领域建模

鹿传明
WINDLU@ITLEAD.COM.CN



中睿信息技术有限公司
ZHONGRUI INFO & TECHNOLOGY CO.,LTD

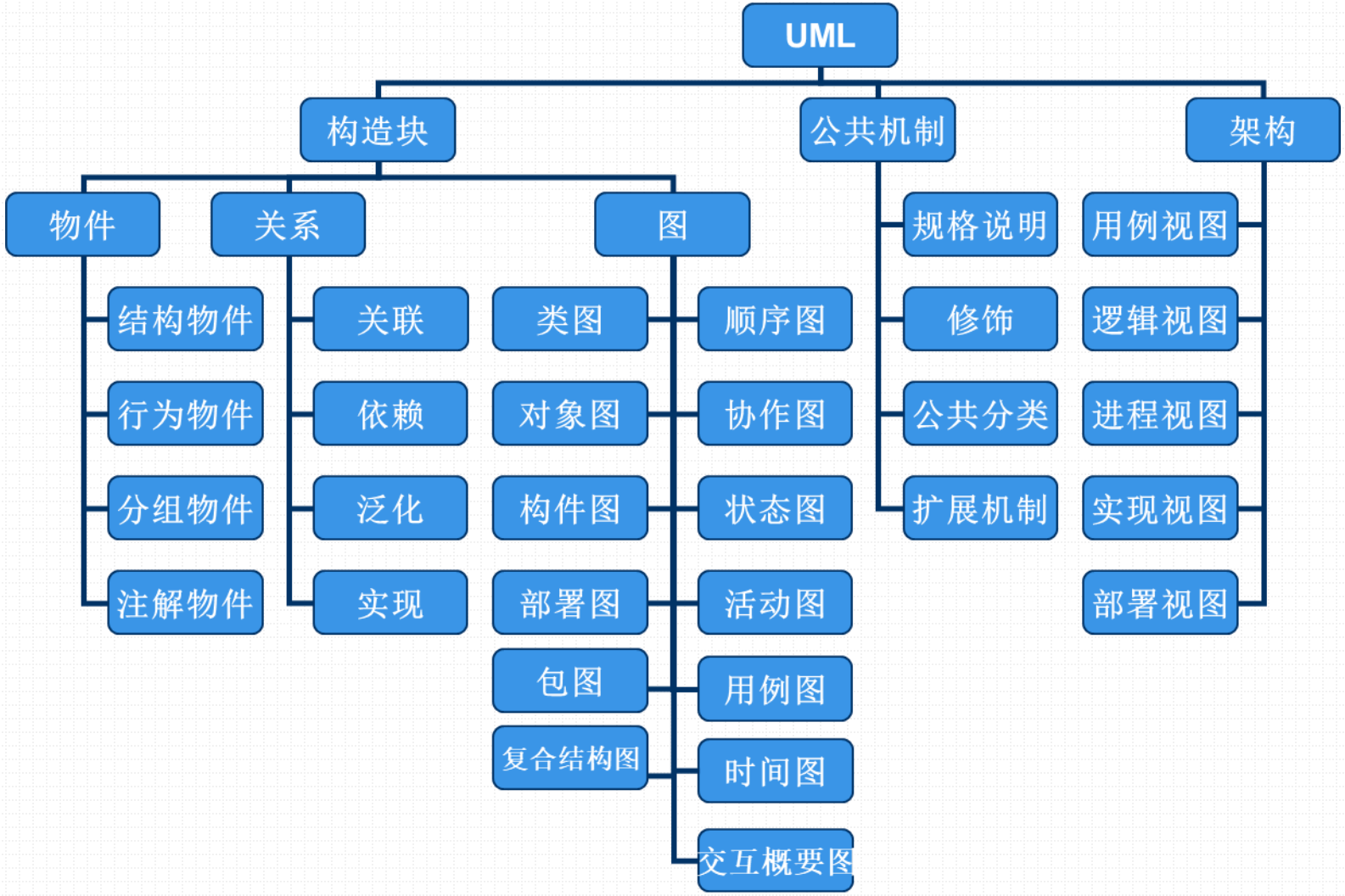
鹿传明

- 中睿信息技术有限公司创始人
- 厦门大学EMBA
- CSM/ EXIN Agile Scrum Master
- EXIN VeriSM数字化转型认证
- 中国特色的规模化敏捷开发过程IAP™创始人
- SAFe官方认证的企业级规模敏捷顾问（SPC）
- MSP 项目群管理专业人士
- 国际项目管理师PMP



专注于软件研发管理领域二十六年，擅长IT战略规划、数据治理、企业级敏捷实施、企业级项目管理等方面，曾服务腾讯、华为、软通动力、广电运通、中广核、中兴通讯、浩云科技、友邦资讯、招商物流、黄埔海关、中国外运、格力电器、中国银行、招商基金、oppo等等

UML结构



务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

类图

- 类图技术是OO方法的核心技术，应用非常广泛，其中类、对象以及它们之间的关系是最基本的建模元素。类模型和对象模型揭示了系统的结构。
- 分类可以有效地使复杂问题简化。建立类模型的过程，是把现实世界中与问题有关的各种对象及其相互关系进行抽象和分类的过程。

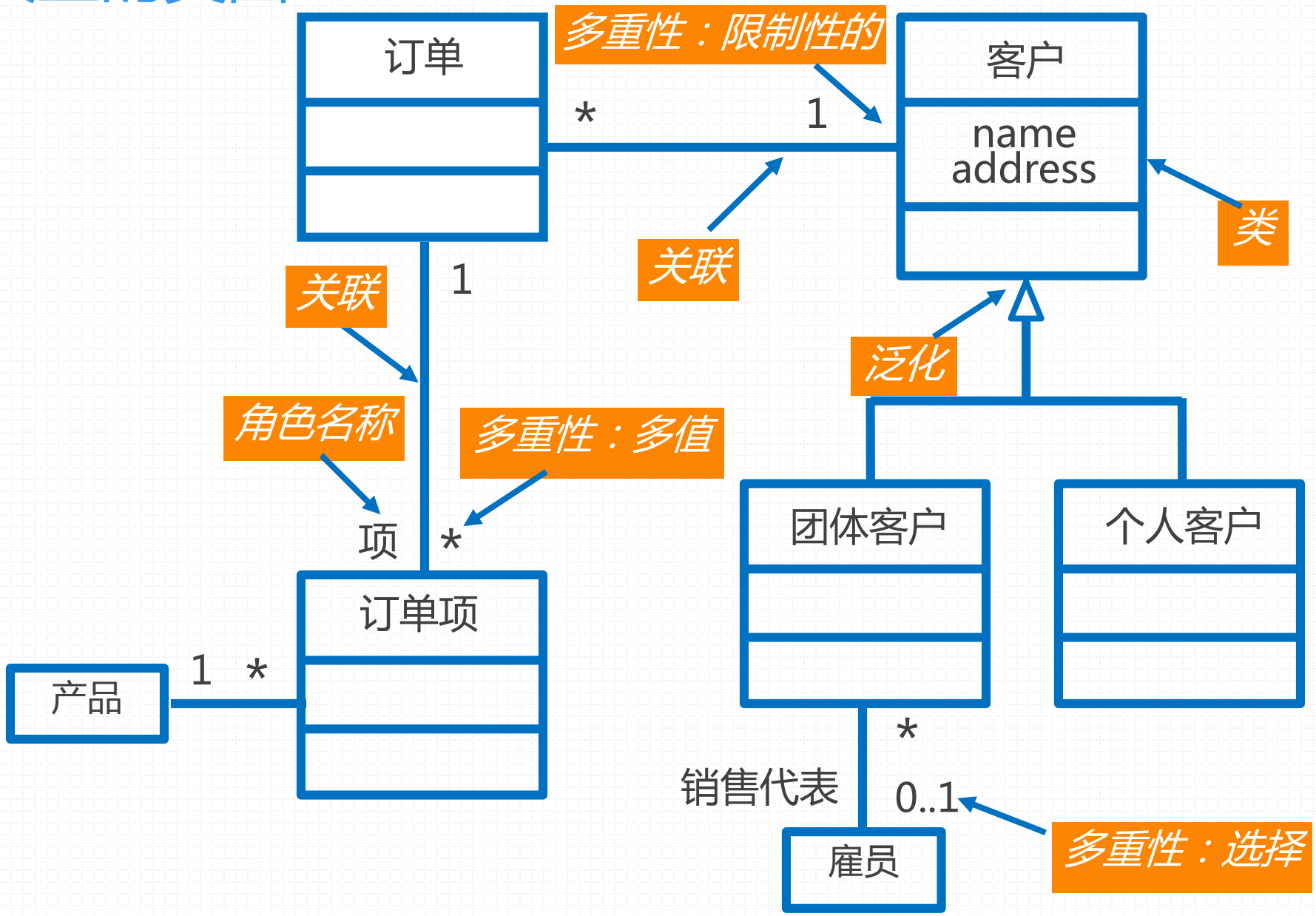
务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

三个层次的观点

- 画类图和理解类图时都应采用三个层次的观点。这些观点也适用于其它模型。
- 三个层次的观点不是UML的组成部分，但对建造模型或评价模型都非常有用，且都可应用于UML
- 概念层描述应用域中的概念，是对现实世界的直接描述，与实现它们的类有关，但与实现方案和实现语言无关。
- 说明层描述软件的接口，而不是软件的实现。一个类型描述一个接口，但可能有多种实现。
- 实现层从实现的角度定义类及其实现，揭示了软件实现体的构成情况。

