21世纪高职高专规划教材

计算机系列

Java网络编程基础

主　编　殷兆麟

副主编　付慧生　赵纪平　姜利群

主　审　袁迎菊

清华大学出版社
http://www.tup.tsinghua.edu.cn

北方交通大学出版社
http://press.njtu.edu.cn

21 世纪高职高专规划教材·计算机系列

Java 网络编程基础

主　编　殷兆麟
副主编　付慧生　赵纪平　姜利群
主　审　袁迎菊
参　编　张爱娟　高　娟　汪玉华

清华大学出版社
北方交通大学出版社
·北京·

内 容 提 要

本书主要介绍 Java 面向对象编程和 Java 小应用程序编程的有关概念、术语、技术。这些是进一步学习 Java 网络编程和 Java 网络高级编程的基础。

本书共 11 章，1～8 章分别介绍 Java 语言特点、Java 开发环境基本知识、Java 语言基础、Java 面向对象的程序设计、异常处理、Java 基本类的使用、图形用户界面、多线程编程。这些内容是 Java 应用程序设计的基础知识。第 9 章简明介绍网络基本知识与 URL 编程，使没有学习过计算机网络的学生在学习本书也不会因缺少必要的网络知识而感到困难。第 10 章介绍 Java 小应用程序的开发。第 11 章介绍 Java 程序的存档及存档文件的使用。

本书可作为高职、高专计算机专业、电子商务专业、通信专业及相关专业的 Java 网络编程基础教材，也可作为 Java 爱好者、网络编程爱好者的自学教材。

图书在版编目（CIP）数据

Java 网络编程基础 / 殷兆麟主编 .—北京 ：清华大学出版社 ；北方交通大学出版社，2004.1
（21 世纪高职高专规划教材·计算机系列）
ISBN 7-81082-220-9

Ⅰ.J… Ⅱ.殷… Ⅲ.JAVA 语言-程序设计-高等学校：技术学校-教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2003）第 096662 号

责任编辑：谭文芳
出 版 者：清华大学出版社　邮编：100084　电话：010-62776969
　　北方交通大学出版社　邮编：100044　电话：010-51686045，62237564
印 刷 者：北京东光印刷厂
发 行 者：新华书店总店北京发行所
开　本：787×1 092　1/16　印张：19.5　字数：499 千字
版　次：2004 年 1 月第 1 版　2004 年 1 月第 1 次印刷
印　数：1～5 000 册　定价：26.00 元

21世纪高职高专规划教材·计算机系列
编审委员会成员名单

主任委员 李兰友　边奠英

副主任委员 周学毛　崔世钢　王学彬　丁桂芝　赵　伟
韩瑞功　汪志达

委　　员 （按姓名笔画排序）

马　辉	万志平	万振凯	王永平	王建明
丰继林	左文忠	叶　华	叶　伟	付晓光
付慧生	冯平安	江　中	佟立本	刘　炜
刘建民	刘　晶	曲建民	孙培民	邢素萍
华铨平	吕新平	陈小东	陈月波	李长明
李　可	李志奎	李　琳	李源生	李群明
李静东	邱希春	沈才梁	宋维堂	汪　繁
张文明	张权范	张宝忠	张家超	张　琦
金忠伟	林长春	林文信	罗春红	苗长云
竺士蒙	周智仁	孟德欣	柏万里	宫国顺
柳　炜	钮　静	胡敬佩	姚　策	赵英杰
高福成	贾建军	徐建俊	殷兆麟	唐　健
黄　斌	章春军	曹豫莪	程　琪	韩其睿
韩　劼	裘旭光	童爱红	谢　婷	曾瑶辉
管致锦	熊锡义	潘玫玫	薛永三	操静涛
鞠洪尧				

出 版 说 明

高职高专教育是我国高等教育的重要组成部分，它的根本任务是培养生产、建设、管理和服务第一线需要的德、智、体、美全面发展的高等技术应用型专门人才，所培养的学生在掌握必要的基础理论和专业知识的基础上，应重点掌握从事本专业领域实际工作的基本知识和职业技能，因而与其对应的教材也必须有自己的体系和特色。

