

Kapitelübersicht

	Vorwort	17
	Konzeption	19
	TEIL I: INTUITIVER EINSTIEG	21
	Das erste Makro	23
1	Einige in der Ausführung einfache, aber in der Idee allgemein gültige Beispiele zeigen, wie Sie eigene Makros erstellen und wozu Sie Makros verwenden können: um Excel nach Ihren Wünschen zu konfigurieren, eine Literaturdatenbank zu verwalten, ein »intelligentes« Formular zu erstellen, neue Tabellenfunktionen zu definieren etc.	
	Neuerungen in Excel	61
2	Das Kapitel gibt einen Überblick über die wichtigsten Neuerungen gegenüber den Vorgängerversionen. Bei Excel 2000 erleichtert die neue Datenbankbibliothek ADO den Zugriff auf externe Datenbanken. In Excel 2002 hat sich die Objektbibliothek nochmals vergrößert: die neuen Objekte ermöglichen beispielsweise den Zugriff auf Smart Tags oder die Steuerung der erweiterten Blattschutzoptionen. Excel 2003 stellt den Anwendern Funktionen zur Verarbeitung von XML-Dateien zur Verfügung.	
	TEIL II: GRUNDLAGEN	81
	Die Entwicklungsumgebung	83
3	Dieses Kapitel beschreibt detailliert die Bedienung der VBA-Entwicklungsumgebung. Die Entwicklungsumgebung ermöglicht die Eingabe von Programmcode und die Definition neuer Formulare, hilft bei der Fehlersuche, enthält eine Objektreferenz (Objektkatalog) mit Querverweisen zur Hilfe, einen Direktbereich zum Test einzelner Anweisungen etc.	

- | | | |
|----------|--|------------|
| | VBA-Konzepte | 101 |
| 4 | Wenn Ihnen Begriffe wie Variablen, Schleifen, Verzweigungen, Objekte, Methoden und Eigenschaften noch unbekannt sind, finden Sie in diesem Kapitel aussagekräftige Erklärungen. Das Kapitel umreißt die formalen Aspekte der Programmiersprache VBA und beschreibt Anwendung und Syntax aller elementaren Sprachelemente. | |
| | Programmiertechniken | 189 |
| 5 | Dieses Kapitel gibt Antworten auf alltägliche Programmierfragen: Wie erfolgt der Zugriff auf Tabellen, Bereiche oder Zellen? Wie werden Berechnungen mit Datum und Uhrzeit durchgeführt? Wie werden Zeichenketten bearbeitet? Wie werden neue Tabellenfunktionen definiert? Wie werden Dateien verwaltet? Wo werden die Konfigurationseinstellungen von Excel gespeichert? Wie kann eine Eurokonvertierung durchgeführt werden? Wie und wo finden sich weiterführende Informationen? | |
| | Fehlersuche und Fehlerabsicherung | 355 |
| 6 | Dass Programme nie fehlerfrei sind (und sein werden), hat sich inzwischen herumgesprochen. Die Softwarehersteller liefern mit unausgegorenen Versionen immer neue Belege für ihre Unvollkommenheit. Wie Sie Fehler in eigenen Anwendungen entdecken und Ihre Programme gegen Fehler absichern, ist Thema dieses Kapitels. | |
| | Dialoge (MS-Forms-Bibliothek) | 371 |
| 7 | Dialoge sind zumeist kleine Fenster zur bequemen Eingabe verschiedener Daten. Excel kennt zahllose Dialoge, etwa zur Dateiauswahl oder zur Einstellung von Optionen. Dieses Kapitel zeigt, wie Sie mit dem Dialogeditor von Excel eigene Dialoge erstellen und verwalten können. | |
| | Menüs und Symbolleisten | 433 |
| 8 | In diesem Kapitel erfahren Sie, wie Sie die Bedienungsoberfläche von Excel individuell gestalten können. Eine besonderes Augenmerk liegt dabei auf der Erstellung von eigenen Menüs und Symbolleisten ein (<i>Command-Bar</i> -Objekt), mit denen Sie die Bedienung von Excel bzw. von eigenen Excel-Anwendungen vereinfachen können. | |

