



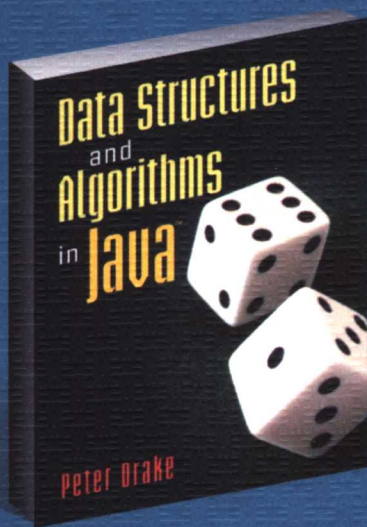
国外经典教材·计算机科学与技术

PEARSON
Prentice
Hall

数据结构与算法

(Java版)

Data Structures and
Algorithms in Java



全新的游戏设计

使用UML

完全的工作代码

Peter Drake 著

朱剑平等译



清华大学出版社

国外经典教材·计算机科学与技术

数据结构与算法（Java 版）

Peter Drake 著

朱剑平 等译

清华大学出版社
北京

Simplified Chinese edition copyright © 2006 by **PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.**

Original English language title from Proprietor's edition of the Work.

Original English language title: **Data Structures and Algorithms in Java** by Peter Drake, Copyright © 2006

EISBN: 0-13-146914-2

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as **Prentice-Hall.**

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 **Prentice-Hall** 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2006-2885 号

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

本书封面贴有 **Pearson Education** (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

数据结构与算法(Java版)/(美)杜兰克(Drake, P.)著;朱剑平等译. —北京:清华大学出版社, 2006.10

书名原文: Data Structures and Algorithms in Java

ISBN 7-302-13798-6

I. 数… II. ①杜… ②朱… III. ①数据结构 ②算法分析 ③JAVA 语言—程序设计 IV. ①TP311.12 ②PT312

中国版本图书馆 CIP 数据核字 (2006) 第 109129 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: 常晓波

印刷者: 北京市清华园胶印厂

装订者: 三河市化甲屯小学装订二厂

发行者: 新华书店总店北京发行所

开 本: 185 × 260 印张: 28.5 字数: 701 千字

版 次: 2006 年 10 月第 1 版 2006 年 10 月第 1 次印刷

书 号: ISBN 7-302-13798-6/TP · 8295

印 数: 1 ~ 3000

定 价: 56.00 元

国外经典教材·计算机科学与技术

编审委员会

主任委员:

孙家广 清华大学教授

副主任委员:

周立柱 清华大学教授

委员(按姓氏笔画排序):

王成山	天津大学教授
王 珊	中国人民大学教授
冯少荣	厦门大学教授
冯全源	西南交通大学教授
刘乐善	华中科技大学教授
刘腾红	中南财经政法大学教授
吉根林	南京师范大学教授
孙吉贵	吉林大学教授
阮秋琦	北京交通大学教授
何 晨	上海交通大学教授
吴百锋	复旦大学教授
李 彤	云南大学教授
杨宗源	华东师范大学教授
沈钧毅	西安交通大学教授
邵志清	华东理工大学教授
陈 纯	浙江大学教授
陈 钟	北京大学教授
陈道蓄	南京大学教授
周伯生	北京航空航天大学教授
孟祥旭	山东大学教授
姚淑珍	北京航空航天大学教授
徐佩霞	中国科学技术大学教授
徐晓飞	哈尔滨工业大学教授
秦小麟	南京航空航天大学教授
钱培德	苏州大学教授
曹元大	北京理工大学教授
龚声蓉	苏州大学教授
谢希仁	中国人民解放军理工大学教授

出版说明

近年来，我国的高等教育特别是计算机学科教育，进行了一系列大的调整和改革，急需一批门类齐全、具有国际先进水平的计算机经典教材，以适应当前我国计算机科学的教学需要。通过使用国外先进的经典教材，可以了解并吸收国际先进的教学思想和教学方法，使我国的计算机科学教育能够跟上国际计算机教育发展的步伐，从而培育出更多具有国际水准的计算机专业人才，增强我国计算机产业的核心竞争力。为此，我们从国外知名的出版集团 Pearson 引进这套“国外经典教材·计算机科学与技术”教材。

