第一图书网, tushu007.com

<<Ruby元编程>>

图书基本信息

书名：<<Ruby元编程>>

13位ISBN编号：9787560974583

10位ISBN编号：7560974589

出版时间：2012-1-10

出版时间：华中科技大学出版社

作者：[意] Paolo Perrotta

页数：288

译者：廖志刚,陈睿杰

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

第一图书网, tushu007.com

内容概要

《Ruby元编程》以案例形式循序渐进地介绍了Ruby特有的实用编程技巧(元编程)。
通过分析案例、讲解例题、回顾Ruby代码库的实现细节，《Ruby元编程》作者Paolo Perrotta不仅向读者展示了Ruby编程的优势和Ruby特有的解决问题的方式，更详细列出了发挥其优势的技巧和常用的Ruby设计模式。

Ruby创始人松本行弘作序推荐。

作者简介

Paolo Perrotta有超过10年的软件设计和开发经验，发表过上百篇技术文章。
他用过Java、C++、C#编程，最后爱上了Ruby。
他是较早接受敏捷方法的实践者，在Yoox（一家大型互联网时尚公司）从事管理敏捷团队的工作。
他还在欧洲各国教授编程培训课。
目前他和女友（还有一只猫）住在意大利，专心钻研Ruby。

<<Ruby元编程>>

书籍目录

前言
致谢
引言
“元”这个字眼
关于本书
关于你
第1部分 Ruby元编程.
第1章 星期一：对象模型
1. 1 跟Bill在一起的星期一
1. 2 打开类
1. 3 类的真相
1. 4 小测验：缺失的连接线
1. 5 调用一个方法时发生了什么
I. 6 小测验：混乱的模块
1. 7 对象模型小结
第2章 星期二：方法
2. 1 一个重复代码的问题
2. 2 动态方法
2. 3 method—mtssing()方法
2. 4 小测验：狩猎Bug
2. 5 关于method—missing()方法的更多内容
第3章 星期三：代码块
3. 1 怎样度过驼峰日
3. 2 小测验：Ruby的#符号
3. 3 闭包
3. 4 itance eval()
3. 5 可调用对象
3. 6 编写一种领域专属语
3. 7 小测验：更好的领域专属语言
第4章星期四：类定义
4. 1类定义揭秘
4. 2 小测验：Taboo类
4. 3 单件方法
4. 4 Eigenelass
4. 5 小测验：模块的麻烦
4. 6 别名
4. 7 小测验：打破数学规律
第5章 星期五：编写代码的代码
5. 1 带路
5. 2 Kernel#eval
5. 3 小测验：校验过的属性(第一步)
5. 4 小测验：校验过的属性(第二步)
5. 5 小测验：校验过的属性(第二三步)
5. 6 小测验：校验过的属性(第四步)
5. 7 钩子方法

5．8 小测验：校验过的属性(第五步)
第6章 尾声
第2部分 Rails中的元编程.
第7章 AcllyeRecord的设计
7．1 准备旅程
7．2 ActiveRecord的设计
7．3 经验教训
第8章 深入AcllyeRecord
8．1 动态属性
8．2 动态查找器
8．3 经验教训
第9章 安全元编程
9．1 元编程的测试
9．2 化解猴子补丁
9．3 经验教训
第3部分 附录
附录A 常见惯用法
A．1 拟态方法
A．2 空指针保护
A．3 关于方法参数的技巧
A．4 Self Yield
A．5 Symbol#to_proc()方法
附录B 领域专属语言
B．1 关于领域专属语言
B．2 内部和外部领域专属语言
B．3 领域专属语言和元编程
附录C 法术手册
C．1 法术集
附录D参考书目
索引

章节摘录

版权页： 插图： ——可以格式化你的硬盘，更糟糕的是，甚至可以读到你有点肉麻的电子情书。
这种行为被称为代码注入攻击。
防止代码注入 很显然，下一个要问Bill的问题就是，“怎样才能保护我的代码免受代码注入攻击呢？”你也许可以解析所有的字符串代码（第142页），以识别其中可能有危险的操作。
不过，这种方式已经被证明不大有效，因为恶意代码的写法成千上万，想要战胜一个一心想搞破坏的黑客，对你和你的计算机来说都是危险的。
当谈到代码注入时，有些字符串会比其他字符串更安全。
只有从外面输入的字符串才可能包含恶意代码，因此可以简单地限制eval（）只执行那些你自己写的字符串。
当然，说起来容易，追踪字符串的来源可能会出乎意料的困难。
由于上面的原因，一些程序员倾向于完全禁止使用eval（）方法。
有些程序员紧张那些可能出错的东西，因此彻底禁止eval（）方法成了一种非常流行的做法。
“如果没有eval（）方法，则只能根据具体问题来寻找替代方法。
还记得“代码建入”（第148页）中的数组探索者程序么？
你可以使用动态派发（第41页）技术来代替eval（）方法：def explore_array(method,*arguments) [' a ' , ' b ' , ' c '],send(method,*arguments) end. 不过，有时候你还是会怀念eval（）方法的。
例如，最近，这个安全版本的数组探索者要求分开输入方法名和参数。
不幸的是，分开输入这种方式可能会让你的Web界面显得不那么方便。
另外，这个安全版本还不能调用像find（）这样接受块的方法。
如果需要支持块，那么不得不允许在系统中输入任意代码。
在大量使用和完全不使用eval（）之间并没有一个容易找到的“甜点，～。
如果不想完全弃用eval（），Ruby的确也提供了一些能让它更安全的方式。
下面看看这些特性。
污染对象和安全级别 Bill继续关于安全性的话题，介绍了污染对象的概念。
Ruby会自动把不安全的对象——尤其是从外部传入的对象——标记为被污染的。
污染对象包括程序从Web表单、文件和命令行读入的字符串，甚至包括系统变量。
每次从污染字符串运算而来的新字符串，也是被污染的。

媒体关注与评论

Ruby的很多特性继承自其他语言，这些语言包括Lisp、Smalltalk、C、Perl等。
其元编程特性来自于Lisp（以及Smalltalk）。
元编程看起来有点像魔术，功能强大。
但是你必须牢记：能力越大，责任越大。
享受Ruby编程吧。
——松本行弘 Ruby之父

编辑推荐

《Ruby元编程》填补了Ruby语言参考手册和编程案例之间的空白。
它不仅解释了各种元编程的技术，还展示了编写更精练、更优良代码的方法。
不过要事先警告你，熟悉了新方法后，你会难以忍受目前主流的编程方法。

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