

영속/비즈니스 계층의 CRUD 구현

영속 계층의 작업 순서

테이블의 컬럼 구조를 반영하는 VO(Value Object) 클래스 생성
MyBatis의 Mapper 인터페이스의 작성 처리 및 XML 처리
작성한 Mapper 인터페이스의 테스트

위의 과정 전에

먼저 JDBC 연결을 테스트하는 과정을 반드시 거친다.

1. 영속 계층의 구현 준비

거의 모든 웹 애플리케이션의 최종 목적은

데이터베이스에 데이터를 기록하거나,
원하는 데이터를 가져오는 것이 목적이기 때문에

어느 정도의 설계가 진행되면 데이터베이스 관련 작업을 진행하도록 한다.

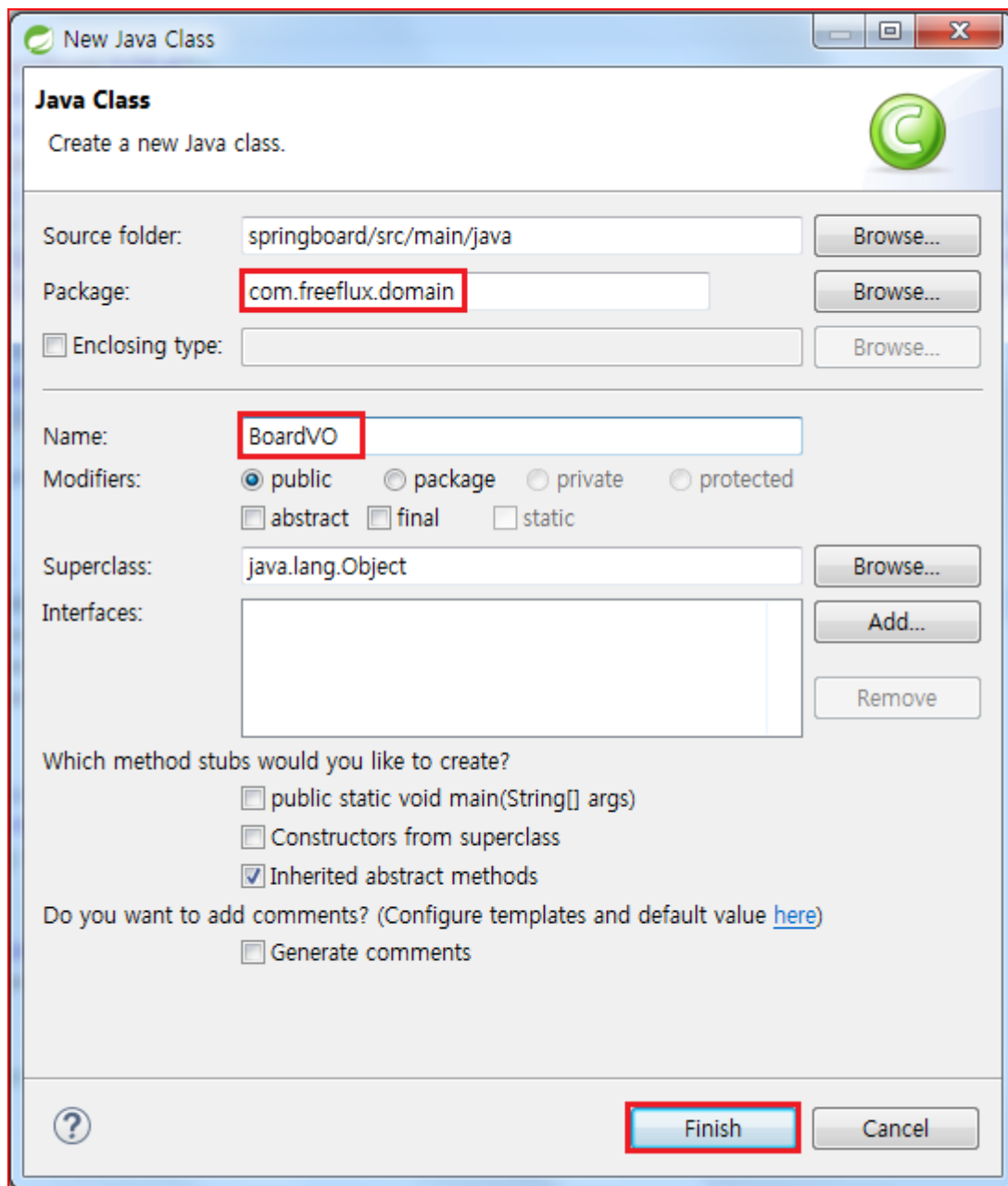
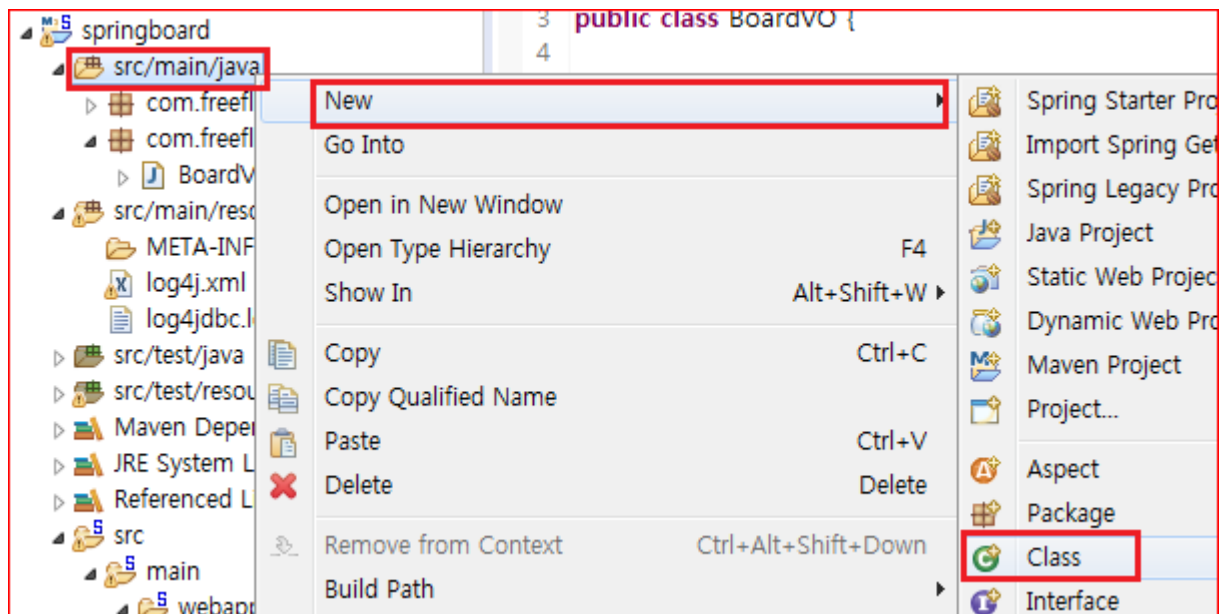
1. VO 클래스 작성

테이블 설계를 기준으로 작성

tbl_board 테이블의 구성.

COLUMN_NAME						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	BNO	NUMBER(10,2)	No	(null)	1	(null)
2	TITLE	VARCHAR2(200 BYTE)	No	(null)	2	(null)
3	CONTENT	VARCHAR2(2000 BYTE)	No	(null)	3	(null)
4	WRITER	VARCHAR2(50 BYTE)	No	(null)	4	(null)
5	REGDATE	DATE	Yes	sysdate	5	(null)
6	UPDATEDATE	DATE	Yes	sysdate	6	(null)

프로젝트에 `org.freeflux.domain` 패키지 생성 후, BoardVO 클래스 생성



BoardVo 클래스 내부 구성

Lombok 을 이용하여

생성자와 getter/setter, toString() 등을 자동으로 만들어 내는 방식을 사용.

따라서, @Data 어노테이션을 적용.

```
BoardVO.java
1 package com.freeflux.domain;
2
3 import java.util.Date;
4
5 import lombok.Data;
6
7 @Data
8 public class BoardVO {
9
10     private Long bno;
11     private String title;
12     private String content;
13     private String writer;
14     private Date regdate;
15     private Date updateDate;
16
17 }
```

2. Mapper 인터페이스의 Mapper XML

MyBatis 는 SQL 을 처리하는데, 어노테이션이나 XML 을 이용할 수 있다.

간단한 SQL 일 경우 : 어노테이션을 이용하여 처리.

복잡하거나 검색 SQL : XML 로 처리.

XML 의 경우

단순 텍스트를 수정하는 과정만으로 처리가 끝나지만,

어노테이션의 경우

코드를 수정하고, 다시 빌드하는 등, 유지 보수성이 떨어진다.

Mapper 인터페이스

root-context.xml 문서 내에

com.freeflux.mapper 패키지를 스캔(조사)하도록 이미 설정되어 있으므로 이를 활용.

