# 实验四 Hibernate基础编程

实验学时：6

实验类型：验证

**一、实验目的**

通过本实验的学习，使学生理解和掌握Hibernate框架原理与应用，重点掌握PO类编写，DAO层及服务层设计，配置和映射文件的配置和理解；掌握Hibernate多种检索方式，尤其是HQL、QBC的理解和应用，了解Hibernate框架的缓存机制，完成Hibernate框架的购物车设计和实现。

**二、实验软件**

Win7操作系统、JDK1.8、Tomcat9.0、MySQL5.7和Eclipse(neon)。

**三、实验步骤**

**基于Hibernate5Cart框架的购物车：**

1．创建DWP项目：Hibernate5Cart(可以复制Struts2Cart项目文件，或改写Struts2Cart项目，保留原Struts2Cart项目，对比学习)

2．继续使用Struts2标签库、OGNL重新编写：用户、商品、订单等界面并优化，以及顾客、银行卡信息等页面

3．继续利用DAO模式编写：用户、商品、订单等模型层、DAO层及服务层以及相关工具类

4．购物车国际化：中英文切换界面，多个页面或整个网站

5．安全和权限管理：用户分为管理员、特殊用户（浏览部分商品）、普通用户和游客

6．数据验证：针对关键信息进行合法性验证，整理所有测试用例进行单元和模块测试

7．利用struts2框架完成用户、商品等图片文件进行上传和下载。

8．利用Hibernate框架完成不同数据类型的映射、存储和管理。如日期Date，计数count，图片BLOB等，符合hibernate类规范，如主键、类名、表名等

9. 将购物车顾客Customer、银行卡Card、购物车Cart、顾客地址Address等实体编程POJO类，编写响应hbx文件，或注解实现。完成各实体类数据CRUD操作

10. 根据Hibernate关联关系映射完成用户User，商品Item，购物车Cart，银行卡Card，顾客Customer，订单Order，地址Address等之间的关系，体现一对一、一对多、多对多等关系查询

11. 针对购物车项目实现，实现5种查询方式的检索，以及动态实体模型查询

12. 总结Hibernate对数据库的批处理（如分页）、存储过程、连接池（D3P0）、事务等优化设计和实现，与其它技术的异同

13. 测试并截图

**四、实验要求**

1．使用Struts2标签库、OGNL重新编写：用户、商品、订单等界面并优化一张截图

2. 使用Struts2标签库、OGNL重新编写顾客、银行卡信息等页面一张截图

3．利用DAO模式编写Hibernate相关服务、工具类一张截图

4．购物车国际化：中英文切换界面，多个页面或整个网站

5．安全和权限管理：用户分为管理员、特殊用户（浏览部分商品）、普通用户和游客一张截图

6. 使用Hibernate框架过滤器进行安全和权限管理一张截图

7．数据验证：利用JUnit测试框架进行单元和模块测试，验证数据和功能一张截图

8．利用Hibernate框架完成不同数据类型的映射、存储和管理。如日期Date，计数count，**图片BLOB**等，符合hibernate类规范，如主键、类名、表名等一张截图

9. 将购物车顾客Customer、银行卡Card、购物车Cart、顾客地址Address等实体编程POJO类，编写响应hbm文件，或注解实现。完成各实体类数据CRUD操作一张或多张截图

10. 根据Hibernate关联关系映射完成用户User，商品Item，购物车Cart，银行卡Card，顾客Customer，订单Order，地址Address等之间的关系，体现一对一、一对多、多对多等关系查询一张或多张截图

11. 针对购物车项目实现，实现5种查询方式的检索，以及动态实体模型查询，一张或多张截图

12. 总结Hibernate对数据库的批处理（如分页）、存储过程、连接池（D3P0）、事务等优化设计和实现，与其它技术的异同，一张或多张截图。可以写在实验小结中

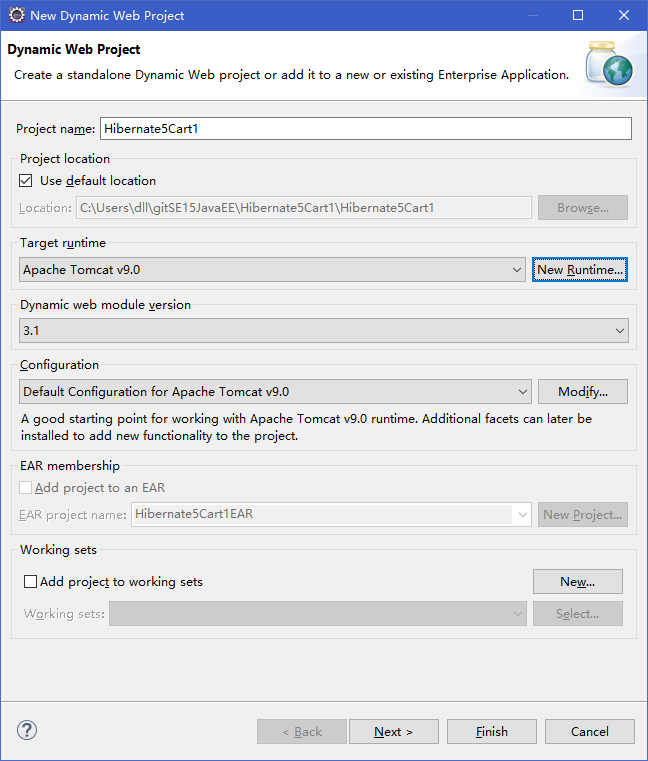
**六、实验小结**

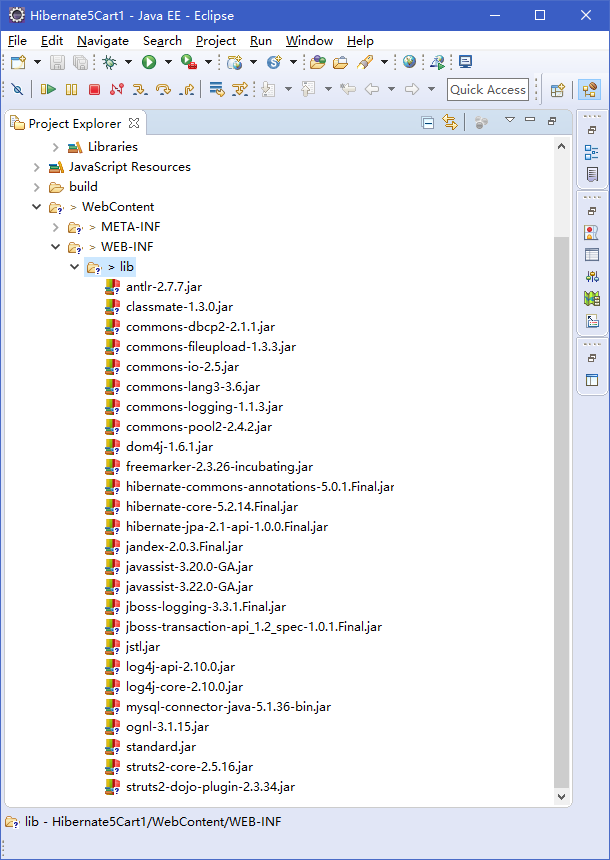
《写自己的学到具体技术、收获和体会，以及意见或建议》

# 附图：

1. **搭建Hibernate5与Struts2开发项目和环境**

项目名Hibernate5Cart1，目标运行时（tomcat9）、26个集成S2和H5的Jar包。





1. **创建**Customer PO类及**映射文件**

根据Customer的需要和HibernatePO类的编写规范（基本与JavaBean规范一致），写好11个私有属性，setter和getter及构造函数利用eclipse完成。

**private** **int** customerId;

**private** String account;

**private** String password;

**private** String name;

**private** Boolean sex;

**private** Date birthday;

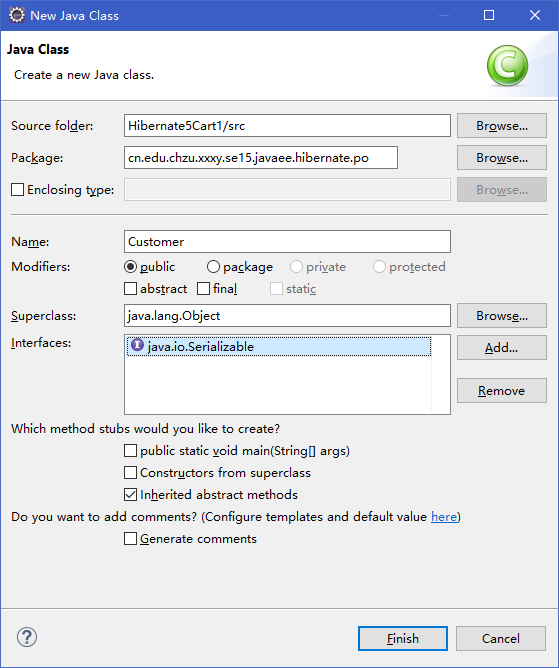
**private** String phone;

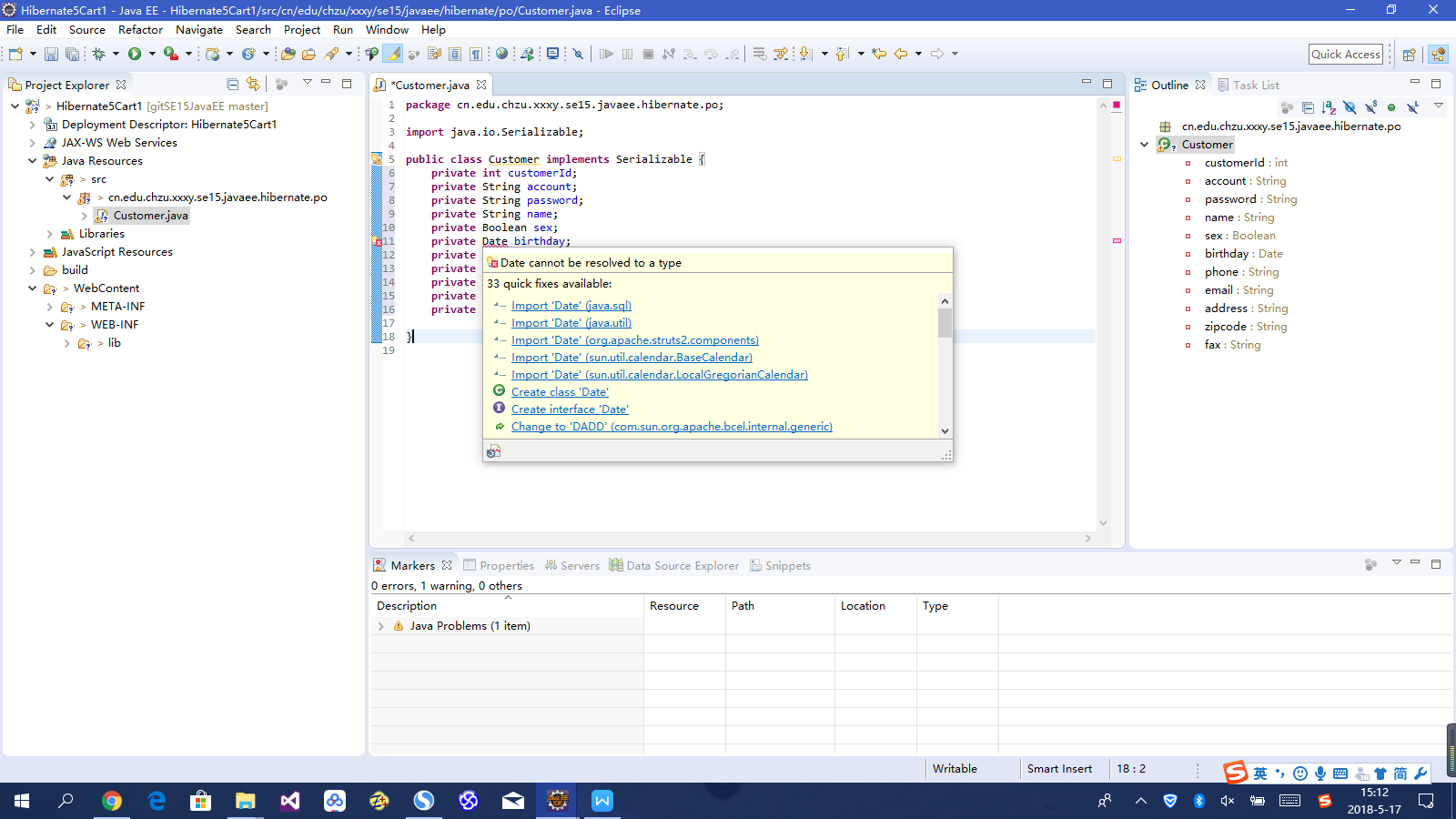
**private** String email;

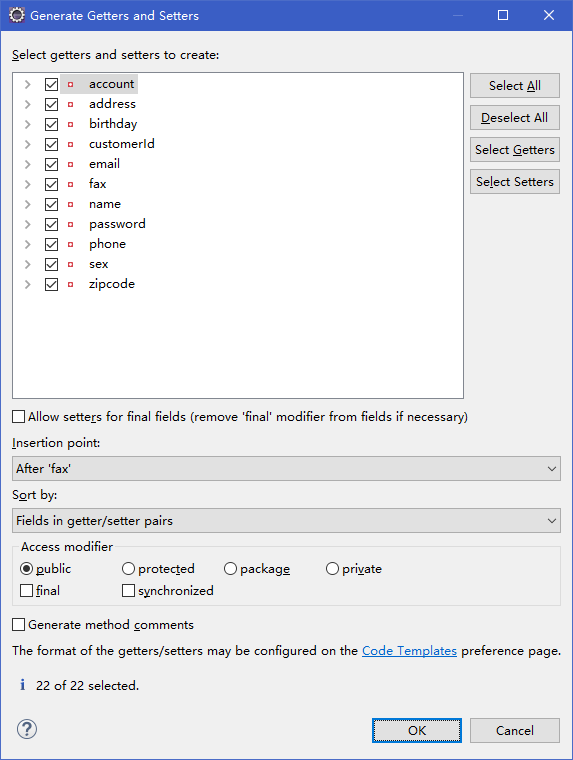
**private** String address;

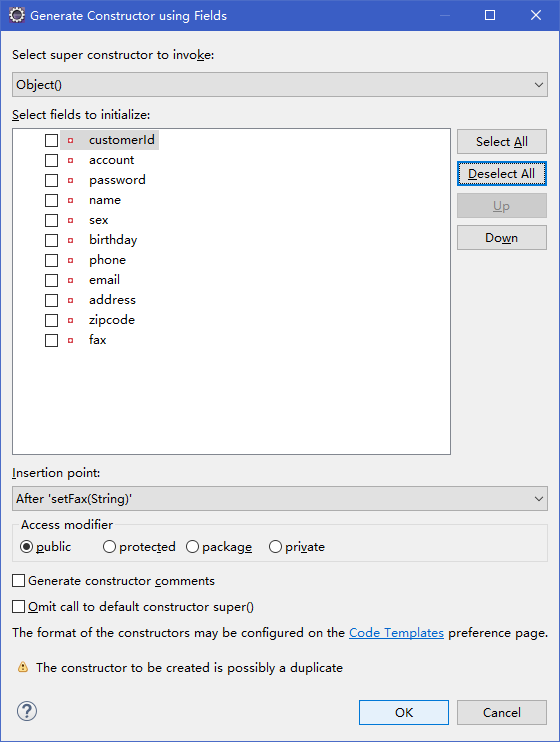
**private** String zipcode;

