

TURING

图灵计算机科学丛书

Java之父
James Gosling
作序推荐

实用Java教程

基于BlueJ的对象优先方法 (第3版)

Objects First with Java: A Practical Introduction Using BlueJ

Third Edition

[英] David J. Barnes Michael Kölling 著
翁 恺 译



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵计算机科学丛书

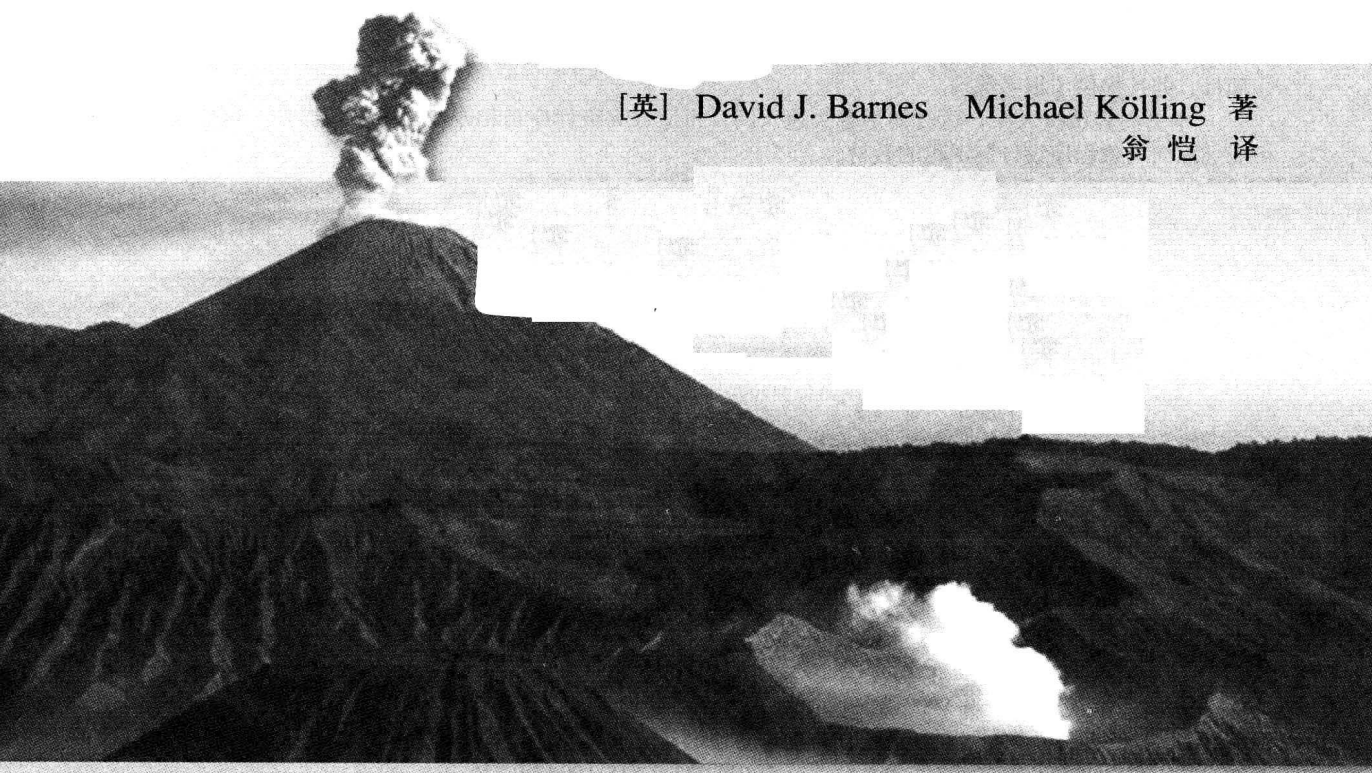
实用Java教程

基于BlueJ的对象优先方法

(第3版)

Objects First with Java: A Practical Introduction Using BlueJ

[英] David J. Barnes Michael Kölling 著
翁 恺 译



人民邮电出版社
北京

图书在版编目 (CIP) 数据

实用Java教程: 基于BlueJ的对象优先方法: 第3版/ (英) 巴恩斯 (Barnes, D. J.), (英) 科灵 (Kölling, M.) 著; 翁恺译. 北京: 人民邮电出版社, 2009. 1

(图灵计算机科学丛书)

书名原文: Objects First with Java: A Practical Introduction Using BlueJ

ISBN 978-7-115-19042-0

I. 实… II. ①巴… ②科… ③翁… III. JAVA语言—程序设计
IV. TP312

中国版本图书馆CIP数据核字 (2008) 第165561号

© Pearson Education Limited 2003, 2005, 2006.

This translation of *OBJECTS FIRST WITH JAVA: A PRACTICAL INTRODUCTION USING BLUEJ, Third Edition* is published by arrangement with Pearson Education Limited.

