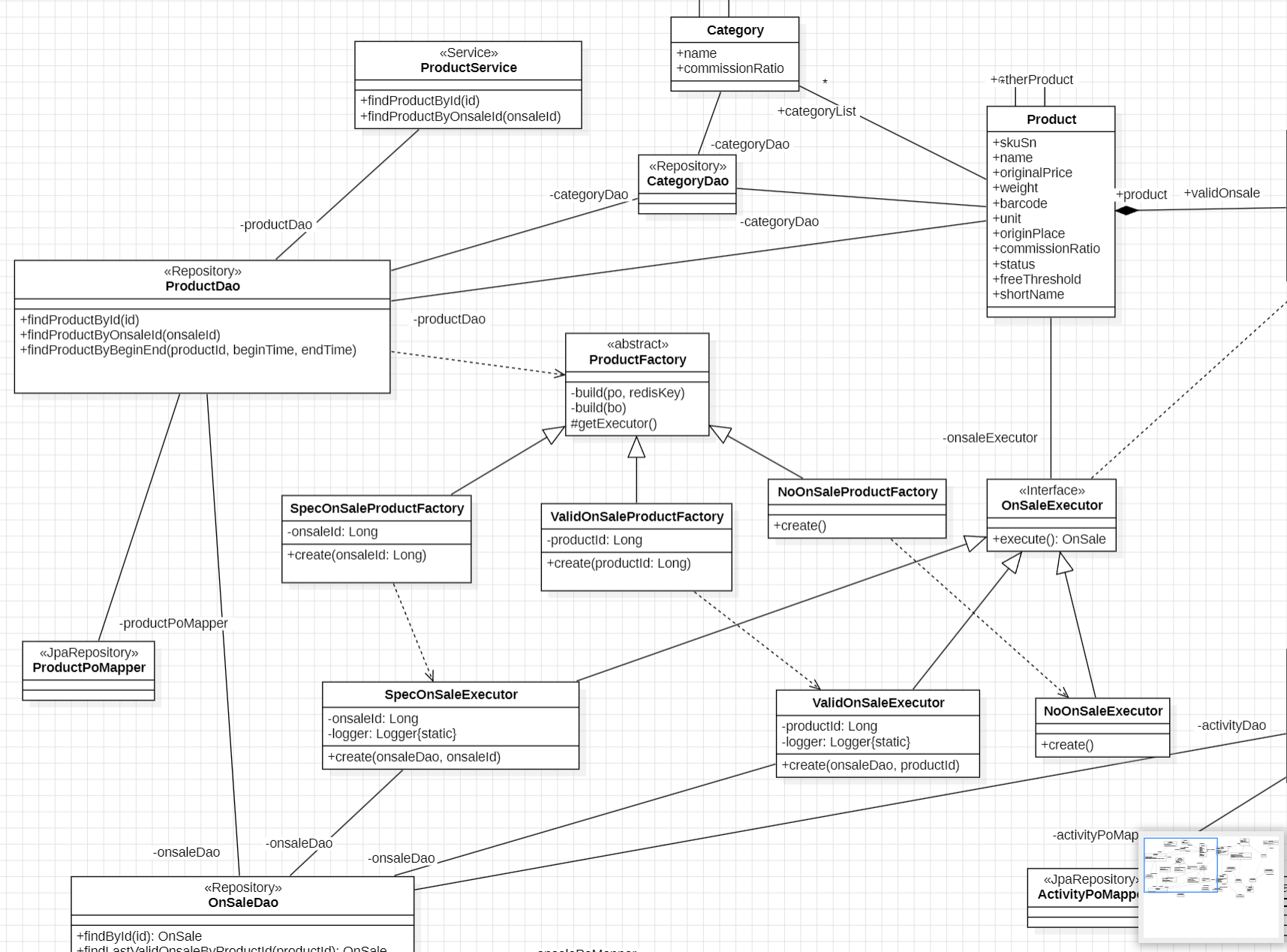
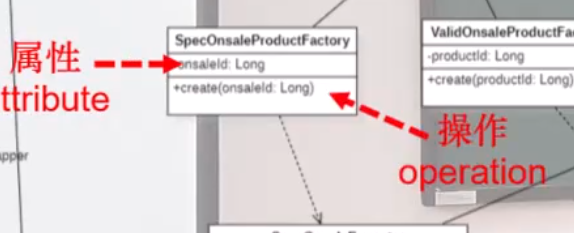
Uml类图基本符号：



