21世纪高等学校计算机科学与技术规划教材

JAVA 程序设计教程

Java Chengxu Sheji Jiaocheng

主编 潘 浩

北京邮电大学出版社
www.buptpress.com

21世纪高等学校计算机科学与技术规划教材

Java程序设计教程

主　编　潘　浩
副主编　张国英　曹正凤

北京邮电大学出版社
www.buptpress.com

内容简介

本书基于 Java SE 6 开发平台系统地讲述了 Java 语言编程知识和 Java 程序的设计与开发。本书共有 13 章，第 1 章讲述了 Java 语言的产生与特点，介绍了面向对象编程知识；第 2 章讲述了 Java 开发环境的安装与配置，以及最简单的 Java 程序的编辑、编译和运行的全过程；第 3 章讲述了 Java 程序编程的基本语法，包括数据类型、变量、运算符和表达式；第 4 章讲述了 Java 的几种流程控制语句，包括 if 语句、while 语句、do… while 语句、for 语句以及循环跳离语句——break 语句和 continue 语句；第 5、6 章讲述了 Java 面向对象编程的核心知识，包括 Java 类和对象的定义、接口的定义、包的定义和加载；第 7 章讲述了 Java 类库的使用以及“java.lang”包中的常用类；第 8 章讲述了数组的定义和使用；第 9 章讲述了 Java 的图形用户接口的编程知识；第 10 章讲述了输入/输出流和文件的属性操作和存取操作；第 11 章讲述了多线程编程的知识；第 12 章讲述了通过 JDBC 访问数据库的知识；第 13 章讲述了 Servlet 和 JSP 编程的基本原理。

本书注重 Java 编程的基础和基本原理、讲究实用性、深入浅出、语言通俗易懂，可以作为高等院校计算机及相关专业的教材，也适合程序设计人员自学使用。

图书在版编目(CIP)数据

Java 程序设计教程/潘浩主编. 北京：北京邮电大学出版社，2008
ISBN 978－7－5635－1645－2

Ⅰ. J… Ⅱ. 潘… Ⅲ. JAVA 语言—程序设计—教材 Ⅳ. TP312

中国版本图书馆 CIP 数据核字(2008)第 114200 号

书　　名 Java 程序设计教程
主　　编 潘　浩
责任编辑 沙一飞
出版发行 北京邮电大学出版社
社　　址 北京市海淀区西土城路 10 号(100876)
电话传真 010－62282185(发行部)　010－62283578(传真)
电子信箱 ctrd@buptpress.com
经　　销 各地新华书店
印　　刷 北京忠信诚胶印厂
开　　本 787mm×1 092mm　1/16
印　　张 21.75
字　　数 511 千字
版　　次 2008 年 10 月第 1 版　2008 年 10 月第 1 次印刷

ISBN 978－7－5635－1645－2　　定价：32.00 元

前　言

Java语言是当前最成熟的网络编程语言之一，基于Java语言的各种技术已经扩展到信息领域的每个角落。Java已不仅仅被称为一门编程语言，也被称为Java开发平台。共有Java SE、Java EE和Java ME 3种Java开发平台，分别用于桌面应用软件的编程、分布式网络应用程序的开发、嵌入式系统领域的开发。其中，Java SE是Java 2的标准版，是其他两个开发平台的基础。

本书主要讲述了Java SE平台的基础知识。Java SE的学习主要包括两部分，一部分是学习Java编程的基本知识，包括Java编程的基本语法，如Java的基本数据类型、Java运算符、Java流程控制语句等，和Java面向对象编程的相关知识，如Java类和对象的定义，接口和包的定义与使用；另一部分是对Java开发类库的学习和掌握。Java类库是已经实现的标准类的集合，使用Java类库能够提高代码的可重用性、提高编程的效率，并能够降低编程的错误发生率。能否正确和熟练地使用Java类库，是衡量Java程序员编程能力高低的重要标准。

本书以实践教学为指导思想，采用案例教学法，强调理论与实例相结合的学习方式。对于Java每个知识点的讲述，都辅以实际的例子实现，并随后附上实例的说明讲解和对实例运行结果的分析。读者通过结合实例的学习，能更好地掌握Java理论的相关知识点，并通过课后习题加深理解Java理论，进而提高读者的实际编程能力。本书配有《Java程序设计上机指导与习题选解》，进一步巩固教材中的知识点，提高学生动手能力。

