



자바빈과 <jsp:useBean> 액션 태그

Contents

- ❖ 1. 자바빈(JavaBeans)의 개요
- ❖ 2. 자바빈 클래스 만들기
- ❖ 3. `<jsp:useBean>` 액션 태그를 이용한 자바 객체 사용

1. 자바빈(JavaBeans)의 개요

- ❖ 자바빈은 속성(데이터), 변경 이벤트, 객체 직렬화를 위한 표준
- ❖ JSP에서는 속성을 표현하기 위한 용도로 사용

- ❖ 현재까지 작성해온 JSP 페이지의 문제점
 - JSP 페이지에 화면 표출 부분과 데이터 핸들링 부분이 혼재
 - 동일한 데이터를 사용할 경우 페이지마다 코드가 반복됨
 - JSP 페이지와 데이터 로직을 분리해서 모듈화하는 작업이 필요

- ❖ 자바빈을 사용하는 목적
 - JSP 페이지의 데이터 속성 부분을 분리
 - 자바빈을 이용한 코드 재사용으로 프로그램의 효율을 높임
 - 프로그램 유지 보수 관리도 용이

1. 자바빈(JavaBeans)의 개요

- ❖ 자바빈은 속성(데이터), 변경 이벤트, 객체 직렬화를 위한 표준
- ❖ JSP에서는 속성을 표현하기 위한 용도로 사용
- ❖ 현재까지 작성해온 JSP 페이지의 문제점
 - JSP 페이지에 화면 표출 부분과 데이터 핸들링 부분이 혼재
 - 동일한 데이터를 사용할 경우 페이지마다 코드가 반복됨
 - JSP 페이지와 데이터 로직을 분리해서 모듈화하는 작업이 필요
- ❖ 자바빈을 사용하는 목적
 - JSP 페이지의 데이터 속성 부분을 분리
 - 자바빈을 이용한 코드 재사용으로 프로그램의 효율을 높임
 - 프로그램 유지 보수 관리도 용이

< Class >
속성 - 필드

동작 - 메소드

1. 자바빈(JavaBeans)의 개요

JVM 메모리의 객체 데이터를
바이트 형태로 변환하는 기술

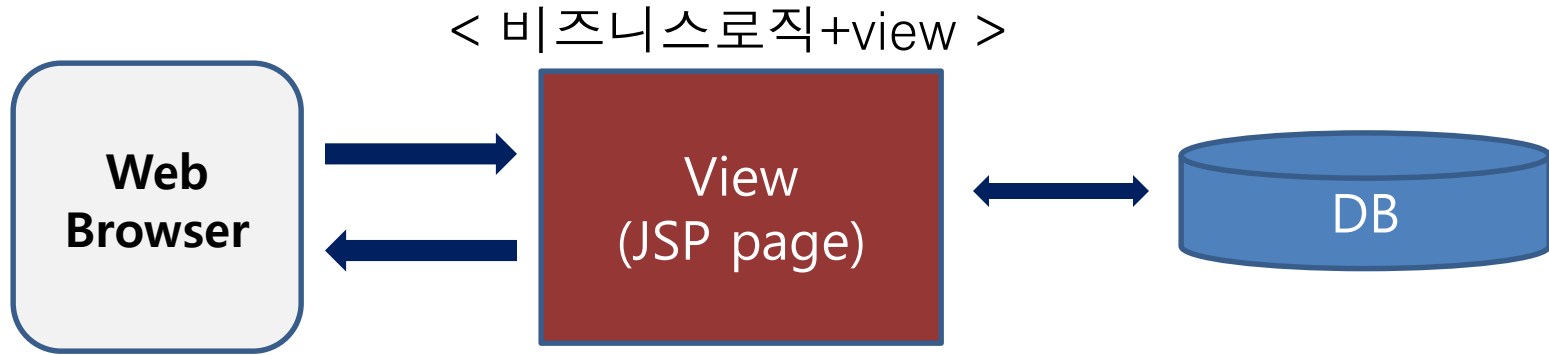
- ❖ 자바빈은 속성(데이터), 변경 이벤트, 객체 직렬화를 위한 표준
- ❖ JSP에서는 속성을 표현하기 위한 용도로 사용
- ❖ 현재까지 작성해온 JSP 페이지의 문제점
 - JSP 페이지에 화면 표출 부분과 데이터 핸들링 부분이 혼재
 - 동일한 데이터를 사용할 경우 페이지마다 코드가 반복됨
 - JSP 페이지와 데이터 로직을 분리해서 모듈화하는 작업이 필요
- ❖ 자바빈을 사용하는 목적
 - JSP 페이지의 데이터 속성 부분을 분리
 - 자바빈을 이용한 코드 재사용으로 프로그램의 효율을 높임
 - 프로그램 유지 보수 관리도 용이

