**第1讲：Hibernate5** hibernate

1. 思想：ORM （Objcet Related Mapping）Hibernate做ORM

类（实例）----------------关系（表）映射（对应起来）

具体：三个方面映射： 类------表，对象---------记录 ，属性----------字段，

hibernate帮助映射内容：

1. 创建一个对象---------在相应的表中插入一条记录
2. 每个对象的属性值-----------插入记录中的一列
3. 查询对象，就可以自动查询数据库
4. 对数据库的各种操作编程人员不再使用sql语句，使用面向对象
5. 编程不再关注数据库的类型

总之：hibernate是一个数据层框架，持久层框架，包装了jdbc。

补充：

DATABASE数据库：关系型数据库、面向对象的数据库

应用程序--------JDBC数据访问技术（微软的ADO）--------------DB（关系型）

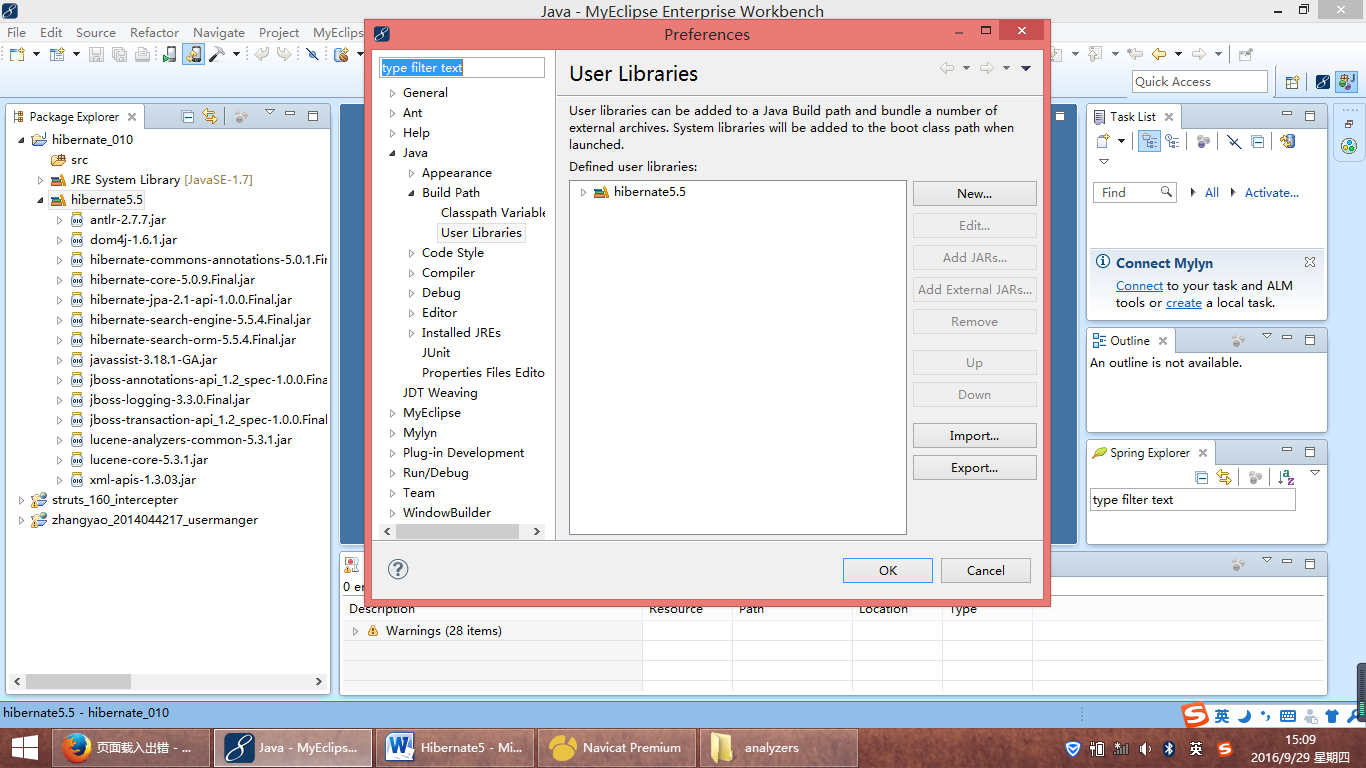
Tomcat 、 Jdoss 、 weblog 、websphere

1. Jar架包：

4开发第一个项目

1. 建立java
2. 添加

1、在Window——Preferences——java——User Libraries

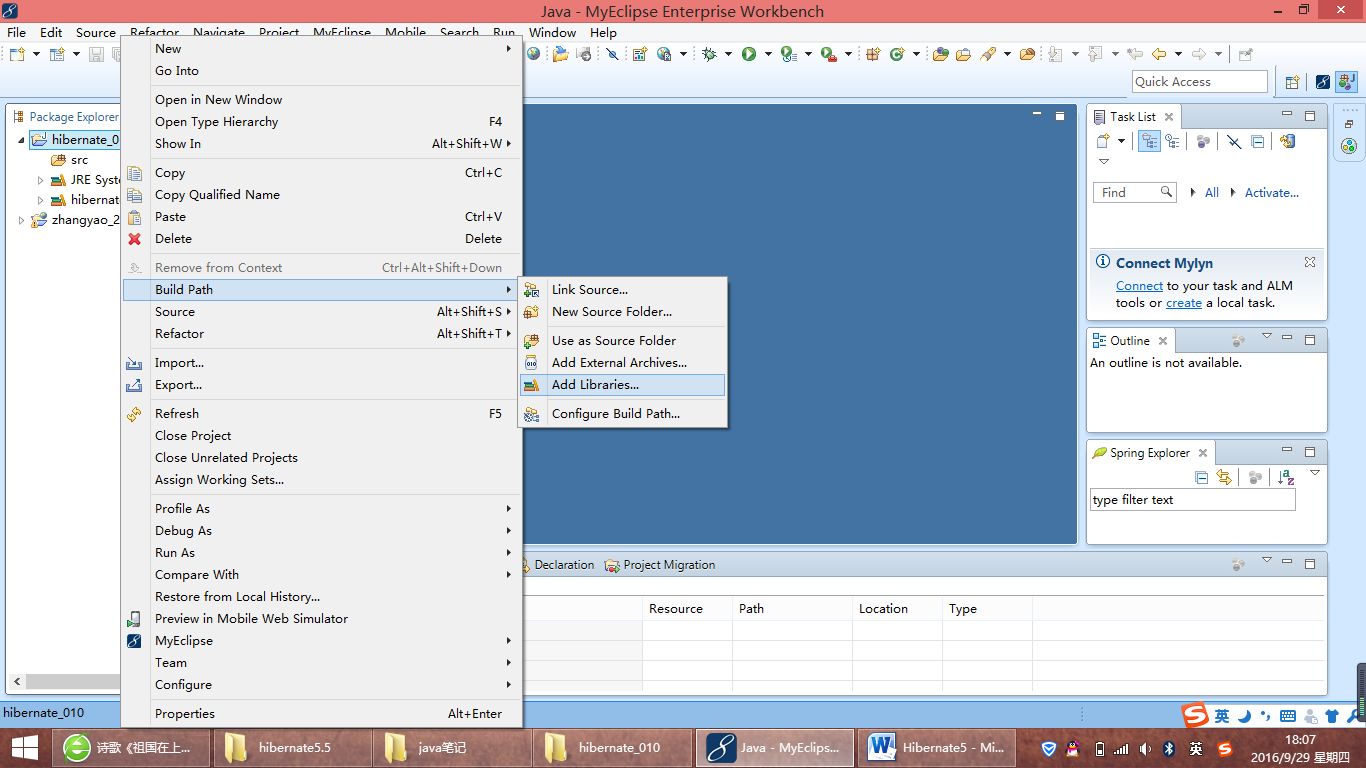


2、点击“New User Library”——添加（添加内部的），随便输入，点击”OK”

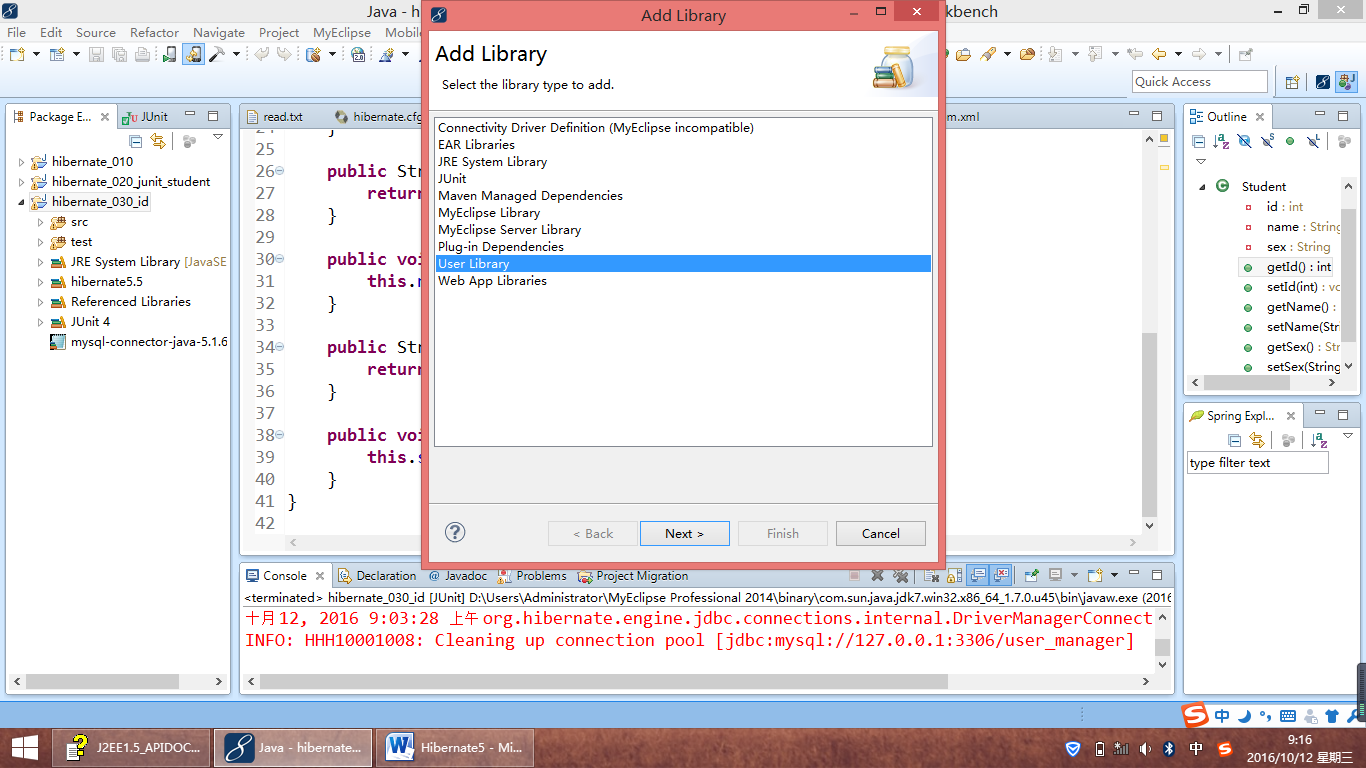
3、随便输的数字，选中点击“Add JAR Selerction”,选择项目

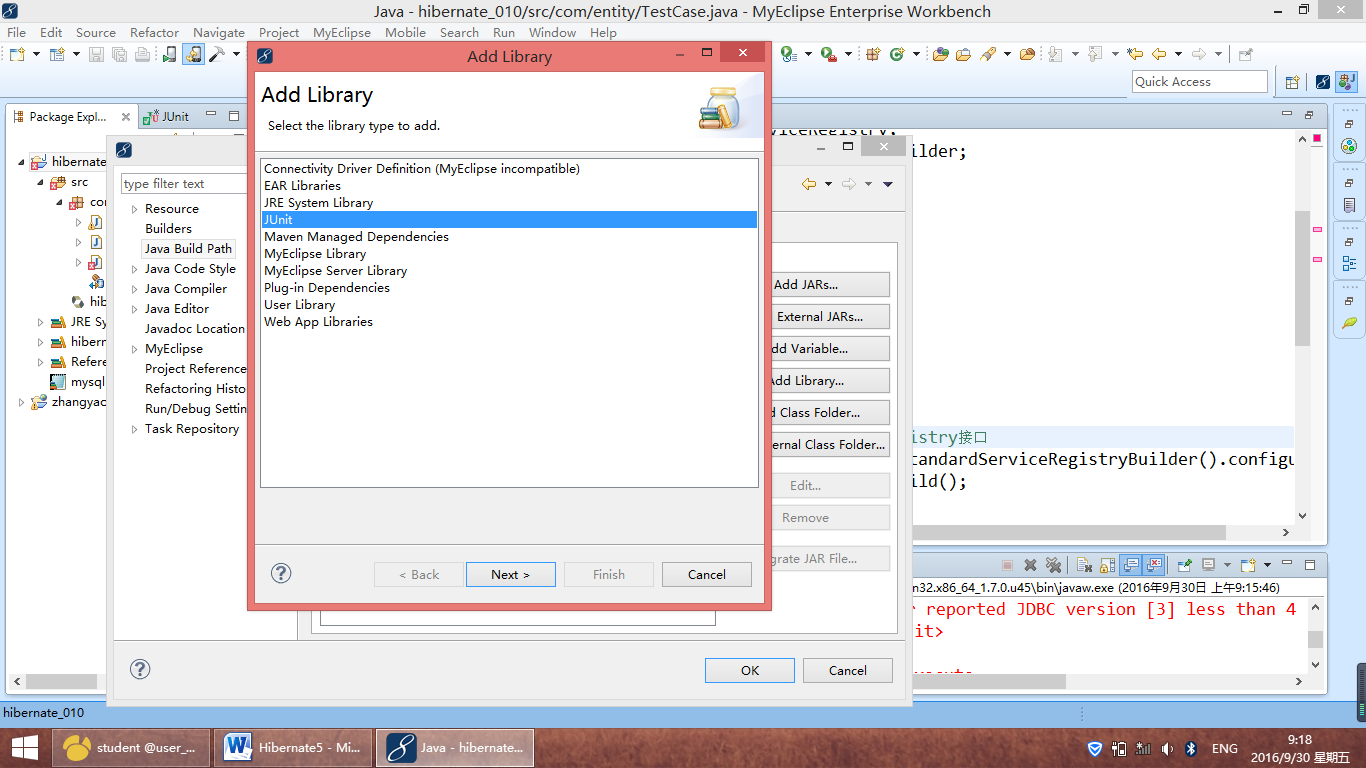
1. 在窗口选择new，在打开的窗口中输入user lib名称 ，确定。选择lib名称，点击“add jars”
2. 右击项目----选择“buit”

