UML科普：一篇文章掌握14种UML图

IT牧场 4月29日

以下文章来源于如逆水行舟，作者iisheng

如逆水行舟

坚持学习，未来可期！

点击上方"IT牧场"，选择"设为星标"
技术干货每日送达！

前言

上一篇文章写了一篇建造者模式，其中有几个UML类图，有的读者反馈看不懂了，我们今天就来解决一哈。

什么是UML？

`UML`是`Unified Model Language`的缩写，中文是`统一建模语言`，是由一整套图表组成的标准化建模语言。

为什么要用UML？

通过使用UML使得在软件开发之前， 对整个软件设计有更好的可读性，可理解性，从而降低开发风险。同时，也能方便各个开发人员之间的交流。

UML提供了极富表达能力的建模语言，可以让软件开发过程中的不同人员分别得到自己感兴趣的信息。

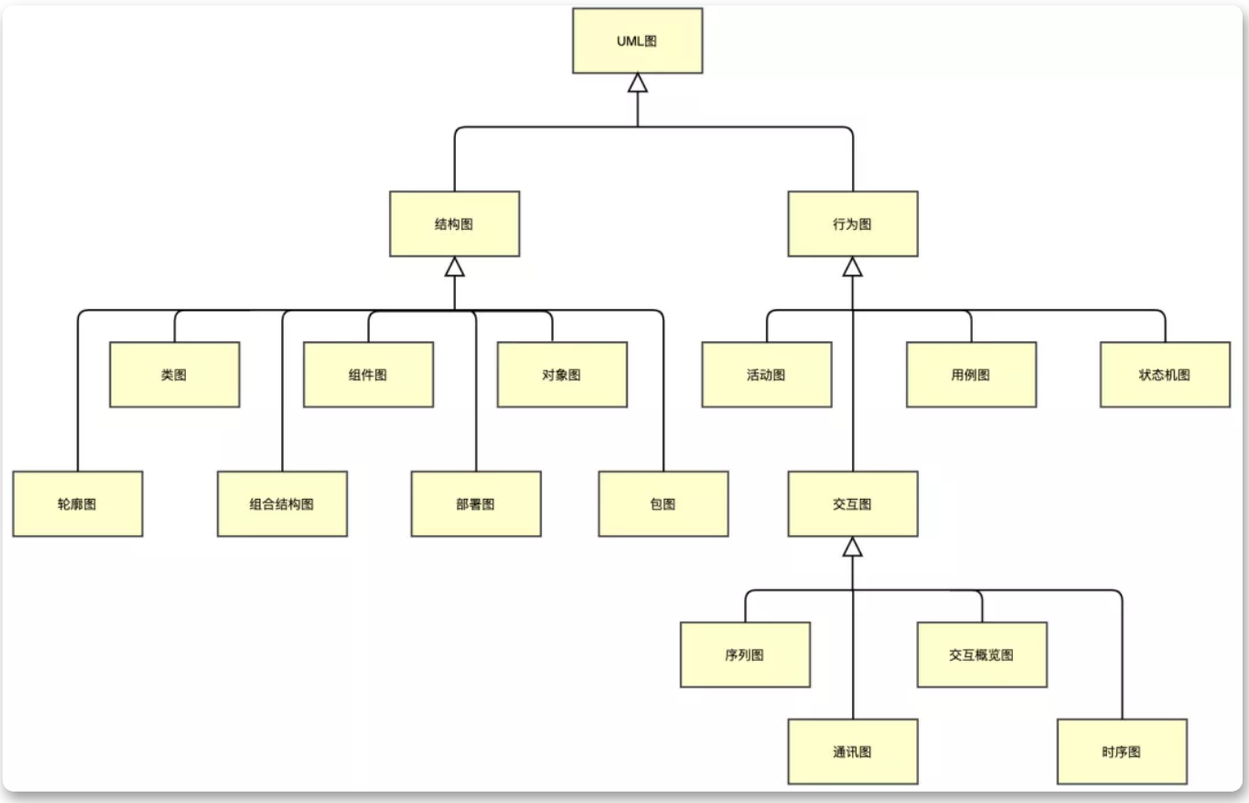
`Page-Jones` 在《Fundamental Object-Oriented Design in UML》一书中总结了UML的主要目的，如下：

1. 为用户提供现成的、有表现力的可视化建模语言，以便他们开发和交换有意义的模型。
2. 为核心概念提供可扩展性（Extensibility）和特殊化（Specialization）机制。
3. 独立于特定的编程语言和开发过程。
4. 为了解建模语言提供一个正式的基础。
5. 鼓励面向对象工具市场的发展。
6. 支持更高层次的开发概念，如协作，框架，模式和组件。
7. 整合最佳的工作方法（Best Practices）。

UML图有哪些？

- UML图分为结构图和行为图。
- 结构图分为类图、轮廓图、组件图、组合结构图、对象图、部署图、包图。

- 行为图又分活动图、用例图、状态机图和交互图。
- 交互图又分为序列图、时序图、通讯图、交互概览图。



UML图概览

UML类型	目的	版本
类图	描述了系统中对象的类型以及它们之间存在的各种静态关系。	UML 1.x
组件图	描绘了系统中组件提供的、需要的接口、端口等，以及它们之间的关系。	UML 1.x，UML 2.0重新定义了
对象图	对象图是类图的一个实例，是系统在某个时间点的详细状态的快照。	UML 1.x
轮廓图	轮廓图提供了一种通用的扩展机制，用于为特定域和平台定制UML模型。	UML 2.0
组合结构图	描述了一个"组合结构"的内部结构，以及他们之间的关系。	UML 2.0
部署图	描述了系统内部的软件如何分布在不同的节点上。	UML 1.x
包图	描绘了系统在包层面上的结构设计。	UML 2.0
用例图	指由参与者、用例，边界以及它们之间的关系构成的用于描述系统功能的视图。	UML 1.x
活动图	描述了具体业务用例的实现流程。	UML 1.x
状态机图	描述了对象在它的整个生命周期里，响应不同事件时，执行相关事件的顺序。	UML 1.x
序列图	描述了在用例的特定场景中，对象如何与其他对象交互。	UML 1.x
时序图	时序图被用来显示随时间变化，一个或多个元素的值或状态的更改。	UML 2.0
交互概览图	交互概览图与活动图类似，但是它的节点是交互图。	UML 2.0
通讯图	描述了收发消息的对象的组织关系，强调对象之间的合作关系而不是时间顺序。	UML 1.x叫协作图，UML 2.0改名了

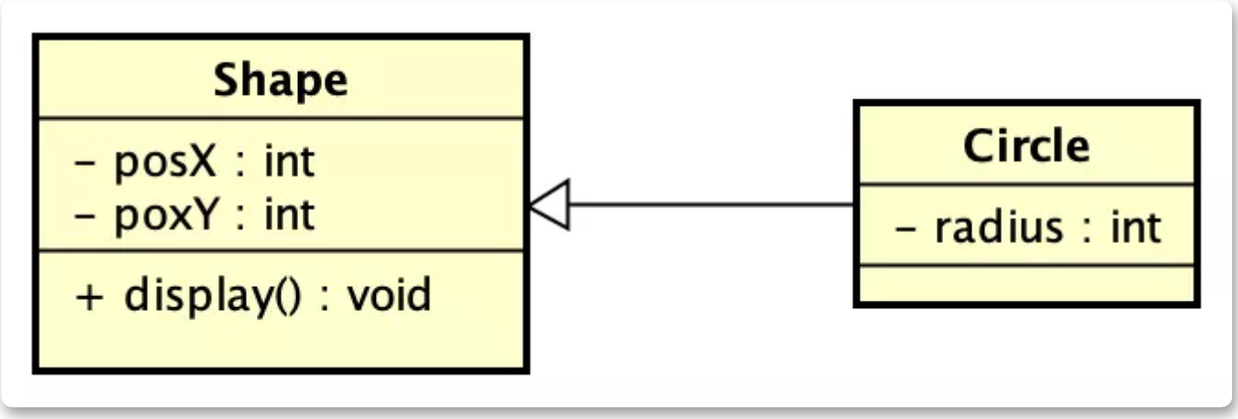
什么是类图？

- 【概念】 类图是一切面向对象方法的核心建模工具。类图描述了系统中对象的类型以及它们之间存在的各种静态关系。
- 【目的】用来表示类、接口以及它们之间的静态结构和关系。