类的表示：



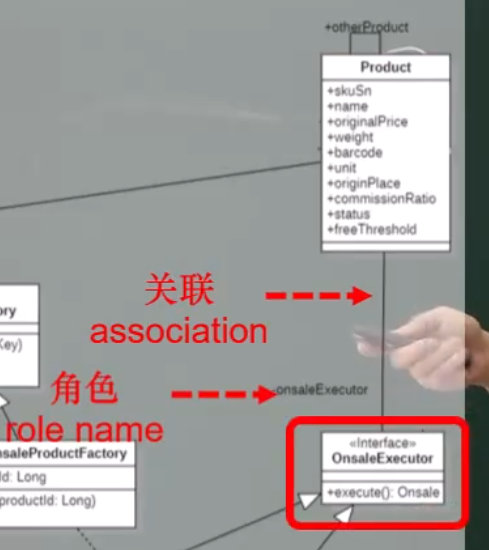


接口的表示：

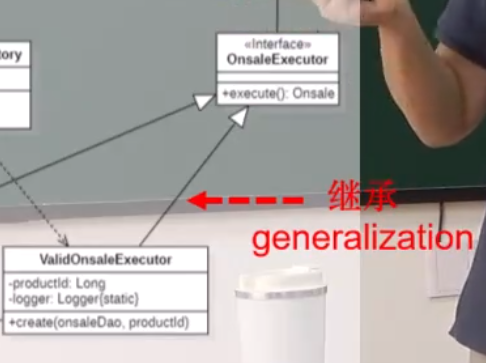


类图中的关系主要可以分为三种：**泛化（继承）**（Generalization）、**依赖**（Dependency）、**关联**（Association）

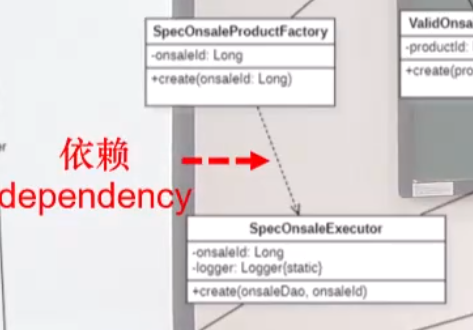
角色加实线表示关联



实线箭头表示继承

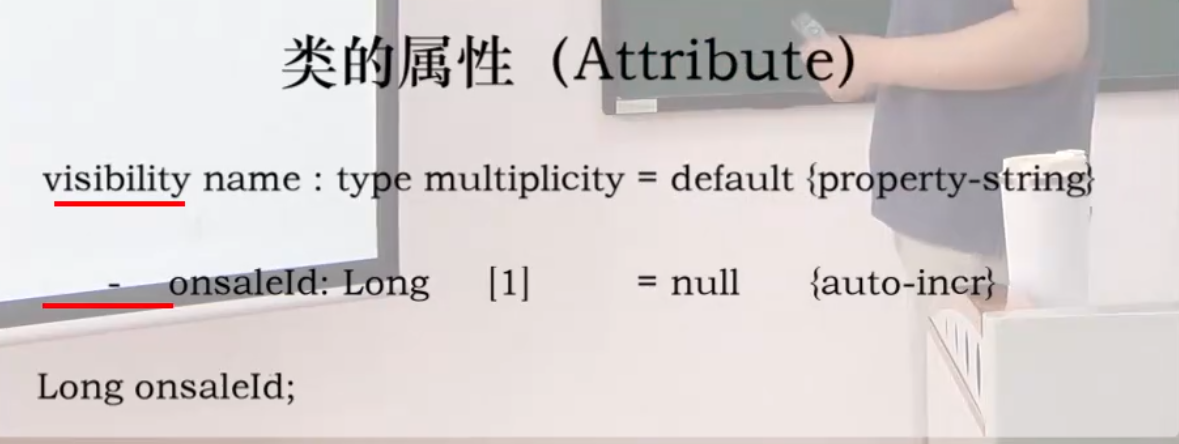


虚线箭头表示依赖



---------------------------------------------------------------------------

类的属性：



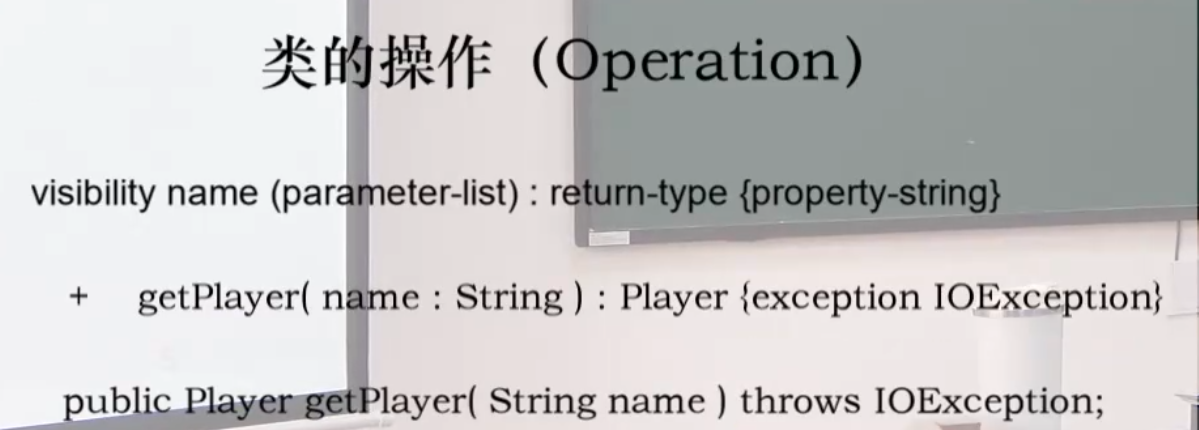