本书具有以下特点：

①本书的定位是Java语言的入门教程，本书的读者不要求具有专门的计算机专业的基础知识和Java编程经验，通过本书的学习，读者可以掌握Java程序的思想，具有实际问题解决能力。

②本书注重基础知识的学习，讲究编程语言的实用性，为Java平台的后续学习打下牢固的基础。

③本书的编写体现了实践教学的思想，每一个案例都是精心设计的，是对相关知识的补充和巩固，使学习者能够很好地实现从理论知识到实际编程的自然过渡。

④理论联系实际，深入浅出，符合教学规律。本书的结构经过精心安排，内容的讲述由浅入深，基本上按照大多数人的学习习惯来安排每一章的内容结构。

⑤本书采用JDK 6.0版本来安排内容和实例，这是目前较新的Java开发包的版本，具有成熟性和先进性的特点。

本书由潘浩、张国英和曹正凤编写，张国英负责编写本书的第1～3章，潘浩负责第4～11章内容的编写，曹正凤编写了本书的第12章和第13章，最后由潘浩进行统稿。在此，我们要感谢易久教授，没有他的帮助，就没有本书的出版，感谢香港浸会大学的王大震博士为本书的编写提供的宝贵意见，感谢董琳媛教授和孙滨丽教授给予的诸多支持和帮助。

本书难免存在一些遗漏和不足之处，恳请读者批评指正，给出修改建议。可以通过“bjpanhao@163.com”邮箱地址直接与作者联系。

作　者

2008年3月

目　录

第 1 章　Java 语言与面向对象概述

本章要点：

- 了解 Java 语言的产生和发展。
- 掌握 Java 语言的特点。
- 了解 Java 的 3 种平台的特点和相互关系。
- 了解面向对象编程和面向过程编程的特点和区别。

1.1　Java 发展简史

Java 是由 Sun Microsystems 公司开发的新一代面向对象的编程语言，它具有跨平台、面向对象、安全、可靠、适用于网络等显著特点，是目前最为流行的网络编程语言之一。

Java 语言自推出后，得到了飞速的发展。Java 的语言特性、核心类及相关技术的添加，使得它的内容不断扩充。Java 不仅仅是一门编程语言，它也被称为 Java 平台，即 Java SE、Java EE 和 Java ME 平台。它们分别应用于桌面应用软件的编程、企业级分布式应用系统领域和嵌入式系统应用领域。

现在，Java 是一个分布式 Web 应用程序的开发标准，几乎所有的软件公司现在都在学习、研究和使用 Java，由 Java 开发的大型企业系统和移动应用系统占有很大的市场比例。

1.1.1　Java 发展简介

Java 语言是由 Sun 公司的 James Gosling 领导的绿色计划(Green Project)开发的。1991 年 4 月，Sun 公司的 James Gosling 领导的 Green Project 开始着力发展一种分布式系统结构，使其能够运行在不同的消费性电子产品上。Green Project 的实现方案是将源代码编译产生通用的中间码，通过在不同类型的机器上安装对应的虚拟机，这种中间码可以在任何一种特定的机器上解释运行。由于 Green Project 的成员都具有 C++语言背景，一开始使用 C++语言来完成这个项目，但很快就发现 C++语言在实现跨平台方面存在很多不足，需要研发一种新的语言来替代它，Java 语言就此诞生了。

Java 语言是从 C++语言而来的。Java 的开发者将 C++语言简化，去掉指针操作、运算符重载、多重继承等容易出错的部分，并将之变为一种解释执行的语言，在每个芯片上装上一个 Java 语言虚拟机，源代码编译成为二进制的字节码后，由 Java 虚拟机解释执行。在 17 个月后，由 Java 语言开发的系统完成了，它注重机顶盒式的操作系统。不过当时这方面的市场并不成熟，项目产品推出后，没有受到人们的青睐。到 1994 年为止，他们的产品甚至没有签订一份合同。直至 1994 年下半年，由于 Internet 的迅猛发展和万维网 WWW 的快速增长，第一个全球信息网络浏览器 Mosaic 诞生了。在 Java 出现以前，Internet 上的信息内容都是一些乏味死板的 HTML 文档。这对于那些迷恋于 Web 浏览的人们来说简直无法忍受。对于 Web 来

