

Übersicht

Vorwort	25
Die DVD zum Buch	29

Teil I Einführung

1 C# und das .NET Framework	33
2 Visual Studio 2010	55
3 Das erste Programm	101

Teil II Grundlagen C#

4 Datentypen	115
5 Ablaufsteuerung	151
6 Klassen und Objekte	167
7 Vererbung und Polymorphie	223
8 Interfaces	241
9 Delegates und Events	265
10 Arbeiten mit Datentypen	291
11 Collections	331
12 Generics	369
13 Language Integrated Query (LINQ)	407
14 Fehlersuche und Fehlerabsicherung	465

Teil III Grundlagen WPF

15 Einführung in WPF	499
16 Steuerelemente von WPF	549
17 Verwenden von Ressourcen	711

Inhaltsübersicht

18 Datenbindung in WPF	739
19 Stile und Vorlagen	799
20 Commands in WPF	827

Teil IV Arbeiten mit dem Framework

21 Dateien und Verzeichnisse	841
22 Serialisierung.....	971
23 Reflection	991
24 Multithreading	1021

Teil V Datenbanken

25 Datenbankgrundlagen.....	1075
26 Überblick über ADO.NET.....	1105
27 LINQ to SQL	1161
28 Das Entity Framework (EF).....	1213

Anhang

Das MVVM-Pattern.....	1249
Index	1271

Inhalt

Vorwort	25
Die DVD zum Buch	29
Teil I Einführung	31
1 C# und das .NET Framework	33
1.1 Die Geburtsstunde von .NET	35
1.2 Die Sprache C#	36
1.3 Vorteile von .NET	36
1.4 Aufbau des .NET Frameworks	39
1.4.1 Übersicht	40
1.4.2 Die Common Language Runtime	40
1.4.3 Programmausführung unter .NET	43
1.4.4 Die Klassenbibliothek	44
1.4.5 Die Benutzerschnittstelle	47
1.4.6 Common Language Specification und Common Type System ..	48
1.4.7 Der Intermediate Language Code (IL-Code)	49
1.5 Neuerungen in .NET	50
1.5.1 Neuerungen in .NET 2.0/C# 2.0	50
1.5.2 Neuerungen in .NET 3.5/C# 3.0	52
1.5.3 Neuerungen in .NET 4.0/C# 4.0	53
2 Visual Studio 2010	55
2.1 Übersicht	56
2.2 Der Startbildschirm	58
2.2.1 Bereich Erste Schritte Willkommen	59
2.2.2 Anpassen der Visual Studio-Startseite	59
2.2.3 Bereich Leitfaden und Ressourcen	66
2.2.4 Bereich Aktuelle Nachrichten	66
2.3 Projekte und Projektmappen	66
2.4 Der WPF-Designer	68
2.4.1 Der Projektmappen-Explorer	70
2.4.2 Die Toolbox	73
2.4.3 Das Eigenschaftenfenster	75

Inhalt

2.5	Projekteigenschaften	77
2.5.1	Bereich Anwendung	78
2.5.2	Bereich Erstellen.....	80
2.5.3	Build-Ereignisse	83
2.5.4	Bereich Debuggen.....	84
2.5.5	Bereich Signieren	85
2.5.6	Bereich Codeanalyse.....	86
2.6	Der Text-Editor	88
2.6.1	IntelliSense	88
2.6.2	SmartTags	89
2.6.3	Änderungsüberwachung.....	90
2.6.4	Refactoring.....	90
2.6.5	Weitere Features des Editors	90
2.7	Tools im Visual Studio	91
2.7.1	GUIDs erstellen	91
2.7.2	Klassendiagramm anzeigen.....	91
2.7.3	Sequenzdiagramm anzeigen	93
2.8	Einstellungen	94
2.8.1	Umgebungseinstellungen.....	94
2.8.2	Schriftarten und Farben	95
2.8.3	Projekte und Projektmappen	96
2.8.4	Text-Editor-Einstellungen	97
2.8.5	Bereich Debugging	99
2.8.6	Einstellungen laden und speichern.....	100
2.9	Fazit	100
3	Das erste Programm.....	101
3.1	Hello World	101
3.1.1	Der Pfad zum .NET Framework	102
3.1.2	Der Quelltext	104
3.1.3	Analyse des Programms.....	105
3.1.4	Besonderheiten zur Methode Main()	108
3.2	Hello World, Variante 2.....	109
3.2.1	Analyse des Programms.....	110
3.3	Fazit	111

