박 경 태
comsi.java@gmail.com

고급 자바 프로그래밍

: STS를 이용한 Spring 프로그래밍

강의 내용

순서	내 용
1	• Spring IoC를 이용한 비즈니스 컴포넌트 만들기
2	• Spring AOP(Aspect Oriented Programming)를 이용한 공통 서비스 만들기 • Spring DAO(Data Access Object)를 이용한 데이터베이스 연동 및 트랜잭션 처리
3	• Spring MVC를 이용한 MVC 아키텍쳐 적용하기
4	• Spring MVC의 부가 기능 사용하기(파일 업로드, 다국어, 예외 처리 등)
5	• Spring과 MyBatis 연동하기 • Spring과 JPA 연동하기

AOP(Aspect Oriented Programming)개요

- 기존 OOP의 한계점
  - 어플리케이션을 개발할 때, 메소드 내에 **핵심적인 비즈니스 로직(핵심관심사)**을 담당하는 소스코드와 Logging, Exception 처럼 **비즈니스 로직과 관계없는 코드(횡단관심사)**를 같이 구현할 수 밖에 없다.
  - Logging, Exception을 처리하기 위한 소스코드는 어플리케이션 소스 코드의 전체 영역에 걸쳐 비슷한 패턴으로 중복해서 구현하는 것이 일반적이다.
  - Logging, Exception과 같이 어플리케이션 전체 영역에 걸쳐서 중복된 소스코드가 존재하고 있기 때문에 Logging, Exception 정책이 바뀔 경우 개발자가 예제 소스를 일일이 찾아서 수정해야 하는 번거로움을 감수해야 한다.

- OOP의 한계점을 극복하기 위한 대안으로 등장한 것이 AOP(Aspect Oriented Programming)이다.
  - OOP는 Logging, Exception과 같은 Infrastructure를 처리하기 위하여 가지고 있었던 문제점까지는 해결하지 못함.
  - AOP를 OOP기반 하에서의 OOP문제점을 보완하는 역할을 함.

OOP(Object-Oriented Programming)

핵심 기능과 로그가 혼합

코드 중복 상승

유지보수성 저하

생산성 저하

재사용성 저하

주 기능 구현

로그

AOP(Aspect Oriented Programming)

- 어플리케이션을 다양한 관점으로 분해하여 객체지향에서 추구하는 모듈화를 더 잘 지원하도록 하는 기법

- 가장 기초가 되는 개념은 '관심의 분리(Separation of Concerns)'이고, **핵심 관심사**에 대한 관점과 **횡단 관심사**에 대한 관점으로 나뉜다.
  - 핵심 관심사(core concerns): 각 모듈에서 수행해야 하는 기본적이고 대표적인 업무 기능
  - 횡단 관심사(cross-cutting concerns): 여러 개의 모듈에 걸치는 시스템의 부가적인 업무로 로깅, 예외처리, 성능 관리 등의 기능

- 현재 주류인 객체 지향을 **대신하는 것이 아니라 보완**하는 새로운 패러다임

AOP(Aspect Oriented Programming)

- OOP의 한계?

객체 지향 프로그램은 추상화나 캡슐화를 통해 관심사에 대해 모듈화를 수행하고 사용하므로 횡단 관심사에 해당하는 로깅, 예외 모듈 등을 핵심 관심사에 직접 포함하게 됨.

- AOP 사용

로깅, 예외 등의 횡단 관심사를 직접 호출하는 것이 아니라 Aspect를 활용해 컴파일 시나 런타임 시에 weaving 과정을 거쳐 하나의 시스템으로 조립하는 것이 가능하게 됨.

Weaving? 핵심관심사와 횡단관심사를 컴파일이나 런타임 시에 합성하여 주는 프로세스로 핵심관심사에서 횡단관심사를 자동으로 삽입하여 호출 실행 시켜줌.

스프링(Spring)은…

객체지향
(OOP)

+

의존관계주입
(DI)

+

관점지향프로그래밍
(AOP)

객체 지향보다 독립성을 높여 재사용성과 보전성 향상

AOP 구현

- 비즈니스 메소드가 실행되기 직전에 공통으로 처리할 로직을 LogAdvice클래스에 printLog()메소드로 구현
- LogAdvice클래스의 printLog() 메소드를 BoardService 컴포넌트에서 사용할 수 있도록 BoardServiceImpl 클래스를 수정
  - 각 매소드에서 비즈니스 로직을 수행하기 전에 printLog() 메소드를 호출

LogAdvice 클래스

- 비즈니스 로직 수행전에 로그 출력하는 클래스

BoardDAO.java | UserDAO.java | JDBCUtil.java | LogAdvice.java

```
1  package kr.ac.inje.comsi.common;
2
3  public class LogAdvice {
4      public void printLog(){
5          System.out.println("[공통 로그] 비즈니스 로직 수행 전 동작");
6      }
7  }
8
```

BoardServiceImpl.java 클래스 수정

BoardDAO.java | UserDAO.java | JDBCUtil.java | LogAdvice.java | BoardServiceImpl.java | BoardServiceClient.java

```java
package kr.ac.inje.comsi.board.impl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import kr.ac.inje.comsi.board.BoardService;
import kr.ac.inje.comsi.board.BoardVO;
import kr.ac.inje.comsi.common.LogAdvice;

@Service("boardService")
public class BoardServiceImpl implements BoardService {

    @Autowired
    private BoardDAO boardDAO;
    private LogAdvice log;

    public BoardServiceImpl() {
        log = new LogAdvice();
    }

    @Override
    public void insertBoard(BoardVO vo) {
        log.printLog();
        boardDAO.insertBoard(vo);
    }

    @Override
    public void updateBoard(BoardVO vo) {
        log.printLog();
        boardDAO.updateBoard(vo);
    }

}
```

추가

추가

추가

추가

```
36      @Override
37      public void deleteBoard(BoardVO vo) {
38          log.printLog();
39          boardDAO.deleteBoard(vo);
40      }
41
42      @Override
43      public void getBoard(BoardVO vo) {
44          log.printLog();
45          boardDAO.getBoard(vo);
46      }
47
48      @Override
49      public List<BoardVO> getBoardList(BoardVO vo) {
50          log.printLog();
51          return boardDAO.getBoardList(vo);
52      }
53
54  }
55
```

