

《Clojure程序设计》

图书基本信息

书名：《Clojure程序设计》

13位ISBN编号：9787115308474

10位ISBN编号：7115308470

出版时间：2013-4-1

出版社：人民邮电出版社

作者：哈罗威 (Stuart Halloway), 拜卓 (Aaron Bedra)

页数：253

译者：温瑞云

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

《Clojure程序设计》

内容概要

Clojure是一种基于Java虚拟机的动态编程语言。如果你要转移到函数式编程，或是要应对来自并发的挑战，那么，Clojure是你的首选语言。Clojure具备以下优点：

- 简单，是这门优雅语言的核心；
- 独到的并发机制和函数式风格；
- Lisp的强大；
- 充分发挥JVM的全部能力；
- 代码量往往比Java少，且易于维护。

本书是介绍Clojure的经典之作。它完全覆盖了Clojure 1.3版本提供的新特性，而且为了体现Clojure新生概念的重要意义，作者重新组织和编写了一些章节。

两位作者通过本书展示了如何从头构建一个应用，帮助读者充分了解Clojure程序的完整开发流程，还帮助读者运用Clojure思想为问题的各个部分找出解决方案。

通过阅读本书，你能够学会运用Clojure的方式进行思考，充分利用这门语言的优势，快速构建强大的程序。

《Clojure程序设计》

作者简介

Stuart Halloway Clojure/core成员、Relevance公司CTO。

Aaron Bedra Clojure/core成员，在担任Relevance公司开发者期间，他成为一名技术领袖、演讲家和作家。他对Clojure语言及其支持库贡献颇多，同时也是Clojure社区中的活跃分子。Aaron牵头开发了一些商业的Clojure项目。目前他在Groupon工作。

书籍目录

目录

第1章 启航

1

1.1 为什么是Clojure

2

1.1.1 Clojure非常优雅

2

1.1.2 Clojure是Lisp的再度崛起

5

1.1.3 为什么是Lisp

5

1.1.4 它是Lisp，但括号少了

6

1.1.5 Clojure是函数式语言

8

1.1.6 Clojure简化了并发编程

9

1.1.7 Clojure与Java虚拟机彼此亲密无间

10

1.2 Clojure编程快速入门

11

1.2.1 使用REPL

12

1.2.2 特殊变量

13

1.2.3 添加共享状态

14

1.3 探索Clojure的程序库

16

1.3.1 require和use

17

1.3.2 查找文档

18

1.4 小结

20

第2章 探索Clojure

21

2.1 形式

21

2.1.1 使用数值类型

22

2.1.2 符号

24

2.1.3 字符串与字符

25

2.1.4 布尔值与nil

27

2.1.5 映射表、关键字和记录	28
2.2 读取器宏	30
2.3 函数	32
2.3.1 匿名函数	34
2.3.2 何时使用匿名函数	36
2.4 变量、绑定和命名空间	36
2.4.1 绑定	37
2.4.2 解构	38
2.4.3 命名空间	40
2.5 调用Java	43
2.5.1 访问构造函数、方法和字段	43
2.5.2 Javadoc	45
2.6 流程控制	45
2.6.1 分支结构与if	45
2.6.2 用do引入副作用	46
2.6.3 循环与loop/recur	47
2.7 我的for循环哪儿去了	48
2.8 元数据	52
2.9 小结	53
第3章 一切皆序列	55
3.1 一切皆序列	56
3.2 使用序列库	61
3.2.1 创建序列	61
3.2.2 过滤序列	64
3.2.3 序列谓词	

65
3.2.4 序列转换
66
3.3 惰性和无限序列
69
3.4 Java亦可序化
71
3.4.1 序化Java容器
71
3.4.2 序化正则表达式
73
3.4.3 序化文件系统
74
3.4.4 序化流
75
3.4.5 序化XML
76
3.5 调用特定于结构的函数
77
3.5.1 列表函数
77
3.5.2 向量函数
78
3.5.3 映射表函数
79
3.5.4 集合函数
82
3.6 小结
85
第4章 函数式编程
86
4.1 函数式编程理念
86
4.1.1 纯函数
87
4.1.2 持久性数据结构
87
4.1.3 惰性和递归
88
4.1.4 引用透明性
89
4.1.5 FP的优势
89
4.1.6 6条规则
90
4.2 怎样偷个懒
91
4.2.1 尾递归
92

