

高职高专计算机系列教材

程序设计基础 (Java版)

李占波 姬莉霞 王海玲 欧 研 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高职高专计算机系列教材

程序设计基础 (Java 版)

李占波 姬莉霞 王海玲 欧 研 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书以面向对象程序设计语言 Java 为载体,从程序设计的基本理念和基本方法入手,运用大量实例,由浅入深地讲述了程序设计的基本过程和需要掌握的基础知识。全书共有九章,内容涉及程序设计的基本过程、数据运算、逻辑思维、面向对象程序设计思想、简单的数据结构和算法设计思想等,并全程贯穿了软件工程理念。

全书内容丰富,结构清晰,所有例题均使用 JDK 6 上机调试通过。本书可作为普通高等院校、高职高专、成人高等院校计算机及其相关专业的基础教材,也可作为广大程序设计爱好者的入门参考书。

图书在版编目(CIP)数据

程序设计基础:Java 版/李占波等编著. —北京:中国铁道出版社,2007.8

(高职高专计算机系列教材)

ISBN 978-7-113-08068-6

I. 程… II. 李… 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(2007)第 122200 号

书 名:程序设计基础(Java 版)

作 者:李占波 姬莉霞 王海玲 欧 研

出版发行:中国铁道出版社(100054,北京市宣武区右安门西街8号)

策划编辑:严晓舟 秦绪好

责任编辑:陈 宏

特邀编辑:李振华

封面设计:高 洋

封面制作:白 雪

责任校对:包 宁

印 刷:三河市宏达印刷有限公司

开 本:787×1092 1/16 印张:14.5 字数:335 千

版 本:2007 年 8 月第 1 版 2007 年 8 月第 1 次印刷

印 数:1~5 000 册

书 号:ISBN 978-7-113-08068-6/TP·2426

定 价:20.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签,无标签者不得销售

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

高职高专计算机系列教材

编 委 会

主 任：汪燮华

副主任：陶 霖 陆 虹

编 委：（以姓氏音序排列）

常桂兰	陈志毅	崔俊杰	韩田君
矫桂娥	李 斌	刘鸿基	刘 敏
刘 燕	刘中原	陆惠茜	聂青林
秦 川	王淑英	王 晴	吴慧萍
熊发涯	徐方勤	赵俊兰	周天亮

前言

程序设计类课程是计算机及其相关专业的重要专业基础课,“程序设计基础”是基础中的基础,打好了这个基础,任何一门程序设计课都可以触类旁通。本书以 Java 语言为例讲述程序设计的基础知识。内容主要包括程序设计基础知识和 Java 语言基础,运用 Java 语言阐述面向对象的程序设计基础理论、主要原则和思维方法,并使用易于理解的程序实例帮助读者较全面和深入地建立起程序设计,特别是面向对象的程序设计的理念。

Java 是 Sun 公司开发的新一代编程语言,它随着互联网的发展而被推广应用,是目前国内外广泛使用的程序设计语言。经过几年的发展,Java 已经真正成为严格、主流的开发语言,如今它不仅仅是一门语言,而是已经发展成为一门技术,包括 Java 芯片技术、Java 编译技术、Java 数据库连接技术、基于 Java 的信息家电的联网技术、企业信息服务综合求解方案技术等。Java 语言是面向对象技术成功应用在程序设计语言中的典范。它具有平台无关性、安全机制、高可靠性和内嵌的网络支持等特点。Java 语言的这些特点,加上 Java 提供的丰富的类库,使得 Java 语言自身不断的发展,成为当前应用程序,特别是网络应用程序编写的首选工具之一。

本书从目前高校的实际教学情况出发,以 Java 语言为载体,着重介绍程序设计的基本理念和基础方法,全书贯穿了软件工程的思想和方法。本书可以作为开发类专业的程序开发语言的入门书籍来学习,使学生掌握程序设计的基本方法和思想,初步体验开发程序的乐趣,激发学生对程序设计的兴趣,使其养成良好的编程习惯。本书亦可作为非开发专业的基础语言书籍,培养学生基本的程序设计思想,了解一门基本的开发语言。

