

## Chapter 08

# JSP 빈즈 ( JSP beans )

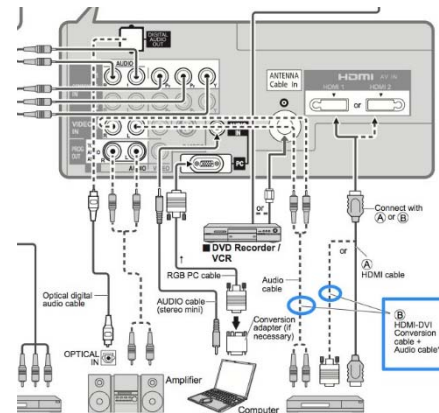
## JSP 빈즈의 개요

# JSP 빈즈의 개요

## ■ 빈즈(beans)

- 특정한 일을 독립적으로 수행하는 컴포넌트

- ▶ 컴포넌트는 각각 독립적인 기능이 있으며, 컴포넌트 조합을 통해 다양한 형태의 결과물을 만들 수 있음
- ▶ [예] 레고 블록이나 컴포넌트 오디오



- ▶ 원래 Java를 이용해 창이나 버튼, 스크롤바 등 화면을 구성하는 다양한 위젯을 작성하기 위해 빈즈를 제작함

# JSP 빈즈의 개요

## ■ JSP를 이용한 프로그래밍의 문제점

- JSP 문서 내부에는 HTML 코드를 외에 많은 JSP 코드를 포함할 수밖에 없음
  - ▶ 브라우저에 데이터를 출력하기 위한 문서를 작성할 경우 많은 JSP 코드를 포함하는 경우가 많음
    - 데이터베이스에 연결이 필요한 경우, 데이터베이스 접속을 위한 다수의 자바 API가 스크립트릿 형태로 포함되어야 함
    - 출력하고자 하는 모든 데이터는 자바 변수로 표현해야 함
- JSP 문서 내의 JSP 코드는 재사용이 불가능함
  - ▶ 다른 JSP 문서에서 재사용이 가능한 JSP 코드도 문서마다 별도로 작성해야 함
    - 예를 들어, 데이터베이스 접속을 위한 코드나 에러를 처리하기 위한 코드 등
- 클라이언트로부터 전달된 파라미터의 처리가 복잡할 수 있음
  - ▶ 클라이언트에서 전달된 파라미터의 수가 많을 경우 추출하고 처리해야 하는 JSP코드가 많아짐

# JSP 빈즈의 개요

## ■ JSP 빈즈

- JSP와 연동하기 위해 만들어진 컴포넌트 클래스
  - JSP 문서 내에 존재하는 JSP 코드(Java 코드)를 별도로 추출한 자바 클래스
- JSP의 표준 액션 태그로 접근할 수 있는 자바 클래스
  - 속성(멤버변수)과 값을 설정하는 메소드(setter), 값을 추출하는 메소드(getter)로 이루어져 있음
  - 일반적으로 문서들 사이에 공동으로 사용 가능하거나, 문서들 사이에 데이터를 전달하는 객체로 사용됨
  - 필요에 따라 메서드를 포함할 수도 있음
- JSP 빈즈는 하나의 JSP 문서에 종속되지 않음
  - 빈즈는 데이터는 여러 문서에서 공용으로 사용할 수 있음

# JSP 빈즈의 개요

## ■ JSP 빈즈의 장점

### • 폼 데이터 처리 용이

- ▶ 클라이언트로부터 전달받은 파라미터 처리에 용이함
  - 만약, 클라이언트로부터 20개의 파라미터가 전달될 경우 50번의 `request.getParameter()`를 통해 추출해야 함
- ▶ 자바빈즈를 사용하면 액션 태그를 한 번만 사용해서 모든 파라미터를 추출하여 빈즈에 객체에 저장할 수 있음

### • 문서들 사이의 데이터 공유 용이

- ▶ HTTP 프로토콜의 무상태 특성 때문에 상태 정보 유지 기술이 필요함
- ▶ 여러 페이지에서 사용해야 하는 데이터는 빈즈에 저장하여 공유할 수 있음

### • 컴포넌트 기반 구현 가능

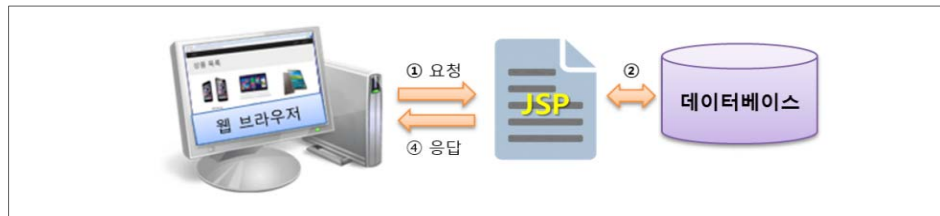
- ▶ JSP 페이지 내에서 필요한 비즈니스 로직을 컴포넌트로 만들어 활용할 수 있음
- ▶ 각 JSP 페이지에서 필요로 하는 기능은 컴포넌트를 재활용하여 웹 애플리케이션을 효과적으로 구현할 수 있음

# JSP 빈즈의 개요

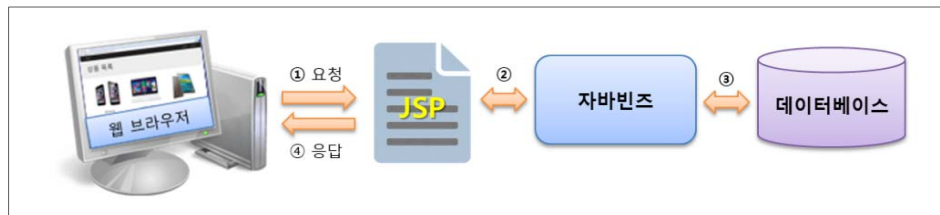
- 빈즈 내의 모든 데이터들이 멤버 메서드를 통해 접근하므로 데이터의 일관성 유지 가능

