

80

Mastering Java 2 J2SE 1.4

Java 2从入门到精通

(J2SE 1.4版)

〔美〕 John Zukowski 著

邱仲潘 等译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

致 谢

一本书完成之时最让人开心，这是一个漫长而艰辛的旅程。有过8次经历后，我原以为会轻松一些。

和过去一样，许多人把我的粗糙手稿变成了你手头漂亮的图书。感谢Sybex公司的生产编辑Kylie Johnston、采购与开发编辑Diane Lowery和编辑Carol Henry善始善终。特别感谢Customware公司的技术编辑Rob Castaneda认真查找例子中的技术问题。Kevin Ly与Dan Mummert出色地编译了本书选配光盘。合成员Maureen Forys和校对小组也做了不少工作。

感谢Terence Parr为本书作序。Ter也许忘记了，如果我不租Jaguar车，他只好凌晨三点送我到达机场。另外，Brian Christeson（原先在jGuru的另一名员工）可以作证，前排座椅只能向前调，无法向后调。由于整个星期都下雨，我没法不用这辆破车。

这里还要感谢一些与本书无关但值得一提的人。祝贺Melissa与Steve Pelton结束了自由的单身生活，把幸福绑在一起。感谢Matt Groening创作的《The Simpsons》给我带来的快乐，感谢我的表弟Rachel决定不搬回波士顿，祝Todd好运。Heinz，继续调制绿色与紫色的果汁吧，何必跟别人一样？

当然，忘不了感谢我的妻子Lisa和老牧羊犬Jaeger，愿它早日有个伴。

—John Zukowski

译 者 的 话

本书翻译过程中得到了刘文红、周阳生、邹能东、彭振庆、黄志坚、李耀平、江文清等同志的大力帮助，刘文琼、温连英、邱冬金、邱燕明等同志完成了本书的录入工作，刘云昌、刘昌和兄弟帮助进行了书稿与打印稿的校对，在此深表感谢。

邱仲潘

序

Java已经确定了作为当今网络编程语言的地位，已经证明是相当有效的一般编程语言。Java的优势在于安全、可移植、明晰的编程环境和支持网络中软件的无缝与无形发布。

20多年来，我遇到了各种开发难题，都在Java中得到妥善解决。例如，我是广泛使用的ANTLR翻译器生成器的设计人员，遇到C/C++的移植性问题（库的变化和整数长度的不同）。在各个作业中，我要一次次重来，这是由于缺乏丰富的库和基础结构所造成的。我花大量时间建立、调试与远程机器交流的代码，跟踪无内存回收功能的语言中内存泄漏问题。为了远程调试接口，我曾经用指示灯和滤波器调试程序！

Java的移植性使服务器方开发人员受益匪浅。我是jGuru.com网站应用程序的主要开发人员，曾经在开发、测试与部署的不同阶段在Linux、PC与Mac OS X之间来回交换软件。我已经习惯于Java的移植性，总以为在任何操作系统中都可以运行我的Web站点应用程序。如果你以为只有服务器方Java代码才要到处运行，我可以告诉你，我曾经顺利地所有三种平台中使用同一个基于Java的IDE，多么奇妙！

使用Java的编程人员比使用任何其他常用语言更具生产效率。单单不用编写跟踪与去配内存的代码就是个巨大的改进。Java具有简单而有效的内置线程模型，可以更方便地建立服务器、以网络为中心的服务和响应性图形用户界面。Java标准库的深度与广度是前所未有的。在C语言中，很难超越使用printf()的标准库。

设计Java时，人们考虑到代码要通过网络传递到信任机器和非信任机器，来自信任源和非信任源。因此，Java的基础结构支持签名与验证代码，检查执行权限，以及自己进行保护。例如，在C/C++中，可能会覆盖其他变量所在内存区，造成奇怪的行为。这种缓冲区溢流不仅很难跟踪，而且很容易造成安全问题。Java的所有代码都在“沙袋”中运行，使恶意代码无法执行优先方法，覆盖缓冲区或进行任何破坏系统完整性的操作。

