Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

Московский государственный технический университет имени Н.Э.Баумана

(МГТУ им. Н.Э.Баумана)

**Пояснительная записка по выполнению домашнего задания по курсу БКИТ**

Выполнил: Труфанов Вадим, ИУ 5-33

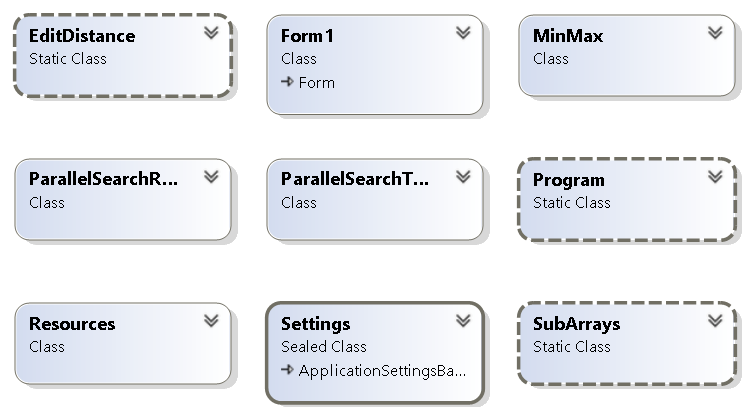
Преподаватель: Гапанюк Ю.Е.

Москва, 2018

1. **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).
4. Реализуйте функцию записи результатов поиска в файл отчета. Файл отчета создается в формате .txt или .html.
5. **Диаграмма классов**



1. **Текст программы**

*Класс Program*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace DZ

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

*Класс EditDistance*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DZ

{

public static class EditDistance

{

/// <summary>

/// Вычисление расстояния Дамерау-Левенштейна

/// </summary>

public static int Distance(string str1Param, string str2Param, bool Damerau)

{

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);

if (Damerau)

{

//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];

}

}

}

*Класс MinMax*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DZ

{

public class MinMax

{

public int Min { get; set; }

public int Max { get; set; }

public MinMax(int pmin, int pmax)

{

this.Min = pmin;

this.Max = pmax;

}

}

}

*Класс ParallelSearchResult*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DZ

{

/// <summary>

/// Результаты параллельного поиска

/// </summary>

public class ParallelSearchResult

{

/// <summary>

/// Найденное слово

/// </summary>

public string word { get; set; }

/// <summary>

/// Расстояние

/// </summary>

public int dist { get; set; }

/// <summary>

/// Номер потока

/// </summary>

public int ThreadNum { get; set; }

}

}

*Класс ParallelSearchThreadParam*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DZ

{

/// <summary>

/// Параметры которые передаются в поток для параллельного поиска

/// </summary>

class ParallelSearchThreadParam

{

/// <summary>

/// Массив для поиска

/// </summary>

public List<string> tempList { get; set; }

/// <summary>

/// Слово для поиска

/// </summary>

public string wordPattern { get; set; }

/// <summary>

/// Максимальное расстояние для нечеткого поиска

/// </summary>

public int maxDist { get; set; }

/// <summary>

/// Номер потока

/// </summary>

public int ThreadNum { get; set; }

public bool IsDamerau { get; set; }

}

}

Класс SubArrays

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DZ

{

/// <summary>

/// Класс для деления массива на последовательности

/// </summary>

public static class SubArrays

{

/// <summary>

/// Деление массива на последовательности

/// </summary>

/// <param name="beginIndex">Начальный индекс массива</param>

/// <param name="endIndex">Конечный индекс массива</param>

/// <param name="subArraysCount">Требуемое количество подмассивов</param>

/// <returns>Список пар с индексами подмассивов</returns>

public static List<MinMax> DivideSubArrays(int beginIndex, int endIndex, int subArraysCount)

{

//Результирующий список пар с индексами подмассивов

List<MinMax> result = new List<MinMax>();

//Если число элементов в массиве слишком мало для деления

//то возвращается массив целиком

if ((endIndex - beginIndex) <= subArraysCount)

{

result.Add(new MinMax(0, (endIndex - beginIndex)));

}

else

{

//Размер подмассива

int delta = (endIndex - beginIndex) / subArraysCount;

//Начало отсчета

int currentBegin = beginIndex;

//Пока размер подмассива укладывается в оставшуюся последовательность

while ((endIndex - currentBegin) >= 2 \* delta)

{

//Формируем подмассив на основе начала последовательности

result.Add(new MinMax(currentBegin, currentBegin + delta));

