

2020년도 2학기

# 웹프로젝트실습 과목

- 11주차 -

장용미

H.P: 010-3309-4849

E-MAIL: changmi29@dongyang.ac.kr

eClass의 쪽지

# 11주차 오늘의 학습 내용

1. JDBC 프로그래밍 (8장)

2. 자바Bean (7장)

- MVC모델

3. 주소록 프로그램 개발 (교재 p342)

4. 팀프로젝트 과제 설명

## IT CookBook, 프로젝트로 배우는 자바 웹 프로그래밍

### [강의교안 이용 안내]

- 본 강의교안의 저작권은 한빛아카데미(주)에 있습니다.
- 이 자료를 무단으로 전제하거나 배포할 경우 저작권법 136조에 의거하여 최고 5년 이하의 징역 또는 5천만원 이하의 벌금에 처할 수 있고 이를 병과(併科)할 수도 있습니다.



# 프로젝트로 배우는 자바 웹 프로그래밍

Servlet, JSP, JDBC

황희정 지음

## Chapter 07. JSP와 자바 빈즈

# 목차

1. 빈즈 개요
2. JSP와 빈즈 연동
3. [기본실습] JSP 빈즈 프로그래밍 : 사용자 로그인 구현
4. [응용실습] JSP 빈즈 프로그래밍 : 주소록 구현

# 학습목표

- JSP 빈즈 구조를 이해하고 활용할 수 있다.
- useBean 액션 활용 방법을 익힌다.
- JSP 빈즈와 데이터베이스와의 연동 관계를 이해한다.

# 01. 빈즈 개요

## ■ 빈즈란 ?

- **빈즈(Beans)는 특정한 일을 독립적으로 수행하는 컴포넌트를 의미한다.**

- 원래 자바에서는 GUI(Graphic User Interface), 즉 창이나 버튼, 스크롤바 등 화면을 구성하는 다양한 위젯을 제작하려고 빈즈를 만들었다.

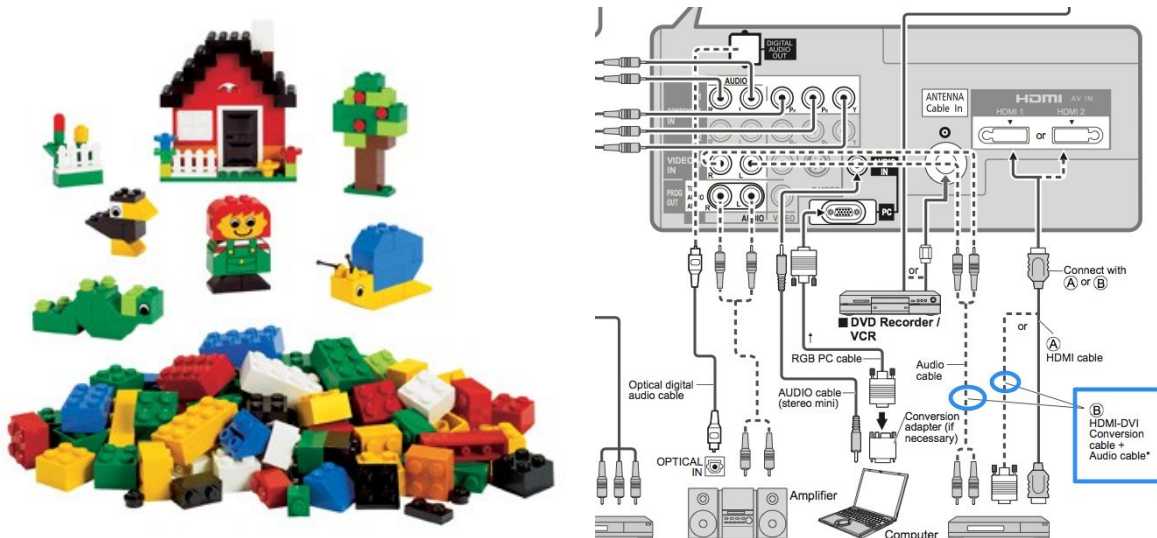
- J2EE가 발표되면서 각각 **엔터프라이즈 자바 빈즈(EJB : Enterprise Java Beans)**와 JSP에서 사용하는 **JSP 빈즈**로 나뉘어 개념이 확장되었다. 이들의 용도는 다르지만 프로그램 모듈화를 위한 컴포넌트라는 기본 전제는 같다.

- 이 절에서는 일반적인 빈즈 개념을 간단히 살펴본 후 JSP 빈즈에 대해 알아보기로 한다.

# 01. 빈즈 개요

## 1. 자바 빈즈

- 일반적으로 컴포넌트라고 하면 다른 무언가를 만들기 위한 부품을 말한다. 컴포넌트는 각각 독립적인 기능이 있으며, 컴포넌트 조합을 통해 다양한 형태의 결과물을 만들 수 있다. 예를 들어, 레고 블록이나 예전의 컴포넌트 오디오 등을 생각해볼 수 있다.
- 이때 각각의 모듈을 서로 조합하려면 규격화된 인터페이스가 있어야 한다. 레고의 경우에는 튀어나온 부분들이 다른 블록의 아래쪽에 결합이 되는 구조고, 컴포넌트 오디오는 일반 스테레오 케이블이나 HDMI, 광케이블 등으로 서로 연결된다.



