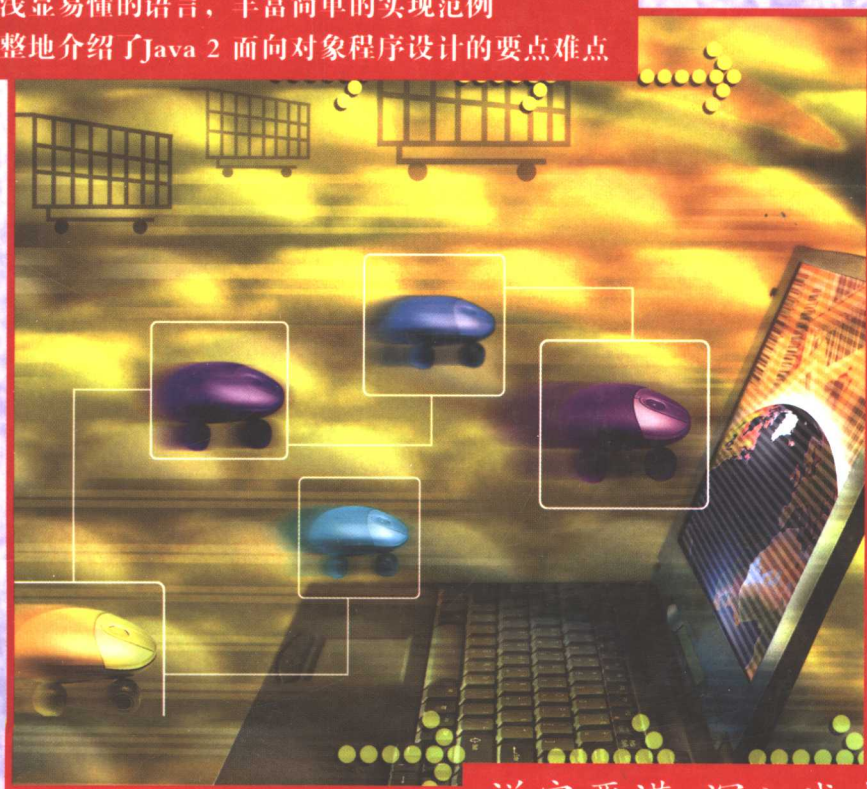


21世纪高等院校计算机教材

Java 2面向对象 程序设计

洪维恩 何嘉 编著

以浅显易懂的语言，丰富简单的实现范例
完整地介绍了Java 2 面向对象程序设计的要点难点



详实严谨 深入浅出

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

21 世纪高等院校计算机教材

Java 2 面向对象程序设计

洪维恩 何 嘉 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是一本全面介绍 Java 语言特点及应用技术的实用书籍。书中内容详实严谨、深入浅出、概念清楚。以 Java 2 基础程序设计、面向对象程序设计与窗口程序设计为三大主轴,利用浅显易懂的语言、丰富简单的实现范例,完整地介绍了 Java 2 面向对象程序设计的要点与难点。Java 的运算符、表达式与语句、循环与选择性程序结构、数组与函数、类与类库、继承、异常处理、多线程、文件处理与事件处理等在本书中都有详尽的介绍说明。

图书在版编目(CIP)数据

Java 2 面向对象程序设计/洪维恩,何嘉编著. —北京:中国铁道出版社,2004.12

(21 世纪高等院校计算机教材)

ISBN 7-113-05867-1

I. J… II. ①洪… ②何… 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 127229 号

书 名: Java 2 面向对象程序设计

作 者: 洪维恩 何 嘉

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 魏 春

责任编辑: 苏 茜 林菁菁 张雅静

封面制作: 白 雪

印 刷: 北京市彩桥印刷厂

开 本: 787×1092 1/16 印张: 23.75 字数: 570 千

版 本: 2005 年 1 月第 1 版 2005 年 1 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-05867-1/TP·1383

