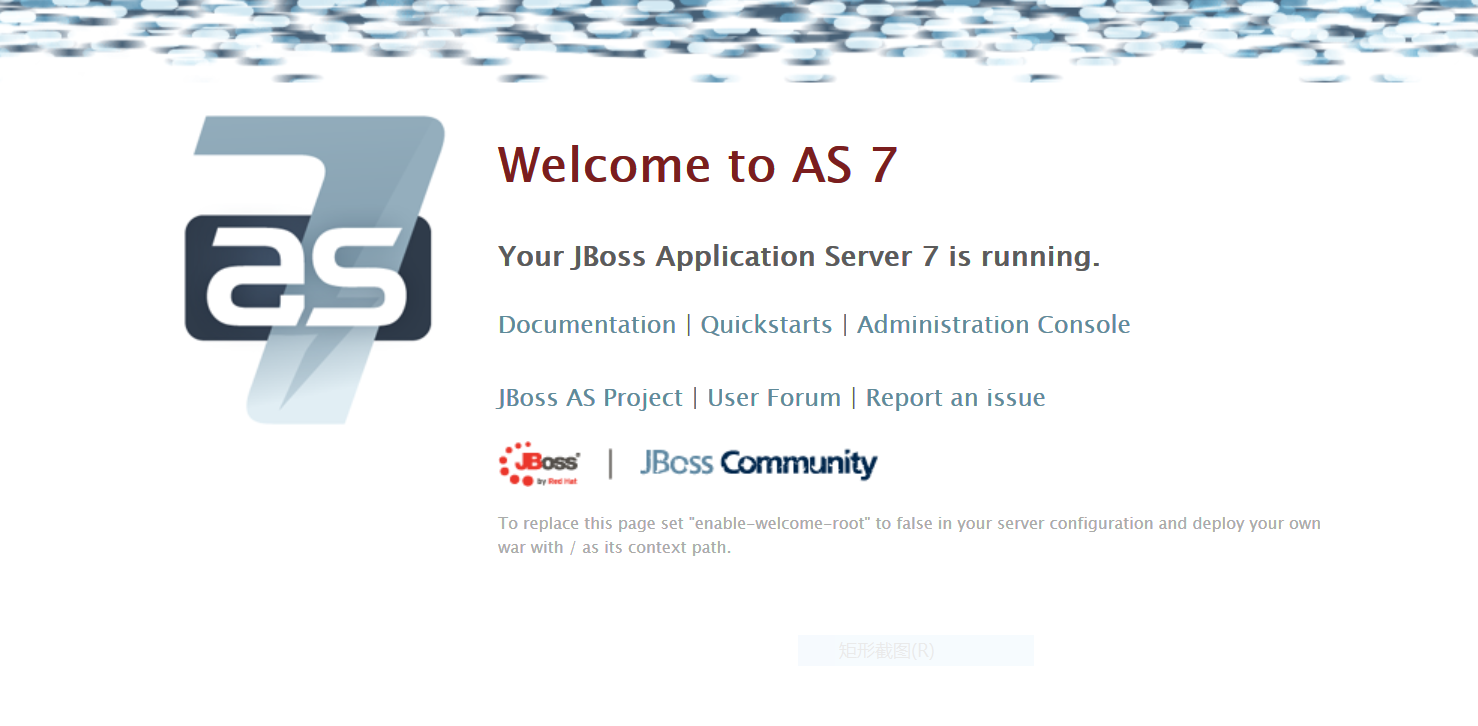
