

UML-Schulung: UML-Grundlagen und Einstieg in die modellbasierte Softwareentwicklung - Live-Online-Training

Ziele - Ihr Nutzen

In der UML-Schulung lernen Sie, wie Sie Analyse und Entwurfsverfahren sowie die Darstellungsform der Unified Modeling Language (UML) kompetent einsetzen.

Teilnehmer

Software-Entwickler, Software-Architekten, Software-Entwicklungsleiter, Systemarchitekten

Voraussetzungen

Programmiererfahrung (z.B. C, C++, Java, C#)

Live Online Training

* Preis je Teilnehmer, in Euro zzgl. USt.

Anmeldecode: L-UML-G

Präsenz-Training - Deutsch

Termin	Dauer
04.05. – 06.05.2026	3 Tage

Präsenz-Training - Englisch

Dauer
3 Tage

UML-Schulung: UML-Grundlagen und Einstieg in die modellbasierte Softwareentwicklung - Live-Online-Training

Inhalt

Einführung in die objektorientierte Entwicklung

- Grundbegriffe der objektorientierten Entwicklung
- Gründe für den Einsatz objektorientierter Methoden
- Vorteile beim Einsatz der objektorientierten Entwicklung

Anforderungen und Anwendungsfälle in der UML

- Einsatz von Use-Case-Diagrammen
- Aufbau von Use-Case-Diagrammen
- Elemente und Beziehungen im Use-Case-Diagramm
- Use-Case-Spezifikation
- Praxistipps zum Umgang mit Use-Cases
- Übung: Entwurf eines Use-Case-Diagramms für ein kleines Softwareprojekt

Klassen und Objekte

- Was sind Objekte?

- Was sind Klassen?
- Welchen Zusammenhang gibt es zwischen Objekt und Klasse?
- Wie werden Objekte identifiziert und Klassen entwickelt?
- UML-Syntax von Klasse und Objekt
- Praxistipps zum Finden von Objekten
- Übung: Modellieren einer Klasse

Klassendiagramm als zentrales UML-Strukturdiagramm

- Zweck und Inhalt des Klassendiagramms
- Beziehungen zwischen Klassen
- Genaue Spezifizierung der Beziehungen
- Generalisierung (Vererbung) und Polymorphie
- Abstrakte Klassen und Interfaces
- Stereotypen und Zusicherungen (Constraints)
- Übung: Finden von Objekten und Abstraktion zu Klassen, Herstellen der Beziehungen

Weitere UML-Strukturdiagramme

- Kompositionssstrukturdiagramm
- Parts und Ports
- Kollaboration
- Komponentendiagramm
- Deployment-Diagramm
- Nutzung des Paketdiagramms zur Beschreibung der Programmarchitektur und des Software-Designs
- Praxistipps zum Einsatz der verschiedenen Strukturdiagramme
- Übung: Strukturieren des Projektes und Aufteilen der Klassen in verschiedene Pakete

Sequenzdiagramm

- Aufbau und Inhalt des Sequenzdiagramms
- Darstellung der Interaktion zwischen Objekten
- Zeitangaben
- Überblick über die verschiedenen Fragmente
- Praxistipps zum sinnvollen Aufbau eines Sequenzdiagramms
- Übung: Darstellen zeitlicher Abläufe im Sequenzdiagramm

Kommunikationsdiagramm

- Aufbau und Inhalt des Kommunikationsdiagramms
- Darstellung des Zusammenspiels von Objekten
- Praxistipps zur sinnvollen Nutzung des Kommunikationsdiagramms
- Übung: Erstellen eines Kommunikationsdiagramms

Aktivitätsdiagramm

- Darstellung von Programm- und Datenflüssen
- Überblick über die Elemente im Aktivitätsdiagramm
- Verschachtelte Aktionen
- Interaktionsübersichtsdiagramm als Möglichkeit der Abstraktion
- Praxistipps zum Einsatz des Aktivitätsdiagramms
- Übung: Darstellen des Programmflusses im Aktivitätsdiagramm

Zustandsfolgediagramm

- Darstellung von Automaten (State Machines) mit dem Zustandsfolgediagramm (State Chart)
- Aufbau des Zustandsfolgediagramms
- Verschachtelte Automaten
- History-Zustand
- Praxistipps zum Umgang mit Zustandsautomaten
- Übung: Identifizieren von aktiven Klassen und Darstellung ihres Verhaltens im Zustandsfolgediagramm

Zeitdiagramm

- Zweck und Inhalt des Zeitdiagramms (Timing-Diagramm)
- Zeitangaben

Implementierungsbeispiele für Klassen, Interfaces und verschiedene Beziehungen

- Objektorientierte Konzepte in prozeduralen Sprachen
- Beispiele in C, C++, C# und Java

Übungen der UML-Schulung

- Entwurf eines Use-Case-Diagramms für ein kleines Softwareprojekt

- Modellieren einer Klasse
- Finden von Objekten und Abstraktion zu Klassen, Herstellen der Beziehungen
- Strukturieren des Projektes und Aufteilen der Klassen in verschiedene Pakete
- Darstellen zeitlicher Abläufe im Sequenzdiagramm
- Erstellen eines Kommunikationsdiagramms
- Darstellung des Programmflusses im Aktivitätsdiagramm
- Identifizierung von aktiven Klassen und Darstellung ihres Verhaltens im Zustandsfolgediagramm
- Die Übungen werden jeweils nach Abschluss der entsprechenden Themen durchgeführt

MicroConsult PLUS

- Sie erhalten von uns Ihre Übungsverzeichnisse und Lösungsbeispiele für alle Übungsaufgaben.