# 综合项目实战

## 项目描述

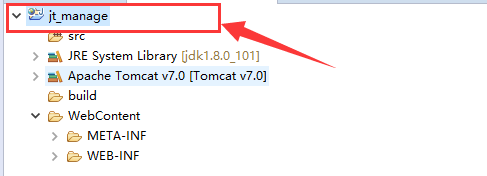
### 项目描述

利用JavaWeb阶段所学的技术，如html、css、Servlet、JSP、el、JSTL、JDBC等技术实现电商项目（京淘jt）中后台管理系统的商品管理模块。实现商品添加、商品列表、商品删除、商品修改等功能。

## 环境搭建

### 创建工程

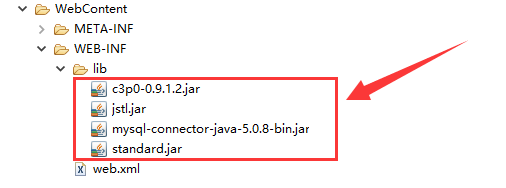
创建一个动态web工程



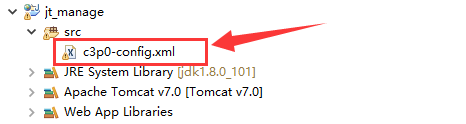
### 导入jar包

导入工程中所需的jar包和配置文件

**1.导入jar包(放在工程的/WEB-INF/lib目录下!!!)**



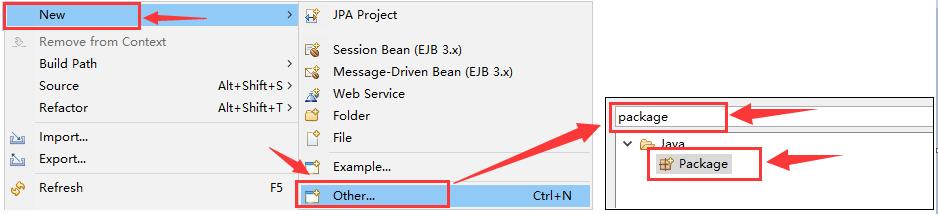
**2.导入配置文件(放在src目录下!!!)**



### 组织工程目录结构

**1.创建包结构**

(1)在src下创建com.tedu.jt.utils包和com.tedu.jt.web.backend包





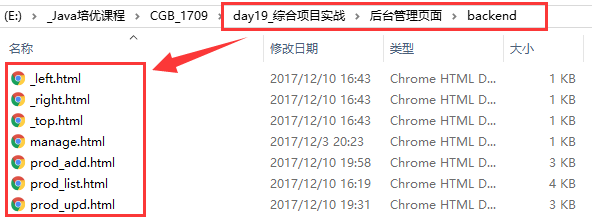
(2)在WebContent目录下，创建backend目录用来存放项目中的jsp页面



