

版权注意事项：

- 1、书籍版权归作者和出版社所有
- 2、本PDF仅限用于个人获取知识，进行私底下的知识交流
- 3、PDF获得者不得在互联网上以任何目的进行传播
- 4、如觉得书籍内容很赞，请购买正版实体书，支持作者
- 5、请于下载PDF后24小时内删除本PDF。

artima

异步图书
www.epubit.com.cn

深入理解 Scala 的有趣途径

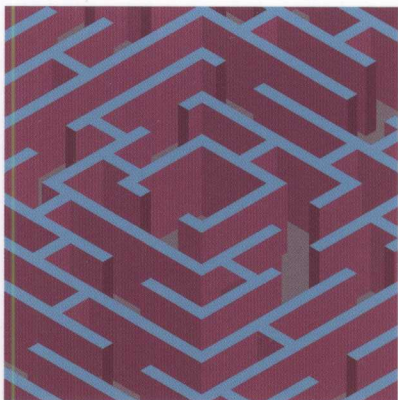
Scala 谜题


[美] Andrew Phillips Nermin Šerifović 著

包春霞 冷钰冰 译

SO
CALA
A

PUZZLERS



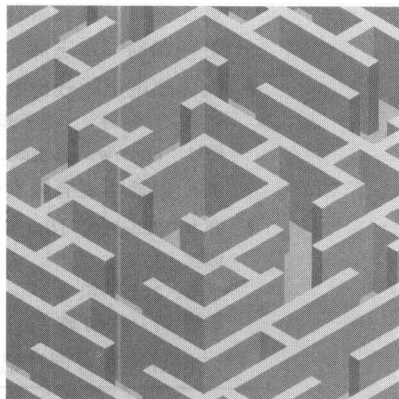
 中国工信出版集团

 人民邮电出版社
POSTS & TELECOM PRESS

Scala 谜题

[美] Andrew Phillips Nermin Šerifović 著

包春霞 冷钰冰 译



人民邮电出版社

北京

图书在版编目 (C I P) 数据

Scala谜题 / (美) 菲利普斯 (Andrew Phillips),
(美) 萨尔法维克著; 包春霞, 冷钰冰译. — 北京: 人
民邮电出版社, 2017. 11
ISBN 978-7-115-46007-3

I. ①S… II. ①菲… ②萨… ③包… ④冷… III. ①
JAVA语言—程序设计 IV. ①TP312.8

中国版本图书馆CIP数据核字(2017)第190917号

版权声明

Simplified Chinese translation copyright ©2017 by Posts and Telecommunications Press
ALL RIGHTS RESERVED

Scala Puzzlers by Andrew Phillips and Nermin Šerifović ISBN 9780981531670

Copyright © 2016 by Artima, Inc.

本书中文简体版由 **Artima, Inc.** 授权人民邮电出版社出版。未经出版者书面许可, 对本书的
任何部分不得以任何方式或任何手段复制和传播。

版权所有, 侵权必究。

-
- ◆ 著 [美] Andrew Phillips Nermin Šerifović
译 包春霞 冷钰冰
责任编辑 陈冀康
责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
大厂聚鑫印刷有限责任公司印刷
 - ◆ 开本: 720×960 1/16
印张: 12.75
字数: 225 千字 2017 年 11 月第 1 版
印数: 1-2 400 册 2017 年 11 月河北第 1 次印刷
- 著作权合同登记号 图字: 01-2016-0525 号
-

定价: 49.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

内容提要

Scala 是一种多范式的编程语言,其设计初衷是要整合面向对象编程和函数式编程的各种特性。

本书整合了众多典型的 Scala 代码示例,深入解密 Scala。书中不仅介绍了 Scala 语言,还介绍了编译器。本书通过有趣的方式带领读者学习并深入理解和掌握 Scala。全书共有 36 个谜题,每一个谜题都可以丰富读者的知识,并能够让读者更深入地了解 Scala。

本书适合于对 Scala 感兴趣的开发者、对 JVM 平台上的语言以及函数式编程感兴趣的程序员阅读。

序 言

谜题很有趣。我还记得第一次遇到 Java 语言的“谜题”的情景。Neal Gafter 给我看了他和 Josh Bloch 刚收集的一道 Java 谜题。那时候，Neal 正在维护我编写的 javac 编译器。这道题太难了，我猜错了几次，让 Neal 乐坏了。

