



普通高等教育“十一五”国家级规划教材

Java程序设计 实用教程（第4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

Java 程序设计实用教程

(第4版)

叶核亚 编著

陈道蓄 主审

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

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、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,输入/输出流和文件操作,网络通信,数据库应用,Web 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本科培养目标的要求。

本书可作为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,或使用 Java 语言从事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. —4 版. —北京:电子工业出版社, 2013.11

ISBN 978-7-121-21733-3

I. ① J… II. ① 叶… III. ① JAVA 语言—程序设计—高等学校—教材 IV. ① TP312

中国版本图书馆 CIP 数据核字 (2013) 第 250414 号

策划编辑: 章海涛

责任编辑: 章海涛

特约编辑: 曹剑锋

印 刷: 北京季峰印刷有限公司

装 订: 北京季峰印刷有限公司

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: 24.75 字数: 680 千字

印 次: 2013 年 11 月第 1 次印刷

定 价: 44.00 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,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@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4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、跨平台、多线程以及强大的网络编程功能等特点。今日 Java 应用无处不在，Java 语言作为程序设计的首选语言，其重要性毋庸置疑。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“理论基础厚实、全面展现应用技术、注重实践环节”为编写目标，力求探索出一套适合工科院校计算机及相关专业的教学方案，确立 Java 语言作为专业主干课程的重要地位。

本书具有以下特点。

（1）理论基础厚实

本书全面、系统地介绍构成 Java 应用程序的基本要素和必备知识，包括 Java 语言跨平台的运行机制、Java 语言的语法基础和面向对象基本概念，着重介绍类的封装、继承和多态等面向对象的核心特性，以及接口、内部类、包、异常处理等 Java 语言特有的实现机制。章节之间有机衔接，前后呼应。全书结构安排合理，由浅入深，层次分明，内容涉及的广度和深度符合本科培养目标的要求。

（2）全面展现应用技术

本书介绍 Java 语言的图形用户界面、多线程、输入/输出流和文件操作、网络通信、数据库应用、Web 应用等多种实用技术。这些知识和技术与现实世界联系紧密，实用性较强，学生易于理解，但实现起来较困难。本书以面向对象理论为基础，以广阔的实际应用为背景，采用一个个贴近生活实际的实例展现 Java 语言的应用技术，展现面向对象思想的作用和使用方法，说明为什么 Java 语言能够在网络环境中被广泛应用，Java 语言的哪些机制使其具有独特的魅力，从而能够更深入地理解面向对象思想的精妙。

（3）注重实践环节

“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。实践性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。

注重理论知识和实用新技术相结合，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。本书将程序设计能力的锻炼和提高设计为一个循序渐进的过程，从原理叙述、例题、

思考题等课堂讲授环节，到课后习题、上机实验、课程设计等实践性环节，经历先见识、再模仿、最后自主创新设计的学习过程，并培养良好的程序设计习惯。

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，本书给出了综合应用程序设计实例，详细说明需求方案、设计目标、设计任务、模块划分、功能实现、调试运行等环节的设计方法，贯彻了理论讲授和案例教学相结合的教学方法，训练学生既具有扎实深厚的基本功，又具有可扩展素质，创新能力强。

本书采用的运行环境包括 JDK 7、MyEclipse 8.5、MySQL 5.6 数据库及 Tomcat 7.0。

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。

本书第 1 版于 2003 年出版，岁月如梭，转眼已十年。感谢电子工业出版社十年来对我的坚定支持；感谢陈老师认真细致地审阅全稿；感谢我的同事们提供了许多帮助；感谢众多读者朋友的坚定支持及提出的宝贵意见。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 yeheya@x263.net 索取。

