# 实验七 使用DOM接口技术解析XML文档

**一、实验目的**

掌握DOM接口技术解析XML文档，包括按条件获取节点数据、遍历XML文档、添加节点及转换输出到XML文档。

**二、实验内容**

使用Eclipse工具，按照实验指导要求，完成利用DOM接口技术对XML文档的解析操作。

**三、实验指导**

1. 请编写一个利用DOM接口技术遍历文档stu.xml的应用程序，要求可以在控制台上输出所有元素及属性节点的内容。

|  |
| --- |
| <?xml version="1.0" encoding="gb2312" ?>  <musiclession>  <student ID="s101" sex="女">  <name>李华</name>  <age>12</age>  <score>92</score>  </student>  <student ID="s102" sex="男">  <name>笑林</name>  <age>22</age>  <score>82</score>  </student>  <student ID="s103" sex="男">  <name>王明</name>  <age>18</age>  <score>90</score>  </student>  </musiclession> |

具体操作：

1. 打开Eclipse应用程序，新建一个项目（项目名自拟）。
2. 在新建项目中创建源文件夹“src”，在src中新建一个包，包名为dom。
3. 在包dom中，创建一个类，类名定义为StuDomParse.java。
4. 该题遍历的主要是元素和属性。为了达到遍历文档的目的，在类StuDomParse中分别定义两个方法printNode(Node node)和printAttr(Element element)。

其中属性一定是附加在元素中，所以printAttr(Element element)方法参数为Element元素节点，该方法主要作用是输出指定元素节点的所有属性内容，代码如下。

|  |
| --- |
| **public** **static** **void** printAttr(Element element){  //获取给定元素element的所有属性节点集合  NamedNodeMap attrMap = element.getAttributes();  //处理每一个属性节点，按照“属性名=属性值”格式输出  **for** (**int** i = 0; i < attrMap.getLength(); i++) {  Attr attr = (Attr)attrMap.item(i);  System.*out*.print(" " + attr.getName() + " = " + attr.getValue());  }  } |

printNode(Node node)方法作用是输出指定节点node的内容，具体内容根据node节点的子节点分类进行处理，代码如下：

|  |
| --- |
| **public** **static** **void** printNode(Node node) {  //获取node节点的所有子节点集合  NodeList nodeList = node.getChildNodes();  //处理每一个子节点  **for** (**int** i = 0; i < nodeList.getLength(); i++) {  Node childNode = nodeList.item(i);  /\*子节点如果是元素节点，输出元素名，并输出元素属性，调用printAttr方法;元素可以含有子节点，因此调用printNode方法继续输出当前元素节点的子节点内容\*/  **if** (childNode.getNodeType() == Node.*ELEMENT\_NODE*) {  Element childEle = (Element)childNode;  System.*out*.print("<" + childEle.getTagName());  *printAttr*(childEle);  System.*out*.print(">");  *printNode*(childEle); //递归调用  }  //子节点如果是文本节点，直接输出文本内容  **if** (childNode.getNodeType() == Node.*TEXT\_NODE*) {  Text text = (Text)childNode;  System.*out*.println(text.getTextContent());  }    }  } |

1. 在类StuDomParse的主方法中定义遍历stu.xml文档的具体代码。获取Document对象代码如下：

|  |
| --- |
| //实例化DocumentBuilderFactory工厂类（javax.xml.parsers包）  DocumentBuilderFactory fac= DocumentBuilderFactory.*newInstance*();  //获取DocumentBuilder对象（javax.xml.parsers包）,称做DOM解析器  DocumentBuilder builder = fac.newDocumentBuilder();  //创建stu.xml文件类对象  File file = **new** File("stu.xml");  //DocumentBuilder对象解析指定的XML文件，返回实现了Document接口的实例（org.w3c.dom包）  Document doc = builder.parse(file); |

在获取了Document文档对象后，可以获取文档根元素并进行相应处理，代码如下。

|  |
| --- |
| Element root = doc.getDocumentElement();//获取文档根元素  String rootName = root.getTagName();//获取根元素名称  System.out.println("<" + rootName + ">");//输出根元素名 |

根元素节点root是Node节点，因此可以调用已定义好的printNode方法，从根元素开始遍历输出元素及属性内容，代码如下：

|  |
| --- |
| *printNode*(root); |

1. 运行定义好的StuDomParse.java文件，可以看到控制台上显示结果如下所示：

|  |
| --- |
| <musiclession>  <student ID = s101 sex = 女>  <name>李华  <age>12  <score>92  <student ID = s102 sex = 男>  <name>笑林  <age>22  <score>82  <student ID = s103 sex = 男>  <name>王明  <age>18  <score>90 |

2. 请编写一个应用程序，利用DOM接口技术访问第1题中的文档stu.xml，要求在控制台上输出所有<student>元素的ID属性的属性值以及<name>元素的文本内容。

3. 为stu.xml对应的DOM树添加一个<student>元素（结构按照文档要求，文本内容自拟），并将转换后的DOM树转换保存为“stu.xml”文件（覆盖源文件）。

**四、作业**

解析“stu.xml”文件，搜索所有<student>元素的ID属性、sex属性以及<name>元素文本内容，并将搜索的数据转换成下面所示格式的XML文档，保存为新文件“newStu.xml”。

newStu.xml内容如下：

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <学生>  <姓名 学号="s101" 性别="女">李华</姓名>  <姓名 学号="s102" 性别="男">笑林</姓名>  <姓名 学号="s103" 性别="男">王明</姓名>  </学生> |

作业!!!!!!!!：Sax解析，用DOM生成文档