

# Auf einen Blick

1	Allgemeine Einführung in .NET .....	33
2	Grundlagen der Sprache C# .....	57
3	Das Klassendesign .....	133
4	Vererbung, Polymorphie und Interfaces .....	205
5	Delegates und Ereignisse .....	259
6	Strukturen und Enumerationen .....	289
7	Fehlerbehandlung und Debugging .....	299
8	Auflistungsklassen (Collections) .....	333
9	Generics – Generische Datentypen .....	359
10	Weitere C#-Sprachfeatures .....	385
11	LINQ .....	435
12	Arbeiten mit Dateien und Streams .....	467
13	Binäre Serialisierung .....	511
14	XML .....	519
15	Multithreading und die Task Parallel Library (TPL) .....	613
16	Einige wichtige .NET-Klassen .....	665
17	Projektmanagement und Visual Studio 2012 .....	705
18	Einführung in die WPF und XAML .....	759
19	WPF-Layout-Container .....	789
20	Fenster in der WPF .....	809
21	WPF-Steuerelemente .....	833
22	Elementbindungen .....	921
23	Konzepte von WPF .....	935
24	Datenbindung .....	975
25	Weitere Möglichkeiten der Datenbindung .....	1005
26	Dependency Properties .....	1041
27	Ereignisse in der WPF .....	1057
28	WPF-Commands .....	1077
29	Benutzerdefinierte Controls .....	1097
30	2D-Grafik .....	1107
31	ADO.NET – Verbindungsorientierte Objekte .....	1127
32	ADO.NET – Das Command-Objekt .....	1149
33	ADO.NET – Der SqlDataAdapter .....	1177
34	ADO.NET – Daten im lokalen Speicher .....	1191
35	ADO.NET – Aktualisieren der Datenbank .....	1231
36	Stark typisierte DataSets .....	1251
37	Einführung in das ADO.NET Entity Framework .....	1277
38	Datenabfragen des Entity Data Models (EDM) .....	1301
39	Entitätsaktualisierung und Zustandsverwaltung .....	1335
40	Konflikte behandeln .....	1363
41	Plain Old CLR Objects (POCOs) .....	1371

# Inhalt

Vorwort zur 6. Auflage .....	31
<b>1 Allgemeine Einführung in .NET</b> .....	<b>33</b>
<b>1.1 Warum .NET?</b> .....	<b>33</b>
1.1.1 Ein paar Worte zu diesem Buch .....	35
1.1.2 Die Beispielprogramme .....	37
<b>1.2 .NET unter die Lupe genommen</b> .....	<b>38</b>
1.2.1 Das Entwicklerdilemma .....	38
1.2.2 .NET – Ein paar allgemeine Eigenschaften .....	39
1.2.3 Das Sprachenkonzept .....	40
1.2.4 Die »Common Language Specification« (CLS) .....	42
1.2.5 Das »Common Type System« (CTS) .....	43
1.2.6 Das .NET Framework .....	44
1.2.7 Die »Common Language Runtime« (CLR) .....	45
1.2.8 Die .NET-Klassenbibliothek .....	45
1.2.9 Das Konzept der Namespaces .....	47
<b>1.3 Assemblys</b> .....	<b>48</b>
1.3.1 Die Metadaten .....	49
1.3.2 Das Manifest .....	50
<b>1.4 Die Entwicklungsumgebung</b> .....	<b>50</b>
1.4.1 Editionen von Visual Studio 2012 .....	50
1.4.2 Hard- und Softwareanforderungen .....	51
1.4.3 Die Installation .....	51
1.4.4 Die Entwicklungsumgebung von Visual Studio 2012 .....	52
<b>2 Grundlagen der Sprache C#</b> .....	<b>57</b>
<b>2.1 Konsolenanwendungen</b> .....	<b>57</b>
2.1.1 Allgemeine Anmerkungen .....	57
2.1.2 Ein erstes Konsolenprogramm .....	57
<b>2.2 Grundlagen der C#-Syntax</b> .....	<b>60</b>
2.2.1 Kennzeichnen, dass eine Anweisung abgeschlossen ist .....	60
2.2.2 Anweisungs- und Gliederungsblöcke .....	61
2.2.3 Kommentare .....	62
2.2.4 Die Groß- und Kleinschreibung .....	63
2.2.5 Die Struktur einer Konsolenanwendung .....	64

<b>2.3</b>	<b>Variablen und Datentypen</b>	66
2.3.1	Variablendeklaration	66
2.3.2	Der Variablenbezeichner	67
2.3.3	Der Zugriff auf eine Variable	68
2.3.4	Ein- und Ausgabemethoden der Klasse »Console«	68
2.3.5	Die einfachen Datentypen	74
2.3.6	Typkonvertierung	80
<b>2.4</b>	<b>Operatoren</b>	87
2.4.1	Arithmetische Operatoren	88
2.4.2	Vergleichsoperatoren	91
2.4.3	Logische Operatoren	91
2.4.4	Bitweise Operatoren	95
2.4.5	Zuweisungsoperatoren	98
2.4.6	Stringverkettung	98
2.4.7	Sonstige Operatoren	99
2.4.8	Operator-Vorrangregeln	99
<b>2.5</b>	<b>Datenfelder (Arrays)</b>	100
2.5.1	Die Deklaration und Initialisierung eines Arrays	100
2.5.2	Der Zugriff auf die Array-Elemente	102
2.5.3	Mehrdimensionale Arrays	103
2.5.4	Festlegen der Array-Größe zur Laufzeit	104
2.5.5	Bestimmung der Array-Obergrenze	105
2.5.6	Die Gesamtanzahl der Array-Elemente	106
2.5.7	Verzweigte Arrays	106
<b>2.6</b>	<b>Kontrollstrukturen</b>	108
2.6.1	Die »if«-Anweisung	108
2.6.2	Das »switch«-Statement	113
<b>2.7</b>	<b>Programmschleifen</b>	117
2.7.1	Die »for«-Schleife	118
2.7.2	Die »foreach«-Schleife	128
2.7.3	Die »do«- und die »while«-Schleife	129
<b>3</b>	<b>Das Klassendesign</b>	133
<b>3.1</b>	<b>Einführung in die Objektorientierung</b>	133
<b>3.2</b>	<b>Die Klassendefinition</b>	136
3.2.1	Klassen im Visual Studio anlegen	136
3.2.2	Das Projekt »GeometricObjectsSolution«	136
3.2.3	Die Deklaration von Objektvariablen	139
3.2.4	Zugriffsmodifizierer einer Klasse	140

3.2.5	Splitten einer Klassendefinition mit »partial«	140
3.2.6	Arbeiten mit Objektreferenzen	141
<b>3.3</b>	<b>Referenz- und Wertetypen</b>	143
3.3.1	Werte- und Referenztypen nutzen	144
<b>3.4</b>	<b>Die Eigenschaften eines Objekts</b>	145
3.4.1	Öffentliche Felder	145
3.4.2	Datenkapselung mit Eigenschaftsmethoden sicherstellen	147
3.4.3	Die Ergänzung der Klasse »Circle«	149
3.4.4	Les- und Schreibgeschützte Eigenschaften	149
3.4.5	Sichtbarkeit der Accessoren »get« und »set«	150
3.4.6	Unterstützung von Visual Studio 2012	151
3.4.7	Automatisch implementierte Eigenschaften	152
<b>3.5</b>	<b>Methoden eines Objekts</b>	153
3.5.1	Methoden mit Rückgabewert	153
3.5.2	Methoden ohne Rückgabewert	157
3.5.3	Methoden mit Parameterliste	157
3.5.4	Methodenüberladung	159
3.5.5	Variablen innerhalb einer Methode (lokale Variablen)	162
3.5.6	Referenz- und Wertparameter	163
3.5.7	Besondere Aspekte einer Parameterliste	168
3.5.8	Zugriff auf private Daten	173
3.5.9	Die Trennung von Daten und Code	174
3.5.10	Namenskonflikte mit »this« lösen	175
3.5.11	Methode oder Eigenschaft?	176
3.5.12	Umbenennen von Methoden und Eigenschaften	177
<b>3.6</b>	<b>Konstrukturen</b>	178
3.6.1	Konstrukturen bereitstellen	179
3.6.2	Die Konstruktoraufrufe	180
3.6.3	Definition von Konstrukturen	180
3.6.4	»public«- und »internal«-Konstrukturen	181
3.6.5	»private«-Konstrukturen	181
3.6.6	Konstruktorenaufrufe umleiten	182
3.6.7	Vereinfachte Objektinitialisierung	183
<b>3.7</b>	<b>Der Destruktor</b>	184
<b>3.8</b>	<b>Konstanten in einer Klasse</b>	185
3.8.1	Konstanten mit dem Schlüsselwort »const«	185
3.8.2	Schreibgeschützte Felder mit »readonly«	186
<b>3.9</b>	<b>Statische Klassenkomponenten</b>	186
3.9.1	Statische Eigenschaften	186
3.9.2	Statische Methoden	189

