

Inhalt

Vorwort	29
----------------------	-----------

Teil I: Einführung in die VB-Programmierung

1	Einführung in die .NET-Technologie	35
1.1	Zur Historie von .NET	36
1.1.1	Vom alten VB zu VB.NET	36
1.1.2	Wie funktioniert eine .NET-Sprache?	37
1.2	Das .NET-Konzept	38
1.2.1	Das .NET-Framework	38
1.2.2	Die Common Language Specification (CLS)	39
1.2.3	Das Common Type System (CTS)	40
1.2.4	Die Common Language Runtime (CLR)	40
1.2.5	Die .NET-Klassenbibliothek	41
1.2.6	Das Konzept der Namespaces	41
1.2.7	Zugriff auf COM-Komponenten	42
1.2.8	Was sind Assemblys?	43
1.2.9	Metadaten und Manifest	43
1.2.10	Der Reflection-Mechanismus	44
1.2.11	Attribute	44
1.2.12	Serialisierung	45
1.2.13	Multithreading	45

1.3	Der Einstieg in Visual Studio	46
1.3.1	Ein erster Blick auf die IDE	46
1.3.2	Der Projektmappen-Explorer	48
1.3.3	Der Designer	49
1.3.4	Das Eigenschaften-Fenster	50
1.3.5	Das Codefenster	50
2	Visual Basic-Crashkurs für Anfänger	53
2.1	Es lebe der gute alte PAP!	54
2.2	Ab ins Regal!	57
2.3	Aufhören oder weitermachen?	59
2.4	Jedes Programm lässt sich verbessern!	62
2.5	Schluss mit langweiliger Konsole!	65
2.6	Darf es auch etwas anspruchsvoller sein?	68
2.7	Hilfe, mein Programm liefert falsche Ergebnisse!	72
3	Einführung in Visual Basic	75
3.1	Grundlagen der VB-Syntax	76
3.1.1	Anweisungen	76
3.1.2	Bezeichner	76
3.1.3	Kommentare	77
3.1.4	Zeilenumbruch	78
3.2	Datentypen und Variablen	78
3.2.1	Fundamentale Typen	79
3.2.2	Namensgebung von Variablen	79
3.2.3	Deklaration von Variablen	80
3.2.4	Mehrfachdeklaration	80
3.2.5	Anfangswerte zuweisen	81
3.2.6	Option Explicit	81
3.2.7	Option Strict	82
3.2.8	Konstanten deklarieren	83
3.2.9	Gültigkeitsbereiche von Deklarationen	83
3.2.10	Lokale Variablen mit Dim	83
3.2.11	Lokale Variablen mit Static	84
3.2.12	Private globale Variablen	85
3.2.13	Public Variablen	85
3.3	Einfache Datentypen im Überblick	85
3.3.1	Byte, Short, Integer und Long	85

3.3.2	Single, Double und Decimal	86
3.3.3	Char und String	87
3.3.4	Date	87
3.3.5	Boolean	88
3.3.6	Object	89
3.4	Weitere wichtige Sprachfeatures	89
3.4.1	Nullable Types	89
3.4.2	Typinferenz	90
3.5	Konvertieren von Datentypen	91
3.5.1	Implizite und explizite Konvertierung	91
3.5.2	Welcher Datentyp passt in welchen?	92
3.5.3	Spezielle Konvertierungsfunktionen	93
3.5.4	Die CType-Funktion	94
3.5.5	Die Convert-Klasse	94
3.5.6	Die Parse-Methode	95
3.6	Operatoren	95
3.6.1	Arithmetische Operatoren	96
3.6.2	Zuweisungsoperatoren	96
3.6.3	Logische Operatoren	97
3.6.4	Vergleichsoperatoren	98
3.6.5	Rangfolge der Operatoren	98
3.7	Kontrollstrukturen	99
3.7.1	Verzweigungsbefehle	99
3.7.2	Schleifenanweisungen	102
3.8	Komplexe Datentypen	103
3.8.1	Aufzählungstypen	103
3.8.2	Strukturen	104
3.9	Nutzerdefinierte Funktionen/Prozeduren	107
3.9.1	Deklaration und Syntax	107
3.9.2	Parameterübergabe allgemein	109
3.9.3	Übergabe mit ByVal und ByRef	110
3.9.4	Optionale Parameter	111
3.9.5	Überladene Funktionen/Prozeduren	111
4	Objektorientiertes Programmieren	113
4.1	Strukturierte versus objektorientierte Programmierung	114
4.1.1	Was bedeutet strukturiert?	114
4.1.2	Was heißt objektorientiert?	115

