高等学校教材

Java语言程序设计教程

主　编　邱桃荣
副主编　林振荣　冯　缨

机械工业出版社
CHINA MACHINE PRESS

高等学校教材

Java 语言程序设计教程

主　编　邱桃荣

副主编　林振荣　冯　缨

参　编　王炜立　洪胜华　伍军云

主　审　薛　峰

机 械 工 业 出 版 社

本书从 Java 语言自身的特点和学生学习 Java 语言的实际要求出发，将 Java 语言编程技术与面向对象程序设计相结合，帮助学生建立面向对象编程的主要原则和思维方法，全面介绍 Java 语言的特点和应用技术，着重实际应用能力的培养。

全书共分 14 章，内容安排合理，实用性很强。主要内容包括：软件开发和 Java 语言、Java 语言的基础、Java 面向对象的程序设计、Java 图形界面、Java 的事件处理机制、异常处理机制、Java 多线程、文件和流、Java 数据库编程、Java 网络编程、多媒体技术、Java Swing 基础。每一章都提供了大量的经过实际调试的例题供学生学习和模仿；同时，在每一章后都给出了相应的上机实验题和课后练习题供学生练习。

本书主要作为高等学校计算机及相关专业本科生、专科生、高职生和各类成人教育学院的面向对象程序设计课程以及 Java 程序设计课程教材，同时也可供自学人员使用。

图书在版编目（CIP）数据

Java 语言程序设计教程/ 邱桃荣主编. —北京：机械工业出版社，2004.7
高等学校教材
ISBN 7-111-14818-5

Ⅰ.J… Ⅱ.邱… Ⅲ. Java 语言-程序设计-高等学校-教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2004）第 063483 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
责任编辑： 于 宁
责任印制： 李 妍
北京蓝海印刷有限公司印刷 • 新华书店北京发行所发行
2004 年 7 月第 1 版第 1 次印刷
787mm×1092mm 1/16 • 18.5 印张 • 456 千字
定价：28.00 元

前　言

Java是一种流行、功能强大的编程语言，它完全面向对象、简单高效、与平台无关、安全、支持多线程，它与Web技术的有机结合尤其适宜进行网络计算和动态多媒体信息的开发处理。Java技术带来的是一场革命，它是第一个真正独立于平台的语言。基于Java语言开发的软件可以实现“一次设计，到处运行”。诞生于1995年的Java语言在短短的四五年间席卷全球，相信在21世纪里，Java计算技术将在网络应用、并行和分布处理、多媒体和虚拟实现技术、计算机辅助教育以及科学与工程计算等领域得到广泛的应用。

本书是作者在多年面向对象程序设计教学和科研实践的基础上进行的归纳总结，并结合相关资料编写而成的，主要介绍Java语言的编程技术。本书首先介绍了Java语言的基础知识和面向对象的设计方法，并主要讲授了Java语言的图形界面设计，事件处理，异常处理机制和Java语言的多线程设计方法。同时还简要介绍了在数据库和网络等具体环境中的应用方法。为了方便读者的学习和实践，本书最后还给出了Java集成开发环境JCreator平台的使用方法。由于考虑到读者往往在学习Java语言之前学习过一到两门其他的程序设计语言，所以本书把重点放在Java的面向对象和具体应用上面。

为了方便读者学习本书，在每一章节中除了介绍基本原理以外还给出大量的经过实际调试的示例程序供学生学习和模仿，同时在每个章节后面给出相应的上机试验题和课后练习题供读者练习。本书选材先进、科学，内容丰富、实用，章节安排按照从易到难、逐步递进的方式，可读性强。读者可以按照本书的顺序循序渐进地学习Java语言，也可根据自己的情况自由地选取其中章节进行学习。本书主要作为高等学校计算机及相关专业本科生、专科生和高职生的Java程序设计课程教材，同时也可供自学人员使用。为方便老师授课，本书特备有免费教学课件，有需要者可向责任编辑索要（010-88379758）。

本书由南昌大学计算机系邱桃荣老师主持编写，南昌大学的林振荣、王炜立、洪胜华、伍军云老师和江苏大学的冯缨老师参与编写。其中第1、2、3章由邱桃荣编写，第4、5章由林振荣编写，第6、7、13章由冯缨编写，第8、10章和附录由王炜立编写，第9、11章由伍军云编写，第12、14章由洪胜华编写。合肥工业大学计算机学院薛峰博士对全书进行审校，并提出了许多宝贵的建议。本书在编写过程中，还得到了南昌大学计算机系胡军博士的指导，以及系领导和其他同事们的大力支持，对此我们致以衷心的感谢。

计算机科学和技术日新月异，Java技术也发展迅速。由于作者的学术水平有限，编写时间较紧，书中难免会有错误和不足之处，敬请专家和读者批评、指正。

编　者

目　录

第 1 章　Java 语言概述

面向对象的软件开发和利用面向对象技术进行问题求解是当今计算机技术发展的重要成果和趋势之一，而 Java 语言的产生与流行则是 Internet 发展的客观要求。本章将简要介绍软件开发方法的变革和面向对象程序设计中的基本概念，介绍 Java 语言的特点及开发 Java 程序的基本步骤等，使读者对面向对象软件开发方法的基本思想和特点有一定的了解，熟悉 Java 语言特点、与 C/C++的主要差异、Java 程序执行过程、Java 运行环境及开发工具等基本知识。

