

## Refactoring in Java

 Duration: 2 Days

 Available Languages: English German

### Audience

Software Developers, Software Architects, Testers.

### Goals

Learn how to improve your Software Structure with Refactoring.

### Contents

- Principles of Refactoring
  - # Definition
  - # When and Why
- Design Smells (Rigidity, Fragility, Inseparability, Opacity, Viscosity)
- The 4 Rules of Simple Design
- Refactoring Safely: Characterization Tests
- Code Smells  
(Excerpt: Anemic class, middle man, long method, divergent change, primitive obsession, temporary field...)
- Refactoring Catalogue
  - # Field and Variable Refactoring
  - # Method Refactoring
  - # Class Refactoring
  - # Generalization
  - # Encapsulation
  - # Moving Features
  - # Organizing Data
  - # Simplifying Conditional Logic
  - # API Refactoring
  - # Dealing with Inheritance
- Refactoring Tools / IDEs
- Outlook:
  - # Clean Code
  - # Design Patterns
  - # Refactoring to Patterns

The course uses IntelliJ IDEA 2022.2.2, OpenJDK 18, Maven 3.8.6, Gradle 7.5.1, JUnit Platform 1.9.1, JUnit Jupiter 5.9.1, Cucumber 7.8.0, Pitest 1.9.5, and Spring Boot 2.7.4. Support for the recent OpenJDK 19 release will be added as soon as Groovy and Gradle have made their update releases for Java 19.

The training is loosely based on two books:

- Martin Fowler, "Refactoring, Improving the Design of Existing Code", 2nd Edition
- Joshua Kerievsky, "Refactoring to Patterns"

The course language is Java. Nelkinda also offers this course in other languages, for example, C, C++, C#, JavaScript, Kotlin, Python, Swift, and TypeScript.

## Booking

Contact Siddhesh Nikude, +91-95-52572354, [training@nelkinda.com](mailto:training@nelkinda.com)