

计算机应用
系列教材

■ 魏衍君 范建华 主编

程序设计 Java

中国商业出版社

计算机应用系列教材

Java 程序设计

魏衍君 范建华 主 编

中国商业出版社

图书在版编目 (CIP) 数据

Java 程序设计/魏衍君, 范建华主编. —北京: 中国商业出版社, 2007. 8

(计算机应用系列教材)

ISBN 978 - 7 - 5044 - 5938 - 1

I. J... II. ①魏... ②范... III. Java 语言 - 程序设计 - 教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 088850 号

责任编辑: 刘树林

中国商业出版社出版发行
(100053 北京广安门内报国寺 1 号)
新华书店总店北京发行所经销
国防工业出版社印刷厂印刷

*
787×1092 毫米 16 开 17.25 印张 388 千字
2007 年 8 月第 1 版 2007 年 8 月第 1 次印刷
定价: 26.00 元

* * * *
(如有印装质量问题可更换)



根据当前我国电子计算机普及和发展的新形势，为适应国家技能型、紧缺型、实用型人才培养工程的需要，我们组织商贸系统部分高等职业技术学院和重点中等职业学校中具有丰富理论与实践经验，并多年从事计算机及应用专业教学的教授、高级讲师、讲师、工程师编写了这套计算机应用系列教材。

本系列教材面向 21 世纪，突出创新和发展，适应新世纪对人才培养高起点的要求，吸收计算机发展的最新成果。体现了科学性、先进性、理论性与普及性、应用性、操作性相结合的原则，做到了理论联系实际，内容翔实，结构严谨，体系合理，是一套较为实用的计算机应用系列教材。

本系列教材的读者对象，以高职为主，同时兼顾了中职教育和社会培训、等级考核的需要，也可以供职工大学、电视大学和高等专科学校使用。

本系列教材编写过程中，得到了有关院校领导和教师的大力支持，在此一并致谢。鉴于计算机飞速发展的实际，本系列教材难免有疏漏和不足之处，敬请广大读者赐教，使之日臻完善。

计算机应用系列教材编委会
2007 年 6 月



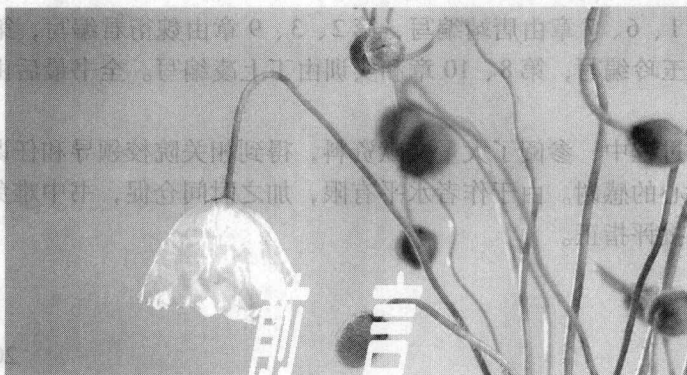
本书从 Java 语言的基本特点入手，逐步介绍了 Java 语言的基本概念和编程方法，并深入介绍了 Java 的高级特性，特别注重实用性，书中包括大量精心设计并调试通过的编程实例，方便初学者使用。本书共分 10 章，内容包括 Java 语言概述、Java 语言基础、面向对象程序设计基础、异常处理、图形用户界面设计、Applet 与 Japplet、多线程、输入输出流、数据库编程、Java 网络编程基础等。

本书语言通俗易懂，内容安排合理，深入浅出。书中每章附有习题，最后附有 17 个涵盖全书知识点的可操作性强的实训。本书可作为高职、高专学校的教材，也可以作为成人高校和在职人员培训班的教材以及自学者的参考书。

本书由清华大学出版社出版，定价 29.00 元。本书可作为高职、高专学校的教材，也可以作为成人高校和在职人员培训班的教材以及自学者的参考书。

清华大学出版社

2005 年 1 月



随着社会经济的高速发展，我国的高等教育已步入从精英教育走向大众化教育的发展阶段。因此我国近年来十分重视高等职业教育，把高等职业教育作为高等教育的重要组成部分，高等职业教育进入了蓬勃发展时期，驶入了高速发展的快车道。为此，我们根据高等职业教育的特点及社会对教材的普遍需求，组织有丰富教学经验的老师，编写了这本教材。

