软件工程

大连理工大学软件学院

第三篇 面向对象方法学

第9章 面向对象的概念与模型

9.1 面向对象方法学概述

- 面向对象(Object-Oriented,缩写为OO)方法学的出发点和基本原则，是尽可能模拟人类习惯的思维方式，使开发软件的方法与过程尽可能接近人类认识世界解决问题的方法与过程。
- 使描述问题的问题空间(也称为问题域)与实现解法的解空间(也称为求解域)在结构上尽可能一致。

- 客观世界中的实体既具有静态的属性又具有动态的行为。
- 传统语言提供的解空间对象实质上却仅是描述实体属性的数据，必须在程序中从外部对它施加操作，才能模拟它的行为。

- 软件系统本质上是信息处理系统。数据和处理原本是密切相关的，把数据和处理人为地分离成两个独立的部分，会增加软件开发的难度。
- 与传统方法相反，面向对象方法是一种以数据或信息为主线，把数据和处理相结合的方法。把对象作为由数据及可以施加在这些数据上的操作所构成的统一体。
- 对象与传统的数据有本质区别，它不是被动地等待外界对它施加操作，相反，它是进行处理的主体。
- 必须发消息请求对象主动地执行它的某些操作，处理它的私有数据，而不能从外界直接对其私有数据进行操作。

- 面向对象方法学所提供的“对象”概念，是让软件开发者自己定义或选取解空间对象，然后把软件系统作为一系列离散的解空间对象的集合。
- 应该使这些解空间对象与问题空间对象尽可能一致。这些解空间对象彼此间通过发送消息而相互作用，从而得出问题的解。
- 也就是说，面向对象方法是一种新的思维方法，它不是把程序看作是工作在数据上一系列过程或函数的集合，而是把程序看作是相互协作而又彼此独立的对象的集合。

- 每个对象就像一个微型程序，有自己的数据、操作、功能和目的。向着减少语义断层的方向迈了一大步，在许多系统中解空间对象都可以直接模拟问题空间的对象，解空间与问题空间的结构十分一致，因此，这样的程序易于理解和维护。

- 概括地说，面向对象方法具有下述四个要点。
  - 认为客观世界是由各种对象组成的，任何事物都是对象，复杂的对象可以由比较简单的对象以某种方式组合而成。按照这种观点，可以认为整个世界就是一个最复杂的对象。
    - 因此，面向对象的软件系统是由对象组成的，软件中的任何元素都是对象，复杂的软件对象由比较简单的对象组合而成。囍
    - 面向对象方法用对象分解取代了传统方法的功能分解。

– 把所有对象都划分成各种对象类(简称为类，Class)，每个对象类都定义了一组数据和一组方法。数据用于表示对象的静态属性，是对象的状态信息。

- 每当建立该对象类的一个新实例时，就按照类中对数据的定义为这个新对象生成一组专用的数据，以便描述该对象独特的属性值。
- 如，荧光屏上不同位置显示的半径不同的几个圆，虽然都是Circle类的对象，但是，各自都有自己专用的数据，以便记录各自的圆心位置、半径等等。囂
- 类中定义的方法是允许施加于该类对象上的操作，是该类所有对象共享的，并不需要为每个对象都复制操作的代码。

– 按照子类(或称为派生类)与父类(或称为基类)的关系，把若干个对象类组成一个层次结构的系统(也称为类等级)。

  - 在这种层次结构中，通常下层的派生类具有和上层的基类相同的特性(包括数据和方法)，这种现象称为继承(Inheritance)。
  - 但是，如果在派生类中对某些特性又做了重新描述，则在派生类中的这些特性将以新描述为准，也就是说，低层的特性将屏蔽高层的同名特性。囍

– 对象彼此之间仅能通过传递消息互相联系。

- 一切局部于该对象的私有信息，都被封装在该对象类的定义中，就好像装在一个不透明的黑盒子中一样，在外界是看不见的，更不能直接使用，这就是“封装性”。囍
- 综上所述，面向对象的方法学可以用下列方程来概括：囍

  OO=Objects+Classes+Inheritance+Communication with messages

- 面向对象既使用对象又使用类和继承等机制，而且对象之间仅能通过传递消息实现彼此通信。

9.2 面向对象方法学的主要优点

1. 与人类习惯的思维方法一致嗇

- 面向对象的软件技术以对象(Object)为核心，用这种技术开发出的软件系统由对象组成。
- 对象是对现实世界实体的正确抽象，它是由描述内部状态表示静态属性的数据，以及可以对这些数据施加的操作(表示对象的动态行为)，封装在一起所构成的统一体。
- 对象之间通过传递消息互相联系，以模拟现实世界中不同事物彼此之间的联系。嗇

- 面向对象的设计方法使用现实世界的概念抽象地思考问题从而自然地解决问题。
- 它强调模拟现实世界中的概念而不强调算法，它鼓励开发者在软件开发的绝大部分过程中都用应用领域的概念去思考。
- 在面向对象的设计方法中，计算机的观点是不重要的，现实世界的模型才是最重要的。
- 面向对象的软件开发过程从始至终都围绕着建立问题领域的对象模型来进行：对问题领域进行自然的分解，确定需要使用的对象和类，建立适当的类等级，在对象之间传递消息实现必要的联系，从而按照人们习惯的思维方式建立起问题领域的模型，模拟客观世界。囍

2. 稳定性好

- 面向对象方法基于构造问题领域的对象模型，以对象为中心构造软件系统。
- 它的基本作法是用对象模拟问题领域中的实体，以对象间的联系刻画实体间的联系。
- 面向对象的软件系统的结构是根据问题领域的模型建立起来的，而不是基于功能的分解，当对系统的功能需求变化时并不会引起软件结构的整体变化，仅需要作一些局部性的修改。如，从已有类派生出一些新的子类以实现功能扩充或修改，增加或删除某些对象等。
- 总之，由于现实世界中的实体是相对稳定的，因此，以对象为中心构造的软件系统也是比较稳定的。囍

3. 可重用性好

- 在面向对象方法所使用的对象中，数据和操作是作为平等伙伴出现的，对象具有很强的自含性。
- 对象所固有的封装性和信息隐藏机理，使得对象的内部实现与外界隔离，具有较强的独立性。
- 对象类提供了比较理想的模块化机制和比较理想的可重用的软件成分。囍

- 面向对象的软件技术在利用可重用的软件成分构造新的软件系统时，有很大的灵活性。
- 有两种方法可以重复使用一个对象类：
  - 一种方法是创建该类的实例，从而直接使用它；
  - 另一种方法是从它派生出一个满足当前需要的新类。
  - 继承性机制使得子类不仅可以重用其父类的数据结构和程序代码，而且可以在父类代码的基础上方便地修改和扩充，这种修改并不影响对原有类的使用。有人把对象类称为“软件IC”。
- 面向对象的软件技术所实现的可重用性是自然的和准确的，在软件重用技术中它是最成功的一个。

4. 较易开发大型软件产品

- 当开发大型软件产品时，组织开发人员的方法不恰当往往是出现问题的主要原因。
- 用面向对象范型开发软件时，可以把一个大型产品看作是一系列本质上相互独立的小产品来处理，这就不仅降低了开发的技术难度，而且也使得对开发工作的管理变得容易。
- 许多软件开发公司的经验都表明，当把面向对象技术用于大型软件开发时，软件成本明显地降低了，软件的整体质量也提高了。

5. 可维护性好

- 由于下述因素的存在，使得用面向对象方法所开发的软件可维护性好。囍
  - 面向对象的软件稳定性比较好。囍
  - 面向对象的软件比较容易修改。囍
  - 面向对象的软件比较容易理解。囍
  - 易于测试和调试。囍

9.3 面向对象的概念

- **9.3.1 对象囍**
- 在应用领域中有意义的、与所要解决的问题有关系的任何事物都可以作为对象(Object)，它既可以是具体的物理实体的抽象，也可以是人为的概念，或者是任何有明确边界和意义的东西。
- 例如，一名职工、一家公司、一个窗口、一座图书馆、一本图书、贷款和借款等，
- 设立某个对象就反映了软件系统保存有关它的信息并具与它进行交互的能力。