在类图中，常见的有以下几种关系。

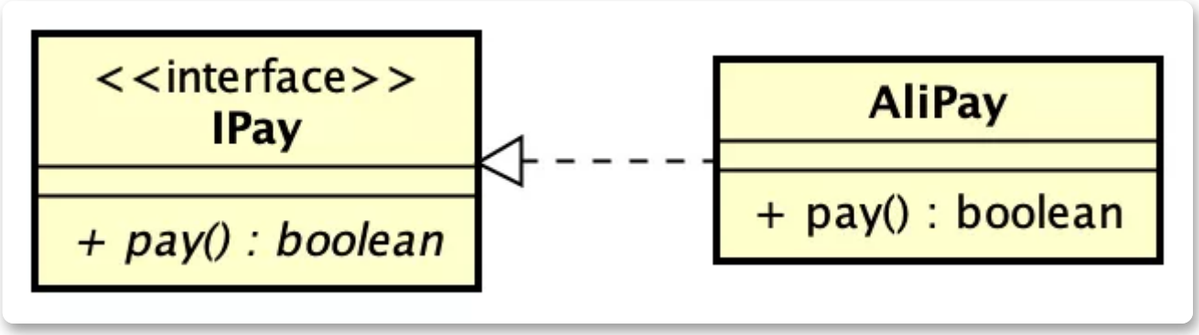
泛化（Generalization）

- 【泛化关系】是一种继承关系，表示子类继承父类的所有特征和行为。
- 【箭头指向】带三角箭头的实线，箭头指向父类。



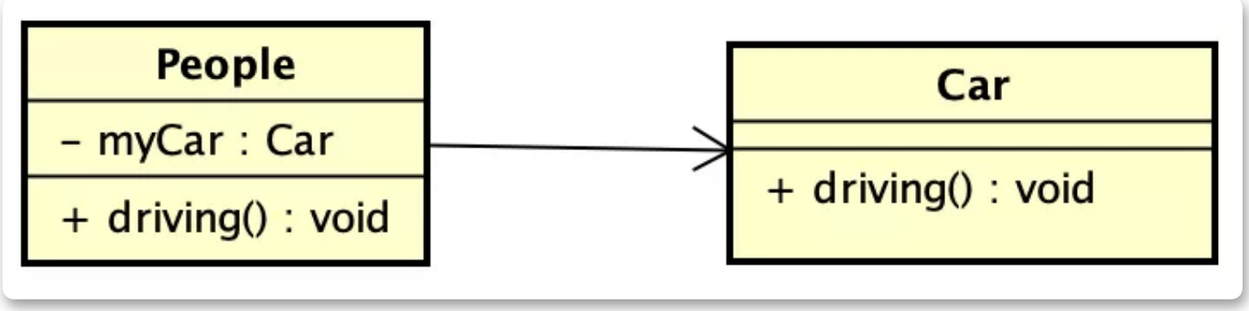
实现（Realization）

- 【实现关系】是一种类与接口的关系，表示类是接口所有特征和行为的实现。
- 【箭头指向】带三角箭头的虚线，箭头指向接口。



关联（Association）

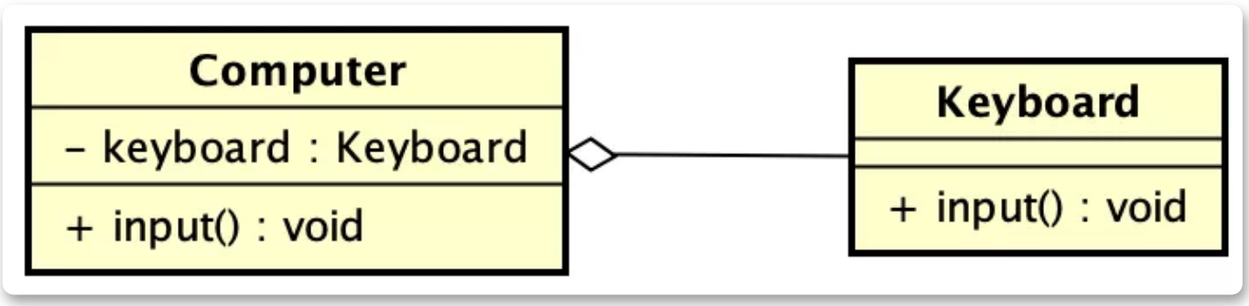
- 【关联关系】是一种拥有关系，它使得一个类知道另一个类的属性和方法。
- 【代码体现】成员变量
- 【箭头指向】带普通箭头的实线，指向被拥有者。双向的关联可以有两个箭头，或者没有箭头。单向的关联有一个箭头。



> 自己买的车，想什么时候开就开。但是车是车，人是人，没有整体与部分的关系。

聚合（Aggregation）

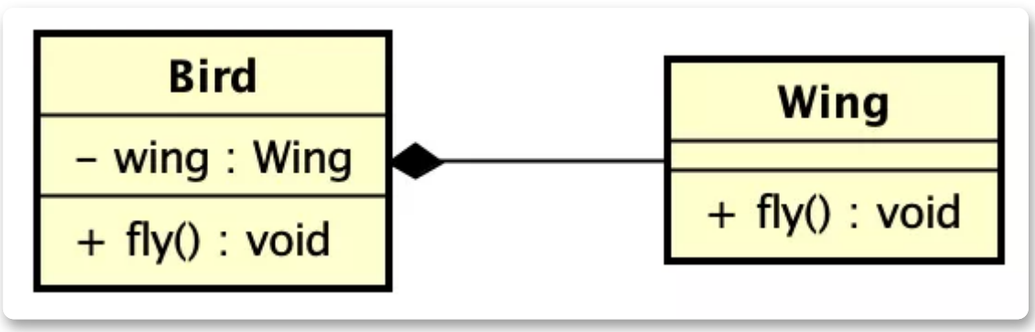
- 【聚合关系】是一种整体与部分的关系。且部分可以离开整体而单独存在。聚合关系是关联关系的一种，是强的关联关系；关联和聚合在语法上无法区分，必须考察具体的逻辑关系。
- 【代码体现】成员变量
- 【箭头指向】带空心菱形的实线，空心菱形指向整体。



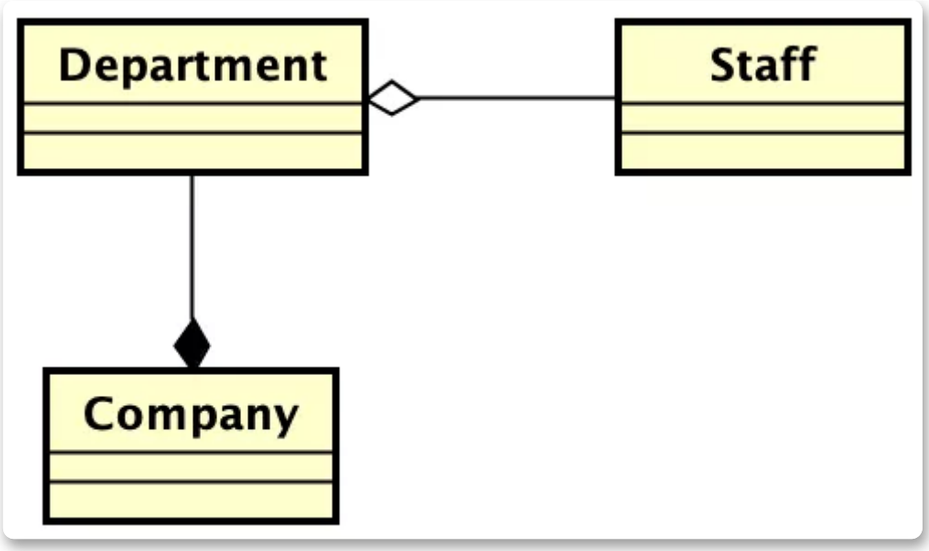
> 电脑有键盘才能输入信息，电脑是整体，键盘是部分，键盘也可以离开电脑，单纯的拿去敲。所以是聚合。

