

国外著名高等院校
信息科学与技术优秀教材



Java 软件开发

(第三版)

Developing **JAVA** software
Third Edition

〔英〕 Russel Winder 著
Graham Roberts
黄湘情 曹永存 译



 人民邮电出版社
POSTS & TELECOM PRESS

国外著名高等院校
信息科学与技术优秀教材

Java 软件开发

(第三版)

〔英〕 Russel Winder 著
Graham Roberts 著
黄湘情 曹永存 译



人民邮电出版社
北京

图书在版编目 (CIP) 数据

Java 软件开发: 第 3 版/ (英) 温德尔 (Winder, R.),
(英) 罗伯茨 (Roberts, G.) 著; 黄湘情, 曹永存译.
北京: 人民邮电出版社, 2008.7

国外著名高等院校信息科学与技术优秀教材
ISBN 978-7-115-17462-8

I. J… II. ①温…②罗…③黄…④…曹 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 (2008) 第 000474 号

版 权 声 明

Russel Winder Graham Roberts

Developing Java Software, Third Edition

Copyright©2006 by Wiley Publishing, Inc., Indianapolis, Indiana

All right reserved. This translation published under license.

Authorized translation from the English language edition published by Wiley Publishing, Inc..

本书中文简体字版由 **Wiley Publishing** 公司授权人民邮电出版社出版, 专有版权属于人民邮电出版社。

国外著名高等院校信息科学与技术优秀教材

Java 软件开发 (第三版)

◆ 著 [英] Russel Winder Graham Roberts
译 黄湘情 曹永存
责任编辑 李 际

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市潮河印业有限公司印刷
新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16
印张: 46
字数: 1 428 千字 2008 年 7 月第 1 版
印数: 1—4 000 册 2008 年 7 月河北第 1 次印刷

著作权合同登记号 图字: 01-2005-6482 号

ISBN 978-7-115-17462-8/TP

定价: 79.00 元

读者服务热线: (010)67132705 印装质量热线: (010)67129223

反盗版热线: (010)67171154

内容提要

本书是专门为 Java 软件开发课程编写的教材，旨在帮助读者学习 Java 程序设计和软件开发。

全书分为 5 个部分。第 1 部分（第 1~10 章）阐述了面向对象的 Java 语言和面向对象开发过程的基本概念和主要思想；第 2 部分（第 11~14 章）是有关程序设计的过程，介绍测试驱动开发的原理、过程、工具和技术，并说明是如何进行编程的；第 3 部分（第 15~17 章）给出了一些可以用于自学的案例，包括一些用来组织讨论会、扩展程序开发的项目基础内容。第 4 部分（第 18~24 章）是 Java 语言的自学指南和参考。附录部分提供了有关 Java 编程的一些有用参考资料。

本书适合作为高等院校计算机专业 Java 语言课程教材，也可以供 Java 语言自学者阅读使用。

前言

如今, Java 已无可非议地成为了出色的程序设计平台——它不仅是一门编程语言, 而且具有丰富的类库和工具。Java 被广泛地应用于企业计算系统、桌面应用程序, 而且在嵌入式系统中的运用也日益成功。在大学和其他一些高等教育机构中, Java 正逐渐成为一门流行的程序设计教学语言。通过在教学中把 Java 作为第一语言, 我们了解到该语言具有良好的性能, 而且可以激发学生的学习兴趣。举一个例子可以很好地说明这一点, 那就是方便性——编程新手们在编写程序时, 会对图形和图形用户界面的结合感到十分有趣和直观。

我们是伦敦大学国王学院 (KCL, King's College London) 的 RW (Russel Winder) 和伦敦学院大学 (UCL, University College London) 的 GR (Graham Roberts), 早在大学和研究生课程中就学习了 Java。那时我们就认识到 Java 将是下一代编程教学语言。当时教学材料很少, 我们必须自己编写。编写的材料是本书的第一版。看过的人都给出了很高的评价, 而且大学和学院都使用本书, 也有许多人购买, 这使我们大受鼓舞。根据我们在程序设计教学过程中使用本书的经验和其他使用本书的教学者反馈的意见, 我们给出了本书的第二版。修改了第一版中的错误, 更重要的是, 我们积累了丰富的经验, 根据这些经验, 改变了给出材料的方式。第二版获得了极大的好评, 仍然是流行的教学素材。我们感到非常欣慰, 但没有沾沾自喜。事情是不断变化的, 需要不断完善。因此我们编写了第三版, 其中包括了目前和以后几年里程序设计中的核心内容。

我们认为, 程序设计学习过程中的一个最大变化是单元测试和测试驱动开发, 这在以前的编程中就很重要, 如今则成为绝对中心和关键的环节, 是学习编程的重点和核心。过去编程, 只要程序能正常工作, 就认为是创建了正确的程序。如今, 这被认为是优秀程序员在理想情况下开发程序的方式。在现实世界中, 编程是创建一个工作系统, 将一个小的工作程序发展成较大的工作程序, 同时要管理程序中的错误。这种观点和想法与方法编程大不相同。现代观点是程序中总会有错误, 现代方法是在发展程序, 使它完成比以前更多功能的同时, 利用所有可用的工具查找和删除程序中的错误。因此学习程序设计的人要掌握面向问题、测试驱动的方法。根据我们的经验, 它不仅是真实世界中程序设计的方法, 也是帮助学生学习的优秀方法。

当然, 本书第二版和第三版最大的技术变化是 Sun 发行的 Java 5.0。就像从 Java 1.0 变到 Java 1.1 一样, 从 Java 1.4 转变到 Java 5.0 在语言上发生了巨大变化。Sun 认识到了这种巨大差异, 因此将新版本命名为 Java 5.0, 而不是 Java 1.5。Java 5.0 介绍的 Java 语言中, 最大的两个变化是引入了泛型 (Generic) 和注释 (Annotation)——还有其他重要的变化, 但这两个最重要。泛型和注释的引入意义重大, 它们改变了使用 Java 的方式。仍然使用 Java 2 平台, 所以有许多相同的地方, 虽然还是 Java, 但变化显著。我们不将泛型和注释当作 Java 的增加部分, 而是将 Java 5.0 作为编程语言。只要是适合学

习的 Java 内容, 我们都会用到。本书不可能覆盖 Java 的全部内容, 当然, 要学的东西还有很多, 还有更复杂的 Java 使用方式, Java 还有许多新特性可以利用, 但人们要选择从某处开始学习。这里假设您是从零开始。

还有一个变化, 见过本书第二版的人会发现, 在该版本中, 我们没有介绍算法和数据结构(ADS)包。在最初编写本书时, 我们将它贯穿于程序设计的前两个过程, 即从针对初学者的程序设计介绍一直到创建较大的库和应用程序。这导致书的篇幅过长。第二版的反馈意见都要求将书一分为二。因此针对这个合理的要求, 我们进行了改动。本书为初学者讲解程序设计的第一个过程, 涵盖元素、进程和编程工具, 也就是针对新接触程序设计的人, 介绍使用 Java 的测试驱动开发。ADS 是另一本书的内容, 该书讨论算法、数据结构、包、术语和模式。

学习编程

任何一个有意从事编程的人, 无论是专业人士还是业余爱好者, 都需要去了解如何使用 Java。这本书就是介绍如何使用 Java 编程的教材, 但是又不仅仅局限于 Java 程序设计语言。市面上有许多描述 Java 特性的书, 我们已经把其中一些曾参考过并深受大家喜爱的书列在了本书的参考文献中(“附录 J”)。我们并不打算和这些书竞争, 而只是想完善它们, 所以尽量做到涉及 Java 语言的所有细节。本书内容覆盖了 Java 语言的方方面面, 这将有助于学习如何使用 Java 编程。

Java 中最关键的问题就是面向对象(object-oriented), 这说明了对象在 Java 中最为重要。这也意味着一个 Java 程序员在他的设计生涯中应该尽早地了解对象。许多人认为在真正处理对象之前必须先学习构建对象的体系结构和技术。现实中存在着诸如以下的观点: “在你创建应用程序之前, 必须先学会构建对象以及构建对象所需的一切基本技术, 因此我们从最基础的部分开始介绍这种语言的所有特性, 然后你就可以进行程序设计了。”这一论点本质上还是“传统的”自底向上(bottom-up)的学习方法, 这种方法通常都是将一个主题详细地解释完才会进行下一个主题。我们认为这不是学习程序设计技巧的正确方法。相反, 您需要用一种整体思维方法, 用这种方法螺旋式上升地研究所有的主题, 每一次都会使您学到更多的知识。本书的教学就贯穿了这种学习方法。

Java 语言的主要的驱动力之一是“复用(reuse)”: 程序员在构建应用程序时, 应该尽量去重新利用以前已有的资源。这是一种很好的思想, 也是我们推荐读者在学习和使用语言时应采取的一种思路。分析如何用各部分的预先构造(preconstructed)软件来构建应用程序, 并且把对这些预先构造软件的解构(deconstructing)当作学习过程的一部分, 这才是实现重用的最佳方法。

我们不赞同用传统的程序设计语言进行教学, 而认为应该早些引入对象的概念, 其中有一些别的原因。世界上没有一种适用于所有人的学习途径, 标准的“自底向上”的学习程序设计的方式在这方面是尤其独断的。如上所述, 学习程序设计是一个更为全面的活动。人们吸取各家所长, 然后自己融会贯通。学习任何一门应用技能, 都是一项需要自己学习使用的工作, 程序设计当然也不例外。更何况它是一项需要人们推理的活动, 迄今为止还没有人可以确切地解释人们究竟如何完成这项工作。在这个领域, 存在一些假设和模型, 但却没有充分证明这些定理。

在程序设计的发展过程中, “自底向上”的方法已经延续了很多年了。许多老师这样教是因为他们自己也是这样学的。或许还存在其他起作用的隐含因素。有经验的专业人士和老师们常常懂得很多语言, 他们学习一种具有相同模式的新语言时, 通常会把这种新语言和他们已经熟悉的语言相比较, 并与以前编程的方法联系起来。对于有经验的人来说这也许是一种非常高效的技巧——因为他们已经掌握了大量的关于程序设计的语言、知识和技巧, 因而学习一种新的语言只是扩展了他们已有的知识。那么当老师认为那些新手们也应当使用同样的方法学习程序设计时(当然事实并非如此), 问题就产生了。

程序设计的模式问题(例如命令的(imperative), 过程的(procedural), 宣告的(declarative)和功能的(functional)等)也影响着程序设计的学习。当模式发生了改变, 例如从“面向过程”转向“面向对象”, 许多经验丰富的程序员认为它很难, 仅仅是因为他们不得不换一种新的思考问题的方式。老师经

常会认为学生在某方面会有困难,那是因为他们自己也曾遇到过,我们有理由相信这是不尽正确的观点。

因此我们的立场就是,对于编程新手,如果没有人告诉他们“面向对象”很难,没有人这样教他们,那么他们根本就不用担心。我们认为根据这种观点得出的方法是成功的。我们的经验是,有些学生在学习程序设计时是有些困难,但他们与您所想的不同,可以形成面向对象方法的概念。

在本书中,我们提出了最佳的“线性”阅读方式。但是,读者可以根据自己的需要自主地决定阅读哪些章节,并进行前后参照。我们并不保证所有 Java 语言的特性在使用前都被解释过。事实上,我们也清楚这确实是不可能的。然而,这种“使用先于解释”的方式符合我们后续的全面学习模式,每个人经过前后参照和索引都可以找到自己需要的知识点。

本书一开始就提出了最基本的问题“程序设计是什么?”,为“对象”的理解提供了基础的概念。接下来,首先介绍类和对象,然后是一致性和继承,最后是多线程。在开始的几章里提供的程序设计练习题,其目的是鼓励学生去编写一些小程序实践一下。即使学生对细节并不完全清楚,也能明白这些语法的用途。我们把这种学习程序设计的方式叫做“middle-out”方法。早点开始学习对象是很重要的。因此,语法和控制语句是最容易学习的部分,是向下继续学习的主线。而向上的主线则是结合类和对象来构造应用程序。

为了符合这一方法,本书被分成 5 个部分。第 1 部分,正如以上谈到的,给出了一些关于程序设计的概念性的材料,特别是使用对象和类编程,并提供了许多程序的例子。第 2 部分介绍编程活动,重点是测试驱动开发原理。第 3 部分是案例(case)学习,看看实际的应用程序是如何实现的。在这个部分里,我们描述了两个较大的程序的设计和实现过程,说明了编写“大程序”的过程中发生的各种活动和思考过程。第 4 部分提供了简短的 Java 语言参考,系统阐述了 Java 语言,提供了许多更深入一些的例题。附录部分分别列出了一些信息,但实际上这些信息并不一定普遍适用。

除以上 5 大部分以外,本书还有一个部分就是在网站上的支持材料:

<http://www.devjavasoft.org/>

我们在这个网站上提供了许多补充材料:讲座的幻灯片、额外的笔记、书中练习部分的习题和答案,这些材料是为那些第一次学习程序设计课程的学生准备的。应 Wiley 的要求,我们将继续使用这个站点并不断补充新的材料和案例研究,使得本书能够不断得以充实和完善。

如何使用本书

本书可以作为使用 Java 语言进行面向对象程序设计学习和教学的参考,同样也可以作为自学者的教材。基于 10 年来用 C++ 和 Java 进行面向对象分析和设计的教学经验,我们经过考虑把本书划分为以上清晰的 5 个部分。

第 1 部分提供初始的编程技术,第 2 部分是有关程序设计的活动和过程。这两部分是基本的课程单元。本书其他部分支持这两部分。因此初始课程是使用第 1 部分(受第 4 部分支持)介绍什么是程序设计,以及如何利用 Java 的特性来编写程序。接下来的课程是利用第 2 部分(受第 3 部分和第 4 部分支持)介绍如何进行编程。当然,本书在整个阶段或程序学习中都是有用的。不会是那种当读者学会了基本技术就可以抛弃的书籍。

在课程中我们极力强调学生应当完成大量的编程工作。为了达到这个目的,我们在每一章的结尾都提供了一些练习题。学生在读完一本书或参加一些讲座和讨论会之前,必须要花费足够长的时间来编写程序 and 解决程序中的问题。如果可能,最理想的办法是在可以得到一些专家帮助的实验室里完成这些工作。我们鼓励初学者亲自动手去实验,实践 Java 的所有特性,而且在此过程中可能会犯许多错误——这是一个优秀的程序员在成长过程中不可避免的,在实践过程中他们可以认识这些错误并加以改正,更重要的是从中学会了如何避免这些错误的再次发生。

每章最后的问题都分为 3 类。第 1 类是自测题,读者可以直接回答,检查自己对这一章内容的领会程度,而且提出了一些有利于讨论的问题和正式的作业。第 2 类是程序设计问题,需要进行一些思考,

并且要编写一些程序，这部分可以作为实验课的基本练习题，本书的网站上有详细的阐述。第3类是极具挑战的题目，读者不仅要处理一些更复杂的编程问题，而且必须做一些辅助研究才能理解和解决这些问题。这样的挑战题非常重要，它提高了读者的编程能力，而且问题所涵盖的内容有可能超出了当前章节的范围。

第1部分阐述了面向对象的Java语言程序设计和面向对象开发过程的关键概念和主要思想。与读者充分交流，帮助初学者有效地理解这种思想，以免曲解这些概念而形成不好的编程习惯。这一部分不拘泥于对Java的简单描述，超出常规的严格界限，内容组织合理。不要忘记，虽然Java最终仅是一门程序设计语言，但其中的概念和思想精华将会被广泛地应用。

第2部分介绍测试驱动开发的原理、过程、工具和技术。是关于“如何进行编程”的。测试驱动开发的原理值得很好地讨论和研究。但只有通过练习实践才能理解这个过程。不管测试驱动开发被说得有多好，只有在使用后才能体会到它的有效性。

第3部分给出了一些可以用于自学的案例研究，包括一些用来组织讨论会、扩展程序开发的项目基础内容。这一部分也强调尽可能多地进行编程实践和程序设计工作。

第4部分是Java语言的自学指南和参考。我们在这些章节中提供了大量的练习，不过更注重研究Java语言的隐含特性，而不是编程学习。学习这一部分也需要做大量的编程练习，当然最理想的办法是参加有组织的实验课。学习一种程序设计语言就是要用它编写程序并获得专家的认可，没有比这更好的方法。因此，我们不主张花大量的课堂时间来把Java语言的每个细节都教给学生。

致谢

尽管实际上是由我们(RW和GR)编写文本和程序，但得到了其他许多人的帮助。作者常常会说“感谢所有的人”这样的话，可能是因为要提到的人有很多，或者不能一一列举那些知道姓名的人，也可能是一种作者避免提到名人的技巧。我们打算这样做，我们想明确感谢某些人的贡献，即使这些致谢有点空洞。

不过在致谢安排上有点麻烦：我们是只赞扬那些帮助编写第三版的人，还是列出之前提到的所有人。但第三版是根据第二版编写的，第二版基于第一版，所以对第一版和第二版做出贡献的人也间接为第三版作了贡献。我们决定在本节中只感谢对编写第三版有功的人（编写的内容是第2版中没有的），然后一字不差地给出第二版中的致谢（其中只更正了某些印刷错误，不过这样做不符合语法）。

下面列出要感谢的人。

Cédic Beust 和 Alexandru Popescu 编写了 TestNG，并在继续开发（和其他人一起）——请参见 <http://www.testing.org/>。他们总是能随时解答我们的疑问，其中有许多都很重要。我们相信 TestNG 会很自然地成为 Java 实际的标准测试框架。

Timothy Wall 编写了 Abbot 包，并在继续开发（和其他人一起）——请参见 <http://abbot.sourceforge.net>。在我们使用 Abbot 遇到问题时，他都能够耐心解答，给予我们帮助。事实上，他给我们准备了一个特别的 Abbot 预发行版，其中混合了各种功能，我们可以运行所有应该起作用的测试，而在发行本书之前这些测试都不工作。Abbot 的功能非常强大。

Dan Alderman、John Artley、Ross Burton、Vincent Chong、Geoff Hunter、Helen Mead、Fang Lim、Alvin Tan、Hua Zhang 和 Peter Dzwig 与 RW 一起创建了针对嵌入式系统和智能卡的 Java 虚拟机的实现。在开发该项目期间，RW 学习了许多有关 Java 2 平台和 Java 程序设计的知识。遗憾的是，该项目不能再继续——不是技术失败，而是因为资金问题而终止了。

从 1998 年开始，UCL 的所有计算机科学系的大学生（和一些硕士生）就开始利用本书的前两版学习使用 Java 进行程序设计。新版本中引入的许多思想和例子都经过了试验和修改，特别是有关单元测试的内容。John Washbrook 教授一直在提供有用的建议，提出各种值得思考的问题，并说明如何编写试卷和给它评分。

Wiley 的 Jonathan Shipley 和 David Barnard 帮助我们编排本书。我们提出了许多问题，他们总是能给予我们最大的帮助。耐心真的是一种美德。

Amanda Makepeace 受 RW 的委托完成了封面的绘制。画的标题是“Wic 的森林”，是一副 406×306 的油画，完成于 2005 年 7 月。希望读者也能够喜欢。有关 Amanda 的更多作品，请访问她的网站 <http://www.amandamakepeace.co.uk/>。

Geri Winder 又是一个极具耐心的人，给了我们许多的支持，有时甚至帮我们做一些杂活。

第二版的致谢

作者在完成本书写作时，许多朋友给予了大力的支持和帮助，我们在此表示感谢。

Peter Wood、Alex Poulouvassilis、Nigel Chapman、Al Sutton、Nahed Sotcks、Malcolm Bird 和 Richard Overill 是 RW (RUSSEL WINDER) 以前的或现在的同事，他们为第一版、第二版投入了很多工作。Vasa Curcin、Jitesh Vassa、Dan Alderman 和 Ross Burton 是 KCL 的学生，他们在第 1 版到第 2 版的改进和资料整理过程中给予了很多的帮助。实际上，两年来在 KCL 大学 RW 教的所有学生都为本书作了贡献（例如提供了一些很好的素材），但由于人数很多（大约 600 人）所以无法一一感谢。Joshua Bloch 是集合框架（Collection Framework）的作者，他与 RW 针对 ADS 包的问题通过电子邮件进行了交流；Doug Lea 对于这个问题也给予了很多支持。最重要的是，Joe Bergin 对于 ADS 包中比较糟糕的部分给予了有建设性的意见。

GR (GRAHAM ROBERTS) 向所有 UCL 大学支持 Java 教学的同事们表示感谢，同时也对倡导使用 Java 完成课程作业或者项目的学生们表示感谢，他们在学习过程中的反馈意见是十分有价值的。还要感谢那些使用本书来学习 Java 的学生们。

来自 John Wiley & Sons 的 Gaynor Redvers-Mutton 从一开始就给予了鼓励（再一次），在本书编写的过程中她多次致电和约见我们，商榷有关事宜。Dawn Booth 也来自 John Wiley & Sons，在编排打印的过程中给予了大力帮助。如果没有他们的支持和努力，我们不可能在如此短的时间里完成这样一本大部头的书籍。还要感谢 Mark Spedding 对本书版式的设计，希望读者也能够喜欢。

很大程度上，与我们一起生活的人们给予的支持促使了本书的完成。RW 的妻子 Geri Winder 甚至对于程序中的 bug 都不放过，更不用说她对本书的内容倾注的心血了。GR 也同样得到了家人极大的支持。

Russel Winder

It'z Interactive Ltd

41 Buckmaster Road, London SW11 1EN, UK

russel@itzinteractive.com

Graham Roberts

UCL (University College London) 计算机科学系

Gower Street, London WC1E 6BT, UK

g.roberts@cs.ucl.ac.uk

目 录

第1部分 对象和类的编程

第1章 简介.....3

- 1.1 引言.....3
- 1.2 Java 简史.....3
- 1.3 Java 生逢其时.....4
- 1.4 Java 是什么.....5
- 1.5 抽象：编程的重要核心.....5
- 1.6 Java 2 平台.....6
- 1.7 Java 的体系结构是平台无关的.....7
- 1.8 Java 不是玩笑.....8
- 1.9 小结.....8

第2章 程序设计基础.....9

- 2.1 引言.....9
- 2.2 抽象和概要.....10
- 2.3 语句的顺序.....11
- 2.4 迭代.....13
 - 2.4.1 Java 的 while 语句和复合语句.....14
 - 2.4.2 Java 的 do 语句.....15
- 2.5 选择语句.....16
 - 2.5.1 Java 的 if 语句.....17
 - 2.5.2 Java 的 break 语句.....18
- 2.6 变量.....18
 - 2.6.1 变量的命名.....19
 - 2.6.2 赋值.....19
 - 2.6.3 变量的类型.....20
 - 2.6.4 初始化变量.....21
 - 2.6.5 常数变量（命名常量）.....22
 - 2.6.6 变量和类型检查.....22
 - 2.6.7 变量的比较.....22
- 2.7 编写一个简单的 Java 程序（程序 1）.....23
- 2.8 使用注释.....26
- 2.9 输出语句.....28
- 2.10 输入语句.....29
- 2.11 交互式程序.....30

- 2.11.1 程序 2——基本的输入输出.....30
- 2.11.2 程序 3——使用字符串.....31
- 2.11.3 程序 4——使用表达式.....34
- 2.12 小结.....39
- 2.13 复习题.....39
- 2.14 练习题.....40
- 2.15 挑战题.....41

第3章 向程序中增加结构.....42

- 3.1 引言.....42
- 3.2 抽象和封装.....43
- 3.3 方法.....43
 - 3.3.1 从方法返回值.....45
 - 3.3.2 void 方法和类型.....46
 - 3.3.3 方法的参数.....46
 - 3.3.4 多个方法参数.....48
 - 3.3.5 递归.....49
- 3.4 如何在程序中使用方法.....51
 - 3.4.1 例程：乘法表程序.....51
 - 3.4.2 例程：回文程序.....53
- 3.5 过程分解.....58
- 3.6 封装.....59
 - 3.6.1 作用域与生存期.....59
 - 3.6.2 命名与作用域.....60
- 3.7 更多的运算符.....61
 - 3.7.1 递增、递减和赋值运算符.....61
 - 3.7.2 算术表达式和类型.....62
 - 3.7.3 条件运算符.....63
- 3.8 更多的控制语句.....63
 - 3.8.1 for 循环语句.....63
 - 3.8.2 switch 语句.....64
- 3.9 更多的实例程序.....65
 - 3.9.1 例程：字符三角形 (CharacterTriangle) 程序.....65
 - 3.9.2 例程：字符矩形 (CharacterRectangle) 程序.....66

3.9.3 例程：二进制转十进制 (ConverBinaryToDecimal)程序	68	5.2.4 创建新的图形	117
3.10 小结	74	5.3 绘图的属性	117
3.11 复习题	74	5.3.1 颜色	117
3.12 练习题	75	5.3.2 线宽	118
3.13 挑战题	76	5.3.3 图形的填充	118
第4章 容器类介绍	78	5.4 绘制文本	119
4.1 引言	78	5.5 程序例程	120
4.2 数组	78	5.5.1 例程：棋盘程序	120
4.2.1 声明数组	78	5.5.2 例程：螺旋线程序	122
4.2.2 数组索引	80	5.5.3 例程：图形程序	124
4.2.3 Foreach 循环	81	5.6 小结	129
4.2.4 数组用作方法参数	81	5.7 复习题	130
4.2.5 数组用作返回值	82	5.8 练习题	130
4.2.6 求平均数 (Mean) 程序	82	5.9 挑战题	131
4.2.7 Array 类	85	第6章 类和对象	132
4.2.8 多维数组	85	6.1 引言	132
4.2.9 数组初始化表达式	87	6.2 创建新的数据类型	132
4.2.10 AddUpColumns 程序	87	6.2.1 栈对象	133
4.2.11 main 方法的参数	88	6.2.2 Stack 类	134
4.2.12 字符数组和字符串	91	6.3 通用类	138
4.3 容器 (Container) 类	91	6.4 方法的命名和作用域	141
4.3.1 ArrayList 类	92	6.5 对象初始化	143
4.3.2 容器类和基本类型	93	6.5.1 实例变量的直接初始化	143
4.3.3 Foreach 循环	95	6.5.2 构造函数 (Constructor)	144
4.3.4 修改 Mean 程序	95	6.5.3 实例初始化块	145
4.4 数据文件	97	6.5.4 多种方法如何选择	146
4.4.1 文本数据	97	6.6 对象和引用	146
4.4.2 用 FileInput 类读取数据	98	6.6.1 删除旧的对象	147
4.4.3 用 FileOutput 类写数据	100	6.6.2 对象、引用和基本类型	148
4.4.4 DisplayTextFile 程序	101	6.6.3 对象与方法参数	149
4.4.5 Copy 程序	103	6.6.4 复制对象	150
4.4.6 WordCount 程序	106	6.6.5 相等性和对象	151
4.5 小结	108	6.6.6 null	152
4.6 复习题	109	6.6.7 this 操作符	153
4.7 练习题	109	6.7 静态变量和静态方法	154
4.8 挑战题	110	6.8 类的几个例子	156
第5章 画图	111	6.8.1 例程：Name 类	157
5.1 引言	111	6.8.2 例程：Matrix 类	161
5.2 绘图	111	6.8.3 例程：Graph 类	163
5.2.1 画一条线段	113	6.9 用类和对象进行程序设计	168
5.2.2 画矩形、正方形、椭圆和圆	115	6.9.1 对象用来提供服务	168
5.2.3 其他的标准图形	116	6.9.2 类与封装	169
		6.10 枚举类型	169
		6.11 示例——创建桥牌	171

6.12 小结	176	8.5 捕获异常	218
6.13 复习题	176	8.6 finally 块	220
6.14 练习题	177	8.7 使用异常的重要性	220
6.15 挑战题	177	8.8 例子	221
第 7 章 类关系详述	179	8.8.1 带异常的 ConvertBinaryToDecimal 程序	221
7.1 引言	179	8.8.2 以更复杂方式编写的 Copy 程序	224
7.2 关联	179	8.8.3 重写 WordCount 程序	232
7.2.1 对象所有权 (Ownership)	180	8.8.4 带异常处理的 Stack 类型	234
7.2.2 有效地使用关联	180	8.9 小结	237
7.3 继承	180	8.10 复习题	237
7.3.1 使用继承	181	8.11 练习题	238
7.3.2 protected	182	第 9 章 使用线程的并发	239
7.3.3 继承和构造函数	183	9.1 一次做多件事情	239
7.3.4 超类	184	9.2 线程	239
7.3.5 Queue<T>与 Dequeue<T> 的实现示例	184	9.3 使用线程	241
7.3.6 静态变量和继承	186	9.4 线程同步	241
7.3.7 隐藏名称	187	9.5 线程调度	242
7.3.8 方法覆盖和动态绑定	187	9.6 示例程序	243
7.4 关联与继承之间的抉择	189	9.6.1 文本时钟	243
7.5 继承的层次体系	191	9.6.2 共享队列	247
7.5.1 Object 类	191	9.6.3 网络应用程序	250
7.5.2 抽象类	192	9.7 小结	256
7.6 接口和类型一致性	193	9.8 复习题	256
7.6.1 可替换性、类型和一致性	194	9.9 练习题	256
7.6.2 接口及其实现	195	第 10 章 用户界面	257
7.7 比较对象的相等性	197	10.1 引言	257
7.7.1 Class 类和 instanceof 运算符	198	10.1.1 可视化组件	258
7.7.2 一个惯例	199	10.1.2 事件和事件驱动系统	258
7.8 嵌套 (Nested) 类	200	10.1.3 AWT 和 Swing	258
7.9 包 (package)	201	10.2 核心的 GUI 概念	259
7.10 例程: 再次研究 Matrix 类	203	10.2.1 例程: HelloGoodbye 程序	259
7.11 可复用性与组件	211	10.2.2 创建窗口	261
7.12 小结	212	10.2.3 创建 Swing GUI 组件	261
7.13 复习题	213	10.2.4 容器和布局管理器	262
7.14 练习题	213	10.2.5 布局管理器优点	264
7.15 挑战题	213	10.2.6 不同的布局管理器	264
第 8 章 异常	214	10.2.7 窗口显示	265
8.1 问题是什么	214	10.2.8 处理事件	266
8.2 错误的种类	215	10.2.9 运行和终止 HelloGoodbye 程序	267
8.3 表示异常	216		
8.4 抛出异常	216		

10.2.10 HelloGoodbye 程序的总结	268
10.3 使用 GUI 作为文本输入	268
10.4 实例：一个非常简单的文本编辑器	270
10.5 菜单	274
第 2 部分 程序设计过程	
第 11 章 程序设计过程	283
11.1 引言	283
11.1.1 面向对象的软件工程	283
11.1.2 敏捷方法	284
11.1.3 设计文档	284
11.2 为什么要用面向对象方法	285
11.2.1 一个比喻——路线计划	285
11.2.2 问题域	286
11.2.3 封装	286
11.3 开始任务	287
11.3.1 概览	287
11.3.2 初始问题陈述	287
11.3.3 使用故事收集用户需求	288
11.3.4 功能需求和用例	289
11.3.5 分析	290
11.3.6 设计、实现和测试	292
11.3.7 交付、部署和维护	293
11.3.8 复审和迭代	294
11.4 测试策略	295
11.4.1 单元测试和功能测试	295
11.4.2 验收测试	296
11.4.3 调试	297
11.5 UML 类、对象和序列图	298
11.5.1 类图	298
11.5.2 包	303
11.5.3 对象图	303
11.5.4 序列图	304
11.5.5 临时表示法	305
11.6 实践和经验	306
11.7 小结	306
11.8 复习题	307
11.9 练习题	307
11.10 挑战题	308
第 12 章 单元测试	309
12.1 引言	309
12.1.1 测试是什么	309
12.1.2 到底为什么要测试呢	310
12.2 单元测试——第一个示例	310
12.2.1 类及其测试	311
12.2.2 测试是如何运行的	312
12.3 单元测试的核心原则	314
12.3.1 问题	314
12.3.2 此代码不能够进行测试	315
12.4 测试驱动的开发	315
12.4.1 重构	315
12.4.2 重复	316
12.4.3 红色/绿色/重构	316
12.4.4 按照规范进行测试	318
12.5 TestNG 框架	318
12.5.1 测试循环	319
12.5.2 TestNG 测试类	319
12.5.3 配置方法	320
12.5.4 测试方法	320
12.5.5 异常	321
12.5.6 断言	321
12.5.7 testng.xml 文件	322
12.6 扩展 Person 类	323
12.6.1 重构	327
12.6.2 无效日期	329
12.6.3 日期验证	331
12.6.4 类的状态	333
12.6.5 现在在哪里呢	337
12.7 小结	337
12.8 复习题	338
12.9 练习题	338
12.10 挑战题	339
第 13 章 测试驱动的程序设计策略	340
13.1 引言	340
13.2 第一步——搜索文件	341

