

Scala程序设计（第2版）



[Scala程序设计（第2版）_下载链接1](#)

著者:[美] Dean Wampler

出版者:人民邮电出版社

出版时间:2016-3

装帧:平装

isbn:9787115416810

在第1版的基础之上，第2版介绍了Scala的最新语言特性，新添了模式匹配、推导式以及高级函数式编程等知识。通过本书，读者还能学会如何使用Scala命令行工具、第三方工具、库以及适用于编辑器和IDE的Scala相关插件。本书既适合Scala初学者入门，

也适合经验丰富的Scala开发者进阶参考。

利用Scala简洁灵活的语法，提高编程效率。

深入学习函数式编程的基本技能和高级技能。

使用Scala函数式组合器，构造“杀手级”大数据应用。

使用Scala提供的trait类型实现mixin组合，使用模式匹配实现数据抽取功能。

学习Scala语言中复杂的类型系统，包括了函数式编程和面向对象编程中的概念。

深入学习包括Akka的Scala并发工具。

掌握如何开发丰富的领域特定语言。

掌握构建可扩展、健壮的Scala应用软件的良好设计技巧。

作为一本强调数据科学的图书，本书中出现的代码示例均保存在公开的Github仓库中。通过立即可启动的虚拟机，这些示例代码可以很容易地获得。该虚拟机中预装了一组IPython Notebook，为我们提供方便的交互式学习环境。

作者介绍:

Dean Wampler

博士是Typesafe公司的大数据产品架构师。Typesafe使用Scala、函数式编程、Spark、Hadoop以及Akka技术编写以数据为中心的工具和服务。Dean是《面向Java开发者的函数式编程》的作者，同时也与他人合著了《Hive编程指南》一书。

Alex Payne

是Twitter的平台组长。在Alex开发的服务基础上，其他的程序开发者构造了一套备受欢迎的社交消息服务。此前，Alex还为政治竞选、公益性组织以及初创企业编写过一些Web应用。

目录: 序 xv

前言 xvii

第1章 零到六十：Scala简介 1

1.1 为什么选择Scala 1

1.1.1 富有魅力的Scala 2

1.1.2 关于Java 8 3

1.2 安装Scala 3

1.2.1 使用SBT 5

1.2.2 执行Scala命令行工具 6

1.2.3 在IDE中运行Scala REPL 8

1.3 使用Scala 8

1.4 并发 17

1.5 本章回顾与下一章提要 27

第2章 更简洁，更强大 28

2.1 分号 28

2.2 变量声明 29

2.3 Range	31
2.4 偏函数	32
2.5 方法声明	33
2.5.1 方法默认值和命名参数列表	33
2.5.2 方法具有多个参数列表	34
2.5.3 Future简介	35
2.5.4 嵌套方法的定义与递归	38
2.6 推断类型信息	40
2.7 保留字	44
2.8 字面量	46
2.8.1 整数字面量	46
2.8.2 浮点数字面量	47
2.8.3 布尔型字面量	48
2.8.4 字符字面量	48
2.8.5 字符串字面量	48
2.8.6 符号字面量	50
2.8.7 函数字面量	50
2.8.8 元组字面量	50
2.9 Option、Some 和None：避免使用null	52
2.10 封闭类的继承	53
2.11 用文件和名空间组织代码	54
2.12 导入类型及其成员	55
2.12.1 导入是相对的	56
2.12.2 包对象	57
2.13 抽象类型与参数化类型	57
2.14 本章回顾与下一章提要	59
第3章 要点详解	60
3.1 操作符重载?	60
3.2 无参数方法	63
3.3 优先级规则	64
3.4 领域特定语言	65
3.5 Scala中的if语句	66
3.6 Scala中的for推导式	67
3.6.1 for循环	67
3.6.2 生成器表达式	67
3.6.3 保护式：筛选元素	67
3.6.4 Yielding	68
3.6.5 扩展作用域与值定义	69
3.7 其他循环结构	70
3.7.1 Scala的while循环	71
3.7.2 Scala中的do-while循环	71
3.8 条件操作符	71
3.9 使用try、catch和final子句	72
3.10 名字调用和值调用	75
3.11 惰性赋值	78
3.12 枚举	79
3.13 可插入字符串	81
3.14 Trait：Scala语言的接口和“混入”	83
3.15 本章回顾与下一章提要	85
第4章 模式匹配	86
4.1 简单匹配	86
4.2 match中的值、变量和类型	87
4.3 序列的匹配	90
4.4 元组的匹配	94