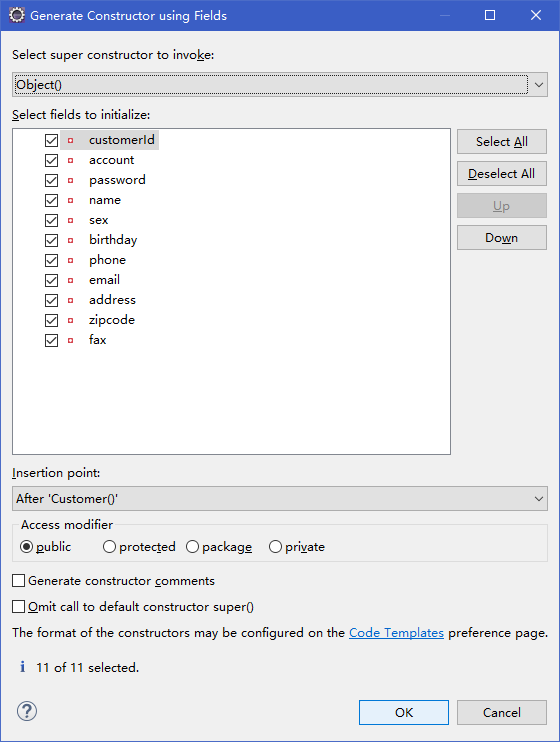
**private** String fax;





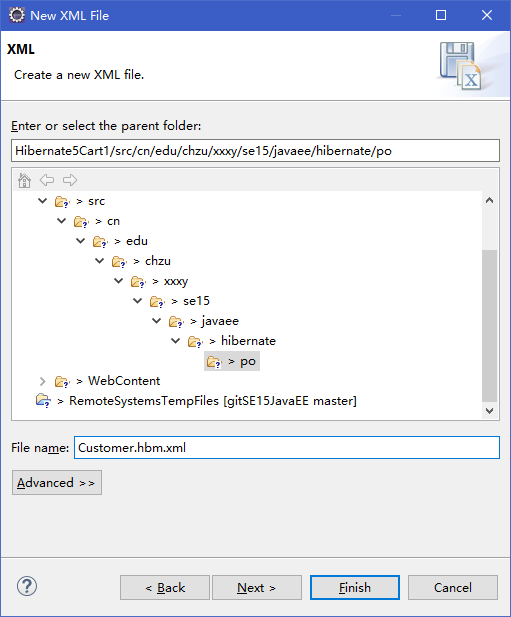


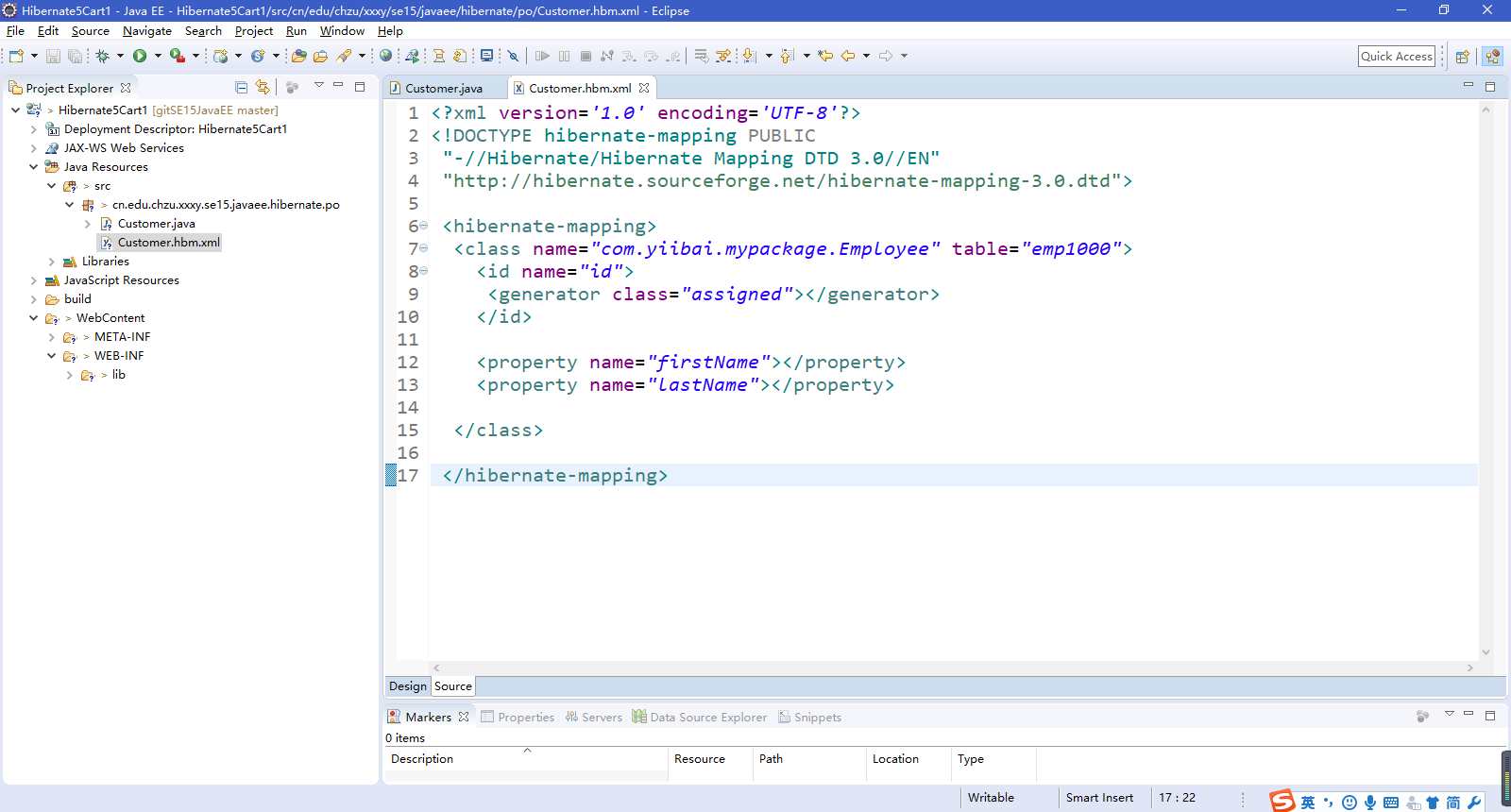


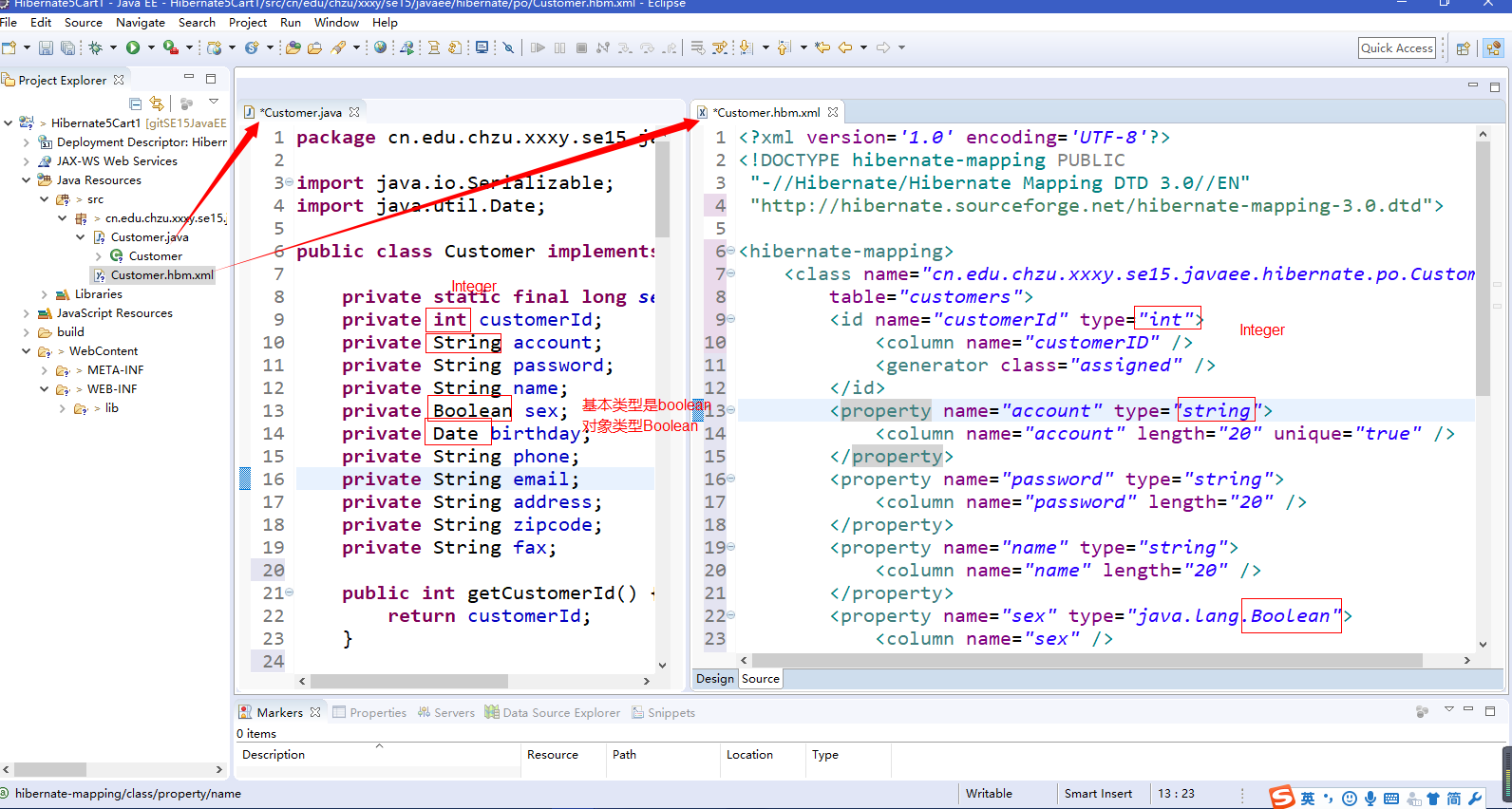


编写映射文件Customer.hbm.xml位于PO类同包中。内容参考：

<https://www.yiibai.com/hibernate/>





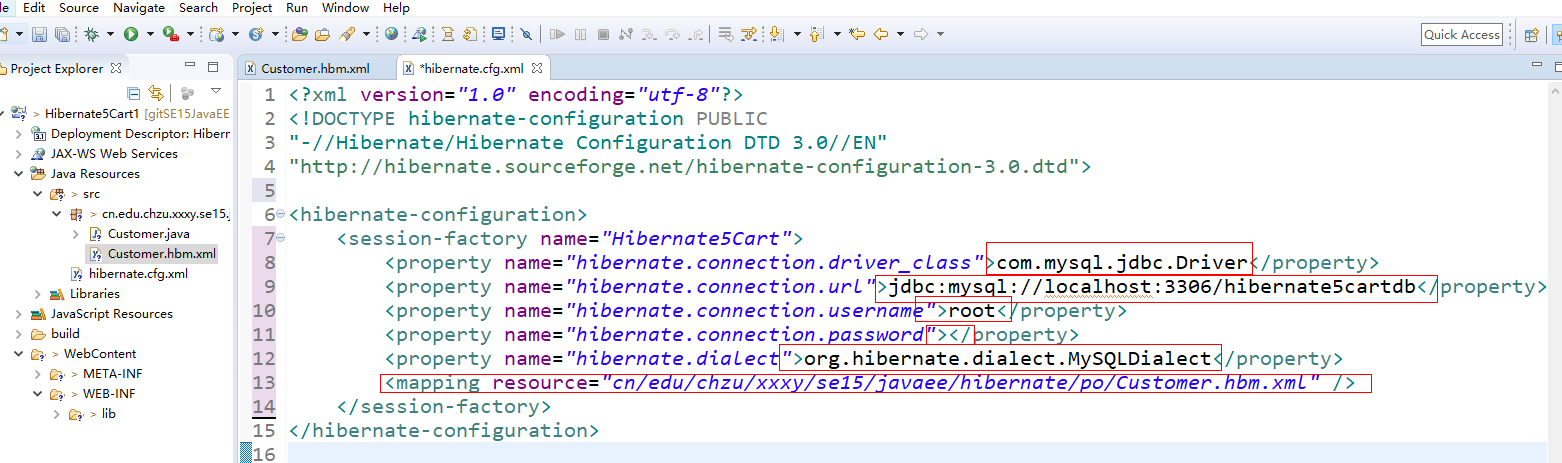


1. **创建Hibernate全局配置文件**

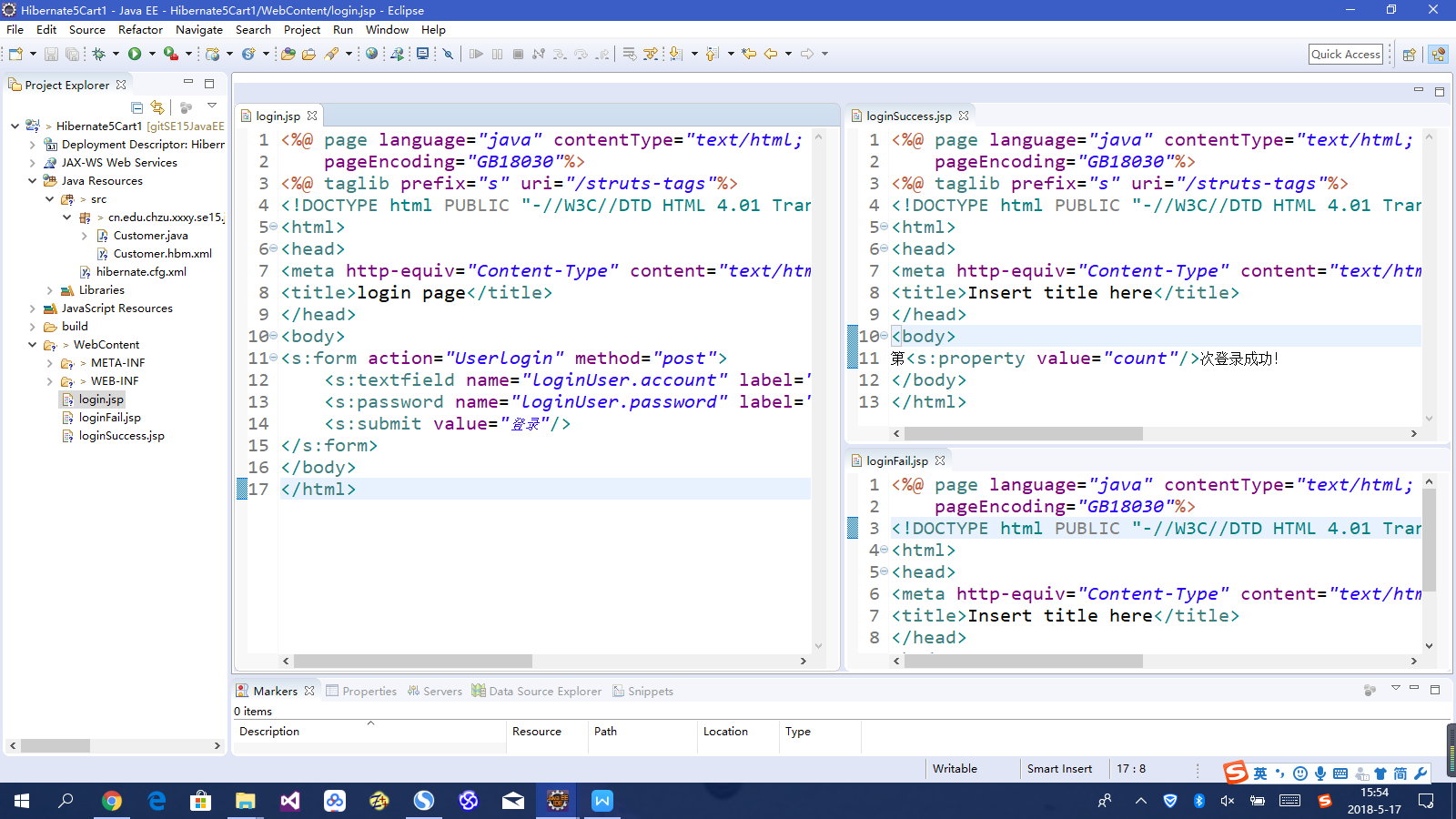
**参考：<https://www.yiibai.com/hibernate/first-hibernate-application.html>**



