**春天的杰作Spring源码分析**

主讲：Cat 老师

北京动力节点教育科技有限公司  
2020 • 北京

动力节点•版权所有•禁止传播

## Spring源码分析课程说明

今天讲Spring AOP源码 + 手写Spring AOP，一次课；

Spring + MyBatis源码，一次课；

原计划Spring事务一次课，这个内容打算结合到MySQL事务和锁机制里面去单独讲解，这样更加连贯，所以Spring事务这块内容会放在MySQL优化里面讲这一块内容，会讲解事务、锁机制，再结合spring、springboot操作；

## Spring Framework三大核心

（1）、IoC 控制反转；

（2）、AOP 面向切面编程；

（3）、MVC 面向web开发的mvc框架；

## IoC

所谓 IOC ，就是由 Spring IOC 容器来负责管理对象的生命周期和对象之间的关系；

## MVC

MVC（模型、视图、控制器），是Spring基于Servlet 推出的MVC框架，主要解决WEB开发的问题，可以处理WEB开发的路径映射和视图跳转渲染等；

## AOP

AOP就是把一些公共的逻辑与主业务逻辑相分离，从而解除代码耦合；

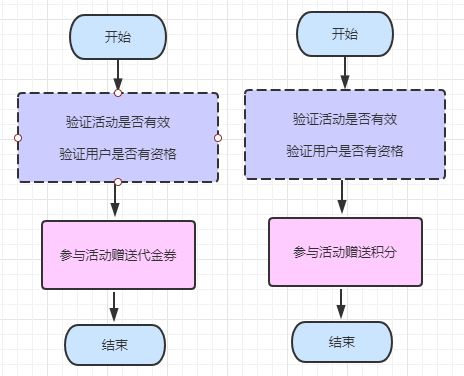
**OOP（面向对象编程）**主要是通过封装、继承等方式实现的代码逻辑是从上而下的，在从上而下的代码逻辑中，如果我想插入一段公共的逻辑（比如记录日志，打印程序运行时间，对方法结果进行缓存，对方法操作进行事务控制等），那么就会造成每个方法都需要写一段相同的逻辑，而这些逻辑与主业务的关系不大，这样让我们的代码变得难以维护，公共逻辑散落一地；

我们把这些公共的逻辑称为**切面；**

AOP就是把这些公共的逻辑与主业务逻辑相分离，从而解除代码耦合；

AOP是一种编程思想，通过这种思想达到我们编程想要实现的某些目的；

Spring 是AOP这种思想的具体落地实现，用Spring AOP就可以目的；



## Spring AOP核心流程

<https://www.eclipse.org/aspectj/>

1. 实现aop，可以采用aspectj注解方式，也可以xml配置方式； <aop:aspectj >

<aop:config>

1. 注解方式或者xml方式都是第一步解析我们的注解配置或者xml配置，根据这些配置得到BeanDefinition，然后把BeanDefinition注册放到一个BeanDefinitionMap中，其实这里依然是IOC；
2. 根据BeanDefinitionMap里边的BeanDefinition实例化bean对象，AOP在创建好bean对象、给bean对象赋好值之后，会使用BeanPostProcessor对实例化后的bean对象进行修改，在修改里面使用动态代理技术：jdk动态代理（invoke方法），cglib（底层基于asm字节类库操作，它是动态创建一个类继承于我们的目标类，覆盖我们目标类的方法），所以最终返回的bean是一个$Proxy0@68451代理类； @smsServiceImpl@@SpringForCGlibxxxxx@1545666
3. 当在调用目标类的方法时会被代理对象的方法拦截，在拦截里面实现方法功能的增强（前置增强、后置增强、环绕增强等）；

Spring aop + 整合了aspectj （主要是简化aop的开发，基于aspectj注解配置，让aop开发更简单一些）+ 动态代理技术；