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

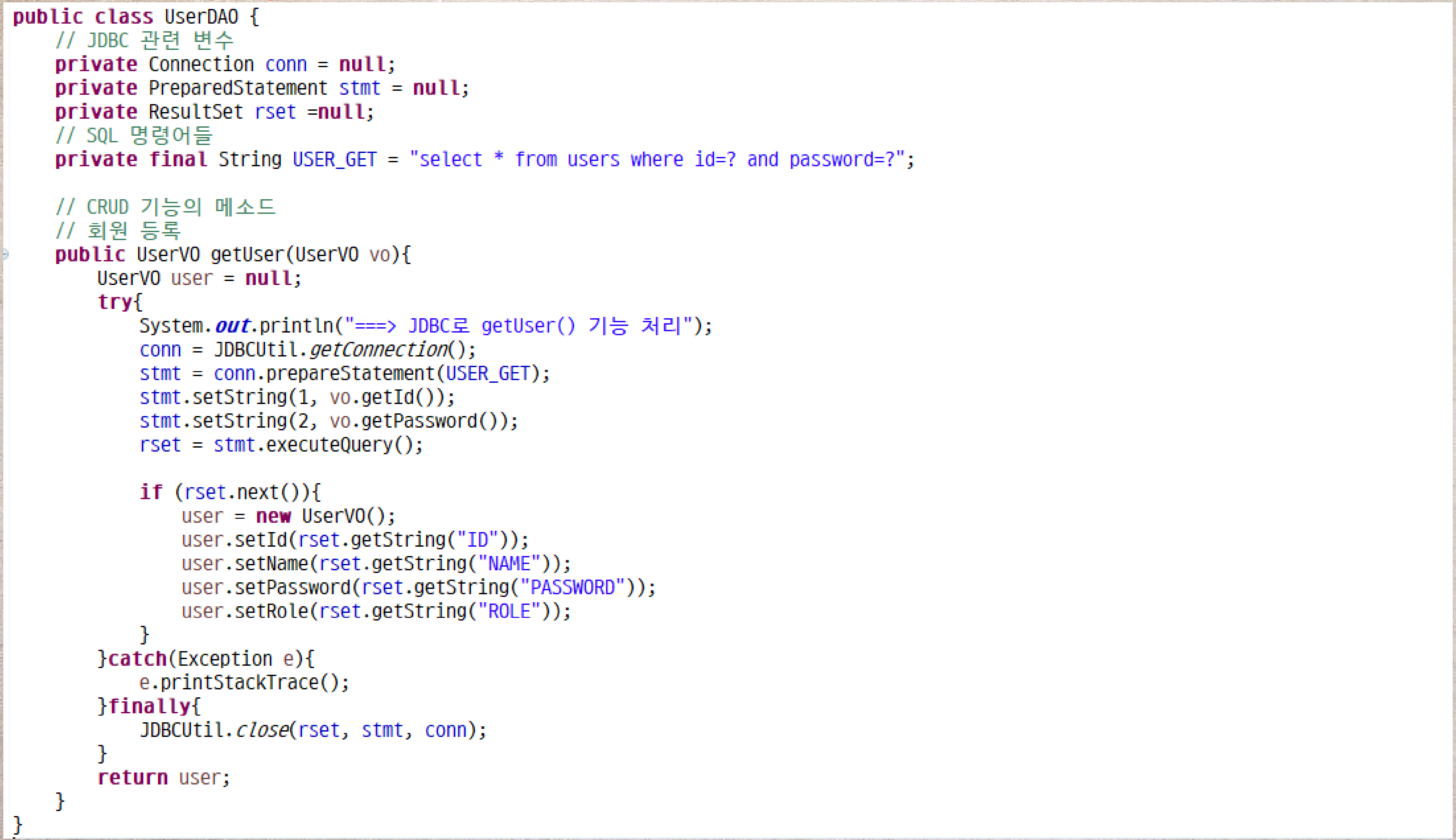
스프링 JDBC

- JDBC(Java Database Connectivity)

➔ 자바에서 데이터베이스에 접속할 수 있도록 하는 자바 API이다. **JDBC**는 데이터베이스에서 자료를 쿼리하거나 업데이트하는 방법을 제공

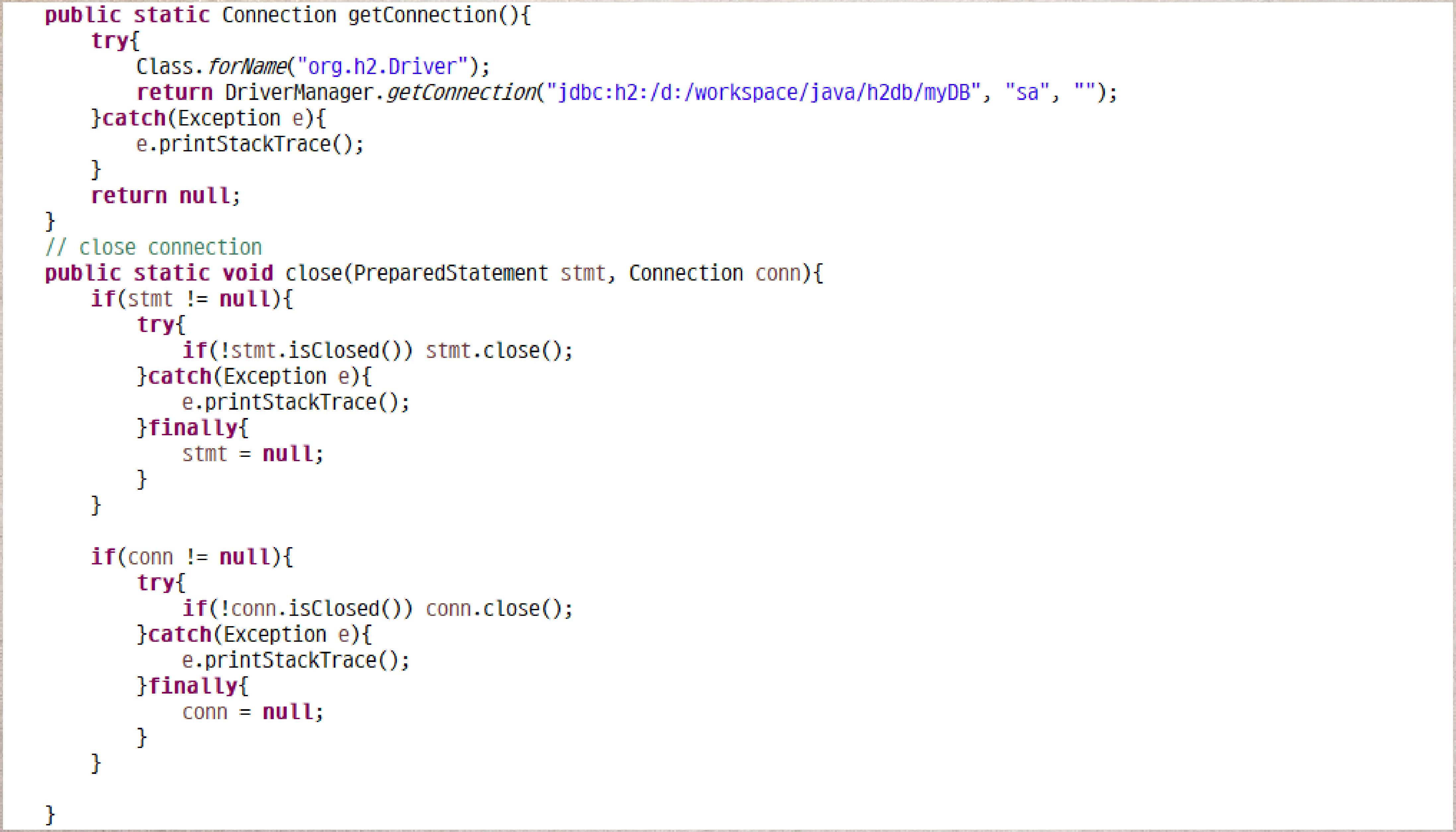
- JDBC를 이용한 DB 연동 프로그램을 개발하면 데이터베이스에 비종속적인 DB 연동 로직을 구현할 수 있다.
  - But, JDBC 프로그램을 이용하면 개발자가 작성해야 할 코드가 너무 많다

UserDAO.java



```java
public class UserDAO {
    // JDBC 관련 변수
    private Connection conn = null;
    private PreparedStatement stmt = null;
    private ResultSet rset =null;
    // SQL 명령어들
    private final String USER_GET = "select * from users where id=? and password=?";

    // CRUD 기능의 메소드
    // 회원 등록
    public UserVO getUser(UserVO vo){
        UserVO user = null;
        try{
            System.out.println("===> JDBC로 getUser() 기능 처리");
            conn = JDBCUtil.getConnection();
            stmt = conn.prepareStatement(USER_GET);
            stmt.setString(1, vo.getId());
            stmt.setString(2, vo.getPassword());
            rset = stmt.executeQuery();

            if (rset.next()){
                user = new UserVO();
                user.setId(rset.getString("ID"));
                user.setName(rset.getString("NAME"));
                user.setPassword(rset.getString("PASSWORD"));
                user.setRole(rset.getString("ROLE"));
            }
        }catch(Exception e){
            e.printStackTrace();
        }finally{
            JDBCUtil.close(rset, stmt, conn);
        }
        return user;
    }
}
```

JDBCUtil.Java - 반복코드??



```
public static Connection getConnection(){
    try{
        Class.forName("org.h2.Driver");
        return DriverManager.getConnection("jdbc:h2:/d:/workspace/java/h2db/myDB", "sa", "");
    }catch(Exception e){
        e.printStackTrace();
    }
    return null;
}
// close connection
public static void close(PreparedStatement stmt, Connection conn){
    if(stmt != null){
        try{
            if(!stmt.isClosed()) stmt.close();
        }catch(Exception e){
            e.printStackTrace();
        }finally{
            stmt = null;
        }
    }

    if(conn != null){
        try{
            if(!conn.isClosed()) conn.close();
        }catch(Exception e){
            e.printStackTrace();
        }finally{
            conn = null;
        }
    }

}
```

