

Auf einen Blick

TEIL I Elementare Konzepte aus C++03

C++	27
-----------	----

TEIL II Neue Sprachmechanismen

Initialisierung	61
Typinferenz	76
Syntax	89
Lambdas und Funktoren	114
Attribute	139
Verschiebesemantik	156
Designfragen	196
Templates	221
Sprachfeatures	232
Zeichenketten	243

TEIL III Neues zu Containern, Pointern und Algorithmen

Neue Container	261
Zeigertypen	275
Hilfsmittel für Bekanntes	295
Änderungen für Container	300

TEIL IV Neues in der Standardbibliothek

Multithreading	321
Spezielle Bibliotheksaufgaben	355
Zufallszahlen	368
Fehlerbehandlung	377

Anhang

Abgewöhnen	397
Wahrscheinlichkeitsverteilungen	399
Compilersupport für C++11	402
Kompilierfähigkeit der Listings	405

Inhalt

Vorwort	11
0 Beispielkapitel	17

TEIL I: Elementare Konzepte aus C++03

C++

1 Resource Acquisition Is Initialization	27
2 Sicherheit durch Typen	36
3 Erinnerung: Exception Safety	43
4 C++-Standardbibliothek als Lösung	48

TEIL II: Neue Sprachmechanismen

Initialisierung

5 Vereinheitlichte Initialisierung	61
6 Selbst geschriebene Initialisierung mit Listen	67
7 Beschränkte automatische Konvertierung	74

Typinferenz

8 Typinferenz bei der Initialisierung mit auto	76
9 Const-Iteratoren mit cbegin und cend	80
10 Typinformationen mit decltype	83

Syntax

11 Das neue Range-For	89
12 nullptr ist das neue NULL	100
13 Garantiert konstante Ausdrücke	102
14 Sicherheit beim Kompilieren	109

Lambdas und Funktoren

15	Funktoren lokal definieren	114
16	Lambdas: anonyme Funktionen	118
17	Zugriffsdeklaration für Lambda	124
18	Funktionspointer sind out	135

Attribute

19	Explizites Überschreiben mit override	139
20	Finale Klassen und Methoden	145
21	Attribute, so viele Sie möchten	152

Verschiebesemantik

22	Kopieren, Verschieben und Weiterleiten	156
23	Klassen fürs Verschieben entwerfen	163
24	Kein Return von RValue-Referenzen	176
25	RValue-Referenzen für Perfect Forwarding	181
26	Überladen auf RValues	190

Designfragen

27	Methoden per delete und default	196
28	Konstruktor-Delegation	201
29	Membervariablen initialisieren	210
30	Konstruktoren erben	213
31	Explizite Konvertierung	216

Templates

32	Templates mit variabler Argumentanzahl	221
33	Alias für Templates	229

Sprachfeatures

34	Klassen-Enums sind typsicher	232
35	Keine Exceptions bei noexcept	236

Zeichenketten

36	Unterstützung von Unicode	243
37	Rohstringliterale	247
38	Rezept für benutzerdefinierte Literale	250

TEIL III: Neues zu Containern, Pointern und Algorithmen

Neue Container

39	Tupel, das bessere Paar	261
40	Ungeordnete Container	265
41	Neue Liste nur für vorwärts	271

Zeigertypen

42	unique_ptr statt auto_ptr	275
43	Smarte Pointer	285

Hilfsmittel für Bekanntes

44	Mehr Algorithmen für Container	295
----	--------------------------------------	-----

Änderungen für Container

45	Sets sind nicht mehr modifizierbar	300
46	Standardcontainer passend schrumpfen	302
47	function und bind	306
48	Die Verwendung von swap	313

TEIL IV: Neues in der Standardbibliothek

Multithreading

49	Parallel arbeiten	321
50	Threads als Basis für Parallelität	326
51	Futures und Promises mit async	329
52	Lebensdauer von Daten	337
53	Mutexe, Locks und mehr	341
54	Memory Order und Fences	348

Inhalt

Spezielle Bibliotheksaufgaben

55	Reguläre Ausdrücke	355
56	Ende der Umrechnungsfehler	362
57	Rechnen mit Zeiteinheiten	365

Zufallszahlen

58	Wahrscheinlichkeitsverteilungen	368
59	Der richtige Würfel	372

Fehlerbehandlung

60	Hierarchie der Exceptions	377
61	Fehlerfälle mit Fehlercode	380
62	Eigene Fehlerwerte erstellen	387

Anhang

A	Abgewöhnen	397
B	Wahrscheinlichkeitsverteilungen	399
C	Compilersupport für C++11	402
D	Kompilierfähigkeit der Listings	405
	Index	407