

Refactoring

Ridi Ferdiana | ridi@acm.org

Version 1.0.0



Agenda

- What is Refactoring?
- How to implement Refactoring
- How to do Refactoring in Visual Studio?

What is Refactoring



What is Refactoring

- A change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behavior

Refactoring Happens

In larger systems (frameworks), code will be read and modified more frequently than it will be written

Refactoring typically involves

Removing duplicated or dead code

Simplifying complex code

Clarifying unclear code

It turns out that removing code can actually be a good thing: your system has fewer lines of code

Easier to develop, maintain, and change

Refactoring can be risky, unless you do it in small steps and have automated tests that can be run anytime

How Implement Refactoring



Refactoring Methods

Encapsulate Field (do this virtually always)

- add getters and/or setters to access a field

Rename Method (often)

Extract SuperClass (recall Shape and Fruit)

- You have two classes with similar features

- Create a superclass and move common features to the superclass

Compose Method

- A variety of refactorings can happen at the method level

Catalog

- | | |
|--|--|
| <ul style="list-style-type: none">1. Add Parameter2. Collapse Hierarchy3. Consolidate Conditional Expression4. Consolidate Duplicate Conditional Fragments5. Decompose Conditional6. Encapsulate Collection7. Encapsulate Field8. Extract Class9. Extract Method10. Extract Superclass11. Hide Method12. Inline Method13. Parameterize Method Pull up Field14. Pull Up Method | <ul style="list-style-type: none">15. Remove Double Negative16. Replace Assignment with Initialization17. Replace Conditional with Polymorphism18. Replace Inheritance with Delegation19. Replace Iteration with Recursion20. Remove Control Flag21. Replace Error Code with Exception22. Replace Exception with Test23. Replace Magic Number with Symbolic Constant24. Replace Nested Conditional with Guard Clauses25. Replace Recursion with Iteration26. Reverse Conditional27. Separate Query from Modifier |
|--|--|

Catalog

<https://www.refactoring.com/catalog/>

Using the Catalog ►

Tags

- ☐ basic
- ☐ encapsulation
- ☐ moving-features
- ☐ organizing-data
- ☐ simplify-conditional-logic
- ☐ refactoring-apis
- ☐ dealing-with-inheritance
- ☐ collections
- ☐ delegation
- ☐ errors
- ☐ extract
- ☐ parameters
- ☐ fragments
- ☐ grouping-function
- ☐ immutability
- ☐ inline
- ☐ remove
- ☐ rename
- ☐ split-phase
- ☐ variables

#

Change Function Declaration

Add Parameter • Change Signature • Remove Parameter • Rename Function • Rename Method

Change Reference to Value

Change Value to Reference

Collapse Hierarchy

Combine Functions into Class

Combine Functions into Transform

Consolidate Conditional Expression

Decompose Conditional

Encapsulate Collection

Remove Dead Code

Remove Flag Argument

Replace Parameter with Explicit Methods

Remove Middle Man

Remove Setting Method

Remove Subclass

Replace Subclass with Fields

Rename Field

Rename Variable

Replace Command with Function

Replace Conditional with Polymorphism

Refactoring in Visual Studio



Demo

Refactoring di Visual Studio 2019