

## 【济南中心】JavaEE就业班同步笔记第三阶段： Hibernate-part02

小鲁哥哥 • 2017-6-11 11:46:50

# 【济南中心】JavaEE就业班同步笔记第三阶段： Hibernate-part02

## 1.1 持久化类的编写规则

### 1.1.1 什么是持久化类

#### 1.1.1.1 什么是持久化类

就是一个JavaBean，这个JavaBean与表建立了映射关系。这个类就称为是持久化类。  
简单理解为 持久化类=JavaBean+映射文件。

#### 1.1.1.2 持久化类的编写规则

需要提供无参数的构造方法。

类的属性需要私有，对私有的属性提供public的get和set方法。

提供一个唯一标识（OID）的属性与表的主键对应。

Java里面使用地址区分是否是同一个对象。数据库中是通过主键区分是否是同一个记录。Hibernate通过唯一标识区分是否在内存中是同一个对象。

类不要使用final修饰。

类中的属性尽量使用包装类类型。

## 1.2 主键生成策略

### 1.2.1 主键的类型

#### 1.2.1.1 自然主键

自然主键指的是类的自身的属性作为主键。

#### 1.2.1.2 代理主键

代理主键指的是不是类本身的属性作为主键。

#### 1.2.2 Hibernate的主键生成策略

## 主键生成策略

名称	描述
increment	用于 long、short、或 int 类型，由 Hibernate 自动以递增的方式生成唯一标识符，每次增量为 1。只有当没有其它进程向同一张表中插入数据时才可以使用，不能在集群环境下使用。适用于代理主键。
identity	采用底层数据库本身提供的主键生成标识符，条件是数据库支持自动增长数据类型。在 DB2、MySQL、MS SQL Server、Sybase 和 HypersonicSQL 数据库中可以使用该生成器，该生成器要求在数据库中把主键定义成为自增长类型。适用于代理主键。
sequence	Hibernate 根据底层数据库序列生成标识符。条件是数据库支持序列。适用于代理主键。
uuid	Hibernate 采用 128 位的 UUID 算法来生成标识符。该算法能够在网络环境中生成唯一的字符串标识符，其 UUID 被编码为一个长度为 32 位的十六进制字符串。这种策略并不流行，因为字符串类型的主键比整数类型的主键占用更多的数据库空间。适用于代理主键。
assigned	由 java 程序负责生成标识符，如果不指定 id 元素的 generator 属性，则默认使用该主键生成策略。适用于自然主键。

回复帖子...

生成标识符的能力来选择 identity、  
回帖

表示符的能力来选择 identity、  
适合跨数据库平台开发。适用于

代理主键。

identity适用于MySQL有自动增长的数据库。sequence适用于有序列数据库Oracle。native叫做本地策略，根据底层的数据库不同自动选择使用identity还是sequence。

### 1.2.3 持久化类的三种状态

#### 1.2.3.1 瞬时态

瞬时态：持久化对象没有持久化标识OID，对象没有被Session管理。

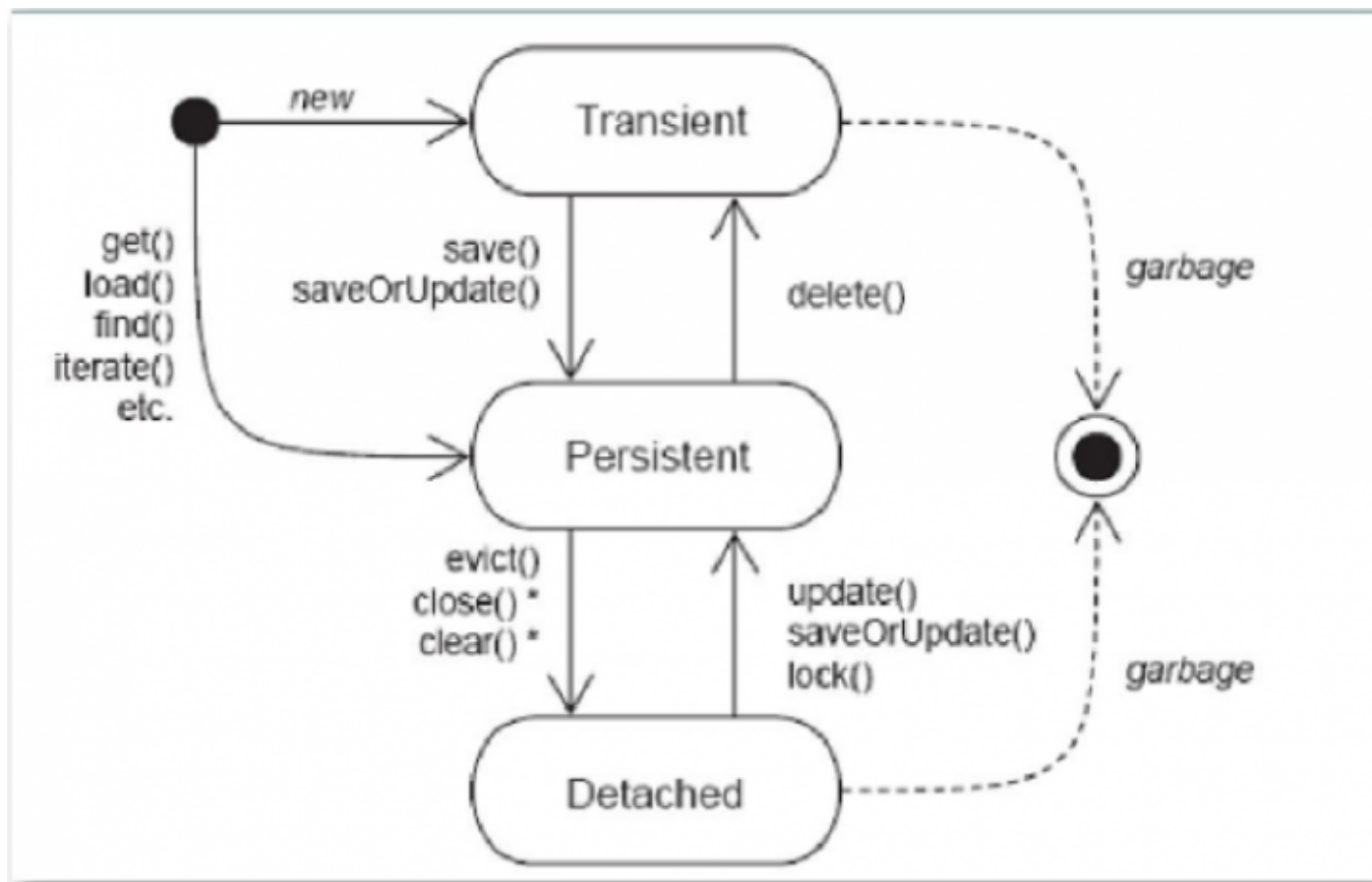
#### 1.2.3.2 持久态 (\*\*\*\*)

