

| | | | |
|--|-----------|--|------------|
| 1 Einführung in C++ | 4 | 6.3 Übung | 61 |
| 1.1 Die Entstehung von C++ | 4 | 7 Polymorphismus | 62 |
| 1.2 Standards..... | 5 | 7.1 Einleitung | 62 |
| 1.3 Bezeichnung der Dateinamen | 5 | 7.2 Rein virtuelle Funktionen..... | 62 |
| 2 Einfache Erweiterungen von C++ | 6 | 7.3 Abstrakte Basisklassen | 63 |
| 2.1 Kommentarzeilen..... | 6 | 7.4 Übung | 65 |
| 2.2 Deklarationen | 6 | 8 Datei- und String-Streams | 66 |
| 2.3 Der Operator :: | 7 | 8.1 Die I/O-Stream-Klassenhierarchie | 66 |
| 2.4 Defaultwerte | 7 | 8.2 Datei-Streams | 68 |
| 2.4.1 Syntax für den Einsatz | | 8.3 String-Streams | 70 |
| von Defaultwerten | 8 | 8.3.1 Zerlegen von Strings: istrstream | 71 |
| 2.5 Inline-Funktionen | 9 | 8.3.2 Zusammensetzen von | |
| 2.6 Mischen von C- und C++-Quellen..... | 10 | Strings: ostrstream | 72 |
| 2.7 Dynamische Speicherverwaltung..... | 10 | 8.4 Übung | 73 |
| 2.7.1 New | 11 | 9 Überladen von Operatoren | 74 |
| 2.7.2 Delete | 12 | 9.1 Einleitung | 74 |
| 2.7.3 Fehlerbehandlung über new-Handler | 13 | 9.2 Operator-Overloading durch friend- | |
| 2.8 Typinformationen..... | 14 | Funktionen | 75 |
| 2.9 Referenzen und Referenzvariablen | 14 | 9.3 Operator-Overloading mit | |
| 2.10 Referenzparameter..... | 16 | Elementfunktionen..... | 77 |
| 2.11 Referenzen als Funktionsergebnisse..... | 18 | 9.4 Allgemeines zum Operator-Overloading..... | 79 |
| 2.12 Überladen von Funktionen..... | 19 | 9.5 Typumwandlungs-Operatoren | 79 |
| 2.13 Übung..... | 21 | 9.5.1 Konstruktoren als Typumwandlungs- | |
| 3 Streams | 22 | Operatoren | 81 |
| 3.1 Einführung..... | 22 | 9.5.2 Typumwandlungsoperator-Funktionen... | 82 |
| 3.2 Standardausgabestream cout..... | 23 | 9.6 Kopieren von Objekten..... | 84 |
| 3.3 Standardeingabestream cin..... | 26 | 9.7 Überladen des Funktionsoperators () | 87 |
| 3.4 Übung..... | 29 | 9.8 Überladen des Komponentenzugriffs -> | 90 |
| 4 Klassen..... | 30 | 9.9 Überladen von new und delete..... | 91 |
| 4.1 Klassen, Instanzen und Objekte | 30 | 9.10 Übung | 94 |
| 4.1.1 Einfache Syntax der | | 10 Templates..... | 96 |
| Klassendeklaration..... | 31 | 10.1 Einführung..... | 96 |
| 4.1.2 Instanzen einer Klasse..... | 32 | 10.2 Funktionen-Templates..... | 97 |
| 4.1.3 Zugriff auf die Members einer Klasse | 32 | 10.2.1 Syntax für Funktionen-Templates | 97 |
| 4.2 Konstruktoren und Destruktoren..... | 34 | 10.3 Klassen-Templates..... | 98 |
| 4.3 Friends | 35 | 10.3.1 Syntax für Klassen-Templates | 99 |
| 4.4 Statische Klasselemente | 36 | 10.4 Übung | 101 |
| 4.5 Übung..... | 37 | 11 Exceptionhandling | 102 |
| 5 Vererbung..... | 38 | 11.1 Einführung..... | 102 |
| 5.1 Einführung..... | 38 | 11.1.1 Syntax der Fehlerbehandlung..... | 103 |
| 5.2 Syntax der Vererbung..... | 40 | 11.2 Übung | 105 |
| 5.3 Art der Ableitung..... | 40 | 12 Objektorientierter Entwurf..... | 106 |
| 5.4 Strukturen und Klassen | 43 | 12.1 Einführung..... | 106 |
| 5.5 Konstruktoren und Destruktoren | | 12.2 Das Objektmodell..... | 107 |
| abgeleiteter Klassen..... | 44 | 12.3 Identifizieren von Klassen | 108 |
| 5.6 Kompatibilität in Klassenhierarchien | 46 | 12.4 Verifizieren der Klassen durch | |
| 5.6.1 Zuweisung von Objekten | | Szenarios | 109 |
| abgeleiteter Klassen..... | 46 | 12.5 Gemeinsamkeiten und Beziehungen | |
| 5.6.2 Zeiger auf Basisklassen und | | zwischen Klassen..... | 109 |
| abgeleitete Klassen..... | 47 | 12.6 Verfeinerung durch Identifizieren | |
| 5.7 Klassenhierarchien | 49 | der Daten | 110 |
| 5.8 Der this-Zeiger..... | 50 | 12.7 Graphische Darstellung von Klassen | |
| 5.9 Virtuelle Funktionen..... | 50 | und Beziehungen | 111 |
| 5.10 Virtuelle Destruktoren | 52 | 12.8 Design-Fehler..... | 114 |
| 5.11 Übung..... | 55 | 12.9 Übung | 114 |
| 6 Mehrfachvererbung..... | 56 | Stichwortverzeichnis | 116 |
| 6.1 Mehrfachvererbung | 56 | | |
| 6.2 Virtuelle Basisklassen..... | 58 | | |