

Auf einen Blick

	Vorwort	17
	Vorwort des Fachgutachters	23
1	Grundlagen in C++	25
2	Höhere und fortgeschrittene Datentypen	129
3	Gültigkeitsbereiche, spezielle Deklarationen und Typumwandlungen	219
4	Objektorientierte Programmierung	259
5	Templates und STL	473
6	Exception-Handling	657
7	C++ Standardbibliothek	691
8	Weiteres zum C++-Guru	817
9	Netzwerkprogrammierung und Cross-Plattform-Entwicklung in C++	899
10	GUI- und Multimediaprogrammierung in C++	973
11	Anhang	1189
	Index	1207

Inhalt

Vorwort	17
Vorwort des Fachgutachters	23
1 Grundlagen in C++	25
1.1 Die Entstehung von C++	25
1.1.1 Aufbau von C++	27
1.2 Erste Schritte der C++-Programmierung	30
1.2.1 Ein Programm erzeugen mit einem Kommandozeilen-Compiler	31
1.2.2 Ausführen des Programms	33
1.2.3 Ein Programm erzeugen mit einer IDE	33
1.3 Symbole von C++	34
1.3.1 Bezeichner	34
1.3.2 Schlüsselwörter	34
1.3.3 Literale	34
1.3.4 Einfache Begrenzer	36
1.4 Basisdatentypen	37
1.4.1 Deklaration und Definition	38
1.4.2 Was ist eine Variable?	38
1.4.3 Der Datentyp bool	39
1.4.4 Der Datentyp char	39
1.4.5 Die Datentypen int	42
1.4.6 Gleitkommazahlen float, double und long double	45
1.4.7 Limits für Ganzzahl- und Gleitpunkttypen	48
1.5 Konstanten	49
1.6 Standard Ein-/Ausgabe-Streams	50
1.6.1 Die neuen Streams – cout, cin, cerr, clog	51
1.6.2 Ausgabe mit cout	52
1.6.3 Ausgabe mit cerr	52
1.6.4 Eingabe mit cin	53
1.7 Operatoren	55
1.7.1 Arithmetische Operatoren	56
1.7.2 Inkrement- und Dekrementoperator	59
1.7.3 Bitoperatoren	60
1.7.4 Weitere Operatoren	64
1.8 Kommentare	64

1.9	Kontrollstrukturen	65
1.9.1	Verzweigungen (Selektionen)	65
1.9.2	Schleifen (Iterationen)	84
1.9.3	Sprunganweisungen	91
1.10	Funktionen	94
1.10.1	Deklaration und Definition	95
1.10.2	Funktionsaufruf und Parameterübergabe	97
1.10.3	Lokale und globale Variablen	104
1.10.4	Standardparameter	105
1.10.5	Funktionen überladen	108
1.10.6	Inline-Funktionen	112
1.10.7	Rekursionen	115
1.10.8	main-Funktion	116
1.11	Präprozessor-Direktiven	117
1.11.1	Die #define-Direktive	118
1.11.2	Die #undef-Direktive	121
1.11.3	Die #include-Direktive	122
1.11.4	Die Direktiven #error und #pragma	123
1.11.5	Bedingte Kompilierung	124

2 Höhere und fortgeschrittene Datentypen 129

2.1	Zeiger	129
2.1.1	Zeiger deklarieren	130
2.1.2	Adresse im Zeiger speichern	131
2.1.3	Zeiger dereferenzieren	133
2.1.4	Zeiger, die auf andere Zeiger verweisen	137
2.1.5	Dynamisch Speicherobjekte anlegen und zerstören – new und delete	138
2.1.6	void-Zeiger	143
2.1.7	Konstante Zeiger	144
2.2	Referenzen	145
2.3	Arrays	147
2.3.1	Arrays deklarieren	147
2.3.2	Arrays initialisieren	148
2.3.3	Bereichsüberschreitung von Arrays	150
2.3.4	Anzahl der Elemente eines Arrays ermitteln	151
2.3.5	Arraywert von Tastatur einlesen	152
2.3.6	Mehrdimensionale Arrays	152
2.4	Zeichenketten (C-Strings) – char-Array	154
2.4.1	C-String deklarieren und initialisieren	155

