普通高等教育“计算机类专业”规划教材

高级程序设计语言

（Java版）

邱仲潘 张星成 宋智军 编著

清华大学出版社

普通高等教育“计算机类专业”规划教材

高级程序设计语言
（Java版）

邱仲潘 张星成 宋智军 编著

清华大学出版社
北京

内 容 简 介

Java 语言是美国 SUN 公司(现已被 Oracle 公司收购)开发的一种功能强大的语言,具有简洁、面向对象、分布式、可移植等性能的多线程动态计算机编程语言。Java 非常适合于企业网络和 Internet 环境,现在已成为 Internet 中最受欢迎、最有影响的编程语言之一。

本书循序渐进,按节细化知识点,并结合知识点介绍了相关的实例。读者可以按照实例编写程序,同时学习 Java 知识,能较快提高程序设计水平。

本书适合作为大学非计算机专业的教材,也可以作为高职高专院校计算机专业的教材,还可作为初学者的自学用书。

图书在版编目(CIP)数据

高级程序设计语言(Java 版)/邱仲潘等编著. --北京:清华大学出版社,2013
普通高等教育"计算机类专业"规划教材
ISBN 978-7-302-33032-5

Ⅰ. ①高… Ⅱ. ①邱… Ⅲ. ①JAVA 语言－程序设计－高等学校－教材 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 145963 号

责任编辑:白立军
封面设计:常雪影
责任校对:白　蕾
责任印制:王静怡

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 刷 者:北京富博印刷有限公司
装 订 者:北京市密云县京文制本装订厂
经　　销:全国新华书店
开　　本:185mm×260mm　　**印　　张**:20　　**字　　数**:484 千字
版　　次:2013 年 8 月第 1 版　　**印　　次**:2013 年 8 月第1次印刷
印　　数:1～2000
定　　价:35.00 元

产品编号:049885-01

这是个信息时代，不仅是计算机专业学生要学习编程语言，其他专业的本科生、研究生也需要学习一种或者几种编程语言。要学习编程语言，当首推 Java！因为这是个开源系统，资料很容易获得。

Java 是由美国 SUN 公司（现已被 Oracle 公司收购）开发的一种功能强大的，具有简洁、面向对象、分布式、可移植等性能的多线程动态计算机编程语言，是跨平台的程序设计语言，可以在各种类型的计算机和操作系统上运行。Java 语言非常适合于企业网络和 Internet 环境，现在已成为 Internet 中最受欢迎、最有影响的编程语言之一。Java 是 Sun Microsystems 公司的 James Gosling 等人于 20 世纪 90 年代初开发的，最初名为 Oak，目标设定在家用电器等小型系统的编程语言，来解决诸如电视机、电话、闹钟、烤面包机等家用电器的控制和通信问题。由于这些智能化家电的市场需求没有预期的高，SUN 公司放弃了该项计划。在 Oak 几近失败之时，随着互联网的发展，SUN 公司看到 Oak 在计算机网络上的广阔应用前景，于是改造了 Oak，在 1995 年 5 月 23 日以 Java 的名称正式发布了。Java 伴随着互联网的迅猛发展而发展，并逐渐成为重要的网络编程语言。

本书深入浅出地介绍 Java 编程语言，图文并茂，理论联系实际，便于理解；习题难度合适，便于巩固。学习这个语言之后，再学习其他编程语言就不难了，很容易融会贯通。

在本书编写过程中，我们征求了许多老师和同学们的意见，期望使教材更加合用。但是，由于时间关系，加上水平有限，错漏之处在所难免，欢迎老师同学们批评指正，以便我们在改版时尽快纠正。

编者

2013 年 3 月

FOREWORD

第1章 绪 论

学习目的与要求

本章主要介绍Java语言的特点以及Java所应用的平台，然后带领读者从第一步做起并完成第一个Java程序，通过对简单Java程序的学习来了解Java环境的搭建和简单开发步骤。

本章主要内容

(1) 了解程序设计语言的发展历程。

(2) 了解不同类型程序设计语言的特点。

(3) 理解Java程序设计语言的实现机制。

(4) 掌握安装并配置Java语言开发环境。

1.1 编程语言的发展历程

1.1.1 机器语言

机器语言是由0和1组成的二进制串，是计算机能够直接识别和执行的一种机器指令的集合。它是计算机诞生和发展初期所使用的语言，对于不同系列CPU的计算机所对应的机器语言是不相同的。

