



웹프로그래밍실무

JAVA BEAN을 이용한
회원관리시스템

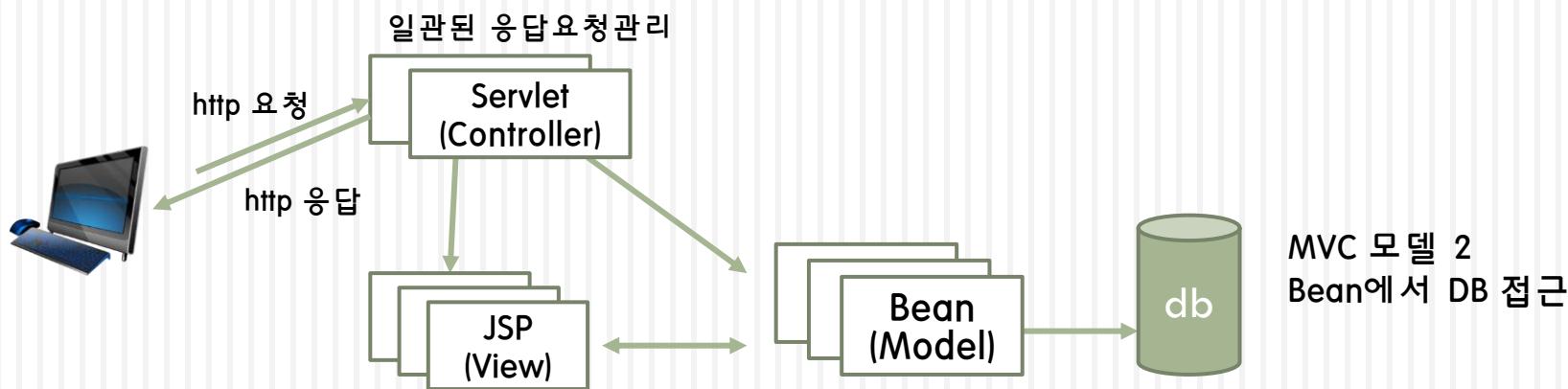
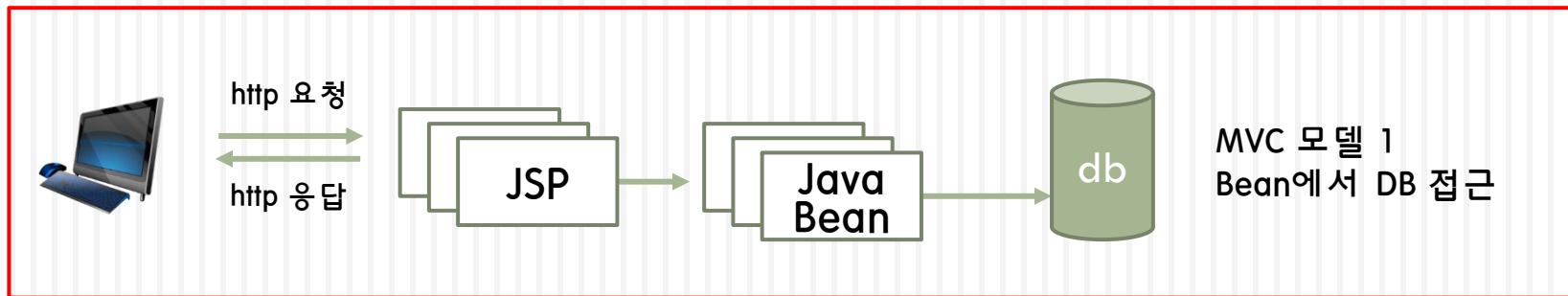
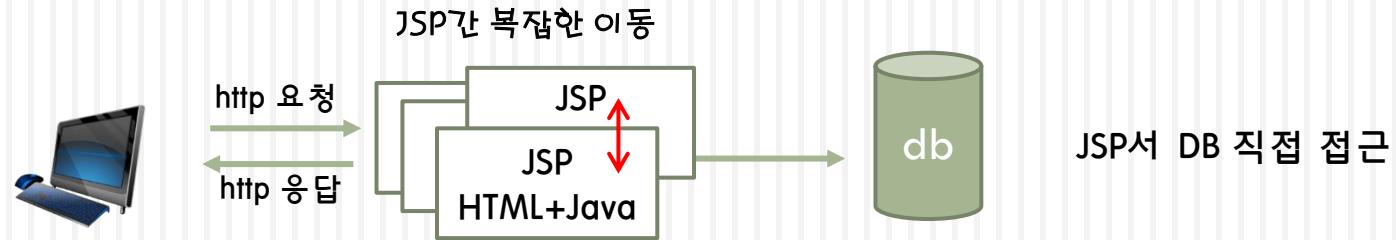
장안대학교
인터넷정보통신과

Java Beans



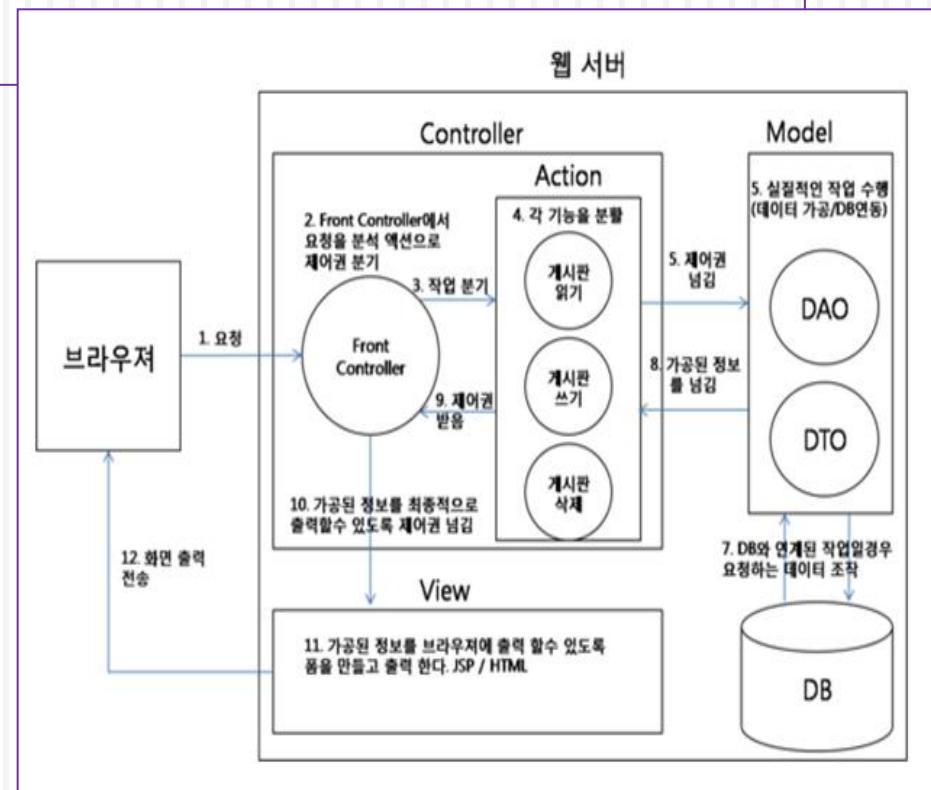
- ▣ 컴퍼넌트 Component :
 - 프로그램을 구성하는 요소, 여러 프로그램에 재사용 가능
- ▣ JavaBeans
 - 빌더형식의 개발도구에서 시각적으로 조작이 가능하고 재사용이 가능한 소프트웨어 컴포넌트
 - JavaBeans 는 Java 로 Software Component 를 만들기 위한 기술
 - Java 로 작성된 소프트웨어 Component : JavaBeans.
 - JavaBeans 는 Sun MicroSystems 에서 나온 객체지향 프로그래밍 인터페이스
- ▣

MVC 모델



MVC Model 2

- **Model**: Business Logic과 데이터베이스 입출력을 담당하는 Layer. View와는 독립적
 - 데이터를 처리하는 일반 Bean 클래스
- **View** : 어플리케이션의 유저 인터페이스만 담당 최소한의 코드를 이용하여 단순하게 작성
 - JSP
- **Controller** : 어플리케이션의 Flow를 담당
 - View와 Business Logic의 중재자 역할
 - 사용자의 특정 서비스 요청을 받아서, 해당 로직에 연결
 - 세션 관리, 권한관리 등의 기능 담당
 - Front Controller



MVC 모델

▣ JSP Model 1 - 장점

- 기본적인 HTML, JavaScript, JAVA만 알아도 배우기 쉽다.
- 객체지향 원리를 잘 몰라도 기능을 구현 할 수 있다.
- 처리 로직이 jsp 페이지에 JAVA 코드로 구현 되어있다.
-

▣ JSP Model 1 – 단점

- 디자인(HTML, CSS, JavaScript)과 로직(JAVA)이 혼합되어 있어 소스 수정시 예상하지 못한 곳에서 많은 시간이 지연된다.
- 같은 JAVA코드가 여러 JSP페이지에 중복되어 있어 소스 수정이 어렵다.
- 권한 등의 구현 시 모든 jsp페이지 상에 SSI등을 이용해 구현해야 한다.

