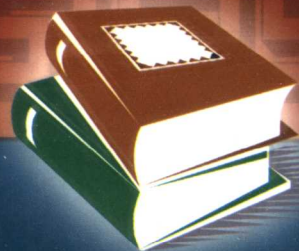


高等学校教育改革推荐教材



Java 程序设计

房晓溪 主编
郝文化 主审



中国电力出版社
www.infopower.com.cn

高等学校教育改革推荐教材

Java

程序设计

房晓溪
郝文化

主
主

江苏工业学院图书馆
藏书章



中国电力出版社

www.infopower.com.cn

内容提要

本书是依据本科和高职高专院校的培养目标及基本要求,结合作者多年来的教学经验和工程实践基础,为实施教学改革,使计算机教学进一步面向软件编程实践而编写的一本教材。包括的主要内容有:面向对象程序设计语言,Java 程序设计入门,程序控制流程,Java 的方法、类与对象、继承与派生,Java 的异常处理,Java 包的组成和使用,设计 Applet 程序,Java 的多线程、I/O 流、网络程序设计等。每章后都配有上机实战和理论巩固题,实现教与学的统一。

本书语言通俗易懂,内容丰富翔实,突出了以实用为中心的特点。使用本书进行学习,可帮助读者用最少的_{时间}掌握最多的知识及工作经验与技巧,是本科和高职高专院校理想的教学教材,同时也可作为软件和技术工程人员的参考用书。

图书在版编目(CIP)数据

Java 程序设计 / 房晓溪主编. —北京:中国电力出版社, 2005

高等学校教育改革推荐教材

ISBN 7-5083-2957-0

I.J... II.房... 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 127872 号

丛 书 名: 高等学校教育改革推荐教材

书 名: Java 程序设计

出版发行: 中国电力出版社

地 址: 北京市三里河路 6 号

邮政编码: 100044

电 话: (010) 68358031

传 真: (010) 68316497, 88383619

本书如有印装质量问题,我社负责退换

服务电话: (010) 88515918

传 真: (010) 88518169

E-mail: infopower@cepp.com.cn

印 刷: 北京丰源印刷厂

开本尺寸: 185×260

印 张: 14.75

字 数: 343 千字

书 号: ISBN 7-5083-2957-0

版 次: 2005 年 2 月北京第 1 版

印 次: 2005 年 2 月第 1 次印刷

印 数: 0001—5000 册

定 价: 21.00 元

版权所有,翻印必究

序

党的十六大以来,举国上下都在贯彻人才强国战略。特别是在我国加入 WTO、面临经济转轨的形势下,我国高等教育事业紧扣世界教育发展的脉搏,已从精英教育走向大众教育,进入迅猛发展的时期。

21 世纪是信息技术高度发展的信息时代,要求大学生具有更加丰富的信息技术知识和更强的应用信息技术的能力。选好一本教材,对提高计算机类专业的教学质量至关重要。在确定课程体系以后,最重要的工作就是根据教学要求编写出高质量的教材。在教材建设中,我们提倡百花齐放、推陈出新,经过实践考验,形成公众认可的精品,从而以推动教学质量的提高。

受出版社委托,我们邀请了一些相关高校的教师,召开了一次别具风格的“新形势下高等院校系列教材建设研讨会”,共同研究了国内外高等教育的教学现状与发展趋势,介绍了当前国内高等院校计算机类专业的教学状况与教材建设情况,探讨了新形势下高等院校的教材建设问题,强调了教材要“体现实用性,突出院校自身教学特点,老师易教,学生易学”的编写原则。

在有关专家、教授的亲切指导和热心支持下,在相关高校计算机学院(系)院长(主任)与骨干教师的热情参与下,教材编委会和众多作者在总结了教材建设上多年探索和实践的基础上,广泛汲取了各类成功教材的有益经验,分析了广大学生的承受能力和学习心得,并听取了计算机界教育专家们建议,博采各家所长,精心打造本套教材。本套教材以培养学生的应用能力为目的,突出实用性,突破了传统教材中理论与实践脱节、偏深、偏难的现象,易于实施教学,发掘学生的潜在学习积极性,能让学生在最短的时间内,全面系统地掌握计算机技能。

本套教材作为“新形势下高等院校系列教材”,突出体现了以下特色:

(1) 各高校领导十分重视,热心于本套教材的建设,并鼓励教师积极参与,热切希望本套教材真实地反映各高校教学水平、教学特点和最新教研成果;

(2) 教材的整个编写过程,自始至终得到有关专家和教授热心、真诚的指导与关怀;