Java允许在万维网上发送数据，以及发送解释这些数据的代码。1995年出现Java之前，Web站点只能向浏览器发送数据；现在，服务器可以无缝地、无形地向浏览器发送Java小程序，像任何其他客户端应用程序一样在浏览器中执行，同时自动执行安全限制。能够在客户计算机上动态自动安装最新软件可以大大减少发布与更新软件的事务成本。使用旧的技术时，大多数大公司要组织一群系统管理员到各个现场更新公司IT软件，毫无疑问，有些计算机总是过时的，升级的成本和工作量很大，因此不能经常进行。

Java的市场份额还在不断上升，图书、新闻组、邮件清单和Web站点对Java强力支持，如Sun公司的Java Developer Connection、IBM公司developerWorks和<shameless-plug>jGuru.com</shameless-plug>。更重要的是，可以取得所有库存的源代码。记住，代码就是文档。

下面介绍一下John Zukowski，John是对Java及其库最了解的人，也是可乐喝得最多的人，娶了个最漂亮的太太的人。1995年，jGuru（原来的MageLang Institute）在Java邮件清

单中寻找一个专家，我们发现John的文章最深刻最优秀。John非常活跃，可见睡得很少。John还是个喜剧演员，几年前从公司撤退的情形让我至今记忆犹新，John开的是租来的Jaguar车，他说这是他开过的唯一一辆车。

请一定要买这本书，因为John最知道如何介绍Java，他从Java推出的第一天起就在使用Java。几年来，他不断改善例子，让你知道需要了解什么，不需要了解什么。John为jGuru编写的教程是我们的学员和读者最欢迎的。本书选配光盘中提供了建立简单程序到建立Web服务器所需的一切。因此，别犹豫，赶快掏钱吧。

—Terence Parr

jGuru.com创始人 and 首席科学家

前 言

欢迎使用《Java 2从入门到精通（J2SE 1.4版）》，这是一本Java编程语言的综合性书籍。通过学习本书中的基础知识直到中级、高级课题，可帮助你成为高水平的Java编程人员。

导游图

本书适合你吗？尽管本书是按一定逻辑顺序编写的，但大多数读者不会拿起来按顺序从封面看到封底。根据你的特定背景，可以按下面介绍找到最适合你的个人需要的阅读方法。

对非C/C++/C#编程人员

如果你不熟悉Java和C/C++/C#风格的编程，则需要阅读全文。有些概念与其他语言相似，但为了掌握Java工作的方法，需要阅读全文。为了充分掌握提出的概念，应当亲自做每一个例子。每一节之间可能要停几个小时或几天，以便消化吸收前面的内容。

对C/C++/C#编程人员

由于C/C++/C#与Java的相似性，C/C++/C#编程人员可以快速翻过本书的部分章节。C编程人员可以跳过第4章、第6章的大部分，而C++编程人员还可以跳过第3章、第5章关于类的部分以及第7章（如果参与过异常处理）。尽管这些章节值得温习（如第7章的断言功能），但并不需要像对待书中其他部分那样仔细琢磨。

对Java编程人员

如果你自己用过Java，想找一本书来帮助提高，第一部分的大部分可能都是复习。扫过一遍看看有什么过去遗漏的东西，特别注意第7章、第8章，分别介绍异常处理和多线程。第5章内部类的介绍、第7章的断言功能和第8章线程局部变量部分也值得看一看。

对所有读者

一旦有所了解，可以用目录表寻找最让你感兴趣的部分。第10章和第11章一起介绍Java动画和图形编程。第12章至第15章是放在一起的，它们涉及建立和使用丰富的Java客户机，有些人也许觉得先看第13章再看第12章更自然。第16章介绍拖放功能，要利用其他Java客户机章节的知识。如果你对网络编程（第22章）有较大兴趣，首先要了解I/O（第19章），因为它是建立在输入和输出流（stream）之上的。打印（第20章）也要求一些I/O与图形知识。其他章节可以按任何顺序阅读。

