Atitit [面向对象编程（OOP）、面向组件编程（COP）、面向方面编程（AOP）和面向服务编程（SOP）](http://blog.csdn.net/ocean181/article/details/6720371)的区别和联系

# 面向组件编程（COP）

所以，组件比起对象来的进步就在于通用的规范的引入。通用规范往往能够为组件添加新的能力（就像上面所讨论的），

COP比OOP更进一步。通常OOP将数据对象组织到实体中。这种方法具有很多优点。但是，OOP有一个大的限制：对象之间的相互依赖关系。去掉这个限制的一个好的想法就是组件。组件和一般对象之间的关键区别是组件是可以替代的。

1. **什么是面向方面编程（Aspect-Oriented Programming）？**   
   将通用需求功能从不相关类之中分离出来；同时，能够使得很多类共享一个行为，一旦行为发生变化，不必修改很多类，只要修改这个行为就可以。   
   AOP就是这种实现分散关注的编程方法，它将“关注”封装在“方面”中。

作者::  ★(attilax)>>>  绰号:老哇的爪子 （ 全名：：Attilax Akbar Al Rapanui 阿提拉克斯 阿克巴 阿尔 拉帕努伊 ） 汉字名：艾龙，  EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

1. **什么是面向服务编程（Service-Oriented Programming）？**   
   SOP是一种体系结构，目标是在软件代理交互中获得松散耦合

这里的好处就是你可以用不同的CD机去播放同一张CD。他们能提供同样的CD播放服务，但是服务质量是不同的。  
    SOP的思想明显不同于面向对象的编程，面向对象编程强烈的建议你应该将数据与其操作绑定。因此在面向对象编程风格中，每张CD 有它自己的CD播放机，他们之间不能被拆开。这听起来很奇怪，但是这就是我们建立许多已存软件系统的方式。  
而SOP就不一样了，为了减少异构性、互操作性和不断改变的要求的问题，这样的体系结构应该提供平台来构建具有下列特征的应用程序服务：   
松散耦合、位置透明、协议独立

   基于这样的面向服务的体系结构，服务使用者甚至不必关心与之通信的特定服务，因为底层基础设施或服务“总线”将代表使用者做出适当的选择。基础设施对请求者隐藏了尽可能多的技术。特别地，来自不同实现技术（如 J2EE 或 .NET）的技术规范不应该影响 SOP用户。如果已经存在一个服务实现，我们就还应该重新考虑用一个“更好”的服务实现来代替，新的服务实现必须具有更好的服务质量。

参考资料

面向对象编程（OOP）、面向组件编程（COP）、面向方面编程（AOP）和面向服务编程（SOP） - 始则转俗成真，终乃回真向俗！ - 博客频道 - CSDN.NET.html