



计算机与信息科学系列规划教材
JISUANJI YU XINXI KEXUE XILIE GUIHUA JIAOCAI

JavaWeb

高级程序设计与 上机指导

郑 龙 何敏藩 康红宴 邢立宁 马 文
编 著

计算机与信息科学系列规划教材

JavaWeb 高级程序 设计与上机指导

编 著 郑 龙 何敏藩 康红宴
邢立宁 马 文

湖南大学 出版社

内 容 简 介

本书对 Java 语言高级技术内容进行了全面、详细的讲解。主要介绍了解析 XML、EL 表达式、自定义标签、JSP 标准标签、过滤器与监听器、连接池与数据源、文件上传下载与图片水印、Web 服务等相关知识,且每章都配有实例、要点和作业,帮助读者理解和掌握书中的内容。本书适合作为计算机相关专业“JavaWeb 程序设计”课程的培训教材,也可以作为程序设计员或对 JavaWeb 编程感兴趣读者的参考书,还适合编程爱好者和自学 JavaWeb 编程的读者使用。

图书在版编目(CIP)数据

JavaWeb 高级程序设计与上机指导/郑龙,何敏藩,康红宴,邢立宁,马文编著. —长沙:湖南大学出版社,2019. 3

(计算机与信息科学系列规划教材)

ISBN 978-7-5667-1683-5

I. ①J… II. ①郑… ②何… ③康… ④邢… ⑤马… III. ①JAVA 语言—程序设计 IV. ①TP312. 8

中国版本图书馆 CIP 数据核字(2018)第 253382 号

JavaWeb 高级程序设计与上机指导

JAVEWEB GAOJI CHENGXU SHEJI YU SHANGJI ZHIDAO

编 著:郑 龙 何敏藩 康红宴 邢立宁 马 文

责任编辑:张建平 责任校对:尚楠欣

印 装:北京虎彩文化传播有限公司

开 本:787×1092 16 开 印张:13. 25 字数:315 千

版 次:2019 年 3 月第 1 版 印次:2019 年 3 月第 1 次印刷

书 号:ISBN 978-7-5667-1683-5

定 价:40. 00 元

出 版 人:雷 鸣

出版发行:湖南大学出版社

社 址:湖南·长沙·岳麓山 邮 编:410082

电 话:0731-88822559(发行部),88821343(编辑室),88821006(出版部)

传 真:0731-88649312(发行部),88822264(总编室)

网 址:<http://www.hnupress.com>

电子邮箱:presszhangjp@hnu.cn

版权所有,盗版必究

湖南大学出版社凡有印装差错,请与发行部联系

计算机与信息科学系列规划教材 编委会

主任:周忠宝

编委:周忠宝 郑 龙 何敏藩

邢立宁 罗俊海 姚 锋

叶朝晖 邓劲生 姚煊道

邹 伟 王 浩 张 章

肖 丹 蔡 琴 付 艳

周 滔 周 舟

编著与设计单位:湖南大学

国防科技大学

电子科技大学

佛山科学技术学院

长沙学院

深圳华大乐业教育科技有限公司

前 言

时光荏苒,如白驹过隙,一转眼中国互联网已走过了 30 年的历程。回首过去,人工智能、云计算、移动支付,这些互联网产物不仅迅速占据了我们的生活,刷新了我们对科技发展的认知,而且也提高了我们的生活质量水平。人们谈论的话题也离不开这些,例如:人工智能是否会替代人类,成为工作的主要劳动力;数字货币是否会代替纸币流通于市场;虚拟现实体验到底会有多真实多刺激。从这种现象中不难发现,互联网的辐射面在不断变广,计算机科学与信息技术发展的普适性在不断变强,信息技术如化雨春风,润物无声地全面融入,颠覆了我们的生活。

1987 年,我国网络专家钱天白通过拨号方式在国际互联网上发出了中国有史以来第一封电子邮件,“越过长城,走向世界”,从此,我国互联网时代开启。30 年间,人类社会仍然遵循着万物自然生长规律,但互联网的枝芽却依托人类的智慧于内部结构中野蛮扩展,并且每一次主流设备、主流技术的迭代速度明显加快。如今,人们的生活是“拇指在手机屏幕方寸间游走的距离,已经超过双脚走过的路程”。

