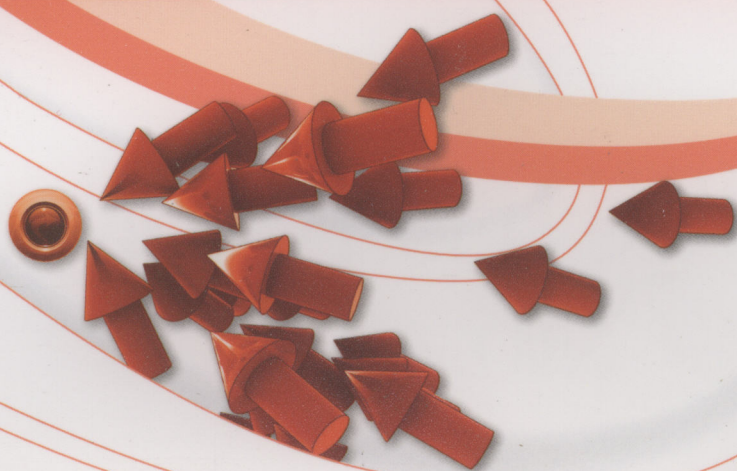




普通高等教育“十一五”国家级规划教材



高等学校计算机规划教材

Java 程序设计实用教程 (第3版)

■ 叶核亚 编著

■ 陈道蓄 主审



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

普通高等教育“十一五”国家级规划教材

高等学校计算机规划教材

-10

Java 程序设计实用教程

(第3版)

叶核亚 编著
陈道蓄 主审

7P312
YU3U05
電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

书号: 7-302-15650-2 Java 2 程序设计实用教程

内 容 简 介

本书为普通高等教育“十一五”国家级规划教材。

本书全面、系统地介绍 Java 语言程序设计的基础知识、运行机制及各种常用编程方法和技术,将面向对象的程序设计思想贯穿其中,建立起牢固扎实的理论基础。本书主要内容包括:Java 开发运行环境 JDK 和 MyEclipse,Java 语言基础,类的封装、继承和多态,接口、内部类和 Java API,异常处理,图形用户界面,多线程,Applet,输入/输出流和文件操作,网络通信,数据库应用和综合应用设计。这些内容是构成 Java 应用程序的基本要素和必备知识。

本书注重理论与实践相结合,注重基础知识的理解与基本技能的培养。全书内容丰富,结构安排合理,由浅入深,层次分明,内容涉及的广度和深度符合教指委制订的教学基本要求。

本书可作为普通高等学校计算机及相近专业本科生(或高职高专学生)的 Java 语言程序设计课程教材,也可作为使用 Java 语言的软件开发人员的参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Java 程序设计实用教程 / 叶核亚编著. —3 版. —北京:电子工业出版社,2010.2

高等学校计算机规划教材

ISBN 978-7-121-10171-7

I. J… II. 叶… 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(2010)第 000281 号

策划编辑:章海涛

责任编辑:章海涛

印 刷:北京丰源印刷厂

装 订:涿州市桃园装订有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:26 字数:670 千字

印 次:2010 年 2 月第 1 次印刷

印 数:5 000 册 定价:35.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系电话:(010) 68279077;邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

第3版前言

程序设计是高等学校计算机学科及电子信息学科各专业本科的核心专业基础课程，是培养学生软件设计能力的重要课程，在计算机学科的本科教学中起着非常重要的作用。

面向对象程序设计方法是目前软件开发的主流方法。Java 语言是目前功能最强、应用最广泛的一种完全面向对象程序设计语言，具有面向对象、与平台无关、多线程以及强大的网络编程功能等特点。因此，以 Java 语言作为程序设计和面向对象方法的基础训练课程是十分恰当的。开设 Java 程序设计课程的目的是，进行程序设计和面向对象方法的基础训练，应用新技术开发各种适用于网络环境的应用程序。

本书是普通高等教育“十一五”国家级规划教材，定位于普通高等学校计算机及相近专业本科生的 Java 语言程序设计课程教材。本书以“理论性、实用性、新技术”为编写目标，全面系统地介绍 Java 语言的知识点及最新发展技术，特色体现在内容全面、注重基础、面向应用、技术前沿、实例丰富。

(1) 内容全面、注重基础

本书全面、系统地介绍 Java 语言程序设计的基本知识、运行机制、多种编程方法和技术，将面向对象程序设计思想贯穿其中，建立起牢固扎实的理论基础。本书主要内容包括：Java 开发运行环境 JDK 和 MyEclipse，Java 语言基础，类的封装、继承和多态，接口、内部类和 Java API，异常处理，图形用户界面，多线程，Applet，输入/输出流和文件操作，网络通信，数据库应用和综合应用设计。这些内容是构成 Java 应用程序的基本要素和必备知识。