2.4.2	C-String einlesen	155
2.4.3	C-Strings Bibliotheksfunktionen	156
2.5	Arrays und Zeiger	160
2.5.1	C-Strings und Zeiger	165
2.5.2	Mehrfache Indirektion	166
2.5.3	C-String-Tabellen	168
2.5.4	Arrays im Heap (Dynamisches Array)	170
2.6	Parameterübergabe mit Zeigern, Arrays und Referenzen	175
2.6.1	Call by value	176
2.6.2	Call by reference – Zeiger als Funktionsparameter	177
2.6.3	Call by reference mit Referenzen nachbilden	178
2.6.4	Arrays als Funktionsparameter	179
2.6.5	Mehrdimensionale Arrays an Funktionen übergeben	182
2.6.6	Argumente an die main-Funktion übergeben	183
2.7	Rückgabewerte von Zeiger, Arrays und Referenzen	185
2.7.1	Zeiger als Rückgabewert	185
2.7.2	Referenz als Rückgabewert	188
2.7.3	const-Zeiger als Rückgabewert	190
2.7.4	Array als Rückgabewert	190
2.7.5	Mehrere Rückgabewerte	191
2.8	Fortgeschrittene Typen	192
2.8.1	Strukturen	192
2.8.2	Unions	213
2.8.3	Aufzählungstypen	215
2.8.4	typedef	216

3 Gültigkeitsbereiche, spezielle Deklarationen und Typumwandlungen 219

3.1	Gültigkeitsbereiche (Scope)	219
3.1.1	Lokaler Gültigkeitsbereich (Local Scope)	220
3.1.2	Gültigkeitsbereich Funktionen	220
3.1.3	Gültigkeitsbereich Namensraum (Namespaces)	222
3.1.4	Gültigkeitsbereich Klassen (Class Scope)	222
3.2	Namensräume (Namespaces)	222
3.2.1	Neuen Namensbereich erzeugen (Definition)	222
3.2.2	Zugriff auf die Bezeichner im Namensraum	225
3.2.3	using – einzelne Bezeichner aus einem Namensraum importieren	228
3.2.4	using – alle Bezeichner aus einem Namensraum importieren	230

3.2.5	Namensauflösung	234
3.2.6	Aliasnamen für Namensbereiche	234
3.2.7	Anonyme (namenlose) Namensbereiche	235
3.2.8	Namensbereich und Headerdateien	236
3.3	C-Funktionen bzw. Bibliotheken in einem C++-Programm	238
3.3.1	C-Funktionen aus einer C-Bibliothek aufrufen	239
3.4	Speicherklassenattribute	243
3.4.1	Speicherklasse auto	244
3.4.2	Speicherklasse register	244
3.4.3	Speicherklasse static	244
3.4.4	Speicherklasse extern	245
3.4.5	Speicherklasse mutable	247
3.5	Typenqualifikatoren	247
3.5.1	Qualifizierer const	248
3.5.2	Qualifizierer volatile	248
3.6	Funktionsattribute	249
3.7	Typumwandlung	249
3.7.1	Standard-Typumwandlung	250
3.7.2	Explizite Typumwandlung	254

4 Objektorientierte Programmierung 259

4.1	OOP-Konzept versus prozedurales Konzept	259
4.1.1	OOP-Paradigmen	260
4.2	Klassen (fortgeschrittene Typen)	261
4.2.1	Klassen deklarieren	262
4.2.2	Elementfunktion (Klassenmethode) definieren	263
4.2.3	Objekte deklarieren	264
4.2.4	Kurze Zusammenfassung	265
4.2.5	private und public – Zugriffsrechte in der Klasse	266
4.2.6	Zugriff auf die Elemente (Member) einer Klasse	268
4.2.7	Ein Programm organisieren	275
4.2.8	Konstruktoren	279
4.2.9	Destruktoren	286
4.3	Mehr zu den Klassenmethoden (Klassenfunktionen)	289
4.3.1	Inline-Methoden (explizit und implizit)	289
4.3.2	Zugriffsmethoden	293
4.3.3	Read-only-Methoden	296
4.3.4	this-Zeiger	299
4.4	Verwenden von Objekten	301
4.4.1	Read-only-Objekte	301

