**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10班 |
| 学 号 姓 名 | 04172104欧建杰 |
| 指 导 教 师 | 郭晓燕 |
| 完 成 时 间 | 2018年5月- |

# 实验题目

实验六 会话管理-购物车（选做）

实验目标：

1、 学习更好的设计Java类。从而灵活使用session对象；

2、 完善购物车的功能；

3、 优化购物车的结构。

实验内容：

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

有以下文件: shopping.jsp cart.jsp buy.jsp

Shopping.jsp主要负责登录界面

Buy.jsp主要负责是购买时的处理页面

Cart.jsp负责购物车信息页面

Detail.jsp 负责展示详细细节

本来打算用.java文件的,不过因为有用bootstrap所以用.java文件out.print一行一行太麻烦了.

# 关键源码（部分）

Shopping.jsp

<%

//当第一次访问采购网页，session对象中还没有Stuff购物车信息时，进行初始化。

if (session.getAttribute("Stuff")==null){

//在session对象中存储图书采购信息

Stuff[] StuffList = new Stuff[3];

StuffList[0] = new Stuff("10101", "维他原味豆奶", "../img/10101.jpg", 0,20);

StuffList[1] = new Stuff("10102", "维他巧克力豆奶", "../img/10102.jpg", 0,30);

StuffList[2] = new Stuff("10103", "维他柠檬茶", "../img/10103.jpg", 0,40);

int sum=0;

//存入session对象中

List list = new ArrayList();

for(int i=0; i<StuffList.length; i++)

{list.add(StuffList[i]);}

session.setAttribute("Stuff", StuffList);

session.setAttribute("list", list);

session.setAttribute("sum", sum);

}

%>

<body>

<!-- 第一部分 -->

<div class="container-fluid">

<img src="../img/top4.jpg" max-width:100% class="img-responsive">

</div><br><!-- 图片 -->

<!-- 购物车

<img src="../img/cart.jpg" height="60px" width="60px">

<a href="cart.jsp">已采购 <%=session.getAttribute("sum") %> 个物品</a><br>

<br>-->

<nav class="navbar navbar-default" role="navigation">

<div class="container-fluid">

<div class="navbar-header">

<a class="navbar-brand">商品</a>

</div>

<ul class="nav navbar-nav">

<li><a href="Shopping.jsp"><span class="glyphicon glyphicon-user"></span>首页</a></li>

<li><a href="cart.jsp"><span class="glyphicon glyphicon-usd"></span>购物车</a></li>

</ul>

</div>

</nav> <!-- 导航 -->

<div id="carousel-example-generic" class="carousel slide" data-ride="carousel">

<!-- Indicators -->

<ol class="carousel-indicators">

<li data-target="#carousel-example-generic" data-slide-to="0" class="active"></li>

<li data-target="#carousel-example-generic" data-slide-to="1"></li>

<li data-target="#carousel-example-generic" data-slide-to="2"></li>

</ol>

<!-- Wrapper for slides -->

<div class="carousel-inner" role="listbox">

<div class="item active">

<img src="../img/top2.jpg" alt="...">

<div class="carousel-caption">

</div>

</div>

<div class="item">

<img src="../img/top.jpg" alt="...">

<div class="carousel-caption">

...

</div>

</div>

</div>

<a class="left carousel-control" href="#carousel-example-generic" role="button" data-slide="prev">

<span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>

<span class="sr-only">Previous</span>

</a>

<a class="right carousel-control" href="#carousel-example-generic" role="button" data-slide="next">

<span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>

<span class="sr-only">Next</span>

</a>

</div>

<div class="container ">

<table border="1" class="table table-bordred ">

<tr><!-- 名字 -->

<c:if test="${not empty list}">

<c:forEach items="${list}" var="stuff" >

<td><a href="detail.jsp?id=${stuff.id}"><font color="Brown" >${stuff.name}</font></a></a></td>

