



Inhaltsverzeichnis

Walter Doberenz, Thomas Gewinnus

Visual C# 2012 - Grundlagen und Profiwissen

ISBN (Buch): 978-3-446-43439-4

Weitere Informationen oder Bestellungen unter

<http://www.hanser-fachbuch.de/978-3-446-43439-4>

sowie im Buchhandel.

Inhaltsverzeichnis

Vorwort	51
----------------------	-----------

Teil I: Grundlagen

1 Einstieg in Visual Studio 2012	57
1.1 Die Installation von Visual Studio 2012	57
1.1.1 Überblick über die Produktpalette	57
1.1.2 Anforderungen an Hard- und Software	59
1.2 Unser allererstes C#-Programm	60
1.2.1 Vorbereitungen	60
1.2.2 Quellcode schreiben	62
1.2.3 Programm kompilieren und testen	63
1.2.4 Einige Erläuterungen zum Quellcode	63
1.2.5 Konsolenanwendungen sind langweilig	64
1.3 Die Windows-Philosophie	65
1.3.1 Mensch-Rechner-Dialog	65
1.3.2 Objekt- und ereignisorientierte Programmierung	65
1.3.3 Programmieren mit Visual Studio 2012	67
1.4 Die Entwicklungsumgebung Visual Studio 2012	68
1.4.1 Neues Projekt	68
1.4.2 Die wichtigsten Fenster	69
1.5 Microsofts .NET-Technologie	72
1.5.1 Zur Geschichte von .NET	72
1.5.2 .NET-Features und Begriffe	75
1.6 Wichtige Neuigkeiten in Visual Studio 2012	82
1.6.1 Die neue Visual Studio 2012 Entwicklungsumgebung	82
1.6.2 Neuheiten im .NET Framework 4.5	85
1.6.3 C# 5.0 – Sprache und Compiler	87

1.7	Praxisbeispiele	87
1.7.1	Windows Forms-Anwendung für Einsteiger	87
1.7.2	Windows-Anwendung für fortgeschrittene Einsteiger	92
2	Grundlagen der Sprache C#	101
2.1	Grundbegriffe	101
2.1.1	Anweisungen	101
2.1.2	Bezeichner	102
2.1.3	Schlüsselwörter	103
2.1.4	Kommentare	103
2.2	Datentypen, Variablen und Konstanten	104
2.2.1	Fundamentale Typen	104
2.2.2	Werttypen versus Verweistypen	105
2.2.3	Benennung von Variablen	106
2.2.4	Deklaration von Variablen	106
2.2.5	Typsuffixe	108
2.2.6	Zeichen und Zeichenketten	109
2.2.7	object-Datentyp	111
2.2.8	Konstanten deklarieren	111
2.2.9	Nullable Types	112
2.2.10	Typinferenz	113
2.2.11	Gültigkeitsbereiche und Sichtbarkeit	114
2.3	Konvertieren von Datentypen	114
2.3.1	Implizite und explizite Konvertierung	114
2.3.2	Welcher Datentyp passt zu welchem?	116
2.3.3	Konvertieren von string	117
2.3.4	Die Convert-Klasse	119
2.3.5	Die Parse-Methode	119
2.3.6	Boxing und Unboxing	120
2.4	Operatoren	121
2.4.1	Arithmetische Operatoren	122
2.4.2	Zuweisungsoperatoren	123
2.4.3	Logische Operatoren	124
2.4.4	Rangfolge der Operatoren	127
2.5	Kontrollstrukturen	128
2.5.1	Verzweigungsbefehle	128
2.5.2	Schleifenanweisungen	131

2.6	Benutzerdefinierte Datentypen	134
2.6.1	Enumerationen	134
2.6.2	Strukturen	135
2.7	Nutzerdefinierte Methoden	137
2.7.1	Methoden mit Rückgabewert	138
2.7.2	Methoden ohne Rückgabewert	139
2.7.3	Parameterübergabe mit ref	140
2.7.4	Parameterübergabe mit out	141
2.7.5	Methodenüberladung	142
2.7.6	Optionale Parameter	143
2.7.7	Benannte Parameter	144
2.8	Praxisbeispiele	145
2.8.1	Vom PAP zur Konsolenanwendung	145
2.8.2	Ein Konsolen- in ein Windows-Programm verwandeln	147
2.8.3	Schleifenanweisungen verstehen	149
2.8.4	Benutzerdefinierte Methoden überladen	151
3	OOP-Konzepte	155
3.1	Kleine Einführung in die OOP	155
3.1.1	Historische Entwicklung	155
3.1.2	Grundbegriffe der OOP	157
3.1.3	Sichtbarkeit von Klassen und ihren Mitgliedern	159
3.1.4	Allgemeiner Aufbau einer Klasse	160
3.1.5	Das Erzeugen eines Objekts	161
3.1.6	Einführungsbeispiel	164
3.2	Eigenschaften	169
3.2.1	Eigenschaften mit Zugriffsmethoden kapseln	169
3.2.2	Berechnete Eigenschaften	171
3.2.3	Lese-/Schreibschutz	173
3.2.4	Property-Accessoren	174
3.2.5	Statische Felder/Eigenschaften	174
3.2.6	Einfache Eigenschaften automatisch implementieren	177
3.3	Methoden	178
3.3.1	Öffentliche und private Methoden	178
3.3.2	Überladene Methoden	179
3.3.3	Statische Methoden	179

3.4	Ereignisse	181
3.4.1	Ereignis hinzufügen	182
3.4.2	Ereignis verwenden	185
3.5	Arbeiten mit Konstruktor und Destruktor	188
3.5.1	Konstruktor und Objektinitialisierer	188
3.5.2	Destruktor und Garbage Collector	191
3.5.3	Mit using den Lebenszyklus des Objekts kapseln	194
3.5.4	Verzögerte Initialisierung	194
3.6	Vererbung und Polymorphie	195
3.6.1	Klassendiagramm	195
3.6.2	Method-Overriding	197
3.6.3	Klassen implementieren	197
3.6.4	Implementieren der Objekte	200
3.6.5	Ausblenden von Mitgliedern durch Vererbung	202
3.6.6	Allgemeine Hinweise und Regeln zur Vererbung	204
3.6.7	Polymorphes Verhalten	205
3.6.8	Die Rolle von System.Object	208
3.7	Spezielle Klassen	209
3.7.1	Abstrakte Klassen	209
3.7.2	Versiegelte Klassen	210
3.7.3	Partielle Klassen	211
3.7.4	Statische Klassen	212
3.8	Schnittstellen (Interfaces)	213
3.8.1	Definition einer Schnittstelle	213
3.8.2	Implementieren einer Schnittstelle	214
3.8.3	Abfragen, ob Schnittstelle vorhanden ist	215
3.8.4	Mehrere Schnittstellen implementieren	215
3.8.5	Schnittstellenprogrammierung ist ein weites Feld	215
3.9	Praxisbeispiele	216
3.9.1	Eigenschaften sinnvoll kapseln	216
3.9.2	Eine statische Klasse anwenden	219
3.9.3	Vom fetten zum schlanken Client	221
3.9.4	Schnittstellenvererbung verstehen	232
4	Arrays, Strings, Funktionen	239
4.1	Datenfelder (Arrays)	239
4.1.1	Array deklarieren	239
4.1.2	Array instanziiieren	240

4.1.3	Array initialisieren	240
4.1.4	Zugriff auf Array-Elemente	241
4.1.5	Zugriff mittels Schleife	242
4.1.6	Mehrdimensionale Arrays	243
4.1.7	Zuweisen von Arrays	245
4.1.8	Arrays aus Strukturvariablen	246
4.1.9	Löschen und Umdimensionieren von Arrays	247
4.1.10	Eigenschaften und Methoden von Arrays	248
4.1.11	Übergabe von Arrays	250
4.2	Verarbeiten von Zeichenketten	251
4.2.1	Zuweisen von Strings	251
4.2.2	Eigenschaften und Methoden von String-Variablen	252
4.2.3	Wichtige Methoden der String-Klasse	254
4.2.4	Die StringBuilder-Klasse	256
4.3	Reguläre Ausdrücke	258
4.3.1	Wozu werden reguläre Ausdrücke verwendet?	259
4.3.2	Eine kleine Einführung	259
4.3.3	Wichtige Methoden/Eigenschaften der Klasse Regex	260
4.3.4	Kompilierte reguläre Ausdrücke	262
4.3.5	RegexOptions-Enumeration	263
4.3.6	Metazeichen (Escape-Zeichen)	263
4.3.7	Zeichenmengen (Character Sets)	264
4.3.8	Quantifizierer	266
4.3.9	Zero-Width Assertions	267
4.3.10	Gruppen	270
4.3.11	Text ersetzen	271
4.3.12	Text splitten	272
4.4	Datums- und Zeitberechnungen	273
4.4.1	Die DateTime-Struktur	273
4.4.2	Wichtige Eigenschaften von DateTime-Variablen	274
4.4.3	Wichtige Methoden von DateTime-Variablen	275
4.4.4	Wichtige Mitglieder der DateTime-Struktur	275
4.4.5	Konvertieren von Datumstrings in DateTime-Werte	276
4.4.6	Die TimeSpan-Struktur	277
4.5	Mathematische Funktionen	278
4.5.1	Überblick	278
4.5.2	Zahlen runden	279
4.5.3	Winkel umrechnen	279

4.5.4	Potenz- und Wurzeloperationen	279
4.5.5	Logarithmus und Exponentialfunktionen	280
4.5.6	Zufallszahlen erzeugen	280
4.6	Zahlen- und Datumsformatierungen	281
4.6.1	Anwenden der ToString-Methode	281
4.6.2	Anwenden der Format-Methode	283
4.7	Praxisbeispiele	285
4.7.1	Zeichenketten verarbeiten	285
4.7.2	Zeichenketten mit StringBuilder addieren	288
4.7.3	Reguläre Ausdrücke testen	291
4.7.4	Methodenaufrufe mit Array-Parametern	293
5	Weitere Sprachfeatures	297
5.1	Namespaces (Namensräume)	297
5.1.1	Ein kleiner Überblick	297
5.1.2	Einen eigenen Namespace einrichten	298
5.1.3	Die using-Anweisung	299
5.1.4	Namespace Alias	300
5.1.5	Namespace Alias Qualifizierer	300
5.2	Operatorenüberladung	301
5.2.1	Syntaxregeln	301
5.2.2	Praktische Anwendung	302
5.3	Collections (Auflistungen)	303
5.3.1	Die Schnittstelle IEnumerable	303
5.3.2	ArrayList	305
5.3.3	Hashtable	307
5.3.4	Indexer	307
5.4	Generics	309
5.4.1	Klassische Vorgehensweise	310
5.4.2	Generics bieten Typsicherheit	311
5.4.3	Generische Methoden	312
5.4.4	Iteratoren	313
5.5	Generische Collections	314
5.5.1	List-Collection statt ArrayList	314
5.5.2	Vorteile generischer Collections	315
5.5.3	Constraints	315

