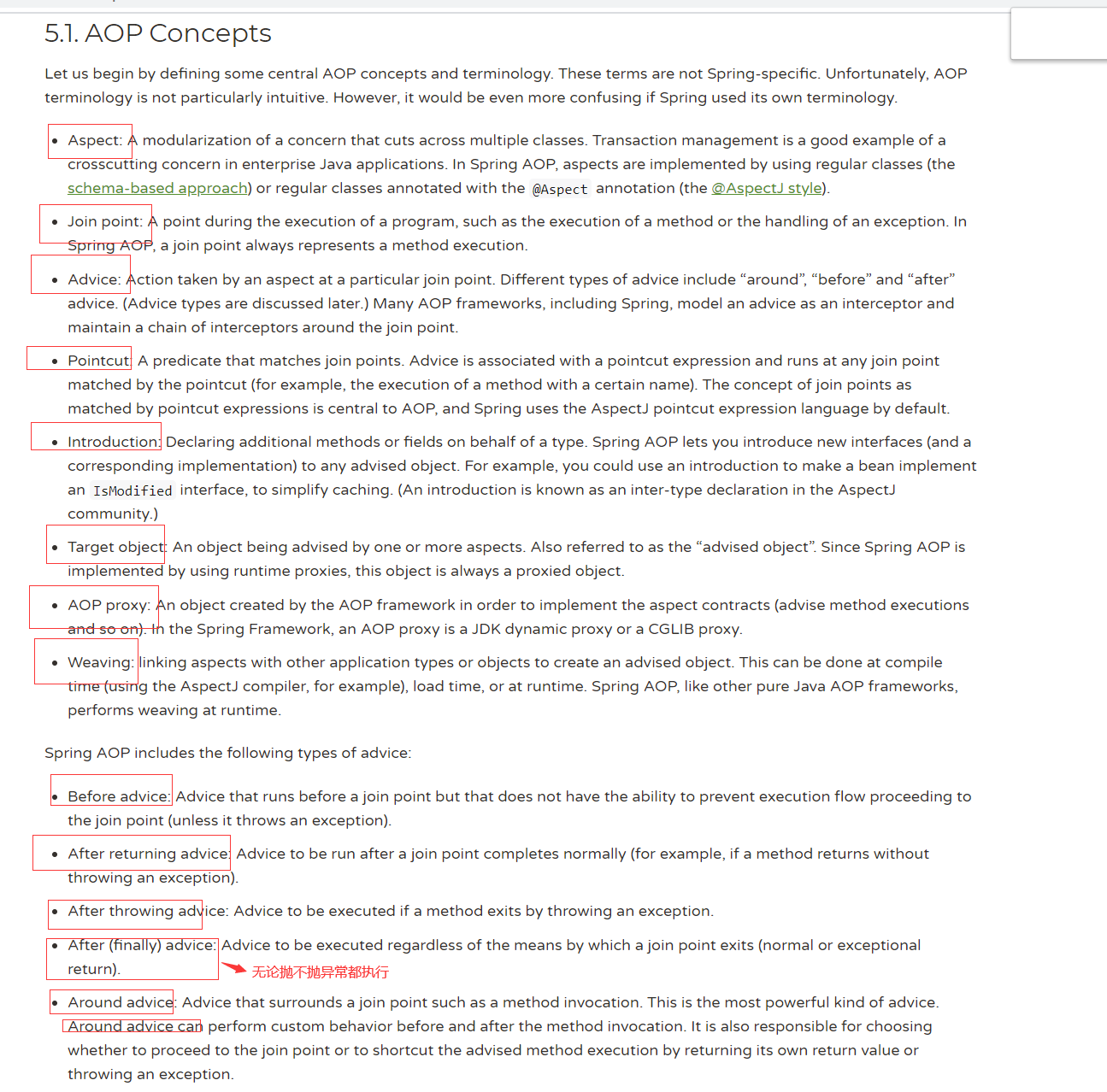
# SpringAOP

# spring framework官方文档介绍

[https://docs.spring.io/spring/docs/5.1.3.RELEASE/spring-framework-reference/core.html#aop](https://docs.spring.io/spring/docs/5.1.3.RELEASE/spring-framework-reference/core.html" \l "aop)