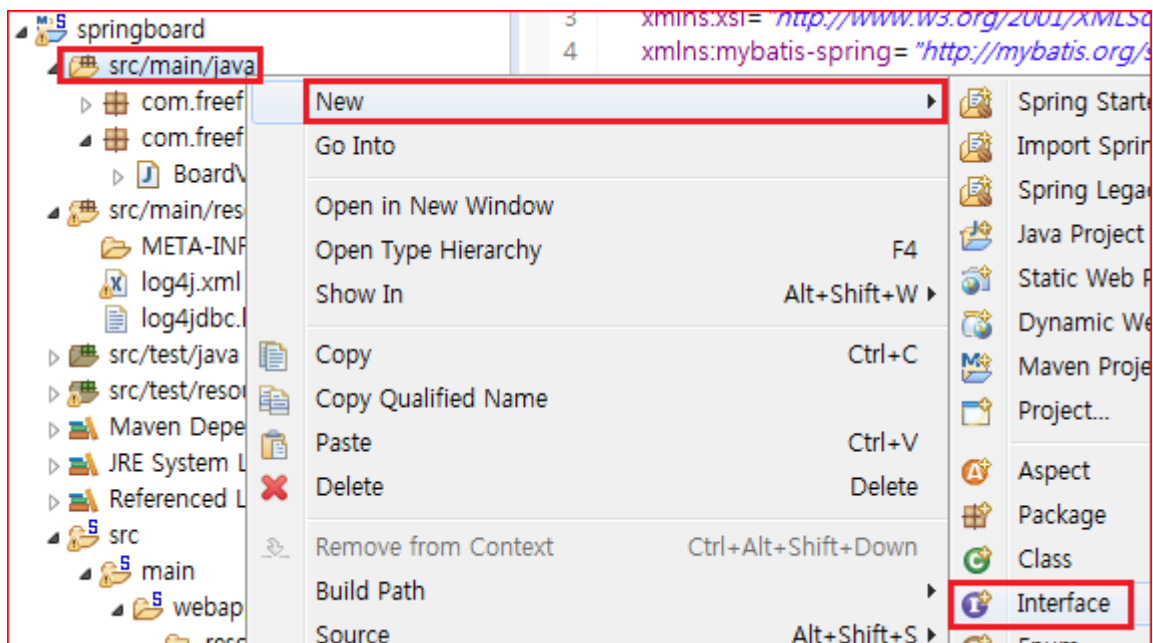
```
<mybatis-spring:scan base-package="com.freeflux.mapper" />
```

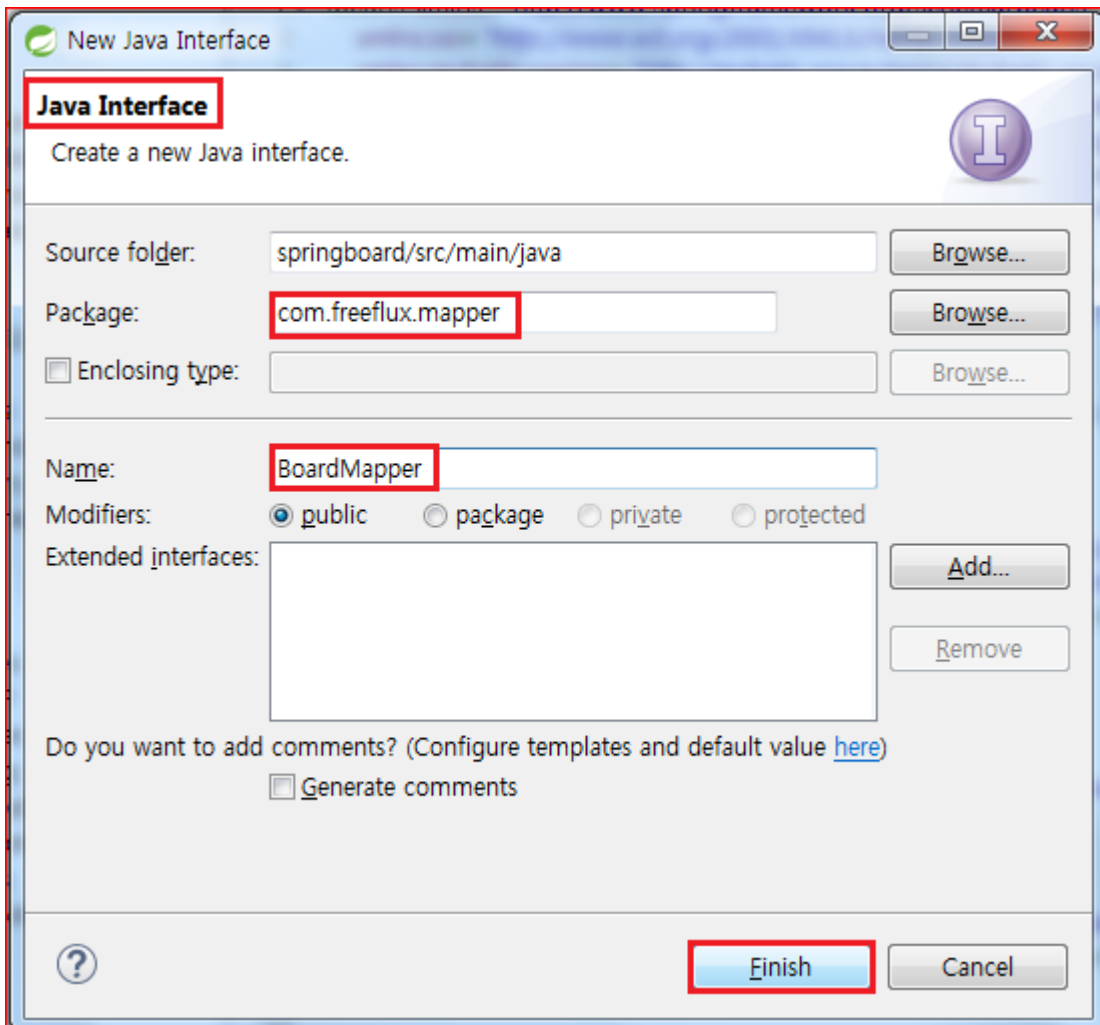
Mapper 인터페이스 작성 시,

리스트(select)와 등록(insert) 작업을 먼저 작성.

org.freeflux.mapper 패키지를 생성하고,

BoardMapper 인터페이스를 추가.





BoardMapper 인터페이스 작성 시,
필요한 SQL 을 어노테이션의 속성값으로 처리할 수 있다.
주의사항 : SQL 작성할 때 ‘;’ 이 없도록 작성해야 한다.
select 문 뒤에 bno 컬럼 조건을 주어서 Primary key(PK)를 이용하도록 한다.

SQL Developer 에서 먼저
SQL 이 문제 없이 실행 가능한 지를 확인.
데이터베이스의 commit 여부를 확인.

```
BoardMapper.java
1 package com.freeflux.mapper;
2
3 import java.util.List;
4
5 import org.apache.ibatis.annotations.Select;
6 import com.freeflux.domain.BoardVO;
7
8 public interface BoardMapper {
9
10     @Select("select * from tbl_board where bno > 0")
11     public List<BoardVO> getList();
12 }
```

select * from tbl_board where bno > 0;						
질의 결과 x						
SQL 인출된 모든 행: 5(0.194초)						
BNO	TITLE	CONTENT	WRITER	REGDATE	UPDATEDATE	
1	1 테스트제목	테스트 내용	user00	19/09/09	19/09/09	
2	2 테스트제목1	테스트 내용1	user01	19/09/09	19/09/09	
3	3 테스트제목2	테스트 내용2	user02	19/09/09	19/09/09	
4	4 테스트제목3	테스트 내용3	user03	19/09/09	19/09/09	
5	5 테스트제목4	테스트 내용4	user04	19/09/09	19/09/09	

테스트 코드(BoardMapperTests.java) 작성

springboard

src/main/java

com.freeflux.controller

com.freeflux.domain

BoardVO.java

com.freeflux.mapper

BoardMapper.java

src/main/resources

META-INF

log4j.xml

log4jdbc.log4j2.properties

src/test/java

com.freeflux.control

com.freeflux.mapper

BoardMapperTest

com.freeflux.persiste

DataSourceTests.j

JDBCTests.java

src/test/resources

Maven Dependencies

JRE System Library [Jav

Referenced Libraries

src

main

webapp

resources

Console

Progress

Problems

<terminated> BoardMapperTests [JUnit] C:\Program Files\Java\

|bno |title |content |writer |regdate |updatedate

|-----|-----|-----|-----|-----|-----|

1 | 테스트제목 | 테스트 내용 | user00 | 2019-09-09 18:05:

2 | 테스트제목1 | 테스트 내용1 | user01 | 2019-09-09 18:07

3 | 테스트제목2 | 테스트 내용2 | user02 | 2019-09-09 18:07

4 | 테스트제목3 | 테스트 내용3 | user03 | 2019-09-09 18:07

5 | 테스트제목4 | 테스트 내용4 | user04 | 2019-09-09 18:07

New

Go Into

Open in New Window

Open Type Hierarchy F4

Show In Alt+Shift+W

Copy Ctrl+C

Copy Qualified Name

Paste Ctrl+V

Delete Delete

Remove from Context Ctrl+Alt+Shift+Down

Build Path

Spring Start

Import Spr

Spring Leg

Java Project

Static Web

Dynamic V

Maven Pro

Project...

