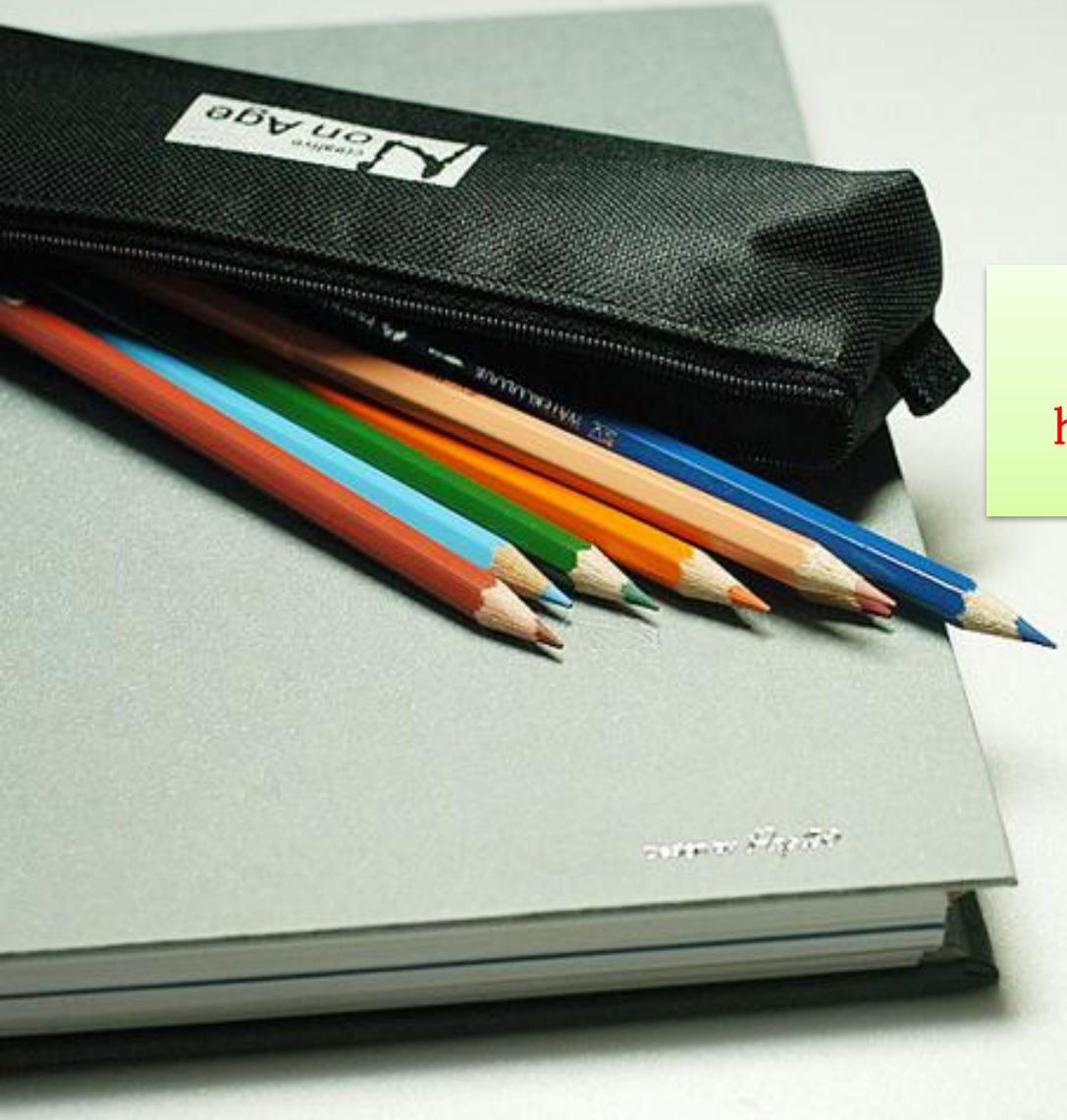
