

写给大忙人看的 *Java*核心技术

[美] Cay S. Horstmann 著
杨谦 王巍 高伟斌 谢志强 译



Core Java for the Impatient

写给大忙人看的 *Java*核心技术

[美] Cay S. Horstmann 著
杨谦 王巍 高伟斌 谢志强 译



Core Java
for the Impatient

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内容简介

《写给大忙人看的Java核心技术》为经典著作*Core Java*作者Cay S. Horstmann的全新力作，以关键的核心技术为着眼点，以全局高度提炼语言精髓，以高浓度、高含金量的方式对Java 8新特性进行细致讲解，将其沉浸于Java多年的真知灼见蕴藏于书中，既扎实覆盖Java编程的基础知识，更独具匠心地从Java 8极其重要的新特性之一lambda表达式发端，带领读者一路探索和实践Java 8的诸多新特性，诸如针对数据处理的Stream、并发编程中的增强部分、全新的时间/日期API，以及脚本语言在Java中的使用。

“For the Impatient”系列风格鲜明，除悉数出自大师之手外，更以文笔轻松、易读易懂、富含生动且可用于实践的示例程序（下载源代码后可直接运行）见长。这种亲身实践的讲授模式，非常便于各阶层读者加深对技术的理解和掌握，初学者可用来参考入门，有开发经验者也可用来巩固编程核心知识，与时俱进地把握技术发展现状与未来趋势。

本书适合有一定Java开发经验的工程师或者架构师阅读，也可以作为Java新特性的参考手册。

Authorized translation from the English language edition, entitled *Core Java for the Impatient*, 9780321996329 by Cay S. Horstmann, published by Pearson Education, Inc., publishing as Addison-Wesley, Copyright © 2015 Pearson Education, 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.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY Copyright © 2016.

本书简体中文版专有出版权由Pearson Education 培生教育出版亚洲有限公司授予电子工业出版社。未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书简体中文版贴有Pearson Education 培生教育出版集团激光防伪标签，无标签者不得销售。

版权贸易合同登记号 图字：01-2015-6194

图书在版编目(CIP)数据

写给大忙人看的Java核心技术 / (美) 霍斯特曼 (Horstmann, C.S.) 著; 杨谦等译. —北京: 电子工业出版社, 2016.1

书名原文: Core Java for the Impatient

ISBN 978-7-121-27837-2

I. ①写… II. ①霍… ②杨… III. ①JAVA语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第301428号

策划编辑: 张春雨

责任编辑: 李云静

印刷: 三河市双峰印刷装订有限公司

装订: 三河市双峰印刷装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路173信箱

邮编: 100036

开本: 787×980 1/16

印张: 30.25

字数: 557千字

版次: 2016年1月第1版

印次: 2016年1月第1次印刷

定 价: 108.00元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至zltz@phei.com.cn, 盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线: (010) 88258888。

推荐序1

作为一个经历过很多面试的求职者，也是一个面试过很多求职者的面试官，深知 Java 基础、核心原理的重要性，其重要性甚至超过你的技术能力。我这里不是宣扬技术能力不重要，技术能力是企业的第一生产力，是老板雇用你的价值所在。但是作为面试者要清楚，你口若悬河介绍自己曾经用过的某工具的时候，面试官未必有共鸣，因为他可能对该工具根本就不熟悉——尤其是你供职的上一家公司和现在应聘的这家公司处于不同领域背景的情况下。但是如果你对于Java核心的东西回答得不好，那么面试官就有足够的理由认为你不行。