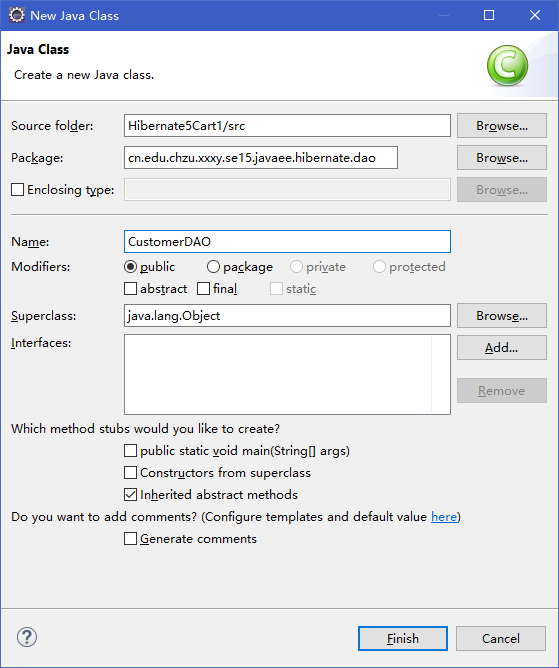
类似映射文件，创建xml，放在src目录下，内容是应用程序链接数据库的信息

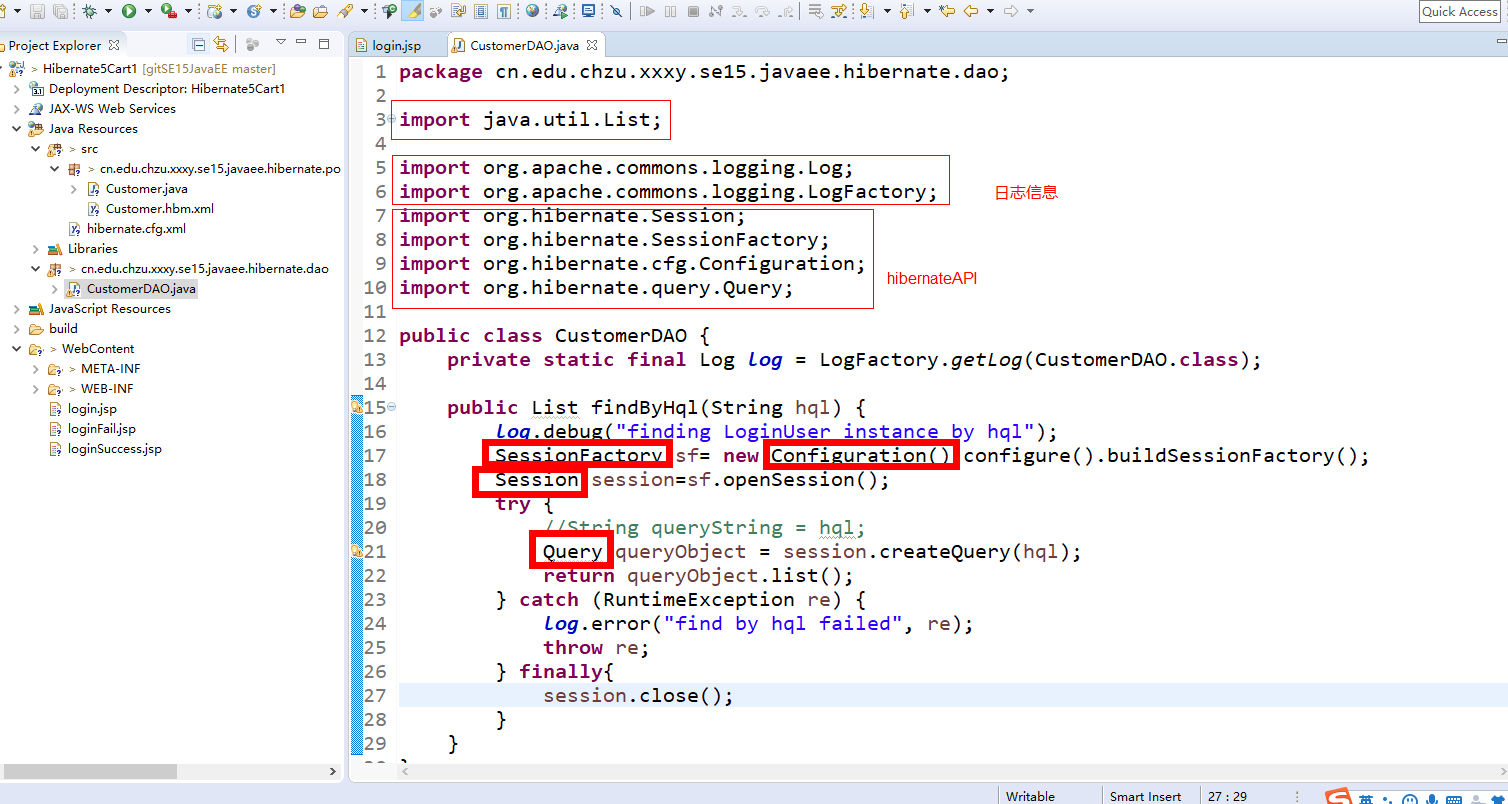


1. **新建或复制拷贝login.jsp、loginSuccess.jsp文件**



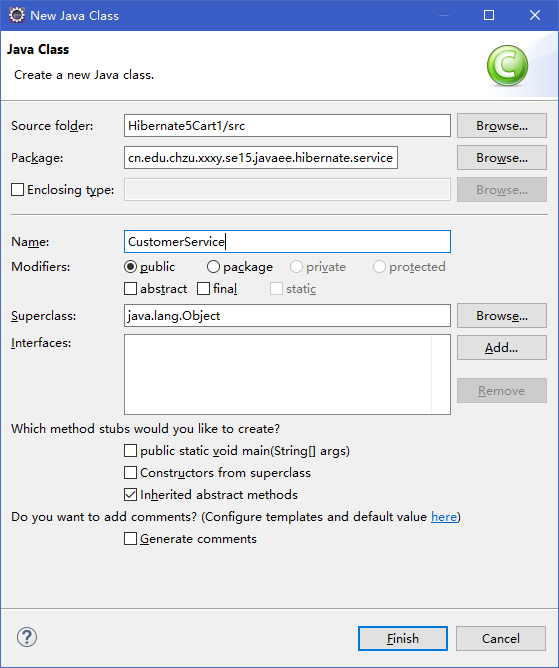
1. **创建数据库操作类CustomerDAO**

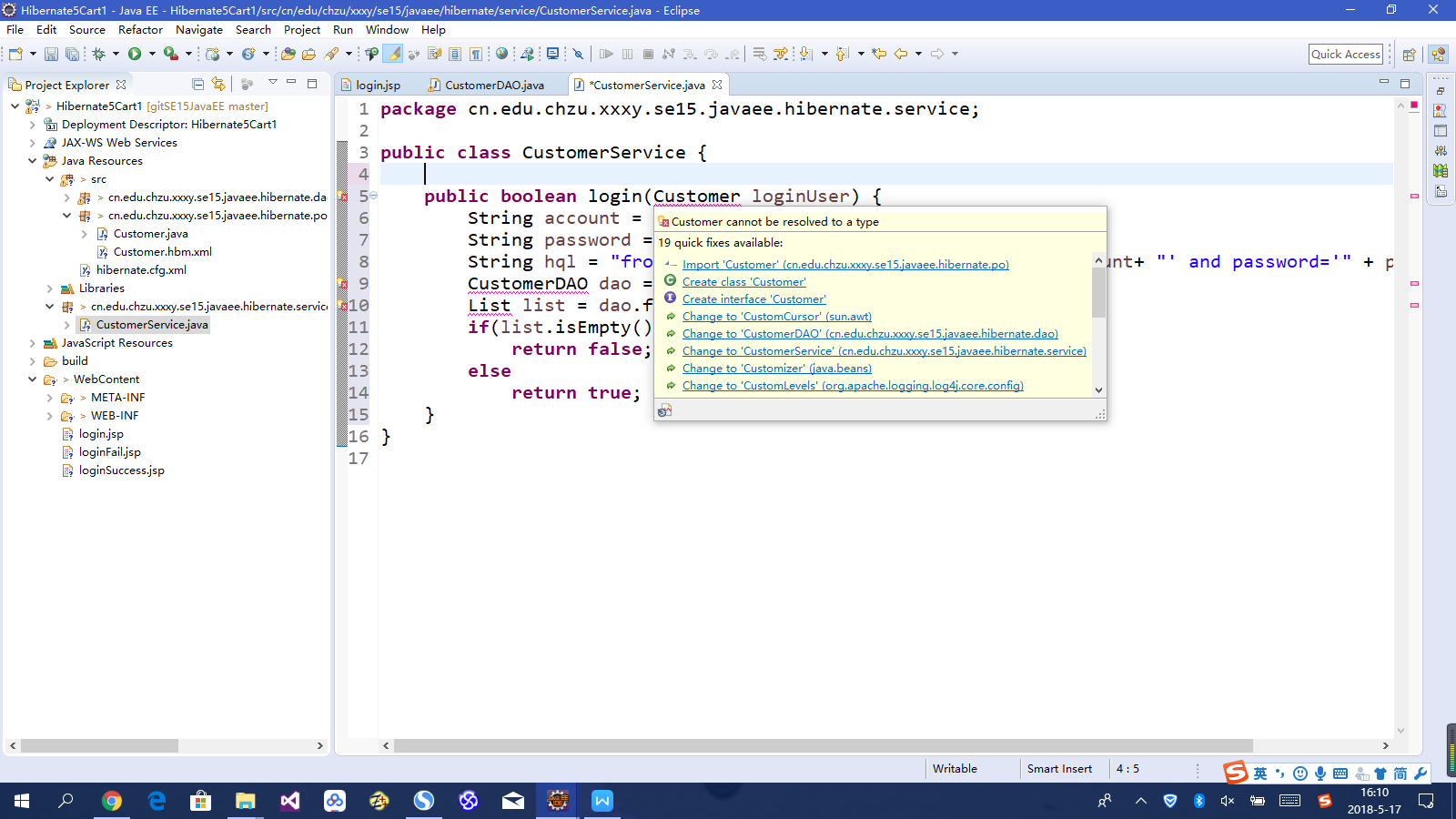




1. **创建CustomerService类实现注册/登录逻辑**

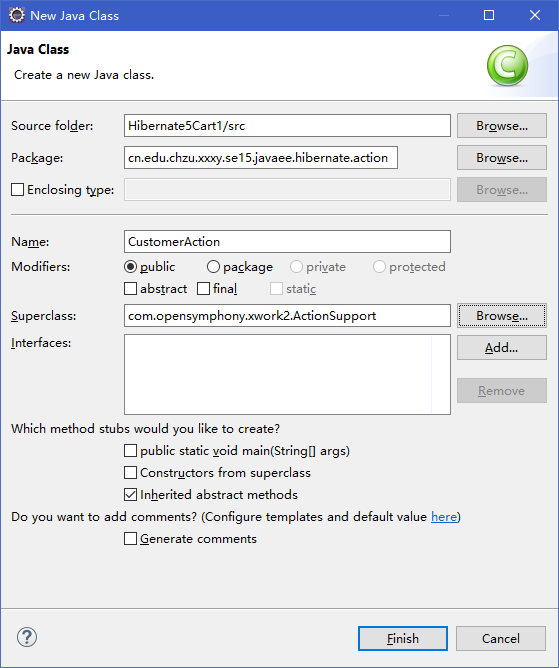
类似DAO类是普通Java类，可以敲入书中代码，并引入相应Jar包。



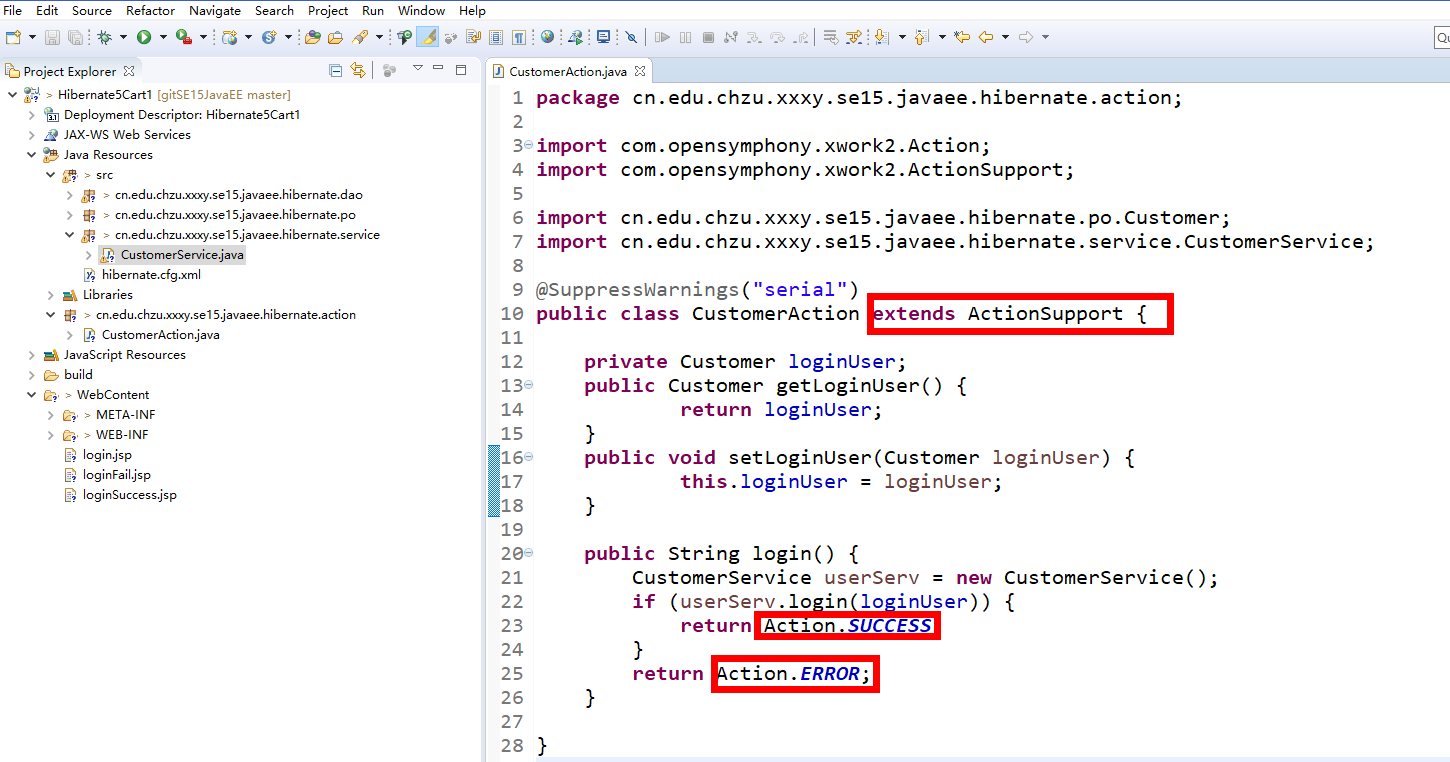


1. **创建CustomerAction类实现注册/登录控制/动作**

**CustomerAction是Struct框架的类，可继承ActionSupport，Action，甚至是普通Java类，像实验教材。但推荐使用继承ActionSupport，并利用Struts框架的规范开发应用程序，否则加载Struts框架就无意义。**



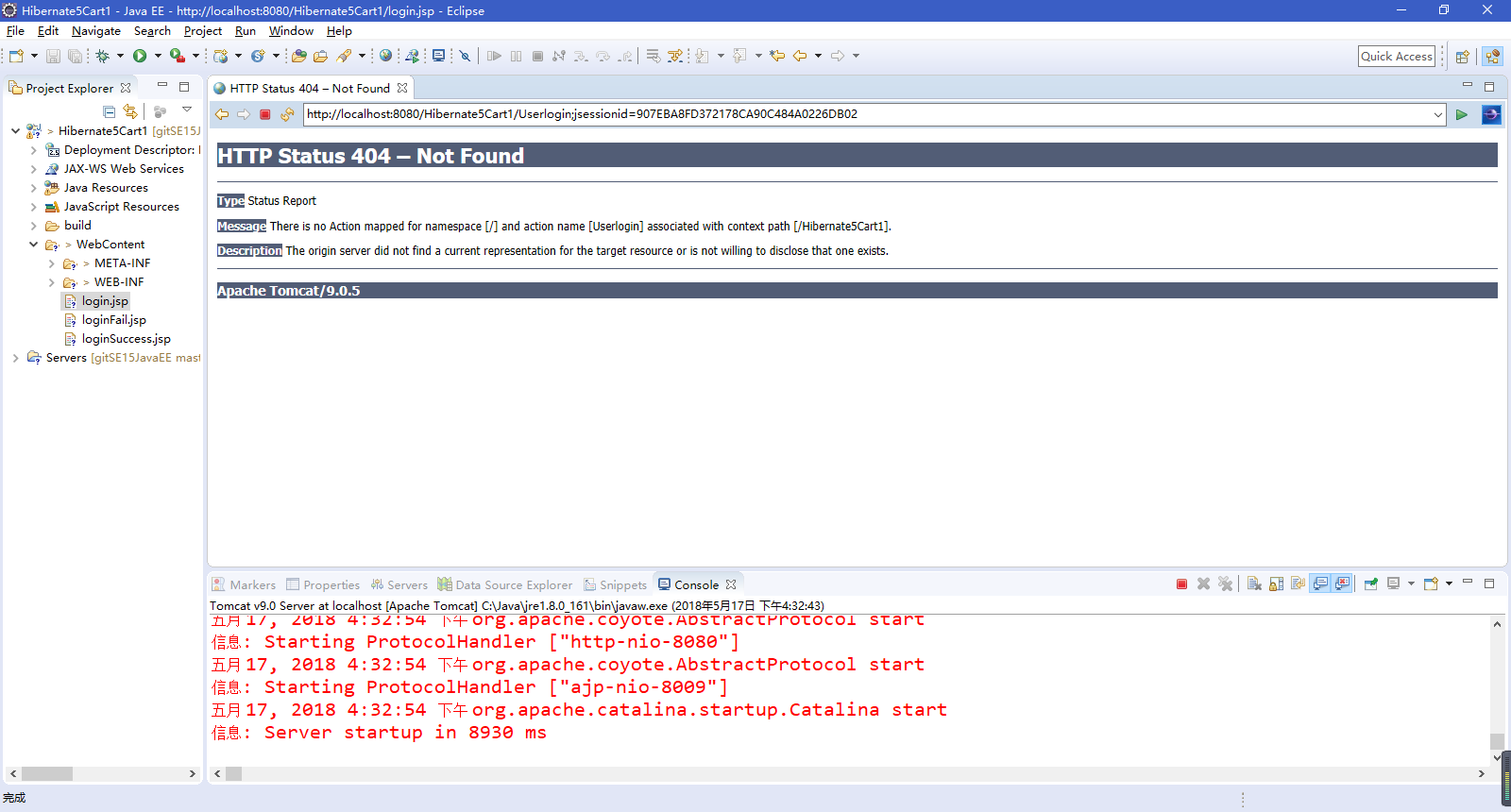