每一章提供的例子帮助解释概念，复习程序提供对专题的更好理解。所有源程序代码都在本书选配光盘上，所以不必输入例子。

本书的特点和结构

本书的目的很简单：让你尽快有效地使用Java。本节内容丰富，目录和索引能帮你找到感兴趣的内容。下面简要介绍本书的内容及所在章节。

第一部分：Java基础

本书第一部分介绍Java的历史、语言和编程概念。第1章介绍什么是Java，Java的由来；第2章开始使用Java开发环境；第3章介绍面向对象的编程基础；第4章介绍Java语言的语法；第5章通过学习类、接口和软件包开始建立Java的概念；第6章介绍Java如何处理数组和流控制语句；第7章介绍Java的异常处理机制；最后，第8章介绍如何在Java中建立多线程程序。

第二部分：介绍Java库

第二部分挖掘Java库（用Java的话来说就是软件包）的内涵。第9章简单概述不同Java软件包及其零件；第10章介绍Java的基本动画功能；第11章介绍更高级的2D功能；第12章介绍窗口软件包，介绍如何在屏幕上定位对象；第13章介绍与用户交互作用的不同对象；第14章介绍通过事件处理完成交互操作；第15章概述用户可以交互的更高级对象；第16章介绍应用程序间传输信息的机制；第17章介绍Java中处理集合的数据结构支持类以及其他实用程序库；第18章介绍部署Java程序的技术；第19章介绍通过Java的流进行I/O编程；第20章介绍如何在Java程序中进行打印。

本书末尾介绍一些更专门的课题。第21章讨论JavaBeans API及其如何扩大软件开发预算。第22章介绍如何使程序变成Internet实用知识。

最后，第23章提供了其他Java API的索引。一本书能介绍的内容是有限的，因此本章提供了其他Java API的索引，并提供了Java 2 Enterprise Edition (J2EE) 中的API，而不是本书介绍的J2SE中的API。

本书选配光盘

本书选配光盘上包含下列内容：

- 本书所有例子中的源程序代码。
- Sun系统公司Software Development Kit (SDK) V1.4，适用于Windows、Solaris与Linux。这是最新版，而不是早期访问版。
- 可以使用的模拟工具集合，包括获奖的Integrated Development Environment (IDE) JBuilder 6 Personal Edition（企业版为30天试用期）；Optimizeit Suite的测试版可以监视线程性能；开放源代码工具Tomcat、Ant与JUnit和Beans Development Kit (BDK)。
- 本书选配光盘中还有个Jindent工具，可以将源代码重新格式化，便于打印。详见光盘的Readme文件。

规则

本章用了许多不同的规则来帮助尽快找到所需的信息。这里显示的提示、说明和警告穿插在文中，帮你尽快找到重要的强调信息。

提示：这是提示，包含帮助澄清文本和使你更好更快地利用Java的信息。

说明：这是说明，包含需要强调的信息。

警告：这是警告，包含标明潜在故障的信息。

此外，本书还利用了不同的字体格式。英文粗体表示用户输入的内容（如在文本字段中的）。英文等线字体用于程序代码。

程序代码本身也符合Java API所有的大写标准规则。例如，在类名中，每个单词大写；而在函数名中，每个单词第一个字母大写。代码格式符合标准编程规则：语句末尾或下一行开头放上左大括号，而右大括号自成一行，并用缩排表示代码的组织。

书中有许多类合成表，列出并描述各个类的成员。在这些表中，加下划线的成员名表示静态成员，斜体的成员名表示保护成员。关于静态成员与保护成员的含义，见第4章介绍。

Java技术支持

如果需要帮助，可以找Java技术支持，详见附录B列出的常见问题清单、论坛、邮件清单、新闻组、用户组，等等。

关于Sybex公司

Sybex提供了各种水平的技术书籍。例如，如果要了解Web服务，见《SOAP Programming with Java》（Sybex公司出版）。如果遇到Java以外的问题，其他Sybex书也能提供答案。关于最新书目，请通过下列地址联系：

Sybex Inc.

