



Pearson



# Java核心技术

## 卷I：基础知识

(第10版·英文版)

[美] Cay S. Horstmann 著

# Core Java

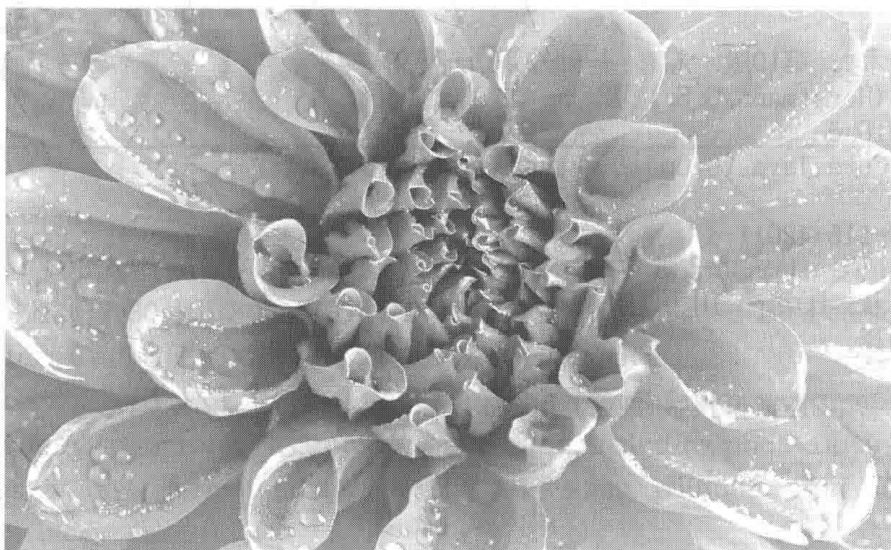
Volume I—Fundamentals, Tenth Edition



中国工信出版集团



人民邮电出版社  
POSTS & TELECOM PRESS



# Java核心技术

## 卷I：基础知识 **上**

(第10版·英文版)

[美] Cay S. Horstmann 著

# Core Java

Volume I—Fundamentals, Tenth Edition

人民邮电出版社  
北京

## 图书在版编目 (C I P) 数据

Java核心技术：第10版. 卷I. 基础知识：英文 / (美) 霍斯特曼 (Horstmann, C. S.) 著. — 北京：人民邮电出版社，2016. 6

书名原文：Core Java Volume I—Fundamentals, Tenth Edition

ISBN 978-7-115-42011-4

I. ①J… II. ①霍… III. ①JAVA语言—程序设计—英文 IV. ①TP312

中国版本图书馆CIP数据核字(2016)第064144号

## 内 容 提 要

本书是经典的《Java 核心技术 卷 I：基础知识》的最新版。这一版针对 Java SE 8 平台进行了全面更新，以反映 Java SE 8 的特性。

书中囊括了 Java 的全部基础知识，提供了大量完整且具有实际意义的应用示例，详细介绍了 Java 语言基础、面向对象编程、反射与代理、接口与内部类、事件监听器模型、使用 Swing GUI 工具进行图形用户界面程序设计、打包应用程序、异常处理、登录与调试、泛型编程、集合框架、多线程、并发等内容。

本书适合想将 Java 应用于实际项目的软件开发人员、高等院校教师和学生参考阅读。

---

◆ 著 [美] Cay S. Horstmann

责任编辑 杨海玲

责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京昌平百善印刷厂印刷

◆ 开本：800×1000 1/16

印张：61.75

字数：1172 千字

2016 年 6 月第 1 版

印数：1—2 500 册

2016 年 6 月北京第 1 次印刷

著作权合同登记号 图字：01-2016-2258 号

---

定价：109.00 元（上、下册）

读者服务热线：(010)81055410 印装质量热线：(010)81055316

反盗版热线：(010)81055315

广告经营许可证：京东工商广字第 8052 号

# 版权声明

---

Original edition, entitled Core Java Volume I—Fundamentals, Tenth Edition (9780134177304), by HORSTMANN, CAY S., published by Pearson Education, Inc., Copyright © 2016 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 Inc.