定 价: 30.00 元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

前 言

本书是一本全面介绍 Java 语言特点及应用技术的实用书籍。书中内容详实严谨、深入浅出、概念清楚。以 Java 2 基础程序设计、面向对象程序设计与窗口程序设计为三大主轴，采用浅显易懂的语言，丰富简单的实现范例，完整地介绍了 Java 2 面向对象程序设计的要点难点！Java 的运算符、表达式与语句、循环与选择性程序结构、数组与函数、类与类库、继承、异常处理、多线程、文件处理与事件处理等在本书中都有详尽的介绍说明。

全书共分 19 章。第 1 章到第 6 章介绍了 Java 语言规范、基本功能及特点。第 7 章到第 18 章介绍了 Java 语言面向对象的概念和方法，Java 语言面向对象的特性。第 19 章简要地介绍了 Applet 程序的概念及简单应用。该书既适合于初学者又可作为广大计算机技术人员的自学参考用书。

本书在编写过程中注重了简明、易学、实用的原则，语言流畅、通俗易懂。全书强调基础知识与大量的实例相结合。在每一章节后面都有本章摘要和自我考评，有益于读者在学习完每章内容后，能及时的复习和总结，帮助读者将所学到的内容融会贯通。

本书由何嘉老师制作完成的精美电子教案，出版社将作为教学服务内容免费提供，欢迎各位老师向出版社索要。同时，感谢赵玻、徐虹、曾文革对本书编写工作的大力支持。

由于编者水平有限，加上编写时间仓促，错误和不足之处在所难免，敬请广大读者朋友批评指正。

编 者
2004 年 12 月

目 录

第 1 章 认识 Java	1
1-1 历史的回顾	1
1-1-1 Java 的发展过程	1
1-1-2 Java 的发展史	1
1-2 Java 的虚拟机	2
1-3 Java 的未来发展	3
1-4 编写第一个 Java 程序	4
1-4-1 编译与运行 Java application	4
1-4-2 编译与运行 Java applet	6
本章摘要	9
自我考评	9
第 2 章 简单的 Java 程序	11
2-1 一个简单的例子	11
2-2 简单的 Java 程序解析	12
2-2-1 类 (class)	13
2-2-2 大括号、段及主体	13
2-2-3 程序运行的起始点——main () 方法	14
2-2-4 变量	14
2-2-5 变量的设置	15
2-2-6 为什么要声明变量	15
2-2-7 println()	15
2-3 再看一个例子：有参数的情况	16
2-4 标识符及关键字	17
2-4-1 标识符	17
2-4-2 关键字	17
2-5 检测	17
2-5-1 语法错误	18
2-5-2 语义错误	18
2-6 提高程序的可读性	19
本章摘要	20
自我考评	21
第 3 章 变量与数据类型	23
3-1 变量与常量	23

3-2 基本数据类型	24
3-2-1 整数类型	24
3-2-2 字符类型	28
3-2-3 单精度浮点数与双精度浮点数	30
3-2-4 布尔类型	31
3-2-5 基本数据类型的默认值	32
3-3 数据类型的转换	32
3-3-1 自动类型转换	32
3-3-2 强制类型转换	33
3-4 由键盘输入数据	34
3-4-1 输入字符串	35
3-4-2 输入数值	35
3-4-3 输入多个数据	36
本章摘要	37
自我考评	37
第 4 章 运算符、表达式与语句	40
4-1 表达式与运算符	40
4-1-1 赋值运算符	40
4-1-2 一元运算符	41
4-1-3 算术运算符	42
4-1-4 关系运算符与 if 语句	44
4-1-5 自增与自减运算符	45
4-1-6 逻辑运算符	46
4-1-7 括号运算符	47
4-2 运算符的优先级	48
4-3 表达式	49
4-4 表达式的类型转换	51
本章摘要	52
自我考评	52
第 5 章 循环与选择性语句	55
5-1 程序的结构设计	55
5-1-1 循环结构	55
5-1-2 选择结构	55
5-1-3 重复结构	56
5-2 循环	56
5-2-1 for 循环	56
5-2-2 while 循环	58

