第8章.面向对象分析与UML建模

韩锐

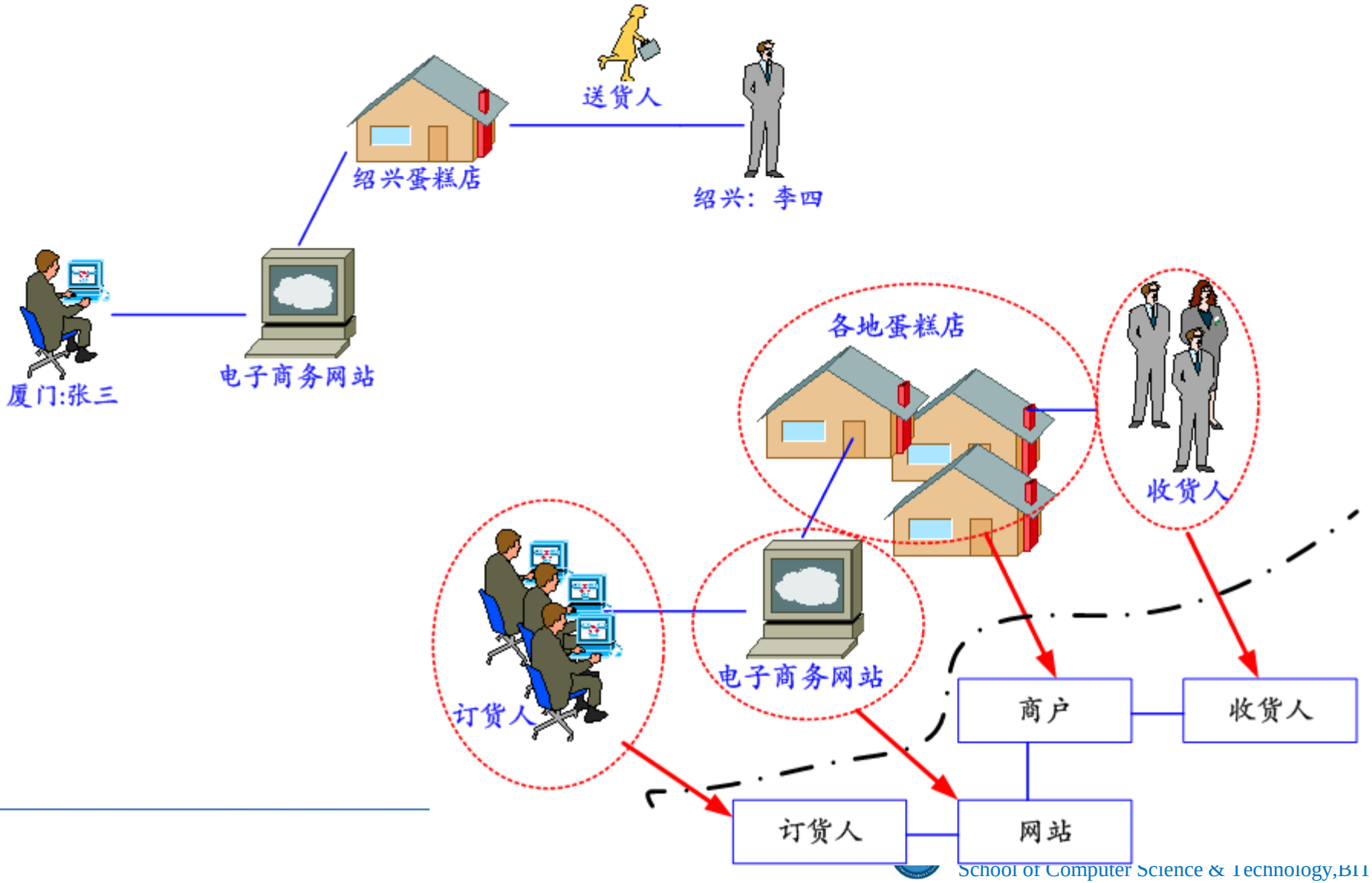
北京理工大学 计算机学院

Email: 379068433@qq.com
hanrui@bit.edu.cn

面向对象分析与UML建模(3)

- 1. 面向对象建模
  - 1.1. 类图和对象建模
  - 1.2. 识别对象和筛选策略
  - 1.3. 从对象抽象类
  - 1.4. 主动对象和控制线程
- 2. 类的命名、属性和操作
  - 2.1. 类的命名
  - 2.2. 类属性及识别筛选策略
  - 2.3. 类操作及识别调整策略
- 3. 类的继承和关联
  - 3.1. 类的继承关系
  - 3.2. 建立类的关联
- 4. 接口类
- 5. 包图

面向对象思想



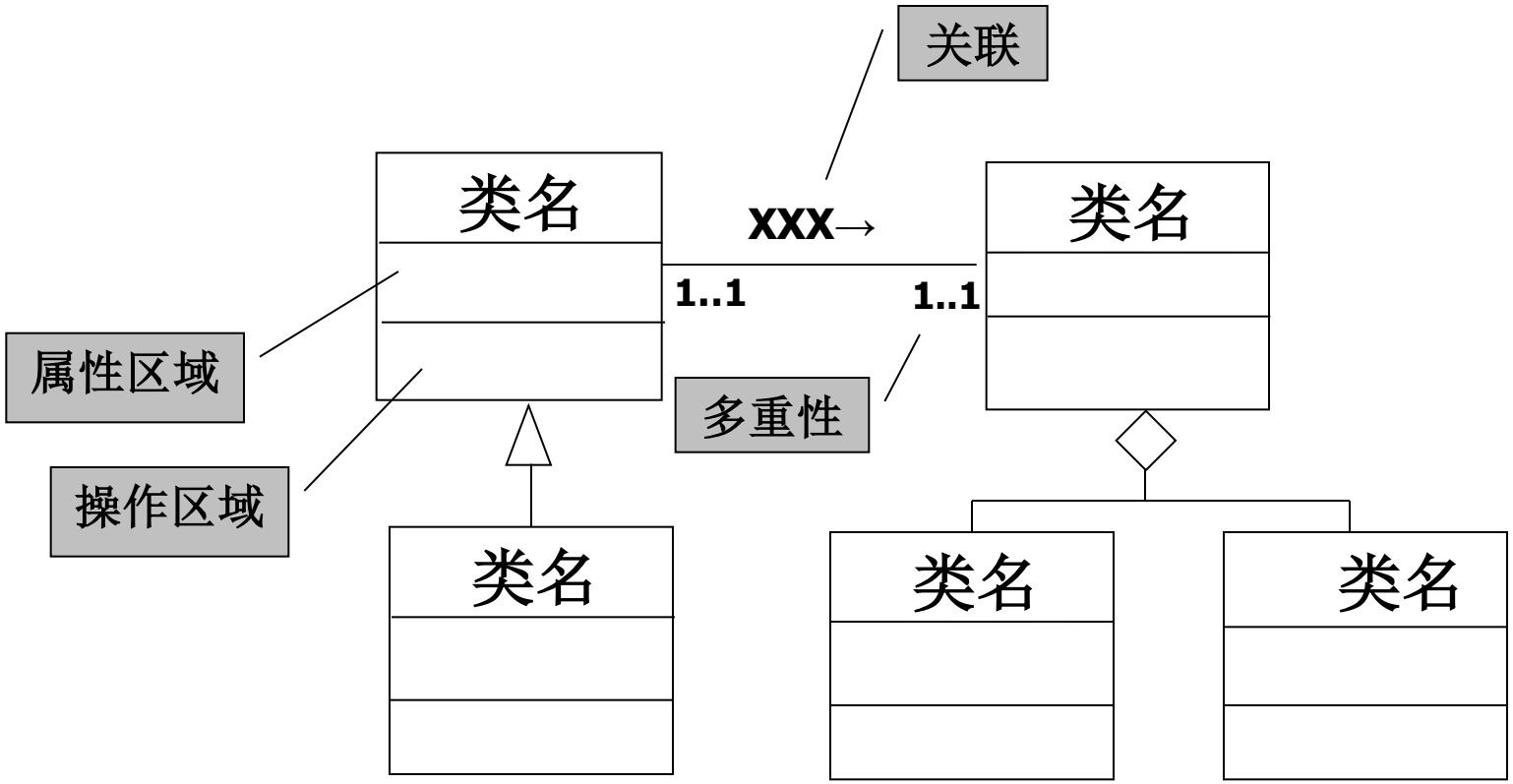
面向对象思想

- 每个对象都扮演了一个角色，并为其它成员提供特定的服务或执行特定的行为。
  - 在面向对象世界中，行为的启动是通过将“消息”传递给对此行为负责的对象来完成的；
  - 同时还将伴随着执行要求附上相关的信息（参数）；
  - 而收到该消息的对象则会执行相应的“方法”来实现需求
- 用类和对象表示现实世界，用消息和方法来模拟现实世界的核心思想

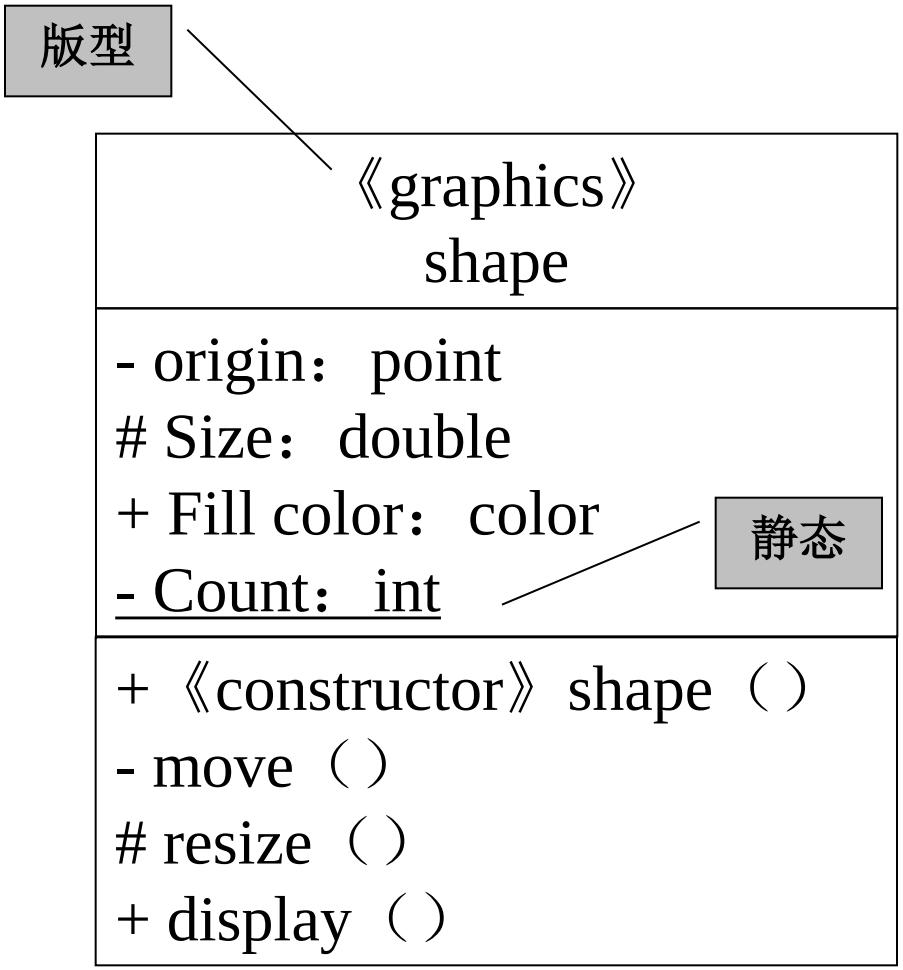
1.1. 类图和对象建模

类图，是系统建模过程中最重要的部分，也是花费精力最大的活动。类图描述系统中各个对象之间存在的关系，表达系统的静态结构，也叫做“对象建模”。

类图的基本图元素



类的属性和操作表示



可见性(visibility)：

private -

public +

protected #

对象和类

对象定义：对象是具有明确语义边界并封装了状态和行为的实体，即它是系统中用来描述客观事物的一个实体，是构成系统的一个基本单位，由一组属性和作用在这组属性上的一组操作构成。

类定义：类是对具有相同属性、操作、关系和语义的对象集合的描述。

1.2. 识别对象和筛选的策略

对象建模的基础：

在用例图完成捕获需求后，将问题域和系统责任作为基础，分析系统中的对象和类。

识别对象

方法1：

利用需求得到的问题陈述，从中挑选名词或代词，以及名词短语来识别对象和类。

1.2. 识别对象和筛选的策略

方法2：

直接考虑现实问题中的对象，对应为系统中的对象。

方法3：

从系统边界外发现与系统进行交互的参与者，寻找系统处理对外接口的对象类。

方法4：

对照系统责任所要求的每项功能，确定能完成这些功能对象。

识别对象示例:提取名词和名词短语

家庭安全系统（SafeHome）问题陈述：

系统由房主在安装时配置，通过系统控制窗口和键盘功能与房主交互进行。系统可以监控所有连接的传感器。

在安装过程中，每个传感器可以设置编号和类型，系统的启动和关闭必须有主人密码控制；传感器事件发生时，软件激活系统上的警报器，在设定的时间延时后，软件自动拨打设定的电话号码，并提供位置信息和事件性质，电话号码将每隔20秒重拨一次，直至电话接通。

系统运行时，系统控制窗口显示提示信息和系统状态。通过键盘可以控制系统运行。

整理提取的名词

SafeHome问题陈述中的名词：

（去掉相同意义的名词和名词短语，归并后保留）

系统、房主、系统控制窗口、键盘功能键、提示信息、
系统状态、安装过程、配置系统、系统运行、
系统连接的传感器、编号、类型、位置信息、事件性质、
系统启动、系统关闭、主人密码、传感器事件、电话、
系统上的警报器、设定的时间延时、设定的电话号码

舍弃不是类的名词

1）是属性：系统需要保存的有用信息

可能是属性：

传感器编号、传感器类型、位置信息、事件性质、主人密码、时间延时、电话号码、系统状态

2）是操作：系统应提供的功能或者某事务过程

可能的操作：

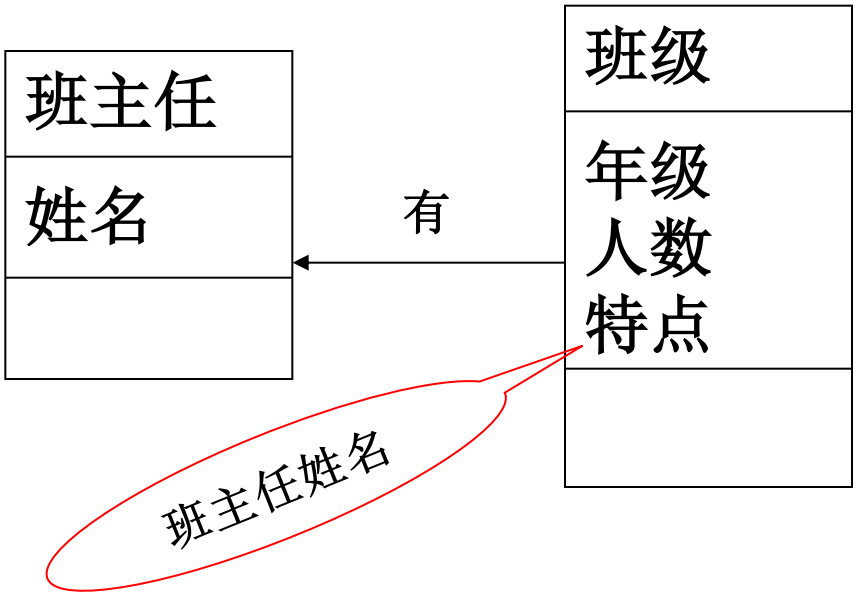
安装过程、配置系统、系统运行、系统启动、系统关闭、提示信息、系统状态显示、

精简对象策略

- 精简对象
  - 如果对象只有一个属性和相应的操作，并且仅被唯一的对象引用，可以合并到引用它的对象中去。
  - 如果一个对象只有一个操作，而没有属性，并且系统中只有一个类的对象请求这个操作，可以把它合并到它的请求对象中。

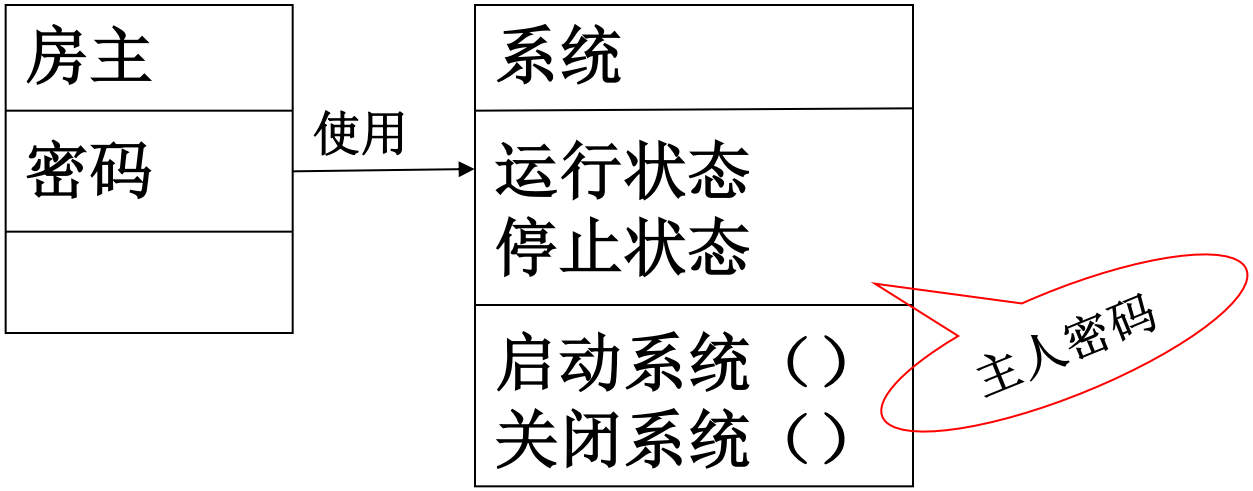
精简对象示例1

如果班主任对象，仅有班级对象需要引用，且仅有姓名属性和相应的操作，则可以将班主任姓名作为属性放到班级对象中，取消班主任对象。



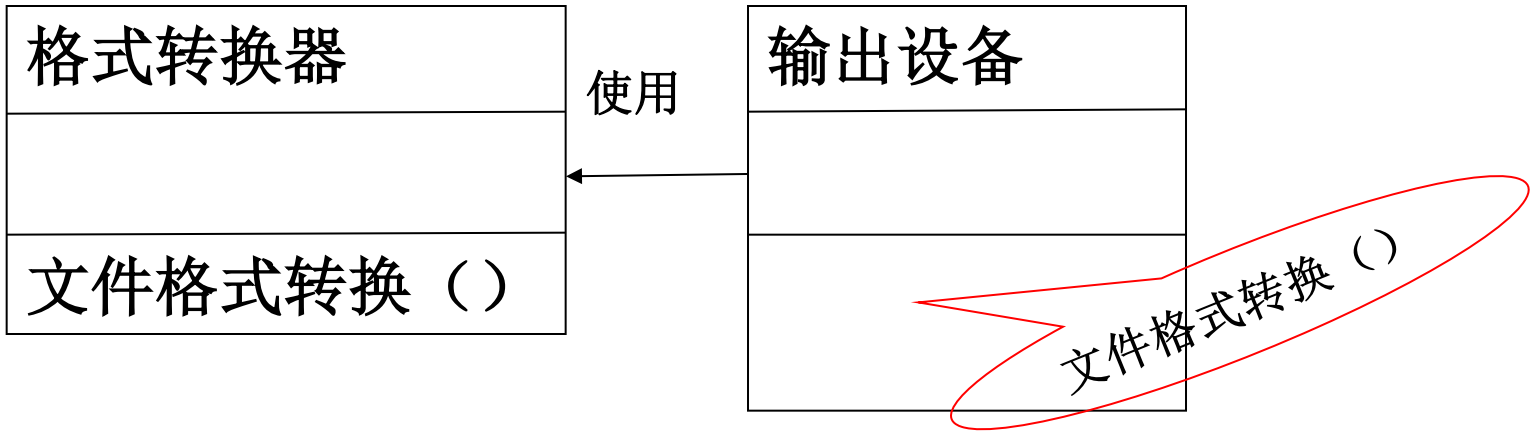
精简对象示例2

在Safehome中，如果房主对象，只有“密码”一个属性，在系统启动和关闭系统时，需要引用房主对象来执行密码读取操作；这样可以考虑把“房主密码”作为一个属性，增加到系统对象中，去掉房主对象。



