

Agenda "C++ Design Pattern"

1. Tag

Literatur

Voraussetzungen (bei Bedarf)

- Schnittstellenkonzept (interface)
- Abstrakte Klasse

SOLID / Entwurfsprinzipien (Design Principles)

- Single Responsibility Prinzip (SRP)
- Open-Closed Prinzip (OCP)
- Interface Segregationsprinzip (ISP)

Strukturmuster (*Structural Design Pattern*).

- Decorator
- Flyweight
- Facade
- Adapter
- Composite
- Proxy

Wenn die Zeit noch reicht:

- Bridge

■ Übungen Teil 1:

- Passwort Validierer
- Semigraphisches Videospiel
- Anzahl der Objekte einer baumartigen Struktur bestimmen
- Password Generator

■ Allgemein: Das Thema `shared_from_this` ist bei folgenden Mustern behandelt:

- Mediator (Beschreibung vorhanden)
- Observer

- Composite
- State
- Visitor

2. Tag

Type Erasure

Verhaltensmuster (*Behavioral Design Pattern*)

- Strategy Pattern
- Command Pattern
- State
- Visitor Method
- Template Method
- Chain of Responsibility
- Mediator
- Memento

Wenn die Zeit noch reicht:

- Observer
- Iterator
- Interpreter

■ Übungen Teil 2:

- Genehmigungssystem
- Bankkonten eines Bankinstituts
- Simulation einer Verkehrsampel
- Ein Textprozessor
- Versenden von Benachrichtigungen

3. Tag

SOLID / Entwurfsprinzipien (Design Principles)

- Dependency Inversion Prinzip (DIP)

Besonderes Muster: DI (Dependency Injection)

- Inversion of Control
- Dependency Inversion Principle
- Dependency Injection
- IoC Container / Service Locator

- Beispiel

Erzeugungsmuster (*Creational Design Pattern*).

- Singleton
- Simple Factory
- Factory Method
- Prototype
- Builder

Wenn Zeit:

- Abstract Factory

Anti-Pattern

- God Object
- Input Kludge
- Interface Bloat
- Copy and Paste Programming
- ...

Weitere besondere Muster

- RAI
- Pimpl
- Null Object
- Intercepting Filter Pattern
- SFINAE
- CRTP

■ Übungen Teil 3:

- "Dependency Inversion Principle"

Entwurfsmuster für Oberflächen

- MVC Model-View-Controller
- MVP Model-View-Presenter
- MVVM Model View View-Model