5-2-3 do while 循环	60
5-2-4 嵌套循环	61
5-3 循环的跳离	62
5-3-1 break 语句	62
5-3-2 continue 语句	63
5-4 我的程序会转弯——选择语句	65
5-4-1 if 语句	65
5-4-2 if-else 语句	65
5-4-3 if 的简洁版——条件运算符	66
5-5 更好用的多重选择——switch 语句	67
5-6 局部变量	69
本章摘要	70
自我考评	71
第 6 章 数组与函数	74
6-1 一维数组	74
6-1-1 一维数组的声明与内存的分配	74
6-1-2 数组中元素的表示方法	75
6-1-3 数组初值的赋值	76
6-1-4 简单的范例：找出数组元素的最大值与最小值	77
6-2 二维数组	78
6-2-1 二维数组的声明与分配内存	78
6-2-2 二维数组元素的引用及访问	80
6-3 三维以上的多维数组	81
6-4 函数	82
6-4-1 简单的范例	82
6-4-2 方法的参数与返回值	83
6-4-3 将数组传递到方法里	85
6-4-4 递归方法	87
本章摘要	88
自我考评	89
第 7 章 类的基本架构	92
7-1 认识类	92
7-1-1 类的基本概念	92
7-1-2 类的声明	93
7-1-3 创建新的对象	94
7-1-4 使用类来设计完整的程序	96
7-1-5 同时创建多个对象	97

7-2 有关方法的使用	97
7-2-1 定义与使用方法	98
7-2-2 再看一个简单的范例	99
7-2-3 数据成员的访问方式	101
7-2-4 在类定义的内部调用方法	102
7-3 参数的传递	103
7-3-1 调用方法并传递参数	103
7-3-2 传递多个参数	104
7-4 方法的返回值	105
7-4-1 没有返回值的方法	106
7-4-2 有返回值的方法	106
7-5 公有成员与私有成员	107
7-5-1 创建私有成员	108
7-5-2 创建公有成员	109
7-5-3 数据的封装	110
7-5-4 省略 public 与 private	111
本章摘要	111
自我考评	111
第 8 章 类的其他功能	114
8-1 函数成员（方法）的重载	114
8-1-1 重载	114
8-1-2 使用重载常犯的错误	117
8-2 构造函数	117
8-2-1 构造函数的基本认识	117
8-2-2 构造函数的调用时机	118
8-2-3 构造函数的重载	119
8-2-4 从某一构造函数调用另一构造函数	120
8-2-5 构造函数的公有与私有	121
8-2-6 构造函数的省略	122
8-3 类变量与类方法	123
8-3-1 实例变量与实例方法	123
8-3-2 类变量（class variable）	124
8-3-3 类方法	126
8-3-4 “类方法”使用的限制	128
8-4 类类型的变量	129
8-4-1 赋值给类类型的变量	129
8-4-2 以类类型的变量传递参数	131

8-4-3 由方法返回类类型的变量	132
8-4-4 释放内存	133
8-5 利用数组来保存对象	134
8-5-1 创建对象数组的范例	135
8-5-2 传递对象数组到方法里	136
8-6 内部类	137
8-6-1 内部类的编写	137
8-6-2 匿名内部类	139
本章摘要	141
自我考评	142
第 9 章 继承	144
9-1 继承的基本概念	144
9-1-1 简单的继承范例	144
9-1-2 调用父类中特定的构造函数	146
9-1-3 使用构造函数常见的错误	148
9-2 由子类访问父类的成员	150
9-3 覆盖	152
9-3-1 覆盖父类的方法	152
9-3-2 以父类的变量访问子类的成员	153
9-4 再谈 super() 与 this()	155
9-5 终止继承	156
9-6 类之源——Object 类	157
本章摘要	161
自我考评	161
第 10 章 抽象类与接口	164
10-1 抽象类	164
10-1-1 定义抽象类	164
10-1-2 抽象类的实现	164
10-1-3 用抽象类类型的变量来创建对象	167
10-1-4 使用抽象类的注意事项	168
10-2 接口的使用	169
10-3 多重继承	172
10-4 接口的扩展	174
本章摘要	176
自我考评	176

第 11 章 大型程序的发展	180
11-1 文件的分割	180
11-2 使用 package	181
11-2-1 package 的基本概念	181
11-2-2 将独立分开的类纳入同一个 package 中	183
11-3 访问在不同 package 里的类	184
11-3-1 简单的范例	184
11-3-2 public、private 与 protected 修饰符的角色	185
11-3-3 导入 packages	186
11-4 构造 package 的层次关系	187
11-5 Java 常用的类库	189
11-5-1 有关字符串的类库	191
11-5-2 StringBuffer 类库	192
11-5-3 wrapper class	193
11-5-4 使用 math 类	195
本章摘要	196
自我考评	197
第 12 章 异常处理	199
12-1 异常的基本概念	199
12-1-1 为何需要异常处理	199
12-1-2 简单的异常范例	199
12-1-3 异常的处理	200
12-1-4 异常处理机制的回顾	202
12-2 异常类的继承架构	203
12-3 抛出异常	204
12-3-1 在程序中抛出异常	204
12-3-2 指定方法抛出异常	206
12-4 自己编写异常类	207
12-5 回顾 IOException 异常类	208
本章摘要	209
自我考评	210
第 13 章 文件处理	212
13-1 关于流	212
13-2 使用 Reader 与 Writer 类	213
13-2-1 文件的读取——使用 FileReader 类	214
13-2-2 文件的写入——使用 FileWriter 类	215
13-2-3 文件的读取——使用 BufferedReader 类	216

13-2-4 文件的读取——使用 <code>BufferedWriter</code> 类.....	218
13-3 使用 <code>InputStream</code> 与 <code>OutputStream</code> 类.....	219
13-3-1 文件的写入——使用 <code>FileInputStream</code> 类.....	219
13-3-2 文件的写入——使用 <code>FileOutputStraem</code> 类.....	220
本章摘要.....	222
自我考评.....	222
第 14 章 多线程	224
14-1 认识线程.....	224
14-2 用 <code>Runnable</code> 接口来创建线程.....	227
14-3 线程的管理.....	228
14-3-1 线程的生命周期.....	228
14-3-2 让线程小睡片刻.....	229
14-3-3 等待线程.....	230
14-4 同步处理.....	232
本章摘要.....	234
自我考评.....	235
第 15 章 AWT 窗口对象	237
15-1 认识 AWT 类.....	237
15-1-1 简单的范例.....	237
15-1-2 窗口对象的类简介.....	240
15-2 创建窗口.....	243
15-3 创建标签.....	245
15-4 创建按钮.....	248
15-5 创建复选框.....	250
15-6 创建文字输入组件.....	252
15-6-1 用 <code>TextField</code> 创建文本框.....	252
15-6-2 用 <code>TextArea</code> 创建文本区.....	254
15-7 页面设置与管理.....	256
15-7-1 使用 <code>BorderLayout</code> 类.....	256
15-7-2 使用 <code>CardLayout</code> 类.....	258
15-7-3 使用 <code>FlowLayout</code> 类.....	259
15-7-4 使用 <code>GridLayout</code> 类.....	261
15-8 使用 <code>Panel</code> 面板.....	262
本章摘要.....	264
自我考评.....	265