精简对象示例3

格式转换器，只有一个“文件格式转换”操作，没有属性。且该操作只有输出设备对象使用，可以考虑将该操作放到输出设备类中，取消格式转换器类。



注意：没有属性而有操作的对象类是经常需要的。
例如：超类。

分析筛选对象策略

- 推迟到设计考虑的对象

系统责任要求的某些功能与实现环境有关，应该把这样的功能所涉及的对象，推迟到设计阶段去考虑，OOA模型应独立于具体的实现环境。

例如：在Safehome中，为系统控制窗口对象而设定的具体对象，包括窗口对象、对话框对象、下拉采单对象、按钮对象等等，应该到设计时再考虑。

SafeHome问题经筛选后的对象

可能的操作：

安装过程
配置系统
系统运行
系统启动
系统关闭
提示信息
系统状态显示

可能是属性：

传感器编号
传感器类型
位置信息
事件性质
主人密码
时间延时
电话号码
系统状态

有用的对象：

系统
控制窗口
键盘功能键
传感器
传感器事件
警报器
电话

精简的类：

房主

1.3. 从对象抽象类

■从对象抽象类

依据对象和类的定义，使用概括来寻找两个或多个具有相同属性和操作的对象，概括这些共同的方面以形成类。

SafeHome问题初步确定的对象类：

系统

显示窗口

控制功能键

传感器

警报器

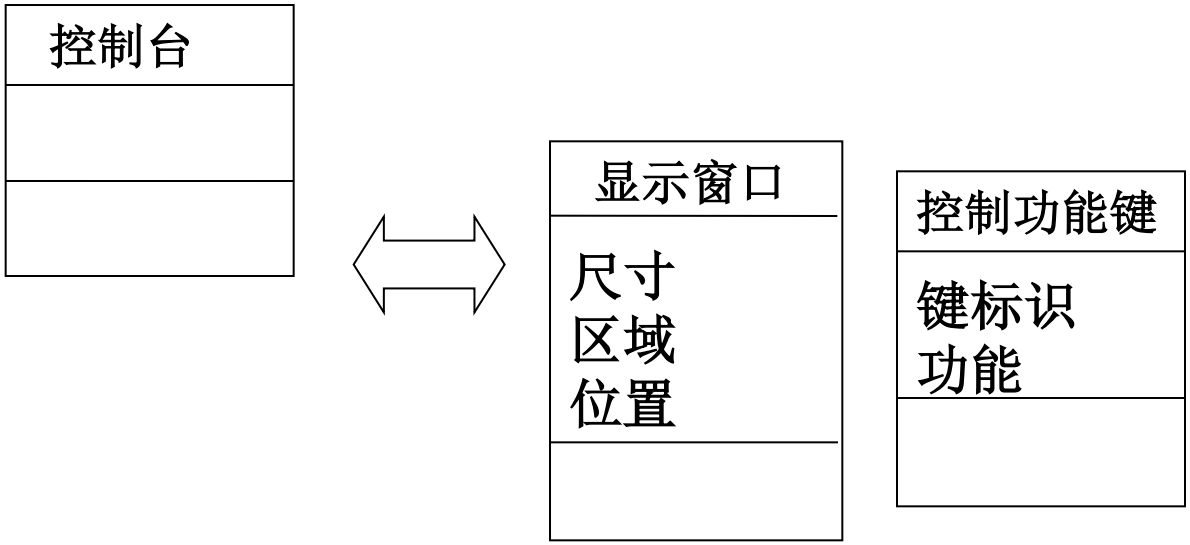
电话

传感器事件

对类进行调整: 划分

1. 如果类的属性或操作不适合该类的全部对象，则应考虑重新划分类。

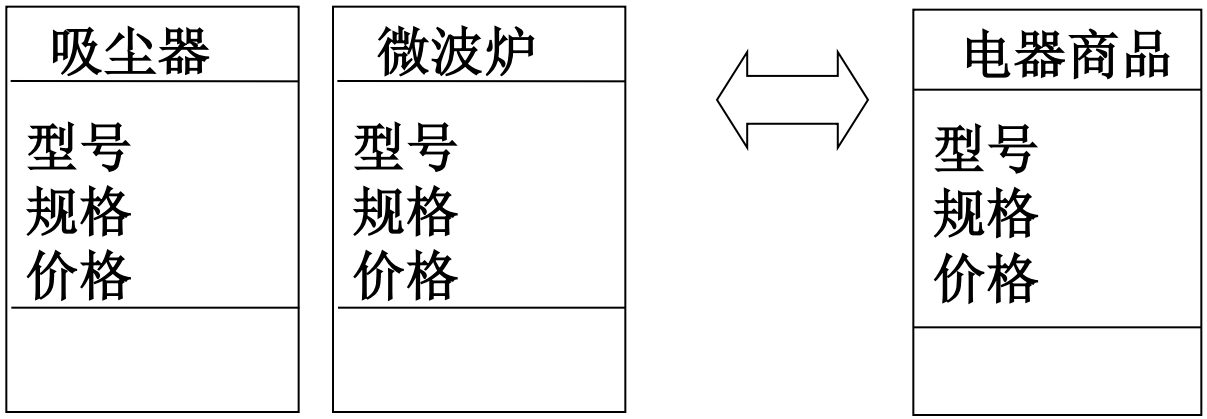
例如：在Safehome中，如果设置控制台类，则其中的属性不适用于窗口和控制键两者。



对类进行调整: 组合

2. 如果属性及操作都相同，即使不是同类，应该重新考虑，应该组织成为一个类。

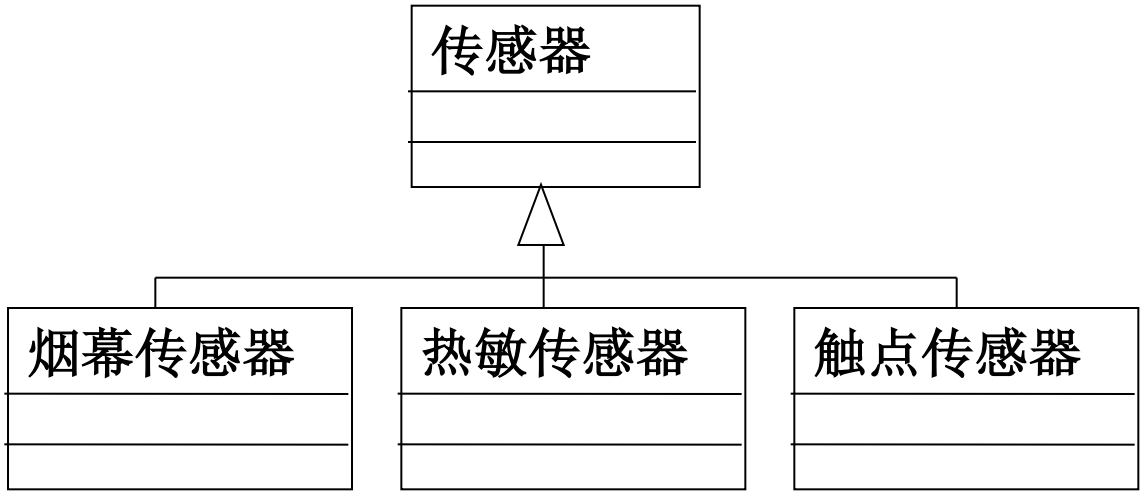
例如：微波炉和吸尘器，虽然不是相同的类，但在商店销售系统中，它们的属性和操作并无差别。



对类进行调整: 分层

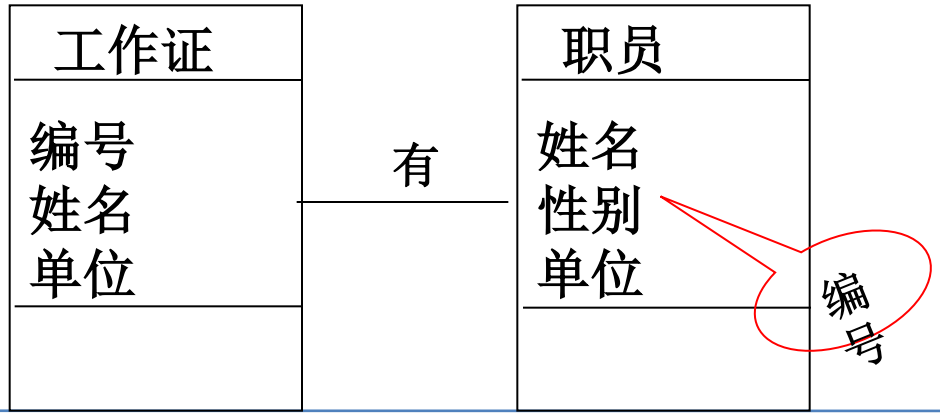
3. 如果属性及操作有很多相同的内容，则应该考虑使用继承，提取超类。

例如：在Safehome中，烟雾传感器、热敏传感器和触点传感器，其中很多属性和操作是相同的，可以用继承关联，提取传感器作为超类。



对类进行调整: 合并

4. 如果两个类的属性有重复，并且有附属关系，则可以考虑合并为一个主要的类。
例如：工作证类和职员类，如果在特定的问题中，工作证除了“编号”以外的属性，其他属性都与职员类属性一致，则可以考虑将“编号”合并到职员类中，去掉工作证类。



1.4. 主动对象和控制线程

认识主动行为对象的目的在于确定系统可独立并发的线程（进程），在设计时，将用于系统对象的分布和配置。

主动对象**定义：** 拥有线程（进程）并启动控制活动的对象。

线程： 一系列连续执行的操作序列构成线程。

一个进程有一个或者多个线程。

一个线程可以在一个进程中，也可以跨越多个进程。

主动类**定义：** 实例为主动对象的类对象。

识别主动对象的策略

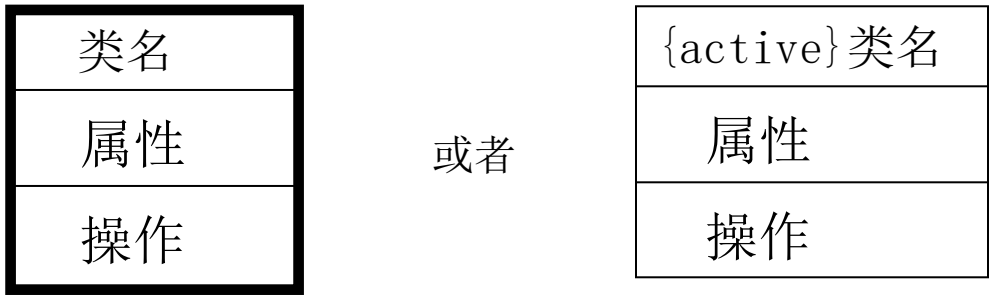
1）从问题域考虑，如果对象所具有的行为是主动发起的，而不是被动引发的行为。

2）考虑系统边界以外的参与者，与系统直接进行交互的责任对象。

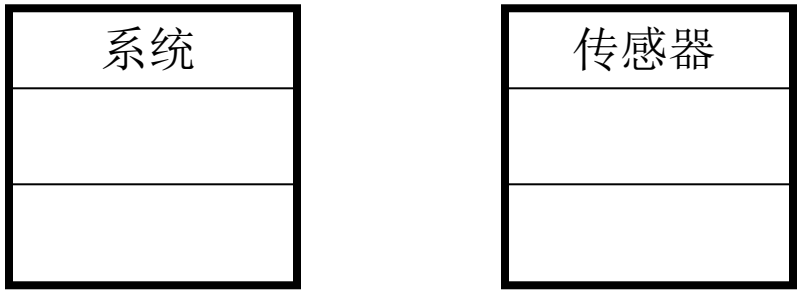
3）根据过程抽象的原则，从系统责任的角度，系统功能的外层往往是内层的驱动。

4）进行操作执行路线的逆向追踪，直到某操作不被其他成分请求为止。

主动对象类的特殊表示



Safehome中的两个主动对象类



面向对象分析与UML建模(3)

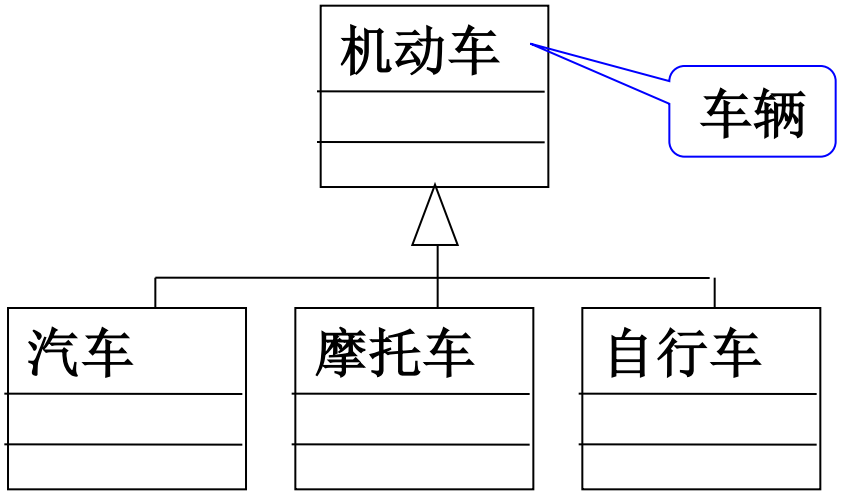
- 1. 面向对象建模
  - 1.1. 类图和对象建模
  - 1.2. 识别对象和筛选策略
  - 1.3. 从对象抽象类
  - 1.4. 主动对象和控制线程
- 2. 类的命名、属性和操作
  - 2.1. 类的命名
  - 2.2. 类属性及识别筛选策略
  - 2.3. 类操作及识别调整策略
- 3. 类的继承和关联
  - 3.1. 类的继承关系
  - 3.2. 建立类的关联
- 4. 接口类
- 5. 包图

