



Inhaltsverzeichnis

Walter Doberenz, Thomas Gewinnus

Visual Basic 2012 - Grundlagen und Profiwissen

ISBN (Buch): 978-3-446-43429-5

ISBN (E-Book): 978-3-446-43522-3

Weitere Informationen oder Bestellungen unter

<http://www.hanser-fachbuch.de/978-3-446-43429-5>

sowie im Buchhandel.

Inhaltsverzeichnis

Vorwort	53
 Teil I: Grundlagen	
1 Einstieg in Visual Studio 2012	59
1.1 Die Installation von Visual Studio 2012	59
1.1.1 Überblick über die Produktpalette	59
1.1.2 Anforderungen an Hard- und Software	61
1.2 Unser allererstes VB-Programm	62
1.2.1 Vorbereitungen	62
1.2.2 Programm schreiben	64
1.2.3 Programm kompilieren und testen	65
1.2.4 Einige Erläuterungen zum Quellcode	65
1.2.5 Konsolenanwendungen sind langweilig	66
1.3 Die Windows-Philosophie	67
1.3.1 Mensch-Rechner-Dialog	67
1.3.2 Objekt- und ereignisorientierte Programmierung	67
1.3.3 Windows-Programmierung unter Visual Studio 2012	69
1.4 Die Entwicklungsumgebung Visual Studio 2012	70
1.4.1 Der Startdialog	70
1.4.2 Die wichtigsten Fenster	71
1.5 Microsofts .NET-Technologie	74
1.5.1 Zur Geschichte von .NET	74
1.5.2 .NET-Features und Begriffe	76
1.6 Wichtige Neuigkeiten in Visual Studio 2012	83
1.6.1 Die neue Visual Studio 2012-Entwicklungsumgebung	83
1.6.2 Neuheiten im .NET Framework 4.5	87
1.6.3 VB 2012 – Sprache und Compiler	88
1.7 Praxisbeispiele	89
1.7.1 Windows-Anwendung für Einsteiger	89
1.7.2 Windows-Anwendung für fortgeschrittene Einsteiger	93

2	Einführung in Visual Basic	101
2.1	Grundbegriffe	101
2.1.1	Anweisungen	101
2.1.2	Bezeichner	102
2.1.3	Kommentare	103
2.1.4	Zeilenumbruch	104
2.2	Datentypen, Variablen und Konstanten	106
2.2.1	Fundamentale Typen	106
2.2.2	Wertetypen versus Verweistypen	107
2.2.3	Benennung von Variablen	107
2.2.4	Deklaration von Variablen	108
2.2.5	Typinferenz	111
2.2.6	Konstanten deklarieren	111
2.2.7	Gültigkeitsbereiche von Deklarationen	112
2.2.8	Lokale Variablen mit Dim	112
2.2.9	Lokale Variablen mit Static	113
2.2.10	Private globale Variablen	113
2.2.11	Public Variablen	114
2.3	Wichtige Datentypen im Überblick	114
2.3.1	Byte, Short, Integer, Long	114
2.3.2	Single, Double und Decimal	115
2.3.3	Char und String	115
2.3.4	Date	116
2.3.5	Boolean	117
2.3.6	Object	117
2.4	Konvertieren von Datentypen	118
2.4.1	Implizite und explizite Konvertierung	118
2.4.2	Welcher Datentyp passt zu welchem?	119
2.4.3	Konvertierungsfunktionen	120
2.4.4	CType-Funktion	121
2.4.5	Konvertieren von Strings	121
2.4.6	Die Convert-Klasse	123
2.4.7	Die Parse-Methode	123
2.4.8	Boxing und Unboxing	124
2.4.9	TryCast-Operator	125
2.4.10	Nullable Types	125

2.5	Operatoren	126
2.5.1	Arithmetische Operatoren	126
2.5.2	Zuweisungsoperatoren	127
2.5.3	Logische Operatoren	128
2.5.4	Vergleichsoperatoren	129
2.5.5	Rangfolge der Operatoren	129
2.6	Kontrollstrukturen	130
2.6.1	Verzweigungsbefehle	130
2.6.2	Schleifenanweisungen	133
2.7	Benutzerdefinierte Datentypen	134
2.7.1	Enumerationen	134
2.7.2	Strukturen	135
2.8	Nutzerdefinierte Funktionen/Prozeduren	138
2.8.1	Deklaration und Syntax	138
2.8.2	Parameterübergabe allgemein	140
2.8.3	Übergabe mit ByVal und ByRef	141
2.8.4	Optionale Parameter	142
2.8.5	Überladene Funktionen/Prozeduren	142
2.9	Praxisbeispiele	143
2.9.1	Vom PAP zum Konsolen-Programm	143
2.9.2	Vom Konsolen- zum Windows-Programm	145
2.9.3	Schleifenanweisungen kennen lernen	147
2.9.4	Methoden überladen	150
2.9.5	Eine Iterationsschleife verstehen	152
3	OOP-Konzepte	157
3.1	Strukturierter versus objektorientierter Entwurf	157
3.1.1	Was bedeutet strukturierte Programmierung?	157
3.1.2	Was heißt objektorientierte Programmierung?	158
3.2	Grundbegriffe der OOP	159
3.2.1	Objekt, Klasse, Instanz	159
3.2.2	Kapselung und Wiederverwendbarkeit	160
3.2.3	Vererbung und Polymorphie	160
3.2.4	Sichtbarkeit von Klassen und ihren Mitgliedern	161
3.2.5	Allgemeiner Aufbau einer Klasse	162
3.3	Ein Objekt erzeugen	163
3.3.1	Referenzieren und Instanzieren	164
3.3.2	Klassische Initialisierung	165

3.3.3	Objekt-Initialisierer	165
3.3.4	Arbeiten mit dem Objekt	165
3.3.5	Zerstören des Objekts	166
3.4	OOP-Einführungsbeispiel	166
3.4.1	Vorbereitungen	166
3.4.2	Klasse definieren	167
3.4.3	Objekt erzeugen und initialisieren	168
3.4.4	Objekt verwenden	168
3.4.5	IntelliSense – die hilfreiche Fee	168
3.4.6	Objekt testen	169
3.4.7	Warum unsere Klasse noch nicht optimal ist	170
3.5	Eigenschaften	170
3.5.1	Eigenschaften kapseln	170
3.5.2	Eigenschaften mit Zugriffsmethoden kapseln	173
3.5.3	Lese-/Schreibschutz für Eigenschaften	174
3.5.4	Statische Eigenschaften	175
3.5.5	Selbst implementierende Eigenschaften	176
3.6	Methoden	177
3.6.1	Öffentliche und private Methoden	177
3.6.2	Überladene Methoden	178
3.6.3	Statische Methoden	178
3.7	Ereignisse	180
3.7.1	Ereignisse deklarieren	180
3.7.2	Ereignis auslösen	180
3.7.3	Ereignis auswerten	181
3.7.4	Benutzerdefinierte Ereignisse (Custom Events)	182
3.8	Arbeiten mit Konstruktor und Destruktor	186
3.8.1	Der Konstruktor erzeugt das Objekt	186
3.8.2	Bequemer geht's mit einem Objekt-Initialisierer	188
3.8.3	Destruktor und Garbage Collector räumen auf	188
3.8.4	Mit Using den Lebenszyklus des Objekts kapseln	191
3.9	Vererbung und Polymorphie	191
3.9.1	Vererbungsbeziehungen im Klassendiagramm	192
3.9.2	Überschreiben von Methoden (Method-Overriding)	193
3.9.3	Klassen implementieren	193
3.9.4	Objekte implementieren	198
3.9.5	Ausblenden von Mitgliedern durch Vererbung	200

3.9.6	Allgemeine Hinweise und Regeln zur Vererbung	201
3.9.7	Polymorphe Methoden	202
3.10	Besondere Klassen und Features	204
3.10.1	Abstrakte Klassen	204
3.10.2	Abstrakte Methoden	205
3.10.3	Versiegelte Klassen	206
3.10.4	Partielle Klassen	206
3.10.5	Die Basisklasse System.Object	208
3.10.6	Property-Accessors	209
3.11	Schnittstellen (Interfaces)	209
3.11.1	Definition einer Schnittstelle	210
3.11.2	Implementieren einer Schnittstelle	210
3.11.3	Abfragen, ob eine Schnittstelle vorhanden ist	211
3.11.4	Mehrere Schnittstellen implementieren	212
3.11.5	Schnittstellenprogrammierung ist ein weites Feld	212
3.12	Praxisbeispiele	212
3.12.1	Eigenschaften sinnvoll kapseln	212
3.12.2	Eine statische Klasse anwenden	216
3.12.3	Vom fetten zum dünnen Client	217
3.12.4	Schnittstellenvererbung verstehen	227
3.12.5	Aggregation und Vererbung gegenüberstellen	231
3.12.6	Eine Klasse zur Matrizenrechnung entwickeln	238
4	Arrays, Strings, Funktionen	245
4.1	Datenfelder (Arrays)	245
4.1.1	Ein Array deklarieren	245
4.1.2	Zugriff auf Array-Elemente	246
4.1.3	Oberen Index ermitteln	246
4.1.4	Explizite Arraygrenzen	246
4.1.5	Arrays erzeugen und initialisieren	246
4.1.6	Zugriff mittels Schleife	247
4.1.7	Mehrdimensionale Arrays	248
4.1.8	Dynamische Arrays	249
4.1.9	Zuweisen von Arrays	250
4.1.10	Arrays aus Strukturvariablen	251
4.1.11	Löschen von Arrays	252
4.1.12	Eigenschaften und Methoden von Arrays	252
4.1.13	Übergabe von Arrays	255

