Java 프로그래밍 길잡이

✓ 원리를 알면 IT가 맛있다

Java Programming for Beginners

chapter 11.

JDBC

- JDBC 의 개요에 관하여 학습한다.
- DAO (Data Access Object) 패턴에 관하여 학습한다.
- DTO (Data Transfer Object) 패턴에 관하여 학습한다.

❑ 1) JDBC [Java DataBase Connectivity]

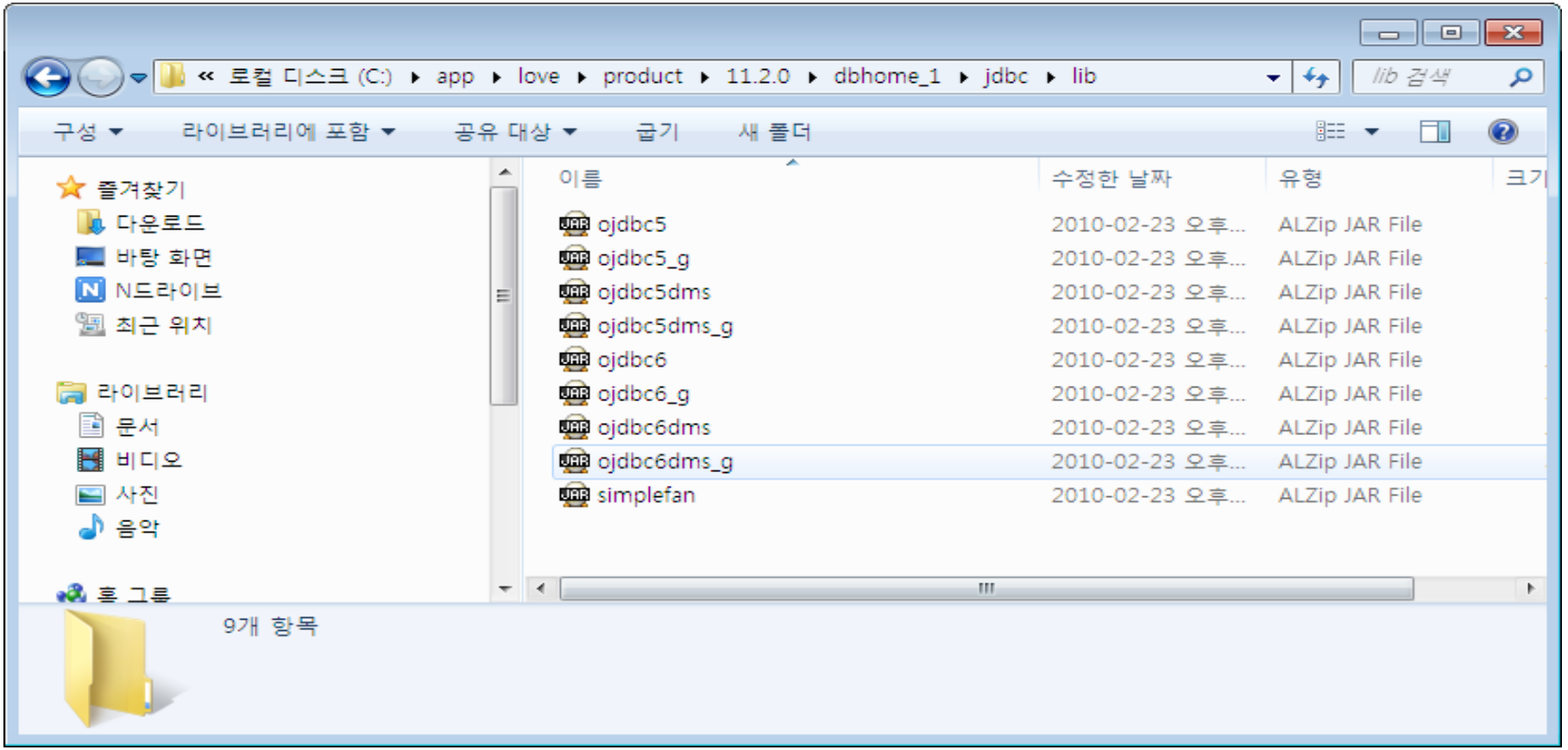
❍ 정의:

JDBC는 자바언어에서 데이터베이스에 접근할 수 있는 프로그래밍 API 이다.