据估计,截至 2017 年 6 月,中国网民规模已达到 7.5 亿人,占全球网民总数的五分之一,而且这个数字还在不断地增加。

然而,面对快速发展的互联网,每一个互联网人亦感到焦虑,感觉它运转的速度已经快到我们追赶的极限。信息时刻在更新,科技不断被颠覆,想象力也一直被挑战,面对这些,人们感到不安的同时又对未来的互联网充满期待。

互联网的魅力正在于此,恰如山之两面,一面阴暗晦涩,一面生机勃勃,一旦跨过山之巅峰,即是不一样的风景。就是这样的挑战会让人着迷,并甘愿为之付出努力。而这个行业还有很多伟大的事情值得去琢磨,去付出自己的匠心。

本系列丛书作为计算机科学与信息科学中的入门与提高教材,在力争保障学科知识广度的同时,也统筹主流技术的深度,既介绍了计算机学科相关主题的历史,也涵盖国内外最新、最热门课题,充分呈现了计算机科学技术的时效性、前沿性。丛书涉及计算机与信息科学多门课程:JAVA 程序设计与开发、C# 与 WinForm 程序设计、SQL Server 数据库、Oracle 大型数据库、Spring 框架应用开发、Android 手机 APP 开发、JDBC/Jsp/Servlet 系统开发,等等;HTML/CSS 前端数据展示、JQuery 前端框架、JAVAScript 页面交互效果实现,等等;大数据基础与应用、大数据技术概论、R 语言预测、PRESTO 技术内幕,等等;Photoshop 制作与视觉效果设计、网页 UI 美工设计、移动端 UI 视觉效果设计与运用、

CorelDraw 设计与创新,等等。

本系列丛书适合初学者,书中内容所涉及的知识点和相关信息应了解、掌握。开发人员可从本系列丛书中找到许多不同领域的兴趣点和各种知识点的用法。丛书实例内容选取市场流行的应用项目或产品项目,章后部分练习题模拟了大型软件开发企业实例项目。

本系列丛书在编写过程中,获得了国家自然科学基金委员会与中国民用航空局联合资助项目(U1733110)、湖南省科学“十三五”规划课题(XJK016BGD009)、湖南省教学改革研究课题(2015001)、湖南省自然科学基金(2017JJ1012)、国家自然科学基金(71371067)的资助,并得到了湖南大学、国防科技大学、电子科技大学、佛山科学技术学院、长沙学院和深圳华大乐业教育科技有限公司各位老师的大力支持,同时参考了一些相关著作和文献,在此向这些老师和文献作者深表感谢。

未来互联网信息技术已扑面而来,汹涌胜于往昔,你做好准备了吗?

作者

2017年9月

目次

理论部分

第 1 章 解析 XML	2
1.1 DOM 技术	2
1.2 DOM 编程	2
1.2.1 DOM 接口	3
1.2.2 使用 DOM 读取 XML 文件	5
1.3 SAX 技术	7
1.4 SAX 编程	7
1.4.1 SAX 常用接口	7
1.4.2 SAX 实现方法	11
1.4.3 使用 SAX 解析 XML	13
1.5 SAX 和 DOM 比较	16
1.6 选择 DOM 和 SAX	16
第 2 章 EL 表达式	18
2.1 EL 概述	18
2.1.1 EL 的基本语法	18
2.1.2 EL 的特点	19
2.2 与低版本的环境兼容——禁用 EL	19
2.2.1 使用斜杠“\”符号	19
2.2.2 使用 page 指令	19
2.2.3 在 web.xml 文件中配置 <el-ignored> 元素	20
2.3 保留关键字	20
2.4 EL 运算符及其优先级	20
2.4.1 通过 EL 访问数据	21
2.4.2 在 EL 中执行算术运算	22
2.4.3 在 EL 中判断对象是否为空	23
2.4.4 在 EL 中执行逻辑关系运算	24
2.4.5 在 EL 中执行条件运算	25
2.5 EL 的隐含对象	26
2.5.1 页面上下文对象	26