Aspect

Package

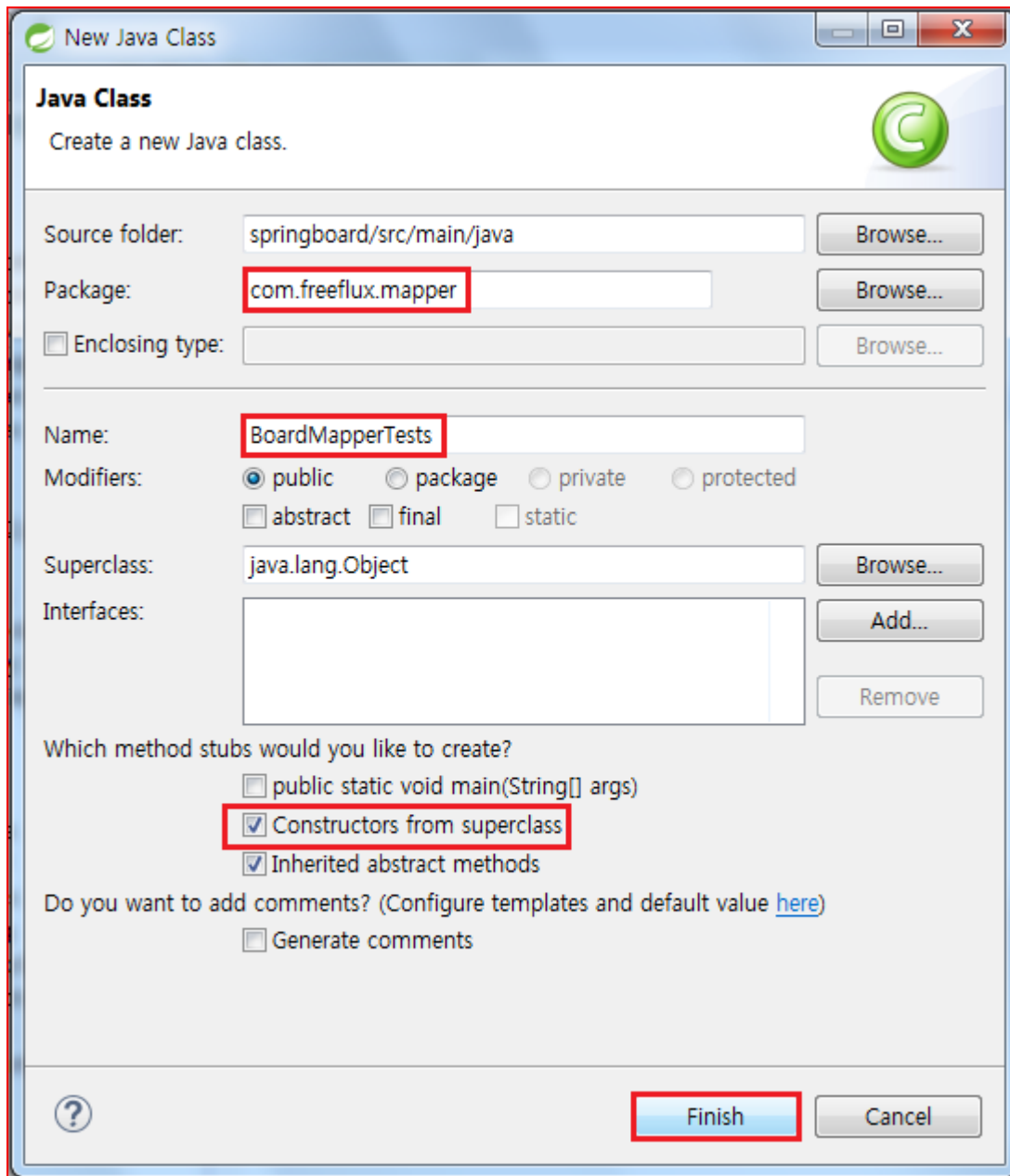
Class

Interface

BoardMapperTests 클래스는

스프링을 이용하여

BoardMapper 인터페이스의 구현체를 주입 받아서 동작하게 한다.



필요한 클래스 import

```
import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

import lombok.Setter;
import lombok.extern.log4j.Log4j;
```

테스트 및 주입을 위한 어노테이션

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration("file:src/main/webapp/WEB-INF/spring/root-context.xml")
@Log4j
public class BoardMapperTests {
```

전체 코드

