

JDBC





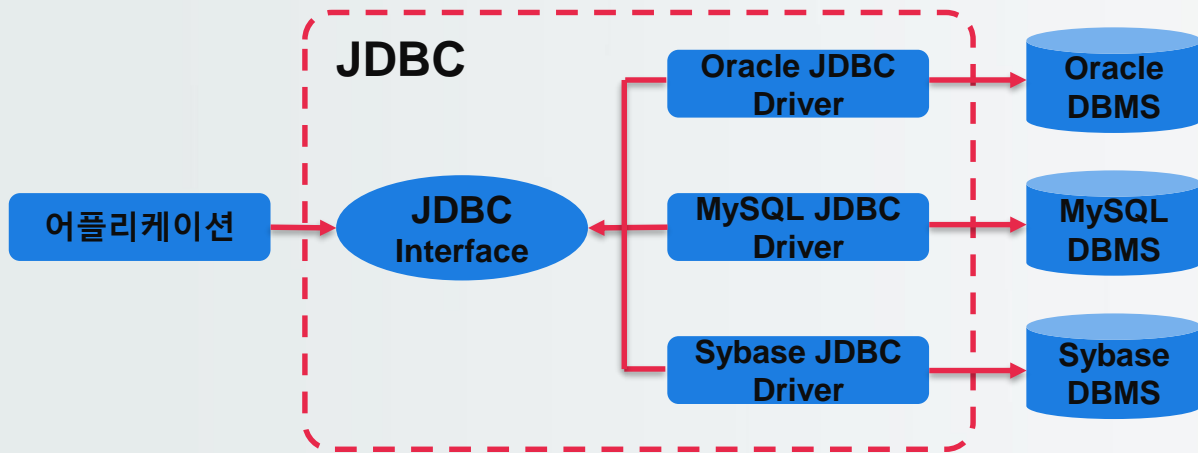
JDBC



JDBC

JDBC(Java DataBase Connectivity)

- 자바 언어에서 DataBase에 접근 할 수 있게 해주는 Programming API
- Java에서 DBMS연동에 필요한 메소드를 Connection 인터페이스로 제공
- 각 DBMS 제조사 별 구현하여(jar파일) DBMS 접속
- java.sql 패키지에서 관리



JDBC

ojdbc

- 오라클에서 제공하는 오라클 DBMS와 자바를 연결하기 위한 라이브러리

ojdbc 라이브러리 다운로드

- 1) PC에 오라클이 설치되어 있지 않은 경우

<http://www.oracle.com/technetwork/apps-tech/jdbc-112010-090769.html>

→ 다운로드

- 2) PC에 오라클이 설치되어 있는 경우

C:\oracle\exe\app\oracle\product\11.2.0\server\jdbc\lib

각각 해당 경로를 접속하여 ojdbc6.jar 파일을 jdk 폴더로 복사

C:\Program Files\Java\jdk1.8.0_144\jre\lib\ext 폴더 내부에 복사

오라클 버전	ojdbc 파일	오라클 버전	ojdbc 파일
oracle 7	classes1111.zip	oracle 8	classes12.zip
oracle 9i	ojdbc14.jar	oracle 10g	ojdbc14.jar
oracle 11g	ojdbc6.jar		





Eclipse Setting



JDBC

JDBC 사용을 위한 Eclipse Setting

1. ojdbc 라이브러리 등록
 - Java의 표준 라이브러리가 아닌 외부 라이브러리를 사용하기 때문에 해당 라이브러리 사용을 위한 등록 과정이 필요
2. 문자 set 인코딩
 - 한글을 표현하는 문자의 인코딩 방식은 다양(MS949, UTF-8, UTF-16, EUC-KR 등등)
 - Eclipse는 기본적으로 MS949 방식으로 한글 인코딩
 - Oracle은 UTF-8 방식으로 한글 인코딩
 - 한글 인코딩 방식이 다르기 때문에 한글이 정상적으로 처리되지 않으므로 인코딩 방식을 UTF-8로 일치