本书注重理论与实践相结合，注重基本知识的理解与基本技能的培养。全书内容丰富，结构安排合理，由浅入深，层次分明，内容涉及的广度和深度符合教指委制订的教学基本要求和本科培养目标的要求。

(2) 面向应用、跟踪前沿新技术

本书以循序渐进方式介绍 Java 程序设计的多种实用技术，并尽可能将最新技术反映在教材中；注重理论知识和实用新技术相结合，注重基本知识的理解与基本技能的培养，训练学生既具有扎实深厚的基本功，又具有可扩展素质和较强的创新能力。

本书采用最新版本的 JDK 6、MyEclipse 7.5、MySQL 作为 Java 开发运行环境。

(3) 实例丰富、生动、联系实际

“Java 程序设计”是一门理论与实践并重的课程，不仅要理解基础知识，更要培养软件设计的基本技能。实践环节是巩固所学理论知识、积累程序设计经验的必不可少的重要环节，是提高程序设计能力和计算机操作技能的有力保障。

本书将程序设计训练穿插在理论叙述中，以多个典型示例体现和巩固理论知识；讲解深入浅出，通俗易懂，易学易用；丰富的实例开阔学生视野，使学生尽快具备应用程序开发能力，并培养良好的程序设计习惯。

注重在实践环节培养程序设计的基本技能，是本书的重要特色，也是本书前两版的重要特色。每章安排有习题和上机实验题，给出详细的实验训练目标、设计内容和设计要求，精心选择实验题，题目形式多样，生动有趣，引人入胜，难度逐步增加。

针对课程设计的实践性环节，本书给出综合应用程序设计实例，给出设计目标、设计任务、模块划分及功能实现等，贯彻了理论讲授和案例教学相结合的教学方法。

由于 Sun 公司自 2005 年起将 Java 版本及平台更名，取消其中的数字“2”，JDK 1.6 更名为 Java SE6，J2SE 更名为 Java SE 等，因此本书第 3 版也相应更名为《Java 程序设计实用教程（第 3 版）》。

这是一本写程序设计的书。程序设计有一些基本原则和道理。程序是设计出来的，程序员必须具备基础知识和基本技能。在动笔写程序之前，需要根据实际应用需求，从全局角度通盘规划考虑，精心策划，选择采取什么策略，清楚有哪些方法可以做得成，每种方法都有什么优缺点，明白为什么要这样做，那样做为什么就不可以，等等。

程序中发现错误了怎么办？这是什么错误，是否能改正，如何改正？这种思路的程序是否值得再继续做下去？作为一个有价值的软件系统，程序要能够预见可能出现的错误，不能预见的错误要事后补救。程序员要知道如何避免和如何补救，不能补救的错误要及时放弃，采取别的策略。总之，无论采用什么方法都要把事情做成。

程序写完了，即使调通了，还必须再想想，程序还有哪些不足？还有哪些情况没有考虑到？是否还能进一步提高算法效率？不但要把事情做成，还要把事情做好，尽一切努力做得更好。这样，不知不觉间，程序设计能力就会提高很多，“轻舟已过万重山”。

写程序是创作，创作过程是艰苦的，也是快乐的。当程序调试通过时，我们会自鸣得意，哼着小曲，心情畅快，“春风得意马蹄疾”。人的一生能有值得沉浸其中的事业是幸福的。

全书由叶核亚编著，南京大学计算机科学与技术系陈道蓄教授主审。陈立、廖雷、阚建飞、陈建红、陈瑞、王青云、徐金宝、李林广、沈晨鸣、王少东、王忠磊等参与了本书的编写和资料收集工作。在写作过程中，得到了许多帮助和支持。感谢陈老师认真细致地审阅了全稿，提出了许多宝贵意见。感谢陈立、廖雷、阚建飞、陈建红、陈瑞、王青云、徐金宝、李林广、沈晨鸣、王少东等老师给予的帮助，感谢电子工业出版社的支持，感谢众多读者朋友对前两版书提出的宝贵意见。大家对我的指导和帮助使我受益匪浅，受用终身。能与志同道合的人一起讨论共同关心的问题是愉快的，工作也因此而变得更有动力。

对书中存在的不妥与错漏之处，敬请读者朋友批评指正。

本书的全部例题和配套课件可从华信教育资源网站 (<http://www.hxedu.com.cn>) 下载，也可发邮件至 unicode@phei.com.cn 或 yeheyax263.net 索取。