本书中文简体字版由Pearson Education授权人民邮电出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

内 容 提 要

本书在软件工程的大背景下, 通过BlueJ这一创新性的开发环境, 全面讲述了面向对象和Java程序设计的基本知识。书中从面向对象的基础知识讲起, 介绍了对象和类; 然后深入到应用结构, 讲解了继承、抽象技术、构建图形用户界面、错误处理; 最后给出了一个完整的案例。全书按照项目驱动的方式来展开, 讨论了大量的程序项目, 并且提供了很多练习。

本书内容由浅入深, 适合初学者快速入门, 也适合高级程序员和专业人士学习参考, 可作为计算机相关专业“面向对象程序设计”课程的教材。

图灵计算机科学丛书

实用Java教程: 基于BlueJ的对象优先方法 (第3版)

◆ 著 [英] David J. Barnes Michael Kölling

译 翁 恺

责任编辑 杨海玲

执行编辑 刘 静

◆ 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>

北京昌平百善印刷厂印刷

◆ 开本: 787×1092 1/16

印张: 25.5

字数: 703 千字

2009年1月第1版

印数: 1-4 000 册

2009年1月北京第1次印刷

著作权合同登记号 图字: 01-2007-3070号

ISBN 978-7-115-19042-0/TP

定价: 49.00元

读者服务热线: (010) 88593802 印装质量热线: (010) 67129223

反盗版热线: (010) 67171154

序

——James Gosling (Java之父)

看着女儿Kate和她的中学同学使用商业IDE吃力地学习Java课程，真是痛苦。复杂的工具给学习增加了很大的难度。我多么希望能更早地了解教学过程中到底出现了什么问题，结果课程都结束了我才真正明白问题所在。这种情况下最适合的是BlueJ。

BlueJ是一个负有这样一种使命的交互式开发环境：为正在学习编程的学生而设计。它是由每天在教室里面对这样的问题的教师们设计的。我曾经与BlueJ的开发者畅谈，他们非常清楚自己的目标。开发过程的讨论更加注重实现，而非承诺。BlueJ非常简洁而且目标明确。

不过，这本书不是关于BlueJ的，而是关于编程的，用Java编程。

历经几年，Java已经成为编程教学中广泛使用的语言。这是有许多原因的。其中之一是Java有许多易于教授的特性，它的定义相对比较简洁，编译器做了大量的静态分析，能在早期告诉学生问题出在哪里，它有一个非常强壮的内存模型，可以消除对象边界越界或类型系统失配时产生的很多“神秘”错误。另外，Java在商业上也变得日益重要。

本书直面最难教的概念：对象。这个概念伴随学生从第一步开始一路走向非常艰深的概念。

它解决了写编程书时最棘手的问题，即如何处理实际键入和运行程序的机制。大多数书都悄悄地跳过这个问题，或只是轻轻地一笔带过，留给教师来解释。同时也给教师留下了沉重的负担：要把上课教的内容与学生做作业所需要的步骤——联系起来。本书则使用BlueJ把理解概念和指导学生写程序完美地结合在一起。

多么希望我女儿去年就能够用上这本书呀，或者明年……

给教师的前言

本书向初学者介绍面向对象程序设计，从软件工程师的角度来讲述基本的面向对象和编程的概念。

第1章是写给毫无编程经验的人看的，而后面几章则适合比较有经验甚至是专业的程序员。实际上，有非面向对象语言经验而想要涉足面向对象的程序员也能从本书中获益。

书中采用两种工具将所介绍的概念付诸实践：Java程序设计语言和Java开发环境BlueJ。

Java

选择Java有两个原因，即语言的设计及其流行的程度。Java程序设计语言本身提供了绝大多数重要的面向对象概念的非常清晰的实现，同时非常适合作为入门级教学语言。它的广泛流行确保了有大量的支持资源可用。

在任何领域，有广泛的信息来源是非常有用的，对老师和学生都是这样。特别是Java，已经有了数不尽的各种形式和风格的书、教程、练习、编译器、环境和测验题。许多都是在线、免费的。许多高质量的支持材料使得Java成为教授面向对象编程的完美选择。

既然已经有了这么多Java材料，还有必要再写一本书吗？当然还有，因为我们使用的开发环境是BlueJ。

BlueJ

关于第2种工具BlueJ需要多说几句。本书的独特性正体现在与BlueJ开发环境的紧密结合上。

BlueJ是由南丹麦大学、澳大利亚迪金大学以及位于英国坎特伯雷的肯特大学开发和维护的一个Java开发环境，专为教授面向对象编程入门而设计。与其他环境相比，它具有以下几个优点。

- 用户界面很简单。给初学者介绍20分钟，他就能学会BlueJ的基本用法。然后，教师就可以重点讲解面向对象和Java中的重要概念，而不需要再浪费时间来讨论环境、文件系统、类的路径、DOS命令或是DLL冲突。
- 支持其他环境所没有的教学工具。其中之一就是类结构的可视化表达。BlueJ自动显示一个类似UML的图来表达工程中的类和类之间的关系。可视化可以极大地帮助教师和学生双方。如果在屏幕上只看见代码行就很难掌握对象的概念。为了适应初学者，图的标注就是UML的子集，因此非常易于理解，而且又能够在后续课程中方便地转移到UML。
- BlueJ环境最重要的优势是用户可以直接创建任何类的对象，然后与它们的方法交互。这样只要一点点开销就可以直接使用对象做实验。学生几乎可以“感觉”到创建对象、调用方法、传递参数或接收返回值的意义。写好一个方法立刻就可以验证它，不再需要写测试驱动程序。这个功能对于理解抽象的概念和语言细节是很有价值的。