```
BoardMapperTests.java
1 package com.freeflux.mapper;
2
3 import org.junit.Test;
4 import org.junit.runner.RunWith;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.test.context.ContextConfiguration;
7 import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
8
9 import lombok.Setter;
10 import lombok.extern.log4j.Log4j;
11
12 @RunWith(SpringJUnit4ClassRunner.class)
13 @ContextConfiguration("file:src/main/webapp/WEB-INF/spring/root-context.xml")
14 @Log4j
15 public class BoardMapperTests {
16
17     @Setter(onMethod_ = @Autowired)
18     BoardMapper mapper;
19
20     @Test
21     public void testGetList() {
22         mapper.getList().forEach(board -> log.info(board));
23     }
24 }
```

테스트

The screenshot shows an IDE interface with a project structure on the left and a context menu open over a file named `BoardMapperTests.java`.

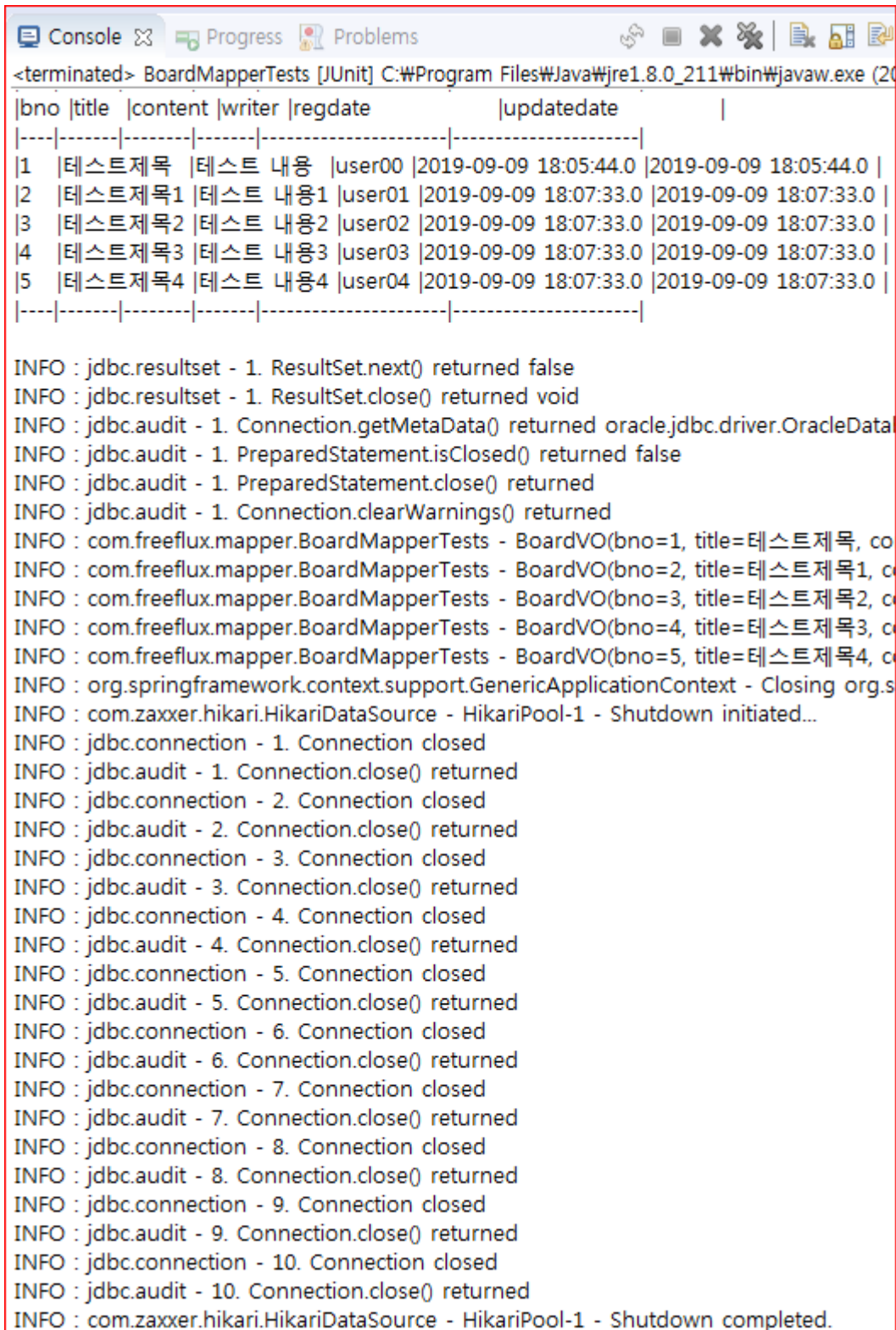
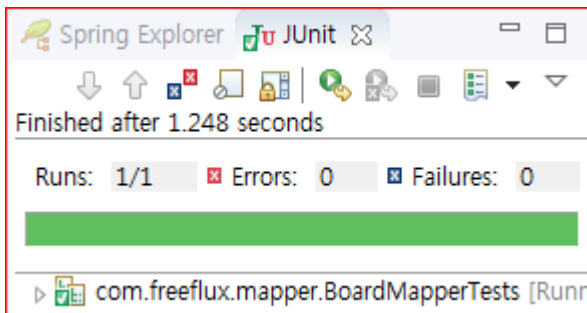
Project Structure (Left Panel):

- `src/test/java`
 - `com.freeflux.controller`
 - `com.freeflux.mapper` (selected)
 - `BoardMapperTests.java` (selected)
 - `com.freeflux.persistence`
 - `DataSourceTests.java`
 - `JDBCTests.java`
- `src/test/resources`
- `Maven Dependencies`
- `JRE System Library [JavaSE-1...]`
- `Referenced Libraries`
- `src`
 - `main`
 - `webapp`
 - `resources`
 - `WEB-INF`
 - `classes`
 - `spring`
 - `appServlet`
 - `root-context.xml`
 - `views`
 - `web.xml`
 - `test`
 - `target`
 - `ora_book.sql`

- `Servers` (with `Boot Dashboard`)

Context Menu (Right Panel):

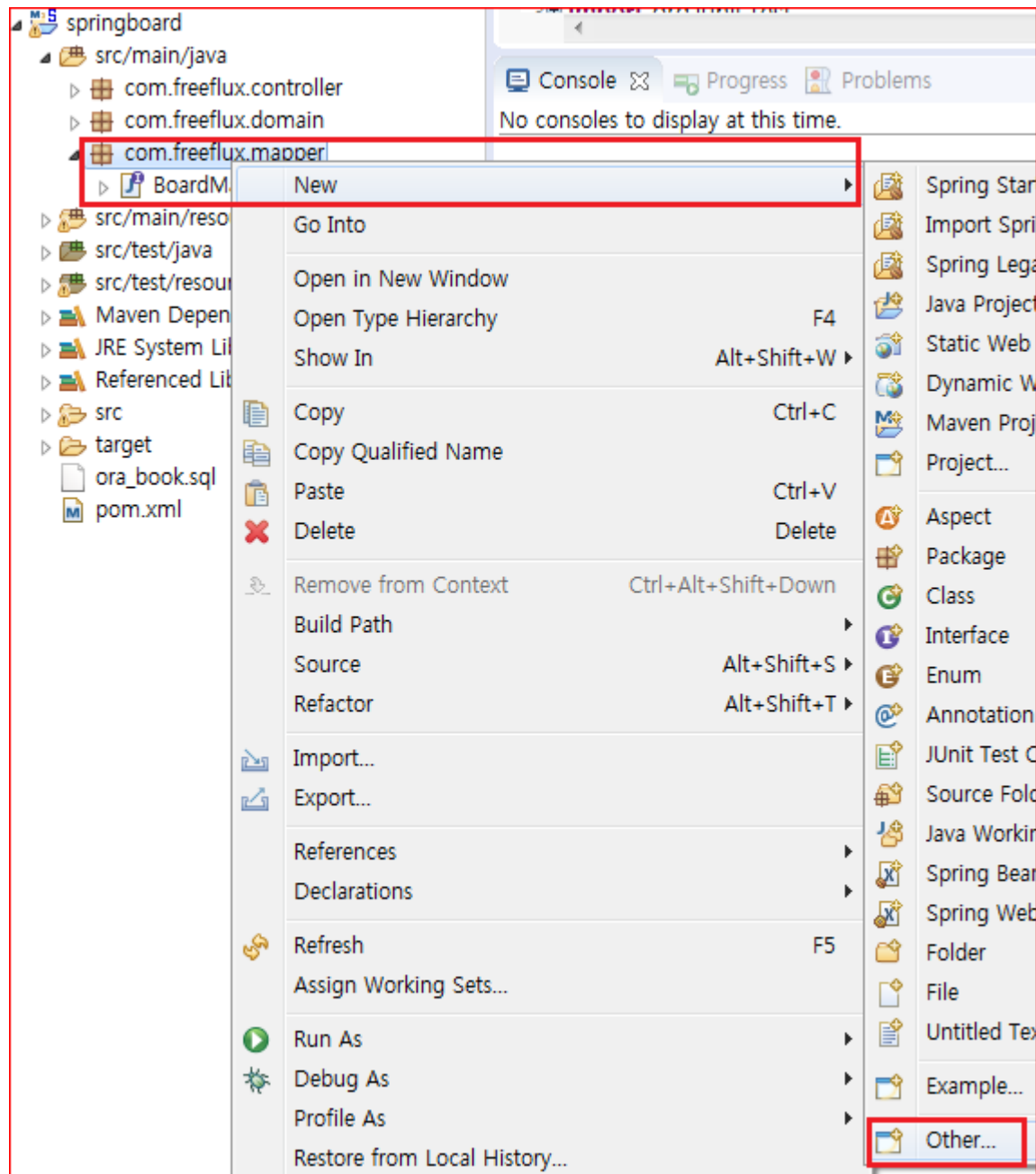
- Open with...
- Open Type Hierarchy F4
- Show In Alt+Shift+W
- Copy Ctrl+C
- Copy Qualified Name
- Paste Ctrl+V
- Delete Delete
- Remove from Context Ctrl+Alt+Shift+Down
- Build Path
- Source Alt+Shift+S
- Refactor Alt+Shift+T
- Import...
- Export...
- References
- Declarations
- Refresh F5
- Assign Working Sets...
- Recreate Test Suite...
- Run As** (highlighted)
 - 1 Run on Server Alt+Shift+X, R
 - 2 JUnit Test** Alt+Shift+X, T
- Debug As
- Profile As

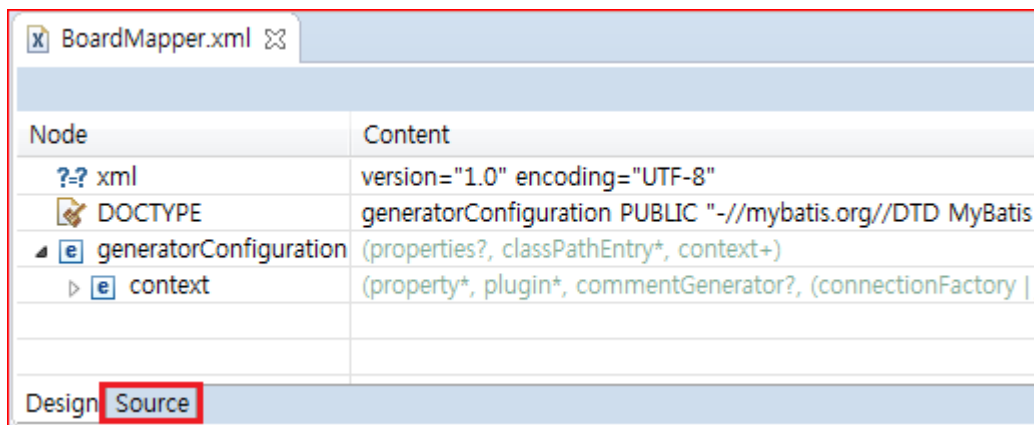
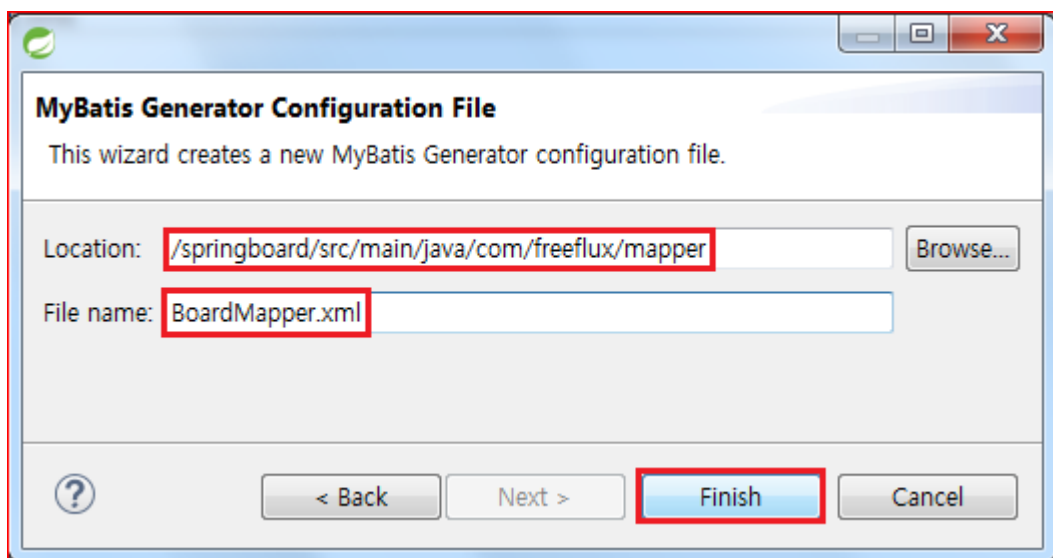
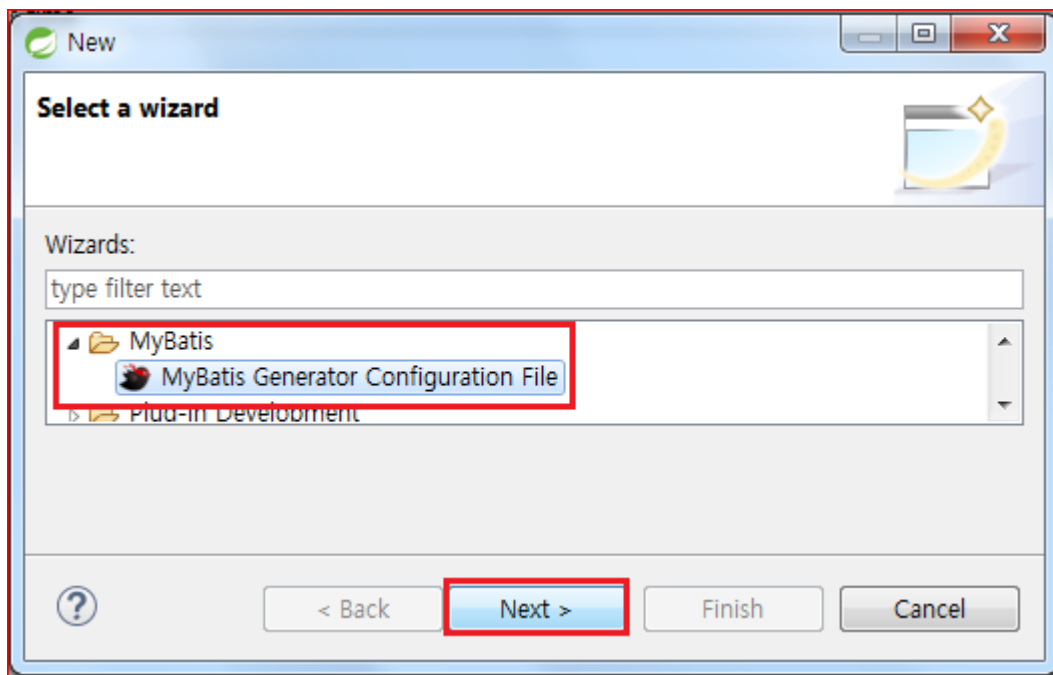


Mapper XML 문서

참고 : 파일의 폴더 구조나 이름은 무방하지만,
패키지와 클래스 이름과 동일하게 해주면 추후 혼란스러운 상황이 발생.

BoardMapper.xml





기본 코드

```
BoardMapper.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE generatorConfiguration
3 PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"
4 "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">
5
6 <generatorConfiguration>
7   <context id="context1">
8     <jdbcConnection connectionURL="???" driverClass="???" password="???" userId="???" />
9     <javaModelGenerator targetPackage="???" targetProject="???" />
10    <sqlMapGenerator targetPackage="???" targetProject="???" />
11    <javaClientGenerator targetPackage="???" targetProject="???" type="XMLMAPPER" />
12    <table schema="???" tableName="???">
13      <columnOverride column="???" property="???" />
14    </table>
15  </context>
16 </generatorConfiguration>
```

선언문 변경

```
BoardMapper.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE mapper
3 PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
4 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
5
```

불필요한 설정 제거 또는 주석 처리

```
<!--
<generatorConfiguration>
  <context id="context1">
    <jdbcConnection connectionURL="???" driverClass="???" password="???" userId="???" />
    <javaModelGenerator targetPackage="???" targetProject="???" />
    <sqlMapGenerator targetPackage="???" targetProject="???" />
    <javaClientGenerator targetPackage="???" targetProject="???" type="XMLMAPPER" />
    <table schema="???" tableName="???">
      <columnOverride column="???" property="???" />
    </table>
  </context>
</generatorConfiguration>
-->
```

XML 을 작성할 때 주의 사항

<mapper>의 namespace 속성값 : Mapper 인터페이스와 동일한 이름

<select> 태그의 id 속성값 : 메서드 이름과 일치.

resultType 속성값 : select 쿼리의 결과를 특정 클래스의 객체로 자동 생성하기 위한 설정.