作 者

目 录

第 1 章 Java 概述	1
1.1 了解 Java	1
1.1.1 Java 的诞生和发展	1
1.1.2 Java 的特点	2
1.1.3 Java 核心技术	4
1.2 Java 应用及其运行方式	5
1.2.1 Application 和 Applet	5
1.2.2 Java 虚拟机执行 Java 程序	6
1.3 JDK	6
1.3.1 JDK 的安装与设置	7
1.3.2 编译和运行 Java 程序	8
1.3.3 包	11
1.4 MyEclipse	15
1.4.1 MyEclipse 集成开发环境	15
1.4.2 创建 Java 项目并运行	17
1.4.3 程序调试技术	20
习题 1	22
实验 1 Java Application 程序的编辑、编译和运行	23
第 2 章 Java 语言基础	24
2.1 语言成分	24
2.1.1 标识符与关键字	24
2.1.2 基本数据类型	25
2.1.3 变量与常量	28
2.1.4 运算符与表达式	29
2.2 流程控制语句	36
2.2.1 流程控制结构	36
2.2.2 顺序结构语句	37
2.2.3 选择语句	38
2.2.4 循环语句	41
2.2.5 转移语句	47
2.3 数组	48
2.3.1 一维数组	48
2.3.2 二维数组	51
2.4 静态方法	54
2.4.1 方法声明与调用	54

2.4.2	参数传递	55
2.4.3	方法重载	59
2.4.4	递归方法	60
2.5	字符串	61
2.5.1	字符串的基本数据类型特性	61
2.5.2	字符串的类特性	62
习题 2		64
实验 2	Java 程序设计基础	65
第 3 章	类的封装、继承和多态	68
3.1	类和对象	68
3.1.1	类	68
3.1.2	对象	70
3.2	类的封装性	72
3.2.1	构造方法	73
3.2.2	this 引用和 instanceof 对象运算符	74
3.2.3	访问控制	75
3.2.4	静态成员	76
3.2.5	析构方法	81
3.2.6	浅拷贝与深拷贝	81
3.3	类的继承性	84
3.3.1	由继承派生类	84
3.3.2	继承原则及作用	85
3.3.3	子类的构造方法	87
3.4	类的多态性	88
3.4.1	子类重定义从父类继承来的成员	88
3.4.2	类型的多态	92
3.4.3	编译时多态和运行时多态	93
3.4.4	多态的方法实现	94
3.5	类的抽象性	97
3.5.1	用继承刻画包含关系	97
3.5.2	抽象类	98
3.5.3	最终类	101
习题 3		102
实验 3	类的封装、继承和多态	104
第 4 章	接口、内部类和 Java API 基础	106
4.1	接口	106
4.1.1	接口与实现接口的类	106
4.1.2	Java 用接口实现多继承	111
4.2	内部类和内部接口	112
4.3	java.lang 包中的基础类库	115

4.3.1	Object 类	115
4.3.2	Math 数学类	116
4.3.3	Comparable 可比较接口	116
4.3.4	基本数据类型的包装类	117
4.3.5	String 字符串类	118
4.3.6	Class 类操作类	119
4.3.7	System 系统类和 Runtime 运行时类	119
4.4	java.util 包中的工具类库	120
4.4.1	日期类	120
4.4.2	Arrays 数组类	122
4.4.3	Random 随机数序列类	126
习题 4		126
实验 4	接口与实现接口的类	127
第 5 章	异常处理	128
5.1	异常处理基础	128
5.1.1	异常处理机制的必要性	128
5.1.2	错误和异常	129
5.2	异常处理措施	132
5.2.1	异常处理语句	132
5.2.2	抛出异常	137
5.2.3	自定义异常类	138
习题 5		140
实验 5	异常的抛出、捕获并处理	141
第 6 章	图形用户界面	142
6.1	AWT 组件及其属性类	142
6.1.1	AWT 组件	142
6.1.2	布局管理	146
6.1.3	颜色和字体	148
6.2	事件处理	149
6.2.1	委托事件模型	149
6.2.2	AWT 事件类和事件监听器接口	152
6.3	Swing 组件及事件	155
6.3.1	Swing 组件关系概述	155
6.3.2	窗口组件	155
6.3.3	文本显示和编辑组件	157
6.3.4	按钮组件	161
6.3.5	列表框和组合框	162
6.3.6	JOptionPane 对话框和 JColorChooser 对话框	165
6.3.7	菜单组件	167
6.3.8	表格	175

