DR3

Disciplina Regular 3

Desenvolvimento Web com Java EE

Graduação em Engenharia de Software - 2020

DR3

Etapa 2 Aula 2

JavaServer Pages

Para acompanhar pelo Moodle você deve estudar a etapa 3

DR3

Etapa 2 Aula 2 - Competências

- **Competências** Trabalhadas Nesta Etapa
  - **Construir aplicativos Java utilizando JSP, JSTL e EL.**
  - Construir aplicativos Java utilizando Servlets.
  - Escrever interfaces simples com HTML e CSS.

DR3

Etapa 2 Aula 2 - Checklist

- **Checklist** DR3:
  - Ter revisado os slides e feito os exercícios das aulas E1_A1, E1_A2 e **E2_A1**.
  - Ter lido o enunciado e tirado as dúvidas da **TP1**. Se possível ter iniciado a tarefa para 04/05/2020.
- **Checklist** PB:
  - Fazer uma primeira reunião com o grupo para tratar do plano de trabalho do ASSESMENT e da especificação do módulo.
  - Criar a conta no GitHub para o grupo.
  - Construir o caso de uso “Manter Cliente” como TP7 a ser entregue dia 04/05/2020.

DR3

Expression Language

DR3

O que é Expression Language

- Linguagem que facilita a manipulação de propriedades em JSP.
- Baseada em EcmaScript (JavaScript) e XPath: familiar para designers e programadores.
- Oferece uma sintaxe simples com operadores, objetos implícitos e literais:

```
Bom dia ${conta.titular}
Seu saldo é ${conta.saldo + conta.limite}
```

EL x Scriptlets x Ações JSP

DR3

- Scriptlets são trechos de código Java dentro do arquivo HTML. Devem ser evitados.
- As ações JSP **não funcionam**:
  - Com propriedades indexadas.
  - Com propriedades aninhadas.
  - Para acessar conjuntos de beans.
- EL oferece uma alternativa com sintaxe mais simples e sem as restrições das ações JSP.

Exemplo

DR3

- Um servlet insere um objeto de ContaCorrente na requisição e o redireciona para o JSP:

```
request.setAttribute("cc", cc);
RequestDispatcher rd = request.getRequestDispatcher("saldo.jsp");
rd.forward(request, response);
```

- O JSP saldo deve exibir o nome do titular e o saldo da conta:
  - Scriptlet:

```
<% ContaCorrente cc = (ContaCorrente) request.getAttribute("cc");
   if (cc != null)
      out.println("Saldo: " + cc.getSaldo()); %>
```

- Ações JSP:

```
<jsp:useBean id="cc" class="br.com.a.ContaCorrente" scope="request"/>
Saldo: <jsp:getProperty name="cc" property="saldo"/>
```

- EL:

```
Saldo: ${cc.saldo}
```

Expressões

- Devem ser escritas entre { } e precedidas de $:

```
${cc.saldo}
```

- Como alternativa, podem ser usados colchetes:

```
${cc["saldo"]}
```

- Podem ser acessadas propriedades aninhadas:

```
${cc.endereço.cidade}
```

- E propriedades indexadas:

```
${cliente.contas[0].saldo}
```

- Expressões podem ser inseridas em atributos HTML:

```
<font size="${util.size}" />
```

Operadores e Literais

DR3

- EL possui praticamente os mesmos operadores do Java e mais:
  - Aritméticos: **div, mod.**
  - Lógicos: **and, or, not.**
  - Relacionais: **eq, ne, lt, gt, le, ge.**
  - Teste vazio: **empty** (testa se é vazio ou null).
- Literais também são semelhantes a Java, mas strings podem ser escritas entre aspas simples.
- Exemplos:

```
${empregado.salario * 1.1}

Parcelas: ${cliente.valor > 100 ? 5 : 2}
```

Objetos Implícitos

DR3

- Vários objetos estão disponíveis para facilitar o acesso a valores e propriedades:
  - param, paramValues: parâmetros de requisição.
  - header, headerValues: cabeçalhos de requisição.
  - cookie: representa os cookies enviados pelo browser.
  - initParam: parâmetros de inicialização.
- Exemplo de uso:

```
${param["login"]}
```

ou

${param.login}

Objetos Implícitos

DR3

- Caso seja necessário especificar onde se encontra um atributo:
  - pageScope.
  - requestScope.
  - sessionScope.
  - applicationScope.
- Exemplo de uso:

```
${requestScope.conta.saldo}
```

