

内附光盘

袁葆宏 刘丰豪 吴建和 编著

Java2

语言实务

能让您真正驾驭Java语言的好书

- 集中焦点介绍Java的核心内容
- 详尽讨论对象及面向对象的概念
- 循序渐进剖析Java语言实际制作对象的相关机制
- 完整介绍Java高级应用的必备基础：I/O、package、exception及Thread等课题
- 各章节均以详实范例配合介绍，让您理清概念

Java2 Java2 Java2 Java2 Java2
Java2 Java2 Java2 Java2 Java2
Java2 Java2 Java2 Java2



Java
系列丛书

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

Java2 语言实务

袁葆宏 刘丰豪 吴建和 编著



Java2 语言实务 袁葆宏 刘丰豪 吴建和 编著
 中国铁道出版社 100051
 北京 宣武门大街 36 号
 电话：(010) 51873000
 邮编：100045
 定价：39.00 元

中国铁道出版社
 2001 年·北京

(京)新登字 063 号

北京市版权局著作权合同登记号: 01-2001-4264 号

版 权 声 明

本书中文繁体字版由台湾旗标出版股份有限公司出版(2001)。本书中文简体字版经台湾旗标出版股份有限公司授权由中国铁道出版社出版(2001)。任何单位或个人未经出版者书面允许不得以任何手段复制或抄袭本书内容。

本书贴有旗标出版激光防伪标签,无标签者不得销售。版权所有,侵权必究。

图书在版编目(CIP)数据

Java2 语言实务/袁葆宏,刘丰豪,吴建和编. —北京:中国铁道出版社,2001.11

ISBN 7-113-04406-9

I. J… II. ①袁…②刘…③吴… III. JAVA 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2001)第 075034 号

书 名: Java2 语言实务

作 者: 袁葆宏 刘丰豪 吴建和

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: 苏 茜

特邀编辑: 李晓霞

封面设计: 孙天昭等

印 刷: 北京兴顺印刷厂

开 本: 787×1092 1/16 印张·20 5 字数: 492 千

版 本: 2001 年 11 月第 1 版 2001 年 11 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-04406-9/TP·627

定 价: 36.00 元

版权所有 盗印必究

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。

出版说明

Java 自 1995 年推出至今，已经历了六个年头的发展。而且随着网络的快速发展与普及，Java 的主流地位已形成，并已成为网络及电子商务系统中不可或缺的角色。然而，十分遗憾的是，Java 在中国的发展及使用却是远远不如欧美！究其原因，主要是因为目前出版的 Java 中文书籍大都没有对 Java 的核心做系统深入地介绍。使得一些学生及一般社会大众学习不得要领，只好放弃对 Java 系统的学习。

本书最突出的特点是：它投注了全书百分之八十以上的份量在彻底解说 Java 程序语言的各种面貌。尤其要强调的是，本书每章均有大量的程序范例来说明各种概念。通过范例的说明，读者可以有效地了解 Java 的各种层面，进而快速地掌握 Java 程序语言的精神，以利于后续 Java 的各种功能，如 AWT, SWING, SERVLET 的学习及开发。

为了方便读者学习，随书附赠一张光盘其中收录了各章范例、文字编辑器 EditPad lite、及 JDK1.3.0 update2，让您的学习能事半功倍。

本书由台湾旗标出版股份有限公司提供版权，中国铁道出版社计算机图书项目中心审选；童冠圣、罗怡、葛兰、彭涛、肖志军、廖康良、裘伟力、陈贤淑等同志完成了本书的整稿及排版工作。

中国铁道出版社

2001 年 11 月

目 录

第 1 章 什么是 Java	1
1-1 Java 的简介	2
1-2 第一个 Java 程序	4
1-3 Java 程序的基本结构	8
本章摘要	10
第 2 章 程序语言基本概念及数据	11
2-1 程序、程序语言、程序结构	12
2-2 变量与数据类型	15
2-3 常量(Constants)	20
2-4 表达式、运算符和操作数	21
指定运算符 (assignment operator)	22
算术运算符	25
比较运算符	26
逻辑运算符 (logic operator)	29
类型转换(Type Conversions)	31
本章摘要	33
第 3 章 控制程序的行为	35
3-1 判断语句: if 及 if-else, switch-case	36
单一条件判断	36
双条件判断	39
多重条件判断	41
switch 的多重条件判断语法	45
3-2 循环(Loop)语句: for ,while ,do-while	48
for 循环语句	49
while 循环语句	51
do-while 循环语句	54
3-3 分支 (Branching) 语句: break,continue	56
break 分支语句	56
continue 分支语句	57
3-4 子程序调用(Function Call)	58
本章摘要	61

第 4 章 数组 (Array)	63
4-1 何谓数组	64
4-2 数组的声明及使用	65
4-3 二维数组(2D-Array)	70
本章摘要	74
第 5 章 以对象为基础的程序设计	75
5-1 类(class), 对象(object)/instance 概念简介	76
5-2 对象的基本概念	78
"."运算符,参考类型(Reference Type)以及 New 运算符	78
原始数据类型(Primitive Type) vs.参考数据类型(Reference Type)	79
New 运算符	80
5-3 生存范围 (Scopes) 及区块 (Block)	83
类生存范围 (Class Scope)	87
区域生存范围 (local scope)	89
5-4 对象/类变量(Instance / Class Variable)	91
Instance Variable (对象变量)	92
Class Variable (类变量)	94
instanceof 运算符	96
垃圾收集 (Garbage Collection): 高级概念	97
5-5 Methods (方法)	98
对象的行为	98
对象方法(Instance Method)/类方法(Class Method)	107
Main 方法	109
Constructors (构造函数)	109
方法重载 (Method Overloading)	111
使用重载构造函数 (Overloaded Constructor)	113
5-6 以对象为主的程序设计	115
存取限制符 (Access Modifier): Public / Private	116
this 关键字	120
本章摘要	124
第 6 章 面向对象程序设计	127
6-1 面向对象程序设计简介	128
6-2 继承 (Inheritance)	129
存取限制: Public, Protected, Private 及默认存取	134
变量覆盖(Overriding) 与方法隐藏	138
super 与父类中的方法及构造函数	146

Final 方法与类	150
6-3 抽象类(Abstract classes)及抽象方法(Abstract Method).....	151
6-4 Interface	153
什么是 Interface	153
implements (实现)	154
interface 的继承	156
6-5 多态(Polymorphism)	158
6-6 Object 类: 所有 Java 类的父类	167
Object 类	167
Object 类中所定义的方法	167
本章摘要	177
第 7 章 异常处理(Exception Handling)	179
7-1 什么是“异常”(Exception)	180
7-2 “异常”的种类	180
7-3 如何指定“异常”的处理方式	181
7-4 打印“异常”对象的说明	182
7-5 使用 throws 语句	183
7-6 使用 throw 语句	185
7-7 Throwable 类层级(Class Hierarchy)	186
7-8 自定“异常”类	186
7-9 Java 内建的“异常”类	188
7-10 Java Runtime 如何决定“异常”处理例程	189
本章摘要	191
第 8 章 包(Package)	193
8-1 什么是程序单元(.class 文件)	194
8-2 什么是 package (包).....	195
如何定义 package	195
8-3 如何在项目中定义 package	196
8-4 如何编译及运行 package 中的程序	197
8-5 如何产生 jar 文件	199
8-6 如何使用其他的 package 及类来开发程序	199
8-7 类在 package 中的存取限制	201
默认存取	202
本章摘要	203
第 9 章 字符串处理	205
9-1 基本概念	206



字符	206
字符串	206
9-2 String 与 StringBuffer	208
String 类	210
StringBuffer 类	216
本章摘要	221
第 10 章 输入及输出处理	223
10-1 文件的基本概念	224
10-2 File 类	225
10-3 Stream 介绍	230
10-4 Stream 类	230
10-5 循序文件的建立与存取	232
FileInputStream	232
FileOutputStream	234
再谈 stream 类	236
DataOutputStream	237
DataInputStream	239
10-6 随机文件的建立与存取	241
10-7 如何把对象写入文件	245
本章摘要	248
第 11 章 Java 线程 (Thread) 简介	251
11-1 什么是“线程”(Thread)	252
11-2 线程的程序结构	253
方法一: 建立 Thread 类的子类, 改写 run() 方法	253
方法二: 定义接口 Runnable, 改写 run() 方法	255
11-3 线程的生命周期 (Life cycle)	257
11-4 Thread 类介绍	258
11-5 线程的同步协调	262
使用 synchronized 保留字	262
使用 wait() 及 notify() 方法	265
11-6 线程优先权 (Thread Priority) 的管理	269
11-7 “线程群组”(Thread Group) 的管理	271
本章摘要	274
第 12 章 Applet	275
12-1 什么是 Applet	276
12-3 Applet 程序结构	276



12-3	Applet 的生命周期.....	277
12-4	运行 Applet 的方式.....	278
12-5	Applet 标记的完整语法.....	282
12-6	读取 Applet 的参数.....	283
12-7	如何将 Application 改写成 Applet.....	285
12-8	Applet 类及常用的方法.....	288
12-9	Graphics 绘图对象.....	291
12-10	多媒体应用.....	294
	显示图形.....	294
	播放音乐.....	296
12-11	文件的存取.....	297
12-12	Applet 窗口更新.....	300
12-13	事件处理 (Event-handling) 简介.....	302
	本章摘要.....	306
附录 A	Java 程序写作规范 (Java Code Conventions).....	309
附录 B	Java 关键字 (保留字).....	313



1-1 Java 的简介

Java语言是美国Sun公司(Sun Micro System)开发出的程序语言,自1991年的绿色计划(Green Project)开始,Java的影响力就一直持续上升。许多的公司从一开始的排斥、怀疑到认识、接受,进而投入甚至鼓吹的例子不甚枚举。以目前 Oracle 数据库公司为例,不但以Java改写其数据库重要部分,更要求公司同仁(包含行销的非开发同仁),都要进行Java语言的训练或认证。所以学习Java语言,不但是信息人员必备的能力,也是进行研究或就业的重要基础。

本书作者任教于大学信息系多年,教授有关程序设计、面向对象及计算机网络的相关基础课程或推广教育。在多年教授Java的经验中,发现初次学习Java的学习者,不论是否有学习其他语言的经验,常苦于难以找到一本好的教材,且现有的书籍多是由翻译而来,其中名词或内容讲述方法晦涩难读。另一常见的教学及学习困扰是,由于现在有很多的学校及自学者是以Java当作第一个学习的计算机语言,而多数有关Java的中文书籍并非由计算机的基础知识开始讨论,因此使得教授者辛苦,学习者茫然。

基于上述理由,本书的提供是要提供一本教学及自学 Java 程序设计的教材,重点是循序渐进引导学习的人能设计 Java 程序,进而能够具有熟悉 Java 应用的基础。所以在学习 Java 的过程中,其发展历史及时间可以参考,但并不是本书的重点,请读者上网^[注 1]或到其他地方,查阅相关数据。而有关 Java 产生的背景及其语言的特性,却在学习及了解 Java 上是非常重要的。我们从相关数据,尤其是 Java 白皮书^[注 2],整理及引申 Java 语言的特性如下:

- ① 简单(Simple): 制定 Java 的原则之一,是要建立一种结构简单而且使用容易的系统。可以让用户不必接受艰深的训练就可以开始设计程序,所以 Java 的语法,尽可能与在当时许多程序设计师都采用的 C 及 C++语言相似。并且 Java 也删除了 C 及 C++许多极少使用、不容易理解或常被混淆的功能,例如运算重载(Operator Overloading)、多重继承(Multiple Inheritance)等。如此不但使无程序设计经验的人可以减少艰深功能的学习,而且已有 C 或 C++程序设计的人也能很快的具有以 Java 设计程序的能力。
- ② 面向对象(Object-Oriented): 面向对象是近年来信息界广为使用的概念及技术。它有许多良好的特性,例如对象内包含属性(Attribute)及方法(Method),以有别于传统数据导向(Data Oriented)或控制导向(Control Oriented)的特性。对象的封装性(Encapsulation)可以使对象的接口定义明确,继承性(Inheritance)可以增加软件的可再用性(Reusability)、有助于分类(Classification)及模板(Template)设计等;实现面向对象的概念及其各种良好的特性是 Java 的设计理念之一。
- ③ 分布式(Distributed): 计算机网络的发展使得信息应用更朝向分布式的环境发展。所以现代的信息开发语言及环境,要有能配合分布式的特性及功能。Java 具有一个网络功能的程序库(Library),其中包含与如 HTTP、URL 和 FTP 等 TCP/IP 网络通讯协议整合的能力。利用 Java 并配合网络程序库所开发的程序,能轻易地在网络上打开、连接并使用对象,不需大费周折

[注 1] <http://www.java.sun.com>

[注 2] <http://java.sun.com/docs/white/index.html>



就能如同使用一个局部文件系统(Local File System), 方便地使用各种网络信息。

- ④ **强壮性(Robust):** 由 Java 所编写出的程序要能在各种情况下运行, 而且必须具有高的稳定性。Java 在制定时即加入了能排除存储器被覆写(Overwriting Memory)和数据损坏(Corrupting Data)的相关处理机制。
- ⑤ **安全性(Secure):** Java 是被设计用于网络及分布性的环境中, 所以安全性是一个很重要的考虑。Java 拥有数个从简单到复杂的安全保护措施, 能有效地防止病毒的侵入和破坏行为的发生。
- ⑥ **结构中立性(Architecture Neutral):** 在网络上存在许多不同类型的计算机, 从中央处理器到操作系统结构均有很高的差异性。因此要使应用程序能在每一种机器上均能运行是相当困难的。针对这个目的, Java 的编译器可以产生一种结构中立的码文件格式(Object File Format)-字节码(Byte Code)。这种字节码可以在许多种不同的计算机上运行, 提供一个结构中立特性的解答。
- ⑦ **便携性(Portable):** Java 的各组成环境及以 Java 环境所产生的程序码都具备有相当好的便携性。例如 Java Runtime 是使用标准的 ANSI C 所写成的, 所以可以很容易地在各种不同计算机平台上建立 Java 的环境。Java 的编译器就以 Java 所写成, 所以可以移植到各种平台。Java 程序库中, 更定义了一些可移植的接口, 例如其中的 Windows 对象类(AWT), 通过使用这个对象类, Java 程序可以在 Unix、Windows 和 Macintosh 各种平台上的实现使用接口。Java 开发出来的程序具有好的便携性, 可以在各种计算机上运行。
- ⑧ **直接解译(Interpreted):** Java 具有一个解译器 (Interpreter)可以直接在任何已安装解译器的机器上运行 Java 字节码(Byte Code); 解译器也让程序更容易纠错。
- ⑨ **高性能(High Performance):** Java 在运行时能将字节码迅速地转换成机器码(Machine Code)再加以运行。虽然 Java 增加了许多的特性及功能且是在运行时才把字节码转换成机器码, 但它的运行效能和以 C 与 C++等传统语言所开发出来的程序几乎与没有分别。
- ⑩ **多线程(Multithreaded):** 在 Java 中, 一个线程就代表一个运行工作。在我们周围的真实世界中, 许多事情会同时发生而需要同时间进行。所以若要如同真实世界一般, 设计一个同时处理许多事件的 Java 程序, 就要把 Java 制定成具有多线程的能力。
- ⑪ **动态(Dynamic):** 为了要能因应随时变化的环境, Java 具有可因需要增加新功能模组到程序库或运行环境中了的扩充及延伸的能力。现存的 Java 程序可以直接使用既有及新增的功能, 不会因程序库的更新, 而必须重新编译程序。

以程序语言的角度来解释 Java, 只能发觉 Java 一小部分的特性。Java 也是一个环境, Java 的应用范畴可以从程序语言到硬件的芯片, 也可从操作系统到应用程序都包含其中。Java 的设计环境, 结合了网络新世代的需要、软件工程的良好特性、信息电子的应用等许许多多令人期待的特色。我们期待由 Java 和其他新技术的引导, 可以把信息世界引领到一个更美好的境界。



