

Scala 编程

Programming in Scala

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING



内 容 介 绍

本书介绍了一种新的编程语言，它把面向对象和函数式编程概念有机地结合为整体，从而形成一种完整统一、语义丰富的新思维体系。本书循序渐进，由浅入深，经作者精心组织、仔细编排，将语言中的各种概念自然地铺陈在字里行间。除此之外，本书还包含了大量富有针对性和趣味性的示例，它们除了提供对语言各个方面的具体演示之外，还从侧面说明了如何将函数式编程的理念切实并广泛地应用到面向对象编程中。本书面向的读者是有一定编程经验的开发人员，他们希望能够开拓眼界，并致力于提高在软件开发各方面的技能。

Original English Edition Copyright © by Martin Odersky, Lex Spoon, Bill Venners.

The Chinese Translation Edition Copyright © 2010 by Publishing House of Electronics Industry in arrangement with Artima, Inc.

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 in writing from Artima, Inc.

本书中文简体版专有出版权由 Artima, Inc. 授予电子工业出版社，专有出版权受法律保护。

版权贸易合同登记号 图字：01-2010-4536

图书在版编目（CIP）数据

Scala编程 / 奥德斯基（Odersky, M.），莱斯彭（Spoon, L.），凡纳斯（Venners, B.）著；
黄海旭，高宇翔译. —北京：电子工业出版社，2010.11

书名原文：Programming in Scala

ISBN 978-7-121-12119-7

I. ①S... II. ①奥... ②莱... ③凡... ④黄... ⑤高...

III. ①JAVA语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字（2010）第212228号

策划编辑：卢鹁翔

责任编辑：杨绣国

印 刷：北京天宇星印刷厂

装 订：三河市皇庄路通装订厂

出版发行：电子工业出版社

北京市海淀区万寿路173信箱 邮编：100036

开 本：787×1092 1/16 印张：33 字数：782千字

印 次：2010年12月第1次印刷

定 价：89.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。服务热线：（010）88258888。

译者序

最初接触 Scala 是在 2008 年的 9 月。当时刚刚换了公司，工作上出现了一段“空档期”，因此非常想找一些有技术含量的事情来做。而编程语言的设计开发，或者更确切地说只是对于“银弹”的幻想，这始终是我的爱好之一。那时总会感觉自己应该为这么多年的开发工作总结一些什么，让它们变成之后工作的基础，或者即使没有那么基础也可以是一种参考，即便连参考都不算，至少也是一种思考和总结吧！

然后我重新审视了自己曾做过的项目，分析了项目中使用过的编程语言（Delphi / C++ / C# / Java），总结其中的优缺点，想象着如果是我做的话会有什么样的改善，甚至开始着手策划语言的语法结构，希望能够自圆其说，从只言片语中建立起完整的逻辑体系。在这样的过程中，我时常会发现自己钻进了死胡同，似乎不断地修改只是在不断地重复，甚至唯一的改进也只是“语法糖”而已。没有发现与其他语言间的标志性差异，不能建立独有的思维方式。

于是我在互联网上寻找有没有什么现成的、别出一格的新观点或想法，能够启发我步出自己的圈子，带给我一种“啊哈，原来是这样！”的感觉。不过真没想到，很快，我就找到了。而且是这么彻底，甚至让我打消了“从轮子造起”的念头。

不得不说，Scala 语言的确具有很新颖的创意。这在之前所用的语言中是不多见的。

相信大多数商业软件的程序员都是从 Fortran、Pascal、C/C++/Java 或 COBOL 语言一路学起的。本书中，这一大类的语言被称为是指令式语言，意指以变量和过程语句构造指令的方式对需要实现的目的进行编码的语言。与之相对的是被称为函数式语言的另一个大类，这种语言把程序的目的转换为函数和函数应用，是以函数计算这种不同的视角来处理一般问题的。

直到现在，Lisp、Scheme 或者 Haskell 这样的函数式语言也仅仅还是“小众”语言。这在 TIOBE 语言排行榜上也可见一斑。上榜的前 20 位几乎被面向对象的或者面向过程的指令式语言占满。尤其是前三甲则几乎始终被 Java、C、和 C++ 所把持。反观函数式语言，Lisp 在第 13 位，是最靠前的了，Scheme 在第 25 位，Haskell 在第 40 位左右。但并不是说函数式语言对于指令式语言就不具有可比性，相反，在很多地方函数式语言都体现了它们独到的优势，如：代码的简洁性、值的不可改变、函数的无副作用、值和函数替换的指称透明等。

但函数式语言与指令式语言总显得有点儿“格格不入”。一个被指令式语言锻炼出来的头脑很难理解函数式语言的精髓，已经用惯了循环语句的方式解决问题，想要把它转换为递归语句也不是很容易的事情。莫非，函数式语言始终只能作为学术派的思维游戏，而不能适应于现代商业软件开发的需

要，因此无缘商业软件开发浪潮的洗礼吗？

所以说，Scala 语言的确具有很新颖的创意。因为它能很巧妙地把指令式语言与函数式语言结合在一起。用 Martin Odersky 的话说，就是：指令式语言与函数式语言本就是一枚硬币的两面，它们不应是互相排斥、非此即彼的，而是可以互相协调，共同发挥出更大力量的。

Scala 语言还从许多其他的语言中汲取了长处，构造出了自己完整而优雅的思想体系。从语言的各个方面来看，似乎都存在一些有别于其他语言的特点，但从语言的整体性来看，它的每个特点都不是孤立的，而是作为一个有机的整体自然而然生发出来的功能。你可以发现，Scala 语言中的基本概念并不多，甚至与其他的语言相比，原生数据类型和语法结构还“显得”颇为不足。但 Scala 具有强大的自发展能力，能够构建出“乱真”的自定义语法库，这使得 Scala 语言成为了一种可以依赖于有限的构件创造出几乎无限可能的语言。

（写到这里我眼前总能立刻浮现出 DNA 结构的四个基本构成部分，谁能想到几乎所有复杂生命体的程序都仅仅使用 4 个单词就能够拼装出来呢？）

不过本书还不止是一部编程语言的教材，它还是一部编程方法论的说明。由于本书三位作者同时也是 Scala 语言的设计和开发者，所以想要了解个中秘密真是再也找不到更好的透露者了。通过本书你不但能够学习到这种语言的语义语法结构、应用场景、用法、实例，还能够发现语言作者在这些结构之后的考量、动机、权衡和折中，从中可以领悟到许多编程专家的思维模式，学习到他们处理问题的方法，从而开拓你的眼界，让你不仅在软件代码开发方面，而且还在软件的架构设计思想上面有新的感悟。

