**Java Web开发**

**实验六**

**作业报告**

|  |  |
| --- | --- |
| 学 院 | 计算机 |
| 所在系 班 级 | 2017软件工程系7班 |
| 学 号 姓 名 | 04171515 |
| 指 导 教 师 | 郭晓燕 |
| 完 成 时 间 | 2019年4月 |

# 实验题目

1、上周没有完成作业的同学请继续完成，完成了作业的同学请为购物车增加功能：

2、参考成熟的购物系统，为购物车增加功能，完善购物车：删除商品，查看商品具体信息，计算商品总价，修改商品购买件数等等。界面尽量美观。

提示：Java文件即Book.java的设计需要重构，需要分离商品基本信息和购买信息，需要有类分别来保存单件商品的购买信息和所有商品的购买信息。所以至少要三个Java类才能完成购物的基本功能。

Servlet也需要响应更多的操作，可以在一个Servlet中也可以分多个文件响应。并在适当的地方调用Java方法。

# 分析与设计

Book.java:数据类，用来保存图书的基本信息（ID、书名、价格、出版社、作者、ISBN、出版时间、书籍库存等）。

BookBuy.java: 数据类，用来保存一本书的购书信息（购书数量）。

BookCart.java:购物车类，保存用户的选购的书的数据并提供相关接口（增加、减少、删除、清空、计算数量、提供只读列表等）。

BookMarket.java:书市类，保存与提供能够被选购的书的信息。

ShoppingServlet.java:根据请求信息增删改采购书目，并重定向到购物车。

InitBookCartServlet.java:初始化用户数据。

cart.jsp:显示购物车。

shopping.jsp:购书页面，通过此页面用户选购图书。

info.jsp:书籍详细信息页。

fail.jsp:失败页面。

cart.js:响应用户的添加、减少、删除、设置数量、选择操作并发送对应请求，计算总价与格式化价格

# 关键源码（部分）

贴上这道题中最关键的部分源码。也是你实验练习的主要知识点相关的源码。并在代码旁附上详细的解释

1. InitBookCartServlet.java

**package** servlet;  
  
**import** entity.Book;  
**import** service.BookCart;  
  
**import** javax.servlet.ServletException;  
**import** javax.servlet.annotation.WebServlet;  
**import** javax.servlet.http.HttpServlet;  
**import** javax.servlet.http.HttpServletRequest;  
**import** javax.servlet.http.HttpServletResponse;  
**import** java.io.IOException;  
  
@WebServlet(name = **"InitBookCartServlet"**, urlPatterns = **"/initBookCart"**)  
**public class** InitBookCartServlet **extends** HttpServlet {  
 **protected void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  
 **if**(request.getSession().getAttribute(**"bookCart"**)==**null**) {  
 request.getSession().setAttribute(**"bookCart"**, **new** BookCart());  
 request.getSession().setAttribute(**"sum"**, 0);  
 }  
 }  
  
 **protected void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  
 doPost(request, response);  
 }  
}

1. Book.java

**package** entity;  
  
**public class** Book {  
 **private** String **id**;  
 **private** String **name**;  
 **private** String **imageUrl**;  
 **private float originalPrice**;  
 **private float price**;  
 **private int capacity**;  
 **private** String **isbn**;  
 **private** String **press**;  
 **private** String **author**;  
 **private** String **publicationDate**;  
  
 **public** Book() {  
 **this**.**id** = **"Unknown"**;  
 **this**.**name** = **"Unknown"**;  
 **this**.**imageUrl** = **"Unknown"**;  
 **this**.**originalPrice** = 0;  
 **this**.**price** = 0;  
 **this**.**capacity** = 0;  
 **this**.**isbn** = **"Unknown"**;  
 **this**.**press** = **"Unknown"**;  
 **this**.**author** = **"Unknown"**;  
 **this**.**publicationDate** = **"Unknown"**;  
 }  
  
 **public** Book(String id, String name, String imageUrl, **float** originalPrice, **float** price, **int** capacity, String isbn, String press, String author, String publicationDate) {  
 **this**.**id** = id;  
 **this**.**name** = name;  
 **this**.**imageUrl** = imageUrl;  
 **this**.**originalPrice** = originalPrice;  
 **this**.**price** = price;  
 **this**.**capacity** = capacity;  
 **this**.**isbn** = isbn;  
 **this**.**press** = press;  
 **this**.**author** = author;  
 **this**.**publicationDate** = publicationDate;  
 }  
  
 **public** String getId() {  
 **return id**;  
 }  
  
 **public void** setId(String id) {  
 **this**.**id** = id;  
 }  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public void** setName(String name) {  
 **this**.**name** = name;  
 }  
  
