

# Inhalt

<b>1 Die Entwicklungsumgebung.....</b>	<b>1</b>
1.1 Visuelle Programmierung: Ein erstes kleines Programm.....	2
1.2 Erste Schritte in C++ .....	6
1.3 Der Quelltexteditor .....	9
1.4 Projekte, Projektdateien und Projektoptionen .....	12
1.5 Einige Tipps zur Arbeit mit Projekten.....	15
1.6 Die Online-Hilfe.....	16
1.7 Das lokale Menü.....	17
1.8 Die Symbolleiste.....	18
1.9 Programmierhilfen .....	19
1.10 Packages und eigenständig ausführbare Programme.....	20
<b>2 Die Komponentenpalette.....</b>	<b>23</b>
2.1 Die Online-Hilfe zu den Komponenten .....	23
2.2 Namen.....	27
2.3 Labels und Datentypen .....	30
2.4 Funktionen, Methoden und die Komponente <i>TEdit</i> .....	35
2.5 Memos, ListBoxen, ComboBoxen und die Klasse <i>TStrings</i> .....	40
2.6 Buttons und Ereignisse .....	44
2.7 CheckBoxen, RadioButtons und einfache <i>if</i> -Abfragen.....	47
2.8 Die Container GroupBox, Panel und RadioGroup .....	49
2.9 ScrollBar .....	52
2.10 Hauptmenüs und Popup-Menüs.....	55
2.11 Die Komponenten der Seite Dialoge .....	58
2.12 Der Aufruf von eigenen Formularen und modale Fenster ⊕ .....	61
2.13 Die Komponenten der Seite „Zusätzlich“ ⊕ .....	66
2.14 Einige Komponenten der Seiten Win32 und Win 3.1 .....	72
2.14.1 Mehrseitige Dialoge ⊕ .....	73
2.14.2 ImageList und ListView ⊕ .....	74

⊕ Siehe den Hinweis auf Seite xx

2.14.3	Komponenten zur Anzeige hierarchischer Datenstrukturen ⊕	76
2.14.4	Komponenten zur Größenanpassung anderer Komponenten ⊕	78
2.14.5	Formatierte Texte	79
2.14.6	Statusanzeigen ⊕	80
2.14.7	Weitere Komponenten ⊕	81
2.15	Einige Komponenten der Seite System ⊕	84
2.16	ActiveX-Komponenten ⊕	87
2.17	Vordefinierte DialogBoxen ⊕	92

### 3 Elementare Datentypen und Anweisungen ..... 95

3.1	Windows-Programme und Units	95
3.2	Syntaxregeln	97
3.3	Variablen und Bezeichner	100
3.4	Ganzzahldatentypen	103
3.4.1	Die interne Darstellung von Ganzzahlwerten	106
3.4.2	Der Datentyp von Ganzzahlliteralen	109
3.4.3	Zuweisungen und Standardkonversionen bei Ganzzahlausdrücken	111
3.4.4	Operatoren und die „üblichen arithmetischen Konversionen“	114
3.4.5	Der Datentyp <i>bool</i>	120
3.4.6	Die <i>char</i> -Datentypen und der ASCII- und ANSI-Zeichensatz	125
3.4.7	Der Datentyp <i>__int64</i>	132
3.5	Kontrollstrukturen und Funktionen	132
3.5.1	Die <i>if</i> - und die Verbundanweisung	133
3.5.2	Wiederholungsanweisungen	134
3.5.3	Funktionen und der Datentyp <i>void</i>	135
3.6	Der integrierte Debugger	140
3.7	Gleitkommadatentypen	146
3.7.1	Die interne Darstellung von Gleitkommawerten	147
3.7.2	Der Datentyp von Gleitkommaliteralen	150
3.7.3	Implizite Konversionen	151
3.7.4	Mathematische Funktionen	157
3.8	Syntaxregeln für einfache Deklarationen	162
3.9	Zeiger, Strings und dynamisch erzeugte Variablen	164
3.9.1	Die Definition von Zeigern	165
3.9.2	Der Adressoperator, Zuweisungen und generische Zeiger	167
3.9.3	Dynamisch erzeugte Variablen: <i>new</i> und <i>delete</i>	171
3.9.4	Zeiger als Parameter und die Zeigertypen der Windows-API	175
3.9.5	Zeigerarithmetik	177
3.9.6	Nullterminierte Strings und Zeiger auf <i>char</i>	178
3.9.7	Zeiger auf Zeiger auf Zeiger auf	184
3.9.8	Referenztypen und Referenzparameter	185
3.10	Konstanten	189
3.11	Deklarationen mit <i>typedef</i> und <i>typeid</i> -Ausdrücke	193
3.12	Aufzählungstypen	195