我必须承认，在接触 Scala 之前，我只是一个普通的 Java 开发人员，是完全“指令式”的软件从业者。是 Scala 语言开始让我领略到了函数式编程思想的奥妙。因此，借此机会想向语言的创造者表示自己的敬意，是你们非凡的工作给了我另一种全新的思考问题的方式，同时也“断送”了我“制造轮子”的梦想（笑）。

本书的翻译工作分为两部分，其中，前 27 章由本人完成，第 28 至 33 章由高宇翔翻译。

另外，还要感谢电子工业出版社不遗余力地引进国外优秀 IT 类书籍，感谢博文视点的徐定翔编辑在本书筹划阶段给予我参与的机会；感谢卢鹤翔编辑为译者在本书翻译阶段提供的大力支持；感谢刘唯一编辑和杨绣国编辑为本书的统稿和排版不辞辛劳的工作。翻译本书的过程中，译者虽已尽最大努力确保专业术语的统一和准确，也尽最大努力将原作者的意境用朴实的中文展现给读者，但囿于个人的水平，书中的问题和疏漏之处在所难免，敬请读者朋友给予批评指正。

黄海旭

2010 年 10 月于上海浦东

序

Martin Odersky 设计的 Pizza 语言曾经震惊了 Java 世界。尽管 Pizza 语言本身并不流行，但它巧妙地把面向对象和函数型语言两种风格融合在了一起，形成了自然而又强有力的组合。Pizza 语言的设计成为了 Java 泛型的基础，Martin 的 GJ (Generic Java) 编译器从 Java 1.3 开始成为了 Sun 公司的标准编译器（尽管关闭了泛型）。我有幸维护这个编译器多年，因此对 Martin 设计与实现语言的能力有非常直接的体会。

那时候我们还在 Sun 公司，尝试用一些零打碎敲的、针对特定问题的解决方案来扩展语言，比如 for-each 循环、枚举、自动装包，以简化程序的开发，而 Martin 则继续着他在更强大的正交语言原语方面的工作，帮助程序员用库来提供解决方案。

近年来，静态类型语言受到了冲击。Java 的广泛应用暴露了静态语言编程会导致大量固定写法的弊病。常见的看法是我们应避免静态类型从而消除这种代码，于是人们对动态语言如 Python，Ruby 和 Groovy 的兴趣开始增加。这种看法被 Martin 最近的构想，Scala 的出现打破。

Scala 是一种类型优雅的语言：它是静态类型的，但仅在需要的地方显式定义类型。Scala 从面向对象和函数式语言两方面获得了强大的特性，然后用全新的理念把它们完美地整合成一体。它的语法是如此的轻量级，而原语（primitive）又如此富有表达力，以至于根本可以认为 API 的使用不会产生语法开销。我们可以在标准库中，如拆分器、组合器和执行器中发现例子。而这也说明 Scala 是支持内嵌的特定领域语言。

Scala 会成为下一个伟大的语言吗？只有时间可以证明一切。我相信 Martin Odersky 的小组绝对有这样的能力和水平做到这一点。不过有一件事是确定无疑的：Scala 语言建立了衡量未来语言的新标准。

Neal Gafter
圣约瑟，加利福尼亚
2008 年 9 月 3 日

致谢

许多人持续关注本书及其相关资料，在这里表示感谢。

Scala 语言本身已经成为许多人努力的集合。版本 1.0 的设计和实现得到了 Philippe Altherr、Vincent Cremet、Gilles Dubochet、Burak Emir、Stéphane Micheloud、Nikolay Mihaylov、Michel Schinz、Erik Stenman 和 Matthias Zenger 等人的帮助。Iulian Dragos、Gilles Dubochet、Philipp Haller、Sean McDirmid、Ingo Maier 和 Adriaan Moors 参与了第二版和当前版语言及工具开发的工作。

Gilad Bracha、Craig Chambers、Erik Ernst、Matthias Felleisen、Shriram Krishnamurti、Gary Leavens、Sebastian Maneth、Erik Meijer、David Pollak、Jon Pretty、Klaus Ostermann、Didier Rémy、Vijay Saraswat、Don Syme、Mads Torgersen、Philip Wadler、Jamie Webb 和 John Williams 通过生动和启发性的讨论，或者对此篇文稿早期版本的评注，热情地与我们分享了他们的观点，从而使语言的设计得以成型。Scala 语言电子邮件列表的建设者们同样提供了非常有用的反馈信息来帮助我们改善语言及相关工具。

George Berger 为本书更为流畅的制作过程和 Web 体验做出了巨大的贡献。令人欣慰的是这个项目没有变成一个技术大杂烩。

许多人给我们的早期版本提供了反馈信息。在这里感谢 Eric Armstrong、George Berger、Gilad Bracha、William Cook、Bruce Eckel、Stéphane Micheloud、Todd Millstein、David Pollak、Frank Sommers、Philip Wadler 和 Matthias Zenger。同样感谢硅谷模式组成员他们大有助益的审校：Dave Astels、Tracy Bialik、John Brewer、Andrew Chase、Bradford Cross、Raoul Duke、John P. Eurich、Steven Ganz、Phil Goodwin、Ralph Jocham、Yan-Fa Li、Tao Ma、Jeffery Miller、Suresh Pai、Russ Rufer、Dave W. Smith、Scott Turnquest、Walter Vannini、Darlene Wallach 和 Jonathan Andrew Wolter。我们同样还要感谢 Dewayne Johnson 和 Kim Leedy 在封面设计上的帮助，还有 Frank Sommers 在索引上的工作。

我们要特别感谢所有那些曾经提供给我们建设性意见的读者。你们的建议对我们把本书做得更好提供了非常大的帮助。我们没办法印出所有参与者的名字，但以下是在 eBook 预印刷阶段通过点击建议链接提供了至少五条评论的读者名单，首先以最高评论数排序，然后是字母顺序，感谢这些人：David Biesack、Donn Stephan、Mats Henricson、Rob Dickens、Blair Zajac、Tony Sloane、Nigel Harrison、Javier Diaz Soto、William Heelan、Justin Forder、Gregor Purdy、Colin Perkins、Bjarte S. Karlsen、Ervin Varga、Eric Willigers、Mark Hayes、Martin Elwin、Calum MacLean、Jonathan Wolter、Les Pruszyński、Seth Tissue、Andrei Formiga、Dmitry Grigoriev、George Berger、Howard Lovatt、John P. Eurich、Marius Scurtescu、Jeff Ervin、Jamie Webb、Kurt Zoglmann、Dean Wampler、Nikolaj Lindberg、Peter McLain、Arkadiusz Stryjski、Shanky Surana、Craig Bordelon、Alexandre Patry、Filip Moens、Fred Janon、Jeff Heon、Boris