**如果我们按照书的内容完完全全打一遍，我们就是打字员，不是程序员。必须要的内容，要！不需要的，删！要知道取舍**



1. **配置Struts的核心配置文件和web.xml文件**

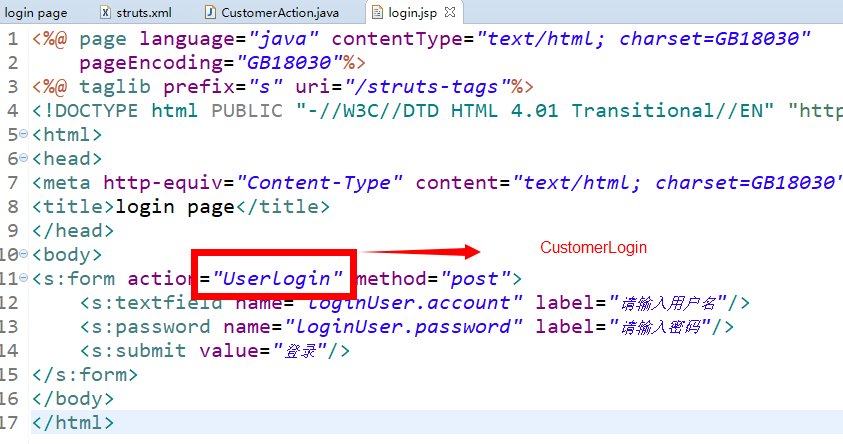


1. **总会有404，它才让我们静心、专心、细心、耐心地琢磨知识、掌握技能！**

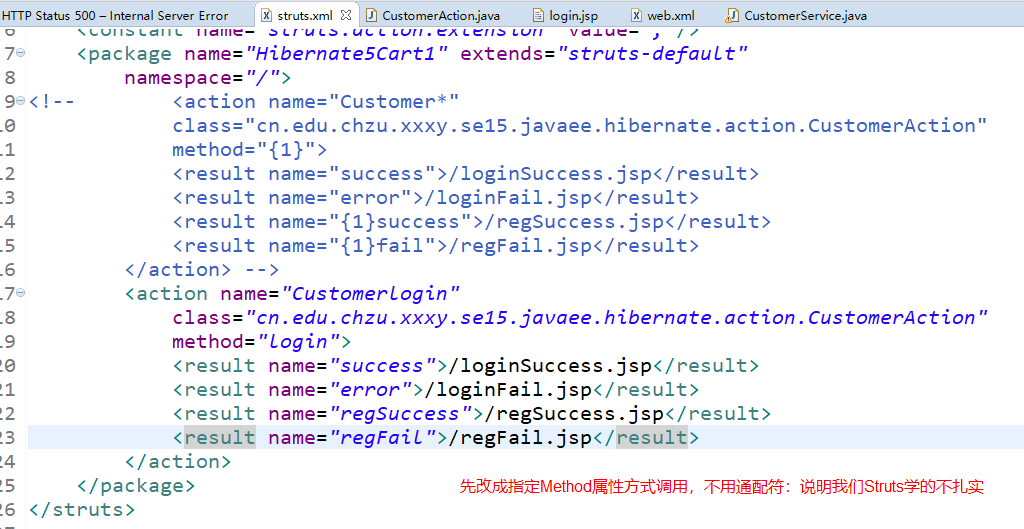


**Message** There is no Action mapped for namespace [/] and action name [Userlogin] associated with context path [/Hibernate5Cart1].

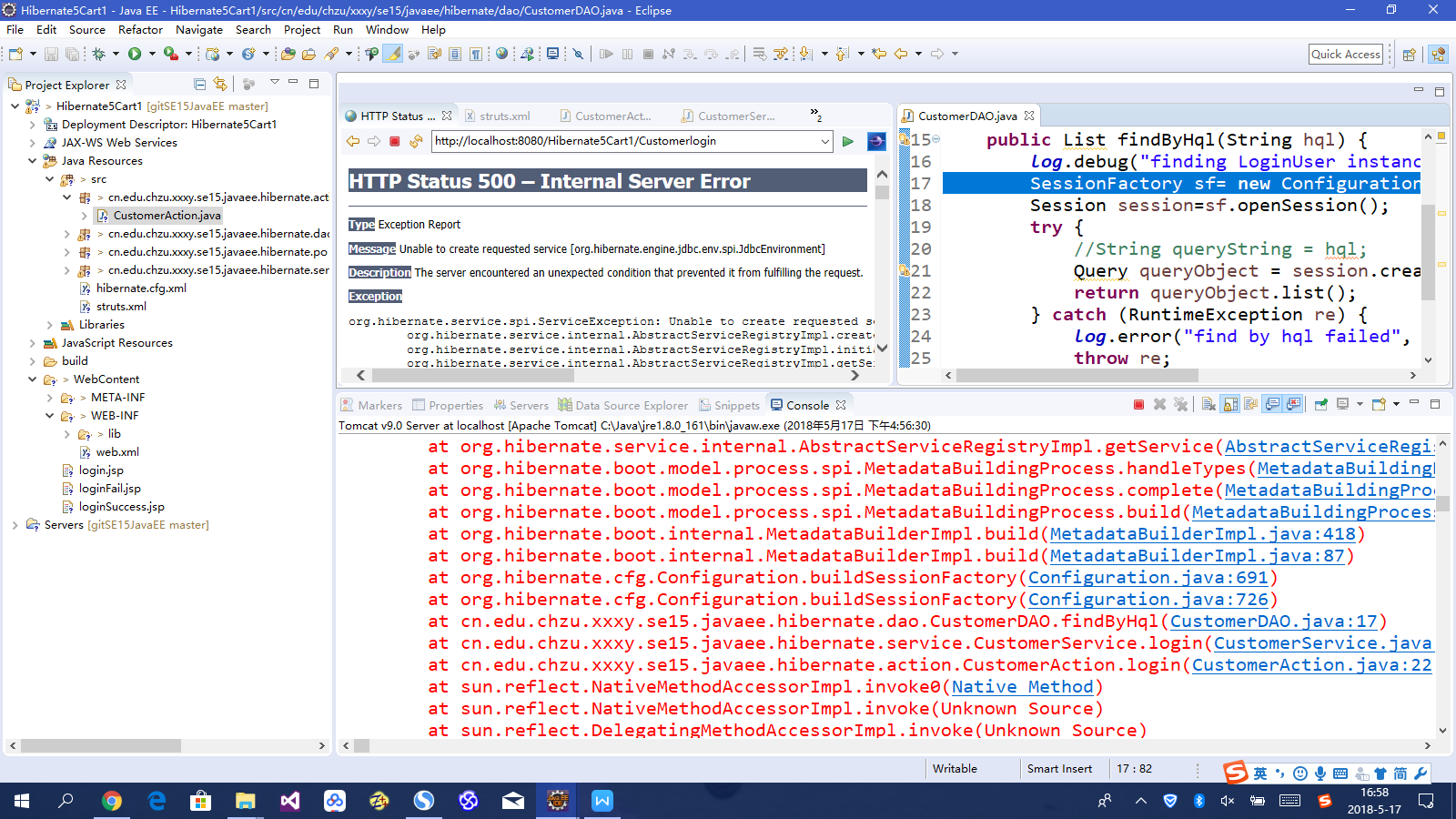
我的Struts.xml文件有错误！

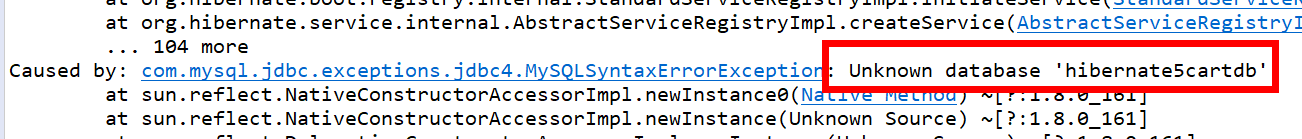


**Message** There is no Action mapped for namespace [/] and action name [Customerlogin] associated with context path [/Hibernate5Cart1].