⊕ Siehe den Hinweis auf Seite xx

3.13	Kommentare und interne Programmdokumentation .....	200
3.14	Präprozessoranweisungen .....	204
3.14.1	Die <i>include</i> -Anweisung.....	204
3.14.2	Makros ⊕.....	206
3.14.3	Bedingte Kompilation ⊕ .....	209
3.14.4	Pragmas ⊕.....	214
<b>4</b>	<b>Strukturierte Datentypen und vordefinierte Klassen.....</b>	<b>217</b>
4.1	Die Stringklassen <i>AnsiString</i> und <i>string</i> .....	217
4.1.1	Die Verwendung von Elementfunktionen vordefinierter Klassen.....	219
4.1.2	Gemeinsamkeiten und Unterschiede der Stringklassen.....	221
4.1.3	Funktionen für die Klasse <i>AnsiString</i> .....	225
4.1.4	Einige Elementfunktionen der Klasse <i>string</i> .....	230
4.2	Arrays und Container .....	235
4.2.1	Eindimensionale Arrays.....	235
4.2.2	Arrays als Container .....	245
4.2.3	Mehrdimensionale Arrays.....	252
4.2.4	Arrays, Zeiger und Zeigerarithmetik .....	256
4.2.5	Arrays als Funktionsparameter.....	259
4.2.6	Dynamisch erzeugte Arrays .....	263
4.2.7	Array-Eigenschaften der VCL.....	267
4.3	Sequenzielle Container der Standardbibliothek .....	268
4.3.1	Die Container-Klasse <i>vector</i> .....	269
4.3.2	Algorithmen der Standardbibliothek .....	275
4.3.3	Die Container-Klassen <i>list</i> und <i>deque</i> .....	277
4.3.4	Die Container-Adapter <i>stack</i> , <i>queue</i> und <i>priority_queue</i> .....	281
4.3.5	Container mit Zeigern.....	282
4.3.6	Die Container-Klasse <i>bitset</i> ⊕.....	283
4.4	Strukturen und Klassen.....	285
4.4.1	Mit <i>struct</i> definierte Klassen .....	285
4.4.2	Verkettete Listen.....	295
4.4.3	„Memory leaks“ und die Überwachung des Heap ⊕.....	306
4.4.4	Datenstrukturen mit generischen Zeigern ⊕.....	308
4.4.5	Mit <i>union</i> definierte Klassen ⊕ .....	309
4.4.6	Die Datentypen <i>TVarRec</i> und <i>Variant</i> ⊕ .....	313
4.4.7	Bitfelder ⊕.....	315
4.5	Einige Klassen der VCL .....	316
4.5.1	Ein wenig Grafik: <i>TCanvas</i> , <i>TImage</i> und <i>TPrinter</i> ⊕ .....	317
4.5.2	Die Steuerung von MS-Office: Word-Dokumente erzeugen ⊕.....	327
4.5.3	Internet-Komponenten ⊕ .....	329