# 分析AOP

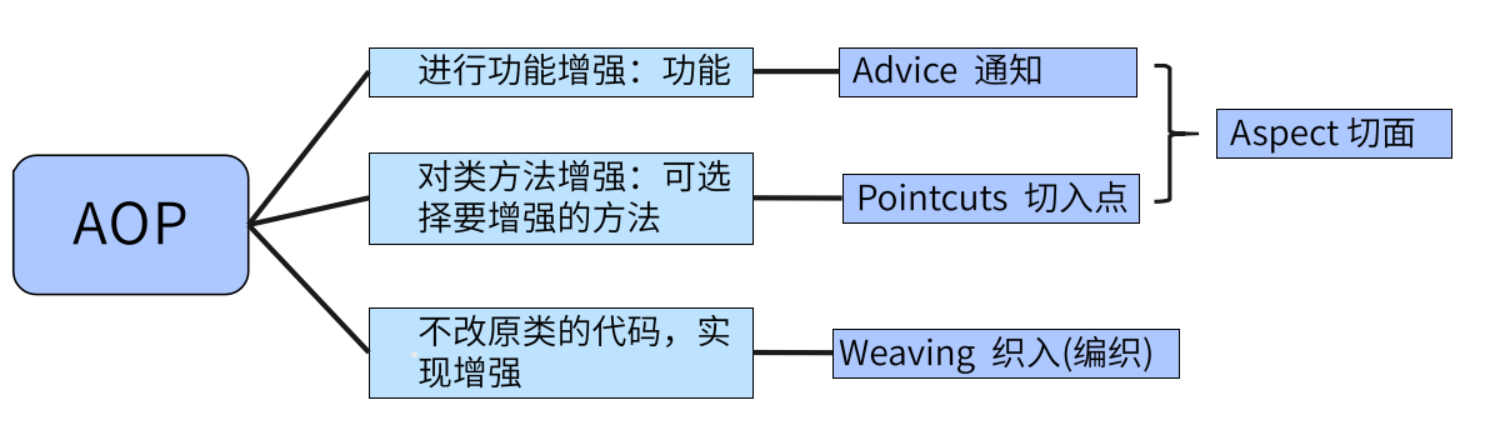
1. AOP是什么

Aspect Oriented Programming 面向切面编程，在不改变类的代码情况下，对类方式进行功能增强。

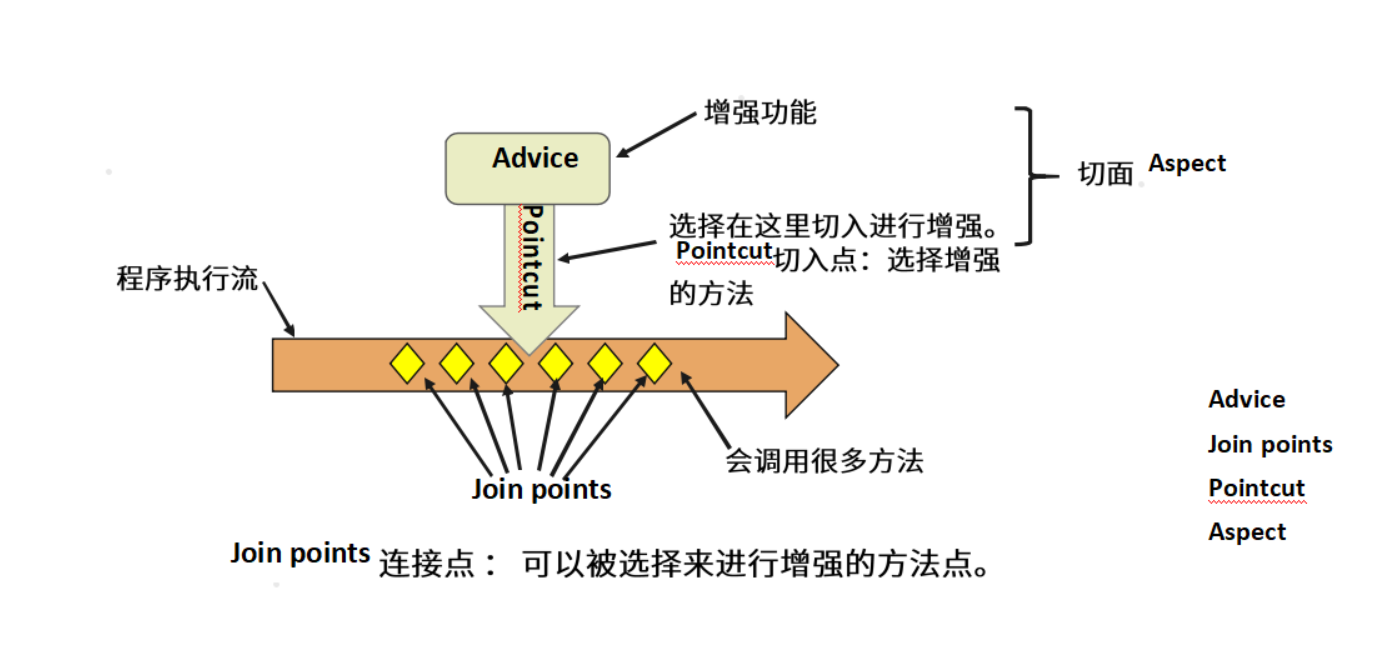
1. AOP能做什么

我们的框架中要想使用用户提供的AOP功能，让他们可以通过AOP技术实现对类方法进行功能增强。

