

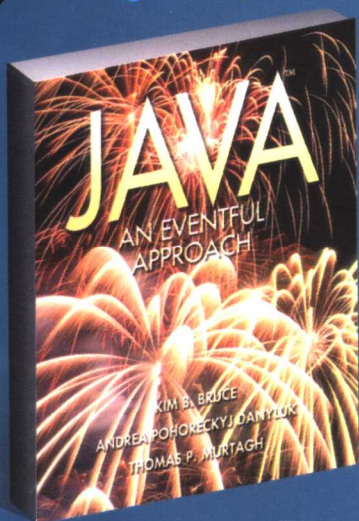


国外经典教材·计算机科学与技术

PEARSON
Prentice
Hall

Java面向事件编程

Java: An Eventful Approach



计算机教育顶级专家编写

从对象和事件入手

更快掌握核心的Java特性

Kim B. Bruce
Andrea Pohoreckyj Danyluk
Thomas P. Murtagh
杨战伟 著
译



清华大学出版社

国外经典教材·计算机科学与技术

2337

2007

Java 面向事件编程

Kim B. Bruce

Andrea Pohoreckyj Danyluk 著

Thomas P. Murtagh

杨战伟 译

清华大学出版社
北 京

Simplified Chinese edition copyright © 2007 by **PEARSON EDUCATION ASIA LIMITED** and
TSINGHUA UNIVERSITY PRESS.

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

Original English language title: *Java : An Eventful Approach* by Kim B. Bruce & Andrea Pohoreckyj
Danyluk & Thomas P. Murtagh, Copyright © 2006

EISBN: 0-13-142415-7

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: 01-2007-0602 号

版权所有, 侵权必究。侵权举报电话: **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

Java 面向事件编程 / (美) 布鲁斯 (Bruce, K. B.), (美) 达尼洛克 (Danyluk, A. P.), (美) 穆尔塔吉 (Murtagh, T. P.) 著; 杨战伟译. —北京: 清华大学出版社, 2007.2

书名原文: *Java: An Eventful Approach*

ISBN 978-7-302-14466-3

I. J… II. ①布… ②达… ③穆… ④杨… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 001128 号

责任编辑: 常晓波

封面设计: 久久度文化

责任校对: 张 剑

责任印制: 何 芊

出版发行: 清华大学出版社 地 址: 北京清华大学学研大厦 A 座

http://www.tup.com.cn 邮 编: 100084

c-service@tup.tsinghua.edu.cn

社 总 机: 010-62770175 邮购热线: 010-62786544

投稿咨询: 010-62772015 客户服务: 010-62776969

印 刷 者: 北京国马印刷厂

装 订 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 36.25 字 数: 967 千字

版 次: 2007 年 2 月第 1 版 印 次: 2007 年 2 月第 1 次印刷

印 数: 1~3000

定 价: 69.00 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: 021893-01

致 谢

献给我的妻子 Fatma——Kim

献给我的孩子 Stephan 和 Katya, 以及我的丈夫 Andrew——Andrea

献给我的妻子 Fern——Tom

出版说明

PUBLISHER'S NOTE

国外经典教材·计算机科学与技术

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

编审委员会

国外经典教材·计算机科学与技术

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

序 言

PREFACE

国外经典教材·计算机科学与技术

基于 Java 的基础课程给教师和学生都提出了新的挑战。虽然 Java 要比 C++ 简单,但 Java 作为一种具有大量标准库的面向对象语言这一事实既带来了新的复杂性,同时也带来了新的机会。

本书作为计算机科学基础教材,提供了一种新的 Java 编程教学方式,它具有以下一些特点:

1. 一种对象优先的编程方式
2. 大量使用面向对象的图形
3. 尽早使用并发
4. 从一开始就使用事件驱动的编程

初看上去,以上主题列表对于基础性课程而言显得有点不堪重负,但是这些特性的协同效应可以在介绍 Java 编程时获得令人惊讶的良好效果,尤其是当使用我们开发的 `objectdraw` 库时会更有帮助。