< Class >
속성 - 필드

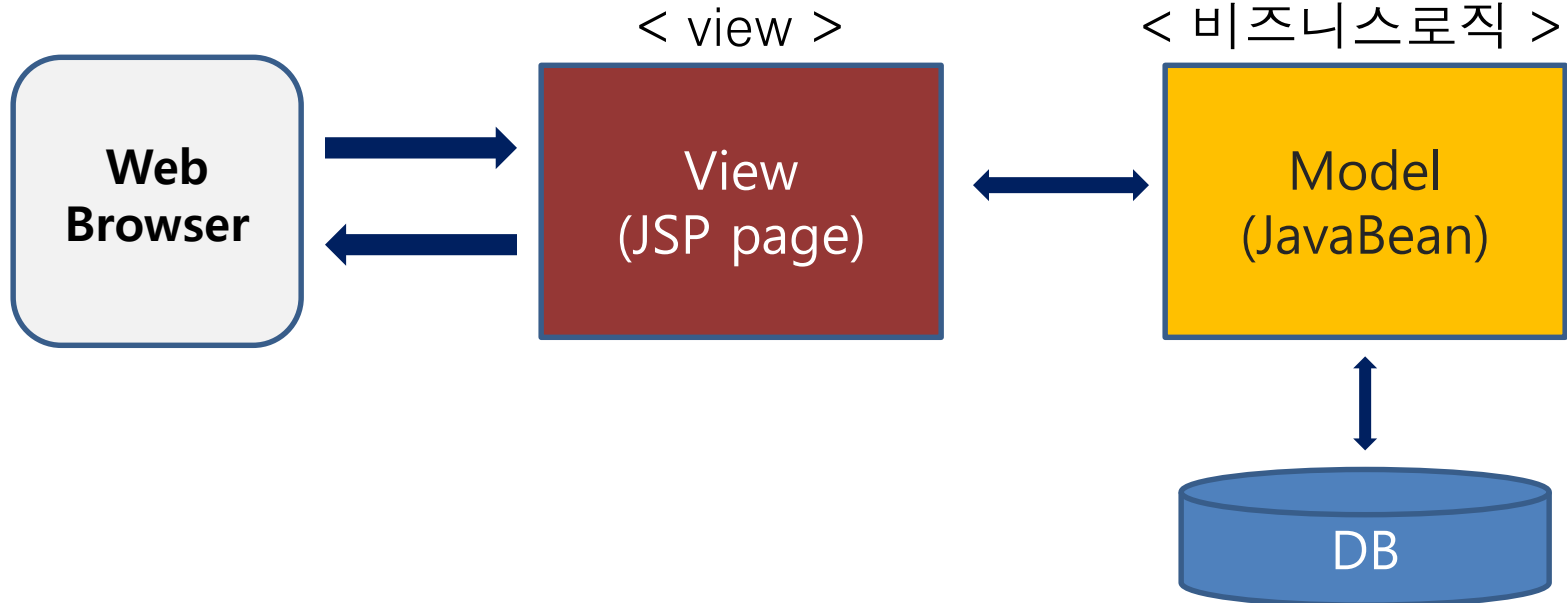
동작 - 메소드

1. 자바빈(JavaBeans)의 개요

❖ 지금까지 작성한 JSP 형태



❖ 자바빈을 사용하는 JSP 형태



1. 자바빈(JavaBeans)의 작성

❖ 자바빈은 자바의 클래스이므로 자바의 클래스를 만드는 것과 동일한 규칙을 따른다

❖ 작성 순서

- package 명 //생략 가능
 - package 패키지명;
- 해당 클래스를 생성할 때 필요한 패키지명을 포함한 클래스명 //생략 가능
 - import 패키지명;
- class 문 //반드시 필요
 - public class 클래스명;

1. 자바빈(JavaBeans)의 작성

❖ 자바빈의 클래스를 선언

- 접근제어자 `class` 클래스명 { }
- 접근제어자는 `public`, `private`, `default`(접근제어자가 없는 경우)가 올 수 있다.
- 자바빈을 작성할 때는 접근제어의 강도가 가장 약한 `public`을 주로 사용
- 웹에서는 불특정 다수의 접근을 허용해야 하기 때문에 누구나 접근할 수 있는 **public**을 사용

❖ 클래스명

- 첫 글자는 영문 대문자로 시작하고 나머지는 소문자를 사용
- 단어가 여러 개 합쳐질 때는 시작되는 단어의 시작은 대문자로 사용
 - `public class MemberInfo { }`

1. 자바빈(JavaBeans)의 작성

❖ 자바빈의 속성(property, 멤버변수) 선언

- property(프로퍼티) 는 값을 저장하기 위한 필드로 접근제어자를 **private**로 선언해서 작성

- `private String name;`
- `private String userId;`

❖ 자바빈의 메소드 선언

- property에 접근하기 위한 `getter()`, `setter()` 메소드는 접근제어자를 **public**으로 선언해서 작성

- `public void setUserId(String userId) {
 this.userId = userId; }`
- `public String getUserId() {
 return userId; }`

1. 자바빈(JavaBeans)의 작성

❖ 자바빈의 속성(property, 멤버변수) 선언

- property(프로퍼티) 는 값을 저장하기 위한 필드로 접근제어자를 **private**로 선언해서 작성

- `private String name;`
- `private String userId;`