4.2	Grundbegriffe der OOP	115
4.2.1	Objekt, Klasse, Instanz	116
4.2.2	Kapselung und Wiederverwendbarkeit	116
4.2.3	Vererbung und Polymorphie	117
4.2.4	Sichtbarkeit von Klassen und ihren Mitgliedern	117
4.2.5	Allgemeiner Aufbau einer Klasse	118
4.3	Ein Objekt erzeugen	120
4.3.1	Referenzieren und Instanzieren	120
4.3.2	Klassische Initialisierung	121
4.3.3	Objekt-Initialisierer	121
4.3.4	Arbeiten mit dem Objekt	122
4.3.5	Zerstören des Objekts	122
4.4	DEMO: OOP-Einführung	123
4.4.1	Vorbereitungen	123
4.4.2	Klasse definieren	124
4.4.3	Objekt erzeugen und initialisieren	124
4.4.4	Objekt verwenden	125
4.4.5	Intellisense – die hilfreiche Fee	125
4.4.6	Objekt testen	126
4.4.7	Warum unsere Klasse noch nicht optimal ist	126
4.5	Eigenschaften	127
4.5.1	Eigenschaften kapseln	127
4.5.2	Eigenschaften mit Zugriffsmethoden berechnen	129
4.5.3	Lese-/Schreibschutz für Eigenschaften	131
4.5.4	Statische Eigenschaften	131
4.6	Methoden	132
4.6.1	Öffentliche und private Methoden	132
4.6.2	Überladene Methoden	133
4.6.3	Statische Methoden	134
4.7	Ereignisse	135
4.7.1	Ereignis deklarieren	136
4.7.2	Ereignis auslösen	136
4.7.3	Ereignis auswerten	137
4.7.4	Benutzerdefinierte Ereignisse (Custom Events)	138
4.8	Arbeiten mit Konstruktor und Destruktor	141
4.8.1	Der Konstruktor erzeugt das Objekt	141
4.8.2	Bequemer geht's mit einem Objekt-Initialisierer	143
4.8.3	Destruktor und Garbage Collector räumen auf	144

4.9	DEMO: Vererbung und Polymorphie	146
4.9.1	Vererbungsbeziehungen im Klassendiagramm	147
4.9.2	Überschreiben von Methoden (Method-Overriding)	148
4.9.3	Klassen implementieren	148
4.9.4	Objekte implementieren	153
4.9.5	Polymorphe Methoden	154
4.9.6	Allgemeine Hinweise und Regeln zur Vererbung	157
4.10	Besondere Klassen und Features	158
4.10.1	Abstrakte Klassen	158
4.10.2	Abstrakte Methoden	158
4.10.3	Versiegelte Klassen	159
4.10.4	Partielle Klassen	160
4.10.5	Die Basisklasse System.Object	161
4.10.6	Property-Accessors	162
4.11	Schnittstellen (Interfaces)	163
4.11.1	Wozu braucht man Schnittstellen?	163
4.11.2	DEMO: Variante 1 (mit Implementierungsvererbung)	163
4.11.3	DEMO: Variante 2 (mit Schnittstellenvererbung)	165
4.11.4	Abfragen, ob Schnittstelle vorhanden ist	167
4.11.5	Mehrere Schnittstellen implementieren	167
4.11.6	Wichtiges zur Schnittstellenprogrammierung	167
5	Weitere VB-Sprachfeatures	169
5.1	Namespaces (Namensräume)	170
5.1.1	Ein kleiner Überblick	170
5.1.2	Die Imports-Anweisung	171
5.1.3	Namespace-Alias	172
5.1.4	Namespaces in Projekteigenschaften	172
5.1.5	Namespace Alias Qualifizierer	173
5.1.6	Eigene Namespaces einrichten	174
5.2	Datenfelder (Arrays)	175
5.2.1	Ein Array deklarieren	175
5.2.2	Zugriff auf Array-Elemente	176
5.2.3	Oberen Index ermitteln	176
5.2.4	Explizite Arraygrenzen	176
5.2.5	Arrays initialisieren	176
5.2.6	Zugriff mittels Schleife	177
5.2.7	Mehrdimensionale Arrays	177

5.2.8	Dynamische Arrays	178
5.2.9	Zuweisen von Arrays	180
5.2.10	Arrays aus Strukturvariablen	181
5.2.11	Löschen von Arrays	182
5.2.12	Eigenschaften und Methoden von Arrays	182
5.2.13	Übergabe von Arrays	184
5.3	Zeichenkettenverarbeitung	185
5.3.1	Strings zuweisen	185
5.3.2	Eigenschaften und Methoden eines Strings	187
5.3.3	Kopieren eines Strings in ein Char-Array	189
5.3.4	Wichtige (statische) Methoden der String-Klasse	189
5.4	Zahlen formatieren	191
5.4.1	Die ToString-Methode	192
5.4.2	Die Format-Methode	193
5.5	Vordefinierten Funktionen	195
5.5.1	Mathematik	195
5.5.2	Datums- und Zeitfunktionen	197
5.6	Überladen von Operatoren	199
5.6.1	Syntaxregeln	199
5.6.2	DEMO: Addition von Arrays	200
5.6.3	Konvertierungsoperatoren überladen	201
5.7	Auflistungen (Collections)	202
5.7.1	Beziehungen zwischen den Schnittstellen	202
5.7.2	IEnumerable	202
5.7.3	ICollection	203
5.7.4	ICollection	204
5.7.5	Die ArrayList-Collection	204
5.7.6	Die Hashtable	205
5.8	Generische Datentypen	206
5.8.1	Wie es früher war	207
5.8.2	Typsicherheit durch Generics	208
5.8.3	List-Collection ersetzt ArrayList	210
5.8.4	Über die Vorzüge generischer Collections	211
5.8.5	Typbeschränkungen durch Constraints	212
5.8.6	Generische Methoden	212
5.9	Was sind Delegates?	213
5.9.1	Delegates sind Methodenzeiger	213
5.9.2	Delegate-Typ definieren	214

