

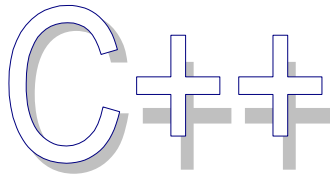
Kurse

Beschreibungen und Inhalte

Stand 16.07.2005

Ralf Schneeweiß
Gölzstraße 8
72072 Tübingen
Tel.: 0179/2292471
e-mail: ralf.schneeweiss@oop-trainer.de

<http://www.oop-trainer.de/>



Einführungskurs in C++

Themen

Historische Entwicklung von C++
Grundlegende Syntax
Erste Beispiele
Kennenlernen des Compilers
und des Linkers

Standarddatentypen
Strukturen
Operatoren
Funktionen
Zeiger und Referenzen
Dynamische Speicherallokation

Klassenkonzept
Datenkapselung
Attribute und Methoden
Instanziierung von Objekten
Konstruktoren und Destruktoren
Vererbung
Virtuelle Methoden
Abstrakte Klassen
Polymorphie

Funktionsüberladung
Operatorüberladung
Typenkonvertierung
Mehrfachvererbung
Sichtbarkeitskonzept

Exception Handling
Templates
Funktionstemplates
Klassentemplates

Aufbaukurs in C++

Themen

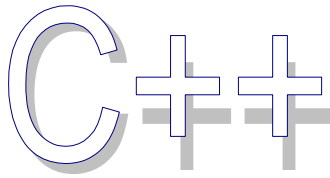
Überblick über die Standardbibliothek
Namespaces und die using Direktive
Der ANSI-C++ Standard
Streamklassen (Methoden,
Operationen und Manipulatoren)
Strings

Der Copy-Konstruktor
Virtuelle Destruktoren
Reimplementieren
Statische Elemente
Statische Methoden
Verschachtelte Klassen

Vererbung unterschiedlicher
Sichtbarkeit
Virtuelle Vererbung
Scopes
V-Tables und RTTI

Die Standard Template Library (STL)
Container im Allgemeinen
Listen
Vektoren
Sets
Multisets
Iteratoren
Algorithmen
Funktionsobjekte
Generische Programmierung

Idiome in C++
Der Einsatz von Exception Handling



Umsteigerkurs C++ für C-Programmierer

Themen

Historische Entwicklung von C++
Grundlegende Syntax im Vergleich zu C
Erste Beispiele

Klassenkonzept
Datenkapselung
Attribute, Methoden und Operatoren
Zeiger und Referenzen
Dynamische Speicherallokation

Instantiierung von Objekten
Konstruktoren und Destruktoren
Vererbung und Polymorphie
Virtuelle Methoden
Abstrakte Klassen

Funktionsüberladung
Operatorüberladung
Typenkonvertierung
Mehrfachvererbung
Sichtbarkeitskonzept

Exception Handling
Templates
Funktionstemplates
Klassentemplates

Die Objektorientierte Programmierung
Vergleich der Sprachen C und C++
Prinzipien der OO-Modellierung
Der Musterbegriff
Sinn und Unsinn von Klassenhierarchien
Diskussion der Abstraktionsebene der OOP
Diskussion von Argumenten für und gegen C++

Betrachtungen zum Laufzeitverhalten
Sprachstandard und Compilerverhalten

Einführung in die STL (Standard Template Library)

Themen

Abgrenzung: STL und C++
Standardbibliothek

Generische Programmierung
Die Konzepte der STL
Überblick über die STL

Container
Adapter
Listen
Vektoren
Deque
Sets und Multisets
Maps und Multimaps
Adapter

Iteratoren
Konstante und gegenläufige Iteratoren
Forward, Bidirectional und Random-
Access Iterator
Stream Iteratoren

Algorithmen
Modifizierende und nicht-modifizierende
Algorithmen

Funktionsobjekte
Allokatoren

Java

Einführungskurs in Java

Themen

Historische Entwicklung von Java
Grundlegende Syntax
Erste Beispiele
Kennenlernen des Compilers
und der VM

Standarddatentypen
Strings

Operatoren
Klassenkonzept
Methoden
Referenzen
Objektinstanziierung mit new
Garbagecollection

