



# 最新 Java

# 2

## 核心技术 卷I: 原理 (原书第5版)



### Core Java 2 Volume I Fundamentals



Cay S. Horstmann 著  
(美) Gary Cornell

李如豹 刚冬梅 等译



机械工业出版社  
China Machine Press

PH  
PTR

Sun公司核心技术丛书

# 最新Java 2核心技术

## 卷I：原理

(原书第5版)

(美) Cay S.Horstmann 著  
Gary Cornell

李如豹 刚冬梅 等译  
前导工作室 审校



机械工业出版社  
China Machine Press

本书覆盖了Java 2平台标准版1.3的基础知识,对面向对象的Java开发、Swing用户界面组件以及其他一些技术进行了深入讨论。内容包括:面向对象的程序设计、反射和代理、接口和内部类、事件监听器模型、使用Swing GUI工具包的图形用户界面设计、异常处理、流输入/输出和对象的序列化。

本书内容全面、深入浅出,极具实用价值。随书所附光盘包括全部源代码、J2SE 1.3版、Forte for Java团体版等非常有价值的资料。本书为Java程序设计员提供了很好的指导。

Authorized translation from the English language edition, entitled Core Java 2 Volume I: Fundamentals, 1 by Cay S.Horstmann and Gary Cornell, published by Pearson Education, Inc., publishing as Sun Microsystems, Inc. Copyright 2001.

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 systems, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION NORTH ASIA LTD and CHINA MACHINE PRESS, Copyright 2002.

This edition is authorized for sale only in People's Republic of China(excluding the Special Administrative Region of Hong Kong and Macau).

版权所有,侵权必究

本书版权登记号:图字:01-2001-5018

图书在版编目(CIP)数据

最新Java 2核心技术卷1:原理(原书第5版)/(美)霍斯特曼(Horstmann, C.S.),  
(美)科奈尔(Cornell, G.)著;李如豹等译.-北京:机械工业出版社,2002.1  
(Sun公司核心技术丛书)

书名原文:Core Java 2 Volume I: Fundamentals

ISBN 7-111-09583-9

I.最… II.①霍… ②科… ③李… III.Java语言-程序设计 IV.TP312

中国版本图书馆CIP数据核字(2001)第074450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:路晓村 张鸿斌

北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

2002年2月第1版第1次印刷

787mm×1092mm 1/16·40.75印张

印数:0 001-4 000册

定价:78.00元(附光盘)

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

# 译者序

作为一个Java程序设计人员，我阅读了很多有关Java的书籍和资料，也经常从网上找一些相关的资料。也许您会发现，在计算机书店中，关于Java的书比比皆是。留心研究，不难发现，这些书大体上可分为两类：一类是手册性书籍，基于某种特定Java环境的使用指南；另一类是教程类用书，从语法、语义到语句由浅入深地讲解。如果您想对Java深入了解一下，您也许经常会感到头疼，为什么就没有适合我的Java图书呢？作为本书译者，我要告诉您的是，也许本书能够满足您的部分要求。至少从我的感觉看来，本书与其他书比起来有一些特别。毕竟，它讲的是“核心技术”！所以，您有足够的理由相信，它讲得更加深入、更加专业。

如果您不仅仅只是想学会怎么来使用Java进行编程，也就是说，您不是一个“普通”的程序员，那么您可能会关心哪些方面的核心技术呢？无外乎是下列几点：反射和代理、接口和内部类、异常处理、流输入/输出和对象的序列化、多线程、网络程序设计、分布式对象、集合类、数据库、高级图形、高级GUI组件、国际化和本地化、JavaBean等内容。而本书所涉及到的内容都可以满足您的“胃口”。请允许我为您简单地介绍一下本书的基本内容：

- 第1章      Java简介
- 第2章      Java编程环境
- 第3章      Java基本编程结构
- 第4章      对象和类
- 第5章      继承
- 第6章      接口和内部类
- 第7章      图形编程
- 第8章      事件处理
- 第9章      Swing用户界面组件
- 第10章     a.pplet
- 第11章     异常和调试
- 第12章     流和文件

我想本书特别重要的一点是，它能够深入地对每项技术进行透彻地讲解，而不仅仅只是停留在教您如何使用的表面层次上，这是本书最值得推荐的一点。译者认为，无论是作为正式教材还是作为自学用书，这本书都是非常优秀的。特别是对有一定Java基础，尤其是对想要在Java程序方面有一些“造诣”的程序员来说，这本书更加适宜。