5.6	Das Prinzip der Delegates	315
5.6.1	Delegates sind Methodenzeiger	316
5.6.2	Einen Delegate-Typ deklarieren	316
5.6.3	Ein Delegate-Objekt erzeugen	316
5.6.4	Delegates vereinfacht instanziiieren	318
5.6.5	Anonyme Methoden	319
5.6.6	Lambda-Ausdrücke	320
5.6.7	Lambda-Ausdrücke in der Task Parallel Library	322
5.7	Dynamische Programmierung	324
5.7.1	Wozu dynamische Programmierung?	324
5.7.2	Das Prinzip der dynamischen Programmierung	324
5.7.3	Optionale Parameter sind hilfreich	327
5.7.4	Kovarianz und Kontravarianz	328
5.8	Weitere Datentypen	328
5.8.1	BigInteger	328
5.8.2	Complex	331
5.8.3	Tuple< >	331
5.8.4	SortedSet< >	332
5.9	Praxisbeispiele	333
5.9.1	ArrayList versus generische List	333
5.9.2	Generische IEnumerable-Interfaces implementieren	337
5.9.3	Delegates, anonyme Methoden, Lambda Expressions	340
5.9.4	Dynamischer Zugriff auf COM Interop	344
6	Einführung in LINQ	347
6.1	LINQ-Grundlagen	347
6.1.1	Die LINQ-Architektur	347
6.1.2	Anonyme Typen	348
6.1.3	Erweiterungsmethoden	350
6.2	Abfragen mit LINQ to Objects	351
6.2.1	Grundlegendes zur LINQ-Syntax	351
6.2.2	Zwei alternative Schreibweisen von LINQ Abfragen	352
6.2.3	Übersicht der wichtigsten Abfrage-Operatoren	353
6.3	LINQ-Abfragen im Detail	354
6.3.1	Die Projektionsoperatoren Select und SelectMany	355
6.3.2	Der Restriktionsoperator Where	356
6.3.3	Die Sortierungsoperatoren OrderBy und ThenBy	357
6.3.4	Der Gruppierungsoperator GroupBy	358

6.3.5	Verknüpfen mit Join	361
6.3.6	Aggregat-Operatoren	361
6.3.7	Verzögertes Ausführen von LINQ-Abfragen	363
6.3.8	Konvertierungsmethoden	364
6.3.9	Abfragen mit PLINQ	364
6.4	Praxisbeispiele	367
6.4.1	Die Syntax von LINQ-Abfragen verstehen	367
6.4.2	Aggregat-Abfragen mit LINQ	370

Teil II: Technologien

7	Zugriff auf das Dateisystem	375
7.1	Grundlagen	375
7.1.1	Klassen für den Zugriff auf das Dateisystem	376
7.1.2	Statische versus Instanzen-Klasse	376
7.2	Übersichten	377
7.2.1	Methoden der Directory-Klasse	377
7.2.2	Methoden eines DirectoryInfo-Objekts	378
7.2.3	Eigenschaften eines DirectoryInfo-Objekts	378
7.2.4	Methoden der File-Klasse	378
7.2.5	Methoden eines FileInfo-Objekts	379
7.2.6	Eigenschaften eines FileInfo-Objekts	380
7.3	Operationen auf Verzeichnisebene	380
7.3.1	Existenz eines Verzeichnisses/einer Datei feststellen	380
7.3.2	Verzeichnisse erzeugen und löschen	381
7.3.3	Verzeichnisse verschieben und umbenennen	381
7.3.4	Aktuelles Verzeichnis bestimmen	382
7.3.5	Unterverzeichnisse ermitteln	382
7.3.6	Alle Laufwerke ermitteln	382
7.3.7	Dateien kopieren und verschieben	383
7.3.8	Dateien umbenennen	384
7.3.9	Dateiattribute feststellen	384
7.3.10	Verzeichnis einer Datei ermitteln	386
7.3.11	Alle im Verzeichnis enthaltenen Dateien ermitteln	386
7.3.12	Dateien und Unterverzeichnisse ermitteln	386

7.4	Zugriffsberechtigungen	387
7.4.1	ACL und ACE	387
7.4.2	SetAccessControl-Methode	388
7.4.3	Zugriffsrechte anzeigen	388
7.5	Weitere wichtige Klassen	389
7.5.1	Die Path-Klasse	389
7.5.2	Die Klasse FileSystemWatcher	390
7.6	Datei- und Verzeichnisdialoge	391
7.6.1	OpenFileDialog und SaveFileDialog	392
7.6.2	FolderBrowserDialog	393
7.7	Praxisbeispiele	394
7.7.1	Infos über Verzeichnisse und Dateien gewinnen	394
7.7.2	Eine Verzeichnisstruktur in die TreeView einlesen	398
7.7.3	Mit LINQ und RegEx Verzeichnisbäume durchsuchen	400
8	Dateien lesen und schreiben	405
8.1	Grundprinzip der Datenpersistenz	405
8.1.1	Dateien und Streams	405
8.1.2	Die wichtigsten Klassen	406
8.1.3	Erzeugen eines Streams	407
8.2	Dateiparameter	407
8.2.1	FileAccess	407
8.2.2	FileMode	407
8.2.3	FileShare	408
8.3	Textdateien	408
8.3.1	Eine Textdatei beschreiben bzw. neu anlegen	408
8.3.2	Eine Textdatei lesen	410
8.4	Binärdateien	411
8.4.1	Lese-/Schreibzugriff	411
8.4.2	Die Methoden ReadAllBytes und WriteAllBytes	412
8.4.3	Erzeugen von BinaryReader/BinaryWriter	412
8.5	Sequenzielle Dateien	413
8.5.1	Lesen und schreiben von strukturierten Daten	413
8.5.2	Serialisieren von Objekten	414
8.6	Dateien verschlüsseln und komprimieren	415
8.6.1	Das Methodenpärchen Encrypt/Decrypt	415
8.6.2	Verschlüsseln unter Vista/Windows 7/Windows 8	415

8.6.3	Verschlüsseln mit der CryptoStream-Klasse	416
8.6.4	Dateien komprimieren	417
8.7	Memory Mapped Files	418
8.7.1	Grundprinzip	418
8.7.2	Erzeugen eines MMF	419
8.7.3	Erstellen eines Map View	420
8.8	Praxisbeispiele	421
8.8.1	Auf eine Textdatei zugreifen	421
8.8.2	Einen Objektbaum persistent speichern	424
8.8.3	Ein Memory Mapped File (MMF) verwenden	431
9	Asynchrone Programmierung	435
9.1	Übersicht	435
9.1.1	Multitasking versus Multithreading	436
9.1.2	Deadlocks	437
9.1.3	Racing	437
9.2	Programmieren mit Threads	439
9.2.1	Einführungsbeispiel	439
9.2.2	Wichtige Thread-Methoden	440
9.2.3	Wichtige Thread-Eigenschaften	442
9.2.4	Einsatz der ThreadPool-Klasse	443
9.3	Sperrmechanismen	445
9.3.1	Threading ohne lock	445
9.3.2	Threading mit lock	447
9.3.3	Die Monitor-Klasse	449
9.3.4	Mutex	452
9.3.5	Methoden für die parallele Ausführung sperren	454
9.3.6	Semaphore	454
9.4	Interaktion mit der Programmoberfläche	456
9.4.1	Die Werkzeuge	456
9.4.2	Einzelne Steuerelemente mit Invoke aktualisieren	456
9.4.3	Mehrere Steuerelemente aktualisieren	458
9.4.4	Ist ein Invoke-Aufruf nötig?	458
9.4.5	Und was ist mit WPF?	459
9.5	Timer-Threads	460
9.6	Die BackgroundWorker-Komponente	461

9.7	Asynchrone Programmier-Entwurfsmuster	464
9.7.1	Kurzübersicht	464
9.7.2	Polling	465
9.7.3	Callback verwenden	467
9.7.4	Callback mit Parameterübergabe verwenden	468
9.7.5	Callback mit Zugriff auf die Programm-Oberfläche	469
9.8	Asynchroner Aufruf beliebiger Methoden	470
9.8.1	Die Beispielklasse	470
9.8.2	Asynchroner Aufruf ohne Callback	472
9.8.3	Asynchroner Aufruf mit Callback und Anzeigefunktion	473
9.8.4	Aufruf mit Rückgabewerten (per Eigenschaft)	474
9.8.5	Aufruf mit Rückgabewerten (per EndInvoke)	474
9.9	Es geht auch einfacher – async und await	475
9.9.1	Der Weg von Synchron zu Asynchron	476
9.9.2	Achtung: Fehlerquellen!	478
9.9.3	Eigene asynchrone Methoden entwickeln	480
9.10	Praxisbeispiele	482
9.10.1	Spieltrieb & Multithreading erleben	482
9.10.2	Prozess- und Thread-Informationen gewinnen	495
9.10.3	Ein externes Programm starten	500
10	Die Task Parallel Library	503
10.1	Überblick	503
10.1.1	Parallel-Programmierung	503
10.1.2	Möglichkeiten der TPL	506
10.1.3	Der CLR-Threadpool	506
10.2	Parallele Verarbeitung mit Parallel.Invoke	507
10.2.1	Aufrufvarianten	508
10.2.2	Einschränkungen	509
10.3	Verwendung von Parallel.For	509
10.3.1	Abbrechen der Verarbeitung	511
10.3.2	Auswerten des Verarbeitungsstatus	512
10.3.3	Und was ist mit anderen Iterator-Schrittweiten?	513
10.4	Collections mit Parallel.ForEach verarbeiten	513
10.5	Die Task-Klasse	514
10.5.1	Einen Task erzeugen	514
10.5.2	Den Task starten	515
10.5.3	Datenübergabe an den Task	517