目标读者

本书的主要目标读者是计算机科学专业的大学一年级本科生、对编程感兴趣的其他专业的大学生,以及学习计算机科学预修课程的高中学生。特别值得一提的是,本书涵盖了 AP A 考试内容。

写给学生

神秘的漂亮 O 号

在本序言开头列出了本书中我们认为特别令人兴奋的一些特点。它们是:

1. 一种对象优先的编程方式
2. 大量使用面向对象的图形

3. 尽早使用并发
4. 从一开始就使用事件驱动的编程

此时你可能想知道这些漂亮的口号究竟意味着什么。

Java 是一种面向对象 (object-oriented) 程序设计语言的例子。就像人类社会具有许多不同类型的语言一样, 计算机语言同样存在各种不同的类型。面向对象编程语言只是一种计算机编程语言。由于编程语言各有不同, 因此应当按照不同的方式来进行教学。由于 Java 是面向对象的语言, 因此我们打算按照一种适合该语言特点的方式来介绍它。

从本书一开始, 将学习如何编写涉及简单图形的程序——例如: 矩形、椭圆、直线等。甚至还将学习使用并发 (concurrency) 机制编写可以创建图形动画的程序。我们发现学生和教师都喜欢编写那些涉及简单而有趣的图形的程序。除了可以给人带来乐趣之外, 图形还非常具体直观。当程序在窗口中绘制和操作图形对象时, 可以实际了解到程序正在进行的工作。我们发现这一点对于初学编程的人很有帮助。

此外我们还尽早介绍了事件驱动的编程 (event-driven programming)。尽管也许你没有听说过“事件驱动的编程”这一名词, 但是很可能已经对此有所熟悉。如果通过拉下菜单、单击图标, 或者使用鼠标拖放项目等方式和计算机交互, 实际上就是在和事件驱动的程序交互。你要学习编写的这种程序允许用户通过鼠标移动、按钮、滚动条等组件来与它们交互。

如何阅读本书

对于学习编程而言, 实践是相当重要的一个组成部分。因此, 在阅读本书时, 其中提供了许多实践机会。每一章都包含有习题, 可以让你检验自己是否理解了书中的内容。在阅读本书时带上纸和笔, 这样在一边学习时可以一边完成简短的习题作业。此外, 在每一章的末尾, 都会看到该章的复习题以及编程题。完成这些复习题有助于确定自己是否理解了该章介绍的主要概念。在觉得自己对这些习题有把握之后, 再尝试完成编程题, 做过的习题越多, 学习 Java 语言的效果就越好。

写给教师

本书的特点

一个针对对象优先教学方式的图形库

在本书的第一个例子中使用了图形, 并且构建了一个真正的面向对象图形类库。我们相信图形对于介绍面向对象编程是一个很好的基点, 原因有以下一些。

首先, 图形是良好的对象示例。图形类 (例如, 带有边框和填充的矩形和椭圆) 提供了良好的对象示例, 因为它们具有形状 (位置和大小) 以及一个有用的方法集合, 这些方法不仅仅用于获取和设置实例变量。其次, 我们的 `objectdraw` 库中的图形类为初学编程者提供了极佳的视觉反馈。在创建一个图形对象时, 它会立即显示在屏幕上。当对 `objectdraw` 库中的对象进行

移动或调整其大小时，屏幕上的图像会立即发生变化。因此，如果程序包含一个逻辑错误，则可以立即在屏幕上反映出来。第三，图形提供了可激发兴趣的例子。使用图形时，对于学生而言，即使非常简单的程序也会变得很有趣。此外，在介绍了动画后，在介绍数组之前很容易提供有趣的例子。最后，图形在课程中可以延续。教师可以在本书的整个课程中一直使用图形库，而不是在介绍了一些对象例子后就将它们丢弃。

我们的 `objectdraw` 库不仅提供了图形类，还提供了一个 `WindowController` 类，这个类通过在窗口中心安装一个 `DrawingCanvas` 来扩展 `JApplet` 类。`DrawingCanvas` 是 `JComponent` 类的扩展，它可以跟踪画布（`canvas`）上的对象，并在必要时重绘它们。这样就使初学编程的人减少了使用图形的复杂性。

