



Software Architectures

Course Overview

Introduction

- ▶ What is architecture?
- ▶ Modeling architecture with UML

Principles of Object Oriented Design

- ▶ Comprehensibility, Correctness, and Extensibility
- ▶ Encapsulation, Abstraction, and Information Hiding
- ▶ Separation of Concerns and the Single Responsibility Principle
- ▶ Interface Segregation Principle
- ▶ Loose Coupling
- ▶ Liskov Substitution Principle
- ▶ Design by Contract
- ▶ Open-Closed Principle
- ▶ Dependency Inversion Principle and Inversion of Control

Software Architectures Overview

- ▶ What is a Software Architecture?
- ▶ Quality of a Software Architecture
- ▶ Reference Architectures, Architecture and Design Patterns
- ▶ The Role of the Software Architect

Architecture and Design Pattern Overview

- ▶ Using Patterns
- ▶ Pattern Properties and Design
- ▶ Architecture Patterns, Design Patterns, and Idioms

Architecture Patterns

- ▶ Layers or Tiers
- ▶ Pipes and Filters
- ▶ Plug-in
- ▶ Broker
- ▶ Service-Oriented Architecture
- ▶ Model-View-Controller

Structural Patterns

- ▶ Adapter
- ▶ Bridge
- ▶ Decorator
- ▶ Façade
- ▶ Composition
- ▶ Proxy

Behavioural Patterns

- ▶ Command
- ▶ Observer
- ▶ Strategy
- ▶ Mediator
- ▶ State
- ▶ Role
- ▶ Visitor
- ▶ Iterator

Generational Patterns

- ▶ Factory
- ▶ Abstract Factory
- ▶ Singleton
- ▶ Object Pool

Literature

- ▶ "Architektur- und Entwurfsmuster der Softwaretechnik", J. Goll and M. Dausmann, Springer Vieweg, 2013
- ▶ "Design Patterns: Elements of Reusable Object-Oriented Software", E. Gamma, R. Helm, R. Johnson, J. Vlissides, Addison-Wesley, 1995

Lab

You will be assigned a larger project you need to work on within a team of three people.

Exam Aids

You are permitted to bring two sheets of DIN A4 paper.