**AOP**

애플리케이션 전반에 걸쳐 영향을 주는 기능(횡단관심사)을 모듈화하여 사용하는 것이다. 객체지향에서 공통기능 재사용 방법인 상속이나 위임만으로 불가능한 모듈화가 가능하다.

* **AOP 용어 정리**

**Aspect**

횡단관심사를 모듈화한 클래스. 어드바이스와 포인트컷을 합한 것을 의미한다.

**Advice**

aspect가 해야 할 작업으로 ‘무엇’을 ‘언제’ 할지 정의한다.

|  |  |
| --- | --- |
| 이전(before) | 어드바이스 대상 메소드가 호출되기 전에 어드바이스 기능을 수행한다. |
| 이후(after) | 결과에 상관없이 어드바이스 대상 메소드가 완료된 후에 어드바이스 기능을 수행한다. |
| 반환 이후(after-returning) | 어드바이스 대상 메소드가 성공적으로 완료된 후에 어드바이스 기능을 수행한다. |
| 예외발생 이후(after-throwing) | 어드바이스 대상 메소드가 예외를 던진 후에 어드바이스 기능을 수행한다. |
| 주위(around) | 어드바이스가 어드바이스 대상 메소드를 감싸서 어드바이스 대상 메소드 호출 전과 후에 몇 가지 기능을 제공한다. |

**Join point**

어플리케이션 실행에 어드바이스를 적용할 수 있는 지점을 말한다.

**Pointcut**

aspect가 할 일을 ‘어디서’ 할지 정의한다. 즉, 어드바이스가 위빙(weaving)돼야하는 하나 이상의 조인포인트를 정의한다.

**Introduction**

기존 클래스에 코드 변경 없이 새 메소드나 멤버 변수를 추가하는 기능이다.

**Weaving**

타깃 객체에 애스펙트를 적용해서 새로운 프록시 객체를 생성하는 절차를 말한다. 애스펙트는 타깃 객체의 조인포인트로 위빙된다. 다음은 위빙이 수행될 수 있는 대상 객체의 생애이다.

|  |  |
| --- | --- |
| 컴파일시간(compile time) | 타깃 클래스가 컴파일될 때 애스펙트가 위빙되며, 별도의 컴파일러가 필요하다. AspectJ의 위빙 컴파일러는 이러한 목적으로 사용된다. |
| 클래스로드시간(classload time) | 클래스가 JVM에 로드될 때 애스펙트가 위빙된다. 이렇게 하려면 애플리케이션에서 사용되기 전에 타깃 클래스의 바이트 코드를 인핸스(소스 코드가 아닌 컴파일된 바이트 코드에 직접 메소드나 멤버 변수 등을 추가하는 것)하는 특별한 ClassLoader가 필요하다. AspectJ의 로드시간위빙(LTW) 기능을 사용하면 클래스로드시간에 위빙된다. |
| 실행시간(run time) | 애플리케이션 실행 중에 애스펙트가 위빙된다. 보통 타깃 객체에 호출을 위임하는 구조의 프록시 객체를 위빙 중에 AOP 컨테이너가 동적으로 만들어낸다. 스프링 AOP 애스펙트가 위빙되는 방식. |

* **스프링 AOP**

- 고전적인 스프링 프록시 기반 AOP

- @AspectJ 애너테이션 기반 애스펙트

- Pure-POJO 애스펙트

- AspectJ 애스펙트에 빈 주입

스프링 AOP의 모든 어드바이스는 표준 자바 클래스로 작성되고 포인트컷은 XML 설정파일에 정의한다. 또한 스프링 애스펙트는 실행시간에 만들어진다. 실행시간에 만들어진 프록시 객체는 타깃 객체로 위장하여 어드바이스 대상 메소드의 호출을 가로채고(이때 애스펙트가 실행된다.) 타깃 객체로 호출을 전달한다. 스프링은 다른 AOP 프레임워크와 달리 동적 프록시를 기반으로 AOP를 구현하므로 메소드 조인포인트만 지원한다.