추가

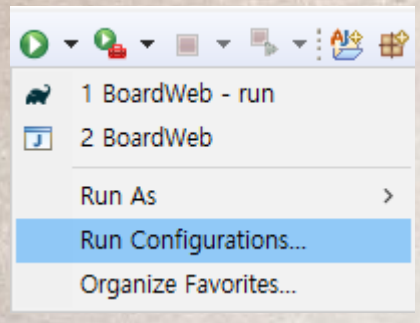
추가

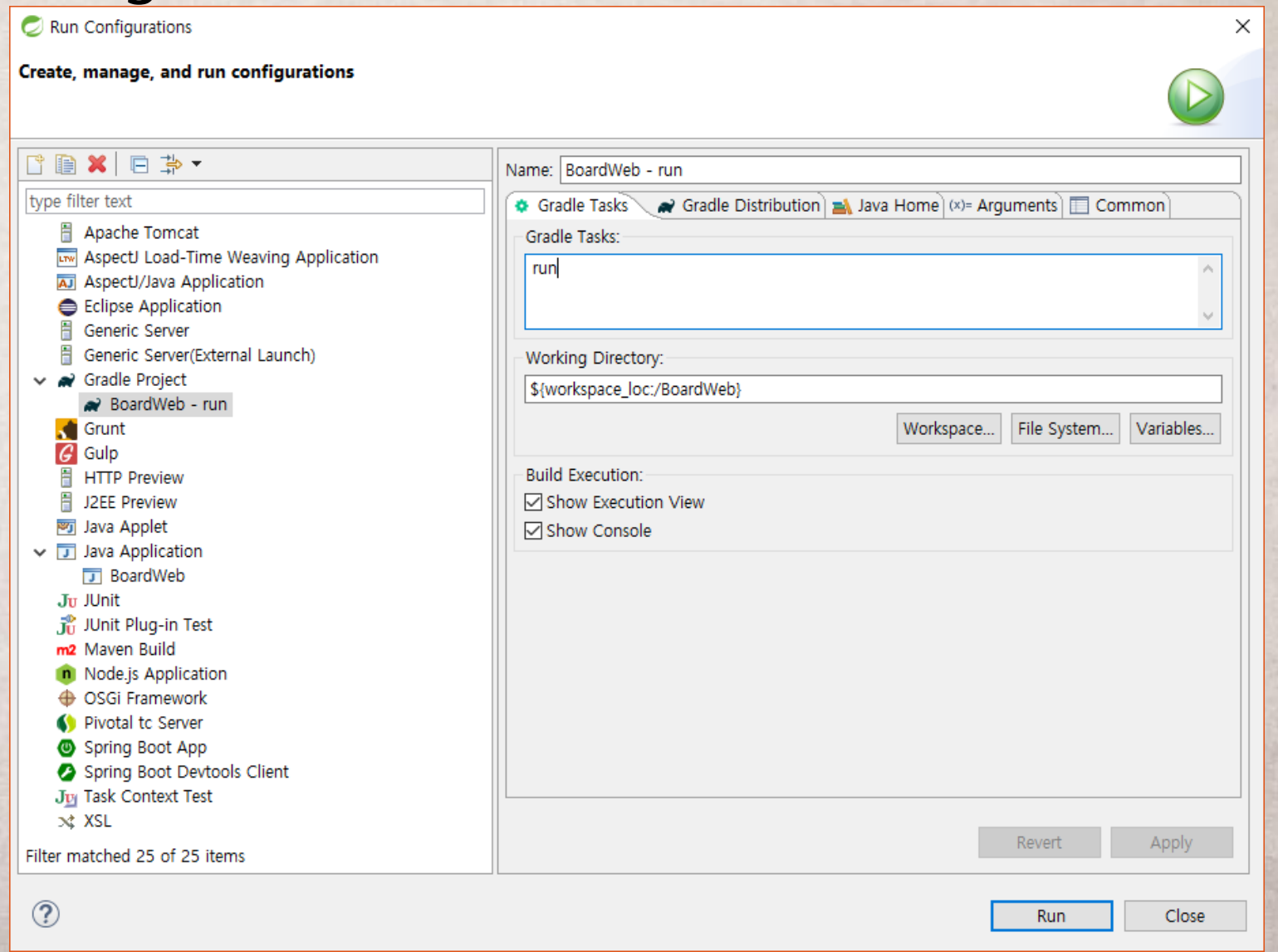
추가

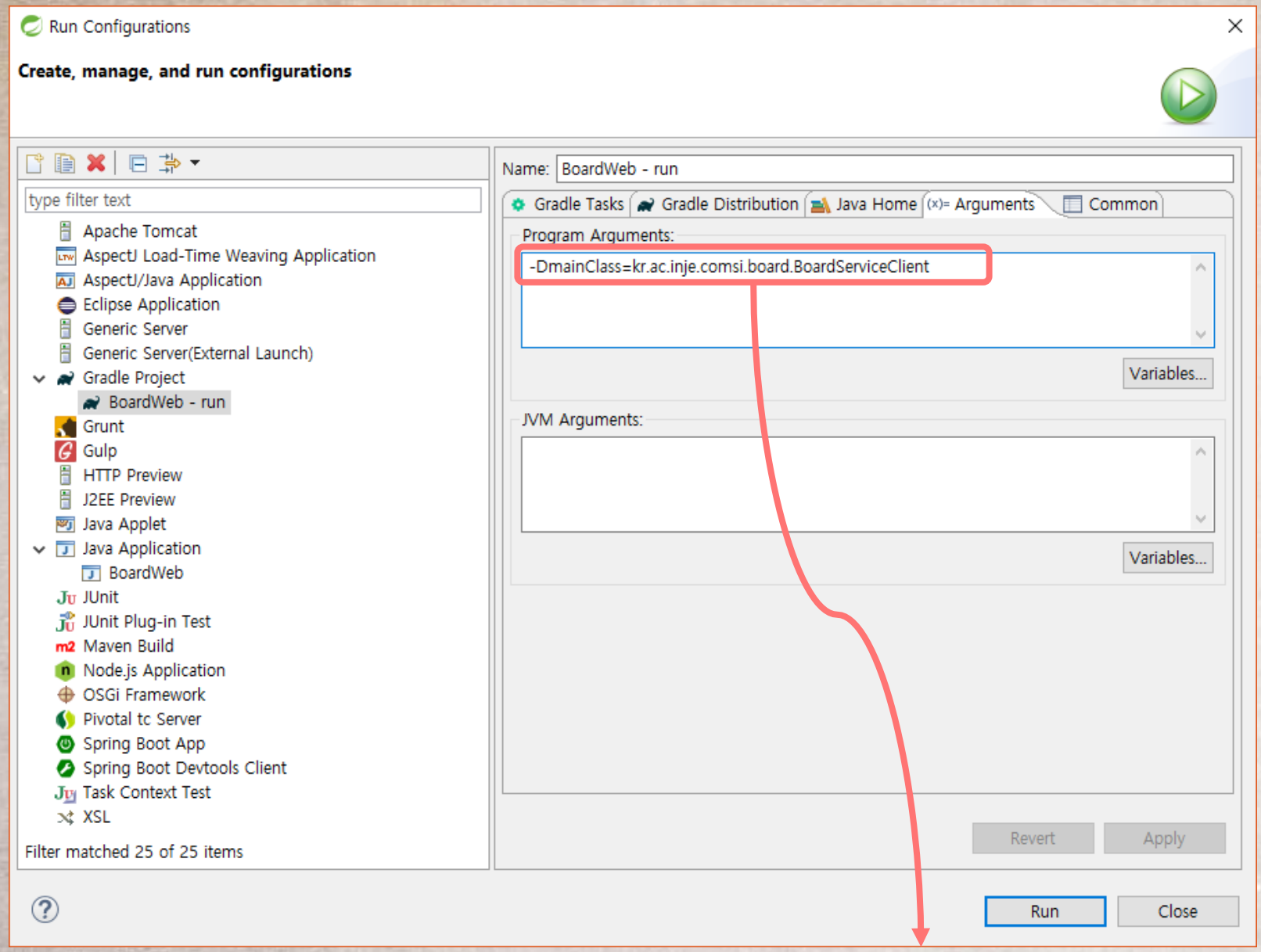
- LogAdvice 객체가 소스에 강하게 결합되어 있음
  - LogAdvice를 다른 클래스로 변경하거나 printLog() 매소드의 시그니처가 변경되는 상황에 유연하게 대처하기가 어렵다.