作为全球最大的图书出版机构，Pearson 在高等教育领域有着不凡的表现，其下属的 Prentice Hall 和 Addison Wesley 出版社是全球计算机高等教育的龙头出版机构。清华大学出版社与 Pearson 出版集团长期保持着紧密友好的合作关系，这次引进的“国外经典教材·计算机科学与技术”教材大部分出自 Prentice Hall 和 Addison Wesley 两家出版社。为了组织该套教材的出版，我们在国内聘请了一批知名的专家和教授，成立了一个专门的教材编审委员会。

教材编审委员会的运作从教材的选题阶段即开始启动，各位委员根据国内外高等院校计算机科学及相关专业的现有课程体系，并结合各个专业的培养方向，从 Pearson 出版的计算机系列教材中精心挑选针对性强的题材，以保证该套教材的优秀性和领先性，避免出现“低质重复引进”或“高质消化不良”的现象。

为了保证出版质量，我们为这套教材配备了一批经验丰富的编辑、排版、校对人员，制定了更加严格的出版流程。本套教材的译者，全部来自于对应专业的高校教师或拥有相关经验的 IT 专家。每本教材的责编在翻译伊始，就定期不间断地与该书的译者进行交流与反馈。为了尽可能地保留与发扬教材原著的精华，在经过翻译、排版和传统的三审三校之后，我们还请编审委员或相关的专家教授对文稿进行审读，以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和受全体制作人员自身能力所限，该套教材在出版过程中很可能还存在一些遗憾，欢迎广大师生来电来信批评指正。同时，也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材，共同为我国高等院校计算机教育事业贡献力量。

清华大学出版社

前

言

P
R
E
F
A
C
E

本书读者对象

本书计划用于计算机科学的第二门大学生课程。假定同学们熟悉 Java 或 C 的基本控制结构，并且熟悉诸如变量以及将问题分解为函数（方法）之类的初级概念。本书只需要同学们具有微积分之前的数学知识。

为什么选择本书？

本书提供了你想在一本数据结构教材中学到的一切内容：大量的习题、易懂的文字、可在线获得的代码，以及合适的深度。当然，每位作者都会做出同样的声明。那么又是什么使得本书与众不同呢？

本书具有以下出众的特点：

- 使用 Java 1.5 的新特性。
- 提供了数量极多的图形，它们都是用统一建模语言（Unified Modeling Language, UML）绘制的。
- 具有“倒金字塔”风格，以及预先学习最重要的内容。
- 循序渐进地介绍抽象的概念。
- 广泛使用游戏作为示例。

Java 1.5 版在该语言中增加了多种人们期盼已久的特性。在全书的许多示例中解释和使用了这些新特性。具体地讲：

- 新的增强型 for 循环允许程序员简明地提示：“对这个数组或集合的每个元素……”。在附录 A 中以及第 5 章介绍集合时将讨论这种循环。
- 现在自动对基本类型进行装箱和取消装箱，从而把程序员从如下笨拙的构造中解脱出来：

```
((Integer)(numbers.get(i))).intValue()
```

在第 2 章中讨论引用类型时将会介绍这种新特性。

- 新的 `java.util.Scanner` 类（附录 A）最终使得从键盘读取输入成为可能，而不必编写大量的代码。

- 最大的改变可能是引入了泛型类型 (generic type) (第 4 章), 它允许程序员指定集合的元素类型, 从而提高了程序清晰性和类型安全性。

本书在解释概念时使用了许多图形, 因为过长的文字、代码和方程式会使读者感到枯燥。而好的图形可以清楚地解释新概念, 提供即时的回顾, 并且可以作为回顾书中内容的醒目的标志。因为统一建模语言 (UML) 已经成为软件图形事实上的标准, 所以本书中的图形都是用 UML 的子集绘制的。本书甚至有意省略了一些中间的 UML 特性, 如访问级标签和聚合菱形, 因为我感觉它们引起的混淆更甚于所带来的清晰性。这种表示法是逐步引入的 (主要集中在本书的第 I 部分), 并在附录 B 中进行了回顾。

相关的书写特性是在代码中使用加重的字体。许多文本要么没有提供字体突出显示, 要么使用字体来突出显示关键字。当输出代码时, 这种语法突出显示是有用的 (有助于防止印刷错误)。但是, 它既不会使代码更易读, 也不会强调重要的过程。本书使用粗斜体文本来突出显示当前讨论中特别感兴趣的代码部分。如果没有什么内容要特别强调, 就只突出显示方法和字段名称, 以描绘出类的主要部分的轮廓。