5.9.3	Delegate-Objekt erzeugen	214
5.9.4	Delegates vereinfacht instanziiieren	215
5.9.5	Automatische Delegate-Typ-Konvertierung	215
5.9.6	Lambda-Ausdrücke	216
6	Einführung in LINQ	219
6.1	LINQ-Grundlagen	220
6.1.1	Die LINQ-Architektur	220
6.1.2	LINQ-Implementierungen	220
6.1.3	Anonyme Typen	221
6.1.4	Erweiterungsmethoden	222
6.2	LINQ-Abfragen	224
6.2.1	Grundlegendes zur LINQ-Syntax	224
6.2.2	Zwei alternative Schreibweisen von LINQ Abfragen	225
6.2.3	Übersicht der wichtigsten Abfrage-Operatoren	226
6.3	DEMO: LINQ to Objects	227
6.3.1	Die Projektionsoperatoren Select und SelectMany	228
6.3.2	Der Restriktionsoperator Where	230
6.3.3	Die Sortierungsoperatoren OrderBy und ThenBy	230
6.3.4	Der Gruppierungsoperator GroupBy	232
6.3.5	Verknüpfen mit Join	233
6.3.6	Aggregat-Operatoren	234
6.3.7	Verzögertes Ausführen von LINQ-Abfragen	236
6.3.8	Konvertierungsmethoden	237

Teil II: Bausteine komplexer Anwendungen

7	Kommandozeilenanwendungen	241
7.1	Grundaufbau/Konzepte	242
7.1.1	Unser Hauptprogramm – Module1.vb	243
7.1.2	Rückgabe eines Fehlerstatus	244
7.1.3	Parameterübergabe	245
7.1.4	Zugriff auf die Umgebungsvariablen	246
7.2	Die Kommandozentrale: System.Console	247
7.2.1	Eigenschaften	247
7.2.2	Methoden/Ereignisse	248
7.2.3	Textausgaben	248
7.2.4	Farbangaben	249

7.2.5	Tastaturabfragen	250
7.2.6	Arbeiten mit Streamdaten	252
7.2.7	DEMO: Farbige Konsolenanwendung	253
7.2.8	Weitere nützliche Informationen zur Console-Klasse	255
8	Einführung in Windows Forms-Anwendungen	257
8.1	Grundaufbau/Konzepte	258
8.1.1	Wo ist das Hauptprogramm?	259
8.1.2	Die Oberflächendefinition – Form1.Designer.vb	263
8.1.3	Die Spielwiese des Programmierers – Form1.vb	264
8.1.4	Die Datei AssemblyInfo.vb	265
8.1.5	Resources.resx/Resources.Designer.vb	266
8.1.6	Settings.settings/Settings.Designer.vb	267
8.1.7	Settings.vb	269
8.2	Ein Blick auf die Application-Klasse	270
8.2.1	Eigenschaften	270
8.2.2	Methoden	271
8.2.3	Ereignisse	272
8.3	Allgemeine Eigenschaften von Komponenten	273
8.3.1	Übersicht	273
8.3.2	Font	274
8.3.3	Handle	276
8.3.4	Tag	276
8.3.5	Modifiers	276
8.4	Allgemeine Ereignisse von Komponenten	277
8.4.1	Die Eventhandler-Argumente	277
8.4.2	Das sender-Objekt	278
8.4.3	Der Parameter e	279
8.4.4	Mausereignisse	279
8.4.5	Tastaturereignisse	280
8.4.6	Wozu brauchen wir KeyPreview?	282
8.4.7	Weitere Ereignisse	282
8.4.8	Validitätsprüfungen	283
8.5	Allgemeine Methoden von Komponenten	284
9	Windows-Formulare verwenden	285
9.1	Übersicht zur Form-Klasse	286
9.1.1	Wichtige Eigenschaften	286

9.1.2	Wichtige Methoden	288
9.1.3	Wichtige Ereignisse	289
9.2	Praktische Aufgabenstellungen	290
9.2.1	Ein Fenster anzeigen	290
9.2.2	Einen Splash Screen beim Anwendungsstart anzeigen	293
9.2.3	Eine Sicherheitsabfrage vor dem Schließen anzeigen	295
9.2.4	Ein Formular durchsichtig machen	296
9.2.5	Die Tabulatorreihenfolge festlegen	297
9.2.6	Ausrichten und Platzieren von Komponenten im Formular	297
9.2.7	Spezielle Panels für flexibles Layout	300
9.2.8	Menüs erzeugen	301
9.3	MDI-Anwendungen	304
9.3.1	Unsere Verwaltungszentrale: Das MDI-Hauptfenster	305
9.3.2	Die Kindfenster	305
9.3.3	Automatisches Anordnen der Kindfenster	307
9.3.4	Zugriff auf die geöffneten MDI-Kindfenster	307
9.3.5	Zugriff auf das aktive MDI-Kindfenster	308
9.3.6	Kombinieren der Kindfenstermenüs mit dem MDIContainer	308
10	Wichtige Windows Forms-Komponenten	309
10.1	Grundlegende Techniken	310
10.1.1	Hinzufügen von Komponenten	310
10.1.2	Komponenten zur Laufzeit erzeugen	311
10.2	Allgemeine Steuerelemente	312
10.2.1	Label	312
10.2.2	LinkLabel	313
10.2.3	Button	314
10.2.4	TextBox	315
10.2.5	MaskedTextBox	318
10.2.6	CheckBox	319
10.2.7	RadioButton	320
10.2.8	ListBox	321
10.2.9	CheckedListBox	322
10.2.10	ComboBox	323
10.2.11	PictureBox	324
10.2.12	DateTimePicker	324
10.2.13	MonthCalendar	325
10.2.14	HScrollBar, VScrollBar	325

10.2.15	TrackBar	326
10.2.16	NumericUpDown	327
10.2.17	DomainUpDown	328
10.2.18	ProgressBar	328
10.2.19	RichTextBox	329
10.2.20	ListView	330
10.2.21	TreeView	335
10.2.22	WebBrowser	340
10.3	Container	341
10.3.1	FlowLayout/TableLayout/SplitContainer	341
10.3.2	Panel	341
10.3.3	GroupBox	342
10.3.4	TabControl	342
10.3.5	ImageList	344
10.4	Menüs und Symbolleisten	345
10.4.1	MenuStrip und ContextMenuStrip	345
10.4.2	ToolStrip	345
10.4.3	StatusStrip	346
10.4.4	ToolStripContainer	346
11	Grafikausgabe und Drucken	347
11.1	Grundlegende Konzepte	348
11.1.1	Die Grafikzentrale: GDI+	348
11.1.2	Wo finde ich was?	349
11.2	Grafikanzeige und -manipulation	350
11.2.1	Grafikanzeige mit der PictureBox	351
11.2.2	Die Image-Klasse	352
11.2.3	Grafiken zur Laufzeit zuweisen	353
11.2.4	Grafiken in verschiedenen Formaten sichern	353
11.2.5	Wichtige Grafikeigenschaften ermitteln	354
11.2.6	Erzeugen von Vorschaugrafiken (Thumbnails)	354
11.2.7	Grafiken drehen	355
11.2.8	Skalieren von Grafiken	355
11.3	Koordinatensystem von GDI+	356
11.3.1	Grundsätzlicher Aufbau	356
11.3.2	Globale Koordinaten	357
11.3.3	Seitenkoordinaten (globale Transformation)	358
11.3.4	Gerätekoordinaten (Seitentransformation)	359

11.4	Wichtige Grafikmethoden	360
11.4.1	Das zentrale Graphics-Objekt	360
11.4.2	Punkte zeichnen/abfragen	361
11.4.3	Linien	362
11.4.4	Kantenglättung mit Antialiasing	363
11.4.5	PolyLine	364
11.4.6	Rechtecke	364
11.4.7	Polygone	365
11.4.8	Splines	366
11.4.9	Bézierkurven	367
11.4.10	Kreise und Ellipsen	368
11.4.11	Tortenstück (Segment)	368
11.4.12	Bogenstück	370
11.4.13	Textausgabe	371
11.4.14	Ausgabe von Grafiken	374
11.5	Die wichtigsten Grafikobjekte	375
11.5.1	Einfache Objekte	375
11.5.2	Vordefinierte Objekte	376
11.5.3	Farben/Transparenz	378
11.5.4	Stifte (Pen)	379
11.5.5	Pinsel (Brush)	381
11.5.6	SolidBrush	381
11.5.7	HatchBrush	381
11.5.8	TextureBrush	382
11.5.9	LinearGradientBrush	383
11.5.10	PathGradientBrush	384
11.5.11	Fonts	385
11.6	Die Grafik-Standarddialoge	386
11.6.1	FontDialog für die Schriftauswahl	386
11.6.2	Farbauswahl mit ColorDialog	387
11.7	Druckausgabe/Druckvorschau	388
11.7.1	Grundkonzept	389
11.7.2	Programmiermodell	390
11.7.3	Komponenten für die Druckausgabe	391
11.7.4	Praktische Aufgabenstellungen	391
11.7.5	Die Druckdialoge	394
11.7.6	Ein eigenes Druckvorschau-Fenster	397
11.7.7	DEMO: Komplette Druckausgabe	398

12	Zugriff auf das Dateisystem	409
12.1	Grundlagen	410
12.1.1	Das Datei-System von Windows	410
12.1.2	Klassen für Verzeichnis- und Dateioperationen	410
12.1.3	Statische versus Instanzen-Klassen	411
12.2	Operationen auf Verzeichnisebene	412
12.2.1	Verzeichnisse erzeugen und löschen	412
12.2.2	Verzeichnisse umbenennen und verschieben	413
12.2.3	Aktuelles Verzeichnis bestimmen	413
12.2.4	Unterverzeichnisse ermitteln	413
12.2.5	Alle Laufwerke ermitteln	414
12.2.6	Im Verzeichnis enthaltene Dateien ermitteln	415
12.2.7	Dateien kopieren und verschieben	415
12.2.8	Dateien umbenennen	416
12.2.9	Dateiattribute feststellen	416
12.2.10	Die FileAttribute-Enumeration	417
12.3	Mehr zur FileInfo-Klasse	418
12.3.1	Weitere wichtige Eigenschaften	418
12.3.2	GetFileSystemInfos-Methode	418
12.4	Zugriffsberechtigungen	419
12.4.1	ACL und ACE	419
12.4.2	SetAccessControl-Methode	419
12.4.3	Zugriffsrechte anzeigen	420
12.5	Weitere wichtige Klassen	421
12.5.1	Die Path-Klasse	421
12.5.2	Die Klasse FileSystemWatcher	422
13	Dateien lesen und schreiben	425
13.1	Das Prinzip der Datenpersistenz	426
13.1.1	Dateien und Streams	426
13.1.2	Die wichtigsten Klassen	426
13.1.3	Erzeugen eines Streams	427
13.2	Dateiparameter	427
13.2.1	Die FileAccess-Enumeration	428
13.2.2	Die FileMode-Enumeration	428
13.2.3	Die FileShare-Enumeration	428
13.3	Textdateien	429
13.3.1	Eine Textdatei beschreiben bzw. neu anlegen	429