실행

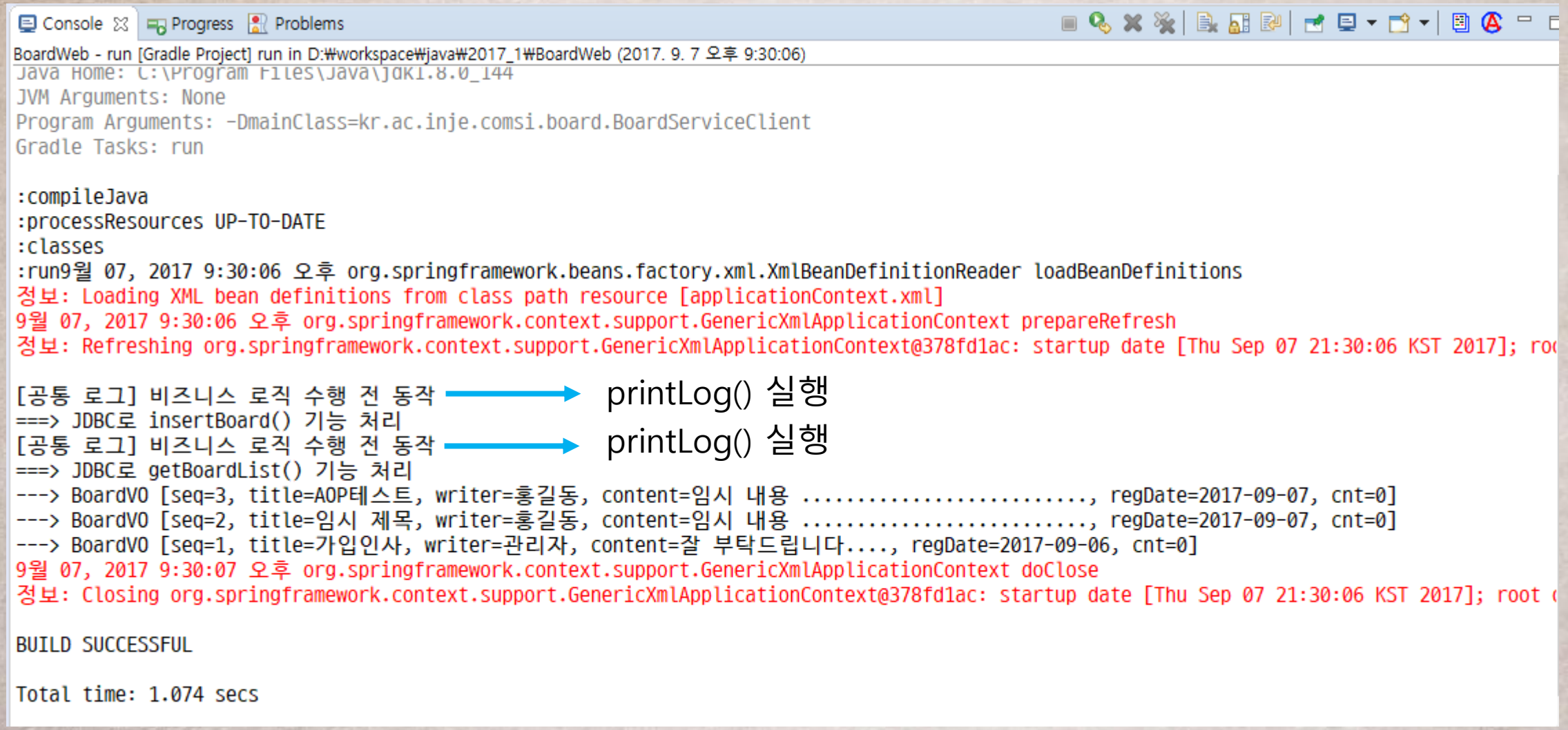
- Run 아이콘 〉 Run Configurations… 실행







-DmainClass=kr.ac.inje.comsi.board.BoardServiceClient


Console
Progress
Problems
BoardWeb - run [Gradle Project] run in D:₩workspace₩java₩2017_1₩BoardWeb (2017. 9. 7 오후 9:30:06)
Java Home: C:\Program Files\Java\jdk1.8.0_144
JVM Arguments: None
Program Arguments: -DmainClass=kr.ac.inje.comsi.board.BoardServiceClient
Gradle Tasks: run
:compileJava
:processResources UP-TO-DATE
:classes
:run9월 07, 2017 9:30:06 오후 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
정보: Loading XML bean definitions from class path resource [applicationContext.xml]
9월 07, 2017 9:30:06 오후 org.springframework.context.support.GenericXmlApplicationContext prepareRefresh
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: startup date [Thu Sep 07 21:30:06 KST 2017]; ro
[공통 로그] 비즈니스 로직 수행 전 동작
printLog() 실행
===> JDBC로 insertBoard() 기능 처리
[공통 로그] 비즈니스 로직 수행 전 동작
printLog() 실행