事件驱动的编程

本书的一些作者赞成在基础性课程中介绍事件驱动的编程方式，因为学生每天所操作的“现实”程序就是以一种事件驱动的方式工作的。在学生使用计算机的过程中，他们很少会看到响应逐行文本输入的程序。因此，事件驱动的编程对于学生而言更有启发性。

我们认为在基础性课程中介绍事件驱动的编程方式在教学上还有一些其他的重要好处。其中一个非常重要的好处是学生能够从一开始就获得编写方法的体验。而且，本书中介绍的方法一般都很短。

在我们的 `objectdraw` 库中提供了一个环境，在该环境中，初学者可以通过定义简单的鼠标事件处理方法来学习编程。例如，我们的 `onMouseDown` 方法类似于标准 Java 的 `mouseDragged` 方法，只不过前者具有更简单的参数。由于在鼠标拖动过程中会重复调用该方法，因此在不使用循环的情况下也可以构建有趣的程序。因此使用事件驱动的编程使我们可以在讨论了类的定义之后再介绍循环，同时仍然为学生提供了令他们感兴趣的程序。

学生通过编写具有固定名称和参数数目的方法，能够获得编写方法和使用参数的体验，这样有助于简化对这些概念的介绍。例如，本书中所有的鼠标事件处理方法都采用单一的 `Location` 参数来表示事件发生时鼠标所在的位置。因此学生习惯于在相关的方法体中使用这些形参，与此同时，学生可以在图形命令中使用实参。

学生在编写具有精心指定的名称和参数的事件驱动的方法时获得的经验以及编写代码来发送具有实参的消息到图形对象时获得的经验，可以使学生很容易地过渡到自行设计和编写自己的类的方法。虽然学生仍然需要理解参数传递过程中的“如何操作”以及“为什么要这么做”等内容，但是他们已经了解并编写了许多例子，这有助于他们编写并理解自己的类。

对象优先

图形和事件驱动编程的结合可以为本书的对象优先教学方式提供支持。学生从本书第 1 章便开始学习使用图形库对象的程序示例。这些示例包含创建新图形对象并发送消息给它们的代码。此外，这些程序使用的类都是 `WindowController` 类的扩展。

由于我们图形库中的 `WindowController` 类是 Java 的 `JApplet` 类的一个扩展，因此学生不需要从静态 `main` 方法开始，也不必了解静态方法和非静态方法之间的区别。相反，他们只需要编写对鼠标事件进行响应的实例方法。因此从本书第 1 章开始，便要求学生使用对象并编写自己的方法。

在本书第 6 章，学生开始学习如何编写自己的类。该章位于介绍循环的章节之前，并且位于介绍条件语句的章节之后。我们使用事件驱动编程的方式使得在本书前面就有机会构建和使

用令人感兴趣的类。

尽早使用并发

我们发现，如果精心挑选所用的示例以避免争用条件，学生就能够更容易地理解并发编程的概念。毕竟，我们这个世界就是并发运转的，因此对于学生而言，使多个线程并发执行没有什么不自然的。此外，使用并发机制编写许多应用程序比把它们编写成单线程要容易一些。

在我们的图形库中提供了一个 `ActiveObject` 类来支持使用和管理线程。从学生的角度来看，我们提供的 `ActiveObject` 类和 Java 内置的 `Thread` 类之间的区别在于：前者提供了 `sleep` 方法的一种变体，这种 `sleep` 方法不会抛出异常。`ActiveObject` 还在后台对线程进行管理，使得当某个程序（或者网页中的 Java 小程序）终止时，所有的线程将优雅地终止。

为什么引入一个库

我们选择引入一个库来支持我们的教学方法，因为这样可以在本书中及早减少句法和概念上的复杂性。尽管我们一开始就使用了这个库，但是本意并不是要讲授一种与 Java 所支持的编程完全不同的编程方式。我们的想法是：不仅尽可能早地提供对事件驱动编程方式的支持，同时向学生讲授 Java 中的“正确”编程方式。

在 Java 中及早使用事件驱动的编程可能存在的一个障碍是必须介绍用于处理事件的一些语言和类库特性。例如，必须介绍侦听器、接口、Java 事件等。此外，如果要实现某个类（例如，`MouseListener` 类），则它必须实现所有的鼠标侦听方法，即使在程序中只需要一个方法就足够了。

