



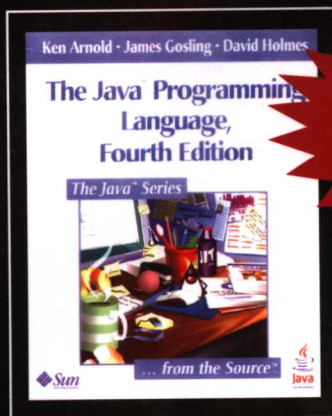
图灵原版计算机科学系列



Java™ Programming Language Fourth Edition

Java™ 程序设计语言 (英文版 · 第4版)

Ken Arnold
James Gosling 著
David Holmes



Java 之父
权威著作!



人民邮电出版社
POSTS & TELECOM PRESS



TURING

图灵原版计算机科学系列

Java™ Programming Language

Fourth Edition

Java™ 程序设计语言

(英文版 ·

江苏工业学院图书馆
第4版
藏书章

Ken Arnold

James Gosling 著

David Holmes



人民邮电出版社

POSTS & TELECOM PRESS

图书在版编目 (CIP) 数据

Java™ 程序设计语言: 第 4 版/ (美) 阿诺德, (美) 高斯林 (美) 霍姆斯著.

—北京: 人民邮电出版社, 2006.5

(图灵原版计算机科学系列)

ISBN 7-115-14762-0

I. J... II. ①阿...②高... III. JAVA 语言—程序设计—教材—英文 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 043586 号

内 容 提 要

本书主要讲授Java程序设计语言, 系统地介绍Java的主要包 (即`java.lang.*`、`java.util`和`java.io`) 中的大多数类, 并以内容丰富的示例对这些类如何工作进行了深入剖析。作者对这部经典著作进行更新, 使其反映了Java 2标准版5.0 (J2SE 5.0) 中主要的增强。本版中增加了一些新的章节, 专门讨论泛型、枚举、注解、断言和正则表达式等J2SE 5.0中引入的强大语言新特性。此外, 本书还强调了当今构建健壮、高效和可维护的Java软件的最佳实践。

本书对所有程序员 (包括那些经验丰富的程序员在内) 都是不可或缺的参考书。

图灵原版计算机科学系列

Java™ 程序设计语言 (英文版·第 4 版)

◆ 著 Ken Arnold James Gosling David Holmes
责任编辑 杨海玲

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

◆ 开本: 800×1000 1/16
印张: 57.5
字数: 1 100 千字 2006 年 5 月第 1 版
印数: 1—3 000 册 2006 年 5 月北京第 1 次印刷
著作权合同登记号 图字: 01-2006-1763 号

ISBN 7-115-14762-0/TP · 5390

定价: 99.00 元

读者服务热线: (010)88593802 印装质量热线: (010)67129223

版 权 声 明

Original edition, entitled *Java Programming Language, Fourth Edition*, 0321349806 by Ken Arnold, James Gosling, and David Holmes, published by Pearson Education, Inc., publishing as Addison-Wesley, Copyright © 2006 by Sun Microsystems, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

China edition published by POSTS & TELECOM PRESS Copyright © 2006.

This edition is manufactured in the People's Republic of China, and is authorized for sale only in the People's Republic of China excluding Hong Kong, Macao and Taiwan.

本书英文版由Pearson Education Asia Ltd.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

仅限于中华人民共和国境内（香港、澳门特别行政区和台湾地区除外）销售发行。

本书封面贴有Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

版权所有，侵权必究。

This book is dedicated to the Java team
From whose hard work and vision
A mighty oak has grown

To Susan—*K.A.*

To Judy and Kate—*J.A.G.*

To Lee, Taylor, and Caitlin —*D.H.*

前 言

美丽的建筑已经超出了科学的范畴。它们是真正的有机体，是心灵的产物；它们是艺术品，使用了由灵感激发出的最好的技术，而绝不是出自个人的癖好或委员会式的平庸想法。

——弗兰克·劳埃德·赖特¹

Java程序设计语言已经在世界范围内被软件开发者和因特网内容提供商热情接纳。因特网和万维网的用户因能够访问来自因特网上任何地方的、独立于平台的安全应用而受益匪浅。用Java程序设计语言创建应用的开发者也大受裨益，因为他们只需开发一次代码，而用不着再将应用“移植”到各种软硬件平台中去。

