

**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

Бондаренко Иван

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю. Е.

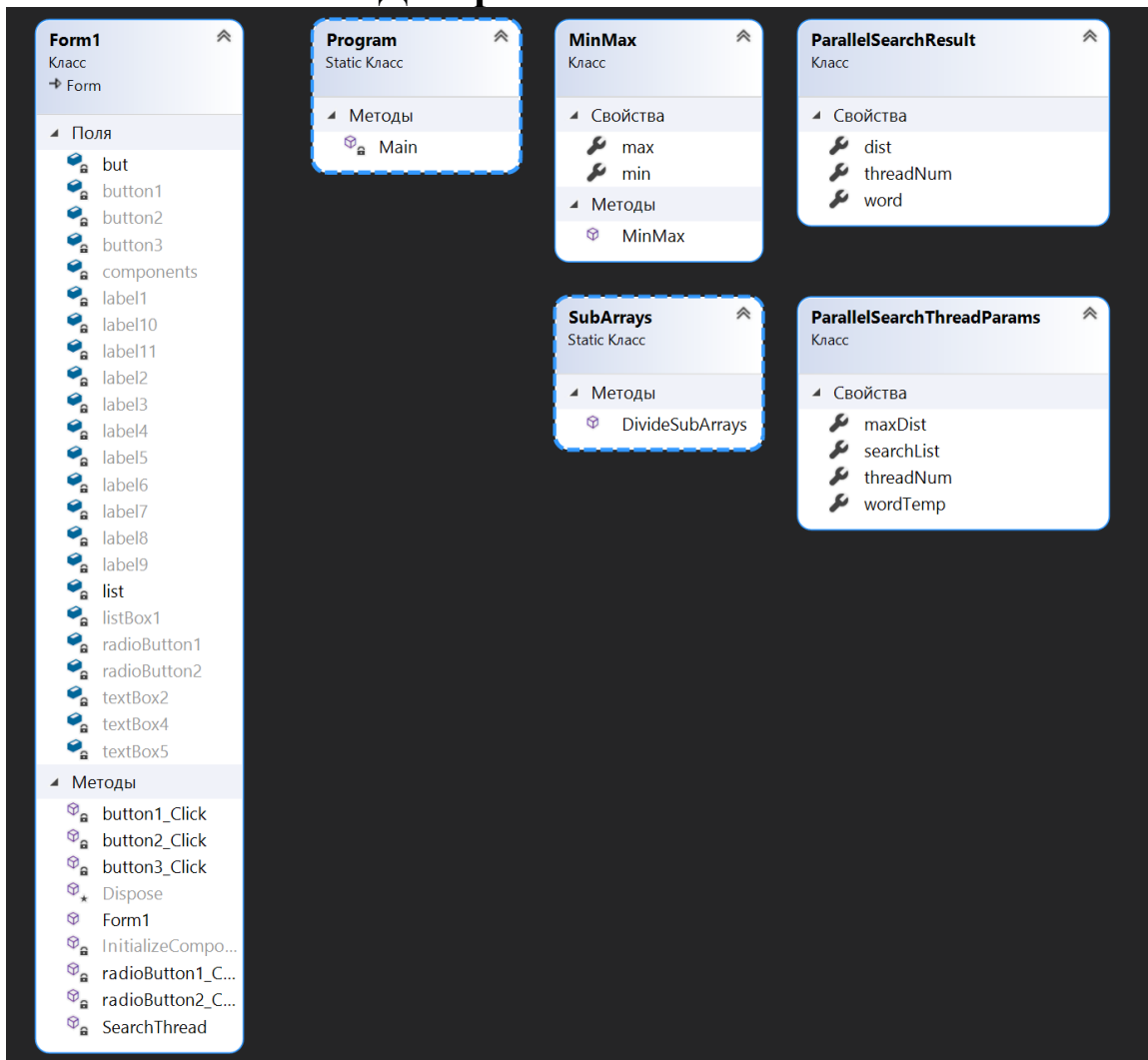
Подпись и дата:

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. Forms1.cs

```
1. using System;
2. using System.Collections.Generic;
3. using System.Windows.Forms;
4. using System.IO;
5. using System.Diagnostics;
6. using Lab5;
7. using System.Text;
8. using System.Threading.Tasks;
9.
10. namespace Lab4
11. {
12.     public partial class Form1 : Form
13.     {
14.         private List<string> list = new List<string>();
15.         private OpenFileDialog but;
16.         public Form1()
17.         {
18.             InitializeComponent();
19.             radioButton1.Checked = true;
20.             label2.Visible = false;
21.             textBox4.Visible = false;
22.             label6.Text = "";
23.             label5.Visible = false;
24.             label6.Visible = false;
25.             label8.Visible = false;
26.             label9.Visible = false;
27.             label10.Visible = false;
28.             label11.Visible = false;
29.         }
30.
31.         private void button1_Click(object sender, EventArgs e)
32.         {
33.             list.Clear();
34.             listBox1.BeginUpdate();
35.             listBox1.Items.Clear();
36.             listBox1.EndUpdate();
37.             Stopwatch sw = new Stopwatch();
38.             but = new OpenFileDialog();
39.             but.Filter = "Текстовый файл|*.txt";
40.             but.ShowDialog();
41.             try
42.             {
43.                 StreamReader read = new StreamReader(but.FileName);
44.                 sw.Start();
45.                 label1.Text = but.FileName;
46.                 string res = read.ReadToEnd();
47.                 string[] resArr;
48.                 char[] seps = { ' ', '.', ',', '?', ')', '(', '!', '\\n', '\\f', '\\t', '\\/', '\\\\' };
49.                 resArr = res.Split(seps);
50.                 foreach (string a in resArr)
```

```

51.         {
52.             string word = a.Trim();
53.             if (!list.Contains(word) && word != "")
54.                 list.Add(word.ToLower());
55.         }
56.         sw.Stop();
57.         label6.Text = sw.Elapsed.ToString();
58.         label5.Visible = true;
59.         label6.Visible = true;
60.     }
61.     catch
62.     {
63.         label6.Text = "Файл не выбран";
64.     }
65. }
66.
67. private List<ParallelSearchResult> SearchThread(object param)
68. {
69.     ParallelSearchThreadParams real_param = param as ParallelSearchThreadParams;
70.     List<ParallelSearchResult> res = new List<ParallelSearchResult>();
71.     foreach (string word in real_param.searchList)
72.     {
73.         int dis;
74.         try
75.         { dis = Fisher.GetLen(word.ToLower(), real_param.wordTemp); }
76.         catch
77.         {
78.             MessageBox.Show(word.ToLower());
79.             continue;
80.         }
81.         if (dis <= real_param.maxDist)
82.         {
83.             ParallelSearchResult temp = new ParallelSearchResult()
84.             {
85.                 dist = dis,
86.                 word = word,
87.                 threadNum = real_param.threadNum
88.             };
89.             res.Add(temp);
90.         }
91.     }
92.     return res;
93. }
94.
95. private void button2_Click(object sender, EventArgs e)
96. {
97.     listBox1.BeginUpdate();
98.     listBox1.Items.Clear();
99.     listBox1.EndUpdate();
100.    if (radioButton1.Checked)
101.    {
102.        listBox1.BeginUpdate();
103.        listBox1.Items.Clear();
104.        Stopwatch sw = new Stopwatch();
105.        sw.Start();