Java 语言从 1995 年正式发布以来，在短短的十多年间就已经风靡全球，扩展到各个应用领域。随着网络技术的发展，Java 语言成为最先进最流行的程序设计语言。Java 语言一改过去程序设计语言过多依赖硬件平台、可移植性差的缺点，提出在硬件平台基础之上构建 Java 虚拟机，并在此之上执行通用的字节码的思想，实现了编程人员梦寐以求的“一次编写，到处执行”的愿望。Java 语言符合程序设计的主流，继承了面向对象程序设计语言的诸多特色，包括封装、继承和多态，具有安全性和健壮性。此外，它是多线程的、动态的语言，具有动画、声音等功能，能实时处理信息。Java 与 Internet 结合紧密，强调网络特性，它还能用于瘦客户机体系架构，实现“零管理”。有人预言，在不久的将来，全球 90% 以上的程序代码将是采用 Java 语言开发的。

本书可作为高职高专学校的教材，也可以作为成人高校和在职人员培训班的教材以及自学者的参考书。书中内容以“实训应用”为主题，理论以“够用”为尺度，选材覆盖了 Java 程序设计的主要内容，本书注重培养读者的实际操作能力，侧重于应用性，力求内容与应用实例相结合。

本书结合高职高专院校学生的学习特点，突出实训，概念叙述简洁，深入浅出，便于读者掌握和理解。全书共分十章。第 1 章 Java 语言概述；第 2 章 Java 语言基础；第 3 章 面向对象程序设计基础；第 4 章 异常处理；第 5 章 图形用户界面设计；第 6 章 Applet 与 Japplet；第 7 章 多线程；第 8 章 输入输出流；第 9 章 数据库编程；第 10 章 Java 网络编程基础。每章均配有适量习题，具有很强的针对性和可操作性。实训部分精选典型实例，旨在帮助学生巩固所学知识。

本书由商丘职业技术学院魏衍君副教授、范建华副教授任主编，任玉玲、唐靖、李

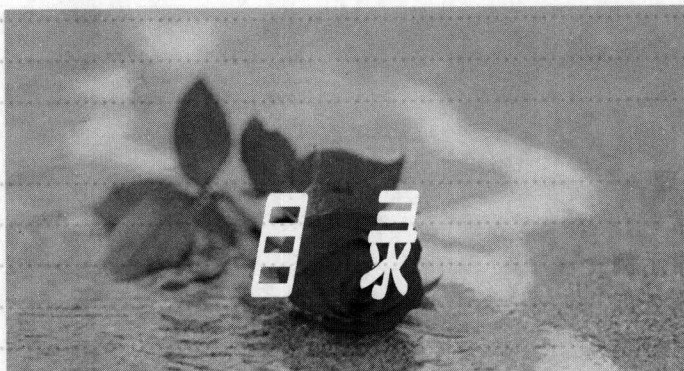


杰任副主编。第1、6、7章由唐靖编写，第2、3、9章由魏衍君编写，第4章由李杰编写，第5章由任玉玲编写，第8、10章和实训由丁上凌编写。全书最后由魏衍君、范建华统稿。

本书在编写过程中，参阅了大量文献资料，得到相关院校领导和任课教师的大力支持，在此表示衷心的感谢。由于作者水平有限，加之时间仓促，书中难免有疏漏和不妥之处，恳请读者批评指正。

编 者

2007年3月



第 1 章 Java 语言概述	(1)
§ 1.1 Java 语言概述	(1)
§ 1.2 Java 程序开发	(4)
§ 1.3 Java 程序的编辑、编译与运行	(7)
§ 1.4 UltraEdit 编辑器的使用	(16)
小结	(18)
习题一	(18)
第 2 章 Java 语言基础	(20)
§ 2.1 Java 语言的基本组成	(21)
§ 2.2 变量和直接量	(21)
§ 2.3 数据类型	(22)
§ 2.4 运算符与表达式	(26)
§ 2.5 Java 语句及其控制结构	(28)
§ 2.6 数组	(45)
§ 2.7 字符串	(49)
小结	(53)
习题二	(53)
第 3 章 面向对象程序设计基础	(55)
§ 3.1 类的基本概念	(55)
§ 3.2 对象	(63)
§ 3.3 类的封装	(66)
§ 3.4 多态机制	(73)
§ 3.5 类的继承	(75)
§ 3.6 接口	(80)



