

This is a course explains best practices, patterns and key concepts of Java Application design.

Domain Modelling

- Object-oriented **ecosystem** (concepts, roles, methods, tools)
- **Requirements** Modelling (business/user/functional/quality requirements)
- Use-case driven design (and alternatives)
- Domain modeling (identifying domain concepts and classes)
- System **Behavioural modelling**
- Object-oriented Software Design (in code, through **re-factoring** and when needed in UML)
- Abstraction (and single abstraction **SLAP**)
- Java and the **Anemic domain** Model
- Cohesion
- Coupling
- Law of Demeter (LoD)

Design and Software Architecture Principles

- Coupling
- DRY (don't repeat yourself)
- Rule of the least power
- YAGNI ("you ain't gonna need it")
- TDD cycles (red-green-refactor)
- EIP (enterprise integration Patterns)
- Don't make code better than needed (flexibility, performant)
- Cost of over-engineering