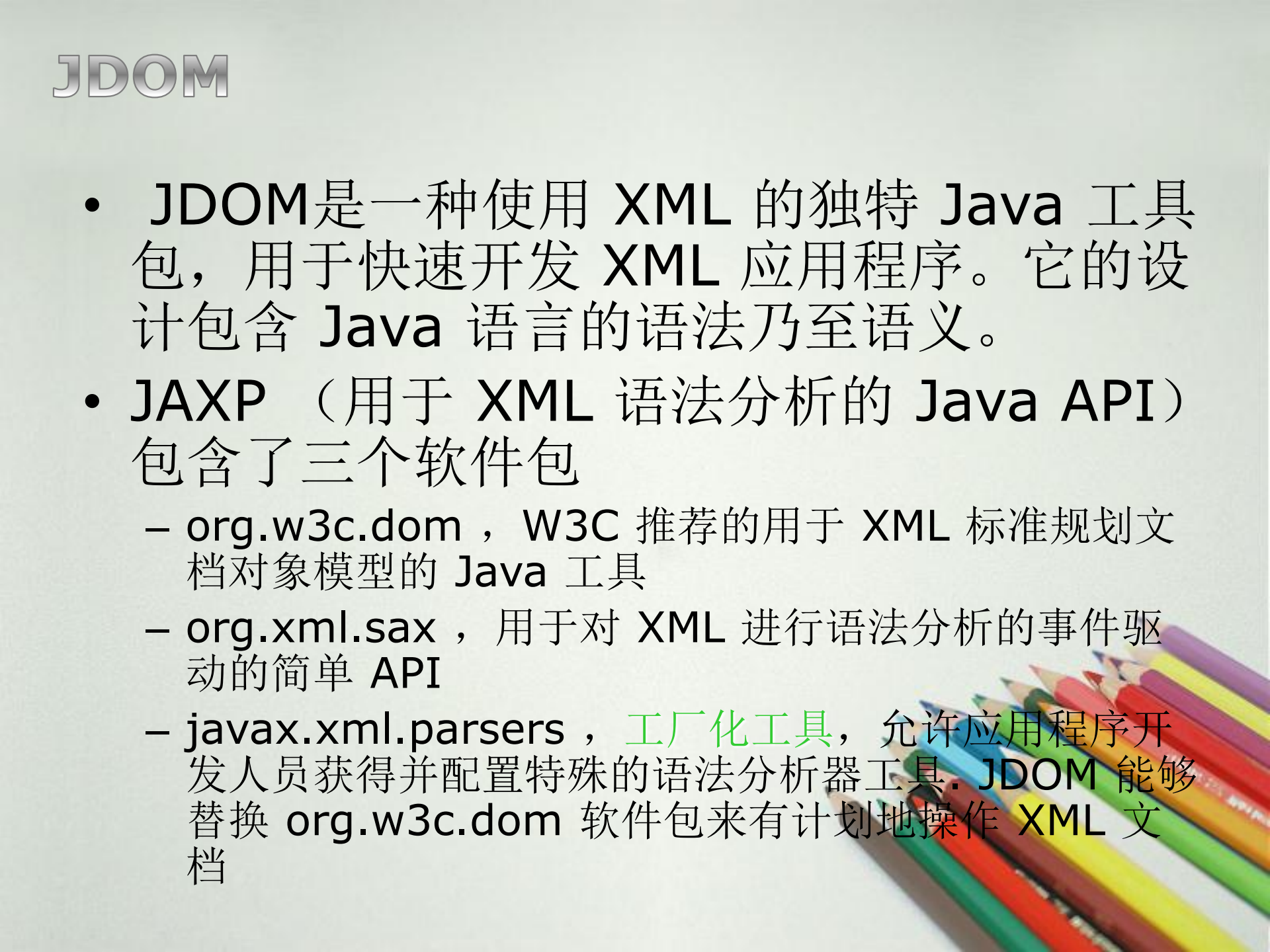