- ① 编写程序：程序设计者根据要完成的功能,使用 (1) “编辑器” (Editor)，写出 Java 程序的源代码(Source Code)，这个源代码的内容是依照 Java 所规定的“语法”所写的。
- ② 编译程序：把源代码交给 (2) 编译器(Compiler)，编译成计算机看得懂的方式。这个步骤的结果会产生一个文件称为目的文件 (Object File)，并保存到磁盘中。在 Java 中的目的文件很特别，因为它要配合能多在不同平台上运行的特性，所以有一个特殊的格式，它的名称称为字节码(Byte Code)。
- ③ 运行程序：在运行时用 (3) 运行器(包含载入器 Loader)自磁盘取出目的文件到计算机中，检查计算机的运行环境无误后，目的文件正式的成为运行程序并开始运行。
- ④ 纠错及修改：在发现程序有错误或有其他原因想要更改程序时，重新回到步骤一修改源代码程序并重新完成程序发展中的其他步骤 (Debug)，直到正式完成运行程序后停止发展步骤。

在了解计算机程序的发展环境及步骤后，现在我们要开始动手在计算机上完成我们的第一个程序了。我们将要让它能够在计算机屏幕上显示“Java is a good language!”，这是我们的第一杯爪哇(Java)咖啡，希望香醇好喝！

为了要让大家能一步一步照着范例完成第一个程序，请读者对应图 1-2 程序开发过程的概念，并配合本书的 Java 开发和运行环境—MS-Windows NT/9X；范例程序的开发步骤如图 1-3 所示，说明如下：

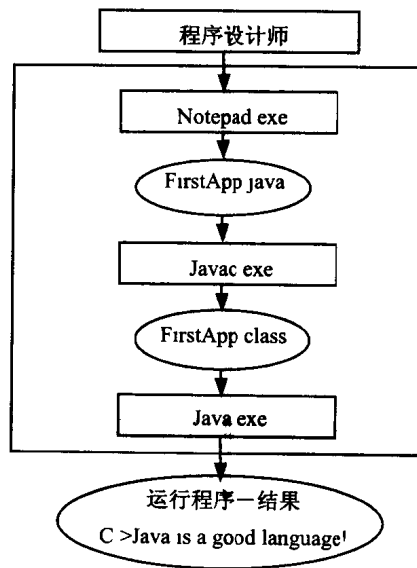


图1-3 范例程序的开发步骤

编写程序

我们可以使用 MS-WINDOWS 所内含的简单工具 Notepad.exe (记事本) 产生源代码程序。其他工具只要是能够编辑文字文件的也可以作为编辑器，但请注意不要使用会加入控制信息在文件中的编辑工具(如 MS-OFFICE 中的 WORD)。





CHAPTER

程序语言基本

概念及数据

JAVA

2

顺利申请到户口簿，有两件事物必须明白的告诉申请的人，第一件事是申请的“流程”。所以一般在服务台上有一份申请的流程图表，告诉申请人要依次到哪些地方或窗口，并一个接着一个步骤地完成相关事项。但是这样并不保证可以顺利地完成任务，例如，因为你可能到达申请的某个窗口才发现少带了需要的身份证，所以只好中断申请，等到带了身份证再继续下面的流程，但是下面的另一个步骤可能又发现缺少了如照片或其他相关的资料，所以一直无法完成申请。这样的情形告诉我们，要完成一件事情，只有步骤流程或只带资料是不够的，必须要包含所运行的流程及相关的资料。

同样地，一个 Java 程序（其他程序也都一样）的程序内容分为两大部分，分别用以描述要处理的流程及所需的数据，流程决定了程序如何去处理相关数据。下面我们举一个例子，说明程序的结构、流程及数据的关系：

范例

程序 2-2 DemoArch.java

```
001: class DemoArch {
002:   public static void main(String args[]) {
003:     int Payrate=1385;    //工作时薪
004:     final int Hours=24; //工作时数
005:     int Payment;        //给付薪资
006:     Payment = Payrate*Hours;
007:     System.out.println("My payment is " + Payment +
008:       " dallors.");
009:     // 在屏幕上显现出内容
010:   }
011: }
```