说，关键是要有一个好的浏览器，能够将超文本页面生动地显示到屏幕上。用户迫切希望能在 Web 上看到一些交互式的内容，开发人员也极希望能够在 Web 上创建一类无需考虑软硬件平台就可以执行的应用程序，当然这些程序还要有极大的安全保障。因此，工业界对适合在网络异构环境下使用的语言有一种非常急迫的需求。Games Gosling 敏锐地察觉到了这一点，他们意识到“我们可以研制一款出色的浏览器，它将追随客户机/服务器架构这一主流，提供一些真正有价值的东西，而这些东西我们已经做出来了，其中包括与平台无关性、实时性、可靠、安全等等。”他们用 Java 语言开发出 HotJava 浏览器的第一个版本，该浏览器能够解释字节码，能够运行 Java 小应用程序。该浏览器在 1995 年 5 月 23 日召开的 SunWorld'95 大会上首次亮相，从此一举成名。1996 年 1 月，Netscape 公司推出了 Netscape 2.0 版本，提供了对 Java 的支持。IBM、Oracle、Microsoft、Netscape、Apple、SGI 等大公司纷纷与 Sun Microsystems 公司签订合同，授权使用 Java 平台技术。经过十多年的发展，Java 技术已经成为世界上最卓越的企业应用和移动应用开发平台之一，Java 平台是目前应用最为广泛的平台之一。

Java 本身从推出到现在，经过十多年的发展，不断地完善和成熟。Sun 公司在 1996 年初发布了 Java 的第一个版本，几个月后，又发布了 Java 1.02 版本。这些版本并不适合进行正规的程序开发，缺乏对很多功能的支持。例如，不支持打印功能。1997 年推出的 Java 1.1 版本实现了许多 Java 类库，对原版本进行了很大的改进，基本上实现了 Java 的重要特性。1998 年问世的 Java 1.2 版本标志着 Java 2 平台的诞生，它改进了原来玩具一样的图形用户界面和图形工具包，使它的功能更加全面。2002 年 J2SE1.4 版本发布，使得 Java 平台的性能获得一个很大的提升。2004 年 Java 2 Platform，Standard Edition 5 发布，对 Java 的语法进行了完善。2006 年，J2SE 6.0 版本推出，使得 Java 类库的功能极大地完善和丰富。J2SE7.0 版本还未正式发布，正在开发之中。

1.1.2 Java 语言的特点

Java 是一种跨平台的、适合于分布式计算环境的面向对象编程语言。具体来说，它具有简单性、面向对象、分布式、解释型、安全可靠、与平台无关、可移植、高性能、多线程、动态性等特点。

1. 简单性

Java 最初是为对家用电器进行集成控制而设计的一种语言，因此它必须简单明了。Java 的风格类似于 C++，因而 C++程序员很容易掌握 Java 语法。Java 中去除了 C++中许多复杂并不容易理解的、极少使用的和令人混淆的功能，如删去了 C++的头文件、指针、结构、多重继承等功能，使得 Java 语言容易理解，并且降低了它的出错概率。Java 中增加了内存的自动垃圾收集功能，降低了程序员的编程难度和编程负担。Java 还提供了功能丰富的类库，Java 程序员通过引用类库中已经实现的功能，实现了代码的重复利用。

2. 面向对象

面向对象是 Java 最重要的特性。Java 不像 C++语言，需要兼容支持 C 语言那样的面向过程的程序设计技术。面向对象其实是现实世界模型的自然延伸。现实世界中任何实体都可以看作是对象，对象之间通过消息相互作用。除了为了兼顾高性能而没有将简单数据类型设计为对象以外，Java 的设计符合面向对象的特点。Java 支持封装、多态性和继承等面向对象特性。具体描述如下：

(1) 封装

现实世界中的对象均有属性和行为，映射到计算机程序上，属性则表示对象的数据，行为表示对象的方法。所谓封装，就是用一个自主式框架把对象的数据和方法联在一起形成一个整体。可以说，对象是支持封装的手段，是封装的基本单位。Java 语言提供了类机制，以对象为基本单位，封装了状态数据和方法。封装特性将对象的状态数据很好地保护起来，外界不能直接访问这些数据，只能通过调用对象的方法获得。

(2) 多态性

多态性就是多种表现形式，可以用“一个对外接口，多个内在实现方法”表示。Java 通过同一类中方法的重载以及子类对父类方法的覆盖实现多态性。方法的重载是指一个类中可以存在相同名字的多个方法，但这些方法的参数的数量和参数的类型不能完全相同。方法的覆盖发生在子类对父类的继承中，子类可以对父类中的方法进行重新定义和实现。多态性使方法的调用更加容易、灵活和方便。

(3) 继承

继承是指一个对象直接使用另一对象的属性和方法。客观世界中的很多实体都有继承的含义。例如，若把电视看成一个实体，它可以分成多个子实体，如黑白电视机、彩色电视机等。这些子实体都具有电视的特性，因此，电视是它们的“父亲”，而这些子实体则是电视的“孩子”。Java 提供的类继承机制只支持单继承，一个父类可以拥有多个子类，而一个子类只能有一个父类。子类不但拥有父类的所有特性，还可以增加新的特性。类继承提高了程序的可重复利用性，减轻了编程的负担。

3. 安全性和可靠性

Java 语言源于 C++语言，它去除了 C++的许多不可靠因素，可以防止许多编程错误，在此基础上还采取了其他的一些安全措施。

① Java 不支持指针操作，这杜绝了对内存的非法访问，从而不会发生由于编码失误造成有效数据被破坏的现象。

② Java 的自动单元收集机制防止了由于动态内存分配导致的内存丢失等问题。

③ Java 解释器运行时实施检查机制。在字节码装载过程中使用字节码检验器对字节码的信息进行检查，从而保证指令参数类型的正确性、对象域访问的合理性，及时发现数组和字符串访问的越界等错误。

④ Java 提供了异常处理机制，可以把一组错误处理代码放在一个地方，简化错误处理任务，便于恢复错误。

⑤ Java 主要用于网络应用程序开发中，通过自身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。使用类装载器，将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。从网络上装载的类只能访问某些文件系统，不能通过任何途径对本地类进行操作。

Java 语言通过将上述几种机制结合，使得它成为安全可靠的编程语言。

4. 平台无关性

Java 是平台无关的语言，用 Java 编写的应用程序不用修改就可在不同的软、硬件平台上运行，而不受计算机硬件和操作系统的限制。Java 基本数据类型的长度是固定的，与具体的计算机无关。

Java 源程序经过编译产生的二进制文件成为字节码，它是一种与具体机器指令无关的指令集合。字节码由 Java 虚拟机(Java Virtual Machine，JVM)在不同平台上解释执行。Java 虚拟机是一种抽象机器，它附着在具体操作系统之上，本身具有一套虚拟机指令，并有自己的栈、寄存器组等。

Java 虚拟机是 Java 平台无关的基础，在 Java 虚拟机上，有一个 Java 解释器用来解释 Java 编译器编译后的程序。Java 编程人员编写完软件后，通过 Java 编译器将 Java 源程序编译为 Java 虚拟机的字节码。任何一台机器只要配备了 Java 解释器，就可以运行这个程序，而不管这种字节码是在何种平台上生成的。

5. 多线程

C 和 C++采用单线程体系结构，而 Java 却提供了多线程支持。Java 在两方面支持多线程：一方面，Java 环境本身就是多线程的。若干个系统线程运行负责必要的无用单元回收和系统维护等系统级操作；另一方面，Java 语言内置多线程控制，可以大大简化多线程应用程序开发，实现支持多任务功能。

Java 提供了 Thread 类和一组内置的方法，负责启动运行、终止线程，并可检查线程状态。Java 的线程还包括一组同步原语，这些原语负责对线程实行并发控制。利用 Java 的多线程编程接口，开发人员只需继承 Thread 类和调用相应方法，就可以方便地写出支持多线程的应用程序，提高程序执行效率。然而，Java 的多线程支持在一定程度上受运行时支持平台的限制。如果操作系统本身不支持多线程，Java 的多线程特性可能就表现不出来。

6. 分布式

分布包括数据分布和操作分布。数据分布是指数据可以分散在网络中的不同主机上；操作分布是指把一个计算分散在不同主机上处理。Java 支持这两种分布。

Java 提供了内容丰富的网络类库，可以处理 HTTP、FTP 等 TCP/IP 协议。Java 提供了一个叫作 URL 的对象，利用这个对象，可以打开并访问对象，访问方式就像访问本地文件系统一样简单。Java 小程序可以从服务器下载到客户端，将部分计算在客户端进行，提高系统执行效率。