1.1　面向对象的软件开发概述

人类已经进入了 21 世纪。21 世纪是信息社会，是知识经济的时代。信息是战略资源，信息的挖掘、增加、管理、流通、利用以及更新离不开计算机。计算机广泛又深刻地改变了人类的生活。计算机系统是由计算机硬件子系统与相应软件子系统构成的，计算机软件是计算机的灵魂。软件是相对计算机硬件而言的，是事先编制好的具有特定功能和用途的程序系统及其相应说明文件的统称。随着计算机硬件的发展和计算机的广泛应用，软件系统的发展也从简单到复杂、从小型到大型、从封闭到开放。就软件开发方法而言：

在 20 世纪四五十年代，由于每台计算机都是单独设计的，计算机作为价格昂贵的特殊计算工具，实现计算任务的程序由极其小部分专业人士专门编制，所以无需什么开发方法。

在 60 年代，虽然程序设计人员开始意识到软件的相对独立性的重要作用，但由于缺乏软件开发方法和技术，编程人员只能针对特定问题，根据所需功能，制定相应的方法。

在 70 年代，软件开发技术有了很大的发展，主要表现在：①数据结构与算法成为一种独立研究对象。通过对数据结构与算法的研究，提高了计算机的时空效率。②将结构化程序设计方法发展为结构化开发方法，提出了重要的软件开发模型——瀑布模型。软件开发也从依靠个人的技巧、经验和智慧发展到按系统方法通过遵从一系列规范进行开发的阶段。

在 80 年代，由于软件系统规模的扩大，单纯的编程技术已经不是开发软件系统的重点，而如何管理系统的结构，如何管理系统各部分之间的接口，如何将系统各部分集成为一体等成为软件系统开发技术的焦点。因此，这一时期强调的是开发小组的协作。

在 90 年代，由于微电子技术的发展，使微机的性能不断提高、价格不断下降，使软件向高质量的图形化界面、丰富的工具和集成开发环境方向发展；计算机软件的规模不断扩大，复杂程度日益提高，需要有多层次的抽象，以满足应用的需要；新的工程技术的发展，如多媒体技术、CAD 等需要描述许多复杂的事物，软件的发展速度远远落后于硬件的发展，不能满足应用的要求，需要有新的软件开发过程模型和新的方法论。基于这些原因，人们在综合以往软件开发中的各种概念和方法的基础上，采取了基于客观世界的对象模型的软件开发方法。

长期的实践表明，在一个软件系统中，与系统所要求的功能相比，问题论域中的对象一般是相对稳定的。例如，在一个教学管理信息系统中，问题论域中的对象如学生、教师、教

学管理人员等是相对稳定的，而系统的功能则可能随着对系统认识和环境的变化不断出现新的要求。例如以前学生学籍记录中无重修课程这一内容，现在却有这一要求，所以系统必须增加新的功能。面向对象的软件开发方法按问题论域来设计模块，以对象代表问题解的中心环节，力求符合人们日常的思维习惯，采用“对象+消息”的程序设计模式，降低或分解问题的难度和复杂性，从而以较小的代价和较高的收益获得较满意的效果，满足软件工程发展需要。

面向对象开发方法的出现和广泛应用是计算机软件技术发展的一个重要变革和飞跃。面向对象技术能够更好地适应当今软件开发在规模、复杂性、可靠性和质量、效率上的种种要求，因而被越来越多地推广和使用，其方法本身也在这诸多实践的检验和磨练中日趋成熟、标准化和体系化，逐渐成为目前公认的主流软件开发方法。

1.2 面向对象的基本概念

1.2.1 对象、类和消息

面向对象技术中的对象就是现实世界中某个具体的物理实体在计算机中的映射和体现。它既包括属性（描述对象的特征，可以是数据或对象，在 Java 语言中称之为变量），也包括作用于属性的操作（是对象执行的动作，可以是对象作出的或施加给对象的，在 Java 语言中称之为方法）。对象是由属性和操作所构成的一个封闭整体。比如，小汽车是现实世界中的一个具体的物理实体，它拥有颜色、车门以及行驶速度等外部特性，具有刹车、加速和减速等内在功能。这样的实体，在面向对象的程序中，就可视为一个“基本程序模块”，可以表达成一个计算机可理解的、可操作的具有一定属性和操作的对象，通过数据结构和提供相应操作来实现。如：

属性用：int color;int door;int speed 等变量来表示。

操作用：void brake{……};

void speedUp{……};

void speedDown{……}等方法表示。

对象在计算机内存中的映像称为实例。对象之间可能存在包含、关联和继承三种关系。包含关系是指整体与部分之间的关系，当对象 X 是对象 Y 的属性时，称对象 Y 包含对象 X。如汽车与轮胎的关系就是一个包含关系。我们知道每辆汽车都对应一个生产厂商，如果把生产厂商抽象成对象，则汽车对象可以或应该记录自己的生产厂商是哪个。这种通过一个对象可以找到另一个对象的关系称为关联关系。在面向对象的 Java 语言中，把可以找到另一个对象的线索称为引用。因此当对象 X 的引用是对象 Y 的属性时，称对象 X 和对象 Y 之间是关联关系。继承关系我们将在下面做详细介绍。