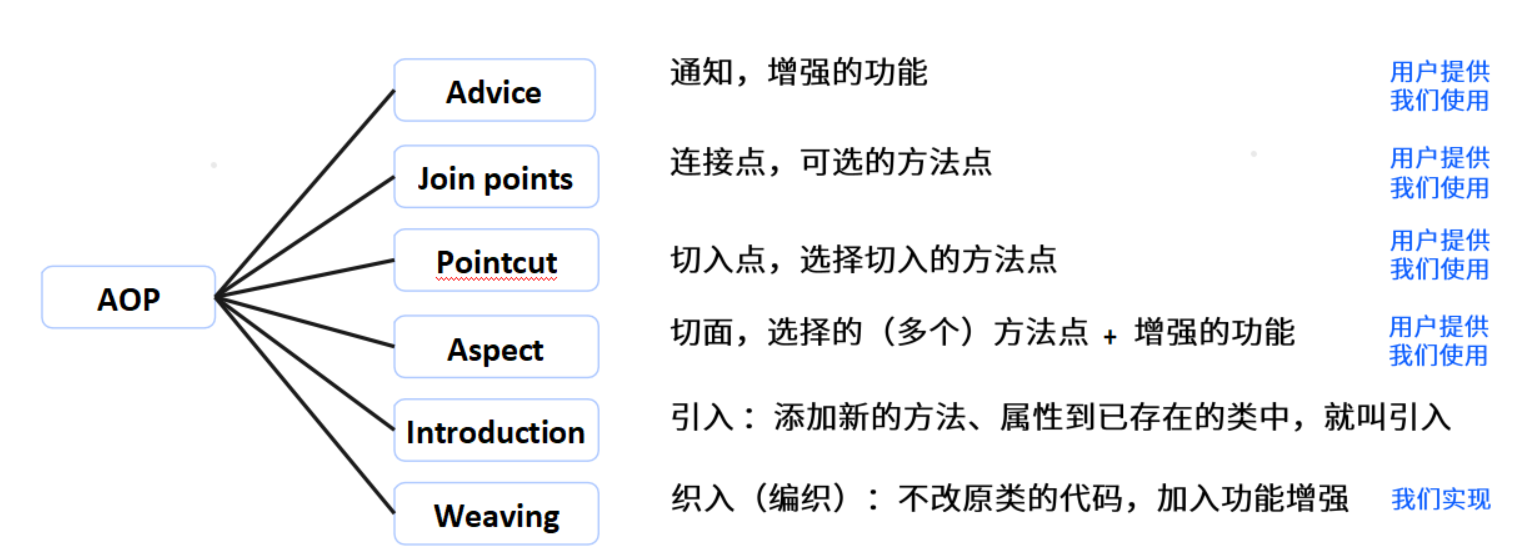
Aspect Oriented Programming 面向切面编程。



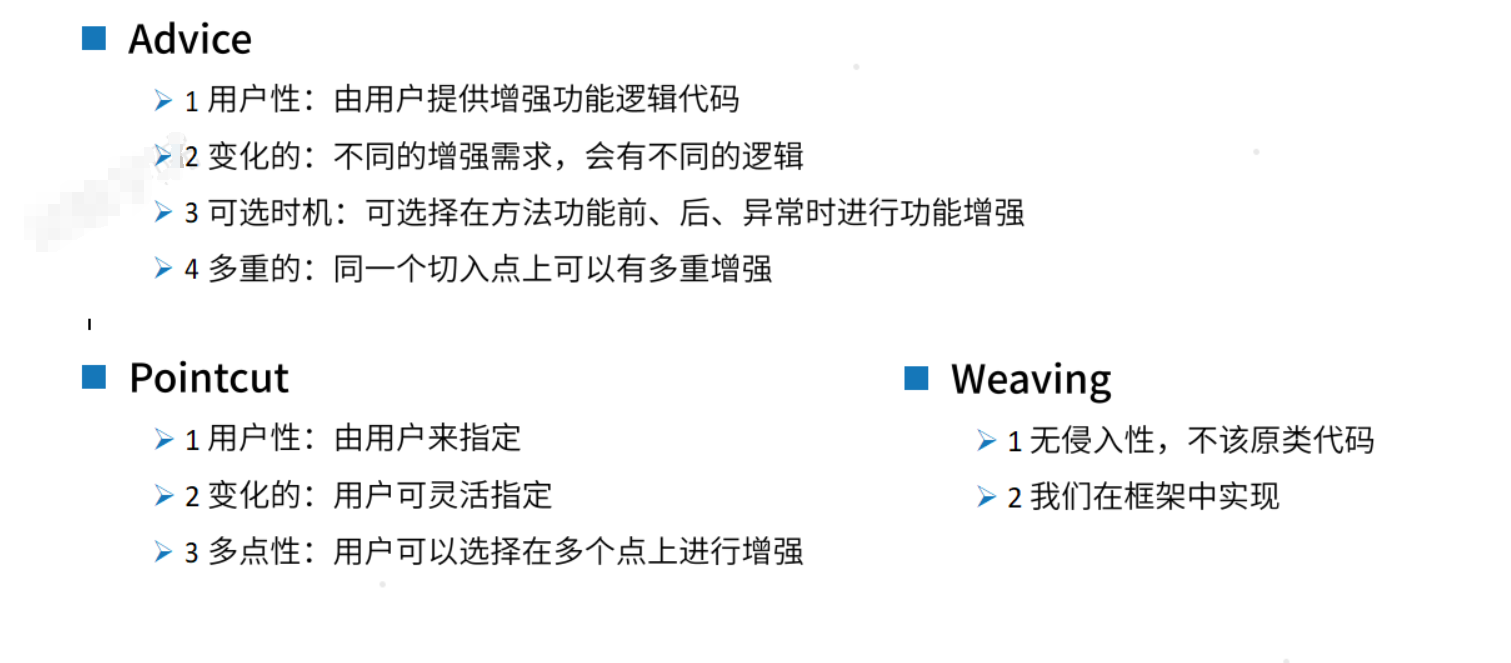
# AOP概念



思考:上述哪些方式使用户提供的，哪些是我们自己要写？



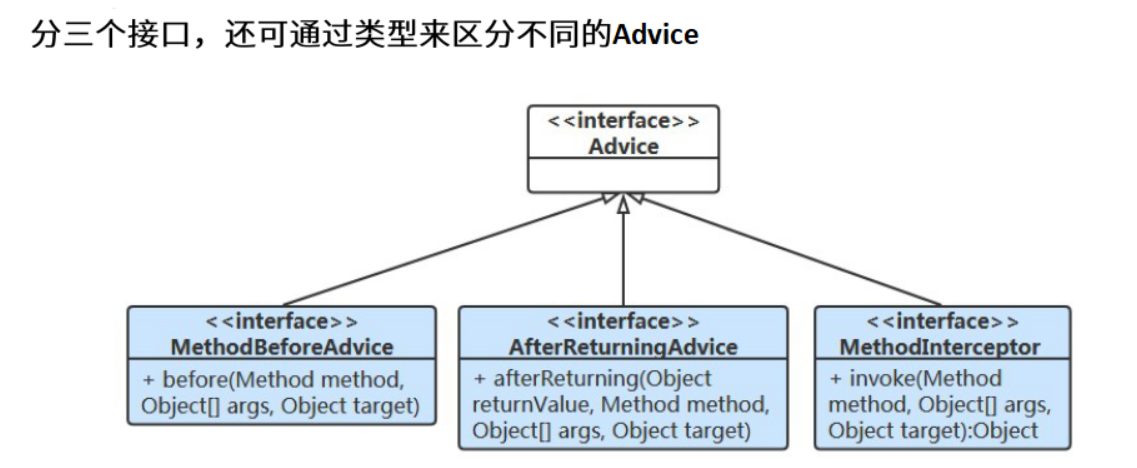
Advice(通知)、Pointcut(切入点)、Weaving(织入)的特点：



