**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет технологий»

Пояснительная записка к домашней работе

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-33Б |  | преподаватель каф. ИУ5 |
| Кузнецов Григорий |  |  |
| Подпись и дата: |  | Подпись и дата: |

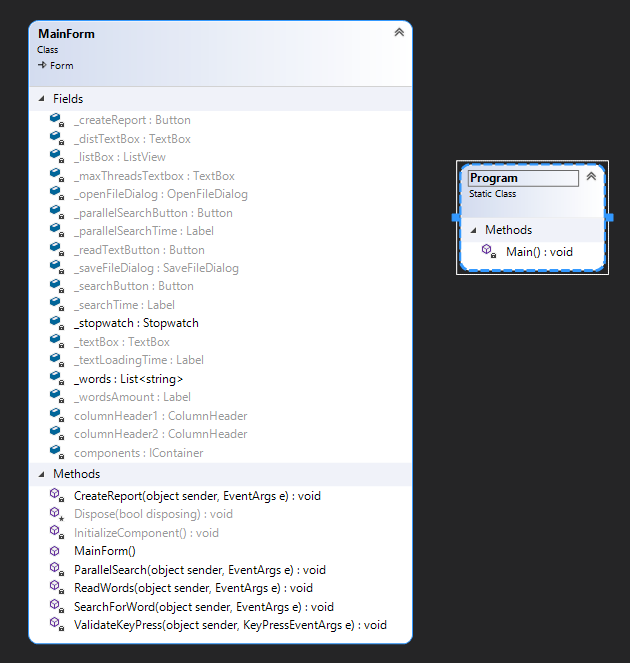
Москва, 2020 г.

**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**

using System;

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Windows.Forms;

using System.IO;

using System.Diagnostics;

using System.Threading.Tasks;

using System.Text;

namespace HMW

{

public partial class MainForm : Form

{

private readonly List<string> \_words;

private readonly Stopwatch \_stopwatch;

public MainForm()

{

InitializeComponent();

\_words = new List<string>();

\_stopwatch = new Stopwatch();

}

private void ReadWords(object sender, EventArgs e)

{

if (\_openFileDialog.ShowDialog() == DialogResult.OK)

{

\_listBox.Items.Clear();

\_stopwatch.Start();

var text = File.ReadAllText(\_openFileDialog.FileName);

foreach (var word in text.Split(" \n\t\f\r,./\\[]{ }<>@#$%^&\*();'\"|?!".ToCharArray()))

{

if (!\_words.Contains(word) && word != "")

\_words.Add(word);

}

\_stopwatch.Stop();

\_textLoadingTime.Text = $"Время загрузки текста: {\_stopwatch.Elapsed}";

\_wordsAmount.Text = $"Количество слов: {\_words.Count}";

\_searchButton.Enabled = true;

\_parallelSearchButton.Enabled = true;

\_maxThreadsTextbox.Enabled = true;

\_distTextBox.Enabled = true;

\_textBox.Enabled = true;

\_stopwatch.Reset();

}

}

private void SearchForWord(object sender, EventArgs e)

{

int maxDist;

try

{

maxDist = Convert.ToInt32(\_distTextBox.Text);

}

catch (FormatException)

{

MessageBox.Show("Неправильный формат ввода максимальной дистанции", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

\_listBox.BeginUpdate();

\_listBox.Items.Clear();

\_stopwatch.Start();

foreach (var word in \_words)

{

if (word.Contains(\_textBox.Text))

{

var dist = Lab5.Methods.LevenshteinDistance(word, \_textBox.Text);

if (dist <= maxDist)

\_listBox.Items.Add(new ListViewItem(new string[] { word, dist.ToString() }));

}

}

\_stopwatch.Stop();

\_listBox.EndUpdate();

\_searchTime.Text = $"Время поиска слов: {\_stopwatch.Elapsed}";

\_stopwatch.Reset();

}

private void ValidateKeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar < '0' || e.KeyChar > '9')

e.Handled = true;

}

private void ParallelSearch(object sender, EventArgs e)

{

if (!int.TryParse(\_maxThreadsTextbox.Text, out int threadsCount))

{

MessageBox.Show("Неправильно введено максимальное количество потокв", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if (!int.TryParse(\_distTextBox.Text, out int maxDist))

{

MessageBox.Show("Неправильный формат ввода максимальной дистанции", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

\_listBox.BeginUpdate();

\_listBox.Items.Clear();

\_stopwatch.Start();

Parallel.ForEach(\_words, new ParallelOptions() { MaxDegreeOfParallelism = threadsCount }, word =>

{

if (word.Contains(\_textBox.Text))

{

var dist = Lab5.Methods.LevenshteinDistance(word, \_textBox.Text);

if (dist <= maxDist)

\_listBox.Items.Add(new ListViewItem(new string[] { word, dist.ToString() }));

}

});

\_stopwatch.Stop();

\_listBox.EndUpdate();

\_parallelSearchTime.Text = $"Время параллельного поиска: {\_stopwatch.Elapsed / threadsCount}";

\_stopwatch.Reset();