为了适应我国高职高专教育发展及其对教学改革和教材建设的需要，在教育部的指导下，我们在全国范围内组织并成立了“21世纪高职高专教育教材研究与编审委员会”（以下简称“教材研究与编审委员会”）。“教材研究与编审委员会”的成员单位皆为教学改革成效较大、办学特色鲜明、办学实力强的高等专科学校、高等职业学校、成人高等学校及高等院校主办的二级职业技术学院，其中一些学校是国家重点建设的示范性职业技术学院。

为了保证规划教材的出版质量，“教材研究与编审委员会”在全国范围内选聘“21世纪高职高专规划教材编审委员会”（以下简称“教材编审委员会”）成员和征集教材，并要求“教材编审委员会”成员和规划教材的编著者必须是从事高职高专教学第一线的优秀教师或生产第一线的专家。“教材编审委员会”组织各专业的专家、教授对所征集的教材进行评选，对列选教材进行审定。

目前，“教材研究与编审委员会”计划用2～3年的时间出版各类高职高专教材200种，范围覆盖计算机应用、电子电气、财会与管理、商务英语等专业的主要课程。此次规划教材全部按教育部制定的“高职高专教育基础课程教学基本要求”编写，其中部分教材是教育部《新世纪高职高专教育人才培养模式和教学内容体系改革与建设项目计划》的研究成果。此次规划教材编写按照突出应用性、实践性和针对性的原则编写并重组系列课程教材结构，力求反映高职高专课程和教学内容体系改革方向；反映当前教学的新内容，突出基础理论知识的应用和实践技能的培养；适应“实践的要求和岗位的需要”，不依照“学科”体系，即贴近岗位群，淡化学科；在兼顾理论和实践内容的同时，避免“全”而“深”的面面俱到，基础理论以应用为目的，以必要、够用为度；尽量体现新知识、新技术、新工艺、新方法，以利于学生综合素质的形成和科学思维方式与创新能力的培养。

此外，为了使规划教材更具广泛性、科学性、先进性和代表性，我们希望全国从事高职高专教育的院校能够积极加入到“教材研究与编审委员会”中来，推荐“教材编审委员会”成员和有特色、有创新的教材。同时，希望将教学实践中的意见与建议及时反馈给我们，以便对已出版的教材不断修订、完善，不断提高教材质量，完善教材体系，为社会奉献更多更新的与高职高专教育配套的高质量教材。

此次所有规划教材由全国重点大学出版社——清华大学出版社与北方交通大学出版社联合出版。适合于各类高等专科学校、高等职业学校、成人高等学校及高等院校主办的二级职业技术学院使用。

21世纪高职高专教育教材研究与编审委员会

2003年9月

前　言

以Java技术为主线，以.net技术为辅线组织计算机应用类的专业课教学得到了越来越多的IT教育专家的认同。为此，我们为高职、高专编写了《Java网络编程基础》、《Java网络编程》、《Java高级网络编程》和《UML及其建模工具的使用》系列教材。本系列教材内容取材、编写以国家三个等级的计算机程序员职业标准为指南，吸取国外软件人员培训教材的优点，力争使教学内容循序前进、理论联系实际。把贯串职业标准的现代软件工程思想融合到相关专业课程的内容中。

《Java网络编程基础》的主要内容包括Java语言基础、Java面向对象的程序设计、Java输入输出流、图形用户界面、多线程程序设计、Java小应用程序的开发和Java数据库技术。这是Java技术教学的第一阶段。学生在掌握这些知识基础后，具有编写Java一般应用程序、小应用程序的能力。

《Java网络编程》的主要内容包括网络进程在网络传送层（Socket）同步和异步通信编程、Java Bean技术、Java命名服务、Servlet技术、JSP技术、J2EE技术。这是Java技术教学的第二阶段。学生在掌握这些知识基础后，具有开发电子商务、电子政务、企业ERP必需的Java基本技术。

《UML及其建模工具的使用》的主要内容包括软件工程基本知识、UML语言、UML建模、Rational Rose 2002的使用。Rational Rose与Java技术有密切关系。学生在掌握这些知识基础上，具有使用面向对象技术对一般中、小系统进行UML建模的基本能力。