데이터 저장소의 역할을 하는 프로퍼티에
값을 저장할 때 사용

❖ 자바빈의 메소드 선언

- property에 접근하기 위한 `getter()`, `setter()` 메소드는 접근제어자를 **public**으로 선언해서 작성

- `public void setUserId(String userId) {
 this.userId = userId; }`
- `public String getUserId() {
 return userId; }`

프로퍼티에 저장된 값을
읽어올 때 사용

2. 자바빈(JavaBeans) 클래스 예제

- ❖ 예제로 사용할 자바빈 클래스 작성
- ❖ 회원 정보를 저장할 때 사용할 클래스
- ❖ MemberInfo 자바빈의 프로퍼티 목록

프로퍼티 이름	값 타입	설 명
id	String	회원 아이디
password	String	회원 암호
name	String	회원 이름
registerDate	java.util.Date	회원 등록일자
email	String	회원 이메일

2. 자바빈(JavaBeans) 클래스 예제

- ❖ week5 > Java Resources > src > week5 > MemberInfo.java
- ❖ package > week5

```
2 public class MemberInfo {
3     private String id;
4     private String password;
5     private String name;
6     private String registerDate;
7     private String email;
8
9     public String getId() {
10         return id;
11     }
12     public void setId(String id) {
13         this.id = id;
14     }
15     public String getPassword() {
16         return password;
17     }
18     public void setPassword(String password) {
19         this.password = password;
20     }
21     public String getName() {
22         return name;
23     }
24     public void setName(String name) {
25         this.name = name;
26     }
```