</c:forEach>

</c:if>

</tr>

<tr>

<c:if test="${not empty list}">

<c:forEach items="${list}" var="stuff" >

<td><img src="${stuff.img}"></td>

</c:forEach>

</c:if>

</tr><!-- 图片 -->

<tr class="warning text-center"><!-- 购买 -->

<c:if test="${not empty list}">

<c:forEach items="${list}" var="stuff" >

<td><a class="btn btn-success btn-lg btn-block"href="buy.jsp?id=${stuff.id}">

$ ${stuff.price} 放进购物车！</a></td>

</c:forEach>

</c:if>

</tr>

</table>

</div>

Buy.jsp

<%

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

**int** sum = (Integer) session.getAttribute("sum");

Stuff[] Stuff = (Stuff[])session.getAttribute("Stuff");

// Stuff[] Stuff = request.getAttribute("Stuff");

out.println("buy.jsp");

**for**(Stuff stuff:Stuff)

{

**if**(id.equals(stuff.getId()))

{

**int** tmp = stuff.getNumber()+1;

stuff.setNumber(tmp);

++sum;

%>

<body>

<div class=*"list-group"*>

<a href=*"#"* class=*"list-group-item active"*>

<h4 class=*"list-group-item-heading"*>

已成功购买<%=stuff.getName()%>1个

</h4>

</a>

<a href=*"buy.jsp?id=*<%=stuff.getId()%>*"* class=*"list-group-item"*>

<h4 class=*"list-group-item-heading"*>

点击再购买一个

</h4>

<p class=*"list-group-item-text"*>目前已成功购买了<%= stuff.getName()%><%=stuff.getNumber()%>个<br>

总数量<%=sum %>个<br>

<%

session.setAttribute("sum",sum);

session.setAttribute("Stuff",Stuff);

}}

%>

</p>

</a>

<a href=*"Shopping.jsp"* class=*"list-group-item"*">

<h4 class=*"list-group-item-heading"*>

回到首页

</h4>

<p class=*"list-group-item-text"*>

继续购买宝贝

</p>

</a>

<a href=*"cart.jsp"* class=*"list-group-item"*>

<h4 class=*"list-group-item-heading"*>

回到购物车

</h4>

<p class=*"list-group-item-text"*>

查看已买宝贝

</p>

</a>

</div>

Cart.jsp

<%

Stuff[] Stuff = (Stuff[])session.getAttribute("Stuff");

**int** sum = (Integer) session.getAttribute("sum");

out.println("cart.view");

%>

<body>

<div class=*"list-group"*>

<a href=*"#"* class=*"list-group-item active"*>

<h4 class=*"list-group-item-heading"*>

订单消息

</h4>

</a>

<a href=*"#"* class=*"list-group-item"*>

<h4 class=*"list-group-item-heading"*>

购物车:

</h4>

<p class=*"list-group-item-text"*>

<br>

<% **for**(Stuff stuff:Stuff){**if**(stuff.getNumber()>0)%>

<%=stuff.getName()%>买了<%= stuff.getNumber()%>个<%System.out.println();} %>

共买了<%=sum %>个

</p>

<%

session.setAttribute("sum",sum);

session.setAttribute("Stuff",Stuff);