(3) 根据各高校与老师的实际需要,与出版社共同规划,共同建设、共同完善教材体系与内容,体现开放与互动交流的宗旨。

这套教材在写法上体现了理论与实践相结合,相关的知识点讲解清晰、透彻,注重教学实际,力求科学实用,符合教学习惯,语言通俗易懂,内容丰富翔实,既注重基本理论及使用方法的深入剖析,又注重实例与技巧的融会贯通。各章后附有课后理论与上机练习,满足学生需要,亦为教师的课堂教学及上机指导提供了有益的参考与帮助。

这套教材是“新形势下高等院校系列教材”,是一种新的尝试。“新”,就会有許多值得修改的地方。本套教材面向各高校,对有志于参与本套教材编写或修改的教师来说是开放的,各学校可以根据自己的特点和教师的特长加以修订和补充。我们热烈欢迎更广泛的学校、教师或作者共同热心参与,更好地规划和完善这套“新形势下高等院校系列教材”。

前 言

近几年,面向对象的程序设计已成为程序开发设计的主流思想。这种思想主要是通过增加软件的重用性和扩充性提高程序员的开发设计能力。目前,面向对象的程序设计中应用最广泛的是 Java 语言,它的推广对程序开发技术产生了深远的影响。Java 语言不仅继承了 C++ 的优良传统,而且对其不足之处进行了改良,从而实现了最大限度地利用网络。Java 语言所具有丰富的类库等诸多特性使其成为最广泛使用的网络程序编程语言。

随着 Java 语言的飞速发展,这就使得 Java 语言程序设计图书在市场上不断推陈出新。本科和高职高专院校计算机专业以及软件开发人员对它提出了更高、更为迫切的要求,在此背景下,笔者们经过认真收集和整理素材,精心编写了《Java 程序设计》一书。本书与 Java 语言最新版本同步,由具有多年从事 Java 语言教学工作的教师共同编写完成,具有很强的针对性、实用性和可读性。

《Java 程序设计》从 Java 语言的类和对象的基本概念和简单操作出发,紧紧围绕程序实例,介绍了如何利用 Java 语言编写高效应用程序的方法,并对常用的 Java 语言地特性做了深入透彻地解析。

主要内容

本书全面细致地介绍了使用 Java 语言实现网络程序编程的相关知识。全书共分为 13 章。具体的内容安排为:第 1 章 Java 程序设计入门,主要介绍了 Jvav 程序设计的入门知识,如 Java 语言简介、Java 语言的特性、Java 特点等;第 2 章 Java 语法规则,主要介绍了 Java 的符号集、Java 的数据类型以及 Java 运算符的使用;第 3 章程序控制语句,主要介绍了程序的三基本结构和循环控制结构;第 4 章 Java 语法规则,主要介绍了 Java 方法的概念、Java 方法的定义和调用、Java 的递归方法;第 5 章 Java 的类,主要介绍了类和对象的概念、类的组成及其运作规则、Java 应用程序的结构;第 6 章继承与多态,主要介绍了继承和多态的概念、类成员的继承以及接口等相关知识;第 7 章 Java 的异常处理,主要介绍了例外、错误与事件例外类的层次、例外处理语句及使用;第 8 章 Java 包的组成和使用,主要介绍了 Java 的包、Java 语言类库的结构、java.lang 包中的常用类、关于 Java 的技术文档;第 9 章 Java 的集合类,主要介绍了集合类概述、集合类、新集合类;第 10 章 Java 的 I/O 流,主要介绍了流的层次结构、InputStream 流和 OnputStream 流、内存读写、管道流、文件输入/输出流、连接多个流、过滤流、PrintStream 流;第 11 章 Applet 基础和多媒体技术,主要介绍了 Applet 的运行平台浏览器、Applet 程序的运行机理、Applet 程序的结构、Applet 程序设计、Applet 的安全机制、多媒体设计类的继承关系、对象的属性、设置字体、设置颜色、Graphics 类、图像处理、动画技术;第 12 章多线程处理,主要介绍了线程的基本概念、线程的基本结构与使用方法、线程的管理、动画线程的制作;第 13 章网络程序设计,主要介绍了网络常识、网络程序设计、服务器和客户机、网络程序设计技术、Java 支持的网络程序设计、移动网络、Java 安全性。

特点

本书用模块化方式深入浅出地讲解了 Java 语言的基本知识点、重点和难点。全书重点突

出、主次分明、结构清晰、逻辑性强，每章都有知识点、概述、小结、实践检验等配套内容，使读者能够在充分掌握 Java 程序设计的知识和技巧的同时，将其尽快运用到实际工作中，从而实现教与学的结合、统一。

