

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

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

Курс «Базовые компоненты интернет технологий»  
Отчет по лабораторной работе №5

Выполнил:  
студент группы РТ5-31Б  
Сысоев Александр

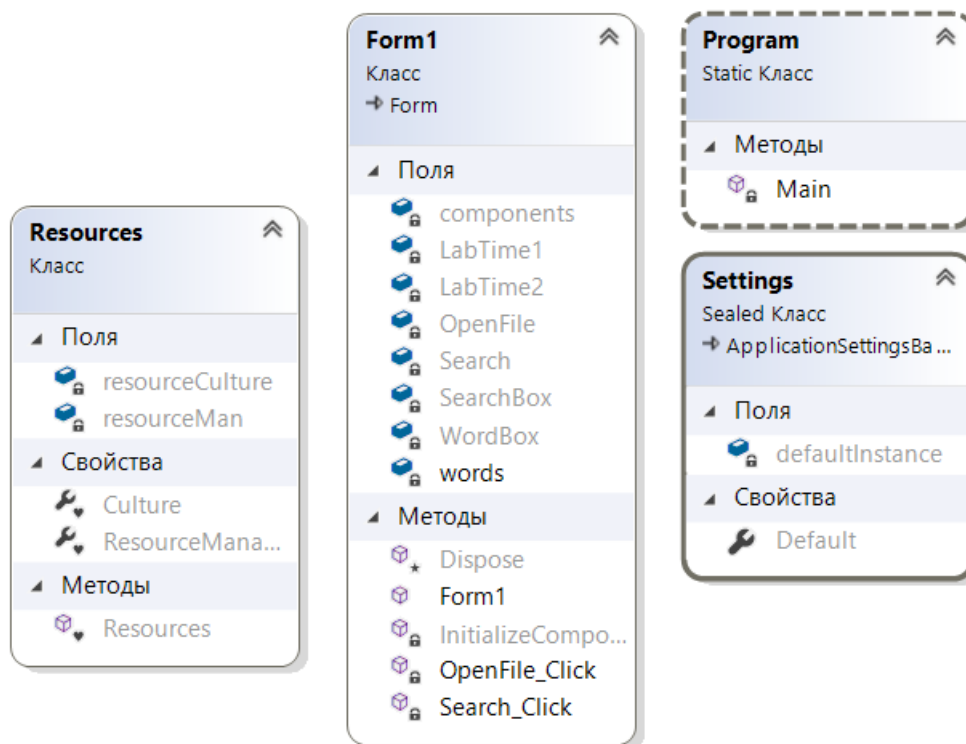
Проверил:  
доцент каф. ИУ5  
Гапанюк Ю.Е.

# Описание задания

Разработать программу, реализующую вычисление расстояния Левенштейна с использованием алгоритма Вагнера-Фишера.

1. Программа должна быть разработана в виде библиотеки классов на языке C#.
2. Использовать самый простой вариант алгоритма без оптимизации.
3. Дополнительно возможно реализовать вычисление расстояния Дamerau-Левенштейна (с учетом перестановок соседних символов).
4. Модифицировать предыдущую лабораторную работу, вместо поиска подстроки используется вычисление расстояния Левенштейна.
5. 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

### 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;

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

        public Form1()
        {
            InitializeComponent();
            LabTime1.Text = "";
            LabTime2.Text = "";
            LabDist.Text = "Введите минимальное расстояние Левенштейна.";
        }

        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;
        if (SearchBox.Text != "" && words != null && int.TryParse(DistBox.Text, out
dist))
        {
            Stopwatch time = Stopwatch.StartNew();
            WordBox.Items.Clear();
            WordBox.BeginUpdate();
            foreach (string s in words)
                if (Lab5.Lib.Levenshtein.Calculate(s, SearchBox.Text) <= dist)
                    WordBox.Items.Add(s);
            WordBox.EndUpdate();
            time.Stop();
            LabTime2.Text = "Время загрузки: " + time.Elapsed.ToString();
        }
    }
}

```

## Результат работы программы