%>

</a>

<a href=*"Shopping.jsp"* class=*"list-group-item"*">

<h4 class=*"list-group-item-heading"*>

回到首页

</h4>

<p class=*"list-group-item-text"*>

继续购买宝贝

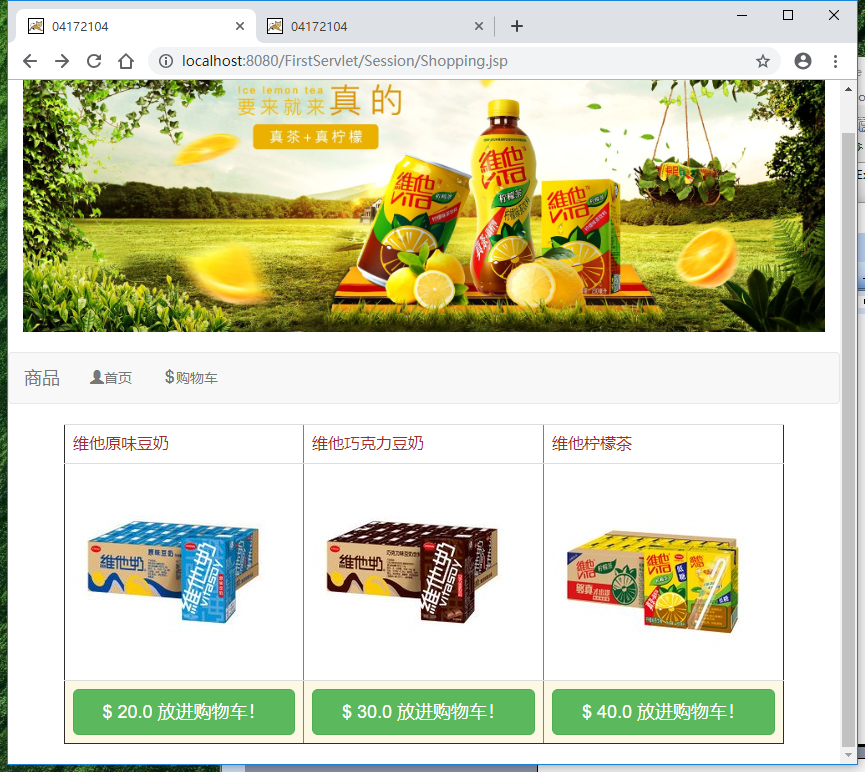
</p>

</a>

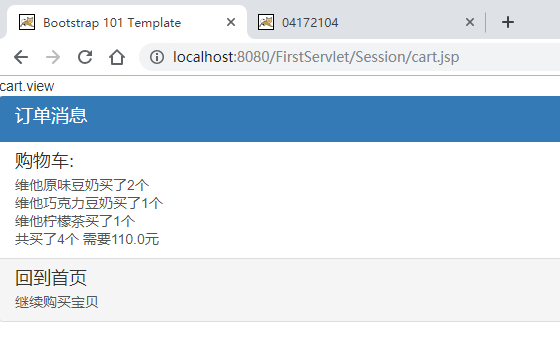
</div>

# 运行效果图

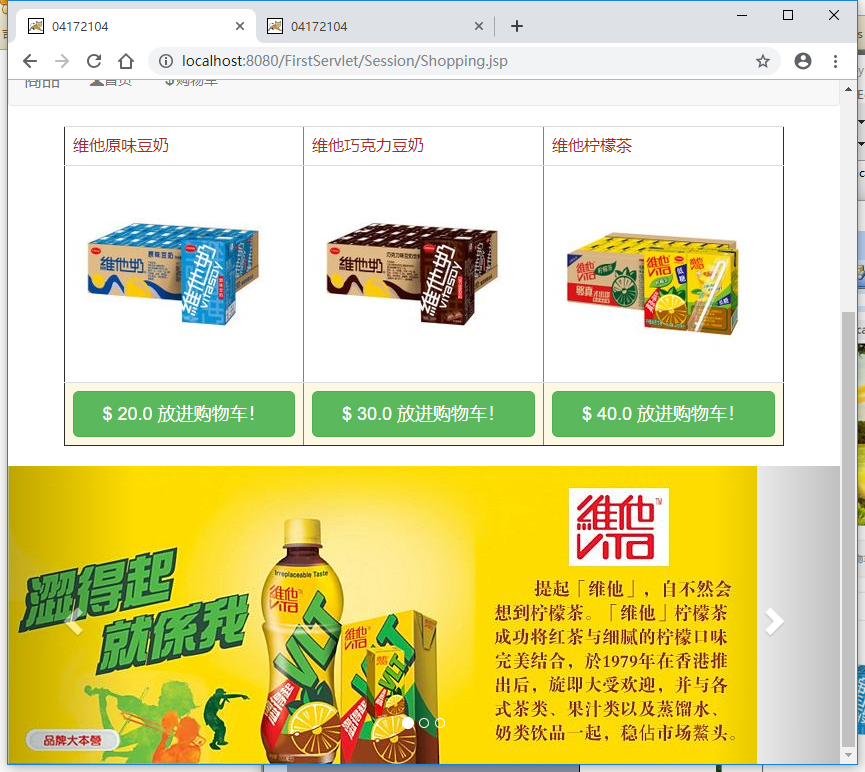
主界面(实验5的改进,加了轮播图,加了导航条上面的图,代码优化展示的表格,新增加了删除商品和计算总价的功能和显示详细信息的功能):