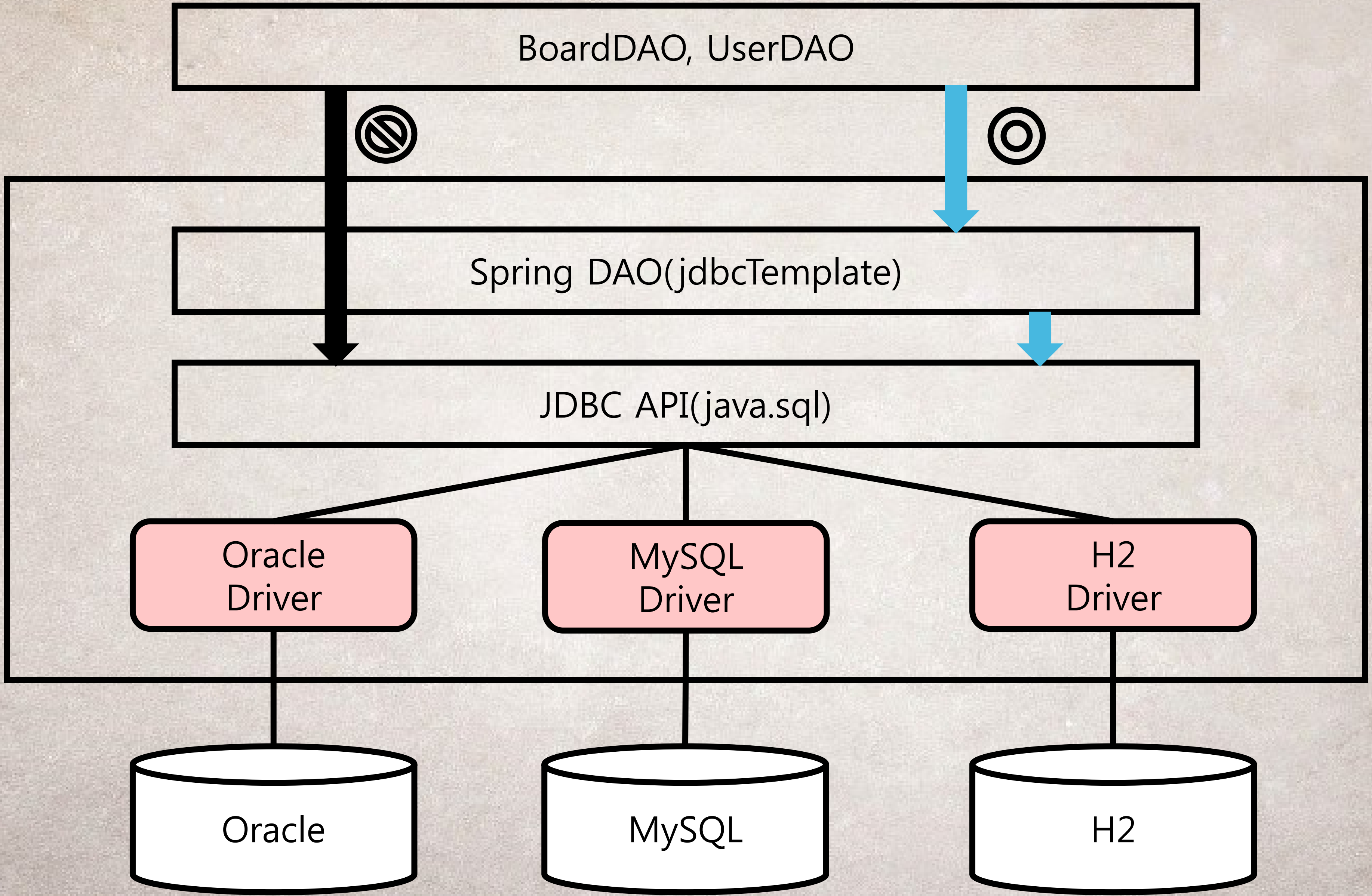
JdbcTemplate 클래스

- 스프링에서 JDBC 기반의 DB 연동 프로그램을 쉽게 개발할 수 있도록 jdbcTemplate 클래스 지원

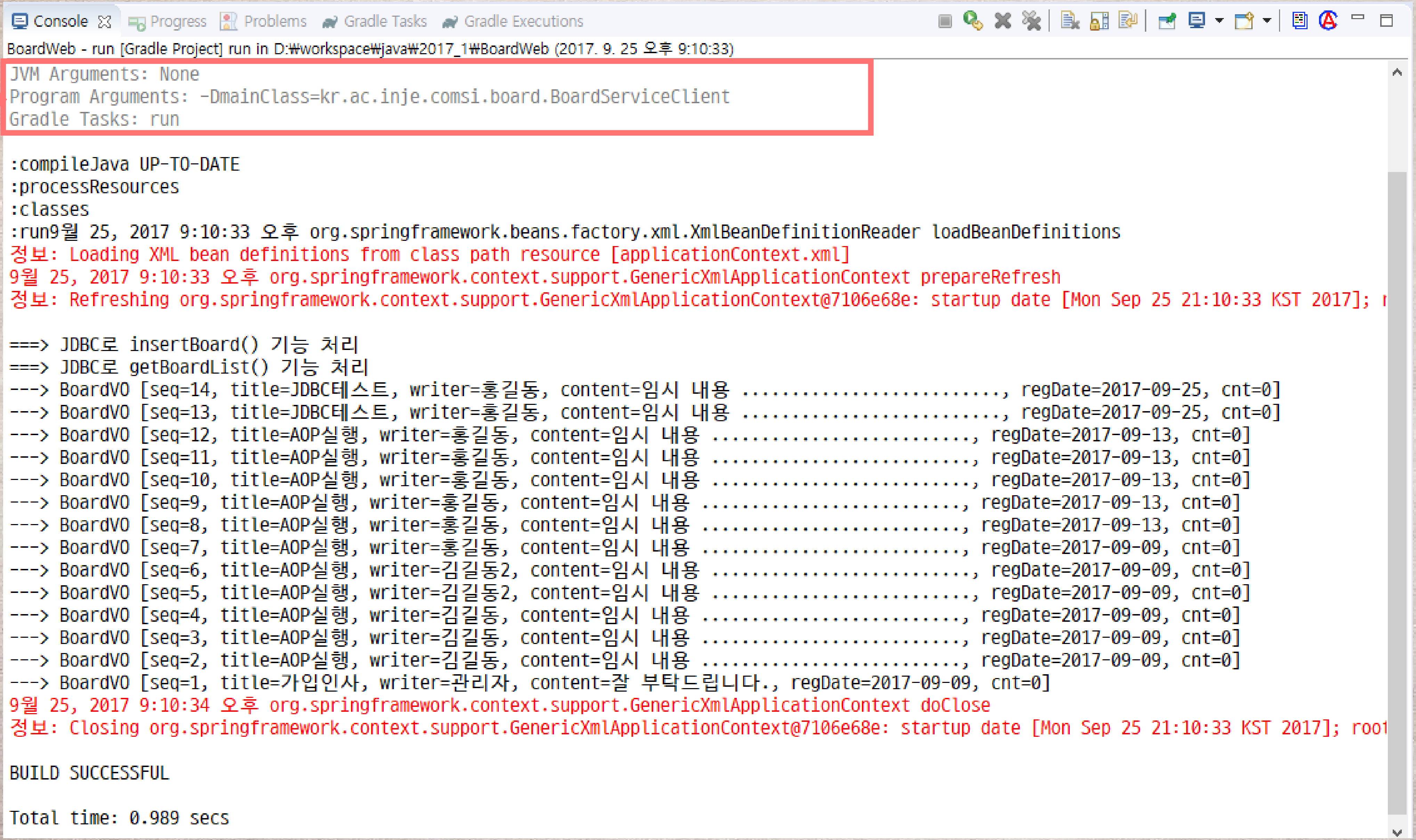
- JdbcTemplate은 Gof 디자인 패턴 중 템플릿 메소드 패턴이 적용된 클래스
  - 템플릿 메소드 패턴은 복잡하고 반복되는 알고리즘을 캡슐화해서 재사용하는 패턴으로 정의

- JDBC처럼 코딩 순서가 정형화된 기술에서 유용하게 사용
  - 반복되는 DB 연동 로직은 JdbcTemplate 클래스의 템플릿 메소드가 제공
  - 개발자는 달라지는 SQL 구문과 설정값 만 신경 씀