而我们引入的库减少了这种复杂性，因为来自该库的 `WindowController` 类实现了鼠标侦听器接口。它还提供了事件处理方法，这些方法接受鼠标的位置 `Location` 作为参数，而不是使用一个更一般的 `MouseEvent`。获取一个 `Location`（一种代表屏幕上某个位置点的库类型，但它使用的是 `double` 而不是 `int` 类型）作为参数的好处是：可以立即获得有用的信息，而不需要程序员首先得把这些信息提取出来。最后，使用我们提供的库，学生只需要编写实际用于程序中的事件处理方法即可。

第 11 章向学生讲授了标准的 Java GUI 组件。此外还讲授了标准的 Java 事件模型。学生将学习把侦听器与用户界面组件相关联，并编写方法来处理所生成的事件。因此在那一章他们将开始学习在不使用所提供库的情况下如何进行编程，不过那时他们已经对所需要的概念有了足够的了解。

如前所述，在不使用我们所提供库的情况下，介绍线程时将需要讨论异常，以便能够暂停某个线程，此外，一个非常糟糕的做法是：在第一个讨论异常的例子中，介绍了那些必须使用 `sleep` 方法来处理的异常，因为它并没有多少需要处理的地方。由于我们认为在后期课程中（例如，在讨论 I/O 时）介绍异常会有更好的启发性，同时也由于它们涉及到继承和子类型的复杂性，因此我们将库设计成可以推迟讨论异常。我们并没有把这个库用作避免讲授 Java 关键组成部分的借口。而是用它来提供一种从教学方法上讲更合理的方式，来展示各个要介绍的概念。

教师的补充材料

教师可以访问 <http://eventfuljava.cs.williams.edu> 来获得联机补充材料。这些材料包括

objectdraw 库、丰富的示例程序集合、使用这个库和本书对应内容的实验作业，以及详细的讲义。

示例程序包括本书中已经包含的程序以及大量的其他示例。补充示例丰富多样，大大充实了本书所介绍的内容。在有些情况下，这些额外的示例程序强调一定的相关性（这和我们编写本书的方式相反，我们希望尽可能地减少各章节之间的相关性）。许多额外的示例涉及 **ActiveObject**，更确切地讲，涉及动画。它们并不总是适合于在课本中作为最佳示例，因为课本是静态的媒体，但是教师可以在课堂上或实验室中极其有效地使用它们。

教师和学生的灵活性

在本书中，我们旨在为读者提供最大程度的灵活性。我们希望读者能够阅读第 1、2、3、4、6、7 和 9 章的核心基础内容。这些章介绍了 `objectdraw` 库、条件、类、`while` 循环以及并发。第 5 章和第 8 章则对第 3 章和第 6 章介绍的主题进行了详细说明。同时，它们还介绍字符串以及声明和作用域的主题。尽管它们很重要，但是如果有必要，这两章可以在以后再学习。

在所有章中，以星号标记的可选章节可以跳过不读。

本书剩下的内容则根据在课程中介绍它们的顺序来讨论多个主题。我们发现这个介绍顺序很不错。但是，前面提到我们的一个目标是尽可能避免各个主题之间的相关性，因此教师可以根据自己的学生和实际情况来调整这些课程。图 0-1 显示了第 10~19 章之间的依存关系。特别注意，递归（以及递归数据结构）和数组可以按照任何顺序来介绍。此外也可以在介绍递归或数组之前介绍继承性。

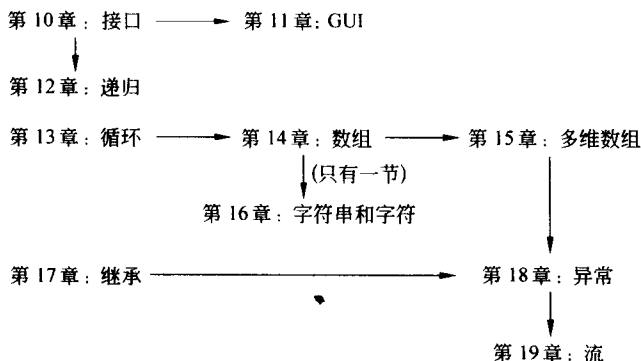


图 0-1 核心基础内容之后的各章节之间的依存关系

除了图 0-1 所示的依存关系之外，还要注意的重要一点是后面的章节要求学生对 GUI 有一定的了解。特别地，这些章节中的例子使用了 `JTextField` 以及 `setText` 和 `getText` 方法。

第 20 章“查找和排序”是一个高级主题，并且假定学生了解数组和递归。不过，有关查找内容的章节同时给出了所介绍算法的迭代版本和递归版本。

第21章“面向对象设计简介”的后半部分假定学生熟悉数组和递归。但是该章的前半部分只要求熟悉核心章节即第1~9章的内容。如果有必要,读者可以将学习设计单元的过程分为两个阶段,在学习了核心的基础内容后学习前半部分,然后在学习了更高级的主题后再学习后半部分。



最后，要提到的重要一点是，本书并不是关于 Java 编程语言的参考大全。我们只是根据适合初学者的水平来介绍 Java 语言的成分。例如，在本书后半部分的一些章节中，虽然对一些重要的概念进行了介绍，但是并没有给出水平更高的学生可能需要了解的细节。我们的目标是为学生提供坚实的基础，并希望他们在获得更多的经验后，将能够对 Java 语句具有更深入全面的了解。

额外的实用内容

在本书的几个附录中加入了大量有用的额外内容。附录 A 提供了编程风格指导原则。尽管 Java 程序员需要遵循一些标准的约定，但是一些编程风格问题显然是比较主观化的。学生在学习时应当注意教师可能提供的自身编程心得。

附录 B 总结了 objectdraw 库中可用的类和方法。在附录 C 中，介绍了浏览 Java API 文档的过程。API (Application Programming Interface, 应用程序编程接口) 指定了程序员在使用某个库所提供的资源时必须了解的细节。我们介绍了 objectdraw 库的部分 API，但是由于许多 API 的文档遵循一个标准格式，因此读者应当能够自行阅读其他的 API 文档。

从我们的 objectdraw 库过渡到标准 Java 相当直观。附录 D 总结了标准 Java 等价于 objectdraw 库的一些特性。

致谢

如果没有许多人的帮助和反馈，本书是不可能问世的。首先要感谢威廉斯学院的大学同事：Barbara Lerner、Jim Teresco、Steven Freund 和 Stacia Wyman，他们很乐意并甘愿使用我们所提供的多少有点粗略的教案以及不是特别稳定的 Java 图形库来进行教学。而且在过去几年中，威廉斯学院的上百名非常积极地接受了一个挑战，即使用我们的教材来体验一种了解计算机科学的新教学方式。我们要特别感谢本科教学助理们，他们中的许多人努力学习新的教材并使得其中的内容更容易被学生接受。此外，威廉斯学院还有许多其他的学生和我们一起开发图形库，编写实验和家庭作业问题及其解答，测试图形库并设计课程的网页。这些学生包括：Peter Applegate、Jing Cao、Brendan Dougherty、Marcus Duyzend、Cheng Hu、Jonathan Kallay、Krishna Kannan、Christine Osterman 和 Ashok Pillai。

在开发我们的教材时，一个很重要的部分是获得来自其他学院的教师的反馈，这些老师在他们的课程中使用了我们的教材。尽管在此由于篇幅所限，不能将他们一一列出，我们挑选出以下人员来感谢他们对我们的教材提出的极具价值的反馈：Barbara Adler、Mary Courtney、Chris Haynes、David Housman、Lonnie Fairchild、James Taylor 和 Douglas Troy。特别感谢 Chris Nevison 在他为高中预修计算机科学课程的老师讲课时使用 objectdraw 库。