===> JDBC로 getBoardList() 기능 처리
---> BoardVO [seq=3, title=AOP테스트, writer=홍길동, content=임시 내용 ....., regDate=2017-09-07, cnt=0]
---> BoardVO [seq=2, title=임시 제목, writer=홍길동, content=임시 내용 ....., regDate=2017-09-07, cnt=0]
---> BoardVO [seq=1, title=가입인사, writer=관리자, content=잘 부탁드립니다...., regDate=2017-09-06, cnt=0]
9월 07, 2017 9:30:07 오후 org.springframework.context.support.GenericXmlApplicationContext doClose
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@378fd1ac: startup date [Thu Sep 07 21:30:06 KST 2017]; root
BUILD SUCCESSFUL
Total time: 1.074 secs

만약?

- LogAdvice 클래스가 Log4jAdvice로 변경된다면?
  - printLog() → printLogging()으로 변경된다면

→LogAdvice클래스를 사용하는 BoardServiceImpl 클래스를 수정

- 생성자 수정
- 로그 출력 메소드 수정

AOP 시작하기

- 비즈니스 클래스 수정(BoardServiceImple.java 클래스를 원상태로 복원)
- AOP 라이브러리 추가(gradle.build)
- 스프링 설정파일(applicationContext.xml) 파일에서 사용할 aop 네임 스페이스 추가

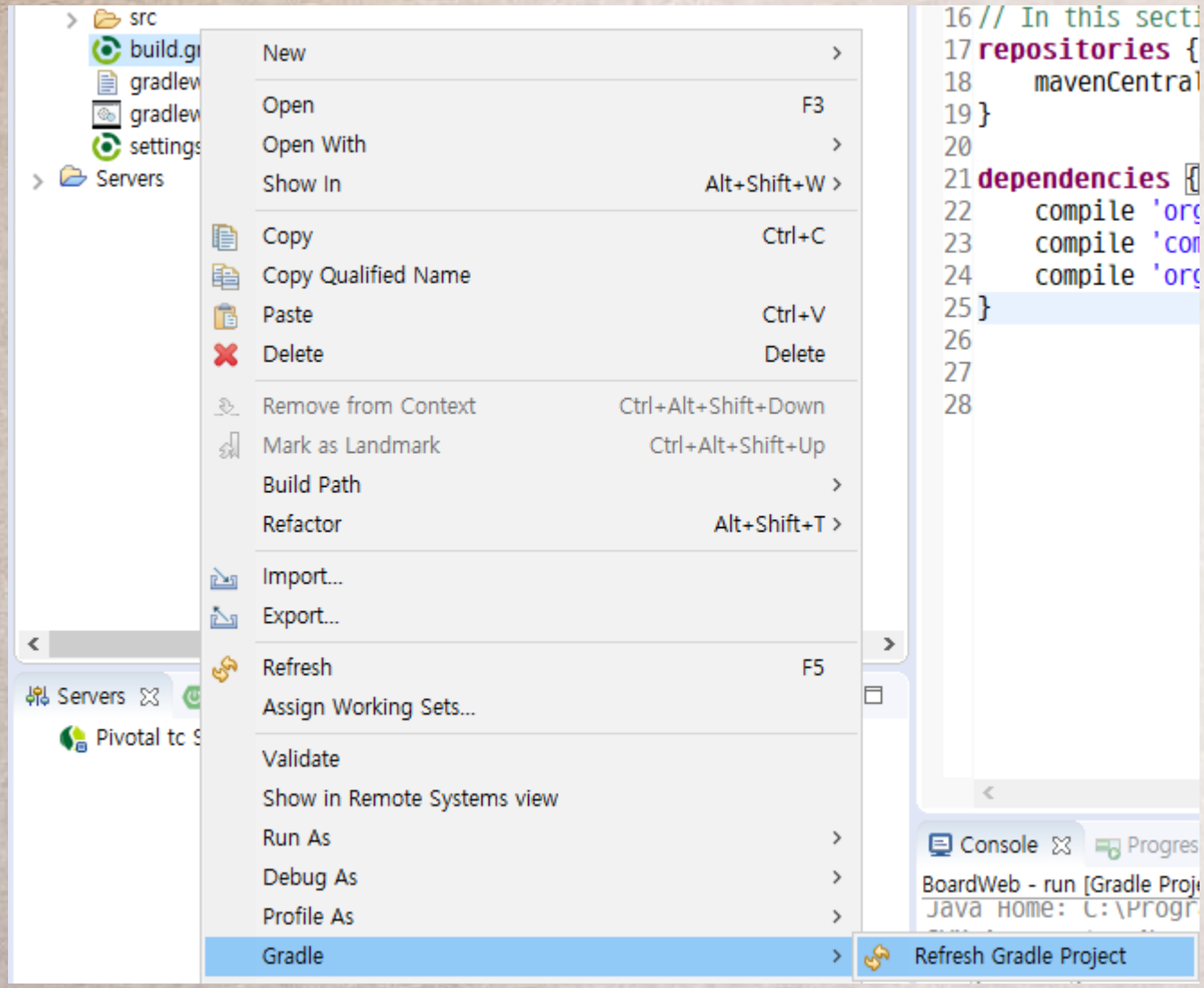
build.gradle 수정 - AOP 라이브러리(aspectjweaver)

