

APLICAÇÕES DE LINGUAGEM DE PROGRAMAÇÃO ORIENTADAS À OBJETOS 7968-90_43701_R_E1_20232

CONTEÚDO

Usuário	
Curso	APLICAÇÕES DE LINGUAGEM DE PROGRAMAÇÃO ORIENTADAS À OBJETOS
Teste	QUESTIONÁRIO UNIDADE IV
Iniciado	
Enviado	
Status	Completada
Resultado da tentativa	
Tempo decorrido	
Resultados exibidos	Todas as respostas, Respostas enviadas, Respostas corretas, Comentários, Perguntas respondidas incorretamente

Pergunta 1

0,25 em 0,25 pontos




JSP é a abreviação de "Java Server Pages", uma linguagem que tem como objetivo gerar um conteúdo dinâmico para as páginas *web*, ao contrário das páginas HTML que geram páginas mais estáticas. Em páginas .jsp, é possível escrever os códigos HTML, juntamente, com os códigos JSP (Java). Basicamente, elas funcionam da seguinte forma: um servidor de aplicações recebe uma requisição para uma página JSP, interpreta esta página gerando a codificação HTML e retorna ao cliente o resultado de sua solicitação. A tarefa é realizada em tempo real pelo servidor (com base no tempo de resposta de um servidor Http). Para criarmos uma Página JSP utilizamos *tags* específicas. A seguir está um programa escrito na linguagem JSP, de forma que, em algumas linhas, estão faltando *tags* JSP específicas:

```
" ...
<html>
<head>
<title> Página JSP </title>
</head>
<body>
<h1> Um exemplo simples de Página JSP </h1>
<alfa
include file="menu.jsp"
%>
<beta
float res;
void somar(float a, float b) {
    res = (a + b);
}
void multiplicar(float a, float b) {
    res = (a * b);
}
%>
<gama
somar(6, 7);
%>
<delta
res
%>
</body>
</html>
..."
```

No código JSP anterior, no lugar das *tags* "<alfa", "<beta", "<gama" e "<delta", devemos colocar quais das seguintes *tags* específicas do JSP (respectivamente)?

Resposta Selecionada:  e. "<%@" ; "<%!" ; "<%%" ; "<%=."

- Respostas:
- a. "<%!" ; "<%@" ; "<%=." ; "<%%".
 - b. "<%!" ; "<%=." ; "<%" ; "<%%@".
 - c. "<%@" ; "<%=." ; "<%!" ; "<%%".
 - d. "<%=." ; "<%@" ; "<%%" ; "<%!".
 -  e. "<%@" ; "<%!" ; "<%" ; "<%=."

Comentário da resposta: Resposta: E
Comentário: de acordo com o código apresentado no enunciado temos que:

- Logo depois da "tag alfa", vem a linha "include file=menu.jsp" que sugere que se está inserindo naquela posição da página, o conteúdo de outra página (a página "menu.jsp"). Isto sugere a utilização de uma "diretiva include", representada pela *tag* <%@.

- Logo depois da "tag beta", vem as linhas:

```
float res;  
void somar(float a, float b) {  
    res = (a + b);  
}
```

... que sugerem a declaração de atributos e de métodos, ou seja, são "declarações" definidas pela tag JSP <%!

- Logo depois da "tag gama" vem a linha:

```
somar(6, 7);
```

... que sugere um comando Java que chama o método existente na página, que é um método que não retorna à informação, ou seja, são linhas de comando Java que devem ser definidas dentro da tag JSP de "scriptlet", ou seja, <%.

- Logo depois da "tag delta", vem a linha:

```
res
```

... que sugere o nome de uma variável Java que, provavelmente, possuirá um valor definido no programa, valor este que será mostrado naquela posição da página, (observe que a linha não termina com ponto e vírgula) o que sugere que se trata do que chamamos de "expressão", representado pela tag JSP <%=.

Pergunta 2

0,25 em 0,25 pontos



Sobre a JSP, qual é a alternativa correta?

Resposta



c.