（第一步）



（第二步-----两个一样）





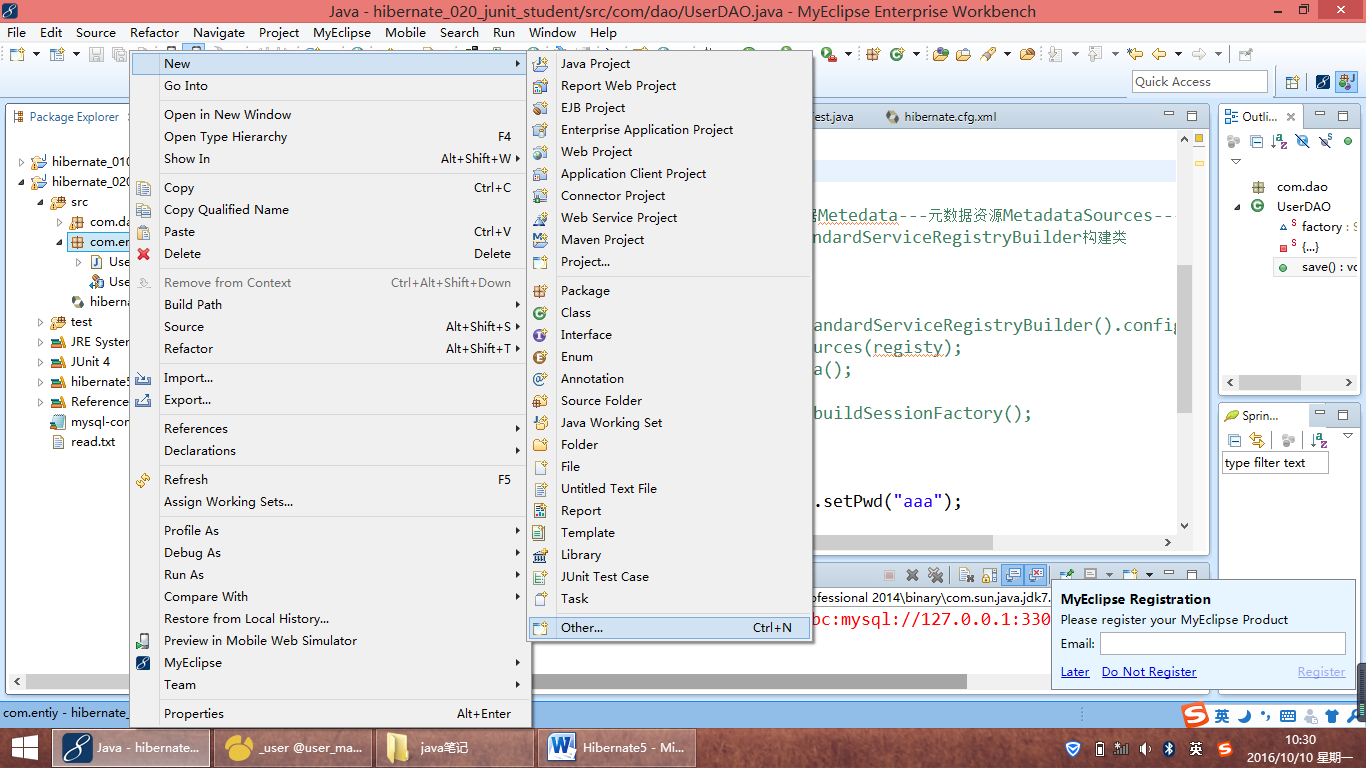
添加mysql的架包jdbc架包

4.配置（两种方式：xml、properties）

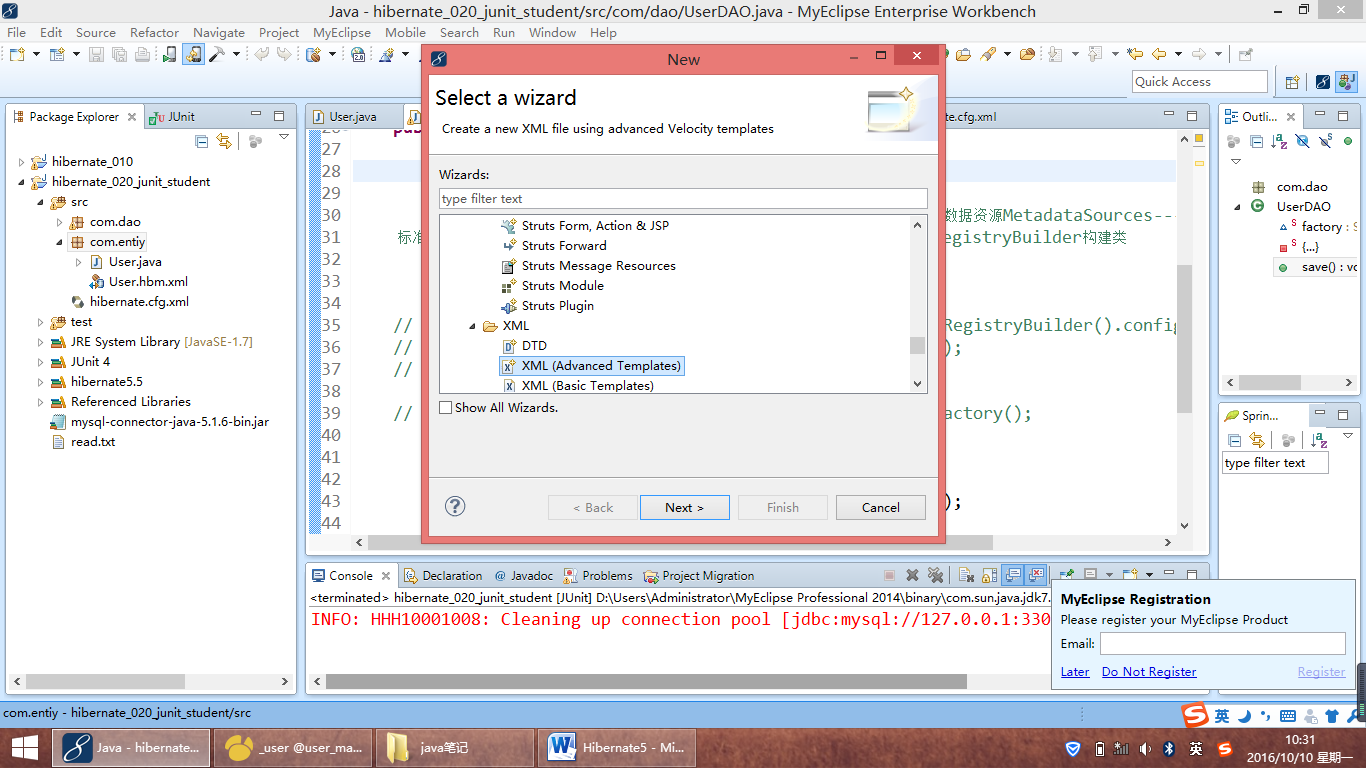
在文件src下：hibernate

创建xml

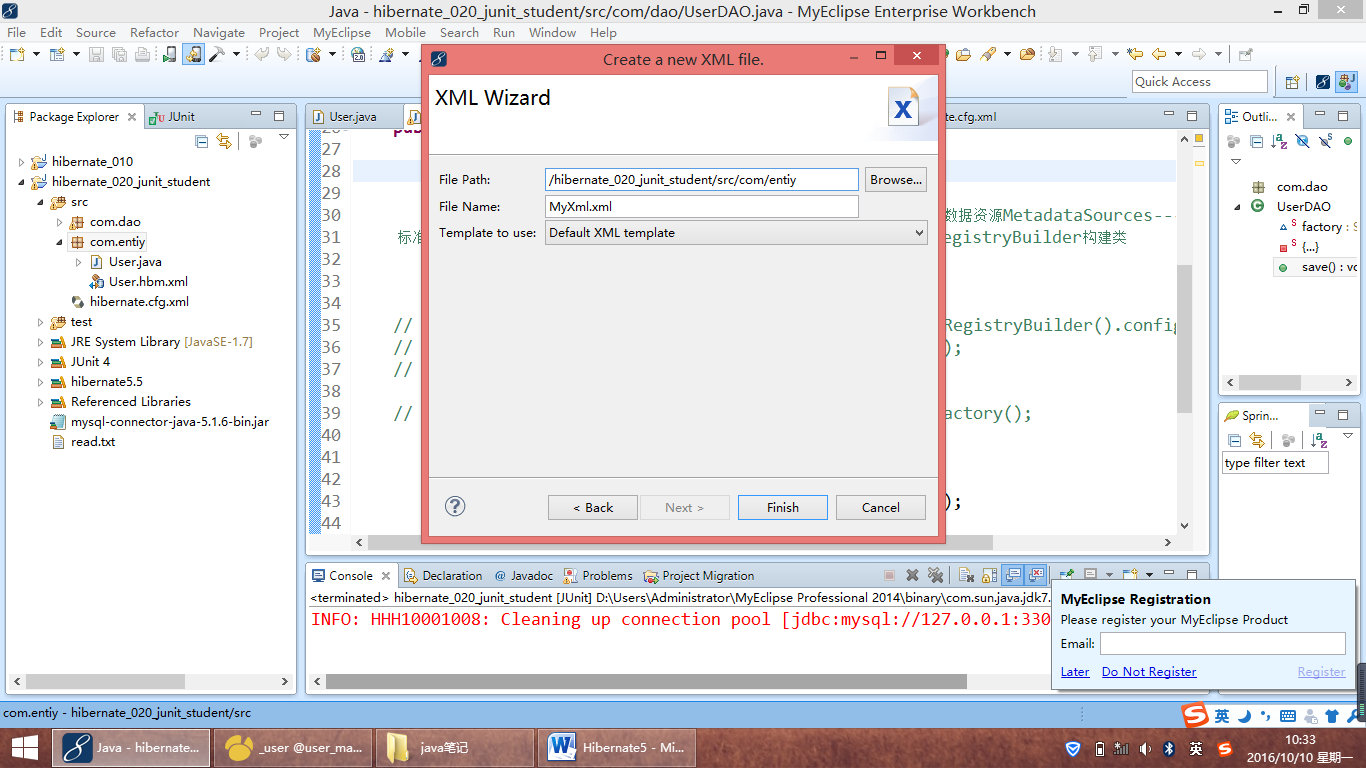