类是面向对象技术中一个非常重要的概念，它是描述对象的“基本原型”，是描述性的类别或模板，即对一组对象的抽象。它定义一组对象所能拥有的共同特征（属性和能完成的操作），用以说明该组对象的能力与性质。在面向对象的程序设计中，类是程序的基本单元，对象是类的实例。如定义 Car 是一个小汽车类，它描述了所有小汽车的性质（包括小汽车的颜色、车门数、速度等）及基于属性的各种操作（刹车、加速以及减速等操作功能）。

对象的动作取决于外界给对象的刺激，这就是消息，即消息是对象之间进行通信的一种数据结构。消息告诉对象要求它完成的功能，程序的执行是靠对象间传递消息来连接的，即

所谓的消息驱动。消息一般由三部分组成，即消息的接收对象名、消息操作名和必要的参数。消息传送与传统的函数调用的主要差别有以下几点：

- 函数调用可带或不带参数，但消息至少带一个参数（即接收该消息的对象）。
- 消息操作名类似于函数名，但他们有本质的不同。函数名代表一段可执行的代码，而消息名的具体功能选定还取决于接收消息的对象本身。
- 函数调用是过程式的，而消息传送是说明式的，具体如何做由对象根据收到的消息自行确定。

1.2.2　封装性、继承性和多态性

1．封装性

所谓封装又称为信息隐蔽，是面向对象的基本特征。封装的目的在于将使用者与设计者分离，使用者不必知道操作实现的细节，只需用设计者提供的消息来访问对象。比如，汽车作为一个对象，则汽车的设计者与制造者通过提供一组操作面板让用户使用这辆汽车，用户不必知道这些面板操作是如何实现的细节，这样就实现了对象的设计者与使用者的分离。在面向对象中封装可按下面具有三个内涵的方式定义：

1）一个清楚的界面，所有对象的内部软件的范围被限定在这个边界内。

2）一个接口，这个接口描述了该对象与其他对象之间的相互作用。

3）受保护的内部实现，这个实现提供对象的相应的软件功能，实现细节不能在定义这个对象的类的外面访问。

由于封装使得对象访问局限于被良好定义的并受控制的界面，这样就防止了由于程序相互依赖而带来的不良影响，这对软件的可靠性设计是很重要的。封装本身即模块性，把定义模块和实现模块分开这就使得用面向对象技术所开发的软件的维护性、修改性大为改善，这是软件技术追求的目标之一。

2．继承性

由于类是具体对象的抽象，所以就可以有不同级别的抽象（形成不同级别抽象的过程称为分类），这样就形成像一棵倒立的树一样的类层次关系。图 1-1 是运输工具的不同级别抽象的分类树。分类是我们组织知识的常用方式，当我们以这种分层方式对对象进行分类时,位于分类树顶的对象包括下面的所有对象范畴。当一个对象范畴出现在分类树中时，它满足在分类树中位于它之上的所有对象范畴的属性，如在图 1-1 中位于客运工具之下的所有范畴都共享四轮的、自驱动的和设计成运送乘客的公共特性。

由此可见，继承是面向对象语言中的一种重要机制，该机制自动地为一个类提供来自另一个类的操作和属性，这样程序员只需在新类中定义已有类中没有的属性与（或）操作来建立新类。假定用结点表示类对象，用连接两结点的无向边表示它们之间的关系，则可用数据结构中的树型图来表示类层次结构，就像图 1-1 一样。在这个树型图中，称子女结点（设为 X 类）是其父结点（设为 Y 类）的子类或派生类，而父结点 Y 类称为子类 X 的超类或父类或基类。如在图 1-1 中，航天工具类、航空工具类、陆地工具类和水上工具类均是运输工具类的子类，而运输工具类是航天工具类、航空工具类、陆地工具类和水上工具类的父类。子类 X 由两部分组成：继承部分和增加部分。继承部分是从父类 Y 中继承下来的，而增加部分是专门为子类 X 编写的新的属性和操作，增加部分可有也可无。如在四轮工具类中没有载客量这个属性，而四轮工具类的子类客运工具类就增加了载客量这个属性。

继承是面向对象程序设计的基本要素。在面向对象软件开发中利用继承性有助于开发快速原型。继承性是实现软件重用性的最有效机制，促进了系统的可扩充性。

当一个类派生出另一个类时，子类继承了父类的属性和操作，这称为单一继承（简称单继承）。当一个类处在类层次树中的多个直接父类时，称这种继承为多重继承（简称多继承）。

图 1-2 是一个多重继承的示意图。其中类 F 从类 C 和类 D 中继承属性和操作，属于多重继承，而类 B、类 C、类 D、类 E 和类 G 为单重继承。Java 语言只提供单重继承。