China edition published by PEARSON EDUCATION ASIA LTD., and POSTS & TELECOMMUNICATIONS PRESS Copyright ©2016.

This edition is manufactured in the People's Republic of China, and is authorized for sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR).

仅限于中华人民共和国境内（不包括中国香港、澳门特别行政区和中国台湾地区）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。

# 前言<sup>①</sup>

## 致读者

1995年年底，Java 编程语言异军突起，登上 Internet 舞台，并且一夜成名。Java 技术希望成为连接用户与信息的“万能胶”，而不论这些信息来自何处——是 Web 服务器、数据库、信息提供商，还是其他任何可以想到的源头。的确，Java 具备实现这一愿景的得天独厚的条件。Java 是已被广泛接受的、极为稳健的工程语言。其固有的安全性不仅令 Java 程序员放心，也令使用 Java 程序的用户放心。Java 为很多高级编程任务提供了内置支持，比如网络编程、数据库连接和并发。

自 1995 年起，Java 开发工具包（Java Development Kit, JDK）已经发布了 9 个主要版本。在过去的 20 年中，应用编程接口（Application Programming Interface, API）从大约 200 个类发展到超过 4000 个类。API 现在涉及诸如用户界面构建、数据库管理、国际化、安全和 XML 处理等多个不同领域。

本书是《Java 核心技术》一书第 10 版的卷 I。本书的每一版都紧跟一个 JDK 版本，而且我们每一次都会重写，以利用最新的 Java 特性。这一版也做了更新，以反映 Java SE 8 的特性。

与之前的版本一样，本书的读者定位仍是那些希望将 Java 应用于实际项目的严谨的程序员。本书假设读者是基础扎实的程序员，已掌握 Java 以外的某种编程语言，而且读者不希望书中充斥着玩具式的示例（比如烤面包机、动物园里的动物甚至“跳动的文字”）。本书不会使用任何那样的示例。我们的目标是让读者充分理解 Java 语言及其库，而不是带给读者一种“已经理解了”的假象。

本书提供了大量的示例代码，用以说明我们所讨论的几乎每一个语言特性和库特性。我们有意让示例代码尽量简单，以便专注于主要问题。不过绝大多数代码既非凭空杜撰，也未偷工减料。对读者编写自己的代码而言，它们是不错的起点。

① 本前言由臧秀涛翻译。

我们假设读者愿意，甚至是渴望学习 Java 提供的所有高级特性。例如，我们会详细讨论如下内容：

- 面向对象编程；
- 反射与代理；
- 接口与内部类；
- 异常处理；
- 泛型编程；
- 集合类框架；
- 事件监听器模型；
- 使用 Swing UI 工具包设计图形用户界面；
- 并发。

随着 Java 类库的爆炸式增长，一本书已经无法涵盖程序员所需了解的所有 Java 特性。因此，我们决定将《Java 核心技术》一书分为两卷。现在读者手中的是卷 I，主要介绍 Java 语言的基本概念，以及图形用户界面编程的基础知识。《Java 核心技术 卷 II：高级特性》会更进一步，介绍企业级特性和高级图形界面编程，详细讨论如下内容：

- Stream API；
- 文件处理与正则表达式；
- 数据库；
- XML 处理；
- 注解（Annotation）；
- 国际化；
- 网络编程；
- 高级 GUI 组件；
- 高级图形编程；
- 原生方法。

书写过程中难免会有错误和不准确之处。如果读者在本书中发现了问题，希望可以通知我们。当然，我们也不希望一个问题被提交多次。为此我们在 <http://horstmann.com/corejava> 上创建了一个页面，存放常见问题（FAQ）、错误更正和解决方案。在错误提交页面上，我们希望读者先阅读已经提交的错误，因此把报告错误和提供改进建议的表格放到了最后。如果我们没有回答问题，或没有即时回复，请不要失望。我们肯定会阅读所有来信，并感谢读者的建议，它们使本书将来的版本更清晰、更有指导价值。

