# XML的案例与作业：

给通讯录程序设计一个xml文件存储联系人的数据。

联系人属性： 编号（联系人的属性）、 姓名、性别、电话、QQ、邮箱

文件名：contact.xml

1. 要求：

1）使用xml语法进行描述联系人信息

2）能够存储多个联系人

结合上课笔记完成

# 通过下列的DTD在idea中写出它对应的XML文件

* DTD约束文件如下：

1. 根元素是contactList，contactList中包含1~n个contact子元素，contact子元素包含name,gender,phone,qq,email元素，这个元素必须依次出现，并且元素中都可以包含文本元素(#PCDATA)
2. 其中contact有id属性，必须，并且不能重复，不能以数字开头
3. contact有一个可选的属性vip，取值是true或false
4. 按此DTD约束创建XML文件，并添加3条contact记录。注：括号前要有空格

* DTD文件

<!ELEMENT contactList (contact+)>

<!ELEMENT contact (name,gender,phone,qq,email)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT gender (#PCDATA)>

<!ELEMENT phone (#PCDATA)>

<!ELEMENT qq (#PCDATA)>

<!ELEMENT email (#PCDATA)>

<!ATTLIST contact

id ID #REQUIRED

vip (true|false) #IMPLIED

>

* 写出如下的XML文件

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

<!DOCTYPE contactList SYSTEM "contact.dtd" >

<contactList>

<contact id="S100" vip="true">

<name>张三</name>

<gender>男</gender>

<phone>13565784567</phone>

<qq>394832048</qq>

<email>zhang@itheima.com</email>

</contact>

<contact id="S200" vip="false">

<name>李四</name>

<gender>女</gender>

<phone>15687960980</phone>

<qq>109456893</qq>

<email>li@itheima.com</email>

</contact>

</contactList>

# 通过Schema在idea下写出对应的XML文件

* 这是这个名为 "shoporder.xsd" 的 schema 文件的文档清单，写出它的XML文件

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.itheima.com/shoporder">

<!-- 订单 -->

<xs:element name="shoporder">

<xs:complexType>

<xs:sequence>

<!-- 依次出现下订单的人 -->

<xs:element name="orderperson" type="xs:string" />

<!-- 收件人的地址 -->

<xs:element name="shopto">

<xs:complexType>

<!-- 依次出现姓名，地址，城市，国家 -->

<xs:sequence>

<xs:element name="name" type="xs:string" />

<xs:element name="address" type="xs:string" />

<xs:element name="city" type="xs:string" />

<xs:element name="country" type="xs:string" />

</xs:sequence>

</xs:complexType>

</xs:element>

<!-- 订单具体的项 -->

<xs:element name="item" maxOccurs="unbounded">

<xs:complexType>

<!-- 依次出现标题，便条(可以不出现)，数量(正整数)，价格(小数) -->

<xs:sequence>

<xs:element name="title" type="xs:string" />

<xs:element name="note" type="xs:string" minOccurs="0" />

<xs:element name="quantity" type="xs:positiveInteger" />

<xs:element name="price" type="xs:decimal" />

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

<!-- 订单有个必须的属性orderid -->

<xs:attribute name="orderid" type="xs:string" use="required" />

</xs:complexType>

</xs:element>

</xs:schema>

* 答案：

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

<shoporder orderid="889923" xmlns="http://www.itheima.com/shoporder"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.itheima.com/shoporder shoporder.xsd ">

<orderperson>张三</orderperson>

<shopto>

<name>张小三</name>

<address>珠吉路300号</address>

<city>广州</city>

<country>中国</country>

</shopto>

<item>

<title>帝国时代游戏</title>

<note>特别版</note>

<quantity>1</quantity>

<price>100.90</price>

</item>

<item>

<title>布娃娃</title>

<quantity>1</quantity>

<price>99.90</price>

</item>

</shoporder>

# 有如下书籍XML文档

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

<bookstore>

<book category="CHILDREN">

<title>Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

<book category="WEB">

<title>Learning XML</title>

<author>Erik T. Ray</author>

<year>2003</year>

<price>39.95</price>

</book>

</bookstore>

封装成List<Book>，并且输出封装好的集合

Book(String category, String title, String author, int year, double price)

将其封装成List<Book>对象返回，并且输出集合中的每个元素

实现步骤：

1. 得到document对象
2. 解析book.xml，得到所有book元素
3. 创建一个List，用于封装Book
4. 每个元素是一本书籍，得到category的属性值，并且封装。
5. 得到子元素的文本，并且封装到每个属性
6. 每循环一次将元素添加到集合中
7. 输出所有书籍

package com.itheima;

import com.itheima.entity.Book;

import org.dom4j.Document;

import org.dom4j.DocumentException;

import org.dom4j.Element;

import org.dom4j.io.SAXReader;

import java.util.ArrayList;

import java.util.List;

public class Demo1ReadBook {

public static void main(String[] args) throws DocumentException {

// 得到document对象

Document document = new SAXReader().read(Demo1ReadBook.class.getResourceAsStream("/book.xml"));

// 解析book.xml，得到所有book元素

Element rootElement = document.getRootElement();

List<Element> bookElements = rootElement.elements("book");

// 创建一个List，用于封装Book

List<Book> books = new ArrayList<>();

// 每个元素是一本书籍，得到category的属性值

for (Element bookElement : bookElements) {

// 遍历每个子元素

Book book = new Book();

// 得到元素的属性

book.setCategory(bookElement.attributeValue("category"));

// 得到子元素的文本，并且封装

book.setTitle(bookElement.elementTextTrim("title"));

book.setAuthor(bookElement.elementTextTrim("author"));

book.setYear(Integer.parseInt(bookElement.elementTextTrim("year")));

book.setPrice(Double.parseDouble(bookElement.elementTextTrim("price")));

// 每循环一次将元素添加到集合中

books.add(book);

}

// 输出所有书籍

for (Book book : books) {

System.out.println(book);

}

}

}