

清华 电脑学堂

Java

程序设计 实践教程

■ 刘万军 梁清华 王松波 宋 强 等编著

- 总结了作者长期教学培训成果，难易适中，实用性强
- 系统全面地介绍了Java的技术要点
- 围绕丰富的实例讲解编程实践知识
- 精心编写了大量“实验指导”，引导学生深入练习编程实践
- 网站提供代码下载和课件支持

清华大学出版社



清华 电脑学堂

Java

程序设计 实践教程

■ 刘万军 梁清华 王松波 宋 强 等编著

清华大学出版社
北 京

内 容 简 介

Java 是一种面向对象的新型编程语言, 应用非常广泛。本书详细介绍了 Java 语言的基础知识, 数据类型、运算符和表达式, 程序控制语句, 类、对象和方法, 继承和接口, 字符串, 包、类库和 Applet 等基础知识。在此基础上, 本书讲解了 Java 编程语言中 GUI 编程, 异常处理, 多线程, I/O 系统, 网络通信和多媒体, 数据库编程和广泛的 Java 数据结构等主题, 读者可以深入学习 Java 语言的应用知识。

本书可以作为读者学习 Java 语言和面向对象开发的教程, 适合于普通高校计算机专业和非计算机专业的程序设计教程, 也可供自学读者使用。

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

本书防伪标签采用特殊防伪技术, 用户可通过在图案表面涂抹清水, 图案消失, 水干后图案复现; 或将表面膜揭下, 放在白纸上用彩笔涂抹, 图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

Java 程序设计实践教程 / 刘万军等编著. —北京: 清华大学出版社, 2006.9

ISBN 7-302-13465-0

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

中国版本图书馆 CIP 数据核字 (2006) 第 099488 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社 总 机: 010-62770175 客户服务: 010-62776969

责任编辑: 夏兆彦

印 装 者: 北京鑫海金澳胶印有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185 × 260 印张: 26 字数: 645 千字

版 次: 2006 年 9 月第 1 版 2006 年 9 月第 1 次印刷

书 号: ISBN 7-302-13465-0/TP · 8450

印 数: 1 ~ 5000

定 价: 38.00 元

Java 是 Sun 公司开发的新一代编程语言。Java 是一种卓越的 Internet 语言，它的平台无关性使其非常适合当前的网络应用开发任务。使程序设计产生了革命，改变了我们考虑程序形式与功能的方式。

本书内容

本书主要介绍了 Java 语言的特点以及面向对象程序设计的基本概念等；使用 Java 语言进行程序的基础；Java 的程序控制语句；Java 数组知识，包括 Java 的一维数组、多维数组、不规则数组、搜索和排序数组等。Java 是一种完全面向对象的编程语言，本书重点介绍了 Java 中的类、对象和方法，内容包括在 Java 中构造类、创建类的实例化对象，控制类成员的访问以及 this 关键字等；Java 面向对象中的继承和接口，内容包括继承、重载、重写、构造抽象类以及 Java 语言的根类和接口等。

本书深入介绍了 Java 语言中字符串的实现；包、类库和 Applet；GUI 编程基础；Java 的异常处理；Java 的多线程程序设计；使用文件保存 Java 程序中的数据。这些知识完善了 Java 面向对象语言的功能。

本书最后介绍了 Java 的网络和多媒体编程；使用 JDBC 开发 Java 数据库程序。这些知识更进一步体现了 Java 语言的实用性。本书最后介绍 Java 程序开发中常用的类，内容包括 Java 的集合架构以及可枚举的类和接口等。这些内容体现了 J2SE 5 的新特性。

本书特色

本书通过实例介绍 Java 程序开发知识，具有实用性教程的特色。

- ❑ 本书汇总了多年的程序员职业教学培训经验，内容组织更合理，实例丰富全面。
- ❑ 本书使用 Java 语言开发了大量实例，读者可以通过这些丰富实例学习 Java 编程实践知识。
- ❑ 本章编写了大量“实验项目”，引导读者应用该章知识独立练习编程项目。
- ❑ 复习题可以帮助学生检查对 Java 开发理论知识的掌握程度。

读者对象

本书由多家院校的教师联合编写，在多家院校成熟教案及原有自编教材的基础上整合编写。作者均从事 Java 教学和开发工作，拥有丰富的 Java 开发案例和教学经验。本书共 15 章，需要 45 个课时。为了给教师授课提供方便，本书提供了教学课件，读者可以从 www.tup.tsinghua.edu.cn 下载使用。

本书可以作为普通高校计算机相关专业 Java 编程初级教程，也可以作为接触过 Java 基础知识，需要深入学习面向对象开发的中级教程。编写过程难免会有错误，欢迎读者与我们联系，帮助我们改正提高。

编 者

第 1 章 Java 概述 1	
1.1 Java 的发展..... 1	
1.1.1 Java 的发展史..... 1	
1.1.2 Java 与 C、C++ 和 C# 的关系..... 2	
1.2 Java 的特点..... 3	
1.3 面向对象程序设计..... 4	
1.3.1 封装..... 5	
1.3.2 继承..... 5	
1.3.3 多态性..... 6	
1.3.4 Java 的开发工具..... 6	
1.4 一个简单的 Java 程序..... 7	
1.4.1 编写 Java 程序..... 7	
1.4.2 编译和执行 Java 程序..... 8	
1.4.3 Java 程序的基本要素..... 9	
1.5 实验指导..... 12	
1.6 思考与练习..... 13	
第 2 章 数据类型与运算符 15	
2.1 常量、变量和表达式..... 15	
2.1.1 常量..... 15	
2.1.2 变量..... 16	
2.2 数据类型..... 18	
2.2.1 整数类型..... 19	
2.2.2 浮点类型..... 20	
2.2.3 字符类型..... 21	
2.2.4 布尔类型..... 22	
2.2.5 类型之间的转换..... 22	
2.2.6 转义序列..... 23	
2.3 运算符和表达式..... 24	
2.3.1 算术运算符..... 25	
2.3.2 关系运算符和逻辑运算符..... 28	
2.3.3 位运算符..... 29	
2.3.4 赋值运算符和条件运算符..... 30	
2.3.5 扩展运算符..... 31	
2.3.6 其他运算符..... 32	
2.3.7 运算符优先级..... 33	
2.4 实验指导..... 33	
2.5 思考与练习..... 34	
第 3 章 程序控制语句 36	
3.1 选择语句..... 36	
3.1.1 简单的 if 语句..... 36	
3.1.2 if-else 语句..... 37	
3.1.3 if 嵌套语句..... 38	
3.1.4 switch 语句..... 40	
3.2 循环语句..... 42	
3.2.1 while 循环语句..... 42	
3.2.2 do-while 循环语句..... 43	
3.2.3 for 循环语句..... 45	
3.3 与程序有关的其他语句..... 47	
3.3.1 break 语句..... 47	
3.3.2 continue 语句..... 51	
3.3.3 return 语句..... 52	
3.4 实验指导..... 53	
3.5 思考与练习..... 53	
第 4 章 数组 56	
4.1 一维数组..... 56	
4.1.1 声明数组和创建数组..... 56	
4.1.2 一维数组初始化及元素引用..... 57	
4.2 多维数组..... 60	
4.2.1 声明数组和创建数组..... 60	
4.2.2 初始化数组..... 61	
4.2.3 数组程序举例..... 62	

4.3 不规则数组	67	6.2.4 何时调用构造函数	112
4.4 搜索数组	70	6.3 方法重载和方法重写	113
4.5 排序数组	72	6.3.1 方法重载	113
4.5.1 选择排序	72	6.3.2 方法重写	114
4.5.2 冒泡排序	73	6.3.3 重写的方法支持多态性	115
4.5.3 快速排序	75	6.3.4 使用重写方法的原因	116
4.6 实验指导	76	6.4 抽象类	117
4.7 思考与练习	78	6.5 继承中的 final 修饰符	119
第 5 章 类、对象和方法	79	6.5.1 使用 final 防止重写	119
5.1 Java 类	79	6.5.2 使用 final 防止继承	120
5.1.1 类的基本形式	79	6.6 Object 类	120
5.1.2 类的定义	80	6.6.1 clone 方法	121
5.2 创建对象	82	6.6.2 equals 方法	121
5.3 对象引用变量	83	6.6.3 toString 方法	122
5.4 方法	84	6.7 接口	122
5.4.1 添加方法	85	6.7.1 接口定义	122
5.4.2 调用对象的方法	86	6.7.2 实现接口	123
5.4.3 将对象传递给方法	87	6.7.3 接口引用	126
5.5 构造函数和垃圾回收	88	6.7.4 接口中的变量	128
5.5.1 构造函数	88	6.7.5 接口的扩展	129
5.5.2 垃圾回收	89	6.8 实验指导	130
5.6 new 运算符	90	6.9 思考与练习	131
5.7 类成员的访问控制	90	第 7 章 字符串	133
5.8 共享变量、常量和静态方法	91	7.1 字符类	133
5.9 this 关键字	94	7.1.1 构造字符串	133
5.10 对象数组	95	7.1.2 匹配字符串	136
5.11 实验指导	97	7.1.3 求字符串长度和检索 单个字符	136
5.12 思考与练习	98	7.1.4 字符串比较	137
第 6 章 继承和接口	101	7.1.5 查找子字符串	138
6.1 继承	101	7.2 Character 类	141
6.1.1 超类和子类	101	7.2.1 Character 类的介绍	141
6.1.2 成员访问和继承	103	7.2.2 Character 类方法的应用	142
6.2 super 关键字	105	7.3 StringBuffer 类	144
6.2.1 使用 super 调用超类 构造函数	106	7.3.1 构造 StringBuffer 类	145
6.2.2 使用 super 访问超类成员	109	7.3.2 StringBuffer 类常用方法	146
6.2.3 创建多级层次结构	110	7.3.3 StringBuffer 类特有的 方法	148

7.3.4	StringBuffer 类特有方法 的应用示例	148	9.6.3	单行文本框和多行文 本域	196
7.4	实验指导	151	9.6.4	复选框和单选按钮	199
7.5	思考与练习	152	9.6.5	组合框	203
第 8 章	包和 Applet	155	9.6.6	滚动条	206
8.1	包	155	9.6.7	滚动面板	209
8.1.1	定义包	155	9.6.8	菜单	210
8.1.2	导入包	156	9.7	GUI 绘图	213
8.1.3	类路径 (CLASSPATH)	157	9.7.1	颜色	214
8.1.4	访问控制	157	9.7.2	绘制几何图形	214
8.2	applet 程序	160	9.7.3	显示字体	218
8.2.1	applet 类	160	9.8	实验指导	221
8.2.2	请求重绘	163	9.9	思考与练习	223
8.3	HTML APPLET 标记	164	第 10 章	异常处理	227
8.4	将参数传递到 applet	165	10.1	异常和异常类型	227
8.5	实验指导	166	10.1.1	异常的含义	227
8.6	思考与练习	167	10.1.2	异常类型	228
第 9 章	GUI 编程基础	169	10.2	理解异常处理	228
9.1	GUI 类	169	10.3	声明异常 (throws)	230
9.2	框架	171	10.4	抛出异常 (throw)	231
9.2.1	创建框架	171	10.5	捕获异常	233
9.2.2	在框架内添加组件	172	10.5.1	使用 try-catch	233
9.3	布局管理	173	10.5.2	使用多重 catch 语句	235
9.3.1	FlowLayout	173	10.5.3	嵌套 try 语句	236
9.3.2	GridLayout	174	10.6	finally 的用法	238
9.3.3	BorderLayout	175	10.7	异常的应用	240
9.3.4	CardLayout	177	10.7.1	Java 中的内置异常	241
9.3.5	GridBagLayout	179	10.7.2	创建自定义的异常类	242
9.3.6	null	182	10.7.3	输出警告信息	244
9.4	面板容器	183	10.7.4	应用异常的建议	245
9.5	事件	185	10.8	实验指导	246
9.5.1	事件和事件源	185	10.9	思考与练习	248
9.5.2	事件监听	186	第 11 章	线程与多线程	251
9.5.3	处理事件	188	11.1	线程与多线程	251
9.6	用户界面对象 (UI)	192	11.1.1	线程的性质	252
9.6.1	按钮	192	11.1.2	线程的状态	254
9.6.2	标签	195	11.1.3	守护线程	254

11.2 创建线程·····	256	12.3.3 LineNumberInputStream·····	311
11.2.1 扩展 Thread 类创建 线程·····	257	12.3.4 PushbackInputStream·····	312
11.2.2 实现 Runnable 接口 创建线程·····	259	12.4 字符流·····	313
11.2.3 创建多线程·····	260	12.4.1 基类 Reader 和 Writer·····	313
11.3 多线程程序设计·····	263	12.4.2 InputStreamReader 和 OutputStreamWriter·····	314
11.3.1 多线程同时运行·····	263	12.4.3 FileReader 和 FileWriter·····	315
11.3.2 多线程优先级调度·····	267	12.4.4 BufferedReader 和 BufferedWriter·····	316
11.3.3 多线程的互斥和同步·····	269	12.4.5 PrintWriter 类·····	317
11.3.4 死锁·····	276	12.5 File 类·····	318
11.4 多线程的控制·····	280	12.5.1 File 类的 3 种构造 函数·····	318
11.4.1 判断线程是否结束·····	281	12.5.2 File 类的常用方法·····	319
11.4.2 线程阻塞·····	283	12.6 对象的串行化·····	321
11.4.3 处理线程阻塞·····	284	12.7 实验指导·····	325
11.5 线程的应用·····	288	12.8 思考与练习·····	326
11.5.1 为 applet 创建线程·····	289		
11.5.2 使用 Timer 类来控制 动画·····	290		
11.6 实验指导·····	295		
11.7 思考与练习·····	295		
第 12 章 输入输出系统·····	297	第 13 章 网络通信和多媒体·····	328
12.1 Java 的 I/O 基于流·····	297	13.1 网络基础知识·····	328
12.1.1 字节流·····	298	13.1.1 域名和 IP 地址·····	328
12.1.2 字符流·····	299	13.1.2 端口号(port)·····	330
12.1.3 预定义流·····	300	13.2 Socket (套接字) 通信·····	330
12.2 字节流·····	302	13.3 UDP 和 FTP 类型通信·····	334
12.2.1 InputStream 类和 OutputStream 类·····	302	13.3.1 UDP 通信介绍·····	334
12.2.2 文件字节流的顺序 输入输出·····	303	13.3.2 FTP 通信介绍·····	335
12.2.3 文件字节流的随机 访问·····	306	13.4 applet 通信·····	336
12.3 过滤流·····	308	13.4.1 同页 applet 之间的 通信·····	336
12.3.1 DataInputStream 和 DataOutputStream·····	309	13.4.2 applet 和浏览器之间的 通信·····	340
12.3.2 BufferedInputStream 和 BufferedOutputStream·····	310	13.5 多媒体·····	342
		13.5.1 Applet 中播放音频·····	342
		13.5.2 Applet 中显示图像·····	345
		13.5.3 在应用程序中载入图像 和音频文件·····	347
		13.5.4 显示一系列图像·····	351
		13.5.5 MediaTracker·····	353

13.6 实验指导.....	355
13.7 思考与练习	356
第 14 章 数据库编程	358
14.1 JDBC 简介.....	358
14.2 JDBC 应用模型.....	359
14.3 java.sql 包	360
14.4 访问数据库	360
14.4.1 配置 ODBC 数据源	360
14.4.2 加载 JDBC 驱动程序.....	361
14.4.3 创建数据库连接.....	362
14.4.4 执行 SQL 语句.....	363
14.4.5 处理结果集.....	364
14.4.6 更新数据库.....	369
14.4.7 数据库连接.....	372
14.5 使用预编译 SQL 语句.....	373
14.6 获取数据库信息	375
14.7 实验指导.....	377
14.8 思考与练习	378
第 15 章 Java 的数据结构	381
15.1 了解 Java 集合架构	381
15.2 Collection 接口、Map 接口及其子接口	383
15.2.1 Collection 接口	383
15.2.2 Set 接口和 SortedSet 接口	384
15.2.3 List 接口.....	384
15.2.4 Map 接口和 SortedMap 接口.....	385
15.3 AbstractSet 类和 HashSet 类.....	386
15.4 TreeSet 类和 Comparator 接口	387
15.4.1 使用 TreeSet 类.....	387
15.4.2 使用 Comparator 接口.....	388
15.5 ArrayList 类和 LinkedList 类.....	389
15.6 Vector 类和 Stack 类	390
15.7 HashMap 类和 TreeMap 类.....	392
15.8 Collections 类和 Arrays 类.....	395
15.8.1 使用 Collections 类.....	395
15.8.2 使用 Arrays 类	398
15.9 实验指导.....	400
15.10 思考与练习	400
附录 思考与练习答案	402

第1章 Java 概述

Java 是一种跨平台、面向对象、基于网络及支持多媒体的编程语言。自 1995 年 5 月发布以来, Java 很快成为开发因特网应用程序的极具魅力的语言。Java 程序可以插入 HTML 页, 由 Web 浏览器下载, 为 Web 客户带来动画和交互式的享受。Java 的强大功能不仅仅限于 Web 应用程序。它还是一种面向对象的通用编程语言, 其编程功能完备, 可以开发单独的应用程序。

Java 体系非常庞大, 本章将介绍 Java 的基本知识和 Java 的基础特性。Java 虽然是一种面向对象的编程语言, 但是也兼容面向过程语言中的数据类型、控制语句等基础知识, 本章将会详细介绍。

本章学习目的是了解 Java, 包括它的历史、设计原理和一些最重要的特性。另外, 通过一个简单的 Java 程序, 我们还要学习 Java 程序的编写、编译和执行以及编码规范等基础知识。对于这些内容我们并不进行深入讨论, 只是关注 Java 程序共有的一些概念。

本章学习要点:

- 了解 Java 的发展
- 了解 Java 与 C、C++、C#之间的关系
- 了解 Java 的特点
- 理解 Java 是一种面向对象程序设计的方法及三种特性
- 掌握 Java 程序开发工具 JDK1.5 的配置及应用
- 掌握 Java 程序的基本要素及编码规范

1.1 Java 的发展

Java 语言是一门发展非常迅速的语言, 从诞生到广泛应用仅用了几年的时间。Java 诞生之初, 只有在美国硅谷的 Sun、IBM、BEA 等几家公司有人知道, 或听说过 Java 这个名词。现在无论在美国还是在中国, 或在世界上的其他地方, 你都能到图书馆、网站上找到大量的 Java 资料。Java 迅速从一种编程语言发展为一个平台、一个社群、一个产业。现在 Java 是最重要的编程技术之一, 全球超过 450 万名程序员使用 Java 开发软件, 14 亿部设备上运行着 Java 编写的程序。Java 已经拥有一个庞大的体系结构。

1.1.1 Java 的发展史

Java 是 Sun 公司于 1991 年开发的新一代编程语言, 最初目的是为家用消费类电子产品开发分布式代码系统。后来为了使整个系统与平台无关, 该项目小组的领导人 James Gosling 决定开发一种新语言, 称为 Oak, 这就是 Java 语言的前身, 后来改名为 Java。

随着 Internet 的迅速发展, Web 的应用日益广泛, Java 语言也得到了迅速发展。1994 年, Gosling 用 Java 开发了一个实时性较高、可靠安全、有交互功能的新型 Web 浏览器, 它不依赖于任何硬件平台和软件平台。这种新的浏览器称为 HotJava。1995 年在业界发表, 引起了巨大轰动, Java 的地位随之而得到肯定。

Java 语言发展非常迅猛, 1995 年 3 月 Sun 公司发布了 Java 语言的 Alpha1.0a2 版本, 1996 年 1 月发布了 Java 语言的第一个开发包 JDK v1.0, 1997 年 2 月发布了 Java 语言的开发包 JDK v1.1, 从而奠定了 Java 语言在计算机语言中的地位。1998 年 12 月 Sun 公司发布 Java 2 开发平台 JDK v1.2。Java 2 平台是 Java 发展史上的里程碑。1999 年 6 月, Sun 公司重新组织 Java 平台的集成方法, 并将企业级应用平台作为 Java 语言发展方向, 包含了 3 个成员:

- **J2ME (Java 2 Micro Edition)** 用于嵌入式应用的 Java 2 平台。
- **J2SE (Java 2 Standard Edition)** 用于工作站、PC 机的 Java 2 标准平台。
- **J2EE (Java 2 Enterprise Edition)** 可扩展的企业级应用的 Java 2 平台。

Java 是天生面向对象的计算机语言。虽然许多面向对象的语言一开始就是严格的过程化语言, Java 却从一开始就被设计为面向对象的语言。

1.1.2 Java 与 C、C++和 C#的关系

Java 又被称作 C++++, 意思是给 C++增加新功能, 去除不良功能。从这一点可以看出 Java 与 C 和 C++直接相关。Java 继承了 C 的大部分语法, Java 的对象模型则从 C++改编来的。

现代编程语言先驱是 C 语言。C++在 C 语言的基础上, 通过增加面向对象的特性扩充形成。因此它包括了 C 所有的特征、属性和优点。这也是 C++作为语言成功的一个关键原因。Java 是为了满足 Web 的需要而在 C 和 C++的基础上开发出来的, Java 大部分的特性是从 C 和 C++中继承的。

因为 Java 与 C++的相似性, 特别是它们对面向对象程序设计的支持, 使得 Java 被简单认为是“C++的 Internet 版”。这种观点是错误的。因为 Java 在实际应用以及基本原理上与 C++有显著的不同。尽管 Java 受到 C++的影响, 但是它绝不是 C++的增强版。例如, Java 不提供对 C++的向上或向下兼容功能。另外, Java 不是为替代 C++而设计的, 两者将在很长的时期内共存。

近来, 一种名为 C#的新语言浮出了水面。由 Microsoft 创建, 用以支持其 .NET 架构的 C#与 Java 也密切相关。事实上, C#的许多功能都是直接从 Java 改编来的。Java 和 C#共享相同的 C++语法风格, 都支持分布式程序设计, 使用相同的对象模型。它们之间存在不同, 但就整体外观而言, 两者极为相似。如果已经掌握了 C#, 那么学习 Java 就很简单, 反之, 如果将来要学习 C#, 那么现在学到的有关 Java 的知识也会使你将来受益匪浅。

鉴于 Java 与 C#两者的相似性, 自然有人要问, “C#会替代 Java 吗”? Java 和 C#是对两种不同类型计算环境的优化, 正如 C++会与 Java 长期共存一样, C#与 Java 也会长期共存。

1.2 Java 的特点

Java 语言由于适用于 Internet 环境, 以及它具有小型化、平台无关性等特点, 引起了计算机界的广泛关注, 从而迅速发展成为计算机网络编程语言。它简单、面向对象, 不依赖于机器结构, 不受环境的限制, 具有可移植性、可靠性、安全性等特点, 并且提供了多线程并发机制, 因此具有很好的性能。另外, Java 语言还提供了丰富的类库, 方便用户进行自定义操作。下面我们就来论述 Java 的具体特点。

1. 简单性

Java 语言是一种面向对象语言, 语法规则和 C++ 类似。它通过提供最基本的方法来完成指定的任务。但 Java 语言对 C++ 进行了简化和提高。例如, 指针和多重继承通常使编程变得复杂, Java 用称为接口的简单语言构造取代了 C++ 中的多重继承, 并取消了指针。Java 语言还通过实现自动垃圾收集大大简化了程序设计人员的内存管理工作。另外, 它对环境要求很低, 包括基本解释器、类、标准类库和线程的支持才仅仅有两百多千字节。

2. 可靠性

Java 语言具有很强的可靠性。它在程序编译和运行时, 对所有可能出现的问题都要进行检查, 可以消除错误的产生。通过在运行时实现面向对象的异常处理机制, Java 可以捕获异常情况并对它做出响应, 程序在发生运行期错误时能继续正常执行并完美终止。另外, Java 取消了在其他语言中发现的易于产生错误的编程构造类型。例如, 它不支持指针, 因此消除了重写内存及破坏数据的可能性。

3. 安全性

Java 语言作为一种因特网编程语言, 当用于网络和分布式环境时, 必须具有防止病毒入侵的能力。Java 语言不支持指针, 一切对内存的访问都必须通过对象的实例变量来实现, 这样就防止用户使用“特洛伊”木马等欺骗手段访问对象的私有成员, 同时也避免了指针操作中容易产生的错误。另外, 如果下载一个 Java applet (特殊种类的程序) 并在计算机上运行, 它不会破坏你的系统, 因为 Java 执行了几种保护系统不受外部程序破坏的安全机制。但是该安全机制是基于“不要信任任何东西”这个前提, 存在着自己的弊端。

4. 解释器通用性

运行 Java 程序需要解释器。Java 解释器直接对 Java 字节码进行解释执行。字节码独立于机器, 它本身携带了许多编译时信息, 使得连接过程更加简单, 因此可以在任何有 Java 解释器的机器上运行。

5. 可移植性

Java 与体系结构无关。这一特征使 Java 程序可以方便地移植到网络的不同机器。同时, Java 的类库中也实现了针对不同平台的接口, 使这些类库可以移植。

6. 分布性

分布式计算涉及到在一个网络上协同工作的多台计算机。因为 Java 语言是面向网络的语言，所以通过它提供的类库可以处理 TCP/IP 协议，用户可以通过 URL 地址在网络上很方便地访问其他对象。

7. 独立性

Java 是可解释的。这一特征使 Java 独立于平台存在。使用 Java 虚拟机 (JVM)，可以编写运行在任何平台上的程序。原因是 Java 解释器生成与体系结构无关的字节码指令，这些字节码指令仅用于 Java 虚拟机，Java 解释器得到字节码后，对它进行转换，使之能够在不同的平台运行。

8. 面向对象性

Java 程序是面向对象的。面向对象编程提供了很大的灵活性、模块性和可重用性。Java 语言的设计集中于对象及其接口，它提供了简单的类机制以及动态的接口模型。对象中封装了它的状态变量以及相应的方法，实现了模块化和信息隐藏；而类则提供了一类对象的原型，并且通过继承机制，子类可以使用父类所提供的方法，实现代码的复用。

9. 高性能性

因为 Java 是可解释的，字节码不是直接由系统执行，而是在解释器中运行。因此，它的速度对多数交互式应用程序来说得到了很大的提高。另外，Java 字节码的设计使之能很容易地直接转换成对应于特定 CPU 的机器码，从而得到较高的性能。

10. 多线程性

多线程是程序同时执行多个任务的一种功能。多线程多用于图形用户界面 (GUI)、多媒体和网络编程中。多线程机制能够使用应用程序并行执行多项任务，而且同步机制保证合并对共享数据的正确操作。通过使用多线程，程序设计人员可以分别用不同的线程完成特定的行为，而不需要采用全局的事件循环机制，这样就很容易实现网络的实时交互行为。