```

BoardMapper.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE mapper 한칸 띄어쓰기
3 PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" 한칸 띄어쓰기
4 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
5
6 <mapper namespace="com.freeflux.mapper.BoardMapper">
7
8 <select id="getList" resultType="com.freeflux.domain.BoardVO">
9   <![CDATA[
10     select * from tbl_board where bno > 0
11   ]]>
12 </select>
13
14 </mapper>
15
16 <!--
17 <generatorConfiguration>
18 <context id="context1">
19   <jdbcConnection connectionURL="???" driverClass="???" password="???" userId="???" />
20   <javaModelGenerator targetPackage="???" targetProject="???" />
21   <sqlMapGenerator targetPackage="???" targetProject="???" />
22   <javaClientGenerator targetPackage="???" targetProject="???" type="XMLMAPPER" />
23 <table schema="???" tableName="???">
24   <columnOverride column="???" property="???" />
25 </table>
26 </context>
27 </generatorConfiguration>
28 -->

```

* XML 에 사용한 CDATA : XML 에서 부등호(<, >)를 사용하기 위함.

BoardMapper 인터페이스 안의 select SQL 은 제거 !!!

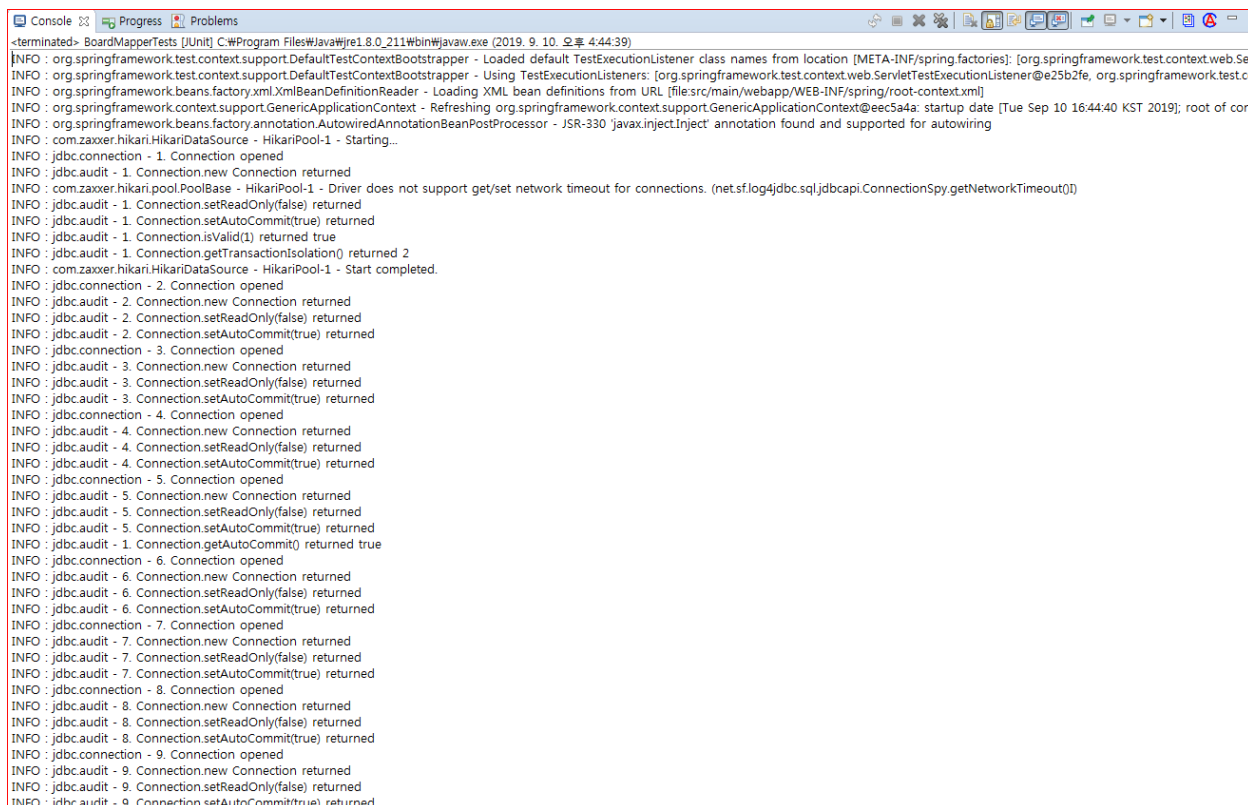
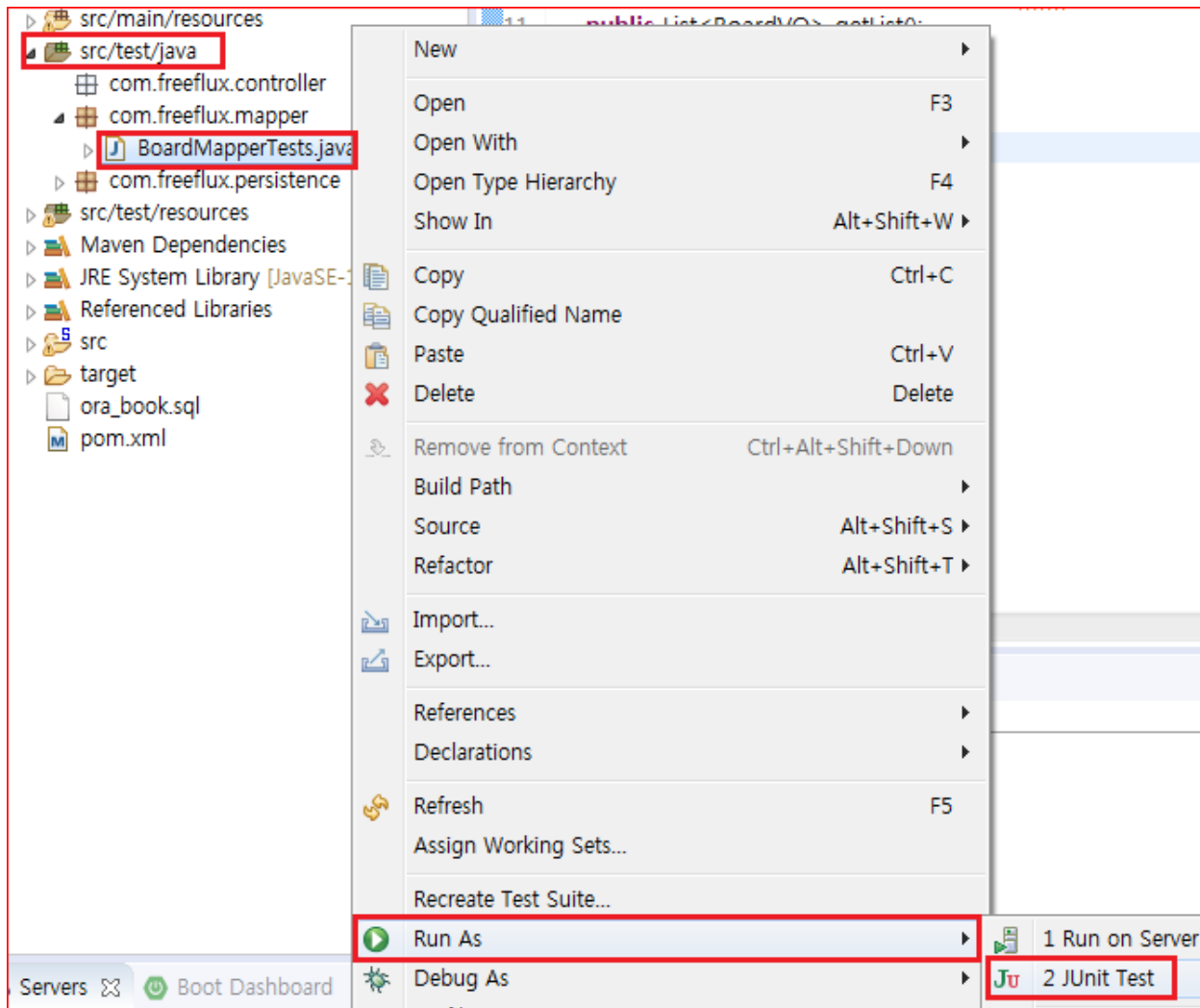
```

