ФЕДЕРАЛЬНОЕ Государственное АВТОНОМНОЕ образовательное УЧРЕЖДЕНИЕ ВЫСШЕГО образования

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ**

**ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**

**(НИУ «БелГУ»)**

ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИРОВЫХ ТЕХНОЛОГИЙ

**Кафедра прикладной информатики и информационных технологий**

**Лабораторная работа 2**

**по дисциплине «Разработка программных приложений на языках высокого уровня»**

**студента очной формы обучения**

**направления подготовки 09.03.03. «Прикладная информатика»**

**3 курса группы 12002005**

**Морозова Максима Андреевича**

Научный руководитель:

ст. пр. Лифиренко Максим Вячеславович

|  |  |
| --- | --- |
| ***Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2023 г.  \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Подпись (расшифровка подписи) |  |

**БЕЛГОРОД, 2023**

Задания выполненные в данной работы:

Результат работы программы, представлен на рисунке 1:

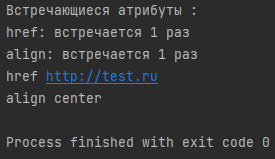


Рисунок 1 – результат работы программы

Листинг программы:  
/\*\*

Класс для извлечения и подсчета атрибутов из HTML-файла.

\* @author Maxim

\* @since 1.0

\*/

package Lab4;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.util.HashMap;

import java.util.Map;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class HtmlAttributesExtractor {

/\*\*

\* Главный метод программы.

\* @param args аргументы командной строки

\*/

public static void main(String[] args) {

// Имя файла для чтения

String inputFileName = "C:\\Users\\Maximka\\IdeaProjects\\lab1\\src\\Lab4\\input.html";

// Имя файла для записи

String outputFileName = "C:\\Users\\Maximka\\IdeaProjects\\lab1\\src\\Lab4\\output.txt";

// Карта для подсчета количества каждого атрибута

Map<String, Integer> attributeCountMap = new HashMap<>();

// Шаблон для поиска атрибутов

Pattern attributePattern = Pattern.compile("\\b(\\w+)\\s\*=\\s\*(['\"])(.\*?)\\2", Pattern.DOTALL);

try (BufferedReader reader = new BufferedReader(new FileReader(inputFileName));

BufferedWriter writer = new BufferedWriter(new FileWriter(outputFileName))) {

String line;

// Чтение файла построчно

while ((line = reader.readLine()) != null) {

Matcher matcher = attributePattern.matcher(line);

// Поиск атрибутов в строке

while (matcher.find()) {

String attributeName = matcher.group(1);

String attributeValue = matcher.group(3);

// Запись атрибута в файл

writer.write(attributeName + " " + attributeValue + "\n");

// Увеличение счетчика для данного атрибута

attributeCountMap.merge(attributeName, 1, Integer::sum);

}

}

} catch (IOException e) {

e.printStackTrace();

}

// Вывод информации о количестве каждого атрибута

System.out.println("Встречающиеся атрибуты :");

for (Map.Entry<String, Integer> entry : attributeCountMap.entrySet()) {

System.out.println(entry.getKey() + ": встречается " + entry.getValue()+ " раз");

}

// Чтение записанных атрибутов из файла

try (BufferedReader br = new BufferedReader(new FileReader("C:\\Users\\Maximka\\IdeaProjects\\lab1\\src\\Lab4\\output.txt"))) {

String line;

while ((line = br.readLine()) != null) {

System.out.println(line);

}

} catch (IOException e) {

e.printStackTrace();

}

}

}