

Inhaltsverzeichnis

Vorwort	23
Einleitung	27

TEIL A EINFÜHRUNG IN BORLAND C++

Kapitel 1	Die Programmiersprachen C und C++	33
1.1	Von Classic C nach ANSI C	34
	Kernighan & Ritchie und Classic C	34
	Standardisierung durch ANSI C	36
1.2	Entwicklung und Standardisierung von C++	37
	Ansatzpunkte für C++	37
	Unterschiede zwischen C und C++	40
	Standardisierung durch ANSI C++	41
Kapitel 2	Programmieren mit Borland C++	43
2.1	Das Entwicklungssystem Borland C++	44
	Neue Leistungsmerkmale von Borland C++	44
	Voraussetzungen für Borland C++	46
	Komponenten von Borland C++	47
	Borland C++ installieren	49
2.2	Borland C++ und DOS	50
	Mit Classic C und ANSI C programmieren	51
	Objektorientiert programmieren	59
2.3	Borland C++ und MS Windows	63
	Voraussetzungen für Windows-Programme	63
	Programmieren für MS Windows	64

Kapitel 3	Von der Aufgabe zum Programm	69
3.1	Teilaufgaben der Programmentwicklung	70
3.2	Aufgabenstellung analysieren	71
3.3	Teilaufgaben bestimmen	72
3.4	Daten- und Algorithmenstrukturen entwerfen	72
3.5	Sourcecode erstellen	76
3.6	Programm übersetzen und linken	81
3.7	Programm ausführen und testen	84
3.8	Programmfehler suchen und beseitigen	85
3.9	Programm dokumentieren	87
3.10	Programm optimieren, anpassen und pflegen	88
Kapitel 4	Die Entwicklungsumgebung (IDE) von Borland C++	91
4.1	Arbeiten mit der Entwicklungsumgebung	92
	Die Einsatzbereiche der IDE	92
	Fenster und Felder	94
	Projekt- und Konfigurations-Dateien	96
4.2	Arbeiten mit dem Editor	96
	Cursor bewegen	96
	Einfügen und Löschen	97
	Blockbefehle	97
	Sonstige Editorbefehle	98
4.3	Die Menühierarchie bei Borland C++	99
4.4	Das Menü System	103
4.5	Das Menü File	104
4.6	Das Menü Edit	104
4.7	Das Menü Search	105
4.8	Das Menü Run	106
4.9	Das Menü Compile	107
4.10	Das Menü Debug	107
4.11	Das Menü Project	108
4.12	Das Menü Options	109
4.13	Das Menü Window	110

4.14	Das Menü Help	111
4.15	Menüs bei Turbo C++ für Windows	112
Kapitel 5	Sprachkonzepte und Programmstrukturen	115
5.1	Die Sprachversionen C und C++	116
5.2	Die Bausteine von C++-Programmen	118
5.3	Allgemeine Programmstrukturen	121
5.4	Präprozessor-Direktiven	125
5.5	Deklarationen und Definitionen	127
5.6	Operatoren und Ausdrücke	128
5.7	Anweisungen	129
5.8	Funktionen und Bibliotheken	130
5.9	Fallstudie: erster Syntax-Check	131
Kapitel 6	Borland C++ und objektorientierte Programmierung	139
6.1	Objektorientierte Erweiterungen	140
6.2	Klassen, Objekte und Elemente	142
6.3	Methoden, Nachrichten und Ereignisse	144
6.4	Zugriffs-Spezifizierer und Friends	145
6.5	Konstruktoren und Destruktoren	146
6.6	Überladen und Polymorphismus	147
6.7	Abgeleitete Klassen und Vererbung	147
TEIL B	SPRACHELEMENTE UND SYNTAX VON C++	
Kapitel 7	Bausteine und lexikalische Elemente	153
7.1	Vom Sourcecode zum Token	154
7.2	Zeichenvorrat und Sonderzeichen	155
7.3	Bezeichner und reservierte Wörter	156
	Bezeichner für Bausteine	156
	Reservierte Wörter für C und C++	156