全书共有九章,分别为“程序设计基础”、“数据运算”、“程序逻辑”、“面向对象的程序设计”、“简单数据结构”、Java Applet、“图形用户界面”、“异常处理”和“多线程”。第 1 章“程序设计基础”介绍计算机的系统组成和计算机程序,并以 Java 程序为例,让学生从直观上感受最简单的 Java 应用程序和 Java 小程序。第 2 章“数据运算”和第 3 章“程序逻辑”是各类语言所共有的知识,旨在使学生掌握最基本的程序设计理念。目前,面向对象方法已逐渐取代面向过程方法成为程序设计领域的主流技术。第 4 章“面向对象的程序设计”借助 Java 语言使学生系统了解面向对象的初步知识。第 5 章“简单数据结构”介绍了什么是数据结构,讲解了数组和向量的使用,并对典型的排序和查找算法进行了介绍,使学生对数据结构和算法设计有初步的掌握。第 6 章 Java Applet 仅对 Java 小程序进行了简单的介绍。第 7 章“用户界面设计”以 Java AWT 为基础,介绍了典型界面设计中的各个组件以及界面布局和一些的响应事件。第 8 章“异常处理”和第 9 章“多线程”,是对程序设计基础的适当延伸,可供学有余力的学生作为了解知识,同时作为深入学习的简单引导。程序设计是一门实践性很强的课程,本书提供了大量的实例,对每个重要知识点都用实例来说明,并在每章配备了实验内容以及大量的习题。另外,本书仅以 Java 作为载体,注重程序设计语言的共性,并不

过多关注 Java 语言本身的细节。

本书由李占波、姬莉霞、王海玲、欧研编著，并调试验证了书中的所有实例。本书可以作为开发类专业的程序设计入门教材，或非开发类专业的基础语言教材，也可作为广大程序设计爱好者的入门参考书。

由于时间仓促，编者水平有限，本书难免存在疏漏之处，敬请广大读者批评指正，以便本书再版时进行改进和完善。

编 者

2007 年 6 月

目 录

第 1 章 程序设计基础	1
1.1 计算机系统概述	1
1.1.1 计算机系统的组成	1
1.1.2 计算机软件概述	2
1.2 计算机程序介绍	2
1.2.1 计算机程序概述	2
1.2.2 程序执行阶段	3
1.2.3 机器语言和高级语言	3
1.2.4 程序逻辑的实现	4
1.3 初识 Java 程序	6
1.3.1 Java 语言的发展和特点	6
1.3.2 JDK 的获得与安装	7
1.3.3 一切都是对象（面向对象初步）	10
1.3.4 第一个 Java 应用程序（Java Application）	12
1.3.5 第一个 Java 小程序（Java Applet）	15
1.3.6 Java 程序的结构	18
本章小结	19
实验 1 熟悉 Java 语言环境	19
习题 1	20
第 2 章 数据运算	22
2.1 标识符、保留字和分隔符	22
2.1.1 标识符	22
2.1.2 保留字	23
2.1.3 分隔符	23
2.2 数据类型	23
2.2.1 数据类型概述	24
2.2.2 常量	24
2.2.3 变量	26
2.3 运算符	29
2.3.1 算术运算符	29
2.3.2 关系运算符	32
2.3.3 布尔运算符	32
2.3.4 赋值运算符	34
2.3.5 位运算符	34
2.3.6 条件运算符	35
2.3.7 字符串运算符	36

2.3.8 类型转换	36
2.3.9 优先级和结合性	38
2.4 输入/输出初步	39
2.4.1 输出	39
2.4.2 输入	39
本章小结	43
实验 2 Java 语言基本数据类型以及运算的使用	43
习题 2	44
第 3 章 程序逻辑	47
3.1 程序结构的基本类型	47
3.2 顺序结构	48
3.2.1 语句	48
3.2.2 块	48
3.3 分支结构	49
3.3.1 if...else 语句	50
3.3.2 分支嵌套	51
3.3.3 switch 语句	54
3.4 循环结构	58
3.4.1 while 语句	58
3.4.2 do...while 语句	60
3.4.3 for 语句	61
3.4.4 循环嵌套	65
3.5 其他流程控制语句	68
3.5.1 标号	68
3.5.2 break 语句	68
3.5.3 continue 语句	70
3.6 方法	72
3.6.1 方法的定义	73
3.6.2 方法的调用	75
3.6.3 方法的重载	76
3.6.4 方法的嵌套和递归	78
3.6.5 方法和变量的作用域	80
本章小结	85
实验 3 程序逻辑	85
习题 3	86
第 4 章 面向对象的程序设计	90
4.1 类	90
4.1.1 类的定义	90
4.1.2 对象	92

