

Das C++ Kompendium

STL, Objektfabriken, Exceptions

Bearbeitet von
Gilbert Brands

1. Auflage 2010. Taschenbuch. xiv, 487 S. Paperback

ISBN 978 3 642 04786 2

Format (B x L): 15,5 x 23,5 cm

Gewicht: 1391 g

[Weitere Fachgebiete > EDV, Informatik > Software Engineering](#)

Zu [Leseprobe](#)

schnell und portofrei erhältlich bei

beck-shop.de
DIE FACHBUCHHANDLUNG

Die Online-Fachbuchhandlung beck-shop.de ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.

Inhaltsverzeichnis

Band I

Einführung in die Programmierung	1
1 (Statt eines) Vorwort(s)	1
2 Die ersten Schritte: Anweisungslisten	4
2.1 Die Grundregeln	4
2.2 Zur Arbeitstechnik	7
2.3 Aufbau der Anweisungsliste	7
2.4 Die Konstruktion der Anweisungen	9
3 Der Anfang	14
3.1 Arbeit mit dem Entwicklungssystem	16
3.2 Erste Erkenntnisse	17
3.3 Weitere Arbeitsschritte	18
4 Die Sprachelemente von C	21
4.1 Die Datentypen	21
4.2 Die Schnittstellendefinitionen	25
4.3 Bibliotheksfunktionen	27
4.4 Weitere Teile des Programmkode	29
4.5 Eigene Datentypen	33
5 Die Sprachelemente von C++	35
5.1 Überladen von Funktionen	35
5.2 Überladen von Operatoren	36
5.3 Namensbereiche	36
5.4 Klassen, Konstruktor, Destruktor	38
5.5 Vererbung	40
5.6 Zeigervariablen in C++	42
5.7 Virtuelle Vererbung	43
5.8 Mehrfachvererbung	45
5.9 Referenzen	46
5.10 Templates	46
6 Zur Arbeitsweise	48
1 Zur professionellen Arbeitsweise	51
1.1 Arbeitsphilosophie und Methodik	51
1.1.1 Die Auswahl der Programmiersprache	51

1.1.2	Anforderungen an eine Anwendung	52
1.1.3	Der Fehlerbegriff	53
1.1.4	Prüfen und Testen	56
1.1.5	Der Einfluss der Theorie	59
1.2	(Wieder-)Verwendbarkeit von Code	60
1.2.1	Bibliotheksmodule	61
1.2.2	Dokumentation von Code	67
1.3	Qualitätssicherung	77
1.4	Schnittstellenkonventionen	78
1.4.1	Erzeugung und Vernichtung von Zeigerbereichen	79
1.4.2	Typzuweisung (cast – Operationen)	84
1.4.3	Eigentumsrechte	88
1.4.4	Die Größe von Feldern	91
1.4.5	Pufferüberläufe	95
1.4.6	Importverwendung	96
1.4.7	Operatorenverwendung	102
2	Container und Algorithmen	107
2.1	Einleitung	107
2.2	Template-Klassen, inline-Funktionen	109
2.2.1	Template-Klassen und Template-Funktionen	109
2.2.2	Spezialisierungen	112
2.2.3	Offener Code	113
2.2.4	Partielle Übersetzung	114
2.2.5	Default-Parameter und template-template-Parameter	115
2.2.6	Rückgabe von Typen	116
2.2.7	Zahlen als Templateparameter	117
2.2.8	Effizienz und inline -Code	118
2.3	Zugriffe auf Daten: Verallgemeinerte Zeiger	119
2.3.1	Iteratoren	119
2.3.2	Einsatz von Iteratoren	121
2.3.3	Spezialisierungen für Container	122
2.3.4	Iteratorkategorien	123
2.3.5	Iteratoren und konstante Iteratoren	124
2.3.6	Iteratorabstand und Iteratorvorschub	126
2.3.7	Iteratorgültigkeit	127
2.3.8	Spezielle Attributtypen	128
2.3.9	Rückwärtsiteratoren	130
2.4	Verwaltung des Objektspeichers	131
2.4.1	Einführung	131
2.4.2	Allokator-Klassen	132
2.4.3	Eigene Allokatorklassen	134
2.5	Feld- oder Listenccontainer	135
2.5.1	Felder(STL-Klasse vector)	136
2.5.2	Segmentierte Felder (STL-Klasse deque)	140

