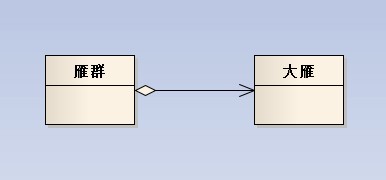
大家都知道UML的类图中一般包含五种关系即  关联 聚合 组合 泛化 依赖  有些人可能会感觉组合跟聚合有点难区分 说难其实是相对其他几种关系而言 实际上想分清这两种关系一点也不难

聚合跟组合其实都属于关联 只不过它们是两种特殊的关联 因为本是同根生 所以它们之间难免会有相似之处 下面让我们一起来看一下它们之间有何不同

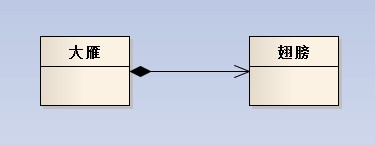
聚合与组合的概念相信不用我在此赘述大家就已经了解了 下面直接上例子

程老师的《大话》里举大那个大雁的例子很贴切 在此我就借用一下 大雁喜欢热闹害怕孤独 所以它们一直过着群居的生活 这样就有了雁群 每一只大雁都有自己的雁群 每个雁群都有好多大雁 大雁与雁群的这种关系就可以称之为聚合 另外每只大雁都有两只翅膀 大雁与雁翅的关系就叫做组合 有此可见 聚合的关系明显没有组合紧密 大雁不会因为它们的群主将雁群解散而无法生存 而雁翅就无法脱离大雁而单独生存——组合关系的类具有相同的生命周期

聚合关系图：



组合关系图：



从从代码上看这两种关系的区别在于：

构造函数不同

雁群类：

**[csharp]** [view plain](http://blog.csdn.net/liushuijinger/article/details/6994265) [copy](http://blog.csdn.net/liushuijinger/article/details/6994265)

1. **public**  **class** GooseGroup
2. {
3. **public** Goose goose;

6. **public** GooseGroup(Goose goose)
7. {
8. **this**.goose = goose;
9. }
10. }

大雁类：

**[csharp]** [view plain](http://blog.csdn.net/liushuijinger/article/details/6994265) [copy](http://blog.csdn.net/liushuijinger/article/details/6994265)

1. **public** **class** Goose
2. {
3. **public** Wings wings;
5. **public** Goose()
6. {
7. wings=**new** Wings();
8. }
9. }

聚合关系的类里含有另一个类作为参数  
  
雁群类（GooseGroup）的构造函数中要用到大雁（Goose）作为参数把值传进来 大雁类（Goose）可以脱离雁群类而独立存在   
  
组合关系的类里含有另一个类的实例化  
  
大雁类（Goose）在实例化之前 一定要先实例化翅膀类（Wings） 两个类紧密耦合在一起 它们有相同的生命周期 翅膀类（Wings）不可以脱离大雁类（Goose）而独立存在  
  
信息的封装性不同  
  
在聚合关系中，客户端可以同时了解雁群类和大雁类，因为他们都是独立的  
  
而在组合关系中，客户端只认识大雁类，根本就不知道翅膀类的存在，因为翅膀类被严密的封装在大雁类中。