很高兴现在有一本书能在 Scala 中延续谜题的传统。本书将为读者展示 36 个谜题，之所以称为谜题，是因为它们会产生令人意外的结果，有特性间的交互作用或者有不同于表面的代码运行结果。这些谜题是从 Scala 社区几年来广泛的输入中收集而来的。

Andrew 和 Nermin 提炼出每个谜题的本质，使其易于理解。在享受了从一系列选项中选出正确答案的乐趣之后，每道题后面会告诉你为什么会这样，什么原因导致了这个令人意外的结果？这正是本书真正的亮点，因为它对观察到的程序行为的深层原理进行了清晰地讲解。我特别喜欢本书的一点是这些讲解总是给你带来全新的视角。它们不仅告诉你令人意外的行为的趣事，而且集 Scala 深入讲解于一体。这样，谜题就会有助于你更深入地理解这门语言。

希望你能像我一样觉得阅读本书并试着解决这些谜题很有趣。是的，有些谜题我还不能解决。

Martin Odersky

Scala 之父

2014.5.26

自序

当开始学习一门新的编程语言时，最初困扰你的只是缺少这门语言的知识。随着你对这门语言的经验和能力的增长，让你头疼的时刻就逐渐变成了系统层面的问题，例如，依赖冲突或很难重现的竞争条件。让人真正喜欢 Josh Bloch 和 Neal Gafter 的 Java 谜题的一点是：它让我们在日常工作和最后期限之外参与 Java。谜题不仅触发了我们解决问题的欲望，而且帮助我们纠正并加深了对这门语言的理解。

开始学习 Scala 不久，我们就认识到收集一套类似的谜题可能会真的很有用——由具有时代精神的社区来建立并维护这些谜题是再好不过的了。因而，Scala 谜题网站 scalapuzzlers.com 就诞生了。

受所收到的积极反馈的鼓励，我们决定将这些谜题收集并扩展成本书。我们的目标是：更加详细地解释这些谜题的行为，并讨论潜在的含义以及可能的权变措施。我们力求将每个谜题当作一个机会，来展示这门语言的不同寻常之处或突出一种常用语言特性里鲜为人知的方面。

详细地研究并解释每个谜题有很多乐趣，也有很多学习经验。希望你像我们一样，能从本书中得到很多收获，包括知识和娱乐。

最后，如果你遇到 Scala 代码的迷惑不解之处，请通过 Scala 谜题网站 scalapuzzlers.com 提交给我们。谜题社区期待你的加入。

Andrew Phillips, Nermin Šerifović

马萨诸塞州波士顿市

2014.5.28

致 谢

感谢每一位参与者，无论是提供了谜题代码，还是写了注释或建议，或者仅仅是传播了文字。感谢你们使本书的编写和 Scala 谜题网站 scalapuzzlers.com 的制作得以实现。

真挚地感谢我们的编辑 Jessica Kerr、Bill Venners 和 Theresa Gonzalez，感谢他们的探索和对本书的全面评审，感谢 Darlene Wallach 和 George Berger 排版并采用发布系统来处理这本书。

还要特别感谢 Scala 谜题 scalapuzzlers.com 网站的所有贡献者：Dominik Gruntz、A. P. Marki、Simon Schäfer、Konstantine Golikov、Seth Tisue、Daniel C. Sobral、Luc Bourlier、Vassil Dichev 和 Andraž Bajt。你们的贡献为本书提供了基础和灵感。

也要对那些指出错误和提出改进建议的读者表示感谢：Cay Horstmann、Harish Hurchurn、Marcin Kubala、Edward G. Prentice、Alex Varju。特别感谢 Dominik Gruntz 的全面评审并提出许多有用的建议。

献 辞

我的母亲 Karin 如涓涓细流般美妙的文笔熏陶我、影响我写成本书；我的两个耐心的朋友 Libby 和 Bill Venners，她们一直在用 Scalac 编译程序，要不是她们，我就不会认识 Scala。

——A.P.

我出色的妻子 Džana 毫无保留的支持，帮我完成了另一个副项目。我的两个调皮的儿子 Immy 和 Rayan，他们很好奇我是否在写一本“真正”的、以“从前”开始，以“结束”结尾的书。我敬爱的父母，Nad 和 Sabrija，他们在孩子的教育上投入大量的时间。

——N.Š.