❍ 특징:

JDBC는 대부분이 인터페이스로 되어있으며 이 인터페이스를 구현한 클래스 파일들의 묶음을 드라이버(driver)라고 한다. JDBC는 자바에서 제공하고 드라이버는 데이터베이스를 만드는 회사(벤더)에서 제공하며, 오라클에서 만든 드라이버를 '오라클 드라이버'라고 한다.
오라클 드라이버는 오라클 데이터베이스를 설치하면 다음 경로에 자동 생성된다.

C:₩app₩사용자명₩product₩11.2.0₩dbhome_1₩jdbc₩lib



상위 파일에서 ojdbc6_g.jar 파일을 CLASSPATH에 추가한다.

❑ 2) JDBC 실행순서

다음은 JDBC를 구현하는 기본적인 프로그램을 개발하는 순서이다.

1. 오라클 데이터베이스 연동을 위한 4가지 정보를 저장한다.

```
String driver = "oracle.jdbc.driver.OracleDriver";
String url = "jdbc:oracle:thin:@localhost:1521:orcl";
String userid = "scott";
String passwd = "tiger";
```

2. 드라이버 로딩

```
Class.forName( driver );
```

3. Connection 맺기

```
Connection con = DriverManager.getConnection( url, userid , passwd );
```

4. SQL 문 작성 (주의할 점은 ;(세미콜론)은 사용 안함)

```
String query = "SELECT deptno,dname,loc FROM dept";
또는
String query = "DELETE FROM dept WHERE deptno = 40";
```

5. PreparedStatement 생성

```
PreparedStatement pstmt = con.prepareStatement( query);
```

6. SQL 문 전송 및 결과값 얻기

-DML 요청 (INSERT, DELETE, UPDATE) 인 경우 코드

```
int n = pstmt.executeUpdate( );
```

- SELECT 요청

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

rs.next() 및 rs.getInt(컬럼명), rs.getString(컬럼명) 이용하여 데이터 추출.

```
예>
    while( rs.next()){
        int deptno = rs.getInt("deptno");
        String dname = rs.getString("dname");
        String loc = rs.getString("loc");
        System.out.println( deptno +“ ” + dname + " " +loc );
}
```

7. 자원 반납 (사용했던 자원 역순)

```
rs.close(); // ResultSet을 사용한 경우
stmt.close();
con.close();
```

* 레코드 검색하기 샘플 (SELECT 샘플)

```
..
11 public static void main( String [] args ) {
12
13    String driver ="oracle.jdbc.driver.OracleDriver";
14    String url = "jdbc:oracle:thin:@localhost:1521:orcl";
15    String userid = "scott";
16    String passwd = "tiger";
17    Connection con = null;
18    PreparedStatement pstmt = null;
19    ResultSet rs = null;
20  try{
21     Class.forName(driver);
22     con = DriverManager.getConnection( url, userid, passwd);
23     String query = "SELECT deptno,dname,loc FROM dept";
24     pstmt = con.prepareStatement(query);
25
```

```
26        rs = pstmt.executeQuery( );
27       while( rs.next() ){
28          int deptno = rs.getInt("deptno");
29         String dname = rs.getString("dname");
30         String loc = rs.getString("loc");
31         System.out.println( deptno +" " + dname + " "+ loc );
32       }
33    }catch(Exception e){
34          e.printStackTrace();
35    }finally{
36       try{
37             if(rs != null ) rs.close();
38             if(pstmt != null ) pstmt.close();
39             if(con != null ) con.close();
40        }catch(SQLException e){
41            e.printStackTrace();
42        }
43  …
```

출력결과

```
10 ACCOUNTING NEW YORK
20 RESEARCH DALLAS
30 SALES CHICAGO
40 OPERATIONS BOSTON
```

* 레코드 저장하기 샘플 (DML 샘플)

```
..
11 public static void main( String [] args ) {
12
13    String driver ="oracle.jdbc.driver.OracleDriver";
14    String url = "jdbc:oracle:thin:@localhost:1521:orcl";
15    String userid = "scott";
16    String passwd = "tiger";
17    Connection con = null;
18    PreparedStatement pstmt = null;
19    ResultSet rs = null;
20  try{
21     Class.forName(driver);
22     con = DriverManager.getConnection( url, userid, passwd);
23     String sql = "INSERT INTO dept (deptno,dname,loc) " +
                                        " VALUES ( ?, ?, ? )";
```

```
25            pstmt.setInt(1, 50 );
26            pstmt.setString(2, "개발");
27            pstmt.setString(3, "서울");
28            int n = pstmt.executeUpdate();
29            System.out.println( n +" 개의 레코드가 저장");
30          }catch(Exception e){
31              e.printStackTrace();
32          }finally{
33          try{
34              if(pstmt != null ) pstmt.close();
35              if(con != null ) con.close();
36          }catch(SQLException e){
37              e.printStackTrace();
38          }
39      }
40  }
41 }
```

