

Лабораторная работа № 5 по курсу
“Базовые компоненты интернет-технологий”

Наврузов Э.Р.
РТ5-31
МГТУ им. Баумана

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 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.

Код программы:

Form1.cs:

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

namespace WindowsFormsApp4
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            List<string> list = new List<string>();

            private void button1_Click(object sender, EventArgs e)
            {
                OpenFileDialog FileDownload = new OpenFileDialog();
                FileDownload.Filter = "текстовые файлы |*.txt";
                if (FileDownload.ShowDialog() == DialogResult.OK)
                {
                    Stopwatch t = new Stopwatch();
                    t.Start();
                    string text = File.ReadAllText(FileDownload.FileName);
```

```

};

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

string[] textArray = text.Split(separators);
foreach (string strTemp in textArray)
{
    string str = strTemp.Trim();
    if (!list.Contains(str)) list.Add(str);
}
t.Stop();
this.textBoxFileReadTime.Text = t.Elapsed.ToString();
this.textBoxFileReadCount.Text = list.Count.ToString();
}
else
{
    MessageBox.Show("Необходимо выбрать файл");
}

}

private void button2_Click(object sender, EventArgs e)
{
    string word = this.textBoxFind.Text.Trim();

    if (!string.IsNullOrEmpty(word) && list.Count > 0)
    {
        string wordUpper = word.ToUpper();

        List<string> tempList = new List<string>();
        Stopwatch t = new Stopwatch();
        t.Start();
        foreach (string str in list)
        {
            if (str.ToUpper().Contains(wordUpper))
            {
                tempList.Add(str);
            }
        }
        t.Stop();
        this.textBoxExactTime.Text = t.Elapsed.ToString();
        this.listBoxResult.BeginUpdate();

        this.listBoxResult.Items.Clear();

        foreach (string str in tempList)
        {
            this.listBoxResult.Items.Add(str);
        }
        this.listBoxResult.EndUpdate();
    }
    else
    {
        MessageBox.Show("Необходимо выбрать файл и ввести слово для поиска");
    }
}

private void button3_Click(object sender, EventArgs e)
{
    string word = this.textBoxFind.Text.Trim();

    if (!string.IsNullOrEmpty(word) && list.Count > 0)
    {
        int maxDist;
    }
}

```

```

        if (!int.TryParse(this.textBox2.Text.Trim(), out maxDist))
        {
            {
                MessageBox.Show("Необходимо указать максимальное расстояние");
                return;
            }
            if (maxDist < 1 || maxDist > 5)
            {
                MessageBox.Show("Максимальное расстояние должно быть в диапазоне
от 1 до 5");
                return;
            }
        }

        string WordUpper = word.ToUpper();

        List<Tuple<string, int>> tempList = new List<Tuple<string, int>>();
        Stopwatch timer = new Stopwatch();

        timer.Start();

        foreach (string str in list)
        {
            int dist = EditDistance.Distance(str.ToUpper(), WordUpper);

            if (dist <= maxDist)
            {
                tempList.Add(new Tuple<string, int>(str, dist));
            }
        }

        timer.Stop();

        this.textBox1.Text = timer.Elapsed.ToString();
        this.listBoxResult.BeginUpdate();
        this.listBoxResult.Items.Clear();

        foreach (var x in tempList)
        {
            string temp = x.Item1 + " (р.Левенштайна : " + x.Item2.ToString() +
            ")";
            this.listBoxResult.Items.Add(temp);
        }
        this.listBoxResult.EndUpdate();
    }

    else
    {
        MessageBox.Show("Отсутствуют данные");
    }
}

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

private void label15_Click(object sender, EventArgs e)
{
}

}
}

```

EditDistance:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WindowsFormsApp4
{
    public static class EditDistance
    {
        public static int Distance(string str1Param, string str2Param)
        {
            if ((str1Param == null) || (str2Param == null)) return -1;

            int str1Len = str1Param.Length;
            int str2Len = str2Param.Length;

            if ((str1Len == 0) && (str2Len == 0)) return 0;
            if (str1Len == 0) return str2Len;
            if (str2Len == 0) return str1Len;

            string str1 = str1Param.ToUpper();
            string str2 = str2Param.ToUpper();

            int[,] matrix = new int[str1Len + 1, str2Len + 1];

            for (int i = 0; i <= str1Len; i++) matrix[i, 0] = i;
            for (int j = 0; j <= str2Len; j++) matrix[0, j] = j;

            for (int i = 1; i <= str1Len; i++)
            {
                for (int j = 1; j <= str2Len; 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);
                    }
                }
            }
            return matrix[str1Len, str2Len];
        }
    }
}
```

Пример работы программы:

Form1

Открыть файл

Время: 00:00:00.0001618 Слов всего: 7

Ваше слово: Anastasia

Найти точное слово Найти слово с опечатками

Время: 00:00:00.0000058 Время: 00:00:00.0006180 Макс. Расстояние: 2

Результат работы программы

```
Anastasia (р.Левенштайна : 0)
Anastdsia (р.Левенштайна : 1)
Ansstdsia (р.Левенштайна : 2)
Nastasia (р.Левенштайна : 1)
```