

编程思想 · 上机实战

Java 2

实用教程

杜江 编著



Java 2 实用教程

杜 江 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书全面地讲解 Java 的基础，从 Java 的环境配置到 Java 的 Web 应用。内容包括面向对象编程、Java 语言基础、流程控制语句、类和对象、包、接口与异常、线程、输入与输出、Java 的网络编程、Java 图形编程、Applet 编程、JDBC 编程、Web 应用编程等 13 章内容。每一部分都有详细的实例，让读者更加深刻地掌握所学内容。本书提供的下载文件包括 PowerPoint 课件和实例源代码。

本书语言简洁，实例丰富，适合 Java 初学者和有一定编程经验的 Java 程序设计者。

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 2 实用教程 / 杜江编著. —北京: 电子工业出版社, 2006.3

ISBN 7-121-02209-5

I. J… II. 杜… III. JAVA 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 001004 号

责任编辑: 毕 宁 bn@phei.com.cn

印 刷: 北京智力达印刷有限公司

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

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 20 字数: 463 千字

印 次: 2006 年 3 月第 1 次印刷

印 数: 6000 册 定价: 29.00 元

凡购买电子工业出版社的图书，如有缺损问题，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。质量投诉请发邮件至 zllts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

Java 是一种面向对象程序设计语言。面向对象技术提倡模拟现实世界的思维方式、数据与操作相捆绑的程序风格，符合现代大规模软件开发的要求和潮流，正逐步替代面向过程的程序设计技术，进而成为计算机应用开发领域的主流趋势。本书将以 Java 语言为基础，介绍面向对象编程设计的主要原则和方法。

本书将重点介绍 Java 的基本知识，通过这些基本的语法讲解，使读者了解面向对象的编程思想。本书内容包括：

第 1 章，Java 概述，讲解 Java 的历史，JDK 的安装和在不同操作系统上如何配置开发环境，以及 Java 常用的几个工具。

第 2 章，Java 语言基础知识，包括基本数据类型、常量和变量、Java 的保留关键字、Java 的运算符、Java 的注释方法、数组的运用等知识。

第 3 章，流程控制语句，讲解 Java 的基础知识，包括 if-else 条件选择语句、switch 分支语句、for 和 while 循环控制语句、跳转语句等知识。

第 4 章，类和对象，讲解面向对象的特点和面向对象的编程思想。

第 5 章，类、接口和包，从面向对象思想讲解什么是 Java 的类、接口和包，怎样创建自己的类、接口和包。

第 6 章，线程，讲解如何实现线程和创建多线程。

第 7 章，输入与输出，介绍 Java 语言的输入/输出 (I/O) 操作。

第 8 章，Java 的网络编程，主要讲解 Socket 基础知识，包括 Socket 的类型和创建，以及如何创建服务器端和客户端程序。

第 9 章，Java 图形用户界面，主要讲解 Swing 的基础知识，包括 Swing 组件、容器和事件处理。

第 10 章，Applet 编程，介绍如何使用 Applet 运行一个 Java 程序，如何使用 HTML 给 Applet 传递参数。

第 11 章，JDBC 编程，主要讲解什么是 JDBC，JDBC 的创建和如何使用 JDBC 连接数据库，如何使用 JDBC 执行 SQL 语句，即对获取的结果集如何处理。

第 12 章，Web 应用编程，主要介绍如何使用 Servlet 和 JSP 技术。

第 13 章，Java 开发工具 NetBeans 的使用，主要介绍 NetBeans IDE 4.1 开发工具的使用方法。

本书的实例源代码与 PPT 课件可以在 <http://www.broadview.com.cn/html/resource/Java2.rar> 下载。

本书编写过程中，以下人员给予了很多帮助，包括张勇、薛年喜、李志伟、刘旭、赵海

云、宋利军、刘咏、郑砚、许社村、黄丽娜、唐兵、李龙、李子婷、尹建民、刘伟、李琪、屈昕等。在此一并表示感谢。

由于作者水平有限，书中难免不妥之处，希望读者原谅，并提出宝贵意见，以便再版时更加完善。读者与作者技术交流，可上书友论坛 <http://forum.broadview.com.cn>，意见反馈请发邮件至 hxz@phei.coml.cn。

