

Inhaltsverzeichnis

Vorwort	35
----------------------	-----------

Teil I: Grundlagen

1 Einstieg in Visual Studio 2008	41
1.1 Die Installation von Visual Studio 2008	41
1.1.1 Überblick über die Produktpalette	41
1.1.2 Anforderungen an Hard- und Software	43
1.1.3 Installation der Express Edition	43
1.1.4 Installation von Visual Studio 2008 Team System	44
1.1.5 Installation der MSDN	46
1.2 Unser allererstes C#-Programm	46
1.2.1 Vorbereitungen	47
1.2.2 Programm schreiben	48
1.2.3 Programm kompilieren und testen	49
1.2.4 Einige Erläuterungen zum Quellcode	50
1.2.5 Konsolenanwendungen sind langweilig	51
1.3 Die Windows-Philosophie	51
1.3.1 Mensch-Rechner-Dialog	51
1.3.2 Objekt- und ereignisorientierte Windows-Programmierung	52
1.3.3 Windows-Programmierung mit Visual Studio 2008	53
1.4 Die Entwicklungsumgebung von Visual Studio 2008	55
1.4.1 Der Startdialog	55
1.4.2 Die wichtigsten Fenster	56
1.5 Microsofts .NET-Technologie	60
1.5.1 Zur Geschichte von .NET	60
1.5.2 .NET-Features und Begriffe	62
1.5.3 Wichtige Neuigkeiten in Visual Studio 2008	70

1.6	Praxisbeispiele	70
1.6.1	Windows-Anwendung für Einsteiger	70
1.6.2	Windows-Anwendung für fortgeschrittene Einsteiger	75
2	Grundlagen der Sprache C#	83
2.1	Grundbegriffe	83
2.1.1	Anweisungen	83
2.1.2	Bezeichner	84
2.1.3	Schlüsselwörter	85
2.1.4	Kommentare	85
2.2	Datentypen, Variablen und Konstanten	86
2.2.1	Fundamentale Typen	86
2.2.2	Benennung von Variablen	87
2.2.3	Deklaration von Variablen	88
2.2.4	Typsuffixe	89
2.2.5	Zeichen und Zeichenketten	89
2.2.6	object-Datentyp	90
2.2.7	Konstanten deklarieren	91
2.2.8	Nullable Types	91
2.2.9	Gültigkeitsbereiche und Sichtbarkeit	93
2.3	Operatoren	93
2.3.1	Arithmetische Operatoren	94
2.3.2	Zuweisungsoperatoren	95
2.3.3	Logische Operatoren	96
2.3.4	Rangfolge der Operatoren	99
2.4	Konvertieren von Datentypen	100
2.4.1	Implizite und explizite Konvertierung	100
2.4.2	Übersicht: Welcher Datentyp passt zu welchem?	101
2.4.3	Konvertieren von bool	102
2.4.4	Konvertieren von string	102
2.4.5	Boxing und Unboxing	104
2.5	Kontrollstrukturen	106
2.5.1	Verzweigungsbefehle	106
2.5.2	Schleifenanweisungen	109
2.6	Benutzerdefinierte Datentypen	112
2.6.1	Enumerationen	112
2.6.2	Strukturen	113

2.7	Nutzerdefinierte Methoden	115
2.7.1	Methoden mit Rückgabewert	116
2.7.2	Methoden ohne Rückgabewert	117
2.7.3	Parameterübergabe mit ref	118
2.7.4	Parameterübergabe mit out	119
2.7.5	Überladene Methoden	120
2.8	Praxisbeispiele	121
2.8.1	Vom PAP zur Konsolenanwendung	121
2.8.2	Ein Konsolen- in ein Windows-Programm verwandeln	123
2.8.3	Schleifenanweisungen verstehen	125
2.8.4	Benutzerdefinierte Methoden überladen	127
3	Objektorientiertes Programmieren	131
3.1	Kleine Einführung in die OOP	131
3.1.1	Zur Entwicklung der objektorientierten Programmierung	131
3.1.2	Grundbegriffe der OOP	133
3.1.3	Sichtbarkeit von Klassen und ihren Mitgliedern	135
3.1.4	Allgemeiner Aufbau einer Klasse	136
3.1.5	Das Erzeugen eines Objekts	137
3.1.6	Einführungsbeispiel	140
3.2	Eigenschaften	144
3.2.1	Eigenschaften mit Zugriffsmethoden kapseln	145
3.2.2	Berechnete Eigenschaften	147
3.2.3	Lese-/Schreibschutz	148
3.2.4	Property-Accessors	149
3.2.5	Statische Felder/Eigenschaften	149
3.2.6	Einfache Eigenschaften automatisch implementieren	152
3.3	Methoden	153
3.3.1	Öffentliche und private Methoden	153
3.3.2	Überladene Methoden	154
3.3.3	Statische Methoden	154
3.4	Ereignisse	156
3.4.1	Ereignis-Delegate hinzufügen	157
3.4.2	Ereignisse verwenden	159
3.5	Arbeiten mit Konstruktor und Destruktor	163
3.5.1	Konstruktor	163
3.5.2	Destruktor und Garbage Collector	165

3.6	Vererbung und Polymorphie	168
3.6.1	Klassendiagramm	168
3.6.2	Method-Overriding	169
3.6.3	Klassen implementieren	170
3.6.4	Implementieren der Objekte	173
3.6.5	Allgemeine Hinweise und Regeln zur Vererbung	175
3.6.6	Polymorphes Verhalten	176
3.6.7	Die Rolle von System.Object	179
3.7	Spezielle Klassen	180
3.7.1	Abstrakte Klassen	180
3.7.2	Versiegelte Klassen	181
3.7.3	Partielle Klassen	182
3.7.4	Statische Klassen	183
3.8	Praxisbeispiele	184
3.8.1	Eigenschaften sinnvoll kapseln	184
3.8.2	Eine statische Klasse anwenden	187
3.8.3	Taschenrechner als fatter Client	189
3.8.4	Abmagerungskur für den fetten Client	194
4	Arrays, Strings und Funktionen	203
4.1	Datenfelder (Arrays)	203
4.1.1	Array deklarieren	203
4.1.2	Array instanziiieren	204
4.1.3	Array initialisieren	204
4.1.4	Zugriff auf Array-Elemente	205
4.1.5	Zugriff mittels Schleife	206
4.1.6	Mehrdimensionale Arrays	207
4.1.7	Zuweisen von Arrays	209
4.1.8	Arrays aus Strukturvariablen	210
4.1.9	Löschen und Umdimensionieren von Arrays	211
4.1.10	Eigenschaften und Methoden der Array-Klasse	212
4.1.11	Übergabe von Arrays	213
4.2	Verarbeiten von Zeichenketten	215
4.2.1	Zuweisen von Strings	215
4.2.2	Instanzeigenschaften und -methoden eines Strings	215
4.2.3	Wichtige Methoden der String-Klasse	217