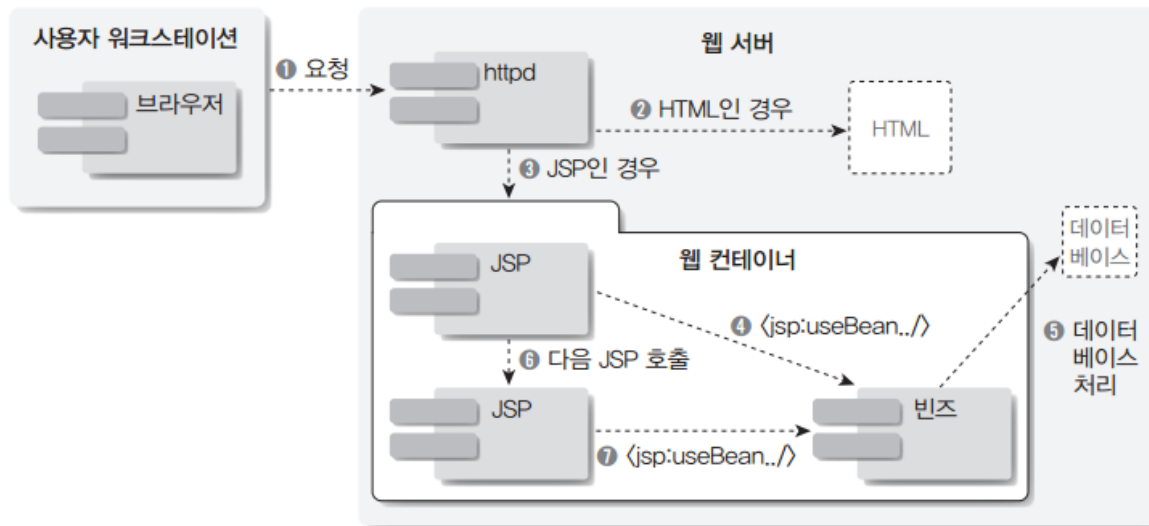
[그림] 레고와 컴포넌트 오디오 인터페이스



# 01. 빈즈 개요

## 2. JSP 빈즈

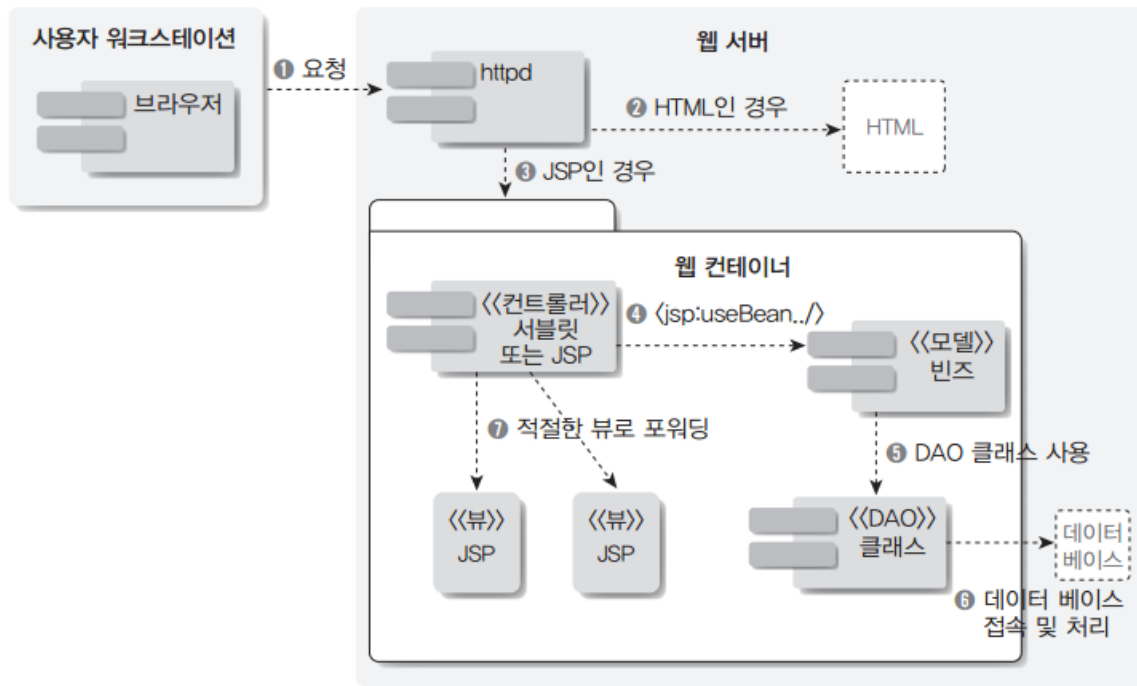
- JSP 빈즈는 JSP와 연동하기 위해 만들어진 컴포넌트 클래스이다.
- 컨테이너에 위치하며, JSP에 데이터베이스 연동 등 프로그램적 요소를 모듈화할 수 있도록 도와준다.
- 데이터 처리와 공용화된 기능을 제공하기 때문에 빈즈를 잘 활용하면 프로그램의 중복을 줄이고 더욱 원활하게 유지·보수할 수 있다.
- 따라서 가능하면 JSP 코드 내에 스크립트릿을 사용하는 것보다는 빈즈를 만들어 사용하는 것이 좋다.



[그림 7-1] 일반적인 JSP 구현에서의 빈즈

# 01. 빈즈 개요

- **MVC 패턴에 기반해** 프로그램 개발시에는 개별 JSP에서 빈즈에 접근하는 것 보다는 컨트롤러에서 빈즈와 연동하고 request, session, application 등 내장객체의 속성 관리 기능을 이용해 해당 뷰(JSP)에 빈즈 객체를 전달하는 방법이 권장된다.



[그림 7-2] MVC 패턴을 적용한 경우에서의 빈즈 사용

# MVC 패턴(Model-View-Controller Pattern)

**MVC 패턴**은 여러가지 디자인 패턴 중에 하나이다. 즉, 건축으로 비유하자면 공법이며, 소프트웨어에 빗대어 말하자면 **어떻게 소프트웨어를 개발할 것인지를 개발 방식을 공식화 한 패턴**이다.

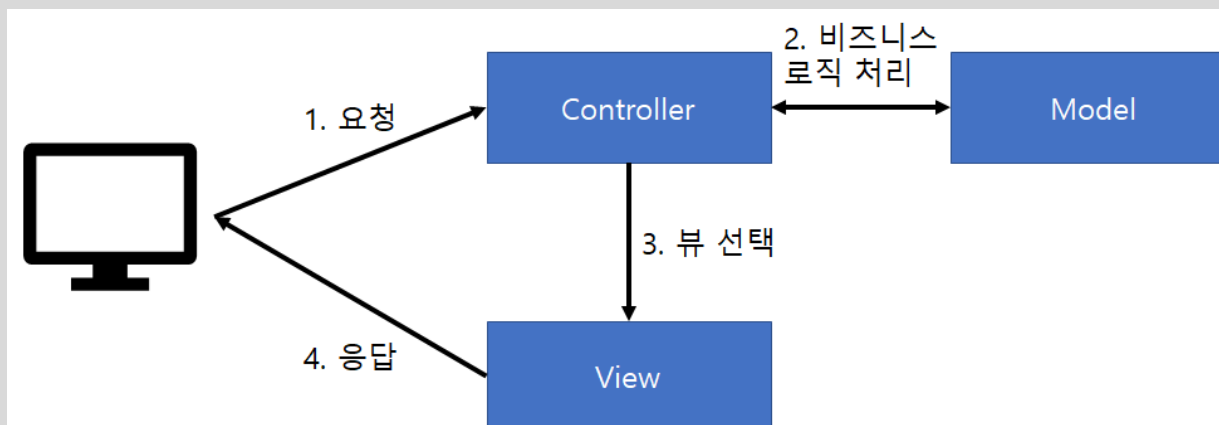
특히 MVC 패턴은 **사용자와 상호작용을 하는 어플리케이션을 개발할 때 사용되는 방식**이다. 예를 들어 웹 어플리케이션과 같이 UI(User Interface)를 통하여 사용자가 직접적으로 소통하는 부분들이 있는 어플리케이션에서 보편적으로 사용된다.

MVC(Model-View-Controller) 패턴은 이름에서도 알 수 있듯이, 어플리케이션을 기능적으로 크게 세 가지로 나눈다. 각 기능의 세부 정보는 아래와 같다.