▣ JSP Model 2 – 장점

- Client의 요청을 받는 부분, 처리 로직 부분, 뷰 부분이 분리되어 있어 유지보수가 편리
- 모든 기능이 세분화해 구조를 이루고 있음으로 기능 확장이 용이
- 컨트롤러 서블릿에서 권한 및 인증 구현을 편리하게 할 수 있음

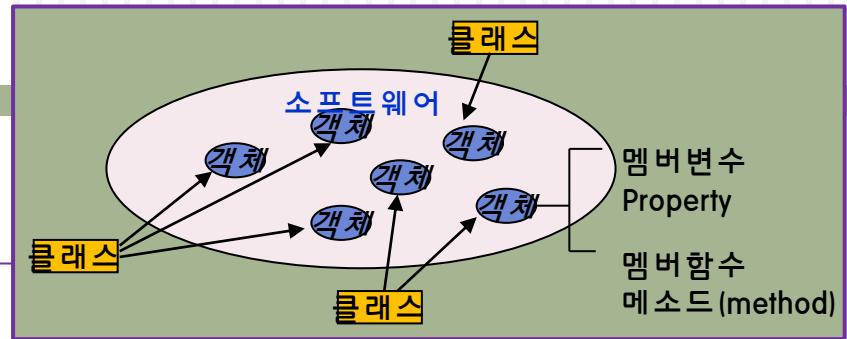
▣ JSP Model 2 – 단점

- 디자인 패턴등의 개념과 자바언어를 잘 알고 있어야 함.
- Application이 처리 파트별로 나뉘어져 있음으로 개발 작업의 양이 많음.
- 디버깅 작업을 처리 파트 별로 살펴야 함으로 어려운 부분이 있음.
- 개발자들이 MVC에 대한 개념을 가지고 있어야 효과를 볼 수 있음.

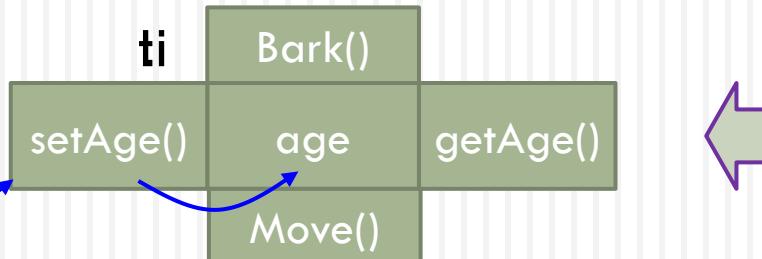
Bean 작성 규칙

Bean 작성 규칙

- 멤버변수(member variable)는 **private**로 선언한다.
- 멤버변수에 값을 설정하는 **setXxx()** method를 만들고, **public**으로 선언한다.
 - 멤버변수의 이름이 aaa이면 **setAaa()** 형태가 되어야 한다.
- 멤버변수에 값을 읽어오는 **getXxx()** method를 만들고, **public**으로 선언한다.
 - 멤버변수의 이름이 aaa이면 **getAaa()** 형태가 되어야 한다.
- 멤버변수가 boolean 형일 경우, **getXxx()** method 대신에 **isXxx()** method를 만든다.
- 기타 필요한 멤버변수와 method를 만든다.
- bean component는 반드시 **WEB-INF\classes**에 package 별로 저장되어야 한다.
- **생성자(Constructor)**는 parameter가 없어야 한다. 즉 default constructor 여야 한다.
- **property** : bean의 멤버변수 중에서 **setXxx()** 또는 **getXxx()** method가 선언되어 있어서 외부에서 참조할 수 있는 멤버변수



Bean(클래스)의 인스턴스



```
import com.Tiger;

public class maintest {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Tiger ti = new Tiger(); //생성자 호출과 객체
        ti.setAge(3);
        ti.bark(3); //다형성 - overloading
        ti.bark(); //다형성 - overriding
        ti.move();
    }
}
```

```
package com;

public class Tiger {

    private int age=0;
    public Tiger() {
    }

    public void bark()
    {
        System.out.println("고양이가");
    }

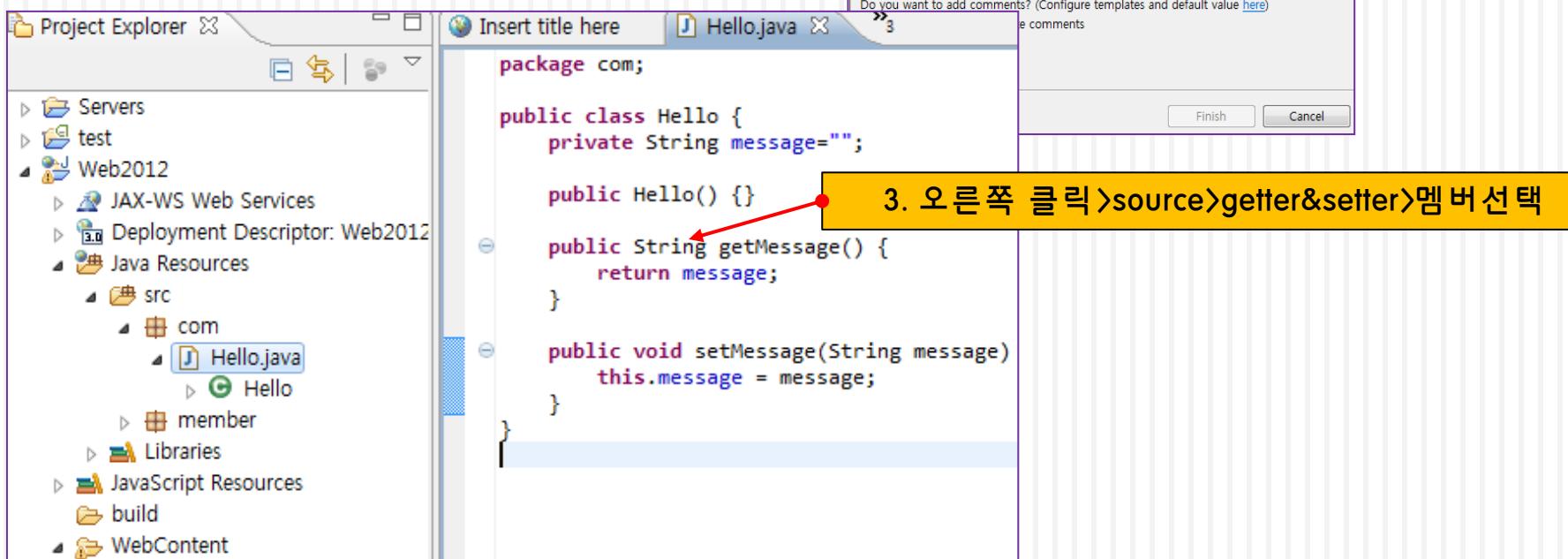
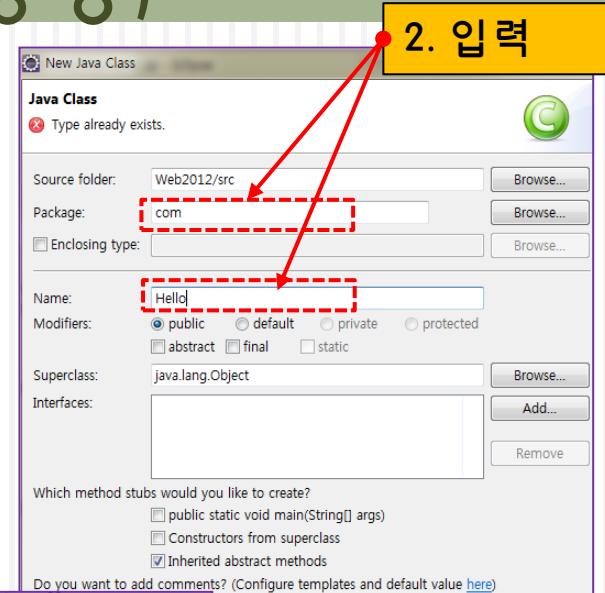
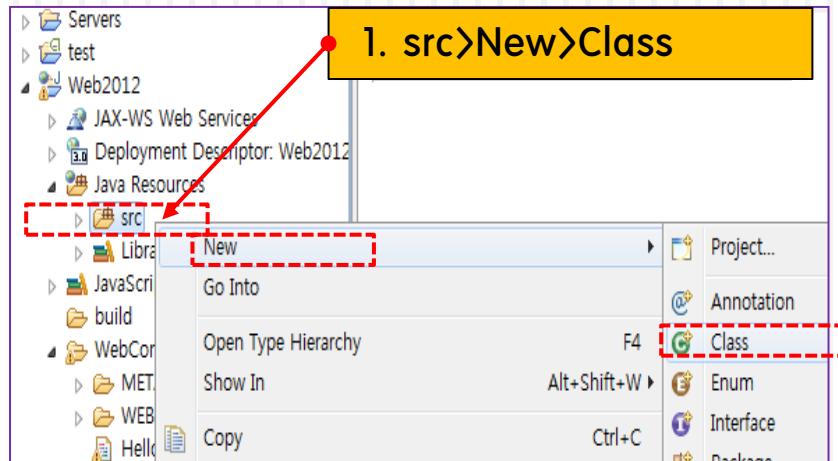
    public void move()
    {
        System.out.println("빠르고");
    }
    //캡슐화
    public int getAge(){
        return age;
    }

    //캡슐화
    public void setAge(int a){
        this.age=a;
    }
}
```