3.9.3	Statische Klasseninitialisierer .....	190
3.9.4	Statische Klassen .....	191
3.9.5	Statische Klasse oder Singleton-Pattern? .....	192
<b>3.10</b>	<b>Namensräume (Namespaces) .....</b>	<b>193</b>
3.10.1	Zugriff auf Namespaces .....	194
3.10.2	Die »using«-Direktive .....	196
3.10.3	Globaler Namespace .....	197
3.10.4	Vermeiden von Mehrdeutigkeiten .....	197
3.10.5	Namespaces festlegen .....	198
3.10.6	Der »::«-Operator .....	200
3.10.7	Unterstützung von Visual Studio 2012 bei den Namespaces .....	201
<b>3.11</b>	<b>Stand der Klasse »Circle« .....</b>	<b>203</b>
<b>4</b>	<b>Vererbung, Polymorphie und Interfaces .....</b>	<b>205</b>
<b>4.1</b>	<b>Die Vererbung .....</b>	<b>205</b>
4.1.1	Basisklassen und abgeleitete Klassen .....	205
4.1.2	Die Ableitung einer Klasse .....	206
4.1.3	Klassen, die nicht abgeleitet werden können .....	208
4.1.4	Konstruktoren in abgeleiteten Klassen .....	208
4.1.5	Der Zugriffsmodifizierer »protected« .....	209
4.1.6	Die Konstruktorverkettung in der Vererbung .....	210
<b>4.2</b>	<b>Der Problemfall geerbter Methoden .....</b>	<b>214</b>
4.2.1	Geerbte Methoden mit »new« verdecken .....	216
4.2.2	Abstrakte Methoden .....	217
4.2.3	Virtuelle Methoden .....	219
<b>4.3</b>	<b>Typumwandlung und Typuntersuchung von Objektvariablen .....</b>	<b>220</b>
4.3.1	Die implizite Typumwandlung von Objektreferenzen .....	220
4.3.2	Die explizite Typumwandlung von Objektreferenzen .....	222
4.3.3	Typuntersuchung mit dem »is«-Operator .....	223
4.3.4	Typumwandlung mit dem »as«-Operator .....	224
<b>4.4</b>	<b>Polymorphie .....</b>	<b>224</b>
4.4.1	Die »klassische« Methodenimplementierung .....	225
4.4.2	Abstrakte Methoden .....	226
4.4.3	Virtuelle Methoden .....	227
<b>4.5</b>	<b>Weitere Gesichtspunkte der Vererbung .....</b>	<b>230</b>
4.5.1	Versiegelte Methoden .....	230
4.5.2	Überladen einer Basisklassenmethode .....	231
4.5.3	Statische Member und Vererbung .....	232
4.5.4	Geerbte Methoden ausblenden? .....	232

<b>4.6</b>	<b>Das Projekt »GeometricObjectsSolution« ergänzen</b>	233
4.6.1	Die Klasse »GeometricObject«	233
<b>4.7</b>	<b>Eingebettete Klassen (Nested Classes)</b>	237
<b>4.8</b>	<b>Interfaces (Schnittstellen)</b>	237
4.8.1	Einführung in die Schnittstellen	237
4.8.2	Die Schnittstellendefinition	238
4.8.3	Die Schnittstellenimplementierung	239
4.8.4	Die Interpretation der Schnittstellen	244
4.8.5	Änderungen am Projekt »GeometricObjects«	249
<b>4.9</b>	<b>Das Zerstören von Objekten – der »Garbage Collector«</b>	251
4.9.1	Die Arbeitsweise des Garbage Collectors	251
4.9.2	Expliziter Aufruf des Garbage Collectors	252
4.9.3	Der Destruktor	253
4.9.4	Die »IDisposable«-Schnittstelle	254
4.9.5	Die Ergänzungen in den Klassen »Circle« und »Rectangle«	257
<b>5</b>	<b>Delegates und Ereignisse</b>	259
<b>5.1</b>	<b>Delegates</b>	259
5.1.1	Einführung in das Prinzip der Delegates	259
5.1.2	Verwendung von Delegates	263
5.1.3	Vereinfachter Delegatenaufruf	263
5.1.4	Multicast-Delegates	264
5.1.5	Anonyme Methoden	266
5.1.6	Kovarianz und Kontravarianz mit Delegaten	268
<b>5.2</b>	<b>Ereignisse eines Objekts</b>	270
5.2.1	Ereignisse bereitstellen	271
5.2.2	Die Reaktion auf ein ausgelöstes Ereignis	273
5.2.3	Allgemeine Betrachtungen der Ereignishandler-Registrierung	275
5.2.4	Wenn der Ereignisempfänger ein Ereignis nicht behandelt	276
5.2.5	Ereignisse mit Übergabeparameter	277
5.2.6	Ereignisse in der Vererbung	281
5.2.7	Hinter die Kulissen des Schlüsselworts »event« geblickt	282
5.2.8	Die Schnittstelle »INotifyPropertyChanged«	284
<b>5.3</b>	<b>Änderungen im Projekt »GeometricObjects«</b>	285
5.3.1	Überarbeitung des Events »InvalidMeasure«	285
5.3.2	Weitere Ereignisse	286

<b>6</b>	<b>Strukturen und Enumerationen</b>	<b>289</b>
<b>6.1</b>	<b>Strukturen – eine Sonderform der Klassen</b>	<b>289</b>
6.1.1	Die Definition einer Struktur	289
6.1.2	Initialisieren einer Strukturvariablen	290
6.1.3	Konstruktoren in Strukturen	291
6.1.4	Änderung im Projekt »GeometricObjects«	292
<b>6.2</b>	<b>Enumerationen (Aufzählungen)</b>	<b>295</b>
6.2.1	Wertzuweisung an enum-Mitglieder	296
6.2.2	Alle Mitglieder einer Aufzählung durchlaufen	297
<b>6.3</b>	<b>Boxing und Unboxing</b>	<b>298</b>
<b>7</b>	<b>Fehlerbehandlung und Debugging</b>	<b>299</b>
<b>7.1</b>	<b>Laufzeitfehler behandeln</b>	<b>299</b>
7.1.1	Laufzeitfehler erkennen	300
7.1.2	Die »try...catch«-Anweisung	302
7.1.3	Behandlung mehrerer Exceptions	304
7.1.4	Die Reihenfolge der »catch«-Zweige	306
7.1.5	Ausnahmen in einer Methodenaufkette	307
7.1.6	Ausnahmen werfen oder weiterleiten	307
7.1.7	Die »finally«-Anweisung	308
7.1.8	Die Klasse »Exception«	309
7.1.9	Benutzerdefinierte Ausnahmen	314
<b>7.2</b>	<b>Debuggen mit Programmcode</b>	<b>319</b>
7.2.1	Einführung	319
7.2.2	Die Klasse »Debug«	320
7.2.3	Die Klasse »Trace«	324
7.2.4	Bedingte Kompilierung	324
<b>7.3</b>	<b>Fehlersuche mit Visual Studio 2012</b>	<b>327</b>
7.3.1	Debuggen im Haltemodus	327
7.3.2	Das »Direktfenster«	330
7.3.3	Weitere Alternativen, um Variableninhalte zu prüfen	331
<b>8</b>	<b>Auflistungsklassen (Collections)</b>	<b>333</b>
<b>8.1</b>	<b>Grundlagen</b>	<b>333</b>
<b>8.2</b>	<b>Collections im Namespace »System.Collections«</b>	<b>333</b>
8.2.1	Die elementaren Schnittstellen der Auflistungsklassen	335