我们深深地感谢我们的家人和朋友。在翻译过程中，他们给予了我们莫大的关心、支持和帮助。

全书由李如豹、刚冬梅、许崇梅、张雪莲、张勇、鲁金贵、邓勃、钟明辉、董金云、王大

军、陈蓓、魏伟、来欣、王建设、文杨、董立伟、田宇明、张洁、肖国尊、王乐春、易晓东、李萍、冯田力等进行翻译，由钱松进行全书的术语、技术问题支持，前导工作室全体工作人员共同完成了本书的翻译、录排、校对等工作。本书最后由李如豹、刚冬梅、钱松统稿。由于时间仓促，且译者的水平有限，在翻译过程中难免会出现一些错误，请读者批评指正。

如果您在阅读中碰到了什么问题，请同我们前导工作室联系：[qiandao@263.net](mailto:qiandao@263.net)。我们会尽力解决您的问题。

2001年10月

# 前 言

1995年底，Java程序设计语言闯入了Internet领域，并迅速占据了显著地位。Java美好的前景在于它将成为把用户和信息连接起来的“万能胶”，不管那些信息是来自Web服务器、数据库、信息提供商，还是任何其他你想像得出来的信息来源。实际上Java正处于一个实现这个前景的独特位置。它是一种非常可靠的设计语言，得到了除Microsoft外所有的主要软件提供商的支持。它内建的安全和保护特性打消了程序员和程序用户的顾虑。Java甚至内建直接对高级程序设计任务的支持，如网络编程、数据库连接和多线程等。

目前Sun Microsystems总共发布了四个Java软件开发包（Java Software Development Kit）主要的修订版本。1.02版发布于1996年，提供了对数据库连接和分布式对象的支持。1.1版发布于1997年，增加了一个强大的事件模型、国际化和Java Beans组件模型。1.2版发布于1998年底，在多方面都有了改进，但是一个主要的改进是：“Swing”用户界面工具包（User Interface Toolkit），它最终使用户可以开发真正的可移植的图形用户界面（GUI）程序。1.3版发布于2000年春季，又有了更多的改进。

本书是第5版。每当新版本的Java开发包发布后，我们都重写这本书，以使读者能从最新的Java特性中获益。

像这本书以前的版本一样，我们仍然着眼于那些希望将Java应用于实际项目中真正的程序开发者。我们仍然保证本书中不会出现模棱两可的文字和莫名其妙的字符。我们假定本书读者是具有坚实的程序设计语言基础的程序员。但是你不必了解C++或面向对象程序设计（Object-oriented programming）。根据我们获得的对本书早期版本的反馈，我们相信有经验的Visual Basic、C或COBOL程序员读这本书不会有困难。（你甚至不需要有任何在Windows、UNIX或Macintosh下建立图形用户界面的经验。）

我们假定你希望：

- 编写真正的代码来解决实际的问题。
- 不喜欢充斥着玩具例子的图书（比如说厨房用具或果树）。

在随书光盘上你能得到大量代码，几乎演示了我们讨论过的所有语言和库的特性。我们有意使用简单的例子来突出要点，但是大多数例子既不是赝品，也没有偷工减料。它们能为你自己的代码开一个好头。

我们假定你想要甚至渴望学习Java为你提供的所有高级特性。例如，我们将为你提供如下的详细内容：

- 面向对象的程序设计。
- 反射和代理。
- 接口和内部类。

- 事件监听器模型。
- 使用Swing GUI工具包的图形用户界面设计。
- 异常处理。
- 流输入/输出和对象的序列化。

我们没有在那些仅仅是为了丰富Web页,虽然有趣但并不十分严肃的Java程序上花很多时间。这方面的书籍很多,我们推荐John Pew的《Instant Java》一书,它也是由Sun Microsystems出版社和Prentice Hall出版的。

最后,随着Java类库爆炸式的增长,单卷本的图书已经无法涵盖真正的程序员必须掌握Java所有特性。我们决定将这本书分为两卷。卷I,就是你手中的这本书,集中讲述了Java语言的基本概念。卷II则更进一步讲述企业级特性和高级用户界面编程。它包括如下的详细内容:

- 多线程。
- 网络程序设计。
- 分布式对象。
- 集合类。
- 数据库。
- 高级图形。
- 高级GUI组件。
- 国际化。
- 本地方法。
- JavaBean。