4.2.2 自递归与recur	94
4.2.3 惰性序列	95
4.2.4 聊聊变现	97
4.2.5 丢弃头元素	98
4.3 懒上加懒	99
4.4 再议递归	105
4.4.1 转换为自递归	106
4.4.2 采用Trampolining技术	107
4.4.3 用惰性化替代递归	109
4.4.4 用快存为递归抄条近路	112
4.5 小结	114
第5章 状态	115
5.1 并发、并行和锁定	116
5.2 引用与软事务内存	117
5.2.1 ref-set	118
5.2.2 事务的属性	118
5.2.3 alter	119
5.2.4 STM的工作原理：MVCC	120
5.2.5 commute	121
5.2.6 alter优先	122
5.2.7 为引用添加验证	123
5.3 使用原子进行非协同、同步的更新	124
5.4 使用代理进行异步更新	125
5.4.1 代理的验证与错误处理	126
5.4.2 在事务中包含代理	

127	
5.4.3 统一的更新模型	
129	
5.5 用变量管理线程内状态	
129	
5.5.1 远距离作用	
131	
5.5.2 用于Java回调API	
132	
5.6 Clojure贪吃蛇	
134	
5.6.1 函数式模型	
135	
5.6.2 使用STM建立可变模型	
139	
5.6.3 贪吃蛇的GUI	
140	
5.6.4 没有引用的贪吃蛇	
143	
5.7 小结	
144	
第6章 协议和数据类型	
145	
6.1 针对抽象编程	
145	
6.2 接口	
148	
6.3 协议	
149	
6.4 数据类型	
153	
6.5 记录	
158	
6.6 具体化	
164	
6.7 小结	
165	
第7章 宏	
166	
7.1 何时使用宏	
166	
7.2 编写流程控制宏	
167	
7.2.1 特殊形式、设计模式和宏	
169	
7.2.2 宏展开	
170	
7.2.3 when与when-not	
172	

7.3 让宏更加简单

173

7.3.1 语法引述、解引述和解引述拼接

175

7.3.2 在宏内部创建名称

176

7.4 宏的分类

178

7.4.1 有条件的求值

180

7.4.2 创建变量

181

7.4.3 Java互操作

183

7.4.4 延迟求值

184

7.4.5 对求值进行包装

184

7.4.6 避免Lambda表达式

186

7.5 小结

187

第8章 多重方法

188

8.1 没有多重方法的日子

188

8.2 定义多重方法

190

8.2.1 调度是继承感知的

192

8.2.2 多重方法的默认值

192

8.3 调度不简单

193

8.4 创建特设分类法

195

8.5 何时应该使用多重方法

199

8.5.1 Inspector

200

8.5.2 clojure.test

201

8.5.3 反例

202

8.6 小结

203

第9章 极尽Java之所能

204

9.1 异常处理

205	
9.1.1 保持简单的异常处理	205
9.1.2 清理资源	206
9.1.3 响应异常	207
9.2 与整数搏斗	209
9.3 性能优化	210
9.3.1 使用基本类型提升性能	210
9.3.2 添加类型提示	213
9.4 在Clojure中创建Java类	215
9.4.1 创建Java代理	215
9.4.2 使用Java容器	217
9.5 真实世界的例子	220
9.5.1 真正的持续循环	222
9.5.2 日志	224
9.5.3 配置	226
9.6 小结	227
第10章 搭建应用	228
10.1 为电码译员游戏计分	229
10.2 测试计分器	232
10.2.1 手工构造输入	233
10.2.2 运行测试	234
10.2.3 验证输出	235
10.2.4 回归测试	236
10.3 test.generative	237
10.3.1 生成数据	237

10.3.2 程序化验证	238
10.3.3 defspec	240
10.3.4 运行测试	241
10.3.5 规格执行失败时	242
10.3.6 计分函数的产生式测试	243
10.4 创建用户界面	244
10.4.1 该是时候处理状态了	245
10.4.2 玩家界面	246
10.5 部署代码	249
10.5.1 Procfile	250
10.5.2 Heroku库	250
10.5.3 Git	251
10.5.4 安置你的应用	251
10.5.5 部署	252
10.6 小结	252
附录 编辑器	254
参考书目	255

