

Inhaltsverzeichnis

1. Ziel und Inhalt dieses Buchs	1
1.1 Zum Inhalt	3
1.2 Motivation	6
1.3 Feedback	8
1.4 Die beiliegende CD-ROM	9
Teil I: Low-Level Konzepte von C++	11
2. Datentypen und Variablen	13
2.1 Primitive Datentypen	13
2.2 Deklaration, Definition und Initialisierung	20
2.3 Das erste C++ Programm	25
2.4 Zusammengesetzte Datentypen	31
2.4.1 Arrays	31
2.4.2 Structures	37
2.4.3 Unions	40
2.5 Scope und Lifetime	46
2.6 Symbolische Konstanten	47
2.7 Eigene Typdefinitionen	49
3. Operatoren	53
3.1 Überblick und Reihenfolge der Auswertung	53
3.2 Arithmetische Operatoren	57
3.3 Logische- und Vergleichsoperatoren	59
3.4 Bitoperatoren	60
3.5 Zuweisungen	61
3.6 Datentypabfragen und explizite Typecasts	61
3.6.1 Type Identification und Run-Time-Type-Information	62
3.6.2 Unchecked Cast	67
3.6.3 Compiletime-checked Cast	67
3.6.4 Runtime-checked Cast	69
3.6.5 Remove-const Cast	70
3.6.6 C-Style Casts	70

4. Kontrollstrukturen	73
4.1 Selection Statements	74
4.2 Schleifen	79
4.3 Das unselige goto Statement	82
5. Funktionen	85
6. Pointer und References	103
6.1 References	103
6.2 Pointer	114
6.2.1 Pointer und Adressen	115
6.2.2 Dynamische Memory Verwaltung	119
6.2.3 Strings	125
6.2.4 Funktionspointer	125
6.2.5 Besondere Aspekte von Pointern	126
Call-by-reference auf Pointer	126
Mehrfachpointer	128
Pointer und Typecasts	131
7. Der Preprocessor	135
7.1 Include Files	136
7.2 Bedingte Übersetzung	137
7.3 Macros	138
Teil II: Objektorientierte Konzepte von C++	141
8. Objektorientierung Allgemein	143
8.1 Module und Abläufe	146
8.1.1 Der Weg zum Arbeitsplatz – ein kleines Beispiel	146
8.2 Klassen und Objekte	155
8.3 Richtige Verwendung der OO Mechanismen	161
9. Klassen in C++	167
9.1 Besonderheiten von Structures in C++	168
9.2 Einfache Klassen	171
9.2.1 Konstruktor und Destruktor genauer beleuchtet	178
9.2.2 Der Copy Konstruktor	184
9.2.3 Initialisierung vs. Zuweisung	188
9.2.4 Deklarieren von Konstruktoren als explicit	189
9.2.5 Object- und Class-Members	192
9.3 Abgeleitete Klassen	196
9.3.1 Mehrfachvererbung	205
9.3.2 Konstruktoren und Destruktoren	215
9.4 Weitere wichtige technische Aspekte	219
9.4.1 Static und Dynamic Binding	219

9.4.2	Abstrakte Methoden und Klassen	232
9.4.3	Virtuelle Ableitung	235
9.4.4	Downcasts von Klassen	241
9.4.5	Friends von Klassen	247
9.4.6	Overloading von <code>const</code> und <code>non-const</code> Methoden ...	250
9.4.7	Besonderheiten bei der Initialisierung	251
9.4.8	Temporäre Objekte	253
10.	Memory – ein kleines Beispiel	257
10.1	Das ADD	257
10.1.1	Identifikation der Grobmodule	258
10.1.2	Weitere Zerlegung der einzelnen Grobmodule	259
	Input Handling	259
	Spielsteuerung	259
	Output Handling	260
	Memory Spielfeld	260
	Memory Karte	260
	Commandline Handling	260
10.2	Das DDD	260
10.2.1	Klassendiagramm	261
10.2.2	Klassendeklarationen	261
10.2.3	<code>Vector</code>	263
10.2.4	<code>ObjectDeletor</code>	266
10.2.5	Konkrete Deletors	267
10.2.6	<code>ArgumentHandler</code>	271
10.2.7	<code>MemoryCommandLineArgumentHandler</code>	272
10.2.8	<code>CommandLineHandling</code>	273
10.2.9	<code>SimpleOutputHandling</code>	275
10.2.10	<code>Displayable</code>	278
10.2.11	<code>OutputContext</code>	279
10.2.12	<code>TextOutputContext</code>	279
10.2.13	<code>GameCard</code>	281
10.2.14	<code>MemoryGameCard</code>	282
10.2.15	<code>MemoryGameboard</code>	284
10.2.16	<code>IntDisplayable</code>	289
10.2.17	<code>TextDisplayable</code>	290
10.2.18	<code>Event</code>	292
10.2.19	<code>WordEvent</code>	293
10.2.20	<code>SimpleInputHandling</code>	294
10.2.21	<code>SimpleEventHandling</code>	296
10.2.22	<code>MemoryGameControl</code>	296
10.2.23	<code>MemoryCardSymbolGenerator</code>	300
10.2.24	<code>MemoryCardpair</code>	302
10.3	Auszüge aus der Implementation	304

