高等学校Java课程系列教材

微课版

教学视频·教学资源·上机实践

Java程序设计精编教程

（第3版）

◎ 耿祥义 张跃平 编著

微课版
70 HOURS
70小时
教学视频

- 基础与实战。相关概念及知识点都辅以相应的实例，通俗易懂，便于理解掌握面向对象的编程思想。
- 实用与流行。涵盖了Java开发过程中重要的及流行的方法和技巧，讲解细致，环环相扣。
- 教学与互动。文字叙述注重可读性，知识组织注意合理性，体现了JDK1.8最新特性。
- 32学时教学+16学时上机实践，143个典型实例。

清华大学出版社

高等学校Java课程系列教材

Java程序设计精编教程

（第3版）

◎ 耿祥义 张跃平 编著

清华大学出版社
北京

内 容 简 介

Java 语言具有面向对象、与平台无关、安全、稳定和多线程等优良特性，是目前软件设计中极为强大的编程语言，特别适合于网络应用程序的设计，已经成为网络时代最重要的语言之一。本书精选 Java 核心内容，注重结合实例，循序渐进地向读者介绍 Java 语言的核心内容，在基础语言上强调 Java 面向对象编程的思想，在实用类上侧重应用。

全书分为 14 章，分别讲解简单数据类型、运算符、表达式和语句、类与对象、子类与继承、接口与实现、内部类与异常类、常用实用类、Java 输入输出流、组件与事件处理、Java 多线程机制、Java 网络编程、JDBC 数据库操作等内容。

本书使用的 JDK 版本是 JDK 1.8(即 JDK 8)，并提供了 70 小时微课教学视频，扫描每章提供的二维码可观看视频讲解。

本书适合高等院校计算机专业作为 Java 语言程序设计的教材以及想掌握 Java 核心内容的自学者。

图书在版编目(CIP)数据

Java 程序设计精编教程：微课版/耿祥义，张跃平编著. —3 版. —北京：清华大学出版社，2017
(高等学校 Java 课程系列教材)
ISBN 978-7-302-47316-9

Ⅰ. ①J…　Ⅱ. ①耿… ②张…　Ⅲ. ①JAVA 语言－程序设计－教材　Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 124416 号

责任编辑：魏江江
封面设计：刘　键
责任校对：白　蕾
责任印制：沈　露

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn,010-62795954
印 装 者：北京泽宇印刷有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：21.25　　**字　　数**：516 千字
版　　次：2010 年 8 月第 1 版　2017 年 9 月第 3 版　　**印　　次**：2017 年 9 月第1 次印刷
印　　数：20501～22500
定　　价：39.50 元

产品编号：075446-01

前　言

本书精选Java核心内容和重要的实用技术，注重Java语言的面向对象特性，强调面向对象的程序设计思想，在实例上侧重实用性和启发性，在类、对象、继承、接口等重要的基础知识上侧重编程思想，在实用类、输入输出流、Java网络技术、JDBC数据库操作等实用技术方面侧重实用。通过本书的学习，读者可以掌握Java面向对象编程的思想和Java编程中的一些重要技术。

在第3版中，除了更新了JDK 1.8(也称为JDK 8)的版本外，在内容上做了适当的调整，特别增加了每章的上机实践内容。对第14章做了比较大的调整，由原来的操作Access数据库，更改为操作JDK新版本自带的Derby数据库。删除了很少使用的第15章的内容。