同时，为了便于多媒体教学，我们为读者提供本书配套的电子教案，为老师教学提供有益的参考和帮助。该服务请登录网址：<http://www.bojia.net>，可在网站下载专区免费下载。

适应对象

本书通俗易懂，内容丰富翔实，突出以实例为中心的特点，既可作为大学本科和高职高专院校计算机专业的教学用书，也可作为从事 Java 应用程序开发的软件网络工程师及软件、电讯技术工程人员的参考用书。

编写分工

本书由房晓溪担任全书的主要编写工作，岳溪、陈念伟、房方和熊波也担任部分编写工作，郝文化负责了全书的审订工作。同时，参与本书编排的还有：邹素琼、王安贵、陈郭宜、程小英、谭小丽、卢丽娟、刘育志、吴淬砺、赵明星、贺洪俊、李小平、史利、张燕秋、周林英、黄茂英、李力、李小琼、李修华、田茂敏、苏萍、巫文斌、邹勤、栗德睿、童芳、李中全、蒋敏、刘华菊、袁媛、李建康、袁涛、杨春华、贾小蓉、张永龙等，在此表示衷心感谢。

配套服务

为充分展现本书编写特点，帮助读者深刻理解本书编写意图与内涵，进一步提高对本书教学的使用效率，我们建立本书使用指导联络方式，它是读者与编者之间交流沟通桥梁。欢迎读者将图书使用过程中的问题与各种探讨、建议反馈与我们，本书作者竭诚为您服务，联系方式 E-mail: bojia@bojia.net。

作 者

2004 年 11 月

目 录

序	
前 言	
第 1 章 Java 程序设计入门	1
1.1 从程序设计语言说起	1
1.2 Java 语言的特性	3
1.3 Java 的特点	7
1.4 本章小结	8
1.5 实践检验	8
第 2 章 Java 语法规则	9
2.1 Java 的符号集	9
2.2 Java 的数据类型	11
2.3 使用 Java 运算符	15
2.4 本章小结	19
2.5 实践检验	19
第 3 章 程序控制语句	21
3.1 选择（条件）结构	21
3.2 循环控制结构	23
3.3 本章小结	26
3.4 实践检验	26
第 4 章 Java 语法规则	30
4.1 Java 方法的概念	30
4.2 Java 方法的定义	31
4.3 方法的调用	34
4.4 Java 的递归方法	37
4.5 本章小结	38
4.6 实践检验	38
第 5 章 Java 的类	40
5.1 类和对象的概念	40
5.2 类的组成及其运作规则	42
5.3 Java 应用程序的结构	49
5.4 本章小结	50
5.5 实践检验	50
第 6 章 继承与多态	54
6.1 继承和多态的概念	54
6.2 类成员的继承	57

6.3	接口.....	68
6.4	本章小结.....	72
6.5	实践检验.....	72
第7章	Java 的异常处理.....	75
7.1	异常、错误与事件.....	75
7.2	异常类的层次.....	76
7.3	异常处理语句及使用.....	77
7.4	异常处理举例.....	78
7.5	本章小结.....	81
7.6	实践检验.....	81
第8章	Java 包的组成和使用.....	83
8.1	Java 的包.....	83
8.2	Java 语言类库的结构.....	84
8.3	java.lang 包中的常用类介绍.....	85
8.4	关于 Java 的技术文档.....	97
8.5	本章小结.....	98
8.6	实践检验.....	98
第9章	Java 的集合类.....	100
9.1	集合类概述.....	100
9.2	集合类.....	100
9.3	新集合类.....	111
9.4	本章小结.....	121
9.5	实践检验.....	121
第10章	Java 的 I/O 流.....	124
10.1	流的层次结构.....	124
10.2	InputStream 流和 OutputStream 流.....	125
10.3	内存读写.....	126
10.4	管道流.....	127
10.5	文件输入/输出流.....	129
10.6	连接多个流.....	138
10.7	过滤流.....	140
10.8	PrintStream 流.....	143
10.9	本章小结.....	144
10.10	实践检验.....	144
第11章	Applet 基础和多媒体技术.....	146
11.1	第一个 Applet 小应用程序.....	146
11.2	Applet 的运行平台浏览器.....	147
11.3	Applet 程序的运行机理.....	150