2.5.3	Warteschlangen (STL-Klassen Stack und Queue)	142
2.5.4	Bitfelder	143
2.5.5	Zeichenketten Strings	144
2.5.6	Objekte und Zeiger in Containern	147
2.5.7	Verkette Listen (STL-Klasse list)	148
2.6	Bäume	152
2.6.1	Teilordnung und Vollordnung	152
2.6.2	Heap (STL-Klasse priority_queue)	154
2.6.3	Binärer (Rot-Schwarz)-Baum	157
2.6.4	STL-Klassen set und map/Hashsortierung	170
2.6.5	B+ – Bäume	175
2.7	Algorithmen und Container	186
2.7.1	Sortierrelationen	187
2.7.2	Suchen in unsortierten Containern	189
2.7.3	Suchen in sortierten Containern	189
2.7.4	Bubblesort-Sortieralgorithmus	190
2.7.5	Quicksort-Sortieralgorithmus	191
2.7.6	Heapsort-Sortieralgorithmus	194
2.8	Suchen in Strings	197
2.8.1	Einführende Bemerkungen	197
2.8.2	Naive Suche	198
2.8.3	Boyer-Moore-Algorithmus	199
2.8.4	Suffix-Bäume	203
2.9	Algorithmen der STL	209
2.9.1	Grunddesign der Algorithmen	210
2.9.2	Suchalgorithmen für einzelne Elemente	212
2.9.3	Suchen nach mehrfach auftretenden Elementen	213
2.9.4	Vollständige Übereinstimmung	214
2.9.5	Binärsuche	214
2.9.6	Anzahlen bestimmter Elemente	214
2.9.7	Unterschiede und Ähnlichkeiten	215
2.9.8	Enthaltensein von Elementen	215
2.9.9	Kopieren von Containern	216
2.9.10	Austauschen von Elementen	216
2.9.11	Löschen von Elementen	217
2.9.12	Reihenfolgeänderungen	217
2.9.13	Extremalwerte	219
2.9.14	Mischen von Containern	220
2.10	Relationen und eigene Algorithmen	222
2.10.1	Binäre und unäre Operatoren	222
2.10.2	Adapterklassen für komplexe Operationen	224
2.10.3	Aufwandsabschätzung	229
2.10.4	Ein Beispiel	231

3 Nützliche Werkzeuge	233
3.1 Namensbereiche und hilfreiche Templates	233
3.2 Umwandeln in Strings	236
3.3 Parameterstrings	241
3.3.1 Grundgerüst	241
3.3.2 Das Zerlegen und Rekonstruieren eines Strings	243
3.3.3 Arbeiten mit dem XMLString	246
3.4 Ablaufverfolgung (TRACE)	247
3.4.1 Debugger oder Tracer?	247
3.4.2 Eine einfache Trace-Klasse	248
3.4.3 Konditionelle Trace-Klassen	249
3.4.4 Trace-Gruppen	250
3.5 Objektstatistiken	253
3.6 Laufzeitmessungen	254
3.7 Datenkompression	257
3.7.1 Ein wenig Theorie	257
3.7.2 .. und eine Kompressionsklasse für die Praxis	261
3.8 Temporäre Dateien	265
3.9 Verschlüsselte Dateien	270
3.9.1 Die Aufgabenstellung	270
3.9.2 Der Algorithmus	271
3.9.3 Der Einsatz des Algorithmus	273
3.9.4 Die Implementation	275
3.9.5 Bemerkungen zur Verschlüsselung	280
3.10 Textdateien und Verzeichnisse	282
3.11 Laufwerksimulation	285
3.11.1 Die „File Allocation Table“ FAT	285
3.11.2 Verzeichnisse	286
3.11.3 Dateideskriptor	287
3.11.4 Simulation eines Laufwerks	288
3.11.5 Freie Sektoren und Zuordnung zu Dateien	290
3.11.6 Initialisierung eines Laufwerks	291
3.11.7 Laufwerk öffnen	294
3.11.8 Dateien öffnen	296
3.11.9 Verzeichnisse erzeugen	297
3.11.10 Verzeichnis wechseln	297
3.11.11 Löschen von Dateien	298
3.11.12 Löschen von Verzeichnissen	299
3.11.13 Abschlussbemerkungen	300
4 Lineare Algebra/mehrdimensionale Felder	301
4.1 Matrizen in C++	301
4.1.1 Normal besetzte Matrizen	301
4.1.2 Schwach besetzte Matrizen	310
4.1.3 Compilezeitoptimierungen – Vektoren und Matrizen	315