全书共分为14章。第1章主要介绍Java产生的背景和Java平台，读者可以了解到Java是怎样做到“一次写成，处处运行”的；第2章讲述Java程序的基本结构；第3章讲解简单数据类型；第4章主要介绍Java运算符和控制语句；第5章、第6章和第7章是本书的重点内容之一，讲述类与对象、子类与继承、接口与实现等内容；第8章讲解内部类和异常类，特别强调使用内部类的原则以及学习自定义异常的重要性；第9章讲述常用的实用类，包括字符串、日期、正则表达式、模式匹配以及数学计算等实用类，特别讲解怎样使用Scanner类解析字符串；第10章讲解Java中的输入输出流技术，特别介绍怎样使用输入输出流来克隆对象、Java的文件锁技术以及使用Scanner解析文件等重要内容；第11章是基于Java Swing的GUI图形用户界面设计，讲解常用的组件和容器，特别详细讲解事件处理；第12章讲述Java多线程技术，通过许多有启发的例子来帮助读者理解多线程编程；第13章讲解Java在网络编程中的一些重要技术，涉及URL、Socket、InetAddress、DatagramPacket等重要的类，特别讲解Java远程调用(RMI)；第14章主要讲解Java怎样使用JDBC操作数据库，讲解了预处理、事务处理、批处理等重要技术。

本书实例的源程序以及电子教案可以从清华大学出版社网站(www.tup.com.cn)上免费下载，以供读者学习使用；也可以加入耿祥义教材教学QQ群(238455879)讨论相关内容。

编　者

2017年6月

目　录

第1章 Java入门

主要内容

- Java的平台无关性；
- Java的地位；
- 安装JDK；
- 一个简单的Java应用程序。

学习Java语言需要读者曾系统学习过一门面向过程的编程语言，例如C语言。读者学习过Java语言之后，可以继续学习与Java相关的一些重要内容，例如，可以学习和数据库设计相关的Java Database Connection(JDBC)、Web设计相关的Java Server Page(JSP)、手机程序设计相关的Android程序设计、网络信息交换相关的eXtensible Markup Language(XML)以及网络中间件设计相关的Java Enterprise Edition(Java EE)，如图1.1所示。

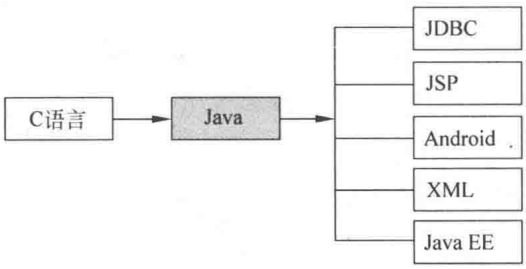

图1.1 Java的先导知识与后继技术

1.1 Java的平台无关性

Java语言的出现是源于对独立于平台语言的需要，希望这种语言能编写出嵌入各种家用电器等设备的芯片上、且易于维护的程序。但是，人们发现当时的编程语言，例如C、C++等都有一个共同的缺点，那就是只能对特定的CPU芯片进行编译。这样，一旦电器设备更换了芯片就不能保证程序正确运行，就可能需要修改程序并针对新的芯片重新进行编译。

1.1.1 平台与机器指令

无论哪种编程语言编写的应用程序都需要经过操作系统和处理器来完成程序的运行，因此这里所指的平台是由操作系统(Operating System，OS)和处理器(Central Processing Unit，CPU)所构成。与平台无关是指软件的运行不因操作系统、处理器的变化导致发生无法运行或出现运行错误。

所谓平台的机器指令就是可以被该平台直接识别、执行的一种由 0、1 组成的序列代码。需要注意的是,相同的 CPU 和不同的操作系统所形成的平台的机器指令可能是不同的,因此,每种平台都会形成自己独特的机器指令,例如,某个平台可能用 8 位序列代码 1000 1111 表示一次加法操作,以 1010 0000 表示一次减法操作,而另一种平台可能用 8 位序列代码 1010 1010 表示一次加法操作,以 1001 0011 表示一次减法操作。

1.1.2 C/C++程序依赖平台

下面讨论为何 C/C++语言编写的程序可能因为操作系统的变化、处理器升级导致程序出现错误或无法运行。

C/C++语言提供的编译器对 C/C++源程序进行编译时,将针对当前 C/C++源程序所在的特定平台进行编译、连接,然后生成机器指令,即根据当前平台的机器指令生成机器码文件(可执行文件)。这样一来,就无法保证 C/C++编译器所产生的可执行文件在所有的平台上都能正确地被运行,这是因为不同平台可能具有不同的机器指令(如图 1.2 所示)。因此,如果更换了平台,可能需要修改源程序,并针对新的平台重新编译源程序。

