

## C++ Standard Template Library: STL Container und Algorithmen

### Ziele -

#### Ihr Nutzen

In der "C++-STL" sind die universell einsetzbaren Bibliotheksklassen und -funktionen für "Container" und "Algorithmen" zusammengefasst. Durch die Abstrahierung des Zusammenspiels beider Bereiche über "Iteratoren" ist die STL ferner in beiden Dimensionen erweiterbar. Neben einem generellen Architektur-Überblick, der wichtig ist für ein tieferes Verständnis des STL-Designs und der sich daraus ergebenden Möglichkeiten (wie auch einigen Einschränkungen), liegt der Fokus des Trainings auf der Nutzung der Container-Klassen (Welcher Container für welchen Zweck?), Nutzung der Algorithmen zur Effizienzsteigerung und Codevereinfachung, der Laufzeit-Performance und dem Memory-Footprint. Den letzten Punkt betreffend wird besonderes Augenmerk auf "Hilfe zur Selbsthilfe" gelegt, um so die Teilnehmer in die Lage zu versetzen, auch im Rahmen ihres C++-Einsatzes bei Bedarf entsprechende Untersuchungen vorzunehmen und dabei die besonderen Bedingungen des geplanten Einsatzes eines Containers oder Algorithmus zu berücksichtigen. Bitte beachten Sie, dass dieses Trainings nicht auf die C++-Standardbibliothek AUSSERHALB der STL eingeht. Sofern Sie ein umfangreiches und vollständiges Wissen zur gesamten C++-Bibliothek anstreben, empfehlen wir zusätzlich den Besuch unseres Trainings "C++: Standard- und Boost-Library Workshop".

#### Teilnehmer

Softwareentwickler mit grundlegenden C++-Kenntnissen, die ihr Know-how auf den Bereich der "Standard-Template-Library" erweitern wollen.

#### Voraussetzungen

Grund-Kenntnisse in C++ - vor allem im Hinblick auf die Sprachsyntax und die grundlegende Verwendung der String- und Stream-Klasse. Hinweis: Notwendige Vorkenntnisse zu fortgeschritteneren C++-Features wie Template-Nutzung und C++11-Lambdas (als Call-Backs aus Algorithmen) werden - soweit nötig - Rahmen des Kurses aufgefrischt, stehen jedoch nicht in dessen Zentrum.

## C++ Standard Template Library: STL Container und Algorithmen

### Inhalt

#### Container-Klassen

- Sequenzielle Container
- Geordnete assoziative Container
- Ungeordnete assoziative Container

#### Iteratoren als Bindeglied

- Konzept der Iteratoren
- Iterator-Kategorien
- Spezielle Iteratoren

#### Algorithmen

- Modifizierende Algorithmen
- Nicht-modifizierende Algorithmen
- Sortieren und (effizientes) Suchen
- Mengen-Operationen
- Sonstige Algorithmen

#### Möglichkeiten zur Erweiterung

- Neue Container (Bereitstellung von Iteratoren)
- Neue Algorithmen (Nutzung von Iteratoren)
- Eigene Iteratoren (auch zur Nutzung als Generatoren)

#### Performance-Betrachtungen (inkl. Memory-Footprint)

- Grundsätzliche Unterschiede der Container
- Performance-Abschätzung ("Big-O"-Notation)
- Eigene Performance-Messungen (Tipps und Tricks)

**Begleitend: Mikro-Projekte**

- Demo-Code und/oder Aufgaben zur eigenen Bearbeitung nach Wahl
- Inkl. anschließender Erläuterung möglicher Variationen

**Offenes Training**

<b>Termin</b>	<b>Preis *</b>	<b>Dauer</b>
23.07.2018 – 25.07.2018	1.100,00 €	2,5 Tage
07.11.2018 – 09.11.2018	1.100,00 €	2,5 Tage
27.02.2019 – 01.03.2019	1.100,00 €	2,5 Tage

\* Alle Preise zzgl. der gesetzlichen USt.

Anmeldecode: C++/STL

**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.