**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Радиотехнический»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет технологий»

Пояснительная записка по домашнему заданию

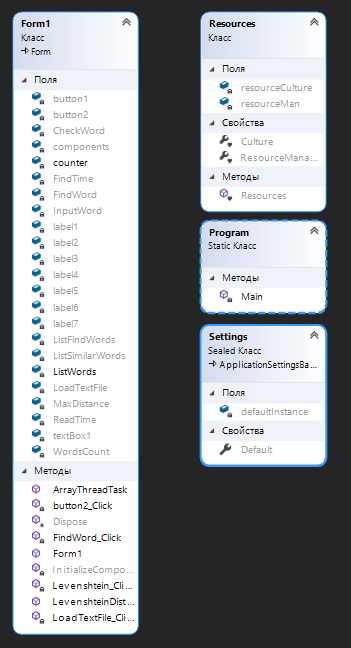
|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы РТ5-31Б |  | преподаватель каф. ИУ5 |
| Незаметдинов Тимур |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

# Описание задания

Разработать программу, реализующую многопоточный поиск в файле.

1. Программа должна быть разработана в виде приложения Windows Forms на языке C#. По желанию вместо Windows Forms возможно использование WPF.
2. В качестве основы используется макет, разработанный в лабораторных работах №4 и №5.
3. Реализуйте функцию поиска с использованием расстояния Левенштейна в многопоточном варианте. Количество потоков для запуска функции поиска вводится на форме в поле ввода (TextBox). В качестве примера используйте проект «Parallel» из примера «Введение в C#».
4. Реализуйте функцию записи результатов поиска в файл отчета. Файл отчета создается в формате .txt или .html. В качестве примера используйте проект «WindowsFormsFiles» (обработчик события кнопки «Сохранение отчета») из примера «Введение в C#».