《Java高级网络编程》可作为高职、高专选修课程的教材。主要内容包括Java RMI、Java CORBA技术、Java安全技术和移动代理编程技术，可进一步提高学生对当前Java分布技术的理解与运用。

本系列教材的组织编写得到江苏省教委“计算机网络人才培养模式的研究”项目及江苏省教委百门优秀课程教材基金的资助；有关教材的配套CAI课件、上机实习多媒体课件得到我校教改基金资助；系列教材建设被列为中国矿业大学新世纪教材工程项目。在此对有关单位表示衷心感谢。

本书适应国家计算机程序员职业标准要求，注意对开发环境的安装、配置、程序调试、存档等技术的介绍；本书示例丰富、由浅入深、图文并茂；习题体现全方位地培养学生编程思想，既注重基本知识的消化、又注重程序自编和示例程序的阅读、理解。与本书配套的授课课件、实验课件、网络自学课件可与主编联系（zhlyin@cumt.edu.cn）。

由于时间仓促，作者水平有限，书中难免有不妥和错误之处，恳请广大读者指正。

殷兆麟

2004年1月于徐州

目　录

第1章 Java语言概述

1.1 Java的语言特性

Java是一种适合于分布式计算机环境的面向对象的编程语言。它具有可移植、安全、面向对象、动态、高性能、简单、与体系结构无关性、动态执行等特性。

1.1.1 面向对象

Java语言的设计是完全面向对象的,这是Java最重要的特性。“面向对象”是目前非常流行的术语，其影响领域涉及操作系统、编程语言及其开发环境、数据库管理系统直至软件工程。

Java是一门“纯”面向对象的编程语言。在一门“纯”面向对象的语言中，语言的任何方面都是基于消息或基于对象的；所有数据类型，无论简单还是复杂，均为对象类。Java的基本数据类型如整型、字符型和浮点型也存在与这些基本数据类型相对应的对象数据类型。例如，基本类型int有对应的类Integer，float有对应的类Float。除基本数据类型外，Java中定义的数据类型都是类。这样，一方面保证了Java的高性能，另一方面又使得Java成为一门纯面向对象的编程语言。

1.1.2 可移植性

程序的可移植性指的是程序不经修改可在不同硬件或软件平台上运行的特性。可移植性包括两种层次：源代码级可移植性和二进制级可移植性。C和C++具有一定程度的源代码级可移植性，这表明C或C++源程序要能够在不同平台运行，必须重新编译。

Java采用了多种机制来保证可移植性，其中最主要的有以下两条。

（1）Java程序编译为字节码，由JUM解释执行

Java既是编译型又是解释型的。因此，Java编程人员在进行软件开发时，不必考虑软件运行平台。不仅开发的源代码是可移植的，甚至源代码经过编译之后形成的二进制代码——字节码，也同样是可移植的。任何一台机器只要配备了Java解释器，就可以运行Java二进制代码，而不管这种字节码是在何种平台上生成的。

（2）Java语言结构的中立性

Java采用的是基于国际标准的数据类型。Java的数据类型在任何机器上都是一致的。

Java编译器所生成的可执行代码并不基于任何具体的硬件平台，而是基于——Java虚拟机（Java Virtual Machine，JVM）。它通过预先把Java源程序编译成字节码，避免了传统的解释型语言的性能瓶颈，确保了其可移植性。

除给出基于JVM的虚拟字节码外，Java语言还规定同一种数据类型在所有各种实现中必须占据相同的空间大小。Java成功地保证了其程序的平台独立性，或者所谓的“体系结构中立性”（Architecture Neutral）。

Java的可移植性还体现在Java的运行环境上。编程人员在开发应用软件时，不用分别开

发 IBM PC 机版本、Macintosh 版本和工作站版本，等等，而只需开发一个“通用”的最终软件，大大加快了软件产品的开发。

1.1.3 稳定性和安全性

分布式计算环境要求软件具有高度的稳定性和安全性。为保证稳定性，Java 采用了 3 个措施：首先，Java 不支持指针数据类型，这样，程序员便不能凭借指针在任意内存空间，甚至是操作系统的内存空间中“遨游”；其次，它提供了检查机制，从而使网络“黑客”无法构造出类似 C 和 C++语言所支持的指针；最后，Java 还提供了自动内存管理机制，即一个自动的“内存垃圾”搜集程序。

