

Ulrich Breymann

C++

Eine Einführung

5., aktualisierte und erweiterte Auflage



HANSER

Inhaltsverzeichnis

Teil I	Einführung in die Sprache	25
1	Einführung	27
1.1	Compiler	30
2	Grundlegende Begriffe	31
2.1	Entwicklung eines einfachen Programms	31
2.2	Einfache Datentypen und Operatoren	40
2.2.1	Ausdruck	40
2.2.2	Ganze Zahlen	40
2.2.3	Reelle Zahlen	45
2.2.4	Komplexe Zahlen	51
2.2.5	Zeichen	52
2.2.6	Logischer Datentyp	56
2.2.7	Referenzen	58
2.3	Gültigkeitsbereich und Sichtbarkeit	58
2.4	Kontrollstrukturen	61
2.4.1	Anweisungen	61
2.4.2	Sequenz (Reihung)	63
2.4.3	Auswahl (Selektion, Verzweigung)	63
2.4.4	Fallunterscheidungen mit switch	70
2.4.5	Schleifen	73
2.4.6	Kontrolle mit break und continue	82
2.5	Benutzerdefinierte und zusammengesetzte Datentypen	84
2.5.1	Aufzählungstypen	84
2.5.2	Arrays: Der C++-Standardtyp vector	86
2.5.3	Zeichenketten: Der C++-Standardtyp string	92
2.5.4	Strukturierte Datentypen	93
2.6	Bitfelder	96
3	Einfache Ein- und Ausgabe	97
3.1	Standardein- und -ausgabe	97
3.2	Ein- und Ausgabe mit Dateien	100

4	Programmstrukturierung	105
4.1	Funktionen	105
4.1.1	Aufbau und Prototypen	106
4.1.2	Gültigkeitsbereiche und Sichtbarkeit in Funktionen	109
4.2	Schnittstellen zum Datentransfer	111
4.2.1	Übergabe per Wert	111
4.2.2	Übergabe per Referenz	113
4.2.3	Gefahren bei der Rückgabe von Referenzen	114
4.2.4	Vorgegebene Parameterwerte und variable Parameterzahl	115
4.2.5	Überladen von Funktionen	116
4.2.6	Funktion main()	119
4.2.7	Spezifikation von Funktionen	119
4.2.8	Beispiel Taschenrechnersimulation	121
4.3	Grundsätze der modularen Gestaltung	127
4.3.1	Steuerung der Übersetzung mit #include	128
4.3.2	Einbinden vorübersetzter Programmteile	128
4.3.3	Dateiübergreifende Gültigkeit und Sichtbarkeit	131
4.3.4	Übersetzungseinheit, Deklaration, Definition	132
4.3.5	Compilerdirektiven und Makros	135
4.3.6	Einbinden von C-Funktionen	141
4.4	Funktionstemplates	142
4.4.1	Spezialisierung von Templates	144
4.4.2	Einbinden von Templates	145
4.5	inline-Funktionen	147
4.6	Standardfunktionen / Bibliotheken	149
5	Objektorientierung 1	151
5.1	Abstrakte Datentypen	151
5.2	Klassen und Objekte	153
5.2.1	inline-Elementfunktionen	157
5.3	Initialisierung und Konstruktoren	159
5.3.1	Standardkonstruktor	159
5.3.2	Allgemeine Konstruktoren	160
5.3.3	Kopierkonstruktor	163
5.3.4	Typumwandlungskonstruktor	166
5.4	Beispiel: Klasse für rationale Zahlen	168
5.4.1	Aufgabenstellung	168
5.4.2	Entwurf	169
5.4.3	Implementation	172
5.5	const-Objekte und Methoden	178
5.6	Faustregeln zur Konstruktion von Schnittstellen	179
5.7	Destruktoren	182
5.8	Wie kommt man zu Klassen und Objekten? Ein Beispiel	184

