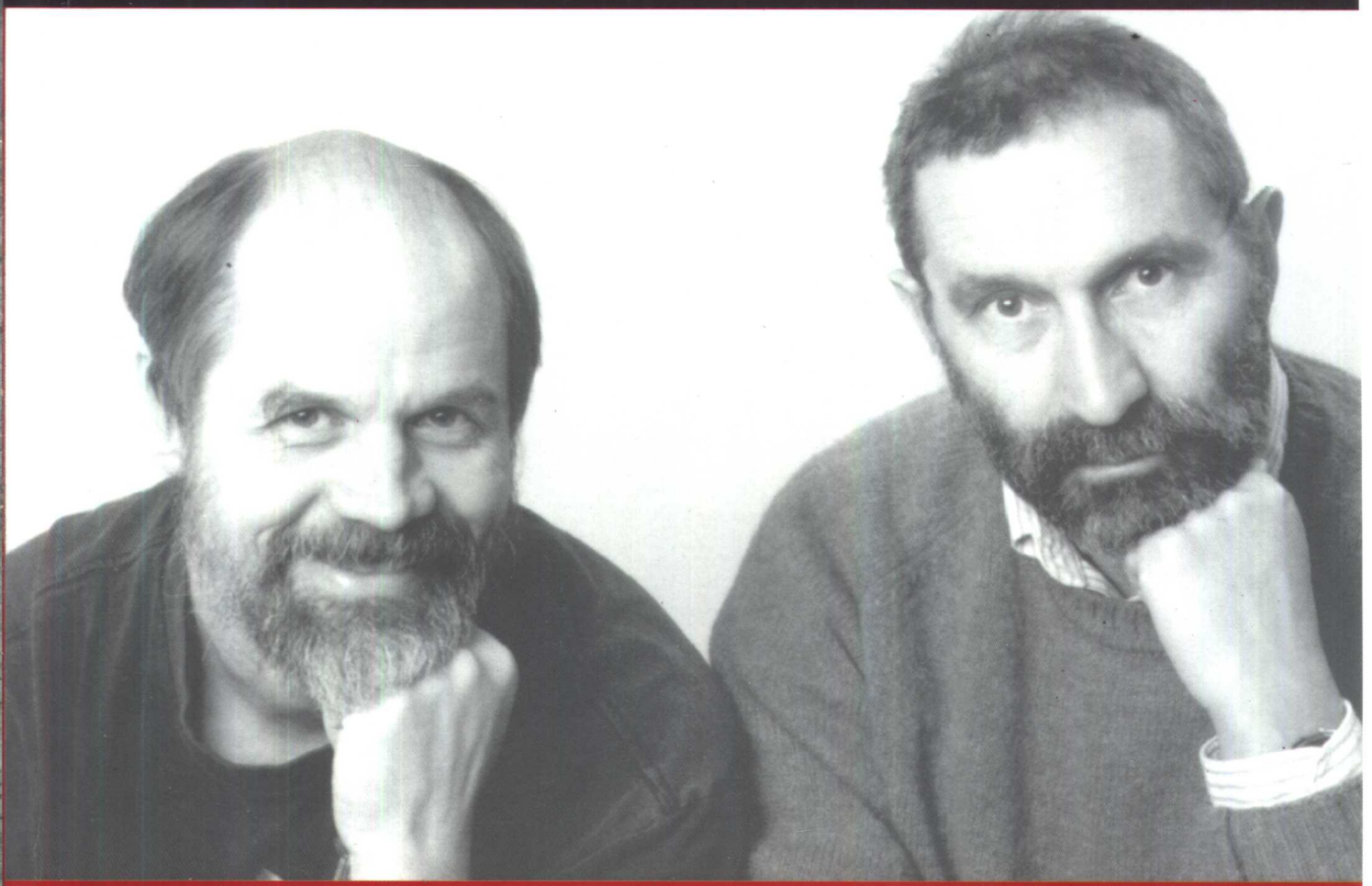
Professional Java XML Programming with **Servlets and JSP**