⊕ Siehe den Hinweis auf Seite xx

4.5.4	Visuell gestaltete Ausdrücke mit <i>QuickReport</i> ⊕.....	334
4.5.5	Kaufmännische Rechnungen: Die Klassen <i>Currency</i> und <i>BCD</i> ⊕ ...	337
4.5.6	Klassen und Funktionen zu Uhrzeit und Kalenderdatum ⊕ .....	339
4.5.7	Die Klasse <i>Set</i> ⊕ .....	343
4.6	Dateien .....	346
4.6.1	Stream-Variablen, ihre Verbindung mit Dateien und ihr Zustand....	346
4.6.2	Fehler und der Zustand von Stream-Variablen .....	350
4.6.3	Lesen und Schreiben von Binärdaten mit <i>read</i> und <i>write</i> .....	352
4.6.4	Lesen und Schreiben von Daten mit den Operatoren << und >>.....	360
4.6.5	Manipulatoren und Funktionen zur Formatierung von Texten.....	369
4.6.6	Textbildschirm-Anwendungen .....	372
4.6.7	Stringstreams .....	373
4.6.8	Dateibearbeitung im Direktzugriff ⊕.....	375
4.6.9	Sortieren, Mischen und Gruppenverarbeitung ⊕ .....	379
4.6.10	C-Funktionen zur Dateibearbeitung ⊕.....	386
4.7	Assoziative Container .....	389
4.7.1	Die Container <i>set</i> und <i>multiset</i> .....	389
4.7.2	Die Container <i>map</i> und <i>multimap</i> .....	391
4.7.3	Iteratoren der assoziativen Container .....	392
4.8	Die numerischen Klassen der Standardbibliothek .....	396
4.8.1	Komplexe Zahlen ⊕.....	396
4.8.2	Valarrays und Slices ⊕ .....	400
<b>5</b>	<b>Anweisungen und Ausdrücke.....</b>	<b>407</b>
5.1	Die Ausdrucksanweisung .....	408
5.2	Ausdrücke .....	410
5.2.1	Primäre Ausdrücke ⊕ .....	411
5.2.2	Postfix-Ausdrücke ⊕.....	414
5.2.3	Unäre Ausdrücke ⊕ .....	416
5.2.4	Typkonversionen in Typecast-Schreibweise ⊕.....	419
5.2.5	Zeiger auf Klasselemente ⊕.....	420
5.2.6	Multiplikative Operatoren ⊕.....	420
5.2.7	Additive Operatoren ⊕.....	420
5.2.8	Shift-Operatoren ⊕ .....	421
5.2.9	Vergleichsoperatoren ⊕ .....	422
5.2.10	Gleichheitsoperatoren ⊕ .....	423
5.2.11	Bitweise Operatoren ⊕.....	423
5.2.12	Logische Operatoren ⊕ .....	424
5.2.13	Der Bedingungsoperator ⊕.....	425
5.2.14	Konstante Ausdrücke ⊕ .....	427
5.2.15	Zuweisungsoperatoren.....	427
5.2.16	Der Komma-Operator ⊕ .....	428
5.2.17	L-Werte und R-Werte ⊕.....	429

⊕ Siehe den Hinweis auf Seite xx

5.2.18 Die Priorität und Assoziativität der Operatoren $\oplus$ .....	430
5.2.19 Alternative Zeichenfolgen $\oplus$ .....	432
5.2.20 Explizite Typkonversionen $\oplus$ .....	434
5.3 Ein wenig Programmierlogik: Symbolische Ausführung .....	441
5.4 Die Deklarationsanweisung .....	449
5.5 Die Verbundanweisung und die Blockstruktur von C++ .....	450
5.6 Lebensdauer und Speicherklassenspezifizierer .....	454
5.7 Auswahlanweisungen .....	457
5.7.1 Die <i>if</i> -Anweisung .....	458
5.7.2 Die <i>switch</i> -Anweisung .....	469
5.7.3 Ein wenig Programmierlogik für Auswahlanweisungen .....	472
5.8 Wiederholungsanweisungen .....	480
5.8.1 Die <i>while</i> -Anweisung .....	481
5.8.2 Die <i>do</i> -Anweisung .....	482
5.8.3 Die <i>for</i> -Anweisung .....	483
5.8.4 Endlosschleifen, Abbruchbedingungen und Windows .....	487
5.8.5 Ein wenig Programmierlogik für Schleifen .....	493
5.9 Die Sprunganweisungen <i>goto</i> , <i>break</i> und <i>continue</i> .....	499
5.10 Exception-Handling .....	504
5.10.1 Die <i>try</i> -Anweisung .....	505
5.10.2 Exception-Handler und Exceptions der Standardbibliothek .....	509
5.10.3 Vordefinierte Exceptions der VCL .....	513
5.10.4 Der Programmablauf bei Exceptions .....	516
5.10.5 Das vordefinierte Exception-Handling der VCL .....	518
5.10.6 <i>throw</i> -Ausdrücke und selbst definierte Exceptions .....	519
5.10.7 Fehler, Exceptions und die Korrektheit von Programmen .....	526
5.10.8 Die Freigabe von Ressourcen bei Exceptions .....	529
5.10.9 Exceptions in <code>&lt;math.h&gt;</code> und die Funktion <code>_matherr</code> $\oplus$ .....	536
5.10.10 Die Klasse <code>auto_ptr</code> $\oplus$ .....	537
5.10.11 Exception-Spezifikationen .....	539
5.10.12 Die Funktion <code>terminate</code> $\oplus$ .....	540
5.10.13 Das Win32-Exception-Handling mit <code>try__except</code> $\oplus$ .....	541
<b>6 Funktionen .....</b>	<b>543</b>
6.1 Die Definition und der Aufruf von Funktionen .....	544
6.2 Die Verwaltung von Funktionsaufrufen über den Stack .....	548
6.3 Funktionen mit Parametern .....	550
6.3.1 Werteparameter .....	552
6.3.2 „Call by reference“ mit Referenzparametern .....	554
6.3.3 Zeiger als Parameter .....	557
6.3.4 Konstante Parameter .....	559
6.3.5 Seiteneffekte und die Reihenfolge von Auswertungen .....	562
6.3.6 Default-Argumente .....	566
6.3.7 Der Datentyp einer Funktion .....	568