<b>8.3</b>	<b>Die Klasse »ArrayList«</b>	337
8.3.1	Einträge hinzufügen	337
8.3.2	Datenaustausch zwischen einem Array und einer »ArrayList«	340
8.3.3	Die Elemente einer »ArrayList« sortieren	341
8.3.4	Sortieren von Arrays mit »ArrayList.Adapter«	346
<b>8.4</b>	<b>Die Klasse »Hashtable«</b>	348
8.4.1	Methoden und Eigenschaften der Schnittstelle »IDictionary«	348
8.4.2	Beispielprogramm zur Klasse »Hashtable«	349
<b>8.5</b>	<b>Die Klassen »Queue« und »Stack«</b>	353
8.5.1	Die Klasse »Stack«	354
8.5.2	Die Klasse »Queue«	355
<b>8.6</b>	<b>Eigene Auflistungen mit »yield« durchlaufen</b>	356
<b>9</b>	<b>Generics – Generische Datentypen</b>	359
<b>9.1</b>	<b>Problembeschreibung</b>	359
<b>9.2</b>	<b>Bereitstellen einer generischen Klasse</b>	360
9.2.1	Mehrere generische Typparameter	362
9.2.2	Vorteile der Generics	363
<b>9.3</b>	<b>Bedingungen (Constraints) festlegen</b>	363
9.3.1	Constraints mit der »where«-Klausel	363
9.3.2	Typparameter auf Klassen oder Strukturen beschränken	365
9.3.3	Mehrere Constraints definieren	365
9.3.4	Der Konstruktor-Constraint »new()«	366
9.3.5	Das Schlüsselwort »default«	366
<b>9.4</b>	<b>Generische Methoden</b>	367
9.4.1	Methoden und Constraints	368
<b>9.5</b>	<b>Generics und Vererbung</b>	368
9.5.1	Virtuelle generische Methoden	369
<b>9.6</b>	<b>Konvertierung von Generics</b>	370
<b>9.7</b>	<b>Generische Delegates</b>	371
9.7.1	Generische Delegates und Constraints	372
9.7.2	Anpassung des Beispiels »GeometricObjects«	372
<b>9.8</b>	<b>Nullable-Typen</b>	373
9.8.1	Konvertierungen mit Nullable-Typen	374
<b>9.9</b>	<b>Generische Collections</b>	374
9.9.1	Die Interfaces der generischen Auflistungsklassen	375
9.9.2	Die generische Auflistungsklasse »List<T>«	375
9.9.3	Vergleiche mit Hilfe des Delegates »Comparison<T>«	378



<b>9.10 Kovarianz und Kontravarianz generischer Typen</b>	379
9.10.1 Kovarianz mit Interfaces	379
9.10.2 Kontravarianz mit Interfaces	381
9.10.3 Zusammenfassung	382
9.10.4 Generische Delegaten mit varianten Typparametern	383
 <b>10 Weitere C#-Sprachfeatures</b>	 385
<b>10.1 Implizit typisierte Variablen</b>	385
<b>10.2 Anonyme Typen</b>	386
<b>10.3 Lambda-Ausdrücke</b>	387
10.3.1 Projektion und Prädikat	389
<b>10.4 Erweiterungsmethoden</b>	389
<b>10.5 Partielle Methoden</b>	393
10.5.1 Wo partielle Methoden eingesetzt werden	394
<b>10.6 Operatorüberladung</b>	396
10.6.1 Einführung	396
10.6.2 Die Syntax der Operatorüberladung	396
10.6.3 Die Operatorüberladungen im Projekt »GeometricObjectsSolution«	397
10.6.4 Die Operatoren »true« und »false« überladen	402
10.6.5 Benutzerdefinierte Konvertierungen	403
<b>10.7 Indexer</b>	407
10.7.1 Überladen von Indexern	409
10.7.2 Parameterbehaftete Eigenschaften	411
<b>10.8 Attribute</b>	414
10.8.1 Das »Flags«-Attribut	415
10.8.2 Benutzerdefinierte Attribute	418
10.8.3 Attribute auswerten	422
10.8.4 Festlegen der Assembly-Eigenschaften in »Assembly-Info.cs«	424
<b>10.9 Dynamisches Binden</b>	426
10.9.1 Eine kurze Analyse	427
10.9.2 Dynamische Objekte	427
<b>10.10 Unsicherer (unsafe) Programmcode – Zeigertechnik in C#</b>	429
10.10.1 Einführung	429
10.10.2 Das Schlüsselwort »unsafe«	429
10.10.3 Die Deklaration von Zeigern	430
10.10.4 Die »fixed«-Anweisung	431
10.10.5 Zeigerarithmetik	432
10.10.6 Der Operator »->«	433

<b>11</b>	<b>LINQ</b>	<b>435</b>
<b>11.1</b>	<b>Was ist LINQ?</b>	<b>435</b>
11.1.1	Verzögerte Ausführung	436
11.1.2	LINQ-Erweiterungsmethoden an einem Beispiel	437
<b>11.2</b>	<b>LINQ to Objects</b>	<b>440</b>
11.2.1	Musterdaten	440
11.2.2	Die allgemeine LINQ-Syntax	442
<b>11.3</b>	<b>Die Abfrageoperatoren</b>	<b>444</b>
11.3.1	Übersicht der Abfrageoperatoren	444
11.3.2	Die »from«-Klausel	445
11.3.3	Mit »where« filtern	446
11.3.4	Die Projektionsoperatoren	449
11.3.5	Die Sortieroperatoren	450
11.3.6	Gruppieren mit »GroupBy«	451
11.3.7	Verknüpfungen mit »Join«	453
11.3.8	Die Set-Operatoren-Familie	456
11.3.9	Die Familie der Aggregatoperatoren	457
11.3.10	Quantifizierungsoperatoren	460
11.3.11	Aufteilungsoperatoren	461
11.3.12	Die Elementoperatoren	463
11.3.13	Die Konvertierungsoperatoren	466
<b>12</b>	<b>Arbeiten mit Dateien und Streams</b>	<b>467</b>
<b>12.1</b>	<b>Einführung</b>	<b>467</b>
<b>12.2</b>	<b>Namespaces der Ein- bzw. Ausgabe</b>	<b>468</b>
12.2.1	Das Behandeln von Ausnahmen bei E/A-Operationen	469
<b>12.3</b>	<b>Laufwerke, Verzeichnisse und Dateien</b>	<b>469</b>
12.3.1	Die Klasse »File«	469
12.3.2	Die Klasse »FileInfo«	475
12.3.3	Die Klassen »Directory« und »DirectoryInfo«	478
12.3.4	Die Klasse »Path«	482
12.3.5	Die Klasse »DriveInfo«	484
<b>12.4</b>	<b>Die »Stream«-Klassen</b>	<b>485</b>
12.4.1	Die abstrakte Klasse »Stream«	486
12.4.2	Die von »Stream« abgeleiteten Klassen im Überblick	488
12.4.3	Die Klasse »FileStream«	489
<b>12.5</b>	<b>Die Klassen »TextReader« und »TextWriter«</b>	<b>496</b>

12.5.1	Die Klasse »StreamWriter« .....	496
12.5.2	Die Klasse »StreamReader« .....	500
<b>12.6</b>	<b>Die Klassen »BinaryReader« und »BinaryWriter« .....</b>	<b>502</b>
12.6.1	Komplexe binäre Dateien .....	504
<b>13</b>	<b>Binäre Serialisierung .....</b>	<b>511</b>
<b>13.1</b>	<b>Einführung in die Serialisierung .....</b>	<b>511</b>
13.1.1	Serialisierungsverfahren .....	512
<b>13.2</b>	<b>Serialisierung mit »BinaryFormatter« .....</b>	<b>513</b>
13.2.1	Die Deserialisierung .....	515
13.2.2	Serialisierung mehrerer Objekte .....	516
<b>14</b>	<b>XML .....</b>	<b>519</b>
<b>14.1</b>	<b>Grundlagen .....</b>	<b>519</b>
<b>14.2</b>	<b>XML-Dokumente .....</b>	<b>519</b>
14.2.1	Wohlgeformte und gültige XML-Dokumente .....	520
14.2.2	Die Regeln eines wohlgeformten XML-Codes .....	522
14.2.3	Kommentare .....	525
14.2.4	Verarbeitungsanweisungen .....	525
14.2.5	Reservierte Zeichen in XML .....	526
14.2.6	CDATA-Abschnitte .....	526
14.2.7	Namensräume (Namespaces) .....	527
<b>14.3</b>	<b>Die Gültigkeit eines XML-Dokuments .....</b>	<b>534</b>
14.3.1	XML Schema Definition (XSD) .....	535
14.3.2	Ein XML-Dokument mit einem XML-Schema verknüpfen .....	536
14.3.3	Die Struktur eines XML-Schemas .....	539
<b>14.4</b>	<b>Die Klasse »XmlReader« .....</b>	<b>545</b>
14.4.1	XML-Dokumente mit einem »XmlReader«-Objekt lesen .....	545
14.4.2	Validieren eines XML-Dokuments .....	551
<b>14.5</b>	<b>Eigenschaften und Methoden der Klasse »XmlReader« .....</b>	<b>554</b>
<b>14.6</b>	<b>Die Klasse »XmlWriter« .....</b>	<b>557</b>
14.6.1	Die Methoden der Klasse »XmlWriter« .....	561
<b>14.7</b>	<b>Navigation durch XML (XPath) .....</b>	<b>562</b>
14.7.1	Die Klasse »XPathNavigator« .....	562
14.7.2	XPath-Ausdrücke .....	566
14.7.3	Der Kontextknoten .....	567
14.7.4	Beispiele mit XPath-Ausdrücken .....	569

14.7.5	Knotenmengen mit der »Select«-Methode .....	571
14.7.6	Auswerten von XPath-Ausdrücken .....	575
<b>14.8</b>	<b>Das Document Object Model (DOM)</b> .....	579
14.8.1	Allgemeines .....	579
14.8.2	Arbeiten mit »XmlDocument« .....	581
14.8.3	»XmlDocument« und »XPathNavigator« .....	582
14.8.4	Die Klasse »XmlNode« (Operationen mit Knoten) .....	582
14.8.5	Manipulieren einer XML-Struktur .....	590
14.8.6	Ändern eines Knotens .....	592
14.8.7	Löschen in einem XML-Dokument .....	594
<b>14.9</b>	<b>Serialisierung mit »XmlSerializer«</b> .....	596
14.9.1	XML-Serialisierung mit Attributen steuern .....	598
<b>14.10</b>	<b>LINQ to XML</b> .....	601
14.10.1	Allgemeines .....	601
14.10.2	Die Klassenhierarchie von LINQ to XML .....	601
14.10.3	Die Klasse »XElement« .....	602
14.10.4	Die Klasse »XDocument« .....	605
14.10.5	Navigation im XML-Dokument .....	605
14.10.6	Änderungen am XML-Dokument vornehmen .....	611
<b>15</b>	<b>Multithreading und die Task Parallel Library (TPL)</b> .....	613
<b>15.1</b>	<b>Überblick</b> .....	613
<b>15.2</b>	<b>Multithreading mit der Klasse »Thread«</b> .....	614
15.2.1	Einführung in das Multithreading .....	614
15.2.2	Threadzustände und Prioritäten .....	614
15.2.3	Zusammenspiel mehrerer Threads .....	616
15.2.4	Die Entwicklung einer einfachen Multithreading-Anwendung .....	616
15.2.5	Die Klasse »Thread« .....	619
15.2.6	Threadpools nutzen .....	627
15.2.7	Die Synchronisation von Threads .....	629
15.2.8	Der »Monitor« zur Synchronisation .....	631
15.2.9	Das Attribut »MethodImpl« .....	637
15.2.10	Das Synchronisationsobjekt »Mutex« .....	638
15.2.11	Grundlagen asynchroner Methodenaufrufe .....	639
15.2.12	Asynchroner Methodenaufruf .....	640
15.2.13	Asynchroner Aufruf mit Rückgabewerten .....	644
15.2.14	Eine Klasse mit asynchronen Methodenaufrufen .....	647
<b>15.3</b>	<b>Die TPL (Task Parallel Library)</b> .....	650
15.3.1	Allgemeines zur Parallelisierung mit der TPL .....	651

15.3.2	Die Klasse »Parallel« .....	651
15.3.3	Die Klasse »Task« .....	657
<b>15.4</b>	<b>Asynchrone Programmierung mit »async« und »await« .....</b>	<b>661</b>
<b>16</b>	<b>Einige wichtige .NET-Klassen .....</b>	<b>665</b>
<b>16.1</b>	<b>Die Klasse »Object« .....</b>	<b>665</b>
16.1.1	Referenzvergleiche mit »Equals« und »ReferenceEquals« .....	666
16.1.2	»ToString« und »GetType« .....	666
16.1.3	Die Methode »MemberwiseClone« und das Problem des Klonens .....	667
<b>16.2</b>	<b>Die Klasse »String« .....</b>	<b>670</b>
16.2.1	Das Erzeugen eines Strings .....	671
16.2.2	Die Eigenschaften von »String« .....	672
16.2.3	Die Methoden der Klasse »String« .....	672
16.2.4	Zusammenfassung der Klasse »String« .....	683
<b>16.3</b>	<b>Die Klasse »StringBuilder« .....</b>	<b>684</b>
16.3.1	Allgemeines .....	684
16.3.2	Die Kapazität eines »StringBuilder«-Objekts .....	685
16.3.3	Die Konstruktoren der Klasse »StringBuilder« .....	686
16.3.4	Die Eigenschaften der Klasse »StringBuilder« .....	686
16.3.5	Die Methoden der Klasse »StringBuilder« .....	687
16.3.6	Allgemeine Anmerkungen .....	689
<b>16.4</b>	<b>Der Typ »DateTime« .....</b>	<b>690</b>
16.4.1	Die Zeitspanne »Tick« .....	690
16.4.2	Die Konstruktoren von »DateTime« .....	691
16.4.3	Die Eigenschaften von »DateTime« .....	692
16.4.4	Die Methoden der Klasse »DateTime« .....	693
<b>16.5</b>	<b>Die Klasse »TimeSpan« .....</b>	<b>694</b>
<b>16.6</b>	<b>Ausgabeformatierung .....</b>	<b>697</b>
16.6.1	Formatierung mit der Methode »String.Format« .....	697
16.6.2	Formatierung mit der Methode »ToString« .....	701
16.6.3	Benutzerdefinierte Formatierung .....	701
<b>17</b>	<b>Projektmanagement und Visual Studio 2012 .....</b>	<b>705</b>
<b>17.1</b>	<b>Der Projekttyp »Klassenbibliothek« .....</b>	<b>705</b>
17.1.1	Mehrere Projekte in einer Projektmappe verwalten .....	706
17.1.2	Die Zugriffsmodifizierer »public« und »internal« .....	707
17.1.3	Friend Assemblys .....	707
17.1.4	Einbinden einer Klassenbibliothek .....	708

<b>17.2</b>	<b>Assemblys</b>	709
17.2.1	Ein Überblick über das Konzept der Assemblys	709
17.2.2	Allgemeine Beschreibung privater und globaler Assemblys	710
17.2.3	Die Struktur einer Assembly	711
17.2.4	Globale Assemblys	716
<b>17.3</b>	<b>Konfigurationsdateien</b>	721
17.3.1	Die verschiedenen Konfigurationsdateien	721
17.3.2	Die Struktur einer Anwendungskonfigurationsdatei	723
17.3.3	Eine Anwendungskonfigurationsdatei mit Visual Studio 2012 bereitstellen	726
17.3.4	Einträge der Anwendungskonfigurationsdatei auswerten	727
17.3.5	Editierbare, anwendungsbezogene Einträge mit <appSettings>	732
<b>17.4</b>	<b>Versionsumleitung in einer Konfigurationsdatei</b>	734
17.4.1	Die Herausgeberrichtliniendatei	735
<b>17.5</b>	<b>XML-Dokumentation</b>	736
17.5.1	Das Prinzip der XML-Dokumentation	737
17.5.2	Die XML-Kommentartags	739
17.5.3	Generieren der XML-Dokumentationsdatei	740
<b>17.6</b>	<b>Der Klassendesigner (Class Designer)</b>	742
17.6.1	Ein typisches Klassendiagramm	742
17.6.2	Hinzufügen und Ansicht von Klassendiagrammen	743
17.6.3	Die Toolbox des Klassendesigners	744
17.6.4	Das Fenster »Klassendetails«	745
17.6.5	Klassendiagramme als Bilder exportieren	747
<b>17.7</b>	<b>Refactoring</b>	747
17.7.1	Methode extrahieren	748
17.7.2	Bezeichner umbenennen	749
17.7.3	Felder inkapseln	750
<b>17.8</b>	<b>Code-Snippets (Codeausschnitte)</b>	750
17.8.1	Codeausschnitte einfügen	751
17.8.2	Die Anatomie eines Codeausschnitts	752
<b>17.9</b>	<b>»ClickOnce«-Verteilung</b>	753
17.9.1	Allgemeine Beschreibung	753
17.9.2	Erstellen einer ClickOnce-Anwendung	754
17.9.3	Die Installation einer ClickOnce-Anwendung	757