用新闻记者的话来讲, 本书是采用“倒金字塔”风格编写的: 最重要的内容出现在最前面, 更细微的细节和更高级的主题分布在各章中介绍。在学完任意一章之后, 都可以合理地停止讲授课程。这使得老师能够根据需要自由地加快或放慢进度, 而不必担心在课程结束之前没有涉猎到重要的主题。

最前面的许多文字由于过于抽象会使读者感到压抑。这是经过充分培训的计算机科学家很容易犯的一个错误, 他们在深入研究细节之前, 更喜欢阅读总的纲要。另一方面, 如果学生在编写一个“真正的”程序之前, 必须全神贯注地学习继承、多态性、递归和算法分析, 那么他们很可能失去兴趣 (如果不是失去知觉的话)。我常常听到学生要求更具体的示例, 而从未听到一个人抱怨示例太多。

在时刻谨记这一点的情况下, 我尽力在每一章中及早提供示例。让学生尽可能早地与完整、有效的程序打交道。在具体问题的环境中消化抽象的概念要容易得多。

困难的概念也是尽可能循序渐进地介绍的。例如, 在调用栈出现在递归的环境中之前就会对其进行讨论。

本书的大多数示例都是游戏, 通常涉及骰子、纸牌和棋盘。游戏激发了学生的想象力, 给他们提供了关心正在讨论的数据结构或算法的某种理由。虽然这些游戏的编程工作足够简单, 无需编写一页页的代码, 但是与找出最大公约数之类的客观任务相比, 它们提出了更现实的挑战。在接近书的中间位置, 一个涉及字典 (其中包含有数万个单词) 的游戏提供了学习高效数据结构和算法的激发人的动机。

本书的特点是在本书的开始处就完成了每个项目的开发, 而且通常提供了代码的多个版本, 并考虑了替代的设计。学生因此会经历精心设计程序的过程, 而不仅仅是热衷于得到结果。虽然程序的重要部分都得到了详尽的介绍, 但是本书把某些增强功能 (如检查有效输入) 留做复习题。

本书给出了每个程序完整、有效的代码, 并且可在线获取它们。我记得, 对一名学生来说, 当把算法或数据结构的困难部分留做习题时, 会使他倍感苦恼。所有的代码全都包括在本书中, 甚至 B 树的代码也是如此; 我知道没有任何其他的大学生教材这样完整地提供这方面的内容。

使用本书中的代码并不需要预先下载任何类。后面的类利用了前面的类。如果需要，可以把后面的这些类轻松地修改成使用来自 Java 集合框架的内置类；本书中开发的类使用了一致的方法名称。

希望同学们在学完本书后会获得相当多的技能来快速开发正确、高效、通用的程序。

本书的组织结构

本书分为 5 个部分：面向对象程序设计、线性结构、算法、树和集合，以及高级主题。

第 I 部分介绍面向对象程序设计，用 3 章分别介绍 3 个重要的原则：封装、多态性和继承。各所学校对本书这一部分的使用有极大的差异。如果某所学校在第一门计算机课程中就讲授了对象，则可以非常快地介绍这一部分，或者完全跳过不讲（尽管绝大多数学生都会通过复习这部分内容而获益）。另外一些学校的第一门计算机课程是用 C 或其他某种语言讲授的，那么这可能就要在这部分内容上花更多的时间。另外，本书的这一部分（以及附录 A）适合于介绍面向对象程序设计的短期课程。

第 II 部分介绍栈、队列和表，并为它们提供了基于数组的实现和链表实现。读者将会学到如何使用这些结构以及如何构建它们，然后将会学到在 Java 集合框架中的什么位置找到它们。

在第 III 部分，读者将开始从单纯的编程步入计算机科学的旅程。这一部分将会介绍算法分析，包括渐近表示法和分析简单算法的逐步过程。在学完这些困难的内容之后，读者将会学习介绍最简单的查找和排序算法的相对较短的一章，这时可以轻松一下。到介绍递归时，读者已经理解了调用栈、排序问题，并且能够感知到递归算法（如归并排序和快速排序）提供的更好性能。