출력결과

```
1 개의 레코드가 저장
```

❑ 3) DAO 패턴과 DTO 패턴

데이터베이스를 연동하는 프로그램을 개발할 때 반드시 사용되는 2 가지 개발 패턴이 있다.

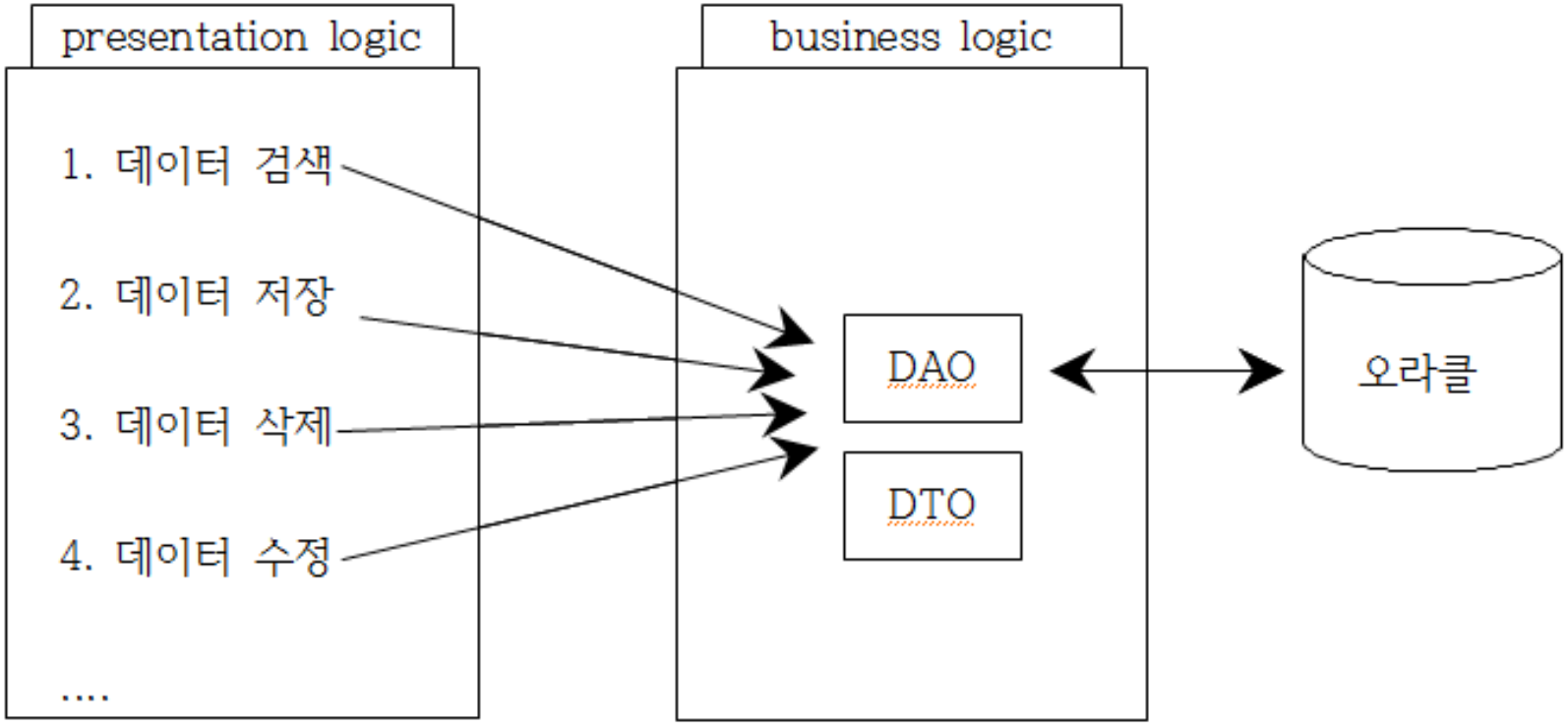
• DAO (Data Access Object) 패턴

GUI 화면을 구성하는 코드를 Presentation Logic 이라고 하며 GUI화면에 데이터를 보여주기 위해서 데이터베이스를 검색하는 코드 및 GUI화면에서 새로 발생된 데이터 (예를 들면 회원가입)를 데이터베이스에 저장하는 코드와 같은 실제적인 작업을 하는 코드를 business logic 이라고 한다. presentation logic 과 business logic을 하나의 클래스로 모두 구현 할 수도 있고 여러 클래스로 모듈화 시켜서 구현할 수도 있다. 모듈화한 클래스들중에서 데이터베이스 접근하는 코드만을 관리하는 클래스를 작성할 수 있는데 이 클래스 파일을 DAO 클래스라고 한다. 일반적으로 DAO 클래스는 테이블 당 한 개씩 생성해서 사용한다. DAO 클래스안에는 특정 테이블에서 수행할 작업을 메소드로 정의해서 구현하며 presentation logic에서는 DAO 클래스의 메소드를 호출하면서 원하는 작업을 구현하게 된다.

• DTO (Data Transfer Object) 패턴

presentation logic과 business logic을 여러 클래스로 분리해서 작업은 하지만 서로 간에 긴밀한 관계가 유지되면서 작업이 이루어진다. presentation logic 에서 보여줄 데이터를 얻기 위해서 business logic에게 요청을 하면 business logic은 필요한 데이터를 데이터베이스에서 검색해서 presentation logic에게 반환하는 작업등을 하게 된다.
이렇게 데이터를 다른 logic에게 전송 및 반환할 때 효율적으로 데이터를 사용할 수 있게 클래스를 작성할 수 있는데 이 클래스를 DTO 클래스라고 한다. 이름 그대로 데이터를 전송할 때 사용되는 클래스이다. DTO 클래스를 사용하면 데이터를 전송할 때와 전송된 데이터를 얻어서 사용할 때 효율적으로 사용할 수 있는 장점이 있다.

❑ 3) DAO 패턴과 DTO 패턴



* DTO 클래스 샘플

```
 ..
03 public class ExDTO {
05    private int deptno;
06    private String dname;
07    private String loc;
08
09    public ExDTO(int deptno, String dname, String loc) {
10          this.deptno = deptno;
11          this.dname = dname;
12          this.loc = loc;
13    }
14    public ExDTO() {}
16    public int getDeptno() {return deptno;}
19    public void setDeptno(int deptno) {this.deptno = deptno;}
22    public String getDname() {return dname;}
25    public void setDname(String dname) {this.dname = dname;}