Lorbeer、Jim Menard、Tim Azzopardi、Thomas Jung、Walter Chang、Jeroen Dijkmeijer、Casey Bowman、Martin Smith、Richard Dallaway、Antony Stubbs、Lars Westergren、Maarten Hazewinkel、Matt Russell、Remigiusz Michalowski、Andrew Tolopko、Curtis Stanford、Joshua Cough、Zemian Deng、Christopher Rodrigues Macias、Juan Miguel Garcia Lopez、Michel Schinz、Peter Moore、Randolph Kahle、Vladimir Kelman、Daniel Gronau、Dirk Detering、Hiroaki Nakamura、Ole Hougaard、Bhaskar Maddala、David Bernard、Derek Mahar、George Kollias、Kristian Nordal、Normen Mueller、Rafael Ferreira、Binil Thomas、John Nilsson、Jorge Ortiz、Marcus Schulte、Vadim Gerassimov、Cameron Taggart、Jon-Anders Teigen、Silvestre Zabala、Will McQueen、还有 Sam Owen。

最后，Bill 还要感谢 Gary Cornell、Greg Doench、Andy Hunt、Mike Leonard、Tyler Ortman、Bill Pollock、Dave Thomas 和 Adam Wright 对本书出版方面提供的观点和建议。



简介

本书是 Scala 编程语言的教程。写给那些直接参与 Scala 开发的人群。我们的目标是通过阅读此书，你能够学会一切所需，成为多产的 Scala 程序员。本书中所有的例子都能在 Scala 版本 2.7.2 下面编译通过。

谁应该阅读此书

本书的主要目标读者是那些想要学习使用 Scala 编程的程序员。如果你打算用 Scala 做下一个软件项目，那么本书是为你准备的。而且，本书还会传授一些新概念以满足那些希望拓展视野的程序员们。比如说，如果你是 Java 程序员，阅读本书将使你领略从函数型编程到高级面向对象思想的许多概念。我们相信学习 Scala 及它隐含的理念，通常能使你成为更好的程序员。

本书假设你已经有了基本的编程知识。当然 Scala 也可以很好地作为首次学习的编程语言，但这不是学习如何编程的书。

从另一方面来说，阅读本书并不需要特定的编程语言知识。尽管大多数人在 Java 平台上使用 Scala 语言，但本书并不设定你具备 Java 基础知识。然而，我们希望读者能够熟悉 Java，这样我们可以在某些时候通过比较 Scala 和 Java 来帮助这些读者明白其中的差别。

如何使用本书

本书的主要目的是作为教材，所以推荐的阅读方式是按照章节的次序，从头到尾完成阅读。我们尽力一次介绍一个话题，并且仅用已经介绍过的话题来说明新的话题。因此，如果你跳到后面想先睹为快，就可能会发现有些东西用了不太明白的概念来解释。我们认为按照章节的顺序阅读，这种一步一个脚印的方式将引导你顺利地获得 Scala 开发的技能。

如果你发现一个不懂的术语，请一定查找一下术语表和索引。许多读者会略过书中的某些部分，这不是问题，术语表和索引有助于你返回到你略过的某些东西。

读完一遍之后，本书还可以作为语言参考书。Scala 语言有正式的定义，但是语言的定义是以可读性变差为代价换取精确性好的文档。尽管本书并未涵盖 Scala 的所有细节，但在你更好地掌控 Scala 编程之前，它已经足够作为一本通信易懂的语言参考书了。

如何学习 Scala

简单地通读本书，你将学到 Scala 的许多东西。但如果再稍微多付出一点努力，就能更快更全面地

了解 Scala。

首先,你可以充分利用本书中包含的许多编程例子。尝试自己输入能强迫你的大脑思考每一行代码。而尝试各种各样的变化能让它们变得更有趣也能让你确信已真正明白它们如何工作。

其次,与多个在线论坛保持联系。采用这种方式,你和其他 Scala 爱好者就能够互相帮助。你还可以访问更多的电子邮件列表、讨论论坛、聊天室、维基百科和多个特别为 Scala 准备的文档资料更新点。花些时间来查找包含你需要的信息的地方。这样,花更少的时间在小问题上,就能花更多的时间在更深入人和更重要的地方。

最后,一旦你已经掌握了部分内容,请把它用在你自己的编程项目上。从草案开始开发一个小程序,或大一点儿程序的附加部分。仅仅看书不能学到更多东西。

印刷体变化

首次使用的术语 (term), 将使用括号演示原来的英文词语。小代码例子, 如 `x+1`, 将用等宽字体演示在文档段落中。大段的代码例子将放在等宽字体的段落中演示:

```
def hello() {
  println("Hello, world!")
}
```

在演示交互式 shell 的时候, shell 的回应将演示为深红色字体。

```
scala> 3 + 4
res0: Int = 7
```

内容概要

- 第 1 章, “可伸展的语言”, 给出了 Scala 的设计, 和它后面的理由, 历史的概要。
- 第 2 章, “Scala 的入门初探”, 展示给你如何使用 Scala 完成若干种基本编程任务, 而不牵涉过多关于如何工作的细节。本章的目的是让你的手指开始敲击并执行 Scala 代码。
- 第 3 章, “Scala 的入门再探”, 演示更多的基本编程任务来帮助你更快速地上手 Scala。本章之后, 你将能够开始在简单的脚本任务中使用 Scala。
- 第 4 章, “类和对象”, 通过描述面向对象语言的基本建设模块和如何编译及运行 Scala 程序的教程开始有深度地覆盖 Scala 语言。
- 第 5 章, “基本类型和操作”, 覆盖了 Scala 的基本类型, 它们的字面量, 可执行的操作, 优先级和关联性是如何工作的, 还有什么是富包装器。
- 第 6 章, “函数式对象”, 进入了 Scala 面向对象特征的更深层次, 使用函数式 (即不可变) 有理数作为例子。