§ 3.7 包	(82)
小结	(85)
习题三	(85)
第 4 章 异常处理	(90)
§ 4.1 异常	(90)
§ 4.2 异常示例	(91)
§ 4.3 异常处理	(92)
§ 4.4 异常分类	(95)
§ 4.5 抛出异常	(95)
§ 4.6 自定义异常	(96)
小结	(97)
习题四	(98)
第 5 章 图形用户界面设计	(99)
§ 5.1 GUI 简介	(99)
§ 5.2 容器组件	(100)
§ 5.3 常用组件及使用	(106)
§ 5.4 组件布局管理	(120)
§ 5.5 事件处理	(127)
§ 5.6 图形处理	(133)
§ 5.7 Java 中的鼠标事件与键盘事件	(140)
§ 5.8 Java 与多媒体	(145)
§ 5.9 Java Swing 基础	(149)
小结	(152)
习题五	(152)
第 6 章 Applet 与 JApplet	(154)
§ 6.1 java. applet 包	(154)
§ 6.2 javax. swing. JApplet 类	(161)
§ 6.3 RootPaneContainer 接口	(164)
§ 6.4 JRootPane 类	(164)
§ 6.5 JLayeredPane 类	(172)
小结	(176)
习题六	(176)



第 7 章 多线程	(177)
§ 7.1 线程概述	(177)
§ 7.2 多线程设计	(181)
§ 7.3 线程同步与死锁	(189)
小结	(190)
习题七	(190)
第 8 章 输入输出流	(192)
§ 8.1 FileInputStream 类	(192)
§ 8.2 FileOutputStream 类	(195)
§ 8.3 FileReader 类和 FileWriter 类	(196)
§ 8.4 数据流	(198)
§ 8.5 Process 类中的流	(202)
小结	(207)
习题八	(207)
第 9 章 数据库编程	(208)
§ 9.1 数据库概述	(208)
§ 9.2 Java 数据库连接 (JDBC)	(210)
§ 9.3 JDBC - ODBC 桥接器	(217)
§ 9.4 数据库编程综合实例	(219)
小结	(228)
习题九	(228)
第 10 章 Java 网络编程基础	(229)
§ 10.1 使用 URL	(229)
§ 10.2 套接字	(231)
§ 10.3 InetAddress 类	(238)
§ 10.4 应用举例	(240)
小结	(251)
习题十	(251)
实 训	(253)
参考文献	(265)



知识要点:

1. Java 语言的概念和特点;
2. Java 程序的运行平台;
3. Java 程序的编辑、编译及运行;
4. Java 小应用程序。

§ 1.1 Java 语言概述

Java 语言是当今流行的网络编程语言，它的面向对象、跨平台、分布应用等特点给编程人员带来了一种崭新的概念，使 WWW 从最初的单纯提供静态信息发展到现在的能够提供各种各样的动态服务。Java 不仅能够通过编写小应用程序实现嵌入网页的声音和动画功能，而且还能够应用于独立的大中型应用程序中，其强大的网络功能可以把整个 Internet 作为一个统一的运行平台。

Java 的诞生从根本上解决了 Internet 的异构、代码交换以及网络程序的安全性等诸多问题。Java 语言具有平台无关性，程序经过编译后，生成字节代码，并运行在 Java 虚拟机 (Java Virtual Machine, JVM) 上。一个操作系统平台只要提供 Java 虚拟机，就可以在上面直接运行 Java 程序了。

Java 语言不仅是一个程序设计语言，也可以看作是一个网络操作系统，这是因为：

- (1) Java 语言不依赖于任何现有操作系统，可以运行在不同类型的机器上；
- (2) Java 语言有自己的内存管理机制；
- (3) Java 语言支持多线程程序，可对处理器资源进行分配和管理；