编 者
2006年1月

目 录

第 1 章 Java 概述 1	第 3 章 流程控制语句32
1.1 Java 技术概述.....1	3.1 条件选择语句.....32
1.2 Java 语言的特点.....3	3.2 switch 分支语句.....34
1.3 面向对象的程序设计思想.....4	3.3 循环控制语句.....37
1.3.1 什么是对象.....5	3.3.1 while 语句.....37
1.3.2 OOP 的要素.....5	3.3.2 do-while 语句.....39
1.4 JDK 的安装和测试.....6	3.3.3 for 语句.....40
1.5 简单的 Java 程序.....8	3.4 跳转语句.....42
1.6 Java 程序开发步骤.....9	3.4.1 break 语句.....42
1.7 思考与练习.....11	3.4.2 continue 语句.....44
第 2 章 Java 语言基础12	3.5 上机实战.....45
2.1 Java 的基本数据类型.....12	3.5.1 判断闰年.....45
2.2 常量和变量.....14	3.5.2 寻找完数.....46
2.2.1 常量.....14	3.5.3 构建魔方阵.....48
2.2.2 变量.....15	3.6 思考与练习.....50
2.3 Java 保留关键字.....16	第 4 章 类和对象51
2.4 运算符.....17	4.1 面向对象编程.....51
2.4.1 算术运算符.....17	4.1.1 面向过程.....51
2.4.2 关系运算符.....18	4.1.2 面向对象.....52
2.4.3 逻辑运算符.....18	4.2 类的描述.....54
2.4.4 位运算符.....19	4.2.1 类的定义.....54
2.4.5 赋值运算符.....20	4.2.2 对象实例.....55
2.4.6 条件运算符.....21	4.2.3 构造器.....56
2.5 表达式.....21	4.3 类的成员变量和方法.....57
2.6 Java 的注释.....22	4.3.1 成员变量的定义.....57
2.7 数组的使用.....23	4.3.2 成员变量的访问权限.....58
2.7.1 一维数组.....23	4.3.3 静态成员变量.....59
2.7.2 多维数组.....25	4.3.4 final 变量.....60
2.8 上机实战.....27	4.3.5 成员方法的定义.....60
2.8.1 基本数学运算.....27	4.3.6 形参和实参.....62
2.8.2 多维数组实例.....28	4.3.7 成员方法重载.....62
2.9 思考与练习.....31	4.3.8 静态成员方法.....64
	4.4 类的继承.....65

4.5	this 和 super 变量	67	7.1.1	I/O 流	111
4.6	抽象类	69	7.1.2	InputStream 类常用接口	113
4.7	上机实战	70	7.1.3	OutputStream 类常用接口	113
4.7.1	创建类	71	7.1.4	Reader 类常用接口	114
4.7.2	静态变量和方法的类	72	7.1.5	Writer 类常用接口	114
4.7.3	模板——抽象类的使用	74	7.2	常用的输入 / 输出流	115
4.8	思考与练习	77	7.2.1	标准输入 / 输出	115
第 5 章	包、接口和异常	79	7.2.2	操作目录和文件	116
5.1	包	79	7.2.3	文件流	120
5.2	接口	81	7.2.4	随机文件的访问	122
5.3	异常及其处理	83	7.3	过滤流	124
5.4	使用异常的正确方式	86	7.3.1	数据流	124
5.5	创建自定义的异常	87	7.3.2	使用 I/O 缓冲区	126
5.6	上机实战	88	7.4	字节流与字符流的转换	129
5.6.1	命令处理	88	7.5	Java 运行外部命令	130
5.7	思考与练习	91	7.6	上机实战	132
5.7.1			7.6.1	系统输入 / 输出实例	132
5.7.2			7.6.2	英文字典实例	135
第 6 章	线程	93	7.7	思考与练习	137
6.1	线程简介	93	第 8 章	Java 的网络编程	138
6.1.1	线程的生命周期	93	8.1	网络基础知识	138
6.2	线程的实现方法	95	8.1.1	TCP/IP 参考模型	139
6.2.1	继承 Thread 类	95	8.2	Socket 套接字	140
6.2.2	实现 Runnable 接口	96	8.3	Java 开发 TCP/IP 程序	140
6.3	线程的控制	97	8.4	多线程服务器	143
6.3.1	启动线程	97	8.5	数据报	147
6.3.2	使线程休眠	98	8.6	URL 资源	149
6.3.3	中断线程	100	8.7	上机实战	151
6.3.4	停止线程	101	8.7.1	查看本机 IP 实例	152
6.4	Java 的多线程实例	101	8.7.2	简单聊天工具	153
6.5	线程的同步与死锁	102	8.7.3	用 POP3 协议接收邮件	156
6.6	线程管理	103	8.8	思考与练习	159
6.7	ThreadLocal 问题	104	第 9 章	Java 图形用户界面	160
6.8	上机实战	105	9.1	Java 图形用户界面简述	160
6.8.1	线程控制实例	106	9.2	容器与基本控件	161
6.8.2	提高线程优先级实例	108	9.2.1	窗口	161
6.9	思考与练习	110	9.2.2	窗体	163
9.2.3			9.2.3	内部窗体	164
第 7 章	输入与输出	111			
7.1	输入 / 输出包	111			

