**Java Web开发**

**实验八**

**作业报告**

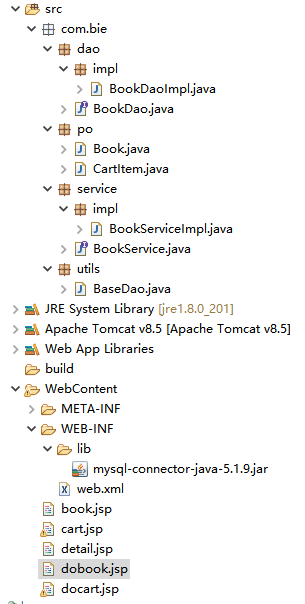
|  |  |
| --- | --- |
| 学 院 | 计算机 |
| 所在系 班 级 | 2017软件工程系6班 |
| 学 号 姓 名 | 04171411 |
| 指 导 教 师 | 郭晓燕 |
| 完 成 时 间 | 2019年5月7号 |

# 实验题目

1、（必做）请用模型2MVC模式完成1个功能，功能自选，例如：登录、注册、发帖、计算税收等；

2、（选做）继续完善网上购图书程序，采用MVC模式，完成增加计算商品总价及添加商品数量等功能，用JavaBean改写，其中购物车和商品价格等内容都用JavaBean实现，JSP页面实现增加、删除所选图书，显示所选图书及其价格的功能。尽量用EL表达式或JSTL标签库，少用程序片及表达式。

# 分析与设计



# 关键源码（部分）

BookDaoImpl.java

**public** **class** BookDaoImpl **implements** BookDao{

@Override

**public** List<Book> select(String sql, Object[] arr) {

Connection con=**null**;

PreparedStatement ps=**null**;

ResultSet rs=**null**;

**try** {

con=BaseDao.*getCon*();//第一步连接数据库

ps=con.prepareStatement(sql);//第二步：预编译

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

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

ps.setObject(i+1, arr[i]);

}

}

//第四步执行sql

rs=ps.executeQuery();

List<Book> list=**new** ArrayList<Book>();

**while**(rs.next()){

Book book=**new** Book();

book.setBookid(rs.getInt("bookid"));

book.setBookname(rs.getString("bookname"));

book.setPrice(rs.getDouble("price"));

book.setAuthor(rs.getString("author"));

book.setPic(rs.getString("pic"));

book.setPublish(rs.getString("publish"));

list.add(book);

}

**return** list;

} **catch** (ClassNotFoundException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}**finally**{

//关闭资源，避免出现异常

BaseDao.*close*(con, ps, rs);

}

**return** **null**;

}

@Override

**public** Book getBook(Integer id) {

Connection con=**null**;

PreparedStatement ps=**null**;

ResultSet rs=**null**;

**try** {

con=BaseDao.*getCon*();//第一步连接数据库

String sql="select \* from book where bookid = ? ";

ps=con.prepareStatement(sql);//第二步：预编译

ps.setInt(1, id);

//第四步执行sql

rs=ps.executeQuery();

**while**(rs.next()){

Book books=**new** Book();

books.setBookid(rs.getInt("bookid"));

books.setBookname(rs.getString("bookname"));

books.setPrice(rs.getDouble("price"));

books.setAuthor(rs.getString("author"));

books.setPic(rs.getString("pic"));

books.setPublish(rs.getString("publish"));

**return** books;

}

} **catch** (ClassNotFoundException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}**finally**{

//关闭资源，避免出现异常

BaseDao.*close*(con, ps, rs);

}

**return** **null**;

}

}

BookDao.java

**public** **interface** BookDao {

**public** List<Book> select(String sql,Object[] arr);

**public** Book getBook(Integer id);

}

Book.java

**public** **class** Book **implements** Serializable{

//实体类实现序列化，避免后面出现异常

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** Integer bookid;

