

The Addison-Wesley Signature Series

*"Any fool can write code that a computer can understand.
Good programmers write code that humans can understand."*

—M. Fowler (1999)



REFACTORING

Improving the Design of Existing Code

Martin Fowler

with contributions by
Kent Beck



SECOND EDITION



List of Refactorings

Change Function Declaration (124)
Change Reference to Value (252)
Change Value to Reference (256)
Collapse Hierarchy (380)
Combine Functions into Class (144)
Combine Functions into Transform (149)
Consolidate Conditional Expression (263)
Decompose Conditional (260)
Encapsulate Collection (170)
Encapsulate Record (162)
Encapsulate Variable (132)
Extract Class (182)
Extract Function (106)
Extract Superclass (375)
Extract Variable (119)
Hide Delegate (189)
Inline Class (186)
Inline Function (115)
Inline Variable (123)
Introduce Assertion (302)
Introduce Parameter Object (140)
Introduce Special Case (289)
Move Field (207)
Move Function (198)
Move Statements into Function (213)
Move Statements to Callers (217)
Parameterize Function (310)
Preserve Whole Object (319)
Pull Up Constructor Body (355)
Pull Up Field (353)
Pull Up Method (350)
Push Down Field (361)
Push Down Method (359)
Remove Dead Code (237)
Remove Flag Argument (314)
Remove Middle Man (192)
Remove Setting Method (331)
Remove Subclass (369)
Rename Field (244)
Rename Variable (137)
Replace Command with Function (344)
Replace Conditional with Polymorphism (272)
Replace Constructor with Factory Function (334)
Replace Derived Variable with Query (248)
Replace Function with Command (337)
Replace Inline Code with Function Call (222)
Replace Loop with Pipeline (231)
Replace Nested Conditional with Guard Clauses (266)
Replace Parameter with Query (324)
Replace Primitive with Object (174)
Replace Query with Parameter (327)
Replace Subclass with Delegate (381)
Replace Superclass with Delegate (399)
Replace Temp with Query (178)
Replace Type Code with Subclasses (362)
Separate Query from Modifier (306)
Slide Statements (223)
Split Loop (227)
Split Phase (154)
Split Variable (240)
Substitute Algorithm (195)

Refactoring: Improving the Design of Existing Code

Table of Contents

Cover

Title Page

Copyright Page

Dedication

Contents

Foreword to the First Edition

Preface

Chapter 1: Refactoring: A First Example

 The Starting Point

 Comments on the Starting Program

 The First Step in Refactoring

 Decomposing the statement Function

 Status: Lots of Nested Functions

 Splitting the Phases of Calculation and Formatting

 Status: Separated into Two Files (and Phases)

 Reorganizing the Calculations by Type

 Status: Creating the Data with the Polymorphic Calculator

 Final Thoughts

Chapter 2: Principles in Refactoring

 Defining Refactoring

 The Two Hats

Table of Contents

Why Should We Refactor?

When Should We Refactor?

Problems with Refactoring

Refactoring, Architecture, and Yagni

Refactoring and the Wider Software Development Process

Refactoring and Performance

Where Did Refactoring Come From?

Automated Refactorings

Going Further

Chapter 3: Bad Smells in Code

Mysterious Name

Duplicated Code

Long Function

Long Parameter List

Global Data

Mutable Data

Divergent Change

Shotgun Surgery

Feature Envy

Data Clumps

Primitive Obsession

Repeated Switches

Loops

Lazy Element

Speculative Generality

Temporary Field

Table of Contents

Message Chains

Middle Man

Insider Trading

Large Class

Alternative Classes with Different Interfaces

Data Class

Refused Bequest

Comments

Chapter 4: Building Tests

The Value of Self-Testing Code

Sample Code to Test

A First Test

Add Another Test

Modifying the Fixture

Probing the Boundaries

Much More Than This

Chapter 5: Introducing the Catalog

Format of the Refactorings

The Choice of Refactorings

Chapter 6: A First Set of Refactorings

Extract Function

Inline Function

Extract Variable

Inline Variable

Change Function Declaration

Encapsulate Variable

Table of Contents

Rename Variable

Introduce Parameter Object

Combine Functions into Class

Combine Functions into Transform

Split Phase

Chapter 7: Encapsulation

Encapsulate Record

Encapsulate Collection

Replace Primitive with Object

Replace Temp with Query

Extract Class

Inline Class

Hide Delegate

Remove Middle Man

Substitute Algorithm

Chapter 8: Moving Features

Move Function

Move Field

Move Statements into Function

Move Statements to Callers

Replace Inline Code with Function Call

Slide Statements

Split Loop

Replace Loop with Pipeline

Remove Dead Code

Chapter 9: Organizing Data

Table of Contents

Split Variable

Rename Field

Replace Derived Variable with Query

Change Reference to Value

Change Value to Reference

Chapter 10: Simplifying Conditional Logic

Decompose Conditional

Consolidate Conditional Expression

Replace Nested Conditional with Guard Clauses

Replace Conditional with Polymorphism

Introduce Special Case

Introduce Assertion

Chapter 11: Refactoring APIs

Separate Query from Modifier

Parameterize Function

Remove Flag Argument

Preserve Whole Object

Replace Parameter with Query

Replace Query with Parameter

Remove Setting Method

Replace Constructor with Factory Function

Replace Function with Command

Replace Command with Function

Chapter 12: Dealing with Inheritance

Pull Up Method

Pull Up Field

Table of Contents

Pull Up Constructor Body

Push Down Method

Push Down Field

Replace Type Code with Subclasses

Remove Subclass

Extract Superclass

Collapse Hierarchy

Replace Subclass with Delegate

Replace Superclass with Delegate

Bibliography

Index