	TEIL III: ANWENDUNG	467
	Mustervorlagen und »intelligente« Formulare	469
9	Eine ebenso einfache wie praktische Anwendung von Excel stellen vorgefertigte Formulare dar. Der Anwender muss nur einige Zahlenwerte einsetzen, dann werden alle weiteren Ergebnisse automatisch berechnet (daher auch die Bezeichnung »intelligent«). Anschließend kann das Formular sofort ausgedruckt und gespeichert werden.	
	Diagramme und Zeichnungsobjekte (Shapes)	501
10	Diagramme stehen im Zentrum vieler Excel-Anwendungen. Dieses Kapitel beschreibt, wie Sie Diagramme programmgesteuert erstellen und ausdrucken können. Das Kapitel geht auch auf die <i>Shape</i> -Objekte ein, mit denen gleichermaßen Diagramme und normale Tabellenblätter beschriftet bzw. verziert werden können.	
	Datenbankverwaltung in Excel	539
11	Excel ist zwar selbst kein richtiges Datenbankprogramm, es ermöglicht aber die einfache Verwaltung kleiner Datenmengen. Das Kapitel beschreibt, wie diese Funktionen in Programmen eingesetzt werden können und wo ihre Grenzen liegen.	
	Zugriff auf externe Daten	585
12	Größere Datenmengen werden besser mit einem richtigen Datenbanksystem als mit Excel verwaltet. Dieses Kapitel zeigt, welche Möglichkeiten es gibt, in Excel dennoch auf diese Daten zuzugreifen. Im Mittelpunkt steht dabei die neue ADO-Bibliothek (ActiveX Data Objects).	
	Datenanalyse in Excel	647
13	Die Stärken Excels liegen weniger bei der Verwaltung der Daten als bei deren Analyse. Dieses Kapitel behandelt in erster Linie Pivottabellen. Dabei handelt es sich um ein ungemein leistungsfähiges Werkzeug, mit dem man Tabellen nach mehreren Parametern gruppieren und gliedern kann.	

	XML- und Listenfunktionen (Excel 2003)	687
14	Die einzige wesentliche Neuerung, die Excel 2003 VBA-Programmierern bietet, sind diverse Funktionen zur Bearbeitung von Listen sowie zum Import und Export von XML-Dokumenten. Dieses Kapitel gibt einen Überblick über die neuen Objekte, Methoden und Eigenschaften und demonstriert deren Anwendung.	
	VBA-Programmierung für Profis	705
15	Ob Sie nun Add-Ins erstellen, DLL-Funktionen aufrufen, Web Services nutzen oder über ActiveX-Automation fremde Programme steuern möchten, dieses Kapitel vermittelt das erforderliche Know-how.	
	TEIL IV: REFERENZ	763
	Objektreferenz	765
16	Den größten Nachteil von VBA stellt die schier unüberblickbare Zahl von Objekten, Methoden und Eigenschaften dar. Dieses Kapitel beschreibt alle Excel- und ADO-Objekte sowie die wichtigsten Objekte einiger anderer Bibliotheken in alphabetischer Reihenfolge.	
	Anhang A: Inhalt der CD-ROM	841
	Quellenverzeichnis	843
	Stichwortverzeichnis	845