**모델(Model)** : 비즈니스 영역의 로직을 처리한다.

**뷰(View)** : 비즈니스 영역에 대한 프레젠테이션 뷰(즉, 사용자가 보게 될 결과 화면)를 담당한다.

**컨트롤러(Controller)** : 사용자의 입력 처리와 흐름 제어를 담당한다.



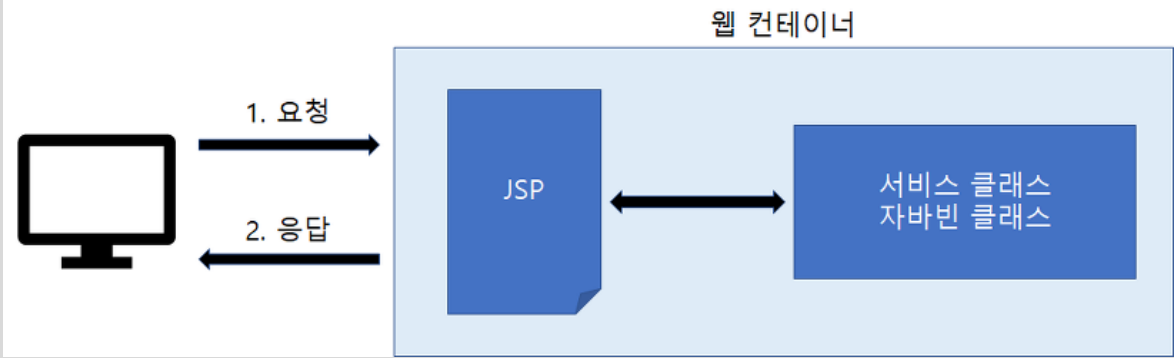
엄밀히 말하면 위의 디자인 패턴의 흐름도는 MVC2 패턴에 가깝지만, 이해를 돕기 위해서 사용하였다.

위와 같은 순서로 사용자의 요청을 처리하여 응답하기까지의 과정을 수행한다. 여기서 MVC 패턴의 핵심적인 부분을 정리하자면, **비즈니스 로직을 처리하는 모델과 결과 화면을 보여주는 뷰를 분리**한다는 것과, **어플리케이션의 흐름 제어나 사용자의 처리 요청은 컨트롤러에 집중**된다는 것이다.

\* **비즈니스 로직(Business Logic)**은 **유저의 요청에 따른 결과물을 만들어내기 위한 일련의 작업들**을 의미한다. 회원가입을 예로들면, 아이디 중복 체크를 할 때 유저에게는 단순히 중복여부를 출력만 하지만, 내부적으로는 DB에 접근하여 유저가 입력한 아이디가 존재하는지 확인하는 절차가 진행된다. 다시 정리하자면, 사용자가 어떤 요청을 했을 때, 그 **요청을 처리하기 위해 내부적으로 진행되는 절차**를 비즈니스 로직이라고 한다.

# MVC1 패턴(MVC1 Pattern)

MVC1 패턴은 JSP를 이용한 단순한 모델이다. 아래의 그림을 통해 자세히 알아보자.



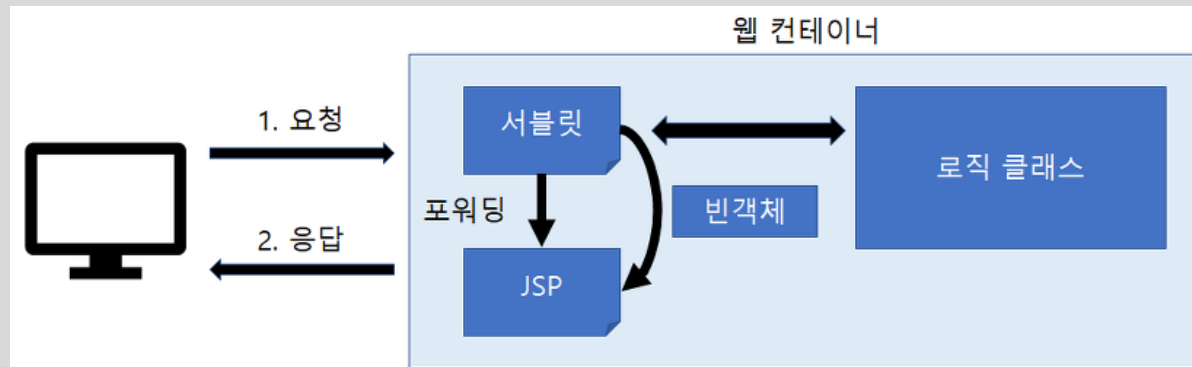
- 1. 웹 브라우저에서 요청한다.
- 2. 요청을 JSP가 받아서 자바빈이나 서비스 클래스를 사용해서 웹 브라우저가 요청한 작업을 처리한다.
- 3. 결과를 클라이언트에게 응답한다.

따라서 JSP에서는 **비즈니스 로직을 처리하기 위한 코드와 결과를 출력하는 코드가 섞이게 된다**. 따라서 클라이언트의 요청을 받은 JSP에서 요청을 처리하고 출력까지 하게 된다. 그리고 글을 게시하고 삭제하는 등의 CRUD작업들은 자바빈으로 생성하여 수행하게 된다. 그런 역할을 하는 객체를 DAO라고 한다.

\* **DAO(Data Access Object)**는 **데이터 접근 객체**로, **DB의 CRUD를 목적으로 하는 객체**이다. 하나의 커넥션을 통해서 객체를 생성하고 외부에서는 인터페이스를 통해서 접근하는 역할만하여, DB의 세부 내용을 노출시키지 않는다.

# MVC2 패턴(MVC2 Pattern)

MVC2 패턴에서는 MVC1 패턴과 달리 **뷰(View)**와 **컨트롤러(Controller)**가 나뉘어 있다.



뷰는 JSP로 작성되며, **컨트롤러는 서블릿(Servlet)으로 작성**되어 요청을 받아들이고 알맞게 처리한 후 그 결과를 보여줄 JSP 페이지로 포워딩한다.

1. 웹 브라우저에서 요청한다.
2. 요청을 서블릿이 받아서 로직 클래스를 통해 요청을 처리한다. 여기서 비즈니스 로직에 알맞는 모델을 선택하여 처리한다.
3. 포워딩을 통해 요청 흐름을 받은 JSP 페이지는 결과 화면을 생성한다.
4. 결과를 클라이언트에게 응답한다.

MVC2 패턴의 가장 큰 특징은 **웹 브라우저의 모든 요청을 단일 진입점(서블릿)에서 처리**한다는 것이다. 서블릿은 웹 브라우저의 요청을 구분하여 알맞는 모델을 찾아 요청을 처리한다.