### 导入页面

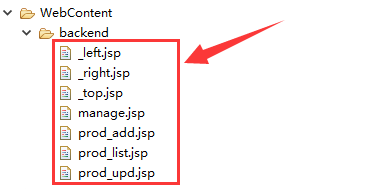
将课程资料中提供的静态页面导入工程中

**1.静态页面**



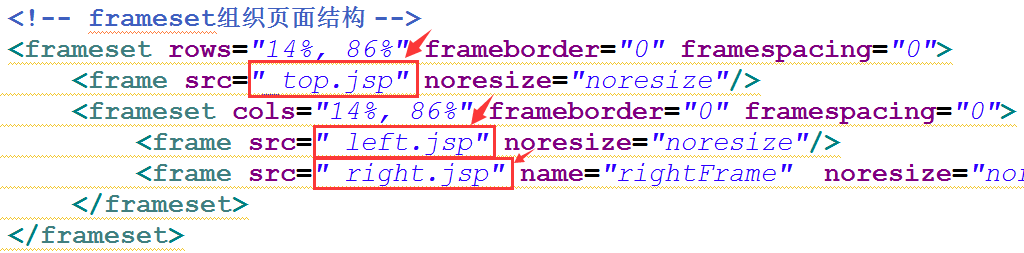
**2.导入工程**

在工程的/WebContent/backend目录下创建对应的jsp页面，例如：manage.html 🡪 manage.jsp

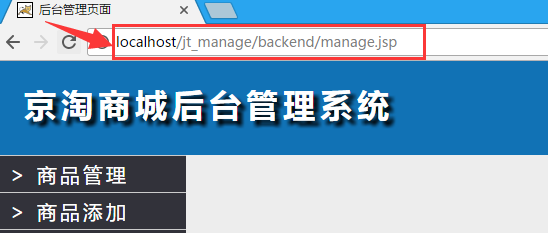


**3.修改页面代码**

(1)修改manage.jsp



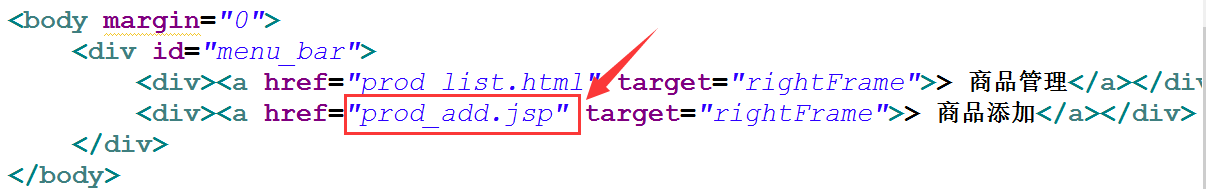
4.测试访问



## 代码实现\_1\_商品添加

### 商品添加入口开发

**1.入口开发：编辑\_left.jsp页面，修改”商品添加”链接地址，点击商品添加跳转到prod\_list.jsp页面。**

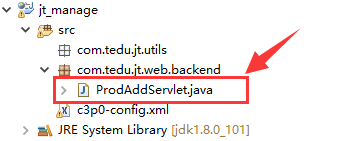


**2.编辑prod\_add.jsp页面，修改商品添加表单提交地址,提交表单访问ProdAddServlet来处理添加商品请求。**

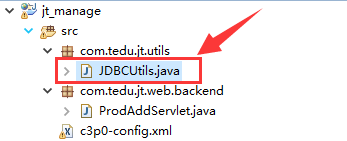


### 商品添加后台实现

**1.创建ProdAddServlet，用于处理客户端的添加商品请求**



**2.导入JDBCUtils工具类**



提示: JDBCUtils工具类代码实现参见本文档<<1.7.1>>章节

**3.设计表结构**

-- 0.删除jt\_db数据库(如果存在)

drop database if exists jt\_db;

-- 1.创建数据库jt\_db

create database jt\_db;

-- 2.选中jt\_db数据库

use jt\_db;

-- 3.在 jt\_db 库中创建product表(商品表)

create table product(

id int primary key auto\_increment, -- 商品ID

name varchar(50), -- 商品名称

category varchar(50), -- 商品分类

price double, -- 商品单价

pnum int, -- 库存数量

description varchar(50) -- 商品描述

);

**4.实现ProdAddServlet，处理商品添加请求**

(1)实现ProdAddServlet的doGet方法



(2)在ProdAddServlet中添加addProd方法用于将商品信息保存到数据库中，代码实现如下:

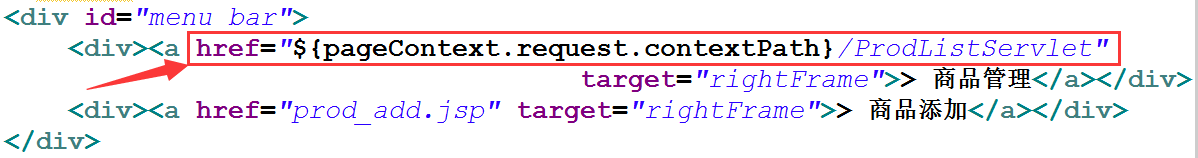


**5.测试:添加一个商品，查看数据中的product表中，是否具有新增的商品记录（略）**

## 代码实现\_2\_商品列表

### 商品列表入口开发

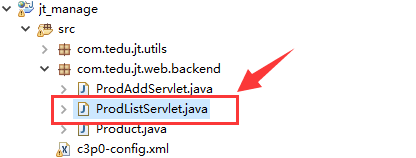
**1.入口开发:编辑\_left.jsp中”商品管理”超链接，点击连接访问ProdListServlet查询所有商品信息.**



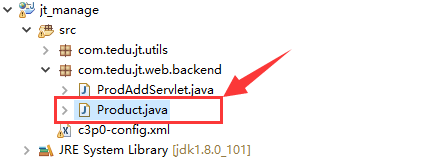
### 商品列表后台实现

创建ProdListServlet，查询所有商品信息，将商品信息封装到Product对象中，并将所有商品存入list集合中，存入到request域中转发到prod\_list.jsp中展示.

