

<<敏捷软件开发>>

图书基本信息

书名：<<敏捷软件开发>>

13位ISBN编号：9787115165756

10位ISBN编号：7115165750

出版时间：2008-1

出版时间：人民邮电出版社

作者：马丁

页数：538

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>

前言

要想成为一名优秀的软件开发人员，需要熟练应用编程语言和开发工具，更重要的是能够领悟优美代码背后的原则和前人总结的经验——这正是本书的主题。

本书凝聚了世界级软件开发大师Robert C. Martin数十年软件开发和培训经验，Java版曾荣获计算机图书最高荣誉——Jolt大奖，是广受推崇的经典著作。

自出版以来一直畅销不衰。

不要被书名误导了，本书不是那种以开发过程为主题的敏捷软件开发类图书。

在书中，作者延续了自己一贯的写作风格，让你亲历现场，并用幽默亲切的语言和插图，通过一步步展示来自开发一线的代码，分析各种设计决策及其得失，以清晰、易于理解的方式讲述了真实程序设计中最为基本然而也是最难做到正确应用的原则(包括SRP、LSP、OCP、DIP、ISP等类设计原则，以及多个包设计原则)与设计模式(不限于GoF经典模式，包括许多作者自己的成果)。

本书不仅是一部深入浅出、生动易懂的面向对象原则与设计模式著作，而且还是一部通俗的敏捷方法导引书和快速实用的UML教程。

通过本书你会发现，许多以前看起来非常枯燥费解的概念，忽然间都豁然开朗，变得鲜活生动起来。

<<敏捷软件开发>>

内容概要

本书中，享誉全球的软件开发专家和软件工程师Robert C. Martin深入而生动地使用真实案例讲解了面向对象基本原则、重要的设计模式、UML和敏捷实践等程序员必备的知识。

本书于2003年荣获第13届Jolt大奖，是C++和Java程序员提高自身水平的绝佳教材，也适于用作高校计算机、软件工程专业相关课程的教材或参考书。

作者简介

Robert C.Martin(“ Bob大叔 ”)世界级的软件开发大师，著名软件咨询公司Object Mento，公司的创始人和总裁。

曾经担任C++Report杂志主编多年，也是设计模式和敏捷开发运动的主要倡导者之一。

<<敏捷软件开发>>

书籍目录

第一部分 敏捷开发第1章 敏捷实践1.1 敏捷联盟1.1.1 人和交互重于过程和工具1.1.2 可以工作的软件重于面面俱到的文档1.1.3 客户合作重于合同谈判1.1.4 随时应对变化重于遵循计划1.2 原则1.3 结论1.4 参考文献第2章 极限编程概述2.1 极限编程实践2.1.1 完整团队2.1.2 用户故事2.1.3 短交付周期2.1.4 验收测试2.1.5 结对编程2.1.6 测试驱动开发2.1.7 集体所有权2.1.8 持续集成2.1.9 可持续的开发速度2.1.10 开放的工作空间2.1.11 计划游戏2.1.12 简单设计2.1.13 重构2.1.14 隐喻2.2 结论2.3 参考文献第3章 计划3.1 初始探索3.2 发布计划3.3 迭代计划3.4 定义“完成”3.5 任务计划3.6 迭代3.7 跟踪3.8 结论3.9 参考文献第4章 测试4.1 测试驱动开发4.1.1 测试优先设计的例子4.1.2 测试促使模块之间隔离4.1.3 意外获得的解耦4.2 验收测试4.3 意外获得的构架4.4 结论4.5 参考文献第5章 重构5.1 素数产生程序：一个简单的重构示例5.1.1 单元测试5.1.2 重构5.1.3 最后审视5.2 结论5.3 参考文献第6章 一次编程实践6.1 保龄球比赛6.2 结论