* **포인트컷을 이용한 조인포인트 선택**

AspectJ의 표현식을 사용함.

|  |  |
| --- | --- |
| AspectJ 지정자 | 설명 |
| args() | 인자가 주어진 타입의 인스턴스인 조인포인트를 정의한다. |
| @args() | 전달된 인자의 런타임 타입이 주어진 타입의 애너테이션을 갖는 조인포인트를 정의한다. |
| execution() | 메소드 실행 조인포인트와 일치시키는 데 사용한다. |
| this() | 빈 레퍼런스가 주어진 타입의 인스턴스를 갖는 조인포인트를 정의한다. |
| target() | 대상 객체가 주어진 타입을 갖는 조인포인트를 정의한다. |
| @target0 | 수행 중인 객체의 클래스가 주어진 타입의 애너테이션을 갖는 조인포인트를 정의한다. |
| within() | 특정 타입에 속하는 조인포인트를 정의한다. |
| @within() | 주어진 애너테이션을 갖는 타입 내 조인포인트를 정의한다. |
| @annotation | 조인포인트의 대상 객체가 주어진 애너테이션을 갖는 조인포인트를 정의한다. |
| Spring 지정자 | 설명 |
| bean() | 인자로 빈 ID나 이름을 받고 해당 빈에 대한 포인트컷의 영향을 제한한다. |

execution(\* 클래스패키지.클래스명.메소드명(..) and 지정자(설정))

‘\*’는 해당 메소드가 반환하는 타입이 무엇이든 상관없음을 뜻한다. ‘..’는 인자목록이 무엇이든 간에 해당 메소드를 선택한다는 의미이다. 지정자와 관계연산자로 연결한다. 이때 XML설정을 사용하는 경우에는 &&, ||, ! 대신에 and, or, not을 사용한다.

* **XML을 이용한 애스펙트 선언**

|  |  |
| --- | --- |
| AOP 설정 엘리먼트 | 목적 |
| <aop:advisor> | AOP 어드바이저를 정의한다. |
| <aop:after> | AOP After 어드바이스를 정의한다.(이 어드바이스는 메소드의 예외발생 여부와는 관계없다.) |
| <aop:after-returning> | AOP 반환 이후 어드바이스를 정의한다. |
| <aop:after-throwing> | AOP 예외발생 이후 어드바이스를 정의한다. |
| <aop:around> | AOP 주위 어드바이스를 정의한다. |
| <aop:aspect> | 애스펙트를 정의한다. |
| <aop:aspectj-autoproxy> | @AspectJ를 이용하여 애너테이션 기반 애스펙트를 가능하게 한다. |
| <aop:before> | AOP Before 어드바이스를 정의한다. |
| <aop:config> | 최상위 AOP 엘리먼트. 대부분의 <aop:\*> 엘리먼트는 이 엘리먼트안에 포함돼야 한다. |
| <aop:declare-parents> | 투명하게 구현된 어드바이스 대상 객체에 추가적인 인터페이스를 도입한다. |
| <aop:pointcut> | 포인트컷을 정의한다. |

|  |
| --- |
| <aop:config>  <aop:aspect ref=”클래스명1”>  <aop:pointcut id=”포인트컷ID” expression=”execution(포인트컷\_정의)”/>  <aop:before pointcut-ref=”포인트컷ID” method=”메소드명1”/>  <aop:after-returning pointcut-ref=”포인트컷ID” method=”메소드명2”/>  <aop:after-throwing pointcut-ref=”포인트컷ID” method=”메소드명3”/>  <aop:around pointcut-ref=”포인트컷ID” method=”메소드명3”/>  </aop:aspect>  <aop:aspect ref=”클래스명2”>  <aop:pointcut id=”포인트컷ID” expression=”execution(포인트컷\_정의) and args(인자)”/>  <aop:before pointcut-ref=”포인트컷ID” method=”메소드명4” arg-name=”인자”/>  </aop:aspect>  </aop:config> |

* **애스펙트 애너테이션**

POJO에 @Aspect를 붙여 애스펙트로 정의할 수 있다. 각 메소드에 어드바이스 애너테이션을 붙여 어드바이스로 정의한다. 이를 사용하기 위해서는 스프링 설정파일에 aop 네임스페이스를 포함시켜줘야한다.

|  |
| --- |
| <beans xmlns=<http://www.springframework.org/schema/beans>  xmlns:xsi=<http://www.w3.org/2001/XMLSchema-instance>  xmlns:aop=<http://www.spingframework.org/schema/aop>  xsi:schemaLocation=”http://www.springframework.org/schema/beans  <http://www.springframework.org/schema/beans/spring-beans-3.0.xsd>  <http://www.springframework.org/schema/aop>  http://www.springframework.org/schema/aop/spring-aop-3.0.xsd”> |