01_XML 概述（理解）

XML 是一种文件格式，只需要知道格式，知道怎么写就行了

XML(eXtensible Markup Language)，是一种 **可扩展的标记语言**. (使用<>括起来) XML 技术是 W3C 组织(World Wide Web Consortium万维网联盟)发布的，目前遵循的是 W3C 组织于1998 年发布的 XML1.0规范. 它的设计宗旨是传输数据，而不是显示数据(HTML). 它的标签没有被预定义,需要自行定义标签. 它是 W3C 的推荐标准.

02_为什么要学XML（理解）

- XML是一种通用的数据交换格式
- 许多系统的配置文件都使用XML格式
- 在日常应用中会经常看见XML格式的文件
- 掌握XML是软件开发人员的一项基本技能
- JavaEE 框架基本都有在使用 xml

XML 结构清晰(树状结构)，不仅让人能够明白，还让计算机也能够明白。 XML 作为一种公订的、开放的标准，不受知识产权的限制.

XML 一般作为配置文件的存在, 用来传输数据.

03_XML 语法（理解）

XML文件是以 .xml 结尾的文件，该文件是否书写正确,可以使用浏览器打开.

constacts.xml

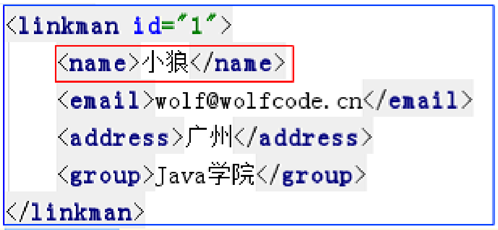
```xml
<?xml version="1.0" encoding="utf-8"?>
<!-- 以上文档声明，versino:版本，
encoding 编码格式：文档内容编码和文件本身编码必须保持一致-->
<contacts>
    <linkman id="1">
        <name>小狼</name>
        <email>wolf@wolfcode.cn</email>
        <address>广州</address>
        <group>Java学院</group>
    </linkman>
    <linkman id="2">
        <name>小码</name>
        <email>horse@520it.com</email>
        <address>广州</address>
        <group>Java学院</group>
    </linkman>
</contacts>
```

1. xml 文档需在文档第一行做声明，声明表示该文档为 xml 文档
    1. `<?xml version="1.0" encoding="UTF-8" standalone="yes"?>`
    2. version: xml 的版本 （了解）
    3. standalone：yes 为不可包含其他文档，no或不写表示可包含也可不包含（了解）
    4. ecoding：编码格式，文档内容的编码和文件的编码需要统一（重要）
2. <> 表示 XML 文档中的元素/标签（Element）
3. 标签有内容需成对出现，开始标签 `<ooxx>`,结束标签 `</ooxx>`
4. 标签如果没内容可使用但标签 `<ooxx/>`
5. 标签中可以带有属性，格式： 属性名=“属性值” ，id="1"
6. 在 XML 文档中，有且仅有一个根标签
7. 在 XML 中，严格区分大小写 `<name>` -- `<Name>`