1151 Marina Village Parkway

Alameda, CA 94501

或访问其Web页<http://www.sybex.com>，其中有一个可搜索目录和本书的升级信息。

了解了《Java 2从入门到精通（J2SE 1.4版）》一书的概况后，现在开始学习吧。

目 录

第一部分 Java基础	1
第1章 Java简介	1
Java简史	1
Java体系结构	2
Microsoft的看法	6
下一章	6
第2章 Java 2 SDK简介	8
小应用程序和应用程序	8
使用Java 2 SDK	10
生成Java应用程序	14
生成Java小应用程序	21
下一章	25
第3章 使用Java对象	26
面向对象编程简介	26
设计类	28
下一章	33
第4章 数据类型、修饰符和表达式	34
说明语句	34
关键字	38
变量	39
数据类型	39
直接数	40
语句与表达式	43
运算符	44
修饰符	52
下一章	57
第5章 Java类、接口和包	58
数据类型的强制类型转换	58

对构造函数和成员引用使用this和super	60
对象内存模型	62
Java接口	64
使用Java包	70
建议规则	73
第6章 数组与流程控制	75
在Java中使用数组	75
使用流控制语句	83
下一章	93
第7章 异常处理与断言	94
Java异常处理结构	94
管理抛出异常的方法	97
自己抛出异常	98
生成定制异常类	99
链接异常	100
读取堆栈转储	101
使用finally从句	101
检查断言	103
下一章	105
第8章 线程与多线程	107
多线程基础	107
高级多线程	125
下一章	146
第二部分 Java库细节	149
第9章 标准Java包	149
实用库	150
多媒体库	152
下一章	155
第10章 动画与图像	156
图形基础	156
动画基础	175
动画加速提示	191

第11章 二维图形与图像	193
图形处理	193
文字处理	217
图像处理	222
图形设备挂钩	227
下一章	233
第12章 GUI布局管理器	235
Java GUI方法	235
用布局管理器设计GUI	236
布局管理器内幕：接口布局管理器	264
下一章	266
第13章 Swing GUI组件	268
AWT GUI类的上级类Component	268
JComponent：GUI类的Swing上级类	280
在GUI中加进Swing组件	285
使用菜单系统组件	326
下一章	330
第14章 事件处理	332
基于委托的事件处理	332
事件类型与听众	335
适配器	369
下一章	370
第15章 高级Swing功能	371
高级Swing事件处理	371
生成用户对话框	382
模型/视图/控制器结构	401
复杂Swing组件	404
客户机属性	440
下一章	441
第16章 数据传输	442
可传输对象	442
剪贴板传输	445
拖放法	456

下一章	459
第17章 Java集合	460
Java 2集合框架	460
使用Java的数学库	493
日期与日历	499
下一章	503
第18章 高级小应用程序编程	504
纯粹GUI驱动小应用程序	504
部署小应用程序	525
优化启动时间	533
下一章	536
第19章 流与输入/输出编程	540
Java的文件管理方法	540
Java I/O流的操作方法	550
对象持续与系列化	578
新I/O包	589
日志	595
下一章	597
第20章 打印	598
历史	598
Java 1.4中打印概述	601
使用StreamPrintService	612
打印屏幕	613
下一章	616
第21章 JavaBeans	617
JavaBean组件模型	617
Java的Beans	618
Bean内省和定置	624
Beans开发工具库 (BDK)	626
下一章	640
第22章 网络编程	641
Java的网络协议: TCP/IP	641