(4) Java 语言可对设备（如屏幕、键盘、鼠标等）进行管理；

(5) Java 语言支持网络文件管理，可以打开远程机器上的文件，因此 Java 语言文件的概念已经扩充到整个 Internet 网络。

1.1.1 Java 的起源、现状及前景

1991 年，SUN MicroSystem 公司的 Jame Gosling, Bill Joe 等人，为在电视机、烤箱等家用消费类电子产品上进行交互式操作面开发了一个名为 Oak（一种橡树的名字）的软件，后改名为 Java。Internet 的迅速发展促进了 Java 语言的发展，使得它逐渐成为 Internet 上受欢迎的开发编程语言。Java 的诞生对整个计算机产业产生了深远的影响，Java 为 Internet 和 WWW 开辟了一个崭新的时代。

Java 是新一代面向对象的程序设计语言，特别适合于 Internet 应用程序的开发，它的平台无关性直接威胁到 Windows 和 Intel 的垄断地位。

Java 的出现迅速引起了 IT 业和工业界的高度重视。Java 提供的强大的图形、图像、动画、音频、视频、多线程和网络交互能力，使它在设计交互式程序、多媒体网页和网络应用方面大显身手。目前，Java 已经成为最卓越的程序设计语言之一。

Java 不仅是一种程序设计语言，更是现代化软件编程技术的基础；Java 还是未来新型 OS 的核心；未来将会出现 Java 芯片；Java 将构成各种应用软件的开发平台与实现环境，成为人们必不可少的开发工具。Java 语言有着广泛的应用前景，例如：

- (1) 所有面向对象的应用开发，包括面向对象的事件描述、处理和综合等；
- (2) 计算过程的可视化、可操作化的软件的开发；
- (3) 动态画面的设计，包括图形图像的调用；
- (4) 交互操作的设计（选择交互、定向交互和控制流程等）；
- (5) Internet 的系统管理功能模块，包括 Web 页面的动态设计、管理和交互操作设计等；
- (6) Internet（企业内部网）上软件开发（直接面向企业内部用户的软件）；
- (7) 与各类数据库连接查询的 SQL 语句实现；
- (8) 其他应用类型的程序。

1.1.2 Java 语言特点

Java 语言适合用来开发网络上的应用程序。作为较晚问世的程序设计语言，Java 集中体现并充分利用了当代软件技术的新成果，如面向对象、多线程等。Java 语言的特点如下。

1. 简单易学

Java 语言是一种面向对象的编程语言，它通过提供最基本的方法来完成指定的任务。只需理解一些基本的概念就可以用它编写出适合于各种情况的应用程序。Java 提供了一个功能强大语言的所有功能，在 Java 语言中不存在模棱两可的概念。Java 语言与 C++ 语言的风格相似，出于安全性考虑，Java 语言去掉了 C++ 语言中不容易理解和掌



握的内容,如最典型的指针操作等,这使得 Java 语言的语法更精炼,并降低了学习的难度。

2. 面向对象

面向对象的技术较好地解决了当今软件开发过程中出现的各种面向过程语言不能处理的问题,包括开发规模的扩大、升级加快、维护难度加大等。面向对象技术是按人们的思维方式建立问题空间模型,利用类和对象的机制将数据及操作封装在一起,通过统一的接口与外界交互。在面向对象技术中,对象和消息表示事物之间的相互联系,类和继承是适应人们一般思维方式的描述方式,方法表示作用在该类对象上的各种操作。

3. 分布式

Java 是面向网络的语言。Java 语言的类库提供了用 HTTP 和 FTP 传送和接受信息的方法,用户可以通过 URL 地址在网络上很方便地访问其他对象。Java 语言 WWW 客户机/服务器模式,这使程序员使用网络上的文件就如同使用本地文件一样容易。Java Applet 小程序是动态、安全、跨平台的网络应用程序,它可以嵌入 HTML 语言,并通过主页发布到 Internet。用户访问服务器的小程序时,通过网络传输,可下载到客户端运行,即部分计算可以在客户端进行,这样就提高了系统运行效率。

4. 解释性

Java 是解释型的语言,它编译后并不生成特定的 CPU 机器代码,而是生成一种被称为字节码的目标代码。Java 解释器可以直接运行字节码指令,对其进行解释执行。Java 虚拟机使 Java 字节码指令可以在某一特定软硬件平台环境中直接运行,而工程人员不必考虑运行环境。

