

领域涉及哪些对象？

工厂对象 -> 用来创建系统中不存在的实体对象。

资源库 -> 用来从系统生查询已经存在的实体对象。

实体对象 -> 通俗点讲就是业务对象，与数据库实体最大的区别是领域实体是有行为的。
一个领域实体可能涉及多个数据库实体，也可能一个都不涉及。

服务对象 -> 最简单的理解就是“交易系统”对外曝露的接口，统一封装到服务对象中。

值对象 -> 为了完成一些复杂业务，我们可能会创建一些中间对象来保存方法的参数、返回值等等。
这样的对象不会被持久化，业务过程结束就销毁掉了，这类对象就是值对象。
值对象只在Domain内部使用，不应该曝露给上层使用。

每个领域实体对象，都有save和remove方法。

save方法用于创建和更新对象。

remove方法用于从系统中删除该对象。



工厂来用创建系统中不存在的实体，通过工厂得到实体后，可以通过实体的save方法将其持久化。



资源库用于查询系统中已经存在的实体，通过资源库查询实体后，可以使用实体的save方法更新实体的属性；或者使用remove方法将该实体对象从系统中删除。



对象层次结构



领域实体的行为？

假设“商品”作为一个领域实体。那么由此能想到的动作有哪些呢？商品下架，增加库存量，设置标签，增加图片等都可以作为商品的行为。

仔细看看上面的四个行为，其中“下架”和“增加库存量”这两个行为，其内部实现都是只涉及商品这个实体本身，完全可以通过save方法就可以实现了，所以像类似的功能，我们约定不将其作为一个行为来设计，而是将其作为服务对象的一个接口来设计。

再看看“设置标签”和“增加图片”这两个行为，其内部实现涉及到多个领域实体：商品，标签和商品图片，所以我们约定涉及多个领域实体共同参与的行为，将其作为其中一个参与实体的行为。

再看看像“订单”这样的实体，有“支付”、“退款”这样的行为，虽然看起来内部实现只涉及“订单”这个实体，但是每一次行为发生之后，其状态也发生了变化，订单日志记录也会增加相应的记录，像类似这样的行为，我们也约定将其作为实体的行为。



- 会员
 - 基本信息
 - 账号
 - 第三方账号
 - 密码
 - 联系信息
- 等级
- 积分
 - 积分明细

- 商品
 - 紫砂壶
 - 精品
 - 茶
 - 商品图片
 - 商品属性
 - 物品清单
- 资询
- 评价
- 追评
- 购物车
- 标签

- 购物订单
 - 发票
 - 物流信息
 - 订单商品
 - 订单事件
 - 配送信息
 - 订单支付信息

- 退换货订单
 - 退换货订单退款信息
 - 退换货订单物流信息
 - 退换货订单物品





