**实验报告**

姓名： 林炬乙 学号： 3180103721

课程名称： Java应用技术 任课老师： 楼学庆

实验名称： Java爬虫 实验日期： 2020/10/13

**1 实验目的和要求**

**1.1 实验目的**

（分点简要说明本次实验需要进行的工作和最终的目的）

跟踪特定网页，下载该网页中所有链接的指定内容，去除广告等无关内容，组合成单一文件。主要作广度搜索，深度暂为1。

(基本)不使用第三方工具，自己作String处理。

(提高)可用第三方工具，如：HttpClient、HtmlParser等。

**2 实验原理**

爬取了<http://www.yuedu88.com/longzu1/> <龙族一>的链接, 下载该网页中所有链接的指定内容，去除广告等无关内容，组合成单一文件。主要作广度搜索，深度暂为1.

Main 方法: 给定url,使用时修改url,即可爬取不同网站的小说.调用*spiderURL方法获得每一章的url, searchThroughUrlFile方法搜索并处理.*

spiderURL方法:爬取该页的所有html写入Spider\_content文件, 正则提取目录链接到Spider\_URL文件.

searchThroughUrlFile方法:

然后从Spider\_URL文件读取url, 每一行读入tempStr,如果有http://www.yuedu88.com/longzu1/的前缀,那么很可能就是需要的url, 那就访问它爬取html,同时三个写入:

①直接写入Spider\_all文件,

②string处理后写入Spider\_simple文件,

③调用*readTextAndLinkAndTitle方法写入Spider\_ByParaser文件*

如果不是,那就下一行.

每一行都读完了,那就关闭其他文件,关闭网络.

readTextAndLinkAndTitle 方法:利用htmlparser来解析html文本,然后写入给定的文件中.

**3 实验内容**

（分点阐述实验步骤）

遇到的错误:

问题1.输入文件的位置作为字符串的时候,遇到Invalid escape sequence (valid ones are  \b  \t  \n  \f  \r  \"  \'  \\ ) 报错

一个正则表达式错误

原因：Java源代码的字符串中的反斜线被解释为Unicode 转义或其他字符转义

解决办法： 在“\”后加（\b   \t   \n   \f   \r   \） 其中的一个就可以了。当然也可以直接把语句中的“\”去掉。

问题2. 东西不对,怎么只有这么一点内容呢?

解决方法:仔细查看具体内容,最后发现,应该是被覆盖掉了, 写入的是(尾声). 应该在后面接着写入而不是覆盖.

追加,是在filewriter中用true参数

我先试试把 pw pw2 放在前面,不要每次都new一个.

但是出现了错误

问题3.Unhandled exception type IOException

这是ioe异常,在read.line方法前加try,catch语句。或者在方法上抛出异常类

解决了之后

还是只有尾声

问题4. 尝试调整了 pw = **new** PrintWriter

reader = **new** BufferedReader

pw.close();

pw2.close();的位置

终于可以全部爬取下来html格式.

接下来还需要选择出文本.Pw2写入简化后的文本.

问题5. 前面有大量空白,中间有大量重复

解决方法: pw2.println(str);//写入简化后的

把这句放在**while**( ( line = br.readLine() ) != **null** ){ 循环外

问题6. 一行里面有多段html包裹的文本

**while**(i >= 0) {

j = line.indexOf("</span>",i);

str += line.substring(i+"<span class=\"calibre8\">".length() , j);

i = line.indexOf("<span class=\"calibre",j);

}

利用循环来提取.

至此, 基础模块基本完成.

提高模块

使用HTMLParser

要使用该库，在编译和运行时，您需要将htmllexer.jar或htmlparser.jar添加到您的类路径中。htmllexer.jar以线性，平坦，顺序的方式提供对页面上的通用字符串，备注和标记节点的低级访问。htmlparser.jar（包含在htmllexer.jar中找到的类）提供对页面的访问

<https://my.oschina.net/u/175660/blog/85205>

对于大多数使用者来说，使用最多的是通过一个URLConnection或者一个保存有网页内容的字符串来初始化Parser，或者使用静态函数来生成一个Parser对象。

问题1怎么打开一个txt然后解析?

