

Auf einen Blick

1 Einführung	17
Teil I Erste Schritte.....	27
2 Aufbau von C#-Programmen	29
3 Konstanten, Variablen & Datentypen	47
4 Zusammengesetzte Datentypen und Namensräume	81
5 Operatoren	101
6 Kontrollkonstrukte	127
Teil II Objektorientierte Programmierung mit C#	153
7 Einführung in die objektorientierte Programmierung ...	155
8 Generische Klassen, Schnittstellen und statische Klassenmitglieder	213
9 Strings & reguläre Ausdrücke	245
10 Ausnahmen – Exceptions	267
11 Überladen von Operatoren	283
12 Delegates und Ereignisse	291
13 Indizierer, Enumeratoren und Collections	315
14 Attribute und Metadaten	341
15 XML-Dokumentation und Präprozessor	367
16 Threading	389
Teil III Windows Forms.....	423
17 Einführung	425
18 Steuerelemente	471
19 Grafik und Drucken in Windows Forms- Anwendungen	505
20 Bibliotheken und CodeDOM	555
A Fehlersuche in Programmen	569
B Hilfequellen	579
Index	583

Inhalt

1	Einführung	17
1.1	.NET	17
1.1.1	Verwendung mehrerer Programmiersprachen für ein Projekt	19
1.1.2	Garbage Collection und Sicherheit	21
1.2	C#	22
1.3	Zielgruppe	23
1.4	Danksagungen	23
1.5	Kontakt	24
1.6	Aufbau des Buches	24
1.6.1	Anmerkungen zur verwendeten Notation	25
1.7	C# Version 1 oder 2?	25
1.8	Einrichten einer Entwicklungsumgebung	25

Teil I Erste Schritte 27

2	Aufbau von C#-Programmen	29
2.1	Hello, World!	29
2.1.1	Der Projektmappen-Explorer	32
2.1.2	Das Eigenschaftsfenster	33
2.1.3	Die Fehlerliste	33
2.1.4	Der Texteditor	33
2.2	Kommentare	35
2.2.1	Kommentarblöcke	36
2.2.2	Zeilenkommentare	37
2.3	Syntax und Semantik	38
2.4	Verwendete Syntaxschreibweise	38
2.5	Eine kurze Einführung zum Thema »Klassen«	38
2.5.1	Deklaration von Klassen	39
2.5.2	Der Einsprungspunkt	40
2.6	Zusammenhang zwischen Klassen- & Dateiname	40
2.7	Ausgaben auf dem Bildschirm	41
2.7.1	Namensräume	41
2.7.2	Umbruch der Ausgabe von WriteLine()	42

2.8	Assembly	42
2.9	Zusammenfassung	44
2.9.1	Bestandteile eines C#-Programmes	44
2.9.2	Zusätzliches über das .NET Framework	45
2.9.3	C# Sprachelemente	45
2.10	Übungen	46

3 Konstanten, Variablen & Datentypen 47

3.1	Das EVA-Prinzip	47
3.2	Variablen	48
3.2.1	Primitive Datentypen in C#	49
3.2.2	Datentypen für ganzzahlige Werte (Menge der ganzen Zahlen)	50
3.2.3	Datentypen für gebrochene Werte (Menge der rationalen Zahlen)	51
3.2.4	Wahrheitswerte	52
3.2.5	Zeichenketten	53
3.2.6	Namenskonventionen für Variablen	53
3.2.7	Deklaration von Variablen	55
3.2.8	Wertzuweisung und Initialisierung von Variablen	56
3.2.9	Ausgabe von Variableninhalten	62
3.2.10	Gültigkeit von Variablen	67
3.3	Felder	68
3.3.1	Deklaration und Initialisierung	69
3.3.2	Zugriff und Ausgabe von Arrays: [...], foreach	70
3.3.3	Mehrdimensionale Arrays	72
3.3.4	Unregelmäßige Arrays	73
3.3.5	Speicherbereinigung	73
3.4	Parameter der Main-Funktion	74
3.5	Typqualifizierer	75
3.5.1	static	76
3.5.2	const	76
3.6	Einlesen von Variablenwerten über die Tastatur	76
3.7	Zusammenfassung	77
3.8	Übungen	79

