优雅的Ruby

CONFIDENT RUBY

[美] Avdi Grimm 著
秦凡鹏 译
张汉东 审校

華中科技大學出版社

优雅的Ruby

CONFIDENT RUBY

[美] Avdi Grimm 著
秦凡鹏 译
张汉东 审校

華中科技大學出版社

内容简介

本书总结了32条Ruby编程技巧，帮助读者写出清晰、优雅、稳定的Ruby代码。作者AvdiGrimm主张Ruby方法应该像故事一样易于阅读。他将Ruby方法分成输入处理（CollectingInput）、功能实现（PerformingWork）、输出处理（DeliveringOutput）、失败处理（HandlingFailures）四个部分，针对每个部分的特点归纳实用的编程模式，并配合丰富的实例讲解，让读者写出优雅实用的Ruby代码，找回Ruby编程的乐趣。

湖北省版权局著作权合同登记　图字：17-2016-459号

图书在版编目（CIP）数据

优雅的Ruby/（美）阿弗迪·格林（Avdi Grimm）著；秦凡鹏，张汉东译. —武汉：华中科技大学出版社，2017.1
ISBN 978-7-5680-2489-1
Ⅰ. ①优… Ⅱ. ①阿… ②秦… ③张… Ⅲ. ①计算机网络-程序设计 Ⅳ. ①TP393.09

中国版本图书馆CIP数据核字（2017）第009157号

优雅的Ruby
Youya de Ruby

[美]Avdi Grimm　著
秦凡鹏　译　张汉东　审校

策划编辑：徐定翔　　责任编辑：徐定翔
责任校对：李　琴　　责任监印：周治超
出版发行：华中科技大学出版社（中国·武汉）　电话：（027）81321913
武汉市东湖新技术开发区华工科技园　邮编：430223
录　　排：华中科技大学惠友文印中心
印　　刷：湖北新华印务有限公司
开　　本：787mm×960mm　1/16
印　　张：15.5
字　　数：260千字
版　　次：2017年1月第1版第1次印刷
定　　价：64.80元

推荐序

Foreword

在 2011 年新奥尔良的 RubyConf 上，我认识了 Avdi。我并无过人之处，不过是一个无名小辈，赶鸭子上架，做了一两个演讲而已。Jim Weirich[1] 听了其中一个演讲，当即招手叫来 Avdi，把我介绍给他。我们三人跳过部分演讲，在会场外聊得非常开心。这次聊天不但开阔了见闻，而且改变了我们的生活。 我们三个当时是那么激动，以至于我都好奇为啥没人叫我们“闭嘴”，也没被休息室的负责人“请出去”。

时至今日，回想起那段时光，我们的谈话仍萦绕耳边，仿佛依然能感受到 Jim 那极具感染力的热情和 Avdi 的诚挚。我都不敢相信这一切是真的，不相信我有那么幸运。我仿佛已羽化登仙——登上了那片只属于程序员的仙境。

我坦言在写一本名为《Practical Object-Oriented Design in Ruby》的书，Avdi 很有风度地表示愿意试读。说是“试读”，他实际上费了很多心血和精力校对。他不仅给出建议，还逐行看 Ruby 代码，纠正错误、提升代码质量。当我感谢他时，他虽很高兴，但总是谦虚地说没做什么。就这点而言，我有相当的发言权，可以负责任地告诉你，他非常友好和有耐心。

---

1 译注：Jim Weirich，Ruby 界大师，Rake 之父。

因此，我很荣幸推荐这本《优雅的 Ruby》。Avdi 的思想贯穿全书。他关于方法如何组织的观点为我们思考代码提供了新的视角，他的“秘籍”为我们编写代码提供了最直接、最通俗易懂的指导。

他有一种天赋，能将严谨的技术理念用独特而轻松的方式讲出来。在本书中，有银行账户，当然也有书迷们喜欢的“Bookface”。当你需要放松时，会有调皮的小猫（emergency kittens）来陪你玩。他的这种天赋让书中那些例子显得既实用合理又有趣。

