

Übersicht

Einführung	15
Teil I Tour de C++11	27
1 Die Standardisierung	29
2 Ziele von C++11	33
3 Kernsprache	35
4 Multithreading	69
5 Die Standardbibliothek	87
Teil II Kernsprache	117
6 Usability	119
7 Entwurf von Klassen	147
8 Rvalue-Referenzen	173
9 Generische Programmierung	197
10 Erweiterte Datenkonzepte und Literale	209
11 Removed und Deprecated	239
Teil III Multithreading	243
12 Das C++11-Speichermodell	245
13 Atomare Datentypen	249
14 Threads	255
15 Gemeinsam von Threads genutzte Daten	267
16 Thread-lokale Daten	287
17 Synchronisation der Threads	291
18 Asynchrone Aufgaben	301

Übersicht

Teil IV Standardbibliothek	319
19 Neue Bibliotheken	321
20 Verbesserte Bibliotheken	397
Teil V Ausblick	481
21 Der nächste C++ Standard	483
Teil VI Anhang	493
A Build-Umgebung installieren	495
B Funktionsobjekte	499
C Resource Acquisition Is Initialization	503
D Promotion Trait	507
E Implizit erzeugte Methoden und Operatoren	511
F Funktionale Programmierung	515
Praxistipps	535
Literaturverzeichnis	537
Index	543

Inhalt

Einführung	15
Fragen und Antworten	15
Was ist C++11?	15
Warum ein Buch über C++11?	16
Welches Ziel verfolgt das Buch?	17
Übersicht	17
Didaktik	18
Übungsaufgaben	18
Exkurse	19
Anhang	19
Praxis	19
Codekonventionen	19
Viele Beispiele	19
Zusatzmaterial	19
Aktuelle Compiler-Unterstützung	20
GCC	21
Microsoft Visual C++	23
Standard	25
Danksagung	25
Kontakt	25
Teil I Tour de C++11	27
1 Die Standardisierung	29
2 Ziele von C++11	33
3 Kernsprache	35
3.1 Usability	35
3.1.1 Die Range-basierte For-Schleife	36
3.1.2 Das automatische Ableiten von Typen	38
3.1.3 Lambda-Funktionen	40
3.1.4 Vereinheitlichte Initialisierung	43

Inhalt

3.2	Entwurf von Klassen	45
3.2.1	Mächtiger Initialisierung	45
3.2.2	Explizite Klassendefinitionen	51
3.3	Rvalue-Referenzen	54
3.4	Generische Programmierung	58
3.4.1	Variadic Templates	58
3.4.2	Zusicherungen zur Compile-Zeit	62
3.4.3	Aliase Templates	62
3.5	Erweiterte Datenkonzepte und Literale	64
3.5.1	Konstante Ausdrücke	64
3.5.2	Plain Old Data (POD)	65
3.5.3	Unbeschränkte Unions	65
3.5.4	Streng typisierte Aufzählungstypen	65
3.5.5	Neue String-Literale	66
3.5.6	nullptr	66
3.6	Weitere Aufräumarbeiten und Integration von C99	67
3.6.1	Aufräumarbeiten	67
3.6.2	Integration von C99	67
4	Multithreading	69
4.1	Threads	71
4.1.1	Erzeugung von Threads	71
4.1.2	Schutz von Daten	75
4.1.3	Sichere Initialisierung der Daten	78
4.2	Thread-lokale Daten	80
4.3	Synchronisation von Threads	80
4.4	Asynchrone Aufgaben	83
5	Die Standardbibliothek	87
5.1	Neue Bibliotheken	89
5.1.1	Reguläre Ausdrücke	89
5.1.2	Type Traits	93
5.1.3	Zufallszahlen	96
5.1.4	Zeitbibliothek	97
5.1.5	Referenz-Wrapper	98
5.2	Verbesserte Bibliotheken	100
5.2.1	Smart Pointer	100
5.2.2	Neue Container	106
5.2.3	Neue Algorithmen	114
5.2.4	bind and function	115

Teil II	Kernsprache	117
6	Usability	119
6.1	Die Range-basierte For-Schleife	119
6.2	Automatische Typableitung	122
6.2.1	auto	122
6.2.2	decltype	126
6.2.3	Alternative Funktionssyntax	128
6.3	Lambda-Funktionen	131
6.3.1	Lambda-Funktionen als temporäre Funktionsobjekte	132
6.3.2	Bindung an den lokalen Bereich: []	134
6.3.3	Argumente: {}	139
6.3.4	Rückgabewert: →	139
6.3.5	Funktionskörper: {}	139
6.4	Vereinheitlichte Initialisierung	142
7	Entwurf von Klassen	147
7.1	Initialisierung von Objekten	147
7.1.1	Initialisiererlisten für Konstruktoren	147
7.1.2	Delegation von Konstruktoren	151
7.1.3	Vererbung von Konstruktoren	152
7.1.4	Direktes Initialisieren von Klasselementen	155
7.2	Explizite Klassendefinitionen	157
7.2.1	default und delete	157
7.2.2	override und final	165
7.2.3	Expliziter Konvertierungsoperator	167
8	Rvalue-Referenzen	173
8.1	Move-Semantik	180
8.2	Perfect Forwarding	191
9	Generische Programmierung	197
9.1	Variadic Templates	197
9.2	Zusicherungen zur Compile-Zeit	203
9.3	Aliase Templates	206
10	Erweiterte Datenkonzepte und Literale	209
10.1	Konstante Ausdrücke	209
10.2	Plain Old Data (POD)	215