BoardMapper.xml BoardMapper.java
1 package com.freeflux.mapper;
2
3 import java.util.List;
4
5 //import org.apache.ibatis.annotations.Select;
6 import com.freeflux.domain.BoardVO;
7
8 public interface BoardMapper {
9
10 // @Select("select * from tbl_board where bno > 0")
11 public List<BoardVO> getList();
12 }

```

인터페이스 수정 후,
반드시 기존의 테스트 코드를 통해서
기존과 동일하게 동작하는 지를 확인해야 한다!!!!

테스트



INFO : jdbc.audit - 1. PreparedStatement.new PreparedStatement returned
INFO : jdbc.audit - 1. Connection.prepareStatement(select * from tbl_board where bno > 0) returned net.sf.log4jdbc.sql.jdbcapi.PreparedStatementSpy@256f8274
INFO : jdbc.sqlonly - select * from tbl_board where bno > 0

INFO : jdbc.connection - 10. Connection opened
INFO : jdbc.audit - 10. Connection.new Connection returned
INFO : jdbc.audit - 10. Connection.setReadOnly(false) returned
INFO : jdbc.audit - 10. Connection.setAutoCommit(true) returned
INFO : jdbc.sqltiming - select * from tbl_board where bno > 0
(executed in 46 msec)
INFO : jdbc.audit - 1. PreparedStatement.execute() returned true
INFO : jdbc.resultset - 1. ResultSet.new ResultSet returned
INFO : jdbc.audit - 1. PreparedStatement.getResultSet() returned net.sf.log4jdbc.sql.jdbcapi.ResultSetSpy@1eb6749b
INFO : jdbc.resultset - 1. ResultSet.getMetaData() returned oracle.jdbc.driver.OracleResultSetMetaData@73db4768
INFO : jdbc.resultset - 1. ResultSet.getType() returned 1003
INFO : jdbc.resultset - 1. ResultSet.next() returned true
INFO : jdbc.resultset - 1. ResultSet.getLong(BNO) returned 1
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(TITLE) returned 테스트제목
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(CONTENT) returned 테스트 내용
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(WRITER) returned user00
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getTimestamp(REGDATE) returned 2019-09-09 18:05:44.0
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getTimestamp(UPDATEDATE) returned 2019-09-09 18:05:44.0
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.next() returned true
INFO : jdbc.resultset - 1. ResultSet.getLong(BNO) returned 2
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(TITLE) returned 테스트제목1
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(CONTENT) returned 테스트 내용1
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(WRITER) returned user01
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getTimestamp(REGDATE) returned 2019-09-09 18:07:33.0
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getTimestamp(UPDATEDATE) returned 2019-09-09 18:07:33.0
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.next() returned true
INFO : jdbc.resultset - 1. ResultSet.getLong(BNO) returned 3
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(TITLE) returned 테스트제목2
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(CONTENT) returned 테스트 내용2
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(WRITER) returned user02

INFO : jdbc.resultset - 1. ResultSet.getString(WRITER) returned user02
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getTimestamp(REGDATE) returned 2019-09-09 18:07:33.0
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getTimestamp(UPDATEDATE) returned 2019-09-09 18:07:33.0
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.next() returned true
INFO : jdbc.resultset - 1. ResultSet.getLong(BNO) returned 4
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(TITLE) returned 테스트제목3
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(CONTENT) returned 테스트 내용3
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(WRITER) returned user03
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getTimestamp(REGDATE) returned 2019-09-09 18:07:33.0
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getTimestamp(UPDATEDATE) returned 2019-09-09 18:07:33.0
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.next() returned true
INFO : jdbc.resultset - 1. ResultSet.getLong(BNO) returned 5
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(TITLE) returned 테스트제목4
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(CONTENT) returned 테스트 내용4
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getString(WRITER) returned user04
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getTimestamp(REGDATE) returned 2019-09-09 18:07:33.0
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultset - 1. ResultSet.getTimestamp(UPDATEDATE) returned 2019-09-09 18:07:33.0
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false

INFO : jdbc.resultsettable -

	bno	title	content	writer	regdate	updatedate
1	테스트제목	테스트 내용	user00	2019-09-09 18:05:44.0	2019-09-09 18:05:44.0	
2	테스트제목1	테스트 내용1	user01	2019-09-09 18:07:33.0	2019-09-09 18:07:33.0	
3	테스트제목2	테스트 내용2	user02	2019-09-09 18:07:33.0	2019-09-09 18:07:33.0	
4	테스트제목3	테스트 내용3	user03	2019-09-09 18:07:33.0	2019-09-09 18:07:33.0	
5	테스트제목4	테스트 내용4	user04	2019-09-09 18:07:33.0	2019-09-09 18:07:33.0	

```
INFO : jdbc.resultset - 1. ResultSet.next() returned false
INFO : jdbc.resultset - 1. ResultSet.close() returned void
INFO : jdbc.audit - 1. Connection.getMetaData() returned oracle.jdbc.driver.OracleDatabaseMetaData@3e34ace1
INFO : jdbc.audit - 1. PreparedStatement.isClosed() returned false
INFO : jdbc.audit - 1. PreparedStatement.close() returned
INFO : jdbc.audit - 1. Connection.clearWarnings() returned
INFO : com.freeflux.mapper.BoardMapperTests - BoardVO(bno=1, title=테스트제목, content=테스트 내용, writer=user00, regdate=Mon Sep 09 18:05:44 KST 2019, updateDate=Mon Sep 09 18:05:44 KST 2019)
INFO : com.freeflux.mapper.BoardMapperTests - BoardVO(bno=2, title=테스트제목1, content=테스트 내용1, writer=user01, regdate=Mon Sep 09 18:07:33 KST 2019, updateDate=Mon Sep 09 18:07:33 KST 2019)
INFO : com.freeflux.mapper.BoardMapperTests - BoardVO(bno=3, title=테스트제목2, content=테스트 내용2, writer=user02, regdate=Mon Sep 09 18:07:33 KST 2019, updateDate=Mon Sep 09 18:07:33 KST 2019)
INFO : com.freeflux.mapper.BoardMapperTests - BoardVO(bno=4, title=테스트제목3, content=테스트 내용3, writer=user03, regdate=Mon Sep 09 18:07:33 KST 2019, updateDate=Mon Sep 09 18:07:33 KST 2019)
INFO : com.freeflux.mapper.BoardMapperTests - BoardVO(bno=5, title=테스트제목4, content=테스트 내용4, writer=user04, regdate=Mon Sep 09 18:07:33 KST 2019, updateDate=Mon Sep 09 18:07:33 KST 2019)
INFO : org.springframework.context.support.GenericApplicationContext - Closing org.springframework.context.support.GenericApplicationContext@eec5a4a: startup date [Tue Sep 10 16:44:40 KST 2019]; root of context
INFO : com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Shutdown initiated...
INFO : jdbc.connection - 1. Connection closed
INFO : jdbc.audit - 1. Connection.close() returned
INFO : jdbc.connection - 2. Connection closed
INFO : jdbc.audit - 2. Connection.close() returned
INFO : jdbc.connection - 3. Connection closed
INFO : jdbc.audit - 3. Connection.close() returned
INFO : jdbc.connection - 4. Connection closed
INFO : jdbc.audit - 4. Connection.close() returned
INFO : jdbc.connection - 5. Connection closed
INFO : jdbc.audit - 5. Connection.close() returned
INFO : jdbc.connection - 6. Connection closed
INFO : jdbc.audit - 6. Connection.close() returned
INFO : jdbc.connection - 7. Connection closed
INFO : jdbc.audit - 7. Connection.close() returned
INFO : jdbc.connection - 8. Connection closed
INFO : jdbc.audit - 8. Connection.close() returned
INFO : jdbc.connection - 9. Connection closed
INFO : jdbc.audit - 9. Connection.close() returned
INFO : jdbc.connection - 10. Connection closed
INFO : jdbc.audit - 10. Connection.close() returned
INFO : com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Shutdown completed.
```


2. 영속 영역의 CRUD 구현

웹 프로젝트 구조에서 마지막 영역이 영속 영역이지만, 실제로 구현을 가장 먼저 할 수 있는 영역이기도 하다.

영속 영역은 기본적으로 CRUD 작업을 하기 때문에 테이블과 VO(DTO) 등 약간의 준비만으로도 비즈니스 로직과 무관하게 CRUD 작업을 작성할 수 있다.

MyBatis 는 내부적으로

JDBC 의 PreparedStatement 를 활용하고

필요한 파라미터를 처리하는 ' ? '에 대한 치환은 ' #{속성} '을 이용하여 처리.

1. create(insert) 처리

tbl_board 테이블은 PK 컬럼으로 bno 을 이용하고,

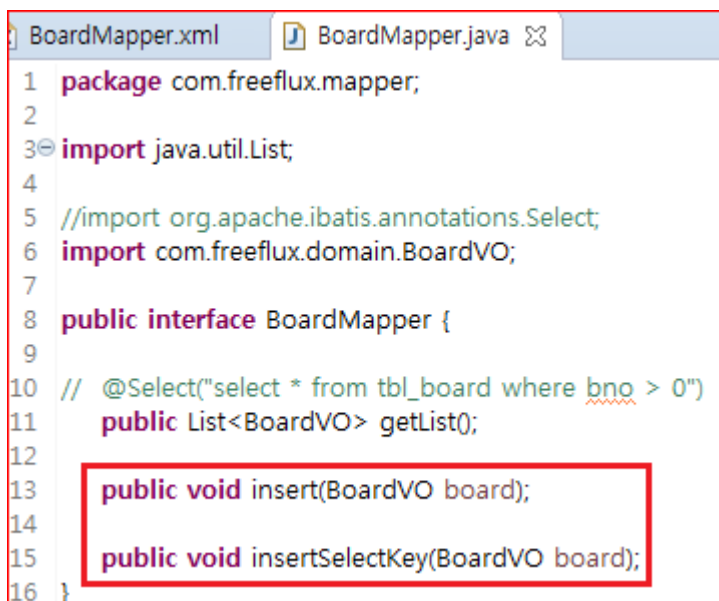
시퀀스(sequence)를 이용하여 데이터가 추가될 때마다 자동으로 번호가 만들어지는 방식을 사용.

이처럼 **자동으로 PK 값이 정해지는 경우 처리 방법**

insert 만 처리되고 생성된 PK 값을 알 필요가 없는 경우

insert 문이 실행되고 생성된 PK 값을 알아야 할 경우

BoardMapper 인터페이스 수정 : 위의 상황을 고려하여 메서드를 추가 선언.