 **public** String getImageUrl() {  
 **return imageUrl**;  
 }  
  
 **public void** setImageUrl(String imageUrl) {  
 **this**.**imageUrl** = imageUrl;  
 }  
  
 **public float** getOriginalPrice() {  
 **return originalPrice**;  
 }  
  
 **public void** setOriginalPrice(**float** originalPrice) {  
 **this**.**originalPrice** = originalPrice;  
 }  
  
 **public float** getPrice() {  
 **return price**;  
 }  
  
 **public void** setPrice(**float** price) {  
 **this**.**price** = price;  
 }  
  
 **public** String getIsbn() {  
 **return isbn**;  
 }  
  
 **public void** setIsbn(String isbn) {  
 **this**.**isbn** = isbn;  
 }  
  
 **public** String getPress() {  
 **return press**;  
 }  
  
 **public void** setPress(String press) {  
 **this**.**press** = press;  
 }  
  
 **public** String getAuthor() {  
 **return author**;  
 }  
  
 **public void** setAuthor(String author) {  
 **this**.**author** = author;  
 }  
  
 **public** String getPublicationDate() {  
 **return publicationDate**;  
 }  
  
 **public void** setPublicationDate(String publicationDate) {  
 **this**.**publicationDate** = publicationDate;  
 }  
  
 **public int** getCapacity() {  
 **return capacity**;  
 }  
  
 **public void** setCapacity(**int** capacity) {  
 **this**.**capacity** = capacity;  
 }  
  
 **synchronized public void** addOne() {  
 **this**.**capacity**++;  
 }  
 **synchronized public void** subOne() {  
 **this**.**capacity**--;  
 }  
  
}

1. BookBuy.java

**package** entity;  
  
**import** service.BookMarket;  
  
**public class** BookBuy {  
 **private** String **id**;  
 **private int number**;  
  
 **public** BookBuy(String id, **int** number) {  
 **this**.**id** = id;  
 **this**.**number** = number;  
 }  
  
 **public** String getId() {  
 **return id**;  
 }  
  
 **public int** getNumber() {  
 **return number**;  
 }  
  
 **public void** setNumber(**int** number) {  
 **this**.**number** = number;  
 }  
  
 **public void** add(**int** num) {  
 **number** += num;  
 }  
  
 **public** Book getBook() {  
 **return** BookMarket.*getInstance*().getBook(**id**);  
 }  
  
}

1. BookCart.java

**package** service;  
  
**import** entity.\*;  
  
**import** java.util.Collections;  
**import** java.util.LinkedHashMap;  
**import** java.util.Map;  
  
**public class** BookCart {  
 Map<String, BookBuy> **books**;  
 **public** BookCart() {  
 **books** = **new** LinkedHashMap<>();  
 }  
 **public** Map<String, BookBuy> getBooks() {  
 **return** Collections.*unmodifiableMap*(**books**);  
 }  
 **public boolean** add(String id) {  
 **return** add(id, 1);  
 }  
 **public boolean** add(String id, **int** num) {  
 BookBuy bb = **books**.getOrDefault(id, **new** BookBuy(id, 0));  
 **int** n = bb.getNumber()+num;  
 **return** set(id, n);  
 }  
 **public boolean** sub(String id) {  
 **return** sub(id, 1);  
 }  
 **public boolean** sub(String id, **int** num) {  
 **return** add(id, -num);  
 }  
 **public void** clear() {  
 **books**.clear();  
 }  
 **public boolean** set(String id, **int** num) {  
 Book b = BookMarket.*getInstance*().getBook(id);  
 **if**(b == **null**)  
 **return false**;  
 **if**(num > b.getCapacity() || num < 0)  
 **return false**;  
 BookBuy bb = **books**.getOrDefault(id, **new** BookBuy(id, 0));  
 bb.setNumber(num);  
 **if**(num == 0)  
 **books**.remove(id);  
 **else  
 books**.put(id, bb);  
 **return true**;  
 }  
 **public boolean** remove(String id) {  
 **if**(!**books**.containsKey(id))  
 **return false**;  
 **books**.remove(id);  
 **return true**;  
 }  
 **public int** sum() {  
 **int** count = 0;  
 **for** (BookBuy bb:**books**.values()) {  
 count += bb.getNumber();  
 }  
 **return** count;  
 }  
 **public float** sumPrice() {  
 **float** sum = 0;  
 **for** (BookBuy bb:**books**.values()) {  
 Book b = BookMarket.*getInstance*().getBook(bb.getId());  
 **if**(b == **null**)  
 **continue**;  
 sum += bb.getNumber()\*b.getPrice();  
 }  
 **return** sum;  
 }  
}

1. BookMarket.java

**package** service;  
  
**import** entity.Book;  
  
**import** java.util.Collections;  
**import** java.util.LinkedHashMap;  
**import** java.util.Map;  
  
**public class** BookMarket {  
 **private static** BookMarket *instance* = **new** BookMarket();  
 Map<String, Book> **books** = **new** LinkedHashMap<>();  
  