**1.创建ProdListServlet**



**2.创建Product类，用于表示商品信息**



提示: Product类实现参见本文档<<1.7.2>>章节

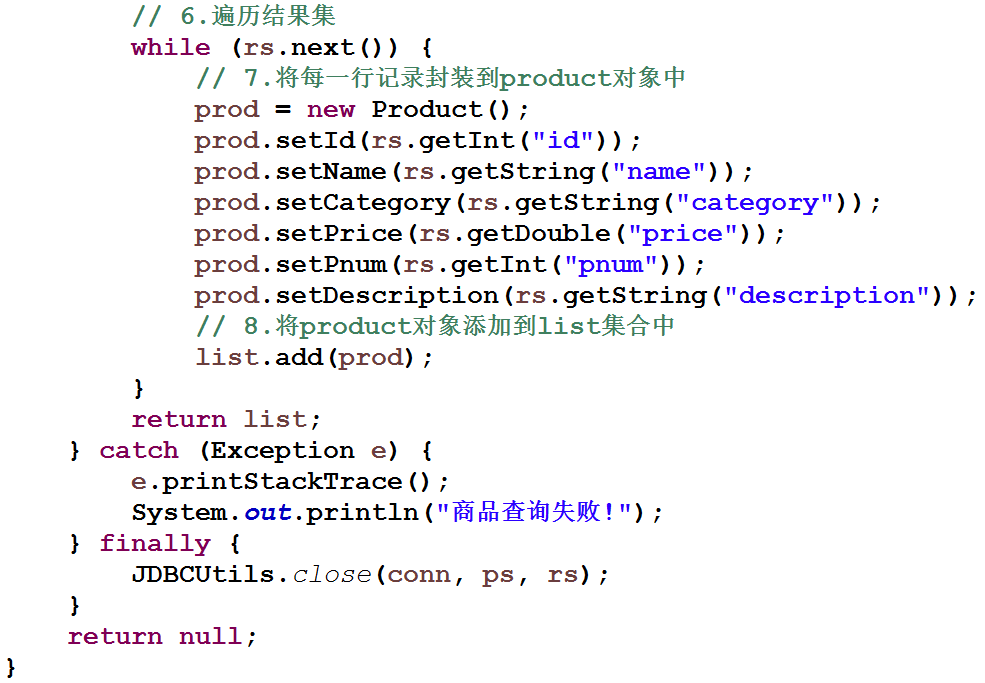
**3.实现ProdListServlet类**

(1)实现ProdListServlet中的doGet方法



(2)在ProdListServlet中添加findProdList方法用于查询所有的商品信息，代码实现如下:



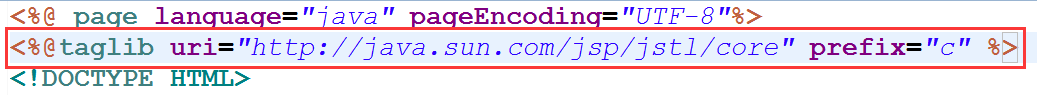


### 商品列表页面实现(prod\_list.jsp)

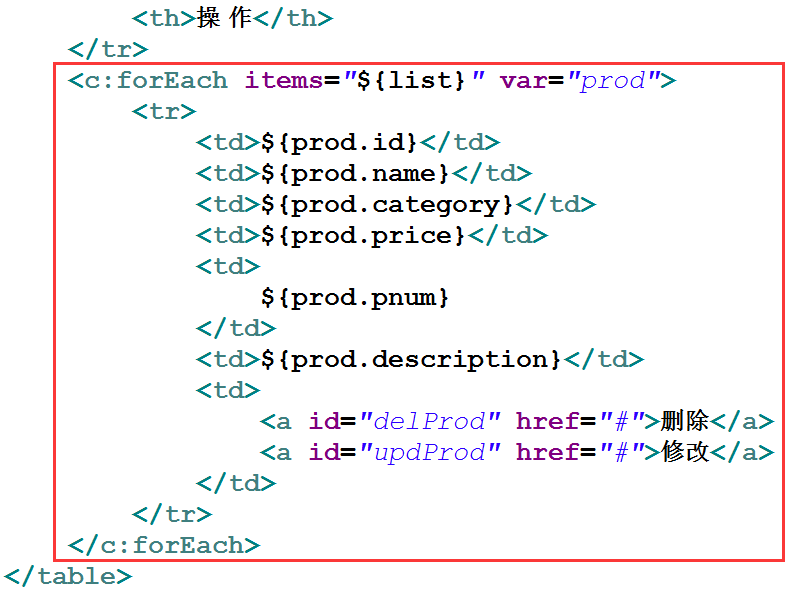
在ProdListServlet中将会查询出所有商品信息，存入List集合并将List集合存入request域中，通过转发带到prod\_list.jsp页面，在prod\_list.jsp中通过JSTL标签(c:forEach)取出集合中的数据进行展示.

**1.编辑prod\_list.jsp，使用EL+JSTL遍历List集合中的商品信息**

(1)通过taglib指令导入JSTL标签库



(2)使用EL+JSTL遍历商品信息

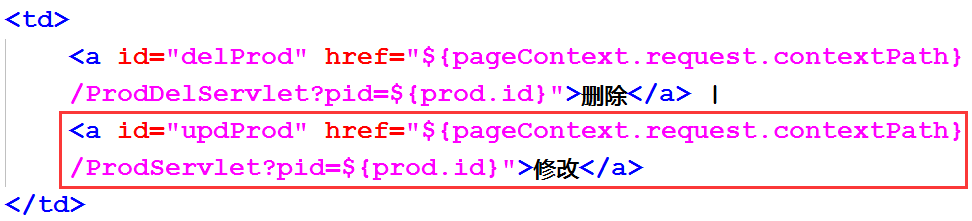


**2.测试: 点击商品管理查询所有商品信息(略)**

## 代码实现\_3\_商品修改

### 商品修改入口开发

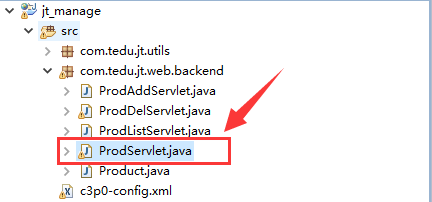
**1.入口开发:编辑prod\_list.jsp中”修改”超链接，点击“修改”连接，访问ProdUpdServlet，根据商品ID查询商品信息.**