Prentice Hall 出版社的外聘审稿人员为本书提供了非常详细的评论，尽管我们并没有一一采纳所有这些建议，但是他们的详细反馈意见使得我们能够对本书的内容进行多处改进和澄清。这些审稿人员包括：印第安那大学的 Chris Haynes、Goldey-Beacom 学院的 Ricard Albright、罗彻斯特理工学院的 Henry A. Etlinger、伦斯勒技术学院的 Carlos A. Varela、卡内基梅隆大学的 Ricard E. Pattis、新墨西哥州立大学的 Ron Zacharski、罗彻斯特理工学院的 Trudy Howles、莱恩

社区大学的 Mark Williams、佩斯大学的 Mary Courtney、德安扎学院的 Mary Ann May-Pumphrey、戈申学院的 David Housman、得克萨斯州农业及工业大学的 Lawrence C. Petersen、西卡罗来纳大学的 Mark A. Holliday、萨斯喀彻温应用科学和技术学院的 Gavin T. Osborne 以及路易斯维尔大学的 Adel S. Elmaghraby。

我们特别感谢 Prentice Hall 出版社的各位编辑：Petra Recter，是他和我们签约编写本书；Kate Hargett，在 Petra 换到一个新部门后是 Kate 接替了他的工作，她不顾刚刚生完第一个小孩后的辛劳就开始着手编辑我们发给她的这本书的原稿；还有 Tracy Dunkelberger，他一直负责本书直到它付印为止。我们还要感谢我们的制作编辑 Irwin Zucker。

正如我们在为基础课程开发这种新的教学方式时所获得的乐趣那样，我们希望你能夠享受使用本书的乐趣。同时我们很乐意倾听有关本书以及相关材料的评论和建议。在网页 <http://eventfuljava.cs.williams.edu> 上可以找到我们的联系方式。

参与本书翻译的人员有：杨战伟、陈宗斌、陈征、陈婷、陈红霞、管学岗、王新彦、金惠敏、张海峰、张瑜、许瑛琪、张靖、张景友、戴锋、张德福、张士华、张锁玲、杜明宗、高玉琢、王涛、许勇光、张新丽、乔健、徐丽萍、李杰、徐昌春、江辉、江丽华、彭青、王信花、郑来强、胡仁衡、郑来勇、薛秋丕、杨月华、丁桂英、赵琰、王晔、孙海霞和徐春艳等人。

第 1 章 什么是编程 1

1.1 在并不了解的情况下	2
1.2 Java 编程语言	5
1.3 Java 初体验	6
1.3.1 简单的响应程序	7
1.3.2 “Class” 和其他关键字	8
1.3.3 关于方法的介绍	9
1.4 编程工具	11
1.5 绘图原始命令	14
1.5.1 图形坐标系	14
1.5.2 构建图形对象	16
1.6 其他事件处理方法	19
1.6.1 鼠标事件处理方法	19
1.6.2 begin 方法	20
1.7 错误分析	21
1.8 小结	23
1.9 本章复习题	24
1.10 编程题	25

第 2 章 Java 中的名称 26

2.1 命名和修改对象	26
2.1.1 修改器方法	26
2.1.2 实例变量声明	28
2.1.3 为变量名指定含义	29
2.1.4 注释	30
2.1.5 其他的修改器方法	31
2.1.6 习题	33
2.2 非图形类对象	35
2.2.1 颜色类	35

2.2.2 Location 类 37

2.3 画布上的层次	40
2.4 访问鼠标的位置	41
2.5 在方法之间共享参数信息	42
2.6 小结	47
2.7 本章复习题	47
2.8 编程题	48

第 3 章 处理数字 50

3.1 访问器方法介绍	50
3.2 访问画布的数值属性	52
3.3 表达式和语句	53
3.4 算术表达式	55
3.5 数值实例变量	58
3.6 初始值设定项	60
3.7 命名数值变量	61
3.8 显示数值信息	63
3.8.1 将数字作为 Text 显示	64
3.8.2 使用 System.out.println	65
3.8.3 混合文本和数字	67
3.9 随机数	69
3.10 小结	71
3.11 本章复习题	72
3.12 编程题	73