11. 动态性

Java 语言的设计使它适合应用于不断变化的动态环境。Applet 作为一种特殊的 Java 程序，能嵌入到 html 文件中，实现用户的交互操作和网页的动态变化，而不是像动画或者声音文件那样反复地播放。另外，在类库中可以自由加入新的方法和实例变量，而不会影响用户程序的执行。并且 Java 通过接口来支持多重继承，使之比严格的类继承具有更灵活的方式和扩展性。

1.3 面向对象程序设计

Java 的核心是面向对象程序设计 (OOP)。OOP 是一种功能强大的程序设计方法。

从计算机诞生以来,为适应程序不断增加的复杂程度,程序设计方法也有了巨大的变化。计算机发展初期,程序设计是通过使用计算机面板输入二进制机器指令来完成的。20世纪70年代,诞生了结构化程序设计方法(如C和Pascal)。结构化语言可以编写中等复杂程度的程序,但不适合于开发大型工程。为了冲破这一束缚,创建了面向对象方法。面向对象程序设计方法采纳了结构化程序设计的思想精华,并且新增了一些概念和特性,使其性能大大提高。所有面向对象程序设计语言,包括Java在内,都有3种特性:封装(encapsulation)、多态性(polymorphism)和继承(inheritance)。下面将分别简单介绍它们的概念及用法。

1.3.1 封装

封装是一种将代码与其处理的数据绑定在一起的编程机制。该机制保证了程序的数据不受外部干扰且不被误用。理解封装性的方法就是把它想成一个黑匣子,它可以阻止在外部定义的代码随意访问内部代码和数据。对黑匣子内的代码和数据的访问通过适当定义的接口严格控制。

在面向对象语言中,代码和数据可以创建自包含的黑匣子。该黑匣中包含了所有必需的数据和代码。当代码和数据以这种方式绑定在一起时,就创建了对象。在对象中的代码或数据,仅可以被对象的其他部分访问,不允许被对象以外的任何程序访问时,我们就称该代码和数据为私有的(private)。对象中的代码或数据,尽管是定义在对象中,却允许对象所在程序的其他部分进行访问,那么我们称该代码和数据为公有的(public)。

Java的基本封装单元是类(class)。本书后面将会详尽地介绍类的知识。类定义了对象的形式,指定了数据和操作数据的代码。Java使用类规范来构造对象。对象是类的实例。因此,类在本质上是指定如何构建对象的一系列计划。组成类的代码或数据被称为类的成员(member)。类定义的数据被称为成员变量(member variable)或实例变量(instance variable)。处理这些数据的代码则被称为成员方法(member method)或简称为方法(method)。方法是子例程在Java中的术语。如果熟悉C/C++,那么这将有助于了解Java程序中的方法和C/C++程序中的函数。

1.3.2 继承

继承是一个对象获得另一个对象属性的过程。继承之所以重要是因为它支持了按层分类的概念,即类层次结构。例如,猎犬是狗类的一种,狗又是哺乳动物类的一种,哺乳动物类又是动物类的一部分。如果不使用层次结构,我们就不得不分别定义每个动物的所有属性。使用了继承,一个对象就只需定义使它在所属类中独一无二的属性即可,因为它可以从它的父类继承所有通用属性。因此,继承机制使对象成为更一般的类的特定实例。

继承性与封装性相互作用。如果一个给定的类封装了一种属性,那么它的任何子类将具有同样的属性,而且还添加了子类自己特有的属性。这是面向对象的程序在复杂性

上呈线性而非几何性增长的一个关键特性。

1.3.3 多态性

多态性是允许一个接口被多个同类动作使用的特性，具体使用哪个动作与应用场合有关。下面我们用一个先进先出型堆栈为例说明。假设你的程序需要 3 个不同类型的堆栈。一个用于处理整型值、一个用于处理浮点型值，一个用于处理字符。在这个示例中，尽管堆栈存储的数据类型不同，但是实现各个堆栈的算法都是一致的。在非面向对象的语言中，需要创建 3 个不同的堆栈例程，每个例程使用不同的名称。但是，如果使用 Java，由于它具有多态性的特点，你只需创建一个基本的堆栈例程就可以实现对这三种特定情况的处理。

多态性的概念经常被说成“一个接口，多种方法”。这就意味着可以为一组相关的动作，设计一个通用的接口。多态性通过允许同一接口被同一类的多个动作使用，从而减少了程序的复杂度。编译器的工作就是选择适用于各个情况的特定动作（即方法）。而你只需记住以及利用这个统一的接口就可以了。

