武汉纺织大学

Java应用开发课程设计

**基于MyBatis的爬虫系统**

**学 院： 数学与计算机学院**

**班 级： 软件11801**

**姓 名： 陶佳乐**

**学 号： 1804240716**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年6月16日**

目 录

[1 需求分析 1](#_Toc2965)

[1.2主菜单 1](#_Toc27274)

[1.3各项功能 1](#_Toc17655)

[1.2.1爬取网页中的手机的信息: 1](#_Toc21322)

[1.2.2 将从网页爬取的数据存入数据库 1](#_Toc30258)

[1.2.3 输出数据库中所有手机的信息 2](#_Toc28610)

[1.2.4 按照价格区间从数据库中输出手机信息 2](#_Toc14439)

[1.2.5 按照手机品牌输出手机的信息 3](#_Toc27115)

[1.2.6 按照手机的型号输出手机的信息 3](#_Toc28851)

[1.2.7 退出程序 3](#_Toc4429)

[2 系统设计 3](#_Toc15323)

[2.1用户用例图 3](#_Toc21899)

[2.2 ER图 4](#_Toc16671)

[2.3 UML类图（Class Diagram） 4](#_Toc1640)

[2.4 UML时序图（Sequence Diagram） 6](#_Toc18606)

[2.5 UML活动图（Activity Diagram） 6](#_Toc5309)

[3 系统实现 9](#_Toc27433)

[3.1 项目结构 9](#_Toc1061)

[3.2 配置文件 9](#_Toc2679)

[3.2.1 jdbc.properties文件 9](#_Toc19162)

[3.2.2 log4j.properties文件 9](#_Toc31259)

[3.2.3 mybatis-config.xml文件 10](#_Toc7117)

[3.3 VO类Iphone.java 10](#_Toc26179)

[3.4 DAO接口类IPhoneDAO.java 11](#_Toc32366)

[3.5 接口映射文件IphoneMapper.xml 11](#_Toc31927)

[3.6 工具包Util 11](#_Toc25901)

[3.6.1 MybatisUtils.java 11](#_Toc6856)

[3.6.2 MD5Util.java 12](#_Toc8757)

[3.7 服务层JsoupIphone.java 13](#_Toc29606)

[3.8 用户界面Driver.java 14](#_Toc26330)

[3.9 \*\*\*\*\*\*.java 14](#_Toc14997)

[4 系统测试 14](#_Toc19341)

[5 系统总结 14](#_Toc30615)

# 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。

退出系统；

## 1.2主菜单

当用户名及密码正确时，进入系统主界面，如下：

1.请输入你想爬虫的手机页面URL地址,将爬取网页:

2.将爬虫得到手机数据存入数据库

3.输出数据库中所有的手机信息

4.输出一个区间手机的价格

5.输出一个手机品牌的手机

6.输出指定型号的手机

7.退出程序

请输入你的选择:如果输入1-6以外的选择，则显示提示信息“输入无效，只能输入1-6”后，返回主界面。

## 1.2各项功能

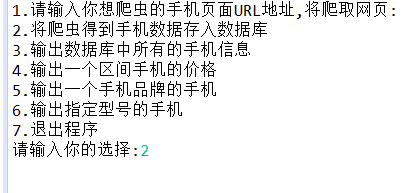
### 1.2.1 爬取网页中的手机信息

进入天猫的官网，输入华为手机，然后在地址栏，得到url地址，将这个地址复制到

对话框中

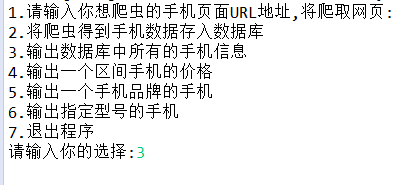


### 1.2.2 将从网页中爬取的信息存入数据库中



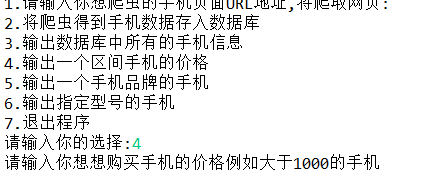
直接在对话框里面敲2就可以把刚才从数据库中的信息存入数据库

### 1.3.3 输出数据库中所有的手机信息



直接在对话框里面敲3就可以把数据库中所有的手机信息都输出

### 1.3.4 按照价格区间输出手机信息

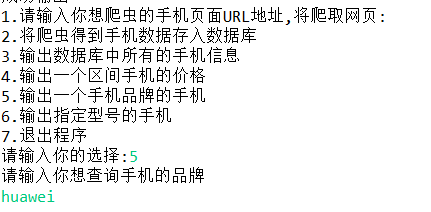


当选择4时，然后我们输出我们想要的手机最低的价格，页面将会输出

我们输入这个价格之上所有手机的信息

### 1.3.5 按照手机的品牌输出手机

当选择5时，显示如下菜单：



我们会根据我们输入的手机品牌在数据库中查询到所有该品牌的手机

### 1.3.7 按照手机的型号输出手机的型号

### 

将显示这个型号手机的信息

### 1.3.7 退出

退出程序

# 2 系统设计

## 2.1用户用例图

系统分为收银员，管理员两个角色，各角色的用户用例如下图2-1所示：

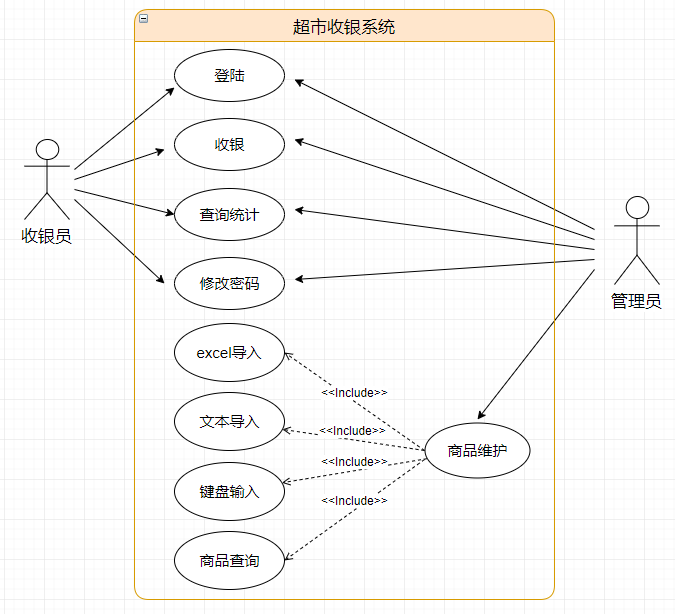
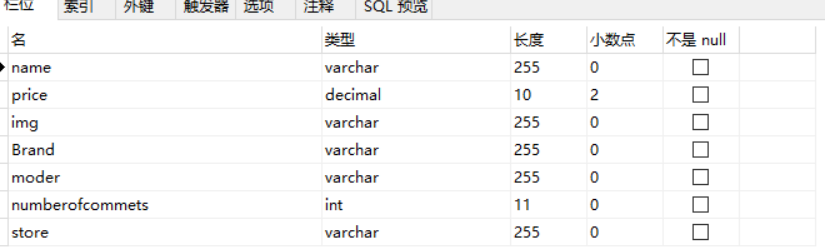