4.2	Numerisch–Mathematische Klassen	316
4.2.1	Das Rundungsproblem	316
4.2.2	Algebraische Eigenschaften	318
4.2.3	Konstantenvereinbarungen	320
4.2.4	Vergleiche und Nullprüfungen	321
4.2.5	Anwendung auf schwach besetzten Matrizen	324
4.3	Einige Algorithmen der linearen Algebra	324
4.3.1	Lineare Gleichungssysteme	325
4.3.2	Eigenwerte von Matrizen	330
5	Ausnahmen und Zeigerverwaltung	333
5.1	Zur Arbeitsweise mit Ausnahmen	334
5.2	Typermittlung und Zugriffsstandardisierung	340
5.2.1	Ableitung definierter Typen	340
5.2.2	Zugriffsnormierung	342
5.2.3	Ermittlung der Typart	343
5.3	Verwaltung von Zeigervariablen	346
5.3.1	Manuelle Ausnahmeverwaltung	346
5.3.2	Platzhalter- oder Trägervariable	347
5.3.3	Eine Instanz – mehrere Variable	349
5.3.4	Mehrfachreferenzen und automatische Verwaltung	354
5.3.5	Zeigerkopien	357
5.3.6	Mischen der Funktionalität, Zulässige Zuweisungen	359
5.3.7	Vollautomatische Policy-Auswahl	361
5.4	Steuerung der Ausnahmeverwaltung	363
5.4.1	Anforderungen an die Ausnahmesteuierung	363
5.4.2	Implementation 1: Realisierung der Objektleitung	366
5.4.3	Implementation II: Mischen von Strategien	371
5.5	Anwendungsbeispiel: Transaktionsmanagement	378
6	Objektfabriken	385
6.1	Laufzeitobjektfabrik	385
6.1.1	Motivation	385
6.1.2	Die Basisklasse für Fabrikobjekte	387
6.1.3	Klassenidentifikation und Persistenzmodell	390
6.1.4	Die eigentliche Objektfabrik	396
6.1.5	Benutzung neuer Methoden	399
6.1.6	Trennung von Anwendung und Bibliothek	405
6.1.7	Dynamische Einbindung einer DLL	408
6.2	Compilezeit-Objektfabriken	409
6.2.1	Typelisten	410
6.2.2	Zugriff auf einen Typ in der Liste	412
6.2.3	Algorithmen auf Typelisten	413
6.2.4	Arbeiten mit Typelisten	420
6.2.5	Beispiel: Completetime-Objektfabrik	427
6.3	Applets und Sandbox in C++	430
6.3.1	Das Sandbox-Konzept	430

6.3.2	Sandbox in C++ Umgebungen	431
6.3.3	Die Applet-Basisklasse	432
6.3.4	Der Security-Manager	434
6.3.5	Aufrufe und Probleme	435
7	Grafen	437
7.1	Grafen und ihre Speicherung	437
7.2	Arten des Eckenverbundes	441
7.2.1	Distanzlisten	442
7.2.2	Verbundenheit von Grafen	444
7.2.3	Abspalten disjunkter Subgrafen	444
7.2.4	Zyklenfreie (Sub)Grafen	445
7.3	Spannende Bäume	446
7.3.1	Breitensuche	447
7.3.2	Tiefensuche	448
7.3.3	Minimale (Maximale) Bäume	449
7.4	Wege in Grafen	450
7.4.1	Beliebige Wege und Zyklen	451
7.4.2	Wege mit kleiner Kantenanzahl	451
7.4.3	Minimale (Maximale) Wege	451
7.4.4	Rundwege in Grafen	454
7.4.5	Rundreise durch die Ecken	457
7.5	Netzwerke	458
7.5.1	Flüsse in Netzwerken	459
7.5.2	Flüsse mit Nebenbedingungen	460
7.5.3	Belegungsprobleme	461
8	Intervalle	463
8.1	Einführung	463
8.2	Funktion eines Intervallcontainers	464
8.3	Intervallimplementation	465
8.4	Relationen zwischen Intervallen	467
8.4.1	Überlappung/Durchschnitt	467
8.4.2	Vereinigung und Differenz	468
8.5	Intervallcontainer	470
8.5.1	Relationen zwischen Intervallen	470
8.5.2	Containerimplementation	471
9	Ausdrücke	475
9.1	Einführung	475
9.2	Zerlegung der Ausdrücke	476
9.2.1	Überführung von Methoden in Objekte	476
9.2.2	Typkonversion	477
9.2.3	Gerüste für binäre und unäre Ausdrücke	480
9.3	Datenobjekte in den Ausdrücken	481
9.3.1	Felder	482