Java语言最初只是作为一种为万维网创建applet的工具而闻名于世的。applet是运行在Web页面中的一种小应用。下载之后，它可以在不使用Web服务器资源的情况下，在用户的浏览器页面中执行任务并与用户进行交互。当然，某些applet也可以与服务器通信以完成其工作，但是这不需要用户操心。

Java语言对于Web这样的分布式网络环境确实非常有价值。然而，它远远超越了这个领域，成为了一种强大的通用编程语言，适用于构建各种不依赖于网络特性的应用，并可满足其他应用的不同需求。它在远程主机上以安全的方式执行下载代码的能力正是许多组织的关键需求。

有些团队将Java作为一种通用编程语言，用于开发对机器无关性要求不高的项目。Java易于编程，安全性强，可用于快速地开发工作代码。由于它具有像垃圾收集和类型安全的引用这样的特性，某些常见的编程错误在Java中是不会发生的。对多线程的支持满足了基于网络和图形化用户界面的现代应用的需要，因为这些应用必须同时执行多个任务；而异常处理机制使得处理错误情况的任务变得简单易行。尽管其内置工具非常强大，Java依然是一种简单的语言，程序员可以很快就精通它。

Java语言被设计为尽可能提高可移植性，尽可能地减少对实现的依赖性。例如，在所有的实现中，int数值都是一个32位有符号的二进制补码整数，而不管执行程序的CPU的

1. 赖特（1867—1959）是20世纪最伟大的建筑师之一，崇尚自然、本性、活力和有机，代表作有流水别墅、古根海姆博物馆。

体系结构如何。对有关Java语言及其运行时环境的一切进行定义，可以使用户在任何地方运行编译过的代码，并与任何有Java运行时环境的人共享代码。

关于本书

本书的目的是向熟知基本编程概念的人讲授Java语言。我们并不刻意追求严密，也不求面面俱到。本书并不打算专门介绍面向对象编程，尽管书中涵盖了某些主题，以提供通用的术语。

第4版全面讲述Java程序设计语言，基于Java 2平台标准版5.0，其语言规范在《Java语言规范》（第3版）中有详细说明。本书还涵盖了由J2SE开发包5.0（通称为JDK 5.0，曾经也叫JDK 1.5.0）实现的主要的包（`java.lang.*`、`java.util`和`java.io`）中的绝大多数类。

如果读过本书的第3版，你会发现无论是Java语言还是本书本身都有一些重大变化，因为本书的第3版涵盖的是JDK 1.3。本版加了几章新内容，专门讨论泛型、枚举以及注解（这些是5.0版引入的主要的新特性），并且加了几小节论述断言和正则表达式。有些内容被重新整理，以适应变化，使行文更为流畅（例如在解释新的装箱和拆箱时）。其实每一章都在不同程度上进行了更新，其中有的牵涉到语言的新特性（像可变参数方法、新的增强型的for循环结构），有的涉及新的类（像用于格式化文本输出的`Formatter`），还有添加泛型之后所引发的类和方法的变化（如集合和反射类）。总之，改动渗透在整个第4版中。

Java语言具有很多与当今使用的大多数编程语言相通的特性。Java语言对于C和C++程序员来说应该是似曾相识的，因为它本身就是用相似的C和C++结构设计的。也就是说，本书既不是语言对比分析，也不是C和C++到Java的“过渡”指南——读者不须具备任何有关C或C++的知识。特别是对于C++程序员而言，那些曾经非常有用的知识反而会妨碍他们对Java的学习，必须学会忘却。

第1章“快速浏览”给出了Java语言的概貌。不熟悉面向对象编程概念的程序员应该阅读这个快速浏览，熟悉面向对象编程范型的程序员也会发现这一章对了解Java语言的面向对象特性是很有意义的。这里介绍的一些基本语言特性，是理解全书例子的基础。

第2章到第6章覆盖了Java语言面向对象的核心特性，即定义程序构件的类声明，以及根据类定义构造的对象。第2章“类与对象”描述了Java语言的基础：类。第3章“类的扩展”描述了已有的类可以如何被扩展，或者说被子类化，以创建具有更多数据和行为的新类。第4章“接口”描述了如何声明对行为进行抽象描述的接口类型，它可以为类的设计者和实现者提供最大的灵活性。第5章“嵌套类和接口”描述了如何在其他的类和接口的内部声明类和接口，以及这样做能带来什么好处。第6章“枚举类型”涵盖了类型安全的枚举常量的定义和使用方式。