 **private** BookMarket() {  
 addBook(**new** Book(**"10101"**, **"Java核心技术"**, **"img/10101.jpg"**, 258.0f, 207.7f, 100, **"9787111547426/9787111573319"**, **"机械工业出版社"**, **"[美] 凯 S.霍斯特曼（Cay S.Horstmann）"**, **"2016-09-01"**));  
 addBook(**new** Book(**"10102"**, **"Java经典实例"**, **"img/10102.jpg"**, 148.0f, 111.0f, 100, **"9787512387751"**, **"中国电力出版社"**, **"[美] Ian F.Darwin（达尔文） 著，李新叶，余晓晔 译"**, **"2016-08-01"**));  
 addBook(**new** Book(**"10103"**, **"Java入门经典(第7版)"**, **"img/10103.jpg"**, 49.0f, 38.7f, 100, **"9787115400369"**, **"人民邮电出版社"**, **"[美] 罗格斯·卡登海德（Rogers Cadenhead） 著，郝记生 译"**, **"2015-09-01"**));  
 addBook(**new** Book(**"10104"**, **"JSP & Servlet学习笔记（第2版 全新改版 附光盘）"**, **"img/10104.jpg"**, 78.0f, 69.6f, 100, **"9787302283669"**, **"清华大学出版社"**, **"林信良"**, **"2012-05-01"**));  
 }  
  
 **public static** BookMarket getInstance() {  
 **return** *instance*;  
 }  
  
 **public** Map<String, Book> getBooks() {  
 **return** Collections.*unmodifiableMap*(**books**);  
 }  
  
 **public boolean** addBook(Book book) {  
 **if** (book == **null** || book.getId().trim().isEmpty())  
 **return false**;  
 **if** (**books**.containsKey(book.getId()))  
 **return false**;  
 **books**.put(book.getId(), book);  
 **return true**;  
 }  
  
 **public boolean** removeBook(String id) {  
 **if** (!**books**.containsKey(id))  
 **return false**;  
 **books**.remove(id);  
 **return true**;  
 }  
  
 **public void** setBook(Book book) {  
 **books**.put(book.getId(), book);  
 }  
  
 **public** Book getBook(String id) {  
 **return books**.getOrDefault(id, **null**);  
 }  
  
 **public void** clearBooks() {  
 **books**.clear();  
 }  
}

1. ShoppingServlet.java

**package** servlet;  
  
**import** service.BookCart;  
  
**import** javax.servlet.ServletException;  
**import** javax.servlet.annotation.WebServlet;  
**import** javax.servlet.http.HttpServlet;  
**import** javax.servlet.http.HttpServletRequest;  
**import** javax.servlet.http.HttpServletResponse;  
**import** javax.servlet.http.HttpSession;  
**import** java.io.IOException;  
  
@WebServlet(name = **"ShoppingServlet"**, urlPatterns = **"/shopping.view"**)  
**public class** ShoppingServlet **extends** HttpServlet {  
 **protected void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  
 doGet(request, response);  
 }  
  
 **protected void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  
 request.getRequestDispatcher(**"/initBookCart"**).include(request, response);  
 String method = request.getParameter(**"method"**);  
 **if** (method == **null** || method.trim().isEmpty())  
 method = **"add"**;  
 method = method.trim();  
 String id = request.getParameter(**"id"**);  
 **if** (id == **null**)  
 id = **""**;  
 id = id.trim();  
 String numStr = request.getParameter(**"num"**);  
 **int** num;  
 **if** (numStr == **null** || numStr.trim().isEmpty())  
 num = 1;  
 **else** num = Integer.*valueOf*(numStr.trim());  
  
 **if** (num < 0) {  
 request.getRequestDispatcher(**"fail.jsp"**).forward(request, response);  
 **return**;  
 }  
  
 HttpSession session = request.getSession();  
 BookCart bookCart = (BookCart) session.getAttribute(**"bookCart"**);  
 **boolean** ret = **false**;  
 **switch** (method) {  
 **case "add"**:  
 **if** (!id.isEmpty())  
 ret = bookCart.add(id, num);  
 **break**;  
 **case "sub"**:  
 **if** (!id.isEmpty())  
 ret = bookCart.sub(id, num);  
 **break**;  
 **case "set"**:  
 **if** (!id.isEmpty())  
 ret = bookCart.set(id, num);  
 **break**;  
 **case "clear"**:  
 bookCart.clear();  
 ret = **true**;  
 **break**;  
 **case "remove"**:  
 **if** (!id.isEmpty())  
 ret = bookCart.remove(id);  
 **break**;  
 **default**:  
 ret = **false**;  
 }  
 **if** (!ret) {  
 request.getRequestDispatcher(**"fail.jsp"**).forward(request, response);  
 **return**;  
 }  
 session.setAttribute(**"bookList"**, bookCart);  
 session.setAttribute(**"sum"**, bookCart.sum());  
 session.setAttribute(**"sumPrice"**, bookCart.sum());  
 response.sendRedirect(**"cart.jsp"**);  
 }  
}

