

## 3. Templates und die STL mit dem C++Builder

Templates sind Vorlagen, aus denen der Compiler Klassen und Funktionen erzeugt. Die Standard Template Library (STL) ist eine Bibliothek, die zum ANSI/ISO-Standard von C++ gehört und die auf der Basis von Templates implementiert ist. Sie enthält viele Algorithmen und Datenstrukturen, die eine einfache Lösung häufig auftretender Programmieraufgaben ermöglichen. Aufgrund ihrer speziellen Architektur ist sie vielseitiger, leistungsfähiger, sicherer und einfacher zu benutzen als viele andere Bibliotheken. Da sowohl Templates als auch die STL in der jetzt gültigen Form erst relativ spät in C++ aufgenommen wurden, sind sie noch nicht sehr bekannt.

In diesem Kurs werden Templates und die STL vorgestellt. Dabei werden die Vorteile der STL-Klassen gegenüber den entsprechenden konventionellen Sprachelementen gezeigt (z.B. Stringklassen – nullterminierte Strings, Containerklassen – Arrays, Streamklassen – *stdio*-Funktionen). Außerdem wird gezeigt, wie die STL, ihre Container und ihre Algorithmen aufgebaut sind und wie man die STL um eigene Algorithmen erweitern kann.

Dieser Kurs ist der dritte von drei aufeinander abgestimmten Kursen, in denen der gesamte Sprachumfang des aktuellen ANSI/ISO-Standards von C++ und die wichtigsten Komponenten des C++Builders behandelt werden. Dabei stehen Zusammenhänge und Sprachkonzepte im Vordergrund vor Detailinformationen, die man auch in der Online-Hilfe findet.

**Voraussetzungen:** Gute Kenntnisse in C++ und der objektorientierten Programmierung mit C++ (im Umfang des Grundlagen-Kurses und des OO-Kurses).

**Zielgruppe:** Erfahrene C++-Programmierer

**Aufteilung:** Vortrag mit vielen Übungen, in denen praxisnahe Programme entwickelt werden

**Dauer:** 5 Tage

### 1 Die Stringklasse *string* der C++-Standardbibliothek

Die Stringklasse *string*  
Stringstreams

### 2 Container der Standardbibliothek

#### 2.1 Sequenzielle Container

Die Container-Klassen *vector*, *list* und *deque*  
Algorithmen der Standardbibliothek  
Die Container-Adapter *stack*, *queue* und *priority\_queue*  
Container mit Zeigern

#### 2.2 Assoziative Container

Die Container *set* und *multiset*  
Die Container *map* und *multimap*  
Iteratoren der assoziativen Container

#### 2.3 Dateien

Stream-Variablen, ihre Verbindung mit Dateien und ihr Zustand  
Fehler und der Zustand von Stream-Variablen  
Lesen und Schreiben von Binärdaten mit *read* und *write*

Lesen und Schreiben von Daten mit den Operatoren << und >>  
Manipulatoren und Funktionen zur Formatierung von Texten  
Dateibearbeitung im Direktzugriff

### **2.4 Die numerischen Klassen der Standardbibliothek**

Komplexe Zahlen, Valarrays und Slices

## **3 Templates und die STL**

### **3.1 Generische Funktionen: Funktions-Templates**

Funktions-Templates mit Typ-Parametern  
Spezialisierungen von Funktions-Templates  
Funktions-Templates mit Nicht-Typ-Parametern  
Explizit instanziierte Funktions-Templates  
Explizit spezialisierte und überladene Templates  
Rekursive Funktions-Templates

### **3.2 Generische Klassen: Klassen-Templates**

Klassen-Templates mit Typ-Parametern  
Spezialisierungen von Klassen-Templates  
Templates mit Nicht-Typ-Parametern  
Explizit instanziierte Klassen-Templates  
Partielle und vollständige Spezialisierungen  
Elemente und friend-Funktionen von Klassen-Templates  
Ableitungen von Templates  
Exportierte Templates

### **3.3 Funktionsobjekte in der STL**

Der Aufrufoperator ()  
Prädikate und arithmetische Funktionsobjekte  
Binder, Funktionsadapter und Negatoren

### **3.4 Iteratoren und die STL-Algorithmen**

Die verschiedenen Arten von Iteratoren  
Umkehriteratoren  
Einfügefunktionen und Einfügeiteratoren  
Stream-Iteratoren  
Container-Konstruktoren mit Iteratoren  
STL-Algorithmen für alle Elemente eines Containers

### **3.5 Die Algorithmen der STL**

Lineares Suchen und Zählen  
Suche nach Teilfolgen  
Minimum und Maximum  
Elemente vertauschen  
Kopieren von Bereichen  
Elemente transformieren, ersetzen und entfernen  
Partitionen und Sortieren  
Binäres Suchen in sortierten Bereichen  
Mischen von sortierten Bereichen  
Mengenoperationen auf sortierten Bereichen