组合（Composition）

- 【组合关系】是一种整体与部分的关系。但部分不能离开整体而单独存在，组合关系是关联关系的一种，是比聚合关系还要强的关系。
- 【代码体现】成员变量
- 【箭头指向】带实心菱形和普通箭头的实线，实心菱形指向整体。



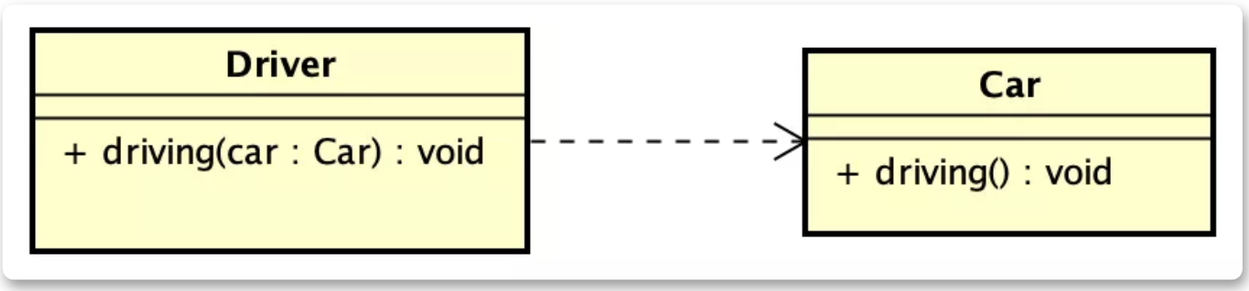
> 鸟是整体，翅膀是部分。鸟死了，翅膀也就不能飞了。所以是组合。我们再看一下，下面的一组经典的聚合组合关系的例子。



一个公司拥有多个部门，公司和部门之间是组合关系，公司破产了，部门就不复存在了。部门和员工是聚合关系，部门被裁掉，员工就换下家了。

依赖（Dependency）

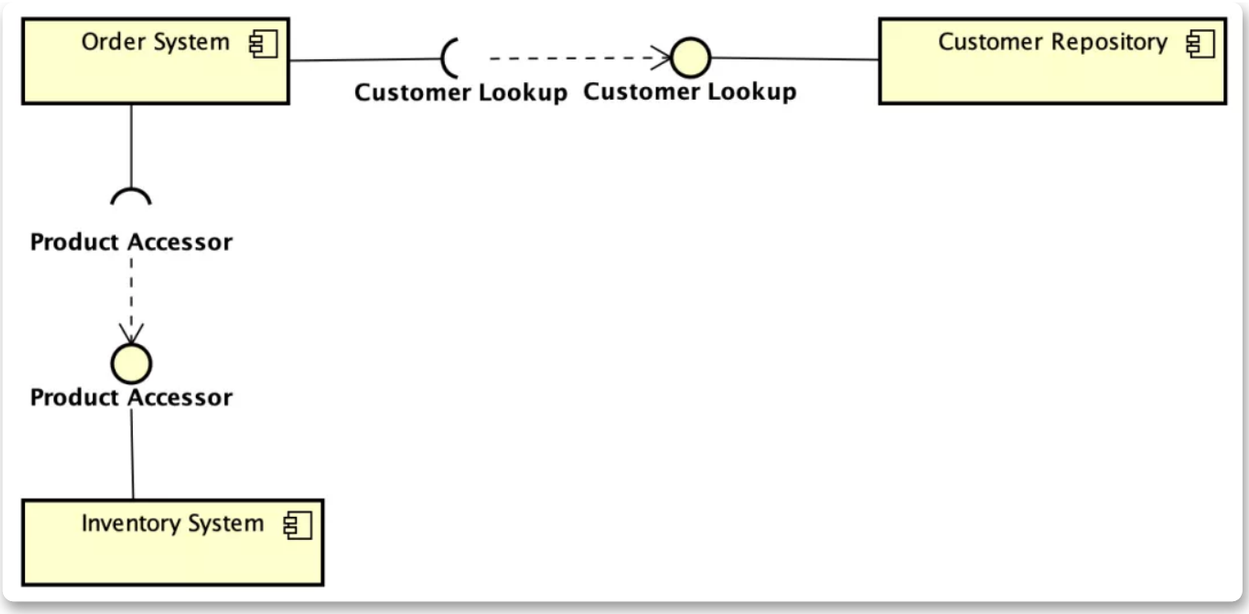
- 【依赖关系】是一种使用关系，即一个类的实现需要另一个类的协助。
- 【箭头指向】带普通箭头的虚线，普通箭头指向被使用者。



老司机只管开车，车是谁的不重要，给什么车开什么车。

什么是组件图？

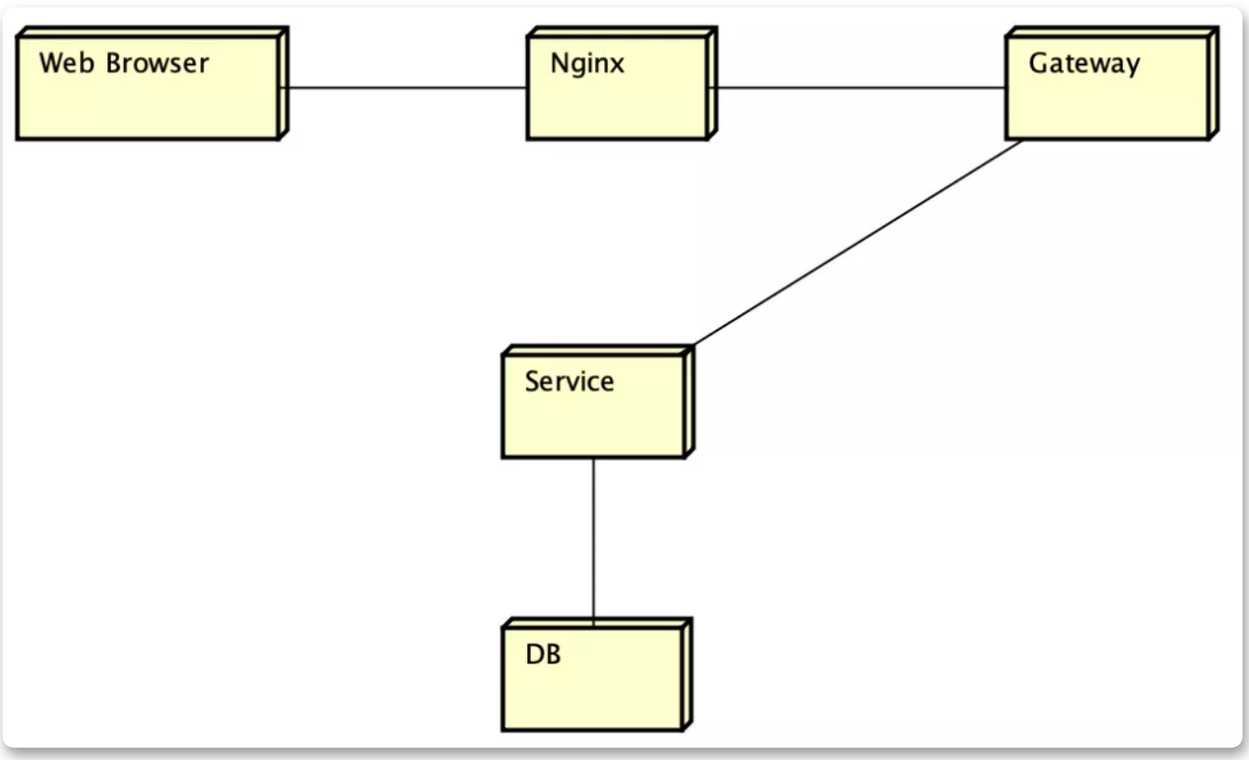
- 【概念】描绘了系统中组件提供的、需要的接口、端口等，以及它们之间的关系。
- 【目的】用来展示各个组件之间的依赖关系。



