



第五章 静态建模

知识点：包和包图



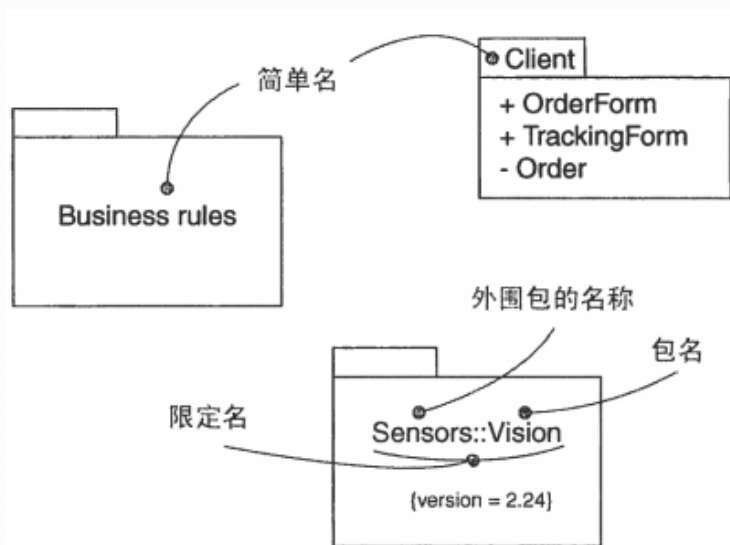


1. 包的定义和用途
2. 包拥有的元素
3. 包之间的关系

- 一个复杂结构有很多个类图，类图之间又有很多关联，形成一个复杂的网络。为了清晰、简洁地描述一个复杂的系统，通常把它们分解成若干较小的系统（子系统）。
- 在UML中使用了“包”的机制，一个包相当于一个子系统。
- 包是UML的模型元素之一，包可以包含其他包和类。
- **包是一种分组机制。**
- 包的实例是没有意义的。因此包仅建模时有用，而不需要转换成可执行的系统。



包的图形表示



- UML中把包画成带标签的文件夹。
- 包的名字放在文件夹中（如果没有展示它的内容）或放在标签上（如果在文件夹里展示内容）。
- 每个包都必须有一个有别于其他包的名称。名称是一个文字串。单独的名称叫做简单名，而限定名是以包所位于的外围包的名称作为前缀的包名，用双冒号 (::) 分隔包名，也称带路径包名。



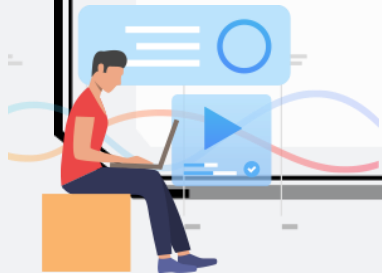
包的构造型

UML中对包预定义的构造型有：

- <<system>>：系统模型
- <<subsystem>>：子系统模型
- <<facade>>：仅仅是其他包的某一个视图，如业务模型中有关产品的信息
- <<stub>>：另一个包公共内容的服务代理
- <<framework>>：模型的体系结构