4.3	Formatieren von Zahlen	219
4.3.1	Anwenden der ToString-Methode	219
4.3.2	Anwenden der Format-Methode	221
4.4	Berechnungen mit vordefinierten Funktionen	223
4.4.1	Mathematik	223
4.4.2	Datums- und Zeitfunktionen	225
4.5	Weitere wichtige Sprachfeatures	228
4.5.1	Operatorenüberladung	228
4.5.2	Namespaces (Namensräume)	229
4.6	Praxisbeispiele	233
4.6.1	Zeichenketten verarbeiten	233
4.6.2	Methodenaufrufe mit Array-Parametern	236
5	Collections, Delegates und LINQ	241
5.1	Collections (Auflistungen)	241
5.1.1	Die Schnittstelle IEnumerable	241
5.1.2	ArrayList	244
5.1.3	Hashtable	245
5.1.4	Indexer	245
5.2	Generics	247
5.2.1	Klassische Vorgehensweise	248
5.2.2	Generics bieten Typsicherheit	249
5.2.3	List-Collection statt ArrayList	250
5.2.4	Vorteile generischer Collections	251
5.2.5	Constraints	251
5.2.6	Generische Methoden	252
5.2.7	Iteratoren	252
5.3	Das Prinzip der Delegates	253
5.3.1	Delegates sind Methodenzeiger	253
5.3.2	Einen Delegate-Typ deklarieren	254
5.3.3	Ein Delegate-Objekt erzeugen	254
5.3.4	Delegates vereinfacht instanziiieren	255
5.3.5	Anonyme Methoden	256
5.3.6	Lambda-Ausdrücke	257
5.4	LINQ for Objects	259
5.4.1	Die LINQ-Architektur	259
5.4.2	Anonyme Typen	261
5.4.3	Erweiterungsmethoden	262

5.4.4	Grundlegendes zur LINQ-Syntax	263
5.4.5	Die Projektionsoperatoren Select und SelectMany	267
5.4.6	Der Restriktionsoperator Where	268
5.4.7	Die Sortierungsoperatoren OrderBy und ThenBy	269
5.4.8	Der Gruppierungsoperator GroupBy	270
5.4.9	Verknüpfen mit Join	272
5.4.10	Aggregat-Operatoren	273
5.4.11	Verzögertes Ausführen von LINQ-Abfragen	275
5.4.12	Konvertierungsmethoden	276
5.5	Praxisbeispiele	276
5.5.1	In einer ArrayList suchen und sortieren	276
5.5.2	In einer generischen Liste suchen und sortieren	279
5.5.3	Delegates, anonyme Methoden und Lambda Expressions	280
6	Dateien und Streams	285
6.1	Operationen mit Verzeichnissen und Dateien	285
6.1.1	Das Datei-System von Windows	285
6.1.2	Klassen für Verzeichnis- und Dateioperationen	286
6.1.3	Verzeichnisse erzeugen und löschen	287
6.1.4	Verzeichnisse verschieben und umbenennen	288
6.1.5	Aktuelles Verzeichnis ermitteln bzw. festlegen	288
6.1.6	Unterverzeichnisse bestimmen	288
6.1.7	Im Verzeichnis enthaltene Dateien ermitteln	289
6.1.8	Alle Laufwerke ermitteln	289
6.1.9	Dateien kopieren, verschieben und umbenennen	290
6.1.10	Dateiattribute feststellen	291
6.1.11	Weitere Eigenschaften und Methoden	293
6.1.12	Anwenden der Path-Klasse	294
6.1.13	Änderungen im Dateisystem überwachen	294
6.1.14	Zugriffsberechtigungen festlegen	295
6.2	Lese- und Schreibzugriff auf Dateien	297
6.2.1	Dateien und Streams	298
6.2.2	Dateiparameter	299
6.2.3	Textdateien	300
6.2.4	Binärdateien	302
6.2.5	Sequenzielle Dateien	303
6.2.6	Dateien verschlüsseln	305
6.2.7	Dateien komprimieren	307

6.3	Dateidialoge	308
6.4	Praxisbeispiele	310
6.4.1	Infos über Verzeichnisse und Dateien gewinnen	310
6.4.2	Die Verzeichnisstruktur in eine TreeView einlesen	314
6.4.3	Daten in einem Objektbaum abspeichern	316
7	XML in Theorie und Praxis	323
7.1	XML – etwas Theorie	323
7.1.1	Übersicht	323
7.1.2	Der XML-Grundaufbau	326
7.1.3	Wohlgeformte Dokumente	327
7.1.4	Processing Instructions (PI)	329
7.1.5	Elemente und Attribute	330
7.1.6	Verwendbare Zeichensätze	331
7.2	XSD-Schemas	333
7.2.1	XSD-Schemas und ADO.NET	334
7.2.2	XSD-Schema in Visual Studio erstellen	336
7.2.3	XML-Datei mit XSD-Schema erzeugen	338
7.2.4	XSD-Schema aus einer XML-Datei erzeugen	339
7.3	Verwendung des DOM unter .NET	339
7.3.1	Übersicht	339
7.3.2	DOM-Integration in C#	340
7.3.3	Laden von Dokumenten	341
7.3.4	Erzeugen von XML-Dokumenten	342
7.3.5	Auslesen von XML-Dateien	344
7.3.6	Direktzugriff auf einzelne Elemente	345
7.3.7	Einfügen von Informationen	346
7.3.8	Suchen in den Baumzweigen	348
7.4	XML-Verarbeitung mit LINQ to XML	351
7.4.1	Die LINQ to XML-API	352
7.4.2	Neue XML-Dokumente erzeugen	353
7.4.3	Laden und Sichern von XML-Dokumenten	355
7.4.4	Navigieren in XML-Daten	356
7.4.5	Auswählen und Filtern	358
7.4.6	Manipulieren der XML-Daten	358
7.4.7	XML-Dokumente transformieren	360
7.5	Weitere Möglichkeiten der XML-Verarbeitung	363
7.5.1	Die relationale Sicht auf XML-Daten mit XmlDataDocument	363

