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 String getFirstname() { return firstname; }
    public String getLastname() { return lastname; }
    public String getName() { 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

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

    public Foobar() {
        a = b = c = d = 0;
    }

    public int getA() { return a; }
    public void setA(int a) { this.a = a; }

    public boolean isEqual() {
        return a == b;
    }

    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 getB( int x ) { return b+x; }
    public void setA( String s ) { this.a = Integer.parseInt(s); }
    public int getA() { return a; }
    public void setX( String s ) { this.x = Integer.parseInt(d); }
}

How many properties does Foobar have??
```

... Common Problems with Bean Property

- A getter must have no argument
- A setter must have exactly one argument
- The type of a property must be consistent in both the getter and the setter

Bean and JSP

Bean Tags and Attributes

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

Example: BGColor.jsp

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

```jsp
<jsp:useBean id="bg" class="cs320.bean.BGBean" />
<jsp:getProperty name="bg" property="r" />
<jsp:setProperty name="bg" property="r" value="255" />
<jsp:setProperty name="bg" property="r" param="r" />
<jsp:setProperty name="bg" property="r" value="3" />
<jsp:setProperty name="bg" property="r" param="r" />
```

Example: Counters.jsp

- Use Counter bean
- How do we increment the counter??
- Understand scopes
  - Application
  - Session
  - Request
  - Page
Scopes and Data Sharing

- 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

Access Scoped Variables in Servlet

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

Need for EL

- Using `<jsp:getProperty>` to access bean properties is tedious

EL

What is EL?

- Expression Language (EL)
  - Part of the JSP 2.0 Specification
  - A more concise way to access bean properties and write JSP expressions
    - vs. `<jsp:getProperty>`
    - vs. `<%= expression %>`
  - Java's answer to scripting languages
- Syntax: `${ expression }

Example: BGColor.jsp

Revisited

- Use EL to access the bean properties

  `${ bean_name.property_name }

Example: ShowCookie.jsp

- Display the value of the cookie JSESSIONID
  - PHP: `echo $HTTP_COOKIE_VARS[JSESSIONID]`
  - JSP Scriptlet?
  - EL
Expression
- Literals
- Operators
- Variables

EL Literals
- true, false
- 23, 0x10, ...
- 7.5, 1.1e13, ...
- "double-quoted", 'single-quoted'
- null
- No char type

EL Operators
- Arithmetic
  - +, -, *, /, %
  - div, mod
- Logical
  - &&, ||, !
  - and, or, not
- Relational
  - ==, !=, <, >, <=, >=
  - eq, ne, lt, gt, le, ge
- Conditional
  - ?:
- empty
  - check whether a value is null or empty
- Other
  - [ ] 

EL Evaluation and Auto Type Conversion

<table>
<thead>
<tr>
<th>Expression</th>
<th>Auto Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(2+4/2)$</td>
<td>$\text{empty} &quot;2&quot;$</td>
</tr>
<tr>
<td>$(2+3/2)$</td>
<td>$\text{empty} \text{param.a}$</td>
</tr>
<tr>
<td>$(2^2+3/2)$</td>
<td>$\text{empty} \text{null}$</td>
</tr>
<tr>
<td>$(2^2+3 \text{ div } 2)$</td>
<td>$\text{empty} \text{&quot;null&quot;}$</td>
</tr>
<tr>
<td>$(\text{&quot;a&quot; + 3 \text{ div } 2})$</td>
<td>$\text{&quot;abc&quot; + \text{&quot;b&quot;}}$</td>
</tr>
<tr>
<td>$(\text{null == &quot;test&quot;})$</td>
<td>$\text{null eq &quot;null&quot;}$</td>
</tr>
<tr>
<td>$(\text{null eq &quot;null&quot;})$</td>
<td>$\text{null eq &quot;null&quot;}$</td>
</tr>
</tbody>
</table>

EL Variables
- You cannot declare new variables using EL (after all, it’s called “expression” language).
- However, you can access beans, implicit objects, and previously defined scoped variables.

Implicit Objects in EL
- pageContext
  - servletContext
  - session
  - request
  - response
- param, paramValues
- header,headerValues
- cookie
- initParam
- pageScope
- requestScope
- sessionScope
- applicationScope
Example: RequestInfo.jsp

- Display some information about the request
  - Client address ...
  - Cookies and parameters
- Use of implicit objects
  - Find the Java class type for the object
  - Look for getters in the API
    - E.g. ${request.remoteAddr}
  - Access elements in a collection
    - cookie and param

Limitation of EL

- Only expressions, no statements, especially *no control-flow statements*

  JSTL