Федеральное агентство связи

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра математической кибернетики и информационных технологий

Отчет по лабораторной работе №7

по дисциплине «Технологии разработки программного обеспечения»

Выполнил: студент группы БВТ1801

Протасова Елена Сергеевна

Руководитель:

Мосева Марина Сергеевна

Москва 2020

Цель работы: необходимо реализовать элементарный веб-сканер. Сканер будет автоматически загружать веб-страницы из Интернета, искать новые ссылки на этих страницах и повторять их. Он будет просто искать новые URL-адреса (местоположения веб-страниц) на каждой странице, собирать их и выводить в конце работы программы.

Crawlers:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Crawlers {
    int depth;
    SocKet socKet;
    String URL;
    BufferedReader bufferedReader;
    LinkedList<URLDepthPair> list;
    PrintWriter printWriter;

    public Crawlers(String url, int port, int depth) throws IOException {
        socKet = new SocKet(url,port);
        bufferedReader = new BufferedReader(new
InputStreamReader(socKet.getInputStream()));
        printWriter = new PrintWriter(socKet.getOutputStream(), true);
        this.depth = depth;
        URL=url;
    }

    public LinkedList<URLDepthPair> StartScanner(int count) throws IOException{
        printWriter.println("GET / HTTP/1.1");
        //printWriter.println("Host: www.google.com:80");
        printWriter.println("Host:"+URL+":80");
        printWriter.println("Connection: Close");
        printWriter.println();
        LinkedList<URLDepthPair> resultlist = new LinkedList<>();
        ArrayList<String> newlinks = new ArrayList<>();
        String str;
        try {
            while ((str = bufferedReader.readLine()) != null) {
                while(str.contains("<a")){
                    while (str.indexOf(">", str.indexOf("<a"))==-1)
str+=bufferedReader.readLine();

                    String link = str.substring(str.indexOf("<a"),str.indexOf(">",
str.indexOf("<a")));
                    if (link.contains("http://")){
                        Pattern pattern= Pattern.compile("(http:\\/\\/[\\w\\-
\\.!~?&=+\\*'(),\\/\\#\\:]+)((?!\\<\\/\\w\\>))*?");
                        Matcher matcher = pattern.matcher(link);
                        matcher.find();
                        String url = matcher.group();
                        newlinks.add(url);
                        URLDepthPair urlDepthPair = new URLDepthPair(url, count);
```

```
                        resultlist.add(urlDepthPair);
                    }
                    str=str.replace(link, "");
                }
            }
        }
        catch (Exception e){
            System.out.println(e+"  OK");
            socKet.close();
            return resultlist;
        }
        socKet.close();
            return resultlist;
    }
    public LinkedList<URLDepthPair> getSites(){
            return list;
    }

    public static void main(String[] args) throws IOException {
        URLDepthPair urlDepthPair = new URLDepthPair(args[0], 0);
        if (urlDepthPair.Available() || args[1].matches("\\D+")) {
            args[0]=args[0].replace("http://", "");
            int depth = Integer.parseInt(args[1]);
            LinkedList<URLDepthPair> list;
            LinkedList<URLDepthPair> newlist = new LinkedList<>();
            LinkedList<URLDepthPair> result = new LinkedList<>();

            Crawlers crawler = new Crawlers(args[0], 80, depth);
            list = crawler.StartScanner(0);
            result.addAll(list);

            for (int i = 1; i < depth; i++) {
                while (list.size() > 0) {
                    crawler = new Crawlers(list.getFirst().getURL(), 80, depth);
                    list.removeFirst();
                    newlist.addAll(crawler.StartScanner(i));
                }
                list.addAll(newlist);
                result.addAll(list);
                newlist.clear();
            }
            for (int i = 0; i < result.size(); i++) {
                System.out.println(result.get(i));
            }

        }
        else System.out.println("usage: java Crawler <"+args[0]+"><"+args[1]+">");
    }
}
```

SocKet:

```
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net. *;
public class SocKet {
    Socket socket;

    public SocKet (String host, int port) throws IOException {
        socket = new Socket(host,port);
        socket.setSoTimeout(10000);
    }
    public void setSoTimeout(int timeout) throws IOException{
        socket.setSoTimeout(timeout);

    }
    public InputStream getInputStream() throws IOException{
       return  socket.getInputStream();
    }
    public OutputStream getOutputStream() throws IOException{
        return socket.getOutputStream();
    }
    public void close() throws IOException{
        socket.close();
    }

}
```

URLDepthPair:

```
import java.net.MalformedURLException;
import java.net.URL;

public class URLDepthPair {
    //private String url;
    String URL;
    int depth;
    public URLDepthPair(String URL, int depth){
        this.URL = URL;
        this.depth = depth;
    }
    public boolean Available(){
        String regex = "\\b(http)://[-a-zA-Z0-9+&@#/%?=~_|!:,.;]*[-a-zA-Z0-
9+&@#/%=~_|]";
        return URL.matches(regex);

    }
    @Override
    public String toString() {
        return "URL = "+URL+" Depth = "+depth;
    }
    public String getURL(){
        //URL = url.substring(0, url.indexOf("http"));
        try {
            java.net.URL url = new URL(this.URL);
            return url.getHost();
        }
        catch (MalformedURLException e) {
            System.err.println("MalformedURLException: " + e.getMessage());
            return null;
        }
        //return URL;
```

```
    }

    public int depth(){
        return depth;
    }
}
```

Результаты работы:

```
C:\Users\Elena\Desktop\все папки\2 курс\2семестр\КТП\GIT\КТР\КТП\лабораторные работы\7 лаб.раб\lab7КТР\src>java Crawlers http://www.google.com 2
URL = http://www.google.ru/imghp?hl=ru&tab=wi Depth = 0
URL = http://maps.google.ru/maps?hl=ru&tab=wl Depth = 0
URL = http://www.youtube.com/?gl=RU&tab=w1 Depth = 0
URL = http://news.google.ru/nwshp?hl=ru&tab=wn Depth = 0
URL = http://translate.google.ru/?hl=ru&tab=wT Depth = 0
URL = http://www.blogger.com/?tab=wj Depth = 0
URL = http://video.google.ru/?hl=ru&tab=wv Depth = 0
URL = http://www.google.com/ Depth = 0
URL = http://www.google.ru/preferences?hl=ru Depth = 0
URL = http://www.google.ru/history/optout?hl=ru Depth = 0
URL = http://www.google.ru/intl/ru/services/ Depth = 0
URL = http://www.google.com/setprefdomain?prefdom=RU&amp Depth = 0
URL = http://www.google.ru/imghp?hl=ru&tab=wi Depth = 1
URL = http://maps.google.ru/maps?hl=ru&tab=wl Depth = 1
URL = http://www.youtube.com/?gl=RU&tab=w1 Depth = 1
URL = http://news.google.ru/nwshp?hl=ru&tab=wn Depth = 1
URL = http://translate.google.ru/?hl=ru&tab=wT Depth = 1
URL = http://www.blogger.com/?tab=wj Depth = 1
URL = http://video.google.ru/?hl=ru&tab=wv Depth = 1
```

Выводы: была изучена работа с Socket, LinkedList, реализован элементарный веб-сканер.