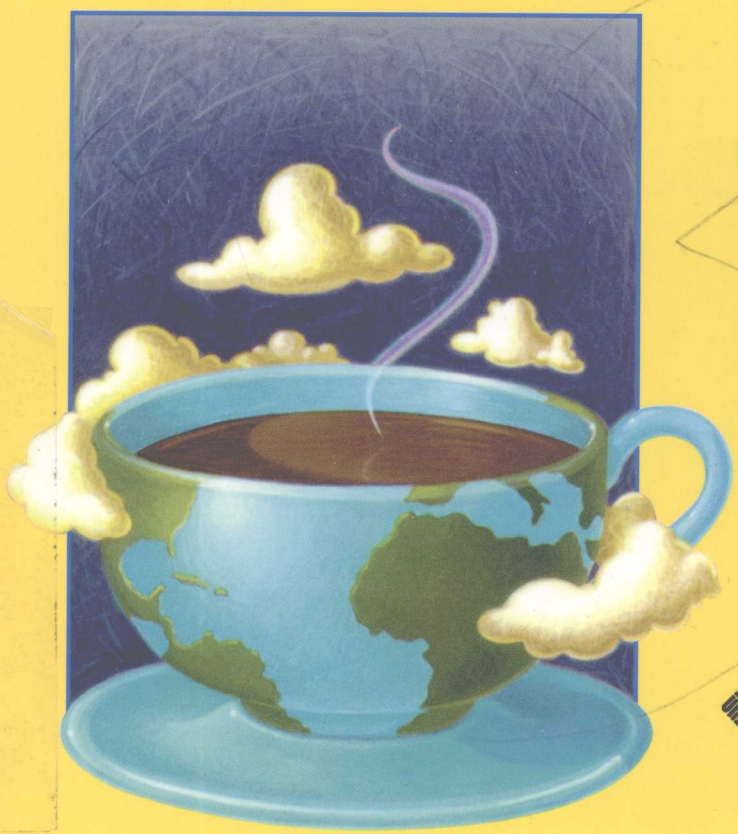
HZ BOOKS
PRENTICE HALL

Sun公司核心技术丛书

JAVA

核心技术 卷Ⅰ：基础知识

Core Java , Volume Ⅰ : Fundamentals Eighth Edition



（美）Cay S. Horstmann
Gary Cornell 著

叶乃文 邝劲筠 杜永萍 译

- 针对Java SE6平台进行了全面更新。
- 涵盖Java语言核心内容。
- 大量精心设计代码示例。
- CSDN Java大版主隆重推荐。

Sun microsystems

机械工业出版社
China Machine Press

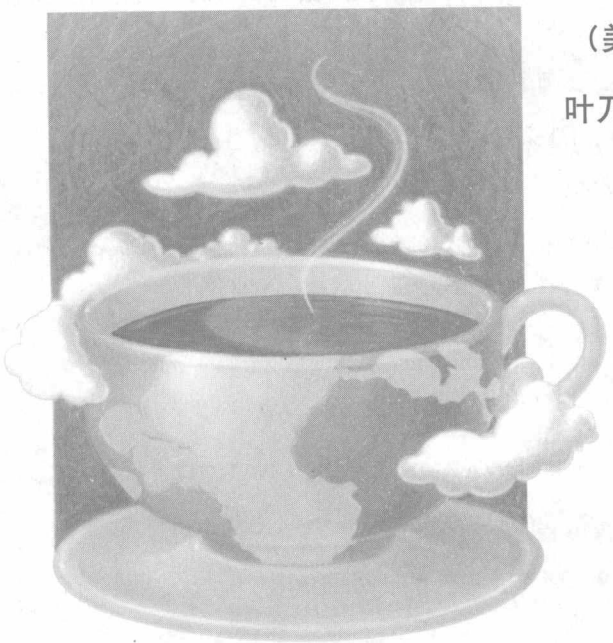
原书第8版

Sun公司核心技术丛书

JAVA

核心技术 卷Ⅰ：基础知识

Core Java , Volume Ⅰ : Fundamentals Eighth Edition



(美) Cay S. Horstmann
Gary Cornell 著

叶乃文 邝劲筠 杜永萍 译

机械工业出版社
China Machine Press

原书第8版

《Java核心技术》出版以来一直畅销不衰，深受读者青睐，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。本版也不例外，它反映了Java SE 6的新特性。全书共14章，包括Java基本的程序结构、对象与类、继承、接口与内部类、图形程序设计、事件处理、Swing用户界面组件、部署应用程序和Applet、异常日志断言和调试、泛型程序设计、集合以及多线程等内容。