Teil II Grundlagen C#	113
4 Datentypen.....	115
4.1 Werte- und Referenztypen	115
4.1.1 Unterschiede zwischen Werte- und Referenztypen	116
4.2 Integrierte Datentypen	120
4.2.1 Der Datentyp bool	121
4.2.2 Der Datentyp char	122
4.2.3 Numerische Datentypen	123
4.2.4 Der Datentyp string	123
4.2.5 Nullable Types	124
4.3 Variablen	126
4.3.1 Deklaration und Initialisierung	127
4.3.2 Bezeichner	128
4.3.3 Gültigkeitsbereich	129
4.3.4 Konstanten	130
4.4 Konvertierungen und Boxing	130
4.4.1 Implizite und explizite Konvertierung	131
4.4.2 Boxing und Unboxing	132
4.4.3 Typumwandlung in und von String	133
4.5 Arrays	134
4.5.1 Eindimensionale Arrays	134
4.5.2 Mehrdimensionale Arrays	135
4.5.3 Ungleichförmige (jagged) Arrays	136
4.5.4 Arbeiten mit Arrays	137
4.5.5 Syntax-Zusammenfassung	141
4.6 Aufzählungstypen (enum)	142
4.6.1 Grundlagen	142
4.6.2 Deklaration und Anwendung	143
4.6.3 Bitfelder (Flag-Enums)	145
4.6.4 Arbeiten mit Aufzählungen	146
4.6.5 Syntax-Zusammenfassung	149
5 Ablaufsteuerung	151
5.1 Verzweigungen	151
5.1.1 Die if-Anweisung	151
5.1.2 Die switch-Anweisung	153
5.1.3 Die bedingte Zuweisung	155
5.1.4 Die goto-Anweisung	155

Inhalt

5.2	Schleifen	157
5.2.1	Die for-Schleife	157
5.2.2	Die while-Schleife	158
5.2.3	Die do-Schleife.....	159
5.2.4	Die foreach-Schleife	159
5.3	Operatoren.....	160
5.3.1	Arithmetische Operatoren	161
5.3.2	Vergleichsoperatoren	162
5.3.3	Logische Operatoren.....	162
5.3.4	Bitweise Operatoren	163
5.3.5	Zusammengesetzte Operatoren.....	164
5.3.6	Sonstige Operatoren	164
6	Klassen und Objekte	167
6.1	Grundlagen der Objektorientierung	168
6.1.1	Geschichtliches	168
6.1.2	Grundkonzepte.....	168
6.1.3	Abstraktion.....	170
6.1.4	Kapselung	171
6.1.5	Vererbung.....	172
6.1.6	Polymorphie.....	174
6.2	Gliederung einer Anwendung: Namespaces	175
6.2.1	Deklaration.....	175
6.2.2	Aufteilen der Deklaration	177
6.3	Klassen	178
6.3.1	Klassendeklaration	178
6.3.2	Felder (Instanzvariablen)	179
6.3.3	Methoden	181
6.3.4	Optionale und benannte Parameter.....	193
6.3.5	Konstruktoren und Destruktoren.....	196
6.3.6	Eigenschaften (Properties).....	202
6.4	Statische Klassenelemente	208
6.4.1	Private Konstruktoren.....	209
6.4.2	Statische Konstruktoren.....	210
6.4.3	Statische Klassen	211
6.5	Modifizierer	211
6.6	Operatorenüberladung.....	213
6.6.1	Überladen mathematischer Operatoren.....	214
6.6.2	Überladen der Konvertierungsoperatoren	215

