

PRENTICE
HALL

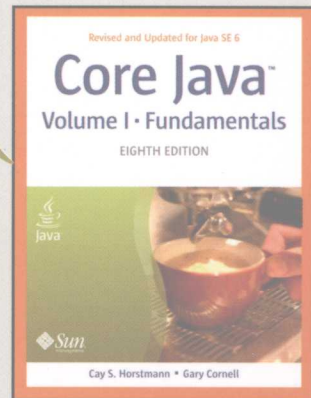


Core Java
Volume I-Fundamentals
EIGHTH EDITION

[美] Cay S. Horstmann 著
Gary Cornell

Java 核心技术

卷I：基础知识 (第8版) (英文版)



人民邮电出版社
POSTS & TELECOM PRESS





Java 核心技术 卷 I: 基础知识

(第 8 版) (英文版)

[美] Cay S. Horstmann 著
Gary Cornell

人民邮电出版社

北 京

图书在版编目 (CIP) 数据

Java核心技术: 第8版. 卷1, 基础知识: 英文 / (美) 霍斯特曼 (Horstmann, C. S.), (美) 科奈尔 (Cornell, G.) 著. —北京: 人民邮电出版社, 2008. 11
ISBN 978-7-115-18833-5

I. J… II. ①霍…②科… III. JAVA语言—程序设计—英文 IV. TP312

中国版本图书馆CIP数据核字 (2008) 第140633号

版权声明

Original edition, entitled Core Java™, Volume I—Fundamentals, 8th edition 0132354764 by Cay S. Horstmann, Gary Cornell, published by Pearson Education, Inc, publishing as Prentice Hall, Copyright © 2008 by 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.

China edition published by PEARSON EDUCATION ASIA LTD., and POSTS & TELECOMMUNICATIONS PRESS Copyright © 2008.

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

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。

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。

Java 核心技术 卷 I: 基础知识 (第 8 版) (英文版)

◆ 著 [美] Cay S. Horstmann Gary Cornell

责任编辑 李 际

执行编辑 付 飞

◆ 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

◆ 开本: 800×1000 1/16

印张: 53.75

字数: 1 149 千字

2008 年 11 月第 1 版

印数: 1—3 000 册

2008 年 11 月北京第 1 次印刷

著作权合同登记号 图字: 01-2008-4083 号

ISBN 978-7-115-18833-5/TP

定价: 99.00 元

读者服务热线: (010)67132705 印装质量热线: (010)67129223

反盗版热线: (010)67171154



内容提要

本书第 1 版出版以来，一直备受广大 Java 程序设计人员的青睐，畅销不衰，是 Java 经典书籍。第 8 版针对 Java SE 6 平台进行了全面更新，囊括了 Java 2 平台、标准版（J2SE）的全部基础知识，提供了大量完整且具有实际意义的应用实例，详细介绍了 Java 语言基础知识、面向对象程序设计、接口与内部类、事件监听器模型、Swing 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、泛型程序设计、集合框架、多线程等内容。

书中示例程序经过精心设计，不但具有实用价值，而且易于阅读理解，可以作为初学者自己编写程序的良好开端，也能够帮助程序员快速地了解 Java SE 6 的新特性，或迅速从其他语言转向 Java 语言。

Acknowledgments

Writing a book is always a monumental effort, and rewriting doesn't seem to be much easier, especially with continuous change in Java technology. Making a book a reality takes many dedicated people, and it is my great pleasure to acknowledge the contributions of the entire Core Java team.

A large number of individuals at Prentice Hall and Sun Microsystems Press provided valuable assistance, but they managed to stay behind the scenes. I'd like them all to know how much I appreciate their efforts. As always, my warm thanks go to my editor, Greg Doench of Prentice Hall, for steering the book through the writing and production process, and for allowing me to be blissfully unaware of the existence of all those folks behind the scenes. I am grateful to Vanessa Moore for the excellent production support. My thanks also to my coauthor of earlier editions, Gary Cornell, who has since moved on to other ventures.

Thanks to the many readers of earlier editions who reported embarrassing errors and made lots of thoughtful suggestions for improvement. I am particularly grateful to the excellent reviewing team that went over the manuscript with an amazing eye for detail and saved me from many more embarrassing errors.