4.1.3	构造方法	94
4.1.4	类的成员和修饰符	95
4.2	类的继承	99
4.2.1	类继承的实现	99
4.2.2	抽象类和抽象方法	102
4.3	接口	103
4.3.1	接口的定义	103
4.3.2	接口的实现	104
4.3.3	接口的继承	104
4.3.4	接口的多态	106
4.4	包	107
4.4.1	JDK 中的常用包	107
4.4.2	引用 Java 定义的包	108
4.4.3	自定义包	108
	本章小结	110
	实验 4 面向对象程序设计	110
	习题 4	111
第 5 章	简单数据结构	113
5.1	数据结构描述	113
5.1.1	基本概念和术语	113
5.1.2	算法	114
5.2	数组	114
5.2.1	一维数组	114
5.2.2	数组的基本操作	115
5.2.3	多维数组	116
5.3	数组的使用	118
5.3.1	数组作为方法的参数	118
5.3.2	数组操作的常用方法	119
5.4	排序	122
5.4.1	选择排序	122
5.4.2	冒泡排序	125
5.4.3	插入排序	126
5.5	查找	127
5.5.1	顺序查找	127
5.5.2	二分查找	128
5.6	向量类	130
	本章小结	132
	实验 5 数组的运用	132
	习题 5	133

第 6 章	Java Applet	135
6.1	Applet 概述	135
6.2	HTML 简介	135
6.2.1	HTML	135
6.2.2	在 HTML 文件中嵌入 Applet	137
6.3	Java Applet 概述	139
6.3.1	Java Applet 示例	140
6.3.2	Applet 的主要方法及生命周期	144
6.3.3	HTML 与 Applet 的参数传递	147
6.3.4	Applet 和 Application	148
6.3.5	Applet 的应用	150
	本章小结	150
	实验 6 Java Applet 小试牛刀	151
	习题 6	151
第 7 章	图形用户界面	153
7.1	概述	153
7.2	文本与字体	154
7.2.1	绘制字符串、字符和字节	154
7.2.2	字体、字型和字号	155
7.2.3	颜色的设置	157
7.3	图形设计	159
7.3.1	画直线	159
7.3.2	画矩形	160
7.3.3	画椭圆和圆弧	162
7.3.4	绘制多边形	164
7.4	显示图像	165
7.5	标签、按钮和文本框	167
7.5.1	标签	167
7.5.2	按钮	168
7.5.3	文本框	170
7.6	复选框、单选按钮和列表	174
7.6.1	复选框	175
7.6.2	单选按钮	177
7.6.3	列表	179
7.7	鼠标事件与键盘事件	180
7.7.1	事件处理模式	181
7.7.2	鼠标事件	184
7.7.3	键盘事件	186

7.8 布局管理器	188
7.8.1 FlowLayout 布局管理器	188
7.8.2 BorderLayout 布局管理器	190
7.8.3 GridLayout 布局管理器	191
本章小结	193
实验 7 用户界面初步设计	194
习题 7	194
第 8 章 异常处理	196
8.1 异常处理基本知识	196
8.1.1 异常的概念	196
8.1.2 Java 的异常处理机制	197
8.1.3 常见异常	197
8.2 异常处理方法	198
8.2.1 try...catch...finally 语句	198
8.2.2 throw 语句和 throws 语句	202
8.2.3 自定义异常类	203
本章小结	205
实验 8 异常处理	205
习题 8	206
第 9 章 多线程	208
9.1 概述	208
9.2 线程的创建	208
9.2.1 通过继承 Thread 类来创建线程	209
9.2.2 通过实现 Runnable 接口来创建线程	211
9.2.3 多线程的 Applet	212
9.3 线程的状态：生命周期	213
9.4 多线程的同步	214
本章小结	217
实验 9 多线程	217
习题 9	218
参考文献	220

