XML、XML解析、设计模式等





今日课程同学们需要学会什么?



在有些业务场景下, 存储数据或者传输 数据给别人的时候, 数据需要满足一定 的规范进行组织。

在有些业务场景下, XML文件中存储的数 存储数据或者传输 据是需要提取出来的。 如何方便的在XML文 件中进行数据的检索?

开发中还有一些比较常见的设计模式是需要掌握,理解设计模式有利于理解某些程序。



> XML

- ◆ XML概述
- ◆ XML的创建、语法规则
- ◆ XML文档约束方式一-DTD约束[了解]
- ◆ XML文档约束方式二-schema约束[了解]
- > XML解析技术
- > XML检索技术: Xpath
- > 设计模式:工厂模式
- > 设计模式:装饰模式



XML概述

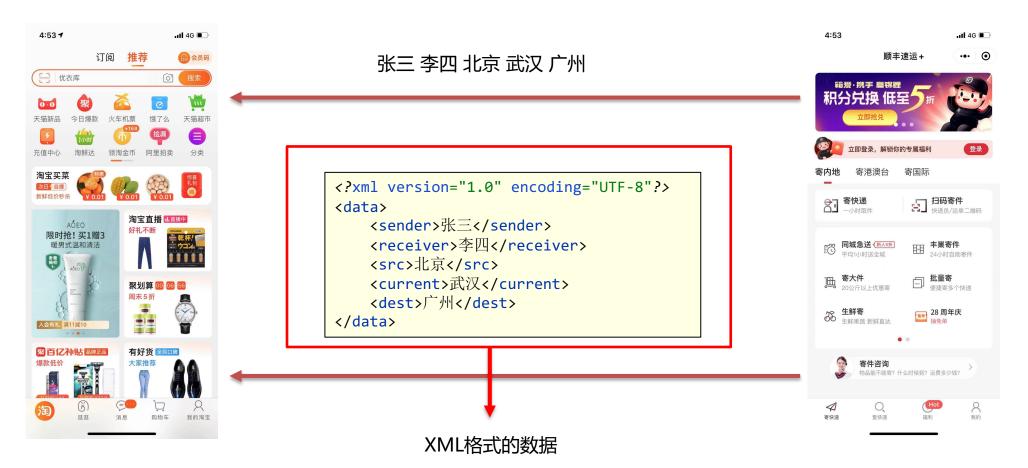
● XML是可扩展标记语言(eXtensible Markup Language)的缩写,它是是一种数据表示格式,可以描述非常复杂的数据结构,常用于传输和存储数据。

XML的几个特点和使用场景

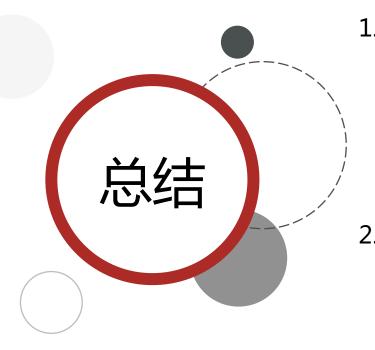
- 一是纯文本,默认使用UTF-8编码;二是可嵌套;
- 如果把XML内容存为文件,那么它就是一个XML文件。
- XML的使用场景:XML内容经常被当成消息进行网络传输,或者作为配置文件用于存储系统的信息。



多一句没有,少一句不行,用最短时间,教会最实用的技术!







1. XML是什么?

- XML的全称为(EXtensible Markup Language),是一种可扩展的标记语言
- 它是是一种数据表示格式,可以用于自定义数据格式。
- 2. XML的作用是什么?
 - 用于进行存储数据和传输数据
 - 作为软件的配置文件