写书可谓是一件苦差事，但写书的动力深深扎根在我心里。正是这种动力促使我们帮助他人，讲解知识，提升自我，改变世界。我问 Avdi 为何要写这本书，他在邮件中提到了几个原因，在此，我想分享其中的两点。他说，“讲解知识很有趣”，而且“整个社区一直以来都对我很好，我一直在想他们为什么这么好，我希望自己也可以（为社区）做点贡献”。

现在你知道了，他就是这样一个充满幽默感和责任感的人。

Avdi 写的代码可读性非常高。在这本书里，他将教你写出优雅的代码。这对开发人员来讲，真是无上的荣耀。

好好欣赏这本书吧！

Sandi Metz[2]
2014 年 3 月 28 日

2 译注：Sandi Metz，《Practical Object-Oriented Design in Ruby : An Agile Primer》的作者。

序

Preface

2010 年 1 月，我在美国马里兰州巴尔的摩给满屋 Ruby 爱好者做了人生的第一次技术演讲，题为“优雅的代码”，主要讲我在 Ruby 工作中积累的一些经验。

从那以后，我就这一主题又做了多次演讲，受到了广泛的欢迎和好评。一直以来，我都想把这一主题做进一步的展开。你手上的这本书，正是这一心愿的结晶。

在本书的创作过程中，有许多人给予过帮助，以至于都不知从何说起。

首先，感谢所有当初听我演讲并鼓励我写书的朋友。我记不清都有哪些人了，但 Mike Subelsky 无疑是其中的第一个。

非常感谢 Sandi Metz 和 Noel Rappin[3]阅读本书早期的手稿和充当本书的编辑。他们的建议极大地提高了本书的可读性。还有 Dave Copeland，他那深刻的技术洞察力帮我阐明了不少模式。

感谢周围的 Ruby 同行 Chuck Wood、James Gray、David Brady、Josh

3 译著：Noel Rappin，著有《Professional Ruby on Rails (Programmer to Programmer)》《Rails Test Prescriptions: Keeping Your Application Healthy》等。

Susser、Katrina Owen。你们给予我精神上的支持，并充当我的倾听者，有些人还给了我写作建议。

感谢所有阅读本书 beta 版并指出谬误 （无论大小）的朋友。如果说本书看起来还算专业，而非心不在焉的黑客写的一本笔记的话，那都是你们的功劳。以下排名不分先后：Dennis Sutch、George Hemmings、Gerry Cardinal III、Evgeni Dzhelyov、Hans Verschooten、Kevin Burleigh、Michael Demazure、Manuel Vidaurre、Richard McGain、Michael Sevestre、Jake Goulding、Yannick Schutz、Mark Ijbema、John Ribar、Tom Close、Dragos .M、Brent Nordquist、Samnang Chhun、Dwayne R. Crooks、Thom Parkin、and Nathan Walls。我肯定遗漏了某些名字，如果你不幸就是其中一位，我在此表示深深的歉意。要知道，我对你的付出万分感激。

