

# chapter06

## JDBC DAO

1. **JDBC-개요**
2. JDBC-환경구성
3. JDBC-활용
4. DAO

## ■ JDBC(Java Database Connectivity)의 정의

- 자바를 이용한 데이터베이스 접속과 SQL 문장의 실행, 그리고 실행 결과로 얻어진 데이터의 핸들링을 제공하는 방법과 절차에 관한 규약
- 자바 프로그램내에서 SQL문을 실행하기 위한 자바 API
- SQL과 프로그래밍 언어의 통합 접근 중 한 형태

## ■ 개발자를 위한 표준 인터페이스인 JDBC API와 데이터베이스 벤더, 또는 기타 써드파티에서 제공하는 드라이버(driver)

# chapter06

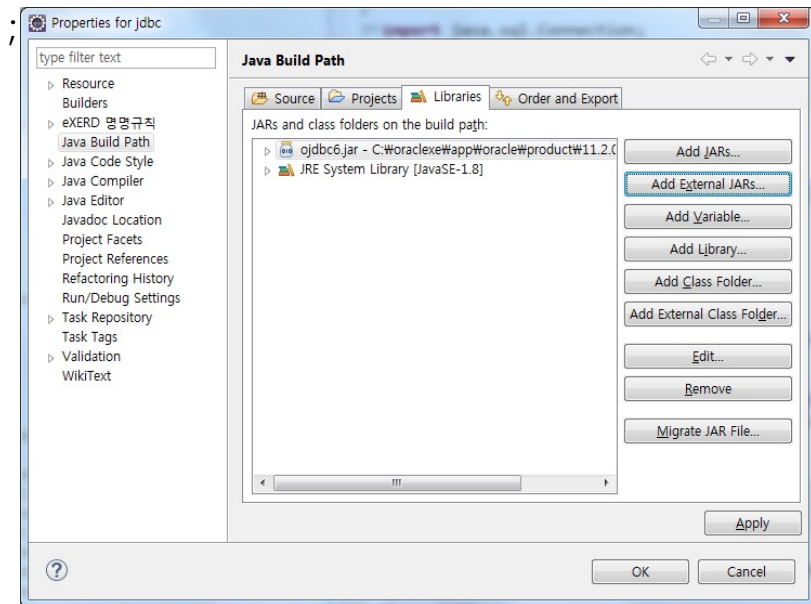
## JDBC DAO

1. JDBC-개요
2. **JDBC-환경구성**
3. JDBC-활용
4. DAO

## ■ 이클립스 JDBC설정

- 1) 프로젝트 우측 클릭 - properties - Java Build Path – Libraries(탭) - Add External JARs 버튼클릭
- 2) ojdbc6.jar 선택

오라클을 설치한 경우 위치 : **(C:\Woraclexe\app\Woracle\product\11.2.0\server\jdbc\lib)**



# chapter06

## JDBC DAO

1. JDBC-개요
2. JDBC-환경구성
3. **JDBC-활용**
4. DAO

## 03 JDBC-활용

### ■ JDBC를 이용한 데이터 베이스 연결방법

//0. import java.sql.\*;

//1. JDBC 드라이버 (Oracle) 로딩

//2. Connection 얻어오기

//3. SQL문 준비 / 바인딩 / 실행

// 4.결과처리

//5. 자원정리

```
// 0. import java.sql.*;
Connection conn = null;
PreparedStatement pstmt = null;
ResultSet rs = null;

try {
    // 1. JDBC 드라이버 (Oracle) 로딩

    // 2. Connection 얻어오기

    // 3. SQL문 준비 / 바인딩 / 실행

    // 4.결과처리

} catch (ClassNotFoundException e) {
    System.out.println("error: 드라이버 로딩 실패 - " + e);
} catch (SQLException e) {
    System.out.println("error:" + e);
} finally {

    // 5. 자원정리
    try {
        if (rs != null) {
            rs.close();
        }
        if (pstmt != null) {
            pstmt.close();
        }
        if (conn != null) {
            conn.close();
        }
    } catch (SQLException e) {
        System.out.println("error:" + e);
    }
}
```