JdbcTemplate의 위치와 역할

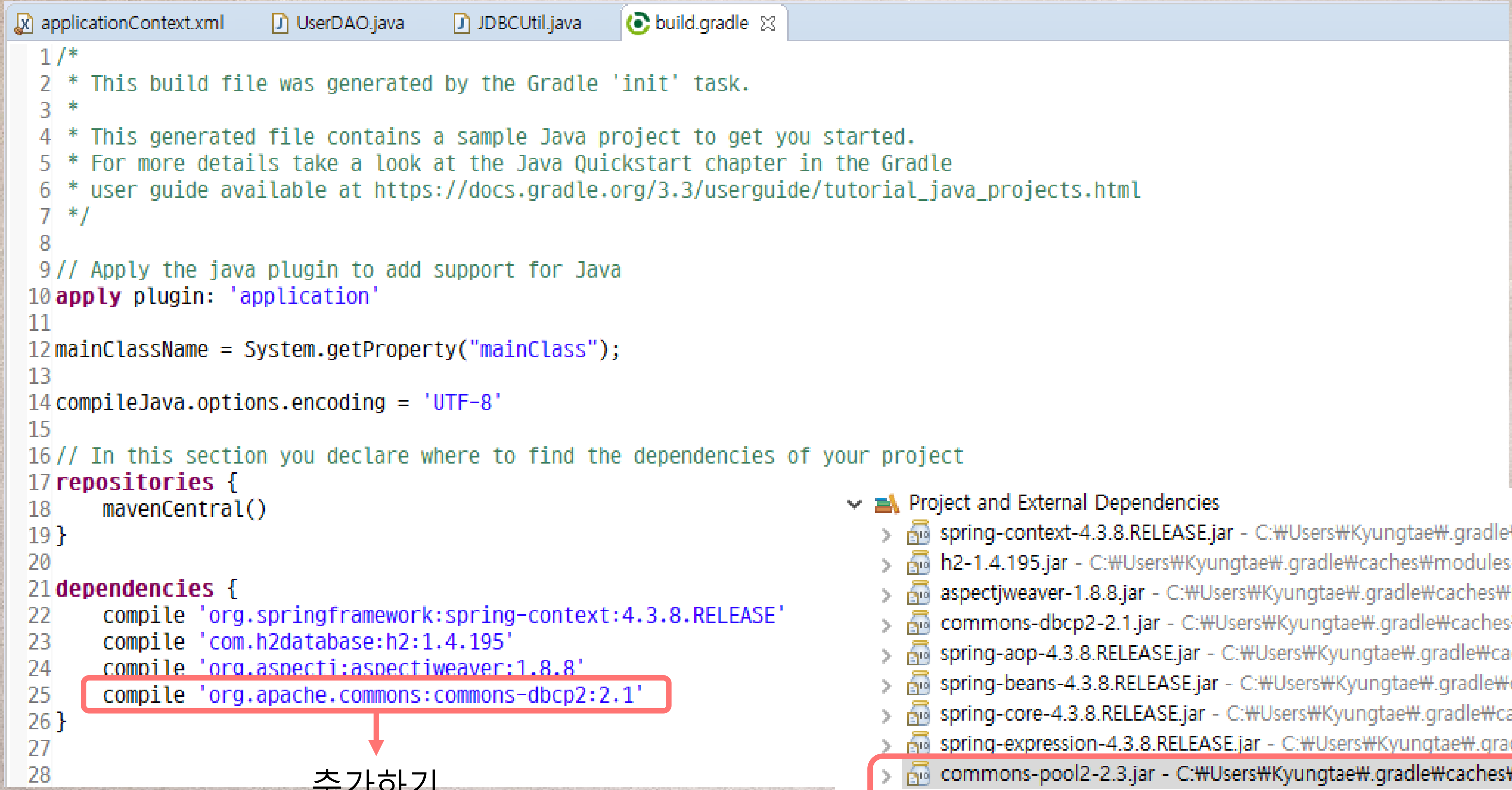


JDBC 설정 후 DB 연동 결과



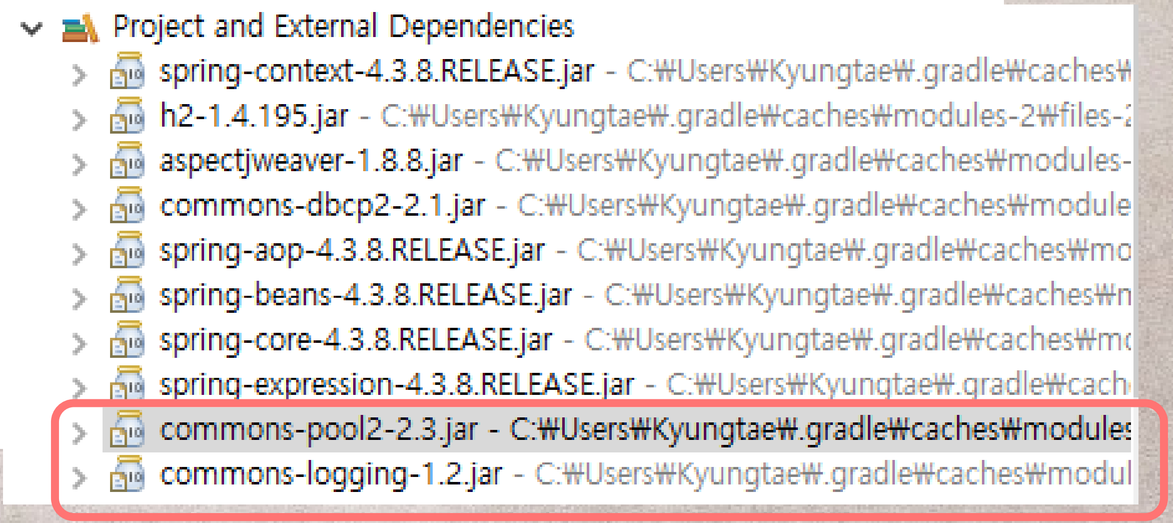
```
BoardWeb - run [Gradle Project] run in D:₩workspace₩java₩2017_1₩BoardWeb (2017. 9. 25 오후 9:10:33)
JVM Arguments: None
Program Arguments: -DmainClass=kr.ac.inje.comsi.board.BoardServiceClient
Gradle Tasks: run

:compileJava UP-TO-DATE
:processResources
:classes
:run9월 25, 2017 9:10:33 오후 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
정보: Loading XML bean definitions from class path resource [applicationContext.xml]
9월 25, 2017 9:10:33 오후 org.springframework.context.support.GenericXmlApplicationContext prepareRefresh
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@7106e68e: startup date [Mon Sep 25 21:10:33 KST 2017]; r

===> JDBC로 insertBoard() 기능 처리
===> JDBC로 getBoardList() 기능 처리
---> BoardVO [seq=14, title=JDBC테스트, writer=홍길동, content=임시 내용 ..........................., regDate=2017-09-25, cnt=0]
---> BoardVO [seq=13, title=JDBC테스트, writer=홍길동, content=임시 내용 ..........................., regDate=2017-09-25, cnt=0]
---> BoardVO [seq=12, title=AOP실행, writer=홍길동, content=임시 내용 ..........................., regDate=2017-09-13, cnt=0]
---> BoardVO [seq=11, title=AOP실행, writer=홍길동, content=임시 내용 ..........................., regDate=2017-09-13, cnt=0]
---> BoardVO [seq=10, title=AOP실행, writer=홍길동, content=임시 내용 ..........................., regDate=2017-09-13, cnt=0]
---> BoardVO [seq=9, title=AOP실행, writer=홍길동, content=임시 내용 ..........................., regDate=2017-09-13, cnt=0]
---> BoardVO [seq=8, title=AOP실행, writer=홍길동, content=임시 내용 ..........................., regDate=2017-09-13, cnt=0]
---> BoardVO [seq=7, title=AOP실행, writer=홍길동, content=임시 내용 ..........................., regDate=2017-09-09, cnt=0]
---> BoardVO [seq=6, title=AOP실행, writer=김길동2, content=임시 내용 ..........................., regDate=2017-09-09, cnt=0]
---> BoardVO [seq=5, title=AOP실행, writer=김길동2, content=임시 내용 ..........................., regDate=2017-09-09, cnt=0]
---> BoardVO [seq=4, title=AOP실행, writer=김길동, content=임시 내용 ..........................., regDate=2017-09-09, cnt=0]
---> BoardVO [seq=3, title=AOP실행, writer=김길동, content=임시 내용 ..........................., regDate=2017-09-09, cnt=0]
---> BoardVO [seq=2, title=AOP실행, writer=김길동, content=임시 내용 ..........................., regDate=2017-09-09, cnt=0]
---> BoardVO [seq=1, title=가입인사, writer=관리자, content=잘 부탁드립니다., regDate=2017-09-09, cnt=0]
9월 25, 2017 9:10:34 오후 org.springframework.context.support.GenericXmlApplicationContext doClose
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@7106e68e: startup date [Mon Sep 25 21:10:33 KST 2017]; root

BUILD SUCCESSFUL

Total time: 0.989 secs
```

스프링 JDBC 설정 - 라이브러리 추가



```
/*
 * This build file was generated by the Gradle 'init' task.
 *
 * This generated file contains a sample Java project to get you started.
 * For more details take a look at the Java Quickstart chapter in the Gradle
 * user guide available at https://docs.gradle.org/3.3/userguide/tutorial_java_projects.html
 */

// Apply the java plugin to add support for Java
apply plugin: 'application'

mainClassName = System.getProperty("mainClass");

compileJava.options.encoding = 'UTF-8'

// In this section you declare where to find the dependencies of your project
repositories {
    mavenCentral()
}

dependencies {
    compile 'org.springframework:spring-context:4.3.8.RELEASE'
    compile 'com.h2database:h2:1.4.195'
    compile 'org.aspectj:aspectjweaver:1.8.8'
    compile 'org.apache.commons:commons-dbcp2:2.1'
}

```

추가하기



```
Project and External Dependencies
  spring-context-4.3.8.RELEASE.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩
  h2-1.4.195.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩modules-2₩files-2
  aspectjweaver-1.8.8.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩modules-
  commons-dbcp2-2.1.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩module
  spring-aop-4.3.8.RELEASE.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩mo
  spring-beans-4.3.8.RELEASE.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩n
  spring-core-4.3.8.RELEASE.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩mo
  spring-expression-4.3.8.RELEASE.jar - C:₩Users₩Kyungtae₩.gradle₩cach
  commons-pool2-2.3.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩modules
  commons-logging-1.2.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩modul
```

JdbcTemplate 메소드

- update() 메소드: “?”에 값을 설정하는 방식에 따라 두 가지 형태가 사용
  - 첫번째: SQL 구문에 설정된 “?”수만큼 값들을 나열하는 방식

메소드	int update(String sql, Object ... args)

사용 예	
	```
	// 글 수정
	public void updateBoard(BoardVO vo) {
	    String BOARD_UPDATE = “update board set title=?, content=? where seq=?”;
	    int cnt = jdbcTemplate.update(BOARD_UPDATE, vo.getTitle(), vo.getContent(), vo.getSeq());
	    System.out.println(cnt + “건 데이터 수정”);
	}
	```

  - 두번째: Object 배열 객체에 SQL 구문에 설정된 “?” 수만큼의 값들을 세팅하여 배열 객체를 두번째 인자로 전달하는 방식

메소드	int update(String sql, Object ... args)

사용 예	
	```
	// 글 수정
	public void updateBoard(BoardVO vo) {
	    String BOARD_UPDATE = “update board set title=?, content=? where seq=?”;
	    Object[] args = {vo.getTitle(), vo.getContent(), vo.getSeq() };
	    int cnt = jdbcTemplate.update(BOARD_UPDATE,args);
	    System.out.println(cnt + “건 데이터 수정”);
	}
	```

- queryForInt()메소드
  - SELECT 구문으로 검색된 정숫값을 받을 때 사용하며 매개변수의 의미는 앞에서 본 update()메소드와 같다.

메소드	int queryForInt(String sql) int queryForInt(String sql, Object... args) int queryForInt(String sql, Object[] args)

사용 예	`// 전체 게시글 수 조회`
	`public int getBoardTotalCount(BoardVO vo) {`
	`    String BOARD_TOT_COUNT = “ select count(*) from board ” ;`
	
	`    Object[] args = {vo.getTitle(), vo.getContent(), vo.getSeq() };`
	`    int cnt = jdbcTemplate.queryForInt(BOARD_TOT_COUNT);`
	`    System.out.println( “ 전체 게시글 수:  ” + cnt + “ 건 ” );`
	`}`