第 1 章 程序设计基础

学习目标

- 了解计算机系统的组成
- 了解计算机软件和程序的基本概念
- 会使用流程图表示程序逻辑
- 了解 Java 语言的发展和特点
- 能够获得、安装和使用 JDK
- 能够编译、解释和运行第一个简单 Java 程序
- 能够大致了解 Java 程序的结构
- 对面向对象有初步的知识
- Java 程序的编辑、解释和运行

1.1 计算机系统概述

计算机是一种自动地、高速地进行数值运算和信息处理的电子设备。它主要由一些机械的、电子的器件组成，再配以适当的软件用以解决某些实际问题。

从对计算机的描述中可以看出，计算机不仅仅是物理上的一个实体，更是一个整体概念，是一个系统。

1.1.1 计算机系统的组成

计算机系统由硬件和软件两大部分组成。硬件是构成计算机的设备，它是计算机系统的物质基础；软件则提供了能够使计算机工作的操作平台、使用界面与应用功能等。

计算机的硬件系统由运算器、控制器、存储器、输入/输出（I/O）接口、输入/输出设备和电源系统组成。其中，运算器和控制器合称为中央处理器（CPU）。CPU、存储器、I/O 接口和电源系统等组成了计算机系统的“主机”，输入/输出设备则被称为外部设备。

计算机软件包括系统软件和应用软件，整个计算机系统的组成如图 1-1 所示。

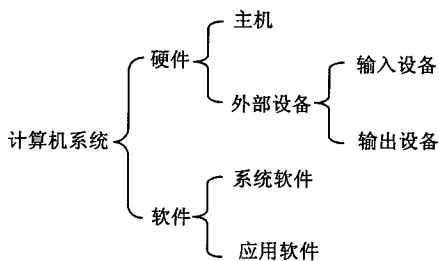


图 1-1 计算机系统的组成

1.1.2 计算机软件概述

计算机软件指能使计算机工作的程序和程序运行时所需要的数据, 以及与这些程序和数据有关的文档。

软件=程序+数据+文档

从广义上讲, 程序就是软件, 在 1.2 节中将着重介绍计算机程序; 数据指计算机要处理和生成的各种信息资料; 文档是对程序和数据进行说明性的相关文本和图表资料。

1.2 计算机程序介绍

计算机具有快速而精确地完成数学计算和信息处理的能力。要完成这些操作, 计算机需要运行专门为某个任务编写的计算机程序。

1.2.1 计算机程序概述

计算机程序是一组精确地告诉计算机执行什么操作和什么时候执行操作的连续指令集。它们是为了完成某种任务而编写的, 由于各项任务的复杂度不同, 计算机程序的大小也各不相同。

计算机程序可分为三种类型。

1. 应用程序

应用程序是用户利用计算机软、硬件资源为解决各类应用问题而编写的程序。应用程序的存在与否并不影响整个计算机系统的运行, 但它必须在系统软件的支持下才能工作。例如, WPS、Word、Excel 等。

2. 操作系统

操作系统是系统软件的核心。操作系统 (Operating System, OS) 是由指挥与管理计算机系统运行的程序模块和数据结构组成的一种大型软件系统, 其功能是管理计算机的全部硬件资源和软件资源, 为用户提供高效、周到的服务平台。例如, IBM-PC 及其兼容机的运行要有 PC-DOS 或 Windows 的支持。

没有配备任何软件的硬件计算机称为裸机。裸机向外部世界提供的界面只是机器指令。用户及其他程序要通过操作系统来使用计算机。

3. 设备驱动程序

操作系统需要通过设备驱动程序来与硬件设备通信。设备驱动程序是帮助操作系统与计算机中的硬件组件进行通信的程序。设备驱动程序用来管理硬件设备, 如打印机、显示器、网卡等。设备驱动程序与硬件和操作系统直接交互。