- 4.5 case中的guard语句 94
- 4.6 case类的匹配 95
 - 4.6.1 unapply方法 96
 - 4.6.2 unapplySeq方法 100
- 4.7 可变参数列表的匹配 101
- 4.8 正则表达式的匹配 103
- 4.9 再谈case语句的变量绑定 104
- 4.10 再谈类型匹配 104
- 4.11 封闭继承层级与全覆盖匹配 105
- 4.12 模式匹配的其他用法 107
- 4.13 总结关于模式匹配的评价 111
- 4.14 本章回顾与下一章提要 111
- 第5章 隐式详解 112
 - 5.1 隐式参数 112
 - 5.2 隐式参数适用的场景 115
 - 5.2.1 执行上下文 115
 - 5.2.2 功能控制 115
 - 5.2.3 限定可用实例 116
 - 5.2.4 隐式证据 120
 - 5.2.5 绕开类型擦除带来的限制 122
 - 5.2.6 改善报错信息 124
 - 5.2.7 虚类型 124
 - 5.2.8 隐式参数遵循的规则 127
 - 5.3 隐式转换 128
 - 5.3.1 构建独有的字符串插入器 132
 - 5.3.2 表达式问题 134
 - 5.4 类型类模式 135
 - 5.5 隐式所导致的技术问题 137
 - 5.6 隐式解析规则 139
 - 5.7 Scala内置的各种隐式 139
 - 5.8 合理使用隐式 146
 - 5.9 本章回顾与下一章提要 146
- 第6章 Scala函数式编程 147
 - 6.1 什么是函数式编程 148
 - 6.1.1 数学中的函数 148
 - 6.1.2 不可变变量 149
 - 6.2 Scala中的函数式编程 151
 - 6.2.1 匿名函数、Lambda与闭包 152
 - 6.2.2 内部与外部的纯粹性 154
 - 6.3 递归 154
 - 6.4 尾部调用和尾部调用优化 155
 - 6.5 偏应用函数与偏函数 157
 - 6.6 Curry化与函数的其他转换 158
 - 6.7 函数式编程的数据结构 162
 - 6.7.1 序列 162
 - 6.7.2 映射表 166
 - 6.7.3 集合 168
 - 6.8 遍历、映射、过滤、折叠与归约 168
 - 6.8.1 遍历 169
 - 6.8.2 映射 170
 - 6.8.3 扁平映射 172
 - 6.8.4 过滤 173
 - 6.8.5 折叠与归约 174
 - 6.9 向左遍历与向右遍历 178

- 6.10 组合器：软件最佳组件抽象 183
- 6.11 关于复制 186
- 6.12 本章回顾与下一章提要 188
- 第7章 深入学习for推导式 189
 - 7.1 内容回顾：for推导式组成元素 189
 - 7.2 for推导式：内部机制 192
 - 7.3 for推导式的转化规则 194
 - 7.4 Option以及其他的一些容器类型 197
 - 7.4.1 Option容器 197
 - 7.4.2 Either：Option类型的逻辑扩展 200
 - 7.4.3 Try类型 205
 - 7.4.4 Scalaz提供的Validation类 206
 - 7.5 本章回顾与下一章提要 209
- 第8章 Scala面向对象编程 210
 - 8.1 类与对象初步 211
 - 8.2 引用与值类型 213
 - 8.3 价值类 214
 - 8.4 父类 217
 - 8.5 Scala的构造器 217
 - 8.6 类的字段 221
 - 8.6.1 统一访问原则 223
 - 8.6.2 一元方法 224
 - 8.7 验证输入 224
 - 8.8 调用父类构造器（与良好的面向对象设计） 226
 - 8.9 嵌套类型 230
 - 8.10 本章回顾与下一章提要 232
- 第9章 特征 233
 - 9.1 Java 8中的接口 233
 - 9.2 混入trait 234
 - 9.3 可堆叠的特征 238
 - 9.4 构造trait 243
 - 9.5 选择类还是trait 244
 - 9.6 本章回顾与下一章提要 245
- 第10章 Scala对象系统（I） 246
 - 10.1 参数化类型：继承转化 246
 - 10.1.1 Hood下的函数 247
 - 10.1.2 可变类型的变异 250
 - 10.1.3 Scala和Java中的变异 252
 - 10.2 Scala的类型层次结构 253
 - 10.3 闲话Nothing（以及Null） 254
 - 10.4 Product、case类和元组 258
 - 10.5 Predef对象 260
 - 10.5.1 隐式转换 260
 - 10.5.2 类型定义 262
 - 10.5.3 条件检查方法 263
 - 10.5.4 输入输出方法 263
 - 10.5.5 杂项方法 265
 - 10.6 对象的相等 265
 - 10.6.1 equals方法 266
 - 10.6.2 == 和!=方法 266
 - 10.6.3 eq 和ne方法 267
 - 10.6.4 数组相等和sameElements方法 267
 - 10.7 本章回顾与下一章提要 268
- 第11章 Scala对象系统（II） 269

- 11.1 覆写类成员和trait成员 269
- 11.2 尝试覆写final声明 272
- 11.3 覆写抽象方法和具体方法 272
- 11.4 覆写抽象字段和具体字段 274
- 11.5 覆写抽象类型 280
- 11.6 无须区分访问方法和字段：统一访问原则 280
- 11.7 对象层次结构的线性化算法 282
- 11.8 本章回顾与下一章提要 287
- 第12章 Scala集合库 288
 - 12.1 通用、可变、不可变、并发以及并行集合 288
 - 12.1.1 scala.collection包 289
 - 12.1.2 collection.concurrent包 290
 - 12.1.3 collection.convert包 291
 - 12.1.4 collection.generic包 291
 - 12.1.5 collection.immutable包 291
 - 12.1.6 scala.collection.mutable包 292
 - 12.1.7 scala.collection.parallel包 294
 - 12.2 选择集合 295
 - 12.3 集合库的设计惯例 296
 - 12.3.1 Builder 296
 - 12.3.2 CanBuildFrom 297
 - 12.3.3 Like特征 298
 - 12.4 值类型的特化 298
 - 12.5 本章回顾与下一章提要 300
- 第13章 可见性规则 301
 - 13.1 默认可见性：公有可见性 301
 - 13.2 可见性关键字 302
 - 13.3 Public可见性 303
 - 13.4 Protected可见性 304
 - 13.5 Private可见性 305
 - 13.6 作用域内私有和作用域内受保护可见性 306
 - 13.7 对可见性的想法 312
 - 13.8 本章回顾与下一章提要 313
- 第14章 Scala类型系统 (I) 314
 - 14.1 参数化类型 315
 - 14.1.1 变异标记 315
 - 14.1.2 类型构造器 315
 - 14.1.3 类型参数的名称 315
 - 14.2 类型边界 315
 - 14.2.1 类型边界上限 316
 - 14.2.2 类型边界下限 316
 - 14.3 上下文边界 320
 - 14.4 视图边界 320
 - 14.5 理解抽象类型 322
 - 14.6 自类型标记 325
 - 14.7 结构化类型 329
 - 14.8 复合类型 332
 - 14.9 存在类型 334
 - 14.10 本章回顾与下一章提要 335
- 第15章 Scala类型系统 (II) 336
 - 15.1 路径相关类型 336
 - 15.1.1 C.this 337
 - 15.1.2 C.super 337
 - 15.1.3 path.x 338

- 15.2 依赖方法类型 339
- 15.3 类型投影 340
- 15.4 值的类型 343
 - 15.4.1 元组类型 343
 - 15.4.2 函数类型 343
 - 15.4.3 中缀类型 343
- 15.5 Higher-Kinded类型 344
- 15.6 类型Lambda 348
- 15.7 自递归类型：F-Bounded多态 350
- 15.8 本章回顾与下一章提要 351
- 第16章 高级函数式编程 352
 - 16.1 代数数据类型 352
 - 16.1.1 加法类型与乘法类型 352
 - 16.1.2 代数数据类型的属性 354
 - 16.1.3 代数数据类型的最后思考 355
 - 16.2 范畴理论 355
 - 16.2.1 关于范畴 356
 - 16.2.2 Functor范畴 356
 - 16.2.3 Monad范畴 360
 - 16.2.4 Monad的重要性 362
 - 16.3 本章回顾与下一章提要 363
- 第17章 并发工具 365
 - 17.1 scala.sys.process包 365
 - 17.2 Future类型 367
 - 17.3 利用Actor模型构造稳固且可扩展的并发应用 371
 - 17.4 Akka：为Scala设计的Actor系统 372
 - 17.5 Pickling和Spores 383
 - 17.6 反应式编程 384
 - 17.7 本章回顾与下一章提要 385
- 第18章 Scala与大数据 386
 - 18.1 大数据简史 386
 - 18.2 用Scala改善MapReduce 387
 - 18.3 超越MapReduce 392
 - 18.4 数学范畴 393
 - 18.5 Scala数据工具列表 394
 - 18.6 本章回顾与下一章提要 394
- 第19章 Scala动态调用 396
 - 19.1 一个较为激进的示例：Ruby on Rails框架中的ActiveRecord库 396
 - 19.2 使用动态特征实现Scala 中的动态调用 397
 - 19.3 关于DSL的一些思考 402
 - 19.4 本章回顾与下一章提要 402
- 第20章 Scala的领域特定语言 403
 - 20.1 DSL 示例：Scala中XML和JSON DSL 404
 - 20.2 内部DSL 406
 - 20.3 包含解析组合子的外部DSL 410
 - 20.3.1 关于解析组合子 410
 - 20.3.2 计算工资单的外部DSL 410
 - 20.4 内部DSL与外部DSL：最后的思考 413
 - 20.5 本章回顾与下一章提要 413
- 第21章 Scala工具和库 414
 - 21.1 命令行工具 414
 - 21.1.1 命令行工具：scalac 414
 - 21.1.2 Scala命令行工具 418
 - 21.1.3 scalap和javap命令行工具 421

