

Inhaltsverzeichnis

Einleitung	15
<hr/>	
Teil I Grundlagen der Programmierung	19
<hr/>	
1 C# und das .NET Framework	21
1.1 Die Sprache C#	21
1.2 Das .NET Framework	22
1.2.1 Die Common Language Specification (CLS)	24
1.2.2 Das Common Type System (CTS)	24
1.2.3 Die Common Language Runtime (CLR)	24
1.3 Assemblies	25
1.4 Metadaten und das Manifest	25
1.5 Zusammenfassung	26
2 Visual Studio	27
2.1 Installation und Einrichtung	28
2.1.1 Systemvoraussetzungen	28
2.1.2 Installation	28
2.2 Projekte in Visual Studio erzeugen	29
2.2.1 Konsolenanwendung	31
2.2.2 Windows-Anwendungen	34
2.2.3 Klassenbibliothek	37
2.3 Zusammenfassung	38
3 Grundlagen	39
3.1 Kommentare	39
3.1.1 Einzeilige Kommentare	39
3.1.2 Mehrzeilige Kommentare	39
3.1.3 Xml-Kommentare	40
3.2 Variablen und Datentypen	41
3.2.1 Bezeichner	41
3.2.2 Schlüsselwörter	41
3.2.3 Primitive Datentypen	42
3.2.4 Typkonvertierungen	44
3.2.5 Überlauf überprüfen	46
3.2.6 Konstanten	47
3.2.7 Enumerationen	47
3.2.8 Das Schlüsselwort var	49

3.3	Operatoren	49
3.3.1	Arithmetische Operatoren	49
3.3.2	Zuweisungsoperatoren	52
3.3.3	Logische Operatoren	53
3.3.4	Bitweise Operatoren	55
3.3.5	Rangfolge der Operatoren	55
3.4	Kontrollstrukturen	56
3.4.1	Verzweigungen mit if und else	56
3.4.2	Fallunterscheidungen mit switch/case	59
3.4.3	For-Schleifen	61
3.4.4	Schleifen mit while/do...while	63
3.4.5	ForEach-Schleifen	64
3.5	Zusammenfassung	65
3.6	Aufgaben	65
4	Objektorientierte Programmierung	67
4.1	Das Typsystem	67
4.1.1	Referenz- vs. Wertetypen	67
4.1.2	Boxing und Unboxing	69
4.1.3	Das Schlüsselwort null	70
4.1.4	Nullable-Typen	70
4.2	Klassen vs. Structs	72
4.2.1	Klasse	72
4.2.2	Zugriffsmodifizierer	72
4.2.3	Das Schlüsselwort readonly	74
4.2.4	Partielle Klassen	75
4.2.5	Ein Objekt erzeugen	75
4.2.6	Konstruktor	76
4.2.7	Der this-Zeiger	78
4.2.8	Structs	80
4.3	Methoden	83
4.3.1	Methoden definieren	83
4.3.2	Methoden ohne Rückgabewert	84
4.3.3	Methoden mit Rückgabewert	85
4.3.4	Methodenparameter	86
4.3.5	Das Schlüsselwort ref	88
4.3.6	Das Schlüsselwort out	89
4.3.7	Methodenüberladung	89
4.4	Operatorenüberladung	90
4.5	Eigenschaften	92
4.5.1	Eigenschaften definieren	92
4.5.2	Automatische Eigenschaften	94
4.6	Objektinitialisierer	95
4.7	Statische Klassen	95
4.7.1	Das Schlüsselwort static	95
4.7.2	Der statische Konstruktor	97

4.8	Die Basisklasse System.Object	98
4.9	Vererbung	98
4.9.1	Vererbung definieren	100
4.9.2	Polymorphie	104
4.9.3	Abstrakte Klassen	106
4.9.4	Interface	107
4.9.5	Versiegelte Klassen	111
4.9.6	Das Schlüsselwort is	111
4.9.7	Das Schlüsselwort as	112
4.10	Namensräume	112
4.10.1	Namensräume definieren	113
4.10.2	Namespace-Alias	114
4.10.3	Namensraum-Alias-Qualifizierer	116
4.11	Region-Anweisung	116
4.12	Beispielimplementierung einer Personalverwaltung	117
4.13	Zusammenfassung	125
4.14	Aufgaben	125
5	Crashkurs: Erzeugen von Windows-Oberflächen	127
5.1	Die grafische Oberfläche	127
5.1.1	Die Menüleiste	127
5.1.2	Nummertasten und Eingabefelder	128
5.1.3	Kontextmenüs	129
5.2	Die logische Komponente	130
5.2.1	Die Schnittstelle ICalculatable	130
5.2.2	Die Klasse Addition	131
5.2.3	Die Klasse Subtraction	131
5.2.4	Die Klasse Multiplication	131
5.2.5	Die Klasse Division	131
5.2.6	Die Klasse Calculator	132
5.3	Zusammenführen der grafischen und logischen Komponenten	132
5.4	Zusammenfassung	136
Teil 2	Fortgeschrittene Programmietechniken	137
6	Fortgeschrittene C#-Techniken	139
6.1	Generics	139
6.1.1	Generische Methoden	143
6.1.2	Constraints	144
6.2	Delegaten	145
6.2.1	Anonyme Methoden	149
6.2.2	Lambda-Ausdrücke	150
6.2.3	Delegaten des .NET Frameworks	153

6.3	Ereignisse	154
6.3.1	Prinzip der Ereignisbenachrichtigung	154
6.3.2	Ereignisse deklarieren	155
6.3.3	Ereignisse auslösen	159
6.4	Exceptionhandling	160
6.4.1	try-catch-Blöcke	161
6.4.2	try-finally-Blöcke	163
6.4.3	Ausnahmen erneut auslösen	164
6.4.4	Eigene Ausnahmen definieren	165
6.5	Zeichenkettenverarbeitung	166
6.5.1	Methoden und Eigenschaften der Klasse string	166
6.5.2	Zeichenformatierung	170
6.5.3	Datum- und Uhrzeitformatierung	171
6.5.4	Die Klasse StringBuilder	173
6.6	Erweiterungsmethoden	175
6.7	Objekte vergleichen	176
6.7.1	Die Methode Equals	178
6.7.2	Die Methode ReferenceEquals	179
6.7.3	Die Schnittstelle IEquatable	179
6.7.4	Die Schnittstelle IComparable	180
6.8	Ressourcenmanagement	183
6.8.1	Die Schnittstelle IDisposable	183
6.8.2	Destruktor	184
6.8.3	Das Dispose-Pattern	185
6.8.4	Objekte aufräumen mit dem using-Block	186
6.9	Dynamische Objekte	187
6.9.1	Die Dynamic Language Runtime	187
6.9.2	Das Schlüsselwort dynamic	188
6.10	Zusammenfassung	191
6.11	Aufgaben	191
7	Collections	193
7.1	Übersicht der Collection-Klassen	194
7.2	Arrays	196
7.2.1	Arrays erzeugen	197
7.2.2	Methoden und Eigenschaften der Klasse Array	199
7.2.3	Mehrdimensionale Arrays	202
7.2.4	Indexer	203
7.2.5	Ko- und Kontravarianz	204
7.3	Collection-Initialisierer	205
7.4	Die Klasse List<T>	206
7.4.1	Eine Liste erzeugen	207
7.4.2	Methoden und Eigenschaften der Klasse List<T>	208
7.5	Die Klasse Dictionary	209
7.5.1	Ein Dictionary erzeugen	211
7.5.2	Methoden und Eigenschaften der Klasse Dictionary	212

7.6	Das Interface IEnumerable	214
7.6.1	Das Iterator-Prinzip	215
7.6.2	foreach-Schleifen und Iteratoren	218
7.6.3	Das Schlüsselwort yield	219
7.7	Weitere Collection-Klassen des .NET Frameworks	221
7.8	Erweiterung der Personalverwaltungssoftware	222
7.9	Zusammenfassung	228
7.10	Aufgaben	228
8	LINQ	229
8.1	Was ist LINQ?	229
8.1.1	Die LINQ-Syntax	231
8.1.2	Anonyme Typen	232
8.1.3	Verzögerte Ausführung von Abfragen	233
8.2	Die Standardabfrageoperatoren	235
8.2.1	Der Where-Operator	242
8.2.2	Die Operatoren Select und SelectMany	243
8.2.3	Der Operator OfType	244
8.2.4	Der Operator FirstOrDefault	245
8.2.5	Das Schlüsselwort let	246
8.3	Expression Trees	247
8.4	Beispiele	249
8.4.1	LINQ to Objects	249
8.4.2	LINQ to XML	250
8.5	Zusammenfassung	252
8.6	Aufgaben	252
9	Dateien und Verzeichnisse	255
9.1	Verzeichnisse	256
9.1.1	Die Klassen Directory und DirectoryInfo	256
9.1.2	Verzeichnis erzeugen	259
9.1.3	Verzeichnis löschen	259
9.1.4	Verzeichnis umbenennen	260
9.1.5	Verzeichnis verschieben	260
9.1.6	Verzeichnis bestimmen	260
9.2	Dateien	262
9.2.1	Die Klassen File und FileInfo	263
9.2.2	Datei erzeugen	267
9.2.3	Datei löschen	268
9.2.4	Datei umbenennen	268
9.2.5	Datei verschieben	268
9.2.6	Datei kopieren	268
9.2.7	Dateien in einem Verzeichnis ermitteln	269
9.2.8	Dateiinformationen	270
9.2.9	Dateien überwachen	272
9.3	Laufwerke	274
9.3.1	Laufwerke ermitteln	275