※ 라이브러리 등록 및 문자set 인코딩은 Eclipse workspace를 새로 만들 때 마다 해야 함
※ 추가로 JDBC 이 후 수업에 사용할 web문서 관련 인코딩도 미리 한번에 진행

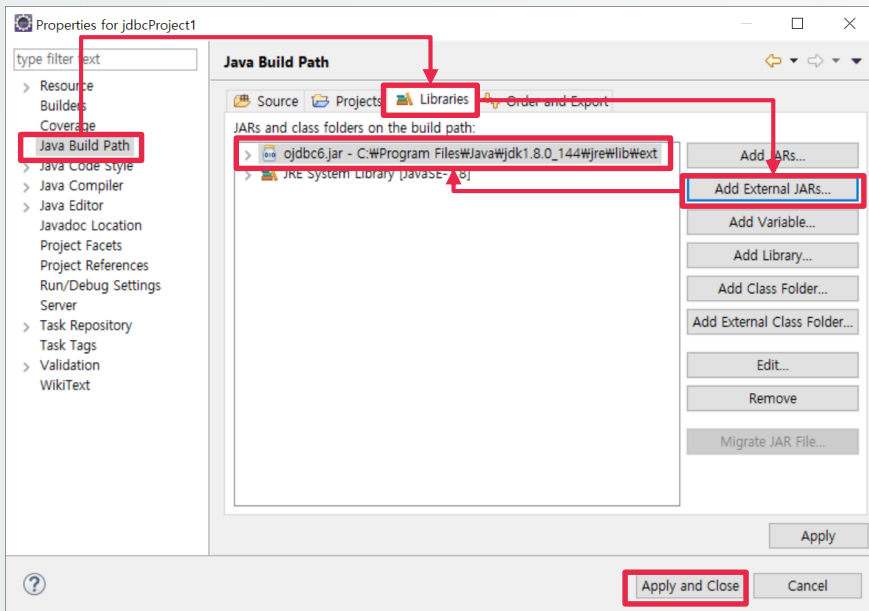
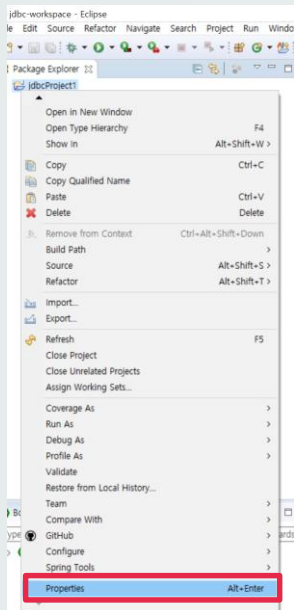


JDBC

JDBC 사용을 위한 Eclipse Setting

1. ojdbc 라이브러리 등록

프로젝트 생성 → 프로젝트 마우스 우클릭 → Java Build Path → Libraries → Add External JARs... → C:\Program Files\Java\jdk1.8.0_144\jre\lib\ext → ojdbc6.jar 선택