7.4	Literale und Konstanten	158
	Integer-Konstanten	158
	Gleitkomma-Konstanten	158
	Zeichen-Konstanten	159
	String-Literale	159
	Aufzählungs-Konstanten	160
7.5	Trennzeichen und Kommentare	160
	Trennzeichen (white space)	160
	Kommentare bei C und C++	160
Kapitel 8	Der Programmaufbau	163
8.1	Die Modularisierung des Programms	164
8.2	Externe und interne Vereinbarungen	166
8.3	Die Bausteine der Programmstruktur	169
Kapitel 9	Datenstrukturen vereinbaren	171
9.1	Datenobjekte vereinbaren	172
9.2	Geltungsbereiche und Sichtbarkeit	173
9.3	Speicherklassen	174
9.4	Den Datentyp spezifizieren	175
9.5	Bezeichner vereinbaren	176
9.6	Variablen initialisieren	177
9.7	Objekte, Klassen und Methoden	178
Kapitel 10	Einfache Datentypen	183
10.1	Einfache und abgeleitete Datentypen	184
10.2	Der Datentyp character	185
10.3	Der Datentyp integer	186
10.4	Der Datentyp floating	187
10.5	Der Datentyp Aufzählung	188
Kapitel 11	Ausdrücke und Operatoren	189
11.1	Syntax von Ausdrücken und Operatoren	190
11.2	Arithmetische Ausdrücke und Operatoren	192

11.3	Logische Ausdrücke und Operatoren	194
11.4	Relationale Ausdrücke und Operatoren	195
11.5	Bitweise Ausdrücke und Operatoren	196
11.6	Bedingte Ausdrücke und Operatoren	198
11.7	Zuweisungs-Ausdrücke und Operatoren	199
11.8	Komma-Ausdrücke und Operatoren	201
11.9	Konstanten-Ausdrücke	201
11.10	Primary-Ausdrücke	202
11.11	Postfix-Ausdrücke	203
11.12	Die Operatoren new und delete	203
11.13	Pointer-to-member-Operatoren	204
11.14	Type-cast-Ausdrücke	205
11.15	Konvertierung von Datentypen	205
	Explizite und implizite Konvertierung	205
	Integrale Promotions	206
	Integrale Konvertierungen	207
	Float- und Double-Konvertierungen	207
	Floating- und integrale Konvertierungen	207
	Arithmetische Konvertierungen	207
	Pointer-Konvertierungen	208
	Referenz-Konvertierungen	208
	Pointer-to-member-Konvertierungen	208
11.16	Rangfolge und Assoziativität	209
Kapitel 12	Anweisungen	211
12.1	Die Anweisungen bei C und C++	212
12.2	Label-Anweisungen	212
12.3	Ausdruck-Anweisungen	213
12.4	Verbund-Anweisungen	215
12.5	Auswahl-Anweisungen	215
	Die bedingte Anweisung if	216
	Die bedingte Anweisung switch	216
12.6	Wiederholungs-Anweisungen	217
	Die Wiederholungs-Anweisung while	218
	Die Wiederholungs-Anweisung do	218
	Die Wiederholungs-Anweisung for	219

12.7	Sprung-Anweisungen	220
	Die Sprung-Anweisung break	220
	Die Sprung-Anweisung continue	221
	Die Sprung-Anweisung return	222
	Die Sprung-Anweisung goto	222
12.8	Vereinbarungs-Anweisungen	223
12.9	Fallstudie: Eingabe abfragen	224
Kapitel 13	Abgeleitete Datentypen	227
13.1	Abgeleitete Datentypen	228
13.2	Der Datentyp array	228
13.3	Der Datentyp structure	229
13.4	Der Datentyp union	231
13.5	Der Datentyp function	232
13.6	Der Datentyp pointer	233
13.7	Der Datentyp class	234
	Eine Klasse vereinbaren	234
	Die Elemente einer Klasse	235
	Vererbung und Basisklassen	236
	Zugriff auf Elemente	237
	Objekte einer Klasse	238
	Spezielle Elementfunktionen	239
13.8	Der Datentyp pointer to class member	241
13.9	Der Datentyp reference	241
13.10	Der Datentyp constant	242
Kapitel 14	Der Datentyp Funktion	245
14.1	Funktionen vereinbaren	246
14.2	Funktion aufrufen	250
14.3	Parameter übergeben	251
14.4	Funktionswerte zurückgeben	254
14.5	Funktionsprototypen	255
Kapitel 15	Der Datentyp Pointer	257
15.1	Pointer-Variablen	258

15.2	Die Operatoren & und *	260
15.3	Mit Pointern arbeiten	261
15.4	Stapel, Warteschlangen, Listen und Bäume	264

Kapitel 16 Objektorientierte Programmstrukturen 269

16.1	Abstraktion, Kapselung und Vererbung	270
16.2	Polymorphismus bei C++	270
16.3	Klassen, Klasselemente und Objekte	271
16.4	Basisklassen und abgeleitete Klassen	274
16.5	Einfache und mehrfache Vererbung	275
16.6	Friend-Funktionen	277
16.7	Templates	277