9.2.4	面板	165	11.1.1	从 ODBC 到 JDBC	218
9.2.5	菜单	166	11.1.2	JDBC 的特点	219
9.2.6	弹出式菜单	169	11.1.3	JDBC 驱动程序	220
9.2.7	工具栏	171	11.2	JDBC 基本编程	221
9.2.8	按钮	174	11.2.1	数据库连接串	221
9.2.9	单行文本域	174	11.2.2	加载驱动和创建连接	222
9.2.10	多行文本域	176	11.2.3	执行 SQL 语句	224
9.2.11	密码域	177	11.2.4	结果集	226
9.2.12	标签	179	11.3	元数据	229
9.2.13	滚动条	179	11.3.1	数据库元数据	229
9.3	布局管理器	181	11.3.2	结果集元数据	230
9.3.1	流控制器	182	11.4	可滚动的结果集	232
9.3.2	栅格控制器	183	11.5	事务处理	236
9.3.3	栅格包控制器	184	11.6	上机实战	237
9.3.4	边界控制器	186	11.6.1	一个简单的通用数据库 连接类	237
9.3.5	卡片控制器	188	11.6.2	操作 MSSQL 数据库	239
9.4	上机实战	190	11.6.3	连接 Oracle 数据库	241
9.4.1	文本编辑器实例	190	11.6.4	连接 MySQL 数据库	242
9.5	思考与练习	197	11.7	思考与练习	244
第 10 章	Applet 编程	198	第 12 章	Web 应用编程	245
10.1	Applet 的基本知识	198	12.1	Tomcat 简介	245
10.1.1	Applet 的工作原理	198	12.1.1	Tomcat 的配置	246
10.1.2	Applet 类的主要方法	200	12.1.2	发布自己的 Web 应用	248
10.2	什么是 Appletviewer	200	12.2	Servlet 简介	248
10.3	Applet 实现	201	12.2.1	第一个 Servlet 程序	249
10.3.1	编写一个 Applet	201	12.2.2	获取表单	251
10.3.2	获取键盘事件	202	12.3	JSP 概述	254
10.3.3	捕获鼠标事件	204	12.3.1	JSP 语法概要	255
10.3.4	Applet 程序与 HTML 交互	206	12.3.2	JSP 获取表单	258
10.4	HTML 中的 Applet 标记	208	12.4	JavaBean 与 JSP	260
10.5	应用 JAR 包	209	12.5	上机实战	261
10.6	上机实战	210	12.5.1	Servlet 留言簿	261
10.6.1	跳动的球实例	210	12.5.2	JSP 数据库留言簿	266
10.6.2	Applet 使用 Swing 界面	214	12.6	思考与练习	272
10.7	思考与练习	217	第 13 章	开发工具: NetBeans 的使用	274
第 11 章	JDBC 编程	218	13.1	NetBeans 简介	274
11.1	JDBC 简介	218			