（第一步）



（第二步）



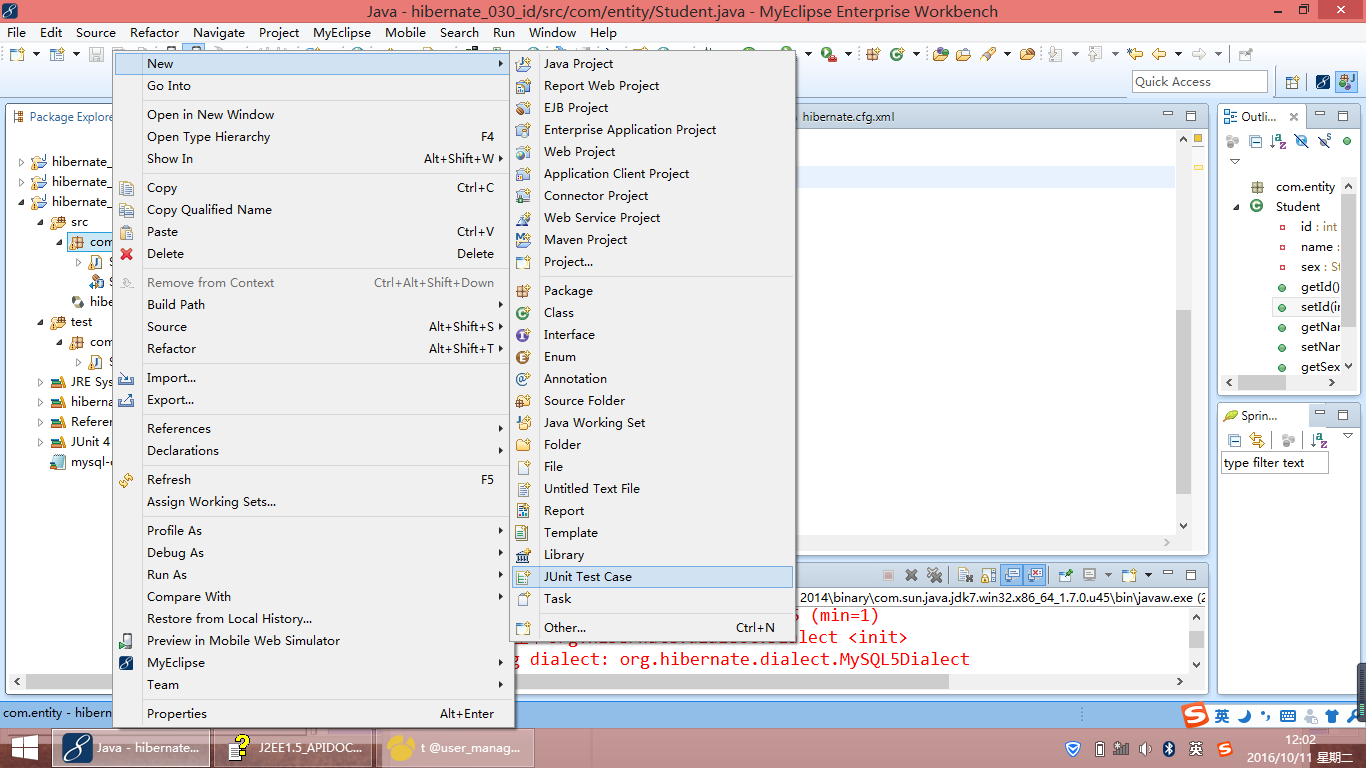
（第三步）



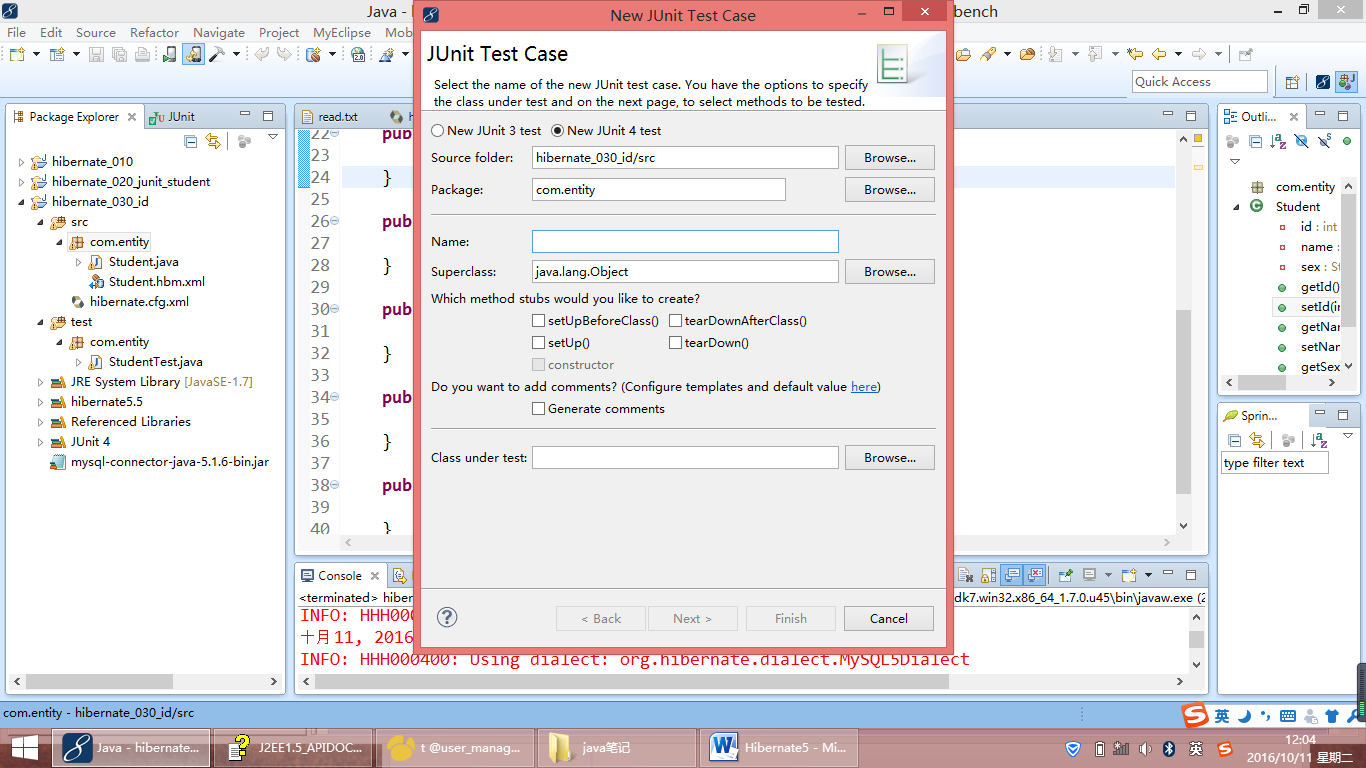
修改完，点击Finish

创建Student.hbm.mxl

（第一步）

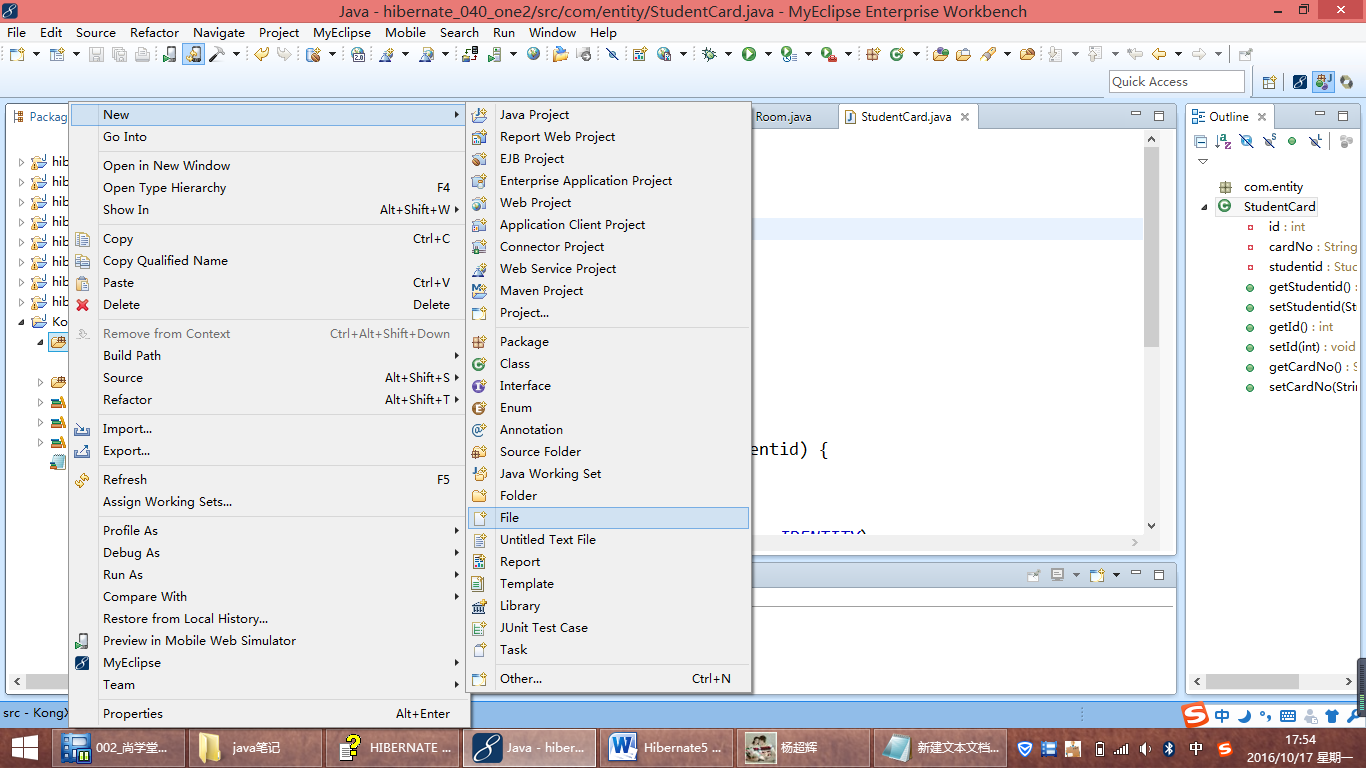


（第二步）

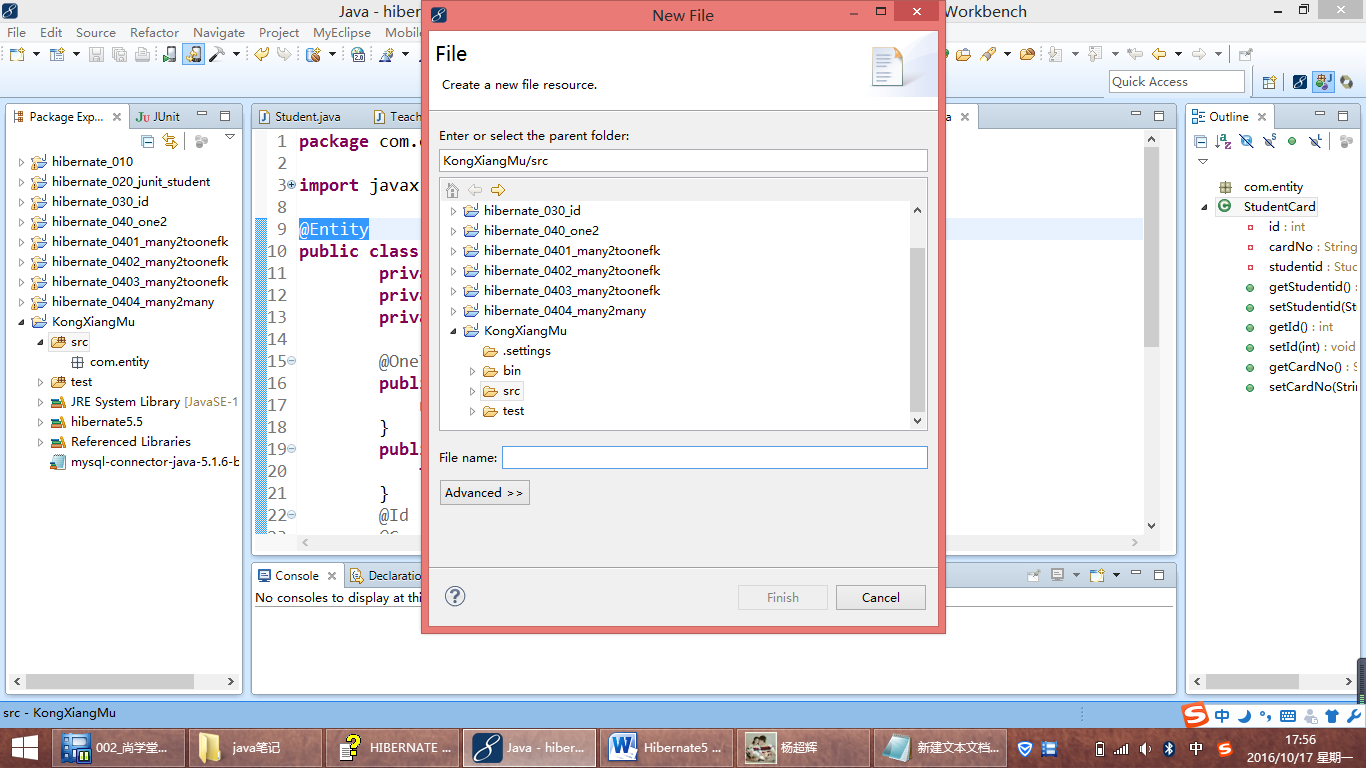