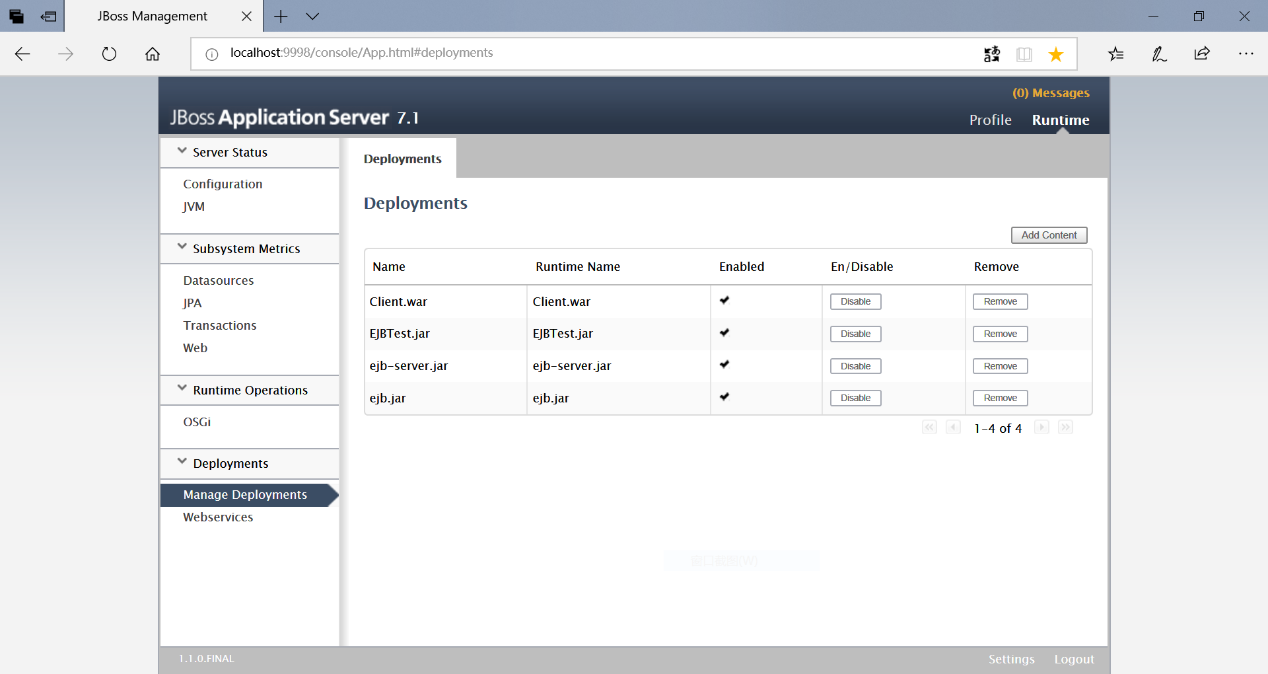
会话Bean