2.5.2	访问作用域范围的隐含对象	28
2.5.3	访问环境信息的隐含对象	29
第 3 章	自定义标签	33
3.1	标签库	33
3.1.1	标签的形式	34
3.1.2	Tag 接口	34
3.1.3	IterationTag 接口	35
3.1.4	BodyTag 接口	37
3.2	标签库描述符	38
3.2.1	<taglib>元素	39
3.2.2	<validator>元素	40
3.2.3	<listener>元素	40
3.2.4	<tag>元素	41
3.2.5	<tag-file>元素	43
3.2.6	<function>元素	44
3.3	简单标签的开发	45
3.3.1	SimpleTag 接口	46
3.3.2	实例: <welcome> 标签	46
第 4 章	JSP 标准标签	49
4.1	JSTL 介绍和编程举例	49
4.1.1	JSTL 介绍	49
4.1.2	JSTL 编程举例	50
4.2	一般用途的标签	51
4.2.1	<c:out>	52
4.2.2	<c:set>	54
4.2.3	<c:remove>	56
4.2.4	<c:catch>	58
4.3	条件标签	59
4.3.1	<c:if>	60
4.3.2	<c:choose>	60
4.3.3	<c:when>	61
4.3.4	<c:otherwise>	62
第 5 章	过滤器与监听器	64
5.1	过滤器	64
5.2	实现一个 Servlet 过滤器	65
5.2.1	编写实现类的程序	65
5.2.2	配置发布 Servlet 过滤器	66
5.3	ServletRequest 和 ServletResponse 的包装类	67

5.4 用 Servlet 过滤器过滤文本信息	68
5.4.1 输出流管理类	68
5.4.2 编写 Servlet 过滤器	69
5.4.3 编写 JSP 和 Servlet 文件	71
5.5 Servlet 监听器简介	76
5.5.1 监听服务器 ServletContext 对象	76
5.5.2 监听客户会话	77
5.5.3 监听客户请求	78
第 6 章 连接池与数据源	80
6.1 连接池	80
6.1.1 连接池原理	80
6.1.2 连接池关键问题分析	80
6.1.3 连接池的实现	82
6.2 数据源及 JNDI	84
6.2.1 什么是数据源	84
6.2.2 什么是 JNDI	84
6.2.3 为什么要用 JNDI	85
6.2.4 JNDI 的架构与实现	85
6.2.5 JNDI 工作原理	86
第 7 章 文件上传下载与图片水印	92
7.1 jspSmartUpload 组件简介	92
7.2 jspSmartUpload 组件的核心类	93
7.2.1 保存信息的 File 类	93
7.2.2 存储文件的 Files 类	94
7.2.3 提供表单信息的 Request 类	95
7.2.4 组件的核心 SmartUpload 类	95
7.2.5 情景应用:采用 jspSmartUpload 组件操作文件	98
7.3 图片水印和文字	101
第 8 章 Web 服务	105
8.1 Web 服务的产生背景和发展	105
8.1.1 问题的产生	105
8.1.2 传统的解决方案	105
8.1.3 Web 服务应运而生	106
8.2 Web 服务简介	106
8.2.1 什么是 Web 服务	106
8.2.2 Web 服务模型	107
8.3 体系结构概览	109
8.3.1 Web 服务协议栈	109

8.3.2	网络层	111
8.3.3	基于 XML 消息传递的分布式计算	112
8.4	Web 服务的创建和部署	113
8.4.1	创建 Web Service 工程	113
8.4.2	运行向导	114
8.4.3	创建服务方法类	115
8.4.4	部署项目	116
8.4.5	测试	116

上机部分

上机 1	解析 XML	120
第 1 阶段	指导	120
指导 1	制作人员信息表	120
指导 2	显示人员信息表	122
第 2 阶段	练习	124
练习 1	制作课程表	124
练习 2	显示课程信息	124
上机 2	EL 表达式	125
第 1 阶段	指导	125
指导 1	获取表单数据	125
指导 2	使用 JSTL+EL 循环打印消息	126
第 2 阶段	练习	127
练习 1	使用 EL 表达式实现注册消息确认	127
练习 2	输出不同范围内的值	128
上机 3	自定义标签	129
第 1 阶段	指导	129
指导	自定义一个迭代标签	129
第 2 阶段	练习	131
练习	编写一个 <max> 标签	131
上机 4	JSP 标准标签	132
第 1 阶段	指导	132
指导 1	猜字母游戏	132
指导 2	用户登录	133
第 2 阶段	练习	135
练习 1	用户注册	135
练习 2	给用户注册程序添加异常处理	135

上机 5 过滤器与事件监听	137
第 1 阶段 指导	137
指导 1 在线人数统计	137
指导 2 限定客户端 IP 地址	142
第 2 阶段 练习	145
练习 1 监听 session	145
练习 2 统一编码	145
上机 6 连接池与数据源	146
第 1 阶段 指导	146
指导 用连接池连接 SQL Server	146
第 2 阶段 练习	148
练习 测试 JNDI、JDBC、ODBC 的性能	148
上机 7 文件上传下载与图片水印	149
第 1 阶段 指导	149
指导 实现文件上传功能	149
第 2 阶段 练习	151
练习 给图片添加水印	151
上机 8 Web 服务 AOP	153
第一阶段 指导	153
指导 1 使用 AOP 实现登录日志记录功能	153
指导 2 使用异常通知	155
第二阶段 练习	156
练习 1 在删除部门时记录日志信息	156
练习 2 在删除员工时记录日志信息	156

附录部分

附录 1 Jetty	158
附录 1.1 Jetty 的基本组成	158
附录 1.1.1 Jetty 的基本架构	158
附录 1.1.2 Handler 的体系结构	159
附录 1.2 Jetty 的工作方式	160
附录 1.2.1 Jetty 的启动过程	160
附录 1.2.2 接受请求	160
附录 1.2.3 基于 HTTP 协议工作	161
附录 1.2.4 基于 AJP 工作	162
附录 1.2.5 基于 NIO 方式工作	164

附录 1.2.6 处理请求	164
附录 1.3 与其他服务器集成	166
附录 1.3.1 与 Jboss 集成	166
附录 1.3.2 与 Tomcat 的比较	166
附录 1.4 性能与总结	167
附录 1.4.1 性能比较	167
附录 1.4.2 特性比较	168
附录 1.4.3 总结	168
附录 2 Apache 服务器	169
附录 2.1 Apache 概述	169
附录 2.2 Apache 服务器的安装	169
附录 2.3 Apache 服务器的基本配置	170
附录 2.3.1 常规配置指令	170
附录 2.3.2 性能配置指令	172
附录 2.3.3 日志配置指令	173
附录 2.3.4 容器与访问控制指令	174
附录 2.3.5 其他配置指令	177
附录 2.4 配置虚拟主机	179
附录 2.4.1 基于主机名的虚拟主机	179
附录 2.4.2 基于 IP 地址的虚拟主机	183
附录 2.5 Apache 服务器的启停与测试	185
附录 2.5.1 启动 Apache 服务器	185
附录 2.5.2 重启 Apache 服务器	185
附录 2.5.3 停止 Apache 服务器	185
附录 2.5.4 开机时自动启动 Apache 服务器	185
附录 2.5.5 测试 Apache 服务器	186
附录 2.6 图形化配置 Apache 服务器	186
附录 2.6.1 “主”选项卡	186
附录 2.6.2 “虚拟主机”选项卡	187
附录 2.6.3 “服务器”选项卡	192
附录 2.6.4 “调整性能”选项卡	193
附录 3 Web 开发框架分析	194
附录 3.1 Web 开发框架介绍	194
附录 3.2 Tapestry	194
附录 3.3 WebWork	196
附录 3.4 其他开发框架	197
附录 3.5 Web 开发框架总结	197
参考文献	198

理 论 部 分

第 1 章 解析 XML

学习目标

- 了解 DOM 并应用 DOM 编程
- 了解 SAX 并应用 SAX 编程
- 了解 SAX 和 DOM 的区别并选择

为了更好地处理 XML 文档解析问题, W3C 提供了 DOM 模型的推荐标准, DOM 模型将 XML 结构化文档映射为一系列具有父子、兄弟关联关系的节点对象集, 从而允许应用程序调用这些节点对象的方法、属性来读取或改变 XML 节点的内容, 从而可以动态地读取或改变 XML 文档的内容。除此之外, XML 行业还有另一个 XML 解析的事实规范——SAX, 它采用一种基于事件处理的方式来解析 XML 文档。SAX 解析器在处理一份 XML 文档时, 会在 XML 文档中依次触发文档开始、元素开始、元素结束、文档结束等事件, 通过监听 SAX 解析过程中所触发的事件即可获取 XML 文档的内容。

1.1 DOM 技术

DOM(Document Object Model, 文档对象模型)早在 XML 之前已经存在, Microsoft Explorer 中就嵌入了 DOM 模型。在应用程序中基于 DOM 的 XML 分析器将一个 XML 文档转换为一个对象模型的集合(通常称为“DOM 树”), 来实现对 XML 文档数据的操作。通过 DOM 接口, 应用程序可以在任何时候访问 XML 文档中的任何一部分数据, 因此这种利用 DOM 接口的机制也被称为“随机访问机制”。

对于 XML 应用开发来说, DOM 就是一个对象化的 XML 数据接口, 即一个与语言和平台无关的标准接口规范。DOM 接口提供了一种通过分层对象模型来访问 XML 文档信息的方式, 这些分层对象模型依据 XML 的文档结构形成了一颗节点数。无论 XML 文档中所描述的是什么类型的信息, 即便是制表数据、项目列表或一个文档, 利用 DOM 所生成的模型都是节点树的形式。即 DOM 强制使用树模型来访问 XML 文档中的信息, 由于 XML 本质上就是一种分层结构, 所以这种描述方法是相当有效的。

1.2 DOM 编程

DOM 是与平台和语言无关的表示 XML 文档的 W3C 官方标准, 是以层次结构组织的节点或信息片段的集合, 允许在其中寻找特定信息。使用 DOM 解析 XML 文档结构需

要加载整个 XML 文档,然后在内存中生成 DOM 树及树上的每个 Node 对象。由于它是基于信息层次的,因而 DOM 被认为是基于树或基于对象的,其优点如下:

(1)DOM 树在内存中是持久的,应用程序能够修改它以更改数据结构。

(2)任何时候都可以在 DOM 树中导航,而不是像 SAX(Simple API for XML,XML 简单应用程序接口)那样一次性处理。

(3)DOM 使用起来要简单得多,不用花费大量编程时间在 DOM 应用上。

DOM 的不足是如果使用 DOM 操作特别大的 XML 文档,解析和加载整个文档速度会变慢,甚至会耗尽内存资源。使用对于这种 XML 文档应使用其他方法来处理,如基于事件模型的 SAX。

1.2.1 DOM 接口

作为 W3C 的标准接口规范,DOM 由 3 个部分组成,即核心(core)、HTML 和 XML。核心部分是结构化文档比较底层对象的集合,定义对象已经完全可以表达任何 HTML 和 XML 文档中的数据;HTML 和 XML 接口两个部分则是专为操作具体的 HTML 和 XML 文档提供的高级接口,使处理这两类文档更加方便。

DOM 接口规范中有 4 个基本接口。

(1)Document 接口。

Document 接口从 Node 接口继承而来,是操作文档的入口,即整个文档的根节点。

该接口的常用属性有 DocumentType 类型的 Doctype 属性和 Element 类型的 documentElement 属性,它们分别用于记录文档的类型和访问文档的根元素。

该接口的常用方法如表 1-1 所示。

表 1-1 Document 接口的常用方法

方法名	返回类型	描述
createAttribute(String)	Attr	创建指定名称的属性,可以将其定义为其他元素的属性
createCDATASection(String)	CDATASection	创建 CDATASession 节点
createComment(String data)	Comment	创建注释节点
createElement(String)	Element	创建指定名称的元素
createEntityReference(String)	EntityReference	创建实体节点
createProcessingInstruction (String,String)	ProcessingInstruction	创建有指定名称和数据处理指令的节点
createTextNode(String)	Text	创建给定文本内容的文本节点
GetDocumentElement()	Element	返回文档的根节点
GetElementById(String)	Element	返回指定 ID 的节点,找不到则返回 null
getElementsByTagName(String)	NodeList	返回指定名称的所有节点的集合,集合中的所有元素自动按顺序排列

