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

根据Java SE 8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10版）

Core Java Volume I–Fundamentals
(10th Edition)

[美] 凯 S. 霍斯特曼（Cay S. Horstmann）著
周立新 陈波 叶乃文 邝劲筠 杜永萍 译

机械工业出版社
China Machine Press

Java
核心技术
系列

Java 核心技术 卷 I

基础知识（原书第10版）

Core Java Volume I—Fundamentals
(10th Edition)

[美] 凯 S. 霍斯特曼（Cay S. Horstmann）著
周立新 陈波 叶乃文 邝劲筠 杜永萍 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 10 版）/（美）凯 S. 霍斯特曼（Cay S. Horstmann）著；周立新等译 . —北京：机械工业出版社，2016.8
（Java 核心技术系列）
书名原文：Core Java Volume I—Fundamentals (Tenth Edition)

ISBN 978-7-111-54742-6

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

中国版本图书馆 CIP 数据核字（2016）第 211440 号

本书版权登记号：图字：01-2016-5145

Java 核心技术　卷 I　基础知识（原书第 10 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：董纪丽
印　　刷：北京诚信伟业印刷有限公司
版　　次：2016 年 9 月第 1 版第 1 次印刷
开　　本：186mm×240mm　1/16
印　　张：45.5
书　　号：ISBN 978-7-111-54742-6
定　　价：119.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88379426　88361066
购书热线：（010）68326294　88379649　68995259
投稿热线：（010）88379604
读者信箱：hzit@hzbook.com

译 者 序

书写 Java 传奇的 Sun Microsystems 曾经堪称“日不落”帝国，但服务器市场的萎缩却让这个声名赫赫的庞大帝国从蓬勃走向落寞。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！从 Sun 公司 2006 年 12 月发布 Java 6 后，经过 5 年多的不懈努力，终于在 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，但对于很多开发者来说，Java 8 却比 Java 7 来得更漫长一些。主要是因为 Oracle 原本计划在 2013 年发布正式版 Java 8，却因受困于安全性的问题经过了两次“跳票”。无论如何，如今 Java 8 来了，全新“革命”而不只是“进化”的功能将会让无数开发者动容。

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 9 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。经过几年的蛰伏，针对 Java 8 打造的《 Java 核心技术》第 10 版终于问世，这一版有了大幅的修订和更新，以反映 Java 8 增补、删改的内容。它将续写从前的辉煌，使人们能及时跟上 Java 前进的脚步。

本书由周立新、陈波等主译，程芳、刘晓兵、张练达、陈峰、江健、谢连宝、张雷生、杨健康、张莹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。特别需要说明的是，按照出版社的要求，这一版的翻译在老版本基础上完成，因此尤其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

2016 年 6 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。其原因在于它将有望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，得到了除微软之外的所有厂家的认可。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，已经发布了 Java 开发工具包（Java Development Kit）的 9 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 10 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，而且每一版都重新改写了部分内容，以便适应 Java 的最新特性。在这一版中，已经反映了 Java 标准版（Java SE 8 ）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
- 使用 Swing UI 工具箱进行图形用户界面设计
- 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（本书）集中介绍 Java 语言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
- 注释
- 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、可启用 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，仔细地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 是一种完全面向对象的语言。第 4 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应在进一步学习 Java 之前花一些时间了解 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。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。掌握接口可以充分获得 Java 的完全的面向对象程序设计能力。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*），即 Java 的一种健壮机制，用于处理可正常运行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，而舍弃不安全的强制类型转换，以及如何处理与旧版本 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，而不要简单地把这些元素放在一个数组中，这是这种情况下最适用的做法。这一章会介绍如何充分利用内建的标准集合。

第 10 章开始介绍 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。

第 11 章将详细讨论抽象窗口工具包（*abstract window toolkit*，AWT）的事件模型。你会看到如何编写代码来响应事件，如鼠标点击事件或按键事件。同时，你还会看到如何处理基本的 GUI 元素，如按钮和面板。

第 12 章详细讨论 Swing GUI 工具包。Swing 工具包允许建立跨平台的图像用户界面。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 13 章介绍如何将程序部署为应用或 applet。在这里我们会描述如何将程序打包在 JAR 文件中，以及如何使用 Java Web Start 和 applet 机制在 Internet 上发布应用。另外还会解释 Java 程序部署之后如何存储和获取配置信息。

第 14 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，你往往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。

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。

注释：“注释”信息会用这样的“注释”图标标志。

提示：“提示”信息会用这样的“提示”图标标志。

警告：对于可能出现的危险，我们用一个“警告”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。对于没有 C++ 程序设计背景，或者不擅长 C++ 程序设计、把它当做一场噩梦不愿再想起的程序员来说，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 API 调用时，我们会在该节的结尾给出一个概要描述。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。类、接口或方法名后面的编号是介绍该特性的 JDK 版本号，如下例所示：