4.4.2	Objekte als Funktionsargumente	301
4.4.3	Objekte als Rückgabewert	308
4.4.4	Klassen-Array (Array von Objekten)	309
4.4.5	Dynamische Objekte	312
4.4.6	Dynamische Klasselemente	318
4.4.7	Objekte kopieren (Kopierkonstruktor)	322
4.4.8	Dynamisch erzeugte Objekte kopieren (operator=())	323
4.4.9	Standardmethoden (Überblick)	325
4.4.10	Objekte als Elemente (bzw. Eigenschaften) in anderen Klassen	325
4.4.11	Teilobjekte initialisieren	332
4.4.12	Klassen in Klassen verschachteln	333
4.4.13	Konstante Klasseeigenschaften (Datenelemente)	335
4.4.14	Statische Klasseeigenschaften (Datenelemente)	337
4.4.15	Statische Klassenmethoden	341
4.4.16	friend-Funktionen bzw. friend-Klassen	343
4.4.17	Zeiger auf Eigenschaften einer Klasse	346
4.5	Operatoren überladen	352
4.5.1	Grundlegendes zur Operator-Überladung	352
4.5.2	Überladen von arithmetischen Operatoren	356
4.5.3	Überladen von unären Operatoren	365
4.5.4	Überladen von ++ und --	368
4.5.5	Überladen des Zuweisungsoperators	370
4.5.6	Überladen des Indexoperators [] (Arrays überladen)	371
4.5.7	Shift-Operatoren überladen	375
4.5.8	()-Operator überladen	378
4.5.9	new- und delete-Operator überladen	381
4.6	Typenumwandlung für Klassen	386
4.6.1	Konvertierungskonstruktor	387
4.6.2	Konvertierungsfunktion	388
4.7	Vererbung (Abgeleitete Klassen)	390
4.7.1	Anwendungsbeispiel (die Vorbereitung)	392
4.7.2	Die Ableitung einer Klasse	395
4.7.3	Redefinition von Klasseelementen	399
4.7.4	Konstrukturen	402
4.7.5	Destruktoren	405
4.7.6	Zugriffsrecht protected	405
4.7.7	Typenumwandlung abgeleiteter Klassen	408
4.7.8	Klassenbibliotheken erweitern	411

4.8	Polymorphismus	412
4.8.1	Statische bzw. dynamische Bindung	413
4.8.2	Virtuelle Methoden	413
4.8.3	Virtuelle Methoden redefinieren	418
4.8.4	Arbeitsweise von virtuellen Methoden	424
4.8.5	Virtuelle Destruktoren bzw. Destruktoren abgeleiteter Klassen	429
4.8.6	Polymorphismus und der Zuweisungsoperator	431
4.8.7	Rein virtuelle Methoden und abstrakte Basisklassen	433
4.8.8	Probleme mit der Vererbung und der <code>dynamic_cast</code> -Operator	437
4.8.9	Fallbeispiel: Verkettete Listen	439
4.9	Mehrfachvererbung	461
4.9.1	Indirekte Basisklassen erben	465
4.9.2	Virtuelle indirekte Basisklassen erben	469

5 Templates und STL **473**

5.1	Funktions-Templates	473
5.1.1	Funktions-Templates definieren	475
5.1.2	Typenübereinstimmung	478
5.1.3	Funktions-Templates über mehrere Module	479
5.1.4	Spezialisierung von Funktions-Templates	479
5.1.5	Verschiedene Parameter	483
5.1.6	Explizite Template-Argumente	484
5.2	Klassen-Templates	485
5.2.1	Definition	486
5.2.2	Methoden von Klassen-Templates definieren	487
5.2.3	Klassen-Template generieren (Instantiierung)	492
5.2.4	Weitere Template-Parameter	497
5.2.5	Standardargumente von Templates	500
5.2.6	Explizite Instantiierung	502
5.3	STL (Standard Template Library)	503
5.3.1	Konzept von STL	504
5.3.2	Hilfsmittel (Hilfsstrukturen)	508
5.3.3	Allokator	521
5.3.4	Iteratoren	521
5.3.5	Container	526
5.3.6	Algorithmen	577
5.3.7	Allokatoren	640