### 商品修改后台实现\_1(ProdServlet)

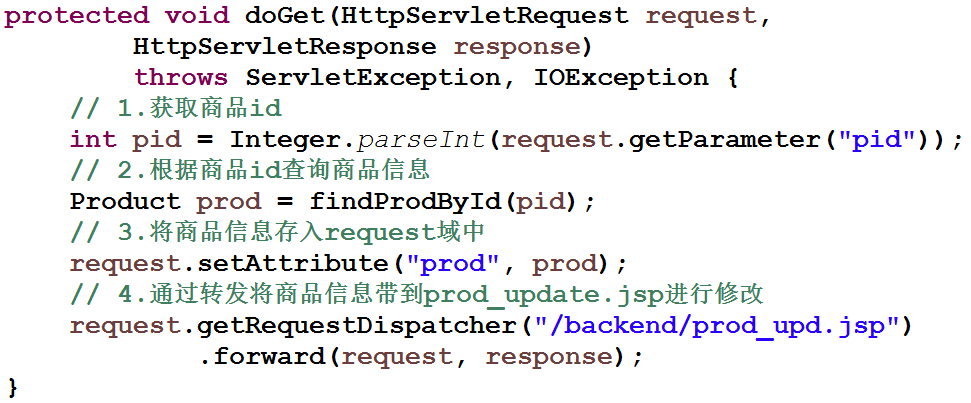
创建ProdServlet，根据商品ID查询商品信息，并将商品信息存入request域通过转发将商品信息带到prod\_upd.jsp回显

**1.创建ProdServlet**



**2.实现ProdServlet**

(1)实现ProdServlet的doGet方法



(2)在ProdServlet中添加findProdById方法并实现，如下:





### 商品修改页面实现(prod\_upd.jsp)

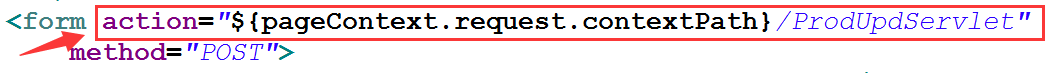
在ProdServlet中将会查询出所要修改的商品信息，存入request域中，通过转发带到prod\_upd.jsp页面，在prod\_upd.jsp中通过EL取出集合中的数据进行回显，供用户修改.

**1.编辑prod\_upd.jsp，通过el表达式取出所要修改的商品信息**





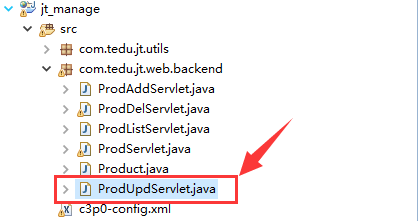
**2.编辑prod\_upd.jsp，修改”修改商品信息”表单的提交地址.**



### 商品修改后台实现\_2(ProdUpdServlet)

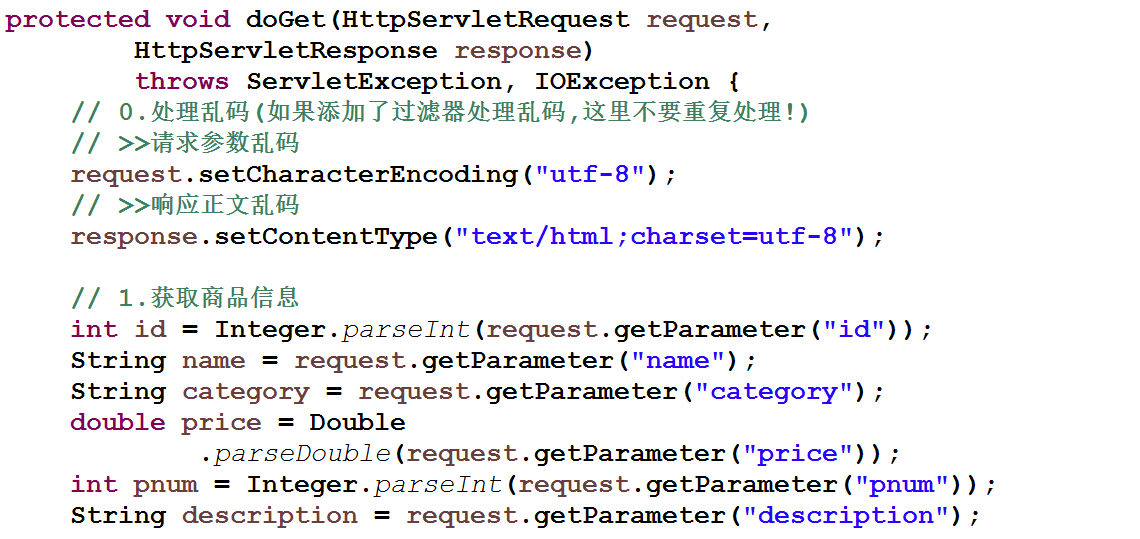
创建ProdUpdServlet，修改指定ID的商品信息，修改完后定时刷新到ProdListServlet查询所有商品信息.