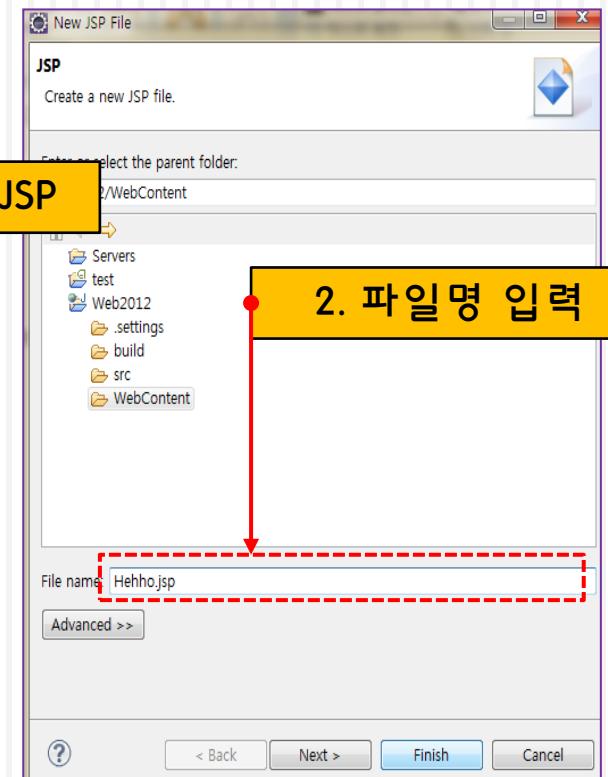
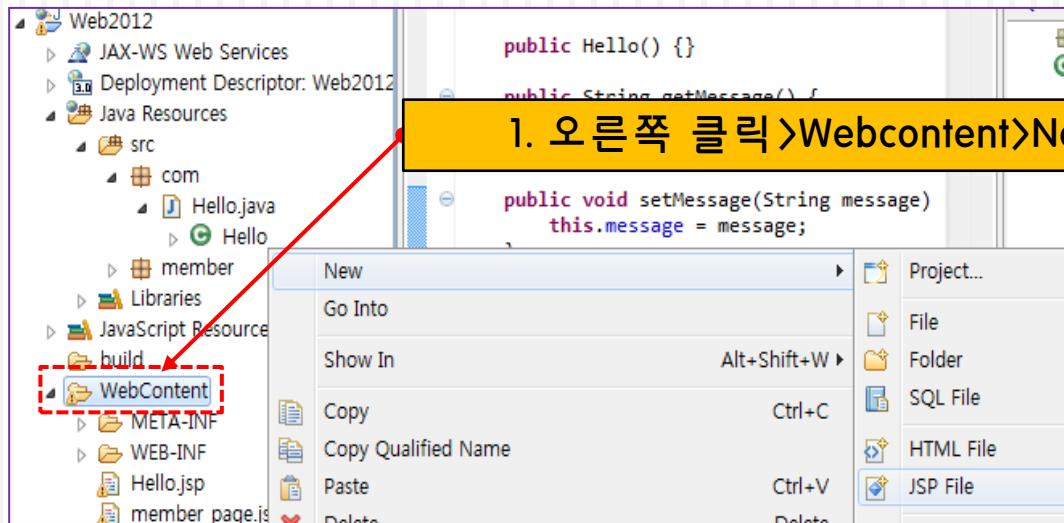
JSP에서 Javabean의 사용 -150p

- ▣ JSP에서 bean을 이용하기 위해서는 <jsp:useBean>, <jsp:setProperty>, <jsp:getProperty> 액션 태그를 활용해야 한다.
- ▣ <jsp:useBean id="name" class="package.Class" scope= "page" />
 - <jsp:useBean id="ti" class="com.Tiger" scope=page" />
 - com.Tiger ti = new com.Tiger();
- ▣ <jsp:getProperty name= "name" property= "property" >
 - <jsp:getProperty name= "ti" property= "age" >
 - ti.getAge();
- ▣ <jsp:setProperty name=" nameClass" property=" name" value=" John" />
 - <jsp:setProperty name="ti" property= "age" value=3 />
 - ti.setAge(3);

자바빈 연습 (bean의 생성)



Jsp 파일 생성



Java bean의 활용 연습

```
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>  
<jsp:useBean id="myBean" class="com.Hello" scope="page"></jsp:useBean>  
<jsp:setProperty name="myBean" property="message" value="My First JavaBean!!!" />  
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">  
<title>Insert title here</title>  
</head>  
<body>  
Bean 연습==> <jsp:getProperty name="myBean" property="message" />  
<hr width="350px" align="left" />  
<br/> <br/> <br/> <br/>  
  
클래스의 인스턴스에서 메서드를 직접 사용  
<hr width="300px" align="left" />  


```
<% myBean.setMessage("메서드에 직접 데이터 저장!!!!"); %>
두 번째 Bean ==> <%= myBean.getMessage() %>
```

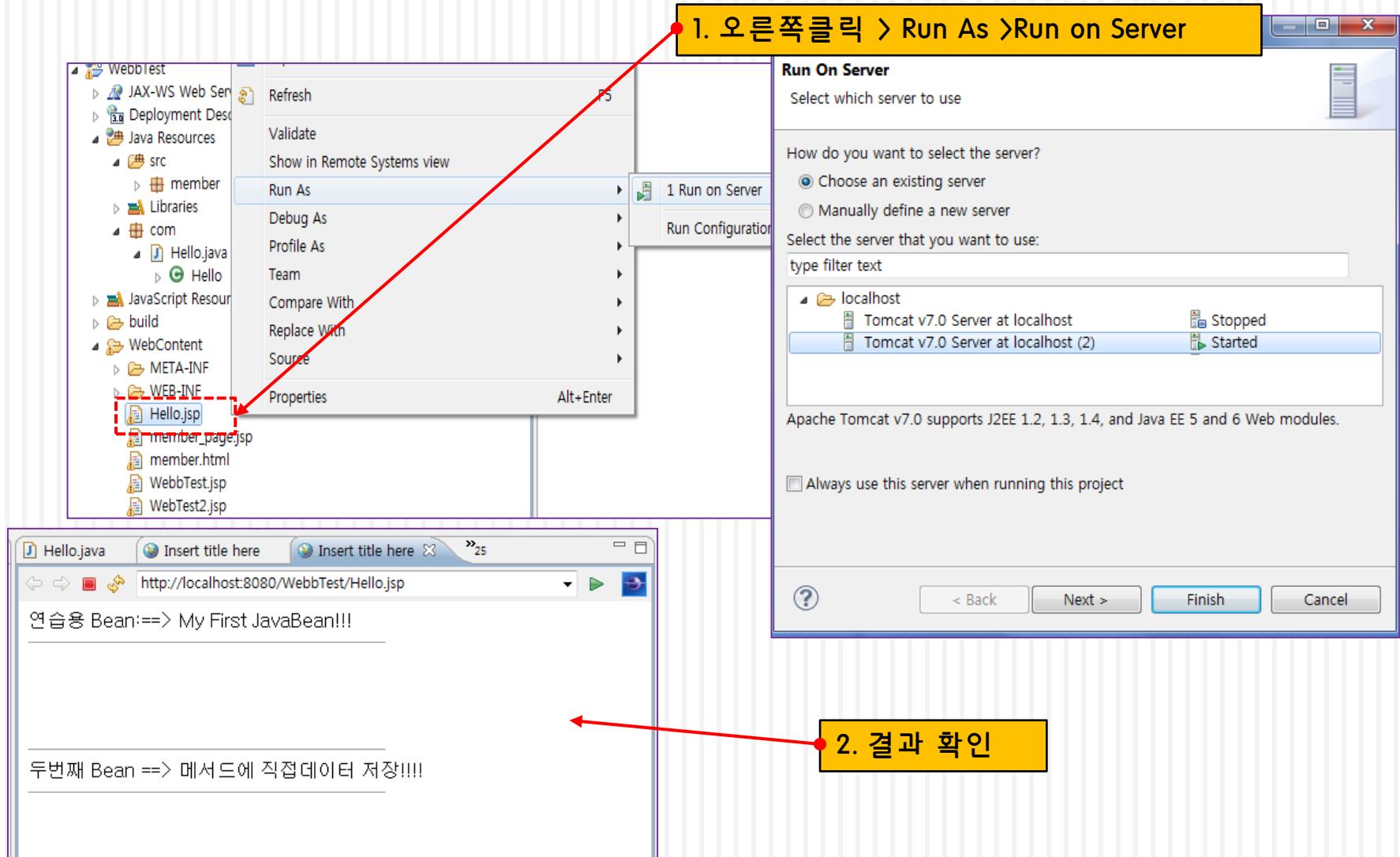
  
<hr width="350px" align="left" />  
</body>  
</html>
```

com.Hello myBean = new com.Hello();
myBean.setMessage(" My First JavaBean");

액션태그를 이용한 getMessage() 메서드 사용
myBean.getMessage();