**Диаграмма классов**



**Текст программы**

namespace WindowsFormsApplication1  
{  
    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.LoadTextFile = new System.Windows.Forms.Button();  
            this.ReadTime = new System.Windows.Forms.TextBox();  
            this.WordsCount = new System.Windows.Forms.TextBox();  
            this.label1 = new System.Windows.Forms.Label();  
            this.label2 = new System.Windows.Forms.Label();  
            this.InputWord = new System.Windows.Forms.TextBox();  
            this.FindWord = new System.Windows.Forms.Button();  
            this.label4 = new System.Windows.Forms.Label();  
            this.FindTime = new System.Windows.Forms.TextBox();  
            this.ListFindWords = new System.Windows.Forms.ListBox();  
            this.label5 = new System.Windows.Forms.Label();  
            this.label7 = new System.Windows.Forms.Label();  
            this.MaxDistance = new System.Windows.Forms.TextBox();  
            this.CheckWord = new System.Windows.Forms.TextBox();  
            this.button1 = new System.Windows.Forms.Button();  
            this.label3 = new System.Windows.Forms.Label();  
            this.label6 = new System.Windows.Forms.Label();  
            this.ListSimilarWords = new System.Windows.Forms.ListBox();  
            this.textBox1 = new System.Windows.Forms.TextBox();  
            this.button2 = new System.Windows.Forms.Button();  
            this.SuspendLayout();  
            //   
            // LoadTextFile  
            //   
            this.LoadTextFile.Location = new System.Drawing.Point(12, 12);  
            this.LoadTextFile.Name = "LoadTextFile";  
            this.LoadTextFile.Size = new System.Drawing.Size(75, 61);  
            this.LoadTextFile.TabIndex = 0;  
            this.LoadTextFile.Text = "Чтение из файла";  
            this.LoadTextFile.UseVisualStyleBackColor = true;  
            this.LoadTextFile.Click += new System.EventHandler(this.LoadTextFile\_Click);  
            //   
            // ReadTime  
            //   
            this.ReadTime.Location = new System.Drawing.Point(244, 15);  
            this.ReadTime.Name = "ReadTime";  
            this.ReadTime.Size = new System.Drawing.Size(123, 20);  
            this.ReadTime.TabIndex = 1;  
            //   
            // WordsCount  
            //   
            this.WordsCount.Location = new System.Drawing.Point(244, 56);  
            this.WordsCount.Name = "WordsCount";  
            this.WordsCount.Size = new System.Drawing.Size(123, 20);  
            this.WordsCount.TabIndex = 2;  
            //   
            // label1  
            //   
            this.label1.AutoSize = true;  
            this.label1.Location = new System.Drawing.Point(128, 15);  
            this.label1.Name = "label1";  
            this.label1.Size = new System.Drawing.Size(80, 13);  
            this.label1.TabIndex = 3;  
            this.label1.Text = "Время чтения:";  
            //   
            // label2  
            //   
            this.label2.AutoSize = true;  
            this.label2.Location = new System.Drawing.Point(128, 56);  
            this.label2.Name = "label2";  
            this.label2.Size = new System.Drawing.Size(96, 13);  
            this.label2.TabIndex = 4;  
            this.label2.Text = "Количество слов:";  
            //   
            // InputWord  
            //   
            this.InputWord.Location = new System.Drawing.Point(145, 133);  
            this.InputWord.Name = "InputWord";  
            this.InputWord.Size = new System.Drawing.Size(222, 20);  
            this.InputWord.TabIndex = 5;  
            //   
            // FindWord  
            //   
            this.FindWord.Location = new System.Drawing.Point(405, 133);  
            this.FindWord.Name = "FindWord";  
            this.FindWord.Size = new System.Drawing.Size(287, 61);  
            this.FindWord.TabIndex = 6;  
            this.FindWord.Text = "Найти слово";  
            this.FindWord.UseVisualStyleBackColor = true;  
            this.FindWord.Click += new System.EventHandler(this.FindWord\_Click);  
            //   
            // label4  
            //   
            this.label4.AutoSize = true;  
            this.label4.Location = new System.Drawing.Point(12, 177);  
            this.label4.Name = "label4";  
            this.label4.Size = new System.Drawing.Size(82, 13);  
            this.label4.TabIndex = 9;  
            this.label4.Text = "Время поиска:";  
            //   
            // FindTime  
            //   
            this.FindTime.Location = new System.Drawing.Point(145, 174);  
            this.FindTime.Name = "FindTime";  
            this.FindTime.Size = new System.Drawing.Size(222, 20);  
            this.FindTime.TabIndex = 8;  
            //   
            // ListFindWords  
            //   
            this.ListFindWords.FormattingEnabled = true;  
            this.ListFindWords.Location = new System.Drawing.Point(12, 231);  
            this.ListFindWords.Name = "ListFindWords";  
            this.ListFindWords.Size = new System.Drawing.Size(741, 173);  