# 切面实现

Advice设计

特点：前置增强，后置增强，环绕增强，异常处理增强



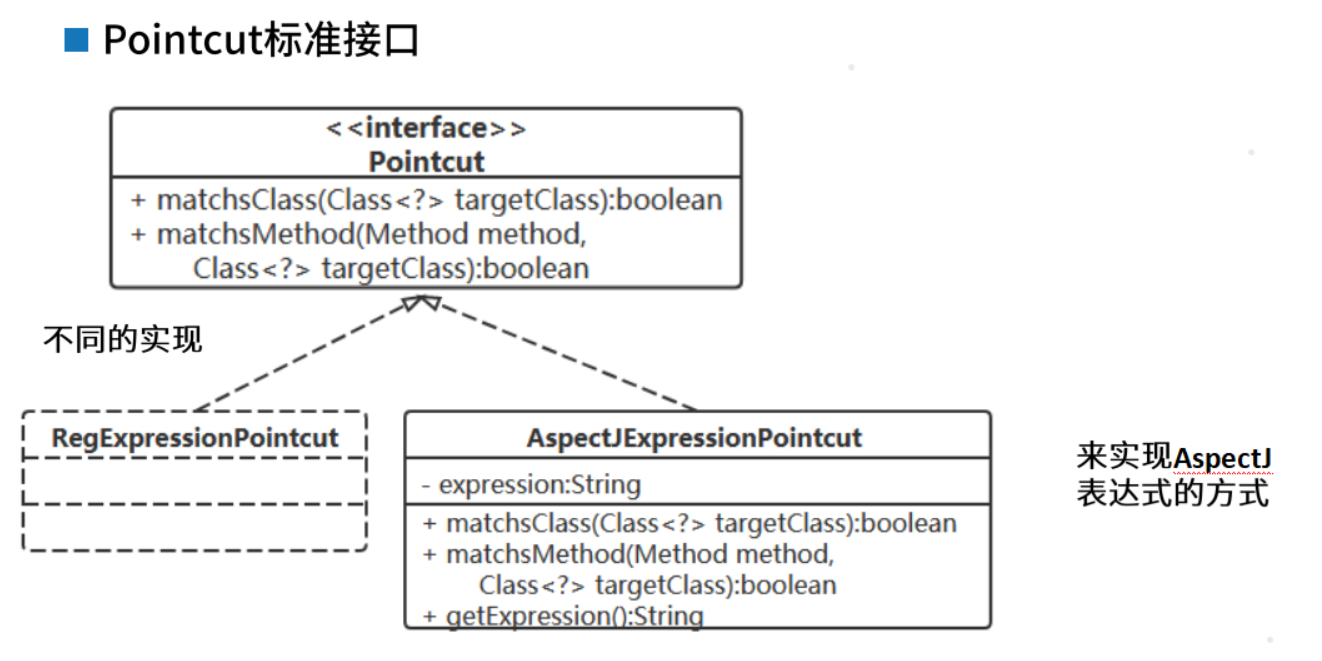
Pointcut 设计

1. 完整的签名方法 com.\*.xx(参数)
2. 根据类型重载方法
3. 正则表达式

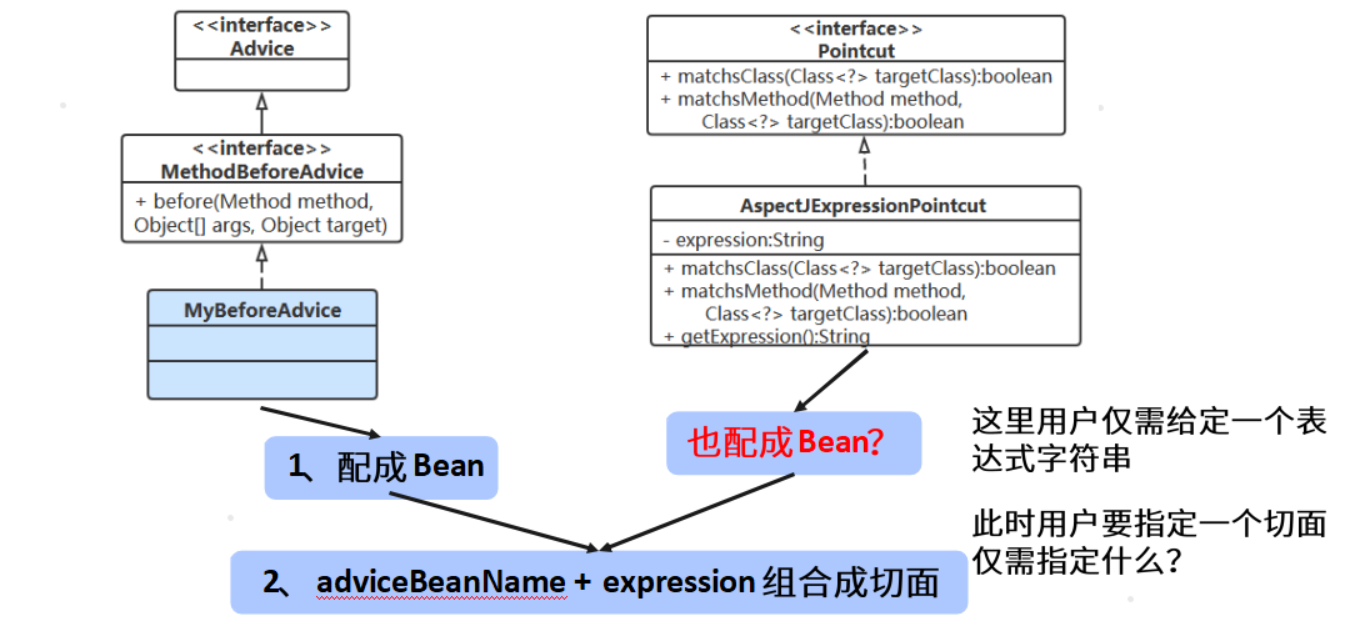
Aspectj的poincut表达式 例如：execution(\* com.wywhdgg.service.AccountService.\*(..))

[https://docs.spring.io/spring/docs/5.1.1.RELEASE/spring-framework-]("https://docs.spring.io/spring/docs/5.1.1.RELEASE/spring-framework-)