//Сдвигаем начало последовательности вперед на размер подмассива

currentBegin += delta;

}

//Оставшийся фрагмент массива

result.Add(new MinMax(currentBegin, endIndex));

}

//Возврат списка результатов

return result;

}

}

}

*Класс Form1*

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace DZ

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private List<String> list;

public List<string> List { get => list; set => list = value; }

private void Form1\_Load(object sender, EventArgs e)

{

}

private void Open\_file\_button\_Click(object sender, EventArgs e)

{

List = new List<string>();

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', '\r' };

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 (str.Length > 0 && str != "-" && !List.Contains(str)) List.Add(str);

}

t.Stop();

this.textBox\_time.Text = t.Elapsed.ToString();

this.textBox\_words.Text = List.Count.ToString();

//foreach (var x in List) Console.WriteLine(x);

}

else

{

MessageBox.Show("Необходимо выбрать файл");

}

}

private void button\_search\_word\_Click(object sender, EventArgs e)

{

//Слово для поиска

string word = this.textBox\_search.Text.Trim();

//Если слово для поиска не пусто

if (!string.IsNullOrWhiteSpace(word) && List.Count > 0)

{

if (!int.TryParse(this.distanceNum.Text.Trim(), out int distanceNum))

{

MessageBox.Show("Необходимо указать максимальное расстояние");

return;

}

if (distanceNum < 1 || distanceNum > 6)

{

MessageBox.Show("Максимальное расстояние должно быть в диапазоне от 1 до 6");

return;

}

int ThreadCount;

if (!int.TryParse(this.textBox\_threads.Text.Trim(), out ThreadCount))

{

MessageBox.Show("Необходимо указать количество потоков");

return;

}

bool isDamerau = checkBox\_Damerau.Checked;

Stopwatch t = new Stopwatch();

t.Start();

//-------------------------------------------------

// Начало параллельного поиска

//-------------------------------------------------

//Результирующий список

List<ParallelSearchResult> Result = new List<ParallelSearchResult>();

//Деление списка на фрагменты для параллельного запуска в потоках

List<MinMax> arrayDivList = SubArrays.DivideSubArrays(0, list.Count, ThreadCount);

int count = arrayDivList.Count;

//Количество потоков соответствует количеству фрагментов массива

Task<List<ParallelSearchResult>>[] tasks = new Task<List<ParallelSearchResult>>[count];

//Запуск потоков

for (int i = 0; i < count; i++)

{

//Создание временного списка, чтобы потоки не работали параллельно с одной коллекцией

List<string> tempTaskList = list.GetRange(arrayDivList[i].Min, arrayDivList[i].Max - arrayDivList[i].Min);

tasks[i] = new Task<List<ParallelSearchResult>>(

//Метод, который будет выполняться в потоке

ArrayThreadTask,

//Параметры потока

new ParallelSearchThreadParam()

{

tempList = tempTaskList,

maxDist = distanceNum,

ThreadNum = i,

wordPattern = word,

IsDamerau = isDamerau

});

//Запуск потока

tasks[i].Start();

}

Task.WaitAll(tasks);

t.Stop();

//Объединение результатов

for (int i = 0; i < count; i++)

{

Result.AddRange(tasks[i].Result);

}

//-------------------------------------------------

// Завершение параллельного поиска

//-------------------------------------------------

t.Stop();

//Вывод результатов

//Время поиска

this.textBox\_search\_time.Text = t.Elapsed.ToString();

//Вычисленное количество потоков

this.textBoxThreadCountAll.Text = count.ToString();

//Начало обновления списка результатов

this.listBox\_result.BeginUpdate();

//Очистка списка

this.listBox\_result.Items.Clear();

//Вывод результатов поиска

foreach (var x in Result)

{

string temp = x.word + " (расстояние=" + x.dist.ToString() + " поток=" + x.ThreadNum.ToString() + ")";

this.listBox\_result.Items.Add(temp);

}

//Окончание обновления списка результатов

this.listBox\_result.EndUpdate();

}

else

{

MessageBox.Show("Необходимо выбрать файл и ввести слово для поиска");

}

}

/// <summary>

/// Выполняется в параллельном потоке для поиска строк

/// </summary>

public static List<ParallelSearchResult> ArrayThreadTask(object paramObj)

{

ParallelSearchThreadParam param = (ParallelSearchThreadParam)paramObj;

//Слово для поиска в верхнем регистре

string wordUpper = param.wordPattern.Trim().ToUpper();

//Результаты поиска в одном потоке

List<ParallelSearchResult> Result = new List<ParallelSearchResult>();

