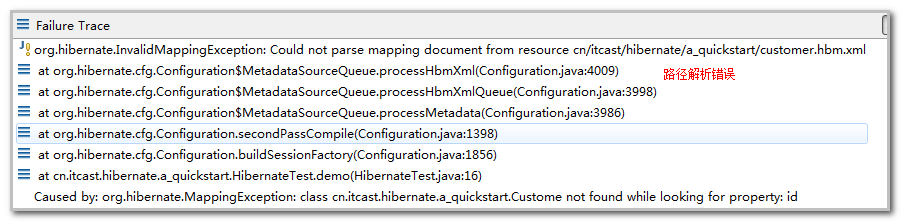
Hibernate问题汇总

# 第一天错误

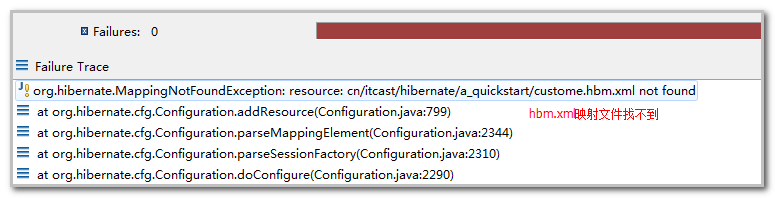
## Hbm.xml映射配置文件路径问题



问题：可能是hbm.xml映射文件配置的实体类路径有问题。

解决方案：检查hbm.xml中的实体类的路径是否正确

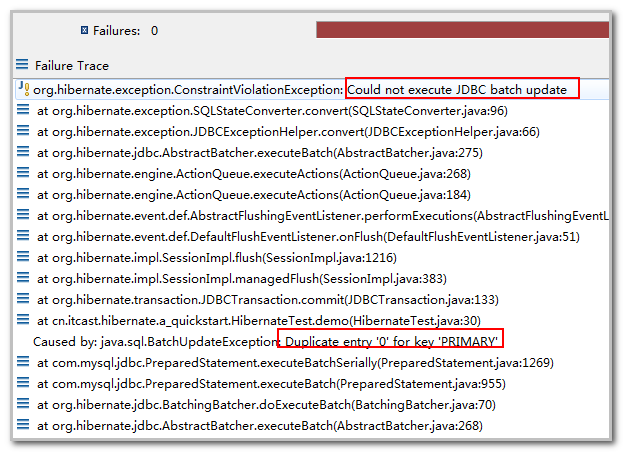
## Hibernate.cfg.xml



问题：hibernate.cfg.xml中配置的hbm.xml映射文件找不到

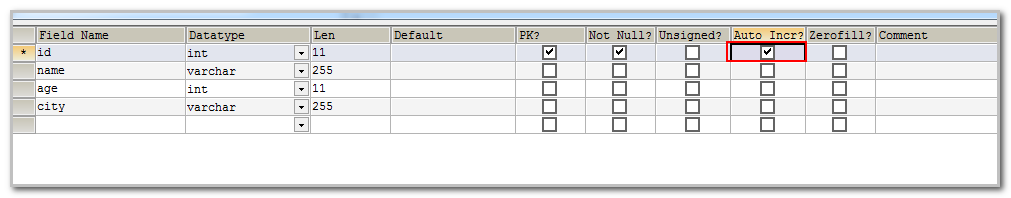
解决方案：检查hibernate.cfg.xml中hbm.xml的配置路径是否正确