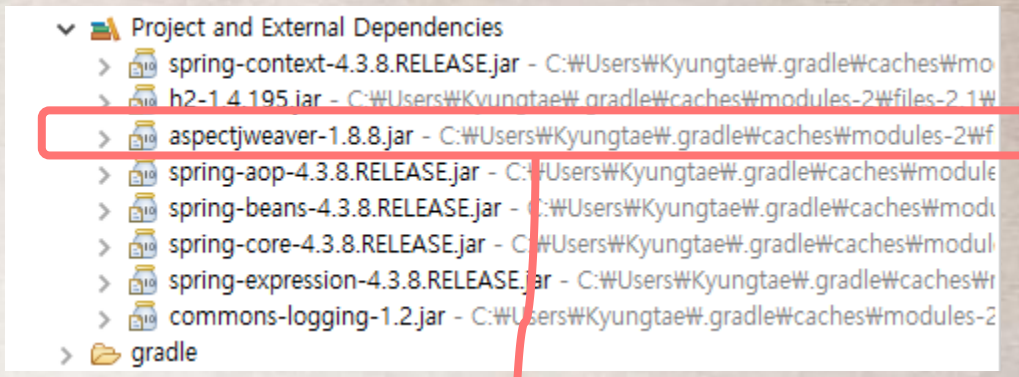
BoardServiceImpl.java | BoardServiceClient.java | build.gradle

```
1 /*
2  * This build file was generated by the Gradle 'init' task.
3  *
4  * This generated file contains a sample Java project to get you started.
5  * For more details take a look at the Java Quickstart chapter in the Gradle
6  * user guide available at https://docs.gradle.org/3.3/userguide/tutorial_java_projects.html
7  */
8
9 // Apply the java plugin to add support for Java
10 apply plugin: 'application'
11
12 mainClassName = System.getProperty("mainClass");
13
14 compileJava.options.encoding = 'UTF-8'
15
16 // In this section you declare where to find the dependencies of your project
17 repositories {
18     mavenCentral()
19 }
20
21 dependencies {
22     compile 'org.springframework:spring-context:4.3.8.RELEASE'
23     compile 'com.h2database:h2:1.4.195'
24     compile 'org.aspectj:aspectjweaver:1.8.8'
25 }
26
27
```

추가

Refresh gradle projec를 통한 라이브러리 추가

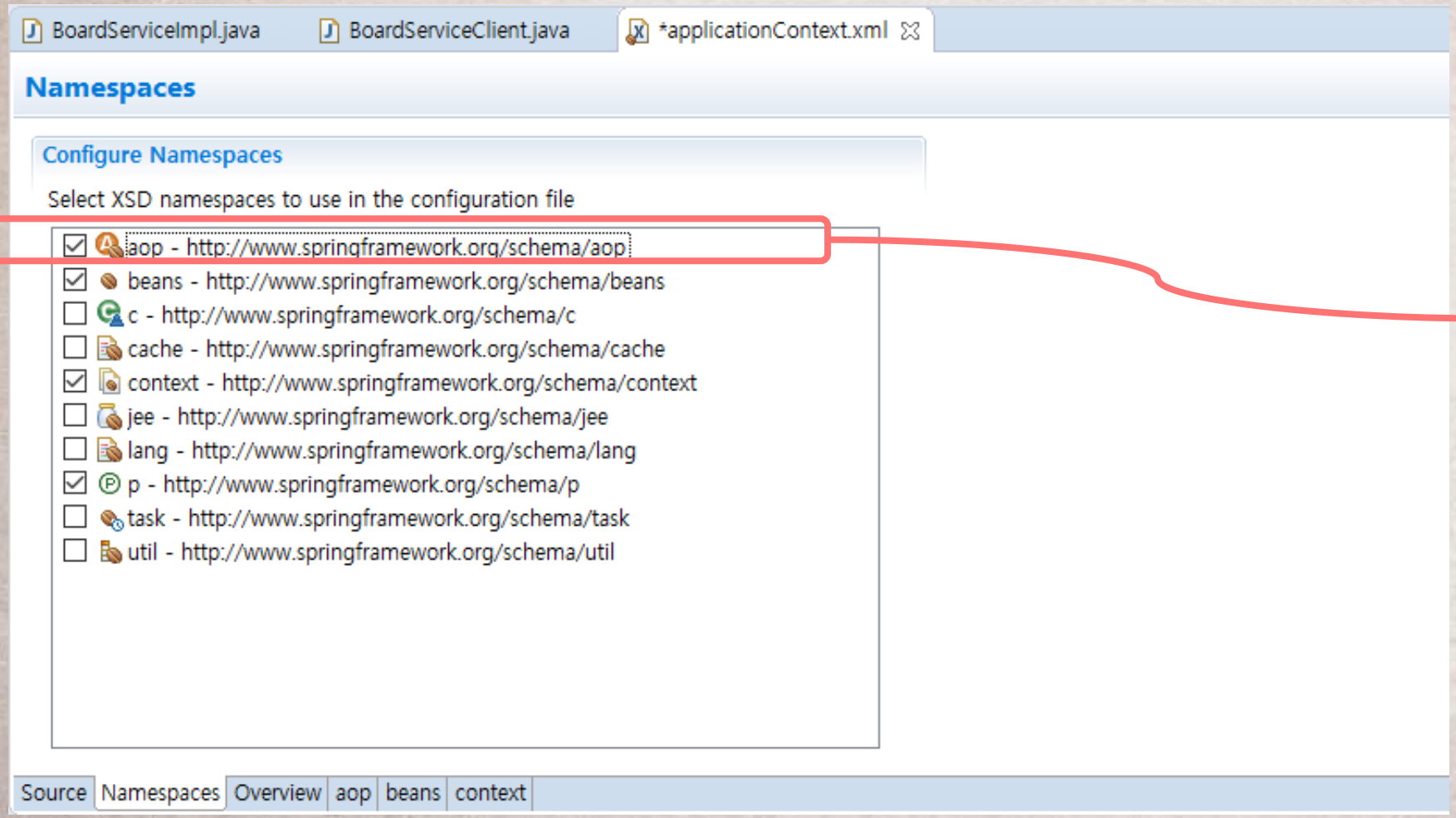




추가됨

applicationContext.xml 수정 - aop 네임스페이스 추가

- applicatonContext.xml을 열고 아래 탭 항목에서 “Namespaces” 선택



aop 항목을 체크(선택)