7.5.2	XML-Daten aus Objektstrukturen erzeugen	366
7.5.3	Schnelles Suchen in XML-Daten mit XPathNavigator	369
7.5.4	Schnelles Auslesen von XML-Daten mit dem XmlReader	371
7.5.5	Erzeugen von XML-Daten mit XmlWriter	373
7.5.6	XML transformieren mit XSLT	376
7.6	Praxisbeispiele	377
7.6.1	Mit dem DOM in XML-Dokumenten navigieren	377
7.6.2	XML-Daten in eine TreeView einlesen	381
8	ADO.NET-Einführung	385
8.1	Eine kleine Übersicht	385
8.1.1	Die ADO.NET-Klassenhierarchie	385
8.1.2	Die Klassen der Datenprovider	386
8.1.3	Das Zusammenspiel der ADO.NET-Klassen	389
8.2	Das Connection-Objekt	390
8.2.1	Allgemeiner Aufbau	390
8.2.2	OleDbConnection	390
8.2.3	Schließen einer Verbindung	391
8.2.4	Eigenschaften des Connection-Objekts	392
8.2.5	Methoden des Connection-Objekts	394
8.2.6	Der ConnectionStringBuilder	395
8.3	Das Command-Objekt	395
8.3.1	Erzeugen und Anwenden eines Command-Objekts	395
8.3.2	Erzeugen mittels CreateCommand-Methode	396
8.3.3	Eigenschaften des Command-Objekts	396
8.3.4	Methoden des Command-Objekts	398
8.4	Parameter-Objekte	400
8.4.1	Erzeugen und Anwenden eines Parameter-Objekts	400
8.4.2	Eigenschaften des Parameter-Objekts	400
8.5	Das CommandBuilder-Objekt	401
8.5.1	Erzeugen	401
8.5.2	Anwenden	402
8.6	Das DataReader-Objekt	402
8.6.1	DataReader erzeugen	403
8.6.2	Daten lesen	403
8.6.3	Eigenschaften DataReaders	404
8.6.4	Methoden des DataReaders	404
8.7	Das DataAdapter-Objekt	405

8.7.1	DataAdapter erzeugen	405
8.7.2	Command-Eigenschaften	406
8.7.3	Fill-Methode	407
8.7.4	Update-Methode	408
8.8	Das DataSet	409
8.8.1	Das Objektmodell	409
8.8.2	Methoden zum Erzeugen eines DataSets	411
8.9	Das DataTable-Objekt	412
8.9.1	DataTable erzeugen	412
8.9.2	Spalten hinzufügen	413
8.9.3	Zeilen zur DataTable hinzufügen	413
8.9.4	Auf den Inhalt einer DataTable zugreifen	414
8.10	Datenansichten mit DataView	416
8.10.1	Erzeugen eines DataView	416
8.10.2	Sortieren und Filtern von Datensätzen	417
8.10.3	Suchen von Datensätzen	417
8.11	Typisierte DataSets	418
8.11.1	Was ist ein typisiertes DataSet?	418
8.11.2	Das Konzept der Datenquellen	419
8.11.3	Typisierte DataSets und TableAdapter	420
8.12	Datenbindung	422
8.12.1	Prinzipielle Möglichkeiten der Datenbindung	422
8.12.2	Manuelle Datenbindung an einfache Datenfelder	422
8.12.3	Manuelle Datenbindung an Listen und Tabelleninhalte	424
8.12.4	Entwurfszeit-Datenbindung an ein typisiertes DataSet	425
8.12.5	Drag & Drop-Datenbindung	426
8.12.6	Navigieren im DataSet	426
8.12.7	Die Anzeige formatieren	428
8.13	Praxisbeispiele	429
8.13.1	Wichtige ADO.NET-Objekte im Einsatz	429
8.13.2	Eine Aktionsabfrage ausführen	430
8.13.3	Eine Auswahlabfrage aufrufen	433
8.13.4	Die Datenbank aktualisieren	435
8.13.5	Master-Detailbeziehungen im DataGrid anzeigen	438
8.13.6	Im DataView sortieren und filtern	440
8.13.7	Suche nach Datensätzen	442
8.13.8	Einrichten und verwenden einer Datenquelle	444
8.13.9	Eine LINQ to SQL-Abfrage realisieren	448

9	OOP-Spezial	453
9.1	Eine kleine Einführung in die UML	453
9.1.1	Use Case-Diagramm	453
9.1.2	Use Case-Dokumentation	455
9.1.3	Objekte identifizieren	456
9.1.4	Statisches Modell	457
9.1.5	Beziehungen zwischen den Klassen	458
9.1.6	Dynamisches Modell	458
9.1.7	Implementierung	459
9.1.8	Test-Client	463
9.2	Der Klassen-Designer	466
9.2.1	Ein neues Klassendiagramm erzeugen	467
9.2.2	Toolbox	468
9.2.3	Enumeration	469
9.2.4	Klasse	471
9.2.5	Struktur	473
9.2.6	Abstrakte Klasse	473
9.2.7	Schnittstelle	475
9.2.8	Delegate	477
9.2.9	Zuordnung	479
9.2.10	Vererbung	480
9.2.11	Diagramme anpassen	480
9.2.12	Objekt-Testcenter	481
9.2.13	Wann lohnt sich der Einsatz des Klassen-Designers?	483
9.3	Das Microsoft Event Pattern	483
9.3.1	Was sind Design Pattern und wozu braucht man sie?	483
9.3.2	Aufbau und Bedeutung des Observer Pattern	484
9.3.3	Implementierung mit Interfaces und Callbacks	489
9.3.4	Implementierung mit Delegates und Events	498
9.3.5	Implementierung des Microsoft Event Pattern	503
9.3.6	Test und Vergleich	507
9.3.7	Klassendiagramm	507
9.3.8	Schritte zur Implementierung eines Event Pattern	508
9.4	Praxisbeispiele	509
9.4.1	Implementierung einer Finite State Machine	509
9.4.2	Anwendung des Klassen-Designers	514
9.4.3	Subjekt und Observer beobachten sich gegenseitig	527