问题2忽然br.close();出错了,不知道为何.

java.lang.NullPointerException

解决方法:把close()放在try语句里面,

问题3

然后报错另一个java.io.FileNotFoundException:

应该是因为没有新建目录

解决了之后, java.lang.NullPointerException还是存在

应该是因为每个try语句里不通?

BufferedReader br = **null**;放在try里面

问题4Unreachable catch block for IOException. This exception is never thrown from the try statement body不可到达的 catch 块异常。try 中的语句，永远不会引发此异常

问题5NoClassDefFoundError发生在JVM在动态运行时，根据你提供的类名，在classpath中找到对应的类进行加载，但当它找不到这个类时，就发生了

解决方法:

1.存放第三方jar包的目录名改为libs之后，重新设置build path.这样还是出现错误.

2.我把所有的jar都包含进来,好像是htmllexer的问题,我去掉这个包就出现了NoClassDefFoundError错误.

问题6显示错误

Error occurred during initialization of boot layer

java.lang.module.FindException: Unable to derive module descriptor for C:\Users\12638\Downloads\htmlparserv16\htmlparser1\_6\libs\sax2.jar

Caused by: java.lang.module.InvalidModuleDescriptorException: SAXDump.class found in top-level directory (unnamed package not allowed in module)

解决方法:

JDK版本：10.0.2

方法1： 新建一个包，将默认包中的类文件拖入新建包中，刷新，默认包会消失

方法2： 将module-info.java删除

但是我没有看到module-info,可能因为我一开始就没有用模块,

我就把sax2.jar删除了

问题7:出现错误:Exception in thread "main" java.lang.Error: Unresolved compilation problem

把其他的都注释掉,应该是这里报错:

The package org.htmlparser is accessible from more than one module: filterbuilder, htmllexer, htmlparser, Thumbelina

原因:

在添加Jar包时选择的是Modulepath进行添加。更改为Classpath后即可解决。

问题8:字体忽然变了

解决方法:这个不是你的*Eclipse*的问题 是你的输入法被你切换为全角模式输入后的情况 只需要把输入法切回半角模式即可.

问题9:Html解析出来的也没有追加,也只有尾声.

解决方法:

PrintWriter构造函数的第二个参数是表示是否自动刷新缓冲区，FileWriter第二个才是才是追加写文件．加上true就可以了.

问题10 :怎么打包有外部jar 的java.

<https://www.cnblogs.com/lanxuezaipiao/p/3291641.html>

还是不太会,

cmd进入项目目录输入以下命令

//执行以下命令后会在当前目录生成一个warname.war

jar -cvf warname.war ./\*

//有一个弊端就是把整个项目所有的文件都弄到了war里面了，体积巨大。

//解压war包

jar xvf warname.war

我用这个方法了,但是还是不知道怎么打开project.

我的发现:

**new** BufferedReader(**new** InputStreamReader(connection.getInputStream())

可以写在一起,就不用给**new** InputStreamReader 命名然后再调用了.

把reader的URL的名字改一下,不要固定名字,

"C:\\Users\\12638\\Desktop\\javalearning\\homework\\9\_27homework2\\txtdir/"+time+"/"+filename+"\_URL.txt"