图2-1 用户用例图

## 2.2 ER图

数据库设计3张表，用户表，商品表，销售记录表，对应的ER图如图3-1所示。



爬虫这个系统中只有手机信息这一张表

图3-1 数据库ER图

## 2.3 UML类图（Class Diagram）

对于用户功能模块，共设计如下6个类。

* 数据库连接类DatabaseConnection：连接数据库并获取连接对象，关闭连接对象。
* VO实体类User：与数据库结构进行映射的类。主要由属性，setter, getter方法组成，VO类中的属性与表中的字段相对应，每一个VO类的对象都表示表中的每一条记录
* DAO接口IUserDAO：主要定义操作的接口，定义一系列数据库的原子性操作，例如增删改查（通常称为CRUD）等。
* DAO实现类UserDAOImpl：DAO接口的真实实现类，主要完成具体数据库操作，但不负责数据库的打开和关闭。
* DAO工厂类DAOFactory：通过工厂类取得一个DAO的实例化对象
* 业务逻辑实现类UserService：对于数据层的原子操作进行整合。还要负责数据库的打开与关闭（不管是否出异常，数据库都要关闭）
* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*：

各类的结构及类之间的关系如图2-5所示：

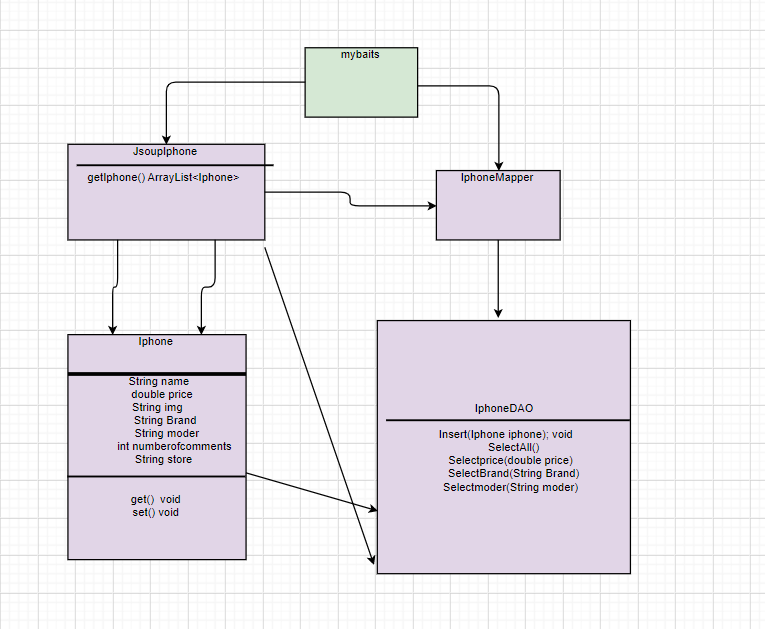


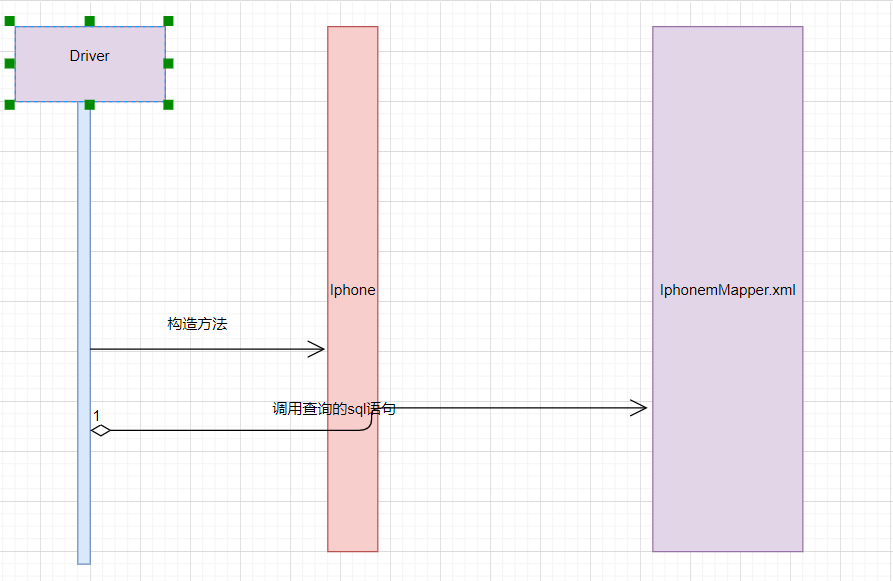
图2-5 用户功能类图

## 2.4 UML时序图（Sequence Diagram）

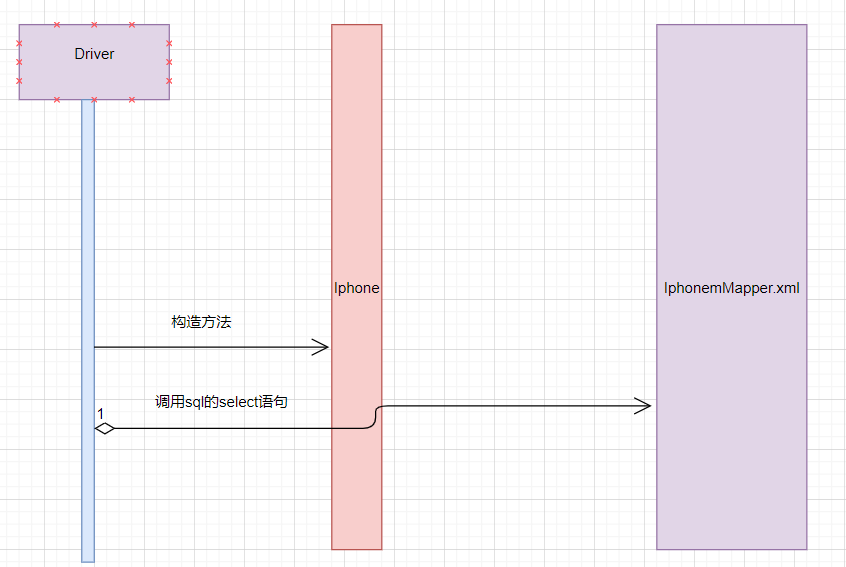
### 2.4.1 爬取网页

### 

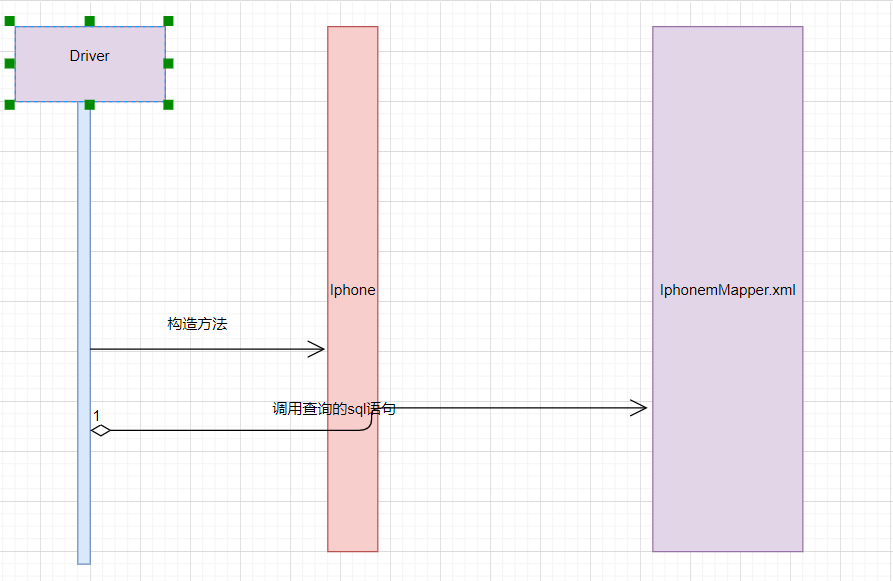
### 2.4.2 将爬取的网页信息存入数据库



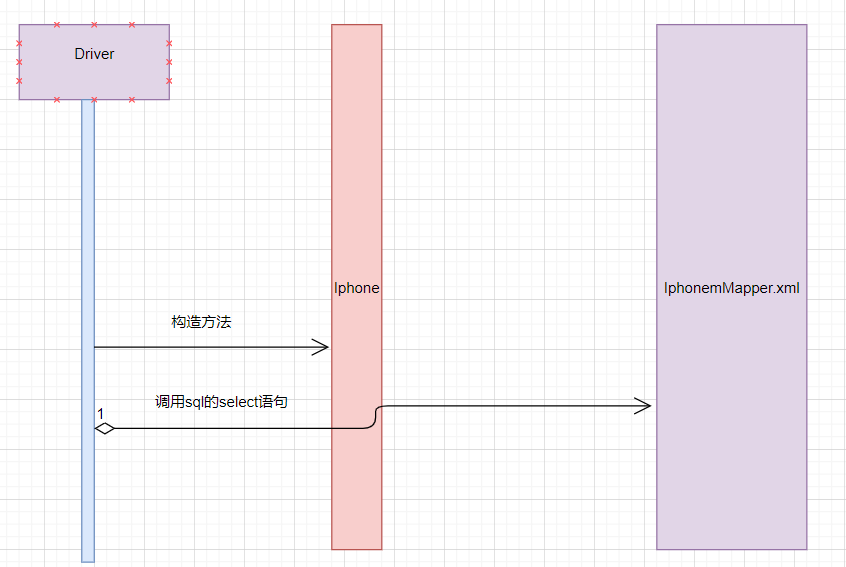
### 2.4.3 输出数据库中所有的信息



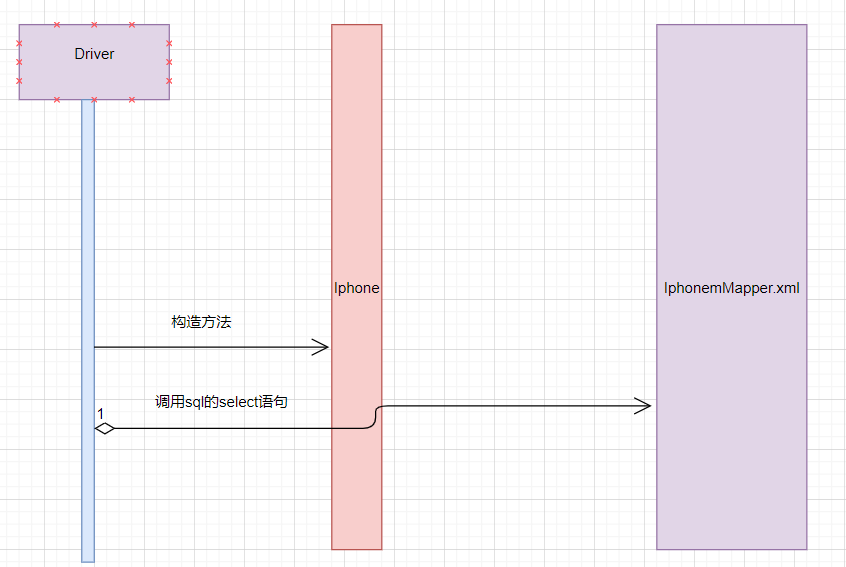
### 2.4.4 输出数据库中一定价格区间的手机信息



### 2.4.5 输出一个品牌的手机信息



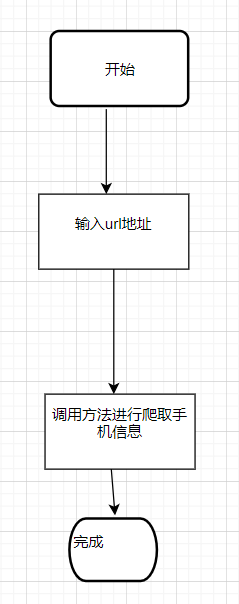
### 2.4.6 输出指定型号手机的信息



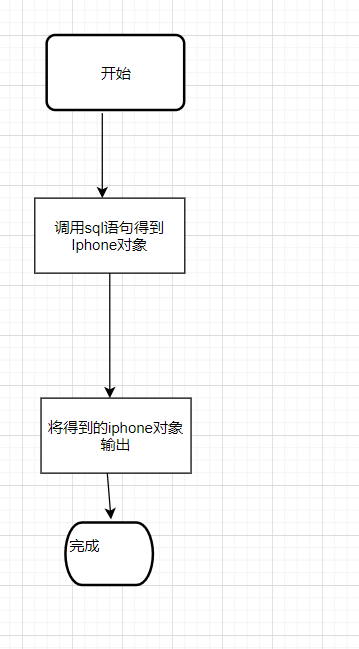
## 2.5 UML活动图（Activity Diagram）

### 2.5.1 输入网址爬取信息

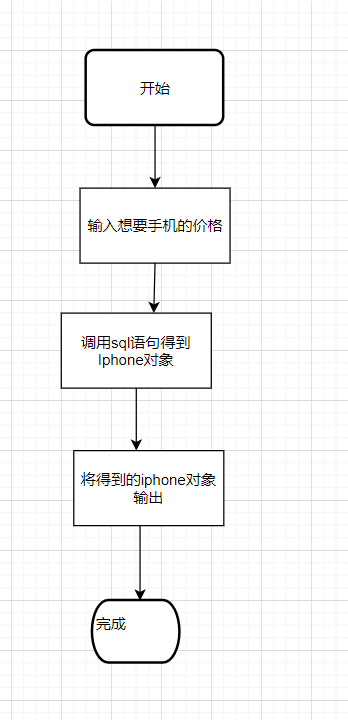
在天猫上面找到一个手机的页面，然后输入地址



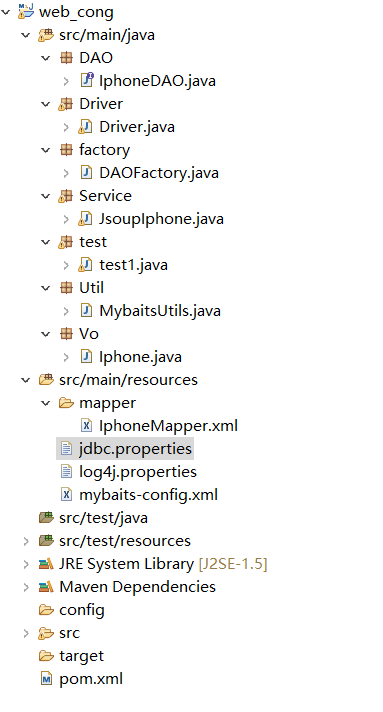
### 2.5.2 输出数据库所有的信息



### 2.5.3 输出数据库中一定价格区间的手机信息



## 3.1 项目结构



## 3.2 配置文件

在项目下新建Source Folder，名为resources

### 3.2.1 jdbc.properties文件

该配置文件主要是以文件形式保存数据库的驱动类名称，连接数据库的URL地址，访问数据库的用户名及对应的密码，程序运行时会读取该文件相关信息，避免硬编码，当相关信息发生变化时，只需修改配置文件而不用修改源代码，增加程序的可扩展型。