北京圣思园科技有限公司
http://www.shengsiyuan.com

主讲人：张龙

JDOM

- JDOM是一种使用 XML 的独特 Java 工具包，用于快速开发 XML 应用程序。它的设计包含 Java 语言的语法乃至语义。
- JAXP （用于 XML 语法分析的 Java API）包含了三个软件包
  - org.w3c.dom ， W3C 推荐的用于 XML 标准规划文档对象模型的 Java 工具
  - org.xml.sax ， 用于对 XML 进行语法分析的事件驱动的简单 API
  - javax.xml.parsers ， 工厂化工具， 允许应用程序开发人员获得并配置特殊的语法分析器工具. JDOM 能够替换 org.w3c.dom 软件包来有计划地操作 XML 文档



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。（http://jdom.org）
- JDOM 直接为JAVA编程服务。它利用更为强有力的JAVA语言的诸多特性（方法重载、集合概念等），把SAX和DOM的功能有效地结合起来。
- JDOM是用Java语言读、写、操作XML的新API函数。在直接、简单和高效的前提下，这些API函数被最大限度的优化。

JDOM

- 在使用设计上尽可能地隐藏原来使用XML过程中的复杂性。利用JDOM处理XML文档将是一件轻松、简单的事。
- JDOM 主要用来弥补DOM及SAX在实际应用当中的不足之处。这些不足之处主要在于SAX没有文档修改、随机访问以及输出的功能，而对于DOM来说，JAVA程序员在使用时来用起来总觉得不太方便。

JDOM

- DOM的缺点主要是由于DOM是一个接口定义语言（IDL）,它的任务是在不同语言实现中的一个最低的通用标准，并不是为JAVA特别设计的。
- 在 JDOM 中，XML 元素就是 Element 的实例，XML 属性就是 Attribute 的实例，XML 文档本身就是 Document 的实例。

W3C DOM 设计的局限性

- 语言独立
  - DOM 并不是用人们心目中的 Java 语言设计的。虽然这种方法保留了在不同语言中非常相似的 API，它也使那些习惯 Java 语言的程序员感到更麻烦。
  - 例如：Java 语言内建了一种 String 类，而 DOM 则规范定义了自己的 Text 类。

W3C DOM 设计的局限性

- 严格的层次结构
  - DOM API 直接沿袭了 XML 规范。在 XML 中，每件东西都是一个结点，因此您能在 DOM 中找到一个几乎每件东西都可以扩展的基于 Node 的接口和返回 Node 的一系列方法。
  - 就多态性的观点来讲，它是优秀的，但鉴于如上解释，它在 Java 语言中的应用是困难而且不便的，其中从 Node 向叶类型作显式向下类型转换会导致代码的冗长和难以理解。

W3C DOM 设计的局限性

- 接口驱动
  - 公共 DOM API 仅由接口组成。w3c 对提供实现并不感兴趣，它只对定义接口（比较有意义）感兴趣。但它也意味着作为 Java 程序员使用 API 在创建 XML 对象时增加了负担，因为 w3c 标准大量使用工厂化的类和类似的灵活的但不直接的模式。
  - 在某些应用中，XML 文档是仅由语法分析器建立的，而从不会由应用程序级代码建立，这是不相关的。但是，随着 XML 更广泛的使用，并不是所有问题都继续需要由语法分析器来驱动。应用程序的开发人员需要一个更方便的方法有计划地构造 XML 对象。

W3C DOM 设计的局限性

- 对于程序员，这些约束意味着庞大（在内存占用和接口大小方面）的和难掌握的 API，学习和使用都很难。
- JDOM 是作为一种轻量级 API 被制定的，最主要的是它是以 Java 为中心的。它在遵循 DOM 主要规则的基础上除去了上述缺点

W3C DOM 设计的局限性

- JDOM 是 Java 平台专用的。
  - 只要有可能，API 都使用 Java 语言的内建 String 支持，因此文本值也适用于 String 。它还可利用 Java 2 平台的类集，如 List 和 Iterator ，给程序员提供了一个丰富的并且和 Java 语言类似的环境。

W3C DOM 设计的局限性

- 没有层次性。
  - 在 JDOM 中，XML 元素就是 Element 的实例，XML 属性就是 Attribute 的实例，XML 文档本身就是 Document 的实例。由于在 XML 中所有这些都代表了不同的概念，因此它们总是作为自己的类型被引用，而不是作为一个含糊的“结点”。

W3C DOM 设计的局限性

- 类驱动。
  - 因为 JDOM 对象就是像 Document 、Element 和 Attribute 这些类的直接实例，因此创建一个新 JDOM 对象就如在 Java 语言中使用 new 操作符一样容易。它还意味着不需要进行工厂化接口配置 -- JDOM 的使用是直截了当的。

JDOM

- 因为 JDOM 对象就是像 Document、Element 和 Attribute 这些类的直接实例，因此创建一个新 JDOM 对象就如在 Java 语言中使用 new 操作符一样容易。JDOM 的使用是直截了当的。
- JDOM 使用标准的 Java 编码模式。只要有可能，它使用 Java new 操作符而不使用复杂的工厂模式，使对象操作即便对于初学用户也很方便。

JDOM

- JDOM是由以下几个包组成的
  - org.jdom包含了所有的xml文档要素的java类
  - org.jdom.adapters包含了与dom适配的java类
  - org.jdom.filter包含了xml文档的过滤器类
  - org.jdom.input包含了读取xml文档的类
  - org.jdom.output包含了写入xml文档的类
  - org.jdom.transform包含了将jdomxml文档接口转换为其他xml文档接口
  - org.jdom.xpath包含了对xml文档xpath操作的类

JDOM

- JDOM类说明
- org.jdom这个包里的类是你解析xml文件后所要用到的所有数据类型。
  - Attribute
  - CDATA
  - Coment
  - DocType
  - Document
  - Element
  - EntityRef
  - Namespace
  - ProscessingInstruction
  - Text

JDOM

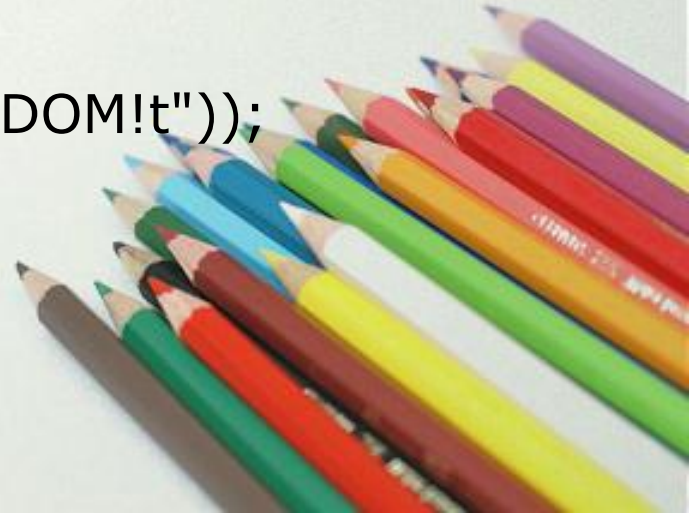
- 输入类，一般用于文档的创建工作
  - SAXBuilder
  - DOMBuilder
  - ResultSetBuilder

JDOM

- org.jdom.output输出类，用于文档转换输出
  - XMLOutputter
  - SAXOutputter
  - DomOutputter
  - JTreeOutputter

JDOM主要使用方法

- Document类
- Document的操作方法：
- Element root = new Element("GREETING");
- Document doc = new Document(root);
- root.setText("Hello JDOM!");
- 或者简单的使用
- Document doc = new Document(new Element("GREETING").setText("Hello JDOM!t"));