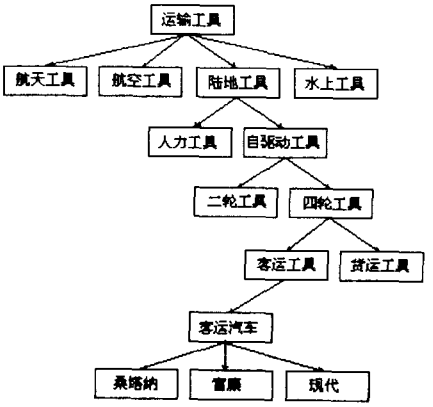

图 1-1 继承分类树章

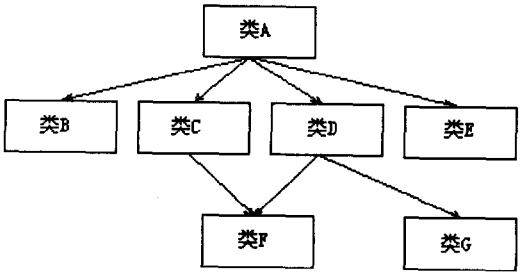

图 1-2 单重继承与多重继承

3. 多态性

多态性是指一个名字具有多种语义，即指同一消息为不同对象所接受时，可以导致不同的操作。在面向对象编程语言中,它是指对象拥有同名，但不同参数格式的许多方法的能力。如：同一个方法名为 abs 的三个方法：abs(int x)、abs(float x)和 abs(double x)，分别对应不同类型的参数（整型、单精度实型和双精度实型），则编译器会根据用户所给定的参数类型，选择对应的方法进行响应。程序设计的多态性有两种基本形式：编译时多态性和运行时多态性。编译时多态性是指在程序编译阶段就可确定选择哪个方法的多态性。而运行时的多态性则必须等到程序动态运行时才可确定的多态性。例如，假定一个程序担负通过通信链接发送对象的责任，该程序可能直到发送对象时也不知道对象所属的类，这种把决定对象所属的类和访问对象的操作的时间推迟到运行时，就是动态多态性。这种能力称为动态联编。

多态性具有使顶层代码只写一次，而底层代码可多次重复使用，实现“同一接口，多种方法”的功能。多态性使程序的表达方式更加灵活，使程序的表示形式尽可能地与所表示的内容无关。

1.3 Java 语言概述

随着 Internet 与 WWW 的兴起和不断发展，需要开发许多大型软件系统，如何简化这些大型系统的开发、设计和维护，使系统具有灵活性、可移植性和互操作性，成为软件开发必须考虑的问题。Java 语言的诞生正是应允了这个要求。Java 语言有着很好的网上移植性、安全性，并且在编程难度上比 C/C++语言简单。

Java 是一个学习起来很有趣但又有点复杂的语言，首先看看它的产生历史。

1.3.1　Java 产生的历史

Java 于 1995 年 5 月公布。1991 年，Sun 公司的 Bill Joy,Patick Naughton,Mike Sheridan 和 James Gosling 等人开始从事一个叫 Green 分布式代码系统项目开发，研制该项目的目的是：通过 E-mail 给电冰箱、电视机等家用电器发送信息，对它们进行控制，和它们进行信息交流。开始，这些开发人员准备采用 C++，但觉得 C++太复杂，安全性差，最后他们基于 C++开发出一种新的语言 Oak(Java 的前身，含义为橡树)。Oak 是一种基于网络的精巧而安全的语言，Sun 公司曾依此投标一个交互式电视项目，但当时并没有投标成功，并且也未引起人们的广泛重视。1994 年下半年，由于 Internet 的迅猛发展以及 WWW 的快速增长，Oak 项目组成员受到了启发，他们意识到 Oak 非常适用于 Internet，于是他们用 Oak 编制了 HotJava 浏览器，并得到了 Sun 公司首席执行官 ScottMcNealy 的支持，促进了 Oak 的研究和开发。随后 Oak 改名为 Java 语言，进军 Internet，并逐渐成为 Internet 上受欢迎的开发与编程语言。

1.3.2　Java 的现状与发展前景

Java 从孕育到现在已经有十年多了，但总的说来，Java 还很年轻。与其他开发语言（如 Pascal、C、C++、VB）相比，使用 Java 语言的程序员会感觉到很多在 Windows 中常用的 API 函数都没有，速度也较慢。但随着 Java 的不断完善与发展，Java 代码的执行效率也随之得到很大的提高，并且与 C/C++相比，Java 具有安全性、跨平台、多线程等优势，特别适合于 Internet 应用程序开发。

由于 Internet 和 WWW 进一步普及，目前几乎所有的软件公司都在学习、研究并使用 Java，采用 Java 语言开发出的大型实用系统已经越来越多。Sun、IBM、Oracle 以及 Netscape 等公司都在大力推进 Java 的应用。Sun 公司的高层人士称 Java 的潜力远远超过作为编程语言带来的好处。从来没有任何一种计算机语言在这么短的时间内就得到这么广泛的响应，取得这么大的成功，由此可见 Java 确实是网络上的“世界语”。

1.3.3　Java 语言的特点

1. 什么是 Java

按照 Sun 公司的说法：Java 是一种简单的(Simple)、面象对象的(Object Oriented)、分布式的(Distributed)、解释的(Interpreted)、健壮的(Robust)、安全的(Secure)、结构中立的(Architecture Neutral)、可移植的(Portable)、性能优异的(High Performance)、多线程的(Multithreaded)动态的(Dynamic)语言。面向对象是指 Java 以一组对象来组织其程序。这实际上不是 Java 独有的特性，其他一些编程语言像 Smarttalk 和 C++已经运用了好几年这种面向对象的结构，但 Java 是一种更容易掌握的程序（特别是与 C 或者 C++相比）。Java 是在 C++相当强大之后才初具规模的。在应用程序开发之中，C++是一种相当普遍的基本程序语言。开发环境由 Unix 转到 Macintosh 又到 Windows 都可以使用 C++。但问题是无论从语言上还是二进制水平来看，C++在这些平台上不易于互相移植，因为虽然上面提到的每一个操作系统都可使用 C++，但都以不同的方式来使用，这就意味着在不同的平台间转换时，需要重新写这个程序，并需在运行的环境中选择其编译器来编译。