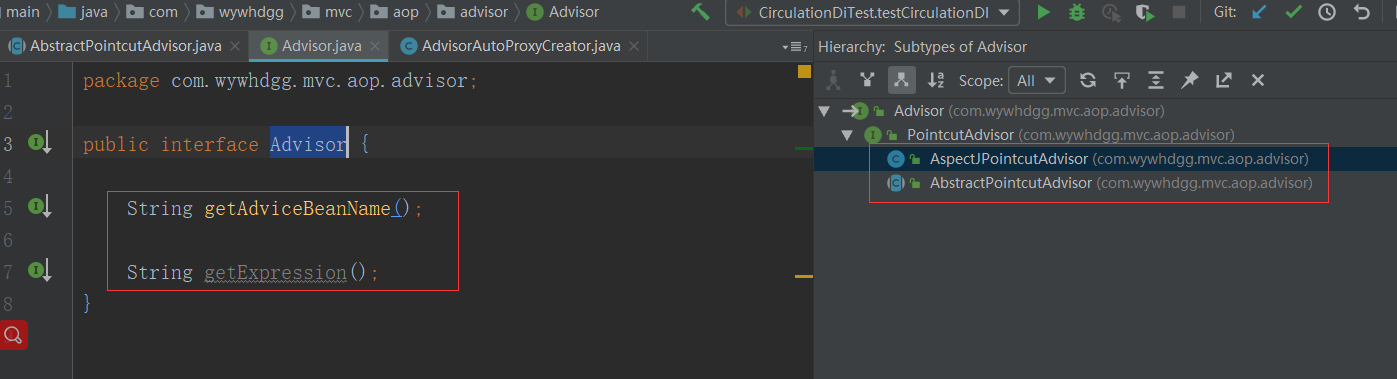
[reference/core.html#aop-pointcuts]("https://docs.spring.io/spring/docs/5.1.1.RELEASE/spring-framework-)