6.7	Strukturen [struct]	216
6.7.1	Deklaration.....	216
6.7.2	Nullable Structs.....	218
6.8	Partielle Datentypen und partielle Methoden	219
6.8.1	Partielle Datentypen	219
6.8.2	Partielle Methoden	219
6.9	Objektinitialisierer	221
7	Vererbung und Polymorphie	223
7.1	Vererbung	224
7.1.1	Ableiten von Klassen.....	224
7.1.2	Gemeinsame Methoden aller Klassen.....	227
7.1.3	Virtuelle Methoden	228
7.2	Polymorphie	231
7.2.1	Verdeckte Member	234
7.3	Konvertierung von Referenztypen.....	236
7.4	Abstrakte, versiegelte und verschachtelte Klassen	238
7.4.1	Versiegelte Klassen.....	238
7.4.2	Abstrakte Klassen	238
7.4.3	Verschachtelte Klassen.....	240
8	Interfaces	241
8.1	Grundlagen	241
8.1.1	Deklaration.....	243
8.1.2	Implementierung.....	244
8.1.3	Interfaces explizit verwenden.....	250
8.1.4	Nicht implementierte Methoden	252
8.2	Spezielle Interfaces	253
8.2.1	Die Interfaces IComparable und IComparer	253
8.2.2	Das Interface IDisposable	259
8.2.3	Das Interface IEnumerable	259
8.3	Erweiterungsmethoden (Extension Methods)	260
8.3.1	Erweiterung von Datentypen	260
8.3.2	Deklaration von Erweiterungsmethoden.....	261
8.3.3	Beispielimplementierung	261

Inhalt

9 Delegates und Events	265
9.1 Grundlagen zu Delegates	266
9.2 Verwenden von Delegates	267
9.2.1 Eine Sortierroutine	267
9.2.2 Ein Delegate – mehrere Methoden	272
9.2.3 Anonyme Methoden	274
9.3 Ereignisse (Events)	277
9.3.1 Aufbau von Ereignissen	278
9.3.2 Ereignisse verwenden	280
9.3.3 Ein Ereignis in fünf Schritten	284
9.3.4 Zurückliefern von Werten in Ereignissen	285
10 Arbeiten mit Datentypen	291
10.1 Zahlen	291
10.1.1 Notation	291
10.1.2 Rundungsfehler	292
10.1.3 Division durch null und der Wert unendlich	293
10.1.4 Arithmetische Funktionen	294
10.1.5 Zahlen runden und andere Funktionen	295
10.1.6 Zufallszahlen	296
10.2 Strings	297
10.2.1 Grundlagen	297
10.2.2 Verketten von Strings	298
10.2.3 Zugriff auf Zeichenketten	300
10.2.4 Vergleiche von Zeichenketten	303
10.2.5 Die Klasse StringBuilder	304
10.2.6 Unicode	307
10.2.7 Syntaxzusammenfassung	309
10.3 Datum und Zeit	313
10.3.1 Die Struktur DateTime	314
10.3.2 Die Struktur TimeSpan	315
10.3.3 Arbeiten mit Datum und Zeit	316
10.3.4 Zeitmessungen – die Klasse Stopwatch	318
10.3.5 Syntaxzusammenfassung	318
10.4 Formatierungsmethoden in .NET	322
10.4.1 Grundlagen	322
10.4.2 Zahlen formatieren	324
10.4.3 Datum und Zeit formatieren	328