1. 环境部署

我用Myeclipse+jdk7+Jboss7.1进行EJB3的开发。在配置完环境变量，添加用户后，打开浏览器，输入地址：<http://localhost:8899/>。将出现jboss的欢迎界面，证明jboss已经工作正常了：



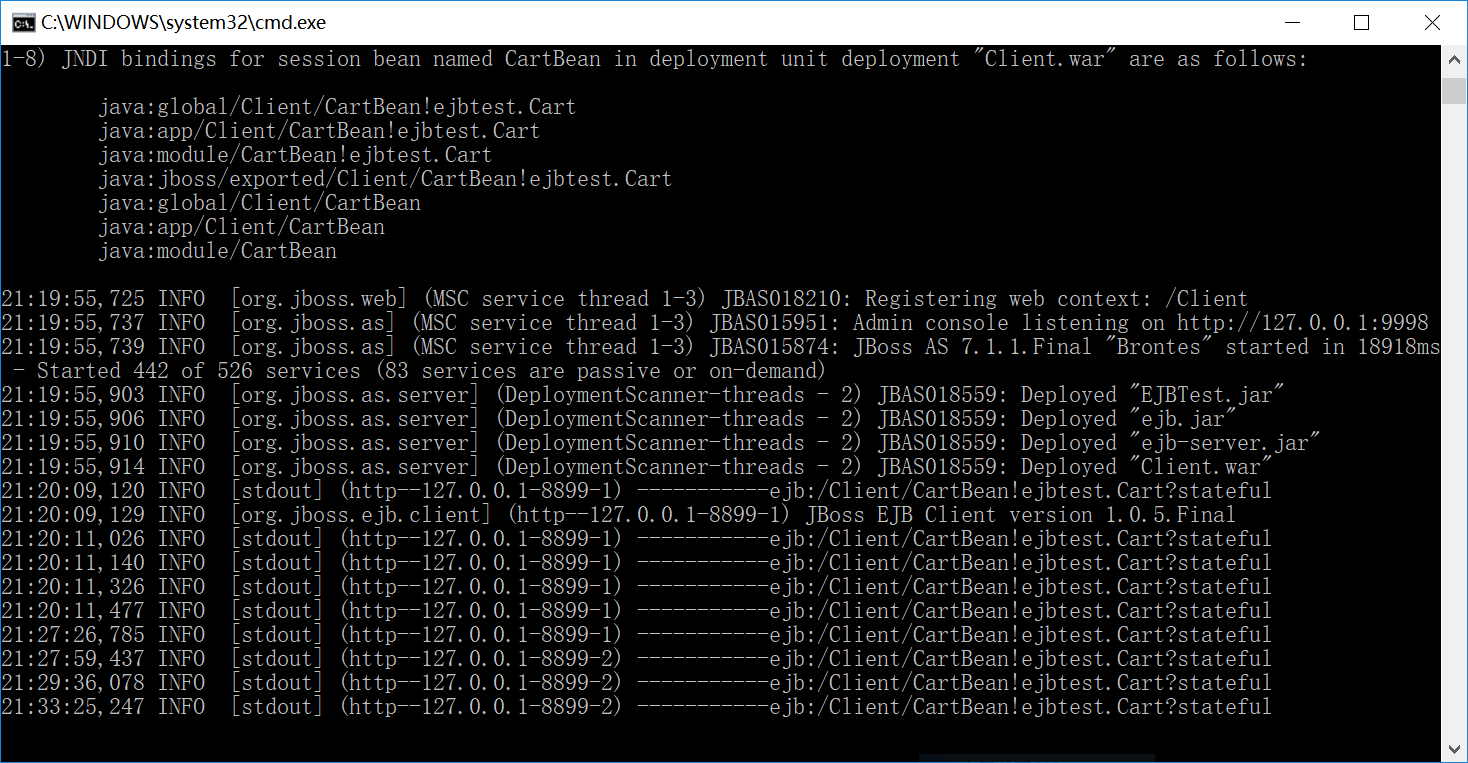
将jboss服务器集成到myeclispe中，以便使用myeclipse启动jboss。然后创建EJB项目，编写业务接口和Bean类，将项目部署到jboss中并运行，jboss服务器的控制界面如图：