5. 健壮性

Java 在编译和运行程序时,都要对可能出现的问题进行检查,以消除错误的产生。Java 自己操纵内存从而减少了内存出错的可能性,它提供自动垃圾收集来进行内存管理,防止程序员在管理内存时容易产生的错误。通过集成面向对象的异常处理机制,在编译时,Java 将提示可能出现但未被处理的例外,帮助程序员正确地进行选择以防止系统的崩溃。另外,类型检查可以帮助程序员检查出许多开发早期出现的错误。Java 在编译时还可捕获类型声明中的许多常见错误,防止动态运行时不匹配问题的出现。

6. 安全性

现今的 Java 语言主要用于网络应用程序的开发,因此对安全性有很高的要求。如果没有安全保证,用户运行从网络下载的 Java 语言应用程序是十分危险的。Java 通过自己的安全机制防止了病毒程序的产生和下载程序对本地系统的威胁破坏。当 Java 字节码进入解释器时,首先必须经过字节校验的检查,包括代码校验,检查代码段的格式,检测指针操作,判断对象操作是否溢出及是否试图改变一个对象的类型等。然后,Java 解释器将决定程序中类的内在布局。随后,类装载器负责把来自网络的类装载到单独的内存区域,避免应用程序之间相互干扰破坏。最后,客户端用户还可以限制从网络上装载的类只能访问某些文件系统。



7. 平台无关性

Java 是平台无关的语言,也就是说用 Java 写的应用程序不用修改就可在不同的软硬件平台上运行,这主要是通过 Java 虚拟机来实现的。Java 源程序被编译成一种高层次的与机器无关的 Byte-code 格式语言,这种语言被设计在虚拟机上运行,由机器相关的运行调试器实现执行。Java 解释器生成与体系结构无关的字节码指令,只要安装了 Java 运行系统,Java 程序就可在任意的处理器上运行。

8. 可移植性

Java 的平台无关性使 Java 程序易于被移植到网络上的不同机器。Java 编译器是由 Java 语言编写的,Java 运行时的系统是由标准 C 语言编写的,这使得 Java 系统本身也具有可移植性。

9. 高性能

Java 是一个解释型语言。众所周知,解释型语言的执行速度要比编译型语言慢,实际上,Java 程序的运行速度仅是 C 程序的 1/20。为了解决这个问题,Java 提供了一种解释运行 JIT,它可以在运行时把 Java 的字节码翻译成特定 CPU 的机器码。这样使得转化成的机器码的运行速度接近于 C 语言或 C++ 语言。

10. 多线程

线程是操作系统的概念,它又被称作轻量进程,是比传统进程更小的、可并发执行的单位。Java 提供了多线程支持,即在一个程序里可同时执行多个小任务。多线程机制使应用程序能够并行执行,而且同步机制保证对共享数据的正确操作。通过使用多线程,程序设计者可以分别用不同的线程完成特定的行为。而不需要采用全局的事件循环机制,这样就很容易地实现了网络上的实时交互行为。所以多线程带来了更好的交互性能和实时控制性能。

11. 动态性

Java 利用了面向对象技术的优点。新的或升级的库函数不需要更改源程序就能正确运行。Java 的设计使它适合一个不断发展的环境。在 Java 类中可以自由地加入新的方法和实例变量而不会影响用户程序的执行并且 Java 通过接口来支持多重继承,使之比严格的类继承具有更灵活的方式和扩展性。

§1.2 Java 程序开发

用 Java 可以设计两种类型的程序:Java 小程序 (Java Applet) 和 Java 应用程序 (Java Application)。Java 应用程序是完整的程序,需要独立的编译器来编译运行;而 Java 小程序是嵌在 HTML 编写的 Web 页面中的非独立程序 Applet,由 Web 浏览器包含的 Java 编译器来编译执行。

小程序更适合作为 Internet 上的应用程序。非网络系统和内存较小的系统更适合用



Java 应用程序。小程序和应用程序之间的技术差别在于运行环境。Java 应用程序和一般独立执行的应用程序并没有什么不同，用户可以直接执行和使用 Java 应用程序，Java 应用程序可以运行在最简单的环境中，它的惟一外部输入法就是命令行参数。Java 小程序不能够直接被执行和使用；必须通过 Java 的网页浏览器（如 IE\Netscape），才能看到其执行结果。

Java 小程序的大小和复杂性都没有限制，Java 小程序有些方面比 Java 应用程序更强大，但是由于 Internet 通信速度的限制，大多数小程序规模较小。

