

[Home](#)

Microsoft Visual C++ 2005/2008 Seminare

Diese Seminare werden von Prof. Richard Kaiser gehalten, der seit vielen Jahren Seminare für zahlreiche Firmen durchführt ([Referenzen](#)). Er ist der Verfasser der Bücher „C++ mit dem Borland C++Builder“ und „[C++ mit Microsoft Visual C++ 2008](#)“ (das im Lauf des Jahres 2008 erscheint) und ist Mitglied im [DIN Normierungsausschuss Informationstechnik NI-22](#).

Teil 1: Einführung in Microsoft Visual Studio 2005/2008

Dieser Kurs ist der erste von fünf aufeinander abgestimmten Kursen, in denen der gesamte Sprachumfang des aktuellen ISO C++-Standards und die wichtigsten Erweiterungen von Visual C++ behandelt werden. Dabei stehen Zusammenhänge und Sprachkonzepte im Vordergrund vor Detailinformationen, die man auch in der Online-Hilfe findet.

Zielgruppe:	Software-Entwickler, die mit Microsoft Visual Studio 2005/2008 in C++ programmieren wollen.
Voraussetzungen:	Gute Windows-Kenntnisse und Interesse am Programmieren.
Methodik:	Vortrag mit vielen Übungen, in denen praxisnahe Programme entwickelt werden
Ort und Zeit:	Dieses Seminar wird als Firmenseminar (inhouse) angeboten.
Dauer:	3 Tage

Inhalt

In diesem Kurs wird die Entwicklungsumgebung Visual Studio 2005/2008 vorgestellt und gezeigt, wie man damit in C++ .NET-Programme mit einer grafischen Benutzeroberfläche erstellen kann.

1 Die Entwicklungsumgebung

- Ein einführendes Beispiel
- Der Quelltexteditor und IntelliSense
- Kontextmenüs und Symbolleisten
- Projekte, Projektdateien und Projektoptionen
- Der integrierte Debugger

2 Komponenten für die Benutzeroberfläche

- Ein- und mehrzeilige Textboxen, Label
- ListBoxen und ComboBoxen
- Buttons und Ereignisse
- CheckBoxen, RadioButtons, GroupBox und Panel
- Hauptmenüs und Kontext-Menüs
- Standarddialoge

3 Ausblick auf C++/CLI

- Garbage collection auf dem GC-Heap
- CLI-Arrays

Die String-Klasse *String*
Einige .NET Klassen

Die Inhalte können an die Wünsche der Teilnehmer angepasst werden.

Kontakt: training@rkaiser.de ,
R. Kaiser, Schwärzlocher Straße 53, 72070 Tübingen, Tel. 07071-41598, FAX 07071-44473

Stand: 14.4.2008

[Home](#)

Microsoft Visual C++ 2005/2008 Seminare

Diese Seminare werden von Prof. Richard Kaiser gehalten, der seit vielen Jahren Seminare für zahlreiche Firmen durchführt ([Referenzen](#)). Er ist der Verfasser der Bücher „[C++ mit dem Borland C++Builder](#)“ und „[C++ mit Microsoft Visual C++ 2008](#)“ (das im Lauf des Jahres 2008 erscheint) und ist Mitglied im [DIN Normierungsausschuss Informationstechnik NI-22](#).

Teil 2: C Grundlagen

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

Zielgruppe:	Software-Entwickler, die mit Microsoft Visual Studio 2005/2008 in C++ programmieren wollen.
Voraussetzungen:	Kenntnisse im Umfang des Kurses "Teil 1: Einführung in Visual Studio 2005/2008"
Methodik:	Vortrag mit vielen Übungen, in denen praxisnahe Programme entwickelt werden
Ort und Zeit:	Dieses Seminar wird als Firmenseminar (inhouse) angeboten.
Dauer:	5 Tage

Inhalt

In diesem Kurs werden diejenigen Sprachelemente von C++ vorgestellt, die dem Sprachumfang von C entsprechen, der auch für die C++-Programmierung benötigt wird, sowie einige C++-Konzepte, die eng damit zusammenhängen.

1 Elementare Datentypen und Anweisungen

- Ganzzahldatentypen (*int*, *bool*, *char* usw.)
- Gleitkommatypen
- Standardkonversionen
- Elementare Anweisungen (*if*, *for*, *while* usw.)
- Funktionen, Werte- und Referenzparameter
- Die Verwendung der C-Standardbibliotheken (z.B. *math.h* und *cmath*)
- Arrays
- Strukturen (*struct* und *union*)
- Zeiger und dynamisch erzeugte Variablen (*new*, *delete*)
- Nullterminierte Strings und zugehörige Bibliotheksfunktionen
- typedef*
- Aufzählungstypen (*enum*)
- Präprozessoranweisungen (Makros, Pragmas)
- Ausdrücke
- Namensbereiche

2 Funktionen

- Werte- und Referenzparameter
- Funktionszeiger
- Rekursion
- inline*-Funktionen
- Default-Argumente
- Überladene Funktionen und Operatoren

3 Statische Bibliotheken

- Projekte, Bindung und Header-Dateien
- Der Aufruf von Funktionen aus C-Bibliotheken in C++

4 Ausblick auf die C++-Standardbibliothek

- Die Stringklasse *string*
- Die Container-Klasse *vector*
- Iteratoren
- Algorithmen der Standardbibliothek

Die Inhalte können an die Wünsche der Teilnehmer angepasst werden.

Kontakt: training@rkaiser.de, www.rkaiser.de

R. Kaiser, Schwärzlocher Straße 53, 72070 Tübingen, Tel. 07071-41598, FAX 07071-44473

Stand: 14.4.2008

[Home](#)

Microsoft Visual C++ 2005/2008 Seminare

Diese Seminare werden von Prof. Richard Kaiser gehalten, der seit vielen Jahren Seminare für zahlreiche Firmen durchführt ([Referenzen](#)). Er ist der Verfasser der Bücher „C++ mit dem Borland C++Builder“ und „C++ mit Microsoft Visual C++ 2008“ (das im Lauf des Jahres 2008 erscheint) und ist Mitglied im [DIN Normierungsausschuss Informationstechnik NI-22](#).

Teil 3: Objektorientierte Programmierung

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

Zielgruppe:	Software-Entwickler, die mit Microsoft Visual Studio 2005 in C++ programmieren wollen.
Voraussetzungen:	Kenntnisse im Umfang der Kurse "Teil 1: Einführung in Visual Studio 2005" und "Teil 2: C Grundlagen"
Methodik:	Vortrag mit vielen Übungen, in denen praxisnahe Programme entwickelt werden
Ort und Zeit:	Dieses Seminar wird als Firmenseminar (inhouse) angeboten.
Dauer:	5 Tage

Inhalt

In diesem Kurs wird die objektorientierte Programmierung und das Exception-Handling nach dem C++-Standard behandelt. Die objektorientierte Programmierung wird mit einer Einführung in die objektorientierte Analyse und das objektorientierte Design verbunden. Dabei werden die oft nicht einfachen Alternativen beim Entwurf von Klassenhierarchien und typische Design-Fehler ausführlich diskutiert.

1 Funktionen

- inline*-Funktionen
- Default-Argumente
- Überladene Funktionen und Operatoren

2 Objektorientierte Programmierung

Klassen

- Datenelemente und Elementfunktionen
- Objekte und die Zugriffsrechte *private* und *public*
- Der Aufruf von Elementfunktionen und der *this*-Zeiger
- Konstruktoren und Destruktoren
- OO Analyse und Design: Der Entwurf von Klassen
- Ein wenig Programmierlogik: Klasseninvarianten und Korrektheit

Klassen als Datentypen

- Der Standardkonstruktor
- Objekte als Klasselemente und Elementinitialisierer
- friend*-Funktionen und -Klassen
- Überladene Operatoren als Elementfunktionen
- Der Copy-Konstruktor
- Der Zuweisungsoperator = für Klassen
- Benutzerdefinierte Konversionen
- Explizite Konstruktoren
- Statische Klasselemente
- Konstante Klasselemente und Objekte
- Klassen und Header-Dateien

Vererbung

- Die Elemente von abgeleiteten Klassen
- Zugriffsrechte auf die Elemente von Basisklassen
- Konstruktoren, Destruktoren und implizit erzeugte Funktionen
- Vererbung bei Formularen in .NET
- OO Design: *public* Vererbung und "ist ein"-Beziehungen
- OO Design: Komposition und "hat ein"-Beziehungen
- Konversionen zwischen *public* abgeleiteten Klassen
- Mehrfachvererbung und virtuelle Basisklassen