> XML

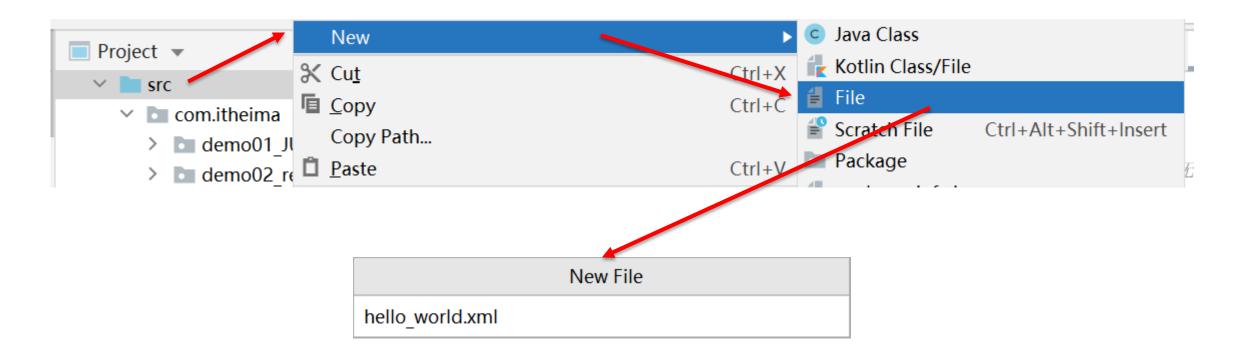
- ◆ XML概述
- ◆ XML的创建、语法规则
- ◆ XML文档约束方式一-DTD约束[了解]
- ◆ XML文档约束方式二-schema约束[了解]
- > XML解析技术
- > XML检索技术: Xpath
- > 设计模式:工厂模式
- > 设计模式:装饰模式



XML的创建

● 就是创建一个XML类型的文件,要求文件的后缀必须使用xml,如hello_world.xml

IDEA创建XML文件的操作步骤





XML的语法规则

● XML文件的后缀名为:xml

●_文档声明必须是第一行

<?xml version="1.0" encoding="UTF-8" ?>

version: XML默认的版本号码、该属性是必须存在的

encoding:本XML文件的编码



XML的标签(元素)规则

- 标签由一对尖括号和合法标识符组成: <name></name>,必须存在一个根标签,有且只能有一个。
- 标签必须成对出现,有开始,有结束: <name> </name>
- 特殊的标签可以不成对,但是必须有结束标记,如:

- 标签中可以定义属性,属性和标签名空格隔开,属性值必须用引号引起来<student id = "1" ></name>
- 标签需要正确的嵌套

<student id="1">
<name>张三
</name>
</student>





XML的其他组成

- XML文件中可以定义注释信息:<!- 注释内容 -->
- XML文件中可以存在以下特殊字符

```
< < 小于
> > 大于
& & 和号
' ' 单引号
" " 引号
```

● XML文件中可以存在CDATA区: <![CDATA[...内容...]]>





1. XML的组成格式要求是什么样的?

- 文件后缀必须是xml
- 文档声明必须是第一行
- 必须存在一个根标签,有且只能有一个
- XML文件中可以定义注释信息:<!- 注释内容 -->
- 标签必须成对出现,有开始,有结束标签: <name></name>
- 必须能够正确的嵌套



> XML

- ◆ XML概述
- ◆ XML的创建、语法规则
- ◆ XML文档约束方式一-DTD约束[了解]
- ◆ XML文档约束方式二-schema约束[了解]
- > XML解析技术
- XML检索技术: Xpath
- > 设计模式:工厂模式
- > 设计模式:装饰模式



什么是文档约束?



问题:由于XML文件可以自定义标签,导致XML文件可以随意定义,程序在解析的时候可能出现问题。



什么是文档约束?

● 文档约束:是用来限定xml文件中的标签以及属性应该怎么写。

以此强制约束程序员必须按照文档约束的规定来编写xml文件。

文档约束的分类

- DTD
- schema





XML文档约束-DTD的使用(了解)

需求:利用DTD文档约束,约束一个XML文件的编写。

分析:

①:编写DTD约束文档,后缀必须是.dtd

<!ELEMENT 书架 (书+)>

<!ELEMENT 书 (书名,作者,售价)>

<!ELEMENT 书名 (#PCDATA)>

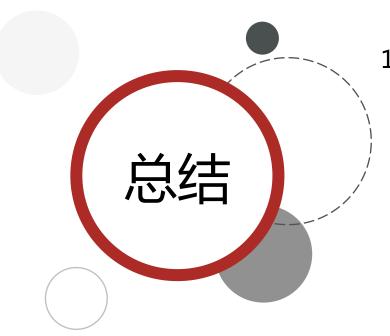
<!ELEMENT 作者 (#PCDATA)>

<!ELEMENT 售价 (#PCDATA)>

②:在需要编写的XML文件中导入该DTD约束文档

③:按照约束的规定编写XML文件的内容。





1. XML的文档约束-DTD的作用和问题?

- 可以约束XML文件的编写。
- 不能约束具体的数据类型。



> XML

- ◆ XML概述
- ◆ XML的创建、语法规则
- ◆ XML文档约束方式一-DTD约束[了解]
- ◆ XML文档约束方式二-schema约束[了解]
- > XML解析技术
- > XML检索技术: Xpath
- > 设计模式:工厂模式
- > 设计模式:装饰模式