## 本书概览

第 1 章概述 Java 有别于其他编程语言的功能。这一章将介绍 Java 语言设计者的初衷，以及成效如何。然后简单了解一下这门语言诞生和发展的历史。

第 2 章介绍如何下载与安装 JDK 及本书示例代码。然后引导读者使用纯 JDK、支持 Java 的文本编辑器和 Java IDE（集成开发环境）来编译并运行 3 个典型的 Java 程序——一个控制台应用、一个图形界面应用和一个 Applet。

第 3 章开始讨论 Java 语言。这一章涵盖变量、循环和简单函数等基础知识。如果读者是 C 或 C++ 程序员，学起来会很容易，因为这些语言特性的语法和 C 语言基本一样。如果没有 C 语言背景，比如熟悉 Visual Basic，则应该仔细阅读这一章。

面向对象编程（Object-Oriented Programming, OOP）是当今主流的编程方式，而 Java 正是一种面向对象编程语言。第 4 章介绍封装（面向对象技术的两个基本构建块之一），以及 Java 语言实现封装的机制，也就是类和方法。除了 Java 语言的语法规则之外，我们还会就如何进行合理的 OOP 设计给出一些建议。这一章最后会介绍奇妙的 javadoc 工具，它可以将代码注释格式化为一组通过超链接关联起来的网页。如果读者熟悉 C++，可以快速浏览一下这一章。如果读者是没有面向对象编程背景的程序员，在进一步学习 Java 之前，应该花点儿时间掌握 OOP 概念。

类和封装只是 OOP 的一部分，第 5 章将介绍另一个重要概念，也就是继承。继承使我们可以利用现有的类，并根据自己的需求加以修改。这是基本的 Java 编程技术。Java 的继承机制与 C++ 非常像。再强调一次，C++ 程序员关注两种语言的不同之处即可。

第 6 章介绍如何使用 Java 的接口。与第 5 章的简单继承模型相比，接口让我们更进一步。掌握了接口，就可以充分发挥 Java 面向对象编程的威力了。在介绍完接口之后，我们将探讨 Lambda 表达式，它可以简洁地表达可能会在未来的某个时间点执行的代码块。之后将介绍 Java 中一个有用的技术特性——内部类。

第7章探讨异常处理。良好的程序也会出现问题，而异常处理就是 Java 用来处理这种状况的健壮性机制。异常处理提供了一种将正常处理代码与错误处理分离的有效手段。当然，即使程序处理了所有异常条件，仍然有可能无法按预期方式工作。这一章最后会给出一些有用的调试技巧。

第8章概要讲解泛型编程。泛型编程使程序可读性更好，也更安全。这一章介绍如何使用强类型及去掉难看且不安全的强制类型转换，还将介绍如何处理为兼容旧的 Java 版本而带来的复杂问题。

第9章的主题是 Java 平台的集合类框架。当需要将多个对象收集到一起，并在之后检索时，应该选择最适合问题环境的集合，而不是将所有元素一股脑儿地扔到数组中。这一章将阐述如何使用 Java 预先设计好的标准集合类。

第10章开始介绍 GUI 编程。这一章将介绍如何创建窗口，如何在窗口中绘图，如何绘制几何图形，如何以多种字体格式化文本，以及如何显示图片。

第11章详细讨论抽象窗口工具包（Abstract Window Toolkit, AWT）的事件模型。读者会看到如何编写响应比如鼠标点击或键盘敲击等事件的代码。读者还会看到如何处理基本的 GUI 元素，如按钮和面板。

第12章将详细探讨 Swing GUI 工具包。利用 Swing 工具包，可以构建跨平台的图形用户界面。这一章将介绍与各种按钮、文本组件、边框、滑块、列表框、菜单和对话框等控件相关的知识。不过有些更高级的组件将在卷 II 中讨论。