BDRIVER=com.mysql.jdbc.Driver

DBURL=jdbc:mysql://localhost:3306/iphone?useUnicode=true&characterEncoding=UTF-8

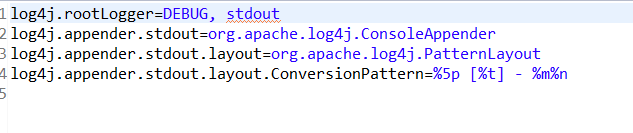
DBUSER=taojiale

### PASSWORD=102311tao

这是数据库相应连接所需要的信息

### 3.2.2 log4j.properties文件

描述该文件作用，并对里面的配置进行解释



这是日志文件

### 3.2.3 mybatis-config.xml文件

描述该文件作用，并对里面的配置进行解释

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE configuration

PUBLIC "-//mybatis.org//DTD Config 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-config.dtd">

<!-- 根标签 -->

<configuration>

<!-- 指定 资源文件 -->

<properties resource=*"jdbc.properties"*></properties>

<!-- 指定 MyBatis 所用日志的具体实现，未指定时将自动查找-->

<settings>

<setting name=*"logImpl"* value=*"LOG4J"* />

</settings>

<!-- 环境，可以配置多个，default：指定采用哪个环境 -->

<environments default=*"test"*>

<!-- id：唯一标识 -->

<environment id=*"test"*>

<!-- 事务管理器，JDBC类型的事务管理器 -->

<transactionManager type=*"JDBC"* />

<!-- 数据源，池类型的数据源 -->

<dataSource type=*"POOLED"*> <!--${BDRIVER} 打错了 -->

<property name=*"driver"* value=*"${BDRIVER}"* />

<property name=*"url"* value=*"${DBURL}"* />

<property name=*"username"* value=*"${DBUSER}"* />

<property name=*"password"* value=*"${PASSWORD}"* />

</dataSource>

</environment>

</environments>

<!-- 配置mapper映射文件 -->

<mappers>

<mapper resource=*"mapper/IphoneMapper.xml"* />

</mappers>

</configuration>

## 3.3 VO类Iphone.java

描述该文件作用，并对里面的关键代码进行解释

