

(1)创建封装商品信息的值 JavaBean——GoodsSingle。在该 JavaBean 中定义了 name、price、num 属性，分别保存商品名称、价格、购买数量。GoodsSingle 的代码如下。

```
package com.yxq.valuebean;
public class GoodsSingle {
    private String name;           //保存商品名称
    private float price;           //保存商品价格
    private int num;               //保存商品购买数量
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getNum() {
        return num;
    }
    public void setNum(int num) {
        this.num = num;
    }
    public float getPrice() {
        return price;
    }
    public void setPrice(float price) {
        this.price = price;
    }
}
```

(2)创建工具 JavaBean——MyTools。MyTools 用来实现将 String 型数据转换为 int 型数据和解决中文乱码问题，其代码如下。

```
package com.yxq.toolbean;

import java.io.UnsupportedEncodingException;

public class MyTools {
    public static int strToint(String str){           //将 String 型数据转换为 int
型数据的方法
        if(str==null||str.equals(""))
            str="0";
        int i=0;
        try{
            i=Integer.parseInt(str);
        }catch(NumberFormatException e){
            i=0;
            e.printStackTrace();
        }
    }
}
```



```

        int i=0;
        for(;i<buylist.size();i++){ //遍历 buylist 集合对象,判断该集合
            中是否已经存在当前要添加的商品
            GoodsSingle temp=(GoodsSingle)buylist.get(i); //获取
            buylist 集合中当前元素
            if(temp.getName().equals(single.getName())){ //判断从
                buylist 集合中获取的当前商品的名称是否与要添加的商品的名称相同
                //如果相同,说明已经购买了该商品,只需要将商品的购买数量
                加 1
                temp.setNum(temp.getNum()+1); //将商品购买数
                量加 1
                break; //结束 for 循环
            }
        }
        if(i>=buylist.size()){ //说明 buylist 中不存在要添加的商
            品
            GoodsSingle temp=new GoodsSingle();
            temp.setName(single.getName());
            temp.setPrice(single.getPrice());
            temp.setNum(single.getNum());
            buylist.add(temp); //存储商品
        }
    }
}
/**
 * @功能 从购物车中移除指定名称的商品
 * @参数 name 表示商品名称
 */
public void removeItem(String name){
    称的商品
    for(int i=0;i<buylist.size();i++){ //遍历 buylist 集合,查找指定名
        位置商品
        GoodsSingle temp=(GoodsSingle)buylist.get(i); //获取集合中当前
        //如果商品的名称为 name 参数指定的名称
        if(temp.getName().equals(MyTools.toChinese(name))){

            if(temp.getNum()>1){ //如果商品的购买数量
                大于 1
                temp.setNum(temp.getNum()-1); //则将购买数量减 1
                break; //结束 for 循环
            }
            else if(temp.getNum()==1){ //如果商品的购买数量为 1

```

```

        buylist.remove(i);           //从 buylist 集合对象中移除
该商品
    }
}
}
}
/**
 * @功能 清空购物车
 */
public void clearCar(){
    buylist.clear();               //清空 buylist 集合对象
}
}

```

(4)创建实例的首页面 index.jsp，在该页面中初始化商品信息列表，然后将请求转发到 show.jsp 页面显示商品。该操作主要是将米格商品封装到对应的 GoodsSingle 类对象中，然后将 GoodsSingle 类对象存储到 ArrayList 集合对象中，最后将该 ArrayList 集合对象保存到 session 范围内。在这里，为了简单起见，将商品信息写在了 JSP 中，实际开发中商品信息应保存到数据库中。Index.jsp 页面的具体代码如下。

```

<%@ page contentType="text/html;charset=gb2312"%>
<%@ page import="java.util.ArrayList" %>
<%@ page import="com.yxq.valuebean.GoodsSingle" %>
<%!
    static ArrayList goodslist=new ArrayList();           //用来存储商品
    static{                                                 //静态代码块
        String[] names={"苹果","香蕉","梨","橘子"}; //商品名称
        float[] prices={2.8f,3.1f,2.5f,2.3f};           //商品价格
        for(int i=0;i<4;i++){                             //初始化商品信息列表
            //定义一个 GoodsSingle 类对象来封装商品信息
            GoodsSingle single=new GoodsSingle();
            single.setName(names[i]);                     //封装商品名称信息
            single.setPrice(prices[i]);                   //封装商品价格信息
            single.setNum(1);                             //封装购买数量信息
            goodslist.add(i,single);                      //保存商品到 goodslist 集合
对象中
        }
    }
%>
<%
    session.setAttribute("goodslist",goodslist);         //保存商品列表到
session 中
    response.sendRedirect("show.jsp");                   //跳转到 show.jsp 页面显示
商品
%>