Internet 寻址	644
与远程系统通信	646
从套接信道读取	666
非阻止读	668
编写服务器系统	673
非阻止服务器	679
下一章	680
 第23章 Java扩展库	683
Java 2平台标准版	683
Java 2平台企业版	707
Java Communications (COMM)	721
下一章	724
 附录A Java语言语法	725
 附录B 联机资源	753
 词汇表	757

第一部分 Java基础

第1章 Java简介

在印度尼西亚，有个小岛，名叫爪哇，岛上的人们过着平静的生活，年复一年谋生度日。20世纪90年代初的一天，一个叫Sun系统公司的公司，建立了Oak编程语言，由于Oak是Sun公司内部使用的语言，因此这个名称并不引人注目，爪哇岛上的居民照样过着快乐的生活。

几年之后，Sun系统公司想把Oak作为产品推出，但Oak这个名称没能通过商标测试，Silk、Ruby和WRL（表示WebRunner Language）也行不通，最后Java一举胜出，顺利通过商标测试。显然，编程语言与岛名相同不会发生冲突，就像公司名可以和星座名相同一样。1995年5月23日，Java出现在Sun系统公司的年度SunWorld会议中，作为公司20项公布项目中的第12项。印度尼西亚人民并不知道他们的爪哇小岛会发生什么事。

Java平台不断演变，Microsoft公司看到它的重要性，开始把Java加进Internet Explorer浏览器中，而爪哇小岛上的火山也开始不得安宁了。1997年，Sun公司因为Java的一些不兼容性而起诉Microsoft公司，喀拉喀托火山也开始活跃起来（从此爪哇小岛每年要发生大约十次火山爆发，也许这只是一种巧合吧）。随着官司越打越大，岛上居民的生活也越来越难过，雅加达的经济危机，造成苏哈托总统下台，东帝汶提出要求政治自治。Sun Java也发生了同样激烈的血战，最终于2001年判决Microsoft公司受罚两千万美元。是巧合吗？问问瓦希德总统吧，看看他有何高见。

Java简史

好了，爪哇小岛与Java平台巧合的故事就讲到这里吧，下面要讲讲Java平台。1995年的Java只是个编程语言，如今则已成为一个平台。Java是平台中立的Internet友好开发语言，渐渐成为生成各种程序的手段。下面从头介绍Java的来历。

1991年，Sun系统公司开始了一个Green项目，开发用于机顶盒（智能有线电视盒）中的软件。由于家用电器中的硬件总是不断改变的，因此Java之父James Gosling发现C++不能胜任这个工作，结果，Oak语言应运而生，更不容易出错，不会使整个系统崩溃。Oak最初在Hammer技术设备*7（Star7）上运行。触摸屏上放上这个小小的数字角色Duke，这是个帮助代理，完成各种工作（后来变成Java吉祥物）。

说明：如果你没有见过Duke，它像个Star Trek通信员，戴着白帽子，牙齿“地包天”，鼻子又大又红，手臂像个Screen Bean（Screen Bean是PowerPoint展示中到处可见的卡通人物形象）。

Oak用于一次TV机顶盒操作系统的投标，这次投标失败了，Sun公司不愿意把Oak卖给Trip Hawkins（后来Gamemaker/3DO公司CEO），因此这个产品就搁在那里。好在后来出现了Mosaic（这是Netscape Navigator Web浏览器的前代产品），使Internet焕然一新。1994年，Oak改头换面，在Web上使用，WebRunner浏览器（后来的HotJava）显示了后来成为Java的新技术。

现在要推广Java了。在如今如雷贯耳的SunWorld会议上，Sun公司的John Gage和Netscape公司的Marc Andreessen向全世界公布了Java语言。Netscape宣布要在下一版浏览器中加进Java，到今年年底，IBM、Oracle、Borland、Adobe、Macromedia、Lotus、Spyglass与Intuit等公司都在自己的产品中加进Java，就连Microsoft公司也购买了Java许可证，宣布在其产品、操作系统和开发工具中加进Java。