Inhalt

	Vorwort	17
	Konzeption des Buchs	19
I	Intuitiver Einstieg	21
1	DAS ERSTE MAKRO	23
1.1	Begriffsdefinition	24
1.2	Was ist Visual Basic für Applikationen?	26
1.3	Beispiel – Eine Formatvorlage mit einem Symbol verbinden	28
1.4	Beispiel – Makro zur Eingabeerleichterung	34
1.5	Beispiel – Einfache Literaturdatenbank	36
1.6	Beispiel – Formular zur Berechnung der Verzinsung von Spareinlagen	42
1.7	Beispiel – Benutzerdefinierte Funktionen	47
1.8	Beispiel – Analyse komplexer Tabellen	48
1.9	Beispiel – Vokabeltrainer	50
1.10	Weitere Beispiele zum Ausprobieren	56
2	NEUERUNGEN IN EXCEL	61
2.1	Neu in Excel 2003	62
2.2	Neu in Excel 2002	63
2.3	Neu in Excel 2000	66
2.4	Neu in Excel 97	70
2.5	Neu in Excel 7	74
2.6	Probleme und Inkompatibilitäten	74
II	Grundlagen	81
3	ENTWICKLUNGSUMGEBUNG	83
3.1	Komponenten von VBA-Programmen	84
3.2	Komponenten der Entwicklungsumgebung	85
3.3	Codeeingabe in Modulen	91
3.4	Makros ausführen	95
3.5	Makroaufzeichnung	97
3.6	Tastenkürzel	99
4	VBA-KONZEPTE	101
4.1	Variablen und Felder	102
4.1.1	Variablenverwaltung	102
4.1.2	Felder	107
4.1.3	Syntaxzusammenfassung	110

4.2	Prozedurale Programmierung	112
4.2.1	Prozeduren und Parameter	112
4.2.2	Gültigkeitsbereich von Variablen und Prozeduren	121
4.2.3	Verzweigungen (Abfragen)	126
4.2.4	Schleifen	129
4.2.5	Syntaxzusammenfassung	132
4.3	Objekte	135
4.3.1	Der Umgang mit Objekten, Methoden und Eigenschaften	135
4.3.2	Der Objektkatalog (Verweise)	141
4.3.3	Übersichtlicher Objektzugriff durch das Schlüsselwort With	145
4.3.4	Objektvariablen	146
4.3.5	Syntaxzusammenfassung	149
4.4	Ereignisse	150
4.4.1	Ereignisprozeduren	151
4.4.2	Ereignisprozeduren deaktivieren	154
4.4.3	Überblick über wichtige Excel-Ereignisse	154
4.4.4	Ereignisse beliebiger Objekte empfangen	160
4.4.5	Ereignisprozeduren per Programmcode erzeugen	161
4.4.6	Syntaxzusammenfassung	164
4.5	Programmierung eigener Klassen	166
4.5.1	Eigene Methoden, Eigenschaften und Ereignisse	168
4.5.2	Collection-Objekt	172
4.5.3	Beispiel für ein Klassenmodul	173
4.5.4	Beispiel für abgeleitete Klassen (Implements)	174
4.5.5	Syntaxzusammenfassung	180
4.6	Operatoren in VBA	181
4.7	Virenschutz	184
5	PROGRAMMIERTECHNIKEN	189
5.1	Zellen und Zellbereiche	190
5.1.1	Objekte, Methoden, Eigenschaften	190
5.1.2	Anwendungsbeispiele	204
5.1.3	Syntaxzusammenfassung	214
5.2	Arbeitsmappen, Fenster und Arbeitsblätter	216
5.2.1	Objekte, Methoden und Eigenschaften	216
5.2.2	Anwendungsbeispiele	221
5.2.3	Syntaxzusammenfassung	225
5.3	Datentransfer über die Zwischenablage	227
5.3.1	Zellbereiche kopieren, ausschneiden und einfügen	227
5.3.2	Zugriff auf die Zwischenablage mit dem DataObject	229
5.3.3	Syntaxzusammenfassung	230
5.4	Umgang mit Zahlen und Zeichenketten	231
5.4.1	Numerische Funktionen, Zufallszahlen	231
5.4.2	Zeichenketten	234
5.4.3	Umwandlungsfunktionen	239
5.4.4	Syntaxzusammenfassung	241