11.4	Applet 程序的结构.....	152
11.5	Applet 程序设计.....	154
11.6	Applet 的安全机制.....	163
11.7	多媒体设计类的继承关系.....	163
11.8	对象的属性.....	163
11.9	设置字体.....	164
11.10	设置颜色.....	167
11.11	Graphics 类.....	170
11.12	图像处理.....	173
11.13	动画技术.....	178
11.14	本章小结.....	182
11.15	实践检验.....	182
第 12 章	多线程处理	184
12.1	线程的基本概念.....	184
12.2	线程的基本结构与使用方法.....	186
12.3	线程的管理.....	191
12.4	用于制作动画的线程.....	196
12.5	本章小结.....	200
12.6	实践检验.....	200
第 13 章	网络程序设计	202
13.1	网络概览.....	202
13.2	网络程序设计简介.....	205
13.3	服务器和客户机.....	206
13.4	网络程序设计技术.....	220
13.5	Java 支持的网络程序设计.....	221
13.6	移动网络.....	222
13.7	Java 安全性概览.....	223
13.8	本章小结.....	224
13.9	实践检验.....	225

第1章 Java 程序设计入门

知识点:

- 程序设计语言的分类
- Java 语言作为面向对象语言所具有的特点
- Java 程序的编译和执行

本章概述:

随着 Internet 的迅猛发展, Java 应运而生。它一推出就受到广大计算机编程人员的欢迎, 这是基于它以下特点: 面向对象、可移植、结构清晰、标准化程度高。Java 有两类程序, 即 Java 应用程序 (Java Application) 和 Java 小程序 (Java Applet), 前者是独立运行的应用程序, 后者是嵌入到 HTML (Hyper Text Mark Language) 浏览器中运行的应用程序。

1.1 从程序设计语言说起

程序设计语言是学习计算机技术的基础, 它由难而易经历了较长的发展过程。

1.1.1 发展历史

程序设计语言 (Program Language) 也称计算机语言, 即编写计算机程序所用的语言, 是人和计算机交流信息的工具。大致可以分为以下几个阶段:

1. 第一代编程语言——机器语言

机器语言是以二进制代码的形式组成的机器指令集合, 它依赖于机器, 即不同的机器有不同的机器语言, 存储安排也由语言本身控制。计算机可以直接识别和执行机器语言程序, 因此用这种语言编制的程序运行效率极高。但是因为程序全部由 0、1 二进制组成, 所以很不直观。即使实现很简单的功能也需要大量的代码, 编写效率低, 很容易出错, 而且不同的计算机有不同的机器语言, 不能通用, 重用性差。

计算机只能存储和识别二进制的数据和指令, 所以在机器语言中, 每条指令的操作码和地址码都用二进制 (或八进制) 编码。存放数据和指令的地址也用二进制 (或八进制) 编码, 数据也需预先转换成二进制, 因此机器语言也称为二进制语言。

2. 第二代编程语言——汇编语言

汇编语言也称符号语言。它是用符号代替机器语言中的二进制编码, 这样比较直观, 使用起来较方便, 错误也相对较少。汇编语言增加了一些功能, 如宏、符号地址等, 存储空间的安排由机器完成。

计算机不能直接识别和执行汇编语言程序, 它必须将一个汇编程序转换成机器语言后, 才能执行, 并且不同指令集的机器仍有不同的汇编语言, 所以它的程序重用性也很低。

3. 第三代编程语言——高级语言

高级语言是与机器无关的一类程序设计语言，更接近人类的自然语言。因此，用高级语言开发的程序可读性较好。同时，由于高级语言并不直接和硬件相关，因此其编制出来的程序可移植性和可重用性也要好得多。这类语言大多采用结构化的程序设计思想，编程思路是自顶向下，逐步求精。

4. 第四代编程语言

第四代编程语言是一种还未成熟的语言。它具有一定的智能，更接近于人们日常使用的语言，它对语言的概括更为抽象，从而使语言也更为简洁。

从程序设计语言的发展过程可以看到，人们在不断地简化编程的难度，使程序开发更快、更好。面向对象的程序设计思想也是在这种情况下产生的。

1.1.2 编译执行和解释执行

1. 编译执行

编译是指将所编写的源代码，通过特定的工具软件将其转换成机器代码，即可执行程序，然后直接在相应的操作系统中执行，也就是说程序是作为一个整体来运行的。它的优点是执行速度比较快。另外，程序经编译、链接之后可以独立在操作系统上运行，不需要其他应用程序的支持。其缺点是不利于调试，每次修改后都要执行编译链接等步骤，才能看到其执行结果。当然，有些集成开发环境可以提供单步追踪等工具，从而方便程序员调试程序。另外，这类程序设计语言的编译器与机器之间存在一定的依赖性，不同操作系统需要的编译器可能不相同，因此，在一个系统上编译的程序到另外一个系统上并不一定能够正常运行。常见的编译执行的程序设计语言有 Pascal 和 C 等。

