Лабораторная работа №7

по дисциплине

“Объектно-ориентированное программирование”

Тема:

«Веб-сканер»

Выполнила студентка

группы БФИ1901

Киселева Анна

Москва 2020

**Цель:**

Реализовать элементарный веб-сканер, который автоматически загружает веб-страницы из Интернета, ищет новые ссылки на этих страницах и повторяет их.

**Код программы:**

1. **Файл Crawler.java:**

import java.io.\*;

import java.net.\*;

import java.util.LinkedList;

public class Crawler {

static LinkedList <URLDepthPair> findLink = new LinkedList <URLDepthPair>();

static LinkedList <URLDepthPair> viewedLink = new LinkedList <URLDepthPair>();

public static void showResult(LinkedList<URLDepthPair> viewedLink) {

for (URLDepthPair c : viewedLink)

System.out.println("Depth : "+c.getDepth() + "\tLink : "+c.getURL());

}

public static void request(PrintWriter out,URLDepthPair pair) throws MalformedURLException {

out.println("GET " + pair.getPath() + " HTTP/1.1");

out.println("Host: " + pair.getHost());

out.println("Connection: close");

out.println();

out.flush();

}

public static void Process(String pair, int maxDepth) throws IOException {

findLink.add(new URLDepthPair(pair, 0));

while (!findLink.isEmpty()) {

URLDepthPair currentPair = findLink.removeFirst();

if (currentPair.depth<maxDepth){

Socket my\_socket = new Socket(currentPair.getHost(), 80);

my\_socket.setSoTimeout(100000);

BufferedReader in = new BufferedReader(new InputStreamReader(my\_socket.getInputStream()));

PrintWriter out = new PrintWriter(my\_socket.getOutputStream(), true);

request(out, currentPair);

String line;

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

if (line.indexOf(currentPair.URL\_PREFIX)!=-1) {

StringBuilder currentLink = new StringBuilder();

for (int i = line.indexOf(currentPair.URL\_PREFIX) + 9; line.charAt(i) != '"'; i++) {

currentLink.append(line.charAt(i));

}

URLDepthPair newPair = new URLDepthPair(currentLink.toString(), currentPair.depth + 1);

if (currentPair.check(findLink, newPair) && currentPair.check(viewedLink, newPair) && !currentPair.URL.equals(newPair.URL))

findLink.add(newPair);

}

}

my\_socket.close();

}

viewedLink.add(currentPair);

}

showResult(viewedLink);

}

public static void main(String[] args) {

String[] arg = new String[]{"http://mtuci.ru/","1"};

try {

Process(arg[0], Integer.parseInt(arg[1]));

} catch (NumberFormatException | IOException e) {

System.out.println("usage: java Crawler <URL><depth>");

}

}

}

1. **Файл URLDepthPair.java:**

import java.util.LinkedList;

import java.net.MalformedURLException;

import java.net.URL;

public class URLDepthPair {

public static final String URL\_PREFIX = "<a href=\"http";

public String URL;

public int depth;

public URLDepthPair (String URL, int depth){

this.URL=URL;

this.depth=depth;

}

public String getHost() throws MalformedURLException {

URL host = new URL(URL);

return host.getHost();

}

public String getPath() throws MalformedURLException {

URL path = new URL(URL);

return path.getPath();

}

public int getDepth() {

return depth;

}

public String getURL() {

return URL;

}

public static boolean check(LinkedList<URLDepthPair> resultLink, URLDepthPair pair) {

boolean isAlready = true;

for (URLDepthPair c : resultLink)

if (c.getURL().equals(pair.getURL()))

isAlready=false;

return isAlready;

}

}

**Пример работы программы:**

