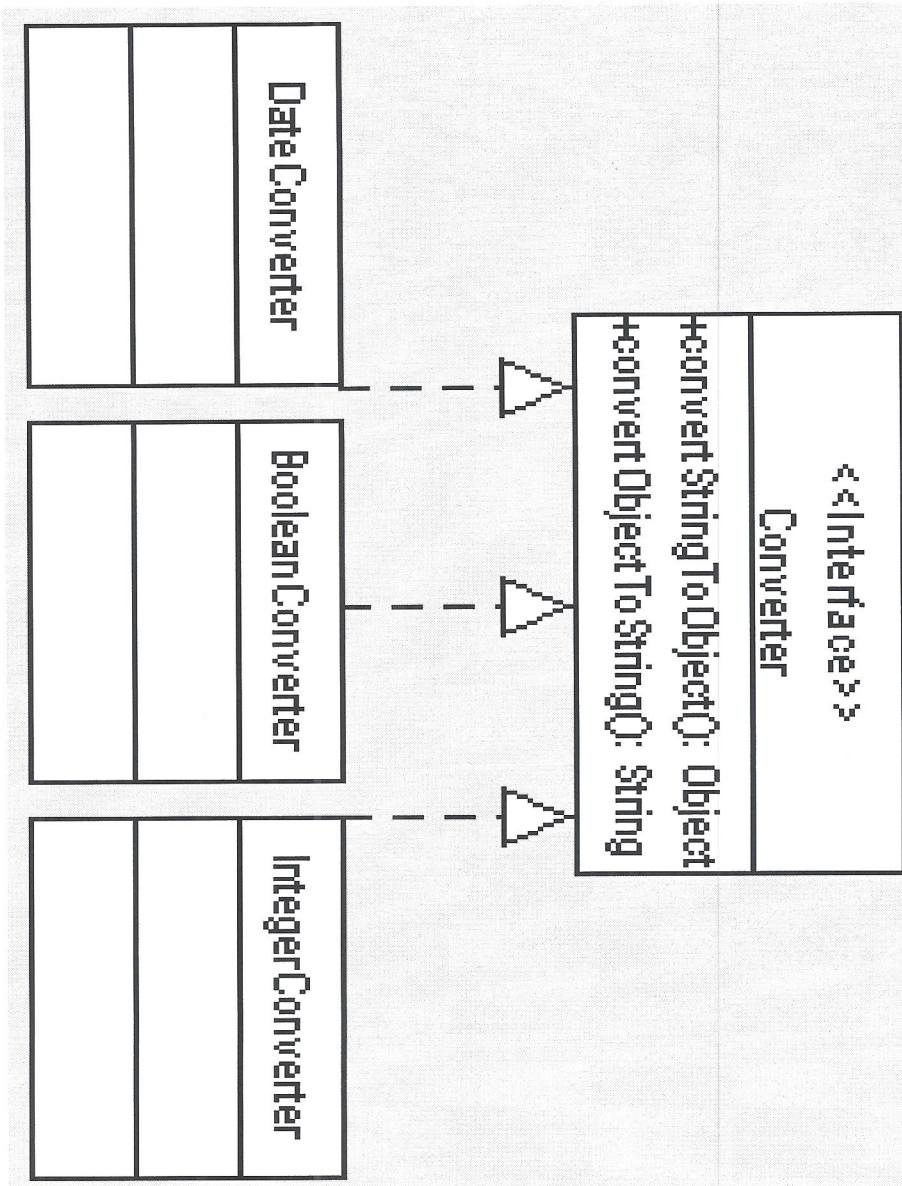


# Key Value Add: Validation

- What do I mean by Validation?
  - Ensuring values entered by the user are in the correct format:
    - Zip code: 5 digits from 0 – 9
    - Userid: at least 6 letters, no special characters
  - Type conversion:
    - In webapps, everything comes in as a String
    - Add support for converting valid data to native types such as Integer, Date, etc., before storing value to back end model object.

