

Inhalt

Vorwort	XI
1 Die Grundlagen	1
1.1 Einführung	1
1.2 Programme	1
1.3 Hello, World!	2
1.4 Funktionen	3
1.5 Typen, Variablen und Arithmetik	5
1.6 Gültigkeitsbereich und Lebensdauer	8
1.7 Konstanten	9
1.8 Zeiger, Arrays und Referenzen	10
1.9 Tests	13
1.10 Ratschläge	15
2 Benutzerdefinierte Typen	17
2.1 Einführung	17
2.2 Strukturen	17
2.3 Klassen	19
2.4 Unions	21
2.5 Aufzählungen	22
2.6 Ratschläge	23
3 Modularität	25
3.1 Einführung	25
3.2 Separate Kompilierung	26
3.3 Namespaces	28
3.4 Fehlerbehandlung	29
3.4.1 Ausnahmen	29
3.4.2 Invarianten	30
3.4.3 Statische Assertionen	32
3.5 Ratschläge	33

4	Klassen	35
4.1	Einführung	35
4.2	Konkrete Typen	36
4.2.1	Ein arithmetischer Typ	37
4.2.2	Ein Container	38
4.2.3	Container initialisieren	40
4.3	Abstrakte Typen	41
4.4	Virtuelle Funktionen	44
4.5	Klassenhierarchien	45
4.5.1	Explizites Überschreiben	47
4.5.2	Vorzüge von Hierarchien	48
4.5.3	Navigation in Hierarchien	50
4.5.4	Ressourcenlecks vermeiden	50
4.6	Kopieren und Verschieben	51
4.6.1	Container kopieren	52
4.6.2	Container verschieben	53
4.6.3	Wichtige Operationen	55
4.6.4	Ressourcenverwaltung	57
4.6.5	Operationen unterdrücken	59
4.7	Ratschläge	59
5	Templates	63
5.1	Einführung	63
5.2	Parametrisierte Typen	63
5.3	Funktions-Templates	65
5.4	Konzepte und generische Programmierung	66
5.5	Funktionsobjekte	68
5.6	Variadische Templates	70
5.7	Alias	71
5.8	Modell der Template-Kompilierung	72
5.9	Ratschläge	73
6	Überblick über die Bibliothek	75
6.1	Einführung	75
6.2	Komponenten der Standardbibliothek	76
6.3	Header und Namespace der Standardbibliothek	77
6.4	Ratschläge	78
7	Strings und reguläre Ausdrücke	79
7.1	Einführung	79
7.2	Strings	79
7.2.1	Eine <code>string</code> -Implementierung	81

7.3	Reguläre Ausdrücke	82
7.3.1	Suchen	83
7.3.2	Notation regulärer Ausdrücke	84
7.3.3	Iteratoren	88
7.4	Ratschläge	89
8	E/A-Streams	91
8.1	Einführung	91
8.2	Ausgabe	92
8.3	Eingabe	93
8.4	E/A-Status	95
8.5	Ein-/Ausgabe von benutzerdefinierten Typen	96
8.6	Formatierung	97
8.7	File Streams	98
8.8	String-Streams	99
8.9	Ratschläge	100
9	Container	103
9.1	Einführung	103
9.2	vector	103
9.2.1	Elemente	106
9.2.2	Bereichsüberprüfung	106
9.3	list	108
9.4	map	109
9.5	unordered_map	110
9.6	Überblick über Container	111
9.7	Ratschläge	113
10	Algorithmen	115
10.1	Einführung	115
10.2	Iteratoren verwenden	116
10.3	Iteratortypen	119
10.4	Stream-Iteratoren	120
10.5	Prädikate	122
10.6	Überblick über Algorithmen	122
10.7	Containeralgorithmen	123
10.8	Ratschläge	124
11	Utilities	125
11.1	Einführung	125
11.2	Ressourcenverwaltung	125
11.2.1	unique_ptr und shared_ptr	126

11.3 Spezialisierte Container	129
11.3.1 array	130
11.3.2 bitset	131
11.3.3 pair und tuple	132
11.4 Zeit	134
11.5 Funktionsadapter	134
11.5.1 bind()	135
11.5.2 mem_fn()	135
11.5.3 function	136
11.6 Typfunktionen	137
11.6.1 iterator_traits	138
11.6.2 Typprädikate	140
11.7 Ratschläge	140
12 Numerik	143
12.1 Einführung	143
12.2 Mathematische Funktionen	143
12.3 Numerische Algorithmen	144
12.4 Komplexe Zahlen	145
12.5 Zufallszahlen	146
12.6 Vektorarithmetik	148
12.7 Numerische Grenzen	148
12.8 Ratschläge	149
13 Nebenläufigkeit	151
13.1 Einführung	151
13.2 Tasks und Threads	152
13.3 Argumente übergeben	153
13.4 Ergebnisse zurückgeben	154
13.5 Daten gemeinsam nutzen	154
13.6 Warten auf Ereignisse	156
13.7 Kommunizierende Tasks	158
13.7.1 future und promise	158
13.7.2 packaged_task	159
13.7.3 async()	160
13.8 Ratschläge	161
14 Geschichte und Kompatibilität	163
14.1 Historische Anmerkungen	163
14.1.1 Chronik	164
14.1.2 Die frühen Jahre	165
14.1.3 Die ISO-C++-Standards	167

14.2 C++11-Erweiterungen	169
14.2.1 Sprachfeatures	169
14.2.2 Komponenten der Standardbibliothek	170
14.2.3 Veraltete Features	171
14.2.4 Typumwandlungen	172
14.3 C/C++-Kompatibilität	173
14.3.1 C und C++ sind Geschwister	173
14.3.2 Kompatibilitätsprobleme	175
14.3.2.1 Stilprobleme	175
14.3.2.2 void*	177
14.3.2.3 C++-Schlüsselwörter	177
14.3.2.4 Bindung	178
14.4 Literaturhinweise	178
14.5 Ratschläge	181
Index	183