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

一个典型的类图



务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

类图和对象图的基本要素

- 属性
- 操作
- 泛化
- 关联
- 授权
- 约束规则

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

在三个层面中的属性

- 在概念层，描述类具有的一些属性(客户对象的名字属性表示客户有名字)。
- 在说明层，规定类对象属性的值并给出设定这些值的方法(表示客户对象的名字并有一些设置名字的方法)。
- 在实现层，设置一个物理存储区来保存属性的值(也可称做一个实例变量或一个数据成员)。

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

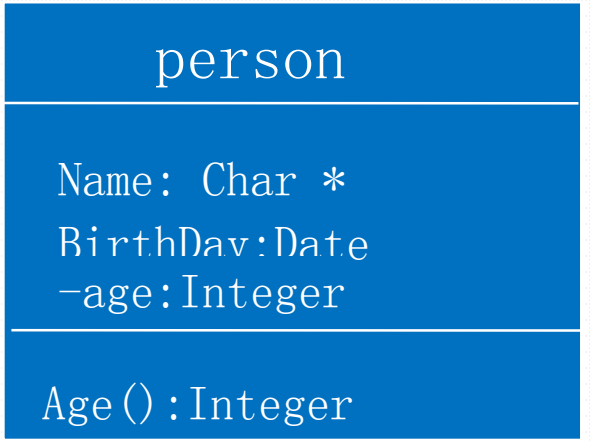
属性的语法

- UML规定其语法为：
  - 可见性 名称:类型 = 缺省值 {约束特性}
    - 可见性：表示该属性对类外的元素是否可见。常用的有公有、受保护和私有三种。
    - 名称：属性的名称，是一个字符串。
    - 类型：定义属性的种类(基本数据类型或用户自定义的类型)。
    - 缺省值：属性的初始值。
    - 约束特性：描述对属性的约束。

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

类的派生属性







{年龄=今天-生日}

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

操作

- 在三个层面中的操作
- 操作的语法
- 补充说明

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

在三个层面中的操作

- 在概念层，操作不是定义类的接口，而是指出类的主要职责，描述类的动态行为。
- 在说明层，主要给出重要的公有操作；然而有可能需要指明哪些属性是只读的或是不可修改的。
- 在实现层，给出操作的不同实现方法，有可能会显示一些私有的和受保护的操作。操作是施于对象的过程调用，而方法是过程体，是操作的一个具体实现 。

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

补充说明

- 两类操作：不改变类(对象)的可见状态的操作；例如，查询操作仅从类中取值，但不改变其可见状态。改变类(对象)的可见状态的操作称为修改操作。

- 查询操作可以按任意的顺序执行，但修改操作的顺序是重要的，如果不按照预定的顺序执行修改操作，有可能得到不同的结果。为了保证这两类操作相互独立，应避免从修改操作中返回值。

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

可见性

- 对“Public”、“Private”和“Protected”等三个可见性标识符的含义，各种语言都有它自己的规定。UML的定义是：
  - +(Public)：公有成员在程序的任何位置都是可见的，系统中的任何对象都可以使用它。
  - -(Private)：私有成员仅可以由定义它的类使用。
  - #(Protected)：受保护的成员仅可以由定义它的类和该类的子类中的对象使用。

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

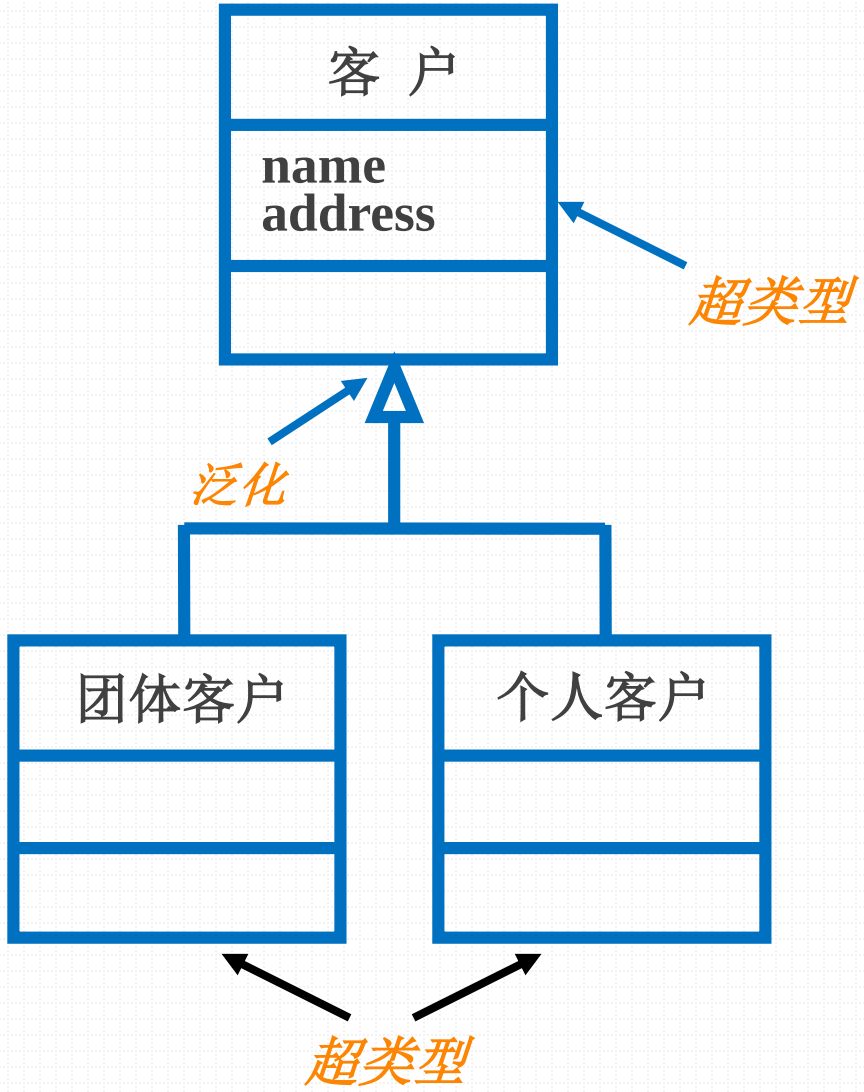
泛化

- 泛化的定义；
- 在三个层面中的泛化；
- 继承与泛化。

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

泛化的定义

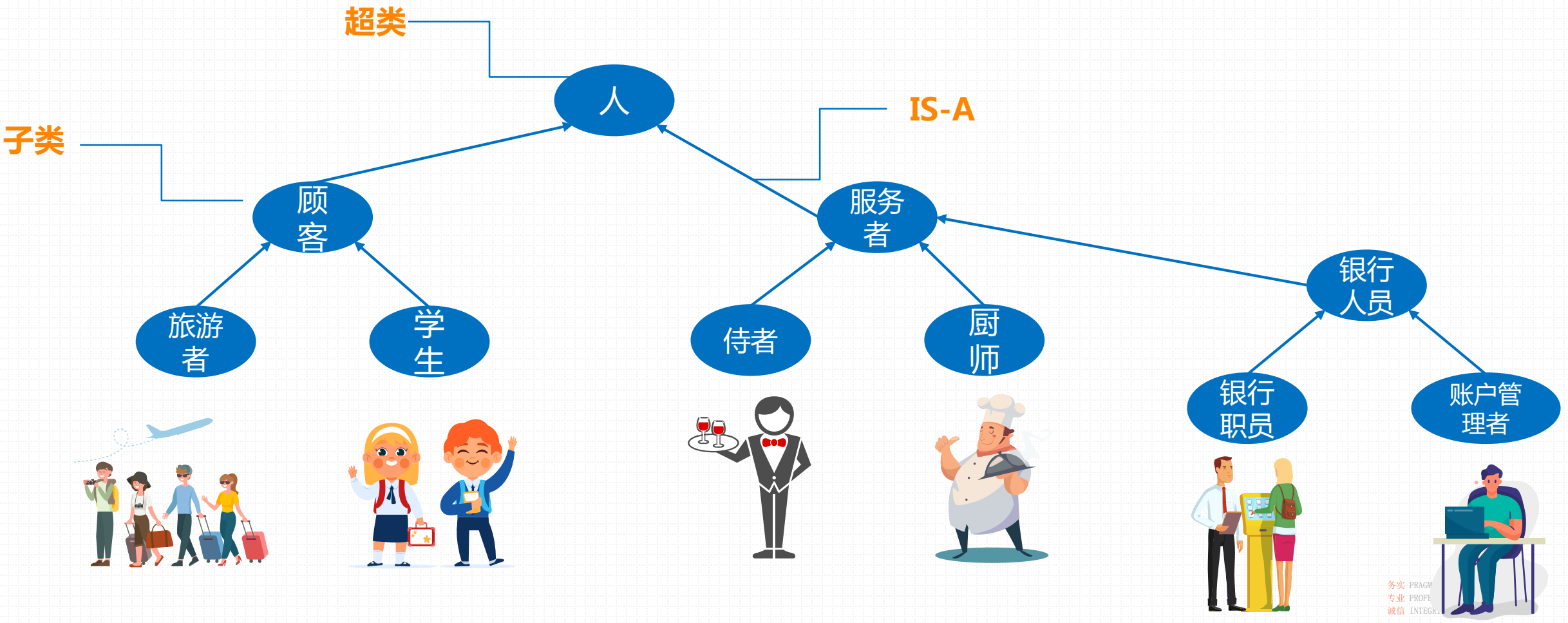
- 泛化关系(继承关系)定义类和包之间的一般元素和特殊元素之间的分类关系。例如，个人客户和团体客户都是客户，可以把他们的相似之处放到客户类(超类型)中，用个人客户和团体客户作为它的子类型。