BlueJ是一个完整的Java环境，并不是功能缩减的、只用于教学的Java简化版本。它是基于Sun公司的Java开发包的，使用了标准的编译器和虚拟机。这保证了它总是遵循官方最新的Java

规范。

本书的作者已经将BlueJ环境用于教学数年，之前更长时间没有BlueJ教学时的经历，深感BlueJ的使用在课程中可以大大增强学生的参与、理解和主动性。本书作者之一也是BlueJ系统的开发者之一。

真正从对象开始

选择BlueJ的理由之一是它让教师可以真正地先解决最重要的概念。“从对象开始”已经被许多作者和教师喊了很久了。遗憾的是，Java语言没有使这个崇高的目标轻易实现，它需要克服大量的语法和细节，然后才能体验到第一个鲜活的对象。最小的可以创建和调用对象的Java程序至少应包括：

- 一个类；
- 一个main方法，签名中就包括了诸如静态方法、参数和数组等概念；
- 一条创建对象的语句（"new"）；
- 为变量赋值；
- 变量声明，包括变量的类型；
- 方法调用，使用点符号；
- 可能还有一个参数列表。

结果，这方面的教材往往是要么先介绍完上述各个部分，然后才能在第4章左右接触到对象；要么就是用"Hello, world"风格的有一个静态main方法的程序作为第一个程序例子，却一个对象也不创建。

有了BlueJ，这些都不是问题。学生一开始就可以创建对象、调用方法。因为用户可以直接创建对象并与对象交互，诸如类、对象、方法和参数这样的概念在接触Java语法之前就可以以很具体的方式讨论。不用在这里解释更多，我们建议读者深入阅读第1章，然后一切就会明了了。

迭代的方法

本书另一个重要的方面就是它遵循了迭代的风格。在计算机教育界，一个广为人知的教育设计模式指出，重要的概念应该尽早提出，然后反复讲述^①。但是教材的作者会在讲述一个知识点的时候禁不住一次把关于这个知识点的所有内容都讲完。比如，一般在介绍类型的时候，都会给出内置数据类型的完整列表；在介绍循环概念的时候，都会讨论所有可能的循环。

这两种方法是矛盾的：我们不能先集中讨论重要的概念，同时又要介绍所有相关主题的完整内容。我们从教材中得出的体会是，如此多的细节会分散注意力，反而把重点丢了，使重点知识更难掌握。

本书中每个重要的概念会多次提及，可能出现在同一章中也可能散布在整本书中。首先引入重要概念，以便理解和应用现有任务，然后再在不同的语境中不断回顾，并随着学习的深入理解也不断加深。这个方法也有助于解决频繁出现的概念之间互相依赖的问题。

有些教师可能不熟悉迭代的方法。读完前面几章后，过去习惯于顺序讲解的教师会惊讶地发现那么早就引入了大量的概念，学习曲线看起来非常陡峭。

重要的是要理解这几章并不是全部，学生不可能立刻完全理解这些概念。这些基础概念会

^① “Early Bird”模式，见J. Bergin: “Fourteen pedagogical patterns for teaching computer science”, *Proceedings of the Fifth European Conference on Pattern Languages of Programs (EuroPLop 2000)*, Irsee, Germany, July 2000.

在整本书中不断地重复,使得学生能够越来越深入地理解它们。随着学生知识水平的提高,学习不断深入,重要概念的不断回顾使得他们最终能相当深入地理解这些概念。

我们多次将这种方法应用于教学,看起来学生比老教师更容易接受这种方法。记住,陡峭的学习曲线不是问题,只要学生能爬得上去。

不求面面俱到

与迭代方法有关,我们决定不在书中覆盖全部的Java语言细节。

本书的主要目的是教授一般的面向对象程序设计理论,而不是Java语言特殊的细节。学习本书的学生可能在今后的30或40年里从事软件专业的工作,他们的大部分工作很可能不会用到Java。每一本严谨的教材当然会给读者准备更基础的知识,而不仅仅是当前流行的语言。

另一方面,许多Java的细节对于实际的工作是很重要的。本书覆盖了尽可能多的Java细节,以展示概念并实现实际的项目,但是还是有些Java特有的结构故意留下没讲。

我们知道有些教师会讲授一些书中没有详细讨论的话题。这是预料之中的,也是必要的。不过,我们用钩子来处理这些话题,而不是试图覆盖所有可能的话题(否则本书还会更厚)。钩子就像指针,常常以问题的形式出现,提出问题并给出到附录或其他材料的引用。这些钩子使得相关的话题在适当的时候提出,留给读者或教师去决定这个话题应该被覆盖到什么程度。这样钩子就成为这个话题存在的提示。同时,作为一个占位符,表明了进一步的讨论可以在哪里插入。

教师可以决定完全按照本书教学,遵循我们建议的顺序,或是跳到正文中钩子建议的其他位置。

每一章包括了一些问题,提出了与话题有关的讨论材料,但是不在本书中展开讨论。我们特别希望教师能在课堂上开展其中一些问题的讨论,或者让学生作为家庭作业来研究出答案。

项目驱动的方法

本书中内容的介绍是按项目驱动的。书中讨论了很多编程项目,提供了很多练习。我们首先介绍项目的目标和问题,而不是先介绍新的结构,然后提供一个练习来利用这个结构去解决问题。分析手头的问题然后决定需要什么样的解决方案。因此,语言结构是随着我们解决问题的需要而引入的。