4.2	Zeichenkettenverarbeitung	256
4.2.1	Strings zuweisen	256
4.2.2	Eigenschaften und Methoden eines Strings	256
4.2.3	Kopieren eines Strings in ein Char-Array	259
4.2.4	Wichtige (statische) Methoden der String-Klasse	259
4.2.5	Die StringBuilder-Klasse	261
4.3	Reguläre Ausdrücke	264
4.3.1	Wozu werden reguläre Ausdrücke verwendet?	264
4.3.2	Eine kleine Einführung	265
4.3.3	Wichtige Methoden der Klasse Regex	265
4.3.4	Kompilierte reguläre Ausdrücke	267
4.3.5	RegexOptions-Enumeration	268
4.3.6	Metazeichen (Escape-Zeichen)	269
4.3.7	Zeichenmengen (Character Sets)	270
4.3.8	Quantifizierer	271
4.3.9	Zero-Width Assertions	272
4.3.10	Gruppen	276
4.3.11	Text ersetzen	276
4.3.12	Text splitten	277
4.4	Datums- und Zeitberechnungen	278
4.4.1	Grundlegendes	278
4.4.2	Wichtige Eigenschaften von DateTime-Variablen	279
4.4.3	Wichtige Methoden von DateTime-Variablen	280
4.4.4	Wichtige Mitglieder der DateTime-Struktur	281
4.4.5	Konvertieren von Datumstrings in DateTime-Werte	282
4.4.6	Die TimeSpan-Struktur	282
4.5	Vordefinierten Funktionen	284
4.5.1	Mathematik	284
4.5.2	Datums- und Zeitfunktionen	286
4.6	Zahlen formatieren	289
4.6.1	Die ToString-Methode	289
4.6.2	Die Format-Methode	291
4.7	Praxisbeispiele	292
4.7.1	Zeichenketten verarbeiten	292
4.7.2	Zeichenketten mittels StringBuilder addieren	295
4.7.3	Reguläre Ausdrücke testen	299
4.7.4	Methodenaufrufe mit Array-Parametern	300

5	Weitere Sprachfeatures	305
5.1	Namespaces (Namensräume)	305
5.1.1	Ein kleiner Überblick	305
5.1.2	Die Imports-Anweisung	307
5.1.3	Namespace-Alias	307
5.1.4	Namespaces in Projekteigenschaften	308
5.1.5	Namespace Alias Qualifizierer	308
5.1.6	Eigene Namespaces einrichten	309
5.2	Überladen von Operatoren	310
5.2.1	Syntaxregeln	310
5.2.2	Praktische Anwendung	311
5.2.3	Konvertierungsoperatoren überladen	312
5.3	Auflistungen (Collections)	313
5.3.1	Beziehungen zwischen den Schnittstellen	313
5.3.2	IEnumerable	313
5.3.3	ICollection	314
5.3.4	ICollection	315
5.3.5	Iteratoren	315
5.3.6	Die ArrayList-Collection	316
5.3.7	Die Hashtable	317
5.4	Generische Datentypen	318
5.4.1	Wie es früher einmal war	318
5.4.2	Typsicherheit durch Generics	320
5.4.3	List-Collection ersetzt ArrayList	321
5.4.4	Über die Vorzüge generischer Collections	322
5.4.5	Typbeschränkungen durch Constraints	323
5.4.6	Collection-Initialisierer	324
5.4.7	Generische Methoden	324
5.5	Delegates	325
5.5.1	Delegates sind Methodenzeiger	325
5.5.2	Delegate-Typ definieren	326
5.5.3	Delegate-Objekt erzeugen	327
5.5.4	Delegates vereinfacht instanziiieren	327
5.5.5	Relaxed Delegates	328
5.5.6	Anonyme Methoden	328
5.5.7	Lambda-Ausdrücke	329
5.5.8	Lambda-Ausdrücke in der Task Parallel Library	330

5.6	Dynamische Programmierung	332
5.6.1	Wozu dynamische Programmierung?	332
5.6.2	Das Prinzip der dynamischen Programmierung	333
5.6.3	Kovarianz und Kontravarianz	336
5.7	Weitere Datentypen	337
5.7.1	BigInteger	337
5.7.2	Complex	339
5.7.3	Tuple(Of T)	340
5.7.4	SortedSet(Of T)	341
5.8	Praxisbeispiele	342
5.8.1	ArrayList versus generische List	342
5.8.2	Delegates und Lambda Expressions	345
5.8.3	Mit dynamischem Objekt eine Datei durchsuchen	348
6	Einführung in LINQ	353
6.1	LINQ-Grundlagen	353
6.1.1	Die LINQ-Architektur	353
6.1.2	LINQ-Implementierungen	354
6.1.3	Anonyme Typen	354
6.1.4	Erweiterungsmethoden	356
6.2	Abfragen mit LINQ to Objects	357
6.2.1	Grundlegendes zur LINQ-Syntax	357
6.2.2	Zwei alternative Schreibweisen von LINQ-Abfragen	358
6.2.3	Übersicht der wichtigsten Abfrage-Operatoren	360
6.3	LINQ-Abfragen im Detail	361
6.3.1	Die Projektionsoperatoren Select und SelectMany	362
6.3.2	Der Restriktionsoperator Where	364
6.3.3	Die Sortierungsoperatoren OrderBy und ThenBy	364
6.3.4	Der Gruppierungsoperator GroupBy	366
6.3.5	Verknüpfen mit Join	367
6.3.6	Aggregat-Operatoren	368
6.3.7	Verzögertes Ausführen von LINQ-Abfragen	370
6.3.8	Konvertierungsmethoden	371
6.3.9	Abfragen mit PLINQ	371
6.4	Praxisbeispiele	374
6.4.1	Die Syntax von LINQ-Abfragen verstehen	374
6.4.2	Aggregat-Abfragen mit LINQ	377

Teil II: Technologien

7	Zugriff auf das Dateisystem	383
7.1	Grundlagen	383
7.1.1	Klassen für Verzeichnis- und Dateioperationen	384
7.1.2	Statische versus Instanzen-Klasse	384
7.2	Übersichten	385
7.2.1	Methoden der Directory-Klasse	385
7.2.2	Methoden eines DirectoryInfo-Objekts	386
7.2.3	Eigenschaften eines DirectoryInfo-Objekts	386
7.2.4	Methoden der File-Klasse	386
7.2.5	Methoden eines FileInfo-Objekts	387
7.2.6	Eigenschaften eines FileInfo-Objekts	388
7.3	Operationen auf Verzeichnisebene	388
7.3.1	Existenz eines Verzeichnisses/einer Datei feststellen	388
7.3.2	Verzeichnisse erzeugen und löschen	389
7.3.3	Verzeichnisse verschieben und umbenennen	390
7.3.4	Aktuelles Verzeichnis bestimmen	390
7.3.5	Unterverzeichnisse ermitteln	390
7.3.6	Alle Laufwerke ermitteln	391
7.3.7	Dateien kopieren und verschieben	392
7.3.8	Dateien umbenennen	393
7.3.9	Dateiattribute feststellen	393
7.3.10	Verzeichnis einer Datei ermitteln	395
7.3.11	Alle im Verzeichnis enthaltene Dateien ermitteln	395
7.3.12	Dateien und Unterverzeichnisse ermitteln	395
7.4	Zugriffsberechtigungen	396
7.4.1	ACL und ACE	396
7.4.2	SetAccessControl-Methode	397
7.4.3	Zugriffsrechte anzeigen	397
7.5	Weitere wichtige Klassen	398
7.5.1	Die Path-Klasse	398
7.5.2	Die Klasse FileSystemWatcher	399
7.6	Datei- und Verzeichnisdialoge	401
7.6.1	OpenFileDialog und SaveFileDialog	401
7.6.2	FolderBrowserDialog	402

7.7	Praxisbeispiele	404
7.7.1	Infos über Verzeichnisse und Dateien gewinnen	404
7.7.2	Die Verzeichnisstruktur in eine TreeView einlesen	407
7.7.3	Mit LINQ und RegEx Verzeichnisbäume durchsuchen	409
8	Dateien lesen und schreiben	415
8.1	Grundprinzip der Datenpersistenz	415
8.1.1	Dateien und Streams	415
8.1.2	Die wichtigsten Klassen	416
8.1.3	Erzeugen eines Streams	417
8.2	Dateiparameter	417
8.2.1	FileAccess	417
8.2.2	FileMode	417
8.2.3	FileShare	418
8.3	Textdateien	418
8.3.1	Eine Textdatei beschreiben bzw. neu anlegen	418
8.3.2	Eine Textdatei lesen	420
8.4	Binärdateien	422
8.4.1	Lese-/Schreibzugriff	422
8.4.2	Die Methoden ReadAllBytes und WriteAllBytes	422
8.4.3	BinaryReader/BinaryWriter erzeugen	423
8.5	Sequenzielle Dateien	423
8.5.1	Lesen und Schreiben von strukturierten Daten	423
8.5.2	Serialisieren von Objekten	424
8.6	Dateien verschlüsseln und komprimieren	425
8.6.1	Das Methodenpärchen Encrypt-/Decrypt	426
8.6.2	Verschlüsseln unter Windows Vista/Windows 7/8	426
8.6.3	Verschlüsseln mit der CryptoStream-Klasse	427
8.6.4	Dateien komprimieren	428
8.7	Memory Mapped Files	429
8.7.1	Grundprinzip	429
8.7.2	Erzeugen eines MMF	430
8.7.3	Erstellen eines Map View	430
8.8	Praxisbeispiele	431
8.8.1	Auf eine Textdatei zugreifen	431
8.8.2	Einen Objektbaum speichern	435
8.8.3	Ein Memory Mapped File (MMF) verwenden	441