- 빈즈 내의 모든 데이터는 setter와 getter 메서드를 사용해 접근해야 함

- 빈즈를 사용하지 않는 경우



- 빈즈를 사용하는 경우



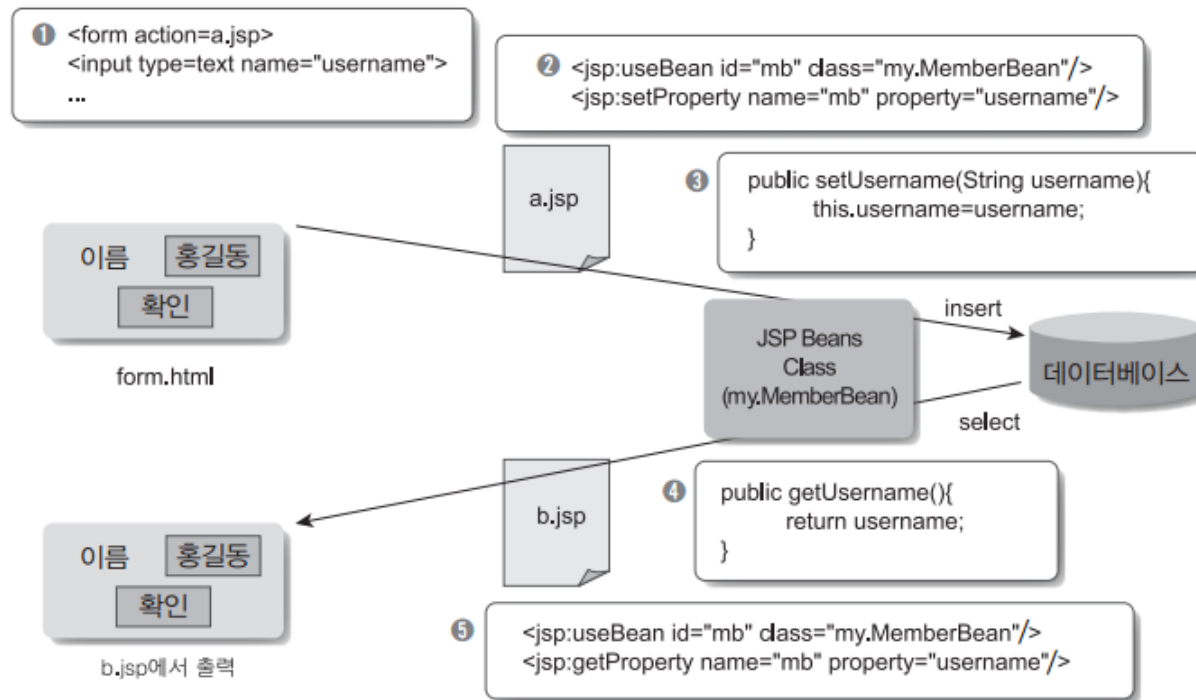
- 결론적으로 자바 빈즈는 데이터를 관리하기 위한 목적으로 생성된 자바 클래스

- 자바 빈즈를 사용하면 데이터를 객체 단위로 묶어서 처리할 수 있음

- 따라서 프레젠테이션 로직과 비즈니스 로직을 분리하기 위한 수단으로 사용 가능

# JSP 빈즈의 개요

## ■ 일반적인 구조에서 HTML, JSP, Beans의 상호작용 예



- ① HTML 폼 입력 값은 각각의 form 요소의 name 속성 값(value)을 통해 a.jsp로 전달됨
- ② a.jsp는 useBean 액션 태그로 빈즈를 불러오고 setProperty 메서드로 전달받은 값을 빈즈의 멤버 변수에 전달
- ③ 이때 빈즈의 setter 메서드들이 자동으로 호출됨
- ④ 빈즈의 값을 참조하려면 getter 메서드를 사용
- ⑤ b.jsp에서는 useBean 액션 태그를 이용해 빈즈를 참조하고 getProperty 메서드를 이용해 멤버 변수 값을 출력



# JSP 빈즈의 개요

## ■ 빈즈와 관련된 객체

- DAO ( Data Access Object )

- ▶ 데이터베이스의 데이터에 접근하기 위해 사용되는 객체
- ▶ 비즈니스 로직과 데이터베이스에 접근하는 로직을 분리시키기 위해 사용

- VO ( Value Object )

- ▶ DTO ( Data Transfer Object )
- ▶ 문서와 문서 사이에서 데이터를 전송하기 위한 용도로 사용되는 객체
- ▶ Controller, View, Business layer 사이의 데이터 전송 역할 수행
- ▶ POJO 형태로 setter와 getter만을 가지는 구조

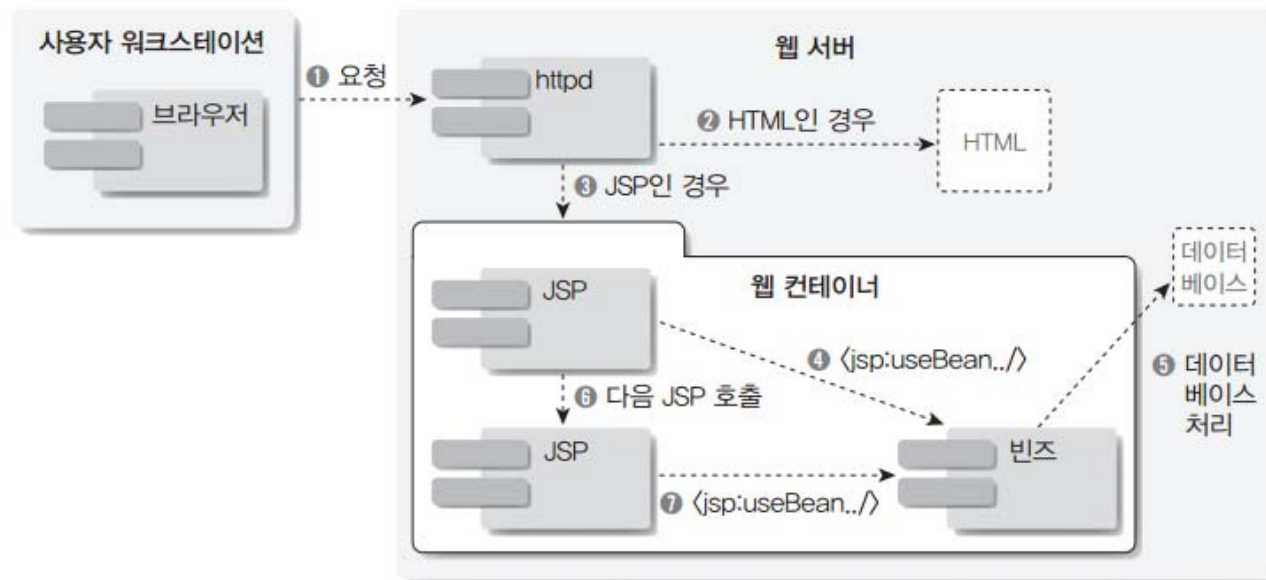
# JSP 빈즈의 활용

## ■ 일반적인 JSP 구조에서의 빈즈

- 데이터베이스 연동을 포함해서 데이터와 여러 기능을 함께 구현하는 방법

- ▶ DAO와 VO 객체의 구분 없이 함께 구현한 구조

- 빈즈 내부에 데이터베이스의 접속 부분과 데이터의 입출력 부분이 함께 존재
- 쉽고 빠르게 개발할 수 있는 장점 때문에 비교적 간단한 프로그램 구현



