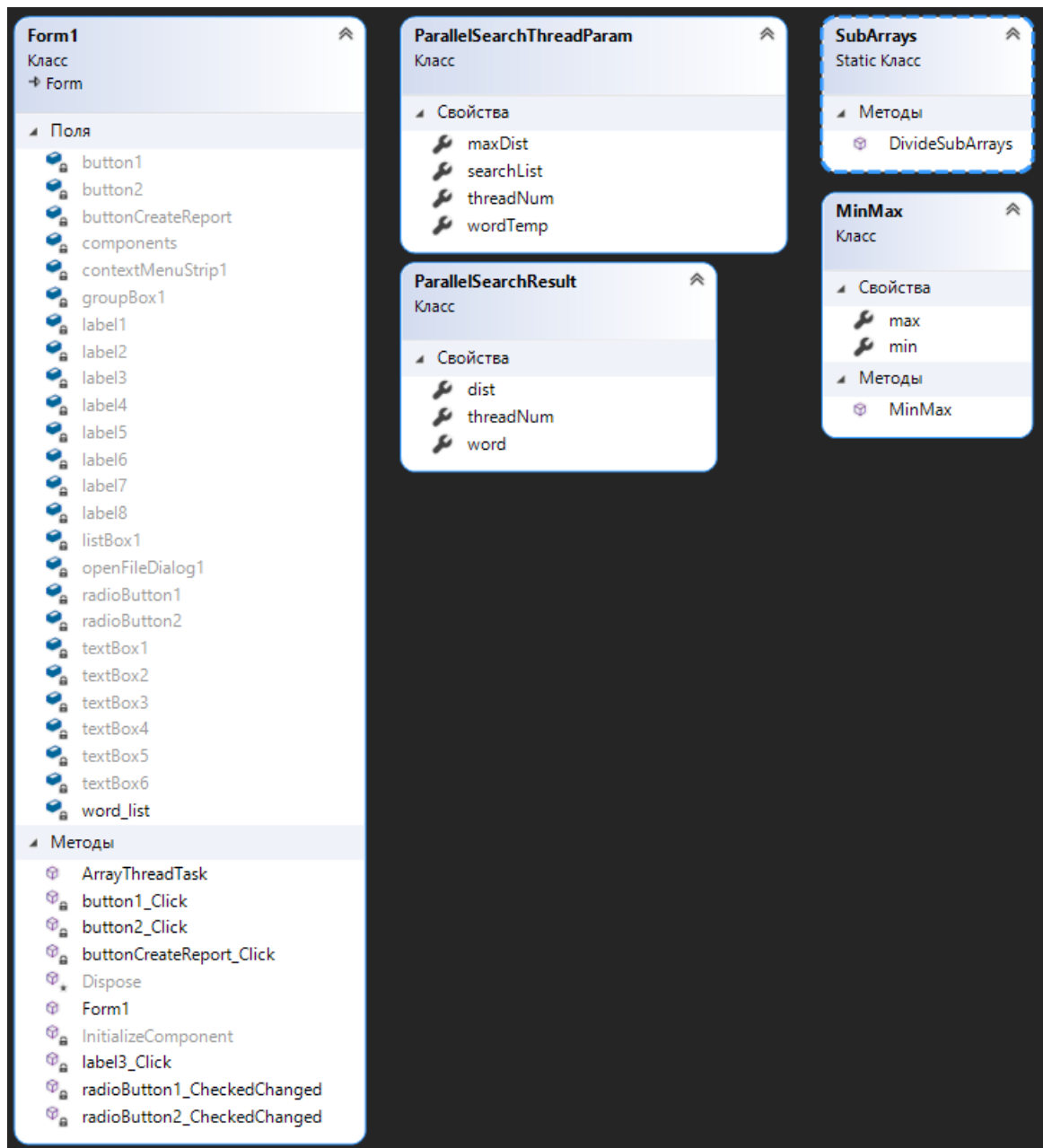
Подпись и дата:

Москва, 2020 г.