10	Programmiertechniken	539
10.1	Die Projekteigenschaften einstellen	539
10.1.1	Anwendung	540
10.1.2	Erstellen	541
10.1.3	Debuggen	541
10.1.4	Ressourcen	542
10.1.5	Einstellungen	542
10.1.6	Signierung	543
10.1.7	Sicherheit	543
10.1.8	Veröffentlichen	543
10.2	Fehlersuche	543
10.2.1	Der Debugger	543
10.2.2	Single-Step	547
10.2.3	Procedure-Step	548
10.2.4	Breakpoints	548
10.2.5	Debugging am Beispiel	549
10.2.6	Das Debug-Objekt	553
10.3	Fehlerbehandlung	553
10.3.1	Anweisungen zur Fehlerbehandlung	554
10.3.2	try-catch	554
10.3.3	try-finally	559
10.3.4	Das Standardverhalten bei Ausnahmen festlegen	561
10.3.5	Die Exception-Klasse	562
10.3.6	Fehler/Ausnahmen auslösen	563
10.3.7	Eigene Fehlerklassen	563
10.3.8	Exceptionhandling zur Entwurfszeit	565
10.4	Der Code-Editor kann mehr	566
10.4.1	Refactoring	566
10.4.2	Surrounding	567
10.4.3	Automatisches Umbenennen von Formularen	567
10.5	Zugriff auf die Zwischenablage	568
10.5.1	Das Clipboard-Objekt	568
10.5.2	Zwischenablage-Funktionen für Textboxen	569
10.6	Arbeiten mit der Registry	570
10.6.1	Allgemeines	570
10.6.2	Registry-Unterstützung in .NET	572

10.7	.NET-Reflection	573
10.7.1	Übersicht	573
10.7.2	Assembly laden	574
10.7.3	Mit Type Informationen sammeln	574
10.7.4	Dynamisches Laden von Assemblies	576
10.8	Ressourcen	579
10.8.1	Erstellen von Manifestressourcen	580
10.8.2	Zugriff auf Manifestressourcen	581
10.8.3	Resources-Dateien	583
10.8.4	Was sind .resx-Dateien?	585
10.8.5	Streng typisierte Ressourcen	585
10.9	Lokalisierte Anwendungen	587
10.9.1	Localizable und Language	587
10.9.2	Anwendungsbeispiel	587
10.9.3	Einstellen der aktuellen Kultur zur Laufzeit	590
10.9.4	Bemerkungen zur Lokalisierung	591
10.10	Der Einsatz des SerialPort-Controls	592
10.10.1	Übersicht	592
10.10.2	Einführungsbeispiele	593
10.10.3	Thread-Probleme bei Windows Forms Anwendungen	596
10.10.4	Ein einfaches Terminalprogramm	599
10.11	ClickOnce-Deployment	604
10.11.1	Ort der Veröffentlichung	604
10.11.2	Anwendungsdateien	605
10.11.3	Erforderliche Komponenten	605
10.11.4	Updates	606
10.11.5	Veröffentlichen	606
10.11.6	Verzeichnisstruktur	607
10.11.7	Webpublishing-Assistent	609
10.11.8	Neue Versionen erstellen	609
10.12	Distribution von Anwendungen	610
10.12.1	Das Setup-Projekt	610
10.12.2	Dateisystem-Editor	612
10.12.3	Erster Test	613
10.12.4	Registrierungs-Editor	614
10.12.5	Dateityp-Editor	615
10.12.6	Benutzeroberflächen-Editor	616

10.12.7	Editor für Startbedingungen	617
10.12.8	Test	619
10.13	Praxisbeispiele	619
10.13.1	Zugriff auf die Registry	619
10.13.2	Dateiverknüpfungen erzeugen	621
10.13.3	Betrachter für Manifestressourcen	623
10.13.4	Ressourcen mit Reflection auslesen	626

Teil II: Windows Forms

11	Windows Forms-Anwendungen	631
11.1	Allgemeine Features der visuellen Komponenten	631
11.1.1	Das Eigenschaftsfenster	631
11.1.2	Allgemeine Eigenschaften	632
11.1.3	Allgemeine Ereignisse	636
11.1.4	Allgemeine Methoden	643
11.2	Windows-Formulare	643
11.2.1	Übersicht	644
11.2.2	Wichtige Eigenschaften des Form-Objekts	644
11.2.3	Wichtige Ereignisse des Form-Objekts	646
11.2.4	Wichtige Methoden des Form-Objekts	648
11.2.5	Startformular und Main-Methode	649
11.2.6	Fenster aufrufen	650
11.2.7	Tabulatorreihenfolge festlegen	653
11.2.8	Ausrichten und Platzieren von Komponenten im Formular	654
11.2.9	Spezielle Panels für flexibles Layout	656
11.2.10	Menüs erzeugen	658
11.2.11	Mehr Transparenz bitte	662
11.2.12	Von echten und falschen MDI-Fenstern	662
11.3	Die Application-Klasse	667
11.3.1	Wichtige Eigenschaften und Methoden von Application	667
11.3.2	Die Datei AssemblyInfo.cs	669
11.4	Praxisbeispiele	670
11.4.1	Informationsaustausch zwischen Formularen	670
11.4.2	Ereigniskette beim Laden/Entladen eines Formulars	677

12 Windows Forms-Steuerelemente	683
12.1 Allgemeine Hinweise	683
12.1.1 Hinzufügen von Steuerelementen	683
12.1.2 Steuerelemente selbst per Code erzeugen	684
12.2 Allgemeine Steuerelemente	686
12.2.1 Label	686
12.2.2 LinkLabel	687
12.2.3 Button	688
12.2.4 TextBox	689
12.2.5 MaskedTextBox	692
12.2.6 CheckBox	693
12.2.7 RadioButton	695
12.2.8 ListBox	695
12.2.9 CheckedListBox	697
12.2.10 ComboBox	697
12.2.11 PictureBox	698
12.2.12 DateTimePicker	698
12.2.13 MonthCalendar	699
12.2.14 HScrollBar, VScrollBar	700
12.2.15 TrackBar	701
12.2.16 NumericUpDown	701
12.2.17 DomainUpDown	702
12.2.18 ProgressBar	702
12.2.19 RichTextBox	703
12.2.20 ListView	704
12.2.21 TreeView	710
12.2.22 WebBrowser	715
12.3 Container	716
12.3.1 FlowLayout/TableLayout/SplitContainer	716
12.3.2 Panel	716
12.3.3 GroupBox	717
12.3.4 TabControl	717
12.3.5 ImageList	719
12.4 Menüs & Symbolleisten	720
12.4.1 MenuStrip und ContextMenuStrip	720
12.4.2 ToolStrip	721
12.4.3 StatusStrip	721
12.4.4 ToolStripContainer	721

