

# Spring Framework

## 13. AOP 개요

# CONTENTS

1

AOP의 개요와 용어

2

Spring AOP의 특징 및 구현방식

3

AspectJ와 Spring AOP 라이브러리 설치

# 학습 목표

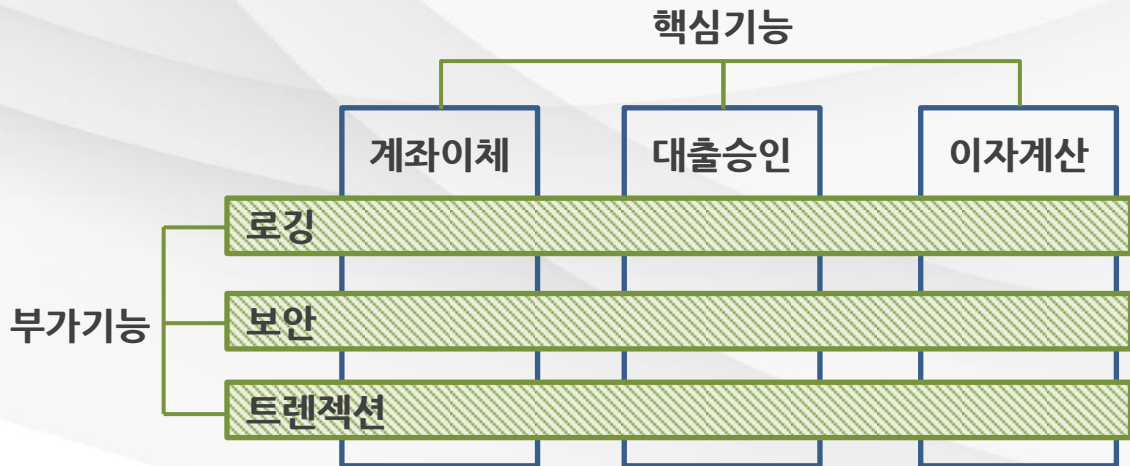
- AOP의 개요와 용어에 대해 이해할 수 있습니다.
- Spring AOP의 특징 및 구현방식에 대해 이해할 수 있습니다.
- AspectJ 와 Spring AOP 설치에 대해 이해할 수 있습니다.

AOP는

## 1. AOP의 개요와 용어

## ■ 핵심기능과 부가기능

- 업무(Biz) 로직을 포함하는 기능을 **핵심 기능(Core Concerns)**
- 핵심기능을 도와주는 부가적인 기능(로깅,보안 등)을 **부가기능(Cross-cutting Concerns)** 이라고 부른다.
- 객체지향의 기본 원칙을 적용하여도 핵심기능에서 부가기능을 분리해서 모듈화하는 것은 매우 어렵다.



## ■ AOP(Aspect Oriented Programming)의 개요

AOP는 애플리케이션에서의 **관심사의 분리(기능의 분리)** 즉, 핵심적인 기능에서 부가적인 기능을 분리한다. 분리한 부가기능을 **애스펙트(Aspect)**라는 **독특한 모듈형태로** 만들어서 설계하고 개발하는 방법

- OOP를 적용하여도 핵심기능에서 부가기능을 **쉽게 분리된 모듈로** 작성하기 **어려운 문제점을** AOP가 해결해 준다고 볼 수 있다.
- AOP는 부가기능을 애스펙트(Aspect)로 정의하여, **핵심기능에서 부가기능을 분리함**으로써 핵심기능을 설계하고 구현할 때 **객체지향적인 가치를** 지킬 수 있도록 도와주는 개념이다.

## ■ 애스펙트(Aspect)

- 애스펙트는 부가기능을 정의한 코드인 **어드바이스(Advice)**와 어드바이스를 어디에 적용하지를 결정하는 **포인트컷(PointCut)**을 합친 개념이다.

Advice + PointCut = Aspect

- AOP 개념을 적용하면 핵심기능 코드 사이에 침투된 부가기능을 독립적인 애스펙트로 구분해 낼 수 있다.
- 구분된 부가기능 애스펙트를 런타임 시에 필요한 위치에 동적으로 참여하게 할 수 있다.