## Описание задания:

- 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:

### 1) Form1.cs:

```
1. using System;
2. using System.Collections.Generic;
3. using System.ComponentModel;
4. using System.Data;
5. using System.Drawing;
6. using System.Linq;
7. using System.Text;
8. using System.Threading.Tasks;
9. using System.Windows.Forms;
10.     using System.IO;
11.     using System.Diagnostics;
12.     using Lab5;
13.
14.     namespace Lab4
15.     {
16.         public partial class Form1 : Form
17.         {
18.             List<string> word_list = new List<string>();
19.             public Form1()
20.             {
21.                 InitializeComponent();
22.
23.             }
24.
25.             private void button1_Click(object sender, EventArgs e)
26.             {
27.                 word_list.Clear();
28.                 Stopwatch extime = new Stopwatch();
29.                 string fileContent;
30.                 string filePath;
31.                 string[] str;
32.                 OpenFileDialog openFile = new OpenFileDialog();
33.                 openFile.Filter = "Текстовый файл|*.txt";
34.                 if (openFile.ShowDialog() == DialogResult.OK)
35.                 {
36.                     extime.Start();
37.                     filePath = openFile.FileName;
```

```

38.                textBox2.Text = filePath.Split('\\').Last();
39.                fileContent = File.ReadAllText(filePath);
40.                char[] separators = new char[] { ' ', '.', ',', '!', '?', '/', '\t', '\n', '\f' };
41.                strs = fileContent.Split(separators);
42.                for (int i = 0; i < strs.Length; i++)
43.                {
44.                    strs[i] = strs[i].Trim();
45.                    if (!word_list.Contains(strs[i]) && (strs[i]
                        != " ") && (strs[i].Length > 0))
46.                        word_list.Add(strs[i]);
47.                }
48.            }
49.            extime.Stop();
50.            textBox1.Text = extime.Elapsed.ToString();
51.        }
52.
53.        private void label3_Click(object sender, EventArgs e)
54.        {
55.
56.        }
57.        public static List<ParallelSearchResult>
            ArrayThreadTask(object t_param)
58.        {
59.            ParallelSearchThreadParam param = t_param as
                ParallelSearchThreadParam;
60.            string word_temp = param.wordTemp.Trim();
61.            List<ParallelSearchResult> result_list = new
                List<ParallelSearchResult>();
62.            foreach (string str in param.searchList)
63.            {
64.                int dist =
                    LevDistance.VagnerCalculation(str.ToUpper(), word_temp.ToUpper());
65.                if (dist <= param.maxDist)
66.                {
67.                    ParallelSearchResult temp_list = new
                        ParallelSearchResult()
68.                    {
69.                        word = str,
70.                        dist = dist,