创建 hibernate.cfg.xml步骤

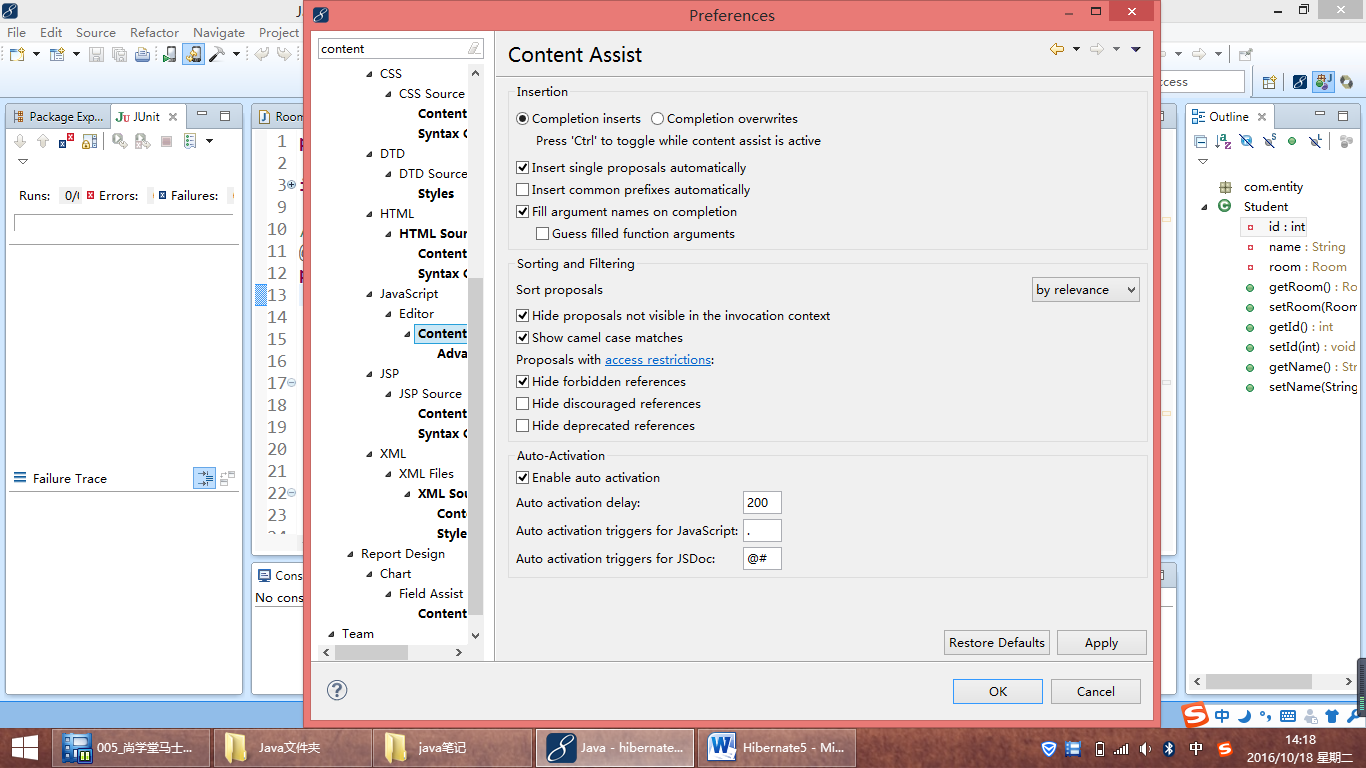
（第一步）



（第二步）



如果@，不提示,在



加上.@，自动提示

5.定义po类的映射文件

<hibernate-mapping>

<class name="events.Event" table="EVENTS">

<id name="id" column="EVENT\_ID">

<generator class="native"/>

</id>

</class>

</hibernate-mapping>

6.编写测试类

五.编写测试类，测试数据插入的方法

1、通过Junit

2、实现数据库的功能

**第2讲：Hibernate5中annatation配置**

1. 注解类
2. 注解
3. 注解主键

update create（生成）

**第3讲：Hibernate5中ID主键生成策略**

ORM----

表的操作： 建表 （字段的设置、约束）、 curd、 表之间的关系（3）、 事务

1. xml配置
2. annotation方式

查api @GeneratedValue

1. 联合主键

|  |  |  |
| --- | --- | --- |
| id | name | sex |
| 1 | A | 11 |
| 1 | B | 2 |
| 2 | B | 2 |

