

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Московский государственный технический университет
имени Н.Э. Баумана»
(национальный исследовательский университет)
(МГТУ им. Н.Э. Баумана)



Факультет Информатика и системы управления

Кафедра Системы обработки информации и управления

Лабораторная работа №5

Студент Бессонова Ксения Сергеевна

Группа ИУ5-31Б

Название дисциплины Базовые компоненты интернет-технологий

Преподаватель

Гапанюк Ю.Е.
Фамилия И.О.

подпись

Москва 2020

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 Дameraу-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.

Текст программы:

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Diagnostics;

namespace Lab4
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
```

```
{  
}
```

```
public List<string> List { get; set; } = new List<string>();
```

```
private void ButtonRead_Click(object sender, EventArgs e)  
{
```

```
    OpenFileDialog fileDialog = new OpenFileDialog  
    {  
        InitialDirectory = Directory.GetCurrentDirectory(),  
        Filter = "Файлы *.txt|*.txt"  
    };  
    fileDialog.ShowDialog();
```

```
    if (fileDialog.FileName.Length == 0)  
    {  
        return;  
    }
```

```
    Stopwatch time = new Stopwatch();  
    time.Start();
```

```
    string text = File.ReadAllText(fileDialog.FileName);
```

```
    foreach (var word in text.Split())  
    {  
        if (!List.Contains(word))  
        {  
            List.Add(word);  
        }  
    }
```

```
    time.Stop();
```

```
    Double result = time.Elapsed.TotalMilliseconds;  
    ReadingTime.Text = result.ToString();  
}
```

```
private void ButtonFind_Click(object sender, EventArgs e)  
{
```

```
    Stopwatch time = new Stopwatch();  
    time.Start();  
    int maxDistance = Int32.Parse(textBox1.Text);
```

```

foreach (string word in List)
{
    if (LevenshteinDistance(word, textBox2.Text) <= maxDistance)
    {
        WordList.Items.Add(textBox2.Text);
        break;
    }
}

time.Stop();

Double result = time.Elapsed.TotalMilliseconds;
Search.Text = result.ToString();
}

public static int LevenshteinDistance(string string1, string string2)
{
    if (string1 == null) throw new ArgumentNullException("string1");
    if (string2 == null) throw new ArgumentNullException("string2");
    int diff;
    int[,] m = new int[string1.Length + 1, string2.Length + 1];

    for (int i = 0; i <= string1.Length; i++) m[i, 0] = i;
    for (int j = 0; j <= string2.Length; j++) m[0, j] = j;

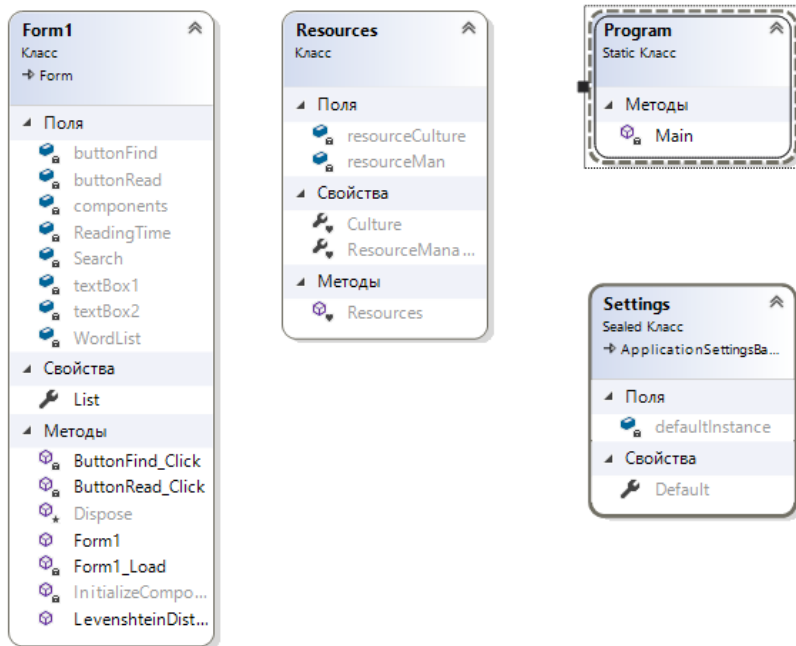
    for (int i = 1; i <= string1.Length; i++)
        for (int j = 1; j <= string2.Length; j++)
        {
            diff = (string1[i - 1] == string2[j - 1]) ? 0 : 1;

            m[i, j] = Math.Min(Math.Min(m[i - 1, j] + 1,
                                         m[i, j - 1] + 1),
                               m[i - 1, j - 1] + diff);
        }