Java XML 编程指南

〔美〕 Tom Myers
Alexander Nakhimovsky 著
王 辉 张晓晖 等译

wrox 乐思
PROGRAMMER TO PROGRAMMER™

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: http://www.phei.com.cn

Professional Java XML Programming with Servlets and JSP

Java XML编程指南

〔美〕 Tom Myers
Alexander Nakhimovsky 著

王辉 张晓晖 等译

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 提 要

分布式Web应用程序是如今最常见、也最值得投入的一种应用程序，建立Web应用程序的最佳方式是使之成为三层应用程序，从而巧妙地区分出其三个组成部分：用户界面、计算逻辑与数据存储，而Java与XML的组合提供了建立三层应用程序的最佳手段。基于以上思想，本书作者面向有经验的程序员和计算机专业的学生，通过开发多个实质性的应用程序，介绍了大量与Java、XML、JSP以及XSLT有关的技术；同时，以元编程方式——编写定制、指导与修改其他程序的程序，在用户、程序员与程序之间建立一种新型关系。

本书探讨的复合技术既可以分开应用，也可以针对不同系统类型以不同方式结合使用。相信读者一定会倍感受益。

Copyright©2001 Wrox Press. All rights reserved. No part of this book may be reproduced, stored in a retrieval system or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical articles or reviews.

本书英文版由Wrox公司出版，Wrox公司已将中文版独家版权授予电子工业出版社及北京美迪亚电子信息有限公司。未经许可，不得以任何形式和手段复制或抄袭本书内容。

图书在版编目（CIP）数据

Java XML编程指南/（美）梅尔斯（Myers，T.）著；王辉译. - 北京：电子工业出版社，2001.4

书名原文：Professional Java XML Programming

ISBN 7-5053-6619-X

Ⅰ. J… Ⅱ. ①梅… ②王… Ⅲ. JAVA语言 - 程序设计 Ⅳ. TP312

中国版本图书馆CIP数据核字（2001）第21072号

书　　名：**Java XML编程指南**

著　　者：〔美〕Tom Myers　Alexander Nakhimovsky

译　　者：王　辉　张晓晖　等

责任编辑：李　莹

印 刷 者：北京天竺颖华印刷厂

装 订 者：三河金马印装有限公司

出版发行：电子工业出版社　URL:http://www.phei.com.cn

北京市海淀区万寿路173信箱　邮编：100036　电话：68279077

北京市海淀区翠微东里甲2号　邮编：100036　电话：68252397

经　　销：各地新华书店

开　　本：787×1092 1/16　印张：44.25　字数：1130千字

版　　次：2001年4月第1版　2001年4月第1次印刷

书　　号：ISBN 7-5053-6619-X / TP·3676

定　　价：72.00元

版权贸易合同登记号　图字：01-2000-2973

凡购买电子工业出版社的图书，如有缺页、倒页、脱页请向购买书店调换，若书店售缺，请与本社发行部联系调换。

简　介

本书的内容

简而言之，本书主要介绍三层Web应用程序的开发，其中包括使用Java编写应用程序，使用XML语言配置应用程序，在其组件之间交换数据，以及为其HTML输出提供定制模板。

继续阅读下面的内容，会看到有关主题的更多解释。但在发现其独特魅力之前，不妨看看下面我们对本书主题的认识。

- 分布式Web应用程序是今天最常见、也最值得投入的一种应用程序。
- 建立Web应用程序的最佳方式是使它成为三层应用程序，因为这样可以巧妙地区分出应用程序的三个组成部分：用户界面、计算逻辑与数据存储。
- Java与XML的组合提供了建立三层应用程序的最佳手段，为此，我们引用David Brownell曾说过的一段话来解释："Java是跨平台的代码，而XML是跨平台的数据"。实际上，XML还不仅如此，稍后我们会介绍它的性能。