务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

识别类之间的泛化

泛化举例



泛化的特征

- #1：Is-A 原则
  - 所有子类的成员必须是超类的成员
  - 这样必须说得通，子类“是一种”超类
- #2:100%原则
  - 超类的定义必须100%适用于所有子类
  - 子类必须100%符合超类的：
    - 属性
    - 关联

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

在三个层面中的泛化

- 在概念层，如果团体客户的所有实例都是客户的实例，那么团体客户类型是客户类型的一个子类型。
- 在说明层，泛化意味着子类型的接口必须包括超类型的接口中的每个元素，即两者必须保持一致。
- 在实现层，子类继承超类的所有属性和方法，并可覆盖继承来的方法。例如，凡适用于客户的代码，只要将将客户替换成团体客户，都适用于团体客户。

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

泛化（Generalization）

- 泛化(Generalization): 抽象化
- 特化(Specialization): 实例化
- 继承(Inheritance):
  - 泛化关系的一种实现机制
  - 并非所有的泛化关系都适合用继承关系实现

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

继承与泛化

- 继承是实现泛化的一种机制。在这种机制中，超类的任何一个子类都须具有其超类的所有行为：*不仅要求其操作界面在文法上一致，而且要求其行为在语义上一致。*
- 当子类中的一个操作重载其超类中相应的操作时，必须确保它提供与超类中的操作相同的服务（内容可以更多或更具体）。
- 如没有证明子类的行为是否与父类相同，就试图用继承来实现新类中的行为，当两者不一致时，会导致难以预测的错误。

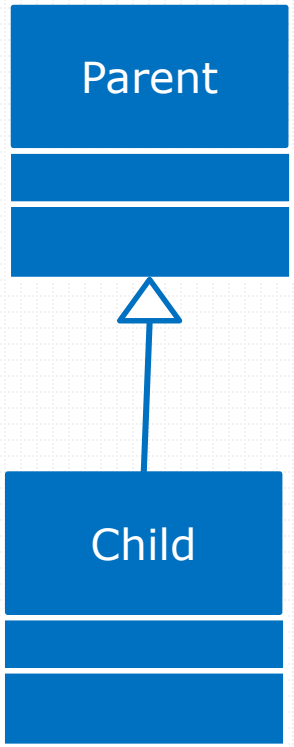
务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

泛化的麻烦

- 波动影响
  - 超类的改动影响到子类
- 多重继承
  - 命名冲突.....

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

从代码角度看泛化




```
Class Parent
{
public:
  Parent();
  ~Parent();
};

Class Child : public Parent
{
public:
  Child();
  ~Child();
};
```

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

识别类之间的泛化

- 根据领域知识
- 自上而下
- 自下而上
- 考虑领域内的复用

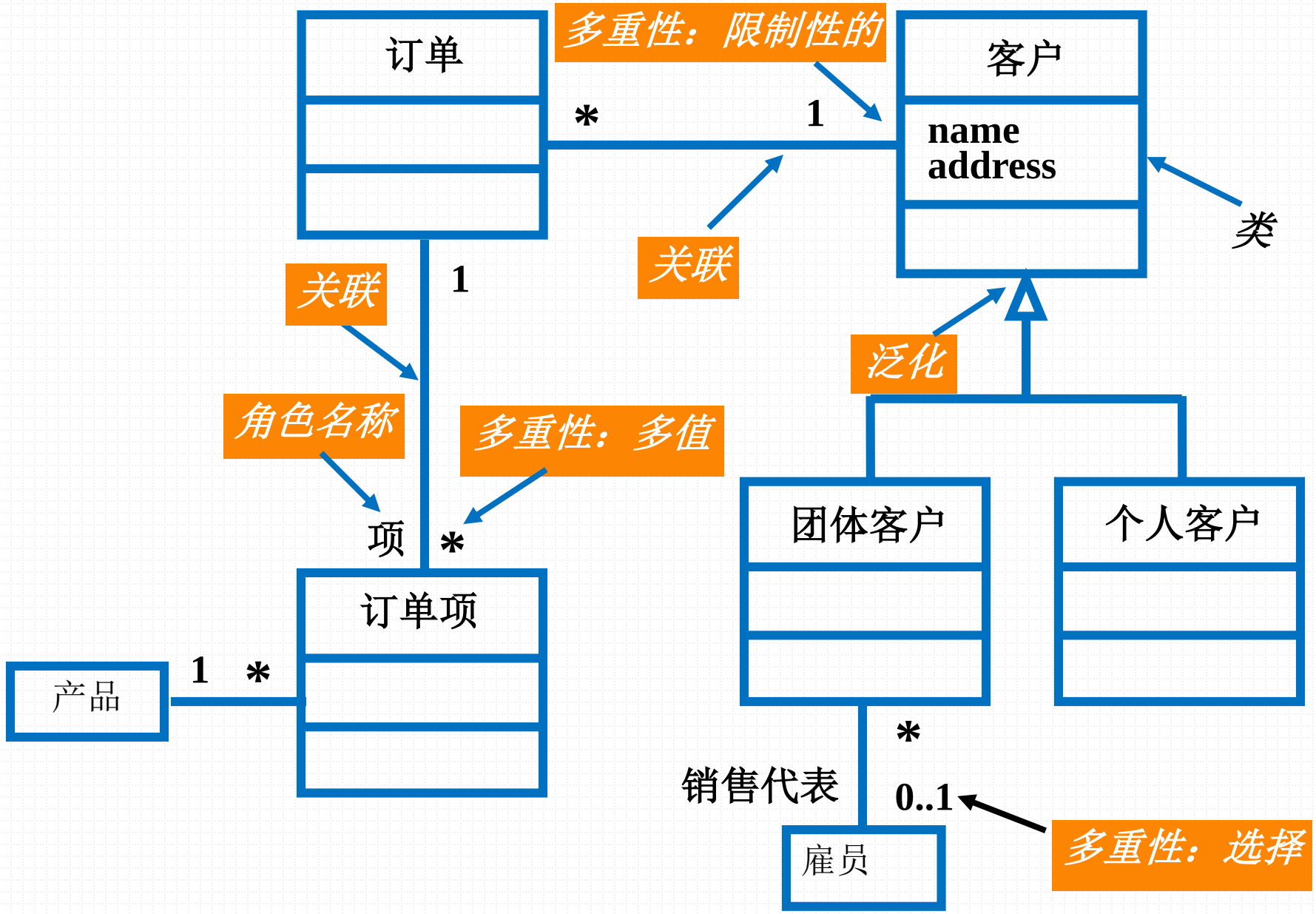
务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

关联

- 一个典型的类图
- 基本概念
- 角色及其命名
- 角色的多元性
- 三个层次中的关联
- 导航特性

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

一个典型的类图



务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

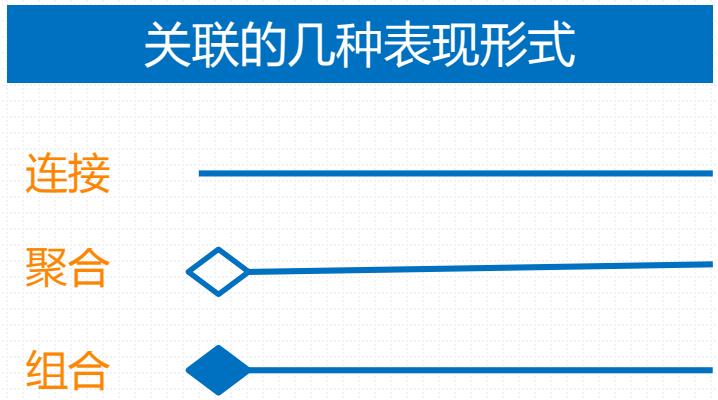
关联的含义

- 关联的含义
  - 通过属性来表示一个对象对另一个对象的静态依赖关系（区别于通过操作）

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

关联的表示

- 关联的表示：
  - 用一条无向线段表示，是一种双向关系。例如客户和订单的关联：从客户看，订单是他提交的；从订单看，它有一个客户。
  - 用一条有向线段表示，是一种单向关系
- 关联的命名：可以用动词词组或名词命名。但只要这个关联的含义明确，则可省略这个名字。