Reviewers of this and earlier editions include Chuck Allison (Contributing Editor, *C/C++ Users Journal*), Alec Beaton (PointBase, Inc.), Cliff Berg (iSavvix Corporation), Joshua Bloch (Sun Microsystems), David Brown, Corky Cartwright, Frank Cohen (PushToTest), Chris Crane (devXsolution), Dr. Nicholas J. De Lillo (Manhattan College), Rakesh Dhoopar (Oracle), David Geary (Sabreware), Brian Goetz (Principal Consultant, Quiotix Corp.), Angela Gordon (Sun Microsystems), Dan Gordon (Sun Microsystems), Rob Gordon, John Gray (University of Hartford), Cameron Gregory (olabs.com), Marty Hall (The Johns Hopkins University Applied Physics Lab), Vincent Hardy (Sun Microsystems), Dan Harkey (San Jose State University), William Higgins (IBM), Vladimir Ivanovic (PointBase), Jerry Jackson (ChannelPoint Software), Tim Kimmet (Preview Systems), Chris Laffra, Charlie Lai (Sun

Microsystems), Angelika Langer, Doug Langston, Hang Lau (McGill University), Mark Lawrence, Doug Lea (SUNY Oswego), Gregory Longshore, Bob Lynch (Lynch Associates), Philip Milne (consultant), Mark Morrissey (The Oregon Graduate Institute), Mahesh Neelakanta (Florida Atlantic University), Hao Pham, Paul Phillion, Blake Ragsdell, Stuart Reges (University of Arizona), Rich Rosen (Interactive Data Corporation), Peter Sanders (ESSI University, Nice, France), Dr. Paul Sanghera (San Jose State University and Brooks College), Paul Sevinc (Teamup AG), Devang Shah (Sun Microsystems), Bradley A. Smith, Steven Stelting (Sun Microsystems), Christopher Taylor, Luke Taylor (Valtech), George Thiruvathukal, Kim Topley (author of *Core JFC*), Janet Traub, Paul Tyma (consultant), Peter van der Linden (Sun Microsystems), and Burt Walsh.

Cay Horstmann
San Francisco, 2007

致读者

1995 年底，Java 语言在 Internet 舞台上一亮相便名声大噪。其原因在于它将有成为通过信息将用户联系起来的“万能胶”，而不论这些信息来自于 Web 服务器、数据库、信息提供商，还是任何其他渠道。事实上，就发展前景而言，Java 的地位是独一无二的。它是一种完全可以信赖的程序设计语言，得到了除微软之外所有厂家的认可。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、多线程等高级程序设计任务的支持。

1995 年以来，SUN Microsystems 公司已经发布了 Java 开发工具箱（Java Development Kit）的 7 个主要版本。在过去的 11 年中，应用程序接口（API）已经从 200 个类扩展到 3000 个类，并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 8 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 开发工具箱发展的步伐，并重新改写部分内容，以适应 Java 的最新特性。在这一版中，已经反映了 Java 标准版（Java SE 6）的特性。

与前几版一样，本版仍然将读者群定位在那些打算将 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设计人员，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地理解书中介绍的 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。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;
- 事件监听器模型;
- 使用 Swing UI 工具箱进行图形用户界面设计;
- 异常处理;
- 泛型程序设计;
- 集合框架;
- 并行操作。

随着 Java 类库的爆炸式增长, 一本书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 (本卷) 集中介绍 Java 的基本概念以及图形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, 其中包含下列内容:

- 文件与流;
- 分布式对象;
- 数据库;
- 高级 GUI 组件;
- 本地方法;
- XML 处理;
- 网络编程;
- 高级图形;
- 国际化;
- JavaBean;
- 注释。

在这一版中, 我们对两卷中的内容进行了调整, 特别是, 鉴于多线程的日趋重要, 我们将它编入卷 I 中, 并以摩尔定律作为结尾。

在编写本书的过程中, 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.coni/coreJava.html> 中以列表的形式给出了常见的问题、bug 修正和出错位置。在勘误页面 (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性能, 解释这种语言的设计初衷, 以及在哪些方面达到了预期的效果, 然后简述了 Java 诞生和发展的历史。

第 2 章详细地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、一个图形应用和一个 applet), 指导读者使用简易

的 JDK、可启用 Java 的文本编辑器和一个 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（Visual Basic）的程序员，仔细地阅读这一章是非常必要的。

第 4 章介绍面向对象程序设计（Object-Oriented Programming, OOP）是当今程序设计的主流，而 Java 是完全面向对象的。本章将介绍面向对象两个基本成分中最重要的——封装，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，还对如何正确地进行 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，应在进一步学习 Java 之前花一些时间了解 OOP 的有关概念。

第 5 章介绍 OOP 的另一部分——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 程序设计中的基础。Java 中的继承机制与 C++ 的继承机制十分相似。C++ 程序员只需关注两种语言的不同之处即可。

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 章的简单继承模型。掌握接口的使用将可以充分发挥 Java 的面向对象程序设计能力。本章还将介绍 Java 的一个有用的技术特性——内部类。内部类可以使代码更清晰、更简洁。

第 7 章开始细致地讨论应用程序设计。每一个 Java 程序员都应该了解一些图形用户界面程序设计的知识，本卷包含了其中的基本内容。本章将展示如何制作窗口、如何在窗口中绘图、如何用几何图形作画、如何用多种字体格式化文本和如何显示图像。

