

Inhaltsverzeichnis

1	Vorwort	17
	.NET	18
	Worin unterscheidet sich dieses Buch von anderen C#/.NET Büchern?	18
	Für wen ist dieses Buch gedacht?	19
	Was ist nicht beschrieben?	19
	Systemanforderungen	19
	.NET 1, .NET 2,	19
	Beispieldateien zu diesem Buch	20
	CD zu diesem Buch	20
	Korrekturen, Kommentare und Hilfe	20
	Danksagungen	20
	Teil A	
	C# Grundlagen	23
2	Einführung	25
	Was ist .NET?	26
	Übersetzungsarchitektur	29
	Programmiersprache C#	29
	Microsoft Intermediate Language	30
	Assemblies	33
	Metadaten	34
	Die Common Language Runtime (CLR)	35
	JIT-Übersetzung	36
	Der Klassenlader	38
	Speicherverwaltung	39
	Sicherheit	40
	.NET-Klassenbibliothek	41
	Multithreading	42
	Das Common Type System (CTS)	44
	Die Common Language Specification (CLS)	46
	Erste Schritte in MSIL und C#	47
	Assemblersprache MSIL	47
	Programmiersprache C#	55
3	Elementare Sprachkonstrukte einer höheren Programmiersprache	61
	Das Lexikon der C# Programmiersprache	62
	Der Zeichensatz	62

Kommentare	62
Bezeichner	63
Schlüsselwörter	63
Elementare Datentypen	65
Konstanten (Literele)	66
Typkonvertierungen	68
Einfache Sprachelemente	70
Deklarationen	70
Operatoren	73
Kontrollstrukturen	81
Wiederholungsanweisungen	81
Sprunganweisungen	83
Bedingte Anweisungen	84
Zusammengesetzte Datentypen	89
Aufzählungen	89
Strukturen	90
Programmstruktur	91
Methoden	91
Programmstart	92
Namensräume	93
Aufgaben	95
Aufgabe 1: Teuflische Folgen	95
Aufgabe 2: Längste und kürzeste teuflische Folge	95
Aufgabe 3: Verflixte Sieben	95
Lösungen	96
Lösung zu Aufgabe 1: Teuflische Folgen	96
Lösung zu Aufgabe 2: Längste und kürzeste teuflische Folge	97
Lösung zu Aufgabe 3: Verflixte Sieben	98
4 Einführung in Klassen und Objekte	99
Klassen und Objekte	100
Motivation	100
Klassen	102
Instanzvariablen	102
Instanzmethoden	103
Objekte	108
Zugriff auf Instanzvariablen und Instanzmethoden	109
Lebensdauer von Variablen und Objekten	110
Ein Blick hinter die Kulissen: C# und C++ im Vergleich	111
Werttypen und Referenztypen	114
Initialisierung von Objekten	118
Konstruktoren	120
Mehrere Konstruktoren	121
Der Standardkonstruktor	122
Der <i>this</i> -Operator	123
Verkettung von Konstruktoren	124
Direkte Initialisierung von Instanzvariablen	125
Sichtbarkeitsattribute	126
Sichtbarkeitsattribute für Instanzvariablen	127

Sichtbarkeitsattribute für Methoden	128
Sichtbarkeitsattribute für Klassen	131
Herkömmliche Zugriffsmethoden	131
Freigabe von Objekten	133
Klassenvariablen und -methoden	136
Klassenvariablen	138
Klassenmethoden	141
Aufgaben	143
Aufgabe 1: Die Klasse <i>Time</i>	143
Aufgabe 2: Die Klasse <i>Complex</i>	144
Lösungen	148
Lösung zu Aufgabe 1: Die Klasse <i>Time</i>	148
5 Arrays	151
Grundlagen von Arrays	152
Arrays definieren	152
Arrays deklarieren	152
Arrays erzeugen	153
Mehrdimensionale Arrays	154
Arrays verwenden	155
Zugriff auf Array-Elemente	155
Arrays initialisieren	156
Arrays als Methodenparameter	158
Ein Beispiel im Überblick	159
Methoden mit variabler Parameteranzahl	160
Aufgaben	162
Aufgabe 1: Sieb des Eratosthenes	162
Aufgabe 2: Die Menge natürlicher Zahlen	162
Lösungen	163
Lösung zu Aufgabe 1: Sieb des Eratosthenes	163
6 Mehr über Klassen und Objekte	167
Eigenschaften	168
Spezielle Zugriffsmethoden für Instanzvariablen	168
Kapselung: Die erste Säule der Objektorientierung	172
Überladen von Methoden	173
Überladen von Operatoren	175
Motivation	175
Definition	176
Konvertierungsoperatoren	180
Indexer	181
Aufgaben	184
Aufgabe 1: Die Menge natürlicher Zahlen (Fortsetzung)	184
Aufgabe 2: Ein mathematischer Vektor	185
Lösungen	186
Lösung zu Aufgabe 1: Die Menge natürlicher Zahlen (Fortsetzung)	186
Lösung zu Aufgabe 2: Ein mathematischer Vektor	188

7	Das Prinzip der Vererbung	189
	Vererbung	190
	Motivation	190
	Spezialisierung und Generalisierung	192
	Basisklasse und abgeleitete Klasse	193
	Das Sichtbarkeitsattribut <i>protected</i>	196
	Instanzvariablen verdecken: Das Schlüsselwort <i>new</i>	197
	Vererbung: Die zweite Säule der Objektorientierung	198
	Methoden verdecken	199
	Noch einmal das Schlüsselwort <i>new</i>	199
	Aufruf von Methoden der Vaterklasse	200
	Methoden überschreiben	203
	Konvertierung zwischen Klassentypen	203
	Die Schlüsselwörter <i>virtual</i> und <i>override</i>	204
	Vererbung anschaulich betrachtet	207
	Polymorphismus: Die dritte Säule der Objektorientierung	208
	Abstrakte und versiegelte Klassen	211
	Abstrakte Klassen	211
	Versiegelte Klassen	213
	Die Klasse <i>System.Object</i>	214
	Die <i>Equals</i> -Methode	215
	Die <i>GetHashCode</i> -Methode	218
	Die <i>ToString</i> -Methode	219
	Die <i>GetType</i> -Methode	219
	Die <i>ReferenceEquals</i> -Methode	222
	Die <i>MemberwiseClone</i> -Methode	223
	Einheitliches Typsystem für Wert- und Referenztypen	223
	Der Mechanismus des Boxing	224
	Der Mechanismus des Unboxings	225
	Strukturen	226
	Definition von Strukturen	226
	Mit Strukturen arbeiten	229
	Attribute	231
	Standardattribute	233
	Benutzerdefinierte Attribute	237
	Aufgaben	240
	Aufgabe 1: Bankkonten und Bankinstitut	240
	Lösungen	241
	Lösung zu Aufgabe 1: Bankkonten und Bankinstitut	241
8	Schnittstellen	245
	Einführung	246
	Schnittstellen definieren	246
	Schnittstellen implementieren	247
	Schnittstellen verwenden	248
	Standardschnittstellen	254
	Objekte kopieren: Schnittstelle <i>ICloneable</i>	254
	Objekte aufzählen: Schnittstellen <i>IEnumerator</i> und <i>IEnumerable</i>	257
	Objekte vergleichen: Schnittstellen <i>IComparer</i> und <i>IComparable</i>	263

Abstrakte Klassen und Schnittstellen im Vergleich	268
Aufgaben	269
Aufgabe 1: Die Menge natürlicher Zahlen (Fortsetzung)	269
Aufgabe 2: Zahlenzauberei mit 6174	269
Lösungen	270
Lösung zu Aufgabe 1: Die Menge natürlicher Zahlen (Fortsetzung)	270
9 Mehr über Arrays, Collections und Generics	273
Mehr über Arrays	274
Die Klasse <i>System.Array</i>	274
Wertzuzuweisung von Arrays	278
Array von Arrays	279
Collections	281
Überblick	281
Listen	283
Hashtabellen	289
Generics	293
Aufgaben	304
Aufgabe 1: Polynome	304
Aufgabe 2: Matrizen	307
Lösungen	308
Lösung zu Aufgabe 1: Polynome	308
Lösung zu Aufgabe 2: Matrizen	310
10 Mehr über Zeichenfolgen und Aufzählungen	313
Die Klasse <i>System.String</i>	314
Zeichenfolgen erzeugen	315
Zeichenfolgen vergleichen	315
Eigenschaften, Klassenvariablen und Indexer der Klasse <i>String</i>	316
Methoden der Klasse <i>String</i>	317
Die Klasse <i>System.Text.StringBuilder</i>	320
<i>StringBuilder</i> -Objekte erzeugen	321
Methoden der <i>StringBuilder</i> -Klasse	322
Formatierung von Zeichenfolgen	324
Ausgabe von Variableninhalten	324
Vordefinierte Formatzeichenfolgen	326
Benutzerdefinierte Formatzeichenfolgen	328
Formatzeichenfolgen und die <i>ToString</i> -Methode	329
Die Klasse <i>System.Enum</i>	333
Methoden der Klasse <i>System.Enum</i>	333
Aufzählungen und Bitfelder	337
Aufgaben	338
Aufgabe 1: Exakte Arithmetik ganzer Zahlen	338
Lösungen	340
Lösung zu Aufgabe 1: Exakte Arithmetik ganzer Zahlen	340