第7章到第10章覆盖了大多数语言共通的标准结构。第7章“语言符号、值和变量”描述了Java中用来构造语句的语言符号、由Java语言定义的类型及它们可能的取值，以及将数据存储到对象、数组和方法内部所用到的变量。第8章“基本类型”探讨了基本类型和它们对应的包装类对象之间的关系，以及装箱和拆箱是如何透明地在它们之间进行转换的。

第9章“运算符和表达式”描述了Java语言的基本运算符，以及如何使用这些运算符构建表达式，这些表达式又如何计算。第10章“控制流”描述了控制语句如何引导语句执行的顺序。

第11章“泛型类型”描述了如何编写和使用泛型类型，以及泛型类型的强大威力和局限性。

第12章“异常与断言”描述了Java语言强大的错误处理能力，以及如何使用断言来验证代码所应有的行为。

第13章“字符串与正则表达式”描述了语言和运行时内置的对String对象的支持、底层字符串集的支持，以及用于正则表达式匹配的功能强大的实用程序。

第14章“线程”解释了Java语言的多线程机制。许多应用，如基于图形界面的软件，必须同时处理多个任务。这些任务必须相互协作，以确保其行为正确，而线程正好满足了多任务协作的需求。

第15章“注解”描述了用于对类和方法的一些语言外属性进行归档的注解类型。

第16章“反射”描述了运行期类型内省机制，以及如何在运行时动态构建和操纵未知类型的对象。

第17章“垃圾收集与内存”叙述了垃圾收集、终结，以及低强度的引用对象。

第18章“包”描述了如何将类和接口分组到彼此互相分离的包中。

第19章“文档注释”展示了如何在注释中编写引用文档。

第20章到第24章涵盖了主要的包。第20章“I/O包”描述基于流的输入/输出系统。第21章“集合”讲述了诸如集和列表这样的集合类（也称容器类）。第22章“其他工具”讲述了诸如位集、格式化输出、文本扫描以及随机数生成等其他的工具类。第23章“系统编程”将指引读者去了解系统类，它们提供了对底层平台特性的访问。第24章“国际化和本地化”讨论了一些工具，它们可用于创建能运行在多种语言和文化环境中的程序。

第25章“标准包”简要探讨了作为标准平台一部分的包，对本书中未详细介绍的包做了简要概述。

附录A“应用演化”探讨了处理应用和Java平台不断演化时所涉及的问题，以及这些问题对某些新的语言特性的影响。

附录B“常用表格”把一些有用的信息以表格的形式列出，供读者速查。

最后，“进阶读物”列出了各种参考文献，帮助读者了解Java的完整细节、面向对象技术、多线程编程、软件设计等主题。

示例和文档

本书的所有示例代码都使用完稿时的Java语言最新版本（即JDK 1.5.0_02产品版本）进行过编译和运行。本书只涉及当前受支持的特性，过时的类型、方法和字段都被忽略了，除非必须用到它们，或者了解过去对于理解现有特性必不可少。我们讨论的问题不限于让程序能够通过编译而已，学习一门语言自然要学会如何用好它。因此，我们还努力向读者展示良好的编程风格和设计原则。

我们在不少地方引用了在线文档。开发环境提供了一种从编译过的类中用它的文档注释自动生成文档（通常是HTML文档）的方式。这种文档通常是使用Web浏览器来阅读的。

致谢（第4版）

我们又一次需要对很多人表示感激，他们从自己忙碌的日程安排中抽出宝贵的时间，甚至不惜暂停写作自己的书，以他们的经验来帮我们完善本书，使其成为迄今为止最好的一个版本。当然，他们中的有些人提出的建议害得我们不得不去做大量的工作，以跟上语言和库的所有新特性，我们并不怪他们。

在审稿方面，我们首先要感谢柴田芳树。芳树是本书第3版日文版的翻译者，我们勘误表中至少85%的错误是他一人独立发现的。为了让第4版尽可能地减少错误，我们认识到需要让芳树尽早成为我们的复审者。于是，芳树成为了我们正式的复审人员，未经他首肯，我们不会把稿子拿出来做常规复审。他不仅对新增的内容进行纠错，提出了很有价值的反馈，而且还指出了多处Java语言已经发生变化而我们没有意识到的地方！

