

# Inhalt

<b>Einleitung</b> .....	<b>1</b>
Programmiermodell im Wandel .....	2
Ereignisorientierung.....	2
Komponentenorientierung.....	2
<b>1 Hello VB – rund um das erste Programm .....</b>	<b>5</b>
1.1 Entwicklungssystem.....	5
1.1.1 Vollversionen .....	7
1.1.2 Eine IDE – was ist das überhaupt? .....	7
1.2 Hilfe .....	10
1.3 Fenster der IDE .....	11
1.4 Erstes Projekt – <i>Hello Fritz</i> .....	14
1.5 IntelliSense – Eingabe im Turboverfahren.....	18
1.5.1 QuickInfos – willkommene Kurzhilfen .....	18
1.5.2 Programmieren wie auf Schienen – das neue Eingabegefühl .....	20
1.6 Projektordner – die Dateiverwaltung der IDE .....	21
1.7 Start des Programms .....	22
1.8 Debugger – tiefere Einblicke in das Geschehen .....	23
1.8.1 Werte zur Laufzeit ändern – Fälscher am Werk .....	23
1.8.2 Von der Debug- zur Release-Version .....	25
1.9 Programmstruktur .....	26
1.9.1 Namensräume .....	28
1.9.2 <i>Main()</i> -Funktion .....	32
1.10 Schnuppertour .....	34
1.11 Aufgaben.....	35
<b>2 OOP, elementare Datentypen und Kontrollstrukturen.....</b>	<b>37</b>
2.1 Einführung in die OOP.....	37
2.1.1 Klassenbibliothek .....	38

2.2	Objektbegriff.....	40
2.2.1	Datentypinstanz als gemeinsam genutztes Datentypobjekt .....	40
2.2.2	Der elementare Datentyp <i>String</i> .....	42
2.2.3	Die anderen elementaren Datentypen.....	45
2.2.4	<i>Date</i> – Zeit- und Datumsangaben.....	48
2.3	Objektbegriff.....	48
2.3.1	Literale als Objekte .....	49
2.3.2	Typumwandlung .....	49
2.4	Kontrollstrukturen.....	51
2.4.1	Bedingte Verzweigung – <i>If</i> -Anweisung .....	51
2.4.2	Bedingte Mehrfachverzweigung – <i>Select</i> -Anweisung.....	53
2.4.3	Schleifen.....	53
2.5	Beispielprogramm – <i>BildschirmTennis</i> .....	57
2.5.1	Implementierungsdetails .....	58
2.6	Aufgaben.....	61
<b>3</b>	<b>Klassen und ihre Operationen.....</b>	<b>63</b>
3.1	Klassen und Strukturen .....	63
3.1.1	Anschauungsmaterial .....	64
3.1.2	Unterschiede zwischen <i>Structure</i> - und <i>Class</i> -Klassen .....	65
3.2	Klassendefinition aus der Nähe.....	71
3.2.1	Zugriffsmodifizierer .....	71
3.2.2	Datenfelder.....	72
3.2.3	Konstruktor .....	72
3.2.4	Methoden .....	77
3.2.5	Parameterübergabe und Wertrückgabe.....	77
3.2.6	Operatoren.....	81
3.3	Beispielprogramm – <i>BildschirmTennisOOP</i> .....	84
3.3.1	Programmdesign .....	84
3.4	Aufgaben.....	88
<b>4</b>	<b>Klassen und ihre Datenelemente .....</b>	<b>89</b>
4.1	Datenfelder .....	89
4.1.1	Zugriffsbeschränkungen .....	90
4.2	Aufzählungstypen – enum .....	92
4.2.1	Bitvektoren.....	93
4.3	Eigenschaften.....	94
4.3.1	Erweiterte Möglichkeiten von Eigenschaften.....	94
4.4	Weitere Klassenelemente .....	96
4.4.1	Verschachteltes Klassendesign.....	96
4.5	Arrays .....	97
4.5.1	Welcher Natur sind Arrays in Visual Basic? .....	97
4.5.2	Umgang mit Arrays .....	97
4.5.3	Mehrdimensionale Arrays .....	101