在写一本书时,错误和不准确是难免的,我们非常乐意找出这些错误。但是我们当然也希望同一个问题只被告知一次。我们建立了一个Web页,它包括常见问题列表、Bug修正列表和工作区,它的URL是:<http://www.horstmann.com/core-java.html>。FAQ的后面是用来提交报告Bug和提出建议的表格(这么做是希望你能够先浏览一遍FAQ)。我们会认真地阅读每一封来信,但可能没有时间回复每一封信。感谢你对这本书将来版本的改进所提供的任何建议。

我们希望你觉得这本书生动有趣且能对你的Java程序设计带来帮助。

## 关于本书

第1章是对Java区别于其他程序设计语言的能力作一个总体描述。我们将解释这个语言设计者的设计初衷,在哪些方面实现了他们的设想。然后,我们简要叙述一下Java是如何诞生和发展的。

在第2章中,我们将告诉你如何安装Java和配书光盘上的软件。然后我们将向你示范如何编译和运行三种典型的Java程序:控制台应用程序、图形应用程序、Applet程序。

第3章开始讨论Java语言。在这一章中,我们将涉及基础内容:变量、循环、简单函数。如果你是一个C或C++程序员,对你来说这会非常简单,因为Java语言的语法从本质上讲是和C相同的。如果你具有非C的编程背景,如Visual Basic或COBOL,你应该仔细阅读这一章。

面向对象的程序设计(Object-Oriented Programming, OOP)是目前应用中的主流方法,而

Java是完全面向对象的。第4章介绍了面向对象的两个最基本的构件——封装，以及Java语言对它的实现机制，也就是类和方法。除了Java语言规则外，我们还给出了正确运用OOP设计的建议。最后，我们介绍了javadoc这种非凡的工具，它能够把你的代码中的注释格式化成一组Web页的超级链接。如果你熟悉C++，那么你可以快速浏览一下这一章。没有面向对象程序设计背景的程序员在进行下一步学习前，应当花一些时间来掌握OOP的概念。

类和封装仅仅是OOP技术中的一部分，第5章介绍了另一种技术，也就是继承。继承使你能够利用一个已有的类并根据你的需要修改它。这也是Java程序设计中一项基本技术。Java中的继承技术与C++中的继承非常类似。C++程序员可以把重点放在两种语言的不同之处。

第6章教你如何运用Java接口的概念。接口使你可以超越第5章中对继承模型的简单使用。掌握接口使你可以充分使用Java完全面向对象的方法的程序设计能力。这里我们也提到了Java的一项很有用的技术特征，也就是所谓的内部类。内部类可以使你的代码更易读、简洁。

在第7章中，我们开始认真讨论应用程序设计。我们将示范如何创建窗口，如何绘制它们，如何绘制几何图型，如何用多种字体格式化文本，以及如何显示图像。

第8章详细讨论了AWT的事件模型，即抽象窗口工具箱（Abstract Window Toolkit，AWT）。我们讨论的是Java 1.1版中新加入的事件模型，不是1.0版中的已经过时的和过于简单的事件模型。你将看到如何编写响应事件的代码，如鼠标点击和击键。同时，你还将学会如何处理GUI元素如按钮和面板。

第9章非常详细地讨论了Swing GUI工具包。Swing工具包使你能够建立跨平台的图形用户界面。你将学习各种的按钮、文本组件、边界、滑块、列表框、菜单以及对话框的所有技术。然而，一些更高级的组件将在卷II中讨论。

完成了第9章后，你便具备了编写applet程序的所有基础，那些微型程序可以嵌入到Web页中，所以第10章的主题就是applet。我们向你演示一些有用的也是很有趣的applet，但更重要的是我们将向你解释它们的内部机制。而且我们还将向你演示如何利用Java插件运行采用了Java最新特性的applet，即使你的用户使用的是老版本的或是怀有敌意的软件提供商的浏览器。

第11章讨论异常处理，即Java的健壮性机制，用来处理这种情况：好的代码也会发生错误。例如，网络连接在文件下载中间可能变得不可用，磁盘可能会装满等等。异常可以让你把正常的代码和错误的处理分开。当然，即使你在你的程序中加入了所有的异常处理代码，它仍然会像预计的那样无法工作。这一章的第二部分我们将给出许多有用的调试技巧。最后，我们将引导你使用各种工具完成一个示例任务，这些工具包括JDB调试器、Forte开发环境调试器、性能统计器、代码覆盖测试工具以及AWT机器人。

我们用输入/输出处理来结束这本书。在Java中，所有的I/O都通过所谓的流来处理。流使你用一个统一的方式来处理任何来源的数据，如文件、网络连接或内存块。我们包括了详细的读取器和写入器类，它们使得Unicode的处理更为简单；而且我们还要介绍当你使用对象序列化机制时的底层实现，它使得存储和装入对象简单、方便。