第 16 章 事件处理	267
16-1 委派事件模型	267
16-1-1 简单的范例	267
16-1-2 定义内部类为聆听者	270
16-2 认识事件处理类	271
16-3 使用 ActionEvent 类	273
16-4 使用 ItemEvent 类	275
16-5 文本事件的处理——TextEvent 类	277
16-6 按键事件的处理——KeyEvent 类	278
16-6-1 以 KeyListener 接口处理 KeyEvent 事件	279
16-6-2 以 KeyAdapter 类事件处理 KeyEvent 事件	281
16-7 鼠标事件的处理——MouseEvent 类	282
16-7-1 以 MouseListener 接口处理 MouseEvent 事件	283
16-7-2 以 MouseMotionListener 接口处理 MouseEvent 事件	285
16-7-3 以 MouseAdapter 类事件处理 MouseEvent 事件	286
16-8 窗口事件的处理——WindowEvent 类	287
本章摘要	289
自我考评	290
第 17 章 菜单、菜单栏与滚动条	292
17-1 选择窗体	292
17-1-1 用 List 类创建选择窗体	292
17-1-2 选择窗体的事件处理	294
17-2 下拉菜单	295
17-2-1 用 Choice 类创建下拉菜单	295
17-2-2 下拉菜单的事件处理	296
17-3 菜单栏	298
17-3-1 创建菜单栏	299
17-3-2 菜单栏的事件处理	300
17-4 滚动条	301
17-5 对话框	304
17-6 文件对话框	307
本章摘要	309
自我考评	310
第 18 章 AWT 的绘图	312
18-1 绘图概述	312
18-1-1 坐标系统	312
18-1-2 Graphics 类	312

18-2 设置颜色与字体	317
18-2-1 设置颜色	317
18-2-2 设置字体	320
18-3 Graphics 类	322
本章摘要	332
自我考评	332
第 19 章 网页的向导——Applet	334
19-1 applet 概述	334
19-1-1 简单的 Java applet	334
19-1-2 Applet 类	336
19-1-3 使用 HTML 的<APPLET>标记	337
19-2 applet 的运行程序	342
19-3 装扮 applet 窗口	344
19-3-1 加入 AWT 组件到 applet 里	344
19-3-2 在 applet 窗口内用鼠标绘出圆形	345
19-4 加载影像与简单的动画制作	347
19-4-1 加载与显示影像	347
19-4-2 移动影像	349
19-5 播放音乐文件	350
本章摘要	352
自我考评	353
附录 下载与安装 JDK 1.4	355
A-1 JDK 的下载与安装	355
A-1-1 下载 JDK	355
A-1-2 安装 JDK	357
A-1-3 设置 JDK 的操作环境	359
A-2 JDK 参考文件下载与安装	362
A-2-1 下载 JDK 参考文件	362
A-2-2 安装 JDK 参考文件	364

第 1 章 认识 Java

本章将对 Java 做一个概念性的解说，从历史的回顾、Java 的发展与扩展，一直到 Java 的未来，本章均从头细说。同时将指导读者用最简单的语法编写第一个 Java 程序。读完本章后，读者一定会对 Java 有更进一步的了解。

1-1 历史的回顾

Java 于 1995 年诞生，它是由美国加州的 Sun 计算机公司（Sun Microsystems, Inc.）所推出，是一种能够跨平台（cross platform）使用的程序语言。据说当初在设计 Java 时，程序设计师们在印尼爪哇岛度假，喝 Java 咖啡时得到灵感，因此得名。

1-1-1 Java 的发展过程

在 1991 年，当 World Wide Web 开始起步，在 Sun 计算机公司中，James Gosling 领导的小组开始研发一种可控制家用电子产品的新型计算机软件技术，并希望能够研究出一种可以跨平台的系统。起先他们试着从 C++ 的功能做修改，但一直无法克服编译器上的问题，所以决定自行开发新的程序语言——Oak。

