CS320 Web and Internet Programming
Java Beans

Chengyu Sun
California State University, Los Angeles

Problems with Scripting Elements

- Mixing presentation and processing
  - hard to read, debug, or maintain
  - No clean and easy way to reuse code

Solution – separate out Java code
- Servlets
- Java Beans
- Custom tag libraries

Java Beans

- A regular Java object, typically used for modeling data, e.g. GuestBookEntry
- A.K.A. POJO (Plain Old Java Object)

Bean Properties

- The properties of a bean are defined by getters and setters
- Properties != Class variables

```java
public class User {
    String firstname;
    String lastname;

    public getName() { return firstname + " " + lastname; }
    public getFirstname() { return firstname; }
    public getLastName() { return lastname; }
    public getUsername() { return firstname + " " + lastname; }
}
```

About Bean Properties

- Property naming conventions
  - 1st letter is always in lower case
  - 1st letter must be capitalized in getter (accessor) and/or setter (mutator)
- Property types
  - read-only property: only getter
  - write-only property: only setter
  - read/write property: both

Bean Property Example

- What properties does FooBar have?
  - Read-only: ??
  - Write-only: ??
  - Read/write: ??

```java
public class FooBar {
    private int a, b, c, d;
    private boolean e;

    public int getA() { return a; }
    public void setA(int a) { this.a = a; }
    public int getB() { return b; }
    public void setB(int b) { this.b = b; }
    public boolean getE() { return e; }
    public void setE(boolean e) { this.e = e; }
}
```
**Common Problems with Bean Property**

```java
public class FooBar {
    private int a, b, c, d;
    public FooBar() { a = b = c = d = 0; }
    public int getA() { return a; }
    public void setA(String s) { this.a = Integer.parseInt(s); }
    public void setB(String s) { this.b = Integer.parseInt(s); }
    public void setC(String s) { this.c = Integer.parseInt(s); }
    public void setD(String s) { this.d = Integer.parseInt(s); }
}
```

How many properties does FooBar have??

**Bean Tags and Attributes**

- `<jsp:useBean`<br>  - class<br>  - id<br>  - scope<br>    - page (default)<br>    - request<br>    - session<br>    - application
- `<jsp:getProperty`<br>  - name<br>  - property
- `<jsp:setProperty`<br>  - name<br>  - property<br>  - value<br>  - param

**Example: BGColor.jsp**

- Use a bean `BGBean` to control the background color of a JSP page

```html
<jsp:useBean id="bg" class="cs320.bean.BGBean" />
<jsp:getProperty name="bg" property="r" />
```

Understand `<jsp:useBean` and `<jsp:getProperty`
**Set Bean Properties**

```jsp
<jsp:setProperty name="bg" property="r" value="255" />
<jsp:setProperty name="bg" property="r" param="r" />
<jsp:setProperty name="bg" property="r" />
<jsp:setProperty name="bg" property="*" />
```

**Understand Scopes**

- **Application scope** – data is valid throughout the life cycle of the web application
- **Session scope** – data is valid throughout the session
  - redirect, multiple separate requests
- **Request scope** – data is valid throughout the processing of the request
  - forward
- **Page scope** – data is valid within current page

**Scopes in Servlet**

- **Application scope**
  - ServletContext
- **Session scope**
  - HttpSession
- **Request scope**
  - HttpServletRequest
- **Page scope (in JSP scriptlet)**
  - pageContext