# MVC1 패턴 vs MVC2 패턴

MVC1 패턴과 MVC2 패턴의 가장 큰 차이점은 **뷰(View)와 컨트롤러(Controller)의 분할 여부**이다. MVC1 모델은 뷰와 컨트롤러가 하나의 JSP페이지로 작성되는 형태이고, MVC2 모델은 서블릿으로 요청을 받아서 알맞게 처리한 후에 JSP 페이지로 포워딩하는 형태로 작동한다.

## < MVC1 패턴을 사용하는 경우 >

MVC1 패턴은 최근에는 거의 사용되지 않는다고 봐도 무방하다. 하지만 **소규모 프로젝트를 진행할 때**는 오히려 MVC2 패턴보다 구조가 단순하여 구축이 쉽다. 하지만 그 **규모가 커질 수록 유지보수에 약점**을 보인다.

## < MVC2 패턴을 사용하는 경우 >

MVC2 패턴은 **최근 거의 표준처럼 사용되는 패턴**이다. **규모가 큰 프로젝트를 하기에 적합**하고, 각 기능이 확실하게 분리되어 있어서 **유지보수에 상당한 이점**을 보인다. 하지만 **소규모 프로젝트를 진행함에 있어서 과도하게 구조가 복잡해질 수 있다**는 단점이 있다. 최근에는 **프레임워크**를 통해 이러한 단점은 보완되었다고 할 수 있다.

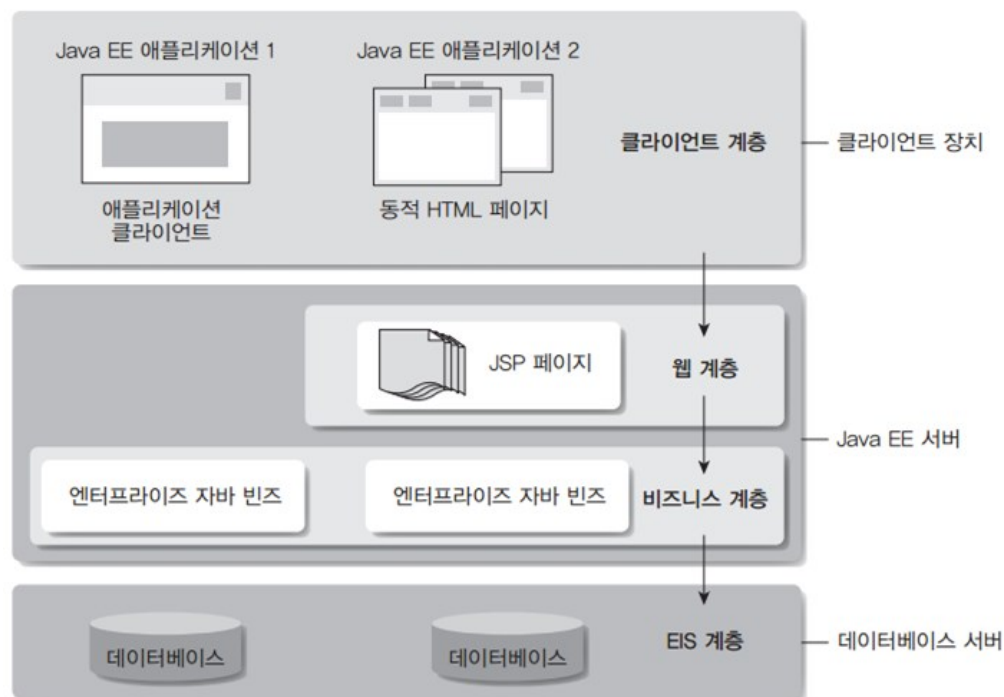
## MVC 패턴을 사용하는 이유

한 마디로 말하자면, **유지보수에서 상당히 유리해진다는 것**이다. MVC패턴으로 잘 짜여진 웹 어플리케이션은 사용자에게 보여지는 부분(View)과 흐름 제어를 담당하는 부분(Controller), 비즈니스 로직을 수행하는 부분(Model)이 독립적으로 나뉘어 있기 때문에 특정 부분을 수정하고자 하면, 그 부분만 열어서 수정하기에 상당히 용이하다.

# 01. 빈즈 개요

## 3. 엔터프라이즈 자바 빈즈

- 엔터프라이즈 자바 빈즈(Enterprise Java Beans, 이하 EJB)는 애플리케이션에서 비즈니스 로직을 구현하려는 컴포넌트 모델로, Java EE(Java Enterprise Edition)의 핵심이다.
- Java EE는 보안을 중시하고 규모가 크며 확장 및 다른 시스템과의 상호·운용을 필요로 하는 애플리케이션 개발에 필요한 핵심 기술이다. 상당수의 금융기관, 증권사, 이동통신사 등의 업무 시스템들이 Java EE를 기반으로 개발된다.
- 여러 장점에도 불구하고 개발의 복잡성, 컨테이너 간 호환성 등 여러 문제로 인해, 그 대안이 지속적으로 논의 됨.
- JPA(Java Persistence API)를 핵심으로 하는 Java EE 5가 발표되면서(최신 버전은 JavaEE 7) 기존 엔티티빈(EJB 빈 유형 중 하나)의 상당 부분이 대체됨.
- JSP도 JavaEE 의 프레젠테이션 레이어의 한 부분으로 볼 수 있음.