运行结果

```
My payment is 33240 dallors.
```

范例说明

范例 2-2 中，程序的第 1 行

```
class DemoArch {
```

将程序命名为“DemoArch”，其中“class”是 Java 固定的语法，读者在一开始学习 Java 时可以暂时不要去理会它的意义，只要记着它在程序结构中的位置，我们会在第 5 章中，详细介绍它的用法及相关内容。

语法说明

变量声明中的“数据类型”，是用以指定变量所要存储的数据值的类型；“变量名称”是指定变量的名字如 Payment；“[=初始值]”是可以存在或不存在的部分，如在范例中 `int Payrate=1385`；是初始值存在而符合 Java 语法的情形，另 `int Payment`；则是初始值不存的情形，在但也符合 Java 语法。

Java 语言预先定义好可以直接使用的数据类型，称为“基本数据类型” (Primitive Data Types)，如表 2-2 示；相对于基本数据类型，有一种非基本的参考类型，会在后续的章节中介绍。

声明变量时，指定了变量的基本数据类型(Primitive Data Types)，就等于同时完成了变量所占内存空间大小的设定(就像要取得装东西的箱子时，先声明东西的种类与大小一样) Java 语言有八种基本数据类型，我们将这八种基本数据类型分成整数(integer)、实数(real numbers)及其他(others)三类，并用表 2-2 列出各种基本数据类型所占用的内存长度及合法值的范围：

表 2-2 基本数据类型

基本数据类型名称	占用内存长度	最小值	最大值	默认值
整数(integer)				
字节(byte)	8-bit	-128	+127	(byte)0
短整数(short)	16-bit	-2^{15}	$+2^{15} - 1$	(short)0
整数(int)	32-bit	-2^{31}	$+2^{31} - 1$	0
长整数(long)	64-bit	2^{63}	$+2^{63} - 1$	0L
实数(real number)				
浮点数(float)	32-bit	IEEE754	IEEE754	0.0f
双精度数(double)	64-bit	IEEE754	IEEE754	0.0d
其他(others)				
字符(char)	16-bit	'\u0000'	'\uFFFF'	'\u0000'(null)
布尔变量(boolean)	1-bit	-	-	false

表 2-2 的 Java 基本数据类型说明如下：

- ① 第一类整数类的数据类型有 byte(字节)、short(短整数)、int(整数)、long(长整数)四种，因为是整数所以可以包含正负整数及 0，各由所占用的内存大小对应出可以表示数值极限的大小。首先 byte 的大小为 8 位，用来存放 8 个位长度的数字，所以可表示 2 的 8 次方种可能值，对应出来的值为从最小值为 -128 到最大值为 127 共 256 个值，其中默认值为 0(所谓默认值就是声明一变量后若无指定值，由计算机自动指定给该变量的初始值)。short、int、long 也可以存放整数，四者的差异为该数据变量的存放空间大小不同，请对应表 2-2 的内容，空间大者可以表示的范围值较大，但在存储及运行时所占用的计算机空间就较大且运行时会耗用较多的时间。
- ② 第二类的基本数据类型为实数，是用来表示含有小数点数值的数据，包含有 float(浮点数)、double(双精度数)两种数据类型；float 是浮点数(floating point)的数据类型。


```
019: System.out.println("The largest integer value is " +
020:     largestInteger);
021: System.out.println("The largest long value is " +
022:     largestLong);
023: System.out.println("The largest float value is " +
024:     largestFloat);
025: System.out.println("The largest double value is " +
026:     largestDouble);
027: System.out.println("The value of aBoolean is " +
028:     aBoolean);
029: }
030: }
```

程序

运行结果

```
The largest byte value is 127
The largest short value is 32767
The largest integer value is 2147483647
The largest long value is 9223372036854775807
The largest float value is 3.4028235E38
The largest double value is 1.7976931348623157E308
The value of aBoolean is true
```

范例说明

在范例 2-2 中的程序内容中程序 1~2 行

```
public class DemoMaxVariables {
    public static void main(String args[]) {
```

为类“DemoMaxVariables”及方法“main”的声明。

第 3 行:

```
// integers
```

为注释,说明其后的范例为“整数”相关的说明。

第 4 行: