

专门针对Java SE 8 (JDK 8)全面更新

Java: The Complete Reference, Ninth Edition



Java 8

编程参考官方教程(第9版)

[美] Herbert Schildt
战晓苏 江凌

著
译

清华大学出版社

Java 8 编程参考 官方教程

(第 9 版)

[美] Herbert Schildt 著

战晓苏 江凌 译

清华大学出版社

北 京

Herbert Schildt

Java: The Complete Reference, Ninth Edition

EISBN: 978-0-07-180855-2

Copyright © 2014 by McGraw-Hill Education.

All Rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including without limitation photocopying, recording, taping, or any database, information or retrieval system, without the prior written permission of the publisher.

This authorized Chinese translation edition is jointly published by McGraw-Hill Education and Tsinghua University Press Limited. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong, Macao SAR and Taiwan.

Copyright © 2015 by McGraw-Hill Education and Tsinghua University Press Limited.

版权所有。未经出版人事先书面许可,对本出版物的任何部分不得以任何方式或途径复制或传播,包括但不限于复印、录制、录音,或通过任何数据库、信息或可检索的系统。

本授权中文简体字翻译版由麦格劳-希尔(亚洲)教育出版公司和清华大学出版社有限公司合作出版。此版本经授权仅限在中华人民共和国境内(不包括香港特别行政区、澳门特别行政区和台湾)销售。

版权©2015 由麦格劳-希尔(亚洲)教育出版公司与清华大学出版社有限公司所有。

北京市版权局著作权合同登记号 图字: 01-2014-7616

本书封面贴有 McGraw-Hill Education 公司防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Java 8 编程参考官方教程: 第 9 版 / (美) 施密特(Schildt, H.) 著; 战晓苏, 江凌 译. —北京: 清华大学出版社, 2015

书名原文: Java: The Complete Reference, Ninth Edition

ISBN 978-7-302-38740-4

I. ①J… II. ①施… ②战… ③江… III. ①JAVA 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 284279 号

责任编辑: 王 军 李维杰

封面设计: 牛艳敏

责任校对: 成凤进

责任印制: 杨 艳

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

装 订 者: 三河市新茂装订有限公司

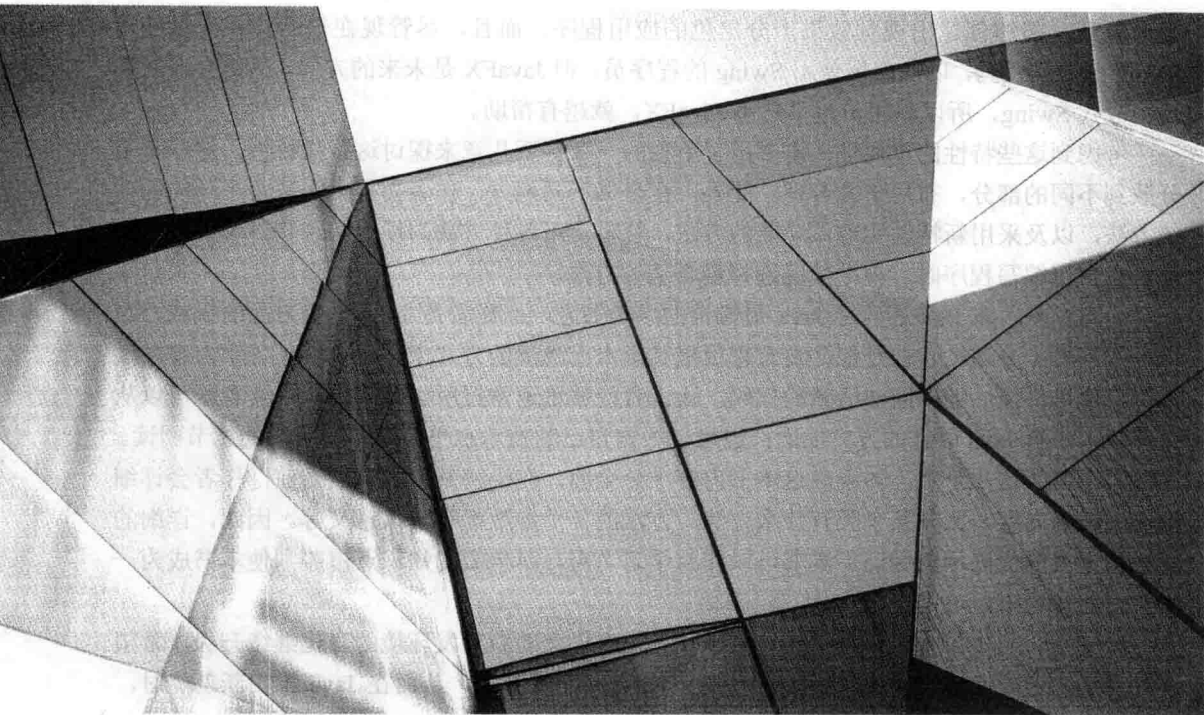
经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 80.25 字 数: 2003 千字

版 次: 2015 年 1 月第 1 版 印 次: 2015 年 1 月第 1 次印刷

印 数: 1~3000

定 价: 139.00 元



译者序

从诞生之初直至现在，Java 一直是最流行的编程语言之一。而在计算机世界里，变化是常态。几年的时间，计算机世界就可能发生天翻地覆的变化。

处理器处理能力的增长，网络带宽的增长，以及编程方法论的进步等，意味着如今的开发环境与几年前发生了不小的变化，遑论 20 年前。这么多年以来，Java 一直保持着旺盛的生命力，被众多公司采纳，受到众多程序员的喜爱，这自然与其不断地根据外界的变化调整更新自身的能力密切相关。

