

Aula 14

JSP

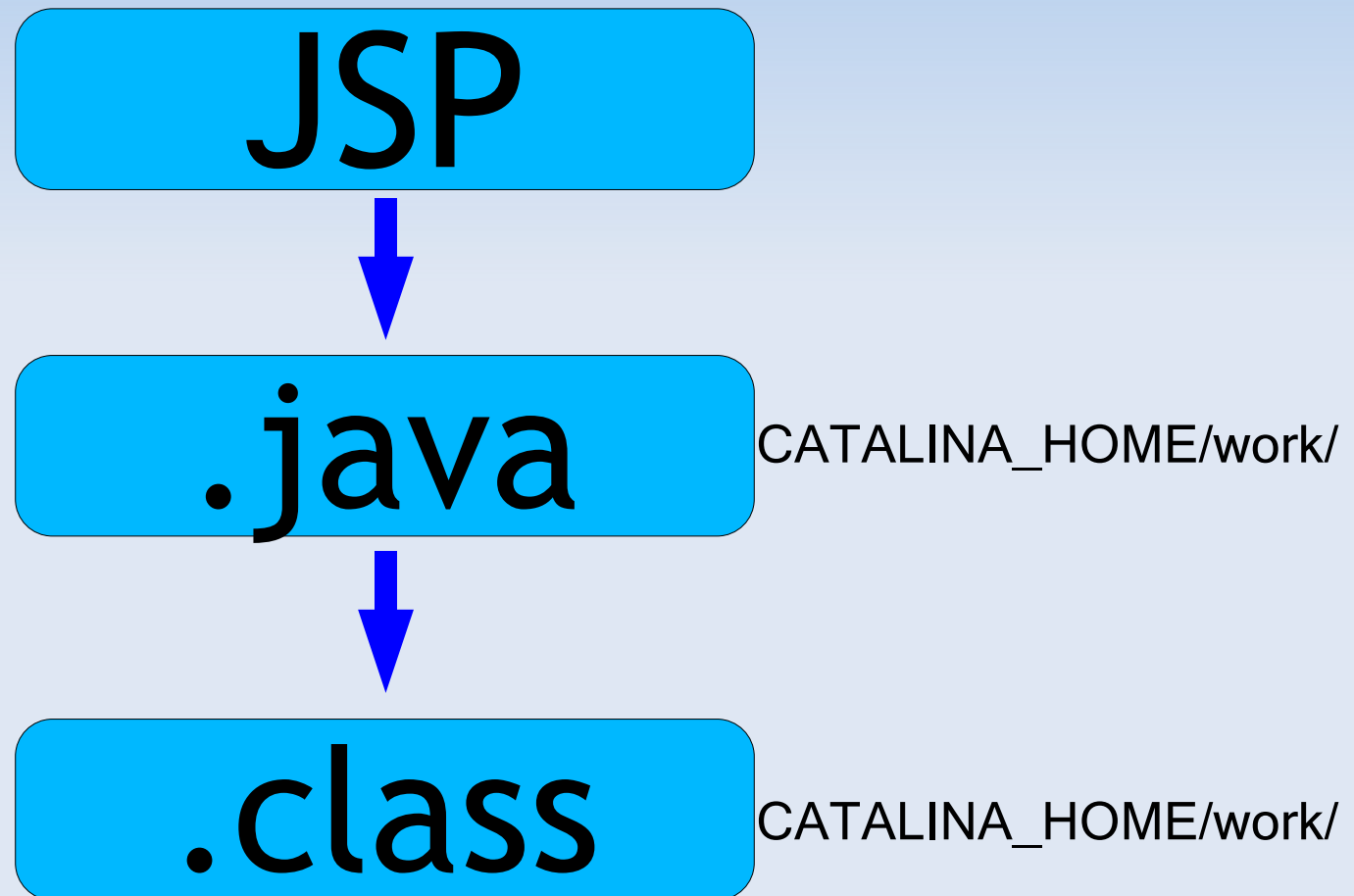
Objetivos:

- O que são JSP;
- Diferença entre JSP e servlet;
- Como criar e implantar JavaServer Pages;
- Como especificar informações globais do JSP com diretivas;
- Como utilizar ações para manipular JavaBeans em uma JSP;

O que são JSP's

- JSP é um servlet;
- Enquanto servlets geram código html de código java, JSP incorporam código java em código html;
- Separação de papéis; O código html fica mais separado das tarefas do aplicativo;
- Antes da execução uma jsp precisa ser convertida em uma servlet;

Tradução e execução



Visão geral:

- Diretivas;
 - `<%@ diretiva...%>`
- Ações;
 - `<jsp: >`
- Elementos de script;
 - `<% %>` scriptlets
 - `<%= %>` atribuições
 - `<%! %>` declarações
- Biblioteca de tags

Um primeiro Exemplo

- Crie primeirajsp.jsp:

```
<%@page import="java.util.*" %>

<html>
<%
Calendar data = Calendar.getInstance();
String resp = "HORA:" + data.get(Calendar.HOUR) + ":" +
    data.get(Calendar.MINUTE);
%>
<head>
<title>
PrimeiraJsp
</title>
</head>
<body bgcolor="#ffffff">
<h1>
<%=resp %>
</h1>
</body>
</html>
```