```

```
71.             threadNum = param.threadNum
72.             };
73.             result_list.Add(temp_list);
74.         }
75.     }
76.     return result_list;
77. }
78. private void button2_Click(object sender, EventArgs e)
79. {
80.     Stopwatch extime = new Stopwatch();
81.     string new_str = "\0";
82.     Random rnd = new Random();
83.     string word_temp = textBox3.Text.Trim();
84.     List<string> tempList = new List<string>();
85.     if (string.IsNullOrEmpty(word_temp))
86.     {
87.         MessageBox.Show("Пожалуйста, введите слово для
            поиска!");
88.     }
89.     button2.BackColor = Color.FromArgb(rnd.Next(0,
            255), rnd.Next(0, 255), rnd.Next(0, 255));
90.     button2.ForeColor = Color.FromArgb(rnd.Next(0, 255),
            rnd.Next(0, 255), rnd.Next(0, 255));
91.     if (radioButton1.Checked == true)
92.     {
93.         extime.Start();
94.         for (int i = 0; i < word_list.Count; i++)
95.         {
96.             if
97.             (word_list[i].ToUpper().Contains(word_temp.ToUpper()) &&
98.              !listBox1.Items.Contains(word_list[i]))
99.             {
100.                 new_str = word_list[i];
101.                 tempList.Add(new_str);
102.             }
103.         }
104.     }
105.     else if (radioButton2.Checked == true)
106.     {
```

```
105.                if (!int.TryParse(textBox6.Text, out int
thread_count))
106.                {
107.                    MessageBox.Show("Пожалуйста, введите
количество потоков!");
108.                    return;
109.                }
110.                if (!int.TryParse(textBox5.Text, out int
max_dist))
111.                {
112.                    MessageBox.Show("Пожалуйста, введите
максимальное редакционное расстояние!");
113.                    return;
114.                }
115.                extime.Start();
116.                List<ParallelSearchResult> result_list = new
List<ParallelSearchResult>();
117.                List<MinMax> div_list =
SubArrays.DivideSubArrays(0, word_list.Count, thread_count);
118.                int countSub = div_list.Count;
119.                Task<List<ParallelSearchResult>>[] tasks = new
Task<List<ParallelSearchResult>>[countSub];
120.                for(int i = 0; i < countSub; i++)
121.                {
122.                    List<string> tempTaskList =
word_list.GetRange(div_list[i].min, div_list[i].max - div_list[i].min);
123.                    tasks[i] = new
Task<List<ParallelSearchResult>>(
124.                        ArrayThreadTask,
125.                        new ParallelSearchThreadParam()
126.                        {
127.                            searchList = tempTaskList,
128.                            maxDist = max_dist,
129.                            threadNum = i,
130.                            wordTemp = word_temp
131.                        });
132.                    tasks[i].Start();
133.                }
134.                Task.WaitAll();
135.                extime.Stop();
```

```
136.         for (int i = 0; i < countSub; i++)
137.         {
138.             result_list.AddRange(tasks[i].Result);
139.         }
140.         label8.Text = "Вычисленное количество потоков: " +
            countSub.ToString();
141.         label8.Visible = true;
142.         listBox1.BeginUpdate();
143.         listBox1.Items.Clear();
144.         for (int i = 0; i < result_list.Count; i++)
145.         {
146.             listBox1.Items.Add(result_list[i].word + "
            [расстояние = " +
147.                 result_list[i].dist.ToString() + " | поток
            = " + result_list[i].threadNum.ToString() + "]");
148.         }
149.         listBox1.EndUpdate();
150.     }
151.     else
152.     {
153.         MessageBox.Show("Пожалуйста, выберите метод поиска
            слова!");
154.     }
155.     if (tempList.Count > 0)
156.     {
157.         listBox1.BeginUpdate();
158.         foreach (string str in tempList)
159.         {
160.             listBox1.Items.Add(str);
161.         }
162.         listBox1.EndUpdate();
163.     }
164.     extime.Stop();
165.     textBox4.Text = extime.Elapsed.ToString();
166. }
167.
168. private void radioButton2_CheckedChanged(object sender,
    EventArgs e)
169. {
170.     listBox1.BeginUpdate();
```



```
171.         listBox1.Items.Clear();
172.         listBox1.EndUpdate();
173.         if (radioButton2.Checked == true)
174.         {
175.             label6.Visible = true;
176.             textBox5.Visible = true;
177.             label7.Visible = true;
178.             textBox6.Visible = true;
179.         }
180.     }
181.
182.     private void radioButton1_CheckedChanged(object sender,
        EventArgs e)
183.     {
184.         listBox1.BeginUpdate();
185.         listBox1.Items.Clear();
186.         listBox1.EndUpdate();
187.         if (radioButton1.Checked == true)
188.         {
189.             label6.Visible = false;
190.             textBox5.Visible = false;
191.             label7.Visible = false;
192.             textBox6.Visible = false;
193.             label8.Visible = false;
194.         }
195.     }
196.
197.     private void buttonCreateReport_Click(object sender,
        EventArgs e)
198.     {
199.         string reportName = "Report_" +
            DateTime.Now.ToString("dd_MM_yyyy_hhmmss");
200.         SaveFileDialog fd = new SaveFileDialog();
201.         fd.FileName = reportName;
202.         fd.DefaultExt = ".html";
203.         fd.Filter = "HTML Reports|*.html";
204.         if (fd.ShowDialog() == DialogResult.OK)
205.         {
206.             StringBuilder rep = new StringBuilder();
207.             rep.AppendLine("<html>");
```

```

208.                rep.AppendLine("<head>");
209.                rep.AppendLine("<meta http-equiv='Content-Type'
content='text/html; charset=UTF-8' />");
210.                rep.AppendLine("<title> Отчет: " + fd.FileName +
"</title>");
211.                rep.AppendLine("</head>");
212.                rep.AppendLine("<body>");
213.                rep.AppendLine("<h1>" + "Отчет: " + fd.FileName +
"</h1>");
214.                rep.AppendLine("<table border='1'>");
215.                rep.AppendLine("<tr>");
216.                rep.AppendLine("<td>Время чтения из файла</td>");
217.                rep.Append("<td>" + textBox1.Text + "</td>\n" +
218.                "</tr>\n" +
219.                "<tr>\n" +
220.                "<td>Количество уникальных файлов в
словаре</td>\n" +
221.                "<td>" + word_list.Count.ToString() +
"</td>\n" +
222.                "</tr>\n" +
223.                "<tr>\n" +
224.                "<td>Слово для поиска</td>\n" +
225.                "<td>" + textBox3.Text + "</td>\n" +
226.                "</tr>\n");
227.                if (radioButton2.Checked == true)
228.                {
229.                    rep.Append(
230.                    "<tr>\n" +
231.                    "<td>Максимальное редакционное
расстояние</td>\n" +
232.                    "<td>" + textBox5.Text + "</td>\n" +
233.                    "</tr>\n" +
234.                    "<tr>\n" +
235.                    "<td>Время поиска методом редакционного
расстояния</td>\n");
236.                }
237.                else if (radioButton1.Checked == true)
238.                {
239.                    rep.Append(
240.                    "<tr>\n" +