## **Key Value Add: Validation**



# Key Value Add: Validation

```
<faces:DeclareBean scope="session"
    id="DateValidator"
    className="basic.DateValidator"/>

<faces:TextEntry_Input id="pin"
    required = "true" modelType ="java.lang.Integer"
    rangeMinimum="1" rangeMaximum="1000" />

<faces:TextEntry_Input id="dob"
    required = "true" converter="DateValidator" />
```

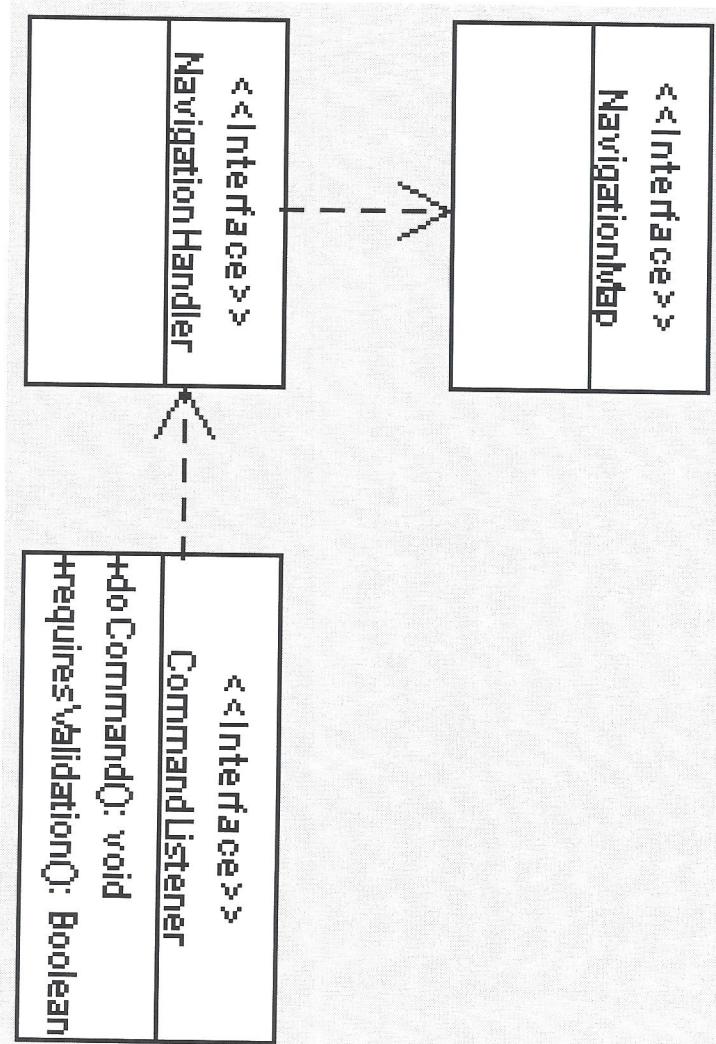
---

```
public class DateValidator implements Converter {
    public Object convertStringToObject(EventContext ctx,
        UIComponent component,
        String componentValue) {...}

    public String convertObjectToString(EventContext ctx,
        UIComponent component,
        Object modelValue) {...}
```

# Key Value Add: Navigation

- Flow from Page to Page in your webapp



# Key Value Add: Navigation

```
<faces:NavigationMap id="navMap" scope="session" >
<faces:outcome commandName="login" outcome="success"
    targetAction="forward" targetPath = "welcome.jsp" />
<faces:outcome commandName="login" outcome="failure"
    targetAction="forward" targetPath = "error.jsp" />
</faces:NavigationMap>

<faces:Form id="basicForm" model="LoginBean" navigationMapId="navMap">
<faces:Listener id="loginListener" scope="session"
    className="CommandListenerImpl"/>
Faces Content
<faces:Command_Button id="login" label="login"
    commandListener="loginListener"/>
</faces:Form>
```

# Key Value Add: Navigation

```
public class CommandListenerImpl implements CommandListener {  
    //...  
  
    public void doCommand(CommandEvent e, NavigationHandler nh) {  
        if (successful) {  
            nh.handleCommandSuccess(cmdName);  
        }  
        else {  
            nh.handleCommandException (cmdName, ce);  
        }  
    }  
}
```

# Key Value Add: Navigation

- Easily specify page flow
- Flexible: supports n way branching
- Will eventually allow out-of-band flow definition
- Will allow pluggable navigation with other systems, ie Struts.

# Summary: Faces Value Adds

- UI Components
- Event Mechanism
- Validation/Type Conversion
- Navigation
- Ease of Use

SESSION #169

D  
e  
m  
o

# JSF in Summary

- Simple, well-defined programming model
- Scalable: performance + ease of maintenance
- Highly extensible
- Tools, tools, tools
- Coming soon to a Container near you