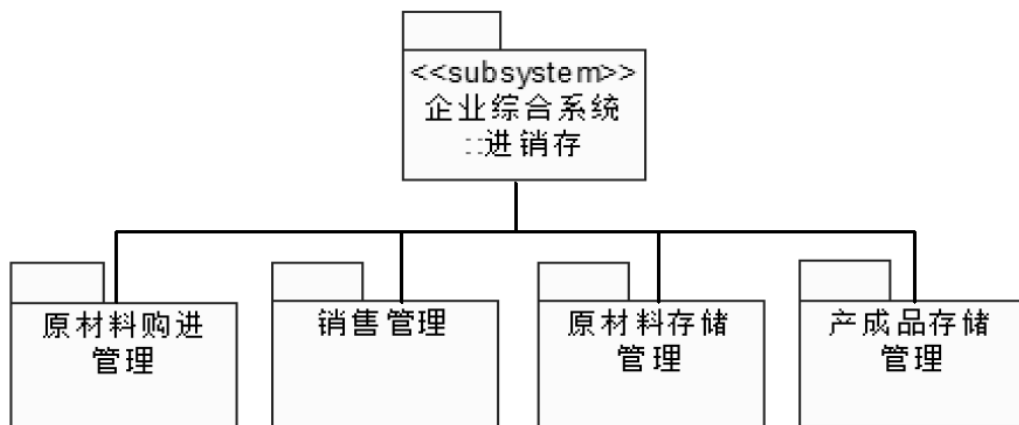


- 包里面可以“拥有”其他元素，这些元素可以是类、接口、构件、结点、协作、用例和图，也可以是其他包。
- 这种“拥有”就是一种组成关系。
- 包形成了一个命名空间，这意味着在一个包的语境中同一种元素的名称必须是唯一的。不同种类的元素可以有相同的名称。
- 可能的话，在不同的包中也应避免重复的名字，以避免造成混乱。
- 拥有关系的语义使包成为一种按规模来处理问题的重要机制。没有包，最后将得到一个庞大的、平铺的模型，其中的所有元素的名称都要唯一，这种情况很难管理。



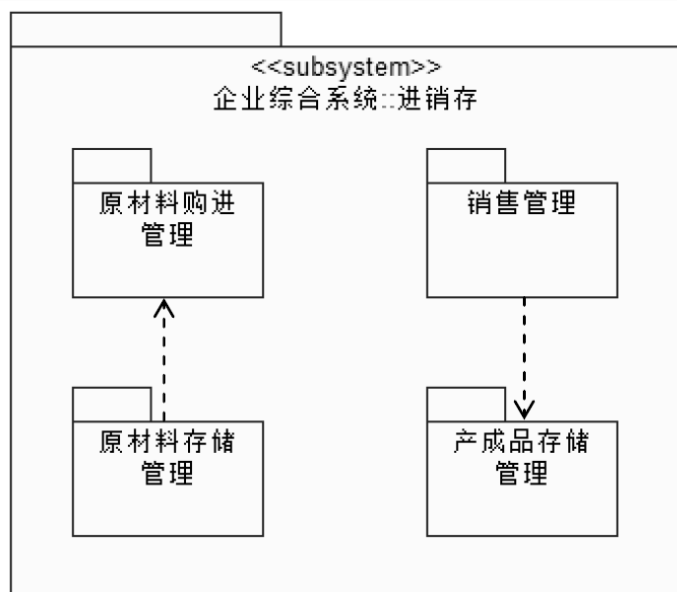
包的嵌套

- 如果包的图符中又含有包，称为包的嵌套。
- UML允许包的嵌套。包的嵌套有两种表示方法：内嵌式表示法和树形层次结构表示法。
- 在实际使用中，最好避免过深地嵌套包，两、三层的嵌套差不多是可管理的极限。