## 数据库表设计错误

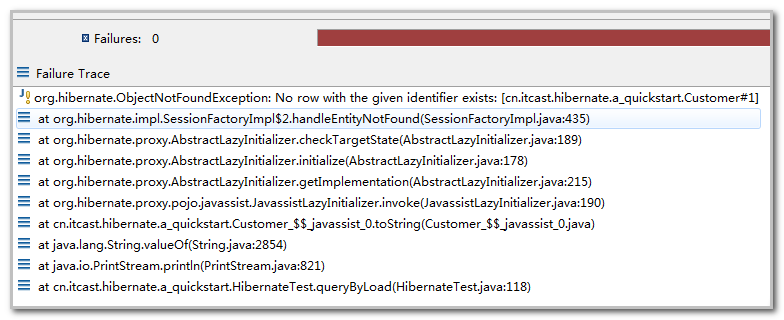


问题：不设置id无法新增数据

解决方案：查看表设计，是否设置了id自动增长。



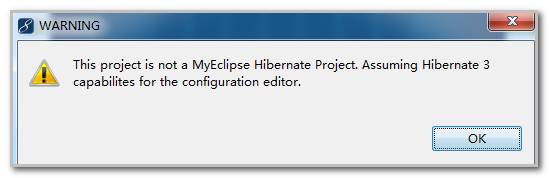
## 查询语句错误



问题：通过load方法进行查询，但是查询的主键不存在。

解决方案：确定Customer customer = (Customer) session.load(Customer.**class**, 1);语句中的红色主键1是否存在。或者可以使用get方法进行查询

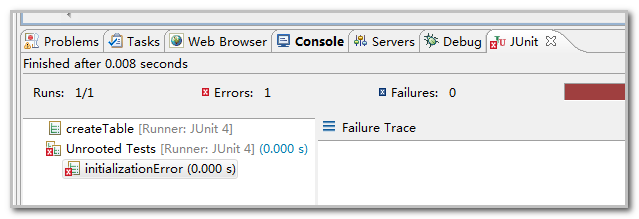
## Myeclipse配置错误



解决方案：

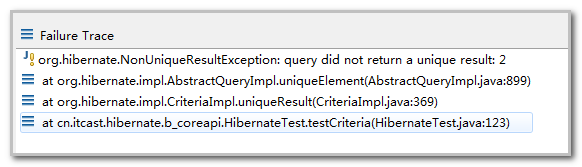
项目名上右键--〉myeclipse-->add hibernate capabilites -->next-->hibernate config file --> existing -->选择现有工程存在的hibernate配置文件--> next --> 不生成factory class --> end

## 测试报错



原因：没有添加测试方法上没有添加@Test

## 条件查询错误

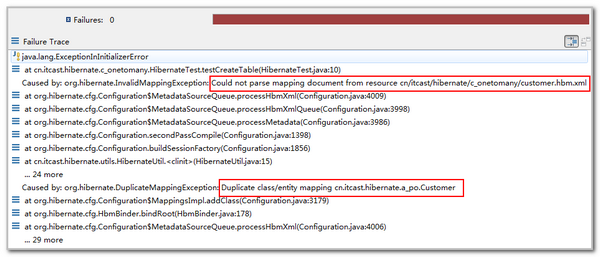


原因：查询时使用了uniqueResult(),然并卵，查询结果并不只有一个。

解决方案：查看数据库是否自己要查询的数据是否只有一个，如果不是请选择list查询。

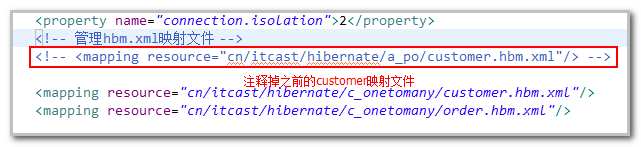
# 第二天错误

## 一对多建表测试失败原因1

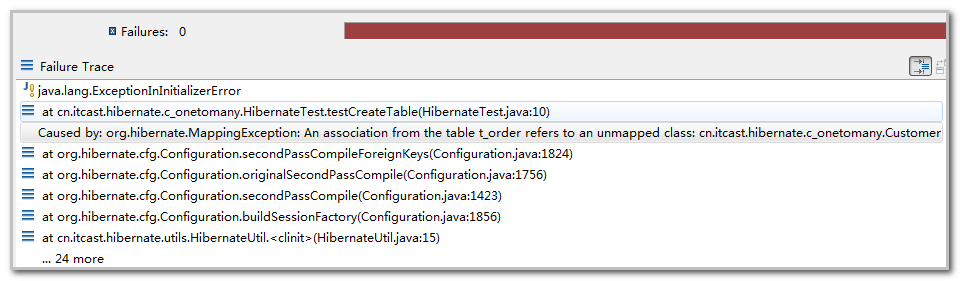


原因：有两个需要建相同表的映射文件，比如之前的customer映射文件和新建的一对多测试的customer映射文件

解决方案：查看hibernate.cfg.xml，将不需要的映射文件注释掉

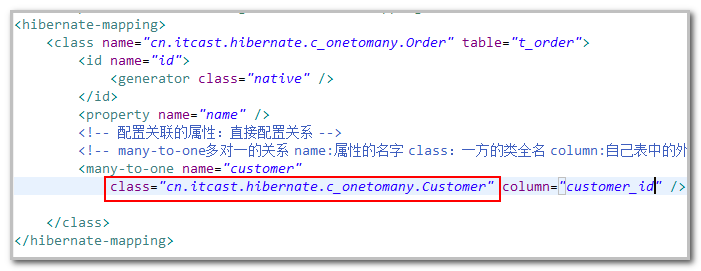


## 一对多建表测试失败原因2

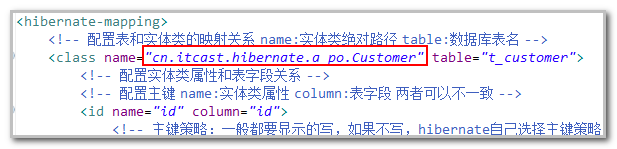


原因：多方找不到一方的类路径

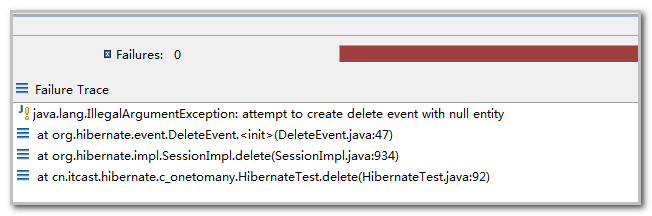
多方如果配置无误：那么很可能是一方映射文件中的cutomer路径不对，可能之前的哥们复制了以前的customer映射文件，结果没有修改类路径导致