感谢 Ryan Biesemeyer 和 Grant Austin 捐赠的电子阅读设备，让我得以确保本书在所有平台上的阅读效果。

感谢 Benjamin Fleischer 和 Matt Swanson 把他们的项目贡献出来，做我重构的小白鼠。

最后，感谢 Stacey 一如既往的支持和鼓励。感谢 Lily、Josh、Kashti、Ebba 和 Ylva，是你们教给了我快乐的真谛。

目录

Contents

第 1 章　引言 1

1.1　当 Ruby 遭遇现实 2

1.2　自信优雅的代码 2

1.3　好的故事，糟糕的讲述 3

1.4　像写故事一样写代码 4

1.5　方法的四个部分 4

1.6　本书组织结构 8

第 2 章　功能实现 11

2.1　发送有效的消息 12

2.2　导入交易记录 13

2.3　识别消息 14

2.4　识别角色 14

2.5　避免马盖先主义 17

2.6　让语言为系统服务 17

2.7　像鸭子一样叫 18

2.8　驯养鸭群 19

第 3 章 收集输入 21
3.1 输入处理概述 21
3.1.1 间接输入 23
3.1.2 从角色到对象 26
3.1.3 保护边界而非内部 27
3.2 使用内置的类型转换协议 28
3.2.1 适用场景 28
3.2.2 摘要 28
3.2.3 基本原理 28
3.2.4 示例：宣布获奖结果 28
3.2.5 示例：Emacs 配置文件 30
3.2.6 标准类型转换方法列表 32
3.2.7 显式转换和隐式转换 33
3.2.8 明确提出参数要求 37
3.2.9 小结 39
3.3 有条件地使用类型转换方法 40
3.3.1 使用场景 40
3.3.2 摘要 40
3.3.3 基本原理 40
3.3.4 示例：打开文件 40
3.3.5 违反鸭子类型的唯一特例 42
3.3.6 小结 45
3.4 自定义类型转换协议 46
3.4.1 使用场景 46
3.4.2 摘要 46
3.4.3 基本原理 46
3.4.4 示例：接收一个点或一对坐标 46

3.4.5 小结 48
3.5 定义自定义类型的转换协议 49
3.5.1 使用场景 49
3.5.2 摘要 49
3.5.3 基本原理 49
3.5.4 示例：将英尺转换为米 49
3.5.5 小结 52
3.6 利用内置强制类型转换方法 53
3.6.1 使用场景 53
3.6.2 摘要 53
3.6.3 基本原理 53
3.6.4 示例：格式化打印数字 53
3.6.5 Hash.[] 57
3.6.6 小结 57
3.7 用 Array()将输入数组化 58
3.7.1 使用场景 58
3.7.2 摘要 58
3.7.3 基本原理 58
3.7.4 示例：可变参数 58
3.7.5 小结 60
3.8 自定义强制类型转换方法 61
3.8.1 使用场景 61
3.8.2 摘要 61
3.8.3 基本原理 61
3.8.4 示例：应用于 2D 图形中的强制类型转换方法 62
3.8.5 关于 module_function 63
3.8.6 结合类型转换协议和强制类型转换方法 64
3.8.7 用 Lambdas 表达式作 case 分支 66

3.8.8 小结 67
3.9 用自定义类替换类字符串类型 68
3.9.1 使用场景 68
3.9.2 摘要 68
3.9.3 基本原理 68
3.9.4 示例：红绿灯的状态问题 69
3.9.5 小结 77
3.10 用适配器装饰输入 78
3.10.1 使用场景 78
3.10.2 摘要 78
3.10.3 基本原理 78
3.10.4 示例：将日志写进 IRC 78
3.10.5 小结 82
3.11 利用透明适配器逐步消除类型依赖 83
3.11.1 适用场景 83
3.11.2 摘要 83
3.11.3 基本原理 83
3.11.4 示例：再探将日志写进 IRC 的示例 83
3.11.5 小结 86
3.12 利用先决条件排除非法输入 87
3.12.1 使用场景 87
3.12.2 摘要 87
3.12.3 基本原理 87
3.12.4 示例：员工入职日期 87
3.12.5 “可执行文档” 91
3.12.6 小结 91
3.13 利用#fetch 确保 Hash 键的存在性 92
3.13.1 使用场景 92

3.13.2 摘要 92
3.13.3 基本原理 92
3.13.4 示例：useradd(8)包装器 92
3.13.5 尝试#fetch 95
3.13.6 自定义#fetch 98
3.13.7 小结 99
3.14 利用#fetch 提供默认参数 100
3.14.1 使用场景 100
3.14.2 摘要 100
3.14.3 基本原理 100
3.14.4 示例：可选的 logger 参数 100
3.14.5 可重用的#fetch 代码块 104
3.14.6 双参数#fetch 106
3.14.7 小结 107
3.15 用断言验证假设 108
3.15.1 使用场景 108
3.15.2 摘要 108
3.15.3 基本原理 108
3.15.4 示例：导入银行记录 108
3.15.5 小结 113
3.16 用卫语句来处理特殊场景 114
3.16.1 使用场景 114
3.16.2 摘要 114
3.16.3 基本原理 114
3.16.4 示例："静音模式"标志 114
3.16.5 提前返回 116
3.16.6 小结 117
3.17 用对象表示特殊场景 118