$\oplus$  Siehe den Hinweis auf Seite xx

6.3.8 Zeiger auf Funktionen.....	570
6.3.9 Unspezifizierte Anzahl von Argumenten ⊕.....	579
6.3.10 Die Funktionen <i>main</i> bzw. <i>WinMain</i> und ihre Parameter.....	580
6.3.11 Der Aufruf von Funktionen aus Delphi im C++Builder ⊕.....	582
6.3.12 Traditionelle K&R-Funktionsdefinitionen ⊕.....	584
6.3.13 Aufrufkonventionen ⊕.....	585
6.4 Schrittweise Verfeinerung als Entwurfstechnik.....	585
6.5 Etwas Programmierlogik und -stil für Funktionen.....	589
6.6 Rekursion.....	595
6.6.1 Quicksort.....	600
6.6.2 Ein rekursiv absteigender Parser.....	605
6.6.3 Rekursiv definierte Kurven ⊕.....	608
6.6.4 Indirekte Rekursion ⊕.....	610
6.6.5 Rekursive Datenstrukturen und binäre Suchbäume.....	614
6.6.6 Rekursive Datenstrukturen in der Standardbibliothek von C++.....	619
6.7 Inline-Funktionen.....	622
6.8 Überladene Funktionen.....	624
6.9 Überladene Operatoren mit globalen Operatorfunktionen.....	633
6.9.1 Globale Operatorfunktionen.....	634
6.9.2 Die Inkrement- und Dekrementoperatoren.....	636
6.9.3 Referenzen als Funktionswerte.....	636
6.9.4 Die Ein- und Ausgabe von selbst definierten Datentypen.....	639
<b>7 Modulare Programmierung und Namensbereiche.....</b>	<b>643</b>
7.1 Separate Kompilation und statische Bibliotheken.....	644
7.1.1 Projekte im C++Builder ⊕.....	644
7.1.2 Bindung ⊕.....	645
7.1.3 Deklarationen und Definitionen ⊕.....	647
7.1.4 Die „One Definition Rule“ ⊕.....	649
7.1.5 Header-Dateien und Bibliotheken ⊕.....	651
7.1.6 Der Aufruf von in C geschriebenen Funktionen ⊕.....	653
7.2 Dynamic Link Libraries (DLLs).....	657
7.2.1 DLLs erzeugen ⊕.....	657
7.2.2 Implizit geladene DLLs ⊕.....	659
7.2.3 Explizit geladene DLLs ⊕.....	660
7.2.4 Hilfsprogramme zur Identifizierung von Funktionen in DLLs ⊕.....	662
7.2.5 DLL-Funktionen mit visuell gestalteten Komponenten ⊕.....	663
7.2.6 Projektgruppen ⊕.....	664
7.2.7 Batch-Dateien ⊕.....	666