21.1.4 scaladoc 命令行工具	422
21.1.5 fsc命令行工具	422
21.2 构建工具	422
21.2.1 SBT: Scala标准构建工具	423
21.2.2 其他构建工具	425
21.3 与IDE或文本编辑器集成	425
21.4 在Scala中应用测试驱动开发	426
21.5 第三方库	427
21.6 本章回顾与下一章提要	429
第22章 与Java的互操作	430
22.1 在Scala代码中使用Java名称	430
22.2 Java泛型与Scala泛型	430
22.3 JavaBean的性质	432
22.4 AnyVal类型与Java原生类型	433
22.5 Java代码中的Scala名称	433
22.6 本章回顾与下一章提要	434
第23章 应用程序设计	435
23.1 回顾之前的内容	435
23.2 注解	437
23.3 Trait即模块	441
23.4 设计模式	442
23.4.1 构造型模式	443
23.4.2 结构型模式	443
23.4.3 行为型模式	444
23.5 契约式设计带来更好的设计	446
23.6 帕特农神庙架构	448
23.7 本章回顾与下一章提要	453
第24章 元编程: 宏与反射	454
24.1 用于理解类型的工具	455
24.2 运行时反射	455
24.2.1 类型反射	455
24.2.2 ClassTag、TypeTag与Manifest	457
24.3 Scala 的高级运行时反射API	458
24.4 宏	461
24.4.1 宏的示例: 强制不变性	463
24.4.2 关于宏的最后思考	466
24.5 本章回顾与下一章提要	466
附录A 参考文献	468
作者简介	473
关于封面	473
• • • • •	(收起)

[Scala程序设计 \(第2版\)](#) [下载链接1](#)

标签

Scala

函数式编程

编程语言

大数据

编程

Java

scala

O'Reilly

评论

翻译稍微有些欠缺，内容其实很好很全很新！最值得推荐的Scala书籍之一！

内容相对全面且有深度，但仍然不够友好。

内容丰富完整又循序渐进，讲解功力非凡。中文版翻译质量一般，注意对照看原版，免得在不通顺的不能理解的地方浪费时间或理解错误。相比而言，Martin Odersky的《Programming in Scala》会更条理化学院派些，通俗讲解比较省而推理式的讲解要多。当然，过了适应期，推理式的讲解更有有效率，能举一反三。

不适合新人入门，新人应该直接进入编程环节，模仿学习别人代码即可，如果编程前非要读这本书，那可能很快就因挫败感放弃 scala 编程了。老实说，scala 我已经有相当的编程经验了，但读这本书还是会觉得非常难受。

书写的不错，但是scala这种语言不喜欢，心智模型太复杂了

其实还蛮有深度的，至少我这样觉得；广度就更不待说了。但是，看得我很难受。而且看完忘记得也很快，心累。

读了一半，类型系统实在读不下去了。另外翻译很一般，把“function under hood”翻译成“Hood下的函数”，这这真不是用google翻译的？

写作风格很像Java的经典书籍《Java编程思想》，内容全面，示例新颖，需要细细品味。

很不错的书，跳着看了下，对Scala有了进一步的认识。

冗长难看，总是生造词汇

挺不错，讲的很细，适合入门并且提供了详细引申

学习scala的入门必选

改良型java语言入门书籍，为了spark看了看……

难度高于Programming in Scala, 涉及Scala语言的许多高级特性。

[Scala程序设计（第2版）_下载链接1](#)

书评

在第1版的基础之上，第2版介绍了Scala的最新语言特性，新添了模式匹配、推导式以及高级函数式编程等知识。通过本书，读者还能学会如何使用Scala命令行工具、第三方工具、库以及适用于编辑器和IDE的Scala相关插件。本书既适合Scala初学者入门，也适合经验丰富的Scala开发者进阶...

A great book for readers with java knowledges. The book covers many topics and goes into details of them. Nice examples are a big plus. Would highly recommend this book for beginners.

[Scala程序设计（第2版）_下载链接1](#)