---

## **18 Einführung in die WPF und XAML** 759

<b>18.1</b>	<b>Die Merkmale einer WPF-Anwendung</b>	759
18.1.1	Anwendungstypen	761

18.1.2	Eine WPF-Anwendung und deren Dateien .....	762
18.1.3	Ein erstes WPF-Beispiel .....	765
18.1.4	Wichtige WPF-Features .....	768
18.1.5	Der logische und der visuelle Elementbaum .....	770
<b>18.2</b>	<b>XAML (Extended Application Markup Language) .....</b>	<b>773</b>
18.2.1	Die Struktur einer XAML-Datei .....	774
18.2.2	Eigenschaften eines XAML-Elements in Attributschreibweise festlegen ....	776
18.2.3	Eigenschaften im Eigenschaftsfenster festlegen .....	776
18.2.4	Die Eigenschaft-Element-Syntax .....	777
18.2.5	Inhaltseigenschaften .....	778
18.2.6	Typkonvertierung .....	781
18.2.7	Markup-Erweiterungen (Markup Extensions) .....	782
18.2.8	XML-Namespaces .....	785
18.2.9	XAML-Spracherweiterungen .....	787
<b>19</b>	<b>WPF-Layout-Container .....</b>	<b>789</b>
<b>19.1</b>	<b>Die Container-Steuerelemente .....</b>	<b>789</b>
19.1.1	Gemeinsame Eigenschaften der Layout-Container .....	790
19.1.2	Das »Canvas« .....	791
19.1.3	Das »StackPanel« .....	792
19.1.4	Das »WrapPanel« .....	795
19.1.5	Das »DockPanel« .....	796
19.1.6	Das »Grid«-Steuerelement .....	798
19.1.7	Das »UniformGrid« .....	804
<b>19.2</b>	<b>Verschachteln der Layout-Container .....</b>	<b>805</b>
<b>20</b>	<b>Fenster in der WPF .....</b>	<b>809</b>
<b>20.1</b>	<b>Hosts der WPF .....</b>	<b>809</b>
<b>20.2</b>	<b>Fenster vom Typ »Window« .....</b>	<b>810</b>
20.2.1	Mehrere Fenster in einer Anwendung .....	812
<b>20.3</b>	<b>Fenster vom Typ »NavigationWindow« .....</b>	<b>814</b>
20.3.1	Das »Page«-Element .....	816
<b>20.4</b>	<b>Hosts vom Typ »Frame« .....</b>	<b>817</b>
<b>20.5</b>	<b>Navigation zwischen den Seiten .....</b>	<b>818</b>
20.5.1	Navigation mit »HyperLink« .....	819
20.5.2	Der Verlauf der Navigation – das Journal .....	820
20.5.3	Navigation mit »NavigationService« .....	822
20.5.4	Navigation im Internet .....	824

20.5.5	Navigieren mit dem Ereignis »RequestNavigate« des »HyperLink«-Elements .....	825
<b>20.6</b>	<b>Datenübergabe zwischen den Seiten</b> .....	825
20.6.1	Datenübergabe mit der Methode »Navigate« .....	826
<b>20.7</b>	<b>Nachrichtenfenster mit »MessageBox«</b> .....	828
20.7.1	Die Methode »MessageBox.Show« .....	829
<b>21</b>	<b>WPF-Steuerelemente</b> .....	833
<b>21.1</b>	<b>Die Hierarchie der WPF-Komponenten</b> .....	833
<b>21.2</b>	<b>Allgemeine Eigenschaften der WPF-Steuerelemente</b> .....	835
21.2.1	Den Außenrand mit der Eigenschaft »Margin« festlegen .....	835
21.2.2	Den Innenrand mit der Eigenschaft »Padding« festlegen .....	835
21.2.3	Die Eigenschaft »Content« .....	836
21.2.4	Die Größe einer Komponente .....	838
21.2.5	Die Ausrichtung einer Komponente .....	839
21.2.6	Die Sichtbarkeit eines Steuerelements .....	840
21.2.7	Die Farbeinstellungen .....	841
21.2.8	Die Schriften .....	842
<b>21.3</b>	<b>Die unterschiedlichen Schaltflächen</b> .....	842
21.3.1	Die Basisklasse »ButtonBase« .....	843
21.3.2	Das Steuerelement »Button« .....	843
21.3.3	Das Steuerelement »ToggleButton« .....	844
21.3.4	Das Steuerelement »RepeatButton« .....	845
21.3.5	Das Steuerelement »CheckBox« .....	847
21.3.6	Das Steuerelement »RadioButton« .....	847
<b>21.4</b>	<b>Einfache Eingabesteuerelemente</b> .....	848
21.4.1	Das Steuerelement »Label« .....	848
21.4.2	Das Steuerelement »TextBox« .....	849
21.4.3	Das Steuerelement »PasswordBox« .....	852
21.4.4	Das Steuerelement »TextBlock« .....	853
<b>21.5</b>	<b>WPF-Listenelemente</b> .....	856
21.5.1	Das Steuerelement »ListBox« .....	857
21.5.2	Die »ComboBox« .....	860
21.5.3	Das Steuerelement »ListView« .....	861
21.5.4	Das Steuerelement »TreeView« .....	863
21.5.5	Das Steuerelement »TabControl« .....	869
21.5.6	Die Menüleiste .....	870
21.5.7	Das Kontextmenü .....	873



21.5.8	Symbolleisten .....	875
21.5.9	Die Statusleiste .....	878
<b>21.6</b>	<b>Weitere Steuerelemente .....</b>	<b>879</b>
21.6.1	Das Steuerelement »ToolTip« .....	879
21.6.2	Die »Progressbar« .....	881
21.6.3	Das Steuerelement »Slider« .....	881
21.6.4	Das »GroupBox«-Steuerelement .....	882
21.6.5	Das Steuerelement »ScrollView« .....	883
21.6.6	Das Steuerelement »Expander« .....	885
21.6.7	Das Steuerelement »Border« .....	886
21.6.8	Die »Image«-Komponente .....	887
21.6.9	»Calendar« und »DatePicker« zur Datumsangabe .....	889
21.6.10	Das Steuerelement »InkCanvas« .....	890
<b>21.7</b>	<b>Das »Ribbon«-Steuerelement .....</b>	<b>893</b>
21.7.1	Voraussetzungen für den Zugriff auf das »Ribbon«-Control .....	893
21.7.2	Ein kurzer Überblick .....	893
21.7.3	Der XAML-Code .....	894
<b>21.8</b>	<b>FlowDocuments .....</b>	<b>899</b>
21.8.1	Allgemeine Beschreibung eines FlowDocuments .....	899
21.8.2	Eigenschaften eines »FlowDocuments« .....	900
21.8.3	Die Blöcke eines »FlowDocuments« .....	900
21.8.4	Inline-Elemente .....	905
21.8.5	»FlowDocuments« mit Code erzeugen .....	907
21.8.6	Speichern und Laden eines »FlowDocuments« .....	910
<b>21.9</b>	<b>Das Element »FlowDocumentViewer« .....</b>	<b>911</b>
21.9.1	Das Anzeigeelement »FlowDocumentScrollView« .....	911
21.9.2	Das Anzeigeelement »FlowDocumentPageViewer« .....	912
21.9.3	Das Anzeigeelement »FlowDocumentReader« .....	912
<b>21.10</b>	<b>XPS-Dokumente mit »DocumentViewer« .....</b>	<b>913</b>
21.10.1	Allgemeines zum XPS-Format .....	913
21.10.2	Beispielprogramm .....	914
<b>21.11</b>	<b>Das Steuerelement »RichTextBox« .....</b>	<b>915</b>
<b>22</b>	<b>Elementbindungen .....</b>	<b>921</b>
<b>22.1</b>	<b>Einführung in die Bindungstechnik .....</b>	<b>921</b>
22.1.1	Ein einfaches Bindungsbeispiel .....	921
<b>22.2</b>	<b>Die Klasse »Binding« .....</b>	<b>924</b>
22.2.1	Die Bindungsrichtung festlegen .....	925
22.2.2	Aktualisierung der Bindung .....	928