配错路径了，配置成以前的customer



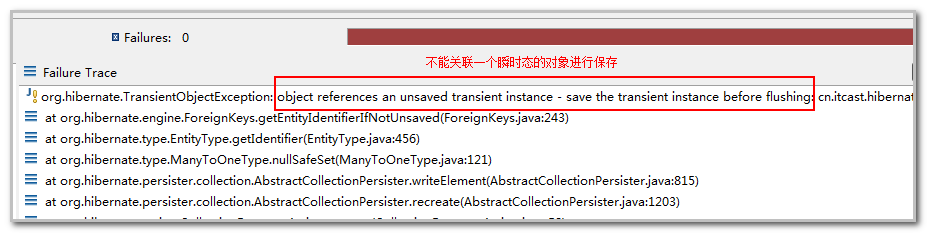
## 级联删除



原因：删除时，数据库中没有找到对应的id

解决方案：检查需要删除的id是否正确

## 关联保存

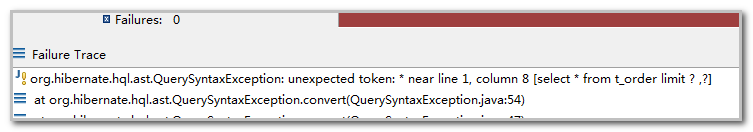


原因：不允许持久态对象关联瞬时态对象

解决方案：要么互相关联一下，要么配置级联

# 第三天错误

## 分页查询错误1



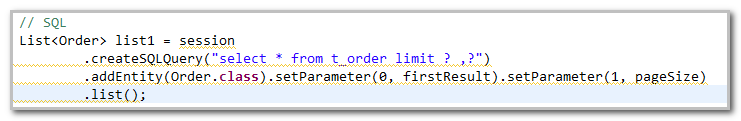
原因：查询语句使用错误

解决方案：检查查询语句是否使用正确。

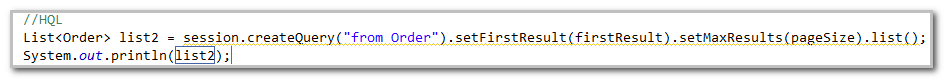
例如：下面的语句用的是HQL，但是HQL是没有limit用法的。



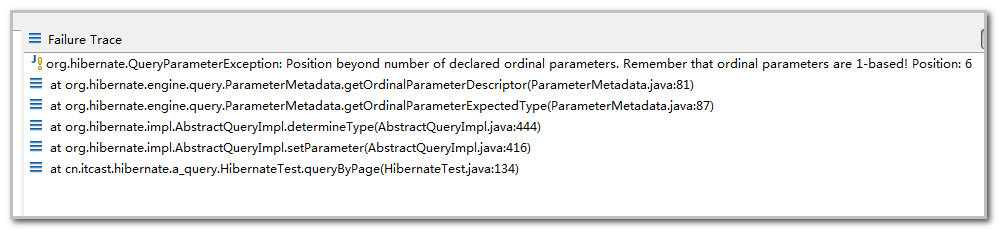
可以修改成SQL用法：



也可以改成HQL用法:



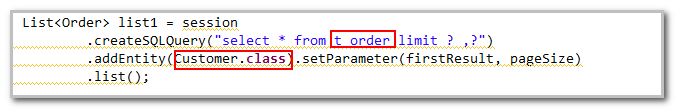
## 分析查询错误2



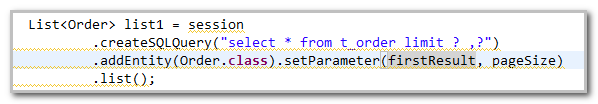
原因：查询语句参数错误

解决方案：检查查询语句的参数是否正确

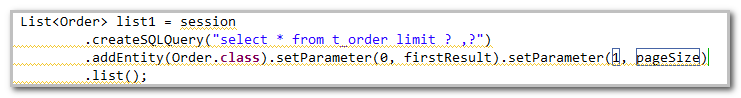
例如：明明是查询order表，而封装的实体参数使用的是customer



又例如：明明需要两个setParameter通过序号来传递参数，却只用了一个

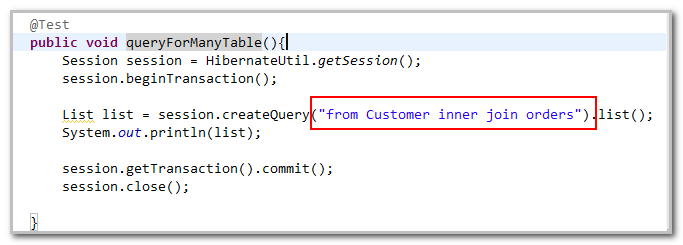


正确的如下：

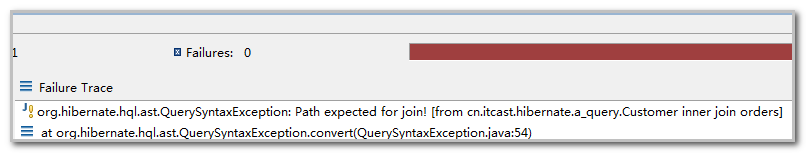


## 多表关联查询

查询语句：

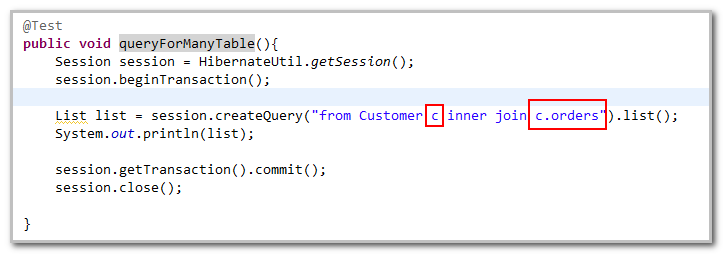


报错：



原因：找不到orders

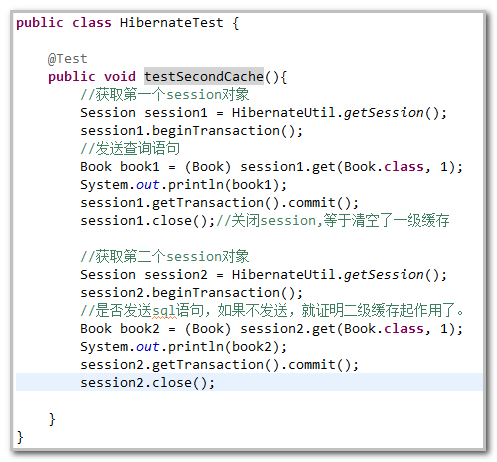
解决方案：添加一个别名即可



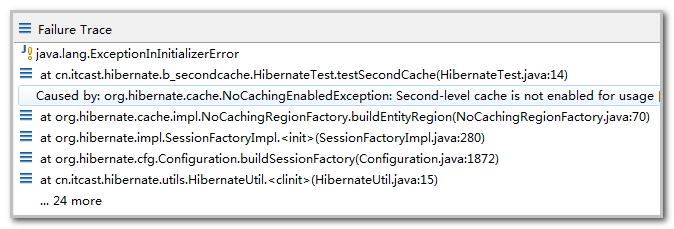
# 第四天错误

## 测试二级缓存是否有效1

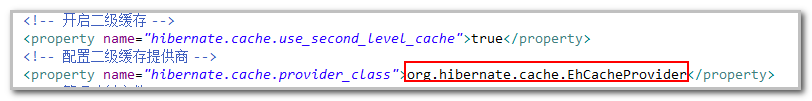
测试二级缓存是否有效：



报错如下：

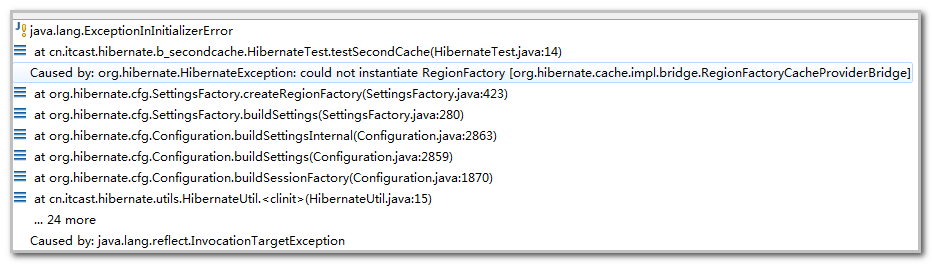


原因：二级缓存提供商没有配置



## 测试二级缓存是否有效2

报错如下：



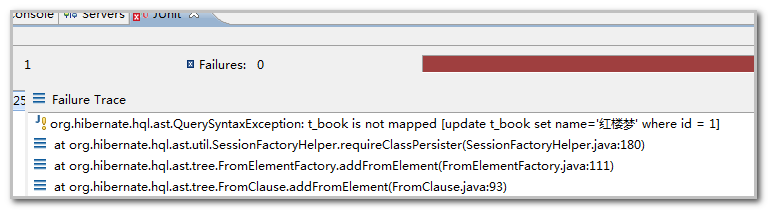
原因：没有导入jar二级缓存所需包

## 更新时间戳测试

测试时间戳：



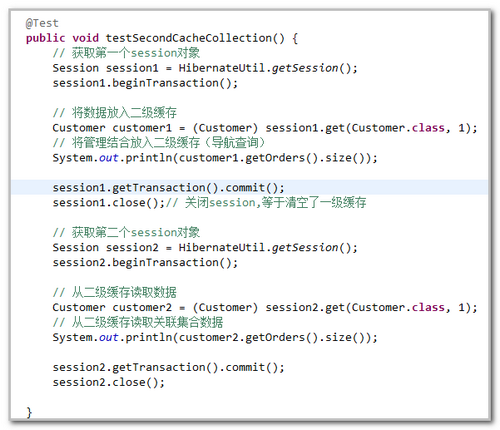
报错如下：



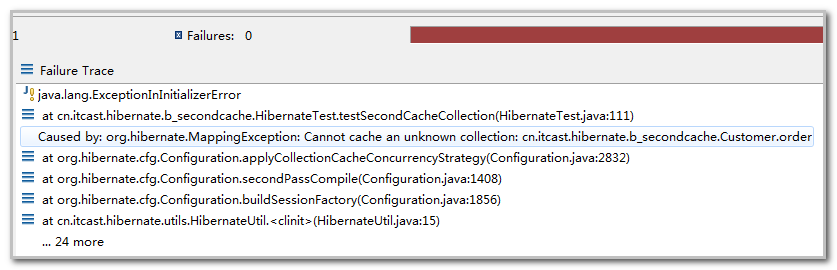
原因及解决方案：使用HQL查询针对的是对象，因此查询语句中不能这对t\_book这个表，而是应该改成Book这个对象来进行查询。

## 关联集合缓存测试

测试方法：

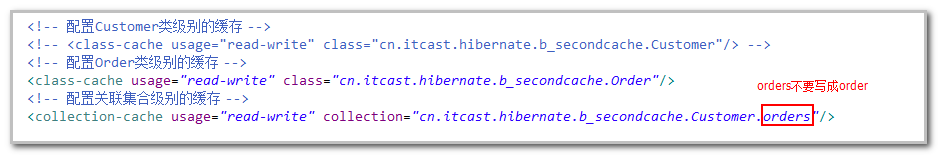


报错如下：



原因：映射文件路径错误

解决方案：检查映射文件的class路径是否正确,比如以下：



## 缓存监控

测试二级缓存的性能。

开启二级缓存监控：

<!-- 开启性能检测 -->

<property name=*"hibernate.generate\_statistics"*>true</property>

测试方法：

@Test

//缓存使用率性能监控(前提是开启了性能监视)

**public** **void** testStatistics(){

SessionFactory sessionFactory = HibernateUtil.*getSessionFactory*();

//通过会话工厂，获取统计对象

Statistics statistics = sessionFactory.getStatistics();

//初始状态下，命中次数和丢失次数都是0

System.*out*.println("二级缓存的命中次数:"+statistics.getSecondLevelCacheHitCount());

System.*out*.println("二级缓存的丢失次数:"+statistics.getSecondLevelCacheMissCount());

Session session1 = sessionFactory.openSession();

//原理讲解

//查询前2条订单信息，放入二级缓存

session1.createQuery("from Book where id<=2").list();

session1.close();

System.*out*.println("-换一个session-------------------------");

Session session2 = sessionFactory.openSession();

//查询1号订单

Book book1 = (Book) session2.get(Book.**class**, 1);

System.*out*.println(book1);

Book book2 = (Book) session2.get(Book.**class**, 2);

System.*out*.println(book2);

System.*out*.println("二级缓存的命中次数:"+statistics.getSecondLevelCacheHitCount());

System.*out*.println("二级缓存的丢失次数:"+statistics.getSecondLevelCacheMissCount());

System.*out*.println("-------------------------");

//查询1号订单

Book book4 = (Book) session2.get(Book.**class**, 3);

System.*out*.println(book4);

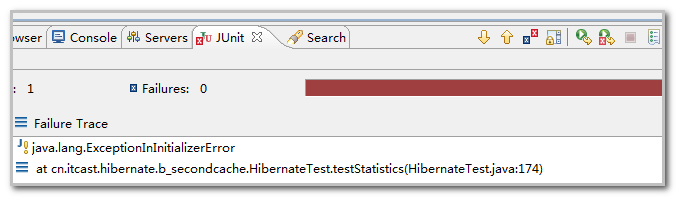
System.*out*.println("二级缓存的命中次数:"+statistics.getSecondLevelCacheHitCount());

System.*out*.println("二级缓存的丢失次数:"+statistics.getSecondLevelCacheMissCount());

session2.close();

}

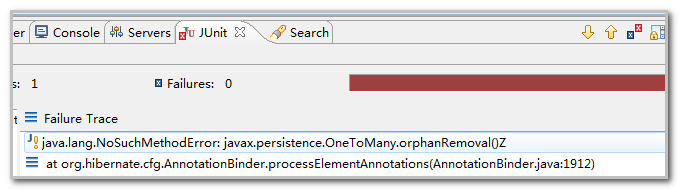
结果：



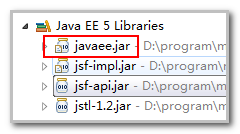
原因：开启二级缓存的监控位置错误，修改一下顺序即可



## 一对多注解建表错误



原因 ：myeclipse优先使用了[**Java EE**](http://lib.csdn.net/base/javaee) 5 Libraries中的javaee.jar



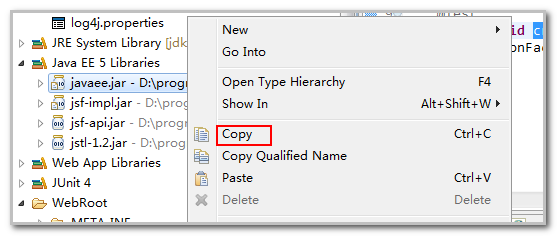
该javaee.jar 包含不完整JPA规范；

解决方案：手动删除 javaee.jar中 javax.persistence 包 ，让其使用jpa规范中的jar包

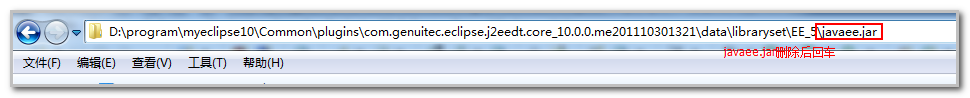


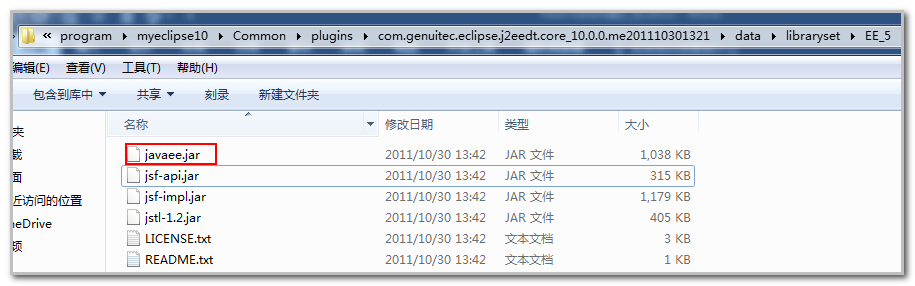
删除方法如下：

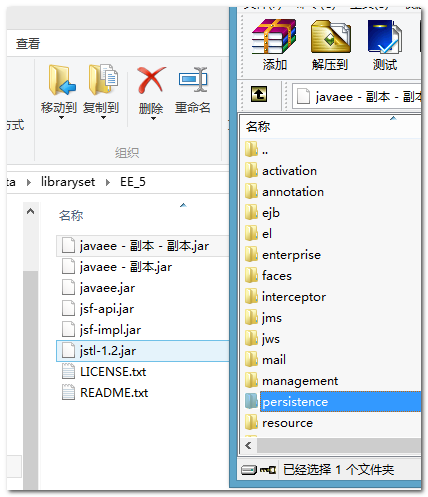
复制[**Java EE**](http://lib.csdn.net/base/javaee) 5 Libraries中的javaee.jar的路径



粘帖到地址栏中：删除最后的jar包名称

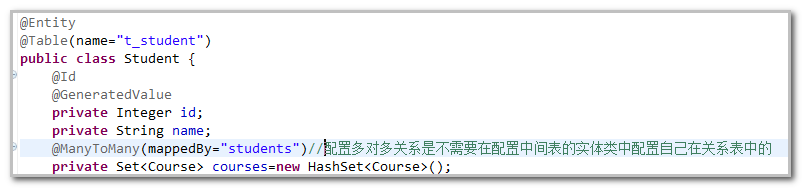




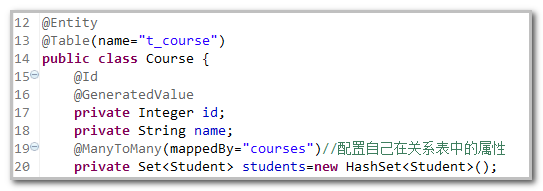


## 多对多注解简单配置的建表错误

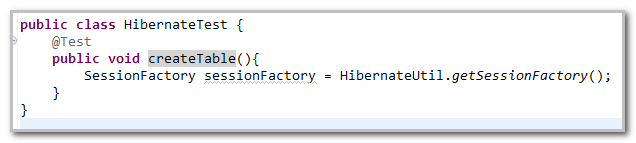
Student:



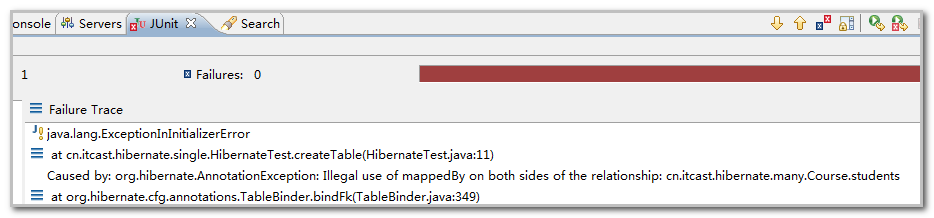
Course:



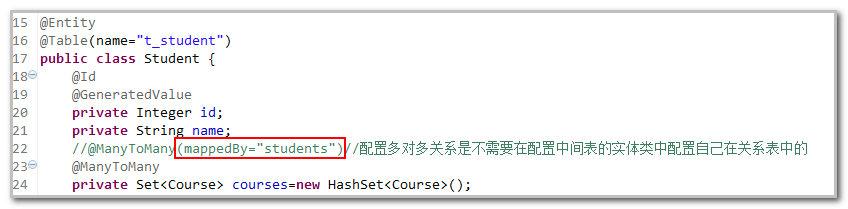
建表测试：



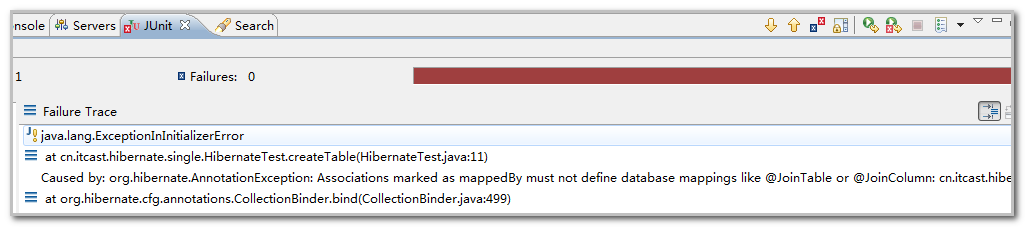
报错如下：



原因：只需要在一个实体类中配置mappedBy属性就可以了

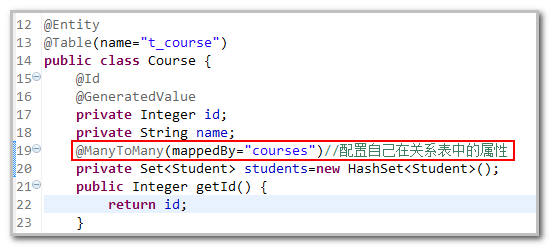


## 多对多注解复杂配置的建表错误

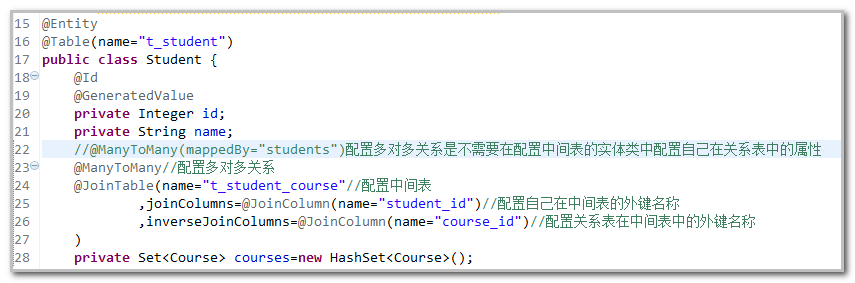


错误原因：没有在Course类中配置自己在Strudent类中的属性或者在student和course实体类中都配置了mappedBy的属性

Course:



Student:在该类中无需配置@ManyToMany(mappedBy="students")，否则还是会报如上的错误，只需要在关系表中配置一个即可



## 映射Collection集合类(不使用@OneToMany)

对象无法获得，使用HibernateTemplate. get()时会提示如下错误:

org.hibernate.LazyInitializationException: failed to lazily initialize a collection of role: org.fireprotection.domain.InfoItem.images, could not initialize proxy - no Session

@Fetch(FetchMode.Join) //添加初始化模式，等同于@OneToMany中的EARGER