Java 的最新发布版本是 Java SE 8，对应的 Java 开发人员工具包为 JDK 8。相比 JDK 7，JDK 8 添加了几个新特性，比如极为重要的 lambda 表达式，以及 `java.util.stream` 包中提供的流 API。另外也在其他地方做了重要更新，比如允许接口方法具有默认实现。这些新的特性无疑会让程序员的工作变得十分方便，不过这也意味着程序员要改变固有的思维方式，在更适合使用新特性的地方积极运用它们。另外一个重要的改变在 GUI 框架方面。JavaFX 特性

强大灵活，能够创建出视觉效果十分出色的应用程序。而且，尽管现在仍然存在大量使用 Swing 框架的代码，以及习惯使用 Swing 的程序员，但 JavaFX 是未来的方向，最终会在新项目中取代 Swing，所以程序员越早熟悉 JavaFX，就越有帮助。

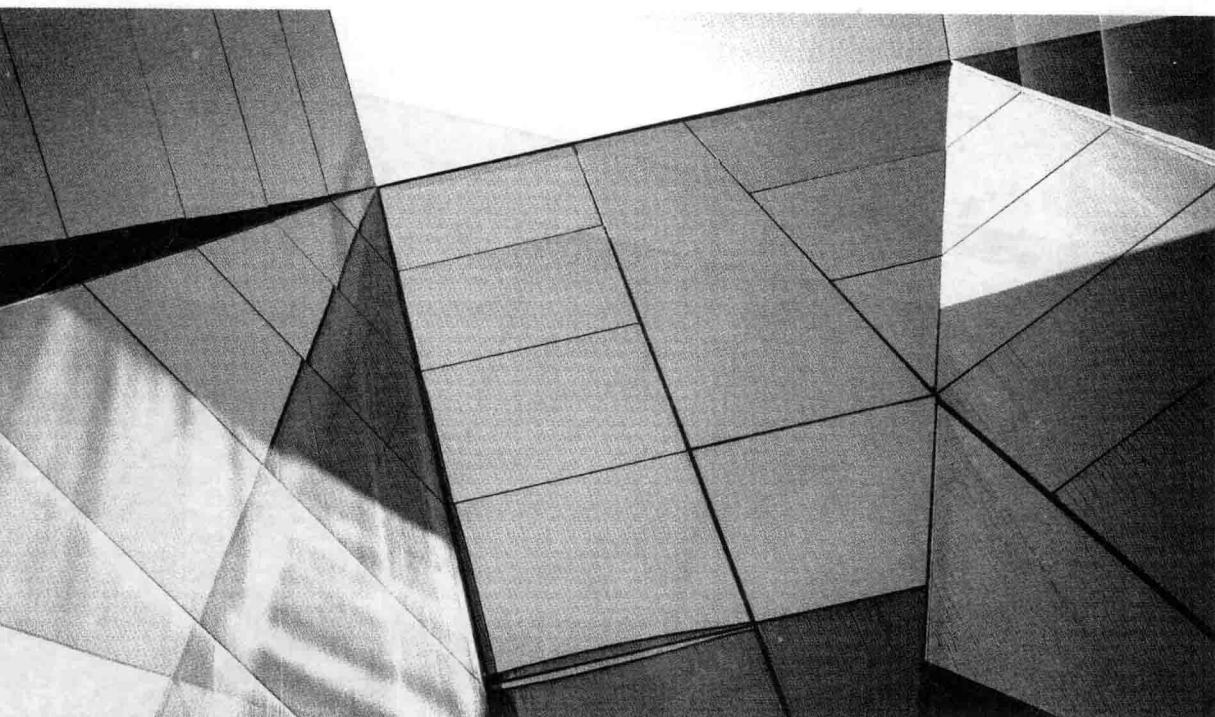
考虑到这些特性的重要性，本书用了单独的一章甚至几章来探讨这些新特性。这些章节分散到不同的部分，布局十分合理。此外，在全书不同地方，作者会指出实现某个功能的传统方法，以及采用新特性实现该方法的方法，解释使用各种方法的场合。这些讲解可以帮助读者在自己编写程序时，根据情况选择最合适的方法。

一直以来，本书都是学习 Java 最畅销的参考书之一。作者是 Java 语言方面的权威，对 Java 的理解十分透彻。但是他并没有采用极其学术而晦涩的方式来进行讲解。各章内容由浅入深，轻松易懂，辅以简洁清晰的代码，让读者能够迅速掌握所讲解的内容。读者既可以从头读起，了解 Java 语言的方方面面；也可以根据自己的技术水平，选择不熟悉的章节阅读。

“参考手册”这几个字，体现出这本书内容十分全面。在讲解某个知识点之前，作者会详细阐述相关的内容，比如某个类有什么方法，方法的各个参数都有什么意义等。因此，详细的解释和丰富的代码示例保证了本书可以作为学习之用，而全面的知识点覆盖则使本书成为一本实用的参考书，可常备案头。

Java 是一门生命力极强的语言，可以预见未来仍会随着开发环境的变化继续升级。希望读者通过本书，能够牢固地掌握 Java 的基础知识和思维方式，从而在 Java 推出新功能时，能够迅速掌握并加以运用！

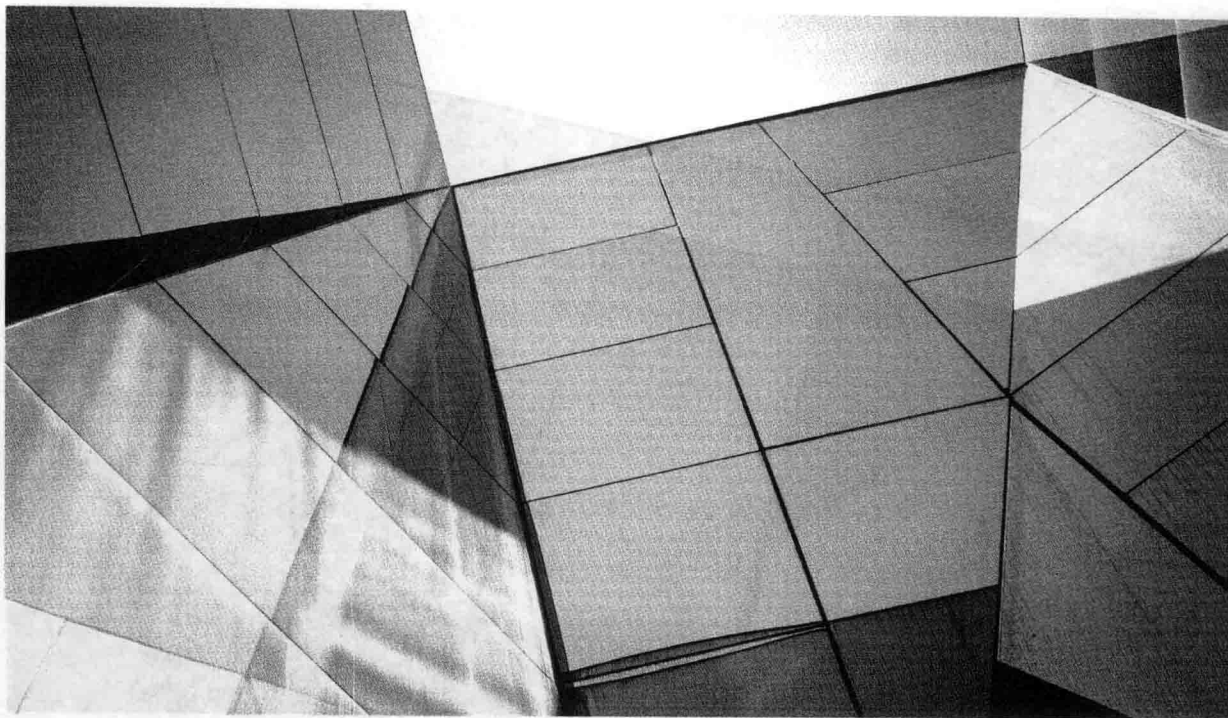
本书全部章节由战晓苏、江凌翻译，参与翻译活动的还有孔祥亮、陈跃华、杜思明、熊晓磊、曹汉鸣、陶晓云、王通、方峻、李小凤、曹晓松、蒋晓冬、邱培强、洪妍、李亮辉、高娟妮、曹小震、陈笑。对于这本经典之作，译者本着“诚惶诚恐”的态度，在翻译过程中力求“信、达、雅”，但是鉴于译者水平有限，错误和失误在所难免，如有任何意见和建议，请不吝指正。感激不尽！



作者简介

作者简介

Herbert Schildt 是一位畅销书作者，在几乎三十年的时间里，他撰写了大量关于编程的图书。Herbert 是 Java 语言的权威。他撰写的编程书籍在世界范围内销售了数百万册，并且已经被翻译成所有主要的外国语言。他撰写了大量 Java 方面的书籍，包括 *Java: A Beginner's Guide*、*Herb Schildt's Java Programming Cookbook* 和 *Swing: A Beginner's Guide*，还撰写了许多关于 C、C++ 和 C# 的图书。尽管对计算机的所有方面都感兴趣，但是他主要关注计算机语言，包括编译器、解释器以及机器人控制语言。他还对语言的标准化非常感兴趣。Schildt 获得了 Illinois 大学的本科和研究生学位。他的咨询办公室的电话是 (217) 586-4683，个人网站是 www.HerbSchildt.com。

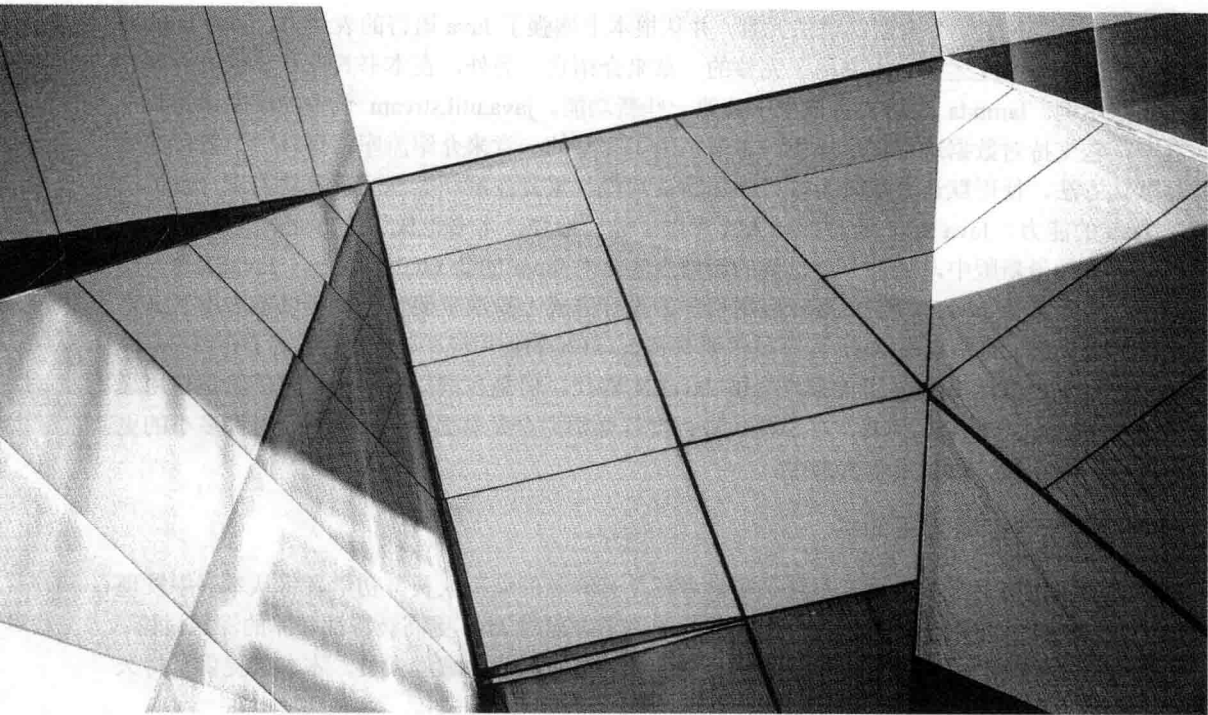


介绍

技术编辑简介

■ Danny Coward 博士在所有版本的 Java 平台上都工作过。他将 Java servlet 的定义引入了 Java EE 平台的第一个版本及后续版本，将 Web 服务引入了 Java ME 平台，并且主持 Java SE 7 的战略和规划设计。他发起了 JavaFX 技术，并且最近还设计了 Java WebSocket API，这是 Java EE 7 标准所做的程度最大的新增内容。他的经历丰富，包括从 Java 编码到与业界专家一起设计 API，到担任了几年 Java 社区进程执行委员会 (Java Community Process Executive Committee) 的成员，所以对于 Java 技术的多个方面有着独特、广泛的见解。另外，他还是 *Java WebSocket Programming* 以及即将出版的一本关于 Java EE 的图书的作者。Coward 博士从牛津大学获得了数学学士、硕士和博士学位。

www.it-ebooks.info



三内件本

前言

Java 是当今世界最重要、也是使用最广泛的计算机语言之一。而且，在多年之前它就已经拥有这一荣誉。与其他一些计算机语言随着时间的流逝、影响也逐渐减弱不同，Java 随着时间的推移反而变得更加强大。从首次发布开始，Java 就跃到了 Internet 编程的前沿。后续的每一个版本都进一步巩固了这一地位。如今，Java 依然是开发基于 Web 的应用程序的最佳选择。在现实世界中，很多应用都是使用 Java 开发的，Java 真的是非常重要！

Java 成功的一个关键原因是它的敏捷性。自从最初的 1.0 版发布以来，Java 不断地进行完善以适应编程环境和开发人员编程方式的变化。最重要的是，Java 不仅仅是在跟随潮流，更是在帮助创造潮流。Java 能够适应计算机世界快速变化的能力，是它一直并且仍将如此成功的关键因素。

本书自从 1996 年首次出版以来，已经经历了数次改版，每次改版都反映了 Java 的不断演化进程。这是本书的第 9 版，已经针对 Java SE 8 (JDK 8) 进行了升级。因为 Java SE 8 向 Java 语言添加了几个新特性，所以本版包含了大量新的内容。最重要的一个新增特性是 lambda

表达式，它引入了一个全新的语法元素，并从根本上增强了 Java 语言的表达力。因为 lambda 表达式的影响如此之大，本书用了完整的一章来介绍它。另外，在本书其他章节也可看到它的用法示例。lambda 表达式也催生了其他一些新功能。java.util.stream 包中的流库就是其中之一，它支持对数据进行管道操作。本书也用了完整的一章来介绍流库。另外一个新特性就是默认方法，使用默认方法可为接口添加默认功能。重复注解和类型注解等功能进一步扩展了 Java 的能力。Java SE 8 还对 Java API 库做了显著增强，本书也做了这方面的介绍。

本书的最新版中，另外一项重要的新增内容是对 Java 的新 GUI 框架——JavaFX——的介绍。因为预计 JavaFX 将在 Java 应用程序的设计方式上扮演重要角色，所以本书用了三章来介绍 JavaFX。简单地说，Java 程序员需要具备 JavaFX 的使用经验。本书还新增了介绍 Swing 菜单的一章。虽然 Swing 可能最终会被 JavaFX 取代，但是在撰写本书时，它仍然是使用最广泛的 Java GUI 框架。因此，对 Swing 进行展开介绍完全是必要的。最后，对许多小的更新方面的介绍被分散到了整本书中。

一本适合所有编程人员的书

本书面向所有开发人员，不管是初学者还是有经验的编程人员。初学者将从本书中发现每个主题的详细讨论，以及许多特别有帮助的例子。而对 Java 更高级特性和库的深入讨论，将会吸引有经验的编程人员。无论是对于初学者还是有经验的编程人员，本书都提供了持久的资源和方便实用的参考。

本书内容

本书是对 Java 语言的全面指导，描述了它的语法、关键字以及基本的编程原则，另外还介绍了 Java API 库的重要部分。本书分为 5 部分，每部分关注 Java 开发环境的不同方面。

第 I 部分是对 Java 语言的深入辅导。该部分从基础知识开始讲解，包括数据类型、运算符、控制语句以及类等。然后本部分介绍继承、包、接口、异常处理以及多线程。之后介绍注解、枚举、自动装箱以及泛型等内容。本部分还介绍了 I/O 以及 applet。本部分最后一章介绍了 lambda 表达式。刚才介绍过，lambda 表达式是 Java SE 8 中最重要的新增特性。

第 II 部分介绍 Java 标准 API 库的关键内容。本部分的主题包括字符串、I/O、网络、标准实用工具、集合框架、applet、AWT、事件处理、图像、并发编程(包括 Fork/Join 框架)、正则表达式和新增的流库。

第 III 部分用三章内容介绍 Swing。

第 IV 部分用三章内容介绍 JavaFX。

第 V 部分包含两章，这两章展示了 Java 的实际应用。本部分的第 37 章介绍了 Java Bean，第 38 章介绍了 servlet。

在线资源

请记住，本书中全部示例程序的源代码都可以从 www.oraclepressbooks.com 免费下载。

致谢

在此我要特别感谢 Patrick Naughton、Joe O'Neil 和 Danny Coward。

Patrick Naughton 是 Java 语言的创立者之一，他还帮助我编写了本书的第 1 版。例如，本书第 20、22 和 27 章的大部分材料最初都是由 Patrick 提供的，他还为本书做出了其他许多贡献。他的洞察力、专业知识和活力都对本书的成功付梓贡献极大。

在准备本书的第 2 版和第 3 版的过程中，Joe O'Neil 为目前本书第 30、32、37 和 38 章的内容提供了原始素材。Joe 对我的好几本书都有帮助，并且他提供的帮助一直都是最高质量的。

Danny Coward 是本书第 9 版的技术编辑。Danny 对我的好几本书都有贡献，他的忠告、洞察力和建议都有巨大的价值，对此表示非常感谢。

如何进一步学习

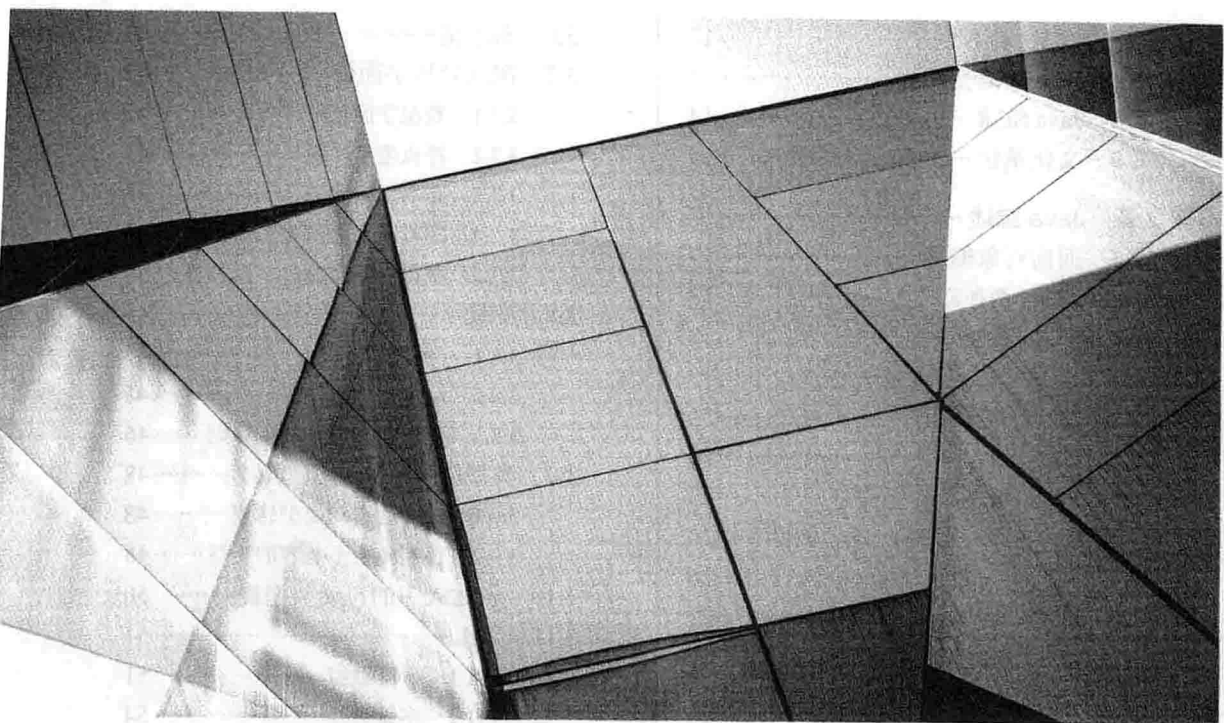
《Java 8 编程参考官方教程(第 9 版)》为读者开启了 Herb Schildt Java 编程图书系列的大门。下面是其他一些您可能感兴趣的图书：

Herb Schildt's Java Programming Cookbook

Java: A Beginner's Guide

Swing: A Beginner's Guide

The Art of Java



目 录

第 I 部分 Java 语言

第 1 章 Java 的历史和演变	3
1.1 Java 的家世	4
1.1.1 现代编程语言的诞生: C 语言	4
1.1.2 C++: 下一个阶段	5
1.1.3 Java 出现的时机已经成熟	6
1.2 Java 的诞生	6
1.3 Java 改变 Internet 的方式	8
1.3.1 Java applet	8

1.3.2 安全性	8
1.3.3 可移植性	8
1.4 Java 的魔力: 字节码	9
1.5 servlet: 服务器端的 Java	10
1.6 Java 的关键特性	10
1.6.1 简单性	10
1.6.2 面向对象	11
1.6.3 健壮性	11
1.6.4 多线程	11
1.6.5 体系结构中立	11
1.6.6 解释执行和高性能	12
1.6.7 分布式	12

1.6.8 动态性	12	3.6 布尔型	41
1.7 Java 的演变历程	12	3.7 深入分析字面值	42
1.8 Java SE 8	14	3.7.1 整型字面值	42
1.9 文化革新	15	3.7.2 浮点型字面值	43
第 2 章 Java 综述	17	3.7.3 布尔型字面值	43
2.1 面向对象编程	17	3.7.4 字符型字面值	43
2.1.1 两种范式	18	3.7.5 字符串字面值	44
2.1.2 抽象	18	3.8 变量	45
2.1.3 OOP 三原则	19	3.8.1 变量的声明	45
2.2 第一个简单程序	23	3.8.2 动态初始化	45
2.2.1 输入程序	23	3.8.3 变量的作用域和生存期	46
2.2.2 编译程序	23	3.9 类型转换和强制类型转换	48
2.2.3 深入分析第一个示例程序	24	3.9.1 Java 的自动类型转换	48
2.3 第二个简短程序	26	3.9.2 强制转换不兼容的类型	48
2.4 两种控制语句	27	3.10 表达式中的自动类型提升	50
2.4.1 if 语句	27	3.11 数组	51
2.4.2 for 循环	29	3.11.1 一维数组	51
2.5 使用代码块	30	3.11.2 多维数组	54
2.6 词汇问题	31	3.11.3 另一种数组声明语法	58
2.6.1 空白符	32	3.12 关于字符串的一些说明	58
2.6.2 标识符	32	3.13 向 C/C++ 程序员提供指针 方面的说明	59
2.6.3 字面值	32		
2.6.4 注释	32	第 4 章 运算符	61
2.6.5 分隔符	32	4.1 算术运算符	61
2.6.6 Java 关键字	33	4.1.1 基本算术运算符	62
2.7 Java 类库	34	4.1.2 求模运算符	63
第 3 章 数据类型、变量和数组	35	4.1.3 算术与赋值复合运算符	64
3.1 Java 是强类型化的语言	35	4.1.4 自增与自减运算符	65
3.2 基本类型	36	4.2 位运算符	66
3.3 整型	36	4.2.1 位逻辑运算符	67
3.3.1 byte	37	4.2.2 左移	70
3.3.2 short	37	4.2.3 右移	71
3.3.3 int	37	4.2.4 无符号右移	72
3.3.4 long	37	4.2.5 位运算符与赋值的组合	74
3.4 浮点型	38	4.3 关系运算符	74
3.4.1 float	38	4.4 布尔逻辑运算符	75
3.4.2 double	39	4.5 赋值运算符	77
3.5 字符型	39	4.6 “?” 运算符	78

4.7	运算符的优先级	79
4.8	使用圆括号	80
第 5 章	控制语句	81
5.1	Java 的选择语句	81
5.1.1	if 语句	82
5.1.2	switch 语句	84
5.2	迭代语句	89
5.2.1	while 语句	89
5.2.2	do-while 语句	91
5.2.3	for 语句	93
5.2.4	for 循环的 for-each 版本	97
5.2.5	嵌套的循环	102
5.3	跳转语句	103
5.3.1	使用 break 语句	103
5.3.2	使用 continue 语句	107
5.3.3	return 语句	108
第 6 章	类	111
6.1	类的基础知识	112
6.1.1	类的一般形式	112
6.1.2	一个简单的类	113
6.2	声明对象	115
6.3	为对象引用变量赋值	117
6.4	方法	118
6.4.1	为 Box 类添加方法	118
6.4.2	返回值	120
6.4.3	添加带参数的方法	121
6.5	构造函数	123
6.6	this 关键字	126
6.7	垃圾回收	127
6.8	finalize()方法	128
6.9	堆栈类	128
第 7 章	方法和类的深入分析	131
7.1	重载方法	131
7.2	将对象用作参数	136
7.3	参数传递的深入分析	139
7.4	返回对象	141
7.5	递归	142

7.6	访问控制	144
7.7	理解 static	147
7.8	final 介绍	149
7.9	重新审视数组	149
7.10	嵌套类和内部类	151
7.11	String 类介绍	154
7.12	使用命令行参数	156
7.13	varargs: 可变长度参数	157
7.13.1	重载 varargs 方法	160
7.13.2	varargs 方法与模糊性	161
第 8 章	继承	165
8.1	继承的基础知识	165
8.1.1	成员访问与继承	167
8.1.2	一个更实际的例子	168
8.1.3	超类变量可以引用子类对象	170
8.2	使用 super 关键字	171
8.2.1	使用 super 调用超类的构造函数	171
8.2.2	super 的另一种用法	175
8.3	创建多级继承层次	176
8.4	构造函数的调用时机	179
8.5	方法重写	180
8.6	动态方法调度	183
8.6.1	重写方法的目的	184
8.6.2	应用方法重写	185
8.7	使用抽象类	186
8.8	在继承中使用 final 关键字	189
8.8.1	使用 final 关键字阻止重写	189
8.8.2	使用 final 关键字阻止继承	190
8.9	Object 类	190
第 9 章	包和接口	193
9.1	包	194
9.1.1	定义包	194
9.1.2	包查找与 CLASSPATH	195
9.1.3	一个简短的包示例	195
9.2	访问保护	196
9.3	导入包	200

9.4 接口	202
9.4.1 定义接口	202
9.4.2 实现接口	203
9.4.3 嵌套接口	206
9.4.4 应用接口	207
9.4.5 接口中的变量	210
9.4.6 接口可以扩展	212
9.5 默认接口方法	213
9.5.1 默认方法的基础知识	214
9.5.2 一个更加实用的例子	215
9.5.3 多级继承的问题	216
9.6 在接口中使用静态方法	217
9.7 关于包和接口的最后说明	218
第 10 章 异常处理	219
10.1 异常处理的基础知识	219
10.2 异常类型	220
10.3 未捕获的异常	221
10.4 使用 try 和 catch	222
10.5 多条 catch 子句	224
10.6 嵌套的 try 语句	225
10.7 throw	228
10.8 throws	229
10.9 finally	230
10.10 Java 的内置异常	232
10.11 创建自己的异常子类	233
10.12 链式异常	235
10.13 3 个近期添加的异常特性	237
10.14 使用异常	238
第 11 章 多线程编程	239
11.1 Java 线程模型	240
11.1.1 线程优先级	241
11.1.2 同步	241
11.1.3 消息传递	242
11.1.4 Thread 类和 Runnable 接口	242
11.2 主线程	243
11.3 创建线程	244
11.3.1 实现 Runnable 接口	245

11.3.2 扩展 Thread 类	247
11.3.3 选择一种创建方式	248
11.4 创建多个线程	248
11.5 使用 isAlive() 和 join() 方法	249
11.6 线程优先级	252
11.7 同步	253
11.7.1 使用同步方法	253
11.7.2 synchronized 语句	255
11.8 线程间通信	257
11.9 挂起、恢复与停止线程	263
11.10 获取线程的状态	266
11.11 使用多线程	267
第 12 章 枚举、自动装箱与注解	
(元数据)	269
12.1 枚举	269
12.1.1 枚举的基础知识	270
12.1.2 values() 和 valueOf() 方法	272
12.1.3 Java 枚举是类类型	273
12.1.4 枚举继承自 Enum 类	275
12.1.5 另一个枚举示例	277
12.2 类型封装器	279
12.2.1 Character 封装器	279
12.2.2 Boolean 封装器	279
12.2.3 数值类型封装器	280
12.3 自动装箱	281
12.3.1 自动装箱与方法	281
12.3.2 表达式中发生的 自动装箱/拆箱	282
12.3.3 布尔型和字符型数值的 自动装箱/拆箱	284
12.3.4 自动装箱/拆箱有助于 防止错误	285
12.3.5 一些警告	285
12.4 注解(元数据)	286
12.4.1 注解的基础知识	286
12.4.2 指定保留策略	287
12.4.3 在运行时使用反射获取 注解	288

12.4.4	AnnotatedElement 接口	293	14.4	泛型类的一般形式	352
12.4.5	使用默认值	293	14.5	有界类型	352
12.4.6	标记注解	295	14.6	使用通配符参数	355
12.4.7	单成员注解	296	14.7	创建泛型方法	363
12.4.8	内置注解	297	14.8	泛型接口	366
12.5	类型注解	299	14.9	原始类型与遗留代码	368
12.6	重复注解	303	14.10	泛型类层次	370
第 13 章	I/O、applet 以及其他主题	307	14.10.1	使用泛型超类	370
13.1	I/O 的基础知识	308	14.10.2	泛型子类	372
13.1.1	流	308	14.10.3	泛型层次中的运行时 类型比较	374
13.1.2	字节流和字符流	308	14.10.4	强制转换	376
13.1.3	预定义流	310	14.10.5	重写泛型类的方法	376
13.2	读取控制台输入	311	14.11	泛型的类型推断	378
13.2.1	读取字符	311	14.12	擦除	379
13.2.2	读取字符串	312	14.13	模糊性错误	381
13.3	向控制台写输出	314	14.14	使用泛型的一些限制	382
13.4	PrintWriter 类	314	14.14.1	不能实例化类型参数	382
13.5	读/写文件	315	14.14.2	对静态成员的一些 限制	383
13.6	自动关闭文件	321	14.14.3	对泛型数组的一些 限制	383
13.7	applet 的基础知识	325	14.14.4	对泛型异常的限制	384
13.8	transient 和 volatile 修饰符	327	第 15 章	lambda 表达式	385
13.9	使用 instanceof 运算符	328	15.1	lambda 表达式简介	386
13.10	strictfp	330	15.1.1	lambda 表达式的 基础知识	386
13.11	本地方法	331	15.1.2	函数式接口	387
13.12	使用 assert	334	15.1.3	几个 lambda 表达式 示例	388
13.13	静态导入	337	15.2	块 lambda 表达式	391
13.14	通过 this() 调用重载的 构造函数	339	15.3	泛型函数式接口	393
13.15	紧凑 API 配置文件	342	15.4	作为参数传递 lambda 表达式	395
第 14 章	泛型	343	15.5	lambda 表达式与异常	398
14.1	什么是泛型	344	15.6	lambda 表达式和变量捕获	399
14.2	一个简单的泛型示例	344	15.7	方法引用	400
14.2.1	泛型只使用引用类型	348	15.7.1	静态方法的方法引用	400
14.2.2	基于不同类型参数的 泛型类型是不同的	348	15.7.2	实例方法的方法引用	401
14.2.3	泛型提升类型安全性的 原理	348			
14.3	带两个类型参数的泛型类	350			

15.7.3	泛型中的方法引用	405
15.8	构造函数引用	408
15.9	预定义的函数式接口	412

第 II 部分 Java 库

第 16 章 字符串处理 417

16.1	String 类的构造函数	418
16.2	字符串的长度	420
16.3	特殊的字符串操作	420
16.3.1	字符串字面值	421
16.3.2	字符串连接	421
16.3.3	字符串和其他数据类型 的连接	421
16.3.4	字符串转换和 toString() 方法	422
16.4	提取字符	423
16.4.1	charAt()	424
16.4.2	getChars()	424
16.4.3	getBytes()	424
16.4.4	toCharArray()	425
16.5	比较字符串	425
16.5.1	equals()和 equalsIgnoreCase()	425
16.5.2	regionMatches()	426
16.5.3	startsWith()和 endsWith()	426
16.5.4	equals()与==	427
16.5.5	compareTo()	427
16.6	查找字符串	429
16.7	修改字符串	430
16.7.1	substring()	430
16.7.2	concat()	431
16.7.3	replace()	432
16.7.4	trim()	432
16.8	使用 valueOf()转换数据	433
16.9	改变字符串中字符的大小写	434
16.10	连接字符串	434
16.11	其他 String 方法	435
16.12	StringBuffer 类	436

16.12.1	StringBuffer 类的 构造函数	437
16.12.2	length()与 capacity()	437
16.12.3	ensureCapacity()	438
16.12.4	setLength()	438
16.12.5	charAt()与 setCharAt()	438
16.12.6	getChars()	439
16.12.7	append()	439
16.12.8	insert()	440
16.12.9	reverse()	440
16.12.10	delete()与 deleteCharAt()	441
16.12.11	replace()	441
16.12.12	substring()	442
16.12.13	其他 StringBuffer 方法	442
16.13	StringBuilder 类	443

第 17 章 探究 java.lang 445

17.1	基本类型封装器	446
17.1.1	Number	446
17.1.2	Double 与 Float	447
17.1.3	理解 isInfinite()与 isNaN()	450
17.1.4	Byte、Short、Integer 和 Long	451
17.1.5	Character	459
17.1.6	对 Unicode 代码点的 附加支持	462
17.1.7	Boolean	463
17.2	Void 类	464
17.3	Process 类	464
17.4	Runtime 类	465
17.4.1	内存管理	466
17.4.2	执行其他程序	467
17.5	ProcessBuilder 类	469
17.6	System 类	471
17.6.1	使用 currentTimeMillis() 计时程序的执行	472

17.6.2	使用 arraycopy()	473
17.6.3	环境属性	474
17.7	Object 类	474
17.8	使用 clone()方法和 Cloneable 接口	475
17.9	Class 类	477
17.10	ClassLoader 类	480
17.11	Math 类	481
17.11.1	三角函数	481
17.11.2	指数函数	481
17.11.3	舍入函数	482
17.11.4	其他数学方法	483
17.12	StrictMath 类	485
17.13	Compiler 类	485
17.14	Thread 类、ThreadGroup 类 和 Runnable 接口	485
17.14.1	Runnable 接口	485
17.14.2	Thread 类	486
17.14.3	ThreadGroup 类	488
17.15	ThreadLocal 和 InheritableThreadLocal 类	492
17.16	Package 类	492
17.17	RuntimePermission 类	494
17.18	Throwable 类	494
17.19	SecurityManager 类	494
17.20	StackTraceElement 类	494
17.21	Enum 类	495
17.22	ClassValue 类	496
17.23	CharSequence 接口	496
17.24	Comparable 接口	496
17.25	Appendable 接口	497
17.26	Iterable 接口	497
17.27	Readable 接口	498
17.28	AutoCloseable 接口	498
17.29	Thread.UncaughtExceptionHandler 接口	498
17.30	java.lang 子包	499
17.30.1	java.lang.annotation	499
17.30.2	java.lang.instrument	499

17.30.3	java.lang.invoke	499
17.30.4	java.lang.management	499
17.30.5	java.lang.ref	499
17.30.6	java.lang.reflect	499
第 18 章	java.util 第 1 部分:	
	集合框架	501
18.1	集合概述	503
18.2	JDK 5 对集合框架的修改	504
18.2.1	泛型从根本上改变了 集合框架	504
18.2.2	自动装箱使得使用基本 类型更加容易	504
18.2.3	for-each 风格的 for 循环	505
18.3	集合接口	505
18.3.1	Collection 接口	506
18.3.2	List 接口	508
18.3.3	Set 接口	509
18.3.4	SortedSet 接口	509
18.3.5	NavigableSet 接口	510
18.3.6	Queue 接口	511
18.3.7	Deque 接口	512
18.4	集合类	514
18.4.1	ArrayList 类	515
18.4.2	LinkedList 类	518
18.4.3	HashSet 类	520
18.4.4	LinkedHashSet 类	521
18.4.5	TreeSet 类	521
18.4.6	PriorityQueue 类	523
18.4.7	ArrayDeque 类	523
18.4.8	EnumSet 类	524
18.5	通过迭代器访问集合	525
18.5.1	使用迭代器	527
18.5.2	使用 for-each 循环替代 迭代器	528
18.6	Splititerator	529
18.7	在集合中存储用户定义类	532
18.8	RandomAccess 接口	533
18.9	使用映射	534