13.3.2	Eine Textdatei lesen	430
13.4	Binärdateien	430
13.4.1	Lese-/Schreibzugriff	430
13.4.2	Die Methoden ReadAllBytes/WriteAllBytes	431
13.4.3	Varianten zum Erzeugen von BinaryReader/BinaryWriter	431
13.5	Sequenzielle Dateien	432
13.5.1	Lesen und schreiben von strukturierten Daten	432
13.5.2	Serialisieren von Objekten	433
13.6	Dateien verschlüsseln und komprimieren	434
13.6.1	Das Methodenpärchen Encrypt-/Decrypt	434
13.6.2	Verschlüsseln unter Windows XP/Vista	434
13.6.3	Verschlüsseln mittels der CryptoStream-Klasse	435
13.6.4	Dateien komprimieren	436
13.7	Dateidialoge	437
13.7.1	Anzeige und Auswertung	438
13.7.2	Wichtige Eigenschaften	438
13.7.3	Verwenden von Dateifiltern	439
14	Einführung in ADO.NET	441
14.1	Eine Übersicht	442
14.1.1	Die ADO.NET-Klassenhierarchie	442
14.1.2	Die Klassen der Datenprovider	443
14.1.3	Beziehungen zwischen den ADO.NET-Klassen	445
14.1.4	DEMO: ADO.NET-Objekte im Einsatz	446
14.2	Das Connection-Objekt	448
14.2.1	Allgemeiner Aufbau	448
14.2.2	OleDbConnection	448
14.2.3	Schließen einer Verbindung	449
14.2.4	Eigenschaften des Connection-Objekts	450
14.2.5	Methoden des Connection-Objekts	452
14.2.6	Der ConnectionStringBuilder	453
14.3	Das Command-Objekt	453
14.3.1	Erzeugen und Anwenden eines Command-Objekts	453
14.3.2	Erzeugen mittels CreateCommand-Methode	454
14.3.3	Eigenschaften des Command-Objekts	454
14.3.4	DEMO: Eine Auswahlabfrage absetzen	456
14.3.5	Methoden des Command-Objekts	459
14.3.6	DEMO: Eine Aktionsabfrage ausführen	460

14.4	Parameter-Objekte	462
14.4.1	Erzeugen und Anwenden eines Parameter-Objekts	462
14.4.2	Eigenschaften des Parameter-Objekts	463
14.5	Das CommandBuilder-Objekt	464
14.5.1	Erzeugen	464
14.5.2	Anwenden	464
14.6	Das DataReader-Objekt	465
14.6.1	DataReader erzeugen	465
14.6.2	Daten lesen	465
14.6.3	Eigenschaften DataReaders	466
14.6.4	Methoden des DataReaders	467
14.7	Das DataAdapter-Objekt	467
14.7.1	DataAdapter erzeugen	468
14.7.2	Command-Eigenschaften	469
14.7.3	Fill-Methode	469
14.7.4	Update-Methode	470
14.7.5	DEMO: Die Datenbank aktualisieren	471
15	Das DataSet	475
15.1	Grundlagen	475
15.1.1	Die wichtigsten Klassen	476
15.1.2	Die Objekthierarchie	476
15.1.3	Erzeugen eines DataSets	477
15.2	Die DataTable	479
15.2.1	DataTable erzeugen	479
15.2.2	Spalten hinzufügen	480
15.2.3	Zeilen zur DataTable hinzufügen	480
15.2.4	Zugriff auf den Inhalt einer DataTable	481
15.3	Die DataView	483
15.3.1	Erzeugen einer DataView	484
15.3.2	Sortieren und Filtern von Datensätzen	484
15.3.3	DEMO: Im DataView sortieren/filtern	484
15.3.4	Suche von Datensätzen	486
15.3.5	DEMO: Datensätze suchen	487
15.4	Typisierte DataSets	488
15.4.1	Was ist ein typisiertes DataSet?	488
15.4.2	Das Konzept der Datenquellen	490
15.4.3	Typisierte DataSets und TableAdapter	490

16	Datenbindung unter ADO.NET	493
16.1	Allgemeine Varianten der Datenbindung	494
16.1.1	Manuelle Datenbindung an einfache Datenfelder	494
16.1.2	Manuelle Datenbindung an Listen und Tabelleninhalte	496
16.1.3	DEMO: Master-Detailbeziehungen im DataGrid	497
16.2	Spezielle Datenbindungen	499
16.2.1	Entwurfszeit-Datenbindung an ein typisiertes DataSet	499
16.2.2	Drag & Drop-Datenbindung	500
16.2.3	Navigieren im DataSet	500
16.2.4	Die Anzeige formatieren	502
16.2.5	DEMO: Arbeiten mit einer Datenquelle	503
 Teil III: Methoden und Werkzeuge der Softwareentwicklung		
17	Fehlersuche/-Fehlerbehandlung	509
17.1	Fehlersuche	509
17.1.1	Der Debugger	510
17.1.2	Einzelschritt-Modus	513
17.1.3	Prozedurschritt-Modus	514
17.1.4	DEMO: Wichtige Debugging-Möglichkeiten	514
17.1.5	Das Debug-Objekt	518
17.2	Fehlerbehandlung	519
17.2.1	Fehlermöglichkeiten	519
17.2.2	Der Try-Catch-Block	519
17.2.3	Der Try-Finally-Block	523
17.2.4	Das Standardverhalten bei Ausnahmen festlegen	525
17.2.5	Die Exception-Klasse	526
17.2.6	Fehler/Ausnahmen auslösen	527
17.2.7	Eigene Fehlerklassen entwickeln	527
17.2.8	Exceptions zur Entwurfszeit behandeln	529
18	Testprojekte	531
18.1	Testgetriebene Entwicklung (TDD)	532
18.1.1	Konventionelle Vorgehensweise	532
18.1.2	Testgetriebene Entwicklung	532
18.1.3	Unit-Tests unter Visual Studio	533

18.2	Einfache Tests	533
18.2.1	DEMO: Test einer Klasse CKugel	533
18.2.2	Eigene Testmethoden hinzufügen	539
18.3	Datengetriebene Tests (DDT)	540
18.3.1	Das DDT-Prinzip	540
18.3.2	DEMO: Test von Spesenberechnungen	540
18.4	Begriffe und Ergänzungen	545
18.4.1	Behauptungen (Asserts)	545
18.4.2	Der Testkontext	545
18.4.3	Zusätzliche Testattribute	547
18.5	Praktische Einsatzkriterien	547
18.5.1	Nutzen von Unit-Tests	547
18.5.2	Grenzen von Unit-Tests	548
19	Arbeiten mit dem Klassendesigner	549
19.1	Der Klassendesigner im Überblick	550
19.1.1	Wie komme ich zu einem Klassendiagramm?	550
19.1.2	Ein Blick in die Toolbox	551
19.2	Die Bausteine des Klassen-Designers	552
19.2.1	Enumeration	552
19.2.2	Klasse	554
19.2.3	Struktur	556
19.2.4	Abstrakte Klasse	557
19.2.5	Schnittstelle	558
19.2.6	Delegate	560
19.2.7	Zuordnung (Assoziation)	562
19.2.8	Vererbung anzeigen	562
19.2.9	Die Anzeige der Diagramme anpassen	562
19.3	DEMO: Modellierung des Bestellsystems einer Firma	563
19.3.1	Unser Ziel	563
19.3.2	Die Klasse CBestellung	565
19.3.3	Die Klasse CPerson	566
19.3.4	Die Klasse CKunde	567
19.3.5	Die Klasse CFirma	569
19.3.6	Die Klasse Form1	571
19.3.7	Abschlusstest	573

19.4	Mehr zum Klassen-Designer	574
19.4.1	Diagramme anpassen	575
19.4.2	Das Objekt-Testcenter	575
19.4.3	Wann lohnt sich der Einsatz des Klassen-Designers?	577
20	Das Microsoft Event Pattern	579
20.1	Das Subjekt-Observer-Entwurfsmuster	580
20.1.1	Warum ist das Observer-Pattern besonders wichtig?	580
20.1.2	Grundprinzip des Observer-Pattern	580
20.2	DEMO: Personalverwaltung	581
20.2.1	Die Datenstruktur	581
20.2.2	Die Ereignisdefinition	582
20.2.3	Die Ereignislogik	584
20.2.4	Das User-Interface	586
20.2.5	Die Datenpersistenz	589
20.2.6	Die Verwaltung des Pattern	590
20.2.7	Test und Diskussion	592
20.2.8	Ein zweiter Observer – kein Problem!	594
20.2.9	Zusammenfassung	594
21	Verteilte Anwendungen in Microkernel-Architektur	597
21.1	Grundkonzepte von Webdiensten	598
21.1.1	Zur Bedeutung von Webdiensten	598
21.1.2	Kommunikation per SOAP	598
21.1.3	Die Sprache WSDL	599
21.1.4	Zum Funktionsprinzip von Webdiensten	599
21.1.5	Anforderungen an eine Webklasse	601
21.1.6	Schnittstellen-Kompatibilität	601
21.2	Grundkonzept unseres Webdienstes	602
21.2.1	Datenaustausch	602
21.2.2	Datenbankstruktur	603
21.3	Klassische Realisierung	604
21.3.1	Klassendiagramm	604
21.3.2	Webmethoden	604
21.4	Realisierung in Microkernel-Architektur	606
21.4.1	UML-Klassendiagramm	606
21.4.2	Sequenzdiagramm	607
21.4.3	Die Klasse BookService als Microkernel	608

