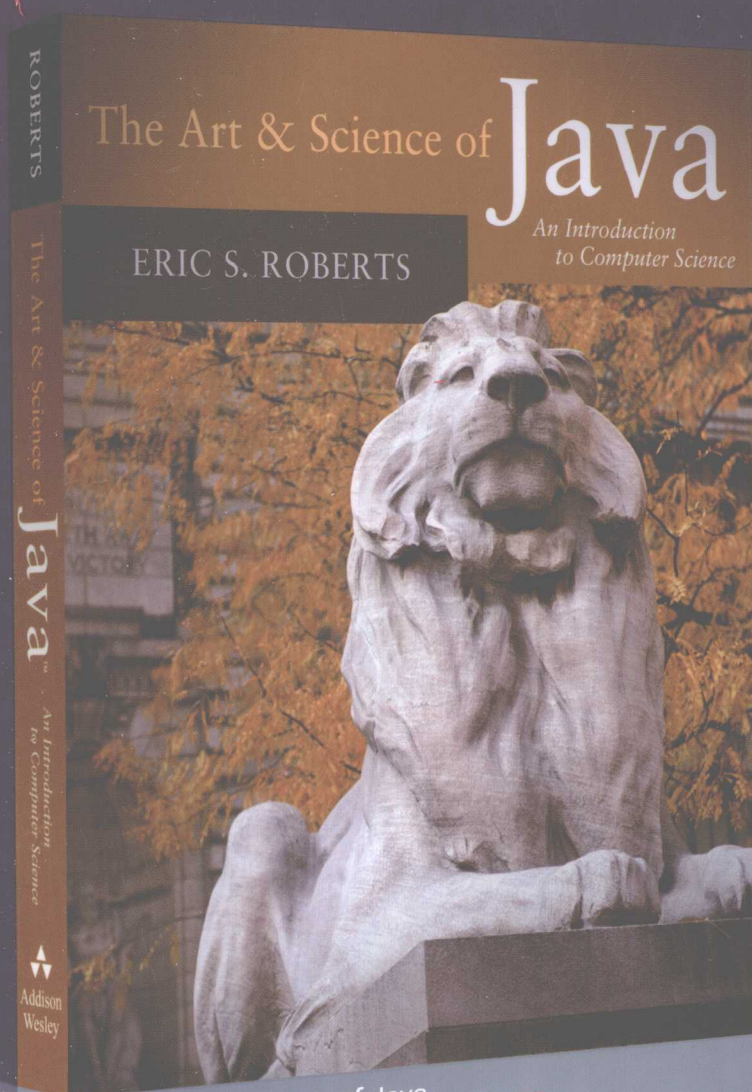


Java

语言的科学与艺术

(美) Eric S. Roberts 著 付勇 译
斯坦福大学



The Art & Science of Java



清华大学出版社

国外计算机科学经典教材

Java 语言的科学与艺术

(美) Eric S. Roberts 著

付勇 译

清华大学出版社

北 京

Authorized translation from the English language edition, entitled The Art & Science of Java, 978-0-321-48612-7 by Eric S.Roberts, published by Pearson Education, Inc, publishing as Addison-Wesley, Copyright © 2008.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and TSINGHUA UNIVERSITY PRESS Copyright © 2008.

北京市版权局著作权合同登记号 图字: 01-2007-2032

本书封面贴有 Pearson Education(培生教育出版集团)防伪标签, 无标签者不得销售。
版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Java 语言的科学与艺术/(美) 罗伯茨(Roberts, E.S.) 著; 付勇 译. —北京: 清华大学出版社, 2009.1
(国外计算机科学经典教材)

书名原文: The Art & Science of Java

ISBN 978-7-302-18441-6

I. J… II. ①罗…②付… III. JAVA 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字(2008)第 130802 号

责任编辑: 王 军 徐燕萍

装帧设计: 孔祥丰

责任校对: 成凤进

责任印制: 何 芊

出版发行: 清华大学出版社

<http://www.tup.com.cn>

社 总 机: 010-62770175

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

邮 购: 010-62786544

印 刷 者: 北京密云胶印厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 30 字 数: 768 千字

版 次: 2009 年 1 月第 1 版 印 次: 2009 年 1 月第 1 次印刷

印 数: 1~4000

定 价: 59.80 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 025079-01

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 and 改革,亟需一批门类齐全、具有国际先进水平的计算机经典教材,以适应我国当前计算机科学的教學需要。通过使用国外优秀的计算机科学经典教材,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,从而培养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,我们从国外多家知名的出版机构 Pearson、McGraw-Hill、John Wiley & Sons、Springer、Thomson 等精选、引进了这套“国外计算机科学经典教材”。