Collections	331
1.1 Grundlagen	332
1.2 Die Listenklassen aus System.Collections	332
11.2.1 Übersicht	332
11.2.2 Übersicht über die verwendeten Interfaces	335
1.3 Grundlegende Programmiertechniken	337
11.3.1 Listenelemente löschen	337
11.3.2 Sortieren von Listen	341
11.3.3 Suchen mit BinarySearch().	347
11.3.4 Queue und Stack verwenden	349
11.3.5 Datenaustausch zwischen Listen	352
1.4 Eigene Listenklassen erstellen	353
11.4.1 Eine neue Art von Eigenschaft: der Indexer	354
11.4.2 Implementierung der foreach-Schleife	357
11.4.3 Beispielprogramm Personenliste	358
1.5 Syntaxzusammenfassung	363
11.5.1 Interfaces	363
11.5.2 Klassen	365
Generics	369
2.1 Grundlagen zu Generics	370
12.1.1 Vorteile von Generics	370
12.1.2 Deklaration	377
2.2 Deklaration generischer Klassen	377
12.2.1 Erstellen einer nicht generischen Liste	378
12.2.2 Umbau der Liste auf Generics	383
2.3 Constraints (Einschränkungen)	388
2.4 Vererbung mit Generics	389
12.4.1 Konkrete Klassen mit generischer Basisklasse	390
12.4.2 Generische Klassen mit generischer Basisklasse	390
2.5 Generische Methoden	391
2.6 Generische Listenklassen	393
12.6.1 Übersicht	393
12.6.2 Zusätzliche Methoden generischer Listenklassen	394
2.7 Covarianz und Contravarianz	395
12.7.1 Covarianz in Interfaces	396
12.7.2 Covarianz in Kürze	400
12.7.3 Contravarianz	401
12.7.4 Contravarianz in Kürze	405

Inhalt

13 Language Integrated Query (LINQ)	407
13.1 Filtern und Abfragen von Listen	408
13.1.1 Filtern einer Liste ohne LINQ	408
13.1.2 Filtern einer Liste mit LINQ	415
13.2 Funktionsweise von Lambda-Expressions	416
13.2.1 Schritt 1: Deklarieren des Delegates und der Where()-Methode .	416
13.2.2 Schritt 2: Erzeugen des predicate-Parameters	417
13.2.3 Schritt 3: Einsatz der Extension Methods	418
13.2.4 Schritt 4: Anonyme Methoden	419
13.2.5 Schritt 5: Der Lambda-Operator	420
13.3 LINQ-Methoden	421
13.3.1 Abfragen und Filtern	424
13.3.2 Definierte Auswahl, Projektionen und anonyme Klassen	428
13.3.3 Aggregatsfunktionen	435
13.3.4 Quantifizierung	438
13.3.5 Gruppierung	439
13.3.6 Joinen von Listen	441
13.3.7 Arbeiten mit nicht generischen Listen	446
13.4 LINQ-Queries	448
13.4.1 Eine einfache LINQ-Query	448
13.4.2 Übersicht über die Query-Operatoren	449
13.4.3 Filtern und Sortieren mit LINQ-Queries	450
13.4.4 Projektionen mit Queries	455
13.4.5 Gruppieren von Listen mit Queries	458
13.4.6 Join mit Query-Syntax	460
13.5 Fazit	464
14 Fehlersuche und Fehlerabsicherung	465
14.1 Fehlerabsicherung	466
14.1.1 Abfangen von Exceptions	467
14.1.2 try-catch-Syntax	469
14.1.3 Verschiedene Exceptions abfangen	472
14.1.4 Der finally-Block	474
14.1.5 Eigenschaften und Methoden der Klasse Exception	480
14.1.6 Eigene Exception-Klassen	481
14.2 Fehlersuche (Debugging)	483
14.2.1 Grundlagen	483
14.2.2 Fehlersuche in der Entwicklungsumgebung	488

14.2.3	Ausgaben der Klasse Debug	494
14.2.4	Syntaxzusammenfassung	496
14.3	Fazit	496
Teil III	Grundlagen WPF	497
15	Einführung in WPF	499
15.1	Features von WPF	500
15.2	Hello WPF	501
15.2.1	Anlegen des Projekts	501
15.2.2	Basisdesign des Fensters	503
15.2.3	Ein Button zum Beenden	508
15.2.4	Verschönern des Fensters	511
15.2.5	Hinzufügen der Funktionalität	514
15.3	Neue Konzepte in WPF	515
15.3.1	Oberflächengestaltung durch XAML	515
15.3.2	Logischer und visueller Elementbaum	522
15.3.3	Dependency Properties und Attached Properties	527
15.3.4	Routed Events	530
15.3.5	Applikationsstart	532
15.4	Programmieren mit XAML	535
15.4.1	XAML Namespaces	535
15.4.2	Festlegen von Eigenschaftswerten in XAML	537
15.4.3	Typkonvertierung	539
15.4.4	Markup-Erweiterungen (Markup-Extensions)	540
15.4.5	Verwenden von XAML-Schlüsselwörtern	545
15.5	Fazit	547
16	Steuerelemente von WPF	549
16.1	Das Objektmodell	550
16.1.1	Eigenschaften der Basisklassen	551
16.1.2	Routed Events der Basisklassen	557
16.2	Layout-Container-Steuerelemente	559
16.2.1	Canvas	560
16.2.2	StackPanel	562
16.2.3	DockPanel	563
16.2.4	WrapPanel	566
16.2.5	Grid	567

Inhalt

16.3	Buttons	585
16.3.1	Die Basisklasse ButtonBase.	585
16.3.2	Das Steuerelement Button	586
16.3.3	Das Steuerelement ToggleButton	586
16.3.4	Das Steuerelement CheckBox	588
16.3.5	Das Steuerelement RadioButton	589
16.3.6	Das Steuerelement RepeatButton	590
16.4	Steuerelemente für Text.	590
16.4.1	Das Steuerelement TextBlock	590
16.4.2	Das Steuerelement Label	592
16.4.3	Das Steuerelement TextBox	595
16.4.4	Das Steuerelement PasswordBox	601
16.4.5	Das Steuerelement RichTextBox	601
16.5	Listenelemente	610
16.5.1	ItemsControls und HeaderedItemsControls	610
16.5.2	Das Steuerelement ListBox	611
16.5.3	Das Steuerelement ComboBox	626
16.5.4	Das Steuerelement TreeView	631
16.5.5	Das Steuerelement ListView.	641
16.5.6	Das Steuerelement DataGrid	648
16.5.7	Das Steuerelement TabControl	648
16.6	Datums-Steuerelemente	652
16.6.1	Das Steuerelement Calendar	652
16.6.2	Das Steuerelement DatePicker	660
16.7	Steuerelemente für Menüs und Toolbars	661
16.7.1	Die Steuerelemente Menu und ContextMenu	661
16.7.2	Das Steuerelement ToolBar	665
16.7.3	Das Steuerelement StatusBar	672
16.8	Weitere Steuerelemente.	673
16.8.1	Das Steuerelement ScrollViewer	673
16.8.2	Das Steuerelement ProgressBar	676
16.8.3	Das Steuerelement GroupBox	677
16.8.4	Das Steuerelement Border.	680
16.8.5	Das Steuerelement Expander.	681
16.8.6	Das Steuerelement Image	684
16.8.7	Das Steuerelement MediaElement	688
16.9	Dialoge.	700
16.9.1	Standarddialoge.	700
16.9.2	Die MessageBox	701
16.9.3	Der TaskDialog.	702

16.9.4	Das Windows API Code Pack	707
16.10	Fazit	709
17	Verwenden von Ressourcen	711
17.1	Binäre und logische Ressourcen	711
17.2	Verwenden binärer Ressourcen	712
17.2.1	Hinzufügen binärer Ressourcen zu einem Projekt	712
17.2.2	Zugriff auf Dateiressourcen	713
17.2.3	Die Pack-Syntax	717
17.2.4	Zugriff auf Ressourcen in Satelliten-Assemblies	719
17.3	Verwenden logischer Ressourcen	722
17.3.1	StaticResource und DynamicResource	722
17.3.2	Definieren logischer Ressourcen	723
17.3.3	Zusätzliche Ressourcendateien	727
17.3.4	Ressourcen in C# verwenden	731
17.4	Verwenden der Datei Resources.resx	732
17.4.1	Zugriff auf Grafiken und Strings aus Resources.resx	733
17.4.2	Direkter Zugriff auf String-Ressourcen aus Resources.resx	737
17.5	Fazit	738
18	Datenbindung in WPF	739
18.1	Grundlagen	740
18.2	Einfache Datenbindung	741
18.2.1	Synchronisieren von TextBox-Steuerelementen	741
18.2.2	Datenbindung an ein Objekt	744
18.2.3	Der DataContext	751
18.2.4	Datenbindung über C#-Code	753
18.3	Konvertierung von Daten bei Datenbindung	755
18.3.1	Das Interface IValueConverter	755
18.3.2	Verwenden der benutzerdefinierten Konvertierung	756
18.4	Datenbindung mit Listen	760
18.4.1	Eigenschaften für die Datenbindung von Listen	761
18.4.2	Einfache Bindung einer Liste	761
18.4.3	Verwenden von Datenvorlagen	763
18.4.4	Auswahl eines DataTemplates über ItemTemplateSelector	772
18.5	Datenbindung an eine TreeView	777
18.5.1	Das Hauptfenster	777
18.5.2	Festlegen des Datenkontextes	778
18.5.3	Festlegen der Hierarchischen Datenvorlagen	779