订单系统组件依赖于客户资源库和库存系统组件。中间的虚线箭头表示依赖关系。另外两个符号，表示组件连接器，一个提供接口，一个需要接口。

什么是部署图？

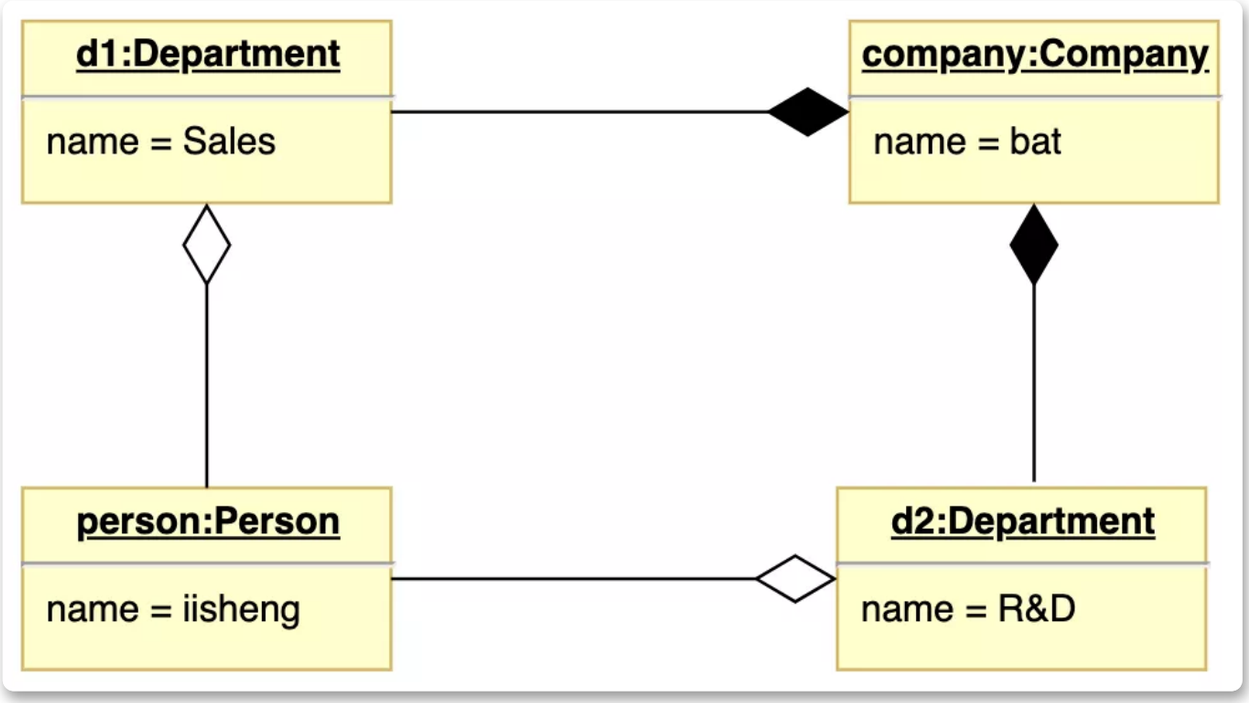
- 【概念】描述了系统内部的软件如何分布在不同的节点上。
- 【目的】用来表示软件和硬件的映射关系。



图中简单的表示，不同机器上面部署的不同软件。

什么是对象图？

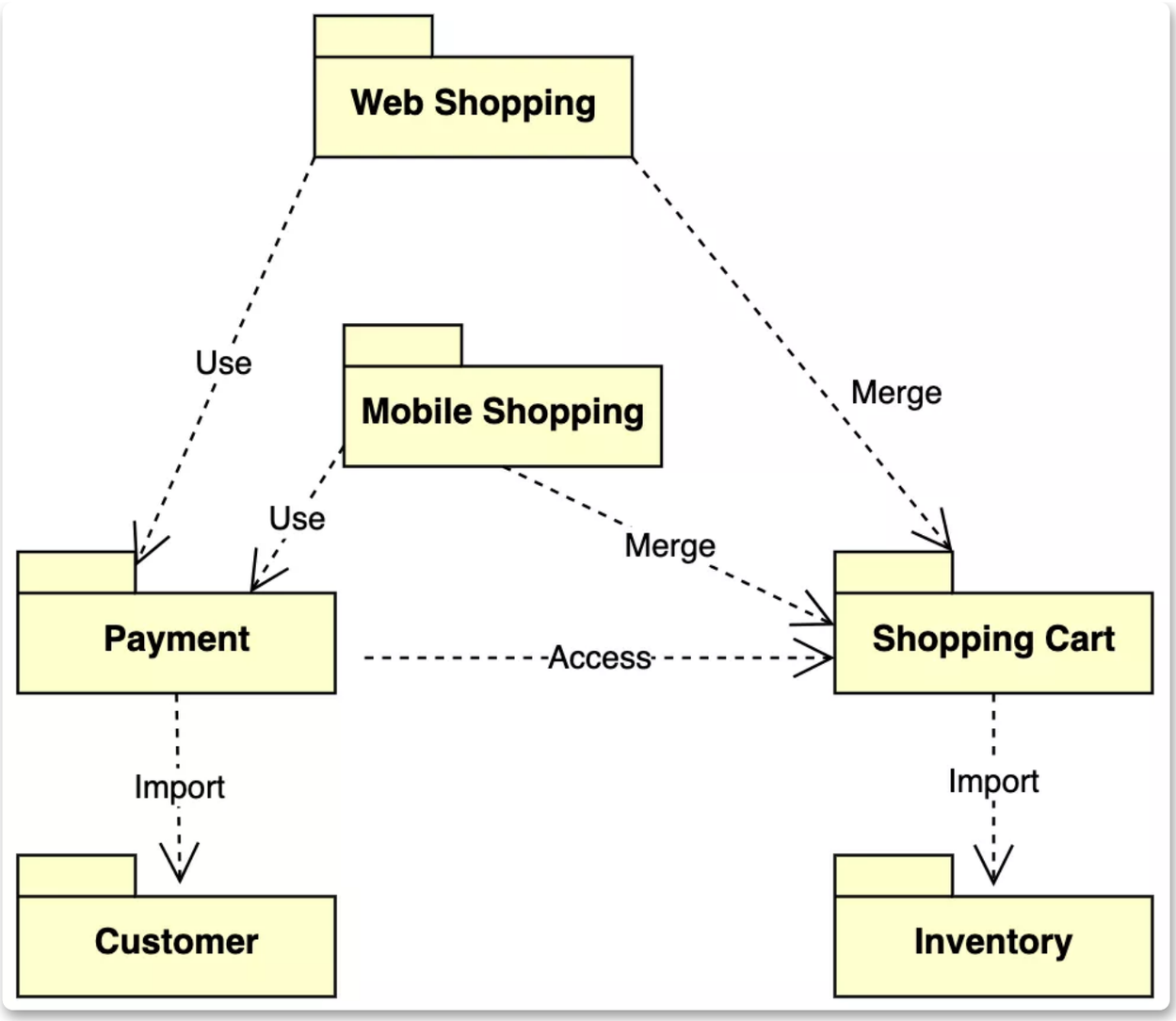
- 【概念】对象图是类图的一个实例，是系统在某个时间点的详细状态的快照。
- 【目的】用来表示两个或者多个对象之间在某一时刻之间的关系。