第二部分 敏捷设计第7章 什么是敏捷设计7.1 设计臭味7.1.1 设计臭味——腐化软件的气味7.1.2 僵化性7.1.3 脆弱性7.1.4 顽固性7.1.5 粘滞性7.1.6 不必要的复杂性7.1.7 不必要的重复7.1.8 晦涩性7.2 软件为何会腐化7.3 Copy程序7.3.1 熟悉的场景7.3.2 Copy程序的敏捷设计7.4 结论7.5 参考文献第8章 SRP：单一职责原则8.1 定义职责8.2 分离耦合的职责8.3 持久化8.4 结论8.5 参考文献第9章 OCP：开放 - 封闭原则9.1 OCP概述9.2 Shape应用程序9.2.1 违反OCP9.2.2 遵循OCP9.2.3 预测变化和“贴切的”结构9.2.4 放置吊钩9.2.5 使用抽象获得显式封闭9.2.6 使用“数据驱动”的方法获取封闭性9.3 结论9.4 参考文献第10章 LSP：Liskov替换原则10.1 违反LSP的情形10.1.1 简单例子10.1.2 更微妙的违反情形10.1.3 实际的例子10.2 用提取公共部分的方法代替继承10.3 启发式规则和习惯用法10.4 结论10.5 参考文献第11章 DIP：依赖倒置原则11.1 层次化11.1.1 倒置的接口所有权11.1.2 依赖于抽象11.2 简单的DIP示例11.3 熔炉示例11.4 结论11.5 参考文献第12章 ISP：接口隔离原则12.1 接口污染12.2 分离客户就是分离接口12.3 类接口与对象接口12.3.1 使用委托分离接口12.3.2 使用多重继承分离接口12.4 ATM用户界面的例子12.5 结论12.6 参考文献第13章 C#程序员UML概观13.1 类图13.2 对象图13.3 顺序图13.4 协作图13.5 状态图13.6 结论13.7 参考文献第14章 使用UML14.1 为什么建模14.1.1 为什么构建软件模型14.1.2 编码前应该构建面面俱到的设计吗14.2 有效使用UML14.2.1 与他人交流14.2.2 脉络图14.2.3 项目结束文档14.2.4 要保留的和要丢弃的14.3 迭代式改进14.3.1 行为优先14.3.2 检查结构14.3.3 想象代码14.3.4 图的演化14.4 何时以及如何绘制图示14.4.1 何时要画图，何时不要画图14.4.2 CASE工具14.4.3 那么，文档呢14.5 结论第15章 状态图15.1 基础知识15.1.1 特定事件15.1.2 超状态15.1.3 初始伪状态和结束伪状态15.2 使用FSM图示15.3 结论第16章 对象图16.1 即时快照16.2 主动对象16.3 结论第17章 用例17.1 编写用例17.1.1 备选流程17.1.2 其他东西呢17.2 用例图17.3 结论17.4 参考文献第18章 顺序图18.1 基础知识18.1.1 对象、生命线、消息及其他18.1.2 创建和析构18.1.3 简单循环18.1.4 时机和场合18.2 高级概念18.2.1 循环和条件18.2.2 耗费时间的消息18.2.3 异步消息18.2.4 多线程18.2.5 主动对象18.2.6 向接口发送消息18.3 结论第19章 类图19.1 基础知识19.1.1 类19.1.2 关联19.1.3 继承19.2 类图示例19.3 细节19.3.1 类衍型19.3.2 抽象类19.3.3 属性19.3.4 聚集19.3.5 组合19.3.6 多重性19.3.7 关联衍型19.3.8 内嵌类19.3.9 关联类19.3.10 关联修饰符19.4 结论19.5 参考文献第20章 咖啡的启示20.1 Mark IV型专用咖啡机20.1.1 规格说明书20.1.2 常见的丑陋方案20.1.3 虚构的抽象20.1.4 改进方案20.1.5 实现抽象模型20.1.6 这个设计的好处20.2 面向对象过度设计20.3 参考文献第三部分 薪水支付案例研究第21章 COMMAND模式和ACTIVE OBJECT模式：多功能与多任务21.1 简单的Command21.2 事务21.2.1 实体上解耦和时间上解耦21.2.2 时间上解耦21.3 Undo()方法21.4 ACTIVE OBJECT模式21.5 结论21.6 参考文献第22章 TEMPLATE METHOD模式和STRATEGY模式：继承和委托22.1 TEMPLATE METHOD模式22.1.1 滥用模式22.1.2 冒泡排序22.2 STRATEGY模式22.3 结论22.4 参考文献第23章 FACADE模式和MEDIATOR模式23.1 FACADE模式23.2 MEDIATOR模式23.3 结论23.4 参考文献第24章 SINGLETON模式和MONOSTATE模式24.1 SINGLETON模式24.1.1 SINGLETON模式的好处24.1.2 SINGLETON模式的代价24.1.3 运用SINGLETON模式24.2