9	Asynchrone Programmierung	445
9.1	Übersicht	446
9.1.1	Multitasking versus Multithreading	446
9.1.2	Deadlocks	447
9.1.3	Racing	447
9.2	Programmieren mit Threads	449
9.2.1	Einführungsbeispiel	449
9.2.2	Wichtige Thread-Methoden	450
9.2.3	Wichtige Thread-Eigenschaften	452
9.2.4	Einsatz der ThreadPool-Klasse	453
9.3	Sperrmechanismen	455
9.3.1	Threading ohne SyncLock	455
9.3.2	Threading mit SyncLock	457
9.3.3	Die Monitor-Klasse	459
9.3.4	Mutex	462
9.3.5	Methoden für die parallele Ausführung sperren	464
9.3.6	Semaphore	464
9.4	Interaktion mit der Programmoberfläche	466
9.4.1	Die Werkzeuge	466
9.4.2	Einzelne Steuerelemente mit Invoke aktualisieren	466
9.4.3	Mehrere Steuerelemente aktualisieren	468
9.4.4	Ist ein Invoke-Aufruf nötig?	468
9.4.5	Und was ist mit WPF?	469
9.5	Timer-Threads	470
9.6	Die BackgroundWorker-Komponente	471
9.7	Asynchrone Programmier-Entwurfsmuster	474
9.7.1	Kurzübersicht	474
9.7.2	Polling	475
9.7.3	Callback verwenden	476
9.7.4	Callback mit Parameterübergabe verwenden	477
9.7.5	Callback mit Zugriff auf die Programm-Oberfläche	478
9.8	Asynchroner Aufruf beliebiger Methoden	479
9.8.1	Die Beispielklasse	479
9.8.2	Asynchroner Aufruf ohne Callback	481
9.8.3	Asynchroner Aufruf mit Callback und Anzeigefunktion	481
9.8.4	Aufruf mit Rückgabewerten (per Eigenschaft)	482
9.8.5	Aufruf mit Rückgabewerten (per EndInvoke)	483

9.9	Es geht auch einfacher – Async und Await	484
9.9.1	Der Weg von synchron zu asynchron	484
9.9.2	Achtung: Fehlerquellen!	486
9.9.3	Eigene asynchrone Methoden entwickeln	488
9.10	Praxisbeispiele	490
9.10.1	Spieltrieb & Multithreading erleben	490
9.10.2	Prozess- und Thread-Informationen gewinnen	502
9.10.3	Ein externes Programm starten	507
10	Die Task Parallel Library	511
10.1	Überblick	511
10.1.1	Parallel-Programmierung	511
10.1.2	Möglichkeiten der TPL	514
10.1.3	Der CLR-Threadpool	514
10.2	Parallele Verarbeitung mit Parallel.Invoke	515
10.2.1	Aufrufvarianten	516
10.2.2	Einschränkungen	517
10.3	Verwendung von Parallel.For	517
10.3.1	Abbrechen der Verarbeitung	519
10.3.2	Auswerten des Bearbeitungsstatus	520
10.3.3	Und was ist mit anderen Iterator-Schrittweiten?	520
10.4	Collections mit Parallel.ForEach verarbeiten	521
10.5	Die Task-Klasse	522
10.5.1	Einen Task erzeugen	522
10.5.2	Task starten	523
10.5.3	Datenübergabe an den Task	524
10.5.4	Wie warte ich auf das Taskende?	525
10.5.5	Tasks mit Rückgabewerten	527
10.5.6	Die Verarbeitung abbrechen	530
10.5.7	Fehlerbehandlung	534
10.5.8	Weitere Eigenschaften	534
10.6	Zugriff auf das Userinterface	536
10.6.1	Task-Ende und Zugriff auf die Oberfläche	536
10.6.2	Zugriff auf das UI aus dem Task heraus	537
10.7	Weitere Datenstrukturen im Überblick	539
10.7.1	Threadsichere Collections	539
10.7.2	Primitive für die Threadsynchronisation	540

10.8	Parallel LINQ (PLINQ)	540
10.9	Die Parallel Diagnostic Tools	541
10.9.1	Fenster für parallele Aufgaben	541
10.9.2	Fenster für parallele Stacks	542
10.9.3	IntelliTrace	543
10.10	Praxisbeispiel: Spieltrieb – Version 2	543
10.10.1	Aufgabenstellung	543
10.10.2	Global-Klasse	544
10.10.3	Controller	545
10.10.4	LKWs	546
10.10.5	Schiff-Klasse	548
10.10.6	Oberfläche	550
11	Fehlersuche und Behandlung	553
11.1	Der Debugger	553
11.1.1	Allgemeine Beschreibung	553
11.1.2	Die wichtigsten Fenster	554
11.1.3	Debugging-Optionen	557
11.1.4	Praktisches Debugging am Beispiel	559
11.2	Arbeiten mit Debug und Trace	563
11.2.1	Wichtige Methoden von Debug und Trace	563
11.2.2	Besonderheiten der Trace-Klasse	566
11.2.3	TraceListener-Objekte	567
11.3	Caller Information	570
11.3.1	Attribute	570
11.3.2	Anwendung	570
11.4	Fehlerbehandlung	571
11.4.1	Anweisungen zur Fehlerbehandlung	571
11.4.2	Try-Catch	571
11.4.3	Try-Finally	576
11.4.4	Das Standardverhalten bei Ausnahmen festlegen	578
11.4.5	Die Exception-Klasse	579
11.4.6	Fehler/Ausnahmen auslösen	579
11.4.7	Eigene Fehlerklassen	580
11.4.8	Exceptionhandling zur Entwurfszeit	582
11.4.9	Code Contracts	582

Teil III: WPF-Anwendungen

12	Einführung in WPF	587
12.1	Neues aus der Gerüchteküche	588
12.1.1	Silverlight	588
12.1.2	WPF	588
12.2	Einführung	589
12.2.1	Was kann eine WPF-Anwendung?	589
12.2.2	Die eXtensible Application Markup Language	591
12.2.3	Verbinden von XAML und Basic-Code	595
12.2.4	Zielplattformen	601
12.2.5	Applikationstypen	601
12.2.6	Vorteile und Nachteile von WPF-Anwendungen	602
12.2.7	Weitere Dateien im Überblick	603
12.3	Alles beginnt mit dem Layout	606
12.3.1	Allgemeines zum Layout	606
12.3.2	Positionieren von Steuerelementen	608
12.3.3	Canvas	611
12.3.4	StackPanel	612
12.3.5	DockPanel	614
12.3.6	WrapPanel	616
12.3.7	UniformGrid	616
12.3.8	Grid	618
12.3.9	ViewBox	622
12.3.10	TextBlock	623
12.4	Das WPF-Programm	626
12.4.1	Die Application-Klasse	627
12.4.2	Das Startobjekt festlegen	627
12.4.3	Kommandozeilenparameter verarbeiten	629
12.4.4	Die Anwendung beenden	629
12.4.5	Auswerten von Anwendungsereignissen	630
12.5	Die Window-Klasse	631
12.5.1	Position und Größe festlegen	631
12.5.2	Rahmen und Beschriftung	631
12.5.3	Das Fenster-Icon ändern	632
12.5.4	Anzeige weiterer Fenster	632
12.5.5	Transparenz	632
12.5.6	Abstand zum Inhalt festlegen	633

12.5.7	Fenster ohne Fokus anzeigen	634
12.5.8	Ereignisfolge bei Fenstern	634
12.5.9	Ein paar Worte zur Schriftdarstellung	635
12.5.10	Ein paar Worte zur Controldarstellung	637
12.5.11	Wird mein Fenster komplett mit WPF gerendert?	639
13	Übersicht WPF-Controls	641
13.1	Allgemeingültige Eigenschaften	641
13.2	Label	643
13.3	Button, RepeatButton, ToggleButton	643
13.3.1	Schaltflächen für modale Dialoge	644
13.3.2	Schaltflächen mit Grafik	645
13.4	TextBox, PasswortBox	646
13.4.1	TextBox	646
13.4.2	PasswordBox	648
13.5	CheckBox	649
13.6	RadioButton	650
13.7	ListBox, ComboBox	652
13.7.1	ListBox	652
13.7.2	ComboBox	655
13.7.3	Den Content formatieren	656
13.8	Image	658
13.8.1	Grafik per XAML zuweisen	658
13.8.2	Grafik zur Laufzeit zuweisen	658
13.8.3	Bild aus Datei laden	659
13.8.4	Die Grafiskalierung beeinflussen	660
13.9	MediaElement	661
13.10	Slider, ScrollBar	664
13.10.1	Slider	664
13.10.2	ScrollBar	665
13.11	ScrollViewer	666
13.12	Menu, ContextMenu	667
13.12.1	Menu	667
13.12.2	Tastenkürzel	668
13.12.3	Grafiken	669
13.12.4	Weitere Möglichkeiten	670
13.12.5	ContextMenu	671
13.13	ToolBar	672