# JSP 빈즈의 개요

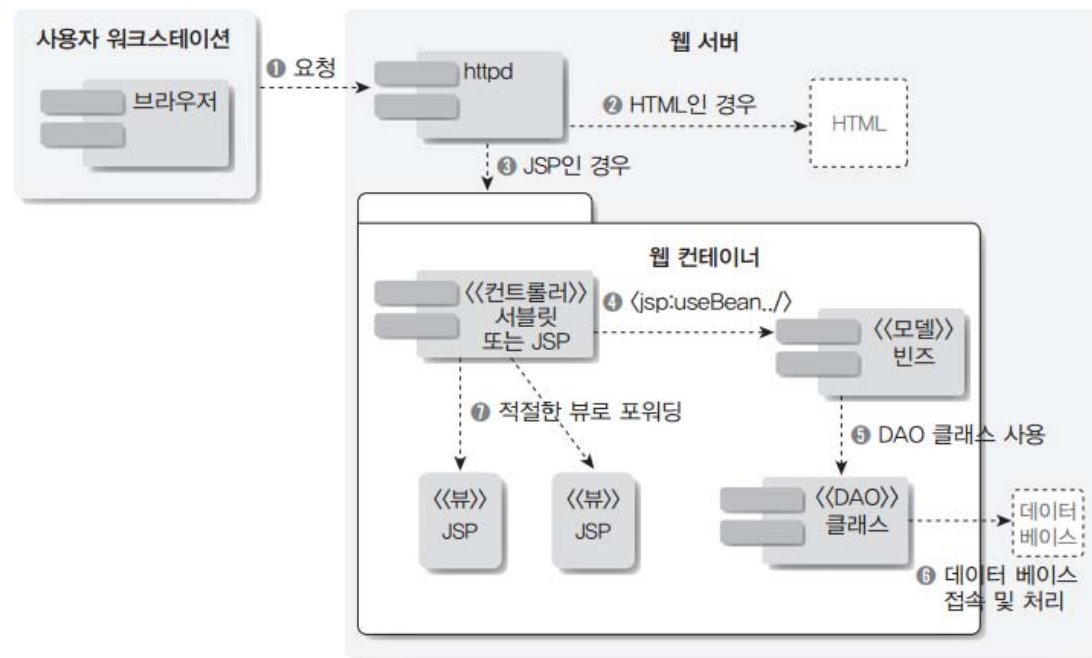
## ■ MVC 패턴을 이용한 빈즈 처리

### • 데이터 매핑과 데이터베이스 처리를 분리해서 구현하는 방법

#### ▶ VO와 DAO 객체를 분리한 형태

- VO 클래스에는 테이블과 대응할 수 있는 필드 관련 기능만 존재
- 데이터베이스 접속과 관련한 정보와 입출력과 같은 기능은 DAO라는 별도 클래스에 구현

#### ▶ 구현은 어렵지만 데이터와 비즈니스 로직을 분리하기 때문에 유지보수에 유리



# JSP 빈즈 클래스의 구조

## ■ 자바 빈즈의 구성 요소

구성 요소	설명
생성자	자바 빈즈 객체 생성시 초기화 기능을 수행
프로퍼티	자바 빈즈 객체 내에서 값을 저장하기 위한 멤버 변수 역할을 수행
setter 메서드	프로퍼티 값을 지정하기 위한 멤버 메서드 역할 수행
getter 메서드	프로퍼티 값을 추출하기 위한 멤버 메서드 역할 수행

```
public class xxxBean {  
  
    private String abc;                // 멤버 변수 ( 데이터베이스 테이블의 칼럼 이름과 매칭 )  
  
    public String getAbc() {            // getter 메서드 (멤버 변수와 매칭)  
        return abc;  
    }  
  
    public setAbc(String abc) {         // setter 메서드 (멤버 변수와 매칭)  
        this.abc = abc;  
    }  
}
```

# JSP 빈즈 클래스의 구조

- 클래스의 지정자

- ▶ 웹의 특성상 JSP 문서는 누구나 접근할 수 있어야 하기 때문에 반드시 `public`로 지정해야 함

- 생성자

- ▶ 자바 빈즈는 기본 생성자를 사용 ( 클래스 내부에서 생성자는 생략할 수 있음 )
- ▶ 매개변수가 있는 생성자를 구현하였다면 기본 생성자를 명시적으로 선언해 주어야함

- 멤버 변수

- ▶ 반드시 `private`로 선언해 외부로부터의 접근을 차단함
- ▶ 멤버 변수를 참조하기 위한 `setter/getter`메서드를 정의해 사용함
- ▶ 필요에 따라 추가로 메서드를 생성할 수 있음
- ▶ 데이터베이스와 연동해 사용하는 경우가 많으므로 해당 컬럼의 이름과 동일한 이름으로 지정

# JSP 빈즈 클래스의 구조

## ■ setter 메서드

- 멤버 변수에 값을 저장함

구분	설명
지정자	public으로 선언되어야 함
반환 타입	멤버 변수에 값을 저장하는 기능만을 수행하므로 void 반환 타입을 지정
이름	set으로 시작하는 멤버 변수의 이름으로 지정 ( 멤버 변수의 첫 글자는 대문자로 지정 )

### ▶ [ 예 ]

- 프로퍼티의 이름이 username인 경우

```
public void setUsername(String username) {  
    this.username = username;  
}
```

# JSP 빈즈 클래스의 구조

## ■ getter 메서드

- 멤버 변수에 저장된 값을 추출

구분	설명
지정자	public으로 선언되어야 함
반환 타입	멤버 변수의 데이터 타입과 동일한 반환 타입을 가져야 함
이름	get으로 시작하는 멤버 변수의 이름으로 지정 ( 변수의 첫 글자는 대문자로 지정 )