应用程序、操作系统和设备驱动程序的关系如图 1-2 所示。另外, 通过一个例子来说明: 当用应用程序 Realone 打开一个存储在“F:\电影”目录下的文件“功夫.rm”时, Realone 向操作系统发出一个命令, 要求从“F:\电影”所在的硬盘位置得到该文件, 随后操作系统通过设备驱动程序与硬盘通信, 以访问该文件。

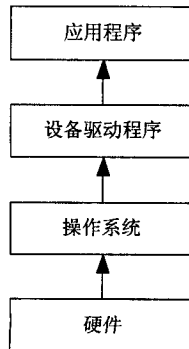


图 1-2 3 种计算机程序的关系

1.2.2 程序执行阶段

程序执行过程从功能上可以分为输入阶段、处理阶段和输出阶段。

- 在输入阶段，需要通过输入设备向计算机提供字符、声音、图片和动画等数据。
- 在处理阶段，计算机在输入阶段提供的数据上执行操作，通过中央处理器进行计算、统计、文本编辑、逻辑判断、图形缩放和色彩配置等数据处理。
- 在输出阶段，计算机通过输出设备以文本、声音、图片或各种控制信号的形式输出处理结果。

如火车站订票系统，执行阶段如图 1-3 所示。在程序执行的输入阶段，需要输入“日期”、“车次”、“目的地”和“座位要求”信息。在程序执行的处理阶段，计算机接受用户输入的信息，然后根据这些数据进行查询处理。在程序执行的输出阶段，显示所有满足要求的车票信息，并可根据需要打印车票。

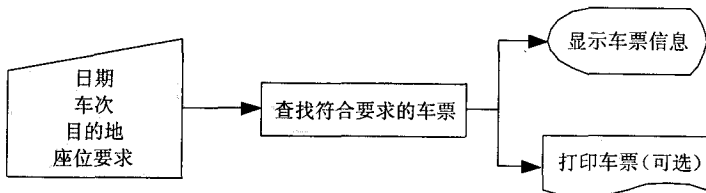


图 1-3 订票系统的执行过程

1.2.3 机器语言和高级语言

人类在相互交谈时使用的是相互都能理解的语言，如汉语、英语、俄语等，它们统称为自然语言。那么计算机又是如何理解人类所编写的程序呢？它又能识别哪些语言呢？

1. 机器语言

计算机每做的一次动作、一个步骤，都是按照有序的指令集合来执行的，每个计算机指令都对应一个计算机应该执行的物理操作，这些操作是用 0 和 1 序列表示的，它是计算机的内部语言，又称为机器语言。例如，某种计算机包含如下的指令集，用“00000011”表示数学运算符“+”号，用“00001000”表示数学符号“*”，等等。

2. 汇编语言

机器语言是面向机器的，即某种机器语言只能在某特定型号的计算机上使用，用机器语言编程烦琐，容易出错，且难以记忆和识别。为了克服机器语言的缺点，人们采用助记符来表示机器指令。这种采用助记符进行程序设计的语言称为汇编语言。用汇编语言编写的程序称为汇编源程序。

3. 高级语言

汇编语言是一种符号语言，比机器语言容易理解和掌握，也容易调试和维护。但是汇编语言仍然是面向机器的语言，即使是一个简单任务也需要多条指令才能完成。掌握和使用这种语言比较困难，因此，人们发明了高级语言。它接近人类的自然语言，比机器语言和汇编语言都易学、易用和易于理解。可以使用的高级编程语言很多，比如 C/C++、BASIC、Pascal、Java 等，每种语言都有其独特的优势。

计算机所能识别的语言只有机器语言，因此人们要控制计算机就要通过计算机语言向

计算机发出命令。要想使计算机理解高级语言，就必须将高级语言“翻译”成机器语言。高级语言按转换方式可以分为两类。

（1）解释类：应用程序源代码一边由相应语言的解释器“翻译”成目标代码，一边执行。效率比较低，而且不能生成可独立执行的可执行文件，应用程序不能脱离其解释器，但这种方式比较灵活，可以动态地调整、修改应用程序。

