

Refactoring: Improving the Design of Existing Code

Foreword

Preface

What Is Refactoring?

What's in This Book?

Who Should Read This Book?

Building on the Foundations Laid by Others

Acknowledgments

1. Refactoring, a First Example

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

2. Principles in Refactoring

Defining Refactoring

Why Should You Refactor?

Refactoring Helps You Find Bugs

When Should You Refactor?

What Do I Tell My Manager?

Problems with Refactoring

Refactoring and Design

Refactoring and Performance

Where Did Refactoring Come From?

3. Bad Smells in Code

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

4. Building Tests

The Value of Self-testing Code
The JUnit Testing Framework
Adding More Tests

5. Toward a Catalog of Refactorings

Format of the Refactorings
Finding References
How Mature Are These Refactorings?

6. Composing Methods

Extract Method
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

7. Moving Features Between Objects

Move Method
Move Field
Extract Class
Inline Class
Hide Delegate
Remove Middle Man
Introduce Foreign Method
Introduce Local Extension

8. Organizing Data

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 State/Strategy
Replace Subclass with Fields

9. Simplifying Conditional Expressions

Decompose Conditional
Consolidate Conditional Expression
Consolidate Duplicate Conditional Fragments
Remove Control Flag

[Replace Nested Conditional with Guard Clauses](#)
[Replace Conditional with Polymorphism](#)
[Introduce Null Object](#)
[Introduce Assertion](#)

[10. Making Method Calls Simpler](#)

[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](#)

[11. Dealing with Generalization](#)

[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](#)

[12. Big Refactorings](#)

[12. Big Refactorings](#)
[Tease Apart Inheritance](#)
[Convert Procedural Design to Objects](#)
[Separate Domain from Presentation](#)
[Extract Hierarchy](#)

[13. Refactoring, Reuse, and Reality](#)

[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](#)
[Endnotes](#)

[14. Refactoring Tools](#)

[Refactoring with a Tool](#)
[Technical Criteria for a Refactoring Tool](#)

Practical Criteria for a Refactoring Tool
Wrap Up

15. Putting It All Together

Bibliography
Bibliography

--