

Inhaltsverzeichnis

Vorwort	IX
1 Motivation	1
2 Die Entwicklung von C++	3
3 C++-Erweiterungen	5
3.1 Kommentare	5
3.2 Der Typ <code>void</code>	5
3.3 Funktionsprototypen	6
3.4 Default-Parameterwerte	7
3.5 Overloading von Funktionen	8
3.6 Referenzen	9
3.6.1 Referenzen in Deklarationen	9
3.6.2 Referenzen bei Funktionsparametern	10
3.7 Deklarationen von Variablen als Anweisung	11
3.8 Der scope-resolution-Operator <code>::</code>	11
3.9 Die <code>inline</code> -Deklaration	12
3.10 <code>const</code> - und <code>volatile</code> -Deklarationen	13
3.11 Explizite Typumwandlung	15
3.11.1 C-Cast-Operatoren	15
3.11.2 Neue C++-Cast-Operatoren	16
3.12 Die <code>enum</code> -Deklaration	18
3.13 Der Typ <code>bool</code>	19
3.14 <code>struct</code> und <code>union</code> als Typnamen	20
3.15 Freispeicherverwaltung	20
3.16 Vereinfachte Terminal Ein-/Ausgabe	22
4 Objektklassen	25
4.1 Einführung und Begriffserklärung	25
4.1.1 Objektklasse	25
4.1.2 Methoden; Member-Funktionen; Elementfunktionen	26
4.1.3 Member-Variablen; Elementdaten	26
4.1.4 Instanz; Objekt; Element	26
4.2 Definition von Klassen in C++	26
4.3 Methoden	29
4.3.1 <code>inline</code> -Methoden	31

4.3.2	<code>const</code> -Member-Funktionen	32
4.4	Sichtbarkeitsbeschränkungen	33
4.5	<code>friend</code> -Funktionen und -Klassen	36
4.6	Konstruktoren und Destruktoren	39
4.6.1	Besonderheiten	41
4.7	Der <code>this</code> -Pointer	42
4.8	Statische Member	43
4.9	Zeiger auf Klasselemente	44
4.10	Memberobjekte	46
4.11	Verschachtelte Klassen	47
5	Namensräume	49
5.1	Grundlagen	49
5.2	Unbenannte Namensräume	50
5.3	Aliase	51
5.4	Die <code>using</code> -Anweisung	52
6	Operator-Overloading	55
6.1	Einführung	55
6.2	Overloading unärer Operatoren	56
6.3	Overloading binärer Operatoren	58
6.4	Typumwandlungen	60
6.4.1	Konstruktoren zur Typumwandlung	60
6.4.2	Explizite Konstruktoren	61
6.4.3	Konvertierungsoperatoren	63
6.4.4	Beispielimplementation von <code>complex</code>	64
6.5	Funktionswahl	65
6.6	Beschränkungen beim Overloading	66
6.7	Zuweisung und Initialisierung	67
6.8	Overloading von <code>++</code> und <code>--</code>	69
6.9	Overloading von <code>[]</code> und <code>()</code>	70
6.9.1	Der Index-Operator <code>[]</code>	70
6.9.2	Der Funktionsoperator <code>()</code>	72
7	Vererbung (abgeleitete Klassen)	77
7.1	Einführung	77
7.2	Ableiten von Klassen in C++	78
7.3	Zugriffsmöglichkeiten	80
7.4	Sichtbarkeitseinschränkungen	81
7.5	Zeiger	83
7.6	Konstruktoren und Destruktoren	84
7.7	Einige Beispiele zur Sichtbarkeit	85

Inhaltsverzeichnis	VII
8 Virtuelle Funktionen	89
8.1 Einführung	89
8.2 Virtuelle Funktionen in C++	92
8.3 pure virtual-Funktionen	95
9 Mehrfachvererbung	97
9.1 Einführung	97
9.2 Mehrfachvererbung in C++	98
9.3 Virtuelle Basisklassen	99
10 Laufzeit-Typinformationen	103
10.1 Dynamische Casts	103
10.2 Ermitteln von Typinformationen	104
11 Templates	107
11.1 Funktions-Templates	107
11.2 Klassen-Templates	109
12 Ausnahmebehandlung	113
12.1 Einführung	113
12.2 Ausnahmebehandlung in C++	114
12.3 Deklaration von Ausnahmen	116
12.4 Ausnahmen der Standardbibliothek	117
13 Eingabe – Ausgabe	119
13.1 Einführung	119
13.2 Standard Ein-/Ausgabe	121
13.2.1 Ausgabe	121
13.2.2 Eingabe	122
13.3 Ein-/Ausgabe von benutzerdefinierten Datentypen	123
13.4 Ein-/Ausgabe-Steuerung	125
13.4.1 Kontrolle des Ausgabeformats	125
13.4.2 Manipulatoren	129
13.4.3 Der Status eines Streams	132
13.5 Dateien und Streams	133
13.6 Strings und Streams	135
13.7 Positionieren innerhalb eines Streams	137
14 C und C++	141
14.1 C Source-Code	141
14.2 C Library-Funktionen	142
Literatur	145
Index	147