在开始的几章中,每章都提供了至少两个讨论的例子。这些项目是用来详细展示每章的重要概念的。用两个不同的例子的做法实践了迭代方法,即每个概念要在初次介绍之后的不同语境中再次回顾。

在设计这本书的时候,我们找了许多范围广泛的不同项目作为例子,希望这样能吸引读者的兴趣,有助于展示概念应用的不同场景。找到好的示例项目是不容易的。我们希望这些项目能够为教师提供良好的起点,由此设计出更丰富有趣的作业。

所有项目的实现代码都写得非常仔细,这样许多相关的内容可以通过阅读项目的源代码来得到。我们坚信通过阅读和模仿可以学到很多东西。不过,为此就必须确保给学生阅读的例子要写得很好,值得模仿。我们努力去做到这一点。

所有的项目都设计成开放的。尽管在书中详细讨论了所有项目的一个或多个版本,但它们还是可以供学生进一步扩展和改进。所有项目的完整源代码在本书网站下载^①。

① 具体网址见“给教师的前言”的“其他材料”部分。——编者注

按概念的顺序而不是语言结构的顺序

另一个使本书显得与众不同的地方是，它是以基本的软件开发的任务组织的，而不是以特殊的Java语言结构组织的。一个证据就是每章的标题。在本书中你不会见到那些传统的章标题，比如“基本数据类型”或“控制结构”。根据基本开发任务组织内容使得我们可以用更普通的名字作为章标题，而不是被复杂的编程语言结构所左右。我们还相信这样也使得学生易于理解为什么要这样安排内容，也使得阅读过程更为有趣。

这种方法导致的一个结果是不能把本书简单地当作参考书。引导性的教材和参考书有着不同的目标，甚至有些还是对立的。在某种程度上，有些书可以试着两者都满足，但是肯定要在某些方面做出妥协。本书非常明确地设计为一本教材，如果有目的冲突发生，我们坚定地坚持做为教材而非参考书。

章节顺序

第1章介绍面向对象最基本的概念：对象、类和方法。形象、直观地介绍这些概念，却完全不涉及Java语法的细节。它还让读者看了一些源代码。为此列举了一个可以交互地画不同形状图形的例子。另一个例子是一个简单的实验课报名系统。

第2章介绍了类的定义，研究如何编写Java源代码来创建对象，讨论了如何定义字段和实现方法。同时还介绍了第一种类型的语句。主要的例子是实现一个售票机系统。还对第1章实验课的例子进行了深入研究。

第3章进一步讨论了多个对象的交互，以及对象如何通过互相调用方法来合作完成一个任务。还探讨了一个对象怎样创建其他对象。一个数字时钟显示器的例子讨论了如何用两个数字显示器对象来分别显示小时和分钟。第二个主要的例子研究了一个可以在邮件客户之间发送消息的电子邮件系统仿真程序。

第4章继续构建更广泛的对象结构。最重要的是，开始使用对象的容器了。通过实现一个电子记事本和一个拍卖系统来介绍容器。与此同时，讨论了对容器的遍历，并第一次看到了循环。我们使用的第一个容器是ArrayList。这一章的后半部分介绍了数组这种特殊形式的容器，以及循环的另一种形式：for循环。通过讨论一个Web日志分析程序来练习使用数组。

第5章处理的是类库和接口。介绍了Java标准类库，并讨论了类库中一些重要的类。更重要的是，解释了如何阅读和理解类库文档。同时还探讨了在软件开发项目中编写文档的重要性，最后通过实践说明如何为自定义的类编写合适的文档。本章遇到的类的例子有Random、Set和Map。通过这些类实现了一个类似Eliza的对话系统和一个弹跳球的图形化仿真系统。

第6章的标题是“使对象正常运行”，讲述了许多关于如何定义正确的、可理解的和可维护的类的问题。探讨的主题从编写清晰易懂的代码（包括风格和注释）到测试和调试。测试策略也有所提及，还详细介绍了很多调试方法。同时使用了一个日程安排系统和一个电子计算器的实现来讨论这些话题。

第7章正式讨论了把问题领域分解为类的实现问题。介绍了类的设计，包括像责任驱动设计、耦合、内聚和重构这样的概念。通过一个交互的、基于文本界面的探险游戏（World of Zuul）来探讨上述问题。经过了几次迭代的过程来改善这个游戏内部的类的结构，并不断扩展它的功能，最后还给出了一个可以用做学生项目的长长的建议列表。

第8章和第9章介绍了继承、多态性以及许多相关的话题。讨论了一个简单的CD和视频数据库来表达这些概念，并详细讨论了代码继承、子类型、多态方法调用和覆写等问题。

第10章实现了一个捕食者-被捕食者仿真。还讨论了基于继承的抽象机制：接口和抽象类。

第11章介绍了两个新的例子：一个图像显示器和一个声音播放器。这两个例子都是用来介绍如何构造图形用户界面（GUI）的。

第12章再次讨论如何处理错误这个难题。讨论了几种可能的问题和解决方案，也详细讨论了Java的异常处理机制。扩展和改进了一个地址簿应用程序来展示这些概念。输入/输出也用一个容易产生错误的案例。