Visibility有三种：+-#，分别表示public，private和protected

Name:type：名称和类型

Multiplicity：可以表示集会里恰好有多少个对象

花括号{ }内可以写任意东西

类的操作：



Visibility：同上

Name：类名

Parameter-list：参数

Return-type：返回类型

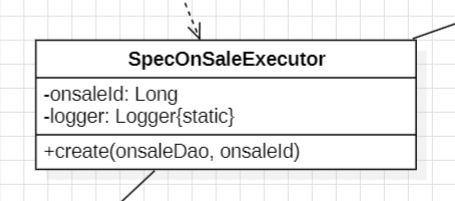
Property-string：属性和名称

类图中不能使用create作为函数名，一般默认是构造函数

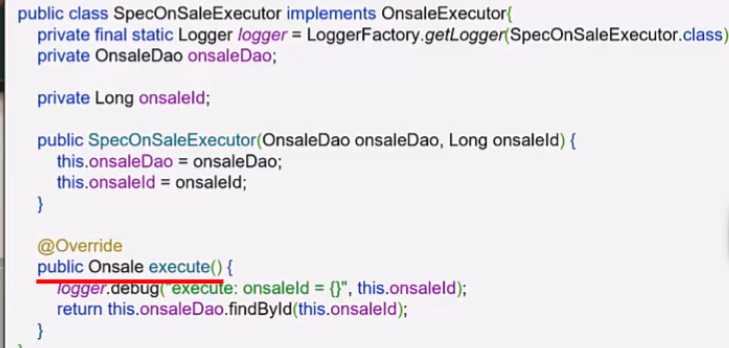
在类里面的get和set方法在图中表示为public即使它是private的，因为它可以被获取和修改，如果只有get那就在花括号中加一句read only