还是出现错误：500——数据库连接问题，应该想到是Hibernate框架出问题了（注意我们始终没有像教材那样创建数据库、表和添加3条记录）

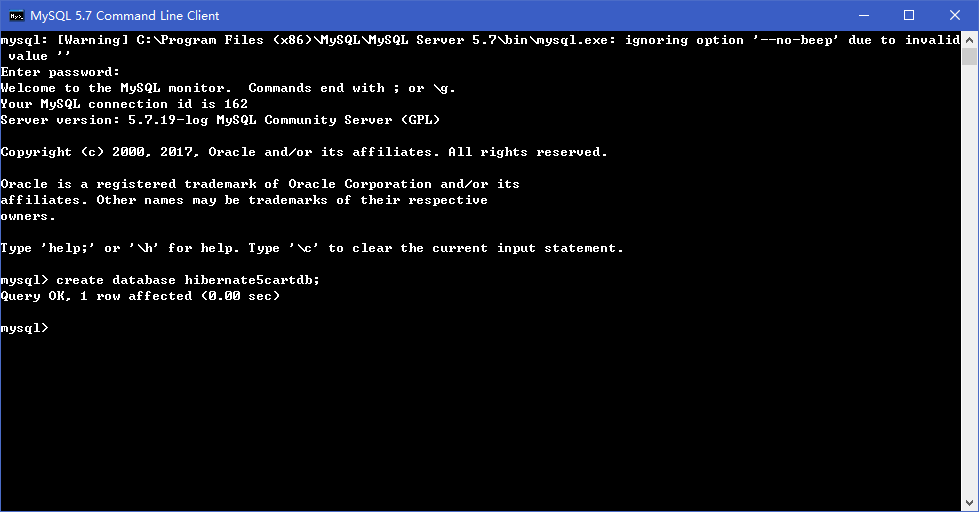




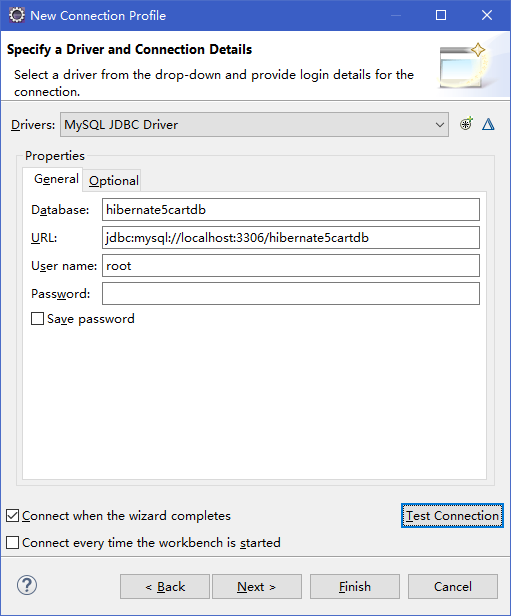
解决方法：手工创建数据库及表和数据

命令行工具成绩数据库：

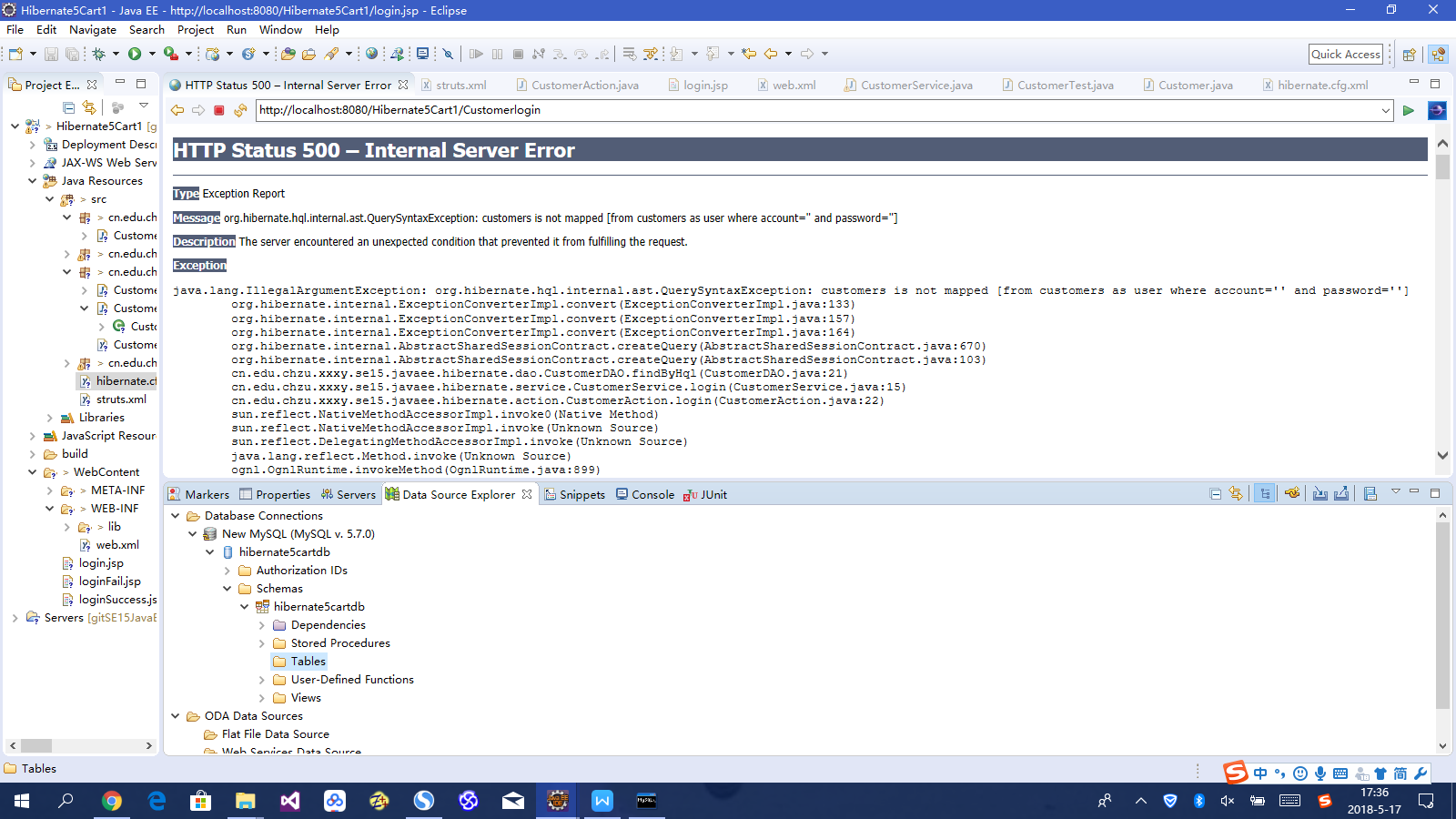
create database hibernate5cartdb;



