

Spring Framework

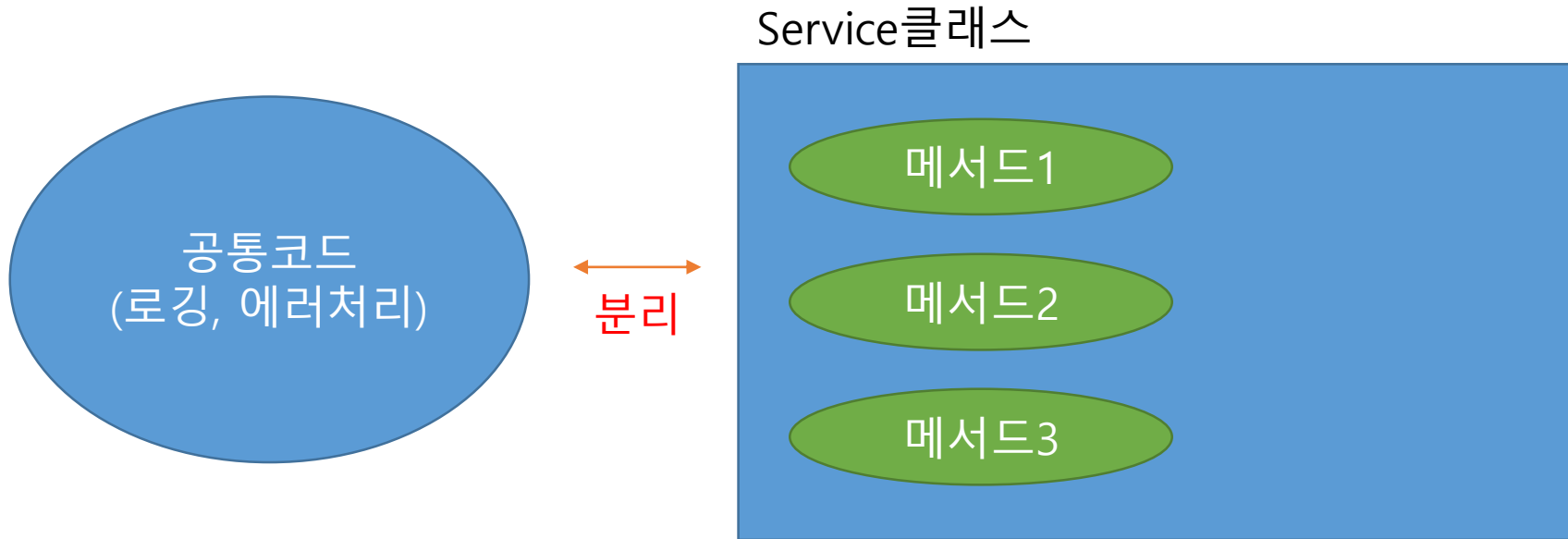
–AOP

1. AOP란
2. AOP의 용어
3. AOP설정

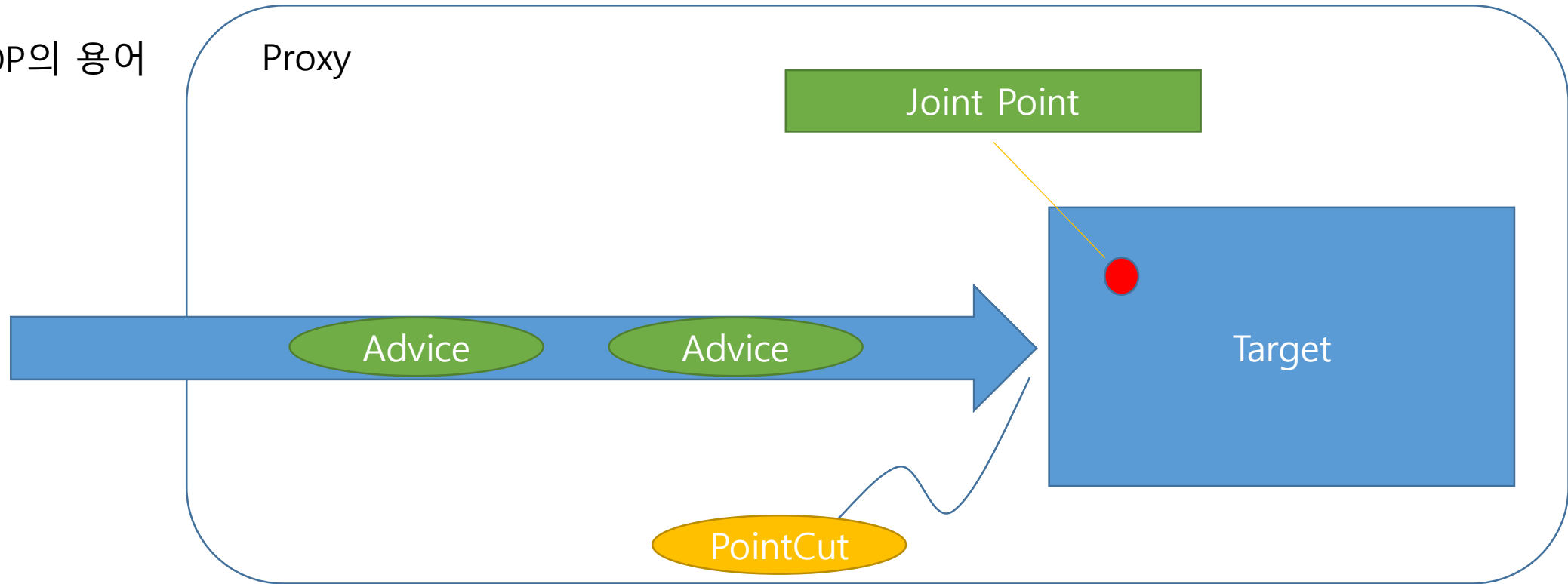
1.AOP란

AOP란 (Aspect-Oriented-Programming)