---------------------------------------------------------------------------

类图——关系

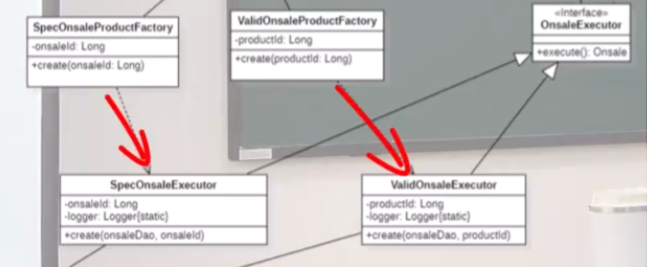


上面那个类图对应的代码如下

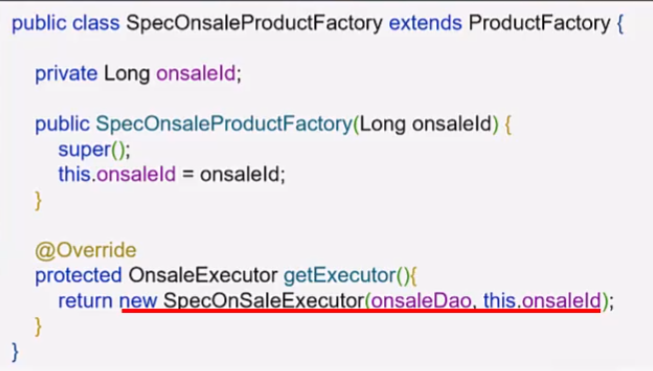


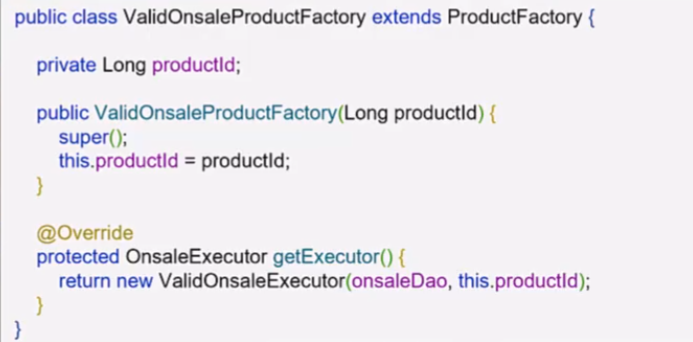
类图表示出来代码的静态结构

依赖关系：



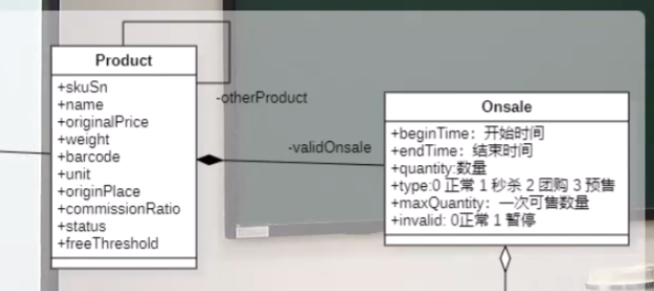
这两个依赖关系在代码中的体现如下





这两个类中都实例化了他们的基类类型返回回来，所以有一跳依赖关系的线

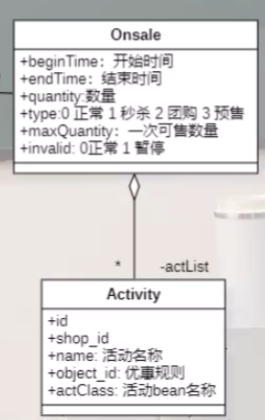
组合和聚合关系：



一个商品在不同时间会卖不同的价格，所以这是一个一对多的关系

这个符号表示组合关系。

组合关系的特征：1.是整体和局部的关系 2.局部只能属于一个整体 3.整体负责创建和销毁局部 4. 部分对象随整体对象的创建而创建，随其销毁而销毁。 部分对象通常由整体对象直接构造和管理，外部无法直接访问。

