

Inhaltsverzeichnis

HINWEIS: Die Bonuskapitel 16, 17, 27 und 37-40 finden sich als PDF-Datei auf der beiliegenden CD. Im Inhaltsverzeichnis sind sie durch ein CD-Symbol gekennzeichnet.

Vorwort	45
----------------------	-----------

Teil I: Grundlagen

1 Einstieg in Visual Studio 2010	51
1.1 Die Installation von Visual Studio 2010	51
1.1.1 Überblick über die Produktpalette	51
1.1.2 Anforderungen an Hard- und Software	53
1.1.3 Installation der Express Edition	53
1.1.4 Installation von Visual Studio 2010 Ultimate	54
1.2 Unser allererstes C#-Programm	55
1.2.1 Vorbereitungen	55
1.2.2 Programm schreiben	57
1.2.3 Programm kompilieren und testen	57
1.2.4 Einige Erläuterungen zum Quellcode	58
1.2.5 Konsolenanwendungen sind langweilig	59
1.3 Die Windows-Philosophie	59
1.3.1 Mensch-Rechner-Dialog	60
1.3.2 Objekt- und ereignisorientierte Windows-Programmierung	60
1.3.3 Windows-Programmierung unter Visual Studio 2010	61
1.4 Die Entwicklungsumgebung von Visual Studio 2010	63
1.4.1 Der Startdialog	63
1.4.2 Die wichtigsten Fenster	64

1.5	Microsofts .NET-Technologie	67
1.5.1	Zur Geschichte von .NET	67
1.5.2	.NET-Features und Begriffe	69
1.6	Wichtige Neuigkeiten in Visual Studio 2010	77
1.6.1	Die verschiedenen Pakete	77
1.6.2	Die neue Visual C# Entwicklungsumgebung	77
1.6.3	Neuheiten im .NET Framework 4.0	79
1.6.4	C# 4.0 Sprache und Compiler	80
1.7	Praxisbeispiele	81
1.7.1	Windows-Anwendung für Einsteiger	81
1.7.2	Windows-Anwendung für fortgeschrittene Einsteiger	86
2	Grundlagen der Sprache C#	95
2.1	Grundbegriffe	95
2.1.1	Anweisungen	95
2.1.2	Bezeichner	96
2.1.3	Schlüsselwörter	97
2.1.4	Kommentare	97
2.2	Datentypen, Variablen und Konstanten	98
2.2.1	Fundamentale Typen	98
2.2.2	Benennung von Variablen	100
2.2.3	Deklaration von Variablen	100
2.2.4	Typsuffixe	102
2.2.5	Zeichen und Zeichenketten	102
2.2.6	object-Datentyp	103
2.2.7	Konstanten deklarieren	104
2.2.8	Nullable Types	104
2.2.9	Typinferenz	105
2.2.10	Gültigkeitsbereiche und Sichtbarkeit	106
2.3	Operatoren	107
2.3.1	Arithmetische Operatoren	107
2.3.2	Zuweisungsoperatoren	109
2.3.3	Logische Operatoren	110
2.3.4	Rangfolge der Operatoren	112
2.4	Konvertieren von Datentypen	113
2.4.1	Implizite und explizite Konvertierung	113
2.4.2	Welcher Datentyp passt zu welchem?	115
2.4.3	Konvertieren von bool	116

2.4.4	Konvertieren von string	116
2.4.5	Die Convert-Klasse	118
2.4.6	Die Parse-Methode	118
2.4.7	Boxing und Unboxing	119
2.5	Kontrollstrukturen	121
2.5.1	Verzweigungsbefehle	121
2.5.2	Schleifenanweisungen	124
2.6	Benutzerdefinierte Datentypen	127
2.6.1	Enumerationen	127
2.6.2	Strukturen	128
2.7	Nutzerdefinierte Methoden	131
2.7.1	Methoden mit Rückgabewert	131
2.7.2	Methoden ohne Rückgabewert	132
2.7.3	Parameterübergabe mit ref	133
2.7.4	Parameterübergabe mit out	135
2.7.5	Methodenüberladung	135
2.7.6	Optionale Parameter	136
2.7.7	Benannte Parameter	138
2.8	Praxisbeispiele	139
2.8.1	Vom PAP zur Konsolenanwendung	139
2.8.2	Ein Konsolen- in ein Windows-Programm verwandeln	141
2.8.3	Schleifenanweisungen verstehen	143
2.8.4	Benutzerdefinierte Methoden überladen	145
3	Objektorientiertes Programmieren	149
3.1	Kleine Einführung in die OOP	149
3.1.1	Zur Entwicklung der objektorientierten Programmierung	149
3.1.2	Grundbegriffe der OOP	151
3.1.3	Sichtbarkeit von Klassen und ihren Mitgliedern	153
3.1.4	Allgemeiner Aufbau einer Klasse	154
3.1.5	Das Erzeugen eines Objekts	155
3.1.6	Einführungsbeispiel	158
3.2	Eigenschaften	163
3.2.1	Eigenschaften mit Zugriffsmethoden kapseln	163
3.2.2	Berechnete Eigenschaften	165
3.2.3	Lese-/Schreibschutz	167
3.2.4	Property-Accessoren	168

3.2.5	Statische Felder/Eigenschaften	168
3.2.6	Einfache Eigenschaften automatisch implementieren	171
3.3	Methoden	172
3.3.1	Öffentliche und private Methoden	172
3.3.2	Überladene Methoden	173
3.3.3	Statische Methoden	173
3.4	Ereignisse	175
3.4.1	Ereignis-Delegate hinzufügen	176
3.4.2	Ereignisse verwenden	179
3.5	Arbeiten mit Konstruktor und Destruktor	182
3.5.1	Konstruktor und Objektinitialisierer	182
3.5.2	Destruktor und Garbage Collector	185
3.5.3	Mit using den Lebenszyklus des Objekts kapseln	188
3.5.4	Verzögerte Initialisierung	188
3.6	Vererbung und Polymorphie	189
3.6.1	Klassendiagramm	189
3.6.2	Method-Overriding	190
3.6.3	Klassen implementieren	191
3.6.4	Implementieren der Objekte	194
3.6.5	Allgemeine Hinweise und Regeln zur Vererbung	196
3.6.6	Polymorphes Verhalten	197
3.6.7	Die Rolle von System.Object	200
3.7	Spezielle Klassen	201
3.7.1	Abstrakte Klassen	201
3.7.2	Versiegelte Klassen	202
3.7.3	Partielle Klassen	203
3.7.4	Statische Klassen	204
3.8	Schnittstellen (Interfaces)	205
3.8.1	Definition einer Schnittstelle	205
3.8.2	Implementieren einer Schnittstelle	206
3.8.3	Abfragen, ob Schnittstelle vorhanden ist	207
3.8.4	Mehrere Schnittstellen implementieren	207
3.8.5	Schnittstellenprogrammierung ist ein weites Feld	208
3.9	Praxisbeispiele	208
3.9.1	Eigenschaften sinnvoll kapseln	208
3.9.2	Eine statische Klasse anwenden	211
3.9.3	Vom fetten zum dünnen Client	213
3.9.4	Schnittstellenvererbung verstehen	225

4	Arrays, Strings und Funktionen	231
4.1	Datenfelder (Arrays)	231
4.1.1	Array deklarieren	231
4.1.2	Array instanziiieren	232
4.1.3	Array initialisieren	232
4.1.4	Zugriff auf Array-Elemente	233
4.1.5	Zugriff mittels Schleife	234
4.1.6	Mehrdimensionale Arrays	235
4.1.7	Zuweisen von Arrays	237
4.1.8	Arrays aus Strukturvariablen	238
4.1.9	Löschen und Umdimensionieren von Arrays	239
4.1.10	Eigenschaften und Methoden von Arrays	240
4.1.11	Übergabe von Arrays	242
4.2	Verarbeiten von Zeichenketten	243
4.2.1	Zuweisen von Strings	243
4.2.2	Instanzeigenschaften und -methoden eines Strings	244
4.2.3	Wichtige Methoden der String-Klasse	246
4.3	Formatieren von Zahlen	248
4.3.1	Anwenden der ToString-Methode	248
4.3.2	Anwenden der Format-Methode	250
4.4	Berechnungen mit vordefinierten Funktionen	252
4.4.1	Mathematik	252
4.4.2	Datums- und Zeitfunktionen	254
4.5	Praxisbeispiele	257
4.5.1	Zeichenketten verarbeiten	257
4.5.2	Methodenaufrufe mit Array-Parametern	260
5	Weitere wichtige Sprachfeatures	263
5.1	Namespaces (Namensräume)	263
5.1.1	Ein kleiner Überblick	263
5.1.2	Einen eigenen Namespace einrichten	265
5.1.3	Die using-Anweisung	266
5.1.4	Namespace Alias	266
5.1.5	Namespace Alias Qualifizierer	267
5.2	Operatorenüberladung	267
5.2.1	Syntaxregeln	268
5.2.2	Praktische Anwendung	268

5.3	Collections (Auflistungen)	269
5.3.1	Die Schnittstelle IEnumerable	269
5.3.2	ArrayList	272
5.3.3	Hashtable	273
5.3.4	Indexer	273
5.4	Generics	275
5.4.1	Klassische Vorgehensweise	276
5.4.2	Generics bieten Typsicherheit	277
5.4.3	Generische Methoden	278
5.4.4	Iteratoren	279
5.5	Generische Collections	280
5.5.1	List-Collection statt ArrayList	280
5.5.2	Vorteile generischer Collections	281
5.5.3	Constraints	281
5.6	Das Prinzip der Delegates	281
5.6.1	Delegates sind Methodenzeiger	282
5.6.2	Einen Delegate-Typ deklarieren	282
5.6.3	Ein Delegate-Objekt erzeugen	282
5.6.4	Delegates vereinfacht instanziiieren	284
5.6.5	Anonyme Methoden	285
5.6.6	Lambda-Ausdrücke	286
5.6.7	Lambda-Ausdrücke in der Task Parallel Library	288
5.7	Dynamische Programmierung	290
5.7.1	Wozu dynamische Programmierung?	290
5.7.2	Das Prinzip der dynamischen Programmierung	290
5.7.3	Optionale Parameter sind hilfreich	293
5.7.4	Kovarianz und Kontravarianz	294
5.8	Weitere Datentypen	294
5.8.1	BigInteger	294
5.8.2	Complex	297
5.8.3	Tuple< >	297
5.8.4	SortedSet< >	298
5.9	Praxisbeispiele	299
5.9.1	ArrayList versus generische List	299
5.9.2	Ein generisches IEnumerable-Interface implementieren	303
5.9.3	Delegates, anonyme Methoden und Lambda Expressions	306
5.9.4	Dynamischer Zugriff auf COM Interop	310

6	Einführung in LINQ	313
6.1	LINQ-Grundlagen	313
6.1.1	Die LINQ-Architektur	313
6.1.2	Anonyme Typen	314
6.1.3	Erweiterungsmethoden	316
6.2	Abfragen mit LINQ to Objects	317
6.2.1	Grundlegendes zur LINQ-Syntax	317
6.2.2	Zwei alternative Schreibweisen von LINQ Abfragen	318
6.2.3	Übersicht der wichtigsten Abfrage-Operatoren	319
6.3	LINQ-Abfragen im Detail	320
6.3.1	Die Projektionsoperatoren Select und SelectMany	321
6.3.2	Der Restriktionsoperator Where	322
6.3.3	Die Sortierungsoperatoren OrderBy und ThenBy	323
6.3.4	Der Gruppierungsoperator GroupBy	324
6.3.5	Verknüpfen mit Join	327
6.3.6	Aggregat-Operatoren	327
6.3.7	Verzögertes Ausführen von LINQ-Abfragen	329
6.3.8	Konvertierungsmethoden	330
6.3.9	Abfragen mit PLINQ	330
6.4	Praxisbeispiele	333
6.4.1	Die Syntax von LINQ-Abfragen verstehen	333
6.4.2	Aggregat-Abfragen mit LINQ	336

Teil II: Technologien

7	Zugriff auf das Dateisystem	341
7.1	Grundlagen	341
7.1.1	Klassen für Verzeichnis- und Dateioperationen	342
7.1.2	Statische versus Instanzen-Klasse	342
7.2	Operationen auf Verzeichnisebene	343
7.2.1	Verzeichnisse erzeugen und löschen	343
7.2.2	Verzeichnisse verschieben und umbenennen	344
7.2.3	Aktuelles Verzeichnis bestimmen	344
7.2.4	Unterverzeichnisse ermitteln	345
7.2.5	Alle Laufwerke ermitteln	345
7.2.6	Alle im Verzeichnis enthaltene Dateien ermitteln	346
7.2.7	Dateien kopieren und verschieben	347

7.2.8	Dateien umbenennen	347
7.2.9	Dateiattribute feststellen	348
7.2.10	Die FileAttribute-Enumeration	348
7.3	Mehr zur FileInfo-Klasse	349
7.3.1	Weitere wichtige Eigenschaften	349
7.3.2	GetFileSystemInfos-Methode	350
7.4	Zugriffsberechtigungen	350
7.4.1	ACL und ACE	350
7.4.2	SetAccessControl-Methode	351
7.4.3	Zugriffsrechte anzeigen	351
7.5	Weitere wichtige Klassen	352
7.5.1	Die Path-Klasse	352
7.5.2	Die Klasse FileSystemWatcher	353
7.6	Dateidialoge	354
7.6.1	Anzeige und Auswertung	354
7.6.2	Wichtige Eigenschaften	355
7.6.3	Dateifilter	356
7.7	Praxisbeispiele	356
7.7.1	Infos über Verzeichnisse und Dateien gewinnen	356
7.7.2	Die Verzeichnisstruktur in eine TreeView einlesen	360
8	Dateien lesen und schreiben	363
8.1	Grundprinzip der Datenpersistenz	363
8.1.1	Dateien und Streams	363
8.1.2	Die wichtigsten Klassen	364
8.1.3	Erzeugen eines Streams	365
8.2	Dateiparameter	365
8.2.1	FileAccess	365
8.2.2	FileMode	365
8.2.3	FileShare	366
8.3	Textdateien	366
8.3.1	Eine Textdatei beschreiben bzw. neu anlegen	366
8.3.2	Eine Textdatei lesen	368
8.4	Binärdateien	369
8.4.1	Lese-/Schreibzugriff	369
8.4.2	Die Methoden ReadAllBytes und WriteAllBytes	370
8.4.3	Varianten zum Erzeugen von BinaryReader/BinaryWriter	370

8.5	Sequenzielle Dateien	371
8.5.1	Lesen und schreiben von strukturierten Daten	371
8.5.2	Serialisieren von Objekten	372
8.6	Dateien verschlüsseln und komprimieren	373
8.6.1	Das Methodenpärchen Encrypt-/Decrypt	373
8.6.2	Verschlüsseln unter Windows XP/Vista/Windows 7	373
8.6.3	Verschlüsseln mit der CryptoStream-Klasse	374
8.6.4	Dateien komprimieren	375
8.7	Memory Mapped Files	376
8.7.1	Grundprinzip	376
8.7.2	Erzeugen eines MMF	377
8.7.3	Erstellen eines Map View	378
8.8	Praxisbeispiele	379
8.8.1	Auf eine Textdatei zugreifen	379
8.8.2	Persistente Daten im Objektbaum speichern	383
8.8.3	Ein Memory Mapped File (MMF) verwenden	389
9	XML in Theorie und Praxis	393
9.1	XML – etwas Theorie	393
9.1.1	Übersicht	393
9.1.2	Der XML-Grundaufbau	396
9.1.3	Wohlgeformte Dokumente	397
9.1.4	Processing Instructions (PI)	400
9.1.5	Elemente und Attribute	400
9.1.6	Verwendbare Zeichensätze	402
9.2	XSD-Schemas	404
9.2.1	XSD-Schemas und ADO.NET	404
9.2.2	XML-Schemas in Visual Studio analysieren	406
9.2.3	XML-Datei mit XSD-Schema erzeugen	409
9.2.4	XSD-Schema aus einer XML-Datei erzeugen	410
9.3	Verwendung des DOM unter .NET	411
9.3.1	Übersicht	411
9.3.2	DOM-Integration in C#	412
9.3.3	Laden von Dokumenten	412
9.3.4	Erzeugen von XML-Dokumenten	413
9.3.5	Auslesen von XML-Dateien	415
9.3.6	Direktzugriff auf einzelne Elemente	417

9.3.7	Einfügen von Informationen	417
9.3.8	Suchen in den Baumzweigen	420
9.4	XML-Verarbeitung mit LINQ to XML	423
9.4.1	Die LINQ to XML-API	423
9.4.2	Neue XML-Dokumente erzeugen	425
9.4.3	Laden und Sichern von XML-Dokumenten	427
9.4.4	Navigieren in XML-Daten	428
9.4.5	Auswählen und Filtern	430
9.4.6	Manipulieren der XML-Daten	430
9.4.7	XML-Dokumente transformieren	432
9.5	Weitere Möglichkeiten der XML-Verarbeitung	435
9.5.1	Die relationale Sicht auf XML-Daten mit XmlDataDocument	435
9.5.2	XML-Daten aus Objektstrukturen erzeugen	438
9.5.3	Schnelles Suchen in XML-Daten mit XPathNavigator	441
9.5.4	Schnelles Auslesen von XML-Daten mit XmlReader	444
9.5.5	Erzeugen von XML-Daten mit XmlWriter	446
9.5.6	XML transformieren mit XSLT	448
9.6	Praxisbeispiele	450
9.6.1	Mit dem DOM in XML-Dokumenten navigieren	450
9.6.2	XML-Daten in eine TreeView einlesen	453
10	Einführung in ADO.NET	457
10.1	Eine kleine Übersicht	457
10.1.1	Die ADO.NET-Klassenhierarchie	457
10.1.2	Die Klassen der Datenprovider	458
10.1.3	Das Zusammenspiel der ADO.NET-Klassen	461
10.2	Das Connection-Objekt	462
10.2.1	Allgemeiner Aufbau	462
10.2.2	OleDbConnection	462
10.2.3	Schließen einer Verbindung	464
10.2.4	Eigenschaften des Connection-Objekts	464
10.2.5	Methoden des Connection-Objekts	466
10.2.6	Der ConnectionStringBuilder	467
10.3	Das Command-Objekt	467
10.3.1	Erzeugen und Anwenden eines Command-Objekts	468
10.3.2	Erzeugen mittels CreateCommand-Methode	468
10.3.3	Eigenschaften des Command-Objekts	469
10.3.4	Methoden des Command-Objekts	471

10.3.5	Freigabe von Connection- und Command-Objekten	472
10.4	Parameter-Objekte	473
10.4.1	Erzeugen und Anwenden eines Parameter-Objekts	473
10.4.2	Eigenschaften des Parameter-Objekts	474
10.5	Das CommandBuilder-Objekt	475
10.5.1	Erzeugen	475
10.5.2	Anwenden	475
10.6	Das DataReader-Objekt	476
10.6.1	DataReader erzeugen	476
10.6.2	Daten lesen	477
10.6.3	Eigenschaften des DataReaders	478
10.6.4	Methoden des DataReaders	478
10.7	Das DataAdapter-Objekt	479
10.7.1	DataAdapter erzeugen	479
10.7.2	Command-Eigenschaften	480
10.7.3	Fill-Methode	481
10.7.4	Update-Methode	482
10.8	Praxisbeispiele	483
10.8.1	Wichtige ADO.NET-Objekte im Einsatz	483
10.8.2	Eine Aktionsabfrage ausführen	484
10.8.3	Eine Auswahlabfrage aufrufen	487
10.8.4	Die Datenbank aktualisieren	489
11	Das DataSet	493
11.1	Grundlegende Features des DataSets	493
11.1.1	Die Objekthierarchie	494
11.1.2	Die wichtigsten Klassen	494
11.1.3	Erzeugen eines DataSets	495
11.2	Das DataTable-Objekt	496
11.2.1	DataTable erzeugen	497
11.2.2	Spalten hinzufügen	497
11.2.3	Zeilen zur DataTable hinzufügen	498
11.2.4	Auf den Inhalt einer DataTable zugreifen	499
11.3	Die DataView	501
11.3.1	Erzeugen eines DataView	501
11.3.2	Sortieren und Filtern von Datensätzen	501
11.3.3	Suchen von Datensätzen	502

11.4	Typisierte DataSets	502
11.4.1	Ein typisiertes DataSet erzeugen	503
11.4.2	Das Konzept der Datenquellen	504
11.4.3	Typisierte DataSets und TableAdapter	504
11.5	Die Qual der Wahl	506
11.5.1	DataReader – der schnelle Lesezugriff	506
11.5.2	DataSet – die Datenbank im Hauptspeicher	507
11.5.3	Objektrelationales Mapping – die Zukunft?	508
11.6	Praxisbeispiele	509
11.6.1	Im DataView sortieren und filtern	509
11.6.2	Suche nach Datensätzen	511
11.6.3	Ein DataSet in einen XML-String serialisieren	512
11.6.4	Ein untypisiertes in ein typisiertes DataSet konvertieren	517
11.6.5	Eine LINQ to SQL-Abfrage ausführen	522
12	Asynchrone Programmierung	527
12.1	Übersicht	527
12.1.1	Multitasking versus Multithreading	528
12.1.2	Deadlocks	529
12.1.3	Racing	529
12.2	Programmieren mit Threads	531
12.2.1	Einführungsbeispiel	531
12.2.2	Wichtige Thread-Methoden	532
12.2.3	Wichtige Thread-Eigenschaften	534
12.2.4	Einsatz der ThreadPool-Klasse	535
12.3	Sperrmechanismen	537
12.3.1	Threading ohne lock	537
12.3.2	Threading mit lock	539
12.3.3	Die Monitor-Klasse	541
12.3.4	Mutex	544
12.3.5	Methoden für die parallele Ausführung sperren	546
12.3.6	Semaphore	546
12.4	Interaktion mit der Programmoberfläche	548
12.4.1	Die Werkzeuge	548
12.4.2	Einzelne Steuerelemente mit Invoke aktualisieren	548
12.4.3	Mehrere Steuerelemente aktualisieren	550
12.4.4	Ist ein Invoke-Aufruf nötig?	550
12.4.5	Und was ist mit WPF?	551

12.5	Timer-Threads	552
12.6	Verwendung der BackgroundWorker-Komponente	553
12.7	Asynchrone Programmier-Entwurfsmuster	556
12.7.1	Kurzübersicht	556
12.7.2	Polling	557
12.7.3	Callback verwenden	559
12.7.4	Callback mit Parameterübergabe verwenden	560
12.7.5	Callback mit Zugriff auf die Programm-Oberfläche	561
12.8	Asynchroner Aufruf beliebiger Methoden	562
12.8.1	Die Beispielklasse	562
12.8.2	Asynchroner Aufruf ohne Callback	564
12.8.3	Asynchroner Aufruf mit Callback und Anzeigefunktion	565
12.8.4	Aufruf mit Rückgabewerten (per Eigenschaft)	566
12.8.5	Aufruf mit Rückgabewerten (per EndInvoke)	566
12.9	Praxisbeispiele	567
12.9.1	Spieltrieb & Multithreading erleben	567
12.9.2	Prozess- und Thread-Informationen gewinnen	581
12.9.3	Ein externes Programm starten	586
13	Die Task Parallel Library	589
13.1	Überblick	589
13.1.1	Parallel-Programmierung	589
13.1.2	Möglichkeiten der TPL	591
13.1.3	Der CLR-Threadpool	592
13.2	Parallele Verarbeitung mit Parallel.Invoke	592
13.2.1	Aufrufvarianten	593
13.2.2	Einschränkungen	594
13.3	Verwendung von Parallel.For	595
13.3.1	Abbrechen der Verarbeitung	596
13.3.2	Auswerten des Bearbeitungsstatus	597
13.3.3	Und was ist mit anderen Schrittweiten für den Iterator?	598
13.4	Verarbeiten von Collections mit Parallel.ForEach	599
13.5	Die neue Task-Klasse	599
13.5.1	Einen Task erzeugen	599
13.5.2	Task starten	600
13.5.3	Datenübergabe an den Task	602
13.5.4	Wie warte ich auf das Taskende?	603
13.5.5	Tasks mit Rückgabewerten	605

13.5.6	Die Verarbeitung abbrechen	608
13.5.7	Fehlerbehandlung	612
13.5.8	Weitere Eigenschaften	613
13.6	Zugriff auf das Userinterface	614
13.6.1	Reaktion auf das Task-Ende und Zugriff auf die Oberfläche	614
13.6.2	Zugriff auf das UI aus dem Task heraus	616
13.7	Weitere Datenstrukturen im Überblick	618
13.7.1	Threadsichere Collections	618
13.7.2	Primitive für die Threadsynchrisation	619
13.8	Parallel LINQ (PLINQ)	619
13.9	Die Parallel Diagnostic Tools	619
13.9.1	Fenster für parallele Aufgaben	620
13.9.2	Fenster für parallele Stacks	620
13.9.3	IntelliTrace	621
13.10	Praxisbeispiel: Spieltrieb – die zweite Version	622
13.10.1	Aufgabenstellung	622
13.10.2	Global-Klasse	623
13.10.3	Controller-Klasse	624
13.10.4	LKW-Klasse	625
13.10.5	Schiff-Klasse	627
13.10.6	Oberfläche	629
14	Fehlersuche und Behandlung	631
14.1	Der Debugger	631
14.1.1	Allgemeine Beschreibung	631
14.1.2	Einzel-schritt-Modus	635
14.1.3	Prozedur-schritt-Modus	636
14.1.4	Breakpoints	636
14.1.5	Debugging am konkreten Beispiel	636
14.1.6	Das Debug-Objekt	640
14.2	Fehlerbehandlung	641
14.2.1	Anweisungen zur Fehlerbehandlung	641
14.2.2	try-catch	642
14.2.3	try-finally	647
14.2.4	Das Standardverhalten bei Ausnahmen festlegen	649
14.2.5	Die Exception-Klasse	650
14.2.6	Fehler/Ausnahmen auslösen	651
14.2.7	Eigene Fehlerklassen	651