在使用机器语言编写程序时，编程人员要在熟记计算机的全部指令及指令含义的基础上，不仅要处理每条指令和数据的存储分配，还要记住编程过程中每步所使用的工作单元处在何种状态等。从而使编程人员既要考虑程序设计的全局，又要花费大量的时间和精力去考虑每一个局部繁杂琐碎的细节。只有经过系统训练的编程人员才能确保程序的正确性、高效性。目前，除了计算机生产厂家的专业人员外，绝大多数据编程人员都已不再学习机器语言了。下面通过一个例子说明机器语言的编程特点。

【例1-1】 在8086 CPU型的计算机中实现两个数据100和256相加功能。

机器语言代码如下：

机器语言代码	对应的十六进制
10111000 01100100 00000000	B8 64 00
00000101 00000000 00000001	05 00 01
10100011 00000000 00100000	A3 00 20

可以看出，由机器语言编写的程序全是0和1的指令代码，书写和阅读这些代码并不容易，需要编程人员对计算机的所有细节熟悉，而程序的质量完全取决于个人的编程水平。上面只是一个非常简单的求两数之和代码，就暴露了机器语言的工作量大、直观性差、难读、难写、易出错、不易查错等缺点，只适合专业人员使用。由于机器语言是针对特定型号计算机的语言，计算机可以直接识别，并不需要进行任何翻译，故运算效率是所有语言中最高的。

1.1.2 汇编语言

为了克服机器语言的缺点，人们就试图用一些容易记忆和简洁的英文字母、符号串来替代某一个特定指令的二进制串，这样就形成了汇编语言。如用 ADD 表示加法，SUB 表示减法，MUL 表示乘法，MOV 表示传送数据等。用汇编语言编写的程序叫做汇编语言源程序，计算机不能直接运行汇编语言源程序，必须通过专门的翻译器将这些符号翻译成二进制的机器语言才能被计算机识别和运行。

在使用汇编语言编写程序时，编程人员可依据指令系统和汇编语言的规定进行编写源程序（汇编语言源程序的扩展名为 asm），然后再利用汇编程序（ASM 或 MASM）对源程序进行汇编生成可重定位的目标程序（目标程序的扩展名为 obj），最后利用链接程序（LINK）将一个或多个.obj 文件进行链接，生成可执行文件（可执行文件的扩展名为 exe）。汇编语言源程序、汇编程序、目标程序、链接程序、执行文件之间的关系如图 1-1 所示。

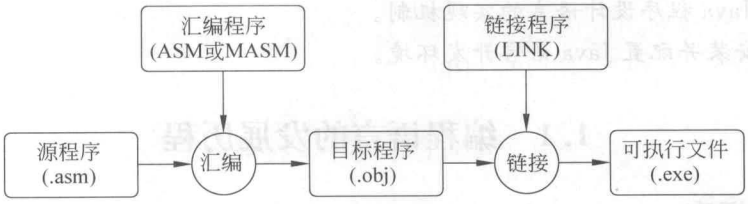

图 1-1 汇编语言程序的处理过程

如果分别使用机器语言和汇编语言实现相同的功能，汇编语言更容易理解。下面通过一个例子来说明汇编语言的编程特点。

【例 1-2】 在 8086 CPU 型的计算机中实现两个数据 100 和 256 相加功能。

汇编语言代码如下：

```
汇编语言代码                    代码解释
MOV AX, 100                 将 100 传送到 AX 寄存器
MOV BX, 256                 将 256 传送到 BX 寄存器
ADD BX, AX          将 AX 与 BX 中的值相加，并把结果存入 BX
```

显然，使用汇编语言编写程序要比机器语言更容易理解和掌握，也容易调试和维护，编写的源程序可读性较强。汇编语言不能被计算机直接理解和执行，需要进行转换。但是，汇编语言本质上还是机器语言，同样十分依赖于机器硬件，针对计算机特定硬件而编制，缺乏通用性，但运行效率仅次于机器语言。由于汇编语言跟硬件紧密相关，能准确发挥计算机硬件的功能和特长，因此目前大多数硬件设备的驱动程序都是使用汇编语言编写的。

1.1.3 高级语言

为了克服机器语言和汇编语言的缺陷，一种形式上接近于算术语言和自然语言，概念上接近于人们日常使用的自然语言且能为计算机所接受的语意确定、规则明确、自然直观和通用易学的计算语言就应运而生了，这就是高级语言。