            this.ListFindWords.TabIndex = 10;  
            //   
            // label5  
            //   
            this.label5.AutoSize = true;  
            this.label5.Location = new System.Drawing.Point(14, 466);  
            this.label5.Name = "label5";  
            this.label5.Size = new System.Drawing.Size(99, 13);  
            this.label5.TabIndex = 15;  
            this.label5.Text = "Макс расстояние:";  
            //   
            // label7  
            //   
            this.label7.AutoSize = true;  
            this.label7.Location = new System.Drawing.Point(14, 500);  
            this.label7.Name = "label7";  
            this.label7.Size = new System.Drawing.Size(113, 13);  
            this.label7.TabIndex = 15;  
            this.label7.Text = "Количество потоков:";  
            //   
            // MaxDistance  
            //   
            this.MaxDistance.Location = new System.Drawing.Point(145, 463);  
            this.MaxDistance.Name = "MaxDistance";  
            this.MaxDistance.Size = new System.Drawing.Size(234, 20);  
            this.MaxDistance.TabIndex = 13;  
            this.MaxDistance.Text = "0";  
            //   
            // CheckWord  
            //   
            this.CheckWord.Location = new System.Drawing.Point(145, 425);  
            this.CheckWord.Name = "CheckWord";  
            this.CheckWord.Size = new System.Drawing.Size(234, 20);  
            this.CheckWord.TabIndex = 12;  
            //   
            // button1  
            //   
            this.button1.Location = new System.Drawing.Point(405, 428);  
            this.button1.Name = "button1";  
            this.button1.Size = new System.Drawing.Size(287, 61);  
            this.button1.TabIndex = 11;  
            this.button1.Text = "Поиск похожих слов";  
            this.button1.UseVisualStyleBackColor = true;  
            this.button1.Click += new System.EventHandler(this.Levenshtein\_Click);  
            //   
            // label3  
            //   
            this.label3.AutoSize = true;  
            this.label3.Location = new System.Drawing.Point(12, 136);  
            this.label3.Name = "label3";  
            this.label3.Size = new System.Drawing.Size(101, 13);  
            this.label3.TabIndex = 7;  
            this.label3.Text = "Слово для поиска:";  
            //   
            // label6  
            //   
            this.label6.AutoSize = true;  
            this.label6.Location = new System.Drawing.Point(12, 428);  
            this.label6.Name = "label6";  
            this.label6.Size = new System.Drawing.Size(101, 13);  
            this.label6.TabIndex = 16;  
            this.label6.Text = "Слово для поиска:";  
            //   
            // ListSimilarWords  
            //   
            this.ListSimilarWords.FormattingEnabled = true;  
            this.ListSimilarWords.Location = new System.Drawing.Point(12, 527);  
            this.ListSimilarWords.Name = "ListSimilarWords";  
            this.ListSimilarWords.Size = new System.Drawing.Size(741, 173);  
            this.ListSimilarWords.TabIndex = 17;  
            //   
            // textBox1  
            //   
            this.textBox1.Location = new System.Drawing.Point(145, 492);  
            this.textBox1.Name = "textBox1";  
            this.textBox1.Size = new System.Drawing.Size(234, 20);  
            this.textBox1.TabIndex = 18;  
            this.textBox1.Text = "1";  
            //   
            // button2  
            //   
            this.button2.Location = new System.Drawing.Point(12, 708);  
            this.button2.Name = "button2";  
            this.button2.Size = new System.Drawing.Size(128, 26);  
            this.button2.TabIndex = 19;  
            this.button2.Text = "Сохранить отчет";  
            this.button2.UseVisualStyleBackColor = true;  
            this.button2.Click += new System.EventHandler(this.button2\_Click);  
            //   
            // Form1  
            //   
            this.ClientSize = new System.Drawing.Size(765, 746);  
            this.Controls.Add(this.button2);  
            this.Controls.Add(this.textBox1);  
            this.Controls.Add(this.ListSimilarWords);  
            this.Controls.Add(this.label7);  
            this.Controls.Add(this.label6);  
            this.Controls.Add(this.label5);  
            this.Controls.Add(this.MaxDistance);  
            this.Controls.Add(this.CheckWord);  
            this.Controls.Add(this.button1);  
            this.Controls.Add(this.ListFindWords);  
            this.Controls.Add(this.label4);  
            this.Controls.Add(this.FindTime);  
            this.Controls.Add(this.label3);  
            this.Controls.Add(this.FindWord);  
            this.Controls.Add(this.InputWord);  
            this.Controls.Add(this.label2);  
            this.Controls.Add(this.label1);  
            this.Controls.Add(this.WordsCount);  
            this.Controls.Add(this.ReadTime);  
            this.Controls.Add(this.LoadTextFile);  
            this.Name = "Form1";  
            this.ResumeLayout(false);  
            this.PerformLayout();  
  