6 Exception-Handling 657

6.1	Exception-Handling in C++	658
6.2	Eine Exception auslösen	658
6.3	Eine Exception auffangen – Handle einrichten	659
6.3.1	Reihenfolge (Auflösung) der Ausnahmen	662
6.3.2	Alternatives catch(...)	662
6.3.3	Stack-Abwicklung (Stack-Unwinding)	664
6.3.4	try-Blöcke verschachteln	666
6.3.5	Exception weitergeben	669
6.4	Ausnahme-Klassen (Fehlerklassen)	672
6.4.1	Klassenspezifische Exceptions	674
6.5	Standard-Exceptions	676
6.5.1	Virtuelle Methode what()	677
6.5.2	Anwenden der Standard-Exceptions	677
6.6	System-Exceptions	682
6.6.1	bad_alloc	683
6.6.2	bad_cast	683
6.6.3	bad_typeid	683
6.6.4	bad_exception	683
6.7	Exception-Spezifikation	684
6.7.1	Unerlaubte Exceptions	685
6.7.2	terminate-Handle einrichten	687

7 C++-Standardbibliothek 691

7.1	Die String-Bibliothek (string-Klasse)	691
7.1.1	Exception-Handling	693
7.1.2	Datentypen	693
7.1.3	Strings erzeugen (Konstruktoren)	694
7.1.4	Zuweisungen	696
7.1.5	Elementzugriff	698
7.1.6	Länge und Kapazität ermitteln bzw. ändern	699
7.1.7	Konvertieren in einen C-String	702
7.1.8	Manipulation von Strings	703
7.1.9	Suchen in Strings	706
7.1.10	Strings vergleichen	713
7.1.11	Die (überladenen) Operatoren	715
7.1.12	Einlesen einer ganzen Zeile	717
7.2	Ein-/Ausgabe Klassenhierarchie (I/O-Streams)	718
7.2.1	Klassen für Ein- und Ausgabe-Streams	721

7.2.2	Klassen für Datei-Streams (File-Streams)	744
7.2.3	Klassen für String-Streams	759
7.2.4	Die Klasse streambuf	766
7.2.5	Die Klasse filebuf	770
7.2.6	Die Klasse stringbuf	771
7.2.7	Die Klasse stdiobuf	772
7.3	Numerische Bibliothek(en)	774
7.3.1	Komplexe Zahlen (complex-Klasse)	774
7.3.2	valarray	776
7.3.3	Globale numerische Funktionen (cmath und cstdlib)	799
7.3.4	Grenzwerte von Zahlentypen	803
7.3.5	Halbnumerische Algorithmen	808
7.4	Typenerkennung zur Laufzeit	811

8 Weiteres zum C++-Guru 817

8.1	Module	817
8.1.1	Aufteilung	818
8.1.2	Die öffentliche Schnittstelle (Headerdatei)	819
8.1.3	Die private Datei	820
8.1.4	Die Client-Datei	822
8.1.5	Speicherklassen extern und static	823
8.1.6	Werkzeuge	825
8.2	Von C zu C++	826
8.2.1	Notizen	826
8.2.2	Kein C++	827
8.2.3	Kein C	829
8.2.4	malloc und free oder new und delete	830
8.2.5	setjmp und longjmp oder catch und throw	831
8.3	»Altes« C++	831
8.3.1	Headerdateien mit und ohne Endung	831
8.3.2	Standardbibliothek nicht komplett oder veraltet	832
8.3.3	Namespace (Namensbereiche)	832
8.3.4	Schleifenvariable von for	832
8.4	UML	833
8.4.1	Wozu UML?	833
8.4.2	UML-Komponenten	835
8.4.3	Diagramme erstellen	835
8.4.4	Klassendiagramme mit UML	836
8.5	Programmierstil	876
8.5.1	Kommentare	877

8.5.2	Code	879
8.5.3	Benennung	879
8.5.4	Codeformatierung	880
8.5.5	Zusammenfassung	881
8.6	Boost	881
8.6.1	Boost.Regex (Reguläre Ausdrücke)	883