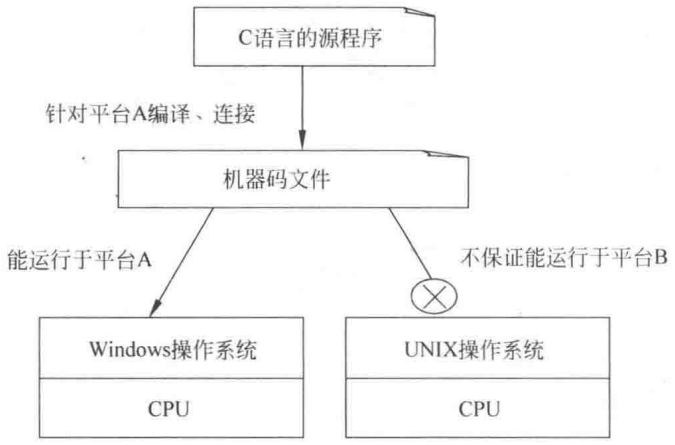

图 1.2 C/C++生成的机器码文件依赖平台

1.1.3 虚拟机与平台

Java 语言和其他语言相比,最大的优势就是它的平台无关性,这是因为 Java 可以在平台之上再提供一个 Java 运行环境(Java Runtime Environment,JRE),该 Java 运行环境由 Java 虚拟机(Java Virtual Machine,JVM)、类库以及一些核心文件组成。Java 虚拟机的核心是所谓的字节码指令,即可以被 Java 虚拟机直接识别、执行的一种由 0、1 组成的序列代码。字节码并不是机器指令,因为它不和特定的平台相关,不能被任何平台直接识别、执行。Java 针对不同平台提供的 Java 虚拟机的字节码指令都是相同的,例如所有的虚拟机都将 1111 0000 识别、执行为加法操作。

与 C/C++不同的是,Java 语言提供的编译器不针对特定的操作系统和 CPU 芯片进行编译,而是针对 Java 虚拟机把 Java 源程序编译为称作字节码的一种“中间代码”,例如,Java 源文件中的+被编译成字节码指令 1111 0000。字节码是可以被 Java 虚拟机识别、执行的

代码，即 Java 虚拟机负责解释运行字节码，其运行原理是：Java 虚拟机负责将字节码翻译成虚拟机所在平台的机器码，并让当前平台运行该机器码，如图 1.3 所示。

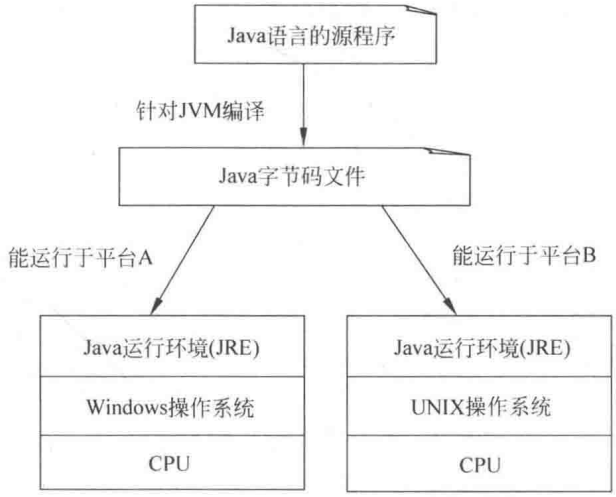

图 1.3　Java 生成的字节码文件不依赖平台

1.2　Java 之父——James Gosling

Java 是 1995 年 6 月由 Sun 公司引进到我们这个世界的革命性的编程语言，它被美国的著名杂志 *PC Magazine* 评为 1995 年十大优秀科技产品。1990 年 Sun 公司成立了由 James Gosling 领导的开发小组，开始致力于开发一种可移植的、跨平台的语言，该语言能生成正确运行于各种操作系统、各种 CPU 芯片上的代码。他们的精心研究和努力促成了 Java 语言的诞生。1995 年 5 月 Sun 公司推出 Java Development Kit(JDK)1.0a2 版本，标志着 Java 的诞生，而 Java 的快速发展得益于 Internet 和 Web 的出现，Internet 上的各种不同计算机可能使用完全不同的操作系统和 CPU 芯片，但仍希望运行相同的程序，Java 的出现标志着真正的分布式系统的到来。