BoardServiceImpl.java | BoardServiceClient.java | applicationContext.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd
        http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.3.xsd">

    <context:component-scan base-package="kr.ac.inje.comsi"></context:component-scan>
    <!--
    <bean id="userService" class="kr.ac.inje.comsi.user.impl.UserServiceImpl">
        <property name="userDAO" ref="userDAO"></property>
    </bean>

    <bean id="userDAO" class="kr.ac.inje.comsi.user.impl.UserDAO"></bean>
     -->
</beans>
```

추가

추가

우선, LogAdvice 클래스를 <bean> 등록하고 AOP 관련 설정을 다음과 같이 추가한다.

변경된 applicationContext.xml - aop 관련 설정 추가

BoardServiceImpl.java | BoardServiceClient.java | applicationContext.xml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xmlns:p="http://www.springframework.org/schema/p"
5      xmlns:context="http://www.springframework.org/schema/context"
6      xmlns:aop="http://www.springframework.org/schema/aop"
7      xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
8          http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd
9          http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.3.xsd">
10
11     <context:component-scan base-package="kr.ac.inje.comsi"></context:component-scan>
12
13     <!-- LogAdvice bean 등록 -->
14     <bean id="log" class="kr.ac.inje.comsi.common.LogAdvice"></bean>
15     <!-- aop 관련 설정 -->
16     <aop:config>
17         <aop:pointcut expression="execution(* kr.ac.inje.comsi..*Impl.*(..))" id="allPointcut"/>
18         <aop:aspect ref="log">
19             <aop:before method="printLog" pointcut-ref="allPointcut"/>
20         </aop:aspect>
21     </aop:config>
22 </beans>
23
24
25
```

실행

Console | Progress | Problems

BoardWeb - run [Gradle Project] run in D:₩workspace₩java₩2017_1₩BoardWeb (2017. 9. 7 오후 10:09:55)

```
Gradle Version: 3.3
Java Home: C:\Program Files\Java\jdk1.8.0_144
JVM Arguments: None
Program Arguments: -DmainClass=kr.ac.inje.comsi.board.BoardServiceClient
Gradle Tasks: run

:compileJava
:processResources
:classes
:run9월 07, 2017 10:09:56 오후 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
정보: Loading XML bean definitions from class path resource [applicationContext.xml]
9월 07, 2017 10:09:56 오후 org.springframework.context.support.GenericXmlApplicationContext prepareRefresh
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@6e2c634b: startup date [Thu Sep 07 22:09:56 KST 2017]; rc

[공통 로그] 비즈니스 로직 수행 전 동작
===> JDBC로 insertBoard() 기능 처리
[공통 로그] 비즈니스 로직 수행 전 동작
===> JDBC로 getBoardList() 기능 처리
---> BoardVO [seq=4, title=AOP실행, writer=홍길동, content=임시 내용 ....................., regDate=2017-09-07, cnt=0]
---> BoardVO [seq=3, title=AOP테스트, writer=홍길동, content=임시 내용 ....................., regDate=2017-09-07, cnt=0]
---> BoardVO [seq=2, title=임시 제목, writer=홍길동, content=임시 내용 ....................., regDate=2017-09-07, cnt=0]
---> BoardVO [seq=1, title=가입인사, writer=관리자, content=잘 부탁드립니다...., regDate=2017-09-06, cnt=0]
9월 07, 2017 10:09:57 오후 org.springframework.context.support.GenericXmlApplicationContext doClose
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@6e2c634b: startup date [Thu Sep 07 22:09:56 KST 2017]; root

BUILD SUCCESSFUL

Total time: 1.296 secs
```

printLog() 실행

printLog() 실행

만약 LogAdvice 클래스를 Log4jAdvice로 변경한다면?

BoardServiceImpl.java | BoardServiceClient.java | applicationContext.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd
        http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.3.xsd">

    <context:component-scan base-package="kr.ac.inje.comsi"></context:component-scan>