高级语言主要是相对于汇编语言而言，它并不是像机器语言和汇编语言一样特指某一种具体的语言，而是包括了很多编程语言。

目前，高级语言的种类已经有上百种，但广泛应用的仅有十几种，它们有各自的特点和使用范围，如从20世纪50年代开始使用的Fortran语言多用于科学及工程计算；20世纪60年代的Cobol语言多用于商业事务处理和金融业；20世纪70年代的Pascal语言多用于结构化程序设计、C语言常用于软件开发；20世纪80年代的Prolog语言多用于人工智能方面；20世纪90年代起使用面向对象的C++语言多用于C/S结构的软件开发、Java语言和C#语言多用于网络环境的程序设计等。

2002年到2012年间，排名前三的语言始终是C语言、Java语言和C++语言，总份额占编程语言的50%左右，显示了三大主流语言在世界范围内的统治地位。其中，Java语言在Web服务器端的地位相当牢固，而C语言和C++语言则是基础软件和大量硬件设备研发的主流开发语言。

高级语言的发展也经历了从早期语言到结构化程序设计语言，从面向过程到非过程化程序设计语言的过程。从高级语言的发展角度对其分类，可以分为以下三类。

1. 面向过程的语言

面向过程的语言有Fortran、BASIC、Pascal、C语言等。在使用面向过程的语言进行编程时，编程人员必须用计算机能够理解的逻辑来描述需要解决的问题以及解决问题的具体方法和详细步骤，因此面向过程编程的思想就尤为重要。面向过程编程的核心思想就是通过自顶向下，逐层细化的方法进行功能分解，将一个大问题划分为几个小问题，再将几个小问题划分为更小的问题，以便能够解决问题。程序需要详细描述解题的过程和细节，每一步不仅要说明做什么，还要告诉计算机如何做。

2. 非过程化的语言

非过程化的语言又称为面向问题的语言。在使用非过程化的语言解决问题时，编程人员不必关心问题的求解算法和求解的过程，只需指出计算机要做什么，以及数据的输入和输出形式，就能得到所需要的结果。非过程化的语言的优点在于语言简洁、易学易用，因此已经成为关系数据库访问和操作数据的标准语言。

3. 面向对象的语言

面向对象的语言继承了面向过程的高级语言的结构化设计、模块化、并行处理等优点，并克服了数据与代码分离的缺点，将客观事物看做具有属性和行为的对象，通过抽象找出同一类对象的共同属性和行为，从而形成类，再通过语言中的对象和类直接模拟现实世界的事物。通过类的继承和多态可以很方便地实现代码重用，这样大大提高了程序的复用性和开发效率。常见的面向对象的语言有C++、C#、Java等。

1.2 Java语言简介

Java语言是由Sun Microsystems公司于1995年5月推出的新一代的程序设计语言，它是一种简单易用、安全可靠、跨平台、多线程的面向对象的程序设计语言，是目前编程领域主流的开发语言之一。Java语言无处不在，应用领域非常广泛，如桌面应用、嵌入式开发、移动应用开发、企业级应用等。

1.2.1 Java语言的起源

Java语言的前身是Oak语言，来自1991年Sun公司的一个名叫Green的项目，其最初

的目的是为开发一种能够在电视、冰箱等家用消费电子产品上进行交互式操作的分布式代码系统(Oak)。由于当时 Java 语言的应用对象只限于家用消费类的电子产品，并未被引起注意。直到 1993 年，全世界第一个 Internet 网页浏览器 Mosaic 的诞生为 Java 语言的发展提供了良好的契机。

当时 Internet 上的信息内容都是一些静态网页，不能与用户进行交互，人们迫切需要能够在浏览器端与用户进行交互的动态网页。Java 语言的主要贡献者 James Gosling 认为 Internet 与 Java 的特性不谋而合，于是便使用 Java 语言在 Internet 平台上编写出高交互性的网页程序，实现了其他程序设计语言所无法实现的如类似时钟、统计图等网页特效。因此，1993 年 Sun 公司将目标市场转向 Internet，针对网络的一些特性对 Java 进行了一系列的改进，融合了 C 和 C++ 等语言的优点，形成了跨平台及可靠性强的面向对象的程序设计语言。随着 Internet 的迅猛发展，1994 年 Green 项目组成员用 Java 编制了 HotJava 浏览器，使得它逐渐成为 Internet 上受欢迎的编程语言。1995 年，Oak 被正式命名为 Java 语言，Java 语言也就正式诞生了。

1.2.2 Java 语言的特点

Sun 公司的 Java 白皮书对 Java 语言的定义是：Java 语言是一种简单的、面向对象的、分布式的、解释执行的、健壮的、安全的、结构中立的、可移植的、高效的、多线程的、动态的语言。这个定义充分解释了 Java 语言的特点，有关 Java 语言特点的说明如表 1-1 所示。