Brian Goetz也应该得到我们的特别感谢，不仅因为他复审了这一版的全部内容，而且有不少章节经过了他多次反复审阅。再次复审所给的时间非常紧迫，因此我们特别感谢Brian在特殊的关键时期给予我们的大力支持。

核心复审团队的其他人员还包括Beth Stearns和Kevin Davis，他们在漫长的过程中一直予以我们支持；Bob Withers、Herb Jellinek、Ken Loudon、Rich Wardwell和Kevin Hammond对前半部分书稿也提供了很有价值的反馈意见。非常感谢你们。

我们还想感谢Tim Peierls，他提供了枚举那一章中的国际象棋的例子；还有Keith Lea和Martin Buchholz，他们对泛型那部分材料进行了很有价值的复审。

毫无疑问，编写泛型这一章（以及附录中相关的部分）是这一版中最艰难的部分。由于手头最初只有《Java语言规范》（第3版）的草稿，所以要想深入探究新的类型系统是相当困难的，因此我们请来了大腕Neal Gafter、Josh Bloch、Gilad Bracha和Peter von der Ahé等人，他们很好地向我们讲授了泛型的知识，为此我们深深地感谢他们。Peter对泛型这一章做了细致的复审，特别有价值，因此我们要再多说一句感谢你。当然，教师不应因为学生的缺点而受到责备，因此在有关泛型的描述中出现的任何错误和遗漏都由我们自己承担全部责任。（我们还要感谢Phil Wadler，他在书临出版前在泛型的描述中及时地发现了两处遗漏。）

我们还要感谢下面的技术专家，他们在各自不同的领域对我们提供了帮助：Doug Lea复审了有关Java内存模型的材料，并且帮我们改进了并发集合的内容；Brian Goetz检查了并发集合的总结（还做了其他不少事情）；Mike “MadBot” McCloskey复审并协助我们改进了Scanner类及其相关正则表达式的用法的内容；Jeremy Manson则尝试着解释了终结与内存模型之间不易理解的深奥关系。

Rosemary Simpson总能在我们需要的时候出现。我一时想不出一句有创意的话来表达我们是何等欣赏她为本书做的索引，但是我可以肯定在第5版中我一定能够找到这样的赞美之词。

在制作方面，第4版的诞生颇费了些周折。起初，在第4版的酝酿阶段是Mike Hendrickson负责生产，之后转由Ann Sellers负责。Ann在Ebony Haight的协助下，经历了第4版早期的酝酿阶段，之后负责人又变为Greg Doench。Greg带领Noreen Regina、Stephane Nakib、Tyrrell Albaugh、Elizabeth Ryan和Mary Lou Nohr等人，帮助我们直至最终付梓。感谢你们所有人，以及所有的幕后人员。

还要感谢邻居Kathy Kerby、Ben Littauer以及Ariana Littauer一家，我们总是从他们家借用一些电脑小物件。有如此慷慨的芳邻是多么惬意的事情呀！

最后但绝非最不important地，要感谢我们的家庭，他们再次忍受了我们这些写书人带给他们的考验。

致谢（第3版）

第3版仍然需要更多的复审工作，而提供帮助的人们非常尽责。Lisa Friendly继续努力使项目保持正轨，以后我们会合作得更好。复审队伍既有新面孔又有老朋友，但对我们的帮助是同样重要，他们是Joshua Bloch、Joseph Bowbeer、Gilad Bracha、Keith Edwards、Joshua Engel、Rich Gillam、Peter Haggard、Cay Horstmann、Alexander Kuzmin、Doug Lea、Keith Lea、Tim Lindholm、David Mendenhall、Andrew M. Morgan、Ray Ortigas、Brian Preston、Mark Schuldenfrei、Peter Sparago、Guy Steele、Antoine Trux，以及我们的俄罗斯朋友们Leonid Arbousov、Valery Shakurov、Viatcheslav Rybalov、Eugene Latkin、Dmitri Khukhro、Konstantin Anisimov、Alexei Kaigorodov、Oleg Oleinik和Maxim Sokolnikov。我们洗耳恭听了Peter Jones、Robert W. Scheifler、Susan Snyder、Guy Steele、Jimmy Torres和Ann Wollrath的意见，努力把书中问题阐述得更清楚。Helen Leary一如既往地后勤保障工作做得很好。