10.5.4	Wie warte ich auf das Ende des Task?	518
10.5.5	Tasks mit Rückgabewerten	520
10.5.6	Die Verarbeitung abbrechen	523
10.5.7	Fehlerbehandlung	526
10.5.8	Weitere Eigenschaften	527
10.6	Zugriff auf das User-Interface	528
10.6.1	Task-Ende und Zugriff auf die Oberfläche	529
10.6.2	Zugriff auf das UI aus dem Task heraus	530
10.7	Weitere Datenstrukturen im Überblick	532
10.7.1	Threadsichere Collections	532
10.7.2	Primitive für die Threadsynchronisation	533
10.8	Parallel LINQ (PLINQ)	533
10.9	Die Parallel Diagnostic Tools	534
10.9.1	Fenster für parallele Aufgaben	534
10.9.2	Fenster für parallele Stacks	535
10.9.3	IntelliTrace	536
10.10	Praxisbeispiel: Spieltrieb – Version 2	536
10.10.1	Aufgabenstellung	536
10.10.2	Global-Klasse	537
10.10.3	Controller-Klasse	538
10.10.4	LKW-Klasse	540
10.10.5	Schiff-Klasse	541
10.10.6	Oberfläche	544
11	Fehlersuche und Behandlung	547
11.1	Der Debugger	547
11.1.1	Allgemeine Beschreibung	547
11.1.2	Die wichtigsten Fenster	548
11.1.3	Debugging-Optionen	551
11.1.4	Praktisches Debugging am Beispiel	553
11.2	Arbeiten mit Debug und Trace	557
11.2.1	Wichtige Methoden von Debug und Trace	557
11.2.2	Besonderheiten der Trace-Klasse	561
11.2.3	TraceListener-Objekte	561
11.3	Caller Information	564
11.3.1	Attribute	564
11.3.2	Anwendung	564

11.4	Fehlerbehandlung	565
11.4.1	Anweisungen zur Fehlerbehandlung	565
11.4.2	try-catch	566
11.4.3	try-finally	571
11.4.4	Das Standardverhalten bei Ausnahmen festlegen	573
11.4.5	Die Exception-Klasse	574
11.4.6	Fehler/Ausnahmen auslösen	575
11.4.7	Eigene Fehlerklassen	575
11.4.8	Exceptionhandling zur Entwurfszeit	577
11.4.9	Code Contracts	578

Teil III: WPF-Anwendungen

12	Einführung in WPF	581
12.1	Neues aus der Gerüchteküche	582
12.1.1	Silverlight	582
12.1.2	WPF	582
12.2	Einführung	583
12.2.1	Was kann eine WPF-Anwendung?	583
12.2.2	Die eXtensible Application Markup Language	585
12.2.3	Verbinden von XAML und C#-Code	589
12.2.4	Zielplattformen	595
12.2.5	Applikationstypen	596
12.2.6	Vor- und Nachteile von WPF-Anwendungen	597
12.2.7	Weitere Dateien im Überblick	597
12.3	Alles beginnt mit dem Layout	600
12.3.1	Allgemeines zum Layout	600
12.3.2	Positionieren von Steuerelementen	602
12.3.3	Canvas	606
12.3.4	StackPanel	606
12.3.5	DockPanel	608
12.3.6	WrapPanel	610
12.3.7	UniformGrid	610
12.3.8	Grid	612
12.3.9	ViewBox	617
12.3.10	TextBlock	618

12.4	Das WPF-Programm	621
12.4.1	Die App-Klasse	621
12.4.2	Das Startobjekt festlegen	622
12.4.3	Kommandozeilenparameter verarbeiten	623
12.4.4	Die Anwendung beenden	624
12.4.5	Auswerten von Anwendungsereignissen	624
12.5	Die Window-Klasse	625
12.5.1	Position und Größe festlegen	625
12.5.2	Rahmen und Beschriftung	626
12.5.3	Das Fenster-Icon ändern	626
12.5.4	Anzeige weiterer Fenster	627
12.5.5	Transparenz	627
12.5.6	Abstand zum Inhalt festlegen	628
12.5.7	Fenster ohne Fokus anzeigen	628
12.5.8	Ereignisfolge bei Fenstern	629
12.5.9	Ein paar Worte zur Schriftdarstellung	629
12.5.10	Ein paar Worte zur Darstellung von Controls	632
12.5.11	Wird mein Fenster komplett mit WPF gerendert?	634
13	Übersicht WPF-Controls	635
13.1	Allgemeingültige Eigenschaften	635
13.2	Label	637
13.3	Button, RepeatButton, ToggleButton	638
13.3.1	Schaltflächen für modale Dialoge	638
13.3.2	Schaltflächen mit Grafik	639
13.4	TextBox, PasswordBox	640
13.4.1	TextBox	640
13.4.2	PasswordBox	642
13.5	CheckBox	643
13.6	RadioButton	645
13.7	ListBox, ComboBox	646
13.7.1	ListBox	646
13.7.2	ComboBox	649
13.7.3	Den Content formatieren	651
13.8	Image	652
13.8.1	Grafik per XAML zuweisen	652
13.8.2	Grafik zur Laufzeit zuweisen	653
13.8.3	Bild aus Datei laden	654

13.8.4	Die Grafiskalierung beeinflussen	655
13.9	MediaElement	656
13.10	Slider, ScrollBar	658
13.10.1	Slider	658
13.10.2	ScrollBar	659
13.11	ScrollView	660
13.12	Menu, ContextMenu	661
13.12.1	Menu	661
13.12.2	Tastenkürzel	662
13.12.3	Grafiken	663
13.12.4	Weitere Möglichkeiten	664
13.12.5	ContextMenu	665
13.13	ToolBar	666
13.14	StatusBar, ProgressBar	669
13.14.1	StatusBar	669
13.14.2	ProgressBar	671
13.15	Border, GroupBox, BulletDecorator	672
13.15.1	Border	672
13.15.2	GroupBox	673
13.15.3	BulletDecorator	674
13.16	RichTextBox	676
13.16.1	Verwendung und Anzeige von vordefiniertem Text	676
13.16.2	Neues Dokument zur Laufzeit erzeugen	678
13.16.3	Sichern von Dokumenten	678
13.16.4	Laden von Dokumenten	680
13.16.5	Texte per Code einfügen/modifizieren	681
13.16.6	Texte formatieren	682
13.16.7	EditingCommands	683
13.16.8	Grafiken/Objekte einfügen	684
13.16.9	Rechtschreibkontrolle	686
13.17	FlowDocumentPageViewer & Co.	686
13.17.1	FlowDocumentPageViewer	686
13.17.2	FlowDocumentReader	687
13.17.3	FlowDocumentScrollViewer	687
13.18	FlowDocument	687
13.18.1	FlowDocument per XAML beschreiben	688
13.18.2	FlowDocument per Code erstellen	690
13.19	DocumentViewer	691

13.20	Expander, TabControl	692
13.20.1	Expander	692
13.20.2	TabControl	694
13.21	Popup	695
13.22	TreeView	697
13.23	ListView	700
13.24	DataGrid	701
13.25	Calendar/DatePicker	702
13.26	InkCanvas	706
13.26.1	Stift-Parameter definieren	706
13.26.2	Die Zeichenmodi	707
13.26.3	Inhalte laden und sichern	708
13.26.4	Konvertieren in eine Bitmap	708
13.26.5	Weitere Eigenschaften	709
13.27	Ellipse, Rectangle, Line und Co.	710
13.27.1	Ellipse	710
13.27.2	Rectangle	710
13.27.3	Line	711
13.28	Browser	711
13.29	Ribbon	714
13.29.1	Allgemeine Grundlagen	714
13.29.2	Download/Installation	715
13.29.3	Erste Schritte	716
13.29.4	Registerkarten und Gruppen	717
13.29.5	Kontextabhängige Registerkarten	718
13.29.6	Einfache Beschriftungen	719
13.29.7	Schaltflächen	719
13.29.8	Auswahllisten	721
13.29.9	Optionsauswahl	724
13.29.10	Texteingaben	724
13.29.11	Screen tips	724
13.29.12	Symbolleiste für den Schnellzugriff	725
13.29.13	Das RibbonWindow	726
13.29.14	Menüs	727
13.29.15	Anwendungsmenü	729
13.29.16	Alternativen	732
13.30	Chart	732
13.31	WindowsFormsHost	733

14	Wichtige WPF-Techniken	737
14.1	Eigenschaften	737
14.1.1	Abhängige Eigenschaften (Dependency Properties)	737
14.1.2	Angehängte Eigenschaften (Attached Properties)	739
14.2	Einsatz von Ressourcen	739
14.2.1	Was sind eigentlich Ressourcen?	739
14.2.2	Wo können Ressourcen gespeichert werden?	739
14.2.3	Wie definiere ich eine Ressource?	741
14.2.4	Statische und dynamische Ressourcen	742
14.2.5	Wie werden Ressourcen adressiert?	743
14.2.6	System-Ressourcen einbinden	744
14.3	Das WPF-Ereignis-Modell	744
14.3.1	Einführung	744
14.3.2	Routed Events	745
14.3.3	Direkte Events	747
14.4	Verwendung von Commands	748
14.4.1	Einführung zu Commands	748
14.4.2	Verwendung vordefinierter Commands	748
14.4.3	Das Ziel des Commands	750
14.4.4	Welche vordefinierten Commands stehen zur Verfügung?	751
14.4.5	Commands an Ereignismethoden binden	752
14.4.6	Wie kann ich ein Command per Code auslösen?	753
14.4.7	Command-Ausführung verhindern	754
14.5	Das WPF-Style-System	754
14.5.1	Übersicht	754
14.5.2	Benannte Styles	755
14.5.3	Typ-Styles	756
14.5.4	Styles anpassen und vererben	757
14.6	Verwenden von Triggern	760
14.6.1	Eigenschaften-Trigger (Property Triggers)	760
14.6.2	Ereignis-Trigger	762
14.6.3	Daten-Trigger	763
14.7	Einsatz von Templates	764
14.7.1	Neues Template erstellen	764
14.7.2	Template abrufen und verändern	768
14.8	Transformationen, Animationen, StoryBoards	771
14.8.1	Transformationen	771
14.8.2	Animationen mit dem StoryBoard realisieren	777

14.9	Praxisbeispiel	781
14.9.1	Arbeiten mit Microsoft Expression Blend	781
15	WPF-Datenbindung	787
15.1	Grundprinzip	787
15.1.1	Bindungsarten	788
15.1.2	Wann eigentlich wird die Quelle aktualisiert?	789
15.1.3	Geht es auch etwas langsamer?	790
15.1.4	Bindung zur Laufzeit realisieren	791
15.2	Binden an Objekte	793
15.2.1	Objekte im XAML-Code instanziiieren	793
15.2.2	Verwenden der Instanz im C#-Quellcode	795
15.2.3	Anforderungen an die Quell-Klasse	795
15.2.4	Instanziiieren von Objekten per C#-Code	797
15.3	Binden von Collections	798
15.3.1	Anforderung an die Collection	798
15.3.2	Einfache Anzeige	799
15.3.3	Navigieren zwischen den Objekten	800
15.3.4	Einfache Anzeige in einer ListBox	802
15.3.5	DataTemplates zur Anzeigeformatierung	803
15.3.6	Mehr zu List- und ComboBox	804
15.3.7	Verwendung der ListView	806
15.4	Noch einmal zurück zu den Details	808
15.4.1	Navigieren in den Daten	808
15.4.2	Sortieren	810
15.4.3	Filtern	810
15.4.4	Live Shaping	811
15.5	Anzeige von Datenbankinhalten	812
15.5.1	Datenmodell per LINQ to SQL-Designer erzeugen	813
15.5.2	Die Programm-Oberfläche	814
15.5.3	Der Zugriff auf die Daten	815
15.6	Drag & Drop-Datenbindung	816
15.6.1	Vorgehensweise	816
15.6.2	Weitere Möglichkeiten	819
15.7	Formatieren von Werten	820
15.7.1	IValueConverter	821
15.7.2	BindingBase.StringFormat-Eigenschaft	823
15.8	Das DataGrid als Universalwerkzeug	825

15.8.1	Grundlagen der Anzeige	825
15.8.2	UI-Virtualisierung	826
15.8.3	Spalten selbst definieren	826
15.8.4	Zusatzinformationen in den Zeilen anzeigen	828
15.8.5	Vom Betrachten zum Editieren	829
15.9	Praxisbeispiele	830
15.9.1	Collections in Hintergrundthreads füllen	830
15.9.2	Drag & Drop-Bindung bei 1:n-Beziehungen	833
16	Druckausgabe mit WPF	839
16.1	Grundlagen	839
16.1.1	XPS-Dokumente	839
16.1.2	System.Printing	840
16.1.3	System.Windows.Xps	841
16.2	Einfache Druckausgaben mit dem PrintDialog	841
16.3	Mehrseitige Druckvorschau-Funktion	844
16.3.1	Fix-Dokumente	844
16.3.2	Flow-Dokumente	850
16.4	Druckerinfos, -auswahl, -konfiguration	853
16.4.1	Die installierten Drucker bestimmen	854
16.4.2	Den Standarddrucker bestimmen	855
16.4.3	Mehr über einzelne Drucker erfahren	855
16.4.4	Spezifische Druckeinstellungen vornehmen	857
16.4.5	Direkte Druckausgabe	859

Teil IV: Windows Store Apps

17	Erste Schritte in WinRT	863
17.1	Grundkonzepte und Begriffe	863
17.1.1	Windows Runtime (WinRT)	863
17.1.2	Windows Store Apps	864
17.1.3	Fast and Fluid	865
17.1.4	Process Sandboxing und Contracts	866
17.1.5	.NET WinRT-Profil	868
17.1.6	Language Projection	868
17.1.7	Vollbildmodus	870

17.1.8	Windows Store	870
17.1.9	Zielpattformen	871
17.2	Entwurfsumgebung	872
17.2.1	Betriebssystem	872
17.2.2	Windows-Simulator	873
17.2.3	Remote-Debugging	875
17.3	Ein (kleines) Einstiegsbeispiel	876
17.3.1	Aufgabenstellung	876
17.3.2	Quellcode	876
17.3.3	Oberflächenentwurf	879
17.3.4	Installation und Test	881
17.3.5	Verbesserungen	882
17.3.6	Fazit	885
17.4	Weitere Details zu WinRT	887
17.4.1	Wo ist WinRT einzuordnen?	887
17.4.2	Die WinRT-API	888
17.4.3	Wichtige WinRT-Namespaces	890
17.4.4	Der Unterbau	891
17.5	Gedanken zum Thema "WinRT & Tablets"	894
17.5.1	Windows 8-Oberfläche versus Desktop	894
17.5.2	Tablets und Touchscreens	894
17.6	Praxisbeispiel	896
17.6.1	WinRT in Desktop-Applikationen nutzen	896
18	WinRT-Oberflächen entwerfen	899
18.1	Grundkonzepte	899
18.1.1	XAML (oder HTML 5) für die Oberfläche	900
18.1.2	Die Page, der Frame und das Window	901
18.1.3	Das Befehlsdesign	902
18.1.4	Die Navigationsdesigns	904
18.1.5	Achtung: Fingereingabe!	905
18.1.6	Verwendung von Schriftarten	906
18.2	Projekttypen und Seitentemplates	906
18.2.1	Leere App	906
18.2.2	Geteilte App (Split App)	908
18.2.3	Raster-App (Grid App)	910
18.2.4	Leere Seite (Blank Page)	911

18.2.5	Standardseite (Basic Page)	912
18.2.6	Ein eigenes Grundlayout erstellen	914
18.3	Seitenauswahl und -navigation	915
18.3.1	Die Startseite festlegen	915
18.3.2	Navigation und Parameterübergabe	915
18.3.3	Den Seitenstatus erhalten	916
18.4	Die vier App-Ansichten	917
18.4.1	Vollbild quer und hochkant	917
18.4.2	Angedockt und Füllmodus	918
18.4.3	Reagieren auf die Änderung	918
18.4.4	Angedockten Modus aktiv beenden	922
18.5	Skalieren von Apps	922
18.6	Praxisbeispiele	924
18.6.1	Seitennavigation und Parameterübergabe	924
18.6.2	Auf Ansichtsänderungen reagieren	926
18.7	Tipps & Tricks	930
18.7.1	Symbole für WinRT-Oberflächen finden	930
18.7.2	Wie werde ich das Groufi-Layout schnell los?	931
19	Die wichtigsten Controls	933
19.1	Einfache WinRT-Controls	933
19.1.1	TextBlock, RichTextBlock	933
19.1.2	Button, HyperlinkButton, RepeatButton	936
19.1.3	CheckBox, RadioButton, ToggleButton, ToggleSwitch	938
19.1.4	TextBox, PasswordBox, RichEditBox	939
19.1.5	Image	943
19.1.6	ScrollBar, Slider, ProgressBar, ProgressRing	945
19.1.7	Border, Ellipse, Rectangle	946
19.2	Layout-Controls	947
19.2.1	Canvas	947
19.2.2	StackPanel	948
19.2.3	ScrollView	948
19.2.4	Grid	949
19.2.5	VariableSizedWrapGrid	950
19.3	Listendarstellungen	951
19.3.1	ComboBox, ListBox	951
19.3.2	ListView	955

19.3.3	GridView	956
19.3.4	FlipView	959
19.4	Sonstige Controls	961
19.4.1	CaptureElement	961
19.4.2	MediaElement	962
19.4.3	Frame	964
19.4.4	WebView	964
19.4.5	ToolTip	965
19.5	Praxisbeispiele	967
19.5.1	Einen StringFormat-Konverter implementieren	967
19.5.2	Besonderheiten der TextBox kennen lernen	969
19.5.3	Daten in der GridView gruppieren	972
19.5.4	Das SemanticZoom-Control verwenden	977
19.5.5	Die CollectionViewSource verwenden	981
19.5.6	Zusammenspiel ListBox/AppBar	985
19.5.7	Musikwiedergabe im Hintergrund realisieren	988
20	Apps im Detail	993
20.1	Ein Windows Store App-Projekt im Detail	993
20.1.1	Contracts und Extensions	994
20.1.2	AssemblyInfo.cs	995
20.1.3	Verweise	996
20.1.4	App.xaml und App.xaml.cs	997
20.1.5	Package.appxmanifest	998
20.1.6	Application1_TemporaryKey.pfx	1002
20.1.7	MainPage.xaml & MainPage.xaml.cs	1002
20.1.8	Datentyp-Konverter/Hilfsklassen	1003
20.1.9	StandardStyles.xaml	1005
20.1.10	Assets/Symbole	1006
20.1.11	Nach dem Kompilieren	1007
20.2	Der Lebenszyklus einer WinRT-App	1007
20.2.1	Möglichkeiten der Aktivierung von Apps	1009
20.2.2	Der Splash Screen	1011
20.2.3	Suspending	1011
20.2.4	Resuming	1012
20.2.5	Beenden von Apps	1013
20.2.6	Die Ausnahmen von der Regel	1014
20.2.7	Debuggen	1014

20.3	Daten speichern und laden	1018
20.3.1	Grundsätzliche Überlegungen	1018
20.3.2	Worauf und wie kann ich zugreifen?	1018
20.3.3	Das AppData-Verzeichnis	1018
20.3.4	Das Anwendungs-Installationsverzeichnis	1021
20.3.5	Das Downloads-Verzeichnis	1022
20.3.6	Sonstige Verzeichnisse	1023
20.3.7	Anwendungsdaten lokal sichern und laden	1023
20.3.8	Daten in der Cloud ablegen/laden (Roaming)	1025
20.3.9	Aufräumen	1028
20.3.10	Sensible Informationen speichern	1028
20.4	Praxisbeispiele	1030
20.4.1	Unterstützung für den Search-Contract bieten	1030
20.4.2	Die Auto-Play-Funktion unterstützen	1037
20.4.3	Einen zusätzlichen Splash Screen einsetzen	1041
20.4.4	Eine Dateiverknüpfung erstellen	1043
21	WinRT-Techniken	1049
21.1	Arbeiten mit Dateien/Verzeichnissen	1049
21.1.1	Verzeichnisinformationen auflisten	1049
21.1.2	Unterverzeichnisse auflisten	1052
21.1.3	Verzeichnisse erstellen/löschen	1053
21.1.4	Dateien auflisten	1055
21.1.5	Dateien erstellen/schreiben/lesen	1057
21.1.6	Dateien kopieren/umbenennen/löschen	1061
21.1.7	Verwenden der Dateipicker	1062
21.1.8	StorageFile-/StorageFolder-Objekte speichern	1069
21.1.9	Verwenden der Most Recently Used-Liste	1071
21.2	Datenaustausch zwischen Apps/Programmen	1072
21.2.1	Zwischenablage	1073
21.2.2	Teilen von Inhalten	1080
21.2.3	Eine App als Freigabeziel verwenden	1083
21.2.4	Zugriff auf die Kontaktliste	1084
21.3	Spezielle Oberflächenelemente	1086
21.3.1	MessageDialog	1086
21.3.2	Popup-Benachrichtigungen	1089
21.3.3	PopUp/Flyouts	1097

