敏捷软件开发

敏捷软件开发_下载链接1_

著者:马丁

出版者:人民邮电出版社

出版时间:2008-01-01

装帧:

isbn:9787115165756

《敏捷软件开发:原则模式和实践(C#版)》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《敏捷软件开发:原则模式和实践(C#版)》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。

作者介绍:

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。

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 意外获得的解耦合
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OCP
9.2.2 遵循OCP
9.2.3 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UML
14.1 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UML
14.2.1 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Command
21.2 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 增加新雇员
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CCP
30.3 应用REP
30.4 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LSP
33.3 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API
34.1 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 GATEWAY
34.3.1 测试和内存TDG
34.3.2 测试DbGateWay
34.4 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-Presenter
38.1 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
・・・・・・(收起)

敏捷软件开发_下载链接1_

标签

敏捷开发

设计模式

软件工程

C

#编程

计算机

敏捷

面向对象

评论

我还要再看一遍

读起来有一种很过瘾的感觉

与其说这是一本讲敏捷的书，其实真的不如说，这是一本适合面向对象的开发者学习面向对象思维的书

某些模式写得稍稍简单了些，瑕不掩瑜。

启蒙思想的好书

Bob大叔父子合作的书，严重怀疑儿子只是负责翻译了代码，各种程序员必读书籍

觉得写的非常棒，老少皆宜的！

神一般的人写的神一般的书 对于面向对象基础的回归 设计的原则 应用 模式的展现
讲的东西太多了。。 要读一遍两遍三遍

敏捷 + 重构 +
模式。如果对敏捷和重构有些概念，模式很熟练的话，可以不看了，否则的话可以说是让思维进入一个新境界。

Bob大叔经典中经典，靠它上了一个层次

为什么不早点读？为什么不早点读？这么重要的书，为什么不早点读？

实践是检验真理的唯一标准

初识敏捷开发这种方法论

C# 进阶必读

以后也会经常翻出来看看

1.模式与用例图都很用心
2.面向对象的设计并非一蹴而就，而是在需求不断的变更中，对软件进行重构
3.最后一章代码很多，没有细读

不做开发了，几乎不大看了～～

第一本完整看完滴老外滴书，他们描述问题，解决问题的方法很有趣

理论实践结合的非常紧密，不过我没看完先还了，需要点时间准备下认真看……或者看java版也行。

重读这本书

[敏捷软件开发_下载链接1_]

书评

好的技术书籍的标准是通俗易懂；文字精炼；耐读，有吸引力；有思想性。 uncle bob的书写功力有目共睹，而且他的技术修为也绝对无人质疑。因此他写的这本书秉承了他一贯的优势。符合所有好处所具备的条件。所有我们可以毫无内疚的宣称，“这本书是我见过最好的书”。 孟岩作序...

敏捷软件开发提倡测试先行，设计适应要求，迭代式渐进开发。
一、通过用例来确认需求，分析软件行为：针对用例中的事物对象建立合理的类结构；分析用例中类似情形的变化因素，尽量用抽象来统一一类变化，由此建立系统的大致静态结构。在此不需要、也很难确定好系统的最终结构...

摆在面前的是本大部头，原则、模式和实践诠释了全书的内容，单讲模式没有其他书籍规范，单从重构看又不如马丁的重构专业，本书许多知识可见其他书籍，比较典型的是设计模式解析，我装逼般的和花了一周读本书，可想我本人是多么的浮躁，对我来说书中的实践大于思想，我总感觉读...

果然是获奖无数的巨著，条理清晰，把软件工程，设计模式及面向对象开发思想很好的融合到实践中去
对每一个观点，都有手把手的展开例子，不止是教怎么做，更多的篇幅是在介绍为什么要这么做，这么做解决了哪些问题，另外可能会引入什么新的问题 之前的设计模式那些书，重点篇...

很早就想看这本书了。在旧书摊买了本旧版的英文影印的，但最终还是看了新出的c#版的。新版把旧版的代码翻成了c#，在内容上做了一些取舍，增加了uml的相关章节。但是感觉作者c#的功力不够，翻得代码有些问题，有些概念也不清楚。如直接把成员变量暴露出去，在需要时再改成属性，...

看到前面有评论说，此书与敏捷的关系不大，颇有同感。所谓敏捷，那就是代码先写了再说，且看我们是如何做到，这就是读了这本书的感受。
中文版没有把特定的英文缩写在第一次引用时列出来（只能在后面的索引表里找到），让我很不爽，比如DIP和SRP。不过，说到底还是中文看得快...