3. <jsp:useBean>태그를 이용한 자바 객체 사용

- ❖ JSP 페이지의 주된 기능은 데이터를 보여주는 것이다.
 - ❖ 게시판의 글 목록보기, 글 읽기, 회원 정보 보기 등
 - ❖ 자바빈과 같은 클래스에 담아서 값을 보여주는 것이 일반적이다.
- ❖ 자바빈을 사용하기 위한 3가지 액션태그

자바빈 액션 태그	설 명
<jsp:useBean>	자바빈 객체를 생성하는 액션 태그
<jsp:setProperty>	생성된 자바빈 객체에 프로퍼티 값을 저장하는 액션 태그
<jsp:getProperty>	생성된 자바빈 객체에서 저장된 프로퍼티 값을 가져오는 액션 태그

3. <jsp:useBean>태그를 이용한 자바 객체 사용

❖ <jsp:useBean> 액션 태그를 사용하여 객체 생성하기

```
<jsp:useBean id="[빈 이름]" class="[자바빈클래스이름]" scope="[범위]" />
```

액션 태그 속성	설 명
id	JSP 페이지에서 자바빈 객체에 접근할 때 사용할 이름을 지정
class	패키지 이름을 포함한 자바빈 클래스의 완전한 이름을 입력
scope	자바빈 객체를 저장할 영역을 지정 page, request, session, application 중 하나를 값으로 가진다. 기본값은 page이다.

❖ 사용 예

```
<jsp:useBean id="member" class="week5.MemberInfo" scope="request" />
```

3. <jsp:useBean>태그를 이용한 자바 객체 사용

❖ <jsp:setProperty> & <jsp:getProperty> 액션 태그

- <jsp:setProperty> 액션 태그를 사용하면 생성한 자바빈 객체의 프로퍼티 값을 변경할 수 있다.

```
<jsp:setProperty name="[자바빈]" property="이름" value="[값]" />
```

액션 태그 속성	설 명
name	프로퍼티의 값을 변경할 자바빈 객체의 이름을 지정한다. <jsp:useBean> 액션 태그의 id 속성에서 지정한 값을 사용한다.
property	값을 지정할 프로퍼티의 이름을 지정한다.
value	프로퍼티의 값을 지정한다. 표현식(<%= 값%>)이나 EL(\${값})을 사용할 수 있다.

■ 사용 예

```
<jsp:useBean id="member" class="week5.MemberInfo" scope="request" />
```

```
<jsp:setProperty name="member" property="name" value="홍길동" />
```

3. <jsp:useBean>태그를 이용한 자바 객체 사용

❖ <jsp:setProperty> & <jsp:getProperty> 액션 태그

- <jsp:getProperty> 액션 태그를 사용하면 생성한 자바빈 객체에 저장된 프로퍼티 값을 읽어올 수 있다.

```
<jsp:getProperty name="[자바빈]" property="프로퍼티이름" />
```

액션 태그 속성	설 명
name	<jsp:useBean> 액션 태그의 id 속성에서 지정한 값을 사용한다.
property	출력할(값을 읽어올) 프로퍼티의 이름을 지정한다.

❖ 사용 예

```
<jsp:useBean id="member" class="week5.MemberInfo" scope="request" />
```

```
<jsp:getProperty name="member" property="name" />
```


3. <jsp:useBean>태그를 이용한 자바 객체 사용

❖ <jsp:useBean> 액션 태그를 사용하는 예제 작성

- MemberInfo 객체를 생성한 후 <jsp:forward>를 사용하여 다른 페이지로 이동

❖ makeObject.jsp

```
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  <title>Insert title here</title>
8  </head>
9  <body>
10 <jsp:useBean id="member" class="week6.MemberInfo" scope="request"/>
11 <jsp:setProperty property="id" name="member" value="admin"/>
12 <jsp:setProperty property="name" name="member" value="홍길동"/>
13 <jsp:setProperty property="email" name="member" value="admin@inhatec.ac.kr"/>
14
15 <jsp:forward page="useObject.jsp"></jsp:forward>
16 </body>
17 </html>
18
```