除此之外，Java 的语言特性和运行时环境还保证 Java 代码具有良好的安全特性。除打算在字节码的传输过程中使用公开密钥加密机制（PKC）外，Java 的运行环境提供字节码校验器（ByteCode Verifier）、类装载器（Class Loader）、运行时内存布局、资源访问限制四级安全保障机制。

当 Java 字节码进入解释器时，首先必须经过字节码校验器的检查，这是非常重要的。即使 Java 编译器生成的是完全正确的字节码，解释器也必须再次对其进行检查，因为在从 Java 程序的编译到解释执行期间，字节码可能被有意无意地改动过。然后，Java 解释器将决定程序中类的内存布局，这意味着“黑客”无法预先得知一个类的内存布局结构，从而也就无法利用该信息来“刺探”或破坏系统。随后，类装载器负责把来自网络的类封装到其单独的内存区域，避免应用程序之间的相互干扰或破坏作用。最后，客户端管理员还可以限制从网络上装载的类只能访问某些被允许的文件系统。

上述机制综合在一起，使得 Java 成了最安全的编程语言和环境之一。

1.1.4 简单性

Java 语言简单而有效，其简单性主要出于如下几种因素。

Java 的风格类似于 C++，因而对 C++程序员而言是非常熟悉的；从某种意义上来说，Java 语言本身是 C 及 C++的一个变种，因此，C++程序员可以很快掌握 Java 编程技术。

Java 摒弃了 C++容易引发程序错误的地方，如指针和内存管理。

Java 提供了自动内存垃圾搜集机制，从而减轻了编程人员进行内存管理的负担，有助于减少软件错误。

Java 是完全面向对象的，它是最容易学习的面向对象编程语言之一；同时它还提供了大量可重用的类库。

Java 的简单性是以增加运行时系统的复杂性为代价的。以内存管理为例，自动内存垃圾处理减轻了面向对象编程的负担，但 Java 运行时系统却必须内嵌一个内存管理模块。但无论如何，对编程人员而言，Java 的简单性仍是一个优点。

1.1.5 高性能

解释型语言的执行效率一般低于直接执行源码的速度。但 Java 所采用的措施却很好的弥补了这些性能差距。这些措施包括以下方面。

1. 多线程

Java 的多线程支持体现在两个方面。首先，Java 环境本身就是多线程的，它可以利用系统的空闲时间来执行诸如必要的垃圾清除和一般性的系统维护等操作；其次，Java 还提供了对多线程的语言支持。利用其多线程编程接口，编程人员可以很方便写出支持多线程的应用程序，提高程序的执行效率。

2. 高效的字节码

Java 编译器生成的字节码和机器码的执行效率相差无几。Java 字节码格式的设计充分考虑了性能因素，其字节码的格式非常简单，这使得经由 Java 解释器解释执行时可产生高效的机器码。据统计，Java 字节码的执行效率非常接近于由 C 和 C++生成的机器码的执行效率。

1.1.6　动态特性

Java 的动态特性使得 Java 程序能够适应不断变化的执行环境。类库的使用是最明显的一个例子。

用 C++编写的应用程序经常会用到各种类库，除基础类库外，很多时候还需要一些从第三方厂商处购买的类库。类库一旦升级，用这些类库编写的应用程序就必须重新编译，并且重新发送到用户手中，否则就无法利用类库升级后的新增功能。

Java 的“滞后联编”却避免了这个问题。“滞后联编”机制使得 Java 完全利用了面向对象编程模式的优点。如前所述，Java 程序的基本组成单元为类，这些类是在运行过程中动态装载的。因此，Java 可以在分布时环境中动态地维护应用程序及其支持类库之间的一致性。这样，对于 Java 而言，其支持类库升级之后，相应的应用程序不必重新编译也一样可以利用类库升级后的新增功能。