1. shopping.jsp

<%@ **page import**="**entity.Book**" %>  
<%@ **page import**="**service.\***" %>  
<%@ **page import**="**java.util.Map**" %>  
<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %>  
<**html**>  
**<%** request.getRequestDispatcher(**"/initBookCart"**).include(request, response);  
**%>**<**head**>  
 <**title**>书籍清单</**title**>  
 <**meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"**>  
 <**script type="text/javascript" src="js/jquery-3.3.1.min.js"**></**script**>  
 <**script type="text/javascript" src="js/bootstrap.min.js"**></**script**>  
 <**link rel="stylesheet" type="text/css" href="css/bootstrap.min.css"**/>  
</**head**>  
<**body**>  
<**nav class="navbar navbar-default" role="navigation"**>  
 <**div class="container-fluid"**>  
 <**div class="navbar-header"**>  
 <**a class="navbar-brand" href="#"**>书籍Shop</**a**>  
 </**div**>  
 <**ul class="nav navbar-nav navbar-right"**>  
 <**li**><**a href="#"**><**img src="img/list.png" height="25px" width="25px"**>书籍清单</**a**></**li**>  
 <**li**><**a href="cart.jsp"**><**img src="img/cart.png" height="25px" width="25px"**>查看购物车( **<%=**((BookCart)session.getAttribute(**"bookCart"**)).sum()**%>** 本)</**a**></**li**>  
 </**ul**>  
 </**div**>  
</**nav**>  
<**table class='table text-center table-hover'**>  
 **<%** Map<String, Book> bookList= BookMarket.*getInstance*().getBooks();  
 **for**(Book book:bookList.values()){  
 **%>** <**tr**>  
 <**td**><**img src='<%=**book.getImageUrl()**%>' height='auto' width='100px'**></**td**>  
 <**td**><**a href='info.jsp?id=<%=**book.getId()**%>'**>**<%=**book.getName()**%>**</**a**></**td**>  
 <**td**><**a href='shopping.view?id=<%=**book.getId()**%>' class='btn btn-info'**>采购此书</**a**></**td**>  
 </**tr**>  
 **<%** }  
 **%>**</**table**>  
</**body**>  
</**html**>

1. info.jsp

<%@ **page import**="**service.\***" %>  
<%@ **page import**="**entity.\***" %>  
<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %>  
<**html**>  
**<%** request.getRequestDispatcher(**"/initBookCart"**).include(request, response);  
 Book book = BookMarket.*getInstance*().getBook(request.getParameter(**"id"**));  
 **if**(book == **null**) {  
 request.getRequestDispatcher(**"/fail.jsp"**).forward(request, response);  
 **return**;  
 }  
**%>**<**head**>  
 <**title**>书籍详情 - **<%=**book.getName()**%>**></**title**>  
 <**meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"**>  
 <**script type="text/javascript" src="js/jquery-3.3.1.min.js"**></**script**>  
 <**script type="text/javascript" src="js/bootstrap.min.js"**></**script**>  
 <**link rel="stylesheet" type="text/css" href="css/bootstrap.min.css"**/>  
</**head**>  
<**body**>  
<**nav class="navbar navbar-default" role="navigation"**>  
 <**div class="container-fluid"**>  
 <**div class="navbar-header"**>  
 <**a class="navbar-brand" href="#"**>书籍Shop</**a**>  
 </**div**>  
 <**ul class="nav navbar-nav navbar-right"**>  
 <**li**><**a href="shopping.jsp"**><**img src="img/list.png" height="25px" width="25px"**>书籍清单</**a**></**li**>  
 <**li**><**a href="cart.jsp"**><**img src="img/cart.png" height="25px" width="25px"**>查看购物车( **<%=**((BookCart)session.getAttribute(**"bookCart"**)).sum()**%>** 本)</**a**></**li**>

本)</**a**></**li**>  
 </**ul**>  
 </**div**>  
</**nav**>  
<**div class="container"**>  
 <**div class="row"**>  
 <**div class="col-sm-4"**>  
 <**img src="<%=**book.getImageUrl()**%>"**>  
 <**p class='text-center center-block'**>**<%=**book.getName()**%>**</**p**>  
 <**a href='shopping.view?id=<%=**book.getId()**%>' class='btn btn-info text-center center-block'**>采购此书</**a**>  
 <**hr class="hidden-sm hidden-md hidden-lg"**>  
 </**div**>  
 <**div class="col-sm-8"**>  
 <**h2**><**strong**>**<%=**book.getName()**%>**</**strong**></**h2**>  
 <**h5 style="padding-left**:2**rem**;**"**>**<%=**book.getAuthor()**%>**</**h5**>  
 <**table class='table text-center'**>  
 <**tbody**>  
 <**tr**>  
 <**td**>价格</**td**>  
 <**td**>**<%=**book.getPrice()**%>**<**s class="text-muted"**>(原价:**<%=**book.getOriginalPrice()**%>**)</**s**></**td**>  
 </**tr**>  
 <**tr**>  
 <**td**>出版社</**td**>  
 <**td**>**<%=**book.getPress()**%>**</**td**>  
 </**tr**>  
 <**tr**>  
 <**td**>ISBN</**td**>  
 <**td**>**<%=**book.getIsbn()**%>**</**td**>  
 </**tr**>  
 <**tr**>  
 <**td**>出版时间</**td**>  
 <**td**>**<%=**book.getPublicationDate()**%>**</**td**>  
 </**tr**>  
 <**tr**>  
 <**td**>书籍余量</**td**>  
 <**td**>**<%=**book.getCapacity()**%>**</**td**>  
 </**tr**>  
 </**tbody**>  
 </**table**>  
 </**div**>  
 </**div**>  
</**div**>  
  
