

# 面向对象的分析 (Object-oriented analysis, OOA)

## 1、OOA 的基本任务：

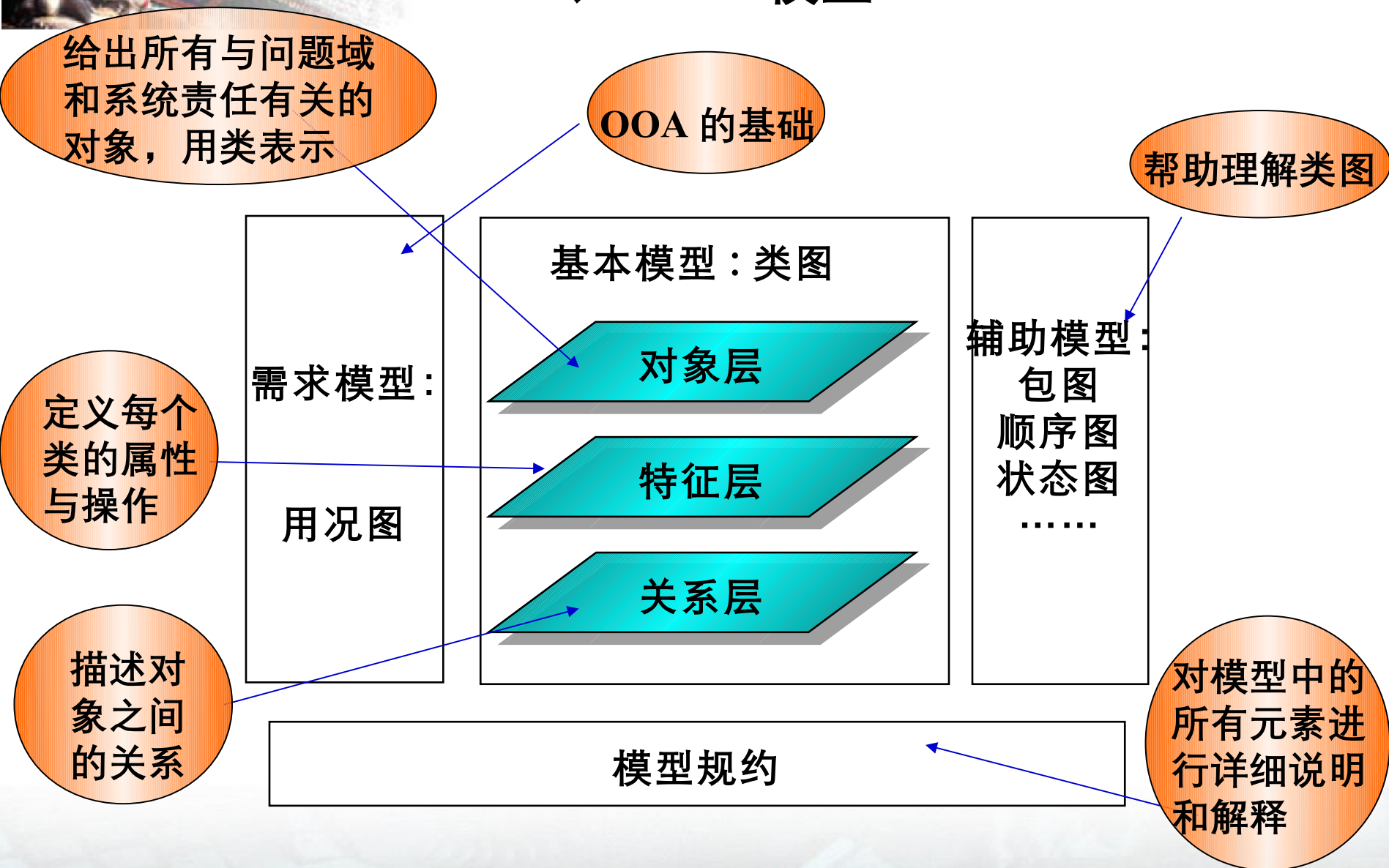
运用面向对象方法，对问题域（被开发系统的应用领域）和系统责任（所开发系统应具备的职能）进行分析和理解，对其中的事物和它们之间的关系产生正确的认识，找出描述问题域和系统责任所需的类和对象，定义这些类和对象的属性和操作，以及它们之间所形成的各种关系。最终目的是产生一个符合用户需求，并能够直接反映问题域和系统责任的 OOA 模型及其规约。

—— 面向对象的系统分析（第二版），邵维忠等编著，  
清华大学出版社，2006.12.



