

**Отчет по лабораторной работе № 4 по курсу  
«Базовые компоненты интернет технологий»**

Исполнил:  
студент группы ИУ5-33  
Ханмагомедов Ренат

Дата: 27.12.2017

Москва МГТУ 2017

---

## Описание задания

Разработать программу, реализующую работу с файлами.

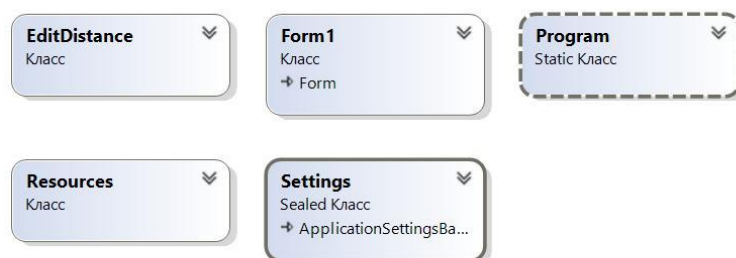
1. Программа должна быть разработана в виде приложения Windows Forms на языке C#. По желанию вместо Windows Forms возможно использование WPF.
2. Добавить кнопку, реализующую функцию чтения файла в список слов `List<string>`.
3. Для выбора имени файла используется класс `OpenFileDialog`, который открывает диалоговое окно с выбором файла. Ограничить выбор только файлами с расширением «.txt».
4. Для чтения из файла рекомендуется использовать статический метод `ReadAllText()` класса `File` (пространство имен `System.IO`). Содержимое файла считывается методом `ReadAllText()` в виде одной строки, далее делится на слова с использованием метода `Split()` класса `string`. Слова сохраняются в список `List<string>`.
5. При сохранении слов в список `List<string>` дубликаты слов не записываются. Для проверки наличия слова в списке используется метод `Contains()`.
6. Вычислить время загрузки и сохранения в список с использованием класса `Stopwatch` (пространство имен `System.Diagnostics`). Вычисленное время вывести на форму в поле ввода (`TextBox`) или надпись (`Label`).
7. Добавить на форму поле ввода для поиска слова и кнопку поиска. При нажатии на кнопку поиска осуществлять поиск введенного слова в списке. Слово считается найденным, если оно входит в элемент списка как подстрока (метод `Contains()` класса `string`).
8. Добавить на форму список (`ListBox`). Найденные слова выводить в список с использованием метода «название\_списка.Items.Add()». Вызовы метода «название\_списка.Items.Add()» должны находиться между вызовами методов «название\_списка.BeginUpdate()» и «название\_списка.EndUpdate()».
9. Вычислить время поиска с использованием класса `Stopwatch`. Вычисленное время вывести на форму в поле ввода (`TextBox`) или надпись (`Label`).

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.

## Диаграмма классов



## Текст 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.Windows.Forms;
using System.IO;
using System.Diagnostics;

namespace lab4_2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        List<string> list = new List<string>();
        private void button1_Click(object sender, EventArgs e)
        {
            OpenFileDialog fd = new OpenFileDialog();
            fd.Filter = "текстовые файлы|*.txt";

            if (fd.ShowDialog() == DialogResult.OK)
            {
                Stopwatch t = new Stopwatch();
            }
        }
    }
}
  
```

```

        t.Start();
        //Чтение файла в виде строки
        string text = File.ReadAllText(fd.FileName);
        //Разделительные символы для чтения из файла
        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.textBox1ReadTime.Text = t.Elapsed.ToString();
        this.textBox1FileReadCount.Text = list.Count.ToString();
    }
    else
    {
        MessageBox.Show("Необходимо выбрать файл");
    }
}

private void buttonExit_Click(object sender, EventArgs e)
{
    this.Close();
    //Application.Exit();
}