计算机程序必须被翻译成计算机处理器能够识别的机器代码后才能够被执行。在 C++、Pascal 等编程语言中，这个工作是由编译器来完成的，编译后形成的机器代码称为可执行的二进制映像文件。不同的处理器能够识别的机器代码是不同的。所以，为了使程序可以在计算机网络环境下所有的机器上运行，则必须分别在每一台机器上都重新编译一次程序。为了

解决这个问题，Java 语言为每个计算机系统都提供一个叫做 Java 虚拟机（JVM）的环境，它包括一个编译器和一套软件系统。Java编译器把Java源程序翻译成被称为字节码的中间代码。与 Java 源程序一样，字节码也是与计算机系统无关的，同一个字节码文件可以被任何计算机所使用。图 1-3 说明传统语言工作模型和 Java 的工作模型的差别。

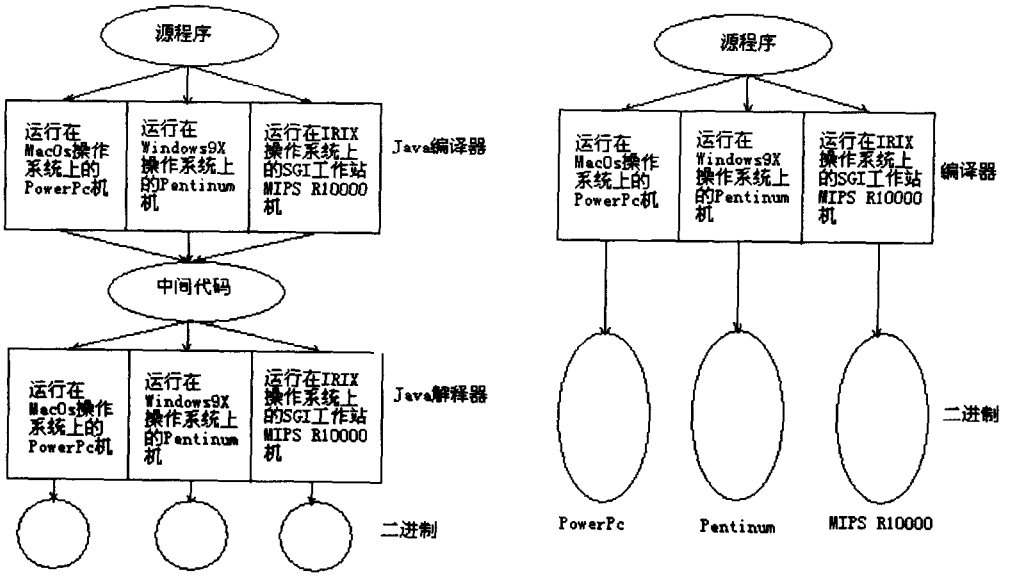

图 1-3 Java 与传统语言工作模型的差别

图 1-4 说明了 Java 程序的执行过程。在图 1-4 中，载入器用于调入包含、继承所用到的所有类，确定内存分配，变成真正可执行的机器码。由于网络的不安全因素较多，因此 Java 虚拟机在执行.class 文件前，首先要对其进行验证。校验器就是用于检验是否有伪造的指针、是否有违反访问权限、非法访问对象和导致操作栈溢出。不同的操作系统有不同的虚拟机

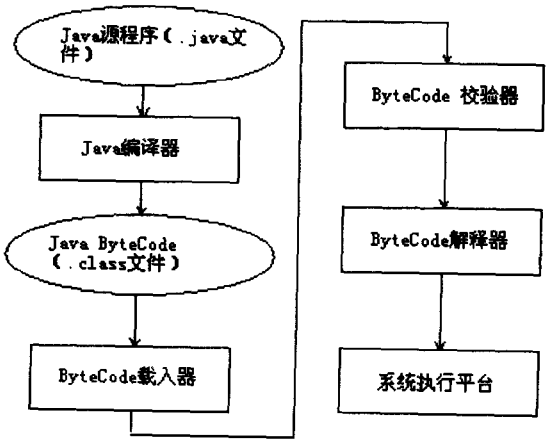

图 1-4 Java 程序的执行过程

（JVM），虚拟机是一个软件系统，它可以翻译并运行 Java 字节码。虚拟机首先翻译 Java 源程序为字节码，然后再执行翻译所生成的字节代码，属于先解释后执行方式，它类似一个小巧而高效的 CPU。字节码（byte-code）是与平台无关的是虚拟机的机器指令。

Java 字节代码运行有两种方式：解释方式和即时编译（Just-in-time）。对于有些程序而言，采用解释方式执行程序，运行速度会很慢。为了提高速度，Java 为每个系统都提供了可以直接把字节码文件编译成可执行的映像文件的编译器，Java 把这类编译器称为即时编译器（JIT），它们被捆绑在一些 Web 浏览器中。