务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

类之间的关联

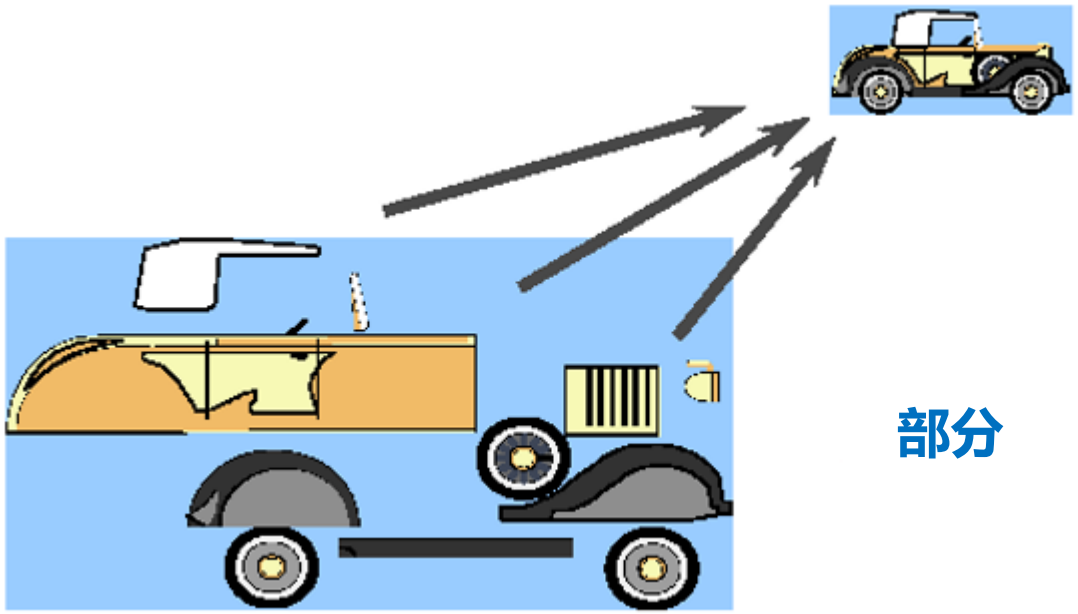
- 连接
  - 最弱的关联，只是表示两个类对象之间有导航关系



务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

类之间的关联

- 聚合
  - 具有“has a”语义—对象A是对象B的一个组成部分



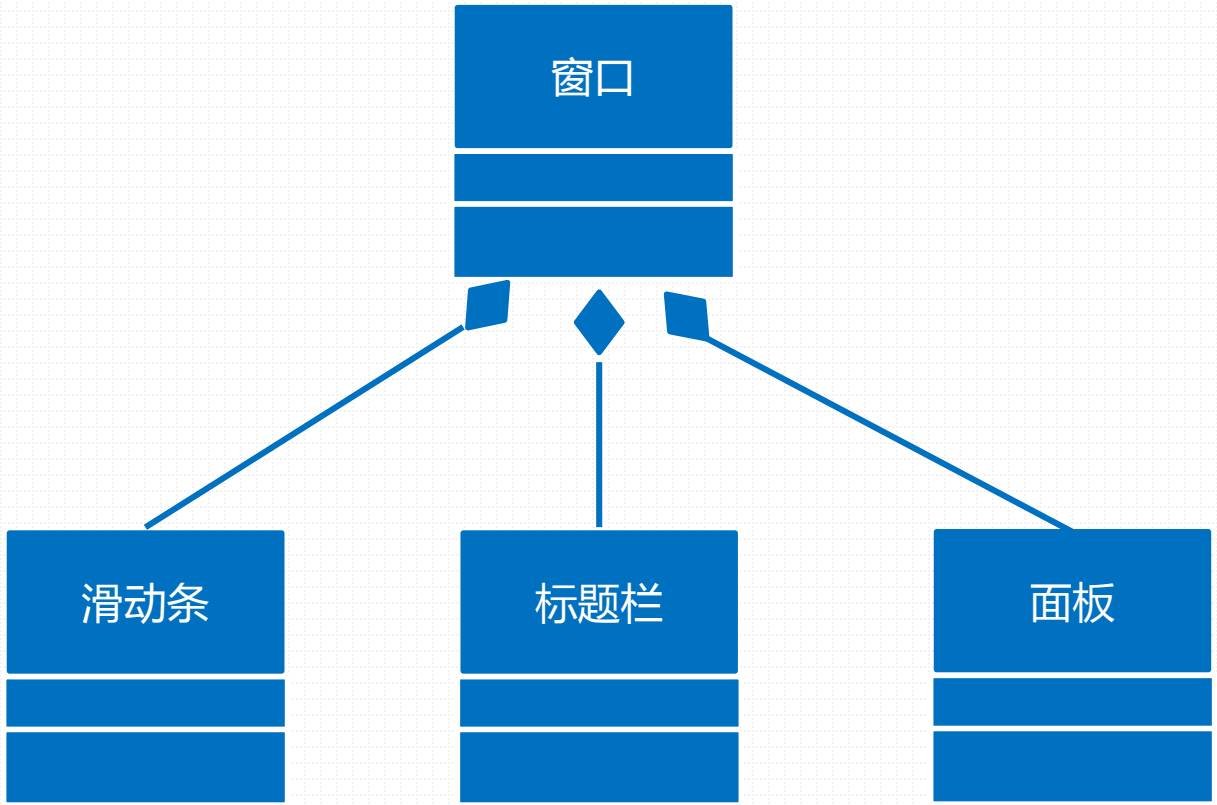
整体

部分

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

类之间的关联

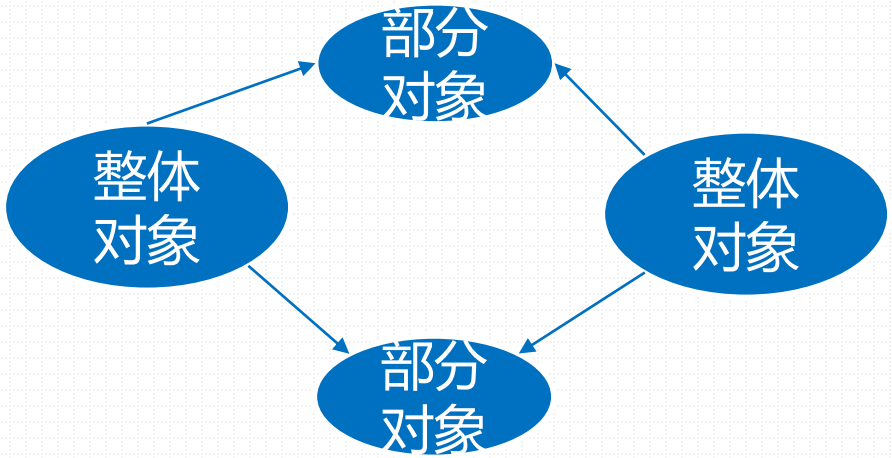
- 组合
  - 强语义的聚合—整体部分消失，部分对象也消失



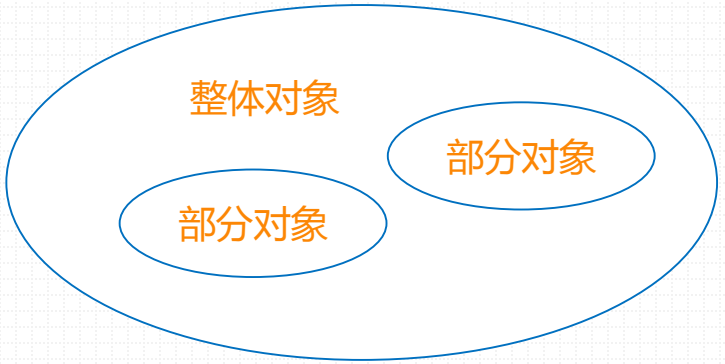
务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

类之间的关联

- 聚合vs.组合（UML观点）



聚合



组合

例：公司和雇员

例：订单和订单项

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

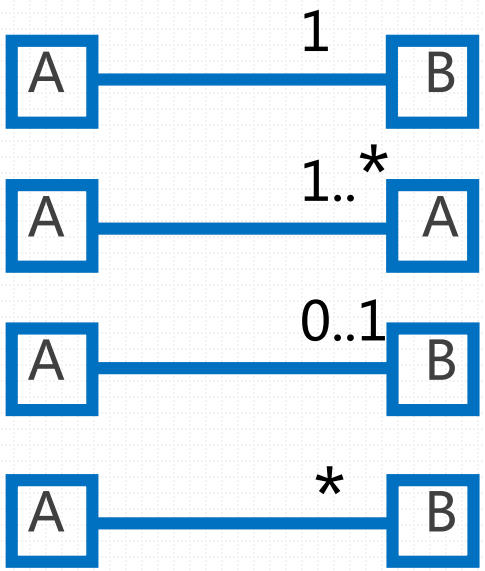
角色及其命名

- 关联的两端与类之间(或与类的实例之间)的接口表示该类(或该类的实体)在这个关联中的行为，称之为角色。
- 每个关联有两个角色。例如，对于客户和订单之间的关联是：客户和订单。
- 可将引出角色的类称作源，将引入角色的类称作目标。例如，从订单到客户的角色的源是订单，目标是客户。
- 为了明确对象在关联中的角色，可以为角色命名。例如从订单到订单项方向上的角色可以命名为项。
- 如果在关联上没有标出角色名，则隐含地用该角色的目标类的名称作为它的名称。例如，从订单到客户的角色应叫做客户。

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

角色的多元性

- 角色可以具有多元性(一个角色可以有多个对象来扮演)。例如，每一个客户对象可以有零个或多个订单对象。
- 多元性的表示。1 表示 1..1 ；*代表零到无穷；0..1是选择符，表示没有或仅有1个；一个数；一个范围；数字和范围不连续的组合。