4 Zusammengesetzte Datentypen und Namensräume 81

4.1	Strukturen	81
4.1.1	Deklaration einer Struktur	81
4.1.2	Eine Struktur zur Aufnahme von Adressen	84
4.1.3	Werte- und Verweistypen	85
4.1.4	Boxing und Unboxing	89

4.2	Aufzählungen (Enumerationen)	89
4.2.1	Deklaration einer Aufzählung	91
4.2.2	Enumerationen und Zahlwerte	91
4.2.3	Basistyp einer Enumeration	93
4.3	Namensräume	94
4.3.1	Definition eines Namensraums	94
4.3.2	Die using-Klausel	96
4.4	Zusammenfassung	97
4.4.1	Strukturen	97
4.4.2	Enumerationen	98
4.4.3	Namespaces	98
4.5	Übungen	99

5 Operatoren 101

5.1	Operatoren in C#	101
5.2	Additive und multiplikative Operatoren	103
5.2.1	Addition, Subtraktion, Multiplikation und Division	103
5.2.2	Division mit Rest (Division modulo x)	104
5.3	Der Zuweisungsoperator	105
5.4	Primäre Operatoren	106
5.4.1	Klammerung "()"	106
5.4.2	Memberzugriff "."	107
5.4.3	Methodenaufruf	107
5.4.4	Array-Zugriff "[]"	108
5.4.5	Post-Inkrement und Post-Dekrement	108
5.4.6	Anlegen von Objekten – Der new-Operator	109
5.4.7	Typ und Größe einer Variablen	111
5.4.8	Geprüfte und ungeprüfte Ausführung von Operationen	113
5.5	Unäre Operatoren	116
5.5.1	Vorzeichen	116
5.5.2	Negationen	116
5.5.3	Pre-Inkrement und Pre-Dekrement	117
5.5.4	Typumwandlung	118
5.6	Schiebe-Operatoren	119
5.7	Relationale und Vergleichs-Operatoren	120
5.7.1	Vergleichsoperatoren	120
5.7.2	Die Operatoren is und as	121
5.8	Logisches UND, ODER und EXKLUSIV ODER (XOR)	122
5.9	Bedingtes UND, ODER und EXKLUSIV ODER	123
5.10	Bedingung	124
5.11	Zusammenfassung	125
5.12	Übungen	125

6 Kontrollkonstrukte 127

6.1	Nassi-Shneiderman	127
6.2	Schleifen	128
6.2.1	for	129
6.2.2	while	133
6.2.3	do-while	136
6.2.4	foreach	138
6.2.5	Steuerung der Schleifenabläufe: break & continue	138
6.3	Bedingungen (bedingte Anweisungen)	139
6.3.1	if	139
6.3.2	switch-case	144
6.4	goto	148
6.5	Unterstützung durch die Entwicklungsumgebung	149
6.6	Zusammenfassung	149
6.7	Übungen	150

Teil II Objektorientierte Programmierung mit C# 153

7 Einführung in die objektorientierte Programmierung 155

7.1	Klassen und Objekte	155
7.1.1	Die Klasse als programmiersprachliche Beschreibung von realen Objekten	155
7.1.2	Objektmethoden	163
7.1.3	Eine spezielle Methode: Der Konstruktor	172
7.1.4	Die Vererbungslehre in der Programmiersprache	177
7.1.5	Gleichheit von Objekten	188
7.1.6	Überladen von Methoden	188
7.1.7	Überschreiben von Methode: Virtuelle Methoden	191
7.2	Abstrakte Klassen	194
7.2.1	Abstrakte Methoden	195
7.3	Eigenschaften (Properties)	196
7.3.1	get	198
7.3.2	set	199
7.3.3	Unterschiedliche Sichtbarkeitsstufen für Getter und Setter	200
7.3.4	Benutzung von Properties	201
7.4	Klassen auf mehrere Dateien verteilen: partielle Klassen	202
7.5	Ref- & Out-Parameter von Methoden	203
7.5.1	Ref-Parameter	203
7.5.2	Out-Parameter	204

