

AOP的概念

2021年10月9日 16:52

AOP的大致概念

AOP——**Aspect Oriented Programming**，面向切面编程

高内聚低耦合是面向对象设计的所追求的终极目标，但其实无论我们怎样设计，都无法解决以下这样的耦合问题：

如——系统中的权限校验、写日志功能

这些功能如沙子一样混杂在系统中，因为系统中很多地方会调用它们，系统会和这些功能紧密地耦合在一起

如果我们想把系统的主要业务和这些零散功能隔离开，用面向对象、面向过程的方法，都是无法做到的

面向切面编程，即AOP编程式解决这样问题的最有效方式

利用AOP，可以把这些零散功能放在一个独立的地方，然后用定义的方式，把它在系统运行或系统编译时再和系统耦合起来，这样就彻底把这些所谓的功能与系统主要业务隔离开了，这就是AOP编程技术

面向切面编程的实现方式

1. 使用面向切面编程语言，如AspectJ
2. 在现有框架上实现切面编程

AOP术语

1. Advice

Advice其实就是上面所说的沙子，即需要插到系统主要业务中的零散功能

Advice中需要包含两个内容——要做什么、在什么时候做

2. Join Point

Join Point是我们要掺沙子的点，即需要插入零散功能的地方

需要插入的地方其实是比较多的，所以我们通常需要一个定义的方式去定义这些点是什么点，这叫做Pointcut

3. Pointcut

Pointcut也叫作切点，它定义了一个或多个Join Point

它通常以通配符的方式来定义Join Point

4. Aspect

Aspect，即切面，它包含了上面的三个概念

Aspect = Advice + Pointcut，Advice表示要做什么、在什么时候做，Pointcut定义这些插入的地方在哪

5. Target

Target是我们要掺沙子的对象

Target是没有沙子的目标，只有主动的业务，把要掺进去的东西交给了Aspect去定义，在编译和运行时再把沙子插入到主系统中来