## I AOP 용어

### 타겟(Target)

- ◉ 핵심기능을 담고 있는 모듈로, 타겟은 부가기능을 부여할 대상이 된다.

### 어드바이스(Advice)

- ◉ 어드바이스는 타겟에 제공할 부가기능을 담고 있는 모듈이다.

### 조인 포인트(Join Point)

- ◉ 어드바이스가 적용될 수 있는 위치를 말한다.
- ◉ 즉, 타겟 객체가 구현한 인터페이스의 모든 메서드는 조인 포인트가 된다.



## I AOP 용어

### 포인트 컷(Pointcut)

- ◉ 어드바이스를 적용할 타겟의 메서드를 선별하는 정규표현식이다.
- ◉ 포인트컷 표현식은 execution으로 시작하고, 메서드의 Signature를 비교하는 방법을 주로 이용한다.

### 애스펙트(Aspect)

- ◉ 애스펙트는 AOP의 기본 모듈이다.
- ◉ 애스펙트 = 어드바이스 + 포인트컷
- ◉ 애스펙트는 싱글톤 형태의 객체로 존재한다.

## I AOP 용어

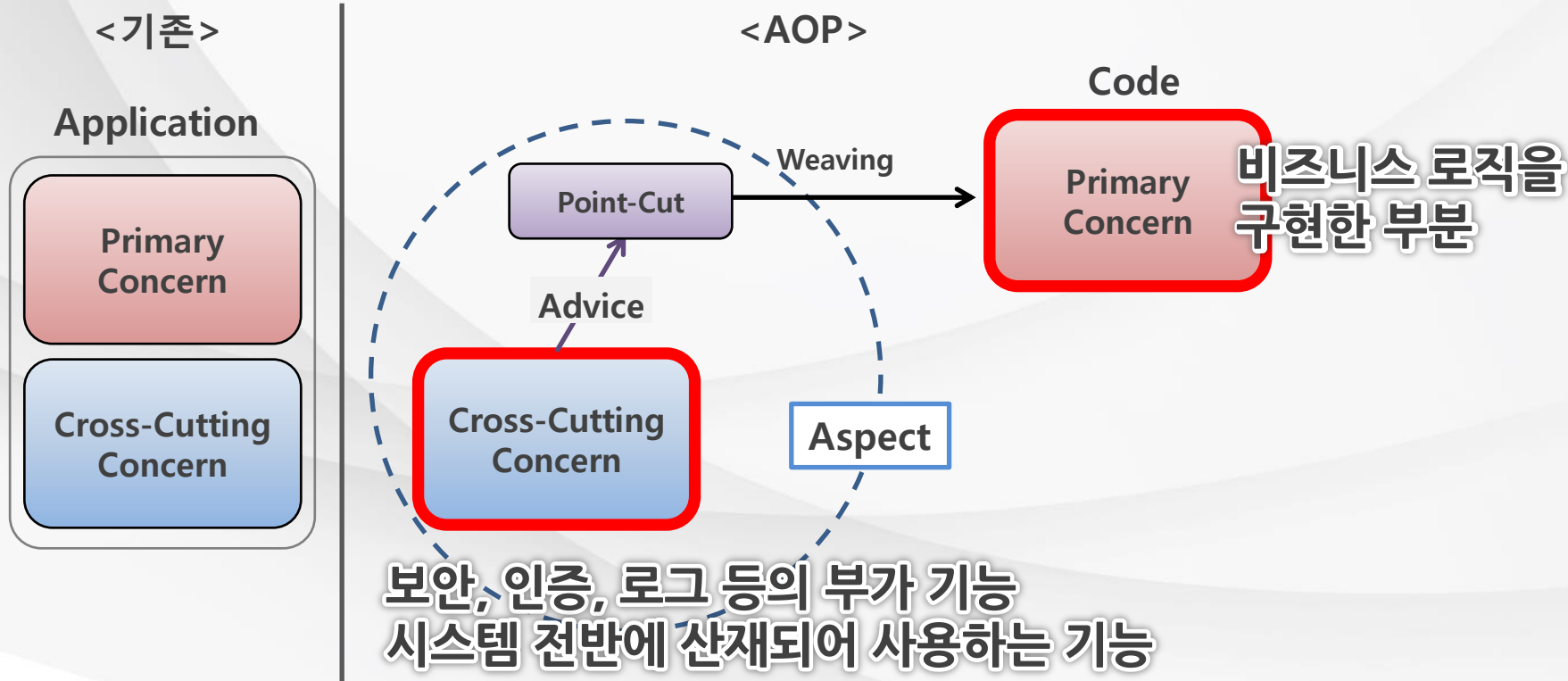
### 어드바이저(Advisor)