Oak 程序语言的由来，是当 James Gosling 在制订 Oak 的树状结构时，抬头看到窗外一棵枝叶茂盛的橡树，即命名之，它也是 Java 语言的前身。Oak 早期的发展并不顺利，因在一次争取电视机上的计算机控制器项目失败后，整个小组被解散，只剩下几个人在研究如何将这种计算机技术应用到多媒体。后来 Sun 的人员发现 Oak 是已经存在的一门计算机语言的名字，因此他们把名字改成了 Java。

Web 于 1994 年在 Internet 上开始盛行，这使得 Java 工作小组灵机一动，将 Java 应用到 Web 上，没想到这个举动竟使 Java 一炮而红，变成 Internet 上最闪亮的巨星。

1-1-2 Java 的发展史

Java 的发展过程很特殊，没有人会想到它是通过家电用品的发展演变而来的吧！一起来了解一下它的成长过程：

- | | |
|--------|---|
| 1991 年 | Sun 公司进军消费电子产品（IA）市场。Sun 成立“Green”小组，以 C++ 为基础开发新的程序语言 Oak。 |
| 1992 年 | Green 小组升格为 First Person 公司，与 Time-Warner 有线电视公司合作失败后，他们的计划几乎被迫取消，随后将这种技术转移到 web 上，并把 Oak 改名为 Java。 |
| 1994 年 | Web 在 Internet 上开始流行，Java 语言的开发者意识到 Java 语言能够设计一个真正酷的浏览器。开始开发 HotJava 浏览器。 |
| 1995 年 | 在 1995 年 5 月 23 日的 Sun World'95 上展出具有了执行网页内嵌代码能力的 HotJava 浏览器。网景（Netscape）与 Sun 公司合作，在 Netscape Navigator 中支持 Java。微软 |

(MicroSoft) IE 加入支持 Java 的行列。它激发了一直持续到今天的 Java 狂热。

1996 年	Java 1.0 版正式诞生。
1997 年	Java 急速发展至 1.1 版, 届时分为 Personal Java 和 Embedded Java 两大派。
1999 年	升级成 1.2 版。
2000 年	升级至 1.3 版。
2001 年	升级到 1.4 版, 并形成 J2EE、J2SE 与 J2ME 体系。

现在, Sun 已经推出 Java 的 1.5beta 版。由于是测试版, 所以本书仍然是针对 J2SE 1.4 版设计。J2SE 的全名为 Java 2 Software Development Kit, Standard Edition, 早期的版本称为 JDK (Java Development Kit), 直到 1.2 版之后才改称 Java 2。

J2SE 是 Java 最通行的版本。除了 J2SE 之外, Java 2 还有 J2ME (精简版, micro edition) 与 J2EE (企业版, enterprise edition)。J2ME 是移动商务最佳的应用典范, 不论是无线通讯、手机、PDA 等小型电子装置均可采用 J2ME 做为开发工具及应用平台。J2EE 则提供了企业 e-Business 架构及 Web Services 服务, 其优越的跨平台能力与开放的标准, 深受广大企业用户的喜爱。

1-2 Java 的虚拟机

大部分的程序语言都必须先经过编译 (compile) 或解释 (interpret) 的操作才能在计算机上运行, 例如 C/C++ 与 Pascal 等是属于编译式的语言, 而 Basic 与 Lisp 等则是属于解释式的语言。

然而, Java 程序却比较特殊, 它必须先经过编译的程序, 然后再利用解释的方式来运行。通过编译器 (compiler), java 程序会被转成与平台无关 (platform-independent) 的机器码, Java 称之为“字节码” (byte-codes)。通过 Java 的解释器 (interpreter) 便可解释并运行 Java 的字节码。

图 1-1 说明了 Java 相关的执行流程:

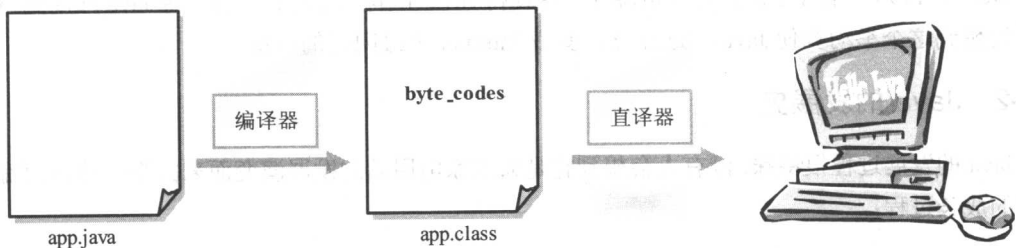


图 1-1 Java 程序运行的过程必须先经过编译, 然后解释

任何一种可以运行 Java 程序的软件均可看成是 Java 的“虚拟机”(Java virtual machine, JVM), 如浏览器与 Java 的开发工具等皆可视为一部 JVM。很自然的, 可以把 Java 的 byte-codes 看成是 JVM 所运行的机器码 (machine code)。

byte-codes 最大的好处是可跨平台运行, 也就是说, Java 的 byte-codes 可让“编写一次, 到处运行” (write once, run anywhere) 的梦想成真。当用任何一种 Java 编译器编译成 byte-codes 时, 便可在含有 JVM 的平台上运行, 无论是 Windows、Mac OS 或 Unix, 如图 1-2 所示。这种跨越平台的特性, 也是让 Java 急速普及的原因之一。

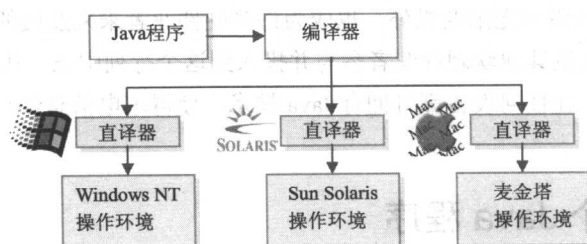


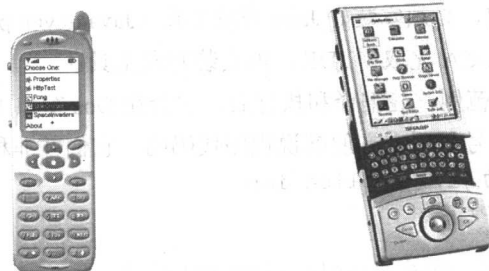
图 1-2 Java 的 byte-codes 可跨越平台来运行

Java 推出后，马上成为世界各种媒体及计算机业界所关注的焦点。经过了数年的成长，Java 的功能不断地增强，使它成为最流行的程序语言之一，并得到越来越多的 IT 相关公司、企业的支持。

1-3 Java 的未来发展

现在 Java 语言的应用非常广泛，在未来几年，会对整个计算机业，甚至家电业产生巨大而深远的影响。它不仅改变计算机的价格和形式，还影响我们对计算机产品的观念。1995 年 12 月，Sun 公司召开记者会时曾公布一长串的名单，包括 IBM、Apple、DEC 等都同意支持 Java。在 2001 年 2 月，微软总裁比尔盖兹（Bill Gates）在一场演讲中说：“Java 是用来推翻我们以前所创建的东西。”所以微软推出 VBScript 这套程序语言，比尔盖兹承诺要用这套语言和 Java 并驾齐驱。可是在 VBScript 还没诞生前，微软只能宣称他们已向 Sun 公司购买 Java 的使用权限并扩充其功能。对于微软这个举动，从当时的股票市场就可以看得出来影响力多大，微软对手 NetScape 的股票在一天之内重挫了 28 块美金。NetScape 的安德瑞森总裁说：“当熊和鳄鱼有冲突时，看是在谁的地盘就是胜败关键处，而微软刚才跑到我们的范围来了。”