(2)Node 接口。

Node 接口是 Document、Element、Attribute、Text 及 Comment 等接口的父类,它是 DOM 树的基本数据类型,可以包含或不包含子节点。

该接口常用的方法如表 1-2 所示。

表 1-2 Node 接口的常用方法

方法名	返回类型	描述
appendChild(Node)	Node	在节点中添加子节点
cloneNode(boolean)	Node	复制当前节点,复制产生的节点没有父节点,包括复制该元素的所有属性和属性值
getAttributes()	NamedNodeMap	获取节点中的所有属性
getChildNodes()	NodeList	获取该节点中的所有子节点
getFirstChild()	Node	获取该节点中的第 1 个子节点
getLastChild()	Node	获取该节点中的最后一个子节点
getNextSibling()	Node	获取该节点的下一个节点
getNodeName()	String	获取该节点的名称
getNodeType()	Short	获取该节点的类型
getNodeValue()	String	获取该节点的值
getParentNode()	Node	获取该节点的父节点
getPreviousSibling()	Node	获取该节点的上一个节点
getTextContent()	String	获取该节点的文本内容
hasAttributes()	boolean	判断节点是否有属性
hasChildNodes()	boolean	判断节点是否有子节点
insertBefore(Node1,Node2)	Node	将节点 Node1 插入到当前节点的子节点 Node2 之前
removeChild(Node)	Node	从当前节点删除指定的子节点并返回该子节点
replaceChild(Node1,Node2)	Node	将当前节点的 Node2 子节点替换为 Node1 节点,并返回 Node1 节点

(3)NodeList 接口。

NodeList 接口是一个节点的集合,其中包含某个节点中的所有子节点,该集合中的所有项目都可以通过列表中的索引处理。NodeList 接口的常用方法如表 1-3 所示。

表 1-3 Node 接口的常用方法

方法名	返回类型	描述
getLength()	Int	获取列表中节点数量
item(int)	Node	获取相应索引的节点对象

这里需要注意的是:

- 如果 item()方法的参数超出索引范围,则返回 null。

• NodeList 对象是动态的,当文档内容改变时会自动更新该对象,而不用 DOM 应用程序重新获取它。

(4)NamedNodeMap 接口。

NamedNodeMap 接口也是一个节点的集合,通过该接口可以像 Map 集合一样使用 key/value 键值对建立节点和节点之间的映射关系,从而通过节点名直接访问特定的节点;另外在这个集合中,节点的排列是没有顺序的。

1.2.2 使用 DOM 读取 XML 文件

使用 DOM 操作 XML 需要创建 DocumentBuilderFactory 工厂,通过它来生成 DocumentBuilder 的实例,该实例用于解析 XML 到 Document 对象中。

定义获取 XML 解析器的工厂对象的语法格式如下:

```
DocumentBuilderFactory dbc=DocumentBuilderFactory.newInstance();
```

创建 DocumentBuilder 对象解析 XML 文件的语法格式如下:

```
DocumentBuilder db=dbf.newDocumentBuilder();
```

从 XML 文件中解析 Document 对象的语法格式如下:

```
Document doc=db.parse(String path);
```

其中参数 path 是 XML 文件的路径信息。

从 XML 的标签名中获得所有属性值的语法格式如下:

```
doc.getElementsByTagName(String tag);
```

- tag:在 XML 文件中定义的标签信息,如<id></id>。
- doc:Document 类的实例对象。

XML 和 JSP 是创建 Web 应用的重要工具,其中 XML 技术是现在最流行的标记语言,它有信息存储的功能,并且是由通用的 XML 格式规范文件的格式。本实例应用 XML 技术存储简单的图书信息,然后读取到 JSP 页面中。

例 1.1 读取 XML 文件。

代码如下:

```
<? xml version="1.0" encoding="UTF-8" ? >
<books>
<book>
<id>324523452345</id>
<name>JavaWeb 高级程序设计</name>
<public>华瑞集团出版社</public>
<price>89</price>
</book>
<book>
<id>567554434567</id>
```