第13章再次详细讨论进一步的抽象，即如何把一个含糊描述的问题构造成类和方法。前几章都是先假设应用程序框架的大部分已经存在，然后来做改进。这一章讨论如何从头开始设计。这就需要详细讨论应该用怎样的类来实现应用程序、它们之间如何交互，以及责任应该如何分配。通过设计一个电影院订票系统，用类-责任-合作者（CRC）卡来解决这个问题。

第14章试着把所有的事情合到一起，综合了本书前面章节的许多话题。这是一个完整的案例研究，从应用程序设计开始，经过类的接口设计，一直到讨论许多重要的功能和非功能的特性和实现细节。之前章节讨论过的话题（诸如可靠性、数据结构、类设计、测试和可扩展性）又一次应用在了新的语境中。

第3版

这是本书的第3版，对之前版本的部分内容做了修改。在第2版中已经有了对JUnit的介绍，还有一章介绍了GUI编程。这个版本最大最明显的改动是使用了Java 5作为实现语言。Java 5引入了新的语言结构，比如，泛型类和枚举类型，为了使用这些新的特性几乎改写了所有例子的代码。正文中的讨论，也已经改写过。不过，总的来说，本书的原则和风格并没有改变。

我们收到的关于读者对前两个版本的反馈，也发挥了积极的作用，许多读者提出了意见和建议，帮助发现错误并告知我们，为本书网站贡献资料、通过邮件列表各抒己见，把本书翻译成其他语言以使其变得更完美。

不过，总的来说，本书还是充分有效的。所以第3版只是作了相同的风格的改进，而不是巨大的改变。

其他材料

本书的支持网站为<http://www.bluej.org/objects-first>。

在此网站上，可以找到例子的更新^①以及其他材料。比如，本书所有例子遵循的格式指南的电子版，这样教师可以修改来满足自己的需求。

网站还有一个受密码保护、只有教师可以进入的链接，提供了其他资料。

用本书授课的一系列幻灯片也在网站上。

讨论组

作者维护着两个邮件讨论组，为本书的读者和其他BlueJ用户提供交流和互助的平台。

第一个邮件列表bluej-discuss是开放的（任何人可以订阅），而且有一个开放的存档。要想加入或阅读存档，请转到<http://groups.google.com/group/bluej-discuss>。

第二个列表objects-first是封闭的，只开放给教师。它可以用来讨论答案、教学技巧、考试和其他教学相关的问题。想要加入的教师请查看本书的网站<http://www.bluej.org/objects-first>。

^① 这些更新的例子也可以从图灵网站（www.turingbook.com）下载。——编者注

致谢

许多人以各种不同的方式对本书做出了贡献，使得本书最终能得以出版。

首先，也是最重要的，是John Rosenberg。John现在是澳大利亚迪金大学的副校长。纯属偶然他才没有成为本书的作者之一。他曾是BlueJ开发的驱动力量之一，他从一开始就提出了其中的创意和教育方法，而且用了好几年的时间和我一起讨论这本书的编写。本书中的很多内容都是在与John的讨论中逐渐成形的。只是因为毕竟一天只有24个小时，而他已经有太多的其他事情要做，才使得他没能实际地来写这本书。在我们写书的时候，John也继续帮助组织内容，并且以各种方式来帮助我们改进。我们深深地感谢他的友谊和协助。

还有其他一些人帮助我们实现BlueJ：澳大利亚的Bruce Quig、Davin McCall和Andrew Patterson，英国的Ian Utting和Poul Henriksen。他们都在BlueJ上倾注了多年心血，在他们承担的其他工作之外，改进和扩展了BlueJ的设计和实现。没有他们的努力，BlueJ不会有现在这样的质量，不会像现在这样流行，而这本书也恐怕写不出来了。

创建BlueJ并使得本书能够出版，另外一个很重要的贡献来自Sun公司。Sun公司的Emil Sarpa从一开始就相信BlueJ会成功。他的支持和毫无官僚作风的协作方式，在整个写作过程中极大地帮助我们。

培生教育出版社的每一个人，都非常努力地工作，容忍我们独特的做事方式，以十分紧凑的时间表来完成本书的工作。感谢Kate Brewin从第一版开始对这个项目坚定的支持，也要感谢Simon Plumtree让这一版能见诸于众。还要感谢出版团队中的其他各位，包括Bridget Allen、Kevin Ancient、Tina Cadle-Bowman、Tim Parker、Veronique Seguin、Fiona Sharples和Owen Knight。要是还落下了谁，请接受我们的道歉。

培生的销售队伍也花费了大量的心血来使本书面世，努力避免出现每个作者最害怕的事情——出版的书可能毫不起眼，无人问津。

审稿人也很辛苦地审阅书稿，而且常常是在一年中他们最忙的时候。我们特别要感谢Michael Caspersen、Devdatt Dubhashi、Khalid Mughal和Richard Snow的鼓励和建设性意见。

将本书翻译为德文的Axel Schmolitzky，一定是我们最认真仔细的读者，他提出了相当多的改进意见，有些还是非常细微的地方。

我(David)还要表达本人对肯特大学计算机系的教师和学生的感谢。选了OO入门课程的学生可以用本书来上课，而他们所表现出来的学习的兴趣和动力，使得教书是如此愉快。而没有同事和研究生导师们的有益帮助，特别是系主任Simon Thompson所提供的大力支持，开班授课几乎是不可能的。在校园以外，很多人提供了很好的休闲和社交活动，以免我们沉溺于本书的写作：感谢我的登山伙伴，Chris Phillips和Martin Stevens，帮助我爬上顶峰；而Joe Rotchell，帮助我脚踏实地。

