

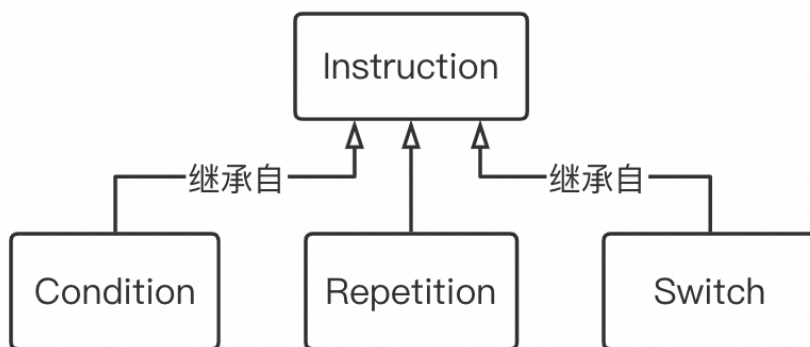
TIPS

类图的6种关系

【泛化关系 (Generalization)】

泛化关系是一种【继承】关系，表示一般与特殊的关系，它指定了子类如何特化父类的所有特征和行为。

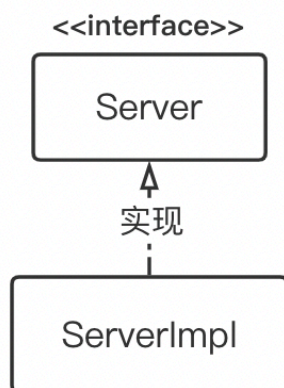
- 带空心三角箭头的实线，箭头指向父类



【实现关系 (Realization)】

实现关系是一种【类与接口】的关系，表示类是接口所有特征和行为的实现。

- 带空心三角箭头的虚线，箭头指向接口

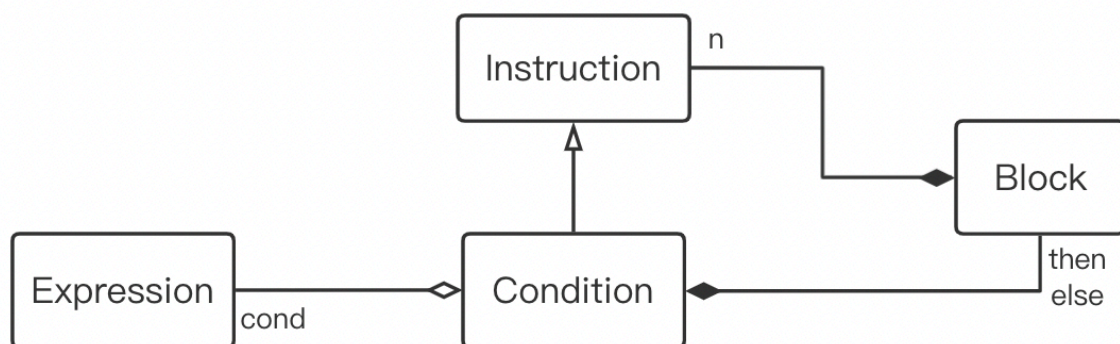


【聚合关系 (Aggregation)】

聚合关系是【整体与部分】的关系，且部分可以离开整体而单独存在。如车和轮胎是整体和部分的关系，轮胎离开车仍然可以存在。

聚合关系是关联关系的一种，是强的关联关系；关联和聚合在语法上无法区分，必须考察具体的逻辑关系。

- 代码体现：成员变量
- 带【空心菱形】的实心线，菱形指向整体



【组合关系 (Composition)】

组合关系是【整体与部分】的关系，但部分不能离开整体而单独存在。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。

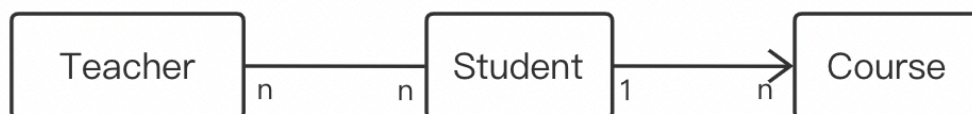
组合关系是关联关系的一种，是比聚合关系还要强的关系，它要求普通的聚合关系中代表整体的对象负责代表部分的对象的生命周期。

- 代码体现：成员变量
- 带【实心菱形】的实线，菱形指向整体（如上图）

【关联关系 (Association)】

关联关系是一种【拥有】的关系，它使一个类知道另一个类的属性和方法。如：老师与学生，关联可以是双向的，也可以是单向的。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。

- 代码体现：成员变量
- 带普通箭头的实心线，指向被拥有者



【依赖关系 (Dependency)】

依赖关系是一种使用的关系，即一个类的实现需要另一个类的协助，所以要尽量不使用双向的互相依赖。

- 代码表现：局部变量、方法的参数或者对静态方法的调用
- 带箭头的虚线，指向被使用者

