

## Zustandsautomaten (Finite State Machines, FSM): Entwurf und Implementierung mit C++

### Ziele - Ihr Nutzen

Sie beherrschen die Grundlagen für den Entwurf von Zustandsautomaten (endlicher Automat - Finite State Machine).

Sie lernen verschiedene Implementierungen für Zustandsautomaten kennen und können dieses Wissen bei der Implementierung eigener Automaten anwenden oder setzen es ein, um über die Auswahl eines Frameworks zu entscheiden.

Weiterhin werden die wichtigsten Grundlagen zum Test von Automaten vermittelt.

### Teilnehmer

Software-Entwickler und Software-Architekten

### Voraussetzungen

Solide C++-Kenntnisse

## Zustandsautomaten (Finite State Machines, FSM): Entwurf und Implementierung mit C++

### Inhalt

#### Einführung in die Gedankenwelt der Automaten

- Warum ist der Einsatz von Zustandsautomaten sinnvoll?
- Gründe, warum noch zu wenig Automaten im Einsatz sind

#### Grundlagen

- Welche unterschiedlichen Automaten gibt es?
- Automaten für den Einsatz in industrieller Software (z.B. Embedded-Systemen)
- Unterschied zwischen Akzeptoren und Transduktoren
- Die Automatenmodelle von Moore und Mealy
- Darstellung von Automaten in der UML

#### Der Entwurf von Zustandsautomaten

- Unterschiede beim Entwurf von Moore- oder Mealy-Automaten
- Vor- und Nachteile der verschiedenen Automatenmodelle und deren Mischformen bzw. Erweiterungen
- Praxistipps zur Auswahl des richtigen Automatenmodells
- Hierarchische Automaten
- History-Zustand
- Hinweise zum Vorgehen beim Entwurf von Automaten
- Typische Fehler beim Entwurf
- Das Thema wird anhand praktischer Beispiele erklärt
- Verschiedene Übungen zum Entwurf von Automaten

#### Implementierung von Zustandsautomaten

- Beispiele in C und C++
- Vorstellung verschiedener Implementierungsformen
- Switch-Case
- State Pattern
- Verschiedene Arten tabellengesteuerter Automaten
- Hierarchische Automaten
- Asynchrone Automaten
- Ausblick auf mögliche Erweiterungen der vorgestellten Implementierungen

- Alternativen zur Eigenentwicklung von Zustandsautomaten
- Allgemeine Hinweise zur Implementierung
- Das Thema wird anhand praktischer Beispiele erklärt
- Verschiedene Übungen zur Implementierung

**Ressourcenverbrauch**

- Gegenüberstellung der Codegröße der verschiedenen Implementierungen
- Betrachtung des Zeitverhaltens

**Test von Zustandsautomaten**

- Welche Fehler können durch einen Test erkannt werden?
- Abdeckungsmaße für Zustandsautomaten
- Definition von Testfällen
- Einsatz von Unit-Tests
- Betrachtung der Testbarkeit der verschiedenen Implementierungen
- Das Thema wird anhand praktischer Beispiele erklärt
- Verschiedene Übungen zum Test von Automaten

**Übungen im FSM-Training**

- Übungen zum Entwurf von Automaten
- Übungen zur Implementierung
- Übungen zum Test
- An jedem Trainingstag finden mehrere Übungen statt

**MicroConsult PLUS**

- Als Teilnehmer haben Sie die folgenden Möglichkeiten, um Ihre Übungsverzeichnisse und Lösungsbeispiele für alle Übungsaufgaben mitzunehmen:
- Sie nehmen die Dateien direkt auf einem von MicroConsult bereitgestellten kostenfreien USB-Stick mit, oder
- Sie senden sich die Dateien selbst per E-Mail zu, oder
- Sie erhalten auf Anfrage Zugriff auf die Dateien per Download.

**Offenes Training**

★ Mit Durchführungsgarantie

<b>Termin</b>	<b>Preis *</b>	<b>Dauer</b>
11.11.2019 – 14.11.2019	2.200,00 €	4 Tage ★
16.03.2020 – 19.03.2020	2.200,00 €	4 Tage
13.07.2020 – 16.07.2020	2.200,00 €	4 Tage
25.01.2021 – 28.01.2021	2.200,00 €	4 Tage

\* Preis je Teilnehmer, in Euro zzgl. USt.

Anmeldecode: FSM-C++

**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.