```

```

241.                "<td>Время поиска методом поиска
                подстроки</td>\n");
242.            }
243.            rep.Append(
244.                "<td>" + textBox4.Text + "</td>\n" +
245.                "</tr>\n" +
246.                "<tr valign='top'>\n" +
247.                "<td>Результаты поиска</td>\n" +
248.                "<td>\n" +
249.                "<ul>\n");
250.            foreach (var x in listBox1.Items)
251.            {
252.                rep.AppendLine("<li>" + x.ToString() +
                "</li>");
253.            }
254.            rep.AppendLine("</ul>");
255.            rep.AppendLine("</td>");
256.            rep.AppendLine("</tr>");
257.            rep.AppendLine("</table>");
258.            rep.AppendLine("</html>");
259.            rep.AppendLine("</body>");
260.            File.AppendAllText(fd.FileName, rep.ToString());
261.            MessageBox.Show("Отчет сформирован. Файл: " +
                fd.FileName);
262.        }
263.    }
264. }
265. }

```

## 2) ExtraClasses.cs:

```

1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6.
7. namespace Lab4
8. {
9.     public class ParallelSearchResult

```

```

10.         {
11.             public string word { get; set; }
12.             public int dist { get; set; }
13.             public int threadNum { get; set; }
14.         }
15.     public class ParallelSearchThreadParam
16.     {
17.         public List<string> searchList { get; set; }
18.         public string wordTemp { get; set; }
19.         public int maxDist { get; set; }
20.         public int threadNum { get; set; }
21.     }
22.     public class MinMax
23.     {
24.         public int min { get; set; }
25.         public int max { get; set; }
26.
27.         public MinMax(int t_min, int t_max)
28.         {
29.             this.min = t_min;
30.             this.max = t_max;
31.         }
32.     }
33.     public static class SubArrays
34.     {
35.         /// <summary>
36.         /// Деление массива на последовательности
37.         /// </summary>
38.         /// <param name="beginIndex">Начальный индекс
39.         массива</param>
40.         /// <param name="endIndex">Конечный индекс массива</param>
41.         /// <param name="subArraysCount">Требуемое количество
42.         подмассивов</param>
43.         /// <returns>Список пар с индексами подмассивов</returns>
44.         public static List<MinMax> DivideSubArrays(int beginIndex,
45.             int endIndex, int subArraysCount)
46.         {

```

```

47.          //Если число элементов в массиве слишком мало для
           деления
48.          //то возвращается массив целиком
49.          if ((endIndex - beginIndex) <= subArraysCount)
50.          {
51.              result.Add(new MinMax(0, (endIndex -
           beginIndex)));
52.          }
53.          else
54.          {
55.              //Размер подмассива
56.              int delta = (endIndex - beginIndex) /
           subArraysCount;
57.              //Начало отсчета
58.              int currentBegin = beginIndex;
59.              //Пока размер подмассива укладывается в оставшуюся
           последовательность
60.              while ((endIndex - currentBegin) >= 2 * delta)
61.              {
62.                  //Формируем подмассив на основе начала
           последовательности
63.                  result.Add(new MinMax(currentBegin,
           currentBegin + delta));
64.                  //Сдвигаем начало последовательности вперед на
           размер подмассива
65.                  currentBegin += delta;
66.              }
67.              //Оставшийся фрагмент массива
68.              result.Add(new MinMax(currentBegin, endIndex));
69.          }
70.          //Возврат списка результатов
71.          return result;
72.      }
73.  }
74.  }