13.1.1	NetBeans 的优点 及获取.....	274	13.4	使用 NetBeans 开发 Applet 程序.....	294
13.1.2	NetBeans 的安装.....	275	13.5	使用 NetBeans 开发 Web 应用.....	296
13.2	使用 NetBeans 开发简单的 Java 程序.....	278	13.6	导入源代码工程.....	303
13.3	使用 NetBeans 开发 GUI 程序.....	285	13.7	思考与练习.....	304
			附录 A	NetBeans IDE 4.1 快捷键.....	306

第 1 章 Java 概述

本章要点

- Java 的特点
- 面向对象编程思想
- Java 运行环境的安装
- Java 程序的编译和运行

本章介绍 Java 的发展历史, Java 语言的特点, 面向对象的程序设计思想, Java 运行环境的安装, 以及 Java 程序的编译和运行。

1.1 Java 技术概述

1991 年 4 月, Sun 公司内部启动了一个 Green 研究计划, 旨在开发一个消费性电子产品的控制软件。当时研发小组使用的程序设计语言是 C++, 但是 C++ 太过复杂而且不够安全, 所以计划主持人 James Gosling 先生便以 C++ 作为基石, 另行开发了一套新的程序语言, 命名为 Oak。

但是, 后来他发现已有程序语言被命名为 Oak, 该计划的成员便重新讨论新程序语言的命名。当时他们在咖啡馆里喝着印度尼西亚爪哇岛出产的咖啡, 其中一个人灵机一动, 说叫 Java 怎样。其他人十分赞赏。于是就将新语言命名为 Java。

Java 自诞生以来发展迅猛, 成为现在最流行的开发语言之一。下面列出了 Java 发展的历史大事。

- 1995 年 5 月 23 日, Sun 公司在 Sun World'95 上正式发布 Java 程序设计语言和 HotJava 浏览器。
- 1995 年 8 月, Netscape 公司获得 Java 许可证。
- 1995 年 9 月, Sun 公司宣布将提供 Java 开发工具。
- 1995 年 10 月, Oracle 公司和 Borland 公司获得 Java 许可证。
- 1995 年 12 月, SGI 公司、Adobe 公司、IBM 公司、AT&T 公司和 Intel 公司获得 Java 许可证。
- Sun 和 Netscape 发布 JavaScript——一种基于 Java 的脚本语言, 可供非专业编程人员使用。
- Sun, Netscape 和 SGI 宣布缔结软件联盟, 共同开发 Java 和 Internet 三维和多媒体交互性工具。
- 1996 年 1 月, Sun 宣布成立新的业务部门——JavaSoft 部, 以开发、销售并支持基于 Java 技术的产品, 由 Alan Baratz 先生任总裁。Sun 推出 Java 开发工具包 JDK (Java

Development Kit) 1.0, 为开发人员提供编制 Java 应用软件所需的工具。

- Sun 推出 Java 数据库连接 JDBC (Java Database Connectivity) 数据库 API。
- 1996 年 3 月, 微软公司、NetManage 公司获得 Java 许可证。
- 1996 年 4 月, SCO 公司、苹果电脑公司获得 Java 许可证。
- Sun 宣布苹果电脑、惠普、日立、IBM、微软、Novell、SiliconGraphics、SCO、Tandem 等公司将把 Java 平台嵌入其操作系统中。
- 1996 年 5 月 29 日, Sun 在旧金山举行“第一届 JavaOne 开发者大会”, 业界人士踊跃参加。Sun 在大会上推出一系列 Java 平台新技术: 如 JavaMediaAPIs, CommerceAPIs, SecurityAPIs, ServletAPIs, ManagementAPIs 等, 以及 JavaRMI 和 JavaIDL, JavaOS, JavaBeans 和新的 HotJava 开发环境等。
- 1996 年 10 月 29 日, Sun 发布 Java 企业计算技术, 包括: JavaStation 网络计算机、65 家公司发布的 85 个 Java 产品和应用、7 个新的 Java 培训课程和 Java 咨询服务、基于 Java 的 Solstice 互联网邮件软件、新的 Java 开发者支持服务、演示 HotJavaViews 和 JavaTutor, 以及完成 JavaCardAPI 等。
- 1997 年 2 月 18 日, Sun 发布 JDK 1.1、Java 商贸工具包、JavaBeans 开发包及一系列 JavaAPI。
- 1997 年 4 月 2 日, Sun 在美国旧金山举行“第二届 JavaOne 开发者大会”, 有超过 1 万人参加。Sun 正式开展 100%纯 Java 计划。
- 1997 年 4 月, Sun 为企业计算推出全面的 Java 平台, 包括一种新技术 Enterprise Java Beans。它使用 Java 来突破建立端对端商务解决方案的复杂性。
- 1997 年 10 月 Sun 发布世界上第一个 Java 微处理器 microJava701 芯片的技术细节, 并在 1998 年下半年大批量生产该芯片。
- 1998 年, JDK 1.2 (称 Java 2) 发布, JFC/Swing ships 技术公布那年 JFC/Swing 被下载 500000 多次。
- 1999 年, Java 技术被分成 J2SE, J2EE 和 J2ME, Java HotSpot 执行引擎发布, JavaServer Pages (JSP) 技术公之于众, J2EE 平台发布, J2SE 平台的 Linux 版发布。
- 2000 年, JDK 1.3 发布, 支持 XML 的 Java APIs 发布, J2EE 授权给 ATG, BEA Systems, Gemstone, Silverstream 和其他几个公司。Java 开发者成员上升到 150 万。
- 2003 年, JDK 1.4 发布, 增加了许多新的特性。