- 第 7 章，“内建控制结构”，演示了如何使用 Scala 的内建控制结构，如 `if`, `while`, `for`, `try` 和 `match`。
- 第 8 章，“函数和闭包”，深度讨论了函数式语言的基础建设模块，函数。
- 第 9 章，“控制抽象”，演示了如何通过定义你自己的控制抽象来增强 Scala 的基本控制结构。
- 第 10 章，“组合与继承”，讨论了更多 Scala 对面向对象编程的支持。这个话题并不像在第 4 章中那样基础，但它们在实践中经常出现。
- 第 11 章，“Scala 的层级”，解释了 Scala 的继承层级并讨论了其全体方法及底层类型。
- 第 12 章，“特质”（`trait`），演示了 Scala 在混入组成（`mixin composition`）中的机制。本章演示了特质如何工作，描述了通常的用法，还解释了为什么特质改善了传统的多继承。
- 第 13 章，“包和引用”，讨论了大项目编程中的事务，包括顶层包，引用语句，还有访问控制修饰符如，`protected` 和 `private`。
- 第 14 章，“断言和单元测试”，演示了 Scala 的断言机制并大致学习了各种可以为 Scala 编写测试的工具。
- 第 15 章，“样本类和模式匹配”，介绍了样本类和模式匹配，这对你在编写正规的非封装的数据结构时用到的工具，尤其对树型递归数据很有用。
- 第 16 章，“使用列表”，详细解释了列表。它或许是在 Scala 程序中最常用到的数据结构。
- 第 17 章，“集合类型”，演示了如何使用基础的 Scala 集合类型，如：列表、数组、元组集（`tuple`）及映射表。
- 第 18 章，“有状态的对象”，解释了什么是有状态（即可变）的对象，Scala 提供的语法层面表达它们的术语。本章包括了一个在离散事件模拟上的案例研究，用来演示一些有状态对象的动作。
- 第 19 章，“类型参数化”，用具体的例子：纯函数队列类的设计，解释了第 13 章介绍过的一些信息隐藏技术。本章建立了关于各种类型参数的描述，以及它如何与信息隐藏实现交互。
- 第 20 章，“抽象成员”，描述了所有 Scala 支持的抽象成员。能够声明为抽象的不仅是方法，还包括字段和类型。
- 第 21 章，“隐式转换和参数”，描述了这两个特性有助于程序员忽略掉源码中那些能由编译器推导出来的繁琐的细节的特性。
- 第 22 章，“实现列表”，描述了 `List` 类的实现。弄明白在 Scala 里面列表如何工作是很重要的，而且，实现本身展示了若干 Scala 特性的应用。
- 第 23 章，“重访 `For` 表达式”，解释了 `for` 表达式是如何翻译成对 `map`、`flatMap`、`filter`

和 `foreach` 的访问。

- 第 24 章 (Extractors), “抽取器”, 展示了如何使用模式匹配任何类, 而不仅仅是用例类。
- 第 25 章, “注解”, 演示了如何通过注解使用语言的扩展部分。本章示范了若干标准注解, 也示范了如何建立你自己的注解。
- 第 26 章, “使用 XML”, 演示了在 Scala 中如何处理 XML。包括 XML 的创建、解析, 以及解析之后的处理等一系列惯用方式。
- 第 27 章, “使用对象的模块化编程”, 演示了说 Scala 的对象已足够丰富的原因, 从而消除了分离式模块系统的使用需求。
- 第 28 章, “对象相等性”, 指出若干在编写 `equals` 方法时要考虑的情况。说明了若干应避免的误区。
- 第 29 章, “结合 Scala 和 Java”, 描述了若干在同一个项目中捆绑使用 Java 和 Scala 时会碰到的状况, 以及建议的解决方法。
- 第 30 章, “Actor 和并发”, 展示如何使用 Scala 的 actor 并发库。尽管你使用 Java 平台的同步原语和来自于 Scala 程序的库, 但 actor 能帮你避免死锁和资源竞争这些影响着传统并发的的问题。
- 第 31 章, “连结符解析”, 演示了如何使用 Scala 的解析器连结符库来创建解析器。
- 第 32 章, “GUI 编程”, 展示了使用 Scala 库简化基于 Swing 的 GUI 编程的快速旅程。
- 第 33 章, “SCells 试算表”, 通过展示一个完整的试算表的实现, 集中演示了 Scala 的一切。

资源

在 Scala 的主网站, <http://www.scala-lang.org>, 你能找到 Scala 最近的发布版和文档、社区资源的链接。Scala 资源链接更全的页面, 请访问本书网站: http://booksites.artima.com/programming_in_scala。与本书其他读者交流, 请访问: <http://www.artima.com/forums/forum.jsp?forum=282>。

源码

你可以从本书的网站下载包含本书源码的 ZIP 文件, 它是以 Apache 2.0 开源许可方式发布的: http://booksites.artima.com/programming_in_scala。

勘误

尽管本书已复审检查多次, 仍不可避免错误的发生。要查阅本书的勘误列表, 请访问: http://booksites.artima.com/programming_in_scala/errata。如果你发现错误, 也请在上述网址报告, 这样我们可以确保在本书将来的印刷或发行版中修正它。

目录概览

目录	ix
图示清单	xvii
表格清单	xix
代码清单	xxi
序	I
致谢	III
简介	V
第 1 章 可伸展的语言	3
第 2 章 Scala 入门初探	15
第 3 章 Scala 入门再探	23
第 4 章 类和对象	37
第 5 章 基本类型和操作	47
第 6 章 函数式对象	61
第 7 章 内建控制结构	73
第 8 章 函数和闭包	89
第 9 章 控制抽象	103
第 10 章 组合与继承	113
第 11 章 Scala 的层级	131
第 12 章 特质	137
第 13 章 包和引用	151
第 14 章 断言和单元测试	161
第 15 章 样本类和模式匹配	171
第 16 章 使用列表	193
第 17 章 集合类型	215
第 18 章 有状态的对象	233
第 19 章 类型参数化	249
第 20 章 抽象成员	265
第 21 章 隐式转换和参数	285
第 22 章 实现列表	301
第 23 章 重访 For 表达式	309
第 24 章 抽取器 (Extractors)	321