最后，我要感谢我的妻子Helen，她的爱是如此特别，还有我的孩子们，让我感受生命的珍贵。

我(Michael)要感谢Andrew和Bruce长时间热烈的讨论。不仅是这些技术成果，讨论的过程就很让人享受其中，我喜欢良好的辩论。John Rosenberg从我的学术生涯开始就做了很多年我的导师。没有他的接待和支持，我就没办法去澳大利亚；不是他做我的博士导师和同事，我

也不会取得今天的成就。与他共事很愉快，我欠他很多。感谢Michael Caspersen，他不仅仅是好朋友，还在我们一起参加的几次研讨会上影响了我对教学的看法。我在丹麦Mærskk研究院软件工程组的同事：Bent Bruun Kristensen、Palle Nowack、Bo Nørregaard Jørgensen、Kasper Hallenborg Pedersen和Daniel May，耐心地容忍由于我写作本书而一再耽误的每一个提交，还带我融入丹麦的生活。

最后，我要感谢我的妻子Leah和我的两个小女儿：Sophie和Feena。在我写作本书的过程中，她们经常要容忍我整天长时间地工作。她们的爱给了我继续下去的力量，使得一切努力都有了意义。

本书详细讨论的项目的列表

shapes	第1章
简单的几何图形的绘制，展示了对象的创建、方法的调用和参数的传递。	
picture	第1章
使用图形对象来画图的例子，介绍了源代码、Java语法和编译。	
lab-classes	第1章、第2章、第8章
简单的有学生类的例子，展示了对象、字段和方法。在第8章又加入了继承。	
ticket-machine	第2章
一个火车票自动售票机的仿真，进一步介绍了字段、构造器、访问器方法和修改器方法、参数和一些简单的语句。	
book	第2章
存储书的信息。增强了在ticket-machine中用过的构造器的概念。	
clock-display	第3章
一个数字时钟的实现，展示了抽象、模块化和对象交互的概念。	
mail-system	第3章
一个简单的电子邮件系统的仿真，用于演示对象的创建和交互。	
notebook	第4章
一个简单的电子记事本的实现，用于介绍容器和循环。	
auction	第4章
一个拍卖系统。更多的关于容器和循环的内容，包括了枚举器。	
weblog-analyzer	第4章
一个分析Web访问日志文件的程序，介绍了数组和for循环。	
tech-support	第5章
一个类似Eliza的对话程序，用来给顾客提供“技术支持”，介绍了类库中类的一般用法和一些特定类的特殊用法，还介绍了阅读类文档和编写类文档。	
balls	第5章
一个弹跳球的图形动画，演示了接口/实现的分离和简单的图形。	
diary	第6章
一个日程记录程序的早期实现，用来讨论测试和调试策略。	
calculator	第6章
一个桌面计算器的实现。这个例子强化了之前介绍的概念，并用于讨论测试和调试。	
bricks	第6章
一个简单的调试练习，模拟用砖块填充来做简单计算。	
world-of-zuul	第7章、第9章
一个基于文本的、交互的探险游戏。高度可扩展，是一个很棒的开放性学生项目。这里用于讨论好的类设计、耦合和内聚。在第9章再次用作继承的例子。	
dome	第8章、第9章

2 本书详细讨论的项目的列表

一个CD和视频的数据库。这个项目以及后来的扩展非常详细地介绍了继承和多态性的基础。

foxes-and-rabbits

第10章

一个典型的捕食者-被捕食者仿真，强化了继承的概念以及抽象类和接口的概念。

image-viewer

第11章

一个简单的图像显示和调整程序。主要关注如何构建一个GUI。

simple-sound

第11章

一个声音播放器，作为另一个构建GUI的例子。

address-book

第12章

一个具有可选的GUI的地址簿的实现。查询是灵活的：可以用部分的名字或电话号码来查询。这个项目用到了大量的异常。

cinema-booking-system

第13章

一个管理电影院的座位预定的系统。这个例子用来讨论发现类和设计程序。没有提供任何代码，因为这个例子表达的是一个程序从头开始的开发过程。

taxi

第14章

出租车是一个综合了预定系统、管理系统和仿真的例子。它用作把全书很多概念和技术串起来的一个情景研究的例子。

目 录

第一部分 面向对象基础

第1章 对象和类	1
1.1 对象和类	1
1.2 创建对象	1
1.3 调用方法	2
1.4 参数	3
1.5 数据类型	4
1.6 多个实例	5
1.7 状态	5
1.8 对象中的字段和方法	6
1.9 对象交互	7
1.10 源代码	8
1.11 另一个例子	9
1.12 返回值	9
1.13 对象作为参数进行传递	10
小结	11
关键术语	11
第2章 理解类的定义	13
2.1 售票机	13
2.2 检查类的定义	14
2.3 字段、构造器和方法	16
2.3.1 字段	17
2.3.2 构造器	19
2.4 通过参数传递数据	20
2.5 赋值	21
2.6 访问器方法	22
2.7 修改器方法	24
2.8 在方法中打印	25
2.9 简易型售票机的总结	27
2.10 对售票机设计的探讨	28
2.11 做选择：条件语句	28
2.12 高级条件语句的例子	31
2.13 局部变量	32
2.14 字段、参数和局部变量	33
2.15 改进型售票机的总结	34
2.16 自我复习练习题	34
2.17 回顾一个熟悉的例子	36
小结	39

