

《Scala与Clojure函数式编程模》

图书基本信息

书名：《Scala与Clojure函数式编程模式》

13位ISBN编号：9787115388946

作者：[美] Michael Bevilacqua-Linn

页数：216

译者：赵震一

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

《Scala与Clojure函数式编程模》

内容概要

StackOverflow刚刚发布了2015开发者调查，最受程序员喜爱的开发语言中，Clojure和Scala分别位列第五和第六。排名第一的是Swift，而Swift相比原先的Objective-C最重要的优点之一，就是对函数式编程提供了更好的支持。同时，新一代分布式计算系统Spark不仅在实现中选择了Scala，其提供的首选编程语言也是Scala。

所有迹象都显示，诞生50多年后，函数式编程却焕发了青春，越来越受到关注和青睐，从边缘地带步入了主流，除了Scala和Clojure这些新生函数式编程语言大行其道之外，Java等老牌面向对象的编程语言也开始支持匿名函数。函数式编程能简化开发过程，尤其是大型知识管理系统应用程序从中受益良多。

本书向读者展示了如何采用函数式方案来替代或简化面向对象编程中使用的诸多通用模式，同时还介绍了一些在函数式世界中广泛使用的模式。如果你是一名Java程序员，希望了解函数式编程能为你的工作效率带来多大的提升，或者你是一名刚刚开始使用Scala和Clojure的新手，尚不能玩转函数式的问题解决方案，那么本书就是为你而准备的。

《Scala与Clojure函数式编程模》

作者简介

作者简介：

Michael Bevilacqua-Linn

康卡斯特杰出工程师，现为该公司的新一代服务的电力基础设施构建分布式系统。他五年级的时候就拖着一台Apple II GS到自己班为同学们解释循环和变量。其博客地址为<http://mblinn.com/>。

译者简介：

赵震一

一线攻城狮，曾涌入创业洪流，多年未有所成。现居杭州，就职于阿里巴巴。热爱技术，关注互联网与大数据。InfoQ社区编辑，闲暇译一些技术文章，译有《RESTful Web APIs中文版》。

书籍目录

第1章 模式和函数式编程	1
1.1 什么是函数式编程	2
1.2 模式词汇表	4
第2章 TinyWeb：让模式协同工作	7
2.1 TinyWeb简介	7
2.2 采用Java来编写TinyWeb	7
2.3 采用Scala来编写TinyWeb	17
2.4 采用Clojure来编写TinyWeb	25
第3章 替代面向对象模式	35
简介	35
模式1 替代函数式接口	35
模式2 替代承载状态的函数式接口	42
模式3 替代命令模式	48
模式4 替代生成器模式来获得不可变对象	55
模式5 替代迭代器模式	64
模式6 替代模板方法模式	73
模式7 替代策略模式	81
模式8 替代空对象	87
模式9 替代装饰器模式	95
模式10 替代访问者模式	99
模式11 替代依赖注入	113
第4章 函数式模式	121
简介	121
模式12 尾递归模式	121
模式13 相互递归模式	128
模式14 Filter-Map-Reduce模式	136
模式15 操作链模式	139
模式16 函数生成器模式	146
模式17 记忆模式	159
模式18 惰性序列模式	163
模式19 集中的可变性	172
模式20 自定义控制流	180
模式21 领域特定语言	191
第5章 结束语	201
参考文献	202

《Scala与Clojure函数式编程模》

精彩短评

- 1、对于学过scala的人来说，此书就略显冗余。不过结合各个编程模式，让新手熟悉scala和clojure还不错。clojure也有其独特之处，还是值得学习。
- 2、名不副实。
- 3、除了认识了一些名词之外没啥用，我本来就这么写代码的
- 4、其实函数式和面向对象是2种思维方式，以设计模式在函数式里面怎么是怎么做的为切入点，一般吧
- 5、看过一遍第四章，大多例子很简单，也有启发性。第三章需要先熟练设计模式，因为书中很少解释模式的原型和使用场景。
- 6、感觉是写给java程序员的，章节乱七八糟，用词诡异，句子生硬。booooooring
- 7、看这本书之前需要对面向对象的设计模式有所了解，书主要是为面向对象的人了解函数式用的。但如何转变思维，作者并没有写太深。不过作者在字里行间也穿插着提供了一些个人的经验。
- 8、发现clojure要比scala简洁，拖着一个静态类型系统就是麻烦。有好多限制，感觉静态类型的代价太大了。
- 9、从面向对象模式到函数式编程模式，需要对面向对象的设计模式有一定了解
- 10、对于有经验的OOP来说，入门函数式编程的好书。
写程序再也不是把java直译成scala了。

章节试读

1、《Scala与Clojure函数式编程模式》的笔记-第47页

类是穷人的闭包，闭包是穷人的类

2、《Scala与Clojure函数式编程模式》的笔记-第129页

Trampoline style靠 (\$TailCalls\$) 里的 (\$done\$) 和 (\$tailcall\$) 以及 (\$TailRec\$) 来实现一般用的时候就把这些直接import就好了

```
import scala.util.control.TailCalls._
```

done和tailcall的用途就是把计算(expr)包装成一个TailRec对象，done这里会直接把expr eval掉((\$T \rightarrow TailRec[T]\$))，而tailcall是call by name ((\$tailcall : (\rightarrow TailRec[T]) \rightarrow TailRec[T]\$))

(\$TailRec\$)实际上就是一个thunk, 可以用result方法来eval，这个方法每次把TailRec eval一层，如果下一层还是个Call (tailcall包装的对象)就再往下算，如果碰到Done (done包装的value)就直接返回这个值，完成TailRec的eval。这样就成迭代计算了。

一般的直接计算方法是使用系统stack，系统不会区分实际上不需要stack记录的尾递归和一般递归，把尾递归完全展开，每次call占用一个stack frame。而Trampoline style把尾递归变成类似thunk的东西，然后每次在(\$原地\$)展开一层计算，依次往下迭代。这样就将recursion转化成了iteration

3、《Scala与Clojure函数式编程模式》的笔记-第41页

函数式接口，源于java的OO作风。为了完成run这个动作，我们必须定义Runner这样的动词，增加了语法开销。

通过高价函数替代函数式接口

4、《Scala与Clojure函数式编程模式》的笔记-第13页

5、《Scala与Clojure函数式编程模式》的笔记-第19页

在函数式的事件，我们主要采用表达式来编程。而该函数的值正好是函数中最后一个表达式的值

6、《Scala与Clojure函数式编程模式》的笔记-第13页

java实现对象不可变性

```
HttpResponse {  
  public static class Builder {  
    public HttpResponse build() {  
    }  
    public static Builder newBuilder() {  
    }  
  }  
}
```

7、《Scala与Clojure函数式编程模式》的笔记-第124页

尾递归 TCO tail call optimization

JVM不支持TCO

scala 用注解@tailrec转译为迭代

相互递归模式

函数生成器

创建函数

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