⊕ Siehe den Hinweis auf Seite xx

7.3 Namensbereiche .....	667
7.3.1 Die Definition von benannten Namensbereichen .....	669
7.3.2 Die Verwendung von Namen aus Namensbereichen .....	671
7.3.3 Aliasnamen für Namensbereiche .....	675
7.3.4 Unbenannte Namensbereiche .....	675
7.3.5 Module und das Geheimnisprinzip .....	677
<b>8 Objektorientierte Programmierung .....</b>	<b>681</b>
8.1 Klassen .....	682
8.1.1 Datenelemente und Elementfunktionen .....	682
8.1.2 Der Gültigkeitsbereich von Klassenelementen .....	686
8.1.3 Objekte und die Zugriffsrechte <i>private</i> und <i>public</i> .....	688
8.1.4 Der Aufruf von Elementfunktionen und der <i>this</i> -Zeiger .....	694
8.1.5 Konstruktoren und Destruktoren .....	696
8.1.6 OO Analyse und Design: Der Entwurf von Klassen .....	709
8.1.7 Ein wenig Programmierlogik: Klasseninvarianten und Korrektheit .....	714
8.1.8 UML-Diagramme für Klassen und Objekte .....	720
8.2 Klassen als Datentypen .....	722
8.2.1 Der Standardkonstruktor .....	723
8.2.2 Objekte als Klassenelemente und Elementinitialisierer .....	725
8.2.3 <i>friend</i> -Funktionen und -Klassen .....	731
8.2.4 Überladene Operatoren als Elementfunktionen .....	735
8.2.5 Der Copy-Konstruktor .....	743
8.2.6 Der Zuweisungsoperator = für Klassen .....	751
8.2.7 Benutzerdefinierte Konversionen .....	760
8.2.8 Explizite Konstruktoren $\oplus$ .....	764
8.2.9 Statische Klassenelemente .....	766
8.2.10 Konstante Klassenelemente und Objekte .....	768
8.2.11 Weitere Deklarationsmöglichkeiten in Klassen $\oplus$ .....	770
8.2.12 Klassen und Header-Dateien .....	771
8.2.13 Überladene Operatoren für <i>new</i> und <i>delete</i> $\oplus$ .....	773
8.3 Vererbung .....	776
8.3.1 Die Elemente von abgeleiteten Klassen .....	776
8.3.2 Zugriffsrechte auf die Elemente von Basisklassen .....	779
8.3.3 Die Bedeutung von Elementnamen in einer Klassenhierarchie .....	781
8.3.4 <i>using</i> -Deklarationen in abgeleiteten Klassen $\oplus$ .....	783
8.3.5 Konstruktoren, Destruktoren und implizit erzeugte Funktionen .....	784
8.3.6 Vererbung bei Formularen im C++Builder .....	790
8.3.7 OO Design: <i>public</i> Vererbung und „ist ein“-Beziehungen .....	791
8.3.8 OO Design: Komposition und „hat ein“-Beziehungen .....	797
8.3.9 Konversionen zwischen <i>public</i> abgeleiteten Klassen .....	799
8.3.10 <i>protected</i> und <i>private</i> abgeleitete Klassen $\oplus$ .....	802
8.3.11 Mehrfachvererbung und virtuelle Basisklassen .....	806

$\oplus$  Siehe den Hinweis auf Seite xx

8.4 Virtuelle Funktionen, späte Bindung und Polymorphie .....	812
8.4.1 Der statische und der dynamische Datentyp .....	813
8.4.2 Virtuelle Funktionen .....	814
8.4.3 Die interne Realisierung von virtuellen Funktionen: <i>vptr</i> und <i>vtbl</i> .....	825
8.4.4 OO-Design: Der Einsatzbereich von virtuellen Funktionen .....	831
8.4.5 Komposition und <i>private</i> Mehrfachvererbung $\oplus$ .....	832
8.4.6 Virtuelle Konstruktoren und Destruktoren .....	833
8.4.7 Virtuelle Funktionen in Konstruktoren und Destruktoren $\oplus$ .....	836
8.4.8 Virtuelle Funktionen und Erweiterbarkeit .....	837
8.4.9 Rein virtuelle Funktionen und abstrakte Klassen .....	840
8.4.10 OO-Design: Virtuelle Funktionen und abstrakte Basisklassen .....	843
8.4.11 Protokollklassen und Programmgerüste .....	848
8.4.12 Muster (Patterns) .....	850
8.4.13 UML-Diagramme für Vererbung und Komposition .....	855
8.4.14 Zeiger auf Klassenelemente $\oplus$ .....	857
8.5 Laufzeit-Typinformationen .....	862
8.5.1 Typinformationen mit dem Operator <i>typeid</i> $\oplus$ .....	862
8.5.2 Typkonversionen mit <i>dynamic_cast</i> $\oplus$ .....	866
8.5.3 Anwendungen von Laufzeit-Typinformationen $\oplus$ .....	870
8.5.4 <i>static_cast</i> mit Klassen $\oplus$ .....	873
8.5.5 Laufzeit-Typinformationen für die Klassen der VCL $\oplus$ .....	875
<b>9 Die Bibliothek der visuellen Komponenten (VCL) .....</b>	<b>879</b>
9.1 Besonderheiten der VCL .....	880
9.2 Visuelle Programmierung und Properties (Eigenschaften) .....	884
9.2.1 Lesen und Schreiben von Eigenschaften .....	884
9.2.2 Array-Properties .....	887
9.2.3 Indexangaben .....	889
9.2.4 Speicherangaben .....	890
9.2.5 Überschriebene Eigenschaften .....	891
9.3 Die Klassenhierarchie der VCL .....	891
9.4 Selbst definierte Komponenten und ihre Ereignisse .....	899
9.5 MDI-Programme .....	904
9.6 Klassenreferenztypen und virtuelle Konstruktoren .....	907
9.7 Botschaften (Messages) .....	912
9.7.1 Die Message Queue und die Window-Prozedur .....	912
9.7.2 Botschaften für eine Anwendung .....	914
9.7.3 Die Behandlung von Botschaften in der VCL .....	915
9.7.4 Selbst definierte Message-Handler für Windows-Steuerelemente .....	921
9.7.5 Botschaften versenden .....	923
9.8 Die Erweiterung der Komponentenpalette .....	927

$\oplus$  Siehe den Hinweis auf Seite xx

<b>10 Templates und die STL.....</b>	<b>935</b>
10.1 Generische Funktionen: Funktions-Templates.....	936
10.1.1 Die Deklaration von Funktions-Templates mit Typ-Parametern ....	937
10.1.2 Spezialisierungen von Funktions-Templates.....	939
10.1.3 Funktions-Templates mit Nicht-Typ-Parametern $\oplus$ .....	944
10.1.4 Explizit instanziierte Funktions-Templates $\oplus$ .....	946
10.1.5 Explizit spezialisierte und überladene Templates .....	946
10.1.6 Rekursive Funktions-Templates .....	950
10.2 Generische Klassen: Klassen-Templates .....	953
10.2.1 Die Deklaration von Klassen-Templates mit Typ-Parametern.....	953
10.2.2 Spezialisierungen von Klassen-Templates .....	954
10.2.3 Templates mit Nicht-Typ-Parametern $\oplus$ .....	962
10.2.4 Explizit instanziierte Klassen-Templates $\oplus$ .....	963
10.2.5 Partielle und vollständige Spezialisierungen $\oplus$ .....	964
10.2.6 Elemente und <i>friend</i> -Funktionen von Klassen-Templates $\oplus$ .....	970
10.2.7 Ableitungen von Templates $\oplus$ .....	974
10.2.8 Exportierte Templates .....	975
10.2.9 UML-Diagramme für parametrisierte Klassen $\oplus$ .....	976
10.3 Funktionsobjekte in der STL.....	980
10.3.1 Der Aufrufoperator ().....	980
10.3.2 Prädikate und arithmetische Funktionsobjekte.....	982
10.3.3 Binder, Funktionsadapter und Negatoren .....	986
10.4 Iteratoren und die STL-Algorithmen.....	992
10.4.1 Die verschiedenen Arten von Iteratoren .....	993
10.4.2 Umkehriteratoren.....	995
10.4.3 Einfügefunktionen und Einfügeiteratoren.....	996
10.4.4 Stream-Iteratoren.....	998
10.4.5 Container-Konstruktoren mit Iteratoren .....	999
10.4.6 STL-Algorithmen für alle Elemente eines Containers .....	1000
10.5 Die Algorithmen der STL .....	1002
10.5.1 Lineares Suchen.....	1003
10.5.2 Zählen .....	1005
10.5.3 Der Vergleich von Bereichen .....	1005
10.5.4 Suche nach Teilfolgen.....	1007
10.5.5 Minimum und Maximum .....	1008
10.5.6 Elemente vertauschen .....	1010
10.5.7 Kopieren von Bereichen.....	1011
10.5.8 Elemente transformieren und ersetzen.....	1012
10.5.9 Elementen in einem Bereich Werte zuweisen .....	1014
10.5.10 Elemente entfernen .....	1014
10.5.11 Die Reihenfolge von Elementen vertauschen .....	1016
10.5.12 Permutationen.....	1017
10.5.13 Partitionen .....	1018
10.5.14 Bereiche sortieren .....	1019
10.5.15 Binäres Suchen in sortierten Bereichen .....	1021