Java最初的版本很小，可以放进一个软盘中，核心解释程序只有大约100KB，数学库有20KB，支持与C库集成的代码大约50KB，主要类库不到375KB，只要加上不同长度的平台特定图形库就可以了。总之，当时的Java只有1MB左右，可以在1MB的ROM和1MB内存的机器上运行（但*7有4MB内存）。

6年后，Java发生了巨大改变，不再限于在浏览器中运行，而是无所不在，包括Web服务器中、闪光的珠宝中、智能烤箱中和最初计划的机顶盒中。核心库也大大加大，最新版本（Java 2 Standard Edition, version 1.4）的运行环境就要占用30到40MB空间。显然，Java的功能更丰富了，因此成了一个平台，组合了编程语言、标准库和运行环境。本书正是介绍这个平台的。

Java体系结构

根据原先的Java白皮书（见<http://java.sun.com/people/jag/OriginalJavaWhitepaper.pdf>）称，Java的目标是成为“简单的、面向对象的、分布式、解释性、壮实的、安全的、体系结构中立的、可移植的、高性能的、多线程的动态语言”。介绍这些特性之后，就可以看看Java的过去与现状。

简单

在第一次C++会议上，Sun公司创造人之一Bill Joy（现为首席科学家）指出，“C++太复杂，我要的是C++ ++ --”。Java正是满足了这个要求。Java加进了自动内存回收之类的新任务（++），减去了C++中费解而不常用的方面，如运算符重载（--）。

Java简单性的另一个方面是Java中没有任何真正的新东西。如果看看Java特性集和计算历史，则可以发现，一切都是借用现成的东西。

- 类来自C++和Smalltalk，但Java只限于单实现继承。
- 接口来自Objective-C，向Java提供多接口继承。
- 包来自Modula，在Java中增加了层次式名字空间和逻辑开发单元。

- 并发性来自Mesa, Java内置多线程支持。
- 异常处理来自Modula-3, 在Java方法中增加了抛出异常的声明。
- 动态链接与自动内存回收来自Lisp, Java可以在需要时装入类, 不需要时将其释放。

随着新特性不断加进Java中, 这些特性也是从摸索与实践中的。例如, 1.4版本中还没有参数化类型支持, 也称为模板或一般化。增加参数化类型支持特性之后, 可以支持类型安全集合, 这个特性是人们渴望已久的。但是, Java不是把什么都加进来, 而是保证加进的东西加得合理。如果不出意外, Java 1.5将增加参数化类型支持。

面向对象

人们称Java为面向对象语言, 面向对象是什么含义呢? 实际上, 面向对象编程 (OOP, object-oriented programming) 是一种软件开发方式, 通过用问题空间中的元素与对象描述问题, 而不是用计算机中执行的一系列步骤来描述。这样, 好的设计就能得到可复用、可扩展和可维护的组件。这些组件相当灵活, 可以处理环境的改变, 因为这些对象的主要工作就是相互之间来回发送消息。

实际上, Core Java APIs是这些预建组件的集合, 称为类库。Java中不需要一切从头开始, 也可以从标准库开始。这些标准库随着时间推移不断改变, 每个新版本推出更多预建组件。

关于面向对象编程的详细介绍, 见第3章。

分布式

从1982年成立时起, Sun公司的思想一直是“网络就是计算机”, Java编程语言要求支持网络和像本地对象一样方便地访问分布式对象。这个目标一开始就已经提出, 但Java的功能是不断改变的。

最初, Java只能通过HTTP之类的标准TCP/IP协议访问分布式对象。而利用最新版的Java平台, 则可以通过CORBA (Common Object Request Broker Architecture) 与RMI (Remote Method Invocation) 等协议和最近增加的Web Services像访问同一执行空间的方法一样方便而直接地调用远程机器上的方法。对每个协议, 系统自动处理所有转换与传输。