7.6	Die Speicherverwaltung von .NET	205
7.6.1	Finalize() und C#-Destruktoren	206
7.6.2	Referenzen auf Objekte während Finalisierung	208
7.7	Zusammenfassung	208
7.7.1	Klassen	208
7.7.2	Properties	209
7.7.3	Methoden	209
7.8	Übungen	209

8 Generische Klassen, Schnittstellen und statische Klassenmitglieder **213**

8.1	Klassen mit Typparametern: Generische Klassen – Generics	213
8.1.1	Anwendung auf die Klasse Stack	214
8.1.2	Instanzen der Klasse Stack erstellen	216
8.1.3	Typparameter und Vererbung	216
8.1.4	Generische Methoden	217
8.2	Schnittstellen	218
8.2.1	Deklaration einer Schnittstelle	218
8.2.2	Implementierung einer Schnittstelle	220
8.2.3	Schnittstellen und Typparameter	222
8.2.4	Einschränkungen für Typparameter festlegen	223
8.3	Klassenmember (statische Member)	224
8.3.1	Statische Methoden	225
8.3.2	Statische Daten	225
8.3.3	Statische Eigenschaften	226
8.3.4	Statische Daten in nicht-statischen Methoden	226
8.3.5	Statische Klassen	226
8.3.6	Erzeugung von Objekten mit privaten Konstruktoren	227
8.4	Objektorientierte Schenkerl5 in C#	228
8.4.1	Neuimplementierung einer Methode	228
8.4.2	sealed vor virtuellen Methoden	233
8.4.3	Zugriff auf Implementierung der Basisklasse: base Teil II	233
8.4.4	Dispose(): Eine bessere Lösung als Finalize()	236
8.5	Weitere Elemente der Unified Modelling Language	237
8.5.1	Schnittstellen	237
8.5.2	Assoziationen	238
8.6	Unterstützung durch die Entwicklungsumgebung	239
8.7	Zusammenfassung	241
8.7.1	Generische Klassen	241
8.7.2	Schnittstellen	241
8.7.3	Klassenmember	242
8.7.4	Methoden und die Vererbung	242
8.8	Übungen	242

9 Strings & reguläre Ausdrücke

245

9.1	Zeichenketten	245
9.1.1	string vs. String	245
9.1.2	Länge von Zeichenketten	246
9.1.3	Iteration über eine Zeichenkette	246
9.1.4	Vergleich zweier Zeichenketten	247
9.1.5	Untersuchung von Zeichenketten	250
9.1.6	Splitting von Strings	251
9.1.7	Zurechtschneiden von Zeichenketten	252
9.1.8	Groß- und Kleinschreibung	253
9.1.9	Löschen und Ersetzen von Zeichen	253
9.1.10	Einfügen von Zeichen	254
9.1.11	Teilstrings herauslösen	254
9.1.12	Weitere Operationen mit Strings im Überblick	255
9.1.13	Format()	255
9.2	Dynamische Zeichenketten – StringBuilder	255
9.2.1	Instantiierung eines StringBuilder-Objektes	256
9.2.2	Eigenschaften und Methoden	256
9.2.3	ToString()	257
9.3	Reguläre Ausdrücke	259
9.3.1	Regex: Eine Klasse für reguläre Ausdrücke	259
9.3.2	Grundlegender Aufbau eines regulären Ausdrucks	260
9.3.3	Optionen	262
9.4	Zusammenfassung	265
9.5	Übungen	266