Addison-Wesley团队一直在提供写作资料方面的支持，他们是Julie DiNicola、Mike Hendrickson和Tracy Russ。

在上一版问世前，Peet咖啡茶餐馆在东海岸开张营业，所以本书住在东部的作者可以定期地享受其美味。这个世界在不断地进步啊。

致谢（第2版）

第2版要感谢的人和第1版基本一样。

丛书编辑Lisa Friendly继续坚定地体现了支持和关心。复审队伍人数有所减少，与第1版有些人员重复，但给予我们的帮助和全心投入是相同的。Steve Byrne、Tom Cargill、Mary Dageforde、Tim Lindholm和Rob Murray对全书进行了通审，对于保证本书表述清晰来说是至关重要的。Brian Beck、Peter Jones、Doug Lea、Bryan O'Sullivan、Sue Palmer、Rosanna Lee、Lori Park、Mark Reinhold、Roger Riggs、Ann Wollrath和Ken Zadek重点复审了本书的重要章节。Guy Steele的支持是一贯而热忱的。Rosemary Simpson为制作本书的索引进行了紧张的工作，付出了巨大的努力，为此我们表示深切的感谢。Carla Carlson

和Helen Leary给予了后勤支持，使所有工作都顺利进行。Gerry Wiener提供了第550页上的中国藏族文字，Craig Preston和Takao Miyatani也提供了这方面的帮助。所有提交第1版的勘误和建议的人都对我们帮助很大。

由于某些无法解释的原因，我们在第1版的致谢中遗漏了来自Addison-Wesley的朋友们——幸亏他们中的大多数又都出现在了这一版中。这两版中来自Addison-Wesley的朋友包括Kate Duffy、Rosa Gonzales、Mike Hendrickson、Marina Lang、Shannon Patti、Marty Rabinowitz、Sarah Weaver和Pamela Yee。我们可能遗漏了一些付出很多的人，但是对他们我们同样致以诚挚的谢意。

修订版还得到了Josh Bloch、Joe Fialli、Jimmy Torres、Benjamin Renaud、Mark Reinhold、Jen Volpe和Ann Wollrath的帮助。

Peet 咖啡茶餐馆一直是我们这些咖啡因饮料“鉴赏家”的赞助商。

致谢（第1版）

任何科技书籍的写作都不是一座无援的孤岛，而我们这本书的写作更像一块紧密相连的大陆。许多人都提供了技术帮助、高水平的复审、极为有用的信息和书籍写作的建议。

Trilithon Software公司的编辑Henry McGilton所起的作用就像是“首席编辑救火官”，他的帮助使本书的出版成为现实。丛书编辑Lisa Friendly坚定地为我们提供了支持与帮助。

许多称职的复审者忙里偷闲对本书的相关材料进行了阅读、编辑、建议、修订和删除，所有这一切都为了使本书更加完美。Kevin Coyle堪称本书最细致的编辑复审者。Karen Bennet、Mike Burati、Patricia Giencke、Steve Gilliard、Bill Joy、Rosanna Lee、Jon Madison、Brian O'Neill、Sue Palmer、Stephen Perelgut、R. Anders Schneiderman、Susan Sim、Bob Sproull、Guy Steele、Arthur van Hoff、Jim Waldo、Greg Wilson和Ann Wollrath都提供了深入的复审。Geoff Arnold、Tom Cargill、Chris Darke、Pat Finnegan、Mick Jordan、Doug Lea、Randall Murray、Roger Riggs、Jimmy Torres、Arthur van Hoff和Frank Yellin在关键环节上提供了有价值的评论和技术信息。

Alka Deshpande、Sharon Flank、Nassim Fotouhi、Betsy Halstead、Kee Hinckley、Dr. K. Kalyanasundaram、Patrick Martin、Paul Romagna、Susan Snyder和Nicole Yankelovich通力合作创造了第164页和第550页的五个非ISO-Latin-1文本的单词。Jim Arnold提供了关于“smoog”和“moorge”的正确拼写、用法和词源方面的研究。Ed Mooney 帮助我们准备了文档。Herb和Joy Kaiser是我们的克罗地亚语顾问。Cookie Callahan、Robert E. Pierce和Rita Tavilla 提供了必要的帮助，使得本项目在很多时候得以继续进行，否则我们将会停滞不前。