由于 Java 未来的应用不仅在动态网页中呈现，换句话说，也就是除了大家所熟知的 Java applet 之外，Java 的应用程序将会扮演更重要的角色，所以 Java 开发工具的类型一定具备多样化，才能够提供给应用程序更完整的使用功能。还有现在每个人的必备品——手机，也将使用 Java 来扩展。在 2001 年 10 月 Nokia 宣布与 SUN 合作开发后，其他厂商（如摩托罗拉等）也表示将陆续推出 Java 手机，以刺激非语音市场的成长，预计相关功能可以如期完成，Java 风潮将再度起飞。Java 手机与 PDA 如图 1-3 所示。



Sharp Zaurus PDA 图片，转载自 java.sun.com

图 1-3 使用 Java 操作系统的新手机与 PDA 问世，令人耳目一新。

Java 除了给软件业者带来无限商机外，也成为许多硬件业者未来发展的重要指针。Java 不断地发展、完善，吸引了大量其他软硬件业者参与并投入到这个行列中来。其次 Java 语言的可移植性、安全性、实用性等，让任何设备都可拥有 Java 技术，使得家电信息化、网络化、智能化在未来成为可能。

1-4 编写第一个 Java 程序

Java 程序可分为下列两种：(1) Java application，(2) 应用在 www 上的 Java applet。

Java application 是指可以在 Java 平台上独立运行的一种程序，通常称之为 Java 应用程序，而 Java applet 则是内嵌在 html 文件中，必须搭配浏览器来运行，因此有些人称 applet 为网页向导，实不为过。Java applet 的编写方式与 Java application 类似，因此只要熟悉 Java application 的编写方式，很快就能学会编写 Java applet。本书除了第 19 章是介绍 Java applet 的编写之外，其余章节所介绍的程序均是属于 Java application。

在开始编写程序代码之前，请先在硬盘 C 中创建一个新的文件夹，并设文件夹名称为“Java”。本书所有的范例均假设存盘于 C:\Java 这文件夹下，如图 1-4 所示。

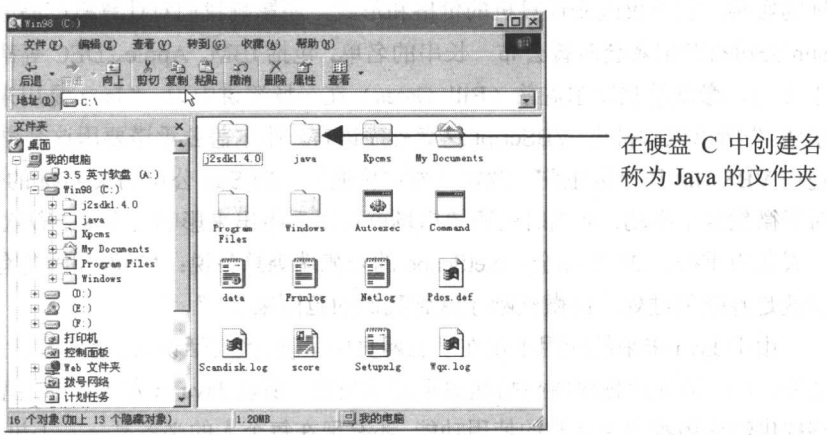


图 1-4 本书所有的范例均假设存盘于 C:\Java 文件夹里

1-4-1 编译与运行 Java application

本节将开始练习编写 Java 程序了。首先介绍如何以最简单的方式来编写、编译与运行 Java application。在编写程序之前，必须安装好 Java 开发工具（Java development kit, JDK），若此时还没完成这个步骤，请参阅附录 A 先取得 JDK，再安装开发工具。

app1_1 是个相当简单的范例，经编译和执行后，它会在 Dos 窗口上显示 Hello Java!! 字符串。app1_1 的程序如下（注意行号是用来方便解说程序代码的，它们并非程序代码的一部分）：

```
01 // app1_1, 简单的 java application
02 public class app1_1
03 {
04     public static void main(String args[])
05     {
06         System.out.println("Hello Java !!");
07     }
08 }
```