### ▶ [ 예 ]

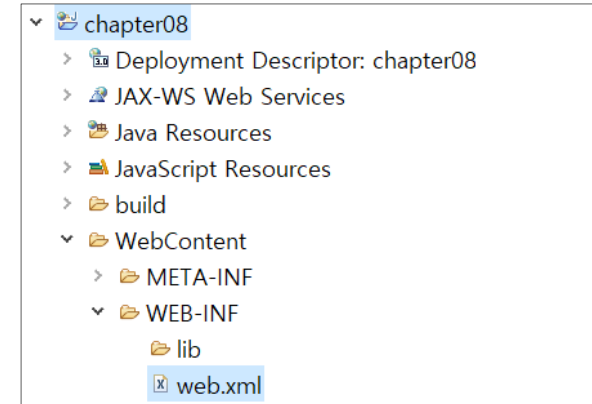
- 프로퍼티의 이름이 username인 경우

```
public String getUsername() {  
    return username;  
}
```

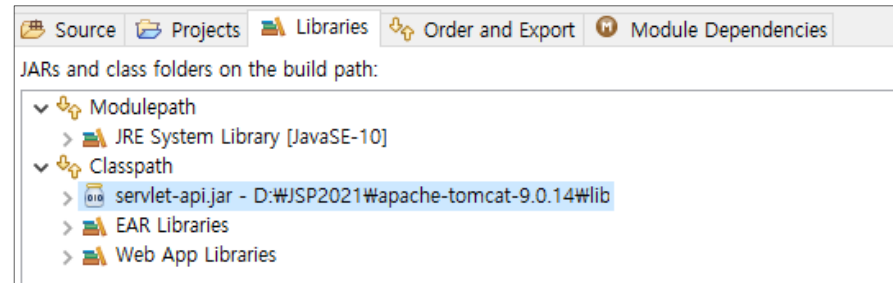
# JSP 빈즈 클래스 생성

## ■ 프로젝트 생성

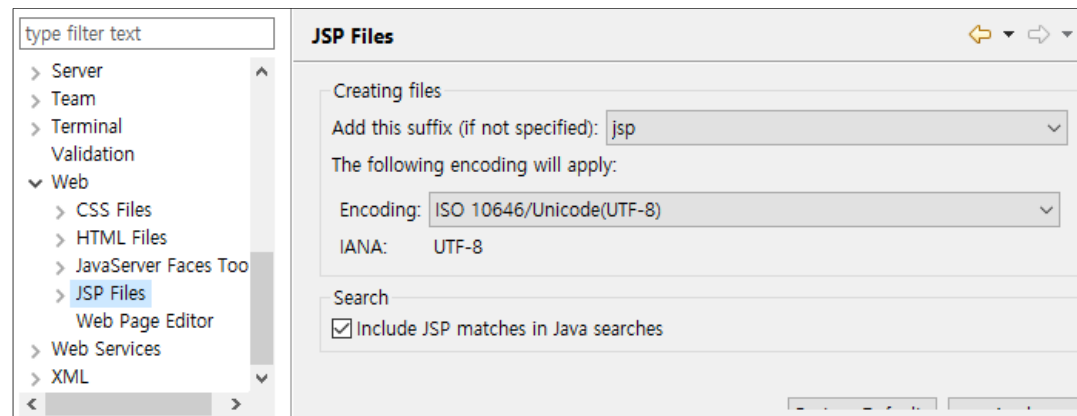
- 프로젝트 이름 : chapter08
  - ▶ web.xml 파일은 반드시 생성해야 함



- servlet-api 라이브러리 추가



- JSP 문서 문자 집합 지정
  - ▶ UTF-8로 지정





# JSP 빈즈 클래스 생성

## ■ 클래스 생성

- chapter08 패키지를 생성한 다음, 패키지 내에 Student 클래스를 생성

**New Java Class**

Create a new Java class.

Source folder: chapter08/src Browse...

Package: chapter08 Browse...

☐ Enclosing type: Browse...

Name: StudentInfo

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

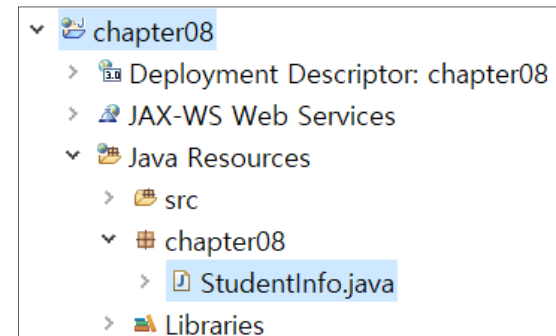
Interfaces: Add... Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)  
☐ Constructors from superclass  
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))  
☐ Generate comments

? Finish Cancel



```
1 package chapter08;  
2  
3 public class StudentInfo {  
4  
5 }  
6
```

# JSP 빈즈 클래스 생성

## ■ 멤버 변수 지정

- 멤버 변수(프로퍼티) 지정

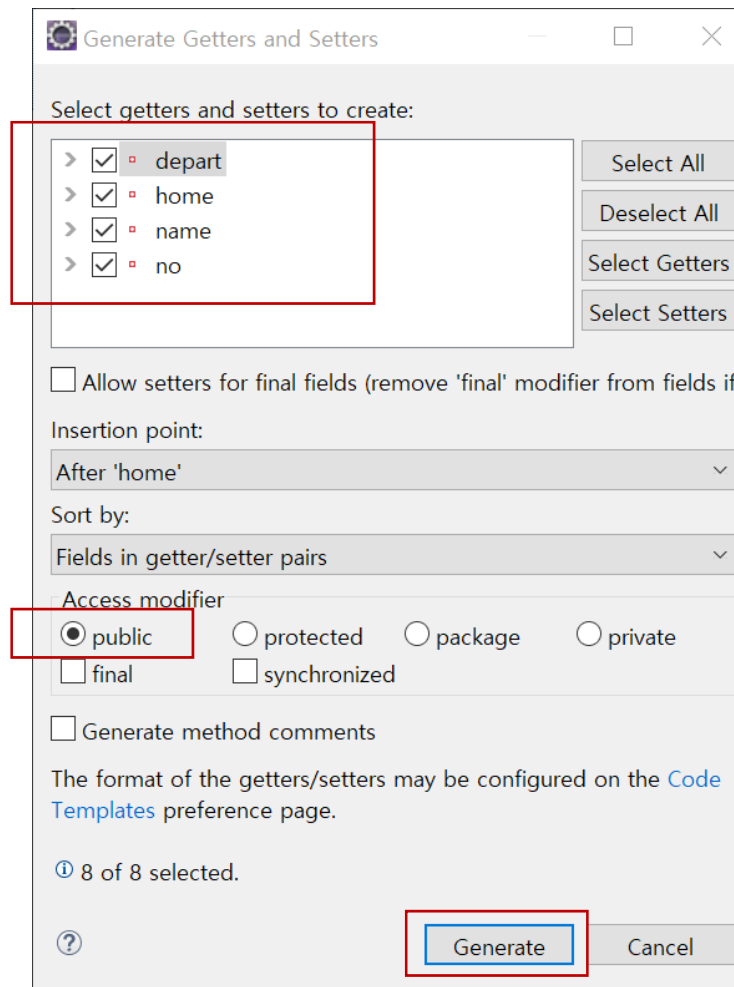
```
1 package chapter08;  
2  
3 public class StudentInfo {  
4  
5     private String name;  
6     private int no;  
7     private String depart;  
8     private String home;  
9  
10  
11 }  
12
```