第 8 章详细讨论 AWT（Abstract Window Toolkit）的事件模型。本章介绍如何编写代码来响应鼠标点击或敲击键盘等事件，同时还介绍如何处理基本的 GUI 元素，如按钮和面板。

第 9 章详细讨论 Swing GUI 工具箱。Swing 工具箱允许建立一个跨平台的图形用户界面。本章将介绍如何建立各种各样的按钮、文本组件、边界、滑块、列表框、菜单和对话框等。

一些更高级的组件将在卷 II 中讨论。

第 10 章阐述如何部署自己编写的应用程序或 applet。描述如何将应用程序打包到 JAR 文件中，以及如何使用 Java 的 Web Start 机制在 Internet 上发布应用程序。最后解释 Java 程序部署之后如何存储、检索配置信息。

第 11 章讨论异常处理，即 Java 的健壮机制，它用于处理调试好的程序可能出现的意外情况。异常提供了一种将正常的处理代码与错误处理代码分开的有效手段。当然，即使程序包含处理所有异常情况的功能，依然有可能无法按照预计的方式工作。这一章的后半部分，将给出大量的实用调试技巧。最后，讲述如何使用各种工具完成一个示例程序。

第 12 章介绍泛型程序设计，这是 Java SE5.0 的一项重要改进。泛型程序设计使程序

拥有更好的可阅读性和安全性。本章展示如何使用强类型机制,而舍弃不安全的强制类型转换,以及如何处理与旧版本 Java 兼容而带来的复杂问题。

第 13 章介绍 Java 平台的集合框架。当需要将大量对象收集到一起,并在过后要对它们进行检索时,可能会想要使用最合适的方法——集合,而不仅仅是将这些元素放置在数组中。本章介绍如何使用预先建立好的标准集合。

第 14 章是本书的最后一章。本章介绍多线程,这是一种可以让程序任务并行执行的特性(线程是程序中的控制流),并阐述如何建立线程、如何处理线程的同步问题。从 JavaSE 5.0 开始,多线程有了很大的改进,本章将介绍所有这些新的机制。

附录列出了 Java 语言的保留字。

约定

本书使用以下图标表示特殊内容。

NOTE (注释): 这个图标表示为正文提供的“注释”信息。

TIP (提示): 这个图标表示为正文提供的“提示”信息。

CAUTION (警告): 这个图标表示为正文提供的“警告”信息。

C++ NOTE (C++注释): 在本书中有许多用来解释 Java 与 C++ 的不同的 C++ 注释。对于没有 C++ 程序设计背景,或者不擅长 C++ 程序设计的程序员,可以跳过这些注释。

(应用程序编程接口): Java 带有一个很大的程序设计库,即应用程序编程接口(API)。第一次调用 API 时,将会在这一节的结尾给出一个概要描述,并标有 API 图标。这些描述十分通俗易懂,希望能够比联机 API 文档提供更多的信息。我们在每一个 API 注释特性上都标记了版本号,用来提示那些不希望使用 Java “风险”版本的读者。

程序源代码按照下列格式给出:

Listing 2-4 WelcomeApplet.Java

示例代码

本书的网站 <http://www.phptr.conl/coreJava> 以压缩文件的形式提供了书中的所有示例代码。可以用相应的解压缩程序或者用 Java 开发工具箱中的 .jar 实用程序展开这个文件。有关安装 Java 开发工具箱和示例程序的详细信息请参看第 2 章。

Contents

<i>1</i>	AN INTRODUCTION TO JAVA	1
	Java As a Programming Platform	2
	The Java "White Paper" Buzzwords	2
	Simple	3
	Object Oriented	3
	Network-Savvy	4
	Robust	4
	Secure	5
	Architecture Neutral	5
	Portable	6
	Interpreted	6
	High Performance	6
	Multithreaded	7
	Dynamic	7

Java Applets and the Internet	7
A Short History of Java	9
Common Misconceptions about Java	11

2 THE JAVA PROGRAMMING ENVIRONMENT 15

Installing the Java Development Kit	16
Downloading the JDK	16
Setting the Execution Path	18
Installing the Library Source and Documentation	20
Installing the <i>Core Java</i> Program Examples	20
Navigating the Java Directories	21
Choosing a Development Environment	21
Using the Command-Line Tools	22
Troubleshooting Hints	24
Using an Integrated Development Environment	25
Locating Compilation Errors	27
Running a Graphical Application	28
Building and Running Applets	31

3 FUNDAMENTAL PROGRAMMING STRUCTURES IN JAVA 35

A Simple Java Program	36
Comments	39
Data Types	40
Integer Types	40
Floating-Point Types	41
The char Type	42
The boolean Type	44
Variables	44
Initializing Variables	45
Constants	46
Operators	46
Increment and Decrement Operators	47