作为一个有十几年 Java 编程经验的开发者，本人深切体会到了具备扎实的 Java 基础知识对工作的重要性。可能有人认为框架更重要。是的，框架把程序员要做的很多事情封装起来，让我们能够专注于企业业务开发，能显著提高开发效率。Java之所以多年来能稳居开发语言排行榜前茅，除了其各种优秀的语言特性，与各开源社区贡献的这些优秀的框架也是分不开的。但是对于初学者，首先应该注重基础和原理，一开始就上手框架，很容易忽略那些基础知识。很多人认为 Java Web 开发就是SSH，只须把一系列的配置文件复制来复制去；很多程序员做了好几年的Java Web，但是对对象的生命周期都无法描述清楚，也不理解为什么要用 spring 管理；还有一少部分人甚至连js、jsp程序哪个在服务器端执行、哪个在客户端执行都分不清楚，弄出来很多本来期望在客户端弹出的窗口却在服务器端弹出的笑话……这些都是一开始上手框架，没有注重基础学习的后果。可以说“成也框架，败也框架”。

性能是架构师无时无刻不在考虑的问题。作为一名在职架构师，除了对系统层次设计的把握之外，我最关注的还是算法、代码层次的性能优化，因为这是我们系统的基础。本人虽然不再写业务代码，但一些基础代码还是亲力亲为，并喜欢用一些工具或者自己写脚本来检测其性能；制定开发规范并对团队成员进行 Code Review 更是我的日常工作之一。勿在浮沙筑高台，基本的代码基础都没打好，谈何大数据、高并发？

IV

推荐序1

刚拿到本书的样稿时，以为它也只不过是一本普通的 Java 基础书而已。因为以前已经接触过很多介绍 Java 基础的书籍，无外乎把 Java API 里的东西用另外一种方式又表达了一遍而已。但读过几章之后，发现能够从中获得的益处绝不仅仅是学习到 Java 的一些新特性，更多的是了解到很多以前没有注意到的细节和陷阱，此外还有这些陷阱背后的一些原理介绍，这样我们对自己编码的某些操作的代价有了一个大概的判断。但对于底层的東西书里又懂得点到为止，不至于让人读起来如坠云里雾里，偏离了本书的主旨。

特将此书推荐给正在求职 Java 开发工作的人，扔掉网上那些千篇一律的面试题，这本书能让你节省时间快速复习。也将此书推荐给 Java 程序员、技术经理、架构师们——总之，还没有完全脱离开发的在职人员，不需要刻意去通读它，把它放在案头，在你用 Java 实现某项功能的时候，看看这本书上的相关介绍——你会发现，程序原来还可以这样写，而且这样写居然更好。

孔德芳

CSDN Java 版大版主、讯联数据架构师

推荐序2

两周前，我曾经在IBM一起战斗过的兄弟杨谦和高伟斌，邀请我为本书作序，我虽然心怀忐忑，但还是欣然应允了。

Java老手都会知道*Core Java*（《Java核心技术》）这部大作，本书的作者Cay S. Horstmann也正是*Core Java*的作者之一。Cay S. Horstmann是圣荷西州立大学计算机系教授、Java语言的积极倡导者，他能够以形象的语言，向大家阐述抽象的Java概念，让初学者能够很容易地掌握Java这门语言。然而光阴似箭，Java到今天，已经经历了二十多个寒暑。这么长的时间里，Java也已经演化到了8.0版本，许多更先进的思想和更简洁的API被加入进来，同时也抛弃了很多过时的内容。在这样的情况下，*Core Java*这本大部头就显得多少有些不合时宜了，这也是Cay S. Horstmann写作本书的原因所在。在本书中，Cay S. Horstmann针对Java应用开发人员，提炼出了Java最核心、最新鲜、工作中最常用到的部分，舍弃了诸如Swing等在现实工作中很少用到的内容，让读者在最短的时间里尽情吸收Java语言的精华，可以快速地把所学的知识应用到实际工作中去。

本书的译者之中，杨谦和高伟斌是我在IBM CDL时朝夕相处的同事，他们都有着十多年的Java实战经验和深厚的技术功底，同时还秉承了IBM一贯的科学严谨作风，再加上长期外企工作环境的熏陶，锻炼出了很好的对英文技术资料的阅读理解能力。我相信，他们完全能够将本书中的内容用中文原汁原味地呈献给读者。

霍振中

北京大米未来科技有限公司联合创始人、CTO

译者序

2014年Oracle发布了Java 8。Java 8引入了很多革命性的新特性，例如，lambda表达式、Stream、函数式接口、新的时间/日期API、增强的并发API等。本书特色鲜明，它不仅为开发者全面介绍这些新特性，而且从Java最核心基础——类、继承、接口、反射、泛型编程、集合等开始。全书开始几章全面细致、重点突出地介绍Java的核心基础并在第3章穿插了lambda表达式。之后各章全面介绍Java 8中的新特性和原有功能的改进，其中第8章详细介绍Stream的使用，第10章介绍最新的并发编程API。全书内容翔实，实例丰富，理论与实践并举。

译者在翻译本书时，一边感到万分的荣幸，一边也颇觉战战兢兢。吉尔伯特·海特（美国教育家）曾经说过：写了一本很糟糕的书只是犯错而已，而把一本好书翻译得很糟糕则是犯罪。但是对于本书这样的Java核心技术和最新特性之作，即便是冒着犯罪的风险，也值得翻译出来并推荐给大家。

本书承载的内容丰富而深入。在翻译的过程中，译者也常感汗颜，原来自诩对Java 8新特性颇为了解的我深刻感受到自己知识面的浅薄和不完整，于是翻译过程也成为自己检讨和学习的过程，收获颇丰。本书的作者是经典书籍*Core Java*（《Java核心技术》）的主要作者，他有着丰富的Java知识，并一直在普及介绍Java的最新特性。因此，希望大家反复阅读，必定受益无穷。

由于本书作者的知识广博、经验丰富，而译者本人知识有限、时间仓促，难免对原文的个别之处无法原汁原味地呈现给读者，在此敬请各位读者多多包涵，不吝指教，并帮忙指出纰漏。也希望能就本书的内容与读者有更多的交流，我的邮箱是yqbjtu@163.com。

最后，我要感谢电子工业出版社博文视点公司的张春雨编辑在译书的过程中给予我们的莫大帮助。我还要感谢本书的其他译者王巍、高伟斌、吴建军、张媛、田敏、王雪芳、杨建轩、康引鸽、刘增涛、杨燕和东莞经济贸易学校的谢志强老师等。特别感谢CSDN论坛Java版大版主（讯联数据架构师）孔德芳、北京大米未来科技有限公司CTO霍振中，他们的参与使得本书的质量得到进一步保证。

杨 谦

2016年1月于宁波江南景苑

谨以此书献给*Chi*，对我最有耐心的人。

前 言

Java已诞生约二十年，经典书籍*Core Java*（《Java核心技术》）第一卷和第二卷用两千多页全面细致地介绍Java语言和所有核心类库，以及各版本之间的大量变化。但是Java 8改变了这一切。许多旧的Java习惯用法不再需要，并且有了更快、更方便的学习Java的途径。在本书中，我们将展示现代Java中“好的地方”，以方便你快速将它们应用到工作中。

与我之前的“*For the Impatient*”系列书籍类似，本书也会开门见山，告诉你解决问题所需要知道的内容，不会就某个范例比另外一个更有优势而展开讨论。我们会采用有序的且篇幅较小的方式展示信息，这样需要信息时，你就可以通过快速检索获取。

假设你精通其他编程语言，例如C++、JavaScript、Objective C、PHP或Ruby，通过本书，你将学习如何成为一个称职的Java程序员。我们会涵盖Java开发者需要知道的方方面面，包括Java 8引入的、功能强大的lambda表达式和Stream。我会告诉你哪里可以找到关于旧式概念的更多信息，你可能依然会在一些遗留代码中看到这些旧式概念，但是我们不会就此展开。

使用Java的一个关键原因是解决并发编程。使用Java类库中现成的并行算法和线程安全的数据结构，彻底改变了程序员处理并发编程的方式。我们会提供新的示例，告诉你如何使用强大的类库特性代替那些易出错、低水平的数据结构。

传统的Java书籍关注UI编程——但是现在，很少有开发者用桌面计算机开发用户界面了。如果你想将Java用在服务器端编程或者Android编程，则可以更有效地使用本书，而不被桌面GUI代码干扰。

最后，本书是写给应用软件编程者的，不是大学教科书，也不适合系统向导。本书包含了一些应用软件编程者需要设法解决的问题，例如日志和文件使用，但是你不

X

前 言

必学习如何实现链表或者开发一个Web服务器。

希望你喜欢对现代Java的快速介绍，同时希望它能提高你的Java开发工作生产效率并愉快地使用Java。

如果你发现本书错误或者有改进的建议，请访问<http://horstmann.com/javaimpatient>并留言。在页面上，你也可以找到包含本书所有代码示例的归档文件。

致 谢

一如既往地感谢我的编辑Greg Doech，是他热情支持出版这一本简短的书，用来介绍Java 8新特性。Dmitry Kirsanov和Alina Kirsanova再次以神奇的速度，细致地将XHTML手稿转化成引人入胜的书。特别感谢优秀的评审团队，他们发现了很多错误并给出深思熟虑的改进建议。他们是：Andres Almiray、Brian Goetz、Marty Hall、Mark Lawrence、Doug Lea、Simon Ritter、Yoshiki Shibata和Christian Ullenboom。

Cay Horstmann
瑞士，比尔市
2015年1月

作者简介

Cay S. Horstmann是《写给大忙人看的Java SE 8》(*Java SE 8 for the Really Impatient*)与《快学Scala》(*Scala for the Impatient*) (都是Addison-Wesley的书, 中文版都由电子工业出版社出版)两本书的作者, 也是*Core Java™, Volumes I and II, Ninth Edition* (Prentice Hall于2013年出版)一书的主要作者。他还编写了一系列针对专业编程人员和计算机专业学生的书籍。他是圣荷西州立大学计算机系教授, 也是一位Java拥护者。

目 录

第1章 基本的编程结构	1
1.1 第一个程序	2
1.1.1 “Hello, World”程序解析	2
1.1.2 编译与运行Java程序	3
1.1.3 方法调用	5
1.2 基本类型	7
1.2.1 整型	7
1.2.2 浮点类型	8
1.2.3 char型	9
1.2.4 布尔型	10
1.3 变量	10
1.3.1 变量的声明	10
1.3.2 名称	10
1.3.3 初始化	11
1.3.4 常量	11
1.4 算术操作	13
1.4.1 赋值	13
1.4.2 基本运算	14
1.4.3 数学方法	15
1.4.4 数字类型转换	16
1.4.5 关系和逻辑操作符	17
1.4.6 大数	19
1.5 字符串	19
1.5.1 字符串连接	20
1.5.2 子字符串	21
1.5.3 字符串比较	21
1.5.4 数字与字符串转换	23
1.5.5 String类API	24
1.5.6 编码点 (code point) 和编码单元 (code unit)	25

XIV

目 录

1.6	输入与输出	26
1.6.1	读取输入	27
1.6.2	格式化输出	28
1.7	控制流	30
1.7.1	分支	30
1.7.2	循环	32
1.7.3	跳出循环与继续循环	34
1.7.4	局部变量作用域	36
1.8	数组和数组列表	38
1.8.1	使用数组	38
1.8.2	构造数组	39
1.8.3	数组列表	39
1.8.4	基本类型包装类	41
1.8.5	增强的for循环	41
1.8.6	数组与数组列表的复制	42
1.8.7	数组算法	43
1.8.8	命令行参数	44
1.8.9	多维数组	45
1.9	功能分解	48
1.9.1	静态方法的声明与调用	48
1.9.2	数组参数与返回值	48
1.9.3	可变参数	49
	练习	50
第2章	面向对象编程	53
2.1	使用对象	54
2.1.1	Accessor（访问器）和Mutator（更改器）方法	56
2.1.2	对象引用	56
2.2	实现类	58
2.2.1	实例变量	58
2.2.2	方法头	59
2.2.3	方法体	59
2.2.4	实例方法调用	60
2.2.5	this引用	61
2.2.6	值调用	62
2.3	构造对象	63
2.3.1	实现构造函数	64
2.3.2	重载	64
2.3.3	调用另一个构造函数	65

2.3.4	默认初始化	66
2.3.5	实例变量初始化	66
2.3.6	final实例变量	67
2.3.7	无参构造函数	68
2.4	静态变量和方法	69
2.4.1	静态变量	69
2.4.2	静态常量	70
2.4.3	静态初始块	71
2.4.4	静态方法	71
2.4.5	工厂方法	73
2.5	包	73
2.5.1	包的声明	74
2.5.2	类路径	75
2.5.3	包作用域	77
2.5.4	导入包	78
2.5.5	静态导入	79
2.6	嵌套类	80
2.6.1	静态嵌套类	80
2.6.2	内部类	82
2.6.3	内部类的特殊语法	85
2.7	文档注释	86
2.7.1	插入注释	86
2.7.2	类注释	87
2.7.3	方法注释	88
2.7.4	变量注释	88
2.7.5	通用注释	89
2.7.6	链接	89
2.7.7	包和概述注释	90
2.7.8	注释的提取	90
	练习	91
第3章	接口和lambda表达式	95
3.1	接口	96
3.1.1	声明接口	96
3.1.2	实现接口	97
3.1.3	转换为接口类型	99
3.1.4	强制类型转换 (cast) 和instanceof操作符	99
3.1.5	继承接口	100
3.1.6	实现多个接口	101

3.1.7	常量	101
3.2	静态方法和默认方法	101
3.2.1	静态方法	102
3.2.2	默认方法	102
3.2.3	解决默认方法冲突	103
3.3	接口示例	105
3.3.1	Comparable接口	105
3.3.2	Comparator接口	107
3.3.3	Runnable接口	108
3.3.4	UI (User Interface) 回调	109
3.4	lambda表达式	110
3.4.1	lambda表达式语法	110
3.4.2	函数式接口	111
3.5	方法引用和构造函数引用	112
3.5.1	方法引用	113
3.5.2	构造函数引用	114
3.6	使用lambda表达式	115
3.6.1	实现延迟执行	115
3.6.2	选择函数式接口	116
3.6.3	实现自己的函数式接口	118
3.7	lambda表达式和变量作用域	119
3.7.1	lambda表达式的作用域	119
3.7.2	访问来自闭合作用域的变量	120
3.8	高阶函数	123
3.8.1	返回函数的方法	123
3.8.2	修改函数的方法	123
3.8.3	Comparator方法	124
3.9	局部内部类	125
3.9.1	局部类	125
3.9.2	匿名类	126
	练习	127
第4章	继承与反射	131
4.1	继承一个类	132
4.1.1	父类与子类	132
4.1.2	定义和继承子类方法	132
4.1.3	方法覆盖	133
4.1.4	子类的构造	134
4.1.5	父类赋值	135

4.1.6	转换	136
4.1.7	final方法和类	136
4.1.8	抽象方法和类	137
4.1.9	受保护访问	138
4.1.10	匿名子类	139
4.1.11	继承和默认方法	140
4.1.12	带super的方法表达式	141
4.2	Object: 终极父类	141
4.2.1	toString方法	142
4.2.2	equals方法	144
4.2.3	hashCode方法	147
4.2.4	克隆对象	148
4.3	枚举	151
4.3.1	枚举方法	152
4.3.2	构造函数、方法和域	153
4.3.3	实例的实现体	153
4.3.4	静态成员	154
4.3.5	switch枚举对象	155
4.4	运行时类型信息和资源	156
4.4.1	Class类	156
4.4.2	资源加载	160
4.4.3	类加载器	160
4.4.4	上下文类加载器	162
4.4.5	服务加载器	164
4.5	反射	165
4.5.1	枚举类成员	165
4.5.2	对象检查	167
4.5.3	方法调用	167
4.5.4	对象构造	168
4.5.5	JavaBeans	169
4.5.6	使用数组	170
4.5.7	代理	172
练习	174
第5章	异常、断言和日志处理	177
5.1	异常处理	178
5.1.1	异常抛出	178
5.1.2	异常继承层次	179
5.1.3	已检查异常的声明	181

5.1.4	异常捕获	182
5.1.5	try-with-resources语句	183
5.1.6	finally子句	185
5.1.7	异常重抛和链接	186
5.1.8	堆栈踪迹	188
5.1.9	Objects.requireNonNull方法	189
5.2	断言	189
5.2.1	使用断言	190
5.2.2	启用和禁用断言	191
5.3	记录日志	191
5.3.1	采用Logger	192
5.3.2	日志记录器	192
5.3.3	日志级别	192
5.3.4	其他日志记录方法	193
5.3.5	日志记录的配置项	195
5.3.6	日志处理器	196
5.3.7	过滤器和格式化器	199
	练习	199
第6章	泛型编程	203
6.1	泛型类	204
6.2	泛型方法	205
6.3	类型限定	206
6.4	类型变异和通配符	207
6.4.1	子类型通配符	208
6.4.2	父类型通配符	209
6.4.3	带类型变量的通配符	210
6.4.4	无限定通配符	212
6.4.5	通配符捕获	212
6.5	Java虚拟机中的泛型	213
6.5.1	类型擦除	213
6.5.2	转换插入	214
6.5.3	桥方法	215
6.6	泛型约束	216
6.6.1	无基本类型参数	217
6.6.2	所有类型在运行时都是原始的	217
6.6.3	不能实例化类型变量	218
6.6.4	不能构造参数化类型的数组	220
6.6.5	静态上下文中的类类型变量不是有效的	221

6.6.6	类型擦除后的方法可能不冲突	222
6.6.7	异常与泛型	223
6.7	反射与泛型	224
6.7.1	Class<T>类	224
6.7.2	虚拟机中的泛型类型信息	225
	练习	227
第7章	集合	233
7.1	集合类框架概要	234
7.2	迭代器	238
7.3	set	240
7.4	map	241
7.5	其他集合	245
7.5.1	Properties	245
7.5.2	位组	247
7.5.3	枚举set和枚举map	248
7.5.4	栈、队列、双端队列、优先级队列	249
7.5.5	弱哈希映射表	250
7.6	视图	251
7.6.1	范围 (range)	251
7.6.2	空视图和单例视图	252
7.6.3	不可修改的视图	252
	练习	253
第8章	Stream	257
8.1	从迭代到Stream操作	258
8.2	创建Stream	259
8.3	filter、map和flatMap方法	261
8.4	提取子流和组合流	262
8.5	其他流转换	263
8.6	简单归约	264
8.7	Optional类型	265
8.7.1	如何使用Optional类型值	265
8.7.2	如何不使用Optional类型值	266
8.7.3	创建Optional类型值	267
8.7.4	使用flatMap来组合可选值函数	267
8.8	收集结果	268
8.9	将结果收集到map中	270

8.10	分组和分片	271
8.11	下游收集器	272
8.12	归约操作	274
8.13	基本类型流	276
8.14	并行流	278
	练习	280
第9章	输入与输出处理	285
9.1	输入/输出流：Reader和Writer	286
9.1.1	获取流对象	286
9.1.2	读取字节	287
9.1.3	写字节	288
9.1.4	字符集编码	288
9.1.5	文本输入	291
9.1.6	文本输出	292
9.1.7	读写二进制数据	294
9.1.8	随机存取文件	295
9.1.9	内存映射文件	295
9.1.10	文件锁	296
9.2	路径、文件和目录	297
9.2.1	路径	297
9.2.2	创建文件和目录	299
9.2.3	复制、移动和删除文件	300
9.2.4	访问目录内容	301
9.2.5	ZIP文件系统	304
9.3	URL连接	305
9.4	正则表达式	307
9.4.1	正则表达式语法	307
9.4.2	匹配一个或多个	311
9.4.3	分组	312
9.4.4	消除或替换匹配结果	313
9.4.5	标记	314
9.5	序列化	315
9.5.1	Serializable接口	315
9.5.2	瞬态实例变量	317
9.5.3	readObject和writeObject方法	317
9.5.4	readResolve和writeReplace方法	319
9.5.5	版本化	320
	练习	321

第10章 并发编程	325
10.1 并发任务	326
10.1.1 运行任务	326
10.1.2 Future和Executor服务	328
10.2 线程安全	331
10.2.1 可见性	331
10.2.2 竞争条件	333
10.2.3 安全并发的策略	336
10.2.4 不可变类	336
10.3 并行算法	337
10.3.1 并行流	338
10.3.2 并行数组操作	338
10.4 线程安全的数据结构	339
10.4.1 ConcurrentHashMap	340
10.4.2 阻塞队列	341
10.4.3 其他线程安全的数据结构	343
10.5 原子值	344
10.6 锁	346
10.6.1 可重入锁 (ReentrantLock)	347
10.6.2 synchronized关键字	348
10.6.3 条件等待	350
10.7 线程	352
10.7.1 启动线程	352
10.7.2 线程中断	353
10.7.3 线程变量	355
10.7.4 线程的其他属性	356
10.8 异步计算	356
10.8.1 UI回调中的长期运行任务	356
10.8.2 可完成的future	358
10.9 进程	361
10.9.1 构建进程	361
10.9.2 运行进程	363
练习	364
第11章 注解	371
11.1 使用注解	372
11.1.1 注解元素	372
11.1.2 多注解和重复注解	373

11.1.3	注解声明	374
11.1.4	注解类型用途	375
11.1.5	明确接收者	376
11.2	定义注解	378
11.3	标准注解	380
11.3.1	编译相关的注解	381
11.3.2	资源管理相关的注解	382
11.3.3	元注解	382
11.4	运行时注解处理	384
11.5	源码级注解处理	387
11.5.1	注解处理器	388
11.5.2	语言模型API	388
11.5.3	使用注解生成源码	389
	练习	392
第12章	日期和时间API	395
12.1	时间线	396
12.2	本地日期	398
12.3	日期调整器	401
12.4	本地时间	402
12.5	时区时间	403
12.6	格式化和解析	406
12.7	应对遗留代码	409
	练习	411
第13章	国际化	413
13.1	本地化	414
13.1.1	指定本地化	414
13.1.2	默认locale	417
13.1.3	显示名称	418
13.2	数字格式化	418
13.3	货币符号	419
13.4	日期和时间格式化	420
13.5	排序和规格化	422
13.6	消息格式化	424
13.7	资源束	426
13.7.1	组织资源束	426
13.7.2	资源束类	428

13.8	字符集编码	429
13.9	首选项	430
	练习	432
第14章	编译与脚本	435
14.1	编译器API	435
14.1.1	调用编译器	436
14.1.2	启动编译任务	436
14.1.3	从内存读取源文件	437
14.1.4	向内存写入字节码	438
14.1.5	捕获诊断信息	440
14.2	脚本API	440
14.2.1	获取脚本引擎	440
14.2.2	绑定	441
14.2.3	重定向输入与输出	442
14.2.4	调用脚本函数和方法	443
14.2.5	编译脚本	444
14.3	Nashorn脚本引擎	445
14.3.1	从命令行运行Nashorn	445
14.3.2	调用get、set和重载方法	446
14.3.3	构造Java对象	447
14.3.4	JavaScript和Java中的字符串	449
14.3.5	数字	449
14.3.6	使用数组	450
14.3.7	列表与映射	451
14.3.8	lambda表达式	452
14.3.9	继承Java类与实现Java接口	453
14.3.10	异常	455
14.4	shell脚本与Nashorn	455
14.4.1	执行shell命令	456
14.4.2	字符串插值	456
14.4.3	脚本输入	457
	练习	458