Escopo de objetos

- Escopo Application:
 - qualquer jsp pode manipular tais objetos;
- Escopo Session:
 - durante toda sessão de navegação do cliente;
- Escopo Page:
 - só existem na página que os definem;
- Escopo Request:
 - estão disponíveis até o fim da solicitação;

Objetos implícitos

- escopo application:
 - application
- escopo session:
 - session
- escopo request:
 - request

Objetos implícitos ... cont

- escopo page:
 - out
 - page
 - response

Script - exemplo:

```
<html>
<head>
<title>
Segundo Exemplo de JSP
</title>
</head>
<body bgcolor="#ffffff">
<%
String nome = request.getParameter("nome");
if( nome == null ){
    %>
    <form action="segundajsp.jsp">
        Nome: <input type="text" name="nome" />
        <input type="submit" />
    </form>
    <%
}
else{
    %>
    <h1>
        Seja bem vindo <%=nome %>!!<br />
        Ao Exemplo de scriptlet
    </h1>
    <%} %></body></html>
```

Diretivas

- `<%@taglib %>`
- `<%@page import = "java.util.*" %>`
- `<%@include file = "" %>`

Script - exemplo:

- Crie terceiroajsp.jsp:

```
<html>
<head>
<title>
terceirajsp
</title>
</head>
<body bgcolor="#ffffff">
<%@include file="primeirajsp.jsp" %>
<%@include file="segundajsp.jsp" %>
</body>
</html>
```

Ações padrão

- Página 967
 - <jsp:include>
 - <jsp:forward>
 - <jsp:plugin>
 - <jsp:param>
 - <jsp:useBean>
 - <jsp:setProperty>
 - <jsp:getProperty>

Ação: <jsp:useBean>

- <jsp:useBean>
- <jsp:setProperty>
- <jsp:getProperty>

Script - exemplo:

- Crie a classe Empregado

```
package exemplojsp;  
  
import java.io.*;  
  
public class Empregado implements Serializable{  
    private String nome, telefone, salario, endereco;  
  
    ... métodos getters e setters omitidos ...  
}
```

Script - exemplo:

- Crie o arquivo quartajsp.jsp

```
<%@page import="exemplojsp.*" %>
<html>
<head>
<title>
quartajsp
</title>
</head>
<jsp:useBean id="emp" scope="session" class="exemplojsp.Empregado" />
<jsp:setProperty name="emp" property="*" />
<body bgcolor="#ffffff">
<form action="quartajsp.jsp">
  Nome:<input type="text" name="nome" /><br />
  Telefone:<input type="text" name="telefone" /><br />
  Salario:<input type="text" name="salario" /><br />
  Endereco:<input type="text" name="endereco" /><br />
  <input type="submit" value="OK" />
</form>
Nome:<jsp:getProperty name="emp" property="nome"/>
Telefone:<jsp:getProperty name="emp" property="telefone"/>

</body>
</html>
```


Script - exemplo:

- Incrementando com acesso a banco de dados:
- Crie o método reqCon na classe AcessoBd

```
public static Connection reqCon(){
    try {
        Class.forName("org.firebirdsql.jdbc.FBDriver");
        return DriverManager.getConnection(
            "jdbc:firebirdsql:instrutor/3050:d:\\aulaservlet.gdb",
            "sysdba", "masterkey");
    } catch (SQLException ex) {ex.printStackTrace();}
    catch (ClassNotFoundException ex) {ex.printStackTrace();}

    return null;
}
```

Script - exemplo:

- Incrementando com acesso a banco de dados:
- Crie o método gravar na classe AcessoBd

```
public static void gravar(Empregado emp) {  
    try {  
        Connection con = reqCon();  
        PreparedStatement pstmt = con.prepareStatement("insert into empregados"+  
            "(nome, salario, endereco, telefone) values (?, ?, ?, ?)");  
        pstmt.setString(1, emp.getNome());  
        pstmt.setString(2, emp.getSalario());  
        pstmt.setString(3, emp.getEndereco());  
        pstmt.setString(4, emp.getTelefone());  
        pstmt.execute();  
        pstmt.close();  
        con.close();  
    } catch (SQLException ex) {ex.printStackTrace();}  
}
```

Script - exemplo:

- Modifique o arquivo quartajsp.jsp

```
Telefone:<jsp:getProperty name="emp" property="telefone"/>
<%
String nome = emp.getNome();
if( nome != null && nome.length() > 0 ){
    AcessoBd.gravar(emp);
}
%>

</body>
```

Modificando o banco de dados

```
create table CONTATO(  
    codigo INTEGER NOT NULL,  
    nome varchar(32),  
    endereco varchar(32),  
    telefone varchar(32),  
    nascimento date  
);  
CREATE TABLE DUAL_GERADORCONTATO (  
    ZERO INTEGER  
);
```

Modificando a classe conexao

```
public static int proximoIndice(String generator){  
    try {  
        CallableStatement cstm = con.prepareCall("CALL NEXT VALUE FOR " +  
generator);  
        ResultSet rs = cstm.executeQuery();  
        rs.next();  
        return rs.getInt(1);  
    } catch (SQLException e) {  
        e.printStackTrace();  
        return 0;  
    }  
}
```

Dúvidas?

Exercícios

- O que é uma jsp?
- Como inserir código java em um jsp?
- O que são scriptlets?

Fim