（2）编译类：编译指在应用源程序执行之前，就将程序源代码“翻译”成目标代码，因此其目标程序可以脱离其语言环境独立执行。现在大多数的编程语言都是编译型的。

4. 程序的执行

要使程序执行，就必须将程序解释或者编译成目标代码，然后由计算机根据有序指令集执行，如图 1-4 所示。

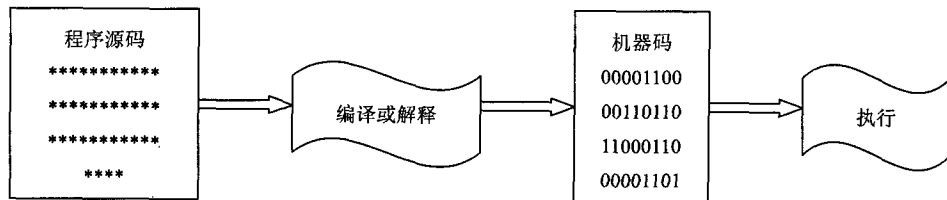


图 1-4 程序的执行示意图

1.2.4 程序逻辑的实现

所有的程序都是为了解决现实生活中的问题而开发的，因此在开始编写程序之前，必须根据一系列的逻辑步骤来分析要解决的问题。这些步骤包括：

- （1）识别输出和输入要求。
- （2）确定相关的处理方法。
- （3）实现逻辑。

用流程图或者伪代码表示出程序逻辑，本书讨论以流程图方式表示程序逻辑。

流程图用一系列符号来表示程序的逻辑，它包含有一个符号集，每个符号表示指定类型的操作。这些符号通常包括接收输入、处理、显示输出和判断等。

如表 1-1 所示描述了一些常用的流程图符号。

表 1-1 常用流程图符号

符 号	活 动
	输入
	处理
	输出（显示）
	输出（文档）
	判断（决策）
	流程线：连接流程图的各步骤，并指示步骤的执行顺序
	起止符：指示流程图的开始或结束

【例 1-1】用程序逻辑表示工人的月薪。

某工厂工资以计件形式发放，每月月末根据工人生产的产品个数来计算其月薪：月薪=每生产一个产品的酬金×该月生产产品的个数，求工人的月薪。

该问题可以用以下步骤来规划解决方案。

(1) 识别输出和输入元素。

输出：

- 工人编号
- 月薪

输入：

- 工人编号
- 产品个数

(2) 确定相关的处理。

- 接收工人编号
- 接收产品个数
- 查看产品个数
- 计算得出月薪
- 显示工人编号和月薪

(3) 实现逻辑（画流程图），流程图如图 1-5 所示。

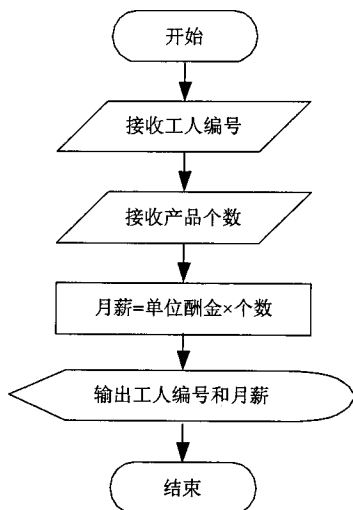


图 1-5 工人月薪流程图

【例 1-2】用程序逻辑表示应聘者的筛选。

某软件公司欲招聘一位资深系统分析员，很多人前来应聘。经过笔试，需要向所有笔试成绩在 60 分以下的应聘者发出拒绝信，同时通知有资格者前来面试。可以用以下步骤来规划解决方案。