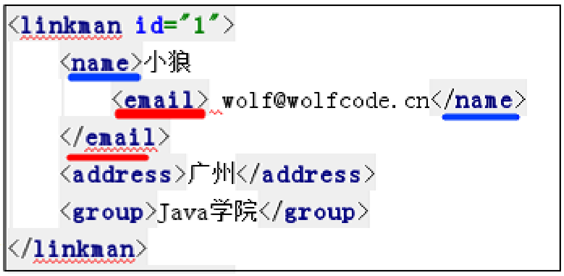
8. 在 XML 中，允许标签嵌套，但是不允许交叉嵌套

9. 在 XML 中，注释使用

正常嵌套

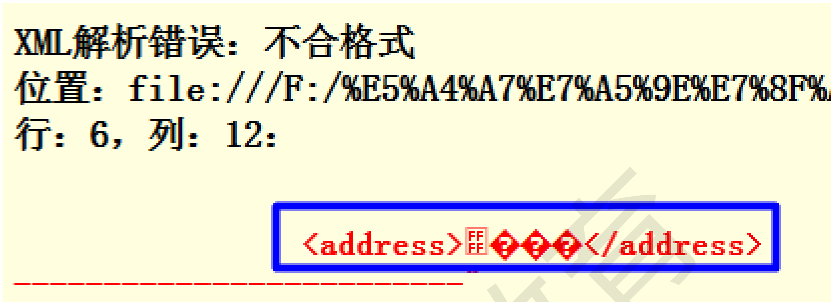


交叉嵌套

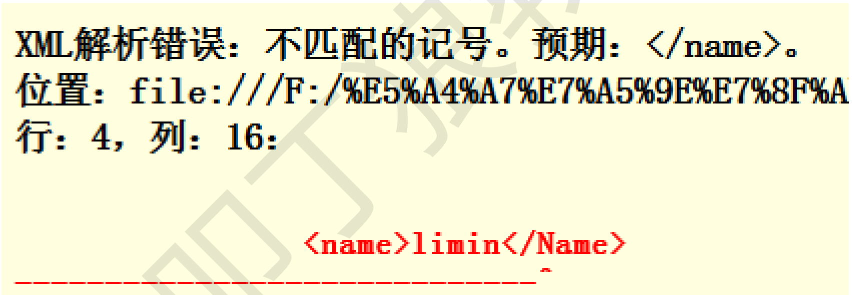


常见错误：

1.文档内容和文件本身编码格式一定要为 UTF-8

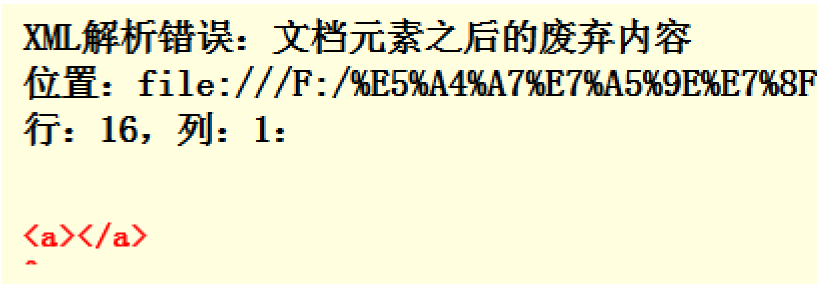


2.严格区分大小写



3.标签允许嵌套,但是不允许交叉嵌套

4.有且仅有一个标签直接属于XML文档.即根标签



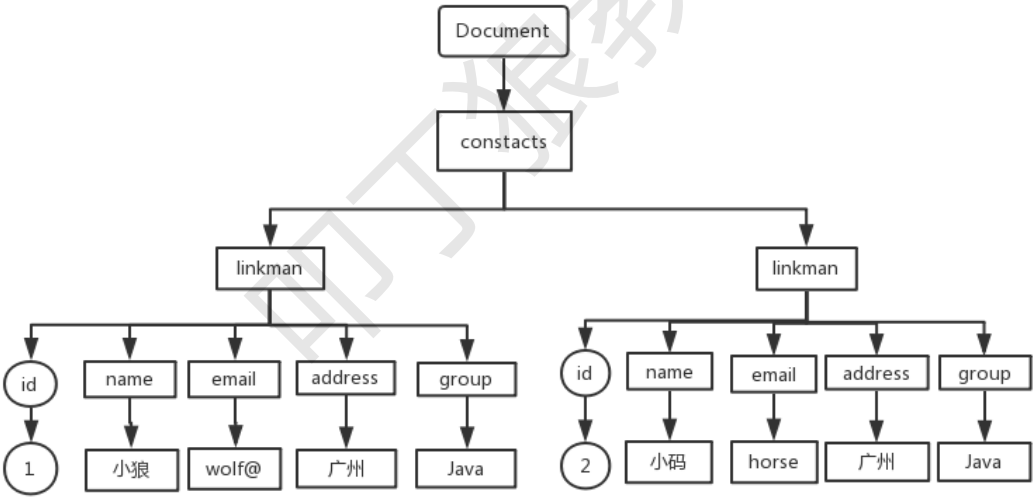
04_XML 结构和 DOM（理解）

4.1_XML 结构分析（理解）

普通 xml 结构图

```
<?xml version="1.0" encoding="utf-8"?>
<contacts>
    <linkman id="1">
        <name>小狼</name>
        <email>wolf@wolfcode.cn</email>
        <address>广州</address>
        <group>Java学院</group>
    </linkman>
    <linkman id="2">
        <name>小码</name>
        <email>horse@520it.com</email>
        <address>广州</address>
        <group>Java学院</group>
    </linkman>
</contacts>
```