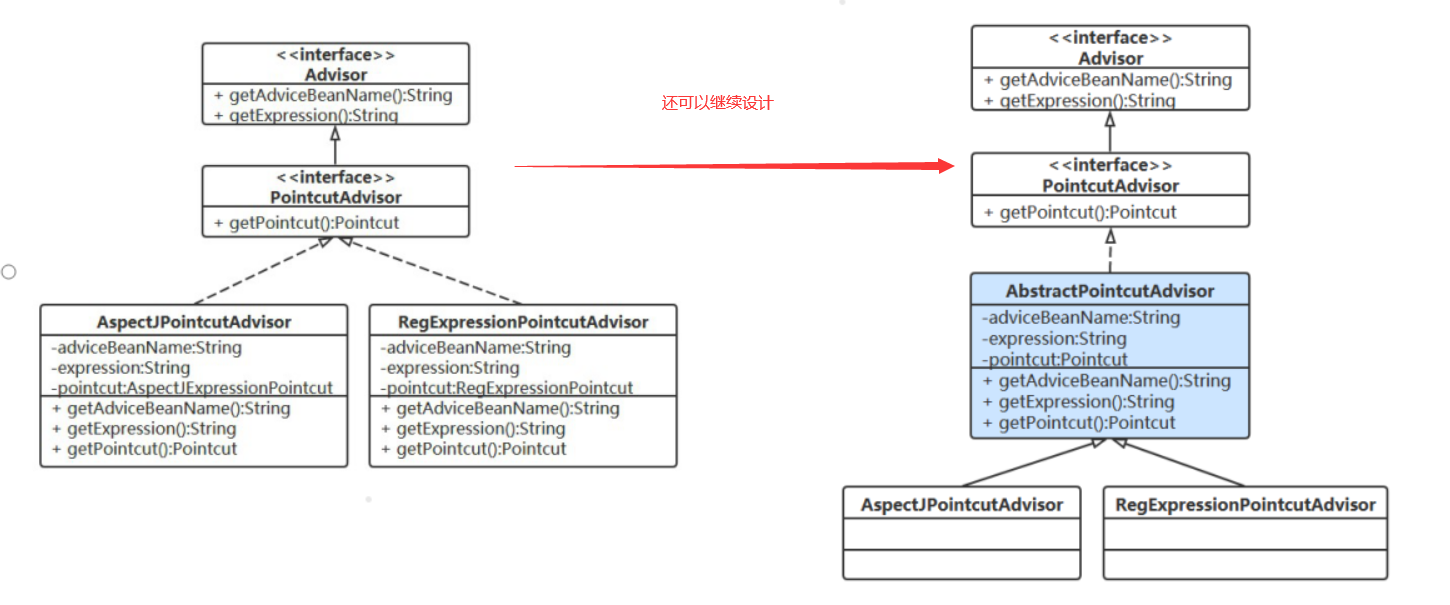
Advice Pointcut 使用



我们可以抽象为一个接口：



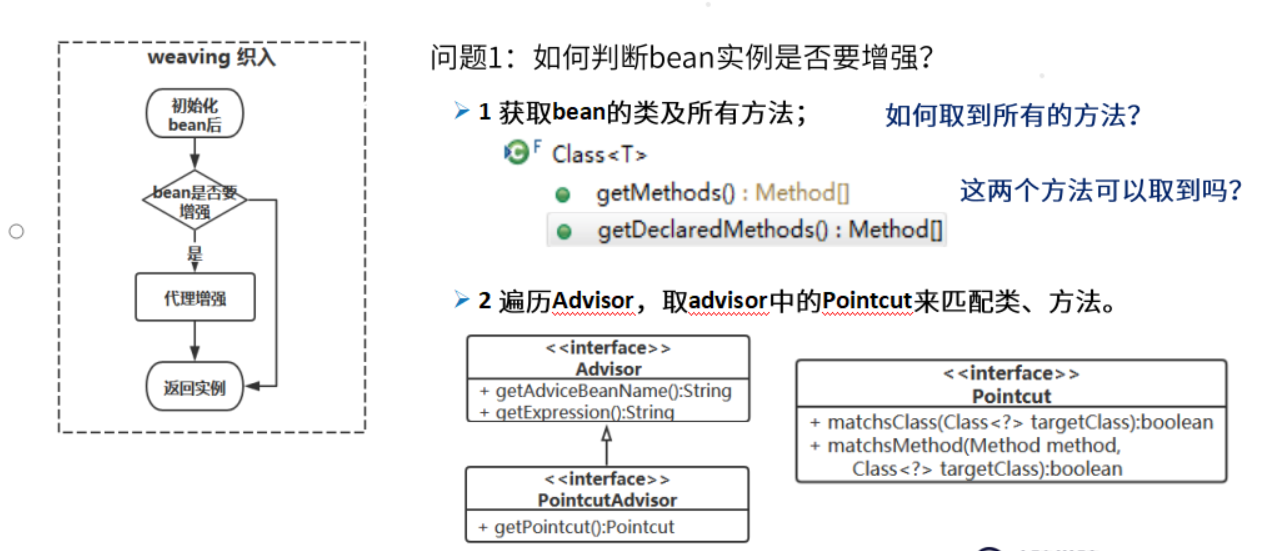