作 者

目 录

第 1 章 Java 概述.....	001	2.3 数组.....	040
1.1 了解 Java.....	001	2.3.1 一维数组.....	040
1.1.1 Java 的诞生和发展.....	001	2.3.2 二维数组.....	043
1.1.2 Java 的特点.....	002	2.4 静态方法.....	046
1.1.3 Java 核心技术.....	005	2.4.1 方法声明与调用.....	046
1.1.4 Application 应用.....	005	2.4.2 参数传递.....	047
1.2 JDK.....	006	2.4.3 方法重载.....	051
1.2.1 JDK 的安装与设置.....	006	2.4.4 递归方法.....	051
1.2.2 编译和运行 Java 程序.....	008	2.5 字符串.....	052
1.2.3 包.....	009	2.5.1 字符串的基本数据类型特性.....	052
1.3 MyEclipse.....	013	2.5.2 字符串的类特性.....	054
1.3.1 MyEclipse 集成开发环境.....	013	习题 2.....	058
1.3.2 创建 Java 项目并运行.....	015	实验 2 Java 程序设计基础.....	059
1.3.3 程序调试技术.....	018	第 3 章 类的封装、继承和多态.....	061
习题 1.....	021	3.1 类和对象.....	061
实验 1 Java Application 程序的编辑、编译 和运行.....	021	3.1.1 类.....	061
第 2 章 Java 语言基础.....	022	3.1.2 对象.....	063
2.1 语言成分.....	022	3.2 类的封装性.....	065
2.1.1 标识符与关键字.....	022	3.2.1 构造与析构.....	065
2.1.2 基本数据类型.....	023	3.2.2 对象的引用和运算.....	067
2.1.3 变量与常量.....	025	3.2.3 访问控制.....	068
2.1.4 运算符与表达式.....	026	3.2.4 静态成员.....	069
2.2 流程控制语句.....	032	3.2.5 浅拷贝与深拷贝.....	074
2.2.1 顺序结构语句.....	032	3.3 类的继承性.....	076
2.2.2 选择语句.....	033	3.3.1 由继承派生类.....	076
2.2.3 循环语句.....	034	3.3.2 继承原则及作用.....	077
2.2.4 转移语句.....	039	3.3.3 子类的构造方法.....	079
		3.4 类的多态性.....	080

3.4.1 子类重定义父类成员	080
3.4.2 类型的多态	083
3.4.3 编译时多态和运行时多态	085
3.4.4 多态的方法实现	085
3.5 类的抽象性	090
3.5.1 用继承刻画包含关系	091
3.5.2 抽象类	091
3.5.3 最终类	095
习题 3	096
实验 3 类的封装、继承和多态	098

第 4 章 接口、内部类和 Java API

基础	100
4.1 接口与实现接口的类	100
4.2 内部类和内部接口	105
4.3 Java API 基础	108
4.3.1 java.lang 包中的基础类库	108
4.3.2 java.util 包中的工具类库	113
4.4 泛型	117
习题 4	120
实验 4 接口与实现接口的类	120

第 5 章 异常处理

5.1 异常处理基础	122
5.1.1 异常处理机制的必要性	122
5.1.2 错误和异常	123
5.2 异常处理措施	125
5.2.1 异常处理语句	126
5.2.2 抛出异常	130
5.2.3 定义异常类	130
习题 5	132
实验 5 异常的抛出、捕获并处理	132

第 6 章 图形用户界面

6.1 AWT 组件及其属性类	134
-----------------------	-----

6.1.1 AWT 组件	134
6.1.2 布局管理	138
6.1.3 颜色和字体	140
6.2 事件处理	141
6.2.1 委托事件模型	141
6.2.2 AWT 事件类和事件监听器接口	144
6.3 Swing 组件及事件	146
6.3.1 Swing 组件与布局	146
6.3.2 文本显示和编辑组件及事件	148
6.3.3 按钮组件	152
6.3.4 列表框和组合框	153
6.3.5 中间容器	155
6.3.6 JOptionPane 和 JColorChooser 对话框	156
6.3.7 菜单组件	162
6.3.8 表格	171
6.4 图形图像	174
6.4.1 图形设计	174
6.4.2 图像	179
习题 6	179
实验 6 图形用户界面设计	180

第 7 章 多线程

7.1 操作系统中的进程与线程	184
7.2 Java 的线程对象	186
7.2.1 Runnable 接口与 Thread 类	187
7.2.2 线程对象的优先级	190
7.2.3 线程对象的生命周期	190
7.2.4 定时器与图形动画设计	195
7.3 线程的同步机制	197
7.3.1 交互线程	197
7.3.2 线程间的竞争关系与线程互斥	200
7.3.3 线程间的协作关系与线程同步	203
习题 7	211