作为世界级的图书出版机构, Pearson、McGraw-Hill、John Wiley & Sons、Springer、Thomson 通过与世界级的计算机教育大师携手,每年都为全球的计算机高等教育奉献大量的优秀教材。清华大学出版社和这些世界知名的出版机构长期保持着紧密友好的合作关系,这次引进的“国外计算机科学经典教材”便全是出自上述这些出版机构。同时,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,从上述这些出版机构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,全部由对应专业的高校教师或拥有相关经验的 IT 专家担任。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

国外计算机科学经典教材

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
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

致学生

恭喜你！当你拿起这本书的时候，就已经步入了计算机科学的殿堂——这一研究领域半个世纪以前还是一片空白，但现在已经发展成为我们这个时代最具活力的学科之一。

现在这个时代，计算机在人类活动的各个领域都创造了非凡的可能。今天，企业领导者能够管理遍布全球的企业，因为计算机让他们能够在任何地方以极快的速度传递数据。科学家现在能够解决很多问题，而在没有计算机进行必要的计算之前，这些问题超出了他们的能力范围。WWW 提供了大量信息，它建立在一个充满活力的行业基础之上，而这个行业在十几年前根本不存在。电影摄制者可以使用计算机技术创建动画特征，而这些动画特征在迪斯尼时代简直不可想象。现代计算促进了许多领域发生变革：让生物学家能够排列人类基因，让经济学家能够模拟国际金融市场，让文学家能够确定非 Elizabethan 所写的原稿是否为 Shakespeare 所写。

计算是一种意义深远的重要技术。随着计算产业不断发展，它提供的就业岗位比 20 世纪 90 年代 Internet 快速发展时提供的岗位还要多。但相对于本世纪我们将要经历的变化而言，现在看到的这些发展微不足道，今天还是学生的你们不久就会担负起引领这种发展的重任。不管将来选择什么领域，理解如何有效使用计算机都非常重要。

就像许多值得学习的技能一样，理解计算机的工作原理、学习如何控制它强大的能力也需要时间。你们不可能一蹴而就，必须从某一点开始。2500 年前，中国哲学家老子曾经说过“千里之行，始于足下”，本书就是一个开端。

然而，对于大多数同学而言，第一步是最艰难的。许多同学感到计算机过于强大，认为计算机科学超出了自己的理解范围。但是，学习编程基础不需要高等数学知识或全面理解电子学。对编程而言，最重要的是能否从问题的陈述中找到它的解决方案。因此，你必须能够从逻辑上思考。必须遵循必要的原则，以计算机可以理解的形式表达自己的逻辑。也许最重要的是，必须能够洞穿任务、成功完成，而不会被困难和挫折所吓倒。如果不断前进，你们就会发现，找到解决方案是多么令人欢欣鼓舞，这足以补偿一路上遇到的所有挫折。

祝愿你们在这条路上一路走好！

Eric Roberts
斯坦福大学
2007 年 1 月

致老师

本书适合普通学院或大学的初级编程课使用。书中介绍了传统 CS1 课程的内容,正如计算机协会(ACM)准备的 *Curriculum '78* 报告中定义的一样。同时也包含了最近的 *Computing Curriculum 2001* 报告计算机科学卷中指定为 CS101₀ 或 CS111₀ 课程的全部主题。

本书使用的方法与我 1995 年所著的 *The Art and Science of C* 一书使用的方法相同。这两本书都使用库来简化编程,从而让它们更适合于初学者。在 C 语言版中,事实证明这些库对于学生而言(不管是在斯坦福大学,还是在其他许多机构)都非常成功,本书使用 ACM Java 库来实现相同的目标。

自 1995 年首次发布以来,Java 编程语言作为一种教育语言变得日益重要,以至于现在成了初级计算课程的标准语言。在好的方面,Java 相对于早期教育语言具有更多优点,特别是它可以让学⽣编写高度交互式程序,这充分激发了他们的兴趣和想象。但 Java 比其他传统教育语言(例如 BASIC 和 Pascal)更复杂,老师和学生们在理解 Java 语言的结构时,复杂性就成为了最大的障碍。

为了解决初级老师在使用 Java 时会遇到的问题,2004 年 ACM 建立了 Java 工作组(Java Task Force),并给出如下指示:

要从初级计算教育的角度讨论 Java 语言、API 和工具,要开发稳定的教育资源集合,以方便给一年级计算学生讲授 Java,让这些学生不至于被其复杂性所吓倒。

在接下来的两年时间内,Java 工作组开发了一组新库,这些库支持在初级层次使用 Java。在发行两个草案初稿进行了社会反馈后,2006 年夏天 Java 工作组发行了最终报告。在下面 Web 站点可以找到报告中描述的 ACM Java 库:

<http://jtf.acm.org/>

除了 ACM Java 库本身之外,该 Web 站点还包含了大量演示程序、教学软件指南、各种程序包设计基本原理的详细讨论,以及确定下面几点为库最大优势的执行概要。ACM Java 库提供:

- 简单的面向对象程序模型。acm.program 程序包里定义的 Program 类为编写简单程序提供了易于使用的模型。除了隐藏静态 main 方法之外,Program 类及其标准子类提供了高度直观的面向对象的类层次结构示例。

- 对称处理传统控制台 I/O 和对话 I/O 的输入输出模型。`acm.io` 程序包定义 `IOConsole` 和 `IODialog` 类，它们共享所有输入输出操作的公共界面。这种设计解决了 Java 缺少简单输入机制的问题，在某种程度上突出了基于界面设计的价值。
- 图形对象的扩充库。`acm.graphics` 程序包实现一种简单但非常强大的模型，这种模型创建基于毡板隐喻的图形图片。在该毡板上，同学们可以构造各种图形对象，并将它们放在画布上。这种设计强调对象的使用，将学生从必须明确响应重画请求中解脱出来。
- 支持图形用户界面开发的一些新类。`acm.gui` 程序包包含一些类，这些类让初级编程人员能够理解 Java 广泛的 GUI 开发资源。
- 后向兼容 `applet`。和今天大多数 Java 代码不一样，使用 ACM Java 库开发的程序通常作为 `applet` 执行，即使是在原来的 Web 浏览器上。这种灵活性让这些库成为了基于 Web 教育工具和讲稿演示的完美基础。

除了 Java 自己的库中可用的大量类和方法，这种书不可避免会省略 Java 的一些特殊特征。包含 Java 所有特征的书不可能适用于一般学生。依据所选择的主题，本书试图提供一种条理分明的介绍，既讲述计算机科学的基础，又不包含 Java 的所有特征。在许多方面，本书主题选择的指导原则来源于如下观点，这些观点是 Antoine de Saint-Exupéry 在他 1942 年的备忘录 *Pilote de Guerre* 中提出来的(后来 Tony Hoare 在他 1980 年获得图灵奖的演说中引用过)：

达到完美的时候，不是没有什么可以添加，而是没有什么可以舍弃。

本书当然不能说完美，但省略一些主题比加入更多主题可能更好。

虽然本书主题的排列顺序在斯坦福大学证明是成功的，但也可以改变介绍的顺序，以适应学生和课程目标。下面是对各章的概述，同时指出了一些重要依据。

第 1 章回顾了计算的历史，介绍了编程的过程。这一章没有介绍编程的本质，但为其他章节进行了铺垫。

我为没有或几乎没有计算背景的学生设计了第 2 章。这一章的表述方法是概念性的，着重让学生形成对面向对象编程的整体理解，而不是强调 Java 语言的细节。新学生面对语法和结构的详细规则时，会集中精力学习规则而不是基本概念，而在这一阶段，这些概念极为重要。如果学生对编程有所了解，就可以快速浏览或略过这一章。

第 3~5 章分别介绍了传统的表达式、语句和方法，以便学生们理解这些概念。

第 6 章和第 7 章继续介绍对象和类的基础知识。第 6 章从高层面上，着重介绍如何使用对象和类而不是重点介绍它们的基本结构。第 7 章转向低层面，详细介绍了对象在内存中的表示方法。虽然第 7 章没有特别要求，但可以理解对象内部结构的学生很可能理解引用的基本概念。

接下来的 3 章介绍了 Java 标准库和 ACM 程序包中的一些特定类。第 8 章讨论了 `String` 类，它可能是所有初级课程的重要主题。第 9 章详细介绍了 `acm.graphics` 类，它可以让学生写出更令人兴奋的程序。前面几章包括了部分 `acm.graphics` 程序包，所以第 9 章只讨论了某些主题。第 10 章简要介绍了事件驱动编程。第 10 章的前几节着重介绍了鼠标和键盘事件；剩下的几节介绍了图形用户界面(GUI)和 `javax.swing` 中的标准交互类。如果要想让学生设计基于 GUI 的程序，那么这一章的第二部分就很有价值，并且不必理解下面几章。

第 11 章和第 12 章从不同角度讨论了数组的概念。第 11 章介绍了内嵌数组类型和 `java.util` 程序包中的 `ArrayList` 实用程序类。在所有基于 Java 的介绍性课程中，这两个主题似乎都必不可少。第 12 章着重介绍了搜索和排序数组的算法。这一章也简要讨论了计算的复杂性，这将

有助于学生理解算法设计的重要性。

第 13 章介绍了 Java 集合架构，它在第二学期的编程课中也许更常见。然而，因为 Java 考虑到了这么多基本细节，所以教初学者如何使用这些类实际上非常合理，即使他们不理解其实现方式。Java 集合架构中唯一必不可少的类是 `ArrayList`，第 11 章已经提到过它。

第 14 章介绍了 4 个重要主题，这些主题有时会在初级编程课程中出现：递归、并发、网络和编程模式。斯坦福大学实行的是 4 学期制，因此在第二学期介绍这些主题。如果要在第一学期介绍递归，我强烈建议您尽早这样做，以便给学生提供更多时间来消化吸收这些材料。一种可能是在第 5 章之后开始讨论递归函数，在第 12 章的最后介绍递归算法。

补充资源

学生适用

本书的所有读者都可以在 Addison-Wesley 的 Web 站点(<http://www.aw.com/cssupport/>)获得下列几项资源：

- 书中所有示例程序的源代码文件。
- 彩色 PDF 版本的样本运行。
- 复习题的答案。

教师适用

认证教师可以在 Addison-Wesley 的教师资源中心(Instructor Resource Center，网址为 <http://www.aw.com/irc/>)获得下列几项资源：

- 书中所有示例程序的源代码文件。
- 彩色 PDF 版本的样本运行。
- 复习题的答案。
- 编程练习的解决方案。
- 基于 Applet 且包含程序示例动画的教学幻灯片。

ACM Java 库的用户适用

计算机协会(ACM)维护 Java 工作组开发了 ACM Java 库的 Web 站点(<http://jtf.acm.org/>)。该站点包括下列资源：

- ACM Java 库用法的执行概要。
- 可以以源代码及编译形式下载的 ACM 库的副本。
- 包含示例源代码的演示图库。
- 使用 ACM 库的初级指南。
- 设计基本原理的全面介绍。

目 录

第 1 章 前言	1	2.6.1 HelloProgram 示例回顾	29
1.1 计算简史	1	2.6.2 向 GObjects 发送消息	30
1.2 计算机科学的含义	3	2.6.3 GObject 类的层次结构	32
1.3 计算机硬件简介	4	2.6.4 GRect 类	33
1.3.1 CPU	5	2.6.5 GOval 类	36
1.3.2 内存	5	2.6.6 GLine 类	37
1.3.3 辅助存储器	5	2.7 小结	38
1.3.4 输入/输出(I/O)设备	5	2.8 复习题	39
1.3.5 网络	5	2.9 编程练习	40
1.4 算法	6	第 3 章 表达式	43
1.5 编程过程的几个阶段	6	3.1 原始数据类型	44
1.5.1 创建和编辑程序	7	3.2 常量与变量	45
1.5.2 编译过程	7	3.2.1 常量	45
1.5.3 编程错误与调试	9	3.2.2 变量	46
1.5.4 软件维护	9	3.2.3 声明	47
1.6 Java 和面向对象范例	10	3.2.4 命名常量	48
1.6.1 面向对象编程的历史	11	3.3 运算符和操作数	48
1.6.2 Java 编程语言	11	3.3.1 合并整数和浮点数	49
1.7 Java 和 WWW	14	3.3.2 整数除法和余数运算符	50
1.8 小结	15	3.3.3 优先级	50
1.9 复习题	16	3.3.4 应用优先级规则	52
第 2 章 编程示例	17	3.3.5 类型转换	53
2.1 “Hello world” 程序	18	3.4 赋值语句	55
2.1.1 注释	19	3.4.1 简写赋值运算符	57
2.1.2 输入	19	3.4.2 递增运算符和递减运算符	58
2.1.3 主类	19	3.5 布尔表达式	58
2.2 编程过程的观点	21	3.5.1 关系运算符	58
2.3 两数相加的程序	22	3.5.2 逻辑运算符	59
2.4 编程习语和模式	25	3.5.3 短路赋值	61
2.5 类和对象	26	3.5.4 标记	62
2.5.1 类的层次结构	27	3.5.5 Boolean 计算示例	62
2.5.2 Program 类的层次结构	28	3.6 设计改变	63
2.6 图形程序	29	3.6.1 可读性的重要性	63

3.6.2	使用命名常量支持程序维护	64	5.1.4	作为消息的方法调用	108
3.6.3	使用命名常量支持程序开发	64	5.2	编写自己的方法	109
3.7	小结	66	5.2.1	方法定义的格式	109
3.8	复习题	68	5.2.2	return 语句	110
3.9	编程练习	69	5.2.3	包含内部控制结构的方法	111
第 4 章	语句形式	73	5.2.4	返回非数字值的方法	112
4.1	Java 的语句类型	74	5.2.5	断言方法	115
4.1.1	简单语句	74	5.3	方法调用过程的技巧	116
4.1.2	复合语句	75	5.3.1	参数传递	117
4.1.3	控制语句	76	5.3.2	从其他方法内调用方法	120
4.2	控制语句和问题解决	76	5.4	分解	126
4.2.1	一般化 Add2Integers 程序	77	5.4.1	逐步细化	126
4.2.2	重复 N 次模式	78	5.4.2	指定参数	128
4.2.3	“读取到指定条件为止”模式	79	5.4.3	自顶向下设计	129
4.3	if 语句	80	5.4.4	寻找共同特征	130
4.3.1	单行 if 语句	82	5.4.5	完成分解	131
4.3.2	多行 if 语句	82	5.5	算法方法	132
4.3.3	if-else 语句	83	5.5.1	“强力”方法	132
4.3.4	级联 if 语句	83	5.5.2	欧几里得算法	133
4.3.5	?:运算符	84	5.5.3	讨论欧几里得算法的正确性	133
4.4	switch 语句	85	5.5.4	两种算法的效率比较	134
4.5	while 语句	88	5.6	小结	135
4.5.1	使用 while 循环	88	5.7	复习题	135
4.5.2	无限循环	90	5.8	编程练习	136
4.5.3	解决“循环到中途”问题	90	第 6 章	对象和类	141
4.6	for 语句	92	6.1	使用 RandomGenerator 类	142
4.6.1	for 和 while 之间的关系	94	6.1.1	伪随机数	142
4.6.2	在浮点数据中使用 for 语句	95	6.1.2	使用 RandomGenerator 类	143
4.6.3	嵌套 for 语句	96	6.1.3	随机数种子的作用	145
4.6.4	简单的图形动画	97	6.2	javadoc 文档系统	146
4.7	小结	99	6.3	定义自己的类	149
4.8	复习题	100	6.3.1	类定义的结构	149
4.9	编程练习	100	6.3.2	控制条目的可见性	150
第 5 章	方法	105	6.3.3	封装	150
5.1	方法概述	105	6.4	表示学生信息	150
5.1.1	作为降低复杂性机制的方法	106	6.4.1	声明实例变量	151
5.1.2	作为编程人员工具而不是 用户工具的方法	106	6.4.2	完成类定义	151
5.1.3	作为表达式的方法调用	107	6.4.3	编写 javadoc 注释	154
			6.4.4	写构造函数	154

6.4.5	getters and setters	155	8.2.3	字符常量	205
6.4.6	toString 方法	155	8.2.4	Unicode 表示法的重要属性	205
6.4.7	定义类中的命名常量	156	8.2.5	特殊字符	206
6.4.8	使用 Student 类	156	8.2.6	字符算法	207
6.5	有理数	157	8.2.7	Character 类中有用的方法	208
6.6	扩展现有类	161	8.2.8	包含字符的控制语句	209
6.6.1	创建类表示实心三角形	161	8.3	作为抽象概念的字符串	210
6.6.2	继承构造函数的规则	165	8.3.1	从整体和简化论的观点 考察字符串	210
6.6.3	继承方法的规则	165	8.3.2	抽象类型的概念	210
6.7	小结	167	8.4	使用 String 类中的方法	211
6.8	复习题	168	8.4.1	确定字符串长度	212
6.9	编程练习	169	8.4.2	从字符串中选择字符	212
第 7 章	对象和内存	175	8.4.3	串联	213
7.1	内存结构	176	8.4.4	摘录部分字符串	215
7.1.1	位、字节和字	176	8.4.5	比较两个字符串	215
7.1.2	二进制和十六进制表示法	176	8.4.6	在字符串内搜索	216
7.1.3	内存地址	178	8.4.7	大小写字母转换	217
7.2	将内存分配给变量	179	8.5	字符串处理案例研究	218
7.2.1	Rational 类的内存图	180	8.5.1	应用自顶向下设计	218
7.2.2	无用单元收集	184	8.5.2	实现 translateLine	219
7.3	原始类型与对象	185	8.5.3	考虑空格和标点符号	220
7.3.1	参数传递	185	8.5.4	StringTokenizer 类	222
7.3.2	包装类	187	8.5.5	完成实现	223
7.3.3	装箱和拆箱	189	8.6	小结	226
7.4	链接对象	190	8.7	复习题	227
7.4.1	链接结构里的消息传递: Gondor 灯塔	191	8.8	编程练习	228
7.4.2	链接结构的内部表示法	193	第 9 章	面向对象图形	237
7.5	小结	194	9.1	acm.graphics 模型	238
7.6	复习题	195	9.2	acm.graphics 程序包的结构	238
7.7	编程练习	195	9.2.1	GCanvas 类	239
第 8 章	字符串和字符	199	9.2.2	Color 类的更多细节	241
8.1	枚举的原则	200	9.2.3	GPoint 类、GDimension 类和 GRectangle 类	241
8.1.1	在计算机内部表示枚举类型	200	9.2.4	GMath 类	242
8.1.2	将枚举类型作为整数表示	201	9.2.5	GObject 类及其子类	243
8.1.3	定义新的枚举类型	202	9.3	使用形状类	246
8.2	字符	203	9.3.1	GLabel 类	246
8.2.1	char 数据类型	203			
8.2.2	ASCII 和 Unicode 编码系统	203			

9.3.2	GRect 类及其子类 (GRoundRect 和 G3DRect)	248	10.7.3	BorderLayout 布局管理器	305
9.3.3	GOval 类	249	10.7.4	FlowLayout 布局管理器	307
9.3.4	GLine 类	249	10.7.5	GridLayout 布局管理器	308
9.3.5	GArc 类	250	10.7.6	标准布局管理器的不足	309
9.3.6	GImage 类	253	10.8	使用 TableLayout 类	309
9.3.7	GPolygon 类	257	10.8.1	比较 GridLayout 与 TableLayout	310
9.4	创建复合对象	262	10.8.2	使用 TableLayout 创建 温度转换器	310
9.4.1	简单的 GCompound 示例	263	10.8.3	指定约束	312
9.4.2	GCompound 坐标系	265	10.8.4	使用 TableLayout 创建 简单的计算器	313
9.4.3	使用 GCompound 的对象分解	265	10.9	小结	319
9.4.4	嵌套 GCompound 对象	269	10.10	复习题	320
9.5	小结	270	10.11	编程练习	321
9.6	复习题	271	第 11 章	数组与 ArrayList 类	327
9.7	编程练习	272	11.1	数组简介	328
第 10 章	事件驱动程序	279	11.1.1	数组声明	328
10.1	Java 事件模型	280	11.1.2	数组选择	329
10.2	简单的事件驱动程序	281	11.1.3	简单数组的示例	330
10.3	响应鼠标事件	283	11.1.4	改变索引范围	331
10.3.1	MouseListener 和 MouseMotionListener 接口	283	11.1.5	对象的数组	332
10.3.2	重写侦听器方法	283	11.1.6	在图形程序中使用数组	332
10.3.3	画线程序	284	11.1.7	++ 和 - 运算符的区别	334
10.3.4	在画布上拖动对象	286	11.2	数组的内部表示法	335
10.4	响应键盘事件	288	11.3	数组作为参数传递	337
10.5	创建简单的 GUI	289	11.4	使用数组制作表格	341
10.6	Swing 交互器层次结构	291	11.5	数组初始化	343
10.6.1	JButton 类	292	11.6	多维数组	344
10.6.2	JToggleButton 类	293	11.6.1	将多维数组传递给方法	345
10.6.3	JCheckBox 类	293	11.6.2	初始化多维数组	346
10.6.4	JRadioButton 类和 ButtonGroup 类	295	11.7	图像处理	346
10.6.5	JSlider 类和 JLabel 类	296	11.7.1	图像的表达方式	346
10.6.6	JComboBox 类	297	11.7.2	使用 GImage 类操作图像	347
10.6.7	JTextField 类、IntField 类和 DoubleField 类	300	11.7.3	位操作	348
10.7	管理组件布局	304	11.7.4	使用位操作分解像素组件	350
10.7.1	Java 窗口层次结构	304	11.7.5	创建灰度图像	350
10.7.2	布局管理器	305	11.7.6	通过平均使图像变得平滑	351
			11.7.7	隐藏复杂性	353

11.8	ArrayList 类	354
11.9	小结	358
11.10	复习题	359
11.11	编程练习	360
第 12 章	搜索与排序	369
12.1	搜索	370
12.1.1	在整数数组中搜索	370
12.1.2	搜索表	370
12.1.3	折半搜索	373
12.1.4	搜索算法的相对效率	375
12.2	排序	376
12.2.1	给整数数组排序	376
12.2.2	选项排序算法	377
12.2.3	评估选项排序的效率	379
12.2.4	测量程序的运行时间	380
12.2.5	分析选项排序算法	381
12.2.6	基数排序算法	382
12.3	评估算法效率	384
12.3.1	big-O 表示法	385
12.3.2	big-O 的标准简化	385
12.3.3	选项排序计算的复杂性	385
12.3.4	根据代码结构预测计算的复杂性	386
12.3.5	最坏情况与一般情况的复杂性	387
12.4	使用数据文件	388
12.4.1	文件的概念	388
12.4.2	阅读 Java 中的文本文件	389
12.4.3	异常处理	391
12.4.4	倒转文件的程序	393
12.4.5	交互地选择文件	394
12.4.6	使用 Scanner 类	395
12.4.7	输出文件	397
12.5	小结	397
12.6	复习题	398
12.7	编程练习	400
第 13 章	数组与 ArrayList 类	405
13.1	ArrayList 类回顾	406
13.1.1	动态分配的能力	406

13.1.2	区分表示法和行为	408
13.2	HashMap 类	409
13.2.1	映射的简单示例	409
13.2.2	探讨可能的实现策略	410
13.2.3	实现 O(1)性能	411
13.2.4	散列法思想	412
13.2.5	散列码	412
13.2.6	复制散列码	413
13.2.7	映射类的简单实现	413
13.2.8	实现 hashCode 方法	416
13.2.9	调整存储区的数量	418
13.3	Java 集合架构	418
13.3.1	Java 集合架构的结构	419
13.3.2	Collection 层次结构	420
13.3.3	区分行为和表示法	421
13.3.4	迭代器	422
13.3.5	Arrays 和 Collections 方法库	424
13.4	面向对象设计的原则	426
13.4.1	统一主题的重要性	427
13.4.2	简单和信息隐藏的原则	428
13.4.3	满足客户的需要	428
13.4.4	灵活性的好处	429
13.4.5	稳定的意义	429
13.5	小结	430
13.6	复习题	430
13.7	编程练习	431
第 14 章	展望	435
14.1	递归	436
14.1.1	递归的简单示例	436
14.1.2	Factorial 函数	437
14.1.3	信任的递归式跳跃	440
14.1.4	递归的范例	441
14.1.5	图形递归	442
14.1.6	递归式思考	444
14.2	并发	445
14.2.1	进程与线程	445
14.2.2	并发的简单示例	445
14.3	使用网络	448

14.4	编程模式.....	448
14.4.1	模型/视图/控制器模式.....	449
14.4.2	说明性示例：用图表示电子 数据表的数据.....	449

14.5	小结.....	453
14.6	复习题.....	454
14.7	编程练习.....	454

第 1 章

前 言

*[The Analytical Engine offers] a new, a vast,
and a powerful language. . . for the purposes of
mankind.*

—Augusta Ada Byron, Lady Lovelace, 1843



Augusta Ada Byron, Lady Lovelace(1815—1852)

Augusta Ada Byron 是英国著名诗人拜伦(Lord Byron)的女儿。当年有人鼓励她从事自己感兴趣的科学和数学专业，尽管那个时候几乎不允许妇女学习这些科目。17 岁时，Ada 遇到了 Charles Babbage。Charles Babbage 是一位杰出的英国科学家，他一生致力于设计实现数学计算功能的机器——尽管他未能完成这种机器的结构。但 Ada 坚信 Babbage 分析机的潜力，她记录了大量有关其设计的笔记，并编写了一些复杂的数学程序。这些程序使许多人把她当作第一位编程人员。为了纪念她，1980 年美国国防部将该编程语言命名为 Ada。

想想我们在 21 世纪初具有的优势，很难相信 1940 年竟然还没有计算机。而今天它们到处都是，用流行语来说(至少标题作者这么说)：我们生活在计算机时代。

1.1 计算简史

从某种意义上说，计算从古代就有了。许多早期数学都是为了解决现实中重要的计算问题，如监控牧群数量，计算小块土地面积，以及记录商业交易等。这些活动要求人们开发新的计算技术，有些情况下，还需要发明机器帮助计算，如算盘。算盘是一种简单的计算设备，由能够