5.8.1	Einige Analyse-Überlegungen	185
5.8.2	Design	189
5.8.3	Eine Lösung	191
5.9	Gegenseitige Abhängigkeit von Klassen	195
6	Intermezzo: Zeiger	199
6.1	Zeiger und Adressen	199
6.2	C-Arrays	203
6.2.1	C-Arrays und sizeof	205
6.2.2	Indexoperator bei C-Arrays	206
6.2.3	Initialisierung von C-Arrays	206
6.2.4	Zeigerarithmetik	207
6.2.5	Mehrdimensionale Felder	208
6.3	C-Zeichenketten	210
6.4	Dynamische Datenobjekte	217
6.4.1	Freigeben dynamischer Objekte	220
6.4.2	Dynamisch erzeugte mehrdimensionale Arrays	223
6.5	Binäre Ein-/Ausgabe	225
6.6	Zeiger und Funktionen	230
6.6.1	Parameter des main-Programms	230
6.6.2	Parameterübergabe mit Zeigern	231
6.6.3	Gefahren bei der Rückgabe von Zeigern	233
6.7	Zeiger auf Funktionen	234
6.8	Komplexe Deklarationen	239
6.9	this-Zeiger	241
6.10	Zeiger auf Elementfunktionen und -daten	241
6.10.1	Zeiger auf Elementfunktionen	242
6.10.2	Zeiger auf Elementdaten	242
7	Objektorientierung 2	245
7.1	Eine String-Klasse	245
7.1.1	friend-Funktionen	252
7.2	Klassenspezifische Daten und Funktionen	253
7.2.1	Klassenspezifische Konstante	258
7.3	Klassentemplates	258
7.3.1	Ein Stack-Template	258
7.3.2	Stack mit statisch festgelegter Größe	261
7.3.3	Ökonomische Instantiierung von Templates	263
7.3.4	Member-Template	265
7.3.5	Rekursive Templates	266

8	Vererbung	269
8.1	Vererbung und Initialisierung	276
8.2	Zugriffsschutz	277
8.2.1	Read-Only Daten	280
8.3	Code-Wiederverwendung	281
8.4	Überschreiben von Funktionen in abgeleiteten Klassen	283
8.5	Polymorphismus	285
8.5.1	Virtuelle Funktionen	285
8.5.2	Abstrakte Klassen	291
8.5.3	Virtuelle Destruktoren	297
8.6	Mehrfachvererbung	299
8.6.1	Namenskonflikte	303
8.6.2	Virtuelle Basisklassen	304
8.6.3	Virtuelle Basisklassen und Initialisierung	306
8.7	Vererbung und andere Beziehungen	307
8.7.1	Vererbung	307
8.7.2	Der Teil und das Ganze	311
8.7.3	Assoziation	311
8.7.4	»Benutzt«-Beziehung	313
9	Überladen von Operatoren	315
9.1	Rationale Zahlen - noch einmal	317
9.1.1	Arithmetische Operatoren	317
9.1.2	Ausgabeoperator <<	320
9.2	Eine Klasse für Vektoren	322
9.2.1	Index-Operator []	325
9.2.2	Zuweisungsoperator =	328
9.2.3	Mathematische Vektoren	330
9.2.4	Multiplikations-Operator	332
9.3	Zuweisungsoperator und Vererbung	334
9.4	Inkrement-Operator ++	339
9.5	Typumwandlungsoperator	343
9.6	Smart-Pointer: Operatoren -> und *	345
9.6.1	Smart Pointer und die C++-Standardbibliothek	351
9.7	Objekte als Funktionen	352
10	Fehlerbehandlung	355
10.1	Ausnahmebehandlung	357
10.1.1	Exception-Spezifikation in Deklarationen	361
10.1.2	Exception-Hierarchie in C++	361
10.1.3	Besondere Fehlerbehandlungsfunktionen	363
10.1.4	Erkennen logischer Fehler	366
10.1.5	Durch Exceptions verursachte Speicherfehler vermeiden	368
10.2	Speicherbeschaffung mit new	369

