

Inhalt

Vorwort zur 4. Auflage	25
------------------------------	----

1 Allgemeine Einführung in .NET

1.1	Warum .NET?	27
1.1.1	Ein paar Worte zu diesem Buch	29
1.1.2	Die Beispielpprogramme	31
1.2	.NET unter die Lupe genommen	32
1.2.1	.NET – Ein paar allgemeine Eigenschaften	33
1.2.2	Das Sprachenkonzept	35
1.2.3	Die »Common Language Specification« (CLS)	36
1.2.4	Das »Common Type System« (CTS)	37
1.2.5	Das .NET Framework	38
1.2.6	Die »Common Language Runtime« (CLR)	39
1.2.7	Die .NET-Klassenbibliothek	40
1.2.8	Das Konzept der Namespaces	41
1.3	Assemblies	42
1.3.1	Die Metadaten	43
1.3.2	Das Manifest	44
1.4	Die Entwicklungsumgebung	44
1.4.1	Editionen von Visual Studio 2008	44
1.4.2	Hard- und Softwareanforderungen	45
1.4.3	Die Installation	45
1.4.4	Die Entwicklungsumgebung von Visual Studio 2008	47

2 Grundlagen der Sprache C#

2.1	Konsolenanwendungen	53
2.1.1	Ein erstes Konsolenprogramm	53
2.2	Grundlagen der C#-Syntax	56
2.2.1	Kennzeichnen, dass eine Anweisung abgeschlossen ist	56
2.2.2	Anweisungs- und Gliederungsblöcke	57
2.2.3	Kommentare	58
2.2.4	Die Groß- und Kleinschreibung	59
2.2.5	Die Struktur einer Konsolenanwendung	60
2.3	Variablen und Datentypen	62
2.3.1	Variablendeklaration	62
2.3.2	Der Variablenbezeichner	64
2.3.3	Der Zugriff auf eine Variable	65
2.3.4	Ein- und Ausgabemethoden der Klasse »Console«	65
2.3.5	Die einfachen Datentypen	71
2.3.6	Typkonvertierung	81
2.4	Operatoren	90
2.4.1	Arithmetische Operatoren	90
2.4.2	Vergleichsoperatoren	93

2.4.3	Logische Operatoren	94
2.4.4	Bitweise Operatoren	97
2.4.5	Zuweisungsoperatoren	99
2.4.6	Stringverkettung	100
2.4.7	Sonstige Operatoren	100
2.4.8	Operator-Vorrangregeln	101
2.5	Datenfelder (Arrays)	101
2.5.1	Die Deklaration und Initialisierung eines Arrays	101
2.5.2	Der Zugriff auf die Array-Elemente	103
2.5.3	Speicherabbild eines Arrays	104
2.5.4	Mehrdimensionale Arrays	106
2.5.5	Festlegen der Array-Größe zur Laufzeit	108
2.5.6	Bestimmung der Array-Obergrenze	109
2.5.7	Die Gesamtanzahl der Array-Elemente	110
2.5.8	Verzweigte Arrays	111
2.6	Kontrollstrukturen	112
2.6.1	Die »if«-Anweisung	112
2.6.2	Das »switch«-Statement	118
2.7	Programmschleifen	123
2.7.1	Die »for«-Schleife	123
2.7.2	Die »foreach«-Schleife	134
2.7.3	Die »do«- und die »while«-Schleife	135

3 Klassendesign und Vererbung **141**

3.1	Einführung in die Objektorientierung	141
3.1.1	Vorteile der objektorientierten Programmierung	143
3.2	Die Klassendefinition	144
3.2.1	Die Deklaration von Objektvariablen	145
3.2.2	Zugriffsmodifizierer einer Klasse	146
3.2.3	Splitten einer Klassendefinition mit »partial«	146
3.3	Referenz- und Wertetypen	147
3.4	Die Eigenschaften eines Objekts (Felder)	149
3.4.1	Vereinfachte Objektinstanziierung	151
3.5	Arbeiten mit Objektreferenzen	151
3.5.1	Prüfen auf Initialisierung	151
3.5.2	Mehrere Referenzen auf ein Objekt	152
3.5.3	Referenzvergleiche	153
3.6	Objektmethoden	157
3.6.1	Der Aufruf einer Methode	159
3.6.2	Methoden mit Parameterliste	159
3.6.3	Rückgabewert einer Methode	160
3.6.4	Variablen in einer Methode	163
3.6.5	Zugriffsmodifizierer einer Methode	165
3.6.6	Besondere Aspekte einer Parameterliste	165
3.6.7	Referenz- und Wertparameter	167
3.6.8	Methodenüberladung	176
3.6.9	Aufruf überladener Methoden mit impliziter Konvertierung	178

3.7	Objekteigenschaften	179
3.7.1	Datenkapselung mit Eigenschaftsmethoden sicherstellen	179
3.7.2	Die Ergänzung der Klasse »Circle«	182
3.7.3	Lese- und schreibgeschützte Eigenschaften	183
3.7.4	Sichtbarkeit der Accessoren »get« und »set«	184
3.7.5	Automatisch implementierte Eigenschaften	184
3.7.6	Konstanten in einer Klasse	185
3.7.7	Methode oder Eigenschaft?	186
3.7.8	Die Trennung von Daten und Code	188
3.7.9	Der Zugriff auf private Daten	188
3.8	Konstruktoren	189
3.8.1	Die Konstruktoren in der Klasse »Circle«	190
3.8.2	Die Konstruktoraufrufe	192
3.8.3	Definition von Konstruktoren	192
3.8.4	»internal«-Konstruktoren	193
3.8.5	»private«-Konstruktoren	193
3.8.6	Konstruktorverkettung	194
3.9	Der Destruktor	195
3.9.1	Das Zerstören von Objekten	195
3.9.2	Der Garbage Collector	195
3.9.3	Die Bereitstellung eines Destruktors	196
3.9.4	Das Zerstören eines Objekts	196
3.9.5	Die »Dispose«-Methode	196
3.9.6	Der Garbage Collector in Aktion	200
3.10	Statische Klassenkomponenten	201
3.10.1	Begrifflichkeiten	201
3.10.2	Die Realisierung eines Objektzählers	202
3.10.3	Zugriff auf statische Komponenten	203
3.10.4	Statische Klassenvariable in der Klasse »Circle«	204
3.10.5	Klassenspezifische Methoden	205
3.10.6	Statische Klasseninitialisierer	209
3.10.7	Statische Klassen	210
3.10.8	Stand der Klasse »Circle«	211
3.11	Die Vererbung	213
3.11.1	Basisklassen und abgeleitete Klassen	213
3.11.2	Die Ableitung einer Klasse	215
3.11.3	Klassen, die nicht abgeleitet werden können	216
3.11.4	Konstruktoren in abgeleiteten Klassen	216
3.11.5	Der Zugriffsmodifizierer »protected«	217
3.11.6	Die Konstruktorverkettung in der Vererbung	218
3.11.7	Destruktor-Verkettung in der Vererbung	223
3.11.8	Die Methoden in einer abgeleiteten Klasse	224
3.11.9	Überladen einer Basisklassenmethode	227
3.11.10	Typumwandlung von Objektvariablen	228
3.12	Abstrakte Klassen und Methoden	236
3.12.1	Problembeschreibung	236
3.12.2	Abstrakte Definitionen	236

