

《面向对象设计实践指南：Ruby语言描》

图书基本信息

书名：《面向对象设计实践指南：Ruby语言描述》

13位ISBN编号：9787115332455

10位ISBN编号：7115332452

出版时间：2014-1-1

出版社：人民邮电出版社

作者：梅茨 (Sandi Metz)

页数：206

译者：张雪平,彭晓东

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

《面向对象设计实践指南：Ruby语言描述》

内容概要

《面向对象设计实践指南：ruby语言描述》是对“如何编写更易维护、更易管理、更讨人喜爱且功能更为强大的ruby应用程序”的全面指导。为帮助读者解决ruby代码难以更改和不易扩展的问题，作者在书中运用了多种功能强大和实用的面向对象设计技术，并借助大量简单实用的ruby示例对这些技术进行全面解释。

全书共9章，主要包含的内容有：如何使用面向对象编程技术编写更易于维护和扩展的ruby代码，单个ruby类所应包含的内容，避免将应该保持独立的对象交织在一起，在多个对象之间定义灵活的接口，利用鸭子类型减少编程间接成本，合理运用继承，通过组合构建对象，设计出最划算的测试，解决不良设计的ruby代码所导致的常见问题等。

《面向对象设计实践指南：ruby语言描述》适合所有对面向对象设计和ruby编程语言感兴趣的程序员阅读参考。

《面向对象设计实践指南：Ruby语言描》

作者简介

sandi metz在处理经历不断变化仍然存活的项目方面，sand引metz拥有30年的工作经验。作为杜克大学的一名软件架构师，她每天都坚持编写代码，在杜克大学里，她的团队不断地为许多客户解决大量实际问题。其中，有些客户的应用程序不仅具有规模大和面向对象的特点，而且已演变了15年之久。她全身心投入的工作是用简单实用的方法将有用的软件提取出来。sandi在ruby nation做过演讲，在gotham ruby用户大会上也演讲过多次。本书是她做过的关于 oo设计的谈话汇集之后的必然成果。

书籍目录

第1章 面向对象设计

- 1.1 设计赞歌
 - 1.1.1 设计解决的问题
 - 1.1.2 为何难以更改
 - 1.1.3 实用的设计定义
- 1.2 设计工具
 - 1.2.1 设计原则
 - 1.2.2 设计模式
- 1.3 设计行为
 - 1.3.1 设计失败
 - 1.3.2 设计时机
 - 1.3.3 设计评
- 1.4 面向对象编程简介
 - 1.4.1 过程式语言
 - 1.4.2 面向对象语言
- 1.5 小结

第2章 设计具有单一职责的类

- 2.1 决定类的内容
 - 2.1.1 将方法分组成类
 - 2.1.2 组织代码以便于更改
- 2.2 创建具有单一职责的类
 - 2.2.1 示例程序：自行车和齿轮
 - 2.2.2 为何单一职责原则很重要
 - 2.2.3 确定一个类是否具有单一职责
 - 2.2.4 确定何时做出设计决策
- 2.3 编写拥抱变化的代码
 - 2.3.1 要依赖行为，不依赖数据
 - 2.3.2 全面推行单一职责原则
- 2.4 最后是真实的轮子需求
- 2.5 小结

第3章 管理依赖关系

- 3.1 理解依赖关系
 - 3.1.1 认识依赖关系
 - 3.1.2 对象间的耦合
 - 3.1.3 其他依赖关系
- 3.2 编写松耦合的代码
 - 3.2.1 注入依赖关系
 - 3.2.2 隔离依赖关系
 - 3.2.3 移除参数顺序依赖关系
- 3.3 管理依赖方向
 - 3.3.1 反转依赖关系
 - 3.3.2 选择依赖方向
- 3.4 小结