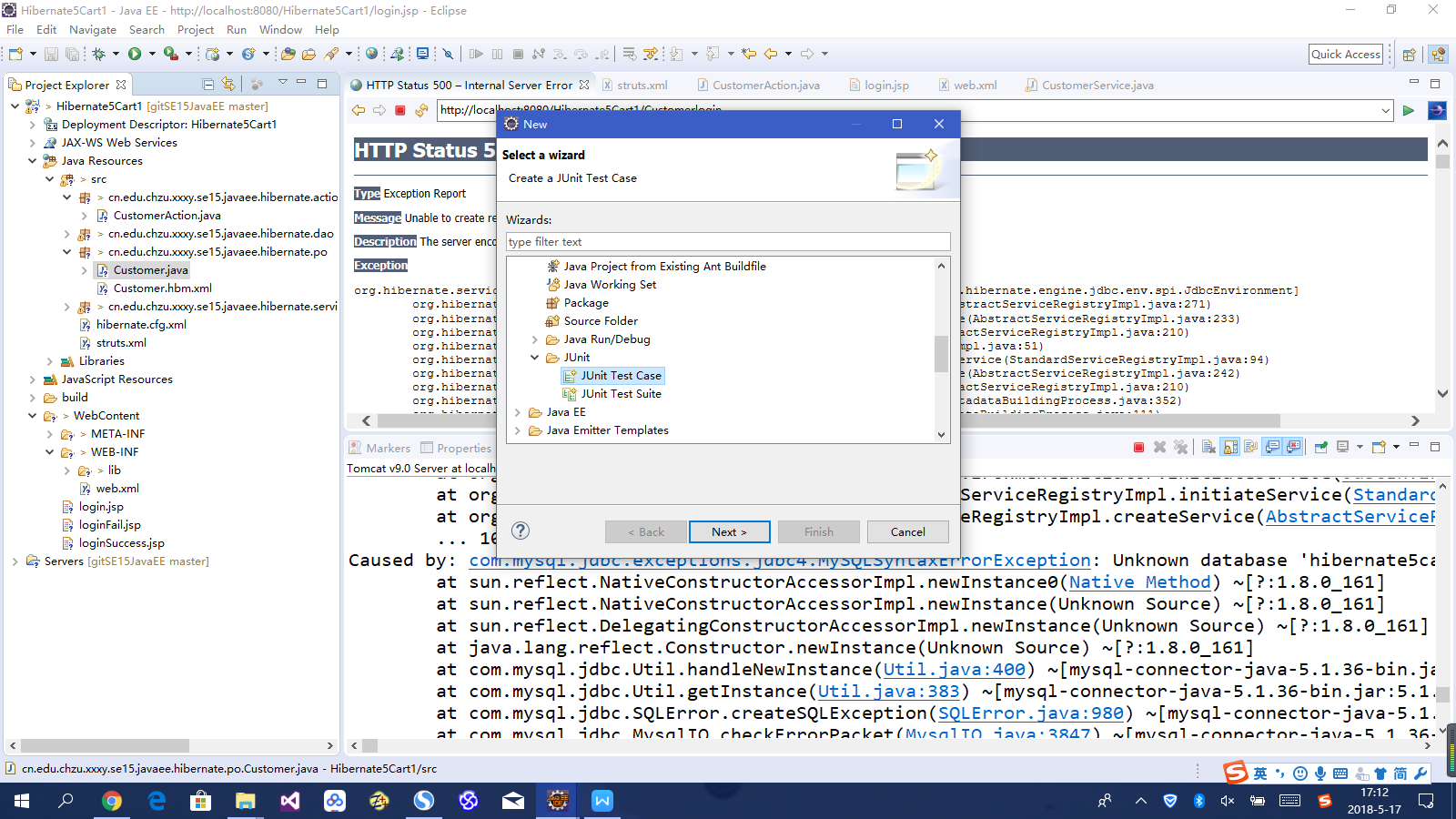
Eclipse链接数据库：

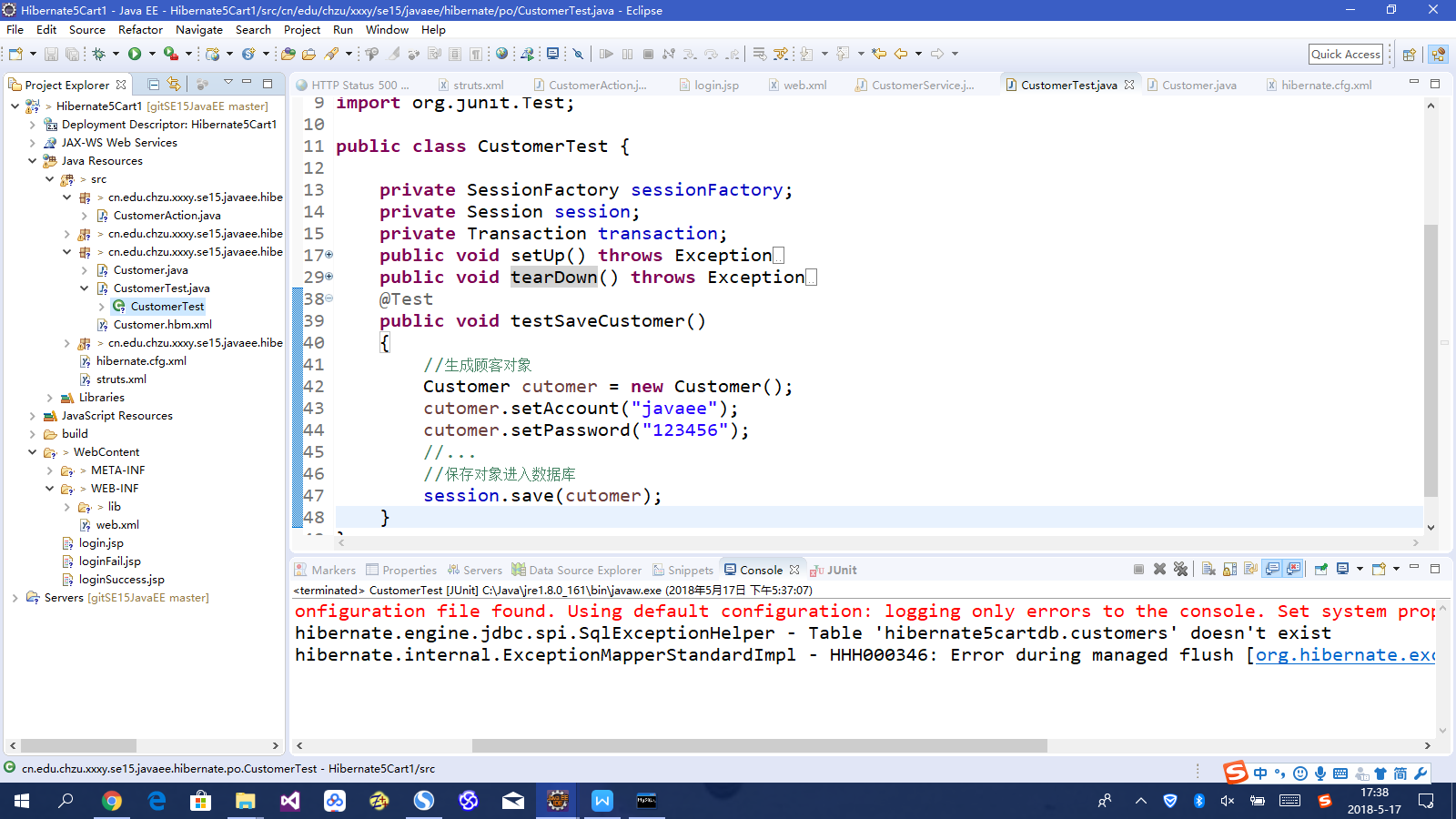


无表：customers



我们试试@Test：使用Junit进行测试，新建一个test包，在包下新建一个Junit测试类

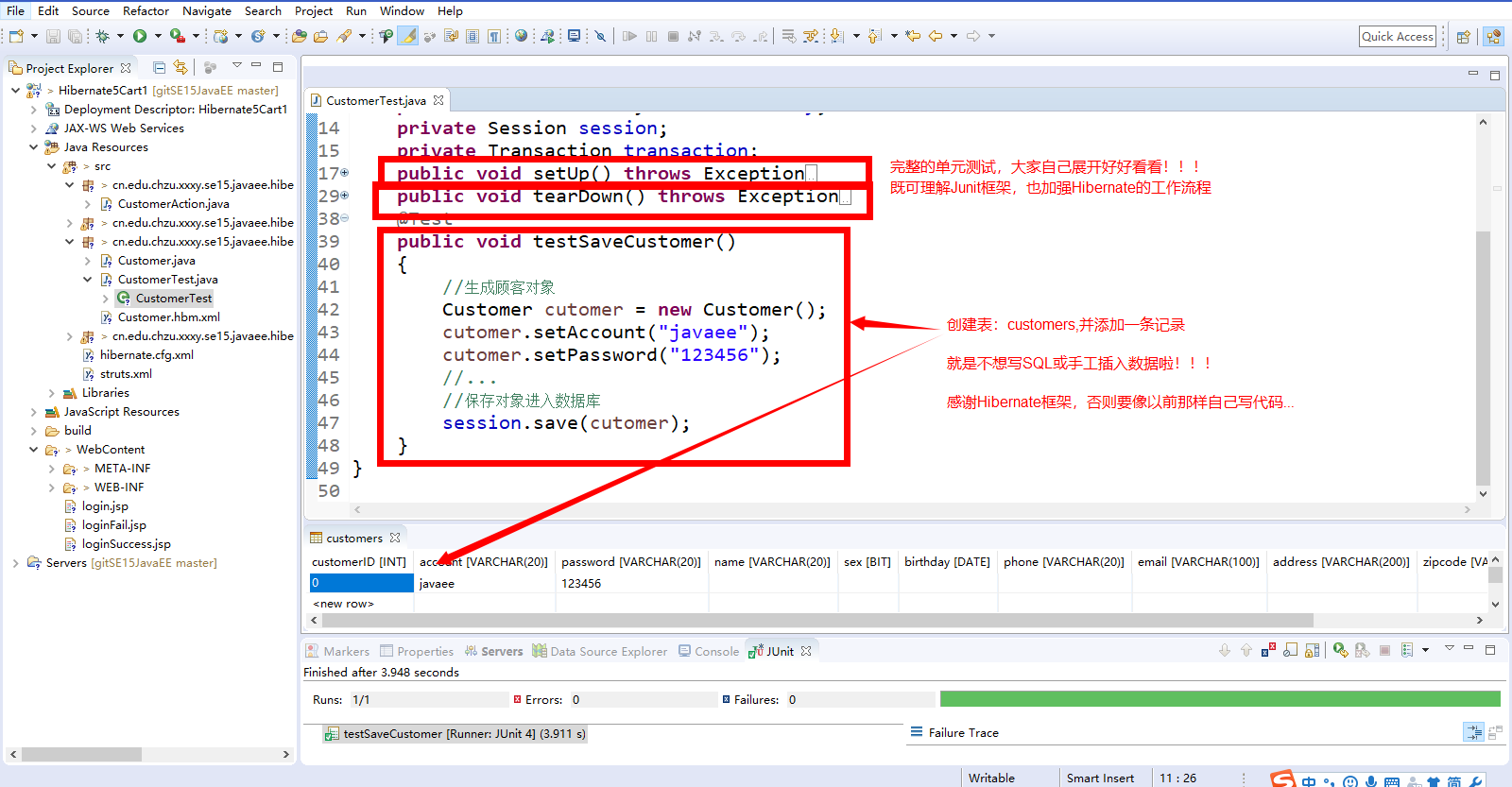




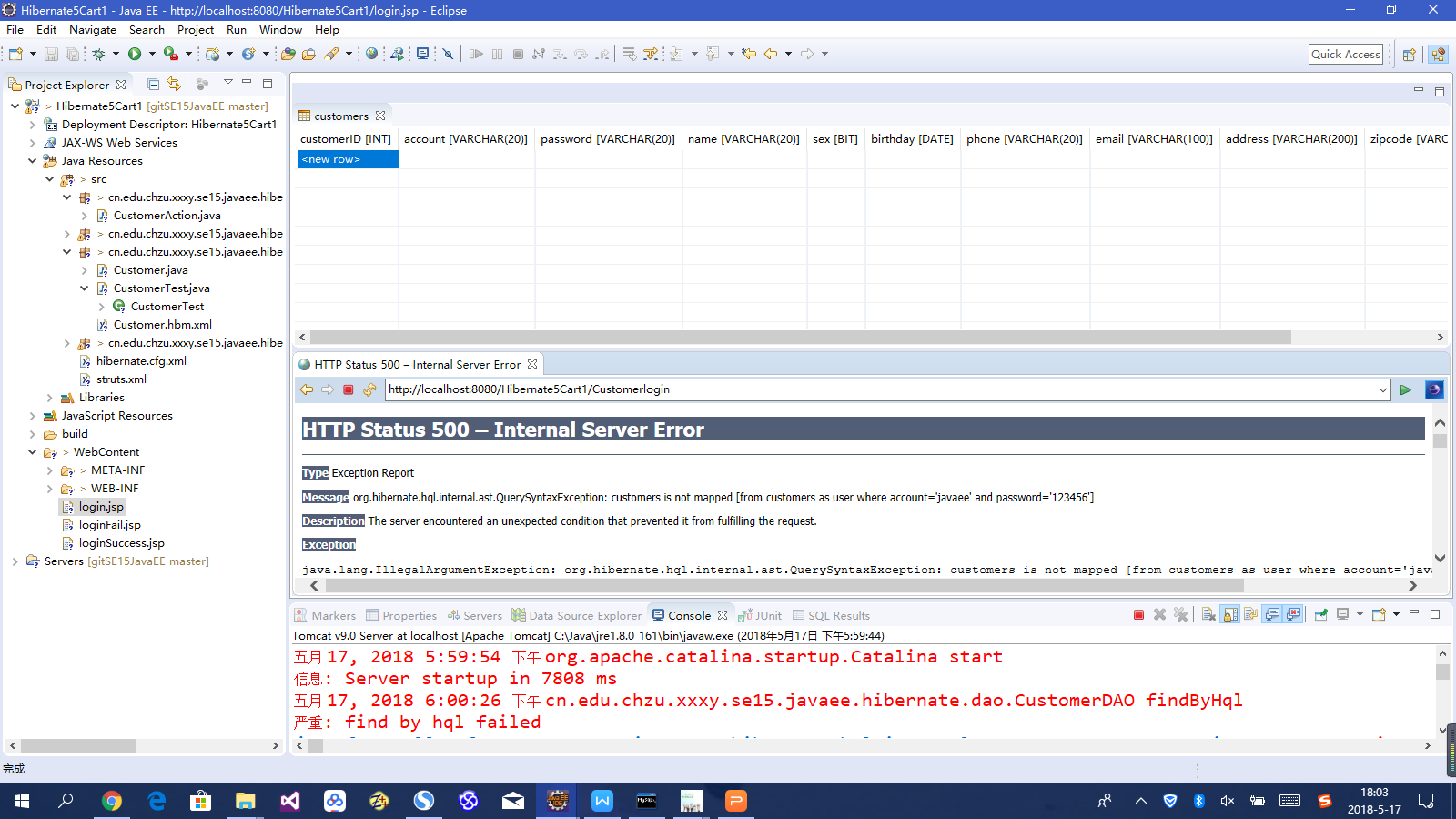
添加和修改hibernate的配置文件：

<property name=*"hbm2ddl.auto"*>create</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.MySQL5InnoDBDialect</property>



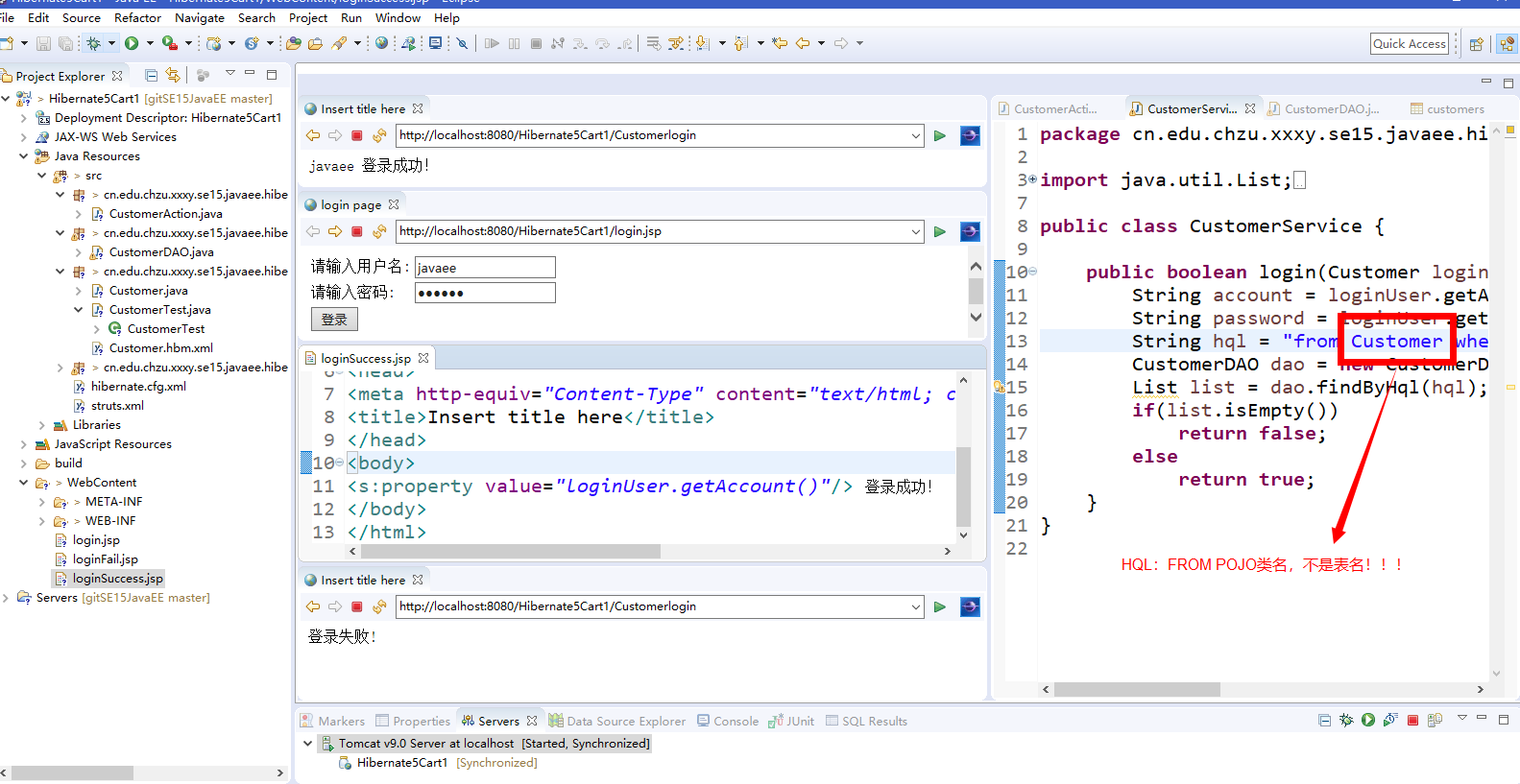
还是错误！执行登录时，将数据库里的记录删除了！



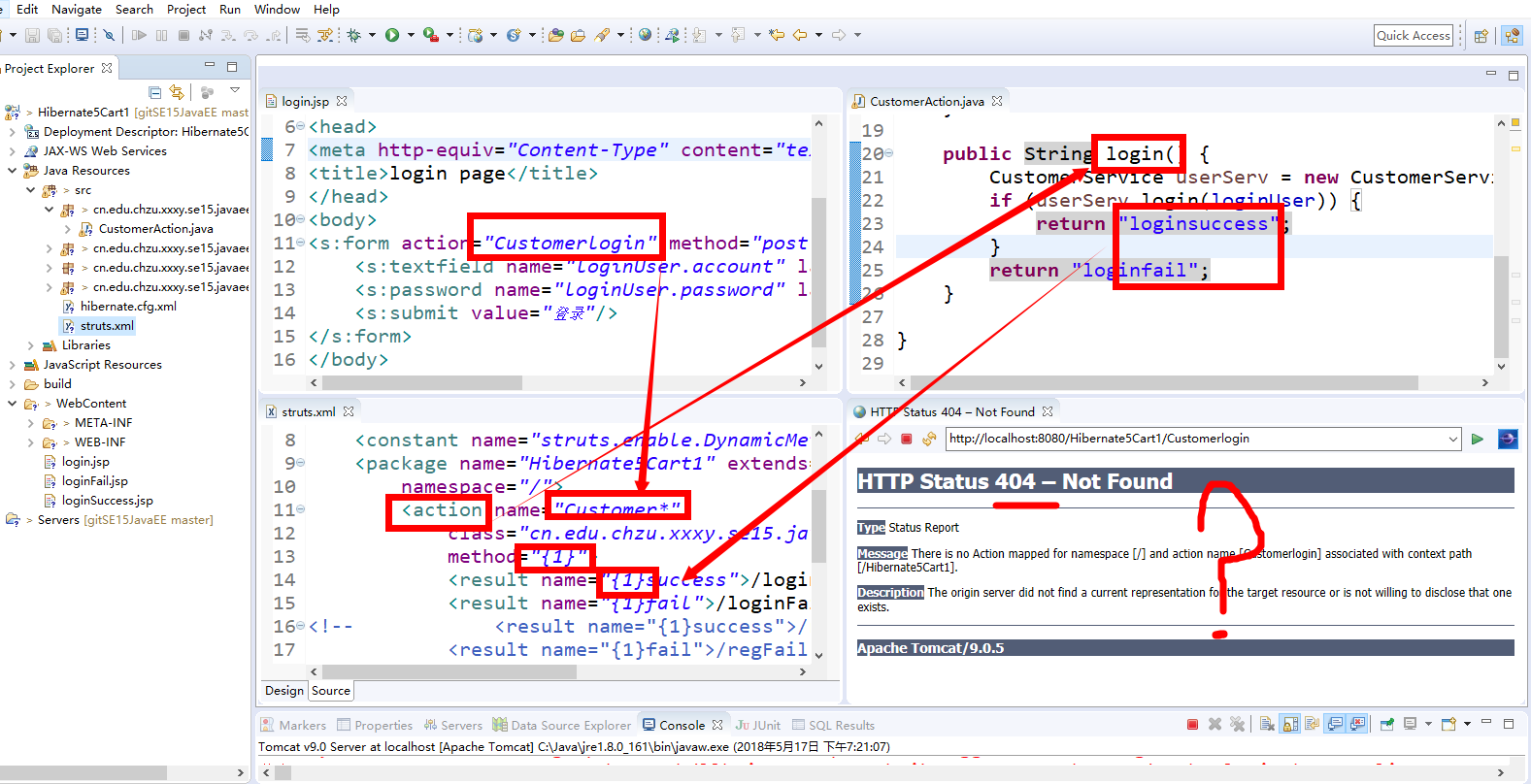
<property name=*"hbm2ddl.auto"*>update</property>

**Message** org.hibernate.hql.internal.ast.QuerySyntaxException: customers is not mapped [from customers where account='javaee' and password='123456']