```
BoardMapper.xml | BoardMapper.java
1 package com.freeflux.mapper;
2
3 import java.util.List;
4
5 //import org.apache.ibatis.annotations.Select;
6 import com.freeflux.domain.BoardVO;
7
8 public interface BoardMapper {
9
10 // @Select("select * from tbl_board where bno > 0")
11 public List<BoardVO> getList();
12
13 public void insert(BoardVO board);
14
15 public void insertSelectKey(BoardVO board);
16 }
```

BoardMapper.xml 수정

```
BoardMapper.xml BoardMapper.java
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE mapper
3 PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
4 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
5
6 <mapper namespace="com.freeflux.mapper.BoardMapper">
7
8 <select id="getList" resultType="com.freeflux.domain.BoardVO">
9 <![CDATA[
10 select * from tbl_board where bno > 0
11 ]]>
12 </select>
13
14
15 <insert id="insert">
16 insert into tbl_board (bno,title,content,writer)
17 values (seq_board.nextval, #{title}, #{content}, #{writer})
18 </insert>
19
20 <insert id="insertSelectKey">
21
22 <selectKey keyProperty="bno" order="BEFORE" resultType="long">
23 select seq_board.nextval from dual
24 </selectKey>
25
26 insert into tbl_board (bno,title,content, writer)
27 values (#{bno}, #{title}, #{content}, #{writer})
28 </insert>
29
30 </mapper>
```

BoardMapper

<insert id="insert"> : 단순히 시퀀스의 다음 값을 구해서 insert 할 때 사용

insert into SQL 문 :

몇 건의 데이터가 변경되었는지 만을 알려주기 때문에
추가된 데이터의 PK 값을 알 수는 없지만,
한번의 SQL 처리만으로 작업이 완료된다는 장점이 있다.

<insert id="insertSelectKey"> : @SelectKey 라는 어노테이션을 사용.

@SelectKey :

주로 PK 값을 미이(before) SQL 문을 통하여 처리해두고,
특정한 이름으로 결과를 보관하는 방식.

@Insert 할 때 SQL 문을 보면 #{bno}와 같이 이미 처리된 결과를 이용.

insert 테스트 코드 작성

BoardMapperTests 클래스에 새로운 메서드를 추가.

```
BoardMapper.xml  BoardMapper.java  BoardMapperTests.java  ⌕
1  package com.freeflux.mapper;
2
3  import org.junit.Test;
4  import org.junit.runner.RunWith;
5  import org.springframework.beans.factory.annotation.Autowired;
6  import org.springframework.test.context.ContextConfiguration;
7  import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
8
9  import com.freeflux.domain.BoardVO;
10
11 import lombok.Setter;
12 import lombok.extern.log4j.Log4j;
```

```
14 @RunWith(SpringJUnit4ClassRunner.class)
15 @ContextConfiguration("file:src/main/webapp/WEB-INF/spring/root-context.xml")
16 @Log4j
17 public class BoardMapperTests {
18
19     @Setter(onMethod_ = @Autowired)
20     BoardMapper mapper;
21
22     @Test
23     public void testGetList() {
24         mapper.getList().forEach(board -> log.info(board));
25     }
26
27     @Test
28     public void testInsert() {
29         BoardVO board = new BoardVO();
30         board.setTitle("새로 작성하는 글");
31         board.setContent("새로 작성하는 내용");
32         board.setWriter("newbie");
33
34         mapper.insert(board);
35
36         log.info(board);
37     }
38
39     @Test
40     public void testInsertSelectKey() {
41         BoardVO board = new BoardVO();
42         board.setTitle("새로 작성하는 글 select key");
43         board.setContent("새로 작성하는 내용 select key");
44         board.setWriter("newbie");
45
46         mapper.insertSelectKey(board);
47
48         log.info(board);
49     }
50 }
```

테스트 코드의

각 메서드에서 마지막에 **log.info(board)**를 작성하는 이유 :

Lombok 이 만들어주는 toString()을 이용하여

bno 멤버 변수(인스턴스 변수)의 값을 알아보기 위함.

testInsert() 의 실행 결과

```
INFO : jdbc.audit - 1. PreparedStatement.new PreparedStatement returned
INFO : jdbc.audit - 1. Connection.prepareStatement(insert into tbl_board (bno,title,content,writer)
      values (seq_board.nextval, ?, ?, ?)) returned net.sf.log4jdbc.sql.jdbcapi.PreparedStatementSpy@68fa0ba8
INFO : jdbc.audit - 1. PreparedStatement.setString(1, "새로 작성하는 글") returned
INFO : jdbc.audit - 1. PreparedStatement.setString(2, "새로 작성하는 내용") returned
INFO : jdbc.audit - 1. PreparedStatement.setString(3, "newbie") returned
INFO : jdbc.sqlonly - insert into tbl_board (bno,title,content,writer) values (seq_board.nextval, '새로 작성하는 글', '새로
작성하는 내용', 'newbie')

INFO : jdbc.sqltiming - insert into tbl_board (bno,title,content,writer) values (seq_board.nextval, '새로 작성하는 글', '새로
작성하는 내용', 'newbie')
      {executed in 17 msec}
INFO : jdbc.audit - 1. PreparedStatement.execute() returned false
INFO : jdbc.audit - 1. PreparedStatement.getUpdateCount() returned 1
INFO : jdbc.audit - 1. PreparedStatement.isClosed() returned false
INFO : jdbc.audit - 1. PreparedStatement.close() returned
INFO : jdbc.audit - 1. Connection.clearWarnings() returned |
INFO : com.freeflux.mapper.BoardMapperTests - BoardVO(bno=null, title=새로 작성하는 글, content=새로 작성하는 내용,
INFO : org.springframework.context.support.GenericApplicationContext - Closing org.springframework.context.support.Ge
INFO : com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Shutdown initiated...
```

테스트 결과의 마지막을 살펴보면

BoardVO 클래스의 toString()의 결과가 출력되는 부분

bno 의 값이 null 로 비어 있는 것을 확인할 수 있다.

testInsertSelectKey() 의 실행 결과

```
INFO : jdbc.sqltiming - select seq_board.nextval from dual
      {executed in 36 msec}
INFO : jdbc.audit - 1. PreparedStatement.execute() returned true
INFO : jdbc.resultset - 1. ResultSet.new ResultSet returned
INFO : jdbc.audit - 1. PreparedStatement.getResultSet() returned net.sf.log4jdbc.sql.jdbcapi.ResultSetSpy@14f5da2c
INFO : jdbc.resultset - 1. ResultSet.getMetaData() returned oracle.jdbc.driver.OracleResultSetMetaData@4e28bdd1
INFO : jdbc.resultset - 1. ResultSet.getType() returned 1003
INFO : jdbc.resultset - 1. ResultSet.next() returned true
INFO : jdbc.resultset - 1. ResultSet.getLong(NEXTVAL) returned 26
INFO : jdbc.resultset - 1. ResultSet.isNull() returned false
INFO : jdbc.resultsettable -
|-----|
|nextval |
|-----|
|26      |
|-----|

INFO : jdbc.resultset - 1. ResultSet.next() returned false
INFO : jdbc.resultset - 1. ResultSet.close() returned void
INFO : jdbc.audit - 1. Connection.getMetaData() returned oracle.jdbc.driver.OracleDatabaseMetaData@1623134f
INFO : jdbc.audit - 1. PreparedStatement.isClosed() returned false
INFO : jdbc.audit - 1. PreparedStatement.close() returned
INFO : jdbc.audit - 1. PreparedStatement.new PreparedStatement returned
INFO : jdbc.audit - 1. Connection.prepareStatement(insert into tbl_board (bno,title,content, writer)
      values (?, ?, ?, ?)) returned net.sf.log4jdbc.sql.jdbcapi.PreparedStatementSpy@7a527389
INFO : jdbc.audit - 1. PreparedStatement.setLong(1, 26) returned
INFO : jdbc.audit - 1. PreparedStatement.setString(2, "새로 작성하는 글 select key") returned
INFO : jdbc.audit - 1. PreparedStatement.setString(3, "새로 작성하는 내용 select key") returned
INFO : jdbc.audit - 1. PreparedStatement.setString(4, "newbie") returned
INFO : jdbc.sqlonly - insert into tbl_board (bno,title,content, writer) values (26, '새로 작성하는 글 select key', '새로 작성하는
내용 select key', 'newbie')
```

실행 결과를 보면

'select seq_board.nextval from dual' 쿼리가 먼저 실행되고

여기서 생성된 결과를 이용하여

bno 값을 처리되는 것을 볼 수 있다.

'insert into tbl_board (bno,title,content, writer) values (?, ?, ?, ?)' 와 같이

파라미터로 전달되는 BoardVO 의 bno 값을 사용하게 되어 있다.

테스트 코드의 마지막 부분을 보면

BoardVO 객체의 bno 값이 지정된 것을 확인할 수 있다.

* 참고 : 시퀀스의 값이므로 테스트 할 때마다 다른 값이 나온다.

시퀀스 값 : 중복 없는 값을 위한 것일 뿐 다른 의미는 없다.

@SelectKey 를 이용하는 방식은

SQL 을 한 번 더 실행하는 부담이 있기는 하지만,

자동으로 추가되는 PK 값을 확인해야 하는 상황에서는 유용하게 사용될 수 있다.

2. read(select) 처리

insert 가 된 데이터를 조회하는 작업

PK 를 이용하여 처리하므로

BoardMapper 의 파라미터 역시 BoardVO 클래스의 bno 타입정보를 이용하여 처리.

BoardMapper 인터페이스에 read() 메서드 추가

```
BoardMapper.xml BoardMapper.java BoardMap
1 package com.freeflux.mapper;
2
3 import java.util.List;
4
5 //import org.apache.ibatis.annotations.Select;
6 import com.freeflux.domain.BoardVO;
7
8 public interface BoardMapper {
9
10 // @Select("select * from tbl_board where bno > 0")
11 public List<BoardVO> getList();
12
13 public void insert(BoardVO board);
14
15 public void insertSelectKey(BoardVO board);
16
17 public BoardVO read(Long bno);
18 }
```

BoardMapper.xml 문서에 <select> 추가

```
<insert id="insertSelectKey">
    <selectKey keyProperty="bno" order="BEFORE" resultType="long">
        select seq_board.nextval from dual
    </selectKey>
    insert into tbl_board (bno,title,content, writer)
    values ({bno}, #{title}, #{content}, #{writer})
</insert>

<select id="read" resultType="org.zerock.domain.BoardVO">
    select * from tbl_board where bno = #{bno}
</select>

