

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

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №5

Выполнил:

студент группы ИУ5-31

Бондаренко Иван

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю. Е.

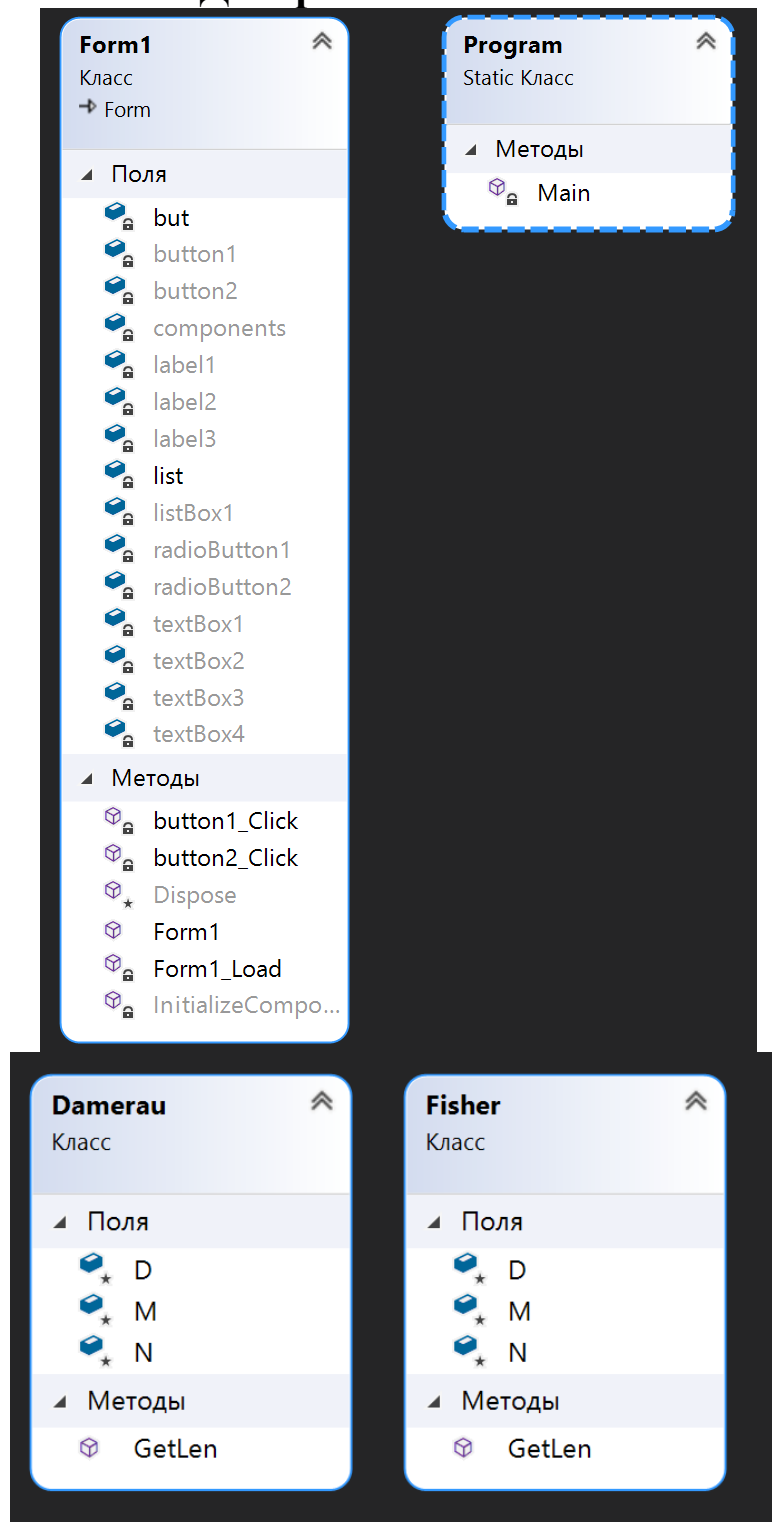
Подпись и дата:

Москва, 2020 г.