Virtuelle Funktionen, späte Bindung und Polymorphie

- Der statische und der dynamische Datentyp
- Virtuelle Funktionen
- Die interne Realisierung von virtuellen Funktionen: *vptr* und *vtbl*
- Virtuelle Konstruktoren und Destruktoren
- Virtuelle Funktionen in Konstruktoren und Destruktoren
- OO-Design: Der Einsatzbereich von virtuellen Funktionen
- OO-Design und Erweiterbarkeit
- Rein virtuelle Funktionen und abstrakte Klassen
- OO-Design: Virtuelle Funktionen und abstrakte Basisklassen
- Interfaces und Mehrfachvererbung
- Protokollklassen und Programmgerüste
- Zeiger auf Klasselemente

Laufzeit-Typinformationen

- Typinformationen mit dem Operator *typeid*
- Typkonversionen mit *dynamic_cast* und *static_cast*

3 Exception-Handling

- Die *try*-Anweisung
- Exception-Handler und Exceptions der Standardbibliothek
- Vordefinierte .NET Exceptions
- Der Programmablauf bei Exceptions
- throw*-Ausdrücke und selbst definierte Exceptions
- Fehler, Exceptions und die Korrektheit von Programmen
- Die Freigabe von Ressourcen bei Exceptions

Die Klasse *shared_ptr*
 Exception-Spezifikationen
 Die Funktion *terminate*

Die Inhalte können an die Wünsche der Teilnehmer angepasst werden.

Kontakt: training@rkaiser.de , www.rkaiser.de

R. Kaiser, Schwärzlocher Straße 53, 72070 Tübingen, Tel. 07071-41598, FAX 07071-44473

Stand: 14.4.2008

[Home](#)

Microsoft Visual C++ 2005/2008 Seminare

Diese Seminare werden von Prof. Richard Kaiser gehalten, der seit vielen Jahren Seminare für zahlreiche Firmen durchführt ([Referenzen](#)). Er ist der Verfasser der Bücher „[C++ mit dem Borland C++Builder](#)“ und „[C++ mit Microsoft Visual C++ 2008](#)“ (das im Lauf des Jahres 2008 erscheint) und ist Mitglied im [DIN Normierungsausschuss Informationstechnik NI-22](#).

Teil 4: C++/CLI und die .NET-Klassenbibliothek

C++/CLI ist eine Weiterentwicklung der Programmiersprache C++, die nahtlos auf C++ aufbaut und die Besonderheiten der CLI (Referenz-, Werte- und Interface-Klassen, Eigenschaften, Ereignisse, Delegat-Typen, garbage collection usw.) in einer für C++-Programmierer intuitiven Form zur Verfügung stellt. Diese Erweiterungen ermöglichen die Entwicklung von **.NET-Anwendungen** und eine Nutzung aller Möglichkeiten der **.NET-Klassenbibliothek** unter **Visual C++ 2005/2008**. Damit ist es insbesondere nicht notwendig, für die .NET-Programmierung auf C# umzusteigen.

Die Änderungen von C++/CLI gegenüber C++ sind vor allem reine **Spracherweiterungen** und keine Änderungen der Bedeutung von C++-Sprachelementen. Deshalb kann ein C++/CLI-Compiler aus einem Standard-C++ Quelltext ein Programm erzeugen, das sich weitgehend genauso verhält wie ein von einem Standard-C++ Compiler erzeugtes Programm. Das ermöglicht die **Verwendung von existierendem C++-Code** in .NET-Anwendungen.

In der ersten Hälfte dieses Kurses wird der **gesamte Sprachumfang von C++/CLI** vorgestellt, und in der zweiten einige der wichtigsten .NET-Klassen. Nach dieser Schulung können C++-Programmierer .NET-Anwendungen entwickeln, ohne C# lernen zu müssen.

Die **.NET-Klassenbibliothek** stellt die gesamte Funktionalität, die unter Win32 von der Win32-API geboten wurde, in Form einer modernen, objektorientierten Klassenbibliothek zur Verfügung. Darüberhinaus enthält sie zahlreiche weitere Klassen, die viele Funktionalitäten bieten, die unter Win32 nicht zur Verfügung stehen.

Dieser Kurs ist der vierte von fünf aufeinander abgestimmten Kursen, in denen der gesamte Sprachumfang des aktuellen ISO C++-Standards und die wichtigsten Erweiterungen von Visual C++ 2005/2008 behandelt werden. Dabei stehen Zusammenhänge und Sprachkonzepte im Vordergrund vor Detailinformationen, die man auch in der Online-Hilfe findet.