前言

让代码做我们希望它做的事，是作为一个开发者的基本目的。所以没有什么比我们自认为理解的一段代码表现出与我们期望相反的行为更迷人、更重要的了。

本书汇集了一些 Scala 的例子。这不仅是以一种寓教于乐的方式来更好地理解这门语言，而且帮助你认识许多反直观的雷区和陷阱，防止它们在生产系统中干扰到你。

本书主要内容

全书共有 36 章，具体内容如下。

- 第 1 章“使用占位符”，本章中，我们通过引入占位符语法从函数调用中删除了一些陈词滥调。
- 第 2 章“初始化变量”，本章中，我们检查了 Scala 里给多个变量赋值的盛衰起伏。
- 第 3 章“成员声明的位置”，本章我们审视了类成员的周边环境对初始化的影响。
- 第 4 章“继承”，本章我们通过类初始化顺序跟踪几个抽象和重载值。
- 第 5 章“集合操作”，这一章我们试图计数几个集合而不用太担心它们确切的类型。
- 第 6 章“参数类型”，本章我们通过定义两种特定类型的帮助函数的通用版本尽量写出好的函数式程序。
- 第 7 章“闭包”，本章对几个延迟的访问器函数上的索引数据序列略窥一二。
- 第 8 章“Map 表达式”，本章我们用 for 表达式并通过 map 的应用来转换一些字母汤。
- 第 9 章“循环引用变量”，本章我们定义并随机地访问一对相互引用的变量。

- 第 10 章“等式的例子”，本章我们在 `case class` 的 `hashCode` 方法中混进调试代码。
- 第 11 章“`lazy val`”，本章我们试图初始化一个随环境而变化的 `lazy` 值。
- 第 12 章“集合的迭代顺序”，本章我们试图按值的顺序输出罗马数字。
- 第 13 章“自引用”，本章我们会查看 Scala 如何处理自引用变量。
- 第 14 章“`Return` 语句”，本章我们将体验程序流控制。
- 第 15 章“偏函数中的 `_`”，本章我们调用两个偏应用函数作为计数器。
- 第 16 章“多参数列表”，本章我们给偏函数提供不同数量的参数。
- 第 17 章“隐式参数”，本章我们从一个有隐式参数的方法创建一个偏应用函数。
- 第 18 章“重载”，本章我们定义并调用参数数量可变的重载方法。
- 第 19 章“命名参数和缺省参数”，本章我们分别调用一个指定和没有指定参数名的方法。
- 第 20 章“正则表达式”，本章我们从正则匹配中删除一些调试代码。
- 第 21 章“填充”，本章我们每次用一个“`*`”字符按希望的长度填充一个字符串。
- 第 22 章“投影”，本章我们将一个从调用 Java 代码（`gulp!`）返回的 `map` 投影到一个 Scala 类型的 `map`。
- 第 23 章“构造器参数”，本章我们实例化一个类的多个实例，这个类的构造器是由按名字传递的参数指定的。
- 第 24 章“`Double.NaN`”，本章我们对几个 `double` 数组进行排序。
- 第 25 章“`getOrElse`”，本章我们组合两个列表，当试图从结果中抽取一个元素时就有些令人迷惑了。
- 第 26 章“`Any Args`”，本章我们重构一个方法，当调用方法的代码忘记改变时，用多个参数列表的方式调用重构的方法依然可以成功。
- 第 27 章“`null`”，本章我们对一个从调用 Java 代码返回的字符串应用模式匹配。
- 第 28 章“`AnyVal`”，本章我们初始化一个其特定的类型在子类中声明的 `AnyVal` 子类型。

- 第 29 章“隐式变量”，本章我们通过导入附加的隐式变量的方式在程序中间创建一个“测试模式”上下文。
- 第 30 章“显式声明类型”，本章我们定义两种隐式转换，让编译器去弄清楚它们的返回类型。
- 第 31 章“View”，本章我们优化一个只能在 map 数据项的值上操作的 map 调用。
- 第 32 章“toSet”，本章我们在增加附加的元素之前将一个列表转换成一个集合。
- 第 33 章“缺省值”，本章我们使用两种不同的方式为 map 数据项指定缺省值。
- 第 34 章“关于 Main”，本章我们对一个用显式 main 方法的“保守”对象进行重构，把它修改成继承 App 特质的方法。
- 第 35 章“列表”，本章我们使用圆括号和大括号定义一个匿名函数。
- 第 36 章“计算集合的大小”，本章我们重构一个计数方法让它可以处理通用的集合类型。