[그림 7-3] 엔터프라이즈 자바 빈즈 구성도

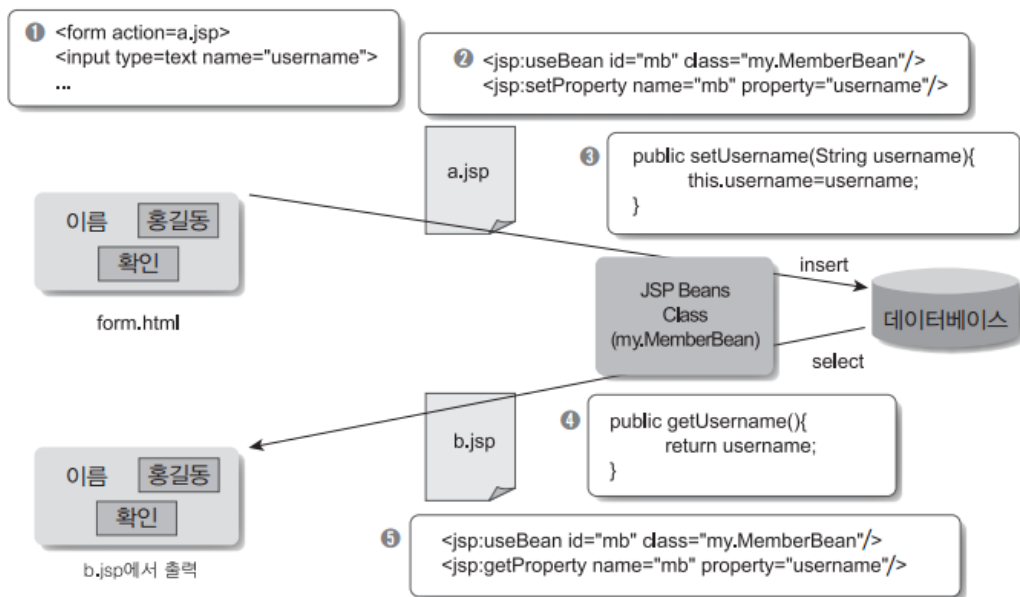
### 1. 빈즈 연동 개요

- JSP빈즈는 JSP에서 사용할 수 있는 자바 컴포넌트로 빈즈 액션과 결합해 웹 프로그램을 더욱 간편하고 단순한 구조로 개발 할 수 있게 해준다.
- 빈즈도 자바 클래스이기 때문에 JSP 파일의 스크립트릿 부분에서 인스턴스를 만들어 사용할 수 있다. 그러나 일반 클래스와 다를 바 없다면 굳이 빈즈라는 이름을 붙일 필요가 없을 것이다.
- 빈즈는 일반 클래스처럼 사용할 수도 있지만, JSP 빈즈만의 특징적인 요소를 잘 활용하면 더욱 편하게 프로그램을 작성할 수 있다.
- 일반적으로 빈즈를 사용할 경우, 사용자 입력 처리에 필요한 HTML이나 JSP와 폼의 액션에 연결된 JSP 파일이 있어야 한다. 또한 폼을 처리하는 JSP 파일에서 참조할 빈즈 클래스도 필요하다.
- 물론 경우에 따라 결과를 보여주는 HTML이나 JSP 파일이 필요할 수도 있다.



## 02. JSP와 빈즈 연동

### 1. 빈즈 연동 개요



[그림 7-4] 빈즈와 상호작용하는 구성요소

- 1** 사용자 입력을 받기 위한 HTML 폼 입력 값은 각각의 form 요소의 name 속성 값(value)을 통해 a.jsp로 전달된다.
- 2** a.jsp는 useBean 액션을 이용해 빈즈 클래스를 불러오고 setProperty를 통해 폼으로부터 전달받은 값을 빈즈 클래스의 멤버변수의 값으로 전달한다.
- 3** 이때 빈즈 클래스의 setter 메서드들이 자동으로 호출된다.
- 4** 빈즈 클래스 값을 참조하려면 getter 메서드를 사용한다.
- 5** 결과를 보여주기 위한 b.jsp에서는 a.jsp와 마찬가지로 useBean 액션을 이용해 빈즈 클래스를 참조하고 getProperty를 이용해 멤버 변수 값을 출력한다.

## 02. JSP와 빈즈 연동

- JSP 에서 빈즈 구현은 데이터베이스와의 연동이 많은 부분을 차지하며 이 경우 다음과 같이 세가지 형태의 구현을 고려할 수 있다.

### ❶ 데이터베이스 연동을 포함해서 데이터와 여러 기능을 함께 구현하는 방법(DO, DAO 클래스의 통합 형태)

빈즈 내부에 데이터베이스 접속과 관련한 정보를 포함하여 입력, 출력, 삭제와 같은 모든 데이터베이스 처리 로직을 갖는 형태. 쉽고 빠르게 개발할 수 있는 장점 때문에 비교적 간단한 프로그램 구현에 이용된다. [그림 7-1]의 일반적인 JSP에서의 빈즈 활용이 여기에 해당.

### ❷ 데이터 매핑과 데이터베이스 처리를 분리해서 구현하는 방법(DO, DAO 클래스의 분리 형태)

Data Object와 DAO(Data Access Object)를 분리한 형태로, DO 클래스에는 테이블과 대응할 수 있는 필드 관련 기능만 존재한다. 데이터베이스 접속과 관련한 정보와 입력, 출력, 삭제와 같은 기능은 DAO라는 별도 클래스에 구현한다. ❶보다 처음에는 구현이 어렵지만 데이터와 비즈니스 로직을 분리하기 때문에 유지보수에 유리하다는 장점이 있음. 이 방법은 DAO 패턴으로 불리기도 하며, 주로 MVC 패턴과 함께 사용된다. 대부분의 프로젝트에 적합한 방법이다. 따라서 처음부터 이 방법으로 프로그래밍 습관을 들이는 것이 좋다. [그림 7-2]가 여기에 해당.

### ❸ 별도의 O-R 매핑 프레임워크를 사용하는 방법

❷에서 실질적인 데이터베이스 처리를 담당하는 DAO 대신 O-R 매핑 프레임워크(O-R Mapping Framework)를 사용하는 방식이다. O-R 매핑이란 빈즈(도메인) 객체와 데이터베이스 테이블의 칼럼을 자동으로 연결하고, 이들의 값을 동기화하며 상태를 유지하는 메커니즘을 제공함으로써 개발의 생산성과 안정성을 높일 수 있는 방법이다.

### 2. 빈즈 클래스 구조

- 기본적으로 빈즈 클래스는 자바 클래스이므로, 일반적인 자바 클래스 구성을 따른다.
- JSP 액션과 연동하기 위해 필요한 몇 가지 필수 구성의 차이가 있지만 이는 문법적인 제약이 아니기 때문에 규칙을 따르지 않더라도 컴파일 오류가 발생하지는 않는다.
- 빈즈는 매개변수가 없는 기본 생성자를 요구하므로 만일 매개변수가 있는 생성자를 구현하였다면 기본 생성자를 명시적으로 선언해 주어야 문제가 발생하지 않는다.
- 빈즈 클래스의 일반적인 구조는 멤버변수, getter/setter 메서드 이다.

```
class xxxBean {  
    // 멤버변수 : 데이터베이스 테이블의 칼럼 이름과 매칭된다.  
    private String xxx;  
  
    // get, set 메서드 : 멤버변수와 매칭된다.  
    public String getXxx() {  
        return xxx;  
    }  
    public setXxx(String xxx) {  
        this.xxx = xxx;  
    }  
}
```

## 02. JSP와 빈즈 연동

### 3. JSP에서 빈즈 선언

- JSP에서 빈즈 사용을 위해 선언하는 방법은 '5장. JSP 기본문법'에서 배운 `<jsp:useBean>` 액션을 사용하는 것이다.

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

[표 7-1] useBean 액션속성 목록

액션	속성	설명
useBean	id	빈즈 클래스의 인스턴스 이름으로 사용할 변수다.
	class	빈즈 클래스의 클래스 이름으로, 패키지 경로를 포함한다.
	scope	빈즈 클래스의 범위로, page, request, session, application이 올 수 있다.

- useBean 액션은 다음과 같은 자바 코드로 변경된다. id, class, scope의 의미를 좀 더 자세히 알 수 있다.