Zielgruppe:	Software-Entwickler, die mit Microsoft Visual Studio 2005/2008 in C++ .NET-Anwendungen entwickeln wollen.
Voraussetzungen:	Kenntnisse im Umfang der Kurse "Teil 1: Einführung in Visual Studio 2005/2008" und "Teil 2: C/C++ Grundlagen" "Teil 3: Objektorientierte Programmierung"
Methodik:	Vortrag mit vielen Übungen, in denen praxisnahe Programme entwickelt werden
Ort und Zeit:	Dieses Seminar wird als Firmenseminar (inhouse) angeboten.
Dauer:	5 Tage

Inhalt

In der ersten Hälfte dieses Kurses wird der gesamte Sprachumfang von C++/CLI vorgestellt, und in der zweiten einige der wichtigsten .NET-Klassen.

1. C++/CLI

- Managed C++ und C++/CLI
- C++/CLI Basistypen, *String*, *Decimal* (ein präziser Gleitkommatyp)
- Die Basisklasse *System::Object*
- C++/CLI-Aufzählungstypen
- CLI-Arrays
- Handles und der Garbage Collected Heap (Managed Extensions)
- Referenzklassen (ref classes),
 - Destruktoren und Finalisierer
 - Virtuelle Funktionen, *new* und *override*
- Werteklassen (value classes)
 - initonly* und *literal* Elemente
 - sealed* Klassen und Funktionen
- Eigenschaften (Properties)
- Ereignisse (Events)
- Parameter-Arrays
- Selbst definierte Komponenten und ihre Ereignisse
- Interface-Klassen
- Delegates und Ereignisse
- Attribute
- Exception-Handling
- Die .NET Klassenhierarchie
- Die Erweiterung der Toolbox
- Laufzeit-Typinformationen mit der Klasse *Type*
- Generics
- Die STL/CLR in Visual C++ 2008
- Dokumentationskommentare und CHM-Hilfdateien
 - Dokumentationskommentare und XML-Dateien
 - Aus XML-Dateien Hilfdateien im CHM-Format erzeugen

2. Weitere .NET Steuerelemente

- Eigene Dialoge und die Klasse *MessageBox*
- Formatierte Texte mit *RichTextBox*
- Tabellen
- Steuerelemente zur Eingabe und Prüfung von Daten
 - Zulässige Benutzereingaben sicherstellen
 - Fehleranzeigen mit *ErrorProvider*
 - Die Anzeige von weiteren Formularen und selbstdefinierte Dialoge
 - Das *Validating*-Ereignis
 - Texteingaben mit einer *MaskedTextBox* filtern
 - Tastatureingaben in *KeyPress* filtern
- Symbolleisten und Statusanzeigen: *ToolStrip* und *ToolStripContainer*
- Größenänderung von Steuerelementen zur Laufzeit
 - Die Eigenschaften *Dock* und *Anchor*
 - SplitContainer*: Zur Größenanpassung von zwei Steuerelementen
 - TableLayoutPanel* und *FlowLayoutPanel*
- ListView* und *TreeView*

- ListView* nach Spalten sortieren
- MDI-Programme
- NotifyIcon*, *ToolTip* und *HelpProvider*