2.1. 类的命名

类的命名应遵循的原则：

1）类的名字应适合类所包含的每一个对象，包括它的子类对象。

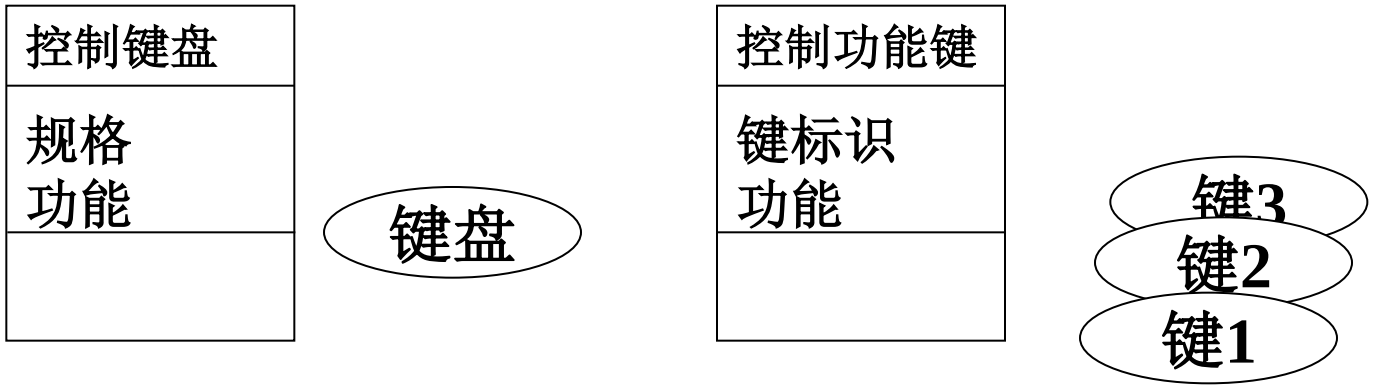
例如：机动车类，适合于摩托车和汽车；如果子类还有自行车，则应改为车辆类。



2.1. 类的命名

2）类的名字反映的是每个对象个体，而不是整个群体。

例如： Safehome中的控制键类，如果用控制键盘做类名，则每个实体就是一个控制键盘，而系统只需要一个控制键盘，实际需要的是对系统中每个功能键进行具体的描绘。



2.1. 类的命名

3）类名应采用名词或带定语的名词，并应注意行业规范的用语，不能使用无实际意义的数字或字符作为类名。

例如：定语名词“线装书”；“出租车”类名，不要用“面的”命名类；在某化学分析软件中，用“碳酸钙沉积岩”做类名，不要用“大理石”来命名。

4）使用适当的语言文字命名类，无论哪种文字，从编程的角度，都应该标注英文符号对照。

2.2. 类属性及识别筛选策略

类的实例属性和类属性

实例属性定义：类的构成元素，用于描述类所对应的事物的一个性质。

- 实例属性与问题域和系统责任紧密相关；它并不是对象具有的全部属性
- 每个实例属性有唯一的名字，其值属于给定的类型
- 实例属性具有可见性

2.2. 类属性及识别筛选策略

类属性定义：类属性是描述一个类的所有对象的共同性质的一个数据项，对于该类的任何对象，它的属性值是相同的。

类属性，对一个类的全部对象只有一份共同的数据空间；在C++中可以用static静态成员说明符标出。

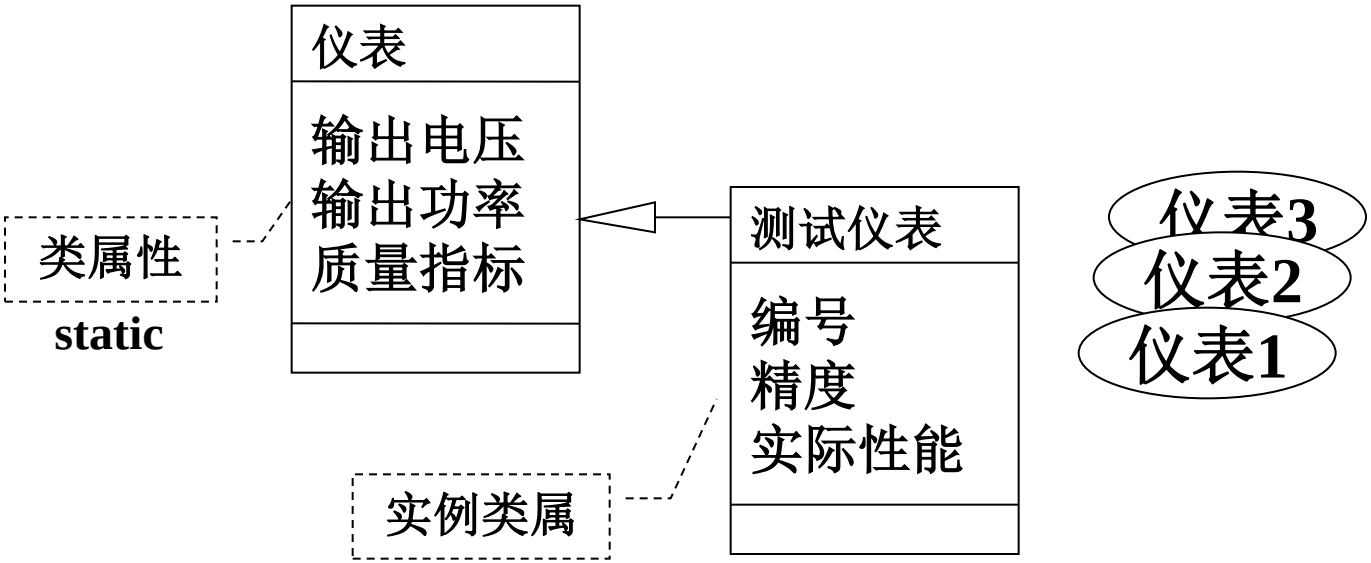
实例属性和类属性的区别和作用：

- 实例属性----用于不同性质的同种个体。
- 类属性------用于相同性质的同种个体。

2.2. 属性及识别筛选策略

注意：在特定的问题中，有些个体同时具备某些不同的性质和共同的性质。

例如：一个仪表产品测试系统中，每台仪表的输入电压、功率及质量指标都是相同的，但每台仪表的编号、精度和实际达到的性能值却不同。



2.2.识别属性的启发性策略

1）根据常识确定对象应用的基本属性；

例如：

传感器属性，包括编号、类型、临界值、……

2）根据问题域，确定对象应该有的属性；

例如：

传感器属性，在Safehome系统中,需要安装地点、性质。同样传感器属性，在设备管理系统中，需要数量、购置时间。

2.2.识别属性的启发性策略

3）根据系统责任，确定对象应该有的属性；
例如：
　　报警器的属性，在Safehome系统中，有自动拨打电话的责任，因此需要延时时间属性

4）根据对象需要状态，确定对象可能的属性
例如：
　　传感器属性，在Safehome系统中,有被设置在线和撤消的状态，则需要有在线状态属性

2.2.识别属性的启发性策略

5）问题陈述中定语的词汇，可帮助确定类的属性。

例如：

“绿色的按钮”，可以确定按钮类有颜色属性。

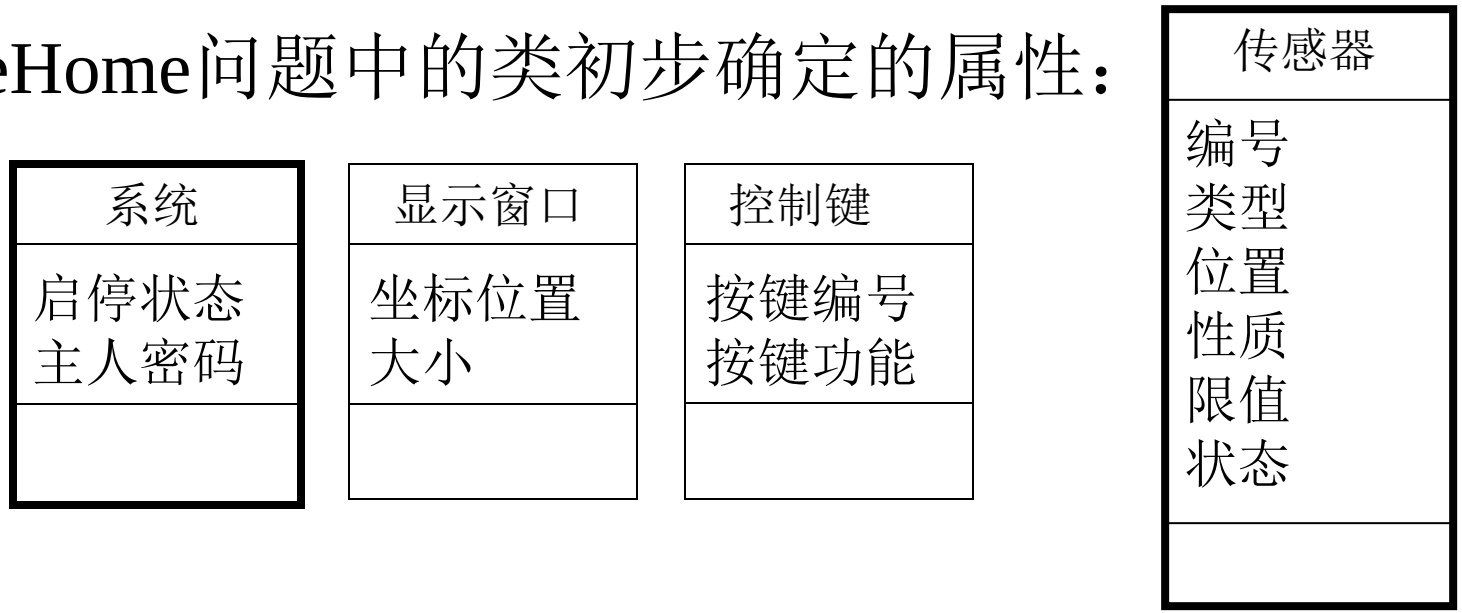
6）类间的关联与聚合是类的特殊属性，它表示类的一个属性值是另一个类的实例。注意：不用属性名表示

例如：某学生的指导教师，用学生类和教师类的关联。

- 控制台的输出显示窗口，用控制台类与输出显示窗口类的聚合关系。

识别属性示例

根据SafeHome问题中的类初步确定的属性：

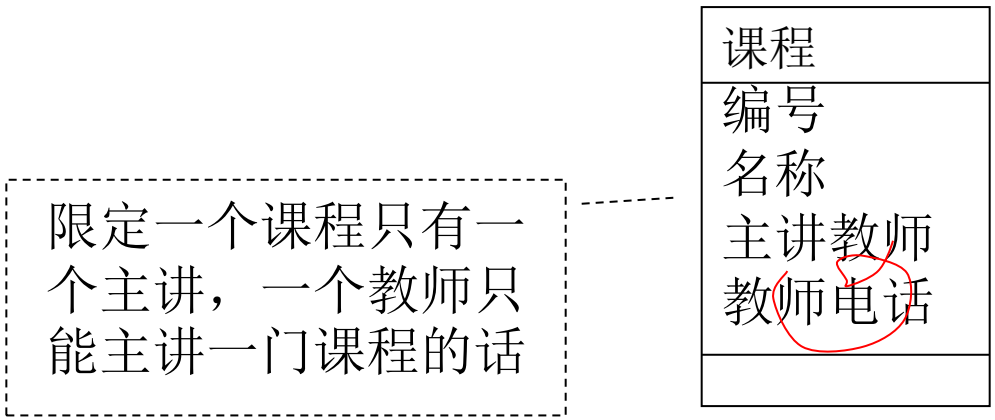


筛选类的属性

对于初步确定的属性进行筛选的策略：

1）对象的属性要描述对象本身固有的性质，否则即使它在系统中提供有用的信息，也不应作为对象的属性。

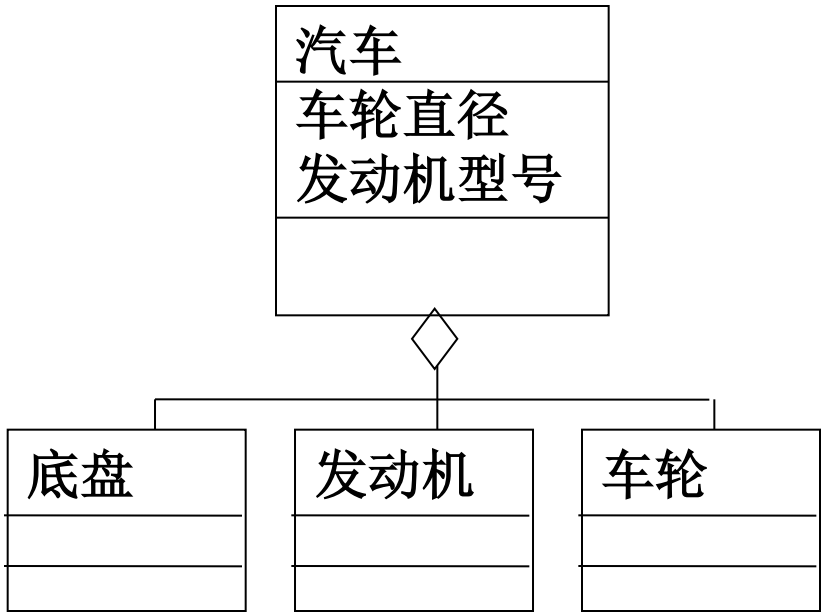
例如：课程类有主讲教师属性，把主讲教师的电话、住址等作为课程的属性会造成概念上的混乱。



筛选类的属性

2）在类间存在有聚合关系时，整体对象类的属性不要包括部分对象类的属性。

例如：汽车由底盘、发动机、车轮和驾驶仓等组成，汽车的属性与组成各部分的属性要分开，不要包含在汽车中。



筛选类的属性

3）属性应按一般的思维习惯，采用原子的（即不可分的）概念。

例如：

房间的属性门窗，应该具体分为门和窗；
犯罪嫌疑人的属性服装，应该具体分为上衣和裤子。

4）一般类定义的属性，在特殊类中不重复出现。

筛选类的属性

5）一个属性值明显可以从另一个属性值直接导出，则应该去掉。但是如果需要较复杂的计算才能导出，则可以考虑保留。

例如：有出生年月属性，不必保留年龄属性；但是，有各分项税额属性，也可以有总税率属性。

2.3. 类操作及识别调整策略

操作定义 操作是类的构成元素，是类的对象被要求执行的服务。

- 内部操作和外部操作：
  内部操作---只供对象内部的其他操作使用。
  外部操作---响应其他对象请求时提供的服务。

- 系统行为和自身行为：
  系统行为---系统施加于对象的行为。
  例如：对象创建、复制、存储、删除
  自身行为---简单算法行为：简单读写属性的值。
  复杂算法行为：完成对象固有的行为算法。

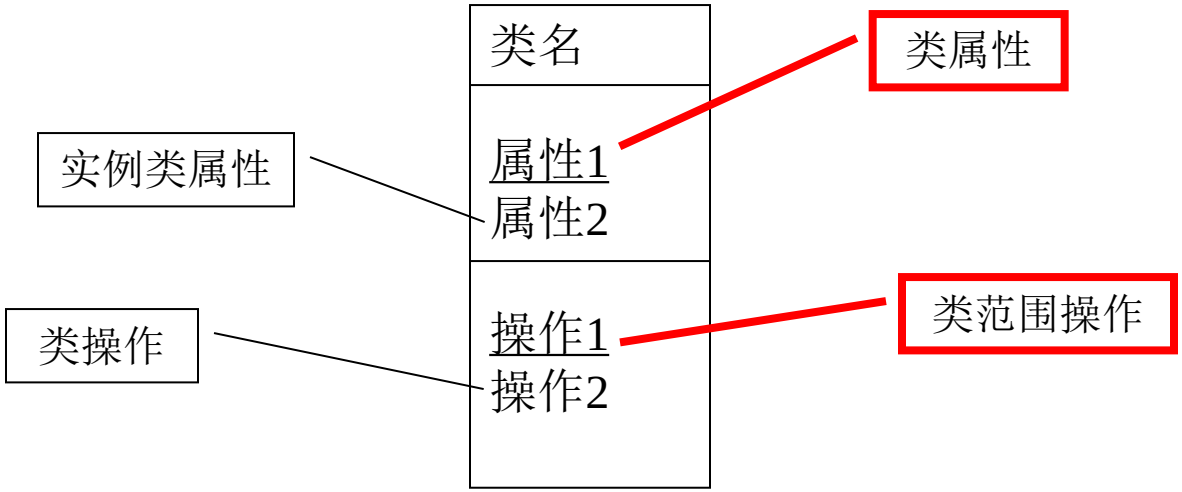
- 分析模型应该以描述复杂算法操作为主。

类范围操作定义

仅用于操纵类属性的操作叫做类范围操作，其余的操作叫做实例范围的操作。

类属性，对一个类的全部对象只有一份共同的数据空间；类范围操作，仅对这样的共同空间数据进行处理。

类属性和类范围操作表示：



对象操作的发现策略

① 问题陈述中的动词或动词短语

② 考虑系统责任----从需求中的每项功能，寻找可对应对象的操作。

③ 考虑问题域----从对象在问题域中的行为，寻找与系统责任有关的对应的操作。

④ 从属性值的设置----对象的属性值必须由对应的具体操作来设置。

⑤ 追踪操作轨迹----从特定场景的对象间交互，寻找对象必须提供的服务响应行为。

⑥ 分析对象状态----从对象生命周期呈现的每种状态，寻找对象保持状态的必然行为。

对象操作示例

根据SafeHome问题中的类及属性初步确定的操作：

系统
启停状态 主人密码
系统启动 密码设置 ……

显示窗口
坐标位置 大小
按坐标位置显示信息 按大小打开窗口 ……

功能键
按键编号 按键功能
功能键操作处理1 功能键操作处理2 ……

传感器
编号 类型 位置 性质 限值
判断限值

警报器
启停状态 延时限制
根据延时限定报警 ……

电话
电话号码
自动拨号处理 ……

传感器事件
事件时间
事件查询 ……

类操作的调整原则

1）如果一个操作没有系统或其他部分的请求（包括外部系统和参与者的请求），则是无用的操作，应该丢弃。

2）操作应该是高内聚的。如果一个操作不仅是完成一项明确定义的、完整的、单一功能，则应该分解该操作。

类操作的调整原则

3）对象的操作,应该反映问题域中实际事物的行为，所以必须放在相应的类中。
例如：售货操作，不是“货物”上的操作，而是售货员的操作

4）操作的命名，应该是动词或动宾结构，应尽量清楚地反映操作的内容。

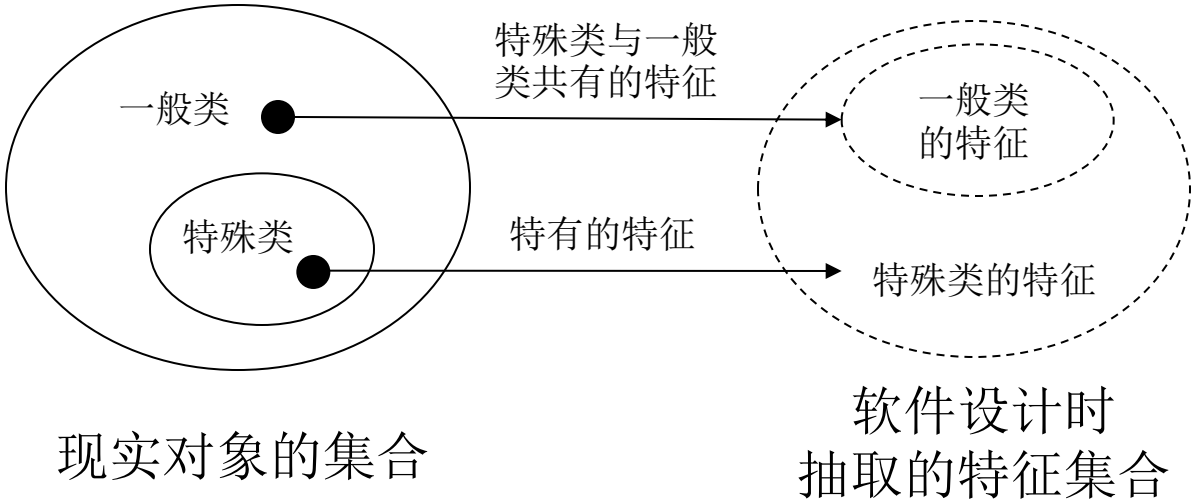
面向对象分析与UML建模(3)

- 1. 面向对象建模
  - 1.1. 类图和对象建模
  - 1.2. 识别对象和筛选策略
  - 1.3. 从对象抽象类
  - 1.4. 主动对象和控制线程
- 2. 类的命名、属性和操作
  - 2.1. 类的命名
  - 2.2. 类属性及识别筛选策略
  - 2.3. 类操作及识别调整策略
- 3. 类的继承和关联
  - 3.1. 类的继承关系
  - 3.2. 建立类的关联
- 4. 接口类
- 5. 包图