第 25 章 注解 331

第 26 章 使用 XML 335

第 27 章 使用对象的模块化编程 345

第 28 章 对象相等性 355

第 29 章 结合 Scala 和 Java 373

第 30 章 Actor 和并发 383

第 31 章 连结符解析 407

第 32 章 GUI 编程 427

第 33 章 Scell 试算表 435

附录 A Unix 和 Windows 的 Scala 脚本 453

术语表 455

参考文献 465

关于作者 467

索引 469



目录

目录	ix
图示清单	xvii
表格清单	xix
代码清单	xxi
序	I
致谢	III
简介	V
第 1 章 可伸展的语言	3
1.1 与你一同成长的语言	3
1.2 是什么让 Scala 具有可扩展性?	6
1.3 为什么选择 Scala?	8
1.4 Scala 的根源	13
1.5 小结	14
第 2 章 Scala 入门初探	15
2.1 第一步 学习使用 Scala 解释器	15
2.2 第二步 变量定义	16
2.3 第三步 函数定义	18
2.4 第四步 编写 Scala 脚本	19
2.5 第五步 用 while 做循环; 用 if 做判断	20
2.6 第六步 用 foreach 和 for 做枚举	21
2.7 小结	22
第 3 章 Scala 入门再探	23
3.1 第七步 使用类型参数化数组 (Array)	23
3.2 第八步 使用列表 (List)	25
3.3 第九步 使用元组 (Tuple)	28
3.4 第十步 使用集 (set) 和映射 (map)	29
3.5 第十一步 学习识别函数式风格	32
3.6 第十二步 从文件里读取文本行	34
3.7 小结	36
第 4 章 类和对象	37
4.1 类、字段和方法	37

- 4.2 分号推断.....40
- 4.3 Singleton 对象.....41
- 4.4 Scala 程序.....43
- 4.5 Application 特质.....45
- 4.6 小结.....45
- 第 5 章 基本类型和操作.....47
 - 5.1 基本类型.....47
 - 5.2 字面量.....48
 - 5.3 操作符和方法.....52
 - 5.4 数学运算.....54
 - 5.5 关系和逻辑操作.....55
 - 5.6 位操作符.....56
 - 5.7 对象相等性.....57
 - 5.8 操作符的优先级和关联性.....58
 - 5.9 富包装器.....60
 - 5.10 小结.....60
- 第 6 章 函数式对象.....61
 - 6.1 类 Rational 的规格说明书.....61
 - 6.2 创建 Rational.....62
 - 6.3 重新实现 toString 方法.....63
 - 6.4 检查先决条件.....63
 - 6.5 添加字段.....64
 - 6.6 自指向.....65
 - 6.7 辅助构造器.....65
 - 6.8 私有字段和方法.....66
 - 6.9 定义操作符.....67
 - 6.10 Scala 的标识符.....68
 - 6.11 方法重载.....70
 - 6.12 隐式转换.....71
 - 6.13 一番告诫.....72
 - 6.14 小结.....72
- 第 7 章 内建控制结构.....73
 - 7.1 If 表达式.....73
 - 7.2 While 循环.....74
 - 7.3 for 表达式.....76
 - 7.4 使用 try 表达式处理异常.....80
 - 7.5 匹配 (match) 表达式.....82
 - 7.6 不再使用 break 和 continue.....83
 - 7.7 变量范围.....84

7.8	重构指令式风格的代码.....	87
7.9	小结.....	88
第 8 章	函数和闭包.....	89
8.1	方法.....	89
8.2	本地函数.....	90
8.3	头等函数.....	91
8.4	函数字面量的短格式.....	93
8.5	占位符语法.....	93
8.6	部分应用函数.....	94
8.7	闭包.....	96
8.8	重复参数.....	98
8.9	尾递归.....	99
8.10	小结.....	102
第 9 章	控制抽象.....	103
9.1	减少代码重复.....	103
9.2	简化客户代码.....	106
9.3	柯里化 (currying)	107
9.4	编写新的控制结构.....	108
9.5	传名参数 (by-name parameter)	110
9.6	小结.....	112
第 10 章	组合与继承.....	113
10.1	二维布局库.....	113
10.2	抽象类.....	114
10.3	定义无参数方法.....	114
10.4	扩展类.....	116
10.5	重写方法和字段.....	117
10.6	定义参数化字段.....	118
10.7	调用超类构造器.....	119
10.8	使用 override 修饰符.....	120
10.9	多态和动态绑定.....	121
10.10	定义 final 成员.....	123
10.11	使用组合与继承.....	124
10.12	实现 above、beside 和 toString	124
10.13	定义工厂对象.....	126
10.14	变高变宽.....	128
10.15	把代码都放在一起.....	129
10.16	小结.....	130
第 11 章	Scala 的层级.....	131
11.1	Scala 的类层级.....	131

11.2	原始类型是如何实现的	134
11.3	底层类型	135
11.4	小结	136
第 12 章	特质	137
12.1	特质是如何工作的	137
12.2	瘦接口对阵胖接口	139
12.3	样例：长方形对象	140
12.4	Ordered 特质	141
12.5	特质用来做可堆叠的改变	143
12.6	为什么不是多重继承？	146
12.7	特质，用还是不用？	148
12.8	小结	149
第 13 章	包和引用	151
13.1	包	151
13.2	引用	153
13.3	隐式引用	156
13.4	访问修饰符	156
13.5	小结	160
第 14 章	断言和单元测试	161
14.1	断言	161
14.2	Scala 里的单元测试	162
14.3	翔实的失败报告	163
14.4	使用 JUnit 和 TestNG	164
14.5	规格测试	166
14.6	基于属性的测试	167
14.7	组织和运行测试	168
14.8	小结	170
第 15 章	样本类和模式匹配	171
15.1	简单例子	171
15.2	模式的种类	174
15.3	模式守卫	180
15.4	模式重叠	181
15.5	封闭类	182
15.6	Option 类型	183
15.7	模式无处不在	184
15.8	一个更大的例子	187
15.9	小结	192
第 16 章	使用列表	193
16.1	列表字面量	193

16.2	List 类型	193
16.3	构造列表	194
16.4	列表的基本操作	194
16.5	列表模式	195
16.6	List 类的一阶方法	196
16.7	List 类的高阶方法	204
16.8	List 对象的方法	210
16.9	了解 Scala 的类型推断算法	212
16.10	小结	214
第 17 章	集合类型	215
17.1	集合库概览	215
17.2	序列	216
17.3	集 (Set) 和映射 (Map)	220
17.4	可变 (mutable) 集合 vs. 不可变 (immutable) 集合	227
17.5	初始化集合	229
17.6	元组	231
17.7	小结	232
第 18 章	有状态的对象	233
18.1	什么让对象具有状态?	233
18.2	可重新赋值的变量和属性	234
18.3	案例研究: 离散事件模拟	237
18.4	为数字电路定制的语言	237
18.5	Simulation API	239
18.6	电路模拟	242
18.7	小结	247
第 19 章	类型参数化	249
19.1	queues 函数式队列	249
19.2	信息隐藏	251
19.3	变化型注解	253
19.4	检查变化型注解	256
19.5	下界	258
19.6	逆变	259
19.7	对象私有数据	261
19.8	上界	263
19.9	小结	264
第 20 章	抽象成员	265
20.1	抽象成员的快速浏览	265
20.2	类型成员	266
20.3	抽象 val	266

