

Inhaltsverzeichnis

Vorwort.....	15
Einleitung	17
1 Das .NET Framework und Visual Studio.....	21
Eine erste Definition von .NET.....	22
Welche Alternativen zu .NET existieren?	25
Entwickeln mit Java.....	26
Native Entwicklungen.....	26
Wie funktioniert .NET	26
Die Installation der Common Language Runtime	27
Die Common Language Runtime	28
Die Dynamic Language Runtime	35
Visual Studio 2010	36
Konfiguration und Einstellungen.....	37
Mit Visual Studio eine Lösung aufbauen	39
Die Konfiguration kontrollieren und korrigieren	42
Code verändern.....	50
Code mithilfe von Codierungsunterstützung erstellen.....	52
Code ausführen	52
Eine Anwendung entwanzen (debuggen).....	54
Tastenkürzel	59
Die Lösung mit einer eigenen Bibliothek ergänzen.....	60
Von Assemblys und Prozessen.....	66
Einen Prozessraum aufbauen.....	66
Konfiguration von .NET-Anwendungen	67
Suchen und Finden von Programmdateien und Assemblys.....	71
2 Die Sprache C#.....	87
Das »Huhn-Ei-Problem«	88
Typen	89
Objekte und Variablen.....	90
Methoden.....	90
Parameter und Rückgabewerte.....	90
Anweisungen und Operatoren	91
Programme und Strukturen	91

Grundlegende Elemente eines C#-Programms.....	91
Die Grundstruktur	92
Schlüsselwörter der Sprache	94
Kommentierungen	99
Variablen oder Objekte	104
Organisation, Schreibweise und Verwendung von Namensräumen	109
Literele.....	115
Anweisungen für die Ablaufsteuerung	118
Präprozessoranweisungen.....	127
Operatoren und Ausdrücke	131
Datentypen	143
Vordefinierte Typen	144
Der Typ object	145
Der Typ string.....	149
Der Typ bool	154
Der Typ byte.....	155
Der Typ sbyte.....	157
Der Typ char	158
Der Typ short.....	162
Der Typ ushort.....	164
Der Typ int	165
Der Typ uint.....	166
Der Typ long.....	168
Der Typ ulong.....	169
Der Typ float.....	170
Der Typ double.....	173
Der Typ decimal	175
Der Typ void	178
Benutzerdefinierte Typen.....	178
Der Typ enum	179
Der Typ Array.....	183
Der Typ class.....	191
Der Typ delegate.....	239
Der Typ interface.....	249
Der Typ struct.....	255
Generische Typen.....	257
Syntaktische Grundlagen	259
Generische Klassen.....	261
Generische Schnittstellen	268
Generische Methoden.....	269
Generische Typen und Arrays.....	269
Generische Delegaten	270

Nullzuweisung bei Objekten aus generischen Typen.....	271
Nullfähige Werttypen.....	272
Anonyme Typen.....	275
Implizites Festlegen von Typen	276
Arbeiten mit Objekten anonymer Typen.....	277
Erweiterungsmethoden.....	279
Eine Erweiterungsmethode erstellen	280
Erweiterungsmethoden für Klassen und Strukturen erstellen.....	281
Erweiterungsmethoden für Schnittstellen erstellen.....	283
Erweiterungsmethoden für Aufzählungen erstellen.....	283
Erweiterungsmethoden nutzen.....	284
3 Objektmengen verarbeiten.....	289
Spezielle Situationen für Objekte.....	290
Konvertieren von Datentypen	290
Kopieren von Objekten.....	298
Deterministische Kontrolle des Lebenszyklus.....	302
Vergleichen von Objekten	310
Ausnahmen beherrschen und nutzen	322
Objektmengen verwalten.....	340
Grundlagen der Auflistungsklassen	341
Grundlagen der generischen Auflistungsklassen.....	343
Die Qual der Wahl von Auflistungsklassen	344
Die konkreten Klassen	348
Auflistungen selber herstellen.....	393
Language Integrated Query – LINQ	401
Die verschiedenen Ebenen von LINQ	402
Grundlegende Syntax der Verwendung von LINQ	403
Lambda-Ausdrücke	405
Elemente von LINQ.....	408
4 Elementare Klassen der Bibliothek.....	457
Der Umgang mit Attributen	458
Anwendung von Attributklassen.....	459
Attribute abfragen.....	465
Eigene Attributklassen erstellen	469
Formatieren von Daten	472
Numerische Standardformate.....	473
Eigene numerische Formate.....	475
Standard-Datumsformate.....	476
Eigene Datumsformate	478
Formate von Aufzählungen.....	480

