

자바 빈

반복적인 작업을 효율적으로 하기 위해 빈을 사용

데이터(속성) + 기능(메소드)로 이루어진 클래스

빈을 만든다 = 데이터 객체를 만들기 위한 클래스를 만든다

따라서 request.getParameter()로 일일이 파라미터 값을 받아올 수고를 덜 수 있다!

<빈을 만드는 규칙>

1. 멤버변수마다 별도의 getter & setter 필요
2. 매개변수 없는 생성자 꼭 필요
3. 멤버변수의 접근제어자는 private / get & set 메소드와 클래스 접근제어자는 public

(예시)

```
public class Person {
    private String name;
    private int age;
    private char gender;
    private String address;
    public Person () {}
    public Person (String name, int age, char gender, String address) {
        this.name = name;
        this.age = age;
        this.gender = gender;
        this.address = address;
    }
    public String getName() {return name;}
    public void setName(String name) {this.name = name;}
    public int getAge() {return age;}
    public void setAge(int age) {this.age = age;}
    public char getGender() {return gender;}
    public void setGender(char gender) {this.gender = gender;}
    public String getAddress() {return address;}
    public void setAddress(String address) {this.address = address;}
}
```

<자바 빈 사용 순서>

1. jsp 페이지 만들기
2. 액션태그 이용하여 빈 사용

(예시) : `<jsp:useBean id="pi" class="com.tj.ex.Person" scope="page"/>`

3. 빈의 내부 데이터 처리

➔ 6단계 : html form 안의 이름들과 자바빈의 속성이름이 같으면 간략화 가능

(예시) : `jsp:setProperty name="pi" property="*/>`

<자바빈 관련 액션태그>

1. <jsp:useBean> : 자바빈 객체를 생성하기 위한 태그

(예시) <jsp:useBean id="빈 이름" class="자바빈 클래스 명" scope="사용범위"/>

- id : 자바빈 객체의 변수 이름
- class : 자바빈에 사용할 클래스명(클래스가 패키지에 있다면 패키지 경로까지 작성)

(예시) com.lec.ex.Person

- scope : 자바빈의 사용범위 (page 디폴트값, request, session, application)

2. <jsp:setProperty> : 자바빈 클래스의 속성 값을 설정하기 위한 태그

➔ <jsp:setProperty>태그로 set 메소드를 호출하여 접근 가능

(예시) <jsp:setProperty name="빈 이름" property="속성명"/>

- name : 생성된 자바빈 객체의 id
- property : 자바빈 객체 안의 속성명

➔ 속성명에 "*"하면 클라이언트로부터 받는 모든 파라미터 값을 속성값으로 할당!

3. <jsp:getProperty> : 자바빈 클래스의 속성 값을 가져오기 위한 태그

➔ <jsp:getProperty>태그로 get메소드를 호출하여 속성들을 가져옴

(예시) <jsp:getProperty name="빈 이름" property="속성명"/>

- name : 값을 가져올 자바빈 객체의 id
- property : 가져올 자바빈 객체의 속성 이름

➔ 생성한 클래스의 매개변수명과 같아야 함

Cf) 수업에서는 step6.jsp파일(프로세스)에서 <jsp:getProperty>태그를 바로 쓰지 않고
pView.jsp파일(뷰)에 빼놓았다.

따라서 step6.jsp(프로세스)를 간략하게 하기 위해서 해당 JSP 페이지에서

<jsp:forward> 액션태그를 이용하여 pView.jsp 페이지로 요청 처리를 전달했다

Cf) pView.jsp에서 <jsp:getProperty>태그를 사용하기 위해서는

<jsp:useBean> 태그가 선행되어야 한다