5.5	Rechnen mit Datum und Uhrzeit	243
5.5.1	VBA-Funktionen	247
5.5.2	Tabellenfunktionen	249
5.5.3	Anwendungs- und Programmier Techniken	250
5.5.4	Feiertage	253
5.5.5	Syntaxzusammenfassung	259
5.6	Umgang mit Dateien, Textimport/-export	261
5.6.1	File System Objects – Überblick	261
5.6.2	Laufwerke, Verzeichnisse und Dateien	263
5.6.3	Textdateien (TextStream)	269
5.6.4	Binärdateien (Open)	271
5.6.5	Excel-spezifische Methoden und Eigenschaften	276
5.6.6	Textdateien importieren und exportieren	278
5.6.7	Textexport für Mathematica-Listen	287
5.6.8	Syntaxzusammenfassung	292
5.7	Benutzerdefinierte Tabellenfunktionen	296
5.7.1	Grundlagen	296
5.7.2	Beispiele	304
5.8	Schutzmechanismen	307
5.8.1	Bewegungsradius einschränken	307
5.8.2	Zellen, Tabellenblätter und Arbeitsmappen schützen	309
5.8.3	Schutzmechanismen für den gemeinsamen Zugriff	313
5.8.4	Programmcode und Symbolleiste schützen	313
5.8.5	Syntaxzusammenfassung	315
5.9	Konfigurationsdateien, individuelle Konfiguration	315
5.9.1	Optionen	315
5.9.2	Optionseinstellungen per Programmcode	317
5.9.3	Konfigurationsdateien	319
5.10	Excel und der Euro	327
5.10.1	Die Euroconvert-Funktion	327
5.10.2	Excel-Dateien auf Euro umstellen	331
5.11	Tipps und Tricks	342
5.11.1	Geschwindigkeitsoptimierung	342
5.11.2	Zeitaufwendige Berechnungen	343
5.11.3	Effizienter Umgang mit Tabellen	347
5.11.4	Zusammenspiel mit Excel-4-Makros	350
5.11.5	Excel-Version feststellen	351
5.11.6	Hilfe zur Selbsthilfe	352
5.11.7	Syntaxzusammenfassung	353
6	FEHLERSUCHE UND FEHLERABSICHERUNG	355
6.1	Hilfsmittel zur Fehlersuche (Debugging)	356
6.1.1	Syntaxkontrolle	356
6.1.2	Reaktion auf Fehler	357
6.1.3	Kontrollierte Programmausführung	359
6.2	Fehlertolerantes Verhalten von Programmen	362
6.3	Reaktion auf Programmunterbrechungen	367
6.4	Syntaxzusammenfassung	368

7	DIALOGE (MS-FORMS-BIBLIOTHEK)	371
7.1	Vordefinierte Dialoge	372
7.1.1	Excel-Standarddialoge	372
7.1.2	Die Funktionen MsgBox und InputBox	375
7.1.3	Die Methode Application.InputBox	376
7.2	Selbst definierte Dialoge	378
7.2.1	Veränderungen gegenüber Excel 5/7	378
7.2.2	Einführungsbeispiel	380
7.3	Der Dialogeditor	384
7.4	Die MS-Forms-Steuer Elemente	388
7.4.1	Bezeichnungsfeld (Label)	389
7.4.2	Textfeld (TextBox)	390
7.4.3	Listenfeld (ListBox) und Kombinationslistenfeld (ComboBox)	393
7.4.4	Kontrollkästchen (CheckBox) und Optionsfelder (OptionButton)	399
7.4.5	Buttons (CommandButton) und Umschaltbuttons (ToggleButton)	401
7.4.6	Rahmenfeld (Frame)	402
7.4.7	Multiseiten (MultiPage), Register (TabStrip)	403
7.4.8	Bildlaufleiste (ScrollBar) und Drehfeld (SpinButton)	408
7.4.9	Anzeige (Image)	409
7.4.10	Formelfeld (RefEdit)	410
7.4.11	Das UserForm-Objekt	412
7.5	Steuerelemente direkt in Tabellen verwenden	415
7.6	Programmiertechniken	423
7.6.1	Zahleneingabe	423
7.6.2	Dialoge gegenseitig aufrufen	424
7.6.3	Dialoge dynamisch verändern	427
7.6.4	Umgang mit Drehfeldern	428
8	MENÜS UND SYMBOLLEISTEN	433
8.1	Menüs und Symbolleisten	434
8.1.1	Manuelle Veränderung von Menüs und Symbolleisten	434
8.1.2	Veränderungen speichern	439
8.1.3	Objekthierarchie	441
8.1.4	Programmiertechniken	446
8.1.5	Blattwechsel über die Symbolleiste	453
8.1.6	Unterschiede gegenüber Excel 5/7	456
8.1.7	Syntaxzusammenfassung	457
8.2	Oberflächengestaltung für eigenständige Excel-Anwendungen	459
8.2.1	Erweiterung des Standardmenüs	460
8.2.2	Eigene Symbolleisten ein- und ausblenden	462
8.2.3	Eigenes Standardmenü verwenden	463