北京大学

## 2、OOA 模型



### 3、OOA 过程

#### 定义 use case（辅助模型，可选）

用 use case 对用户需求进行规范化描述。

#### 建立类图（基本模型）

- \* 发现对象、定义对象类
- \* 识别对象的内部特征
- \* 识别对象的外部关系

#### 原型开发

结合其他活动反复进行

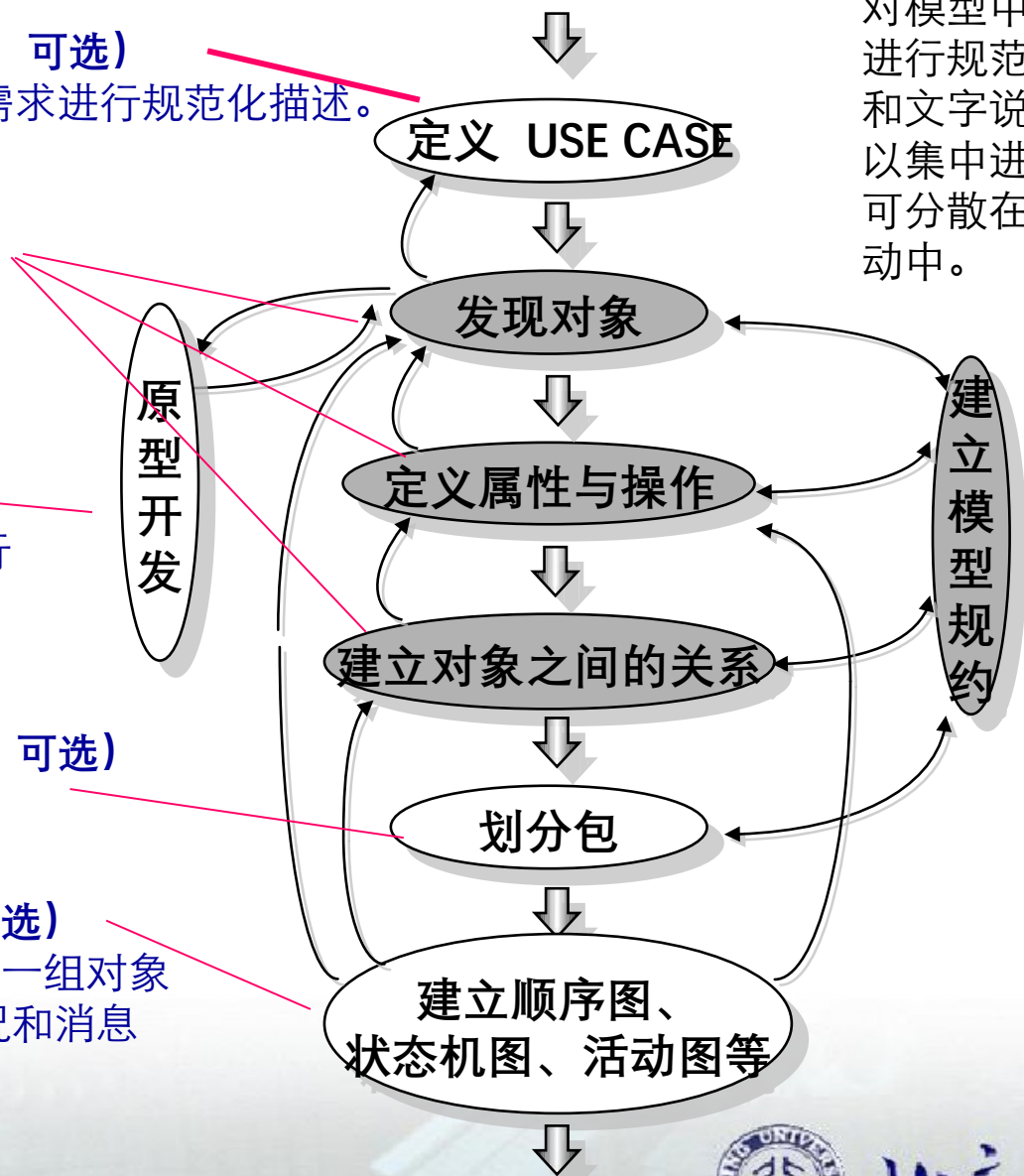
#### 划分包，建立包图（辅助模型，可选）

#### 建立交互图（辅助模型，可选）

对照 use case，描述一组对象进行协作时的交互情况和消息的时序关系。

#### 建立模型规约

对模型中的成分进行规范的定义和文字说明。可以集中进行，也可分散在各个活动中。





# 识别类、属性和操作



北京大学

# 一、识别类



## 1、研究问题域和用户需求

### (1) 研究用户需求，明确系统责任

阅读：阅读一切与用户需求有关的书面材料

交流：与用户交流，澄清疑点，

纠正用户不切实的要求或不确切的表达

调查：到现场调查（只限于澄清需求）

记录、整理：产生一份符合工程规范、确切表达系统责任的需求文档

### (2) 研究问题域

亲临现场调查，掌握第一手资料

听取问题域专家的见解

阅读与问题域有关材料

借鉴相同或类似问题域已有的系统开发经验及文档



北京大学





### (3) 确定系统边界

就是划出被开发的系统和该系统打交道的人或物之间的明确界限，并确定它们之间的接口。

在系统边界之内，是系统本身所包含的对象。在系统边界以外，是系统外部的活动者。主要是人、设备和外系统三种外部活动者。





## 2、策略与启发

### (1) 考虑问题域：



可以启发分析员发现对象的因素包括：人员、组织、物品、设备、抽象事物、事件、文件及结构等。

- 人员：（a）需要由系统保存和管理其信息的人员，如户籍管理系统中的每个居民；（b）应该在系统中完成某些功能，提供某些服务的人员，如户籍管理员。符合上述情况之一者，应考虑用相应的人员对象来描述。





- **组织**：在系统中发挥一定作用的组织结构。如工作班组等。
- **物品**：需要由系统管理的各种物品。如经营的商品等。
- **设备**：在系统中动态地运行、由系统进行监控或供系统使用的各种设备、仪表、机器及运输工具等。
- **抽象事物**：指没有具体的物理形态，但对用户的业务具有实际意义的逻辑上的事物。
- **事件**：指那些需要由系统长期记忆的事件。
- **文件**：泛指在人类日常的管理和业务活动中使用的各种各样的表格、档案、证件和票据等文件。
- **结构**：通过考虑结构可以得到一种启发——从已经发现的对象联想到其他更多的对象







## (2) 考虑系统边界:

考虑系统边界，可以启发分析员发现一些与系统边界以外的参与者进行交互，并且处理系统对外接口的对象。



**人员：**作为系统以外的参与者与系统进行直接交互的各类人员，如系统的操作员、直接使用系统的用户等。

**设备：**作为系统以外的参与者与系统相连并交换信息的设备。

**外系统：**与系统相连并交换信息的其他系统。

## (3) 考虑系统责任:

检查每一项功能需求是否有相应的对象提供，发现新的对象



北京大学



### 3、审查与筛选

#### (1) 舍弃无用的对象

##### 通过属性判断：

是否通过属性记录了一些对参与者或对系统的其他对象有用的信息？（即这个对象所对应的事物，是否有些信息需要在系统中进行保存和处理？）

##### 通过操作判断：

是否通过操作提供了某些有用的功能？（即这个对象所对应的事物，是否有某些行为需要在系统中模拟，并在系统中发挥一份作用？）

二者都不是——无用

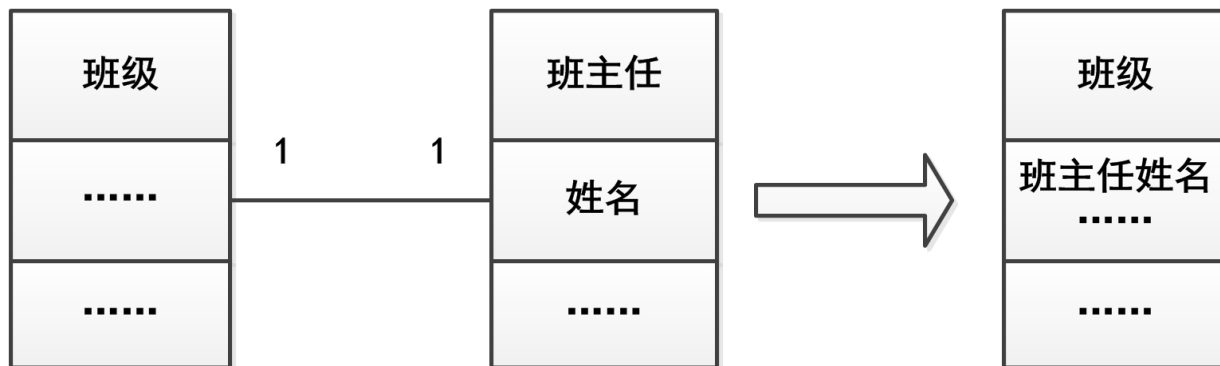


北京大学

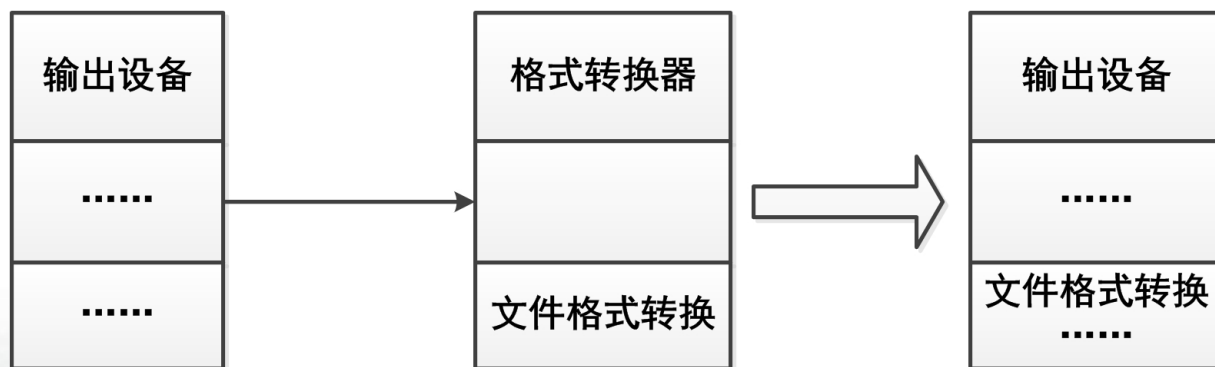


## (2) 对象的精简

### 只有一个属性的对象



### 只有一个操作的对象





### ( 3 ) 与实现条件有关的对象

例如：与

图形用户界面（ GUI ）系统、

数据管理系统、

硬件 及

操作系统有关的对象

—— 推迟到 OOD 考虑



北京大学



## 4、识别主动对象

(1) 考虑问题域和系统责任  
哪些对象需呈现主动行为?

(2) 从需求考虑系统的执行情况是否需要并发执行?  
控制线程的起点在哪个对象?

(3) 考虑系统边界以外的参与者与系统中哪些对象直接进行交互?

如果一个交互是由系统外的参与者发起的，第一个处理该交互的对象是主动对象

在分析  
阶段  
不能完  
全确定



北京大学





## 5、对象分类，建立类图中的类

### (1) 对象分类

使用问题域和系统责任知识，为每组具有相同属性和操作的对象定义一个类

### (2) 异常情况的检查和调整

#### \* 类的属性或操作不适合全部对象实例

例：“汽车”类的“乘客限量”属性

问题：分类不够详细

—— 进一步划分特殊类

#### \* 属性及操作相同的类

经过抽象，差别很大的事物可能只保留相同的特征

例如“吸尘器”和“电子琴”作为商品销售

—— 考虑能否合并为一个类

#### \* 属性及操作相似的类

—— 考虑能否提升出一个一般类或部分类（如轿车和货车，提取增加一般类“汽车”；机床和抽风机，提取部分类“电动机”）



## \* 同一事物的重复描述

例：“职员”和“工作证”  
——取消其中一个

### （3）类的命名

- \* 使用名词，避免无意义的符号
- \* 反映个体而不是群体（如书而非书籍）
- \* 适合该类及其特殊类的全部对象实例
- \* 使用问题域通用、规范的词汇
- \* 在中国：可用中、英文双重命名

### （4）建立类图的对象层

- \* 用类符号表示每个对象类，填写类的名称；
- \* 对于已经确认的主动对象，注意标注为主动类
- \* 填写类规约中关于每个类的详细信息；
- \* 发现的属性与操作、结构与连接  
可以随时加到类符号中。





## 二、识别属性



### (1) 策略与启发

- \* 按常识这个对象应该有哪些属性?(例如人的姓名、职业、地址等)
- \* 在当前的问题域中, 对象应该有哪些属性?(例如商品的条形码)
- \* 根据系统责任, 这个对象应具有哪些属性?(持卡人的使用地点)
- \* 建立这个对象是为了保存和管理哪些信息?
- \* 对象为了实现操作的功能, 需要增设哪些属性?  
(例如传感器对象, 为了实现其定时采集信号的功能, 需要一个“时间间隔”属性, 为了实现其报警功能, 需要一个“临界值”属性。)
- \* 对象是否需要通过专设的属性描述其状态?  
(例如设备对象, 在关闭、待命、运行、故障等不同状态将呈现不同的行需要为其设置一个“状态”属性)
- \* 用什么属性表示聚合和关联?  
(对于关联, 应该在关联一端的类中定义一个属性, 来指出另一端的哪个对象与本端的对象发生关联, 其数据类型是指向另一端对象的指针或对象标识)



北京大学



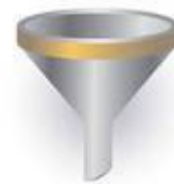
## \* 用什么属性表示聚合和关联?