private void buttonSaveReport_Click(object sender, EventArgs e)
{
    //Имя файла отчета
    string TempReportFileName = "Report_"
+ DateTime.Now.ToString("dd_MM_yyyy_hhmmss");
    //Диалог сохранения файла отчета
    SaveFileDialog fd = new SaveFileDialog();
    fd.FileName = TempReportFileName;
    fd.DefaultExt = ".html";
    fd.Filter = "HTML Reports|*.html";
    if (fd.ShowDialog() == DialogResult.OK)
    {
        string ReportFileName = fd.FileName;
        //Формирование отчета
        StringBuilder b = new StringBuilder();
        b.AppendLine("<html>");
        b.AppendLine("<head>");
        b.AppendLine("<meta http-equiv='Content-Type'
content='text/html; charset=UTF-8'/>");
        b.AppendLine("<title>" + "Отчет: " + ReportFileName + "</title>");
        b.AppendLine("</head>");
        b.AppendLine("<body>");
        b.AppendLine("<h1>" + "Отчет: " + ReportFileName +
"</h1>"); b.AppendLine("<table border='1'>");
        b.AppendLine("<tr>");
        b.AppendLine("<td>Время чтения из файла</td>");
        b.AppendLine("<td>" + this.textBox1ReadTime.Text + "</td>");
        b.AppendLine("</tr>");
        b.AppendLine("<tr>");
        b.AppendLine("<td>Количество уникальных слов в файле</td>");
        b.AppendLine("<td>" + this.textBox1FileReadCount.Text + "</td>");
        b.AppendLine("</tr>");
        b.AppendLine("<tr>");
    }
}

```

```

        b.AppendLine("<td>Слово для поиска</td>");
        b.AppendLine("<td>" + this.textBoxFind.Text + "</td>");
        b.AppendLine("</tr>");
        b.AppendLine("<tr>");
        b.AppendLine("<td>Максимальное расстояние для нечеткого поиска</td>");
        b.AppendLine("<td>" + this.textBoxMaxDist.Text + "</td>");
        b.AppendLine("</tr>");
        b.AppendLine("<tr>");
        b.AppendLine("<td>Время четкого поиска</td>");
        b.AppendLine("<td>" + this.textBoxExactTime.Text + "</td>");
        b.AppendLine("</tr>");
        b.AppendLine("<tr>");
        b.AppendLine("<td>Время нечеткого поиска</td>");
        b.AppendLine("<td>" + this.textBoxApproxTime.Text + "</td>");
        b.AppendLine("</tr>");
        b.AppendLine("<tr valign='top'>");
        b.AppendLine("<td>Результаты поиска</td>");
        b.AppendLine("<td>");
        b.AppendLine("<ul>");
        foreach (var x in this.listBoxResult.Items)
        {
            b.AppendLine("<li>" + x.ToString() + "</li>");
        }
        b.AppendLine("</ul>");
        b.AppendLine("</td>");
        b.AppendLine("</tr>");
        b.AppendLine("</table>");
        b.AppendLine("</body>");
        b.AppendLine("</html>");
        //Сохранение файла
        File.AppendAllText(ReportFileName, b.ToString());
        MessageBox.Show("Отчет сформирован. Файл: " + ReportFileName);
    }
}

private void buttonExact_Click(object sender, EventArgs e)
{
    //Слово для поиска
    string word = this.textBoxFind.Text.Trim();
    //Если слово для поиска не пусто
    if (!string.IsNullOrEmpty(word) && list.Count > 0)
    {
        //Слово для поиска в верхнем регистре
        string wordUpper = word.ToUpper();
        //Временные результаты поиска
        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();
        //Очистка списка
        this.listBoxResult.Items.Clear();
        //Вывод результатов поиска
        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 buttonApprox_Click(object sender, EventArgs e)
{
    //Слово для поиска
    string word = this.textBoxFind.Text.Trim();
    //Если слово для поиска не пусто
    if (!string.IsNullOrEmpty(word) && list.Count > 0)
    {
        int maxDist;
        if (!int.TryParse(this.textBoxMaxDist.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;
        }
        //Слово для поиска в верхнем регистре
        string wordUpper = word.ToUpper();
        //Временные результаты поиска
        List<Tuple<string, int>> tempList = new List<Tuple<string, int>>();
        Stopwatch t = new Stopwatch();
        t.Start();
        foreach (string str in list)
        {
            //Вычисление расстояния Дамерау-Левенштейна
            int dist = EditDistance.Distance(str.ToUpper(), wordUpper);
            //Если расстояние меньше порогового, то слово добавляется в результат
            if (dist <= maxDist)
            {
                tempList.Add(new Tuple<string, int>(str, dist));
            }
        }
        t.Stop();
        this.textBoxApproxTime.Text = t.Elapsed.ToString();
        this.listBoxResult.BeginUpdate();
        //Очистка списка
        this.listBoxResult.Items.Clear();
        //Вывод результатов поиска
        foreach (var x in tempList)
        {
            string temp = x.Item1 + "(расстояние=" + x.Item2.ToString() + ")";
            this.listBoxResult.Items.Add(temp);
        }
        this.listBoxResult.EndUpdate();
    }
    else
    {
        MessageBox.Show("Необходимо выбрать файл и ввести слово для поиска");
    }
}
}
}
}

```

## Form1.Designer.cs