9 Netzwerkprogrammierung und Cross-Plattform-Entwicklung in C++ 899

9.1	Begriffe zur Netzwerktechnik	900
9.1.1	IP-Nummern	900
9.1.2	Portnummer	901
9.1.3	Host- und Domainname	902
9.1.4	Nameserver	903
9.1.5	Das IP-Protokoll	903
9.1.6	TCP und UDP	903
9.1.7	Was sind Sockets?	904
9.2	Headerdateien zur Socketprogrammierung	905
9.2.1	Linux/UNIX	905
9.2.2	Windows	905
9.3	Client-/Server-Prinzip	907
9.3.1	Loopback-Interface	908
9.4	Erstellen einer Client-Anwendung	908
9.4.1	socket() – Erzeugen eines Kommunikationsendpunkts ...	909
9.4.2	connect() – Client stellt Verbindung zum Server her	910
9.4.3	Senden und Empfangen von Daten	916
9.4.4	close(), closesocket()	918
9.5	Erstellen einer Server-Anwendung	919
9.5.1	bind() – Festlegen einer Adresse aus dem Namensraum	919
9.5.2	listen() – Warteschlange für eingehende Verbindungen einrichten	921
9.5.3	accept() und die Server-Hauptschleife	922
9.6	Cross-Plattform-Development	924
9.6.1	Abstraktion Layer	925
9.6.2	Headerdatei (socket.h)	925
9.6.3	Quelldatei (socket.cpp)	927
9.6.4	TCP-Echo-Server (Beispiel)	937
9.6.5	Exception-Handling integrieren	940
9.6.6	Server- und Client-Sockets erstellen (TCP)	946

9.6.7	Ein UDP-Beispiel	955
9.7	Mehrere Clients gleichzeitig behandeln	958
9.8	Weitere Anmerkungen zur Netzwerkprogrammierung	967
9.8.1	Das Datenformat	968
9.8.2	Der Puffer	968
9.8.3	Portabilität	969
9.8.4	Von IPv4 nach IPv6	969
9.8.5	RFC-Dokumente (Request for Comments)	971
9.8.6	Sicherheit	971
9.8.7	Fertige Bibliotheken	972

10 GUI- und Multimediaprogrammierung in C++ 973

10.1	GUI-Programmierung – Überblick	973
10.1.1	Low-Level	973
10.1.2	High-Level	974
10.1.3	Überblick zu plattformunabhängigen Bibliotheken	975
10.1.4	Überblick zu plattformabhängigen Bibliotheken	977
10.2	Multimedia- und Grafikprogrammierung – Überblick	977
10.2.1	Überblick zu plattformunabhängigen Bibliotheken	978
10.2.2	Überblick zu plattformabhängigen Bibliotheken	980
10.3	GUI-Programmierung mit wxWidgets	981
10.3.1	Warum wxWidgets?	981
10.3.2	Das erste Programm – Hallo Welt	982
10.3.3	Die grundlegende Struktur eines wxWidgets-Programm	985
10.3.4	Event-Handle (Ereignisse behandeln)	991
10.3.5	Die Fenster-Grundlagen	999
10.3.6	Übersicht zu den wxWidgets-(Fenster-)Klassen	1001
10.3.7	wxWindow, wxControl und wxControlWithItems – Die Basisklassen	1002
10.3.8	Top-Level-Fenster	1006
10.3.9	Container Fenster	1030
10.3.10	Nicht statische Kontroll-Elemente	1057
10.3.11	Statische Kontroll-Elemente	1116
10.3.12	Menüs	1121
10.3.13	Ein Beispiel – Text-Editor	1137
10.3.14	Standarddialoge	1152
10.3.15	Weitere Elemente und Techniken im Überblick	1176

11 Anhang	1189
11.1 Operatoren in C++ und deren Bedeutung (Übersicht)	1189
11.2 Vorrangtabelle der Operatoren	1191
11.3 Schlüsselwörter von C++	1192
11.4 Informationsspeicherung	1192
11.4.1 Zahlensysteme	1193
11.5 Zeichensätze	1200
11.5.1 ASCII-Zeichensatz	1201
11.5.2 ASCII-Erweiterungen	1202
11.5.3 Unicode	1204
Index	1207