表示 A 和一个 B 关联

表示 A 和一个或多个 B 关联

表示 A 和零个或一个 B 关联

表示 A 和零个、一个或多个 B 关联。

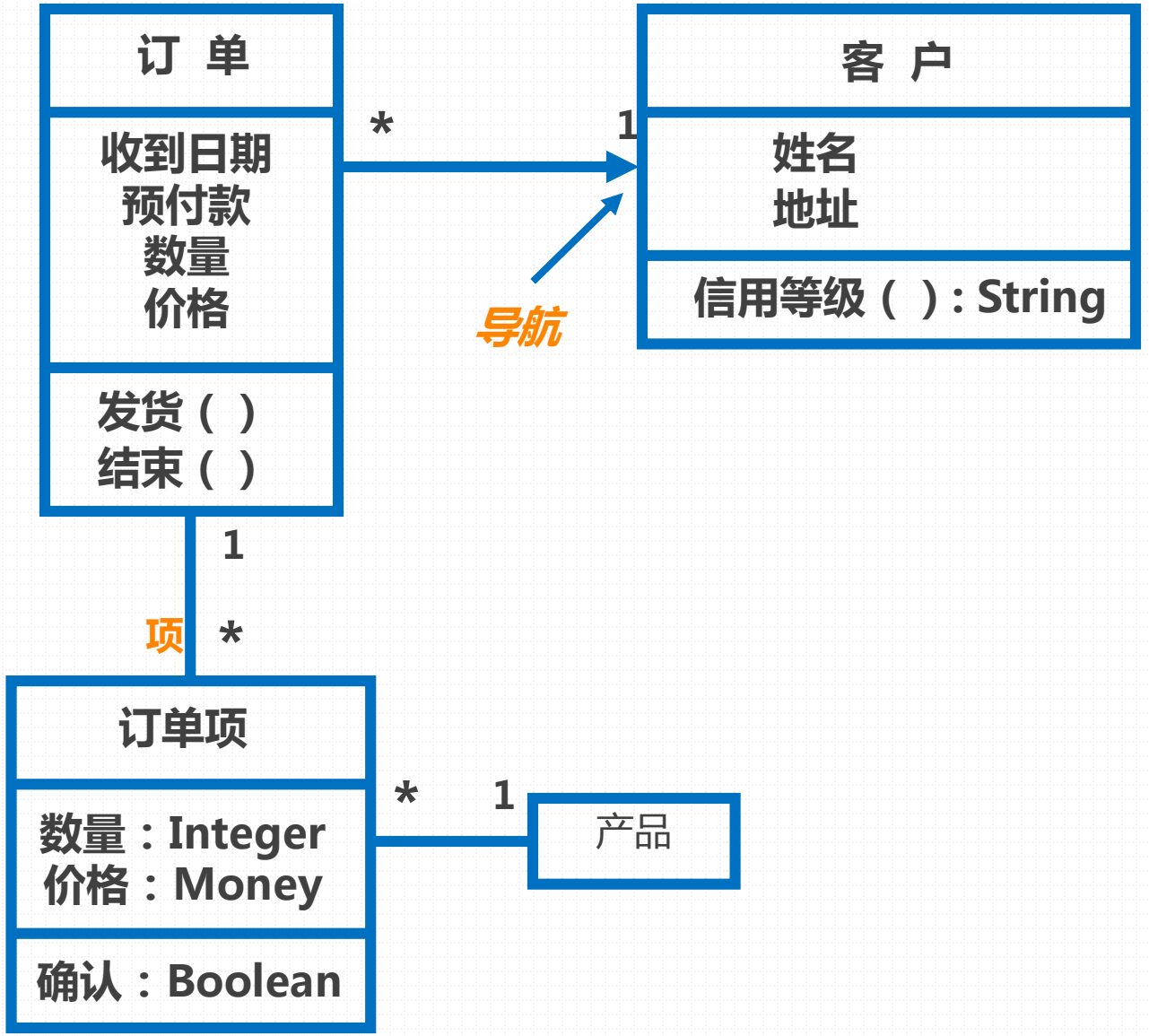
务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

三个层面中，关联性意味着什么？

- 概念层：在应用域中两类对象之间存在的某种关系
- 说明层：表示一种职责(Responsibility),一方向另一方请求或发送某种消息或服务，但并不涉及实现这种服务的具体方法(methods)
  - 例，定单的一个职责是记住和报告它的所有者是谁
- 实现层：意味着(用指针来)建立某种关联性。

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

导航特性



务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

导航特性(续)

- 箭头表示导航特性。如果只在一个方向上有导航表示，称作单向关联。如果在两个方向上都有导航表示，称作双向关联。如果不带箭头，表示未知或尚未确定。
- 单向关联时，说明模型中的订单指出它是由哪个客户发出的；实现模型中的订单包含一个指向客户的指针。
- 对双向关联的限制是两个角色必须互逆。

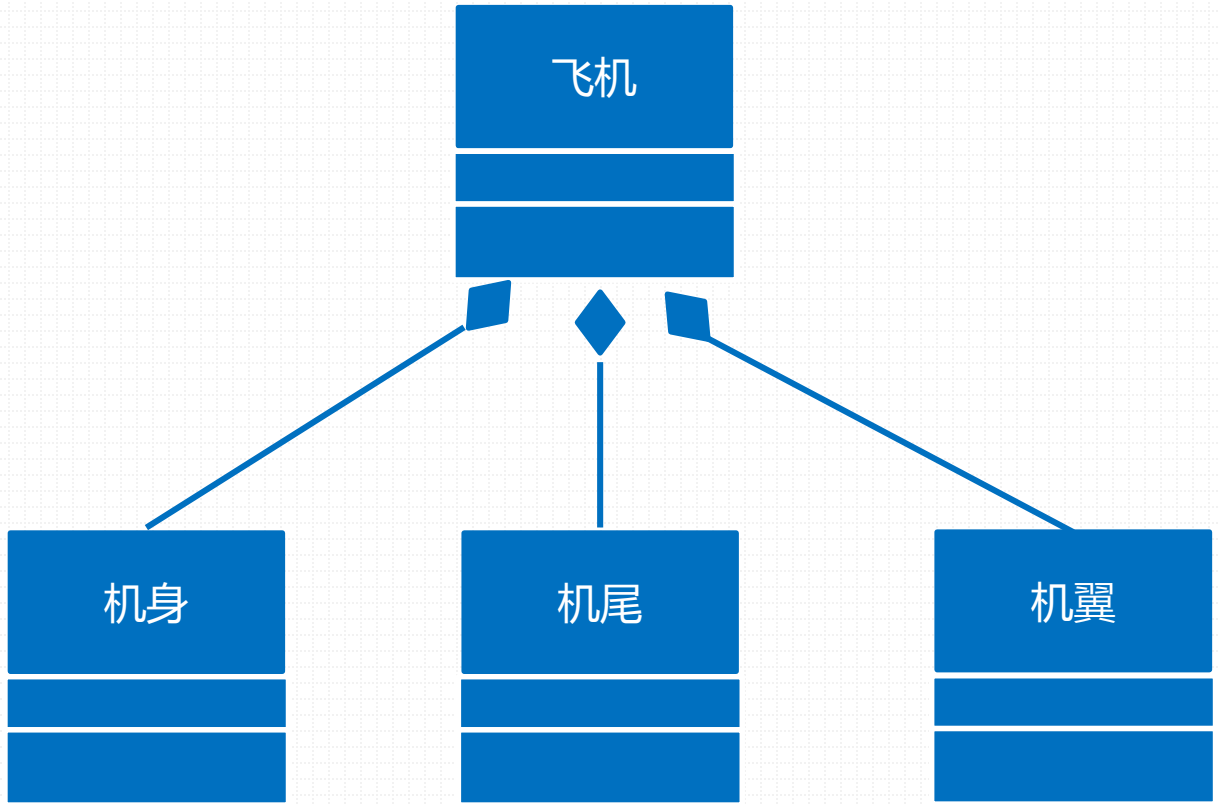
务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

识别类之间的聚合

- 思路
  - 物理上的整体事物和它的组成部分
  - 组织机构和它的下级组织
  - 团队（组织）和成员
  - 空间上的包容
  - 抽象事物的整体和部分
  - 具体事物和它的某个抽象方面