第13章介绍如何部署程序，包括应用和 Applet。这一章将描述如何将程序打包到 JAR 文件中，以及如何使用 Java Web Start 和 Applet 机制在 Internet 上发布应用。这一章还会介绍 Java 程序部署之后如何存储和检索配置信息。

第14章是最后一章，探讨并发。该特性支持多个程序任务并行完成。大部分处理器都有多个核心，而且我们想让这些核心保持忙碌的状态，这是 Java 技术的一个重要且激动人心的应用。

附录列出了 Java 语言的保留字。

## 排版约定

和很多计算机图书一样，我们使用等宽字体表示计算机代码。





**NOTE (注解):** 注解会使用这样的图标标记。



**TIP (提示):** 提示会使用这样的图标标记。



**CAUTION (警告):** 如果使用不当会存在危险，我们会使用这样的图标警示。



**C++ NOTE (C++注解):** 书中有很多 C++ 注解，用于解释 Java 和 C++ 的差别。如果没有 C++ 背景，或者虽有相关背景，但却视如梦魇、不愿想起，都可以跳过。

Java 提供了一个规模庞大的编程库，或者说 API。在第一次使用某个 API 调用时，我们会在该节末尾添加一个简短的描述。这些描述不那么正式，不过我们希望它们比官方的在线 API 文档更有指导意义。类名、接口名或方法名后的数字表示该特性是在 JDK 的哪个版本中引入的。如下所示：

Application Programming Interface 1.2

程序的源代码都在本书的配套网站上，在书中它们会以代码清单形式列出。例如：

**Listing 1.1** InputTest/InputTest.java

## 示例代码

本书的配套网站 <http://horstmann.com/corejava> 提供了书中所有的示例代码。代码是以压缩文件格式给出的。读者既可以使用常用的解压程序解压，也可以使用 JDK 中的 jar 工具处理。至于如何安装 JDK，以及如何使用示例代码，详见第 2 章。

# 致谢

写一本书需要投入大量精力，重写看上去好像容易一些，其实并非如此，尤其是 Java 技术还在持续更新。一本书能成功面世，凝聚了很多人的心血。在此衷心感谢整个《Java 核心技术》团队的贡献。

Prentice Hall 的很多人居功至伟，却甘居幕后。我希望他们都能知道我是多么感激。和往常一样，衷心感谢我的编辑 Greg Doench，他掌控着本书从编写到制作的流程，让我觉察不到幕后的那些人。非常感谢 Julie Nahil 在图书制作方面的支持，感谢 Dmitry Kirsanov 和 Alina Kirsanova 完成手稿的编辑和排版。还要感谢本书早期版本的合著者 Gary Cornell，如今他已开始别的事业。

感谢早期版本的许多读者，他们指出了很多令人尴尬的错误，提供了很多深思熟虑的改进建议。特别感谢非常优秀的审稿团队，他们仔细审读了手稿，使我减少了很多令人尴尬的错误。

本书及早期版本的审阅专家有：Chuck Allison（犹他谷州立大学）、Lance Andersen（甲骨文）、Paul Anderson（Anderson Software Group）、Alec Beaton（IBM）、Cliff Berg、Andrew Binstock（甲骨文）、Joshua Bloch、David Brown、Corky Cartwright、Frank Cohen（PushToTest）、Chris Crane（devXsolution）、Dr. Nicholas J. De Lillo（曼哈顿大学）、RakeshDhoopar（甲骨文）、David Geary（Clarity Training）、Jim Gish（甲骨文）、Brian Goetz（甲骨文）、Angela Gordon、Dan Gordon（Electric Cloud）、Rob Gordon、John Gray（哈特福大学）、Cameron Gregory（olabs.com）、Marty Hall（coreservlets.com, Inc.）、Vincent Hardy（Adobe 公司）、Dan Harkey（圣何塞州立大学）、William Higgins（IBM）、Vladimir Ivanovic（PointBase）、Jerry Jackson（CA 科技）、Tim Kimmet（沃尔玛公司）、Chris Laffra、Charlie Lai（苹果）、Angelika Langer、Doug Langston、Hang Lau（麦吉尔大学）、Mark Lawrence、Doug Lea（纽约州立大学奥斯威戈分校）、Gregory Longshore、Bob Lynch（Lynch Associates）、Philip Milne（咨询师）、Mark Morrissey（俄勒冈研究院）、Mahesh Neelakanta（佛罗里达州大西洋大学）、Hao Pham、Paul Pillion、Blake Ragsdell、Stuart Reges（亚利桑那大学）、Rich Rosen（Interactive Data Corporation）、Peter Sanders（ESSI University,