</**body**>  
</**html**>

1. fail.jsp

<%@ **page import**="**entity.Book**" %>  
<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %>  
<**html**>  
**<%** request.getRequestDispatcher(**"/initBookCart"**).include(request, response);  
 response.sendError(405, **"Operator Fail!!!"** );  
**%>**<**head**>  
 <**title**>操作失败</**title**>  
 <**link rel="stylesheet" type="text/css" href="css/bootstrap.min.css"**/>  
 <**script type="text/javascript" src="js/jquery-3.3.1.min.js"**></**script**>  
 <**script type="text/javascript" src="js/bootstrap.min.js"**></**script**>  
</**head**>  
<**body**>  
<**nav class="navbar navbar-default" role="navigation"**>  
 <**div class="container-fluid"**>  
 <**div class="navbar-header"**>  
 <**a class="navbar-brand" href="#"**>书籍Shop</**a**>  
 </**div**>  
 <**ul class="nav navbar-nav navbar-right"**>  
 <**li**><**a href="shopping.jsp"**><**img src="img/list.png" height="25px" width="25px"**>书籍清单</**a**></**li**>  
 <**li**><**a href="cart.jsp"**><**img src="img/cart.png" height="25px" width="25px"**>查看购物车</**a**></**li**>  
 </**ul**>  
 </**div**>  
</**nav**>  
<**div style="padding**:10**px**;**margin**:10**px"**>  
 <**h2 class="text-center"**>操作失败</**h2**>  
</**div**>  
</**body**>  
</**html**>

1. cart.js

**$**(***document***).**ready**(**function** () {  
  
 */\*  
 \* 计算购物车中每一个产品行的金额小计  
 \*  
 \* 参数 row 购物车表格中的行元素tr  
 \*  
 \*/* **function** *getSubTotal*(row) {  
 **var** price = *parseFloat*(**$**(row).find(**".selling-price"**).data(**"bind"**));  
 **var** qty = *parseInt*(**$**(row).find(**":text"**).val());  
 **var** result = price \* qty;  
 **$**.**ajax**({  
 **url**: **"shopping.view?method=set&id="** + **$**(row).find(**".book-id"**).data(**"id"**) + **"&num="** + qty,  
 **async**: **false**,  
 success: **function** (data, status) {  
 **if** (status != **"success"**) {  
 *alert*(**"修改失败"**);  
 ***window***.**location**.reload();  
 }  
 }  
 });  
 **$**(row).find(**".selling-price"**).**text**(**$**.formatMoney(price, 2));  
 **$**(row).find(**".subtotal"**).**text**(**$**.formatMoney(result, 2)).data(**"bind"**, result.toFixed(2));  
 }  
 */\*  
 \* 计算购物车中产品的累计金额  
 \*/* **function** *getTotal*() {  
 **var** qtyTotal = 0;  
 **var** itemCount = 0;  
 **var** priceTotal = 0;  
 **$**(cartTable).find(**"tr:gt(0)"**).each(**function** () {  
 *getSubTotal*(**this**);  
 **if** (**$**(**this**).find(**":checkbox"**).prop(**"checked"**) == **true**) {  
 itemCount++;  
 qtyTotal += *parseInt*(**$**(**this**).find(**":text"**).val());  
 priceTotal += *parseFloat*(**$**(**this**).find(**".subtotal"**).data(**"bind"**));  
 }  
 });  
 **$**(**"#itemCount"**).**text**(itemCount).data(**"bind"**, itemCount);  
 **$**(**"#qtyCount"**).**text**(qtyTotal).data(**"bind"**, qtyTotal);  
 **$**(**"#priceTotal"**).**text**(**$**.formatMoney(priceTotal, 2)).data(**"bind"**, priceTotal.toFixed(2));  
 }  
 **var** cartTable = **$**(**"#cartTable"**);  
  
 *getTotal*();  
  
 *//为每一个勾选框指定单击事件* **$**(cartTable).find(**":checkbox"**).**click**(**function** () {  
 *//全选框单击* **if** (**$**(**this**).hasClass(**"check-all"**)) {  
 **var** checked = **$**(**this**).prop(**"checked"**);  
 **$**(cartTable).find(**".check-one"**).prop(**"checked"**, checked);  
 }  
  
 *//如果手工一个一个的点选了所有勾选框，需要自动将“全选”勾上或是取消* **var** items = cartTable.find(**"tr:gt(0)"**);  
 **$**(cartTable).find(**".check-all"**).prop(**"checked"**, items.find(**":checkbox:checked"**).**length** == items.**length**);  
 *//设置结算按钮disabled属性* **$**(**"#btn\_settlement"**).attr(**"disabled"**, items.find(**":checkbox:checked"**).**length** == 0);  
  
 *getTotal*();  
 });  
  
 *//为数量调整的＋ －号提供单击事件，并重新计算产品小计  
 /\*  
 \* 为购物车中每一行绑定单击事件，以及每行中的输入框绑定键盘事件  
 \* 根据触发事件的元素执行不同动作  
 \* 增加数量  
 \* 减少数量  
 \* 删除产品  
 \*  
 \*/* **$**(cartTable).find(**"tr:gt(0)"**).each(**function** () {  
 **var** input = **$**(**this**).find(**":text"**);  
  