Umsetzung der Formate mit Sprachen	481
Eigene Formatierungen erstellen	482
Der Umgang mit der Zeit	485
Der Umgang mit dem Kalender	486
Mit Zeitpunkten arbeiten	491
Zeitzone	499
Die Stoppuhr	503
.NET und die Mathematik	506
Die Klasse Math	506
Präzision bei mathematischen Berechnungen	511
Der Zufallszahlengenerator	516
Die Struktur BigInteger	517
Die Struktur Complex	520
Die Systemumgebung erforschen	523
Die Klasse Environment	525
Die externen Speicher	527
Weitere Systeminformationen und das Bildschirmsystem	529
Informationen über die Netzwerkausrüstung	533
Auf die Windows-Registrierung zugreifen	536
Von Prozessen, Anwendungsdomänen und Assemblys	541
Informationen über laufende Prozesse abfragen	542
Einen Prozess starten	546
Einen anderen Prozess beenden	549
Einen Prozess unter einem anderen Anwenderkonto starten	550
Die Anwendungsdomäne, der Isolator innerhalb des Prozesses	552
Informationen über eine Anwendungsdomäne abfragen	553
Eine zusätzliche Anwendungsdomäne erstellen und nutzen	557
Anwendungen konfigurierbar gestalten	563
Das Projekt vorbereiten	563
Eigene Konfigurationswerte erstellen	565
Eigene Konfigurationswerte lesen und schreiben	566
Eigene Konfigurationswerte in Bibliotheken (DLL)	567
Diagnose und Debugging	570
Informationen für die Diagnose sammeln und interpretieren	571
Das Ereignisprotokoll von Windows nutzen	576
Den Ablauf einer Anwendung verfolgen	582
Der Garbage Collector	593
Funktionsweise des Garbage Collectors	593
Einflussnahme auf die Wirkungsweise des Garbage Collectors	596
Verhalten der GC in mehreren Prozessen	597

5	Der Umgang mit Dateien	601
	Arbeiten mit dem Dateisystem	602
	Informationen über Laufwerke abrufen	603
	Handhaben von Verzeichnissen	604
	Handhabung von Dateien	607
	Verwalten von Berechtigungen	608
	Das Dateisystem überwachen	610
	Datenströme und das E/A-Konzept von .NET	612
	Serialisierung von einfachen Textdaten	615
	Von Zeichen und Zeichencodierungen	615
	Mit statischen Methode der Klasse File arbeiten	621
	Arbeiten mit Klassen für Textdatenströme	622
	Binäre Serialisierung	623
	Manuelle binäre Serialisierung	624
	Automatische binäre Serialisierung	627
	XML-Serialisierung	638
	Manuelle XML-Serialisierung	639
	Automatische XML-Serialisierung	645
	Die Schnittstelle IXmlSerializable nutzen	653
	LINQ to XML	655
6	Auf Datenbanken zugreifen	663
	Relationales und objektorientiertes Modell	664
	Gerichtete Assoziation	665
	Bidirektionale Assoziation	666
	Gerichtete 1:m-Assoziation	666
	Bidirektionale 1:m-Assoziation	666
	n:m-Assoziation	667
	Attributierte Assoziation	667
	Aggregationen und Kompositionen	668
	Vererbung	668
	Andere Datenbanksysteme	671
	Architektur der .NET-Datenbankzugriffe	671
	Vergleich der verschiedenen Techniken	672
	Und wie steht es um die Geschwindigkeit der Techniken?	674
	Details in den Architekturüberlegungen	675
	Verwendete Namensräume	678
	Umgang mit dem Beispiel des Kapitels	679
	Installieren der Datenbanken	679
	Betrachten und Ändern von Dateninhalten	683
	Struktur der Beispiele	685
	Direkte Nutzung der Datenprovider	686
	Grundlagen der Datenveränderung	687
	Konfigurieren einer Anwendung für einen Datenbankzugriff	688

Herstellen von providerabhängigen Zugriffsobjekten.....	691
Das Verbindungsobjekt.....	692
Erstellen und Nutzen des Befehlsobjekts.....	696
Daten mit dem Datenreader lesen.....	701
Der Datenadapter und CommandBuilder.....	704
Das Transaktionsobjekt.....	707
Nutzen der Datenkonsumenten.....	711
Die Klassen DataSet und DataTable in der Übersicht.....	712
Datenprovider typisiert oder nicht typisiert benutzen.....	715
Erstellen eines typisierten Datenzugriffs.....	716
Nutzen typisierter Datasets.....	719
LINQ to SQL.....	726
Die Wirkungsweise von LINQ to SQL.....	727
Erstellen eines Datenbankzugriffs mit LINQ to SQL.....	731
Verfeinern der Vererbungsbeziehung.....	734
Nutzen von LINQ to SQL.....	735
Nutzen einer eigenen Transaktion.....	741
Entity Framework.....	741
Die Wirkungsweise des Entity Framework.....	742
Erstellen eines Modells für eine Datenbank.....	747
Erstellen einer Vererbungsbeziehung aus der Datenbank.....	751
Nutzen des Entity Framework.....	753
Nutzen einer eigenen Transaktion.....	759
7 Verarbeitungen parallelisieren.....	761
Von Prozessen und Threads.....	762
Wann parallelisieren wir Aufgaben.....	764
Wichtige Eigenschaften von Threads.....	766
Anatomie einer Threadmethode.....	769
Mittel für das Parallelisieren von Aufgaben.....	770
Anwendung mehrfach starten (Multitasking).....	772
Timer nutzen.....	773
Methoden asynchron aufrufen.....	775
Den Threadpool nutzen.....	778
Eigene Threads erstellen.....	780
Die Klasse BackgroundWorker nutzen.....	783
Parallele Programmierung.....	787
Von datenorientierter und aufgabenorientierter Verarbeitung.....	788
Nutzen der datenorientierten parallelen Verarbeitung.....	789
Nutzen aufgabenorientierter Verarbeitungen.....	799
Abbrechen einer parallelen Verarbeitung von außen.....	805
Fehlerbehandlung in der parallelen Verarbeitung.....	806
Client-Server-Anwendungen verwenden.....	808