14.2.8	Exceptionhandling zur Entwurfszeit	653
14.2.9	Code Contracts	654
15	Unit-Tests	655
15.1	Testgetriebene Entwicklung (TDD)	655
15.1.1	Konventionelle Vorgehensweise	655
15.1.2	Testgetriebene Entwicklung mit Unit-Tests	656
15.1.3	Unit-Tests unter Visual Studio	656
15.2	Einfache Tests	657
15.2.1	Test einer Klasse am Beispiel	657
15.2.2	Eigene Testmethoden hinzufügen	663
15.3	Datengetriebene Tests (DDT)	664
15.3.1	Das DDT-Prinzip	664
15.3.2	DDT am Beispiel	664
15.4	Begriffe und Ergänzungen	668
15.4.1	Behauptungen (Asserts)	669
15.4.2	Der Testkontext	669
15.4.3	Zusätzliche Testattribute	670
15.5	Einsatzkriterien von Unit Tests	671
15.5.1	Nutzen	671
15.5.2	Beschränkungen	672
16	OOP-Spezial	673
16.1	Eine kleine Einführung in die UML	673
16.1.1	Use Case-Diagramm	673
16.1.2	Use Case-Dokumentation	675
16.1.3	Objekte identifizieren	676
16.1.4	Statisches Modell	677
16.1.5	Beziehungen zwischen den Klassen	678
16.1.6	Dynamisches Modell	678
16.1.7	Implementierung	679
16.1.8	Test-Client	683
16.2	Der Klassen-Designer	686
16.2.1	Ein neues Klassendiagramm erzeugen	687
16.2.2	Toolbox	688
16.2.3	Enumeration	689
16.2.4	Klasse	691
16.2.5	Struktur	693

16.2.6	Abstrakte Klasse	693
16.2.7	Schnittstelle	695
16.2.8	Delegate	697
16.2.9	Zuordnung	699
16.2.10	Vererbung	700
16.2.11	Diagramme anpassen	700
16.2.12	Objekt-Testcenter	701
16.2.13	Wann lohnt sich der Einsatz des Klassen-Designers?	703
16.3	Praxisbeispiele	703
16.3.1	Implementierung einer Finite State Machine	703
16.3.2	Modellierung des Bestellsystems einer Firma	709
17	Das Microsoft Event Pattern	723
17.1	Was sind Design Pattern und wozu braucht man sie?	723
17.2	Aufbau und Bedeutung des Observer Pattern	724
17.2.1	Subjekt und Observer	724
17.2.2	Sequenzdiagramme	726
17.2.3	Die Registration-Sequenz	726
17.2.4	Die Notification-Sequenz	727
17.2.5	Die Unregistration-Sequenz	727
17.2.6	Was bedeuten diese Sequenzen für unser Geschäftsmodell?	728
17.2.7	Die Rolle des Containers	728
17.3	Implementierung mit Interfaces und Callbacks	729
17.3.1	Übersicht und Klassendiagramm	729
17.3.2	Schnittstellen IObserver und IObservable	731
17.3.3	Abstrakte Klasse Subject	731
17.3.4	Observer1	732
17.3.5	Observer2	733
17.3.6	Model	734
17.3.7	Form1	735
17.3.8	Ein zweites Klassendiagramm	736
17.3.9	Testen	737
17.4	Implementierung mit Delegates und Events	738
17.4.1	Multicast-Events	739
17.4.2	IObserver, IObservable und Subject	739
17.4.3	Observer1 und Observer2	740
17.4.4	Model	740
17.4.5	Form1	741

17.4.6	Test und Vergleich	741
17.4.7	Klassendiagramm	742
17.5	Implementierung des Microsoft Event Pattern	743
17.5.1	Namensgebung für Ereignisse	743
17.5.2	Namensgebung und Signatur der Delegates	743
17.5.3	Hinzufügen einer das Ereignis auslösenden Methode	744
17.5.4	Model	745
17.5.5	Observer1	746
17.5.6	Form1	747
17.6	Test und Vergleich	747
17.7	Klassendiagramm	747
17.8	Schritte zur Implementierung eines Event Pattern	748
17.9	Praxisbeispiel	750
17.9.1	Subjekt und Observer beobachten sich gegenseitig	750
18	Weitere Programmiertechniken	761
18.1	Zugriff auf die Zwischenablage	761
18.1.1	Das Clipboard-Objekt	761
18.1.2	Zwischenablage-Funktionen für Textboxen	763
18.2	Arbeiten mit der Registry	763
18.2.1	Allgemeines	764
18.2.2	Registry-Unterstützung in .NET	766
18.3	.NET-Reflection	767
18.3.1	Übersicht	767
18.3.2	Assembly laden	767
18.3.3	Mittels GetType und Type Informationen sammeln	768
18.3.4	Dynamisches Laden von Assemblies	770
18.4	Das SerialPort-Control	773
18.4.1	Übersicht	773
18.4.2	Einführungsbeispiele	774
18.4.3	Thread-Probleme bei Windows Forms Anwendungen	777
18.4.4	Ein einfaches Terminalprogramm	780
18.5	Praxisbeispiele	785
18.5.1	Zugriff auf die Registry	785
18.5.2	Dateiverknüpfungen erzeugen	787
18.5.3	Betrachter für Manifestressourcen	789
18.5.4	Ressourcen mit Reflection auslesen	792

19 Konsolenanwendungen	795
19.1 Grundaufbau/Konzepte	795
19.1.1 Unser Hauptprogramm – Program.cs	796
19.1.2 Rückgabe eines Fehlerstatus	797
19.1.3 Parameterübergabe	798
19.1.4 Zugriff auf die Umgebungsvariablen	799
19.2 Die Kommandozentrale: System.Console	800
19.2.1 Eigenschaften	801
19.2.2 Methoden/Ereignisse	801
19.2.3 Textausgaben	802
19.2.4 Farbangaben	803
19.2.5 Tastaturabfragen	804
19.2.6 Arbeiten mit Streamdaten	805
19.3 Praxisbeispiele	807
19.3.1 Farbige Konsolenanwendung	807
19.3.2 Weitere Hinweise und Beispiele	809
20 Verteilen von Anwendungen	811
20.1 ClickOnce-Deployment	811
20.1.1 Übersicht/Einschränkungen	811
20.1.2 Die Vorgehensweise	812
20.1.3 Ort der Veröffentlichung	813
20.1.4 Anwendungsdateien	813
20.1.5 Erforderliche Komponenten	814
20.1.6 Aktualisierungen	815
20.1.7 Veröffentlichen	815
20.1.8 Verzeichnisstruktur	816
20.1.9 Der Webpublishing-Assistent	818
20.1.10 Neue Versionen erstellen	818
20.2 Setup-Projekte	819
20.2.1 Ein neues Setup-Projekt	819
20.2.2 Dateisystem-Editor	821
20.2.3 Ein erster Test	823
20.2.4 Registrierungs-Editor	823
20.2.5 Dateityp-Editor	824
20.2.6 Benutzeroberflächen-Editor	825
20.2.7 Editor für Startbedingungen	828
20.2.8 Finaler Test	829

Teil III: Windows Forms

21 Windows Forms-Anwendungen	833
21.1 Grundaufbau/Konzepte	833
21.1.1 Das Hauptprogramm – Program.cs	834
21.1.2 Die Oberflächendefinition – Form1.Designer.cs	838
21.1.3 Die Spielwiese des Programmierers – Form1.cs	839
21.1.4 Die Datei AssemblyInfo.cs	840
21.1.5 Resources.resx/Resources.Designer.cs	841
21.1.6 Settings.settings/Settings.Designer.cs	842
21.1.7 Settings.cs	843
21.2 Ein Blick auf die Application-Klasse	844
21.2.1 Eigenschaften	844
21.2.2 Methoden	845
21.2.3 Ereignisse	847
21.3 Allgemeine Eigenschaften von Komponenten	847
21.3.1 Font	848
21.3.2 Handle	850
21.3.3 Tag	851
21.3.4 Modifiers	851
21.4 Allgemeine Ereignisse von Komponenten	852
21.4.1 Die Eventhandler-Argumente	852
21.4.2 Sender	852
21.4.3 Der Parameter e	854
21.4.4 Mausereignisse	854
21.4.5 KeyPreview	856
21.4.6 Weitere Ereignisse	857
21.4.7 Validitätsprüfungen	857
21.4.8 SendKeys	858
21.5 Allgemeine Methoden von Komponenten	859
22 Windows Forms-Formulare	861
22.1 Übersicht	861
22.1.1 Wichtige Eigenschaften des Form-Objekts	862
22.1.2 Wichtige Ereignisse des Form-Objekts	864
22.1.3 Wichtige Methoden des Form-Objekts	865
22.2 Praktische Aufgabenstellungen	866
22.2.1 Fenster anzeigen	866

22.2.2	Einen Splash Screen beim Anwendungsstart anzeigen	869
22.2.3	Eine Sicherheitsabfrage vor dem Schließen anzeigen	872
22.2.4	Ein Formular durchsichtig machen	872
22.2.5	Die Tabulatorreihenfolge festlegen	873
22.2.6	Ausrichten und Platzieren von Komponenten im Formular	874
22.2.7	Spezielle Panels für flexibles Layout	876
22.2.8	Menüs erzeugen	878
22.3	MDI-Anwendungen	882
22.3.1	Die "falschen" MDI-Fenster bzw. Verwenden von Parent	882
22.3.2	Die echten MDI-Fenster	883
22.3.3	Die Kindfenster	884
22.3.4	Automatisches Anordnen der Kindfenster	885
22.3.5	Zugriff auf die geöffneten MDI-Kindfenster	886
22.3.6	Zugriff auf das aktive MDI-Kindfenster	887
22.3.7	Kombinieren der Kindfenstermenüs mit dem MDIContainer	887
22.4	Praxisbeispiele	888
22.4.1	Informationsaustausch zwischen Formularen	888
22.4.2	Ereigniskette beim Laden/Entladen eines Formulars	895
23	Windows Forms-Komponenten	901
23.1	Allgemeine Hinweise	901
23.1.1	Hinzufügen von Komponenten	901
23.1.2	Komponenten zur Laufzeit per Code erzeugen	902
23.2	Allgemeine Steuerelemente	904
23.2.1	Label	904
23.2.2	LinkLabel	905
23.2.3	Button	907
23.2.4	TextBox	907
23.2.5	MaskedTextBox	910
23.2.6	CheckBox	911
23.2.7	RadioButton	913
23.2.8	ListBox	913
23.2.9	CheckedListBox	915
23.2.10	ComboBox	915
23.2.11	PictureBox	916
23.2.12	DateTimePicker	916
23.2.13	MonthCalendar	917
23.2.14	HScrollBar, VScrollBar	918