3.13	Polymorphismus	238
3.13.1	Virtuelle Methoden	240
3.13.2	Inhomogene Mengen	241
3.13.3	Verdecken und Überschreiben geerbter Methoden	244
3.13.4	Versiegelte Methoden	245
3.13.5	Das Projekt »CircleApplication« ergänzen	246
3.14	Hat-eine-Beziehungen (Aggregation)	249
3.14.1	Innere Klassen	254
3.15	Namensräume (Namespaces)	258
3.15.1	Zugriff auf Namespaces	259
3.15.2	Die »using«-Direktive	260
3.15.3	Vermeiden von Mehrdeutigkeiten	261
3.15.4	Aufrufe mit dem »::«-Operator umleiten	262
3.15.5	Namespaces festlegen	263

4 Weitere NET-Datentypen 265

4.1	Interfaces (Schnittstellen)	265
4.1.1	Schnittstellendefinition	265
4.1.2	Schnittstellenimplementierung	266
4.1.3	Typumwandlung mit dem »as«-Operator	273
4.1.4	Abstrakte Klassen vs. Schnittstellen	274
4.2	Strukturen – eine Sonderform der Klassen	280
4.2.1	Die Definition einer Struktur	280
4.2.2	Initialisieren einer Struktur	281
4.2.3	Weitere Merkmale einer Struktur	282
4.2.4	Verschachtelte Strukturen	284
4.2.5	Änderung der Klasse »Circle«	284
4.2.6	Zusammenfassung aller Änderungen am Projekt »CircleApplication«	286
4.3	Enumerationen (Aufzählungen)	286
4.3.1	Wertzuweisung an enum-Mitglieder	287
4.3.2	Alle Mitglieder einer Aufzählung durchlaufen	288
4.4	Delegates	289
4.4.1	Einführung in das Prinzip der Delegates	289
4.4.2	Vereinfachter Aufruf eines Delegates	293
4.4.3	Multicast-Delegates	293
4.4.4	Anonyme Methoden	297
4.5	Ereignisse eines Objekts	299
4.5.1	Ergänzung eines Ereignisses in einer Ereignisquelle	300
4.5.2	Die Behandlung eines Ereignisses im Ereignisempfänger	302
4.5.3	Wenn der Ereignisempfänger ein Ereignis nicht behandelt	304
4.5.4	Ereignisse mit Übergabeparameter	304

5 Weitere Möglichkeiten von C# 307

5.1	Operatorüberladung	307
5.1.1	Die Syntax der Operatorüberladung	307
5.1.2	Beispiel einer Operatorüberladung	308

5.1.3	Überladungsbeispiele	310
5.1.4	Benutzerdefinierte Konvertierungen	312
5.2	Indexer	317
5.2.1	Überladen von Indexern	319
5.2.2	Parameterbehaftete Eigenschaften	321
5.3	Collections (Auflistungen)	325
5.3.1	Die elementaren Schnittstellen der Auflistungsklassen	325
5.3.2	Die Schnittstelle »IList«	327
5.3.3	Die Klasse »ArrayList«	328
5.3.4	Die Elemente einer »ArrayList« sortieren	331
5.3.5	Die Schnittstelle »IDictionary«	338
5.3.6	Die Klasse »Hashtable«	339
5.3.7	Die Klassen »Queue« und »Stack«	343
5.3.8	Collection-Initialisierer	345
5.3.9	Objektauflistungen im Überblick	345
5.4	Generics – Generische Datentypen	346
5.4.1	Ein paar allgemeine Worte	346
5.4.2	Die Typproblematik am Beispiel der Klasse »Stack«	347
5.4.3	Die Lösung mit einer generischen Klasse	348
5.4.4	Typparameter mit Constraints einschränken	351
5.4.5	Generische Methoden	353
5.4.6	Generics und Vererbung	354
5.4.7	Konvertierung von Generics	355
5.4.8	Generische Delegates	356
5.4.9	Generische Klassen in der .NET-Klassenbibliothek	357
5.4.10	Zwei Beispiele mit generischen Klassen	358
5.5	Eigene Auflistungen mit »yield« durchlaufen	363
5.6	Daten durch »null« beschreiben	366
5.7	Attribute	367
5.7.1	Das »Flags«-Attribut	368
5.7.2	Anmerkungen zu den Attributen	369
5.7.3	Benutzerdefinierte Attribute	369
5.7.4	Festlegen der Assembly-Eigenschaften	375
5.8	Unsicherer (unsafe) Programmcode – Zeigertechnik in C#	376
5.8.1	Das Schlüsselwort »unsafe«	376
5.8.2	Die Deklaration von Zeigern	377
5.8.3	Die »fixed«-Anweisung	378
5.8.4	Zeigerarithmetik	379
5.8.5	Der Operator »->«	380

6 Projektmanagement und Visual Studio 2008 381

6.1	Der Projekttyp »Klassenbibliothek«	381
6.1.1	Mehrere Projekte in einer Projektmappe verwalten	382
6.1.2	Die Zugriffsmodifizierer »public« und »internal«	383
6.1.3	Friend Assemblys	383
6.1.4	Einbinden einer Klassenbibliothek	384

6.2	Assemblys	386
6.2.1	Ein Überblick über das Konzept der Assemblys	386
6.2.2	Allgemeine Beschreibung privater und globaler Assemblys	387
6.2.3	Die Struktur einer Assembly	388
6.2.4	Globale Assemblys	393
6.3	Konfigurationsdateien	402
6.3.1	Die verschiedenen Konfigurationsdateien	403
6.3.2	Die Struktur einer Anwendungskonfigurationsdatei	405
6.3.3	Eine Anwendungskonfigurationsdatei mit Visual Studio 2008 bereitstellen	408
6.3.4	Einträge der Anwendungskonfigurationsdatei auswerten	409
6.3.5	Editierbare, anwendungsbezogene Einträge mit <appSettings>	414
6.4	Versionsumleitung in einer Konfigurationsdatei	416
6.4.1	Die Herausgeberrichtliniendatei	421
6.5	Die XML-Dokumentation	422
6.5.1	Die XML-Kommentar-Tags	424
6.5.2	Generieren der XML-Dokumentationsdatei	425
6.6	Der Klassendesigner (Class Designer)	426
6.6.1	Ein typisches Klassendiagramm	427
6.6.2	Hinzufügen und Ansicht von Klassendiagrammen	428
6.6.3	Die Toolbox des Klassendesigners	429
6.6.4	Das Fenster »Klassendetails«	430
6.6.5	Klassendiagramme als Bilder exportieren	432
6.7	Refactoring	432
6.7.1	Methode extrahieren	433
6.7.2	Bezeichner umbenennen	434
6.7.3	Felder inkapseln	436
6.7.4	Schnittstelle extrahieren	437
6.7.5	Die Parameterliste beeinflussen	438
6.8	Code Snippets (Codeausschnitte)	440
6.8.1	Codeausschnitte einfügen	440
6.8.2	Die Anatomie eines Codeausschnitts	441
6.8.3	Eigene Code Snippets definieren	443

7 Fehlerbehandlung und Debugging 445

7.1	Die Behandlung von Laufzeitfehlern	445
7.1.1	Laufzeitfehler erkennen	446
7.1.2	Die Behandlung von Exceptions	448
7.1.3	Die »try...catch«-Anweisung	448
7.1.4	Behandlung mehrerer Exceptions	451
7.1.5	Die »finally«-Anweisung	453
7.1.6	Das Weiterleiten von Ausnahmen	454
7.1.7	Die Hierarchie der Exceptions	460
7.1.8	Die Reihenfolge der »catch«-Klauseln	461
7.1.9	Die Basisklasse »Exception«	463
7.1.10	Benutzerdefinierte Exceptions	463

7.2	Debuggen mit Programmcode	466
7.2.1	Die Klasse »Debug«	466
7.2.2	Die Klasse »Trace«	470
7.2.3	Ablaufverfolgung mit »TraceListener«-Objekten	470
7.2.4	Steuerung der Protokollierung mit Schaltern	475
7.2.5	Bedingte Kompilierung	478
7.3	Debuggen mit dem Visual Studio 2008	480
7.3.1	Debuggen im Haltemodus	481
7.3.2	Das »Direktfenster«	483
7.3.3	Weitere Alternativen, um Variableninhalte zu prüfen	484
7.4	Das Objekttestcenter (Object Test Bench – OTB)	486

8 LINQ 489

8.1	Was ist LINQ?	489
8.2	Neue Sprachfeatures in C# 2008	490
8.2.1	Implizit typisierte Variablen (Typinferenz)	490
8.2.2	Lambda-Ausdrücke	491
8.2.3	Erweiterungsmethoden	493
8.2.4	Anonyme Typen	497
8.2.5	C# 3.0 und LINQ-Abfragen	498
8.3	LINQ to Objects	500
8.3.1	Musterdaten	501
8.3.2	Die Abfrage-Syntax	503
8.3.3	Übersicht über die Abfrageoperatoren	504
8.3.4	Die »from«-Klausel	505
8.3.5	Der Restriktionsoperator »where«	506
8.3.6	Die Projektionsoperatoren	509
8.3.7	Sortieroperatoren	510
8.3.8	Gruppieren mit »GroupBy«	511
8.3.9	Verknüpfungen mit »Join«	513
8.3.10	Die Set-Operatoren-Familie	516
8.3.11	Die Familie der Aggregatoperatoren	517
8.3.12	Generierungsoperatoren	520
8.3.13	Quantifizierungsoperatoren	521
8.3.14	Aufteilungsoperatoren	522
8.3.15	Die Elementoperatoren	524

9 Multithreading und asynchrone Methodenaufrufe 529

9.1	Multithreading	529
9.2	Threadzustände und Prioritäten	530
9.3	Einsatz von mehreren Threads	532
9.4	Die Entwicklung einer Multithreading-Anwendung	532
9.4.1	Die Klasse »Thread«	535
9.4.2	Threadpools nutzen	550
9.5	Die Synchronisation von Threads	552
9.5.1	Unsynchronisierte Threads	552

9.5.2	Der »Monitor« zur Synchronisation	554
9.5.3	Das Synchronisationsobjekt »Mutex«	563
9.5.4	Das Attribut »MethodImpl«	565
9.6	Asynchrone Methodenaufrufe	565
9.6.1	Asynchroner Methodenaufruf	567
9.6.2	Asynchroner Aufruf mit Rückgabewerten	573
9.6.3	Eine Klasse mit asynchronen Methodenaufrufen	577

10 Arbeiten mit Dateien und Streams 581

10.1	Einführung	581
10.2	Namespaces der Ein- bzw. Ausgabe	582
10.2.1	Das Behandeln von Ausnahmen bei E/A-Operationen	583
10.3	Laufwerke, Verzeichnisse und Dateien	583
10.3.1	Die Klasse »File«	583
10.3.2	Die Klasse »FileInfo«	592
10.3.3	Die Klassen »Directory« und »DirectoryInfo«	595
10.3.4	Die Klasse »Path«	598
10.3.5	Die Klasse »DriveInfo«	600
10.3.6	Die Klasse »SpecialDirectories«	601
10.4	Die »Stream«-Klassen	602
10.4.1	Die abstrakte Klasse »Stream«	603
10.4.2	Die von »Stream« abgeleiteten Klassen im Überblick	605
10.4.3	Die Klasse »FileStream«	606
10.5	Die Klassen »TextReader« und »TextWriter«	612
10.5.1	Die Klasse »StreamWriter«	613
10.5.2	Die Klasse »StreamReader«	616
10.5.3	Die Klassen »StringWriter« und »StringReader«	619
10.6	Die Klassen »BinaryReader« und »BinaryWriter«	620
10.6.1	Komplexe binäre Dateien	623

11 Serialisierung 629

11.1	Einführung in die Serialisierung	629
11.1.1	Serialisierungsverfahren	630
11.2	Serialisierung mit »BinaryFormatter«	630
11.3	Serialisierung mit »SoapFormatter«	637
11.4	Serialisierung mit »XmlSerializer«	637
11.4.1	XML-Serialisierung mit Attributen steuern	639

12 Einige wichtige .NET-Klassen 643

12.1	Die Klasse »Object«	643
12.1.1	Der Konstruktor	643
12.1.2	Die Methoden der Klasse »Object«	643
12.2	Die Klasse »String«	651
12.2.1	Das Erzeugen eines Strings	652
12.2.2	Unveränderliche »String«-Objekte	653
12.2.3	Die Eigenschaften von »String«	654