 *//为数量输入框添加事件，计算金额小计，并更新总计* **$**(input).keyup(**function** () {  
 **var** val = *parseInt*(**$**(**this**).val());  
 **if** (*isNaN*(val) || (val < 1)) {  
 **$**(**this**).val(**"1"**);  
 }  
 *getSubTotal*(**$**(**this**).**parent**().**parent**()); *//tr element  
 getTotal*();  
 });  
  
 *//为数量调整按钮、删除添加单击事件，计算金额小计，并更新总计* **$**(**this**).click(**function** () {  
 **var** val = *parseInt*(**$**(input).val());  
 **if** (*isNaN*(val) || (val < 1)) {  
 val = 1;  
 }  
  
 **if** (**$**(***window***.**event**.**srcElement**).hasClass(**"minus"**)) {  
 **if** (val <= 1)  
 *alert*(**"删除请点删除按钮"**);  
 **else** {  
 **if** (val > 1) val--;  
 input.val(val);  
 *getSubTotal*(**this**);  
 }  
 } **else if** (**$**(***window***.**event**.**srcElement**).hasClass(**"plus"**)) {  
 **if** (val < 9999) val++;  
 input.val(val);  
 *getSubTotal*(**this**);  
 } **else if** (**$**(***window***.**event**.**srcElement**).hasClass(**"delete"**)) {  
 **var** t = **$**(**this**);  
 **if** (*confirm*(**"确定要从购物车中删除此书籍？"**)) {  
 **$**.**ajax**({  
 **url**: **"shopping.view?method=remove&id="** + **$**(**this**).find(**".book-id"**).data(**"id"**),  
 **async**: **false**,  
 success: **function** (data, status) {  
 **if** (status == **"success"**)  
 t.remove();  
 **else** *alert*(**"删除失败"**);  
 }  
 });  
 }  
 }  
 *getTotal*();  
 });  
 });  
});  
  
**function** *supporthHtml5*() {  
 **return** (**typeof** (***Worker***) !== **"undefined"**) ? **true** : **false**;  
}  
  
**$**(***document***).**ready**(**function** () {  
 **jQuery**.ajaxSetup({**cache**: **false**});  
});  
  
*//select all items***function** *selectAll*() {  
 **var** aChecked = **$**(**".select\_all"**).prop(**"checked"**);  
 **if** (**typeof** (aChecked) == **"undefined"**) {  
 aChecked = **false**;  
 }  
 **$**(**".selectable"**).each(**function** () {  
 **if** (**typeof** (**$**(**this**).attr(**"disabled"**)) == **"undefined"**) {  
 **var** subchecked = **$**(**this**).prop(**"checked"**);  
 **if** (subchecked != aChecked) {  
 **$**(**this**).prop(**"checked"**, aChecked);  
 }  
 }  
 });  
}  
  
**function** *EnsureDecimal*(e) {  
 e.**value** = e.**value**.replace(/[^\d.]/g, **""**); *//先把非数字的都替换掉，除了数字和.* e.**value** = e.**value**.replace(/^\./g, **""**); *//必须保证第一个为数字而不是.* e.**value** = e.**value**.replace(/\.{2,}/g, **"."**); *//保证只有出现一个.而没有多个.* e.**value** = e.**value**.replace(**"."**, **"$#$"**).replace(/\./g, **""**).replace(**"$#$"**, **"."**); *//保证.只出现一次，而不能出现两次以上*}  
  
**function** *EnsureInt*() {  
 **if** (((***event***.**keyCode** < 48) || (***event***.**keyCode** > 57))) {  
 ***event***.**returnValue** = **false**;  
 }  
}  
  