6.4 图形图像	177
6.4.1 图形设计	178
6.4.2 图像	183
习题 6	184
实验 6 图形用户界面设计	185
第 7 章 多线程	188
7.1 操作系统中的进程与线程	188
7.1.1 进程	188
7.1.2 线程	189
7.1.3 并发程序设计	191
7.2 Java 的线程对象	191
7.2.1 Runnable 接口与 Thread 类	191
7.2.2 线程对象的生命周期	195
7.2.3 线程对象的优先级	200
7.2.4 守护线程	201
7.2.5 定时器与图形动画设计	201
7.3 线程的同步机制	204
7.3.1 交互线程	204
7.3.2 线程间的竞争关系与线程互斥	207
7.3.3 线程间的协作关系与线程同步	211
习题 7	220
实验 7 线程设计	221
第 8 章 Applet 应用程序	222
8.1 Web 应用基础	222
8.1.1 Internet	222
8.1.2 WWW 浏览	224
8.2 运行 Applet	226
8.2.1 什么是 Applet	226
8.2.2 Applet 的生命周期	228
8.3 HTML 中的 Applet 标记	230
8.4 Applet 支持的多媒体技术	232
8.4.1 获得资源文件的地址	232
8.4.2 显示图像	232
8.4.3 播放音频	233
习题 8	235
实验 8 Applet 应用程序设计	236
第 9 章 输入/输出流和文件操作	237
9.1 文件和流的概念	237
9.1.1 操作系统中的文件和目录概念	237
9.1.2 流的概念	239

9.1.3	Java 的输入/输出流与文件操作概述	240
9.2	字节输入/输出流类	241
9.2.1	抽象字节流	241
9.2.2	Java 标准输入/输出	242
9.2.3	文件字节流	244
9.2.4	数据字节流	247
9.2.5	对象字节流	249
9.2.6	管道字节流	251
9.3	字符输入/输出流类	256
9.3.1	抽象字符流	256
9.3.2	文件字符流	257
9.3.3	缓冲字符流	257
9.4	文件操作	259
9.4.1	文件操作类	259
9.4.2	文件过滤器接口	262
9.4.3	文件对话框组件	264
9.4.4	随机存取文件类	276
习题 9		278
实验 9	输入/输出流与文件操作	278
第 10 章	网络通信	280
10.1	使用 URL 访问网络资源	280
10.1.1	URL 和 IP 地址	280
10.1.2	JTabbedPane、JSplitPane 和 JEditorPane	282
10.2	TCP Socket 通信	287
10.2.1	TCP Socket 通信原理	287
10.2.2	Java 的 TCP Socket 通信	289
10.3	UDP 数据报通信	300
10.3.1	UDP 数据报	301
10.3.2	UDP 组播数据报	304
习题 10		310
实验 10	网络通信	310
第 11 章	数据库应用	312
11.1	关系数据库系统	312
11.1.1	数据库系统	312
11.1.2	关系模型	312
11.1.3	C/S 结构的关系数据库系统	314
11.1.4	结构化查询语言 SQL	316
11.1.5	Access 数据库	319
11.1.6	MySQL 数据库	321
11.2	JDBC	328

11.2.1 JDBC 的作用和功能	328
11.2.2 指定 JDBC 驱动程序	330
11.2.3 连接数据库	331
11.2.4 执行 SQL 语句	333
11.2.5 处理数据查询的结果集	334
习题 11	339
实验 11 数据库应用设计	340
第 12 章 综合应用设计	341
12.1 Java 集合框架	341
12.2 使用复杂 Swing 组件	346
12.2.1 BorderLayout 和 GridBagLayout 布局	346
12.2.2 JList 列表框	348
12.2.3 组合框模型	353
12.2.4 JTable 表格	353
12.2.5 JTree 树	360
12.2.6 多文档界面	368
12.3 数据库应用	373
12.4 课程设计的要求和选题	388
附录 A ASCII 字符与 Unicode 值	394
附录 B Java 关键字	395
附录 C Java 基本数据类型	397
附录 D Java 运算符及其优先级	398
附录 E Java 类库（部分）	399
E.1 java.lang 语言包	399
E.2 java.util 实用包	403
E.3 java.text 文本包	404
附录 F MyEclipse 常用菜单命令	405

第 1 章 Java 概述

物竞天择，适者生存。同自然界的进化规律一样，程序设计语言、程序设计思想的变化和发展也是随着实际应用需要而变化 and 发展的。我们今天所看到、所使用的程序设计语言，经历了一系列竞争和淘汰之后仍然存在，说明它们有存在的道理，必定各有所长。这是一种自然选择的结果。

