Mybatic的缓存模块使用了装饰者模式的变体

## Mybatis + Hibernate

Hibernate的核心类和接口： Configuration ，SessionFactory，Session，Transaction，Query

Hibernate pojo的3种状态

瞬态 持久化 托管（离线）

一个PO有三种状态：

1、未被持久化的vO==pojo

此时就是一个内存对象VO，由JVM管理生命周期

2、已被持久化的PO，并且在Session生命周期内

此时映射数据库数据，由数据库管理生命周期

3、曾被持久化过，但现在和Session已经detached了，以VO的身份在运行这种和Session已经detached的PO还能够进入另一个Session，继续进行PO状态管理，此时它就成为PO的第二种状态了。这种PO实际上是跨了Session进行了状态维护的。在传统的JDO1.x中，PO只有前面两种状态，一个PO一旦脱离PM，就丧失了状态了，不再和数据库数据关联，成为一个纯粹的内存VO，它即使进入一个新的PM，也不能恢复它的状态了

Hibernate的缓存分为：

一级缓存(内置缓存)->session级别缓存,是Hibernate 的内置缓存 一级缓存在Hibernate中对应的即为session范围的缓存，也就是当session关闭时缓存即被清除，一级缓存在Hibernate中是不可配置的部分

二级缓存(应用级缓存)：

二级缓存也称进程级的缓存或SessionFactory级的缓存，二级缓存可以被所有的session共享

Hibernate 延迟加载(lazy)

1、lazy的概念，指在需要数据的时候才发出sql

2、lazy策略只是在session打开期间才是有效的

hibernate 抓取策略 ( 查询抓取 子查询抓取 连接抓取 批量抓取)

连接抓取 fetch="join" 通过select语句使用外连接来加载其关联实体或集合 此时lazy会失效

查询抓取 fetch="select" 另外发送一条select语句抓取当前对象关联实体或集合

子查询抓取 fetch="subselect" 另外发送一条select语句抓取在前面查询到的所有实对象的关联集合 通过子查询in完成

批量抓取 batch-size设置

Hibernate使用乐观锁：只需在在class标签上 optimistic-lock="version"，等配置。

悲观锁的实现，通常依赖于数据库机制，在整个过程中将数据锁定，其它任何用户都不能读取或修改 利用LockMode.UPGRADE。p=(Person)session.load(Person.class, 6,LockMode.UPGRADE);