Федеральное агентство связи

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра Информатики

Отчет по лабораторной работе №7

по предмету «КТП»:

Выполнил: студент группы БВТ1802

Самаков Владислав Владимирович

Руководитель:

Ксения Андреевна Полянцева

Москва 2020

1 Цель работы

Цель работы: изучить работу простейшего веб-сканера.

2 Задание

Написать программу, которая будет получать в аргументах командной строки URL-адрес и глубину поиска и посещать все ссылки, которые находятся на исходной web-странице в пределах указанной глубины поиска.

3 Текст программы

ScannerApp.java

```
import java.io.IOException;

public class ScannerApp {
    public static void main(String args[]) throws IOException {
        // first program arguments is a link, second argument is depth of the search
        Crawler crawler = new Crawler(args[0], Integer.parseInt(args[1]));
        crawler.Scan();
        System.out.println("Depth: " + Integer.parseInt(args[1]));
        crawler.getSites();
    }
}
```

Crawler.java

```
import java.io.*;
import java.net.*;
import java.util.LinkedList;


public class Crawler {

    // just alias for depth which is ignored in URLDepthPair.equals()
    final static int AnyDepth = 0;


    private LinkedList<URLDepthPair> visited = new LinkedList<URLDepthPair>();
    private LinkedList<URLDepthPair> notVisited = new LinkedList<URLDepthPair>();

    private int depth;
    private String startHost;
    // prefix has no slash to support https too
    private String prefix = "http";

    public Crawler(String host, int depth) {
        startHost = host;
        this.depth = depth;
        notVisited.add(new URLDepthPair(startHost, this.depth));
    }

    public void Scan() throws IOException {

       while (notVisited.size() > 0) {
```

```
            Process(notVisited.removeFirst());
            }
        }

    public void getSites() {
        // printing the links
        for (URLDepthPair elem : visited)
            System.out.println(elem.getURL());
        System.out.println("Links visited: " + visited.size());
    }

    public void Process(URLDepthPair pair) throws IOException{
        // set up a connection and follow the redirect
        URL url = new URL(pair.getURL());
        URLConnection connection = url.openConnection();
        String redirect = connection.getHeaderField("Location");
        if (redirect != null) {
            connection = new URL(redirect).openConnection();
        }
        visited.add(pair);
        if (pair.getDepth() == 0) return;

        // reading references
        BufferedReader reader = new BufferedReader(new
InputStreamReader(connection.getInputStream()));
        String input;
        while ((input = reader.readLine()) != null) {
            while (input.contains("a href=\"" + prefix)) {
                input = input.substring(input.indexOf("a href=\"" + prefix) + 8);
                String link = input.substring(0, input.indexOf('\"'));
                if(link.contains(" "))
                    link = link.replace(" ", "%20");
                // avoid multiple visiting of the same link
                if (notVisited.contains(new URLDepthPair(link, AnyDepth)) ||
                        visited.contains(new URLDepthPair(link, AnyDepth))) continue;
                notVisited.add(new URLDepthPair(link, pair.getDepth() - 1));
            }
        }
        // close the connection
        reader.close();

    }




}
```

URLDepthPair.java

```
import java.util.Objects;

public class URLDepthPair {

    private String url;
    private int depth;

    public URLDepthPair(String host, int depth) {
        url = host;
        this.depth = depth;
    }
```

```
    public String getURL() {
        return url;
    }

    public int getDepth() {
        return depth;
    }

    @Override
    public boolean equals(Object obj) {
        if (obj instanceof URLDepthPair) {
            URLDepthPair o = (URLDepthPair)obj;
            return this.url.equals(o.getURL());
        }
        return false;
    }

    @Override
    public int hashCode() {
        return Objects.hash();
    }
}
```

4 Работа программы

Стартовый url - https://www.youtube.com/ Глубина поиска - 2

```
"C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2019.3.2\jbr\bin\java.exe" "-javaagent:C:
Depth: 2
https://www.youtube.com/
https://www.google.ru/intl/ru/policies/privacy/
https://accounts.google.com/ServiceLogin?hl=ru&amp;continue=https%3A%2F%2Fwww.youtube.com%2Fsignin%3F
https://policies.google.com/privacy?hl=ru
https://accounts.google.com/AccountChooser?hl=ru
https://www.google.com/intl/ru/about
https://accounts.google.com/TOS?loc=RU&hl=ru&privacy=true
https://accounts.google.com/TOS?loc=RU&hl=ru
http://www.google.com/support/accounts?hl=ru
Links visited: 9

Process finished with exit code 0
```

6: TODO 9: Version Control Terminal

up-to-date (a minute ago)

Вывод

Язык программирования java обладает удобными средствами для работы с web-страницами. С их помощью я сделал простой web-сканер, который посещает все ссылки на указанном сайте до достижения заданной глубины.