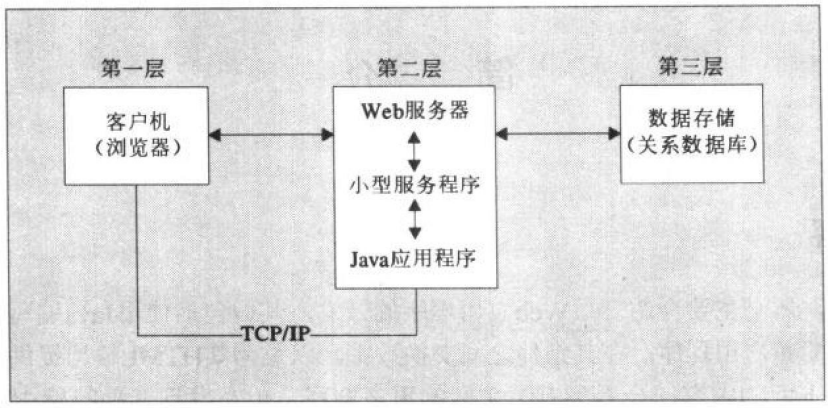
三层Web应用程序

通过将应用程序划分为三层结构，我们可以区分出应用程序的三个逻辑组件：用户界面、计算逻辑与数据存储。然后针对每个逻辑单元独立进行开发，从而极大地增强应用程序设计的灵活性。例如，通过将计算逻辑与用户界面分隔开，来开发各种各样的用户界面。这样不仅允许不同类型的用户访问同一计算层（应用程序的"工作单元"），也可以根据每一类用户的特定需要定制各自的界面。数据存储层可以由任何结构完善的数据源构成，例如数据库或XML文档，它允许在对数据存储方式完全改变的情况下，不影响应用程序的计算逻辑或用户界面。

从三层结构中获益最大的是中间层，因为三层结构允许计算逻辑的开发达到任意复杂的程度，它自己也可以包含多层。另外，三层结构还允许任意改变计算逻辑，且不会影响到用户与之交互的方式。通过将应用程序逻辑置于一台单机（每个用户都必须访问）上，可以保证对应用程序软件的任何更新都会自动实施于所有用户，避免维护同一应用程序多种版本的烦恼。

事实上，三层Web应用程序通常包含一个作为用户界面的Web浏览器、一个与"中间层"应用程序相连的Web服务器、一个持久稳定的存储器（通常是一种关系型数据库）。在本书中，Java是中间层的语言。前面已经讲过，中间层可以非常复杂，而且在设计时可以进一步划分它，将应用程序分成四层、五层或N层。为了简单起见，我们主要讨论三层结构，也可能会提到中间层的进一步划分。

下面的示意图说明了三层Web应用程序的概念：



XML

Extensible Markup Language或XML实际上并不是一种语言，而是定义标记语言的一种元语言。例如，它可以用于定义XHTML，为HTML标明后继符。最初，开发XML的宗旨是用其标记文档的特定内容。现在，它显然还可以作为了一种优秀的数据描述工具，因为它包容的数据很广泛，甚至包括控制流程或计算序列。在本书中，我们主要使用XML来配置应用程序，并从一个组件向另一组件传送数据。我们不准备直接在浏览器中显示XML文件，而是提供其HTML替代文件。

本书的独特之处

本书独立于浏览器并且基于标准。它还有一个主题：设法明确Java与XML的一种组合方式，并使其可以转变Web应用程序编写的惯例。

独立于浏览器

我们非常保守地使用了JavaScript，并采取了独立于浏览器的方式。我们虽然使用了Cascading Style Sheets（CSSI），使我们的应用程序能够被一种浏览器所理解，但不同的是，没有浏览器的限制。特别是，本书中的所有应用程序都可以通过Internet Explorer 4与Netscape Communicator 4，以及第五代浏览器进行使用。

基于标准

与以前很多关于XML及相关技术的书籍不同，本书只使用它们的稳定版本：非Working Drafts或Beta测试版。这样保证了本书及其代码在长时间内（只要Sun与W3C保持它们产品版本的向后兼容性）都稳定适用。特别是，我们只使用W3C的建议书或Proposed建议书，如：

试读结束，需要全本PDF请购买 www.ertongbook.com

- XML Path Language（XPath）Version 1.0
- eXtensible Stylesheet Language for Transformations（XSLT）Version 1.0
- XHTML 1.0: The Extensible HyperText Markup Language - A Reformulation of HTML 4.0 in XML
- 与XML文档相关的Style Sheets
- XML中的Namespaces
- Document Object Model（DOM）Level 1
- XML 1.0

我们惟一加以讨论的Working Draft版本（但没有在代码中使用）是XHTML 1.1——基于模块的XHTML。尽管其规格细节仍不固定，但模块化的原理与机制已经稳定了一段时间，而且不太可能发生变量了。无疑，未来的Web文档将使用模块化的版本。

对于Document Type Definitions（DTD）的范例，我们使用了诸如DocBook、XHTML与WML（Wireless Markup Language）这样的行业标准。另外在我们的代码中使用了JavaServer Pages（JSP），而且只使用了其稳定的1.0版本。