20.4	抽象 var	267
20.5	初始化抽象 val	267
20.6	抽象类型	273
20.7	路径依赖类型	274
20.8	枚举	276
20.9	案例研究：货币	277
20.10	小结	284
第 21 章	隐式转换和参数	285
21.1	隐式转换	285
21.2	隐式操作规则	287
21.3	隐式转换为期望类型	289
21.4	转换（方法调用的）接收者	290
21.5	隐式参数	292
21.6	视界	296
21.7	隐式操作调试	297
21.8	小结	299
第 22 章	实现列表	301
22.1	List 类原理	301
22.2	ListBuffer 类	305
22.3	实际的 List 类	306
22.4	外在的函数式（风格）	308
22.5	小结	308
第 23 章	重访 For 表达式	309
23.1	For 表达式	310
23.2	皇后问题	311
23.3	使用 for 表达式做查询	313
23.4	for 表达式的转译	314
23.5	反其道而行之	317
23.6	泛化的 for	318
23.7	小结	319
第 24 章	抽取器（Extractors）	321
24.1	例子：抽取 email 地址	321
24.2	抽取器	322
24.3	0 或 1 个变量的模式	324
24.4	变参抽取器	325
24.5	抽取器和序列模式	327
24.6	抽取器 VS. 样本类	327
24.7	正则表达式	328
24.8	小结	330

第 25 章 注解331

 25.1 为什么要有注解?331

 25.2 注解语法332

 25.3 标准注解333

 25.4 小结334

第 26 章 使用 XML335

 26.1 半结构化数据335

 26.2 XML 概览335

 26.3 XML 字面量336

 26.4 序列化338

 26.5 拆解 XML339

 26.6 反序列化340

 26.7 加载和保存341

 26.8 XML 的模式匹配342

 26.9 小结344

第 27 章 使用对象的模块化编程345

 27.1 问题345

 27.2 食谱应用346

 27.3 抽象概念348

 27.4 把模块拆分为特质350

 27.5 运行期链接352

 27.6 跟踪模块实例353

 27.7 小结354

第 28 章 对象相等性355

 28.1 Scala 中的相等性355

 28.2 编写相等性方法355

 28.3 定义带参数类型的相等性365

 28.4 equals 和 hashCode 的制作方法368

 28.5 小结371

第 29 章 结合 Scala 和 Java373

 29.1 在 Java 中使用 Scala373

 29.2 注解375

 29.3 存在类型379

 29.4 小结381

第 30 章 Actor 和并发383

 30.1 天堂中的烦恼383

 30.2 actor 和消息传递384

 30.3 将原生线程当作 actor387

 30.4 通过重用线程获得更好的性能387

- 30.5 良好的 actor 风格389
- 30.6 更长一些的示例：并行离散事件模拟394
- 30.7 小结406
- 第 31 章 连结符解析407
 - 31.1 示例：算术表达式408
 - 31.2 运行你的解析器409
 - 31.3 基本的正则表达式解析器410
 - 31.4 另一个示例：JSON410
 - 31.5 解析器输出412
 - 31.6 实现连结符解析器416
 - 31.7 字符串字面量和正则表达式421
 - 31.8 词法分析和解析422
 - 31.9 错误报告423
 - 31.10 回溯 vs. LL(1)424
 - 31.11 小结425
- 第 32 章 GUI 编程427
 - 32.1 第一个 Swing 应用427
 - 32.2 面板和布局429
 - 32.3 处理事件430
 - 32.4 示例：摄氏/华氏温度转换器432
 - 32.5 小结434
- 第 33 章 Scell 试算表435
 - 33.1 可视化框架435
 - 33.2 将数据录入和显示分开437
 - 33.3 公式439
 - 33.4 解析公式440
 - 33.5 求值444
 - 33.6 操作库446
 - 33.7 修改传达448
 - 33.8 小结451
- 附录 A Unix 和 Windows 的 Scala 脚本453
- 术语表455
- 参考文献465
- 关于作者467
- 索引469

图示清单

图 2.1	Scala 函数的基本构成	18
图 2.2	Scala 函数字面量的语法	22
图 3.1	Scala 里所有的操作符都是方法调用	24
图 3.2	Scala 的 Set 类层级	30
图 3.3	Scala 的 Map 类层级	31
图 10.1	ArrayElement 的类关系图	116
图 10.2	LineElement 的类关系图	120
图 10.3	布局元素的类层级	121
图 10.4	修改了 LineElement 后的类层级	124
图 11.1	Scala 类层级图	132
图 12.1	Cat 类的继承层级和线性化次序	147
图 14.1	ScalaTest 的图形报表器	169
图 17.1	Scala 集合类型的类层级	215
图 17.2	Iterator 的类层级	216
图 18.1	基本门电路	237
图 18.2	半加法电路	238
图 18.3	全加法电路	239
图 19.1	函数类型参数中的协变与逆变	261
图 22.1	Scala 列表的类层级	302
图 22.2	展示的 Scala 列表构造	304
图 32.1	一个简单的 Swing 应用程序：初始状态（左）和拉大之后的状态（右）	428
图 32.2	一个有反应的 Swing 应用程序：初始状态（左）及点击后的状态（右）	429