1.2　虚拟机、Java 虚拟机与 Java 运行环境

1.2.1　虚拟机使语言与其运行环境无关

通常的语言程序如果要使计算机能执行就必须把它翻译成计算机机器语言。计算机配置了一种语言的翻译程序，使计算机能“明白”这种语言的程序。按语言程序的翻译方式不同，语言翻译程序分为解释型翻译程序和编译型翻译程序。

解释型翻译程序在语言程序翻译时，读入源程序一句，翻译一句，执行一句，这样反反复复直到最终完成。Basic 语言是典型的解释型语言。

编译型翻译程序也叫编译程序，它在翻译语言程序时，加工整个源程序，最终翻译成机器语言，然后交给计算机执行。编译程序有利于目标程序的优化和提高目标程序的运行速度。C，C++，Delphi 语言都是编译型语言。

不言而喻，语言翻译程序、由翻译程序产生的目标程序都与操作系统和计算机硬件有关。不同操作系统下的同一种语言的语言翻译程序是不一样的，同一操作系统下（如 Windows）硬件不一样（如 x86 和 PowerPC），同一种语言的翻译程序也是不一样的（如图 1-1 所示）。

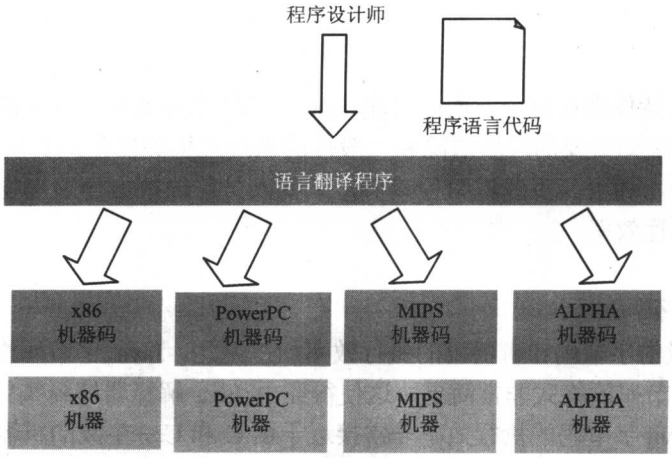

图 1-1　语言翻译程序与机器硬件有关

1.2.2　Java 虚拟机与 Java 运行环境

由上可见，翻译程序与操作系统和计算机硬件有关，为了提高翻译程序的可移植性，人们提出了虚拟机的理论。虚拟机好似通用的计算机，有自己的指令系统，但本身没有实际的硬件。为了虚拟机代码可以执行，必须由虚拟机实时运行支持系统把虚拟机代码转换成相应硬件机器的代码如图 1-2 所示，然后加以执行。有了虚拟机，编译程序首先把语言程序翻译成虚拟机代码，这样的编译程序可移植性就大大提高了。虚拟机代码与机器无关，虚拟机代码不仅可以在本机上执行，也可以通过网络流通到其他配制了相应的虚拟机实时运行支持系统的网点上执行。这一特征，显示了虚拟机在网络时代的无限生机。