    <!-- LogAdvice bean 등록 -->
    <bean id="log" class="kr.ac.inje.comsi.common.LogAdvice"></bean>
    <!-- aop 관련 설정 -->
    <aop:config>
        <aop:pointcut expression="execution(* kr.ac.inje.comsi..*Impl.*(..))" id="allPointcut"/>
        <aop:aspect ref="log">
            <aop:before method="printLog" pointcut-ref="allPointcut"/>
        </aop:aspect>
    </aop:config>
</beans>
```

Log4jAdvice로 변경

printLogging으로 변경

포인트 컷 표현식의 구성과 의미

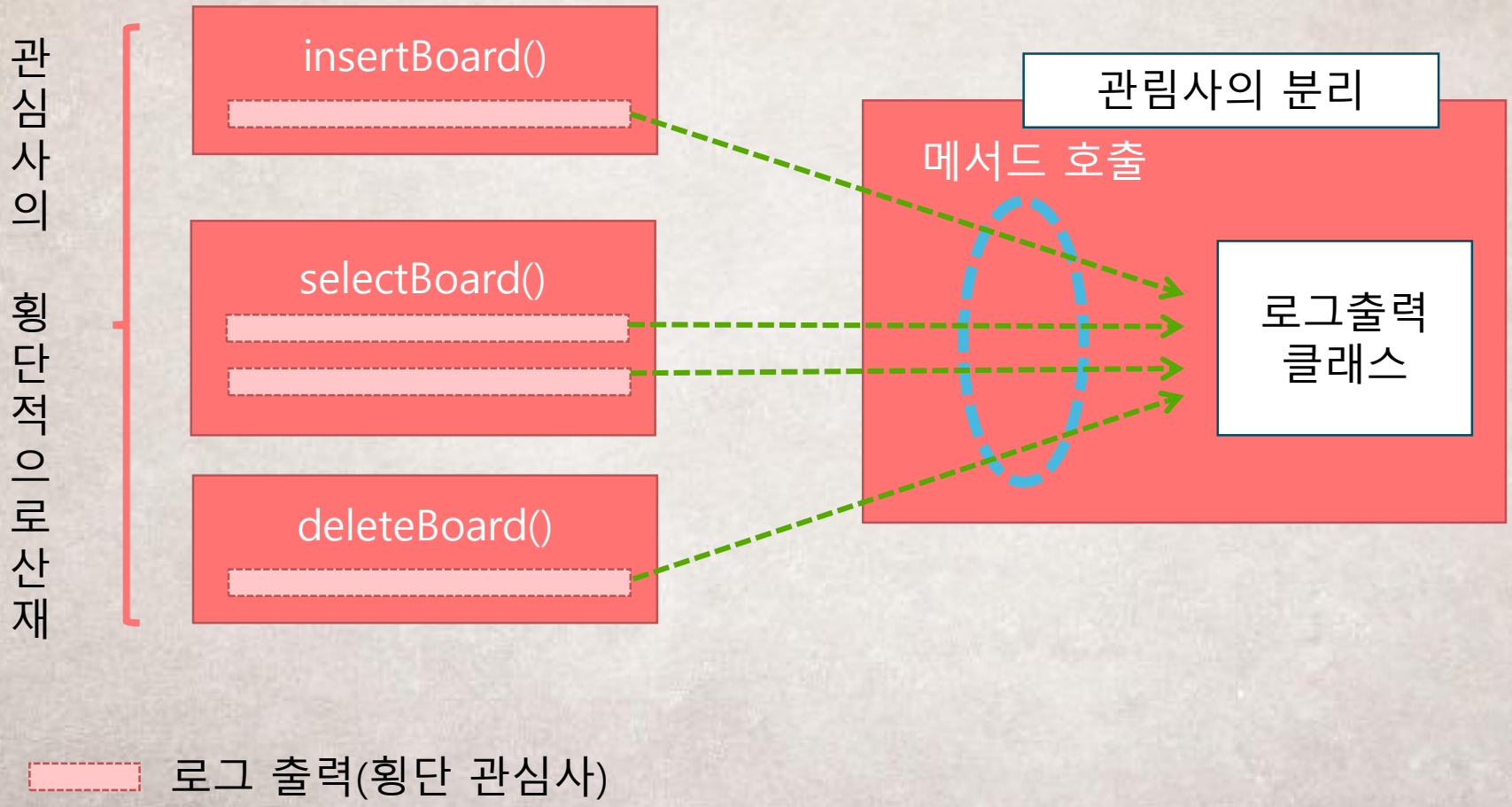
```
*    kr.ac.inje.comsi. .    *impl    .    *(..)
*    kr.ac.inje.comsi. .    *impl    .    get*(..)
```

리턴타입 패키지 경로 클래스 명 메소드명 및 매개변수

정리하면,

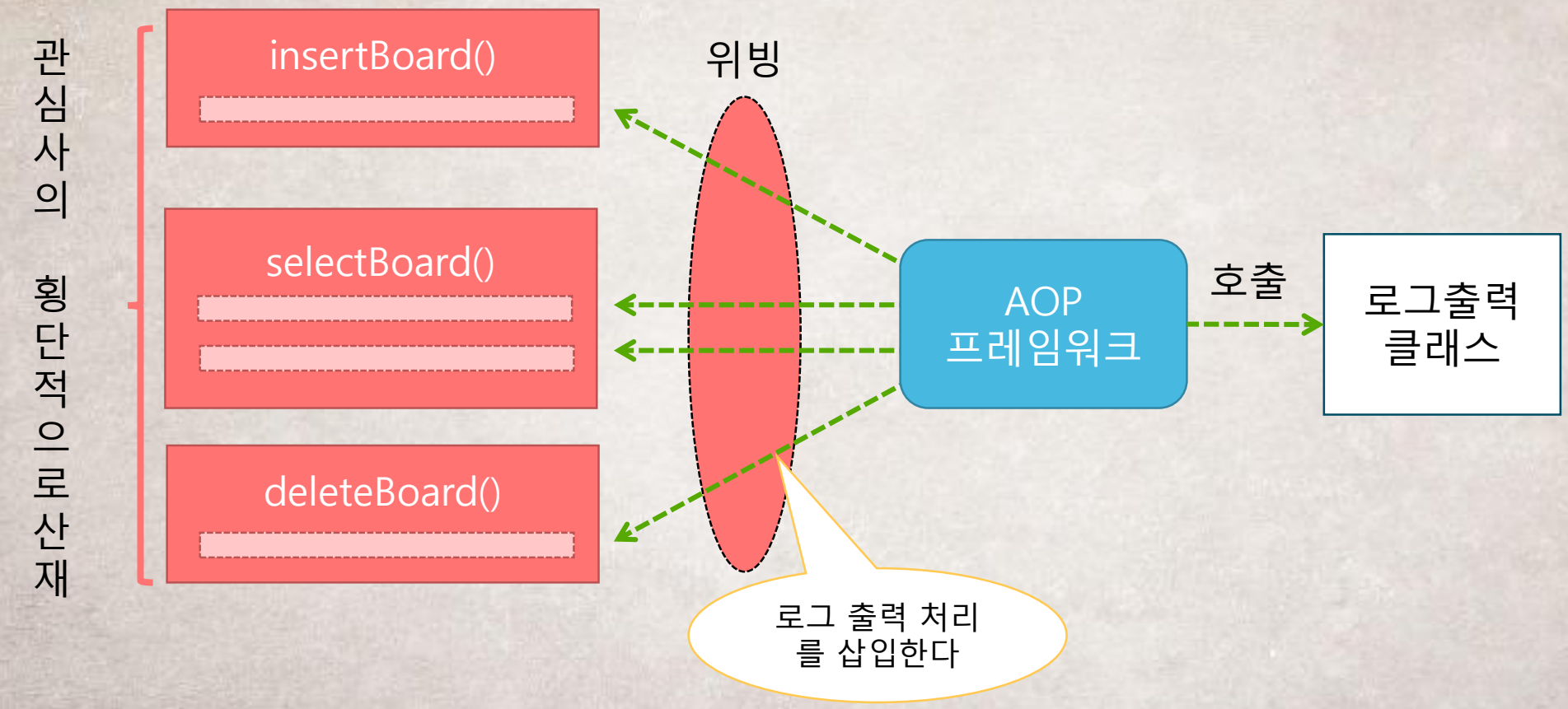
관점 지향과 횡단 관심사의 분리

- 기존 객체 지향에서 관심사 분리



관점 지향과 횡단 관심사의 분리

- AOP에서 횡단 관심사 분리와 위빙

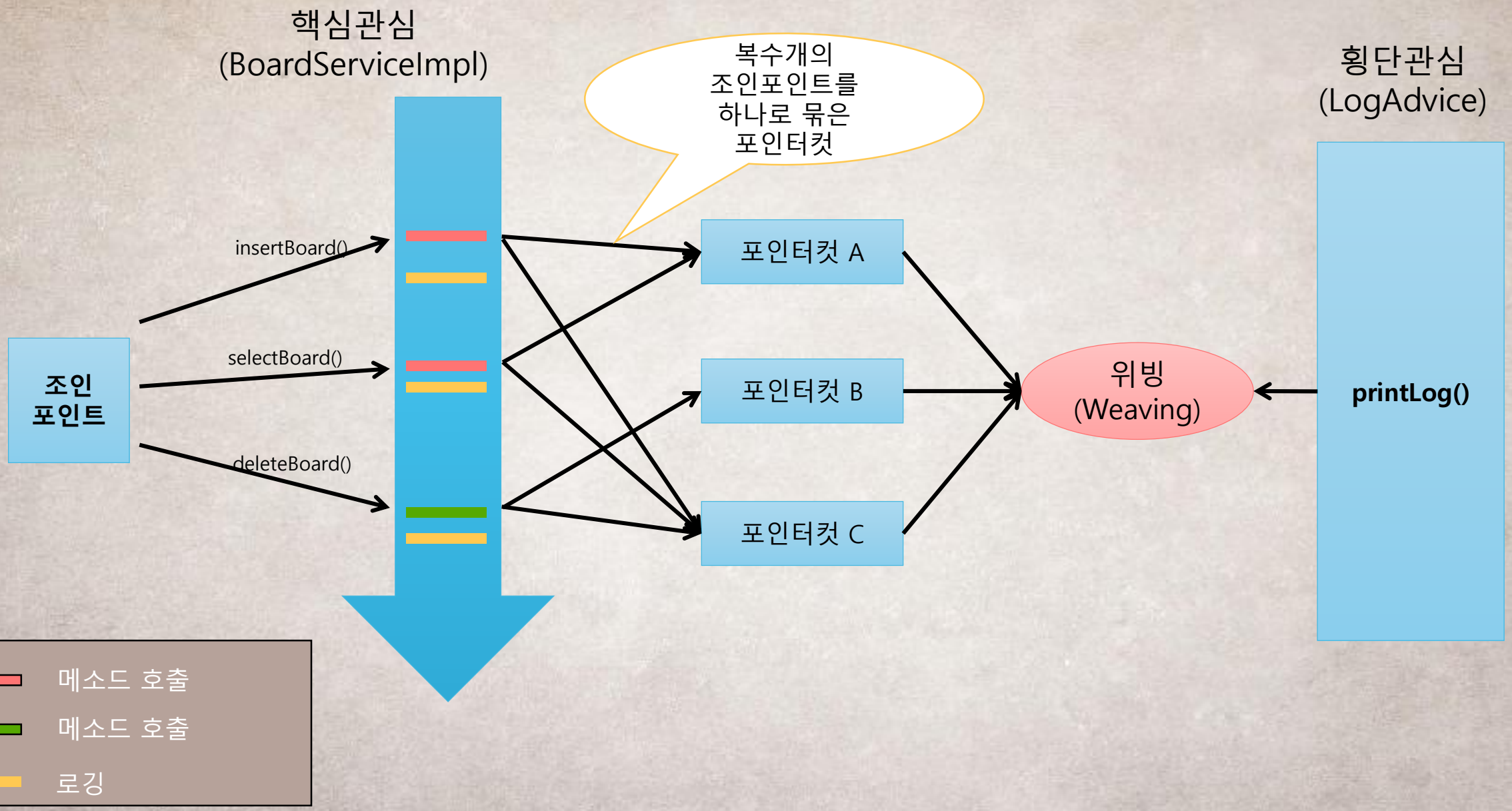


