MICRO CONSULT

Stand 10.06.2026

Design Patterns Schulung (nicht nur) für Embedded-Systeme - Präsenz-Training

**Ziele -
Ihr Nutzen**

Lernen Sie abzuschätzen, unter welchen Bedingungen Sie klassische Entwurfsmuster - Design Patterns - auch in ressourcenlimitierten Embedded-Systemen gewinnbringend einsetzen können.

Die Schulung macht Sie mit den Mustern vertraut, die sich besonders gut für die typischen Anwendungen von Embedded-Systemen eignen. Sie können so nicht nur deren Speicherplatz- und Laufzeitkosten beurteilen, sondern diese auch durch Laufzeitmessungen überprüfen.

Sie erfahren, welche häufig verwendeten Lösungsansätze besser vermieden werden sollten ("Anti-Pattern"), lernen Sie Einsatzmöglichkeiten von Entwurfsmustern zum Zweck der Fehlersuche kennen und können diese anwenden.

Teilnehmer

Die Design Patterns Schulung richtet sich an C++ Software-Entwickler und Software-Architekten.

Voraussetzungen

Gute Kenntnisse der Programmiersprache C++

Design Patterns Schulung (nicht nur) für Embedded-Systeme - Präsenz-Training

Inhalt

Einführung in die Design Patterns (Entwurfsmuster)

- Geschichtliche Entwicklung
- Was ist ein Entwurfsmuster?
- GoF Entwurfsmuster (GoF Design Pattern)
- Typische Probleme in Embedded-Systemen
- Entwurfsmuster (Design Patterns) in Embedded Systemen

Erzeugungsmuster

- Beispiel: Applikation zur Steuerung eines Motors
- Flexibles Design auf Basis von Schnittstellen (Interfaces)
- Praktische Übung: Messung der Speicherplatz- und Laufzeitkosten einer Schnittstelle
- Statische Polymorphie und dynamische Polymorphie im Vergleich
- Beispiel: Positionsverfolgung für ein Warentransportsystem
- Wiederverwendung des Positionsverfolgungssystems für Flugzeuge
- Positionsverfolgung auf Basis des Entwurfsmusters "Abstrakte Fabrik" (Design Pattern "Abstract Factory")
- Fabrikerzeugung mithilfe des Entwurfsmusters "Singleton" (Design Pattern "Singleton")

Strukturmuster

- Beispiel: Applikation zur Steuerung eines Motors
- Alternatives Design auf Basis des Entwurfsmusters "Adapter" (Design Pattern "Adapter")
- Workshop-Übung: Debugging einer Counter-Applikation
- Lösung mithilfe des Entwurfsmusters "Dekorierer" (Design Pattern "Decorator")
- Beispiel: Multithread-Applikation
- Identifikation der Probleme typischer Lösungsansätze
- Flexibler Lösungsansatz auf Basis des Entwurfsmusters "Proxy" (Design Pattern "Proxy")
- Schutz-Proxy, Virtueller Proxy, Remote Proxy
- Smart-Reference / Smart-Pointer

Verhaltensmuster

- Beispiel: Behandlung von Timer-Ereignissen

- Flexible Lösung auf Basis des Entwurfsmusters "Beobachter" (Design Pattern "Observer")
- Praktische Übung: Anwendung des Beobachter-Musters in einer Aufzug-Steuerung
- Fallstricke beim Design bzw. der Implementierung von Interfaces
- "Horizontale" und "vertikale" Interfaces
- Ereignisbehandlung auf Basis des Entwurfsmusters "Befehl" (Design Pattern "Command")
- Praktische Übung: Anwendung des Befehlsmusters in der Aufzug-Steuerung
- Beispiel: Traditionelle Implementierung eines Zustandsautomaten in C
- Objektorientierte Lösung mithilfe des Entwurfsmusters "Zustand" (Design Pattern "State")
- Praktische Übung: Anwendung des Zustandsmusters in der Aufzug-Steuerung
- Beispiel: Benutzerdefinierte Speicherverwaltung
- Partitionen und deren Verwaltung mithilfe von Partition-Managern
- Flexible Speicherverwaltung auf Basis des Entwurfsmusters "Strategie" (Design Pattern "Strategy")
- Praktische Übung: Anwendung des Strategiemusters in der Aufzug-Steuerung
- Beispiel: Strategien mit gemeinsamer Grundstruktur
- Realisierung auf der Basis des Entwurfsmusters "Schablonenmethode" (Design Pattern "Template Method")

Weitere Muster

- Entwurfsmuster "Fabrikmethode" (Design Pattern "Factory Method")
- Entwurfsmuster "Prototyp" (Design Pattern "Prototype")
- Entwurfsmuster "Fassade" (Design Pattern "Facade")
- Entwurfsmuster "Kompositum" (Design Pattern "Composite")
- Entwurfsmuster "Memento" (Design Pattern "Memento")
- Entwurfsmuster "Zuständigkeitskette" (Design Pattern "Chain of Responsibility")
- Entwurfsmuster "Fliegengewicht" (Design Pattern "Flyweight")
- Entwurfsmuster "Iterator (Design Pattern Iterator)
- Entwurfsmuster "Vermittler" (Design Pattern "Mediator")

Praktische Übungen in der Design Pattern Schulung

- Die Übungen werden mit der Plattform IAR Embedded Workbench und dem Designwerkzeug Enteprise Architect durchgeführt

Zusätzlich zu den praktischen Übungen in den einzelnen Kursmodulen gewinnen Sie folgende Praxiskenntnisse:

- Wie führe ich Speicherplatz- und Laufzeitmessungen durch?
- Wie kann ich in Projekten Entwurfsmuster als Mittel zur Steigerung der Softwarequalität einsetzen?
- Wie kann ich Entwurfsmuster zu Debug-Zwecken verwenden?
- Wie kann ich gegebene Projekte durch die Anwendung von Entwurfsmustern weiterentwickeln?

Präsenz-Training

★ Mit Durchführungsgarantie

Termin	Preis *	Dauer
06.07.2026 – 10.07.2026	3.000,00 €	4,5 Tage ★

* Preis je Teilnehmer, in Euro zzgl. USt.

Anmeldecode: DP

Live-Online - Deutsch

Termin	Dauer
09.11. – 13.11.2026	5 Tage

Präsenz-Training - Englisch

Dauer
4,5 Tage

Live-Online - Englisch

Dauer
5 Tage

Coaching

Unsere Coaching-Angebote bieten den großen Vorteil, dass unsere Experten ihr Wissen und ihre Erfahrungen direkt in Ihren Lösungsprozess einbringen und damit unmittelbar zu Ihrem Projekterfolg beitragen.

Für Ihre Anfrage oder weiterführende Informationen stehen wir Ihnen gern zur Verfügung.