终于成功啦！



最后的问题留给大家：使用通配符



**后记：**

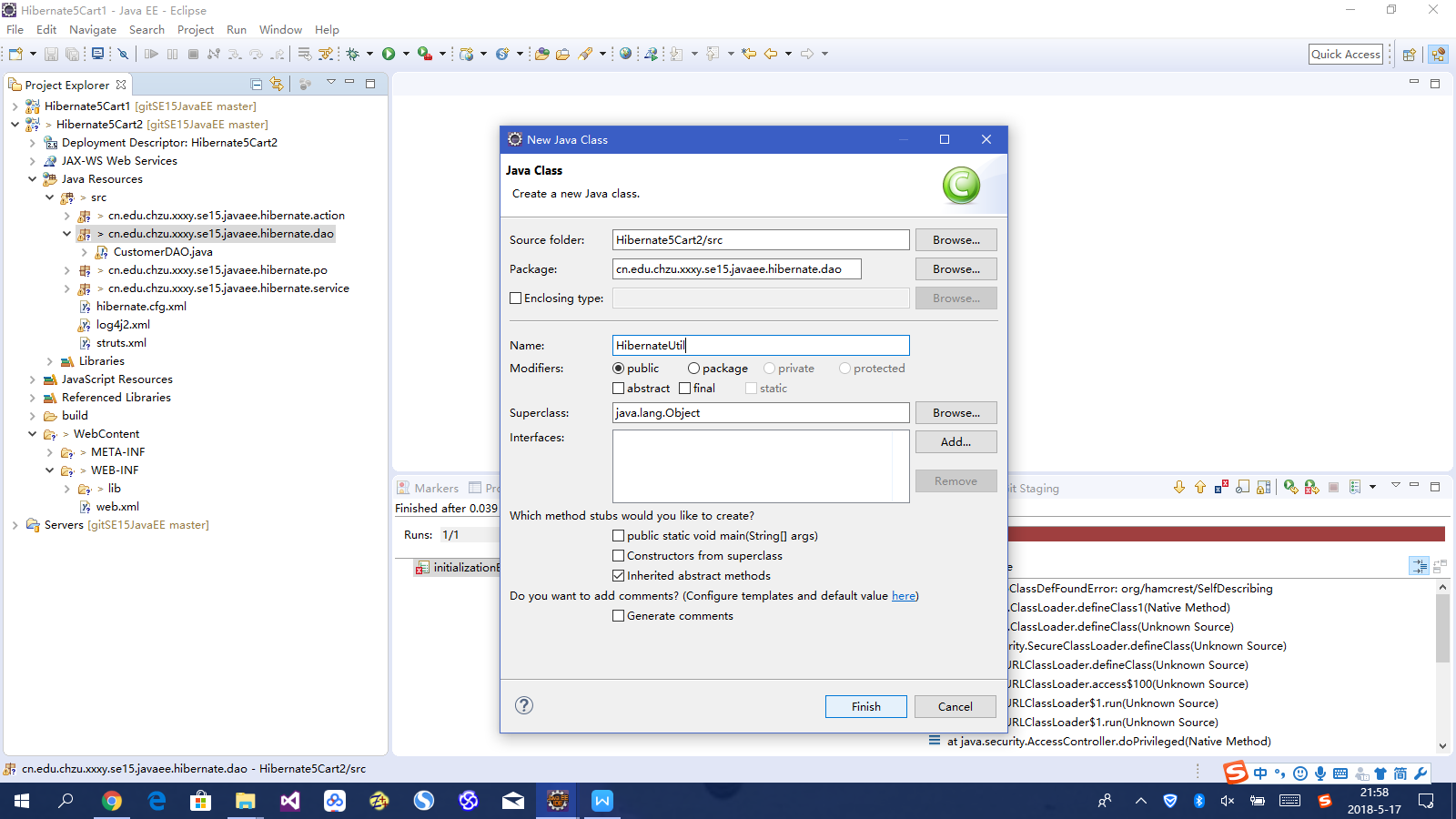
20多年了，总是想放弃，但不知道为什么，就是不甘心！于是，静下来，还是默默地做着自己的该做想做的事情！

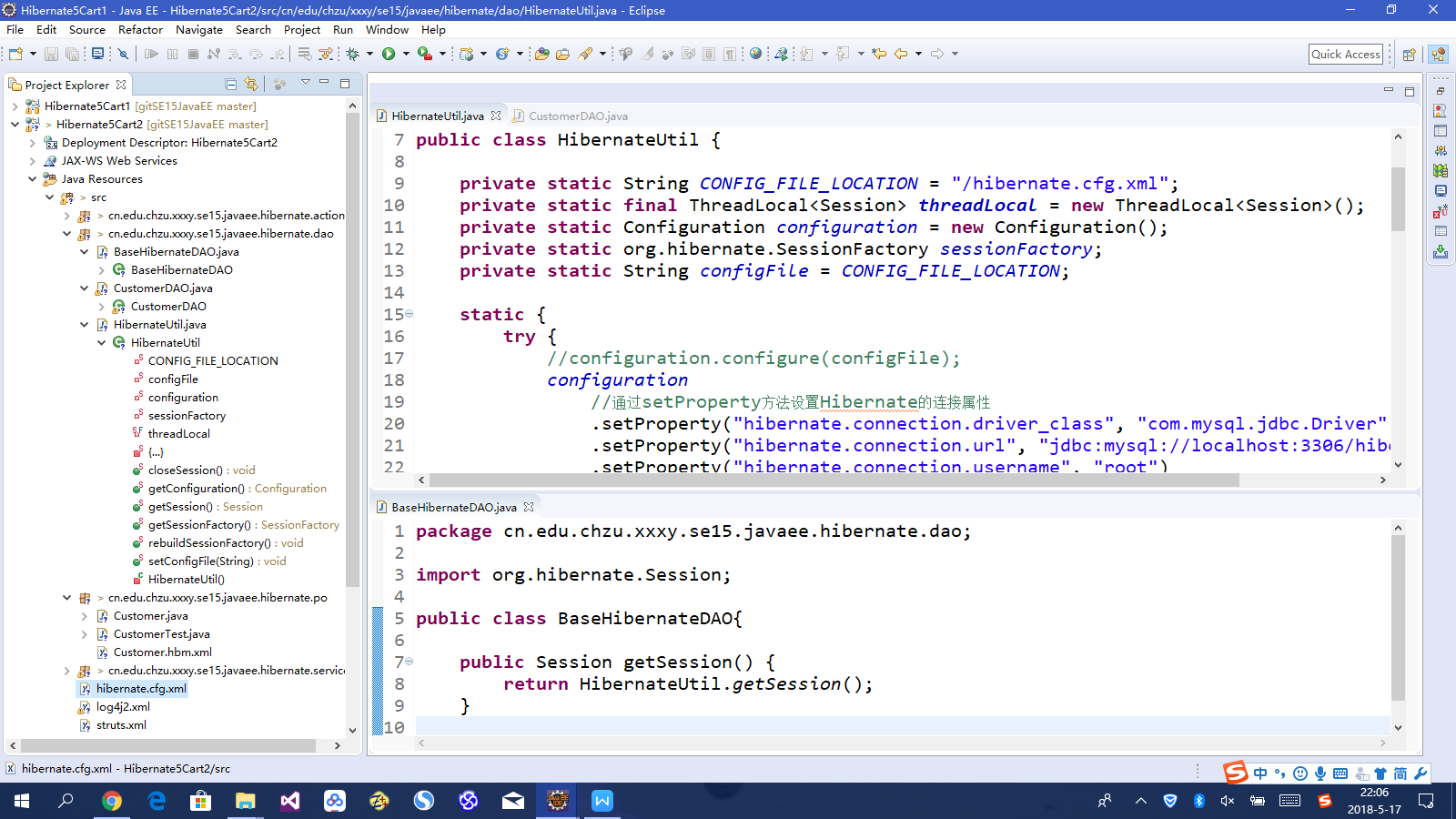
**-----------------------------------------实验五的基础实验-----------------------------------------**

