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

<bookstore>

<book id="001" number="2045">

<name>song fo ice and fire</name>

<author>judgematy</author>

<year>2014</year>

<price>89</price>

</book>

<book id="002">

<name>fail story</name>

<year>2004</year>

<price>77</price>

<language>English</language>

</book>

</bookstore>

# DOM解析方式

import java.io.\*;

import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.Document;

import org.w3c.dom.NodeList;

import org.w3c.dom.Node;

import org.w3c.dom.NamedNodeMap;

import javax.xml.parsers.DocumentBuilder;

import org.xml.sax.SAXException;

import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom .Element;

public class main

{

public static void main(String[] args)

{

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

try

{

//创建DocumentBuilder

DocumentBuilder db = dbf.newDocumentBuilder();

//加载xml文件

Document document = db.parse("F:\\javaStuff\\books.xml");

//获取所有book结点

NodeList bookList = document.getElementsByTagName("book");

System.out.println("sum of books is "+bookList.getLength());

//NodeList.length()获取长度

for (int i = 0;i<bookList.getLength() ;i++ )

{

//获取每一个book

Node book = bookList.item(i);

//获取book节点的所有属性集合

NamedNodeMap attrs = book.getAttributes();

System.out.println("book "+(i+1)+" have attrs "+attrs.getLength());

for (int j =0;j<attrs.getLength() ; j++)

{

System.out.println(attrs.item(j).getNodeName()+"："+attrs.item(j).getNodeValue());

}

NodeList childNodes = book.getChildNodes();

//System.out.println("\n"+childNodes.getLength());

for (int k=0;k<childNodes.getLength() ;k++ )

{

//区分出text类型的node以及element类型的node

if (childNodes.item(k).getNodeType() == Node.ELEMENT\_NODE)

{

System.out.println(childNodes.item(k).getNodeName()+"："+childNodes.item(k).getFirstChild().getNodeValue());

}

}

//前提：知道book节点有且只有一个id属性

/\*Element book = (Element)bookList.item(i);

String attr = book.getAttribute("id");

\*/

}

}

catch (ParserConfigurationException e)

{

e.printStackTrace();

}

catch (SAXException e)

{

e.printStackTrace();

}

catch (IOException e)

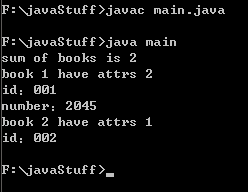
{

e.printStackTrace();

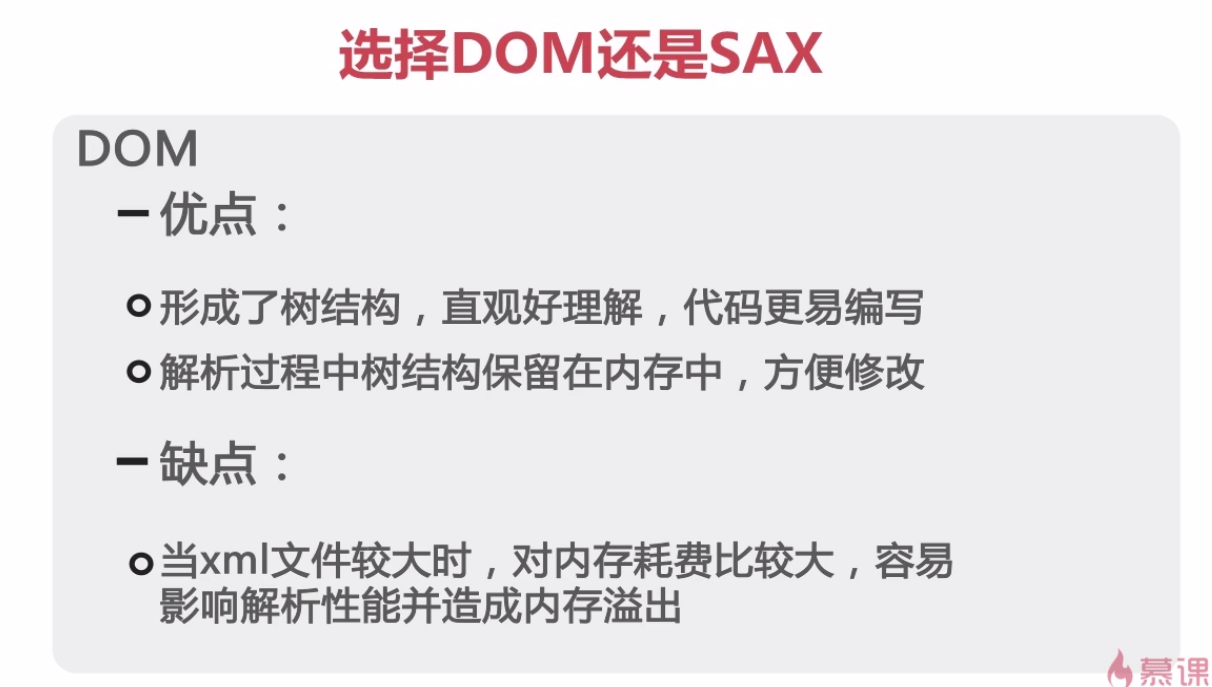
}

}

}



# DOM4J解析方式（建议使用）





import org.dom4j.Document;

import org.dom4j.io.SAXReader;

import org.dom4j.Element;

import org.dom4j.DocumentException;

import org.dom4j.Attribute;

import java.io.File;

import java.util.Iterator;

import java.util.List;

class xmlDemo

{

public static void main(String[] args)

{

//解析book.xml

//创建SAXReader以加载.xml文件

SAXReader reader = new SAXReader();

try

{

//获取document对象

Document document = reader.read(new File("books.xml"));

//获得根节点

Element bookstore = document.getRootElement();

//获得迭代器

Iterator it = bookstore.elementIterator();

//遍历迭代器

while (it.hasNext())

{

Element book = (Element)it.next();

//获取book的属性名和属性值

List<Attribute> bookAttrs = book.attributes();

for (Attribute arr : bookAttrs )

{

System.out.println("---------------------------------");

System.out.println("节点："+arr.getName()+" "+arr.getValue());

System.out.println("---------------------------------");

}

Iterator itt = book.elementIterator();

while (itt.hasNext())

{

Element bookChild = (Element)itt.next();

System.out.println(" "+bookChild.getName()+bookChild.getStringValue());

}

}

}

catch (DocumentException e)

{

e.printStackTrace();

}

}

}