章节摘录

版权页：插图：特殊形式有作为程序员的你所不具有的几个特权。特殊形式提供了最基本的流程控制结构，例如if和recur。所有的流程控制宏可能最终都会调用某个特殊形式。特殊形式提供了对Java的直接访问。每当你要从Clojure中调用Java，你至少会通过一个诸如，或new这样的特殊形式来达到目的。名称是通过特殊形式得以创建及被绑定的，无论是通过def定义变量、通过let创建词法绑定、还是通过bindin9创建动态绑定。特殊形式之所以如此强大，是因为它们不是函数。但它们也无法做到一些函数可以做到的事情。你不能apply一个特殊形式，把特殊形式保存到变量中，或是在使用序列库时把特殊形式用作过滤器。简而言之，特殊形式不是语言中的一等公民。特殊形式的特殊性可能是一个重大的问题，并会导致在你的代码中产生重复和难以维护的模式。但是宏却巧妙的解决了这个问题，因为你可以使用宏来生成特殊形式。从实践角度来理解，在宏展开期间，所有的语言特性都是一等特性。生成特殊形式的宏往往是最难编写的，但同时也是最值得一试的。就像被施加了魔法一般，这样的宏就好像是在往语言中添加新的特性。宏俱乐部规则的例外是调用方的便利性：你可以编写任何相较于其等价函数而言，能让调用者的生活变得更加轻松的宏。因为宏并不对它们的参数求值，因此调用者就可以向宏传入原始代码，而不必将这些代码包装到一个匿名函数中。或者，调用者可以传入未转义的名称，而不是被引述的符号或是字符串。我们已经考察了Clojure和分享库中的宏，它们几乎所有都遵循了宏俱乐部的规则。此外，它们还可以被归入到表7—3中的一或多个类别，这个表格展示了Clojure宏的分类法。

《Clojure程序设计》

编辑推荐

《Clojure程序设计》是介绍Clojure的经典之作。它完全覆盖了Clojure 1.3版本提供的新特性，而且为了体现Clojure新生概念的重要意义，作者重新组织和编写了一些章节。两位作者通过《Clojure程序设计》展示了如何从头构建一个应用，帮助读者充分了解Clojure程序的完整开发流程，还帮助读者运用Clojure思想为问题的各个部分找出解决方案。通过阅读《Clojure程序设计》，你能够学会运用Clojure的方式进行思考，充分利用这门语言的优势，快速构建强大的程序。

《Clojure程序设计》

名人推荐

本书仍然提供了令人振奋的Clojure简介，我希望它能激发你加入我们的社区，并最终对Clojure生态系统做出贡献。——Rich Hickey，Clojure之父Clojure是目前最有趣的编程语言之一，同时，作为学习Clojure的最佳途径，本书增加了许多最新的信息和丰富实用的范例，此外还有大量关于如何学习和使用Clojure的提示，这真是太管用了。——Ola Bini Ioke编程语言之父，ThoughtWorks公司开发者被Clojure给吓住啦？在读过本书之后，你就不再会这样了。它文风清晰、平易近人且令人愉悦，带你循序渐进地学习这门语言。——Tim Berglund August Technology Group的创始人及负责人作者凭借这本组织良好且易于阅读的著作，为学习Clojure语言设计了一条最为平坦的路线。本书用简单有效的示例来展示了Clojure独一无二的特性。——Chris Houser Clojure的主要贡献者及程序库作者Clojure是Java虚拟机上一门美丽、优雅且极其强大的编程语言。在本书中，两位经验丰富的开发者帮助你学习和掌握这门语言。你将会乐在其中，并从中获益。——Venkat Subramaniam Agile Developer公司创始人