23.2.15	TrackBar	919
23.2.16	NumericUpDown	919
23.2.17	DomainUpDown	920
23.2.18	ProgressBar	920
23.2.19	RichTextBox	921
23.2.20	ListView	922
23.2.21	TreeView	928
23.2.22	WebBrowser	933
23.3	Container	934
23.3.1	FlowLayout/TableLayout/SplitContainer	934
23.3.2	Panel	934
23.3.3	GroupBox	935
23.3.4	TabControl	935
23.3.5	ImageList	938
23.4	Menüs & Symbolleisten	939
23.4.1	MenuStrip und ContextMenuStrip	939
23.4.2	ToolStrip	939
23.4.3	StatusStrip	940
23.4.4	ToolStripContainer	940
23.5	Daten	940
23.5.1	DataSet	940
23.5.2	DataGridView/DataGrid	941
23.5.3	BindingNavigator/BindingSource	941
23.5.4	Chart	941
23.6	Komponenten	943
23.6.1	ErrorProvider	943
23.6.2	HelpProvider	943
23.6.3	ToolTip	943
23.6.4	BackgroundWorker	943
23.6.5	Timer	943
23.6.6	SerialPort	944
23.7	Drucken	944
23.7.1	PrintPreviewControl	944
23.7.2	PrintDocument	944
23.8	Dialoge	944
23.8.1	OpenFileDialog/SaveFileDialog/FolderBrowserDialog	944
23.8.2	FontDialog/ColorDialog	945
23.9	WPF-Unterstützung mit dem ElementHost	945

23.10	Praxisbeispiele	945
23.10.1	Mit der CheckBox arbeiten	945
23.10.2	Steuerelemente per Code selbst erzeugen	947
23.10.3	Controls-Auflistung des Formulars im TreeView anzeigen	949
23.10.4	WPF-Komponenten mit dem ElementHost anzeigen	953
24	Einführung Grafikausgabe	957
24.1	Übersicht und erste Schritte	957
24.1.1	GDI+ – Ein erster Blick für Umsteiger	958
24.1.2	Namespaces für die Grafikausgabe	959
24.2	Darstellen von Grafiken	961
24.2.1	Die PictureBox-Komponente	961
24.2.2	Das Image-Objekt	962
24.2.3	Laden von Grafiken zur Laufzeit	963
24.2.4	Sichern von Grafiken	963
24.2.5	Grafikeigenschaften ermitteln	964
24.2.6	Erzeugen von Vorschaugrafiken (Thumbnails)	965
24.2.7	Die Methode RotateFlip	966
24.2.8	Skalieren von Grafiken	967
24.3	Das .NET-Koordinatensystem	968
24.3.1	Globale Koordinaten	969
24.3.2	Seitenkoordinaten (globale Transformation)	970
24.3.3	Gerätekoordinaten (Seitentransformation)	972
24.4	Grundlegende Zeichenfunktionen von GDI+	973
24.4.1	Das zentrale Graphics-Objekt	973
24.4.2	Punkte zeichnen/abfragen	976
24.4.3	Linien	977
24.4.4	Kantenglättung mit Antialiasing	978
24.4.5	PolyLine	979
24.4.6	Rechtecke	979
24.4.7	Polygone	981
24.4.8	Splines	982
24.4.9	Bézierkurven	983
24.4.10	Kreise und Ellipsen	984
24.4.11	Tortenstück (Segment)	985
24.4.12	Bogenstück	986
24.4.13	Wo sind die Rechtecke mit den runden Ecken?	987
24.4.14	Textausgabe	988

24.4.15	Ausgabe von Grafiken	992
24.5	Unser Werkzeugkasten	993
24.5.1	Einfache Objekte	993
24.5.2	Vordefinierte Objekte	995
24.5.3	Farben/Transparenz	997
24.5.4	Stifte (Pen)	998
24.5.5	Pinsel (Brush)	1001
24.5.6	SolidBrush	1002
24.5.7	HatchBrush	1002
24.5.8	TextureBrush	1003
24.5.9	LinearGradientBrush	1004
24.5.10	PathGradientBrush	1006
24.5.11	Fonts	1007
24.5.12	Path-Objekt	1008
24.5.13	Clipping/Region	1011
24.6	Standarddialoge	1014
24.6.1	Schriftauswahl	1014
24.6.2	Farbauswahl	1015
24.7	Praxisbeispiele	1017
24.7.1	Ein Graphics-Objekt erzeugen	1017
24.7.2	Zeichenoperationen mit der Maus realisieren	1020
25	Druckausgabe	1025
25.1	Einstieg und Übersicht	1025
25.1.1	Nichts geht über ein Beispiel	1025
25.1.2	Programmiermodell	1027
25.1.3	Kurzübersicht der Objekte	1028
25.2	Auswerten der aktuellen Druckereinstellungen	1028
25.2.1	Die vorhandenen Drucker	1028
25.2.2	Der Standarddrucker	1029
25.2.3	Verfügbare Papierformate/Seitenabmessungen	1030
25.2.4	Der eigentliche Druckbereich	1031
25.2.5	Die Seitenausrichtung ermitteln	1032
25.2.6	Ermitteln der Farbfähigkeit	1032
25.2.7	Die Druckauflösung abfragen	1032
25.2.8	Ist beidseitiger Druck möglich?	1033
25.2.9	Einen "Informationsgerätekontext" erzeugen	1033
25.2.10	Abfragen von Werten während des Drucks	1034

25.3	Festlegen von Druckereinstellungen	1035
25.3.1	Einen Drucker auswählen	1035
25.3.2	Drucken in Millimetern	1035
25.3.3	Festlegen der Seitenränder	1036
25.3.4	Druckjobname	1037
25.3.5	Anzahl der Kopien	1038
25.3.6	Beidseitiger Druck	1038
25.3.7	Seitenzahlen festlegen	1039
25.3.8	Druckqualität verändern	1042
25.3.9	Ausgabemöglichkeiten des Chart-Controls nutzen	1043
25.4	Die Druckdialoge verwenden	1043
25.4.1	PrintDialog	1044
25.4.2	PageSetupDialog	1045
25.4.3	PrintPreviewDialog	1047
25.4.4	Ein eigenes Druckvorschau-Fenster realisieren	1048
25.5	Drucken mit OLE-Automation	1049
25.5.1	Kurzeinstieg in die OLE-Automation	1049
25.5.2	Drucken mit Microsoft Word	1051
25.6	Praxisbeispiele	1053
25.6.1	Den Drucker umfassend konfigurieren	1053
25.6.2	Diagramme mit dem Chart-Control drucken	1063
25.6.3	Druckausgabe mit Word	1065
26	Windows Forms-Datenbindung	1071
26.1	Prinzipielle Möglichkeiten	1071
26.2	Manuelle Datenbindung an einfache Datenfelder	1072
26.2.1	BindingSource erzeugen	1072
26.2.2	Binding-Objekt	1073
26.2.3	DataBindings-Collection	1073
26.3	Manuelle Datenbindung an Listen und Tabelleninhalte	1073
26.3.1	DataGridView	1074
26.3.2	Datenbindung von ComboBox und ListBox	1074
26.4	Entwurfszeit-Datenbindung an typisierte DataSets	1074
26.5	Drag & Drop-Datenbindung	1076
26.6	Navigations- und Bearbeitungsfunktionen realisieren	1076
26.6.1	Navigieren zwischen den Datensätzen	1076
26.6.2	Hinzufügen und Löschen	1076