Selecionada:

Uma JSP é basicamente uma "página web" criada, utilizando-se da linguagem HTML misturada com os códigos em Java.

Respostas:

a. Uma página JSP é lida e traduzida por qualquer *browser* independentemente de onde ela estiver localizada.

b. A JSP pode conter o código Java em qualquer *tag* HTML.



c.

Uma JSP é basicamente uma "página web" criada, utilizando-se da linguagem HTML misturada com os códigos em Java.

d. É possível criar os sistemas *desktop* (com as janelas de formulários) a partir da linguagem JSP.

e. As JSPs somente podem ser rodadas a partir de servidores HTTP (ou servidores *web*).

Comentário

Resposta: C

da resposta:

Comentário: JSP é a sigla de "Java Server Pages", e que são arquivos codificados com os códigos na linguagem HTML (como se fossem as páginas *web*) misturado, juntamente, com os códigos em Java.

Uma JSP simplifica a geração do conteúdo dinâmico para os desenvolvedores *web*, e tem o objetivo de facilitar a construção dessas páginas dinâmicas.

O arquivo de uma página com código em JSP deve ser salvo com a extensão ".jsp". Ele é um arquivo de texto simples e pode ser aberto e editado a partir de um editor de texto simples (como, por exemplo, o Bloco de Notas). Para que uma página JSP funcione corretamente, ela deve ser aberta a partir de um *browser*, porém ela deve estar inserida (localizada) em um "Servidor de Aplicação" em funcionamento (em um diretório específico do "Apache Tomcat"). Todo código Java deve estar inserido em *tags* especificamente criadas para o JSP.

Pergunta 3

0,25 em 0,25 pontos



Sobre os Servidores de Aplicação Java, qual é a alternativa **incorreta**?

Resposta



a.

Selecionada:

Um Servidor de Aplicação é um sistema que permite abrir a um único usuário as páginas construídas para a *web*.

Respostas:



a.

Um Servidor de Aplicação é um sistema que permite abrir a um único usuário as páginas construídas para a *web*.

b.

É um programa que, juntamente com os servidores *web*, permite disponibilizar os sistemas construídos (criados) em determinadas linguagens dinâmicas (como, por exemplo: em JSP).

c.

Existem vários programas servidores de aplicação disponíveis no mercado (uns proprietários e outros gratuitos).

d.

O Tomcat é um Servidor de Aplicação que, juntamente com o Servidor *Web* Apache, pode ser baixado gratuitamente na *web*.

e.

Uma página HTML pode ser acessada (visualizada) se ela estiver localizada em um Servidor *Web*.

Comentário

Resposta: A

da resposta:

Comentário: os Servidores de Aplicação Java foram criados para rodarem o sistema em redes corporativas, de forma a permitir o acesso de vários usuários. Já um Servidor *Web* (ou Servidor HTTP) é utilizado para disponibilizar as páginas *web* (ou páginas HTML) a diversos usuários, ao mesmo tempo. Juntando os dois servidores, construiu-se um sistema que possibilita disponibilizar a vários usuários sistemas construídos em linguagens dinâmicas e que funcionam dinamicamente em qualquer *browser* instalado no usuário. Existem no mercado, vários Servidores de Aplicação sendo que alguns são gratuitos e outros são proprietários (não gratuito), tais como:

O Jboss (da Red Hat Jboss Middleware)

O WebSphere (da IBM)
O GlassFish Server (*open source*)
O Apache Tomcat (*open source*)
... este último foi o que utilizamos como exemplo neste curso, o qual foi preparado para rodar "Java Server Pages".

Pergunta 4

0,25 em 0,25 pontos



Sobre JSTL, qual é a alternativa **incorreta**?

Resposta

☒ d.

Selecionada:

Para que uma página com código JSTL funcione, a partir de um Servidor de Aplicação, é necessário que sejam inseridas algumas bibliotecas (arquivos com a extensão .jstl) no diretório de bibliotecas do servidor.