应用程序编程接口 1.2

程序（源代码见本书网站）以程序清单形式给出，例如：

程序清单 1-1　InputTest/InputTest.java

示例代码

本书网站 http://horstmann.com/corejava 以压缩的形式提供了书中的所有示例代码。可以用熟悉的解压缩程序或者用 Java 开发包中的 jar 实用程序解压这个文件。有关安装 Java 开发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，改写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。编著一本书让很多人耗费了很多心血，在此衷心地感谢《 Java 核心技术》编写小组的每一位成员。

Prentice Hall 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，我希望每一位都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，Prentice Hall 公司的 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，还要感谢 Dmitry Kirsanov 和 Alina Kirsanova 完成手稿的编辑和排版工作。我还要感谢早期版本中我的合作者，Gary Cornell，他已经转向其他的事业。

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阅小组，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阅专家包括：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）、Rakesh Dhoopar（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 Kimmet（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（consultant）、Mark Morrissey（Oregon 研究院）、Mahesh Neelakanta（Florida Atlantic 大学）、Hao Pham、Paul Philion、Blake Ragsdell、Stuart Reges（Arizona 大学）、Rich Rosen（Interactive Data Corporation）、Peter Sanders（法国尼斯 ESSI 大学）、Dr. Paul Sanghera（San Jose 州立大学 Brooks 学院）、Paul Sevinc（Teamup AG）、Devang Shah（Sun Microsystems）、Yoshiki Shibata、Bradley A. Smith、Steven Stelting（Oracle）、Christopher Taylor、Luke Taylor（Valtech）、George Thiruvathukal、Kim Topley（StreamingEdge）、Janet Traub、Paul Tyma（consultant）、Peter van der Linden、Christian Ullenboom、Burt Walsh、Dan Xu（Oracle）和 John Zavgren（Oracle）。

Cay Horstmann

2015 年 11 月于瑞士比尔

目　录

第 1 章　Java 程序设计概述

- ▲ Java 程序设计平台
- ▲ Java 发展简史
- ▲ Java “白皮书”的关键术语
- ▲ 关于 Java 的常见误解
- ▲ Java applet 与 Internet

1996 年 Java 第一次发布就引起了人们的极大兴趣。关注 Java 的人士不仅限于计算机出版界，还有诸如《纽约时报》《华盛顿邮报》《商业周刊》这样的主流媒体。Java 是第一种也是唯一一种在 National Public Radio 上占用了 10 分钟时间来进行介绍的程序设计语言，并且还得到了 $100 000 000 的风险投资基金。这些基金全部用来支持用这种特别的计算机语言开发的产品。重温那些令人兴奋的日子是很有意思的。本章将简要地介绍一下 Java 语言的发展历史。

1.1　Java 程序设计平台

本书的第 1 版是这样描写 Java 的：“作为一种计算机语言，Java 的广告词确实有点夸大其辞。然而，Java 的确是一种优秀的程序设计语言。作为一个名副其实的程序设计人员，使用 Java 无疑是一个好的选择。有人认为：Java 将有望成为一种最优秀的程序设计语言，但还需要一个相当长的发展时期。一旦一种语言应用于某个领域，与现存代码的相容性问题就摆在了人们的面前。”

我们的编辑手中有许多这样的广告词。这是 Sun 公司高层的某位不愿透露姓名的人士提供的（Sun 是原先开发 Java 的公司）。Java 有许多非常优秀的语言特性，本章稍后将会详细地讨论这些特性。由于相容性这个严峻的问题确实存在于现实中，所以，或多或少地还是有一些“累赘”被加到语言中，这就导致 Java 并不如想象中的那么完美无瑕。

但是，正像我们在第 1 版中已经指出的那样，Java 并不只是一种语言。在此之前出现的那么多种语言也没有能够引起那么大的轰动。Java 是一个完整的平台，有一个庞大的库，其中包含了很多可重用的代码和一个提供诸如安全性、跨操作系统的可移植性以及自动垃圾收集等服务的执行环境。

作为一名程序设计人员，常常希望能够有一种语言，它具有令人赏心悦目的语法和易于理解的语义（C++ 不是这样的）。与许多其他的优秀语言一样，Java 完全满足了这些要求。有些语言提供了可移植性、垃圾收集等，但是，没有提供一个大型的库。如果想要有奇特的绘图功能、网络连接功能和数据库存取功能就必须自己动手编写代码。Java 具备所有这些特性，它是一种功能齐全的出色语言，是一个高质量的执行环境，还提供了一个庞大的库。正是因为它集多种优势于一身，所以对广大的程序设计人员有着不可抗拒的吸引力。