

是利用一种称为“横切”的技术，剖解开封装的对象内部，并将那些影响了多个类的公共行为封装到一个可重用模块，并将其名为“Aspect”，即方面。就是将与业务无关，却为业务模块所共同调用的逻辑或责任封装起来，便于减少系统的重复代码，降低模块间的耦合度，并有利于未来的可操作性和可维护性。AOP代表的是一个横向的关系。

Spring中AOP代理由Spring的IOC容器负责生成、管理，其依赖关系也由IOC容器负责管理。因此，AOP代理可以直接使用容器中的其它bean实例作为目标，这种关系可由IOC容器的依赖注入提供。Spring创建代理的规则为：

- 1、默认使用Java动态代理来创建AOP代理，这样就可以为任何接口实例创建代理了
- 2、当需要代理的类不是代理接口的时候，Spring会切换为使用CGLIB代理，也可强制使用CGLIB

代理对象的方法=增强处理+被代理对象的方法。

程序员只需要参与三个部分：

- 1、定义普通业务组件
- 2、定义切入点（pointcut），一个切入点可能横切多个业务组件
- 3、定义增强处理（advice），增强处理就是在AOP框架为普通业务组件织入的处理动作

AOP使用场景

Authentication 权限

Caching 缓存

Context passing 内容传递

Error handling 错误处理

Lazy loading 懒加载

Debugging 调试

logging, tracing, profiling and monitoring 记录跟踪 优化 校准

Performance optimization 性能优化

Persistence 持久化

Resource pooling 资源池

Synchronization 同步

Transactions 事务

相关概念

方面（Aspect）：一个关注点的模块化，可能横切多个对象。事务管理是J2EE应用中一个很好的横切关注点例子。方面用Spring的 Advisor或拦截器实现。

连接点（Joinpoint）：程序执行过程中明确的点，如方法的调用或特定的异常被抛出。

通知（Advice）：在特定的连接点，AOP框架执行的动作。各种类型的通知（“around”、“before”、“throws”等）。Spring是以拦截器做通知模型，维护一个“围绕”连接点的拦截器链。

切入点（Pointcut）：指定一个通知将被引发的一系列连接点的集合。Spring定义了Pointcut接口，用来组合MethodMatcher和ClassFilter：MethodMatcher检查目标类的方法是否被应用此通知，而ClassFilter检查Pointcut是否应用到目标类上。

引入（Introduction）：添加方法或字段到被通知的类。Spring允许引入新的接口到任何被通知的对象。例如，你可以使用一个引入使任何对象实现 IsModified接口，来简化缓存。Spring中要使用Introduction，可有通过DelegatingIntroductionInterceptor来实现通知，通过DefaultIntroductionAdvisor来配置Advice和代理类要实现的接口