}

private void CreateReport(object sender, EventArgs e)

{

\_saveFileDialog.FileName = "Report\_" + DateTime.Now.ToString("yyyy\_mm\_dd\_h\_m\_s");

if (\_saveFileDialog.ShowDialog() == DialogResult.OK)

{

var report = new StringBuilder();

report.AppendLine($"Отчёт: {\_saveFileDialog.FileName}");

report.AppendLine(\_textLoadingTime.Text);

report.AppendLine(\_wordsAmount.Text);

report.AppendLine($"Слово для поиска: {\_textBox.Text}");

report.AppendLine($"Максимальное расстояние для поиска: {\_distTextBox.Text}");

report.AppendLine($"Количество потоков: {\_maxThreadsTextbox.Text}");

report.AppendLine(\_searchTime.Text);

report.AppendLine(\_parallelSearchTime.Text);

report.AppendLine("Результаты поиска:");

foreach (ListViewItem item in \_listBox.Items)

{

report.AppendLine($"\t{item.SubItems[0].Text} - {item.SubItems[1].Text}");

}

File.WriteAllText(\_saveFileDialog.FileName, report.ToString());

}

}

}

}

Text.txt:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Senectus e

t netus et malesuada fames ac turpis egestas maecenas. Vel fringilla est ullamcorper eget. Amet tellus cras adipiscing enim eu

turpis egestas. Mi eget mauris pharetra et ultrices neque ornare. Nec feugiat nisl pretium fusce id. Amet consectetur adipiscing elit ut. Id consectetur purus ut faucibus. Feugiat nisl pretium fusce id velit ut tortor pretium viverra. Quisque non tellus orci ac auctor augue mauris. Tellus elementum sagittis vitae et leo duis. Blandit libero volutpat sed cras ornare arcu dui.

Blandit aliquam etiam erat velit. Euismod elementum nisi quis eleifend quam adipiscing vitae. Duis at consectetur lorem donec massa

sapien faucibus et. Sapien nec sagittis aliquam malesuada. Sed risus pretium quam vulputate dignissim suspendisse in est. Pulvinar neque laoreet suspendisse interdum consectetur. Dapibus ultrices in iaculis nunc sed. Morbi tincidunt augue interdum velit euismod in pellentesque massa. Ultricies integer quis auctor elit sed vulputate mi sit amet. Massa ultricies mi quis hendrerit dolor magna eget est lorem. Non sodales neque sodales ut.

Ultrices in iaculis nunc sed. Viverra orci sagittis eu volutpat odio facilisis mauris sit amet. At augue eget arcu dictu

m varius duis. Accumsan lacus vel facilisis volutpat est velit egestas dui. Velit scelerisque in dictum non consectetur a erat nam. Tellus mauris a diam maecenas sed enim ut. Ullamcorper sit amet risus nullam eget felis eget. Ullamcorper dignissim cras tincidunt lobortis. Urna id volutpat lacus laoreet non curabitur gravida arcu. Nunc faucibus a pellentesque sit amet porttitor.

Adipiscing diam donec adipiscing tristique risus nec feugiat. Odio eu feugiat pretium nibh ipsum consequat. Eu ultrices vita

e auctor eu augue. Mattis vulputate enim nulla aliquet porttitor. Hendrerit gravida rutrum quisque non tellus. Sit amet risus nullam eget felis. Dolor sit amet consectetur adipiscing elit ut aliquam. Et tortor consequat id porta nibh. Faucibus scelerisque eleifend donec pretium. Odio ut enim blandit volutpat maecenas.

Accumsan tortor posuere ac ut consequat semper viverra nam libero. A arcu cursus vitae congue mauris rhoncus. Tincidunt augue interdum velit euismod in. Dictum varius duis at consectetur lorem donec massa. Quam id leo in vitae turpis massa sed elementum tempus. In hac habitasse platea dictumst vestibulum rhoncus. Diam vel quam elementum pulvinar etiam non.

Euismod nisi porta lorem mollis aliquam ut porttitor.

Mattis aliquam faucibus purus in massa tempor nec. In nulla p

osuere sollicitudin aliquam. Tortor aliquam nulla facilisi cras fermentum odio eu feugiat. Nibh praesent tristique magna sit. Sed vulputate mi sit amet mauris commodo quis. Risus quis varius quam

quisque id diam vel quam elementum. Etiam sit amet nisl purus in mollis. Non pulvinar neque laoreet suspendisse interdum consectetur libero. Fames ac turpis egestas integer eget aliquet nibh praesent tristique. Sagittis nisl rhoncus mattis rhoncus. Amet porttitor eget dolor morbi non arcu risus quis. Eget lorem dolor sed viverra ipsum nunc aliquet bibendum enim. Convallis tellus id interdum velit. Iaculis eu non diam phasellus vestibulum. Aliquet lectus proin nibh nisl condimentum id venenatis. Fames ac turpis egestas maecenas pharetra convallis posuere morbi leo. Enim nunc faucibus

a pellentesque. Odio ut sem nulla pharetra diam sit.

**Пример выполнения программы**