文档约束-schema

- schema可以约束具体的数据类型,约束能力上更强大。
- schema本身也是一个xml文件,本身也受到其他约束文件的要求,所以编写的更加严谨







XML文档约束-schema的使用(了解)

需求:利用schema文档约束,约束一个XML文件的编写。

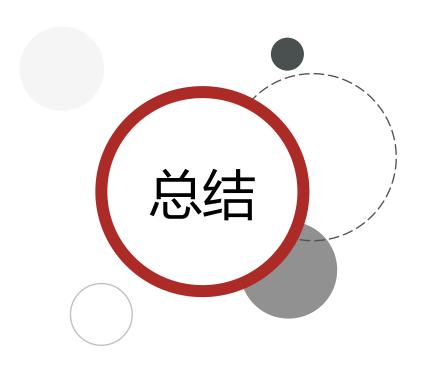
分析:

①:编写schema约束文档,后缀必须是.xsd,具体的形式到代码中观看。

②:在需要编写的XML文件中导入该schema约束文档

③:按照约束内容编写XML文件的标签。





- 1. XML的文档约束-schema的优点?
 - 可以约束XML文件的标签内容格式,以及具体的数据类型。
 - 本身也是xml文件,格式更严谨。



- > XML
- > XML解析技术
 - ◆ XML解析技术概述
 - ◆ Dom4J解析XML文件
 - ◆ Dom4J解析XML文件-案例实战
- > XML检索技术: Xpath
- > 设计模式:工厂模式
- > 设计模式:装饰模式





- 1、XML的数据的作用是什么,最终需要怎么处理?
 - 存储数据、做配置信息、进行数据传输。
 - 最终需要被程序进行读取,解析里面的信息。

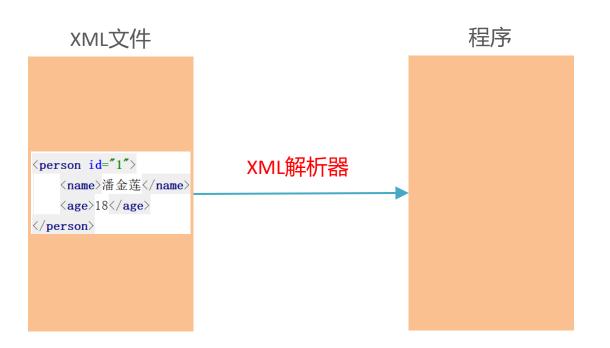


什么是XML解析

● 使用程序读取XML中的数据

两种解析方式

- SAX解析
- DOM解析



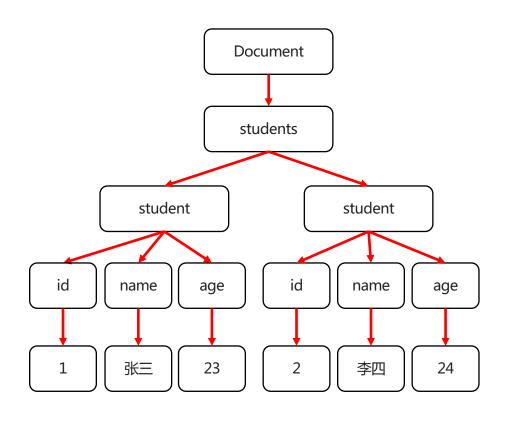


Dom常见的解析工具

名称	说明
JAXP	SUN公司提供的一套XML的解析的API
JDOM	JDOM是一个开源项目,它基于树型结构,利用纯JAVA的技术对XML文档实现解析、生成、 序列化以及多种操作。
dom4j	是JDOM的升级品,用来读写XML文件的。具有性能优异、功能强大和极其易使用的特点,它的性能超过sun公司官方的dom 技术,同时它也是一个开放源代码的软件,Hibernate也用它来读写配置文件。
jsoup	功能强大DOM方式的XML解析开发包,尤其对HTML解析更加方便



DOM解析解析文档对象模型



Document对象:整个xml文档

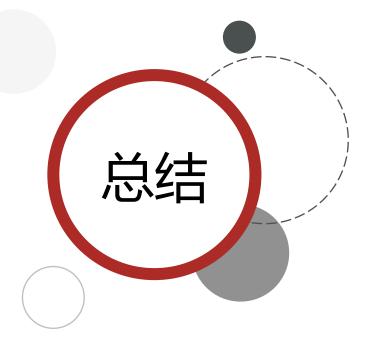
Element对象:标签

Attribute对象:属性 _

Text对象:文本内容

Node对象





1. Dom解析的文档对象模型是怎么样的

Document对象:整个xml文档

Node对象

Element对象:标签

Attribute对象:属性

Text对象:文本内容

2. Dom解析常用技术框架

Dom4J



- > XML
- > XML解析技术
 - ◆ XML解析技术概述
 - ◆ Dom4J解析XML文件
 - ◆ Dom4J解析XML文件-案例实战
- > XML检索技术: Xpath
- ▶ 设计模式:工厂模式
- > 设计模式:装饰模式