1) 对于关联, 应该在关联一端的类中定义一个属性, 来指出另一端的哪个对象与本端的对象发生关联, 其数据类型是指向另一端对象的指针或对象类。

## 2) 对于聚合关系

- 如果是紧密、固定的聚合: 在整体对象中定义一个属性, 它是一个嵌套在整体对象中的部分对象, 所以它的数据类型就是部分对象类。
- 松散的聚合关系, 既可以在整体对象中定义一个属性指向另一端的哪个对象是它的组成部分, 也可以在部分对象中定义一个属性指向另一端的哪个对象是它的整体。其方向取决于聚合关系的多重性。





## (2) 审查与筛选

- \* 是否体现了以系统责任为目标的抽象  
(例：书的重量)？
- \* 是否描述对象本身的特征  
(例：课程—电话号码)？
- \* 是否破坏了对对象特征的“原子性”
- \* 是否可通过继承得到？
- \* 可以从其它属性直接导出的属性

## (3) 与实现条件有关的问题都推迟到 OOD 考虑

- \* 规范化问题 (OOA 中定义的对象属性可以是任何数据类型，数据类型的规范化工作在 OOD 中考虑)

- \* 对象标识问题

- \* 性能问题 (如为了提高操作的执行速度，增加一些属性来保持操作的阶段性执行结果)



北京大学





#### (4) 属性的命名：原则与类的命名相同：

- 使用名词或带定语的名词；
- 使用规范的、问题域通用的词汇；
- 避免使用无意义的字符和数字。

语言文字的选择要和类的命名要一致。

定位原则：一个类的属性必须适合这个类和它的全部特殊类的所有对象，并在此前提下充分运用继承。

#### (5) 属性的详细说明

要在类规约中对属性进行详细说明，其中包括：属性的解释、数据类型和具体限制等。

\* **属性的文字解释**：例如“课程”对象的“学时”属性，其解释为“课堂讲授学时数，每学时为 50 分钟”

\* **属性的数据类型**：常用的数据类型；表示整体 - 部分结构或关联的属性类型可以是类或某一类对象的指针。



北京大学



**\* 属性所体现的关系：**用于表示整体 - 部分关系或关联的属性，应该特别指明并加以解释。

例如对“课程”对象的“主讲教师”属性，可说明为：“表示课程与教师对象间的关联，指出该课程由哪个教师主讲。”

**\* 实现要求及其它：**如属性的取值范围、精度要求、初始值、度量单位、数据完整性及安全性要求、存取限制条件等。

如果一个属性实现时应作为类属性处理，也要在这里明确指出。





### 三、识别操作



#### 1、区分对象行为的类型

为了明确 OOA 应该定义对象的哪些操作，首先区分对象行为的不同类型：

##### (1) 系统行为

例：创建、删除、复制、转存

##### (2) 对象自身的行为——算法简单的操作

例：读、写属性值

##### (3) 对象自身的行为——算法复杂的操作计算或监控

类 名
属性1 ..... 属性n
操作1 ..... 操作m

仅用于操纵类属性的操作叫做类范围的操作，其余的操作叫做实例范围的操作。





## 2、发现操作的策略与启发

### (1) 考虑系统责任

要逐项审查用户需求中提出的每一项功能要求，看它应由哪些对象来提供，从而在该对象中设立相应的操作。

### (2) 考虑问题域

对象在问题域对应的事物有哪些行为？

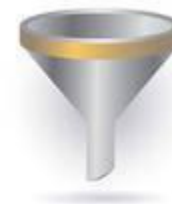
### (3) 分析对象状态

对象状态的转换，是由哪些操作引起的？

### (4) 追踪操作的执行路线

模拟操作的执行，并在整个系统中跟踪





### 3、审查与调整

**审查对象的每个操作  
是否真正有用，即**

是否直接提供系统责任所要求的某项功能？

**或者**

响应其它操作的请求间接地完成这种功能的某些局部操作？

**调整——取消无用的操作**

**是不是高内聚的**

一个操作只完成一项**单一的、完整的**功能

**调整：**

- (1) 拆分（一个操作中包括了多项可独立定义的功能）
- (2) 合并（一个独立的功能分割到多个对象操作中完成）



北京大学





## 4、认识对象的主动行为

### (1) 考虑问题域

对象行为是被引发的，还是主动呈现的？

(2) 与系统边界以外的活动者直接进行交互的对象操作

(3) 根据系统责任观察系统功能的构成层次，考虑完成最外层功能的对象操作。

(4) 操作执行路线逆向追踪，考察每个操作是被其它哪些对象的哪些操作请求的，直到发现某个操作不被其它成分所请求，则它应该是一个主动对象的主动操作。

OOA 标注的主动对象和主动操作不一定是最终的定局，因为在 OOD 阶段可能增加一些新的主动对象，还可能为提高或降低系统的并发度而人为的增加或减少主动对象。



北京大学



## 5、操作的命名和定位

命名：动词或动宾结构

定位：

与实际事物一致

例：售货员——售货，商品——售出

- 在一般 - 特殊结构中的位置

- —— 通用的操作放在一般类，专用的操作放在特殊类，一个类中的操作应适合这个类及其所有特殊类的每一个对象实例。



北京大学



## 6、操作的详细说明

**要在类规约中，要对操作进行详细说明，包括操作的解释、操作的特征标记、操作要发送的消息和约束条件等。**

- (1) **操作的文字解释**：解释该操作的作用和功能。
- (2) **操作名、输入输出参数、参数类型**：给出操作的入口消息格式。
- (3) **消息发送**：指出在这个操作执行时，需要请求哪些其他的对象操作。内容包括接收消息的对象类名以及执行这个消息的操作名。
- (4) **约束条件**：操作的执行的前置条件、后置条件以及执行时间的要求等事项说明。
- (5) **操作流程**：对于功能比较复杂的操作，要给出一个操作的流程图或活动图，表明该操作是怎样执行的。





# 案例分析

**例题：超级市场销售管理系统**

超级市场业务管理系统的子系统，只负责前台的销售管理

**功能需求：**

- 为顾客选购的商品计价、收费、打印清单。
- 记录每一种商品的编号、单价及现有数量。
- 帮助供货员发现哪些商品将要脱销，以及时补充货源。
- 随时按上级系统的要求报告当前的款货数量、增减商品种类或修改商品定价。
- 交接班时结算货款数目，报告上级系统。



北京大学



## 第一步：发现对象类

(1) 通过建立系统的 use case 图，在系统边界以外与系统进行交互的活动者有收款员、供货员和它的上级系统。这样，可以启发我们发现一些对象：

❖**收款机**：该对象直接与收款员这种活动者进行交互，模拟收款员的登陆、售货和结算等行为。由于某些行为是收款员的意愿主动发生的，所以将“收款机”定义为主动对象。

❖**供货员**：此类对象用来与实际的供货员进行交互（提醒他们及时补充货物）并模拟他们的行为（在增加货物时修改系统中的商品数量），这些行为是从系统内部引发的，所以它们是被动对象。

❖**上级系统接口**：用来处理与上级系统的交互。它的某些行为（如查账、更改商品的种类和价格）是由上级系统（而不是从本系统内部）引发的，所以它也应该是主动对象。

(2) 考虑系统问题域内部的事物和系统责任可以发现下述对象：

❖**商品**：这是系统中最明显的对象。每一个对象实例表示一种商品，记录该商品的名称、价格、数量等信息，并通过相应的服务动态地保持这些信息的准确性。

❖**特价商品**：这是一类特殊的商品，该类商品在指定的时间内按特殊价格销售，它有其特殊的属性。



北京大学





## 第二步：审查和筛选对象类

按照常识，在一个超级市场中收款员和管理人员都应该是一些值得考虑的对象，假如，我们把“收款员”和“经理”列为候选的对象类，现在考虑对象类的筛选，看看是否有必要保留这两个类：

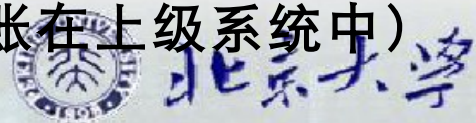
（1）本系统的功能需求没有要求对各类人员的信息进行计算机管理，所以各类人员对象是否有必要存在只是看这些人员的行为和个人信息是否对系统功能的履行起到一定的作用；