2. 解释执行

解释执行是程序读入一句执行一句，而不需要整体编译链接。虽然这样的语言与操作系统的相关性较小，但运行效率低，而且需要一定的软件环境来做源代码的解释器。当然，有些解释执行的程序并不是使用源代码来执行，而是需要预先编译成一种解释器能够识别的格式，再解释执行。Java 就是这样一种语言。

1.1.3 从面向过程到面向对象

在面向对象技术产生之前，一直使用结构化程序设计方法，也就是面向过程的程序设计方法。但是，随着计算机科学的发展和应用领域的不断扩大，对计算机技术的要求也越来越高。由于结构化程序设计存在可重用性差和难以维护等缺点，因此结构化程序设计语言和结构化分析与设计已无法满足用户需求的变化。提高软件质量，缩短软件开发周期，发展软件可靠性、可扩充性和可重用性，这些要求促使软件界人士不断研究新方法、新技术，探索新途径，由此推动了面向对象的新技术的出现。

1. 面向过程方法面临的问题

面向过程的程序设计方法诞生于 20 世纪 60 年代，它的主要思想是自顶向下、逐步求精，换句话说，就是将功能不断细分，直到程序容易实现为止。当程序功能十分复杂，无法直接用算法进行描述时，可以将大的功能拆分为一系列较小的功能模块；如果已拆分的功能模块仍然复杂，则再将其进行拆分，直到这些功能模块易于实现为止。例如，计算一个班级中每个学生的平均成绩是一项较为复杂的任务，可以将其进行如下分解。

- (1) 找出一个学生的成绩;
- (2) 计算总共有多少学生;
- (3) 计算总成绩;
- (4) 用总成绩除以学生人数。

计算总成绩本身又可以分为一系列子任务:

- (1) 找出每个学生的档案;
- (2) 读出成绩;
- (3) 统计成绩;
- (4) 读出下一个学生的档案。

与此类似, 读出每个学生档案中的记录又可以分解为一系列子任务:

- (1) 打开学生的档案;
- (2) 找出正确记录;
- (3) 从磁盘读取数据。

面向过程的程序设计为处理复杂问题提供了有力的手段。该程序设计方法采用了模块分解, 可以有效地将一个复杂的程序系统的设计任务分成许多易于控制和处理的子任务, 这些子任务都是可独立编程的子程序模块。这些子程序都有各自清晰的接口, 使用起来非常方便。

但是, 人们在使用面向过程的方法进行程序设计的时候, 逐渐感到该方法的局限性。到 20 世纪 80 年代末, 它的一些缺点越来越突出。如当数据量增大时, 数据与处理这些数据的方法之间的分离使得程序的可读性和可重用性变差。因此, 人们试图开发一种新的程序设计方法。

2. 面向对象的概念

“面向对象”是软件程序设计中的一种新思想。这种新思想的引入使得程序设计更能贴近现实, 花费的精力更小。面向对象方法的出发点和所追求的基本目标是使人们分析、设计与实现一个系统的方法尽可能接近人们认识一个系统的方法, 也就是使描述问题的问题空间和解决问题的方法空间在结构上尽可能一致。其基本思想是: 对问题空间进行自然分割, 以更接近人类思维的方式建立问题域模型, 以便对客观实体进行结构模拟和行为模拟, 从而使设计出的软件尽可能直观地描述现实世界, 构造出模块化的、可重用的、维护性好的软件, 同时限制软件的复杂性和降低开发维护费用。

1.2 Java 语言的特性

在所有的面向对象程序设计语言当中, Java 是网络应用程序的主要语言, 它严格遵守着面向对象的编程思想和理念, 是学习面向对象程序设计思想最好、最有效的一种语言。同时, Java 又具备构造非常复杂应用的能力, 现在以及未来的很多应用程序都将基于 Java 来开发。

1.2.1 Java 简介

1995 年, 美国 Sun 公司正式向 IT 业界推出了 Java 语言, 该语言具有安全、跨平台、面向对象、简单、适用于网络等显著特点。当时以 Web 为主要形式的互联网正在迅猛发展, Java 语言的出现很快引起业内人士和公司的极大关注, 程序员们纷纷尝试用 Java 语言编写网络应用程序, 并利用网络把程序发布到世界各地。IBM、Oracle、Microsoft、Netscape、Apple、

SGI 等大公司纷纷与 Sun 公司签订合同, 授权使用 Java 平台技术。Microsoft 公司总裁比尔·盖茨在经过研究后认为, “Java 语言是很长时间以来最卓越的程序设计语言”。目前, Java 语言已经成为最流行的网络编程语言, 截止到 2001 年, 全世界大约有 310 万 Java 程序员, 许多大学也纷纷开设 Java 课程。Java 正逐步成为世界上程序员使用最多的编程语言。