11. Exceptions	311
12. Operator Overloading	335
12.1 Grundprinzipien des Operator Overloadings	335
12.2 Typumwandlungen	354
12.3 Speicherverwaltung	363
12.3.1 Einfaches <code>new</code> und <code>delete</code>	363
12.3.2 Array <code>new</code> und <code>delete</code>	371
12.3.3 Placement Operator <code>new</code>	374
12.3.4 <code>delete</code> mit zwei Parametern	381
12.3.5 Globale <code>new</code> und <code>delete</code> Operatoren	385
12.3.6 Weitere Aspekte der eigenen Speicherverwaltung	391
Vererbung von <code>new</code> und <code>delete</code>	391
Verhalten bei “Ausgehen” des Speichers	397
12.4 Abschließendes zu overloadable Operators	400
13. Templates	405
13.1 Function Templates	407
13.2 Overloading Aspekte von Function Templates	414
13.3 Class Templates	419
13.4 Ableiten von Class Templates	428
13.5 Explizite Spezialisierungen	431
13.6 Verschiedenes zu Templates	444
13.7 Source Code Organisation	447
14. Namespaces	453
15. Verschiedenes	463
15.1 <code>mutable</code> Member Variablen	463
15.2 Unions im OO Kontext	465
15.3 Funktionspointer	470
15.4 Besondere Keywords, Diagraphs und Trigraphs	475
15.5 <code>volatile</code> Objekte	477
15.6 RTTI und <code>dynamic_cast</code> im OO Kontext	477
15.7 Weiterführendes zu Exceptions	482
Teil III: Ausgesuchte Teile aus der C++ Standard Library ...	491
16. Die C++ Standard Library	493
16.1 Übersicht	493
16.2 Container	497
16.2.1 Vektoren	497
16.2.2 Listen	499
16.2.3 Double-Ended Queues	501
16.2.4 Standard Queues	502

16.2.5	Priority Queues	503
16.2.6	Stacks	505
16.2.7	Maps	506
16.2.8	Sets	507
16.2.9	Zusammenfassung der Container-Operationen	510
16.3	Iterators	513
16.4	Allocators	517
16.5	Strings	518
16.6	Streams	521
16.7	Numerik	531
16.8	Algorithmen und Funktionsobjekte	534
A.	Coding-Standard	537
A.1	Generelle Regeln	537
A.2	Coding-Rules	538
A.3	Design Guidelines	541
B.	Vollständige Implementation des Memory Spiels	543
B.1	Implementationen der einzelnen Klassen	543
B.1.1	Das Hauptprogramm	543
B.1.2	Implementation von <code>Vector</code>	545
B.1.3	Implementation von <code>CommandLineHandling</code>	546
B.1.4	Implementation von <code>SimpleOutputHandling</code>	547
B.1.5	Implementation von <code>GameCard</code>	547
B.1.6	Implementation von <code>MemoryGameCard</code>	547
B.1.7	Implementation von <code>MemoryGameboard</code>	548
B.1.8	Implementation von <code>SimpleInputHandling</code>	552
B.1.9	Implementation von <code>MemoryGameControl</code>	552
B.1.10	Implementation von <code>MemoryCardSymbolGenerator</code>	554
B.1.11	Implementation von <code>MemoryCardpair</code>	556
B.1.12	Variablen für die konkreten Deletors	557
B.1.13	Das <code>MemoryMakefile</code>	557
	Literaturverzeichnis	561
	Index	563