13.14	StatusBar, ProgressBar	675
13.14.1	StatusBar	675
13.14.2	ProgressBar	677
13.15	Border, GroupBox, BulletDecorator	678
13.15.1	Border	678
13.15.2	GroupBox	679
13.15.3	BulletDecorator	680
13.16	RichTextBox	682
13.16.1	Verwendung und Anzeige von vordefiniertem Text	682
13.16.2	Neues Dokument zur Laufzeit erzeugen	684
13.16.3	Sichern von Dokumenten	684
13.16.4	Laden von Dokumenten	686
13.16.5	Texte per Code einfügen/modifizieren	687
13.16.6	Texte formatieren	688
13.16.7	EditingCommands	690
13.16.8	Grafiken/Objekte einfügen	690
13.16.9	Rechtschreibkontrolle	692
13.17	FlowDocumentPageViewer & Co.	692
13.17.1	FlowDocumentPageViewer	692
13.17.2	FlowDocumentReader	693
13.17.3	FlowDocumentScrollViewer	693
13.18	FlowDocument	693
13.18.1	FlowDocument per XAML beschreiben	694
13.18.2	FlowDocument per Code erstellen	696
13.19	DocumentViewer	697
13.20	Expander, TabControl	698
13.20.1	Expander	698
13.20.2	TabControl	700
13.21	Popup	701
13.22	TreeView	703
13.23	ListView	706
13.24	DataGrid	707
13.25	Calendar/DatePicker	707
13.26	InkCanvas	711
13.26.1	Stift-Parameter definieren	712
13.26.2	Die Zeichenmodi	713
13.26.3	Inhalte laden und sichern	713

13.26.4	Konvertieren in eine Bitmap	714
13.26.5	Weitere Eigenschaften	715
13.27	Ellipse, Rectangle, Line und Co.	715
13.27.1	Ellipse	715
13.27.2	Rectangle	716
13.27.3	Line	716
13.28	Browser	717
13.29	Ribbon	719
13.29.1	Allgemeine Grundlagen	719
13.29.2	Download/Installation	721
13.29.3	Erste Schritte	721
13.29.4	Registerkarten und Gruppen	722
13.29.5	Kontextabhängige Registerkarten	723
13.29.6	Einfache Beschriftungen	724
13.29.7	Schaltflächen	725
13.29.8	Auswahllisten	726
13.29.9	Optionsauswahl	729
13.29.10	Texteingaben	729
13.29.11	Screen tips	730
13.29.12	Symbolleiste für den Schnellzugriff	731
13.29.13	Das RibbonWindow	731
13.29.14	Menüs	732
13.29.15	Anwendungsmenü	734
13.29.16	Alternativen	737
13.30	Chart	737
13.31	WindowsFormsHost	738
14	Wichtige WPF-Techniken	741
14.1	Eigenschaften	741
14.1.1	Abhängige Eigenschaften (Dependency Properties)	741
14.1.2	Angehängte Eigenschaften (Attached Properties)	742
14.2	Einsatz von Ressourcen	743
14.2.1	Was sind eigentlich Ressourcen?	743
14.2.2	Wo können Ressourcen gespeichert werden?	743
14.2.3	Wie definiere ich eine Ressource?	745
14.2.4	Statische und dynamische Ressourcen	746
14.2.5	Wie werden Ressourcen adressiert?	747
14.2.6	System-Ressourcen einbinden	748

14.3	Das WPF-Ereignis-Modell	748
14.3.1	Einführung	748
14.3.2	Routed Events	749
14.3.3	Direkte Events	751
14.4	Verwendung von Commands	751
14.4.1	Einführung in Commands	752
14.4.2	Verwendung vordefinierter Commands	752
14.4.3	Das Ziel des Commands	754
14.4.4	Vordefinierte Commands	755
14.4.5	Commands an Ereignismethoden binden	755
14.4.6	Wie kann ich ein Command per Code auslösen?	757
14.4.7	Command-Ausführung verhindern	757
14.5	Das WPF-Style-System	757
14.5.1	Übersicht	757
14.5.2	Benannte Styles	758
14.5.3	Typ-Styles	760
14.5.4	Styles anpassen und vererben	761
14.6	Verwenden von Triggern	763
14.6.1	Eigenschaften-Trigger (Property triggers)	763
14.6.2	Ereignis-Trigger	765
14.6.3	Daten-Trigger	766
14.7	Einsatz von Templates	767
14.7.1	Template abrufen und verändern	771
14.8	Transformationen, Animationen, StoryBoards	774
14.8.1	Transformationen	774
14.8.2	Animationen mit dem StoryBoard realisieren	779
14.9	Praxisbeispiel	783
14.9.1	Arbeiten mit Microsoft Expression Blend	783
15	WPF-Datenbindung	789
15.1	Grundprinzip	789
15.1.1	Bindungsarten	790
15.1.2	Wann wird eigentlich die Quelle aktualisiert?	791
15.1.3	Geht es auch etwas langsamer?	792
15.1.4	Bindung zur Laufzeit realisieren	793
15.2	Binden an Objekte	795
15.2.1	Objekte im Code instanziiieren	795
15.2.2	Verwenden der Instanz im VB-Quellcode	796

15.2.3	Anforderungen an die Quell-Klasse	797
15.2.4	Instanziiieren von Objekten per VB-Code	799
15.3	Binden von Collections	800
15.3.1	Anforderung an die Collection	800
15.3.2	Einfache Anzeige	801
15.3.3	Navigation zwischen den Objekten	802
15.3.4	Einfache Anzeige in einer ListBox	803
15.3.5	DataTemplates zur Anzeigeformatierung	805
15.3.6	Mehr zu List- und ComboBox	806
15.3.7	Verwenden der ListView	808
15.4	Noch einmal zurück zu den Details	810
15.4.1	Navigieren in den Daten	810
15.4.2	Sortieren	811
15.4.3	Filtern	812
15.4.4	Live Shaping	813
15.5	Anzeige von Datenbankinhalten	814
15.5.1	Datenmodell per LINQ to SQL-Designer erzeugen	814
15.5.2	Die Programm-Oberfläche	815
15.5.3	Der Zugriff auf die Daten	817
15.6	Drag & Drop-Datenbindung	818
15.6.1	Vorgehensweise	818
15.6.2	Weitere Möglichkeiten	821
15.7	Formatieren von Werten	822
15.7.1	IValueConverter	823
15.7.2	BindingBase.StringFormat-Eigenschaft	825
15.8	Das DataGrid als Universalwerkzeug	826
15.8.1	Grundlagen der Anzeige	827
15.8.2	Vom Betrachten zum Editieren	831
15.9	Praxisbeispiele	831
15.9.1	Collections in Hintergrundthreads füllen	831
15.9.2	Drag & Drop-Bindung bei 1:n-Beziehungen	835
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	Zielplattformen	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	917
18.4.3	Reagieren auf die Änderung	918
18.4.4	Angedockten Modus aktiv beenden	921
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	929
18.7.1	Symbole für WinRT-Oberflächen finden	929
18.7.2	Wie werde ich das Grufti-Layout schnell los?	930
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	944
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	ScrollViewer	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	957
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	982
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.vb	995
20.1.3	Verweise	996
20.1.4	App.xaml und App.xaml.vb	997
20.1.5	Package.appxmanifest	998
20.1.6	Application1_TemporaryKey.pfx	1002
20.1.7	MainPage.xaml & MainPage.xaml.vb	1002
20.1.8	Datentyp-Konverter/Hilfsklassen	1002
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	1019
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	1024
20.3.8	Daten in der Cloud ablegen/laden (Roaming)	1026
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	1054
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	1063
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	1072
21.2.2	Teilen von Inhalten	1079
21.2.3	Eine App als Freigabeziel verwenden	1083
21.2.4	Zugriff auf die Kontaktliste	1083
21.3	Spezielle Oberflächenelemente	1085
21.3.1	MessageDialog	1085
21.3.2	Popup-Benachrichtigungen	1088
21.3.3	PopUp/Flyouts	1096
21.3.4	Das PopupMenu einsetzen	1100
21.3.5	Eine AppBar verwenden	1102
21.4	Datenbanken und Windows Store Apps	1107
21.4.1	Der Retter in der Not: SQLite!	1107
21.4.2	Verwendung/Kurzüberblick	1107
21.4.3	Installation	1109
21.4.4	Wie kommen wir zu einer neuen Datenbank?	1111
21.4.5	Wie werden die Daten manipuliert?	1115
21.5	Vertrieb der App	1117
21.5.1	Verpacken der App	1117
21.5.2	Windows App Certification Kit	1119
21.5.3	App-Installation per Skript	1121
21.6	Ein Blick auf die App-Schwachstellen	1122
21.6.1	Quellcodes im Installationsverzeichnis	1122
21.6.2	Zugriff auf den App-Datenordner	1124

21.7	Praxisbeispiele	1124
21.7.1	Ein Verzeichnis auf Änderungen überwachen	1124
21.7.2	Eine App als Freigabeziel verwenden	1127
21.7.3	ToastNotifications einfach erzeugen	1132

Anhang

A	Glossar	1139
B	Wichtige Dateiextensions	1145
	Index	1147

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 XML-Integration in Visual Basic	1215
22.3.1 XML-Literale	1215
22.3.2 Einfaches Navigieren durch späte Bindung	1218
22.3.3 Die LINQ to XML-API	1220
22.3.4 Neue XML-Dokumente erzeugen	1221
22.3.5 Laden und Sichern von XML-Dokumenten	1223
22.3.6 Navigieren in XML-Daten	1225
22.3.7 Auswählen und Filtern	1227
22.3.8 Manipulieren der XML-Daten	1227
22.3.9 XML-Dokumente transformieren	1229
22.4 Verwendung des DOM unter .NET	1232
22.4.1 Übersicht	1232
22.4.2 DOM-Integration in Visual Basic	1233