12.2.4	Die Methoden der Klasse »String«	655
12.2.5	Zusammenfassung der Klasse »String«	665
12.3	Die Klasse »StringBuilder«	665
12.3.1	Die Kapazität eines »StringBuilder«-Objekts	666
12.3.2	Die Konstruktoren der Klasse »StringBuilder«	667
12.3.3	Die Eigenschaften der Klasse »StringBuilder«	667
12.3.4	Die Methoden der Klasse »StringBuilder«	668
12.3.5	Allgemeine Anmerkungen	670
12.4	Der Typ »DateTime«	670
12.4.1	Die Zeitspanne »Tick«	671
12.4.2	Die Konstruktoren von »DateTime«	671
12.4.3	Die Eigenschaften von »DateTime«	672
12.4.4	Die Methoden der Klasse »DateTime«	673
12.4.5	Die Klasse »TimeSpan«	677
12.5	Die Klasse »Array«	680
12.5.1	Das Erzeugen eines »Array«-Objekts	680
12.5.2	Die Eigenschaften eines »Array«-Objekts	682
12.5.3	Die Methoden der Klasse »Array«	682
12.5.4	Array-Elemente sortieren	684
12.6	Ausgabeformatierung	685
12.6.1	Formatierung mit der Methode »String.Format«	685
12.6.2	Formatierung mit der Methode »ToString«	689
12.6.3	Benutzerdefinierte Formatierung	689

13 Grundlagen zum Erstellen einer Windows-Anwendung 693

13.1	Die Entwicklungsumgebung	693
13.2	Eine erste Windows-Anwendung	694
13.3	Das Codegerüst einer Windows-Anwendung	700
13.3.1	Die Datei »Program.cs«	700
13.3.2	Die Dateien »Form1.cs« und »Form1.Designer.cs«	701
13.4	Ereignisse grafischer Windows-Komponenten	704
13.4.1	Grundlegende Anmerkungen	704
13.4.2	Ereignisse mit Ereignisdaten	706
13.4.3	Ereignishandler mit dem Visual Studio 2008 bereitstellen	707
13.4.4	Ereignisbehandlung mit den »OnXxx«-Methoden	708
13.5	Die Basisklassen einer Form	709
13.5.1	Die Klasse »Control«	710
13.5.2	Die Klasse »ScrollableControl«	710
13.5.3	Die Klasse »ContainerControl«	711
13.6	Die Eigenschaften einer Form	711
13.6.1	Den Rahmen einer Form festlegen	712
13.6.2	Formspezifisches Symbol festlegen	712
13.6.3	Die Schaltflächen in der Titelleiste	713
13.6.4	Spielereien mit »Opacity« und »TransparencyKey«	713
13.6.5	Fenster, die nicht verdeckt werden können	714
13.6.6	Den Mauszeiger verändern	714
13.6.7	Farbeinstellungen mit dem Typ »Color«	714

13.6.8	Schriftart mit dem Typ »Font«	715
13.6.9	Die Abmessungen einer Form festlegen	717
13.6.10	Die Position eines Fensters	717
13.6.11	Der Anzeigezustand eines Fensters nach dem Öffnen	719
13.6.12	Die Arbeitsfläche des Fensters (der Clientbereich)	719
13.6.13	Die Fokussierreihenfolge	719
13.6.14	Die Standardschaltflächen eines Fensters	720
13.6.15	Die Auflistung »ControlsCollection«	720
13.6.16	Zusammenfassung der Eigenschaften des Form-Objekts	721
13.7	Ereignisse einer Form	723
13.7.1	Ereignisse beim Erzeugen eines Fensters	723
13.7.2	Größenänderung einer Form	724
13.7.3	Ereignisse beim Schließen eines Fensters	724
13.8	Anwendungen mit mehreren Fenstern	725
13.8.1	Neue Forms hinzufügen	725
13.8.2	Formulare laden, anzeigen, verstecken und schließen	727
13.8.3	Mehrere Fenster verwalten	727
13.8.4	Splash-Fenster	732
13.9	Modale Dialogfenster	733
13.9.1	Layout eines Dialogfensters	734
13.9.2	Die Eigenschaft »DialogResult«	734
13.9.3	Eine Form modal öffnen	735
13.10	Meldungsfenster mit »MessageBox«	736
13.10.1	Die Methode »MessageBox.Show«	737
13.10.2	Der Rückgabewert der Meldungsfenster	739
13.11	Die Klasse »Application«	740
13.11.1	Die Datei »AssemblyInfo.cs«	740
13.11.2	Mit »Application.DoEvents« wartende Ereignisse abrufen	740
13.11.3	Zusammenfassung der Eigenschaften und Methoden	742

14 Die wichtigsten Steuerelemente 745

14.1	Gemeinsame Eigenschaften, Methoden und Ereignisse	745
14.1.1	Größe und Position	745
14.1.2	Die Sichtbarkeit und der Aktivierungszustand	745
14.1.3	Die Eigenschaft »Text«	745
14.1.4	Die Farbeigenschaften	746
14.1.5	Grafische Darstellung	746
14.1.6	Die »Modifiers«-Eigenschaft	746
14.1.7	Die Eigenschaft »Tag«	746
14.1.8	Die Größe von Steuerelementen dynamisch der Formgröße anpassen	747
14.1.9	Die »Dock«-Eigenschaft	748
14.1.10	Steuerelemente fokussieren	748
14.1.11	Maus- und Tastaturreignisse	750
14.2	Schaltflächen der Klasse »Button«	750
14.2.1	Die Rahmendarstellung einer Schaltfläche	750
14.2.2	Die Eigenschaft »FlatAppearance«	751

14.2.3	Beschriftung und Grafiken	751
14.3	Auswahlkästchen mit der Klasse »CheckBox«	752
14.3.1	Eigenschaften eines Kontrollkästchens	755
14.3.2	Checkboxes mit drei Aktivierungszuständen	755
14.4	Die Klasse »RadioButton« (Optionsschaltflächen)	756
14.4.1	Die Gruppierung der Optionsschaltflächen	756
14.4.2	Die Eigenschaften von Optionsschaltflächen	757
14.4.3	Den Zustandswechsel programmieren	757
14.5	Die »GroupBox« als übergeordneter Container	758
14.6	Texteingabefelder mit der Klasse »TextBox«	758
14.6.1	Einzeilige Eingabefelder	759
14.6.2	Mehrzeilige Eingabefelder	765
14.7	Beschriftungen mit dem Steuerelement »Label«	767
14.8	Die Anzeige eines Quickinfo-Texts	768
14.8.1	Methoden des »ToolTip«-Steuerelements	768
14.8.2	Aktivierungsdauer des »QuickInfo«-Steuerelements	769
14.8.3	Weitere Eigenschaften	770
14.9	Das »ListBox«-Steuerelement	770
14.9.1	Die Auflistung »ListBox.ObjectCollection«	771
14.9.2	Eigenschaften zur Darstellung einer Listbox	773
14.9.3	Einfach- und Mehrfachauswahl der Listenelemente	775
14.9.4	Programmgesteuerter Zugriff auf Listboxen mit Einfachauswahl	775
14.9.5	Benutzerdefiniertes Sortieren der Listenelemente	780
14.9.6	Füllen einer Listbox mit »DataSource«	783
14.10	Das Steuerelement »CheckedListBox«	785
14.11	Die »ComboBox« (Kombinationslistenfeld)	786
14.11.1	Ereignisse eines Kombinationslistenfeldes	787
14.11.2	Autovervollständigung in einer »ComboBox«	788
14.12	Standarddialoge	788
14.12.1	Die Klasse »OpenFileDialog«	789
14.12.2	Die Klasse »SaveFileDialog«	795
14.12.3	Der Dialog »FolderBrowserDialog«	801
14.12.4	Die Klasse »ColorDialog«	801
14.12.5	Die Klasse »FontDialog«	803
14.13	Menü-, Symbol- und Statusleiste sowie Kontextmenüs	804
14.13.1	Das Steuerelement »ToolStripContainer«	804
14.13.2	Bereitstellen eines Menüs	805
14.13.3	Kontextmenüs	816
14.13.4	Die Symbolleiste	817
14.13.5	Die Statusleiste	827
14.13.6	Eine Statusleiste bereitstellen	827
14.14	Bildlaufleisten mit »HScrollBar« und »VScrollBar«	828
14.15	Das »ProgressBar«-Steuerelement	831
14.16	Das »Timer«-Steuerelement	833
14.17	Das »Panel«-Steuerelement	835
14.18	Registerkarten mit »TabControl«	835
14.18.1	Die Klasse »TabControl«	835

14.19	Das »TreeView«-Steuerelement	838
14.19.1	Knotenpunkte im »TreeView« definieren	838
14.19.2	Eigenschaften des »TreeView«-Steuerelements	844
14.19.3	Die Ereignisse des »TreeView«-Steuerelements	845
14.19.4	Weitere Eigenschaften und Methoden des »TreeView«-Objekts	846
14.19.5	Eigenschaften und Methoden des »TreeNode«-Objekts	847
14.20	Die beiden »Splitter«-Steuerelemente	848
14.20.1	Das Steuerelement »Splitter«	848
14.20.2	Das Steuerelement »SplitContainer«	849
14.21	Das »ListView«-Steuerelement	850
14.21.1	Die Klassen des »ListView«-Steuerelements	850
14.21.2	Die Eigenschaften der »ListView«	851
14.21.3	Listenelemente vom Typ »ListViewItem«	854
14.21.4	Das Element »ListViewSubItem«	856
14.21.5	Der Typ »ColumnHeader«	857
14.21.6	Listenelemente Gruppen zuordnen	858
14.21.7	Sortierung der Spalten	859
14.21.8	Listenelemente ändern	861
14.21.9	Beispielanwendung	862

15 Tastatur- und Mausereignisse 867

15.1	Die Tastaturschnittstelle	867
15.1.1	Die Tastaturereignisse im Überblick	867
15.1.2	Die Ereignisse »KeyDown« und »KeyUp«	868
15.1.3	Das Ereignis »KeyPress«	874
15.1.4	Die Tastaturereignisse der Form	877
15.1.5	Senden von Tastatureingaben	878
15.2	Die Mausschnittstelle	879
15.2.1	Die Ereignisse »MouseDown«, »MouseMove« und »MouseUp«	879
15.2.2	Das Ereignis »MouseWheel«	883
15.2.3	Weitere Mausereignisse	883
15.2.4	Die »Click«-Ereignisse	884
15.2.5	Eigenschaften der Maus	884

16 MDI-Anwendungen 887

16.1	Einführung	887
16.2	Das MDI-Hauptfenster	888
16.3	Die Subfenster	889
16.4	Zugriff auf die Subfenster	890
16.4.1	Zugriff auf alle Subfenster	890
16.4.2	Zugriff auf das aktive Subfenster	891
16.4.3	Zugriff auf die Subfenster eines bestimmten Typs	891
16.5	Das Menü in einer MDI-Anwendung	892
16.5.1	Subfenster anordnen	892
16.5.2	Subfenster mit eigenen Menüs	893

16.6	Symbol- und Statusleiste des Subfensters	895
16.7	Die Liste der geöffneten untergeordneten Fenster	895
16.8	Beispiel einer MDI-Anwendung	896

17 Grafische Programmierung mit GDI+ 899

17.1	Die Namespaces der GDI+-Schnittstelle	899
17.2	Die Klasse »Graphics«	900
17.2.1	Überblick über die Klasse »Graphics«	900
17.2.2	Eine Referenz auf das »Graphics«-Objekt besorgen	900
17.2.3	Das Neuzeichnen einer Grafik mit »ResizeRedraw« und »Invalidate«	902
17.2.4	Zerstören von grafischen Objekten (Dispose)	903
17.2.5	Das Koordinatensystem von GDI+	904
17.2.6	Einen anderen Ursprungspunkt festlegen	906
17.2.7	Die grafischen Methoden der Klasse »Graphics«	908
17.2.8	Eine Linie zeichnen	909
17.2.9	Mehrere Linien zeichnen	909
17.2.10	Rechtecke zeichnen	910
17.2.11	Polygone zeichnen	910
17.2.12	Ellipsen, Ellipsenbogen und Ellipsesegment	911
17.2.13	Kurvenzüge	911
17.2.14	Bézierkurven	912
17.3	Elementare Klassen für grafische Operationen	913
17.3.1	Die Klasse »Pen«	913
17.3.2	Farbeinstellungen mit »Color«	914
17.4	Die Schriftdarstellung	915
17.4.1	Die Klassen »Font« und »FontFamily«	915
17.4.2	Der Schriftstil mit »FontStyle«	916
17.4.3	Die grafische Ausgabe einer Zeichenfolge	916
17.4.4	Die Abmessungen mit »MeasureString« ermitteln	918
17.4.5	Die Klasse »StringFormat«	918
17.5	Bilddateien	919
17.5.1	Raster- und Vektorgrafiken	919
17.5.2	Bilder und Grafiken der .NET-Klassenbibliothek	921
17.5.3	Die Bitmap-Dateiformate	921
17.5.4	Bilder mit der Klasse »Image«	922
17.5.5	Bitmaps	926

18 Das Drucken (Printing) 933

18.1	Ein einführender Überblick	933
18.2	Einen Druckauftrag erteilen	934
18.2.1	Methoden und Eigenschaften von »PrintDocument«	934
18.2.2	Die Ereignisse eines »PrintDocument«-Objekts	935
18.2.3	Die Ereignisse »BeginPrint« und »EndPrint«	935
18.2.4	Das Ereignis »QueryPageSettings«	936
18.2.5	Das Ereignis »PrintPage«	936

18.2.6	Den Druckauftrag beenden	940
18.2.7	Programmbeispiel	940
18.3	Seiteneinstellungen mit »PageSettings«	942
18.4	Der Drucker als »PrinterSettings«-Objekt	943
18.4.1	Mehrere installierte Drucker	944
18.4.2	Wichtige Eigenschaften des »PrinterSettings«-Objekts	944
18.5	Die Steuerelemente zum Drucken	945
18.5.1	Das »PrintDocument«-Steuerelement	945
18.5.2	Das »PrintDialog«-Steuerelement	946
18.5.3	Das »PageSetupDialog«-Steuerelement	947
18.5.4	Das »PrintPreviewDialog«-Steuerelement	948
18.5.5	Das »PrintPreviewControl«-Steuerelement	949
18.6	Drucken von Grafiken	953
18.6.1	Das Problem der verschiedenen Maßeinheiten	953
18.6.2	Festlegung der Einheiten und Skalierung	954
18.6.3	Beispielprogramm zum Drucken einer Grafik	955
18.7	Ausdruck von mehrseitigem Text	957

19 Steuerelemente entwickeln 963

19.1	Einführung	963
19.1.1	Entwicklungsszenarien	963
19.1.2	Testen eines Steuerelements	964
19.2	Steuerelemente mit »UserControl«	964
19.2.1	Beispielprogramm eines Benutzersteuerelements	966
19.2.2	Die Eigenschaften des Benutzersteuerelements	969
19.2.3	Attribute zur Beeinflussung eines Benutzersteuerelements	971
19.2.4	Ereignisse des Benutzersteuerelements	974
19.2.5	Bereitstellen einer Toolbox-Bitmap	976
19.2.6	Zusammenfassung des Codes für »EllipseLabel«	977
19.3	Ableiten eines vorhandenen Steuerelements	979
19.4	Ein eigenes Steuerelement entwerfen	981

20 Programmiertechniken 985

20.1	Drag&Drop-Operationen	985
20.1.1	Der Ablauf einer Drag&Drop-Operation	985
20.1.2	Das Einleiten einer Drag&Drop-Operation	986
20.1.3	Die Ereignisse des Empfängers einer Drag&Drop-Operation	986
20.1.4	Programmbeispiele	989
20.2	API-Aufrufe mit PInvoke	995
20.2.1	Das Attribut »DllImportAttribute«	996
20.2.2	Datentypen von API-Funktionen	997
20.3	Entwickeln von Windows-Dienstanwendungen	1001
20.3.1	»Windows-Dienst«-Projekte in der Entwicklungsumgebung	1002
20.3.2	Die Methoden eines Dienstes	1004
20.3.3	Die Eigenschaften eines Dienstes	1005
20.3.4	Die Installation eines Windows-Dienstes	1007

20.3.5	Beispielprogramm »FileWatchService«	1009
20.3.6	Dienste mittels Programmcode steuern	1014
20.4	Die Zwischenablage	1019
20.4.1	Speichern und Abrufen von Daten aus der Zwischenablage	1019
20.4.2	Andere Methoden, um Daten in die Zwischenablage zu schreiben und abzurufen	1022
20.4.3	Mehrere Datenformate in der Zwischenablage	1023
20.4.4	Eigene Datenformate in die Zwischenablage schreiben	1026
20.4.5	Beispielprogramm – das Menü »Bearbeiten«	1026
20.5	Lokalisierung von Programmen	1031
20.5.1	Lokalisierung von Komponenten	1032
20.5.2	Zusätzliche Lokalisierungsdateien	1035

21 WPF – die Grundlagen 1037

21.1	Die Merkmale einer WPF-Anwendung	1037
21.2	Anwendungstypen	1038
21.2.1	WPF-Anwendung	1039
21.2.2	WPF-Browseranwendung	1039
21.3	Eine WPF-Anwendung und deren Dateien	1039
21.3.1	Die Datei »Window1.xaml«	1040
21.3.2	Die Datei »Window1.xaml.cs«	1041
21.3.3	Die Datei »App.xaml«	1041
21.3.4	Die Datei »App.xaml.cs«	1042
21.3.5	Die Dateien ».baml« und ».g.cs«	1042
21.4	XAML (eXtended Markup Language)	1042
21.4.1	Namespaces, Tags und Attribute	1043
21.5	Markup-Erweiterungen und XAML-Schlüsselwörter	1046
21.5.1	Markup-Erweiterungen	1046
21.5.2	XAML-Schlüsselwörter	1047

22 Die Layoutcontainer 1049

22.1	Übersicht über die Layoutcontainer	1049
22.2	Das »Canvas«	1050
22.3	Das »StackPanel«	1051
22.4	Das »WrapPanel«	1053
22.5	Das »DockPanel«	1055
22.6	Das »UniformGrid«	1057
22.7	Der Container »Grid«	1058
22.8	Steuerelemente positionieren	1063
22.8.1	Die Eigenschaften »Top«, »Bottom«, »Right« und »Left«	1063
22.8.2	Den Außenrand mit der Eigenschaft »Margin« festlegen	1064
22.8.3	Den Innenrand mit der Eigenschaft »Padding« festlegen	1064
22.9	Verschachtelte Layout-Container	1065