精彩短评

- 1、还是得拿项目练手
- 2、Clojure中文资料比较少. 这本用来入门很不错. 学习Clojure的时候, 建议先看看 SICP 前两章
- 3、印刷错误太多。感谢译者赠书。
- 4、大家好，我是这本书的译者。说Clojure是目前jvm上最优雅的编程语言毫不为过，其诸多设计思想代表了整个Java世界的发展方向。而本书的重点，就是通过clojure这门语言，传递这些足以改变我们观念的思想。作为一名在面向对象世界摸爬滚打了10多年的程序员，我自己在接受了clojure哲学之后，很多方面都豁然开朗了。现在即便是使用其他语言编写程序，我也会不知不觉地将这些思想运用其中，从而获得更好的代码质量，开发效率也大幅提升。推荐本书给所有希望不断提升编程技艺的同学，相信我，尽管Clojure（可鲁耶？）读起来有点怪异，但本书却一点都不难懂，也许读完前言和第一章之后，你就会发现，一扇通往另一个世界的窗户已经打开了。PS，欢迎各类意见和建议的反馈。
- 5、翻译的中规中矩，作为一本入门的图书很适合我。接下来应该着一个实际的应用场景和深入一点的书。
- 6、内容很棒，入门不错。其他的还得靠自己的。
- 7、一些地方就像API文档.....
- 8、很好的Clojure教程的书，不过最好有点相关基础会更好。
- 9、当Lisp乘上JVM，那将是什么情况
- 10、这书完全就不适合初学者看，还是去刷另外一本Clojure Programming好一点....
- 11、书本的纸质个人感觉还是不错的。。只是身为外行。。内容就不评论什么了
- 12、书很薄，讲的太泛
- 13、内容挺全面的，不过没有Lisp基础的人学起来会有点困难。
- 14、感觉比较凌乱。
- 15、这本书的质量不错，内容不是很深入
- 16、Clojure (Lisp) 入门绝佳
- 17、预购的书今天终于盼到了.人民邮电出版社的书质量真的很好!
- 18、入门，这个是下一步
- 19、初次接触函数式编程，还是很不适应
- 20、还剩两章读完。总的说来，主要的要点都提到了，但是都没有一个特别详细的例子，当然书中的案例选得还是不错的，就是有些细节还是没有说得太明白，另外内容的安排上不是那么的循序渐进。买了本Clojure Programming，准备也读了比较一下。
- 21、虽然有些地方并没有仔细讲解缘由。但是我觉得经过自己仔细地推敲，会对这个“被忽略”点有更深，更好的理解。
- 22、Rich Hickey 推荐怎么会有错。书不厚，但是介绍的应该都是核心内容，不啰嗦。介绍某一种编程语言的书类里应该算是水平很高的。学习clojure，(repeat 推荐). 再者建议看完书后去看Rich Hickey 和 Brian Beckman的expert to expert talk，会有更深入的体会。
- 23、看着这本书入的clojure的门
- 24、没有joy of clojure啰嗦，入门就是应该简洁直接的
- 25、对于入门来说还是不错的，但是感觉条理上，系统性上不够完善，不过，毕竟是本小册子，可以挺快看完，加入实践的行列；
- 26、从基础学起,注重理论和基本应用,是学习一个新软件语言的不二法则
- 27、这是一本侧重于讲述Clojure编程思想的好书。也许它并不全面和深入，但蕴含其中的编程理念对于Clojure初学者来说，是极为可贵的。
- 28、只讲最核心部分，蛮好的
- 29、挺不错的，适合初学者
- 30、clojure编程的基本内容都涵盖了，对于初次接触clojure且没有学过LISP的朋友来说，是最佳的入门教材，语言通俗易懂，没有像一些教程故意夸大难度，也没有忽悠多么易学。翻译的比较符合汉语的阅读习惯，译者看来是下了不少功夫，没有像不少奸商那样用翻译软件直接拷贝过来忽悠读者。美中不足的是本书的编辑不知是专业能力不足，还是实在是太不敬业了，书中示例代码部分错误多多！！

《Clojure程序设计》

！还是根据作者的提示到网上把代码下载来对照着学吧，不然，书不一定会误入歧途，南辕北辙！！
31、章节编排很不好导致阅读时总是需要前后翻页，更不适合未接触过LISP的新手