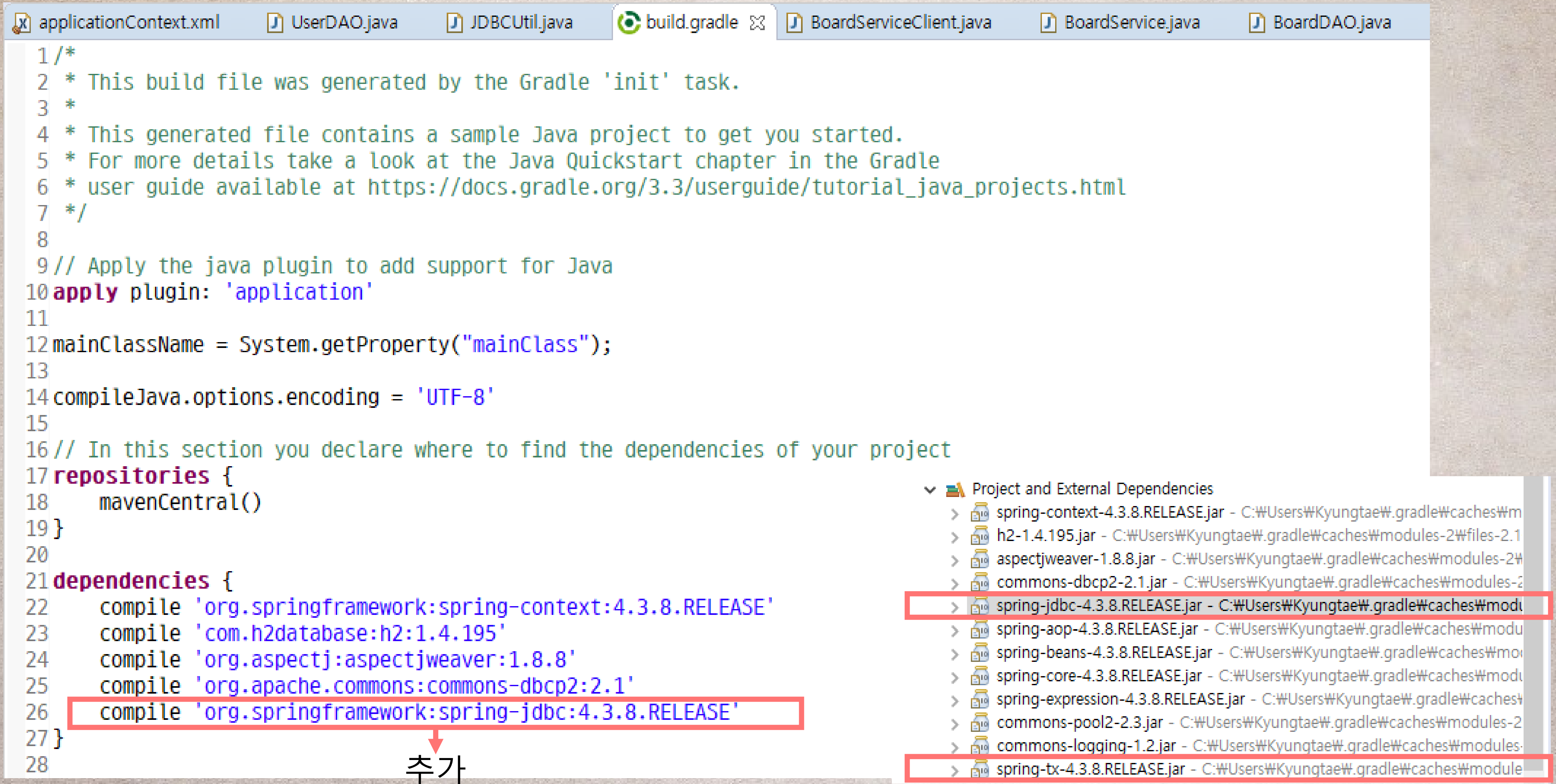
- queryForObject()메소드
  - SELECT 구문의 실행 결과를 특정 자바 객체(Value Object)로 매핑하여 리턴받을 때 사용한다.
  - 검색 결과가 없거나 검색 결과가 두개 이상이면 예외(IncorrectResultSizeDataAccessException)를 발생시킨다.

메소드	int queryForObject(String sql) int queryForObject(String sql, RowMapper<T> rowMapper) int queryForObject(String sql, Object[] args, RowMapper<T> rowMapper)

사용 예	`// 글 상세 조회`
	`public BoardVO getBoard(BoardVO vo) {`
	`    String BOARD_GET = " select * from board where seq=? ";`
	`    Object[] args = {vo.getSeq() };`
	`    return jdbcTemplate.queryForObject(BOARD_GET, args, new BoardRowMapper());`
	`}`

- 검색 결과를 특정 VO(Value Object) 객체에 매핑하여 리턴하려면 RowMapper 인터페이스를 구현한 RowMapper 클래스가 필요
- 따라서 RowMapper 클래스는 테이블당 하나씩 필요함
- RowMapper 인터페이스는 mapRow() 메소드가 있어서 검색 결과로 얻어낸 Row 정보를 어떤 VO에 어떻게 매핑할 것인지 구현하면 된다.

RowMapper 인터페이스 구현 클래스 만들기 예제



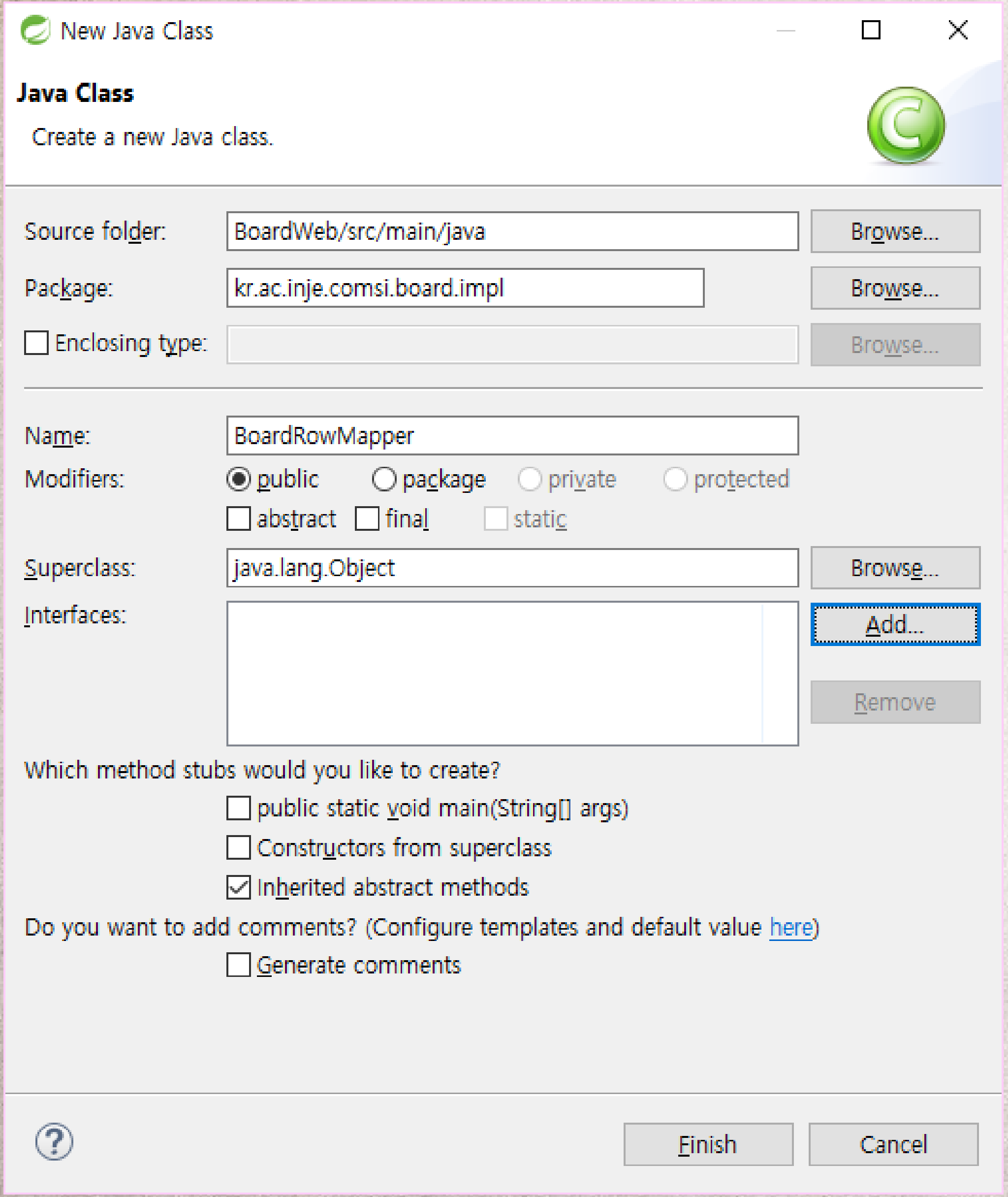
applicationContext.xml | UserDAO.java | JDBCUtil.java | build.gradle | BoardServiceClient.java | BoardService.java | BoardDAO.java

```
/*
 * This build file was generated by the Gradle 'init' task.
 *
 * This generated file contains a sample Java project to get you started.
 * For more details take a look at the Java Quickstart chapter in the Gradle
 * user guide available at https://docs.gradle.org/3.3/userguide/tutorial_java_projects.html
 */

// Apply the java plugin to add support for Java
apply plugin: 'application'

mainClassName = System.getProperty("mainClass");

compileJava.options.encoding = 'UTF-8'

// In this section you declare where to find the dependencies of your project
repositories {
    mavenCentral()
}

dependencies {
    compile 'org.springframework:spring-context:4.3.8.RELEASE'
    compile 'com.h2database:h2:1.4.195'
    compile 'org.aspectj:aspectjweaver:1.8.8'
    compile 'org.apache.commons:commons-dbcp2:2.1'
    compile 'org.springframework:spring-jdbc:4.3.8.RELEASE'
}

```

추가

```
Project and External Dependencies
  spring-context-4.3.8.RELEASE.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩m
  h2-1.4.195.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩modules-2₩files-2.1
  aspectjweaver-1.8.8.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩modules-2₩
  commons-dbcp2-2.1.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩modules-2
  spring-jdbc-4.3.8.RELEASE.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩modu
  spring-aop-4.3.8.RELEASE.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩modu
  spring-beans-4.3.8.RELEASE.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩mo
  spring-core-4.3.8.RELEASE.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩modu
  spring-expression-4.3.8.RELEASE.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩
  commons-pool2-2.3.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩modules-2
  commons-logging-1.2.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩modules-
  spring-tx-4.3.8.RELEASE.jar - C:₩Users₩Kyungtae₩.gradle₩caches₩module
```

RowMapper 인터페이스 구현 클래스 만들기



New Java Class

Java Class

Create a new Java class.

Source folder: BoardWeb/src/main/java

Package: kr.ac.inje.comsi.board.impl

☐ Enclosing type:

Name: BoardRowMapper

Modifiers: [✓ public] package private protected

☐ abstract ☐ final ☐ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

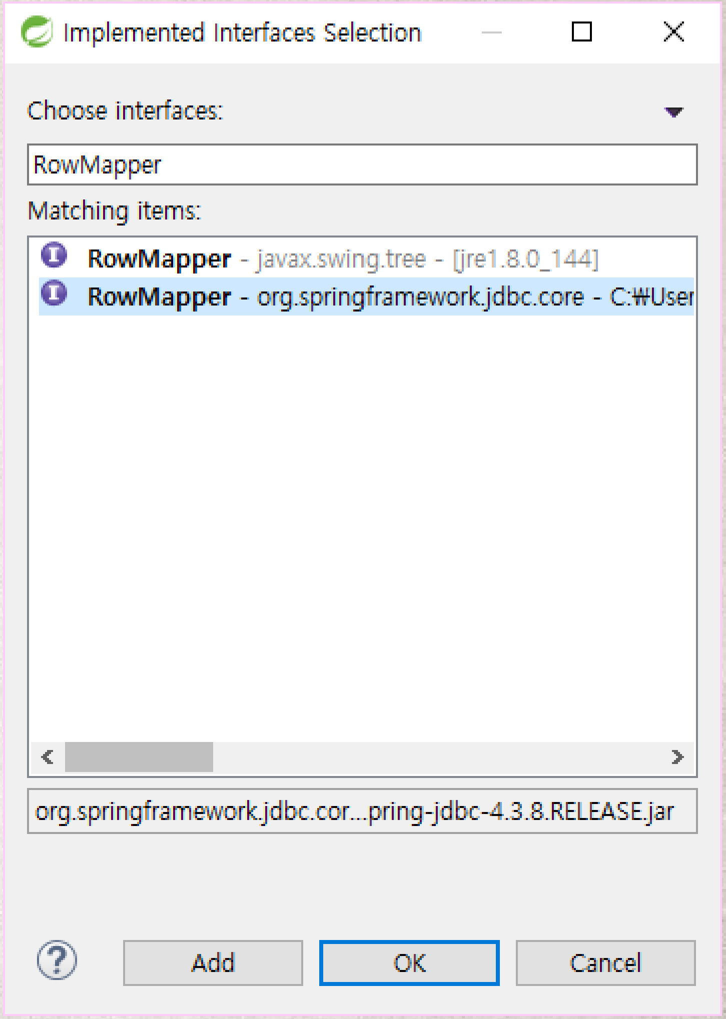
☐ public static void main(String[] args)