*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HTML5 Notification Functions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
/\*\*  
 \* 金额按千位逗号分割  
 \* Example  
 \* <code>  
 \* alert($.formatMoney(1234.345, 2)); //=>1,234.35  
 \* alert($.formatMoney(-1234.345, 2)); //=>-1,234.35  
 \* alert($.unformatMoney(1,234.345)); //=>1234.35  
 \* alert($.unformatMoney(-1,234.345)); //=>-1234.35  
 \* </code>  
 \*/*!(**function** ($) {  
 $.extend({  
 */\*\*  
 \* 数字千分位格式化  
 \** ***@public*** *\** ***@param*** *mixed mVal 数值  
 \** ***@param*** *int iAccuracy 小数位精度(默认为2)  
 \** ***@return*** *string  
 \*/* formatMoney: **function** (mVal, iAccuracy) {  
 **var** fTmp = 0.00;*//临时变量* **var** iFra = 0;*//小数部分* **var** iInt = 0;*//整数部分* **var** aBuf = []; *//输出缓存* **var** bPositive = **true**; *//保存正负值标记(true:正数)  
 /\*\*  
 \* 输出定长字符串，不够补0  
 \* <li>闭包函数</li>  
 \** ***@param*** *int iVal 值  
 \** ***@param*** *int iLen 输出的长度  
 \*/* **function** *funZero*(iVal, iLen) {  
 **var** sTmp = iVal.toString();  
 **var** sBuf = [];  
 **for** (**var** i = 0, iLoop = iLen - sTmp.**length**; i < iLoop; i++)  
 sBuf.push(**'0'**);  
 sBuf.push(sTmp);  
 **return** sBuf.join(**''**);  
 }  
 **if** (**typeof** (iAccuracy) === **'undefined'**)  
 iAccuracy = 2;  
 bPositive = (mVal >= 0);*//取出正负号* fTmp = (*isNaN*(fTmp = *parseFloat*(mVal))) ? 0 : ***Math***.abs(fTmp);*//强制转换为绝对值数浮点  
 //所有内容用正数规则处理* iInt = *parseInt*(fTmp); *//分离整数部分* iFra = *parseInt*((fTmp - iInt) \* ***Math***.pow(10, iAccuracy) + 0.5); *//分离小数部分(四舍五入)* **do** {  
 aBuf.unshift(*funZero*(iInt % 1000, 3));  
 } **while** ((iInt = *parseInt*(iInt / 1000)));  
 aBuf[0] = *parseInt*(aBuf[0]).toString();*//最高段区去掉前导0* **return** ((bPositive) ? **''** : **'-'**) + aBuf.join(**','**) + **'.'** + ((0 === iFra) ? **'00'** : *funZero*(iFra, iAccuracy));  
 },  
 */\*\*  
 \* 将千分位格式的数字字符串转换为浮点数  
 \** ***@public*** *\** ***@param*** *string sVal 数值字符串  
 \** ***@return*** *float  
 \*/* unformatMoney: **function** (sVal) {  
 **var** fTmp = *parseFloat*(sVal.replace(/,/g, **''**));  
 **return** (*isNaN*(fTmp) ? 0 : fTmp);  
 },  
 });  
})(**jQuery**);

1. cart.jsp

<%@ **page import**="**entity.\***" %>  
<%@ **page import**="**service.\***" %>  
<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" **pageEncoding**="**UTF-8**" %>  
<**html**>  
**<%** request.getRequestDispatcher(**"/initBookCart"**).include(request, response);  
**%>**<**head**>  
 <**title**>购物车</**title**>  
 <**meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"**>  
 <**script type="text/javascript" src="js/jquery-3.3.1.min.js"**></**script**>  
 <**script type="text/javascript" src="js/bootstrap.min.js"**></**script**>  
 <**script type="text/javascript" src="js/cart.js"**></**script**>  
 <**link rel="stylesheet" type="text/css" href="css/bootstrap.min.css"**/>  
 <**link rel="stylesheet" type="text/css" href="css/cart.css"** />  
</**head**>  
<**body**>  
<**nav class="navbar navbar-default" role="navigation"**>  
 <**div class="container-fluid"**>  
 <**div class="navbar-header"**>  
 <**a class="navbar-brand" href="#"**>书籍Shop</**a**>  
 </**div**>  
 <**ul class="nav navbar-nav navbar-right"**>  
 <**li**><**a href="shopping.jsp"**><**img src="img/list.png" height="25px" width="25px"**>书籍清单</**a**></**li**>  
 <**li**><**a href="#"**><**img src="img/cart.png" height="25px" width="25px"**>查看购物车</**a**></**li**>  
 </**ul**>  
 </**div**>  
</**nav**>  
<**div class="container cart-wrap"**>  
 <**table id="cartTable" class="cart table table-condensed"**>  
 <**thead**>  
 <**tr**>  
 <**th class="t-checkbox"**><**label**><**input class="check-all check" type="checkbox"** />**&nbsp;**全选</**label**></**th**>  
 <**th class="text-center"**><**label**></**label**></**th**>  
 <**th class="t-goods text-center"**><**label**>书名</**label**></**th**>  
 <**th class="t-selling-price text-center"**><**label**>单价</**label**></**th**>  
 <**th class="t-qty text-center"**><**label**>采购数量</**label**></**th**>  
 <**th class="t-subtotal text-center"**><**label**>金额小计</**label**></**th**>  
 <**th class="t-action"**><**label**>操作</**label**></**th**>  
 </**tr**>  
 </**thead**>  
 <**tbody**>  
 **<%** BookCart bookCart= (BookCart)session.getAttribute(**"bookCart"**);  
 **boolean** isEmpty = **true**;  
 **for**(BookBuy bb:bookCart.getBooks().values()){  
 Book book = bb.getBook();  
 **if**(book == **null**)  
 **continue**;  
 isEmpty = **false**;  
 **%>** <**tr**>  
 <**td colspan="2" class="goods"**><**label**> <**input type="checkbox" class="check-one check"**/><**img src='<%=**book.getImageUrl()**%>' height='auto' width='100px'**></**label**></**td**>  
 <**td class="goods text-center book-id" data-id="<%=**book.getId()**%>"**><**label**><**a href='info.jsp?id=<%=**book.getId()**%>'**>**<%=**book.getName()**%>**</**a**></**label**></**td**>  
 <**td class="selling-price number small-bold-red text-center"  
 style="padding-top**: 1.1**rem**;**" data-bind="<%=**book.getPrice()**%>"**>**<%=**book.getPrice()**%>**</**td**>  
 <**td class="t-qty"**>  
 <**div class="input-group input-group-sm"**>  
 <**span class="input-group-addon minus"**>-</**span**> <**input  
 type="text" class="number form-control input-sm" value="<%=**bb.getNumber()**%>"** />  
 <**span class="input-group-addon plus"**>+</**span**>  
 </**div**>  
 </**td**>  
 <**td class="subtotal number small-bold-red text-right" style="padding-top**: 1.1**rem**;**"**></**td**>  
 <**td class="action" style="padding-top**: 1.1**rem**;**"**><**span class="delete btn btn-xs btn-warning"**>删除</**span**></**td**>  
 </**tr**>  
 **<%** }  
 **%>** </**tbody**>  
 </**table**>  
  