精彩书评

- 1、很遗憾，我发了两次包含本书勘误的下载地址（ubuntu one的共享地址）的书评，都被豆瓣迅速删掉了，无法理解豆瓣的判断策略。无奈，只能麻烦各位需要下载的同学使用关键字“Clojure程序设计勘误表”进行搜索了，深表歉意。
- 2、被Clojure给吓住啦？在读过本书之后，你就不再会这样了。它文风清晰、平易近人且令人愉悦，带你循序渐进地学习这门语言。——Tim Berglund August Technology Group的创始人及负责人 作者凭借这本组织良好且易于阅读的著作，为学习Clojure语言设计了一条最为平坦的路线。本书用简单有效的示例来展示了Clojure独一无二的特性。——Chris Houser Clojure的主要贡献者及程序库作者 Clojure是Java虚拟机上一门美丽、优雅且极其强大的编程语言。在本书中，两位经验丰富的开发者帮助你学习和掌握这门语言。你将会乐在其中，并从中获益。——Venkat Subramaniam Agile Developer公司创始人
- 3、大家好，我是这本书的译者。说Clojure是目前jvm上最优雅的编程语言毫不为过，其诸多设计思想代表了整个Java世界的发展方向。而本书的重点，就是通过clojure这门语言，传递这些足以改变我们观念的思想。作为一名在面向对象世界摸爬滚打了10多年的程序员，我自己在接受了clojure哲学之后，很多方面都豁然开朗了。现在即便是使用其他语言编写程序，我也会不知不觉地将这些思想运用其中，从而获得更好的代码质量，开发效率也大幅提升。推荐本书给所有希望不断提升编程技艺的同学，相信我，尽管Clojure（可鲁耶？）读起来有点怪异，但本书却一点都不难懂，也许读完前言和第一章之后，你就会发现，一扇通往另一个世界的窗户已经打开了。PS，欢迎各类意见和建议的反馈。
- 4、本书仍然提供了令人振奋的Clojure简介，我希望它能激发你加入我们的社区，并最终对Clojure生态系统做出贡献。——Rich Hickey，Clojure之父 Clojure是目前最有趣的编程语言之一，同时，作为学习Clojure的最佳途径，本书增加了许多最新的信息和丰富实用的范例，此外还有大量关于如何学习和使用Clojure的提示，这真是太管用了。——Ola Bini Ioke编程语言之父，ThoughtWorks公司开发者

章节试读

1、《Clojure程序设计》的笔记-第27页

对于python程序员来说 clojure的布尔值是个大坑

注意了，空列表 空哈希表 空vector什么的统统都不为false

clojure属于starting forth里那种 严格false

2、《Clojure程序设计》的笔记-第26页

有一个str函数的调用似乎有印刷错误, 原文是

```
(str& args)
```

实际应该是

(str & args)吧? 少个空格。不过我不知道clojure对这类格式是否跟c的指针声明那样可以随便靠后还是靠前

3、《Clojure程序设计》的笔记-第13页

UPDATE: 是我前面没看仔细，他定义 hello的时候不是用println而是str
(defn hello [name] (str "hello, " name))

环境是 clojure 1.4.0
使用 lein repl 起的

这里的示例代码跑起来跟文章里的不一样

我想原因应该是

hello 作为一个function在前面的定义中虽然有字符串输出，但是他的调用返回的求值却是nil

这一点大家可以看到

```
(hello "Stu") 的输出是  
hello, Stu  
nil
```

输出的第一行是println的副作用
第二行才是整个表达式的值

所以 *1的结果是nil

所以下面的

```
(str *1 " and " *2)
```

的结果会是
and

与预期的不一样

4、《Clojure程序设计》的笔记-第18页

那个 find-doc 很不错 有了这个 我终于找到所有跟正则有关关系的函数了 以前在网上搜的介绍 clojure 里用正则的都不全

不过有个地方如果改进下会更好
目前 find-doc 是个宏,会在执行时候打印匹配到的

我觉得如果把打印 doc 换成 返回匹配到的函数名会更好,因为我在搜索关于正则的文档时候 输出太多了,在 repl 里滚了好几屏 我又没办法用 less 慢慢看 所以弄个返回列表 让用户自己控制怎么查看具体的函数 应该比较不错

5、《Clojure程序设计》的笔记-第1页

单作为解决大数据计算来说, clojure 无疑是一个合适的选择。个人认为,如果一个公司选择的开发语言太过晦涩难懂,是很不利于项目的持续推进和发展的,后期维护成本太高。clojure 的语法简单,但是理解起来太难。有点类似于古汉语的文言文,字虽少,理解却不易。clojure 的函数式编程思想,作为一个分支去了解 and 掌握对于经验丰富的开发者来说是有益的补充,但是对于初学者来说,不足以作为安身立命,赚钱养家的基础技能。

6、《Clojure程序设计》的笔记-第65页

take-while 和 drop-while
的示例感觉用截取数字段的例子会比较好

比如二者结合起来 截取 90-100 之间的整数

```
(take-while #( < % 100) (drop-while #( < % 90) (iterate inc 1)))
```

再加个 filter 什么的 就是实际应用了

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