如何阅读本书

本书中的谜题并非按特定顺序编写。你可以随机打开一个谜题阅读，会与从头到尾地读一样轻松。

如果你对 Scala 语言的特定方面感兴趣并且在寻找相关的谜题，本书最后的主题索引正是为你准备的。我们尽量根据读者要探索的主题对谜题进行了分类。

谜题中所有的代码例子都要用 2.11 Scala REPL^①解释执行，最近的一些变更，例如一些不支持的写法会让程序行为与 Scala 2.10.X 版本稍微不同，我们已经增加了注释或注解。

尽管本书经过了严格评审，但错误还是不可避免的。如果你发现任何错误，请通过 <http://booksites.artima.com/scalapuzzlers/errata> 告诉错误所在的页码。

① 从命令行输入“Scala”来启动 REPL（读—评估—打印—循环，Read-Evaluate-Print-Loop，REPL）。

电子书

本书同时提供印刷版书籍和 PDF 格式电子书。电子书并不是本书印刷版的简单的电子版复制。在保持与印刷版内容相同的同时，电子书还经过仔细设计和优化以适应应用计算机屏幕阅读。

首先要注意的是，电子书里大部分引用是超链接的形式。如果你选择进入某章、某个图形或词汇条目的超链接，PDF 查看器会立即将你带到选择的条目而不必翻来翻去地寻找。

此外，电子书的每页底部是一些导航链接。“封面”“概述”“目录”链接将你带到本书的开头；“索引”链接将你带到本书最后的索引部分；“讨论”链接将你带到论坛，在那里你可以与其他读者、作者和更大的 Scala 社区讨论问题。如果你发现一个拼写错误或者某些你认为能解释得更好的地方，请单击“建议”链接进入在线 Web 应用，在这里你可以给作者提供反馈。

尽管电子版与印刷版有相同的页面，但我们已经删除了电子书的空白页，并对剩下的页面重新编排了页码。各页的页码不同以便当你只打印本书的一部分时很容易决定 PDF 的页数。电子书的页码已经严格按照 PDF 阅读器计数方式进行了编排。

排版约定

一个术语首次使用的时候是斜体字。小的代码样例在同一行里用单间隔字体，例如 `x + 1`。较大的代码样例放在单行间隔的引用块中，如下所示：

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

当出现交互式 Shell 时，Shell 返回结果用亮字体显示：

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

谜题概览

1. 使用占位符	1
2. 初始化变量	5
3. 成员声明的位置	9
4. 继承	14
5. 集合操作	21
6. 参数类型	24
7. 闭包	29
8. Map 表达式	33
9. 循环引用变量	37
10. 等式的例子	44
11. lazy val	51
12. 集合的迭代顺序	54
13. 自引用	58
14. Return 语句	62
15. 偏函数中的_	67
16. 多参数列表	73
17. 隐式参数	78
18. 重载	83
19. 命名参数和缺省参数	88
20. 正则表达式	93
21. 填充	97
22. 投影	101

| 2 Scala 谜题

23. 构造器参数	106
24. Double.NaN	111
25. getOrElse	116
26. Any Args	120
27. null	124
28. AnyVal	129
29. 隐式变量	135
30. 显式声明类型	141
31. View	145
32. toSet	148
33. 缺省值	154
34. 关于 Main	159
35. 列表	165
36. 计算集合的大小	169

目 录

第 1 章	使用占位符	1
	可能的结果	1
	解释	2
	讨论	3
第 2 章	初始化变量	5
	可能的结果	5
	解释	6
	讨论	7
第 3 章	成员声明的位置	9
	可能的结果	10
	解释	10
	讨论	12
第 4 章	继承	14
	可能的结果	14
	解释	15
	讨论	16
	解决方法	17
第 5 章	集合操作	21
	可能的结果	21
	解释	22
	讨论	23
第 6 章	参数类型	24
	可能的结果	25
	解释	25
	讨论	27

第 7 章	闭包	29
	可能的结果	29
	解释	30
	讨论	32
第 8 章	Map 表达式	33
	可能的结果	33
	解释	34
	讨论	35
第 9 章	循环引用变量	37
	可能的结果	37
	解释	38
	讨论	40
第 10 章	等式的例子	44
	可能的结果	45
	解释	46
	讨论	48
第 11 章	lazy val	51
	可能的结果	51
	解释	52
	讨论	53
第 12 章	集合的迭代顺序	54
	可能的结果	55
	解释	55
	讨论	57
第 13 章	自引用	58
	可能的结果	58
	解释	58
	讨论	60
第 14 章	Return 语句	62
	可能的结果	62