– Diese Kapitel finden Sie nur im E-Book –

22.4.3	Laden von Dokumenten	1233
22.4.4	Erzeugen von XML-Dokumenten	1234
22.4.5	Auslesen von XML-Dateien	1236
22.4.6	Direktzugriff auf einzelne Elemente	1237
22.4.7	Einfügen von Informationen	1238
22.4.8	Suchen in den Baumzweigen	1240
22.5	Weitere Möglichkeiten der XML-Verarbeitung	1244
22.5.1	Die relationale Sicht mit XmlDocument	1244
22.5.2	XML-Daten aus Objektstrukturen erzeugen	1247
22.5.3	Schnelles Suchen in XML-Daten mit XPathNavigator	1250
22.5.4	Schnelles Auslesen von XML-Daten mit XmlReader	1253
22.5.5	Erzeugen von XML-Daten mit XmlWriter	1255
22.5.6	XML transformieren mit XSLT	1257
22.6	Praxisbeispiele	1259
22.6.1	Mit dem DOM in XML-Dokumenten navigieren	1259
22.6.2	XML-Daten in eine TreeView einlesen	1262
23	Einführung in ADO.NET	1267
23.1	Eine kleine Übersicht	1267
23.1.1	Die ADO.NET-Klassenhierarchie	1267
23.1.2	Die Klassen der Datenprovider	1268
23.1.3	Das Zusammenspiel der ADO.NET-Klassen	1271
23.2	Das Connection-Objekt	1272
23.2.1	Allgemeiner Aufbau	1272
23.2.2	OleDbConnection	1272
23.2.3	Schließen einer Verbindung	1274
23.2.4	Eigenschaften des Connection-Objekts	1274
23.2.5	Methoden des Connection-Objekts	1276
23.2.6	DerConnectionStringBuilder	1277
23.3	Das Command-Objekt	1278
23.3.1	Erzeugen und Anwenden eines Command-Objekts	1278
23.3.2	Erzeugen mittels CreateCommand-Methode	1279
23.3.3	Eigenschaften des Command-Objekts	1279
23.3.4	Methoden des Command-Objekts	1281
23.3.5	Freigabe von Connection- und Command-Objekten	1282
23.4	Parameter-Objekte	1284
23.4.1	Erzeugen und Anwenden eines Parameter-Objekts	1284
23.4.2	Eigenschaften des Parameter-Objekts	1284

23.5	Das CommandBuilder-Objekt	1285
23.5.1	Erzeugen	1285
23.5.2	Anwenden	1286
23.6	Das DataReader-Objekt	1286
23.6.1	DataReader erzeugen	1287
23.6.2	Daten lesen	1287
23.6.3	Eigenschaften des DataReaders	1288
23.6.4	Methoden des DataReaders	1288
23.7	Das DataAdapter-Objekt	1289
23.7.1	DataAdapter erzeugen	1289
23.7.2	Command-Eigenschaften	1290
23.7.3	Fill-Methode	1291
23.7.4	Update-Methode	1292
23.8	Praxisbeispiele	1293
23.8.1	Wichtige ADO.NET-Objekte im Einsatz	1293
23.8.2	Eine Aktionsabfrage ausführen	1295
23.8.3	Eine Auswahlabfrage aufrufen	1297
23.8.4	Die Datenbank aktualisieren	1299
23.8.5	Den ConnectionString speichern	1302
24	Das DataSet	1305
24.1	Grundlegende Features des DataSets	1305
24.1.1	Die Objekthierarchie	1306
24.1.2	Die wichtigsten Klassen	1306
24.1.3	Erzeugen eines DataSets	1307
24.2	Das DataTable-Objekt	1309
24.2.1	DataTable erzeugen	1309
24.2.2	Spalten hinzufügen	1309
24.2.3	Zeilen zur DataTable hinzufügen	1310
24.2.4	Auf den Inhalt einer DataTable zugreifen	1311
24.3	Die DataView	1313
24.3.1	Erzeugen eines DataView	1313
24.3.2	Sortieren und Filtern von Datensätzen	1313
24.3.3	Suchen von Datensätzen	1314
24.4	Typisierte DataSets	1314
24.4.1	Ein typisiertes DataSet erzeugen	1315
24.4.2	Das Konzept der Datenquellen	1316
24.4.3	Typisierte DataSets und TableAdapter	1317

24.5	Die Qual der Wahl	1318
24.5.1	DataReader – der schnelle Lesezugriff	1319
24.5.2	DataSet – die Datenbank im Hauptspeicher	1319
24.5.3	Objektrelationales Mapping – die Zukunft?	1320
24.6	Praxisbeispiele	1321
24.6.1	Im DataView sortieren und filtern	1321
24.6.2	Suche nach Datensätzen	1323
24.6.3	Ein DataSet in einen XML-String serialisieren	1325
24.6.4	Untypisierte in typisierte DataSets konvertieren	1329
24.6.5	Eine LINQ to SQL-Abfrage ausführen	1334
25	OOP-Spezial	1339
25.1	Eine kleine Einführung in die UML	1339
25.1.1	Use Case-Diagramm	1339
25.1.2	Use Case-Dokumentation	1341
25.1.3	Objekte identifizieren	1342
25.1.4	Statisches Modell	1343
25.1.5	Beziehungen zwischen den Klassen	1344
25.1.6	Dynamisches Modell	1344
25.1.7	Implementierung	1345
25.1.8	Test-Client	1349
25.2	Der Klassen-Designer	1352
25.2.1	Ein neues Klassendiagramm erzeugen	1352
25.2.2	Werkzeugkasten	1354
25.2.3	Enumeration	1355
25.2.4	Klasse	1356
25.2.5	Struktur	1358
25.2.6	Abstrakte Klasse	1359
25.2.7	Schnittstelle	1361
25.2.8	Delegate	1363
25.2.9	Zuordnung	1365
25.2.10	Vererbung	1366
25.2.11	Diagramme anpassen	1366
25.2.12	Wann sollten Sie den Klassen-Designer verwenden?	1367
25.3	Praxisbeispiele	1367
25.3.1	Implementierung einer Finite State Machine	1367
25.3.2	Modellierung des Bestellsystems einer Firma	1373

26	Das Microsoft Event Pattern	1387
26.1	Einführung in Design Pattern	1387
26.2	Aufbau und Bedeutung des Observer Pattern	1388
26.2.1	Subjekt und Observer	1389
26.2.2	Sequenzdiagramme	1390
26.2.3	Die Registration-Sequenz	1390
26.2.4	Die Notification-Sequenz	1391
26.2.5	Die Unregistration-Sequenz	1391
26.2.6	Bedeutung der Sequenzen für das Geschäftsmodell	1392
26.2.7	Die Rolle des Containers	1392
26.3	Implementierung mit Interfaces und Callbacks	1393
26.3.1	Übersicht und Klassendiagramm	1393
26.3.2	Schnittstellen IObserver und IObservable	1395
26.3.3	Die abstrakte Klasse Subject	1395
26.3.4	Observer1	1396
26.3.5	Observer2	1397
26.3.6	Model	1398
26.3.7	Form1	1399
26.3.8	Ein zweites Klassendiagramm	1400
26.3.9	Testen	1401
26.4	Implementierung mit Delegates und Events	1402
26.4.1	Multicast-Events	1403
26.4.2	IObserver, IObservable und Subject	1403
26.4.3	Observer1 und Observer2	1404
26.4.4	Model	1404
26.4.5	Form1	1404
26.4.6	Test und Vergleich	1405
26.4.7	Klassendiagramm	1406
26.5	Implementierung des Microsoft Event Pattern	1407
26.5.1	Namensgebung für Ereignisse	1407
26.5.2	Namensgebung und Signatur der Delegates	1407
26.5.3	Hinzufügen einer das Ereignis auslösenden Methode	1408
26.5.4	Neue Klasse NumberChangedEventArgs	1408
26.5.5	Model	1409
26.5.6	Observer1	1410
26.5.7	Form1	1410
26.6	Test und Vergleich	1411

26.7	Klassendiagramm	1411
26.8	Implementierung eines Event Pattern	1412
26.9	Praxisbeispiel	1413
26.9.1	Subjekt und Observer beobachten sich gegenseitig	1413
27	Verteilen von Anwendungen	1423
27.1	ClickOnce-Deployment	1424
27.1.1	Übersicht/Einschränkungen	1424
27.1.2	Die Vorgehensweise	1425
27.1.3	Ort der Veröffentlichung	1425
27.1.4	Anwendungsdateien	1426
27.1.5	Erforderliche Komponenten	1426
27.1.6	Aktualisierungen	1427
27.1.7	Veröffentlichungsoptionen	1428
27.1.8	Veröffentlichen	1429
27.1.9	Verzeichnisstruktur	1429
27.1.10	Der Webpublishing-Assistent	1431
27.1.11	Neue Versionen erstellen	1432
27.2	InstallShield	1432
27.2.1	Installation	1432
27.2.2	Aktivieren	1433
27.2.3	Ein neues Setup-Projekt	1433
27.2.4	Finaler Test	1441
28	Weitere Techniken	1443
28.1	Zugriff auf die Zwischenablage	1443
28.1.1	Das Clipboard-Objekt	1443
28.1.2	Zwischenablage-Funktionen für Textboxen	1445
28.2	Arbeiten mit der Registry	1445
28.2.1	Allgemeines	1446
28.2.2	Registry-Unterstützung in .NET	1447
28.3	.NET-Reflection	1449
28.3.1	Übersicht	1449
28.3.2	Assembly laden	1449
28.3.3	Mittels GetType und Type Informationen sammeln	1450
28.3.4	Dynamisches Laden von Assemblies	1452
28.4	Das SerialPort-Control	1454
28.4.1	Übersicht	1455