图中就是描述的，某时间点`bat`这个公司有一个研发部，一个销售部，两个部门只有一个人`iisheng`。

什么是包图？

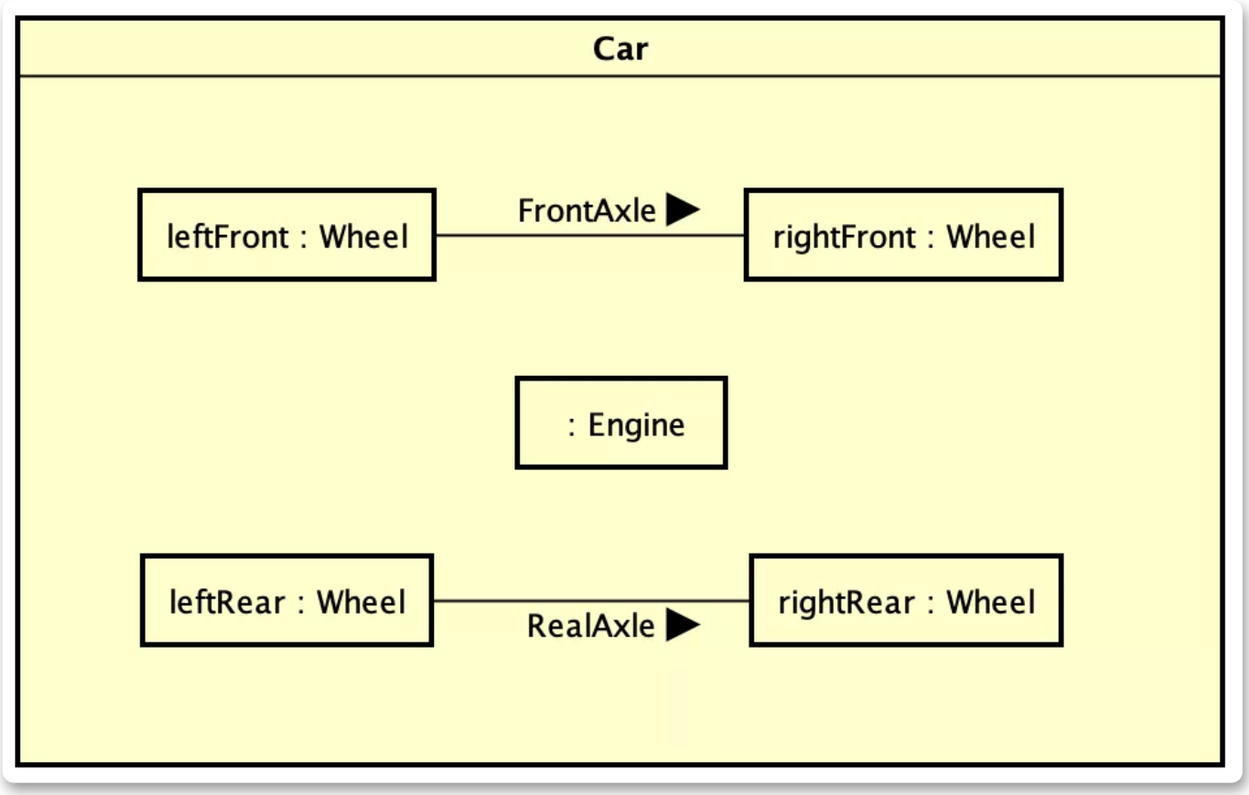
- 【概念】描绘了系统在包层面上的结构设计。
- 【目的】用来表示包和包之间的依赖关系。



- 《Use》关系表示使用依赖，`Web Shopping`依赖`Payment`
- 《Merge》关系表示合并，`Web Shopping`合并了`Shopping Cart`就拥有了`Shopping Cart`的功能
- 《Access》关系表示私有引入，比如代码中的指定包名类名
- 《Import》关系表示公共引入，比如Java中的`import`之后，就可以直接使用`import`包中的类了。

什么是组合结构图？

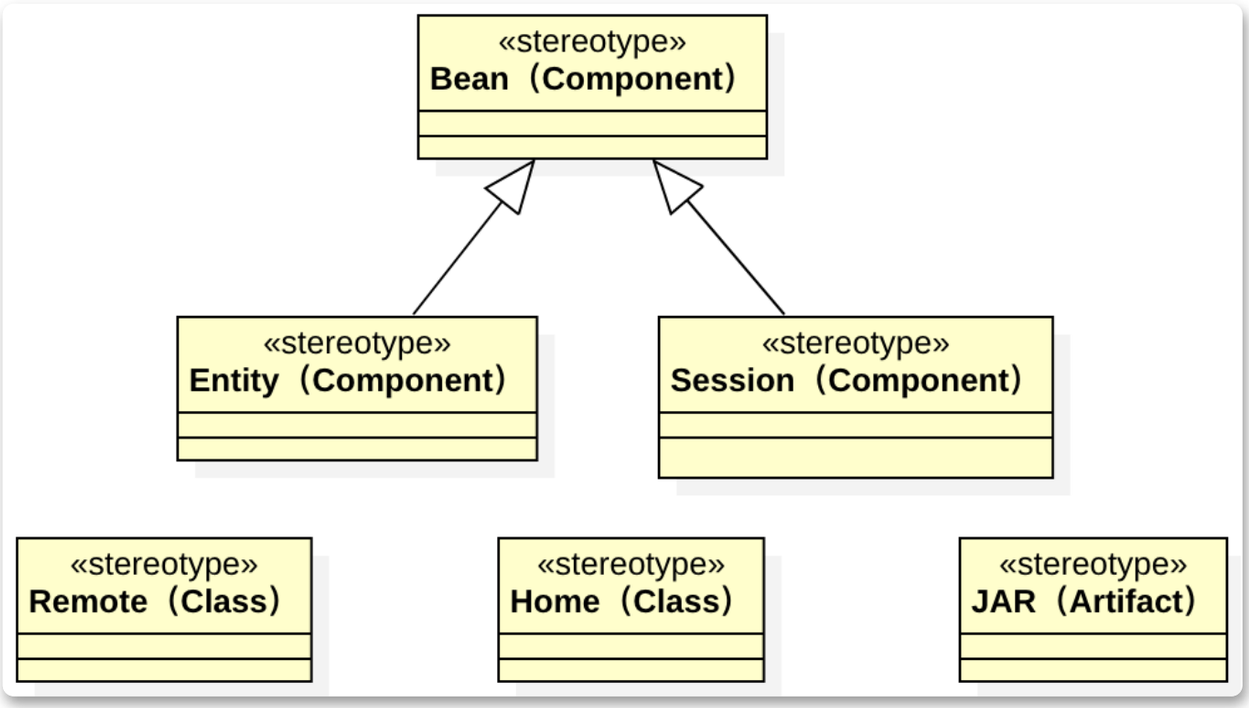
- 【概念】描述了一个"组合结构"的内部结构，以及他们之间的关系。这个"组合结构"可以是系统的一部分，或者一个整体。
- 【目的】用来表示系统中逻辑上的"组合结构"。