22.2.3	Die Ereignisse »SourceUpdated« und »TargetUpdated« .....	930
22.2.4	Beenden einer Bindung .....	931
<b>22.3</b>	<b>Bindungsalternativen</b> .....	931
22.3.1	Die Eigenschaft »Source« .....	931
22.3.2	Anbindung an relative Datenquellen .....	932
22.3.3	Die Bindung an »DataContext« .....	934
<b>23</b>	<b>Konzepte von WPF</b> .....	935
<b>23.1</b>	<b>Anwendungsspezifische Ressourcen</b> .....	935
<b>23.2</b>	<b>Anwendungsübergreifende Ressourcen</b> .....	937
23.2.1	Mehrere Ressourcenwörterbücher .....	939
23.2.2	Die Suche nach einer Ressource .....	940
<b>23.3</b>	<b>Logische Ressourcen</b> .....	940
23.3.1	Statische Ressourcen .....	941
23.3.2	Dynamische Ressourcen .....	944
23.3.3	Ressourcen mit C#-Code bearbeiten .....	945
23.3.4	Abrufen von Systemressourcen .....	946
<b>23.4</b>	<b>Styles</b> .....	948
23.4.1	Einfache Styles .....	948
23.4.2	Typisierte Styles .....	952
23.4.3	Erweitern von Styles .....	953
23.4.4	EventSetter .....	954
<b>23.5</b>	<b>Trigger</b> .....	956
23.5.1	Eigenschaftstrigger .....	957
23.5.2	Datentrigger .....	960
23.5.3	Ereignistrigger .....	961
<b>23.6</b>	<b>Templates</b> .....	962
23.6.1	Allgemeines zu »ControlTemplates« .....	963
23.6.2	Definition innerhalb eines Styles .....	968
<b>23.7</b>	<b>Ermitteln des visuellen Elementbaums</b> .....	969
23.7.1	Das Tool »Expression Blend« .....	969
23.7.2	Standard-Template mit Code abfragen .....	971
<b>24</b>	<b>Datenbindung</b> .....	975

<b>24.1</b>	<b>Bindung benutzerdefinierter Objekte</b> .....	975
24.1.1	Ein Objekt mit XAML-Code erzeugen und binden .....	976
24.1.2	Ein Objekt mit C#-Code erzeugen und binden .....	977
24.1.3	Aktualisieren benutzerdefinierter Objekte .....	979

<b>24.2</b>	<b>Auflistungen binden</b>	981
24.2.1	Allgemeine Gesichtspunkte	981
24.2.2	Anbindung an eine »ListBox«	982
24.2.3	Änderungen der Collection an die bindenden Elemente weiterleiten	984
<b>24.3</b>	<b>Validieren von Bindungen</b>	987
24.3.1	Die Validierung im Datenobjekt	988
24.3.2	Eine benutzerdefinierte »ValidationRule«	990
24.3.3	Validierung mit der Schnittstelle »IDataErrorInfo«	991
24.3.4	Fehlerhinweise individuell anzeigen	993
24.3.5	Ereignisauslösung bei einem Validierungsfehler	995
<b>24.4</b>	<b>Daten konvertieren</b>	995
24.4.1	Mehrfachbindungen und Konverterklassen	999
<b>24.5</b>	<b>Datenbindung an ADO.NET- und LINQ-Datenquellen</b>	1000
24.5.1	Das Binden an ADO.NET-Objekte	1001
24.5.2	Das Binden an LINQ-Ausdrücke	1002

## **25 Weitere Möglichkeiten der Datenbindung** 1005

---

<b>25.1</b>	<b>»ItemsControl«-Steuerelemente anpassen</b>	1005
25.1.1	Den Style eines »ListBoxItem«-Elements ändern	1006
25.1.2	DataTemplates festlegen	1008
25.1.3	»DataTemplates« mit Trigger	1010
<b>25.2</b>	<b>Alternative Datenbindungen</b>	1014
25.2.1	Die Klasse »ObjectDataProvider«	1014
<b>25.3</b>	<b>Navigieren, Filtern, Sortieren und Gruppieren</b>	1016
25.3.1	Navigieren	1018
25.3.2	Sortieren	1021
25.3.3	Filtern	1022
25.3.4	Gruppieren	1026
<b>25.4</b>	<b>Das Steuerelement »DataGrid«</b>	1030
25.4.1	Elementare Eigenschaften des »DataGrid«	1032
25.4.2	Spalten definieren	1033
25.4.3	Details einer Zeile anzeigen	1039

## **26 Dependency Properties** 1041

---

<b>26.1</b>	<b>Die Charakteristik von Abhängigkeitseigenschaften</b>	1041
<b>26.2</b>	<b>Den Wert einer Abhängigkeitseigenschaft bilden</b>	1042
<b>26.3</b>	<b>Definition einer Dependency Property</b>	1043
26.3.1	Registrieren einer Abhängigkeitseigenschaft	1044

26.3.2	Der Eigenschaftswrapper .....	1045
26.3.3	Die Eigenschaftsmetadaten .....	1046
26.3.4	Freigabe des spezifischen Eigenschaftswertes .....	1050
26.3.5	Vererbung von Abhängigkeitseigenschaften .....	1050
<b>26.4</b>	<b>Validieren einer Abhängigkeitseigenschaft</b> .....	1051
26.4.1	Validieren mit »ValidateValueCallback« .....	1051
26.4.2	Validieren mit »CoerceValueCallback« .....	1052
<b>26.5</b>	<b>Angehängte Eigenschaften (Attached Property)</b> .....	1053
26.5.1	Angehängte Eigenschaften zur Laufzeit ändern .....	1055
<b>27</b>	<b>Ereignisse in der WPF</b> .....	1057
<b>27.1</b>	<b>Ereignishandler bereitstellen</b> .....	1057
<b>27.2</b>	<b>Routing-Strategien</b> .....	1058
27.2.1	Der durchlaufene Elementbaum .....	1060
27.2.2	Beispielanwendung .....	1060
27.2.3	Sonderfall der Mausereignisse .....	1062
<b>27.3</b>	<b>Der Ereignishandler</b> .....	1063
27.3.1	Die Klasse »RoutedEventArgs« .....	1063
27.3.2	Die Quelle des Routing-Prozesses .....	1064
27.3.3	Die Eigenschaft »Handled« .....	1065
27.3.4	Registrieren und Deregistrieren eines Ereignishandlers mit Code .....	1066
<b>27.4</b>	<b>Definition eines Routed Events</b> .....	1066
27.4.1	Ereignisauslösung .....	1068
27.4.2	Das Ereignis als Attached Event verwenden .....	1069
27.4.3	Unterdrückte Ereignisse .....	1070
<b>27.5</b>	<b>Mausereignisse in der WPF</b> .....	1071
27.5.1	Ziehen der Maus .....	1071
27.5.2	Auswerten der Mausklicks .....	1072
27.5.3	Capturing .....	1073
<b>28</b>	<b>WPF-Commands</b> .....	1077
<b>28.1</b>	<b>Allgemeine Beschreibung</b> .....	1077
28.1.1	Ein einführendes Beispiel .....	1077
<b>28.2</b>	<b>Vordefinierte WPF-Commands</b> .....	1079
<b>28.3</b>	<b>Commands verwenden</b> .....	1080
28.3.1	Command-Bindungen einrichten .....	1081
28.3.2	Lokalität der Befehlsbindung .....	1083
28.3.3	Befehlsbindung mit Programmcode .....	1083