28.4.2	Einführungsbeispiele	1456
28.4.3	Thread-Probleme bei Windows Forms-Anwendungen	1458
28.4.4	Ein einfaches Terminalprogramm	1460
28.5	Praxisbeispiele	1465
28.5.1	Zugriff auf die Registry	1465
28.5.2	Dateiverknüpfungen erzeugen	1467
29	Konsolenanwendungen	1469
29.1	Grundaufbau/Konzepte	1469
29.1.1	Unser Hauptprogramm – Module1.vb	1470
29.1.2	Rückgabe eines Fehlerstatus	1471
29.1.3	Parameterübergabe	1472
29.1.4	Zugriff auf die Umgebungsvariablen	1474
29.2	Die Kommandozentrale: System.Console	1475
29.2.1	Eigenschaften	1475
29.2.2	Methoden/Ereignisse	1475
29.2.3	Textausgaben	1476
29.2.4	Farbangaben	1477
29.2.5	Tastaturabfragen	1478
29.2.6	Arbeiten mit Streamdaten	1479
29.3	Praxisbeispiel: Farbige Konsolenanwendung	1481

Teil VI: Windows Forms

30	Windows Forms-Anwendungen	1485
30.1	Grundaufbau/Konzepte	1485
30.1.1	Wo ist das Hauptprogramm?	1486
30.1.2	Die Oberflächendefinition – Form1.Designer.vb	1491
30.1.3	Die Spielwiese des Programmierers – Form1.vb	1492
30.1.4	Die Datei AssemblyInfo.vb	1493
30.1.5	Resources.resx/Resources.Designer.vb	1494
30.1.6	Settings.settings/Settings.Designer.vb	1495
30.2	Ein Blick auf die Application-Klasse	1497
30.2.1	Eigenschaften	1497
30.2.2	Methoden	1498
30.2.3	Ereignisse	1499

30.3	Allgemeine Eigenschaften von Komponenten	1500
30.3.1	Font	1501
30.3.2	Handle	1502
30.3.3	Tag	1503
30.3.4	Modifiers	1503
30.4	Allgemeine Ereignisse von Komponenten	1504
30.4.1	Die Eventhandler-Argumente	1504
30.4.2	Sender	1504
30.4.3	Der Parameter e	1506
30.4.4	Mausereignisse	1506
30.4.5	KeyPreview	1508
30.4.6	Weitere Ereignisse	1509
30.4.7	Validierung	1510
30.4.8	SendKeys	1510
30.5	Allgemeine Methoden von Komponenten	1511
31	Windows Forms-Formulare	1513
31.1	Übersicht	1513
31.1.1	Wichtige Eigenschaften des Form-Objekts	1514
31.1.2	Wichtige Ereignisse des Form-Objekts	1516
31.1.3	Wichtige Methoden des Form-Objekts	1517
31.2	Praktische Aufgabenstellungen	1518
31.2.1	Fenster anzeigen	1518
31.2.2	Splash Screens beim Anwendungsstart anzeigen	1521
31.2.3	Eine Sicherheitsabfrage vor dem Schließen anzeigen	1523
31.2.4	Ein Formular durchsichtig machen	1524
31.2.5	Die Tabulatorreihenfolge festlegen	1524
31.2.6	Ausrichten und Platzieren von Komponenten	1525
31.2.7	Spezielle Panels für flexibles Layout	1528
31.2.8	Menüs erzeugen	1529
31.3	MDI-Anwendungen	1533
31.3.1	"Falsche" MDI-Fenster bzw. Verwenden von Parent	1533
31.3.2	Die echten MDI-Fenster	1534
31.3.3	Die Kindfenster	1535
31.3.4	Automatisches Anordnen der Kindfenster	1536
31.3.5	Zugriff auf die geöffneten MDI-Kindfenster	1538
31.3.6	Zugriff auf das aktive MDI-Kindfenster	1538
31.3.7	Mischen von Kindfenstermenü/MDIContainer-Menü	1538

31.4	Praxisbeispiele	1539
31.4.1	Informationsaustausch zwischen Formularen	1539
31.4.2	Ereigniskette beim Laden/Entladen eines Formulars	1547
32	Komponenten-Übersicht	1553
32.1	Allgemeine Hinweise	1553
32.1.1	Hinzufügen von Komponenten	1553
32.1.2	Komponenten zur Laufzeit per Code erzeugen	1554
32.2	Allgemeine Steuerelemente	1556
32.2.1	Label	1556
32.2.2	LinkLabel	1557
32.2.3	Button	1558
32.2.4	TextBox	1559
32.2.5	MaskedTextBox	1562
32.2.6	CheckBox	1563
32.2.7	RadioButton	1565
32.2.8	ListBox	1566
32.2.9	CheckedListBox	1567
32.2.10	ComboBox	1568
32.2.11	PictureBox	1569
32.2.12	DateTimePicker	1569
32.2.13	MonthCalendar	1570
32.2.14	HScrollBar, VScrollBar	1570
32.2.15	TrackBar	1571
32.2.16	NumericUpDown	1572
32.2.17	DomainUpDown	1573
32.2.18	ProgressBar	1573
32.2.19	RichTextBox	1574
32.2.20	ListView	1575
32.2.21	TreeView	1581
32.2.22	WebBrowser	1586
32.3	Container	1587
32.3.1	FlowLayout/TableLayout/SplitContainer	1587
32.3.2	Panel	1587
32.3.3	GroupBox	1588
32.3.4	TabControl	1589
32.3.5	ImageList	1591

32.4	Menüs & Symbolleisten	1592
32.4.1	MenuStrip und ContextMenuStrip	1592
32.4.2	ToolStrip	1592
32.4.3	StatusStrip	1592
32.4.4	ToolStripContainer	1593
32.5	Daten	1593
32.5.1	DataSet	1593
32.5.2	DataGridView/DataGrid	1594
32.5.3	BindingNavigator/BindingSource	1594
32.5.4	Chart	1594
32.6	Komponenten	1595
32.6.1	ErrorProvider	1596
32.6.2	HelpProvider	1596
32.6.3	ToolTip	1596
32.6.4	Timer	1596
32.6.5	BackgroundWorker	1596
32.6.6	SerialPort	1597
32.7	Drucken	1597
32.7.1	PrintPreviewControl	1597
32.7.2	PrintDocument	1597
32.8	Dialoge	1597
32.8.1	OpenFileDialog/SaveFileDialog/FolderBrowserDialog	1597
32.8.2	FontDialog/ColorDialog	1597
32.9	WPF-Unterstützung mit dem ElementHost	1598
32.10	Praxisbeispiele	1598
32.10.1	Mit der CheckBox arbeiten	1598
32.10.2	Steuerelemente per Code selbst erzeugen	1600
32.10.3	Controls-Auflistung im TreeView anzeigen	1602
32.10.4	WPF-Komponenten mit dem ElementHost anzeigen	1605
33	Grundlagen der Grafikausgabe	1611
33.1	Übersicht und erste Schritte	1611
33.1.1	GDI+ – Ein erster Blick für Umsteiger	1612
33.1.2	Namespaces für die Grafikausgabe	1613
33.2	Darstellen von Grafiken	1615
33.2.1	Die PictureBox-Komponente	1615
33.2.2	Das Image-Objekt	1616
33.2.3	Laden von Grafiken zur Laufzeit	1617

33.2.4	Sichern von Grafiken	1617
33.2.5	Grafikeigenschaften ermitteln	1618
33.2.6	Erzeugen von Vorschaugrafiken (Thumbnails)	1619
33.2.7	Die Methode RotateFlip	1620
33.2.8	Skalieren von Grafiken	1621
33.3	Das .NET-Koordinatensystem	1622
33.3.1	Globale Koordinaten	1623
33.3.2	Seitenkoordinaten (globale Transformation)	1624
33.3.3	Gerätekoordinaten (Seitentransformation)	1626
33.4	Grundlegende Zeichenfunktionen von GDI+	1627
33.4.1	Das zentrale Graphics-Objekt	1627
33.4.2	Punkte zeichnen/abfragen	1630
33.4.3	Linien	1631
33.4.4	Kantenglättung mit Antialiasing	1632
33.4.5	PolyLine	1633
33.4.6	Rechtecke	1633
33.4.7	Polygone	1635
33.4.8	Splines	1635
33.4.9	Bézierkurven	1637
33.4.10	Kreise und Ellipsen	1638
33.4.11	Tortestück (Segment)	1638
33.4.12	Bogenstück	1640
33.4.13	Wo sind die Rechtecke mit den "runden Ecken"?	1640
33.4.14	Textausgabe	1642
33.4.15	Ausgabe von Grafiken	1646
33.5	Unser Werkzeugkasten	1647
33.5.1	Einfache Objekte	1647
33.5.2	Vordefinierte Objekte	1648
33.5.3	Farben/Transparenz	1650
33.5.4	Stifte (Pen)	1652
33.5.5	Pinsel (Brush)	1655
33.5.6	SolidBrush	1655
33.5.7	HatchBrush	1655
33.5.8	TextureBrush	1657
33.5.9	LinearGradientBrush	1657
33.5.10	PathGradientBrush	1659
33.5.11	Fonts	1660