❖**计量商品**：这是另一类特殊的商品，它的包装是不标准的，或没有包装，需要在收款时按照它们的重量、长度或容积等单位进行计量，并按计量结果计算其价格。

❖**商品一览表**：考虑系统责任，为了在收款时能根据输入的商品编号快速找到相应商品的信息，需要设计一个“商品一览表”对象，它保持一个商品目录表，并提供对商品项的检索及增删等功能。

❖**销售事件**：顾客购买一组商品，只要通过一次计价收款完成的，就称为一个销售事件。每个这样的事件都需要保留一段时间，以便汇成账目并在必要时复查。所以，设立“销售事件”对象。

❖**账册**：记录一个收款员在一个班次内经手的所有销售事件的款、货账目，负责向上级系统报账，并在换班时进行账目交接。它的一个对象实例只针对一个收款员的一个班次，不是总账（总账在上级系统中）。





(2) 需求中没有包括对经理的工作进行计算机处理或提供辅助的支持，经理本人的信息对于完成需求中规定的业务处理功能也没有用处，所以不必设立“经理”对象；

(3) 收款员域系统的功能需求有密切关系，他们是与系统对话的活动者，系统应提供相应的对象处理这种对话，但在上述发现的对象中，“收款机”对象就是进行这种处理的，如果愿意，也可以把“收款机”对象改名为“收款员”对象，但没有必要设立两类对象。

**筛选原则：系统中任何对象都是为了提供某些信息或履行某些功能，如果没有这些用途，则这种对象就没有必要在系统中存在。**





## 建立 OOA 模型的对象层如下图所示：

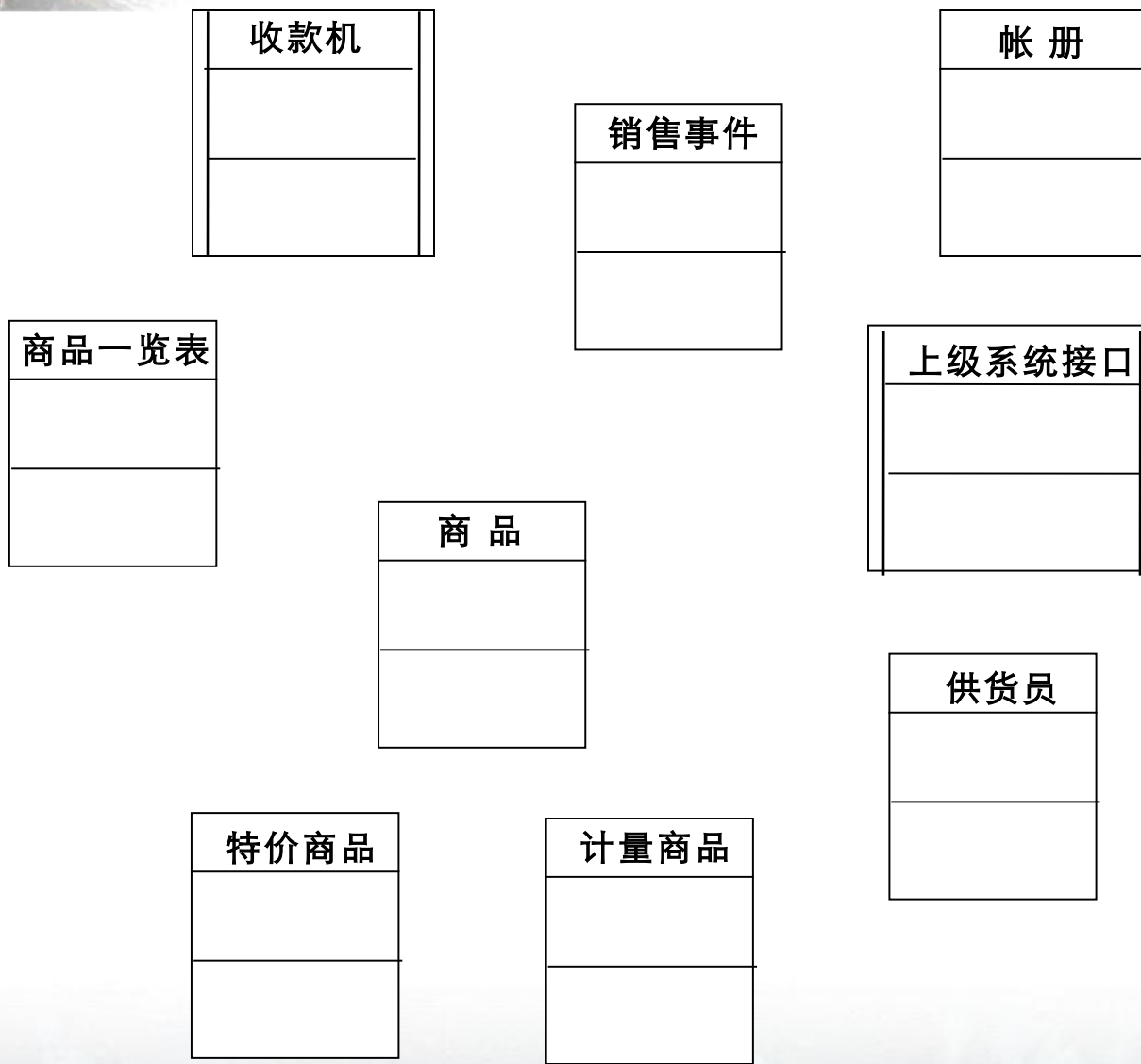


图 超级市场销售管理系统（对象层）





### 第三步：分析每个类的属性和操作

#### (1) 收款机：

属性：应指明当前是哪个收款员在本台收款机上工作（本班收款员），她或他在本次工作的开始和结束时间（开始时间和结束时间）。

操作：

- 登录：本班收款员开始工作，它是一个主动操作；
- 售货“循环地为每个顾客计价收款；
- 结账：在收款员下班或交班时结算本班的账目。

#### (2) 商品：

属性：商品的编号、名称、单价、架上数量及下限

操作：

- 售出：从架上数量减去已售出商品的数目，若剩余的数目低于下限则向“供货员”对象发消息；
- 补充：当供货员补充了一些商品时，把补充数量与该种商品原先的架上数量相加；
- 价格更新：由“上级系统接口”对象请求修改商品的价格；

#### (3) 特价商品：

继承了“商品”的所有属性和操作，同时具有自己的特殊属性（开始日期和结束日期），指明实行特价的时间范围；



北京大学



#### (4) 计量商品：

继承了商品的所有属性和操作。

属性：“单价”属性的语义发生了变化，指的是一个计量商品的价格。

补充的特殊属性有“计量单位”和“计价方式”。

操作：对于从“商品”类中继承的操作要重新定义，因为算法发生了变化。

#### (5) 商品一览表：

属性：是一个汇总店内所有商品的“商品目录”，它是一个数组，每一个元素包括一件商品的编号和指向“商品”对象的指针（或对象标识）

操作：

- 检索：通过编号查找相应的商品对象；
- 种类增删：增添或删除“商品目录”中的商品项；

#### (6) 供货员：

属性：一个缺货记录表，每当某种商品的架上数量低于其下限时，就在这个登记表种记录下来。

操作：

- 缺货登记：将告缺的商品名称及编号记到登记表中；
- 供货：在供货员补充了货物之后，向“商品”对象发服务请求更改其数量，并删除缺货登记表中相应的条目。



北京大学





## (7) 销售事件:

### 属性:

- 收款人: 记录由哪个收款员在哪台收款机上处理这个销售事件;
- 购物清单: 记录顾客选购的每件商品的编号、名称、数量及价格;
- 应收款: 累计所有被选购商品的价格总和;
- 售出日期及时间

### 操作:

- 销售计价: 逐条记录商品清单, 并累计应收款数;
- 入账: 将本次销售事件的信息计入账册;

## (8) 账册:

属性: 记录一个收款员在一个班次中的每个销售事件, 并累计其销售收入。实际上, 用一组指针 (或对象标识) 指向每个销售事件 (销售时间表), 就可得到它们的明细信息。因此, “账册” 与 “销售事件” 有聚合关系。

前班结余、本班结余、上交款, 以及该账册开始使用和结账日期和时间等。

### 操作:

- 接班: 记录从上一班收款员那里接收了多少未上交的货款 (因为收款机上总需要保留一些零钱);
- 记账: 记录每一个 “销售事件” 对象, 并累计其收入金额;
- 报账: 向上级系统报账;
- 交班: 向上级系统报账, 记录上交的款数和移交给下一班收款员的款数;