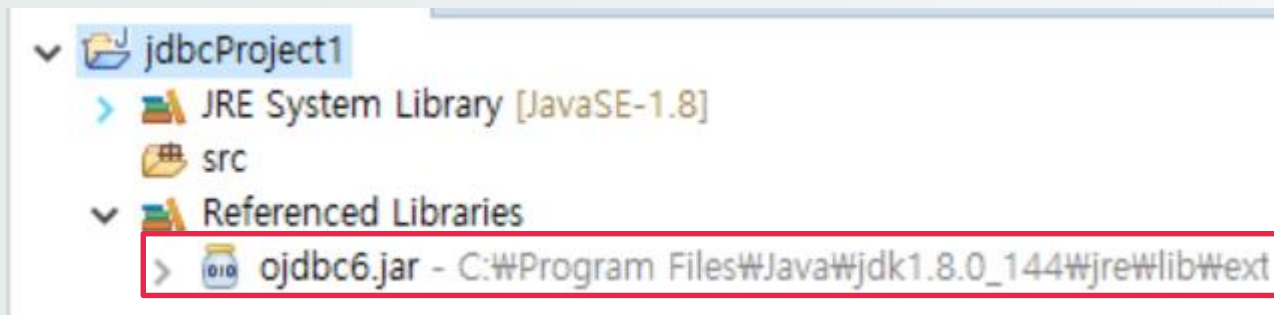


JDBC

JDBC 사용을 위한 Eclipse Setting

1. ojdbc 라이브러리 등록

적용 후 project → Referenced Libraries → ojdbc6.jar 확인

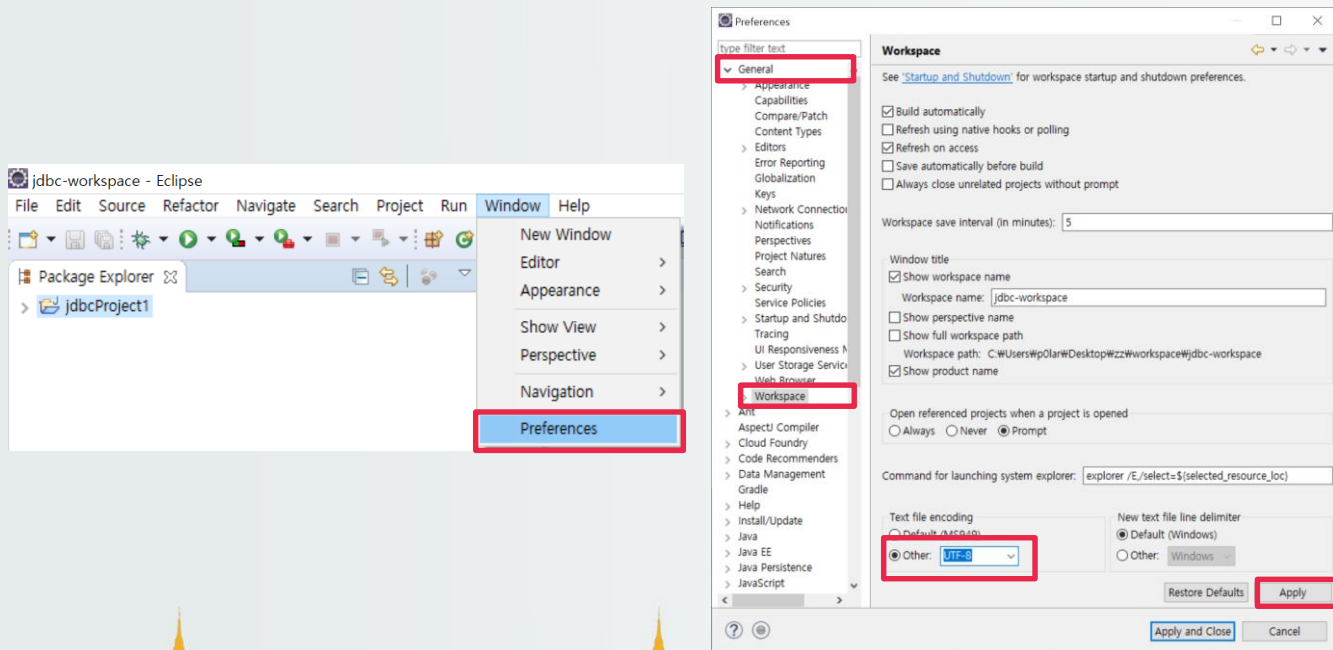


JDBC

JDBC 사용을 위한 Eclipse Setting

2. 문자 set 인코딩