23 Die WPF-Controls	1069
23.1 Ereignisse programmieren	1069
23.1.1 Weiterleiten von Ereignissen	1070
23.2 Die Basisklasse »Control«	1075
23.3 Die Familie der Buttons	1076
23.3.1 Die Klasse »Button«	1077
23.3.2 Die Klasse »RepeatButton«	1077
23.3.3 Die Klasse »ToggleButton«	1078
23.4 Textboxen und Artverwandte	1078
23.4.1 Das Steuerelement »TextBox«	1079
23.4.2 Die »PasswordBox«	1079
23.5 Das Steuerelement »Label«	1080
23.6 »CheckBox« und »RadioButton«	1080
23.6.1 Das »CheckBox«-Control	1080
23.6.2 Der »RadioButton«	1081
23.7 »ListBox« und »ComboBox«	1081
23.7.1 Das Steuerelement »ListBox«	1081
23.7.2 Die »ComboBox«	1083
23.8 Die Menüleiste	1084
23.8.1 Weitere Möglichkeiten der Menüleiste	1085
23.9 Das Kontextmenü	1087
23.10 Symbolleisten	1088
23.10.1 Positionieren mit der Komponente »ToolBarTray«	1090
23.11 Die Statusleiste	1090
23.12 Die »Image«-Komponente	1091
23.12.1 Grafik zur Laufzeit laden	1092
24 Konzepte von WPF	1095
24.1 Ressourcen	1097
24.1.1 Eine Ressource definieren	1098
24.1.2 Statische und dynamische Ressourcen	1099
24.1.3 Abrufen von Systemressourcen	1101
24.2 Stile	1102
24.2.1 Eine einfache Style-Definition	1102
24.2.2 Typisierte Stile	1103
24.3 Trigger	1106
24.3.1 Eigenschaftstrigger	1106
24.3.2 Ereignistrigger	1108
24.3.3 Datentrigger	1110
24.4 Commands	1111
24.4.1 Vordefinierte Commands	1111
24.4.2 Beispielanwendung	1112
24.4.3 Das Commando-Ziel festlegen	1113
24.4.4 Commands an Ereignisse binden	1113
24.4.5 Commands programmieren	1115

24.5	Datenbindung	1116
24.5.1	Eine einfache Datenbindung	1116
24.6	Bindungsmodi	1118
24.6.1	Aktualisierung der Quelle	1120
24.6.2	Verschiedene Datenbindungsquellen	1121
24.6.3	Collections anbinden	1124

25 ADO.NET – die Verbindung zu einer Datenbank herstellen 1129

25.1	Die ADO.NET-Provider	1129
25.2	Verbindungen mit dem Sql-Datenprovider	1130
25.2.1	Das »Connection«-Objekt	1130
25.2.2	Die Verbindungszeichenfolge	1131
25.2.3	Die Verbindung mit einer bestimmten SQL Server-Instanz aufbauen	1132
25.2.4	Änderung des Passworts bei der SQL Server-Authentifizierung	1134
25.2.5	Verbindungszeichenfolgen mit dem »SqlConnectionStringBuilder«-Objekt	1135
25.2.6	Das Öffnen und Schließen einer Verbindung	1136
25.3	Das Verbindungspooling	1140
25.3.1	Beispiel für das Verbindungspooling	1140
25.3.2	Das Verbindungspooling deaktivieren	1142
25.3.3	Die Verbindungspoolgröße manipulieren	1142
25.3.4	Freigabe gepoolter Verbindungen	1143
25.4	Die Ereignisse eines »Connection«-Objekts	1143
25.4.1	Das Ereignis »InfoMessage«	1143
25.4.2	Das Ereignis »StateChange«	1145
25.5	Verbindungszeichenfolgen aus einer Konfigurationsdatei abrufen	1145
25.6	Die Klasse »SqlConnection« im Überblick	1147
25.6.1	Eigenschaften eines »Connection«-Objekts	1148
25.6.2	Methoden des »Connection«-Objekts	1148
25.7	Verbindungen mit dem OleDb-Datenprovider	1149
25.7.1	Verbindungsaufbau zu einer SQL Server-Datenbank	1150
25.7.2	Verbindungsaufbau zu einer Access-Datenbank	1150
25.7.3	Authentifizierung mit dem OleDb-Provider	1151

26 Die Datenbankabfrage 1153

26.1	Das »SqlCommand«-Objekt	1153
26.1.1	Ein »SqlCommand«-Objekt erzeugen	1154
26.1.2	Ausführen des »SqlCommand«-Objekts	1155
26.1.3	Die Eigenschaft »CommandTimeout« des »SqlCommand«- Objekts	1156
26.1.4	Aktionsabfragen absetzen	1156
26.1.5	Abfragen, die genau ein Ergebnis liefern	1158
26.2	Das »SqlDataReader«-Objekt	1159
26.2.1	Datensätze einlesen	1159
26.2.2	Schließen des »SqlDataReader«-Objekts	1162

26.2.3	MARS (Multiple Active Resultsets)	1162
26.2.4	Batch-Abfragen mit »NextResult« durchlaufen	1164
26.2.5	Das Schema eines »SqlDataReader«-Objekts untersuchen	1165
26.3	Parametrisierte Abfragen	1167
26.3.1	Parametrisierte Abfragen mit dem SqlClient-Datenprovider	1167
26.3.2	Die Klasse »SqlParameter«	1170
26.3.3	Parametrisierte Abfragen mit dem OleDb-Datenprovider	1170
26.4	Asynchrone Abfragen	1171
26.4.1	Das Polling-Verfahren	1172
26.4.2	Bereitstellen einer Rückrufmethode	1173
26.5	Gespeicherte Prozeduren (Stored Procedures)	1174
26.5.1	Allgemeines zu gespeicherten Prozeduren	1174
26.5.2	Gespeicherte Prozeduren im Visual Studio 2008 erstellen	1175
26.5.3	Eine gespeicherte Prozedur aufrufen	1178
26.5.4	Komplexe gespeicherte Prozeduren	1180

27 Der SqlDataAdapter **1183**

27.1	Die Konstruktoren der Klasse »DataAdapter«	1185
27.2	Die Eigenschaft »SelectCommand«	1185
27.3	Den lokalen Datenspeicher füllen (die Fill-Methode)	1186
27.3.1	Verbindungen öffnen und schließen	1187
27.3.2	Doppelter Aufruf der »Fill«-Methode	1187
27.3.3	Mehrere »DataAdapter«-Objekte aufrufen	1188
27.3.4	Die Spalten- und der Tabellenbezeichner einer »DataTable«	1189
27.3.5	Paging mit der »Fill«-Methode	1189
27.4	Tabellenzuordnung mit »TableMappings«	1189
27.4.1	Spaltenzuordnungen in einem »DataSet«	1191
27.4.2	Spaltenzuordnungen einer »DataTable«	1193
27.4.3	Die Eigenschaft »MissingMappingAction« des DataAdapters	1193
27.5	Das Ereignis »FillError« des »SqlDataAdapters«	1194