北京大学



### (9) 上级系统接口：

属性：设立一个“账册目录”属性，记录店内所有正在使用和已经结算的账册的指针信息，以便及时找到它们；

操作：

- 消息收发：负责与上级系统通信，并通过请求其他对象类的服务完成上级系统要求处理的事项，该操作是主动操作；
- 查账：按上级系统的要求查阅账目并报告结果；
- 报账：从本系统向上级系统报告账目；
- 价格更新：按照上级系统传来的信息，更新指定商品的单价；
- 种类增删：按照上级系统的信息增添或删除商品对象及其在商品一览表中的条目。

根据以上对属性和操作的分析，得到该系统 OOA 模型的特征层，如下图所示：



北京大学



收款机
本班收款员 开始时间 结束时间
登录 售货 结账

销售事件
收款人 购物清单 应收款 .....
销售计价 入账

账 册
前班结余 销售事件表 收入累计 上交款 本班结余
接班 记账 报账 交班

商品一览表
商品目录
检索 种类增删

商品
编号 名称 单价 架上数量 下限
售出 补充 价格更新

上级系统接口
帐册目录
消息收发 查账 报账 价格更新 种类增删

供货员
缺货登记表
缺货登记 供货

特价商品
开始日期 结束日期

计量商品
单价 计量单位 计价方式
售出 补充 价格更新



北京大学



# 识别对象之间的关系



北京大学

# 一、识别继承（泛化）



## 1、识别继承（泛化）关系的策略

### （1）学习当前领域的分类学知识

因为问题域现行的分类方法往往比较正确地反映事物的特征、类别以及各种概念的一般性和特殊性。按照问题域已有的分类方法，可以找出一些与它对应的继承关系。

### （2）按常识考虑事物的分类

如果问题域没有可供参考的现行分类方法，可以按照自己的常识，从各种不同的角度考虑事物的分类，从而发现继承关系。

例如对于“人员”可以从以下几种角度去分类：

- 青年人、成年与老年；
- 男人与女人；
- 黄种人、白种人和黑种人；
- 在职人员与离退休人员；
- 正式职工与临时工；
- 。。。。。



从不同的角度考虑问题域中事物的分类，可以形成一些建议一般 - 特殊关系的初步设想，从而启发自己发现一些确实需要的一般 - 特殊关系。



北京大学





### (3) 使用继承的定义

使用两种思路去发现继承关系：

- (a) 一种思路是把每个类看作是一个对象集合，分析这些集合之间的包含关系，如果一个类是另一个类的子集（例如“职员”是“人员”的子集，“轿车”是“汽车”的子集），则它们应组织到同一个一般 - 特殊关系中。
- (b) 看一个类是不是具有另一个类的全部特征，这又包括以下两种情况：
  - 一种是建立这些类时已经计划让某个类继承另一个类的全部属性与操作，现在应建立继承关系来落实；
  - 另一种是起初只是孤立地建立每个类，现在发现一个类中定义的属性与操作全部在另一个类中重新出现，此时应考虑建立继承关系，把后者作为前者的特殊类，以简化其定义。

