

<<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

出版时间：人民邮电出版社

作者：Stuart · Halloway, Aaron · Bedra

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，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<Clojure程序设计>>

内容概要

《Clojure程序设计》内容简介：Clojure是一种基于Java虚拟机的动态编程语言。

它功能强大，支持函数式编程，简化了并发编程，并且能调用Java程序。

正是这些优点，使其赢得了众多程序员的青睐。

《Clojure程序设计》分别介绍了Clojure概览和基本特性、序列和函数式编程、并发编程模型、协议和数据类型、宏、多重方法，以及Clojure对Java的调用。

最后提供了一个完整了解Clojure应用开发全过程的实例。

《Clojure程序设计》适合有不同语言背景而想要学习和了解Clojure编程的程序员阅读。

函数式程序员、Java和C#程序员、Lisp程序员，以及Perl、Python和Ruby程序员，都能够通过阅读

《Clojure程序设计》得到不同程度的收获。

<<Clojure程序设计>>

作者简介

作者：（美国）Stuart Halloway （美国）Aaron Bedra 译者：温瑞云
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。

<<Clojure程序设计>>

书籍目录

第1章启航1.1.1为什么是Clojure2 1.1.1Clojure非常优雅2 1.1.2Clojure是Lisp的再度崛起5 1.1.3为什么是Lisp5 1.1.4它是Lisp，但括号少了6 1.1.5Clojure是函数式语言8 1.1.6Clojure简化了并发编程9 1.1.7Clojure与Java虚拟机彼此亲密无间10 1.2Clojure编程快速入门11 1.2.1使用REPL12 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 1.3.1require和use17 1.3.2查找文档18 1.4小结20 第2章探索Clojure21 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nil27 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Java43 2.5.1访问构造函数、方法和字段43 2.5.2Javadoc45 2.6流程控制45 2.6.1分支结构与if45 2.6.2用do引入副作用46 2.6.3循环与loop/recur47 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 3.4Java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XML76 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 4.1.5FP的优势89 4.1.66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recur94 4.2.3惰性序列95 4.2.4聊聊天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 5.2.1ref—set118 5.2.2事务的属性118 5.2.3alter119 5.2.4STM的工作原理：MVCC120 5.2.5commute121 5.2.6alter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API132 5.6Clojure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GUI140 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 7.2.3when与when—not172 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 7.4.3Java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 8.5.1Inspector200 8.5.2clojure.test201 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 10.3test.generative237 10.3.1生成数据237 10.3.2程序化验证238 10.3.3defspect240 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 10.5.1Procfile250 10.5.2Heroku库250 10.5.3Git251 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

<<Clojure程序设计>>

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binding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

<<Clojure程序设计>>

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:<http://www.tushu007.com>