MICRO CONSULT

Stand 07.10.2022

Objektorientierte Softwareentwicklung: Spezielle Programmierprinzipien mit C++ - Der Weg zum Clean Code - Live-Online-Training

**Ziele -
Ihr Nutzen**

Sie lernen die wichtigsten Regeln der Softwareentwicklung und ihre Bedeutung sowie Muster kennen, nach denen Sie Ihre Codestruktur verbessern können.

Teilnehmer

Software-Entwickler, Software-Architekten

Voraussetzungen

Kenntnisse einer objektorientierten Programmiersprache (Übungen werden wahlweise in C++ oder C# durchgeführt)

Live Online Training

17.10. – 19.10.2022	1.800,00 €	3 Tage	★
16.01. – 18.01.2023	1.800,00 €	3 Tage	
27.03. – 29.03.2023	1.800,00 €	3 Tage	
17.07. – 19.07.2023	1.800,00 €	3 Tage	
16.10. – 18.10.2023	1.800,00 €	3 Tage	
08.01. – 10.01.2024	1.800,00 €	3 Tage	

* Preis je Teilnehmer, in Euro zzgl. USt.

Anmeldecode: L-OOPFC++

Präsenz-Training - Deutsch

Termin	Dauer
16.01. – 18.01.2023	3 Tage
27.03. – 29.03.2023	3 Tage
10.07. – 12.07.2023	3 Tage
16.10. – 18.10.2023	3 Tage
08.01. – 10.01.2024	3 Tage

Objektorientierte Softwareentwicklung: Spezielle Programmierprinzipien mit C++ - Der Weg zum Clean Code - Live-Online-Training

Inhalt

Softwareentwicklung als Handwerkskunst - 'Software Craftsmanship' - Der Weg zum Clean Code

Warum ist guter Code wichtig?

- Was zeichnet guten Code aus?
- Welche Probleme verursacht schlechter Code?
- Was sind die Ursachen für schlechten Code?
- Warum ist es sinnvoll, auf guten Code Wert zu legen?

- Wie entsteht guter Code?

Grundregeln zur Erstellung guten Codes

- DRY - Don't Repeat Yourself
- KISS - Keep it simple, stupid
- Geheimnisprinzip
- Programming to an Interfac
- Modularisierung
- Prinzip der losen Kopplung
- Prinzip der hohen Kohäsion
- Vorsicht vor Optimierungen
- POLS - Principle of Least Surprise
- Übungen zum besseren Verständnis der Prinzipien

Die SOLID-Prinzipien

- Single-Responsibility-Prinzip
- Open-Closed-Prinzip
- Liskovsches Substitutionsprinzip
- Interface-Segregation-Prinzip
- Dependency-Inversion-Prinzip
- Übungen zum besseren Verständnis der Prinzipien

Weitere Prinzipien

- FCoI - Favour Composition over Inheritance
- SLA - Single Level of Abstraction
- Tell don't ask
- Law of Demeter
- YAGNI - You Ain't Gonna Need It
- Nutze Source Code Konventionen
- Übungen zum besseren Verständnis der Prinzipien

Refaktorisierung von Code

- Was ist Refaktorisierung?
- Welche Arten gibt es?
- Wie wird eine Refaktorisierung durchgeführt?
- 'Smells', die auf die Notwendigkeit einer Refaktorisierung hinweisen
- Refaktorisierungspatterns
- Übung: Finden von 'Smells' im Code
- Übungen zum Refactoring: Einsatz ausgewählter Patterns

Hinweise zur Verbesserung der Codequalität im Projekt

- Wie lässt sich Bewusstsein für guten Code schaffen?
- Wie lässt sich Code kontinuierlich verbessern?

Praktische Übungen

- Übungen zum besseren Verständnis der Programmierprinzipen
- Finden von 'Code-Smells'
- Übungen zum Einsatz von Refactoring-Patterns

MicroConsult PLUS

- Ihre Übungsverzeichnisse und Lösungsbeispiele für alle Übungsaufgaben stellen wir Ihnen zum Download bereit.