1. **新建Hibernate5Cart2项目，复制Hibernate5Cart1的文件**

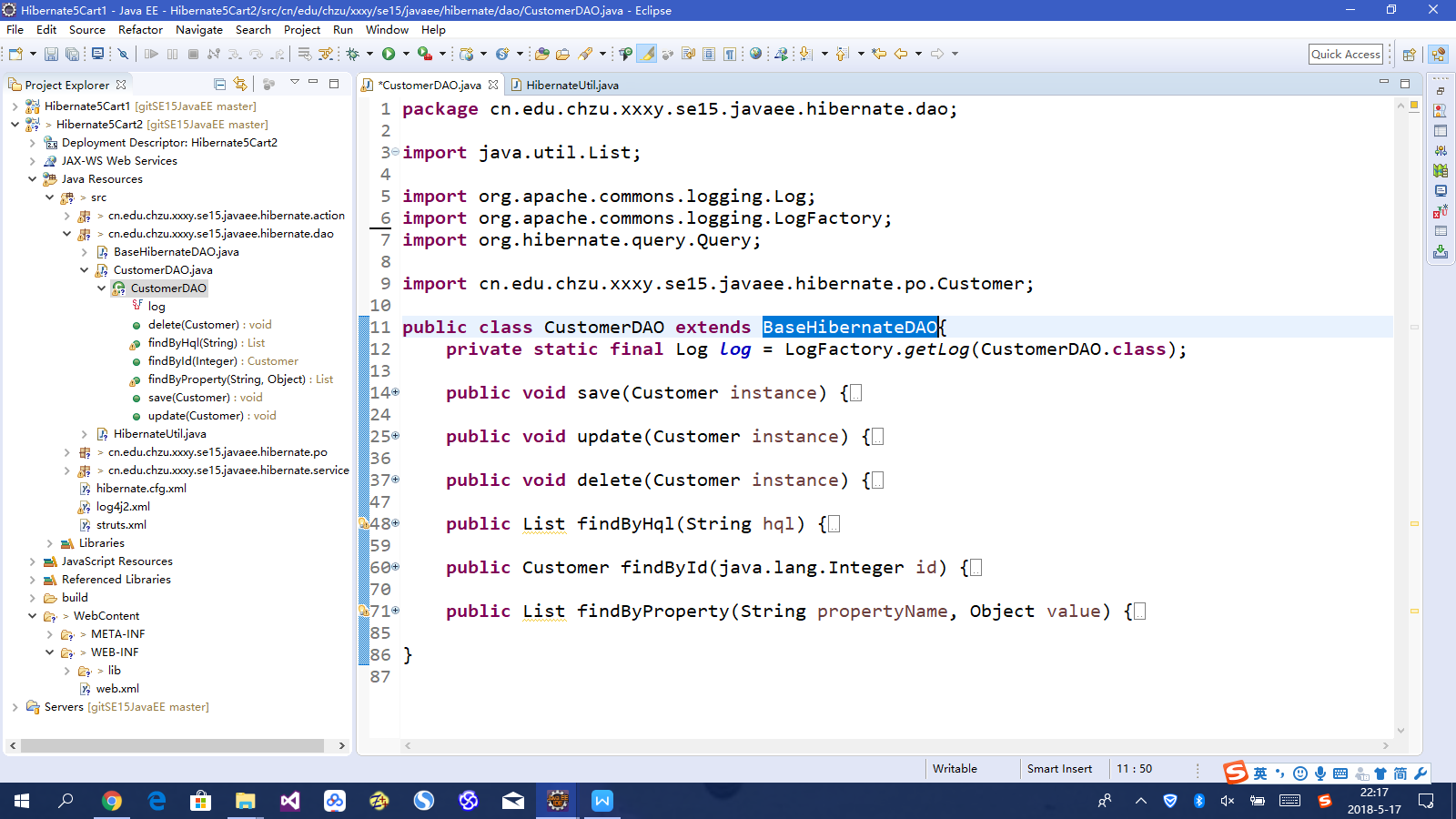
**主要是src、lib、WebContent目录下的文件。**

1. **创建DAO辅助类HibernateUtil、BaseHibernateDAO**

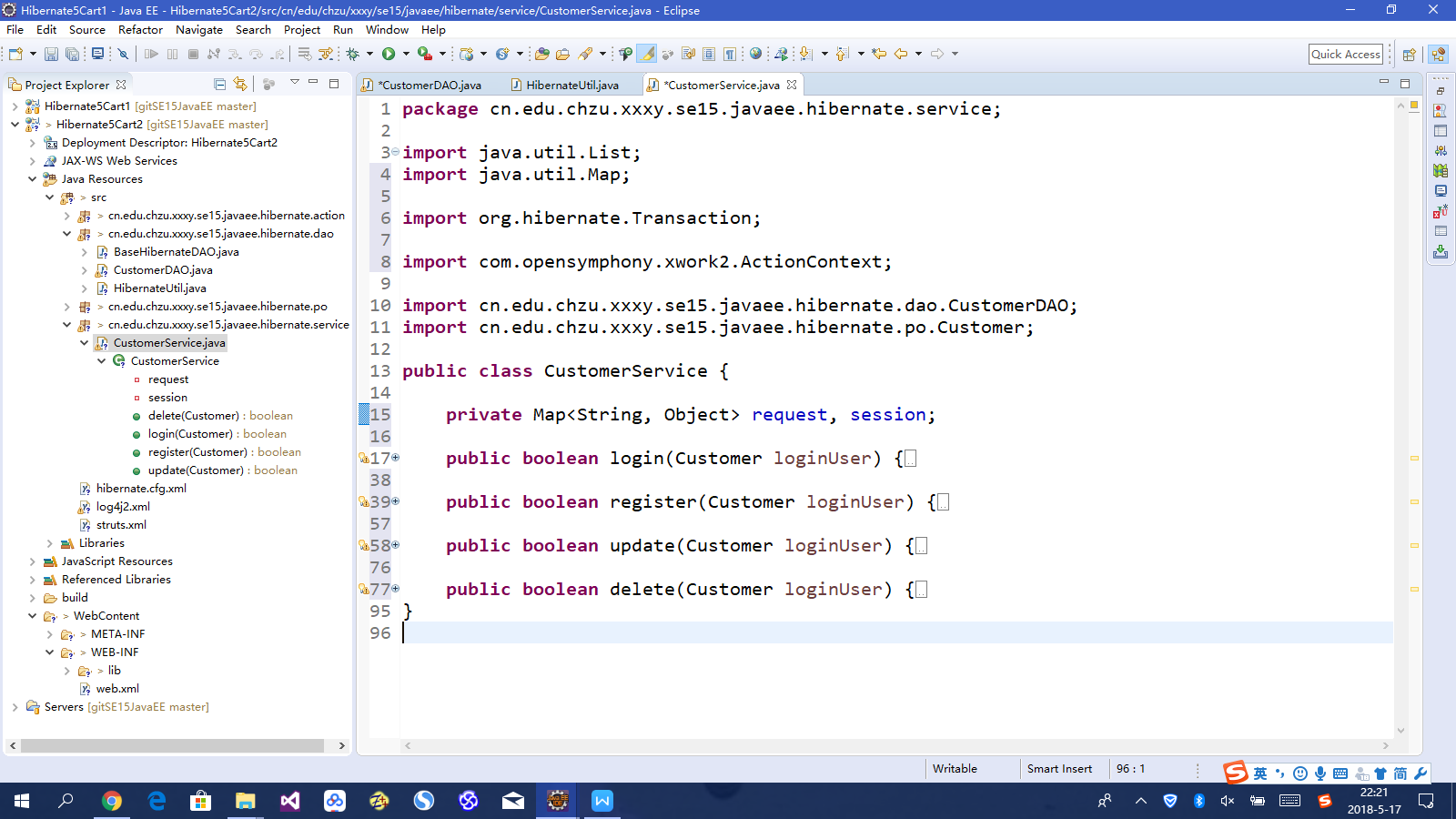




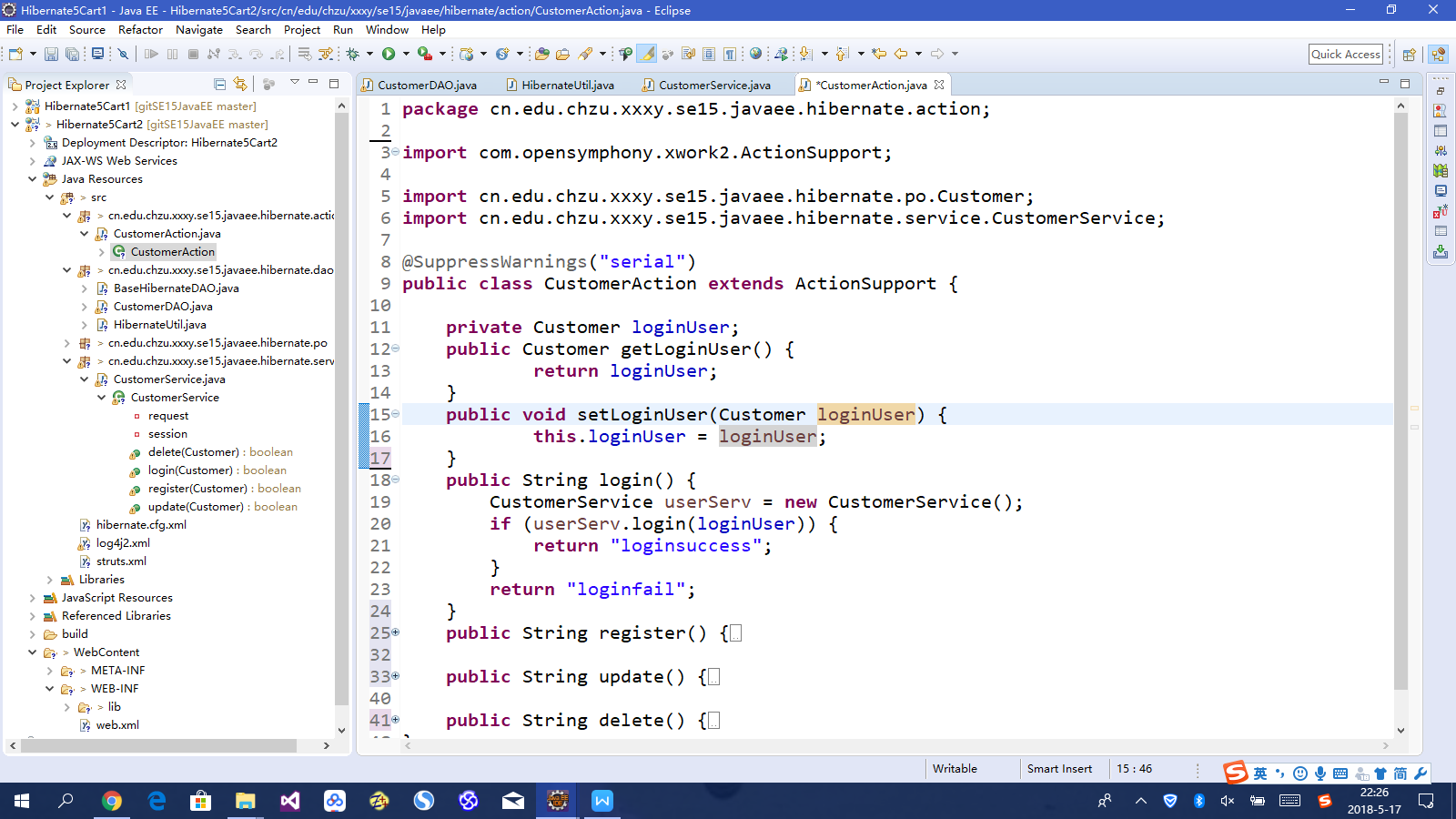
1. **改写CustomerDAO类并继承BaseHibernateDAO**



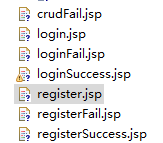
1. **改写CustomerService类**



1. **改写**CustomerAction**类**



1. **编写JSP页面**



1. **编写struts.xml配置文件**

<?xml version=*"1.0"* encoding=*"UTF-8"* ?>

<!DOCTYPE struts PUBLIC

"-//Apache Software Foundation//DTD Struts Configuration 2.5//EN"

"http://struts.apache.org/dtds/struts-2.5.dtd">

<struts>

<package name=*"Hibernate5Cart2"* extends=*"struts-default"*

namespace=*"/Customer"*>

<action name=*"Customer\*"* class=*"cn.edu.chzu.xxxy.se15.javaee.hibernate.action.CustomerAction"* method=*"{1}"*>

<result name=*"{1}success"*>/{1}Success.jsp</result>

<result name=*"{1}fail"*>/{1}Fail.jsp</result>

<result name=*"updatesuccess"*>/crudSuccess.jsp</result>

<result name=*"updatefail"*>/crudFail.jsp</result>

<result name=*"deletesuccess"*>/login.jsp</result>

<result name=*"deletefail"*>/crudFail.jsp</result>

<!-- Thank Sun -->

<allowed-methods >login,register,update,delete</allowed-methods>

</action>

</package>

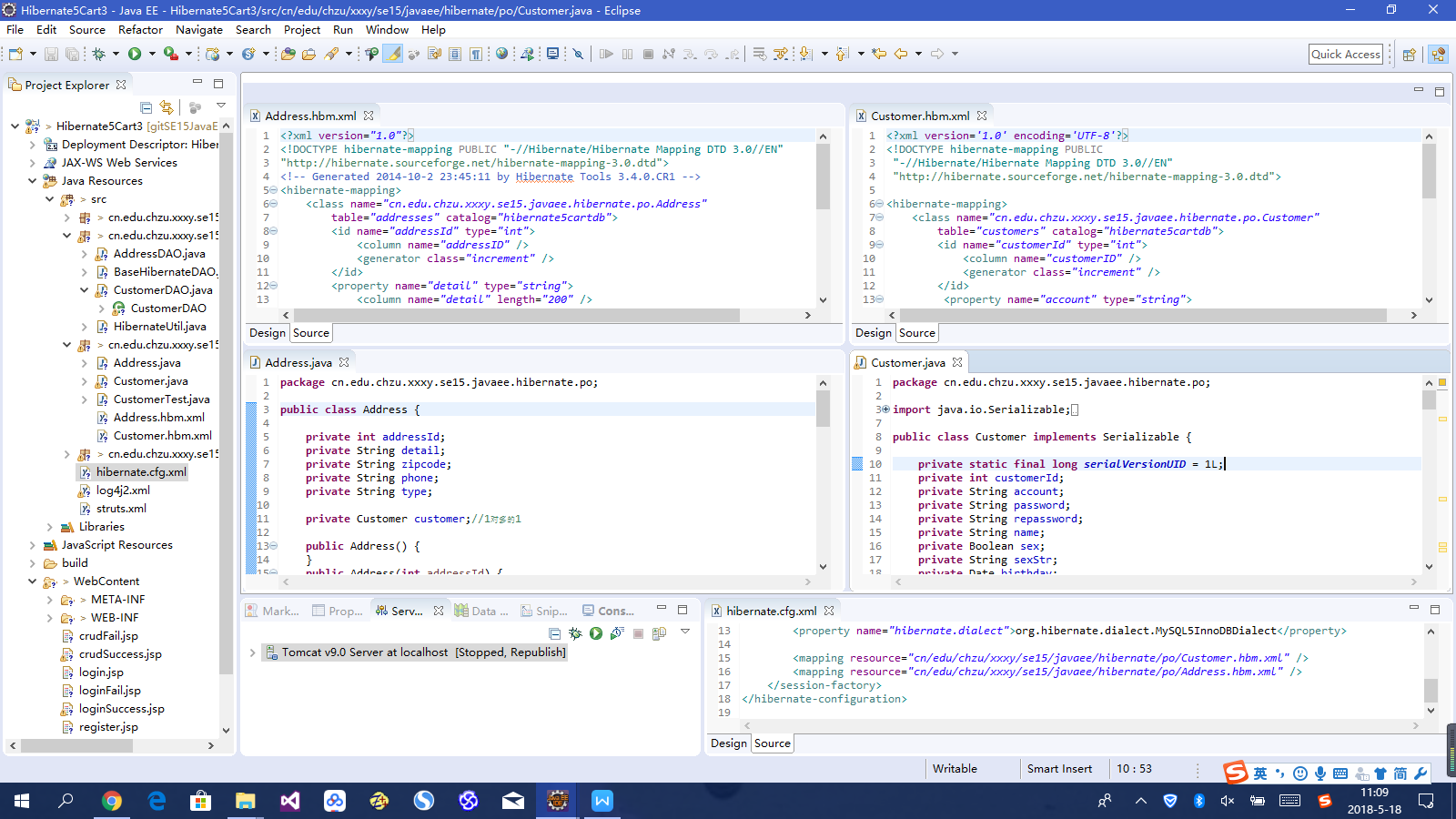
</struts>

1. **增删改查CRUD成功！下载测试吧**



**-----------------------------------------实验六的基础实验-----------------------------------------**

1. **修改**CustomerPO类、增加AddressPO类及其映射文件及配置文件



<many-to-one name=*"customer"* class=*"cn.edu.chzu.xxxy.se15.javaee.hibernate.po.Customer"*

fetch=*"select"*>

<column name=*"cust\_id"* />

</many-to-one>

<set name=*"addresses"* inverse=*"true"* cascade=*"all"* lazy=*"false"*>

<key column=*"cust\_id"*/>

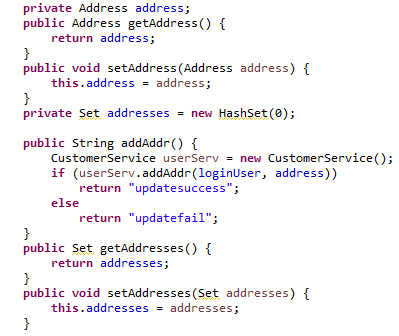
<one-to-many class=*"cn.edu.chzu.xxxy.se15.javaee.hibernate.po.Address"*/>

</set>

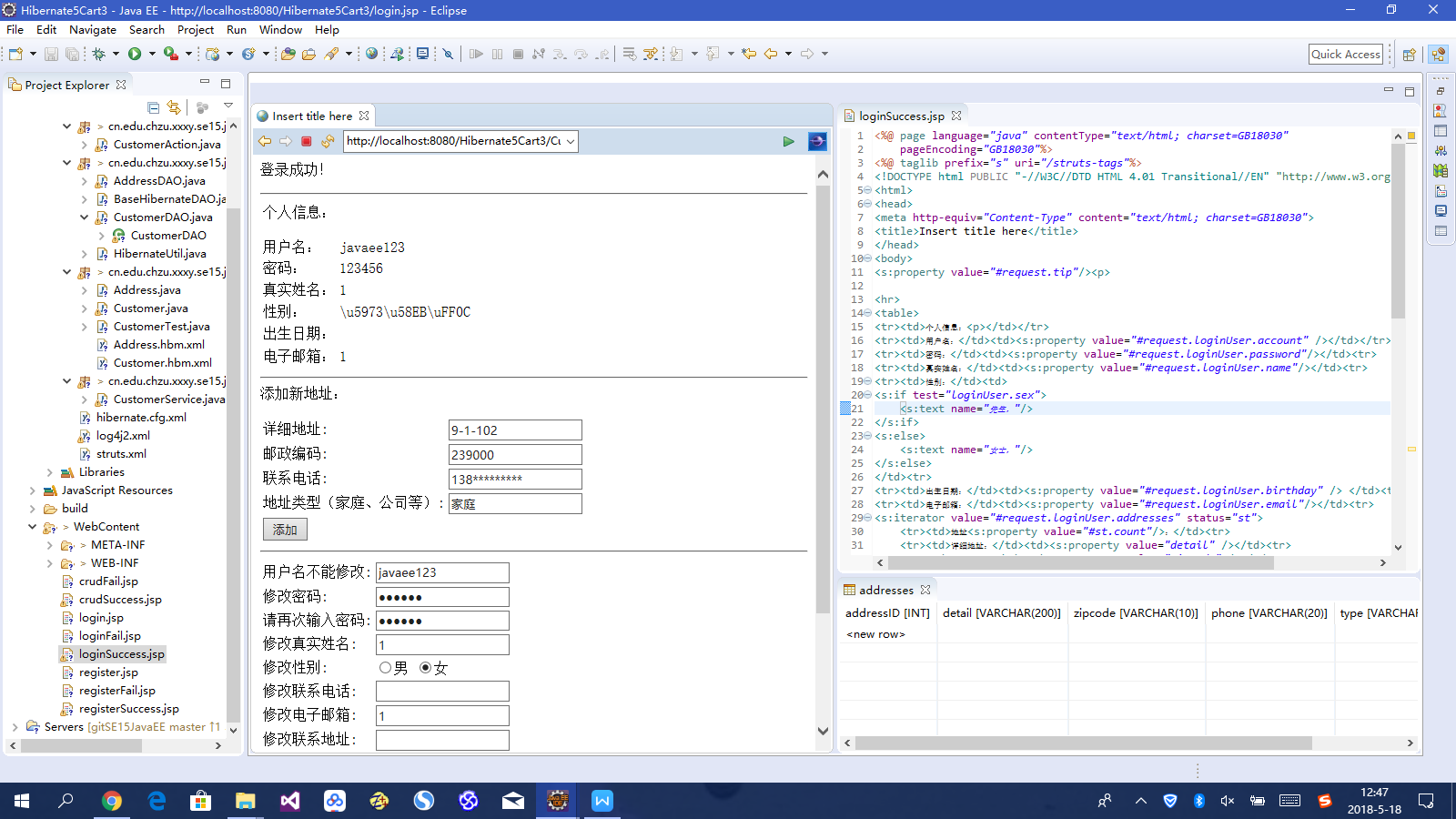
1. **增加AddressDAO类，参考CustomerDAO，略**
2. **增加addAddr方法到CustomerService类中**



1. **添加addAddr方法到**CustomerAction类中



1. **添加loginSuccess.jsp页面并测试**





**-----------------------------------------实验七的基础实验-----------------------------------------**