附录中列出了Java语言的关键字。

## 约定

像很多其他计算机书籍一样，我们使用courier型字体来表示计算机代码。



这里是用来解释Java和C++之间的区别。如果你没有C++背景，或者你觉得你的C++的编程经历对你来说像一场不愿提及的噩梦，那么你可以略过它们。

通常Java有一个很大的编程库或应用编程接口（API）。当第一次使用一个API调用时，我们会在这一段的结尾处加上一段简要描述，用API来表示。这些描述不太正式，但是我们希望它们能比正式的API联机文档容易理解。

### 关于CD-ROM

书后附送的光盘中包含最新版的Java软件开发包（Java Software Development Kit）。在我们编写本书时，它只能运行在Windows 95/NT或Solaris 2平台上。

当然光盘上也包括了书上所有示例代码，以压缩文件的格式存放。你可以用常见的解压缩程序或Java软件开发包附带的工具jar来解压缩。

光盘上还有一些“一流”的程序，你也许会发现它们对你的开发有帮助。一般说来，如果超过了试用期，这些程序的提供商会要求你付一些费用，我们与这些提供商没有关系，如果你有任何关于这些程序的问题，请直接联系这些提供商。

**注意** 人们经常会问，在商业应用中使用这些示例代码有没有什么许可要求。在非商业应用中你可以自由使这本书中的代码。然而，如果你想将这些代码作为商业产品的基础，我们要求开发团队中的Java程序员人手一册这本书，这是我们惟一的要求。

Cay Horstmann  
Cupertino, 2000年11月

# 目 录

译者序	
前言	
第1章 Java简介	1
1.1 Java是一种编程工具	1
1.2 Java的优点	2
1.3 Java的关键特点	2
1.3.1 简单	3
1.3.2 面向对象	3
1.3.3 分布式	4
1.3.4 健壮性	4
1.3.5 安全	4
1.3.6 中立体系结构	5
1.3.7 可移植性	5
1.3.8 解释型	6
1.3.9 高性能	6
1.3.10 多线程	6
1.3.11 动态	7
1.4 Java和因特网	7
1.4.1 实际applet	8
1.4.2 服务器端的Java	9
1.5 Java简史	9
1.6 常见的Java误解	10
第2章 Java编程环境	14
2.1 安装Java软件开发工具箱	14
2.1.1 设置执行路径	14
2.1.2 安装库源文件和文档	15
2.1.3 安装Java核心技术的程序例子	16
2.1.4 浏览Java目录	16
2.2 开发环境	17
2.3 使用命令行工具	17
2.4 使用集成开发环境	20
2.5 从文本编辑器中编译和运行程序	22
2.6 图形化应用程序	25
2.7 Applet	28
第3章 Java基本编程结构	33
3.1 简单的Java程序	33
3.2 注释	35
3.3 数据类型	36
3.3.1 整型	36
3.3.2 浮点类型	37
3.3.3 字符类型	38
3.3.4 布尔类型	39
3.4 变量	39
3.5 赋值和初始化	40
3.6 运算符	41
3.6.1 递增和递减运算符	42
3.6.2 关系和布尔运算符	43
3.6.3 位运算符	43
3.6.4 数学函数和常量	44
3.6.5 数据类型之间的转换	45
3.6.6 造型	46
3.6.7 圆括号和运算符级别	46
3.7 字符串	47
3.7.1 串连接	47
3.7.2 子串	48
3.7.3 串编辑	48
3.7.4 测试串是否相等	49
3.7.5 阅读在线API文档	51
3.7.6 读取输入	53
3.7.7 格式输出	55
3.8 控制流程	58
3.8.1 块作用域	58
3.8.2 条件语句	59
3.8.3 不确定循环	62