第4章 创建灵活的接口

- 4.1 理解接口
- 4.2 定义接口
 - 4.2.1 公共接口

- 4.2.2 私有接口
- 4.2.3 职责、依赖关系和接口
- 4.3 找出公共接口
 - 4.3.1 示例程序：自行车旅游公司
 - 4.3.2 构建意图
 - 4.3.3 使用时序图
 - 4.3.4 请询问"要什么"，别告知"如何做"
 - 4.3.5 寻求上下文独立
 - 4.3.6 信任其他对象
 - 4.3.7 使用消息来发现对象
 - 4.3.8 创建基于消息的应用程序
- 4.4 编写能展现其（内在）最好面的代码
 - 4.4.1 创建显式接口
 - 4.4.2 善用其他类的公共接口
 - 4.4.3 避免依赖私有接口
 - 4.4.4 最小化上下文
- 4.5 迪米特法则
 - 4.5.1 定义迪米特法则
 - 4.5.2 违规的后果
 - 4.5.3 避免违规
 - 4.5.4 听从迪米特法则
- 4.6 小结
- 第5章 使用鸭子类型技术降低成本
 - 5.1 理解鸭子类型
 - 5.1.1 鸭子类型概述
 - 5.1.2 让问题复杂些
 - 5.1.3 发现鸭子类型
 - 5.1.4 鸭子类型的后果
 - 5.2 编写依赖于鸭子类型的代码
 - 5.2.1 识别出隐藏的鸭子类型
 - 5.2.2 信任你的鸭子类型
 - 5.2.3 记录好鸭子类型
 - 5.2.4 在鸭子类型之间共享代码
 - 5.2.5 合理选择鸭子类型
 - 5.3 克服对鸭子类型的恐惧
 - 5.3.1 使用静态类型颠覆鸭子类型
 - 5.3.2 静态类型与动态类型
 - 5.3.3 拥抱动态类型
 - 5.4 小结
- 第6章 通过继承获得行为
 - 6.1 理解经典的继承
 - 6.2 弄清使用继承的地方
 - 6.2.1 从一个具体类开始
 - 6.2.2 嵌入多种类型
 - 6.2.3 找出嵌入的类型
 - 6.2.4 选择继承
 - 6.2.5 描绘出继承关系
 - 6.3 误用继承
 - 6.4 找出抽象

- 6.4.1 创建抽象父类
- 6.4.2 提升抽象行为
- 6.4.3 从具体分离出抽象
- 6.4.4 使用模板方法模式
- 6.4.5 实现所有模板方法
- 6.5 管理父类与子类之间的耦合
 - 6.5.1 理解耦合
 - 6.5.2 使用钩子消息解耦子类
- 6.6 小结
- 第7章 使用模块共享角色行为
 - 7.1 理解角色
 - 7.1.1 找出角色
 - 7.1.2 组织职责
 - 7.1.3 删除不必要的依赖关系
 - 7.1.4 编写具体代码
 - 7.1.5 提取抽象
 - 7.1.6 查找方法
 - 7.1.7 继承角色行为
 - 7.2 编写可继承的代码
 - 7.2.1 识别出反模式
 - 7.2.2 坚持抽象
 - 7.2.3 重视契约
 - 7.2.4 使用模板方法模式
 - 7.2.5 预先将类解耦
 - 7.2.6 创建浅层结构
 - 7.3 小结
- 第8章 组合对象
 - 8.1 Parts组合成Bicycle
 - 8.1.1 更新Bicycle类
 - 8.1.2 创建Parts层次结构
 - 8.2 组合成Parts对象
 - 8.2.1 创建Part
 - 8.2.2 让Parts对象更像一个数组
 - 8.3 制造Parts
 - 8.3.1 创建PartsFactory
 - 8.3.2 借助PartsFactory
 - 8.4 组合成Bicycle
 - 8.5 继承和组合的抉择
 - 8.5.1 接受继承带来的后果
 - 8.5.2 接受组合带来的后果
 - 8.5.3 选择关系
 - 8.6 小结
- 第9章 设计最划算的测试
 - 9.1 意图测试
 - 9.1.1 了解测试的意图
 - 9.1.2 了解测试的内容
 - 9.1.3 了解测试的时机
 - 9.1.4 了解测试的方法
 - 9.2 测试输入消息

- 9.2.1 删除未使用的接口
 - 9.2.2 证明公共接口
 - 9.2.3 隔离测试对象
 - 9.2.4 注入使用类的依赖关系
 - 9.2.5 将依赖关系注入成角色
 - 9.3 测试私有方法
 - 9.3.1 在测试过程中忽略私有方法
 - 9.3.2 从测试类里移除私有方法
 - 9.3.3 选择测试私有方法
 - 9.4 测试输出消息
 - 9.4.1 忽略查询消息
 - 9.4.2 证明命令消息
 - 9.5 测试鸭子类型
 - 9.5.1 测试角色
 - 9.5.2 用角色测试验证测试替身
 - 9.6 测试继承代码
 - 9.6.1 指定继承接口
 - 9.6.2 指定子类责任
 - 9.6.3 测试独特行为
 - 9.7 小结
- 后记

《面向对象设计实践指南：Ruby语言描》

精彩短评

- 1、把面向对象讲解得透彻之极
- 2、还是写太少

《面向对象设计实践指南：Ruby语言描》

精彩书评

1、各种术语翻译的让人吐槽无力，感觉像是机器翻译然后人工校验。或者是非计算机专业人员的生硬造词。而且出现了两次错印 181页 和 183页 都印了两次。我的评论时太短了，但是已经足够表达我对本书的看法了。亏我买了两本打算和同事一起学习。下次购买已这种出版了1年，但是没什么人评论的书，一定要慎之又慎啊。

《面向对象设计实践指南：Ruby语言描》

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