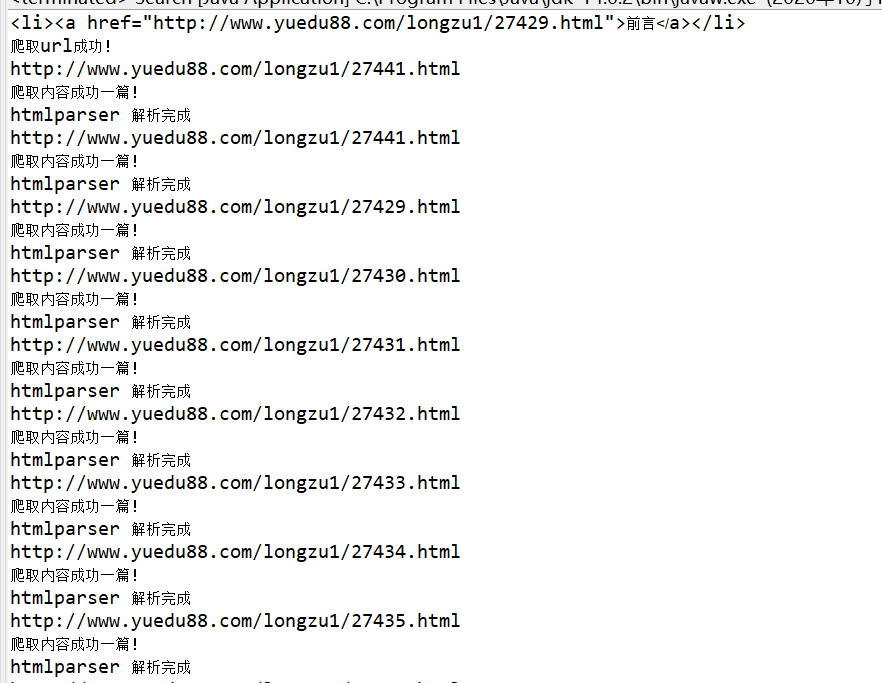
然后new FileWriter("HTMLparaser/"+time+"/"+filename+"\_ByParaser.txt”, true),true);