//Перебор всех слов во временном списке данного потока

foreach (string str in param.tempList)

{

//Вычисление расстояния Дамерау-Левенштейна

int dist = EditDistance.Distance(str.ToUpper(), wordUpper, param.IsDamerau);

//Если расстояние меньше порогового, то слово добавляется в результат

if (dist <= param.maxDist)

{

ParallelSearchResult temp = new ParallelSearchResult()

{

word = str,

dist = dist,

ThreadNum = param.ThreadNum

};

Result.Add(temp);

}

}

return Result;

}

private void button\_simple\_search\_Click(object sender, EventArgs e)

{

//Слово для поиска

string word = this.textBox\_search.Text.Trim();

//Если слово для поиска не пусто

if (!string.IsNullOrWhiteSpace(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.textBox\_simple\_time.Text = t.Elapsed.ToString();

this.listBox\_result.BeginUpdate();

//Очистка списка

this.listBox\_result.Items.Clear();

//Вывод результатов поиска

foreach (string str in tempList)

{

this.listBox\_result.Items.Add(str);

}

this.listBox\_result.EndUpdate();

}

else

{

MessageBox.Show("Необходимо выбрать файл и ввести слово для поиска");

}

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void textBox\_search\_time\_TextChanged(object sender, EventArgs e)

{

}

private void label8\_Click(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

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

{

this.Close();

}

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 = ".txt";

fd.Filter = "текстовые файлы|\*.txt";

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

{

string ReportFileName = fd.FileName;

StringBuilder b = new StringBuilder();

b.AppendLine("Отчет: " + ReportFileName + "\n\n\n");

b.AppendLine("Время чтения из файла: ");

b.AppendLine("\t" + this.textBox\_time.Text + "\n\n");

b.AppendLine("Количество уникальных слов в файле: ");

b.AppendLine("\t" + this.textBox\_words.Text + "\n\n");

b.AppendLine("Слово для поиска: ");

b.AppendLine("\t" + this.textBox\_search.Text + "\n\n");

b.AppendLine("Максимальное расстояние для нечеткого поиска: ");

b.AppendLine("\t" + this.distanceNum.Text + "\n\n");

b.AppendLine("Время четкого поиска: ");

b.AppendLine("\t" + this.textBox\_simple\_time.Text + "\n\n");

b.AppendLine("Время нечеткого поиска");

b.AppendLine("\t" + this.textBox\_search\_time.Text + "\n\n");

b.AppendLine("Результаты поиска:");

foreach (var x in this.listBox\_result.Items)

{

b.AppendLine("\t" + x.ToString());

}

//Сохранение файла

File.AppendAllText(ReportFileName, b.ToString());

MessageBox.Show("Отчет сформирован. Файл: " + ReportFileName);

}

}

}

}

*Класс Form1.Designer*

namespace DZ

{

partial class Form1

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

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

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

this.Open\_file\_button = new System.Windows.Forms.Button();

this.textBox\_time = new System.Windows.Forms.TextBox();

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

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

this.textBox\_words = new System.Windows.Forms.TextBox();

this.label3 = new System.Windows.Forms.Label();

this.textBox\_search = new System.Windows.Forms.TextBox();

this.button\_search\_word = new System.Windows.Forms.Button();

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

this.textBox\_search\_time = new System.Windows.Forms.TextBox();

this.listBox\_result = new System.Windows.Forms.ListBox();

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

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

this.checkBox\_Damerau = new System.Windows.Forms.CheckBox();

this.label6 = new System.Windows.Forms.Label();

this.button\_simple\_search = new System.Windows.Forms.Button();

this.textBox\_simple\_time = new System.Windows.Forms.TextBox();

this.label7 = new System.Windows.Forms.Label();

this.textBox\_threads = new System.Windows.Forms.TextBox();

this.label8 = new System.Windows.Forms.Label();

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

this.label9 = new System.Windows.Forms.Label();

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

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

this.SuspendLayout();

//

// Open\_file\_button

//

this.Open\_file\_button.Location = new System.Drawing.Point(25, 14);

this.Open\_file\_button.Name = "Open\_file\_button";

this.Open\_file\_button.Size = new System.Drawing.Size(175, 48);

this.Open\_file\_button.TabIndex = 0;

this.Open\_file\_button.Text = "Чтение из файла";

this.Open\_file\_button.UseVisualStyleBackColor = true;

this.Open\_file\_button.Click += new System.EventHandler(this.Open\_file\_button\_Click);