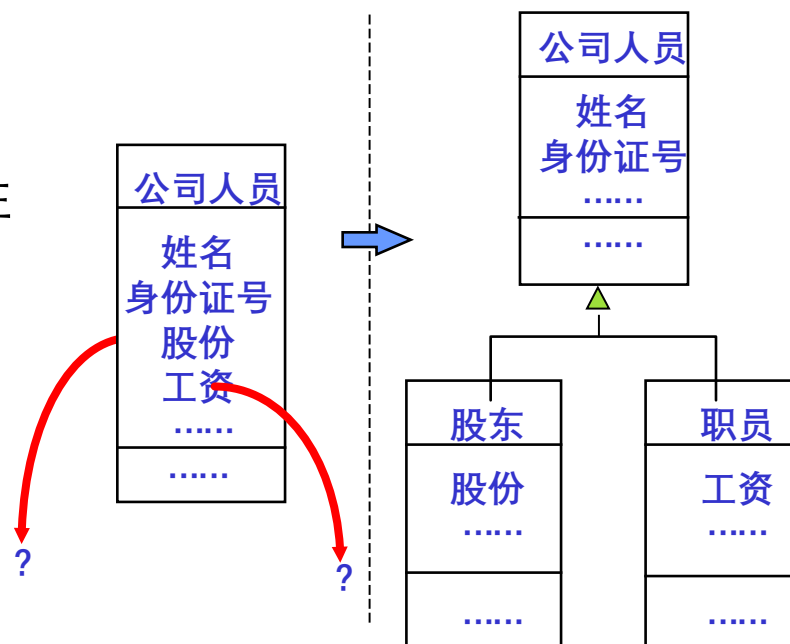




## (4) 考察属性与操作的适用范围

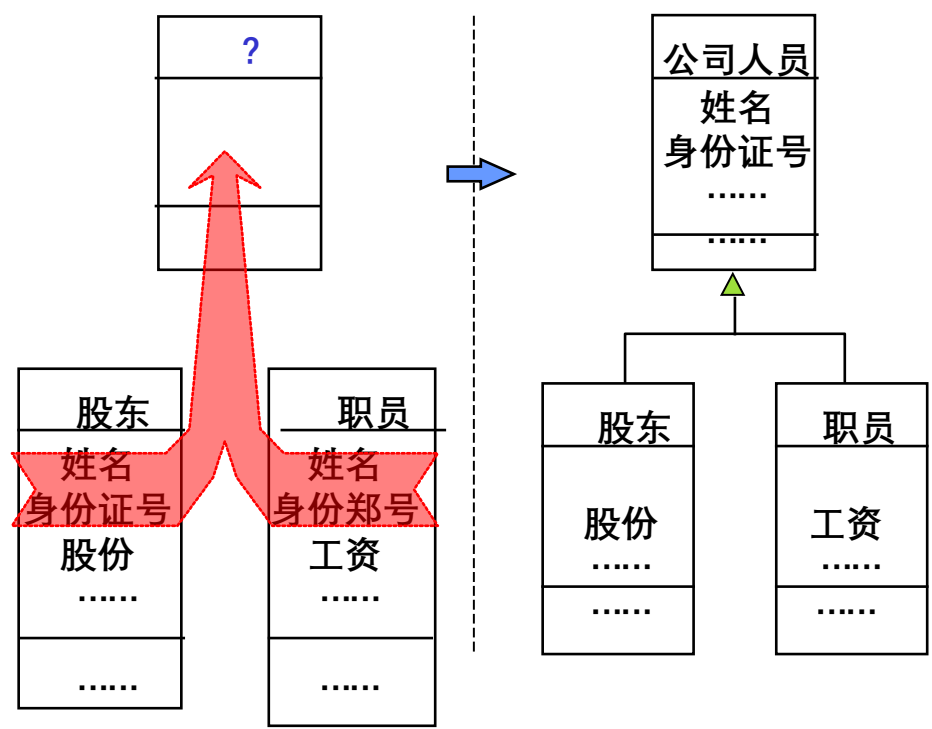
对系统中的每个类，从以下两方面考虑它们的属性与操作：

- 看一个类的属性与操作是否适合这个类的全部对象。如果某些属性或操作只适合该类的一部分对象，说明应从这个类中划分出一些特殊类，建立继承关系。



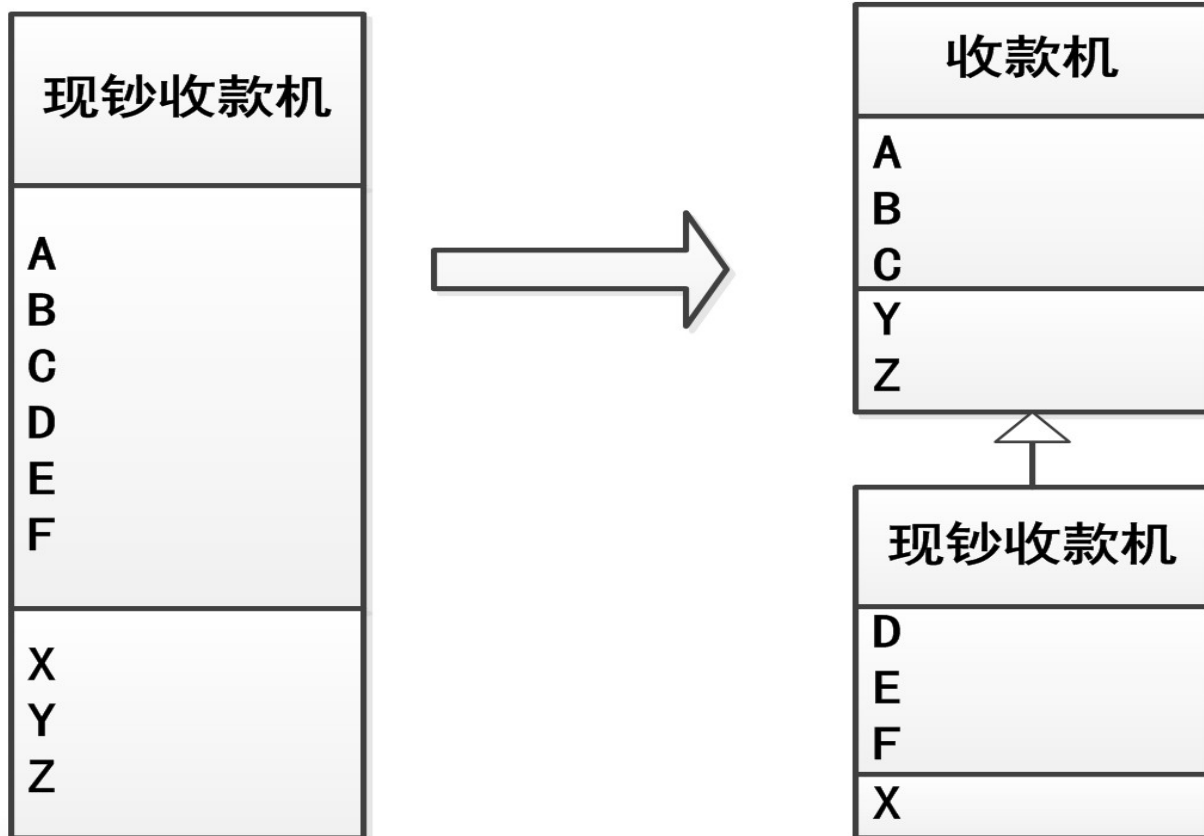


- **检查是否有两个（或更多）的类含有一些共同的属性与操作。**如果有则考虑，若把这些共同的属性与操作提取出来，能否构成一个在概念上包含原先那些类的一般类，形成一个继承关系。





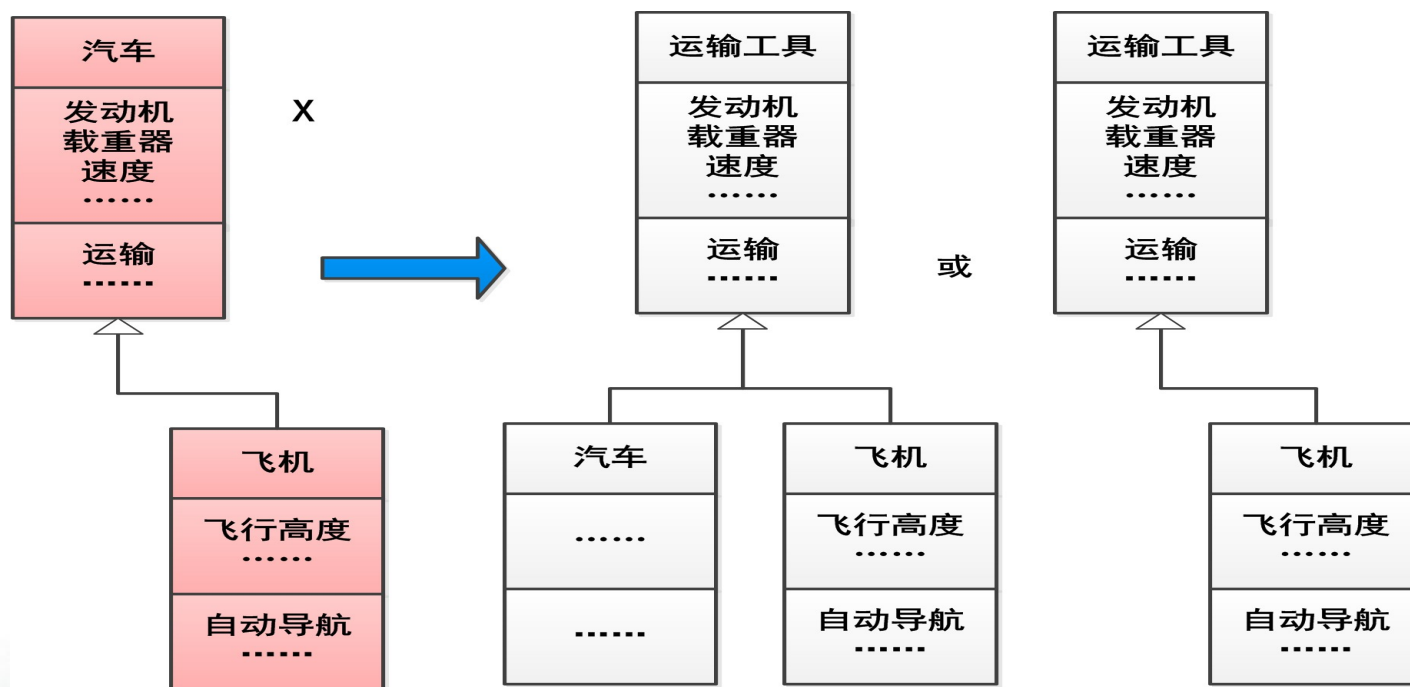
## (5) 考虑领域范围内的复用





## 2、审查与调整

- (1) 问题域是否需要这样的分类? (例: 书—线装书)
- (2) 系统责任是否需要这样的分类? (例: 职员—本市职员)
- (3) 是否符合分类学的常识? (用“is a kind of”去套)



- (4) 是否构成了继承关系? (确实继承了一些属性或操作)

例如: 航标船



北京大学





### 3、继承关系的简化

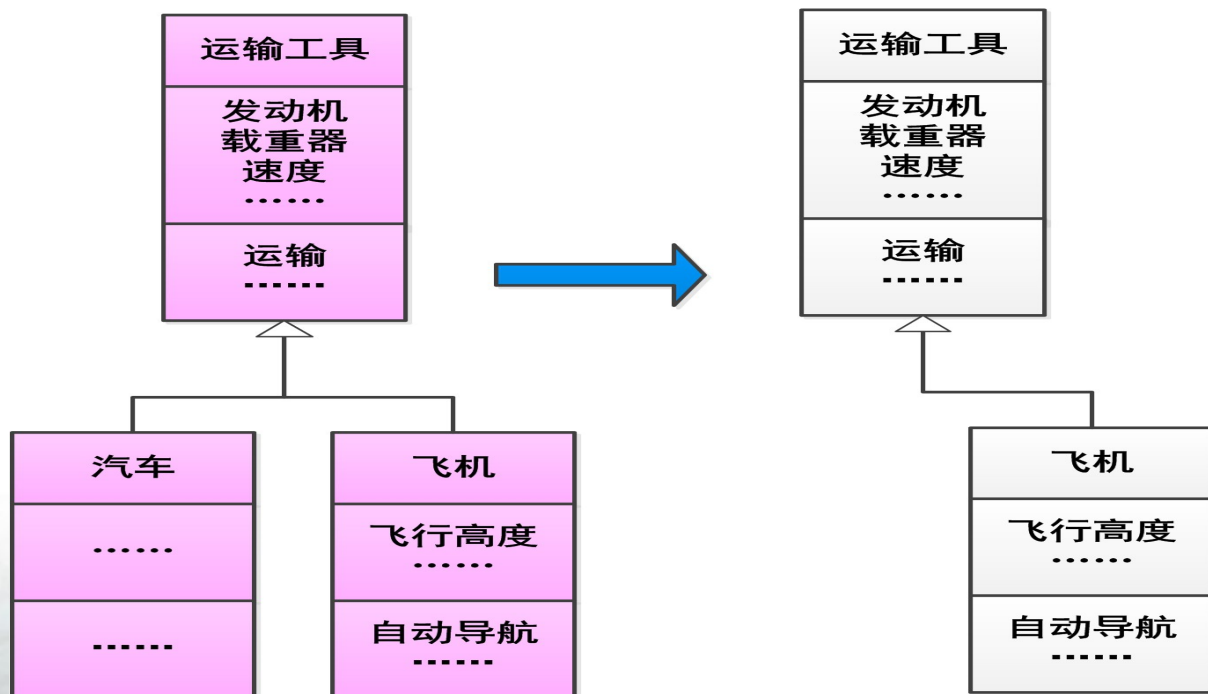
(1) 从一般类划分出太多的特殊类，使系统中类的设置太多，增加了系统的复杂性；

(2) 建立过深的继承层次，增加了系统的理解难度和处理开销。

对继承关系的运用要适度

重点考察以下情况：

(1) 取消没有特殊性的特殊类



北京大学



(2) 增加属性简化继承关系 (某些特殊类之间的差别可以由一般类的某个属性值来体现, 而且除此之外没有太多的不同)