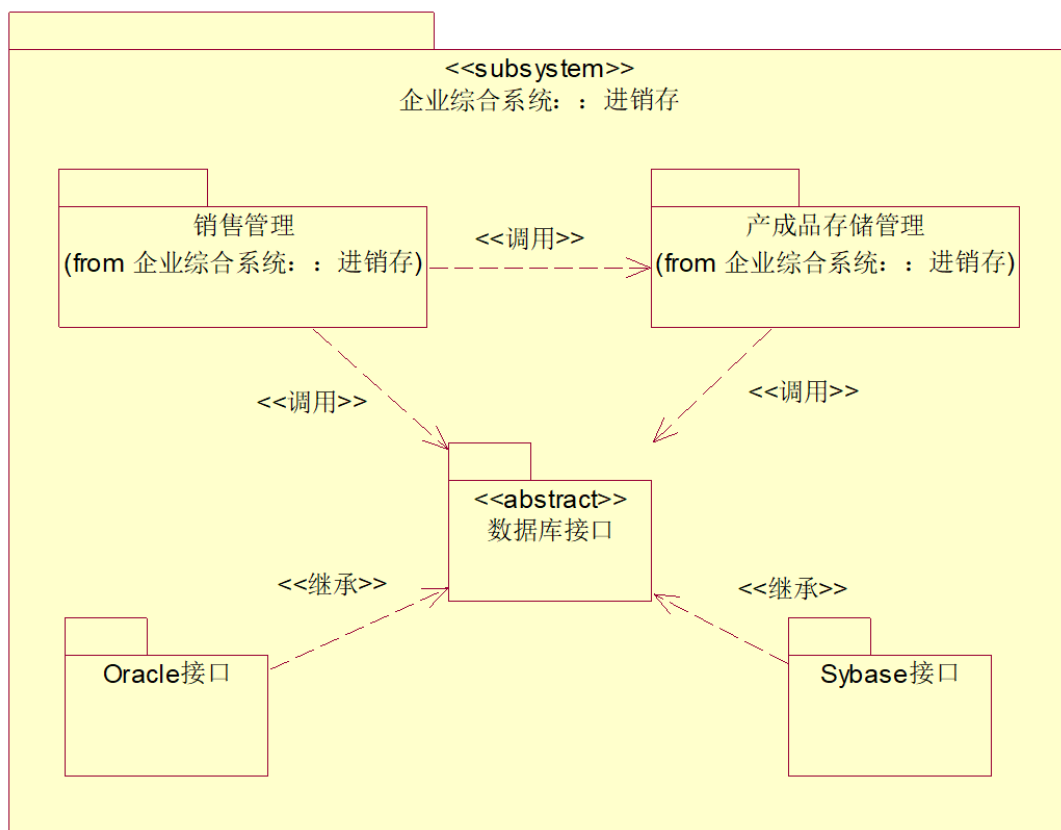


内嵌式表示法

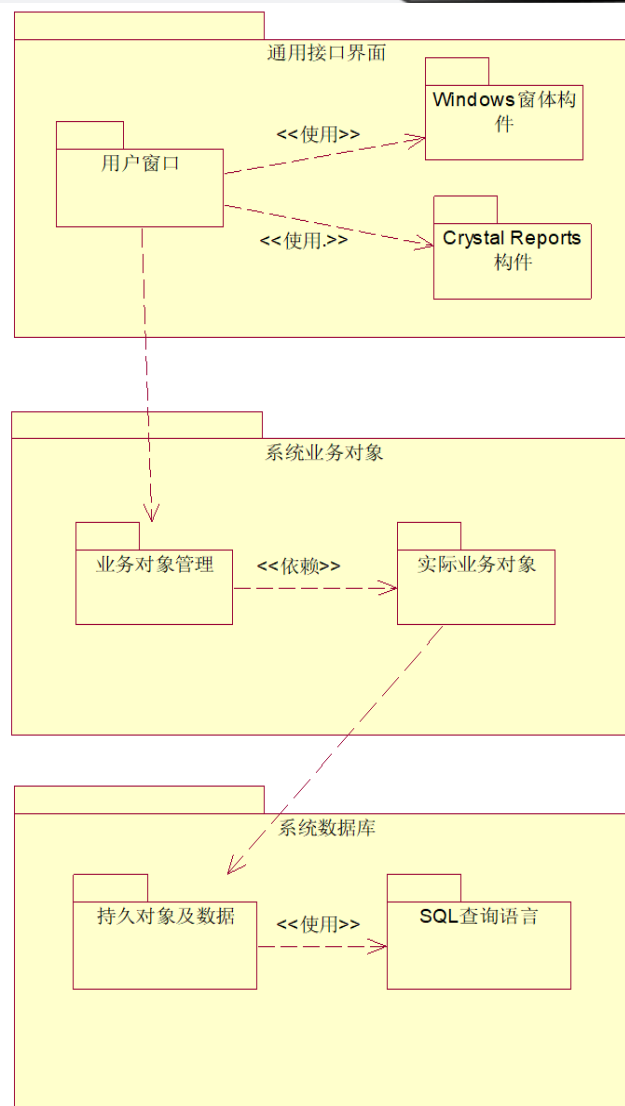
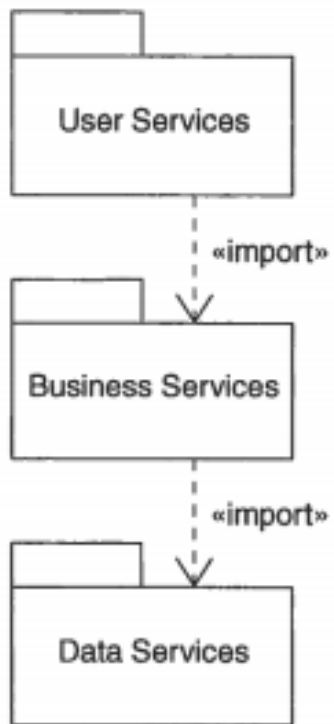
- 一般只表示两层。
- 适合描述包和包之间的关系。



