

Clean Code TypeScript

 Duration: 2 Days

 Available Languages: English German

Audience

Software Crafter: Software Developers, Architects, Project Managers, Scrum Masters.

Precondition

Good knowledge of TypeScript.

Goals

Learn how to develop software using the Clean Code principles in TypeScript.

Contents

- The two values of Software
- Design Smells: Rigidity, Fragility, Immobility/Inseparability, Opacity, Viscosity
- Programming Paradigms
 - # Structured Programming
 - # Object-Oriented Programming, Polymorphism
 - # Functional Programming
 - # Predicate Logic
- Craft Culture, Professionalism, and Clean Code
- Clean Code Fundamentals: Comments, Meaningful Names, Clean Functions and Classes, Source Code Structure, Code Style, Formatting, Boundaries, Error Handling
- Error Handling
- Core Architecture: Coupling and Cohesion, Vertical (Domain) vs Horizontal (Technical) Software Architecture
- The Clean SDLC: Agile, Scrum, XP, CI/CD, DevOps
- The SOLID and Package Principles: 11 Principles of Clean Architecture and Design
 - # SRP, OCP, LSP, ISP, DIP
 - # Package Principles and Package Metrics
- Design Patterns in the light of Clean Code
 - # Creational Patterns
 - # Structural Patterns
 - # Behavioral Patterns
- Smells and Heuristics: Comments, Environment, Functions, General, Names, Tests
- Some Pragmatic Tips
- Basics of Test-Driven Development

The training can be offered in all major programming languages. For corporate training, please contact us in advance for planning the training to meet needs and environment of your development teams.

Booking

Contact Siddhesh Nikude, +91-95-52572354, training@nelkinda.com