Kapitel 17 Bibliotheks-Routinen bei Borland C++ 279

17.1	Header-Dateien nach ANSI C	280
17.2	Weitere Header-Dateien für Borland C	281
17.3	Weitere Header-Dateien für Borland C++	281
17.4	Bibliotheks-Routinen nach Gruppen	282
17.5	Klassifizierungs-Routinen	283
17.6	Konvertierungs-Routinen	285
17.7	Verzeichnis-Routinen	289
17.8	Diagnose-Routinen	293
17.9	Grafik-Routinen	293
17.10	Input-/Output-Routinen	321
17.11	DOS-, BIOS- und hardwarenahe Routinen	360
17.12	String- und Speicher manipulations-Routinen	376
17.13	Mathematische Routinen	393
17.14	Dynamische Speicherverwaltungs-Routinen	410
17.15	Sonstige Routinen	417
17.16	Prozeß-Routinen	418
17.17	Standard-Routinen	426
17.18	Bildschirmsteuerungs-Routinen	434
17.19	Datum- und Uhrzeit-Routinen	440
17.20	Routinen für variable Parameterliste	444

Kapitel 18	Der Präprozessor	445
18.1	Die Direktiven #define und #undef	446
	Textersatz mit #define	446
	Makrodefinition mit #define	446
	Bezeichner löschen mit #undef	447
18.2	Die Direktiven #include, #line und #error	447
	Datei einbinden mit #include	447
	Ortsangaben mit #line	448
	Meldung ausgeben mit #error	448
18.3	Bedingte Übersetzung mit #if ... #endif	449
	Die Direktiven #if, #elif, #else und #endif	450
	Die Direktiven #ifdef und #ifndef	450
18.4	Pragmas und vordefinierte Makronamen	450
	Die Direktive #pragma	450
	Vordefinierte Makronamen	451
TEIL C	PROGRAMMIEREN MIT BORLAND C++	
Kapitel 19	Konzepte und Anregungen	455
19.1	Programmieransätze und Programmierstil	456
19.2	Speicherplatz und Laufzeit optimieren	457
19.3	Lesbarkeit, Flexibilität und Portabilität	458
19.4	Fallstudie: Dateiattribute anzeigen	461
Kapitel 20	Mit Bibliotheken arbeiten	465
20.1	Funktionen der Standard-Bibliothek einsetzen	466
20.2	Formatierte Ein- und Ausgabe	467
20.3	Funktionen zur Stringbearbeitung	470
20.4	Funktionen zur Dateibearbeitung	473
20.5	Grafische Funktionen einsetzen	476
20.6	Fallstudie: Dateien prüfen	477

Kapitel 21	Mit Daten und Algorithmen programmieren	491
21.1	Aufgaben- und Programmstrukturen	492
21.2	Programm = Daten + Algorithmen	493
21.3	Daten- und Algorithmenstrukturen definieren	494
21.4	Stringverarbeitung	495
21.5	Dateiverarbeitung	496
21.6	Daten anzeigen, eingeben und prüfen	499
21.7	Eine grafische Menüoberfläche programmieren	500
21.8	Fallstudie: Datei dumpen	503
Kapitel 22	Mit Objekten und Methoden programmieren	515
22.1	Programm = Objekte + Methoden	516
22.2	Benutzerdefinierte Strukturen	516
22.3	Methoden und Funktionen	517
Kapitel 23	Text und Grafik ausgeben	523
23.1	Bildschirme und Geräte-Treiber	524
23.2	Video-Modus für Text und Grafik	525
23.3	Im Text-Modus programmieren	528
23.4	Im Grafik-Modus programmieren	531
Kapitel 24	Low-level-Programmierung	543
24.1	Ansätze und Bedarf	544
24.2	Ebenen der Low-level-Programmierung	546
24.3	DOS-Funktionsaufrufe	547
24.4	BIOS-Interrupts	548
24.5	Sprachenschnittstelle	552
24.6	Assembler-Routinen in Borland C++	553
24.7	C-Funktionen in Assembler-Routinen aufrufen	555
24.8	Pseudo-Variablen, Inline-Assemblierung und Interrupt-Aufrufe	556

24.9	Fallstudie: Tasten abfragen	559
24.10	Low-level-Bibliothek zur Bildschirmansteuerung	561
24.11	Anforderungen an Funktionen und Interface	562
24.12	Symbolische Konstanten- und Typ-Definitionen	563
24.13	Direct-Memory-Access-Funktionen	564
24.14	Implementierung der Low-level-Bibliothek	565