- 공통 코드, 개별코드(비즈니스 로직) 를 분리해서 작성한다.
- ex) java의 공통 기능을 부모클래스로 정의하고 상속 관계로 사용
- 기존 코드를 수정하지 않고, **외부**에서 원하는 기능에 접근하여 결합



2.AOP의 용어

**Target**

-개발자가 작성한 Service클래스라 보면 된다

Joint Point

-Target의 메서드 (Service클래스 안에 메서드)

Advice

-**공통 코드**(로깅, 에러처리 등)

PointCut

-Advice를 어떤 Joint Point에 결합할 것인지 **설정**(공통 코드를 어떤 메서드에 적용할 것인가)

2.AOP의 용어

Advice

-공통 코드(로깅, 에러처리 등)

Advice 동작위치

구분	설명
@Before	메서드를 호출 전 실행
@After	메서드를 호출 후 실행
@Around	메서드에 결합해서 처리

PointCut

-Advice를 어떤 Joint Point에 결합할 것인지 **설정**
(공통 코드를 어떤 메서드에 적용할 것인가)

-execution(표현식) : 해당 메서드를 기준으로 PointCut 실행

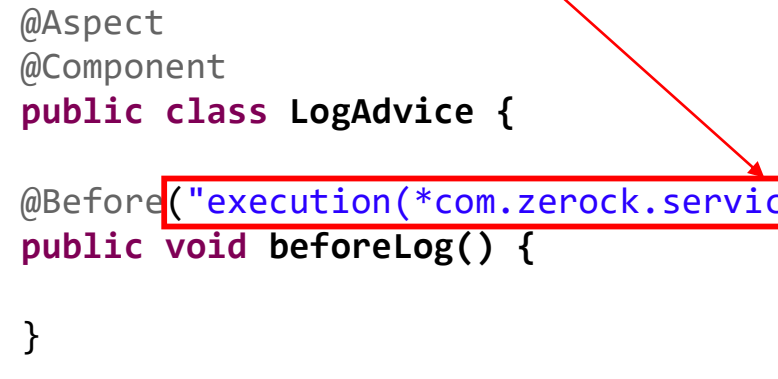
```
@Aspect
@Component
public class LogAdvice {

    @Before("execution(*com.zerock.service.memberServiceImpl*.*(..))")
    public void beforeLog() {

    }

    @After("execution(*com.zerock.service.memberServiceImpl*.*(..))")
    public void afterLog() {

    }
}
```



2.AOP의 용어

- ① **@Around** – 메서드와 결합해서 실행 구조를 바꾸는 강력한 기능
- ② Before, after와 다르게 반드시 반환 유형이 존재 해야함
- ③ ProceedingJoinPoint 타입 은 타겟 메서드의 직접 관여할 수 있음
- ④ proceed() 메서드의 실행이 없다면, 타겟 메서드는 실행되지 않음

```
@Aspect
```

```
@Component
```

```
public class LogAdvice {
```

```
① @Around("execution(* com.zerock.service.memberServiceImpl.*(..))")
```

```
public Object aroundLog(ProceedingJoinPoint jp) {
```

```
②    Object result = null;
```

```
    try {
```

```
        ④    result = jp.proceed();
```

```
    } catch (Throwable e) {
```

```
        e.printStackTrace();
```

```
    }
```

```
    return result;
```

```
}
```

3.AOP설정

-AOP사용 설정

pom.xml 라이브러리 다운

```
<dependency>  
  <groupId>org.aspectj</groupId>  
  <artifactId>aspectjweaver</artifactId>  
  <version>${org.aspectj-version}</version>  
</dependency>
```

servlet.xml 태그 추가

```
<aop:aspectj-autoproxy/>
```

-위 태그 선언으로 aop 어노테이션을 사용할 수 있습니다

Namespace설정

Namespaces

Configure Namespaces

Select XSD namespaces to use in the configuration file

- ☒ aop - http://www.springframework.org/schema/aop
- ☒ beans - http://www.springframework.org/schema/beans
- ☐ c - http://www.springframework.org/schema/c
- ☐ cache - http://www.springframework.org/schema/cache
- ☒ context - http://www.springframework.org/schema/context
- ☐ jee - http://www.springframework.org/schema/jee
- ☐ lang - http://www.springframework.org/schema/lang
- ☒ mvc - http://www.springframework.org/schema/mvc
- ☐ p - http://www.springframework.org/schema/p
- ☐ task - http://www.springframework.org/schema/task
- ☐ util - http://www.springframework.org/schema/util