- ◉ 어드바이저 = 어드바이스 + 포인트컷
- ◉ 어드바이저는 Spring AOP에서만 사용되는 특별한 용어이다.

### 위빙(Weaving)

- ◉ 위빙은 포인트컷에 의해서 결정된 타겟의 조인 포인트에 부가기능(어드바이스)을 삽입하는 과정을 뜻한다.
- ◉ 위빙은 AOP가 핵심기능(타겟)의 코드에 영향을 주지 않으면서 필요한 부가기능(어드바이스)을 추가할 수 있도록 해주는 핵심적인 처리과정이다.

## AOP 용어정리



A person's hands are shown holding a smartphone, with the screen glowing. The background is dark with out-of-focus, colorful bokeh lights in shades of yellow, orange, and blue. A semi-transparent dark banner is at the bottom, containing a yellow decorative element and the title text.

## 2. Spring AOP의 특징 및 구현방식

### ■ Spring AOP의 특징

#### ❖ (1) Spring은 프록시(Proxy) 기반 AOP를 지원한다.

- ◉ Spring은 타겟(target) 객체에 대한 프록시를 만들어 제공한다.
- ◉ 타겟을 감싸는 프록시는 실행시간(Runtime)에 생성된다.
- ◉ 프록시는 어드바이스를 타겟 객체에 적용하면서 생성되는 객체이다.



### ■ Spring AOP의 특징

#### ❖ (2) 프록시(Proxy)가 호출을 가로챈다(Intercept).

- 프록시는 타겟 객체에 대한 호출을 가로챈 다음 어드바이스의 부가기능 로직을 수행하고 난 후에 타겟의 핵심기능 로직을 호출한다. (전처리 어드바이스)
- 또는 타겟의 핵심기능 로직 메서드를 호출한 후에 부가기능(어드바이스)을 수행하는 경우도 있다.(후처리 어드바이스)



### ■ Spring AOP의 특징

#### ❖ (3) Spring AOP는 메서드 조인 포인트만 지원한다.

- ◉ Spring은 동적 프록시를 기반으로 AOP를 구현하므로 메서드 조인 포인트만 지원한다. 즉, 핵심기능(타겟)의 메서드가 호출되는 런타임 시점에만 부가기능(어드바이스)을 적용할 수 있다.
- ◉ 반면에 AspectJ 같은 고급 AOP 프레임워크를 사용하면 객체의 생성, 필드값의 조회와 조작, static 메서드 호출 및 초기화 등의 다양한 작업에 부가기능을 적용할 수 있다.

### ■ Spring AOP의 구현 방식

#### XML 기반의 POJO 클래스를 이용한 AOP 구현

- ◉ 부가기능을 제공하는 Advice 클래스를 작성한다.
- ◉ XML 설정 파일에 <aop:config>를 이용해서 애스펙트를 설정한다.  
(즉, 어드바이스와 포인트컷을 설정함)

#### @Aspect 어노테이션을 이용한 AOP 구현

- ◉ @Aspect 어노테이션을 이용해서 부가기능을 제공하는 Aspect 클래스를 작성한다. 이때 Aspect 클래스는 어드바이스를 구현하는 메서드와 포인트컷을 포함한다.
- ◉ XML 설정 파일에 <aop:aspectj-autoproxy />를 설정한다.



A person's hands are shown holding a black smartphone with a white screen. The background is dark with out-of-focus, colorful bokeh lights in shades of yellow, orange, and blue. A semi-transparent dark banner with a yellow decorative element on the left is positioned at the bottom of the image.