Kapitel 25 Speicher und Bibliotheken verwalten 575

25.1	Speichermodelle	576
25.2	Speicherzugriff mit Near und Far	577
25.3	Arbeiten mit Overlays	578
25.4	Programme modularisieren	579
25.5	Eigene Bibliotheken aufbauen	580
25.6	Bibliotheken aktualisieren	581
25.7	Fallstudie: Maus implementieren	583

Kapitel 26 Hilfen zum Debugging 591

26.1	Fehlerentstehung und Fehlerklassen	592
26.2	Fehlerrangfolge, Testumgebungen und Testdaten	593
26.3	Fehlersuche und Programmentwicklung	593
26.4	Fehlereinkreisung und Fehlerdiagnose	594
26.5	Typische Fehlerquellen	595
26.6	Debugging mit Borland C++	596
26.7	Programmablauf verfolgen (IDE)	597
26.8	Breakpoints und Watches setzen (IDE)	598
26.9	Daten prüfen und ändern (IDE)	599

TEIL D OBJEKTORIENTIERTE PROGRAMMIERUNG MIT BORLAND C++

Kapitel 27 Neue Klassen vereinbaren 603

27.1	Prozedurale vs. objektorientierte Programmierung	604
27.2	Klassen bilden	604

27.3	Ableitungen, Protokolle und Zugriffsrechte	605
Kapitel 28	Objekte, Methoden und Container	609
28.1	Fallstudie 1: objektorientierte Listenstrukturen	610
28.2	Fallstudie 2: Definition einer Container-Klasse	612
28.3	Fallstudie 3: generische Datenstrukturen	614
Kapitel 29	Container-Klassen und Klassen-Bibliothek	619
29.1	Container-Klassen und Klassenhierarchie	620
29.2	Die Klassen-Bibliothek	622
	Dateikommentare und Verzeichnisse	622
	Referenz der Klassen-Bibliothek	623
Kapitel 30	Arbeiten mit objektorientierten Streams	639
30.1	Die I/O-Stream-Bibliothek	640
30.2	Ausgabe und Eingabe mit Streams	640
	Die Stream-Ausgabe	640
	Die Stream-Eingabe	642
30.3	Stream-Klassen	642
Kapitel 31	Debugging von Klassen und Objekten	651
31.1	Debugging im Hierarchy-Fenster	652
31.2	Debugging im Inspector-Fenster	653
TEIL E	BORLAND C++ UND MS WINDOWS	
Kapitel 32	Programmentwicklung für MS Windows	659
32.1	Kompilieren und Linken mit der IDE	660
32.2	Programmieren mit Turbo C++ für Windows	662
32.3	Arbeiten mit dem Kommandozeilen-Compiler	664
32.4	Fallstudie 1: WINHALLO.C	666
32.5	Fallstudie 2: WINEDIT.C	675

Kapitel	33	Arbeiten mit dem Resource Workshop	681
	33.1	Windows-Ressourcen, Editoren und Projekte	682
	33.2	Der Dialogfenster-Editor	685
	33.3	Der Menü-Editor	688
	33.4	Der Tastenkürzel-Editor	690
	33.5	Der String-Editor	692
	33.6	Der Paint-Editor	694
		Icons erstellen	695
		Cursor definieren	698
		Bitmaps erstellen	701
		Schriftfonts erstellen	705
	33.7	Benutzerdefinierte Ressourcen	709
Kapitel	34	Windows-API und DLL	713
	34.1	Schnittstellen zu MS Windows	714
	34.2	DLL-Bibliotheken	723
Kapitel	35	Debugging mit dem Turbo Debugger für Windows	725
	35.1	Leistungsmerkmale und Voraussetzungen	726
	35.2	Windows-Nachrichten	727
	35.3	Speicher- und Modullisten	728
	35.4	DLL debuggen	729
TEIL	F	FORTGESCHRITTENE KOMPONENTEN DER PROGRAMMENTWICKLUNG	
Kapitel	36	Kommandozeilen-Compiler und Projektverwaltung	735
	36.1	Programmieren mit dem Kommandozeilen-Compiler	736
		Arbeiten mit dem Compiler	736