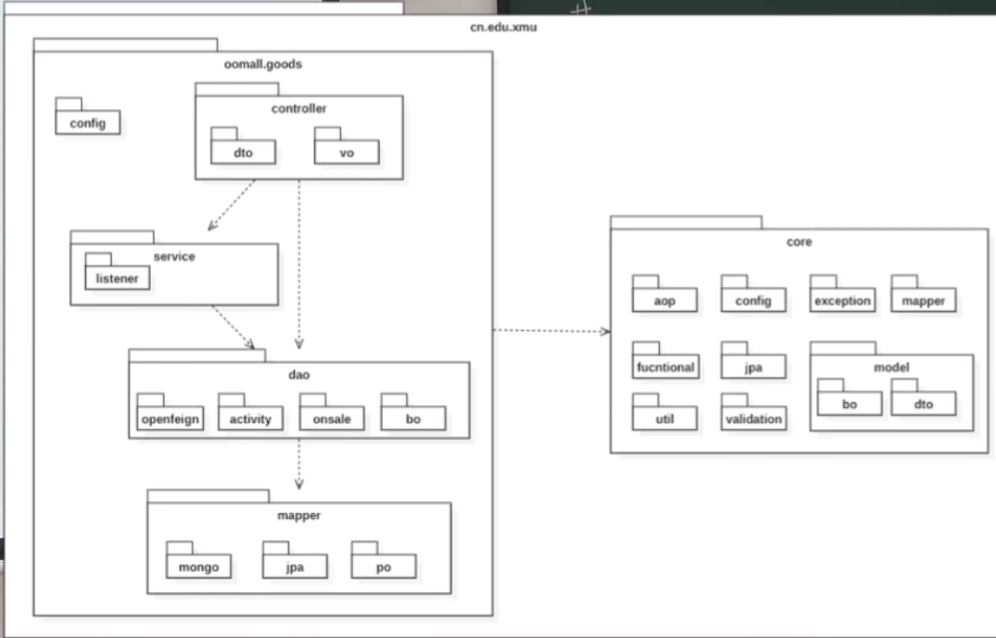


一个商品可以做不同的活动，一个活动也可以包括多个销售，所以这是一个多对多的关系。

这个符号表示聚和

包和包之间关系的描述

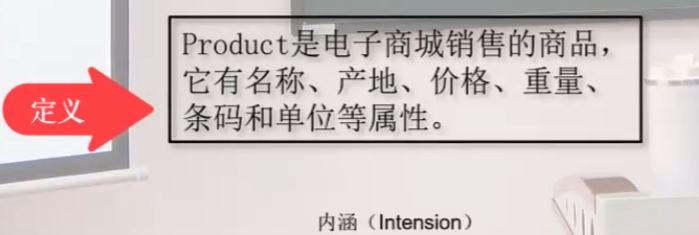
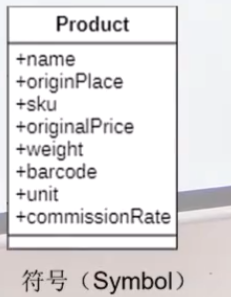
如下图：