实验 7 线程设计	212	9.4.1 UDP 数据报	275
第 8 章 输入/输出流和文件操作	213	9.4.2 UDP 组播数据报	278
8.1 文件和流	213	习题 9	283
8.1.1 操作系统中的文件和目录概念	213	实验 9 网络通信	283
8.1.2 流的概念	215	第 10 章 数据库应用	284
8.2 字节输入/输出流类	216	10.1 关系数据库系统	284
8.2.1 抽象字节流	216	10.1.1 关系模型	284
8.2.2 文件字节流	218	10.1.2 客户-服务器结构的关系数据库 系统	286
8.2.3 数据字节流	221	10.1.3 结构化查询语言 SQL	287
8.2.4 对象字节流	225	10.2 MySQL 数据库	292
8.2.5 管道字节流	227	10.3 JDBC	294
8.3 字符输入/输出流类	230	10.3.1 JDBC 的作用和功能	294
8.3.1 抽象字符流	230	10.3.2 指定 JDBC 驱动程序	296
8.3.2 字节/字符转换流	231	10.3.3 连接数据库	297
8.3.3 文件字符流	231	10.3.4 执行 SQL 语句	298
8.3.4 缓冲字符流	232	10.3.5 处理数据查询结果集	299
8.3.5 格式化字符输出流	232	习题 10	303
8.3.6 Java 标准输入/输出	234	实验 10 数据库应用设计	304
8.4 文件操作	237	第 11 章 Web 应用	305
8.4.1 文件类及其过滤器	237	11.1 Web 浏览基础	305
8.4.2 文件选择对话框组件	241	11.2 Applet 应用程序	307
8.4.3 随机存取文件类	249	11.3 JSP	309
习题 8	253	11.3.1 JSP 原理	309
实验 8 输入/输出流与文件操作	253	11.3.2 运行 JSP	311
第 9 章 网络通信	255	11.3.3 JSP 语法	315
9.1 网络编程基础	255	习题 11	323
9.2 使用 URL 访问网络资源	257	实验 11 基于 JSP 的 Web 应用设计	324
9.2.1 URL 和 IP 地址	257	第 12 章 综合应用设计	326
9.2.2 使用选项卡窗格和编辑器窗格	258	12.1 集合框架	326
9.3 TCP Socket 通信	262	12.2 反射	331
9.3.1 TCP Socket 通信原理	262	12.3 使用复杂 Swing 组件	331
9.3.2 Java 的 TCP Socket 通信	264		
9.4 UDP 数据报通信	274		

12.3.1 列表框.....	331	附录 D Java 运算符及其优先级	378
12.3.2 表格.....	334	附录 E Java 类库（部分）	379
12.3.3 多文档界面.....	335	E.1 java.lang 语言包	379
12.3.4 树	339	E.2 java.util 实用包	383
12.4 数据库应用	354	E.3 java.text 文本包	384
12.5 课程设计的要求和选题.....	368	附录 F MyEclipse 常用菜单命令	385
附录 A ASCII 字符与 Unicode 值	374	参考文献	387
附录 B Java 关键字.....	375		
附录 C Java 基本数据类型.....	377		

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。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。

本章简要介绍 Java 特点和核心技术；介绍 Java Application 应用程序的基本形式，以及由虚拟机支持的程序运行机制；以 Windows 平台的 Java SE7 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 MyEclipse 集成开发环境中编辑、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

1991 年，Sun Microsystems 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、容易联网和跨平台等特性，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，保留了 C++ 语言的语法，同时放弃了一些具有潜在危险特性的内容，如资源引用、指针、运算符重载等。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代码应用到微波炉或其他电器上，而不必改变软件，这显著降低了开发成本。不仅如此，Oak 还适合于 Internet 编程。1994 年，Green 项目组用 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 JDK 1.0 和 HotJava。HotJava 通过嵌入在 Web

网页中的 Applet 运行 Java 程序，一年之内，Microsoft 公司和 Netscape 公司的 Web 浏览器宣布支持 Applet，IBM、Apple、DEC、Adobe、Silicon Graphics、HP、Oracle 和 Microsoft 等公司相继购买 Java 技术许可证。从此，Java 成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)。得益于跨平台特性，Java 2 不仅能够应用于智能卡和小型消费类设备，还能够应用于大型服务器系统；它提供的接口机制使得软件开发商、服务提供商和设备制造商能够紧密配合，降低了软件开发和维护的工作量。

Sun 公司采取开放性策略，在其网站上可以免费获取 JDK，这也是 Java 语言能够迅速发展的一个重要因素。不同的操作系统平台需要使用不同版本的 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，Sun 发布 J2SE 1.5，自此 J2SE 1.5 更名为 J2SE 5.0。2005 年，发布 Java SE6，并取消 Java 2 名称，Java 三大平台被更名为 Java EE、Java SE、Java ME。

三大平台说明如下。

① Java ME (Java Micro Edition) 是适用于小型设备和智能卡的 Java 嵌入式平台，提供智能卡 (Java Card) 业务、移动通信 (Java Telephony)、电视机顶盒 (Java TV) 等智能电器控制功能。

② 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 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规范，具有开放性、可扩展性、集成性以及 Java EE 服务器之间的互操作性。随着 Web 应用技术日益成熟成为展示和操纵数据的主流技术，Java EE 在 Web 应用技术开发过程中表现出愈来愈显著的优势，成为开发电子商务应用的事实标准。

2009 年，Oracle 公司收购 Sun Microsystems 公司。2011 年 7 月 Oracle 公司发布 JDK 7。

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，其地位已被肯定。它在如此短暂的历史过程中，经历如此规模的发展壮大，显然并不是偶然的，而是有着内在的基础和外在的机遇。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，它所具备的跨平台特性、面向对象和可靠性、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。不仅于网络应用，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 技术是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. 跨平台特性

跨平台特性，也称平台无关性，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”。跨平台特性使 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跨平台特性。Java 虚拟机（Java Virtual Machine, JVM）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 虚拟机运行于操作系统之上。

C++等其他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）经编译生成的是字节码文件（*.class），也称类文件，它由 Java 虚拟机解释执行，解释一句，执行一句。Java 虚拟机执行 Java 程序过程如图 1.1 所示。Sun 公司为各种操作系统配有相应的 Java 虚拟机，因此，Java 源程序能够跨平台运行在不同操作系统之上，无须移植。

Java 虚拟机执行应用程序具有以下 3 个特点：

① 动态性。管理功能全部采用动态方式，如动态分配数组的存储空间、动态创建对象、动态链接数据库等，使得 Java 程序适合在网络上运行。

② 异常处理。提供可靠的异常处理机制。

③ 多线程。采用多线程方式运行，各线程独立执行，并协调一致地处理共享数据。

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

面向对象是当前软件开发的先进技术和重要方法。面向对象的概念是基于信息隐藏和数据抽象类型的概念，利用类和对象的机制将数据和方法封装在一起，通过统一的接口与外界交互；通过类的继承机制实现代码重用。面向对象方法反映了客观世界中现实的实体在程序中的独立性和继承性。这种方法有利于提高程序的可维护性和可重用性，还有利于提高软件开发效率和程序的可管理性。

Java 语言从 C++语言发展而来，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++语言中一些含义模糊、过于复杂、安全性差、不适合网络应用的规则，但没有放弃与此相关的功能。Java 语言采用更简单、功能更强、性能更好的方式代替。

Java 语言对 C/C++语言的基本语法改进说明如下。

- ① 不支持全局变量和宏替换，使用最终变量代替宏替换，避免全局变量和宏替换的副作用。
- ② 为每种数据类型分配固定长度，实现数据类型的平台无关性。
- ③ 进行类型相容性检查，防止不安全的类型转换。
- ④ 不支持 goto 语句。
- ⑤ 不支持指针类型，通过引用模型实现了指针的功能。
- ⑥ 不支持结构类型，使用类代替；不支持联合类型。
- ⑦ 不支持头文件，用 import 语句声明导入指定包中的类或接口。
- ⑧ 内存动态存储且自动管理，动态申请数组和对象的存储空间，自动释放空间，没有指针操作方式。

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。Java 语言对 C/C++语言的面向对象机制改进说明如下。