//

// textBox\_time

//

this.textBox\_time.Location = new System.Drawing.Point(429, 39);

this.textBox\_time.Name = "textBox\_time";

this.textBox\_time.Size = new System.Drawing.Size(293, 26);

this.textBox\_time.TabIndex = 3;

//

// label1

//

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(226, 42);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(197, 20);

this.label1.TabIndex = 4;

this.label1.Text = "Время чтения из файла:";

//

// label2

//

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(226, 92);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(309, 20);

this.label2.TabIndex = 5;

this.label2.Text = "Количество уникальных слов в файле: ";

//

// textBox\_words

//

this.textBox\_words.Location = new System.Drawing.Point(542, 92);

this.textBox\_words.Name = "textBox\_words";

this.textBox\_words.Size = new System.Drawing.Size(180, 26);

this.textBox\_words.TabIndex = 6;

//

// label3

//

this.label3.AutoSize = true;

this.label3.Location = new System.Drawing.Point(226, 138);

this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(223, 20);

this.label3.TabIndex = 7;

this.label3.Text = "Введите слово для поиска: ";

//

// textBox\_search

//

this.textBox\_search.Location = new System.Drawing.Point(455, 135);

this.textBox\_search.Name = "textBox\_search";

this.textBox\_search.Size = new System.Drawing.Size(267, 26);

this.textBox\_search.TabIndex = 8;

//

// button\_search\_word

//

this.button\_search\_word.Location = new System.Drawing.Point(25, 264);

this.button\_search\_word.Name = "button\_search\_word";

this.button\_search\_word.Size = new System.Drawing.Size(175, 72);

this.button\_search\_word.TabIndex = 9;

this.button\_search\_word.Text = "Нечеткий поиск (параллельный)";

this.button\_search\_word.UseVisualStyleBackColor = true;

this.button\_search\_word.Click += new System.EventHandler(this.button\_search\_word\_Click);

//

// label4

//

this.label4.AutoSize = true;

this.label4.Location = new System.Drawing.Point(226, 400);

this.label4.Name = "label4";

this.label4.Size = new System.Drawing.Size(204, 20);

this.label4.TabIndex = 10;

this.label4.Text = "Время нечеткого поиска: ";

this.label4.Click += new System.EventHandler(this.label4\_Click);

//

// textBox\_search\_time

//

this.textBox\_search\_time.Location = new System.Drawing.Point(436, 397);

this.textBox\_search\_time.Name = "textBox\_search\_time";

this.textBox\_search\_time.Size = new System.Drawing.Size(286, 26);

this.textBox\_search\_time.TabIndex = 11;

this.textBox\_search\_time.TextChanged += new System.EventHandler(this.textBox\_search\_time\_TextChanged);

//

// listBox\_result

//

this.listBox\_result.FormattingEnabled = true;

this.listBox\_result.ItemHeight = 20;

this.listBox\_result.Location = new System.Drawing.Point(25, 473);

this.listBox\_result.Name = "listBox\_result";

this.listBox\_result.Size = new System.Drawing.Size(693, 184);

this.listBox\_result.TabIndex = 12;

//

// label5

//

this.label5.AutoSize = true;

this.label5.Location = new System.Drawing.Point(226, 264);

this.label5.Name = "label5";

this.label5.Size = new System.Drawing.Size(393, 20);

this.label5.TabIndex = 13;

this.label5.Text = "Введите макимальное расстояние Левенштейна: ";

//

// distanceNum

//

this.distanceNum.Location = new System.Drawing.Point(622, 264);

this.distanceNum.Name = "distanceNum";

this.distanceNum.Size = new System.Drawing.Size(100, 26);

this.distanceNum.TabIndex = 14;

//

// checkBox\_Damerau

//

this.checkBox\_Damerau.AutoSize = true;

this.checkBox\_Damerau.Location = new System.Drawing.Point(673, 224);

this.checkBox\_Damerau.Name = "checkBox\_Damerau";

this.checkBox\_Damerau.Size = new System.Drawing.Size(22, 21);

this.checkBox\_Damerau.TabIndex = 15;

this.checkBox\_Damerau.UseVisualStyleBackColor = true;

//

// label6

//

this.label6.AutoSize = true;

this.label6.Location = new System.Drawing.Point(226, 223);

this.label6.Name = "label6";

this.label6.Size = new System.Drawing.Size(371, 20);

this.label6.TabIndex = 16;

this.label6.Text = "Отметьте флажок для Дамерау-Левенштейна ";