**package** Vo;

**public** **class** Iphone {

**private** String name;//姓名

**private** **double** price;//价格

**private** String img;//图片

**private** String Brand;//品牌

**private** String moder;//型号

**private** **int** numberofcomments;//评论数

**private** String store;//店铺

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **double** getPrice() {

**return** price;

}

**public** **void** setPrice(**double** price) {

**this**.price = price;

}

**public** String getImg() {

**return** img;

}

**public** **void** setImg(String img) {

**this**.img = img;

}

**public** String getBrand() {

**return** Brand;

}

**public** **void** setBrand(String brand) {

Brand = brand;

}

**public** String getModer() {

**return** moder;

}

**public** **void** setModer(String moder) {

**this**.moder = moder;

}

**public** **int** getNumberofcomments() {

**return** numberofcomments;

}

**public** **void** setNumberofcomments(**int** numberofcomments) {

**this**.numberofcomments = numberofcomments;

}

**public** String getStore() {

**return** store;

}

**public** **void** setStore(String store) {

**this**.store = store;

}

@Override

**public** String toString() {

**return** "Iphone [name=" + name + ", price=" + price + ", img=" + img + ", Brand=" + Brand + ", moder="

+ moder + ", numberofcomments=" + numberofcomments + ", store=" + store + "]";

}

**public** Iphone(String name, **double** price, String img, String brand, String moder, **int** numberofcomments,

String store) {

**super**();

**this**.name = name;

**this**.price = price;

**this**.img = img;

Brand = brand;