```
namespace lab4_2
{
    partial class Form1
    {
        /// <summary>
        /// Обязательная переменная конструктора.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        /// <param name="disposing">истинно, если управляемый ресурс должен быть
        удален; иначе ложно.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Код, автоматически созданный конструктором форм Windows

        /// <summary>
        /// Требуемый метод для поддержки конструктора – не изменяйте
        /// содержимое этого метода с помощью редактора кода.
        /// </summary>
        private void InitializeComponent()
        {
            this.button1 = new System.Windows.Forms.Button();
            this.textBox1ReadTime = new System.Windows.Forms.TextBox();
            this.textBox1FileReadCount = new System.Windows.Forms.TextBox();
            this.buttonExact = new System.Windows.Forms.Button();
            this.textBoxFind = new System.Windows.Forms.TextBox();
            this.buttonApprox = new System.Windows.Forms.Button();
            this.textBoxMaxDist = new System.Windows.Forms.TextBox();
            this.listBoxResult = new System.Windows.Forms.ListBox();
            this.textBoxApproxTime = new System.Windows.Forms.TextBox();
            this.buttonExit = new System.Windows.Forms.Button();
            this.buttonSaveReport = new System.Windows.Forms.Button();
            this.label1 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.label4 = new System.Windows.Forms.Label();
            this.textBoxExactTime = new System.Windows.Forms.TextBox();
            this.label5 = new System.Windows.Forms.Label();
            this.label6 = new System.Windows.Forms.Label();
            this.SuspendLayout();
            //
            // button1
            //
            this.button1.BackColor =
                System.Drawing.Color.FromArgb(((int)((byte)(255))),
                ((int)((byte)(224))), ((int)((byte)(192))));
            this.button1.Location = new System.Drawing.Point(24,
                36); this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(137,
                68); this.button1.TabIndex = 0;
            this.button1.Text = "Чтение из файла";
        }
    }
}
```

```

        this.button1.UseVisualStyleBackColor = false;
        this.button1.Click += new System.EventHandler(this.button1_Click);
        //
        // textBox1ReadTime
        this.textBox1ReadTime.Location = new System.Drawing.Point(529,
        29); this.textBox1ReadTime.Name = "textBox1ReadTime";
        this.textBox1ReadTime.Size = new System.Drawing.Size(135,
        26); this.textBox1ReadTime.TabIndex = 1;
        //
        // textBox1FileReadCount
        //
        this.textBox1FileReadCount.Location = new System.Drawing.Point(529,
        71); this.textBox1FileReadCount.Name = "textBox1FileReadCount";
        this.textBox1FileReadCount.Size = new System.Drawing.Size(135, 26);
        this.textBox1FileReadCount.TabIndex = 2; //
        // buttonExact
        //
        this.buttonExact.BackColor =
        System.Drawing.Color.FromArgb(((int)(((byte)(255)))),
        ((int)(((byte)(224)))), ((int)(((byte)(192)))));
        this.buttonExact.Location = new System.Drawing.Point(24, 181);
        this.buttonExact.Name = "buttonExact";
        this.buttonExact.Size = new System.Drawing.Size(137, 58);
        this.buttonExact.TabIndex = 3;
        this.buttonExact.Text = "Четкий поиск";
        this.buttonExact.UseVisualStyleBackColor = false;
        this.buttonExact.Click += new System.EventHandler(this.buttonExact_Click);
        //
        // textBoxFind
        this.textBoxFind.Location = new System.Drawing.Point(250,
        141); this.textBoxFind.Name = "textBoxFind";
        this.textBoxFind.Size = new System.Drawing.Size(193,
        26); this.textBoxFind.TabIndex = 4;
        //
        // buttonApprox
        //
        this.buttonApprox.BackColor =
        System.Drawing.Color.FromArgb(((int)(((byte)(255)))), ((int)(((byte)(224)))),
        ((int)(((byte)(192)))));
        this.buttonApprox.Location = new System.Drawing.Point(24,
        260); this.buttonApprox.Name = "buttonApprox";
        this.buttonApprox.Size = new System.Drawing.Size(137, 74);
        this.buttonApprox.TabIndex = 5;
        this.buttonApprox.Text = "Параллельный нечеткий поиск:";
        this.buttonApprox.UseVisualStyleBackColor = false;
        this.buttonApprox.Click += new System.EventHandler(this.buttonApprox_Click);
        //
        // textBoxMaxDist
        this.textBoxMaxDist.Location = new System.Drawing.Point(601,
        216); this.textBoxMaxDist.Name = "textBoxMaxDist";
        this.textBoxMaxDist.Size = new System.Drawing.Size(135,
        26); this.textBoxMaxDist.TabIndex = 6;
        //
        // listBoxResult
        //
        this.listBoxResult.FormattingEnabled = true;
        this.listBoxResult.ItemHeight = 20;
        this.listBoxResult.Location = new System.Drawing.Point(24, 353);
        this.listBoxResult.Name = "listBoxResult";
        this.listBoxResult.Size = new System.Drawing.Size(712, 184);
        this.listBoxResult.TabIndex = 8;

```



```

//
// textBoxApproxTime
//
this.textBoxApproxTime.Location = new System.Drawing.Point(601,
260); this.textBoxApproxTime.Name = "textBoxApproxTime";
this.textBoxApproxTime.Size = new System.Drawing.Size(135,
26); this.textBoxApproxTime.TabIndex = 9;
//
// buttonExit
//
this.buttonExit.BackColor = System.Drawing.Color.Red;
this.buttonExit.Location = new System.Drawing.Point(582, 563);
this.buttonExit.Name = "buttonExit";
this.buttonExit.Size = new System.Drawing.Size(154, 35);
this.buttonExit.TabIndex = 10;
this.buttonExit.Text = "Выход";
this.buttonExit.UseVisualStyleBackColor = false;
this.buttonExit.Click += new System.EventHandler(this.buttonExit_Click);
//
// buttonSaveReport
//
this.buttonSaveReport.BackColor =
System.Drawing.Color.FromArgb(((int)(((byte)(255)))),
((int)(((byte)(128)))), ((int)(((byte)(128))))));
this.buttonSaveReport.Location = new System.Drawing.Point(24,
563); this.buttonSaveReport.Name = "buttonSaveReport";
this.buttonSaveReport.Size = new System.Drawing.Size(154, 32);
this.buttonSaveReport.TabIndex = 11; this.buttonSaveReport.Text =
"Отчет"; this.buttonSaveReport.UseVisualStyleBackColor = false;
this.buttonSaveReport.Click += new
System.EventHandler(this.buttonSaveReport_Click);
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(188,
36); this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(197,
20); this.label1.TabIndex = 12;
this.label1.Text = "Время чтения из файла:";
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(188,
77); this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(305,
20); this.label2.TabIndex = 13;
this.label2.Text = "Количество уникальных слов в
тексте:"; //
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(20,
141); this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(151,
20); this.label3.TabIndex = 14;
this.label3.Text = "Слово для поиска:";
//
// label4
//
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(396,
181); this.label4.Name = "label4";

```

```

        this.label4.Size = new System.Drawing.Size(182,
        20); this.label4.TabIndex = 15;
        this.label4.Text = "Время четкого поиска:";
        //
        // textBoxExactTime

        this.textBoxExactTime.Location = new System.Drawing.Point(601,
        175); this.textBoxExactTime.Name = "textBoxExactTime";
        this.textBoxExactTime.Size = new System.Drawing.Size(135,
        26); this.textBoxExactTime.TabIndex = 16;