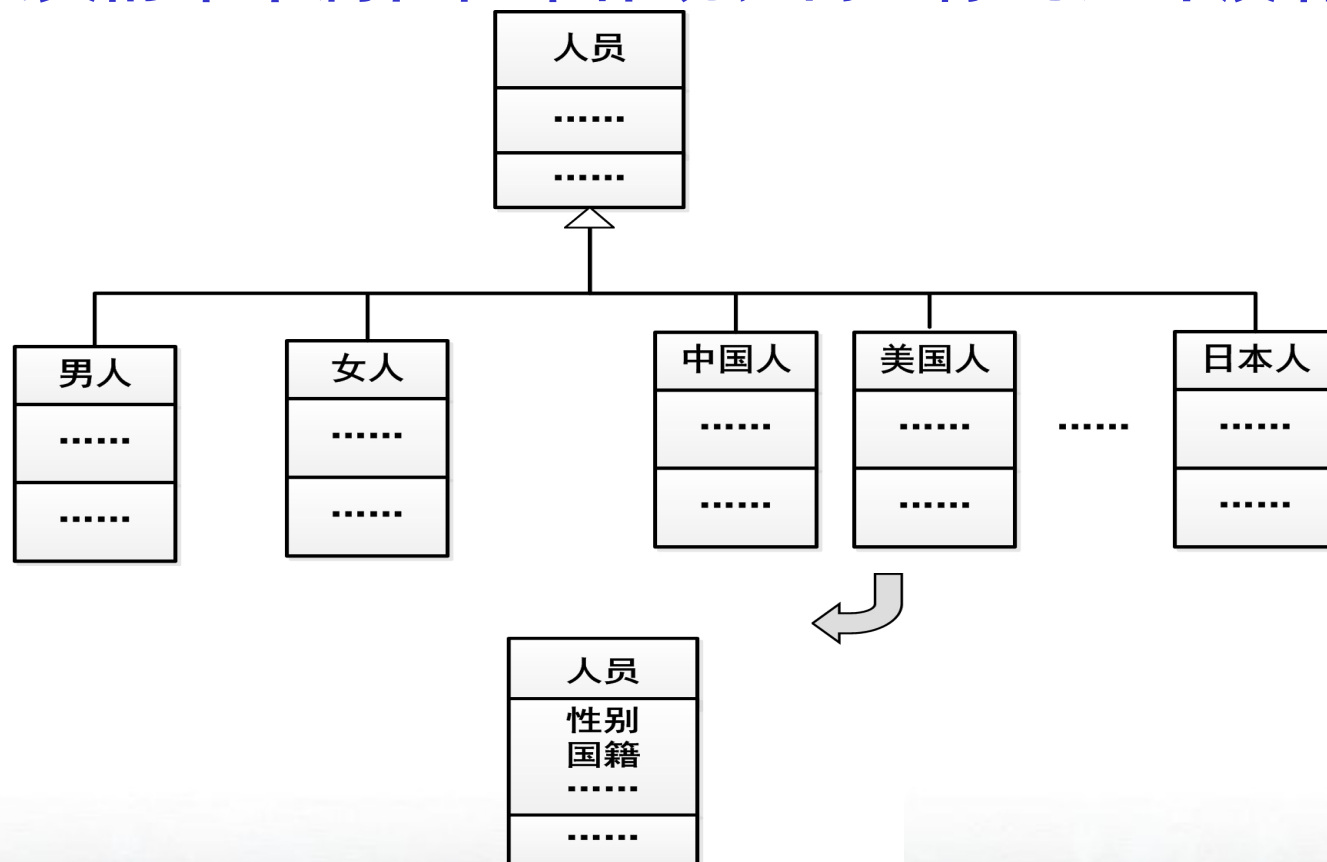


图 通过增加属性简化继承关系





### (3) 取消用途单一的一般类，减少继承层次

一般类存在的理由：

- 有两个或两个以上的特殊类
- 需要用它创建对象实例
- 有助于软件复用



北京大学



## 4、调整对象层和特征层

定义继承的活动，将使分析员对系统中的对象和类及其特征有更深入的认识，在很多情况下，随着继承的逐步建立，需要对类图的对象层和特征层进行某些修改，包括增加、删除、合并或分开某些类，以及增、删某些属性与操作或把它们移到其他类。



## 二、识别关联



### 1、识别关联的策略

#### (1) 认识对象之间的静态联系

考虑问题域和系统责任——哪些类的对象实例之间的关系需要在系统中表达。

#### (2) 认识关联的属性与操作

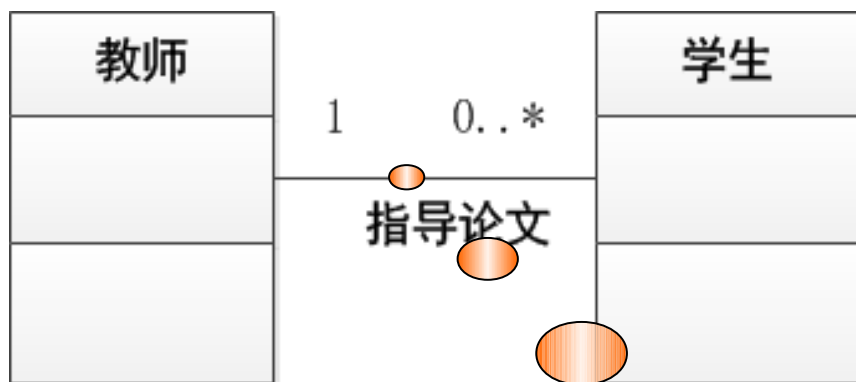
对于考虑中的每一种关联，进一步分析它是否应该带有某些属性和操作。就是说，是否含有一些仅凭一个简单的关联不能充分表达的信息。

例：在教师和学生的“指导毕业论文”的关联中，是否需要给出毕业论文的题目、答辩时间、成绩等属性信息？

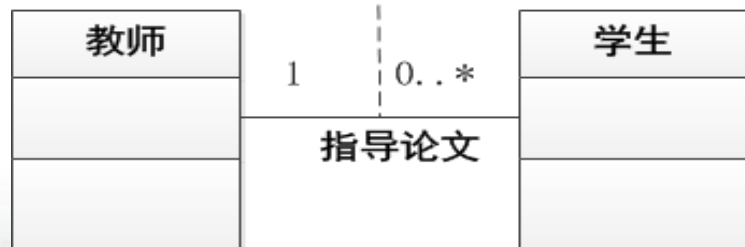
如果有，则可先在关联线上附加一个关联类符号来容纳这些属性和操作，或在两个类之间插入一个类来描述这些属性与操作。







有某些信息  
需要描述  
(如题目、  
时间、成  
绩)