13.2.1	真实文件与模拟文件	343	14.3.5	clean 目标	383
13.2.2	使用真实文件	343	14.3.6	init 目标	384
13.2.3	到目前为止的回顾过程	347	14.3.7	compile.classes 目标	384
13.2.4	模拟	347	14.3.8	compile.tests 目标	385
13.2.5	完整的 FileNameSearch 类	352	14.3.9	test 目标	385
13.3	GUI	353	14.3.10	run 目标	386
13.3.1	GUI 布局	353	14.3.11	javadoc 目标	386
13.3.2	测试 GUI 布局	353	14.3.12	deploy 目标	387
13.3.3	GUI 类	357	14.3.13	all 目标	388
13.3.4	创建 GUI	361	14.3.14	使用构建文件	388
13.3.5	选择按钮	363	14.4	版本控制	389
13.3.6	模拟对象	366	14.4.1	CVS 和 Subversion	389
13.3.7	搜索按钮	368	14.4.2	Subversion 系统信息库	390
13.4	完整的搜索器	371	14.4.3	开始项目	390
13.5	小结	372	14.4.4	检验修订版	391
13.6	复习题	372	14.4.5	更新、登记和冲突	392
13.7	练习题	373	14.4.6	添加、删除和移动	393
13.8	挑战题	373	14.4.7	修订	393
第 14 章	编程工具	374	14.4.8	Status 和 Logs	394
14.1	引言	374	14.4.9	diff	394
14.2	项目结构	374	14.4.10	复制和分支	395
14.2.1	src 目录	375	14.4.11	进一步研究	395
14.2.2	build 目录	375	14.5	集成开发环境	396
14.2.3	lib 目录	376	14.5.1	使用 IDE	396
14.2.4	doc 和 javadoc 目录	376	14.5.2	IDE 和项目	396
14.2.5	reports 目录	376	14.5.3	GUI 构建器	399
14.2.6	deploy 目录	376	14.5.4	调试器	400
14.2.7	保证文件安全	377	14.5.5	UML 工具	401
14.3	Ant——构建工具	377	14.5.6	命令行工具	402
14.3.1	Ant 概述	378	14.5.7	项目 Web 站点	403
14.3.2	项目元素	381	14.6	小结	403
14.3.3	属性	381	14.7	复习题	403
14.3.4	path 元素	382	14.8	练习题	404
			14.9	挑战题	404
			第 3 部分	程序开发案例	
第 15 章	案例研究概述	407	16.2	简要介绍	410
15.1	引言	407	16.3	回顾——一些研究	411
15.2	案例研究	408	16.3.1	XML	411
15.3	案例研究的介绍	408	16.3.2	DOM 和 SAX	413
第 16 章	邮件合并系统	410	16.3.3	JDOM 和 dom4j	414
16.1	引言	410	16.3.4	初始试验	414
			16.4	数据存储	422
			16.4.1	初始分析	422

16.4.2 初始设计思想	423	17.1 引言	463
16.4.3 具体设计	426	17.2 最初的问题定义	463
16.5 GUI 设计	435	17.3 最初的思考	464
16.5.1 初始需求分析	435	17.3.1 交通灯	464
16.5.2 GUI 的体系结构和设计	435	17.3.2 车辆指挥灯	464
16.6 显示邮件合并列表	435	17.3.3 行人指挥灯	465
16.6.1 鼠标操作	436	17.4 首轮设计	466
16.6.2 显示邮件合并细节	440	17.5 获得 GUI	479
16.7 菜单和操作	449	17.6 GUI 设计	491
16.8 更多操作	462	17.7 控制中的问题	498
16.9 练习题	462	17.8 进一步设计	509
16.10 挑战题	462	17.9 小结	509
第 17 章 行人交叉路口模拟	463	17.10 练习题	509
		17.11 挑战题	509
第 4 部分 Java 编程语言详述			
第 18 章 Java 语言参考	513	19.5.3 描述	524
18.1 引言	513	19.5.4 例子	524
18.2 语法和语义	513	19.6 类型	526
18.3 基本描述	514	19.6.1 基本类型	526
18.4 示例程序	514	19.6.2 引用类型	527
18.5 小结	516	19.6.3 自动类型转换	527
第 19 章 变量、类型和表达式	517	19.7 作用域	528
19.1 引言	517	19.7.1 目的	528
19.2 注释	517	19.7.2 概述	528
19.2.1 目的	517	19.7.3 描述	528
19.2.2 概述	517	19.7.4 例子	529
19.2.3 语法	517	19.8 变量	529
19.2.4 描述	518	19.8.1 变量声明	530
19.2.5 例子	519	19.8.2 变量初始化	531
19.3 标识符	522	19.8.3 数组变量	533
19.3.1 目的	522	19.9 表达式和运算符	537
19.3.2 概述	522	19.9.1 基本表达式	537
19.3.3 描述	522	19.9.2 运算符	538
19.3.4 例子	523	19.10 源文件	547
19.4 Unicode 转义符	523	19.11 小结	548
19.4.1 目的	523	19.12 复习题	548
19.4.2 概述	523	19.13 练习题	548
19.4.3 描述	523	第 20 章 程序流程控制	550
19.4.4 例子	524	20.1 引言	550
19.5 字面量	524	20.2 选择	550
19.5.1 目的	524	20.2.1 if 语句	550
19.5.2 概述	524	20.2.2 switch 语句	554
		20.3 迭代	558

20.3.1 while 循环语句	559	21.6.1 目的	610
20.3.2 do 循环语句	561	21.6.2 概述	611
20.3.3 for 循环语句	563	21.6.3 语法	611
20.3.4 循环行为检验	566	21.6.4 描述	611
20.3.5 foreach 循环语句	567	21.6.5 例子	612
20.4 跳转语句	568	21.7 静态加载	613
20.4.1 break 语句	568	21.7.1 目的	613
20.4.2 continue 语句	570	21.7.2 概述	613
20.4.3 return 语句	572	21.7.3 语法	613
20.5 递归	574	21.7.4 描述	613
20.6 小结	575	21.7.5 例子	614
20.7 复习题	575	21.8 小结	615
20.8 练习题	575	21.9 复习题	615
第 21 章 类和包	576	21.10 练习题	615
21.1 引言	576	第 22 章 继承和接口	616
21.2 类	576	22.1 引言	616
21.2.1 类声明	576	22.2 继承	616
21.2.2 public、private 和 protected	578	22.2.1 继承的概念	616
21.2.3 实例变量	579	22.2.2 关键字 private 和 protected 与继承性	618
21.2.4 实例初始化器	580	22.2.3 Final 类	620
21.2.5 静态变量或类变量	581	22.2.4 方法覆盖 (Overriding)	621
21.2.6 静态初始化器	583	22.2.5 Final 方法	623
21.2.7 方法	584	22.2.6 方法调用表达式	624
21.2.8 静态方法或类方法	588	22.2.7 修改后的成员类名字查找	630
21.2.9 构造函数	590	22.2.8 构造函数与继承	631
21.2.10 关键字 this	593	22.2.9 超类 (Super) 和隐含命名	633
21.3 顶层类	595	22.2.10 抽象 (Abstract) 类	634
21.3.1 目的	595	22.2.11 抽象 (Abstract) 方法	635
21.3.2 概述	595	22.2.12 从类对象继承的方法	637
21.3.3 语法	595	22.3 接口	642
21.3.4 描述	595	22.3.1 接口声明	643
21.4 嵌套类	595	22.3.2 实现	645
21.4.1 顶层嵌套类	595	22.4 注释 (Annotation)	647
21.4.2 成员类	598	22.4.1 目的	648
21.4.3 局部类	602	22.4.2 概述	648
21.4.4 匿名类	605	22.4.3 语法	648
21.5 枚举类型	607	22.4.4 描述	648
21.5.1 目的	607	22.4.5 例子	649
21.5.2 概述	607	22.5 小结	649
21.5.3 语法	608	22.6 复习题	649
21.5.4 描述	608	22.7 练习题	649
21.5.5 例子	609		
21.6 包	610		

第 23 章 异常处理	650	24.2.4 描述	664
23.1 引言	650	24.2.5 例子	665
23.1.1 异常类	650	24.3 同步方法	669
23.1.2 Try、Catch 与 Finally	652	24.3.1 目的	669
23.1.3 异常的传递	656	24.3.2 概述	669
23.1.4 抛出 (Throws) 异常声明	657	24.3.3 语法	669
23.1.5 关键字 throw	659	24.3.4 描述	669
23.2 小结	661	24.3.5 例子	670
23.3 复习题	661	24.4 同步语句	673
第 24 章 线程与并发	662	24.4.1 目的	673
24.1 引言	662	24.4.2 概述	673
24.2 Thread 类	663	24.4.3 语法	673
24.2.1 目的	663	24.4.4 描述	673
24.2.2 概述	663	24.4.5 例子	673
24.2.3 语法	663	24.5 小结	674
		24.6 复习题	674
		24.7 练习题	675
第 5 部分 附 录			
附录 A 术语表	679	附录 E Input 类	694
附录 B CRC 方法	685	附录 F FileInputStream 类	702
B.1 引言	685	附录 G FileOutputStream 类	704
B.2 基本概念	685	附录 H DrawFrame 类	708
B.3 CRC 卡片	685	附录 I DrawPanel 类	711
B.4 CRC 方法过程	686	附录 J 参考文献	713
B.5 一些问题	687	J.1 特性和 Java 使用	713
B.6 文档	687	J.2 单元测试和重构	714
B.7 设计	687	J.3 面向对象分析、设计和编程	714
附录 C 在线文档	688	J.4 通用程序设计	715
附录 D 运行 Java 程序	691	J.5 算法和数据结构	715
D.1 从这里开始	691	J.6 并发和线程	715
D.2 编译和运行 Java 程序	691	J.7 工具	716
D.3 使用 Input、FileInput 和 FileOutput 类	692	J.8 人机交互	716
D.4 使用 DrawFrame 和 DrawPanel 类	692		
D.5 使用 CLASSPATH	692		