---------------------------------------------------------------------------

领域建模：领域建模是对领域内概念类或现实世界中对象的 可视化 表示，也称为概念模型

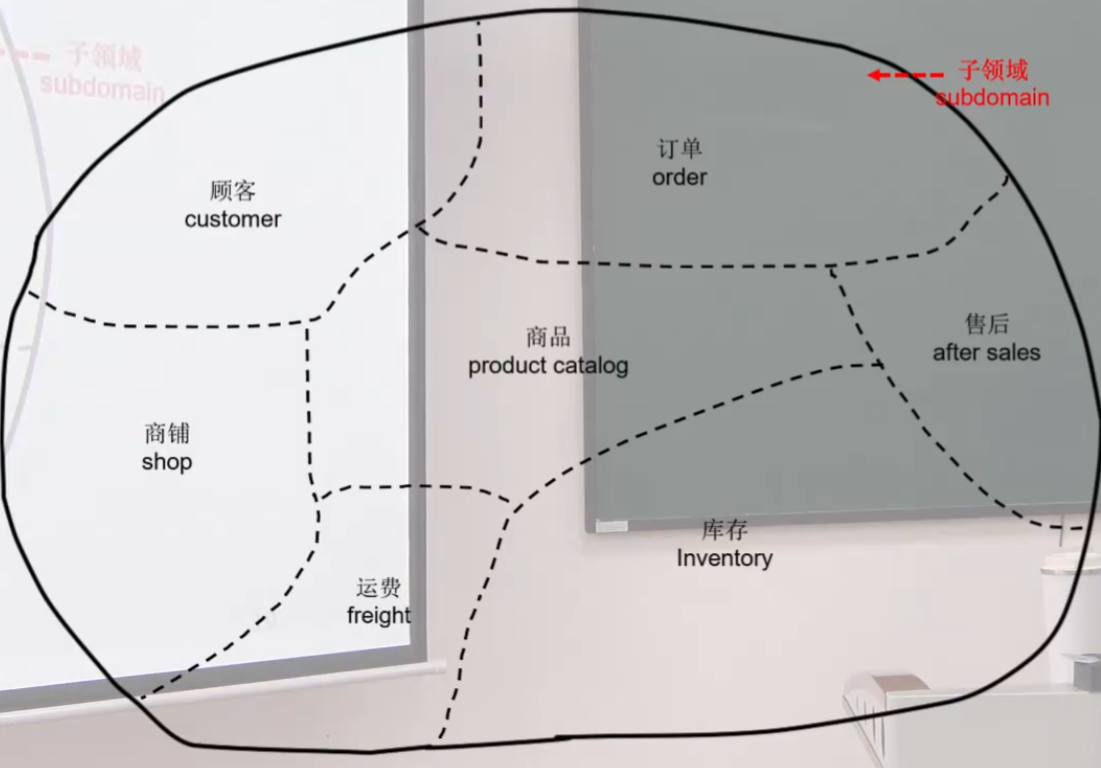
什么是概念？下图是一个概念的例子





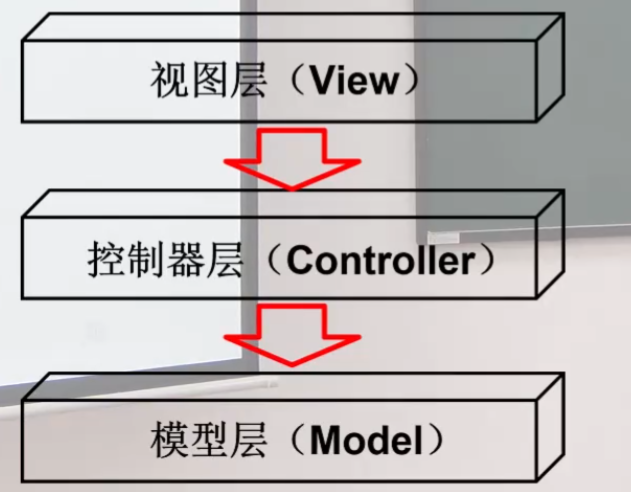
概念的主要的作用就是用来描述我们的需求，发现我们的需求的背后的逻辑关系。