</mapper>
```

MyBatis 는

Mapper 인터페이스의 리턴 타입에 맞게 select 의 결과를 처리하기 때문에
tbl_board 테이블 내 모든 컬럼은 BoardVO 의
'bno, title, content, writer, regdate, updateDate' 속성값으로 처리된다.

MyBatis 는 bno 라는 컬럼이 존재하면

BoardVO 인스턴스의 setBno()를 호출하여 해당 데이터를 세팅한다.

MyBatis 의 모든 파라미터와 리턴 타입의 처리는

get 파라미터명(), set 컬럼명()의 규칙으로 호출된다.

다만,

위와 같이 #{속성}이 1 개만 존재하는 경우에는

별도의 get 파라미터명()을 사용하지 않고 처리된다.

테스트 코드 (BoardMapperTests.java)

현재 테이블에 존재하는 데이터의 bno 컬럼의 값을 이용해서 확인.

데이터 읽기 위한 테스트 메서드 추가

```
@Test
public void testRead() {

    // 존재하는 게시물 번호로 테스트
    BoardVO board = mapper.read(5L);

    log.info(board);
}
```

```
INFO : jdbc.audit - 1. PreparedStatement.close() returned
INFO : jdbc.audit - 1. Connection.clearWarnings() returned
INFO : com.freeflux.mapper.BoardMapperTests - BoardVO(bno=5, title=테스트제목4, content
INFO : org.springframework.context.support.GenericApplicationContext - Closing org.springf
INFO : com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Shutdown initiated...
```

3. delete 처리

특정한 데이터를 삭제하는 작업 역시 PK 값을 이용하여 처리하므로. 조회하는 작업과 유사하게 처리하면 된다.

등록, 삭제, 수정과 같은 DML 작업은 '몇 건의 데이터가 삭제(혹은 수정)되었는지'를 반환 받을 수 있다.

BoardMapper 인터페이스에 메서드 선언 추가.

delete() 메서드의 리턴 타입은 int 로 지정하여 만일 정상적으로 데이터가 삭제 되면 1 이상의 값을 가지도록 작성.

```
BoardMapper.xml BoardMapper.java BoardMap
1 package com.freeflux.mapper;
2
3 import java.util.List;
4
5 //import org.apache.ibatis.annotations.Select;
6 import com.freeflux.domain.BoardVO;
7
8 public interface BoardMapper {
9
10 // @Select("select * from tbl_board where bno > 0")
11 public List<BoardVO> getList();
12
13 public void insert(BoardVO board);
14
15 public void insertSelectKey(BoardVO board);
16
17 public BoardVO read(Long bno);
18
19 public int delete(Long bno);
20 }
```

BoardMapper.xml 에 삭제 관련 태그 추가

```
<select id="read" resultType="com.freeflux.domain.BoardVO">
    select * from tbl_board where bno = #{bno}
</select>

<delete id="delete">
    delete from tbl_board where bno = #{bno}
</delete>

</mapper>
```


테스트 코드 (BoardMapperTests.java)

테스트는 현재 존재하는 번호의 데이터를 삭제해보고 '1'이라는 값이 출력되는 지를 확인.
만약 해당 게시물이 없다면 '0'이 출력된다.

```
@Test
public void testDelete() {
    log.info("DELETE COUNT : " + mapper.delete(3L));
}
```

위의 testDelete() 메서드 경우, 3 번 데이터가 존재했다면
아래와 같은 로그가 출력된다.

```
INFO : jdbc.connection - 9. Connection opened
INFO : jdbc.audit - 9. Connection.new Connection returned
INFO : jdbc.audit - 9. Connection.setReadOnly(false) returned
INFO : jdbc.audit - 9. Connection.setAutoCommit(true) returned
INFO : jdbc.sqltiming - delete from tbl_board where bno = 3
      {executed in 32 msec}
INFO : jdbc.audit - 1. PreparedStatement.execute() returned false
INFO : jdbc.audit - 1. PreparedStatement.getUpdateCount() returned 1
INFO : jdbc.audit - 1. PreparedStatement.isClosed() returned false
INFO : jdbc.audit - 1. PreparedStatement.close() returned
INFO : jdbc.audit - 1. Connection.clearWarnings() returned
INFO : com.freeflux.mapper.BoardMapperTests - DELETE COUNT : 1
INFO : org.springframework.context.support.GenericApplicationContext - Closing
INFO : com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Shutdown initiated...
```

4. update 처리

게시물의 업데이트는 제목, 내용, 작성자를 수정한다고 가정.

업데이트할 때에는

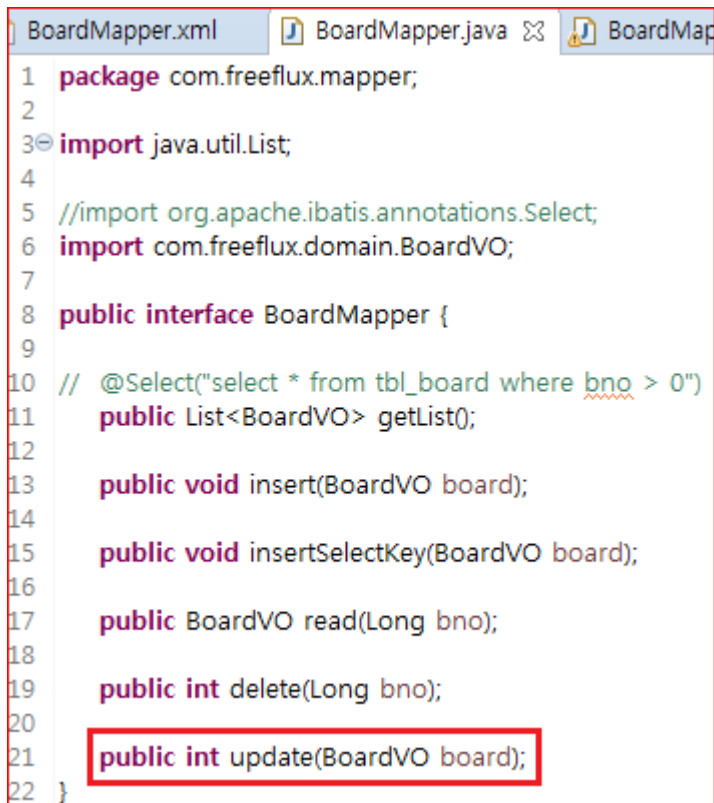
최종 수정시간을 데이터베이스 내의 현재 시간으로 수정한다.

update 는 delete 와 마찬가지로

'몇 개의 데이터가 수정되었는가'를 반환 받을 수 있도록

int 타입으로 메서드의 리턴 값을 설계.

BoardMapper 인터페이스에 메서드 선언 추가



```
1 package com.freeflux.mapper;
2
3 import java.util.List;
4
5 //import org.apache.ibatis.annotations.Select;
6 import com.freeflux.domain.BoardVO;
7
8 public interface BoardMapper {
9
10 // @Select("select * from tbl_board where bno > 0")
11 public List<BoardVO> getList();
12
13 public void insert(BoardVO board);
14
15 public void insertSelectKey(BoardVO board);
16
17 public BoardVO read(Long bno);
18
19 public int delete(Long bno);
20
21 public int update(BoardVO board);
22 }
```

BoardMapper.xml 에 update 태그 추가

update 컬럼 : 최종 수정 시간을 의미하는 컬럼이므로

현재 시간으로 변경하기 위해 sysdate 로 설정.

regdate 컬럼 : 최초 생성 시점(게시 시점)이므로 수정하지 않는다.

각 수정 값(#{title})은

파라미터로 전달된 BoardVO 객체의 getTitle()과 같은 메서드를 호출하여 처리.

```

</delete>

<update id="update">
    update tbl_board
    set title = #{title}, content = #{content}, writer = #{writer}, updateDate = sysdate
    where bno = #{bno}
</update>

</mapper>

```

테스트 코드 (BoardMapperTests.java)

직접 BoardVO 객체를 생성하여 처리.

```

@Test
public void testUpdate() {
    BoardVO board=new BoardVO();
    board.setBno(5L);
    board.setTitle("수정된 제목");
    board.setContent("수정된 내용");
    board.setWriter("user00");

    int count=mapper.update(board);
    log.info("UPDATE COUNT : " + count);
}

```

만약 데이터베이스에 5 번 글이 존재한다면
아래와 같은 로그들이 출력된다.

```

INFO : jdbc.connection - 10. Connection opened
INFO : jdbc.audit - 10. Connection.new Connection returned
INFO : jdbc.audit - 10. Connection.setReadOnly(false) returned
INFO : jdbc.audit - 10. Connection.setAutoCommit(true) returned
INFO : jdbc.audit - 1. PreparedStatement.setString(1, "수정된 제목") returned
INFO : jdbc.audit - 1. PreparedStatement.setString(2, "수정된 내용") returned
INFO : jdbc.audit - 1. PreparedStatement.setString(3, "user00") returned
INFO : jdbc.audit - 1. PreparedStatement.setLong(4, 5) returned
INFO : jdbc.sqlonly - update tbl_board set title = '수정된 제목', content = '수정된 내용', updateDate = sysdate where bno = 5

INFO : jdbc.sqltiming - update tbl_board set title = '수정된 제목', content = '수정된 내용', updateDate = sysdate where bno = 5
{executed in 0 msec}
INFO : jdbc.audit - 1. PreparedStatement.execute() returned false
INFO : jdbc.audit - 1. PreparedStatement.getUpdateCount() returned 1
INFO : jdbc.audit - 1. PreparedStatement.isClosed() returned false
INFO : jdbc.audit - 1. PreparedStatement.close() returned
INFO : jdbc.audit - 1. Connection.clearWarnings() returned
INFO : com.freeflux.mapper.BoardMapperTests - UPDATE COUNT : 1
INFO : org.springframework.context.support.GenericApplicationContext - Closing org.springframework.context.support.GenericApplicationContext@7445e54d: startup = true, shutdown = true, closed = false, refresh = false
INFO : com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Shutdown initiated...

```