[**详解UML中的6大关系(关联、依赖、聚合、组合、泛化、实现)**](http://www.yongfa365.com/item/UML-Association-Dependency-Aggregation-Composition-Realization-Generalization.html)

[类别：.Net + C#](http://www.yongfa365.com/Csharp/)　|　[浏览(11778)](http://www.yongfa365.com/Item/UML-Association-Dependency-Aggregation-Composition-Realization-Generalization.html###)　|　2010-10-24 22:00:49

更多[9](http://www.yongfa365.com/Item/UML-Association-Dependency-Aggregation-Composition-Realization-Generalization.html)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 大话设计模式上的一个图，我用EA画出来的：  大话设计模式类图.PNG  **UML中的6大关系相关英文及音标：**   |  |  |  | | --- | --- | --- | | 依赖关系 | dependency | [di'pendənsi] | | 关联关系 | association | [ə,səuʃi'eiʃən] | | 聚合关系 | aggregation | [ˌægrɪˈgeɪʃən] | | 组合关系 | composition | [,kɔmpə'ziʃən] | | 实现 | realization | [,ri:əlɪ'zeɪʃən] | | 泛化 | generalization | [,dʒenərəlɪ'zeɪʃən] |       **UML中的6大关系简单解释：**   1. 关联：连接模型元素及链接实例，用一条实线来表示； 2. 依赖：表示一个元素以某种方式依赖于另一个元素，用一条虚线加箭头来表示； 3. 聚合：表示整体与部分的关系，用一条实线加空心菱形来表示； 4. 组成：表示整体与部分的有一关系，用一条实线加实心菱形来表示； 5. 泛化（继承）：表示一般与特殊的关系，用一条实线加空心箭头来表示； 6. 实现：表示类与接口的关系，用一条虚线加空心箭头来表示；   注意：泛化关系和实现关系又统称为一般关系； 总之：一般关系表现为继承或实现(is a)，关联关系、聚合关系、合成/组合关系表现为成员变量(has a)，依赖关系表现为函数中的参数(use a)；  转自：<http://www.cnblogs.com/ForEverKissing/archive/2007/12/13/993818.html>  **UML中的6大关系详细说明：**  **1、关联关系：** 　　含义：类与类之间的连结，关联关系使一个类知道另外一个类的属性和方法；通常含有“知道”，“了解”的含义 体现：在C#中，关联关系是通过成员变量来实现的； 方向：双向或单向； 图示：实线 + 箭头；箭头指向被关联的类； 举例：“渔民”需要知道“天气”情况才能够出海 　　　//公司关联雇员 public class Company {     private Employee employee;     public Employee GetEmployee()     {         return employee;     }     public void SetEmployee(Employee employee)     {         this.employee = employee;     }     //公司运作         public void Run()     {         employee.StartWorking();     }  }  //A关联B class A {     B b = new B(); } class B { }  **2、依赖关系：** 　　含义：是类与类之间的连接，表示一个类依赖于另外一个类的定义；依赖关系仅仅描述了类与类之间的一种使用与被使用的关系； 体现：在C#中体现为局部变量、方法/函数的参数或者是对静态方法的调用； 方向：单向； 图示：虚线 + 箭头； 举例：人依赖于水和空气；汽车依赖汽油 　　　　//人划船，人依赖于船 public class Person {     //划船     public void Oarage(Boat boat)     {         boat.Oarage();     } }   //A依赖于B class A {     public void Function(B b)     { } } class B { }    **3、聚合关系：** 　　含义：是关联关系的一种，是一种强关联关系；聚合关系是整体和个体/部分之间的关系；关联关系的两个类处于同一个层次上，而聚合关系的两个类处于不同的层次上，一个是整体，一个是个体/部分；在聚合关系中，代表个体/部分的对象有可能会被多个代表整体的对象所共享； 体现：C++中，聚合关系通过将被聚合者的（数组）指针作为内部成员来实现的； 方向：单向； 图示：空心菱形 + 实线 + 箭头；箭头指向被聚合的类，也就是说，箭头指向个体/部分； 举例：鸭群与鸭子具有聚合关系；汽车由引擎、轮胎以及其它零件组成，因为汽车坏掉了，没有坏掉的引擎，轮胎和其他零件还可以继续使用。  **4、组合关系：** 　　含义：它也是关联关系的一种，但它是比聚合关系更强的关系.组合关系要求聚合关系中代表整体的对象要负责代表个体/部分的对象的整个生命周期；组合关系不能共享；在组合关系中，如果代表整体的对象被销毁或破坏，那么代表个体/部分的对象也一定会被销毁或破坏，而聚在合关系中，代表个体/部分的对象则有可能被多个代表整体的对象所共享，而不一定会随着某个代表整体的对象被销毁或破坏而被销毁或破坏； 体现：在C#中，组合关系是通过成员变量来实现的； 方向：单向； 图示：实心菱形 + 实线 + 箭头；箭头指向代表个体/部分的对象，也就是被组合的类的对象； 举例：一个人由头、四肢、等各种器官组成，因为人与这些器官具有相同的生命周期，人死了，这些器官也挂了；  **5、泛化关系：** 　　含义：它表示一个更泛化的元素和一个更具体的元素之间的关系；也就是通常所说的类的继承关系； 体现：在C#中，泛化关系通过类的继承来实现的； 方向：单向；子类继承父类； 图示：空心箭头 + 实线；箭头指向父类； 举例：动物下面可以分为哺乳动物，脊椎动物，爬行动物等  **6、实现关系：** 　　含义：它指定了两个实体之间的一份合同；即：一个实体定义一份合同，另外一个实体则保证履行该合同； 体现：在C#中，实现关系通过类实现接口来实现的，即：一个类实现某个接口； 方向：单向；子类实现接口； 图示：空心箭头 + 虚线；箭头指接口向接口； 举例：唐老鸭（对象）会说话（接口），因为一般鸭子不会说话，所以不会将说话这个方法给一般的鸭子带上；超人（对象）会飞（接口）  转自：<http://www.cnblogs.com/aicro/archive/2010/08/23/1806584.html>  **UML中的6大关系参考：**  你是我的玫瑰-类关系阐微  <http://www.cnblogs.com/zjzkiss/archive/2007/05/06/736725.html>  UML中关联、依赖、聚集等关系的异同 <http://www.cnblogs.com/ForEverKissing/archive/2007/12/13/993818.html>  UML 类与类之间的关系 <http://yanwenhan.javaeye.com/blog/149450>  类间的关系 <http://www.cnblogs.com/floodpeak/archive/2008/02/27/1083533.html>  UML解惑：图说UML中的六大关系 <http://developer.51cto.com/art/201004/194204.htm>  大话设计模式  引用： [详解UML中的6大关系(关联、依赖、聚合、组合、泛化、实现)](http://www.yongfa365.com/item/UML-Association-Dependency-Aggregation-Composition-Realization-Generalization.html) <http://www.yongfa365.com/item/UML-Association-Dependency-Aggregation-Composition-Realization-Generalization.html> |