10 Ausnahmen – Exceptions

267

10.1	Der klassische Ansatz: Rückgabewerte	267
10.2	Exception-Mechanismus	269
10.2.1	Exceptions – Allgemein	269
10.2.2	Exceptions – Wie funktioniert's?	269
10.2.3	Exceptions – Nachteile	270
10.2.4	Exceptions und .NET	271
10.3	Exceptions in C#	271
10.3.1	Das Werfen einer Exception	271
10.3.2	Das Fangen einer Exception	272
10.3.3	Member der Klasse Exception	273
10.3.4	Exception-Klassen	274
10.3.5	Eigene Exception-Klassen	277
10.3.6	Aufräumarbeiten: finally	278
10.3.7	Verhaltensweisen beim Auftreten einer Ausnahme	279
10.4	Zusammenfassung	281
10.5	Übungen	282

11	Überladen von Operatoren	283
11.1	Unäre Operatoren	283
11.2	Binäre Operatoren	285
11.3	Vergleichsoperatoren	287
	11.3.1 Equals()	287
	11.3.2 GetHashCode()	288
11.4	Einschränkungen	289
11.5	Zusammenfassung	289
11.6	Übungen	290
12	Delegates und Ereignisse	291
12.1	Delegates	291
	12.1.1 Beispiel: Motorüberwachung	291
	12.1.2 Die Deklaration eines Delegates	292
	12.1.3 Die Verwendung von Delegates	293
	12.1.4 Erstellen eines Delegates	295
	12.1.5 Multicast-Delegates	297
	12.1.6 Callback-Methoden und Ausnahmen	301
	12.1.7 Delegates und Rückgabewerte	301
12.2	Ereignisse	302
	12.2.1 Delegate vs. Ereignis	302
	12.2.2 Ereignisse: Hintergrund	305
	12.2.3 Deklaration eines Ereignisses	305
	12.2.4 Hinweise	307
	12.2.5 Ereignisse und Rückgabewerte	309
12.3	Generische Delegates	309
12.4	Anonyme Methoden	309
12.5	Leichtere Erzeugung von Delegate-Instanzen	310
12.6	Unterstützung durch die Entwicklungsumgebung	311
12.7	Zusammenfassung	312
12.8	Übungen	313
13	Indizierer, Enumeratoren und Collections	315
13.1	Indizierer	315
	13.1.1 Allgemeines über Indizierer	315
	13.1.2 Deklaration eines Indizierers	315
	13.1.3 Die Verwendung eines Indizierers	318
	13.1.4 Indizierer mit mehreren Parametern	320

13.2	Enumeratoren	321
13.2.1	IEnumerable und IEnumerator	321
13.2.2	Ein Enumerator für BitVector64	323
13.2.3	Erweiterungen für die Unterstützung des Enumerators an BitVector64	326
13.2.4	Verwendung des Enumerators	327
13.2.5	Zur Perfektion fehlt noch etwas	328
13.2.6	Noch einmal foreach	330
13.3	Collections in .NET	330
13.3.1	Vorwort zu Collections für .NET ab Version 2.0	331
13.3.2	ArrayList	331
13.3.3	Queue	333
13.3.4	Stack	335
13.3.5	Hashtable	337
13.4	Zusammenfassung	338
13.5	Übungen	339

14 Attribute und Metadaten 341

14.1	Attribute	341
14.1.1	Attribute im Code platzieren	342
14.1.2	Reservierte Attribute	342
14.1.3	Globale Attribute	349
14.1.4	Eigene Attribute entwickeln	349
14.1.5	Ziele für Attribute	352
14.2	Metadaten	353
14.2.1	Typinformationen zur Laufzeit ermitteln	354
14.2.2	Attribute auslesen	362
14.2.3	Ausblick	364
14.3	Zusammenfassung	364
14.4	Übungen	365

