<<代码整洁之道>>

图书基本信息

书名：<<代码整洁之道>>

13位ISBN编号：9787115216878

10位ISBN编号：7115216878

出版时间：2010-1-1

出版时间：人民邮电出版社

作者：[美]Robert C. Martin

页数：388

译者：韩磊

前言

乐嚼（Ga.J01）是在丹麦最受欢迎的糖果品种之一，它浓郁的甘草味道，完美地弥补了此地潮湿且时常寒冷的天气。
对于我们这些丹麦人，乐嚼的妙处还在于包装盒顶上印制的哲言慧语。
今早我买了一包两件装，在其包装盒上发现这句丹麦谚语：“小处诚实非小事。
”这句话正好是我想在这里说的。
以小见大。
本书写到了一些价值殊胜的小主题。
神在细节之中，建筑师（路德维希·密斯·范·德·罗）如是说。
这句话引发了有关软件开发、特别是敏捷软件开发中架构所处地位的若干争论。
鲍勃（Bob）2和我时常发现自己沉湎于此类对话中。
没错，LudwigmiesVanderRohe的确专注于效用和基于宏伟架构之上的永恒建筑形式。
然而，他也为自己设计的每所房屋挑选每个门把手。
为什么？
因为小处见大。
就TDD3话题展开目前仍在继续的“辩论”时，鲍勃和我认识到，我们均同意软件架构在开发中占据重要地位，但就其确切意义而言，我们之间还有分歧。
然而，这种矛与盾孰利的讨论相对而言并不重要，因为在项目开始之时，我们理所当然应该让专业人士投入些许时间去思考及规划。
20世纪90年代末期有关仅以测试和代码驱动设计的概念已一去不返。
相对于任何宏伟愿景，对细节的关注甚至是更为关键的专业性基础。
首先，开发者通过小型实践获得可用于大型实践的技能和信用度。
其次，宏大建筑中最细小的部分，比如关不紧的门、有点儿没铺平的地板，甚至是凌乱的桌面，都会将整个大局的魅力毁灭殆尽。
这就是整洁代码之所系。
架构只是软件开发用到的借喻之一，主要用在那种等同于建筑师交付毛坯房一般交付初始软件产品的场合。
在Serum和敏捷（Agile）的日子里，人们关注的是快速将产品推向市场。
我们要求工厂全速运转、生产软件。
这就是人类工厂：懂思考、会感受的编码人，他们由产品备忘或用户故事开始创造产品。
来自制造业的借喻在这种场合大行其道。
例如，Serum就从装配线式的日本汽车生产方式中获益良多。

<<代码整洁之道>>

内容概要

软件质量，不但依赖于架构及项目管理，而且与代码质量紧密相关。
这一点，无论是敏捷开发流派还是传统开发流派，都不得不承认。
本书提出一种观念：代码质量与其整洁度成正比。
干净的代码，既在质量上较为可靠，也为后期维护、升级奠定了良好基础。
作为编程领域的佼佼者，本书作者给出了一系列行之有效的整洁代码操作实践。
这些实践在本书中体现为一条条规则(或称“启示”)，并辅以来自现实项目的正、反两面的范例。
只要遵循这些规则，就能编写出干净的代码，从而有效提升代码质量。
本书阅读对象为一切有志于改善代码质量的程序员及技术经理。
书中介绍的规则均来自作者多年的实践经验，涵盖从命名到重构的多个编程方面，虽为一“家”之言，然诚有可资借鉴的价值。

作者简介

作者：(美国)马丁(Robert C. Martin) 译者：韩磊Robert C. Martin，是软件工程领域的大师级人物，是《敏捷软件开发：原则、模式与实践》、《敏捷软件开发：原则、模式与实践(C#版)》（邮电）、《极限编程实践》（邮电）等国内引进的畅销书的作者，其中第一本原著荣获美国《软件开发》第13届震憾（Jolt）大奖，Martin的敏捷系列书是软件工程界的权威书籍。
本书是他的又一最新力作。
Martin在书中对代码具有革命性的解读阐述了整洁代码的最佳敏捷实践的方法书中介绍规则均来自Martin多年的经验，拥有很高的借鉴价值韩磊，互联网产品与运营专家，技术书籍著译者。
曾在全球著名的IT中文社区CSDN及《程序员》杂志任副总经理、总编辑等职。
现居广州。
译著有《梦断代码》和《C#编程风格》。
与刘韧合著《网络媒体教程》，与戴飞合译《BeginningC#Objects中文版：概念到代码》。

<<代码整洁之道>>

书籍目录

章节摘录

插图：这也意味着函数不应该大到足以容纳嵌套结构。
所以，函数的缩进层级不该多于一层或两层。
当然，这样的函数易于阅读和理解。
代码清单3－1显然想做好几件事。
它创建缓冲区、获取页面、搜索继承下来的页面、渲染路径、添加神秘的字符串、生成HTML，如此等等。
代码清单3－1手忙脚乱。
而代码清单3－3则只做一件简单的事。
它将设置和拆解包纳到测试页面中。
过去30年以来，以下建议以不同形式一再出现：函数应该做一件事。
做好这件事。
只做这一件事。
问题在于很难知道那件该做的事是什么。
代码清单3.3只做了一件事，对吧？
其实也很容易看作是三件事：（1）判断是否为测试页面；（2）如果是，则容纳进设置和分拆步骤；（3）渲染成HTML。
如果函数只是做了该函数名下同一抽象层上的步骤，则函数还是只做了一件事。
编写函数毕竟是为了把大一些的概念（换言之，函数的名称）拆分为另一抽象层上的一系列步骤。
代码清单3.1明显包括了处于多个不同抽象层级的步骤。
显然，它所做的不止一件事。
即便是代码清单3-2也有两个抽象层，这已被我们将其缩短的能力所证明。
然而，很难再将代码清单3.3做有意义的缩短。
可以将if语句拆出来做一个名为include Setup And Teardonws Ifrestpage的函数，但那只是重新诠释代码，并未改变抽象层级。
所以，要判断函数是否不止做了一件事，还有一个方法，就是看是否能再拆出一个函数，该函数不仅只是单纯地重新诠释其实现。

<<代码整洁之道>>

编辑推荐

《代码整洁之道》：细节之中自有天地，整洁成就卓越代码尽管糟糕的代码也能运行，但如果代码不整洁，会使整个开发团队泥足深陷，写得不好的代码每年都要耗费难以计数的时间和资源。
然而这种情况并非无法避免。
著名软件专家RoberfC.Marlin在《代码整洁之道》中为你呈现出了革命性的视野。
Martin携同ObjectMetltor公司的同事，从他们有关整洁代码的最佳敏捷实践中提炼出软件技艺的价值观，以飨读者，让你成为更优秀的程序员——只要你着手研读《代码整洁之道》。
阅读《代码整洁之道》需要你做些什么呢？
你将阅读代码——大量代码。
《代码整洁之道》促使你思考代码中何谓正确，何谓错误。
更重要的是，《代码整洁之道》将促使你重新评估自己的专业价值观，以及对自己技艺的承诺。
从《代码整洁之道》中可以学到：好代码和糟糕的代码之间的区别：如何编写好代码，如何将糟糕的代码转化为好代码：如何创建好名称、好函数、好对象和好类；如何格式化代码以实现其可读性的最大化：如何在不妨碍代码逻辑的前提下充分实现错误处理；如何进行单元测试和测试驱动开发。

版权说明