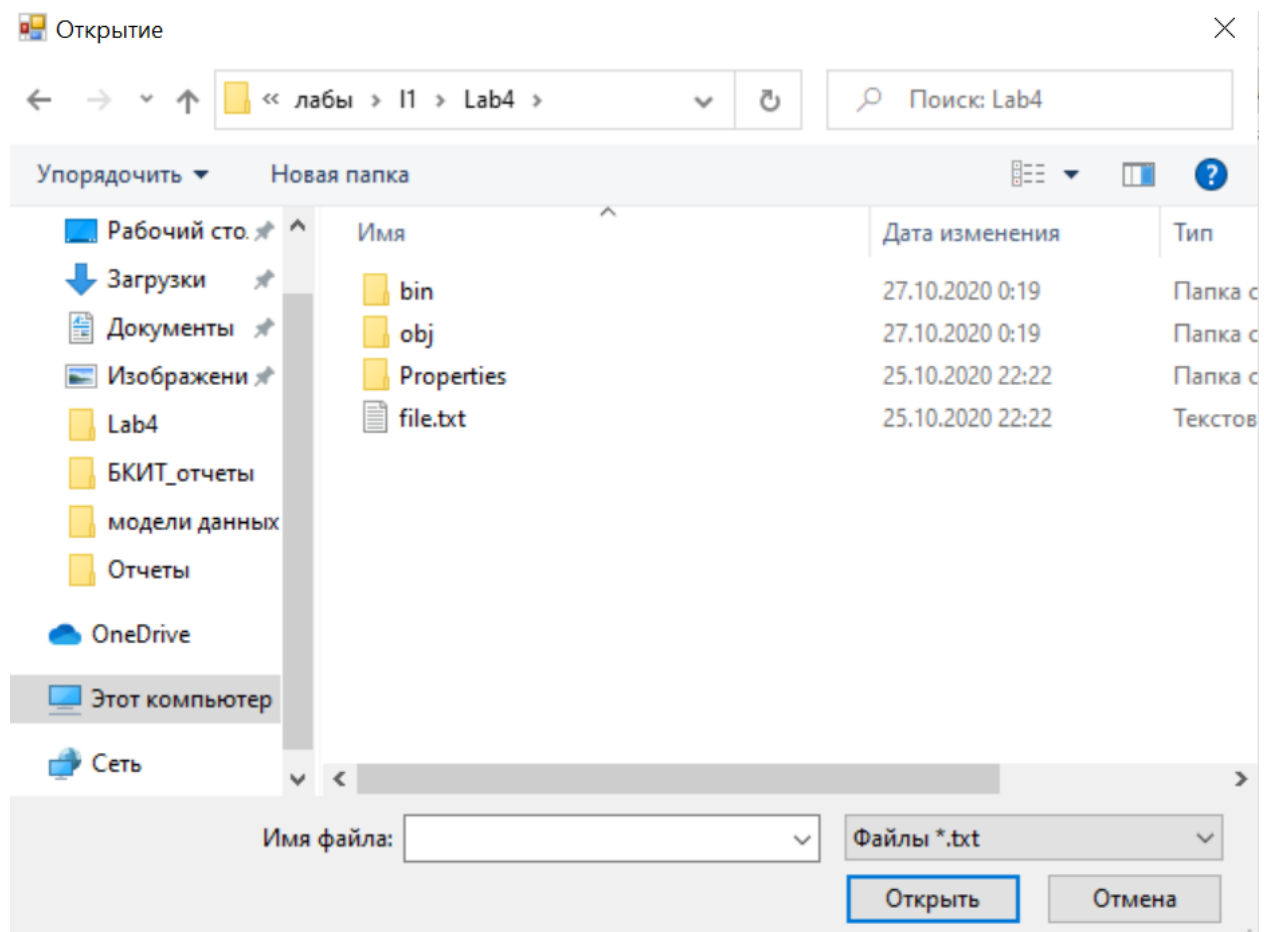
    return m[string1.Length, string2.Length];
}
}
}

```

Диаграммы классов:



Результаты программы:



Form1

Введите слово

0

ReadingTime

Search

Открыть

Найти

Form1

маленьк

2

маленьк

15,1155

60,135

Открыть

Найти