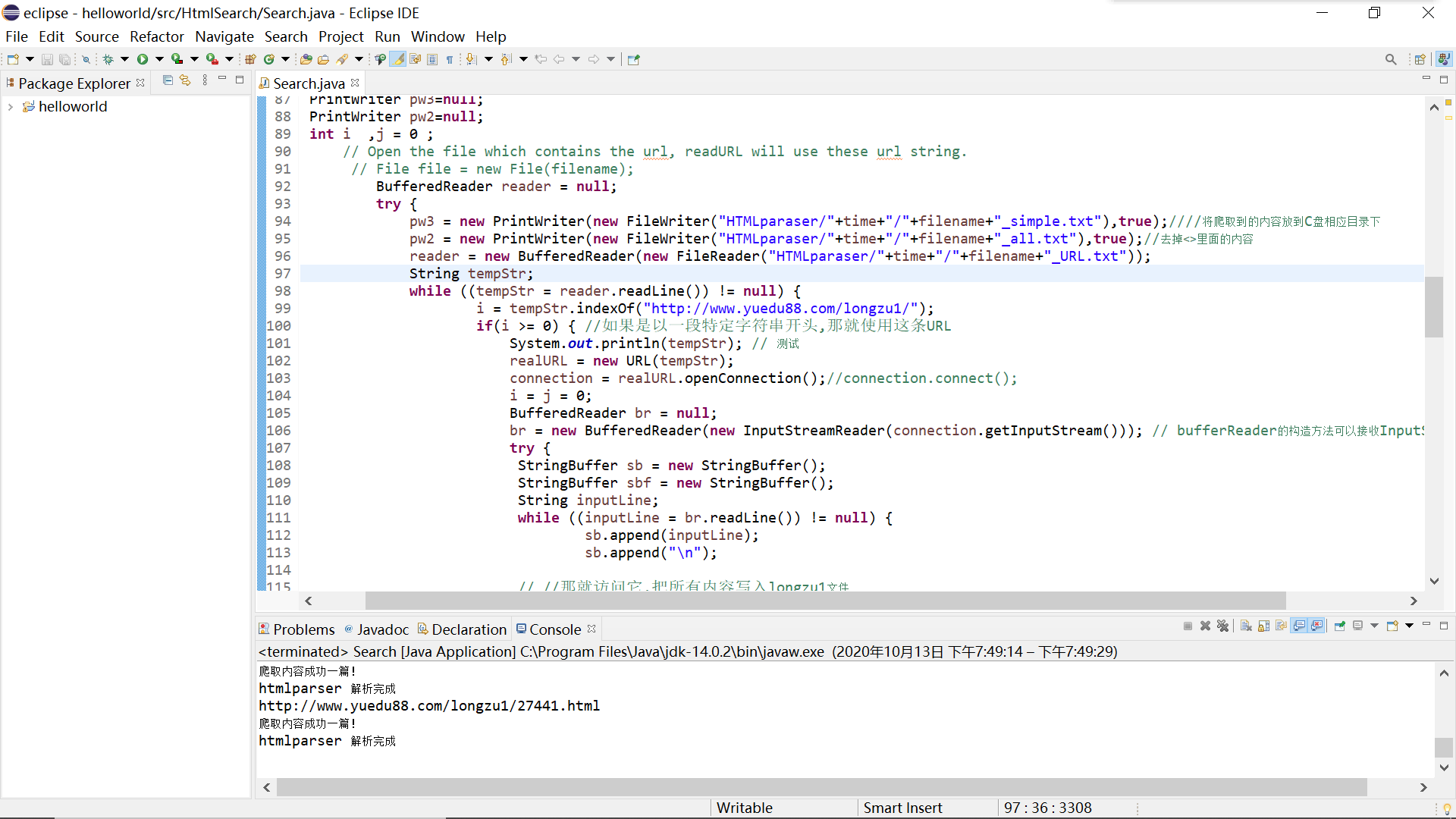
这样就可以在当前目录下创建文件夹然后输出,就可以做到不同电脑通用

Strbuffer可以append(“\n”)就可以换行了.视觉效果很好.