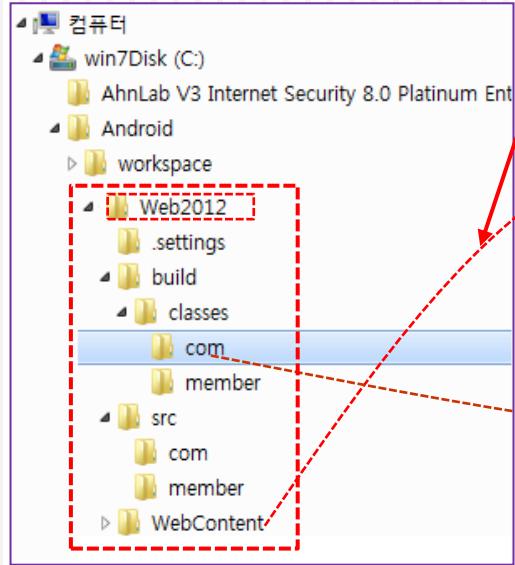
인스턴스를 이용한 메서드 직접 사용

실행



웹서버로 이전 서비스 개시

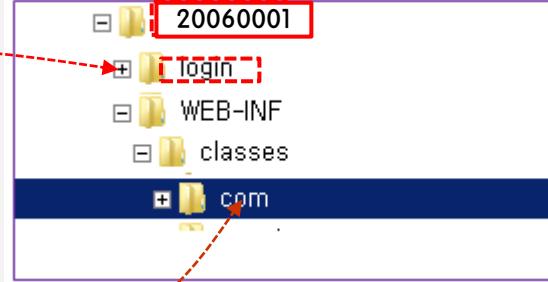
PC의 WebProject 폴더
(eclipse workspace or 기타 위치)



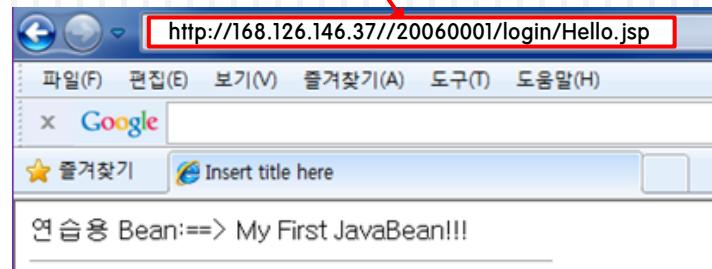
1. WebContents
폴더의 모든 *.jsp
파일이나 폴더를
서버의 login
폴더로 복사
(예:hello.jsp)

2. 폴더를 그대로 복사하거나
서버에 폴더 이름을 동일하게
만들고 폴더 안의 파일을
(예:Hello.class) 복사

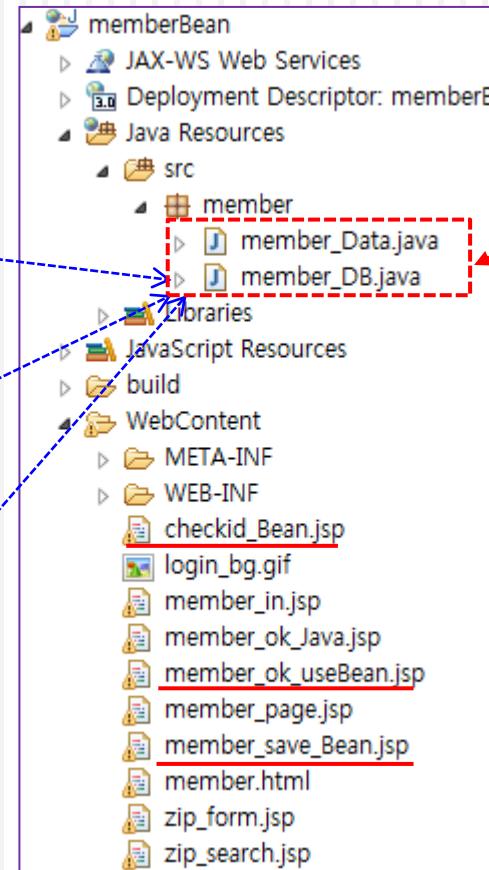
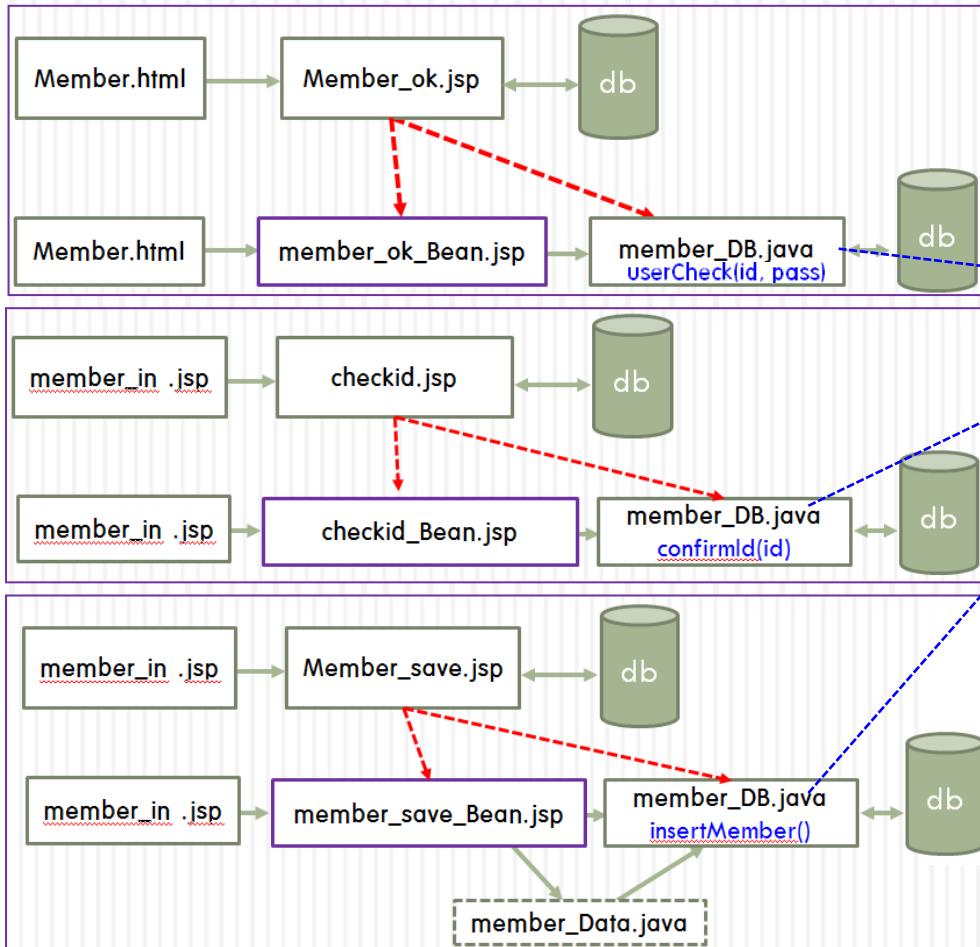
서버의 학생홈(예:2006001) 폴더 아래의
login 폴더 부터 시작할
경우 (168.126.146.37)



3. IE에서 서비스 확인

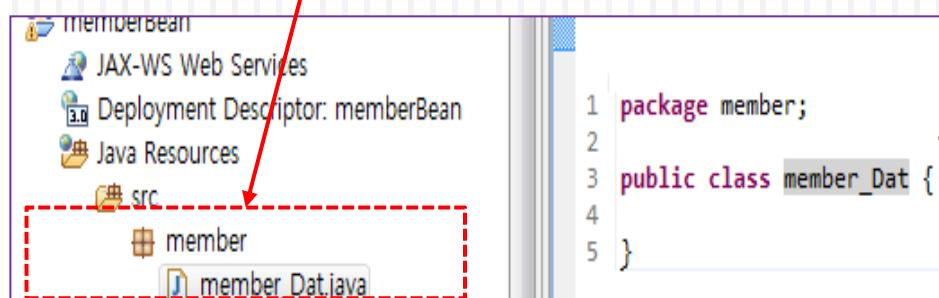
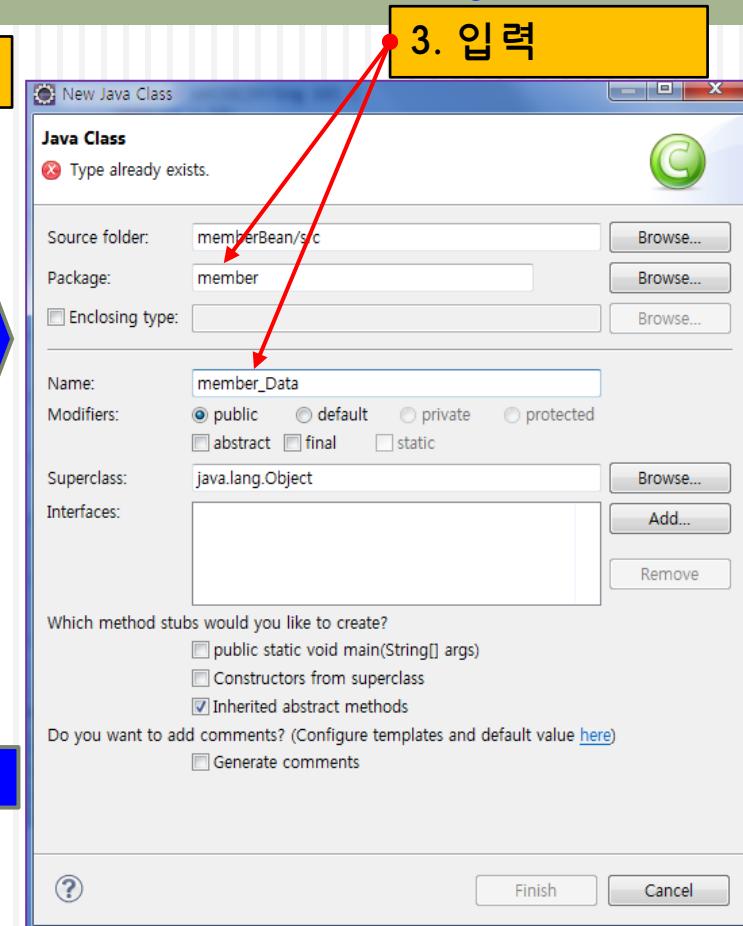
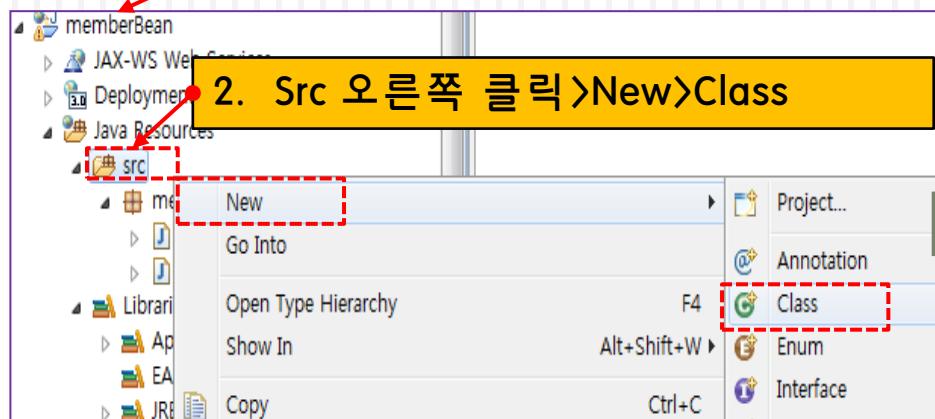