注：印度尼西亚有一个重要的盛产咖啡的岛屿叫 Java，中文译名为爪哇，开发人员为这种新的语言起名为 Java，其寓意是为世人端上一杯热咖啡。

1.3　Java 的地位

1.3.1　网络地位

网络已经成为信息时代最重要的交互媒介，那么基于网络的软件设计就成为软件设计领域的核心。Java 的平台无关性让 Java 成为编写网络应用程序的佼佼者，而且 Java 也提供了许多以网络应用为核心的技术，使得 Java 特别适合于网络应用软件的设计与开发。

1.3.2　语言地位

Java 是面向对象编程，并涉及网络、多线程等重要的基础知识，是一门很好的面向对象

语言。通过学习 Java 语言不仅可以学习怎样使用对象来完成某些任务、掌握面向对象编程的基本思想,而且会为今后进一步学习设计模式奠定一个较好的语言基础。C 语言无疑是非常基础和实用的语言之一,目前,Java 语言已经获得了和 C 语言同样重要的语言地位,即不仅是一门正在被广泛使用的编程语言,而且已成为软件设计开发者应当掌握的一门基础语言。

1.3.3 需求地位

目前,由于很多新的技术领域都涉及 Java 语言,例如,用于设计 Web 应用的 JSP、设计手机应用程序的 Android 等,导致 IT 行业对 Java 人才的需求正在不断地增长,可以经常看到许多培训或招聘 Java 软件工程师的广告,因此掌握 Java 语言及其相关技术意味着较好的就业前景和工作酬金。

1.4 安装 JDK

Java 要实现“编写一次,到处运行”(write once,run anywhere)的目标,就必须提供相应的 Java 运行环境,即运行 Java 程序的平台。目前 Java 平台主要分为下列三个版本。

1.4.1 平台简介

1. Java SE

Java SE(曾称为 J2SE)称为 Java 标准版或 Java 标准平台。Java SE 提供了标准的 Java Development Kit(JDK)。利用该平台可以开发 Java 桌面应用程序和低端的服务器应用程序。当前最新的 JDK 版本为 JDK 1.8,Sun 公司把这一最新的版本命名为 JDK 8,但人们仍然习惯地称作 JDK 1.8。

2. Java EE

Java EE(曾称为 J2EE)称为 Java 企业版或 Java 企业平台。使用 Java EE 可以构建企业级的服务应用,Java EE 平台包含了 Java SE 平台,并增加了附加类库,以便支持目录管理、交易管理和企业级消息处理等功能。

Java 标准平台的核心是 Java 虚拟机,虚拟机负责将字节码文件(包括程序使用的类库中的字节码)加载到内存,然后采用解释方式来执行字节码文件,即根据相应平台的机器指令翻译一句执行一句。

1.4.2 安装 Java SE 平台

学习 Java 最好选用 Java SE 提供的 Java 软件开发工具箱 JDK。Java SE 平台是学习掌握 Java 语言的最佳平台,而掌握 Java SE 又是进一步学习 Java EE 和 Android 手机开发所必需的。目前有许多很好的 Java 集成开发环境(Integrated Development Environment,IDE)可用,例如 NetBean,Eclipse 等。Java 集成开发环境都将 JDK 作为系统的核心,非常有利于快速地开发各种基于 Java 语言的应用程序。但学习 Java 最好直接选用 Java SE 提供的 JDK,因为 Java 集成开发环境的目的是更好、更快地开发程序,不仅系统的界面往往比较复杂,而且也会屏蔽掉一些知识点。在掌握了 Java 语言之后,再去熟悉、掌握一个流行的