关联关系使用一条直线表示，比如 A 与 B 关联

用于描述不同类的对象之间的结构关系，将多个类的实例联系在一起

是一种静态关系，基本与程序的运行没有关系

比如，部门与员工的关系，就是关联关系

关联关系**一般不强调方向**，表示互相“知道”对方，也就是**存在引用**

关联关系有多重性 比如一对一关联 一对多关联等 可以任意关联 N 对 N 关联

如果特别强调方向，就使用箭头，比如

那么表示 A 知道 B 但是 B 不知道 A 也就是说关联关系有两种图形 ： 直线或者直线箭头

关联关系表示存在引用，比如员工类的定义中有“部门”属性字段

实现关系是带空心箭头的虚线表示的，比如 A 实现 B，箭头指向父类、接口

实现可以狭隘的认为是一种实现类与父类、接口的关系（其实在 UML 中实现的含义远不止实现类这层含义）

泛化关系是带空心箭头的直线表示的，比如 A 继承 B

用于说明继承关系

泛化关系是从子类到父类的关系，箭头指向的是父类

聚合关系是带空心的菱形的直线表示的，比如 A 聚合到 B 上，也就是 B 由 A 组成

聚合关系用于类图，表达整体由部分构成的语义，比如部门由许多人员组成

整体和部分不是强依赖的，即使整体不存在，依然可以存在部分，即使没有部门，人员仍旧存在

组合关系是带实心的菱形的直线表示的，比如 A 组合成 B，或者说 B 由 A 构成

表达整体拥有部分的含义，组合关系是一种特殊的强依赖的聚合关系

如果整体不存在，那么部分也不存在了

比如，汽车由轮胎底盘发动机构成，汽车不存在了，自然也不存在发动机了

依赖关系使用带箭头的虚线表示，比如 A 依赖 B

用于描述一个对象在运行期间会使用到另外一个对象的关系

依赖关系是一种临时性的，简言之就是不同场景会发生变化

比如人和车

如果是驾驶场景，车依赖人（驾驶员），如果是乘车出行，那就是人依赖车（公交、出租）

很显然，依赖关系比关联关系更加弱

依赖关系是一种使用关系

比如一个类的方法中的局部变量、方法的参数或者对静态方法的调用，都是一种依赖