용어 정리

AOP(Aspect Oriented Programming) - 용어

용어	내용
어드바이스 (Advice)	실질적인 비즈니스 로직을 구현하고 있는 "핵심 비즈니스 로직 구현부"에 추가적인 기능을 제공하는 것(횡단관심사)
조인포인트 (Joinpoint)	클래스의 인스턴스 생성 시점, 메소드를 호출하는 시점, Exception이 발생하는 시점과 같이 어플리케이션을 실행할 때, 특정 작업이 실행되는 시점(비즈니스 매소드)을 의미한다.
포인트컷 (Pointcut)	분리된 기능(횡단관심사, 로깅 ...)들을 결합시키기 위한 규칙(패턴)이 필요한 데, 이와 같은 패턴을 포인터컷이라고 한다.
위빙 (Weaving)	"엮는다, 실로 짠다" 분리된 기능들을 하나로 결합시키는 것을 의미. 분리하여 개발된 기능들을 Weaving하는 작업을 도와주는 것이 AOP 툴이 하는 역할이다.
어스펙트 (Aspect)	어드바이스와 포인트컷을 합쳐서 하나의 Aspect라고 한다. 즉, 일정한 패턴을 가지는 클래스에 어드바이스를 적용하도록 지원할 수 있는 것을 Aspect라고 한다. 혹은 어드바이저(Advisor)라고 한다.
어드바이저 (Advisor)	어드바이스와 포인트컷을 하나로 묶어 다루는 것을 말한다.

AOP 용어와 개념도


핵심관심
(BoardServiceImpl)
복수개의
조인포인트를
하나로 묶은
포인터컷
횡단관심
(LogAdvice)
insertBoard()
selectBoard()
deleteBoard()
조인
포인트
포인터컷 A
포인터컷 B
포인터컷 C
위빙
(Weaving)
printLog()
메소드 호출
메소드 호출
로깅