解释性

Java程序是解释性的, 不是编译成自然执行码, 而是转换成平台中立的字节码。这些字节码可以传输到任何具有Java运行环境 (JRE, Java Runtime Environment) 的平台, 其中包括Java虚拟机 (JVM, Java Virtual Machine), 从而在执行时不需要重新编译或重新链接。也许你以为这个过程会使速度很慢, 但稍后“高性能”一节将会介绍, 情况并非如此。平台中立字节码实际上还包含其他信息, 可以在运行时根据编译时无法进行的决策进行执行优化。

健壮

健壮性反映程序的可靠性。Java的几个内置特性使程序的可靠性得到改进:

- Java是强类型语言。编译器和类装入器保证所有方法调用的正确性, 防止隐式类型失

配和版本不兼容性。

- **Java**没有指针，不能引用内存指针，搞乱内存或越过数组边界。**Java**用引用而不用指针，不能通过删除引用直接操纵内存空间（但**Java 1.4**可以对视频内存进行更加直接的控制）。
- **Java**进行自动内存回收，编程人员无法意外忘记释放内存，不需要判断何处需要释放内存。
- **Java**鼓励用接口而不用类。第5章将会介绍，接口定义一组行为，而类实现这些行为。我们传递的是接口而不是类，从而隐藏这些实现细节。实现细节改变时，只要新类实现旧接口，其余一切照样正常工作。

安全

Java有许多内置安全特性，1.4版中增加了更多标准安全特性。这些安全特性在几个不同方面起作用。许多保证**Java**健壮性的因素也保证程序执行的安全性——例如，没有内存指针，因此不会搞乱内存。其他安全特性内置在执行模型中：字节码验证器、类装入器和安全管理器，保证代码执行安全，防止非信任代码进行硬盘读取之类的信任操作。安全的最后方面与验证、授权和加密有关，保护隐私性和保证数据完整性。

字节码验证器提供了安全的第一道防线。实际上，字节码验证器不信任编译器。类装入执行环境之前，要验证字节码。如果有人对文件采用字节级编辑器，通过修改字节而访问一些不该访问的项目，则字节码验证器能够发现这种修改，防止装入类和执行程序。事实上，**JRE**中运行的程序不一定非要用**Java**编程语言编写。

另一个安全工具是类装入器。顾名思义，类装入器负责装人类。从安全方面，类装入器负责装人类并把装入的类保持在不同的信任区，与非信任代码分开。首先，类装入器将本地类与网络装人类放在不同名字空间，保证网络装人类无法取代本地类。第二，不同地方装入的类，也相互分开，保证非系统类无法伪装成系统类，也不能伪装第三方**Web**站点装入的类。

运行环境中的第三个安全组件是安全管理器对运行环境执行安全政策。需要一定权限来完成的所有任务都要通过安全管理器进行。如果任务无法从安全管理器中取得权限，则无法执行这个任务。这对**Internet**上通过非信任**Web**站点装入的代码特别重要。这种在浏览器中执行的代码称为小应用程序，在浏览器的“沙袋”中运行（沙袋是个安全环境，可以运行代码，但不能访问客户端环境）。

从**Java 1.1**到**Java 1.4**，安全功能不断加进**Core Java API**集合中。开发人员可以用数字签名签注**Java**类，允许用户信任其完整性，即使其代码越过沙袋范围，还可以生成消息摘要来保证完整性。**Java**还支持证书管理、访问控制表、安全套接层/传输层安全（**SSL/TLS**，**Secure Sockets Layer/Transport Layer Security**）和可导出加密库。

所有这些安全相关特性保证分布式网络程序的顺利构造与执行。

体系结构中立

体系结构中立性指**Java**的平台中立字节码。**Java**程序不是编译成平台特定的二进制码，