孟岩为这本书写了一个代序.这个代序很长,有两页半,其中一页半用来讲述孟岩本人和这本书的感情纠葛.
我为大家复述一下这段感人至深的故事.下面孟先生代表孟岩,小doocaubm和Asd代表什么,请您自己判断. 2001年秋天,北京,孟先生那时候已经颇有些成就了,见识也颇有些广泛了,但是他...

各位前辈，小女子我是工科妹子，有一些编程基础，但是工作不是开发或者测试，目前做的是科研管理岗，说白了就是有点偏文职了，毕业2年了，觉悟还是想趁年轻学点技术，现在想往敏捷开发方向发展。请问各位，现在我看这个书合适吗？ 看了大家的评论，貌似这本书很需要一些软件...

之前看过很多本OO和设计模式的书，
收获都不大。这本书是真正让我理解敏捷，理解OO，以及设计模式的书。非常喜欢里面的语言风格，讲解方法，甚至插图和章节开始的引用词。力荐给所有想学习OO和设计模式的人。

书绝对是经典，但是翻译的实在太牵强，还不如去读原版或者注释版。从第一章看起，手头一本注释版的看着虽然慢些，但不至于一句话读好几遍才知道说的是什么，然而这本中文版上的汉字虽然都认识，但连成一句话后却要反复几遍才能知其所云，句与句之间的过渡处理的生硬，读起来一...

根据最近所阅读到的，对断言语义（assert
semantic）感触颇深。断言的实际应用莫过于契约编程，而契约是一种人与人之间社会行为。我说了，你定要做到，你做不到，那就得给予我赔偿。我觉得不妨理解为自省，一种超我自我超越本我的自发行为。我发现自己这块做不到，我就要努力去...

书是好书，但译者水平太差，还没有google翻译的结果好。语法混乱，语句难以理解就不说了，好多地方不知道该怎么翻译了，直接把原书的单词放在句中了事。原书中重要的、斜体字部分在译文中也没有相应的表现。就连敏捷软件开发宣言这种重要的内容，在本书中竟然出现了多个版本。...

第一次写书评，就根据自己看大家的经验来写吧。书的前几章比较清楚的说明了Agile的思想和团队管理的方法方式，但是后面较多的介绍了设计模式，对此部分没有深入阅读，不敢乱下评论，但此部分篇幅过大。因此如果想学习Agile建议看电子书入门或深入的了解Agile的具体措施。 对Agi...

这本书是我见过的讲述敏捷设计、开发书籍中最棒的一本！尤其是前半部分中OOP设计原则的讲述，非常佩服Bob大叔对设计原则的总结。后半部分感觉涉及到细节太繁琐了就没看完，不过这无损于这本名著的光芒！这本书可以和其它讲述设计模式的相关书籍一起阅读，相得益彰。 读书笔记...

这本书的书名说的很准确，它讲的是一种思想，一种开发过程，这个过程中需要注意的原则，会用到的模式。
也许没有人能一遍就完全看懂这本书，所以，我的评价是，这本书值得你对无数遍，知道你了解其中所有的细节...

通过这本书，你可以有以下收获： 1.更深入的理解模式。 2.提供了更好的软件开发的方法。
3.具有了总体理解系统架构的能力。
我以前总想看懂DELPHI的源码，总觉得一头雾水，现在知道是我没明白他的设计思想，不能从上往下看，越看东西越多就糊涂啦。

介绍面向对象设计原则、设计模式、包原则等方面的内容写的非常好，挺容易理解的。比其他类型的书讲的更透。后面介绍项目的就非常晦涩难懂了。也不知道是作者说的不清楚还是翻译的问题，亦或者是我水平的问题，反正看的非常焦躁。有些话感觉是个人都看不懂。 介绍面向对象设计原...

帮助理解设计原则，例子不错，比很多设计模式的书好理解很多，有例子代码对比，容易理解为何这样设计，解决知其然而不知其所以然的问题。
计划多读几遍，充分理解变成自己的习惯。10多年前打印过，一直未认真读，很遗憾啊。觉得国内软件水平落后10年啊，发现最近几年开源流行，...

首先我以个人开发者的角度来评论这本书，因个人经历所限，并未有大型团队协作，多人并行开发的经历，所以我比较关注的地方在于如何能适应需求变更，快速高质量的满足客户需求。我想每个开发者都应该有感受，需求是不断不断变化的，特别现在互联网时代，客户很可能都不知道自己...

敏捷软件开发_下载链接1_