**iOS 组件化**

AOP: Aspect Oriented Programming 面向切面编程。

面向切面编程(也叫面向方面)：Aspect Oriented Programming(AOP),是目前软件开发中的一个热点。利用AOP可以对业务逻辑的各个部分进行隔离，从而使得业务逻辑各部分之间的耦合度降低，提高程序的可重用性，同时提高了开发的效率。

AOP是OOP的延续，是（Aspect Oriented Programming）的缩写，意思是面向切面（方面）编程。

主要的功能是：日志记录，性能统计，安全控制，事务处理，异常处理等等。

主要的意图是：将日志记录，性能统计，安全控制，事务处理，异常处理等代码从业务逻辑代码中划分出来，通过对这些行为的分离，我们希望可以将它们独立到非指导业务逻辑的方法中，进而改 变这些行为的时候不影响业务逻辑的代码。

**AOP的优势：**

1. 减少切面业务的开发量，“一次开发终生使用”，比如日志
2. 减少代码耦合，方便复用。切面业务的代码可以独立出来，方便其他应用使用
3. 提高代码review的质量，比如我可以规定某些类的某些方法才用特定的命名规范，这样review的时候就可以发现一些问题

**AOP的弊端：**

1. 它破坏了代码的干净整洁。  
   （因为 Logging 的代码本身并不属于 ViewController 里的主要逻辑。随着项目扩大、代码量增加，你的 ViewController 里会到处散布着 Logging 的代码。这时，要找到一段事件记录的代码会变得困难，也很容易忘记添加事件记录的代码）