Relational and boolean Operators	48
Bitwise Operators	49
Mathematical Functions and Constants	49
Conversions between Numeric Types	51
Casts	51
Parentheses and Operator Hierarchy	52
Enumerated Types	53
Strings	53
Substrings	54
Concatenation	54
Strings Are Immutable	54
Testing Strings for Equality	56
Code Points and Code Units	56
The String API	57
Reading the On-Line API Documentation	59
Building Strings	62
Input and Output	63
Reading Input	63
Formatting Output	65
File Input and Output	69
Control Flow	71
Block Scope	71
Conditional Statements	72
Loops	76
Determinate Loops	80
Multiple Selections—The switch Statement	84
Statements That Break Control Flow	86
Big Numbers	88
Arrays	90
The “for each” Loop	91
Array Initializers and Anonymous Arrays	92
Array Copying	93
Command-Line Parameters	95

Array Sorting	95
Multidimensional Arrays	99
Ragged Arrays	102

4

OBJECTS AND CLASSES 105

Introduction to Object-Oriented Programming	106
Classes	107
Objects	108
Identifying Classes	108
Relationships between Classes	109
Using Predefined Classes	111
Objects and Object Variables	111
The GregorianCalendar Class of the Java Library	114
Mutator and Accessor Methods	115
Defining Your Own Classes	122
An Employee Class	122
Use of Multiple Source Files	125
Dissecting the Employee Class	125
First Steps with Constructors	126
Implicit and Explicit Parameters	127
Benefits of Encapsulation	129
Class-Based Access Privileges	131
Private Methods	132
Final Instance Fields	132
Static Fields and Methods	132
Static Fields	132
Static Constants	133
Static Methods	134
Factory Methods	135
The main Method	135
Method Parameters	138
Object Construction	144
Overloading	144

Default Field Initialization	145
Default Constructors	145
Explicit Field Initialization	146
Parameter Names	147
Calling Another Constructor	148
Initialization Blocks	148
Object Destruction and the finalize Method	152
Packages	153
Class Importation	153
Static Imports	155
Addition of a Class into a Package	155
Package Scope	158
The Class Path	160
Setting the Class Path	161
Documentation Comments	162
Comment Insertion	162
Class Comments	163
Method Comments	164
Field Comments	164
General Comments	164
Package and Overview Comments	165
Comment Extraction	166
Class Design Hints	167
5 INHERITANCE	171
Classes, Superclasses, and Subclasses	172
Inheritance Hierarchies	179
Polymorphism	179
Dynamic Binding	181
Preventing Inheritance: Final Classes and Methods	183
Casting	184
Abstract Classes	186
Protected Access	191

Object: The Cosmic Superclass	192
The equals Method	193
Equality Testing and Inheritance	194
The hashCode Method	196
The toString Method	198
Generic Array Lists	204
Accessing Array List Elements	206
Compatibility between Typed and Raw Array Lists	210
Object Wrappers and Autoboxing	211
Methods with a Variable Number of Parameters	214
Enumeration Classes	215
Reflection	217
The Class Class	217
A Primer on Catching Exceptions	219
Using Reflection to Analyze the Capabilities of Classes	221
Using Reflection to Analyze Objects at Runtime	226
Using Reflection to Write Generic Array Code	231
Method Pointers!	234
Design Hints for Inheritance	238
6 INTERFACES AND INNER CLASSES	241
Interfaces	242
Properties of Interfaces	247
Interfaces and Abstract Classes	248
Object Cloning	249
Interfaces and Callbacks	255
Inner Classes	258
Use of an Inner Class to Access Object State	260
Special Syntax Rules for Inner Classes	263
Are Inner Classes Useful? Actually Necessary? Secure?	264
Local Inner Classes	266
Accessing final Variables from Outer Methods	267

	Anonymous Inner Classes	269
	Static Inner Classes	271
	Proxies	275
	Properties of Proxy Classes	279
7	GRAPHICS PROGRAMMING	281
	Introducing Swing	282
	Creating a Frame	285
	Positioning a Frame	288
	Frame Properties	290
	Determining a Good Frame Size	290
	Displaying Information in a Component	294
	Working with 2D Shapes	299
	Using Color	307
	Using Special Fonts for Text	310
	Displaying Images	318
8	EVENT HANDLING	323
	Basics of Event Handling	324
	Example: Handling a Button Click	326
	Becoming Comfortable with Inner Classes	331
	Creating Listeners Containing a Single Method Call	334
	Example: Changing the Look and Feel	334
	Adapter Classes	338
	Actions	342
	Mouse Events	349
	The AWT Event Hierarchy	357
	Semantic and Low-Level Events	358
9	USER INTERFACE COMPONENTS WITH SWING	361
	Swing and the Model-View-Controller Design Pattern	362
	Design Patterns	362