11	Delegierte und Ereignisse	343
	Ein Blick zurück: Methodenadressen in C und C++	344
	Methodenadressen objektorientiert: Delegierte	347
	Definition eines Delegate-Typs	348
	Delegate-Instanz erzeugen	348
	Delegate-Instanz aufrufen	350
	Delegate-Instanzen mit Klassenmethoden	350
	Ein Beispiel im Überblick	351
	Singlecast versus Multicast Delegierte	352
	Ein Blick hinter die Kulissen von Delegates	354
	Ereignisverarbeitung	358
	Wichtige Begriffe im Umfeld des .NET-Ereignismodells	358
	Ereignisse definieren	358
	An- und Abmelden an einer Ereignisquelle	359
	Anonyme Methoden	360
	Ereignisse auslösen	362
	Das Beispiel im Überblick	362
	Entwurfsmuster für Ereignisse	365
	Ein Blick hinter die Kulissen von Events	368
	Aufgaben	370
	Aufgabe 1: Ein generischer BubbleSort-Algorithmus	370
	Aufgabe 2: Simulation einer elektrischen Lampenschaltung	372
	Lösungen	372
	Lösung zu Aufgabe 1: Ein generischer BubbleSort-Algorithmus	372
	Lösung zu Aufgabe 2: Simulation einer elektrischen Lampenschaltung	375
12	Behandlung von Fehlersituationen	379
	Motivation	380
	Ausnahmen werfen und fangen	381
	Ausnahmeobjekte	382
	Ausnahmen werfen: Anweisung <i>throw</i>	383
	Ausnahmen fangen: Der <i>try</i> - und <i>catch</i> -Block	384
	Mehrere <i>catch</i> -Blöcke	387
	Anweisung <i>finally</i>	390
	Benutzerdefinierte Ausnahmen	392
	Ausnahmen und arithmetischer Überlauf	394
	Aufgaben	396
	Aufgabe 1: Matrizen (Fortsetzung)	396
	Lösungen	396
	Lösung zu Aufgabe 1: Matrizen (Fortsetzung)	396
13	Fallstudien	401
	Fundamentale Datenstrukturen: Stapel und Warteschlange	402
	Dynamische Datenstrukturen: Die verkettete Liste	407

Teil B	
.NET Framework Programmierung	413
14 Einfache Grafikelemente und Text	415
Erstellung einer Benutzeroberfläche	416
Fensterobjekte	416
Eigenschaften	417
Verhalten	419
Spezialisierung	421
Ereignisse	423
Ressourcen	426
Das Graphics Device Interface	429
Elementare Grafikausgaben	429
Das Koordinatensystem der Klasse <i>Graphics</i>	431
Farben	433
Schriftarten	436
Einige elementare Zeichenoperationen	444
Linien und Zeichenstifte	444
Rechtecke und Füllfarben	447
Kreise und Ellipsen	451
Polygone	455
Aufgaben	458
Aufgabe 1: Die Mandelbrotmenge	458
Aufgabe 2: Der Baum des Pythagoras	461
Lösungen	462
Lösung zu Aufgabe 1: Die Mandelbrotmenge	462
15 Ereignisse in grafischen Anwendungen	467
Entwurfsmuster für die Ereignisbearbeitung	468
Variante 1: Implementierung eines Ereignishandlers	468
Variante 2: Überschreiben eines Ereignishandlers	472
Grafische Ausgaben: Ereignis <i>Paint</i>	473
Schließen eines Fensters: Ereignisse <i>Closing</i> und <i>Closed</i>	476
Aufgaben	481
Aufgabe 1: Der Baum des Pythagoras (Fortsetzung)	481
Aufgabe 2: Ein einfaches Malprogramm	482
Lösung	483
Lösung zu Aufgabe 1: Der Baum des Pythagoras (Fortsetzung)	483
Lösung zu Aufgabe 2: Ein einfaches Malprogramm	484
16 Vordefinierte Steuerelemente	489
Steuerelemente und Behälter	490
Die wichtigsten Standardsteuerelemente	495
Schaltflächen und Beschriftungselemente	495
Textfelder	499
Kontrollkästchen und Optionsfelder	504
Listfelder und Auswahlfelder	507