- 0. import

ctrl + shift + o 자동 인포트됨

- 1. JDBC 드라이버(Oracle) 로딩

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

- 2. Connection 얻어오기

```
String url = "jdbc:oracle:thin:@localhost:1521:xe";  
conn = DriverManager.getConnection(url, "webdb", "webdb");
```

### • 3. SQL문 준비 / 바인딩 / 실행

```
int count = pstmt.executeUpdate();    //insert, update, delete
```

```
ResultSet rs = pstmt.executeQuery(); //select
```

### • 4. 결과처리

#### ResultSet rs

	author_id	author_name	author_desc
Cursor	1	이문열	경북 영양
	2	박경리	경상남도 통영
	3	유시민	17대 국회의원
	4	기안84	기안동에서 산 84년생
	5	강풀	온라인 만화가 1세대
	6	김영하	알쓸신잡

next()



getInt(1)

getString(2)

getString(author\_desc)

getInt(숫자/컬럼명)

getString(숫자/컬럼명)



- 5. 자원정리

```
    if (rs != null) {  
        rs.close();  
    }  
    if (pstmt != null) {  
        pstmt.close();  
    }  
    if (conn != null) {  
        conn.close();  
    }
```

## ■ 실습

AuthorInsertTest.java

AuthorSelectTest.java

AuthorUpdateTest.java

AuthorDeleteTest.java

## ■ 실습2

BookInsert.java

BookSelect.java

BookUpdate.java

BookDelete.java

BookInfo.java (Join 활용)

FindBookInfo.java  
( 저자 이름으로 등록된 책 정보들 출력)

**package** com.kosta.jdbc.본인이름이니셜(imh);

FindPostCode.java <- 우편번호 검색 프로그램 완성후 (세종시)

jongukjeong@gmail.com

# chapter06

## JDBC DAO

1. JDBC-개요
2. JDBC-환경구성
3. JDBC-활용
4. **DAO**

## ■ DAO(Data Access Object)

- DB를 사용해 데이터를 조회하거나 조작하는 기능을 전담하도록 만든 오브젝트



## ■ DAO(Data Access Object)

- DB를 사용해 데이터를 조회하거나 조작하는 기능을 전담하도록 만든 오브젝트



## ■ 실습

### AuthorDao 연습

- AuthorVo 작성
- AuthorDao 작성
  - 입력하기
  - 전체가져오기
  - 삭제하기
  - 업데이트하기

### ■ 실습

#### AuthorDao 연습

- AuthorVo 작성

- AuthorDao 작성

- 입력하기

- 전체가져오기

- 삭제하기

- 업데이트하기

#### BookDAO 연습

- BookVo 작성

- BookDao 작성

- 입력하기

- 전체가져오기

- 삭제하기

- 업데이트하기

+ 검색하기



## ■ 문제

- Scanner 를 사용해서 키워드를 입력받아 아래와 같이 도서 정보를 검색 프로그램을 작성해 보세요
- title, pubs, author\_name 필드중 한곳이라도 해당 키워드를 포함하고 있으면 출력합니다.
- Dao, Vo 클래스를 사용하세요

자동증가

book_id	title	pubs	pub_date	author_name
1	우리들의 일그러진 영웅	다림	1998-02-22	이문열
2	삼국지	민음사	2002-03-01	이문열
3	토지	마로니에북스	2012-08-15	박경리
4	자바프로그래밍 입문	위키북스	2015-04-01	이고잉
5	패션왕	중앙북스(books)	2012-02-22	기안84
6	순정만화	재미주의	2011-08-03	강풀
7	오직두사람	문학동네	2017-05-04	김영하
8	26년	재미주의	2012-02-04	강풀