武汉纺织大学

Java应用开发课程设计

**基于Maven的java爬虫**

**学 院： 数学与计算机学院**

**班 级： 软工11802**

**姓 名： 李文爽**

**学 号： 1804240316**

**指导老师：**

**成 绩：**

**完成日期： 2020年6月20日**

目 录

[1 需求分析 1](#_Toc43473558)

[1.1各项功能 1](#_Toc43473559)

[1.1.1 收银 1](#_Toc43473560)

[运行程序后输入要爬取的网站与爬页数 1](#_Toc43473561)

[1.1.2 查询统计 1](#_Toc43473562)

[2 系统设计 1](#_Toc43473563)

[2.3 UML类图（Class Diagram） 1](#_Toc43473564)

[3 系统实现 3](#_Toc43473565)

[3.1 项目结构 3](#_Toc43473566)

[3.2 配置文件 3](#_Toc43473567)

[3.2.1 jdbc.properties文件 3](#_Toc43473568)

[3.2.2 log4j.properties文件 3](#_Toc43473569)

[3.2.3 mybatis-config.xml文件 4](#_Toc43473570)

[3.3 VO类User.java 4](#_Toc43473571)

[3.4 DAO接口类IUserDAO.java 5](#_Toc43473572)

[3.5 接口映射文件UserMapper.xml 5](#_Toc43473573)

[3.6 工具包Util 5](#_Toc43473574)

[3.6.1 MybatisUtils.java 5](#_Toc43473575)

[3.6.2 MD5Util.java 6](#_Toc43473576)

[3.7 服务层UserService.java 7](#_Toc43473577)

[3.8 用户界面Driver.java 8](#_Toc43473578)

[3.9 \*\*\*\*\*\*.java 8](#_Toc43473579)

[4 系统测试 8](#_Toc43473580)

[5 系统总结 8](#_Toc43473581)

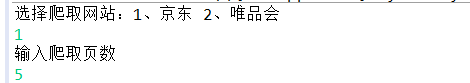
# 1 需求分析

任意选取两个购物网站（如淘宝，京东，国美、唯品会、品多多等），爬取手机搜索页面的信息，从返回页面中利用正则表达式（或Jsoup库）获取各手机相关信息（名称，价格，品牌、型号、评论数、店铺、图片等），保存到MySQL数据库中，形成一个可用于实际分析的数据表格。开发一个简单的查询程序，可根据多种条件（如手机名称、价格区间、型号等）查询显示手机信息。

## 1.1各项功能

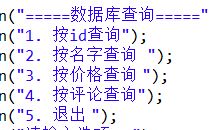
### 1.1.1 菜单

### 运行程序后输入要爬取的网站与爬页数



### 1.1.2 查询统计

爬取完成后出现数据库查询菜单，



选择1后提示输入要查询的产品id号，而后控制台给出结果，

选择2后提示输入要查询的产品名字，而后控制台给出结果，

选择3后提示输入要查询的产品价格范围，而后控制台给出结果，

选择4后提示输入要查询的产品评论数范围，而后控制台给出结果，

# 2 系统设计

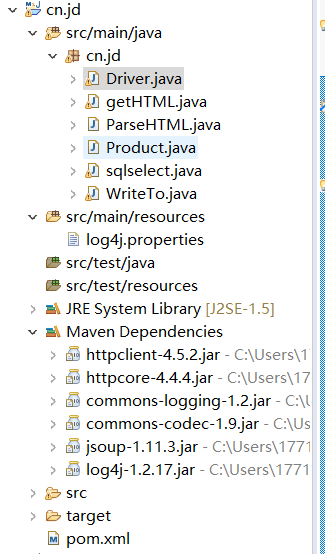
## 2.1 UML类图（Class Diagram）

对于用户功能模块，共设计如下6个类。

* Driver：实现菜单
* 实体类Product：与数据库结构进行映射的类。主要由setter, getter方法组成
* sqlselect：主要定义操作的接口，定义一系列数据库的原子性操作
* parseHtml：用于分析Html，得到所需要的各项数据，并添加到对象中
* getHtml：用于url的生成并用HttpClient抓取html内容
* writeto：用于把对象写入数据库

# 3 系统实现

## 3.1 项目结构



## 3.2 配置文件

### 3.2.1pom.xml文件

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.jd</groupId>

<artifactId>cn.jd</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/httpclient -->

<dependency>

<groupId>org.apache.httpcomponents</groupId>

<artifactId>httpclient</artifactId>

<version>4.5.2</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.jsoup/jsoup -->

<dependency>

<groupId>org.jsoup</groupId>

<artifactId>jsoup</artifactId>

<version>1.11.3</version>

</dependency>

<!-- https://mvnrepository.com/artifact/log4j/log4j -->

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.17</version>

</dependency>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.15</version>

</dependency>

</dependencies>

### </project>

### 3.2.2 log4j.properties文件

log4j.rootLogger=DEBUG, stdout

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=%5p [%t] - %m%n

</configuration>

## 3.3Product.java

**public** **class** User {

**private** String id;

**private** String name;

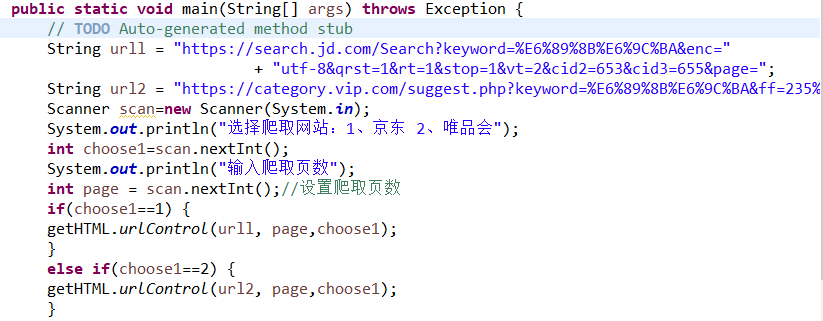
**private** String price;