表 1-1 Java 语言的特点

特 点	描 述
简单的	① 数据类型、数组、字符串、文件 I/O 等都封装成类，并采用包装(package)技术按树形层次封装类包 ② 风格类似于 C++，但摒弃了 C++ 中容易引发程序错误的地方，如指针操作、多重继承、运算符重载、宏、结构、共用体等 ③ 提供了丰富的类库
面向对象的	① 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 ② 提供类、接口和继承，只支持类之间的单继承，但支持接口之间的多继承，并支持类和接口之间的实现机制(关键字为 implements)
分布式的	① 提供一个支持 HTTP 和 FTP 等基于 TCP/IP 的子类库，包括 URL、URLConnection、Socket 等 ② Java 应用程序可通过统一资源定位器 URL 地址直接访问网络上的任何对象 ③ Java 的远程方法调用机制(RMI)也是开发分布式应用的重要手段
解释执行的	① 不同于 C++ 的编译执行，Java 程序经过编译形成字节码(.class 文件)，并非针对机型、操作系统的目标文件 ② Java 虚拟机负责执行字节码文件，它是解释执行的，是 Java 解决平台无关性的关键
健壮的	① 提供面向对象的异常(Exception)处理机制，如数组边界检测、检测异常出口、字节代码校验等 ② 实现了真数据，避免了覆盖数据的可能 ③ 提供自动内存垃圾收集功能，很好地解决了正确计算内存地址的问题，也省去了在编程时管理内存分配的额外工作量

续表

特 点	描 述
安全的	① 语言结构设计严谨，对象的方法和变量具有 public、protected、private 和友元不同的保护机制 ② 取消了像 C++ 中的指针和释放内存等功能，避免了非法内存操作 ③ 在编译时过行语法、语义的检查，在连接时再进行编译级的类型检查并消除间接对象访问，在运行时将进行字节码检验等 ④ 浏览器在运行. class 文件时，也要对其进行安全检验
结构中立的	① 基本数据类型的大小固定不变，如整型总是 32 位，长整型总是 64 位，消除了代码移植时数据类型大小不一致的问题 ② 将源程序编译成一种结构中立的中间文件格式(字节码)，与运行平台无关
可移植的	① 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 ② 通过定义独立于平台的基本数据类型及运算，Java 数据可以在任何硬件平台上保持一致
高效的	① 字节码可以在运行时被快速地翻译成运行该应用程序的特定 CPU 的机器码 ② 提供“即时编译”方式，即一次将字节码编译成本地代码，并将结果缓存起，在需要的时候再重新调用 ③ 可以监控代码被执行的“热度”，即将最常执行的字节码部分可以逐渐翻译成本地代码并小心地优化，能够极大地提高程序执行速度
多线程的	① 内置多线程功能，一个程序中可以同时创建多个线程来执行不同的工作，提供了更好的交互性和实时控制性 ② 提供了线程同步机制，该机制使不同线程在访问共享资源时能够相互配合，保证数据的一致性，避免出错
动态的	① 支持不断变化的运行环境 ② 允许程序动态地装入运行过程中所需要的类，可以在类库中自由地加入新的方法和实例变量，而不影响用户的程序执行

Java 语言是新一代面向对象的程序设计语言，由于它的硬件和软件平台的无关性的特点，已经逐步从一种单纯的高级程序设计语言发展为一种重要的 Internet 平台，并从传统的计算机应用向其他数字设备领域扩展，如移动电子商务、分布式计算技术、企业的综合信息化处理、嵌入式 Java 技术方面都得到广泛应用。

1.2.3 Java 语言实现机制

Java 语言是一门跨平台、高性能和健壮性的语言，只有弄清楚 Java 语言的实现机制才能更好地掌握该语言的精髓。Java 语言实现机制主要由 Java 虚拟机、垃圾回收机制和安全检查机制三部分组成。

1. Java 虚拟机

Java 虚拟机(Java Virtual Machine，JVM)是用软件模拟各种计算机功能的虚拟计算机，是一种用于计算机设备的规范。它定义了指令集、寄存器集、类文件结构栈、垃圾收集堆、内存区域等组件，提供了跨平台能力的基础框架。JVM 在 Java 技术体系(如图 1-2 所示)中处于核心地位，是程序与底层操作系统和硬件无关的关键，从而实现了 Java 的平台无关性。