Datenkapselung
Attribute und Methoden
Konstruktoren
Vererbung
Die Methode finalize()
Interfaces
Polymorphie

Funktionsüberladung
Statische Methoden und Elemente
Typenkonvertierung
Sichtbarkeitsregeln
Exception Handling

Arrays
Pakete
Das Paket java.util
Abstract Windowing Toolkit
GUI Programmierung
Das Listenerkonzept

Aufbaukurs in Java

Themen

Verschachtelte Klassen
Statische Verschachtelung
Anonyme Klassen
Die Klasse Objekt
Die Klasse Class
Laden von Klassen zur Laufzeit
Auslesen von Typinformationen

Die Methoden clone() und equals()
Der statische Kontruktor

Das Streamkonzept von Java

Programmierung von Applets

Die Swing-Bibliothek
GUI Programmierung mit Swing
Herstellen einer Netzverbindung
Auslesen einer URL

Java Reflection
Remote Method Invocation (RMI)
Das Native Code Interface (NCI)
Objektserialisierung
Das JAR Utility

Die JDBC Datenbankschnittstelle

Java

Java Umsteigerkurs für C++ Programmierer

Themen

Historische Entwicklung von Java
Grundlegende Syntax
Erste Beispiele
Kennenlernen des Compilers
und der VM

Standarddatentypen
Strings
Operatoren
Klassenkonzept
Methoden
Referenzen
Objektinstanziierung mit new
Garbagecollection

Datenkapselung
Attribute und Methoden
Konstruktoren
Vererbung
Die Methode finalize()
Interfaces
Polymorphie
Funktionsüberladung
Statische Methoden und Elemente
Typenkonvertierung
Sichtbarkeitsregeln
Exception Handling

Arrays
Pakete
Das Paket java.util
Abstract Windowing Toolkit
GUI Programmierung
Das Listenerkonzept

OOP

Objektorientiertes Design in C++ mit Hilfe von Entwurfsmustern

Objektorientiertes Design in Java mit Hilfe von Entwurfsmustern

- Grundlegende Begriffe der Objektorientierten Programmierung: Klasse, Objekt, Vererbung, Assoziation, Aggregation, Komposition, Methode, Schnittstelle etc.
- Polymorphie und ihre Anwendungsbereiche
- Vererbung versus Aggregation
- Grundprinzipien des Objektorientierten Designs
- Kopplung, Kohäsion, Open/Close-Prinzip
- Die Mehrfachvererbung und ihre Problembereiche
- Ausführliche Diskussion der Themen
- Eine kurze Exkursion zu den drei Amigos (UML)
- Was sind Entwurfsmuster?
- Zielsetzung und Anwendung von Entwurfsmustern
- Klassifizierung
- Ausführliche Diskussion der Entwurfsmuster Adapter, Singleton, Proxy, Facade, Bridge, Abstract Factory, Factory Method, Iterator, Command, Composite, Decorator, Observer, Visitor, State und anderer.

C++ spezifische Beispiele für die Realisierung von Entwurfsmustern

Java – spezifische Beispiele für die Realisierung von Entwurfsmustern

Kursbegleitende Übungen in C++

Kursbegleitende Übungen in Java

Anwendung und Diskussion bestehender Frameworks wie z.B. die MFC aus dem Gesichtspunkt der Musterprogrammierung

Anwendung und Diskussion bestehender Frameworks wie z.B. das awt oder Swing aus dem Gesichtspunkt der Musterprogrammierung

UML

Einführung in das Konzept und die Elemente der Unified Modelling Language.

Es werden alle Diagrammartentypen der UML dargestellt und in den Kontext der objektorientierten Entwicklung gebracht. Dafür werden Beispiele herangezogen, die gemeinsam geübt und besprochen werden.

Die Inhalte des Kurses:

- Use-Cases
- Aktivitätsdiagramme
- Sequenzdiagramme
- Kollaborationsdiagramme
- Analyseklassendiagramme
- Entwurfsklassendiagramme
- Einführung in die Entwurfsmuster
- Zustandsdiagramme
- Moduldiagramme
- Installationsdiagramme

Diskussion der Diagrammtypen von dem Hintergrund einer prozessorientierten Softwareentwicklung.