Nice, France)、Dr. Paul Sanghera (圣何塞州立大学和布鲁克斯学院)、Paul Sevinc (Teamup AG)、Devang Shah (太阳微系统公司)、Yoshiki Shibata、Bradley A. Smith、Steven Stelting (甲骨文)、Christopher Taylor、Luke Taylor (Valtech)、George Thiruvathukal、Kim Topley (StreamingEdge)、Janet Traub、Paul Tyma (咨询师)、Peter van der Linden、Christian Ullenboom、Burt Walsh、Dan Xu (甲骨文) 和 John Zavgren (甲骨文)。

Cay Horstmann

2015年11月于瑞士比尔

# 目录

<b>Chapter 1: An Introduction to Java / Java 概述</b>	<b>1</b>
1.1 Java as a Programming Platform / Java 程序设计平台	1
1.2 The Java “White Paper” Buzzwords / Java “白皮书”中的口号	2
1.2.1 Simple / 简单	3
1.2.2 Object-Oriented / 面向对象	4
1.2.3 Distributed / 分布式	4
1.2.4 Robust / 健壮	4
1.2.5 Secure / 安全	4
1.2.6 Architecture-Neutral / 体系结构中立	5
1.2.7 Portable / 可移植	6
1.2.8 Interpreted / 解释型	7
1.2.9 High-Performance / 高性能	7
1.2.10 Multithreaded / 多线程	7
1.2.11 Dynamic / 动态	8
1.3 Java Applets and the Internet / Java Applet 与 Internet	8
1.4 A Short History of Java / Java 简史	10
1.5 Common Misconceptions about Java / 对 Java 的常见误解	13
<b>Chapter 2: The Java Programming Environment / Java 编程环境</b>	<b>17</b>
2.1 Installing the Java Development Kit / 安装 Java 开发包 (JDK)	18
2.1.1 Downloading the JDK / 下载 JDK	18
2.1.2 Setting up the JDK / 设置 JDK	20
2.1.3 Installing Source Files and Documentation / 源文件与文档的下载与设置	22
2.2 Using the Command-Line Tools / 使用命令行工具	23
2.3 Using an Integrated Development Environment / 使用集成开发环境	26
2.4 Running a Graphical Application / 运行图形化应用程序	30
2.5 Building and Running Applets / 构建并运行 Applet	33

<b>Chapter 3: Fundamental Programming Structures in Java /</b>	
<b>Java 的基本编程结构</b>	<b>41</b>
3.1 A Simple Java Program / 一个简单的 Java 程序	42
3.2 Comments / 注释	46
3.3 Data Types / 数据类型	47
3.3.1 Integer Types / 整型	47
3.3.2 Floating-Point Types / 浮点型	48
3.3.3 The char Type / char 类型	50
3.3.4 Unicode and the char Type / Unicode 与 char 类型	51
3.3.5 The boolean Type / boolean 类型	52
3.4 Variables / 变量	53
3.4.1 Initializing Variables / 初始化变量	54
3.4.2 Constants / 常量	55
3.5 Operators / 运算符	56
3.5.1 Mathematical Functions and Constants / 数学函数与常量	57
3.5.2 Conversions between Numeric Types / 数值类型之间的转换	59
3.5.3 Casts / 强制类型转换	60
3.5.4 Combining Assignment with Operators / 组合赋值运算符	61
3.5.5 Increment and Decrement Operators / 自增运算符与自减运算符	61
3.5.6 Relational and Boolean Operators / 关系与 boolean 运算符	62
3.5.7 Bitwise Operators / 位运算符	63
3.5.8 Parentheses and Operator Hierarchy / 括号与运算符 优先级	64
3.5.9 Enumerated Types / 枚举类型	65
3.6 Strings / 字符串	65
3.6.1 Substrings / 子串	66
3.6.2 Concatenation / 拼接	66
3.6.3 Strings Are Immutable / String 是不可变的	67
3.6.4 Testing Strings for Equality / 测试字符串是否相等	68
3.6.5 Empty and Null Strings / 空串与 null 串	69