Inhalt

18.6	Das Steuerelement DataGridView	781
18.6.1	Eigenschaften	781
18.6.2	Beispielapplikation: Datenbindung mit DataGridView	782
18.6.3	Verwenden eigener Spaltendefinitionen	783
18.6.4	Ein DataGridView mit eigenem Stil	784
18.7	Datenbindung an mehrere Quellen	785
18.7.1	MultiBinding	785
18.7.2	PriorityBinding	792
18.8	Fazit	797
19	Stile und Vorlagen	799
19.1	Verwendung und Aufbau von Stilen	800
19.1.1	Setzen von Eigenschaften	800
19.1.2	Verwenden von Triggern	804
19.2	Vorlagen für Steuerelemente	805
19.2.1	Ermitteln der Standard-ControlTemplates	805
19.2.2	Erweiterung des Programms	812
19.2.3	Wichtige Bestandteile von ControlTemplates	817
19.2.4	Datenübernahme mittels TemplateBinding	817
19.2.5	Eine einfache Steuerelementvorlage: Button	818
19.2.6	Eine komplexere Steuerelementvorlage: Slider	820
19.3	Fazit	826
20	Commands in WPF	827
20.1	Grundlegender Aufbau der Command-Architektur	828
20.1.1	Befehlsaufrufe auf klassische Art	828
20.1.2	Befehlsaufrufe mit Commands	829
20.1.3	Vordefinierte Befehle	830
20.2	Verwenden vordefinierter Befehle	830
20.2.1	Definition der Kommandos	832
20.2.2	Deaktivierte Buttons in der ToolBar	833
20.3	Eigene RoutedCommands erstellen	834
20.3.1	Erstellen und Einbauen des Command-Objekts	835
20.3.2	Verwenden des eigenen Befehls in der Anwendung	838
20.4	Fazit	838

Teil IV	Arbeiten mit dem Framework	839
21	Dateien und Verzeichnisse	841
21.1	Grundlagen	841
21.1.1	Streams	842
21.1.2	Klassen von System.IO	842
21.2	Verzeichnisse und Dateien	844
21.2.1	Datei- und Verzeichnisinformationen	845
21.2.2	Ermittlung von Dateien in einem Verzeichnis	847
21.2.3	Manipulation von Dateien und Verzeichnissen	855
21.2.4	Verzeichnisse, Dateien und Laufwerke ermitteln	863
21.2.5	Datei- und Verzeichnisnamen bearbeiten	868
21.2.6	Beispielanwendung Backup	869
21.2.7	Syntaxzusammenfassung	881
21.3	Dialoge für Verzeichnisse und Dateien	892
21.3.1	Der Dialog zum Öffnen einer Datei	892
21.3.2	Der Dialog zum Speichern einer Datei	895
21.3.3	Der Dialog zur Verzeichnisauswahl	895
21.3.4	Syntaxzusammenfassung	899
21.4	Textdateien	900
21.4.1	Codierung von Textdateien	900
21.4.2	Textdateien lesen (mit StreamReader)	901
21.4.3	Textdateien schreiben (StreamWriter)	913
21.4.4	Beispielprogramm – Textdatei erstellen und lesen	914
21.4.5	Beispielprogramm – Textcodierung ändern	919
21.4.6	Zeichenketten lesen und schreiben (StringReader und StringWriter)	920
21.4.7	Syntaxzusammenfassung	921
21.5	Binäre Dateien	925
21.5.1	Die Klasse FileStream	926
21.5.2	Beispielprogramm – Dateien zerteilen	928
21.5.3	BufferedStream	937
21.5.4	MemoryStream (Streams im Arbeitsspeicher)	937
21.5.5	BinaryReader und -Writer (Variablen binär speichern)	943
21.5.6	Beispielprogramm – unterschiedliche Daten schreiben und lesen	945
21.5.7	Syntaxzusammenfassung	951
21.6	Asynchroner Dateizugriff	954
21.6.1	Verwendung eines asynchronen Streams	955
21.6.2	Asynchrones Schreiben mit Callback	957