3.1. 类的继承关系

继承关系定义 如果类A具有类B的全部属性和全部操作，而且还具有自己特有的一些属性或操作，则A叫做B的特殊类，B叫做A的一般类，A与B之间的关系称为继承关系。

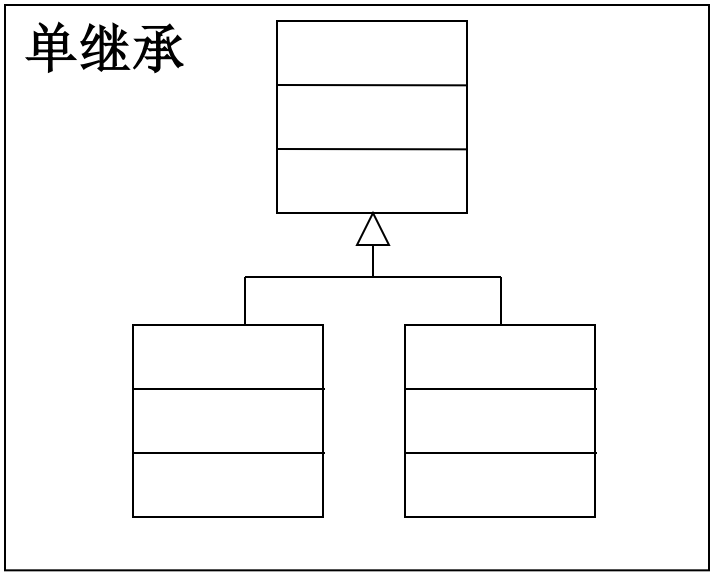
一般类与特殊类

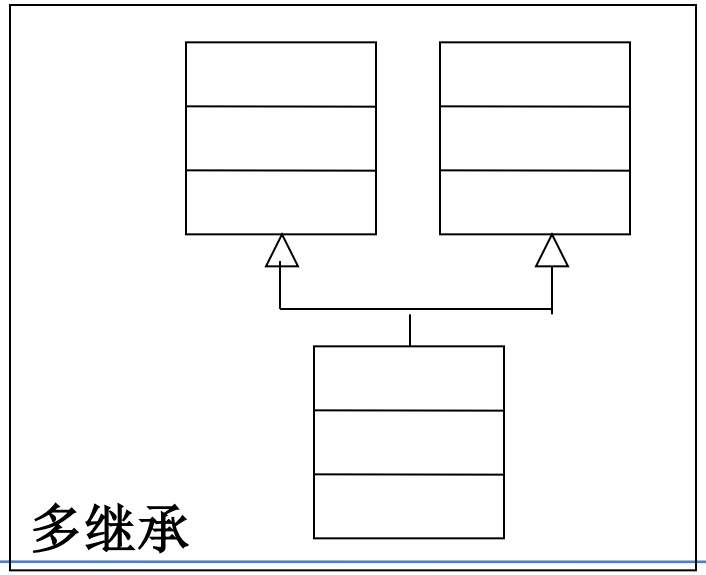


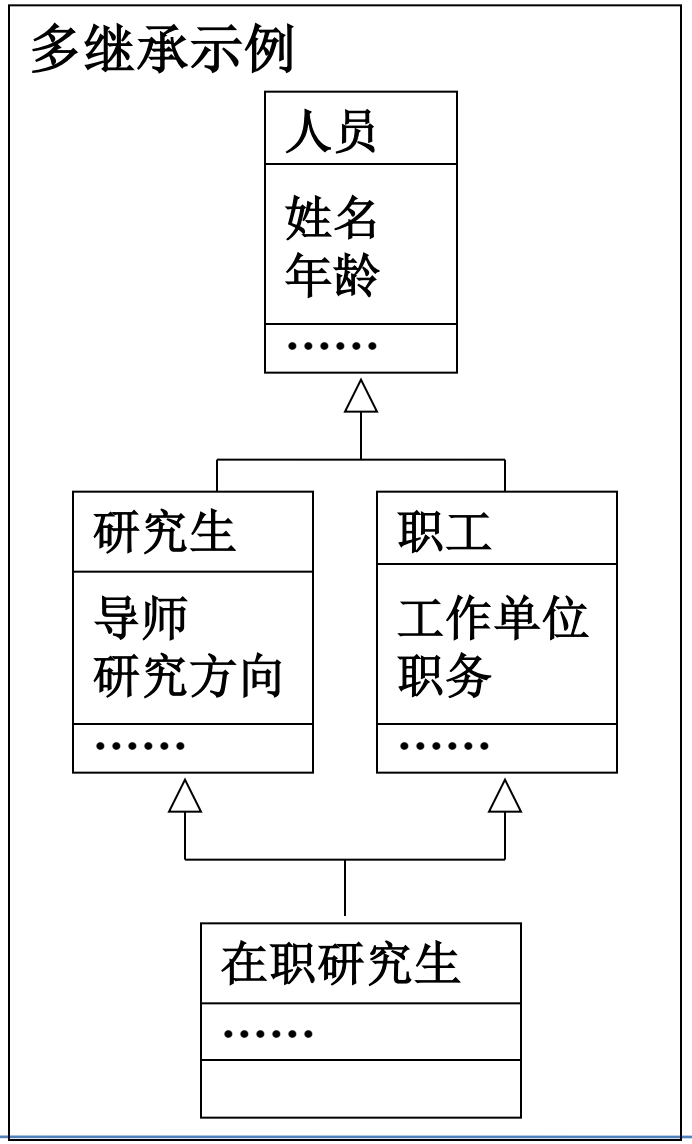
继承的性质

- 非对称性----继承关系语义为“is a”关系，如果类A是类B的后代，则类B不能是类A的后代，因为，B将不会与类A有“is a”关系。
- 传递性----如果类A继承类B，类B继承类C，则类A继承类C。
- 单继承----特殊类只直接地继承一个一般类
- 多继承----允许特殊类可直接地继承两个以上的一般类

继承关系表示

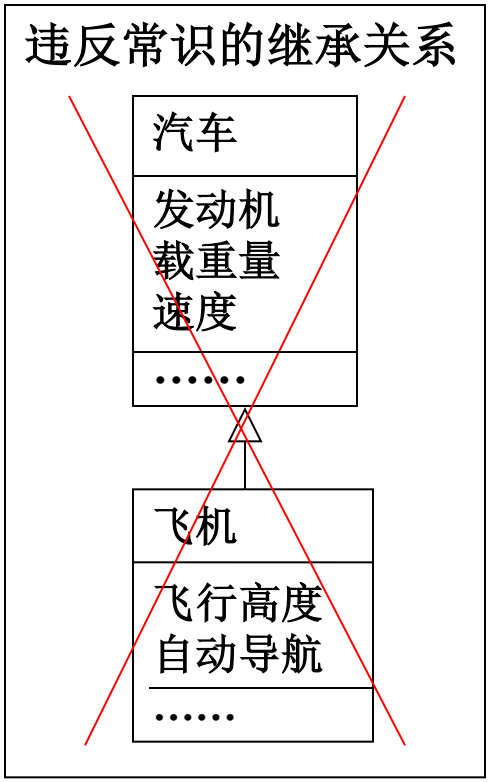






确定继承的策略和原则

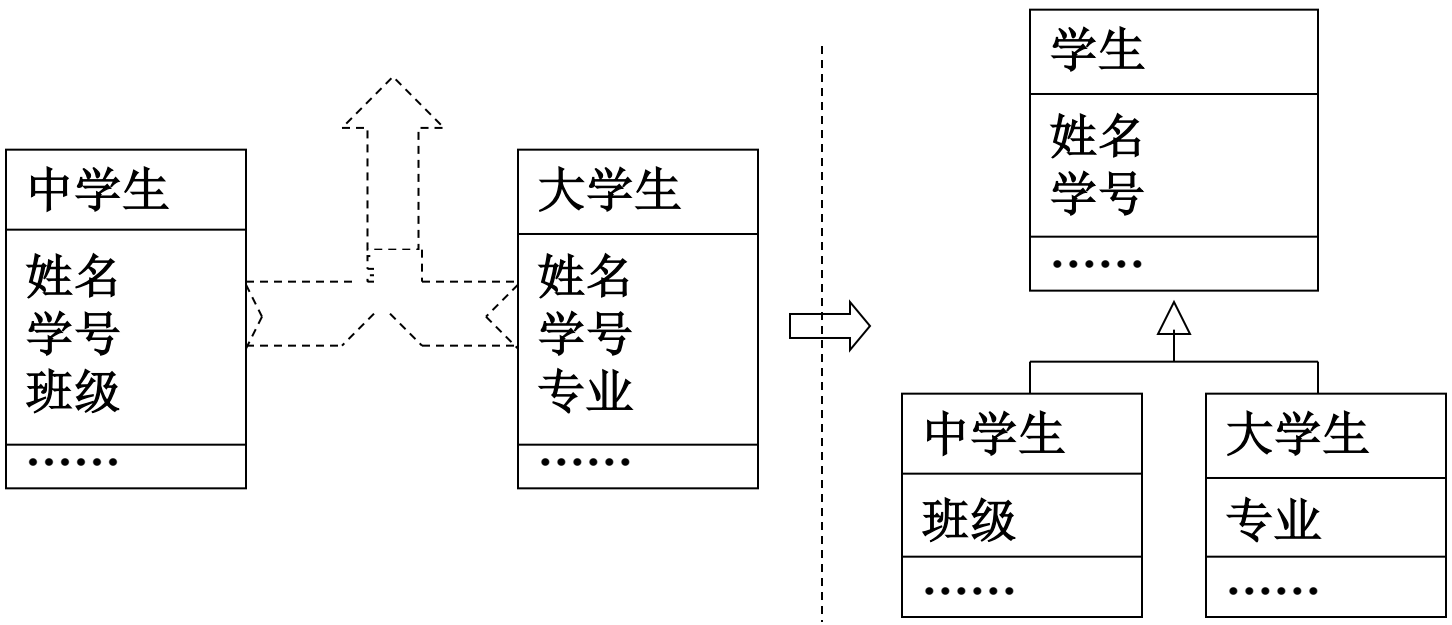
1）按照问题领域的分类知识以及事物的分类常识，寻找相应的继承关系。



确定继承的策略和原则

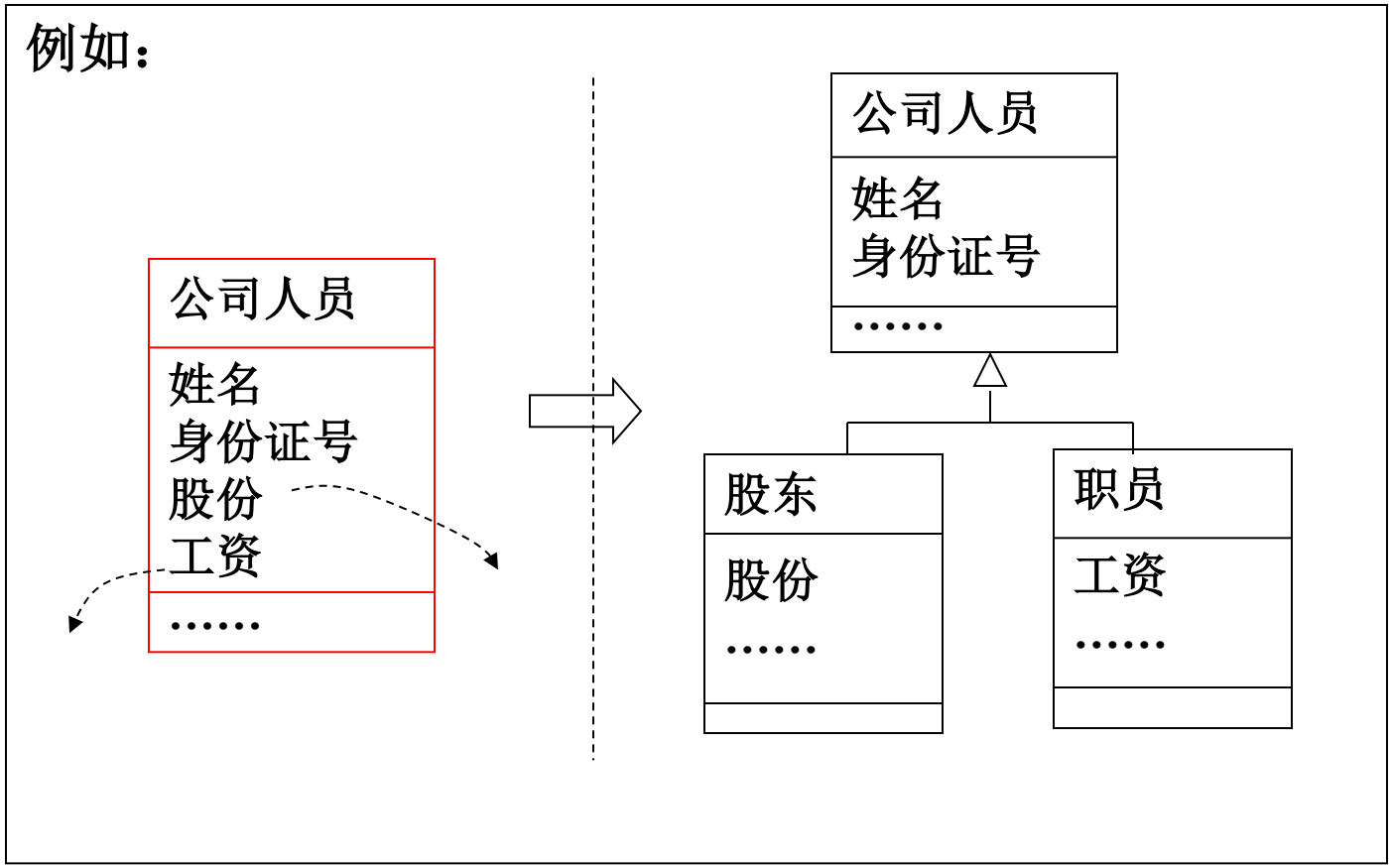
2）寻找类之间的包含关系。如果一个类是另一个类的子集，并且，类之间的语义有“is a”的关系，则它们是继承关系。