在杆上滑动的珠子组成，已经在亚洲使用了几千年，很可能在公元前 2000 年就有了。

纵观其整个历史，计算的发展相对缓慢。1623 年，德国科学家 Wilhelm Schickard 发明了第一台为人所知的机械计算机，它能够自动执行简单的算术计算。虽然 Schickard 的机器在“三十年战争(1618—1648)”的破坏中消失于历史，但法国哲学家 Blaise Pascal 在 17 世纪 40 年代使用相同的技术构造了一台机械加法器，其仿制品现在陈列于法国巴黎国立工艺学院(Conservatoire des Arts et Métiers)。1673 年，德国数学家 Gottfried Leibniz 开发了一种相当精密的设备，它能够进行加减乘除运算。所有这些设备都完全是机械的，没有引擎，也没有其他能量来源。操作员通过将金属轮设置到特殊位置来输入数字，运行过程中，转动这些轮子，可以设置机器的其他部分，改变输出显示。

工业革命时期，技术的快速发展使开发新的机械计算方法成为可能。蒸汽机已经提供了工厂和铁路所需的能量。是否有人能够用蒸汽机来驱动更加精密的计算机(这些机器能够利用自身动力实现重要计算)呢？在那种情况下，问这种问题合情合理。然而，在取得进展之前，人们不得不提出这一问题，并开始寻找答案。必然的思想火花来自于英国数学家 Charles Babbage。在计算历史中，他是最值得关注的人之一。

Babbage 一生设计了两台不同的计算机，分别命名为差分机和分析机，每台机器都对当时的计算机进行了极大改进。他一生的遗憾是这两个项目都未能完成。他设计用来产生数学函数表的差分机，最终由瑞典发明家在 1854 年——提出最初设计 30 年后制造成功。分析机是 Babbage 一生的梦想，直到 1871 年他去世时仍未完成。即便如此，其设计仍然包含了现代计算机中的许多重要特征。更重要的是，Babbage 将分析机构想为一种多用途机器，依靠编程能够执行许多不同功能。在 Babbage 的设计中，分析机的运行由一段小孔控制，这些小孔被打在一张机器能够阅读的卡片上。改变小孔的样式，就可以改变机器的行为，执行不同类型的计算。

Augusta Ada Byron 是诗人拜伦(Lord Byron)与妻子 Annabella 的女儿。我们知道，Babbage 的许多工作都来源于她的著作。和大多数同时代的人一样，Ada 意识到了分析机的潜力，并成为了它的拥护者。她为该机器设计了一些复杂程序，也因此成为了第一位编程人员。20 世纪 70 年代，美国国防部为了纪念她在这方面做出的贡献，将自己的编程语言命名为 Ada。

Babbage 设计的有些方面确实影响了后来的计算历史，例如使用打孔卡来控制进程。法国发明家 Joseph Marie Jacquard 首先将这种思想作为织布机设备(后来被称为 Jacquard 织布机)的一部分，使织布过程自动化。1890 年，Herman Hollerith 使用打孔卡片使美国人口普查数据表格统计实现了自动化。为了让这种技术市场化，Hollerith 建立了一个公司，这就是后来的 IBM(国际商用机器)公司，它在 20 世纪大部分时间占据着计算机产业的主导地位。

Babbage 版的可编程计算机直到 20 世纪 40 年代才变为现实，电子学的出现使超越机械设备成为可能，之前都是机械设备在计算领域占主导地位。1939 年末，爱荷华州州立学院的 John Atanasoff 和他的学生 Clifford Barry 组装了第一台电子计算机原型。1942 年 5 月，他们完成了包含 300 个电子管的具有全面执行能力的机器。这台计算机能够解决线性方程这类小的系统问题。Atanasoff-Barry 计算机在设计方面做了一些调整，它可以执行更复杂的计算，但是这一项目的工作被第二次世界大战中断了。

第一台大规模电子计算机是电子数字积分计算机(Electronic Numerical Integrator and Computer, 简称 ENIAC)。1946 年，在宾夕法尼亚大学摩尔学校 J.Presper Eckert 和 John Mauchly 指导下完成的 ENIAC，包含了 18 000 多个电子管，占满了一个 30ft 长、50ft 宽的房间。ENIAC 通过将电线插入被称为配线架(patch pannel)的小钉板状设备里进行编程。通过将配线架上不同