**思考题二：**

**package** 测试;

**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("C:/Users/Administrator/Desktop/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() + ">");

}

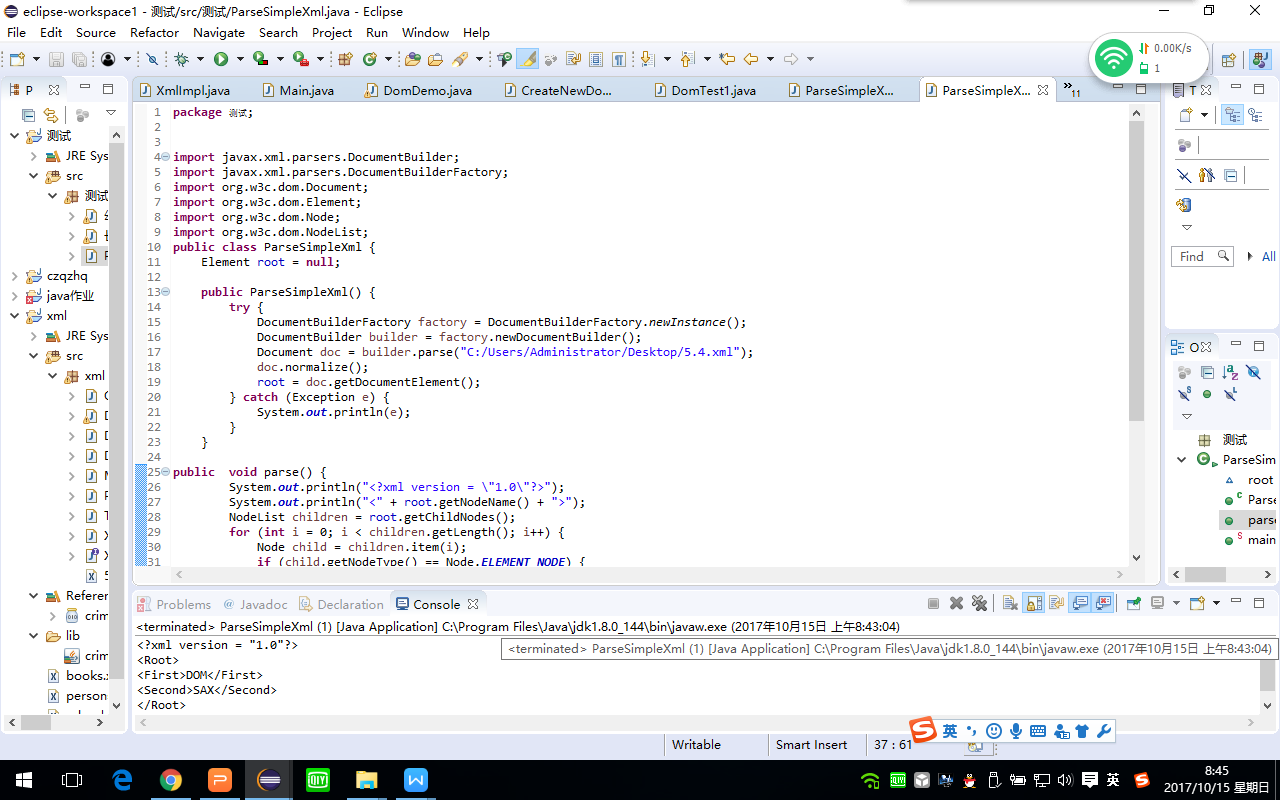
**public** **static** **void** main(String[] args) {

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

parseSimpleXml.parse();

}

}



**思考题三：**

**package** xml;

**import** java.io.FileOutputStream;

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

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

**import** org.apache.crimson.tree.XmlDocument;

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

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

**public** **class** CreateNewDom {

/\*\*

\* 创建xml文档

\* \*/

**public** **static** **void** createDom(){

Document doc;

Element Team,TeamMember;

Element Empno = **null**;

Element Name = **null**;

Element Designation = **null**;

// Element address = null;

// Element tel = null;

**try**{

//得到DOM解析器的工厂实例

DocumentBuilderFactory dbFactory = DocumentBuilderFactory.*newInstance*();

//从DOM工厂中获得DOM解析器

DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();

//创建文档树模型对象

doc = dbBuilder.newDocument();

**if**(doc != **null**){

//创建Team元素

Team = doc.createElement("Team");

//创建TeamMember元素

TeamMember = doc.createElement("TeamMember");

Empno = doc.createElement("Empno");

Empno.setAttribute("value","30772");

Name = doc.createElement("Name");

Name.setAttribute("value","Manjeet Singh");

Designation = doc.createElement("Designation");

Designation.setAttribute("value","Team Leader.");

//将TeamMember作为子元素添加到树的根节点Team

Team.appendChild(TeamMember);

TeamMember.appendChild(Empno);

TeamMember.appendChild(Name);

TeamMember.appendChild(Designation);

//添加到文档树中

doc.appendChild(Team);

//将内存中的文档通过文件流生成C:/Users/Administrator/Desktop/5.6.xml下

((XmlDocument)doc).write(**new** FileOutputStream("C:/Users/Administrator/Desktop/5.6.xml"));

System.***out***.println("创建成功");

}

}**catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

}

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

CreateNewDom.*createDom*();

}

}