1.创建ProdUpdServlet.



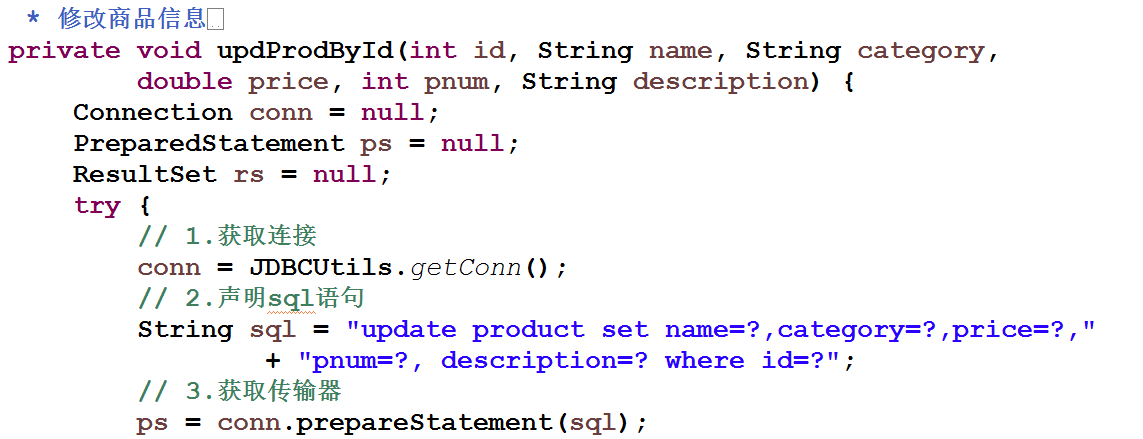
2.实现ProdUpdServlet.

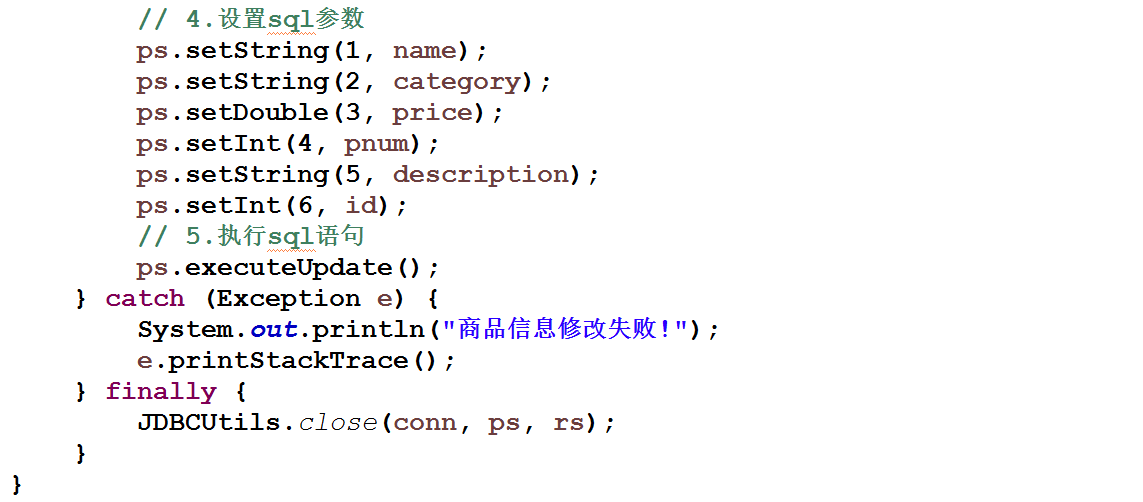
(1)实现ProdUpdServlet中的doGet方法.





(2)在ProdUpdServlet中添加updProdById方法并实现，如下:

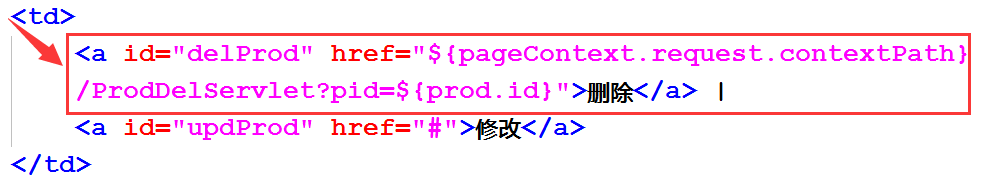




## 代码实现\_4\_商品删除

### 商品删除入口开发

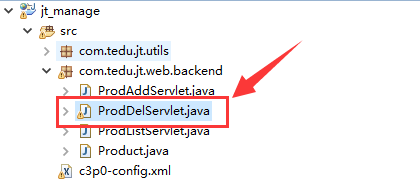
**1.入口开发，编辑prod\_list.jsp中”删除”超链接，点击连接，访问ProdDelServlet删除指定ID的商品信息.**



### 商品删除后台实现

创建ProdDelServlet，根据请求参数中的商品ID删除指定ID的商品记录. 删除后定时刷新到ProdListServlet查询所有商品信息.