3.6.6	Code Points and Code Units / 码位与编码单元	70
3.6.7	The String API / String 类的 API	71
3.6.8	Reading the Online API Documentation / 阅读在线 API 文档	74
3.6.9	Building Strings / 构建字符串	77
3.7	Input and Output / 输入输出	78
3.7.1	Reading Input / 读取输入	79
3.7.2	Formatting Output / 格式化输出	82
3.7.3	File Input and Output / 文件输入输出	87
3.8	Control Flow / 控制流	89
3.8.1	Block Scope / 块作用域	89
3.8.2	Conditional Statements / 条件语句	90
3.8.3	Loops / 循环	94
3.8.4	Determinate Loops / 确定性循环	99
3.8.5	Multiple Selections—The switch Statement / 多重选择: switch 语句	103
3.8.6	Statements That Break Control Flow / 用于跳出控制流的语句	106
3.9	Big Numbers / 大数	108
3.10	Arrays / 数组	111
3.10.1	The “for each” Loop / “for each” 循环	113
3.10.2	Array Initializers and Anonymous Arrays / 数组初始化与匿名数组	114
3.10.3	Array Copying / 数组复制	114
3.10.4	Command-Line Parameters / 命令行参数	116
3.10.5	Array Sorting / 数组排序	117
3.10.6	Multidimensional Arrays / 多维数组	120
3.10.7	Ragged Arrays / 不规则数组	124
<b>Chapter 4: Objects and Classes / 对象与类</b>		<b>129</b>
4.1	Introduction to Object-Oriented Programming / 面向对象编程简介	130
4.1.1	Classes / 类	131
4.1.2	Objects / 对象	132
4.1.3	Identifying Classes / 识别类	133

4.1.4	Relationships between Classes / 类之间的关系	133
4.2	Using Predefined Classes / 使用预定义类	135
4.2.1	Objects and Object Variables / 对象与对象变量	136
4.2.2	The LocalDate Class of the Java Library / Java 库中的 LocalDate 类	139
4.2.3	Mutator and Accessor Methods / 更改器方法与访问器方法	141
4.3	Defining Your Own Classes / 定义自己的类	145
4.3.1	An Employee Class / Employee 类	145
4.3.2	Use of Multiple Source Files / 使用多个源文件	149
4.3.3	Dissecting the Employee Class / 分析 Employee 类	149
4.3.4	First Steps with Constructors / 从构造器开始	150
4.3.5	Implicit and Explicit Parameters / 隐式参数与显式参数	152
4.3.6	Benefits of Encapsulation / 封装的好处	153
4.3.7	Class-Based Access Privileges / 基于类的访问权限	156
4.3.8	Private Methods / 私有方法	156
4.3.9	Final Instance Fields / final 实例字段	157
4.4	Static Fields and Methods / 静态字段与静态方法	158
4.4.1	Static Fields / 静态字段	158
4.4.2	Static Constants / 静态常量	159
4.4.3	Static Methods / 静态方法	160
4.4.4	Factory Methods / 工厂方法	161
4.4.5	The main Method / main 方法	161
4.5	Method Parameters / 方法参数	164
4.6	Object Construction / 对象构建	171
4.6.1	Overloading / 重载	172
4.6.2	Default Field Initialization / 默认字段初始化	172
4.6.3	The Constructor with No Arguments / 无参构造器	173
4.6.4	Explicit Field Initialization / 显式字段初始化	174
4.6.5	Parameter Names / 参数名	175
4.6.6	Calling Another Constructor / 调用另一个构造器	176
4.6.7	Initialization Blocks / 初始化块	177
4.6.8	Object Destruction and the finalize Method / 对象析构与 finalize 方法	181