持久态：持久化对象有持久化标识OID，对象被Session管理。

#### 1.2.3.3 脱管态

脱管态：持久化对象有持久化标识OID，对象没有被Session管理。

#### 1.2.3.4 状态转换(了解)



### 【瞬时态】

获得瞬时态：

```
* Customer c = new Customer();
```

瞬时态对象转持久态：

```
* session.save(c); session.saveOrUpdate(c);
```

瞬时态对象转脱管态：

```
* c.setCust_id(1l);
```

### 【持久态】

获得持久态：

```
* session.get(),session.load()...;
```

持久态对象转脱管态：

```
* session.close(),session.clear(),session.evict(Object obj);
```

持久态对象转瞬时态：

```
* session.delete();
```

## 【脱管态】

获得脱管态：

```
[mw_shl_code=java,true]Customer c = new Customer();  
c.setCust_id(11);[/mw_shl_code]
```

脱管态对象转持久态

```
[mw_shl_code=java,true] session.update();  
session.saveOrUpdate();[/mw_shl_code]
```

脱管态对象转瞬时态

```
[mw_shl_code=java,true] c.setCust_id(null);[/mw_shl_code]
```

### 1.2.4 Hibernate的一级缓存

#### 1.2.4.1 什么是一级缓存

一级缓存称为是Session级别的缓存，一级缓存的生命周期与Session是一致的。一级缓存其实由Session中的一组集合构成的。

#### 1.2.4.2 特殊区域

### 1.2.5 Hibernate的事务管理

#### 1.2.5.1 什么是事务

事务是逻辑上的一组操作，组成这组操作的各个单元要么全都成功，要么全都失败！

#### 1.2.5.2 事务的特性

原子性：不可分割。

一致性：事务执行前后，数据的完整性保持一致。

隔离性：一个事务执行的过程中不应该受到其他事务的干扰。

持久性：事务一旦结束，数据持久到数据库。

#### 1.2.5.3 如果不考虑隔离性，引发哪些问题？

脏读：一个事务读到另一个事务未提交的数据。

不可重复读：一个事务读到另一个事务已经提交的update数据，导致多次查询结果不一致。

虚读：一个事务读到另一个事务已经提交的insert数据，导致多次查询结果不一致。

#### 1.2.5.4 Hibernate中设置事务隔离级别

```
[mw_shl_code=xml,true]<!-- 设置事务的隔离级别 -->
```

```
<property name="hibernate.connection.isolation">4</property>[/mw_shl_code]
```

#### 1.2.5.5 与线程绑定的Session的使用

```
[mw_shl_code=xml,true]<!-- 配置与当前线程绑定的Session -->
```

```
<property name="hibernate.current_session_context_class">thread</property>[/mw_shl_co]
```

de]

## 1.2.5 Hibernate的其他的API

### 1.2.5.1 Query:支持HQL查询

获得Query接口可以通过`session.createQuery(String hql)`;获得。

HQL:Hibernate Query Language。Hibernate查询语言。语法与SQL是类似的。HQL中查询的是对象。

### 1.2.5.2 Criteria:支持QBC查询

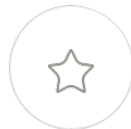
获得Criteria接口可以通过`session.createCriteria()`;获得。

QBC:Query By Criteria。条件查询。一种更加面向对象的方式。

### 1.2.5.3 SQLQuery:支持SQL查询

获得SQLQuery接口可以通过`session.createSQLQuery()`;获得。

SQLQuery:通过SQL语句进行查询。



回帖

— 5条回帖 —



dhj

感谢分享!!!!

沙发 • 2017-6-11 16:44:41

回帖



dhj

感谢无私分享!!!!

藤椅 • 2017-6-11 17:01:01

回帖



南宫飞儿

感谢楼主分享

板凳 • 2017-6-12 19:55:58

回帖



烽火狼烟

谢谢楼主分享

报纸 • 2017-6-13 22:37:16

回帖



来自宇宙超级黑马专属苹果客户端



battlexing

没有一级缓存的快照机制，差评

地板 • 2017-11-1 19:21:14

回帖