III	Anwendung	467
9	MUSTERVORLAGEN UND »INTELLIGENTE« FORMULARE	469
9.1	Grundlagen	470
9.1.1	Gestaltungselemente für »intelligente« Formulare	471
9.1.2	Mustervorlagen mit Datenbankanbindung	478
9.2	Beispiel – Das »Speedy«-Rechnungsformular	482
9.3	Beispiel – Abrechnungsformular für einen Car-Sharing-Verein	491
9.4	Grenzen »intelligenter« Formulare	497
10	DIAGRAMME UND ZEICHNUNGSOBJEKTE (SHAPES)	501
10.1	Umgang mit Diagrammen	502
10.1.1	Grundlagen	502
10.1.2	Diagrammtypen	503
10.1.3	Diagrammelemente (Diagrammobjekte) und Formatierungsmöglichkeiten	504
10.1.4	Ausdruck	508
10.2	Programmierung von Diagrammen	509
10.2.1	Objekthierarchie	510
10.2.2	Programmiertechniken	513
10.3	Beispiel – Automatische Datenprotokollierung	517
10.3.1	Die Bedienung des Beispielprogramms	517
10.3.2	Programmcode	519
10.4	Syntaxzusammenfassung Diagramme	531
10.5	Zeichnungsobjekte (Shapes)	532
10.6	Organigramme und andere Diagramme	536
11	DATENVERWALTUNG IN EXCEL	539
11.1	Grundlagen	540
11.1.1	Einleitung	540
11.1.2	Kleines Datenbankglossar	541
11.1.3	Excel versus Datenbanksysteme	542
11.2	Datenverwaltung innerhalb von Excel	545
11.2.1	Eine Datenbank in Excel erstellen	545
11.2.2	Daten über die Datenbankmaske eingeben, ändern und löschen	548
11.2.3	Daten sortieren, suchen, filtern	550
11.3	Datenverwaltung per VBA-Code	557
11.3.1	Programmiertechniken	557
11.3.2	Beispiel – Word-Serienbrief	560
11.3.3	Syntaxzusammenfassung	563
11.4	Datenbank-Tabellenfunktionen	563
11.5	Tabellen konsolidieren	567
11.5.1	Grundlagen	567
11.5.2	Konsolidieren per VBA-Code	569

11.6	Beispiel – Abrechnung eines Car-Sharing-Vereins	570
11.6.1	Bedienung	570
11.6.2	Überblick über die Komponenten der Anwendung	574
11.6.3	Programmcode	575
12	ZUGRIFF AUF EXTERNE DATEN	585
12.1	Grundkonzepte relationaler Datenbanken	586
12.2	Import externer Daten	592
12.2.1	Daten aus Datenbanken importieren (MS Query)	592
12.2.2	Das QueryTable-Objekt	604
12.2.3	Excel-Daten exportieren	608
12.3	Datenbankzugriff mit der ADO-Bibliothek	609
12.3.1	Einführung	609
12.3.2	Verbindungsaufbau (Connection)	614
12.3.3	Datensatzlisten (Recordset)	617
12.3.4	SQL-Kommandos (Command)	624
12.3.5	SQL-Grundlagen	625
12.3.6	Syntaxzusammenfassung	629
12.4	Beispiel – Fragebogenauswertung	630
12.4.1	Überblick	630
12.4.2	Aufbau des Fragebogens	634
12.4.3	Aufbau der Datenbank	635
12.4.4	Programmcode	637
13	DATENANALYSE IN EXCEL	647
13.1	Daten gruppieren (Teilergebnisse)	648
13.1.1	Einführung	648
13.1.2	Programmierung	649
13.2	Pivottabellen (Kreuztabellen)	651
13.2.1	Einführung	651
13.2.2	Gestaltungsmöglichkeiten	656
13.2.3	Pivottabellen für externe Daten	662
13.2.4	Pivottabellenoptionen	666
13.2.5	Pivotdiagramme	667
13.3	Programmiertechniken	668
13.3.1	Pivottabellen erzeugen und löschen	668
13.3.2	Aufbau und Bearbeitung vorhandener Pivottabellen	673
13.3.3	Interne Verwaltung (PivotCache)	678
13.3.4	Syntaxzusammenfassung	685
14	XML- UND LISTEN-FUNKTIONEN (EXCEL 2003)	687
14.1	Bearbeitung von Listen	688
14.2	XML-Grundlagen	690
14.3	XML-Funktionen interaktiv nutzen	693
14.4	XML-Programmierung	696

