Отчет по лабораторной работе № 5

по курсу «Базовые компоненты интернет технологий»

Выполнил:

студент группы ИУ5-33
Бондаренко Алина

Описание задания лабораторной работы:

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 Дамерау-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 на языке C#.

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;
using System.IO;
using System.Diagnostics;

namespace Valera_lab5
{
    public partial class Form1 : Form
    {
        Stopwatch t = new Stopwatch();
        bool FilesBool = false;
        string Stroka;
        List<string> Files = new List<string>();
        public Form1()
        {
            InitializeComponent();
            this.Visible = false;
        }
        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void button2_Click(object sender, EventArgs e)
        {
            OpenFileDialog fd = new OpenFileDialog();
            fd.Filter = "Текстовые файлы|*.txt";
            if (fd.ShowDialog() == DialogResult.OK)
            {
                FilesBool = true;
                t.Start();

                //Чтение файла в виде строки
                string text = File.ReadAllText(fd.FileName);

                char[] separators = new char[] { ' ', '.', ',', '!', '?', '/', '\t', '\n'
};
```

```
            string[] textArray = text.Split(separators);
            foreach (string strTemp in textArray)
            {
                //Удаление пробелов в начале и конце строки
                string str = strTemp.Trim();
                //Добавление строки в список, если строка не содержится в списке
                if (!Files.Contains(str))
                {
                    Files.Add(str);
                }
            }
            t.Stop();
            this.Visible = true;
            this.Text = "FileReadTime: " + t.Elapsed.ToString();
        }
        else
        {
            MessageBox.Show("Необходимо выбрать файл");
        }
        t.Reset();
    }

    public class Distance
    {
        private string str1;
        private string str2;
        private int _Size = 0;

        public int Size
        {
            get
            { return this._Size; }
            private set
            {
                this._Size = value;
            }
        }

        public Distance(string str1, string str2)
        {
            this.str1 = str1;
            this.str2 = str2;
            Size = VichislenieDistance();
        }

        public void Zamena(string str1, string str2)
        {
            this.str1 = str1;
            this.str2 = str2;
            Size = VichislenieDistance();
        }
        private int VichislenieDistance()
        {
            if ((this.str1 == null) || (this.str2 == null))
            {
                return -1;
            }

            int str1Size = this.str1.Length;
            int str2Size = this.str2.Length;
            if ((str1Size == 0) && (str2Size == 0))
            {
                return 0;
            }
            if (str1Size == 0)
```

```
                {
                    return str2Size;
                }
                else if (str2Size == 0)
                {
                    return str1Size;
                }
                string str1 = this.str1.ToUpper();
                string str2 = this.str2.ToUpper();
                int[,] matrix = new int[str1Size + 1, str2Size + 1];
                for (int i = 0; i <= str1Size; i++)
                {
                    matrix[i, 0] = i;
                }
                for (int j = 0; j <= str2Size; j++)
                {
                    matrix[0, j] = j;
                }
                for (int i = 1; i <= str1Size; i++)//Вычисление расстояния Дамерау-
Левенштеи?на
                {
                    for (int j = 1; j <= str2Size; j++)
                    {
                        int symbEqual = ((str1.Substring(i - 1, 1) == str2.Substring(j -
1, 1)) ? 0 : 1);
                        int ins = matrix[i, j - 1] + 1;
                        int del = matrix[i - 1, j] + 1;
                        int subst = matrix[i - 1, j - 1] + symbEqual;
                        matrix[i, j] = Math.Min(Math.Min(ins, del), subst);
                        if ((i > 1) && (j > 1) && (str1.Substring(i - 1, 1) ==
str2.Substring(j - 2, 1)) && (str1.Substring(i - 2, 1) == str2.Substring(j - 1, 1)))
                        {
                            matrix[i, j] = Math.Min(matrix[i, j], matrix[i - 2, j - 2] +
symbEqual);
                        }
                    }
                }
                //Возвращается нижни? правыи? элемент матрицы
                return matrix[str1Size, str2Size];
            }
        }
        private void button1_Click(object sender, EventArgs e)
        {
            Distance A1 = new Distance(Stroka, "");
            foreach (String Stroka2 in Files)
            {
                A1.Zamena(Stroka, Stroka2);
                if (A1.Size > Convert.ToInt32(textBox2.Text))
                {
                    checkedListBox1.Items.Add(Stroka2);
                }
            }
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            Stroka = textBox1.Text;
        }

        private void checkedListBox1_SelectedIndexChanged(object sender, EventArgs e)
        {
        }

        private void label1_Click(object sender, EventArgs e)
        {
```

```
        }

        private void label2_Click(object sender, EventArgs e)
        {

        }

        private void Form1_Load_1(object sender, EventArgs e)
        {

        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {
        }
    }
}
```

Диаграмма классов: