



Java领域最有影响力和价值的著作之一，与《Java编程思想》齐名，
10余年全球畅销不衰，广受好评

针对Java SE 9、10、11全面更新，系统全面讲解Java语言的核心概
念、语法、重要特性和开发方法，包含大量案例，实践性强



Java 核心技术 卷 I

基础知识 (原书第11版)

Core Java Volume I—Fundamentals
(Eleventh Edition)

[美] 凯·S. 霍斯特曼 (Cay S. Horstmann) 著
林琪 苏钰涵 等译



机械工业出版社
China Machine Press



Java

核心技术 卷 I

基础知识 (原书第11版)

Core Java Volume I—Fundamentals
(Eleventh Edition)

[美] 凯·S. 霍斯特曼 (Cay S. Horstmann) 著
林琪 苏钰涵 等译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Java 核心技术 卷 I 基础知识 (原书第 11 版)/(美) 凯·S. 霍斯特曼 (Cay S. Horstmann) 著; 林琪等译. —北京: 机械工业出版社, 2019.9 (2020.5 重印)

(Java 核心技术系列)

书名原文: Core Java, Volume I—Fundamentals (Eleventh Edition)

ISBN 978-7-111-63666-3

I. J… II. ①凯… ②林… III. JAVA 语言—程序设计 IV. TP312.8

中国版本图书馆 CIP 数据核字 (2019) 第 198657 号

本书版权登记号: 图字 01-2018-8484

Authorized translation from the English language edition, entitled *Core Java, Volume I—Fundamentals (Eleventh Edition)*, ISBN: 978-0-13-516630-7, by Cay S. Horstmann, published by Pearson Education, Inc., Copyright © 2019 Pearson Education Inc., Portions copyright © 1996-2013 Oracle and/or its affiliates.

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 China Machine Press, Copyright © 2020.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括香港、澳门特别行政区及台湾地区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

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

Java 核心技术 卷 I 基础知识 (原书第 11 版)

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 关 敏

责任校对: 李秋荣

印 刷: 北京诚信伟业印刷有限公司

版 次: 2020 年 5 月第 1 版第 4 次印刷

开 本: 186mm×240mm 1/16

印 张: 40.75

书 号: ISBN 978-7-111-63666-3

定 价: 149.00 元

客服电话: (010) 88361066 88379833 68326294

投稿热线: (010) 88379604

华章网站: www.hzbook.com

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

内容简介

本书由拥有20多年教学与研究经验的资深Java技术专家撰写（获Jolt大奖），是程序员的首选Java指南。本版针对Java SE 9、10和11全面更新。

全书共12章。第1章概述Java语言的特色功能；第2章讲解如何下载和安装JDK以及本书的程序示例；第3章开始讨论Java语言，包括变量、循环和简单的函数；第4章介绍面向对象两大基石中最重要的一个概念——封装，以及Java语言实现封装的机制，即类与方法；第5章介绍面向对象的另一个重要概念——继承，继承使程序员可以使用现有的类，并根据需要进行修改；第6章展示如何使用Java的接口、lambda表达式和内部类；第7章讨论异常处理，并给出大量实用的调试技巧；第8章概要介绍泛型程序设计，使程序更可读、更安全；第9章讨论Java平台的集合框架，以及如何充分利用内建的标准集合；第10章介绍GUI程序设计（包括如何建立窗口、如何在窗口中绘图、如何利用几何图形绘图、如何采用多种字体格式化文本，以及如何显示图像）和编写代码来响应事件（如鼠标点击事件和按键事件）；第11章详细讨论Swing GUI工具包，以建立跨平台的图形用户界面，包括各种按钮、文本组件、边框、滑块、列表框、菜单以及对话框的有关内容；第12章讨论并发。本书最后还有一个附录，其中列出了Java语言的保留字。

作者简介

凯·S.霍斯特曼
(Cay S. Horstmann)

圣何塞州立大学计算机科学系教授、Java的倡导者。他是《Java核心技术》两卷本的作者，并著有《Core Java SE 9 for the Impatient, Second Edition》(Addison-Wesley 2018)和《Scala for the Impatient, Second Edition》(Addison-Wesley, 2017)。他还为专业程序员和计算机科学专业的学生撰写过数十本其他图书。



华章图书

一本打开的书，
一扇开启的门，
通向科学殿堂的阶梯，
托起一流人才的基石。

译者序

书写 Java 传奇的 Sun 公司曾经堪称“日不落”帝国，但服务器市场的萎缩让这个声名赫赫的庞大帝国从蓬勃走向没落。在 2009 年被 Oracle 公司收购之后，Sun 公司逐渐淡出了人们的视线，而与此同时，我们也在很长一段时间内没能看到 Java 当初活跃的身影。

Java 就这样退出历史舞台了吗？当然不是！Sun 公司从 2006 年 12 月发布 Java 6 后，经过 5 年多的不懈努力，在 2011 年 7 月底发布了 Java 7 正式版。3 年后，被冠名为“跳票王”的 Oracle 终于发布了 Java 8 的正式版。又是 3 年后，Java 9 发布，而且从 2018 年开始，为了更快地引入新特性，每 6 个月就会发布一个 Java 版本，目前已经发布到 Java 11 版。

值得一提的是，伴随着 Java 的成长，《Java 核心技术》也从第 1 版到第 10 版一路走来，得到了广大 Java 程序设计人员的青睐，成为一本畅销不衰的 Java 经典图书。2019 年，针对 Java SE 9、10 和 11，《Java 核心技术》第 11 版新鲜出炉，这一版有了大幅的修订和更新，不仅补充了 Java 新版本的最新特性，还对之前比较晦涩的部分做了删改，用更清晰明了的示例加以解释，特别是对内容结构做了调整，从而更有利于读者学习和应用。它将续写从前的辉煌，使人们能及时跟上 Java 前进的脚步。

本书由林琪、苏钰涵等主译，耿文东、丁哲峰、王鹏、汪荣峰、陈浩光、管清波、邓友义、肖斌、庄锦山、黄文清、戴增刚、陈凌云、张志威、张守玉、宋芳参与了全书的修改整理，并完善了关键部分的翻译。全体人员共同完成了本书的翻译工作。特别感谢之前版本的译者叶乃文、邝劲筠和杜永萍，他们的辛勤工作为新版本的翻译奠定了很好的基础。

书中文字与内容力求忠实原著，不过由于译者水平有限，译文肯定有不当之处，敬请批评指正。

译者

2019 年 5 月于北京

前 言

致读者

1995 年年底，Java 语言在 Internet 舞台一亮相便名声大噪。原因在于它将有成为连接用户与信息的万能胶，而不论这些信息来自 Web 服务器、数据库、信息提供商，还是任何其他可以想象的渠道。事实上，就发展前景而言，Java 的地位是独一无二的。它是一种完全可信赖的程序设计语言，并得到了广泛认可。其固有的可靠性与安全性不仅令 Java 程序员放心，也令使用 Java 程序的用户放心。Java 内建了对网络编程、数据库连接和并发等高级程序设计任务的支持。

1995 年以来，已经发布了 Java 开发工具包（Java Development Kit）的 11 个主要版本。在过去的 20 年中，应用程序编程接口（API）已经从 200 个类扩展到超过 4000 个类。现在这些 API 覆盖了用户界面构建、数据库管理、国际化、安全性以及 XML 处理等各个不同的领域。

本书是《Java 核心技术》第 11 版的卷 I。自《Java 核心技术》出版以来，每个新版本都尽可能快地跟上 Java 开发工具包发展的步伐，而且每一版都重新改写了部分内容，以便利用 Java 的最新特性。这一版经过修订，反映了 Java 标准版（Java SE）9、10 和 11 的特性。

与本书前几版一样，这一版仍然将读者群定位为那些打算将 Java 应用到实际工程项目中的程序设计人员。本书假设读者是一名具有程序设计语言（除 Java 之外）坚实背景知识的程序员，并且不希望书中充斥着玩具式的示例（诸如，烤面包机、动物园的动物或神经质的跳动文本）。这些内容绝对不会在本书中出现。本书的目标是让读者充分理解书中介绍的 Java 语言及 Java 类库的相关特性，而不会产生任何误解。

在本书中，我们选用大量的示例代码演示所讨论的每一个语言特性和类库特性。我们有意使用简单的示例程序以突出重点，然而，其中的大部分既不是赝品也没有偷工减料。它们将成为读者自己编写代码的良好开端。

我们假定读者愿意（甚至渴望）学习 Java 提供的所有高级特性。例如，本书将详细介绍下列内容：

- 面向对象程序设计
- 反射与代理
- 接口与内部类
- 异常处理
- 泛型程序设计
- 集合框架

- 事件监听器模型
- 图形用户界面设计
- 并发

随着 Java 类库的爆炸式增长，只用一卷无法涵盖程序员需要了解的所有 Java 特性。因此，我们决定将本书分为两卷。卷 I（本书）集中介绍 Java 语言的基本概念以及用户界面程序设计的基础知识。卷 II（高级特性）涉及企业特性以及高级的用户界面程序设计，其中详细讨论下列内容：

- 流 API
- 文件处理与正则表达式
- 数据库
- XML 处理
- 注解
- 国际化
- 网络编程
- 高级 GUI 组件
- 高级图形
- 原生方法

本书中难免出现错误和不准确之处。我们很想知道这些错误，当然，也希望同一个问题只被告知一次。我们在网页 <http://horstmann.com/corejava> 中以列表的形式给出了常见的问题、bug 修正和解决方法。在勘误页（建议先阅读一遍）最后附有用来报告 bug 并提出修改意见的表单。如果我们不能回答每一个问题或没有及时回复，请不要失望。我们会认真地阅读所有的来信，感谢您的建议使本书后续的版本更清晰、更有指导价值。

关于本书

第 1 章概述 Java 与其他程序设计语言不同的功能。解释这种语言的设计初衷，以及在哪些方面达到了预期。然后，简要叙述 Java 诞生和发展的历史。

第 2 章详细论述如何下载和安装 JDK 以及本书的程序示例。然后，通过编译和运行 3 个典型的 Java 程序（一个控制台应用、一个图形应用、一个 applet），指导读者使用简易的 JDK、启用 Java 的文本编辑器以及一个 Java IDE。

第 3 章开始讨论 Java 语言。这一章涉及的基础知识有变量、循环以及简单的函数。对于 C 或 C++ 程序员来说，学习这一章的内容将会感觉一帆风顺，因为这些语言特性的语法本质上与 C 语言相同。对于没有 C 语言程序设计背景，但使用过其他程序设计语言（如 Visual Basic）的程序员来说，仔细阅读这一章是非常必要的。

面向对象程序设计（Object-Oriented Programming, OOP）是当今程序设计的主流，而 Java 是一种面向对象的语言。第 4 章将介绍面向对象两大基石中最重要的一個——封装概念，以及 Java 语言实现封装的机制，即类与方法。除了 Java 语言规则之外，还对如何完成

合理的 OOP 设计给出了忠告。最后，介绍奇妙的 javadoc 工具，它将代码注释转换为一组包含超链接的网页。熟悉 C++ 的程序员可以快速地浏览这一章，而没有面向对象程序设计背景的程序员在进一步学习 Java 之前应当先花一些时间了解 OOP 的有关概念。

类与封装仅仅是 OOP 中的一部分，第 5 章将介绍另一部分——继承。继承使程序员可以使用现有的类，并根据需要进行修改。这是 Java 程序设计中的一个基础技术。Java 中的继承机制与 C++ 的继承机制十分相似。重申一次，C++ 程序员可以只关注两种语言的不同之处。

第 6 章展示如何使用 Java 的接口。接口可以让你的理解超越第 5 章中的简单继承模型。掌握接口可以充分获得 Java 的完全面向对象的程序设计能力。介绍接口之后，我们将转而介绍 lambda 表达式 (lambda expression)，这是一种简洁的方法，用来表述可以在以后某个时间点执行的代码块。本章还将介绍 Java 的一个有用的技术特性——内部类。

第 7 章讨论异常处理 (exception handling)，这是 Java 的一种健壮机制，用于处理正常运行的程序可能出现意外的情况。异常提供了一种将正常处理代码与错误处理代码分开的有效手段。当然，即使程序能够处理所有异常条件，仍然有可能无法按照预计的方式工作。这一章的后半部分将给出大量实用的调试技巧。

第 8 章概要介绍泛型程序设计。泛型程序设计可以让程序更可读、更安全。我们会展示如何使用强类型机制，而舍弃不好看也不安全的强制类型转换，以及如何处理与老版本 Java 兼容所带来的复杂问题。

第 9 章讨论的是 Java 平台的集合框架。如果希望收集多个对象并在以后获取这些对象，就应当使用集合，这是这种情况下最适用的做法，不要简单地把这些元素放在一个数组中。这一章会介绍如何充分利用内建的标准集合。

第 10 章介绍 GUI 程序设计。我们会讨论如何建立窗口、如何在窗口中绘图、如何利用几何图形绘图、如何采用多种字体格式化文本，以及如何显示图像。接下来你会了解如何编写代码来响应事件，如鼠标点击事件或按键事件。

第 11 章详细讨论 Swing GUI 工具包。Swing 工具包允许建立跨平台的图形用户界面。在这里你会了解各种按钮、文本组件、边框、滑块、列表框、菜单以及对话框的有关内容。不过，一些更高级的组件会在卷 II 中讨论。


第 12 章是本书的最后一章，这一章将讨论并发，并发能够让程序任务并行执行。在当今这个时代，大多数处理器都有多个内核，你通常希望这些内核都在工作，并发是 Java 技术的一个重要而且令人振奋的应用。


还有关于 JavaFX 的额外的一章，其中会给出 JavaFX 的简要介绍，这是开发桌面应用的一个现代 GUI 工具包。如果你看的是本书的纸质版，可以从本书英文版网站 <http://horstmann.com/corejava> 或华章公司网站 <http://www.hzbook.com> 下载这一章。


附录列出了 Java 语言的保留字。


约定

与很多计算机图书一样，本书使用等宽字体 (monospace type) 表示计算机代码。


 **注释：**“注释”信息会用这样的图标标志。

 **提示：**“提示”信息会用这样的图标标志。

 **警告：**对于可能出现的危险，我们用这样的图标做出警示。

 **C++ 注释：**在本书中有许多用来解释 Java 与 C++ 之间差别的 C++ 注释。没有 C++ 程序设计背景，或者不擅长 C++ 程序设计而把它当作一场噩梦不愿再想起的程序员可以跳过这些注释。

Java 提供了一个很大的程序设计库，即应用程序编程接口。第一次使用某个 API 调用时，我们会在该节的结尾给出它的概要描述。这些描述有些不太正式，但我们希望能够比联机 API 文档提供更多的信息。类、接口或方法名后面的编号是引入该特性的 JDK 版本号，如下例所示：

 **应用程序编程接口 9**

源代码在本书英文版网站上以程序清单形式给出，例如：

程序清单 1-1 InputTest/InputTest.java

示例代码

本书英文版网站 <http://horstmann.com/corejava> 提供了书中的所有示例代码。有关安装 Java 开发工具包和示例代码的详细信息请参看第 2 章。

致 谢

写一本书需要投入大量的精力，升级一本书也并不像想象的那样轻松，尤其是 Java 技术一直在持续不断地更新。出版一本书会让很多人耗费很多心血，在此衷心地感谢《Java 核心技术》小组的每一位成员。