(1) 识别输出和输入元素。

输出：

- 应聘者编号、姓名
- 发送信件类型

输入：

- 应聘者编号、姓名
- 面试分数

(2) 确定相关的处理。

- 接收应聘者编号、姓名和分数
- 查看应聘者分数
- 根据分数判断是否合格
- 显示信件类型

(3) 实现逻辑（画流程图），流程图如图 1-6 所示。

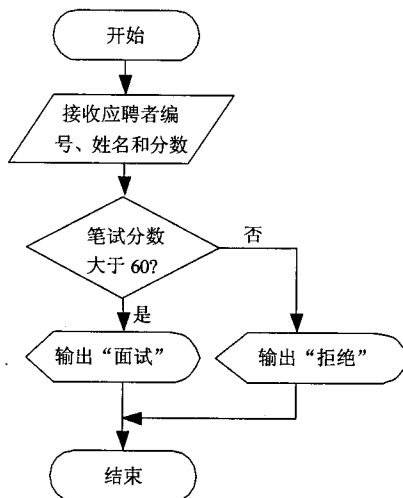


图 1-6 应聘筛选流程图

1.3 初识 Java 程序

美国硅谷有一句行话，每 10~15 年有一次轮回。最近的一次轮回就是从 Java 开始。Java 是由 Sun 公司开发的新一代编程语言。使用它可在各式各样不同种机器、不同种操作平台的网络环境中开发软件。Java 正在逐步成为 Internet 应用的主要开发语言。它彻底改变了应用软件的开发模式，带来了自 PC 以来又一次技术革命，为迅速发展的信息世界增添了新的活力。

1.3.1 Java 语言的发展和特点

Java 语言的设计可以追溯到 1991 年，美国 Sun 公司为了开发消费电子产品市场，成立了由 Jame Gosling、Bill Joe 等人组成的 Green 开发小组。他们为在电视、控制烤箱等家用消费类电子产品上进行交互式操作而开发了一种与平台无关、可靠性强、小而灵活的名为 Oak（橡树）的编程语言，但当时并没有引起人们的注意。

直到 1994 年下半年，随着 Internet 的迅猛发展和环球信息网 WWW 的快速增长，Sun 公司把 Oak 的技术应用于网络，并正式命名为 Java。1995 年，Sun 公司正式发布了 Java。Java 语言在网络上的独特优势使其逐渐成为 Internet 上受欢迎的程序设计语言。

Java 语言的广泛应用是因为它在网络程序设计方面有传统程序设计语言难以比拟的特点。首先，作为一种程序设计语言，它简单，面向对象，不依赖于机器的结构，具有可移植性、分布性和安全性，并且提供了并发的机制，具有很高的性能。其次，它最大限度地利用了网络，Java 的小程序（Java Applet）可以在网络上传输而不受机器和操作系统的限制。Java 语言的主要特点有以下几方面。

1. 简单性

Java 语言最初是对家用电器进行交互式控制而设计的一种语言，因此它必须简单明了。Java 语言是一种面向对象语言，它通过提供最基本的方法来完成指定任务。Java 语言略去了 C++ 运算符重载、多重继承、指针运算等一系列内容，更利于初学者学习。另外，Java 通过实现自动垃圾收集，大大简化了设计者的内存管理工作，且运行时占用很少的内存资源。

2. 平台无关性

Java 源程序经过编译器编译，会被转换成一种“字节码”的中间码，最后通过运行操作系统平台环境相对应的一种称为 Java 解释器的运行机构来执行字节码。字节码的最大特点便是可以跨平台运行，即“编写一次，到处运行”，如图 1-7 所示，正是这一特点使得 Java 得到迅速的普及。

3. 分布性和安全性

Java 从诞生起就与网络联系在一起，它强调网络特性，内置 TCP/IP、HTTP、FTP 协议类库，便于开发网上应用系统，Java 2 平台采用了域管理方式的安全模型，无论是本地代码还是远程代码都可以通过配置的策略，设定可访问的资源域。可以说，目前在 Internet 上没有比 Java 更安全的技术了。

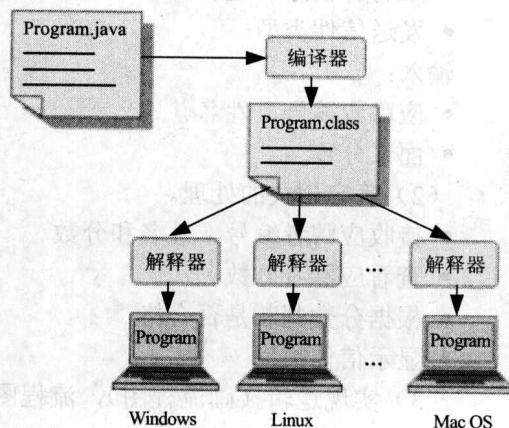


图 1-7 Java 的平台无关性