#### Московский государственный технический университет имени Н.Э. Баумана

## Лабораторная работа по дисциплине БКИТ №5

*Выполнил: Ханмурзин Тагир, ИУ5-34, 24.10.2017*

**1. 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 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.

**2. Листинг программного кода**

**Form1.cs**

using System;

using System.IO;

using System.Diagnostics;

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;

namespace Lab\_5

{

    public partial class Form1 : Form

    {

        Stopwatch cl = new Stopwatch();

        public Form1()

        {

            InitializeComponent();

        }

        private void button1\_Click(object sender, EventArgs e)

        {

            cl.Reset();

            OpenFileDialog win1 = new OpenFileDialog();

            win1.InitialDirectory = "\\Mac/Home/Documents/Course\_2\bkIT/Lab\_4";

            win1.Filter = "txt files (\*.txt)|\*.txt";

            win1.FilterIndex = 2;

            win1.RestoreDirectory = true;

            if (win1.ShowDialog() == DialogResult.OK)

            {

                try

                {

                    cl.Start();

                    string buf = File.ReadAllText(win1.FileName);

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

                    string[] buf2 = buf.Split();

                    foreach (string l in buf2)

                    {

                        if (!a.Contains(l))

                            a.Add(l);

                    }

                    a.Sort();

                    addToListBox(a);

                    cl.Stop();

                    label1.Text = "Чтение файла, сортировка и вывод слов: " + cl.ElapsedMilliseconds.ToString() + " миллисекунд";

                }

                catch (Exception ex)

                {

                    MessageBox.Show("Ошибка чтения файла!\n" + ex.Message);

                }

            }

        }

        void addToListBox(List<string> a)

        {

            listBox1.BeginUpdate();

            foreach (string l in a)

            {

                listBox1.Items.Add(l);

            }

            listBox1.EndUpdate();

        }

        private void findb\_Click(object sender, EventArgs e)

        {

            cl.Reset();

            cl.Start();

            listBox1.SelectedIndex = listBox1.FindStringExact(textBox1.Text);

            cl.Stop();

            label2.Text = "Поиск в списке: " + cl.ElapsedMilliseconds.ToString() + " миллисекунд";

            LevFormlist(textBox1.Text);

        }

        public static int LevDist(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];

        }

        void LevFormlist(string s)

        {

            cl.Reset();

            int p = 0;

            bool f = int.TryParse(textBox2.Text, out p);

            if (!f || p < 0)

            {

                throw new ArgumentException("Неправильные значения!");

            }

            listBox2.BeginUpdate();

            listBox2.Items.Clear();

            cl.Start();

            foreach (string l in listBox1.Items)

            {

                if (LevDist(s, l) <= p)

                    listBox2.Items.Add(l);

            }

            cl.Stop();

            listBox2.EndUpdate();

            label5.Text = " Создание списка с Расстоянием Левенштейна: " + cl.ElapsedMilliseconds + " миллисекунд";

        }

    }

}

**Form1.Designer.cs**

namespace Lab\_5

{

    partial class Form1

    {

        private System.ComponentModel.IContainer components = null;

        protected override void Dispose(bool disposing)

        {

            if (disposing && (components != null))

            {

                components.Dispose();

            }

            base.Dispose(disposing);

        }

        private void InitializeComponent()

        {

            this.loadb = new System.Windows.Forms.Button();

            this.listBox1 = new System.Windows.Forms.ListBox();

            this.label1 = new System.Windows.Forms.Label();

            this.label2 = new System.Windows.Forms.Label();

            this.findb = new System.Windows.Forms.Button();

            this.textBox1 = new System.Windows.Forms.TextBox();

            this.label4 = new System.Windows.Forms.Label();

            this.listBox2 = new System.Windows.Forms.ListBox();

            this.label5 = new System.Windows.Forms.Label();

            this.textBox2 = new System.Windows.Forms.TextBox();

            this.SuspendLayout();

            //

            // Кнопка загрузки новых данных

            //

            this.loadb.Location = new System.Drawing.Point(28, 12);

            this.loadb.Name = "loadb";

            this.loadb.Size = new System.Drawing.Size(120, 23);

            this.loadb.TabIndex = 0;

            this.loadb.Text = "Загрузить данные";

            this.loadb.UseVisualStyleBackColor = true;

            this.loadb.Click += new System.EventHandler(this.button1\_Click);

            //

            // Окно разобранных слов

            //

            this.listBox1.FormattingEnabled = true;

            this.listBox1.Location = new System.Drawing.Point(28, 48);

            this.listBox1.Name = "listBox1";

            this.listBox1.Size = new System.Drawing.Size(200, 300);

            this.listBox1.TabIndex = 1;

            //

            // label1

            //

            this.label1.AutoSize = true;

            this.label1.Location = new System.Drawing.Point(255, 48);

            this.label1.Name = "label1";

            this.label1.Size = new System.Drawing.Size(0, 13);

            this.label1.TabIndex = 2;

            //

            // label2

            //

            this.label2.AutoSize = true;

            this.label2.Location = new System.Drawing.Point(255, 63);

            this.label2.Name = "label2";

            this.label2.Size = new System.Drawing.Size(0, 13);

            this.label2.TabIndex = 3;

            //

            // Кнопка поиска текста

            //

            this.findb.Location = new System.Drawing.Point(411, 12);

            this.findb.Name = "findb";

            this.findb.Size = new System.Drawing.Size(120, 23);

            this.findb.TabIndex = 5;

            this.findb.Text = "Найти слово";

            this.findb.UseVisualStyleBackColor = true;

            this.findb.Click += new System.EventHandler(this.findb\_Click);