本书的主题

本书开发了多个实质性的应用程序，涉及大量与Java、XML、JPS及XSLT有关的技术。本书的主题是元编程，也就是编写定制、指导与修改其他程序的程序。元编程并不是很新的思想（LISP程序员已经遵循它数十年了），但XML却赋予了它全新的理念。在该简介中，我们只能预先对它们的支持提出大胆的主张，但我们相信XML会大大增加组件用户（而不是程序员）通过编辑可读性文本文件，控制计算机程序的能力。简而言之，就是用.xml文件与一种XML编辑器代替.ini文件与Notepad，极大地增加通过文本文件控制程序的可能性，或许引入一种新的编程方式，在用户、程序员与程序之间建立一种新型关系。本书的目标就是容入这一新的可能性，以帮助建立这种新型关系。

本书的目标也可以说是开发通用的程序，使之尽可能少地了解自己实际将承担的任务。而它们的结构与性能（模式、视图、与控制）在指定域的XML语言中用相应的DTD来描述，并且由程序自己来“解释”这些描述。熟悉设计模式文献（见Gamma等编著的《Design Patterns》，由Addison-Wesley 1995年出版）的读者会认识到本书受到了“Little Language”模式（我们称之为Minilanguage，微型语言）的影响，并大量使用了该模式。

语言与分析程序

为了在程序组织（不是显示）中使用XML，我们必须在内存的数据结构中分析XML文档。我们会介绍如何安装并使用一种分析程序，但对技术的深入了解对于有效使用XML才是最重要的。因此，我们在介绍XML并进行分析之前，设计出了一种微型语言，并为它编写了一个分析程序。该语言并不只是为了教学使用：它是一种为HTML输出编写模板文件的语言，而且给出了基于模板/微型语言方法与使用JSP生成HTML方法间的比较。结论是，两种

方法根据具体情况都有自己的优点，但在Web开发人员（组件用户，不是程序员）不能获得JSP工具开发人员持续支持的情况下，微型语言方法具有一定的优势。

内容概述

本书的很多内容都相互依赖，所以如果读者认识了本书内容的整体设计思想，在阅读时就会事半功倍。另外，本书内容分为几大块，每一块都由两章组成，简单地从章节序号上并不能严格区分这些内容。

第1章和第2章是准备部分。它们通过逐步建立一些非常简单而且未加组织的范例，来介绍Java Web应用程序－servlets（用Java编写）与JDBC（Java DataBase Connectivity）的基本结构。通过这些简单范例的描述，我们会逐渐引出中间层的组件：作为调度程序的主服务程序，数据库连接部分，与HTML生成部分。如果读者已经具有了三层结构的开发经验，可能会发现这些简单的范例违背了你已掌握的知识，因为它们不是线程安全的，或它们没有使用连接查询等等。然而，这为后来的开发（届时会解决这些问题）做好了准备。

第3章和第4章仍与Java有关，且还看不到与XML有关的迹象。在此我们要推出第一个实质性的应用程序，三层应用程序的一个定制化“外壳”。作为该外壳的一部分，我们将在第4章开发一种微型语言，用于编写输出模板文件与初始化/配置文件。该章的目的有两个：一个明显目的，一个教学目的。明显目的是要开发一种微型语言及其分析程序。尽管对于输出模板已经存在其他一些微型语言（在该章中我们会提到），但我们的微型语言有自己的优点，而且我们对重要的开发任务使用了其XML化的版本。教学目的是为读者学习XML技术而做准备。要真正理解与有效使用XML，理解正式的语法与分析程序是非常重要的。而且这对于熟练阅读EBNF（Extended Backus-Naur Form）编写的语法规则（或“产品”）也很有用，因为这些产品要将大量信息压缩到很小的空间里。

第5章和第6章是关于XML的（包括Namespaces），而且根本不涉及Java。在第7章和第8章又返回到Java（但不针对三层应用程序！），在这两章中，我们将介绍XML分析及DOM与SAX规范之间的关系。这一部分导致我们在第9章中为开发XML微型语言而建立起一个通用的框架：实现一种通用的机制，向XML语言添加解释方法，并使用它编写从XML文件到数据库表的通用转换程序。

在第10章中，我们返回三层应用程序，且带来新的转变：我们开始使用JSP页替代小型服务程序。JSP是一种令人称奇的技术，在讨论微型语言、输出模板、小型服务程序与中间层Java应用程序的书中不涉及JSP是不可能的，因为它们可以实现所有这些任务。但这也是它们最大的弱点：正因为功能太强、太灵活，所以很容易导致程序员误入歧途，编写大量杂乱的代码。我们还要花费大量时间来考虑相关的结构问题：如何将JSP与JavaBean最优化地组合成结构完善的应用程序。