11 Ein Werkzeugkasten	373
11.1 Behälterklassen (Container)	373
11.1.1 Container-Arten	376
11.2 Listen	377
11.3 Warteschlangen	384
11.4 Iteratoren	388
11.4.1 Iterator für eine Listen-Klasse	390
11.5 Sortierte Listen	396
11.6 Binärer Suchbaum	400
11.7 Trennung von Schnittstelle und Implementation	407
11.7.1 Entkopplung durch abstrakte Basisklassen	407
11.7.2 Handles	409
11.8 Mehrdimensionale Matrizen	413
11.8.1 Zweidimensionale Matrix	414
11.8.2 Dreidimensionale Matrix	416
11.9 Fehlersuche mit trace-Objekten	420
11.9.1 Klasse trace	422
11.9.2 Anwendungsbeispiel	426
12 Vermischtes	429
12.1 Namensräume	429
12.2 C++-Header	432
12.3 Standard-Typumwandlungsoperatoren	434
12.4 Typinformationen zur Laufzeit	437
12.4.1 Typidentifizierung mit typeid()	437
12.4.2 Anwendung: Eine Menge graphischer Objekte	438
12.4.3 Vor- und Nachteile der Laufzeittypinformation	444
13 Dateien und Ströme	447
13.1 Ausgabe	449
13.1.1 Formatierung der Ausgabe	450
13.2 Eingabe	455
13.3 Manipulatoren	457
13.3.1 Eigene Manipulatoren	461
13.4 Fehlerbehandlung	463
13.5 »Geheimnis« der Statusabfrage	465
13.6 Arbeit mit Dateien	466
13.6.1 Positionierung in Dateien	466
13.6.2 Lesen und Schreiben in derselben Datei	468
13.7 Eingabe benutzerdefinierter Typen	469
13.8 Umleitung auf Strings	471

Teil II Die C++ Standardbibliothek	475
14 Aufbau und Übersicht	477
14.1 Auslassungen	479
14.2 Beispiele des Buchs und die C++-Standardbibliothek	480
15 Hilfsfunktionen und -klassen	483
15.1 Paare	483
15.1.1 Relationale Operatoren	484
15.2 Funktionsobjekte	484
15.2.1 Arithmetische, vergleichende und logische Operationen	485
15.2.2 Funktionsobjekte zum Negieren logischer Prädikate	486
15.2.3 Binden von Argumentwerten	487
15.2.4 Zeiger auf Funktionen in Objekte umwandeln	488
16 Container	489
16.1 Notation für Iteratoren und Bereiche	489
16.2 Bitset	491
16.3 Deque	494
16.4 List	497
16.5 Map	500
16.5.1 Multimap	504
16.6 Queue	504
16.6.1 Priority-Queue	506
16.7 Set	507
16.7.1 Multiset	511
16.8 Stack	511
16.9 Vector	512
16.9.1 <code>vector<bool></code>	515
17 Iteratoren	519
17.1 Iteratorkategorien	520
17.2 <code>distance()</code> und <code>advance()</code>	521
17.3 Reverse-Iteratoren	522
17.4 Insert-Iteratoren	523
17.5 Stream-Iteratoren	525
18 Algorithmen	527
18.1 <code>for_each</code>	527
18.2 <code>find</code> und <code>find_if</code>	527
18.3 <code>find_end</code>	528
18.4 <code>find_first_of</code>	529
18.5 <code>adjacent_find</code>	529
18.6 <code>count</code>	530

