



# Java Server Faces Expression Language

Prof. Rodrigo Henrique Cunha Palácios  
rodrigopalacios@utfpr.edu.br

Slides fornecidos pelo professor Prof. Edson Shozo Nishi



# Expression Language

- Motivação para usar a expression language
  - Comparar com a EL do JSP 2.0
- Acessar propriedades de beans
- Submeter propriedades de beans
- Acessar elementos de collections
- Usar objetos implícitos e operadores



# Vantagens da JSF EL

- Notação prática para acessar propriedades de beans
  - `bean.getPropriedade()`
  - `#{bean.propriedade}`
- Acesso simples a elementos de collections
  - Referenciar elemento de array, List, Map
    - `#{variavel[indiceOuChave]}`



# Vantagens da JSF EL

- Acesso sucinto a parâmetros, cookies e outros dados de request
- Pequeno, porém útil conjunto de operadores
- Saída condicional
  - `{teste ? opcao1 : opcao 2}`
- Conversão automática de tipos
- Valores vazios em vez de mensagens de erros...



# JSF EL x JSP 2.0 EL

- Pode ser usada somente em tags JSF
  - Requer declaração de taglib
  - Usa `{var}`
  - Representa valores submetidos e de output
  - Analisa request, session, application e managed beans
- Pode ser usada em qualquer lugar da página JSP
  - Não requer declaração de taglib
  - Usa `${var}`
  - Representa valores de output somente
  - Analisa request, session e application, somente



# Mostrando valores de beans

- `#{var.nomePropriedade}`
  - Significa buscar em `HttpServletRequest`, `HttpSession`, `ServletContext` e managed beans (nessa ordem) e mostrar a propriedade especificada
  - Deve ser utilizada com um atributo em uma tag JSF



# Acessando propriedades internas

- `#{var.prop1.prop2}`
  - Primeiro busca as variáveis de escopo e managed beans para uma entrada
  - Depois acessa prop1
  - Então acessa prop2
  - Pode ser aninhada arbitrariamente



# Três significados de #{...}

- Designar valores de saída
  - `<h:outputText value="#{emp.endereco}"/>`
- Designar valores submetidos
  - `<h:inputText value="#{emp.endereco}"/>`
- Designar chamada de método após submissão
  - `<h:commandButton value="Ok" action="#{emp.processar}" />`





# Acessando collections

- Equivalência de notações por ponto ou array
- Formas equivalentes
  - `#{nome.propriedade}`
  - `#{nome["propriedade"]}`
- Razões para usar notação array
  - Para acessar collections (arrays, lists, etc)
  - Calcular nome de propriedade em tempo de request
    - `#{var1[var2]}` (sem haspas em var2)
  - Usar nomes que seriam ilegais para variáveis Java
    - `#{var["var-var"]}`
    - `#{var["var.var"]}`



# Acessando collections

- #{nome[nomeEntrada]}
- Funciona para:
  - Array
    - array[indice]
  - List
    - list.get(indice) / list.set(indice, valor)
  - Map
    - map.get(chave) / map.put(chave, valor)



# Objetos implícitos

- JSF possui quase as mesmas variáveis pré-definidas do que JSP 2.0
- facesContext
  - FacesContext
  - `{facesContext.externalContext.session.id}`
- param e paramValues
  - Parâmetros de request
  - `{param.valor}`
- header e headerValues
  - Request headers
  - `{header.Accept}` ou `{header["Accept"]}`
  - `{header["Accept-Encoding"]}`



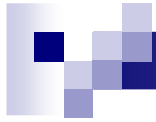
# Objetos implícitos

- cookie
  - Objeto Cookie (não seu valor)
  - `{cookie.userCookie.value}`
  - `{cookie["userCookie"].value}`
- `initParam`
  - Parâmetros de inicialização de contexto
- `requestScope`, `sessionScope`, `applicationScope`
  - Escopos fixos
- Problema
  - Usar objetos implícitos geralmente é uma solução pobre juntamente com MVC



# Operadores

- Aritméticos
  - + - \* / div % mod
- Relacionais
  - == eq != ne < lt > gt <= le >= ge
- Lógicos
  - && and || or ! Not
- Empty
  - True para null, string vazia, array vazia, list vazia, map vazio. False em todos os outros casos
- ADVERTÊNCIA
  - Use com parcimônia para preservar o modelo MVC



# Operadores

- Exemplos:

- $\#{3+2+1}$

- $\#{3/4 == 0.75}$



# Expressões condicionais

- `{teste ? expressao1 : expressao 2}`
- Semelhante a operador de Java
- Problemas
  - Relativamente fraco
    - `c:if` e `c:choose` da JSTL são melhores
  - Tenta o programador a colocar lógica de negócios ou processamento em páginas
  - Deve ser usado somente para lógica de apresentação
    - Ainda assim, considerando alternativas



Dúvidas?