Advisor 扩展

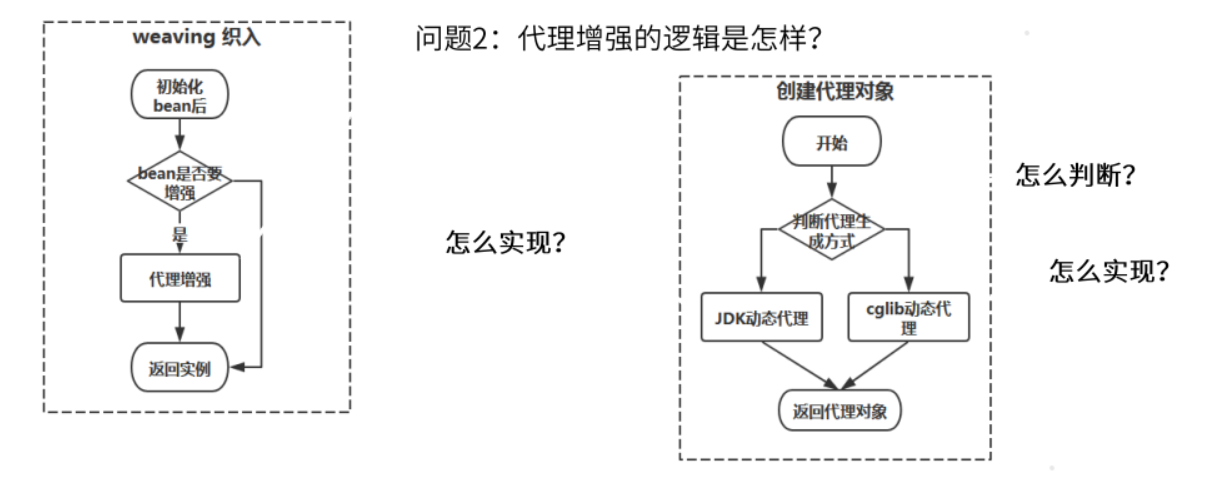


Aspect 设计

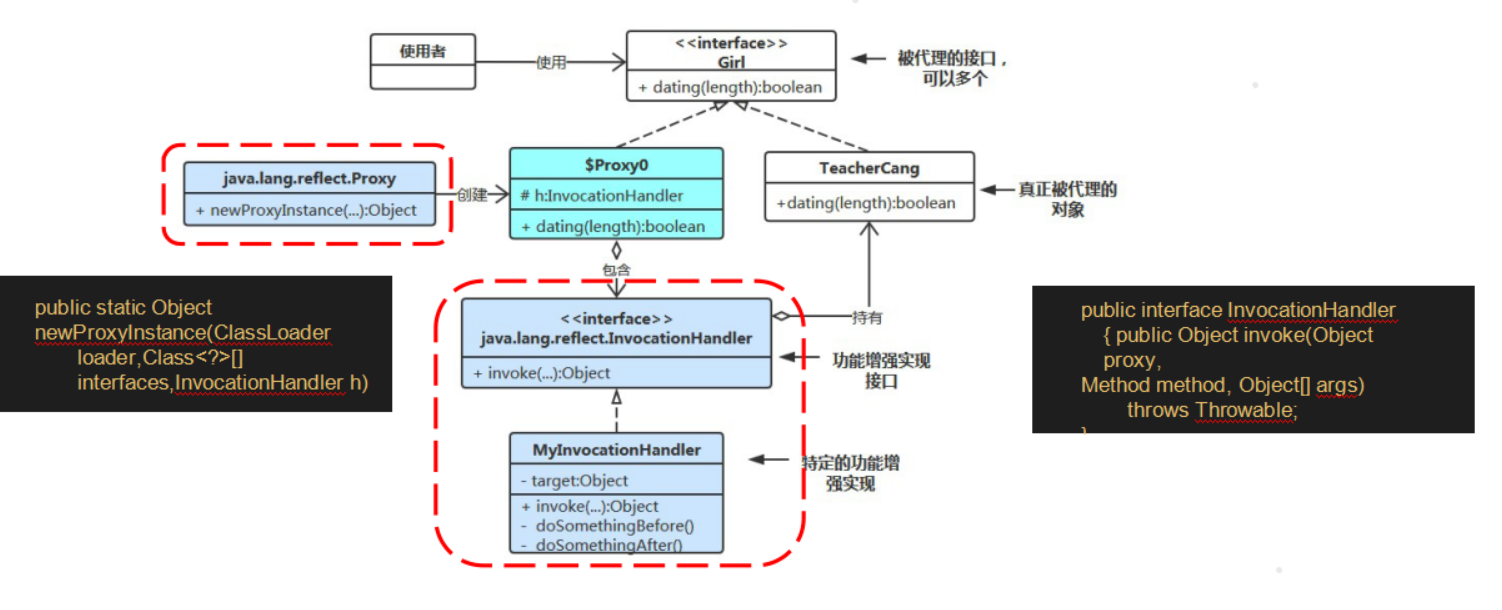
