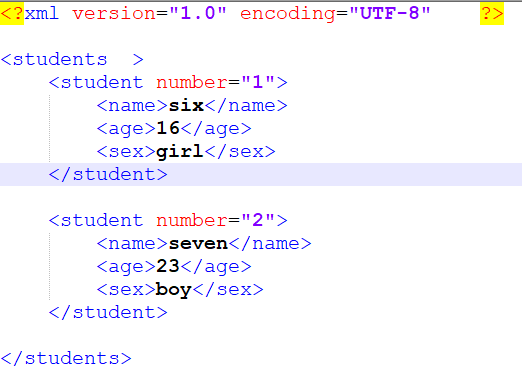
### Java第一阶段—DAY19-JAVA案例

1. 有xml文档，内容如下，请使用dom4j解析



**public class** Demo02Xml {  
   
 **public static void** main(String[] args) **throws** DocumentException {  
 Document doc = getDocumentBySax() ;   
 Element root = doc.getRootElement();  
*// System.out.println(root.getName());* Iterator<Element> iterator = root.elementIterator();  
  
 **while**(iterator.hasNext()){  
 Element next = iterator.next();  
 System.out.println(next.getName()+**" "**+next.attributeValue(**"number"**));  
 Element nameEle = next.element(**"name"**);  
 System.out.println(nameEle.getName());  
 System.out.println(nameEle.getText());  
 }  
   
 }  
   
 **public static** Document getDocumentBySax() **throws** DocumentException{  
 InputStream stream = Demo02.**class**.getClassLoader().getResourceAsStream(**"com/blb/demo04/class.xml"**);  
 SAXReader reader = **new** SAXReader();  
 **return** reader.read(stream);  
 }  
   
 **public static** Document getDocumentByDom() **throws** Exception{  
 DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();  
 DocumentBuilder db = dbf.newDocumentBuilder();  
 InputStream in = Demo02.**class**.getClassLoader().getResourceAsStream(**"com/blb/demo04/class.xml"**);  
 org.w3c.dom.Document w3cdoc=db.parse(in);  
 DOMReader domReader=**new** DOMReader();  
 Document document=domReader.read(w3cdoc);  
 **return** document;  
 }  
   
}

1. 有XML内容如下，请分别使用DOM以及SAX解析。

*<?*xml version="1.0" encoding="UTF-8" *?>*<students >  
 <student number="1">  
 <name>six</name>  
 <age>16</age>  
 <sex>girl</sex>  
 </student>  
   
 <student number="2">  
 <name>seven</name>  
 <age>23</age>  
 <sex>boy</sex>  
 </student>  
   
</students>

**public class** Demo03Xml {  
   
 **public static void** main(String[] args) **throws** Exception {  
 Document doc = getDocumentBySax() ;   
   
*// Node node = doc.selectSingleNode("//students/student[@number='2']");  
// System.out.println(node.getName());* List<Node> list = doc.selectNodes(**"//students/student"**);  
 **for** (Node node : list) {  
   
 **if**(node.matches(**"@number='2'"**)){  
 System.out.println(node.getName()+**"\t"**+node.valueOf(**"@number"**));  
   
 Node nameNode = node.selectSingleNode(**"name"**);  
 System.out.println(nameNode.getText());  
 }  
   
 }  
   
 }  
   
 **public static** Document getDocumentBySax() **throws** DocumentException{  
 InputStream stream = Demo03.**class**.getClassLoader().getResourceAsStream(**"com/blb/demo04/class.xml"**);  
 SAXReader reader = **new** SAXReader();  
 **return** reader.read(stream);  
 }  
   
 **public static** Document getDocumentByDom() **throws** Exception{  
 DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();  
 DocumentBuilder db = dbf.newDocumentBuilder();  
 InputStream in = Demo03.**class**.getClassLoader().getResourceAsStream(**"com/blb/demo04/class.xml"**);  
 org.w3c.dom.Document w3cdoc=db.parse(in);  
 DOMReader domReader=**new** DOMReader();  
Document document=domReader.read(w3cdoc);  
 **return** document;  
 }  
  
}

1. 使用程序往XML中写入如下内容

*<?*xml version="1.0" encoding="UTF-8"*?>*<students>  
 <student number="1">  
 <name>six</name>  
 <age>16</age>  
 <sex>girl</sex>  
 </student>  
 <student number="2">  
 <name>seven</name>  
 <age>23</age>  
 <sex>boy</sex>  
 </student>  
</students>