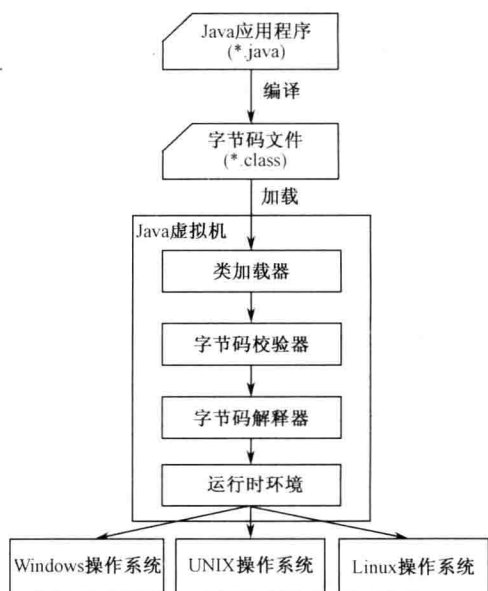


图1.1 Java 虚拟机执行 Java 程序过程

① 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，体现完全面向对象。

② 将数组设计为引用类型，每个数组都有长度属性。

③ 不支持类似 C 语言那样的面向过程设计，不支持全局函数，所有函数都是类中成员方法；方法参数不支持默认值形式，避免因默认值造成的二义性；成员方法内不能用 `static` 声明局部变量。

④ 不支持友元类和运算符重载，因为友元破坏封装性。

⑤ 提供单继承机制，即一个类只有一个父类，这样使得所有类（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。子类不能继承父类的构造方法，但可以继承析构方法；所有成员方法都可在运行时被覆盖，都是 C++ 含义的虚方法；不支持多继承，提供接口，通过“单继承+接口”方式实现多继承功能。

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；通过成员方法实现类的操作，则不需要重载运算符等。因此，放弃结构、指针、多继承等，并没有影响 Java 语言的功能，Java 提供的机制具有更强的功能和更高的性能。

3. 可靠性

C++ 语言在稳定性和可靠性方面最大的隐患是使用指针和内存缺乏自动管理。

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，`boolean` 与 `int` 类型数据不能进行运算；数组下标不能越界，避免有效数据被覆盖；等等。

Java 语言提供异常处理机制，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，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 语言自己操纵内存减小了内存出错的可能性，减少了程序员的工作量，提高了程序运行的可靠性。

4. 安全性

Java 语言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，更好地支持企业级应用，同时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 语言将进行严格的代码安全性检测，限制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。多线程是实现 Socket 通信的技术基础。

6. 支持分布式网络应用

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 页面具有动态特性,提高了 Web 页面的智能化和灵活性。目前, JSP (Java Server Pages) 提供服务端的动态 Web 页面设计,功能更强,效率更高。

Java 支持多种协议的 URL 寻址,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相同。Java 支持 Socket 网络通信方式。Java 核心类库包括一个支持 HTTP 等基于 TCP/IP 协议的类库,为分布式环境尤其是 Internet 提供动态内容。

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 Java 应用程序与关系型数据库连接规范,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 服务端的标记语言,它将 Java 语言嵌入到 HTML 文档中。JSP 是基于 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 JavaBean 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 JavaBean。EJB (Enterprise JavaBean) 提供企业级的 JavaBean。

④ JavaMail 提供 E-mail 邮件服务的支持类库,支持 SMTP 和 POP3 服务器。

1.1.4 Application 应用

Java 应用程序有两种形式: Application 和 Applet。

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。

Applet 是可以嵌入 Web 页面的最小应用,它不能独立运行,必须嵌入超文本 (*.html) 中,由浏览器中的 Java 解释器解释执行。Applet 是 Application 的特殊形式,详见第 11 章。

【例 1.1】 接受命令行参数的 Application 应用程序。

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

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[])
    {
        if (args.length==0)                //没有命令行参数时
            System.out.println("Hello!");
        else                                //显示命令行参数字符串
        {
            for (int i=0; i<args.length; i++)
                System.out.println(args[i]);
        }
    }
}
```