Design Patterns

Entwurfsmuster im Prozeß der objektorientierten Softwareentwicklung

- Was sind Entwurfsmuster?
- Struktur-, Erzeuger- und Verhaltensmuster
- Besprechen und einüben aller Muster des Buches von Erich Gamma, Richard Helm, Ralph Johnson und John Vlissides: Entwurfsmuster; Elemente wiederverwendbarer objektorientierter Software.

Besprochen werden die Muster:

- ◆ Abstrakte Fabrik
- ◆ Erbauer
- ◆ Fabrikmethode
- ◆ Prototyp
- ◆ Singleton
- ◆ Adapter
- ◆ Brücke
- ◆ Dekorierer
- ◆ Fassade
- ◆ Fliegengewicht
- ◆ Kompositum
- ◆ Proxy
- ◆ Befehl
- ◆ Beobachter
- ◆ Besucher
- ◆ Interpreter
- ◆ Iterator
- ◆ Memento
- ◆ Schablonenmethode
- ◆ Strategie
- ◆ Vermittler
- ◆ Zustand
- ◆ Zuständigkeitskette

Die meisten Muster werden durch praktische Übungen in C++ oder Java realisiert.

Der Kurs kann mit C++ oder Java durchgeführt werden. Auf Wunsch können auch beide Programmiersprachen eingesetzt werden.

Frameworks

Entwurfsprinzipien von Frameworks und Komponententechnologien in der objektorientierten Softwareentwicklung.

- Was sind Frameworks? Abgrenzung zur Library.
- Was sind Komponententechnologien?
- Abgrenzung der Begriffe Framework und Komponentenarchitektur.
- Abgrenzung von Architektur und Entwurf.
- Besprechen einiger verbreiteter Frameworks und Komponententechnologien: MFC, AWT, Swing, COM, JavaBeans, SOAP, WebServices.
- Entwurfsmuster als Grundprinzipien der Frameworkarchitektur:
 - Schablonenmethoden
 - Observer
 - Command
 - Factory
 - Bridge
 - Prototype
- Entwurfsmuster als Grundprinzipien von Komponentenarchitektur:
 - Proxy
 - Fassade
- Architekturmuster
 - Reflection-Pattern
- Runtime Typeinformation in C++ und Java, generelle Voraussetzungen der beiden Sprachen für Framework- und Komponentenarchitekturen.
- Die Rolle von IDL und XML.
- Die zentrale Bedeutung des Typsystems.
- Middleware. Beispiele: RMI und CORBA.
- Enterprise Architekturen: J2EE und .NET
- Schichtenmodelle unter dem Gesichtspunkt des Einsatzes von Frameworks und Komponenten.

Embedded System

C++ Entwicklung im Embedded System

- Wie verhält sich C++ im Vergleich zu C im Embedded System?
- Welche Entscheidungen fällt der ANSI-C++ Standard für die Embedded Entwicklung?
- Welche Grundsätzlichen Fehler kann man beim Einsatz von C++ machen?
- Welche Auswirkungen haben unterschiedliche Speicherverwaltungsstrategien für Applikationsdaten auf das Gesamtsystem?
- Welche praktischen Auswirkungen haben physikalische und virtuelle Speicherverwaltung auf das Design von Software?
- Was bedeutet Speicherfragmentierung und wie beherrscht man Probleme damit?
- Welche Auswirkungen haben der Einsatz von Polymorphie auf Laufzeit und Größe des Programmcodes?
- Wie können Templates eingesetzt werden?
- Laufzeitaspekte der Standard Template Library.
- Exception Handling im Embedded System: Laufzeit, Strukturierung und Footprint.
- Fehlerbehandlung und Code-Coverage.
- Entwicklung nebenläufiger Systeme.
- Der C++-Compiler
- Der Linker
- Simulation

Multithreading

Seminar:

Standardprobleme und deren Lösungen in der Programmierung parallel ablaufender Threads

Themen:

Prozesse und Threads

POSIX, Pthreads

Threads unter MS Windows

Synchronisierung

Deadlocks

Architekturen

Critical Sections, Spin-Lock Verfahren

Semaphore

Mutexe

Shared Memory