//

// button\_simple\_search

//

this.button\_simple\_search.Location = new System.Drawing.Point(25, 125);

this.button\_simple\_search.Name = "button\_simple\_search";

this.button\_simple\_search.Size = new System.Drawing.Size(175, 46);

this.button\_simple\_search.TabIndex = 17;

this.button\_simple\_search.Text = "Четкий поиск";

this.button\_simple\_search.UseVisualStyleBackColor = true;

this.button\_simple\_search.Click += new System.EventHandler(this.button\_simple\_search\_Click);

//

// textBox\_simple\_time

//

this.textBox\_simple\_time.Location = new System.Drawing.Point(408, 179);

this.textBox\_simple\_time.Name = "textBox\_simple\_time";

this.textBox\_simple\_time.Size = new System.Drawing.Size(314, 26);

this.textBox\_simple\_time.TabIndex = 19;

//

// label7

//

this.label7.AutoSize = true;

this.label7.Location = new System.Drawing.Point(226, 179);

this.label7.Name = "label7";

this.label7.Size = new System.Drawing.Size(186, 20);

this.label7.TabIndex = 18;

this.label7.Text = "Время четкого поиска: ";

//

// textBox\_threads

//

this.textBox\_threads.Location = new System.Drawing.Point(429, 314);

this.textBox\_threads.Name = "textBox\_threads";

this.textBox\_threads.Size = new System.Drawing.Size(293, 26);

this.textBox\_threads.TabIndex = 21;

this.textBox\_threads.TextChanged += new System.EventHandler(this.textBox1\_TextChanged);

//

// label8

//

this.label8.AutoSize = true;

this.label8.Location = new System.Drawing.Point(226, 317);

this.label8.Name = "label8";

this.label8.Size = new System.Drawing.Size(170, 20);

this.label8.TabIndex = 20;

this.label8.Text = "Количество потоков ";

this.label8.Click += new System.EventHandler(this.label8\_Click);

//

// textBoxThreadCountAll

//

this.textBoxThreadCountAll.Location = new System.Drawing.Point(509, 354);

this.textBoxThreadCountAll.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);

this.textBoxThreadCountAll.Name = "textBoxThreadCountAll";

this.textBoxThreadCountAll.ReadOnly = true;

this.textBoxThreadCountAll.Size = new System.Drawing.Size(213, 26);

this.textBoxThreadCountAll.TabIndex = 23;

//

// label9

//

this.label9.AutoSize = true;

this.label9.Location = new System.Drawing.Point(226, 357);

this.label9.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.label9.Name = "label9";

this.label9.Size = new System.Drawing.Size(275, 20);

this.label9.TabIndex = 22;

this.label9.Text = "Вычисленное количество потоков:";

//

// buttonSaveReport

//

this.buttonSaveReport.Location = new System.Drawing.Point(25, 671);

this.buttonSaveReport.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);

this.buttonSaveReport.Name = "buttonSaveReport";

this.buttonSaveReport.Size = new System.Drawing.Size(210, 35);

this.buttonSaveReport.TabIndex = 24;

this.buttonSaveReport.Text = "Сохранение отчета";

this.buttonSaveReport.UseVisualStyleBackColor = true;

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

//

// buttonExit

//

this.buttonExit.Location = new System.Drawing.Point(540, 671);

this.buttonExit.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);

this.buttonExit.Name = "buttonExit";

this.buttonExit.Size = new System.Drawing.Size(178, 35);

this.buttonExit.TabIndex = 25;

this.buttonExit.Text = "Выход";

this.buttonExit.UseVisualStyleBackColor = true;

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

//

// Form1

//

this.AutoScaleDimensions = new System.Drawing.SizeF(9F, 20F);

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

this.ClientSize = new System.Drawing.Size(771, 720);

this.Controls.Add(this.buttonExit);

this.Controls.Add(this.buttonSaveReport);

this.Controls.Add(this.textBoxThreadCountAll);

this.Controls.Add(this.label9);

this.Controls.Add(this.textBox\_threads);

this.Controls.Add(this.label8);

this.Controls.Add(this.textBox\_simple\_time);

this.Controls.Add(this.label7);

this.Controls.Add(this.button\_simple\_search);

this.Controls.Add(this.label6);

this.Controls.Add(this.checkBox\_Damerau);

this.Controls.Add(this.distanceNum);

this.Controls.Add(this.label5);

this.Controls.Add(this.listBox\_result);

this.Controls.Add(this.textBox\_search\_time);

this.Controls.Add(this.label4);

this.Controls.Add(this.button\_search\_word);

this.Controls.Add(this.textBox\_search);

this.Controls.Add(this.label3);

this.Controls.Add(this.textBox\_words);

this.Controls.Add(this.label2);

this.Controls.Add(this.label1);

this.Controls.Add(this.textBox\_time);

this.Controls.Add(this.Open\_file\_button);

this.Name = "Form1";

this.Text = "Form1";

this.Load += new System.EventHandler(this.Form1\_Load);