        //
        // label5
        //
        this.label5.AutoSize = true;
        this.label5.Location = new System.Drawing.Point(188,
        219); this.label5.Name = "label5";
        this.label5.Size = new System.Drawing.Size(390,
        20); this.label5.TabIndex = 17;
        this.label5.Text = "Максимальное расстояние для нечеткого
        поиска:"; //
        // label6

        this.label6.AutoSize = true;
        this.label6.Location = new System.Drawing.Point(365,
        260); this.label6.Name = "label6";
        this.label6.Size = new System.Drawing.Size(200,
        20); this.label6.TabIndex = 18;
        this.label6.Text = "Время нечеткого поиска:";

        //
        // Form1
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(9F, 20F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))),
        ((int)(((byte)(192)))), ((int)(((byte)(192)))));
        this.ClientSize = new System.Drawing.Size(780, 607);
        this.Controls.Add(this.label6);
        this.Controls.Add(this.label5);
        this.Controls.Add(this.textBoxExactTime);
        this.Controls.Add(this.label4);
        this.Controls.Add(this.label3);
        this.Controls.Add(this.label2);
        this.Controls.Add(this.label1);
        this.Controls.Add(this.buttonSaveReport);
        this.Controls.Add(this.buttonExit);
        this.Controls.Add(this.textBoxApproxTime);
        this.Controls.Add(this.listBoxResult);
        this.Controls.Add(this.textBoxMaxDist);
        this.Controls.Add(this.buttonApprox);
        this.Controls.Add(this.textBoxFind);
        this.Controls.Add(this.buttonExact);
        this.Controls.Add(this.textBox1FileReadCount);
        this.Controls.Add(this.textBox1ReadTime);
        this.Controls.Add(this.button1);
        this.Name = "Form1";
        this.Text = "Form1";
        this.ResumeLayout(false);
        this.PerformLayout();

    }

#endregion

private System.Windows.Forms.Button button1;
private System.Windows.Forms.TextBox textBox1ReadTime;

```

```

private System.Windows.Forms.TextBox textBox1FileReadCount;
private System.Windows.Forms.Button buttonExact;
private System.Windows.Forms.TextBox textBoxFind;
private System.Windows.Forms.Button buttonApprox;
private System.Windows.Forms.TextBox textBoxMaxDist;
private System.Windows.Forms.ListBox listBoxResult;
private System.Windows.Forms.TextBox
textBoxApproxTime; private System.Windows.Forms.Button
buttonExit; private System.Windows.Forms.Button
buttonSaveReport; private System.Windows.Forms.Label
label1; private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.TextBox
textBoxExactTime; private System.Windows.Forms.Label
label5; private System.Windows.Forms.Label label6;
    }
}

```

## EditDistance.cs

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;

namespace lab4_2
{
    class EditDistance
    {
        /// <summary>
        /// Вычисление расстояния Дамерау-Левенштейна
        /// </summary>
        public static int Distance(string str1Param, string str2Param)
        {
            if ((str1Param == null) || (str2Param == null)) return -1;
            int str1Len = str1Param.Length; int str2Len =
            str2Param.Length;
            //Если хотя бы одна строка пустая, возвращается длина другой
            строки if ((str1Len == 0) && (str2Len == 0)) return 0; if (str1Len
            == 0) return str2Len;
            if (str2Len == 0) return str1Len;
            //Приведение строк к верхнему регистру
            string str1 = str1Param.ToUpper();
            string str2 = str2Param.ToUpper();
            //Объявление матрицы
            int[,] matrix = new int[str1Len + 1, str2Len + 1];
            //Инициализация нулевой строки и нулевого столбца матрицы
            for (int i = 0; i <= str1Len; i++) matrix[i, 0] = i;
            for (int j = 0; j <= str2Len; j++) matrix[0, j] =
            j; //Вычисление расстояния Дамерау-Левенштейна
            for (int i = 1; i <= str1Len; i++)
            {
                for (int j = 1; j <= str2Len; j++)
                {
                    //Эквивалентность символов, переменная symbEqual соответствует
                    m(s1[i],s2[j])
                    int symbEqual = ((str1.Substring(i - 1, 1) == str2.Substring(j - 1,
                    1)) ? 0 : 1);

                    int ins = matrix[i, j - 1] + 1; //Добавление
                    int del = matrix[i - 1, j] + 1; //Удаление
                    int subst = matrix[i - 1, j - 1] + symbEqual; //Замена
                                                                    //Элемент матрицы
                    //вычисляется как минимальный из трех случаев
                    matrix[i, j] = Math.Min(Math.Min(ins, del), subst);
                }
            }
        }
    }
}