26.6.3	Aktualisieren und Abbrechen	1077
26.6.4	Verwendung des BindingNavigators	1077
26.7	Die Anzeigedaten formatieren	1078
26.8	Praxisbeispiele	1078
26.8.1	Einrichten und Verwenden einer Datenquelle	1078
26.8.2	Eine Auswahlabfrage im DataGridView anzeigen	1082
26.8.3	Master-Detailbeziehungen im DataGridView anzeigen	1085
26.8.4	Datenbindung Chart-Control	1086
27	Erweiterte Grafikausgabe	1091
27.1	Transformieren mit der Matrix-Klasse	1091
27.1.1	Übersicht	1091
27.1.2	Translation	1092
27.1.3	Skalierung	1092
27.1.4	Rotation	1093
27.1.5	Scherung	1093
27.1.6	Zuweisen der Matrix	1094
27.2	Low-Level-Grafikmanipulationen	1094
27.2.1	Worauf zeigt Scan0?	1095
27.2.2	Anzahl der Spalten bestimmen	1096
27.2.3	Anzahl der Zeilen bestimmen	1097
27.2.4	Zugriff im Detail (erster Versuch)	1097
27.2.5	Zugriff im Detail (zweiter Versuch)	1099
27.2.6	Invertieren	1101
27.2.7	In Graustufen umwandeln	1102
27.2.8	Heller/Dunkler	1103
27.2.9	Kontrast	1105
27.2.10	Gamma-Wert	1106
27.2.11	Histogramm spreizen	1106
27.2.12	Ein universeller Grafikfilter	1109
27.3	Fortgeschrittene Techniken	1113
27.3.1	Flackerfrei dank Double Buffering	1113
27.3.2	Animationen	1115
27.3.3	Animated GIFs	1118
27.3.4	Auf einzelne GIF-Frames zugreifen	1120
27.3.5	Transparenz realisieren	1122
27.3.6	Eine Grafik maskieren	1123
27.3.7	JPEG-Qualität beim Sichern bestimmen	1125

27.4	Grundlagen der 3D-Vektorgrafik	1126
27.4.1	Datentypen für die Verwaltung	1126
27.4.2	Eine universelle 3D-Grafik-Klasse	1127
27.4.3	Grundlegende Betrachtungen	1128
27.4.4	Translation	1131
27.4.5	Streckung/Skalierung	1132
27.4.6	Rotation	1133
27.4.7	Die eigentlichen Zeichenroutinen	1135
27.5	Und doch wieder GDI-Funktionen	1137
27.5.1	Am Anfang war das Handle	1137
27.5.2	Gerätekontext (Device Context Types)	1140
27.5.3	Koordinatensysteme und Abbildungsmodi	1142
27.5.4	Zeichenwerkzeuge/Objekte	1146
27.5.5	Bitmaps	1148
27.6	Praxisbeispiele	1152
27.6.1	Die Transformationsmatrix verstehen	1152
27.6.2	Eine 3D-Grafikausgabe in Aktion	1155
27.6.3	Einen Fenster-Screenshot erzeugen	1158
28	Ressourcen/Lokalisierung	1161
28.1	Manifestressourcen	1161
28.1.1	Erstellen von Manifestressourcen	1161
28.1.2	Zugriff auf Manifestressourcen	1162
28.2	Typisierte Ressourcen	1164
28.2.1	Erzeugen von .resources-Dateien	1164
28.2.2	Hinzufügen der .resources-Datei zum Projekt	1165
28.2.3	Zugriff auf die Inhalte von .resources-Dateien	1165
28.2.4	ResourceManager direkt aus der .resources-Datei erzeugen	1166
28.2.5	Was sind .resx-Dateien?	1167
28.3	Streng typisierte Ressourcen	1167
28.3.1	Erzeugen streng typisierter Ressourcen	1167
28.3.2	Verwenden streng typisierter Ressourcen	1168
28.3.3	Streng typisierte Ressourcen per Reflection auslesen	1168
28.4	Anwendungen lokalisieren	1171
28.4.1	Localizable und Language	1171
28.4.2	Beispiel "Landesfahnen"	1171
28.4.3	Einstellen der aktuellen Kultur zur Laufzeit	1174

29	Komponentenentwicklung	1177
29.1	Überblick	1177
29.2	Benutzersteuerelement	1178
29.2.1	Entwickeln einer Auswahl-ListBox	1178
29.2.2	Komponente verwenden	1180
29.3	Benutzerdefiniertes Steuerelement	1181
29.3.1	Entwickeln eines BlinkLabels	1181
29.3.2	Verwenden der Komponente	1184
29.4	Komponentenklasse	1184
29.5	Eigenschaften	1185
29.5.1	Einfache Eigenschaften	1185
29.5.2	Schreib-/Lesezugriff (Get/Set)	1185
29.5.3	Nur Lese-Eigenschaft (ReadOnly)	1186
29.5.4	Nur-Schreibzugriff (WriteOnly)	1187
29.5.5	Hinzufügen von Beschreibungen	1187
29.5.6	Ausblenden im Eigenschaftenfenster	1187
29.5.7	Einfügen in Kategorien	1188
29.5.8	Default-Wert einstellen	1188
29.5.9	Standard-Eigenschaft (Indexer)	1189
29.5.10	Wertebereichsbeschränkung und Fehlerprüfung	1189
29.5.11	Eigenschaften von Aufzählungstypen	1191
29.5.12	Standard Objekt-Eigenschaften	1192
29.5.13	Eigene Objekt-Eigenschaften	1192
29.6	Methoden	1194
29.6.1	Konstruktor	1195
29.6.2	Class-Konstruktor	1196
29.6.3	Destruktor	1197
29.6.4	Aufruf des Basisklassen-Konstruktors	1197
29.6.5	Aufruf von Basisklassen-Methoden	1198
29.7	Ereignisse (Events)	1198
29.7.1	Ereignis mit Standardargument definieren	1199
29.7.2	Ereignis mit eigenen Argumenten	1200
29.7.3	Ein Default-Ereignis festlegen	1201
29.7.4	Mit Ereignissen auf Windows-Messages reagieren	1201
29.8	Weitere Themen	1203
29.8.1	Wohin mit der Komponente?	1203
29.8.2	Assembly-Informationen festlegen	1204
29.8.3	Assemblies signieren	1207

29.8.4	Komponenten-Ressourcen einbetten	1207
29.8.5	Der Komponente ein Icon zuordnen	1208
29.8.6	Den Designmodus erkennen	1209
29.8.7	Komponenten lizenzieren	1209
29.9	Praxisbeispiele	1213
29.9.1	AnimGif – Komponente für die Anzeige von Animationen	1213
29.9.2	Eine FontComboBox entwickeln	1216
29.9.3	Das PropertyGrid verwenden	1218

Teil IV: WPF-Anwendungen

30	Einführung in WPF	1223
30.1	Einführung	1224
30.1.1	Was kann eine WPF-Anwendung?	1224
30.1.2	Die eXtensible Application Markup Language	1226
30.1.3	Verbinden von XAML und C#-Code	1230
30.1.4	Zielpattformen	1236
30.1.5	Applikationstypen	1237
30.1.6	Vor- und Nachteile von WPF-Anwendungen	1238
30.1.7	Weitere Dateien im Überblick	1238
30.2	Alles beginnt mit dem Layout	1241
30.2.1	Allgemeines zum Layout	1241
30.2.2	Positionieren von Steuerelementen	1243
30.2.3	Canvas	1247
30.2.4	StackPanel	1247
30.2.5	DockPanel	1249
30.2.6	WrapPanel	1251
30.2.7	UniformGrid	1251
30.2.8	Grid	1253
30.2.9	ViewBox	1258
30.2.10	TextBlock	1259
30.3	Das WPF-Programm	1262
30.3.1	Die App-Klasse	1262
30.3.2	Das Startobjekt festlegen	1263
30.3.3	Kommandozeilenparameter verarbeiten	1264
30.3.4	Die Anwendung beenden	1265
30.3.5	Auswerten von Anwendungsereignissen	1265

30.4	Die Window-Klasse	1266
30.4.1	Position und Größe festlegen	1267
30.4.2	Rahmen und Beschriftung	1267
30.4.3	Das Fenster-Icon ändern	1267
30.4.4	Anzeige weiterer Fenster	1268
30.4.5	Transparenz	1268
30.4.6	Abstand zum Inhalt festlegen	1269
30.4.7	Fenster ohne Fokus anzeigen	1270
30.4.8	Ereignisfolge bei Fenstern	1270
30.4.9	Ein paar Worte zur Schriftdarstellung	1271
30.4.10	Ein paar Worte zur Darstellung von Controls	1273
30.4.11	Wird mein Fenster komplett mit WPF gerendert?	1275
31	Übersicht WPF-Controls	1277
31.1	Allgemeingültige Eigenschaften	1277
31.2	Label	1279
31.3	Button, RepeatButton, ToggleButton	1279
31.3.1	Schaltflächen für modale Dialoge	1280
31.3.2	Schaltflächen mit Grafik	1281
31.4	TextBox, PasswordBox	1282
31.4.1	TextBox	1282
31.4.2	PasswordBox	1284
31.5	CheckBox	1285
31.6	RadioButton	1287
31.7	ListBox, ComboBox	1288
31.7.1	ListBox	1288
31.7.2	ComboBox	1291
31.7.3	Den Content formatieren	1292
31.8	Image	1294
31.8.1	Grafik per XAML zuweisen	1294
31.8.2	Grafik zur Laufzeit zuweisen	1294
31.8.3	Bild aus Datei laden	1295
31.8.4	Die Grafikskalierung beeinflussen	1296
31.9	MediaElement	1297
31.10	Slider, ScrollBar	1300
31.10.1	Slider	1300
31.10.2	ScrollBar	1301
31.11	ScrollViewer	1302