28.3.4	Das Befehlsziel mit »CommandTarget« angeben .....	1084
28.3.5	Zusätzliche Daten bereitstellen .....	1085
28.3.6	Befehle mit Maus oder Tastatur aufrufen .....	1086
<b>28.4</b>	<b>Die Anatomie eines »Command«-Objekts</b> .....	<b>1087</b>
28.4.1	Das Interface »ICommand« .....	1088
28.4.2	Die Klassen »RoutedCommand« und »RoutedUICommand« .....	1088
28.4.3	Das Interface »ICommandSource« .....	1090
<b>28.5</b>	<b>Das MVVM-Pattern</b> .....	<b>1091</b>
28.5.1	Ein simples Beispielprogramm .....	1093
<b>29</b>	<b>Benutzerdefinierte Controls</b> .....	<b>1097</b>
<b>29.1</b>	<b>Erstellen eines benutzerdefinierten Steuerelements</b> .....	<b>1097</b>
<b>29.2</b>	<b>Der XAML-Code</b> .....	<b>1099</b>
<b>29.3</b>	<b>Die Programmlogik des Steuerelements</b> .....	<b>1100</b>
29.3.1	Die Eigenschaften .....	1100
29.3.2	Ein Ereignis bereitstellen .....	1102
29.3.3	Das Steuerelement um einen »Command« ergänzen .....	1103
<b>29.4</b>	<b>Testanwendung</b> .....	<b>1104</b>
<b>30</b>	<b>2D-Grafik</b> .....	<b>1107</b>
<b>30.1</b>	<b>Shapes</b> .....	<b>1107</b>
30.1.1	Allgemeine Beschreibung .....	1107
30.1.2	Line-Elemente .....	1108
30.1.3	Ellipse- und Rectangle-Elemente .....	1109
30.1.4	Polygon- und Polyline-Elemente .....	1109
30.1.5	Darstellung der Linien .....	1109
<b>30.2</b>	<b>Path-Elemente</b> .....	<b>1111</b>
30.2.1	GeometryGroup .....	1112
30.2.2	CombinedGeometry .....	1113
30.2.3	PathGeometry .....	1114
<b>30.3</b>	<b>Brush-Objekte</b> .....	<b>1115</b>
30.3.1	SolidColorBrush .....	1116
30.3.2	LinearGradientBrush .....	1117
30.3.3	RadialGradientBrush .....	1119
30.3.4	TileBrush .....	1120
30.3.5	ImageBrush .....	1122
30.3.6	VisualBrush .....	1123
30.3.7	DrawingBrush .....	1125

## **31 ADO.NET – Verbindungsorientierte Objekte** 1127

---

<b>31.1 Allgemeines</b>	1127
<b>31.2 Die Datenprovider</b>	1128
<b>31.3 Die Verbindung zu einer Datenbank herstellen</b>	1129
31.3.1 Das Connection-Objekt	1129
31.3.2 Die Verbindungszeichenfolge	1130
31.3.3 Die Verbindung mit einer SQL Server-Instanz aufbauen	1131
31.3.4 Öffnen und Schließen einer Verbindung	1134
31.3.5 Das Verbindungspooling	1138
31.3.6 Die Ereignisse eines »Connection«-Objekts	1142
31.3.7 Verbindungszeichenfolgen aus einer Konfigurationsdatei abrufen	1144
31.3.8 Verbindungen mit dem OleDb-Datenprovider	1146

## **32 ADO.NET – Das Command-Objekt** 1149

---

<b>32.1 Die Datenbankabfrage</b>	1149
<b>32.2 Das SqlCommand-Objekt</b>	1149
32.2.1 Erzeugen eines SqlCommand-Objekts	1150
32.2.2 Die Methode »CreateCommand« des Connection-Objekts	1151
32.2.3 Ausführen des SqlCommand-Objekts	1151
32.2.4 Die Eigenschaft »CommandTimeout« des SqlCommand-Objekts	1152
<b>32.3 Aktionsabfragen absetzen</b>	1152
32.3.1 Datensätze hinzufügen	1152
32.3.2 Datensätze löschen	1153
32.3.3 Datensätze ändern	1154
32.3.4 Abfragen, die genau ein Ergebnis liefern	1154
<b>32.4 Das SqlDataReader-Objekt</b>	1154
32.4.1 Datensätze einlesen	1155
32.4.2 Schließen des SqlDataReader-Objekts	1157
32.4.3 MARS (Multiple Active Resultsets)	1158
32.4.4 Batchabfragen mit »NextResult« durchlaufen	1159
32.4.5 Das Schema eines SqlDataReader-Objekts untersuchen	1160
<b>32.5 Parametrisierte Abfragen</b>	1162
32.5.1 Parametrisierte Abfragen mit dem SqlClient-Datenprovider	1162
32.5.2 Die Klasse »SqlParameter«	1165
32.5.3 Asynchrone Abfragen	1165
32.5.4 Gespeicherte Prozeduren (Stored Procedures)	1169

<b>33</b>	<b>ADO.NET – Der SqlDataAdapter</b>	<b>1177</b>
<b>33.1</b>	<b>Was ist ein DataAdapter?</b>	<b>1177</b>
<b>33.2</b>	<b>Die Konstruktoren der Klasse DataAdapter</b>	<b>1179</b>
<b>33.3</b>	<b>Arbeiten mit dem SqlDataAdapter</b>	<b>1179</b>
33.3.1	Die Eigenschaft »SelectCommand«	1179
33.3.2	Den lokalen Datenspeicher mit »Fill« füllen	1180
33.3.3	Öffnen und Schließen von Verbindungen	1181
33.3.4	Doppelter Aufruf der Fill-Methode	1182
33.3.5	Mehrere DataAdapter-Objekte aufrufen	1182
33.3.6	Die Spalten- und der Tabellenbezeichner einer DataTable	1183
33.3.7	Paging mit der Fill-Methode	1183
<b>33.4</b>	<b>Tabellenzuordnung mit der Klasse »TableMappings«</b>	<b>1184</b>
33.4.1	Spaltenzuordnungen in einem DataSet	1186
33.4.2	Spaltenzuordnungen einer DataTable	1187
33.4.3	Die Eigenschaft »MissingMappingAction« des DataAdapters	1188
<b>33.5</b>	<b>Das Ereignis »FillError« des SqlDataAdapter</b>	<b>1188</b>
<b>34</b>	<b>ADO.NET – Daten im lokalen Speicher</b>	<b>1191</b>
<b>34.1</b>	<b>Allgemeines</b>	<b>1191</b>
<b>34.2</b>	<b>Verwenden des DataSet-Objekts</b>	<b>1192</b>
34.2.1	Ein DataSet-Objekt erzeugen	1192
34.2.2	Die Anatomie einer DataTable	1192
34.2.3	Der Zugriff auf eine Tabelle im DataSet	1193
34.2.4	Der Zugriff auf die Ergebnisliste	1194
34.2.5	Dateninformationen in eine XML-Datei schreiben	1195
<b>34.3</b>	<b>Gültigkeitsprüfung im DataSet</b>	<b>1196</b>
34.3.1	Dem DataSet Schemainformationen übergeben	1196
34.3.2	Eigenschaften einer DataColumn, die der Gültigkeitsprüfung dienen	1198
34.3.3	Die Constraints-Klassen einer »DataTable«	1199
34.3.4	Das Schema mit Programmcode erzeugen	1200
34.3.5	Schemainformationen mit SqlDataAdapter abrufen	1201
<b>34.4</b>	<b>Änderungen in einer DataTable vornehmen</b>	<b>1204</b>
34.4.1	Editieren einer DataRow	1204
34.4.2	Löschen einer Datenzeile	1206
34.4.3	Eine neue Datenzeile hinzufügen	1206
34.4.4	Der Sonderfall: Autoinkrementspalten	1207
34.4.5	Was bei einer Änderung einer Datenzeile passiert	1209
34.4.6	Manuelles Steuern der Eigenschaft »DataRowState«	1213

<b>34.5</b>	<b>Mit mehreren Tabellen arbeiten</b>	1214
34.5.1	Der Weg über JOIN-Abfragen	1214
34.5.2	Mehrere Tabellen in einem DataSet	1216
34.5.3	Eine DataRelation erzeugen	1216
34.5.4	DataRelations und Einschränkungen	1217
34.5.5	In Beziehung stehende Daten suchen	1219
34.5.6	Ergänzung zum Speichern von Schemainformationen in einer XML-Schemadatei	1221
<b>34.6</b>	<b>Filtern und suchen in einer DataTable</b>	1222
34.6.1	Die Methode »Find«	1222
34.6.2	Die Methode »Select«	1223
<b>34.7</b>	<b>Objekte vom Typ »DataView«</b>	1224
34.7.1	Einen »DataView« erzeugen	1225
34.7.2	Auf die Datenzeilen in einem »DataView« zugreifen	1225
34.7.3	Die Eigenschaft »Sort« und die Methode »Find«	1226
34.7.4	Die Methode »FindRows«	1226
34.7.5	Die Eigenschaft »RowFilter«	1227
34.7.6	Die Eigenschaft »RowStateFilter«	1227
34.7.7	Änderungen an einem »DataView«-Objekt	1227
34.7.8	Aus einem »DataView« eine »DataTable« erzeugen	1229
<b>35</b>	<b>ADO.NET – Aktualisieren der Datenbank</b>	1231
<b>35.1</b>	<b>Aktualisieren mit dem »CommandBuilder«</b>	1231
35.1.1	Die von »SqlCommandBuilder« generierten Aktualisierungsstatements	1233
35.1.2	Konfliktsteuerung in einer Mehrbenutzerumgebung	1233
35.1.3	Die Eigenschaft »ConflictOption« des »SqlCommandBuilders«	1236
35.1.4	Die Eigenschaft »SetAllValues«	1237
<b>35.2</b>	<b>Manuell gesteuerte Aktualisierung</b>	1238
35.2.1	Eigene Aktualisierungslogik	1239
35.2.2	Das Beispielprogramm	1240
<b>35.3</b>	<b>Konfliktanalyse</b>	1242
35.3.1	Den Benutzer über fehlgeschlagene Aktualisierungen informieren	1243
35.3.2	Konfliktverursachende Datenzeilen bei der Datenbank abfragen	1244
<b>35.4</b>	<b>Neue Autoinkrementwerte abrufen</b>	1249
<b>36</b>	<b>Stark typisierte DataSets</b>	1251
<b>36.1</b>	<b>Ein stark typisiertes DataSet erzeugen</b>	1251
36.1.1	Typisierte DataSets mit dem Visual Studio Designer erstellen	1251