**this**.moder = moder;

**this**.numberofcomments = numberofcomments;

**this**.store = store;

}

}

## 3.4 DAO接口类IphoneDAO.java

描述该文件作用，并对里面的关键代码进行解释

**public** **interface** IphoneDAO {

**public** Iphone getByname(String name);//得到手机的信息

**public** **void** insert(Iphone iphone);//像数据库中插入手机的信息

**public** ArrayList<Iphone> SelectAll();//查找所有手机的信息

**public** ArrayList<Iphone> Selectprice(**double** price);//根据价格区间找到手机信息

**public** ArrayList<Iphone> SelectBrand(String Brand);//根据品牌找手机

**public** Iphone Selectmoder(String moder);//根据手机型号查找

## }

## 3.5 接口映射文件IphoneMapper.xml

描述该文件作用，并对里面的关键信息进行解释

<?xml version=*"1.0"* encoding=*"UTF-8"* ?>

<!DOCTYPE mapper

PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<!--这个上面的配置是版本问题 你现在用的mybatis版本低 -->

<mapper namespace=*"DAO.IphoneDAO"*>

<select id=*"getByname"* resultType=*"Vo.Iphone"* >

select \* from tiphone where name=#{name}

</select>

<select id=*"SelectAll"* resultType=*"Vo.Iphone"*>

select \* from tiphone

</select>

<select id=*"Selectprice"* resultType=*"Vo.Iphone"*>

select \* from tiphone where price&gt;#{price} order by price

</select>

<select id=*"SelectBrand"* resultType=*"Vo.Iphone"*>

select \* from tiphone where Brand like CONCAT('%',#{Brand},'%') order by price

</select>

<select id=*"Selectmoder"* resultType=*"Vo.Iphone"*>

select \* from tiphone where moder like CONCAT('%',#{moder},'%')

</select>

<insert id=*"insert"*>

insert into tiphone(name,price,img,Brand,moder,numberofcommets,store)

values

(#{name},#{price},#{img},#{Brand},#{moder},#{numberofcomments},#{store})

</insert>

</mapper>

## 3.6 工具包Util

### 3.6.1 MybatisUtils.java

描述该文件作用，并对里面的关键代码进行解释

/\*\*

\* 为服务层提供sqlSession对象

\*/

**public** **class** MybatisUtils {

**private** **static** SqlSessionFactory *sqlSessionFactory*;

**public** MybatisUtils() {

**super**();

}

/\*

\* SqlSessionFactory一旦被创建,应该在应用执行期间都存在.在应用运行期间不要重复创建多次

\*/

**static** {

String resourse = "mybatis-config.xml";

**try** {

InputStream inputStream = Resources.*getResourceAsStream*(resourse);

*sqlSessionFactory* = **new** SqlSessionFactoryBuilder().build(inputStream);

} **catch** (IOException e) {

e.printStackTrace();

}

}

**public** **static** SqlSession getSqlSession() {

// 如果不设置参数或者参数为false就是手动提交事务，参数设置为true就是自动提交事务

SqlSession sqlSession = *sqlSessionFactory*.openSession();

**return** sqlSession;

}

}

## 3.7 服务层JsoupIphone.java

描述该文件作用，并对里面的关键代码进行解释

**package Service;**

**import java.io.File;**

**import java.io.IOException;**

**import java.util.ArrayList;**

**import org.jsoup.Jsoup;**

**import org.jsoup.nodes.Document;**

**import org.jsoup.nodes.Element;**

**import org.jsoup.select.Elements;**

**import Vo.Iphone;**

**public class JsoupIphone {**

**//返回值可以是手机的容器**

**public ArrayList<Iphone> getIphone(String url) throws IOException {**

**//https://list.tmall.com/search\_product.htm?q=%BB%AA%CE%AA+%CA%D6%BB%FA&type=p&redirect=notRedirect**

**Document doc = Jsoup.connect(url) .timeout(60000).get();**

**System.out.println(doc.toString());**

**Elements productPrice=doc.getElementsByClass("product-iWrap");**

**int i=0;**

**//创建一个容器**

**ArrayList<Iphone> list=new ArrayList<Iphone>();**

**for(Element element:productPrice) {**

**//价格**

**String p\_price = element.getElementsByClass("productPrice").first().getElementsByTag("p").first().text();**

**String p\_tag=element.getElementsByClass("productPrice").first().getElementsByTag("b").first().text();**

**p\_price=p\_price.replace(p\_tag,"");**

**double price=Double.parseDouble(p\_price);**

**//应该将手机的价格适当的过滤下**

**if(price>300) {**

**String name=element.getElementsByClass("productTitle").first().getElementsByTag("a").first().text();**

**//手机的字符长度有限**

**//手机名字**

**//将第一个空格或者|的字符抓取**

**String[] namelist=name.split(" |/");**

**//品牌**

**String Brand=namelist[0];**

**//型号**

**String moder="";**

**for(int j=1;j<namelist.length;j++) {**

**moder=moder+namelist[j];**

**}**

**//评论数**

**int numberofcomments=0;**

**String p\_comments= element.getElementsByClass("productStatus").first().getElementsByTag("em").first().text();**

**//我们需要把评论数的万和笔改造一下**

**//将最后的笔去掉**

**p\_comments=p\_comments.replace("笔"," ");**

**//将万这个数字进行 改造**

**if(p\_comments.indexOf("万")!=-1) {**

**//将万字去除**

**p\_comments=p\_comments.replace("万", "");**

**//我们需要将点去掉,相当于数字乘以10倍了**

**p\_comments=p\_comments.replace(".", "");**

**numberofcomments=(int)(Double.parseDouble(p\_comments)\*1000);**

**}**

**else {**

**numberofcomments=(int)Double.parseDouble(p\_comments);**

**}**

**//店铺名字**

**String store= element.getElementsByClass("productShop-name").first().getElementsByTag("a").first().text();**

**//手机图片**

**String img=element.getElementsByClass("proThumb-wrap").first().getElementsByTag("img").first().attr("data-ks-lazyload");**

**// public Iphone(String name, double price, String img, String brand, String moder, int numberofcomments,**

**// String store) {**

**//接下来我们实例化手机**

**Iphone iphone=new Iphone(name,price,img,Brand,moder,numberofcomments,store);**

**list.add(iphone);**

**}**

**}**

**return list;**

**}**

**}**

## 3.8 用户界面Driver.java

描述该文件作用，并对里面的关键代码进行解释

package Driver;

import java.io.IOException;

import java.io.InputStream;

import java.util.ArrayList;

import java.util.Scanner;

import org.apache.ibatis.io.Resources;

import org.apache.ibatis.session.SqlSession;

import org.apache.ibatis.session.SqlSessionFactory;

import org.apache.ibatis.session.SqlSessionFactoryBuilder;

import DAO.IphoneDAO;

import Service.JsoupIphone;

import Vo.Iphone;

public class Driver {

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

// 指定全局配置文件

String resource = "mybaits-config.xml";