31.12	Menu, ContextMenu	1303
31.12.1	Menu	1303
31.12.2	Tastenkürzel	1304
31.12.3	Grafiken	1305
31.12.4	Weitere Möglichkeiten	1306
31.12.5	ContextMenu	1307
31.13	ToolBar	1308
31.14	StatusBar, ProgressBar	1311
31.14.1	StatusBar	1311
31.14.2	ProgressBar	1313
31.15	Border, GroupBox, BulletDecorator	1314
31.15.1	Border	1314
31.15.2	GroupBox	1315
31.15.3	BulletDecorator	1316
31.16	RichTextBox	1318
31.16.1	Verwendung und Anzeige von vordefiniertem Text	1318
31.16.2	Neues Dokument zur Laufzeit erzeugen	1320
31.16.3	Sichern von Dokumenten	1320
31.16.4	Laden von Dokumenten	1322
31.16.5	Texte per Code einfügen/modifizieren	1323
31.16.6	Texte formatieren	1324
31.16.7	EditingCommands	1325
31.16.8	Grafiken/Objekte einfügen	1326
31.16.9	Rechtschreibkontrolle	1328
31.17	FlowDocumentPageViewer, -Reader, -ScrollViewer	1328
31.17.1	FlowDocumentPageViewer	1328
31.17.2	FlowDocumentReader	1329
31.17.3	FlowDocumentScrollViewer	1329
31.18	FlowDocument	1329
31.18.1	FlowDocument per XAML beschreiben	1330
31.18.2	FlowDocument per Code erstellen	1332
31.19	DocumentViewer	1333
31.20	Expander, TabControl	1334
31.20.1	Expander	1334
31.20.2	TabControl	1336
31.21	Popup	1337
31.22	TreeView	1339
31.23	ListView	1342

31.24	DataGrid	1343
31.25	Calendar/DatePicker	1344
31.26	InkCanvas	1348
31.26.1	Stift-Parameter definieren	1348
31.26.2	Die Zeichenmodi	1349
31.26.3	Inhalte laden und sichern	1350
31.26.4	Konvertieren in eine Bitmap	1350
31.26.5	Weitere Eigenschaften	1351
31.27	Ellipse, Rectangle, Line und Co.	1352
31.27.1	Ellipse	1352
31.27.2	Rectangle	1352
31.27.3	Line	1353
31.28	Browser	1353
31.29	Ribbon	1356
31.30	Chart	1358
31.31	WindowsFormsHost	1358
32	Wichtige WPF-Techniken	1361
32.1	Eigenschaften	1361
32.1.1	Abhängige Eigenschaften (Dependency Properties)	1361
32.1.2	Angehängte Eigenschaften (Attached Properties)	1363
32.2	Einsatz von Ressourcen	1363
32.2.1	Was sind eigentlich Ressourcen?	1363
32.2.2	Wo können Ressourcen gespeichert werden?	1363
32.2.3	Wie definiere ich eine Ressource?	1365
32.2.4	Statische und dynamische Ressourcen	1366
32.2.5	Wie werden Ressourcen adressiert?	1367
32.2.6	System-Ressourcen einbinden	1368
32.3	Das WPF-Ereignis-Modell	1368
32.3.1	Einführung	1368
32.3.2	Routed Events	1369
32.3.3	Direkte Events	1371
32.4	Verwendung von Commands	1372
32.4.1	Einführung Commands	1372
32.4.2	Verwendung vordefinierter Commands	1372
32.4.3	Das Ziel des Commands	1374
32.4.4	Welche vordefinierten Commands stehen zur Verfügung?	1375
32.4.5	Commands an Ereignismethoden binden	1376

32.4.6	Wie kann ich ein Command per Code auslösen?	1377
32.4.7	Command-Ausführung verhindern	1378
32.5	Das WPF-Style-System	1378
32.5.1	Übersicht	1378
32.5.2	Benannte Styles	1379
32.5.3	Typ-Styles	1380
32.5.4	Styles anpassen und vererben	1381
32.6	Verwenden von Triggern	1384
32.6.1	Eigenschaften-Trigger (Property Triggers)	1384
32.6.2	Ereignis-Trigger	1386
32.6.3	Daten-Trigger	1387
32.7	Einsatz von Templates	1388
32.8	Transformationen, Animationen und StoryBoards	1394
32.8.1	Transformationen	1394
32.8.2	Animationen mit dem StoryBoard realisieren	1398
33	Grundlagen der WPF-Datenbindung	1405
33.1	Grundprinzip	1405
33.1.1	Bindungsarten	1406
33.1.2	Wann wird eigentlich die Quelle aktualisiert?	1407
33.1.3	Bindung zur Laufzeit realisieren	1408
33.2	Binden an Objekte	1410
33.2.1	Objekte im Code instanziiieren	1410
33.2.2	Verwenden der Instanz im C#-Quellcode	1412
33.2.3	Anforderungen an die Quell-Klasse	1412
33.2.4	Instanziiieren von Objekten per C#-Code	1414
33.3	Binden von Collections	1415
33.3.1	Anforderung an die Collection	1415
33.3.2	Einfache Anzeige	1416
33.3.3	Navigieren zwischen den Objekten	1417
33.3.4	Einfache Anzeige in einer ListBox	1419
33.3.5	DataTemplates zur Anzeigeformatierung	1420
33.3.6	Mehr zu List- und ComboBox	1421
33.3.7	Verwendung der ListView	1423
33.4	Anzeige von Datenbankinhalten	1425
33.4.1	Datenmodell per LINQ to SQL-Designer erzeugen	1425
33.4.2	Die Programm-Oberfläche	1426
33.4.3	Der Zugriff auf die Daten	1427

33.5	Noch einmal zurück zu den Details	1428
33.5.1	Navigieren in den Daten	1429
33.5.2	Sortieren	1430
33.5.3	Filtern	1431
33.6	Drag & Drop-Datenbindung	1431
33.6.1	Vorgehensweise	1432
33.6.2	Weitere Möglichkeiten	1434
33.7	Formatieren von Werten	1435
33.7.1	IValueConverter	1436
33.7.2	BindingBase.StringFormat-Eigenschaft	1438
33.8	Das DataGrid als Universalwerkzeug	1439
33.8.1	Grundlagen der Anzeige	1440
33.8.2	Vom Betrachten zum Editieren	1444
34	Drucken/Druckvorschau mit WPF	1445
34.1	Grundlagen	1445
34.1.1	XPS-Dokumente	1445
34.1.2	System.Printing	1446
34.1.3	System.Windows.Xps	1447
34.2	Einfache Druckausgaben mit dem PrintDialog	1447
34.3	Mehrseitige Dokumente und Druckvorschau-Funktion	1450
34.3.1	Fix-Dokumente	1450
34.3.2	Flow-Dokumente	1456
34.4	Druckerinfos und Druckerauswahl/-konfiguration	1459
34.4.1	Die installierten Drucker bestimmen	1460
34.4.2	Den Standarddrucker bestimmen	1461
34.4.3	Mehr über einzelne Drucker erfahren	1461
34.4.4	Spezifische Druckeinstellungen vornehmen	1463
34.4.5	Direkte Druckausgabe	1465
35	WPF-Entwicklung	1467
35.1	Entwicklungstools für WPF-Anwendungen	1467
35.1.1	Microsoft Visual Studio 2010	1467
35.1.2	Microsoft Expression Blend	1468
35.1.3	Weitere Tools	1472
35.2	Distribution	1472
35.2.1	.NET Framework 4 Client Profil	1473

35.2.2	Was fehlt im Client-Profil?	1474
35.2.3	Installationsgröße weiter verringern	1474
36	Silverlight-Entwicklung	1475
36.1	Einführung	1475
36.1.1	Zielpattformen	1476
36.1.2	Silverlight-Applikationstypen	1476
36.1.3	Wichtige Unterschiede zu den WPF-Anwendungen	1478
36.1.4	Vor- und Nachteile von Silverlight-Anwendungen	1480
36.1.5	Entwicklungstools	1482
36.1.6	Installation auf dem Client	1482
36.2	Die Silverlight-Anwendung im Detail	1484
36.2.1	Ein kleines Beispielprojekt	1484
36.2.2	Das Application Package und das Test-Web	1486
36.3	Die Projektdateien im Überblick	1490
36.3.1	Projektverwaltung mit App.xaml & App.xaml.cs	1490
36.3.2	MainPage.xaml & MainPage.xaml.cs	1492
36.3.3	AssemblyInfo.cs	1493
36.4	Fenster und Seiten in Silverlight	1494
36.4.1	Das Standardfenster	1494
36.4.2	Untergeordnete Silverlight-Fenster	1495
36.4.3	UserControls für die Anzeige von Detaildaten verwenden	1497
36.4.4	Navigieren in Silverlight-Anwendungen	1498
36.5	Datenbanken/Datenbindung	1503
36.5.1	ASP.NET Webdienste/WCF Dienste	1504
36.5.2	WCF Data Services	1512
36.6	Isolierter Speicher	1523
36.6.1	Grundkonzept	1523
36.6.2	Das virtuelle Dateisystem verwalten	1524
36.6.3	Arbeiten mit Dateien	1526
36.7	Praxisbeispiele	1528
36.7.1	Eine Out-of-Browser-Applikation realisieren	1528
36.7.2	Out-of-Browser-Anwendung aktualisieren	1532
36.7.3	Testen, ob die Anwendung mit dem Internet verbunden ist	1533
36.7.4	Auf Out-of-Browser-Anwendung testen	1534
36.7.5	Den Browser bestimmen	1534
36.7.6	Parameter an das Plug-in übergeben	1535
36.7.7	Auf den QueryString zugreifen	1537