**private** String bookname;

**private** Double price;

**private** String author;

**private** String pic;

**private** String publish;

**public** Integer getBookid() {

**return** bookid;

}

**public** **void** setBookid(Integer bookid) {

**this**.bookid = bookid;

}

**public** String getBookname() {

**return** bookname;

}

**public** **void** setBookname(String bookname) {

**this**.bookname = bookname;

}

**public** Double getPrice() {

**return** price;

}

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

**this**.price = price;

}

**public** String getAuthor() {

**return** author;

}

**public** **void** setAuthor(String author) {

**this**.author = author;

}

**public** String getPic() {

**return** pic;

}

**public** **void** setPic(String pic) {

**this**.pic = pic;

}

**public** String getPublish() {

**return** publish;

}

**public** **void** setPublish(String publish) {

**this**.publish = publish;

}

@Override

**public** String toString() {

**return** "Book [bookid=" + bookid + ", bookname=" + bookname + ", price=" + price + ", author=" + author

+ ", pic=" + pic + ", publish=" + publish + "]";

}}

Cartltem.java

**public** **class** CartItem {

**private** Book book;// 图书对象的成员变量

**private** Integer number;// 购买的数量；

**public** Book getBook() {

**return** book;

}

**public** **void** setBook(Book book) {

**this**.book = book;

}

**public** Integer getNumber() {

**return** number;

}

**public** **void** setNumber(Integer number) {

**this**.number = number;

}

}

BookServiceImpl.java

**public** **class** BookServiceImpl **implements** BookService{

**private** BookDao dao=**new** BookDaoImpl();

**public** List<Book> select(Book book){

//String sql="select \* from book ";

StringBuilder sql=**new** StringBuilder("select \* from book where 1=1 ");

//sql语句

List<Object> list=**new** ArrayList<Object>();

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

**if**(book.getBookid()!=**null** && book.getBookid()!=0){

sql.append(" and bookid=? ");

list.add(book.getBookid());

}

/\*list.add(book.getBookname());

list.add(book.getPrice());

list.add(book.getAuthor());

list.add(book.getPic());

list.add(book.getPublish());\*/

}

**return** dao.select(sql.toString(), list.toArray());

}

@Override

**public** Book getBook(Book book) {

**if**(book.getBookid()!=**null** && book.getBookid()!=0){

**return** dao.getBook(book.getBookid());

}

**return** **null**;

}

}

BaseDao.java

**public** **class** BaseDao {

**private** **static** String *driver*="com.mysql.jdbc.Driver";

**private** **static** String *url*="jdbc:mysql:///shoppingcart";

**private** **static** String *user*="root";

**private** **static** String *password*="admin";

**public** **static** Connection getCon() **throws** ClassNotFoundException, SQLException{

Class.*forName*(*driver*);//加载数据库驱动

System.***out***.println("测试加载数据库成功");

Connection con=DriverManager.*getConnection*(*url*, *user*, *password*);

System.***out***.println("测试数据库链接成功");

**return** con;

}

**public** **static** **void** close(Connection con,PreparedStatement ps,ResultSet rs){

**if**(rs!=**null**){//关闭资源，避免出现异常

**try** {

rs.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

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

**try** {

ps.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

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

**try** {

con.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

**public** **static** **boolean** addUpdateDelete(String sql,Object[] arr){

Connection con=**null**;

PreparedStatement ps=**null**;

**try** {

con=BaseDao.*getCon*();//第一步 ：连接数据库的操作

ps=con.prepareStatement(sql);//第二步：预编译

//第三步：设置值

**if**(arr!=**null** && arr.length!=0){

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

ps.setObject(i+1, arr[i]);

}

}

**int** count=ps.executeUpdate();//第四步：执行sql语句

**if**(count>0){

**return** **true**;

}**else**{

**return** **false**;

}

} **catch** (ClassNotFoundException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** **false**;

}

**public** **static** **void** main(String[] args) {

**try** {

BaseDao.*getCon*();

} **catch** (ClassNotFoundException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

book.jsp

<body>

<%

Book book=**new** Book();

BookServiceImpl service=**new** BookServiceImpl();

List<Book> list=service.select(book);

%>

<div style="text-align:*right*;font-size:*36px*;">

<a href=*"docart.jsp"*>我的购物车</a>

</div>

<table align=*"center"* width=*"100%"*>

<tr>

<th>编号</th>

<th>书名</th>

<th>价格</th>

<th>作者</th>

<th>封皮</th>

<th>出版社</th>

</tr>

<%

**for**(Book b:list){

%>

<tr align=*"center"*>

<td><%=b.getBookid() %></td>

<td><a href=*"dobook.jsp?id=*<%=b.getBookid()%>*"*><%=b.getBookname() %></a></td>