3.8.4 确定循环 .....	65	4.6 对象构造 .....	124
3.8.5 多种选择——switch语句 .....	69	4.6.1 重载 .....	124
3.8.6 中断控制流程 .....	70	4.6.2 默认字段初始化 .....	125
3.9 大数字 .....	73	4.6.3 默认构造器 .....	125
3.10 数组 .....	75	4.6.4 显式字段初始化 .....	126
3.10.1 数组初始化和匿名数组 .....	76	4.6.5 参数名 .....	127
3.10.2 拷贝数组 .....	76	4.6.6 调用另一个构造器 .....	128
3.10.3 命令行参数 .....	78	4.6.7 初始化块 .....	128
3.10.4 对数组排序 .....	79	4.6.8 对象析构和finalize方法 .....	132
3.10.5 多维数组 .....	82	4.7 包 .....	132
3.10.6 不规则数组 .....	84	4.8 文档注释 .....	139
第4章 对象和类 .....	88	4.8.1 如何插入注释 .....	139
4.1 面向对象程序设计入门 .....	88	4.8.2 类注释 .....	140
4.1.1 OOP术语 .....	89	4.8.3 方法注释 .....	141
4.1.2 对象 .....	90	4.8.4 字段注释 .....	141
4.1.3 类之间的关系 .....	91	4.8.5 常见的注释 .....	141
4.1.4 OOP和传统过程化程序设计 技术的比较 .....	92	4.8.6 包和概要注释 .....	143
4.2 使用现有类 .....	94	4.8.7 如何提取注释 .....	143
4.2.1 对象和对象变量 .....	94	4.9 类设计技巧 .....	143
4.2.2 Java库中的GregorianCalendar类 .....	97	第5章 继承 .....	146
4.3 创建自己的类 .....	103	5.1 扩展类 .....	146
4.3.1 Employee类 .....	103	5.1.1 继承层次图 .....	152
4.3.2 使用多个源文件 .....	106	5.1.2 多态 .....	153
4.3.3 分析Employee类 .....	107	5.1.3 动态绑定 .....	153
4.3.4 创建构造器的第一步 .....	107	5.1.4 防止继承: Final类和方法 .....	155
4.3.5 Employee类的方法 .....	109	5.1.5 造型转换 .....	156
4.3.6 对私有数据访问的方法 .....	112	5.1.6 抽象类 .....	158
4.3.7 私有方法 .....	112	5.1.7 受保护访问 .....	162
4.3.8 Final实例字段 .....	113	5.2 Object: 所有类的超类 .....	163
4.4 静态字段和方法 .....	113	5.2.1 equals和toString方法 .....	163
4.4.1 静态字段 .....	113	5.2.2 通用编程 .....	170
4.4.2 常量 .....	114	5.2.3 数组列表 .....	172
4.4.3 静态方法 .....	115	5.2.4 对象包装器 .....	178
4.4.4 工厂方法 .....	116	5.3 Class类 .....	181
4.4.5 main方法 .....	116	5.4 反射 .....	184
4.5 方法参数 .....	119	5.4.1 使用反射分析类的功能 .....	184
		5.4.2 在运行时使用反射分析对象 .....	188

5.4.3 使用反射编写通用数组代码	194	8.4.4 鼠标事件	308
5.4.4 方法指针	197	8.5 动作	316
5.5 设计继承的提示	201	8.6 多点传送	323
第6章 接口和内部类	203	8.7 事件队列	327
6.1 接口	203	第9章 Swing用户界面组件	336
6.1.1 接口的属性	207	9.1 模型 - 视图 - 控制器设计模式	336
6.1.2 接口和抽象类	208	9.2 布局管理器概述	341
6.1.3 接口和回调	209	9.2.1 边界布局	343
6.2 对象克隆	212	9.2.2 面板	344
6.3 内部类	217	9.3 文本输入	345
6.3.1 使用内部类来访问对象状态	218	9.3.1 文本域	346
6.3.2 内部类的特殊语法规则	222	9.3.2 输入确认	352
6.3.3 内部类是否有用	223	9.3.3 密码域	360
6.3.4 局部内部类	225	9.3.4 文本区	360
6.3.5 静态内部类	230	9.3.5 标签和标签组件	364
6.4 代理	233	9.3.6 选择文本	365
第7章 图形编程	239	9.3.7 编辑文本	366
7.1 Swing概述	239	9.4 选择组件	368
7.2 创建一个框架	242	9.4.1 复选框	368
7.3 框架定位	245	9.4.2 单选按钮	371
7.4 在一个面板中显示信息	249	9.4.3 边界	375
7.5 2D图形	254	9.4.4 组合框	379
7.6 颜色	261	9.4.5 滑块	382
7.7 文本和字体	265	9.5 菜单	388
7.8 图像	274	9.5.1 创建菜单	389
第8章 事件处理	280	9.5.2 菜单项中的图标	391
8.1 事件处理基础	280	9.5.3 复选框和单选按钮菜单项	393
8.1.1 实例: 处理点击按钮事件	282	9.5.4 弹出菜单	394
8.1.2 选择事件监听器	288	9.5.5 快捷键和加速器	395
8.1.3 实例: 改变观感	290	9.5.6 启用和禁用菜单项	397
8.1.4 实例: 捕获窗口事件	293	9.5.7 工具栏	402
8.2 AWT事件层次图	296	9.5.8 工具提示	404
8.3 AWT中的语义事件和低级事件	298	9.6 复杂的布局管理	407
8.4 低级事件类型	300	9.6.1 网格布局	409
8.4.1 焦点事件	301	9.6.2 箱式布局	413
8.4.2 键盘事件	302	9.6.3 网格组布局	417
8.4.3 销毁事件	308	9.6.4 gridx、gridy、gridwidth	