# JSP 빈즈 클래스 생성

## ■ setter/ getter 메서드 생성

- 소스의 10번 라인에 커서를 위치시킨 다음 [source] / [Generate Getters and Setters]를 실행

- ▶ 두개의 멤버 변수를 모두 체크
- ▶ 지정자는 public으로 지정



```
1 package chapter08;
2
3 public class StudentInfo {
4
5     private String name;
6     private int no;
7     private String depart;
8     private String home;
9
10    public String getName() {
11        return name;
12    }
13
14    public void setName(String name) {
15        this.name = name;
16    }
17
18    public int getNo() {
19        return no;
20    }
21
22    public void setNo(int no) {
23        this.no = no;
24    }
```

```
25
26 public String getDepart() {
27     return depart;
28 }
29
30 public void setDepart(String depart) {
31     this.depart = depart;
32 }
33
34 public String getHome() {
35     return home;
36 }
37
38 public void setHome(String home) {
39     this.home = home;
40 }
41
42
43 }
```

## JSP 빈즈의 생성과 참조

# JSP 빈즈 생성

## ■ 빈즈의 생성과 참조를 위한 액션 태그

- JSP 빈즈는 액션 태그를 사용해 JSP 문서에서 생성하고 참조할 수 있음
  - JSP 빈즈는 JSP 문서 내에서 표준 액션 태그로 접근할 수 있는 자바 클래스임
- JSP 빈즈 관련 액션 태그

액션 태그	설명
<jsp:useBean>	JSP 문서에서 사용할 JSP 빈즈를 생성
<jsp:setProperty>	빈즈의 setter 메서드에 접근하여 멤버 변수(프로퍼티)에 값을 지정
<jsp:getProperty>	빈즈의 getter 메서드에 접근하여 멤버 변수(프로퍼티)에 값을 추출

# JSP 빈즈 생성

## ■ useBean 액션 태그

```
<jsp:useBean id="생성할 빈즈" scope="영역" class="빈즈 클래스" />
```

- JSP에서 사용할 빈즈 클래스의 인스턴스를 생성해주는 역할을 수행
  - ▶ 빈즈 인스턴스가 존재한다면 해당 빈즈 인스턴스를 사용
  - ▶ 빈즈 인스턴스가 존재하지 않는다면 빈즈 인스턴스를 새로 생성

속성	설명
id	생성할 빈즈의 이름
class	빈즈 클래스의 이름 ( 패키지 경로를 포함 )
scope	빈즈 인스턴스의 생존 범위 ( page, request, session application )를 지정 ( 기본 값은 page )



# JSP 빈즈 생성

- useBean 액션 태그의 예

```
<jsp:useBean id="mybean" scope="request" class="MyBean" />
```

- ▶ 위의 액션 태그는 다음과 같은 자바 코드로 변환됨

```
MyBean mybean = (MyBean)request.getAttribute("mybean");

if(mybean == null) {
    mybean = new MyBean();
    request.setAttribute("mybean", mybean);
}
```

- 빈즈가 존재하면 사용하고, 빈즈가 존재하지 않으면 생성함
- JSP 페이지의 스크립트릿에서 mybean을 참조 변수로 사용할 수 있음

# JSP 빈즈 참조

## ■ setProperty 액션 태그

```
<jsp:setProperty name="빈즈이름" property="프로퍼티" [value="지정값" | param="파라미터"] />
```

- 빈즈의 setter 메서드에 접근하여 멤버 변수(프로퍼티)에 값을 지정하기 위한 메서드
  - setProperty 액션 태그는 해당 setter 메서드를 자동으로 호출하여 값을 저장

속성 값	설명
name	- useBean 액션 태그의 id 속성 값으로 빈즈의 이름
property	- 자바 빈즈 클래스에 지정한 프로퍼티(멤버 변수) 이름을 의미 - 프로퍼티 이름 외에 * 기호를 값으로 가질 수 있음
value	- 자바 빈즈의 프로퍼티 값을 전달받은 값이 아닌 새로운 값으로 지정하거나 변경 - 변경하지 않을 경우 사용하지 않음
param	- 자바 빈즈의 프로퍼티에 값을 전달하는 <form> 요소의 매개변수 이름 - value 속성과 함께 사용할 수는 없음

# JSP 빈즈 참조

## ■ setProperty 액션 태그의 동작

```
<tr>
  <td align=center width=120>성명</td>
  <td width=180><input type=text name="name" size=15></td>
</tr>
```

```
1 package chapter08;
2
3 public class StudentInfo {
4
5     private String name;
6
7
8     public String getName() {
9         return name;
10    }
11
12    public void setName(String name) {
13        this.name = name;
14    }
15 }
```

저장

student 빈즈

name

setName()

생성

실행

```
<jsp:useBean id="student" class="chapter08.StudentInfo" scope="request" />
```

```
<jsp:setProperty property="name" name="student"/>
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <meta charset="UTF-8">
7 </head>
8 <body>
9
10   <form name="memberinfo" method=post action="StudentInfoProc.jsp">
11
12   <table align=center width=300 cellspacing=0 cellpadding=5 border>
13
14     <tr>
15       <td align=center width=120>학번</td>
16       <td width=180><input type=text name="no" size=15></td>
17     </tr>
18
19     <tr>
20       <td align=center width=120>성명</td>
21       <td width=180><input type=text name="name" size=15></td>
22     </tr>
23
24     <tr>
25       <td align=center width=120>학과</td>
26       <td width=180><input type=text name="depart" size=15></td>
27     </tr>
```