例如：



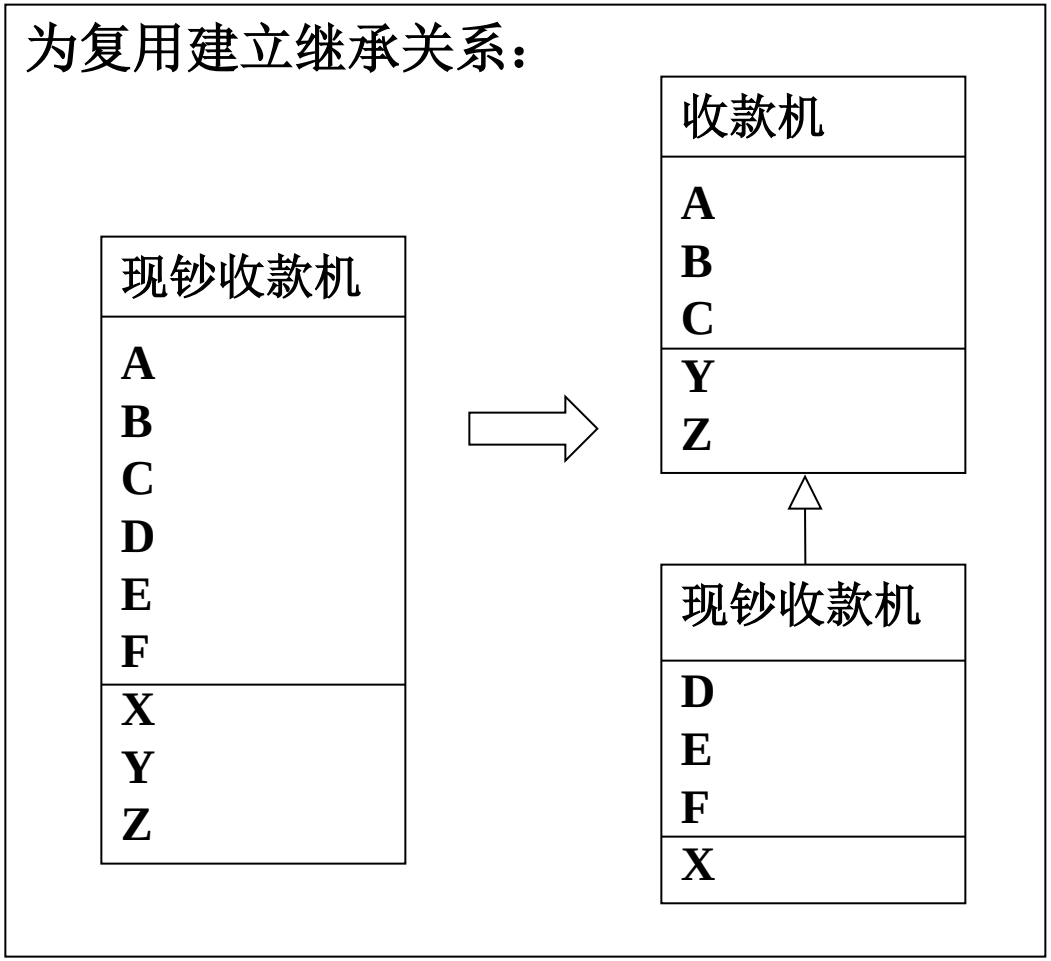
确定继承的策略和原则

3）判断类的属性与操作是否适合于所有的对象，从中发现特殊类，建立继承关系。



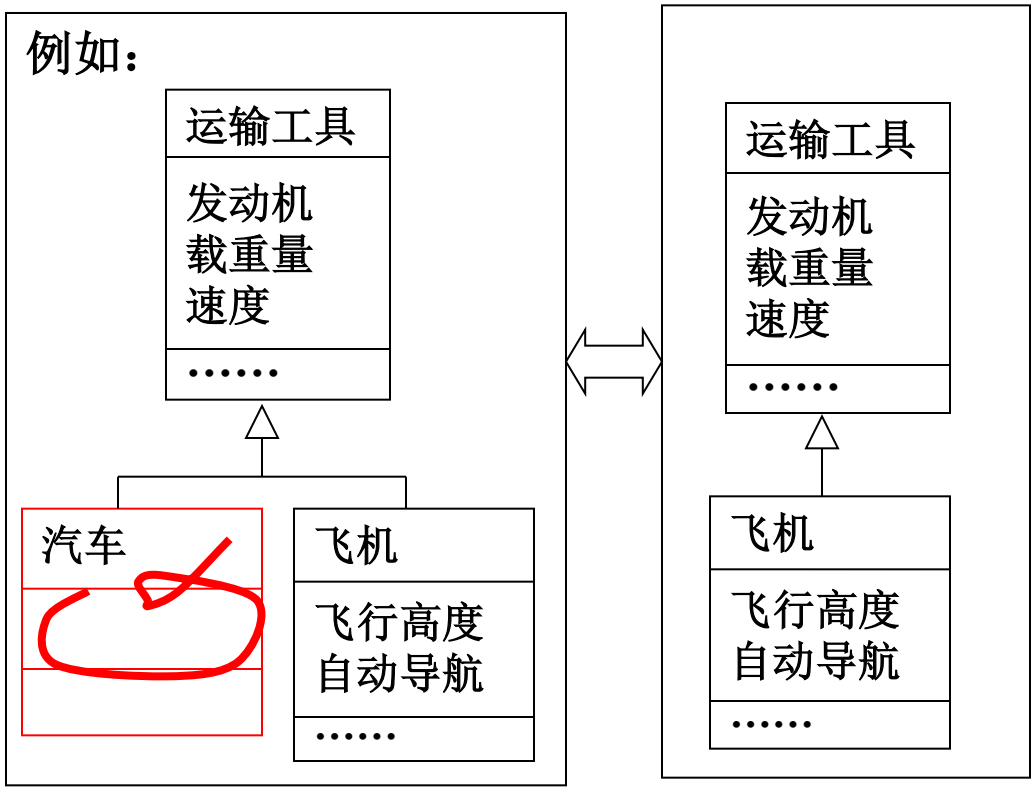
确定继承的策略和原则

4）为适应领域范围内更多的复用，分解类为继承关系。



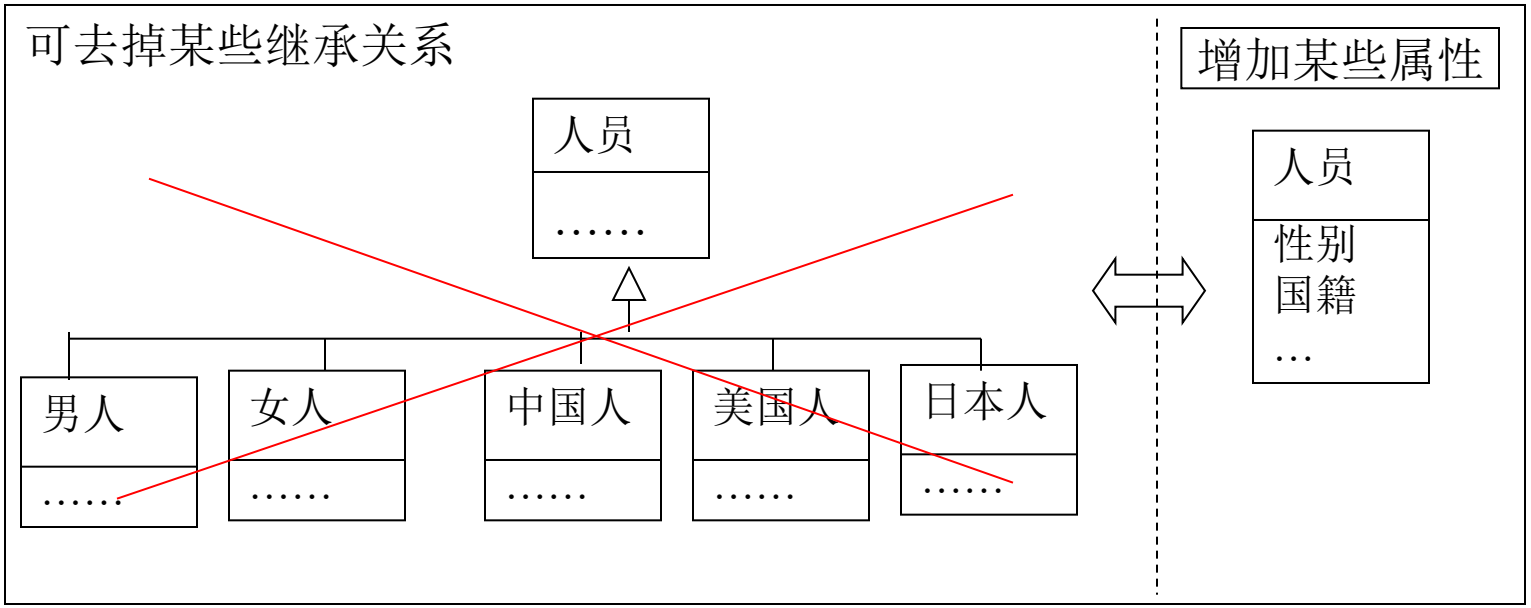
确定继承的策略和原则

5）若特殊类不具有自己特殊的属性和操作，则可以去掉该特殊类。



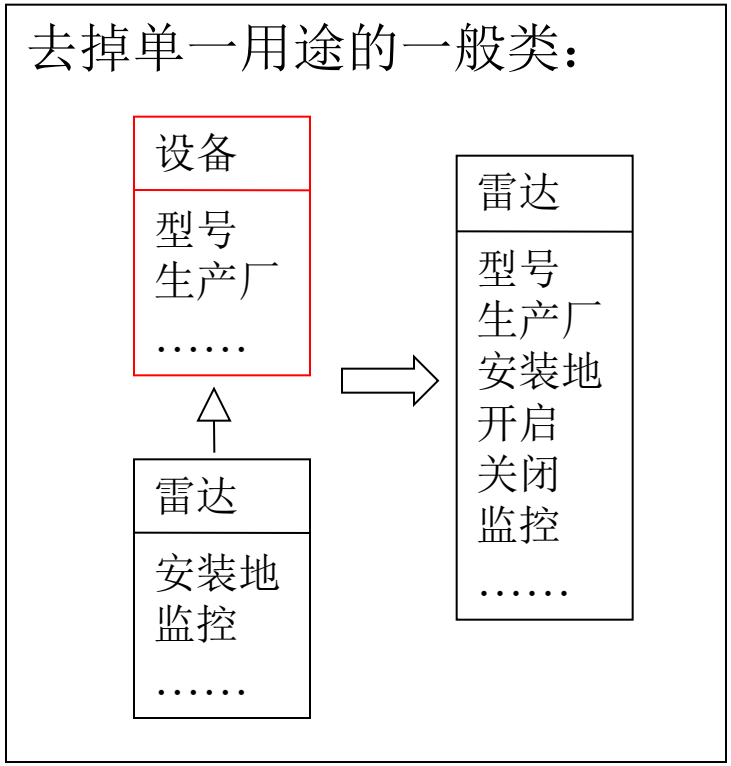
确定继承的策略和原则

6）如果某些特殊类之间的差别，可以通过某属性值来体现，则可以去掉这些特殊类。



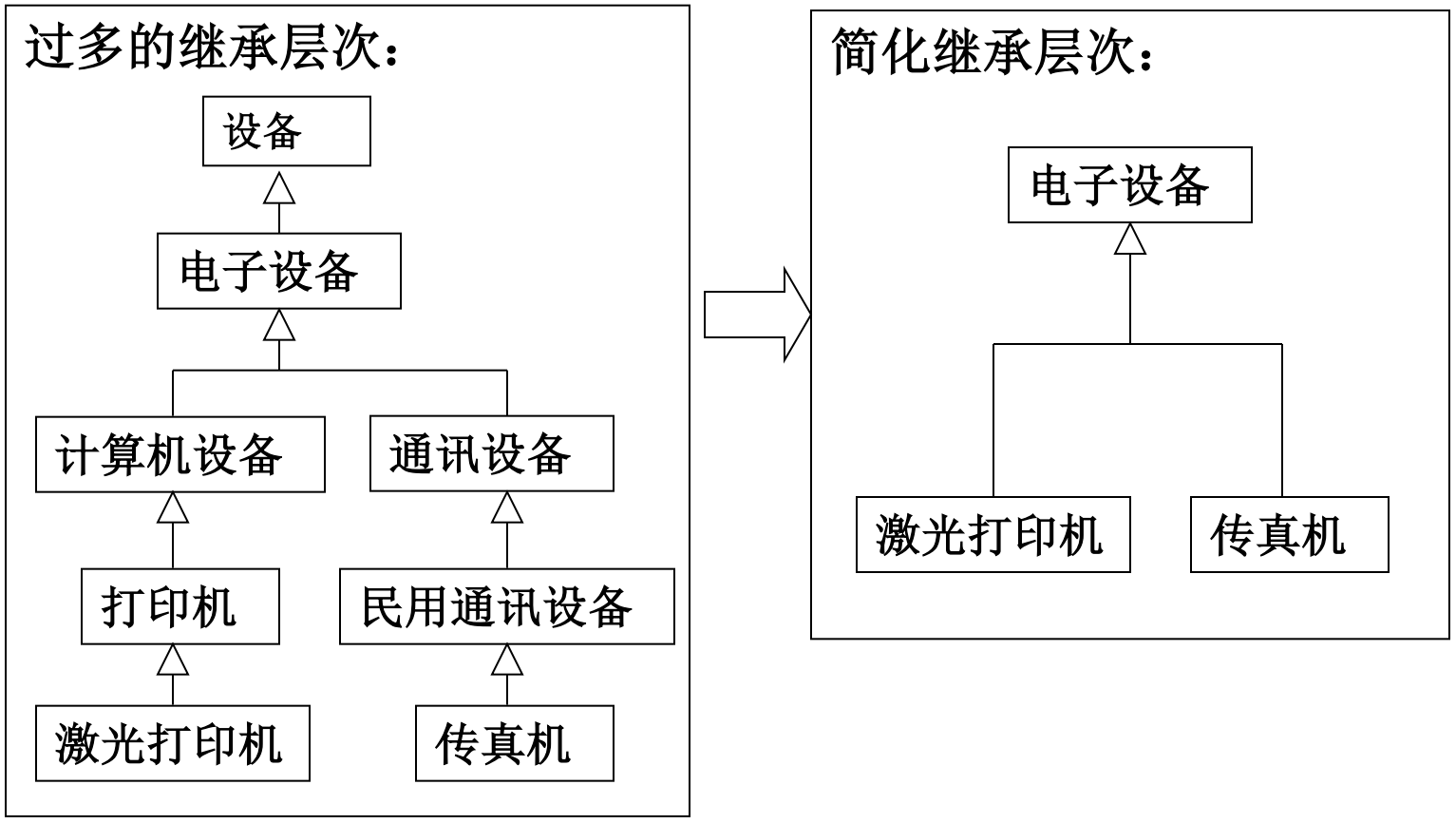
确定继承的策略和原则

7）如果只有唯一特殊类，且该一般类不用于创建对象，也不用于复用，则取消该一般类。



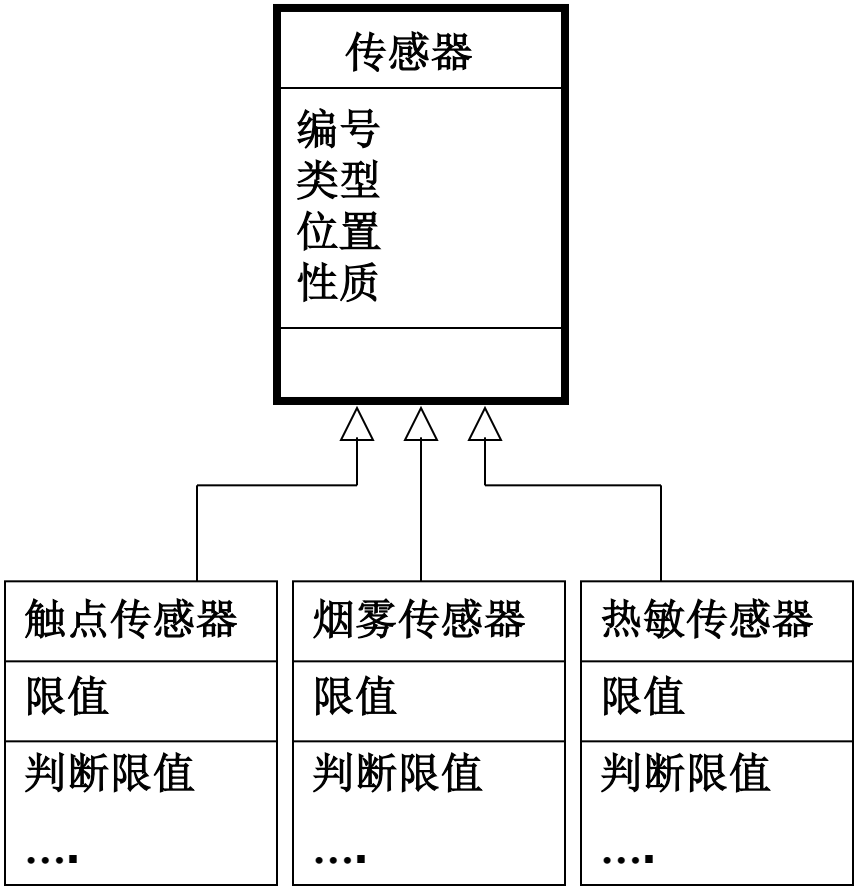
确定继承的策略和原则

8）应该尽量避免类的层次过深，使系统结构过于复杂。



示例

Safehome中传感器继承关系



3.2. 建立类的关联

- 类的关联
- 关联的重数
- 对象链
- 关联角色
- 关联类
- N元关联
- 关联的限定符
- 聚合关联
- 类的依赖关系表示

(1)类的关联

关联的基本概念：

关联-----表示对象类之间的静态关系。
静态关系----表示对象之间固有的联系。

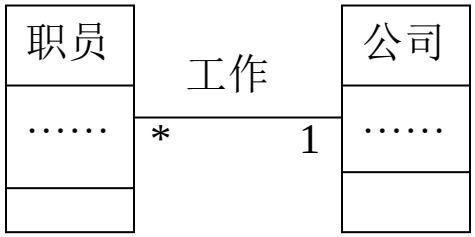
要点：

- 两个类之间的关联，实质上是通过属性来表示对象之间的联系，即一个类的对象属性值是另一个类的对象实例（用指针实现）。
- 静态关联与系统责任相关，如果这些关系是系统责任目标的必要信息，则需要表达它们。