Java 除了具有平台独立性之外，与其他一些语言如 C++，smarttalk 和 Visualbasic 相比，也很容易使用，而且很容易学。此外，Java 加强了 C++的功能，除去了一些过于复杂的部分。

2．Java 语言的特点

（1）简单性　Java 是个精简的系统，无需强大的硬件环境便可以很好地运行。Java 的风格和语法类似于 C++，从某种意义上讲，Java 语言是 C 及 C++语言的一个变种，因此，C++程序员可以很快就掌握 Java 编程技术。Java 摒弃了 C++中容易引发程序错误的地方，如多重继承、运算符重载、指针和内存管理等，Java 语言具有支持多线程、自动垃圾收集和采用引用等特性。Java 提供了丰富的类库，方便用户迅速掌握 Java。

（2）面向对象　面向对象可以说是 Java 最重要的特性。Java 语言的设计完全是面向对象的，它不支持类似 C 语言那样的面向过程的程序设计技术。所有的 Java 程序和 applet 均是对象，Java 支持静态和动态风格的代码继承及重用。

（3）分布式　Java 包括一个支持 HTTP 和 FTP 等基于 TCP/IP 协议的子库。因此，Java 应用程序可凭借 URL 打开并访问网络上的对象，就像访问本地文件一样简单方便。Java 的分布性为实现在分布环境尤其是 Internet 下实现动态内容提供了技术途径。

（4）健壮性　Java 是一种强类型语言，它在编译和运行时要进行大量的类型检查。类型检查帮助检查出许多开发早期出现的错误。Java 自己操纵内存减少了内存出错的可能性。Java 的数组并非采用指针实现，从而避免了数组越界的可能。Java 通过自动垃圾收集器避免了许多由于内存管理而造成的错误。Java 在程序中由于不采用指针来访问内存单元，从而也避免了许多错误发生的可能。

（5）结构中立　作为一种网络语言，Java 编译器将 Java 源程序编译成一种与体系结构无关的中间文件格式。只要有 Java 运行系统的机器都能执行这种中间代码。从而使同一版本的应用程序可以运行在不同的平台上。

（6）安全性　作为网络语言，安全是非常重要的。Java 的安全性可从两个方面得到保证。一方面，在 Java 语言里，象指针和释放内存等 C++功能被删除，避免了非法内存操作。另一方面，当 Java 用来创建浏览器时，语言功能和一类浏览器本身提供的功能结合起来，使它更安全。Java 语言在你的机器上执行前，要经过很多次的测试。它经过代码校验，检查代码段的格式，检测指针操作，对象操作是否过分以及试图改变一个对象的类型。另外，Java 拥有多个层次的互锁保护措施，能有效地防止病毒的入侵和破坏行为的发生。

（7）可移植　Java 与体系结构无关的特性使得 Java 应用程序可以在配备了 Java 解释器和运行环境的任何计算机系统上运行，这成为 Java 应用软件便于移植的良好基础。但仅仅如此还不够。如果基本数据类型设计依赖于具体实现，也将为程序的移植带来很大不便。Java 通过定义独立于平台的基本数据类型及其运算，使 Java 数据得以在任何硬件平台上保持一

致，这也体现了 Java 语言的可移植性。还有 Java 编译器本身就是用 Java 语言编写的，Java 运算系统的编制依据 POSIX 方便移植的限制，用 ANSIC 语言写成，Java 语言规范中也没有任何“同具体实现相关”的内容，这说明 Java 本身也具有可移植性。同时 Java 语言的类库也具有可移植性。

（8）解释的 Java 解释器(运行系统)能直接对 Java 字节码进行解释执行。链接程序通常比编译程序所需资源少。

（9）高性能 虽然 Java 是解释执行程序，但它具有非常高的性能。另外，Java 可以在运行时直接将目标代码翻译成机器指令。

（10）多线程 线程有时也称小进程，是一个大进程里分出来的小的独立运行的基本单位。Java 提供的多线程功能使得在一个程序里可同时执行多个小任务，即同时进行不同的操作或处理不同的事件。多线程带来的更大的好处是具有更好的网上交互性能和实时控制性能，尤其是实现多媒体功能。

（11）动态性 Java 的动态特性是其面向对象设计方法的扩展。它允许程序动态地装入运行过程中所需要的类，而不影响使用这一类库的应用程序的执行，这是采用 C++语言进行面向对象程序设计时所无法实现的。

3. Java 与 C/C++和 C#比较

（1）Java 与 C 和 C++语言的主要差异 C 或 C++是几乎所有基于 Windows 的软件的主要开发语言，同时也是用于编写 Windows95 和 NT 操作系统的语言。Java 基于 C++，与之有许多相似之处，但其设计更易于使用，它们之间主要差异有：

- Java 中无 C / C++中最复杂并有潜在危险的指针。
- Java 无 C/C++中的#include 、#define 和头文件。
- Java 无 C/C++中的 structure,union 及 typedef。
- Java 无 C/C++中的函数、指针和多重继承。
- Java 无 C/C++中的 goto 指令。
- Java 无 C/C++中的操作符重载(Operatior Overloading)、自动类型的转换。
- Java 系统要求对对象进行相容性检查，以防止不安全的类型转换。Java 是动态的，可以自动管理内存，减少了程序中内存丢失的情况，有效地利用了系统资源。而 C/C++要求用户自己释放所不再使用的内存，如果用户一旦忘记释放就可能使内存资源耗尽，造成系统崩溃。
- Java 语言最强大的特性之一是它的平台独立性，Java 可以处理好平台之间的移植问题。
- Java 语言中没有全局变量的定义，只能通过公用的静态的变量实现，从而减少了引起错误的地方。