图中描述了 `Car` 是由车轴连接着的两个前面轮子、两个后面轮子，和引擎组合的。

什么是轮廓图？

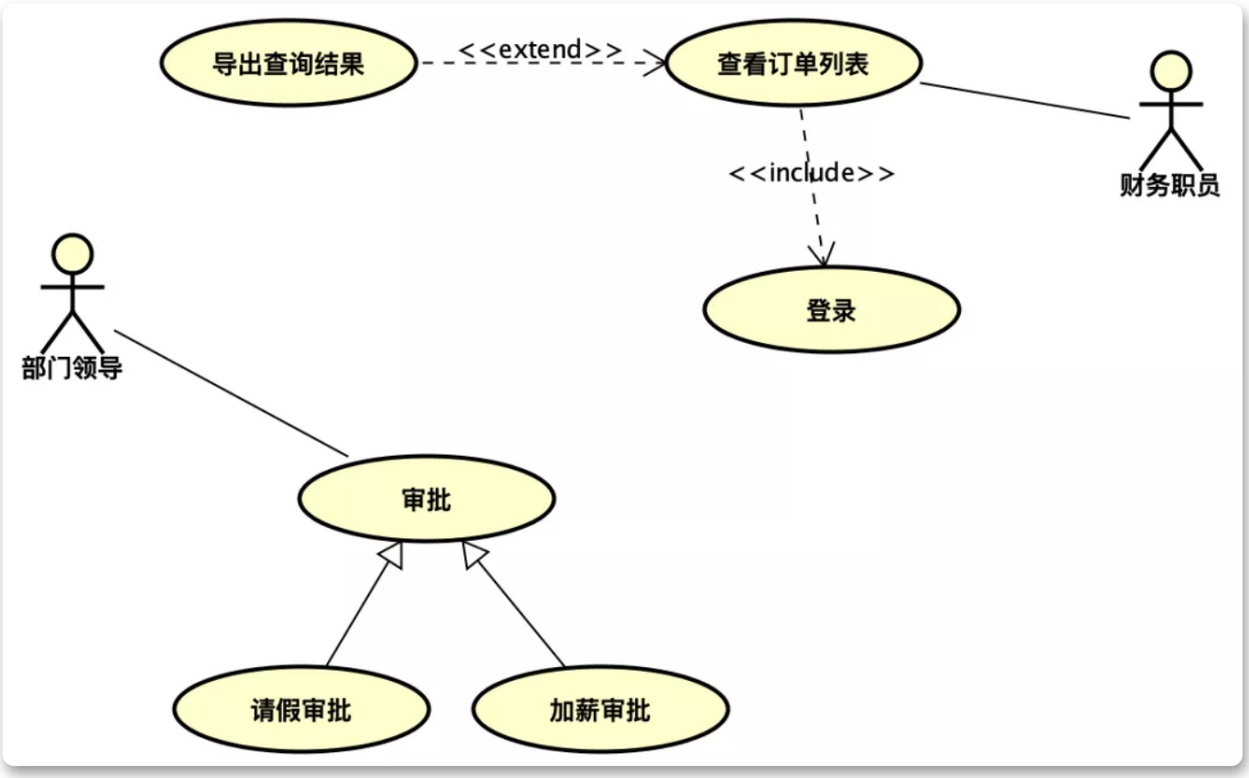
- 【概念】轮廓图提供了一种通用的扩展机制，用于为特定域和平台定制UML模型。
- 【目的】用于在特定领域中构建UML模型。



图中我们定义了一个简易的 `EJB` 的概要图。`Bean` 是从 `Component` 扩展来的。`Entity Bean` 和 `Session Bean` 继承了 `Bean`。`EJB` 拥有 `Remote` 和 `Home` 接口，和 `JAR` 包。

什么是用例图？

- 【概念】用例图是指由参与者、用例，边界以及它们之间的关系构成的用于描述系统功能的视图。
- 【目的】用来描述整个系统的功能。

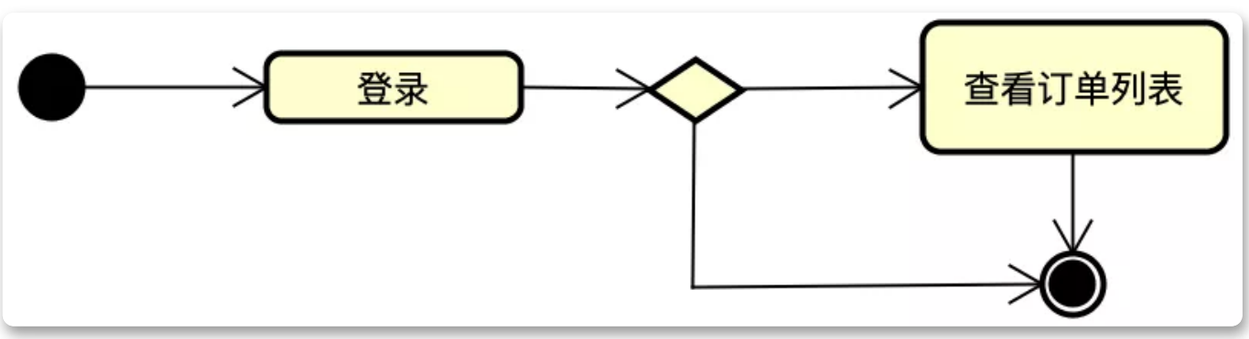


用例图中包含以下三种关系：

- 包含关系使用符号《include》，想要查看订单列表，前提是需要先登录。
- 扩展关系使用符号《extend》，基于查询订单列表的功能，可以增加一个导出数据的功能
- 泛化关系，子用例继承父用例所有结构、行为和关系。

什么是活动图？

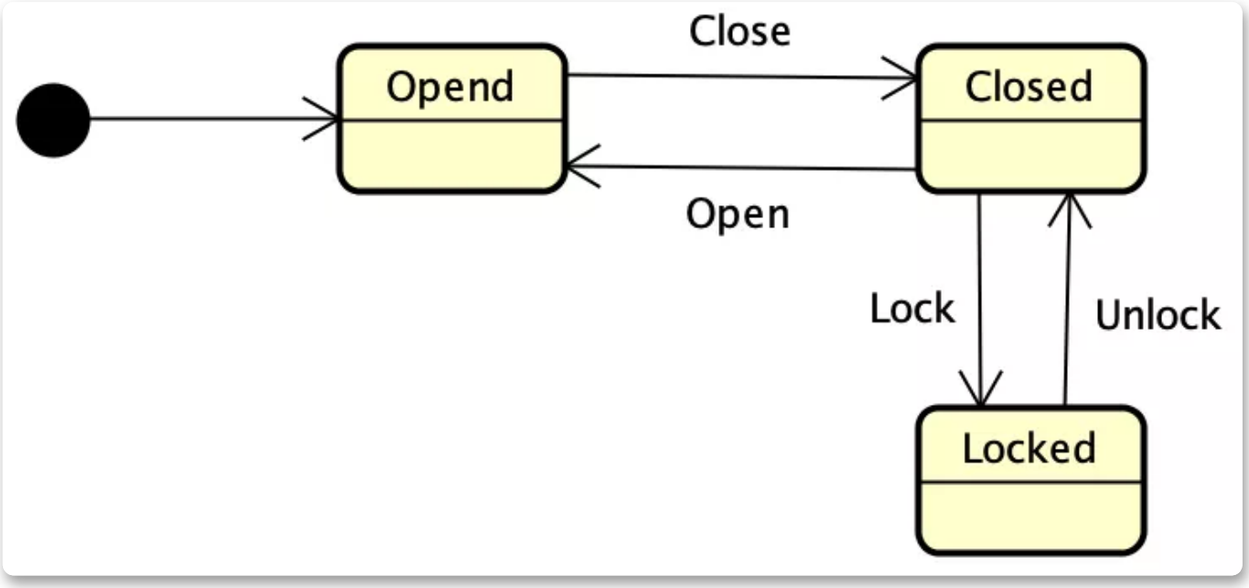
- 【概念】描述了具体业务用例的实现流程。
- 【目的】用来表示用例实现的工作流程。



图中简单描述了，从开始到登录到查看订单列表，或者登录失败直接结束。

什么是状态机图？

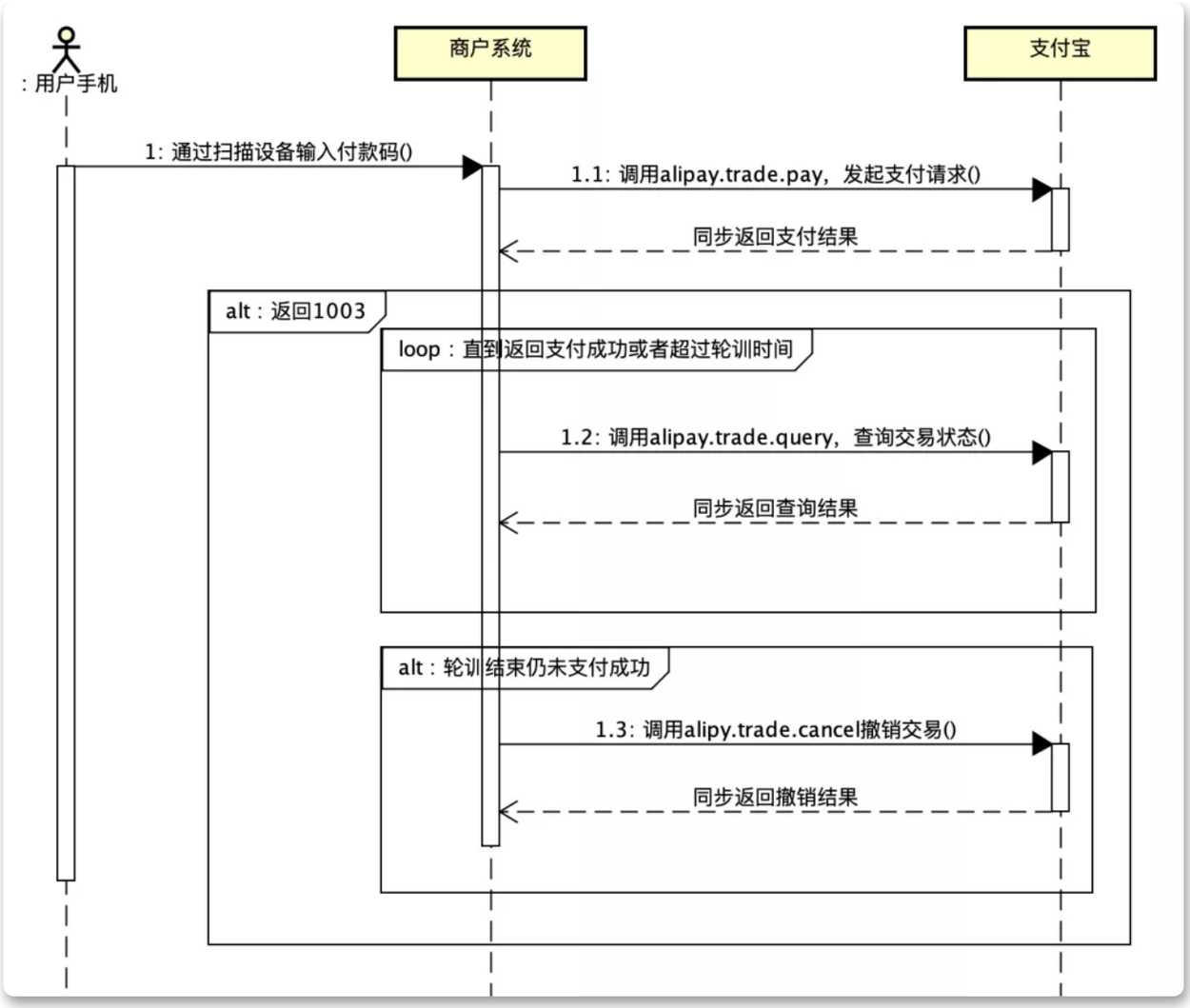
- 【概念】状态机图对一个单独对象的行为建模，指明对象在它的整个生命周期里，响应不同事件时，执行相关事件的顺序。
- 【目的】用来表示指定对象，在整个生命周期，响应不同事件的不同状态。



图中描述了，门在其生命周期内所经历的状态。

什么是序列图？

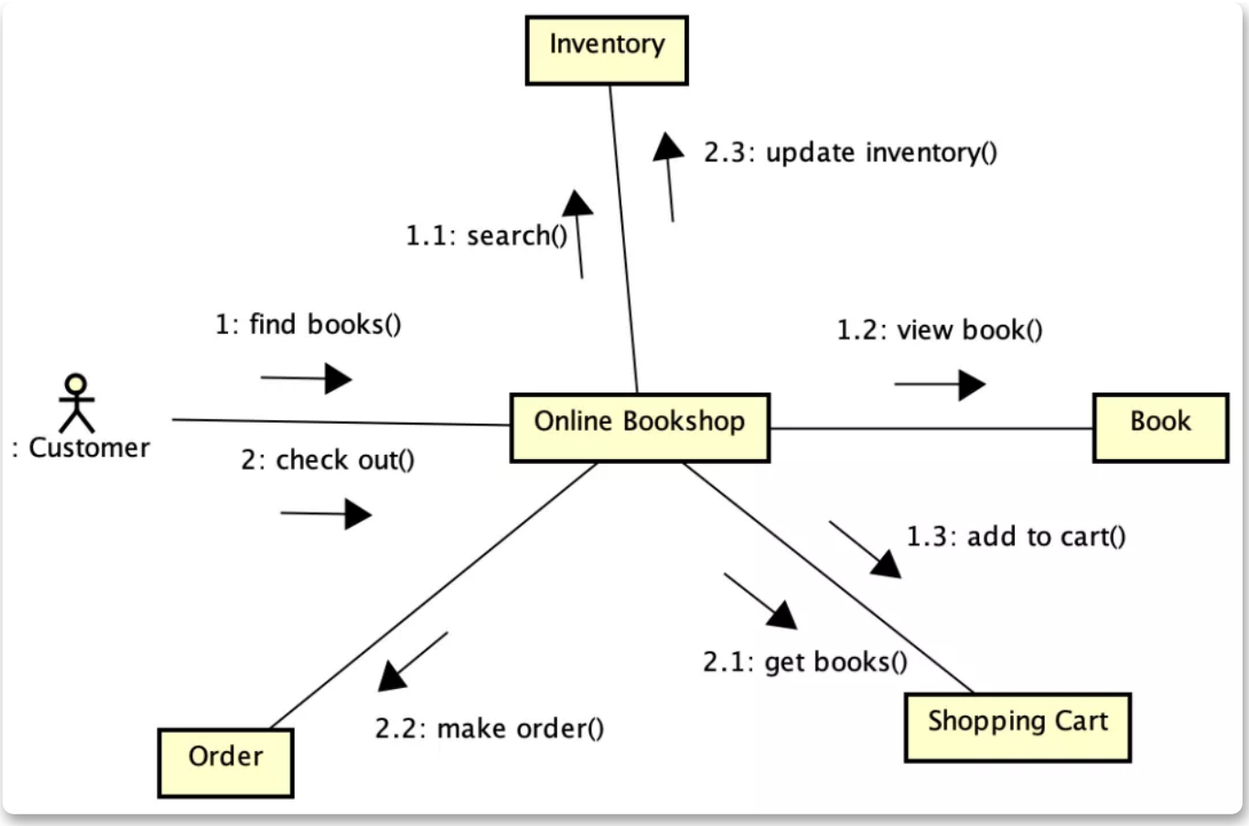
- 【概念】序列图根据时间序列展示对象如何进行协作。它展示了在用例的特定场景中，对象如何与其他对象交互。
- 【目的】通过描述对象之间发送消息的时间顺序显示多个对象之间的动态协作。