<td><%=b.getPrice() %></td>

<td><%=b.getAuthor() %></td>

<td><%=b.getPic() %></td>

<td><%=b.getPublish() %></td>

</tr>

<%} %>

</table>

</body>

Cart.jsp

<body>

<table width=*"100%"* align=*"center"* border=*"1px"*>

<tr>

<th>书本编号</th>

<th>书本名称</th>

<th>书本单价</th>

<th>书本数量</th>

<th>书本小计</th>

</tr>

<%

//1:将添加到购物车里面的物品显示出来

Map<Integer,CartItem> map=(Map<Integer,CartItem>)session.getAttribute("cart");

//2:将购物车里面的内容遍历出来

**double** count=0;//显示出总价格

**for**(Map.Entry<Integer,CartItem> entry : map.entrySet()){

//计算出每一样的书籍一共花了多少钱

**double** price=entry.getValue().getBook().getPrice() \* entry.getValue().getNumber();

//计算出一共花了多少钱

count=count+price;

%>

<tr align=*"center"*>

<td><%=entry.getKey() %></td>

<td><%=entry.getValue().getBook().getBookname() %></td>

<td><%=entry.getValue().getBook().getPrice() %></td>

<td><%=entry.getValue().getNumber() %></td>

<td><%=entry.getValue().getBook().getPrice() \* entry.getValue().getNumber()%></td>

</tr>

<%} %>

<tr>

<td colspan=*"4"* align=*"right"*>价格总计</td>

<td><%=count %></td>

</tr>

</table>

<div style="text-align:*center*;font-size:*36px*;">

<a href=*"book.jsp"*>图书列表</a>

</div>

</body>

Cart.jsp

<%@page import=*"com.bie.po.CartItem"*%>

<%@page import=*"java.util.Map"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>我的购物车的页面</title>

</head>

<body>

<table width=*"100%"* align=*"center"* border=*"1px"*>

<tr>

<th>书本编号</th>

<th>书本名称</th>

<th>书本单价</th>

<th>书本数量</th>

<th>书本小计</th>

</tr>

<%

//1:将添加到购物车里面的物品显示出来

Map<Integer,CartItem> map=(Map<Integer,CartItem>)session.getAttribute("cart");

//2:将购物车里面的内容遍历出来

**double** count=0;//显示出总价格

**for**(Map.Entry<Integer,CartItem> entry : map.entrySet()){

//计算出每一样的书籍一共花了多少钱

**double** price=entry.getValue().getBook().getPrice() \* entry.getValue().getNumber();

//计算出一共花了多少钱

count=count+price;

%>

<tr align=*"center"*>

<td><%=entry.getKey() %></td>

<td><%=entry.getValue().getBook().getBookname() %></td>

<td><%=entry.getValue().getBook().getPrice() %></td>

<td><%=entry.getValue().getNumber() %></td>

<td><%=entry.getValue().getBook().getPrice() \* entry.getValue().getNumber()%></td>

</tr>

<%} %>

<tr>

<td colspan=*"4"* align=*"right"*>价格总计</td>

<td><%=count %></td>

</tr>

</table>

<div style="text-align:*center*;font-size:*36px*;">

<a href=*"book.jsp"*>图书列表</a>

</div>

</body>

</html>

detail.jsp

<body>

<%

Book book=(Book)session.getAttribute("book");

%>

<div style="text-align:right;font-size:36px;">

<a href="docart.jsp">我的购物车</a>

</div>

<table align="center" cellpadding="20" cellspacing="20">

<tr>

<td>图书编号</td>

<td>图书名称</td>

<td>图书价格</td>

<td>图书作者</td>

<td>图书封皮</td>

<td>图书出版社</td>

</tr>

<tr>

<td><%=book.getBookid() %></td>

<td><%=book.getBookname() %></td>

<td><%=book.getPrice() %></td>

<td><%=book.getAuthor() %></td>

<td><img src="images/<%=book.getPic() %>"></td>

<td><%=book.getPublish() %></td>

</tr>

<tr>

<td colspan="2"></td>

<td><a href="cart.jsp">添加到购物车</a></td>

<td><a href="book.jsp">图书列表</a></td>

<td colspan="2"></td>

</tr>

</table>

</body>

dobook.jsp

<body>

<%

Book book=new Book();

String sid=request.getParameter("id");

Integer id=Integer.parseInt(sid);