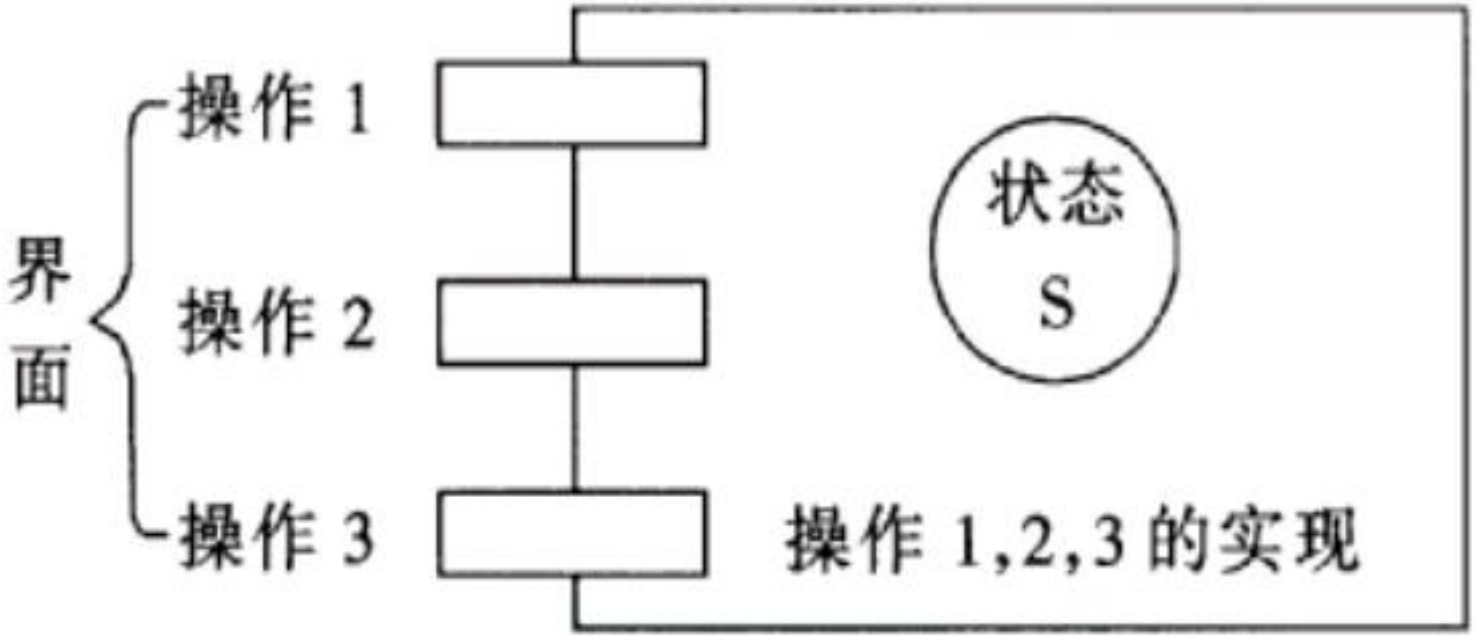

图6.1 对象的形象表示

1. 对象的形象表示

2. 对象的定义

1. 定义1

- 对象是具有相同状态的一组操作的集合。
- 主要是从面向对象程序设计角度看“对象”。

2. 定义2

- 对象是对问题域中某个东西的抽象，这种抽象反映了系统保存有关这个东西的信息或与它交互的能力。对象是对属性值和操作的封装。
- 着重从信息模拟的角度看待“对象”。

3. 定义3囍

- 对象:: =<ID，MS，DS，MI>囍

  其中，ID是对象的标识或名字囍

  MS是对象中的操作集合囍

  DS是对象的数据结构囍

  MI是对象受理的消息名集合（即对外接口）

- 一个形式化的定义。囍

- 对象是封装了数据结构及可以施加在这些数据结构上的操作的封装体，这个封装体有可以唯一地标识它的名字，而且向外界提供一组服务(即公有的操作)。
- 对象中的数据表示对象的状态，一个对象的状态只能由该对象的操作来改变。每当需要改变对象的状态时，只能由其他对象向该对象发送消息。对象响应消息时，按照消息模式找出与之匹配的方法，并执行该方法。
- 从动态角度或对象的实现机制来看，对象是一台自动机。

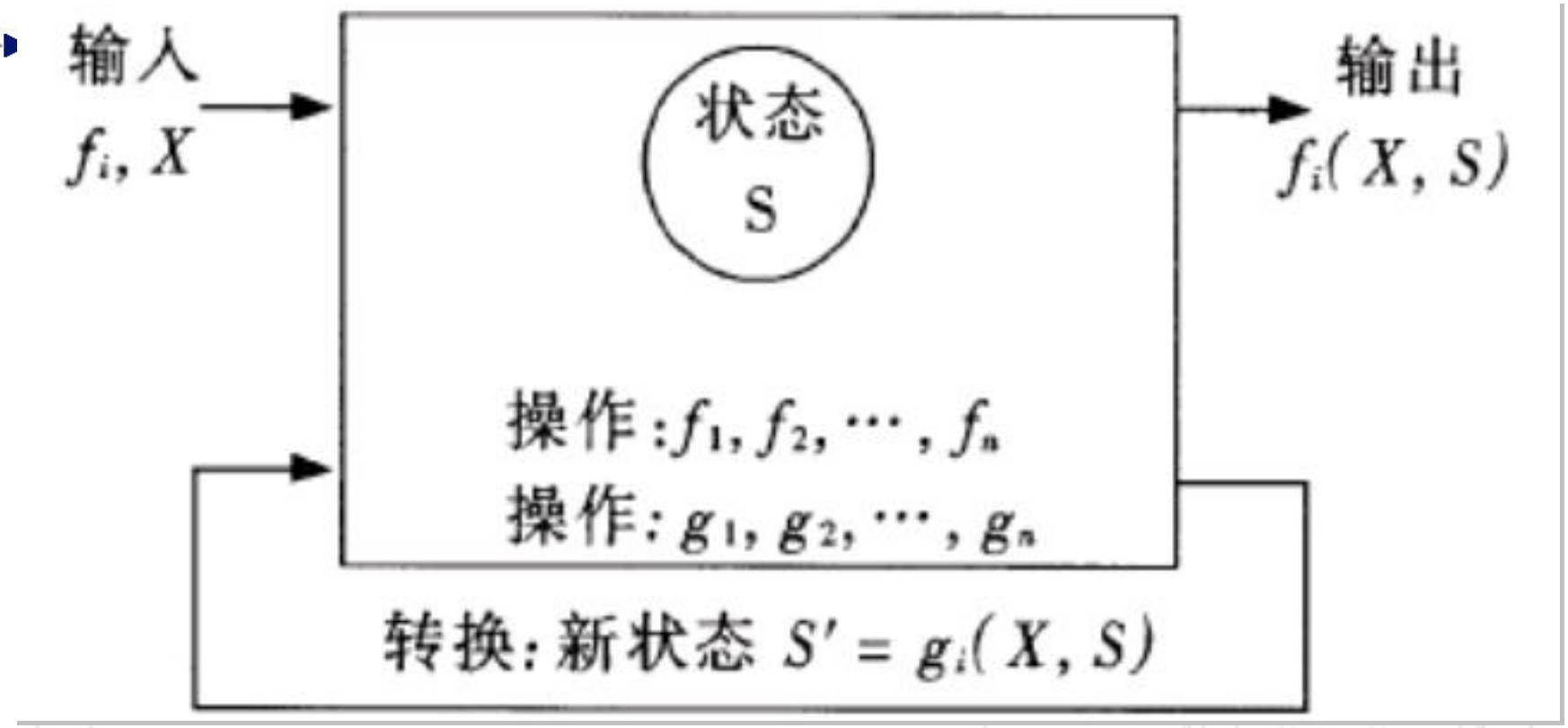

图6.2 用自动机模拟对象

3. 对象的特点

- 以数据为中心。
- 对象是主动的。
- 实现了数据封装。
- 本质上具有并行性。
- 模块独立性好。

- **9.3.2 其他概念囍**

1. 类(Class)囍

– 在面向对象的软件技术中，“类”就是对具有相同数据和相同操作的一组相似对象的定义，类是对具有相同属性和行为的一个或多个对象的描述，通常在这种描述中也包括对怎样创建该类的新对象的说明。