18.7	mismatch	530
18.8	equal	531
18.9	search	532
18.10	search_n	533
18.11	copy und copy_backward	533
18.12	swap, iter_swap und swap_ranges	534
18.13	transform	535
18.14	replace und Varianten	536
18.15	fill und fill_n	537
18.16	generate und generate_n	537
18.17	remove und Varianten	538
18.18	unique	539
18.19	reverse	540
18.20	rotate	541
18.21	random_shuffle	541
18.22	partition	542
18.23	sort	543
18.23.1	partial_sort	545
18.24	nth_element	546
18.25	Binäre Suche	547
18.25.1	binary_search	547
18.25.2	lower_bound	548
18.25.3	upper_bound	548
18.25.4	equal_range	548
18.26	Verschmelzen (Mischen)	550
18.26.1	Verschmelzen an Ort und Stelle	552
18.27	Mengenoperationen auf sortierten Strukturen	553
18.27.1	includes	554
18.27.2	set_union	554
18.27.3	set_intersection	556
18.27.4	set_difference	556
18.27.5	set_symmetric_difference	557
18.28	Heap-Algorithmen	558
18.28.1	pop_heap	560
18.28.2	push_heap	561
18.28.3	make_heap	562
18.28.4	sort_heap	562
18.29	Minimum und Maximum	563
18.30	Lexikographischer Vergleich	564
18.31	Permutationen	564

19 Ein- und Ausgabe	567
19.1 Ergänzungen	568
19.1.1 Streams verbinden mit tie()	568
19.1.2 locale-Objekt ermitteln	568
19.1.3 sentry	568
20 Nationale Besonderheiten	571
20.1 Sprachumgebungen festlegen und ändern	571
20.1.1 locale-Elementfunktionen	572
20.1.2 Globale Funktionen	573
20.2 Zeichenklassifizierung und -umwandlung	573
20.3 Kategorien	574
20.3.1 collate	574
20.3.2 ctype	575
20.3.3 numeric	577
20.3.4 monetary	579
20.3.5 time	582
20.3.6 messages	585
20.4 Konstruktion eigener Facetten	585
21 Numerisches	587
21.1 Komplexe Zahlen	587
21.2 Grenzwerte von Zahltypen	588
21.3 Halbnumerische Algorithmen	590
21.3.1 accumulate	590
21.3.2 inner_product	591
21.3.3 partial_sum	592
21.3.4 adjacent_difference	593
21.4 Optimierte numerische Arrays (valarray)	594
21.4.1 Konstruktoren	595
21.4.2 Elementfunktionen	595
21.4.3 Binäre Valarray-Operatoren	598
21.4.4 Mathematische Funktionen	600
21.4.5 slice	602
21.4.6 slice_array	604
21.4.7 gslice	604
21.4.8 gslice_array	607
21.4.9 mask_array	608
21.4.10 indirect_array	609
22 String	611
23 Typerkennung zur Laufzeit	621

24 Speichermanagement	623
24.1 <new>	623
24.2 <memory>	624
25 C-Header	627
25.1 <cassert>	627
25.2 <cctype>	627
25.3 <cerrno>	628
25.4 <cfloat>	628
25.5 <ciso646>	628
25.6 <climits>	629
25.7 <locale>	629
25.8 <cmath>	629
25.9 <csetjmp>	629
25.10 <csignal>	629
25.11 <cstddef>	629
25.12 <cstdarg>	630
25.13 <cstdio>	631
25.14 <stdlib>	631
25.15 <cstring>	632
25.15.1 Funktionen für C-Strings	633
25.15.2 Funktionen für C-Strings maximaler Länge	634
25.15.3 Funktionen für Bytefelder	634
25.16 <ctime>	635
25.16.1 Datentypen	635
25.16.2 Funktionen	635
26 Nachwort	637
A Anhang	641
A.1 Programmierhinweise	641
A.2 Lösungen zu einigen Übungsaufgaben	644
A.3 Make-Dateien	671
A.3.1 Programmerzeugung aus vorgegebenen Dateien	672
A.3.2 Programmerzeugung für ein Projekt	673
A.4 C++-Schlüsselwörter	674
A.5 ASCII-Tabelle	674
Literaturverzeichnis	679
OOP-Glossar	681
Stichwortverzeichnis	689