21.4.4	Die Klasse CCommon	609
21.4.5	Die internen Server	610
21.4.6	Ein erster Test	615
21.5	Webdienst-Client	616
21.5.1	Das User-Interface	616
21.5.2	Die Programmierung	617
21.5.3	Abschlusstest	623
21.5.4	Fehlerbeseitigung	625
21.5.5	Schlussbemerkungen	626
22	Verwenden von Ressourcen	627
22.1	Manifestressourcen	628
22.1.1	Erstellen von Manifestressourcen	628
22.1.2	Zugriff auf Manifestressourcen	629
22.2	Typisierte Ressourcen	631
22.2.1	Erzeugen von .resources-Dateien	631
22.2.2	Hinzufügen der .resources-Datei zum Projekt	631
22.2.3	Zugriff auf die Inhalte von .resources-Dateien	632
22.2.4	ResourceManager direkt aus der .resources-Datei erzeugen	632
22.2.5	Was sind .resx-Dateien?	633
22.3	Streng typisierte Ressourcen	633
22.3.1	Erzeugen streng typisierter Ressourcen	634
22.3.2	Verwenden streng typisierter Ressourcen	634
22.3.3	Streng typisierte Ressourcen per Reflection auslesen	635
22.4	Lokalisierte Anwendungen	637
22.4.1	Localizable und Language	637
22.4.2	DEMO: Landesfahnen	637
22.4.3	Einstellen der aktuellen Kultur zur Laufzeit	640
23	Verteilen von Anwendungen	643
23.1	Übersicht	644
23.1.1	Was ist ClickOnce-Deployment?	644
23.1.2	Was sind Setup-Projekte?	645
23.2	DEMO: ClickOnce-Deployment	645
23.2.1	Ort der Veröffentlichung	645
23.2.2	Anwendungsdateien	646
23.2.3	Erforderliche Komponenten	647
23.2.4	Aktualisierungen	647

23.2.5	Veröffentlichen	648
23.2.6	Verzeichnisstruktur	649
23.2.7	Der Webpublishing-Assistent	650
23.2.8	Neue Versionen erstellen	651
23.3	Demo: Setup-Projekt	651
23.3.1	Ein neues Setup-Projekt	651
23.3.2	Dateisystem-Editor	653
23.3.3	Ein erster Test	655
23.3.4	Registrierungs-Editor	655
23.3.5	Dateityp-Editor	656
23.3.6	Benutzeroberflächen-Editor	657
23.3.7	Editor für Startbedingungen	659
23.3.8	Finaler Test	660

Teil IV: Rezepte/Lösungen

24	Wie kann ich ... (Sprache/OOP)	665
24.1	... Code von C# nach Visual Basic portieren?	665
24.1.1	Die augenfälligsten Unterschiede	666
24.1.2	Datentypen	667
24.1.3	Operatoren	668
24.1.4	Verzweigungen	668
24.1.5	Schleifen	669
24.1.6	Arrays	669
24.1.7	Strukturen	670
24.1.8	Enumerationen	670
24.1.9	Funktionen, Prozeduren, Methoden	670
24.1.10	Klassendefinition	671
24.1.11	Erzeugen eines Objekts	672
24.1.12	Ereignis definieren und auslösen	672
24.1.13	Ereignis mit Ereignishandler verbinden	672
24.2	... einen String in ein Array kopieren?	673
24.3	... ein Byte-Array in einen String konvertieren?	675
24.4	... Strukturvariablen in Arrays einsetzen?	676
24.5	... eine einzelne Spalte aus einer Matrix kopieren?	679
24.6	... in einer ArrayList suchen und sortieren?	680
24.7	... in einer generischen Liste suchen und sortieren?	682
24.8	... mit Bubblesort sortieren?	684

24.9 ... Zufallszahlen erzeugen?	686
24.10 ... Iterationen verstehen?	687
24.11 ... den Goldenen Schnitt ermitteln?	690
24.12 ... Funktionen rekursiv aufrufen?	691
24.13 ... Zeitmessungen durchführen?	692
24.14 ... Strings vergleichen?	696
24.15 ... Datumsdifferenzen ermitteln?	699
24.16 ... das Alter in Jahren bestimmen?	701
24.17 ... die Monatsdifferenz berechnen?	703
24.18 ... das Datum beweglicher Feiertage berechnen?	704
24.19 ... ersten und letzten Wochentag eines Monats bestimmen?	706
24.20 ... Abschreibungen auf Monatsbasis berechnen?	707
24.21 ... Geldbeträge kaufmännisch runden?	711
24.22 ... Fehler bei mathematischen Operationen behandeln?	712
24.23 ... mit Potenzen und Wurzeln rechnen?	715
24.24 ... überladene/überschriebene Methoden unterscheiden?	717
24.25 ... Delegates verstehen?	720
24.26 ... LINQ-Abfragen verstehen?	722
24.27 ... Eigenschaften sinnvoll kapseln?	724
24.28 ... Aggregation und Vererbung unterscheiden?	727
25 Wie kann ich ... (Oberfläche/Komponenten)	735
25.1 ... den Inhalt des UI sichern?	735
25.2 ... die Anzeige löschen?	739
25.3 ... die Maus abfragen?	740
25.4 ... Dezimal komma in Dezimalpunkt umwandeln?	742
25.5 ... mit der TextBox arbeiten?	743
25.6 ... in einer TextBox suchen?	745
25.7 ... die ListBox kennen lernen?	746
25.8 ... RadioButtons und CheckBoxen einsetzen?	749
25.9 ... Objekte in ListBox und ComboBox anzeigen?	751
25.10 ... zur Laufzeit ein Steuerelement erzeugen?	753
25.11 ... Eingaben validieren?	755
25.12 ... ein Graphics-Objekt erzeugen?	757
25.12.1 Variante 1: Verwendung des Paint-Events	758
25.12.2 Variante 2: Überschreiben der OnPaint-Methode	758