3. Elemente der .NET Klassenbibliothek

- Klassen für Uhrzeit und Kalenderdatum
 - Die Klassen *DateTime* und *TimeSpan*
 - Steuerelemente zur Eingabe von Kalenderdaten und Zeiten
 - Timer-Klassen
 - Hochauflösende Zeitmessung mit der Klasse *Stopwatch*
 - Kulturspezifische Datumsformate und Kalender
- Threads und ihre Synchronisation
 - Multithreading mit der Klasse *BackgroundWorker*
 - Multithreading mit der Klasse *Thread*
 - Kritische Abschnitte und die Synchronisation von Threads
 - Atomare Operationen mit *Interlocked*
 - ThreadPool*
- Grafiken zeichnen mit *PictureBox* und *Graphics*
 - Welt- und Bildschirmkoordinaten
 - Figuren, Farben, Stifte und Pinsel
 - Texte zeichnen
 - Drucken und Druckvorschau
 - 3D-Grafik mit *Viewport3D*
- Die Steuerung von MS-Office Anwendungen
 - Word
 - Excel
- Collection-Klassen
 - Die Collection-Klassen *ArrayList* und die generische Klasse *List<T>*
 - Die Interface-Klassen *ICollection<T>* und *IList<T>*
 - Die generischen Collection-Klassen *Queue<T>*, *LinkedList<T>* und *Stack<T>*
 - Dictionaries und die generische Interface-Klasse *IDictionary*
 - Die generischen Klassen *Dictionary*, *SortedDictionary* und *SortedList*
 - Spezielle Collection-Klassen
- .NET-Klassen zur Dateibearbeitung
 - StreamReader* und *StreamWriter* und *FileStream*
 - BinaryReader/Writer* und *StreamReader/Writer* mit *FileStreams*
 - Der gleichzeitige Zugriff auf eine Datei und Record-Locking
 - Komprimieren und Dekomprimieren von Dateien
 - Einige weitere Elemente aus dem Namensbereich *System::IO*
 - XML-Dateien
- Serialisierung
 - Serialisierung mit *BinaryFormatter*, *SoapFormatter* und *XmlSerializer*
- Datenbank-Komponenten
 - Die Verbindung zu einer Datenbank herstellen
 - Tabellen und die Komponente *DataSet*
 - Einige Elemente der Klasse *DataSet*
 - SQL-Abfragen
 - Datenbanken mit mehreren Tabellen
- Reguläre Ausdrücke
- Internet-Komponenten
 - Die *WebBrowser*-Komponente der Toolbox
 - Up- und Downloads mit der Klasse *WebClient*
 - Die Klassen *HTMLDocument* und *HTMLElement*

Emails versenden mit *SmtClient*
 URLs, URIs und IP-Adressen und zulässige Strings
 Netzwerkinformationen und die Klasse *Ping*
 TCP-Clients und Server mit *TcpClient* und *TcpListener*

Die Inhalte können an die Wünsche der Teilnehmer angepasst werden.

Kontakt: training@rkaiser.de , www.rkaiser.de
 R. Kaiser, Schwärzlocher Straße 53, 72070 Tübingen, Tel. 07071-41598, FAX 07071-44473

Stand: 14.4.2008

[Home](#)

Microsoft Visual C++ 2005/2008 Seminare

Diese Seminare werden von Prof. Richard Kaiser gehalten, der seit vielen Jahren Seminare für zahlreiche Firmen durchführt ([Referenzen](#)). Er ist der Verfasser der Bücher „[C++ mit dem Borland C++Builder](#)“ und „[C++ mit Microsoft Visual C++ 2008](#)“ (das im Lauf des Jahres 2008 erscheint) und ist Mitglied im [DIN Normierungsausschuss Informationstechnik NI-22](#).

Teil 5: Die C++-Standardbibliothek (STL) und Templates

Dieser Kurs ist der fünfte von fünf aufeinander abgestimmten Kursen, in denen der gesamte Sprachumfang des aktuellen ISO C++-Standards und die wichtigsten Erweiterungen von Visual C++ behandelt werden. Dabei stehen Zusammenhänge und Sprachkonzepte im Vordergrund vor Detailinformationen, die man auch in der Online-Hilfe findet.

Zielgruppe:	Software-Entwickler, die mit Microsoft Visual Studio 2005 in C++ programmieren wollen.
Voraussetzungen:	Kenntnisse im Umfang der Kurse "Teil 1: Einführung in Visual Studio 2005" und "Teil 2: C Grundlagen" "Teil 3: Objektorientierte Programmierung"
Methodik:	Vortrag mit vielen Übungen, in denen praxisnahe Programme entwickelt werden
Ort und Zeit:	Dieses Seminar wird als Firmenseminar (inhouse) angeboten.
Dauer:	5 Tage

Inhalt

Templates sind Vorlagen, aus denen der Compiler Klassen und Funktionen erzeugt. Die Standard Template Library (STL) ist eine Bibliothek, die zum C++-Standard 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 C++-Standardbibliothek (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.

Diese Themen überschneiden sich zum Teil mit denen von Teil 4 "C++/CLI-Erweiterungen und die .NET Klassenbibliothek". Die in diesem Kurs vorgestellten Techniken sind aber im Gegensatz zu den .NET-spezifischen von Teil 4 plattformübergreifend.

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*
 Iteratoren
 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 Streams und 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

Die Inhalte können an die Wünsche der Teilnehmer angepasst werden.

Kontakt: training@rkaiser.de, www.rkaiser.de

R. Kaiser, Schwärzlocher Straße 53, 72070 Tübingen, Tel. 07071-41598, FAX 07071-44473

Stand: 14.4.2008