1.2.1 Java 程序开发环境

使用 Java 语言开发 Applet 小程序，需要在计算机上安装支持 Java Applet 技术的浏览器 IE 或 Netscape Navigator，以及 Java 语言开发包 JDK（Java Development Kit）。

目前，大多数个人计算机中使用的浏览器都支持 Applet。但有些用户为安全起见，将浏览器中的 Internet 选项设置为不支持 Java，这样 Applet 就不能正常运行。实际上 SUN 公司在 Applet 安全性方面做了许多限制，如 Applet 不允许读取和修改存储在用户系统上的文件，只能在 Web 浏览器为 Applet 特别分配的区域运行，因而 Applet 在安全性方面对本地系统造成威胁几率是很小的。为使浏览器能够运行 Applet，应该选中此选项。

Java 语言开发包是一级 API，也可以说是一些 Java 类的库。目前比较稳定的是 JavaTM 平台，它共发布了 J2EE（Enterprise Edition 企业版）、J2SE（Standard Edition 标准版）和 J2ME（Micro Edition 微型版）3 个不同产品。目前正式发布的最新版本是 1.5.0，可以直接从 Java 官方网站 java.sun.com 下载获得。这 3 个不同产品分别用于不同的应用开发规模。J2EE 在电子商务平台的架构上得到了青睐；J2ME 用于机顶盒设备和掌上电子产品软件的开发；J2SE 也就是 Java 早期产品中的 JDK，用于开发常规的 Java 应用程序和 Java 小程序。无论是哪个产品，J2SE 都是一切 Java 应用程序的基础，可以说，所有的 Java 应用程序都是构建在这个产品上的。

J2SETM 平台有 JavaTM Environment（JRE）和 Java 2 SDK 两个主要产品，它们分别提供给不同的用户即消费者和开发者使用。虽然目前有些操作系统内嵌了 Java 虚拟机支持 Java 应用程序的运行，但 Sun 为了保证 Java 应用程序可以在所有平台运行，也免费提供了 JavaTM 2 Runtime Environment（JRE）以便重新分发运行环境。JRE 包含了 Java 虚拟机、运行时类库和启动以 Java 程序语言所撰写的程序所需的 Java 应用程序启动器。Java 2 SDK 软件 JRE 中所有内容，也可以作为运行环境，还包括开发用的编译器和调试器等工具。普通 Java 产品的终端用户不需要开发和调试用的工具，只需要运行已开发好的 Applet 和 Java 应用程序。JRE 相对于 Java 2 SDK 来说要小，因而容易下载和随同 Java 软件分发，适合于消费者使用。

J2sdk1.5.0 可以从 Sun 官方网站 http://java.sun.com/javase/downloads/index_jdk5.jsp 下载。下载的 `jdk-1_5_0_11-windows-i586-p.exe` 是个自解压安装文件，双击鼠标就可以安装。



下载的 jdk-1_5_0-doc.zip 是帮助文件,安装开发包帮助文档时,可将 jdk-1_5_0-doc.zip 解压到 jdk-1_5_0-doc 目录下,系统会自动创建 jdk-1_5_0-doc 目录,并将所有帮助文件安装在该目录下。通过双击 jdk-1_5_0-doc\docs\index.html 文件,就可使用 JDK 的 API 帮助文档了。

1.2.2 Java 程序运行平台

平台是程序运行的硬件或软件环境。大多数平台可以被描述为操作系统和硬件的组合。Java 平台与大多数其他平台的不同之处在于,它是一种运行在其他硬件平台上的纯软件平台。

Java 平台有两个组件:

- (Java 虚拟机 (Java Virtual Machine, JVM);
- Java 应用程序编程接口 (Java Application Programming Interface, Java API)。

1. Java 虚拟机

Java 虚拟机类似一个小巧而高效的 CPU。Java 虚拟机底层的运行系统把字节码转化成实际的硬件调用,但未必一定要运行在操作系统上,它的下面可以直接是各种 CPU 芯片。

首先,Java 编译器在获取 Java 应用程序的源代码后,把它编译成符合 Java 虚拟机规范的字节码的 Class 文件。Class 文件是 JVM 中可执行文件的格式。Java 编译器针对 Java 虚拟机产生 Class 文件,Java 虚拟机规范为不同的硬件平台提供了不同的编译代码规范,该规范使 Java 软件独立于平台。然后,Java 解释器负责将 Java 字节码文件解释运行。