关键术语	39
------	----

第3章 对象交互	42
3.1 时钟的例子	42
3.2 抽象与模块化	42
3.3 软件中的抽象	43
3.4 在时钟例子中的模块化	43
3.5 实现时钟显示器	44
3.6 类图与对象图	45
3.7 基本类型和对对象类型	47
3.8 ClockDisplay的源代码	48
3.8.1 NumberDisplay类	48
3.8.2 字符串拼接	49
3.8.3 取模运算符	50
3.8.4 ClockDisplay类	51
3.9 对象创建对象	54
3.10 多个构造器	55
3.11 方法调用	55
3.11.1 内部方法调用	55
3.11.2 外部方法调用	56
3.11.3 时钟显示器的总结	57
3.12 另一个对象交互的例子	57
3.12.1 邮件系统的例子	58
3.12.2 this关键字	59
3.13 使用调试器	60
3.13.1 设置断点	60
3.13.2 单步执行	61
3.13.3 单步进入方法	62
3.14 方法调用回顾	63
小结	63
关键术语	64
第4章 对象集合	65
4.1 在可变容量的容器中存放对象	65
4.2 个人记事本	65
4.3 初涉类库	66
4.4 容器的对象结构	67
4.5 泛型类	69
4.6 容器内的编号	69
4.7 从容器中删除对象	70

4.8 处理整个容器	71	5.10 编写类文档	116
4.8.1 for-each循环	72	5.10.1 在BlueJ中使用javadoc	117
4.8.2 while循环	73	5.10.2 类文档的组成	117
4.8.3 遍历容器	76	5.11 公共与私有	118
4.8.4 索引存取与迭代器的比较	77	5.11.1 信息隐藏	119
4.9 记事本例子的总结	77	5.11.2 私有方法和公共字段	119
4.10 另一个例子: 拍卖系统	78	5.12 通过接口研究类	120
4.10.1 Lot类	78	5.13 类变量和常量	122
4.10.2 Auction类	79	5.13.1 static关键字	122
4.10.3 匿名对象	81	5.13.2 常量	124
4.10.4 使用容器	82	小结	124
4.11 可变容量容器的总结	84	关键术语	125
4.12 固定容量的容器	84	第6章 使对象正常运行	126
4.12.1 日志文件分析器	84	6.1 引言	126
4.12.2 声明数组变量	86	6.2 测试和调试	126
4.12.3 创建数组对象	87	6.3 用BlueJ进行单元测试	127
4.12.4 使用数组对象	88	6.3.1 使用检查器	130
4.12.5 分析日志文件	88	6.3.2 正向测试和反向测试	132
4.12.6 for循环	89	6.4 测试自动化	132
小结	93	6.4.1 回归测试	132
关键术语	93	6.4.2 自动检查测试结果	134
第5章 高级行为	95	6.4.3 记录测试	136
5.1 类库的文档	95	6.4.4 夹具	138
5.2 TechSupport系统	96	6.5 模块化和接口	139
5.2.1 研究TechSupport系统	96	6.6 调试场景	140
5.2.2 阅读代码	97	6.7 注释与风格	141
5.3 阅读类文档	101	6.8 人工走查	141
5.3.1 接口与实现	102	6.8.1 高层次的走查	142
5.3.2 使用类库中的类的方法	102	6.8.2 用走查来检查状态	143
5.3.3 检查字符串是否相等	104	6.8.3 口头走查	145
5.4 增加随机行为	104	6.9 打印语句	145
5.4.1 Random类	105	6.10 选择测试策略	148
5.4.2 在有限范围内产生随机数	105	6.11 调试	148
5.4.3 产生随机的回答	106	6.12 在实战中使用这些技术	149
5.4.4 阅读参数化的类的文档	108	小结	149
5.5 包和导入	109	关键术语	149
5.6 使用映射表示关联	109	第7章 类的设计	150
5.6.1 映射的概念	110	7.1 引言	150
5.6.2 使用HashMap	110	7.2 world-of-zuul游戏的例子	151
5.6.3 在TechSupport系统中使用映射	111	7.3 耦合和内聚的介绍	152
5.7 使用集合	113	7.4 代码重复	153
5.8 分割字符串	114	7.5 扩展	157
5.9 完成TechSupport系统	115	7.5.1 任务	157

