

别让你的计算机闲着



探秘 JAVA

如何像计算机科学家一样思考

How to Think Like a Computer Scientist
Java Version

[美] Allen Downey 著
张平译



浙江大学出版社

ZHEJIANG UNIVERSITY PRESS

别让你的计算机闲着

TP312JA
286

TP312JA
286

探秘 Java

——如何像计算机科学家一样思考

[美] Allen Downey 著

张 平 译

浙 江 大 学 出 版 社

图书在版编目(CIP)数据

探秘 Java: 如何像计算机科学家一样思考/(美)唐尼(Downey, A.B.)著; 张平译. —杭州: 浙江大学出版社, 2004.12

(别让你的计算机闲着)

ISBN 7-308-04036-4

I. 探... II. ①唐... ②张... III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 119868 号

Copyright © 2003 Allen Downey

Permission is granted to copy, distribute, and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with Invariant Sections being “Preface”, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the appendix entitled “GNU Free Documentation License”.

本书依据 *How to Think Like a Computer Scientist(Java Version)*—书译出

丛书策划 张明

封面设计 张作梅

责任编辑 张明 宋铮

出版发行 浙江大学出版社

(杭州浙大路 38 号 邮政编码 310027)

(E-mail: zupress@mail.hz.zj.cn)

(网址: <http://www.zjupress.com>)

排 版 浙江大学出版社电脑排版中心

印 刷 杭州杭新印务有限公司

开 本 787 mm × 960 mm 1 / 16

印 张 25.25

字 数 495 千

版 印 次 2004 年 12 月第 1 版 2004 年 12 月第 1 次印刷

印 数 0000—3000

书 号 ISBN 7-308-04036-4 / TP · 277

定 价 37.00 元 (附赠光盘一张)

内 容 简 介

Java 语言是目前最流行的面向对象的编程语言。它具有强大的功能，既能面向一般跨平台的工程开发，又能面向因特网应用，为广大编程者和学习者所青睐。

本书的特点有四：一是用类似英语教学中的情景教学方法，尽可能用简洁明快的方式按部就班地叙述。二是多用读者日常生活中随手可拈的趣味性实例作为编程材料。三是把计算机科学家们在编程实践中总结出来的诀窍“捅”给读者。四是所有的习题均给出答案，并加以简略分析。

本书从结构上可分两部分。其中 1~13 章为 Java 语言的基础部分，循序渐进地介绍 Java 语言的各种要素：变量、操作符、条件语句、循环语句、嵌套与递归、引用、数组、结构、类以及对象等。14~20 章为数据结构部分，分门别类地介绍各种常用的数据结构：链表、堆栈、队列、优先队列、树、堆、映射表、哈夫曼树以及文件等。

另外，本书附赠的光盘含有 Java 编程语言(Sun 公司的 J2SE 1.4.2SDK 和 J2SE 1.5.0 Beta 2)，优秀编程环境 BlueJ，Netbeans 3.6 for Windows，Eclipse SDK 3.0RC2 以及其他一些资料。

阅读本书的读者可以是真正的初学者，可以是在校学习的研究生、本科生或大专生，也可以是企、事业单位的初、中级用户。最合适的读者是非计算机专业的大学本科或大专生，以及中、小学生中信息技术的爱好者。

本书可用为各类学校的计算机课程教科书，也可作为学习计算机编程的参考书。

目 录

前 言.....	1
此书的缘起.....	1
此书的宗旨.....	2
面向对象的程序设计.....	3
数据结构.....	3
计算机 AP 考试.....	4
有关此书的版权.....	5
关于此书书名.....	5
第 1 章 按部就班的方式.....	7
1.1 什么是程序语言.....	7
1.2 什么是程序.....	10
1.3 什么是调试.....	11
1.4 形式化语言和自然语言.....	13
1.5 第一个程序.....	15
1.6 术 语.....	16
1.7 练 习.....	17
第 2 章 变量和类型.....	19
2.1 再打印一些东西.....	19
2.2 变 量.....	21
2.3 赋 值.....	22
2.4 打印变量.....	23
2.5 关键字.....	24
2.6 操作符.....	25
2.7 操作符的执行顺序.....	26
2.8 对字符串的操作.....	27
2.9 组合句.....	27
2.10 术 语.....	28

2.11 练习.....	29
第3章 方法.....	31
3.1 浮点数.....	31
3.2 把 double 类型转换成 int 类型.....	33
3.3 数学函数.....	33
3.4 组合.....	35
3.5 自己编写新方法.....	35
3.6 类和方法.....	38
3.7 具有多个方法的程序.....	39
3.8 形式参数和实际参数.....	40
3.9 堆栈状态图.....	41
3.10 具有多个参数的方法.....	42
3.11 具有结果的方法.....	43
3.12 术语.....	43
3.13 练习.....	44
第4章 条件和递归.....	47
4.1 模除运算.....	47
4.2 条件执行.....	47
4.3 选择执行.....	48
4.4 链式条件.....	49
4.5 嵌套条件.....	50
4.6 返回语句.....	51
4.7 类型转换.....	51
4.8 递归.....	52
4.9 递归调用方法的堆栈图.....	54
4.10 惯例和神圣规则.....	55
4.11 术语.....	56
4.12 练习.....	56
第5章 “开花结果”的方法.....	60
5.1 返回值.....	60
5.2 “逐渐生长”的程序.....	62
5.3 组合.....	65

5.4	重 载.....	66
5.5	布尔函数.....	67
5.6	逻辑运算符.....	68
5.7	布尔方法.....	69
5.8	递归的例子.....	70
5.9	确信跳跃.....	72
5.10	另一个递归例子.....	73
5.11	术 语.....	74
5.12	练 习.....	75
第 6 章	重 复.....	80
6.1	变量的多次赋值.....	80
6.2	重 复.....	81
6.3	while 语句.....	81
6.4	表 格.....	83
6.5	两维表.....	86
6.6	封装和泛化.....	87
6.7	方法.....	88
6.8	进一步封装.....	88
6.9	局部变量.....	89
6.10	进一步泛化.....	90
6.11	术 语.....	92
6.12	练 习.....	92
第 7 章	字符串和其他.....	96
7.1	调用依附于某个对象的方法.....	96
7.2	长 度.....	97
7.3	遍 历.....	98
7.4	运行错误.....	99
7.5	阅读说明文档.....	99
7.6	indexOf 方法.....	100
7.7	循环和计数.....	101
7.8	加一和减一操作符.....	103
7.9	字符的算术.....	104
7.10	不可更改的字符串.....	105

7.11	字符串之间不能直接比较.....	106
7.12	术 语.....	107
7.13	练 习.....	108
第 8 章 趣味对象.....		113
8.1	有趣的是什么.....	113
8.2	包.....	113
8.3	Point 对象.....	114
8.4	实例变量.....	115
8.5	把对象作为参数.....	116
8.6	矩 形.....	116
8.7	把对象作为返回值.....	117
8.8	对象是可变动的.....	118
8.9	别 名.....	119
8.10	空引用.....	121
8.11	垃圾回收机制.....	122
8.12	对象和基类.....	123
8.13	术 语.....	123
8.14	练 习.....	124
第 9 章 定做对象.....		129
9.1	类的定义和对象类型.....	129
9.2	Time 类.....	130
9.3	构造器.....	131
9.4	编写多个构造器.....	132
9.5	生成新对象.....	133
9.6	打印对象.....	134
9.7	操纵对象.....	135
9.8	无瑕作用.....	135
9.9	改动作用.....	137
9.10	填入作用.....	139
9.11	哪一种更好.....	139
9.12	发展型风格 vs 规划型风格.....	140
9.13	泛化处理.....	141
9.14	算 法.....	142

9.15 术语.....	142
9.16 练习.....	143
第 10 章 数组.....	147
10.1 数组元素的存取.....	148
10.2 数组的拷贝.....	149
10.3 for 循环.....	150
10.4 数组和对象.....	151
10.5 数组长度.....	151
10.6 随机数.....	152
10.7 随机数数组.....	153
10.8 计数.....	154
10.9 频率直方图.....	156
10.10 一次遍历解决问题.....	156
10.11 术语.....	157
10.12 练习.....	157
第 11 章 对象数组.....	162
11.1 各种组合.....	162
11.2 扑克牌对象.....	162
11.3 printCard 方法.....	164
11.4 sameCard 方法.....	166
11.5 compareCard 方法.....	168
11.6 扑克牌数组.....	169
11.7 printDeck 方法.....	171
11.8 寻找和搜索.....	171
11.9 一整副牌和一部分牌.....	175
11.10 术语.....	176
11.11 练习.....	176
第 12 章 数组对象.....	178
12.1 deck (一副纸牌) 类.....	178
12.2 洗牌.....	180
12.3 选择排序.....	181
12.4 一手牌 (一部分牌).....	182

12.5	洗牌和发牌.....	183
12.6	混合排序.....	184
12.7	术 语.....	187
12.8	练 习.....	187
第 13 章 面向对象程序设计.....		190
13.1	程序设计语言及风格.....	190
13.2	对象方法和类方法.....	191
13.3	当前对象.....	191
13.4	介绍一种对象——复数对象.....	191
13.5	第一个复数方法.....	193
13.6	另一个复数方法.....	194
13.7	具有改动作用的方法.....	195
13.8	改换成字符串的方法.....	195
13.9	等于方法.....	197
13.10	在对象方法里调用对象方法.....	198
13.11	小心无大错.....	198
13.12	继 承.....	199
13.13	可作图的矩形类.....	199
13.14	类的层级.....	201
13.15	面向对象的程序设计.....	202
13.16	术 语.....	202
13.17	练 习.....	202
第 14 章 链 表.....		204
14.1	对象的引用.....	204
14.2	节点 (Node) 类.....	204
14.3	链表是聚集器.....	206
14.4	链表与递归.....	209
14.5	无穷链表.....	210
14.6	原义含糊定理.....	211
14.7	针对节点的对象方法.....	212
14.8	改动链表.....	212
14.9	“外包装”和“内贤助”.....	213
14.10	IntList 类.....	214

14.11	真实量.....	216
14.12	术 语.....	216
14.13	练 习.....	217
第 15 章 堆 栈.....		219
15.1	抽象数据结构.....	219
15.2	抽象数据结构——堆栈 (Stack)	220
15.3	Java 堆栈对象	220
15.4	打包类.....	223
15.5	创建打包类对象.....	223
15.6	创建更多的打包类对象.....	224
15.7	重新获得打包类的值.....	224
15.8	打包类中的几种方法.....	225
15.9	后缀表达式.....	225
15.10	语法分析.....	226
15.11	抽象数据结构的实施.....	228
15.12	运用数组来实施堆栈.....	228
15.13	变动数组大小.....	230
15.14	术 语.....	232
15.15	练 习.....	232
第 16 章 队列和优先队列.....		235
16.1	队列抽象数据结构.....	236
16.2	装饰板.....	238
16.3	链接型队列.....	239
16.4	循环缓冲区.....	241
16.5	优先队列.....	246
16.6	元 类.....	247
16.7	优先队列的数组实施.....	247
16.8	使用优先队列.....	249
16.9	高尔夫球手记分类.....	250
16.10	术 语.....	252
16.11	练 习.....	253

第 17 章 树.....	255
17.1 树的节点.....	255
17.2 创建树结构.....	256
17.3 周游一棵树.....	257
17.4 表达式树.....	258
17.5 树的遍历.....	259
17.6 对象封装.....	260
17.7 定义 Visitable 接口类.....	261
17.8 接口的实施.....	262
17.9 矢量 (Vector) 类.....	263
17.10 Iterator (遍历) 类.....	265
17.11 术 语.....	266
17.12 练 习.....	267
第 18 章 堆.....	270
18.1 用数组来实施树.....	271
18.2 算法占用资源分析.....	275
18.3 mergesort 排序的分析.....	277
18.4 附加消耗.....	279
18.5 优先队列的实施.....	280
18.6 堆的定义.....	282
18.7 从堆中删除元素.....	283
18.8 向堆中添加元素.....	285
18.9 堆中操作的时间特性.....	286
18.10 堆排序.....	287
18.11 术 语.....	288
18.12 练 习.....	288
第 19 章 映射表.....	290
19.1 数组、矢量与映射表.....	290
19.2 映射表抽象数据结构.....	291
19.3 内置的哈希表.....	291
19.4 用矢量实施映射表.....	294
19.5 链表(List)接口类.....	297

19.6	哈希表的实施.....	297
19.7	哈希函数.....	298
19.8	哈希表的伸缩.....	300
19.9	哈希表伸缩的时间特性.....	301
19.10	术 语.....	302
19.11	练 习.....	302
第 20 章	哈夫曼码.....	305
20.1	变长码.....	305
20.2	字母频率表.....	306
20.3	哈夫曼树.....	308
20.4	super 方法.....	311
20.5	解 码.....	313
20.6	编 码.....	314
20.7	术 语.....	315
20.8	练 习.....	315
附录 A	按部就班开发程序.....	317
附录 B	除 错.....	324
B.1	编译时出现的错误.....	324
B.2	运行时出现的错误.....	328
B.3	逻辑错误.....	331
附录 C	输入与输出.....	337
C.1	System 中的对象.....	337
C.2	键盘输入.....	338
C.3	文件输入.....	339
附录 D	图 形.....	340
D.1	画板和图形对象.....	340
D.2	在 Graphics 类的对象里调用各种方法.....	341
D.3	坐标系.....	342
D.4	米老鼠的脸.....	343
D.5	另外的绘图命令.....	344
D.6	分形的米老鼠.....	345

D.7 有关图形的一些类定义.....	347
D.8 练 习.....	349
附录 E 与 Java 环境混个脸熟.....	354
E.1 下载和安装 Java 语言.....	354
E.2 BlueJ 开发环境.....	356
E.3 有关 BlueJ 的项目.....	361
E.4 有关 BlueJ 的类.....	365
E.5 执行程序.....	369
E.6 程序的调试和排错.....	374
E.7 终端窗口.....	377
E.8 改变设置.....	379
附录 F 太过完美的洗牌.....	383
F.1 52 张纸牌的洗牌.....	383
F.2 n 张纸牌的 AB 顺序洗牌.....	385
F.3 n 张纸牌的 BA 顺序洗牌.....	387
F.4 练 习.....	388
附录 G 练习分析与解答.....	389
附录 H GNU Free Documentation License.....	389

前 言

从其他人的发明中我们获得了巨大的利益，所以，我们也应该让其他人享用自己的发明成果，并且应该是自由享用和无私慷慨的。

——本杰明·富兰克林

此书的缘起

自 1999 年开始，我一直在科尔比学院教书。那时就写了此书，今天呈现在读者面前的已是第四版了。那个时候，当我想用 Java 程序设计语言，向学生们介绍计算机科学的基础知识时，却怎么也找不到一本称心如意的教材。为什么呢？因为这些著作的“块头”都太大了，班上的学生们根本无法阅读这些厚达 800 多页的、满是技术名词术语的“砖头”大书，即使我强制要求他们这么做，他们也还是力不从心。当然我没有理由、也不忍心让他们这样做。同时，这些书中的材料都太专门、太琐碎了——包括了各种各样 Java 语言和库函数的细节——以致到学期结束后，学生也会早已模糊和遗忘。

另外一个是，这些教材在向学生介绍面向对象的程序设计方式时，显得太过唐突和生硬。有许多对学习过程式语言非常适应、成绩也非常好的学生，一旦学到面向对象的设计方法时，就像是碰到了一堵墙，竟然无法逾越。我也尝试在程序语言课一开始时，就教学面向对象的设计方法，或者在中间，甚至在最后，但效果却全都一样。