### **Описание задания:**

- Программа должна быть разработана в виде библиотеки классов на языке C#.
- Использовать самый простой вариант алгоритма без оптимизации.
- Дополнительно возможно реализовать вычисление расстояния Дamerau- Левенштейна (с учетом перестановок соседних символов).
- Модифицировать предыдущую лабораторную работу, вместо поиска подстроки используется вычисление расстояния Левенштейна.
- Предусмотреть отдельное поле ввода для максимального расстояния. Если расстояние Левенштейна между двумя строками больше максимального, то строки считаются несовпадающими и не выводятся в список результатов.

## Диаграмма классов:



## Текст программы:

### 1. Форма:

```
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.
8. namespace Lab4
9. {
10.     public partial class Form1 : Form
11.     {
12.         private List<string> list = new List<string>();
13.         private OpenFileDialog but;
14.         public Form1()
15.         {
16.             InitializeComponent();
17.             radioButton1.Checked = true;
18.         }
19.
20.         private void Form1_Load(object sender, EventArgs e)
21.         {
22.         }
23.
24.         private void button1_Click(object sender, EventArgs e)
25.         {
26.             list.Clear();
27.             listBox1.BeginUpdate();
28.             listBox1.Items.Clear();
29.             listBox1.EndUpdate();
30.             Stopwatch sw = new Stopwatch();
31.             but = new OpenFileDialog();
32.             but.Filter = "Текстовый файл|*.txt";
33.             but.ShowDialog();
34.             try
35.             {
36.                 StreamReader read = new StreamReader(but.FileName);
37.                 sw.Start();
38.                 label1.Text = but.FileName;
39.                 string res = read.ReadToEnd();
40.                 string[] resArr;
41.                 resArr = res.Split();
42.                 foreach (string a in resArr)
43.                 {
44.                     string word;
45.                     word = a.Replace(", ", "");
46.                     word = word.Replace(" ", "");
47.                     word = word.Replace(".", "");
48.                     word = word.Replace(";", "");
49.                     word = word.Replace(":", "");
50.                     word = word.Replace("(", "");
51.                     word = word.Replace(")", "");
52.                     if (!list.Contains(word) && word != "")
53.                         list.Add(word.ToLower());
54.                 }
55.                 sw.Stop();
56.                 textBox1.Text = sw.Elapsed.ToString();
57.                 listBox1.BeginUpdate();
58.                 foreach (string a in list)
59.                 {
60.                     listBox1.Items.Add(a);
61.                 }
62.                 listBox1.EndUpdate();
63.             }
64.             catch
65.             {
```

```

66.         textBox1.Text = "Файл не выбран";
67.     }
68. }
69. private void button2_Click(object sender, EventArgs e)
70. {
71.     if (radioButton1.Checked)
72.     {
73.         listBox1.BeginUpdate();
74.         listBox1.Items.Clear();
75.         Stopwatch sw = new Stopwatch();
76.         sw.Start();
77.         for (int i = 0; i < list.Count; i++)
78.         {
79.             if (list[i].Contains(textBox2.Text))
80.                 listBox1.Items.Add(list[i]);
81.         }
82.         sw.Stop();
83.         textBox3.Text = sw.Elapsed.ToString();
84.         sw.Reset();
85.         if (listBox1.Items.Count == 0)
86.         {
87.             textBox3.Text = "Не найдено!";
88.         }
89.         listBox1.EndUpdate();
90.     }
91.     else
92.     {
93.         if (radioButton2.Checked)
94.         {
95.             if (textBox4.Text == "" || Convert.ToInt32(textBox4.Text) <= 0)
96.             {
97.                 MessageBox.Show("Введите максимальную длину > 0");
98.             }
99.             else
100.            {
101.                listBox1.BeginUpdate();
102.                listBox1.Items.Clear();
103.                Stopwatch sw = new Stopwatch();
104.                sw.Start();
105.                for (int i = 0; i < list.Count; i++)
106.                {
107.                    if (Fisher.GetLen(list[i], textBox2.Text) <=
108.                        Convert.ToInt32(textBox4.Text))
109.                        listBox1.Items.Add(list[i]);
110.                }
111.                sw.Stop();
112.                textBox3.Text = sw.Elapsed.ToString();
113.                sw.Reset();
114.                if (listBox1.Items.Count == 0)
115.                {
116.                    textBox3.Text = "Не найдено!";
117.                }
118.                listBox1.EndUpdate();
119.            }
120.        }
121.    }
122. }
123. }
124. }