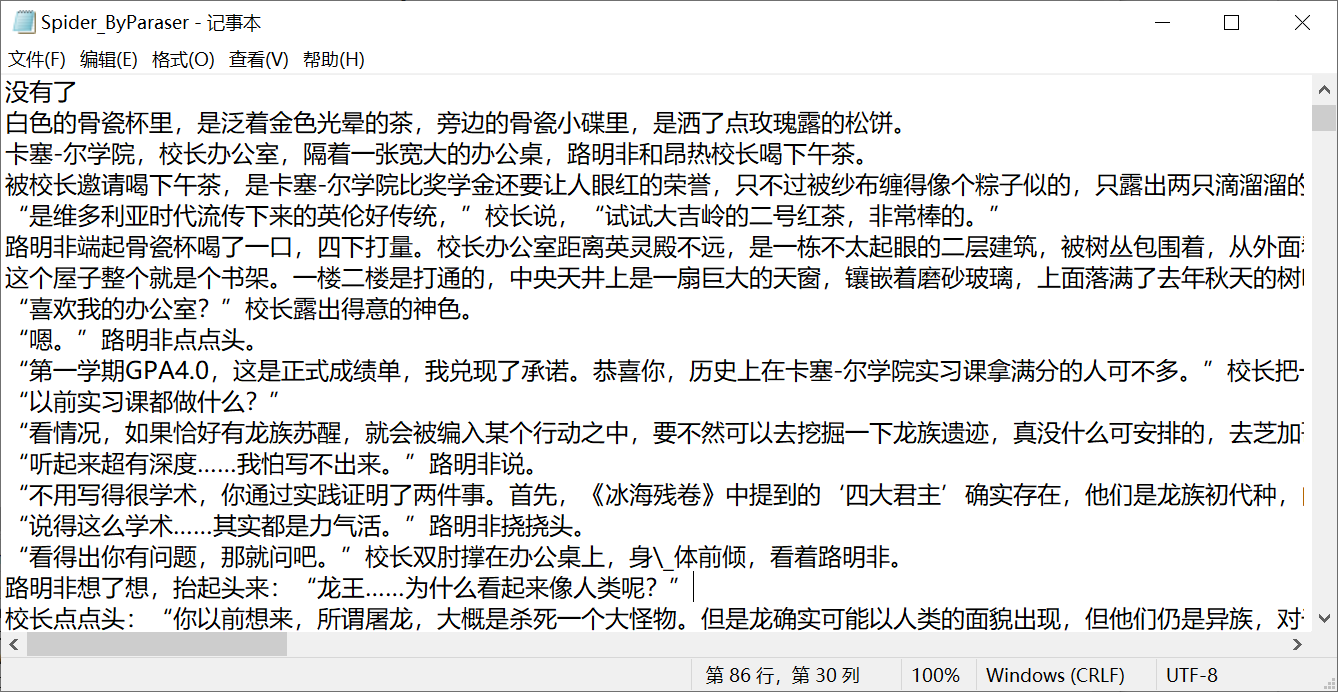
**4 实验结果和分析**

（使用图片和文字叙述实验结果，并对这些结果进行适当分析）

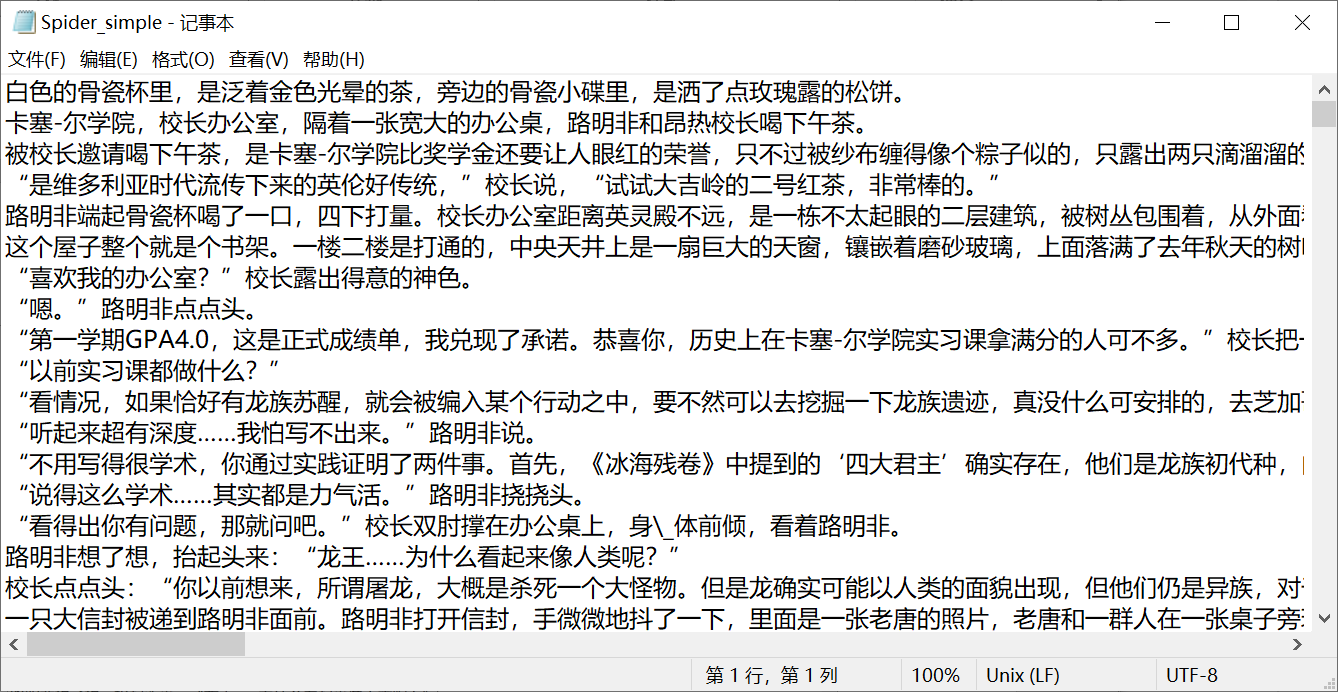




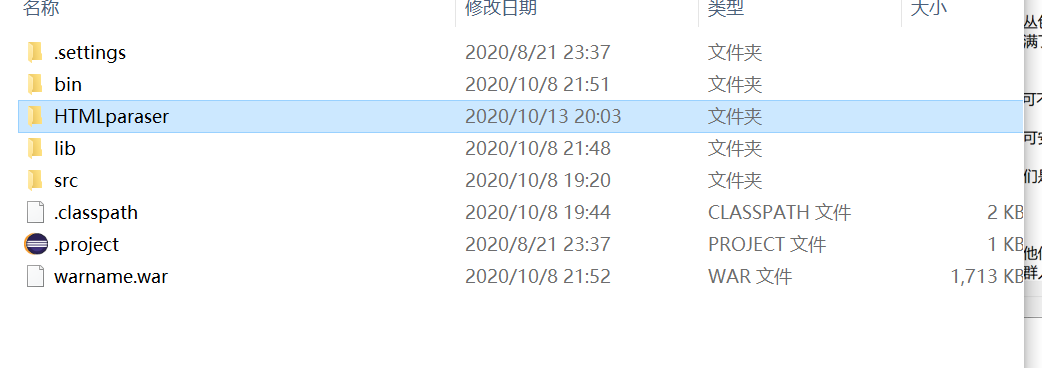
可以在console显示 爬取成功和解析完成.