```

(5)创建 show.jsp 页面，该页面显示 index.jsp 页面中初始化的商品列表。页面中首先获取存储在 session 范围中的 goodslist 集合对象，然后遍历 goodslist 集合依次输出存储的商品。其关键代码如下。

```
<%@ page contentType="text/html;charset=gb2312"%>
<%@ page import="java.util.ArrayList" %>
<%@ page import="com.yxq.valuebean.GoodsSingle" %>
<% ArrayList goodslist=(ArrayList)session.getAttribute("goodslist");%>
<table border="1" width="450" rules="none" cellspacing="0" cellpadding="0">
  <tr height="50"><td colspan="3" align="center">提供商品如下</td></tr>
  <tr align="center" height="30" bgcolor="lightgrey">
    <td>名称</td>
    <td>价格(元/斤)</td>
    <td>购买</td>
  </tr>
  <% if(goodslist==null||goodslist.size()==0){ %>
    <tr height="100"><td colspan="3" align="center">没有商品可显示！
  </td></tr>
  <%
    }
    else{
      for(int i=0;i<goodslist.size();i++){
        GoodsSingle single=(GoodsSingle)goodslist.get(i);
      %>
      <tr height="50" align="center">
        <td><%=single.getName()%></td>
        <td><%=single.getPrice()%></td>
        <td><a href="docar.jsp?action=buy&id=<%=i%>">购买</a></td>
      </tr>
      <%
        }
      }
    %>
    <tr height="50">
      <td align="center" colspan="3"><a href="shopcar.jsp">查看购物车</a></td>
    </tr>
  </table>
```

(6)创建 docar 页面，docar.jsp 用来处理用户触发的“购买”、“移除”和“清空”操作。该页面通过获取请求中传递的 action 参数来判断当前请求的是什么操作。其具体代码如下。

```
<%@ page contentType="text/html;charset=gb2312"%>
<%@ page import="java.util.ArrayList" %>
<%@ page import="com.yxq.valuebean.GoodsSingle" %>
<%@ page import="com.yxq.toolbean.MyTools" %>
<jsp:useBean id="myCar" class="com.yxq.toolbean.ShopCar" scope="session"/>
```

```

<%
    String action=request.getParameter("action");
    if(action==null)
        action="";
    if(action.equals("buy")){                                //购买商品
        ArrayList goodslist=(ArrayList)session.getAttribute("goodslist");
        int id=MyTools.strToInt(request.getParameter("id"));
        GoodsSingle single=(GoodsSingle)goodslist.get(id);
        myCar.addItem(single);    //调用 ShopCar 类中的 addItem()方法添
加商品
        response.sendRedirect("show.jsp");
    }
    else if(action.equals("remove")){                        //移除商品
        String name=request.getParameter("name");           //获取商品名称
        myCar.removeItem(name);    //调用 ShopCar 类中的 removeItem()方法
移除商品
        response.sendRedirect("shopcar.jsp");
    }
    else if(action.equals("clear")){                          //清空购物车
        myCar.clearCar();    //调用 ShopCar 类中的 clearCar()方法清
空购物车
        response.sendRedirect("shopcar.jsp");
    }
    else{
        response.sendRedirect("show.jsp");
    }
}
%>

```

(7)创建 shopcar.jsp 页面，该页面用来显示用户购买的商品。首先通过<jsp:useBean>标示获取在 docar.jsp 页面中存储在 session 中的 ShopCar 类实例，即购物车，然后获取 ShopCar 类实例中用来保存购买的商品的 buylist 集合对象，最后遍历该集合对象，输出购买的商品。其关键代码如下。

```

<% @ page contentType="text/html;charset=gb2312"%>
<% @ page import="java.util.ArrayList" %>
<% @ page import="com.yxq.valuebean.GoodsSingle" %>
<!-- 通过动作标识，获取 ShopCar 类实例 -->
<jsp:useBean id="myCar" class="com.yxq.toolbean.ShopCar" scope="session"/>
<%
    ArrayList buylist=myCar.getBuylist(); //获取实例中用来存储购买的商品
的集合
    float total=0;                        //用来存储应付金额
%>

<table border="1" width="450" rules="none" cellspacing="0" cellpadding="0">

```

```

        <tr height="50"><td colspan="5" align="center">购买的商品如下
</td></tr>
        <tr align="center" height="30" bgcolor="lightgrey">
            <td width="25%">名称</td>
            <td>价格(元/斤)</td>
            <td>数量</td>
            <td>总价(元)</td>
            <td>移除(-1/次)</td>
        </tr>
        <% if(buylist==null||buylist.size()==0){ %>
            <tr height="100"><td colspan="5" align="center">您的购物车为空!
</td></tr>
            <%
                }
            else{
                for(int i=0;i<buylist.size();i++){
                    GoodsSingle single=(GoodsSingle)buylist.get(i);
                    String name=single.getName();           //获取商品
名称
                    float price=single.getPrice();           //获取商品
价格
                    int num=single.getNum();                 //获取购买数量
                    float money=((int)((price*num+0.05f)*10))/10f;           //计算
当前商品总价，并进行四舍五入
                    total+=money;                           //计算应付
金额
                }
            }
        %>
        <tr align="center" height="50">
            <td><%=name%></td>
            <td><%=price%></td>
            <td><%=num%></td>
            <td><%=money%></td>
            <td>
                <a href="docar.jsp?action=remove&name=<%=single.getName() %>">移除</a>
            </td>
        </tr>
        <%
            }
        %>
        <tr height="50" align="center"><td colspan="5">应付金额:
<%=total%></td></tr>
        <tr height="50" align="center">

```

```
<td colspan="2"><a href="show.jsp">继续购物</a></td>
<td colspan="3"><a href="docar.jsp?action=clear">清空购物车
</a></td>
</tr>
</table>
```

至此，应用 JavaBean 实现购物车的实例创建完成。