Java Bean을 이용한 회원관리



정리
모든 파일
를 나누기
위해서 Model
을 떼어
내는 것

회원테이블 데이터 bean의 작성 (memebn_Data.java)



회원 테이블 Gatter & setter 설정 (memebr_Data.java)

1. 테이블 컬럼 멤버변수 입력

```
package member;
```

```
public class member_Data {
```

```
    private String id;  
    private String pass;  
    private String name;  
    private String jumin;  
    private String zip;  
    private String addr1;  
    private String addr2;  
    private String phone;  
    private String email;
```

```
}
```

4. 결과 확인

The screenshot shows the 'Generate Getters and Setters' dialog box overlaid on a database table definition and the generated Java code.

Database Table: JNESQLJNEDB1 - jjin.member_tbl

| 열 이름 | 데이터 형식 | Null 허용 |
|-------|-------------|-------------------------------------|
| no | int | <input type="checkbox"/> |
| id | varchar(10) | <input type="checkbox"/> |
| pass | varchar(10) | <input type="checkbox"/> |
| name | varchar(20) | <input type="checkbox"/> |
| jumin | char(13) | <input type="checkbox"/> |
| zip | char(7) | <input checked="" type="checkbox"/> |
| addr1 | | |
| addr2 | | |
| phone | | |
| email | | |

Dialog Box: Select getters and setters to create:

- > addr1
- > addr2
- > email
- > id
- > jumin
- > name
- > pass
- > phone
- > zip

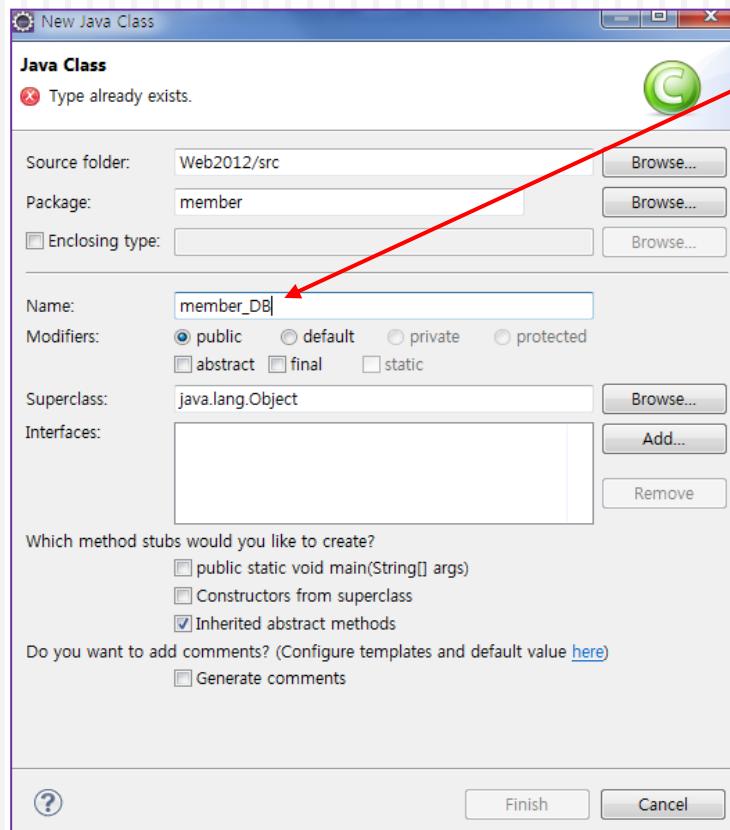
Buttons: Select All, Deselect All, Select Getters, Select Setters

Generated Java Code:

```
package member;  
  
public class member_Data {  
    String id;  
    String pass;  
    String name;  
    String jumin;  
    String zip;  
    String addr1;  
    String addr2;  
    String phone;  
    String email;  
  
    public String getId() {  
        return id;  
    }  
    public void setId(String id) {  
        this.id = id;  
    }  
    public String getPass() {  
        return pass;  
    }  
}
```

데이터 관리 빈 클래스 생성(member_DB.java)

1. Src >New>Class



2. 입력

3. 확인

The screenshot shows the Eclipse IDE interface. On the left, the package structure is displayed under 'memberBean':

- JAX-WS Web Services
- Deployment Descriptor: memberBean
- Java Resources
 - src
 - member
 - member_Data.java
 - member_DB.java
 - Libraries

On the right, the content of the 'member_DB.java' file is shown:

```
1 package member;
2
3 import java.sql.*;
4
5
6 public class member_DB {
7
8
9 }
10
11
12 }
```

생성자와 connection 객체 생성 메서드 member_DB.java

```
package member;

import java.sql.*;
import java.sql.DriverManager;

public class member_DB {
    private static member_DB instance = new member_DB();

    public static member_DB getInstance(){
        return instance;
    }
    public member_DB(){}
    private static Connection getConnection() throws Exception{
        Connection con = null;
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        con = DriverManager.getConnection("jdbc:sqlserver://168.126.146.35:1433;DatabaseName=JNEDB1","jjin","jjinpang");
        return con;
    }
    ... 추가 메서드 입력
}
```

블록 > CTRL+SHIFT + O

The UML class diagram illustrates the `member_DB` class with the following methods:

- `getConnection()`
- `userCheck()`
- `confirmID()`
- `insertMember()`
- `getIDSearch()`
- ...

Method `A()` contains a try-catch block where `Method_B()` is called. If an exception occurs in `Method_B()`, it is caught and handled within the same block.

Method `B()` throws an `Exception`.

Method `C()` also throws an `Exception`.

Annotations in the diagram include: "예외 발생..." (Exception occurs...) pointing to the catch block, and "<< 예외 처리를 포함한 메서드로 전달 >>" (<< Transfer to a method including exception handling >>) pointing to the call from Method A to Method B.

로그인 메서드 member_DB.java

//로그인 시 ID, Password 체크하는 코드

```
public static int userCheck(String me_id, String me_pass) throws Exception
{
    Connection con = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    String dbpass = "";
    int x = -1;

    try {
        con = getConnection();
        pstmt = con.prepareStatement("SELECT pass FROM member_tbl WHERE id = ?");
        pstmt.setString(1, me_id);

        rs = pstmt.executeQuery();

        if (rs.next()) {
            dbpass = rs.getString("pass");
            if (dbpass.equals(me_pass)) {
                x = 1; //아이디와 비번이 일치할 때
            }
            else {
                x = 0; //아이디는 일치하고 비번이 일치하지 않을 때
            }
        }
        else {
            x = -1; //아이디가 없을 때
        }
    }
```

```
private static Connection getConnection() throws Exception{
    Connection con = null;
    Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
    con = DriverManager.getConnection("jdbc:sqlserver://168.126.146");
    return con;
}
```

```
}
catch (Exception e){}
finally {
    if (rs != null) {
        try{ rs.close(); }
        catch(SQLException se){}
    }
    if (pstmt != null) {
        try{ pstmt.close(); }
        catch(SQLException se){}
    }
    if (con != null) {
        try{ con.close(); }
        catch(SQLException se){}
    }
}
return x;
}
```