4.7	Packages / 包	182
4.7.1	Class Importation / 导入类	183
4.7.2	Static Imports / 静态导入	185
4.7.3	Addition of a Class into a Package / 将类添加到某个包中	185
4.7.4	Package Scope / 包作用域	189
4.8	The Class Path / 类路径	190
4.8.1	Setting the Class Path / 设置类路径	193
4.9	Documentation Comments / 文档注释	194
4.9.1	Comment Insertion / 插入注释	194
4.9.2	Class Comments / 类注释	195
4.9.3	Method Comments / 方法注释	195
4.9.4	Field Comments / 字段注释	196
4.9.5	General Comments / 通用注释	196
4.9.6	Package and Overview Comments / 包与概述注释	198
4.9.7	Comment Extraction / 提取注释	198
4.10	Class Design Hints / 类设计建议	200
<b>Chapter 5: Inheritance / 继承</b>		<b>203</b>
5.1	Classes, Superclasses, and Subclasses / 类、超类与子类	204
5.1.1	Defining Subclasses / 定义子类	204
5.1.2	Overriding Methods / 覆盖方法	206
5.1.3	Subclass Constructors / 子类构造器	207
5.1.4	Inheritance Hierarchies / 继承层次	212
5.1.5	Polymorphism / 多态	213
5.1.6	Understanding Method Calls / 理解方法调用	214
5.1.7	Preventing Inheritance: Final Classes and Methods / 阻止继承: final 修饰的类和方法	217
5.1.8	Casting / 强制类型转换	219
5.1.9	Abstract Classes / 抽象类	221
5.1.10	Protected Access / 受保护访问	227
5.2	Object: The Cosmic Superclass / Object: 所有类的超类	228
5.2.1	The equals Method / equals 方法	229
5.2.2	Equality Testing and Inheritance / 相等测试与继承	231



5.2.3	The hashCode Method / hashCode 方法	235
5.2.4	The toString Method / toString 方法	238
5.3	Generic Array Lists / 泛型数组列表	244
5.3.1	Accessing Array List Elements / 访问泛型数组列表的元素	247
5.3.2	Compatibility between Typed and Raw Array Lists / 泛型数组列表与原始数组列表的兼容性	251
5.4	Object Wrappers and Autoboxing / 对象包装器与自动装箱	252
5.5	Methods with a Variable Number of Parameters / 参数数量可变的方法	256
5.6	Enumeration Classes / 枚举类	258
5.7	Reflection / 反射	260
5.7.1	The Class Class / Class 类	261
5.7.2	A Primer on Catching Exceptions / 捕获异常简介	263
5.7.3	Using Reflection to Analyze the Capabilities of Classes / 使用反射分析类的能力	265
5.7.4	Using Reflection to Analyze Objects at Runtime / 在运行时使用反射分析对象	271
5.7.5	Using Reflection to Write Generic Array Code / 使用反射编写泛型数组代码	276
5.7.6	Invoking Arbitrary Methods / 调用任意方法	279
5.8	Design Hints for Inheritance / 继承的设计建议	283
<b>Chapter 6: Interfaces, Lambda Expressions, and Inner Classes / 接口、Lambda 表达式和内部类</b>		<b>287</b>
6.1	Interfaces / 接口	288
6.1.1	The Interface Concept / 接口的概念	288
6.1.2	Properties of Interfaces / 接口的特性	295
6.1.3	Interfaces and Abstract Classes / 接口与抽象类	297
6.1.4	Static Methods / 静态方法	298
6.1.5	Default Methods / 默认方法	298
6.1.6	Resolving Default Method Conflicts / 解决默认方法的冲突	300
6.2	Examples of Interfaces / 接口示例	302
6.2.1	Interfaces and Callbacks / 接口与回调	302