1.2.2 面向对象的四个特色

面向对象的学习是一个反复了解, 逐渐深入的过程, 在这里, 只简单介绍其涉及的基本概念, 为引入 Java 程序做一个铺垫。

1. 对象与类

对象 (Object) 就是真实世界中的实体, 对象与实体是一一对应的, 也就是说现实世界中每一个实体都是一个对象, 它是一个具体的概念。

类 (Class) 是具备某些共同特征的实体的集合, 是一个抽象 (Abstract) 的概念。用程序设计的语言来说, 类是一种抽象的数据类型, 它是对所具有相同特征实体的抽象。例如, 通常把“轿车”、“公交车”、“卡车”分别看成 3 个相互独立的对象, 这 3 个对象有一些共同的特点, 如它们都有轮子, 都有控制方向的系统, 而且它们都会向前、后、左、右 4 个方向移动等等, 将这些特点抽象到一起, 就得到了一个类——汽车, 汽车是一种集合的概念, 是抽象的, 它可以被具体化。

可见, 类是对象的抽象, 而对象是类的具体化, 在专业术语中把这种具体化称为实例化 (Instantiation)。

2. 抽象

抽象就是把事物的共同点抽取出来, 以统一的方式进行概要描述的过程。这种过程是提高程序重用性的根本原因, 正是对许多类似的事物进行了抽象, 过去已经产生的成果才可以在相同或类似的环境下重用。

抽象是存在不同层次的。例如, 把汽车当成类, 把卡车当成车的一个实例, 而汽车实际上可以说是一种交通工具, 那么如果把“交通工具”当做一个类, 这时“汽车”可能就和船、飞机等事物一起成为交通工具的一个实例。因此, 在不同的需求环境下, 把什么抽象成类, 把什么当做实例或者对象对待, 是面向对象设计最关键的一步。如果抽象过度, 可能导致程序设计层次加重; 但如果抽象不够, 在极端的情况下可能会退化到面向过程设计的那种一段程序对应一个问题的情形。

3. 属性与方法

前面提到, 不同对象具有相同特点, 就可能抽象为一定的类, 那么这些特点基本上可以分为两类, 一类是描述对象静态状态的, 就是对象的属性 (Attribute), 在程序设计中, 可以称之为变量 (Variable); 另一类是描述对象的动作, 就是对象的方法 (Method), 在程序设计中, 可以称之为函数 (Function)。属性和方法是一个对象所具备的两大基本要素, 也是编程工作的核心。例如, 类“车”或者说对象“汽车”具有一些基本的属性, 包括颜色、功率、速度、位置等, 而它们的方法则包括前进、后退、加速、减速等。每种方法都像函数一样, 有一定的调用格式, 利用这些方法, 就可以驱动这些对象实现相应的工作。

4. 封装

从字面上理解, 封装就是将某事物包围起来, 使外界不知道其实际内容。在面向对象的方法中, 封装是指将一段程序代码“包装”起来, 只需知道这段程序代码所完成的功能, 而

不必弄懂完成这些功能的实现细节（这段程序代码包括的一组数据和与这组数据相关的操作）。例如，可以把理发师看成是一个封装体。如果需要理发，那么只需要告诉理发师所想要的发型，理发师则可以按提出的要求来完成这个操作。在这里，理发师相当于被包装起来的一段代码，理发是他所能完成的功能，而理发者则相当于使用这段代码的用户，理发者并不需要懂得如何理发。

1.2.3 Java 的开发及运行

Java 有两类程序，即 Java 应用程序和 Java 小程序，前者是在命令行中独立运行的应用程序，后者是需要嵌入 HTML 浏览器中执行的应用程序。

1. Java 程序的步骤

(1) 编写源代码。在这个阶段，需要一个无格式的文本编写器，如 Windows 的记事本等。程序编写好以后就可以保存到软盘或硬盘的某一目录下，这样的文件称为源程序。存储时，其后缀名必须是 Java。如果使用记事本，请注意存储时首先选择*. 的文件类型，然后在文件名后加上 Java 的后缀。建议读者把所有源文件都保存到一个目录下，便于运行。

(2) 编译源代码。虽然 Java 是解释型语言，但它仍有一个类似编译的过程，这和 Basic 不一样。在这个阶段，需要一个 Java 编译器——Javac.exe。源文件经过编译后生成以.class 为后缀的同名文件。编译有两个作用，一是可检查程序的语法错误，二是可在此阶段引入 Java 类库中的类。

(3) 解释执行程序。在这个阶段，需要一个 Java 解释器，如果编写的是 Java 小程序，它可以是浏览器，例如 Microsoft 公司的 Internet Explorer 或者网景公司的 Navigator。如果编写的是应用程序，就需要 Java.exe。