window → preferences → General → Workspace → UTF-8 설정 → Apply

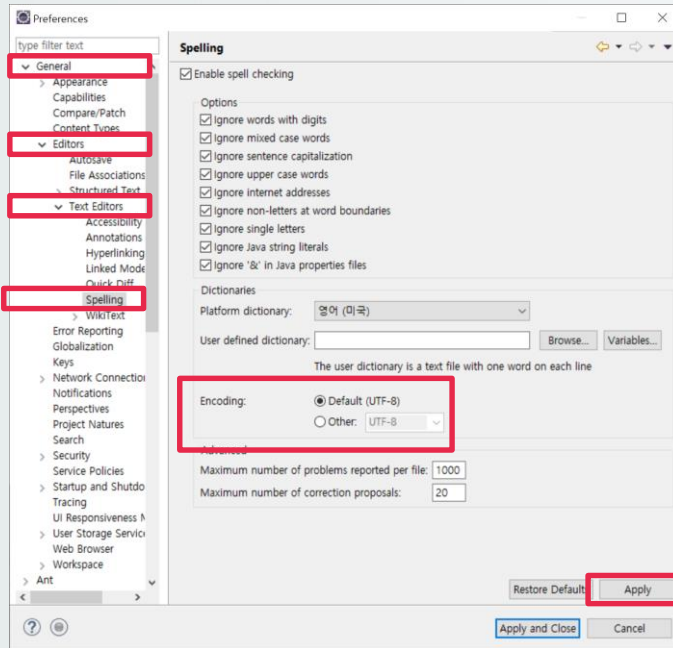


JDBC

JDBC 사용을 위한 Eclipse Setting

2. 문자 set 인코딩

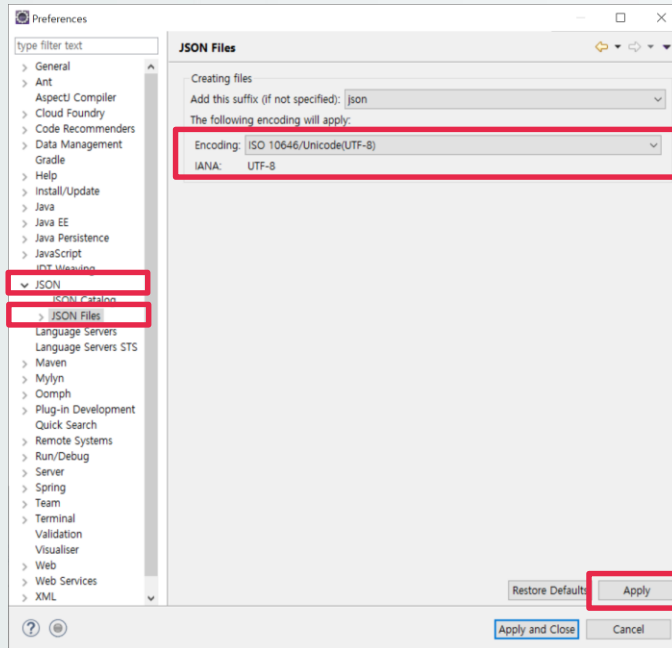
General → Editors → Text Editors → Spelling → UTF-8 설정 → Apply