第11章和第12章开发出本书到目前为止最长也最雄心勃勃的应用程序，兑现了我们建立一个通用应用程序的许诺，该程序由一个XML配置文件（通过HTML窗体中的用户输入来定制）动态配置。该应用程序是一个邮件客户机，可以用于发送与接收邮件，并将邮件信息存储到一个关系型数据库中。它的功能可能不如主流商业软件那么强，但它解释了一些功能

更强的技术。例如，它在启动时根据XML中定义的JavaBean组件配置自己；一旦配置成功，组件会生成并使用XML数据，根本不考虑数据是来自文件、字符串变量、套接字、还是数据库。换句话说，它不仅是一个专用的应用程序，还可以作为开发分布式应用程序的通用框架。

最后在第13章，我们介绍本书最后一项技术，XSLT（包括XPath）。XSLT是一种用于转换XML文档的语言。对于XML转换来说，它并不是一种完全通用的机制——有些问题只能借助于分析程序——但XSLT以一种说明方式，通过相对较少的代码，可以完美地完成很多任务。而且，XSLT可以包括用Java编写的扩展函数。虽然扩展机制还没有标准化，但这是必然趋势，而且届时在XSLT与JSP之间会出现实质性的交迭。XSLT是一种必须熟悉与加以关注的工具。

如果我们有更多的时间与篇幅，还会开发另一个应用程序。这是名为N-topus、一种像章鱼一样的工具，具有N条腿与K个头，K<=N。而且每个头都将成为应用程序或数据转换程序的入口点，每条腿都用于将XML编码的数据从一个转换头传送给另一个头。一些转换程序将使用XSLT，另一些会使用XML分析程序与定制设计的JavaBean。整个结构将由XML文件动态配置，该文件由组件用户（不是确认分析程序监视下的程序员）生成。如果本书有第二版的话，我们将编写这样一个应用程序，希望如此。

本书面向的读者

本书适于有经验的程序员与计算机专业的学生。我们假设本书的读者已经熟悉了Java编程的基础知识，特别是已经阅读过了Sun的Java Tutorial中的下列基本材料：

- 学习Java语言
- 基本的Java类

这些材料见http://java.sun.com/books/tutorial/trailmap.html。有关材料还可以参考Ivor Horton的《Beginning Java 2》（Wrox出版社，1999, ISBN 1-861002-23-8）一书。

我们还假设读者已经熟悉了下列基础知识：

- HTML、CSSI与JavaScript
- HTTP协议与CGI编程
- 关系型数据库
- SQL语法与词汇

我们的HTML与JavaScript设计得很简单，这样可以使读者的注意力集中在本书的主题上，而不是停留在目前所有浏览器的功能上。HTTP、CGI、关系型数据库与SQL在书后附录中都有概述，另外还提供了更广泛的参考意见。

本书需要的软硬件环境

分布式Web应用程序，特别是使用Java与XML的应用程序，需要的软件支持包括：一个Web浏览器、一个装备有小型服务程序引擎的Web服务器、一个具有JDBC匹配驱动程序的

关系型数据库、一个XML分析程序及其XSL伙伴。当然还有JDK 1.2。令人惊奇的是，整个架构可以安装在一台Win9x机器上：大多数开发任务是在一台Windows 98 Pentium II计算机上完成的。对于Web服务器与小型服务程序，我们使用了两个配置：具有JRun2.3 servlet引擎的Microsoft的Personal Web Server与Sun的jswdk-1.0。对于数据存储，我们使用了Microsoft的Access数据库与JDBC-ODBC桥接器。对于XML与XSLT，我们使用了Sun的XML分析程序与James Clark的XT（1999年11月5日颁布）。读者还需要安装我们的支持代码（从Wrox站点上下载比输入更容易），一个实用程序库（在本书中会陆续介绍）。

版式约定

在本书的文本与排版上，我们使用了一些与众不同的样式，以帮助读者区分不同类型的信息。下面是所使用的样式范例以及它们含义的解释：

建议、提示、与背景信息用这种字体。

重要信息像这样用黑体字单独列出。

重点术语用粗体表示。

屏幕菜单中出现的词，比如File或Window，用屏幕上的相同字体表示。

代码也会用一些不同的格式来表示。如果代码出现在文中——例如，我们讨论doPost()方法时——它用一种固定宽度的字体表示。程序代码也是如此，譬如：

```
public void doPost (HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
{
    doGet(req,res);
}
```