Htmlparser处理结果



String处理结果



HTMLparaser文件下保存产生的五个文件.

**5 源代码与分析**

（粘贴本次实验使用的源代码，并使用注释的方式进行适当的分析）

package HtmlSearch;

import java.io.\*;

import java.net.\*;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import org.htmlparser.filters.\*;

import org.htmlparser.\*;

import org.htmlparser.nodes.\*;

import org.htmlparser.tags.\*;

import org.htmlparser.util.\*;

import org.htmlparser.visitors.\*;

/\*\*

\* java实现爬虫跟踪特定网页，下载该网页中所有链接的指定内容，去除广告等无关内容，组合成单一文件。主要作广度搜索，深度暂为1。

实现：

1、(基本)不使用第三方工具，自己作String处理。

2、(提高)可用第三方工具，如：HttpClient、HtmlParser等。

\*/

public class Search {

//爬取该页的所有html写入Spider\_content文件, 正则提取目录链接到Spider\_URL文件.

public static void spiderURL(String url,String regex,String filename){

SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd");

String time = sdf.format(new Date());

URL realURL = null;

URLConnection connection = null;

BufferedReader br=null;

PrintWriter pw=null;

PrintWriter pw1=null;

int i ;

Pattern pattern=Pattern.compile(regex);//正则

try{

realURL = new URL(url);

connection = realURL.openConnection();

File fileDir = new File("HTMLparaser/"+time);

if (!fileDir.exists()) {

fileDir.mkdirs();

}

//将爬取到的内容放到当前目录下

pw = new PrintWriter(new FileWriter("HTMLparaser/"+time+"/"+filename+"\_content.txt"),true);

pw1 = new PrintWriter(new FileWriter("HTMLparaser/"+time+"/"+filename+"\_URL.txt"),true);

i = 0;

br = new BufferedReader(new InputStreamReader(connection.getInputStream())); // bufferReader的构造方法可以接收InputStreamReader作为参数

String line = null;

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

i = line.indexOf("<li><a href=\"http://www.yuedu88.com/longzu1/27429.html\">前言</a></li>");

if(i >= 0) {

System.out.println(line.substring(i));

}

pw.println(line);//写入全部内容

Matcher matcher=pattern.matcher(line);

while(matcher.find()){

pw1.println(matcher.group()); //写入url

}

}

System.out.println("爬取url成功！");

}catch(Exception e){

e.printStackTrace();

}finally{

try {

br.close();

pw.close();

pw1.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

//打开存储有url的文件,每一行读入tempStr,如果有http://www.yuedu88.com/longzu1/的前缀,那么很可能就是需要的url, 那就访问它,把所有内容写入longzu1文件

//如果不是,那就下一行.

//每一行都读完了,那就关闭其他文件,关闭网络,处理longzu1

public static void searchThroughUrlFile(String regex,String filename) throws Exception {

// open the file which contains the url , and open these url one by one ,spider the text.

SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd");

String time = sdf.format(new Date());

URL realURL = null;

URLConnection connection = null;

PrintWriter pw3=null;

PrintWriter pw2=null;

int i ,j = 0 ;

// Open the file which contains the url, readURL will use these url string.