```

```
106.         for (int i = 0; i < list.Count; i++)
107.         {
108.             if (list[i].Contains(textBox2.Text))
109.                 listBox1.Items.Add(list[i]);
110.         }
111.         sw.Stop();
112.         label9.Text = sw.Elapsed.ToString();
113.         sw.Reset();
114.         if (listBox1.Items.Count == 0)
115.         {
116.             label9.Text = "Не найдено!";
117.         }
118.         label8.Visible = true;
119.         label9.Visible = true;
120.         listBox1.EndUpdate();
121.     }
122.     else
123.     {
124.         if (radioButton2.Checked)
125.         {
126.             int thread_c = -1;
127.             int dis = -1;
128.             try
129.             {
130.                 dis = Convert.ToInt32(textBox4.Text);
131.                 if (dis <= 0)
132.                     throw new Exception();
133.             }
134.             catch
135.             {
136.                 MessageBox.Show("Введите максимальную длину > 0");
137.                 return;
138.             }
139.         }
140.         try
141.         {
142.             thread_c = Convert.ToInt32(textBox5.Text);
143.             if (thread_c <= 0)
144.                 throw new Exception();
145.         }
146.         catch
147.         {
148.             MessageBox.Show("Введите число потоков");
149.             return;
150.         }
151.     }
152. }
153.
154. Stopwatch sw = new Stopwatch();
155. sw.Start();
156. List<ParallelSearchResult> results = new List<ParallelSearchResult>();
157. List<MinMax> iterators = SubArrays.DivideSubArrays(0, list.Count, thread_c);
158. int sub_c = iterators.Count;
159. Task<List<ParallelSearchResult>>[] task_list = new
160.     Task<List<ParallelSearchResult>>[sub_c];
```

```

161.         for (int i = 0; i < sub_c; i++)
162.         {
163.             int delta = iterators[i].max - iterators[i].min;
164.             List<string> temp_list = list.GetRange(iterators[i].min, delta);
165.             task_list[i] = new Task<List<ParallelSearchResult>>(
166.                 SearchThread,
167.                 new ParallelSearchThreadParams()
168.                 {
169.                     wordTemp = textBox2.Text,
170.                     searchList = temp_list,
171.                     maxDist = dis,
172.                     threadNum = i
173.                 }
174.             );
175.             task_list[i].Start();
176.         }
177.         Task.WaitAll();
178.         sw.Stop();
179.         label9.Text = sw.Elapsed.ToString();
180.         label11.Text = thread_c.ToString();
181.         sw.Reset();
182.         for (int i = 0; i < sub_c; i++)
183.         {
184.             results.AddRange(task_list[i].Result);
185.         }
186.         listBox1.BeginUpdate();
187.         listBox1.Items.Clear();
188.         foreach (var el in results)
189.         {
190.             listBox1.Items.Add(el.word + " [расстояние = " + el.dist.ToString() +
191.                                 + " | поток = " + el.threadNum.ToString() + "]);
192.         }
193.         listBox1.EndUpdate();
194.         if (listBox1.Items.Count == 0)
195.         {
196.             label9.Text = "Не найдено!";
197.             label10.Visible = false;
198.             label11.Visible = false;
199.         }
200.         else
201.         {
202.             label10.Visible = true;
203.             label11.Visible = true;
204.         }
205.         label8.Visible = true;
206.         label9.Visible = true;
207.     }
208. }
209. }
210.
211. private void radioButton1_CheckedChanged(object sender, EventArgs e)
212. {
213.     listBox1.BeginUpdate();
214.     listBox1.Items.Clear();
215.     listBox1.EndUpdate();