多态性、封装性与继承性相互作用。精心设计的类层次结构是程序的基础，封装可以使在不破坏依赖于类公共接口的代码基础上对程序进行升级迁移，多态性则有助于编写清楚、易懂、易读、易修改的程序。在由多态性、封装性与继承性共同组成的编程环境中可以写出比面向过程模型环境更可靠、扩展性更好的程序。

1.3.4 Java 的开发工具

Java 语言提供了一个免费的开发工具集 Java Developer's Kit，简称 JDK，又称为 J2SDK。可以到 <http://www.javasoft.com> 上下载，其安装过程十分简单，是 Java 最基本的开发工具。目前最新版本是 JDK1.5，又称为 JDK5.0。JDK 由用于开发和测试 Java 程序的许多单独程序组成的，每个程序由命令行激活。在 JDK 中，有两个重要程序 `javac.exe` 和 `java.exe`。`javac.exe` 是 Java 语言编译器，用来将 java 源程序编译成 Java 字节码。`java.exe` 是 Java 语言解释器，用来执行 Java 字节码得到结果。

当安装了 JDK 以后，在编译和运行 Java 程序时，有两个环境变量需要设置：`path` 和 `classpath`，`path` 用于设置命令文件搜索的路径，如 `javac.exe`、`java.exe` 等，`classpath` 用于设置用户类文件搜索的路径。例如，JDK 安装在 `C:\Program Files\Java\jdk1.5.0_07` 中，而源文件及编译生成的目标文件在 `D:\mycode` 中。要能正确编译与运行。在 MS-DOS 方式下应该设置如下：

```
D:\>set path=.; C:\Program Files\Java\jdk1.5.0_07\bin;
D:\>set classpath=.;D:\mycode;
```

在 Windows 2000/NT/XP 环境中，右击桌面上“我的电脑”，选择“属性”，然后选择【高级】选项卡，单击【环境变量】按钮，出现如图 1-1 所示的【环境变量】对话框。选中系统变量下的 `path` 选项，单击【编辑】按钮，打开如图 1-2 所示的【编辑系统变量】

对话框,在变量值里添加或输入 JDK 开发工具包的路径。若在系统变量下没有 path 选项,则单击【新建】按钮,出现如图 1-3 所示的【新建系统变量】对话框,在变量名中输入 path,再在变量值中添加或输入 JDK 开发工具包的路径。classpath 的设置步骤与 path 设置完全类似。

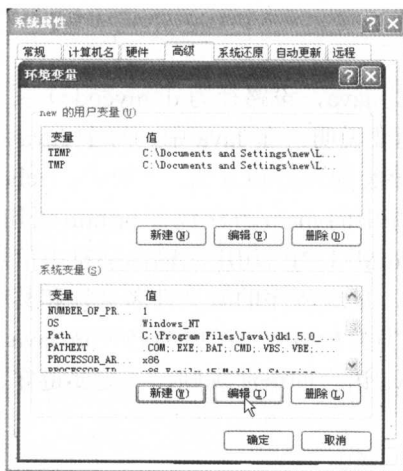


图 1-1 【环境变量】对话框

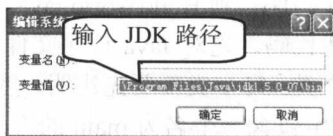


图 1-2 【编辑系统变量】对话框

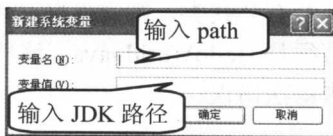


图 1-3 【新建系统变量】对话框

除传统的 JDK 之外,许多公司还推出了可视化的基于 Java 语言的集成开发工具,如 Visual Age for Java (即 WebSphere Studio Application Developer), JCreator, Workshop, JBuilder 等。其中 Borland 公司推出的 JBuilder 是目前较流行的 Java 可视化集成开发工具。在此我们不做过多介绍,有兴趣的读者可以查看其他相关资料。

1.4 一个简单的 Java 程序

Java 程序分为两种: Java Application 和 Java Applet。Java Application 可以独立运行。Java Applet 不能独立运行,它可以使用 Applet Viewer 或其他支持 Java 的浏览器运行。不论是哪种 Java 程序,都保存成后缀为 java 的文件。

本节将通过一个简单的 Java Application 程序,初步了解 Java 程序编写、编译、运行的基本过程;同时说明 Java 程序的基本要素。

1.4.1 编写 Java 程序

Java 程序的代码可以用多种工具编写,如记事本、Word、EditPlus、UltraEdit 等文本编辑工具。现在我们用记事本输入一段代码,实现在屏幕上输出“Hello World!”字符串的操作。代码如下:

```
/*
This is a simple Java program.
Call this file HelloWorld.java.
```