36.1.2	Das Kommandozeilentool XSD.exe .....	1254
<b>36.2</b>	<b>Die Anatomie eines typisierten DataSets .....</b>	<b>1255</b>
36.2.1	Die Datenzeilen einer Tabelle ausgeben .....	1255
36.2.2	Datenzeilen hinzufügen .....	1258
36.2.3	Datenzeilen bearbeiten .....	1259
36.2.4	Datenzeilen suchen .....	1259
36.2.5	NULL-Werte im typisierten DataSet .....	1260
36.2.6	Die Daten in einem hierarchischen DataSet .....	1260
<b>36.3</b>	<b>Typisierte DataSets manuell im Designer erzeugen .....</b>	<b>1261</b>
36.3.1	Eine »DataTable« manuell erzeugen .....	1261
36.3.2	Der »DataTable« Spalten hinzufügen .....	1262
36.3.3	Beziehungen zwischen den Tabellen erstellen .....	1262
<b>36.4</b>	<b>Weiter gehende Betrachtungen .....</b>	<b>1264</b>
<b>36.5</b>	<b>Der »TableAdapter« .....</b>	<b>1264</b>
36.5.1	Einen »TableAdapter« mit Visual Studio erzeugen .....	1264
36.5.2	Die Methode »Fill« des »TableAdapters« .....	1269
36.5.3	Die Methode »GetData« .....	1270
36.5.4	Die Methode »Update« .....	1270
36.5.5	Aktualisieren mit den DBDirect-Methoden .....	1270
36.5.6	TableAdapter mit mehreren Abfragen .....	1271
36.5.7	Änderungen an einem »TableAdapter« vornehmen .....	1274
<b>36.6</b>	<b>Fazit: Typisierte oder nicht typisierte DataSets? .....</b>	<b>1275</b>
<b>37</b>	<b>Einführung in das ADO.NET Entity Framework .....</b>	<b>1277</b>
<b>37.1</b>	<b>Kritische Betrachtung von ADO.NET .....</b>	<b>1277</b>
<b>37.2</b>	<b>Ein erstes Entity Data Model (EDM) erstellen .....</b>	<b>1279</b>
<b>37.3</b>	<b>Das Entity Data Model im Designer .....</b>	<b>1283</b>
37.3.1	Die übergeordneten Eigenschaften einer Entität .....	1283
37.3.2	Eigenschaften eines Entitätsobjekts .....	1284
37.3.3	Assoziationen im Entity Data Model .....	1287
37.3.4	Der Kontext der Entitäten .....	1288
<b>37.4</b>	<b>Der Aufbau des Entity Data Models .....</b>	<b>1289</b>
<b>37.5</b>	<b>Die Klassen des Entity Data Models (EDM) .....</b>	<b>1292</b>
37.5.1	Die Entitätsklassen .....	1293
37.5.2	DerObjectContext .....	1296
<b>37.6</b>	<b>Die Architektur des Entity Frameworks .....</b>	<b>1297</b>
37.6.1	Object Services .....	1298
37.6.2	Die Schichten des Entity Frameworks .....	1298

<b>38</b>	<b>Datenabfragen des Entity Data Models (EDM)</b>	<b>1301</b>
<b>38.1</b>	<b>Abfragen mit LINQ to Entities</b>	<b>1302</b>
38.1.1	Allgemeine Begriffe in LINQ	1302
38.1.2	Einfache Abfragen	1302
38.1.3	Navigieren in Abfragen	1309
38.1.4	Aggregatmethoden	1314
38.1.5	Joins in LINQ definieren	1315
38.1.6	In Beziehung stehende Daten laden	1318
<b>38.2</b>	<b>Abfragen mit Entity SQL</b>	<b>1324</b>
38.2.1	Ein erstes Beispiel mit Entity SQL	1324
38.2.2	Die fundamentalen Regeln der Entity-SQL-Syntax	1325
38.2.3	Filtern mit Entity SQL	1326
38.2.4	Parametrisierte Abfragen	1328
<b>38.3</b>	<b>Der EntityClient-Provider</b>	<b>1329</b>
38.3.1	Verbindungen mit »EntityConnection«	1330
38.3.2	Die Klasse »EntityCommand«	1331
<b>38.4</b>	<b>Abfrage-Generator-Methoden (QueryBuilder-Methoden)</b>	<b>1332</b>
<b>38.5</b>	<b>SQL-Direktabfragen</b>	<b>1333</b>
<b>39</b>	<b>Entitätsaktualisierung und Zustandsverwaltung</b>	<b>1335</b>
<b>39.1</b>	<b>Aktualisieren von Entitäten</b>	<b>1335</b>
39.1.1	Entitäten ändern	1335
39.1.2	Hinzufügen neuer Entitäten	1337
39.1.3	Löschen einer Entität	1341
<b>39.2</b>	<b>Der Lebenszyklus einer Entität im Objektkontext</b>	<b>1344</b>
39.2.1	Der Zustand einer Entität	1344
39.2.2	Das Team der Objekte im Überblick	1344
39.2.3	Neue Entitäten im Objektkontext	1345
39.2.4	Die Zustände einer Entität	1347
39.2.5	Zusätzliche Entitäten in den Datencache laden	1349
39.2.6	Die Zustandsverfolgung mit »MergeOption« steuern	1349
<b>39.3</b>	<b>Das »ObjectStateEntry«-Objekt</b>	<b>1352</b>
39.3.1	Die Current- und Originalwerte abrufen	1354
39.3.2	Die Methode »TryGetObjectStateEntry«	1355
39.3.3	Abrufen bestimmter Gruppen	1355
39.3.4	Die Methode »GetModifiedProperties«	1356
<b>39.4</b>	<b>Die Klasse »EntityKey«</b>	<b>1357</b>
39.4.1	Die Methoden »GetObjectByKey« und »TryGetObjectByKey«	1357

<b>39.5</b>	<b>Komplexere Szenarien</b>	1358
39.5.1	Die Methode »ChangeState«	1359
39.5.2	Die Methoden »ApplyCurrentChanges« und »ApplyOriginalChanges«	1360
<b>40</b>	<b>Konflikte behandeln</b>	1363
<b>40.1</b>	<b>Allgemeine Betrachtungen</b>	1363
40.1.1	Das pessimistische Sperren	1364
40.1.2	Das optimistische Sperren	1364
<b>40.2</b>	<b>Konkurrierende Zugriffe mit dem Entity Framework</b>	1365
40.2.1	Das Standardverhalten des Entity Frameworks	1365
40.2.2	Das Aktualisierungsverhalten mit »Fixed« beeinflussen	1366
40.2.3	Auf die Ausnahme »OptimisticConcurrencyException« reagieren	1367
40.2.4	Das »ClientWins«-Szenario	1368
40.2.5	Das »StoreWins«-Szenario	1370
<b>41</b>	<b>Plain Old CLR Objects (POCOs)</b>	1371
<b>41.1</b>	<b>Ein erstes Projekt mit POCO-Klassen</b>	1371
41.1.1	Erstellen einfacher POCO-Klassen	1371
41.1.2	Erstellen des Objektkontextes	1373
<b>41.2</b>	<b>Datenabfrage mit Hilfe der POCOs</b>	1375
41.2.1	In Beziehung stehende Daten laden	1375
<b>41.3</b>	<b>Änderungen verfolgen</b>	1377
41.3.1	Die Methode »DetectChanges«	1377
41.3.2	In Beziehung stehende POCOs aktualisieren	1379
<b>Index</b>		1385