


Diese Leseprobe haben Sie beim  
 **edv-buchversand.de** heruntergeladen.  
Das Buch können Sie online in unserem  
Shop bestellen.

[Hier zum Shop](#)

## Auf einen Blick

1	Einstieg in die Welt von C++ .....	17
2	Erste Schritte in C++ .....	30
3	Die eingebauten C++-Basisdatentypen .....	41
4	Arbeiten mit den eingebauten Typen .....	72
5	Kontrollstrukturen .....	88
6	Arrays und Strings .....	116
7	Referenzen und Zeiger .....	137
8	Funktionen .....	150
9	Modularisierung und Präprozessor .....	184
10	Strukturen, Aufzählungen und dynamische Speicherobjekte .....	215
11	Klassen .....	237
12	Objekte und Klasselemente .....	283
13	Operatoren überladen .....	323
14	Vererbung (Abgeleitete Klassen) .....	354
15	Templates .....	376
16	Ausnahmebehandlung (Fehlerbehandlung) .....	399
17	Ein-/Ausgabestreams für Dateien .....	422
18	Weitere Sprachelemente und die Bibliotheken .....	435

## Inhalt

<b>1</b>	<b>Einstieg in die Welt von C++</b>	<b>17</b>
<hr/>		
1.1	Der C++-Standard .....	17
1.2	Die nötigen Werkzeuge für C++ .....	18
1.3	Übersetzen mit g++ und clang++ .....	22
1.4	Übersetzen mit einer Entwicklungsumgebung .....	24
1.5	Listings zum Buch .....	28
1.6	Kontrollfragen und Aufgaben im Buch .....	28
1.7	Aufgabe .....	28
<b>2</b>	<b>Erste Schritte in C++</b>	<b>30</b>
<hr/>		
2.1	Das erste Programm in C++ .....	30
2.2	Anweisungen und Ausdrücke .....	32
2.3	Die Standard-Eingabe- und -Ausgabestreams .....	33
2.3.1	Die Streams von C++ .....	34
2.3.2	Ausgabe mit »std::cout« .....	34
2.3.3	Eingabe mit »std::cin« .....	35
2.3.4	Ausgabe mit »std::cerr« .....	36
2.4	Einige Begriffe zu C++ .....	38
2.4.1	Bezeichner .....	38
2.4.2	Literale .....	39
2.4.3	Kommentare .....	39
2.5	Kontrollfragen und Aufgaben .....	40

<b>3 Die eingebauten C++-Basisdatentypen</b>	41
<b>3.1 Variablen</b>	41
<b>3.2 Definition und Deklaration von Variablen</b>	42
<b>3.3 Initialisierung und Zuweisung von Werten</b>	43
<b>3.4 Ganzzahltypen</b>	45
3.4.1 Literale von Ganzzahltypen	50
<b>3.5 Ganzzahldatentyp für Zeichen</b>	53
3.5.1 Weitere Datentypen für Zeichen	58
3.5.2 Die Unicode-Typen »char8_t«, »char16_t« und »char32_t«	60
<b>3.6 Fließkommazahlentypen</b>	62
3.6.1 Literale von Fließkommazahlen	64
<b>3.7 Der »auto«-Typ</b>	64
<b>3.8 Konstanten</b>	66
<b>3.9 Die Byte-Größe mit dem »sizeof«-Operator</b>	67
<b>3.10 Limits für die Basisdatentypen</b>	68
<b>3.11 Kontrollfragen und Aufgaben</b>	71
<b>4 Arbeiten mit den eingebauten Typen</b>	72
<b>4.1 Arithmetische Operatoren</b>	72
4.1.1 Kurzschreibweise arithmetischer Operatoren	76
4.1.2 Inkrement- und Dekrementoperator	77
<b>4.2 Ungenaue Fließkommazahlen</b>	79
<b>4.3 Typumwandlung</b>	81
4.3.1 Implizite Umwandlung durch den Compiler	81

4.3.2 Automatische Typumwandlung beschränken	84
4.3.3 Explizite Typumwandlung	85
<b>4.4 Kontrollfragen und Aufgaben</b>	87
<b>5 Kontrollstrukturen</b>	88
<b>5.1 Der eingebaute Datentyp »bool«</b>	88
<b>5.2 Vergleichsoperatoren</b>	89
<b>5.3 Bedingte Anweisung mit »if«</b>	91
<b>5.4 Anweisungsblock für Kontrollstrukturen</b>	93
<b>5.5 Alternative »else«-Verzweigung</b>	94
<b>5.6 Bedingte Anweisung mit Initialisierung</b>	95
<b>5.7 Mehrfache Verzweigung</b>	96
<b>5.8 Der Bedingungsoperator »?:«</b>	99
<b>5.9 Logische Operatoren</b>	100
<b>5.10 Die Fallunterscheidung – »switch«</b>	102
<b>5.11 Die kopfgesteuerte »while«-Schleife</b>	105
<b>5.12 Die fußgesteuerte »do while«-Schleife</b>	107
<b>5.13 Die Zählschleife »for«</b>	108
<b>5.14 Kontrollierte Sprunganweisungen</b>	112
5.14.1 Die »break«-Anweisung	112
5.14.2 Die »continue«-Anweisung	113
<b>5.15 Kontrollfragen und Aufgaben</b>	115

<b>6</b>	<b>Arrays und Strings</b>	116
<b>6.1</b>	<b>Arrays</b>	116
6.1.1	Der C++-Container »std::vector«	118
6.1.2	Der C++-Container »std::array«	125
6.1.3	C-Arrays	126
<b>6.2</b>	<b>Strings in C++</b>	130
6.2.1	Der C++-Container »std::string«	130
6.2.2	Unterstützung von Unicode	132
6.2.3	C-Zeichenketten	133
6.2.4	Zeichenkettenlitterale	135
<b>6.3</b>	<b>Kontrollfragen und Aufgaben</b>	136
<b>7</b>	<b>Referenzen und Zeiger</b>	137
<b>7.1</b>	<b>Referenzen</b>	137
<b>7.2</b>	<b>Zeiger</b>	139
7.2.1	Die Syntax von Zeigern	140
7.2.2	Zeiger dereferenzieren	142
7.2.3	Der Zeiger »nullptr«	144
7.2.4	Zeiger prüfen	145
7.2.5	Adresse einer Referenz	146
7.2.6	Verwendung von Zeigern und Alternativen	147
<b>7.3</b>	<b>Kontrollfragen und Aufgaben</b>	148
<b>8</b>	<b>Funktionen</b>	150
<b>8.1</b>	<b>Grundlage zu den Funktionen</b>	150
8.1.1	Funktionen definieren	151
8.1.2	Funktionen aufrufen	152

8.1.3	Funktionen deklarieren	153
8.1.4	Funktionsparameter (Call-by-Value)	155
8.1.5	Konstante Funktionsparameter	157
8.1.6	Standardparameter	158
8.1.7	Rückgabe aus Funktionen	160
8.1.8	Funktionen überladen	163
8.1.9	Gültigkeitsbereich und Sichtbarkeit von Variablen	166
8.1.10	Die »main()«-Funktion	169
8.1.11	Aufruf eines Programms mit Parametern	169
8.1.12	Programmende	171
<b>8.2</b>	<b>Referenzen als Parameter und Rückgabe</b>	171
8.2.1	Referenzen als Parameter	173
8.2.2	Konstante Funktionsparameter	174
8.2.3	Referenzen als Rückgabe	176
<b>8.3</b>	<b>Zeiger als Parameter und Rückgabewert</b>	178
8.3.1	Referenzen vs. Zeiger als Parameter	179
<b>8.4</b>	<b>Übergabe großer Elemente als Funktionsparameter</b>	179
<b>8.5</b>	<b>C-Arrays oder C-Strings als Funktionsparameter</b>	181
<b>8.6</b>	<b>Kontrollfragen und Aufgaben</b>	182
<b>9</b>	<b>Modularisierung und Präprozessor</b>	184
<b>9.1</b>	<b>Präprozessor-Direktiven</b>	184
9.1.1	Die »#include«-Direktive	185
9.1.2	Die »#define«-Direktive	186
9.1.3	Bedingte Kompilierung	187
<b>9.2</b>	<b>Modularisierung</b>	189
9.2.1	Aufteilung	189
9.2.2	Die öffentliche Schnittstelle (Headerdatei)	191

9.2.3	Die nicht öffentliche(n) Datei(en) .....	192
9.2.4	Die Client-Datei .....	193
9.2.5	Aufgabe .....	194
9.2.6	Nur Objektcode oder Bibliothek vorhanden .....	195
<b>9.3</b>	<b>Namensräume</b> .....	195
9.3.1	Namensraum deklarieren und verwenden .....	196
9.3.2	Namensraum verschachteln .....	199
9.3.3	Ein Namensraum ist ein eigener Gültigkeitsbereich .....	199
9.3.4	Namensraum mit »using« importieren .....	202
9.3.5	Einzelne Bezeichner mit »using« importieren .....	203
9.3.6	Alias für Namensräume .....	203
9.3.7	Anonymer Namensraum .....	204
9.3.8	Der Namensraum »std« .....	205
<b>9.4</b>	<b>Spezifizierer und Qualifikatoren</b> .....	206
9.4.1	Das Schlüsselwort »static« .....	207
9.4.2	Das Schlüsselwort »extern« .....	209
9.4.3	Das Schlüsselwort »constexpr« .....	210
9.4.4	Das Schlüsselwort »const« .....	211
9.4.5	Das Schlüsselwort »inline« .....	212
<b>9.5</b>	<b>Kontrollfragen und Aufgaben</b> .....	213
<b>10</b>	<b>Strukturen, Aufzählungen und dynamische Speicherobjekte</b> .....	215
<b>10.1</b>	<b>Erste eigene Datentypen mit Strukturen</b> .....	215
10.1.1	Strukturen definieren, Elemente erzeugen und initialisieren .....	216
10.1.2	Zugriff auf die Strukturelemente .....	218
10.1.3	Zugriff auf die Elemente in einer Funktion .....	219
10.1.4	Strukturen in einem Vektor oder Array .....	220

10.1.5	Methoden statt Funktionen .....	221
10.1.6	Strukturen vergleichen .....	223
<b>10.2</b>	<b>Aufzählungstyp »enum«</b> .....	223
<b>10.3</b>	<b>Eigene Namen mit »using«</b> .....	225
<b>10.4</b>	<b>Dynamische Speicherobjekte</b> .....	226
10.4.1	Dynamisch Objekte mit »new« und »delete« anlegen und freigeben .....	228
10.4.2	Dynamisch Arrays mit »new[ ]« anlegen und mit »delete[ ]« freigeben .....	231
10.4.3	Der smarte »unique_ptr«-Pointer .....	233
<b>10.5</b>	<b>Kontrollfragen und Aufgaben</b> .....	236
<b>11</b>	<b>Klassen</b> .....	237
<b>11.1</b>	<b>Klassen</b> .....	237
11.1.1	Klassendefinition .....	237
11.1.2	Zugriffskontrolle mit »public« und »private« .....	239
11.1.3	Methoden definieren .....	243
11.1.4	Objekte erzeugen und benutzen .....	245
<b>11.2</b>	<b>Konstruktoren</b> .....	251
11.2.1	Konstruktoren deklarieren .....	252
11.2.2	Konstruktoren definieren .....	253
11.2.3	Konstruktoren delegieren .....	256
11.2.4	Der Standardkonstruktor (Default-Konstruktor) .....	258
11.2.5	Implizite Konvertierungen und ihre Verhinderung – »explicit« .....	260
11.2.6	Der Kopierkonstruktor (Copy-Konstruktor) .....	262
11.2.7	Der Verschiebekonstruktor (Move-Konstruktor) .....	264
<b>11.3</b>	<b>Destruktoren</b> .....	268
11.3.1	Die Lebensdauer eines Objekts .....	268

11.3.2	Wann ist ein Destruktor erforderlich? .....	268
11.3.3	Destruktor deklarieren .....	269
11.3.4	Destruktor definieren .....	270
<b>11.4</b>	<b>Weitere Formen von Methoden .....</b>	<b>273</b>
11.4.1	»inline«-Methoden .....	273
11.4.2	Konstante Methoden (»nur-lesend«) .....	276
11.4.3	Konstante Methoden explizit ausschließen .....	278
11.4.4	»this«-Zeiger .....	279
<b>11.5</b>	<b>Kontrollfragen und Aufgaben .....</b>	<b>281</b>
<b>12</b>	<b>Objekte und Klasselemente .....</b>	<b>283</b>
<b>12.1</b>	<b>Objekt als Parameter .....</b>	<b>283</b>
12.1.1	Objekte an eine Funktion übergeben .....	283
12.1.2	Objekte an eine Methode übergeben .....	286
<b>12.2</b>	<b>Freundfunktionen (»friend«) .....</b>	<b>288</b>
<b>12.3</b>	<b>Objekte einer Klasse als Rückgabewerte .....</b>	<b>290</b>
12.3.1	Referenzen auf eine Klasse als Rückgabewerte .....	293
<b>12.4</b>	<b>Arrays von Objekten .....</b>	<b>296</b>
<b>12.5</b>	<b>Dynamische Objekte .....</b>	<b>297</b>
<b>12.6</b>	<b>Klassenobjekte als Klassenattribute .....</b>	<b>299</b>
<b>12.7</b>	<b>Containerklasse als Klassenattribut .....</b>	<b>304</b>
<b>12.8</b>	<b>Smart Pointer als Klassenattribut .....</b>	<b>307</b>
<b>12.9</b>	<b>Statische und konstante Klasselemente .....</b>	<b>311</b>
12.9.1	Statische Klasselemente .....	311
12.9.2	Konstante Klasselemente .....	316
12.9.3	Rohe Zeiger als Klasselemente oder direkt die Nullregel .....	317

<b>12.10</b>	<b>Die Nullregel (Rule of Zero) .....</b>	<b>318</b>
12.10.1	Die großen fünf .....	319
<b>12.11</b>	<b>Kontrollfragen und Aufgaben .....</b>	<b>322</b>
<b>13</b>	<b>Operatoren überladen .....</b>	<b>323</b>
<b>13.1</b>	<b>Das Schlüsselwort »operator« .....</b>	<b>325</b>
<b>13.2</b>	<b>Zweistellige (arithmetische) Operatoren überladen .....</b>	<b>326</b>
13.2.1	Operatorüberladung als Methode einer Klasse .....	328
13.2.2	Operatorüberladung als globale Hilfsfunktion .....	331
<b>13.3</b>	<b>Einstellige Operatoren überladen .....</b>	<b>333</b>
<b>13.4</b>	<b>Den Zuweisungsoperator überladen .....</b>	<b>337</b>
<b>13.5</b>	<b>Ausgabe- und Eingabeoperatoren überladen .....</b>	<b>342</b>
13.5.1	Den Ausgabeoperator »<<« überladen .....	342
13.5.2	Den Eingabeoperator »>>« überladen .....	343
<b>13.6</b>	<b>Vergleichsoperatoren .....</b>	<b>345</b>
13.6.1	Der Drei-Wege-Vergleichsoperator .....	346
<b>13.7</b>	<b>Weitere Operatorüberladungen .....</b>	<b>348</b>
<b>13.8</b>	<b>Konvertierungsoperatoren .....</b>	<b>348</b>
13.8.1	Der Konvertierungskonstruktor .....	349
13.8.2	Die Konvertierungsfunktion .....	350
<b>13.9</b>	<b>Kontrollfragen und Aufgaben .....</b>	<b>352</b>
<b>14</b>	<b>Vererbung (Abgeleitete Klassen) .....</b>	<b>354</b>
<b>14.1</b>	<b>Die Vorbereitung .....</b>	<b>355</b>
<b>14.2</b>	<b>Das Ableiten einer Klasse .....</b>	<b>357</b>
14.2.1	Erben und erweitern .....	358

14.2.2	»public«-Zugriffsrechte einer abgeleiteten Klasse .....	359
14.2.3	Methoden überschreiben .....	361
14.2.4	Konstruktoren .....	362
14.2.5	Programmbeispiel .....	364
14.2.6	Konstruktoren vererben .....	365
14.2.7	Destruktor .....	367
14.2.8	Die Zugriffsspezifikation »protected« .....	367
14.2.9	Implizite Typumwandlung abgeleiteter Klassen .....	369
14.2.10	Überschreiben mit virtuellen Methoden .....	370
14.2.11	Abstrakte Klassen und rein virtuelle Methoden .....	373
<b>14.3</b>	<b>Kontrollfragen und Aufgaben .....</b>	<b>375</b>
<b>15</b>	<b>Templates .....</b>	<b>376</b>
<b>15.1</b>	<b>Funktions-Templates .....</b>	<b>376</b>
15.1.1	Funktions-Template definieren .....	377
15.1.2	Funktions-Templates über mehrere Module .....	380
15.1.3	Ein Funktions-Template spezialisieren .....	380
15.1.4	Templates mit verschiedenen Parametern .....	382
15.1.5	Explizite Template-Argumente .....	383
15.1.6	Wiederverwendbare Templates .....	384
<b>15.2</b>	<b>Klassen-Templates .....</b>	<b>385</b>
15.2.1	Klassen-Template definieren .....	385
15.2.2	Template-Methoden definieren .....	386
15.2.3	Template-Methoden spezialisieren .....	387
15.2.4	Klassen-Template instanziiieren .....	388
15.2.5	Klassen-Template mit mehreren formalen Parametern .....	390

<b>15.3</b>	<b>Templates der Standardbibliothek .....</b>	<b>390</b>
15.3.1	Iteratoren .....	391
15.3.2	Algorithmen .....	394
<b>15.4</b>	<b>Kontrollfragen und Aufgaben .....</b>	<b>397</b>
<b>16</b>	<b>Ausnahmebehandlung (Fehlerbehandlung) .....</b>	<b>399</b>
<b>16.1</b>	<b>Eine Ausnahme auslösen .....</b>	<b>401</b>
<b>16.2</b>	<b>Ausnahme auffangen und behandeln .....</b>	<b>402</b>
16.2.1	Alternatives »catch (...)« .....	405
16.2.2	Ausnahmen aus der Standardbibliothek .....	407
16.2.3	Ausnahme mit »throw« weiterwerfen .....	409
16.2.4	»noexcept« .....	411
16.2.5	Stack-Abwicklung .....	412
<b>16.3</b>	<b>Ausnahmeklassen (Fehlerklassen) .....</b>	<b>412</b>
<b>16.4</b>	<b>Standardausnahmen .....</b>	<b>415</b>
16.4.1	Die virtuelle Methode »what()« .....	416
16.4.2	Die Standardausnahmen verwenden .....	416
<b>16.5</b>	<b>Systemausnahmen .....</b>	<b>420</b>
<b>16.6</b>	<b>Kontrollfragen .....</b>	<b>421</b>
<b>17</b>	<b>Ein-/Ausgabestreams für Dateien .....</b>	<b>422</b>
<b>17.1</b>	<b>Der Umgang mit Dateien in C++ .....</b>	<b>422</b>
<b>17.2</b>	<b>Verschiedene Streams für Dateien .....</b>	<b>422</b>
<b>17.3</b>	<b>Datei öffnen und schließen .....</b>	<b>423</b>
17.3.1	Verschiedene Modi zum Öffnen von Dateien .....	425
17.3.2	Byteweise lesen und schreiben .....	428

17.3.3	Zeilenweise lesen und schreiben .....	430
17.3.4	Blockweise lesen und schreiben .....	431
17.3.5	Die Lese- oder Schreibposition ändern .....	433
<b>17.4</b>	<b>Kontrollfragen und Aufgaben</b> .....	<b>434</b>
<b>18</b>	<b>Weitere Sprachelemente und die Bibliotheken</b> .....	<b>435</b>
<hr/>		
<b>18.1</b>	<b>Weitere Sprachelemente</b> .....	<b>435</b>
18.1.1	»decltype« .....	435
18.1.2	Rückgabesyntax mit nachlaufendem Rückgabetyt .....	436
18.1.3	Automatische Rückgabetypermittlung .....	438
18.1.4	Lambda-Funktionen .....	439
<b>18.2</b>	<b>Smart Pointer</b> .....	<b>442</b>
18.2.1	Die Move-Semantik .....	450
<b>18.3</b>	<b>Elemente der Standardbibliothek</b> .....	<b>452</b>
18.3.1	Sequenzielle Container .....	453
18.3.2	Assoziative Container .....	455
<b>18.4</b>	<b>Die Zeitbibliothek</b> .....	<b>458</b>
18.4.1	Zeitdauer (»duration«) .....	459
18.4.2	Vorhandene Zeitgeber .....	460
<b>18.5</b>	<b>Die »ratio«-Bibliothek</b> .....	<b>463</b>
<b>18.6</b>	<b>Multithreading</b> .....	<b>464</b>
	Lösungen der Übungsaufgaben .....	466
	Index .....	489