第 IV 部分重点关注数据结构，介绍了树和集合实现（有序表、二叉树和散列表）。

第 V 部分介绍高级主题，教师可从中自由选择他们感兴趣的主体，包括高级线性结构、串、高级树、图、内存管理，以及磁盘存储涉及的问题。

习题、复习题和项目

大多数小节的末尾都提供了习题。求解这些习题应该花不了几分钟的时间。习题适合于在课堂上或考试中提出。

各章的末尾都提供了复习题。它们稍微棘手一些，可能要花 5~10 分钟来求解。这些复习题适合于作为考试题或家庭作业。

最后，各章的末尾都给出了一个或多个项目。它们通常要花一小时或更长的时间来求解，适合于作为独立的家庭作业或实验室作业。

致谢

我想感谢这些人：我的妻子 Heather，她审校了上面的内容并远远超出了她应尽的责任，同时她还提醒我注意饮食；Dan Friedman，当我在研究所撰写本书的一个早期版本时，他使我相信我不是在闹着玩，而是在做一件严肃的工作；James Ernest（21 世纪的 Sid Sackson），

他允许我使用他的游戏设计；Alan Apt，他给我提供了很多指导，并且还在重要的时间给我提出了一针见血的意见；Paul Purdon 和 David Wise，他们照亮了数据结构和算法的一些黑暗的角落；同时我还要感谢多年来给我提供评论、建议和审校的所有人，包括 Kevin 和 Rebecca Djang、Jerry Franke、Matt Jadud、Sid Kitchel、Jim Levenick、印第安那大学和路易斯-克拉克学院的计算机科学系的无数学生，以及匿名评论人。

Peter Drake
俄勒冈州波特兰市

目

录

C
O
N
T
E
N
T
S

第 I 部分 面向对象程序设计

第 1 章 封装

1.1 软件开发	2	1.2.6 完成 Die 类	13
1.1.1 良好的程序	2	习题	14
1.1.2 封装	3	1.3 使用对象	14
1.1.3 软件开发周期	5	1.3.1 Beetle 类	14
习题	6	1.3.2 toString()方法	15
1.2 类和对象	7	1.3.3 BeetleGame 类	19
1.2.1 类	7	习题	24
1.2.2 对象、字段和方法	8	1.4 小结	24
1.2.3 构造函数	9	1.5 术语	25
1.2.4 访问器、修改器和 this	11	1.6 复习题	26
1.2.5 静态与非静态	12	1.7 项目	27

第 2 章 多态性

2.1 引用类型	29	2.2.3 示例: Domineering	37
2.1.1 null	30	习题	42
2.1.2 引用和相等性	30	2.3 接口	43
2.1.3 多态类型对象	31	习题	47
2.1.4 基本类型和包装器	33	2.4 重载	47
2.1.5 String	33	习题	48
习题	34	2.5 小结	48
2.2 数组	34	2.6 术语	49
2.2.1 声明、分配和初始化	34	2.7 复习题	50
2.2.2 多维数组	35	2.8 项目	50

第 3 章 继承

3.1 扩展类	53	3.1.1 多态性和继承	55
---------	----	--------------	----

3.1.2 继承链	57	3.3 包和访问级别	63
3.1.3 is-a 和 has-a	58	访问级别	64
习题	60	习题	65
3.2 Object 类	61	3.4 小结	65
3.2.1 Object 类的方法	61	3.5 术语	66
3.2.2 隐式构造函数	62	3.6 复习题	66
习题	62	3.7 项目	66

第II部分 线性结构

第4章 栈和队列

4.1 Stack 接口	70	习题	86
4.1.1 泛型	71	4.4 Queue 接口	86
4.1.2 示例: Idiot's Delight	72	习题	91
习题	77	4.5 小结	91
4.2 调用栈	78	4.6 术语	91
习题	80	4.7 复习题	92
4.3 异常	80	4.8 项目	93

第5章 基于数组的结构

5.1 收缩和加长数组	95	5.4 迭代器	111
5.1.1 Card 类	96	5.4.1 Iterator 接口	111
5.1.2 收缩数组	97	5.4.2 Iterable 接口	112
5.1.3 加长数组	100	5.4.3 ArrayIterator 类	112
习题	101	5.4.4 示例: Go Fish	114
5.2 实现栈和队列	101	习题	120
5.2.1 ArrayStack 类	101	5.5 初识 Java 集合框架	121
5.2.2 ArrayQueue 类	103	抽象类	121
习题	105	习题	122
5.3 List 接口	106	5.6 小结	123
5.3.1 接口	106	5.7 术语	123
5.3.2 ArrayList 类	107	5.8 复习题	123
习题	110	5.9 项目	124