9.4	Zugriffsberechtigungen	276
9.5	Die Klasse Path	279
9.6	Dateien lesen und schreiben	280
9.6.1	Textdateien	281
9.6.2	Binärdateien	283
9.6.3	Serialisieren von Objekten	284
9.7	Dialoge für die Datei- und Ordnerverwaltung	286
9.7.1	Die Klasse OpenFileDialog	286
9.7.2	Die Klasse SaveFileDialog	288
9.8	Dateien verschlüsseln	288
9.8.1	Verschlüsseln mit der Klasse File	289
9.8.2	Verschlüsseln mit der Klasse CryptoStream	289
9.9	Dateien komprimieren	290
9.10	Zusammenfassung	291
9.11	Aufgaben	292
10	Unit Tests	293
10.1	Das Framework NUnit	293
10.1.1	Einen Unit Test definieren	294
10.1.2	Mehrere Testfälle definieren	298
10.1.3	Ausnahmen testen	298
10.2	Abhängigkeiten	299
10.2.1	Dependency Injection	300
10.2.2	Mocks und Stubs	301
10.2.3	Das Framework RhinoMock	302
10.3	Zusammenfassung	304

Teil III Oberflächenprogrammierung und Datenbanken

11	Konsolenanwendungen	307
11.1	Grundlagen zu Konsolenanwendungen	307
11.2	Die Klasse Console	309
11.2.1	Methoden und Eigenschaften der Klasse Console	309
11.3	Farbangaben	318
11.4	Ein- und Ausgabe	319
11.5	Tastaturabfragen	320
11.6	Konsolen und Streams	321
11.7	Beispielimplementierung: Protokollierung von Eingaben über eine Konsole in einer Textdatei	321
11.8	Zusammenfassung	323
11.9	Aufgaben	323
12	Windows Presentation Foundation (WPF)	325
12.1	Was ist die Windows Presentation Foundation?	326
12.1.1	Extensible Markup Language (XAML)	326
12.1.2	Ein Fenster erzeugen	329

12.1.3	Eigenschaften festlegen	330
12.1.4	Dependency Properties	332
12.1.5	Attached Properties	334
12.1.6	Ereignisse in XAML definieren	335
12.1.7	Die Klasse App	336
12.2	Allgemeine Steuerelemente	337
12.2.1	Label	340
12.2.2	TextBox	341
12.2.3	TextBlock	341
12.2.4	PasswordBox	342
12.2.5	Button	343
12.2.6	CheckBox und RadioButton	344
12.2.7	Toggle- und RepeatButton	345
12.2.8	Border	345
12.2.9	Image	346
12.2.10	ProgressBar	347
12.2.11	ToolTip	348
12.3	Layout definieren	349
12.3.1	StackPanel	349
12.3.2	WrapPanel	350
12.3.3	DockPanel	351
12.3.4	Canvas	352
12.3.5	Grid	353
12.3.6	UniformGrid	356
12.3.7	ScrollViewer	357
12.3.8	ViewBox	357
12.3.9	Ausrichtung des Inhalts eines Layoutcontainers	358
12.3.10	Dynamische Inhalte	360
12.4	Styles	361
12.4.1	Styles definieren	361
12.4.2	Ressourcen	363
12.4.3	Trigger	364
12.5	Templates	365
12.5.1	Templates definieren	365
12.6	Datenbindung	368
12.6.1	Bindungsrichtung festlegen	371
12.6.2	Änderungen verfolgen	372
12.6.3	Daten konvertieren	373
12.6.4	Datenquellen	375
12.6.5	Validierung von Eingaben	376
12.7	Visuelle Effekte	377
12.7.1	Transformationen	377
12.7.2	Effekte	381
12.7.3	Animationen	382
12.8	Beispielanwendung: Erfassung von Personendaten	385