Respostas:

a. A sigla JSTL significa "JSP Standard Tag Library".

b. A JSTL é uma biblioteca de TAGS criadas para funcionar com JSP.

c.

Com as *tags* da JSTL é possível escrever as páginas JSPs (páginas que possuem a dinâmica JSP) sem se utilizar de código Java, deixando parte dos elementos na forma de *tags*.

☒ d.

Para que uma página com código JSTL funcione, a partir de um Servidor de Aplicação, é necessário que sejam inseridas algumas bibliotecas (arquivos com a extensão .jstl) no diretório de bibliotecas do servidor.

e. A JSTL aumenta a facilidade do *design* e da compreensão do código para os programadores *web*.

Comentário da resposta:

Resposta: D

Comentário: o JSTL é uma biblioteca de classes, que permite que uma página JSP contenha os códigos específicos em forma de *tags*, com funcionalidades próprias que agilizam e facilitam o desenvolvimento das páginas dinâmicas. Por ser uma biblioteca de classes, ela está contida em arquivos com a extensão ".jar" (já que não existem arquivos com a extensão ".jstl").

Pergunta 5

0,25 em 0,25 pontos



Sobre os programas em Java que geram os relatórios no Excel®, qual é a alternativa correta?

Resposta

☒ b.

Selecionada:

Para gerarmos um arquivo de relatório do Excel®, é necessário adicionarmos, no projeto, uma biblioteca específica de classes (como, por exemplo, a biblioteca "jxl.jar", que permitem esta interação do Java com arquivos .xls).

Respostas:

a.

Depois de se compilar com sucesso o programa que gera os arquivos de relatório em Excel® utilizando-se da biblioteca jxl.jar, para rodar o programa a biblioteca não é mais necessária.

☒ b.

Para gerarmos um arquivo de relatório do Excel®, é necessário adicionarmos, no projeto, uma biblioteca específica de classes (como, por exemplo, a biblioteca "jxl.jar", que permitem esta interação do Java com arquivos .xls).

c. Não é possível criar programas em Java que gere relatórios em Excel®, mas sim, apenas, em arquivos PDF.

d.

Com um programa na linguagem Java, é possível gerar planilhas Excel® com as informações, porém, não é possível formatar a planilha, a não ser que se abra o arquivo no próprio Excel®.

Para que a planilha do Excel® seja preenchida com os dados de um banco de dados, basta que adicionemos o *Driver* do Banco de Dados, sem necessitar adicionar a biblioteca de classes do Excel®.

Comentário da resposta:

Resposta: B



Comentário: para que possamos acessar as planilhas do Excel® com os programas criados na linguagem Java, é necessário que o projeto tenha acesso a uma biblioteca de classes específica (um arquivo .jar, como, por exemplo, o arquivo "jxl.jar"). Tanto na fase de desenvolvimento do sistema, quanto ao longo da sua utilização quando em funcionamento, se ele vai acessar algum arquivo de planilha do Excel®, é necessário que esta biblioteca esteja presente, e tenha sido incluída no diretório de bibliotecas do Java, ou pertencente ao Servidor de Aplicação (quando o sistema foi criado para a *web*, por exemplo), ou em um diretório criado especificamente para isso no projeto (quando o sistema foi criado para o *desktop*).

Pergunta 6

0,25 em 0,25 pontos



Sobre a core "catch" do JSTL, qual é a alternativa **incorreta**?