3. <jsp:useBean>태그를 이용한 자바 객체 사용

❖ <jsp:useBean> 액션 태그를 사용하는 예제 작성

- MemberInfo 객체를 생성한 후 <jsp:forward>를 사용하여 다른 페이지로 이동

❖ useObject.jsp

```
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   <title>Insert title here</title>
8 </head>
9 <body>
10  <jsp:useBean id="member" class="week6.MemberInfo" scope="request"/>
11  <jsp:getProperty property="name" name="member"/>
12  (<jsp:getProperty property="id" name="member"/>) 회원님, 안녕하세요.<br>
13  당신의 이메일은 <jsp:getProperty property="email" name="member"/>입니다.
14 </body>
15 </html>
```

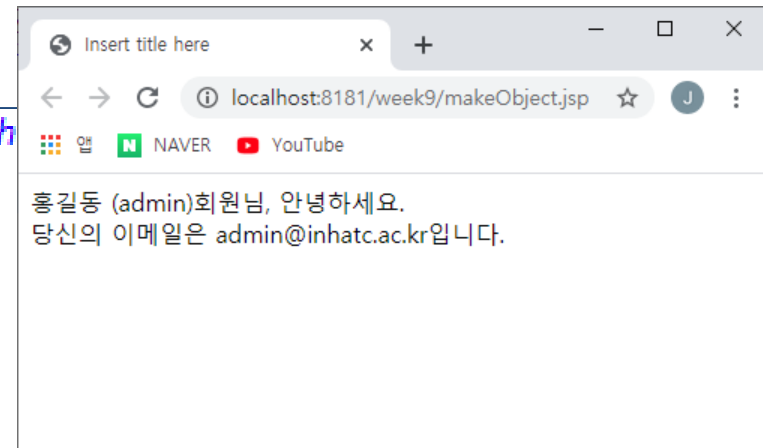
3. <jsp:useBean>태그를 이용한 자바 객체 사용

❖ <jsp:useBean> 액션 태그를 사용하는 예제 작성

- MemberInfo 객체를 생성한 후 <jsp:forward>를 사용하여 다른 페이지로 이동

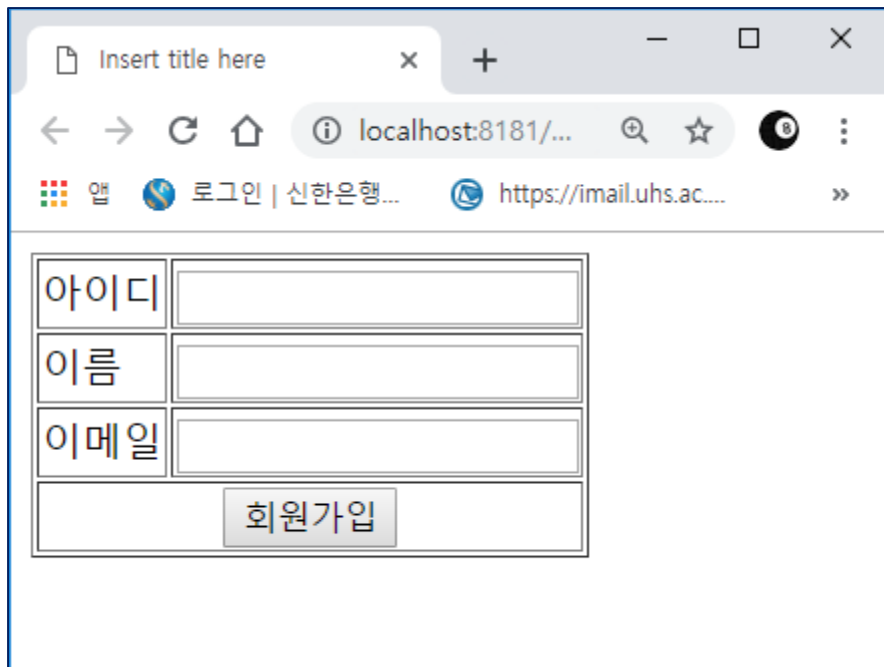
❖ useObject.jsp

```
1 <%@ page language="java" contentType="text/html" %>
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10 <jsp:useBean id="member" class="week6.MemberInfo" scope="request"/>
11 <jsp:getProperty property="name" name="member"/>
12 (<jsp:getProperty property="id" name="member"/>) 회원님, 안녕하세요.<br>
13 당신의 이메일은 <jsp:getProperty property="email" name="member"/>입니다.
14 </body>
15 </html>
```



3. <jsp:useBean>태그를 이용한 자바 객체 사용

❖ 회원 가입 입력 폼 만들기



The screenshot shows a web browser window with a single tab titled "Insert title here". The address bar displays "localhost:8181/...". Below the address bar, there are several icons and text: a grid icon, the word "앱", a globe icon, the text "로그인 | 신한은행...", a clock icon, and the URL "https://imail.uhs.ac...". The main content area of the browser contains a registration form with the following fields and a button:

아이디	<input type="text"/>
이름	<input type="text"/>
이메일	<input type="text"/>
<input type="button" value="회원가입"/>	

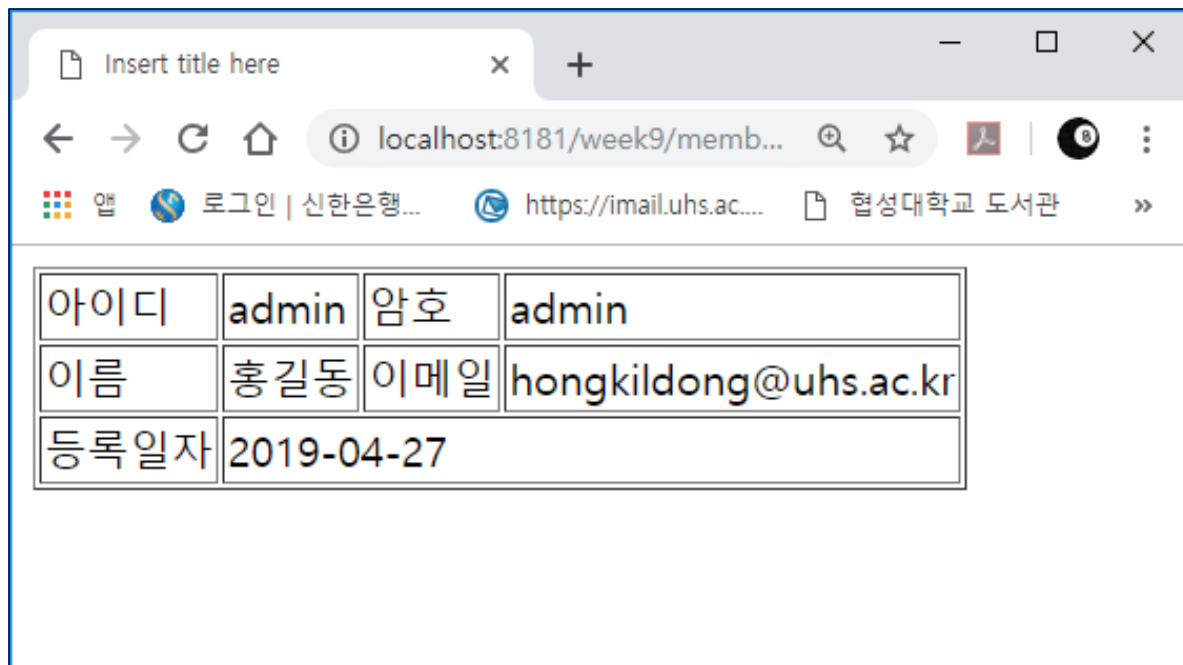
3. <jsp:useBean>태그를 이용한 자바 객체 사용

❖ 회원 가입 입력 폼 만들기

```
memberForm.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html><html><head><meta charset="UTF-8">
4 <title>Insert title here</title></head>
5 <body>
6   <form action="memberJoin.jsp" method="post">
7     <table border="1">
8       <tr><td>아이디</td>
9         <td><input type="text" name="id"></td>
10      </tr>
11     <tr><td>이름</td>
12       <td><input type="text" name="name"></td>
13     </tr>
14    <tr><td>이메일</td>
15      <td><input type="text" name="email"></td>
16    </tr>
17    <tr><td colspan="2" align="center">
18      <input type="submit" value="회원가입"></td>
19    </tr>
20  </table>
21 </form>
22 </body></html>
```

3. <jsp:useBean>태그를 이용한 자바 객체 사용

❖ 회원 가입 처리 및 결과 조회 페이지 만들기



The screenshot shows a web browser window with the address bar displaying 'localhost:8181/week9/memb...'. The browser's taskbar at the bottom shows several open applications, including a web browser, a login page, and a document viewer. The main content area of the browser displays a table with the following data:

아이디	admin	암호	admin
이름	홍길동	이메일	hongkildong@uhs.ac.kr
등록일자	2019-04-27		

3. <jsp:useBean>태그를 이용한 자바 객체 사용

❖ 회원 가입 처리 및 결과 조회 페이지 만들기

```
memberForm.jsp  memberJoin.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="java.text.SimpleDateFormat, java.util.Date" %>
4 <!DOCTYPE html><html><head><meta charset="UTF-8">
5 <title>Insert title here</title></head>
6 <body>
7 <% request.setCharacterEncoding("utf-8");
8   SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
9   Date date = new Date();
10  String strDate = sdf.format(date);
11 %>
12 <jsp:useBean id="memberInfo" class="week9.MemberInfo"/>
13 <jsp:setProperty property="*" name="memberInfo"/>
14 <jsp:setProperty property="password" name="memberInfo"
15   value="<%=memberInfo.getId() %>"/>
16 <jsp:setProperty property="registerDate" name="memberInfo"
17   value="<%=strDate %>"/>
18
19 <table border="1">
20   <tr><td>아이디</td>
21     <td><jsp:getProperty property="id" name="memberInfo"/></td>
22   <td>암호</td>
23     <td><jsp:getProperty property="password" name="memberInfo"/></td>
24   </tr>
25   <tr><td>이름</td>
26     <td><jsp:getProperty property="name" name="memberInfo"/></td>
27     <td>이메일</td>
28     <td><jsp:getProperty property="email" name="memberInfo"/></td>
29   </tr>
30   <tr><td>등록일자</td>
31     <td colspan="3"><jsp:getProperty property="registerDate" name="memberInfo"/></td>
32   </tr>
33 </table>
34 </body></html>
```

실습 문제

❖ Calculator 클래스를 이용한 계산기 프로그램

프로퍼티 이름	값 타입	설 명
num1	int	숫자1
num2	int	숫자2
op	String	연산자

계산기 - useBean

* 실행

계산기 - useBean

결과 : 408

실습 문제

❖ Calculator 클래스를 이용한 계산기 프로그램

```
2
3 public class Calculator {
4     private int num1;
5     private int num2;
6     private String op;
7
8     public int getNum1() {
9         return num1;
10    }
11
12    public void setNum1(int num1) {
13        this.num1 = num1;
14    }
15
16    public int getNum2() {
17        return num2;
18    }
19
20    public void setNum2(int num2) {
21        this.num2 = num2;
22    }
23
24    public String getOp() {
25        return op;
26    }
27
28    public void setOp(String op) {
29        this.op = op;
30    }
31
```

```
31
32    public int calc() {
33        int result = 0;
34        switch(op) {
35            case "+":
36                result = num1 + num2;
37                break;
38            case "-":
39                result = num1 - num2;
40                break;
41            case "*":
42                result = num1 * num2;
43                break;
44            case "/":
45                result = num1 / num2;
46                break;
47        }
48        return result;
49    }
50 }
51
```