目前 JDK 1.5 正在开发之中, 其测试版本 Sun 网站已经提供下载。如果没有特殊说明, 本书默认使用 JDK 1.4 作为 Java 程序的运行环境。

Java 的技术越来越庞大, 越来越复杂, 使得 Java 已经不仅仅是一种编程语言, 而是正在形成的一门技术。它包括以下几个方面。

- Java 软件: Java 语言及其编程技术、Java 操作系统 (Java OS)、Java 开发环境及开发工具、数据库接口 JDBC、Java 语言的软件包等。
- Java 虚拟机及其规范。
- Java 嵌入技术: Java 芯片和基于 Java 技术的计算机系统。

Java 技术通常分为三大部分: Java 2 Platform Standard Edition (J2SE), Java 2 Platform

Enterprise Edition (J2EE), Java2 Platform Micro Edition (J2ME)。

J2SE 为桌面计算机或服务器提供一个完整开发的环境, 也为植入式开发提供环境。J2SE 是最基础的 Java 技术, 定义了一般的 Java 语言规范, 如 I/O、程序界面、多线程、网络编程等。

J2EE 技术为实现分布式企业开发定义了一个标准, 是目前网络开发中最热门的技术, 包括 JSP, Servlet, EJB, JavaMail, JNDI, RMI, RMI-IIOP 等。

J2ME 为嵌入式技术提供了一个开发运行环境。但这不是本书要介绍的内容。

1.2 Java 语言的特点

Java 能成为目前软件开发中最流行的语言, 与其自身特点分不开。Java 是一种简单的面向对象的分布式解释语言, 它健壮, 安全, 结构中立, 可移植, 性能优异, 多线程, 具有动态性。

1. 简单

Java 设计之初是为了简化开发, 并且面向不同的设备。Java 要充当小家用电器的编程语言, 用来解决诸如电视机、电话、闹钟、烤面包机等家用电器的控制和通信问题, 因此 Java 的结构和语言都简单明了。Java 语言的简单性主要体现在以下三个方面。

① Java 语言的风格类似于 C++, 因而 C++ 程序员对它非常熟悉。从某种意义上讲, Java 语言是 C 及 C++ 语言的一个变种, 因此, C++ 程序员可以很快掌握 Java 编程技术。

② Java 抛弃了 C++ 中容易引发程序错误的部分, 如指针和内存管理。删除了 C++ 中某些比较复杂的特性——Java 中没有结构联合, 抛弃了 C++ 的多重继承, 不提供运算符重载。

③ Java 提供了丰富的类库。Java 丰富的类库是让所有 C++ 程序员羡慕的。Java 的类库从标准的输入/输出到数据库的操作, 以及窗口动画, 应有尽有。有时 Java 编程如同搭积木, 将不同的模块通过接口连接起来组成一个完整的系统。而且很多构件是 Java 系统内部的, 而不需要程序员进行开发。

2. 面向对象

Java 是从 C++ 这个面向对象程序语言基础上发展形成的。它继承了 C++ 中面向对象的编程思想, 扩充了 C++ 中的面向对象模型, 因此 Java 语言的设计完全是面向对象的。面向对象是 Java 最重要的特性。Java 支持静态和动态风格的代码继承及重用。

3. 分布式

Java 类库包括了一个支持 HTTP 和 FTP 等基于 TCP/IP 协议的子库。因此, Java 应用程序可凭借 URL 访问方式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几乎完全相同。在分布式调用方面, Java 有许多技术, 例如 RMI (远程调用), EJB, CORBA 等。这些技术从高层次对分布式调用进行封装, 从而简化编程。

4. 健壮

Java 致力于检查程序在编译和运行时的错误。在编译时, Java 编译器就能对源程序进行语法检查, 降低了开发早期出现的错误。Java 还有一套完善的异常处理机制, 对发生的错误

进行捕获并做出相应的处理。而且，Java 运行程序不需要关心内存的分配，避免了由于程序对内存直接操作造成的内存泄漏事件。Java 还实现了数组对象，而不是简单的地址列表。这避免了覆盖数据的可能。这些功能和特征大大缩短了开发 Java 应用程序的周期。

5. 安全

作为一种网络编程语言，其安全性是起决定作用的。Java 作为一种应用语言，从根本上解决了语言的安全问题。Java 运行在虚拟机上，没有指针和内存操作，不直接对硬件操作。Java 程序上机执行前，要经过很多次的测试。Java 程序经过代码校验，检查代码段的格式，检测指针操作、对象操作是否过分。通过以上手段，大多数病毒无法破坏数据，因而，用 Java 可以构造出无病毒、安全的系统。

6. 可移植（跨平台）

可移植性一直是 Java 程序设计师们的精神指标，也是 Java 之所以能够受到程序设计师们喜爱的原因之一，最大的功臣就是 JVM 的技术。大多数编译器产生的目标代码只能运行在一种 CPU 上（如 Intel 的 x86 系列），即使那些能支持多种 CPU 的编译器也不能同时产生适合多种 CPU 的目标代码。如果你需要在三种 CPU（如 x86，SPARC 和 MIPS）上运行同一程序，就必须编译三次。

但 Java 编译器就不同了。Java 编译器产生的目标代码针对一种并不存在的 CPU——JAVA 虚拟机（Java Virtual Machine），而不是某一实际的 CPU。Java 虚拟机能掩盖不同 CPU 之间的差别，使目标代码能运行于任何安装有 Java 虚拟机的机器上。

Java 使用 Unicode 作为其标准字符，这项特性使得 Java 的程序能在不同语言的平台上撰写和执行。

7. 解释

Java 解释器（运行系统）能直接运行目标代码指令。链接程序通常比编译程序所需资源少，所以程序员可以在创建源程序上花更多的时间。

8. 多线程

单线程应用程序只能执行一个线程，一次只能做一件事，不会响应其他操作。单线程在实际应用中没有什么意义。多线程是操作系统的一种新概念，是在一个进程中存在可执行的多个小任务，是比传统进程更小的可并发执行的单位。多线程带来的好处是更好的交互性能和实时控制性能。当然实时控制性能还取决于操作系统本身（如 UNIX，Windows，Macintosh 等），开发难易程度比进程开发简单，但性能比进程好。

1.3 面向对象的程序设计思想

面向对象的程序设计（Object Oriented Programming，OOP）是目前主流的程序设计技术，它试图用描述客观世界中一个事物对象的方法来描述程序需要解决的问题。如果读者以前没有学过面向对象编程或者对面向对象编程还了解不太清楚，那么本节将让你完全了解什么是 OOP。

1.3.1 什么是对象

对象 (Object) 是 OOP 中最重要的概念, 清楚了对对象也就清楚了 OOP。简单说, 对象是一个客观实体的描述, 是既有数据又有数据操作的代码的集合体。

程序设计是一个人们对编程设计的认识过程, 经历过面向地址、面向过程和面向对象三个阶段。面向地址是编程语言完全模仿计算机的操作过程, 是最基本的机器语言。典型的语言是汇编。随着计算机的发展和人们对编程的认识, 出现了许多高级语言。这些面向过程的语言出现了一定的抽象和封装, 但其设计观点仍然停留在模拟计算机的操作过程之上。随着时间的推移, 面向对象的编程思想被提出, 这是一个质的飞越, 从根本上摆脱了程序设计要符合计算机操作的束缚, 而是首次从人类对事物对象的描述出发进行程序设计。

那么人类是如何来描述一个对象呢? 举个典型的例子。如果有人告诉你他种一棵树, 那么你对他种的树的第一反应就是有一根树干, 树干下是树根, 树干上有树枝, 树枝上长着绿色叶子。你的反应就是在描述一棵树的对象。在软件设计中可以将这些描述代码化, 如:

```
树 {  
    属性有:  
        树干  
        树根  
        树枝  
        树叶: 绿色  
    行为:  
        光合作用 ()  
}
```

接着, 如果他又告诉你他的树是一棵柳树, 那么你会进一步描述对柳树的描述。你的描述还是在树的基础之上。这就是对象继承性的特点。

现在代码化你的思想就是:

```
柳树 继承 树 {  
    树叶: 绿色、窄形  
}
```

这里有一个“继承”的概念, 简单地说继承就是拥有所有被继承对象的属性。整个描述过程不带有任何计算机特征, 这种描述就是面向对象的。

1.3.2 OOP 的要素

Java 是一种面向对象的程序设计语言, 完全支持面向对象程序设计。面向对象的程序设计包括以下几个要素。

1. 封装性

将描述对象的数据和对这些数据操作的代码有机组合成一个整体, 形成一个模块, 对其数据和模块进行有效的访问控制, 使模块完全独立。对象的这种特性称为封装性。Java 对对

象的封装是通过类来实现的，对象是类的一个实例。封装性非常有利于程序的调试和维护。

2. 继承和派生性

一个类可以派生出多个子类，派生出的类可以全部或者部分继承父类的数据和成员方法。这种特性称为继承和派生性。就如同柳树是树的派生类，柳树继承了树的特性。这种特性使得编程人员不用做太多的重复性工作。并且简化了编程，加快了程序设计，提高了工作效率。

3. 重载性

类的方法名可以根据不同的对象完成不同的功能或运算，这种特性称为重载。在 Java 中分为运算符重载和方法重载，本书会重点讲方法的重载。

4. 多态性

通过系统提供的机制，对象之间可以进行信息传递。按照一定格式传递的信息称为消息，相同的消息在不同的对象中可以导致完全不同的行为，这种特性称多态性。这种特性不仅提高了程序设计的灵活性，而且也减轻了类体使用的记忆负担。

1.4 JDK 的安装和测试

要使得 Java 程序能够正常地在目标计算机上运行，必须要给目标计算机安装 JDK。JDK (Java Development Kit) 也就是 Java 开发运行环境。读者可以从 Sun 公司的 JavaSoftWeb 站点 (<http://java.sun.com>) 下载。目前 JDK 最新版本是 JDK 1.5 版，不过本书使用的是最常用和稳定的 1.4 版。读者下载时可以参考图 1-1。



图 1-1 JDK 下载页面



请读者看清楚所需要下载的平台版本，如果读者在 Windows 平台的计算机上使用，请选择 Windows 平台的 JDK 版本。

安装 JDK 非常简单，读者甚至可以一直单击“下一步”按钮。JDK 会默认安装在读者计算机的 C:\JDK1.4 目录下。安装完 JDK，要在系统中配置 Java 的运行环境，必须保证正确设置两个环境变量 PATH 和 CLASSPATH。这两个变量的正确设置能使 JDK 安装到计算机的正确位置上。

当用户要运行一个 Java 类而又没有指定该类的完整路径，操作系统就到 PATH 变量指定的位置去查找该可执行类文件。为了能够在计算机的任何位置正常使用 JDK，必须改动现有的 PATH 环境变量或定义一个新的 PATH 参数值。在不同的操作系统中设置的方法有所不同，下面分别进行说明。