和gridheight参数 .....	419	10.4.2 在浏览器中显示信息 .....	497
9.6.5 增量字段 .....	419	10.4.3 书签applet .....	499
9.6.6 fill和anchor参数 .....	420	10.4.4 双重身份: 既是applet又 是应用程序 .....	502
9.6.7 填塞 .....	420	10.5 JAR文件 .....	507
9.6.8 指定gridx、gridy、gridwidth和 gridheight参数的另一个方法 .....	420	10.5.1 文件清单 .....	508
9.6.9 不使用布局管理器 .....	424	10.5.2 JAR缓存 .....	509
9.6.10 定制布局管理器 .....	425	10.5.3 自运行JAR文件 .....	510
9.6.11 遍历顺序 .....	429	10.5.4 资源 .....	510
9.7 对话框 .....	431	10.5.5 可选包 .....	514
9.7.1 选项对话框 .....	431	10.5.6 密封 .....	515
9.7.2 创建对话框 .....	442	第11章 异常和调试 .....	516
9.7.3 数据交换 .....	446	11.1 处理错误 .....	516
9.7.4 文件对话框 .....	452	11.1.1 异常的分类 .....	517
9.7.5 颜色选择器 .....	463	11.1.2 通告被方法抛出的异常 .....	518
第10章 applet .....	469	11.1.3 如何抛出异常 .....	520
10.1 applet基础 .....	469	11.1.4 创建异常类 .....	521
10.1.1 一个简单的applet .....	471	11.2 捕获异常 .....	522
10.1.2 运行applet查看器 .....	472	11.2.1 捕获多个异常 .....	524
10.1.3 在浏览器中查看applet .....	474	11.2.2 重新抛出异常 .....	525
10.1.4 把应用程序转化成applet .....	477	11.2.3 关于Java错误和异常处理 的综合实例 .....	528
10.1.5 一个applet的生存周期 .....	479	11.3 使用异常技巧 .....	532
10.1.6 安全基础 .....	480	11.4 调试技术 .....	535
10.1.7 在applet中弹出一个窗口 .....	481	11.4.1 调试技巧 .....	535
10.2 applet HTML标记和属性 .....	483	11.4.2 断言 .....	539
10.2.1 applet定位属性 .....	484	11.4.3 使用控制台窗口 .....	541
10.2.2 applet代码属性 .....	484	11.4.4 跟踪AWT事件 .....	542
10.2.3 applet无法正常显示时 的替代属性 .....	487	11.4.5 AWT机器人 .....	546
10.2.4 OBJECT标记 .....	487	11.4.6 性能统计 .....	550
10.2.5 Java插件标记 .....	488	11.4.7 覆盖测试 .....	553
10.2.6 把信息传递到applet中 .....	489	11.5 调试器的使用 .....	554
10.3 多媒体 .....	494	11.5.1 JDB调试器 .....	555
10.3.1 URL .....	495	11.5.2 Forte调试器 .....	559
10.3.2 获得多媒体文件 .....	495	第12章 流和文件 .....	562
10.4 applet环境 .....	497	12.1 流 .....	562
10.4.1 applet间通信 .....	497	12.2 完整的流集合 .....	564

12.2.1 对流过滤器进行分层 .....	566	12.5.2 对象序列化文件格式 .....	606
12.2.2 数据流 .....	570	12.5.3 保存对象引用问题 .....	609
12.2.3 随机访问文件流 .....	573	12.5.4 对象引用的输入格式 .....	615
12.3 ZIP文件流 .....	581	12.5.5 安全 .....	618
12.4 流的实际应用 .....	589	12.5.6 版本问题 .....	622
12.4.1 写分隔输出 .....	589	12.5.7 使用序列化来克隆 .....	624
12.4.2 字符串标记器和分隔文本 .....	590	12.6 文件管理 .....	626
12.4.3 读取分隔输入 .....	591	附录A Java关键字 .....	632
12.4.4 随机访问流 .....	595	附录B 关于光盘 .....	634
12.5 对象流 .....	602		
12.5.1 存储可变类型的对象 .....	602		