☐ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

☐ Generate comments



Implemented Interfaces Selection

Choose interfaces:

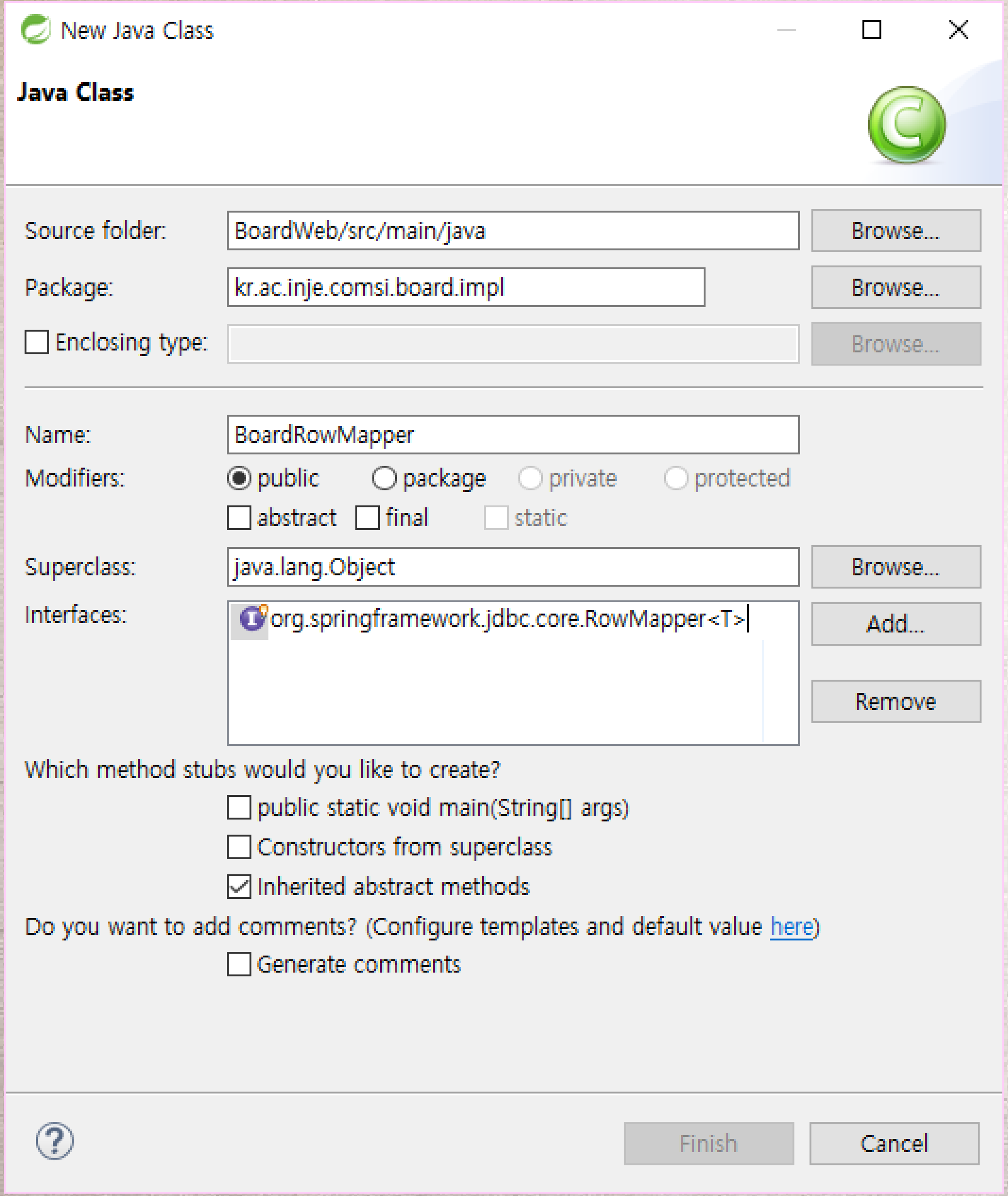
RowMapper

Matching items:

RowMapper - javax.swing.tree - [jre1.8.0_144]

RowMapper - org.springframework.jdbc.core - C:₩User

org.springframework.jdbc.cor...pring-jdbc-4.3.8.RELEASE.jar


New Java Class

Java Class

Source folder: BoardWeb/src/main/java

Package: kr.ac.inje.comsi.board.impl

[ ] Enclosing type:

Name: BoardRowMapper

Modifiers: [✓ public] package private protected

[ ] abstract [ ] final [ ] static

Superclass: java.lang.Object

Interfaces: org.springframework.jdbc.core.RowMapper<T>

Which method stubs would you like to create?

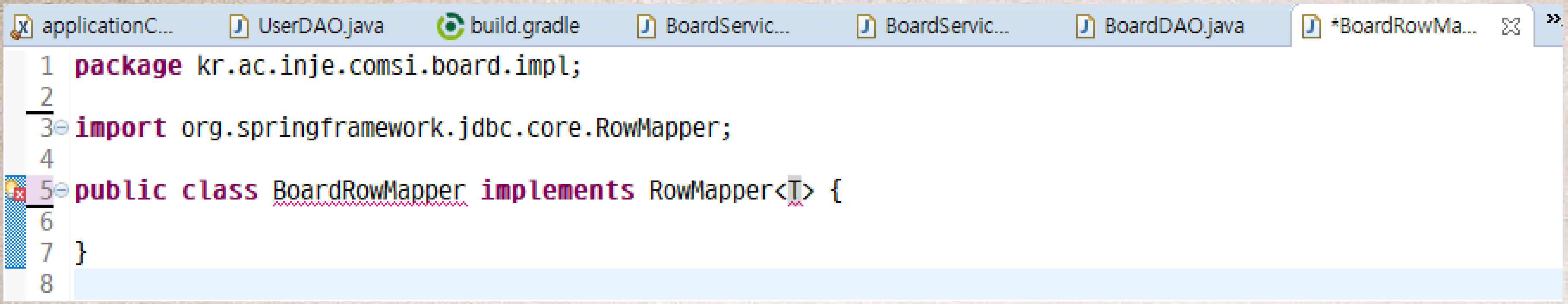
[ ] public static void main(String[] args)

[ ] Constructors from superclass

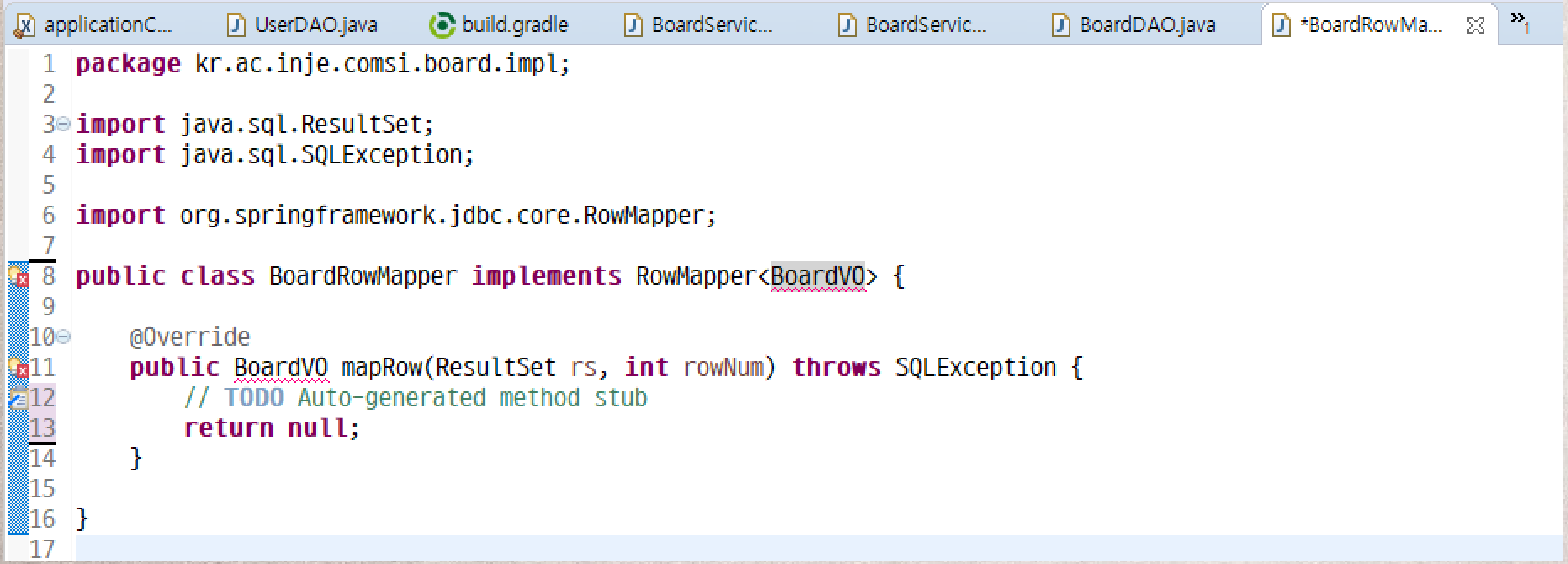
[✓] Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