2. 实例(Instance)囍

– 实例就是由某个特定的类所描述的一个具体的对象。

3. 消息(Message)囍

– 消息，就是要求某个对象执行在定义它的那个类中所定义的某个操作的规格说明。通常，一个消息由下述三部分组成：

- 接收消息的对象；囍
- 消息选择符(也称为消息名)；囍
- 零个或多个变元。囍

4. 方法(Method)囍

– 方法，就是对象所能执行的操作，也就是类中所定义的服务。方法描述了对象执行操作的算法，响应消息的方法。

5. 属性(Attribute)囍

– 属性，就是类中所定义的数据，它是对客观世界实体所具有的性质的抽象。类的每个实例都有自己特有的属性值。

6. 封装(Encapsulation)囍

- 从字面上理解，所谓封装就是把某个事物包起来，使外界不知道该事物的具体内容。
- 具有封装性的条件如下：囍
  - 有一个清晰的边界。所有私有数据和实现操作的代码都被封装在这个边界内，从外面看不见更不能直接访问。
  - 有确定的接口(即协议)。这些接口就是对象可以接受的消息，只能通过向对象发送消息来使用它。
  - 受保护的内部实现。实现对象功能的细节(私有数据和代码)不能在定义该对象的类的范围外进行访问。
- 封装性也就是信息隐藏，通过封装把对象的实现细节对外界隐藏起来了。

7. 继承(Inheritance)囍

- 广义地说，继承是指能够直接获得已有的性质和特征，而不必重复定义它们。在面向对象的软件技术中，继承是子类自动地共享基类中定义的数据和方法的机制。囍
- 当一个类只允许有一个父类时，也就是说，当类等级为树形结构时，类的继承是单继承；当允许一个类有多个父类时，类的继承是多重继承。

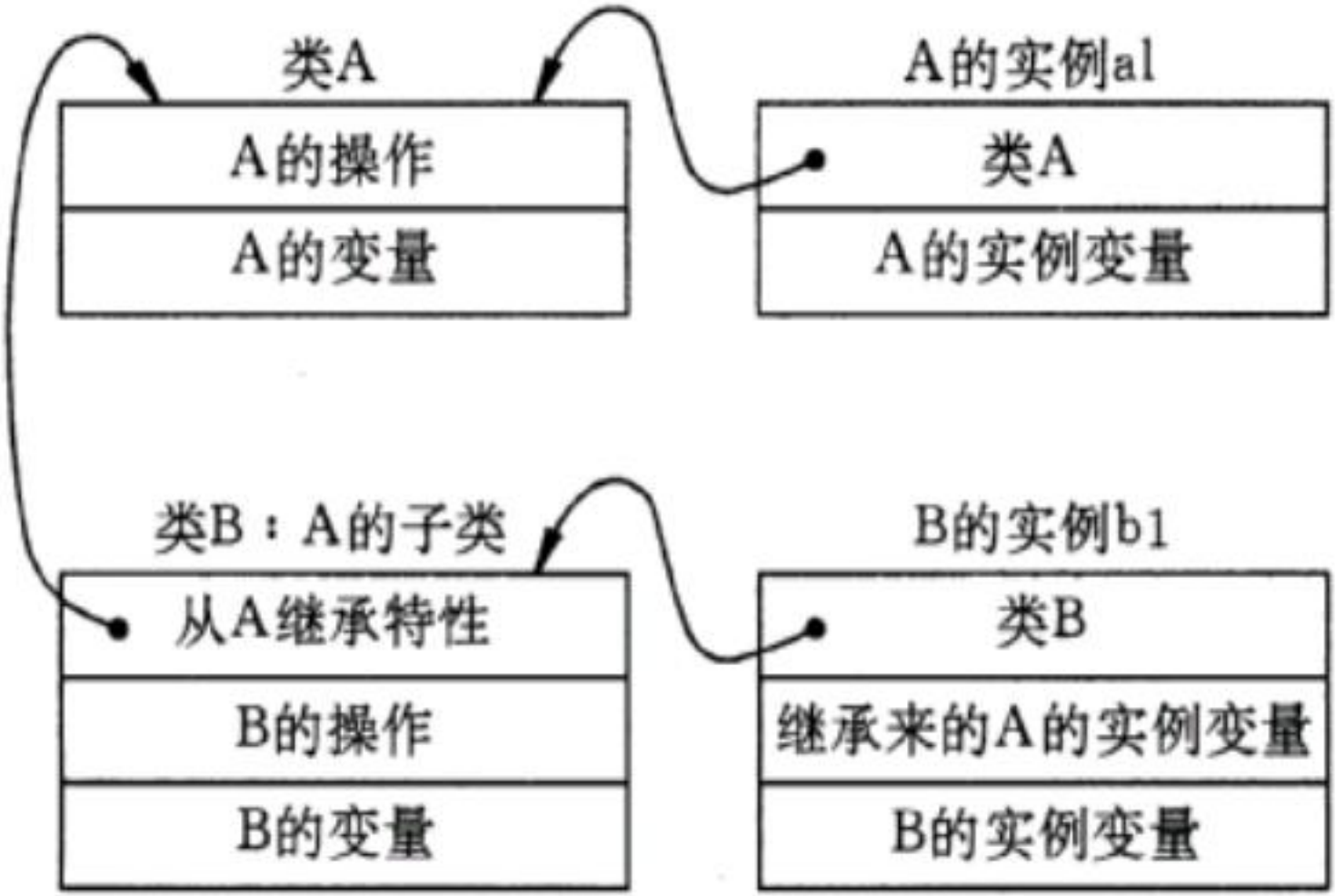

图6.3 实现继承机制的原理

8. 多态性(Polymorphism)囍

- 多态性一词来源于希腊语，意思是“有许多形态”。
- 在面向对象的软件技术中，多态性是指子类对象可以像父类对象那样使用，同样的消息既可以发送给父类对象也可以发送给子类对象。
- 在类等级的不同层次中可以共享(公用)一个行为(方法)的名字，然而不同层次中的每个类却各自按自己的需要来实现这个行为。
- 当对象接收到发送给它的消息时，根据该对象所属于的类动态选用在该类中定义的实现算法。囍

9. 重载(Overloading)

- 有两种重载：
  - 函数重载是指在同一作用域内的若干个参数特征不同的函数可以使用相同的函数名字；
  - 运算符重载是指同一个运算符可以施加于不同类型的操作数上面。
  - 当然，当参数特征不同或被操作数的类型不同时，实现函数的算法或运算符的语义是不相同的。

9.4 面向对象建模

- 建立问题模型的方法是为了更好地理解问题。
  - 所谓模型，就是为了理解事物而对事物作出的一种抽象，是对事物的一种无歧义的书面描述。
  - 通常，模型由一组图示符号和组织这些符号的规则组成，利用它们来定义和描述问题域中的术语和概念。
  - 更进一步讲，模型是一种思考工具，利用这种工具可以把知识规范地表示出来。

- 用面向对象方法开发软件，通常需要建立三种形式的模型，它们分别是描述系统数据结构的对象模型，描述系统控制结构的动态模型和描述系统功能的功能模型。
- 这种模型从三个不同但又密切相关的角度模拟目标系统，它们各自从不同侧面反映了系统的实质性内容，综合起来则全面地反映了对目标系统的需求。
- 一个典型的软件系统组合了上述三方面内容：它使用数据结构(对象模型)，执行操作(动态模型)，并且完成数据值的变化(功能模型)。囍