使用Dom4J解析出XML文件

需求:使用Dom4J把一个XML文件的数据进行解析

分析:

- ① 下载Dom4j框架,官网下载。
- ② 在项目中创建一个文件夹: lib
- ③ 将dom4j-2.1.1.jar文件复制到 lib 文件夹
- ④ 在jar文件上点右键,选择 Add as Library -> 点击OK
- ⑤ 在类中导包使用



Dom4j解析XML-得到Document对象

SAXReader类

构造器/方法	说明
public SAXReader()	创建Dom4J的解析器对象
Document read(String url)	加载XML文件成为Document对象

Document类

方法名	说明
Element getRootElement()	获得根元素对象



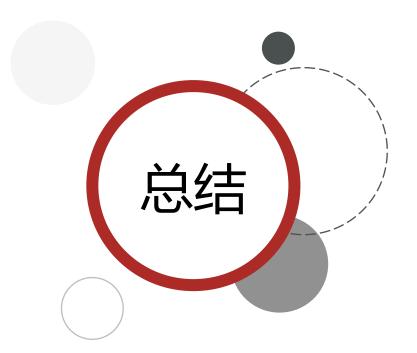
- > XML
- > XML解析技术
 - ◆ XML解析技术概述
 - ◆ Dom4J解析XML文件
 - ◆ Dom4J解析XML文件中的各种节点
 - ◆ Dom4J解析XML文件-案例实战
- > XML检索技术: Xpath
- > 设计模式:工厂模式
- > 设计模式:装饰模式



Dom4j解析XML的元素、属性、文本

方法名	说明
List <element> elements()</element>	得到当前元素下所有子元素
List < Element > elements (String name)	得到当前元素下指定名字的子元素返回集合
Element element(String name)	得到当前元素下指定名字的子元素,如果有很多名字相同的返回第一个
String getName()	得到元素名字
String attributeValue(String name)	通过属性名直接得到属性值
String elementText(子元素名)	得到指定名称的子元素的文本
String getText()	得到文本





- 1. Dom4J的解析思想?
 - 得到文档对象Document,从中获取元素对象和内容。



- > XML
- > XML解析技术
 - ◆ XML解析技术概述
 - ◆ Dom4J解析XML文件
 - ◆ Dom4J解析XML文件中的各种节点
 - ◆ Dom4J解析XML文件-案例实战
- > XML检索技术: Xpath
- > 设计模式:工厂模式
- > 设计模式:装饰模式





XML解析案例

需求:利用Dom4J的知识,将Contact.xml文件中的联系人数据封装成List集合,其中每个元素是实体类Contact。打印输出 List 中的每个元素。

```
<?xml version="1.0" encoding="UTF-8"?>
<contactList>
    <contact id="1">
       <name>潘金莲</name>
       <gender>女</gender>
       <email>panpan@itcast.cn</email>
    </contact>
    <contact id="2">
       <name>武松</name>
       <gender>男</gender>
       <email>wusong@itcast.cn</email>
   </contact>
    <contact id="3">
       <name>武大狼</name>
       <gender>男</gender>
       <email>wuda@itcast.cn</email>
    </contact>
</contactList>
```

```
Contact{id=1, name='潘金莲', gender='女', email='panpan@itcast.cn'}
Contact{id=2, name='武松', gender='男', email='wusong@itcast.cn'}
Contact{id=3, name='武大狼', gender='男', email='wuda@itcast.cn'}
```





- 1. Dom4J的解析后的数据形式。
 - 通常数据会封装成Java的对象,如单个对象,或者集合对象形式

0



- > XML
- > XML解析技术
- ➤ XML检索技术: Xpath
- 设计模式:工厂模式
- > 设计模式:装饰模式





如果需要从XML文件中检索需要的某个信息(如name)怎么解决?