아이디 중복체크 – id를 매개변수로 - member_DB.java

```
public static int confirmId(String id) throws Exception{
    Connection con = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    //String dbpasswd="";
    int x = -1;

    try{
        con = getConnection();
        pstmt = con.prepareStatement("select id from member_tbl where id=?");
        pstmt.setString(1, id);
        rs= pstmt.executeQuery();

        if(rs.next()){
            x=1;      //해당아이디 있음
        }
        else{
            x=-1;    //해당아이디 없음
        }

    }catch(Exception ex){
    }finally{
        if(rs != null){
            try{ rs.close(); }
            catch(SQLException ex){}
        }
        if(pstmt != null){
            try{ pstmt.close(); }
            catch(SQLException ex){}
        }
        if(con != null){
            try{ con.close(); }
            catch(SQLException ex){}
        }
    }
    return x;
}
```

회원 등록 -데이터 빈 객체를 매개변수로- member_DB.java

```
public static int insertMember(member_Data member) throws Exception {
    Connection con = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    String sql = null;
    int x = -1; //등록 실패

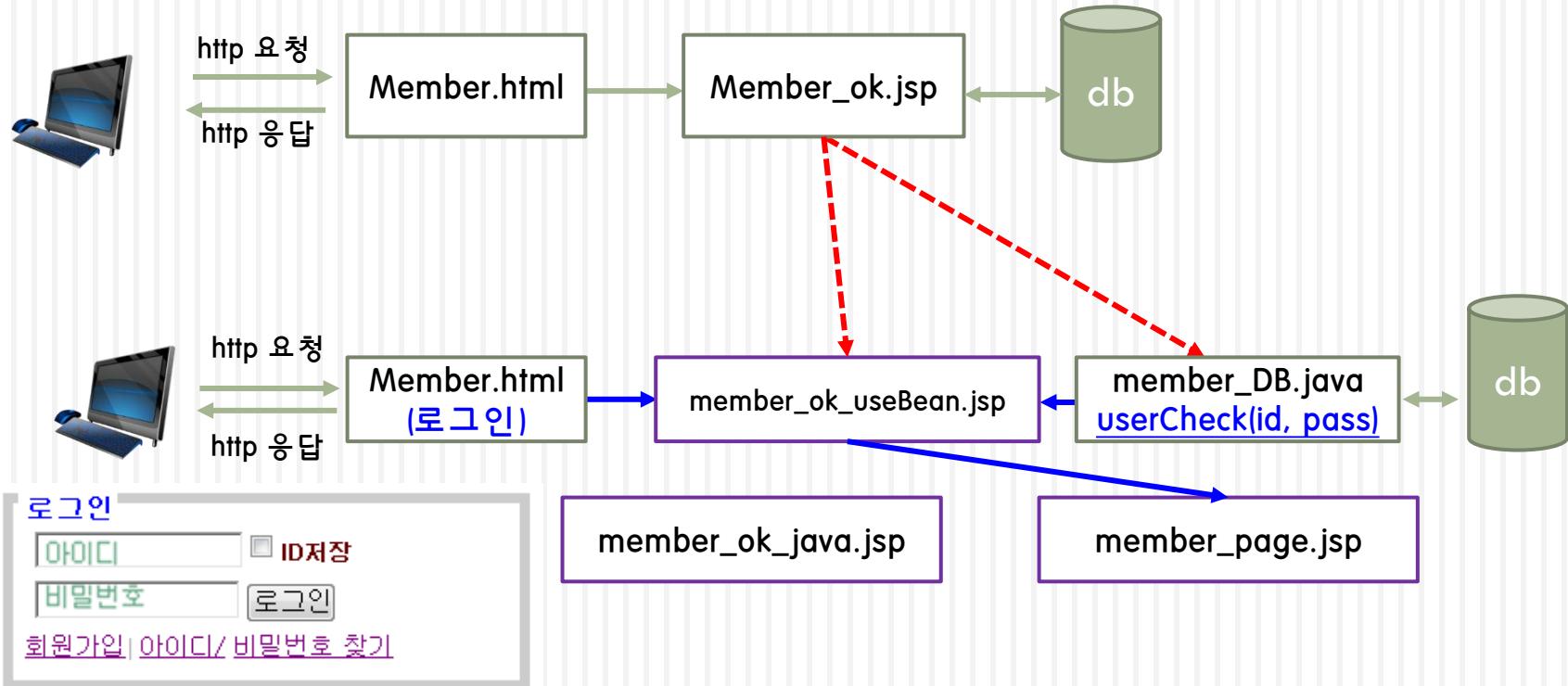
    try{
        con = getConnection();
        sql = "insert into member_tbl (id, pass, name, jumin, zip, addr1, addr2, phone,
              email) values(?,?,?,?,?,?,?,?,?)";
        pstmt = con.prepareStatement(sql);

        pstmt.setString(1, member.getId());
        pstmt.setString(2, member.getPass());
        pstmt.setString(3, member.getName());
        pstmt.setString(4, member.getJumin());
        pstmt.setString(5, member.getZip());
        pstmt.setString(6, member.getAddr1());
        pstmt.setString(7, member.getAddr2());
        pstmt.setString(8, member.getPhone());
        pstmt.setString(9, member.getEmail());

        pstmt.executeUpdate();
        x = 1; // 등록 성공
    } //try

    catch (Exception e){ e.printStackTrace(); }
    finally {
        if (rs != null) {
            try{ rs.close(); }
            catch(SQLException se){}
        }
        if (pstmt != null) {
            try{ pstmt.close(); }
            catch(SQLException se){}
        }
        if (con != null) {
            try{ con.close(); }
            catch(SQLException se){}
        }
    }
    return x;
}
```

자바빈을 활용한 회원인증



자바빈을 활용한 회원인증 member_ok_useBean.jsp

● Webcontents > New > JSP file > member_ok_useBean.jsp 입력

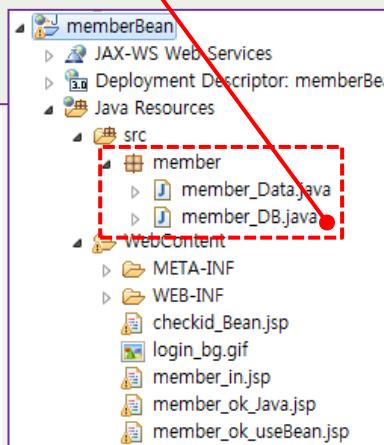
```
<%@ page contentType="text/html; charset=utf-8" %>
<% request.setCharacterEncoding("utf-8"); %>

<%
String id = request.getParameter("txtID");
String pass = request.getParameter("pass");
String sessionID = "yes";

int check = 0;
%>

<jsp:useBean id="mem" class="member.member_DB" />

<%
check = mem.userCheck(id, pass);
```



```
if(check == 1){ //아이디와 비밀번호 일치
    session.setAttribute("user", sessionID);
    response.sendRedirect("member_page.jsp");
}

else if(check == 0){
%>
<script language="JavaScript">
alert("비밀번호가 일치하지 않습니다");
history.go(-1);
</script>

<% } else if(check == -1){ %>

<script language="JavaScript">
alert("아이디가 일치하지 않습니다");
history.go(-1);
</script>

<% } %>
```