**private** String comment;

//构造方法、getter,setter，toString方法,自行补充

}

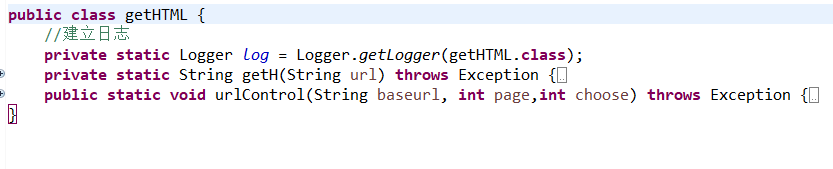
## 3.4 用户界面Driver.java



爬取初始选项，输入后调用urlControl（）；完成数据库插入后，显示 数据库查询菜单



## 3.5 getHTML.java



**private** **static** String getH(String url) **throws** Exception {

控制台输出日志，这样每条访问的URL都可以在控制台上看到访问情况

*log*.info("正在解析" + url);

HttpClient建立连接

CloseableHttpClient client = HttpClients.*createDefault*();

HttpGet get = **new** HttpGet(url);

CloseableHttpResponse response = client.execute(get);

将HTML内容转化为String

HttpEntity entity = response.getEntity();

String content;

**if** (entity != **null**) {

content = EntityUtils.*toString*(entity, "utf-8");

client.close();

response.close();

**return** content;

} **else**

**return** **null**;

}

urlControl(String baseurl, **int** page,**int** choose) {

实际访问的URL为不变的URL值拼接上URL变化的值

String u = baseurl + (2 \* count - 1) + "&click=1";

调用ParseHTML类中的方法对URL中的HTML页面进行处理

product = ParseHTML.*parse1*(*getH*(u));

调用WriteTo类中的方法对解析出来的内容写入到对象

WriteTo.*writeto*(product);

}

## 3.6 ParseHTML.java

parse1(String content)

{}

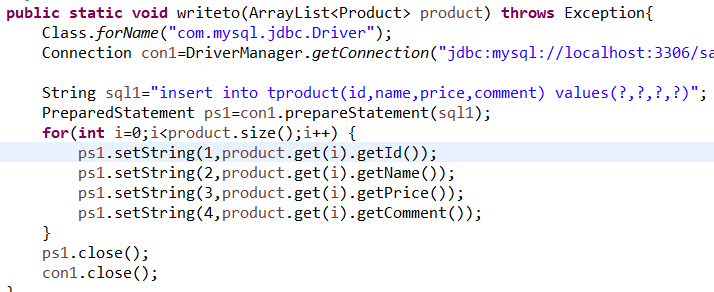
对京东html页面各个属性的 获取

Parse2(String content)

{}

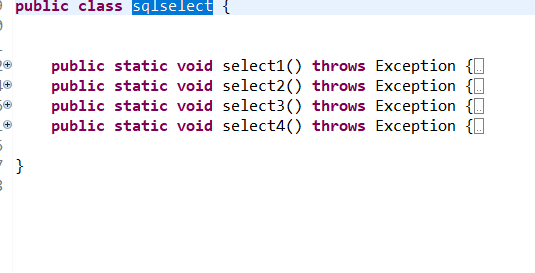
对唯品会html页面各个属性的 获取

## 3.7WriteTo.java



数据库的连接和写入

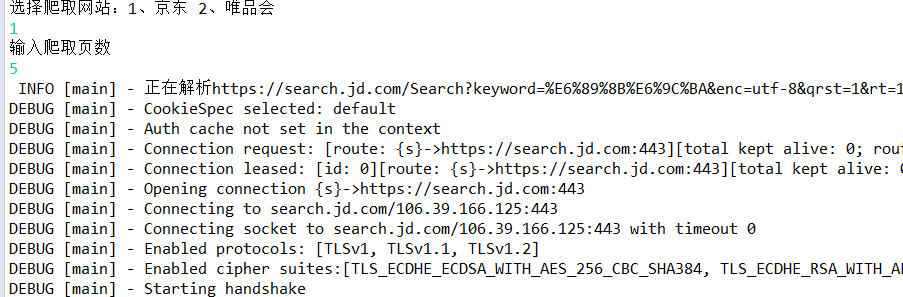
## 3.8sqlselect.java



数据库查询子语句的实现

# 4 系统测试

设计测试用例，给出程序每个功能模块的运行结果截图。



# 5 系统总结

在设计程序的时候，遇到的问题首先是框架的选择，对dao和mybite没有很好的掌握，maven的配置也存在 一些问题，在网上找了很多资料，解决了maven无法引入包的问题。然后就是京东 ，唯品会解析 html，获取指定属性要用到的标签，在网上查了很多，但是都不一样，而且自身 去查看网页源码是也没能 找到全部有用的具体标签。整个程序 写完后，运行出现了很多错误 ，去学习了引入日志 ，来查看问题所在，但在网上搜索问题，得到的信息非常杂乱。最后这次作业完成的不是很好，此次java课程中进一步加强了自己的编程能力，但还是存在一些问题，例如对一些框架的理解不深。这次课程给自己打60分。