```

2) Вид конструктора формы:

Путь:

Файл

Найти

listBox1

Максимальная длина:

☐ Поиск подстроки

☐ Вычисление расстояния

## 2) Lib.cs:

```

1. using System;
2. using System.Collections.Generic;
3.
4. namespace Lab5
5. {
6.     public class Fisher
7.     {
8.         static protected int M, N;
9.         static protected int[,] D;
10.        public static int GetLen(string S1, string S2)
11.        {
12.            M = S1.Length+1;
13.            N = S2.Length+1;
14.            D = new int[M, N];
15.            for (int i = 0; i < M; i++)
16.                D[i, 0] = i;
17.            for (int j = 0; j < N; j++)
18.                D[0, j] = j;
19.            for (int i = 1; i < M; i++)
20.            {
21.                for(int j = 1; j<N; j++)
22.                {
23.                    int difference = (S1[i-1] == S2[j-1]) ? 0 : 1;
24.                    int res = Math.Min(D[i - 1,j]+1, D[i,j - 1]+1);
25.                    res = Math.Min(res, D[i - 1,j - 1]+difference);
26.                    D[i, j] = res;
27.                }
28.            }
29.            return D[M-1, N-1];
30.        }
31.    }
32.
33.    public class Damerau
34.    {
35.        static protected int M, N;
36.        static protected int[,] D;
37.        public static int GetLen(string S1, string S2, int dCost, int iCost, int rCost,
38.        int tCost)
39.        {
40.            Dictionary<char, int> dict = new Dictionary<char, int>();
41.            M = S1.Length;
42.            N = S2.Length;
43.            if (S1 == "")
44.                if (S2 == "")
45.                    return 0;