21.3.4	Das PopupMenu einsetzen	1101
21.3.5	Eine AppBar verwenden	1103
21.4	Datenbanken und Windows Store Apps	1108
21.4.1	Der Retter in der Not: SQLite!	1108
21.4.2	Verwendung/Kurzüberblick	1108
21.4.3	Installation	1110
21.4.4	Wie kommen wir zu einer neuen Datenbank?	1112
21.4.5	Wie werden die Daten manipuliert?	1116
21.5	Vertrieb der App	1118
21.5.1	Verpacken der App	1118
21.5.2	Windows App Certification Kit	1120
21.5.3	App-Installation per Skript	1122
21.6	Ein Blick auf die App-Schwachstellen	1123
21.6.1	Quellcodes im Installationsverzeichnis	1123
21.6.2	Zugriff auf den App-Datenordner	1125
21.7	Praxisbeispiele	1125
21.7.1	Ein Verzeichnis auf Änderungen überwachen	1125
21.7.2	Eine App als Freigabeziel verwenden	1128
21.7.3	ToastNotifications einfach erzeugen	1133

Anhang

A	Glossar	1141
B	Wichtige Dateiextensions	1147
	Index	1149

Bonuskapitel im E-Book

Zweites Vorwort	1193
 Teil V: Weitere Technologien	
22 XML in Theorie und Praxis	1197
22.1 XML – etwas Theorie	1197
22.1.1 Übersicht	1197
22.1.2 Der XML-Grundaufbau	1200
22.1.3 Wohlgeformte Dokumente	1201
22.1.4 Processing Instructions (PI)	1204
22.1.5 Elemente und Attribute	1204
22.1.6 Verwendbare Zeichensätze	1206
22.2 XSD-Schemas	1208
22.2.1 XSD-Schemas und ADO.NET	1208
22.2.2 XML-Schemas in Visual Studio analysieren	1210
22.2.3 XML-Datei mit XSD-Schema erzeugen	1213
22.2.4 XSD-Schema aus einer XML-Datei erzeugen	1214
22.3 Verwendung des DOM unter .NET	1215
22.3.1 Übersicht	1215
22.3.2 DOM-Integration in C#	1216
22.3.3 Laden von Dokumenten	1216
22.3.4 Erzeugen von XML-Dokumenten	1217
22.3.5 Auslesen von XML-Dateien	1219
22.3.6 Direktzugriff auf einzelne Elemente	1221
22.3.7 Einfügen von Informationen	1221
22.3.8 Suchen in den Baumzweigen	1224
22.4 XML-Verarbeitung mit LINQ to XML	1227
22.4.1 Die LINQ to XML-API	1227
22.4.2 Neue XML-Dokumente erzeugen	1229

22.4.3	Laden und Sichern von XML-Dokumenten	1231
22.4.4	Navigieren in XML-Daten	1232
22.4.5	Auswählen und Filtern	1234
22.4.6	Manipulieren der XML-Daten	1234
22.4.7	XML-Dokumente transformieren	1236
22.5	Weitere Möglichkeiten der XML-Verarbeitung	1239
22.5.1	Die relationale Sicht mit XmlDocument	1239
22.5.2	XML-Daten aus Objektstrukturen erzeugen	1242
22.5.3	Schnelles Suchen in XML-Daten mit XPathNavigator	1245
22.5.4	Schnelles Auslesen von XML-Daten mit XmlReader	1248
22.5.5	Erzeugen von XML-Daten mit XmlWriter	1250
22.5.6	XML transformieren mit XSLT	1252
22.6	Praxisbeispiele	1254
22.6.1	Mit dem DOM in XML-Dokumenten navigieren	1254
22.6.2	XML-Daten in eine TreeView einlesen	1257
23	Einführung in ADO.NET	1261
23.1	Eine kleine Übersicht	1261
23.1.1	Die ADO.NET-Klassenhierarchie	1261
23.1.2	Die Klassen der Datenprovider	1262
23.1.3	Das Zusammenspiel der ADO.NET-Klassen	1265
23.2	Das Connection-Objekt	1266
23.2.1	Allgemeiner Aufbau	1266
23.2.2	OleDbConnection	1266
23.2.3	Schließen einer Verbindung	1268
23.2.4	Eigenschaften des Connection-Objekts	1268
23.2.5	Methoden des Connection-Objekts	1270
23.2.6	DerConnectionStringBuilder	1271
23.3	Das Command-Objekt	1271
23.3.1	Erzeugen und Anwenden eines Command-Objekts	1272
23.3.2	Erzeugen mittels CreateCommand-Methode	1272
23.3.3	Eigenschaften des Command-Objekts	1273
23.3.4	Methoden des Command-Objekts	1275
23.3.5	Freigabe von Connection- und Command-Objekten	1276
23.4	Parameter-Objekte	1277
23.4.1	Erzeugen und Anwenden eines Parameter-Objekts	1277
23.4.2	Eigenschaften des Parameter-Objekts	1278

23.5	Das CommandBuilder-Objekt	1279
23.5.1	Erzeugen	1279
23.5.2	Anwenden	1279
23.6	Das DataReader-Objekt	1280
23.6.1	DataReader erzeugen	1280
23.6.2	Daten lesen	1281
23.6.3	Eigenschaften des DataReaders	1282
23.6.4	Methoden des DataReaders	1282
23.7	Das DataAdapter-Objekt	1283
23.7.1	DataAdapter erzeugen	1283
23.7.2	Command-Eigenschaften	1284
23.7.3	Fill-Methode	1285
23.7.4	Update-Methode	1286
23.8	Praxisbeispiele	1287
23.8.1	Wichtige ADO.NET-Objekte im Einsatz	1287
23.8.2	Eine Aktionsabfrage ausführen	1288
23.8.3	Eine Auswahlabfrage aufrufen	1291
23.8.4	Die Datenbank aktualisieren	1293
24	Das DataSet	1297
24.1	Grundlegende Features des DataSets	1297
24.1.1	Die Objekthierarchie	1298
24.1.2	Die wichtigsten Klassen	1298
24.1.3	Erzeugen eines DataSets	1299
24.2	Das DataTable-Objekt	1300
24.2.1	DataTable erzeugen	1301
24.2.2	Spalten hinzufügen	1301
24.2.3	Zeilen zur DataTable hinzufügen	1302
24.2.4	Auf den Inhalt einer DataTable zugreifen	1303
24.3	Die DataView	1305
24.3.1	Erzeugen einer DataView	1305
24.3.2	Sortieren und Filtern von Datensätzen	1305
24.3.3	Suchen von Datensätzen	1306
24.4	Typisierte DataSets	1306
24.4.1	Ein typisiertes DataSet erzeugen	1307
24.4.2	Das Konzept der Datenquellen	1308
24.4.3	Typisierte DataSets und TableAdapter	1309

24.5	Die Qual der Wahl	1310
24.5.1	DataReader – der schnelle Lesezugriff	1311
24.5.2	DataSet – die Datenbank im Hauptspeicher	1311
24.5.3	Objektrelationales Mapping – die Zukunft?	1312
24.6	Praxisbeispiele	1313
24.6.1	In der DataView sortieren und filtern	1313
24.6.2	Suche nach Datensätzen	1315
24.6.3	Ein DataSet in einen XML-String serialisieren	1316
24.6.4	Untypisiertes in ein typisiertes DataSet konvertieren	1321
24.6.5	Eine LINQ to SQL-Abfrage ausführen	1326
25	OOP-Spezial	1331
25.1	Eine kleine Einführung in die UML	1331
25.1.1	Use Case-Diagramm	1331
25.1.2	Use Case-Dokumentation	1333
25.1.3	Objekte identifizieren	1334
25.1.4	Statisches Modell	1335
25.1.5	Beziehungen zwischen den Klassen	1336
25.1.6	Dynamisches Modell	1336
25.1.7	Implementierung	1337
25.1.8	Test-Client	1341
25.2	Der Klassen-Designer	1344
25.2.1	Ein neues Klassendiagramm erzeugen	1345
25.2.2	Werkzeugkasten	1346
25.2.3	Enumeration	1347
25.2.4	Klasse	1349
25.2.5	Struktur	1351
25.2.6	Abstrakte Klasse	1351
25.2.7	Schnittstelle	1353
25.2.8	Delegate	1355
25.2.9	Zuordnung	1357
25.2.10	Vererbung	1358
25.2.11	Diagramme anpassen	1358
25.2.12	Wann lohnt sich der Klassen-Designer?	1359
25.3	Praxisbeispiele	1359
25.3.1	Implementierung einer Finite State Machine	1359
25.3.2	Modellierung des Bestellsystems einer Firma	1365

26	Das Microsoft Event Pattern	1379
26.1	Einführung in Design Pattern	1379
26.2	Aufbau und Bedeutung des Observer Pattern	1380
26.2.1	Subjekt und Observer	1380
26.2.2	Sequenzdiagramme	1382
26.2.3	Die Registration-Sequenz	1382
26.2.4	Die Notification-Sequenz	1383
26.2.5	Die Unregistration-Sequenz	1383
26.2.6	Bedeutung der Sequenzen für das Geschäftsmodell	1384
26.2.7	Die Rolle des Containers	1384
26.3	Implementieren mit Interfaces und Callbacks	1385
26.3.1	Übersicht und Klassendiagramm	1385
26.3.2	Die Schnittstellen IObserver und IObservable	1387
26.3.3	Die Abstrakte Klasse Subject	1387
26.3.4	Observer1	1388
26.3.5	Observer2	1389
26.3.6	Model	1390
26.3.7	Form1	1391
26.3.8	Ein zweites Klassendiagramm	1392
26.3.9	Testen	1393
26.4	Implementierung mit Delegates und Events	1394
26.4.1	Multicast-Events	1395
26.4.2	IObserver, IObservable und Subject	1395
26.4.3	Observer1 und Observer2	1396
26.4.4	Model	1396
26.4.5	Form1	1397
26.4.6	Test und Vergleich	1397
26.4.7	Klassendiagramm	1398
26.5	Implementierung des Microsoft Event Pattern	1399
26.5.1	Namensgebung für Ereignisse	1399
26.5.2	Namensgebung und Signatur der Delegates	1399
26.5.3	Hinzufügen einer das Ereignis auslösenden Methode	1400
26.5.4	Model	1401
26.5.5	Observer1	1402
26.5.6	Form1	1403
26.6	Test und Vergleich	1403
26.7	Klassendiagramm	1403

26.8	Implementierung eines Event Pattern	1404
26.9	Praxisbeispiel	1406
26.9.1	Objekte beobachten sich gegenseitig	1406
27	Verteilen von Anwendungen	1417
27.1	ClickOnce-Deployment	1418
27.1.1	Übersicht/Einschränkungen	1418
27.1.2	Die Vorgehensweise	1419
27.1.3	Ort der Veröffentlichung	1419
27.1.4	Anwendungsdateien	1420
27.1.5	Erforderliche Komponenten	1420
27.1.6	Aktualisierungen	1421
27.1.7	Veröffentlichungsoptionen	1422
27.1.8	Veröffentlichen	1423
27.1.9	Verzeichnisstruktur	1423
27.1.10	Der Webpublishing-Assistent	1425
27.1.11	Neue Versionen erstellen	1426
27.2	InstallShield	1426
27.2.1	Installation	1426
27.2.2	Aktivieren	1427
27.2.3	Ein neues Setup-Projekt	1427
27.2.4	Finaler Test	1435
28	Weitere Techniken	1437
28.1	Zugriff auf die Zwischenablage	1437
28.1.1	Das Clipboard-Objekt	1437
28.1.2	Zwischenablage-Funktionen für Textboxen	1439
28.2	Arbeiten mit der Registry	1439
28.2.1	Allgemeines	1440
28.2.2	Registry-Unterstützung in .NET	1442
28.3	.NET-Reflection	1443
28.3.1	Übersicht	1443
28.3.2	Assembly laden	1443
28.3.3	Mittels GetType und Type Informationen sammeln	1444
28.3.4	Dynamisches Laden von Assemblies	1446
28.4	Das SerialPort-Control	1449
28.4.1	Übersicht	1449
28.4.2	Einführungsbeispiele	1450

28.4.3	Thread-Probleme bei Windows Forms-Anwendungen	1453
28.4.4	Ein einfaches Terminalprogramm	1456
28.5	Praxisbeispiele	1461
28.5.1	Zugriff auf die Registry	1461
28.5.2	Dateiverknüpfungen erzeugen	1463
28.5.3	Betrachter für Manifestressourcen	1465
28.5.4	Ressourcen mit Reflection auslesen	1468
29	Konsolenanwendungen	1471
29.1	Grundaufbau/Konzepte	1471
29.1.1	Unser Hauptprogramm – Program.cs	1472
29.1.2	Rückgabe eines Fehlerstatus	1473
29.1.3	Parameterübergabe	1474
29.1.4	Zugriff auf die Umgebungsvariablen	1475
29.2	Die Kommandozentrale: System.Console	1476
29.2.1	Eigenschaften	1477
29.2.2	Methoden/Ereignisse	1477
29.2.3	Textausgaben	1478
29.2.4	Farbangaben	1479
29.2.5	Tastaturabfragen	1480
29.2.6	Arbeiten mit Streamdaten	1481
29.3	Praxisbeispiel	1483
29.3.1	Farbige Konsolenanwendung	1483
29.3.2	Weitere Hinweise und Beispiele	1485

Teil VI: Windows Forms

30	Windows Forms-Anwendungen	1489
30.1	Grundaufbau/Konzepte	1489
30.1.1	Das Hauptprogramm – Program.cs	1490
30.1.2	Die Oberflächendefinition – Form1.Designer.cs	1494
30.1.3	Die Spielwiese des Programmierers – Form1.cs	1495
30.1.4	Die Datei AssemblyInfo.cs	1496
30.1.5	Resources.resx/Resources.Designer.cs	1497
30.1.6	Settings.settings/Settings.Designer.cs	1498
30.1.7	Settings.cs	1499

30.2	Ein Blick auf die Application-Klasse	1500
30.2.1	Eigenschaften	1500
30.2.2	Methoden	1501
30.2.3	Ereignisse	1503
30.3	Allgemeine Eigenschaften von Komponenten	1503
30.3.1	Font	1504
30.3.2	Handle	1506
30.3.3	Tag	1507
30.3.4	Modifiers	1507
30.4	Allgemeine Ereignisse von Komponenten	1508
30.4.1	Die Eventhandler-Argumente	1508
30.4.2	Sender	1508
30.4.3	Der Parameter e	1510
30.4.4	Mausereignisse	1510
30.4.5	KeyPreview	1512
30.4.6	Weitere Ereignisse	1513
30.4.7	Validitätsprüfungen	1513
30.4.8	SendKeys	1514
30.5	Allgemeine Methoden von Komponenten	1515
31	Windows Forms-Formulare	1517
31.1	Übersicht	1517
31.1.1	Wichtige Eigenschaften des Form-Objekts	1518
31.1.2	Wichtige Ereignisse des Form-Objekts	1520
31.1.3	Wichtige Methoden des Form-Objekts	1521
31.2	Praktische Aufgabenstellungen	1522
31.2.1	Fenster anzeigen	1522
31.2.2	Splash Screens beim Anwendungsstart anzeigen	1525
31.2.3	Eine Sicherheitsabfrage vor dem Schließen anzeigen	1528
31.2.4	Ein Formular durchsichtig machen	1528
31.2.5	Die Tabulatorreihenfolge festlegen	1529
31.2.6	Ausrichten von Komponenten im Formular	1530
31.2.7	Spezielle Panels für flexible Layouts	1532
31.2.8	Menüs erzeugen	1534
31.3	MDI-Anwendungen	1538
31.3.1	"Falsche" MDI-Fenster bzw. Verwenden von Parent	1538
31.3.2	Die echten MDI-Fenster	1539
31.3.3	Die Kindfenster	1540

31.3.4	Automatisches Anordnen der Kindfenster	1541
31.3.5	Zugriff auf die geöffneten MDI-Kindfenster	1542
31.3.6	Zugriff auf das aktive MDI-Kindfenster	1543
31.3.7	Mischen von Kindfenstermenü/MDIContainer-Menü	1543
31.4	Praxisbeispiele	1544
31.4.1	Informationsaustausch zwischen Formularen	1544
31.4.2	Ereigniskette beim Laden/Entladen eines Formulars	1551
32	Windows Forms-Komponenten	1557
32.1	Allgemeine Hinweise	1557
32.1.1	Hinzufügen von Komponenten	1557
32.1.2	Komponenten zur Laufzeit per Code erzeugen	1558
32.2	Allgemeine Steuerelemente	1560
32.2.1	Label	1560
32.2.2	LinkLabel	1561
32.2.3	Button	1562
32.2.4	TextBox	1563
32.2.5	MaskedTextBox	1566
32.2.6	CheckBox	1567
32.2.7	RadioButton	1569
32.2.8	ListBox	1569
32.2.9	CheckedListBox	1571
32.2.10	ComboBox	1571
32.2.11	PictureBox	1572
32.2.12	DateTimePicker	1573
32.2.13	MonthCalendar	1573
32.2.14	HScrollBar, VScrollBar	1574
32.2.15	TrackBar	1575
32.2.16	NumericUpDown	1575
32.2.17	DomainUpDown	1576
32.2.18	ProgressBar	1576
32.2.19	RichTextBox	1577
32.2.20	ListView	1578
32.2.21	TreeView	1584
32.2.22	WebBrowser	1589
32.3	Container	1590
32.3.1	FlowLayout/TableLayout/SplitContainer	1590
32.3.2	Panel	1590

32.3.3	GroupBox	1591
32.3.4	TabControl	1592
32.3.5	ImageList	1594
32.4	Menüs & Symbolleisten	1595
32.4.1	MenuStrip und ContextMenuStrip	1595
32.4.2	ToolStrip	1595
32.4.3	StatusStrip	1595
32.4.4	ToolStripContainer	1596
32.5	Daten	1596
32.5.1	DataSet	1596
32.5.2	DataGridView/DataGrid	1597
32.5.3	BindingNavigator/BindingSource	1597
32.5.4	Chart	1597
32.6	Komponenten	1599
32.6.1	ErrorProvider	1599
32.6.2	HelpProvider	1599
32.6.3	ToolTip	1599
32.6.4	BackgroundWorker	1599
32.6.5	Timer	1599
32.6.6	SerialPort	1600
32.7	Drucken	1600
32.7.1	PrintPreviewControl	1600
32.7.2	PrintDocument	1600
32.8	Dialoge	1600
32.8.1	OpenFileDialog/SaveFileDialog/FolderBrowserDialog	1600
32.8.2	FontDialog/ColorDialog	1601
32.9	WPF-Unterstützung mit dem ElementHost	1601
32.10	Praxisbeispiele	1601
32.10.1	Mit der CheckBox arbeiten	1601
32.10.2	Steuerelemente per Code selbst erzeugen	1603
32.10.3	Controls-Auflistung im TreeView anzeigen	1605
32.10.4	WPF-Komponenten mit dem ElementHost anzeigen	1609
33	Grundlagen Grafikausgabe	1613
33.1	Übersicht und erste Schritte	1613
33.1.1	GDI+ – Ein erster Einblick für Umsteiger	1614
33.1.2	Namespaces für die Grafikausgabe	1615

33.2	Darstellen von Grafiken	1617
33.2.1	Die PictureBox-Komponente	1617
33.2.2	Das Image-Objekt	1618
33.2.3	Laden von Grafiken zur Laufzeit	1619
33.2.4	Sichern von Grafiken	1619
33.2.5	Grafikeigenschaften ermitteln	1620
33.2.6	Erzeugen von Vorschaugrafiken (Thumbnails)	1621
33.2.7	Die Methode RotateFlip	1622
33.2.8	Skalieren von Grafiken	1623
33.3	Das .NET-Koordinatensystem	1624
33.3.1	Globale Koordinaten	1625
33.3.2	Seitenkoordinaten (globale Transformation)	1626
33.3.3	Gerätekoordinaten (Seitentransformation)	1628
33.4	Grundlegende Zeichenfunktionen von GDI+	1629
33.4.1	Das zentrale Graphics-Objekt	1629
33.4.2	Punkte zeichnen/abfragen	1632
33.4.3	Linien	1633
33.4.4	Kantenglättung mit Antialiasing	1634
33.4.5	PolyLine	1635
33.4.6	Rechtecke	1635
33.4.7	Polygone	1637
33.4.8	Splines	1638
33.4.9	Bézierkurven	1639
33.4.10	Kreise und Ellipsen	1640
33.4.11	Tortenstück (Segment)	1640
33.4.12	Bogenstück	1642
33.4.13	Wo sind die Rechtecke mit den runden Ecken?	1643
33.4.14	Textausgabe	1644
33.4.15	Ausgabe von Grafiken	1648
33.5	Unser Werkzeugkasten	1649
33.5.1	Einfache Objekte	1649
33.5.2	Vordefinierte Objekte	1651
33.5.3	Farben/Transparenz	1653
33.5.4	Stifte (Pen)	1654
33.5.5	Pinsel (Brush)	1657
33.5.6	SolidBrush	1658
33.5.7	HatchBrush	1658
33.5.8	TextureBrush	1659