1995 年，作为一种完全面向对象的程序设计语言，Java 以一种全新的面貌问世。至今短短的十几年间，凭借平台无关、健壮、安全、高效这些适应网络运行需要的特点，Java 快速成长，不仅在 Internet 上游刃有余，而且通过 Java ME、Java SE、Java EE 三大平台，其应用领域全面覆盖了嵌入式应用、桌面应用和企业级应用，所表现出的强大的应用系统设计能力，使 Java 无处不在。

本章将简要介绍 Java 特点和核心技术，介绍 Java 两种应用程序 Application 和 Applet 的基本形式，以及由虚拟机支持的程序运行机制；以 Windows 平台的 Java SE6 版本为例，介绍 JDK 的安装和设置方法，以及编译、运行 Java 程序的方法；介绍在 Java 集成开发环境 MyEclipse 中编辑、编译和运行 Java 程序的方法。

1.1 了解 Java

1.1.1 Java 的诞生和发展

1. 前身

Java 的故事最早与一棵大橡树有关。1991 年，Sun 公司成立 Green 项目组，目的是开发嵌入家用电器的分布式软件系统，如交互式有线电视（Cable TV）和家用电器的设备控制等，使电器更加智能化。由于这些电子设备品种繁多且标准各异，Green 项目组希望该控制系统具有简单、可靠、安全、容易连网和跨平台等特性，并且具有支持系统开发的编程工具。

Green 项目组最初采用 C++ 语言开发，由于 C++ 语言太复杂而且安全性差，不能满足要求，于是 Green 项目组研究设计了一种新语言，取名为 Oak（橡树），这是因为 Green 项目组负责人 James Gosling 办公室窗外有一棵大橡树。

Oak 是一种适用于网络编程的精巧而安全的语言，它保留了 C++ 语言的语法，但去除了明确的资源引用、指针算法和操作符重载等潜在的危险特性。并且，Oak 语言具有与硬件无关的特性，制造商只需要更改芯片，就可以将烤面包机上的程序代码移植到微波炉上或其他电器上，而不必改变软件，这就大大降低了开发成本。

当 Oak 语言成熟时，全球 Internet 也在迅速发展。Sun 公司的开发小组认识到 Oak 非常适合于 Internet 编程。1994 年，他们完成了一个用 Oak 语言编写的 Web 浏览器，称为 HotJava，展示了 Oak 作为 Internet 开发工具的能力。

2. 诞生

由于商标冲突，1995 年，Oak 语言更名为 Java 语言。Java 取名于印度尼西亚位于印度洋

中的爪哇岛，它盛产咖啡。Java 语言的标志就是一杯热咖啡。美国著名杂志《PC Magazine》将 Java 语言评为 1995 年十大优秀科技产品。

Java 包括 Java 编程语言、开发工具和环境、Java 类库等。JDK (Java Development Kit, Java 开发工具包) 提供 Java 运行时环境。1996 年, Sun 公司发布了 JDK 1.0 和 HotJava。HotJava 通过嵌入在 Web 网页中的 Applet 运行 Java 程序, 这一特性相继被 Microsoft 公司和 Netscape 公司的 Web 浏览器所支持。IBM、Apple、DEC、Adobe、Silicon Graphics、HP、Oracle 和 Microsoft 等公司相继购买了 Java 技术许可证, 从此, Java 成为了日益流行的计算机语言。

3. Java 2 平台

1998 年, Sun 公司发布 JDK 1.2, 称为 Java 2 SDK (Software Development Kit), 此后的 Java 就称为 Java 2 平台。Java 2 不仅兼容于智能卡和小型消费类设备, 还兼容于大型服务器系统, 它使软件开发商、服务提供商和设备制造商更加容易抢占市场机遇。这一开发工具极大地简化了编程人员编制企业级 Web 应用的工作, 把“一次编程到处使用”的诺言应用到服务器领域。

Sun 公司采取开放性策略, 在其网站 <http://java.sun.com> 上可以免费获取 JDK, 这也是 Java 语言能够迅速发展的重要因素。不同的操作系统平台需要使用不同的 JDK。

4. Java ME、Java SE、Java EE 三大平台

1999 年 6 月, Sun 公司推出的 JDK 1.3 将 Java 平台划分为 J2ME、J2SE 和 J2EE, 这三个平台分别定位于嵌入式应用、桌面应用和企业级应用, 使 Java 技术获得了最广泛的应用。用户可根据实际应用领域的需求选择不同的 Java 平台。

2004 年, J2SE 1.5 发布, 这是 Java 语言在其发展史上的又一里程碑事件。为了表示这个版本的重要性, J2SE 1.5 更名为 J2SE 5.0。2005 年, JavaOne 大会召开, Sun 公司公开 Java SE6。此时, Java 的各种版本被更名, 取消其中的数字“2”: J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。

