



Arnold Willemer

# C++ Der Einstieg

WILEY

# Inhaltsverzeichnis

	Vorwort . . . . .	13
1	Einstieg in die Programmierung . . . . .	15
1.1	Programmieren . . . . .	15
1.1.1	Start eines Programms . . . . .	16
1.1.2	Eintippen, übersetzen, ausführen . . . . .	17
1.1.3	Der Algorithmus . . . . .	19
1.1.4	Die Sprache C++ . . . . .	20
1.2	Compiler beschaffen und einrichten . . . . .	24
1.2.1	Installation . . . . .	25
1.2.2	IDE-Aufbau . . . . .	26
1.2.3	Ein Projekt anlegen . . . . .	26
1.2.4	Übersetzen und starten . . . . .	27
1.3	Grundgerüst eines Programms . . . . .	28
1.3.1	Kommentare . . . . .	28
1.3.2	Anweisungen . . . . .	31
1.3.3	Blöcke . . . . .	31
1.4	Variablen . . . . .	32
1.4.1	Variablendefinition . . . . .	33
1.4.2	Geltungsbereich . . . . .	35
1.4.3	Namensregeln und Syntaxgraph . . . . .	37
1.4.4	Typen . . . . .	39
1.4.5	Syntax der Variablendefinition . . . . .	50
1.4.6	Konstanten . . . . .	51
1.5	Verarbeitung . . . . .	61
1.5.1	Zuweisung . . . . .	62
1.5.2	Rechenkünstler . . . . .	63
1.5.3	Abkürzungen . . . . .	64
1.5.4	Funktionen am Beispiel der Zufallsfunktion . . . . .	67
1.5.5	Typumwandlung . . . . .	69
1.6	Ein- und Ausgabe . . . . .	70
1.6.1	Ausgabestrom nach cout . . . . .	70

1.6.2	Formatierte Ausgabe . . . . .	72
1.6.3	Eingabestrom aus cin . . . . .	73
1.7	Übungen . . . . .	74
2	<b>Ablaufsteuerung</b> . . . . .	75
2.1	Verzweigungen . . . . .	76
2.1.1	Nur unter einer Bedingung: if . . . . .	77
2.1.2	Andernfalls: else . . . . .	78
2.1.3	Fall für Fall: switch case . . . . .	82
2.1.4	Kurzabfrage mit dem Fragezeichen . . . . .	86
2.2	Boolesche Ausdrücke . . . . .	88
2.2.1	Variablen und Konstanten . . . . .	88
2.2.2	Operatoren . . . . .	89
2.2.3	UND, ODER, NICHT – Verknüpfung von booleschen Ausdrücken . . . . .	91
2.2.4	De Morgan . . . . .	95
2.3	Immer diese Wiederholungen: Schleifen . . . . .	97
2.3.1	Kopfgesteuert: while . . . . .	98
2.3.2	Fußgesteuert: do...while . . . . .	101
2.3.3	Abgezählt: for . . . . .	103
2.3.4	Schleifensprünge: break und continue . . . . .	106
2.3.5	Der brutale Sprung: goto . . . . .	108
2.4	Beispiele . . . . .	109
2.4.1	Primzahlen . . . . .	110
2.4.2	Größter gemeinsamer Teiler . . . . .	114
2.5	Übungen . . . . .	117
3	<b>Datentypen und -strukturen</b> . . . . .	119
3.1	Das Array . . . . .	119
3.1.1	Beispiel: Bubblesort . . . . .	125
3.1.2	C-Zeichenketten . . . . .	129
3.1.3	Beispiel: Zahleneingabe auswerten . . . . .	130
3.1.4	Mehrere Dimensionen . . . . .	133
3.1.5	Beispiel: Bermuda . . . . .	134
3.2	Der Zeiger und die Adresse . . . . .	137
3.2.1	Indirekter Zugriff . . . . .	141
3.2.2	Arrays und Zeiger . . . . .	143
3.2.3	Zeigerarithmetik . . . . .	145
3.2.4	Konstante Zeiger . . . . .	147
3.2.5	Anonyme Zeiger . . . . .	148

3.3	Der Variablenverbund: struct . . . . .	148
3.3.1	Beispiel: Bermuda . . . . .	152
3.4	Dynamische Strukturen . . . . .	154
3.4.1	Anlegen und Freigeben von Speicher . . . . .	154
3.4.2	Zur Laufzeit erzeugte Arrays . . . . .	156
3.4.3	Verkettete Listen . . . . .	156
3.5	Die Union . . . . .	159
3.6	Aufzählungstyp enum . . . . .	160
3.7	Typen definieren . . . . .	162
3.8	Übungen . . . . .	163
<b>4</b>	<b>Funktionen . . . . .</b>	<b>165</b>
4.1	Anweisungen zusammenfassen . . . . .	166
4.2	Rückkehr und Rückgabewert . . . . .	169
4.3	Parameter . . . . .	171
4.3.1	Prototypen . . . . .	174
4.3.2	Zeiger als Parameter . . . . .	175
4.3.3	Arrays als Parameter . . . . .	178
4.3.4	Referenzparameter . . . . .	180
4.3.5	Beispiel: Stack . . . . .	181
4.3.6	Vorbelegte Parameter . . . . .	183
4.3.7	Die Parameter der Funktion main . . . . .	184
4.3.8	Variable Anzahl von Parametern . . . . .	186
4.4	Überladen von Funktionen . . . . .	188
4.5	Kurz und schnell: Inline-Funktionen . . . . .	189
4.6	Top-Down . . . . .	190
4.6.1	Beispiel: Bermuda . . . . .	191
4.7	Geltungsbereich von Variablen . . . . .	196
4.7.1	Globale Variablen . . . . .	196
4.7.2	Lokale Variablen . . . . .	197
4.7.3	Statische Variablen . . . . .	198
4.8	Selbstaufrufende Funktionen (Rekursion) . . . . .	200
4.8.1	Einsatzbereich . . . . .	202
4.8.2	Beispiel: binärer Baum . . . . .	202
4.8.3	Türme von Hanoi . . . . .	206
4.8.4	Beispiel: Taschenrechner . . . . .	208
4.9	Funktionszeiger und Lambda-Funktionen . . . . .	214
4.10	Übungen . . . . .	217
<b>5</b>	<b>Klassen . . . . .</b>	<b>219</b>
5.1	Die Klasse als Datenstruktur . . . . .	220

	5.1.1	Funktion und Datenstruktur heiraten . . . . .	222
	5.1.2	Zugriff auf Klassenelemente . . . . .	224
5.2		Geburt und Tod eines Objekts . . . . .	224
	5.2.1	Der Konstruktor initialisiert . . . . .	225
	5.2.2	Konstruktor und Parameter . . . . .	227
	5.2.3	Destruktor . . . . .	231
	5.2.4	Konstruktor und Destruktor bei Arrays . . . . .	232
5.3		Öffentlichkeit und Privatsphäre . . . . .	233
	5.3.1	private und public . . . . .	233
	5.3.2	Beispiel: Stack . . . . .	237
	5.3.3	Freunde . . . . .	239
5.4		Überladen von Elementfunktionen . . . . .	241
5.5		Kür: Operatoren überladen . . . . .	242
	5.5.1	Addition . . . . .	244
	5.5.2	Globale Operatorfunktionen . . . . .	246
	5.5.3	Inkrementieren und Dekrementieren . . . . .	247
	5.5.4	Die Vergleichsoperatoren . . . . .	248
	5.5.5	Der Ausgabeoperator . . . . .	250
	5.5.6	Der Indexoperator . . . . .	251
	5.5.7	Der Aufrufoperator () . . . . .	253
	5.5.8	Der Konvertierungsoperator . . . . .	254
5.6		Kopieren und Verschieben . . . . .	255
	5.6.1	Der Zuweisungsoperator . . . . .	255
	5.6.2	Kopierkonstruktor . . . . .	258
	5.6.3	Verschiebekonstruktor und Move-Semantik . . . . .	262
5.7		static und const . . . . .	264
	5.7.1	Statische Variablen und Funktionen in Klassen . . . . .	264
	5.7.2	Konstante Parameter und Funktionen . . . . .	265
5.8		Vererbung . . . . .	268
	5.8.1	Zugriff nur für Erben: protected . . . . .	273
	5.8.2	Zugriffsattribute der Vererbung . . . . .	275
	5.8.3	Konstrukorenvererbung . . . . .	277
	5.8.4	Kopierkonstruktor und Zuweisungsoperator . . . . .	278
	5.8.5	Mehrfachvererbung . . . . .	279
	5.8.6	Polymorphie durch virtuelle Funktionen . . . . .	280
	5.8.7	Abstrakte Basisklasse . . . . .	285
	5.8.8	Polymorphie bei grafischen Oberflächen . . . . .	287
	5.8.9	Exkurs: Grafische Applikationen . . . . .	288
5.9		Klassendefinition und Syntaxgraph . . . . .	292
5.10		Übungen . . . . .	293

<b>6</b>	<b>Programmierwerkzeuge</b>	<b>295</b>
6.1	Der C++-Compiler	295
6.1.1	Compiler-Aufruf	296
6.1.2	Compiler-Optionen	297
6.1.3	Fehlermeldungen	297
6.2	Präprozessor	300
6.2.1	Einbinden von Dateien: <code>#include</code>	300
6.2.2	Konstanten und Makros: <code>#define</code>	300
6.2.3	Makro-Programmierung mit <code>#define</code>	301
6.2.4	Abfragen: <code>#if</code>	303
6.2.5	Vordefinierte Makros	304
6.3	Aufteilung der Quelltexte	306
6.3.1	Beispiel: Bermuda	306
6.3.2	Dateikennungen	309
6.3.3	Deklaration und Definition	310
6.3.4	Header-Dateien	311
6.3.5	Statische Funktionen	312
6.3.6	Verborgene Implementierung	313
6.4	Linker und Bibliotheken	315
6.4.1	Statische Bibliotheken einbinden	315
6.4.2	Dynamische Bibliotheken	316
6.5	Auf Fehlersuche mit dem Debugger	320
6.6	<code>make</code>	321
6.6.1	Makros im Makefile	323
6.6.2	Mehrere Ziele	324
<b>7</b>	<b>Weitere Sprachelemente von C++</b>	<b>325</b>
7.1	Generische Programmierung	325
7.1.1	Template-Funktionen	326
7.1.2	Template-Klassen	329
7.2	Namensräume	333
7.2.1	Definition eines Namensraums	333
7.2.2	Zugriff	334
7.2.3	Besondere Namensräume	334
7.2.4	Anonyme Namensräume	335
7.2.5	Syntaxgraph	336
7.3	Katastrophenschutz mit <code>try</code> und <code>catch</code>	336
7.3.1	Eigene Ausnahmen erzeugen	338
7.3.2	Erstellen von Fehlerklassen	341
7.3.3	Die Ausnahmen der Standardbibliotheken	345

7.4	Systemnahe Programmierung . . . . .	351
7.4.1	Bit-Operatoren . . . . .	351
7.4.2	Shift-Operatoren . . . . .	354
7.4.3	Zugriff auf Hardware-Adressen . . . . .	355
7.4.4	Bit-Strukturen . . . . .	356
7.5	Übungen . . . . .	356
<b>8</b>	<b>Zeichenketten und Dateien . . . . .</b>	<b>357</b>
8.1	Zeichenketten und Strings . . . . .	358
8.1.1	Die Standardklasse string . . . . .	358
8.1.2	Andere String-Bibliotheken . . . . .	369
8.1.3	Klassische C-Zeichenkettenfunktionen . . . . .	369
8.2	iostream für Fortgeschrittene . . . . .	376
8.2.1	Eingabe über cin . . . . .	376
8.2.2	Manipulatoren . . . . .	378
8.3	Dateioperationen . . . . .	382
8.3.1	Öffnen und Schließen . . . . .	383
8.3.2	Lesen und Schreiben . . . . .	385
8.3.3	Zustandsbeobachtung . . . . .	390
8.3.4	Dateizugriffe nach ANSI-C . . . . .	393
8.3.5	Dateisystemkommandos . . . . .	395
8.3.6	Dateieigenschaften ermitteln . . . . .	398
8.4	Übungen . . . . .	401
<b>9</b>	<b>Bibliotheken für Mathematik und die Zeit . . . . .</b>	<b>403</b>
9.1	Mathematische Funktionen . . . . .	403
9.1.1	Die mathematische Standardbibliothek . . . . .	403
9.1.2	Komplexe Zahlen . . . . .	406
9.2	Zeitfunktionen . . . . .	407
9.2.1	Datum und Uhrzeit . . . . .	407
9.2.2	Zeit stoppen . . . . .	409
<b>10</b>	<b>Die Standard Template Library (STL) . . . . .</b>	<b>411</b>
10.1	Die Container-Klasse vector . . . . .	412
10.1.1	Dimensionsänderung . . . . .	413
10.1.2	Iteratoren . . . . .	416
10.1.3	Weitere Funktionen . . . . .	418
10.2	Die Container-Klasse deque . . . . .	422
10.3	Die Container-Klasse list . . . . .	425
10.3.1	Einfügen und Löschen . . . . .	426
10.3.2	Umhängen von Elementen: splice . . . . .	428

	10.3.3	Sortierung, Zusammenführen und Umdrehen . . . . .	430
10.4		Die Container-Klassen set und multiset . . . . .	434
	10.4.1	Einfügen und Löschen . . . . .	434
	10.4.2	Suchen und Sortieren . . . . .	436
10.5		Die Container-Klassen map und multimap . . . . .	438
10.6		Container-Adapter . . . . .	439
	10.6.1	Der Container-Adapter stack . . . . .	440
	10.6.2	Der Container-Adapter queue . . . . .	442
	10.6.3	Der Container-Adapter priority_queue . . . . .	443
10.7		Iteratortypen . . . . .	444
10.8		Die Algorithmen der STL . . . . .	445
	10.8.1	Suchen: find() . . . . .	445
	10.8.2	Sortieren . . . . .	447
	10.8.3	Binäres Suchen . . . . .	448
	10.8.4	Kopieren: copy() . . . . .	448
	10.8.5	Umdrehen: reverse() . . . . .	449
	10.8.6	Füllen: fill() . . . . .	449
	10.8.7	equal() . . . . .	450
	10.8.8	Funktion als Parameter: find_if() . . . . .	450
	10.8.9	for_each . . . . .	453
10.9		Die Template-Klasse bitset . . . . .	454
10.10		Übungen . . . . .	455

## Anhänge

A		<b>C++ für Hektiker . . . . .</b>	<b>457</b>
A.1		Ein Programm . . . . .	457
A.2		Abfrage und Schleifen . . . . .	462
	A.2.1	Abfrage und boolesche Ausdrücke . . . . .	462
	A.2.2	Die while-Schleife . . . . .	464
	A.2.3	Die for-Schleife . . . . .	465
A.3		Arrays . . . . .	466
A.4		Funktionen . . . . .	470
	A.4.1	Programmaufteilung . . . . .	470
	A.4.2	Rückgabewert . . . . .	473
	A.4.3	Parameter . . . . .	475
A.5		Klassen . . . . .	478
	A.5.1	Konstruktor . . . . .	482
	A.5.2	Vererbung . . . . .	486
	A.5.3	Polymorphie . . . . .	489
A.6		Templates . . . . .	490



B	Musterlösungen . . . . .	493
	Glossar . . . . .	521
	Stichwortverzeichnis . . . . .	525