JDBC

JDBC 사용을 위한 Eclipse Setting

2. 문자 set 인코딩(웹문서 인코딩)
JSON → JSON Files → UTF-8 설정 → Apply

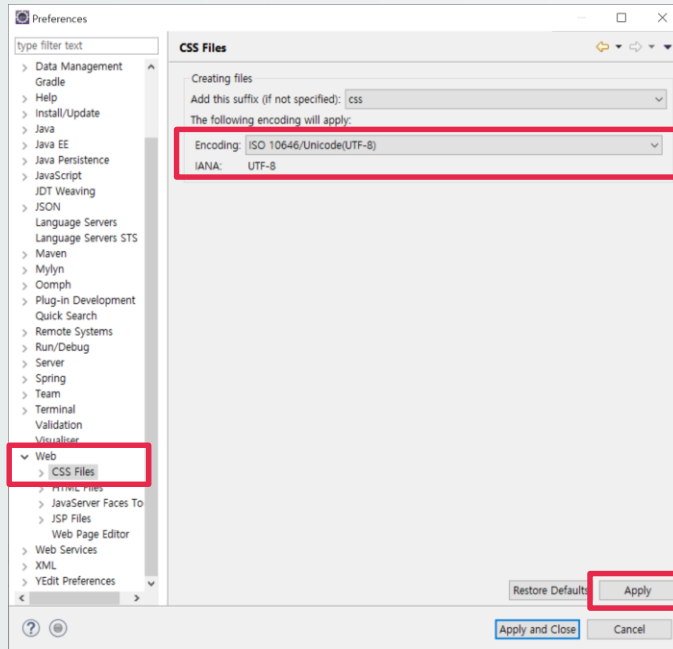


JDBC

JDBC 사용을 위한 Eclipse Setting

2. 문자 set 인코딩(웹문서 인코딩)

Web → CSS Files → UTF-8 설정 → Apply

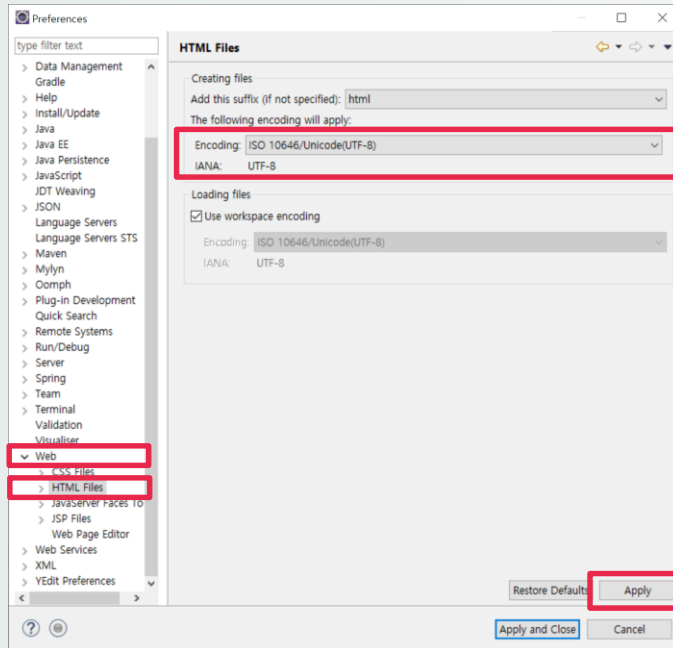


JDBC

JDBC 사용을 위한 Eclipse Setting

2. 문자 set 인코딩(웹문서 인코딩)

Web → HTML Files → UTF-8 설정 → Apply

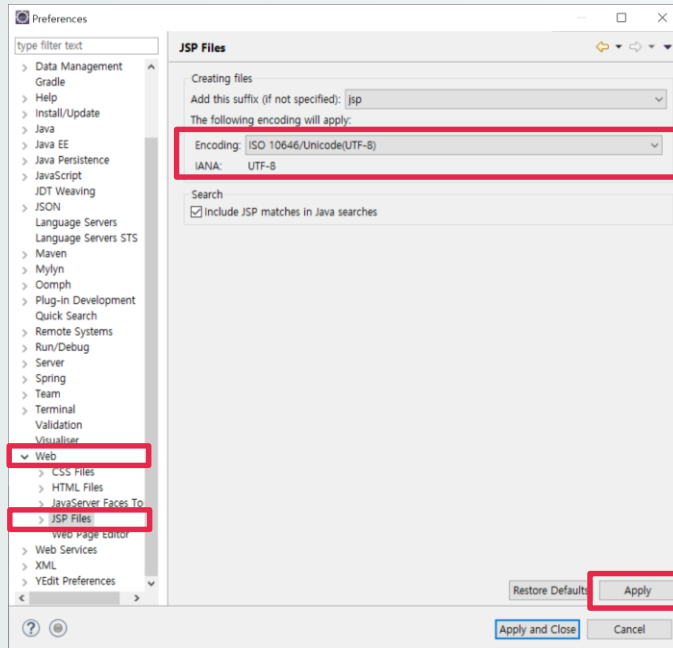


JDBC

JDBC 사용을 위한 Eclipse Setting

2. 문자 set 인코딩(웹문서 인코딩)