三大平台说明如下。

① 嵌入式平台 Java ME (Java Micro Edition) 是适用于小型设备和智能卡的 Java 嵌入式平台, 提供 Java Card、Java Telephony 和 Java TV 等技术, 支持智能卡业务、移动通信、电视机顶盒等功能。

② 标准平台 Java SE (Java Standard Edition) 是适用于桌面系统的 Java 标准平台。Java SE SDK 也简称 JDK, 它为创建和运行 Java 程序提供了最基本的环境, 包含 Java 编译器、Java 类库、Java 运行时环境和 Java 命令行工具。

③ 企业级平台 Java EE (Java Enterprise Edition) 是 Java 的企业级应用平台, 提供分布式企业软件组件架构的规范, 具有 Web 性能, 具有更高的特性、灵活性、简化的集成性、便捷性以及 Java EE 服务器之间的互操作性。

进入 21 世纪以来, 随着 Web 技术成为展示和操纵数据的事实标准, 企业利用 Java EE 平台对原来分散的子系统进行整合。尽管应用整合可以通过多种手段来实现, 但在 Java EE 出现后, 因其天生具备良好的开放性和可扩展性, 使之在应用整合和开发的过程中发挥了越来越显著的优势。Java EE 逐渐成为开发商创建电子商务应用的事实标准。

1.1.2 Java 的特点

Java 伴随着网络的飞速发展而发展, 作为软件开发的一种革命性技术, 其地位已被确定。

虽然它的历史短暂，但它的发展壮大并不是偶然的，而是有着内在的基础和外在的机遇的。Java 语言建立在成熟的算法语言和坚实的面向对象理论的基础之上，具有强大的应用系统设计能力，而它所具备的平台无关性、面向对象和安全性等特点是其能够充分适应网络需要的无可比拟的优势。Java 成为目前网络编程的首选语言，充分说明了 Java 语言的设计思想和其所具有的特点适应了网络发展的特殊需要。不仅网络应用，在企业级应用领域，Java 以更简单、更精练的方式实现了 C++ 语言的所有功能。如今，Java 技术已被列为当今世界信息技术的主流之一。

Java 应用如此广泛是因为 Java 具有多方面的优势，其特点说明如下。

1. 平台无关性

平台无关性，是指一个应用程序能够运行于不同的操作系统平台，也就是 Sun 公司设计 Java 的宗旨“Write once, run anywhere”。平台无关性使 Java 应用程序可以运行在多种操作系统（Windows、UNIX 等）平台上，这也是 Java 区别于其他高级语言的最重要标志。

Java 采用虚拟机技术支持平台无关性。Java 虚拟机（Java Virtual Machine, JVM）是一套支持 Java 语言运行的软件系统，运行于操作系统之上。Sun 公司为各种操作系统配有相应的 Java 虚拟机。

高级语言通常将程序编译生成可执行文件（*.exe），再由操作系统直接执行；与此方式不同，Java 源程序（*.java）经编译生成的是一种中间代码，称为字节码文件（*.class），它由 Java 虚拟机解释执行，因而与操作系统平台无关。

Java 语言本身设计也体现出平台无关性。例如，int 型整数占用 32 位，与操作系统是 16 位、32 位或 64 位无关。

2. 完全面向对象和简单性

Java 语言从 C++ 语言发展而来，它有选择地继承了 C++ 语言的语法规则和面向对象的基本机制，放弃了 C++ 语言模糊、过于复杂、安全性差、不适合网络应用的诸多特性。

Java 语言是完全面向对象的，所有设计都必须在类中实现，一个 Java 程序就是多个类的集合；为 8 种基本数据类型提供相应的基本数据类型包装类，使基本数据类型与类相关联；将数组设计为引用类型，使用方法与对象相同，并且每个数组都有长度属性；不支持类似 C 语言的面向过程设计，不支持类似 C++ 语言的全局变量、goto 语句、宏定义、全局函数等语法规则，不支持结构、联合和指针等数据类型。Java 语言通过引用模型实现了指针的功能，通过类实现了结构类型。

Java 语言提供单重继承机制，使得所有类（包括 Java 声明的类和程序员声明的类）能够形成具有树结构的类的层次体系。Java 语言为这个树结构设置了根类 Object。Object 类声明对象的基本状态和行为，这些行为可被所有对象继承。Java 不支持多重继承、友元类和运算符重载，通过“单重继承+接口”方式实现了多重继承功能。