4.6	Codeperspektiven.....	102
4.6.1	Die vier möglichen Sichten bei der Programmierung.....	102
4.6.2	Indexer.....	104
4.7	Beispielprogramm – FallendeBälle.....	107
4.7.1	Die Klasse Bälle .....	107
4.8	Aufgaben.....	112
<b>5</b>	<b>Vererbung .....</b>	<b>113</b>
5.1	Schnittstellenvererbung.....	113
5.1.1	Leeres Erbe mit Verpflichtung .....	114
5.1.2	Die <i>IComparable</i> -Schnittstelle .....	114
5.1.3	Eine von <i>IComparable</i> abgeleitete Klasse .....	115
5.1.4	Mehrfachvererbung .....	116
5.1.5	Der tiefere Sinn von Schnittstellen .....	116
5.2	Implementierungsvererbung.....	117
5.2.1	Innere Objekte .....	118
5.2.2	Besonderheiten .....	118
5.2.3	Modifikation ererbter Elemente.....	120
5.3	Virtuelle Vererbung .....	121
5.3.1	Polymorphie .....	121
5.3.2	Anwendungsbereiche virtueller Methoden .....	124
5.3.3	Typspielereien und das <i>Type</i> -Objekt .....	124
5.4	Abstraktes Erbe .....	126
5.5	Beispielprogramm – <i>MovingObjects</i> .....	127
5.5.1	Klassendesign.....	127
5.5.2	Testcode .....	131
5.6	Aufgaben.....	134
<b>6</b>	<b>Ereignisse.....</b>	<b>135</b>
6.1	Ereignisbasiertes Programmiermodell.....	135
6.1.1	Erst einmal ins kalte Wasser mit <i>HelloWindows</i> .....	136
6.1.2	Codegerüst einer Windows-Anwendung .....	139
6.1.3	Navigation im Codegerüst.....	142
6.2	Ereignisverarbeitung .....	146
6.2.1	Nachrichtenwarteschlange.....	146
6.2.2	Nachrichtenschleife .....	148
6.2.3	Das <i>Click</i> -Ereignis im Detail.....	149
6.2.4	Andere Ereignisse.....	153
6.3	Beispielprogramm – <i>Gummiband</i> .....	156
6.3.1	Textausgabe im Formular .....	156
6.3.2	<i>Timer</i> -Steuerelement .....	157
6.3.3	Gummibandauswahl .....	157
6.4	Aufgaben.....	159

<b>7</b>	<b>Komponenten.....</b>	<b>161</b>
7.1	Mit Komponenten der Toolbox programmieren .....	162
7.1.1	Über das visuelle Design zum fertigen Code .....	162
7.2	Beispiel: Auswahl und Anzeige einer Bilddatei.....	163
7.2.1	Ansatz .....	164
7.2.2	Pflichtenheft .....	164
7.2.3	Bildanzeige.....	166
7.3	Selbst konfektionierte Komponenten .....	168
7.3.1	Test-Container.....	168
7.3.2	Dem Projekt ein Benutzersteuerelement hinzufügen.....	169
7.3.3	Innenleben eines Benutzersteuerelements .....	171
7.4	Ereignissignalisierung.....	171
7.4.1	Delegaten .....	171
7.4.2	Ereignisse .....	175
7.5	Beispielprogramm – <i>Messwertschreiber</i> .....	178
7.5.1	Zur Implementierung.....	178
7.5.2	Entwurfszeit – Laufzeit .....	181
7.5.3	Testumgebung .....	182
7.6	Aufgaben.....	185
<b>8</b>	<b>Komponenten und Exception-Handling.....</b>	<b>187</b>
8.1	Komponenten.....	188
8.1.1	Toolbox-Komponenten .....	188
8.1.2	Gummiband als Toolbox-Komponente .....	193
8.2	Codekomponenten .....	203
8.2.1	.NET-Compiler erzeugen Assemblies .....	203
8.3	Ausnahmen .....	208
8.3.1	Fehlerbehandlung – die sprachliche Seite .....	208
8.4	Aufgaben.....	211
<b>9</b>	<b>WPF-Anwendungen und XAML .....</b>	<b>213</b>
9.1	HelloWPF .....	214
9.1.1	Kompatibilität mit Windows Forms .....	215
9.1.2	Das Codegerüst einer WPF-Anwendung.....	217
9.1.3	Was steht da in der XAML-Datei? .....	220
9.2	XAML-Syntax im Überblick .....	223
9.2.1	XAML-Dokument .....	224
9.2.2	Elemente.....	224
9.2.3	Attribute .....	224
9.2.4	Angefügte Eigenschaften .....	227
9.2.5	Ereignisse .....	228
9.2.6	Markup-Erweiterungen .....	230
9.3	WPF-Techniken in der Praxis .....	233
9.3.1	Abhängigkeitseigenschaften implementieren .....	233
9.3.2	Angefügte Abhängigkeitseigenschaften implementieren .....	236
9.3.3	Konverter .....	241