Exemplo

DR3

- A exibição do nome do titular e do saldo da conta fica bem mais simples com a utilização de EL:

```
Nome: ${cc.titular}
Saldo: ${cc.saldo}
```

ou

```
Nome: ${requestScope.cc.titular}
Saldo: ${requestScope.cc.saldo}
```

Exercício

DR3

- Crie um projeto web com uma página index.jsp que exiba uma drop-down list com Real, Dólar e Euro e um input para a digitação de um valor monetário.
- Crie um POJO Moeda que contenha três atributos: um para cada moeda do drop-down list. Os métodos “set” devem converter Real para Dólar 5X1 e Euro 6X1.
- Crie o servlet **ConversorServlet**, que recebe os dados do formulário, instancia um objeto Moeda e o coloca na requisição, redirecionando de volta para index.jsp.
- A página index.jsp deve exibir os valores usando EL.

DR3

JSTL

JSP Standard Tag Library

DR3

- Conjunto de bibliotecas de tags que são usadas em substituição a scriptlets.
- JSTL + EL = JSP sem código Java.
- Possui cinco bibliotecas:
  - Core: controle de fluxo e url.
  - XML: manipulação de XML.
  - Internationalization: formatação e localização.
  - Database: manipulação de SQL.
  - Functions: tratamento de strings.

Usando Bibliotecas de Tags

DR3

- Para utilizar qualquer biblioteca de tags é necessário incluir a diretiva **taglib** no jsp:

```
<%@ taglib uri="biblioteca" prefix="prefixo" %>
```

- O atributo **uri** indica o nome da biblioteca, que normalmente é descrito por uma url e **prefix**. Esse último informa a palavra que deve ser usada antes do nome da tag.

- Para usar a biblioteca "Core" da JSTL:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

- A JSTL deve ser baixada e os arquivos jar disponibilizados para a aplicação (usar Maven).

Core JSTL

DR3

- Biblioteca básica que possui tags para:
  - Controle de fluxo (condições e repetições).
  - Manipulação de atributos.
  - Gerenciamento de URL (inclusão de conteúdo, redirecionamento, reescrita de url).
  - Tratamento de erros.
  - Controle da saída.

Repetição

DR3

- A EL resolve o problema referente ao acesso de índices de propriedades e JavaBeans, mas não contém uma forma de “caminhar” pelo conjunto.

- Solução: usar EL com a tag forEach.

```
<c:forEach var=“objeto"  items=“coleção">
</c:forEach>
```

Exemplo

- A listagem de contas de uma agência consultada pelo gerente é feita usando forEach:

```
<c:forEach var="conta" items="${contas}">
    Titular: ${conta.titular} <br>
    Saldo: ${conta.saldo} <br><br>
</c:forEach>
```

Condições

- Condições simples podem ser testadas com a tag **if**:

```
<c:if test="condição">
</c:if>
```

- Condições múltiplas são feitas com **choose**:

```
<c:choose>
    <c:when test="condição" >
    </c:when>
</c:choose>
```

Exemplo

- Se o saldo da conta for negativo, ele deve aparecer em vermelho:

```
<c:choose>
    <c:when test="${conta.saldo >= 0}">
        Saldo: <font color="black">${conta.saldo}</font>
    </c:when>
    <c:when test="${conta.saldo < 0}">
        Saldo: <font color="red">${conta.saldo}</font>
    </c:when>
</c:choose>
```

DR3

Atribuição

- Os valores de atributos podem ser alterados com as duas formas da tag **set.**
  - Com o atributo value:

```
<c:set var="nome" scope="escopo" value="valor"/>
```

  - Com o corpo da tag:

```
<c:set var="nome"> valor </c:set>
```

- Para remover um atributo, coloque o valor null ou use a tag **remove.**

DR3

Exercício

- Construir uma aplicação simples com um formulário da entidade Contato, com nome, email e fone.

- Construir um servlet que obtenha os dados passados pelo formulário, valide os dados conforme as regras abaixo e redirecione de volta para o formulário:
  - nome → obrigatório.
  - email → obrigatório.
  - fone → obrigatório, numérico.

- Quando redirecionar, exibir uma mensagem de “Sucesso” ou uma lista de mensagens de erro.