```

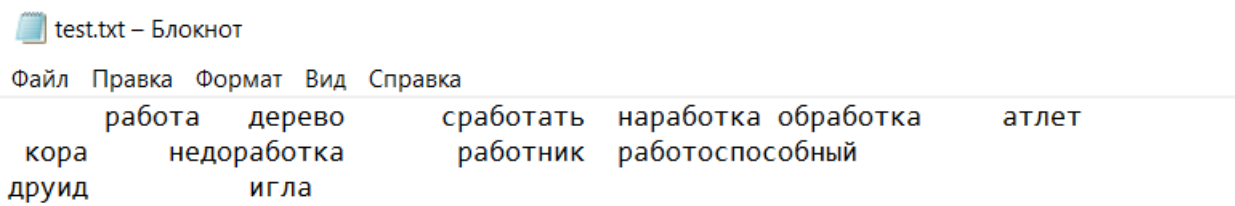
### 3) Вид конструктора формы:

The screenshot shows a Windows application window titled "Lab4". The window contains a form with the following elements:

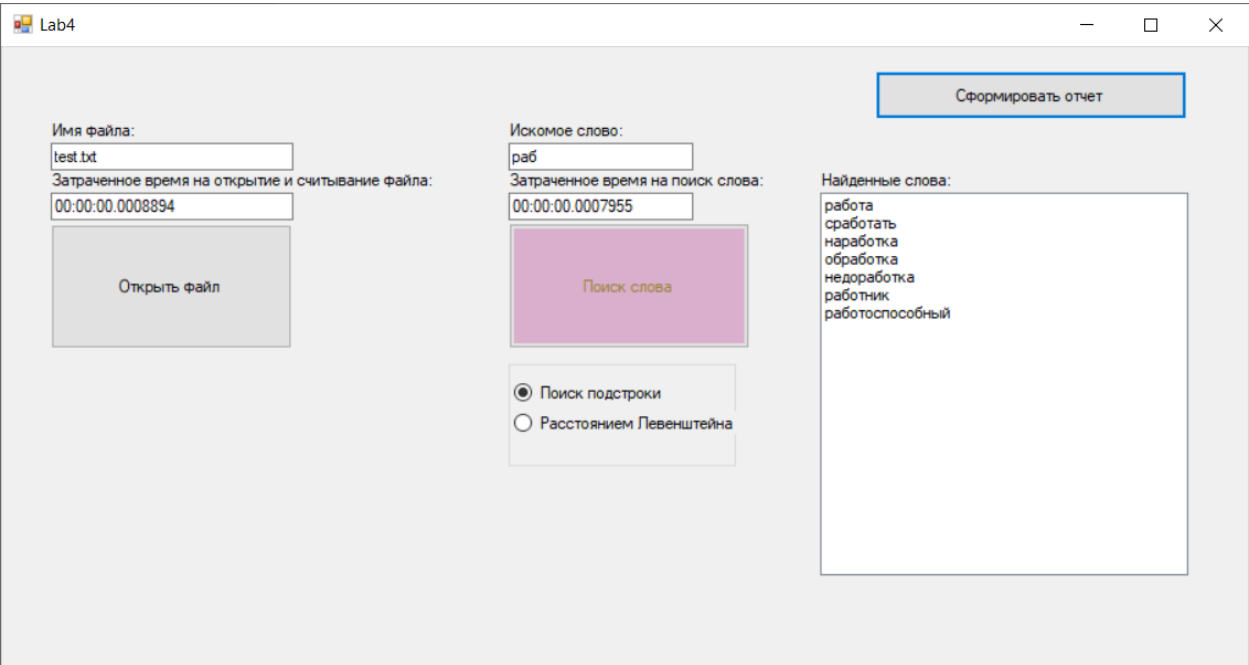
- File Input Section:**
  - Label: "Имя файла:" followed by a text input field.
  - Label: "Затраченное время на открытие и считывание файла:" followed by a text input field.
  - Button: "Открыть файл"
- Search Input Section:**
  - Label: "Искомое слово:" followed by a text input field.
  - Label: "Затраченное время на поиск слова:" followed by a text input field.
  - Button: "Поиск слова"
- Search Options Section:**
  - Two radio buttons: "Поиск подстроки" and "Расстоянием Левенштейна".
  - Label: "Максимальное расстояние:" followed by a text input field.
  - Label: "Количество потоков:" followed by a text input field.
  - Label: "Вычисленное количество потоков:" followed by a text input field.
- Report Section:**
  - Button: "Сформировать отчет"
- Output Section:**
  - Label: "Найденные слова:" followed by a list box containing the text "listBox1".