```

```

216.         label4.Visible = false;
217.         label2.Visible = false;
218.         textBox4.Visible = false;
219.         textBox5.Visible = false;
220.         label10.Visible = false;
221.         label11.Visible = false;
222.     }
223.
224.     private void radioButton2_CheckedChanged(object sender, EventArgs e)
225.     {
226.         listBox1.BeginUpdate();
227.         listBox1.Items.Clear();
228.         listBox1.EndUpdate();
229.         label4.Visible = true;
230.         label2.Visible = true;
231.         textBox4.Visible = true;
232.         textBox5.Visible = true;
233.         label10.Visible = false;
234.         label11.Visible = false;
235.     }
236.
237.     private void button3_Click(object sender, EventArgs e)
238.     {
239.         StringBuilder str = new StringBuilder();
240.         string log_name = "log_" + DateTime.Now.ToString("dd_MM_yyyy_hhmmss");
241.         SaveFileDialog save = new SaveFileDialog();
242.         save.FileName = log_name;
243.         save.DefaultExt = ".html";
244.         save.Filter = "HTML Logs|*.html";
245.         if(save.ShowDialog() == DialogResult.OK)
246.         {
247.             str.AppendLine("<html>");
248.             str.AppendLine("<head>");
249.             str.AppendLine("<meta http-equiv='Content-Type' content='text/html; charset=UTF-8' />");
250.             str.AppendLine("<title> Отчет: " + save.FileName + "</title>");
251.             str.AppendLine("</head>");
252.             str.AppendLine("<body>");
253.             str.AppendLine("<h1> " + "Отчет: " + save.FileName + "</h1>");
254.             str.AppendLine("<table border='1'>");
255.             str.AppendLine("<tr>");
256.             str.AppendLine("<td>Файл</td>");
257.             str.Append("<td>" + label1.Text + "</td>\n" +
258.                 "</tr>\n" +
259.                 "<tr>\n" +
260.                 "<td>Время открытия файла</td>\n" +
261.                 "<td>" + label6.Text + "</td>\n" +
262.                 "</tr>\n" +
263.                 "<tr>\n" +
264.                 "<td>Исследуемое слово</td>\n" +
265.                 "<td>" + textBox2.Text + "</td>\n" +
266.                 "</tr>\n" +
267.                 "<tr>\n" +
268.                 "<td>Время поиска слова</td>\n" +
269.                 "<td>" + label9.Text + "</td>\n" +
270.                 "</tr>\n");

```



```

271.         str.Append(
272.             "<tr>\n" +
273.             "<td>Метод поиска</td>\n");
274.         if (radioButton2.Checked == true)
275.         {
276.             str.Append(
277.                 "<td>Расстояние Левенштейна</td>\n" +
278.                 "</tr>\n" +
279.                 "<tr>\n" +
280.                 "<td>Максимальная длина</td>\n" +
281.                 "<td>" + textBox4.Text + "</td>\n" +
282.                 "</tr>\n" +
283.                 "<tr>\n" +
284.                 "<td>Количество потоков</td>\n" +
285.                 "<td>" + textBox5.Text + "</td>\n" +
286.                 "</tr>\n" +
287.                 "<tr>\n" +
288.                 "<td>Использовано потоков</td>\n" +
289.                 "<td>" + label11.Text + "</td>\n" +
290.                 "</tr>\n");
291.         }
292.         else if (radioButton1.Checked == true)
293.         {
294.             str.Append(
295.                 "<td>Поиск подстроки</td>\n" +
296.                 "</tr>\n");
297.         }
298.         str.Append(
299.             "<tr valign='top'>\n" +
300.             "<td>Результаты поиска</td>\n" +
301.             "<td>\n" +
302.             "<ul>\n");
303.         foreach (var word in listBox1.Items)
304.         {
305.             str.AppendLine("<li>" + word.ToString() + "</li>");
306.         }
307.         str.AppendLine("</ul>");
308.         str.AppendLine("</td>");
309.         str.AppendLine("</tr>");
310.         str.AppendLine("</table>");
311.         str.AppendLine("</html>");
312.         str.AppendLine("</body>");
313.         File.AppendAllText(save.FileName, str.ToString());
314.         MessageBox.Show("Отчет был успешно создан в: " + save.FileName);
315.     }
316. }
317. }
318. }