[ ] Generate comments

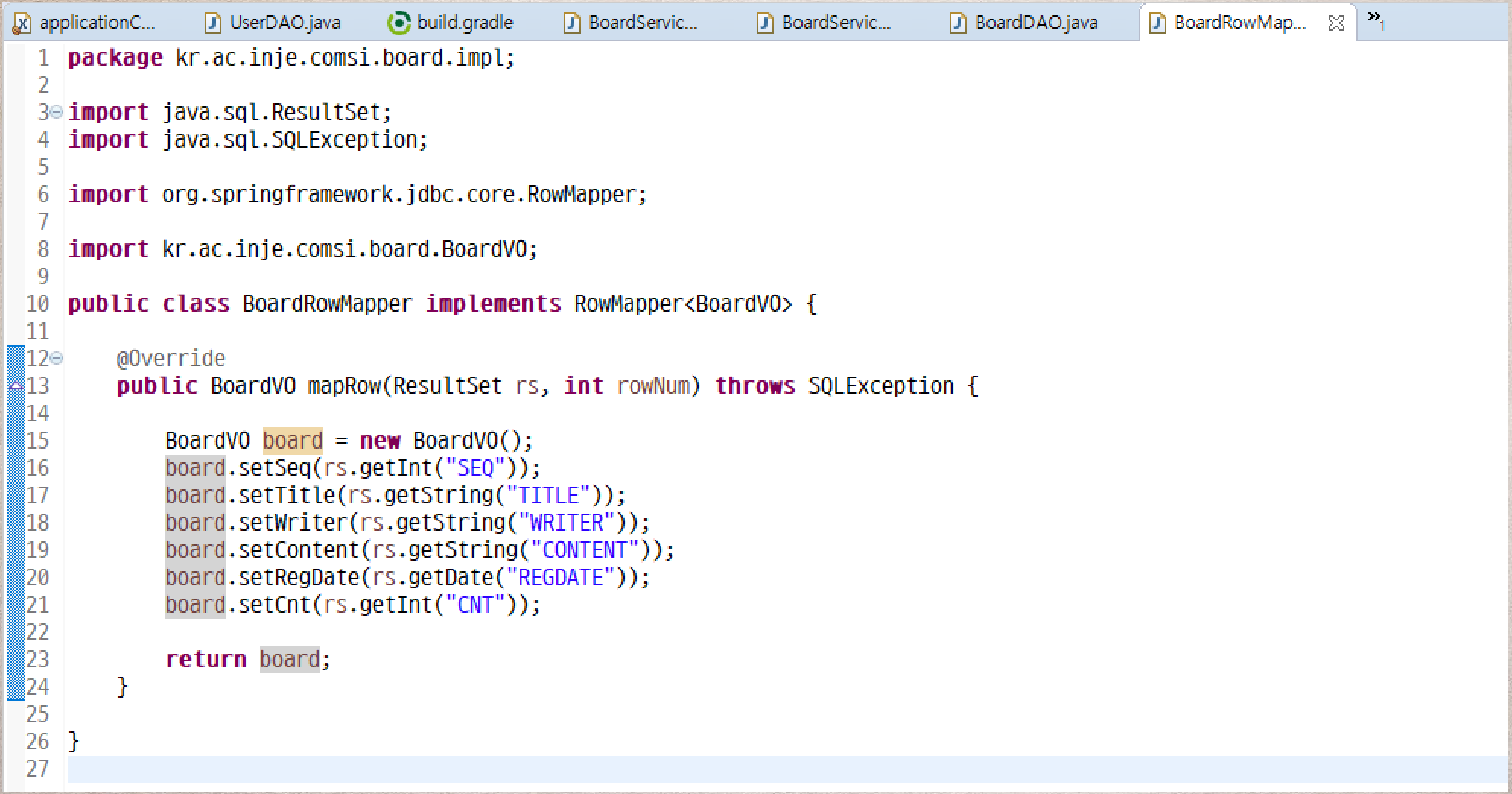


```java
package kr.ac.inje.comsi.board.impl;

import org.springframework.jdbc.core.RowMapper;

public class BoardRowMapper implements RowMapper<T> {

}
```



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

import java.sql.ResultSet;
import java.sql.SQLException;

import org.springframework.jdbc.core.RowMapper;

public class BoardRowMapper implements RowMapper<BoardVO> {

    @Override
    public BoardVO mapRow(ResultSet rs, int rowNum) throws SQLException {
        // TODO Auto-generated method stub
        return null;
    }

}
```

완성된 RowMapper 클래스 - BoardRowMapper 클래스



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

import java.sql.ResultSet;
import java.sql.SQLException;

import org.springframework.jdbc.core.RowMapper;

import kr.ac.inje.comsi.board.BoardVO;

public class BoardRowMapper implements RowMapper<BoardVO> {

    @Override
    public BoardVO mapRow(ResultSet rs, int rowNum) throws SQLException {

        BoardVO board = new BoardVO();
        board.setSeq(rs.getInt("SEQ"));
        board.setTitle(rs.getString("TITLE"));
        board.setWriter(rs.getString("WRITER"));
        board.setContent(rs.getString("CONTENT"));
        board.setRegDate(rs.getDate("REGDATE"));
        board.setCnt(rs.getInt("CNT"));

        return board;
    }

}
```

- query 메소드
  - queryForObject()가 SELECT문으로 객체 하나를 검색할 때 사용하는 메소드라면 query() 메소드는 SELECT문의 실행 결과가 목록일 때 사용한다.

메소드	int query(String sql) int query(String sql, RowMapper<T> rowMapper) int query(String sql, Object[] args, RowMapper<T> rowMapper)

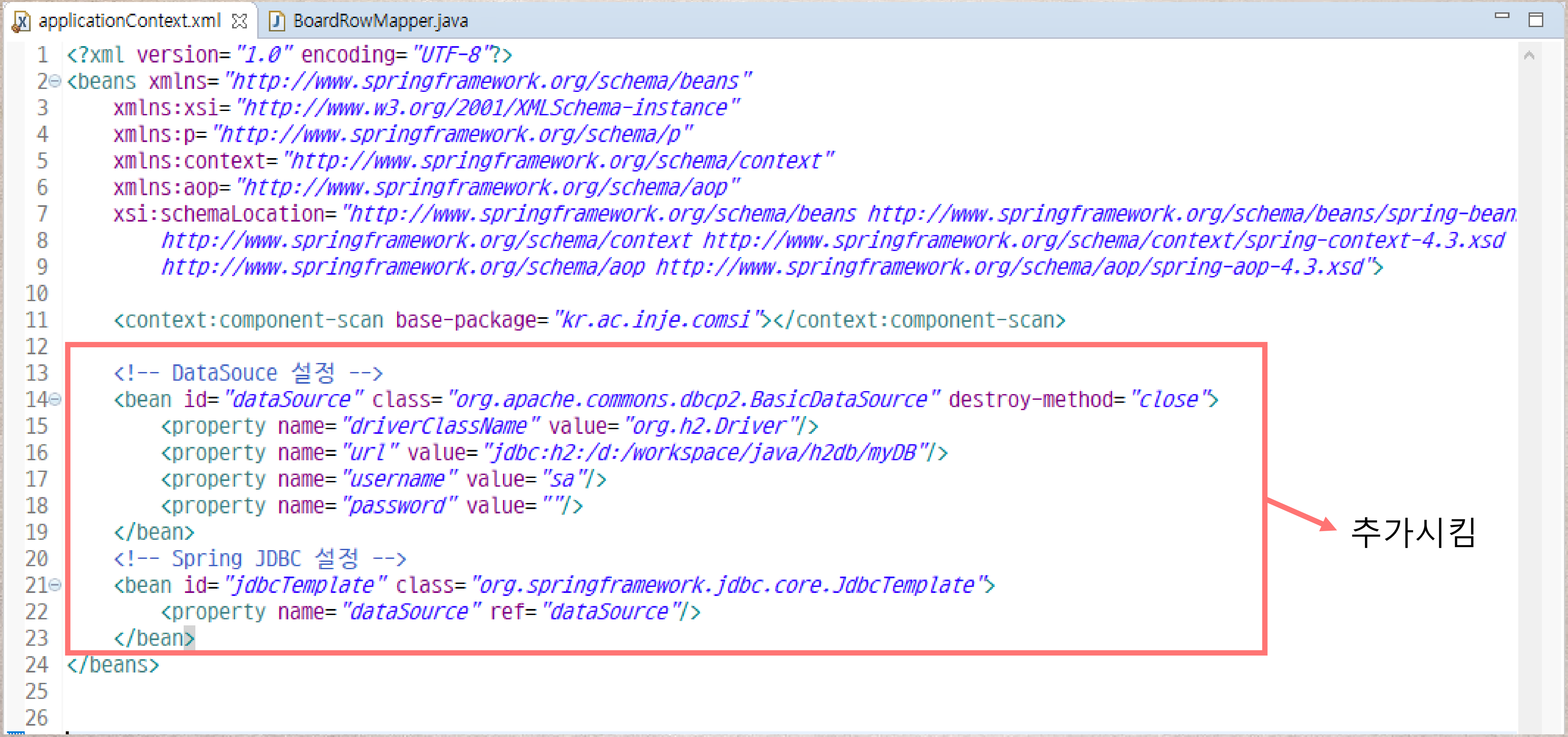
사용 예	`// 글 상세 조회`
	`public List<BoardVO> getBoardList(BoardVO vo) {`
	`    String BOARD_LIST = “ select * from board order by seq desc ”;`
	`    return jdbcTemplate.query(BOARD_LIST, args, new BoardRowMapper());`
	`}`

  - query() 메소드가 실행되면 여러 건의 row 정보가 검색되며,row 수 만큼 RowMapper 객체의 mapRow() 메소드가 실행
  - row 정보가 매핑된 VO 객체 여러 개가 List 컬렉션에 저장되어 리턴된다.

JdbcTemplate를 이용한 DAO

- 스프링 JDBC 사용을 위한 설정이 마무리되었고
- JdbcTemplate를 이용한 DAO 클래스 구현(2가지 방법)
  - jdbcDaoSupport 클래스를 상속
  - **jdbcTemplate 클래스를 <bean> 등록, 의존성 주입**