9.3.2	Variablen	483
9.3.3	Konstante	484
9.3.4	Funktionsobjekte	486
9.4	Ein Beispiel	486

Band II

10	Speicherverwaltung (und ein wenig mehr)	489
10.1	Die Laufzeitproblematik	489
10.2	Das einfache Referenzkonzept	490
10.3	Referenzen mit temporärer Zwischenspeicherung	494
10.3.1	Die Strategie	495
10.3.2	Die Basisklasse	496
10.3.3	Die Ankerobjekte der Speicherverwaltung	498
10.4	Ein universeller Datenpuffer	500
10.5	Speicherklasse für kleine Objekte	504
10.5.1	Basis einer Speicherbank	505
10.5.2	Objekte fester Größe	507
10.5.3	Die allgemeine Allokatorklasse	508
10.5.4	Eine Basisklasse für die Allokatornutzung	510
11	Koordination von Abläufen	511
11.1	Grafische Anwenderschnittstellen	511
11.1.1	Bildschirmobjekte und Ereignisse	512
11.1.2	Ereignisketten	515
11.1.3	Änderung des Objektbaumes	521
11.1.4	Das Gesamtdesign	522
11.1.5	Grafische Anwendungsentwicklung	523
11.2	Funktoren – Aktoren	527
11.2.1	Verschieben von Funktionsaufrufen	528
11.2.2	Aufruf von (virtuellen) Klassenmethoden	532
11.3	Filterschlangen	534
11.3.1	Einfache Schlangen (Einführung)	534
11.3.2	Filterobjekt aus der Datenübertragung (Beispiel)	541
11.3.3	Verzweigungen	543
12	Bildverarbeitung	551
12.1	Vorbemerkungen	551
12.2	Analogbearbeitung von Bildern	553
12.2.1	Farbe, Kontrast, Helligkeit	553
12.2.2	Größenänderungen, Drehungen, Verzerrungen	554
12.2.3	Schärfung des Bildes	557
12.3	Strukturelle Bearbeitung/Digitalisierung	560
12.3.1	Digitalisierung von Konturen	560

12.3.2	Relationen zwischen Kantenlinien	565
12.3.3	Vorverarbeitung/Skelettierung von Bildern	569
12.4	Bildvergleiche	570
12.4.1	Pixelgestützte Ähnlichkeitsanalyse	570
12.4.2	Methoden der linearen Algebra	571
13	Computergrafik	575
13.1	Einleitung	575
13.2	Systemumgebung	576
13.2.1	Systeminitialisierung	576
13.2.2	System-Basisklasse und aktives Objekt	577
13.2.3	Objektinitialisierung und Projektionsmatrizen	580
13.2.4	Ereignisfunktionen	582
13.3	Daten und Datencontainer	586
13.3.1	Punkte und Punktcontainer	586
13.3.2	Punkte auf einem Gitter und Flächendarstellung	589
13.3.3	Indizierte Punktlisten	591
13.4	Objekte und Szenen	593
13.4.1	Basisklasse	593
13.4.2	Objektklasse	594
13.4.3	Szenen	595
13.4.4	Objektbibliotheken	597
13.5	Beleuchtungseffekte	599
13.5.1	Grundlagen	599
13.5.2	Lichtquellen	600
13.5.3	Objekteigenschaften	603
13.6	Perspektivische Projektion	606
13.6.1	Grundlagen der perspektivischen Darstellung	606
13.6.2	Projektionsdefinition	608
13.7	Flächendarstellungen	610
13.7.1	Texturen	610
13.7.2	Funktionen	612
13.7.3	Bezierflächen	613
13.7.4	NURBS-Freiformflächen	615
13.8	Listenverwaltung durch OpenGL	617
13.9	Offene Probleme	618
14	Datenstrukturen und ASN.1	619
14.1	Einführung in die Syntax	620
14.2	Binärkodierung	629
14.3	Übersetzen von Quellcode: Interpreter-Modus	636
14.3.1	Parsen der Kodebestandteile	636
14.3.2	Konstruktion der Felddatentypen	643
14.3.3	Bereichsdefinitionen	648
14.3.4	Elimination selbstdefinierter Typen	650
14.3.5	Auflösung der gegenseitigen Abhängigkeiten	652