// File file = new File(filename);

BufferedReader reader = null;

try {

pw3 = new PrintWriter(new FileWriter("HTMLparaser/"+time+"/"+filename+"\_simple.txt"),true);////将爬取到的内容放到C盘相应目录下

pw2 = new PrintWriter(new FileWriter("HTMLparaser/"+time+"/"+filename+"\_all.txt"),true);//去掉<>里面的内容

reader = new BufferedReader(new FileReader("HTMLparaser/"+time+"/"+filename+"\_URL.txt"));

String tempStr;

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

i = tempStr.indexOf("http://www.yuedu88.com/longzu1/");

if(i >= 0) { //如果是以一段特定字符串开头,那就使用这条URL

System.out.println(tempStr); // 测试

realURL = new URL(tempStr);

connection = realURL.openConnection();//connection.connect();

i = j = 0;

BufferedReader br = null;

br = new BufferedReader(new InputStreamReader(connection.getInputStream())); // bufferReader的构造方法可以接收InputStreamReader作为参数

try {

StringBuffer sb = new StringBuffer();

StringBuffer sbf = new StringBuffer();

String inputLine;

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

sb.append(inputLine);

sb.append("\n");

// //那就访问它,把所有内容写入longzu1文件

// my String process

i = inputLine.indexOf("<span class=\"calibre8\">");

while(i >= 0) {

j = inputLine.indexOf("</span>",i);

sbf.append(inputLine.substring(i+"<span class=\"calibre8\">".length() , j) );

sbf.append("\n");

i = inputLine.indexOf("<span class=\"calibre",j);

}

pw2.println(inputLine);//写入全部内容

}

pw3.println(sbf);//写入简化后的文本

System.out.println("爬取内容成功一篇！");

// use htmlparaser to process

String result = sb.toString();

readTextAndLinkAndTitle(result,"Spider");

//use htmlparaser to process finished

br.close();

}catch(Exception e){

e.printStackTrace();

}

}

//这条URL使用完毕，下一条URL．

}

reader.close();

}catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}finally{ //最后关闭

pw3.close();

pw2.close();

}

}

/\* 利用htmlparser来解析html文本,然后写入给定的文件中\*/

public static void readTextAndLinkAndTitle(String result,String filename) throws Exception {

SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd");

String time = sdf.format(new Date());

Parser parser;

NodeList nodelist;

parser = Parser.createParser(result, "utf8");

NodeFilter textFilter = new NodeClassFilter(TextNode.class);

NodeFilter linkFilter = new NodeClassFilter(LinkTag.class);

NodeFilter titleFilter = new NodeClassFilter(TitleTag.class);

OrFilter lastFilter = new OrFilter();

lastFilter.setPredicates(new NodeFilter[] { textFilter, linkFilter, titleFilter });

nodelist = parser.parse(lastFilter);

Node[] nodes = nodelist.toNodeArray();

String line = "";

// Print in the file

PrintWriter pw1 = null;

pw1 = new PrintWriter(new FileWriter("HTMLparaser/"+time+"/"+filename+"\_ByParaser.txt",true),true);////将爬取到的内容放到C盘相应目录下

for (int i = 0; i < nodes.length; i++) {

Node node = nodes[i];

if (node instanceof TextNode) {

TextNode textnode = (TextNode) node;

line = textnode.getText();

}

/\*

else if (node instanceof LinkTag) {

LinkTag link = (LinkTag) node;

line = link.getLink();

} else if (node instanceof TitleTag) {

TitleTag titlenode = (TitleTag) node;

line = titlenode.getTitle();

}

\*/

if (isTrimEmpty(line))

continue;

pw1.println(line);//写入简化后的文本

}

pw1.close();

System.out.println("htmlparser 解析完成");

}

/\* 去掉左右空格后字符串是否为空

\*\*/

public static boolean isTrimEmpty(String astr) {

if ((null == astr) || (astr.length() == 0)) {

return true;

}

if (isBlank(astr.trim())) {

return true;

}

return false;

}

/\*\* 字符串是否为空:null或者长度为0.

\*/

public static boolean isBlank(String astr) {

if ((null == astr) || (astr.length() == 0)) {

return true;

} else {

return false;

}

}

public static void main(String[] args) throws Exception {

String url="http://www.yuedu88.com/longzu1/";

String regex= "(http|https)://[\\w+\\.?/?]+\\.[A-Za-z]+"; //http或https开头,一个词,

spiderURL(url,regex,"Spider");

searchThroughUrlFile(regex,"Spider");

}

}