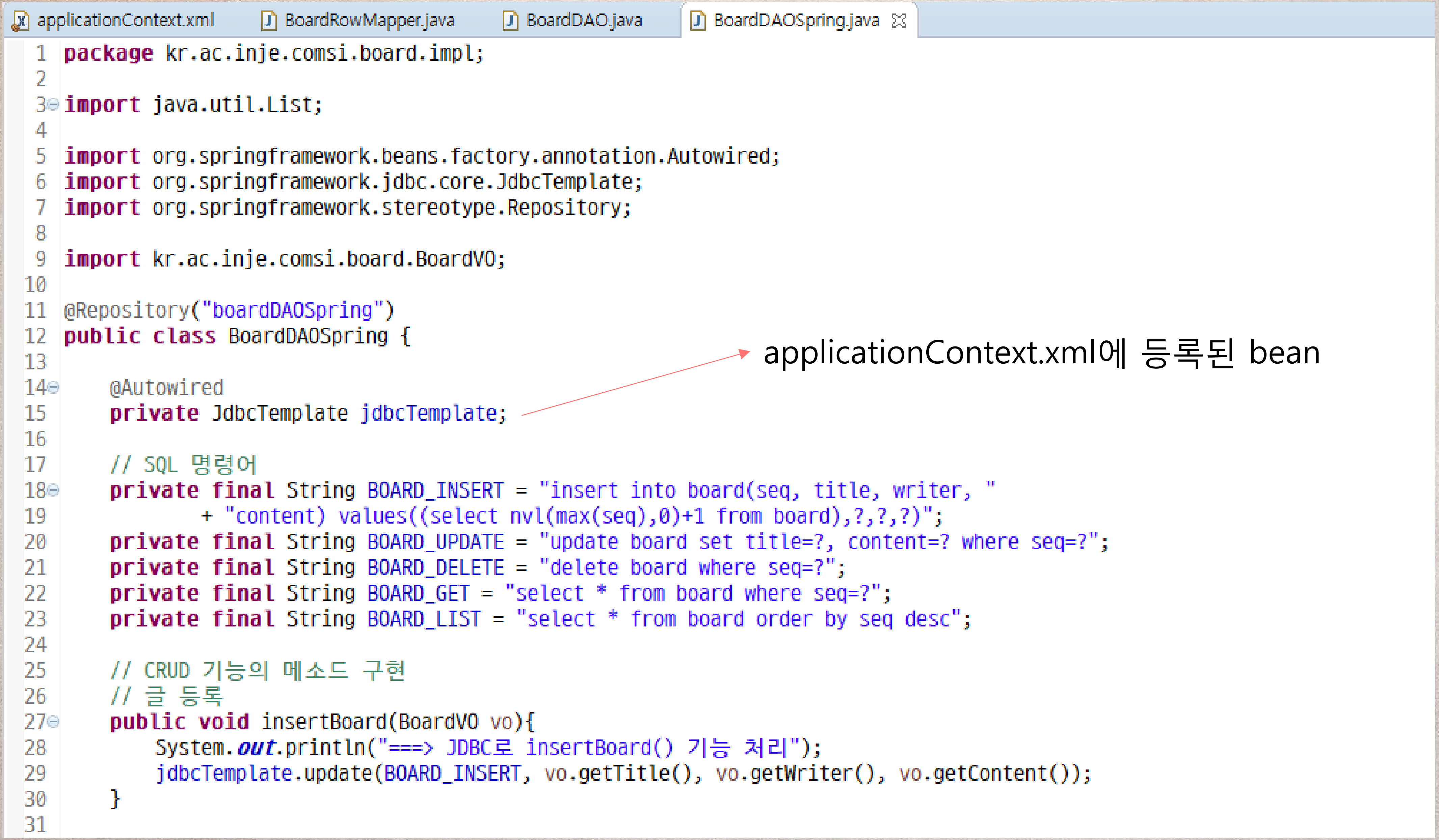
JdbcTemplate 클래스를 <bean> 등록 및 의존성 주입



```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-bean
        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>

    <!-- DataSouce 설정 -->
    <bean id="dataSource" class="org.apache.commons.dbcp2.BasicDataSource" destroy-method="close">
        <property name="driverClassName" value="org.h2.Driver"/>
        <property name="url" value="jdbc:h2:/d:/workspace/java/h2db/myDB"/>
        <property name="username" value="sa"/>
        <property name="password" value=""/>
    </bean>
    <!-- Spring JDBC 설정 -->
    <bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
        <property name="dataSource" ref="dataSource"/>
    </bean>
</beans>
```

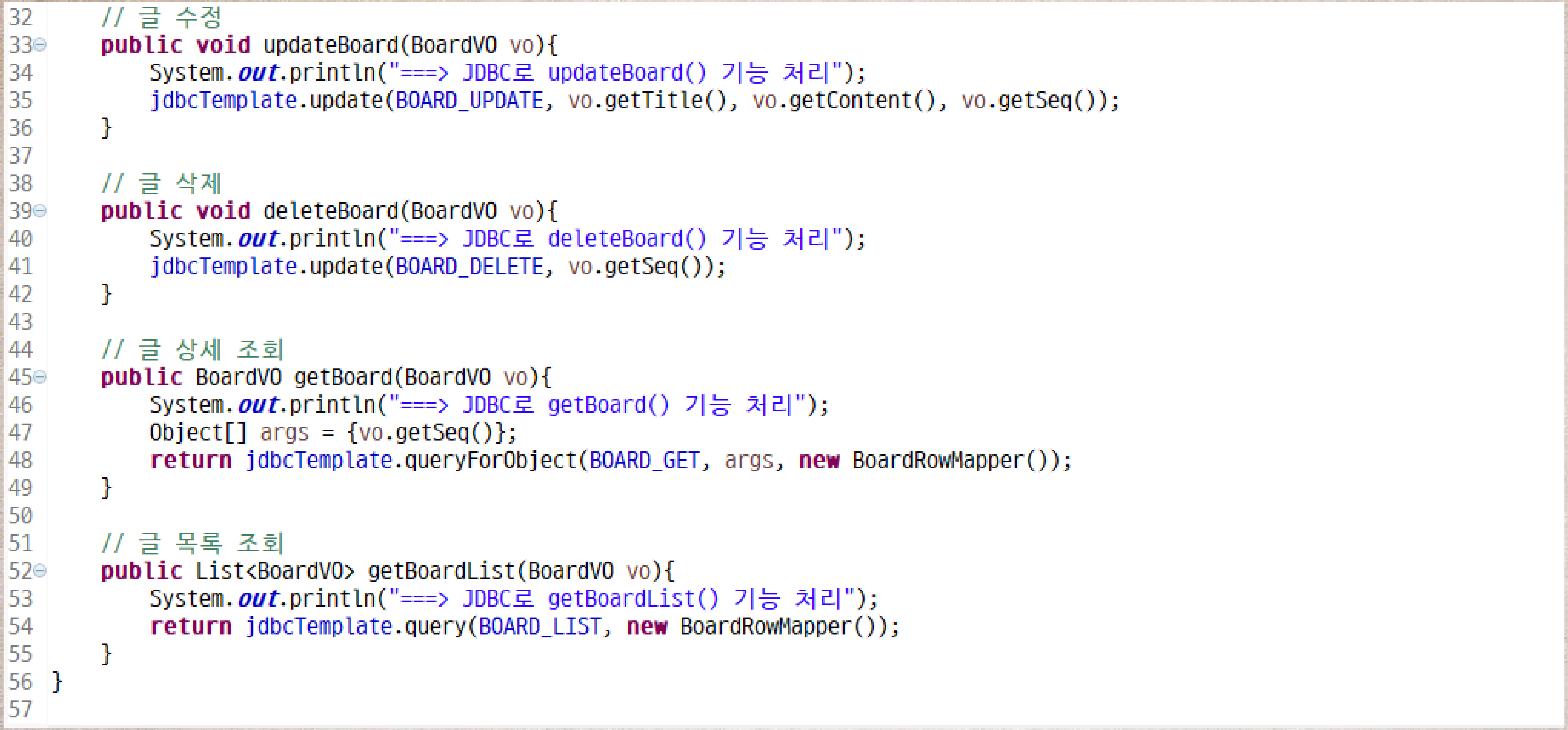
추가시킴

BoardDAOSpring 클래스 생성



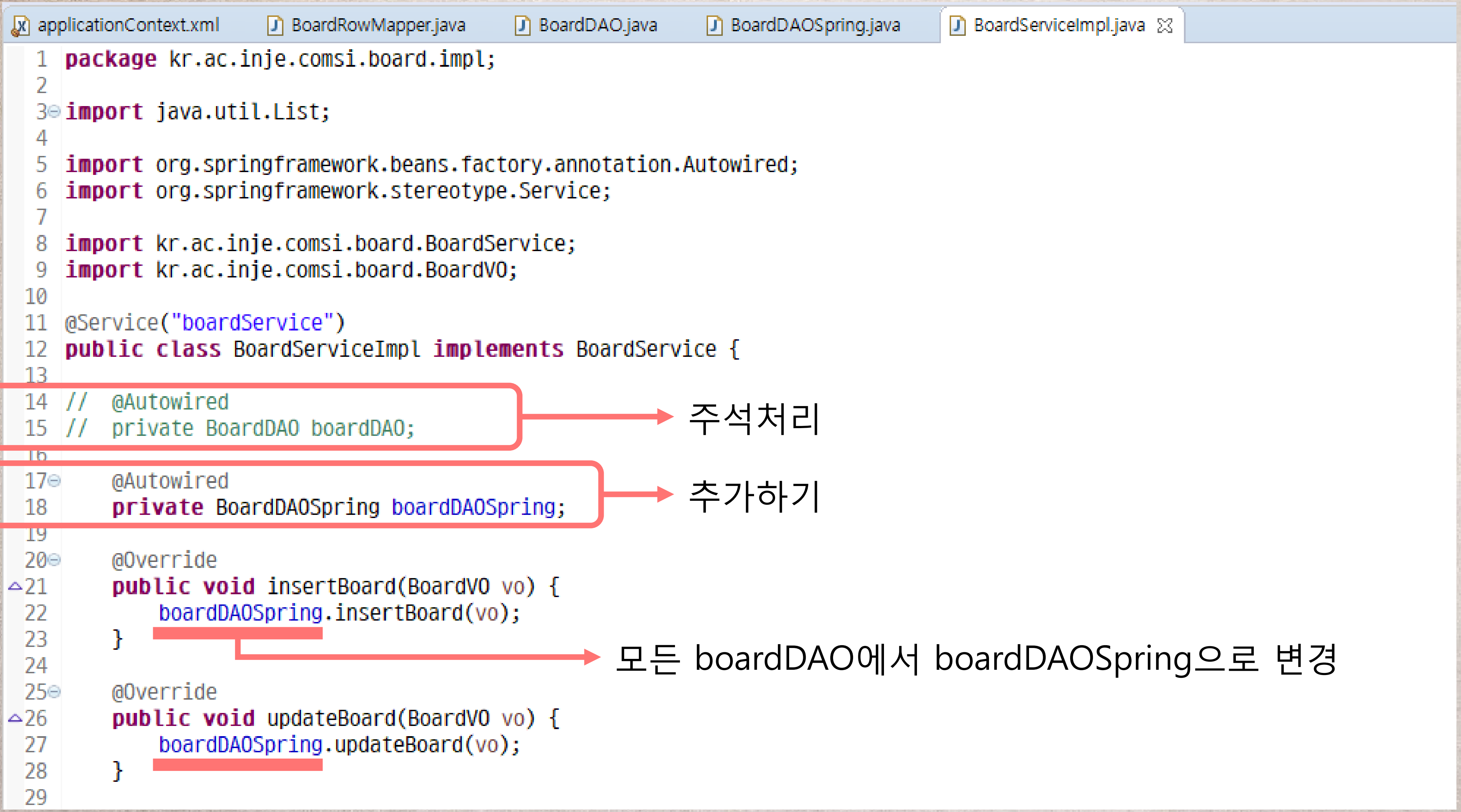
```java
package kr.ac.inje.comsi.board.impl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Repository;

import kr.ac.inje.comsi.board.BoardVO;

@Repository("boardDAOSpring")
public class BoardDAOSpring {

    @Autowired
    private JdbcTemplate jdbcTemplate;

    // SQL 명령어
    private final String BOARD_INSERT = "insert into board(seq, title, writer, "
            + "content) values((select nvl(max(seq),0)+1 from board),?,?,?)";
    private final String BOARD_UPDATE = "update board set title=?, content=? where seq=?";
    private final String BOARD_DELETE = "delete board where seq=?";
    private final String BOARD_GET = "select * from board where seq=?";
    private final String BOARD_LIST = "select * from board order by seq desc";

    // CRUD 기능의 메소드 구현
    // 글 등록
    public void insertBoard(BoardVO vo){
        System.out.println("===> JDBC로 insertBoard() 기능 처리");
        jdbcTemplate.update(BOARD_INSERT, vo.getTitle(), vo.getWriter(), vo.getContent());
    }

