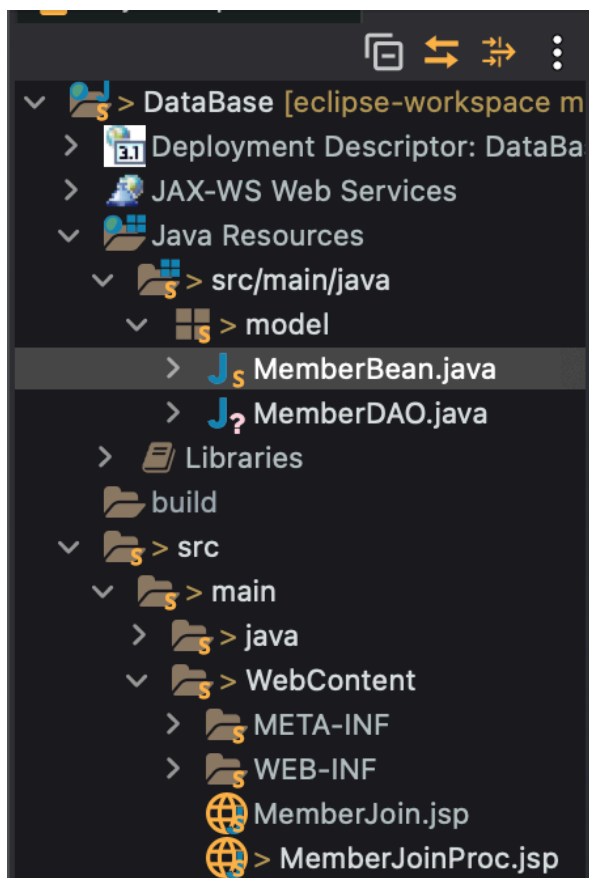


Data Access Object (DAO)

DB

- 데이터 처리를 위한 쿼리를 jsp 파일 내에서 작성하고 사용한다면 HTML 소스를 위한 공간이 부족하고 가독성도 떨어질 수 밖에 없다.
- 이를 해결하기 위해 jsp는 DAO 패턴을 사용하는데 이는 모델 2 방식과도 같다.
- java bean을 작성했던 공간에 DAO를 위한 새로운 클래스를 작성하는것으로 시작
- 클래스의 멤버는 공통으로 사용 되는 속성들을 지정하고, 데이터 입력, 수정, 삭제 등 자주 사용 되는 기능들은 메소드로 만들어준다.



```

// 오라클 데이터 베이스에 연결 하고 select, insert, update, delete 작업을 실행 해 주는 클래스
public class MemberDAO {

    // oracle에 접속하는 소스를 작성
    String id = "ONDD";
    String pass = "1234";
    String url = "jdbc:oracle:thin:@localhost:1521:XE"; // 접속 url

    Connection con; // 데이터베이스에 접근 할 수 있도록 설정 해 주는 객체
    PreparedStatement pstmt; // 데이터 베이스에서 쿼리를 실행시켜주는 객체
    ResultSet rs; // 데이터베이스의 테이블의 결과를 리턴받아 자바에 저장해주는 객체

    // 데이터 베이스에 접근할 수 있도록 도와주는 메소드 생성
    public void getCon() {

        try {
            // 1. 해당 데이터 베이스를 사용한다고 선언(클래스를 등록 = 오라클용을 사용)
            Class.forName("oracle.jdbc.driver.OracleDriver");
            // 2. 해당 데이터 베이스에 접속
            con = DriverManager.getConnection(url, id, pass);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

// 데이터 베이스에 접근할 수 있도록 도와주는 메소드 생성
public void getCon() {

    try {
        // 1. 해당 데이터 베이스를 사용한다고 선언(클래스를 등록 = 오라클을 사용)
        Class.forName("oracle.jdbc.driver.OracleDriver");
        // 2. 해당 데이터 베이스에 접속
        con = DriverManager.getConnection(url,id,pass);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

// 데이터베이스에 한 사람의 회원 정보를 저장해주는 메소드
public void insertMember(MemberBean mbean) {

    try{
        getCon();
        // 3.접속 후 쿼리 준비하여
        String sql = "insert into member values(?,?,?,?,?,?,?,?)";
        // 4.쿼리를 사용하도록 설정
        PreparedStatement pstmt = con.prepareStatement(sql); //jsp에서 쿼리를 사용하도록 설정
        // 물음표에 맞게 데이터를 맵핑
        pstmt.setString(1, mbean.getId());
        pstmt.setString(2, mbean.getPass1());
        pstmt.setString(3, mbean.getEmail());
        pstmt.setString(4, mbean.getTel());
        pstmt.setString(5, mbean.getHobby());
        pstmt.setString(6, mbean.getJob());
        pstmt.setString(7, mbean.getAge());
        pstmt.setString(8, mbean.getInfo());
        // 5.오라클에서 쿼리를 실행하시오.
        pstmt.executeUpdate(); // insert나 update, delete시 사용하는 메소드

        // 6.자원 반납
        con.close();
    }catch(Exception e){
        e.printStackTrace();
    }
}

```

- insert 메소드의 인자는 use bean을 통해 받아온 요소로 처리한다.
- DAO 패턴이 적용 된 jsp 파일

```

15 ◉ <%
16     request.setCharacterEncoding("UTF-8");
17
18 // 취미 부분은 별도로 읽어드려 다시 빈클래스에 저장
19     String[] hobby = request.getParameterValues("hobby");
20 // 배열에 있는 내용을 하나의 스트링으로 저장
21     String textHobby = "";
22
23     for(int i = 0; i < hobby.length; i++){
24         textHobby += hobby[i] + " ";
25     }
26
27 %>
28 <!-- USEBEAN을 이용해 한꺼번에 데이터 받아오기 -->
29 ◉ <jsp:useBean id="mbean" class="model.MemberBean">
30     <!-- 맵핑 -->
31     <jsp:setProperty name="mbean" property="*" />
32 </jsp:useBean>
33
34 ◉ <%
35 // 기존 취미는 주소 번지가 저장 되기 때문에 위에 배열의 내용을 하나의 스트링으로 저장한 변수를
36 // 다시 입력 해준다.
37     mbean.setHobby(textHobby);
38
39     //데이터 베이스 클래스 객체 생성
40     MemberDAO mdao = new MemberDAO();
41     mdao.insertMember(mbean);
42 %>
43
44     오라클 완료 ~
45

```

#DataBase