第 4 章 进行选择 76

4.1 一个简单的例子：使用 if 语句统计 选票	76
4.2 if 语句	79

4.3 了解条件	81	6.6 重载方法和构造函数	145
4.4 在多个选项之间选择	84	6.7 小结	148
4.5 有关 Boolean 表达式的更多内容	87	6.8 本章复习题	148
4.6 嵌套条件	90	6.9 编程题	149
4.7 小结	94		
4.8 本章复习题	95	第 7 章 控制结构	151
4.9 编程题	97	7.1 重复过程和 while 循环	151
第 5 章 基本类型、运算符和字符串	99	7.2 使用 while 循环的更多例子	155
5.1 运算符和方法调用	100	7.3 可以计数的循环	157
5.2 不同但相等	103	7.4 嵌套循环	158
5.3 运算符和优先级规则	105	7.5 控制结构风格指导意见	161
5.4 Double 类型	106	7.6 第摩根法则和复杂的 boolean 表达式	164
5.4.1 Java 需要多种数值类型	107	7.7 简化条件语句的语法	166
5.4.2 使用 double 和 int 的算术	108	7.8 switch 语句*	172
5.4.3 显示 double 值	111	7.9 小结	174
5.4.4 将有理数称为 double 的原因	112	7.10 本章复习题	174
5.4.5 选择某种数值类型	113	7.11 编程题	175
5.5 数值信息的方便来源	113		
5.5.1 确定时间信息	113	第 8 章 声明和作用域	178
5.5.2 高级数学计算	115	8.1 访问控制: public 和 private	178
5.6 字符串	119	8.2 使用实例变量、参数和局部变量	181
5.7 本章复习题	122	8.3 标识符的作用域	183
5.8 编程题	124	8.4 使用 static	186
第 6 章 类	126	8.5 小结	187
6.1 一个没有类的例子	126	8.6 本章复习题	187
6.2 编写 FunnyFace 类	129	第 9 章 活动对象	189
6.2.1 实例变量	130	9.1 动画	189
6.2.2 方法和参数	130	9.2 活动对象	190
6.2.3 编写访问器方法	131	9.3 Image 和 VisibleImage	192
6.2.4 构造函数	132	9.4 与活动对象交互	194
6.2.5 组装到一起	134	9.5 使活动对象影响到其他的对象	197
6.3 加入方法到 FunnyFace	136	9.5.1 与非活动对象交互	197
6.3.1 加入一些类似已编写方法 的方法	137	9.5.2 构建其他活动对象的 活动对象	198
6.3.2 间接定义方法	137	9.6 不带循环的活动对象	201
6.3.3 使用 this 作为参数	139	9.7 使动画效果更平滑	202
6.4 另外一个例子: 实现 Timer 类	140	9.8 有关计时的更多提示	203
6.5 局部变量	142	9.9 小结	205

9.10	本章复习题	205
9.11	编程题	206
第 10 章	接口	207
10.1	提供灵活性的接口	207
10.1.1	第一个例子	207
10.1.2	将对象和类与接口相关	208
10.1.3	使用接口	212
10.1.4	接口的其他特性	214
10.1.5	小结	215
10.2	在 <code>objectdraw</code> 库中使用接口	215
10.3	扩展接口*	218
10.4	小结	220
10.5	本章复习题	220
10.6	编程题	221
第 11 章	Java 中的图形用户界面	223
11.1	文本字段	223
11.1.1	加入文本字段到窗口中	225
11.1.2	从文本字段获得信息	226
11.2	Java 中的按钮和事件	227
11.2.1	创建和加入按钮	228
11.2.2	处理事件	228
11.3	在程序中使用 GUI 组件的步骤 清单	229
11.4	组合框	230
11.5	面板和布局管理器	235
11.5.1	面板	235
11.5.2	有关布局管理器的更多 内容	238
11.6	其他 GUI 组件	240
11.6.1	滑块	240
11.6.2	标签	243
11.6.3	<code>JTextField</code> 和 <code>JTextArea</code>	244
11.7	处理击键和鼠标事件	247
11.7.1	击键事件	247
11.7.2	鼠标事件	249
11.8	小结	250
11.9	本章复习题	252
11.10	编程题	253