### 3. AspectJ와 Spring AOP 라이브러리 설치

#### ■ AspectJ Runtime 라이브러리 설치

<http://mvnrepository.com>에 접근한다.

`aspectj runtime`로 검색한다.

`aspectj runtime 1.7.4` 버전을 `pom.xml`에 추가한다.

```
<!-- https://mvnrepository.com/artifact/org.aspectj/aspectjrt -->
<dependency>
  <groupId>org.aspectj</groupId>
  <artifactId>aspectjrt</artifactId>
  <version>1.7.4</version>
</dependency>
```

### 3. AspectJ와 Spring AOP 라이브러리 설치

#### ■ AspectJ Weaver 라이브러리 설치

<http://mvnrepository.com>에 접근한다.

[aspectj weaver](#)로 검색한다.

[aspectj weaver 1.7.4](#) 버전을 [pom.xml](#)에 추가한다.

```
<!-- https://mvnrepository.com/artifact/org.aspectj/aspectjweaver -->
<dependency>
  <groupId>org.aspectj</groupId>
  <artifactId>aspectjweaver</artifactId>
  <version>1.7.4</version>
</dependency>
```

### 3. AspectJ와 Spring AOP 라이브러리 설치

#### ■ Spring AOP 라이브러리 설치

<http://mvnrepository.com>에 접근한다.

`spring aop` 로 검색한다.

`spring aop 3.2.17` 버전을 `pom.xml`에 추가한다.

```
<!-- http://mvnrepository.com/artifact/org.springframework/spring-aop -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-aop</artifactId>
  <version>3.2.17.RELEASE</version>
</dependency>
```

## ■ AspectJ Runtime API 문서



aspectj runtime api doc



전체

동영상

뉴스

이미지

지도

더보기 ▼

검색 도구

### Overview (AspectJ(tm) runtime API) - Eclipse

[eclipse.org/aspectj/doc/next/runtime-api/](https://eclipse.org/aspectj/doc/next/runtime-api/) ▼ 이 페이지 번역하기

org.aspectj.lang, Provides several interfaces for obtaining reflective information about a join point, as well as several exceptions that can be thrown by AspectJ ...

#### JoinPoint

public interface JoinPoint. Provides reflective access to ...

#### JoinPoint.StaticPart

Interface JoinPoint.StaticPart. All Known Subinterfaces: JoinPoint.

#### ProceedingJoinPoint

ProceedingJoinPoint exposes the

#### frames

The Overview page is the front page of this API document and ...

#### Signature

public interface Signature. Represents the signature at a ...

#### MethodSignature

org.aspectj.lang.reflect. Interface

## ■ AspectJ Runtime API 문서

URL

<http://www.eclipse.org/aspectj/doc/next/runtime-api/>



www.eclipse.org/aspectj/doc/next/runtime-api/

**Overview** Package Class Tree Deprecated Index

PREV NEXT FRAMES NO FRAMES

Packages	
<a href="#">org.aspectj.lang</a>	Provides several interfaces for defining join points, as well as several interfaces for defining advice code.
<a href="#">org.aspectj.lang.reflect</a>	Contains interfaces that define the structure of the reflection API about each possible join point.
<a href="#">org.aspectj.runtime.reflect</a>	

Browser의  
즐거찾기에 추가함



학습정리

지금까지 **[AOP 개요]**에 대해서 살펴보았습니다.

## AOP의 개요와 용어

- ◉ 핵심기능(타겟)과 부가기능(어드바이스)의 분리, 포인트컷
- ◉ 조인포인트, 위빙, 애스펙트, 어드바이저

## Spring AOP의 특징 및 구현방식

- ◉ Spring은 프록시 기반의 AOP, 메서드 조인 포인트만 지원
- ◉ AOP 구현방식은 XML 기반, 어노테이션 기반

## AspectJ와 Spring AOP 라이브러리 설치

- ◉ Aspectj runtime, Aspectj weaver 설치
- ◉ Spring AOP 설치, Aspectj runtime api 문서 찾기