

《函数式编程思维》

图书基本信息

书名：《函数式编程思维》

13位ISBN编号：9787115400415

出版时间：2015-8

作者：[美] Neal Ford

页数：162

译者：郭晓刚

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《函数式编程思维》

内容概要

本书脱离特定的语言特性，关注各种OOP语言的共同实践做法，展示如何通过函数式语言解决问题。例如，如何利用函数式语言，通过高阶函数、多元方法等完成代码重用。知名软件架构师Neal Ford为我们展示不同的编程范式，帮我们完成从Java命令式编程人员，到使用Java、Clojure、Scala函数式编程人员的完美转变，建立对函数式语言的语法和语义的良好理解。

《函数式编程思维》

作者简介

Neal Ford

在跨国IT咨询公司ThoughtWorks担任总监、软件架构师和文化基因传播人。他精通各种编程语言，主要的咨询业务是大规模企业应用的设计、构建和工程实践。他还是一位国际知名的讲师，登上过全世界各种开发者会议的讲台。

书籍目录

译者序	ix
前言	xi
第1章 为什么	1
1.1 范式转变	2
1.2 跟上语言发展的潮流	4
1.3 把控制权让渡给语言/运行时	4
1.4 简洁	5
第2章 转变思维	9
2.1 普通的例子	9
2.1.1 命令式解法	9
2.1.2 函数式解法	10
2.2 案例研究：完美数的分类问题	15
2.2.1 完美数分类的命令式解法	15
2.2.2 稍微向函数式靠拢的完美数分类解法	16
2.2.3 完美数分类的Java 8实现	18
2.2.4 完美数分类的Functional Java实现	19
2.3 具有普遍意义的基本构造单元	21
2.3.1 筛选	22
2.3.2 映射	23
2.3.3 折叠/化约	25
2.4 函数的同义异名问题	28
2.4.1 筛选	28
2.4.2 映射	31
2.4.3 折叠/化约	33
第3章 权责让渡	37
3.1 迭代让位于高阶函数	37
3.2 闭包	38
3.3 柯里化和函数的部分施用	41
3.3.1 定义与辨析	41
3.3.2 Groovy的情况	42
3.3.3 Clojure的情况	44
3.3.4 Scala的情况	44
3.3.5 一般用途	47
3.4 递归	48
3.5 Stream 和作业顺序重排	53
第4章 用巧不用蛮	55
4.1 记忆	55
4.1.1 缓存	56
4.1.2 引入“记忆”	59
4.2 缓求值	65
4.2.1 Java语言下的缓求值迭代子	65
4.2.2 使用Totally Lazy框架的完美数分类实现	67
4.2.3 Groovy语言的缓求值列表	69
4.2.4 构造缓求值列表	72
4.2.5 缓求值的好处	74
4.2.6 缓求值的字段初始化	76
第5章 演化的语言	79

5.1	少量的数据结构搭配大量的操作	79
5.2	让语言去迎合问题	81
5.3	对分发机制的再思考	82
5.3.1	Groovy对分发机制的改进	82
5.3.2	“身段柔软”的Clojure语言	83
5.3.3	Clojure的多重方法和基于任意特征的多态	85
5.4	运算符重载	87
5.4.1	Groovy	87
5.4.2	Scala	89
5.5	函数式的数据结构	91
5.5.1	函数式的错误处理	91
5.5.2	Either类	92
5.5.3	Option类	100
5.5.4	Either树和模式匹配	100
第6章	模式与重用	107
6.1	函数式语言中的设计模式	107
6.2	函数级别的重用	108
6.2.1	Template Method模式	109
6.2.2	Strategy模式	111
6.2.3	Flyweight模式和记忆	113
6.2.4	Factory模式和柯里化	116
6.3	结构化重用和函数式重用的对比	117
第7章	现实应用	125
7.1	Java 8	125
7.1.1	函数式接口	126
7.1.2	Optional类型	128
7.1.3	Java 8的stream	128
7.2	函数式的基础设施	129
7.2.1	架构	129
7.2.2	Web 框架	132
7.2.3	数据库	133
第8章	多语言与多范式	135
8.1	函数式与元编程的结合	136
8.2	利用元编程在数据类型之间建立映射	137
8.3	多范式语言的后顾之忧	140
8.4	上下文型抽象与复合型抽象的对比	141
8.5	函数式金字塔	143
	作者简介	147
	封面介绍	147