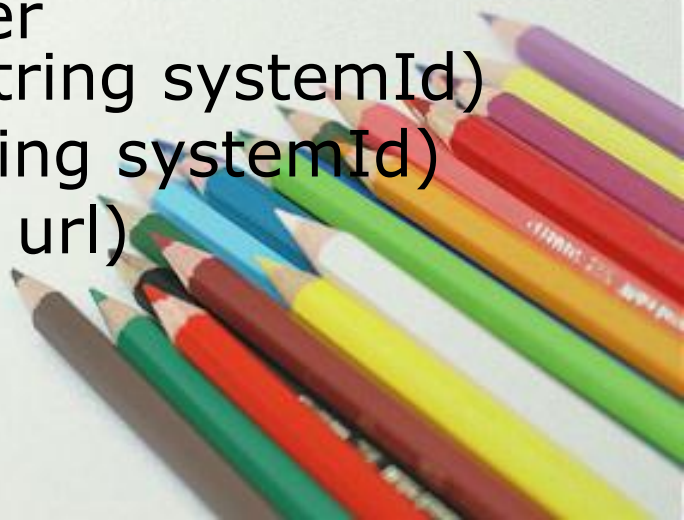


JDOM主要使用方法

- 这点和DOM不同。Dom则需要更为复杂的代码，如下：
- DocumentBuilderFactory factory =DocumentBuilderFactory.newInstance();
- DocumentBuilder builder =factory.newDocumentBuilder();
- Document doc = builder.newDocument();
- Element root =doc.createElement("root");
- Text text = doc.createText("This is the root");
- root.appendChild(text);
- doc.appendChild(root);

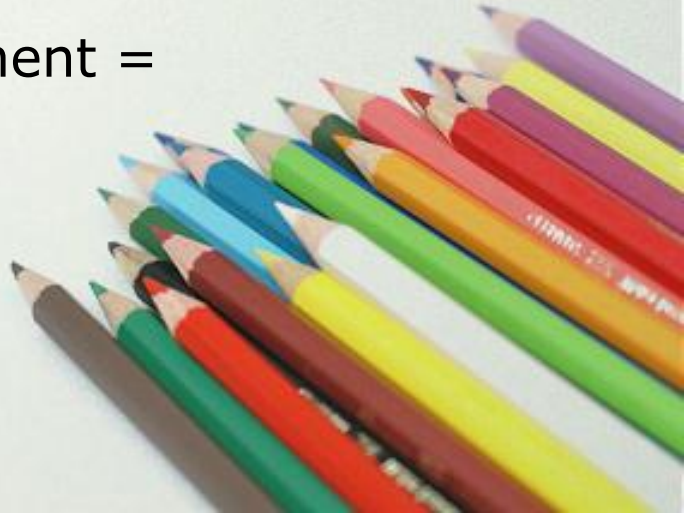
JDOM主要使用方法

- 可以使用SAXBuilder的build方法来解析一个流从而得到一个Document对象
  - Document build(java.io.File file)
  - Document build(org.xml.sax.InputSource in)
  - Document build(java.io.InputStream in)
  - Document build(java.io.InputStream in, java.lang.String systemId)
  - Document build(java.io.Reader characterStream)
  - Document build(java.io.Reader characterStream, java.lang.String systemId)
  - Document build(java.lang.String systemId)
  - Document build(java.net.URL url)



JDOM主要使用方法

- DOM的Document和JDOM的Document之间的相互转换使用方法
  - DOMBuilder builder = new DOMBuilder();
  - org.jdom.Document jdomDocument = builder.build(domDocument);
  - DOMOutputter converter = new DOMOutputter();// work with the JDOM document…
  - org.w3c.dom.Document domDocument = converter.output(jdomDocument);
  - // work with the DOM document…



XML文档输出

- XMLOutPutter类：
- JDOM的输出非常灵活,支持很多种io格式以及风格的输出
- Document doc = new Document(...);
- XMLOutputter outp = new XMLOutputter();
- outp.output(doc, fileOutputStream); // Raw output
- outp.setTextTrim(true); // Compressed output
- outp.output(doc, socket.getOutputStream());
- outp.setIndent(" ");// Pretty output
- outp.setNewlines(true);
- outp.output(doc, System.out);