7.5.2 找到相关的源代码	157	8.7.5 换型	200
7.6 耦合	158	8.8 Object类	201
7.7 责任驱动的设计	162	8.9 自动包装与包裹类	202
7.8 改动局部化	164	8.10 容器层次	202
7.9 隐性耦合	165	小结	202
7.10 预见性	167	关键术语	203
7.11 内聚	168	第9章 继承深入	204
7.11.1 方法的内聚	168	9.1 问题: DoME的打印方法	204
7.11.2 类的内聚	169	9.2 静态类型和动态类型	205
7.11.3 内聚对于可读性的好处	170	9.3 覆写	208
7.11.4 内聚对于重用的好处	170	9.4 动态查找方法	209
7.12 重构	171	9.5 方法中的超类调用	211
7.12.1 重构与测试	171	9.6 方法多态性	212
7.12.2 重构的例子	171	9.7 Object的方法: toString	212
7.13 语言无关的重构	174	9.8 受保护的访问	214
7.13.1 枚举类型	174	9.9 有覆写的继承的另一个例子	216
7.13.2 命令界面的进一步解耦	175	小结	218
7.14 设计原则	177	关键术语	218
7.15 脱离BlueJ执行程序	178	第10章 高级抽象技术	220
7.15.1 类方法	178	10.1 仿真	220
7.15.2 main方法	178	10.2 狐狸与兔子仿真	220
7.15.3 类方法的限制	179	10.2.1 foxes-and-rabbits工程	221
小结	179	10.2.2 Rabbit类	223
关键术语	179	10.2.3 Fox类	225
第二部分 应用程序结构		10.2.4 Simulator类: 设置	228
第8章 通过继承改进结构	181	10.2.5 Simulator类: 仿真的一步	231
8.1 DoME例子	181	10.2.6 采取措施改进仿真	232
8.1.1 DoME的类与对象	182	10.3 抽象类	232
8.1.2 DoME源代码	184	10.3.1 Animal超类	233
8.1.3 DoME程序的讨论	189	10.3.2 抽象方法	233
8.2 使用继承	189	10.3.3 抽象类	235
8.3 继承层次	191	10.4 进一步的抽象方法	237
8.4 Java的继承	191	10.5 多继承	239
8.4.1 继承与访问权限	192	10.5.1 Actor类	239
8.4.2 继承与初始化	192	10.5.2 通过抽象实现的灵活性	240
8.5 DoME: 增加另一种媒体类型	194	10.5.3 有选择地绘画	240
8.6 (迄今为止) 继承的优点	196	10.5.4 可以画的角色: 多继承	240
8.7 子类型	196	10.6 接口	241
8.7.1 子类与子类型	197	10.6.1 Actor接口	241
8.7.2 子类型与赋值	198	10.6.2 接口的多继承	243
8.7.3 子类型与参数传递	199	10.6.3 作为类型的接口	243
8.7.4 多态变量	199	10.6.4 作为规范的接口	244
		10.6.5 接口的其他例子	245

10.6.6 抽象类还是接口	246
10.7 继承的总结	246
小结	246
关键术语	246
第11章 构造图形用户界面程序	248
11.1 引言	248
11.2 组件、布局 and 事件处理	248
11.3 AWT和Swing	249
11.4 ImageViewer例子	249
11.4.1 第一个实验：创建一个框架	250
11.4.2 加入简单的组件	251
11.4.3 加入菜单	253
11.4.4 事件处理	253
11.4.5 集中接收事件	254
11.4.6 内部类	256
11.4.7 匿名的内部类	257
11.5 ImageViewer 1.0：第一个完整的版本	258
11.5.1 图像处理类	259
11.5.2 加入图像	260
11.5.3 布局	261
11.5.4 嵌套的容器	263
11.5.5 图像过滤器	265
11.5.6 对话框	267
11.6 ImageViewer 2.0：改进的程序结构	269
11.7 ImageViewer 3.0：更多界面组件	273
11.7.1 按钮	273
11.7.2 边界	275
11.8 进一步的扩展	276
11.9 另一个例子：SoundPlayer	277
小结	279
关键术语	280
第12章 错误处理	281
12.1 address-book工程	281
12.2 防御性编程	285
12.2.1 客户-服务器交互	285
12.2.2 参数检查	286
12.3 服务器错误报告	287
12.3.1 通知用户	287
12.3.2 通知客户对象	287
12.4 异常抛出原则	290
12.4.1 抛出异常	290
12.4.2 异常类	291
12.4.3 异常的影响	292
12.4.4 不受检异常	292
12.4.5 阻止对象创建	293
12.5 异常处理	294
12.5.1 受检异常：throws子句	295
12.5.2 捕捉异常：try语句	295
12.5.3 抛出和捕捉多个异常	296
12.5.4 扩散异常	298
12.5.5 finally子句	298
12.6 定义新的异常类	299
12.7 使用断言	300
12.7.1 内部一致性检查	300
12.7.2 断言语句	301
12.7.3 使用断言的指南	302
12.7.4 断言与BlueJ单元测试框架	303
12.8 错误恢复和错误避免	303
12.8.1 错误恢复	303
12.8.2 错误避免	304
12.9 案例研究：文本输入-输出	305
12.9.1 阅读器、写入器和流	306
12.9.2 address-book-io工程	306
12.9.3 用FileWriter类做文本输出	308
12.9.4 用FileReader类做文本输入	309
12.9.5 Scanner：从终端读输入	310
12.9.6 对象串行化	310
小结	311
关键术语	311
第13章 应用程序设计	312
13.1 分析与设计	312
13.1.1 动词-名词方法	312
13.1.2 电影院订票系统的例子	312
13.1.3 发现类	313
13.1.4 使用CRC卡	313
13.1.5 场景	314
13.2 类的设计	316
13.2.1 设计类的界面	317
13.2.2 用户界面设计	317
13.3 文档	318
13.4 合作	318
13.5 原型	318
13.6 软件成长	319
13.6.1 瀑布模型	319
13.6.2 迭代开发	319
13.7 使用设计模式	320