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

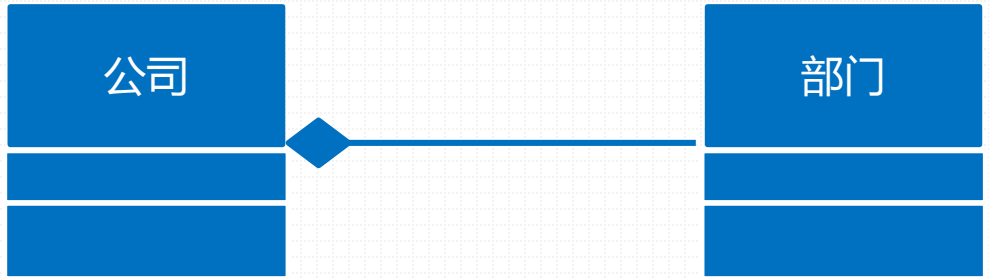
识别类之间的聚合（组合）



物理上的整体事物和它的组成部分

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

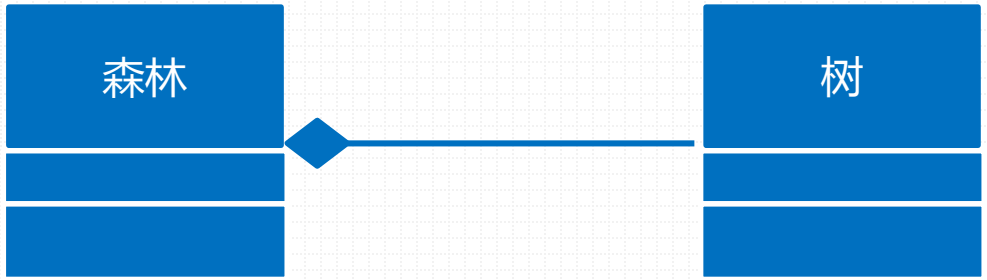
识别类之间的聚合（组合）



组织机构和它的下级组织

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

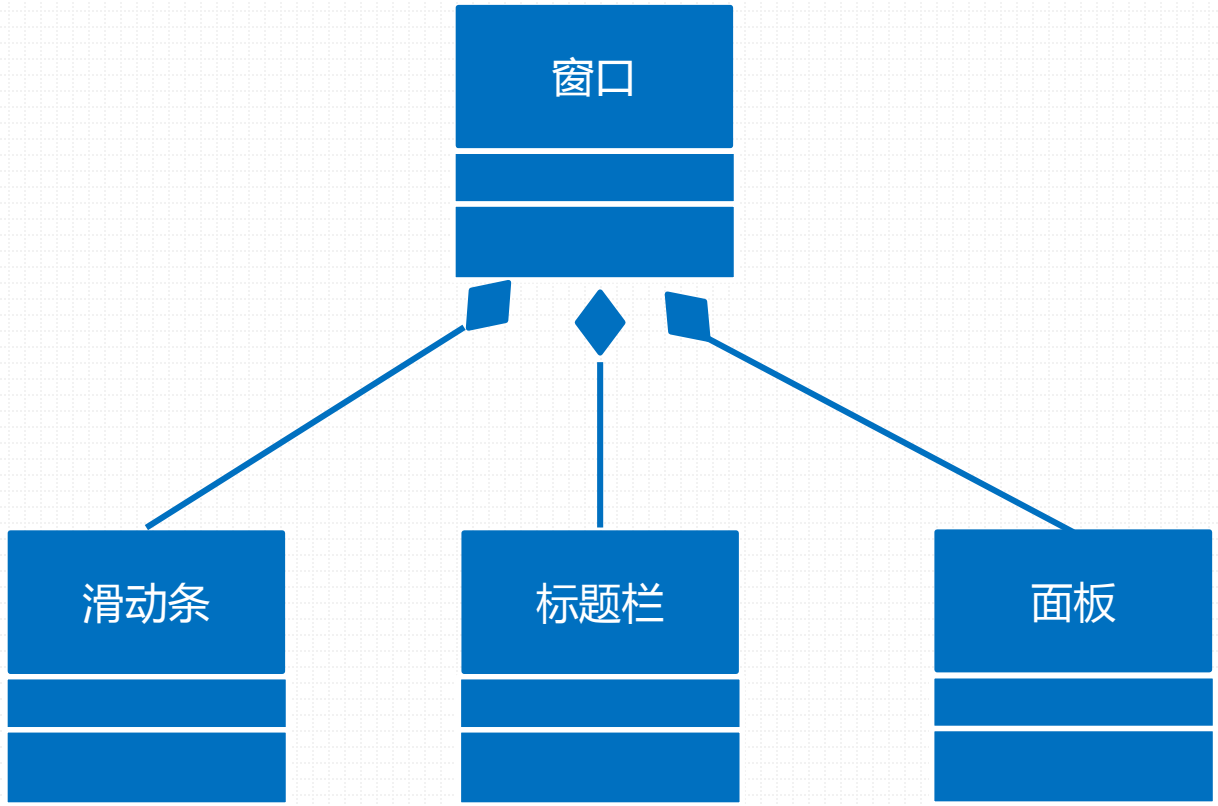
识别类之间的聚合（组合）



团队（组织）和成员

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

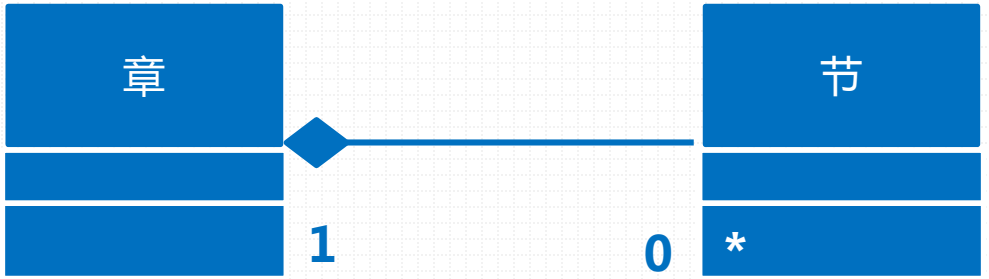
识别类之间的聚合（组合）



空间上的包容

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

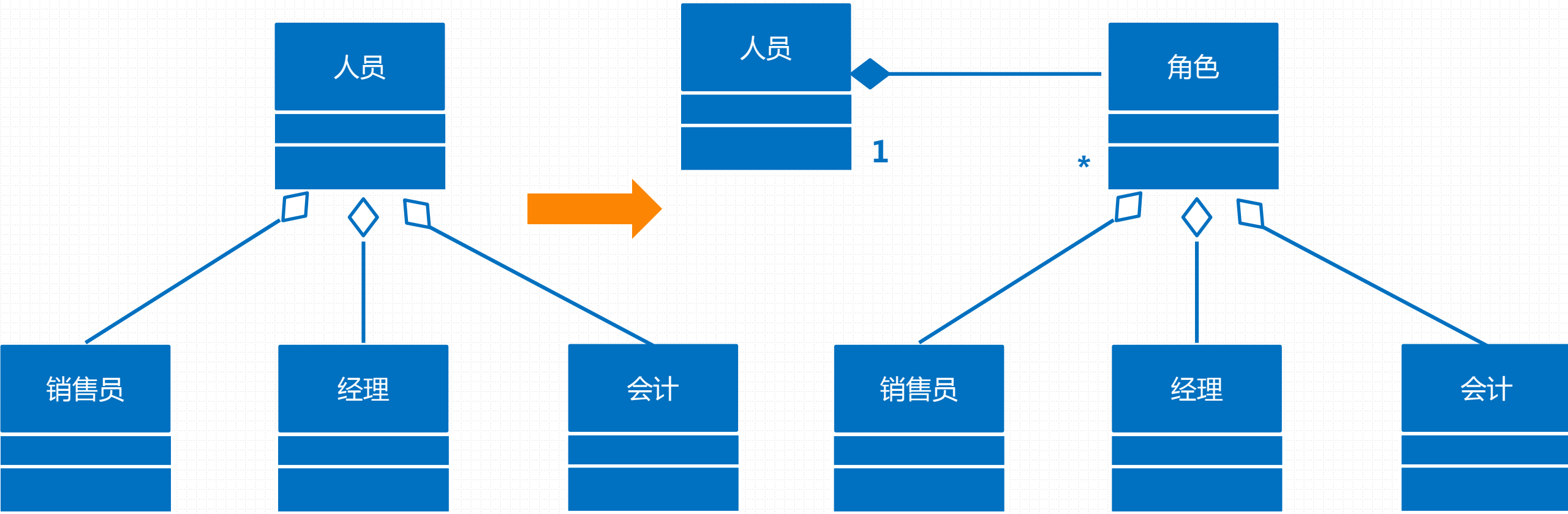
识别类之间的聚合（组合）



抽象事物的整体和部分

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

识别类之间的聚合（组合）



具体事物和它的某个抽象方面

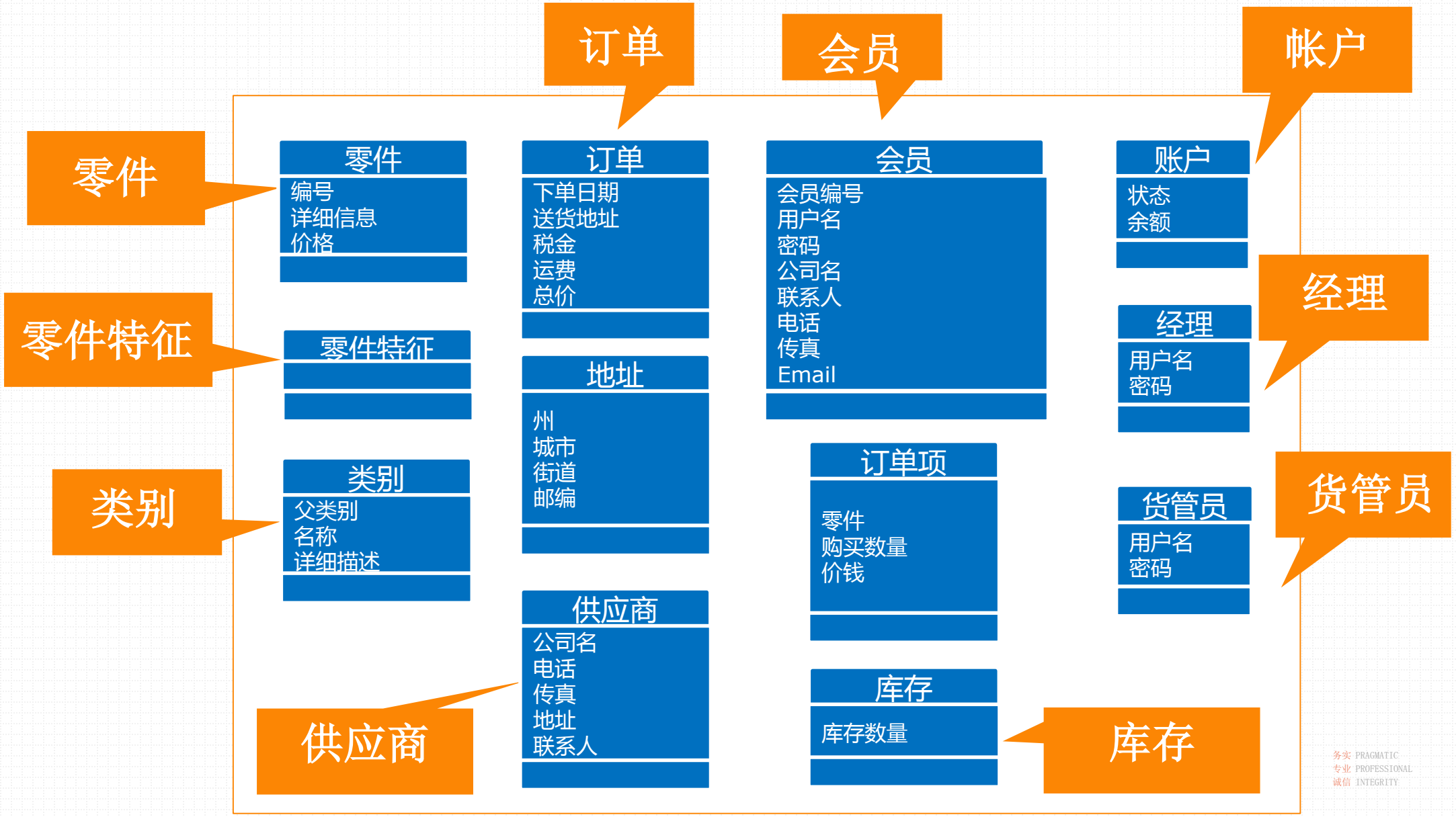
务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

审查聚合（组合）

- 是否属于问题域和系统责任的范围
  - 企业业务管理系统：公司职员—他们的家庭
- 部分对象的属性有多少
- 是否有明显的聚合关系

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

画出类图之间的关系



务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

约束规则

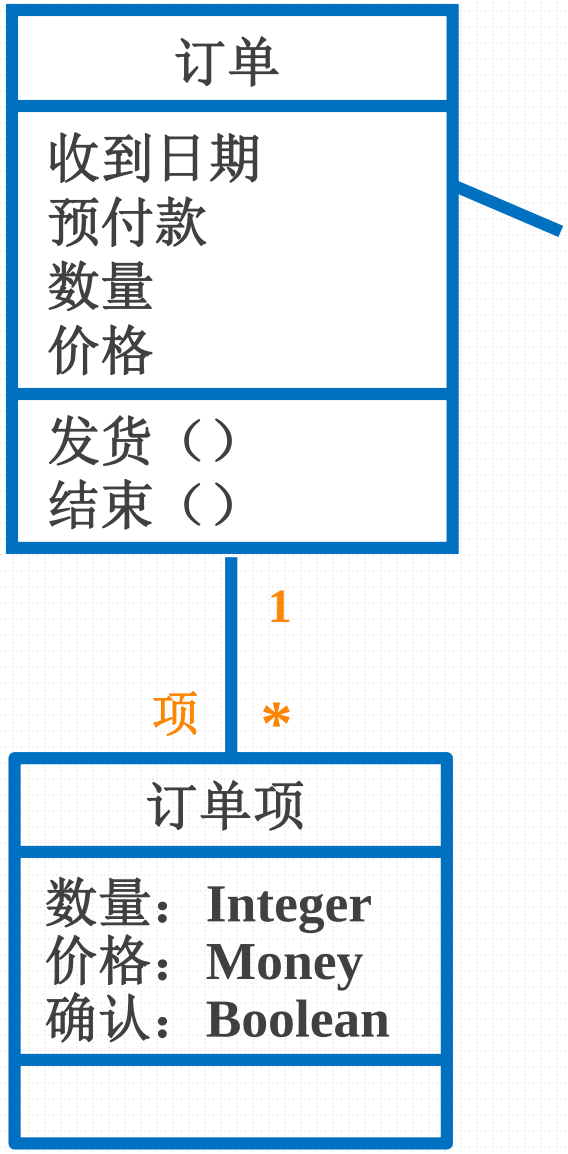
- 约束规则的语法
- 约束规则实例

约束规则的语法

- 在画类图的过程中，关联、属性和操作等基本要素都要为模型加注约束条件。
- 约束规则的语法：将约束条件放在括号{ }中，用自然语言或其他常见的设计语言来描述，其描述要简洁准确。
- 在理想的情况下，在所使用的程序设计语言中，规则应该作为断言来实现，并在调试代码时调用它。

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

约束规则实例




```
{ if
      定货单.客户.信用等级==“低”
  then
      定货单.预付款 必须是“真”
}
```

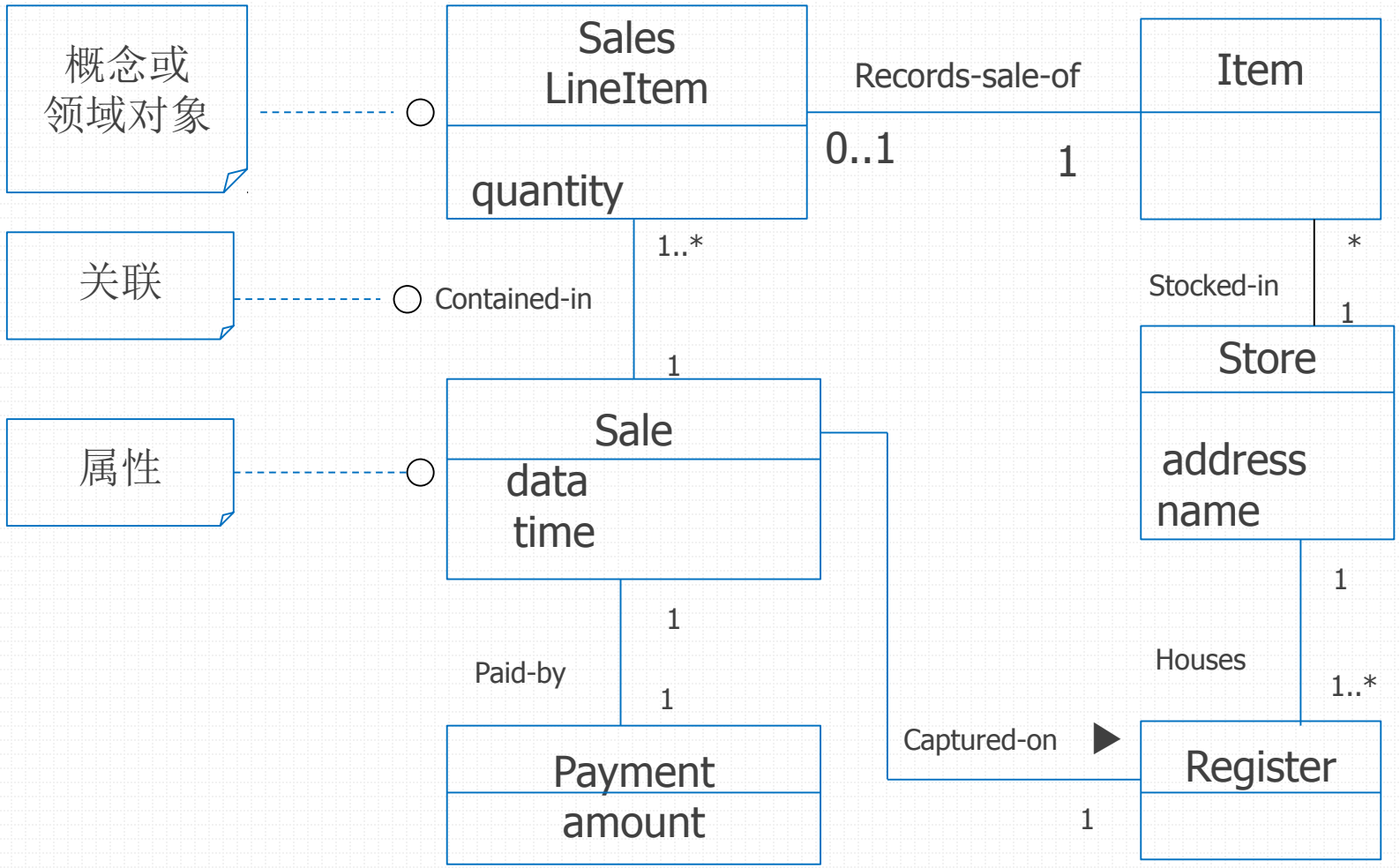
务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

领域模型

- 是对领域内的概念类或现实世界中对象的可视化表示
- 也称概念模型、领域对象模型和分析对象模型
- 是对现实世界概念类的表示，而非软件对象的表示，该术语并不是指用来描述软件类、软件构架类领域层或有职责软件对象的一组图
- UML，领域模型被描述为一组没有定义操作的类图，它提供了概念透视图
  - 领域对象或概念类
  - 概念类之间的关联
  - 概念类的属性

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

领域模型



务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

领域模型准则

- 如何创建领域模型
  - 寻找概念类
  - 将其绘制为UML类图中的类
  - 添加关联和属性
- 如何找到概念类
  - 重用和修改现有的模型
  - 使用分类列表
  - 确定名词短语

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

主成功场景

- ***顾客***携带所购***商品***或***服务***到***POS机付款处***进行购买交易
- ***收银员***开始一次新的***销售***交易
- ***收银员***输入***商品标识***
- 系统逐条记录出售的***商品***，并显示该商品***项目的描述***、***价格***和***累计额***。价格通过一组价格规则来计算
- 收银员重复3 - 4步，直到结束
- 系统显示总额和所计算的***税金***
- 收银员告知顾客总额，并提请***付款***
- 顾客支付，系统处理支付
- 系统记录完整的***销售***信息，并将销售和支付信息发送到外部的***帐务系统***（进行帐务处理和***提成***）和***库存系统***（更新库存）
- 系统打印***票据***
- 顾客携带商品和票据（如果有）离开
- 扩展
- ...
- 现金支付
  - 收银员输入收取的现金额
  - 系统显示找零金额，并弹出现金抽屉
  - 收银员放入收取的现金，并给顾客找零
  - 系统记录该次现金支付