# 第1章 Java 简介

很长一段时间以来，随便翻开一本计算机杂志，你都会看到有关Java的内容。甚至那些主流报纸杂志，如《纽约时报》、《华盛顿邮报》以及《商业周刊》等也都充斥着大量关于Java的文章。CNN、CNBC这些家喻户晓的大众媒体现在都在谈论着Java的一切。

本书的读者定位是专业程序员，所以本书不会对那些铺天盖地的关于Java的天花乱坠的广告或宣传感兴趣，而是会详尽地介绍Java这种编程语言本身的知识（当然，这包括那些使得世界开始关注Java的一些特性）。

在Java语言发展的早期，它的能力和宣传有点距离。但当Java成熟以后，技术变得稳定可靠，同时人们对Java的期待变得更为合理。在本书写作时，Java主要用于开发中间件（middleware）——用于处理客户机和服务器资源（例如数据库）之间通信的技术。尽管这不十分引人注目，但它却是Java的真正用武之地，这得益于Java语言的可移植性、多线程机制以及网络性能。同时Java也广泛用于嵌入式系统开发，它正逐渐变成开发手持设备、互联网信息站、车载计算机等应用的标准。然而，早期尝试过的用Java开发常规PC程序的思路，现在看来不是很有前途，这是因为那些用Java开发的应用程序功能低下并且非常慢。在当前的Java版本中，这样的一些问题已经得到解决，但是用户仍然不要为用Java开发这些应用程序报太大希望。我们认为Java的种种优势在它应用于新型设备或新应用领域中才能得以体现，而不是用它来重写那些已存在的应用程序。

## 1.1 Java是一种编程工具

作为一种计算机语言，Java的宣传有点过火。不过它确实是一种很好的编程语言，对于真正的程序员来说，Java确实是一个好的选择。我们认为Java曾有可能成为一个伟大的编程语言，但似乎不可能了。当一个语言应用在某个领域时，它同现存代码的兼容性问题便随之而来。而且，即使存在不需废弃现存代码稍做改动的情形，也会如此。对于Java的设计者来说也不可能说“可能我们在X上犯了错误，Y是个更好的选择”。总而言之，尽管我们仍期待以后会有一些改进，但基本上Java语言的结构不会做任何改变了。

那么Java的改进在哪里呢？答案不会是对Java编程语言底层机制的改变，而是对Java库的重大变化。一直以来，Sun公司改变了从许多库函数的名字（变得更为一致）到图形工作机制（改变了事件处理模型并重写了非正式部分）的一切东西，并且增加了许多重要的功能，如打印功能（这不是Java 1.0的一部分）。这样做的结果是产生了一个有用的、强大的、远远优于早期版本的Java编程平台。

**注意** 微软发布了一个称作J++的产品，它同Java具有家族关系。同Java类似，J++也需要一个虚拟机来解释执行。该虚拟机在执行Java字节码时同Java虚拟机兼容，但在同外部代码交互时它同Java虚拟机有着明显的区别。J++基本编程语法同Java几乎完全一样，然而，微软为该语言增加了调用Windows API的功能。除了语法一样以外，J++和Java

在基本类库（字符串、工具类、网络、多线程、数学库等等）上也完全一样。然而，在图形库、用户界面以及远程对象访问上二者完全不同。现在微软不再支持J++，而是全力开发一种新型的称作C#的语言作为替代产品。C#在许多方面同Java类似，但是它使用一种不同的虚拟机。本书不涉及有关J++或C#的知识。

## 1.2 Java的优点

Java的一个显著优点是运行时环境提供了平台独立性：你可以在Windows、Solaris、Linux或其他操作系统上使用完全一样的代码。这点对于在各种不同平台上运行从因特网上下载的程序来说是非常有必要。

另一个优点是Java具有同C++类似的语法，这点使得C或C++程序员可以很容易的学习Java。当然，Visual Basic程序员可能要为语法头疼一下。

**注意** 如果你熟悉其他语言，而不是C++，那么你对本节用到的一些术语可能不很熟悉。请跳过本节！你会在第6章结束时熟悉所有的这些术语。

Java也是完全面向对象的，这点要强于C++。Java中任何一切，除了几个基本类型（如数字）以外，都是对象（面向对象编程已经取代了早期的结构化技术，这是因为它在处理复杂的项目具有很多优势。如果你不熟悉面向对象编程，请参阅从第3章到第6章的有关内容）。