（2）Java 与 C#相似之处 2000 年 6 月，微软发布 C#语言和.NET 平台。C#语言是一种强类型的面向对象的语言。它具有语法简单、表达力强的特点。而.NET 平台则是构成微软的“.NET 计划”的基石。

.NET 平台的核心包括两方面：一方面就是著名的通用语言运行机 CLR(Common Language Runtime)。通用语言运行机类似 Java 的虚拟机，二者完成的任务大致相同；另一方面就是一大堆通用库函数，这些库函数可以被多种语言调用，并且通过编译都产生一种共同

的中间语言（Intermediate Language）。这种中间语言类似于 Java 的字节码，虽然两者完成的方式有些不一样。

可以说 C#是微软用来和 Java 抗衡的武器。二者在很大程度上有着惊人的相似，主要有以下几个方面。

- 二者都编译成跨平台的、跨语言的代码。具有自动回收垃圾内存，并且消除了指针（在 C#中可以使用指针，不过必须注明 unsafe 关键字）。
- 二者都不需要头文件，所有的代码都被“包(package)”限制在某个范围内，并且因为没有头文件，所以消除了类定义的循环依赖。
- 所有的类都是从对象派生出来，并且必须使用 new 关键字分配内存。
- 二者都用对象加锁的方式来支持多线程。
- 二者都具有接口(interface)的概念。
- 二者都没有全局函数或者常量，一切必须属于类。
- 二者数组或者字符串都自带长度计算和边界检查。
- 二者都使用“.”操作符，而没有“->”和“::”操作符。

1.3.4　Java 的应用程序类型和 Java 相关技术名词介绍

1. Java 的应用程序类型

用 Java 可以开发几乎所有的应用程序类型，主要有以下几种：

- 多平台应用程序：Java 是跨平台的应用开发工具，用 Java 开发的网络应用系统可以在各种平台上运行，大大增加了开发效率，减少重复劳动。
- Web 应用程序：开发 Web 应用程序是 Java 的基本功能。Web 浏览是现在国际网甚至局域网的主要使用方式。文档能很容易地显示文本和各种图片，并提供超文本链接。
- 基于 GUI 的应用程序：用 Java 语言可以开发出一般 Windows 下的标准图形用户界面。
- 面向对象的应用程序：由于 Java 是一种纯面向对象的编程语言，因此常用 Java 语言开发面向对象的应用程序。
- 多线程应用程序：利用 Java 语言提供的多进程机制可以方便开发各种动画应用等程序。
- 关键任务的应用程序：如电子商务和数据库方面的应用程序。
- 分布式网络应用程序：Java 是网络编程语言，常利用 Java 进行分布式网络应用程序的开发，如 Sun 公司的 hotJava 浏览器就是用 Java 开发的。
- 安全性应用程序：Java 设计为在其编译器、运行系统及相应的浏览器中嵌入多层安全机制。

2. Java 相关技术名词介绍

自 1995 年以来，Java 的应用范围越来越广，也因此派生出许多专用名词。

1）JVM（Java Virtual Machine），即 Java 虚拟机。它是一个软件系统，它可以翻译并运行 Java 字节码。它是 Java 的核心，保证了在任何异构的环境下都可运行 Java 程序，解决了 Java 的跨平台的问题。

2）JRE（Java Runtime Environment），即 Java 运行环境。JRE 只是 Java 的运行环境，提供了 Java 程序运行所需的基本类库，它包含 JVM。

3）JDK（Java Development Kit），即 Java 开发环境。Sun 公司开发的一个免费的 Java 开发工具集，提供了 Java 开发、运行和测试一体的环境，它包含完整的 JRE。

4）Servlet Servlet 是指利用 Java 技术设计的、运行在服务器端的一种程序。它的功能类似于传统的 CGI，可以接收来自浏览器的请求，动态地生成响应。

5）JSP（Java Server Pages），是一种以 Java 为主的跨平台 Web 开发语言。

6）AWT（Abstract Window Toolkit），即抽象窗口工具包，用于支持图形用户界面编程，是 Java 基础类库 JFC 的一部分。它提供了用于设计用户界面的组件以及事件处理模型，它还包括一系列图形与图像工具、布局管理器以及用于本地平台的剪贴板交换数据的传送类。

7）JFC（Java Function Class），即 Java 基础类库，是一系列预先写好的 100%纯 Java 的 GUI 组件和 API，可用于快速开发多功能的 Java 程序。