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <% request.setCharacterEncoding("utf-8"); %>
4
5 <jsp:useBean id="student" class="chapter08.StudentInfo" scope="request" />
6
7 <jsp:setProperty property="no" name="student"/>
8
9 <jsp:setProperty property="name" name="student"/>
10 <jsp:setProperty property="depart" name="student"/>
11 <jsp:setProperty property="home" name="student"/>
12
13 <jsp:forward page="StudentInfoView.jsp"/>
14
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <jsp:useBean id="student" class="chapter08.StudentInfo" scope="request" />
4
5 학번 : <jsp:getProperty name="student" property="no" /> <br>
6 이름 : <jsp:getProperty name="student" property="name" /> <br>
7 학과 : <jsp:getProperty name="student" property="depart" /> <br>
8 주소 : <jsp:getProperty name="student" property="home" />
9
```

학번	<input type="text" value="12345"/>
성명	<input type="text" value="홍길동"/>
학과	<input type="text" value="소프트웨어전공"/>
주소	<input type="text" value="서울"/>

학번 : 12345  
이름 : 홍길동  
학과 : 소프트웨어전공  
주소 : 서울

# 파라미터가 많을 경우 처리

## ■ 전달되는 파라미터가 많을 경우

- 요청한 문서에서 전달된 프로퍼티의 수만큼 setProperty 액션 태그가 존재해야 함
  - ▶ 이전 예제에서 4개의 프로퍼티를 전달하기 위해 4개의 setProperty 액션 태그를 사용함
  - ▶ 요청한 문서에서 전달하는 프로퍼티의 수가 많다면 코드가 길어질 수 있음
    - 예를 들어, 회원 가입 문서의 경우 회원 정보 저장을 위한 많은 파라미터 전달이 요구됨
- 해결 방안
  - ▶ setProperty 속성의 값으로 '\*'를 지정
  - ▶ [중요] <form> 요소의 파라미터 이름과 빈즈의 프로퍼티의 이름이 모두 동일해야 함



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <% request.setCharacterEncoding("utf-8"); %>
4
5 <jsp:useBean id="student" class="chapter08.StudentInfo" scope="request" />
6
7 <jsp:setProperty property="*" name="student"/>
8
9 <!-- 임시로 주석처리
10
11 <jsp:setProperty property="no" name="student"/>
12
13 <jsp:setProperty property="name" name="student"/>
14 <jsp:setProperty property="depart" name="student"/>
15 <jsp:setProperty property="home" name="student"/>
16 -->
17
18 <jsp:forward page="StudentInfoView.jsp"/>
```

## 파라미터가 많을 경우 처리

학번	<input type="text" value="54321"/>
성명	<input type="text" value="이순신"/>
학과	<input type="text" value="소프트웨어전공"/>
주소	<input type="text" value="광주"/>

학번 : 54321  
이름 : 이순신  
학과 : 소프트웨어전공  
주소 : 광주

# 파라미터 값의 변경

## ■ 매개변수 값의 변경

- 클라이언트에서 전달된 매개변수의 값이 아닌 다음 값을 지정할 수 있음
  - ▶ 클라이언트의 <form> 요소에 입력된 값이 아닌 다른 값으로 지정 가능
  - ▶ hidden 형식으로 전달된 파라미터에 특정한 값으로 초기화 할 수 있음
- 변경 방법
  - ▶ setProperty 액션 태그에 value 속성을 사용해 새로운 값을 지정
  - ▶ [사용 예]

```
<jsp:setProperty property="home" name="student" value="인천" />
```

- 사용자에게 의해 입력된 home 파라미터의 값은 '인천'으로 변경됨

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <% request.setCharacterEncoding("utf-8"); %>
4
5 <jsp:useBean id="student" class="chapter08.StudentInfo" scope="request" />
6
7 <jsp:setProperty property="no" name="student"/>
8
9 <jsp:setProperty property="name" name="student"/>
10 <jsp:setProperty property="depart" name="student"/>
11 <jsp:setProperty property="home" name="student" value="인천"/>
12
13 <jsp:forward page="StudentInfoView.jsp"/>
```

학번	<input type="text" value="12345"/>
성명	<input type="text" value="홍길동"/>
학과	<input type="text" value="소프트웨어전공"/>
주소	<input type="text" value="서울"/>

학번 : 12345  
이름 : 홍길동  
학과 : 소프트웨어전공  
주소 : 인천

# 파라미터 이름이 다를 경우 처리

## ■ 파라미터 이름이 다를 경우 처리

- 일반적으로 같은 데이터를 저장하고 처리할 경우 다음의 세가지 요소는 같은 이름으로 지정함
  - ▶ <form> 요소의 파라미터 이름
  - ▶ JSP 빈즈의 프로퍼티 이름
  - ▶ 데이터베이스 테이블의 필드 이름
- <form> 요소의 파라미터 이름과 JSP 빈즈의 프로퍼티 이름 서로 다를 경우 처리 방법
  - ▶ setProperty 액션 태그에 param 속성을 사용해 <form> 요소의 파라미터 이름을 지정
  - ▶ [사용 예]
    - <form> 요소의 파라미터 이름이 userNo이고, 이를 저장하기 위한 JSP 빈즈의 프로퍼티 이름이 no인 경우

```
<jsp:setProperty property= " no" name="student" param="userNo" />
```

## 파라미터 이름이 다를 경우 처리

```
<tr>
  <td align=center width=120>학번</td>
  <td width=180><input type=text name="userNo" size=15></td>
</tr>
```

```
package chapter08;

public class StudentInfo {

    private String name;
    private int no;
    private String depart;
    private String home;
```

