1 XML简介

XML 的全称为可扩展标记语言。

XML 的作用：

1. 保存数据，并且这些数据具有自我描述性。
2. 作为项目或模块的配置文件。
3. 作为网络数据传输的格式。（现在被 JSON 代替）

2 XML语法

2.1 文档声明

```
<?xml version="1.0" encoding="utf-8"?>
```

以上内容是 XML 文件的声明。`version="1.0"`表示 XML 的版本，`encoding="utf-8"`表示 XML 文件的编码。

2.2 XML注释

XML 的注释语法和 HTML 相同：`<!--注释内容-->`。

2.3 XML元素

XML 元素指的是从开始标签直到结束标签的部分。元素可包含其他元素、文本或两者的混合物。元素也可以拥有属性。

XML 标签可以使用任何名称，没有保留字。

XML 标签的命名规则：

1. 可以包含数字、字母和其他字符。
2. 不能以数组或标点符号开始。
3. 不能包含空格。

XML 中的标签也分为单标签和双标签，语法和 HTML 相同。

2.4 XML属性

一个标签上可以书写多个属性，每个属性的值必须用引号引起来。属性和标签名之间、属性和属性之间用空格隔开。

2.5 语法规则

1. 所有 XML 元素必须有结束标签。
2. XML 标签对大小写敏感。
3. XML 文档必须有唯一的根元素。
4. XML 的转义字符，如：`<`表示`<`，`>`表示`>`，等等。
5. 文本区域（CDATA 区）：`<![CDATA[内容]]>`。将 CDATA 区中的内容作为纯文本，不进行 XML 语法解析。

3 XML解析技术

XML 与 HTML 类似，也是标记型文档，也可以使用 DOM 技术来解析。

XML 文档对象模型（DOM）定义访问和操作 XML 文档的标准方法。DOM 将 XML 文档组织成一个树形结构，根元素为根节点。`document`对象表示整个 XML 文档。

早期 JDK 提供了两种 XML 解析技术：DOM 和 SAX（Simple API for XML）。它们都已经过时。

第三方解析技术：jdom、dom4j、pull。

4 dom4j解析技术

要使用 dom4j，首先要下载相应的 jar 包，并把它引入到项目中。

dom4j 编程步骤：

1. 加载 XML 文件，创建`Document`对象。
2. 通过`Document`对象得到根元素对象。
3. 通过`根元素对象.elements(标签名)`获得指定标签名的子元素对象集合。
4. 找到要修改、删除的子元素，进行相应的操作。
5. 保存文件。

例如：

```xml
<?xml version="1.0" encoding="utf-8"?>
<books>
    <book sn="SN1">
        <name>辟邪剑谱</name>
        <price>9.9</price>
        <author>班主任</author>
    </book>
    <book sn="SN2">
        <name>葵花宝典</name>
        <price>99.9</price>
        <author>班长</author>
    </book>
</books>
```

```java
import org.dom4j.io.SAXReader;
import org.dom4j.Document;
import org.dom4j.DocumentException;
import org.dom4j.Element;

public class Dom4jTest
{
    public void test() throws DocumentException
    {
        // 创建一个 SAXReader 对象
        SAXReader reader = new SAXReader();
        // 读取 XML 文件，创建 Document 对象
        Document document = reader.read("test.xml");
        // 通过 Document 对象获取根元素对象
        Element rootElement = document.getRootElement();
        // 通过根元素对象获取子元素对象
        // elements() 函数返回指定标签名的元素对象集合
        List<Element> books = rootElement.elements("book");
        // 遍历集合，处理每个元素对象
        for (Element book: books)
        {
            // asXML() 函数将元素对象转换成字符串
            System.out.println(book.asXML());
            // element() 函数返回指定标签名的子元素对象
            Element nameElement = book.element("name");
            // getText() 函数返回标签中的文本内容
            String nameText = nameElement.getText();
            // elementText() 函数返回指定标签名的文本内容
            String priceText = book.elementText("price");
            String authorText = book.elementText("author");
            // attributeValue() 函数返回指定属性的值
            String snValue = book.attributeValue("sn");
            // 将获取到的数据封装为对象
            // Book 类是自定义类，其中的属性与 XML 文档中的标签保持一致
            System.out.println(new Book(snValue, nameText,
                                        Double.parseDouble(priceText),
                                        authorText));
        }
    }
}
```