感谢Kim Polese，他给我们总结了为什么Java编程语言对计算机用户和程序员同等重要。

Susan Jones、Bob Sproull、Jim Waldo和Ann Wollrath在关键时刻提供了支持和建议。我们还要感谢我们的家人，他们除了用亲情支持我们写书之外，还经常使我们劳逸结合，张弛有度，为此我们要深深地感谢他们。

还要感谢Peet咖啡茶餐馆的朋友们，他们让我们品尝到了这个星球上最好的咖啡。

尽管所有人员都做出了不懈的努力，本书中难免还会存在错误和缺陷，全部责任都由作者承担。

是结果！朋友，我已经取得了大量的结果，

我已知道有好几千种方法是行不通的。

托马斯·爱迪生

Contents

1	A Quick Tour	1
1.1	Getting Started	1
1.2	Variables	3
1.3	Comments in Code	6
1.4	Named Constants	7
1.5	Unicode Characters	8
1.6	Flow of Control	9
1.7	Classes and Objects	12
1.7.1	Creating Objects	13
1.7.2	Static or Class Fields	14
1.7.3	The Garbage Collector	15
1.8	Methods and Parameters	15
1.8.1	Invoking a Method	15
1.8.2	The <code>this</code> Reference	17
1.8.3	Static or Class Methods	17
1.9	Arrays	18
1.10	String Objects	21
1.10.1	String Conversion and Formatting	23
1.11	Extending a Class	24
1.11.1	Invoking Methods of the Superclass	25
1.11.2	The <code>Object</code> Class	26
1.11.3	Type Casting	27
1.12	Interfaces	27
1.13	Generic Types	29
1.14	Exceptions	32
1.15	Annotations	35
1.16	Packages	36
1.17	The Java Platform	38
1.18	Other Topics Briefly Noted	39

2	Classes and Objects	41
2.1	A Simple Class	42
2.1.1	Class Members	42
2.1.2	Class Modifiers	43
2.2	Fields	44
2.2.1	Field Initialization	44
2.2.2	Static Fields	45
2.2.3	final Fields	46
2.3	Access Control	47
2.4	Creating Objects	49
2.5	Construction and Initialization	50
2.5.1	Constructors	50
2.5.2	Initialization Blocks	54
2.5.3	Static Initialization	55
2.6	Methods	56
2.6.1	Static Methods	58
2.6.2	Method Invocations	58
2.6.3	Methods with Variable Numbers of Arguments	60
2.6.4	Method Execution and Return	62
2.6.5	Parameter Values	63
2.6.6	Using Methods to Control Access	65
2.7	this	68
2.8	Overloading Methods	69
2.9	Importing Static Member Names	71
2.10	The main Method	73
2.11	Native Methods	74
3	Extending Classes	75
3.1	An Extended Class	76
3.2	Constructors in Extended Classes	80
3.2.1	Constructor Order Dependencies	81
3.3	Inheriting and Redefining Members	84
3.3.1	Overriding	84
3.3.2	Hiding Fields	86
3.3.3	Accessing Inherited Members	86
3.3.4	Accessibility and Overriding	88
3.3.5	Hiding Static Members	89
3.3.6	The super Keyword	89
3.4	Type Compatibility and Conversion	90
3.4.1	Compatibility	90
3.4.2	Explicit Type Casting	91
3.4.3	Testing for Type	92
3.5	What protected Really Means	93