全书对Java技术的阐述精确到位，叙述方式深入浅出，并包含大量示例，从而帮助读者充分理解Java语言以及Java类库的相关特性。

本书版权登记号：图字：01-2008-2682

图书在版编目（CIP）数据

Java核心技术，卷Ⅰ：基础知识（原书第8版）/（美）昊斯特曼（Horstmann, C. S.）著；叶乃文，邝劲筠，杜永萍译. —北京：机械工业出版社，2008. 5

（Sun公司核心技术丛书）

书名原文：Core Java. Volume I, Fundamentals, Eighth Edition

ISBN 978-7-111-23950-5

Ⅰ. J…　Ⅱ. ①昊…　②叶…　Ⅲ. JAVA语言－程序设计　Ⅳ. TP312

中国版本图书馆CIP数据核字（2008）第053980号

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：李南丰　　周茂辉

北京京北制版厂印刷 · 新华书店北京发行所发行

2008年6月第1版第1次印刷

186mm × 240mm · 44.5印张

标准书号：ISBN 978-7-111-23950-5

定价：98.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线（010）68326294

译 者 序

《Java核心技术》自第1版出版以来，一直备受广大Java程序设计人员的青睐，是一本畅销不衰的Java经典书籍。本书的两位作者Cay S.Horstmann和Gary Cornell都具有编写程序设计方面书籍的丰富经验。

众所周知，Java程序设计语言仍处于不断完善和发展的活跃时期，为了能够及时地跟上Java 的前进步伐，在短短的10余年间，本书已经修订了7次，第8版同样是为了适应Java的最新特性而重新修订的。新版主要增加了对Java标准版（Java SE 6）特性的全面介绍，并对第7版中两卷的内容安排做了部分调整。即将第7版第Ⅰ卷中的“流与文件”调到第Ⅱ卷中，将第7版第Ⅱ卷中的“集合”与“多线程”调到第Ⅰ卷中。

我们诚心地向您推荐这本书籍。它囊括了Java 2平台、标准版（J2SE）的全部基础知识。作为一本精练的技术指南和可信赖的参考书籍，其中提供了大量的应用实例，用来说明Java的重要语言规则和库功能，而且，这些示例程序都是完整且具有实际意义的。最重要的是：所有程序都已经被升级为Java SE 6，它们将成为独自编写Java程序的良好开端。

参加本书翻译的有叶乃文、邝劲筠以及杜永萍。书中文字与内容力求忠实原著，但限于译者水平有限，加上时间仓促，译文中难免有疏漏之处，敬请广大读者批评指正。

译者

2008年4月于北京

前 言

致读者

1995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，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，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 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地跳动文本）。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（本书）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（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，通过编译和运行三个典型的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（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章介绍类与封装仅仅是OOP中的一部分，本章将介绍另一部分——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获得Java完全的面向对象程序设计的能力。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，本卷中包含了其中的基本内容部分。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以及如何显示图像。

第8章详细讨论AWT（Abstract Window Toolkit ）的事件模型。我们将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，还将介绍如何处理基本的GUI元素，比如：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、菜单以及对话框等等。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。在这里将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，将解释Java程部署之后如何存储、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的情况。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章概要介绍泛型程序设计，这是Java SE5.0的一项重要改进。泛型程序设计使得程序拥有更好的可阅读性和安全性。在这里，将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，可能会想要使用集合，这是目前最为合适的，它取代了将这些元素放置在数组中。本章将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。在这章中，将介绍多线程，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Java SE 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。