9.5 对象模型

- 对象模型表示静态的、结构化的系统的“数据”性质。它是对模拟客观世界实体的对象以及对象彼此间的关系的映射，描述了系统的静态结构。

- **9.5.1 表示类—&—对象的图形符号囍**

1. 类—&—对象囍

– “类—&—对象”是一个专用术语，它的含义是“一个类及属于该类的对象”。图6.4(a)是表示类—&—对象的图形符号，图6.4(b)是表示类的图形符号。囍

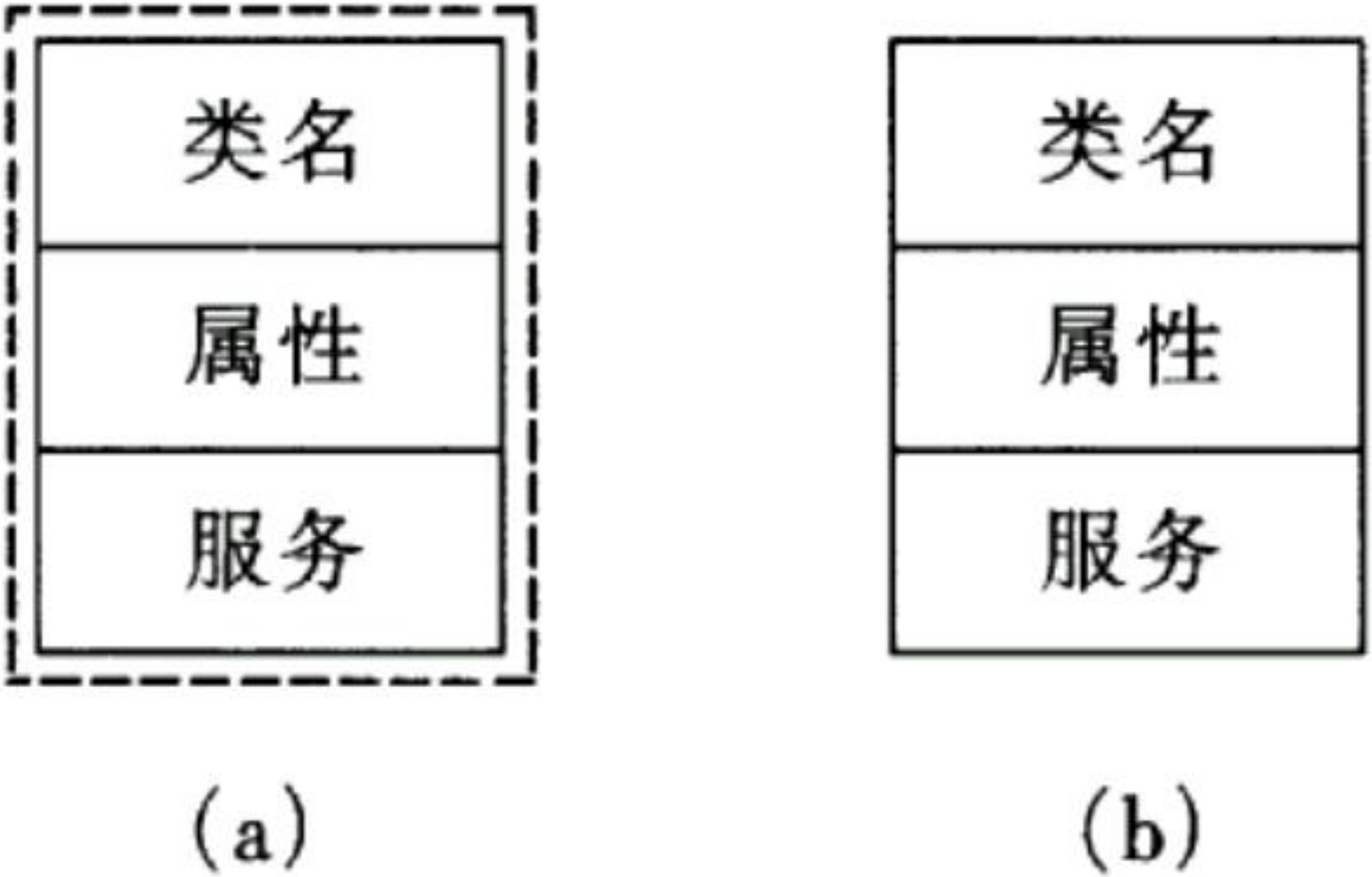

图6.4 表示符号

2. 命名

- 类名是一类对象的名字。命名是否恰当对系统的可理解性影响相当大，因此，命名时应该遵守以下几条准则。

1. 使用标准术语囍
2. 使用具有确切含义的名词囍
3. 必要时用名词短语作名字囍

- 名字应该是富于描述性的、简洁的而且无二义性的。囍

- **9.5.2 表示结构的图形符号囍**
- 在面向对象分析和面向对象设计中，结构表示了问题域中的复杂关系，是对客观世界实体相互间关系的抽象。
- 结构与目标系统的任务直接相关，即目标系统的任务决定了系统的结构。
- 类—&—对象间的关系可以概括为归纳关系、组合关系及关联关系。囍

1. 归纳关系囍

– 归纳关系就是“一般—特珠”关系，它反映了一个类与若干个互不相容子类之间的分类关系。

  - 高层类(即基类)说明一般性的属性，低层类(即派生类)说明特殊属性。
  - 低层类对象“即是(ISA)”某种特殊的高层类对象，它继承了在高层类中定义的属性和服务。囍

– 图6.5是表示归纳关系的图形符号。

2. 组合关系菖

- 组合关系就是“整体—部分”关系，它反映了对象之间的构成关系。组合关系也称为聚集关系。菖
- 图6.6是表示组合关系的图形符号。
- 组合关系具有的最重要的性质是传递性。也就是说，如果A是B的一部分，B是C的一部分，则A也是C的一部分。菖
- 当组合关系有多个层次时，可以用一棵简单的聚集树来表示它。聚集树是多级组合关系的一种简化表示形式。