Inhalt

21.7	Verzeichnisse überwachen	961
21.7.1	FileSystemWatcher verwenden	961
21.7.2	Verzeichnisüberwachung mit Log-Datei	962
21.8	Fazit	970
22	Serialisierung	971
22.1	Grundlagen	972
22.2	Serialisieren mit BinaryFormatter und SoapFormatter	972
22.2.1	Beispielprogramm.....	972
22.2.2	Angepasste Serialisierung	981
22.3	XML-Serialisierung.....	983
22.3.1	Die Klasse XmlSerializer	983
22.3.2	Anpassen der Serialisierungsart.....	988
22.4	Fazit	990
23	Reflection	991
23.1	Grundlagen zu Reflection.....	992
23.1.1	Grundlegende Eigenschaften und Methoden von Type	992
23.1.2	Relevante Klassen für Reflection.....	993
23.2	Beispielapplikation: Informationen über die BCL ermitteln	994
23.2.1	Das Hauptfenster der Anwendung.....	995
23.2.2	Implementierung eines SplashScreens	996
23.2.3	Die Klasse AssemblyReflector	999
23.2.4	Die Klasse TypeReflector	1009
23.2.5	Die Funktionalität des Hauptfensters	1014
23.2.6	Programmstart mit SplashScreen	1017
23.3	Fazit	1019
24	Multithreading	1021
24.1	Grundlagen	1022
24.1.1	Präemptives Multitasking.....	1022
24.1.2	Multithreading-Modelle	1023
24.1.3	Wozu Multithreading?.....	1023
24.2	Arbeiten mit Threads	1024
24.2.1	Die Klasse Thread	1024
24.2.2	Beispielanwendung.....	1028
24.2.3	Syntaxzusammenfassung.....	1032

Inhalt

24.3	Synchronisation	1034
24.3.1	Wozu synchronisieren?	1034
24.3.2	Die Klasse Monitor	1035
24.3.3	Die Anweisung lock()	1043
24.4	Die Komponente BackgroundWorker	1043
24.4.1	Methoden und Ereignisse	1044
24.4.2	Beispielapplikation: Fibonacci	1045
24.5	Threadpools	1052
24.5.1	Die Klasse ThreadPool	1052
24.5.2	Verwenden eines Threadpool	1053
24.6	Parallel Extensions	1058
24.6.1	Bestandteile der Parallel Extensions	1058
24.6.2	Arbeiten mit Tasks	1061
24.6.3	Arbeiten mit parallelen Schleifen	1064
Teil V	Datenbanken	1073
25	Datenbankgrundlagen	1075
25.1	Datenbanksysteme	1076
25.1.1	Übersicht über gebräuchliche Datenbanksysteme	1076
25.2	Einrichten der Datenbankumgebung	1078
25.2.1	Installation des SQL Server 2008	1078
25.2.2	Die Beispieldatenbank des Buchs	1085
25.3	Das SQL Server Management Studio	1086
25.3.1	Verbinden mit einer Datenbank	1087
25.3.2	Datenbanken und Tabellen erzeugen	1089
25.3.3	Tabellen per Code erzeugen	1090
25.3.4	Abfragen erzeugen	1090
25.4	SQL-Grundlagen	1092
25.4.1	Data Definition Language	1092
25.4.2	Data Manipulation Language	1094
25.4.3	Data Query Language	1095
25.4.4	Stored Procedures	1099
25.5	Datenbankaufbau	1102
25.5.1	Globally Unique Identifiers (GUIDs)	1102
25.5.2	Normalformen	1103
25.6	Fazit	1104