正由于以上两个因素，我开始撰写此书。我以每天一章的速度写了 13 天，在

第 14 天对此书进行了校正，然后再把它复印装订成册。当我把此书发给学生时，我对学生们说，希望他们每周阅读一章。换句话说，他们的阅读速度是我写作速度的 1/7。

此书的宗旨

以下是有关写作此书的一些想法：

- 术语非常重要。当上课时，学生需要理解程序设计的原理，而下课后，学生需要谈论程序设计的过程。虽然写此书时，我总是尽可能少地介绍各种专门术语，首次遇到时，也句斟字酌地进行讲解，并且在章末以术语表的形式再介绍一遍。但是当我上课时，为了让学生记住和理解术语，经常采用提问、猜谜或者测试的方法，让学生们用短小精悍的语句回答。
- 为了编写程序，学生们总得理解算法，总得知道设计程序的语句，总得自己排除程序错误。我发现，许多书中都缺乏介绍如何进行程序排错的方法。于是，我在此书的附录中加了关于如何排错的一段，也加了如何用“生长”的方式进行程序设计的一段。（用“生长”的方式进行程序设计可以避免许多排错过程。）在这里，我推荐学生及早阅读附录中的这两段，并且经常地反复地“光顾”它们。
- 有些概念的理解需要时间。有些特别难以理解的概念，就不得不多次重复。在书中，递归的概念就是如此处理的。当重复这些概念的时候，学生就又一次进行了复习，加深了印象，而那些第一次接触时没有完全搞懂的话，则可在此时把它搞定。
- 我尝试用最少的 Java 语言知识来获取编写程序的最大威力。而此书的真正目的，是介绍程序设计的一般内容和计算机科学的基础知识，并不仅仅是介绍 Java 程序语言。故在此书中，有时甚至故意放弃了介绍如 switch 语句这样的一些 Java 语句，而且也尽量避免介绍 Java 的库函数，尤其是 AWT，因为它们变动很迅速，而且今后有可能被更好的所替代。

当我写作此书时，运用的是一种最小主义。最小主义有某些优点。每一章的内容，我都将之控制在十几页。当我上课前，都要求学生先在家里预习这一章，然后再在课堂中讨论它。结果是学生不但愿意这样做，而且对课文的理解也非常棒。经过这样的预习，在课堂中就有充分的时间分析和讨论一些比较抽象深奥的问题，

也可在课堂中做一些作业，甚至增加一些本书之外的新内容。

当然，运用最小主义也有某些缺点。使用这样的方式，就无法介绍 Java 语言的一些精髓与独到之处，于是，书中所举的例子都是 Java 语言最基本的应用，而且许多应用例子还需要学生运用数学课和语文课的背景知识。我的主观愿望当然想让计算机语言课上得生动活泼，但是，还是有那些特别使同学们感兴趣和着迷的领域，比如图形、声音和网络等方面的应用，却不得不割爱了。

还必须指出，Java 语言有许多有用特性只与细节有关，而与整体无关。从教育的角度看，这意味着得用大量的课堂时间解释这些细枝末节，但却没有相对应的教学效果。所以，本书选用的教学材料是受学生们欢迎的内容与尽可能丰富的 Java 知识相互结合与平衡的结果。当然，这也为采用此书作为教材的教师留下巨大的调整与发挥的空间。为了补偿，我专门在本书末尾附上了有关图形、键盘输入与文件输入输出等有关的一些内容。

面向对象的程序设计

对于如何向学生们引入、介绍、教学关于面向对象程序设计的特性，各人见仁见智。有些人是开门见山，直入正题；另一些人却是先用过程型的程序设计作为“热身”和过渡，最后才逐渐地引入、介绍、教学关于面向对象程序设计特性。而本书则可归属为“拖拖拉拉”地介绍面向对象程序特性这一类型的极端例子。

Java 的许多面向对象特性，来源于解决一些早期编程语言碰到的难以处理和解决的问题，并且面向对象特性的实现也受到这些历史因素的影响。于是，如果学生们不知道这些问题和历史，则对一些面向对象特性的解释和教学就十分困难。

当然，这并不能成为尽可能拖延介绍面向对象程序设计特性的理由，相反，只要有可能，只要学生理解没有问题，“火候”已到，就应马上介绍面向对象特性。不过，我坚持以最清晰最准确的方式，每次只介绍其中的一项特性，并且留下足够的时间让学生对此概念进行专门理解。只有在这样做之后，我才会讲解和引入下一项面向对象的特性。

通过整理，我发现用 13 步就可以分别介绍 Java 面向对象的特性。

数据结构

从 2000 年秋季学期开始，我又开始执教 Java 语言入门后的后继课程——数据