1.2 Java 虚拟机

Java 不仅仅是一种语言，更是一种区别于传统系统，遵循“网络就是计算机”信条的平台技术。Java 平台将面向对象系统扩展成包括程序和数据的网络计算机，而这个平台的核心就是 Java 虚拟机，Java 的平台无关性的实现源于 Java 虚拟机的概念。

1.2.1 Java 虚拟机的概念

从底层看，Java 虚拟机就是以 Java 字节码为指令组的软 CPU。如图 1-1 所示为 Java 程序的运行过程。从图中可以看出，在服务器端首先由开发人员编写 Java 程序并存储为“.java”文件；然后，Java 编译器将“.java”文件编译成由字节码组成的“.class”文件；最后，将“.class”文件存放在 Web 服务器上。到此，Java 程序已作为 Internet 或 Intranet 资源存放在 Web 服务器上随时可供客户使用。在客户机端，用户使用 WWW 浏览器，通过 Internet 或 Intranet

将 Web 服务器上的含有 Java 程序的主页下载，再依赖本地 Java 虚拟机对“.class”文件解释执行。这样，内容丰富的 Java 应用资源便由服务器传送到客户端，并在用户浏览器上显示出来。

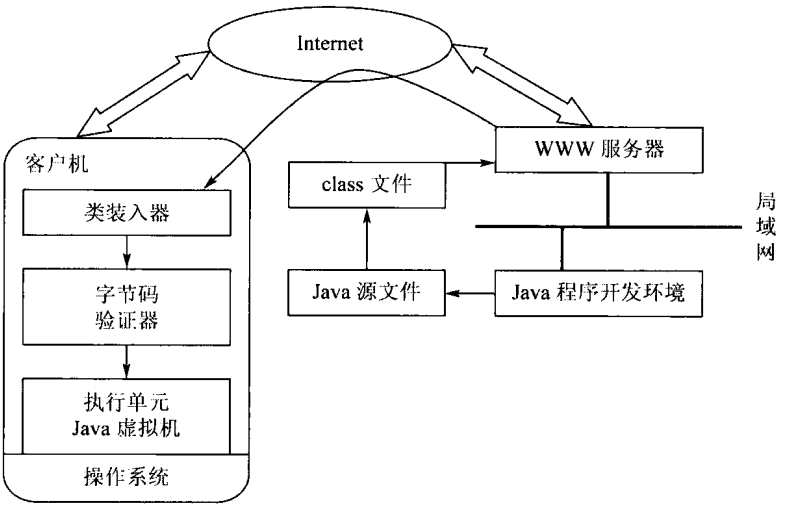

图 1-1　Java 程序开发和运行过程

Java 虚拟机执行程序时首先从网络或本地存储器中装入“.class”文件。由于网络的不安全因素较多，如黑客的恶意攻击、网络病毒的侵袭等，因此，Java 虚拟机在执行“.class”文件前，首先要对其进行验证。如果没有通过验证，则不能执行类文件，并给出错误信息。相反，如果程序成功地经过验证阶段，Java 虚拟机将运行翻译器读取字节码，把字节码转换成操作系统硬件相关的指令，并在真正的 CPU 上执行。

字节码是 Java 虚拟机的指令组（很像 CPU 上的微码）。由于字节码指令数目种类繁多，相对于机器码语义层次较高，因此，Java 语言编译成字节码后文件尺寸较小便于网络传输。为了运行 Java 字节码，硬件厂商或操作系统厂商在自己的硬件和操作系统组合中实现 Java 虚拟机。Java 虚拟机程序模块通常用 C、C++或相应 CPU 支持的汇编语言编写。Java 虚拟机用主机操作系统帮助完成内存、文件系统、显示器、鼠标、键盘、网络和其他设备驱动器以及线程处理等。

从概念上讲，Java 虚拟机的基本执行单元是“.class”文件。一个 Java 程序经过编译后将形成多个“.class”文件，而每个“.class”文件都对应一个类定义。因此，带有多个“.class”文件的 Java 程序在执行时与传统程序相比就有其特殊性。传统的程序在运行前，系统要装入含有全部程序码的单一执行文件，而 Java 虚拟机在执行程序时则不同，它遵循“即用即装入”的原则。具体讲，由于一个“.class”文件可以引用许多其他“.class”文件（在 Java 语言中，通过 import、implement 或 extends 语句实现），当运行的类需要其他类时，Java 虚拟机即从网络或本地文件系统装入“.class”文件。Java 虚拟机能够动态地装入和连接所需要的类文件。

1.2.2　Java 虚拟机的结构

Java 虚拟机包括方法区、Java 堆、Java 栈、程序计数器和本地方法栈这 5 个部分，它们与类装载机制和执行引擎机制一起组成的体系结构如图 1-2 所示。Java 虚拟机的每个实例都有一个自己的方法域和一个堆，运行于 Java 虚拟机内的所有的线程都共享这些区域。当虚拟机装载类文件的时候，它解析其中的二进制数据所包含的类信息，并把它们放到方法域中。当程

序运行的时候，Java 虚拟机把程序初始化的所有对象置于堆上，而每个线程创建的时候，都会拥有自己的程序计数器和 Java 栈。其中，程序计数器中的值指向下一条即将被执行的指令，线程的 Java 栈则存储为该线程调用 Java 方法的状态。本地方法调用的状态被存储在本地方法栈，该方法栈依赖于具体的实现。

执行引擎处于 Java 虚拟机的核心位置，在 Java 虚拟机规范中，它的行为是由指令集所决定的。Java 虚拟机支持大约 248 个字节码，每个字节码执行一种基本的 CPU 运算。Java 虚拟机规定了统一的数据类型以及类型长度。例如，整型为 32 位，浮点为 IEEE754 格式等。这样，对于不同的操作系统，运行于其上的 Java 的数据类型长度是相同的，保证了 Java 的跨平台特性。Java 虚拟机是栈式的，它不定义或使用寄存器来传递或接受参数，其目的是为了保证指令集的简洁性和实现时的高效性。Java 虚拟机的方法区是编译后的代码区域，它包括方法代码、符号表等。Java 虚拟机的栈有 3 个区域：局部变量区、执行环境区、操作数区。Java 虚拟机并不保证任何内存单元回收的时间安排，所以 Java 编程人员不要误以为只要对象不再在程序中使用，对象占用的内存就立即可用。这就意味着，开发需要大量内存的程序时必须格外小心，连续申请内存可能引发内存不足的错误。

由于 Java 程序可以从网络上下载运行，这必然带来许多不安全因素。一些恶意的黑客们可以直接写出字节代码段攻击用户节点，如删除文件或未经用户许可在网上发送本地信息等。这些防止本地机被攻击的任务就交给了 Java 虚拟机完成。Java 虚拟机提供了几种安全机制，其中两种主要的机制是安全管理器和 Java 类文件认证器。安全管理器是安全的实施者，它是一个可扩展类，提供加在应用程序和特写系统上的安全措施。安全管理器实现 Java 虚拟机的安全策略。

Java 类文件认证器在“.class”文件运行前完成该文件的安全检查，确保 Java 字节码符合 Java 虚拟机规范。Java 平台通过使用认证器查看类文件的句法和词法正确性，检查版本及应用编程接口（Application Programming Interface，API）符合性等，保证病毒和其他恶意程序不会侵犯本地系统。此外，认证器在检查期间还能识别算法操作的上溢和下溢等其他可能发生在运行期间的程序错误。

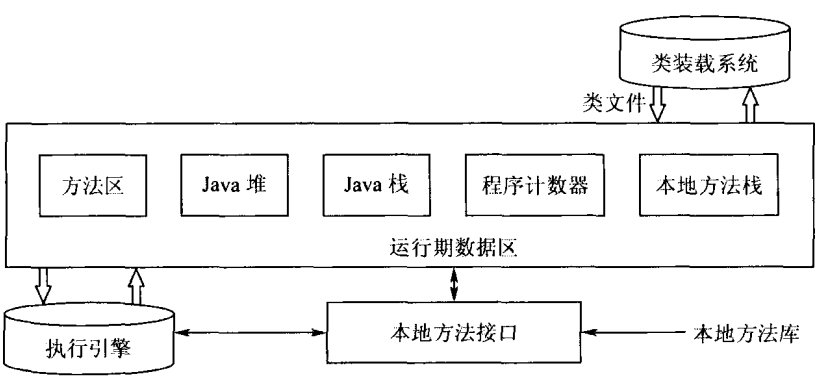

图 1-2 Java 虚拟机的结构组成

1.3 Java 2 平台的 3 种版本

Java 语言自推出以来，在互联网应用方面取得了很大的成功。随着 Java 技术的发展，为