34  ..
```

* DAO 클래스 샘플

```
..  //DB 연동코드 작성후
   public ArrayList<ExDTO> select(){
16       ArrayList<ExDTO> list = new ArrayList<ExDTO>();
17      Connection con = null;
18      PreparedStatement pstmt = null;
19      ResultSet rs = null;
20   try{
21      con = DriverManager.getConnection(url, userid , passwd );
22     String query = "SELECT deptno,dname,loc FROM dept";
23     pstmt = con.prepareStatement( query );
24     rs = pstmt.executeQuery();
25     while( rs.next() ){
26        ExDTO dto = new ExDTO();
27        dto.setDeptno( rs.getInt("deptno" ) );
28        dto.setDname( rs.getString( "dname" ) );
29        dto.setLoc( rs.getString("loc") );
```

```
30          list.add( dto );
31       }
32    }catch(Exception e){
33          e.printStackTrace();
34    }finally{
35       try{
36          if(rs !=null) rs.close();
37          if(pstmt !=null) pstmt.close();
38          if(con !=null) con.close();
39       }catch(SQLException e){
40             e.printStackTrace();
41       }
42     }
43          return list;
44    }
45 }
```

* Presentation logic 샘플

```
..
05 public class Ex14_5{
06    public static void main( String [] args ) {
07       ExDAO dao = new ExDAO();
08       ArrayList<ExDTO> list = dao.select();
09       for (ExDTO dto : list) {
10            int deptno = dto.getDeptno();
11            String dname = dto.getDname();
12            String loc = dto.getLoc();
13            System.out.println(deptno+"₩t"+dname+"₩t"+loc);
14       }
15    }
16 }
```

❑ 요약

- JDBC 의 개요
- DAO (Data Access Object) 패턴
- DTO (Data Transfer Object) 패턴

Thank you