Schieberegler, Schieberegler und Fortschrittsanzeige	510
Up-Down Steuerelemente	515
Steuerelemente dynamisch anordnen	516
Steuerelemente verankern	516
Steuerelemente andocken	518
Steuerelemente aufsplitten	519
Steuerelemente hierarchisch anordnen	521
Die Klasse <i>Panel</i>	521
Die Klasse <i>GroupBox</i>	522
Steuerelemente verschalten	524
Verschaltung mit Ereignissen	525
Verschaltung mit Adapterklassen	526
Aufgaben	528
Aufgabe 1: Eingabe der Kontonummer in einem Textfeld	528
Aufgabe 2: Ein einfaches Malprogramm (Fortsetzung)	528
Aufgabe 3: Vier Gewinnt	529
Lösungen	530
Lösung zu Aufgabe 1: Eingabe der Kontonummer in einem Textfeld	530
Lösung zu Aufgabe 2: Ein einfaches Malprogramm (Fortsetzung)	532
17 Benutzerdefinierte Steuerelemente	535
Entwicklung benutzerdefinierter Steuerelemente	536
Die Klassen <i>Component</i> , <i>Control</i> und <i>UserControl</i>	536
Gestaltung der Oberfläche	537
Eigenschaften implementieren	539
Methoden implementieren	542
Ereignisse implementieren	546
Mehrere Steuerelemente zusammenfassen	552
Das Model-View-Controller Paradigma	554
Grundlagen	554
Kommunikationswege	556
Eine Ventilsteuerung mit MVC-Architektur	557
Aufgaben	564
Aufgabe 1: Kuchendiagramm	564
Aufgabe 2: Verkehrsampelkomponente	566
Lösungen	566
Lösung zu Aufgabe 2: Verkehrsampelkomponente	566
18 Fallstudie: Bézier-Kurven	571
Bézier-Kurven	572
Bernsteinpolynome	575
Der Algorithmus von de Casteljau	580
Aufgaben	583
Aufgabe 1: Bézier-Applikation	583
Lösungen	584
Lösung zu Aufgabe 1: Bézier-Applikation	584

19	Fallstudie: Zelluläre Automaten	595
	Konzepte zellulärer Automaten	596
	Der Zellraum	596
	Nachbarschaft	597
	Randbedingungen	598
	Zustandsmenge	599
	Zustandsübergang	600
	Eindimensionale zelluläre Automaten	600
	Einführung	600
	Der „Modulo-2 Automat“	601
	Zweidimensionale zelluläre Automaten	603
	Das Spiel des Lebens	603
	Interessante Spielmuster	604
	Aufgaben	606
	Aufgabe 1: Der Modulo-2 Automat	606
	Aufgabe 2: Allgemeine Komponente für eindimensionale zelluläre Automaten	606
	Aufgabe 3: Spiel des Lebens	611
	Lösungen	611
	Lösung zu Aufgabe 3: Spiel des Lebens	611
20	Grundlagen von Threads	615
	Einführung	616
	Erste Schritte mit Threads	619
	Erzeugung und Start von Threads	619
	Die Klasse <i>Thread</i>	624
	Threads beenden	629
	Threadzustände	632
	Synchrone versus asynchrone Methodenausführung	635
	Aufgaben	642
	Aufgabe 1: Bewegte geometrische Objekte	642
	Aufgabe 2: Laufschrift	642
	Lösung	643
	Lösung zu Aufgabe 1: Bewegte geometrische Objekte	643
	Lösung zu Aufgabe 2: Laufschrift	645
21	Synchronisation von Threads	649
	Konkurrierender Zugriff auf Variablen	650
	Probleme des konkurrierenden Zugriffs auf eine Variable	650
	Atomare Operationen	655
	Konkurrierender Zugriff auf Objekte	658
	Probleme des konkurrierenden Zugriffs auf ein Objekt	658
	Kritische Abschnitte und das Monitorprinzip	663
	Betreten und Verlassen eines Monitors	664
	Konkurrierender Zugriff auf Windows Forms Steuerelemente	667
	Fehlende Thread-Sicherheit von Windows Forms Steuerelementen	667
	Korrektes Multithreading mit Windows Forms Steuerelementen	669