자바 클래스 직접 호출을 이용한 회원인증

● Webcontents > New > JSP file > member_ok_Java.jsp 입력

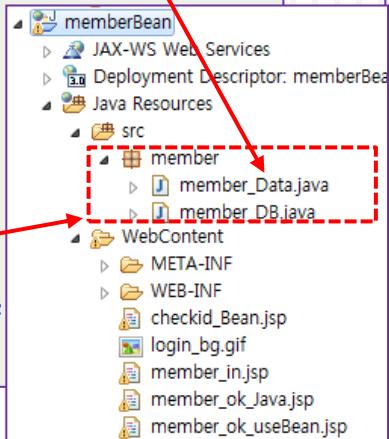
```
<%@ page contentType="text/html; charset= utf-8" %>
<%@ page import="member.member_DB" %>
<%@ page import="member.member_Data" %>
<%@ page import="java.util.*" %>

<% request.setCharacterEncoding("utf-8"); %>

<%
String id = request.getParameter("txtID");
String pass = request.getParameter("pass");
String sessionId = "yes";

int check = 0;

member_DB mem = member_DB.getInstance();
check = mem.userCheck( id, pass );
```



```
if(check == 1){ //아이디와 비밀번호 일치
    session.setAttribute("user",sessionId);
    response.sendRedirect("member_page.jsp");
}

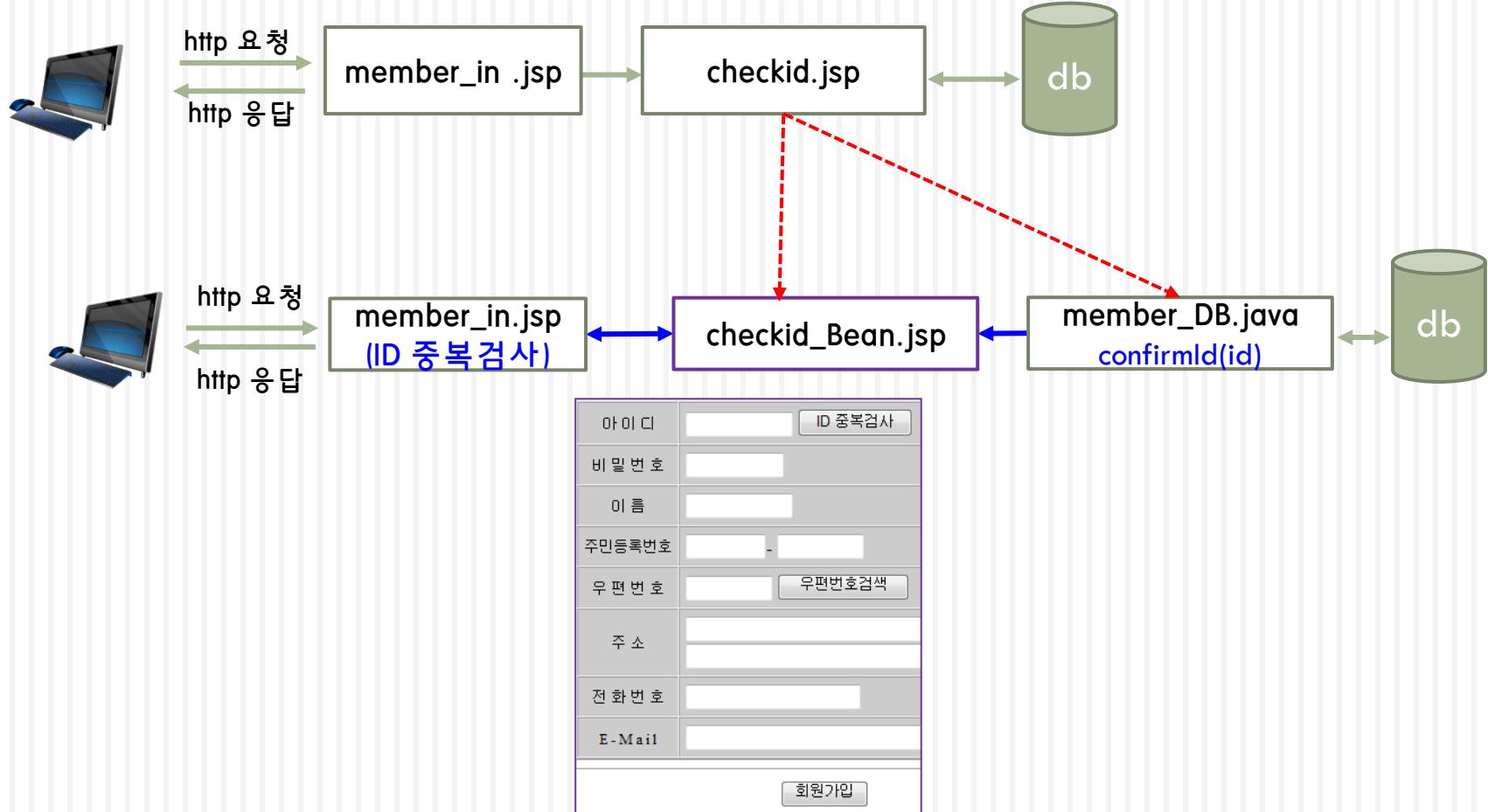
else if(check == 0){
%>
<script language= "JavaScript">
    alert("비밀번호가 일치하지 않습니다");
    history.go(-1);
</script>

<% }else if(check == -1){ %>

<script language= "JavaScript">
    alert("아이디가 일치하지 않습니다");
    history.go(-1);
</script>

<% } %>
```

id 중복체크



Id 중복 체크

● Webcontents>New>JSP file> checkid_Bean.jsp 입력

```
<%@ page contentType = "text/html; charset=utf-8" %>
<%@ page import = "java.sql.*" %>

<HTML>
<HEAD>
<TITLE> 아이디 중복 체크 </TITLE>
<META http-equiv= "Content-Type" content="text/html; charset=utf-8">
<style type='text/css'>
    a:link {font-family:"";color:black;text-decoration:none;}
    a:visited {font-family:"";color:black;text-decoration:none;}
    a:hover {font-family:"";color:black;text-decoration:underline;}
</style>

<SCRIPT language="JavaScript">
function strData(reid, txtID) {
    opener.document.joinForm.re_id.value= reid;
    opener.document.joinForm.id.value= txtID;
    window.close();
}
</SCRIPT>
</HEAD>

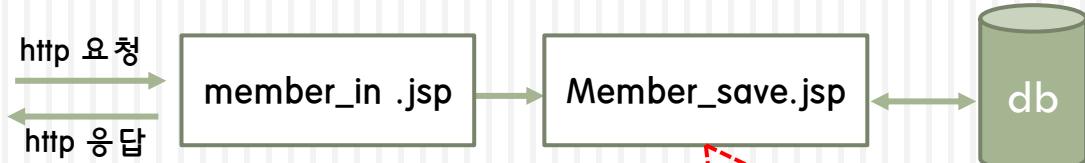
<BODY>
<TABLE border='1' width=250>
<%
String txtID = request.getParameter("id");
String reid="";
int check = 0;
%>
```

```
<jsp:useBean id="mem" class="member.member_DB" />

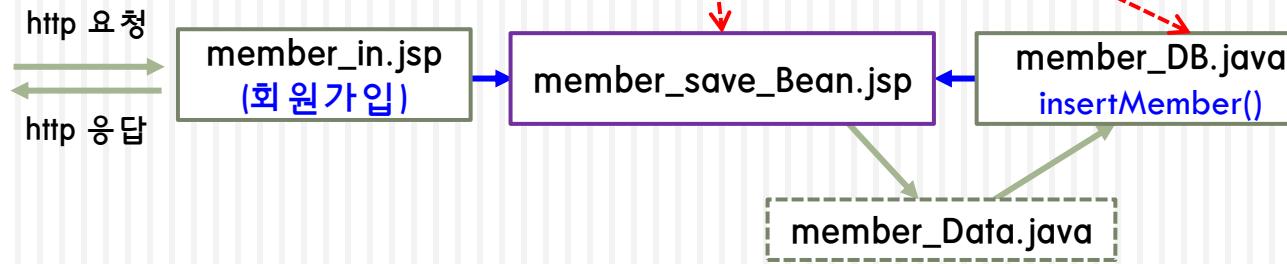
<%
check = mem.confirmId(txtID);

if(check == -1){ //사용 가능한 아이디
%>
<TR>
<TD align='center' bgcolor='cccccc'>
<font size='2'>ID : <%=txtID%> <BR> 사용 가능한 아이디입니다.</font>
<% reid = txtID; %>
</TD>
</TR>
<TR>
<TD align='center'>
<a href=javascript:strData('<%= reid %>','<%=txtID%>')>[닫기]</a></TD>
</TR>
<%
}
else {
%>
<TR>
<TD align='center' bgcolor='cccccc'>
<font size='2'>ID : <%=txtID%> <BR> 사용할 수 없는 아이디입니다.</font>
<% reid= "";txtID= "";%>
</TD>
</TR>
<TR>
<TD align='center'>
<a href=javascript:strData('<%= reid %>','<%=txtID%>')>[닫기]</a>
</TD>
</TR>
<%
}
%>
</TABLE>
</BODY>
</HTML>
```

회원가입



| | | |
|-------------------------------------|---|--|
| 아이디 | <input type="text"/> | <input type="button" value="ID 중복검사"/> |
| 비밀번호 | <input type="password"/> | |
| 이름 | <input type="text"/> | |
| 주민등록번호 | <input type="text"/> - <input type="text"/> | |
| 우편번호 | <input type="text"/> | <input type="button" value="우편번호검색"/> |
| 주소 | <input type="text"/> | |
| 전화번호 | <input type="text"/> | |
| E-Mail | <input type="text"/> | |
| <input type="button" value="회원가입"/> | | |



회원가입

● Webcontents>New>JSP file> member_save_Bean.jsp 입력

```

1 <%@ page contentType = "text/html; charset=utf-8" %>
2 <%@ page import = "java.sql.*" %>
3 <% request.setCharacterEncoding("utf-8"); %>
4
5<HTML>
6<HEAD>
7 <TITLE> 회원가입 </TITLE>
8 <META http-equiv="Content-Type" content="text/html; charset=utf-8">
9<style type='text/css'>
10    a:link      {font-family:"";color:black;text-decoration:none;}
11    a:visited   {font-family:"";color:black;text-decoration:none;}
12    a:hover     {font-family:"";color:black;text-decoration:underline;}
13 </style>
14 </HEAD>
15
16
17<%
18 String juminnum1 = request.getParameter("jumin1");
19 String juminnum2 = request.getParameter("jumin2");
20
21 String jumin_num = juminnum1 + juminnum2; |
22
23%>
24 <jsp:useBean id="mem" class="member.member_DB" />
25 <jsp:useBean id="memData" class="member.member_Data" />
26
27 <jsp:setProperty name="memData" property="*" />
28<%
29     memData.setJumin(jumin_num);
30
31     int check = mem.insertMember(memData);
32
33     if(check==1) {
34%>
```

| | | |
|-------------------------------------|---|---------|
| 아이디 | <input type="text"/> | ID 중복검사 |
| 비밀번호 | <input type="password"/> | |
| 이름 | <input type="text"/> | |
| 주민등록번호 | <input type="text"/> - <input type="text"/> | |
| 우편번호 | <input type="text"/> | 우편번호검색 |
| 주소 | <input type="text"/> | |
| 전화번호 | <input type="text"/> | |
| E-Mail | <input type="text"/> | |
| <input type="button" value="회원가입"/> | | |

```

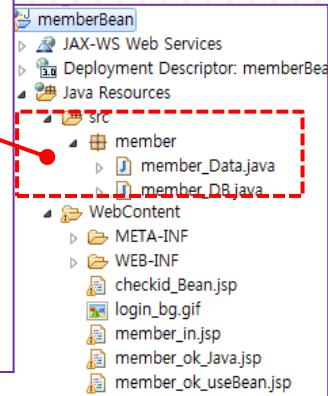
("id");
ter("pass");
ter("name");
parameter("jumin1");
parameter("jumin2");
er("zip");
rameter("addr1");
rameter("addr2");
eter("phone");
eter("email");
juminum2;
```