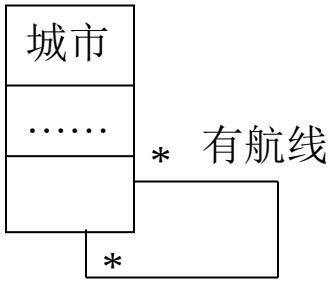
(1)类的关联

关联定义 如果一个类的对象与另一个类的对象之间有语义连接关系，则这两个类之间的语义关联就是关联。

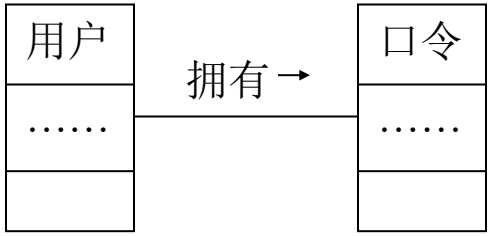
双向关联例：



自身关联例：



单向关联例：

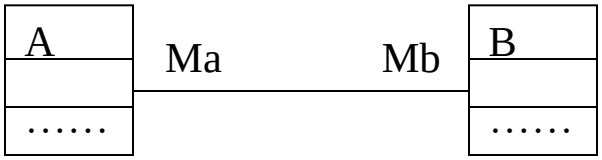


关联一般是双向的，若有单向限制用箭头表示。

(2) 关联的重数

重数定义 关联端上的重数，表示一端的对象需要另一端对象的个数。

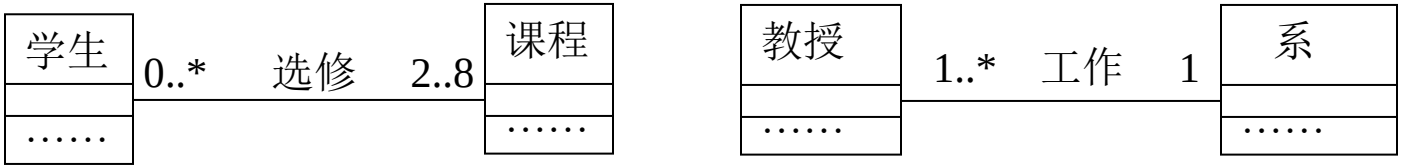
重数的表示：A端的一个对象需要B端Mb个对象
B端的一个对象需要A端Ma个对象



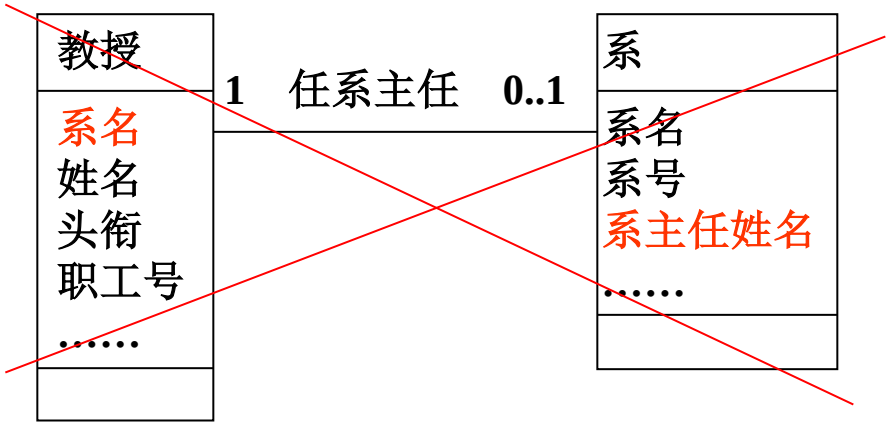
重数值的表示：

值	表 示
n ..m	表示与类的n至m个对象关联

例：

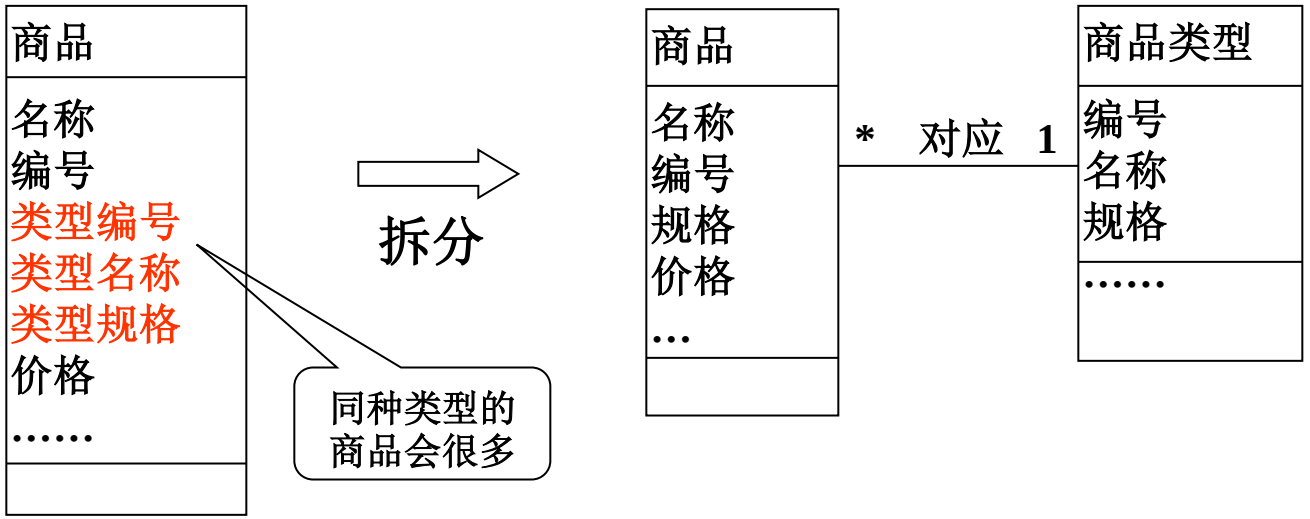


在类中不用表示关联的具体项（外键）。

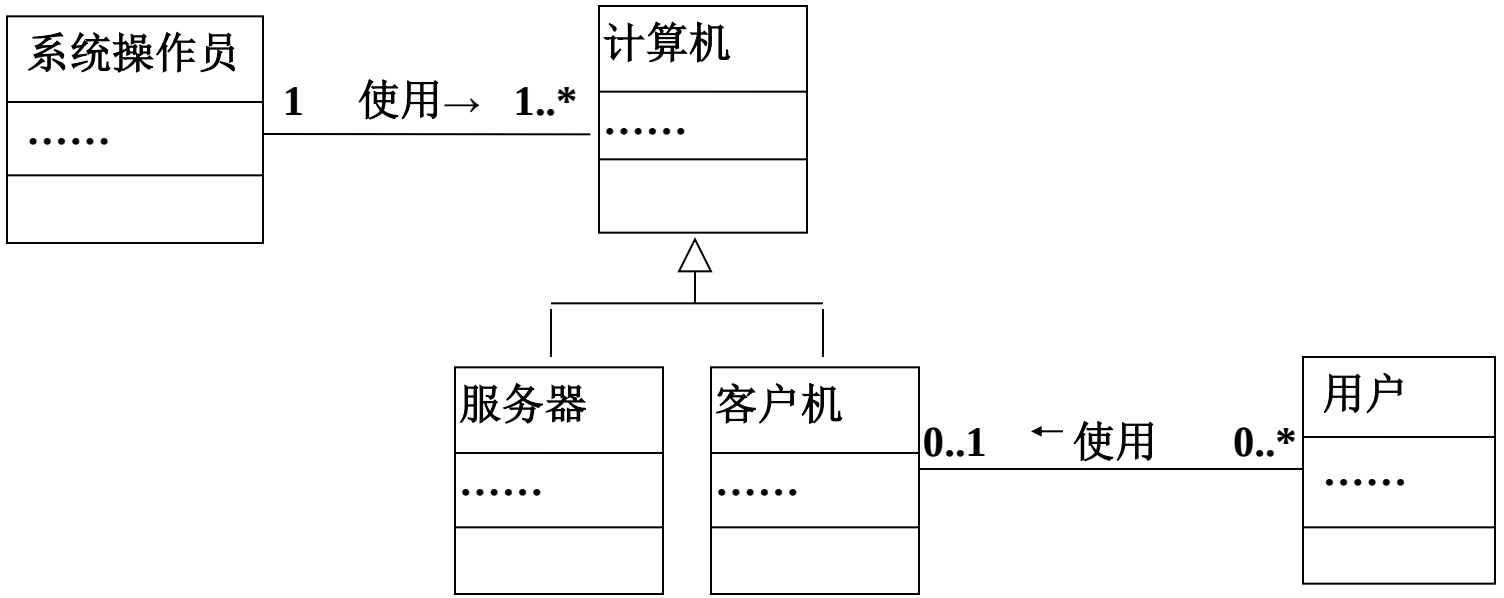


有数据冗余的类，可拆分类并关联它们。

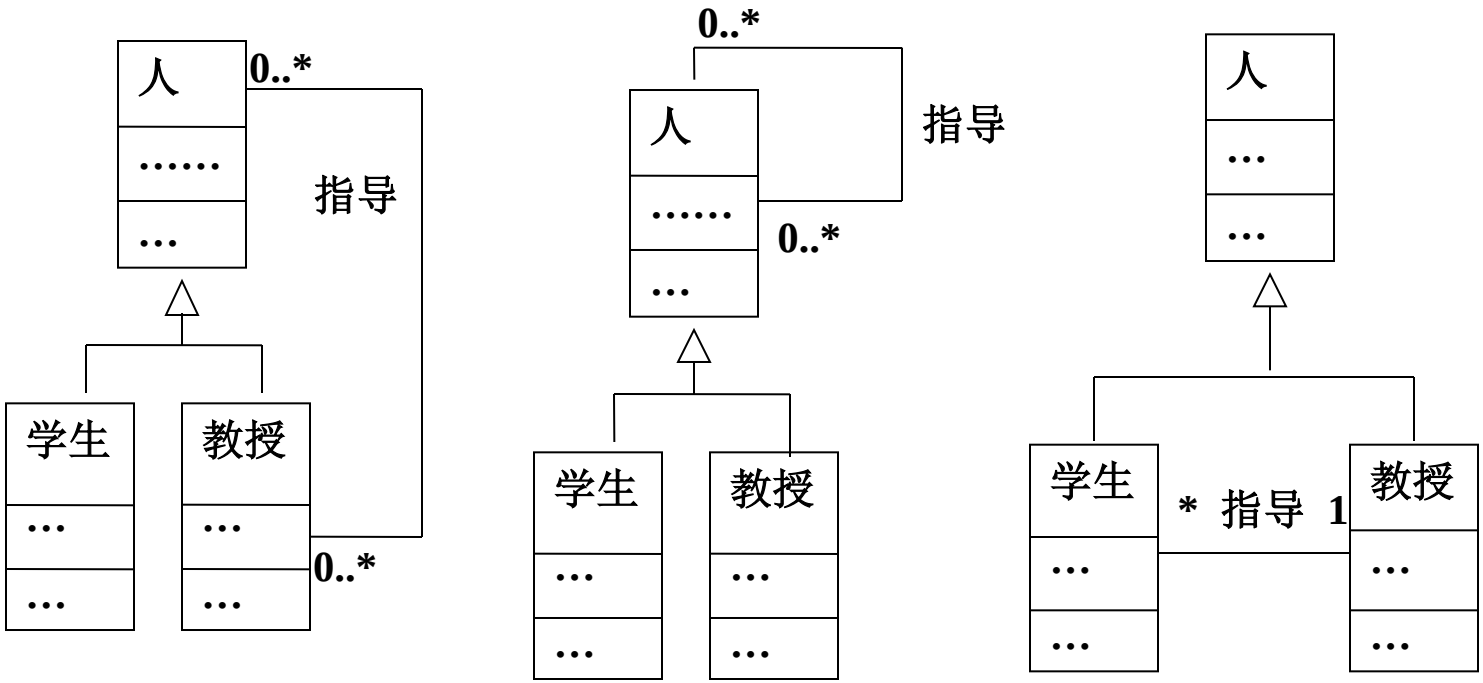
例：



特殊类直接继承一般类的关联：

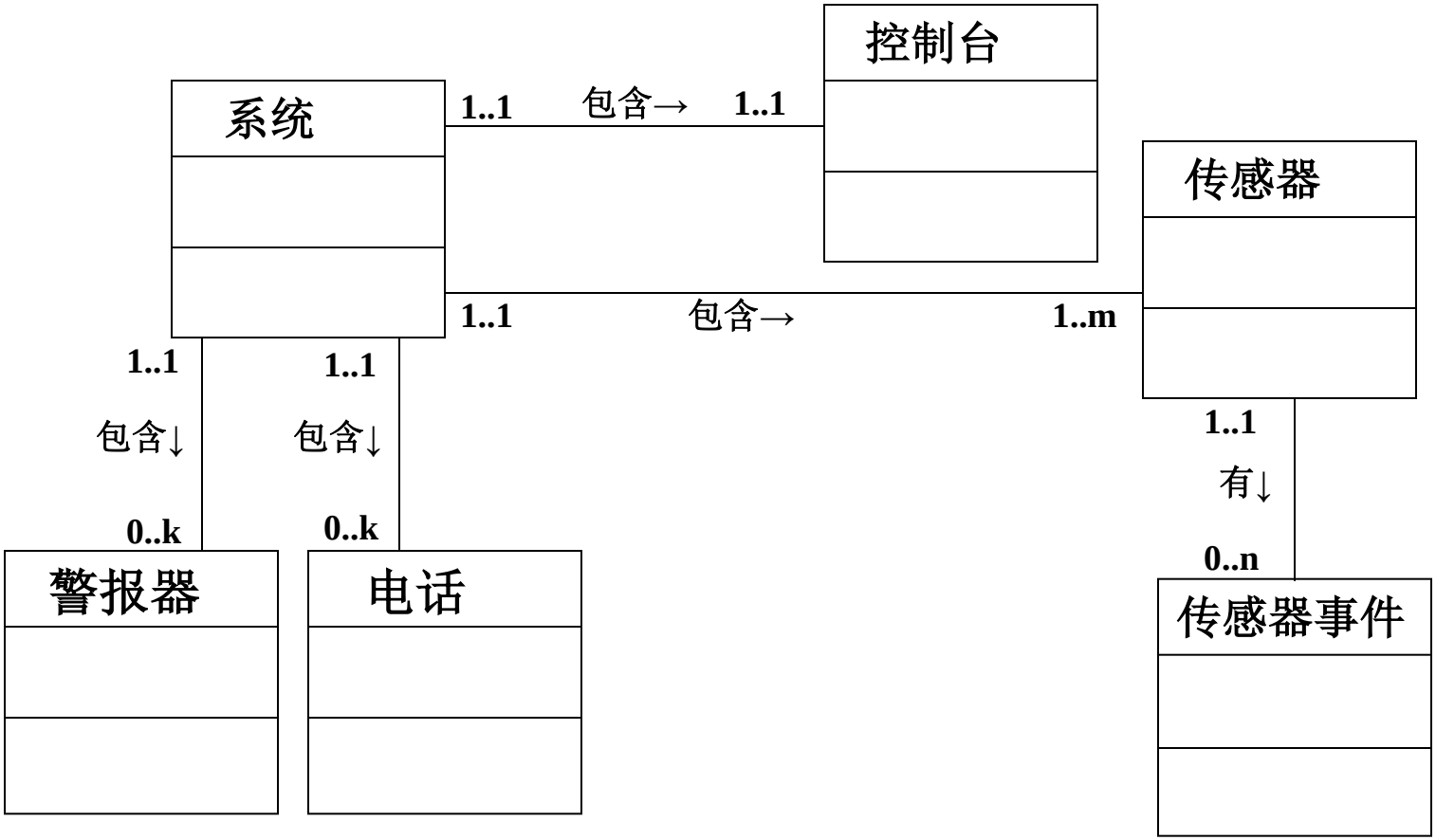


同样的类，由于关联不同，可以有不同的表示：



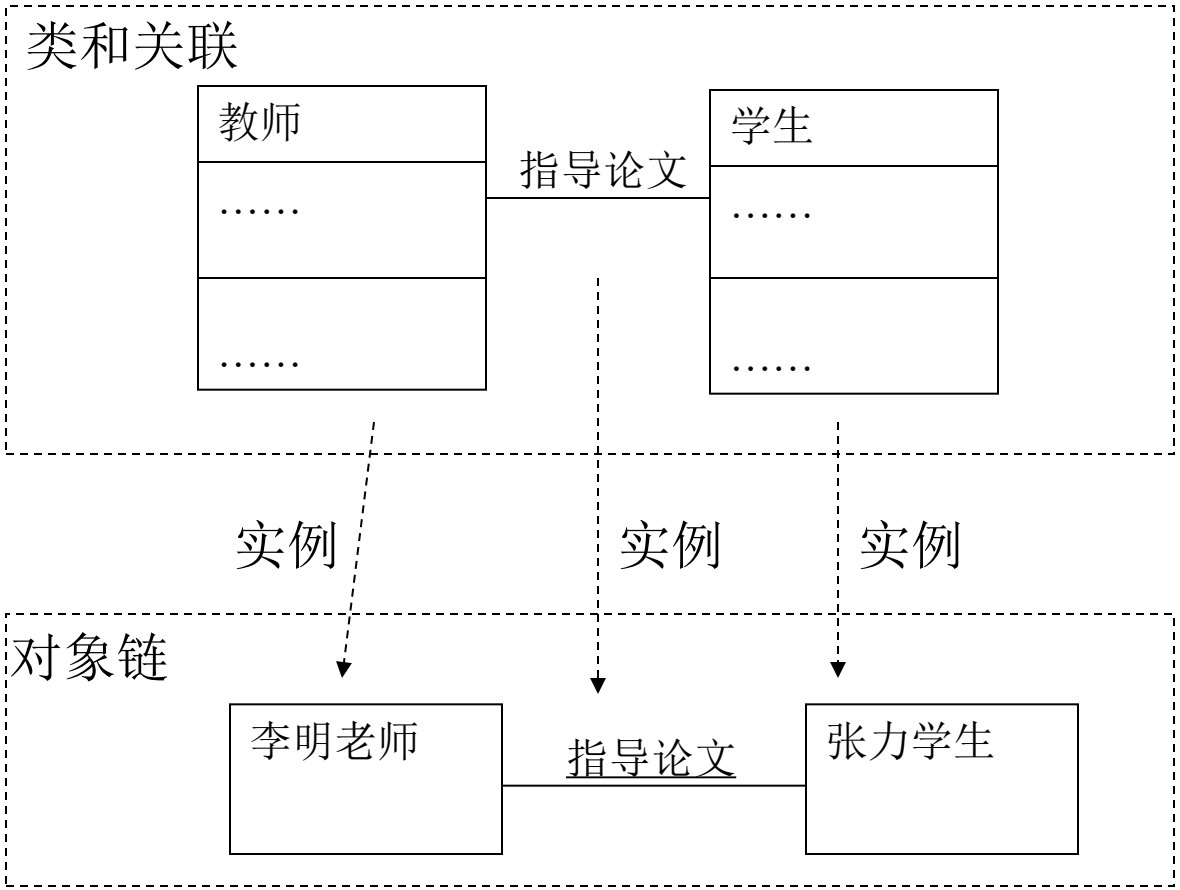
关联实例

Safehome系统的类图及重数表示



(3)对象链

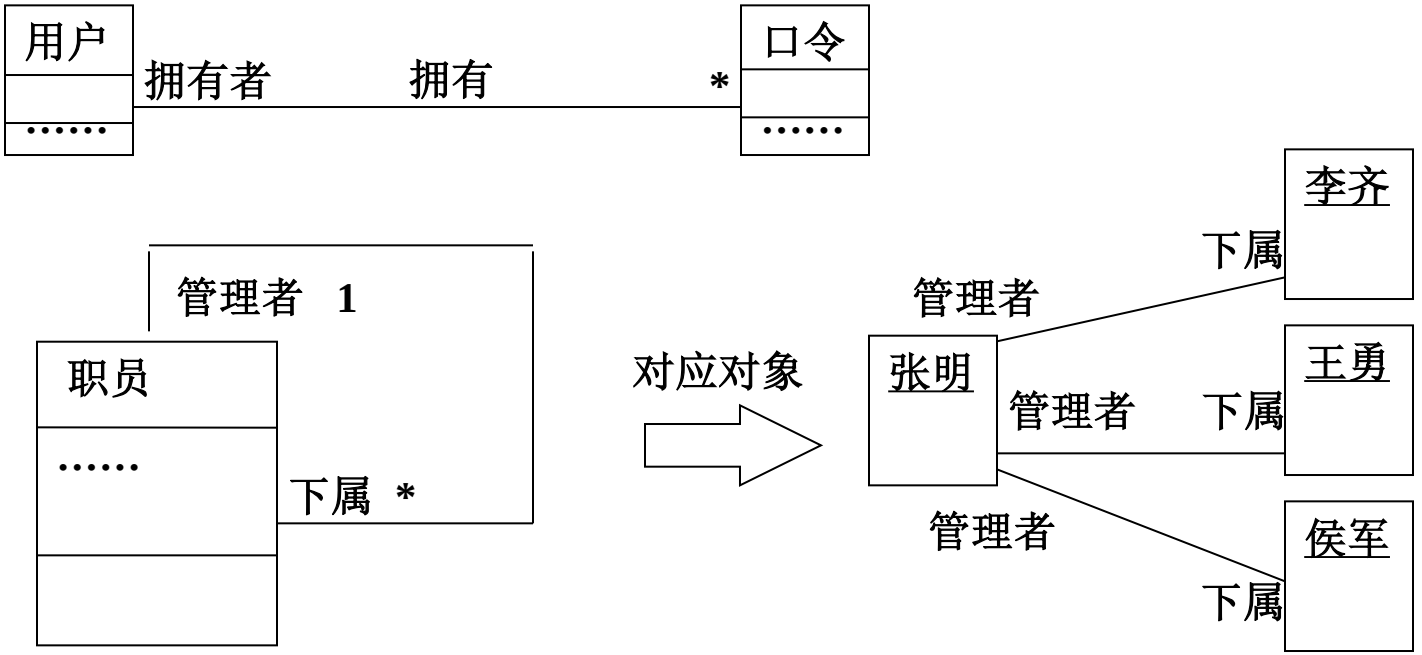
链定义 链是关联的实例，是对象间的语义连接。



(4)关联角色

关联角色----表示需要明确的一个角色属性。在关联的端点，可以表示相连接类所扮演的角色，关联角色的名字称为“关联角色名”。

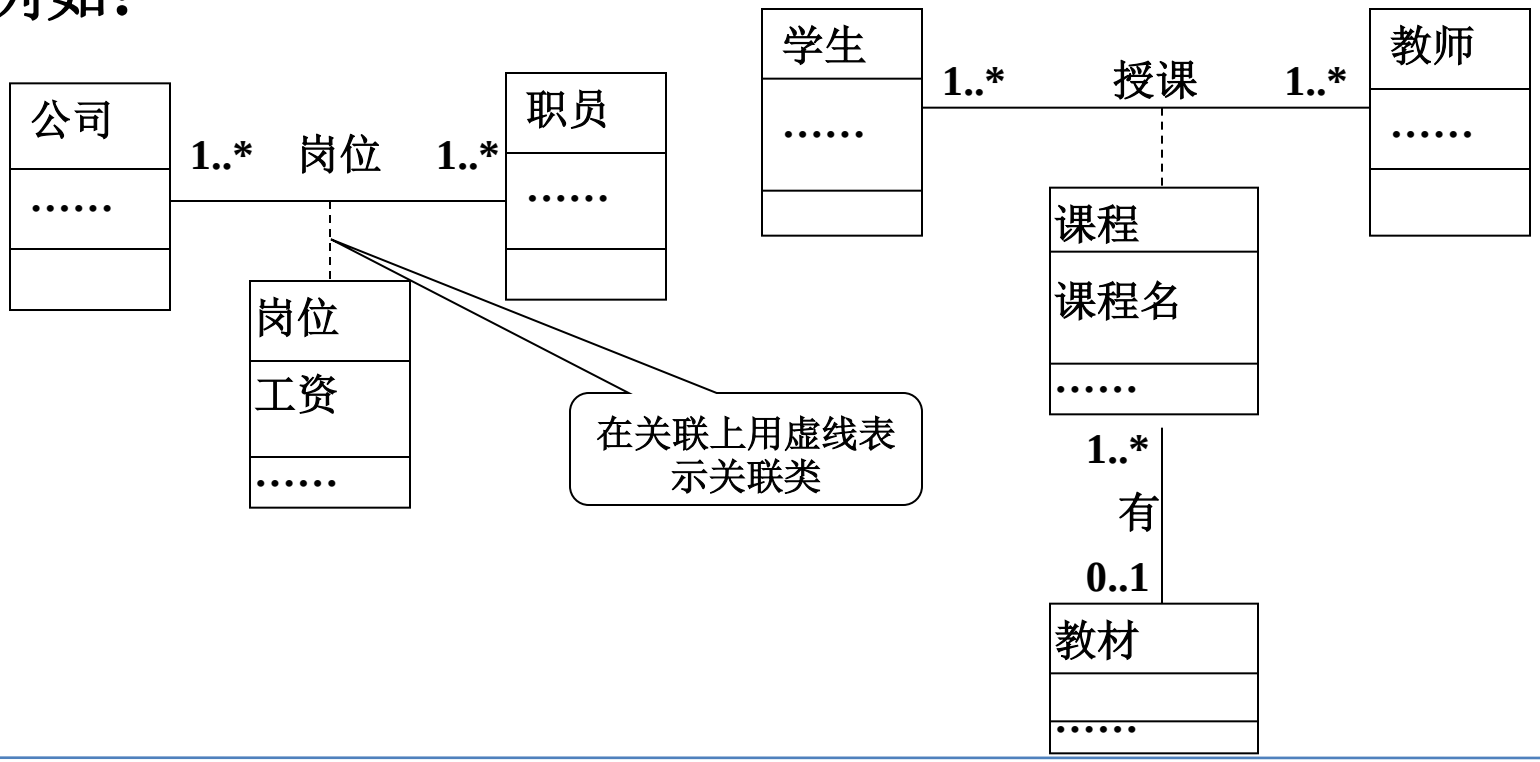
带有角色名的关联：



(5)关联类

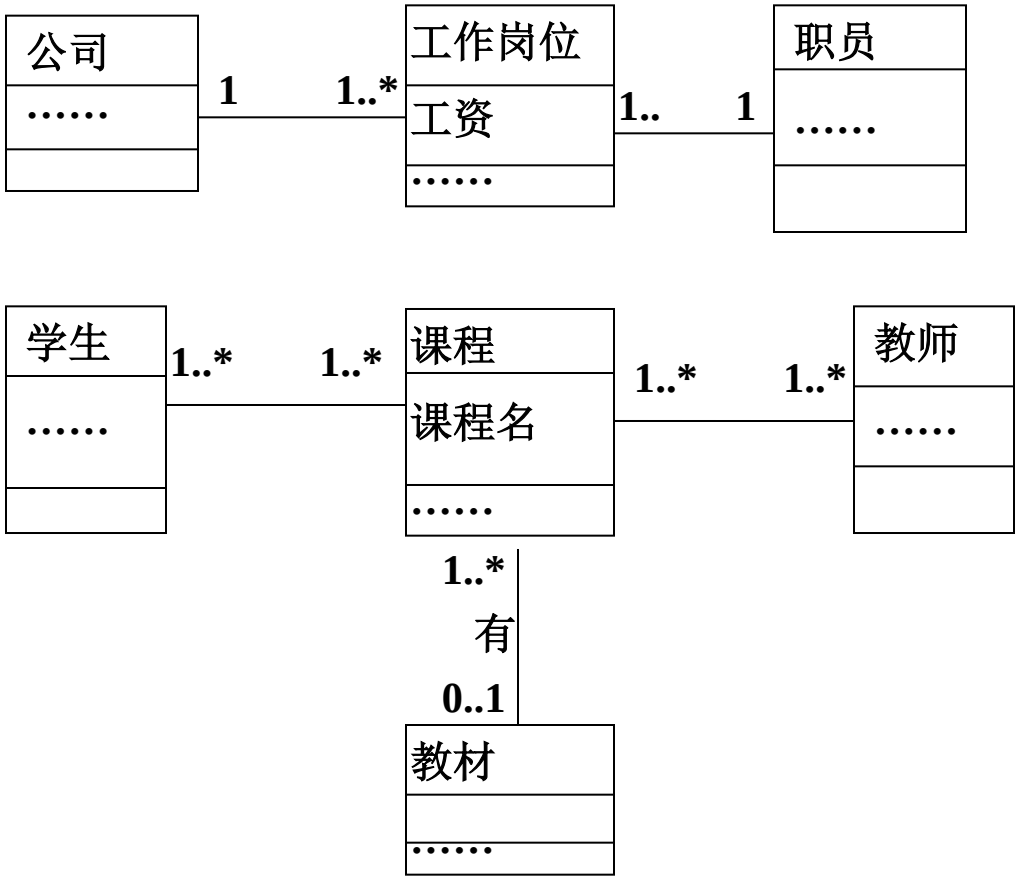
关联类是一种建模元素，表示关联本身也是一个类。关联类兼有关联和类的双重特征，可把它看作具有类性质的关联，也可看作是具有关联性质的类。