        }  
  
  
        #endregion  
  
        private System.Windows.Forms.Button LoadTextFile;  
        private System.Windows.Forms.TextBox ReadTime;  
        private System.Windows.Forms.TextBox WordsCount;  
        private System.Windows.Forms.Label label1;  
        private System.Windows.Forms.Label label2;  
        private System.Windows.Forms.TextBox InputWord;  
        private System.Windows.Forms.Button FindWord;  
        private System.Windows.Forms.Label label4;  
        private System.Windows.Forms.TextBox FindTime;  
        private System.Windows.Forms.ListBox ListFindWords;  
        private System.Windows.Forms.Label label5;  
        private System.Windows.Forms.TextBox MaxDistance;  
        private System.Windows.Forms.TextBox CheckWord;  
        private System.Windows.Forms.Button button1;  
        private System.Windows.Forms.Label label3;  
        private System.Windows.Forms.Label label6;  
        private System.Windows.Forms.Label label7;  
        private System.Windows.Forms.ListBox ListSimilarWords;  
        private System.Windows.Forms.TextBox textBox1;  
        private System.Windows.Forms.Button button2;  
    }  
}

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.Diagnostics;  
using System.IO;  
  
namespace WindowsFormsApplication1  
{  
    public partial class Form1 : Form  
    {  
        public Form1()  
        {  
            InitializeComponent();  
        }  
  
        private List<string> ListWords = new List<string>();  
        private int counter = 0;  
        //Загрузка текстового файла  
        private void LoadTextFile\_Click(object sender, EventArgs e)  
        {  
            ListWords.Clear();  
            OpenFileDialog dialog = new OpenFileDialog();  
            dialog.Filter = "Текстовый файл (.txt)|\*.txt";  
            if (dialog.ShowDialog() == DialogResult.OK)  
            {  
                char[] separators = new char[] { ' ', ',', '.', '?', '!', '/', '|', '"', '\n', '\t', '\_', '-', '(', ')', '\*', '{', '}', '[', ']' };  
                Stopwatch timer = new Stopwatch();  
                timer.Start();  
                string filetext = File.ReadAllText(dialog.FileName);  
                string[] ListWordsBuf = filetext.Split(separators);  
                foreach (string word in ListWordsBuf)  
                {  
                    if (!ListWords.Contains(word)) ListWords.Add(word);  
                }  
  
                this.ReadTime.Text = timer.Elapsed.ToString();  
                this.WordsCount.Text = ListWords.Count.ToString();  
            }  
            else  
            {  
                MessageBox.Show("Ошибка! Файл не выбран!");  
            }  
            counter = ListWords.Count;  
        }  
  
        //Поиск одинаковых слов  
        private void FindWord\_Click(object sender, EventArgs e)  
        {  
            string findword = this.InputWord.Text.Trim();  
            if (!string.IsNullOrWhiteSpace(findword) && ListWords.Count > 0)  
            {  
                string findwordup = findword.ToUpper();  
                List<string> FindWords = new List<string>();  
                Stopwatch timer = new Stopwatch();  
                timer.Start();  
                foreach (string word in ListWords)  
                {  
                    if (word.ToUpper().Contains(findwordup))  
                    {  
                        FindWords.Add(word);  
                    }  
                }  
                timer.Stop();  
                this.FindTime.Text = timer.Elapsed.ToString();  
                this.ListFindWords.BeginUpdate();  
                this.ListFindWords.Items.Clear();  
                foreach (string word in FindWords)  
                {  
                    this.ListFindWords.Items.Add(word);  
                }  
                this.ListFindWords.EndUpdate();  
            }  
        }  
  
        //Нахождения расстояния Левенштейна  
        public static int LevenshteinDistance(string str1, string str2)  
        {  
            //Проверка на исключительные случаи  
  
            if ((str1 == null && str2 == null) || (str1 == str2)) return 0;  
            if (str1 == null || str2 == null) throw new ArgumentNullException("Одна из строк пустая!\n");  
  
            //Алгоритм Вагнера -- Фишера  
  
            int[,] matrix = new int[str1.Length + 1, str2.Length + 1];  
  
            for (int i = 0; i <= str1.Length; i++)  
            {  
                matrix[i, 0] = i;  
            }  
            for (int j = 0; j <= str2.Length; j++)  
            {  
                matrix[0, j] = j;  
            }  
  
            for (int i = 1; i <= str1.Length; i++)  
            {  
                for (int j = 1; j <= str2.Length; j++)  
                {  
                    int d = 1;  
                    if (str1[i - 1] == str2[j - 1]) d = 0;  
                    matrix[i, j] = Math.Min(Math.Min(matrix[i - 1, j] + 1, matrix[i, j - 1] + 1), matrix[i - 1, j - 1] + d);  
  