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

- page scope인 경우

```
<%
MyBean mybean = new MyBean();
%>
```

## 02. JSP와 빈즈 연동

### 4. JSP에서 빈즈 속성 설정

- JSP 에서 빈즈를 선언한 다음, setProperty, getProperty 액션을 통해 빈즈 값을 설정하고 가져올 수 있다.

```
<jsp:setProperty name="mybean" property="userid" />
<jsp:setProperty name="mybean" property="userpasswd" />
```

[표 7-2] setProperty 속성 목록

액션	속성	설명
setProperty	name	빈즈 클래스의 인스턴스 이름으로, id 값에 설정했던 변수 이름이다.
	property	속성 값으로 빈즈 클래스의 setXxx(Xxx는 속성 이름) 메서드와 매칭될 속성 값이다. *를 지정하면 모든 setXxx에 자동으로 매칭된다.

- 입력항목이 많은 경우 모든 멤버변수에 대해 setProperty를 할 필요 없이 property="\*" 를 사용하면 됨.
- setProperty는 다음과 같은 자바 소스 코드로 변환되며 스크립트릿에서도 동일하게 사용할 수 있다.

```
<%
    mybean.setUserId(request.getParameter("username"));
    mybean.setPasswd(request.getParameter("userpasswd"));
%>
```

## 02. JSP와 빈즈 연동

### 5. 빈즈 에서 JSP로 속성 데이터 가져오기

- 빈즈에 설정되어 있는 값은 getProperty를 이용해 읽어 올 수 있다. 앞에서 setProperty로 설정된 값은 다음과 같이 JSP에서 가져올 수 있다.

```
<jsp:getProperty name="mybean" property="username" />  
<jsp:getProperty name="mybean" property="userpasswd" />
```

- getProperty 는 화면에 데이터를 출력하는 용도로 사용하기 때문에 HTML과 함께 쓰는 경우가 많다.

```
<TR><TD>이름</TD>  
<TD><jsp:getProperty name="mybean" property="username" /></TD></TR>
```

- 코드가 길어 불편한 경우 <%= %>을 사용할 수도 있다.

```
<TR><TD>ID</TD>  
<TD><%= mybean.getUsername() %></TD></TR>
```

- 추후 표현언어를 배우게 되면 \${mybean.username} 과 같이 사용하게 된다.

## 03. [기본실습] JSP 빈즈 프로그래밍 : 사용자 로그인 구현

### ■ 실습 개요

- 웹에서 사용자 로그인을 처리하는 사용자 로그인을 구현하는 간단한 예제를 통해 빈즈 구성과 주변 파일의 구성, 그리고 빈즈를 사용하기 위한 액션 태그 사용을 실습 한다.
- 실제 구현에서는 데이터베이스가 연동되어야 하지만 여기서는 데이터베이스 처리를 가정하고 생성자에서 미리 사용자 인증에 필요한 아이디와 비밀번호를 설정해 둔다.

[표 7-3] 프로그램 소스 목록

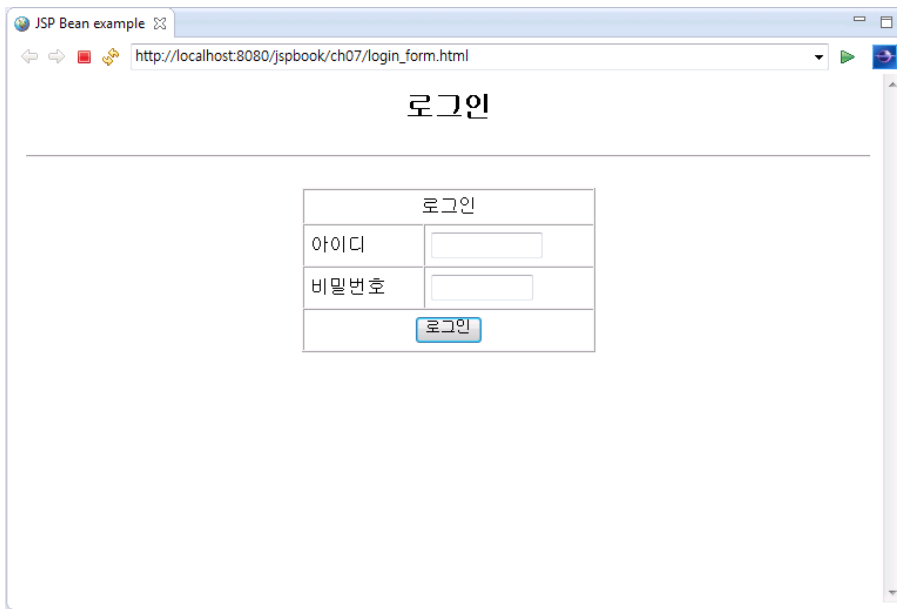
파일 이름	역할
login_form.html	사용자 로그인을 위해 아이디와 비밀번호를 입력받는 화면이다.
login.jsp	입력받은 아이디 정보를 빈즈 클래스를 이용해서 확인하고 처리하는 jsp다.
LoginBean.java	사용자가 입력한 계정 정보를 매핑하는 빈즈 클래스로써, 미리 저장된 계정 값과 비교해 로그인 성공 여부를 반환하는 <code>checkUser()</code> 메서드를 포함한다.

### 03. [기본실습] JSP 빈즈 프로그래밍 : 사용자 로그인 구현

#### 1. 사용자 인터페이스 화면 준비

##### ■ [실습] 사용자 정보 입력 화면(login\_form.html)

- 교재 p.259 ~ 260 참고



The screenshot shows a web browser window titled "JSP Bean example". The address bar displays "http://localhost:8080/jspbook/ch07/login\_form.html". The page content is a login form with the title "로그인" (Login). The form consists of a table with three rows: the first row is the title "로그인", the second row is for the "아이디" (ID) with an input field, and the third row is for the "비밀번호" (Password) with an input field. Below the password field is a "로그인" (Login) button.

로그인	
아이디	<input type="text"/>
비밀번호	<input type="password"/>
<input type="button" value="로그인"/>	