            //

            // Строка поиска слова

            //

            this.textBox1.Location = new System.Drawing.Point(256, 14);

            this.textBox1.Name = "textBox1";

            this.textBox1.Size = new System.Drawing.Size(140, 20);

            this.textBox1.TabIndex = 6;

            //

            // label4

            //

            this.label4.AutoSize = true;

            this.label4.Location = new System.Drawing.Point(258, 197);

            this.label4.Name = "label4";

            this.label4.Size = new System.Drawing.Size(110, 13);

            this.label4.TabIndex = 8;

            this.label4.Text = "Расстояние Левенштейна";

            //

            // listBox2

            //

            this.listBox2.FormattingEnabled = true;

            this.listBox2.Location = new System.Drawing.Point(258, 217);

            this.listBox2.Name = "listBox2";

            this.listBox2.Size = new System.Drawing.Size(200, 121);

            this.listBox2.TabIndex = 9;

            //

            // label5

            //

            this.label5.AutoSize = true;

            this.label5.Location = new System.Drawing.Point(252, 78);

            this.label5.Name = "label5";

            this.label5.Size = new System.Drawing.Size(0, 13);

            this.label5.TabIndex = 10;

            //

            // textBox2

            //

            this.textBox2.Location = new System.Drawing.Point(200, 14);

            this.textBox2.Name = "textBox2";

            this.textBox2.Size = new System.Drawing.Size(50, 20);

            this.textBox2.TabIndex = 11;

            this.textBox2.Text = "0";

            //

            // Главное окно

            //

            this.BackColor = System.Drawing.SystemColors.AppWorkspace;

            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

            this.ClientSize = new System.Drawing.Size(590, 350);

            this.Controls.Add(this.textBox2);

            this.Controls.Add(this.label5);

            this.Controls.Add(this.listBox2);

            this.Controls.Add(this.label4);

            this.Controls.Add(this.textBox1);

            this.Controls.Add(this.findb);

            this.Controls.Add(this.label2);

            this.Controls.Add(this.label1);

            this.Controls.Add(this.listBox1);

            this.Controls.Add(this.loadb);

            this.Name = "Form1";

            this.Text = "Lab\_5";

            this.ResumeLayout(false);

            this.PerformLayout();

        }

        private System.Windows.Forms.Button loadb;

        private System.Windows.Forms.ListBox listBox1;

        private System.Windows.Forms.Label label1;

        private System.Windows.Forms.Label label2;

        private System.Windows.Forms.Button findb;

        private System.Windows.Forms.TextBox textBox1;

        private System.Windows.Forms.Label label4;

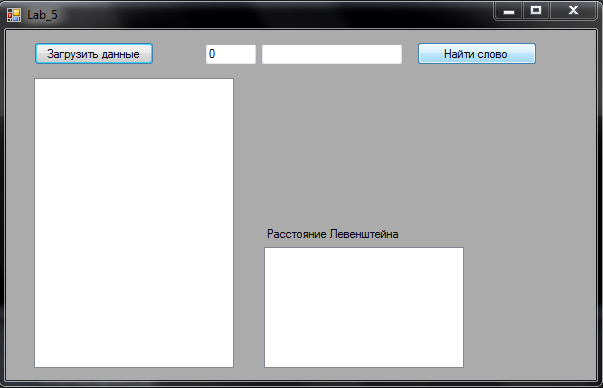
        private System.Windows.Forms.ListBox listBox2;

        private System.Windows.Forms.Label label5;

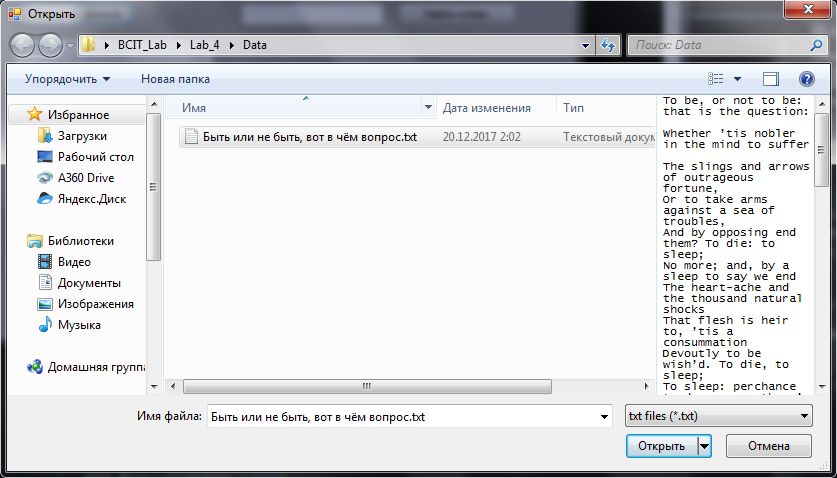
        private System.Windows.Forms.TextBox textBox2;

    }

}

**3. Тест работы**

Нажав на кнопку «Загрузить данные» выбираем нужный документ в формате «.txt».



Программа заполнит словами из документа память для дальнейших операций. Так же произведёт расчёт затраченного на операции времени и выведет на панель. Введя слово из документа, указав расстояние Левенштейна и нажав на кнопку «Найти слово» произведётся поиск данного слова в памяти, а также слов, которые находятся на данном расстоянии Левенштейна. При успешном нахождении, результат будет выделен в поле, а также в дополнительном поле с подписью «Расстояние Левенштейна» выведутся слова удовлетворяющие поиску.