把带有属性和操作的关联表示为  
关联类（如指导论文）



北京大学



### (3) 分析并表示关联的多重性

对于每个关联，从连接线的每一端看本端的一个对象与另一端的几个对象发生连接，把结果标注在连接线的另一端。

### (4) 进一步分析关联的性质

若需要，使用关联角色和**限定符**，以详细描述关联的性质。

限定符的值用于确定该关联的另一端类的对象。即给定类的一个对象，并指定限定符内的属性值，能选定另一端类的一个对象或一组对象。





把限定符画成一个与关联的一端和该端的类图符相连的小矩形，并把受限的类（如“定单行”）的属性放在小矩形中



通常产品定单由定单行和一些其他描述信息组成。上图使用带有限定符的组合描述定单、定单行以及它们之间的关系。

对于一份定单，并指定了产品名，在另一端可能有或没有一个定单与其对应。如果没有这个限定符，给定一份定单，对应的定单行可能有许多。



北京大学





## 2、命名与定位命名：

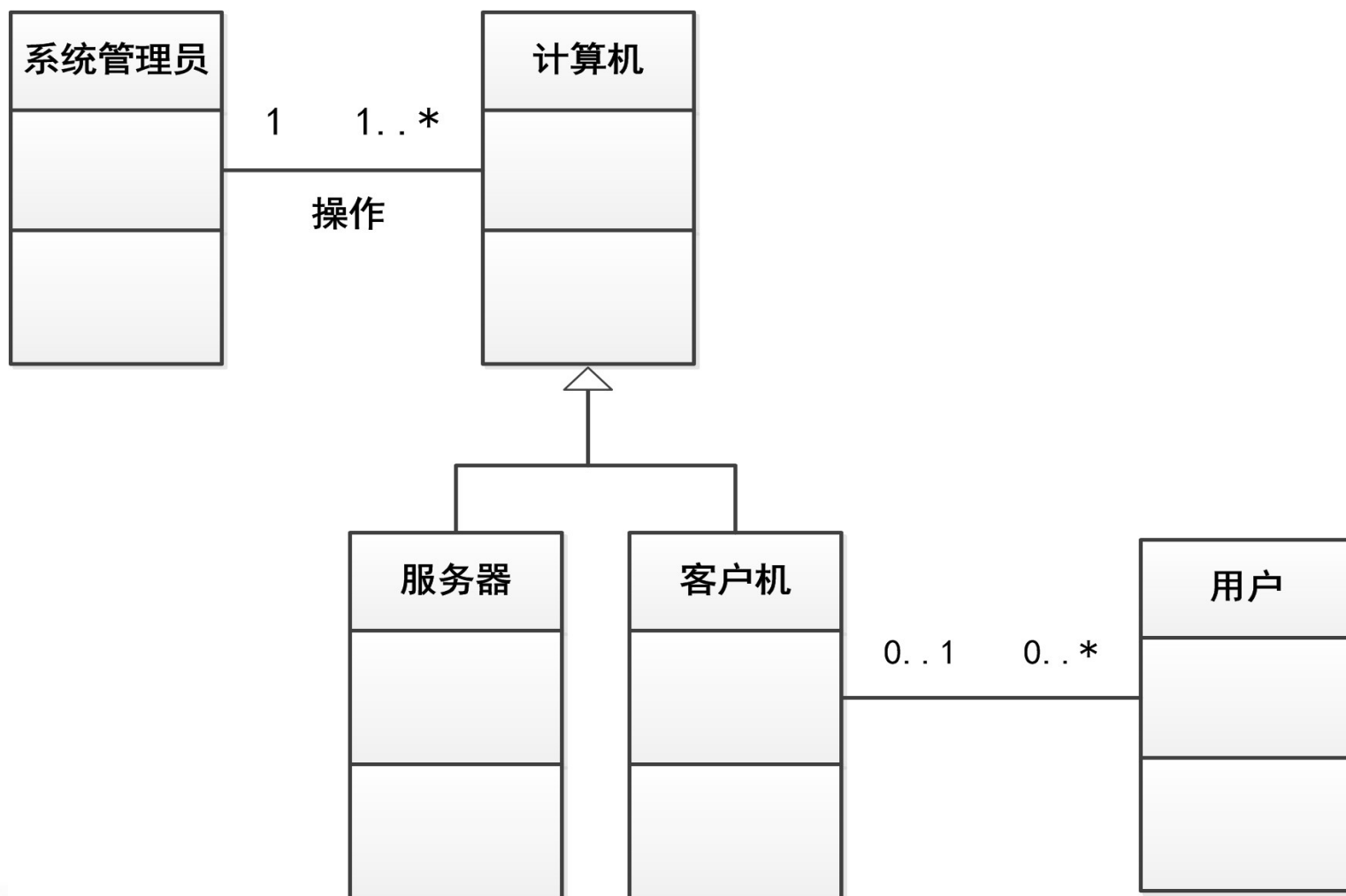
当关联线的语义很清晰时，则关联的命名可缺省。否则，**关联可用动词或动宾结构命名。**

**定位问题：**当连接线的某一端是一个继承结构时，要考虑连接线画到结构中的哪个类符号上。

**原则：**如果这个关联适应结构中的每个类的对象，则画到一般类上，如果只适应其中某些特殊类，则画到相应的特殊类上。









### 3、调整对象层和特征层

在建立关联的过程中可能增加一些新的类，要把这些新增加的类补充到类图的对象层中，并建立它们的类规约。

对于每一个关联，要给出其有关性质的说明，至少要说明它所表示的实际意义。



# 三、识别聚合



## 1、识别聚合的策略

(1) 物理上的整体事物和它的组成部分

例：机器、设备和它的零部件

(2) 组织机构和它的下级组织及部门

例：公司与子公司、部门

(3) 团体（组织）与成员

例：公司与职员

(4) 一种事物在空间上包容其它事物

例：生产车间与机器

(5) 抽象事物的整体与部分

例：学科与分支学科、法律与法律条款

(6) 具体事物和它的某个抽象方面

例：人员与身份、履历

(7) 在材料上的组成关系

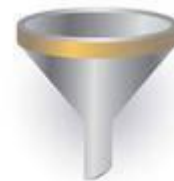
例如，面包由面粉、糖和酵母组成，汽车是由钢、塑料和玻璃组成。



北京大学



## 2、审查与筛选



(1) 是否属于问题域？

例：公司职员与家庭

(2) 是不是系统责任的需要？

例：公司与工会

(3) 部分对象是否有一个以上的属性？

例：汽车与轮胎（规格）

(4) 是否有明显的整体 - 部分关系？

例：学生与课程



## 3、调整对象层和属性层

定义聚合关系的活动可能发现一些新的对象类，或者从整体对象的类定义中分割出一些部分对象的类定义，应把它们加入到对象层中，并给出它们的详细说明。



北京大学



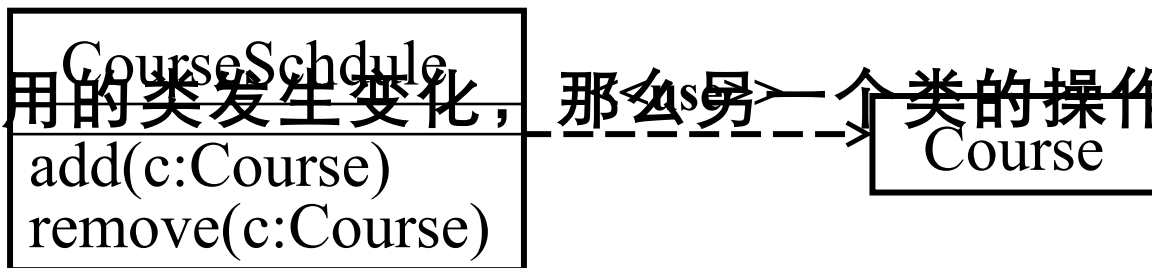
## 四、识别依赖



依赖是一种使用关系，用于描述一个事物（如类 Window）使用另一事物（如类 Event）的信息和服务。

- ① 在大多数情况里，使用依赖来描述一个类使用另一个的操作；

- ② 如果被使用的类发生变化，那么另一个类的操作也会受到影响；



建议：在初步建立类之间的关系时，可以暂时使用依赖。在最终的类图中，若能用其他关系明确地指明类之间关系的含义，就不要使用依赖。







# 案例分析（续）

## 例题：超级市场销售管理系统

超级市场业务管理系统的子系统，只负责前台的销售管理

### 功能需求：

- 为顾客选购的商品计价、收费、打印清单。
- 记录每一种商品的编号、单价及现有数量。
- 帮助供货员发现哪些商品将要脱销，以及时补充货源。
- 随时按上级系统的要求报告当前的款货数量、增减商品种类或修改商品定价。
- 交接班时结算货款数目，报告上级系统。

### 发现对象：

收款机、供货员、上级系统接口、商品、特价商品、  
计量商品、商品一览表、销售事件、帐册





## 第四步：定义对象类之间的关系

### (1) 继承：

一般类“商品”和它的两个特殊类“特价商品”及“计价商品”构成继承关系，并且特殊类中的某些属性和操作是多态的。

### (2) 聚合：

“商品一览表”和“商品”构成聚合关系，通过前者的“商品目录”属性体现这种关系；

“账册”和“销售事件”构成聚合关系，通过前者的“销售事件表”属性体现这种关系。

### (3) 关联：

这个例子中没有其他关联关系。

### (4) 依赖：

依赖是一种使用关系，说明一个事物（如类 windows）使用另一个事物（如类 event）的信息和服务。

➤由于 UML 在类图上不能体现对象在行为上的关系，如果开发者想在类图上体现对象行为上的关系，就只能借用“依赖”这个概念及其表示法来弥补消息的空缺。





➤如果一类对象的行为依赖另一类对象的行为，即一类对象的操作在执行时需要依靠和使用另一类对象的操作所提供的功能。

——《面向对象的系统分析（第2版）》，邵维忠，杨芙清著，北京：清华大学出版社，2006.12.）

从收款员、供货员和上级系统这三类活动者的相关对象开始执行路线追踪，以发现对象行为之间的依赖关系：

（1）“收款机”

- “收款机”对象在执行“售货”操作时向“商品一览表”对象发消息，请求其“检索”服务以找到相应的“商品”对象；
- “收款机”对象向“商品”对象发消息以获知该种商品的属性信息并请求其“售出”服务；
- 若“商品”对象发现该种商品数量低于规定的下限，则向“供货员”对象发消息，请求其“缺货登记”服务；
- “收款机”对象向“销售事件”对象发消息，请求“销售计价”和“入账”服务；
- 在执行“入账”操作时，“销售事件”对象向“账册”对象发消息，以请求其“记账”服务；



北京大学



- “收款机”对象在执行自己的“登陆”操作时，向“账册”对象发消息，请求“接班”服务；
- “收款机”对象在执行“结账”操作时，也要向“账册”对象发消息，请求其“报账”和“交班”服务；
- “账册”对象向“上级系统接口”对象发消息，请求其“报账”服务。

## （2）上级系统接口

- “上级系统接口”对象在上级系统要求查账时向“账册”对象”发消息，请求它的”报账“服务；
- 在上级系统要求进行价格更新或商品种类增删时，分别向“商品”和“商品一览表”对象发消息，请求相应的服务。

## （3）供货员

- “供货员”对象在执行“供货”操作时，向“商品”对象发消息，请求其“补充”服务。





例如：超级市场  
销售管理系统  
(关系层、完整  
的类图)