Pearson 公司的许多人提供了非常有价值的帮助，却甘愿做幕后英雄。在此，我希望大家能够知道我对他们努力的感恩。与以往一样，我要真诚地感谢我的编辑 Greg Doench，从本书的写作到出版他一直在给予我们指导，同时感谢那些不知其姓名的为本书做出贡献的幕后人士。非常感谢 Julie Nahil 在图书制作方面给予的支持，感谢 Dmitry Kirsanov 和 Alina Kirsanova 完成手稿的编辑与排版工作。我还要感谢早期版本中我的合作者 Gary Cornell，他已经转向了其他行业。

感谢早期版本的许多读者，他们指出了许多令人尴尬的错误并给出了许多具有建设性的修改意见。我还要特别感谢本书优秀的审校小组，他们仔细地审阅我的手稿，使本书减少了许多错误。

本书及早期版本的审校专家包括：Chuck Allison (Utah Valley 大学)、Lance Andersen (Oracle)、Paul Anderson (Anderson Software Group)、Alec Beaton (IBM)、Cliff Berg、Andrew Binstock (Oracle)、Joshua Bloch、David Brown、Corky Cartwright、Frank Cohen (PushToTest)、Chris Crane (devXsolution)、Dr. Nicholas J. De Lillo (Manhattan 学院)、RakeshDhoopar (Oracle)、David Geary (Clarity Training)、Jim Gish (Oracle)、Brian Goetz (Oracle)、Angela Gordon、Dan Gordon (Electric Cloud)、Rob Gordon、John Gray (Hartford 大学)、Cameron Gregory (olabs.com)、Marty Hall (coreservlets.com 公司)、Vincent Hardy (Adobe Systems)、Dan Harkey (San Jose 州立大学)、William Higgins (IBM)、Vladimir Ivanovic (PointBase)、Jerry Jackson (CA Technologies)、Tim Kimmert (Walmart)、Chris Laffra、Charlie Lai (Apple)、Angelika Langer、Doug Langston、Hang Lau (McGill 大学)、Mark Lawrence、Doug Lea (SUNY Oswego)、Gregory Longshore、Bob Lynch (Lynch Associates)、Philip Milne (顾问)、Mark Morrissey (Oregon 研究院)、Mahesh Neelakanta (Florida Atlantic 大学)、Hao Pham、Paul Phillion、Blake Ragsdell、Stuart Reges (Arizona 大学)、Simon Ritter (Azul Systems)、Rich Rosen (Interactive Data Corporation)、Peter Sanders (法国尼斯 ESSI 大学)、Dr. Paul Sanghera (圣何塞州立大学 Brooks 学院)、Paul Sevinc (Teamup AG)、Devang Shah (Sun Microsystems)、Yoshiki Shibata、Bradley A. Smith、Steven Stelling (Oracle)、Christopher Taylor、Luke Taylor (Valtech)、George Thiruvathukal、Kim Topley (StreamingEdge)、Janet Traub、Paul Tyma (顾问)、Peter van der Linden、Christian Ullenboom、Burt Walsh、Dan Xu (Oracle) 和 John Zavgren (Oracle)。

Cay Horstmann

2018 年 6 月于加州旧金山

目 录

译者序	
前言	
致谢	
第 1 章 Java 程序设计概述	1
1.1 Java 程序设计平台	1
1.2 Java “白皮书”的关键术语	2
1.2.1 简单性	2
1.2.2 面向对象	3
1.2.3 分布式	3
1.2.4 健壮性	3
1.2.5 安全性	3
1.2.6 体系结构中立	4
1.2.7 可移植性	4
1.2.8 解释型	5
1.2.9 高性能	5
1.2.10 多线程	5
1.2.11 动态性	6
1.3 Java applet 与 Internet	6
1.4 Java 发展简史	7
1.5 关于 Java 的常见误解	10
第 2 章 Java 程序设计环境	12
2.1 安装 Java 开发工具包	12
2.1.1 下载 JDK	12
2.1.2 设置 JDK	14
2.1.3 安装库源文件和文档	16
2.2 使用命令行工具	16
2.3 使用集成开发环境	20
2.4 JShell	23
第 3 章 Java 的基本程序设计结构	26
3.1 一个简单的 Java 应用程序	26
3.2 注释	29
3.3 数据类型	29
3.3.1 整型	30
3.3.2 浮点类型	31
3.3.3 char 类型	32
3.3.4 Unicode 和 char 类型	33
3.3.5 boolean 类型	34
3.4 变量与常量	34
3.4.1 声明变量	34
3.4.2 变量初始化	35
3.4.3 常量	36
3.4.4 枚举类型	37
3.5 运算符	37
3.5.1 算术运算符	37
3.5.2 数学函数与常量	38
3.5.3 数值类型之间的转换	40
3.5.4 强制类型转换	41
3.5.5 结合赋值和运算符	41
3.5.6 自增与自减运算符	41
3.5.7 关系和 boolean 运算符	42
3.5.8 位运算符	43
3.5.9 括号与运算符级别	43
3.6 字符串	44
3.6.1 子串	45
3.6.2 拼接	45
3.6.3 不可变字符串	45
3.6.4 检测字符串是否相等	47
3.6.5 空串与 Null 串	48
3.6.6 码点与代码单元	48
3.6.7 String API	49
3.6.8 阅读联机 API 文档	51

3.6.9	构建字符串	54	4.3.4	从构造器开始	107
3.7	输入与输出	55	4.3.5	用 var 声明局部变量	109
3.7.1	读取输入	55	4.3.6	使用 null 引用	109
3.7.2	格式化输出	57	4.3.7	隐式参数与显式参数	110
3.7.3	文件输入与输出	61	4.3.8	封装的优点	111
3.8	控制流程	62	4.3.9	基于类的访问权限	113
3.8.1	块作用域	63	4.3.10	私有方法	114
3.8.2	条件语句	63	4.3.11	final 实例字段	114
3.8.3	循环	66	4.4	静态字段与静态方法	115
3.8.4	确定循环	69	4.4.1	静态字段	115
3.8.5	多重选择: switch 语句	72	4.4.2	静态常量	116
3.8.6	中断控制流程的语句	74	4.4.3	静态方法	116
3.9	大数	76	4.4.4	工厂方法	117
3.10	数组	79	4.4.5	main 方法	118
3.10.1	声明数组	79	4.5	方法参数	121
3.10.2	访问数组元素	80	4.6	对象构造	126
3.10.3	for each 循环	81	4.6.1	重载	126
3.10.4	数组拷贝	82	4.6.2	默认字段初始化	127
3.10.5	命令行参数	82	4.6.3	无参数的构造器	127
3.10.6	数组排序	83	4.6.4	显式字段初始化	128
3.10.7	多维数组	86	4.6.5	参数名	129
3.10.8	不规则数组	88	4.6.6	调用另一个构造器	129
第 4 章	对象与类	92	4.6.7	初始化块	130
4.1	面向对象程序设计概述	92	4.6.8	对象析构与 finalize 方法	134
4.1.1	类	93	4.7	包	134
4.1.2	对象	94	4.7.1	包名	134
4.1.3	识别类	94	4.7.2	类的导入	135
4.1.4	类之间的关系	95	4.7.3	静态导入	136
4.2	使用预定义类	96	4.7.4	在包中增加类	137
4.2.1	对象与对象变量	96	4.7.5	包访问	139
4.2.2	Java 类库中的 LocalDate 类	99	4.7.6	类路径	140
4.2.3	更改器方法与访问器方法	100	4.7.7	设置类路径	142
4.3	用户自定义类	103	4.8	JAR 文件	143
4.3.1	Employee 类	104	4.8.1	创建 JAR 文件	143
4.3.2	多个源文件的使用	106	4.8.2	清单文件	144
4.3.3	剖析 Employee 类	107	4.8.3	可执行 JAR 文件	145

4.8.4	多版本 JAR 文件	145	5.6	枚举类	196
4.8.5	关于命令行选项的说明	146	5.7	反射	198
4.9	文档注释	148	5.7.1	Class 类	199
4.9.1	注释的插入	148	5.7.2	声明异常入门	201
4.9.2	类注释	149	5.7.3	资源	202
4.9.3	方法注释	149	5.7.4	利用反射分析类的能力	203
4.9.4	字段注释	150	5.7.5	使用反射在运行时分析对象	208
4.9.5	通用注释	150	5.7.6	使用反射编写泛型数组代码	213
4.9.6	包注释	151	5.7.7	调用任意方法和构造器	216
4.9.7	注释抽取	151	5.8	继承的设计技巧	219
4.10	类设计技巧	152	第 6 章	接口、lambda 表达式与内部类	222
第 5 章	继承	155	6.1	接口	222
5.1	类、超类和子类	155	6.1.1	接口的概念	222
5.1.1	定义子类	155	6.1.2	接口的属性	228
5.1.2	覆盖方法	157	6.1.3	接口与抽象类	229
5.1.3	子类构造器	158	6.1.4	静态和私有方法	230
5.1.4	继承层次	162	6.1.5	默认方法	230
5.1.5	多态	162	6.1.6	解决默认方法冲突	231
5.1.6	理解方法调用	163	6.1.7	接口与回调	233
5.1.7	阻止继承: final 类和方法	165	6.1.8	Comparator 接口	235
5.1.8	强制类型转换	166	6.1.9	对象克隆	236
5.1.9	抽象类	168	6.2	lambda 表达式	242
5.1.10	受保护访问	173	6.2.1	为什么引入 lambda 表达式	242
5.2	Object: 所有类的超类	174	6.2.2	lambda 表达式的语法	243
5.2.1	Object 类型的变量	174	6.2.3	函数式接口	245
5.2.2	equals 方法	175	6.2.4	方法引用	247
5.2.3	相等测试与继承	176	6.2.5	构造器引用	250
5.2.4	hashCode 方法	179	6.2.6	变量作用域	250
5.2.5	toString 方法	181	6.2.7	处理 lambda 表达式	252
5.3	泛型数组列表	186	6.2.8	再谈 Comparator	254
5.3.1	声明数组列表	187	6.3	内部类	255
5.3.2	访问数组列表元素	189	6.3.1	使用内部类访问对象状态	256
5.3.3	类型化与原始数组列表的兼容性	191	6.3.2	内部类的特殊语法规则	259
5.4	对象包装器与自动装箱	192			
5.5	参数数量可变的方法	195			

6.3.3	内部类是否有用、必要和安全	260	7.5.6	过滤器	312
6.3.4	局部内部类	262	7.5.7	格式化器	313
6.3.5	由外部方法访问变量	263	7.5.8	日志技巧	313
6.3.6	匿名内部类	264	7.6	调试技巧	321
6.3.7	静态内部类	267	第 8 章	泛型程序设计	326
6.4	服务加载器	270	8.1	为什么要使用泛型程序设计	326
6.5	代理	273	8.1.1	类型参数的好处	326
6.5.1	何时使用代理	273	8.1.2	谁想成为泛型程序员	328
6.5.2	创建代理对象	273	8.2	定义简单泛型类	328
6.5.3	代理类的特性	277	8.3	泛型方法	330
第 7 章	异常、断言和日志	279	8.4	类型变量的限定	331
7.1	处理错误	279	8.5	泛型代码和虚拟机	333
7.1.1	异常分类	280	8.5.1	类型擦除	333
7.1.2	声明检查型异常	282	8.5.2	转换泛型表达式	335
7.1.3	如何抛出异常	284	8.5.3	转换泛型方法	335
7.1.4	创建异常类	285	8.5.4	调用遗留代码	337
7.2	捕获异常	286	8.6	限制与局限性	338
7.2.1	捕获异常	286	8.6.1	不能用基本类型实例化类型参数	338
7.2.2	捕获多个异常	288	8.6.2	运行时类型查询只适用于原始类型	338
7.2.3	再次抛出异常与异常链	289	8.6.3	不能创建参数化类型的数组	338
7.2.4	finally 子句	290	8.6.4	Varargs 警告	339
7.2.5	try-with-Resources 语句	292	8.6.5	不能实例化类型变量	340
7.2.6	分析堆栈轨迹元素	294	8.6.6	不能构造泛型数组	341
7.3	使用异常的技巧	297	8.6.7	泛型类的静态上下文中类型变量无效	342
7.4	使用断言	300	8.6.8	不能抛出或捕获泛型类的实例	343
7.4.1	断言的概念	300	8.6.9	可以取消对检查型异常的检查	343
7.4.2	启用和禁用断言	301	8.6.10	注意擦除后的冲突	345
7.4.3	使用断言完成参数检查	302	8.7	泛型类型的继承规则	346
7.4.4	使用断言提供假设文档	303	8.8	通配符类型	348
7.5	日志	304	8.8.1	通配符概念	348
7.5.1	基本日志	305			
7.5.2	高级日志	305			
7.5.3	修改日志管理器配置	307			
7.5.4	本地化	308			
7.5.5	处理器	309			

8.8.2	通配符的超类型限定	349	9.5.5	检查型视图	407
8.8.3	无限定通配符	351	9.5.6	关于可选操作的说明	407
8.8.4	通配符捕获	352	9.6	算法	411
8.9	反射和泛型	354	9.6.1	为什么使用泛型算法	411
8.9.1	泛型 Class 类	354	9.6.2	排序与混排	412
8.9.2	使用 Class<T> 参数进行类型 匹配	355	9.6.3	二分查找	414
8.9.3	虚拟机中的泛型类型信息	356	9.6.4	简单算法	415
8.9.4	类型字面量	359	9.6.5	批操作	417
第 9 章	集合	365	9.6.6	集合与数组的转换	418
9.1	Java 集合框架	365	9.6.7	编写自己的算法	418
9.1.1	集合接口与实现分离	365	9.7	遗留的集合	419
9.1.2	Collection 接口	368	9.7.1	Hashtable 类	419
9.1.3	迭代器	368	9.7.2	枚举	419
9.1.4	泛型实用方法	370	9.7.3	属性映射	421
9.2	集合框架中的接口	373	9.7.4	栈	424
9.3	具体集合	375	9.7.5	位集	424
9.3.1	链表	375	第 10 章	图形用户界面程序设计	429
9.3.2	数组列表	384	10.1	Java 用户界面工具包简史	429
9.3.3	散列集	385	10.2	显示窗体	430
9.3.4	树集	388	10.2.1	创建窗体	431
9.3.5	队列与双端队列	391	10.2.2	窗体属性	433
9.3.6	优先队列	392	10.3	在组件中显示信息	435
9.4	映射	394	10.3.1	处理 2D 图形	439
9.4.1	基本映射操作	394	10.3.2	使用颜色	445
9.4.2	更新映射条目	397	10.3.3	使用字体	446
9.4.3	映射视图	398	10.3.4	显示图像	452
9.4.4	弱散列映射	399	10.4	事件处理	453
9.4.5	链接散列集与映射	400	10.4.1	基本事件处理概念	453
9.4.6	枚举集与映射	401	10.4.2	实例: 处理按钮点击 事件	454
9.4.7	标识散列映射	402	10.4.3	简洁地指定监听器	457
9.5	视图与包装器	403	10.4.4	适配器类	458
9.5.1	小集合	404	10.4.5	动作	460
9.5.2	子范围	405	10.4.6	鼠标事件	464
9.5.3	不可修改的视图	405	10.4.7	AWT 事件继承层次	469
9.5.4	同步视图	406	10.5	首选项 API	472