$\oplus$  Siehe den Hinweis auf Seite xx

10.5.16 Mischen von sortierten Bereichen .....	1022
10.5.17 Mengenoperationen auf sortierten Bereichen.....	1023
10.5.18 Heap-Operationen .....	1024
10.5.19 Verallgemeinerte numerische Operationen .....	1025

## **11 Verschiedenes..... 1027**

11.1 3D-Grafik mit OpenGL.....	1028
11.1.1 Initialisierungen.....	1028
11.1.2 Grafische Grundelemente: Primitive .....	1031
11.1.3 Modelltransformationen .....	1036
11.1.4 Vordefinierte Körper.....	1039
11.1.5 Lokale Transformationen .....	1041
11.1.6 Beleuchtungseffekte .....	1044
11.1.7 Texturen .....	1046
11.2 Win32-Funktionen zur Dateibearbeitung .....	1051
11.2.1 Elementare Funktionen .....	1051
11.2.2 Der gleichzeitige Zugriff von mehreren Anwendern auf e. Datei ..	1055
11.2.3 Record-Locking .....	1056
11.2.4 VCL-Funktionen zur Dateibearbeitung und <i>TFileStream</i> .....	1057
11.3 Datenübertragung über die serielle Schnittstelle.....	1059
11.3.1 Grundbegriffe .....	1060
11.3.2 Standards für die serielle Schnittstelle: RS-232C bzw. V.24.....	1061
11.3.3 Win32-Funktionen zur seriellen Kommunikation.....	1063
11.4 Datenbank-Komponenten der VCL.....	1067
11.4.1 Tabellen und die Komponente <i>TTable</i> .....	1067
11.4.2 Die Anzeige von Tabellen mit der Komponente <i>DBGrid</i> .....	1072
11.4.3 Indizierte Tabellen .....	1075
11.4.4 Datenbanken mit mehreren Tabellen.....	1078
11.4.5 SQL-Abfragen .....	1080
11.4.6 Transaktionen und Cached Updates .....	1082
11.4.7 Die BDE am Beispiel von ODBC und MS Access Datenbanken ..	1084
11.4.8 Visuell gestaltete Datenbank-Ausdrucke mit <i>QuickReport</i> .....	1086

## **Literaturverzeichnis..... 1091**

## **Verschiedenes..... 1095**

⊕ Angesichts des Umfangs dieses Buches habe ich einige Abschnitte mit dem Zeichen ⊕ in der Überschrift als „weniger wichtig“ gekennzeichnet. Damit wollte ich dem Anfänger nur eine kleine Orientierung durch die Fülle des Stoffes geben. Eine solche Kennzeichnung soll aber überhaupt nicht bedeuten, dass dieser Teil unwichtig ist. Es kann sogar gut sein, dass gerade Sie diese Inhalte laufend benötigen.

⊕ Siehe den Hinweis auf Seite xx