```

applicationContext.xml에 등록된 bean



```java
    // 글 수정
    public void updateBoard(BoardVO vo){
        System.out.println("===> JDBC로 updateBoard() 기능 처리");
        jdbcTemplate.update(BOARD_UPDATE, vo.getTitle(), vo.getContent(), vo.getSeq());
    }

    // 글 삭제
    public void deleteBoard(BoardVO vo){
        System.out.println("===> JDBC로 deleteBoard() 기능 처리");
        jdbcTemplate.update(BOARD_DELETE, vo.getSeq());
    }

    // 글 상세 조회
    public BoardVO getBoard(BoardVO vo){
        System.out.println("===> JDBC로 getBoard() 기능 처리");
        Object[] args = {vo.getSeq()};
        return jdbcTemplate.queryForObject(BOARD_GET, args, new BoardRowMapper());
    }

    // 글 목록 조회
    public List<BoardVO> getBoardList(BoardVO vo){
        System.out.println("===> JDBC로 getBoardList() 기능 처리");
        return jdbcTemplate.query(BOARD_LIST, new BoardRowMapper());
    }
}
```

BoardServiceImpl 클래스 수정

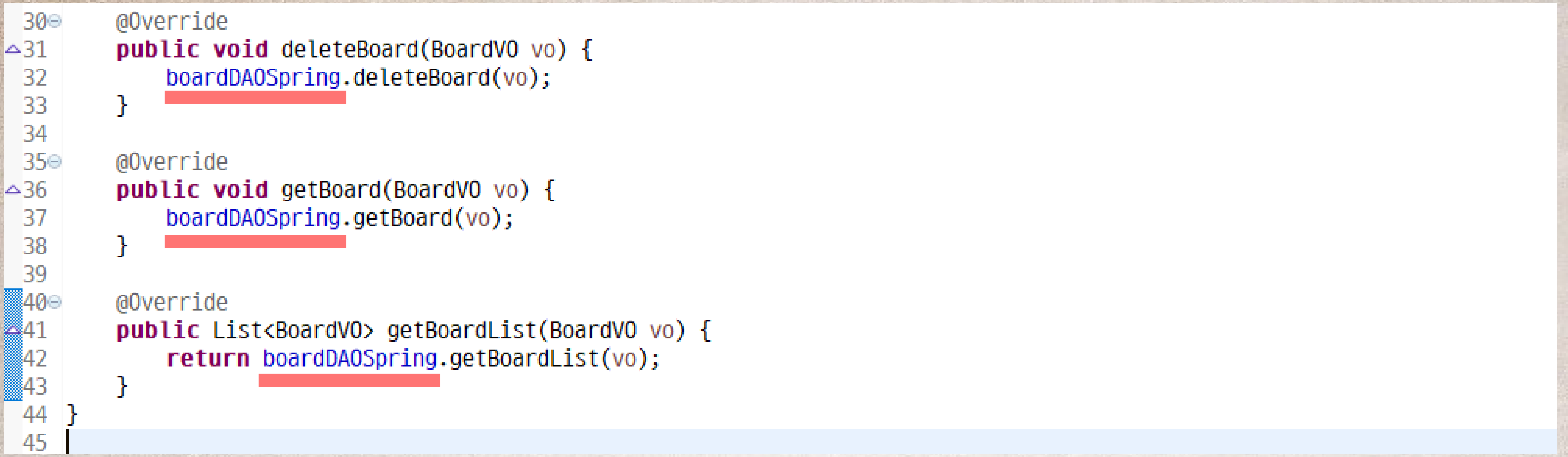


```java
package kr.ac.inje.comsi.board.impl;

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;

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

//	@Autowired
//	private BoardDAO boardDAO;

	@Autowired
	private BoardDAOSpring boardDAOSpring;

	@Override
	public void insertBoard(BoardVO vo) {
		boardDAOSpring.insertBoard(vo);
	}

	@Override
	public void updateBoard(BoardVO vo) {
		boardDAOSpring.updateBoard(vo);
	}

```

주석처리

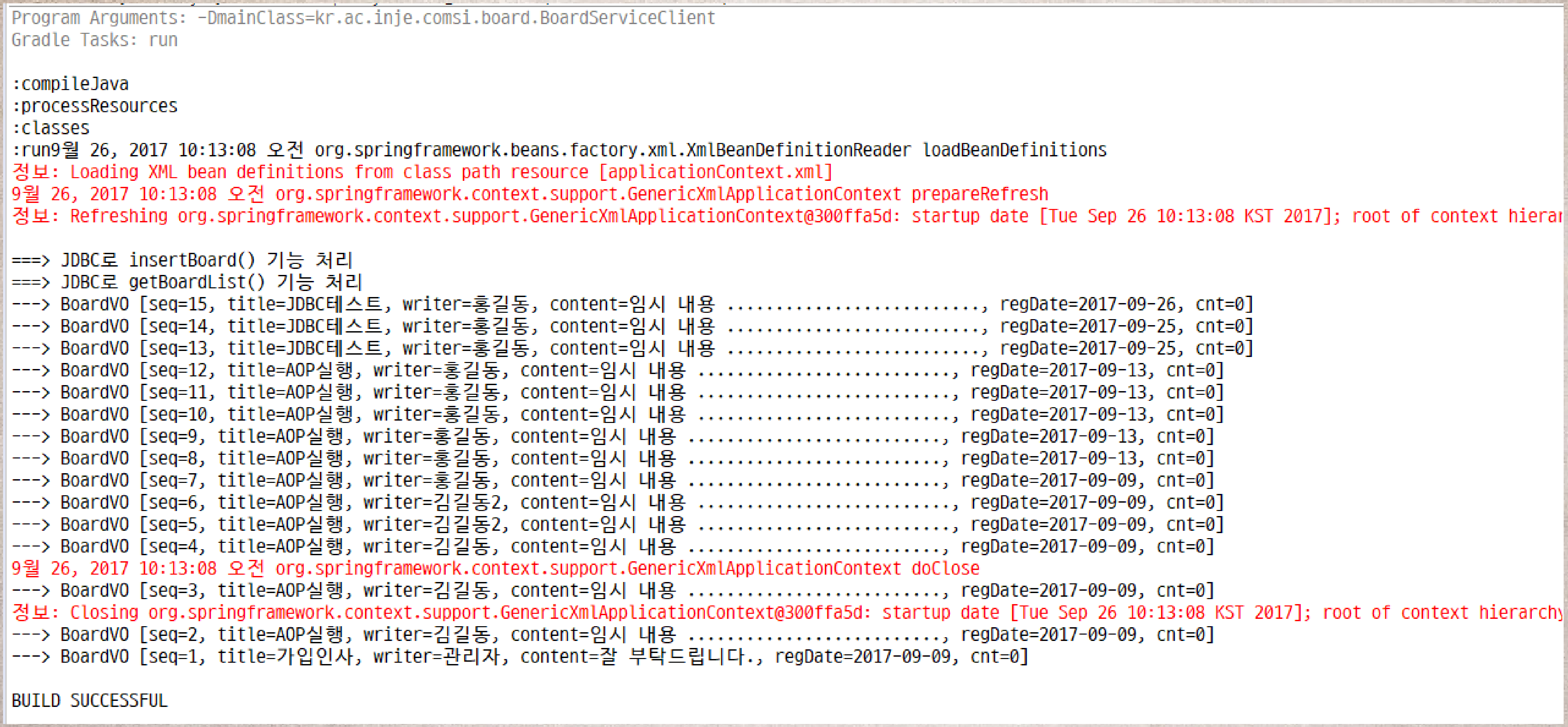
추가하기

모든 boardDAO에서 boardDAOSpring으로 변경



```
    @Override
    public void deleteBoard(BoardVO vo) {
        boardDAOSpring.deleteBoard(vo);
    }

    @Override
    public void getBoard(BoardVO vo) {
        boardDAOSpring.getBoard(vo);
    }

    @Override
    public List<BoardVO> getBoardList(BoardVO vo) {
        return boardDAOSpring.getBoardList(vo);
    }
}
```

실행 결과



```
Program Arguments: -DmainClass=kr.ac.inje.comsi.board.BoardServiceClient
Gradle Tasks: run

:compileJava
:processResources
:classes
:run9월 26, 2017 10:13:08 오전 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
정보: Loading XML bean definitions from class path resource [applicationContext.xml]
9월 26, 2017 10:13:08 오전 org.springframework.context.support.GenericXmlApplicationContext prepareRefresh
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@300ffa5d: startup date [Tue Sep 26 10:13:08 KST 2017]; root of context hierar

===> JDBC로 insertBoard() 기능 처리
===> JDBC로 getBoardList() 기능 처리
---> BoardVO [seq=15, title=JDBC테스트, writer=홍길동, content=임시 내용 ........................., regDate=2017-09-26, cnt=0]
---> BoardVO [seq=14, title=JDBC테스트, writer=홍길동, content=임시 내용 ........................., regDate=2017-09-25, cnt=0]
---> BoardVO [seq=13, title=JDBC테스트, writer=홍길동, content=임시 내용 ........................., regDate=2017-09-25, cnt=0]
---> BoardVO [seq=12, title=AOP실행, writer=홍길동, content=임시 내용 ........................., regDate=2017-09-13, cnt=0]
---> BoardVO [seq=11, title=AOP실행, writer=홍길동, content=임시 내용 ........................., regDate=2017-09-13, cnt=0]
---> BoardVO [seq=10, title=AOP실행, writer=홍길동, content=임시 내용 ........................., regDate=2017-09-13, cnt=0]
---> BoardVO [seq=9, title=AOP실행, writer=홍길동, content=임시 내용 ........................., regDate=2017-09-13, cnt=0]
---> BoardVO [seq=8, title=AOP실행, writer=홍길동, content=임시 내용 ........................., regDate=2017-09-13, cnt=0]
---> BoardVO [seq=7, title=AOP실행, writer=홍길동, content=임시 내용 ........................., regDate=2017-09-09, cnt=0]
---> BoardVO [seq=6, title=AOP실행, writer=김길동2, content=임시 내용 ........................., regDate=2017-09-09, cnt=0]
---> BoardVO [seq=5, title=AOP실행, writer=김길동2, content=임시 내용 ........................., regDate=2017-09-09, cnt=0]
---> BoardVO [seq=4, title=AOP실행, writer=김길동, content=임시 내용 ........................., regDate=2017-09-09, cnt=0]
9월 26, 2017 10:13:08 오전 org.springframework.context.support.GenericXmlApplicationContext doClose
---> BoardVO [seq=3, title=AOP실행, writer=김길동, content=임시 내용 ........................., regDate=2017-09-09, cnt=0]
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@300ffa5d: startup date [Tue Sep 26 10:13:08 KST 2017]; root of context hierarchy
---> BoardVO [seq=2, title=AOP실행, writer=김길동, content=임시 내용 ........................., regDate=2017-09-09, cnt=0]
---> BoardVO [seq=1, title=가입인사, writer=관리자, content=잘 부탁드립니다., regDate=2017-09-09, cnt=0]

BUILD SUCCESSFUL
```