12.5	Daten	722
12.5.1	DataSet	722
12.5.2	DataGridView/DataGrid	722
12.5.3	BindingNavigator/BindingSource	723
12.6	Komponenten	723
12.6.1	ErrorProvider	723
12.6.2	HelpProvider	723
12.6.3	ToolTip	723
12.6.4	BackgroundWorker	723
12.6.5	Timer	724
12.6.6	SerialPort	724
12.7	Drucken	724
12.7.1	PrintPreviewControl	724
12.7.2	PrintDocument	724
12.8	Dialoge	725
12.8.1	OpenFileDialog/SaveFileDialog/FolderBrowserDialog	725
12.8.2	FontDialog/ColorDialog	725
12.9	Praxisbeispiele	725
12.9.1	Mit der CheckBox arbeiten	725
12.9.2	Steuerelemente per Code selbst erzeugen	727
12.9.3	Controls-Auflistung des Formulars im TreeView anzeigen	729
13	Grundlagen der Grafikausgabe	733
13.1	Übersicht und erste Schritte	733
13.1.1	Warum GDI+?	733
13.1.2	Das Grundkonzept von GDI+	735
13.1.3	Namespaces für die Grafikausgabe	736
13.2	Darstellen von Grafiken	737
13.2.1	Die PictureBox-Komponente	737
13.2.2	Das Image-Objekt	739
13.2.3	Laden von Grafiken zur Laufzeit	739
13.2.4	Sichern von Grafiken	740
13.2.5	Grafikeigenschaften ermitteln	741
13.2.6	Erzeugen von Vorschaugrafiken (Thumbnails)	741
13.2.7	Die Methode RotateFlip	743
13.2.8	Skalieren von Grafiken	744
13.3	Das .NET-Koordinatensystem	745
13.3.1	Globale Koordinaten	745

13.3.2	Seitenkoordinaten (globale Transformation)	746
13.3.3	Gerätekoordinaten (Seitentransformation)	749
13.4	Grundlegende Zeichenfunktionen von GDI+	750
13.4.1	Das zentrale Graphics-Objekt	750
13.4.2	Punkte zeichnen/abfragen	751
13.4.3	Linien	752
13.4.4	Kantenglättung mit Antialiasing	753
13.4.5	PolyLine	754
13.4.6	Rechtecke	754
13.4.7	Polygone	756
13.4.8	Splines	757
13.4.9	Bézierkurven	758
13.4.10	Kreise und Ellipsen	759
13.4.11	Tortenstück (Segment)	759
13.4.12	Bogenstück	761
13.4.13	Wo sind die Rechtecke mit den "runden Ecken"?	762
13.4.14	Textausgabe	763
13.4.15	Ausgabe von Grafiken	767
13.5	Unser Werkzeugkasten	768
13.5.1	Einfache Objekte	768
13.5.2	Vordefinierte Objekte	769
13.5.3	Farben/Transparenz	771
13.5.4	Stifte (Pen)	773
13.5.5	Pinself (Brush)	776
13.5.6	SolidBrush	776
13.5.7	HatchBrush	777
13.5.8	TextureBrush	778
13.5.9	LinearGradientBrush	779
13.5.10	PathGradientBrush	780
13.5.11	Fonts	781
13.5.12	Path-Objekt	782
13.5.13	Clipping/Region	785
13.6	Standarddialoge	788
13.6.1	Schriftauswahl	788
13.6.2	Farbauswahl	790
13.7	Praxisbeispiele	791
13.7.1	Ein Graphics-Objekt erzeugen	791
13.7.2	Zeichenoperationen mit der Maus realisieren	794

14	Erweiterte Grafikprogrammierung	799
14.1	Transformieren mit der Matrix-Klasse	799
14.1.1	Übersicht	799
14.1.2	Translation	800
14.1.3	Skalierung	800
14.1.4	Rotation	801
14.1.5	Scherung	801
14.1.6	Zuweisen der Matrix	802
14.2	Low-Level-Grafikmanipulationen	802
14.2.1	Worauf zeigt Scan0?	803
14.2.2	Anzahl der Spalten bestimmen	804
14.2.3	Anzahl der Zeilen bestimmen	805
14.2.4	Zugriff im Detail (erster Versuch)	805
14.2.5	Zugriff im Detail (zweiter Versuch)	807
14.2.6	Invertieren	809
14.2.7	In Graustufen umwandeln	810
14.2.8	Heller/Dunkler	811
14.2.9	Kontrast	812
14.2.10	Gamma	814
14.2.11	Histogramm spreizen	814
14.2.12	Ein universeller Grafikfilter	817
14.3	Fortgeschrittene Techniken	821
14.3.1	Flackerfrei dank Double Buffering	821
14.3.2	Animationen	822
14.3.3	Animated GIFs	826
14.3.4	Auf einzelne GIF-Frames zugreifen	828
14.3.5	Transparenz realisieren	830
14.3.6	Eine Grafik maskieren	831
14.3.7	JPEG-Qualität beim Sichern bestimmen	833
14.4	Grundlagen der 3-D-Vektorgrafik	834
14.4.1	Datentypen für die Verwaltung	834
14.4.2	Eine universelle 3D-Grafik-Klasse	835
14.4.3	Grundlegende Betrachtungen	836
14.4.4	Translation	839
14.4.5	Streckung/Skalierung	840
14.4.6	Rotation	841
14.4.7	Die eigentlichen Zeichenroutinen	843

14.5	Und doch wieder GDI-Funktionen ...	845
14.5.1	Am Anfang war das Handle ...	845
14.5.2	Gerätekontext (Device Context Types) ...	847
14.5.3	Koordinatensysteme und Abbildungsmodi ...	850
14.5.4	Zeichenwerkzeuge/Objekte ...	854
14.5.5	Bitmaps ...	856
14.6	Praxisbeispiele ...	859
14.6.1	Die Transformationsmatrix verstehen ...	859
14.6.2	Eine 3D-Grafikausgabe in Aktion ...	862
14.6.3	Einen Fenster-Screenshot erzeugen ...	866
15	Druckausgabe	869
15.1	Einstieg und Übersicht	869
15.1.1	Nichts geht über ein Beispiel	869
15.1.2	Programmiermodell	871
15.1.3	Kurzübersicht der Objekte	872
15.2	Auswerten der aktuellen Druckereinstellungen	872
15.2.1	Die vorhandenen Drucker	872
15.2.2	Der Standarddrucker	873
15.2.3	Verfügbare Papierformate/Seitenabmessungen	873
15.2.4	Der eigentliche Druckbereich	875
15.2.5	Die Seitenausrichtung ermitteln	876
15.2.6	Ermitteln der Farbfähigkeit	876
15.2.7	Die Druckauflösung abfragen	876
15.2.8	Ist beidseitiger Druck möglich?	877
15.2.9	Einen "Informationsgerätekontext" erzeugen	877
15.2.10	Abfragen von Werten während des Drucks	878
15.3	Festlegen von Druckereinstellungen	879
15.3.1	Einen Drucker auswählen	879
15.3.2	Drucken in Millimetern	879
15.3.3	Festlegen der Seitenränder	880
15.3.4	Druckjobname	881
15.3.5	Anzahl der Kopien	881
15.3.6	Beidseitiger Druck	882
15.3.7	Seitenzahlen festlegen	883
15.3.8	Druckqualität verändern	886
15.4	Die Druckdialoge verwenden	886
15.4.1	PrintDialog	886

