

---

# UML 中表示类与类的关系

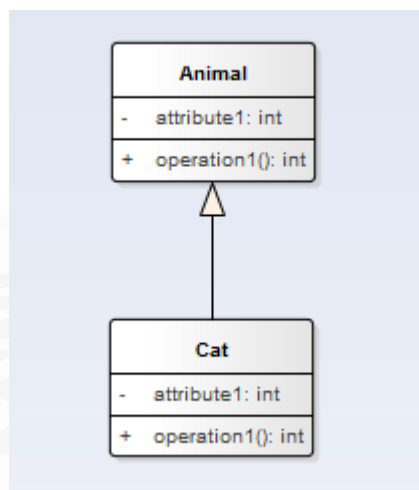
UML(Unified Modeling Language), 又称统一建模语言或标准建模语言,它是一个支持模型化和软件系统开发的图形化语言,为软件开发的所有阶段提供模型化和可视化支持,可用于多种类型软件系统开发建模的各个阶段。我们可以用 UML 图例来表示类与类之间的各种关系。

类的层次结构代表类与类之间的关系,类的关系有很多种,包括泛化、依赖、关联、聚合和组合。这些关系在大的类别上还可分为两种:横向关系和纵向关系,纵向关系就是泛化关系,而横向关系就是剩下的 4 种,它们的强弱关系:依赖 < 关联 < 聚合 < 组合。理解这些类的关系,并应用于类的层次结构设计中,有助于优化代码组织结构。

接下来我们用 Enterprise Architect 12.0.1210 Corporate Edition 建模软件为大家演示这几种类与类关系的表示。

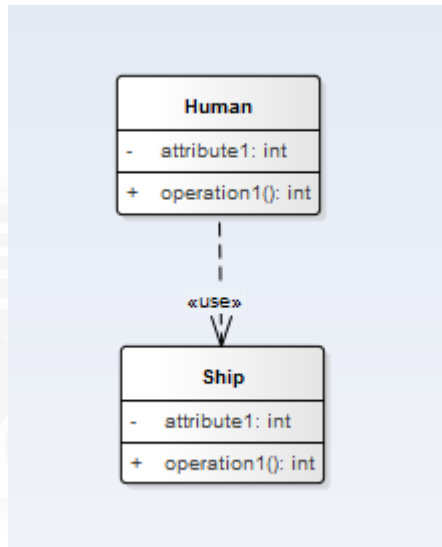
## 1. 泛化

- 泛化关系就是继承,表示一般与特殊的关系,即“一般”元素是“特殊”元素的泛化。在继承中,一般类就是父类,特殊类就是子类。
- 它们之间是“is a”的关系。
- 如:一个 Cat 类继承 Animal 类,也就是说 Cat is an Animal。在 UML 类图设计中,继承用一条带空心三角箭头的实线表示,从子类指向父类,或者子接口指向父接口。



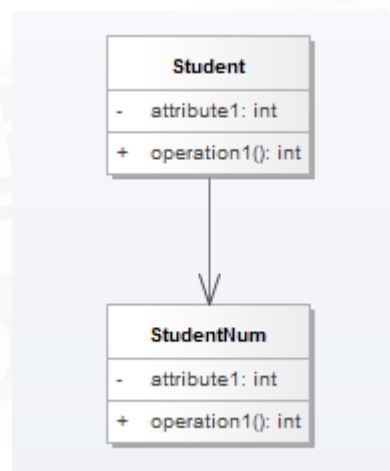
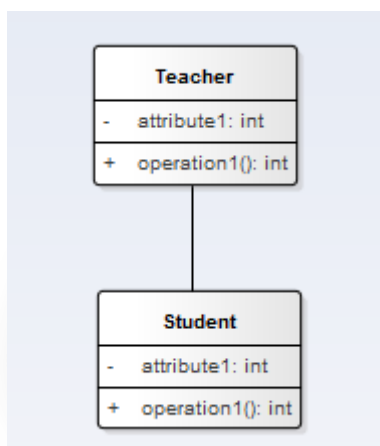
## 2. 依赖

- 表示一个元素以某种方式依赖于另一种元素。依赖就是一个类 A 使用到了另一个类 B，而这种使用关系是具有偶然性的、临时性的、非常弱的，但是类 B 的变化会影响到类 A。
- 它们之间是 “use a” 的关系。
- 依赖的例子有：一个类把另一个类的对象作为参数，一个类访问另一个类的全局变量，或者一个类调用另一个类的类操作。
- 比如某人要过河，需要借用一条船，此时人与船之间的关系就是依赖。表现在代码层面，为类 B 作为参数被类 A 在某个 method 方法中使用。在 UML 类图设计中，依赖关系用由类 A 指向类 B 的带箭头虚线表示。



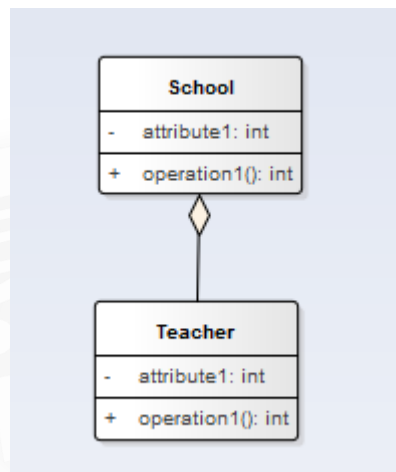
### 3. 关联

- 关联体现的是两个类之间一种强依赖关系，这种关系比依赖更强、不存在依赖关系的偶然性、关系也不是临时性的，一般是长期性的，而且双方的关系一般是平等的。关联可以是单向、双向的。
- 比如老师和学生是双向关联，老师有多名学生，学生也可能有多名老师。学生跟学生学号之间是单向关联，一个学生只能对应一个唯一的学号。在 UML 类图设计中，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。



### 4. 聚合

- 聚合关系是关联关系的一种，是强的关联关系，它体现的是整体与部分的关系，且部分可以离开整体而单独存在；关联和聚合在语法上无法区分，必须考察具体的逻辑关系。
- 它们之间是“use a”的关系。
- 此时整体与部分之间是可分离的，它们可以具有各自的生命周期，部分可以属于多个整体对象，也可以为多个整体对象共享。
- 在 UML 类图设计中，聚合关系以空心菱形加实线表示。如学校有很多老师。



## 5. 组合

- 是整体与部分的关系，是比聚合关系还要强的关系，但此时整体与部分是不可分的，整体的生命周期结束也就意味着部分的生命周期结束，如公司和部门是整体和部分的关  
系，没有公司就不存在部门。
- 它体现的是一种“contains-a”的关系。
- 在 UML 类图设计中，组合关系以实心菱形加实线表示。