 **<%  
 if**(isEmpty) {  
 out.println(**"<p class=\"text-center\">购物车为空<p>"**);  
 }  
 **%>** <**div class="row"**>  
 <**div class="col-md-12 col-lg-12 col-sm-12"**>  
 <**div class="cart-summary"**>  
 <**div style="margin-left**: 2**rem**;**" class="pull-right"**>  
 <**a href="#"  
 id="btn\_settlement" type="button" class="btn btn-primary" disabled**>去结算</**a**>  
 </**div**>  
 <**div style="margin-left**: 1**rem**; **margin-top**: 0.4**rem**;**" class="pull-right total"**>  
 <**label**>金额合计:<**span id="priceTotal" class="price-total large-bold-red"**>0.00</**span**></**label**>  
 </**div**>  
 <**div style="margin-top**: 4**px**;**" class="pull-right"**>  
 <**label**>您选择了<**span id="itemCount" class="large-bold-red" style="margin**: 0 4**px**;**"**></**span**>种书，共计<**span id="qtyCount" class="large-bold-red" style="margin**: 0 4**px**;**"**></**span**>本  
 </**label**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
</**div**>  
</**body**>  
</**html**>

# 运行效果图

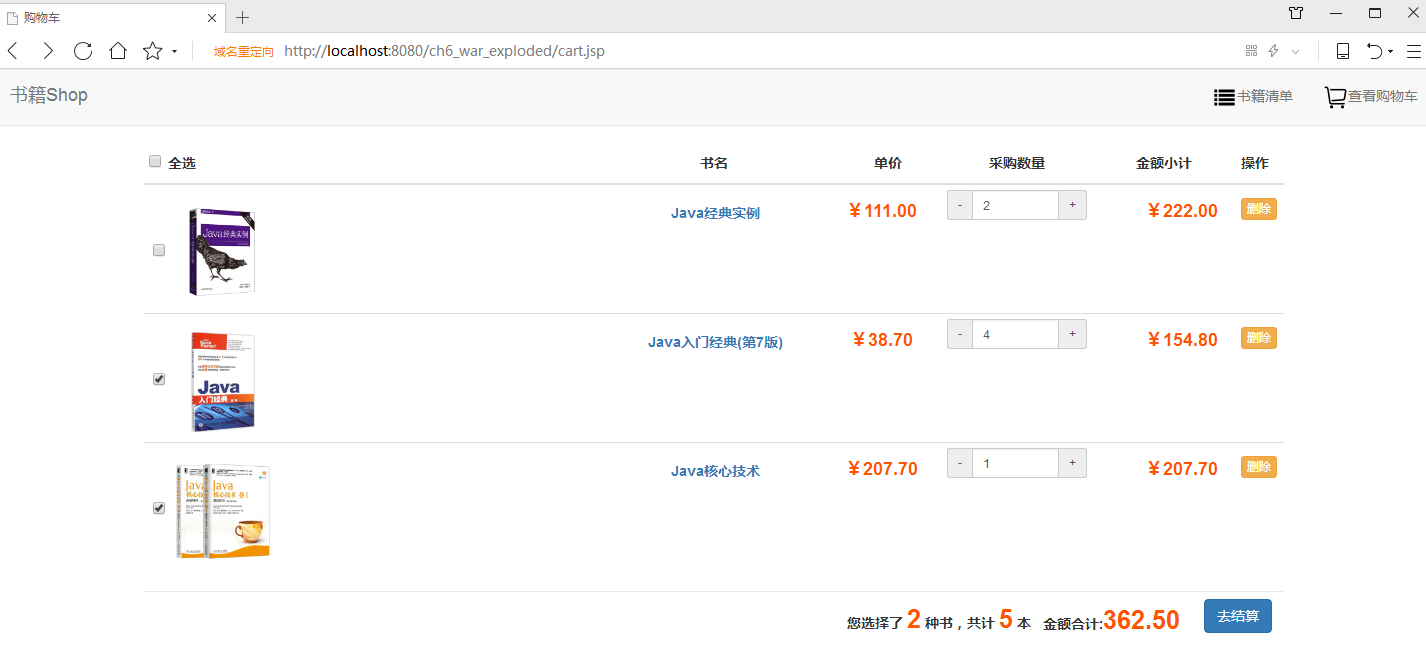
数据清单页面



数据详情页面



购物车页面



# 收获与问题