Parallelisieren von LINQ-Abfragen.....	809
Umstellen einer Abfrage auf PLINQ.....	810
Steuern des Verhaltens von PLINQ.....	811
Abbrechen einer Abfrage.....	812
Parallele Verarbeitungen synchronisieren	814
Die Qual der Wahl.....	814
Grundregeln der Synchronisation.....	815
Synchronisieren mit sprachintegrierten Mitteln.....	818
Synchronisieren mit Mitteln der BCL	820
Die Nutzung der BCL für die parallele Verarbeitung	837
Nutzung von ICollection realisierenden Auflistungsklassen	837
Nutzung der Techniken für grafische Benutzerschnittstellen	838
Nutzung der Techniken für Client-Server-Anwendungen	838
Fallen der parallelen Programmierung.....	838
Vorsicht vor unbewusstem Multithreading.....	838
Parallelisieren lohnt sich nicht immer.....	839
Parallelisieren ist nicht immer schneller	839
In parallelen Verarbeitungen Aufrufe zu nicht threadsicheren Methoden verhindern.....	840
Den parallelen Overkill verhindern	840
Vorsicht vor threadaffinen Mechanismen	840
Wahl der richtigen Synchronisation.....	841
Multithreading und Visual Studio-Debugging	842
Das Verhalten von Visual Studio für das Debugging.....	842
Das Visual Studio-Fenster Threads.....	843
Das Visual Studio-Fenster Parallel Tasks.....	844
8 Reflexion	847
Grundlagen der Reflexion	848
Laden von Assemblys	851
Auf Elemente in einer Assembly zugreifen	854
Typen suchen und nutzen.....	856
Suchen von Elementen in Containern (BindingFlags)	859
Objekte mit Reflexion erstellen und Methoden nutzen.....	862
Felder und Eigenschaften mit Reflexion verarbeiten	866
Ereignisse mit Reflexion nutzen	868
Eine Assembly zur Laufzeit herstellen	869
Code zur Laufzeit kompilieren	871
Code aus einer Zeichenfolge erzeugen	874
Code mithilfe von CodeDom herstellen.....	875

9	Globalisierung und Lokalisierung von Anwendungen.....	887
	Herausforderungen der verschiedenen Kulturen	888
	Die Sprache.....	888
	Formatieren von Informationen	889
	Sortieren von Ausdrücken	889
	Aufpassen bei Bildern	890
	Farben verwenden.....	890
	Von Formaten und Darstellungen.....	890
	Voraussetzungen für die Globalisierung.....	891
	Verwenden Sie keine fest codierten Zeichenfolgen.....	891
	Vermeiden von festen Zeichenfolgenzusammensetzungen.....	891
	Einfache Sprache ohne Slang.....	892
	Textausgaben in der Benutzeroberfläche großzügig bemessen.....	892
	Symbole allgemein auslegen	892
	Formatierungen.....	892
	Datenformate	892
	Eingaben von Daten und Plausibilisierungen.....	893
	Spezifische Regeln der eingesetzten Technologie.....	893
	Grundlagen der Lokalisierung.....	893
	Die Klasse CultureInfo	894
	Kulturspezifische Formatierungen vornehmen.....	897
	Kulturspezifische Daten variieren.....	898
	Spezifische Verwendung des Ressourcenmanagers	906
	Sprachversionen von .NET.....	908
	Anhang A – Entwicklungswerkzeuge	909
	Werkzeuge von .NET	910
	Werkzeuge von Visual Studio	911
	Anhang B – Dokumentieren von Quellcode	913
	Übersicht über die XML-Tags für die Kommentierung	914
	Tag-Beispiele	915
	<summary>	915
	<remarks>.....	915
	<param>.....	916
	<paramref>	916
	<returns>	917
	<value>.....	917
	<exception>	918
	<c>	918
	<code>.....	919
	<example>	919
	<list>	919

<see>	921
<seealso>	922
<para>.....	922
<typeparam>.....	922
<typeparamref>.....	923
<include>	923
<permission>.....	924
Anhang C – Zeichensatz.....	925
UTF-8 Zeichensatz	926
Anhang D – Codierungen der Kulturen.....	937
Übersicht der Kulturnamen und LCID	938
Stichwortverzeichnis.....	945
Rolf Wenger, zur Person	959

