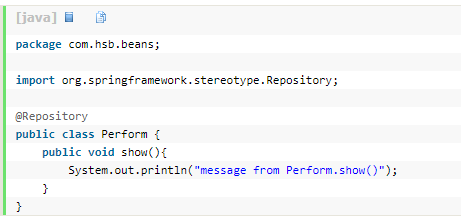
# 概述

在程序运行时，动态的将代码切入到类的指定方法、指定位置上的编程思想就是**面向切面编程。**

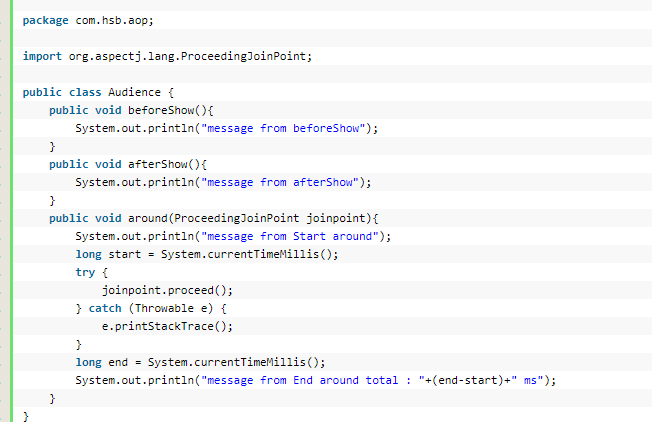
* 一般而言，我们管切入到指定类指定方法的代码片段为切面，而切入的哪些类、哪些方法则叫切入点。有了AOP，我们就可以把几个类共有的代码，抽取到一个切片中，等到需要时在切入对象中去，从而改变其原有的行为。
* AOP就是OOP的补充而已。OOP从横向上区分出一个个类来，而AOP则从纵向上向对象加入特定的代码。有了AOP，OOP变得立体了，如果加上时间维度，AOP使OOP从二维变成三维。
* 从技术上来说，AOP基本上是通过代理机制实现的。
* 经典的Spring AOP是基于动态代理技术的。在实现方式上，最常用的是实现MethodInterceptor接口来提供环绕通知，创建若干代理，然后使用ProxyBeanFactory配置工厂Bean，生成拦截器链，完成拦截。

# 基于配置文件

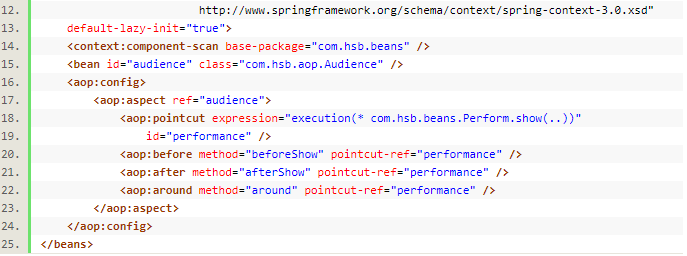
1、编写一个切点



2、编写一个切面



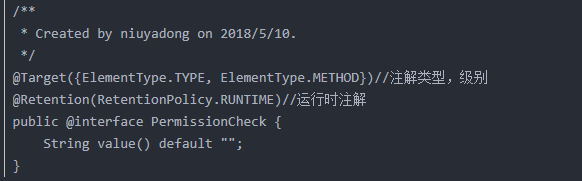
3、配置xml



上面的<bean id="audience" class="com.hsb.aop.Audience" />将切面类生命为一个bean，而<aop:config></aop:config>声明中引用了这个bean.简单解释一下<aop:config>声明中各句的意思。配置声明的时候，一定注意使用英文字符进行配置，不然会报很多稀奇古怪的错误，大致就是说切面无法使用或者其他什么的。

# 基于注解

1、定义一个注解类



2、根据这个注解编写一个注解切面，这个类上需要注解@Aspect

切面类中定义pointcut，以及before、after等切面。

Pintcut可以根据正则表达式表示的方法也可以用上述定义的注解类，然后再到需要切人的方法上添加该注解类。



3、如何在java配置类上定义切面类（添加注解@EnableAspectJAutoProxy //启用 AspectJ 自动代理）或者在xml配置中定义个bean指向这个类。

