

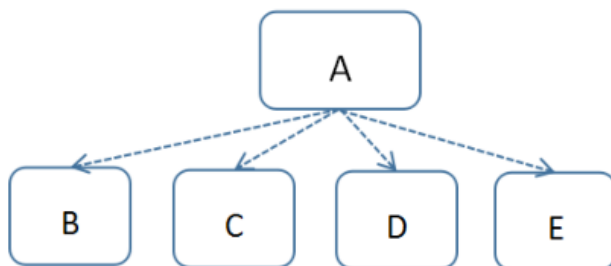
对象的关联—>简单地说，是指一个对象中使用了另一个对象。

## 1、依赖关系 (Dependency)

对象之间最弱的一种关联方式，是临时性的关联。代码中一般指由局部变量、函数参数、返回值建立的对于其他对象的调用关系。

```
class A{  
    public B method(C c,D d){  
        E e = new E();  
        ...  
        B b = new B();  
        ...  
        return b;  
    }  
}
```

这个代码结构中，表示 A 类依赖了 B,C,D,E 类



## 2、关联关系 (Association)

对象之间一种引用关系，比如客户类与订单类之间的关系。这种关系通常使用类的属性表达。关联可以有方向，即导航。一般不作说明的时候，导航是双向的，不需要在线上标出箭头。

大部分情况下导航是单向的，可以加一个箭头表示。

```
class Employee{
    private int eid;//员工编号
    private String name;//员工姓名
    private Computer coumputer;//员工所使用的电脑
    //....
}
class Computer{
}
```



```
class Husband{
    private Wife wife;
}
class Wife{
    private Husband husband;
}
```



关联表示类之间的“持久”关系，这种关系一般表示一种重要的业务之间的关系，需要保存的，或者说需要“持久化”的，或者说需要保存到数据库中的。另外，依赖表示类之间的是一种“临时、短暂”关系，这种关系是不需要保存的。

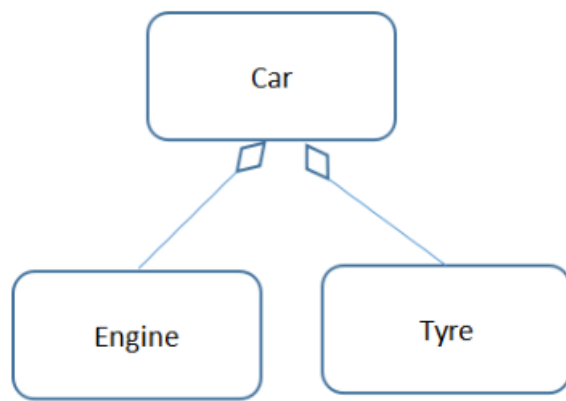
### 3、聚合 (Aggregation )

聚合（关联关系的一种）：表示 has-a 的关系。与关联关系一样，聚合关系也是通过实例变量来实现这样关系的。关联关系和聚合关系来语法上是没办法区分的，从语义上才能更好的区分两者的区别。

如汽车类与引擎类，轮胎类之间的关系就是整体与个体的关系。

与关联关系一样，聚合关系也是通过实例变量来实现的。空心菱形

```
class Car{
    private Engine engine;//引擎
    private Tyre[] tyres;//轮胎
}
```



关联和聚集（聚合）的区别：

关联关系所涉及的两个对象是处在同一个层次上的。比如人和自行车就是一种关联关系，而不是聚合关系，因为人不是由自行车组成的。

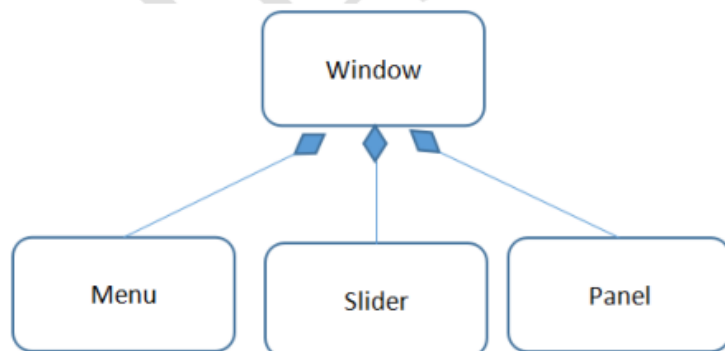
聚合关系涉及的两个对象处于不平等的层次上，一个代表整体，一个代表部分。比如电脑和它的显示器、键盘、主板以及内存就是聚集关系，因为主板是电脑的组成部分。

## 4、组合 (Composite)

对象 A 包含对象 B，对象 B 离开对象 A 没有实际意义。是一种更强的关联关系。人包含手，手离开人的躯体就失去了它应有的作用。

组合：表示 `contains-a` 的关系，是一种强烈的包含关系。组合类负责被组合类的生命周期。也使用属性表达组合关系，是关联关系的一种，是比聚合关系强的关系。

```
class Window{
    private Menu menu;//菜单
    private Slider slider;//滑动条
    private Panel panel;//工作区
}
```



## 5、继承 (Generalization , 又称为泛化, is-a 的关系)

类与类的继承关系，类与接口的实现关系。

场景：父与子、动物与人、植物与树