                }  
            }  
            return matrix[str1.Length, str2.Length];  
        }  
  
        //Поиск слов по расстоянию Левенштейна  
        private void Levenshtein\_Click(object sender, EventArgs e)  
        {  
            string checkword = this.CheckWord.Text.Trim();  
            int maxdistance = Convert.ToInt32(this.MaxDistance.Text.Trim());  
            int threadcount = Convert.ToInt32(this.textBox1.Text.Trim());  
            int wordthread = counter / threadcount;  
            int ifmore = counter % threadcount;  
            Task<List<string>>[] tasks = new Task<List<string>>[threadcount];  
  
            int min = 0;  
            this.ListSimilarWords.BeginUpdate();  
            this.ListSimilarWords.Items.Clear();  
            for (int i = 0; i < threadcount; i++)  
            {  
                List<string> tempList = new List<string>();  
                if (i == threadcount - 1)  
                    for (int j = min; j < ListWords.Count; j++)  
                        tempList.Add(ListWords[j]);  
                else  
                    for (int j = min; j < min + wordthread; j++)  
                    {  
                        tempList.Add(ListWords[j]);  
                    }  
  
                tasks[i] = new Task<List<string>>(ArrayThreadTask, new Tuple<string, int, List<string>>(checkword, maxdistance, tempList));  
  
                //Запуск потока  
                tasks[i].Start();  
                min += wordthread;  
            }  
            Task.WaitAll(tasks);  
            for (int i = 0; i < threadcount; i++)  
            {  
                foreach (var x in tasks[i].Result)  
                {  
                    this.ListSimilarWords.Items.Add(x.ToString());  
                }  
            }  
            this.ListSimilarWords.EndUpdate();  
        }  
  
        public List<string> ArrayThreadTask(object paramObj)  
        {  
            Tuple<string, int, List<string>> param = (Tuple<string, int, List<string>>)paramObj;  
            string checkword = param.Item1;  
            int maxdistance = param.Item2;  
            List<string> voidList = new List<string>();  
            List<string> tempList = param.Item3;  
            if (!string.IsNullOrWhiteSpace(checkword) && tempList.Count > 0)  
            {  
                string checkwordup = checkword.ToUpper();  
                List<string> SimilarWords = new List<string>();  
                foreach (string word in tempList)  
                {  
                    int distance = LevenshteinDistance(checkwordup, word.ToUpper());  
                    if (distance <= maxdistance) SimilarWords.Add(word);  
                }  
                return SimilarWords;  
            }  
            return voidList;  
        }  
  
        private void button2\_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.ReadTime.Text + "</td>");  
                b.AppendLine("</tr>");  
  
                b.AppendLine("<tr>");  
                b.AppendLine("<td>Количество слов в файле</td>");  
                b.AppendLine("<td>" + this.WordsCount.Text + "</td>");  
                b.AppendLine("</tr>");  
  
                b.AppendLine("<tr>");  
                b.AppendLine("<td>Слово для поиска</td>");  
                b.AppendLine("<td>" + this.CheckWord.Text + "</td>");  
                b.AppendLine("</tr>");  
  
                b.AppendLine("<tr>");  
                b.AppendLine("<td>Максимальное расстояние для нечеткого поиска</td>");  
                b.AppendLine("<td>" + this.MaxDistance.Text + "</td>");  
                b.AppendLine("</tr>");  
  
                b.AppendLine("<tr>");  
                b.AppendLine("<td>Количество потоков</td>");  
                b.AppendLine("<td>" + this.textBox1.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.ListSimilarWords.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);  
            }  
        }  
    }  
}

using System;  
using System.Windows.Forms;  
  
namespace WindowsFormsApplication1  
{  
    static class Program  
    {  
        /// <summary>  
        /// Главная точка входа для приложения.  
        /// </summary>  
        [STAThread]  
        static void Main()  
        {  
            Application.EnableVisualStyles();  
            Application.SetCompatibleTextRenderingDefault(false);  
            Application.Run(new Form1());  
        }  
    }  
}

# Результат работы программы