Java 语言提倡简单性原则，对一个问题只提供一种简单精练的表达方法，这样使程序简单、直接并且不造成歧义。例如，使用下标形式对数组元素进行操作，则不需要使用指针；方法（函数）采用返回值或引用类型参数返回结果，也不需要使用指针；有了类，则不需要结构类型；构造方法采用重载方式，则不需要采用参数默认值形式，避免产生歧义；通过方法实现类的操作，则不需要运算符重载等。因此，放弃结构、指针、多重继承等，并没有影响 Java 语言的功能，而 Java 提供的机制具有更强的功能和更高的性能。

3. 可靠性

Java 在语言和运行架构两个级别上提供程序运行稳定性和可靠性保证。

(1) 语言级别

Java 语言提供严密的语法规则，在编译和运行时进行严格检查，降低程序出错的可能性。例如，不同类型的数据不能进行运算；数组下标不能越界，避免有效数据被覆盖；不能使用指针对指定地址的内存区域进行操作等，避免非法内存操作的危险。

Java 语言提供的异常处理机制，使程序具备在运行过程中及时发现并处理运行时错误的的能力，保证 Java 程序运行的稳定和可靠。

(2) 运行架构级别

Java 语言提供的资源回收机制（也称为垃圾回收机制），对内存资源进行自动管理，跟踪程序使用的所有内存资源，自动收回不再被使用的内存资源。因此，程序中不需要写释放内存空间的语句。Java 自己操纵内存减小了内存出错的可能性，减轻了程序员的工作量，提高了程序运行的可靠性。

4. 安全性

Java 采用域管理方式的安全模型，无论是本地代码还是远程代码，都可以通过配置策略设定可访问的资源域。这种策略使未经授权的代码不能对用户本地资源进行操作，更好地支持了企业级应用，也消除了区分本地代码和远程代码带来的困难。例如，Applet 应用程序在将远程 Web 页面下载到本地运行时，Java 将进行严格的代码安全性（code security）检测，限制了许多可能危害网络安全的操作，如不能访问本地文件、不能建立新的网络连接等。

5. 多线程

就像多进程机制能够使一个应用程序的多个进程并发执行一样，多线程机制使一个进程能够被划分成多个线程，每个线程执行一个特定功能，多个线程能够并发执行。Java 语言具有内置的多线程机制，多线程的同步特性提供对共享数据操作的正确性。

6. 直接支持分布式的网络应用

Java 早期采用将 Applet 嵌入浏览器运行的方式实现 Web 应用，完成动画、游戏、交互式事务处理等功能，使 Web 页面具有动感，而不只是被动展现文本和图形，提高了 Web 页面的智能化和灵活性。目前，JSP（Java Server Pages）提供服务器端的动态 Web 页面设计，功能更强，效率更高。Java 支持多种协议的 URL 寻址，支持 Socket 等网络通信方式。

除了支持基本的语言功能以外，Java 核心类库还包括一个支持 HTTP、SMTP 和 FTP 等基于 TCP/IP 协议的类库。因此，Java 应用程序可凭借 URL 打开并访问网络上的对象，其访问方式与访问本地文件系统完全相同。在 Java 出现以前，为分布式环境尤其是 Internet 提供动态的内容无疑是一项非常宏伟、难以想象的任务，但 Java 语言的特性却使我们能很容易地达到这个目标。

1.1.3 Java 核心技术

Java 支持的多种核心技术，满足了企业级应用对软件在安全性、可靠性、高效性上的高要求，使 Java EE 成为开发分布式企业级应用的工业标准。

Java 的部分核心技术说明如下。

① JDBC (Java DataBase Connectivity, Java 数据库连接) 定义了 Java 应用程序与关系型数据库连接的规范, 类似于 Windows 的 ODBC (Open DataBase Connectivity, 开放式数据库连接)。

② JSP 是 Sun 公司 1999 年推出的一种动态网页技术标准, 是一种运行于 Web 服务器端的标记语言, 它能够将 Java 语言直接嵌入到 HTML 文档中。JSP 是基于 Java Servlet 及 Java 体系的 Web 开发技术, 可以建立跨平台、安全、高效的动态网站。

JSP 与 Microsoft 的 ASP (Active Server Pages) 技术相似, ASP 基于 Windows 操作系统, 采用 VBScript、JavaScript 脚本语言; JSP 是跨平台的, 采用 Java 语言, 功能更强, 效率更高, 具有动态页面与静态页面分离、编译后运行的特点。

③ JavaBeans 是 Java 的对象组件技术, 提供组件复用的关键技术, 类似于 Windows 的 ActiveX。在 JSP 中可以使用 JavaBeans。EJB (Enterprise JavaBeans) 提供企业级的 JavaBeans。