```
<jsp:useBean id="student" class="chapter08.StudentInfo" scope="request" />

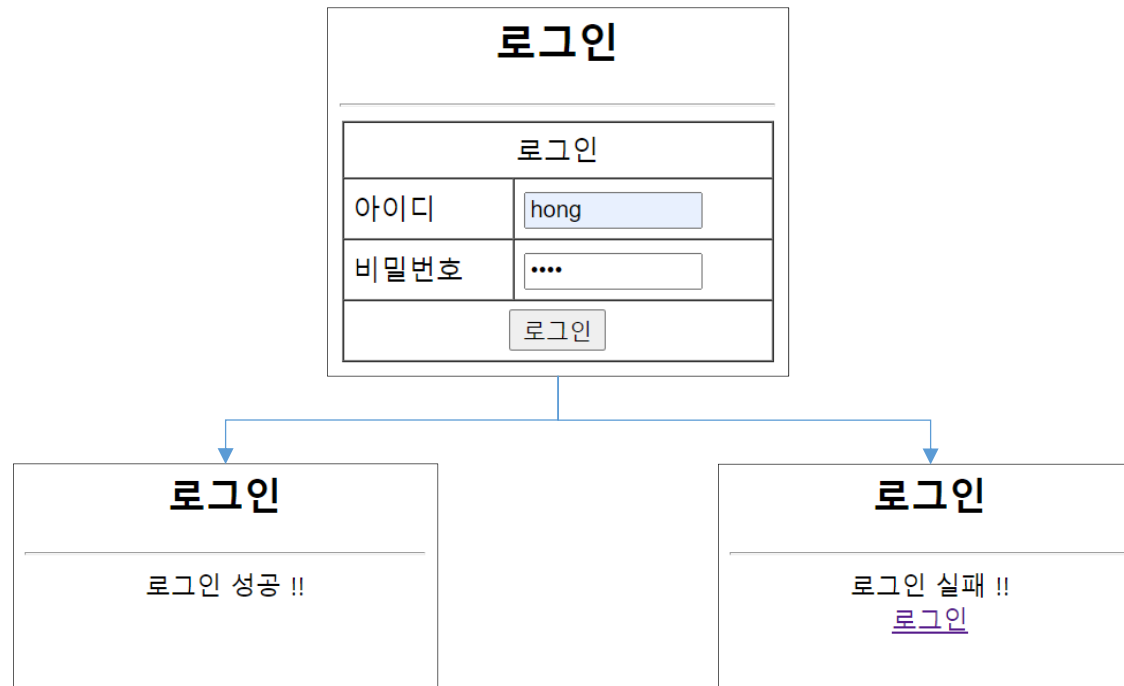
<jsp:setProperty property="no" name="student" param="userNo"/>
```

학번	<input type="text" value="11223344"/>
성명	<input type="text" value="강감찬"/>
학과	<input type="text" value="소프트웨어전공"/>
주소	<input type="text" value="부산"/>

학번 : 11223344  
이름 : 강감찬  
학과 : 소프트웨어전공  
주소 : 부산

# 빈즈 활용 예제 – 간단한 로그인 구현

## ■ 간단한 로그인 구현



파일명	설명
Loginform.jsp	사용자로부터 계정 정보를 입력 받음
LoginBean.java	사용자 입력 정보를 저장하는 JSP 빈즈 클래스 ( 계정 정보 일치 여부를 조사하는 메서드 포함 )
Login.jsp	로그인 여부 출력

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <meta charset="UTF-8">
7 </head>
8 <body>
9
10   <div align=center>
11
12     <h2>로그인</h2>
13
14     <hr width=250 align=cneter>
15
16     <form method="post" action="Login.jsp">
17
18       <table width="250" border="1" align="center" cellspacing="0" cellpadding="5">
19
20         <tr>
21           <td colspan="2" align="center">로그인</td>
22         </tr>
23
24         <tr>
25           <td>아이디</td>
26           <td><input type="text" name="userid" size=10></td>
27         </tr>
```



```
28
29     <tr>
30         <td>비밀번호</td>
31         <td><input type="password" name="passwd" size=10></td>
32     </tr>
33
34     <tr>
35         <td colspan="2" align="center">
36             <input type="submit" name="Submit" value="로그인">
37         </td>
38     </tr>
39
40 </table>
41
42 </form>
43
44 </div>
45
46
47 </body>
48 </html>
```

```
1 package chapter08;
2
3 public class LoginBean {
4
5     // 사용자로부터 입력받은 아이디와 비밀번호 저장
6     private String userid;
7     private String passwd;
8
9     // DB로부터 가져온 아이디와 비밀번호로 가정함
10    final String _userid="hong";
11    final String _passwd="1234";
12
13    public String getUserid() {
14        return userid;
15    }
16
17    public void setUserid(String userid) {
18        this.userid = userid;
19    }
20
21    public String getPasswd() {
22        return passwd;
23    }
24 }
```

```
24
25 public void setPasswd(String passwd) {
26     this.passwd = passwd;
27 }
28
29 // 계정 정보가 일치하는지 확인하는 메서드
30 public boolean checkUser() {
31     if(userid.equals(_userid)&&passwd.equals(_passwd)) {
32         return true;
33     }
34     else {
35         return false;
36     }
37 }
38
39 }
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <jsp:useBean id="login" class="chapter08.LoginBean" scope="page" />
4 <jsp:setProperty name="login" property="*" />
5
6 <!DOCTYPE html>
7 <html>
8 <head>
9   <meta charset="UTF-8">
10 </head>
11 <body>
12
13   <div align=center>
14     <h2>로그인</h2>
15     <hr width=250 align=center>
16 <%
17     if(!login.checkUser()) {
18       out.println("로그인 실패 !! <br> ");
19       out.println("<a href='LoginForm.jsp'>로그인</a>");
20     }
21     else {
22       out.println("로그인 성공 !!");
23     }
24 %>
25   </div>
26
27 </body>
28 </html>
```

## 빈즈 활용 예제 – 주소록 구현

### ■ 빈즈를 이용한 간단한 주소록 구현

- 데이터베이스를 아직 학습하지 않았으므로 주소 데이터 관리를 위해 ArrayList를 사용

파일명	설명
AddrForm.jsp	- 주소록 데이터를 받아들이기 위한 form 양식 문서
AddrAdd.jsp	- addr_form.jsp 문서로 받아들인 데이터를 AddrBean에 매핑 - addrManager를 이용해 데이터를 추가
AddrList.jsp	- application scope에 저장된 AddrManager를 통해 주소 데이터를 추출하고 화면에 출력
AddrBean.java	- 요청된 문서의 파라미터를 매핑하기 위한 빈즈 클래스
AddrManager.java	- 주소 데이터를 관리하기 위한 빈즈 클래스 - scope를 application으로 지정하여 전체 어플리케이션에서 데이터를 사용할 수 있도록 함

# 빈즈 활용 예제 – 주소록 구현

## 주소록 등록

이름	<input type="text" value="이광"/>
전화번호	<input type="text" value="010-5455-5268"/>
이메일	<input type="text" value="gwang@ut.ac.kr"/>
성별	<input type="text" value="남"/>
<input type="button" value="확인"/> <input type="button" value="취소"/>	

## 등록내용

이름	이광
전화번호	010-5455-5268
이메일	gwang@ut.ac.kr
성별	남

[목록 보기](#)

## 주소록 등록

이름	<input type="text"/>
전화번호	<input type="text"/>
이메일	<input type="text"/>
성별	<input type="text" value="남"/>
<input type="button" value="확인"/> <input type="button" value="취소"/>	

## 주소록 목록

이름	전화번호	이메일	성별
이광	010-5455-5268	gwang@ut.ac.kr	남

[주소추가](#)

```
1 package chapter08;
2
3 public class AddrBean {
4
5     private String username;
6     private String usertel;
7     private String usermail;
8     private String usergender;
9
10    public String getUsername() {
11        return username;
12    }
13
14    public void setUsername(String username) {
15        this.username = username;
16    }
17
18    public String getUsertel() {
19        return usertel;
20    }
21
22    public void setUsertel(String usertel) {
23        this.usertel = usertel;
24    }
25
```

```
26 public String getUsermail() {  
27     return usermail;  
28 }  
29  
30 public void setUsermail(String usermail) {  
31     this.usermail = usermail;  
32 }  
33  
34 public String getUsergender() {  
35     return usergender;  
36 }  
37  
38 public void setUsergender(String usergender) {  
39     this.usergender = usergender;  
40 }  
41  
42 }
```



```
1 package chapter08;
2
3 import java.util.ArrayList;
4
5
6 public class AddrManager {
7
8     // 주소록 목록을 관리하기 위한 ArrayList
9     List<AddrBean> addrlist = new ArrayList<AddrBean>();
10
11     // 주소록 추가 메서드
12     public void add(AddrBean ab) {
13         addrlist.add(ab);
14     }
15
16     // 주소록 목록 전달 메서드
17     public List<AddrBean> getAddrList() {
18         return addrlist;
19     }
20
21 }
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6     <meta charset="UTF-8">
7 </head>
8 <body>
9     <div align="center">
10
11         <h2>주소록 등록</h2>
12         <hr width=300 align=center>
13
14         <form name=form1 method="post" action=AddrAdd.jsp>
15
16             <table border=1 width=300 cellspacing="1" cellpadding="5">
17
18                 <tr>
19                     <td width=100>이 름</td>
20                     <td width=200><input type=text size=25 name=username></td>
21                 </tr>
22
23                 <tr>
24                     <td>전화번호</td>
25                     <td><input type=text size=25 name=usertel></td>
26                 </tr>
```

```
28<tr>
29    <td>이메일</td>
30    <td><input type=text size=25 name=usermail></td>
31</tr>
32
33<tr>
34    <td>성별</td>
35    <td>
36        <select name=usergender >
37            <option selected>남</option>
38            <option>여</option>
39        </select>
40    </td>
41</tr>
42
43<tr>
44    <td colspan=2 align=center>
45        <input type=submit value="확인">
46        <input type=reset value="취소">
47    </td>
48</tr>
49
50</table>
51
52</form>
53
54</div>
55</body>
56</html>
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3
4 <% request.setCharacterEncoding("UTF-8"); %>
5
6 <jsp:useBean id="addr" class="chapter08.AddrBean"/>
7 <jsp:setProperty name="addr" property="*/>
8
9 <jsp:useBean id="am" class="chapter08.AddrManager" scope="application"/>
10
11 <%
12     // JSP빈즈 am의 addr() 호출
13     // 입력된 파라미터 값을 저장한 addr 빈즈를 ArrayList에 저장
14
15     am.add(addr);
16
17 %>
18
19
20 <!DOCTYPE html>
21 <html>
22 <head>
23     <meta charset="UTF-8">
24 </head>
```

```
26 <body>
27
28     <div align="center">
29
30         <H2>등록내용</H2>
31
32         <hr width=300 align=center>
33
34         <table border=1 width=300 cellspacing="1" cellpadding="5">
35
36             <tr>
37                 <td width=100>이 름</td>
38                 <td width=200>
39                     <jsp:getProperty property="username" name="addr"/>
40                 </td>
41             </tr>
42
43             <tr>
44                 <td width=100>전화번호</td>
45                 <td width=200>
46                     <jsp:getProperty property="usertel" name="addr"/>
47                 </td>
48             </tr>
49
```

```
50         <tr>
51             <td width=100>이메일</td>
52             <td width=200>
53                 <jsp:getProperty property="usermail" name="addr"/>
54             </td>
55         </tr>
56
57         <tr>
58             <td width=100>성별</td>
59             <td width=200>
60                 <jsp:getProperty property="usergender" name="addr"/>
61             </td>
62         </tr>
63
64     </table>
65
66     <hr width=300 align=center>
67
68     <a href="AddrList.jsp">목록 보기</a>
69
70 </div>
71
72 </body>
73 </html>
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="chapter08.AddrBean"%>
4
5 <jsp:useBean id="am" class="chapter08.AddrManager" scope="application"/>
6
7 <!DOCTYPE html>
8 <html>
9 <head>
10   <meta charset="UTF-8">
11 </head>
12 <body>
13   <div align=center>
14
15     <H2>주소록 목록</H2>
16     <hr width=550 align=center>
17
18     <table border=1 width=550 cellspacing="1" cellpadding="5">
19
20       <tr align=center>
21         <th width=100>이름</th>
22         <th width=150>전화번호</th>
23         <th width=200>이메일</th>
24         <th width=100>성별</th>
25       </tr>
```

```
26
27<%
28    for(AddrBean ab : am.getAddrList()) {
29    %>
30        <tr>
31            <td align=center><%=ab.getUsername() %></td>
32            <td><%=ab.getUstertel() %></td>
33            <td><%=ab.getUsermail() %></td>
34            <td align=center><%=ab.getUsergender() %></td></tr>
35    <%
36        }
37    %>
38
39    </table>
40
41    <a href="AddrForm.jsp">주소추가</a><P>
42
43    </div>
44 </body>
45 </html>
```



# [ 과제 ]

## ■ 과제

- 위에서 실습한 주소록을 작성하고 소스코드를 제출하시오.
  - ▶ 주소록에 자신의 이름을 입력하고 출력한 결과를 캡처하여 소스코드와 함께 제출합니다.
    - 캡처 화면은 이미지 파일, 워드 파일, 한글 파일 모두 가능합니다
  - ▶ 제출 방법
    - e-campus에 제출
  - ▶ 제출 형식
    - 5개의 소스 파일과 캡처 화면을 압축하여 제출
    - 파일이름 : 학번-이름.zip
  - ▶ 제출 기한
    - 이번 주 일요일까지

**수고하셨습니다**