3.17.1 使用场景 118
3.17.2 摘要 118
3.17.3 基本原理 118
3.17.4 示例：游客用户 118
3.17.5 用特例对象来表示当前用户 121
3.17.6 小步改进 126
3.17.7 保持特例对象和普通对象的同步 128
3.17.8 小结 129
3.18 用空对象表示不做事的情况 130
3.18.1 使用场景 130
3.18.2 摘要 130
3.18.3 基本原理 130
3.18.4 示例：输出日志到 shell 命令行 131
3.18.5 通用空对象 133
3.18.6 穿越事界 134
3.18.7 让空对象返回 false 138
3.18.8 小结 140
3.19 用良性值替代 nil 142
3.19.1 使用场景 142
3.19.2 摘要 142
3.19.3 基本原理 142
3.19.4 示例：显示会员地理位置信息 142
3.19.5 无害就好 145
3.19.6 小结 146
3.20 用 symbols 做占位符 147
3.20.1 使用场景 147
3.20.2 摘要 147
3.20.3 基本原理 147

3.20.4 示例：web service 可选认证 147
3.20.5 都是 nil 惹的祸 149
3.20.6 带语义的占位符 152
3.20.7 小结 154
3.21 将参数封装到参数对象中 155
3.21.1 使用场景 155
3.21.2 摘要 155
3.21.3 基本原理 155
3.21.4 参数对象回顾 155
3.21.5 添加可选参数 159
3.21.6 小结 163
3.22 提取参数构建器 164
3.22.1 使用场景 164
3.22.2 摘要 164
3.22.3 基本原理 164
3.22.4 示例：方便的绘点 API 164
3.22.5 Net/HTTP vs. Faraday 168
3.22.6 提取参数 Builder 170
3.22.7 小结 172

第 4 章 输出处理 173
4.1 用全函数作为方法返回值 174
4.1.1 使用场景 174
4.1.2 摘要 174
4.1.3 基本原理 174
4.1.4 示例：单词搜索 174
4.1.5 小结 178
4.2 执行回调而非返回状态 179

4.2.1 使用场景 179
4.2.2 摘要 179
4.2.3 基本原理 179
4.2.4 示例 179
4.2.5 小结 182
4.3 用良性值表示失败 183
4.3.1 使用场景 183
4.3.2 摘要 183
4.3.3 基本原理 183
4.3.4 示例：在侧边栏上显示推文 183
4.3.5 小结 185
4.4 用特例对象表示失败 186
4.4.1 使用场景 186
4.4.2 摘要 186
4.4.3 基本原理 186
4.4.4 示例：游客用户 186
4.4.5 小结 187
4.5 返回状态对象 188
4.5.1 使用场景 188
4.5.2 摘要 188
4.5.3 基本原理 188
4.5.4 示例：记录导入结果 188
4.5.5 小结 192
4.6 将状态对象传给回调 193
4.6.1 使用场景 193
4.6.2 摘要 193
4.6.3 基本原理 193
4.6.4 示例：将导入结果传给回调 193

4.6.5 测试状态对象 198
4.6.6 小结 199
4.7 用 throw 提前终止执行 200
4.7.1 使用场景 200
4.7.2 摘要 200
4.7.3 示例：提前终止 HTML 文档解析 200
4.7.4 小结 205

第 5 章 失败处理 207
5.1 优先使用顶层异常捕获 208
5.1.1 使用场景 208
5.1.2 摘要 208
5.1.3 基本原理 208
5.1.4 示例 208
5.1.5 小结 209
5.2 用受检方法封装危险操作 210
5.2.1 使用场景 210
5.2.2 摘要 210
5.2.3 基本原理 210
5.2.4 示例 210
5.2.5 演进成 Adapters 212
5.2.6 小结 212
5.3 使用护卫方法 213
5.3.1 使用场景 213
5.3.2 摘要 213
5.3.3 基本原理 213
5.3.4 示例：子进程状态检测 213
5.3.5 小结 216

第 6 章　为了优雅重构 217
6.1　MetricFu 218
6.1.1　Location 218
6.1.2　HotspotAnalyzedProblems 222
6.1.3　排名 225
6.2　Stringer 227
后记 231