然而，Java不仅仅简单的是C++的一种方言：使用Java能够比使用C++更容易开发没有bug的代码。

为什么？Java的设计者仔细研究了是什么使得C++代码具有如此多的bug，因此他们为Java增加了很多功能，来减少出现常见bug的可能性：

- Java设计者限制了手工内存分配和释放。在Java中，内存是自动回收的。你永远不必担心会出现内存崩溃现象。
- 他们设计了真正的数组并且限制了指针算法。你永远不必担心因为处理指针操作时出现的偏移错误而重写一块内存区域。
- 在Java中，条件语句中的相等测试同赋值之间的冲突被消除了。例如，你不可能能够编译 `if (ntries = 3) . . .` 这样的语句。（VB程序员可能从未见过这样的程序，但是请相信，在C/C++代码中这种程序正是错误之源！）
- Java中取消了多重继承。替代方案是采用一个新关键字“interface”——接口（这从Objective C中借鉴而来）。接口能够实现多重继承的大部分功能，并且它消除了使用多重继承带来的复杂性和麻烦。（本书第5章解释了继承概念。）

**注意** Java语言规范是公开的。你可以在下面站点找到它：<http://java.sun.com/docs/books/jls/html/index.html>。

## 1.3 Java的关键特点

Java的作者写了一本很有影响的白皮书，解释了设计者的设计目标和成就。该白皮书提出了



11个关键特点：

(1) 简单 (2) 可移植性 (3) 面向对象 (4) 解释型 (5) 分布式 (6) 高性能 (7) 健壮性 (8) 多线程 (9) 安全 (10) 动态 (11) 中立体系结构

我们将在最后一节接触这些要点。在本节，我们的工作：

- 通过引用白皮书中的那些设计者所说的话来介绍一下每个关键特点。
- 告诉读者我们对每个关键点的理解，这完全基于我们对Java当前版本的使用经验。

**注意** 在本书写作时，白皮书可以在这里找到：[http://java.sun.com/doc/language\\_environment](http://java.sun.com/doc/language_environment)。

### 1.3.1 简单

我们希望构件一个无需深奥专业训练就可以进行编程的系统，并且它要结合当前的标准惯例。所以，尽管我们发现C++是不适合的，我们仍然把Java设计成同C++尽可能的接近，易以便能够使得系统更易懂。Java剔除了C++中的一些很少使用的、难理解的以及易混淆的特性，它变得很简洁。

Java的语法实际上是C++语法的一个“纯净”版。根本没有必要使用头文件、指针算法（或指针语法）、结构体、联合、操作符重载、虚基类等等（看看本书中多处“C++注意”的内容就知道了）。然而，设计者并没有试图纠正C++中所有的拙劣特性，例如对switch语句就没有做任何改变。如果你熟悉C++，你就会很容易转到Java上来。

如果你曾使用过一种可视化编程环境（例如Visual Basic），你将不会认为Java很简单。你会发现其中有太多的奇怪语法（尽管熟悉它们并不麻烦）。更重要的是，在Java中你必须自己做大量的编程工作。Visual Basic的强大就在于它的可视化设计环境自动地为一个应用程序提供了很多基础结构，而在Java中的同等功能却需要手工编程实现（写一行一行的代码）。然而，现在有一些第三方开发环境也提供了拖放风格的程序开发工具。

简单的另一方面是小。Java的一个目标就是能够使得软件可以在很小的机器上独立运行。基础解释器和类支持的大小约为40KB。增加基本的标准库和线程支持（本质上是一个自包含的微内核）大约需要增加175KB。

这是了不起的成就。然而，图形用户界面（GUI）库是非常巨大的！

### 1.3.2 面向对象

简单的说，面向对象设计是把重点放在数据（等于对象）以及对象的接口上的一个编程技术。以木匠为例：一个“面向对象”的木匠会首先关心他要制造的椅子，然后才是制造这把椅子需要的工具；而一个“非面向对象”的木匠会首先考虑他的工具。Java的面向对象特性比C++的要好。

在过去的30年间，面向对象的程序设计充分证明了它的价值，如果一个现代编程语言不使用它，那简直是不可想像的。事实上，Java的面向对象特性同C++的有很多相似之处，二者的主要区别在于多重继承的处理（Java通过一个更好的方案实现）以及Java的元类模型。反射机制