```

2. Parallel.cs

```

1.     using System.Collections.Generic;
2.
3.     namespace Lab4

```

```

4.  {
5.      public class ParallelSearchResult
6.      {
7.          public string word { get; set; }
8.          public int dist { get; set; }
9.          public int threadNum { get; set; }
10.     }
11.     public class ParallelSearchThreadParams
12.     {
13.         public List<string> searchList { get; set; }
14.         public string wordTemp { get; set; }
15.         public int maxDist { get; set; }
16.         public int threadNum { get; set; }
17.     }
18.     public class MinMax
19.     {
20.         public int min { get; set; }
21.         public int max { get; set; }
22.
23.         public MinMax(int t_min, int t_max)
24.         {
25.             this.min = t_min;
26.             this.max = t_max;
27.         }
28.     }
29.     public static class SubArrays
30.     {
31.         /// <summary>
32.         /// Деление массива на последовательности
33.         /// </summary>
34.         /// <param name="beginIndex">Начальный индекс массива</param>
35.         /// <param name="endIndex">Конечный индекс массива</param>
36.         /// <param name="subArraysCount">Требуемое количество подмассивов</param>
37.         /// <returns>Список пар с индексами подмассивов</returns>
38.         public static List<MinMax> DivideSubArrays(int beginIndex, int endIndex, int subArraysCount)
39.         {
40.             //Результирующий список пар с индексами подмассивов
41.             List<MinMax> result = new List<MinMax>();
42.
43.             //Если число элементов в массиве слишком мало для деления
44.             //то возвращается массив целиком
45.             if ((endIndex - beginIndex) <= subArraysCount)
46.             {
47.                 result.Add(new MinMax(0, (endIndex - beginIndex)));
48.             }
49.             else
50.             {
51.                 //Размер подмассива
52.                 int delta = (endIndex - beginIndex) / subArraysCount;
53.                 //Начало отсчета
54.                 int currentBegin = beginIndex;
55.                 //Пока размер подмассива укладывается в оставшуюся последовательность
56.                 while ((endIndex - currentBegin) >= 2 * delta)
57.                 {
58.                     //Формируем подмассив на основе начала последовательности

```

```
59.         result.Add(new MinMax(currentBegin, currentBegin + delta));
60.         //Сдвигаем начало последовательности вперед на размер подмассива
61.         currentBegin += delta;
62.     }
63.     //Оставшийся фрагмент массива
64.     result.Add(new MinMax(currentBegin, endIndex));
65. }
66. //Возврат списка результатов
67. return result;
68. }
69. }
70. }
```

Использована библиотека Lab5, разработанная в лабораторной работе №5.

Примеры выполнения программы:

Исходный текст:

Расстояние.txt – Блокнот

Файл Правка Формат Вид Справка

водосток
водоскатный
обводняться
паводок
приводнить
водоворот
водопользование
водный
водяной
водянистый
водомёт
водохранилище
водохранилищный
наводнять
наводнить
наводнение
водолей
водичка
водица
водительский
водопроводчик
водопроводный
пчеловодство
водопровод
уводить
пчеловод
путеводитель
сопроводить
отводить
водопровод
вводный
развод
птицеводство
проводной
водиться
проводиться
доводчик
привод

Результат работы программы (метод поиска подстроки):

Lab4

Путь: C:\Users\Иван\source\repos\Labs_2\Lab4\Расстояние.txt

Файл

Время открытия: 00:00:00.0007186

Найти слово

Введите слово: водн

Время поиска: 00:00:00.0004984

☒ Поиск подстроки
☐ Вычисление расстояния

обводняться
приводнить
водный
наводнять
наводнить
наводнение
водопроводный
вводный
проводной

Создать отчет

Результат работы программы (метод редакционного расстояния):

Lab4

Путь: C:\Users\Иван\source\repos\Labs_2\DZ\Lab4\Расстояние.txt

Файл

Время открытия: 00:00:00.0007186

Найти слово

Введите слово: води

Время поиска: 00:00:00.0114005

☐ Поиск подстроки

☒ Вычисление расстояния

Максимальная длина: 3

Количество потоков: 2

Использовано потоков: 2

Создать отчет

водный [расстояние = 3 | поток = 0]
водичка [расстояние = 3 | поток = 0]
водица [расстояние = 2 | поток = 1]
уводить [расстояние = 3 | поток = 1]