主键类5：必须序列化、必须复写equalse、hashcode方法

属性（主键类）---字段----联合主键（2个以上字段）

主键和主键类关系：（1）没有包含（2）包含

**第4讲：Hibernate5中---session增、删、改、查**

Transient Persistent Detached

4.1知识点

1、comfigure(“xml文件名”)

2、

设置session的上下文，有两种上下文。Jta、thread

Jta—java transaction api事务支持分布式处理、jpa(java persisten api)---需要web服务去支持处理，tomcat不支持，jboss、weblogic、等支持

3、两种方式创建

瞬态transient：内存有、缓存没有、无id,数据库没有

持久态persistent：内存有、缓存有，有id,数据有

游离态detached：内存有、缓存没有、有id,数据有

Transaction（事物）transition（瞬息的）transport（透明）

1. 增

Session.delete（Obj）

1. 删

Session.delete（Obj）

1. 查

懒加载，使用对象的时候，才发出sql

1. 改

更新一个get到对象

更新一个指定id的非get得对象

不使用update在事务中直接修改属性实现更新

使用merge合并更新

**第5讲：Hibernate5中—对象的关联**

反向—逆向工程：从代码到设计图，从数据库到数据库设计图

表的关系：一对一、一对多、多对多

5.1、对象一对一关联：单向、双向关联

单向类图：



双向类图：

步骤：

1. 类中互相包含另一个类的对象，（通过主类可以找到子类的、反之也可）
2. 两边属性注解为@On20ne, 实现双向
3. 属性注释为@JoinColumn（name=“\_指定名\_”）实现数据库中命名外键字段
4. 在子类中@On20ne(mappedBy="对方的属性名 ")实现数据库中只有一个外键字段

5.2、组件映射：一对一关系中的表格会合并为一个表。通过配置

1. 对象间包含：

5.3、多对一

1、再多的一方注解

5.5、一对一、多对一的双向关联一一

1、在类中互相包含对方的属性，（注：在一的一方属性类型的集合ser）

2、配置：各自的get方法都注解：

在一的这方：属性的get方法：@ManyToMany

在多一方：属性的get方法： @ManyToOne

3、在一的着方注解说明、外键配置参照多的一方的对象属性，避免表中产生第三个表格

@ManyToMany(mappedBy="对方的属性名")

**总结：**hibernate

ORM-----

对象---表映射@Entity

属性---字段映射：属性---主键映射 @Id @Transent(不映射)、

属性如何映射------主键自动生成：生成策略@GeneratedValue()

对象关联-----表的关系映射

@OneToOne------@ManyToMan--@ManyToOne---@ManyToMany

外键名控制：@JoinColumn(name=“­\*\*\*\*\*”)

双向重叠参照(解开):@OneToOne(mappedBy="对方的属性名")

@ManyToMany(mappedBy="对方的属性名")

对单表的增删该查

Session.save

Session.deleter

Session.updare, Session.merge,持久化状态的对象只要修改对象属性，自动更新到数据库

查：load----懒加载 ,在使用对象前，不能关闭Session

get---直接发送select，关闭Session也可以使用对象

对象的三个状态：

Session：其实是一个缓冲区域，与数据库的连接信息、包含了映射的对象（map方式保存）

事务处理Transaction：对数据库的操作放在事务处理里，开启、提交、回滚

对关系表的增删该查

MyBatis

**第6讲：Hibernate5中-----关联对象的curd**

在数据库中，有关联的表之间做curd时，通过外键约束限制：

1. 在删除（主表记录）时：字表的动作有四种：

层叠 置mull 无动作 限制不可许

1. 更新时（增、删、改）时：层叠 置mull 无动作 限制不可许
2. 查：通过字表找到主表、主表记录无法找到字表

**步骤：**

1. 保证 主键字段值 自动生成
2. 创建两个类：保证room-student是一对多-多对一的双向关联
3. 保证Student表有外键，主表room无外键（@OneToMany（mappedBy="room"） ）
4. 修改外键字段名为roomId
5. CURD:

1)保存room(父表"一"方)，同时保存关联的子对象

@OneToMany(mappedBy="room",cascade={CascadeType.PERSIST})

或@OneToMany(mappedBy="room",cascade={CascadeType.ALL})

2)保存room(子表"多"方)，同时保存关联的子对象

@ManyToOne(cascade={CascadeType.ALL|PERSIST})

或者session.save(r);

3) 查找room（父表对象，“一”方）：同时找到room中的有关对象

查找student（子表对象，“多”方）：同时找到student中的有关父对象

只要建立关联（ManyToOne、OneToMany）就都可以找到

======与casecade的设置无关系无关 ============

4) 删除room（父表对象，“一”方）：同时删除到room中的有关对象

@OneToMany(mappedBy="room",cascade={CascadeType.ALL})

@OneToMany(mappedBy="room",cascade={CascadeType.REMOVE})

否则不可以删除：“一”方的记录，当子表中有记录引用父表记录时候。

5) 删除student（子表，“多”方）:同时删除student中的有关room对象

@ManyToOne(cascade={CascadeType.REMOVE|ALL})

否则，只能删除子表记录，父表不受影响

6）修改：

**第7讲：Hibernate5中----集合映射、树映射、继承射**

**EJB2、3（企业级、ejbql、）**

Sun:java—javaScript---C#,php

javaEE—(web,ejb3,RMl,Corber,分布式编程)

一个程序分布在多个机器运行，大数据处理（hadoop,python语言）

Java Bean----Student、room

**第14讲：HQL查询语言（hibernate、query、language）**

Sql---对关系查询：select\*，name from room as r

Hql---对象的查询：select name from Room as r

ORM---

14.1 标准对象化查询

14.2 hql查询语言----hibernate查询