```

```

        //Дополнение Дамерау по перестановке соседних
        символов 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 нижний правый элемент
матрицы return matrix[str1Len, str2Len];
}
}
}
}

```

## Form1.cs[Конструктор]

The screenshot shows a Windows application window titled "Form1". The interface is designed for a string search algorithm. It features several input fields and buttons:

- Чтение из файла** (Read from file): A button in the top left.
- Время чтения из файла:** (File reading time): An input field.
- Количество уникальных слов в тексте:** (Number of unique words in the text): An input field.
- Слово для поиска:** (Search word): An input field.
- Четкий поиск** (Exact search): A button.
- Время четкого поиска:** (Exact search time): An input field.
- Максимальное расстояние для нечеткого поиска:** (Maximum distance for fuzzy search): An input field.
- Параллельный нечеткий поиск:** (Parallel fuzzy search): A button.
- Время нечеткого поиска:** (Fuzzy search time): An input field.
- listBoxResult**: A large text area for displaying search results.
- Отчет** (Report): A button at the bottom left.
- Выход** (Exit): A button at the bottom right.

Экранные формы с примерами выполнения программы

Первоначальная форма:

The screenshot shows a Windows-style window titled 'Form1'. The interface has a light pink background. On the left, there are three yellow buttons with black text: 'Чтение из файла' (top), 'Четкий поиск' (middle), and 'Параллельный нечеткий поиск' (bottom). The 'Четкий поиск' button is currently selected, indicated by a blue border. To the right of these buttons are several input fields: 'Время чтения из файла:' and 'Количество уникальных слов в тексте:' are at the top right; 'Слово для поиска:' is in the middle left; 'Время четкого поиска:', 'Максимальное расстояние для нечеткого поиска:', and 'Время нечеткого поиска:' are on the right side. At the bottom, there are two red buttons: 'Отчет' (left) and 'Выход' (right). A large white rectangular area for results is located in the center-bottom of the window.

Результаты четкого поиска в случае существования слова:

This screenshot shows the same 'Form1' window after a search. The 'Четкий поиск' button remains selected. The input fields now contain data: 'Время чтения из файла:' shows '00:00:00.0003537', 'Количество уникальных слов в тексте:' shows '75', and 'Слово для поиска:' shows 'разработка'. The search time fields are also populated: 'Время четкого поиска:' shows '00:00:00.0000298', while the other two are empty. The large white result area at the bottom now contains the word 'разработка' at the top. The 'Отчет' and 'Выход' buttons are still present at the bottom.

Результаты нечеткого поиска:

Чтение из файла

Время чтения из файла:00:00:00.0003537

Количество уникальных слов в тексте:75

Слово для поиска:разработка

Четкий поиск

Время четкого поиска:00:00:00.0000298

Максимальное расстояние для нечеткого поиска:2

Параллельный нечеткий поиск:

Время нечеткого поиска:00:00:00.0012939

разработка(расстояние=0)  
разработки(расстояние=1)

Отчет

Выход

Отчет по поиску заданного слова:

Отчет: C:\Users\User\Desktop\отчет.html

Время чтения из файла	00:00:00.0003537
Количество уникальных слов в файле	75
Слово для поиска	разработка
Максимальное расстояние для нечеткого поиска	2
Время четкого поиска	00:00:00.0000298
Время нечеткого поиска	00:00:00.0012939
Результаты поиска	<ul style="list-style-type: none"><li>разработка(расстояние=0)</li><li>разработки(расстояние=1)</li></ul>