开发WEB客户端进行远程访问EJB,先将EJB项目打包成.jar的格式，然后在WEB项目中导入该jar包，并编写配置文件jboss-ejb-client.properties。

最后将客户端项目使用war包方式发布，需要把war包复制到\jboss-as-7.1.1.Final\standalone\deployments目录下，运行jboss即可自动部署，在不同的浏览器中打开jsp客户端进行测试。

Jboss已经自动部署：

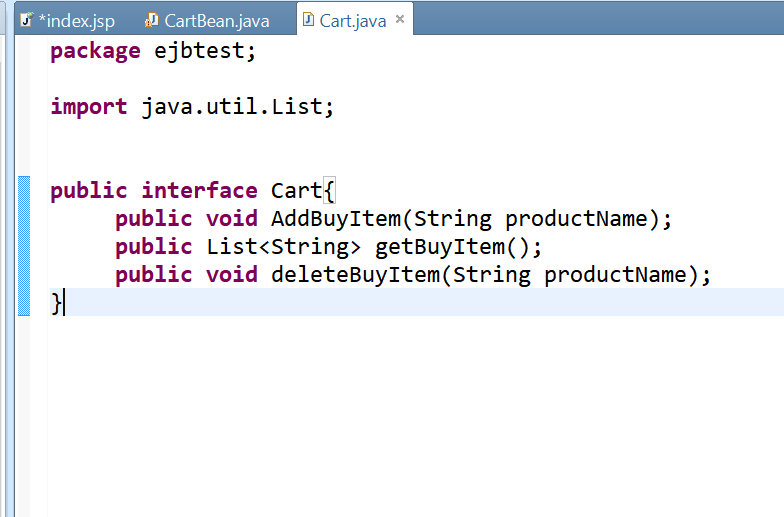


二、基于四种基础情况的分析

1.第一种情况：

**Bean设置为stateful，客户端JSP中使用session方法。**

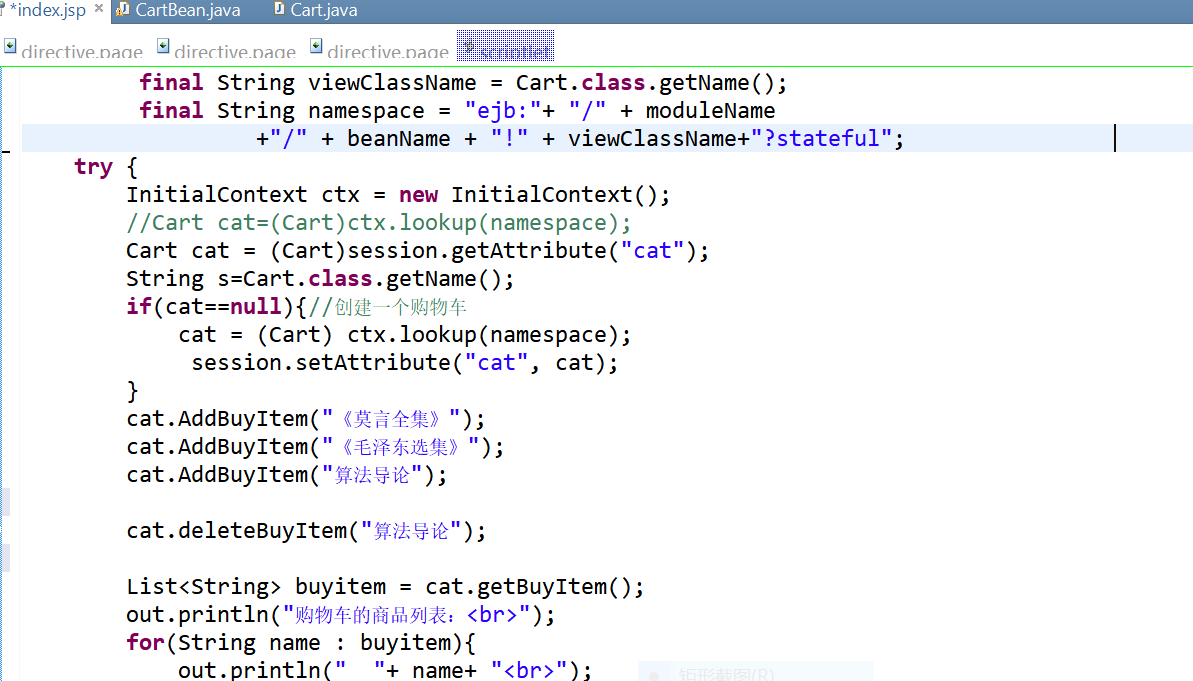
接口：



Bean类：



客户端jsp:



运行结果：见 录屏.zip 中的 情况1.gif

**分析**：不同的浏览器同时访问jsp时，购物车中的图书只添加一次;同一浏览器不断刷新，多次访问jsp时，购物车中的图书都会多次累计添加。

**原因**：EJB中的sessionBean是有状态会话Bean，每个客户的不同状态都会被保存下来；而客户端通过环境对象InitialContext获得要访问的EJB的Remote接口，并将该接口放入session中以供下一次使用，然后通过该对象接口执行相应操作。由于不同浏览器之间的session是不能共享的，每个浏览器的访问都是一个单独的线程，所以不同浏览器在访问jsp时，获取的ejb是不同的，又因为ejb中的sessionBean是有状态的，所以购物车中的图书只添加一次；而同一浏览器在不断刷新时，ejb都是从session中获取的，它们是同一个ejb，即可以看做是同一个客户端的状态在不断变化，故购物车中的图书都会多次累计添加。

**后台机制：**有状态会话Bean始终和一个客户相联系，在众多的方法调用过程中，这个实例代表的客户都保持一定的状态。也就是说有状态的session bean实例和会话对象之间有一对一的关系。同一客户端的下一个方法请求必须得到保存在Bean中的会话状态。

有状态的session bean类的实例总是同一个对象联系在一起，并同该对象绑定在一起用来指明一个确切的客户。在客户端引用期间维护Bean中的所有实例数据的状态值，这些数据在引用期间可以被Bean其他方法所引用，其他客户不会共享同一个Session Bean的实例。