14.4	Prüfung von Datensätzen	654
14.5	Datenbank und Anwendungsverknüpfung	657
14.5.1	Ein einfaches Datenbankmodell	657
14.5.2	Anwendung auf die ASN.1-Objekte	660
14.5.3	Verknüpfung mit anderen Datenobjekten	661
14.6	Verknüpfung mit den Filterklassen	663
14.7	Compilezeit – Implementation	671
15	Zahlendarstellungen	677
15.1	Ganze Zahlen	679
15.1.1	Basisalgorithmen	680
15.1.2	Anmerkungen zur Implementation	696
15.1.3	Verbesserung der Effizienz	697
15.2	Quotientenkörper	712
15.3	Restklassenkörper	715
15.3.1	Theoretische Grundlagen	715
15.3.2	Implementation der Restklasse	717
15.4	Fliesskommazahlen	719
15.4.1	Grundlagen	719
15.4.2	Klassenkonstruktion	721
15.4.3	Addition und Subtraktion	723
15.4.4	Division i	724
15.4.5	Division ii	724
15.4.6	Division iii	726
15.4.7	Relationen	726
15.4.8	Reelle Konstanten und Funktionen	728
15.4.9	Interpolation von Werten	731
15.5	Die Körper F_{2^m}, F_{p^m}	732
15.6	Metaprogramme und Körpererweiterungen	734
15.6.1	Theoretische Vorbemerkungen	734
15.6.2	Implementation der Algorithmen	736
16	Numerische Anwendungen	745
16.1	Rundungsfehler	745
16.2	Kontrolle von Fehlern	748
16.3	Arbeiten mit Polynomen	751
16.3.1	Eigenschaften von Operatoren	751
16.3.2	Nullstellen I: Berechnen	754
16.3.3	Nullstellen II: Finden	756
16.4	Intervallmathematik	758
16.4.1	Grundlagen	758
16.4.2	Vergleiche gerundeter Zahlen	759
16.4.3	Zwischenbilanz	763
16.4.4	Intervalltypen	764
16.4.5	Implementierung einer Intervallklasse	769
16.4.6	Einsatz der Intervallrechnung	776

17 Prä- und Postprocessing	781
17.1 Hintergrund	781
17.2 Präprocessing	781
17.2.1 Präprocessing mit spezieller Funktion	782
17.2.2 Präprocessing durch Objektmethode	782
17.2.3 Präprocessing mit Singleton-Objekt	783
17.2.4 Varianten, Kritik	784
17.3 Prä- und Postprocessing	785
17.3.1 Prä- und Postprocessing-Methodenverwaltung	786
17.3.2 Methodentypisierung	787
17.3.3 Die Funktorklasse(n)	788
17.3.4 Instanziierungsmethoden	790
18 Programm- und Prozesssteuerung	791
18.1 Allgemeines	791
18.2 Threads	793
18.2.1 Allgemeines	793
18.2.2 Erzeugen und Kontrollieren	794
18.2.3 Exklusive Programmteile	796
18.2.4 Synchronisation von Threads	799
18.3 Kommunikation zwischen Prozessen	803
18.3.1 Sockets	803
18.3.2 Verteilte Objekte	808
18.4 Parallelle und massiv parallele Prozesse	810
Stichwortverzeichnis	813