计算机网络的发展，对计算机语言不断提出新的要求，特别是要求语言具有可移植性、安全性，这正是 Java 成为网络开发主流语言的原因。

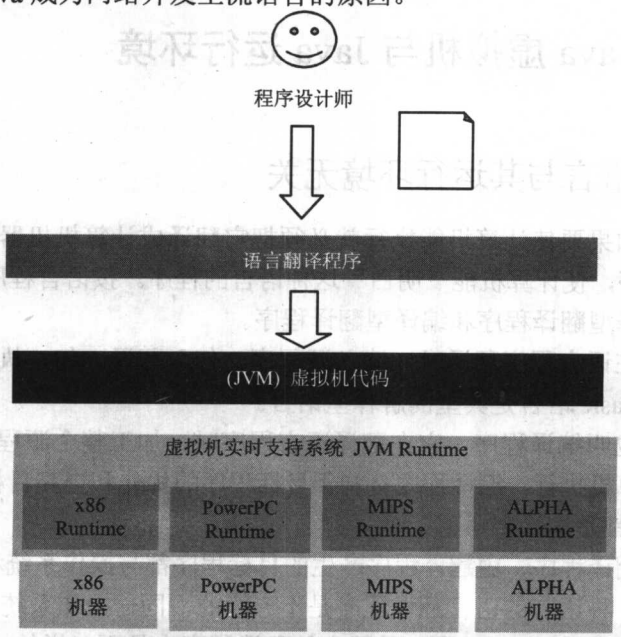

图 1-2　虚拟机原理示意图

Java 虚拟机（JVM）从结构上看与实际的计算机相似，它的作用是使一台实际的机器能够运行 Java 字节码程序。Java 字节码是 Java 源程序编译后的程序，它不能被计算机直接执行，但它可以被所有的 Java 虚拟机执行。这就是 Java 字节码程序可以在网络上移植的原因。

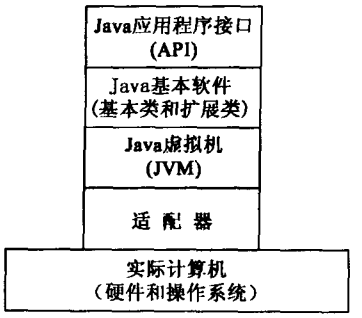

图 1-3　Java 运行环境

Java 程序必须有自己的运行环境。一个 Java 运行环境包括实际计算机、适配器、Java 虚拟机、Java 基本软件和 Java 应用程序接口，如图 1-3 所示。JVM 是 Java 运行环境的核心，JVM 的下面是与实际计算机连接的接口，被称为适配器（虚拟机实时运行支持系统），不同类型的计算机其适配器是不同的。Java 的基本软件也称为基本类，而 Java 应用程序接口（Application Program Interface，API）是已编译好的程序代码库，可以直接使用它们以节约编程的时间。事实上，Java 的基本类和 API 的规模并不固定，许多 Java 平台对这两部分内容进行补充，即除了基本类外，还有扩展类。

由此可见，Java 虚拟机与操作系统和计算机硬件无关。Java 经过编译后的字节码程序可以在网络上流动到任意支持 Java 虚拟机的网点机上运行。

利用 Java 语言可以开发两种形式的应用程序：Java 应用程序（Java Application）和 Java 小应用程序（Java Applet）。Java 小应用程序是借助浏览器运行的程序。

1.3　Java 程序的运行

Java 程序是半编译半解释型语言，其程序的运行过程与编译型和解释型语言都不同。

首先，编译程序将 Java 源程序编译成与实际计算机无关的字节码，然后 Java 运行系统解释并执行字节码。图 1-4 描述 Java 编译系统和运行系统的功能。

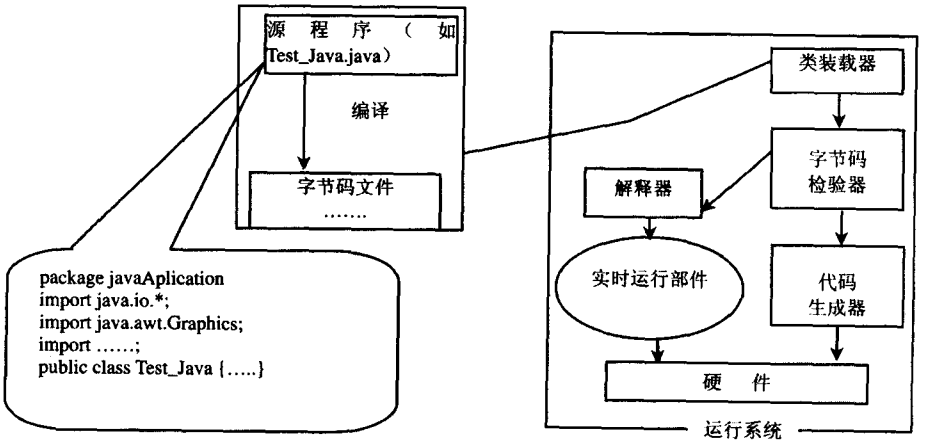

图 1-4　Java 程序编译及其运行

图中，运行系统解释 Test-Java.Java 应用程序，执行过程分以下三步进行。

第一步，由类装载器具体完成字节码的装载，装载时运行系统确定程序的内存分配。