**public class** Demo04Xml {  
  
 **public static void** main(String[] args) **throws** Exception {  
*// 1,�����ڴ��е�Document����* Document document = DocumentHelper.createDocument();  
*// 2����document���������* Element root = document.addElement(**"students"**);  
 Element s1 = root.addElement(**"student"**).addAttribute(**"number"**, **"1"**);  
 s1.addElement(**"name"**).addText(**"six"**);  
 s1.addElement(**"age"**).addText(**"16"**);  
 s1.addElement(**"sex"**).addText(**"girl"**);  
   
 Element s2 = root.addElement(**"student"**).addAttribute(**"number"**, **"2"**);  
 s2.addElement(**"name"**).addText(**"seven"**);  
 s2.addElement(**"age"**).addText(**"23"**);  
 s2.addElement(**"sex"**).addText(**"boy"**);  
OutputFormat of = OutputFormat.createPrettyPrint();  
 FileWriter fw = **new** FileWriter(**"src/com/blb/seven06/student.xml"**);  
 XMLWriter xw = **new** XMLWriter(fw,of);  
 xw.write(document);  
 xw.flush();  
 }  
  
}

1. 练习正则API

public class Demo07Regular {  
  
 public static void main(String[] args) {  
 System.out.println(Pattern.matches("a\*b", "aaaaabb"));  
 System.out.println(Pattern.matches("[ab]", "ab"));  
 System.out.println(Pattern.matches("[a-zA-H]", "Z"));  
 System.out.println(Pattern.matches("[^abc]", "d"));  
 System.out.println(Pattern.matches("[^abc]", "d"));  
 System.out.println(Pattern.matches("[a-k&&c-z]", "z"));  
 System.out.println(Pattern.matches("a[a-z]c", "abc"));  
 System.out.println(Pattern.matches("\\d", "10"));  
 System.out.println(Pattern.matches("\\D", "a"));  
 System.out.println(Pattern.matches("\\s", "\t"));  
 System.out.println(Pattern.matches("\\S", "a"));  
 System.out.println(Pattern.matches("\\w", "a"));*//[a-zA-Z\_0-9]* System.out.println(Pattern.matches("\\W", "a"));  
 System.out.println(Pattern.matches("^a\\d{4}f$", "a1234f"));  
 System.out.println(Pattern.matches("ab?c", "abbc"));  
 System.out.println(Pattern.matches("ab\*c", "abbbbbbbc"));  
 System.out.println(Pattern.matches("a\\d+c", "a12312312312c"));  
 System.out.println(Pattern.matches("a\\d{3}c", "a123c"));  
 System.out.println(Pattern.matches("a\\d{3,}c", "a12312312312c"));  
 System.out.println(Pattern.matches("a\\d{3,5}c", "a123456c"));  
 }  
  
}

1. 定义日期字符串"2021-01-06"，使用正则分组技术分别获取年份、月份、日期信息。

public class Demo08Regular {  
  
 public static void main(String[] args) {  
 String s = "2021-01-06";  
 Pattern p = Pattern.compile("(\\d{4})-(\\d{2})-(\\d{2})");  
 Matcher matcher = p.matcher(s);  
*// 使用find方法来匹配* if(matcher.find()){  
*// matcher.group() 返回的是整个串的数据* System.out.println(matcher.group());*//2021-01-06  
 // 分组：通过索引来获取第几组，索引为从左往右数第几个 (* System.out.println(matcher.group(0));*//2021-01-06* System.out.println(matcher.group(1));*//2021* System.out.println(matcher.group(2));*//01* System.out.println(matcher.group(3));*//06* }  
 }  
}

**public class** Demo09Regular {  
  
 **public static void** main(String[] args) {  
 String s = **"2021-01-06"**;  
 Pattern p = Pattern.compile(**"(?<year>\\d{4})-(?<month>\\d{2})-(?<day>\\d{2})"**);  
 Matcher matcher = p.matcher(s);  
*// 使用find方法来匹配* **if**(matcher.find()){  
*// matcher.group() 返回的是整个串的数据* System.out.println(matcher.group());*//2021-01-06  
 // 分组：通过分组别名获取对应字段数据值* System.out.println(matcher.group(**"year"**));*//2021* System.out.println(matcher.group(**"month"**));*//01* System.out.println(matcher.group(**"day"**));*//06* }  
 }  
}