

Exercícios XML

Programação em Ambiente Visual

Emerson C. Lima

Questão 1. Nesse exercício você construirá uma aplicação XML simples que irá reportar o elemento raiz de um documento de um arquivo XML.

i. Use o seguinte código

```
1 import java.io.FileInputStream;
2 import java.io.InputStream;
3
4 import javax.xml.parsers.DocumentBuilder;
5 import javax.xml.parsers.DocumentBuilderFactory;
6
7 import org.w3c.dom.Document;
8
9 public class RaizXML {
10
11     public static void main(String[] args) {
12         String nomeArquivo = args[0];
13         System.out.print("O elemento raiz do documento é... ");
14
15         try {
16             DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
17             DocumentBuilder builder = factory.newDocumentBuilder();
18             InputStream conteudoXML = new FileInputStream(nomeArquivo);
19             Document doc = builder.parse(conteudoXML);
20             System.out.println(doc.getDocumentElement().getNodeName() + ".");
21         } catch (Exception e) {
22             System.out.println("Oops! " + e.getMessage());
23         }
24     }
25 }
```

Responda os itens a seguir executando o programa acima usando a linha de comando

java RaizXML arquivo.xml

ii. Qual a saída do programa ao ser executado passando o arquivo vazio.xml como argumento?

```
1 <?xml version="1.0"?>
```

iii. Qual a saída do programa ao ser executado passando um arquivo que não é XML, como /etc/fstab, como argumento?

Questão 2. Construa um programa que percorre o DOM. Para isso escreva um programa que percorra a hierarquia de nós do documento DOM imprimindo os na tela.

Questão 3. Construa um programa que remova todos os nós que forem textos em brancos. Ou seja, nós do tipo text com valor vazio ("") ou nulo (null).

Questão 4. Construa um programa que cria um documento DOM do zero.

i. Construa uma classe ou método para criar um objeto Document vazio.

```
1 DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
2 DocumentBuilder builder = docBuilderFactory.newDocumentBuilder();
3 doc = builder.newDocument();
```

ii. Construa outra classe ou método para preencher o objeto Document com os elementos desejados, os atributos desejados e os nós de textos desejados. Use os métodos createElement, appendChild, createTextNode, e os métodos de navegação da API DOM.

```

1 private Node criarElementoDoUsuario(Document documento,
2     String nome,
3     String valor) {
4
5     Element node = doc.createElement(nome);
6     node.appendChild(doc.createTextNode(valor));
7
8     return node;
9 }
10
11 private Node criarUsuario(Document documento,
12     String id,
13     String nome,
14     String email,
15     String ocupacao) {
16     Element usuario = documento.createElement("usuario");
17
18     usuario.setAttribute("id", id);
19     usuario.appendChild(criarElementoDoUsuario(documento, "nome", nome));
20     usuario.appendChild(criarElementoDoUsuario(documento, "email", email));
21     usuario.appendChild(criarElementoDoUsuario(documento, "ocupacao", ocupacao));
22
23     return usuario;
24 }

```

- iii. Em seguida use as classes do pacote `javax.xml.transform` para converter a variável `doc` para a saída padrão e para um arquivo no disco.

```

1 TransformerFactory transformerFactory = TransformerFactory.newInstance();
2 Transformer transf = transformerFactory.newTransformer();
3
4 transf.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
5 transf.setOutputProperty(OutputKeys.INDENT, "yes");
6
7 DOMSource source = new DOMSource(doc);
8
9 StreamResult console = new StreamResult(System.out);
10 transf.transform(source, console);
11
12 File arquivo = new File("usuarios.xml");
13 StreamResult fluxo = new StreamResult(fluxo);
14 transf.transform(source, fluxo);

```

Questão 5. Construa um programa para imprimir o sobrenome dos terceirizados acessando os elementos e atributos do documento DOM pelo nome.

- i. Para isso considere o seguinte arquivo XML:

```

1 <?xml version="1.0"?>
2 <colaboradores>
3   <terceirizado>
4     <info sobrenome="albertson" nome="albert" cpf="123456789"/>
5     <funcao>Programador C++</funcao>
6     <data-contratacao>1/1/2018</data-contratacao>
7   </terceirizado>
8   <empregado>
9     <info sobrenome="bartholemew" nome="bart" cpf="223456789"/>
10    <funcao>Diretor de Tecnologia</funcao>
11    <data-contratacao>1/1/2010</data-contratacao>
12    <data-demissao>1/11/2010</data-demissao>
13  </empregado>
14  <parceiro>
15    <info sobrenome="carlson" nome="carl" cpf="323456789"/>
16    <funcao>Advogado do Trabalho</funcao>
17    <data-contratacao>10/1/2012</data-contratacao>
18  </parceiro>
19  <terceirizado>
20    <info sobrenome="denby" nome="dennis" cpf="423456789"/>
21    <funcao>Programador Java</funcao>
22    <data-contratacao>1/1/2014</data-contratacao>
23  </terceirizado>
24  <empregado>
25    <info sobrenome="edwards" nome="eddie" cpf="523456789"/>
26    <funcao>Gerente de Projeto</funcao>
27    <data-contratacao>4/4/2013</data-contratacao>
28  </empregado>
29  <parceiro>
30    <info sobrenome="fredericks" nome="fred" cpf="623456789"/>
31    <funcao>Advogado de Propriedade Intelectual</funcao>
32    <data-contratacao>10/1/2009</data-contratacao>
33  </parceiro>
34 </colaboradores>

```

- ii. Recupere o sobrenome dos terceirizados usando o método `getElementsByTagName` para recuperar uma lista de nós (`NodeList`).

```

1 NodeList nodeList = doc.getElementsByTagName("colaboradores");
2 Element elm = (Element) nodeList.item(0);

```

- iii. Busque então os elementos com a tag `terceirizado` a partir da variável `elm`.

```

1 NodeList terceirizados = elm.getElementsByTagName("terceirizado");

```

- iv. Itere a variável `terceirizados` e recupere o elemento `info` com o método `getElementsByTagName`.

```

1 for (int i = 0; i < terceirizados.getLength(); i++) {
2   Element terceirizado = (Element) terceirizados.item(i);
3
4   //Não precisamos iterar os elementos
5   //sabemos que há somente um
6   Element info = (Element)terceirizado.getElementsByTagName("info").item(0);
7   System.out.println("Sobrenome do terceirizado" + info.getAttribute("sobrenome"));
8 }

```