精彩短评

- 1、很受启发！
- 2、只说了一点函数编程术语，实现过程不讲，而是直接调用别的语言的实现，看了没啥收获。
- 3、感觉一般吧，内容浅显，概念一笔带过，不太深入
- 4、很棒的书，然而后面两章并没有看完。还算是很易懂的，例子也很多。
- 5、前三章很受启发，后面几章很一般。
- 6、函数式入门书，挺不错。你要是学了scala或者groovy，但是对函数式不太了解，看了这本书，会让你更加理解语言和其函数式特性。
- 7、非常值得推荐的一本书，通过jvm语言总结归纳出函数式的特点，对于学习scala、groovy都很有帮助
- 8、对比OO，过程式编程，讲述函数式编程的思维。
- 9、还行，没那么差。对比了老的开发模型（Java + OO）和函数式模型（CQRS，我们一直使用但不知道叫什么的模型）。自嘲了Java黑了Maven。最后给出了动静态语言的推荐和理由。主要还是用Java 8讲解的。当然，我自己还是更习惯从数据抽象，而不是对象抽象。
- 10、一本言简意赅的薄书，把一些基本原则讲了一下。
- 11、FP入门读物
- 12、Java 8命令式到对象式到函数式，里面例子一目了然。关键是设计模式的进化论。
- 13、读了这本书，很有启发。
- 14、入门不错，介绍了FP的一些思想和概念。如果已经有了解过FP，花个半天也就能翻完了
- 15、函数式编程，弱化数据结构的设计，可以专注程序的逻辑。喜欢。。。
- 16、没看懂，囧rz。。。
- 17、用三种语言讲了函数式编程，多少让人读着有些烦躁。或许是我太急躁了。
- 18、说得不透彻，函数式编程的东西也说得不深入。
- 19、对我这个学clojure有一段时间的人而言意义巨大。遇到的问题几乎都得到了有一定的解答，不可多得的良品。
- 20、在没啥FP实战经验的前提下看这本书，有点刻舟求剑的意味。话说Dietzler的Access定律说的真对。。。（第一本用kindle看完的书，mark下
- 21、读起来酣畅淋漓，不过阅读前需要事先对Groovy或Scala或Clojure有一定了解
- 22、差了那么一口气。
- 23、前面几章还不错，后面几章比较水。
- 24、作者用groovy scala clojure java四种语言跟你讲什么是函数式，我觉得如果没有基础的话，更有可能让大家更难follow他
- 25、讲道理就是用来入门的
- 26、英国办公室的大神，awayday时候听过他的session，很不错。
- 27、对相关api有了新的了解，但其实那么多语言的范例并不是很必要，当然，因为我只熟java和Scala。
- 28、作为前端，看这本书了解其他需要的函数式方案还是很不错的。但很多内容在js场景是不能用的，需要配合js版本的函数式书籍一起看
- 29、一直觉得讲函数式编程在于思想，而不在于用几种语言实现。书中更多的是用几种语言实现同一个特性，有点像语法对比。反而弱化了函数式思想的内容。
- 30、看了之后收货很少。也许是因为之前写过不少Scala,Lisp的原因,换句话说,如果已经系统仔细地看过某个函数式语言的书,就用不着看这本书。
- 31、对函数式编程的思想讲解的挺深入浅出的
- 32、看的英文原版，讲的挺到位的，比较基础。关于函数式与命令式的思考方式。不过已经对scala有1年左右的学习经验，所以这本书并没有带来太多新的给我，讲的东西基本上都在学习scala的过程中通过各种博客，书籍掌握了

《函数式编程思维》

精彩书评

1、本书脱离特定的语言特性，关注各种OOP语言的共同实践做法，展示如何通过函数式语言解决问题。例如，如何利用函数式语言，通过高阶函数、多元方法等完成代码重用。知名软件架构师Neal Ford为我们展示不同的编程范式，帮我们完成从Java命令式编程人员，到使用Java、Clojure、Scala函数式编程人员的完美转变，建立对函数式语言的语法和语义的良好理解。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com