Resposta Selecionada:	 a. Programando em JSTL, deve-se tomar cuidado pois no interior de uma <i>tag</i> "<c:catch>", não funcionarão linhas de código em que se utiliza outras <i>tags core</i> .
Respostas:	 a. Programando em JSTL, deve-se tomar cuidado pois no interior de uma <i>tag</i> "<c:catch>", não funcionarão linhas de código em que se utiliza outras <i>tags core</i> . <p>b. A <i>tag</i> "<c:catch>" é uma <i>tag</i> JSTL utilizada para o tratamento de erros.</p> <p>c. A <i>tag</i> "<c:catch>" captura qualquer erro ou exceção derivada da classe Throwable, do Java.</p> <p>d. A <i>Tag</i> "<c:catch>" captura os erros e as exceções desde que venham a ocorrer na execução dos códigos descritos no interior da <i>tag</i>.</p> <p>e. Com esta <i>tag</i> é possível se capturar de exceções que podem ocorrer no corpo da página ao longo (e depois) do processo de abertura da página.</p>
Comentário da resposta:	Resposta: A Comentário: a <i>core</i> do JSTL é um conjunto de <i>tags</i> que permitem uma programação mais ágil, de forma que contém <i>tags</i> que geram estruturas de repetição, outras de estruturas condicionais, criando essas estruturas lógicas que podem ser programadas, diretamente, na linguagem de <i>tags</i> (próxima aos códigos que já estão sendo utilizados na página). A <i>core catch</i> foi gerada com a mesma ideia de funcionalidade da estrutura de exceções do Java, mas utilizando-se da codificação em forma de <i>tags</i> . Muitas vezes, é preciso exibir para o usuário qual é o erro ocorrido na página, para que ele possa saber qual é o procedimento correto a se fazer com aquela situação (nem que seja procurar a equipe de <i>Help Desk</i> da empresa em que trabalha) de forma que, para isso, pode-se utilizar a <i>tag</i> "<c: if>", que implementa um laço de decisão, criando, assim, uma lógica que se acontecer algum erro (ou alguma exceção), é possível exibir o procedimento adequado a cada erro ocorrido. Desta forma, as <i>tags</i> podem ser "misturadas" uma estrutura dentro da outra, a fim de se definir o procedimento mais adequado ao sistema.

Pergunta 7

0 em 0,25 pontos



Sobre o HTML, qual é a alternativa **incorreta**?

Resposta Selecionada:	 d. As <i>tags</i> são os marcadores, sempre descritos entre os sinais "<" e ">", sendo textos que definem uma área da página que será formatada ou que terá uma função específica na página.
Respostas:	<p>a. O HTML é uma linguagem utilizada no desenvolvimento de páginas <i>web</i>.</p> <p>b. Esta linguagem permite a criação de arquivos que podem ser executados, ou abertos em, praticamente, qualquer tipo de computador (desde que o micro possua um <i>browser</i> instalado).</p> <p>c. Para se criar as páginas HTML, é necessário, além do conhecimento da linguagem, apenas, um editor de texto simples, como, por exemplo, o Bloco de Notas.</p> <p>d. As <i>tags</i> são os marcadores, sempre descritos entre os sinais "<" e ">", sendo textos que definem uma área da página que será formatada ou que terá uma função específica na página.</p> <p> e. A sigla HTML significa <i>Hyper Text Meta-Language</i> de forma que é uma linguagem de hipertextos com significados que vão além da linguagem da internet.</p>

Pergunta 8


0,25 em 0,25 pontos



Sobre os elementos de JSP, analise às seguintes afirmativas:

- I. Uma Expression é um trecho de Código Java, inserido no JSP, cujo resultado da execução da linha de código deve ser uma String, de forma que o texto resultante será exibido no lugar onde ela se encontra na página;
- II. Uma Scriptlet define alguns elementos de uma classe como atributos e métodos dentro de uma página *web*;
- III. Os Declarations são blocos de código na linguagem Java, inseridos em uma página JSP;
- IV. A diretiva "Include" inclui o texto do código de um arquivo externo em uma página JSP, no lugar exato onde aquela declaração da diretiva está localizada.

De acordo com as afirmativas, estão corretas:

Resposta Selecionada:	 c. Apenas as afirmativas I e IV estão corretas.
Respostas:	<p>a. Apenas as afirmativas I e II estão corretas.</p> <p>b. Apenas as afirmativas II e III estão corretas.</p>

- ☒ c. Apenas as afirmativas I e IV estão corretas.
- ☐ d. Apenas as afirmativas III e IV estão corretas.
- ☐ e. Todas as afirmativas estão corretas.