25.12.3	Variante 3: Graphics-Objekt mit CreateGraphics erzeugen	759
25.12.4	Variante 4: Graphics-Objekts einer PictureBox nutzen	760
25.13	... Texte gedreht ausgeben?	760
25.14	... einen Markierungsrahmen erzeugen?	762
25.15	... mit Drag & Drop arbeiten?	763
25.16	... eine Komponente zur Farbauswahl entwickeln?	767
26	Wie kann ich ... (Sonstiges)	773
26.1	... die Zwischenablage verwenden?	773
26.2	... mittels Reflection Typinformationen sammeln?	776
26.3	... den mehrfachen Anwendungsstart verhindern?	780
26.4	... eine Pause realisieren?	781
26.5	... Systemtöne und WAV-Dateien wiedergeben?	783
26.6	... diverse Systeminformationen ermitteln?	785
26.6.1	Betriebssystem (Name, Version, Bootmode)	786
26.6.2	Schriftarten/-Informationen	787
26.6.3	Bildschirme	787
26.6.4	Environment Variablen auslesen	788
26.6.5	Netzwerk (User-Name, PC-Name ...)	789
26.6.6	Hardware-Informationen	790
26.6.7	Energiestatus	791
26.6.8	Anwendung (Pfad, Name, Assembly)	791
26.6.9	Soundkarte(n)	792
26.6.10	CLR-Version	793

Teil V: Komplexbeispiele

27	Kleine Textverarbeitung	797
27.1	Oberflächenentwurf	798
27.1.1	Das MDI-Rahmenfenster	798
27.1.2	MDI-Kindfenster	799
27.1.3	Menüs zusammenführen	800
27.1.4	Ein PopUp-Menü hinzufügen	802
27.2	Quellcode für MDI-Hauptfenster	803
27.2.1	Datei-Menü	803
27.2.2	Fenster-Menü	804
27.2.3	Hilfe-Menü	804

27.3	Quellcode für MDI-Kindfenster	805
27.3.1	Datei-Menü	806
27.3.2	Bearbeiten-Menü	807
27.3.3	Zeichen-Menü	807
27.3.4	PopUp-Menü programmieren	808
27.3.5	Programmtest	808
27.4	Dokumente drucken	809
27.4.1	Ergänzungen der Oberfläche der Kindfenster	809
27.4.2	Ergänzungen zum Quellcode der ChildForm	810
27.4.3	Test	812
27.4.4	Bemerkungen zur RichTextBox	814
28	Wissenschaftlicher Rechner	815
28.1	Basisversion des Rechners	816
28.1.1	Entwurf der Benutzerschnittstelle	816
28.1.2	Klasse Calculator	817
28.1.3	Quellcode Form1	818
28.1.4	Test	819
28.2	Die Assembly als Datei speichern	820
28.2.1	Oberfläche	820
28.2.2	Quelltext (Form1)	821
28.2.3	Test	823
28.3	Berechnungsergebnisse als Diagramm darstellen	824
28.3.1	Oberfläche	824
28.3.2	Die Klasse CCalculatorX	825
28.3.3	Quellcode Form1	826
28.3.4	Test	827
29	Datenverwaltung mittels Random Access Datei	829
29.1	Vorbereitungen	830
29.1.1	Klassenübersicht	830
29.1.2	Entwurf der Benutzerschnittstelle	830
29.2	Programmieren des Geschäftsmodells	831
29.2.1	Klasse CRandomKunde	831
29.2.2	Klasse CNewRandomFile	833
29.2.3	Klasse CRandomAccess	834

29.3	Abschließende Arbeiten	838
29.3.1	Programmieren der Bedienoberfläche	838
29.3.2	Programmtest	841
30	Verkehrsmanagement per Multithreading	843
30.1	Etwas Theorie	844
30.1.1	Kurzeinführung Threading	844
30.1.2	Wichtige Thread-Methoden	845
30.1.3	Wichtige Thread-Eigenschaften	846
30.2	Vorbereitungen	847
30.2.1	Aufgabenstellung	847
30.2.2	Klassenübersicht	847
30.2.3	Oberfläche	847
30.3	Programmierung	849
30.3.1	Klasse LKW	849
30.3.2	Klasse Schiff	851
30.3.3	Klasse Controller	854
30.3.4	Klasse Global	855
30.3.5	Klasse Form1	856
30.3.6	Abschlusstest	858
	Register	859