// 读取配置文件

InputStream inputStream = Resources.getResourceAsStream(resource);

// 构建sqlSessionFactory

SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);

// 获取sqlSession

SqlSession sqlSession = sqlSessionFactory.openSession();

// 执行方式2：自动获取相应的DAO实现代理对象

IphoneDAO dao = sqlSession.getMapper(IphoneDAO.class);

// https://list.tmall.com/search\_product.htm?q=%BB%AA%CE%AA+%CA%D6%BB%FA&type=p&redirect=notRedirect

JsoupIphone j = new JsoupIphone();

ArrayList<Iphone> list=new ArrayList<Iphone>();//存放网页中返回手机的信息

int x;

Scanner in = new Scanner(System.in);

meau();

System.out.print("请输入你的选择:");

x = in.nextInt();

while (x != 7) {

Scanner in2 = new Scanner(System.in);

switch (x) {

case 1:

System.out.println("请输入url:");

String url=in2.nextLine();

list=j.getIphone(url);

break;

case 2:

for(Iphone iphone:list) {

dao.insert(iphone);

}

System.out.println("成功插入");

sqlSession.commit();//提交事务

break;

case 3:

list=dao.SelectAll();

for(Iphone iphone:list) {

System.out.println(iphone.toString());

}

System.out.println("成功输出");

break;

case 4:

System.out.println("请输入你想想购买手机的价格例如大于1000的手机");

String sprice=in2.nextLine();

//将数据分隔开

double price=Double.parseDouble(sprice);

list=dao.Selectprice(price);

for(Iphone iphone:list) {

System.out.println(iphone.toString());

}

System.out.println("成功输出");

break;

case 5:

System.out.println("请输入你想查询手机的品牌");

String Brand=in2.nextLine();

list=dao.SelectBrand(Brand);

for(Iphone iphone:list) {

System.out.println(iphone.toString());

}

System.out.println("成功输出");

break;

case 6:

System.out.println("请输入你想查询手机的型号");

String moder=in2.nextLine();

Iphone i= dao.Selectmoder(moder);

if(i==null) {

System.out.println("没有该型号的手机");

}

else {

System.out.println(i.toString());

System.out.println("成功输出");

}

}

meau();

System.out.print("请输入你的选择:");

x = in.nextInt();

}

System.out.println("退出程序");

}

public static void meau() {

System.out.println("1.请输入你想爬虫的手机页面URL地址,将爬取网页:");

System.out.println("2.将爬虫得到手机数据存入数据库");

System.out.println("3.输出数据库中所有的手机信息");

System.out.println("4.输出一个区间手机的价格");

System.out.println("5.输出一个手机品牌的手机");

System.out.println("6.输出指定型号的手机");

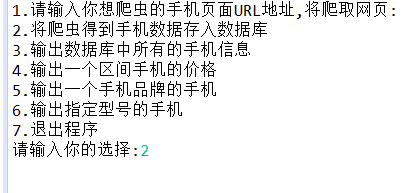
System.out.println("7.退出程序");