33.5.12	Path-Objekt	1661
33.5.13	Clipping/Region	1664
33.6	Standarddialoge	1667
33.6.1	Schriftauswahl	1667
33.6.2	Farbauswahl	1668
33.7	Praxisbeispiele	1670
33.7.1	Ein Graphics-Objekt erzeugen	1670
33.7.2	Zeichenoperationen mit der Maus realisieren	1673
34	Druckausgabe	1677
34.1	Einstieg und Übersicht	1677
34.1.1	Nichts geht über ein Beispiel	1677
34.1.2	Programmiermodell	1679
34.1.3	Kurzübersicht der Objekte	1680
34.2	Auswerten der Druckereinstellungen	1680
34.2.1	Die vorhandenen Drucker	1680
34.2.2	Der Standarddrucker	1681
34.2.3	Verfügbare Papierformate/Seitenabmessungen	1681
34.2.4	Der eigentliche Druckbereich	1683
34.2.5	Die Seitenausrichtung ermitteln	1683
34.2.6	Ermitteln der Farbfähigkeit	1684
34.2.7	Die Druckauflösung abfragen	1684
34.2.8	Ist beidseitiger Druck möglich?	1685
34.2.9	Einen "Informationsgerätekontext" erzeugen	1685
34.2.10	Abfragen von Werten während des Drucks	1686
34.3	Festlegen von Druckereinstellungen	1687
34.3.1	Einen Drucker auswählen	1687
34.3.2	Drucken in Millimetern	1687
34.3.3	Festlegen der Seitenränder	1688
34.3.4	Druckjobname	1689
34.3.5	Die Anzahl der Kopien festlegen	1689
34.3.6	Beidseitiger Druck	1690
34.3.7	Seitenzahlen festlegen	1691
34.3.8	Druckqualität verändern	1694
34.3.9	Ausgabemöglichkeiten des Chart-Controls nutzen	1694
34.4	Die Druckdialoge verwenden	1695
34.4.1	PrintDialog	1695
34.4.2	PageSetupDialog	1696

34.4.3	PrintPreviewDialog	1698
34.4.4	Ein eigenes Druckvorschau-Fenster realisieren	1699
34.5	Drucken mit OLE-Automation	1700
34.5.1	Kurzeinstieg in die OLE-Automation	1700
34.5.2	Drucken mit Microsoft Word	1703
34.6	Praxisbeispiele	1705
34.6.1	Den Drucker umfassend konfigurieren	1705
34.6.2	Diagramme mit dem Chart-Control drucken	1715
34.6.3	Drucken mit Word	1717
35	Windows Forms-Datenbindung	1723
35.1	Prinzipielle Möglichkeiten	1723
35.2	Manuelle Bindung an einfache Datenfelder	1724
35.2.1	BindingSource erzeugen	1724
35.2.2	Binding-Objekt	1725
35.2.3	DataBindings-Collection	1725
35.3	Manuelle Bindung an Listen und Tabellen	1725
35.3.1	DataGridView	1726
35.3.2	Datenbindung von ComboBox und ListBox	1726
35.4	Entwurfszeit-Bindung an typisierte DataSets	1726
35.5	Drag & Drop-Datenbindung	1728
35.6	Navigations- und Bearbeitungsfunktionen	1728
35.6.1	Navigieren zwischen den Datensätzen	1728
35.6.2	Hinzufügen und Löschen	1728
35.6.3	Aktualisieren und Abbrechen	1729
35.6.4	Verwendung des BindingNavigators	1729
35.7	Die Anzeigedaten formatieren	1730
35.8	Praxisbeispiele	1730
35.8.1	Einrichten und Verwenden einer Datenquelle	1730
35.8.2	Eine Auswahlabfrage im DataGridView anzeigen	1734
35.8.3	Master-Detailbeziehungen im DataGrid anzeigen	1737
35.8.4	Datenbindung Chart-Control	1738
36	Erweiterte Grafikausgabe	1743
36.1	Transformieren mit der Matrix-Klasse	1743
36.1.1	Übersicht	1743
36.1.2	Translation	1744
36.1.3	Skalierung	1744

36.1.4	Rotation	1745
36.1.5	Scherung	1745
36.1.6	Zuweisen der Matrix	1746
36.2	Low-Level-Grafikmanipulationen	1746
36.2.1	Worauf zeigt Scan0?	1747
36.2.2	Anzahl der Spalten bestimmen	1748
36.2.3	Anzahl der Zeilen bestimmen	1749
36.2.4	Zugriff im Detail (erster Versuch)	1749
36.2.5	Zugriff im Detail (zweiter Versuch)	1751
36.2.6	Invertieren	1753
36.2.7	In Graustufen umwandeln	1754
36.2.8	Heller/Dunkler	1755
36.2.9	Kontrast	1756
36.2.10	Gamma-Wert	1757
36.2.11	Histogramm spreizen	1758
36.2.12	Ein universeller Grafikfilter	1760
36.3	Fortgeschrittene Techniken	1764
36.3.1	Flackerfrei dank Double Buffering	1764
36.3.2	Animationen	1766
36.3.3	Animated GIFs	1769
36.3.4	Auf einzelne GIF-Frames zugreifen	1772
36.3.5	Transparenz realisieren	1773
36.3.6	Eine Grafik maskieren	1775
36.3.7	JPEG-Qualität beim Sichern bestimmen	1776
36.4	Grundlagen der 3D-Vektorgrafik	1777
36.4.1	Datentypen für die Verwaltung	1778
36.4.2	Eine universelle 3D-Grafik-Klasse	1779
36.4.3	Grundlegende Betrachtungen	1780
36.4.4	Translation	1783
36.4.5	Streckung/Skalierung	1783
36.4.6	Rotation	1784
36.4.7	Die eigentlichen Zeichenroutinen	1786
36.5	Und doch wieder GDI-Funktionen	1789
36.5.1	Am Anfang war das Handle	1789
36.5.2	Gerätekontext (Device Context Types)	1791
36.5.3	Koordinatensysteme und Abbildungsmodi	1793
36.5.4	Zeichenwerkzeuge/Objekte	1798
36.5.5	Bitmaps	1800

36.6	Praxisbeispiele	1804
36.6.1	Die Transformationsmatrix verstehen	1804
36.6.2	Eine 3D-Grafikausgabe in Aktion	1807
36.6.3	Einen Fenster-Screenshot erzeugen	1810
37	Ressourcen/Lokalisierung	1813
37.1	Manifestressourcen	1813
37.1.1	Erstellen von Manifestressourcen	1813
37.1.2	Zugriff auf Manifestressourcen	1815
37.2	Typisierte Ressourcen	1816
37.2.1	Erzeugen von .resources-Dateien	1816
37.2.2	Hinzufügen der .resources-Datei zum Projekt	1817
37.2.3	Zugriff auf die Inhalte von .resources-Dateien	1817
37.2.4	ResourceManager direkt aus der .resources-Datei erzeugen	1818
37.2.5	Was sind .resx-Dateien?	1819
37.3	Streng typisierte Ressourcen	1819
37.3.1	Erzeugen streng typisierter Ressourcen	1819
37.3.2	Verwenden streng typisierter Ressourcen	1820
37.3.3	Streng typisierte Ressourcen per Reflection auslesen	1820
37.4	Anwendungen lokalisieren	1823
37.4.1	Localizable und Language	1823
37.4.2	Beispiel "Landesfahnen"	1823
37.4.3	Einstellen der aktuellen Kultur zur Laufzeit	1826
37.5	Praxisbeispiel	1828
37.5.1	Betrachter für Manifestressourcen	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	1837
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)	1840
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	1843
38.5.10	Wertebereichsbeschränkung und Fehlerprüfung	1844
38.5.11	Eigenschaften von Aufzählungstypen	1845
38.5.12	Standard Objekt-Eigenschaften	1846
38.5.13	Eigene Objekt-Eigenschaften	1847
38.6	Methoden	1849
38.6.1	Konstruktor	1850
38.6.2	Class-Konstruktor	1851
38.6.3	Destruktor	1852
38.6.4	Aufruf des Basisklassen-Konstruktors	1853
38.6.5	Aufruf von Basisklassen-Methoden	1853
38.7	Ereignisse (Events)	1853
38.7.1	Ereignis mit Standardargument definieren	1854
38.7.2	Ereignis mit eigenen Argumenten	1855
38.7.3	Ein Default-Ereignis festlegen	1856
38.7.4	Mit Ereignissen auf Windows-Messages reagieren	1856
38.8	Weitere Themen	1858
38.8.1	Wohin mit der Komponente?	1858
38.8.2	Assembly-Informationen festlegen	1859
38.8.3	Assemblies signieren	1861
38.8.4	Komponenten-Ressourcen einbetten	1862
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	1868
38.9.1	AnimGif für die Anzeige von Animationen	1868
38.9.2	Eine FontComboBox entwickeln	1870
38.9.3	Das PropertyGrid verwenden	1873

Teil VII: ASP.NET

39	ASP.NET-Einführung	1877
39.1	ASP.NET für Ein- und Umsteiger	1877
39.1.1	ASP – der 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 Visual Basic zu tun?	1882
39.2	Unsere erste Web-Anwendung	1885
39.2.1	Visueller Entwurf der Bedienoberfläche	1885
39.2.2	Zuweisen der Objekteigenschaften	1887
39.2.3	Verknüpfen der Objekte mit Ereignissen	1888
39.2.4	Programm kompilieren und testen	1890
39.3	Die ASP.NET-Projektdateien	1891
39.3.1	ASP.NET-Website	1892
39.3.2	ASP.NET-Web-Anwendungen	1893
39.3.3	ASPX-Datei(en)	1893
39.3.4	Die aspx.vb-Datei(en)	1896
39.3.5	Die Datei Global.asax	1896
39.3.6	Das Startformular	1897
39.3.7	Die Datei Web.config	1897
39.3.8	Masterpages (master-Dateien)	1900
39.3.9	Sitemap (Web.sitemap)	1900
39.3.10	Benutzersteuerelemente (ascx-Dateien)	1901
39.3.11	Die Web-Projekt-Verzeichnisse	1901
39.4	Lernen am Beispiel	1902
39.4.1	Erstellen des Projekts	1902
39.4.2	Oberflächengestaltung	1903
39.4.3	Ereignisprogrammierung	1904
39.4.4	Ein Fehler, was nun?	1905
39.4.5	Ereignisse von Textboxen	1907
39.4.6	Ein gemeinsamer Ereignis-Handler	1907
39.4.7	Eingabefokus setzen	1908
39.4.8	Ausgaben in einer Tabelle	1908
39.4.9	Scrollen der Anzeige	1911
39.4.10	Zusammenspiel mehrerer Formulare	1911