# Примеры выполнения программы:

Исходный файл:



Результат работы программы (метод поиска подстроки):



## Результат работы программы (метод редакционного расстояния):

Lab4

Имя файла:  
test.txt  
Затраченное время на открытие и считывание файла:  
00:00:00.0006473

Открыть файл

Искомое слово:  
раб  
Затраченное время на поиск слова:  
00:00:00.0008066

Поиск слова

☐ Поиск подстроки  
☒ Расстоянием Левенштейна

Максимальное расстояние:  
3

Количество потоков:  
3  
Вычисленное количество потоков: 3

Сформировать отчет

Найденные слова:  
работа [расстояние = 3 | поток = 0]  
кора [расстояние = 3 | поток = 1]

Lab4

Имя файла:  
test.txt  
Затраченное время на открытие и считывание файла:  
00:00:00.0008894

Открыть файл

Искомое слово:  
раб  
Затраченное время на поиск слова:  
00:00:00.0007254

Поиск слова

☐ Поиск подстроки  
☒ Расстоянием Левенштейна

Максимальное расстояние:  
10

Количество потоков:  
4  
Вычисленное количество потоков: 4

Сформировать отчет

Найденные слова:  
работа [расстояние = 3 | поток = 0]  
дерево [расстояние = 5 | поток = 0]  
сработать [расстояние = 6 | поток = 0]  
наработка [расстояние = 6 | поток = 1]  
обработка [расстояние = 6 | поток = 1]  
атлет [расстояние = 5 | поток = 1]  
кора [расстояние = 3 | поток = 2]  
недоработка [расстояние = 8 | поток = 2]  
работник [расстояние = 5 | поток = 2]  
друид [расстояние = 4 | поток = 3]  
игла [расстояние = 4 | поток = 3]



Примеры отчета:

## Отчет: D:\DZ\_Reports\Report\_15\_12\_2020\_052844.html

Время чтения из файла	00:00:00.0008894
Количество уникальных слов в файле	12
Слово для поиска	раб
Максимальное редакционное расстояние	10
Время поиска методом редакционного расстояния	00:00:00.0007254
Результаты поиска	<ul style="list-style-type: none"><li>• работа [расстояние = 3   поток = 0]</li><li>• дерево [расстояние = 5   поток = 0]</li><li>• сработать [расстояние = 6   поток = 0]</li><li>• наработка [расстояние = 6   поток = 1]</li><li>• обработка [расстояние = 6   поток = 1]</li><li>• атлет [расстояние = 5   поток = 1]</li><li>• кора [расстояние = 3   поток = 2]</li><li>• недоработка [расстояние = 8   поток = 2]</li><li>• работник [расстояние = 5   поток = 2]</li><li>• друид [расстояние = 4   поток = 3]</li><li>• игла [расстояние = 4   поток = 3]</li></ul>

## Отчет: D:\DZ\_Reports\Report\_15\_12\_2020\_053233.html

Время чтения из файла	00:00:00.0008894
Количество уникальных слов в файле	12
Слово для поиска	раб
Время поиска методом поиска подстроки	00:00:00.0007955
Результаты поиска	<ul style="list-style-type: none"><li>• работа</li><li>• сработать</li><li>• наработка</li><li>• обработка</li><li>• недоработка</li><li>• работник</li><li>• работоспособный</li></ul>