15.4.2	PageSetupDialog	887
15.4.3	PrintPreviewDialog	889
15.4.4	Ein eigenes Druckvorschau-Fenster realisieren	890
15.5	Drucken mit OLE-Automation	891
15.5.1	Kurzeinstieg in die OLE-Automation	892
15.5.2	Drucken mit Microsoft Word	894
15.6	Praxisbeispiele	896
15.6.1	Den Drucker umfassend konfigurieren	896
15.6.2	Drucken mit Word	905

16 Asynchrone Programmierung 909

16.1	Programmieren mit Threads	909
16.1.1	Einführung Threading	909
16.1.2	Einführungsbeispiel	910
16.1.3	Wichtige Thread-Methoden	912
16.1.4	Wichtige Thread-Eigenschaften	913
16.1.5	Verwendung der ThreadPool-Klasse	914
16.1.6	Übersicht Thread-Locking	916
16.1.7	Threading ohne lock	916
16.1.8	Threading mit lock	917
16.1.9	Die Monitor-Klasse	918
16.1.10	Mutex	919
16.1.11	Interaktion mit der Programmoberfläche	921
16.2	Timer-Threads	923
16.3	Verwendung der BackgroundWorker-Komponente	924
16.4	Asynchrone Programmierentwurfsmuster verwenden	926
16.4.1	Kurzübersicht	927
16.4.2	Polling	928
16.4.3	Callback verwenden	929
16.4.4	Callback mit Parameterübergabe verwenden	930
16.4.5	Callback mit Zugriff auf die Programm-Oberfläche	931
16.5	Asynchroner Aufruf beliebiger Methoden	933
16.5.1	Die Beispielklasse	933
16.5.2	Asynchroner Aufruf ohne Callback	934
16.5.3	Asynchroner Aufruf mit Callback und Anzeigefunktion	935
16.5.4	Aufruf mit Rückgabewerten (per Eigenschaft)	936
16.5.5	Aufruf mit Rückgabewerten (per EndInvoke)	936

16.6	Praxisbeispiele	937
16.6.1	Spieltrieb & Multithreading erleben	937
16.6.2	Prozess- und Thread-Informationen gewinnen	950
16.6.3	Ein externes Programm starten	956
17	Komponentenentwicklung	959
17.1	Überblick	959
17.2	Benutzersteuerelement	960
17.2.1	Entwickeln einer Auswahl-ListBox	960
17.2.2	Komponente verwenden	962
17.3	Benutzerdefiniertes Steuerelement	963
17.3.1	Entwickeln eines BlinkLabels	963
17.3.2	Verwenden der Komponente	966
17.4	Komponentenklasse	966
17.5	Eigenschaften	967
17.5.1	Einfache Eigenschaften	967
17.5.2	Schreib-/Lesezugriff (Get/Set)	968
17.5.3	Nur Lese-Eigenschaft (ReadOnly)	968
17.5.4	Nur-Schreibzugriff (WriteOnly)	969
17.5.5	Hinzufügen von Beschreibungen	969
17.5.6	Ausblenden im Eigenschaftenfenster	969
17.5.7	Einfügen in Kategorien	970
17.5.8	Default-Wert einstellen	970
17.5.9	Standard-Eigenschaft (Indexer)	971
17.5.10	Wertebereichsbeschränkung und Fehlerprüfung	971
17.5.11	Eigenschaften von Aufzählungstypen	973
17.5.12	Standard Objekt-Eigenschaften	974
17.5.13	Eigene Objekt-Eigenschaften	974
17.6	Methoden	976
17.6.1	Konstruktor	976
17.6.2	Class-Konstruktor	978
17.6.3	Destruktor	979
17.6.4	Aufruf des Basisklassen-Konstruktors	979
17.6.5	Aufruf von Basisklassen-Methoden	979
17.7	Ereignisse (Events)	980
17.7.1	Ereignis mit Standardargument definieren	980
17.7.2	Ereignis mit eigenen Argumenten	981
17.7.3	Ein Default-Ereignis festlegen	983

17.7.4	Mit Ereignissen auf Windows-Messages reagieren	983
17.8	Weitere Themen	985
17.8.1	Wohin mit der Komponente?	985
17.8.2	Assembly-Informationen festlegen	986
17.8.3	Assemblies signieren	988
17.8.4	Komponenten-Ressourcen einbetten	989
17.8.5	Der Komponente ein Icon zuordnen	989
17.8.6	Den Designmodus erkennen	990
17.8.7	Komponenten lizenzieren	990
17.9	Praxisbeispiele	995
17.9.1	AnimGif – Komponente für die Anzeige von Animationen	995
17.9.2	Eine FontComboBox entwickeln	997
17.9.3	Das PropertyGrid verwenden	999

Teil III: WPF-Anwendungen

18	WPF-Einführung	1005
18.1	Einführung	1005
18.1.1	Was kann eine WPF-Anwendung?	1005
18.1.2	Die eXtensible Application Markup Language	1007
18.1.3	Verbinden von XAML und C#-Code	1012
18.1.4	Zielpattformen	1017
18.1.5	Applikationstypen	1018
18.1.6	Vorteile und Nachteile von WPF-Anwendungen	1019
18.1.7	Weitere Dateien im Überblick	1019
18.2	Alles beginnt mit dem Layout	1022
18.2.1	Allgemeines zum Layout	1022
18.2.2	Positionieren von Steuerelementen	1024
18.2.3	Canvas	1027
18.2.4	StackPanel	1028
18.2.5	DockPanel	1030
18.2.6	WrapPanel	1032
18.2.7	UniformGrid	1032
18.2.8	Grid	1033
18.2.9	ViewBox	1038
18.2.10	TextBlock	1039

18.3	Das WPF-Programm	1042
18.3.1	Die App-Klasse	1043
18.3.2	Das Startobjekt festlegen	1043
18.3.3	Kommandozeilenparameter verarbeiten	1044
18.3.4	Die Anwendung beenden	1044
18.3.5	Auswerten von Anwendungsereignissen	1045
18.3.6	Die Window-Klasse	1046
19	Übersicht WPF-Controls	1049
19.1	Allgemeingültige Eigenschaften	1049
19.2	Label	1051
19.3	Button, RepeatButton, ToggleButton	1051
19.3.1	Schaltflächen für modale Dialoge	1052
19.3.2	Schaltflächen mit Grafik	1053
19.4	TextBox, PasswortBox	1054
19.4.1	TextBox	1054
19.4.2	PasswordBox	1056
19.5	CheckBox	1056
19.6	RadioButton	1058
19.7	ListBox, ComboBox	1059
19.7.1	ListBox	1059
19.7.2	ComboBox	1062
19.8	Image	1064
19.8.1	Grafik per XAML zuweisen	1064
19.8.2	Grafik zur Laufzeit zuweisen	1064
19.8.3	Bild aus Datei laden	1065
19.8.4	Die Grafikskalierung beeinflussen	1066
19.9	MediaElement	1067
19.10	Slider, ScrollBar	1070
19.10.1	Slider	1070
19.10.2	ScrollBar	1071
19.11	ScrollViewer	1072
19.12	Menu, ContextMenu	1073
19.12.1	Menu	1073
19.12.2	Tastenkürzel	1074
19.12.3	Grafiken	1075
19.12.4	Weitere Möglichkeiten	1076
19.12.5	ContextMenu	1077

19.13	ToolBar	1077
19.14	StatusBar, ProgressBar	1081
19.14.1	StatusBar	1081
19.14.2	ProgressBar	1083
19.15	Border, GroupBox, BulletDecorator	1084
19.15.1	Border	1084
19.15.2	GroupBox	1085
19.15.3	BulletDecorator	1086
19.16	RichTextBox	1088
19.16.1	Verwendung und Anzeige von vordefiniertem Text	1088
19.16.2	Neues Dokument zur Laufzeit erzeugen	1089
19.16.3	Sichern von Dokumenten	1090
19.16.4	Laden von Dokumenten	1092
19.16.5	Texte per Code einfügen/modifizieren	1092
19.16.6	Texte formatieren	1093
19.16.7	EditingCommands	1095
19.16.8	Grafiken/Objekte einfügen	1096
19.16.9	Rechtschreibkontrolle	1097
19.17	FlowDocumentPageViewer, -Reader, -ScrollViewer	1097
19.17.1	FlowDocumentPageViewer	1098
19.17.2	FlowDocumentReader	1098
19.17.3	FlowDocumentScrollViewer	1099
19.18	FlowDocument	1099
19.19	Expander, TabControl	1101
19.19.1	Expander	1101
19.19.2	TabControl	1103
19.20	Popup	1104
19.21	TreeView	1106
19.22	ListView	1109
19.23	DataGridView	1110
19.24	InkCanvas	1111
19.24.1	Stift-Parameter definieren	1111
19.24.2	Die Zeichenmodi	1112
19.24.3	Inhalte laden und sichern	1113
19.24.4	Konvertieren in eine Bitmap	1113
19.24.5	Weitere Eigenschaften	1114

19.25	Ellipse, Rectangle, Line und Co.	1115
19.25.1	Ellipse	1115
19.25.2	Rectangle	1115
19.25.3	Line	1116
20	WPF-Entwicklung	1117
20.1	Wichtige Konzepte	1117
20.1.1	Abhängige Eigenschaften (Dependency Properties)	1117
20.1.2	Angehängte Eigenschaften (Attached Properties)	1118
20.1.3	Einsatz von Ressourcen	1119
20.1.4	Das WPF-Ereignis-Modell	1124
20.1.5	Verwendung von Commands	1127
20.2	Das WPF-Style-System	1133
20.2.1	Übersicht	1133
20.2.2	Benannte Styles	1133
20.2.3	Typ-Styles	1135
20.2.4	Styles anpassen und vererben	1136
20.2.5	Verwenden von Triggern	1138
20.2.6	Einsatz von Templates	1142
20.2.7	Storyboard	1148
20.3	Datenbindung	1149
20.3.1	Grundprinzip	1150
20.3.2	Bindungsarten	1151
20.3.3	Binden an Objekte	1153
20.3.4	Binden von Collections	1158
20.3.5	Anzeige von Datenbankinhalten	1163
20.3.6	Formatieren von Werten	1167
20.4	Entwicklungstools für WPF-Anwendungen	1170
20.4.1	Microsoft Visual Studio 2008	1170
20.4.2	Microsoft Expression Blend	1171
20.4.3	Weitere Tools	1174

Teil IV: ASP.NET-Anwendungen

21 Einführung in ASP.NET	1179
21.1 ASP.NET für Ein- und Umsteiger	1179
21.1.1 ASP – der Blick zurück	1179
21.1.2 Was ist bei ASP.NET anders?	1180
21.1.3 Was gibt es noch in ASP.NET?	1182
21.1.4 Vorteile von ASP.NET gegenüber ASP	1183
21.1.5 Voraussetzungen für den Einsatz von ASP.NET	1184
21.1.6 Und was hat das alles mit C# zu tun?	1184
21.2 Unsere erste Web-Anwendung	1187
21.2.1 Visueller Entwurf der Bedienoberfläche	1187
21.2.2 Zuweisen der Objekteigenschaften	1190
21.2.3 Verknüpfen der Objekte mit Ereignissen	1191
21.2.4 Programm kompilieren und testen	1192
21.3 Die ASP.NET-Projektdateien	1193
21.3.1 ASPX-Datei(en)	1193
21.3.2 Die aspx.cs-Datei(en)	1196
21.3.3 Die Datei Global.asax	1197
21.3.4 Das Startformular	1197
21.3.5 Die Datei Web.config	1198
21.3.6 Masterpages (master-Dateien)	1201
21.3.7 Sitemap (Web.sitemap)	1201
21.3.8 Benutzersteuerelemente (ascx-Dateien)	1201
21.3.9 Die Web-Projekt-Verzeichnisse	1202
21.3.10 Die ASP.NET-Projekttypen	1203
21.4 Lernen am Beispiel	1204
21.4.1 Erstellen des Projekts	1204
21.4.2 Oberflächengestaltung	1205
21.4.3 Ereignisprogrammierung	1206
21.4.4 Ein Fehler, was nun?	1207
21.4.5 Ereignisse von Textboxen	1209
21.4.6 Ein gemeinsamer Ereignis-Handler	1209
21.4.7 Eingabefokus setzen	1210
21.4.8 Ausgaben in einer Tabelle	1210
21.4.9 Scrollen der Anzeige	1213
21.4.10 Zusammenspiel mehrerer Formulare	1214
21.4.11 Umleiten bei Direktaufruf	1215

21.4.12	Ärger mit den Cookies	1216
21.4.13	Export auf den IIS	1217
22	Übersicht ASP.NET-Controls	1221
22.1	Einfache Steuerelemente im Überblick	1221
22.1.1	Label	1221
22.1.2	TextBox	1223
22.1.3	Button, ImageButton, LinkButton	1224
22.1.4	CheckBox, RadioButton	1225
22.1.5	CheckBoxList, BulletList, RadioButtonList	1226
22.1.6	Table	1227
22.1.7	Hyperlink	1229
22.1.8	Image, ImageMap	1229
22.1.9	Calendar	1231
22.1.10	Panel	1232
22.1.11	HiddenField	1232
22.1.12	Substitution	1233
22.1.13	XML	1234
22.1.14	FileUpload	1236
22.1.15	AdRotator	1237
22.2	Steuerelemente für die Seitennavigation	1238
22.2.1	Mehr Übersicht mit Web.Sitemap	1239
22.2.2	Menu	1240
22.2.3	TreeView	1243
22.2.4	SiteMapPath	1246
22.2.5	MultiView, View	1247
22.2.6	Wizard	1248
22.3	Webseitenlayout/-design	1250
22.3.1	Masterpages	1250
22.3.2	Themes/Skins	1253
22.3.3	Webparts	1256
22.4	Die Validator-Controls	1257
22.4.1	Übersicht	1257
22.4.2	Wo findet die Fehlerprüfung statt?	1258
22.4.3	Verwendung	1258
22.4.4	RequiredFieldValidator	1259
22.4.5	CompareValidator	1260
22.4.6	RangeValidator	1262

22.4.7	RegularExpressionValidator	1262
22.4.8	CustomValidator	1263
22.4.9	ValidationSummary	1265
22.4.10	Weitere Möglichkeiten	1267
22.5	Praxisbeispiele	1267
22.5.1	Themes und Skins verstehen	1267
22.5.2	Masterpages verwenden	1272
22.5.3	Webparts verwenden	1275

23 Datenbindung unter ASP.NET 1281

23.1	Alt und Neu im Vergleich	1281
23.1.1	Variante 1: Konventionelle Programmierung	1281
23.1.2	Variante 2: Mit DataSource	1284
23.2	Einführung	1287
23.2.1	Konzept	1288
23.2.2	Übersicht DataSource-Steuerelemente	1289
23.3	SQLDataSource	1290
23.3.1	Datenauswahl mit Parametern	1291
23.3.2	Parameter für INSERT, UPDATE und DELETE	1292
23.3.3	Methoden	1294
23.3.4	Caching	1295
23.3.5	Aktualisieren/Refresh	1296
23.4	AccessDataSource	1296
23.5	ObjectDataSource	1296
23.5.1	Verbindung zwischen Objekt und DataSource	1296
23.5.2	Ein Beispiel sorgt für Klarheit	1298
23.5.3	Geschäftsobjekte in einer Session verwalten	1302
23.6	SitemapDataSource	1304
23.7	LinqDataSource	1305
23.7.1	Bindung von einfachen Collections	1305
23.7.2	Bindung eines LINQ to SQL-DataContext	1306
23.8	XmlDataSource	1308
23.9	GridView	1309
23.9.1	Auswahlfunktion (Zeilenauswahl)	1310
23.9.2	Auswahl mit mehrspaltigem Index	1311
23.9.3	Hyperlink-Spalte für Detailansicht	1311
23.9.4	Spalten erzeugen	1312
23.9.5	Paging realisieren	1313

23.9.6	Edit, Update, Delete	1315
23.9.7	Keine Daten, was tun?	1315
23.10	DetailsView	1316
23.11	FormView	1317
23.12	DataList	1320
23.12.1	Bearbeitungsfunktionen implementieren	1321
23.12.2	Layout verändern	1322
23.13	Repeater	1323
23.14	ListView	1324

24 ASP.NET-Objekte und Techniken 1325

24.1	Wichtige ASP.NET-Objekte	1325
24.1.1	HTTPApplication	1325
24.1.2	Application	1328
24.1.3	Session	1329
24.1.4	Page	1330
24.1.5	Request	1333
24.1.6	Response	1336
24.1.7	Server	1341
24.1.8	Cookies verwenden	1342
24.2	Fehlerbehandlung unter ASP.NET	1344
24.2.1	Fehler beim Entwurf	1344
24.2.2	Laufzeitfehler	1345
24.2.3	Eine eigene Fehlerseite	1346
24.2.4	Fehlerbehandlung im Web Form	1347
24.2.5	Fehlerbehandlung in der Anwendung	1348
24.2.6	Alternative Fehlerseite einblenden	1349
24.2.7	Lokale Fehlerbehandlung	1350
24.2.8	Seite nicht gefunden! – Was nun?	1351
24.3	E-Mail-Versand in ASP.NET	1351
24.3.1	Übersicht	1352
24.3.2	Mail-Server bestimmen	1352
24.3.3	Einfache Text-E-Mails versenden	1354
24.3.4	E-Mails mit Dateianhang	1355
24.4	Sicherheit von Webanwendungen	1356
24.4.1	Authentication	1356
24.4.2	Forms Authentication realisieren	1357
24.4.3	Impersonation	1361

24.4.4	Authorization	1365
24.4.5	Administrieren der Website	1367
24.4.6	Steuerelemente für das Login-Handling	1371
24.4.7	Programmieren der Sicherheitseinstellungen	1375
24.5	Die Verwendung von AJAX in ASP.NET-Anwendungen	1377
24.5.1	Was ist eigentlich AJAX und was kann es?	1377
24.5.2	Die AJAX-Controls in Visual Studio 2008	1378
24.5.3	AJAX-Control-Toolkit	1382
24.6	User Controls/Webbenutzersteuerelemente	1383
24.6.1	Ein simples Einstiegsbeispiel	1383
24.6.2	Dynamische Grafiken in einem User Control anzeigen	1387
24.6.3	Grafikausgaben per User Control realisieren	1392

Anhang

A	Glossar	1397
B	Wichtige Dateiextensions	1403
	Index	1405