# 织入实现

定义：将用户提供的增强功能加到指定的方法上。

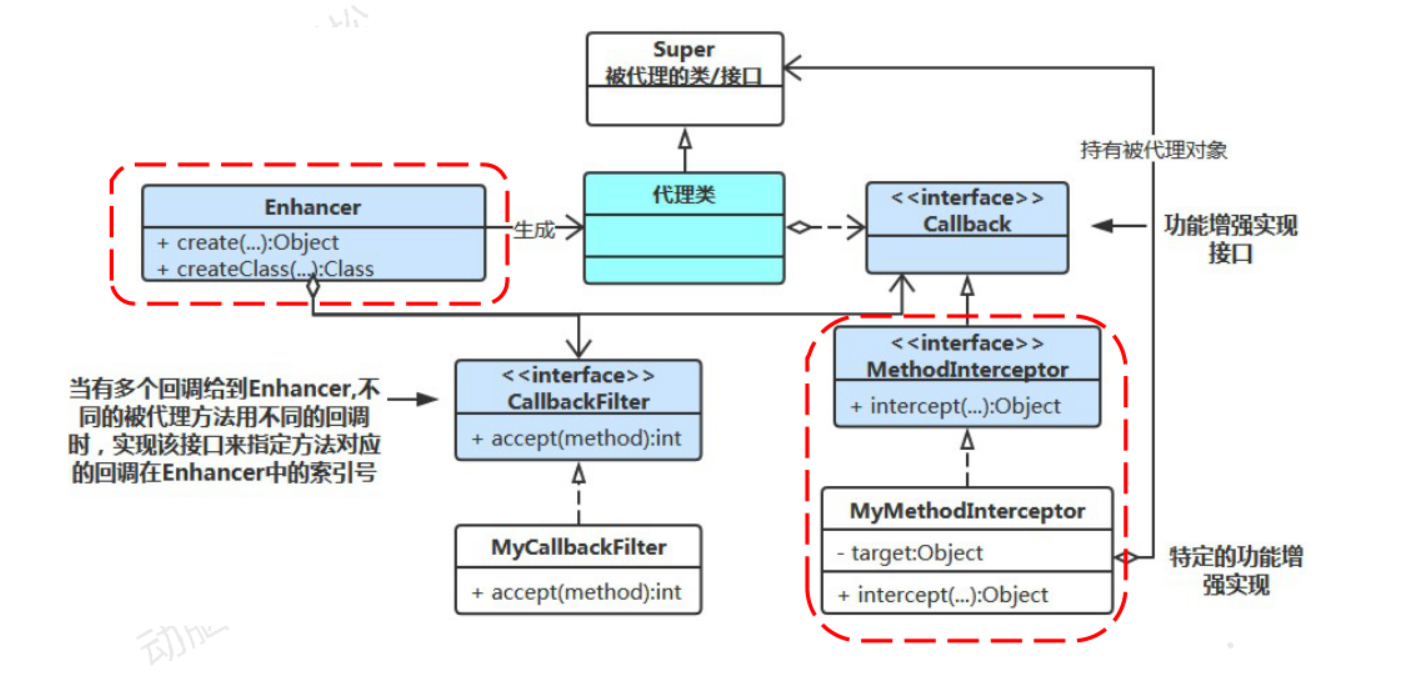




JDK动态代理



CGLB动态代理代理



Weaving织入 创建AopProxy