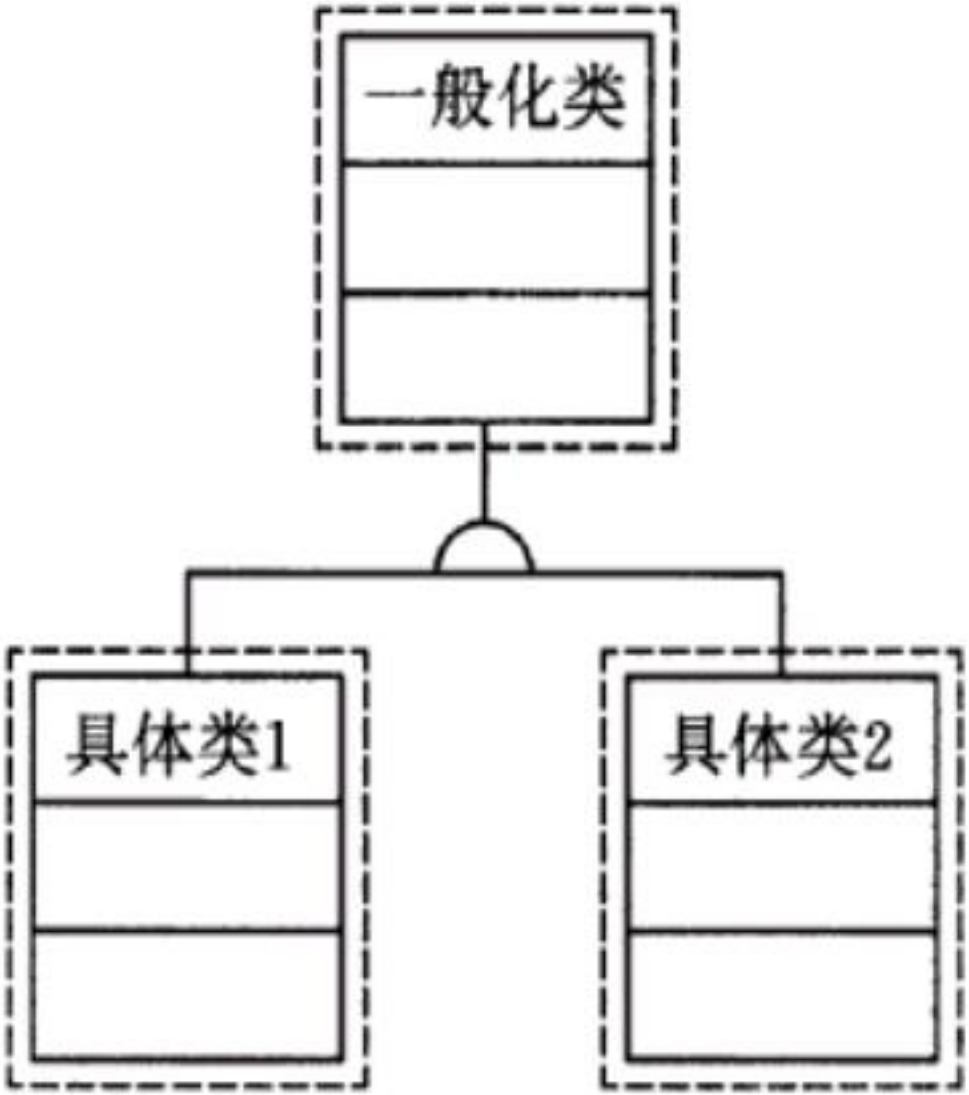

图6.5 归纳关系

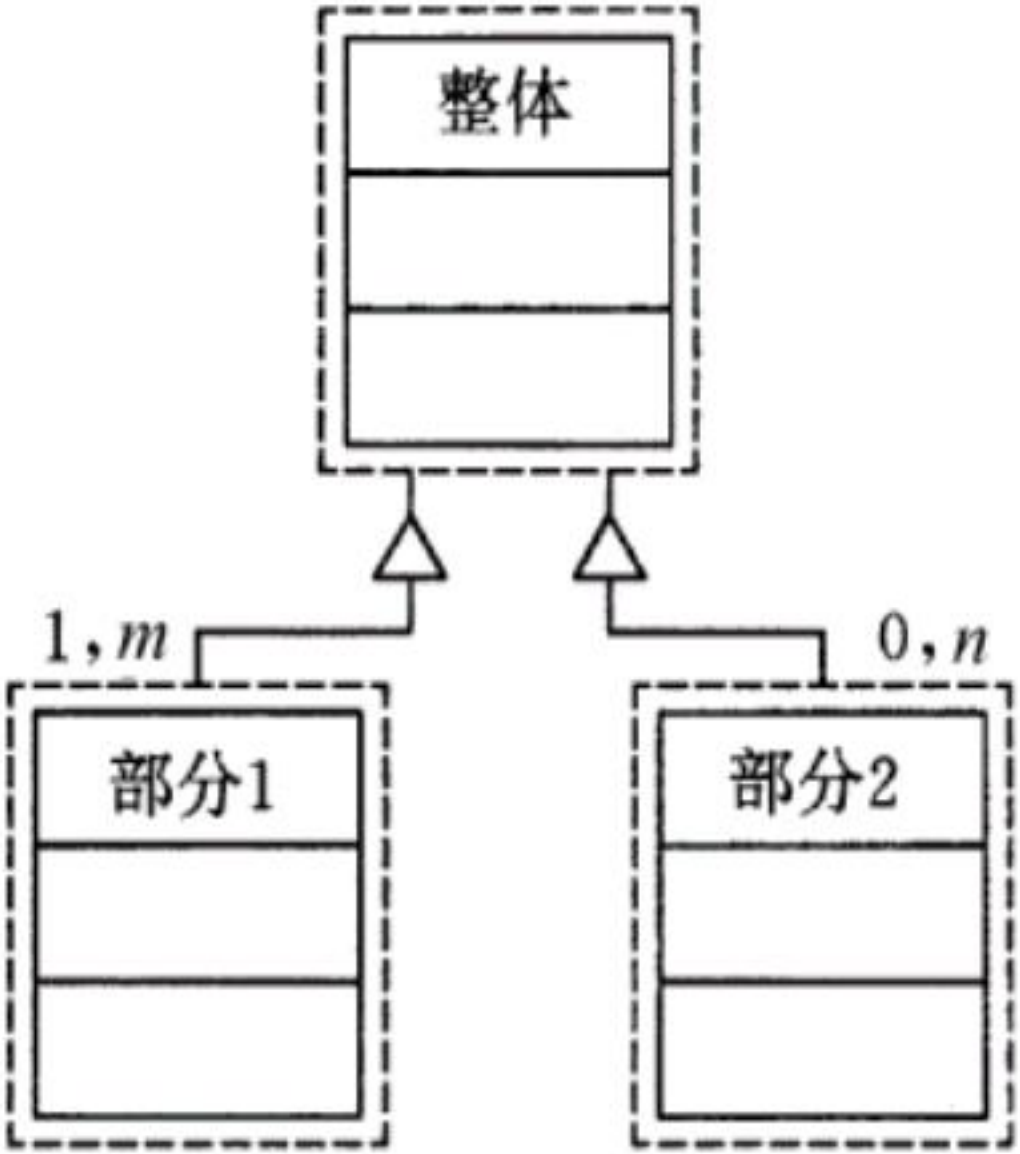

图6.6 组合关系

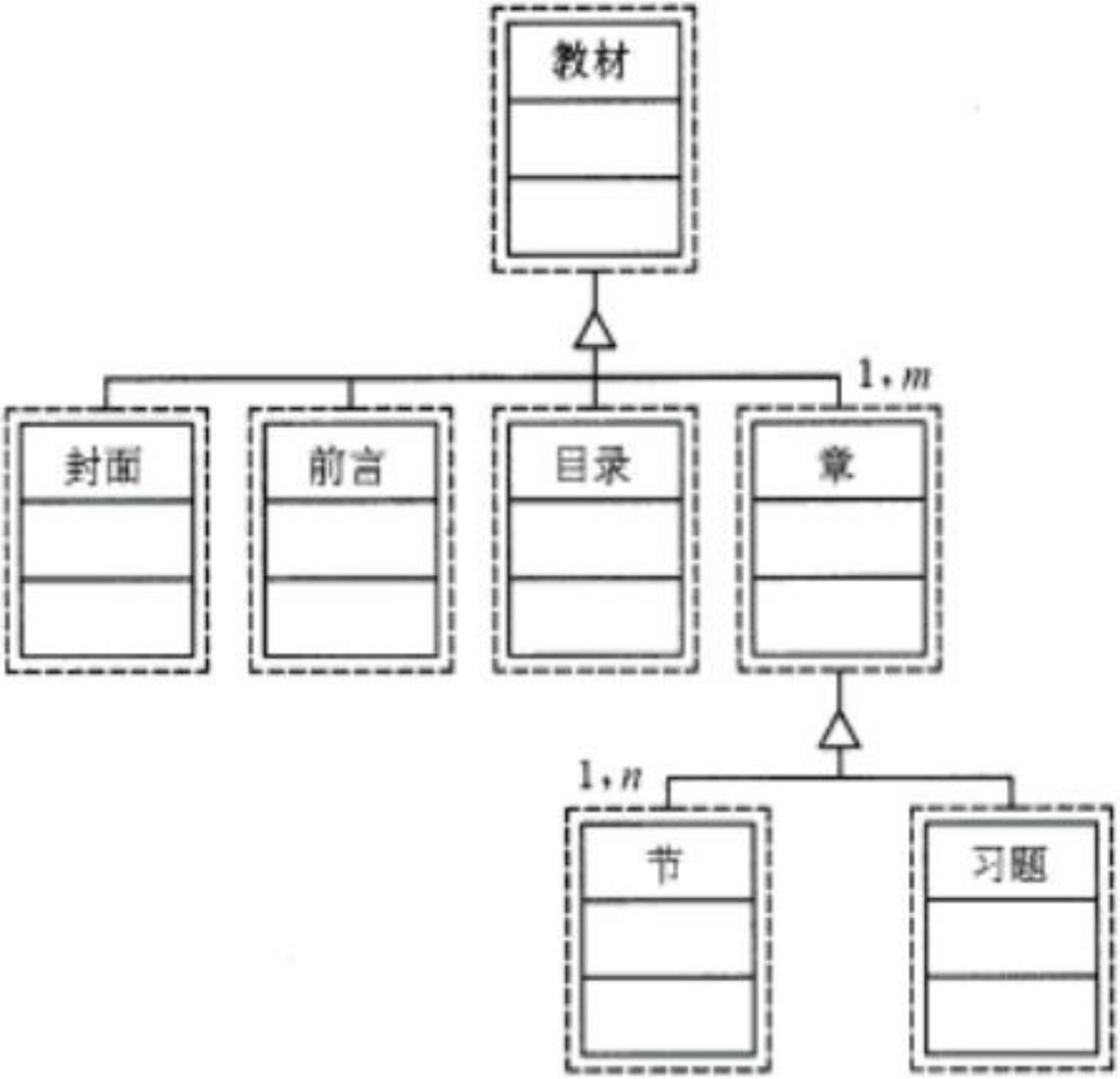

图6.7 描绘教材结构的聚集树

3. 关联关系菖

- 关联关系反映对象之间相互依赖、相互作用的关系。通常把两类对象之间的二元关系再细分为一对一(1：1)、一对多(1：M)和多对多(M：N)等三种基本类型，类型的划分依据参与关联的对象的数目。