BookService service=new BookServiceImpl();

book.setBookid(id);

Book books=service.getBook(book);

session.setAttribute("book", books);

response.sendRedirect("detail.jsp");

%>

</body>

docart.jsp

<body>

<%

Book book=(Book)session.getAttribute("book");

Map<Integer,CartItem> cart=(Map<Integer,CartItem>)session.getAttribute("cart");

if(cart==null){

cart=new HashMap<>();

}

CartItem item=(CartItem)cart.get(book.getBookid());

if(item==null){

item=new CartItem();

item.setBook(book);

item.setNumber(1);

}else{

item.setNumber(item.getNumber()+1);

}

cart.put(book.getBookid(),item);

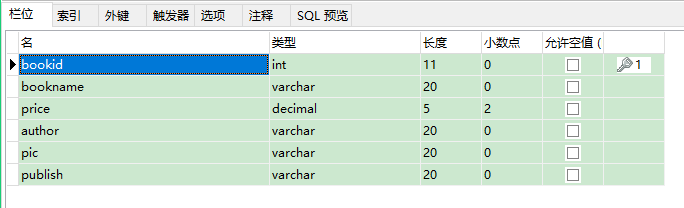
session.setAttribute("cart", cart);

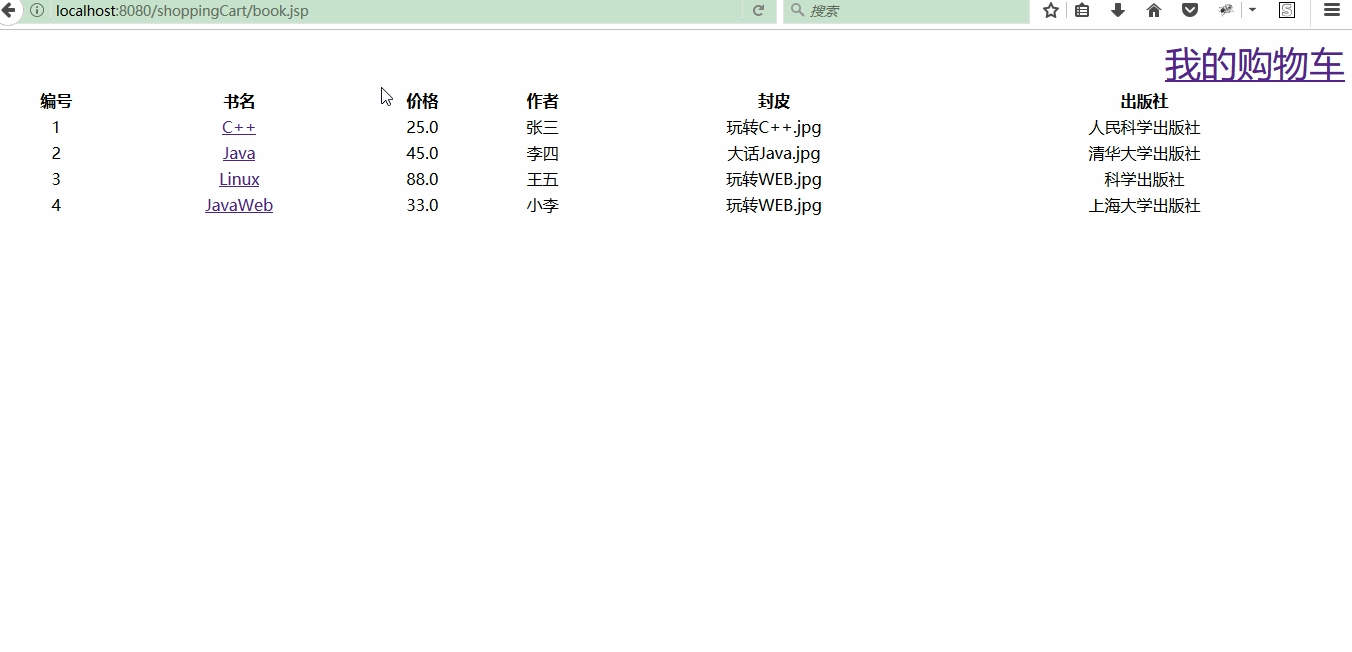
response.sendRedirect("book.jsp");

%>

</body>

# 运行效果图





# 收获与问题

理解mvc设计模式