**1.创建ProdDelServlet**



2.实现ProdDelServlet，根据商品ID删除商品  
 (1)实现ProdDelServlet中的doGet方法.



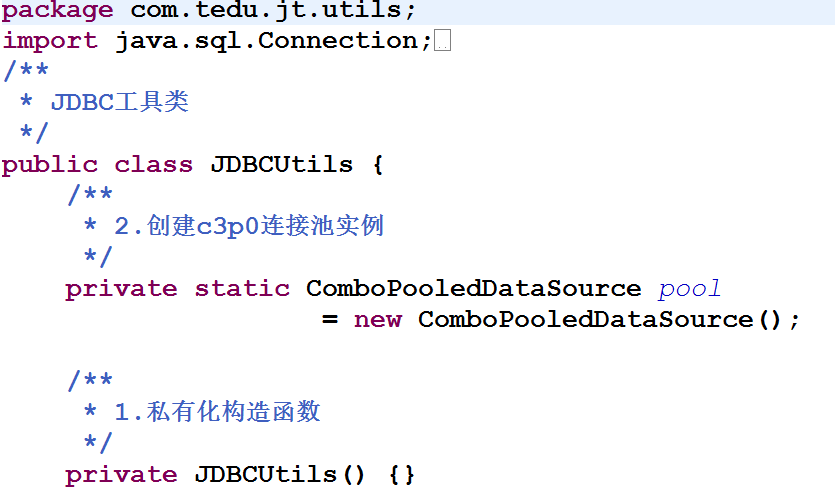
(2)在ProdDelServlet中添加delProdById方法并实现，如下:



## 代码实现\_5\_细节处理

### JDBCUtils工具类实现

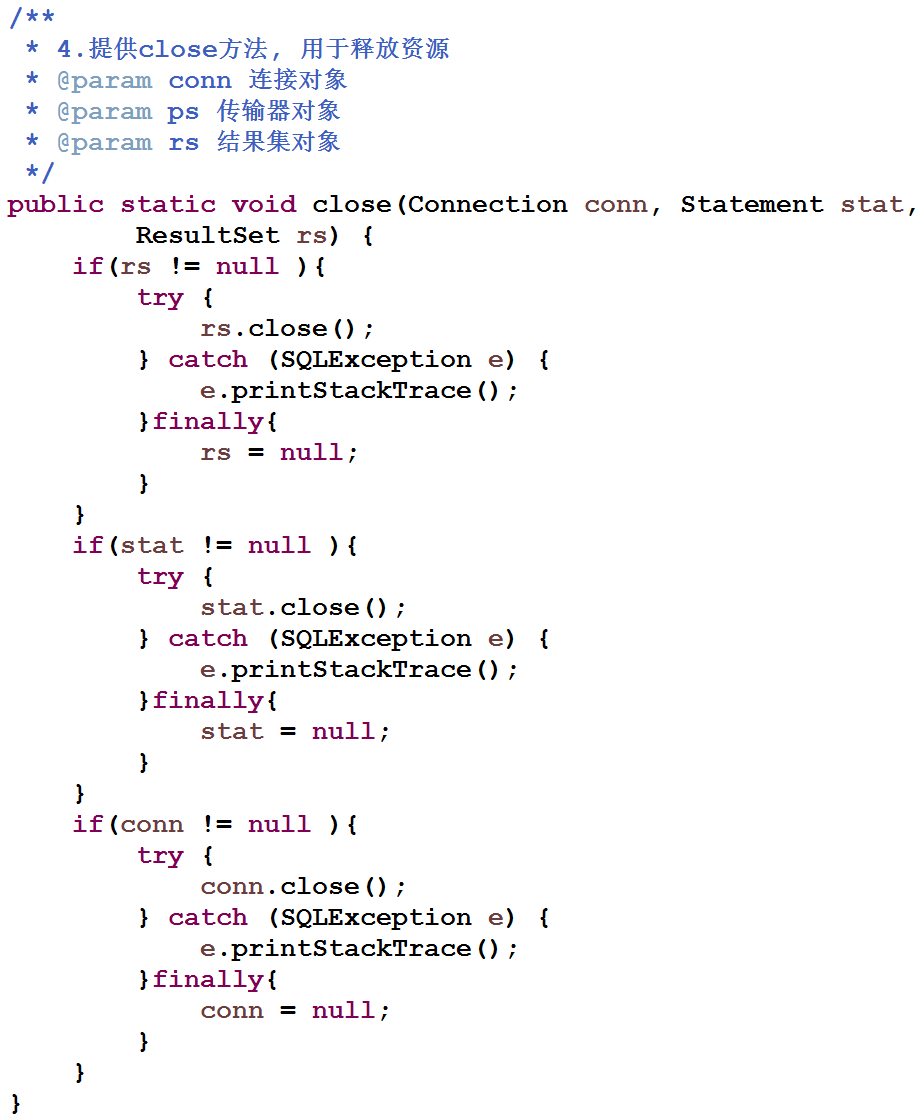
**1.私有化构造函数、创建c3p0连接池实例**



**2.提供getConnection方法，用于从连接池中获取一个连接对象**

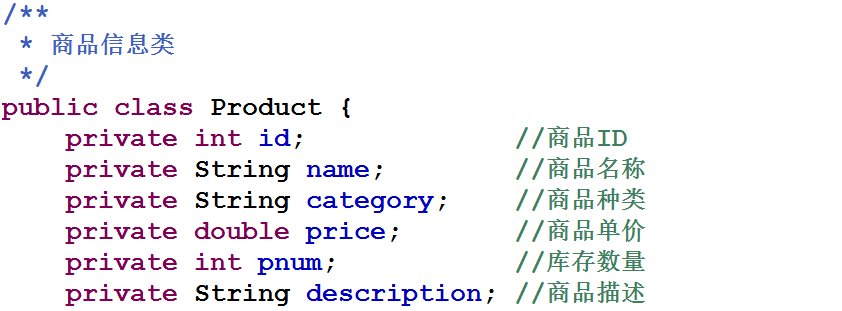


**3．提供close方法，用于释放资源（其中调用conn.close()是将连接还回连接池中而不是关闭连接）。**

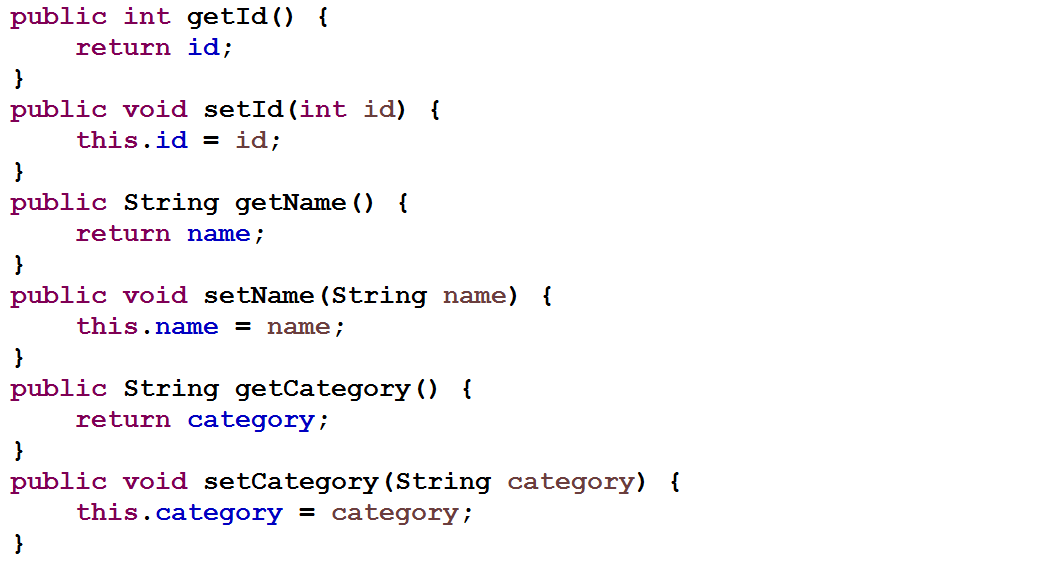


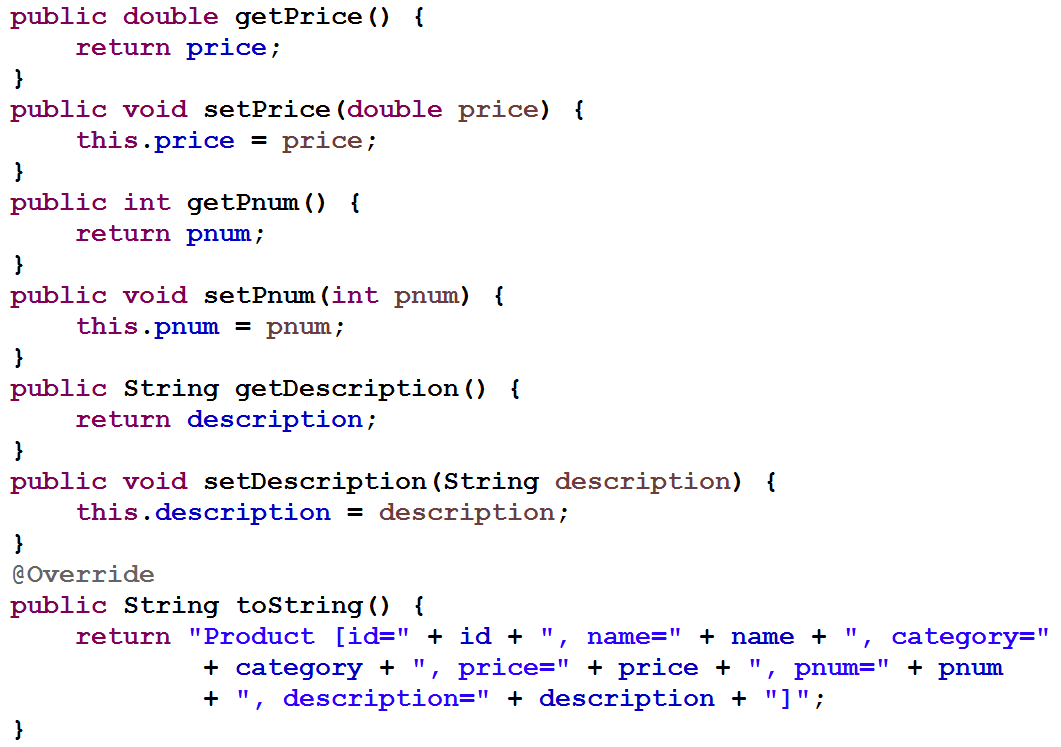
### Product(商品类)实现

**1.创建商品类，添加如下成员**



**2.为成员变量提供对应的setter和getter方法**





### 异常处理

**问题描述：** 当商品的添加、查询、删除、修改成功后会提示“商品添加、查询、修改、删除成功！”，但是如果操作失败了，还是会提示“xxx成功”，但其实应该提示“xxx失败”的信息。

**解决方案：**

1、在web.xml文件中为整个项目的异常信息配置友好错误提示页面，当抛出异常时，不要给用户显示提示异常信息的页面，而是提示友好错误提示页面. 配置如下：

<error-page>

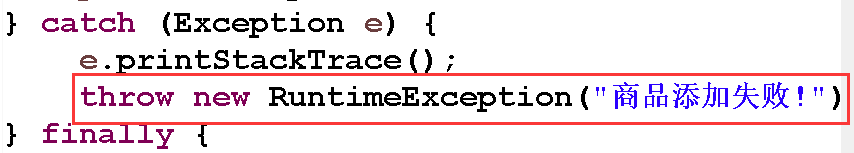
<error-code>500</error-code>