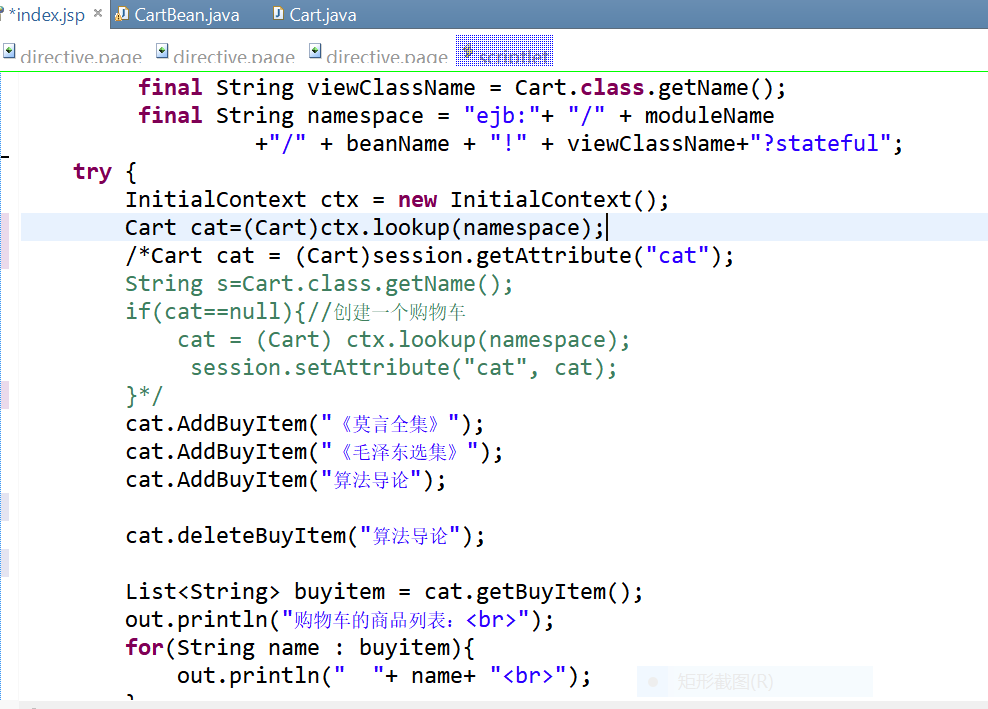
2.第二种情况：

**Bean设置为stateful，客户端JSP中不使用session，而是直接lookup。**

**sessionBean是有状态会话bean：**



**直接使用lookup获取ejb：**



运行结果：见 录屏.zip 中的 情况2.gif

**分析**：不同的浏览器同时访问jsp时，购物车中的图书只添加一次;同一浏览器不断刷新，多次访问jsp时，购物车中的图书也只添加一次。

**原因**：EJB中的sessionBean是有状态会话Bean，每个客户的不同状态都会被保存下来；而客户端通过环境对象InitialContext获得要访问的EJB的Remote接口，并将该接口放入session中以供下一次使用，然后通过该对象接口执行相应操作。由于不同浏览器之间的session是不能共享的，每个浏览器的访问都是一个单独的线程，所以不同浏览器在访问jsp时，获取的ejb是不同的，又因为ejb中的sessionBean是有状态的，所以购物车中的图书只添加一次；而同一浏览器在不断刷新时，ejb都是直接通过lookup方法获得，它们不是同一个ejb，即可以看做是不同一个客户端在访问ejb，故购物车中的图书只会添加一次。

**后台机制：**有状态会话Bean始终和一个客户相联系，在众多的方法调用过程中，这个实例代表的客户都保持一定的状态。也就是说有状态的session bean实例和会话对象之间有一对一的关系。同一客户端的下一个方法请求必须得到保存在Bean中的会话状态。

有状态的session bean类的实例总是同一个对象联系在一起，并同该对象绑定在一起用来指明一个确切的客户。在客户端引用期间维护Bean中的所有实例数据的状态值，这些数据在引用期间可以被Bean其他方法所引用，其他客户不会共享同一个Session Bean的实例。有状态会话Bean需要保存特定客户端的信息，一个客户端对应一个实例，既是在当时客户端有连接没有访问的情况下，也要为这个客户端保留这个实例。

3.第三种情况：

**Bean设置为stateless，客户端JSP中使用session方法。**

SessionBean是无状态会话bean：



使用session获取ejb



运行结果：见 录屏.zip 中的 情况3.gif

**分析**：不同的浏览器同时访问jsp和同一浏览器不断刷新，多次访问jsp时，购物车中的图书都会多次累计添加。

**原因**：EJB中的sessionBean是无状态会话Bean，不会保存每个客户的状态；客户端通过环境对象InitialContext获得要访问的EJB的Remote接口，并将该接口放入session中以供下一次使用，然后通过该对象接口执行相应操作，每次都是调用的同一个ejb对象，故购物车中的图书会累加。