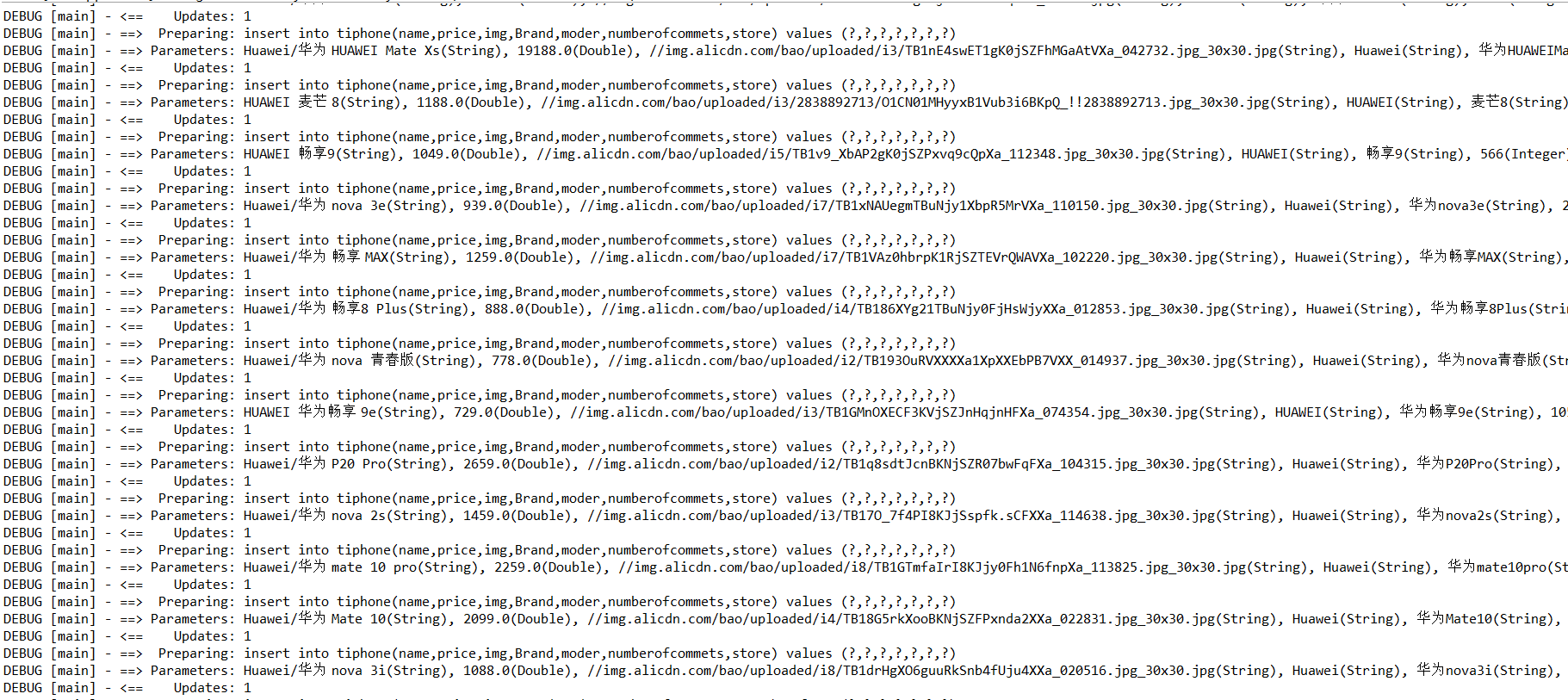
}

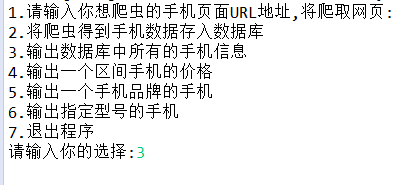
}

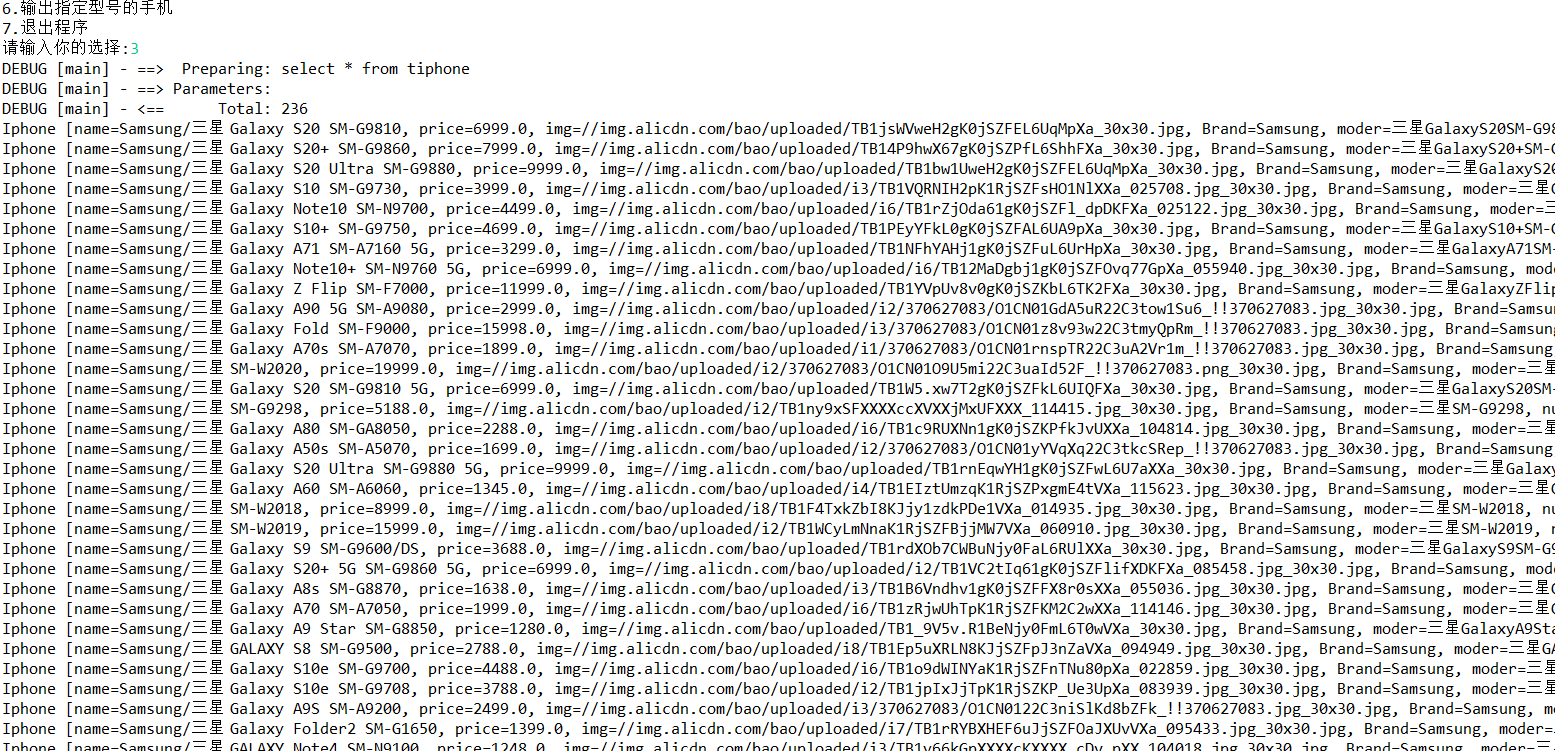
# 4 系统测试

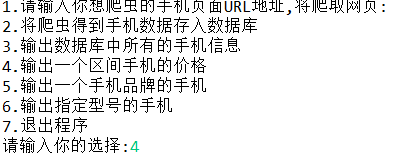
设计测试用例，给出程序每个功能模块的运行结果截图。

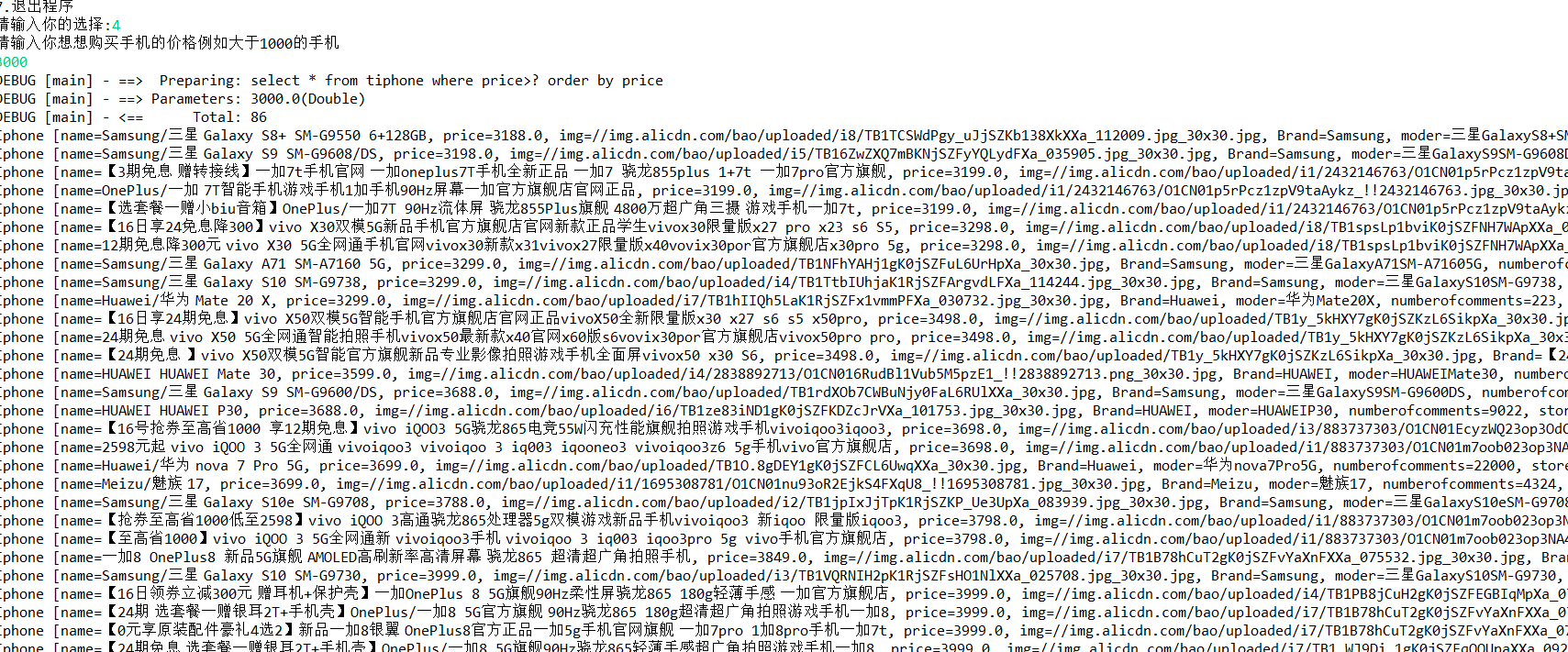


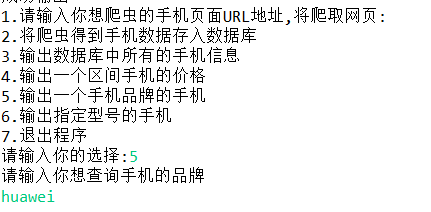


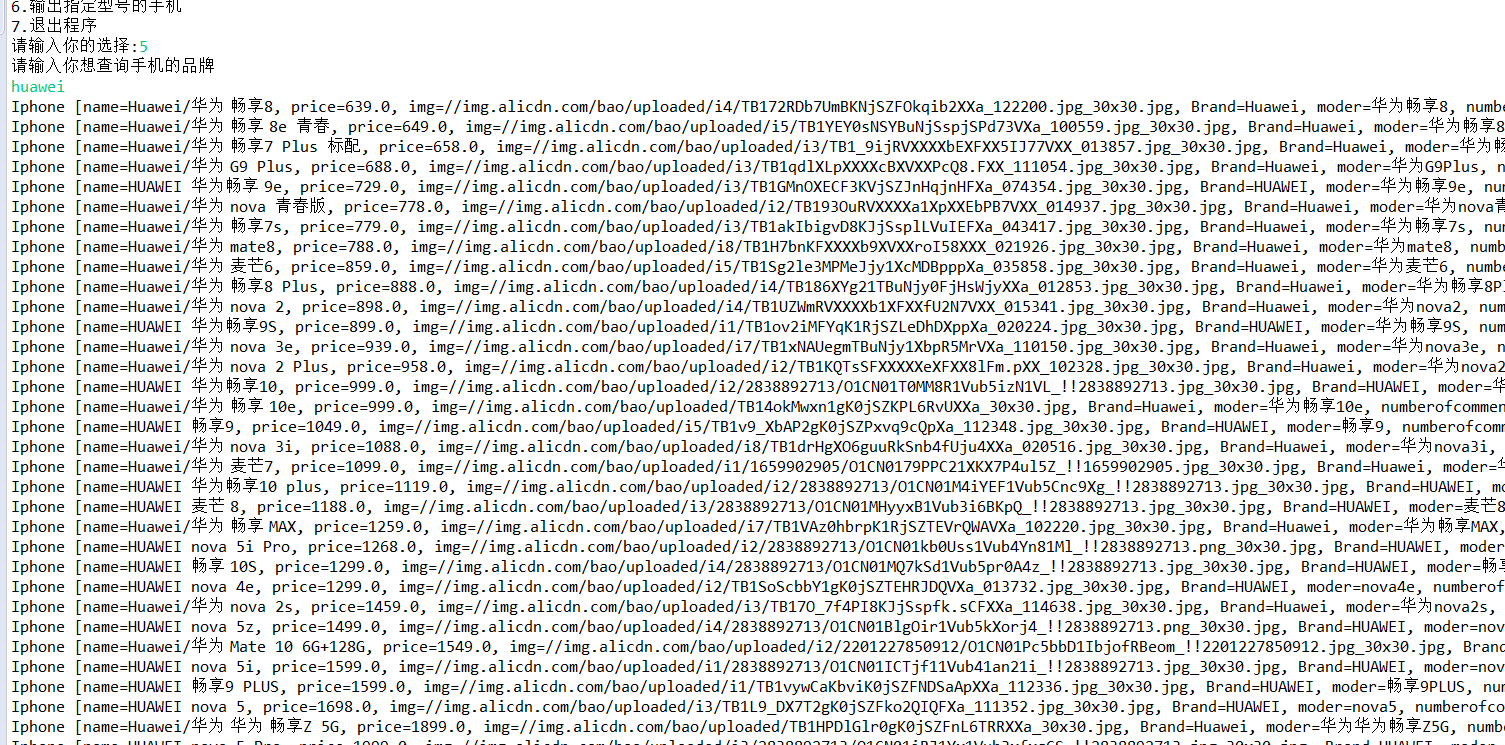














# 5 系统总结

**主要问题的解决过程**

在最开始的时候配置Maven的时候遇到了很多的错误，然后根据自己不断地上网翻查资料

以及向别人请教才终于配置完成，然后再配置mybaits的Xml的文件时候，自己下载的版本号比较老，一直在报错误，这个问题困扰了我很久，然后自己换了一个文件就配置成功了

**系统存在的不足**

目前我的爬虫系统只是在天猫页面上面爬取手机的信息

但是并不是所有手机的信息都可以爬取到

**本门课程的收获**

我认为此门课程让我更加具备面向对象的思想，然后这门课程讲的东西都是非常实用的知识对我们将来就业有很大的帮助

**课程的建议**

建议把一些知识点讲细化一点，有很多东西感觉不是很懂

**以后的发展方向**

自己以后比较想写后端

自评成绩

自己给自己打分75分，自己前面的学习比较认真，到后面课程难度加大，自己又松懈