MONOSTATE模式24.2.1 MONOSTATE模式的好处24.2.2 MONOSTATE模式的代价24.2.3 运用MONOSTATE模式24.3 结论24.4 参考文献第25章 NULL OBJECT模式25.1 描述25.2 结论25.3 参考文献第26章 薪水支付案例研究：第一次迭代开始26.1 初步的规格说明26.2 基于用例分析26.2.1 增加新员工26.2.2 删除雇员26.2.3 登记考勤卡26.2.4 登记销售凭条26.2.5 登记工会服务费26.2.6 更改雇员明细26.2.7 发薪日26.3 反思：找出底层的抽象26.3.1 雇员支付类别抽象26.3.2 支付时间表抽象26.3.3 支付方式26.3.4 从属关系26.4 结论26.5 参考文献第27章 薪水支付案例研究：实现27.1 事务27.1.1 增加雇员27.1.2 删除雇员27.1.3 考勤卡、销售凭条以及服务费用27.1.4 更改雇员属性27.1.5 犯了什么晕27.1.6 支付雇员薪水27.1.7 支付领月薪的雇员薪水27.1.8 支付钟点工薪水27.2 主程序27.3 数据库27.4 结论27.5 关于本章27.6 参考文献第四部分 打包薪水支付系统第28章 包和组件的设计原则28.1 包和组件28.2 组件的内聚性原则：粒度28.2.1 重用—发布等价原则28.2.2 共同重用原则28.2.3 共同封闭原则28.2.4 组件内聚性总结28.3 组件的耦合性原则：稳定性28.3.1 无环依赖原则28.3.2 稳定依赖原则28.3.3 稳定抽象原则28.4 结论第29章 FACTORY模式29.1 依赖问题29.2 静态类型与动态类型29.3 可替换的工厂29.4 对测试支架使用对象工厂29.5 工厂的重要性29.6 结论29.7 参考文献第30章 薪水支付案例研究：包分析30.1 组件结构和符号30.2 应用CCP30.3 应用REP30.4 耦合和封装30.5 度量30.6 度量薪水支付应用程序30.6.1 对象工厂30.6.2 重新思考内聚的边界30.7 最终的包结构30.8 结论30.9 参考文献第31章 COMPOSITE模式31.1 组合命令31.2 多重性还是非多重性31.3 结论第32章 OBSERVER——演化至模式32.1 数字时钟32.2 OBSERVER模式32.2.1 模型32.2.2 面向对象设计原则的运用32.3 结论32.4 参考文献第33章 ABSTRACT SERVER模式、ADAPTER模式和BRIDGE模式33.1 ABSTRACT SERVER模式33.2 ADAPTER模式33.2.1 类形式的ADAPTER模式33.2.2 调制解调器问题、适配器以及LSP33.3 BRIDGE模式33.4 结论33.5 参考文献第34章 PROXY模式和GATEWAY模式：管理第三方API34.1 PROXY模式34.1.1 实现PROXY模式34.1.2 小结34.2 数据库、中间件以及其他第三方接口34.3 TABLE DATA GATEWAY34.3.1 测试和内存TDG34.3.2 测试DbGateWay34.4 可以用于数据库的其他模式34.5 结论34.6 参考文献第35章 VISITOR模式35.1 VISITOR模式35.2 ACYCLIC VISITOR模式35.3 DECORATOR模式35.4 EXTENSION OBJECT模式35.5 结论35.6 参考文献第36章 STATE模式36.1 嵌套switch/case语句36.1.1 内部作用域的状态变量36.1.2 测试动作36.1.3 代价和收益36.2 迁移表36.2.1 使用表解释36.2.2 代价和收益36.3 STATE模式36.3.1 STATE模式和STRATEGY模式36.3.2 代价和收益36.4 状态机编译器36.4.1 SMC生成的Turnstile.cs以及其他支持文件36.4.2 代价和收益36.5 状态机应用的场合36.5.1 作为GUI中的高层应用策略36.5.2 GUI交互控制器36.5.3 分布式处理36.6 结论36.7 参考文献第37章 薪水支付案例研究：数据库37.1 构建数据库37.2 一个代码设计缺陷37.3 增加雇员37.4 事务37.5 加载Employee对象37.6 还有什么工作第38章 薪水支付系统用户界面：Model-View-Presenter38.1 界面38.2 实现38.3 构建窗口38.4 Payroll窗口38.5 真面目38.6 结论38.7 参考文献附录A 双公司记Rufus公司：“日落”项目Rupert工业公司：“朝晖”项目附录B 什么是软件索引

<<敏捷软件开发>>

编辑推荐

软件开发的不朽经典；生动阐述面向对象原则、敏捷实践、UML和模式；大量C#实战示例，让你亲历现场。

“我最喜爱的技术作家Robert Martin善于通过实践展示技术，让读者能够以自己喜欢的方式逐步理解……请把Bob大叔当作你在敏捷世界里的导师。

”——Chris Sells，.NET资深技术专家。

微软“软件传奇人物” “本书是对敏捷编程和敏捷原则最全面和最有价值的介绍……绝对是所有.NET程序员必读之作。

”——Jesse Liberty，微软资深技术专家，Programming C#作者 要想成为一名优秀的软件开发人员，需要熟练应用编程语言和开发工具，更重要的是能够领悟优美代码背后的原则和前人总结的经验——这正是本书的主题。

本书凝聚了世界级软件开发大师Robert C.Martin数十年软件开发和培训经验。

Java版曾荣获计算机图书最高荣誉——Jolt大奖。

是广受推崇的经典著作，自出版以来一直畅销不衰。

不要被书名误导了。

本书不是那种以开发过程为主题的敏捷软件开发类图书。

在书中，作者延续了自己一贯的写作风格。

让你亲历现场，并用幽默亲切的语言和插图，通过一步步展示来自开发一线的代码，分析各种设计决策及其得失，以清晰、易于理解的方式讲述了真实程序设计中最基本然而也是最难做到正确应用的原则（包括SRP、LSP、OCP、DIP、ISP等类设计原则，以及多个包设计原则）与设计模式（不限于GoF经典模式。

包括许多作者自己的成果）。

本书不仅是一部深入浅出、生动易懂的面向对象原则与设计模式著作。

而且还是一部通俗的敏捷方法导引书和快速实用的LJML教程。

通过本书你会发现，许多以前看起来非常枯燥费解的概念，忽然间都豁然开朗。

变得鲜活生动起来。

C#版与此前的Java版相比，主要的更新包括加强了UML的介绍章节。

使其更加贴近实战；增加了对MVP模式的介绍等。

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>