36.7.8	Timer in Silverlight nutzen	1538
36.7.9	Dateien lokal speichern	1539

Teil IV: ASP.NET-Anwendungen

37 Einführung in ASP.NET 1543

37.1	ASP.NET für Ein- und Umsteiger	1543
37.1.1	ASP – der Blick zurück	1543
37.1.2	Was ist bei ASP.NET anders?	1544
37.1.3	Was gibt es noch in ASP.NET?	1546
37.1.4	Vorteile von ASP.NET gegenüber ASP	1547
37.1.5	Voraussetzungen für den Einsatz von ASP.NET	1548
37.1.6	Und was hat das alles mit C# zu tun?	1549
37.2	Unsere erste Web-Anwendung	1551
37.2.1	Visueller Entwurf der Bedienoberfläche	1551
37.2.2	Zuweisen der Objekteigenschaften	1554
37.2.3	Verknüpfen der Objekte mit Ereignissen	1555
37.2.4	Programm kompilieren und testen	1556
37.3	Die ASP.NET-Projektdateien	1557
37.3.1	Die ASP.NET-Projekttypen	1558
37.3.2	ASPX-Datei(en)	1559
37.3.3	Die aspx.cs-Datei(en)	1562
37.3.4	Die Datei Global.asax	1563
37.3.5	Das Startformular	1563
37.3.6	Die Datei Web.config	1564
37.3.7	Masterpages (master-Dateien)	1567
37.3.8	Sitemap (Web.sitemap)	1567
37.3.9	Benutzersteuerelemente (ascx-Dateien)	1567
37.3.10	Die Web-Projekt-Verzeichnisse	1568
37.4	Lernen am Beispiel	1568
37.4.1	Erstellen des Projekts	1569
37.4.2	Oberflächengestaltung	1570
37.4.3	Ereignisprogrammierung	1571
37.4.4	Ein Fehler, was nun?	1572
37.4.5	Ereignisse von Textboxen	1574
37.4.6	Ein gemeinsamer Ereignis-Handler	1574
37.4.7	Eingabefokus setzen	1575

37.4.8	Ausgaben in einer Tabelle	1575
37.4.9	Scrollen der Anzeige	1578
37.4.10	Zusammenspiel mehrerer Formulare	1579
37.4.11	Umleiten bei Direktaufruf	1580
37.4.12	Ärger mit den Cookies	1581
37.4.13	Export auf den IIS	1582
38	Übersicht ASP.NET-Controls	1585
38.1	Einfache Steuerelemente im Überblick	1585
38.1.1	Label	1585
38.1.2	TextBox	1587
38.1.3	Button, ImageButton, LinkButton	1588
38.1.4	CheckBox, RadioButton	1589
38.1.5	CheckBoxList, BulletList, RadioButtonList	1590
38.1.6	Table	1591
38.1.7	Hyperlink	1593
38.1.8	Image, ImageMap	1593
38.1.9	Calendar	1595
38.1.10	Panel	1596
38.1.11	HiddenField	1596
38.1.12	Substitution	1597
38.1.13	XML	1598
38.1.14	FileUpload	1600
38.1.15	AdRotator	1601
38.2	Steuerelemente für die Seitennavigation	1602
38.2.1	Mehr Übersicht mit Web.Sitemap	1603
38.2.2	Menu	1604
38.2.3	TreeView	1607
38.2.4	SiteMapPath	1610
38.2.5	MultiView, View	1611
38.2.6	Wizard	1612
38.3	Webseitenlayout/-design	1614
38.3.1	Masterpages	1614
38.3.2	Themes/Skins	1617
38.3.3	Webparts	1620
38.4	Die Validator-Controls	1621
38.4.1	Übersicht	1621
38.4.2	Wo findet die Fehlerprüfung statt?	1622

38.4.3	Verwendung	1622
38.4.4	RequiredFieldValidator	1623
38.4.5	CompareValidator	1624
38.4.6	RangeValidator	1626
38.4.7	RegularExpressionValidator	1626
38.4.8	CustomValidator	1627
38.4.9	ValidationSummary	1630
38.4.10	Weitere Möglichkeiten	1631
38.5	Praxisbeispiele	1631
38.5.1	Themes und Skins verstehen	1631
38.5.2	Masterpages verwenden	1636
38.5.3	Webparts verwenden	1639
39	Datenbindung unter ASP.NET	1645
39.1	Alt und Neu im Vergleich	1645
39.1.1	Variante 1: Konventionelle Programmierung	1645
39.1.2	Variante 2: Mit DataSource	1648
39.2	Einführung	1651
39.2.1	Konzept	1652
39.2.2	Übersicht DataSource-Steuerelemente	1653
39.3	SQLDataSource	1654
39.3.1	Datenauswahl mit Parametern	1656
39.3.2	Parameter für INSERT, UPDATE und DELETE	1657
39.3.3	Methoden	1658
39.3.4	Caching	1660
39.3.5	Aktualisieren/Refresh	1660
39.4	AccessDataSource	1660
39.5	ObjectDataSource	1660
39.5.1	Verbindung zwischen Objekt und DataSource	1661
39.5.2	Ein Beispiel sorgt für Klarheit	1662
39.5.3	Geschäftsobjekte in einer Session verwalten	1666
39.6	SitemapDataSource	1668
39.7	LinqDataSource	1669
39.7.1	Bindung von einfachen Collections	1669
39.7.2	Bindung eines LINQ to SQL-DataContext	1671
39.8	EntityDataSource	1672
39.8.1	Entity Data Model erstellen	1673

39.8.2	EntityDataSource anbinden	1675
39.8.3	Datenmenge filtern	1678
39.9	XmlDataSource	1678
39.10	QueryExtender	1680
39.10.1	Grundlagen	1680
39.10.2	Suchen	1681
39.10.3	Sortieren	1684
39.11	GridView	1684
39.11.1	Auswahlfunktion (Zeilenauswahl)	1685
39.11.2	Auswahl mit mehrspaltigem Index	1686
39.11.3	Hyperlink-Spalte für Detailansicht	1686
39.11.4	Spalten erzeugen	1687
39.11.5	Paging realisieren	1688
39.11.6	Edit, Update, Delete	1689
39.11.7	Keine Daten, was tun?	1690
39.12	DetailsView	1690
39.13	FormView	1692
39.14	DataList	1695
39.14.1	Bearbeitungsfunktionen implementieren	1695
39.14.2	Layout verändern	1697
39.15	Repeater	1697
39.16	ListView	1699
39.17	Chart	1699
40	ASP.NET-Objekte und Techniken	1701
40.1	Wichtige ASP.NET-Objekte	1701
40.1.1	HTTPApplication	1701
40.1.2	Application	1704
40.1.3	Session	1705
40.1.4	Page	1707
40.1.5	Request	1710
40.1.6	Response	1713
40.1.7	Server	1717
40.1.8	Cookies verwenden	1718
40.2	Fehlerbehandlung unter ASP.NET	1721
40.2.1	Fehler beim Entwurf	1721
40.2.2	Laufzeitfehler	1721
40.2.3	Eine eigene Fehlerseite	1723

40.2.4	Fehlerbehandlung im Web Form	1724
40.2.5	Fehlerbehandlung in der Anwendung	1725
40.2.6	Alternative Fehlerseite einblenden	1726
40.2.7	Lokale Fehlerbehandlung	1727
40.2.8	Seite nicht gefunden – was nun?	1728
40.3	E-Mail-Versand in ASP.NET	1728
40.3.1	Übersicht	1729
40.3.2	Mail-Server bestimmen	1729
40.3.3	Einfache Text-E-Mails versenden	1731
40.3.4	E-Mails mit Dateianhang	1732
40.4	Sicherheit von Webanwendungen	1733
40.4.1	Authentication	1733
40.4.2	Forms Authentication realisieren	1734
40.4.3	Impersonation	1738
40.4.4	Authorization	1739
40.4.5	Administrieren der Website	1741
40.4.6	Steuerelemente für das Login-Handling	1745
40.4.7	Programmieren der Sicherheitseinstellungen	1749
40.5	Die Verwendung von AJAX in ASP.NET-Anwendungen	1751
40.5.1	Was zum Teufel ist AJAX und was kann es?	1751
40.5.2	Die AJAX-Controls	1752
40.5.3	AJAX-Control-Toolkit	1756
40.6	User Controls/Webbenutzersteuerelemente	1757
40.6.1	Ein simples Einstiegsbeispiel	1757
40.6.2	Dynamische Grafiken in einem User Control anzeigen	1761
40.6.3	Grafikausgaben per User Control realisieren	1766

Anhang

A	Glossar	1771
B	Wichtige Dateiextensions	1777
	Index	1779