第 12 章	递归	254
12.1	递归结构	255
12.1.1	嵌套矩形	255
12.1.2	使用递归结构来创建并 搜索数据集合	263
12.1.3	设计递归结构	266
12.1.4	为什么能够奏效	268
12.1.5	花菜图案	269
12.2	递归方法	274
12.2.1	快速求幂	275
12.2.2	汉诺塔	277
12.3	小结	280
12.4	本章复习题	280
12.5	编程题	281
第 13 章	Java 中的一般循环	282
13.1	确定循环模式	282
13.2	计数和 <code>for</code> 循环	283
13.2.1	使用 <code>for</code> 循环的例子	284
13.2.2	<code>for</code> 循环的其他变化形式	287
13.3	<code>do-while</code> 循环	288
13.4	避免循环错误	289
13.4.1	循环次数相差 1 的错误	289
13.4.2	无限循环	290
13.4.3	在终止条件中使用 <code>double</code> 类型	291
13.5	小结	292
13.6	本章复习题	293
13.7	编程题	294
第 14 章	数组	295
14.1	声明数组名称	295
14.2	创建数组	296
14.3	使用数组: 一个三角类	300
14.4	Java 1.5 中的增强型 <code>for</code> 循环	303
14.5	收集数组信息	304
14.5.1	超速驾驶驾驶员计数	306
14.5.2	绘制直方图	307
14.6	包含变量大小的集合	311

14.6.1 并行数组和对象数组.....	313	16.2.6 比较字符串.....	372
14.6.2 跟踪集合的大小.....	314	16.3 字符.....	374
14.6.3 加入项目到数组.....	315	16.3.1 字符和字符串.....	375
14.6.4 显示结果.....	316	16.3.2 对字符进行操作.....	377
14.7 添加和删除数组元素.....	322	16.4 小结.....	379
14.7.1 加入某个元素到一个顺序 数组.....	322	16.5 本章复习题.....	381
14.7.2 从数组删除某个元素.....	328	16.6 编程题.....	383
14.8 小结.....	330	第 17 章 继承	385
14.9 本章复习题.....	331	17.1 特殊化扩展.....	385
14.10 编程题.....	335	17.2 使用 extends.....	386
第 15 章 多维数组	338	17.3 protected 和 public.....	391
15.1 一般的二维数组.....	338	17.4 设计用于扩展.....	394
15.1.1 声明数组的数组.....	339	17.5 继承的层次结构.....	397
15.1.2 创建数组的数组.....	339	17.6 重写方法定义.....	401
15.1.3 对数组的数组进行索引.....	341	17.6.1 object 类以及 equals 和 toString 方法.....	403
15.1.4 遍历二维数组.....	342	17.6.2 使用 super 访问重写的 方法.....	403
15.1.5 超越二维: 扩展日历程序.....	344	17.6.3 动态方法调用.....	405
15.2 矩阵.....	347	17.6.4 有计划的废弃.....	406
15.2.1 魔力方块.....	348	17.6.5 抽象类和抽象方法.....	410
15.2.2 声明和构建矩阵.....	348	17.7 小结.....	411
15.2.3 遍历矩阵.....	348	17.8 本章复习题.....	411
15.2.4 填充魔力方块.....	351	17.9 编程题.....	417
15.2.5 数字图像平滑.....	353	第 18 章 异常	420
15.3 小结.....	357	18.1 异常处理程序.....	421
15.4 本章复习题.....	357	18.2 异常是对象.....	424
15.5 编程题.....	359	18.3 异常传递.....	424
第 16 章 字符串和字符	362	18.4 处理多个异常.....	426
16.1 小字符串和大字符串.....	362	18.5 异常子类.....	427
16.1.1 空白字符串.....	362	18.6 检查异常和未检查异常.....	429
16.1.2 长字符串.....	363	18.7 抛出异常.....	432
16.2 有用的 String 方法集锦.....	365	18.8 小结.....	433
16.2.1 创建一个 URL 字符串.....	365	18.9 本章复习题.....	434
16.2.2 使用 indexOf 来查找 子字符串的位置.....	365	18.10 编程题.....	436
16.2.3 处理小写和大写.....	367	第 19 章 流	438
16.2.4 剪切和粘贴字符串.....	369	19.1 文本流.....	439
16.2.5 修整字符串.....	371		