8）J2EE（Java 2 Platform，Enterprice Edition），即 Java 2 企业级平台，是 Sun 公司推出的为解决分布式企业级运算的一套开发与运行平台。J2EE 由 EJB 和八种企业级 API 组成。八种企业级 API 对应于八种服务，分别是 Java 命名和目录接口（JNDI，Java Naming and Directory Interface）、Java 数据库互连（JDBC，Java Database Connectivity）、Java 远程调用（RMI，Remote Method Invocation）、Java 管理 API（JMAPI，Java Management API）、Java 事务 API（JTA，Java Transaction API）、Java 事务服务（JTS，Java Transaction Service）、Java 消息服务（JMS，Java Messaging Service）、Java 安全 API(Java Security API)和 Java Server&Servlet。它对开发基于 Web 的多层应用提供了功能支持。如今，J2EE 已广泛成为开发企业级服务端解决方案的首选平台。

9）JavaBean JavaBean 是一种专门为 Java 软件开发者设计的全新的组件技术。

10）EJB（Enterprise Java Bean），即企业级 Java 组件。它提供了一个框架来开发和实施分布式商务逻辑，是“只需编写一次，可以四处运行”的中间层组件，它由实现业务规则的方法组成，由此显著地简化了具有可伸缩性和高度复杂的企业级应用的开发。

11）RMI（Remote Method Invocation），即远程方法调用。它的功能是让分布式应用程序中间层内的远程对象可以互相通信。RMI 是一种 EJB 使用的更下层的协议，它能够实现分布式的企业计算模式，实现了在运行于不同虚拟机的对象之间的方法调用。

12）JINI JINI 技术规范提供了构成电子设备、服务和应用程序网络所使用的机制。可使范围广泛的多种硬件和软件（即可与网络相连的任何实体）能够自主联网。JINI 的目标是最大限度地简化与网络的交互性。

13）JDBC（Java Database Connectivity），即 Java 数据库连接。它是由 Sun 公司提出的一系列对数据库进行操作的规范，它以一种统一的方式来对各种各样的数据库进行存取。它可以向数据库提交 SQL 查询和检索以及处理 SQL 查询的结果。

14）JNDI（Java Naming and Directory Interface），即 Java 命名和目录接口。命名服务为定位分布式对象提供了机制。目录服务在层次结构中组织分布式对象和其他资源（如文件）。JNDI 提供访问各种各样关于用户、机器、网络、服务和应用程序的信息，使开发人员以标准方式开发无缝连接企业命名和目录接口，使用者就可以通过这些统一的界面去访问低层的各种服务。

15）JMS（Java Message Service），即 Java 通信服务。JMS 用于编写业务应用程序，它们可以异步发送和接收数据。

1.4　基于 Windows 系统中 Java 程序的举例

1.4.1　Java 程序的开发环境

一个软件系统的开发过程包括设计、编码、测试、调试、文档生成、系统维护和系统升级。选择哪种开发环境是由多种因素决定的，但一个好的软件开发环境对软件开发的效率和质量都是至关重要的。Java 程序的开发环境有基于 Java 开发工具（JDK）和基于集成软件的开发环境（JDE）。

1. Java 开发工具（JDK）

Java 开发工具（JDK）是一些程序集合，它可以帮助开发者编译、运行和调式程序。开发者可以从 Sun 公司的 Java 站点（http://Java.sun.com）上下载 JDK。下面对 JDK 作一简单介绍。

JDK 中有 Java 编译器和字节码解释器。appletviewer 是小应用程序的字节码解释器。

（1）JDK 的安装　直接执行下载的安装文件后，根据提示就可正常安装。

（2）安装后产生如下目录结构　JDK 安装目录*\bin 的目录下包含以下主要文件：

- javac:Java 编译器，用来将 Java 程序编译成 bytecode。
- java:Java 编译器，执行已经转换成 bytecode 的 Java 应用程序。
- jdb:Java 调试器，用来调试 Java 程序。
- javap:反编译，用来返回 Java 程序的成员变量及方法等信息。
- javadoc:文档生成器，创建 HTML 文件。
- javaprof:资源分析工具，用于分析 Java 程序在运行过程中都调用了哪些资源，包括类和方法的调用次数和时间，以及各数据类型的内存使用情况等。javah:C 代码处理工具，用于从 Java 类调用 C++代码。appletviwer:Java 解释器，用来解释已经转换成 bytecode 的 Java 小应用程序。

（3）设置环境变量　为了能从任何目录编译和运行 Java 程序，需要在计算机系统中设置 PATH 和 CLASSPATH 变量。

① 设置 PATH 变量。PATH 变量用于标识安装 Java 开发工具包（JDK）的位置。假设将 JDK1.3 安装在 D 盘的根目录下，则需要将环境变量 PATH 设置为 d:\jdk1.3\bin。可按下列步骤进行设置：

- 打开 Control Panel（控制面板）。
- 双击 Control Panel（控制面板）对话框内的 System（系统）图标。
- 在 System Properties（系统属性）对话框内，单击 Advanced（高级）选项卡。
- 在 Advanced（高级）页面上，单击 Environment Variables（环境变量）按钮。
- 在 Environment Variables（环境变量）对话框中，选择修改（编辑）PATH 变量，将 d:\jdk1.3\bin 包含在其中，并以分号分开。
- 保存设置。

设置时最好在 PATH 变量的末尾插入“;.”（分号和句号），句号说明是当前目录。在设置 PATH 变量后，就可以方便编译和运行 Java 程序。然而在运行程序时，可能需要从与包含类文件的目录所不同的另外一个目录中运行.class 文件。为了达到这个目的，需要设置 CLASSPATH 变量。