

10강. AOP(Aspect Oriented Programming)-II

- @Aspect를 이용한 AOP구현
- AspectJ Pointcut 표현식

Lecturer Kim Myoung-Ho
Nickname 불스
blogstudy@naver.com

10-1. @Aspect를 이용한 AOP구현

작업 순서 (spring_10_1_ex1_springex)

- 1) 의존 설정(pom.xml)
- 2) @Aspect 어노테이션을 이용한 Aspect클래스 제작
- 3) XML파일에 <aop:aspectj-autoproxy /> 설정



10-2. AspectJ Pointcut 표현식

Pointcut을 지정할 때 사용하는 표현식으로 AspectJ 문법을 사용 합니다. (spring_10_2_ex1_springex)

* : 모든
 . : 현재
 .. : 0개 이상

Execution

```
// @Pointcut("execution(public void get*(..))") // public void인 모든 get메소드
// @Pointcut("execution(* com.javalec.ex.*.*())") // com.javalec.ex 패키지에 파라미터가 없는 모든 메소드
// @Pointcut("execution(* com.javalec.ex.*.*())") // com.javalec.ex 패키지 & com.javalec.ex 하위 패키지에 파라미터가 없는 모든 메소드
// @Pointcut("execution(* com.javalec.ex.Worker.*())") // com.javalec.ex.Worker 안의 모든 메소드
```

within

```
// @Pointcut("within(com.javalec.ex.*)") //com.javalec.ex 패키지 안에 있는 모든 메소드
// @Pointcut("within(com.javalec.ex..*)") //com.javalec.ex 패키지 및 하위 패키지 안에 있는 모든 메소드
// @Pointcut("within(com.javalec.ex.Worker)") //com.javalec.ex.Worker 모든 메소드
```

bean

```
// @Pointcut("bean(student)") //student 빈에만 적용
// @Pointcut("bean(*ker)") //~ker로 끝나는 빈에만 적용
```