15 XML-Dokumentation und Präprozessor 367

15.1	XML-Dokumentation	367
15.1.1	XML als Datenformat	367
15.1.2	Eine weitere Form des Kommentars: ///	371
15.1.3	Dokumentationstags	372
15.1.4	Textauszeichnung und Verweise	378
15.2	Präprozessor	381
15.2.1	Ein- und Ausblenden von Code	381
15.2.2	Weitere Direktiven	384
15.3	Zusammenfassung	386
15.4	Übungen	386

16 Threading **389**

16.1 Betriebssystemhintergrund: Prozesse und Threads	389
16.1.1 Programme: Historie	389
16.1.2 Prozesse und Threads	391
16.1.3 Parallelität durch den Einsatz von Threads	392
16.2 Threads in C#	392
16.2.1 Thread-Erzeugung	392
16.2.2 Auch Threads brauchen ihren Schlaf	396
16.2.3 Suspendieren von außen	400
16.2.4 Abbruch eines Threads	402
16.2.5 Warten auf das Ende eines Threads	404
16.2.6 Aktueller Thread-Zustand	404
16.2.7 Thread-Prioritäten	405
16.2.8 Name eines Threads	407
16.2.9 Zusammenfassung der wichtigsten Methoden und Properties der Klasse Thread	407
16.2.10 Übergabe und Rückgabe von Daten	408
16.3 Synchronisierung	408
16.3.1 Die Gefahr von nicht synchronisierten Threads	408
16.3.2 Die Klasse Monitor	410
16.3.3 Ein Monitor in C#	412
16.4 Asynchrone Methodenaufrufe	414
16.4.1 BeginInvoke() und EndInvoke()	414
16.4.2 Methodenaufruf ohne Parameter und Rückgabewert	415
16.4.3 Methodenaufruf mit Parametern und Rückgabewert	417
16.5 Zusammenfassung	420
16.6 Übungen	421

Teil III Windows Forms **423**

17 Einführung **425**

17.1 Die Definition und Funktionsweise eines Fensters	425
17.2 Die Windows Forms-Bibliothek	427
17.2.1 Der Namensraum	427
17.2.2 Fenster vs. Formulare	427
17.2.3 Grundgerüst für Windows Forms-Programme	428
17.3 Die Klasse Form im Detail	430
17.3.1 Hinweise zur Online-Hilfe	430
17.3.2 Wichtige Eigenschaften der Klasse Form	430
17.3.3 Wichtige Methoden der Klasse Form	437
17.3.4 Wichtige Ereignisse der Klasse Form	441

17.4	Einfügen von Elementen in ein Fenster	447
17.4.1	Einfügen einer Schaltfläche in ein Fenster	448
17.4.2	Das Koordinatensystem von Windows Forms	450
17.4.3	Die Positionierung von Elementen in einem Fenster mit Hilfe der Dock- und Anchor-Eigenschaft	451
17.4.4	Das Baukastensystem von Windows Forms	455
17.5	Dialoge	456
17.5.1	Dialoge mit ShowDialog() anzeigen	456
17.5.2	Vorgefertigte Dialoge	458
17.5.3	Die Klasse MessageBox	458
17.6	Unterstützung durch die Entwicklungsumgebung	460
17.6.1	Ein neues Projekt anlegen	460
17.6.2	Hinzufügen und Parametrieren eines Steuerelements	462
17.6.3	Automatisch generierter Code	465
17.7	Zusammenfassung	468
17.8	Übungen	468