<location>/backend/error/500.jsp</location>

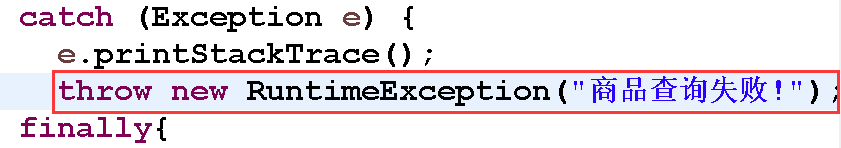
</error-page>

2、修改Servlet中针对异常抛出时处理的代码：

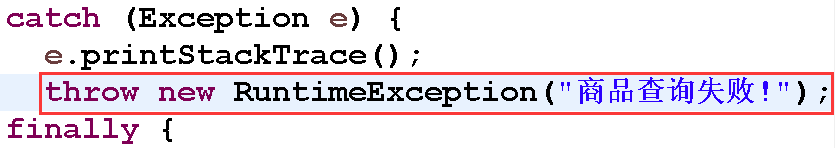
（1）在ProdAddServlet中的addProd方法中， 修改如下：



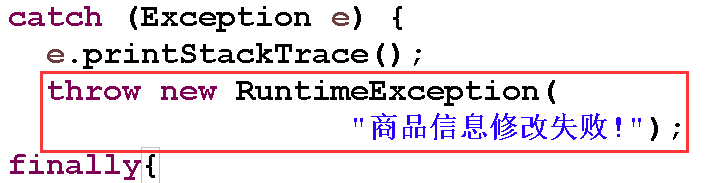
（2）在ProdListServlet中的findProdList方法中， 修改如下：



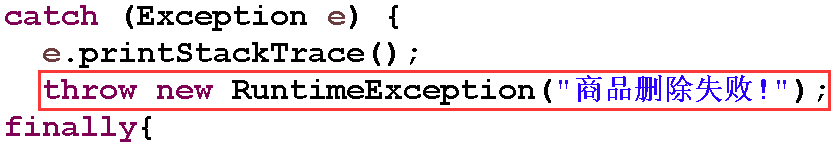
（3）在ProdInfoServlet中的findProdById方法中， 修改如下：



（4）在ProdUpdServlet中的updProdById方法中， 修改如下：



（5）在ProdDelServlet中的delProdById方法中， 修改如下：



## MVC设计模式

MVC设计模式: 通用的软件编程思想, 在MVC设计模式中认为, 任何软件都可以分为：控制程序流转的控制器（Controller）、封装数据处理数据的模型（Model）、负责展示数据的视图（view）三部分组成。并且在MVC设计思想中要求一个符合MVC设计思想的软件应该保证上面这三部分相互独立，互不干扰，每一个部分只负责自己擅长的部分。如果某一个模块发生变化，应该尽量做到不影响其他两个模块。这样做的好处是，软件的结构会变得更加的清晰，可读性强。有利于后期的扩展和维护，并且代码可以实现复用。