Inhalt

26 Überblick über ADO.NET	1105
26.1 Grundlegender Datenbankzugriff	1106
26.1.1 Namespaces für die Datenbankprovider	1106
26.1.2 Die Klasse SqlConnection	1107
26.1.3 Verwenden der Klasse SqlCommandBuilder	1109
26.1.4 Datenbankkommandos absetzen (SqlCommand)	1114
26.1.5 Daten ermitteln (SqlDataReader)	1115
26.1.6 Verwenden von Parametern	1120
26.1.7 Arbeiten mit Stored Procedures	1128
26.2 In-Memory-Datenbank: das DataSet	1132
26.2.1 Klassen für das DataSet	1132
26.2.2 Laden der Daten: SqlDataAdapter	1133
26.2.3 Erzeugen von Kommandos: der SqlCommandBuilder	1137
26.2.4 Speichern der Daten mit SqlDataAdapter	1141
26.2.5 Datenspeicher: die Klasse DataTable	1142
26.2.6 Verknüpfen von Tabellen (DataRelation)	1144
26.3 Typisierte DataSets	1149
26.3.1 Beispielprogramm: Zugriff mittels typisiertem DataSet	1150
26.3.2 Verwenden eines typisierten DataSet	1153
26.4 Fazit	1160
27 LINQ to SQL	1161
27.1 Grundlagen zu LINQ to SQL	1162
27.1.1 Objektidentität	1163
27.1.2 Relationen	1164
27.1.3 Änderungsüberwachung	1164
27.2 Klassen und Attribute für LINQ to SQL	1165
27.2.1 Die Klasse DataContext	1165
27.2.2 Die Klasse Table	1166
27.2.3 Die Klasse EntitySet	1166
27.2.4 Die Struktur EntityRef	1167
27.2.5 Attributklassen	1167
27.3 Manuelles Erstellen der LINQ-to-SQL-Klassen	1169
27.3.1 Erstellen der Entitätsklassen	1170
27.3.2 Erstellen der Klasse MiniLibraryContext	1178
27.3.3 Testen der Applikation	1180
27.4 Verwenden des LINQ to SQL-Designers	1182
27.4.1 Festlegen der Eigenschaften für den DataContext	1183
27.4.2 Einfügen der Entitätsklassen im Designer	1185
27.4.3 Erweitern der Datenklassen mit eigenen Methoden	1189

Inhalt

27.5	Beispiel KnowledgeBase	1190
27.5.1	Die Datenbank	1190
27.5.2	Design der Entitätsklassen (Businessklassen)	1192
27.5.3	Grafiken, Hintergründe und Stile	1195
27.5.4	Fensterdesign	1199
27.5.5	Der Quellcode	1203
27.6	Fazit	1212
28	Das Entity Framework (EF)	1213
28.1	Unterschiede zwischen LINQ to SQL und EF	1214
28.2	Grundlagen zum Entity Framework	1215
28.2.1	Aufbau des Entity Frameworks	1215
28.2.2	Wichtige Klassen des Entity Frameworks	1216
28.2.3	Die Metadaten des Entity Frameworks	1220
28.3	Beispielapplikation KnowledgeBase	1224
28.3.1	Zugriff auf Datenbankdaten	1225
28.3.2	Die Ressourcen der Anwendung	1231
28.3.3	Die Fenster der Anwendung	1234
28.3.4	Die Funktionalität	1239
28.4	Fazit	1247
Anhang	1249
Das MVVM-Pattern	1249
A.1	Genereller Aufbau von MVVM	1250
A.2	Vom Model zum ViewModel	1252
A.2.1	Aufbau des Hauptfensters	1252
A.2.2	Implementieren der Funktionalität	1252
A.3	Implementieren von Funktionalität	1257
A.3.1	Ändern und Überwachen von Daten	1257
A.3.2	Ausführen von Befehlen	1259
A.3.3	Die Implementierung des Beispielprogramms	1261
A.4	Fazit	1270
Index	1271