**后台机制：**

无状态会话Bean的请求执行过程

（1）客户对无状态会话Bean的调用传递给EJB服务器

（2）EJB服务器到EJB缓冲池中查找是否有空闲的EJB组件对象存在，如果有，就将客户请求交给一个对象执行。对象执行完客户请求后，继续被停留在EJB缓冲池中。

（3）如果缓冲池目前为空，则EJB服务器就生成一个EJB组件，并使用该组件处理用户请求，当一个请求执行完后，就将EJB对象归还到组件池中，以备后面请求使用。

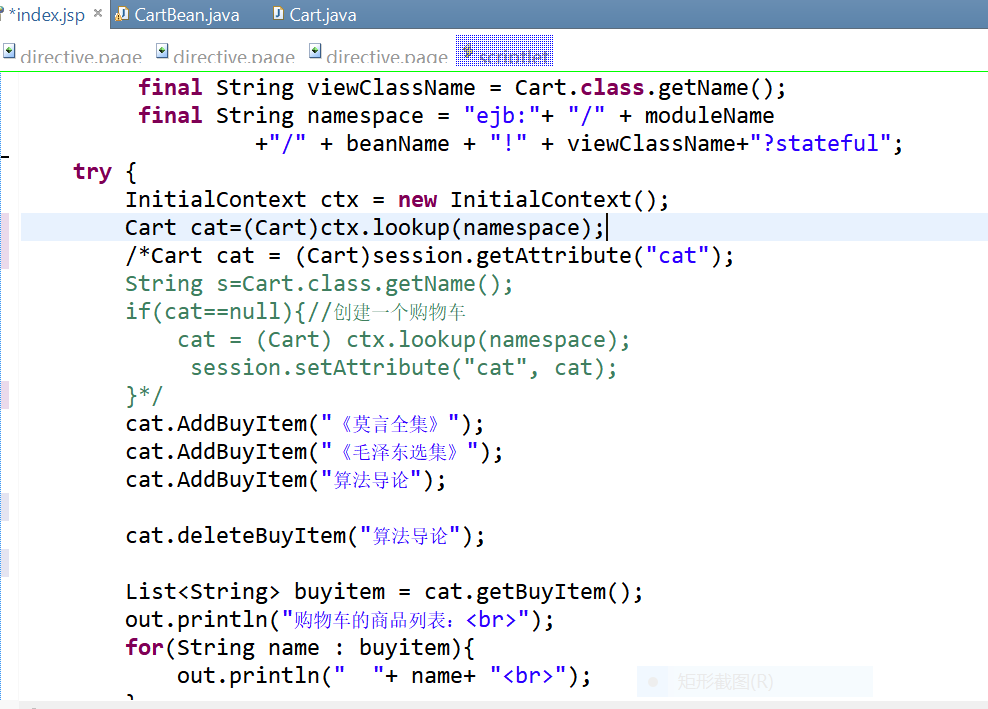
（4）如果组件池中对象长时间不被使用，则系统在超过特定时间后，会将组件池中对象销毁，以释放特定资源，降低服务器负担。

4.第四种情况：

**Bean设置为stateless，客户端JSP中使用lookup方法。**

sessionBean是无状态会话bean：





运行结果：见 录屏.zip 中的 情况4.gif

**分析**：不同的浏览器同时访问jsp和同一浏览器不断刷新，多次访问jsp时，购物车中的图书都会多次累计添加。

**原因**：EJB中的sessionBean是无状态会话Bean，不会保存每个客户的状态；客户端通过环境对象InitialContext获得要访问的EJB的Remote接口，虽然每次都是通过lookup方法获取ejb组件对象，但由于无状态会话bean的调用机制，服务器端会维持一个实例池，创建好若干个实例对象供客户端调用。当从客户端发送创建会话Bean的请求时，并不一定会真的创建EJB，多数情况下是从实例池中得到一个实例，用完之后重新放回实例池。如果下次再访问，再从实例池中取出一个实例使用，并不一定是上次的实例。即使两次访问使用的是同一个实例，在两次访问之间也有可能有其他的客户端访问了该实例。所以，并不能保证在多次访问之间的信息会被保存。故无论是不同的浏览器还是同一个浏览器的多次访问，购物车中的图书都会累加。

**后台机制：**

无状态会话Bean的请求执行过程同情况三。

三、心得体会

在前期环境部署阶段浪费了很多时间，而真正打代码却没有用很长时间。因为我电脑上安装的jdk8跟jboss7不兼容，所以又安装了兼容的jdk7，后期部署也出现了很多错误，幸好都百度到了解决方案，所以整体项目进行的还算顺利，也真是理解了sessionBean的机制和使用。