图中展示的是支付宝条码支付场景的序列图。其中，`loop`是循环，`alt`是选择，序列图的其他关系这里就不介绍了。

什么是通讯图？

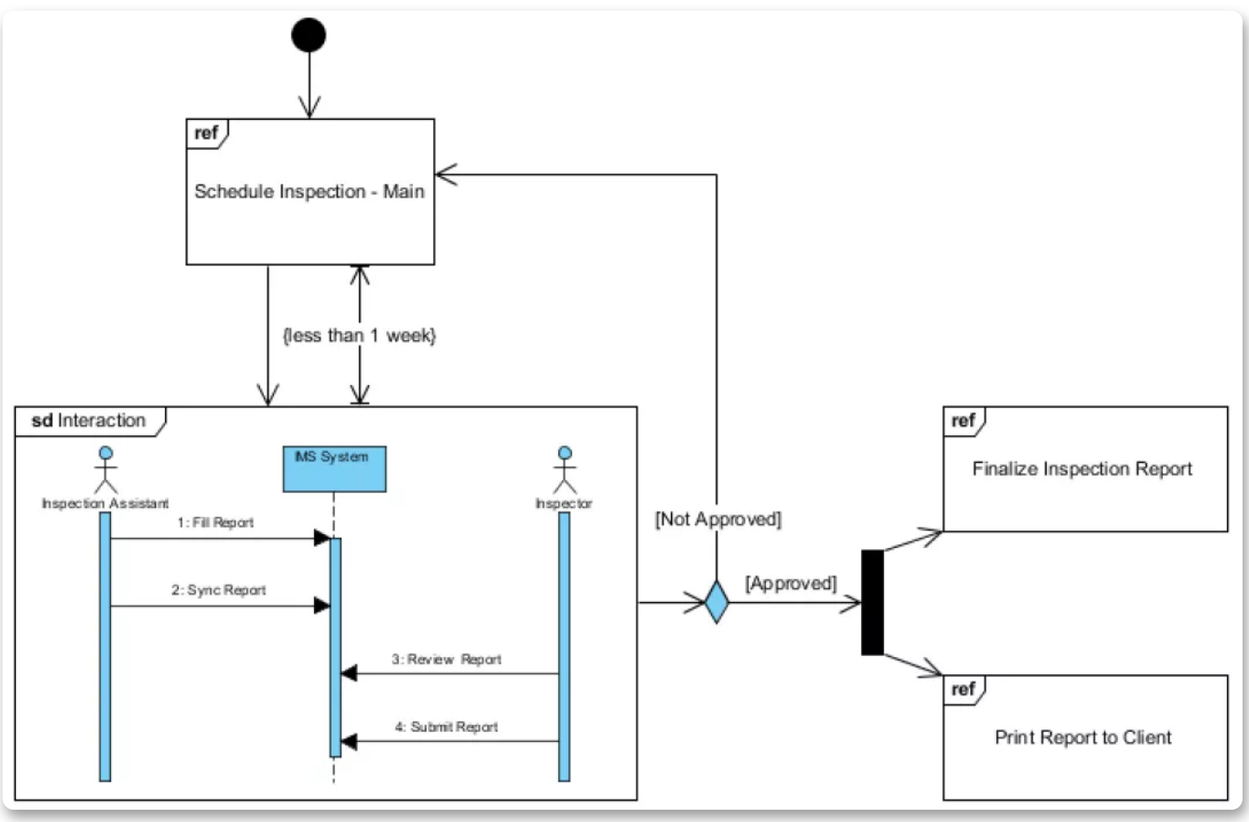
- 【概念】描述了收发消息的对象的组织关系，强调对象之间的合作关系而不是时间顺序。
- 【目的】用来显示不同对象的关系。



图中展示了一个线上书店的通讯图，方框和小人表示生命线，不同生命线之间可以传递消息，消息前面的数字可以表达序列顺序。

什么是交互概览图？

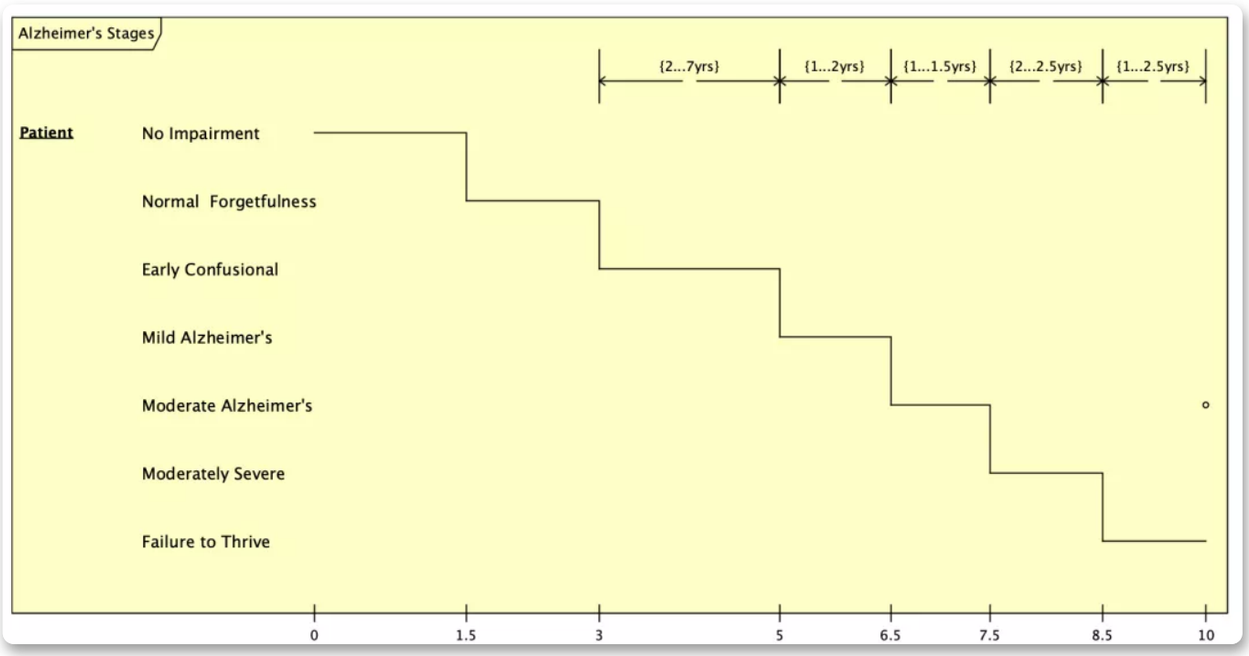
- 【概念】交互概览图与活动图类似，但是它的节点是交互图。
- 【目的】提供了控制流的概述。



图中表示一个调度系统的交互概览图，跟活动图很像。其中`sd`的框代表具体的交互流程，`ref`框代表使用交互。

什么是时序图？

- 【概念】时序图被用来显示随时间变化，一个或多个元素的值或状态的更改。也显示时控事件之间的交互和管理它们的时间和期限约束。
- 【目的】用来表示元素状态或者值随时间的变化而变化的视图。



图中展示了老年痴呆病人随着时间的变化病情的变化。

总结

学习UML，我们没必要纠结比如像聚合关系是带箭头还是不带箭头，这样的问题。更重要的是UML图所给我们带来的画图思想，让我们画UML图或者其他图能让其他人更好的理解我们的设计思想。

当然，你要是明确知道带箭头或者不带箭头哪个是错误的，欢迎留言告诉我。

参考文献：

[1]:《Learning UML 2.0》

[2]: https://www.uml-diagrams.org/

[3]: https://www.visual-paradigm.com/guide/

[4]: https://sparxsystems.com/resources/tutorials/