

Kurs-Dokumentation



Zentrum für Informatik ZFI AG

OO Design (ODES)

<http://www.zfi.ch/ODES>

Weitere Infos finden Sie unter www.zfi.ch oder via Adresse:

Zentrum für Informatik ZFI AG
Zentralsekretariat
Rütistrasse 28
CH-8952 Zürich-Schlieren
Telefon: 044 732 40 00
Telefax: 041 530 31 68

Zürich, Basel, Bern, ZÄ¼rich, Schweiz

| | |
|------------------------|--|
| Titel | OO Design |
| Untertitel | Systemmodellierung in der Praxis |
| Einleitung | Objektorientierte Konzepte erlauben, die Entwicklung komplexer Systeme überschaubar zu halten. Die Mechanismen des objektorientierten Designs ermöglichen modulare und flexible Systeme mit vertretbarem Designaufwand. Mit objektorientierten Entwicklungsmethoden können Sie langlebige Systeme so realisieren, dass sie einfach gewartet, an neue Anforderungen angepasst und weiterentwickelt werden können. Dabei ist objektorientiertes Design mehr als eine Aufteilung des Systems in Klassen und Objekte: gute objektorientierte Systeme bestehen aus einem austariertem Zusammenspiel von Komponenten und ihren Beziehungen, die einen Kompromiss für die verschiedenen Anforderungen an ein System darstellen. |
| Ihr Nutzen | Im ZFI-Seminar "OO Design" lernen Sie, komplexe Systeme objektorientiert zu entwerfen. Sie setzen die verschiedenen OO-Mechanismen ein, um die funktionalen und die nicht-funktionalen Anforderungen an das System zu erfüllen. Sie können beurteilen, wann zusätzliche Design-Klassen sinnvoll sind und setzen sie zur Modularisierung und Flexibilisierung Ihres Systems ein. Sie treffen Ihre Design-Entscheidungen bewusst und dokumentieren sie in UML. |
| Voraussetzungen | Sie haben Erfahrung in Software-Design und Grundkenntnisse der objektorientierten Konzepte entsprechend dem Seminar Objektorientierte Grundlagen (OUGL). |
| Teilnehmerkreis | Systemarchitekten, Programmierer, Software-Entwickler, Software-Designer, Tester, Integratoren, Systemverantwortliche, Library-Entwickler, System-Designer |
| Unterlagen | |
| Folgekurse | OO Design Patterns (OODP), Java Advanced (JPF2), C++ Advanced (CADV) |
| Inhalt | <p>Einführung</p> <ul style="list-style-type: none"> - Verbesserung der Kommunikation zwischen Entwicklern und Anwendungs-Experten - Besser wartbare Software - Modularisierung <p>UML (Diagramm-Typen)</p> <ul style="list-style-type: none"> - Diagramme der statischen Modelle (Struktur, Aufbau, ..) - Diagramme der dynamischen Modelle (Verhalten, Ablauf, ?) <p>Vorgehens- und Prozessmodelle</p> <ul style="list-style-type: none"> - Wasserfall, iterative und inkrementelle Vorgehensmodelle |

- Phasenmodelle, Agile-Methoden

Anforderungen

- Use-Cases
- Funktionale Anforderungen
- Nicht funktionale Anforderungen

Analyse

- Use-Case-Diagramme
- Use-Case-Beschreibungen
- Prototypen

Architektur

- Mehrschicht-Architektur (Multi-Tier)
- Model-View-Controller (MVC)

Entwurf

- Statische Modelle (Klassendiagramm: Klassen ihre Beziehungen, Klassen und ihre Attribute und Operationen)
- Dynamische Modelle (Verhalten der Klassen und Operationen, Interaktion zwischen den Klassen)

Entwurfsmuster

- Einführung (Überblick, Beschreibung und Einteilung der Muster: Erzeuger, Struktur, Verhalten)
- Einige Beispiele (Singleton, Observer, ?)

Implementierung

- Entwicklungsumgebungen
- Programmiersprachen

- Codierung (Style-Guides) und Bewertung des Code (Metriken)

Refactoring

- Der Architektur

- Des Codes

Testen

- Überblick (Vorgehen, Testarten, Testfälle)

- Review

- Unit-Test

Beitrag

Der Teilnehmerbeitrag versteht sich rein netto. Das ZFI ist (gemäss MwSt-Gesetz) nicht Mehrwertsteuerpflichtig und erhebt somit keine MwSt. Bei länger als einen Monat dauernden Lehrgängen ist die Zahlung des Teilnehmerbeitrages in mehreren Raten möglich (pro rata temporis).

Bildungsweg OO-Methoden

OO

Objektorientierte Methoden