39.4.11	Umleiten bei Direktaufruf	1913
39.4.12	Ärger mit den Cookies	1914
39.4.13	Export auf den IIS	1915
39.5	Tipps & Tricks	1916
39.5.1	Nachinstallieren IIS 7 bzw. 7.5 (Windows 7)	1916
39.5.2	Nachinstallieren IIS8 (Windows 8)	1917
40	Übersicht ASP.NET-Controls	1919
40.1	Einfache Steuerelemente im Überblick	1919
40.1.1	Label	1919
40.1.2	TextBox	1921
40.1.3	Button, ImageButton, LinkButton	1922
40.1.4	CheckBox, RadioButton	1923
40.1.5	CheckBoxList, BulletList, RadioButtonList	1924
40.1.6	Table	1925
40.1.7	Hyperlink	1927
40.1.8	Image, ImageMap	1927
40.1.9	Calendar	1929
40.1.10	Panel	1930
40.1.11	HiddenField	1930
40.1.12	Substitution	1931
40.1.13	XML	1932
40.1.14	FileUpload	1934
40.1.15	AdRotator	1935
40.2	Steuerelemente für die Seitennavigation	1936
40.2.1	Mehr Übersicht mit Web.Sitemap	1936
40.2.2	Menu	1938
40.2.3	TreeView	1941
40.2.4	SiteMapPath	1944
40.2.5	MultiView, View	1945
40.2.6	Wizard	1946
40.3	Webseitenlayout/-design	1948
40.3.1	Masterpages	1948
40.3.2	Themes/Skins	1951
40.3.3	Webparts	1954
40.4	Die Validator-Controls	1955
40.4.1	Übersicht	1955
40.4.2	Wo findet die Fehlerprüfung statt?	1956

40.4.3	Verwendung	1956
40.4.4	RequiredFieldValidator	1957
40.4.5	CompareValidator	1958
40.4.6	RangeValidator	1960
40.4.7	RegularExpressionValidator	1960
40.4.8	CustomValidator	1961
40.4.9	ValidationSummary	1964
40.4.10	Weitere Möglichkeiten	1965
40.5	Praxisbeispiele	1965
40.5.1	Themes und Skins verstehen	1965
40.5.2	Masterpages verwenden	1970
40.5.3	Webparts verwenden	1973
41	ASP.NET-Datenbindung	1979
41.1	Einstiegsbeispiel	1979
41.1.1	Erstellen der ASP.NET-Website	1979
41.2	Einführung	1984
41.2.1	Konzept	1984
41.2.2	Übersicht DataSource-Steuerelemente	1985
41.3	SQLDataSource	1986
41.3.1	Datenauswahl mit Parametern	1988
41.3.2	Parameter für INSERT, UPDATE und DELETE	1989
41.3.3	Methoden	1991
41.3.4	Caching	1992
41.3.5	Aktualisieren/Refresh	1993
41.4	AccessDataSource	1993
41.5	ObjectDataSource	1993
41.5.1	Verbindung zwischen Objekt und DataSource	1993
41.5.2	Ein Beispiel sorgt für Klarheit	1995
41.5.3	Geschäftsobjekte in einer Session verwalten	1999
41.6	SitemapDataSource	2001
41.7	LinqDataSource	2002
41.7.1	Bindung von einfachen Collections	2002
41.7.2	Bindung eines LINQ to SQL-DataContext	2003
41.8	EntityDataSource	2005
41.8.1	Entity Data Model erstellen	2006
41.8.2	EntityDataSource anbinden	2008
41.8.3	Datenmenge filtern	2011

41.9	XmlDataSource	2011
41.10	QueryExtender	2013
41.10.1	Grundlagen	2013
41.10.2	Suchen	2014
41.10.3	Sortieren	2016
41.11	GridView	2017
41.11.1	Auswahlfunktion (Zeilenauswahl)	2017
41.11.2	Auswahl mit mehrspaltigem Index	2018
41.11.3	Hyperlink-Spalte für Detailansicht	2018
41.11.4	Spalten erzeugen	2019
41.11.5	Paging realisieren	2020
41.11.6	Edit, Update, Delete	2022
41.11.7	Keine Daten, was tun?	2023
41.12	DetailsView	2023
41.13	FormView	2025
41.14	DataList	2028
41.14.1	Bearbeitungsfunktionen implementieren	2028
41.14.2	Layout verändern	2030
41.15	Repeater	2030
41.16	ListView	2032
41.17	Typisierte Datenbindung	2032
41.18	Model Binding	2033
41.19	Chart	2035
42	ASP.NET-Objekte/-Techniken	2037
42.1	Wichtige ASP.NET-Objekte	2037
42.1.1	HTTPApplication	2037
42.1.2	Application	2040
42.1.3	Session	2041
42.1.4	Page	2043
42.1.5	Request	2045
42.1.6	Response	2048
42.1.7	Server	2053
42.1.8	Cookies verwenden	2054
42.2	ASP.NET-Fehlerbehandlung	2056
42.2.1	Fehler beim Entwurf	2056
42.2.2	Laufzeitfehler	2057
42.2.3	Eine eigene Fehlerseite	2058

42.2.4	Fehlerbehandlung im Web Form	2059
42.2.5	Fehlerbehandlung in der Anwendung	2060
42.2.6	Alternative Fehlerseite einblenden	2061
42.2.7	Lokale Fehlerbehandlung	2062
42.2.8	Seite nicht gefunden! – Was nun?	2063
42.3	E-Mail-Versand in ASP.NET	2063
42.3.1	Übersicht	2063
42.3.2	Mail-Server bestimmen	2064
42.3.3	Einfache Text-E-Mails versenden	2065
42.3.4	E-Mails mit Dateianhang	2066
42.4	Sicherheit von Webanwendungen	2067
42.4.1	Authentication	2067
42.4.2	Forms Authentication realisieren	2069
42.4.3	Impersonation	2072
42.4.4	Authorization	2073
42.4.5	Administrieren der Website	2076
42.4.6	Steuerelemente für das Login-Handling	2079
42.4.7	Programmieren der Sicherheitseinstellungen	2084
42.5	AJAX in ASP.NET-Anwendungen	2086
42.5.1	Was ist eigentlich AJAX und was kann es?	2086
42.5.2	Die AJAX-Controls	2087
42.5.3	AJAX-Control-Toolkit	2090
42.6	User Controls/Webbenutzersteuerelemente	2092
42.6.1	Ein simples Einstiegsbeispiel	2093
42.6.2	Dynamische Grafiken im User Control anzeigen	2096
42.6.3	Grafikausgaben per User Control realisieren	2101

Teil VIII: Silverlight

43	Silverlight-Entwicklung	2107
43.1	Einführung	2107
43.1.1	Zielplattformen	2108
43.1.2	Silverlight-Applikationstypen	2108
43.1.3	Wichtige Unterschiede zu den WPF-Anwendungen	2110
43.1.4	Vor- und Nachteile von Silverlight-Anwendungen	2112
43.1.5	Entwicklungstools	2114
43.1.6	Installation auf dem Client	2114

43.2	Die Silverlight-Anwendung im Detail	2115
43.2.1	Ein kleines Beispielprojekt	2116
43.2.2	Das Application Package und das Test-Web	2118
43.3	Die Projektdateien im Überblick	2121
43.3.1	Projektverwaltung mit App.xaml & App.xaml.vb	2122
43.3.2	MainPage.xaml & MainPage.xaml.vb	2124
43.3.3	AssemblyInfo.vb	2125
43.4	Fenster und Seiten in Silverlight	2125
43.4.1	Das Standardfenster	2125
43.4.2	Untergeordnete Silverlight-Fenster	2126
43.4.3	UserControls für die Anzeige von Detaildaten	2128
43.4.4	Echte Windows	2129
43.4.5	Navigieren in Silverlight-Anwendungen	2130
43.5	Datenbanken/Datenbindung	2135
43.5.1	ASP.NET-Webdienste/WCF-Dienste	2136
43.5.2	WCF Data Services	2144
43.6	Isolierter Speicher	2155
43.6.1	Grundkonzept	2155
43.6.2	Das virtuelle Dateisystem verwalten	2156
43.6.3	Arbeiten mit Dateien	2159
43.7	Fulltrust-Anwendungen	2160
43.8	Praxisbeispiele	2163
43.8.1	Eine Out-of-Browser-Applikation realisieren	2163
43.8.2	Out-of-Browser-Anwendung aktualisieren	2167
43.8.3	Testen auf aktive Internetverbindung	2168
43.8.4	Auf Out-of-Browser-Anwendung testen	2169
43.8.5	Den Browser bestimmen	2169
43.8.6	Parameter an das Plug-in übergeben	2170
43.8.7	Auf den QueryString zugreifen	2172
43.8.8	Timer in Silverlight nutzen	2173
43.8.9	Dateien lokal speichern	2174
43.8.10	Drag & Drop realisieren	2175
43.8.11	Auf die Zwischenablage zugreifen	2177
43.8.12	Weitere Fenster öffnen	2179
Index		2183