33.5.9	LinearGradientBrush	1660
33.5.10	PathGradientBrush	1661
33.5.11	Fonts	1662
33.5.12	Path-Objekt	1663
33.5.13	Clipping/Region	1666
33.6	Standarddialoge	1670
33.6.1	Schriftauswahl	1670
33.6.2	Farbauswahl	1671
33.7	Praxisbeispiele	1673
33.7.1	Ein Graphics-Objekt erzeugen	1673
33.7.2	Zeichenoperationen mit der Maus realisieren	1675
34	Druckausgabe	1679
34.1	Einstieg und Übersicht	1679
34.1.1	Nichts geht über ein Beispiel	1679
34.1.2	Programmiermodell	1681
34.1.3	Kurzübersicht der Objekte	1682
34.2	Auswerten der Druckereinstellungen	1682
34.2.1	Die vorhandenen Drucker	1682
34.2.2	Der Standarddrucker	1683
34.2.3	Verfügbare Papierformate/Seitenabmessungen	1684
34.2.4	Der eigentliche Druckbereich	1685
34.2.5	Die Seitenausrichtung ermitteln	1686
34.2.6	Ermitteln der Farbfähigkeit	1686
34.2.7	Die Druckauflösung abfragen	1686
34.2.8	Ist beidseitiger Druck möglich?	1687
34.2.9	Einen "Informationsgerätekontext" erzeugen	1687
34.2.10	Abfragen von Werten während des Drucks	1688
34.3	Festlegen von Druckereinstellungen	1689
34.3.1	Einen Drucker auswählen	1689
34.3.2	Drucken in Millimetern	1689
34.3.3	Festlegen der Seitenränder	1690
34.3.4	Druckjobname	1691
34.3.5	Anzahl der Kopien	1692
34.3.6	Beidseitiger Druck	1692
34.3.7	Seitenzahlen festlegen	1693
34.3.8	Druckqualität verändern	1696
34.3.9	Ausgabemöglichkeiten des Chart-Controls nutzen	1697

34.4	Die Druckdialoge verwenden	1697
34.4.1	PrintDialog	1698
34.4.2	PageSetupDialog	1699
34.4.3	PrintPreviewDialog	1701
34.4.4	Ein eigenes Druckvorschau-Fenster realisieren	1702
34.5	Drucken mit OLE-Automation	1703
34.5.1	Kurzeinstieg in die OLE-Automation	1703
34.5.2	Drucken mit Microsoft Word	1705
34.6	Praxisbeispiele	1707
34.6.1	Den Drucker umfassend konfigurieren	1707
34.6.2	Diagramme mit dem Chart-Control drucken	1717
34.6.3	Druckausgabe mit Word	1719
35	Windows Forms-Datenbindung	1725
35.1	Prinzipielle Möglichkeiten	1725
35.2	Manuelle Bindung an einfache Datenfelder	1726
35.2.1	BindingSource erzeugen	1726
35.2.2	Binding-Objekt	1727
35.2.3	DataBindings-Collection	1727
35.3	Manuelle Bindung an Listen und Tabellen	1727
35.3.1	DataGridView	1728
35.3.2	Datenbindung von ComboBox und ListBox	1728
35.4	Entwurfszeit-Bindung an typisierte DataSets	1728
35.5	Drag & Drop-Datenbindung	1730
35.6	Navigations- und Bearbeitungsfunktionen	1730
35.6.1	Navigieren zwischen den Datensätzen	1730
35.6.2	Hinzufügen und Löschen	1730
35.6.3	Aktualisieren und Abbrechen	1731
35.6.4	Verwendung des BindingNavigators	1731
35.7	Die Anzeigedaten formatieren	1732
35.8	Praxisbeispiele	1732
35.8.1	Einrichten und Verwenden einer Datenquelle	1732
35.8.2	Eine Auswahlabfrage im DataGridView anzeigen	1736
35.8.3	Master-Detailbeziehungen im DataGrid anzeigen	1739
35.8.4	Datenbindung Chart-Control	1740

36	Erweiterte Grafikausgabe	1745
36.1	Transformieren mit der Matrix-Klasse	1745
36.1.1	Übersicht	1745
36.1.2	Translation	1746
36.1.3	Skalierung	1746
36.1.4	Rotation	1747
36.1.5	Scherung	1747
36.1.6	Zuweisen der Matrix	1748
36.2	Low-Level-Grafikmanipulationen	1748
36.2.1	Worauf zeigt Scan0?	1749
36.2.2	Anzahl der Spalten bestimmen	1750
36.2.3	Anzahl der Zeilen bestimmen	1751
36.2.4	Zugriff im Detail (erster Versuch)	1751
36.2.5	Zugriff im Detail (zweiter Versuch)	1753
36.2.6	Invertieren	1755
36.2.7	In Graustufen umwandeln	1756
36.2.8	Heller/Dunkler	1757
36.2.9	Kontrast	1759
36.2.10	Gamma-Wert	1760
36.2.11	Histogramm spreizen	1760
36.2.12	Ein universeller Grafikfilter	1763
36.3	Fortgeschrittene Techniken	1767
36.3.1	Flackerfrei dank Double Buffering	1767
36.3.2	Animationen	1769
36.3.3	Animated GIFs	1772
36.3.4	Auf einzelne GIF-Frames zugreifen	1774
36.3.5	Transparenz realisieren	1776
36.3.6	Eine Grafik maskieren	1777
36.3.7	JPEG-Qualität beim Sichern bestimmen	1779
36.4	Grundlagen der 3D-Vektorgrafik	1780
36.4.1	Datentypen für die Verwaltung	1780
36.4.2	Eine universelle 3D-Grafik-Klasse	1781
36.4.3	Grundlegende Betrachtungen	1782
36.4.4	Translation	1785
36.4.5	Streckung/Skalierung	1786
36.4.6	Rotation	1787
36.4.7	Die eigentlichen Zeichenroutinen	1789

36.5	Und doch wieder GDI-Funktionen ...	1791
36.5.1	Am Anfang war das Handle ...	1791
36.5.2	Gerätekontext (Device Context Types) ...	1794
36.5.3	Koordinatensysteme und Abbildungsmodi ...	1796
36.5.4	Zeichenwerkzeuge/Objekte ...	1800
36.5.5	Bitmaps ...	1802
36.6	Praxisbeispiele ...	1806
36.6.1	Die Transformationsmatrix verstehen ...	1806
36.6.2	Eine 3D-Grafikausgabe in Aktion ...	1809
36.6.3	Einen Fenster-Screenshot erzeugen ...	1812
37	Ressourcen/Lokalisierung	1815
37.1	Manifestressourcen	1815
37.1.1	Erstellen von Manifestressourcen	1815
37.1.2	Zugriff auf Manifestressourcen	1817
37.2	Typisierte Ressourcen	1819
37.2.1	Erzeugen von .resources-Dateien	1819
37.2.2	Hinzufügen der .resources-Datei zum Projekt	1819
37.2.3	Zugriff auf die Inhalte von .resources-Dateien	1820
37.2.4	ResourceManager einer .resources-Datei erzeugen	1820
37.2.5	Was sind .resx-Dateien?	1821
37.3	Streng typisierte Ressourcen	1821
37.3.1	Erzeugen streng typisierter Ressourcen	1822
37.3.2	Verwenden streng typisierter Ressourcen	1822
37.3.3	Streng typisierte Ressourcen per Reflection auslesen	1823
37.4	Anwendungen lokalisieren	1825
37.4.1	Localizable und Language	1825
37.4.2	Beispiel "Landesfahnen"	1825
37.4.3	Einstellen der aktuellen Kultur zur Laufzeit	1828
38	Komponentenentwicklung	1831
38.1	Überblick	1831
38.2	Benutzersteuerelement	1832
38.2.1	Entwickeln einer Auswahl-ListBox	1832
38.2.2	Komponente verwenden	1834
38.3	Benutzerdefiniertes Steuerelement	1835
38.3.1	Entwickeln eines BlinkLabels	1835
38.3.2	Verwenden der Komponente	1838

38.4	Komponentenklasse	1838
38.5	Eigenschaften	1839
38.5.1	Einfache Eigenschaften	1839
38.5.2	Schreib-/Lesezugriff (Get/Set)	1839
38.5.3	Nur Lese-Eigenschaft (ReadOnly)	1840
38.5.4	Nur-Schreibzugriff (WriteOnly)	1841
38.5.5	Hinzufügen von Beschreibungen	1841
38.5.6	Ausblenden im Eigenschaftenfenster	1841
38.5.7	Einfügen in Kategorien	1842
38.5.8	Default-Wert einstellen	1842
38.5.9	Standard-Eigenschaft (Indexer)	1843
38.5.10	Wertebereichsbeschränkung und Fehlerprüfung	1843
38.5.11	Eigenschaften von Aufzählungstypen	1845
38.5.12	Standard Objekt-Eigenschaften	1846
38.5.13	Eigene Objekt-Eigenschaften	1846
38.6	Methoden	1848
38.6.1	Konstruktor	1849
38.6.2	Class-Konstruktor	1850
38.6.3	Destruktor	1851
38.6.4	Aufruf des Basisklassen-Konstruktors	1851
38.6.5	Aufruf von Basisklassen-Methoden	1852
38.7	Ereignisse (Events)	1852
38.7.1	Ereignis mit Standardargument definieren	1853
38.7.2	Ereignis mit eigenen Argumenten	1854
38.7.3	Ein Default-Ereignis festlegen	1855
38.7.4	Mit Ereignissen auf Windows-Messages reagieren	1855
38.8	Weitere Themen	1857
38.8.1	Wohin mit der Komponente?	1857
38.8.2	Assembly-Informationen festlegen	1858
38.8.3	Assemblies signieren	1861
38.8.4	Komponenten-Ressourcen einbetten	1861
38.8.5	Der Komponente ein Icon zuordnen	1862
38.8.6	Den Designmodus erkennen	1863
38.8.7	Komponenten lizenzieren	1863
38.9	Praxisbeispiele	1867
38.9.1	AnimGif – Anzeige von Animationen	1867
38.9.2	Eine FontComboBox entwickeln	1870
38.9.3	Das PropertyGrid verwenden	1872