在 Java 的运行环境中,每个 Java 解释器,不管它是 Java 开发工具,还是可运行 Applet 的 Web 浏览器,都可运行 Java 字节码。字节码的运行要经过下面 3 个步骤。

- (1) 加载代码:由 Class Loader 完成。
- (2) 校验代码:由 Bytecode Verifier 完成。
- (3) 执行代码:由 Runtime Interpreter 完成。

部分校验过的字节码被编译成原始机器码并直接运行于硬件平台,这就使 Java 软件代码能够以与 C 或 C++ 接近的速度运行,只是在加载时,因需编译成原始机器码而略有延迟。

构成 Java 软件程序的字节码在运行时被加载、校验并在解释器中运行。当运行时,字节码可被下载,然后由嵌在浏览器中的 JVM 进行解释。解释器具备两种功能,一是执行字节码,二是对底层硬件平台做适当调用。Java 虚拟机提供了编译代码规范,它要求代码的格式由字节码构成,由 JVM 字节码编写的程序必须保持适当的类型约束。对这种类型约束的检查,大部分是在编译时完成的。

Java 虚拟机也提供了平台规范,它能够解读独立于平台的已编译好的字节码文件,每个由 Sun 批准的 Java 虚拟机必须能够运行任何含有 Class 文件的程序,当然这些类文件应



符合 Java 虚拟机规范中所指定的类文件格式，如每个文件都包含最多一个 Public 类。

2. Java 应用程序编程接口

Java 应用程序编程接口是预先建立的软件组件的集合，它们能提供许多有用的功能，如图形用户界面（Graphical User Interface，GUI）部件。Java API 被分组为相关类和接口的库，这些库称为包。包形成了树形结构的类层次，其中包括核心包 Java、扩展包 JavaX、ORJ 包。

下面简单介绍一些重要的包和它们的类。

java.lang: 包含一些形成语言核心的类，即提供用 Java 语言开发编程所用的基本类，在这个包中封装了一些很有用的类，如 Object、Math、String、System、Integer 等。

java.lang.ref: 提供对垃圾回收功能支持的类，Phantom-Reference 封装一组引用对象，当垃圾回收器确定它们可以回收时，就把它们加入到引用队列中。

java.lang.reflect: 在程序加载运行时，提供对动态检测类的描述功能。类的描述包括成员变量、构造方法和成员方法等。

java.awt: 提供了构成抽象窗口工具包（AWT）的类，这个包被用来构建和管理应用程序的图形用户界面。

java.applet: 提供了可执 Applet 特殊行为的类。

java.net: 提供了执行与网络相关的操作的类及处理接口和 URL 的类。

java.io: 提供处理输入、输出文件的类。

java.util: 包含为任务设置的实用程序类和集合框架类，如随机数发生、定义系统特性和使用与日期日历相关的函数，集合 Collection、Map、Set、List、Arrays 等类。

3. Java 的运行结构

对于大多数语言来说，程序要么需要编译，要么需要解释，才能在计算机上运行。Java 编程语言的特殊之处在于程序既被编译又被解释。首先，使用编译器将程序翻译为一种称为 Java 字节码的中间语言，这是与平台无关的代码。解释器在计算机上分析并运行每条 Java 字节编码指令。编译只发生一次，而解释在每次执行程序时都发生。每一种操作系统的 Java 解释器是不同的，但 Java 虚拟机的实现是相同的。

这样，Java 虚拟机就可以把 Java 字节码程序与具体的软硬件平台分隔开来，只要在不同的计算机上安装针对其特定具体平台特点的 Java 虚拟机，就可以把这种不同软硬件平台的具体差别隐藏起来，使得 Java 字节码程序在不同的计算机上能够面对相同的 Java 虚拟机，而不必考虑具体的平台差别，从而实现了程序的真正二进制代码级的跨平台可移植性。Java API 和 Java 虚拟机将程序与硬件隔离开。

§1.3 Java 程序的编辑、编译与运行

1.3.1 J2SDK 的环境设置及其使用

在编写 Java 程序之前，要首先安装 J2SDK，其次还要进行相应的环境变量的设置，