1. 表示符号
2. 阶，所谓阶就是参与关联的对象的个数。

3. 链属性，链属性就是关联链的性质。

4. 限定囍

- 一个受限的关联由两个对象及一个限定词组成。可以把限定词看作是一种特殊的链属性。利用限定词通常能有效地减少关联的阶数。

5. 消息连接囍

- 消息连接的表示符号，是从消息发送者指向消息接收者的箭头线，如图6.11所示。囍

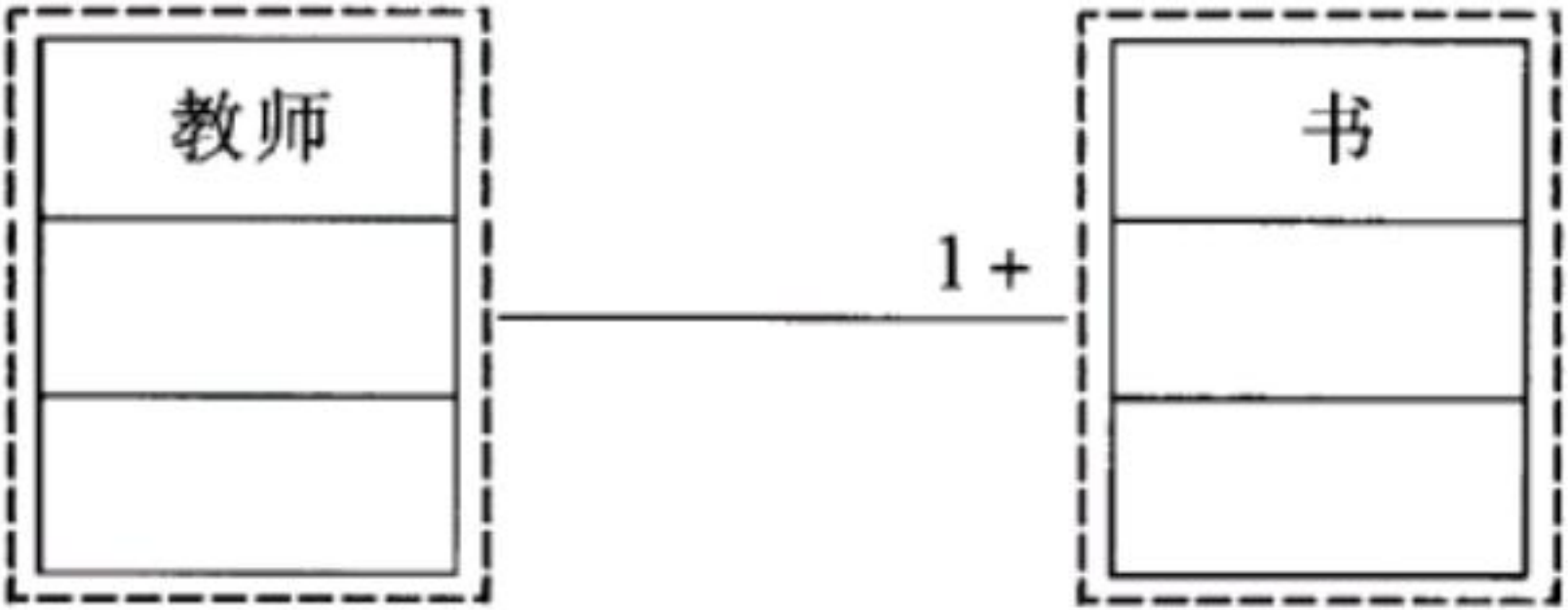

图6.8 教师与属于他的书之间的关联关系

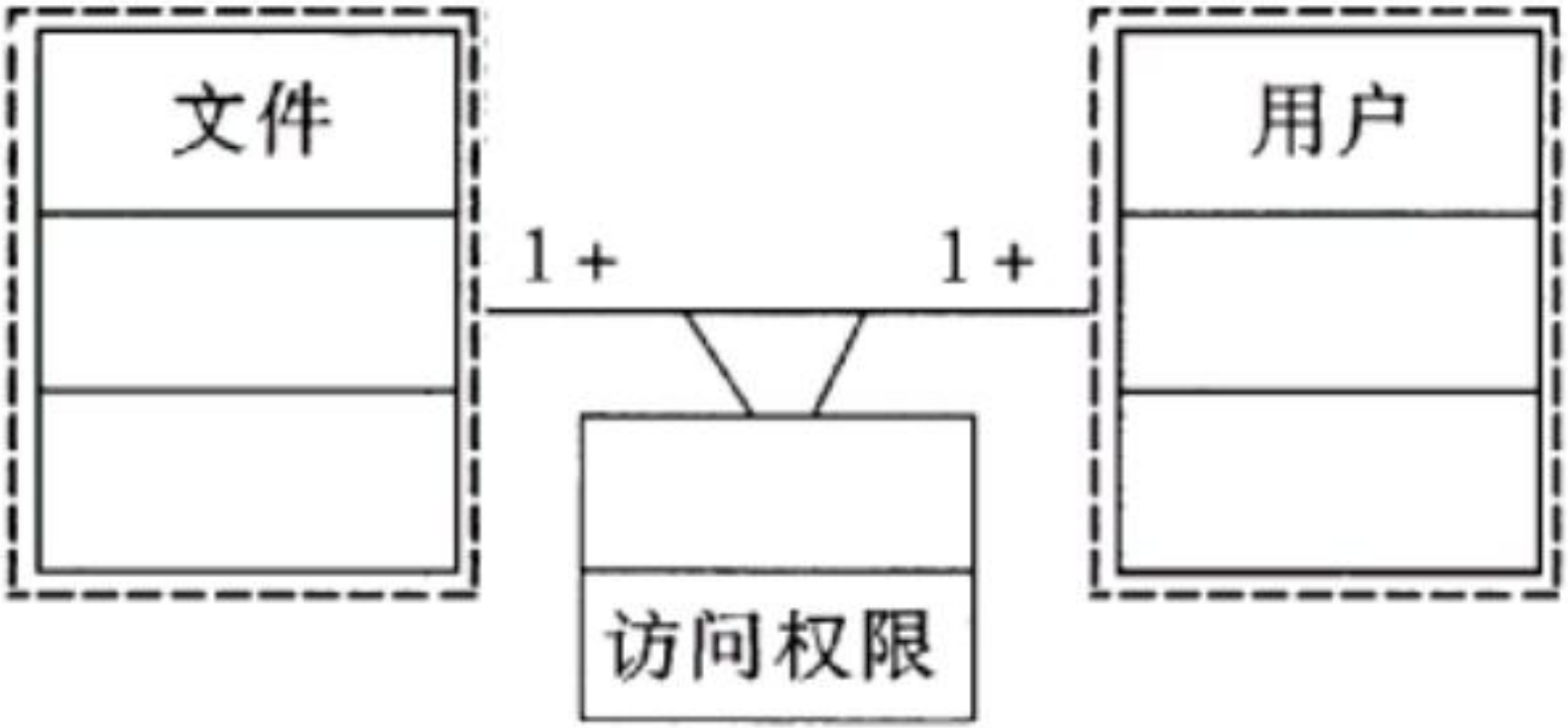

图6.9 链属性的表示方法

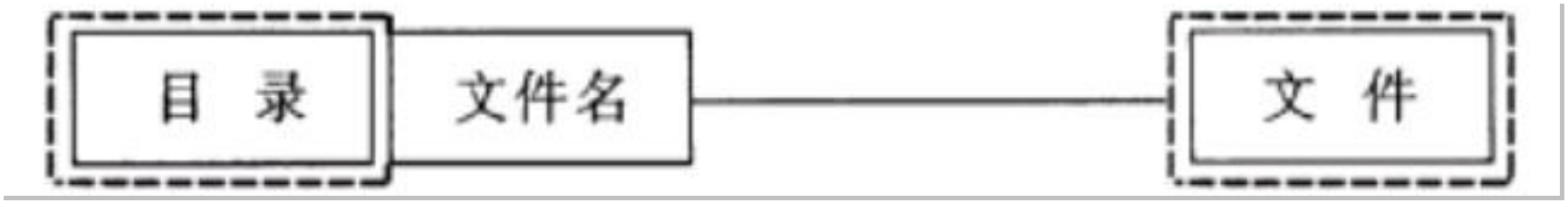

图6.10 一个受限的关联

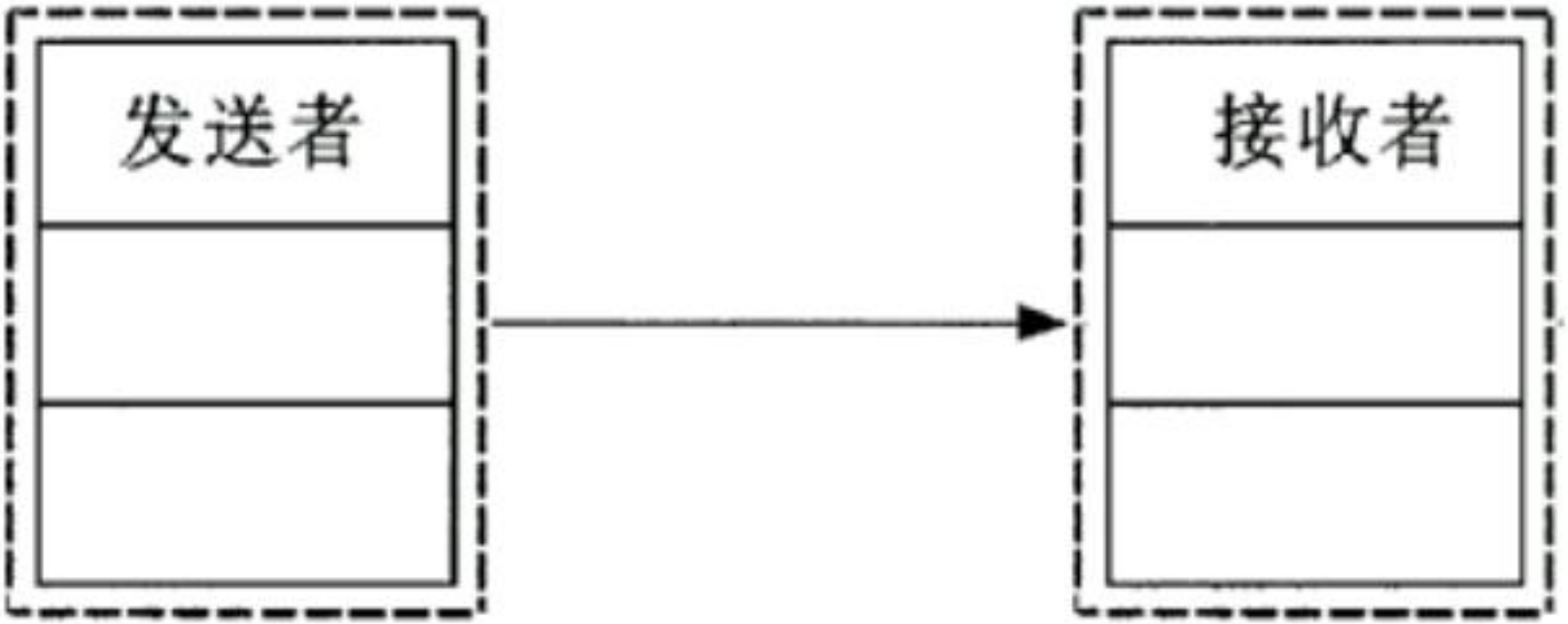

图6.11 消息连接的表示符

- **9.5.3 对象模型举例**

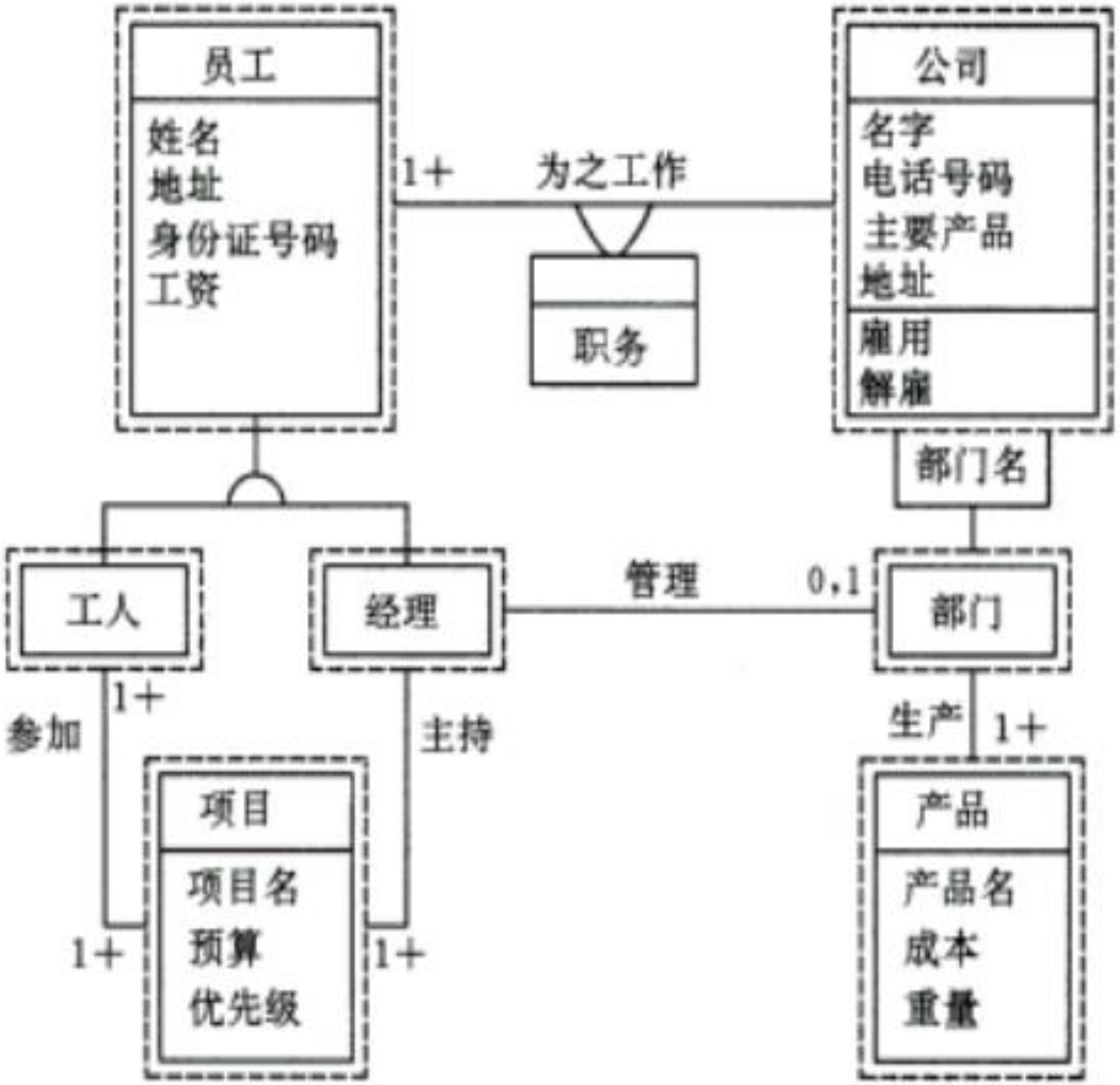

图6.12 一家公司的对象模型

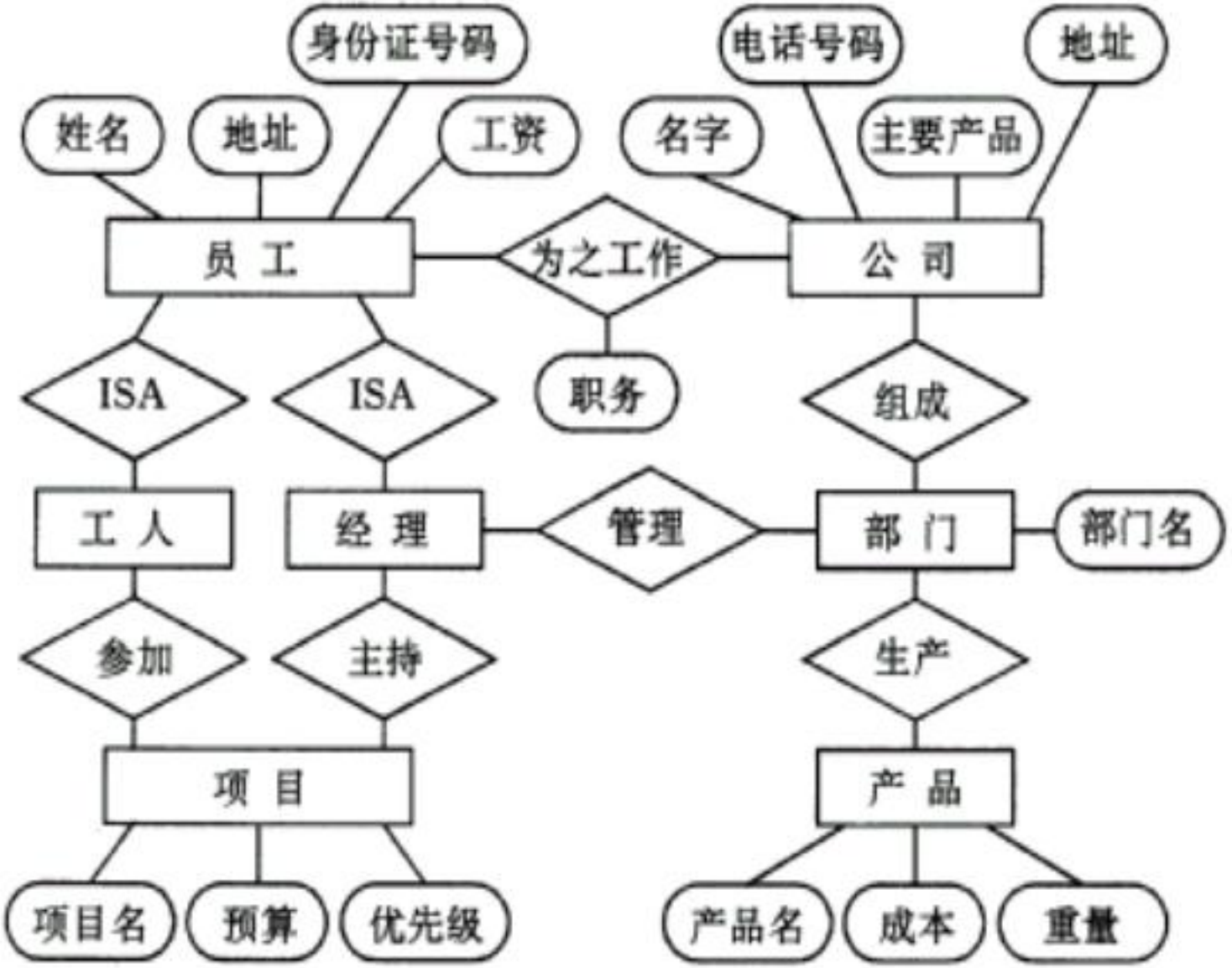

图6.13 与图6.12对应的ER图

9.6 动态模型

- 动态模型表示瞬时的、行为化的系统的“控制”性质，它规定了对象模型中的对象的合法变化序列。
- 通常，用状态图来描绘对象的状态、触发状态转换的事件、以及对象的行为(对事件的响应)。囍
- 每个类的动态行为用一张状态图来描绘，各个类的状态图通过共享事件合并起来，从而构成系统的动态模型。也就是说，动态模型是基于事件共享而互相关联的一组状态图的集合。

- **9.6.1 概念菑**

1. 事件菑

- 事件是某个特定时刻发生的事情，它是对引起对象从一种状态转换到另一种状态的现实世界中的事件的抽象。事件没有持续时间，是瞬间完成的。

2. 状态菑

