

<<重构>>

图书基本信息

书名：<<重构>>

13位ISBN编号：9787121134500

10位ISBN编号：7121134500

出版时间：2011-6

出版时间：电子工业出版社

作者：Martin Fowler Kent Beck John Brant William Opdyke Don Roberts

页数：425

译者：张逸 评注

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>

<<重构>>

内容概要

重构，一言以蔽之，就是在不改变外部行为的前提下，有条不紊地改善代码。多年前，正是本书原版的出版，使重构终于从编程高手们的小圈子走出，成为众多普通程序员日常开发工作中不可或缺的一部分。本书也因此成为与《设计模式》齐名的经典著作，被译为中、德、俄、日等众多语言，在世界范围内畅销不衰。

本书凝聚了软件开发社区专家多年摸索而获得的宝贵经验，拥有不因时光流逝而磨灭的价值。今天，无论是重构本身，业界对重构的理解，还是开发工具对重构的支持力度，都与本书最初出版时不可同日而语，但书中所蕴涵的意味和精华，依然值得反复咀嚼，而且往往能够常读常新。

本评注版力邀国内资深专家执笔，在英文原著基础上增加中文点评与注释，旨在以先行者的学研心得与实践感悟，对读者阅读与学习加以点拨、指明捷径。

<<重构>>

作者简介

Martin

Fowler世界级软件开发大师，在面向对象分析设计、UML、模式、XP和重构等领域都有卓越贡献，曾任著名软件开发咨询公司ThoughtWorks的首席科学家。

他的多部著作《分析模式》、《UML精粹》和《企业应用架构模式》等都已经成为脍炙人口的经典。

<<重构>>

书籍目录

- Chapter 1 Refactoring, a First Example (新增批注共21条)
 - The Starting Point
 - The First Step in Refactoring
 - Decomposing and Redistributing the Statement Method
 - Replacing the Conditional Logic on Price Code with Polymorphism
 - Final Thoughts
- Chapter 2 Principles in Refactoring (新增批注共10条)
 - Defining Refactoring
 - Why Should You Refactor?
 - When Should You Refactor?
 - What DO I Tell My Manager?
 - Problems with Refactoring
 - Refactoring and Design
 - Refactoring and Performance
 - Where Did Refactoring Come From?
- Chapter 3 Bad Smells in Code (by Kent Beck and Martin Fowler) (新增批注共17条)
 - Duplicated Code
 - Long Method
 - Large Class
 - Long Parameter List
 - Divergent Change
 - Shotgun Surgery
 - Feature Envy
 - Data Clumps
 - Primitive Obsession
 - Switch Statements
 - Parallel Inheritance Hierarchies
 - Lazy Class
 - Speculative Generality
 - Temporary Field
 - Message Chains
 - Middle Man
 - Inappropriate Intimacy
 - Alternative Classes with Different Interfaces
 - Incomplete Library Class
 - Data Class
 - Refused Bequest
 - Comments
- Chapter 4 Building Tests (新增批注共2条)
 - The Value of Self . testing Code
 - The JUnit Testing Framework
 - Adding More Tests
- Chapter 5 Toward a Catalog of Refactorings (新增批注共1条)

<<重构>>

Format of the Refactorings
Finding References
How Mature Are These Refactorings?
Chapter 6 Composing Methods (新增批注共7条)
Extract Method110
Inline Method
Inline Temp
Replace Temp with Query
Introduce Explaining Variable
Split Temporary Variable
Remove Assignments to Parameters
Replace Method with Method Object
Substitute Algorithm
Chapter 7 Moving Features Between Objects (新增批注共11条)
Move Method
Move Field
Extract Class
Inline Class
Hide Delegate
Remove Middle Man
Introduce Foreign Method
Introduce Local Extension
Chapter 8 Organizing Data (新增批注共9条)
Self Encapsulate Field
Replace Data Value with Object
Change Value to Reference
Change Reference to Value
Replace Array with Object
Duplicate Observed Data
Change Unidirectional Association to Bidirectional
Change Bidirectional Association to Unidirectional
Replace Magic Number with Symbolic Constant
Encapsulate Field
Encapsulate Collection
Replace Record with Data Class
Replace Type Code with Class
Replace Type Code with Subclasses
Replace Type Code with StateStrategy
Replace Subclass with Fields
Chapter 9 Simplifying Conditional Expressions (新增批注共6条)
Decompose Conditional
Consolidate Conditional Expression
Consolidate Duplicate Conditional Fragments
Remove Contr01 Flag
Replace Nested Conditional with Guard Clauses
Replace Conditional with Polymorphism
Introduce Null Object

<<重构>>

Introduce Assertion
Chapter 10 Making Method Calls Simpler (新增批注共13条)
Rename Method
Add Parameter
Remove Parameter
Separate Query from Modifier
Parameterize Method
Replace Parameter with Explicit Methods
Preserve Whole Object
Replace Parameter with Method
Introduce Parameter Object
Remove Setting Method
Hide Method
Replace Constructor with Factory Method
Encapsulate Downcast
Replace Error Code with Exception
Replace Exception With Test
Chapter 11 Dealing with Generalization (新增批注共6条)
Pull Up Field
Pull Up Method
Pull Up Constructor Body
Push Down Method
Push Down Field
Extract Subclass
Extract Superclass
Extract Interface
Collapse Hierarchy
Form Template Method
Replace Inheritance with Delegation
Replace Delegation with Inheritance
Chapter 12 Big Refactorings (By Kent Beck and Martin Fowler) (新增批注共11条)
Tease Apart Inheritance
Convert Procedural Design to Objects
Separate Domain from Presentation
Extract Hierarchy
Chapter 13 Refactoring, Reuse, and Reality (by William Opdyke) (新增批注共8条)
A Reality Check
Why Are Developers Reluctant to Refactor Their Programs?
A Reality Check (Revisited)
Resources and References for Refactoring
Implications Regarding Software Reuse and Technology
Transfer
A Final Note
References

<<重构>>

Chapter 14 Refactoring Tools (by Don Roberts and John Brant)

Refactoring with a Tool

Technical Criteria for a Refactoring Tool

Practical Criteria for a Refactoring Tool

Wrap Up

Chapter 15 Putting It All Together (by Kent Beck) (新增批注共2条)

References

List of Soundbites

Index

<<重构>>

编辑推荐

Martin Fowler和本书另几位作者清楚揭示了重构过程，他们为面向对象软件开发所做的贡献难以衡量。

《重构——改善既有代码的设计(评注版)》解释了重构的原理和最佳实践，并指出何时何地你应该开始挖掘你的代码以求改善。

本书的核心是一系列完整的重构方法，其中每一项都介绍一种经过实践检验的代码变换手法的动机和技术。

某些项目如Extract Method和Move Field看起来可能很浅显，但不要掉以轻心，因为理解这类技术正是有条不紊地进行重构的关键。

本书所提的这些重构手法将帮助你一次一小步地修改你的代码，这就减少了过程中的风险。

很快你就会把这些重构手法和其名称加入自己的开发词典中，并且朗朗上口。

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>