例如：



关联类的概念在建模中不是必不可少的，可以通过增设类把关联类表示为普通类。

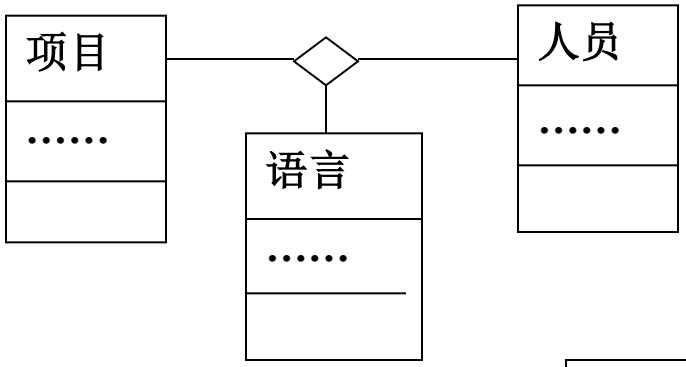
例如：



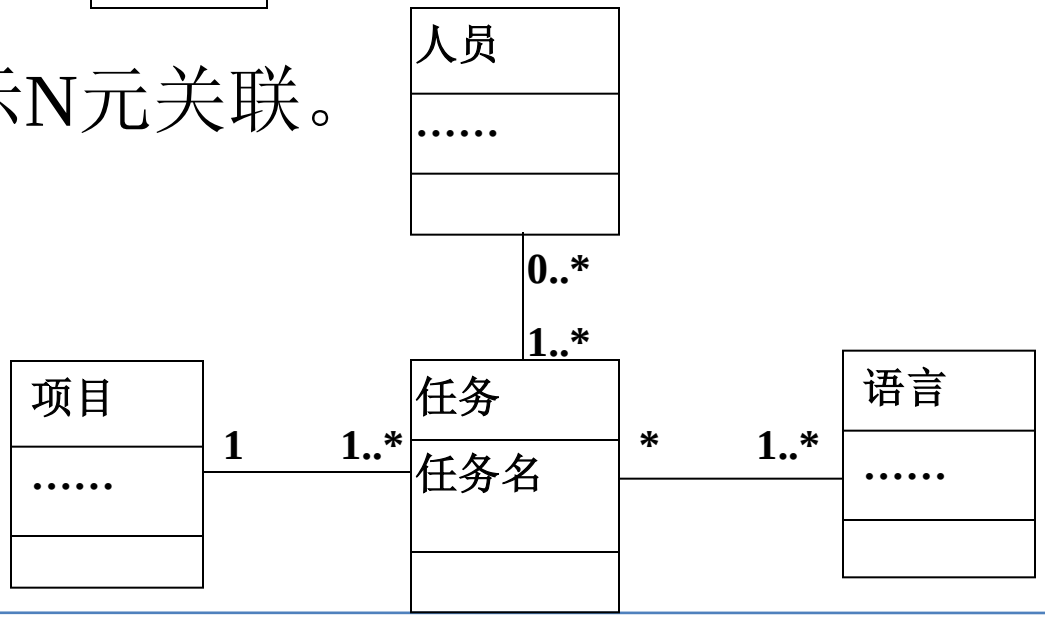
(6) N元关联

N元关联是三个或多个类之间的关联。

N元关联



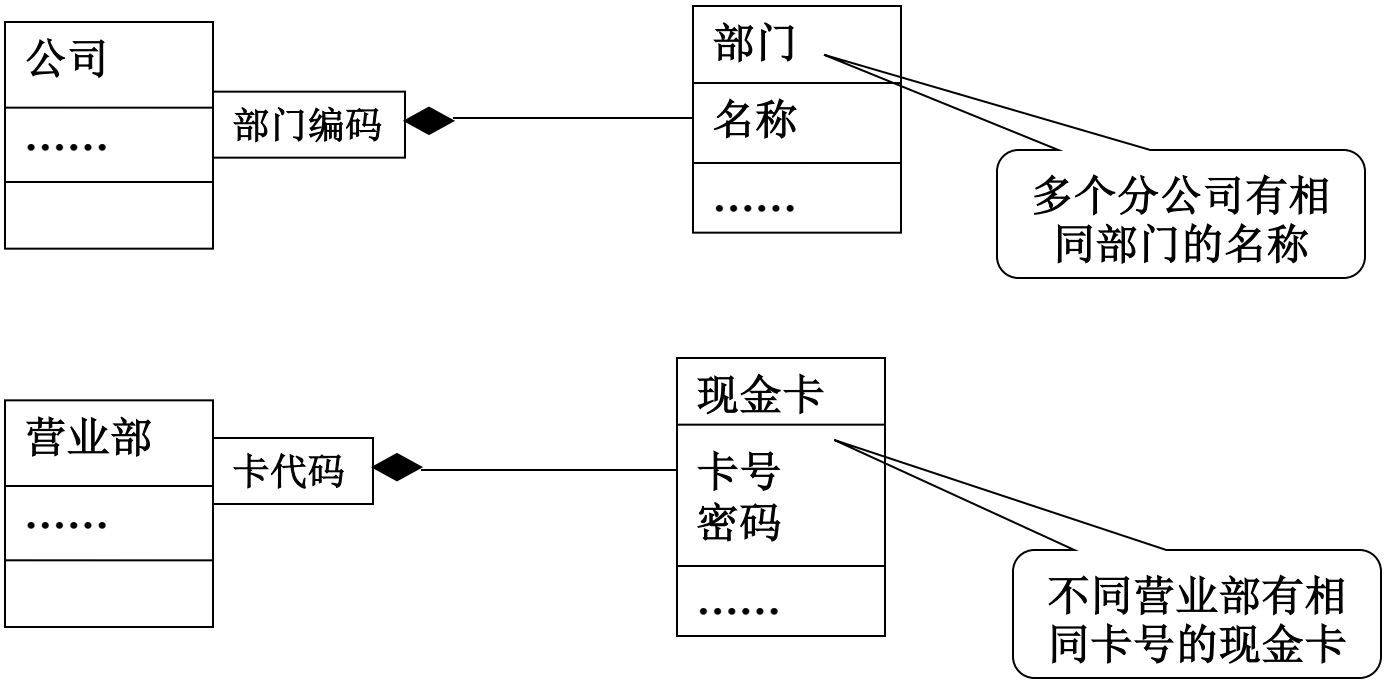
增设类可以表示N元关联。



(7) 关联限定

关联限定----表示作为查找另一端对象的特征属性，在特定的约束下提供快捷的搜索路径。

例:



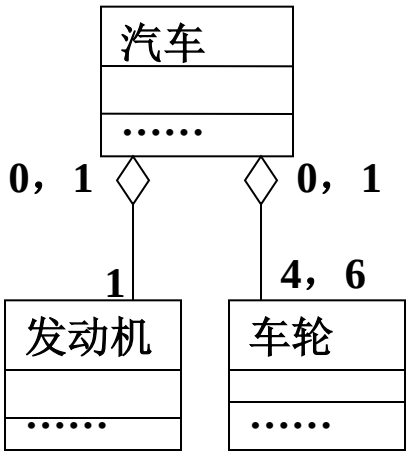
(8) 聚合关联(Aggregation)

聚合关联表示一种特殊的关联。

聚合定义 聚合表示整体类和部分类之间的“整体—部分”关系。

聚合----部分对象和整体对象可以独立生存

例:

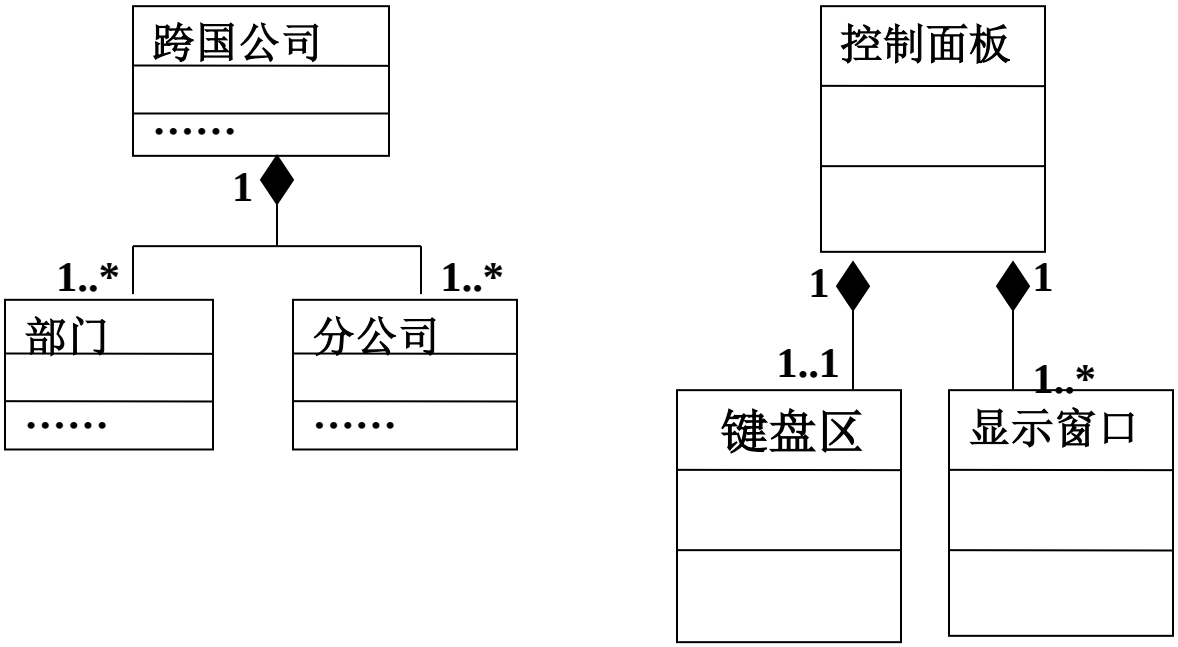


组合关联(Composition)

组合定义　组合是聚合的一种形式，其中部分和整体之间具有很强的“属于”关系，且它们的生存期是一致的。

组合----部分对象和整体对象生存期一致

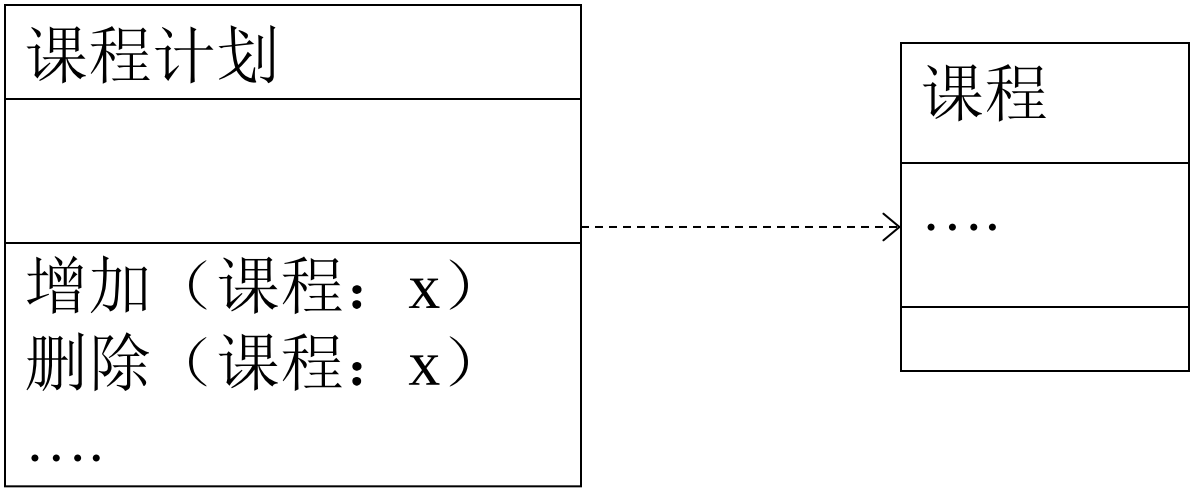
例:



(9)类的依赖关系表示

依赖关系定义　模型元素(或模型元素集合)之间的一种语义关系，目标元素的改变需要根据源元素的改变而变化。

例：课程计划依赖课程的变化而变化



面向对象分析与UML建模(3)

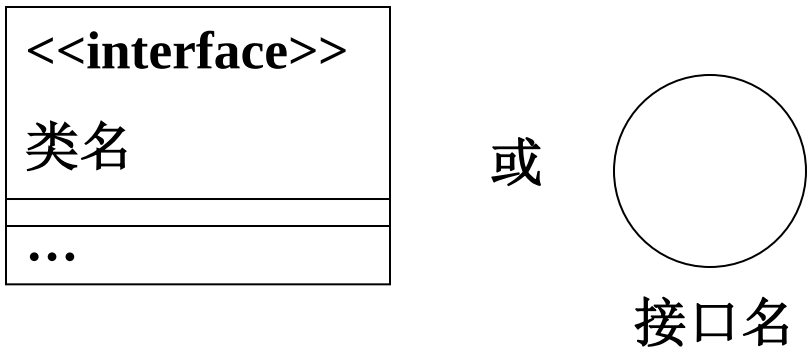
- 1. 面向对象建模
  - 1.1. 类图和对象建模
  - 1.2. 识别对象和筛选策略
  - 1.3. 从对象抽象类
  - 1.4. 主动对象和控制线程
- 2. 类的命名、属性和操作
  - 2.1. 类的命名
  - 2.2. 类属性及识别筛选策略
  - 2.3. 类操作及识别调整策略
- 3. 类的继承和关联
  - 3.1. 类的继承关系
  - 3.2. 建立类的关联
- 4. 接口类
- 5. 包图