第6章 链表结构

6.1 表节点	125	6.2.1 LinkedStack 类	128
习题	128	6.2.2 LinkedQueue 类	131
6.2 栈和队列	128	习题	133

6.3	LinkedList 类	134	习题	142
6.3.1	Predecessor 接口	136	6.5 小结	142
6.3.2	两指算法	138	6.6 术语	142
6.3.3	ListIterator 类	139	6.7 复习题	143
	习题	140	6.8 项目	143
6.4	再论 Java 集合框架	141		

第Ⅲ部分 算 法

第 7 章 算法分析

7.1	计时	146	习题	158
	习题	148	7.5 平摊分析	159
7.2	渐近表示法	148	习题	160
	习题	152	7.6 小结	160
7.3	统计步骤数	153	7.7 术语	161
	习题	157	7.8 复习题	161
7.4	最好、最坏和平均情况	157	7.9 项目	162

第 8 章 查找和排序

8.1	线性查找	163	8.4 Comparable 接口	170
	习题	164	习题	173
8.2	折半查找	164	8.5 排序链表	173
8.2.1	折半查找分析	165	习题	174
8.2.2	假定 n 是 2 的幂	166	8.6 小结	174
	习题	167	8.7 术语	175
8.3	插入排序	167	8.8 复习题	175
	习题	169	8.9 项目	176

第 9 章 递归

9.1	递归地思考	177	习题	193
	习题	183	9.5 避免递归	193
9.2	分析递归算法	183	9.5.1 尾部递归	194
	习题	186	9.5.2 动态规划	195
9.3	归并排序	186	习题	197
	归并排序分析	189	9.6 小结	197
	习题	189	9.7 术语	198
9.4	快速排序	190	9.8 复习题	198
	快速排序分析	192	9.9 项目	200

第IV部分 树和集合

第10章 树

10.1 二叉树	202	10.3.1 表示广义树	214
10.1.1 有关树的术语	203	10.3.2 示例: 智能的 Tic Tac Toe 玩家	215
10.1.2 实现二叉树	205	习题	221
习题	208	10.4 小结	221
10.2 树的遍历	210	10.5 术语	221
习题	213	10.6 复习题	223
10.3 广义树	213	10.7 项目	223

第11章 集合

11.1 Set 接口	224	11.4.2 散列函数和散列码	244
习题	229	11.4.3 冲突解决方法	245
11.2 有序表	230	11.4.4 查找	248
11.2.1 查找	232	11.4.5 插入	249
11.2.2 插入	233	11.4.6 删除	250
11.2.3 删除	233	习题	250
习题	234	11.5 再论 Java 集合框架	251
11.3 二叉查找树	234	映射	252
11.3.1 查找	235	习题	252
11.3.2 插入	236	11.6 小结	253
11.3.3 删除	238	11.7 术语	253
习题	241	11.8 复习题	254
11.4 散列表	242	11.9 项目	255
11.4.1 直接定址法	242		

第V部分 高级主题

第12章 高级线性结构

12.1 位向量	258	12.4.1 插值查找	271
BitSet	264	12.4.2 比较排序的下界	273
习题	264	12.4.3 桶排序	273
12.2 稀疏数组	265	习题	275
习题	267	12.5 小结	275
12.3 多维数组的连续表示法	267	12.6 术语	276
习题	271	12.7 复习题	276
12.4 高级查找和排序	271	12.8 项目	276

第 13 章 字符串

13.1 String 和 StringBuilder	277	习题	289
习题	280	13.3 小结	289
13.2 字符串匹配	280	13.4 术语	290
13.2.1 朴素的字符串匹配	282	13.5 复习题	290
13.2.2 RK 指纹识别算法	283	13.6 项目	291
13.2.3 KMP 跳跃算法	285		

第 14 章 高级主题

14.1 堆	292	14.4 红黑树	308
14.1.1 优先级队列	294	14.4.1 红黑树的性质	308
14.1.2 堆排序	296	14.4.2 查找	309
14.1.3 Java 的 PriorityQueue 类	297	14.4.3 插入	309
习题	298	14.4.4 删除	311
14.2 不相交集合簇	298	14.4.5 实现	312
14.2.1 按高度合并	300	习题	320
14.2.2 路径压缩	301	14.5 小结	320
习题	302	14.6 术语	321
14.3 数字查找树	302	14.7 复习题	321
习题	308	14.8 项目	321