1. Windows 2000/XP 操作系统下的设置

Windows 2000 操作系统和 Windows XP 操作系统设置方法基本相同，此处以 Windows XP 为例介绍。

① 在桌面的“我的电脑”图标上单击鼠标右键，在弹出的菜单中选择“属性”，系统弹出“系统属性”对话框，再选择“高级”选项卡，如图 1-2 所示。

② 单击“环境变量”按钮，弹出如图 1-3 所示的对话框。

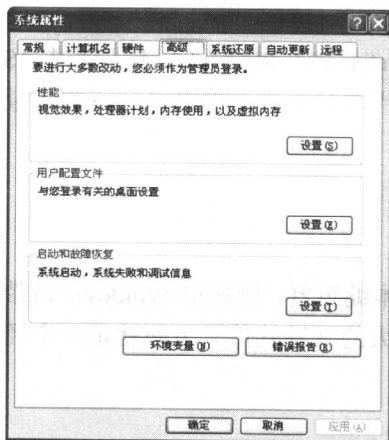


图 1-2 “系统属性”对话框

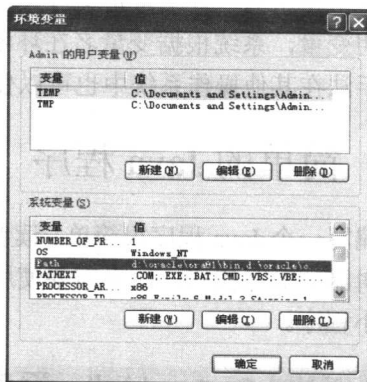


图 1-3 “环境变量”对话框

③ 单击“Admin 的用户变量”列表框下的“新建”按钮，在弹出的对话框中对应的“变量名”文本框输入“PATH”，对应的“变量值”输入“c:\jdk1.4\bin”，也就是 JDK 的安装路径。单击“确定”按钮。

④ 按照上面的步骤，再新建一个变量名为“CLASSPATH”的变量，变量值为“.;C:\jdk1.4\jre\rt.jar;C:\jdk1.4\lib\tool.jar”。其中的“.”代表用户所在的当前目录，在编译和运行 Java 程序时非常有用，请读者不要忽略。

2. Solaris 操作系统下的设置

Solaris 用户通常在系统外壳的开始文件中设置环境变量，在大多数情况下通过 sh 外壳或 csh 外壳的设置来完成。如果把 JDK 安装在安装程序所提供的默认位置/usr/local/jdk1.4/bin，那么每次运行 JDK 工具时，JDK 工具就会自动把 JDK 自带的类库附加在设置的 CLASSPATH 上。但是如果使用自己创建的类文件，那么就应该创建一个合适的 CLASSPATH 设置以让 JDK 工具查找这些程序。

以.cshrc 文件为例，若要修改 PATH 环境变量的设置和 CLASSPATH 环境变量的设置，需要在该文件中插入如下内容：

```
set path=(/usr/local/jdk1.4/bin $path);
set classpath=(/usr/local/jdk1.4/jre/lib; /usr/local/jdk1.4/lib);
```

然后让操作系统通过输入命令“source ~/.cshrc”重新读取改动后的.cshrc 文件。若在.profile 文件中修改这两个环境变量，方法类似。

3. Linux 操作系统下的设置

Linux 系统配置和 Solaris 配置差不多，不过编辑的文件是.bash_profile。在.bash_profile 文件后追加下面几行：

```
JAVA_HOME=/usr/java/j2sdk1.4;
PATH=$PATH:$JAVA_HOME/bin;
CLASSPATH=.;$JAVA_HOME/lib/tools.jar;$JAVA_HOME/jre/lib/rt.jar;
```

这里使用另一种变量定义方式，默认 JDK 安装在/usr/java/j2sdk1.4 目录下。通过“\$”符号声明变量，系统根据变量名在环境中找到变量值并替代现有变量，最终得到新的变量值。这种手法在其他操作系统中也可以使用。

1.5 简单的 Java 程序

编写一个 Java 程序很简单，读者可以使用任何的文本编辑器，例如用 Windows 自带的文本编辑器来书写代码，而不需要使用像 Visual C 那么庞大的开发软件。下面就是一个经典的 Java 小程序。

实例 1-1 Hello 程序

```
public class Hello{ //第 1 行
    public static void main(String[] arg){ //第 2 行
        System.out.println("Hello, Java World!"); //第 3 行
    } //第 4 行
} //第 5 行
```

这个例子之所以经典，是因为它包含了 Java 应用程序的大多数特性。