目标对象 (Target Object) : 包含连接点的对象。也被称作被通知或被代理对象。POJO

AOP代理 (AOP Proxy) : AOP框架创建的对象, 包含通知。在Spring中, AOP代理可以是JDK动态代理或者CGLIB代理。

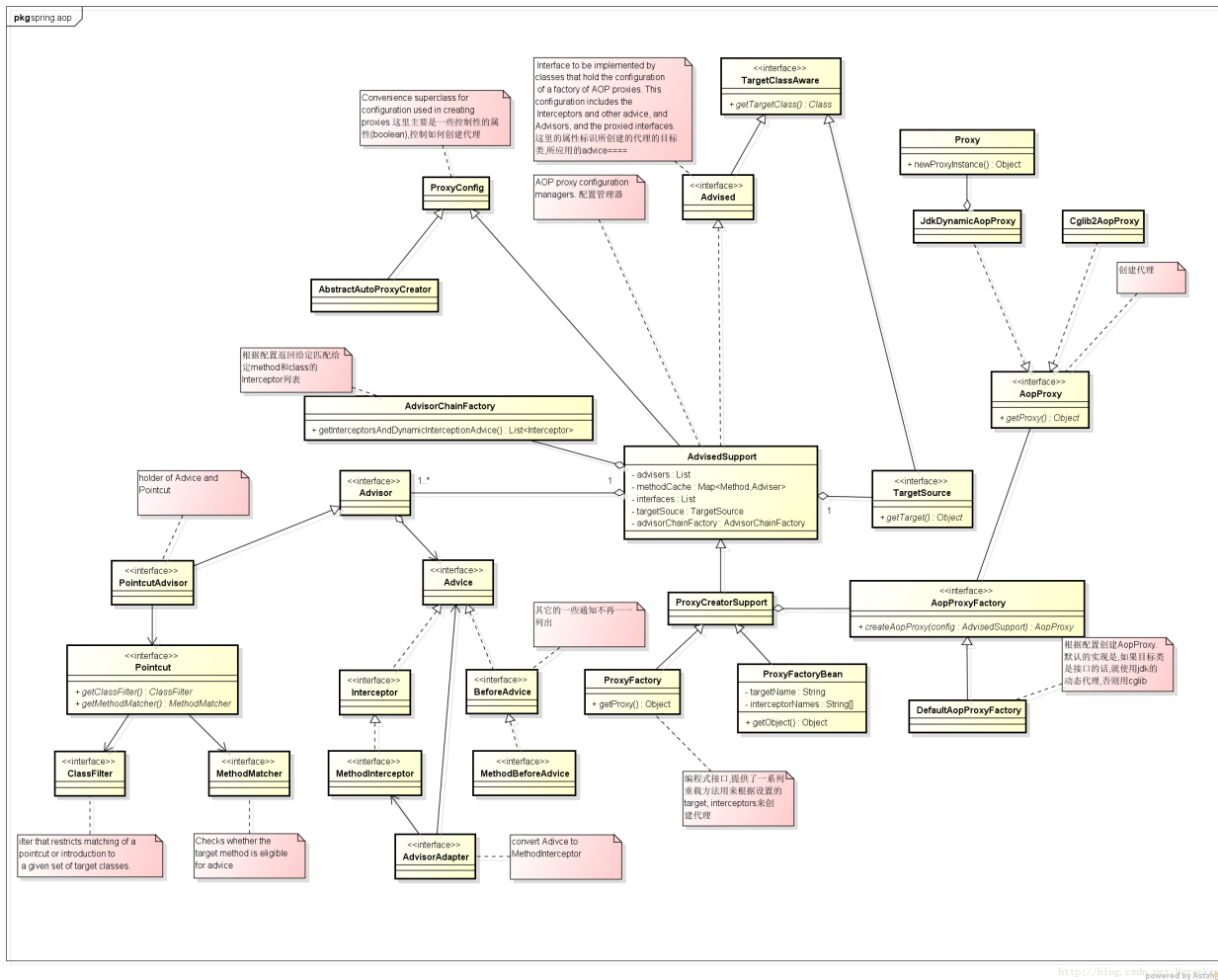
织入 (Weaving) : 组装方面来创建一个被通知对象。这可以在编译时完成 (例如使用AspectJ编译器), 也可以在运行时完成。Spring和其他纯Java AOP框架一样, 在运行时完成织入。

使用Spring AOP

通过配置文件或编程的方式。配置可通过xml文件来进行, 四种方式:

1. 配置ProxyFactoryBean, 显式地设置advisors, advice, target等
2. 配置AutoProxyCreator, 使用定义的bean, 但是从容器中获得的已经是代理对象
3. 通过<aop:config>来配置
4. 通过<aop:aspectj-autoproxy>来配置, 使用AspectJ的注解来标识通知及切入点

也可直接使用ProxyFactory以编程的方式使用Spring AOP, 通过ProxyFactory提供的方法可以设置target对象, advisor等相关配置, 最终通过 getProxy() 方法来获取代理对象。



Spring AOP代理对象的生成

Spring提供了两种方式生成代理对象：JDKProxy和Cglib。具体使用哪种方式生成由AopProxyFactory根据AdvisedSupport对象的配置来决定。默认如果目标类是接口，使用JDK动态代理技术，否则使用Cglib来生成代理。