接口类和信号

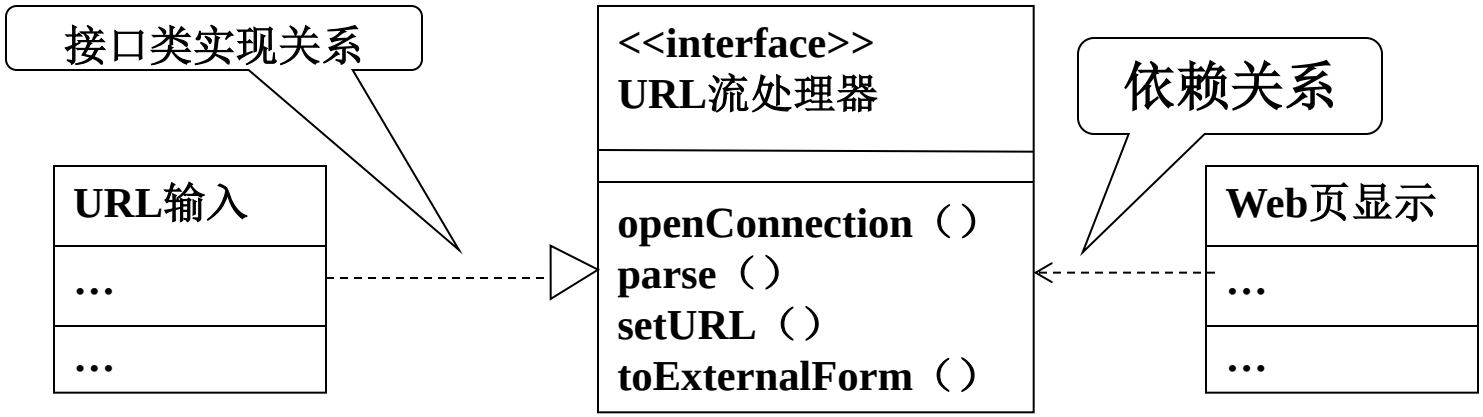
接口类----把类的公共可见性操作组织在一起，提供的服务集合。

- 接口类作为类之间交互操作的契约。
- 接口类两端的类可独立变更，但操作契约不变。
- 多个类可使用一个接口类呈现整体服务。

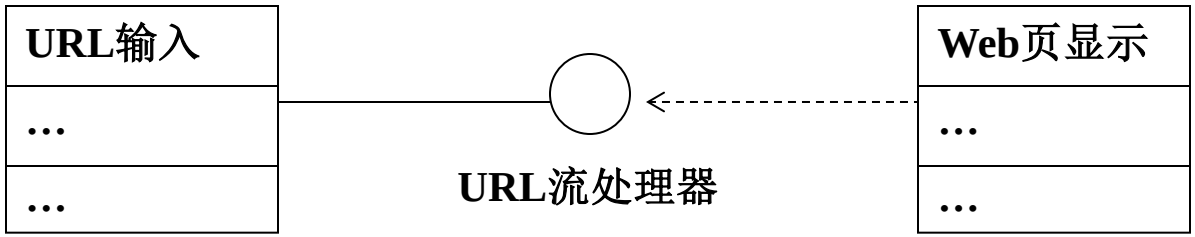
接口类的表示：



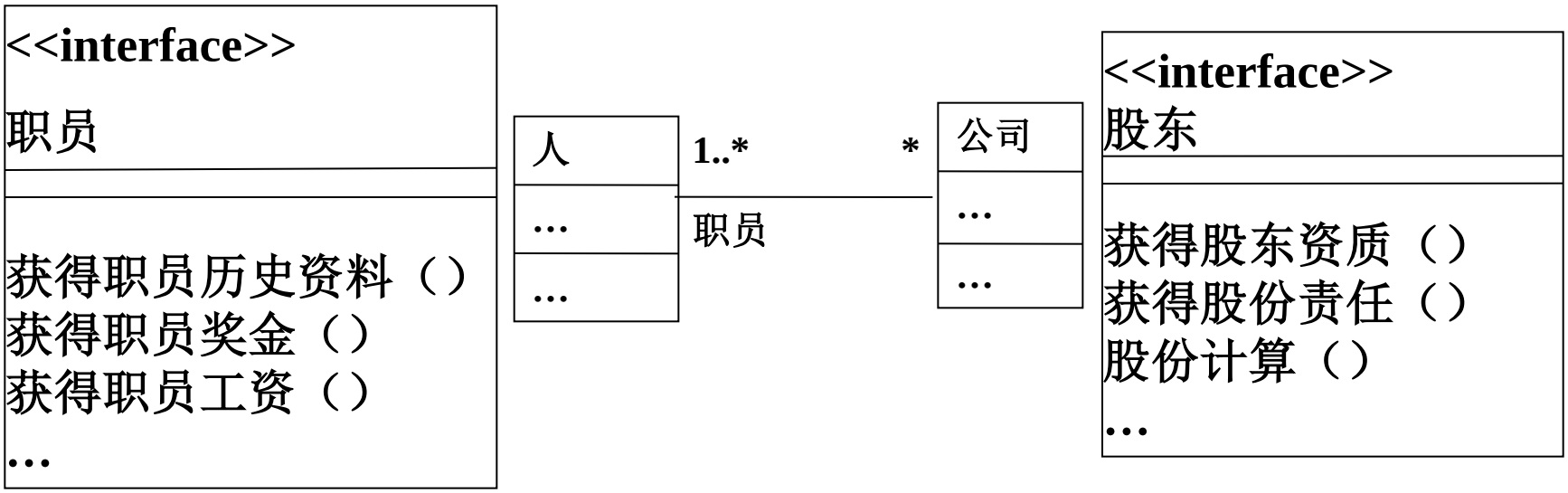
接口类示例



简化形式的类接口表示：



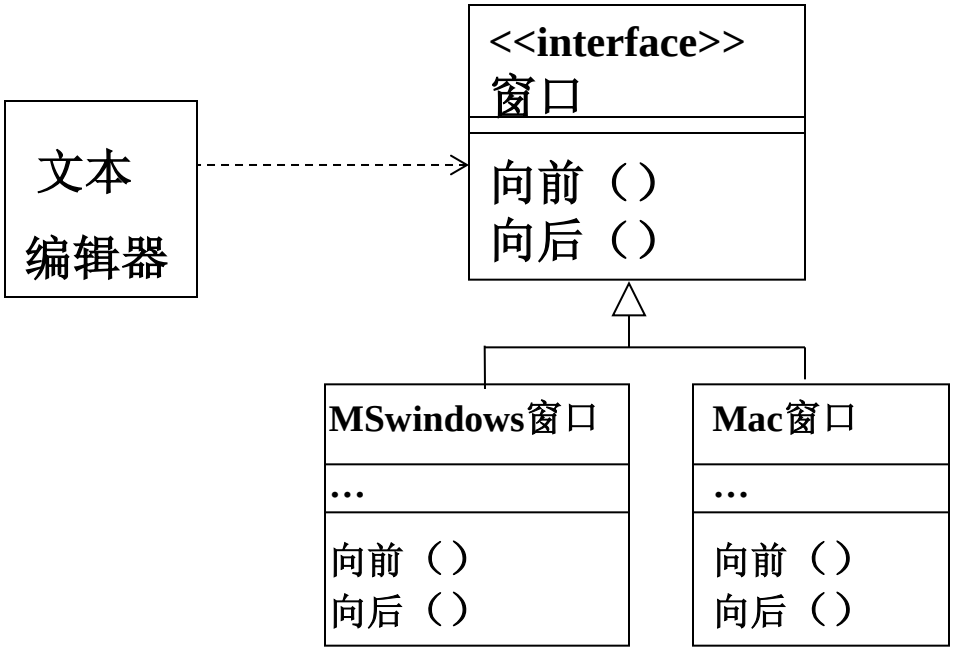
表示关联上的接口



在人和公司之间定义了职员角色的接口类，其中提供了角色的相关操作。

类似地，可以提供股东角色的接口。

公共接口的抽象类



可以把继承中的抽象类作为接口类，用特殊类对接口中的抽象操作进行具体的操作执行。

面向对象分析与UML建模(3)

- 1. 面向对象建模
  - 1.1. 类图和对象建模
  - 1.2. 识别对象和筛选策略
  - 1.3. 从对象抽象类
  - 1.4. 主动对象和控制线程
- 2. 类的命名、属性和操作
  - 2.1. 类的命名
  - 2.2. 类属性及识别筛选策略
  - 2.3. 类操作及识别调整策略
- 3. 类的继承和关联
  - 3.1. 类的继承关系
  - 3.2. 建立类的关联
- 4. 接口类
- 5. 包图

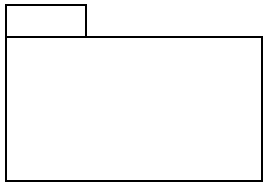
包图（Packages）

■包图概念与表示

包定义：包是对模型成分分组的机制

要点：

- 把模型成分组织成为包；
- 模型成分包括类或用况；
- 包有唯一的名称，可以被独立引用。



包的表示：

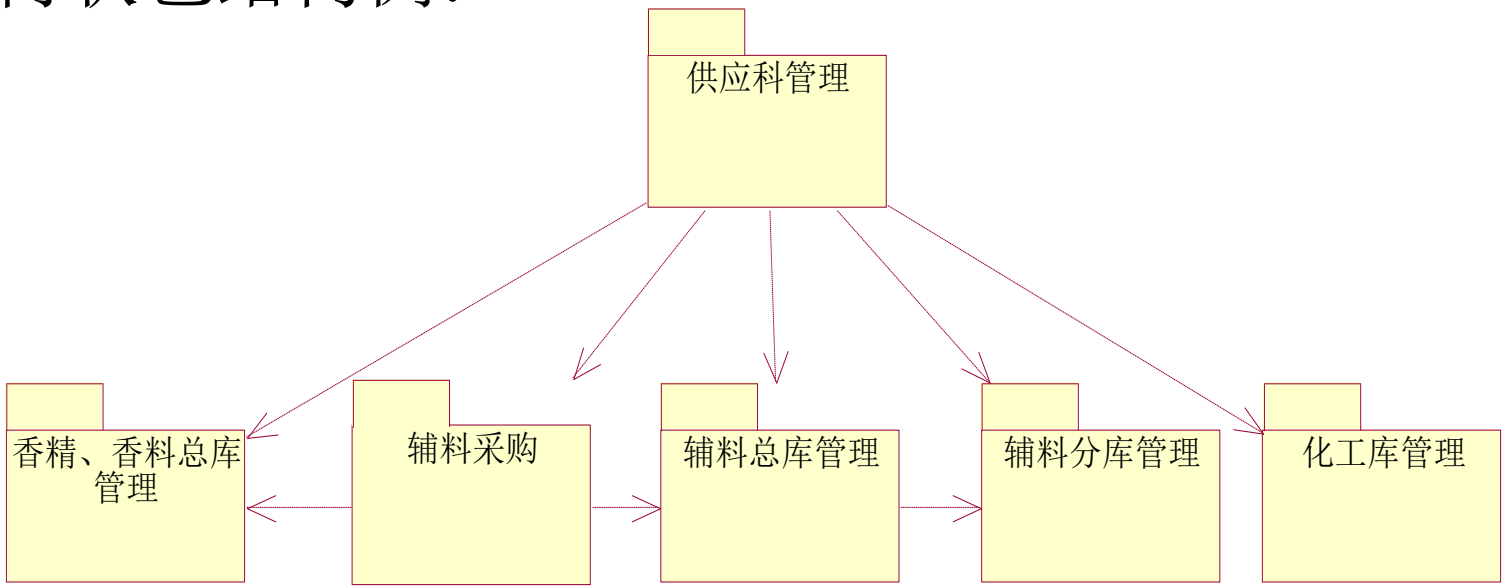
目的：

- 用包组织系统成分，使系统形成整体组织结构
- 包作为独立系统成分，可被整体利用

包的层次性

多个包可以形成严格的树形层次结构，用于描述系统的组织结构。

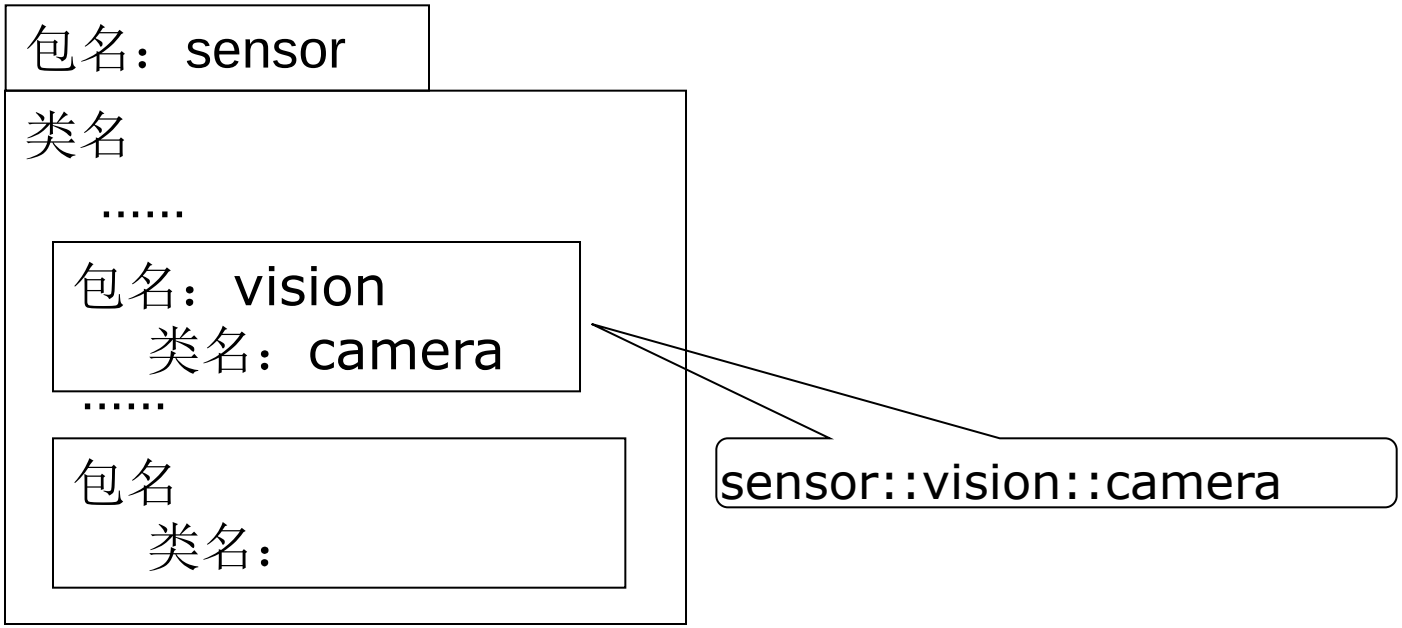
树状包结构例：



包的嵌套性

一个包可以嵌套在另一个包内，内层的包成分，同时属于内层和外层两个包。

包的嵌套结构例:



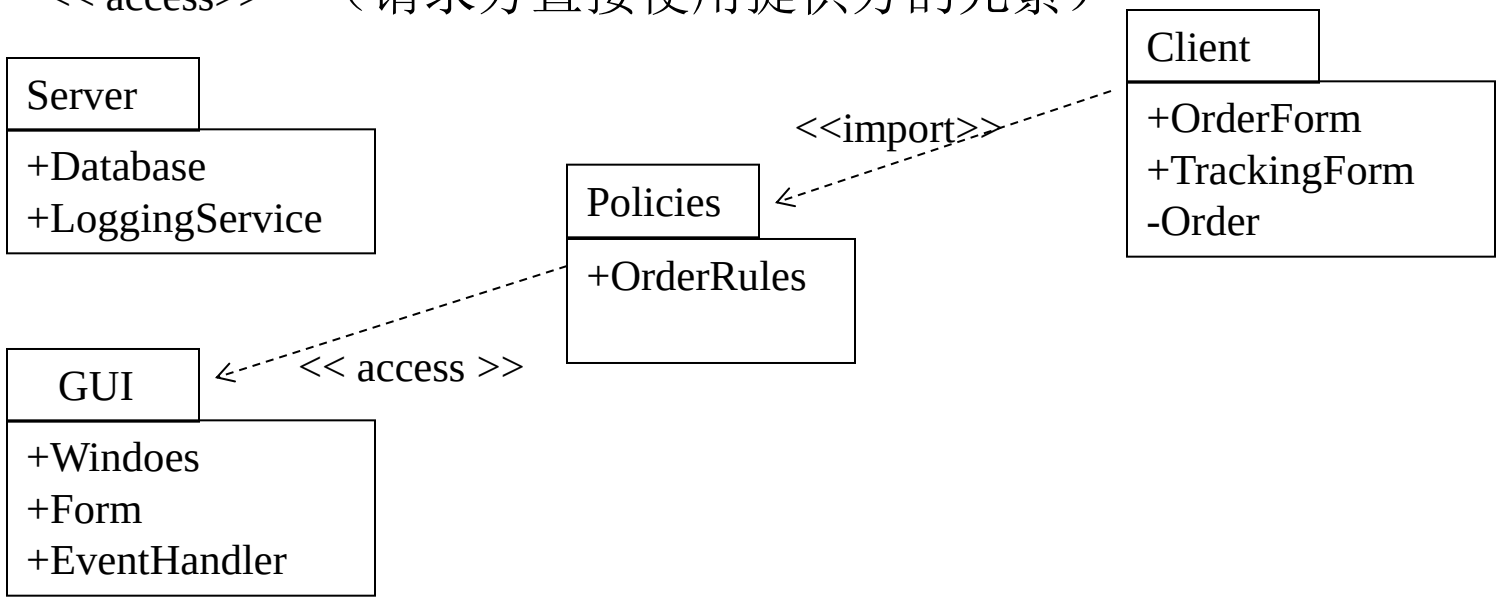
包之间依赖关系

引入依赖---包中可见的元素可以被另一个包引用

<<import>>　　（即提供方的元素直接附加到请求方）

访问依赖---包中可见的元素可以被另一个包使用

<< access>>　　（请求方直接使用提供方的元素）



包中元素的可见性分为：

公共的（+）、私有的（-）和受保护的（#）

包的组织和划分

划分包的策略：

1）把在语义上接近并需要一起变化的成分组织成包；

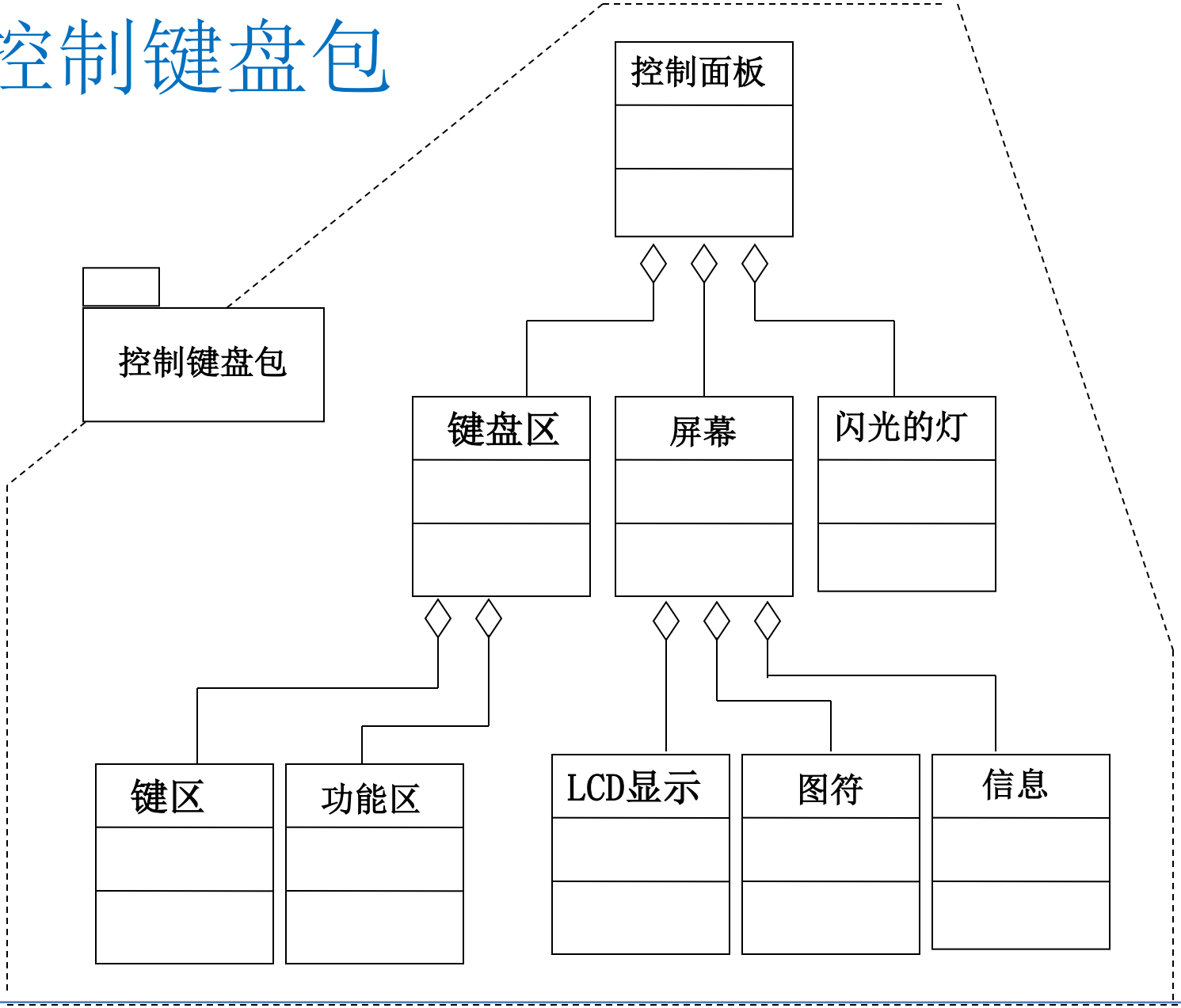
2）可以组织合并包，形成包的嵌套结构，每个包的内层成分最多5-7个；

3）组织包形成树层次结构；

4）标识包中模型成分的可见性；

5）标识包之间的依赖关系。

控制键盘包



本节小结

- 类图
  - 类图和面向对象建模
  - 类的命名、属性和操作
  - 类的继承和关联
  - 接口类
- 包图