有时，读者还会看到代码用下面这种混合格式表示：

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE collection [
<!ELEMENT collection (tagpair*, overview?, list)>

<!ELEMENT tagpair EMPTY>
<!ATTLIST tagpair
  xtag CDATA #REQUIRED
  htag CDATA #REQUIRED
>
<!ELEMENT overview (#PCDATA)>

<!ELEMENT list (item+)>
<!ELEMENT item (type, title, description)>
<!ELEMENT type (#PCDATA)>
```

```
<!ELEMENT title (#PCDATA)>
<!ELEMENT description (#PCDATA)>
]>
```

仅在此类情况下，未加黑的代码是已经介绍过的、不需要再重复讨论的代码。

设计这些格式可以保证读者确切地知道正在阅读的内容。希望它们能简化读者的学习过程。

将你的意见告诉我们

对于这本书，我们下了很大工夫希望它尽量有用并适合读者。能受到读者青睐与认可，就是对我们最大的奖赏。为此，我们愿意努力与读者沟通并满足你们的期望。

请将你的意见告诉我们，讲述一下你最喜欢的和怎样我们能做得更好。如果你认为这仅是一番市场炒做之辞，那么请检验我们并给我们写信。我们将回答你，并把你所说的采纳在以后的版本中。最容易的方法是使用电子信箱：

feedback@wrox.com

在Wrox Press网站你也许会发现更多细节。在那里，你能从我们最新的图书上找到代码，预见即将出版的标题和作者编辑的信息。从网站上你可以直接订购Wrox书籍或找出离你最近的当地书店。Wrox Press网站的地址如下：

http://www.wrox.com

作 者 简 介

Alexander Nakhimovsky

Alexander Nakhimovsky于1972年获Lenigrad大学数学硕士学位，1978年获Cornell大学大众语言学博士学位，并辅修了计算机学。在1985年进入Colgate大学的计算机系之前，他曾于Cornell与SUNY Oswego讲授大众与斯拉夫语言学。他出版了一部关于理论与计算语言学的专著和多篇文章，多部俄语教科书，一本Nabokov's Lolita字典，并与Tom Myers合作出版了“Javascript Objects”，Wrox 1998，以及“Professional Java XML Programming with Servlets and JSP”，Wrox 1999。

Tom Myers

Tom于1975年获St.John's学院（Santa Fe, New Mexico）学士学位，1980年获Pennsylvania大学计算机学博士学位。他在专职从事软件开发与咨询之前，曾于Delaware与Colgate大学教授计算机学。他是“Equations, Models, and Programs: A Mathematical Introduction to Computer Science” Prentice-Hall Software Series，1988的作者，并撰写了多篇关于理论计算机学方面的论文。他还与Alexander Nakhimovsky合作出版了“Javascript Objects”，Wrox 1998，以及“Professional Java XML Programming with Servlets and JSP”，Wrox 1999。

试读结束，需要全本PDF请购买 www.ertongbook.com

致　　谢

由衷地感谢对本书的出版做出过贡献的人们。

首先是Wrox公司称职的编辑们：Tim Briggs, Jeremy Beacock, Gregory Beekman以及Paul Cooper。是他们的努力，避免了许多谬误；是他们的建议，提供了诸多帮助。David Brownell也值得言谢，他曾在许多场合帮助过我们。

还有Colgate大学信息技术服务部的人员，特别是Ross Miller, Jim Nesbitt, Bill Howell, 以及SBIO的其他善良人士。感谢Dylan Strong，感谢计算机科学系的秘书Charlotte Jablonski，他们的工作总是那么高效。

本书中的一些论点源于1999年春季的独立研究课题。参与该课题的学生们利用了一学期的时间紧张工作，在此，一并感谢小组成员Hui Cheng, Alan Lewis, Sameer Panjwani和Jon Seidman; Dave Blank和Alison Hartwell; Karthik Jayaraman（Java老师）；以及Matt Seeve和Chris Towt。尽管Colgate 99级的Armando Singer并非该组成员，但他同样给予了极大帮助，并显示出卓越才华。

同样，感谢我们的家人，没有他们的支持，就没有我们今天的成功。

目　录

试读结束，需要全本PDF请购买 www.ertongbook.com