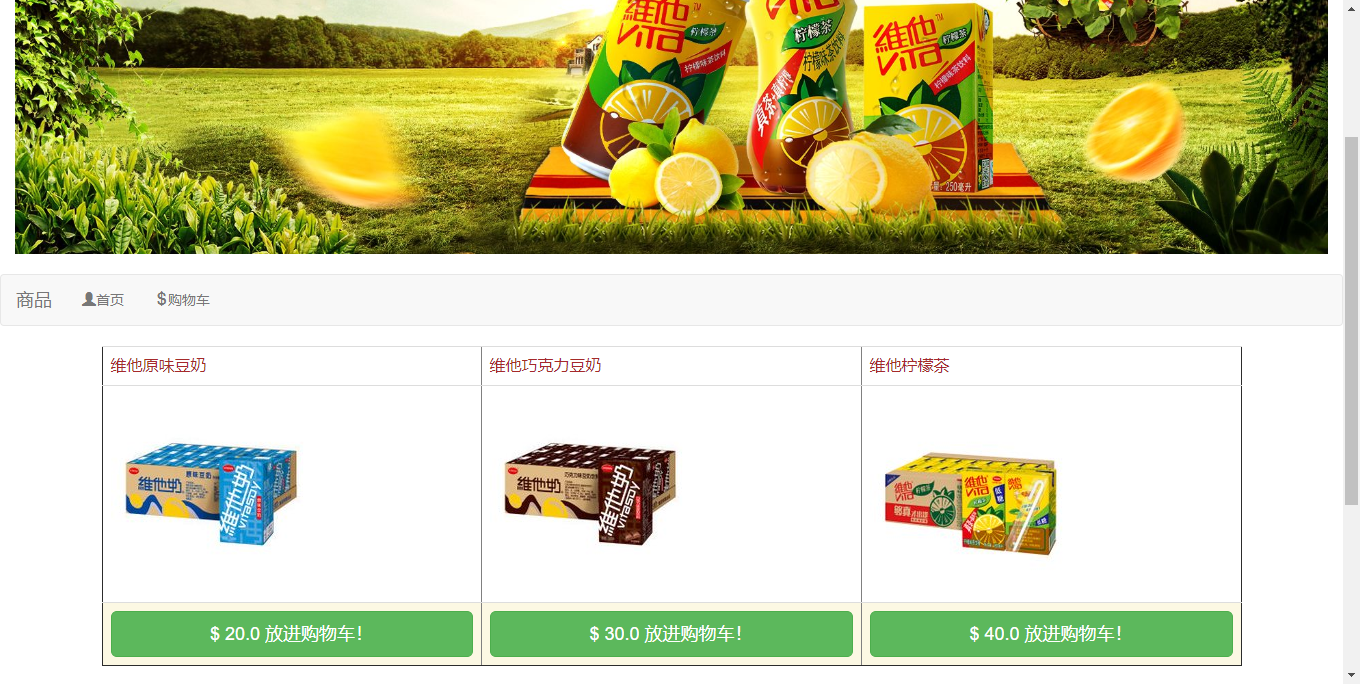
计算总价的:



轮播图;



栅格问题:



# 收获与问题

学会和熟悉用el标签,学会bootstrap的栅格自适应的问题,也学会用轮播图播放图片.用el标签大大简化了jsp的java代码,更具有可看性.