[그림 7-5] 사용자 로그인 양식 화면



# 03. [기본실습] JSP 빈즈 프로그래밍 : 사용자 로그인 구현

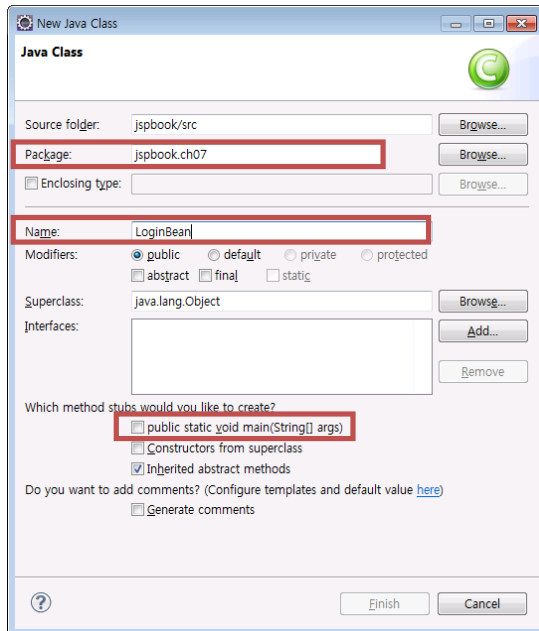
## 2. 폼 처리를 위한 JSP 파일 준비

### ■ [실습] 로그인을 처리하는 파일(login.jsp)

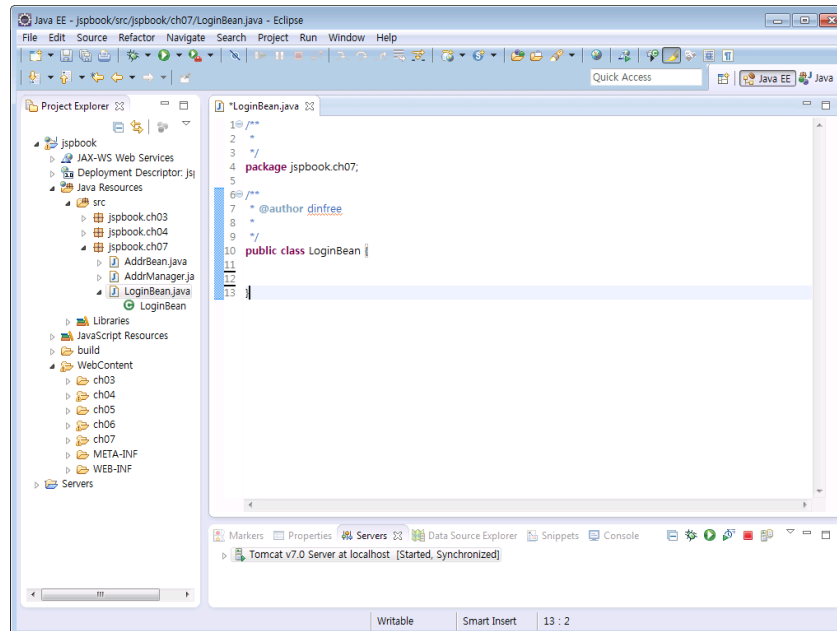
- 교재 p.260 ~ 261참고

## 3. 사용자 인증을 위한 빈즈 작성

- Package(패키지) : jspbook.ch07
- Name(이름) : LoginBean



[그림 7-6] 클래스 정보 입력



[그림 7-7] 빈즈 클래스 생성

## 03. [기본실습] JSP 빈즈 프로그래밍 : 사용자 로그인 구현

### ■ 주요 소스코드 분석

- HTML 폼(login\_form.html)

```
18 <td> 아이디</td>
19 <td><input type="text" name="userid" size=10></td>
20 </tr>
```

- 입력 항목의 name 속성값을 빈즈 클래스의 멤버변수명과 일치 시켜야 함.

- 폼 처리를 위한 JSP (login.jsp)

```
02 <jsp:useBean id="login" class="jspbook.ch07.LoginBean" scope="page" />
03 <jsp:setProperty name=:login" property="*" />
```

- useBean 액션과 setProperty를 이용해 HTML 폼 값을 빈즈 클래스 멤버변수로 매핑.

## 03. [기본실습] JSP 빈즈 프로그래밍 : 사용자 로그인 구현

### ■ 주요 소스코드 분석

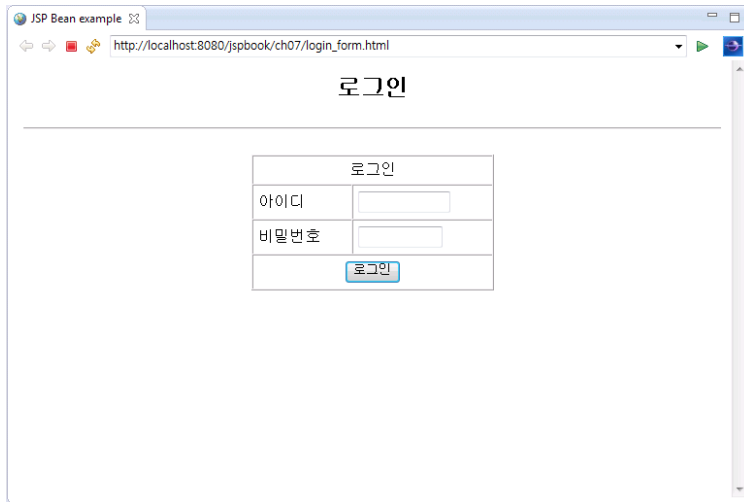
- 사용자 인증을 위한 빈즈 클래스(LoginBean.java)

```
01 // 패키지 선언
02 package jspbook.ch07;
03
04 // 클래스 선언
05 public class LoginBean{
06
07     // 멤버변수 선언
08     private String userid;
09     private String passwd;
10
11     final String _userid = "myuser";
12     final String _passwd = "1234";
13
14     public LoginBean() {}
15
16     public void setUserid(String userid) {
17         this.userid = userid;
18     }
19
20     public void setPasswd(String passwd) {
21         this.passwd = passwd;
22     }
23
24     public String getUserid() {
25         return userid;
26     }
27
28     public String getPasswd() {
29         return passwd;
30     }
31 }
```

- 패키지는 반드시 선언해 주어야 한다.
- getter, setter 메서드는 이클립스에서 자동 생성
- 멤버변수와 인증을 위한 아이디, 비밀번호 설정

### 03. [기본실습] JSP 빈즈 프로그래밍 : 사용자 로그인 구현

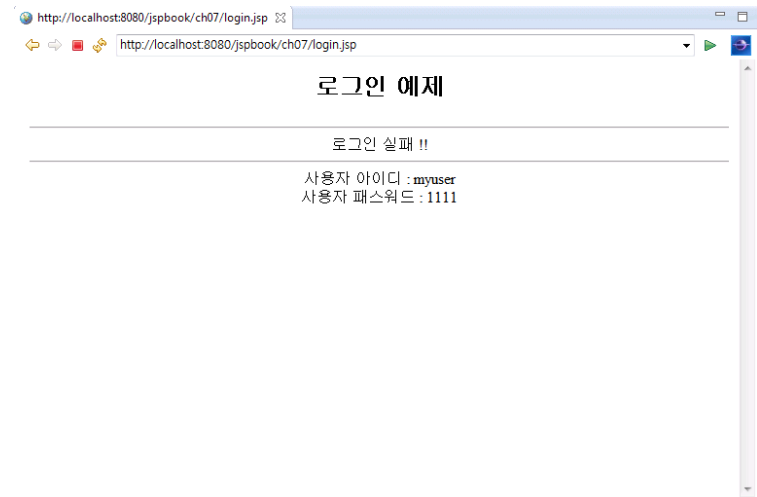
#### ■ 빈즈 컴파일과 실행



[그림 7-9] 입력 화면



[그림 7-10] 성공 화면



[그림 7-11] 실패 화면

## 04. [응용실습] JSP 빈즈 프로그래밍 : 주소록 구현

### ■ 실습 개요

- 실제 프로그램 개발에 근접한 응용예제로 주소록 프로그램을 개발한다.
- 데이터베이스는 포함되지 않으며 대신 application scope을 이용해 ArrayList로 데이터를 관리한다.
- DAO, DO 패턴이 적용되었다.