到此，了解了 Java 的运行过程，并知道第 (2)、(3) 阶段都需要一些工具。所以，在学习编程之前应在机器上建立起 Java 的运行环境。初学者所需要的工具都可以从 Sun 公司提供的 JDK 中找到，并到 www.sun.com 网站上去下载最新的 JDK。本书以 J2SDK 为例，它是 JDK 的 1.4.2 版，将 JDK 安装到合适的目录下，例如 d:\jdk 1.4.2 目录下。

2. Java 应用程序示例

下面是一个以 Windows 中的记事本编辑的 Java 应用程序，并在 JDK 下运行的过程。

(1) 编写程序。打开记事本，编辑例 1-1 的程序，保存到某一目录。需要注意的是：Java 是区分大小写的，保存的文件名与程序类名相同，其扩展名为 Java。



在记事本中编写 Java 文件，保存时一定要把文件名和扩展名用双引号引起来，否则，将默认保存为文本文件。例如，如果要保存的 Java 文件名为 HELLO.Java，则在保存时在 { 文件名 } 文本框中一定要输入“HELLO.Java”。

【例 1-1】HELLO.Java。

HELLO.Java 程序的内容如下：

```
public class HELLO
{
    public static void main (String args[ ])
    {
```



```
System.out.println("你好,很高兴学习 Java");
}
}
```

(2) 编译程序。在任务栏中执行“开始”|“运行”命令,在打开的“运行”对话框的命令行中输入“cmd”(在 Windows 98 中应输入“command”),按回车键,这时即可打开一个命令窗口,将目录转换到编写的 Java 源程序所在的目录,输入“JavaHELLO.Java”。

(3) 执行程序。同样在命令窗口中输入“JavaHELLO”。

3. Java 小程序示例

【例 1-2】MyApplet.html。

(1) 编辑一个如下所示的 Java 程序文件。

```
import Java.applet.Applet;
import Java.awt.Graphics;
public class Hello World extends Applet
{
    public void paint (Graphics g)
    {g.drawString ("您好, JAVA 世界! ",2,20);
    }
}
```

(2) 再用编辑器编辑一个 Applet.html 文件来调用该 Java 小程序。Applet.html 文件的内容如下:

```
<html>
<title>Applet Test Page</title>
<hl>Applet Test Page</hl>
<applet
    code="JavaProgram2.class"
    width=250
    height=250
    name= "MyApplet">
</applet>
</html>
```

使用与例 1-1 中同样的方法编译之后,再输入“MyApplet.html”。

假如读者的机器上安装了支持 Java 的浏览器,可以直接使用浏览器调用这个 HTML 文件,也可以直接用鼠标双击 HTML 文件的名字。如果读者使用的是 Microsoft 的 Internet Explorer,可以打开一个新的 IE 窗口,在地址栏中输入 HTML 文件的全路径名。

1.2.4 Java 平台

Java 平台由 Java 应用编程接口 (Application Programming Interface, API) 和 Java 虚拟机 (Java Virtual Machine, JVM) 构成。Java 应用编程接口是已编译的、可在程序中使用的代码库。它们使用户能够添加现成的、可定制的功能,以节约编程时间。一般认为,Java 虚拟机是建立在实际的处理器基础上的假想的计算机。Java 虚拟机规范为 Java 虚拟机作了如下定义:在真实机器中用软件模拟实现的一种想像的机器。Java 虚拟机代码被存储在.class 文件中,每个文件都最多包含一个 public 类。

Java 虚拟机规范为不同的硬件平台提供了一种编译 Java 技术代码的规范,该规范使 Java 软件独立于平台,因为编译是针对作为虚拟机的“一般机器”的,而这个“一般机器”可用软件模拟中并运行于各种现存的计算机系统中,也可用硬件来实现。

Java 程序并不是在本机操作系统上直接运行,而是通过 Java 虚拟机向本机操作系统进行解释来运行。这就是说,任何安装有 Java 虚拟机的计算机系统都可以运行 Java 程序,而不论最初开发应用程序的是何种计算机系统。

开始编写 Java 程序前,需要在用户的计算机系统中安装 Java 平台。Java 平台可从 www.sun.com 网站免费下载,该网站上提供了免费的 Java 开发工具集 (Java Developers Kits, JDK),用户可以在这里下载适合于自己计算机操作系统的 JDK,利用这些工具即可开发 Java 程序。

使用集成开发环境有很多的优点,无论是在调试程序的过程中,还是编译运行程序,都方便很多,所以一般的程序员都会选择一个集成开发环境来开发程序。Java 集成开发环境有 IBM 公司的 Visual Age For Java、Microsoft 公司的 Visual J++、Borland 公司的 JBuilder 等。