9.3.4	Windows Forms-Komponenten in WPF nutzen .....	245
9.3.5	Eigene WPF-Komponenten .....	246
9.4	Aufgaben.....	251
<b>10</b>	<b>Generische Klassen.....</b>	<b>255</b>
10.1	Abstrakte Datentypen.....	256
10.2	Generische Implementierungen ohne Generics .....	256
10.2.1	Beispiel: Implementierung einer Schlange .....	257
10.2.2	Typflexion.....	259
10.2.3	<i>For Each</i> -Auflistungen.....	261
10.3	Generische Klassen mit Generics .....	263
10.3.1	Typpräzisierung.....	263
10.3.2	Implementierung von <i>Queue(Of T)</i> .....	265
10.3.3	Generische Auflistungen .....	266
10.3.4	Generische Methoden.....	267
10.3.5	Seitenbedingungen für Typparameter.....	268
10.4	Nullbare Typen .....	270
10.4.1	Notation.....	270
10.5	Aufgaben.....	271
<b>11</b>	<b>LINQ .....</b>	<b>273</b>
11.1	Einführung in LINQ .....	273
11.1.1	Die neue Implizitheit .....	274
11.1.2	Implizite Typisierung .....	274
11.1.3	Objekte mit implizitem Datentyp .....	275
11.2	LINQ – Einführung .....	277
11.2.1	Erste LINQ-Abfrage .....	279
11.2.2	Query-Operatoren .....	281
11.3	Unter der Haube von LINQ .....	285
11.3.1	Erweiterungsmethoden .....	285
11.3.2	Standard-Query-Operatoren .....	287
11.3.3	Lambda-Ausdrücke .....	291
11.4	Datenbanking .....	294
11.4.1	LINQ to XML .....	294
11.4.2	LINQ to Dataset .....	295
11.4.3	LINQ to SQL und LINQ to Entities .....	298
11.5	Aufgaben.....	303
<b>12</b>	<b>Multithreading und Finalisierung .....</b>	<b>305</b>
12.1	Multithreading .....	305
12.1.1	Der GUI-Thread .....	306
12.1.2	Thread-Pool .....	309
12.1.3	Nackte Threads .....	312
12.1.4	Gefährliche „Umstände“ .....	313
12.1.5	Asynchrone Operationen .....	317
12.1.6	Synchronisation mit GUI-Objekten .....	320

## Inhalt

12.2	Finalisierung .....	323
12.2.1	Destruktion.....	323
12.2.2	Garbage Collection.....	324
12.2.3	Vorzeitige Finalisierung.....	326
12.2.4	Rückkehr der Zombies .....	328
12.2.5	Überlebenskampf .....	329
12.3	Aufgabe.....	329
	<b>Register.....</b>	<b>331</b>