Name이 일치되는 것만 자동 set

```
<jsp:setProperty name="memData" property="*" />
```

```

public class member_Data {
    private String id;
    private String pass;
    private String name;
    private String jumin;
    private String zip;
    private String addr1;
    private String addr2;
    private String phone;
    private String email;
```



회원가입

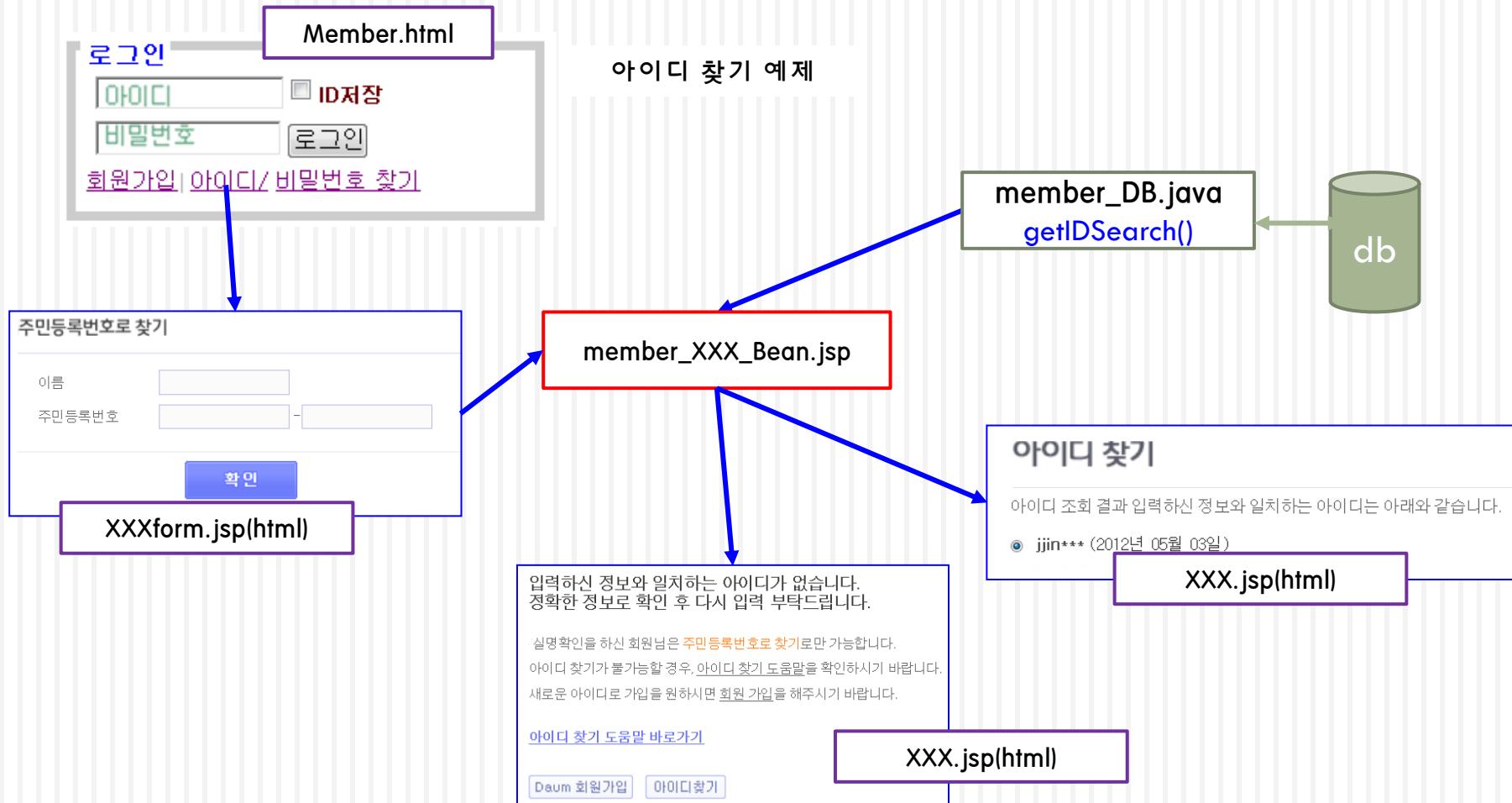
```
<BODY>
<center><font size='3'><b> 회원 가입 성공 </b></font>
<TABLE border='0' width='600' cellpadding='0' cellspacing='0'>
<TR><TD><hr size='1' noshade></TD> </TR>
</TABLE>

<TABLE cellSpacing='0' cellPadding='10' align='center' border='0'>
<TR>
<TD align='center'>
<font size='2'>회원 가입을 축하합니다.<BR>로그인 하세요.</font>
</TD>
</TR>
<TR>
<TD align='center'>
<font size='2'><a href="member.html">[로그인]</a></font>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
<%
    }
    else {
%>
```

```
<BODY>
<center><font size='3'><b> 회원 가입 실패 </b></font>
<TABLE border='0' width='600' cellpadding='0' cellspacing='0'>
<TR><TD><hr size='1' noshade></TD> </TR>
</TABLE>

<TABLE cellSpacing='0' cellPadding='10' align='center' border='0'>
<TR>
<TD align='center'>
<font size='2'>회원 가입에 실패 했습니다. <BR>다시 가입서를 작성해 주세요.</font>
</TD>
</TR>
<TR>
<TD align='center'>
<font size='2'><a href="member_in.jsp">[가입서 작성]</a></font>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
<% } %>
```

member_DB.java에 추가할 부분



주민번호와 이름으로 아이디 찾기 메서드 -

```
public static String getIdSearch(String me_name, String me_ssn) throws Exception{
    Connection con = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    String me_id = "-1";

    try{
        con = getConnection();

        pstmt = con.prepareStatement("select id from member_tbl where name = ? and jumin = ?");
        pstmt.setString(1, me_name);
        pstmt.setString(2, me_ssn);

        rs = pstmt.executeQuery();

        if(rs.next()){
            me_id = rs.getString("id");
        }
    }
    catch(Exception ex){
        ex.printStackTrace();
    }
    finally{
        if(rs != null){
            try{ rs.close(); }
            catch(SQLException ex){}
        }
        if(pstmt != null){
            try{ pstmt.close(); }
            catch(SQLException ex){}
        }
        if(con != null){
            try{ con.close(); }
            catch(SQLException ex){}
        }
    }
}
return me_id;
```

아이디와 주민번호로 패스워드 찾기 - member_DB.java

```
//아이디와 주민번호로 패스워드 찾기
public static String getPassSearch(String me_id, String me_ssn) throws Exception{
    Connection con = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    String me_pass = "-1";

    try{

        con = getConnection();
        pstmt = con.prepareStatement("select pass from member_tbl where id = ? and jumin = ?");

        pstmt.setString(1, me_id);
        pstmt.setString(2, me_ssn);

        rs = pstmt.executeQuery();

        if(rs.next()){
            me_pass = rs.getString("pass");
        }

    }

    ...

}

return me_pass;
}
```

아이디와 패스워드로 회원정보 불러오는 메서드 member_DB.java

```
public static member_Data getMember(String me_id, String me_pass ) throws Exception{  
    Connection con = null;  
    PreparedStatement pstmt = null;  
    ResultSet rs = null;  
  
    member_Data member = new member_Data();  
  
    try{  
  
    }  
    return member;  
}
```

회원 정보 업데이트 메서드 - member_DB.java

```
public static int modifyMember(member_Data member, String me_id ) throws Exception{
    Connection con = null;
    PreparedStatement pstmt = null;
    int x = -1;

    try {
        // 1. JDBC 드라이버를 로드합니다.
        Class.forName("oracle.jdbc.driver.OracleDriver");
        // 2. Connection을 생성합니다.
        con = DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:xe", "hr", "hr");
        // 3. Statement를 생성합니다.
        pstmt = con.prepareStatement("update member set name=?, address=? where id=?");
        pstmt.setString(1, member.getName());
        pstmt.setString(2, member.getAddress());
        pstmt.setString(3, me_id);
        // 4. SQL문을 실행합니다.
        x = pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        // 5. 자원 해제
        if (pstmt != null)
            try {	pstmt.close(); } catch (SQLException ex) {}
        if (con != null)
            try {	con.close(); } catch (SQLException ex) {}
    }
    return x;
}
```

이름과 주민번호로 회원 가입여부 판단 메서드 - member_DB.java

```
public static int joinCheck(String me_name, String me_jumin) throws Exception{
    Connection con = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    int x = -1;

    try {
        con = DBManager.getConnection();
        pstmt = con.prepareStatement("SELECT COUNT(*) FROM MEMBER WHERE NAME=? AND JUMIN=?");
        pstmt.setString(1, me_name);
        pstmt.setString(2, me_jumin);
        rs = pstmt.executeQuery();
        if(rs.next()) {
            x = rs.getInt(1);
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        DBManager.close(con, pstmt, rs);
    }
}

return x;
}
```