ФЕДЕРАЛЬНОЕ Государственное АВТОНОМНОЕ образовательное УЧРЕЖДЕНИЕ ВЫСШЕГО образования

**БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ**

**ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**

**(НИУ «БелГУ»)**

**ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ**

КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

**Отчет**

по дисциплине «Разработка программных приложений на языках высокого уровня»

специальности 09.03.03 «Прикладная информатика»

**Лабораторная работа № 4**

студента очного отделения

3 курса 12002005 группы

Чернышева Владислава Максимовича

Проверил:

старший преподаватель

Лифиренко Максим Вячеславович

БЕЛГОРОД 2023

**Цель работы:** получить практические навыки работы с Collections в Java, изучить работу с регулярными выражениями.

(Вариант №20)

**Листинг класса Main**

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 Main {  
 */\*\*  
 \* Вариант №20  
 \* Этот метод принимает на считку html файл. Находит в нём ссылки и считает их количество далее ссылки и их количество записываются в файл.  
 \** ***@param*** *args  
 \*/* public static void main(String[] args) {  
 try {  
 // читаем файл с HTML-кодом  
 BufferedReader reader = new BufferedReader(new FileReader("src/input.html"));  
 StringBuilder sb = new StringBuilder();  
 String line;  
 while ((line = reader.readLine()) != null) {  
 sb.append(line);  
 }  
 reader.close();  
 String StringHtml = sb.toString();  
  
 // ищем все ссылки в HTML-коде с помощью регулярного выражения  
 Pattern pattern = Pattern.*compile*("<a\\s+[^>]\*href=['\"]([^'\"]+)['\"][^>]\*>");  
 Matcher matcher = pattern.matcher(StringHtml);  
 // создаем HashMap для хранения ссылок и их количества  
 Map<String, Integer> linkCount = new HashMap<>();  
 // проходимся по всем ссылкам  
 while (matcher.find()) {  
 // получаем найденную ссылку  
 String url = matcher.group(1);  
 // удаляем параметры запроса, если они есть  
 if (url.contains("?")) {  
 url = url.substring(0, url.indexOf("?"));  
 }  
 // если ссылка уже есть в HashMap, увеличиваем счетчик на 1  
 if (linkCount.containsKey(url)) {  
 linkCount.put(url, linkCount.get(url) + 1);  
 } else { // иначе добавляем новую ссылку в HashMap со счетчиком 1  
 linkCount.put(url, 1);  
 }  
 }  
 // записываем результат в файл  
 BufferedWriter writer = new BufferedWriter(new FileWriter("output.txt"));  
 for (Map.Entry<String, Integer> entry : linkCount.entrySet()) {  
 writer.write(entry.getKey() + " " + entry.getValue() + "\n");  
 }  
 writer.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

**Содержание файла для считки input.html**

<html>  
<a href='http://ya.ru '>Яндекс</a>  
<a href='http://ya.ru/maps '>Яндекс карты</a>  
<a  
 href='http://www.google.ru/search?hl=ru&client=firefox&hs=mpE&rls=en '>Поиск</a  
>  
<a href='http://ya.ru '>Яндекс</a>  
<a href='http://ya.ru/maps '>Яндекс карты</a>  
  
<a href='http://ya.ru '>Яндекс</a>  
</p>  
</html>

Результаты работы программы представлен на рисунке 1.

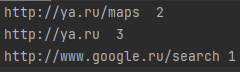


Рисунок 1 –Вывод информации в файл

Вывод: в ходе выполнения лабораторной работы были получены практические навыки работы с Collections в Java, и изучена работа с регулярными выражениями.