12.9	Zusammenfassung	391
12.10	Aufgaben	391
13	Zugriff auf Datenbanken mit ADO.NET	393
13.1	Einführung in relationale Datenbanken	393
13.2	Übersicht der ADO.NET-Klassen	396
13.2.1	Erzeugen einer Datenbank in Visual Studio	397
13.3	Mit Datenbanken arbeiten	401
13.3.1	Eine Verbindung herstellen mit der Klasse SqlConnection	401
13.3.2	Datenbankabfragen erzeugen mit der Klasse SqlCommand	403
13.3.3	Daten aus der Datenbank lesen mit der Klasse SqlDataReader	406
13.3.4	Transaktionen durchführen	407
13.4	DataSet	408
13.4.1	Wichtige Klassen im Zusammenhang mit DataSets	409
13.4.2	Die Klasse SqlDataAdapter	409
13.4.3	Die Klasse DataTable	411
13.4.4	Daten anzeigen mit der Klasse DataView	415
13.4.5	Typisierte DataSets	417
13.5	Das ADO.NET Entity Framework	419
13.5.1	Möglichkeiten des Mappings zwischen Klassen und Datenbanken	420
13.5.2	Eine Verbindung zur Datenbank erzeugen	424
13.5.3	Daten abfragen	424
13.5.4	Daten einfügen	425
13.5.5	Daten aktualisieren	425
13.5.6	Daten löschen	426
13.6	Datenbindung an das Steuerelement DataGridView	426
13.6.1	Datenbindung unter WPF	427
13.7	Beispielanwendung: Personalverwaltung	428
13.8	Zusammenfassung	435
13.9	Aufgaben	436

Teil III Parallele Programmierung und Windows 8 437

14	Parallele Programmierung	439
14.1	Was ist parallele Programmierung?	439
14.1.1	Die Klasse Thread	441
14.1.2	Die Klasse BackgroundWorker	443
14.2	Die Task Parallel Library	448
14.2.1	Die Klasse Task	449
14.3	Die Klasse Parallel	453
14.3.1	Parallel.Invoke	454
14.3.2	Parallel.For	454
14.3.3	Parallel.ForEach	455

14.4	Die Operatoren await und async	455
14.5	PLINQ	457
14.5.1	Der Operator AsOrdered	457
14.5.2	Der Operator AsSequential	458
14.5.3	Der Operator ForAll	459
14.6	Parallele Collections	459
14.6.1	Die Klasse ConcurrentStack	460
14.6.2	Die Klasse ConcurrentQueue	461
14.6.3	Die Klasse ConcurrentBag	462
14.7	Zusammenfassung	463
14.8	Aufgaben	463
15	Windows Store Apps	465
15.1	Die Windows Runtime	466
15.2	Grundlagen einer Windows Store App	467
15.2.1	Designprinzipien	467
15.2.2	Visual-Studio-Vorlagen	469
15.2.3	Das Application Manifest	470
15.2.4	Die Klasse App	474
15.2.5	Aufbau einer Seite	475
15.2.6	Der Simulator	476
15.3	Navigation	477
15.3.1	Navigationsmuster	477
15.3.2	Die Klasse Frame	479
15.3.3	Navigation innerhalb einer Windows Store App	481
15.4	Die AppBar	483
15.4.1	Eine AppBar erzeugen	484
15.4.2	Eine NavBar erzeugen	485
15.4.3	Flyouts	486
15.5	Layout, Kacheln und der Lebenszyklus einer Windows Store App	489
15.5.1	Ansichtstypen	489
15.5.2	Die Klasse VisualStateManager	491
15.5.3	Livekacheln	493
15.5.4	Das Tombstone-Modell	496
15.5.5	Contracts	498
15.6	Umsetzung einer Windows Store App anhand einer Personalverwaltung	499
15.7	Zusammenfassung	512
Teil V	Praxisbeispiel	515
16	Fallbeispiel: Filmausleihe	517
16.1	Die Architektur	517
16.2	Erzeugung der Datenbank	518
16.2.1	Einbindung der Datenbank	520

16.3	Das Enterprise-Modell	520
16.3.1	Die zentrale Klasse RentalService	520
16.3.2	Erweiterung der Entitätsklassen	528
16.3.3	Ausnahmenbehandlung für Fortgeschrittene	528
16.4	Die Oberfläche	530
16.4.1	Vorbereitung	530
16.4.2	Die Ansichten	534
16.4.3	Zusammenführen der Ansichten	554
16.4.4	Abfangen von unerwarteten Ausnahmen	556
16.5	Zusammenfassung	557
A	Installationsanleitung	559
A.1	Systemvoraussetzungen	559
A.2	»Visual Studio Express 2012 für Windows Desktop« bzw. »Visual Studio Express 2012 für Windows 8«	559
A.3	Einbinden der Visual Studio Express-Installations-DVD bzw. -ISO-Datei	560
A.4	Programminstallation	560
A.5	Auswerfen der Installationsdatei	562
	Stichwortverzeichnis	563