图 32.3 摄氏度与华氏度之间的转换器432

图 33.1 一个简单的试算表表格436

图 33.2 显示自身的单元格439

图 33.3 显示方程式的单元格444

图 33.4 可以求值的单元格448



表格清单

表 3.1 List 的一些方法和作用27

表 5.1 一些基本类型47

表 5.2 特殊字符字面量转义序列50

表 5.3 操作符优先级59

表 5.4 一些富操作60

表 5.5 富包装类60

表 12.1 Cat 层级中类型的线性化148

表 13.1 LegOfJourney.distance 上的私有限定符效果159

表 16.1 基本列表操作195

表 17.1 集的常用操作222

表 17.2 映射的常用操作223

表 17.3 默认的不可变集实现225

表 17.4 默认的不可变映射实现225

表 31.1 解析器连结符汇总415



代码清单

清单 3.1	用类型参数化数组	23
清单 3.2	创造和初始化数组	25
清单 3.3	创造和初始化列表	26
清单 3.4	创造和使用元组	28
清单 3.5	创造、初始化和使用不可变集	29
清单 3.6	创建、初始化和使用可变集	30
清单 3.7	创造、初始化和使用可变映射	31
清单 3.8	创造、初始化和使用不可变映射	32
清单 3.9	没有副作用或 var 的函数	33
清单 3.10	从文件中读入行	34
清单 3.11	对文件的每行记录打印格式化的字符数量	36
清单 4.1	类 ChecksumAccumulator 的最终版	40
清单 4.2	类 ChecksumAccumulator 的伴生对象	42
清单 4.3	程序 Summer	43
清单 4.4	使用 Application 特质	45
清单 6.1	带字段的 Rational	64
清单 6.2	带有从构造器的 Rational	66
清单 6.3	带私有字段和方法的 Rational	67
清单 6.4	带操作符方法的 Rational	68
清单 6.5	含有重载方法的 Rational	71
清单 7.1	在 Scala 里根据条件做初始化的惯例	74
清单 7.2	用 while 循环计算最大公约数	74
清单 7.3	用 do-while 从标准输入读取信息	75
清单 7.4	使用递归计算最大公约数	75
清单 7.5	用 for 表达式列举目录中的文件	76
清单 7.6	用带过滤器的 for 发现 .scala 文件	77
清单 7.7	在 for 表达式中使用多个过滤器	78
清单 7.8	在 for 表达式中使用多个发生器	78
清单 7.9	在 for 表达式里的流间赋值	78
清单 7.10	用 for 把 Array[File] 转换为 Array[Int]	79
清单 7.11	Scala 的 try-catch 子句	80
清单 7.12	Scala 的 try-finally 子句	81

清单 7.13 能够产生值的 catch 子句.....82

清单 7.14 有副作用的 match 表达式.....82

清单 7.15 生成值的 match 表达式.....83

清单 7.16 不带 break 或 continue 的循环.....84

清单 7.17 不用 var 做循环的递归替代方法.....84

清单 7.18 打印乘法表时的变量范围.....85

清单 7.19 创建乘法表的函数式方法.....87

清单 8.1 带私有的 processLine 方法的 LongLines 对象.....89

清单 8.2 带局部 processLine 方法的 LongLines.....91

清单 9.1 使用闭包减少代码重复.....105

清单 9.2 定义和调用“陈旧的”函数.....107

清单 9.3 定义和调用柯里化的函数.....107

清单 9.4 使用借贷模式写文件.....110

清单 9.5 使用传名参数.....111

清单 10.1 定义抽象方法和类.....114

清单 10.2 定义无参数方法 width 和 height.....114

清单 10.3 定义 ArrayElement 为 Element 的子类.....116

清单 10.4 用字段重写无参数方法.....117

清单 10.5 定义 contents 为参数化字段.....118

清单 10.6 调用超类构造器.....119

清单 10.7 声明 final 方法.....123

清单 10.8 声明 final 类.....123

清单 10.9 带有 above、beside 和 toString 的类 Element.....126

清单 10.10 带有工厂方法的工厂对象.....127

清单 10.11 重构以使用工厂方法的类 Element.....127

清单 10.12 用私有类隐藏实现.....128

清单 10.13 有了 widen 和 heighten 方法的 Element.....129

清单 10.14 Spiral 程序.....130

清单 12.1 Philosophical 特质的定义.....137

清单 12.2 使用 extends 混入特质.....137

清单 12.3 使用 with 混入特质.....138

清单 12.4 混入多个特质.....138

清单 12.5 定义丰满了的特质.....141

清单 12.6 抽象类 IntQueue.....143

清单 12.7 使用 ArrayBuffer 实现 BasicIntQueue.....143

清单 12.8 Doubling 可堆叠改动特质.....144

清单 12.9 在使用 new 实例化的时候混入特质.....145

清单 12.10 可堆叠改动特质 Incrementing 和 Filtering.....145

清单 13.1 把文件的全部内容放进包里.....151

清单 13.2	同一个文件嵌入不同的包	152
清单 13.3	较少缩进的嵌入包	152
清单 13.4	Scala 的包确实是嵌套的	152
清单 13.5	访问隐藏的包名	153
清单 13.6	鲍勃最爱的水果，已为引用做好准备	154
清单 13.7	引用规范的（不是单例）对象的成员	154
清单 13.8	引用包名	155
清单 13.9	Scala 和 Java 的 private 访问差异	157
清单 13.10	Scala 和 Java 的 protected 访问差异	157
清单 13.11	使用访问修饰词的灵活的保护作用域	158
清单 13.12	访问伴生类和对象的私有成员	160
清单 14.1	使用断言	161
清单 14.2	使用 ensuring 断言函数结果	162
清单 14.3	使用 Suite 编写测试方法	162
清单 14.4	使用 FunSuite 编写测试函数	163
清单 14.5	使用 JUnit3Suite 编写 JUnit 测试	165
清单 14.6	使用 TestNGSuite 编写 TestNG 测试	165
清单 14.7	使用 ScalaTest 的 Spec 规格化和测试行为	166
清单 14.8	使用 specs 架构规格化和测试行为	167
清单 14.9	使用 ScalaCheck 编写基于属性的测试	167
清单 14.10	使用 Checkers 在 JUnit 的 TestCase 里检查属性	168
清单 15.1	定义样本类	171
清单 15.2	simplifyTop 函数，使用了模式匹配	173
清单 15.3	带有空“默认”情况的模式匹配	174
清单 15.4	使用通配模式的模式匹配	174
清单 15.5	使用常量模式的模式匹配	175
清单 15.6	使用变量模式的模式匹配	175
清单 15.7	带有构造器模式的模式匹配	176
清单 15.8	有固定长度的序列模式	177
清单 15.9	匹配任意长度的序列模式	177
清单 15.10	带有元组模式的模式匹配	177
清单 15.11	带有类型化模式的模式匹配	177
清单 15.12	使用 isInstanceOf 和 asInstanceOf（不好的风格）	178
清单 15.13	带有变量绑定（用@符号）的模式	180
清单 15.14	带有模式守卫的 match 表达式	180
清单 15.15	样本依次被调用的匹配表达式	181
清单 15.16	封闭了的样本类层级	182
清单 15.17	用一个赋值语句定义多个变量	184
清单 15.18	带有元组模式的 for 表达式	186
清单 15.19	从列表中拣出匹配于模式的元素	187
清单 15.20	表达式格式化器的上半截	189