- 状态就是对象在其生命周期中的某个特定阶段所具有的行为模式，它是对影响对象行为的属性值的一种抽象。菑
- 状态规定了对象对输入事件的响应方式。

3. 行为菑

- 所谓行为，是指对象达到某种状态时所做的一系列处理操作。这些操作是需要耗费时间的。

- **9.6.2 符号菖**
- 状态图中使用的主要表示符号如下所述：
- 状态用圆形框或椭圆框表示，框内可标上状态名也可以不给状态起名字，行为在框内用关键字do(后接冒号)标明。
- 从一个状态到另一个状态的转换用箭头线表示，线上标以事件名。事件名的方括号内可标注状态转换的条件。即仅当方括号内所列出的条件为真时，该事件的发生才引起箭头所示的状态转换。

9.7 功能模型

- 功能模型表示变化的系统的“功能”性质，它指明了系统应该“做什么”，因此更直接地反映了用户对目标系统的需求。囍
- **9.7.1 表示方法**囍
  - 通常，功能模型由一组数据流图组成。

- **9.7.2 三种模型之间的关系菖**
- 在面向对象方法学中，对象模型是最基本最重要的，为其他两种模型奠定基础，依靠对象模型完成三种模型的集成。
- 三种模型之间的关系：
  - 针对每个类建立的动态模型，描述了类实例的生命周期或运行周期。
  - 状态转换驱使行为发生，这些行为在数据流图中被映射成处理，它们同时与对象模型中的服务相对应。菖

- 功能模型中的处理，对应于对象模型中类—&—对象所提供的服务。
- 通常，在顶层数据流图中的处理，对应于复杂对象提供的服务；
- 在低层数据流图中的处理，对应于更基本的对象(基本对象是复杂对象的组成部分)的服务。
- 有时一个处理对应多个服务，也有一个服务对应多个处理的时候。囍

- 功能模型中的数据存储，以及数据的源点/终点(在功能模型中称为动作对象)，通常是对象模型中的对象。囍
- 功能模型中的数据流，往往是对象模型中的属性值，也可能是整个对象。囍
- 功能模型中的处理可能产生动态模型中的事件。