注释：这个图标表示为正文提供的“注释”信息。

提示：这个图标表示为正文提供的“提示”信息。

警告：这个图标表示为正文提供的“警告”信息。

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。

致　谢

写一本书需要投入大量的精力，改写一本书也并不是想像的那样轻松，尤其是Java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和Sun Microsystems出版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一直给予了指导，同时感谢那些不知其姓名的为本书做出贡献的幕后人士。感谢Vanessa Moore提供了优秀的产品支持。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（《C/C++Users Journal》的特约编辑），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 (曼哈顿学院), Rakesh Dhoopar (Oracle), David Geary (Sabreware), Brian Goetz (Quiotix公司首席顾问), Angela Gordon (Sun Microsystems), Dan Gordon (Sun Microsystems), Rob Gordon, John Gray (哈特福大学), Cameron Gregory (olabs.com), Marty Hall (约翰斯·霍普金斯大学应用物理实验室), Vincent Hardy (Sun Microsystems), Dan Harkey (圣约瑟州立大学), 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 (麦吉尔大学), Mark Lawrence, Doug Lea (SUNY Oswego), Gregory Longshore, Bob Lynch (Lynch Associates), Philip Milne (顾问), Mark Morrissey (俄勒冈研究院), Mahesh Neelakanta (佛罗里达大西洋大学), Hao Pham, Paul Philion, Blake Ragsdell, Stuart Reges (亚利桑那大学), Rich Rosen (Interactive Data Corporation), Peter Sanders (ESSI 大学, Nice, France), Dr. Paul Sanghera (圣约瑟州立大学与布鲁克斯学院), Paul Sevinc (Teamup AG), Devang Shah (Sun Microsystems), Bradley A. Smith, Steven Stelting (Sun Microsystems), Christopher Taylor, Luke Taylor (Valtech), George Thiruvathukal, Kim Topley (《Core JFC》的作者), Janet Traub, Paul Tyma (顾问), Peter van der Linden (Sun Microsystems), and Burt Walsh.

Cay Horstmann
2007于旧金山

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是第一种也是惟一的一种在National Public Radio上占用了10分钟时间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。然而，现在看起来，当初的这些预测还是有一定准确性的。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并不如想像中的那么完美无瑕。

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恰恰满足了这些要求。有些语言提供了可移植性、垃圾收集器等等，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这种功能齐全的出色语言，具有高质量的执行环境以及庞大的库。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。

1.2 Java“白皮书”的关键术语

Java的设计者已经编写了颇有影响力的“白皮书”，用来解释设计的初衷以及完成的情况，并且发布了一个简短的摘要。这个摘要用下面11个关键术语进行组织：

简单性
面向对象
网络技能（Network-Savvy）
健壮性
安全性
体系结构中立
可移植性
解释型
高性能
多线程
动态性

本节将论述下列主要内容：

- 给出白皮书中对每个关键术语的概述，这是Java设计者对相关术语的论述。
- 凭借Java当前版本的使用经验，给出对这些术语的理解。

注释：白皮书可以在http://java.sun.com/docs/white/langenv/上找到。对于11个关键术语的论述请参看http://java.sun.com/docs/overviews/java/java-overview-1.html。

1.2.1 简单性

人们希望构建一个无需深奥的专业训练就可以进行编程的系统，并且要符合当今的标准惯例。因此，尽管人们发现C++不太适用，但在设计Java的时候还是尽可能地接近C++，以便系统更易于理解。Java剔除了C++中许多很少使用、难以理解、易混淆的特性。在目前看来，这些特性带来的麻烦远远多于其带来的好处。

的确，Java语法是C++语法的一个“纯净”版本。这里没有头文件、指针运算（甚至指针语法）、结构、联合、操作符重载、虚基类等等（请参阅本书各个章节给出的C++注释，那里比较详细地解释了Java与C++之间的区别）。然而，设计者并没有试图清除C++中所有不适当的特性。例如，`switch`语句的语法在Java中就没有改变。如果知道C++就会发现可以轻而易举地将其转换成Java。

如果已经习惯于使用可视化的编程环境（例如Visual Basic），你就不会觉得Java简单了。Java有许多奇怪的语法（尽管掌握其要领并不需要很长时间），更重要的是，使用Java需要自己编写大量的程序。Visual Basic的魅力在于它的可视化设计环境几乎自动地为应用程序提供了大量的基础结构。而使用Java实现同样的功能却需要手工地编制代码，通常代码量还相当大。然而，已经有一些支持“拖放”风格程序开发的第三方开发环境。

简单的另一个方面是小。Java的目标之一是支持开发能够在小型机器上独立运行的软件。基本的解释器以及类支持大约仅为40KB；再加上基础的标准类库和对线程的支持（基本上是一个自包含的微内核）大约需要增加175KB。

在当时，这是一个了不起的成就。当然，由于不断的扩展，类库已经相当庞大了。现在有一个独立的具有较小类库的Java微型版（Java Micro Edition）用于嵌入式设备。