Aufgaben	672
Aufgabe 1: Paralleles Sortieren eines Arrays	672
Aufgabe 2: Eine reaktionsfähige Mandelbrotmengenanwendung	674
Lösungen	674
Lösung zu Aufgabe 1: Paralleles Sortieren eines Arrays	674
Lösung zu Aufgabe 2: Eine reaktionsfähige Mandelbrotmengenanwendung	681
22 Kooperation von Threads	687
Zusammenarbeit von Threads	688
Erweiterung des Monitorprinzips: Der Interimsraum	688
Benachrichtigungen über Zustandsänderungen eines Monitorobjekts	691
Beispiel: Ein Alarmgeber	692
Erster Schritt: Grundgerüst des Alarmgebers	693
Zweiter Schritt: Erfüllung der Randbedingungen	693
Dritter Schritt: Ein Monitor, mehrere Bedingungen	697
Das Erzeuger-/Verbraucherproblem	701
Aufgaben	703
Aufgabe 1: Visualisierung des BubbleSort-Algorithmus	703
Aufgabe 2: Das Problem der dinierenden Philosophen	707
Lösungen	710
Lösung zu Aufgabe 1: Visualisierung des BubbleSort-Algorithmus	710
Lösung zu Aufgabe 2: Das Problem der dinierenden Philosophen	712
23 Eingabe und Ausgabe	715
Dateien und Verzeichnisse	716
Datenströme	722
Klasse <i>Stream</i>	722
Drei Realisierungen für Datenströme	724
Zeichenbasierte Ein- und Ausgabe	728
Reader- und Writer-Klassen	728
Arbeiten mit der <i>StreamReader</i> - und <i>StreamWriter</i> -Klasse	728
Arbeiten mit der <i>StringReader</i> - und <i>StringWriter</i> -Klasse	732
Die Klasse <i>System.Console</i>	735
Binäre Ein- und Ausgabe	738
Die Klassen <i>BinaryReader</i> und <i>BinaryWriter</i>	738
Arbeiten mit der <i>BinaryReader</i> - und <i>BinaryWriter</i> -Klasse	739
Aufgaben	741
Aufgabe 1: Betrachter für eine Verzeichnishierarchie	741
Aufgabe 2: Cross-Referenz-Liste für C#-Quelldateien	741
Lösungen	742
Lösung zu Aufgabe 1: Betrachter für eine Verzeichnishierarchie	742
Lösung zu Aufgabe 2: Cross-Referenz-Liste für C#-Quelldateien	743
24 Netzwerkprogrammierung	747
Etwas Netzwerktheorie	748
Programmierung mit Sockets	750
Die Klasse <i>System.Net.Sockets.Socket</i>	750
Ein einfacher Socket-Server und -Client	754

Ein Server, mehrere Clients	758
Wrapperklassen für Sockets	761
High-Level Client eines Echo-Servers	762
High-Level Echo-Server	765
Ein Blick hinter die Kulissen	767
Aufgaben	768
Aufgabe 1: Server für Datum und Uhrzeit	768
25 Fallstudien	771
Ein Chatraum	772
Hinweise zur Architektur	772
Chatraum-Server	772
Chatraum-Client	773
Hinweise zur Gestaltung der Oberfläche	773
Aufgaben	774
Aufgabe 1: Chatraum-Server und -Client	774
Lösungen	775
Lösung zu Aufgabe 1: Chatraum-Server und -Client	775
Stichwortverzeichnis	783