通过 xml 层次结构整理如下图：

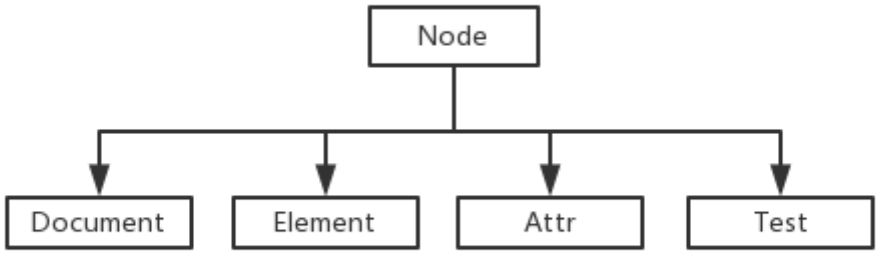


XML 是用来做数据传输的,最终是需要读取到程序保存到内存中的.而从面相对象的角度来思考,

XML 的各个组成部分都需要使用一个类型来描述.

1. **XML 文件**：把 XML 文档加载到内存中,使用 Document 对象来描述整个文档.
2. **标签/元素**：所有的标签,使用 Element 对象来描述.
3. **属性**：标签的属性,使用 Attribute 来描述.
4. **文本**：文本内容（文本/空格·车）使用 Text 来描述.

根据上面四种成员的共性,继续抽象出父类:org.w3c.dom.Node.所以说，在XML中,一切皆节点.



而这种把 XML 文档加载到内存之后，形成一个一个的对象，这种操作我们称为 **DOM 解析。**

4.2_DOM 简介（了解）

DOM: Document Object Model: 文档对象模型,把文档中的成员描述成一个个对象.

后期使用场景： 使用 JavaScript (js) 来解析 HTML 中的数据.

特点: 在加载的时候,一次性把整个XML文档加载进内存,在内存中形成一颗树(**Document对象**)/**DOM树**. 我们以后使用代码操作Document,其实操作的是内存中的 DOM树,和本地磁盘中的XML文件没有直接关系. 比如: 保存了一个联系人, 仅仅是内存中多了一个联系人,但是在XML文件中没有新增的痕迹. 除非做同步操作(把内存中的数据更新到XML文件).-----> 增删改操作完之后,都需要做同步操作.

缺点: 若 XML 文件过大,可能造成内存溢出. 操作 XML 的增删改查(CRUD)的时候很简单,但是性能比较低下.

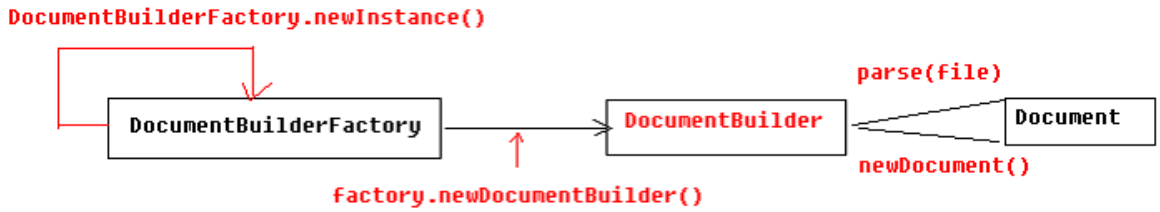
05_Document 获取（掌握）

XML 被程序读到内存中会形成一个Document对象，所有要解析 XML ，首先得先获取到 Document 对象

获取文档 Document 对象的步骤

1. 声明文件 xml 文件 File file = new File(path);
2. 通过 DocumentBuilderFactory 的 newInstance 方法获取本身的对象 DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
3. 通过 DocumentBuilderFactory 对象去获取 DocumentBuilder 对象 DocumentBuilder builder = factory.newDocumentBuilder();
4. 通过 DocumentBuilder 对象去解析获取 Document 对象 builder.parse(file);

步骤流程图以及获取 Document 对象方法选用



什么时候使用parse 什么时候去使用newDocument

newDocument() : 当没有xml文件的时候直接在内存中去创建DOM树

parse(file) : 当存在xml文件的时候使用该方法去解析xml文件到内存中形成DOM树(Document对象)

06_获取联系人信息(掌握)

需求: 获取第二个联系人的名字

步骤: 1.获取 Document 对象. 2.获取根节点/根标签. 3.获取第二个 linkman 节点. 4.获取名字节点. 5.获取名字节点的文本内容.

常用API: Document 对象: Element getDocumentElement(): 获取根节点.

Element 对象: NodeList getElementsByTagName(String name):通过标签名获取标签列表.