- 对象模型描述了功能模型中的动作对象、数据存储以及数据流的结构。囍

9.8 小结

- 面向对象方法学日益受到人们的重视，特别是在用这种方法开发大型软件产品时，可以把该产品看作是一系列本质上相互独立的小产品，这就不仅降低了开发工作的技术难度，而且也使得对开发工作的管理变得容易了。
- 对于大型软件产品来说，面向对象范型明显优于结构化范型。
- 使用面向对象范型能够开发出稳定性好、可重用性好和可维护性好的软件，这些都是面向对象方法学突出优点。

- 面向对象方法学比较自然地模拟了人类认识客观世界的思维方式，它所追求的目标和遵循的基本原则，就是使描述问题的问题空间和在计算机中解决问题的解空间，在结构上尽可能一致。
- 面向对象方法学认为，客观世界由对象组成。任何事物都是对象，每个对象都有自己的内部状态和运动规律，不同对象彼此间通过消息相互作用、相互联系，从而构成了所要分析和构造的系统。

- 系统中每个对象都属于一个特定的对象类。类是对具有相同属性和行为的一组相似对象的定义。
- 应该按照子类、父类的关系，把众多的类进一步组织成一个层次系统，这样做了之后，如果不加特殊描述，则处于下一层次上的对象可以自动继承位于上一层次的对象的属性和行为。
- 用面向对象观点建立系统的模型，能够促进加深对系统的理解，有助于开发出更容易理解、更容易维护的软件。

- 通常，从三个互不相同然而又密切相关的角度建立起三种不同的模型。它们分别是描述系统静态结构的对象模型、描述系统控制结构的动态模型、以及描述系统计算结构的功能模型。其中，对象模型是最基本、最核心、最重要的。
- 本章所讲述的面向对象方法及定义的概念和表示符号，可以适用于整个软件开发过程。软件开发人员无须像用结构分析、设计技术那样，在开发过程的不同阶段转换概念和表示符号。
- 实际上，用面向对象方法开发软件时，阶段的划分是十分模糊的，通常在分析、设计和实现等阶段间多次迭代。

作业

- **第9题**