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.Button Open\_file\_button;

private System.Windows.Forms.TextBox textBox\_time;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.TextBox textBox\_words;

private System.Windows.Forms.Label label3;

private System.Windows.Forms.TextBox textBox\_search;

private System.Windows.Forms.Button button\_search\_word;

private System.Windows.Forms.Label label4;

private System.Windows.Forms.TextBox textBox\_search\_time;

private System.Windows.Forms.ListBox listBox\_result;

private System.Windows.Forms.Label label5;

private System.Windows.Forms.TextBox distanceNum;

private System.Windows.Forms.CheckBox checkBox\_Damerau;

private System.Windows.Forms.Label label6;

private System.Windows.Forms.Button button\_simple\_search;

private System.Windows.Forms.TextBox textBox\_simple\_time;

private System.Windows.Forms.Label label7;

private System.Windows.Forms.TextBox textBox\_threads;

private System.Windows.Forms.Label label8;

private System.Windows.Forms.TextBox textBoxThreadCountAll;

private System.Windows.Forms.Label label9;

private System.Windows.Forms.Button buttonSaveReport;

private System.Windows.Forms.Button buttonExit;

}

}

1. **Экранные формы с примерами выполнения программы**

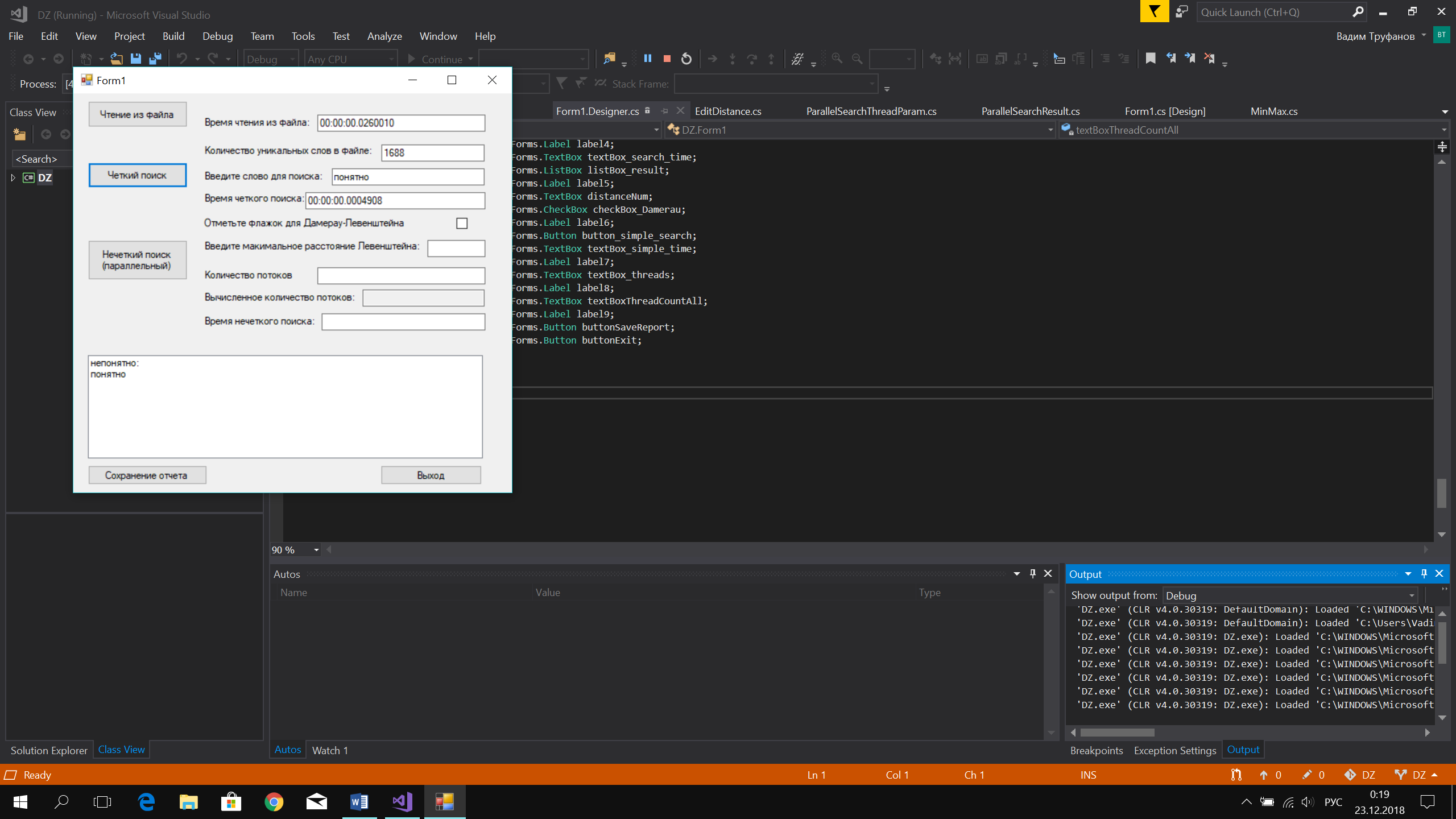


Рисунок . Четкий поиск

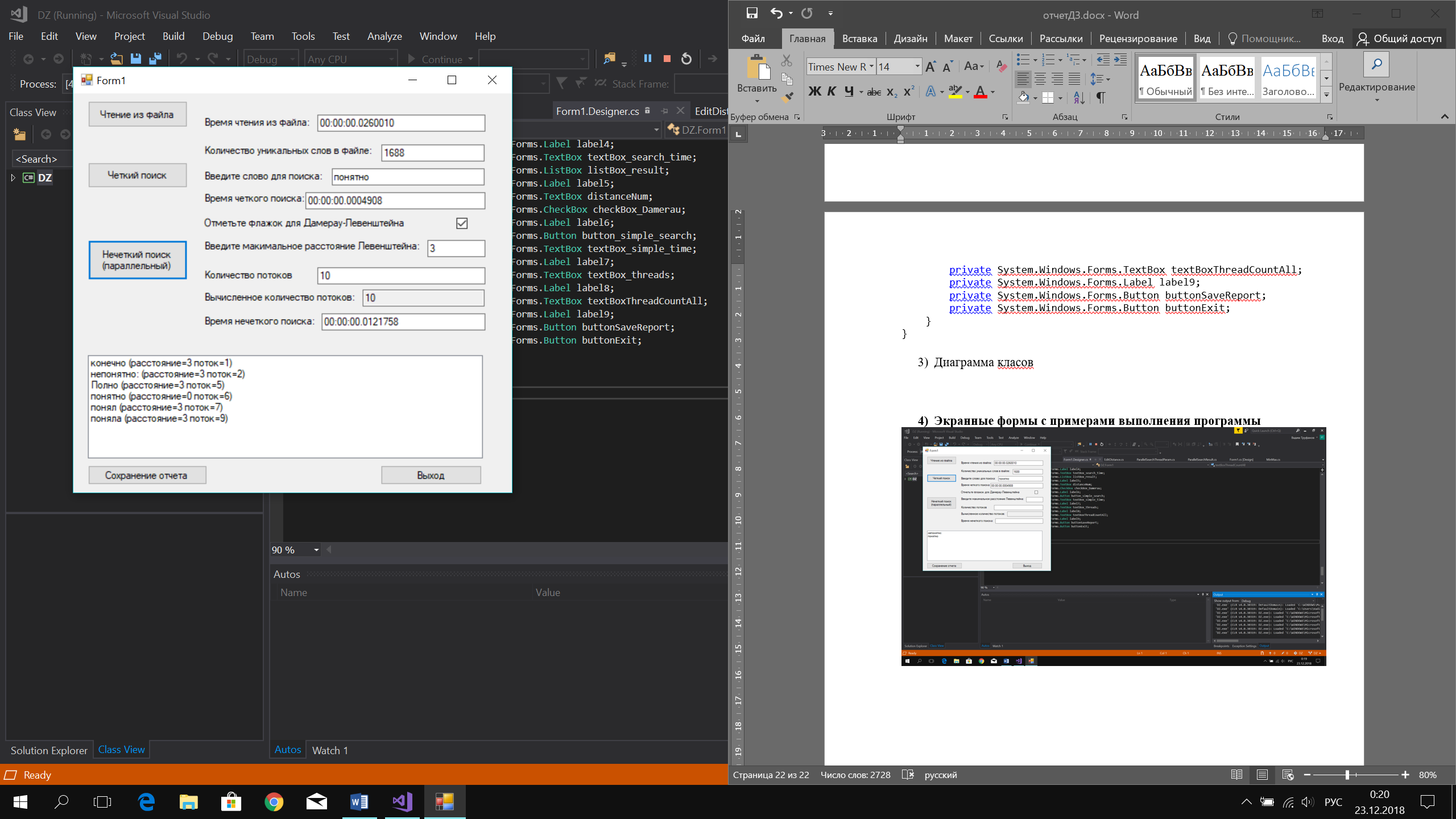


Рисунок . Нечеткий поиск

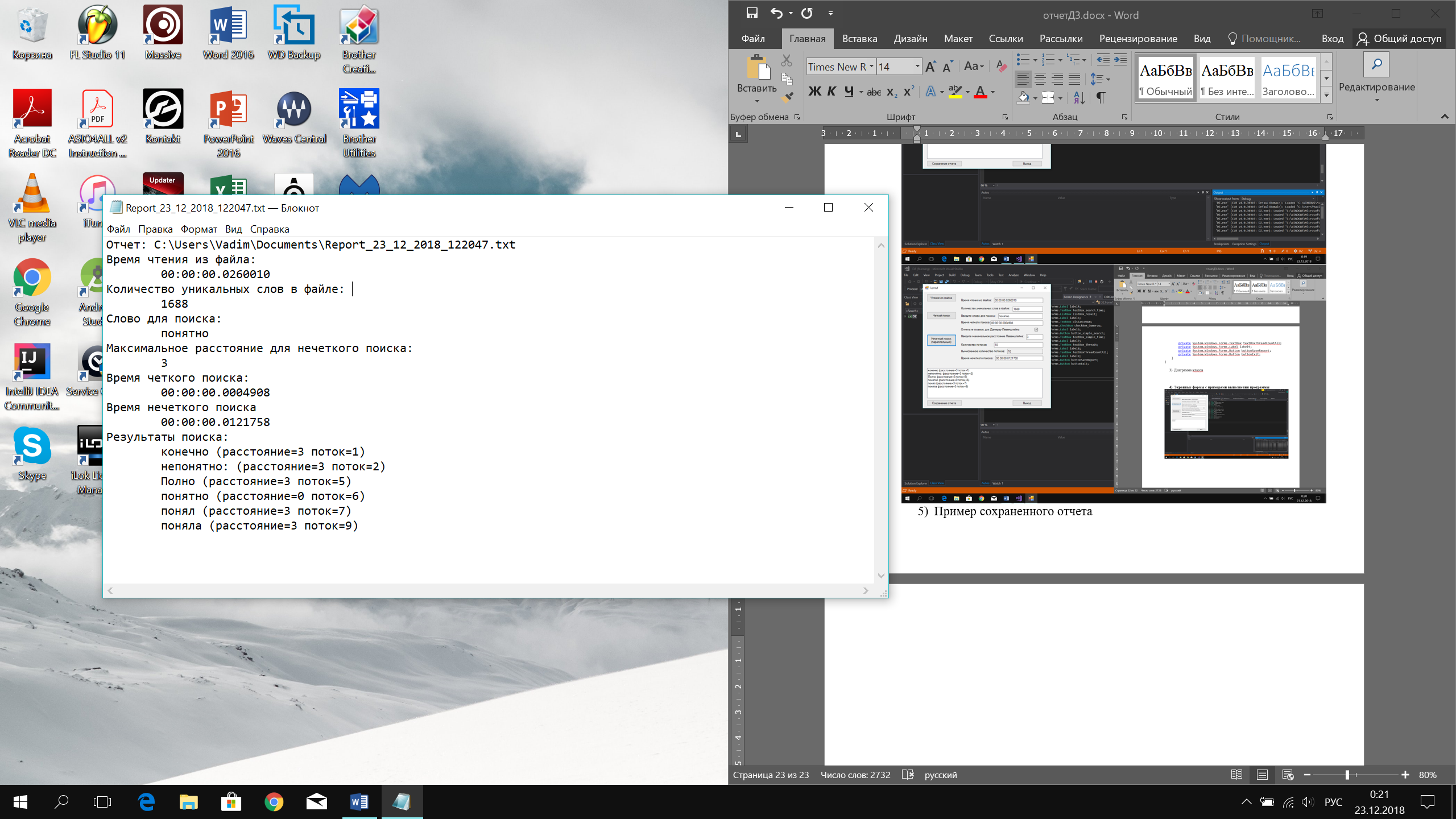


Рисунок . Пример сохраненного отчета в формате .txt