④ JavaMail 提供 E-mail 邮件服务的支持类库, 不仅支持 SMTP 服务器, 也支持 IMAP 服务器。

1.2 Java 应用及其运行方式

1.2.1 Application 和 Applet

Java 应用程序有两种形式: Application 和 Applet, 其编译和运行方式见 1.3 节和 1.4 节。

1. Application

Application 是能够独立运行的应用程序, 有控制台和图形用户界面两种运行方式。

Java 应用由一个或多个扩展名为“.java”的文件组成。一个 Java 应用程序就是多个类的集合, 类由关键字 class 声明, 类中包含成员变量和成员方法, main()方法是类首先执行的方法, 语句必须写在类的方法中。

【例 1.1】 显示字符串的 Application 应用程序。

本例演示基于控制台运行的 Application 应用程序, 没有图形用户界面。程序如下:

```
public class Hello
{
    public static void main(String args[])
    {
        System.out.println("Hello!");
    }
}
```

Java 程序所有语句都必须写在类中, class 是类声明的关键字, Hello 是类名。在 main()方法中, 用标准输出语句 System.out.println()在屏幕上显示一个字符串“Hello!”。文件名为“Hello.java”。

2. Applet

Applet 是可以嵌入 Web 页面的最小应用, 它不能独立运行, 必须嵌入到超文本 (*.html)中, 由浏览器中的 Java 解释器解释执行。Applet 只是 Application 的特殊形式, 要编写 Applet, 必须知道如何编写 Application。

【例 1.2】 显示字符串的 Applet 应用程序。

本例演示 Applet 应用程序。显示一个字符串“Hello!”的 Applet 程序如下：

```
import java.awt.*;
import java.applet.Applet;
public class HelloApplet extends Applet
{
    public void paint(Graphics g)
    {
        g.setColor(Color.red);
        g.drawString("Hello!",20,20);
    }
}
```

1.2.2 Java 虚拟机执行 Java 程序

Java 是一种编译语言（compiled language），但它不直接生成硬件处理器的指令，而是生成一种与操作系统平台是无关的字节码指令，这种字节码由 Java 虚拟机解释执行。Java 虚拟机

是一套支持 Java 语言运行的软件系统，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，提供了跨平台能力的基础框架。Java 虚拟机的解释器在得到字节码后，对它进行转换，使之能够在不同的平台上运行。Java 虚拟机执行 Java 程序过程如图 1.1 所示。

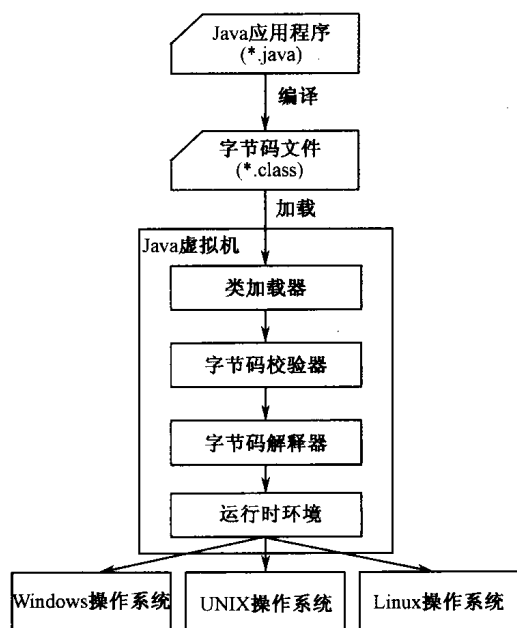


图 1.1 Java 虚拟机执行 Java 程序过程

Java 虚拟机执行应用程序具有以下 3 个典型特点：

① 动态性——Java 虚拟机在内存管理等功能方面全部采用动态方式，如动态分配数组的存储空间、动态创建对象、动态连接数据库等，这些动态特性使 Java 程序适合在网络上运行。

② 异常处理——Java 虚拟机提供可靠的异常处理机制。

③ 多线程——Java 虚拟机采用多线程方式运行，各线程独立地执行 Java 代码，并能够协调一致地处理共享数据。

1.3 JDK

JDK 是 Sun 公司推出的 Java 开发工具包，包括 Java 类库、Java 编译器、Java 解释器、Java 运行时环境和 Java 命令行工具。JDK 提供 Java 程序的编译和运行命令，但没有提供程序编辑环境。其他一些 Java 开发工具提供集成开发环境(Integrated Development Environment, IDE)，如 Eclipse、MyEclipse、JBuilder 等，它们都是建立在 JDK 运行环境之上的。