第 15 章 图

15.1 术语	323	15.5.1 Dijkstra 的单源点算法	340
习题	327	15.5.2 Floyd-Warshall 所有顶点对算法	341
15.2 表示法	327	习题	342
习题	332	15.6 最小生成树	342
15.3 图的遍历	332	习题	346
习题	334	15.7 小结	346
15.4 拓扑排序	335	15.8 术语	346
习题	339	15.9 复习题	348
15.5 最短路径	339	15.10 项目	348

第 16 章 内存管理

16.1 显式内存管理	350	习题	358
16.1.1 自由表	352	16.2 自动内存管理	358
16.1.2 使用节点池	356	16.2.1 引用计数	358

16.2.2 标记和清理无用单元收集	359	16.4 术语	366
16.2.3 复制无用单元收集	359	16.5 复习题	367
习题	365	16.6 项目	367
16.3 小结	366		

第 17 章 输出到磁盘

17.1 与文件交互	368	17.4 B 树	389
17.1.1 文本文件	368	17.4.1 查找	390
17.1.2 数据文件	372	17.4.2 插入	390
习题	376	17.4.3 删除	391
17.2 压缩	376	17.4.4 实现	392
17.2.1 霍夫曼编码方式	376	习题	404
17.2.2 Lempel-Ziv 编码方式	381	17.5 小结	405
习题	384	17.6 术语	405
17.3 外部排序	384	17.7 复习题	406
习题	389	17.8 项目	406

第 VI 部分 附录

附录 A Java 知识回顾

A.1 第一个程序	408	A.6 方法和中断	417
A.2 变量和类型	410	A.7 常量	418
A.3 循环	412	A.8 运算符	420
A.4 与用户交互	414	A.9 调试	423
A.5 分支	414	A.10 编码约定	424

附录 B 统一建模语言

B.1 类图	428	B.2 实例图	431
--------	-----	---------	-----

附录 C 求和公式

C.1 求和符号	433	C.4 二等分与加倍之和	435
C.2 常量求和	434	C.5 函数之和的上限	435
C.3 前 n 个整数之和	434	C.6 常数因子	436

附录 D 进一步的阅读材料

D.1 数据结构和算法	437	D.3 游戏	437
D.2 Java	437		

第 I 部分 面向对象程序设计

封 装

本章介绍软件开发的面向对象方法。1.1 节讨论了软件开发过程和封装的思想：把一个程序分成多个具有有限交互的不同组件。在 1.2 节和 1.3 节，我们开发了一个程序，用于玩 Beetle（甲虫）游戏。在 1.2 节中，当我们编写类为游戏中使用的骰子建模时，介绍了与对象相关的术语和概念。1.3 节在此基础上进行了扩展，开发了另外两个类来完成程序。

有些读者可能希望在开始学习本章之前，先阅读附录 A。

1.1 软件开发

1.1.1 良好的程序

良好的计算机程序是什么特点？程序必须是**正确**（correct）的，按人们设想的那样执行工作，并且不包含任何错误。程序应该是**高效**（efficient）的，只使用必需的时间和内存。程序应该是**通用**（general-purpose）的，使得下次构建类似的程序时不必从头开始。最后，与所有其他的事物一样，程序应该是**快速开发**（rapidly developed）的。虽然有一些变化可以不付出任何代价地推进这些目标之一，但是通常需要进行一些折衷，如图 1-1 所建议的那样。

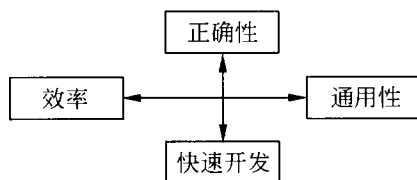


图 1-1 在良好的计算机程序的各个特点之间通常会有折衷

我们想让所有程序都完全正确。在某些应用（如医疗设备、飞机导航和核电站控制）中，人们的生命与软件的正确性息息相关。而在另外一些应用（如游戏和 Web 浏览器）中，偶尔的崩溃可能只会令人烦恼。确保正确性的最佳方式是精确指定人们设想程序将如何运行，然后彻底地测试程序，以验证它是否会按人们设想的那样工作。不幸的是，这可能要花极多的时间。实用的考虑通常会导致发布有错误的软件。