18 Steuerelemente 471

18.1	Die Basisklasse Control	471
18.1.1	Eigenschaften	471
18.1.2	Methoden	475
18.1.3	Ereignisse	477
18.2	Containersteuerelemente	478
18.2.1	Formulare – Fenster	479
18.2.2	Panels – Platten	479
18.2.3	FlowLayoutPanel und TableLayoutPanel	480
18.2.4	SplitContainer	482
18.2.5	GroupBox	484
18.2.6	TabControl	484
18.2.7	Anmerkungen zum Umgang mit Containersteuerelementen	484
18.3	Schaltflächen, Checkboxes und Radiobuttons	485
18.3.1	Schaltflächen – Buttons	485
18.3.2	Optionsfelder Teil 1: Checkboxes	488
18.3.3	Optionsfelder Teil 2: Radiobuttons	489
18.4	Textfelder	490
18.4.1	Einfache Textein-/ausgaben: TextBox	490
18.4.2	Formatierte Eingaben: MaskedTextBox	490
18.4.3	Dokumente mit Formatierungen: RichTextBox	491
18.5	Beschriftungen	492
18.6	Bilder	494
18.7	Listen und Tabellen	494
18.7.1	Einfache Listen: ListBox und CheckedListBox	495
18.7.2	Aufklappbare Listen: ComboBox	495
18.7.3	Listen und Tabellen: ListView	496
18.7.4	Bäume: TreeView	497

18.8	Menüs, Symbolleisten und Kontextmenüs	497
18.8.1	Menüs	497
18.8.2	Symbolleisten	499
18.8.3	Kontextmenüs	500
18.9	Sonstige Steuerelemente	500
18.9.1	Darstellung von Objekteigenschaften: PropertyGrid	500
18.9.2	Zeitlich wiederkehrende Aufgaben auslösen	501
18.9.3	Fortschrittsanzeige und Schiebepalken: ProgressBar und TrackBar	502
18.10	Zusammenfassung	503
18.11	Übungen	504

19 Grafik und Drucken in Windows Forms-Anwendungen **505**

19.1	Grafik – GDI+	505
19.1.1	Einfache, grafische Ausgaben: Das Graphics-Objekt	505
19.1.2	Der Garbage Collector und das Graphics-Objekt	506
19.1.3	Farben	507
19.1.4	Stifte und Pinsel	510
19.1.5	Koordinaten und Punkte	515
19.1.6	Rechtecke	515
19.1.7	Größe des Zeichenbereichs ermitteln	516
19.1.8	Methoden zum Zeichnen	516
19.1.9	Methoden zum Füllen von Flächen	531
19.1.10	Fehler in der grafischen Ausgabe	534
19.1.11	Ausgabe einzelner Bildpunkte	534
19.1.12	Grafische Pfade	535
19.1.13	Regionen	537
19.1.14	Erzeugen von Graphics-Objekten	537
19.2	Icons und Bilder	537
19.2.1	Icons	538
19.2.2	Bilder	539
19.3	Drucken	542
19.3.1	Ereignisse der Klasse PrintDocument	545
19.3.2	Eigenschaften der Klasse PrintDialog	546
19.3.3	Erstellen einer Druckvorschau	548
19.4	Unterstützung durch die Entwicklungsumgebung	552
19.5	Zusammenfassung	552
19.6	Übungen	553

20 Bibliotheken und CodeDOM **555**

20.1	Bibliotheken	555
20.1.1	Statische und dynamische Bibliotheken	555

20.1.2	Unterschied zwischen einem Programm und einer Bibliothek	556
20.1.3	Ein einfaches Beispiel	556
20.1.4	Projekte mit mehreren Code-Dateien	559
20.2	CodeDOM	560
20.2.1	Ein Simulator für ein Makro	561
20.2.2	Die Oberfläche der Anwendung	562
20.2.3	Code zur Laufzeit erzeugen	562
20.2.4	Vorbemerkungen	562
20.2.5	Die Methode OnClickedStart()	563
20.3	Zusammenfassung	567
20.4	Übungen	568

A Fehlersuche in Programmen 569

A.1	Trace-Ausgaben	569
A.2	Der Debugger von Visual C# 2005 Express	571
A.2.1	Programme im Debugger starten	571
A.2.2	Variablenwerte anzeigen	574
A.2.3	Weitere Debug-Fenster	576

B Hilfequellen 579

B.1	Online-Hilfe	579
B.2	Quickstart Tutorials	580
B.3	Webseiten	581
B.4	Bücher	582

Index 583