Inhalt

10.3	Unbeschränkte Unions	218
10.4	Streng typisierte Aufzählungstypen	221
10.5	Raw-String-Literale	225
10.6	Unicode-Unterstützung	226
10.7	Benutzerdefinierte Literale	230
10.8	nullptr	234
11	Removed und Deprecated	239
11.1	Removed	239
11.2	Deprecated	240
Teil III	Multithreading	243
12	Das C++11-Speichermodell	245
13	Atomare Datentypen	249
14	Threads	255
14.1	Erzeugen von Threads	255
14.2	Lebenszeit der Daten	257
14.3	Operationen auf Threads	262
15	Gemeinsam von Threads genutzte Daten	267
15.1	Schutz der Daten	267
15.1.1	Mutexe	267
15.1.2	Locks	272
15.2	Sichere Initialisierung der Daten	280
15.2.1	Konstante Ausdrücke	280
15.2.2	call_once und once_flag	284
15.2.3	Statische Variablen in einem Blockbereich	285
16	Thread-lokale Daten	287
17	Synchronisation der Threads	291
18	Asynchrone Aufgaben	301
18.1	async	302
18.2	packaged_task	305
18.3	future und promise	310

Teil IV Standardbibliothek	319
19 Neue Bibliotheken	321
19.1 Reguläre Ausdrücke	321
19.1.1 Syntax der regulären Ausdrücke	324
19.1.2 Objekte vom Typ regulärer Ausdruck	331
19.1.3 Analyse des Suchergebnisses mit <code>match_results</code>	334
19.1.4 Exakte Treffer mit <code>regex_match</code>	338
19.1.5 Suchen mit <code>regex_search</code>	340
19.1.6 Ersetzen mit <code>regex_replace</code>	343
19.1.7 Formatieren mit <code>regex_replace</code> und <code>match_results.format</code>	346
19.1.8 Wiederholtes Suchen mit <code>regex_iterator</code> und <code>regex_token_iterator</code>	349
19.2 Type-Traits	357
19.2.1 Typeeigenschaften abfragen	361
19.2.2 Typen vergleichen	369
19.2.3 Typen transformieren	369
19.3 Zufallszahlen	373
19.3.1 Zufallszahlenerzeuger	377
19.3.2 Zufallszahlenverteilung	379
19.4 Zeitbibliothek	381
19.4.1 Zeitdauer	383
19.4.2 Zeitgeber	387
19.4.3 Zeitpunkt	389
19.5 Referenz-Wrapper	390
19.5.1 Die Hilfsfunktionen <code>ref</code> und <code>cref</code>	393
20 Verbesserte Bibliotheken	397
20.1 Smart Pointer	397
20.1.1 <code>unique_ptr</code>	397
20.1.2 <code>shared_ptr</code>	407
20.1.3 <code>weak_ptr</code>	420
20.2 Neue Container	430
20.2.1 Tupel	430
20.2.2 Array	441
20.2.3 Einfach verkettete Liste	445
20.2.4 Hashtabellen	448
20.3 Neue Algorithmen	464
20.3.1 Praktische Helferlein	468

Inhalt

20.4	bind und function	471
20.4.1	bind	472
20.4.2	function	473
Teil V	Ausblick	481
21	Der nächste C++ Standard	483
21.1	Für C++11 geplant	484
21.1.1	Module	484
21.1.2	Spezielle mathematische Funktionen	484
21.1.3	Concepts	485
21.2	Technical Report 2	487
21.2.1	Erweiterte Thread-Funktionalität	488
21.2.2	Netzwerkunterstützung	488
21.2.3	Signale und Slots	489
21.2.4	Dateisystem-Bibliothek	489
21.2.5	Boost Any-Bibliothek	490
21.2.6	Bibliothek zur lexikalischen Konvertierung	491
21.2.7	Neue String-Algorithmen	492
Teil VI	Anhang	493
A	Build-Umgebung installieren	495
A.1	Aktueller C++-Compiler	496
A.1.1	GNU Compiler Collection (GCC)	496
A.1.2	Visual C++ 2010 Express	497
A.2	Boost-Bibliothek	497
B	Funktionsobjekte	499
B.1	Wie funktioniert ein Funktionsobjekt?	499
B.2	Welche Vorteile bietet ein Funktionsobjekt?	500

C Resource Acquisition Is Initialization 503

D Promotion Trait 507

E Implizit erzeugte Methoden und Operatoren 511

F Funktionale Programmierung 515

 F.1 Programmieren mit mathematischen Funktionen 516

 F.2 Charakteristiken funktionaler Programmierung 517

 F.2.1 First-class functions 517

 F.2.2 Funktionen höherer Ordnung 519

 F.2.3 Reine Funktionen 525

 F.2.4 Rekursion 527

 F.2.5 Verarbeitung von Listen 530

 F.2.6 Bedarfsauswertung 531

Praxistipps 535

Literaturverzeichnis 537

Index 543