在领域建模的过程中通常采用分而治之的方法



将一个大领域分成若干个子领域来进行建模

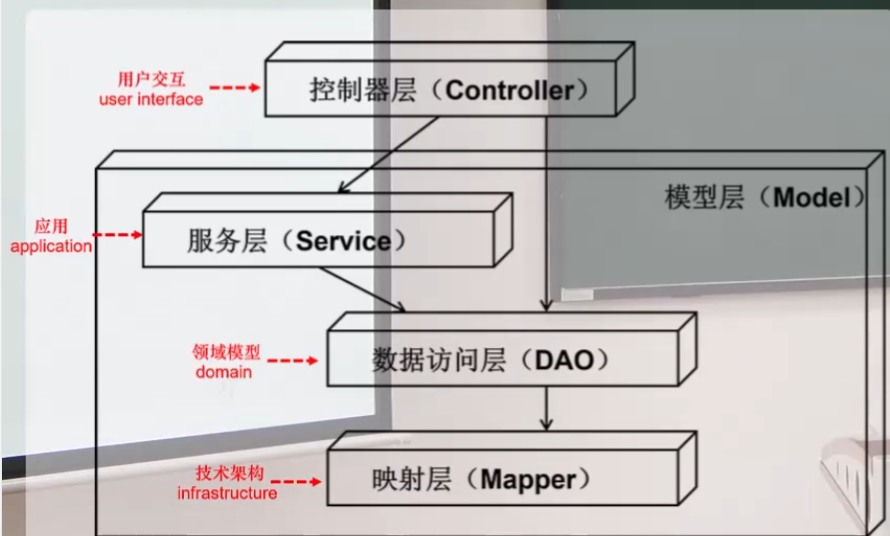
软件体系结构：层次体系结构和



View层是前端部分，Model层是后端部分

Controller层是用来处理前端的和后端的信息交互，是单向的

松散层次的结构：



Controller层：负责数据的转换，也就是交互

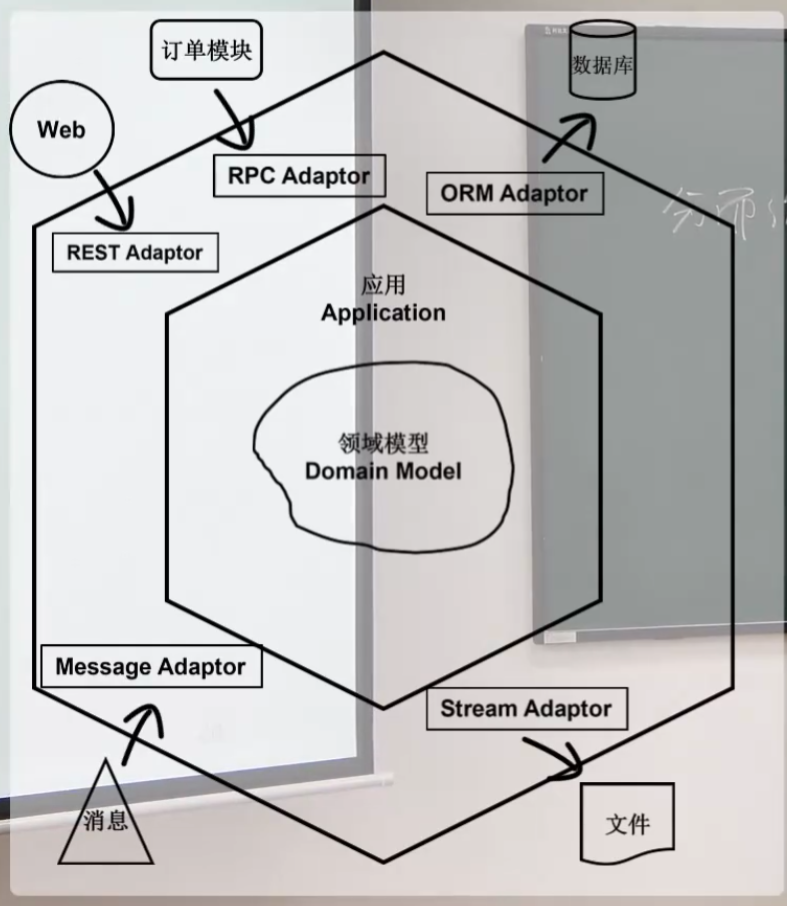