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

业务建模准则

- 敏捷建模 - 绘制类图的草图
- 敏捷建模 - 是否要实用工具维护模型
- 报表对象 - 模型中是否要包含“票据”
  - 一般来说，在领域模型中显示其他信息的报表并没有意义，因为其所有的信息都是源于或复制于其他信息源的
- 像地图绘制者一样思考；使用领域术语
  - 使用地域中现有的名称
  - 排除无关或超出范围的特性
  - 不要凭空增加事物
- 如何对非现实世界建模
  - 吸取领域专家所使用的核心词汇和概念

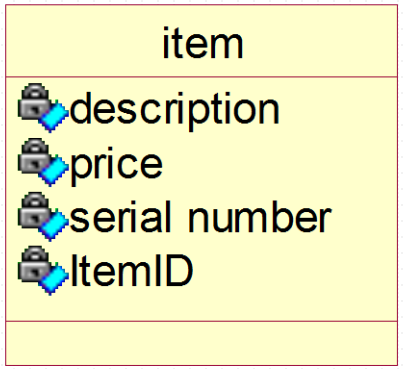
务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

何时使用”描述”类建模

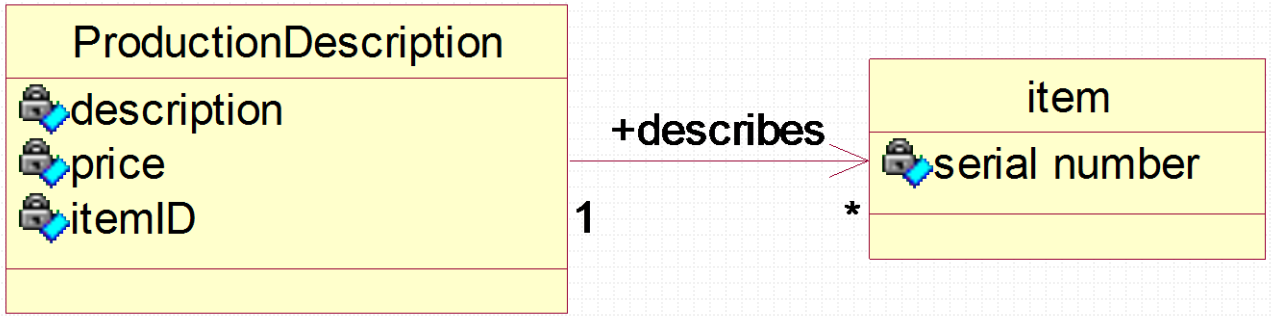
- 描述类包含描述其他事物的信息
  - 产品描述（ProductDescription）
- 在以下情况下需要增加描述类
  - 需要有关商品或服务的描述，独立于任何商品或服务的现有实例
  - 删除其所有描述事物的实例后，导致信息丢失，而这些信息是需要维护的，但是被错误地与所删除的事物关联起来
  - 减少冗余或重复信息

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

关于其他事物的描述



较差



较好

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

表9 - 1 概念类分类列表

概念类的类别	准则	示例
业务交易	十分关键（涉及金钱），所以为起点	Sale, Payment, Reservation
交易项目	交易中通常会涉及项目，所以置为二	SalesLineItem
与交易或交易项目相关的产品服务	（产品或服务）是交易的对象。所以置为第三	Flight, Seat, Meal
交易记录在何处	重要	Register, Ledger FlightMainifest
与交易相关的人或组织的角色；用例的参与者	我们通常要知道交易所涉及的各方	Cashier, Customer, Store MonopolyPlayer Passenger, Airline
交易的地点；服务的地点		Store Airport, Plane, Seat
重要事件，通常包括我们需要记录的时间或地点		Sale, Payment MonopolyGame Flight
物理对象	特别是在创建设备控制软件或进行仿真时非常有用	Item, Register, Board, Piece, airplane

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

表9 - 1 概念类分类列表

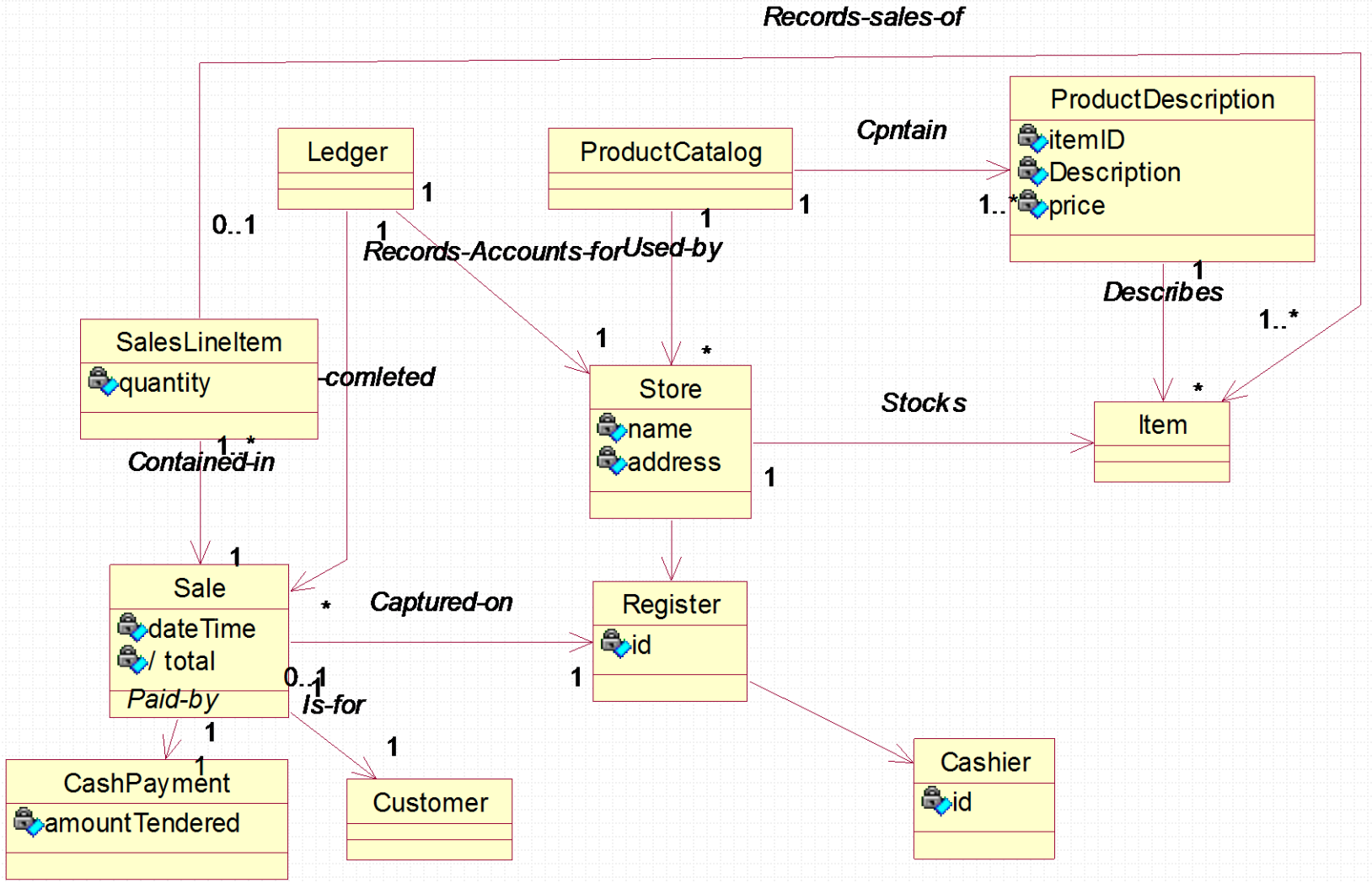
概念类的类别	准则	示例
事物的描述		ProductDescription FlightDescription
类别	描述通常有类别	ProductCatalog
事务（物理或信息）的容器		FlightCat Store, Bin Board Airplane
容器中的事物		Item Square Passenger
其他协作的系统		CreiditAuthorizationSystem AirTrafficControl
金融、工作、合约、法律材料记录		Receipt, Ledger MaintenanceLog
金融手段		Cash Check LineOfCredit TicketCredit
执行工作所需的进度表、手册、文档等		DailyPriceChangeList RepairSchedule

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

类型	示例
A是与交易B相关联的交易	CashPayment-Sale Cancelation-Reservation
A是交易B中的一个项目	SaleLineItem-Sale
A是交易B 的产品或服务	Item-SalesLineItem(Sale) Flight-Reservation
A是与交易B相关的角色	Customer-Payment Passenger-Payment
A是B的物理或逻辑部分	Drawer-Register Square-Board Seat-Airplane
A被逻辑地或物理地包含在B中	Register-Store, Item-Shelf Square-Board Passenger-Airplane
A是B的描述	ProductDescription-Item FlightDescription-Flight
A在B中被感知/记日记/记录/生成报表/捕获	Sale-Register Piece-Square Reservation-Flightmainfest
A是B的成员	Cashier-Store Player-MonopolyGame pilot-Airline
A是B的组织化单元	Department-Store Maintenance-Airline
A使用/管理/拥有B	Cashier-Register Player-Price Pilot-Airplane
A与B相邻	SalesLineItems-SalesLineItem Square-Square City-City

务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

NextGen Pos部分领域模型



务实 PRAGMATIC
专业 PROFESSIONAL
诚信 INTEGRITY

Thanks

中睿信息技术有限公司 广州·深圳

广州：广州市天河区翰景路1号金星大厦18层中睿信息

Tel：020-38931912 Fax：020-38931912

深圳：深圳市福田区车公庙有色大厦509-510室

Tel：0755-25855012