Node 对象: String getTextContent():获取节点的文本内容 void setTextContent(String content);设置节点的文本内容

代码实现

```
//得到某个具体的文本节点的内容:取出第二个联系人的名字
@Test
public void test1() throws Exception {
    // 1 获取文档对象Document
    DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
    DocumentBuilder documentBuilder = factory.newDocumentBuilder();
    Document doc = documentBuilder.parse(file);
    // 2 获取跟元素
    Element root = doc.getDocumentElement();
    // 3 获取第二个linkman元素
    Element linkmanEl = (Element)root.getElementsByTagName("linkman").item(1);
    // 4 获取第二个linkman元素下name元素
    Element nameEl = (Element)linkmanEl.getElementsByTagName("name").item(0);
    // 5 获取文本内容
    String content = nameEl.getTextContent();
    System.out.println(content);
}
```

07 插入一个联系人（了解）

需求: 增加一个新的联系人信息

步骤: 1 获取文档对象 2 获取根元素 3 创建一个linkman代码片段 3.1 创建name,email,address,group子元素 3.2 给name,email,address,group添加文本内容 3.3 将name,email,address,group元素作为linkman的子元素 3.4 将linkman作为根元素的子元素 4 同步操作

常用API: Document 对象: Element getDocumentElement(): 获取根节点.

Element 对象: NodeList getElementsByTagName(String name):通过标签名获取标签列表.

Node 对象: String getTextContent():获取节点的文本内容 void setTextContent(String content);设置节点的文本内容

注意 XML加载到内存之后,使用一个 Doucment 对象来描述 XML 的结构.之后操作,都是在操作内存中的 Java 对象而已,跟磁盘文件的XML文件没有关系.想要让XML文件也发生变化.需要进行同步(**内存中的Document->磁盘XML**)操作.这个操作之后,才能保证内存中的数据和磁盘的数据一致.

同步API：

Transformer : 同步转换器. public abstract void transform(Source xmlSource, Result outputTarget):同步操作

Source : 源是内存中的 Document，所以使用 DOMSource(doc) 实现类

Result : 内容写到磁盘，使用流操作文件，所有使用 StreamResult(file) 实现类

代码实现

```
Document doc = DocumentBuilderFactory.newInstance().newDocumentBuilder().pa
// 2 获取根元素
Element root = doc.getDocumentElement();
// 3 创建一个linkman代码片段
Element linkmanEl = doc.createElement("linkman");
// 3.1 创建name,email,address,group 子元素
Element nameEl = doc.createElement("name");
Element emailEl = doc.createElement("email");
Element addressEl = doc.createElement("address");
Element groupEl = doc.createElement("group");
// 3.2 给name,email,address,group添加文本内容
nameEl.setTextContent("stef");
emailEl.setTextContent("stef@");
addressEl.setTextContent("四川");
groupEl.setTextContent("叮丁狼");
// 3.3 将name,email,address,group元素作为linkman的子元素
linkmanEl.appendChild(nameEl);
linkmanEl.appendChild(emailEl);
linkmanEl.appendChild(addressEl);
linkmanEl.appendChild(groupEl);

// 3.4 将linkman作为根元素的子元素
root.appendChild(linkmanEl);

// 4 同步操作
TransformerFactory.newInstance().newTransformer()
            .transform(new DOMSource(doc)
            , new StreamResult(file));
```

08_XML 约束（了解）

XML 文件,主要的作用是用来传输数据,以及作为配置文件. 如下面的代码,使用XML文件,将数据由 A 传输到 B,那么 B 就可以创建一个 User对象来保存 XML 中的数据.

```
<users>
    <user>
        <username>xiaoming</username>
    </user>
</users>
```

但是,B 在读取数据的时候,只能读取 user 标签,把该标签中的内容放在对应的 User 对象中,如果 XML 中的标签不再是 user,而是 user1,如下图:

```
<users>
    <user1>
        <username>xiaoming</username>
    </user1>
</users>
```

上面的XML文件,语法没有问题,传递给别人之后.不能将数据封装到User对象中.

可以使用XML约束来规范 XML 内容的写法.(dtd,schema) XML约束,了解即可,因为以后我们更多的是使用到别人提供好的约束文件,使用 idea 等工具,利用代码提示的方式写好 XML 即可.

小结

1 理解xml 的语法结构，能够看得懂xml语法，能够修改xml内容即可

2 理解 DOM Document Object Model

2 掌握使用 DOM 操作去读取xml中第二个联系人的信息