清单 15.21 表达式格式化器的下半截.....190

清单 15.22 输出格式化了了的表达式的应用程序.....191

清单 16.1 List 类的归并排序.....203

清单 17.1 Predef 中定义的默认映射和集.....221

清单 17.2 混入 SynchronizedMap 特质.....226

清单 18.1 可变的银行账号类.....233

清单 18.2 带有公开 var 变量的类.....235

清单 18.3 公开的 var 变量是如何扩展为 getter 和 setter 方法的.....235

清单 18.4 直接定义 getter 和 setter 方法.....236

清单 18.5 定义不带关联字段的 getter 和 setter.....236

清单 18.6 halfAdder 方法.....239

清单 18.7 fullAdder 方法.....239

清单 18.8 Simulation 类.....240

清单 18.9 BasicCircuitSimulation 类的前半部分.....243

清单 18.10 BasicCircuitSimulation 类的后半部分.....244

清单 18.11 CircuitSimulation 类.....246

清单 19.1 基本的函数式队列.....251

清单 19.2 通过私有化隐藏主构造器.....251

清单 19.3 伴生对象的 apply 工厂方法.....252

清单 19.4 函数式队列的类型抽象.....253

清单 19.5 非协变（严谨）的 Cell 类.....254

清单 19.6 带有下界的类型参数.....258

清单 19.7 逆变的输出通道.....259

清单 19.8 Function1 的协变和逆变.....260

清单 19.9 函数类型参数变化型的演示.....261

清单 19.10 优化了的函数式队列.....262

清单 19.11 混入了 Ordered 特质的 Person 类.....263

清单 19.12 带有上界的归并排序.....264

清单 20.1 重写的抽象 val 及无参方法.....267

清单 20.2 声明抽象 var.....267

清单 20.3 抽象 var 是如何被扩展为 getter 和 setter 方法的.....267

清单 20.4 使用了抽象 val 的特质.....268

清单 20.5 匿名类表达式中的预初始化字段.....269

清单 20.6 对象定义中的预初始化字段.....270

清单 20.7 类定义中的预初始化字段.....270

清单 20.8 使用懒加载 val 初始化特质.....271

清单 20.9 用抽象类型建模合适的食物.....274

清单 20.10 在子类中实现抽象类型.....274

清单 20.11 美国货币区域.....281

清单 20.12	欧洲和日本的货币区域.....	282
清单 20.13	带有汇率表的转换器对象.....	282
清单 20.14	CurrencyZone 的完整代码.....	283
清单 21.1	带多个参数的隐式参数列表.....	293
清单 21.2	带有上界的函数.....	294
清单 21.3	带有隐式参数的函数.....	295
清单 21.4	使用了内部隐式参数的函数.....	296
清单 21.5	带有视界的函数.....	297
清单 21.6	使用隐式参数的样例代码.....	299
清单 21.7	完成类型检查和隐式内容插入之后的样例代码.....	299
清单 22.1	Nil 单例对象的定义.....	302
清单 22.2	在子类型列表上前缀超类型元素.....	304
清单 22.3	List 类的:: (cons) 方法定义.....	304
清单 22.4	List 类的::: 方法定义.....	305
清单 22.5	List 类的 map 方法定义.....	306
清单 22.6	List 子类:: 的定义.....	307
清单 24.1	EMail 字符串抽取器对象.....	322
清单 24.2	Twice 字符串抽取器对象.....	324
清单 24.3	UpperCase 字符串抽取器对象.....	324
清单 24.4	Domain 字符串抽取器对象.....	325
清单 24.5	ExpandedEMail 抽取器对象.....	326
清单 24.6	定义了 unapplySeq 方法的抽取器.....	327
清单 24.7	RichString 是如何定义 r 方法的.....	329
清单 27.1	简化的 Food 实体类.....	346
清单 27.2	简化的 Recipe 实体类.....	347
清单 27.3	可以用于测试的 Food 和 Recipe 例子.....	347
清单 27.4	仿制的数据库和浏览器模块.....	347
清单 27.5	添加了类别的数据库和浏览器模块.....	348
清单 27.6	带抽象数据库 val 字段的 Browser 类.....	349
清单 27.7	带抽象方法的 Database 类.....	349
清单 27.8	Database 的子类对象 SimpleDatabase.....	350
清单 27.9	Browser 子类对象 SimpleBrowser.....	350
清单 27.10	学生数据库与浏览器.....	350
清单 27.11	食品分类的特质.....	351
清单 27.12	混入了 FoodCategories 特质的 Database.....	351
清单 27.13	仅由混入组成的 SimpleDatabase 对象.....	351
清单 27.14	SimpleFoods 特质.....	351
清单 27.15	带有自身类型的 SimpleRecipes.....	352
清单 27.17	使用单例类型.....	354

清单 28.1 调用 canEqual 的超类 equals 方法363

清单 28.2 调用 canEqual 的子类 equals 方法364

清单 28.3 二叉树的类继承关系.....365

清单 28.4 带有 equals 和 hashCode 方法的参数化类型368

清单 28.5 带有 equals 和 hashCode 的 Rational 类369

清单 29.1 声明了 Java throws 语句的 Scala 方法376

清单 30.1 一个简单的 actor.....384

清单 30.2 调用 receive 的 actor386

清单 30.3 调用 react 的 actor389

清单 30.4 使用 loop 的 actor 方法 act389

清单 30.5 使用助手 actor 来避免阻塞自己的 actor391

清单 30.6 使用样本类作为消息的 actor394

清单 30.7 Simulant 特质.....399

清单 30.8 加法器组件404

清单 31.1 一个算术表达式解析器.....408

清单 31.2 针对 Java 标识符的正则表达式解析器.....410

清单 31.3 JSON 格式的数据411

清单 31.4 一个简单的 JSON 解析器411

清单 31.5 返回有意义的结果的完整 JSON 解析器414

清单 31.6 ~ 连结符方法.....419

清单 32.1 一个用 Scala 编写的简单 Swing 应用程序427

清单 32.2 在面板上组装组件429

清单 32.3 实现一个有反应的 Swing 应用程序.....432

清单 32.4 温度转换器的一个实现.....433

清单 33.1 图 33.1 中的试算表的代码436

清单 33.2 试算表应用程序的主程序437

清单 33.3 带有 rendererComponent 方法的试算表438

清单 33.4 第一版的 Model 类.....438

清单 33.5 代表公式的类.....440

清单 33.6 解析公式的试算表.....443

清单 33.7 Evaluator 特质的 evaluate 方法.....445

清单 33.8 一个算术操作库.....446

清单 33.9 完成的试算表组件.....451