[표 7-4] 프로그램 소스 목록

파일 이름	역할
addr_list.jsp	application scope에 저장된 AddrManager 객체를 통해 저장된 데이터를 가져와서 화면에 출력한다.
addr_form.html	주소록을 등록하기 위한 양식이다.
addr_add.jsp	addr_form.html에서 입력된 내용을 AddrBean으로 매핑하고 AddrManager를 이용해 데이터 추가 기능을 수행한다.
AddrBean.java	주소록 데이터 구조 매핑을 위한 빈즈 클래스다.
AddrManager.java	ArrayList를 이용해 AddrBean 형태의 데이터를 관리하는 클래스다.

### 03. [기본실습] JSP 빈즈 프로그래밍 : 사용자 로그인 구현

## 1. 화면 구성

### ■ 주소록 목록 화면

[실습] 주소록 목록(addr\_list.jsp) - 교재 p.270 ~ 271 참고

### ■ 주소록 등록 화면과 등록 처리

[실습] 주소록 등록(addr\_form.html) - 교재 p.271 ~ 272 참고

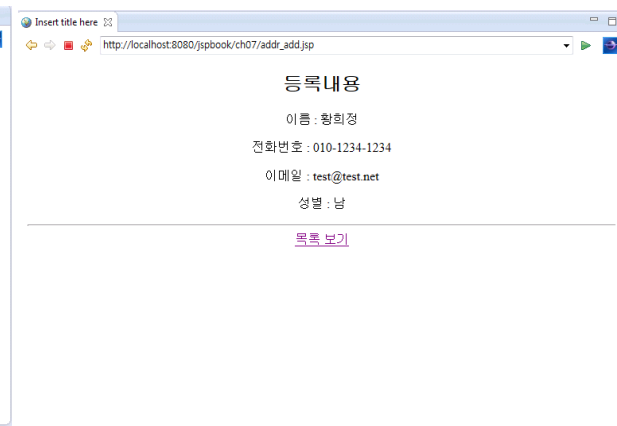
[실습] 주소록 등록 처리(addr\_add.jsp) - 교재 p.273 ~ 274 참고



[그림 7-12] 주소록 목록 화면(addr\_list.jsp)



[그림 7-13] 주소록 등록 화면(addr\_form.html)



[그림 7-14] 주소록 등록 결과 화면(addr\_add.jsp)

### 2. 빈즈 프로그램의 구현

#### ■ AddrBean 클래스 구현

[실습] 빈즈 클래스로 구현한 주소록(AddrBean.java) - [교재 p.274 ~ 275 참고](#)

#### ■ AddrManager 클래스 구현

[실습] 빈즈 클래스를 이용한 데이터 관리(AddrMamager.java) - [교재 p.276 참고](#)

## 04. [응용실습] JSP 빈즈 프로그래밍 : 주소록 구현

### ■ [실습] 예제 7-3 ~ 7-7 [교재 p. 268 ~ 276](#) 참고

### ■ 주요 소스코드 분석(addr\_list.jsp)

- 저장된 게시글을 가지고 오기 위한 빈즈 선언

```
03 <jsp:useBean id="am" class="jspbook.ch07.AddrManager" scope="application"/>
```

- application scope 으로 객체를 생성하기 때문에 생성된 객체는 톰캣 종료 전까지 유지됨.
- 처음 요청이라면 객체가 생성되고 그렇지 않으면 기존 객체를 가지고 옴.
- 가지고 온 ArrayList 를 for 문을 이용해 출력함.
- HTML과 표현식을 적절히 혼합하여 화면을 구성.

```
17 <%  
18 for(AddrBean ab : am.getAddrList()) {  
19 %>  
20 <tr>  
21 <td><%=ab.getUsername() %> </td>  
22 <td><%=ab.getTel() %> </td>  
23 <td><%=ab.getEmail() %> </td>  
24 <td><%=ab.getSex() %> </td></tr>  
25 <%  
26 }  
27 %>
```



## 04. [응용실습] JSP 빈즈 프로그래밍 : 주소록 구현

### ■ 주요 소스코드 분석(addr\_form.html)

- 특이한 부분은 없으며 HTML 파일로 form 요소의 name 값에 주의해서 작성

```
12 <form name=form1 method="post" action=addr_add.jsp>
13 <table border=1 cellspacing="1" cellpadding="5">
14 <tr>
15 <td>이 름 </td>
16 <td><input type=text size=20 name=username> </td>
17 </tr>
18 <tr>
19 <td>전화번호 </td>
20 <td><input type=text size=20 name=tel> </td>
21 </tr>
```

- form을 submit하면 action에 지정된 addr\_add.jsp로 데이터 전달. 이때 전달 방식을 POST로 해야 함.
- input 항목들의 name 값들은 빈즈 클래스 멤버변수에 일치.

## 04. [응용실습] JSP 빈즈 프로그래밍 : 주소록 구현

### ■ 주요 소스코드 분석(AddrBean.java)

- DO 클래스에 해당하며 주소록 데이터 구조를 가짐.
- 멤버변수와 getter, setter로만 구성됨.

```
03 public class AddrBean {  
04     private String username  
05     private String tel  
06     private String email  
07     private String sex  
08  
09     public String getUsername() {  
10         return username  
11     }  
12     public void setUsername(String username) {  
13         this.username = username;  
14     }
```

## 04. [응용실습] JSP 빈즈 프로그래밍 : 주소록 구현

### ■ [실습] 예제 7-3 ~ 7-7

### ■ 주요 소스코드 분석(AddrManager.java)

- DAO 클래스에 해당하며 주소록 데이터의 입력, 출력을 담당.
- 추후 데이터베이스 버전으로 업그레이드 가능하며 이 경우 AddrManager 변경만으로 데이터 베이스 연동이 가능해짐.

```
08 List<AddrBean> addrlist = new ArrayList<AddrBean>();
09
10 public void add(AddrBean ab) {
11     addrlist.add(ab);
12 }
13
14 public List<AddrBean> getAddrList() {
15     return addrlist
16 }
```

- 데이터 저장을 위한 ArrayList 를 만들고 데이터 입력과 목록을 전달하는 메서드 구현.
- ArrayList 의 타입은 AddrBean 으로 구성됨.



# 프로젝트로 배우는 자바 웹 프로그래밍

Servlet, JSP, JDBC

황희정 지음