**为XML文档编写DOM API的Java程序**

1.考虑下面的XML文件，文件名为5.4.xml。

<?xml version = "1.0"?>

<!DOCTYPE Root[<!ENTITY First "<First>DOM</First>">]>

<Root>

&First;

<Second>SAX</Second>

</Root>

建立一个使用DOM API的Java程序。该程序可以解析上面的XML文档，并且得到如下所示的结果。

<?xml version = "1.0"?>

<Root>

<First>DOM</First>

<Second>SAX</Second>

</Root>

结果：

**import** javax.xml.parsers.DocumentBuilder;

**import** javax.xml.parsers.DocumentBuilderFactory;

**import** org.w3c.dom.Document;

**import** org.w3c.dom.Element;

**import** org.w3c.dom.Node;

**import** org.w3c.dom.NodeList;

**public** **class** ParseSimpleXml {

Element root = **null**;

**public** ParseSimpleXml() {

**try** {

DocumentBuilderFactory factory = DocumentBuilderFactory.*newInstance*();

DocumentBuilder builder = factory.newDocumentBuilder();

Document doc = builder.parse("5.4.xml");

doc.normalize();

root = doc.getDocumentElement();

} **catch** (Exception e) {

System.***out***.println(e);

}

}

**public** **void** parse() {

System.***out***.println("<?xml version = \"1.0\"?>");

System.***out***.println("<" + root.getNodeName() + ">");

NodeList children = root.getChildNodes();

**for** (**int** i = 0; i < children.getLength(); i++) {

Node child = children.item(i);

**if** (child.getNodeType() == Node.***ELEMENT\_NODE***) {

System.***out***.print("<" + child.getNodeName() + ">");

System.***out***.print(child.getFirstChild().getNodeValue());

System.***out***.println("</" + child.getNodeName() + ">");

}

}

System.***out***.println("</" + root.getNodeName() + ">");

}

**public** **static** **void** main(String[] args) {

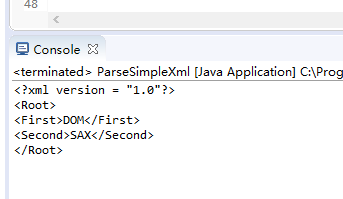
// **TODO** Auto-generated method stub

ParseSimpleXml parseSimpleXml = **new** ParseSimpleXml();

parseSimpleXml.parse();

}

}



2.用DOM接口生成如下内容并且把它们写到一个新的XML文档中。

<?xml version="1.0" encoding="UTF-8"?>

<Team>

<TeamMember>

<Empno value="30772" />

<Name value="Manjeet Singh" />

<Designation value="Team Leader." />

</TeamMember>

</Team>

结果：

**import** javax.xml.parsers.DocumentBuilder;

**import** javax.xml.parsers.DocumentBuilderFactory;

**import** javax.xml.transform.Transformer;

**import** javax.xml.transform.TransformerFactory;

**import** javax.xml.transform.dom.DOMSource;

**import** javax.xml.transform.stream.StreamResult;

**import** org.w3c.dom.Attr;

**import** org.w3c.dom.Document;

**import** org.w3c.dom.Element;

**public** **class** CreateSimpleXml {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**try** {

DocumentBuilderFactory factory = DocumentBuilderFactory.*newInstance*();

DocumentBuilder builder = factory.newDocumentBuilder();

Document doc = builder.newDocument();

Element team = doc.createElement("Team");

Element teamMember = doc.createElement("TeamMember");

Element empno = doc.createElement("Empno");

Attr empnoValue = doc.createAttribute("value");

empno.setAttribute(empnoValue.getNodeName(), "30772");

Element name = doc.createElement("Name");

Attr nameValue = doc.createAttribute("value");

name.setAttribute(nameValue.getNodeName(), "Manjeet Singh");

Element designation = doc.createElement("Designation");

Attr designationValue = doc.createAttribute("value");

designation.setAttribute(designationValue.getNodeName(), "Team Leader.");

team.appendChild(teamMember);

teamMember.appendChild(empno);

teamMember.appendChild(name);

teamMember.appendChild(designation);

doc.appendChild(team);

TransformerFactory tFactory = TransformerFactory.*newInstance*();

Transformer transformer = tFactory.newTransformer();

DOMSource source = **new** DOMSource(doc);

StreamResult result = **new** StreamResult(**new** java.io.File("4.5.xml"));

transformer.transform(source, result);

} **catch** (Exception e) {

e.printStackTrace();

}

}

}