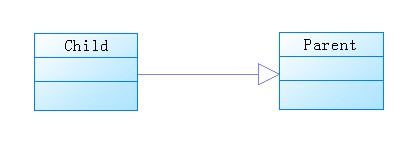
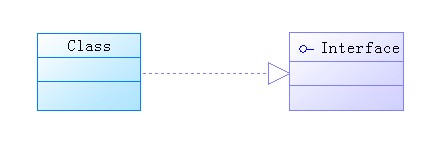
UML类图中的6种关系

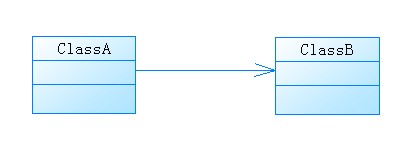
**继承**：空心三角形箭头的实线，子类指向父类。代码中通常有extends关键字（Java）。



**实现**：空心三角形箭头的虚线，实现类指向接口。代码中通常有implements关键字（Java）。



**关联**：通常是类A中需要引用另一个类B的实例作为成员变量，表示类A需要“知道”类B。实线箭头，类A指向类B。

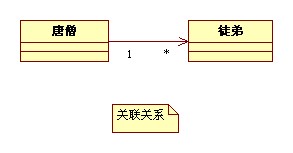


（1）.类与类之间的联接，他使一个类知道另一个类的属性和方法。

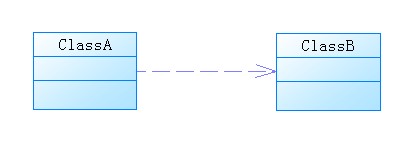
（2）.关联双向的或是单向的。

Class 徒弟 {}

Class 唐僧 {protected: list<徒弟> tdLsit; }

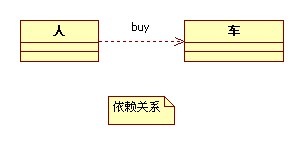


**依赖**：通常是类A中的方法需要类B的实例作为其参数或者变量，而类A本身并不需要引用类B的实例作为其成员变量。虚线箭头，类A指向类B。

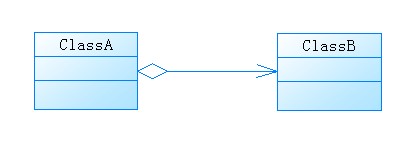


(1).类之间的链接。单向。局部变量、参数、方法的调用。

  public class Person  
   {   
   public void buy(Car car)   
   { ... }   
   }



**聚合**：通常是类A中需要包含另一个类B的实例或实例的集合，表示一种弱的“拥有”关系，即类A包含类B，但类B不是类A的一部分，而是一个独立存在的类。尾部为空心菱形的实线箭头，类A指向类B。



(1).是强的关联关系。

（2）.聚合是整体和部分之间的关系，例如汽车由引擎、轮胎及其他零件组成。

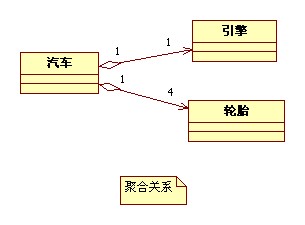
(3).聚合关系也是通过成员变量实现的。关联两个类同意层次，聚合为不同层次。

(4).必须考虑所涉及的类之间的逻辑关系。

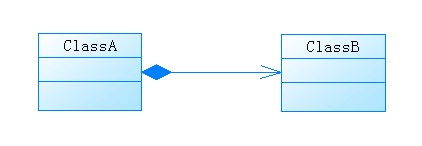
Class 引擎 {}

Class 轮胎 {}

Class 汽车{protected:引擎 engine; 轮胎 tyre[4];}



**合成**：通常需要在实例化类A的同时在其构造方法之中实例化类B，使类B作为类A的一部分，表示一个强的“拥有”关系，即类B是类A的一部分，二者共生命周期，具有严格的“整体—部分”关系。尾部为实心菱形的实现箭头，类A指向类B。

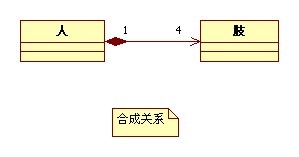


（1）.比聚合还要强的关系。

（2）.整体对象负责部分对象生命周期。

Class 肢 {}

Class 人{ protected：肢 limb[4]}



总结：

对于聚合关系、合成关系，由于都是关联关系的一种，因此在不确定的情况下，可以以关联关系来描述它们。