**3.1.4  Advisor通知器**

Advisor（通知器）：当我们完成对目标方法的切面增强设计（advice）和关注点的设计（pointcut）以后，需要一个对象把它们结合起来，完成这个作用的就是Advisor。通过Advisor，可以定义应该使用哪个Advice并在哪个Pointcut使用它，也就是说通过Advisor把Advice和Pointcut结合起来了，这个结合为应用使用IoC容器配置AOP应用，或者说即开即用地使用AOP基础设施。在Spring AOP中，我们举一个Advisor的实现（DefaultPointcutAdvisor）作为例子，去了解Advisor的工作原理。在DefaultPointcutAdvisor中有两个属性，分别是Advice和Pointcut，通过这两个属性，可以分别配置Advice和Pointcut，DefaultPointcutAdvisor的实现如代码清单3-7所示。

代码清单3-7  DefaultPointcutAdvisor的实现

1. **public** **class** DefaultPointcutAdvisor **extends**
2. AbstractGenericPointcutAdvisor
3. **implements** Serializable {
5. **private** Pointcut pointcut = Pointcut.TRUE;
6. **public** DefaultPointcutAdvisor() {
7. }
8. **public** DefaultPointcutAdvisor(Advice advice) {
9. **this**(Pointcut.TRUE, advice);
10. }
11. **public** DefaultPointcutAdvisor(Pointcut pointcut,
12. Advice advice) {
13. **this**.pointcut = pointcut;
14. setAdvice(advice);
15. }
16. **public** **void** setPointcut(Pointcut pointcut) {
17. **this**.pointcut = (pointcut != **null** ?
18. pointcut : Pointcut.TRUE);
19. }
20. **public** Pointcut getPointcut() {
21. **return** **this**.pointcut;
22. }
23. **public** String toString() {
24. **return** getClass().getName() + ": pointcut
25. [" + getPointcut() + "]; advice ["
26. + getAdvice() + "]";
27. }
28. }

在DefaultPointcutAdvisor中默认的Pointcut被设置为Pointcut.True，这个Pointcut.True是在Pointcut接口中被定为的：

1. Pointcut TRUE = TruePointcut.INSTANCE;

TruePointcut的INSTANCE是一个单件。在它的实现中，可以看到单件模式的具体应用和它的典型使用方法，比如使用static类变量来持有单件实例，使用private私有构造函数来确保除了在当前单件实现中，单件不会再次被创建和实例化，从而保证它的"单件"特性。在TruePointcut的methodMatcher实现中，使用的是TrueMethodMatcher作为方法匹配器。在这个方法匹配器中，它对任何的方法匹配要求都返回true的结果，也就是说对任何方法名的匹配要求，它都会返回匹配成功的结果，和TruePointcut一样，这个TrueMethodMatcher也是一个单件实现。TruePointcut和TrueMethodMatcher的实现如代码清单3-8和代码清单3-9所示。

代码清单3-8  TruePointcut的实现

1. **class** TruePointcut **implements** Pointcut, Serializable {
2. **public** **static** **final** TruePointcut INSTANCE = **new** TruePointcut();
3. // Enforce Singleton pattern.
4. **private** TruePointcut() {
5. }
6. **public** ClassFilter getClassFilter() {
7. **return** ClassFilter.TRUE;
8. }
9. **public** MethodMatcher getMethodMatcher() {
10. **return** MethodMatcher.TRUE;
11. }

代码清单3-9  TrueMethodMatcher的实现

1. **class** TrueMethodMatcher **implements** MethodMatcher,
2. Serializable {
3. **public** **static** **final** TrueMethodMatcher INSTANCE =
4. **new** TrueMethodMatcher();
5. // Enforce Singleton pattern.
7. **private** TrueMethodMatcher() {
8. }
9. **public** **boolean** isRuntime() {
10. **return** **false**;
11. }
12. **public** **boolean** matches(Method method, Class targetClass) {
13. **return** **true**;
14. }
15. **public** **boolean** matches(Method method, Class targetClass,
16. Object[] args) {
17. // Should never be invoked as isRuntime returns false.
18. **throw** **new** UnsupportedOperationException();
19. }
20. **private** Object readResolve() {
21. **return** INSTANCE;
22. }
24. **public** String toString() {
25. **return** "MethodMatcher.TRUE";
26. }
27. }