- Dom4j需要进行文件的全部解析,然后再寻找数据。
- Xpath技术更加适合做信息检索。



XPath介绍

- XPath在解析XML文档方面提供了一独树一帜的路径思想,更加优雅,高效
- XPath使用路径表达式来定位XML文档中的元素节点或属性节点。

示例

- /元素/子元素/孙元素
- //子元素//孙元素





使用Xpath检索出XML文件

需求:使用Dom4J把一个XML文件的数据进行解析

分析:

- ① 导入jar包(dom4j和jaxen-1.1.2.jar), Xpath技术依赖Dom4j技术
- ② 通过dom4j的SAXReader获取Document对象
- ③ 利用XPath提供的API,结合XPath的语法完成选取XML文档元素节点进行解析操作。
- ④ Document中与Xpath相关的API如下:

方法名	说明
Node selectSingleNode("表达式")	获取符合表达式的唯一元素
List <node> selectNodes("表达式")</node>	获取符合表达式的元素集合



Xpath的四大检索方案

- 绝对路径
- 相对路径
- 全文检索
- 属性查找



XPath:绝对路径

● 采用绝对路径获取从根节点开始逐层的查找/contactList/contact/name节点列表并打印信息

方法名	说明
/根元素/子元素/孙元素	从根元素开始,一级一级向下查找,不能跨级



XPath:相对路径

- 先得到根节点contactList
- 再采用相对路径获取下一级contact 节点的name子节点并打印信息

方法名	说明
./子元素/孙元素	从当前元素开始,一级一级向下查找,不能跨级



XPath:全文搜索

● 直接全文搜索所有的name元素并打印

方法名	。 说明
//contact	找contact元素,无论元素在哪里
//contact/name	找contact,无论在哪一级,但name一定是contact的子节点
//contact//name	contact无论在哪一种,name只要是contact的子孙元素都可以找到

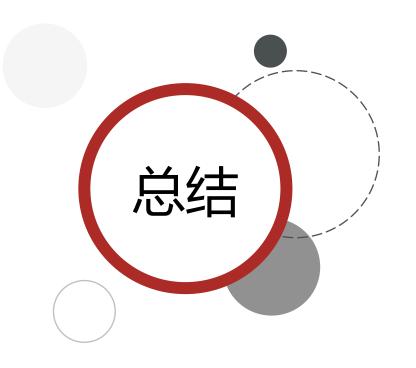


XPath:属性查找

● 在全文中搜索属性,或者带属性的元素

方法名	说明
//@属性名	查找属性对象,无论是哪个元素,只要有这个属性即可。
//元素[@属性名]	查找元素对象,全文搜索指定元素名和属性名。
//元素//[@属性名='值']	查找元素对象,全文搜索指定元素名和属性名,并且属性值相等。





1. Xpath作用,四大类。

● 检索XML文件中的信息

● 绝对路径:/根元素/子元素/孙元素

● 相对路径:./子元素/孙元素

● 全文检索://contact

■ 属性查找://@属性名 、//元素[@属性名]、//元素//[@属性名= '值

]



- > XML
- > XML解析技术
- XML检索技术: Xpath
- > 设计模式:工厂模式
- 〉 设计模式:装饰模式



什么是工厂设计模式?

- 之前我们创建类对象时,都是使用new对象的形式创建,在很多业务场景下也提供了不直接new的方式。
- 工厂模式(Factory Pattern)是 Java 中最常用的设计模式之一, 这种类型的设计模式属于创建型模式,它提供了一种获取对象的方式。

工厂设计模式的作用:

- 工厂的方法可以封装对象的创建细节,比如:为该对象进行加工和数据注入。
- 可以实现类与类之间的解耦操作(核心思想)。





1. 工厂设计模式的作用

- 对象通过工厂的方法创建返回,工厂的方法可以为该对象进行加工和数据注入
- 可以实现类与类之间的解耦操作(核心思想)



- > XML
- > XML解析技术
- XML检索技术: Xpath
- 设计模式:工厂模式
- > 设计模式:装饰模式



什么是装饰设计模式?

● 创建一个新类,包装原始类,从而在新类中提升原来类的功能。

装饰设计模式的作用:

● 作用:装饰模式指的是在不改变原类的基础上,动态地扩展一个类的功能。

InputStream(抽象父类)

FileInputStream(实现子类,读写性能较差)

BufferedInputStream(实现子类,装饰类,读写性能高)

- ① 定义父类。
- ② 定义原始类,继承父类,定义功能。
- ③ 定义装饰类,继承父类,包装原始类,增强功能!!





- 1. 装饰设计模式的作用?
 - 装饰模式指的是在不改变原类的基础上, 动态地扩展一个类的功能







传智教育旗下高端IT教育品牌