Ted M. Young
Java Technical Coach, & Live Coder

ted@tedmyoung.com
https://ted.dev/about

Clear Up Messy Code

with Refactoring Maneuvers in IntelliJ IDEA

(The Art of Refactoring)

Ted M. Young

ted@tedmyoung.com

I Can Help Your Team…

Write more Testable code
with more Effective tests

Be more productive in
Java & **Spring**

Effectively use
TDD

Refactor Messy Code

IntelliJ IDEA and ...

The Art of Refactoring

Refactoring is an Art

But we do it for a reason

Refactoring's essence is applying a series of small, behavior-preserving transformations

— Martin Fowler

Refactoring to Improve

Maintainability, Readability, Understandability

Refactoring to Learn

Get to know the codebase

Evolutionary Design

Less up-front design: don't start perfect

Evolutionary Design

Small, behavior-preserving changes get you closer

Continuous Refactoring Keeps Cost of Change Low

Cost can even go DOWN over time

Small, frequent refactorings > **Big refactorings done rarely**

Break Big Refactorings into Small, Safe Steps*

** Many More, Much Smaller, (Safer) Steps – GeePaw Hill*

Don't Refactor without a Goal, a Reason...Ask

How Will the Code Improve?

How Will You Know It's Better?

Skilled Evolutionary Design Requires Continuous Refactoring

Refactoring Confidence

Practice is Key

Make an LLM Generate "Bad Code" and Refactor it!

Explore and Experiment

Refactoring Discovery

Refactor Goal

Fix Primitive Obsession

Let's Go to the IDE...

Leave a Breadcrumb for Scaffolding

@Deprecated // goal = remove

Automated Refactoring

Ctrl-Opt-N:
Create New Class

Automated Refactoring

Cmd-N:

Create Constructor

Context-Sensitive Action Menu

Opt-Enter: Context Action

Selection

Opt-⬆/⬇ : Extend/Shrink Selection

Multi-Caret

Opt-Opt (hold): Clone Caret

Navigation

Cmd-B: Find Usages

(or Cmd-Opt-F7)

Navigation Shortcut

F2: Jump to Next Error

Stay in the GREEN

Ctrl-R: Run Tests

(compiles as side-effect)

Automated Refactoring

Cmd-DEL: Safe Delete

Advanced Find-Replace – Understands Your Code Structure

Cmd-Shift-A: Find Action Replace Structurally…

What Next?

Probe

Refactor Sub-Goal

Fix Feature Envy

Automated Refactoring

Cmd-Opt-M: Extract Method

Prepare for Move Instance Method

Cmd-Opt-P: Introduce Param

Postfix

.arg: Add Method Call

Multi-Caret

Ctrl-G:

Select Next Occurrence

Automated Refactoring

F6: Move [Instance] Method

Automated Refactoring

Shift-F6: Rename [Method]

(or Variable, Method, etc.)

Lost? Not Sure What to Do Next?

Find Scaffolding

Send in the LLM 

Cmd-\: Generate Code with AI

Automated Refactoring

Cmd-Opt-N: Inline [Field]

(or Variable, Method, etc.)

Automated Refactoring

Cmd-Opt-C: Introduce Constant

Refactoring Maneuvers

- Minimize Time in RED
- Probe and fix, or *revert*
- Leave Breadcrumbs

There is not always one path to the goal.

Goals change as you walk the path.

Pay attention to your surroundings 👀

Change direction when you discover a better goal

Ted M. Young
Java Technical Coach, & Live Coder

Get in touch: ted@tedmyoung.com
About me: https://ted.dev/about

Refactor in small steps, with clear intention

Ted M. Young
Java Technical Coach, & Live Coder

Get in touch: ted@tedmyoung.com
About me: https://ted.dev/about

Thank You...

Source Code? Slides?
https://ted.dev/talks/