Web → JSP Files → UTF-8 설정 → Apply





JDBC Class와 Interface



JDBC

1. DriverManager

- 데이터 원본에 JDBC 드라이버를 통하여 커넥션을 만드는 역할
- Class.forName() 메소드를 통해 생성되며, 반드시 예외처리를 해야 함
- 직접 객체 생성이 불가능하고, getConnection() 메소드로 객체 생성

2. Connection

- 특정 데이터 원본과 연결된 커넥션
- Statement 객체를 생성할 때도 Connection 객체를 이용해야 함
- SQL 문장을 실행시키기 전 우선 Connection 객체가 필요



JDBC

3. Statement

- Connection 객체에 의해 프로그램에 리턴 되는 객체에 의해 구현되는 일종의 메소드 집합
- Connection 클래스의 createStatement() 메소드를 호출하여 객체 생성
- Statement 객체로 SQL문을 String 객체에 담아 인자로 전달하여 질의를 수행

4. PreparedStatement

- Statement 클래스를 상속하여 만들어진 클래스로 기본적인 수행역할은 동일
- SQL 문장이 미리 컴파일 되고, 실행시간 동안 인수 값을 위한 공간을 확보할 수 있다는 점에서 Statement 와 다름
- 각 각의 인수에 대해 위치홀더(?)를 사용하여 SQL 문장을 정의 할 수 있음
- Connection 클래스의 preparedStatement() 메소드를 호출하여 객체 생성



JDBC

5. ResultSet

- SELECT 문을 사용한 질의 성공 시 반환되는 객체
- ResultSet은 SQL 질의에 의해 생성된 결과를 담고 있으며 '커서(cursor)'를 이용하여 특정 행에 대한 참조를 조작





JDBC 작성 절차



JDBC 작성절차



- ORACLE Driver 등록

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

- 다른 DB

```
Class.forName("com.Microsoft.jdbc.sqlserver.SqlServerDriver");//MSSql
```

```
Class.forName("org.gjt.mm.mysql.Driver");//MySQL
```

※ 반드시 **ClassNotFoundException** 처리를 해야 한다.
ojdbc 라이브러리를 등록하지 않는 경우 에러 발생

JDBC 작성절차



- DB와 접속하여 Connection객체에 저장
`Connection conn = DriverManager.getConnection(
"jdbc:oracle:thin:@127.0.0.1:1521:xe",
"kh",
"kh"
);`

※ 반드시 **SQLException** 처리를 해야 한다.
DB접속 정보가 틀린 경우 에러 발생



JDBC 작성절차



- Connection객체를 이용하여 Statement 객체 생성
`Statement stmt = conn.createStatement();`

※ 반드시 **SQLException** 처리를 해야 한다.



JDBC 작성절차



- String 객체를 이용하여 SQL Query를 작성하여 전송 후 결과를 받음

i) SELECT 문의 경우 리턴되는 데이터가 ResultSet

```
String query = "select emp_id,emp_name from employee";
```

```
ResultSet rset = stmt.executeQuery(query);
```

ii) INSERT, UPDATE, DELETE 문은 리턴되는 데이터가 int(변경된 행의 수)

```
String query = "update employee set salary=9000000 where  
emp_id=200";
```

```
int result = stmt.executeUpdate(query);
```

※ 반드시 **SQLException** 처리를 해야 한다.



JDBC 작성절차



- 사용한 객체들을 반환
`rset.close();` //SELECT를 수행하여 ResultSet을 사용한 경우에만 반환
`stmt.close();`
`conn.close();`

※ 반드시 **SQLException** 처리를 해야 한다.