3.6	Marking Methods and Classes <code>final</code>	96
3.7	Abstract Classes and Methods	97
3.8	The <code>Object</code> Class	99
3.9	Cloning Objects	101
3.9.1	Strategies for Cloning	101
3.9.2	Correct Cloning	103
3.9.3	Shallow versus Deep Cloning	106
3.10	Extending Classes: How and When	107
3.11	Designing a Class to Be Extended	108
3.11.1	Designing an Extensible Framework	109
3.12	Single Inheritance versus Multiple Inheritance	114
4	Interfaces	117
4.1	A Simple Interface Example	118
4.2	Interface Declarations	120
4.2.1	Interface Constants	121
4.2.2	Interface Methods	122
4.2.3	Interface Modifiers	122
4.3	Extending Interfaces	122
4.3.1	Inheriting and Hiding Constants	123
4.3.2	Inheriting, Overriding, and Overloading Methods	125
4.4	Working with Interfaces	126
4.4.1	Implementing Interfaces	127
4.4.2	Using an Implementation	129
4.5	Marker Interfaces	130
4.6	When to Use Interfaces	131
5	Nested Classes and Interfaces	133
5.1	Static Nested Types	133
5.1.1	Static Nested Classes	134
5.1.2	Nested Interfaces	135
5.2	Inner Classes	136
5.2.1	Accessing Enclosing Objects	138
5.2.2	Extending Inner Classes	139
5.2.3	Inheritance, Scoping, and Hiding	140
5.3	Local Inner Classes	142
5.3.1	Inner Classes in Static Contexts	144
5.4	Anonymous Inner Classes	144
5.5	Inheriting Nested Types	146
5.6	Nesting in Interfaces	148
5.6.1	Modifiable Variables in Interfaces	149
5.7	Implementation of Nested Types	149

6	Enumeration Types	151
6.1	A Simple Enum Example	151
6.2	Enum Declarations	152
6.2.1	Enum Modifiers	154
6.3	Enum Constant Declarations	154
6.3.1	Construction	155
6.3.2	Constant Specific Behavior	156
6.4	<code>java.lang.Enum</code>	159
6.5	To Enum or Not	160
7	Tokens, Values, and Variables	161
7.1	Lexical Elements	161
7.1.1	Character Set	161
7.1.2	Comments	163
7.1.3	Tokens	164
7.1.4	Identifiers	164
7.1.5	Keywords	165
7.2	Types and Literals	166
7.2.1	Reference Literals	167
7.2.2	Boolean Literals	167
7.2.3	Character Literals	167
7.2.4	Integer Literals	167
7.2.5	Floating-Point Literals	168
7.2.6	String Literals	168
7.2.7	Class Literals	169
7.3	Variables	169
7.3.1	Field and Local Variable Declarations	170
7.3.2	Parameter Variables	171
7.3.3	<code>final</code> Variables	171
7.4	Array Variables	173
7.4.1	Array Modifiers	174
7.4.2	Arrays of Arrays	174
7.4.3	Array Initialization	175
7.4.4	Arrays and Types	177
7.5	The Meanings of Names	178
8	Primitives as Types	183
8.1	Common Fields and Methods	184
8.1.1	Construction	185
8.1.2	Constants	185
8.1.3	Common Methods	186
8.2	<code>Void</code>	187
8.3	<code>Boolean</code>	187

8.4	Number	188
8.4.1	The Integer Wrappers	188
8.4.2	The Floating-Point Wrapper Classes	191
8.5	Character	192
8.5.1	Working with UTF-16	196
8.6	Boxing Conversions	198
9	Operators and Expressions	201
9.1	Arithmetic Operations	201
9.1.1	Integer Arithmetic	202
9.1.2	Floating-Point Arithmetic	202
9.1.3	Strict and Non-Strict Floating-Point Arithmetic	203
9.2	General Operators	204
9.2.1	Increment and Decrement Operators	205
9.2.2	Relational and Equality Operators	206
9.2.3	Logical Operators	207
9.2.4	instanceof	208
9.2.5	Bit Manipulation Operators	208
9.2.6	The Conditional Operator ?:	210
9.2.7	Assignment Operators	212
9.2.8	String Concatenation Operator	214
9.2.9	new	214
9.3	Expressions	214
9.3.1	Order of Evaluation	214
9.3.2	Expression Type	215
9.4	Type Conversions	216
9.4.1	Implicit Type Conversions	216
9.4.2	Explicit Type Casts	219
9.4.3	String Conversions	220
9.5	Operator Precedence and Associativity	221
9.6	Member Access	223
9.6.1	Finding the Right Method	224
10	Control Flow	229
10.1	Statements and Blocks	229
10.2	if-else	230
10.3	switch	232
10.4	while and do-while	235
10.5	for	236
10.5.1	Basic for Statement	236
10.5.2	Enhanced for Statement	239
10.6	Labels	241
10.7	break	241