Teil VII: ASP.NET

39	Einführung in ASP.NET	1877
39.1	ASP.NET für Ein- und Umsteiger	1877
39.1.1	ASP – Ein kurzer Blick zurück	1877
39.1.2	Was ist bei ASP.NET anders?	1878
39.1.3	Was gibt es noch in ASP.NET?	1880
39.1.4	Vorteile von ASP.NET gegenüber ASP	1881
39.1.5	Voraussetzungen für den Einsatz von ASP.NET	1882
39.1.6	Und was hat das alles mit C# zu tun?	1882
39.2	Unsere erste Web-Anwendung	1885
39.2.1	Visueller Entwurf der Bedienoberfläche	1885
39.2.2	Zuweisen der Objekteigenschaften	1888
39.2.3	Verknüpfen der Objekte mit Ereignissen	1889
39.2.4	Programm kompilieren und testen	1890
39.3	Die ASP.NET-Projektdateien	1891
39.3.1	Die ASP.NET-Projekttypen	1892
39.3.2	ASPX-Datei(en)	1894
39.3.3	Die aspx.cs-Datei(en)	1896
39.3.4	Die Datei Global.asax	1897
39.3.5	Das Startformular	1898
39.3.6	Die Datei Web.config	1898
39.3.7	Masterpages (master-Dateien)	1901
39.3.8	Sitemap (Web.sitemap)	1901
39.3.9	Benutzersteuerelemente (ascx-Dateien)	1902
39.3.10	Die Web-Projekt-Verzeichnisse	1902
39.4	Lernen am Beispiel	1903
39.4.1	Erstellen des Projekts	1903
39.4.2	Oberflächengestaltung	1904
39.4.3	Ereignisprogrammierung	1905
39.4.4	Ein Fehler, was nun?	1907
39.4.5	Ereignisse von Textboxen	1908
39.4.6	Ein gemeinsamer Ereignis-Handler	1909
39.4.7	Eingabefokus setzen	1909
39.4.8	Ausgaben in einer Tabelle	1910
39.4.9	Scrollen der Anzeige	1912
39.4.10	Zusammenspiel mehrerer Formulare	1913
39.4.11	Umleiten bei Direktaufruf	1914

39.4.12	Ärger mit den Cookies	1915
39.4.13	Export auf den IIS	1917
39.5	Tipps & Tricks	1918
39.5.1	Nachinstallieren IIS 7 bzw. 7.5 (Windows 7)	1918
39.5.2	Nachinstallieren IIS8 (Windows 8)	1919
40	Übersicht ASP.NET-Controls	1921
40.1	Einfache Steuerelemente im Überblick	1921
40.1.1	Label	1921
40.1.2	TextBox	1923
40.1.3	Button, ImageButton, LinkButton	1924
40.1.4	CheckBox, RadioButton	1925
40.1.5	CheckBoxList, BulletList, RadioButtonList	1926
40.1.6	Table	1927
40.1.7	Hyperlink	1929
40.1.8	Image, ImageMap	1929
40.1.9	Calendar	1931
40.1.10	Panel	1932
40.1.11	HiddenField	1932
40.1.12	Substitution	1933
40.1.13	XML	1934
40.1.14	FileUpload	1936
40.1.15	AdRotator	1937
40.2	Steuerelemente für die Seitennavigation	1938
40.2.1	Mehr Übersicht mit Web.Sitemap	1938
40.2.2	Menu	1940
40.2.3	TreeView	1943
40.2.4	SiteMapPath	1946
40.2.5	MultiView, View	1947
40.2.6	Wizard	1948
40.3	Webseitenlayout/-design	1950
40.3.1	Masterpages	1950
40.3.2	Themes/Skins	1953
40.3.3	Webparts	1956
40.4	Die Validator-Controls	1957
40.4.1	Übersicht	1957
40.4.2	Wo findet die Fehlerprüfung statt?	1958
40.4.3	Verwendung	1958

40.4.4	RequiredFieldValidator	1959
40.4.5	CompareValidator	1960
40.4.6	RangeValidator	1962
40.4.7	RegularExpressionValidator	1962
40.4.8	CustomValidator	1963
40.4.9	ValidationSummary	1966
40.4.10	Weitere Möglichkeiten	1967
40.5	Praxisbeispiele	1967
40.5.1	Themes und Skins verstehen	1967
40.5.2	Masterpages verwenden	1972
40.5.3	Webparts verwenden	1975
41	Datenbindung unter ASP.NET	1981
41.1	Einstiegsbeispiel	1981
41.1.1	Erstellen der ASP.NET-Website	1981
41.2	Einführung	1986
41.2.1	Konzept	1986
41.2.2	Übersicht über die DataSource-Steuerelemente	1987
41.3	SQLDataSource	1988
41.3.1	Datenauswahl mit Parametern	1990
41.3.2	Parameter für INSERT, UPDATE und DELETE	1991
41.3.3	Methoden	1993
41.3.4	Caching	1994
41.3.5	Aktualisieren/Refresh	1995
41.4	AccessDataSource	1995
41.5	ObjectDataSource	1995
41.5.1	Verbindung zwischen Objekt und DataSource	1995
41.5.2	Ein Beispiel sorgt für Klarheit	1997
41.5.3	Geschäftsobjekte in einer Session verwalten	2001
41.6	SitemapDataSource	2003
41.7	LinqDataSource	2004
41.7.1	Bindung von einfachen Collections	2004
41.7.2	Bindung eines LINQ to SQL-DataContext	2005
41.8	EntityDataSource	2007
41.8.1	Entity Data Model erstellen	2007
41.8.2	EntityDataSource anbinden	2010
41.8.3	Datenmenge filtern	2013
41.9	XmlDataSource	2013

41.10	QueryExtender	2015
41.10.1	Grundlagen	2015
41.10.2	Suchen	2016
41.10.3	Sortieren	2018
41.11	GridView	2019
41.11.1	Auswahlfunktion (Zeilenauswahl)	2019
41.11.2	Auswahl mit mehrspaltigem Index	2020
41.11.3	Hyperlink-Spalte für Detailansicht	2020
41.11.4	Spalten erzeugen	2021
41.11.5	Paging realisieren	2022
41.11.6	Edit, Update, Delete	2024
41.11.7	Keine Daten, was tun?	2024
41.12	DetailsView	2024
41.13	FormView	2026
41.14	DataList	2029
41.14.1	Bearbeitungsfunktionen implementieren	2030
41.14.2	Layout verändern	2031
41.15	Repeater	2032
41.16	ListView	2033
41.17	Typisierte Datenbindung	2033
41.18	Model Binding	2034
41.19	Chart	2036
42	ASP.NET-Objekte/-Techniken	2039
42.1	Wichtige ASP.NET-Objekte	2039
42.1.1	HTTPApplication	2039
42.1.2	Application	2042
42.1.3	Session	2043
42.1.4	Page	2045
42.1.5	Request	2048
42.1.6	Response	2051
42.1.7	Server	2055
42.1.8	Cookies verwenden	2056
42.2	Fehlerbehandlung unter ASP.NET	2059
42.2.1	Fehler beim Entwurf	2059
42.2.2	Laufzeitfehler	2059
42.2.3	Eine eigene Fehlerseite	2061
42.2.4	Fehlerbehandlung im Web Form	2062

42.2.5	Fehlerbehandlung in der Anwendung	2063
42.2.6	Alternative Fehlerseite einblenden	2064
42.2.7	Lokale Fehlerbehandlung	2065
42.2.8	Seite nicht gefunden – was nun?	2066
42.3	E-Mail-Versand in ASP.NET	2066
42.3.1	Übersicht	2067
42.3.2	Mail-Server bestimmen	2067
42.3.3	Einfache Text-E-Mails versenden	2069
42.3.4	E-Mails mit Dateianhang	2070
42.4	Sicherheit von Webanwendungen	2071
42.4.1	Authentication	2071
42.4.2	Forms Authentication realisieren	2072
42.4.3	Impersonation	2076
42.4.4	Authorization	2077
42.4.5	Administrieren der Website	2079
42.4.6	Steuerelemente für das Login-Handling	2083
42.4.7	Programmieren der Sicherheitseinstellungen	2087
42.5	AJAX in ASP.NET-Anwendungen	2089
42.5.1	Was ist AJAX und was kann es?	2089
42.5.2	Die AJAX-Controls	2090
42.5.3	AJAX-Control-Toolkit	2094
42.6	User Controls/Webbenutzersteuerelemente	2095
42.6.1	Ein simples Einstiegsbeispiel	2096
42.6.2	Dynamische Grafiken im User Control anzeigen	2099
42.6.3	Grafikausgaben per User Control realisieren	2104

Teil VIII: Silverlight

43	Silverlight-Entwicklung	2109
43.1	Einführung	2109
43.1.1	Zielplattformen	2110
43.1.2	Silverlight-Applikationstypen	2110
43.1.3	Wichtige Unterschiede zu den WPF-Anwendungen	2112
43.1.4	Vor- und Nachteile von Silverlight-Anwendungen	2114
43.1.5	Entwicklungstools	2116
43.1.6	Installation auf dem Client	2116

43.2	Die Silverlight-Anwendung im Detail	2117
43.2.1	Ein kleines Beispielprojekt	2118
43.2.2	Das Application Package und das Test-Web	2120
43.3	Die Projektdateien im Überblick	2123
43.3.1	Projektverwaltung mit App.xaml & App.xaml.cs	2124
43.3.2	MainPage.xaml & MainPage.xaml.cs	2126
43.3.3	AssemblyInfo.cs	2127
43.4	Fenster und Seiten in Silverlight	2127
43.4.1	Das Standardfenster	2128
43.4.2	Untergeordnete Silverlight-Fenster	2129
43.4.3	UserControls für die Anzeige von Detaildaten	2131
43.4.4	Echte Windows	2132
43.4.5	Navigieren in Silverlight-Anwendungen	2133
43.5	Datenbanken/Datenbindung	2138
43.5.1	ASP.NET-Webdienste/WCF-Dienste	2139
43.5.2	WCF Data Services	2148
43.6	Isolierter Speicher	2159
43.6.1	Grundkonzept	2159
43.6.2	Das virtuelle Dateisystem verwalten	2160
43.6.3	Arbeiten mit Dateien	2163
43.7	Fulltrust-Anwendungen	2164
43.8	Praxisbeispiele	2167
43.8.1	Eine Out-of-Browser-Applikation realisieren	2167
43.8.2	Out-of-Browser-Anwendung aktualisieren	2171
43.8.3	Testen auf aktive Internetverbindung	2172
43.8.4	Auf Out-of-Browser-Anwendung testen	2173
43.8.5	Den Browser bestimmen	2173
43.8.6	Parameter an das Plug-in übergeben	2174
43.8.7	Auf den QueryString zugreifen	2176
43.8.8	Timer in Silverlight nutzen	2177
43.8.9	Dateien lokal speichern	2178
43.8.10	Drag & Drop realisieren	2180
43.8.11	Auf die Zwischenablage zugreifen	2181
43.8.12	Weitere Fenster öffnen	2183
	Index	2187