```

    }
}
}

```

其中, class 是声明类的关键字, Hello 是类名, 必须符合标识符语法, 约定类名首字母大写。文件名为 “Hello.java”, 文件名与类名必须相同。main(String args[])方法的参数 args 是字符串数组, 它接受命令行参数。System.out 是标准输出常量, 调用 print()或 println()重载方法, 可将基本数据类型、字符串数组、字符串及对象等各数据类型参数值转换成字符串输出。

Java 源程序文件 (*.java) 通过编译器编译生成字节码文件 (*.class), 再由解释器执行, 运行流程如图 1.2 所示。

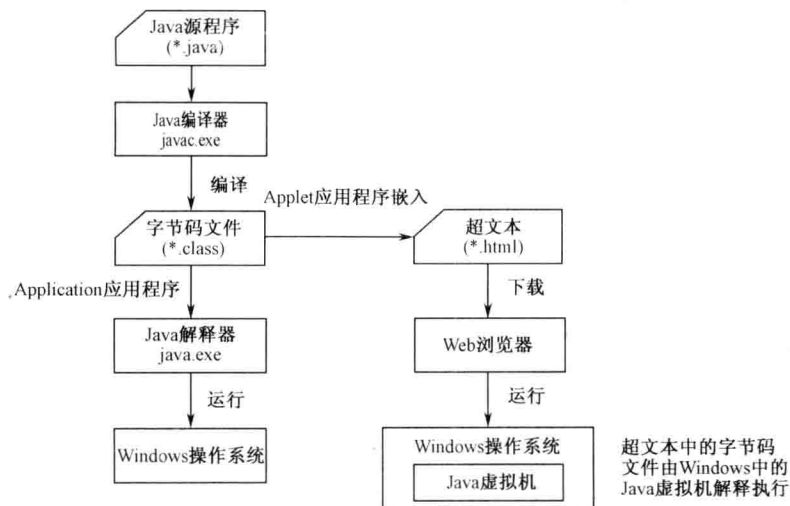


图 1.2 Java 程序运行流程

1.2 JDK

JDK 是 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。Eclipse、MyEclipse、JBuilder 等 Java 开发工具提供集成开发环境 (Integrated Development Environment, IDE), 它们依赖 JDK 提供运行环境。

1.2.1 JDK 的安装与设置

1. 安装 JDK

从 Oracle 公司网站 <http://www.oracle.com/technetwork/java> 下载以下两个文件:

- ◎ jdk-7u7-windows-i586.exe, 32 位 Windows 的 JDK 7 安装程序。
- ◎ jdk-7u7-apidocs.zip, JDK 7 文档。

运行安装文件, 安装过程中可以设置安装路径及选择组件, 系统默认安装路径为 C:\Program Files\Java\jdk1.7.0_07, 默认组件选择是全部安装。⚠注意: 默认安装路径中包含当前版本信息。

安装成功的 JDK 目录结构如图 1.3 所示。

JDK 的目录结构说明如表 1-1 所示, bin 中包括的主要工具如表 1-2 所示。

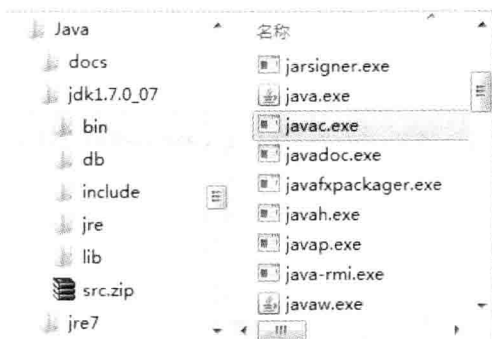


图 1.3 JDK 的目录结构

表 1-1 JDK 的目录结构

文件/目录名	功 能 说 明	文件/目录名	功 能 说 明
src.zip	核心 API 所有类的源文件	include	编写 JNDI 等程序需要的 C 语言头文件
bin	包含编译器、解释器等可执行文件	jre	Java 运行时环境
demo	包含源代码的程序示例	lib	Java 类库

表 1-2 bin 中的主要工具

文件名	功 能 说 明
javac.exe	Java 编译器，将 Java 源程序编译成字节码文件
java.exe	Java 解释器，执行字节码文件对应的 Java 类
appletviewer.exe	Applet 应用程序浏览器
javadoc.exe	根据 Java 源码及说明语句生成 HTML 文档
jdb.exe	Java 调试器，可以逐行执行程序，设置断点和检查变量
jar.exe	压缩文件，扩展名为 JAR（Java Archive、Java 归档），与 Zip 压缩文件格式相同

2. 设置环境变量

由于 Java 是平台无关的，安装 JDK 时 Java 不会自动设置路径，也不会修改注册表，需要用户自己设置环境变量，但不需要修改注册表。

在 Windows 中需要设置 path 和 classpath 两个环境变量。path 变量指出可执行文件路径，classpath 变量指出 Java 包的路径。以下介绍设置环境变量的批命令，创建一个批命令文件 jdk7.bat，添加两行设置命令如下：

```
set path=%path%;C:\Program Files\Java\jdk1.7.0_07\bin
```

```
set classpath=.;C:\Program Files\Java\jdk1.7.0_07\lib
```

设置环境变量需要注意的事项如下：

① C:\Program Files\Java\jdk1.7.0_07 是 JDK 的默认安装路径，环境变量的值需要根据实际的安装路径而更改。Windows 系统不区分字母大小写，因此变量名和路径字符串中字母，大小写均可。

② path 值中的“%path%”表示 path 的原有路径，可以省略。如果省略，同时省略“;”，此时 path 原路径将不复存在，会影响其他程序运行。

③ 分号“;”是两个路径之间的分隔。变量值中不能有多余的分号或空格。

④ classpath 值中的“.”表示当前目录，通常写在最前面，作为系统查找类的第一个路径。

也可在 Windows “系统属性”对话框中设置 path 和 classpath 环境变量。

1.2.2 编译和运行 Java 程序

1. 执行批命令设置环境变量

执行“开始 ▸ 程序 ▸ 附件 ▸ 命令提示符”，打开 MS-DOS 窗口，输入以下命令：

```
C:\>D:                      //D: 转换盘符
D:\>cd myjava                //进入 myjava 文件夹
D:\myjava>jdk7               //执行批命令文件 jdk7.bat
```

执行批命令 jdk7.bat，完成 path 和 classpath 变量的路径设置，如图 1.4 所示，设已创建 D:\myjava 文件夹，且 jdk7.bat 在该文件夹中。其中，“>”是 DOS 提示符，每行“>”之前的是 DOS 显示的当前路径，“>”之后的是输入的 DOS 命令。

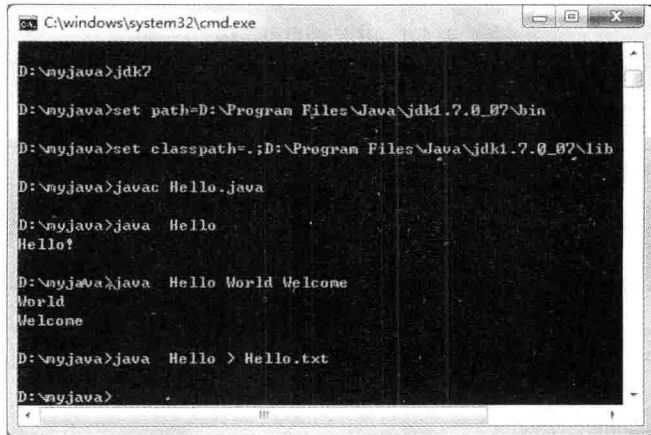


图 1.4 设置环境变量和编译、运行 Application 应用程序

2. 编译

设采用文本编辑器已将例 1.1 的 Java 源程序编辑为一个源程序文件 D:\myjava\Hello.java，文件名与类名 Hello 相同。执行编译命令 javac.exe 对 Hello.java 文件进行编译，在 MS-DOS 窗口中，输入编译命令如下（见图 1.4）：

```
D:\myjava>javac Hello.java
```

如果系统未找到 javac.exe 命令，说明 path 环境变量设置不正确；如果程序中有语法错误，系统将终止编译并给出错误信息；如果编译正确，将生成字节码文件 Hello.class。

3. 运行 Application 应用程序

对于 Application 应用程序，执行命令 java.exe 可运行字节码文件 Hello.class，命令如下，其中，省略字节码文件的后缀（.class），文件名首字母仍然大写，运行结果见图 1.4。

```
D:\myjava>java Hello
D:\myjava>java Hello > Hello.txt           //将运行结果写入 Hello.txt 文本文件中
```

4. 命令行参数

命令行参数是指运行时跟在执行命令后作为输入数据的多个字符串，它是控制台应用程序的一种数据输入方式。

带参数的运行命令如下，多个字符串参数以空格分隔，运行结果见图 1.4。如果没有命令行参数，则数组长度 args.length 的值为 0。

```
D:\myjava>java Hello World Welcome
```