```

```

45.         else
46.             return N;
47.     else if (S2 == "")
48.         return M;
49.
50.     D = new int[M + 1, N + 1];
51.     int Con = Math.Max(dCost, iCost);
52.     Con = Math.Max(Con, rCost);
53.     Con = Math.Max(Con, tCost);
54.     Con *= (M + N);
55.
56.     D[0, 0] = Con;
57.     for (int i = 0; i < M; i++)
58.     {
59.         D[i + 1, 1] = i * (dCost);
60.         D[i + 1, 0] = Con;
61.     }
62.     for (int i = 0; i < N; i++)
63.     {
64.         D[1, i+1] = i * (iCost);
65.         D[0, i+1] = Con;
66.     }
67.
68.     foreach (char let in (S1 + S2))
69.         dict[let] = 0;
70.
71.     for(int i = 1; i<M; i++)
72.     {
73.         int last = 0;
74.         for (int j = 1; j < N; j++)
75.         {
76.             int k = dict[S2[j]];
77.             int m = last;
78.             if (S1[i] == S2[j])
79.             {
80.                 D[i + 1, j + 1] = D[i, j];
81.                 last = j;
82.             }
83.             else
84.             {
85.                 int res = Math.Min(D[i, j] + rCost, D[i + 1, j] + iCost);
86.                 res = Math.Min(res, D[i, j + 1] + dCost);
87.                 D[i + 1, j + 1] = res;
88.             }
89.             D[i + 1, j + 1] = Math.Min(D[i + 1, j + 1], D[k, m] + (i - k - 1) *
dCost + tCost + (j - m - 1) * iCost);
90.         }
91.         dict[S1[i]] = i;
92.     }
93.     return D[M, N];
94.
95.     }
96. }
97. }

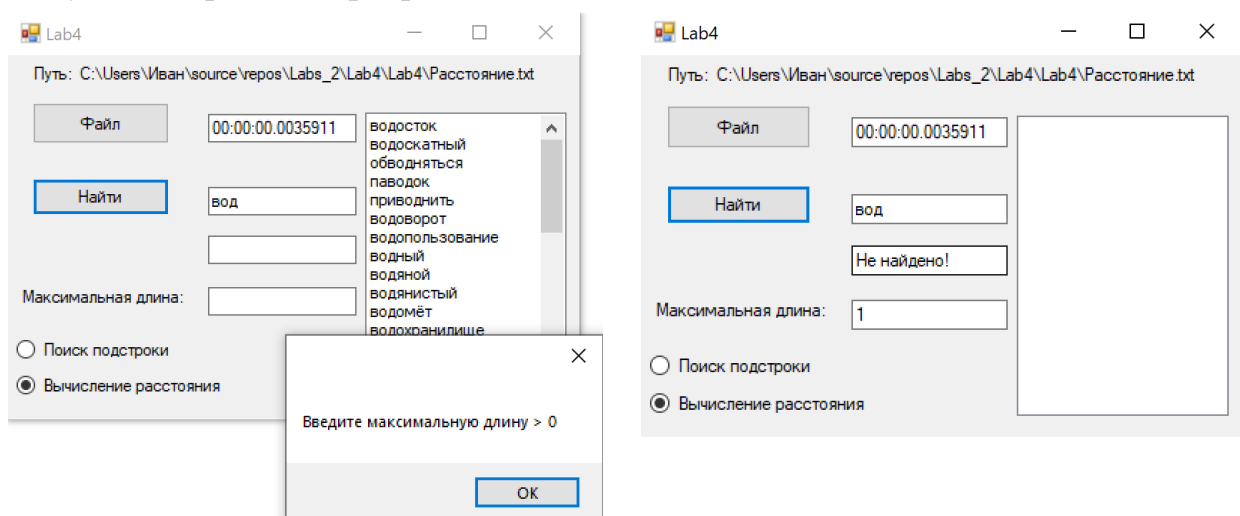
```

# Пример выполнения программы

## Исходный файл:

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

## Результаты работы программы:





Lab4

Путь: C:\Users\Иван\source\repos\Labs\_2\Lab4\Lab4\Расстояние.txt

Файл 00:00:00.0035911 водный  
водичка  
водица

Найти вода 00:00:00.0002583

Максимальная длина: 3

☐ Поиск подстроки  
☒ Вычисление расстояния

Lab4

Путь: C:\Users\Иван\source\repos\Labs\_2\Lab4\Lab4\Расстояние.txt

Файл 00:00:00.0035911 паводок  
водный  
водяной  
водоём  
водолей  
водичка  
водица  
уводить  
водный  
развод  
привод

Найти вода 00:00:00.0002985

Максимальная длина: 4

☐ Поиск подстроки  
☒ Вычисление расстояния

Lab4

Путь: C:\Users\Иван\source\repos\Labs\_2\Lab4\Lab4\Расстояние.txt

Файл 00:00:00.0035911 водица

Найти водица 00:00:00.0002669

Максимальная длина: 1

☐ Поиск подстроки  
☒ Вычисление расстояния

Lab4

Путь: C:\Users\Иван\source\repos\Labs\_2\Lab4\Lab4\Расстояние.txt

Файл 00:00:00.0035911 водный  
водичка  
водица  
уводить

Найти водица 00:00:00.0003692

Максимальная длина: 3

☐ Поиск подстроки  
☒ Вычисление расстояния

Lab4

Путь: C:\Users\Иван\source\repos\Labs\_2\Lab4\Lab4\Расстояние.txt

Файл 00:00:00.0035911 водный  
водичка  
водица

Найти водца 00:00:00.0002738

Максимальная длина: 3

☐ Поиск подстроки  
☒ Вычисление расстояния