실습 문제

❖ calForm.jsp

```
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  <title>Insert title here</title>
8  </head>
9  <body>
10
11  <h1>계산기 - useBean</h1>
12  <hr>
13  <form action="calcResult.jsp">
14      <input type="text" name="num1" size="10">
15      <select name="op">
16          <option>+</option>
17          <option>-</option>
18          <option>*</option>
19          <option>/</option>
20      </select>
21      <input type="text" name="num2" size="10">
22      <input type="submit" value="실행">
23  </form>
24  </body>
25  </html>
26
```

실습 문제

❖ calResult.jsp

```
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      <title>Insert title here</title>
8  </head>
9  <body>
10     <jsp:useBean id="calc" class="week6.Calculator"/>
11     <jsp:setProperty property="*" name="calc"/>
12
13     <h1>계산기 - useBean</h1>
14     <hr>
15     결과 : <%=calc.calc() %>
16
17 </body>
18 </html>
19
```

실습 문제

❖ Book 클래스를 이용한 데이터 저장 및 조회

프로퍼티 이름	값 타입	설 명
code	String	코드
title	String	제목
writer	String	저자
pubDate	String	출판일자

◆ 도서 등록 화면 ◆

도서 코드	<input type="text"/>
도서 제목	<input type="text"/>
도서 저자	<input type="text"/>
<input type="button" value="등록"/>	

◆ 도서 등록 결과 ◆

코드	제목	저자	출판일자
A0001	자바빈 이용 방법	작가 J.	2019-10-13

실습 문제

❖ Book 클래스를 이용한 데이터 저장 및 조회

```
3 public class Book {  
4     private String code;  
5     private String title;  
6     private String writer;  
7     private String pubDate;  
8     public String getCode() {  
9         return code;  
10    }  
11    public void setCode(String code) {  
12        this.code = code;  
13    }  
14    public String getTitle() {  
15        return title;  
16    }  
17    public void setTitle(String title) {  
18        this.title = title;  
19    }  
20    public String getWriter() {  
21        return writer;  
22    }  
23    public void setWriter(String writer) {  
24        this.writer = writer;  
25    }  
26    public String getPubDate() {  
27        return pubDate;  
28    }  
29    public void setPubDate(String pubDate) {  
30        this.pubDate = pubDate;  
31    }  
32 }
```

실습 문제

❖ bookForm.jsp

```
<!-- bookForm.jsp -->
<h1> ♣ 도서 등록 ♣ </h1>
<form action="bookJoin.jsp" method="post">
  <table border="1" cellpadding="3">
    <tr>
      <td bgcolor="orange">도서코드 </td>
      <td>
        <input type="text" name="code">
      </td>
    </tr>
    <tr>
      <td bgcolor="orange">도서제목 </td>
      <td>
        <input type="text" name="title">
      </td>
    </tr>
    <tr>
      <td bgcolor="orange">도서저자 </td>
      <td>
        <input type="text" name="writer">
      </td>
    </tr>
    <tr>
      <td colspan="2" align="center">
        <input type="submit" value="등록">
      </td>
    </tr>
  </table>
</form>
```

❖ bookJoin.jsp

```
<body>
<!-- bookJoin.jsp -->
<%
    request.setCharacterEncoding("utf-8");
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    Date date = new Date();
    String strDate = sdf.format(date);
%>

<jsp:useBean id="book" class="week5.Book" />
<jsp:setProperty property="*" name="book"/>
<jsp:setProperty property="pubDate" name="book"
    value="<%=strDate %>"/>

<h1> ♣ 도서 등록 결과 ♣ </h1>

<table border="1" cellpadding="3">
<tr bgcolor="pink">
    <td>도서코드 </td>
    <td>도서제목 </td>
    <td>도서저자 </td>
    <td>출판일자 </td>
</tr>
<tr>
    <td><%=book.getCode() %> </td>
    <td><%=book.getTitle() %> </td>
    <td><%=book.getWriter() %> </td>
    <td><%=book.getPubDate() %> </td>
</tr>
</table>
```