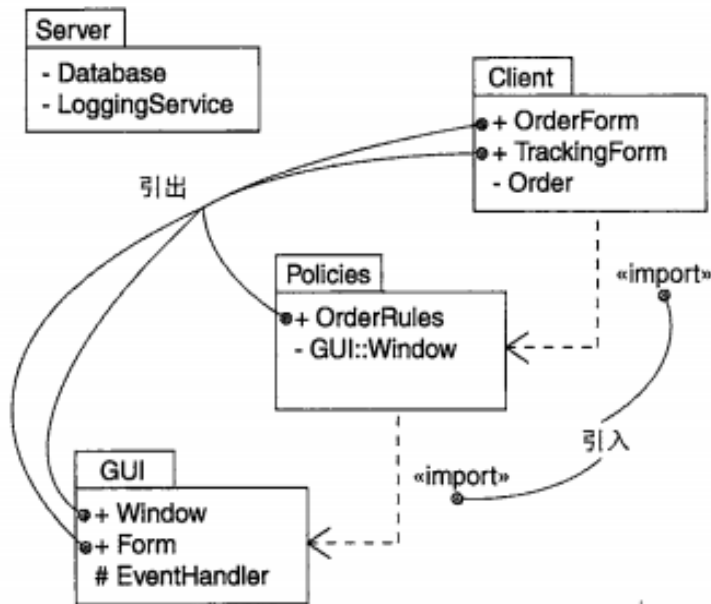
- 依赖：两个包之中的元素（例如类）之间有依赖，则两个包有依赖
- 继承：包之间的继承关系常常是描述了系统的接口



举例



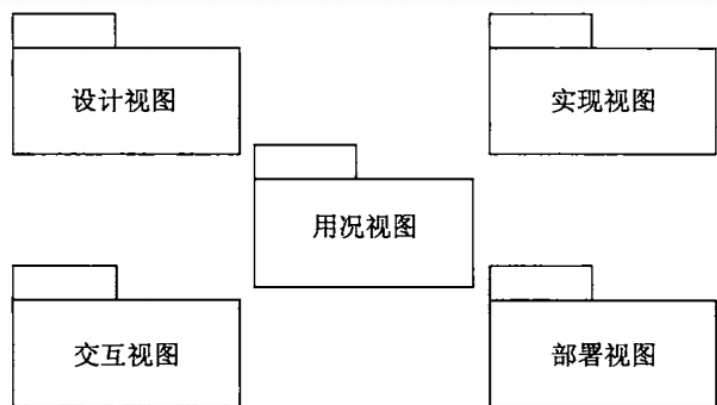
包的引入引出



- 假设有两个名为A和B的并列的类。因为两者是平等的，A能看见B，B也能看见A,因此它们相互依赖。如果两者正好可以组成一个小系统，则不需要任何包装机制。
- 实际系统中可能有几百个这样的类，形成错综复杂的关系网。因此，就需要用受控的包机制来组织抽象。
- 假设A和B类被分别放入两个包里面，而且这两个包并列。再假设A和B都在各自的包里面被声明为公共的。则可以用构造型import修饰的依赖进行建模。
- 包的公共部分被称为它的引出（export）。



包的分组机制是灵活的



- 一个结构良好的包，应满足以下要求：
 - 是内聚的，给出环绕一组相关元素的清晰边界；
 - 是松耦合的，仅引出其他包确实需要看到的那些元素；
 - 嵌套层次不要过深；
 - 内容要均衡，系统的各个包彼此相称，既不要太大，也不要太小。





软件系统分析与设计



苏州大学 杨 洋