	Compiler-Optionen	736
36.2	Projekt-Dateien und Programmverwaltung	738
Kapitel 37	Debugging mit dem Turbo Debugger	745
37.1	Anwendungen, Aufgaben und Leistungsmerkmale	746
	Anwendungen und Aufgaben	746
	Leistungsmerkmale	748
	Hardware- und Software-Voraussetzungen	749
	Turbo Debugger installieren	749
37.2	Arbeiten mit dem Turbo Debugger	750
	Aufrufoptionen und Starten	750
	Bildschirmaufteilung und Fensterelemente	751
	Hilfeinformationen anfordern	753
	Eine Debugger-Arbeitssitzung	753
37.3	Die Menühierarchie beim Turbo Debugger	757
	Hauptmenü und Menüstruktur	757
	Menüoptionen auswählen und aufrufen	759
	Menüs, Befehle und Optionen	760
37.4	Der Debugger im Einsatz	765
	Programmausführung steuern	765
	Breakpoints setzen	767
	Daten analysieren und ändern	769
	Dateien untersuchen und editieren	770
	Ausdrücke auswerten	771
37.5	Fortgeschrittenes Debugging	772
	Debugging von MS Windows-Programmen	772
	Debugging von C++-Programmen	772
	Assemblercode debuggen	773
	Virtuell debuggen	774
	Debugging im Protected Mode	774
	TSR-Programme und Gerätetreiber debuggen	775
	Remote-Debugging	777
Kapitel 38	Analysieren mit dem Turbo Profiler	779
38.1	Anwendungen, Aufgaben und Leistungsmerkmale	780
	Anwendungen und Aufgaben	780

	Leistungsmerkmale	780
	Hardware- und Software-Voraussetzungen	781
	Turbo Profiler installieren	782
38.2	Arbeiten mit dem Turbo Profiler	783
	Aufrufoptionen und Starten	783
	Bildschirmaufteilung und Fensterelemente	784
	Hilfeinformationen anfordern	785
	Zeitanalyse von Bereichsdurchläufen	785
	Zeit- und Häufigkeitsanalyse von Bereichsdurchläufen	787
	Statistik der Programmausführung auswerten	788
	Die Arbeitssitzung beenden	789
38.3	Die Menühierarchie beim Turbo Profiler	789
	Hauptmenü und Menüstruktur	789
	Menüoptionen auswählen und aufrufen	791
	Menüs, Befehle und Optionen	791
38.4	Programmanalyse und -optimierung	795
	Zur Arbeitsweise des Turbo Profilers	796
	Vorgehensweise und Ansätze	796
	Die Programmanalyse	796
	Die Programmoptimierung	798
38.5	Analyseverfahren	799
	Zeit-Analysen	800
	Häufigkeits-Analysen	801
	Aufruf-Analysen	801
	Overlay-Analysen	801
	Interrupt-Analysen	802
	Files-Analysen	803
	Disassembly-Analysen	804
38.6	Fortgeschrittene Analysen	805
	Remote-Analysen	805
	Analysen im Virtual Mode	806
Kapitel 39	Low-level-Programmierung mit dem Turbo Assembler	809
39.1	Leistungsmerkmale und Voraussetzungen	810
	Leistungsmerkmale	810

	Hardware- und Software-Voraussetzungen	811
	Turbo Assembler installieren	811
39.2	Arbeiten mit dem Turbo Assembler	811
	Aufrufoptionen und Starten	811
	Eine Arbeitssitzung mit dem Turbo Assembler	813
	Ideal-Modus und Quirks-Modus des Turbo Assemblers	814
39.3	Sprachelemente beim Turbo Assembler	815
	Vordefinierte Symbole	815
	Rangfolge von Operatoren	817
	Operatoren	818
	Makro-Operatoren	822
	Direktiven	822
	Prozessor-Befehle	837
39.4	Dienstprogramme für den Turbo Assembler	839
	Programme verwalten mit MAKE	839
	Dateien linken mit TLINK	839
	Bibliotheken verwalten mit TLIB	840

TEIL G ANHÄNGE

Anhang A	Reservierte Wörter	843
Anhang B	Präprozessor (ANSI C++)	845
Anhang C	Syntax (ANSI C++)	846
C.1	Keywords (reservierte Wörter)	846
C.2	Expressions (Ausdrücke)	846
C.3	Declarations (Vereinbarungen)	849
C.4	Declarators (Deklaratoren)	851
C.5	Class Declarations (Klassen)	853
C.6	Statements (Anweisungen)	855
C.7	Templates	856
C.8	Exception Handling	856

Literaturverzeichnis	859
-----------------------------	-----

Stichwortverzeichnis	867
-----------------------------	-----