Service层：负责面向对象向面向功能的转换

DAO层：主要负责与数据库进行交互，封装数据的增删改查操作，确保业务逻辑层与数据存储层的解耦

Mapper层：它是应用程序中负责数据持久化的部分，也就是与数据库直接进行交互的层次

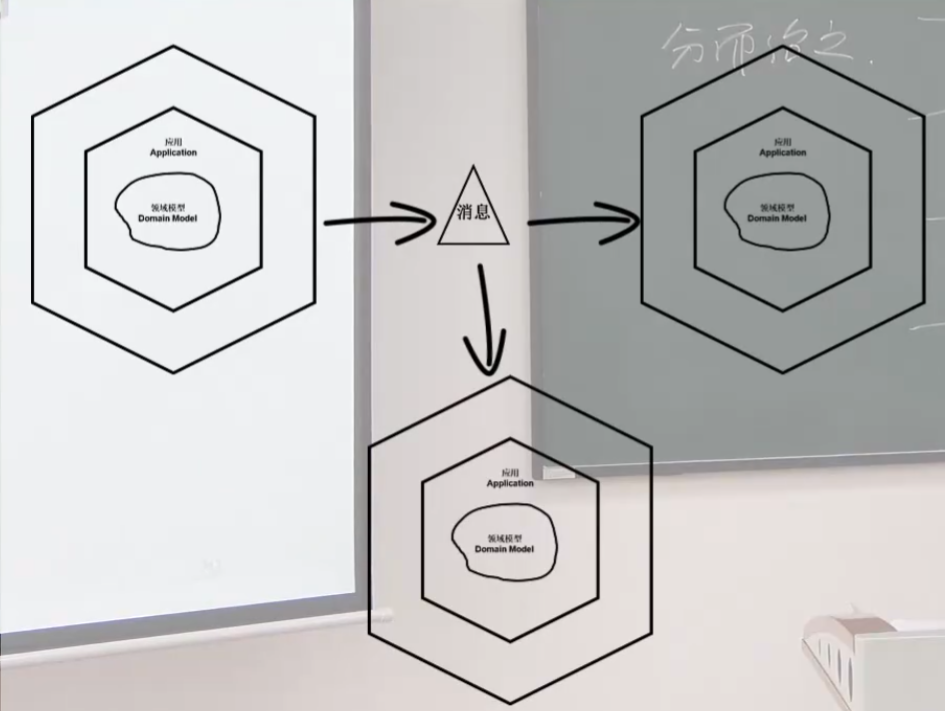
六边形体系结构：

领域模型是核心



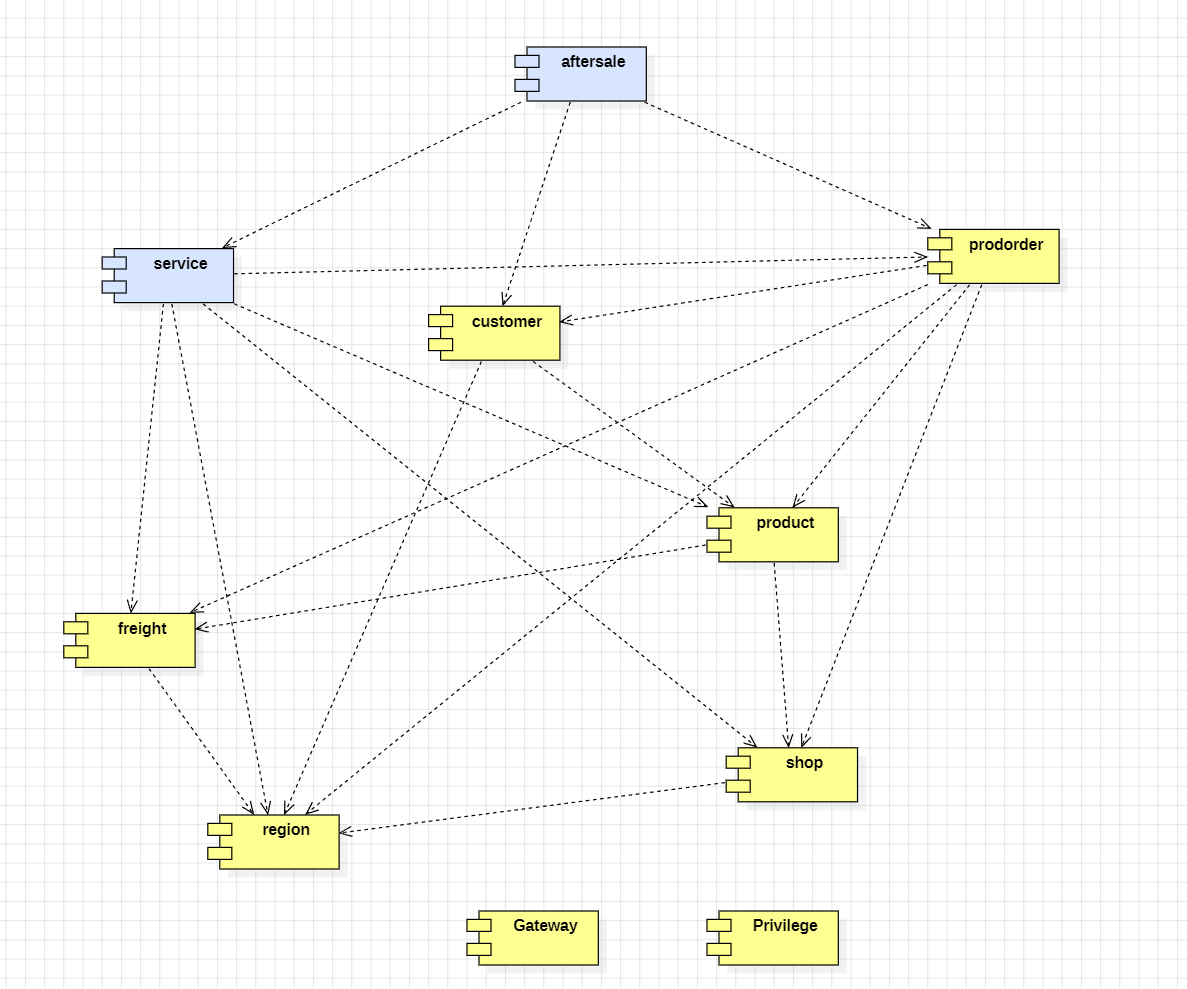
消息驱动的体系结构：

消息是一对多的



组件图和部署图：

组件图



部署图：