28 Daten im lokalen Speicher – das DataSet **1197**

28.1	Verwenden des »DataSet«-Objekts	1198
28.1.1	Ein »DataSet«-Objekt erzeugen	1198
28.1.2	Die Anatomie einer »DataTable«	1198
28.1.3	Der Zugriff auf eine Tabelle im »DataSet«	1199
28.1.4	Der Zugriff auf die Ergebnisliste	1200
28.1.5	Dateninformationen in eine XML-Datei schreiben	1201
28.2	Dem »DataSet« Schemainformationen übergeben	1202
28.2.1	Schemainformationen bereitstellen	1204
28.2.2	Eigenschaften einer »DataColumn«, die der Gültigkeitsprüfung dienen	1204
28.2.3	Die »Constraints«-Klassen einer »DataTable«	1205
28.2.4	Das Schema mit Programmcode erzeugen	1206
28.2.5	Schemainformationen mit dem »SqlDataAdapter« abrufen	1207
28.2.6	Schemainformationen aus einer XML-Schemadatei beziehen	1209

28.3	Änderungen in einer »DataTable« vornehmen	1211
28.3.1	Editieren einer »DataRow«	1211
28.3.2	Löschen einer Datenzeile	1214
28.3.3	Eine neue Datenzeile hinzufügen	1214
28.4	Was bei der Änderung einer Datenzeile passiert	1217
28.4.1	Die Eigenschaft »RowState«	1217
28.4.2	Der ursprüngliche und der aktualisierte Inhalt einer Datenzeile	1218
28.4.3	Die Eigenschaft »DataRowState« manuell steuern	1221
28.5	Mit mehreren Tabellen arbeiten	1222
28.5.1	Der Weg über JOIN-Abfragen	1222
28.5.2	Mehrere Tabellen in einem DataSet	1224
28.5.3	Eine »DataRelation« erzeugen	1224
28.5.4	»DataRelations« und Einschränkungen	1225
28.5.5	In Beziehung stehende Daten suchen	1228
28.5.6	Ergänzung zum Speichern von Schemainformationen in einer XML-Schemadatei	1231
28.6	Suchen und filtern im »DataSet«	1231
28.6.1	Die »Find«-Methode	1231
28.6.2	Die »Select«-Methode	1233
28.7	Objekte vom Typ »DataView«	1236
28.7.1	Eine »DataView« erzeugen	1236
28.7.2	Auf die Datenzeilen in einer »DataView« zugreifen	1237
28.7.3	Die Eigenschaft »Sort« und die Methode »Find«	1237
28.7.4	Die Methode »FindRows«	1238
28.7.5	Die Eigenschaft »RowFilter«	1238
28.7.6	Die Eigenschaft »RowStateFilter«	1238
28.7.7	Änderungen an einem »DataView«-Objekt	1239
28.7.8	Aus einer »DataView« eine »DataTable« erzeugen	1240

29 Eine Datenbank aktualisieren 1243

29.1	Aktualisieren mit dem »CommandBuilder«-Objekt	1246
29.1.1	Simulation eines Parallelitätskonflikts	1248
29.1.2	Die von »SqlCommandBuilder« generierten Aktualisierungsstatements	1248
29.1.3	Weitere Aktualisierungsoptionen des SqlCommandBuilders	1249
29.1.4	Die Vor- und Nachteile des SqlCommandBuilders	1251
29.2	Manuell gesteuerte Aktualisierungen	1251
29.2.1	Manuelles Aktualisieren mit dem DataAdapter	1251
29.2.2	Aktualisieren mit »ExecuteNonQuery«	1255
29.3	Den Benutzer über die fehlgeschlagenen Aktualisierungen informieren	1260
29.4	Die konfliktverursachenden Datenzeilen bei der Datenbank abfragen	1263
29.5	Eine geänderte »DataRow« im »DataSet« aktualisieren	1268
29.5.1	Die Eigenschaft »UpdatedRowSource« des »SqlCommands«	1269
29.6	Hierarchische Änderungen an die Datenbank übermitteln	1274
29.6.1	Hierarchische Änderungen mit der »Select«-Methode	1275
29.6.2	Hierarchische Änderungen mit der Methode »GetChanges«	1278

30 Stark typisierte DataSets	1279
30.1 Ein stark typisiertes DataSet erzeugen	1279
30.1.1 Typisierte DataSets mit dem Visual Studio Designer erstellen	1279
30.1.2 Das Kommandozeilentool XSD.exe	1282
30.2 Die Anatomie eines typisierten DataSets	1283
30.2.1 Die Datenzeilen einer Tabelle ausgeben	1283
30.2.2 Datenzeilen hinzufügen	1286
30.2.3 Datenzeilen bearbeiten	1287
30.2.4 Datenzeilen suchen	1287
30.2.5 NULL-Werte im typisierten DataSet	1288
30.2.6 Die Daten in einem hierarchischen DataSet	1288
30.3 Typisierte DataSets manuell im Designer erzeugen	1289
30.3.1 Eine »DataTable« manuell erzeugen	1289
30.3.2 Der »DataTable« Spalten hinzufügen	1289
30.3.3 Beziehungen zwischen den Tabellen erstellen	1290
30.4 Weitergehende Betrachtungen	1291
30.5 Der »TableAdapter«	1291
30.5.1 Einen »TableAdapter« mit Visual Studio erzeugen	1292
30.6 TableAdapter im Code verwenden	1297
30.6.1 Die Methode »Fill« des »TableAdapters«	1297
30.6.2 Die Methode »GetData«	1297
30.6.3 Die Methode »Update«	1298
30.6.4 Aktualisieren mit den DBDirect-Methoden	1298
30.6.5 TableAdapter mit mehreren Abfragen	1299
30.6.6 Änderungen an einem »TableAdapter« vornehmen	1301
30.7 Fazit: Typisierte oder nicht typisierte DataSets?	1302
31 Weitergabe von Anwendungen	1305
31.1 Weitergabe mit MS-Installer	1305
31.1.1 Weitergabeprojekte	1305
31.1.2 Der »Windows Installer«	1306
31.1.3 Weitergabeprojekte mit dem »Setup-Assistenten«	1307
31.1.4 Die Editoren eines Weitergabeprojekts	1309
31.1.5 Der »Dateisystem-Editor«	1311
31.1.6 Der »Registrierungs-Editor«	1317
31.1.7 Der »Dateityp-Editor«	1319
31.1.8 Der »Benutzeroberflächen-Editor«	1322
31.1.9 Der »Editor für benutzerdefinierte Aktionen«	1332
31.1.10 Der »Editor für Startbedingungen«	1333
31.2 »ClickOnce«-Verteilung	1337
31.2.1 Erstellen einer ClickOnce-Anwendung	1339
31.2.2 Die Installation einer ClickOnce-Anwendung	1342
Über den Autor	1345
Index	1347