1.3 Java 的特点

随着 Internet 的迅猛发展,Java 应运而生。当 WWW 兴起时它就被众人看好。1995 年 Java 被评为十大优秀科技产品之一,再次掀起学习 Java 的热潮。与传统编程语言相比,Java 有以下几个特点。

1. 结构中立

平台无关性就是指 Java 能运行于不同的平台。Java 引进了虚拟机原理,并运行于虚拟机,实现了不同平台间的 Java 接口。使用 Java 编写的程序能在世界范围内共享。Java 的数据类型与机器无关,Java 虚拟机是建立在硬件和操作系统之上,实现 Java 二进制代码的解释执行功能,提供不同平台的接口。

2. 安全

Java 的编程类似于 C++,学习过 C++的读者将很快能掌握 Java 的精髓。Java 舍弃了 C++的指针对存储器地址的直接操作,程序运行时,内存由操作系统分配,这样可以避免病毒通过指针侵入系统。Java 对程序提供了安全管理器,以防止程序的非法访问。

3. 面向对象

Java 吸取了 C++面向对象的概念,利用了类的特点,将数据封装于类中,使程序简洁且便于维护性。类的封装性、继承性等特性,使程序代码只需一次编译就可反复利用。因此程序员只需把主要精力用在类和接口的设计和应用上。Java 提供了众多的一般对象的类,通过继承即可使用父类的方法。Java 已有类的功能十分强大,如输入/输出类,进行普通输入/输出和文件操作;抽象窗口设计 AWT 类集,拥有设计图形界面的常用部件,在进行图形用户界面的设计时给程序员带来很大的方便。

Java 中类的继承关系是单一的,而非多重的,即一个子类仅有一个父类,子类的父类又可以有一个父类。Java 提供的 Object 类及其子类的继承关系就像一棵倒立的树形,根类为 Object 类。Object 类的功能非常强大,在后面的内容中将会反复用到 Object 类以及由它派生而来的子类。

4. 简单

Java 舍弃了 C++的头文件,没有全局变量,从而节省了维护头文件的工作量。头文件中有全局变量的定义,这对可维护性和安全性都是不可取的。Java 舍弃了 C++的多重继承,引

进了垃圾自动回收机制，程序员无需考虑设计内存管理方面的问题。

5. 动态

Java 源程序经过编译后生成的二进制代码存在于网络计算机中。Java 运行时动态加载，即当程序运行到所需类时，便在网上寻找并下载到本地机进行动态加载。

6. 分布式

Java 允许将编译后的二进制代码分布于网络上。Java 应用程序可以通过 URL 统一资源定位来寻找应用程序所需的类，就像访问本地机一样。

7. 多线程

多线程是 Java 的一大特点，利用它能够在程序中实现多任务操作。传统的程序设计语言的程序只能单任务操作，效率非常低，例如，程序往往在接受数据输入时被阻塞，只有等到程序获得数据后才能够继续运行。而多线程程序可以在创建一个线程来进行输入/输出操作的同时，创建另一个线程在后台进行数据处理。输入/输出的线程在接收数据时阻塞，而另一个线程仍然在运行，大大提高了运行的效率和处理能力。

Java 提供了有关线程的操作、创建、管理和废弃等处理。Java 虚拟机也是一个多线程程序，虚拟机启动后，每时每刻都在运行一个线程，该线程的优先级最低，它是用于在后台负责垃圾处理工作。多线程使程序能够处理多项任务，其发展前景非常广阔。

1.4 本章小结

本章首先介绍了对学习 Java 比较有帮助的程序设计语言分类。然后介绍了 Java 之所以能够在计算机和网络技术高度发达的今天占据这样关键性的地位所具有的优点：面向对象、可移植性、结构清晰、标准化程度高。

如果有一个 Java 集成开发环境，不仅可以提供一个能够识别 Java 语法的编辑器，使读者简单地编译和执行 Java 程序，还可以建立应用程序或者小程序的框架，并调试程序，将做好的程序打包成可执行文件。

1.5 实践检验

理论巩固

1. Java 对软件开发技术的影响有哪些？
2. Java 的应用前景及特点是什么？
3. Java 语言的特点有哪些？
4. Java Applet 的特点是什么？
5. 请说出 Java 丰富的类库的作用。
6. 什么是面向对象的软件开发？面向对象软件开发的一般过程分为哪些阶段？
7. 面向对象软件开发的最主要特点是什么？

上机实战

请读者在自己的计算机上安装并配置 Java 运行环境，并编辑运行本章中的两个例程。