Lab4

Путь: C:\Users\Иван\source\repos\Labs_2\DZ\Lab4\Расстояние.txt

Файл

Время открытия: 00:00:00.0007186

Найти слово

Введите слово: водиц

Время поиска: 00:00:00.0398682

☐ Поиск подстроки

☒ Вычисление расстояния

Максимальная длина: 6

Количество потоков: 8

Использовано потоков: 8

Создать отчет

водосток [расстояние = 5 | поток = 0]
паводок [расстояние = 4 | поток = 0]
приводить [расстояние = 6 | поток = 1]
водоворот [расстояние = 6 | поток = 1]
водный [расстояние = 3 | поток = 1]
водяной [расстояние = 4 | поток = 2]
водянистый [расстояние = 6 | поток = 2]
водомёт [расстояние = 4 | поток = 2]
наводнять [расстояние = 6 | поток = 3]
наводнить [расстояние = 5 | поток = 3]
наводнение [расстояние = 6 | поток = 3]
водолей [расстояние = 4 | поток = 4]
водичка [расстояние = 3 | поток = 4]
водица [расстояние = 1 | поток = 4]
уводить [расстояние = 3 | поток = 6]
отводить [расстояние = 4 | поток = 7]
вводный [расстояние = 4 | поток = 7]
развод [расстояние = 5 | поток = 7]
проводной [расстояние = 6 | поток = 8]
водиться [расстояние = 4 | поток = 8]
доводчик [расстояние = 4 | поток = 8]

Сформированные отчеты:

Отчет: C:\Users\Иван\Desktop\log_26_12_2020_035615.html

Файл	C:\Users\Иван\source\repos\Labs_2\DZ\Lab4\Расстояние.txt
Время открытия файла	00:00:00.0008181
Исследуемое слово	вода
Время поиска слова	00:00:00.0005505
Метод поиска	Расстояние Левенштейна
Максимальная длина	7
Количество потоков	6
Использовано потоков	6
Результаты поиска	<ul style="list-style-type: none"> • водосток [расстояние = 5 поток = 0] • водоскатный [расстояние = 7 поток = 0] • паводок [расстояние = 4 поток = 0] • приводнить [расстояние = 7 поток = 0] • водоворот [расстояние = 6 поток = 0] • водный [расстояние = 3 поток = 1] • водяной [расстояние = 4 поток = 1] • водянистый [расстояние = 7 поток = 1] • водомёт [расстояние = 4 поток = 1] • наводнять [расстояние = 6 поток = 2] • наводнить [расстояние = 6 поток = 2] • наводнение [расстояние = 7 поток = 2] • водолей [расстояние = 4 поток = 2] • водичка [расстояние = 3 поток = 2] • водица [расстояние = 2 поток = 3] • водопровод [расстояние = 7 поток = 3] • уводить [расстояние = 4 поток = 4] • пчеловод [расстояние = 6 поток = 4] • отводить [расстояние = 5 поток = 4] • вводный [расстояние = 4 поток = 4] • развод [расстояние = 4 поток = 5] • проводной [расстояние = 6 поток = 5] • водиться [расстояние = 5 поток = 5] • доводчик [расстояние = 5 поток = 5] • привод [расстояние = 4 поток = 5]

Отчет: C:\Users\Иван\Desktop\log_26_12_2020_035628.html

Файл	C:\Users\Иван\source\repos\Labs_2\DZ\Lab4\Расстояние.txt
Время открытия файла	00:00:00.0008181
Исследуемое слово	водн
Время поиска слова	00:00:00.0001590
Метод поиска	Поиск подстроки
Результаты поиска	<ul style="list-style-type: none"> • обводняться • приводнить • водный • наводнять • наводнить • наводнение • водопроводный • вводный • проводной