Comentário da resposta: Resposta: C
Comentário: vimos que existem *tags* JSP com as funcionalidades específicas caracterizadas pelo símbolo "<%>" (que só são utilizados pelas páginas *web* dinâmicas). Um Scriptlet, caracterizado por um bloco formado entre as *tags* "<%>" e "%>" é uma *tag* que forma um bloco onde existirá um programa codificado, diretamente, na linguagem Java. Nesta *tag* não declaramos atributos e nem métodos de uma classe Java, mas, apenas, podemos nos utilizar de atributos e métodos que foram declarados em "Declarations", que é um elemento JSP caracterizado por um bloco formado entre as *tags* "<%!" e "%>".

Pergunta 9

0,25 em 0,25 pontos



Sobre os programas em Java que geram os relatórios no Excel®, qual é a alternativa correta que define o passo seguinte à instância de um objeto do tipo Label, como, por exemplo:

Label objLabel = new Label(numColuna, numLinha, strTexto);
... a fim de que o texto definido na String strTexto seja, efetivamente, inserido na célula localizada na linha e coluna definidas também no parâmetro do método construtor da classe Label.

Resposta Selecionada: ☒ b.
Deve-se adicionar o objeto Label à planilha com o comando "addCell(...)", método este que leva como parâmetro o objeto Label, e que é acionado a partir de um objeto do tipo "WritableSheet", que representa a planilha.

Respostas:

- ☐ a.
Deve-se acionar o método "writeLabel" do objeto que representa a planilha – como, por exemplo: sheet.writeLabel(objetoLabel).
- ☒ b.
Deve-se adicionar o objeto Label à planilha com o comando "addCell(...)", método este que leva como parâmetro o objeto Label, e que é acionado a partir de um objeto do tipo "WritableSheet", que representa a planilha.
- ☐ c.
O texto definido para o objeto Label já estará na planilha sem necessitar de mais algum passo, já que este objeto representa, diretamente, a célula definida na planilha.
- ☐ d.
Deve-se transformar o objeto do tipo Label em um objeto do tipo Cell, já que não há como inserir um objeto do tipo Label em uma planilha.
- ☐ e.
Deve-se gerar um objeto do tipo Cell, pertencente à planilha e, depois, acionar o método "show()" do objeto que representa a planilha, para mostrar esta célula.

Comentário da resposta: Resposta: B
Comentário: quando se está construindo uma planilha Excel®, a partir de um programa codificado na linguagem Java, os passos para a criação podem ser resumidos à/a:
- Criação do arquivo de planilha (gerando-se o objeto que representa a planilha);
- Geração do texto que será salvo na planilha (uma String) inserindo-o em um objeto do tipo Label;
- Inserção do Label criado em uma célula da planilha utilizando-se o método "addCell(...)" cujo parâmetro leva o objeto do tipo Label com o texto a ser inserido;
- Salvar o arquivo, fechando-o.

Pergunta 10

0,25 em 0,25 pontos



Sobre os programas em Java que geram relatórios no Excel®, tem-se em um programa a seguinte situação:

- Foi criado um programa em Java onde o objeto "wBook" representa um objeto do tipo "WritableWorkbook", que, por sua vez, representa um arquivo XLS (um arquivo do Excel) gerado pelo programa.

Qual é a alternativa correta que mostra a linha de comando que salva este arquivo no HD?

Resposta Selecionada: ☒ d. wBook.write().

Respostas:

- ☐ a. wBook.save().
- ☐ b. wBook.saveFile().
- ☐ c. WritableWorkbook.save(wBook).
- ☒ d. wBook.write().
- ☐ e. File.save(wBook).

Comentário da resposta: Resposta: D
Comentário: o comando da linguagem Java que salva, efetivamente, um arquivo de planilha do Excel®, após este ter sido

preenchido com todas as informações necessárias, é o método "write()" da classe "WritableWorkbook", que é a classe que representa uma planilha editável do Excel®. Desta forma, é possível acionar este método por um objeto gerado a partir daquela classe, já que não é um método estático.

← OK