15	VBA-PROGRAMMIERUNG FÜR PROFIS	705
15.1	Add-Ins	706
15.2	Excel und das Internet	710
15.2.1	Excel-Dateien als E-Mail versenden	710
15.2.2	HTML-Import	712
15.2.3	HTML-Export, Webkomponenten	714
15.3	Smart Tags	717
15.4	Web Services nutzen	720
15.5	Dynamic Link Libraries (DLLs) verwenden	727
15.6	ActiveX-Automation	732
15.6.1	Excel als Client (Steuerung fremder Programme)	734
15.6.2	Excel als Server (Steuerung durch fremde Programme)	739
15.6.3	Neue Objekte für Excel (Clipboard-Beispiel)	744
15.6.4	Object Linking and Embedding (OLE)	747
15.6.5	Automation und Visual Basic .NET	751
15.6.6	Programme ohne ActiveX starten und steuern	760
15.6.7	Syntaxzusammenfassung	762
IV	Referenz	763
16	OBJEKTFERENZ	765
16.1	Objekthierarchie	766
16.2	Alphabetische Referenz	773
	ANHANG	841
A	Die beiliegende CD	841
	Quellenverzeichnis	843
	Stichwortverzeichnis	845

Vorwort

Excel bietet von Haus aus ein riesiges Spektrum von Funktionen. Wozu also brauchen Sie dann noch Makros? Mit VBA können Sie ...

- eigene Tabellenfunktionen programmieren, die einfacher anzuwenden sind als komplizierte Formeln.
- Excel nach Ihren Vorstellungen konfigurieren und auf diese Weise eine einfachere und effizientere Bedienung erreichen.
- komplexe Arbeitsschritte wie etwa das Ausfüllen von Formularen durch »intelligente« Formulare (Mustervorlagen) strukturieren und erleichtern.
- immer wieder auftretende Arbeitsvorgänge automatisieren. Das empfiehlt sich besonders dann, wenn regelmäßig große Datenmengen anfallen, die verarbeitet, analysiert und grafisch aufbereitet werden sollen.
- eigenständige Excel-Programme erstellen, die sich durch eigene (kurze) Menüs, eigene Dialoge etc. auszeichnen. Damit lassen sich Excel-Anwendungen in ihrer Bedienung so weit vereinfachen, dass sie von anderen Personen (auch von Excel-Laien) ohne lange Einweisung verwendet werden können.

Was ist neu in Excel 2000, Excel 2002 und Excel 2003?

Mit Excel 97 haben sich sowohl das VBA-Programmiermodell als auch die wichtigsten Objektbibliotheken weitgehend stabilisiert. Zwar wurden auch in Excel 2000, 2002 und 2003 diverse neue Objekte eingeführt, davon hat sich aber nur ein Teil in der Praxis durchgesetzt. Die größte Bedeutung haben sicherlich die *Scripting*- und die ADO-Bibliothek, die beide seit Excel 2000 zur Verfügung stehen und den Zugang auf Dateien, Verzeichnisse und externe Datenbanken erleichtern. In ganz neue Anwendungswelten stößt Excel 2003 mit seinen XML-Funktionen vor (siehe Kapitel 14).

Excel und .NET

Im Zusammenhang mit der .NET-Offensive von Microsoft sollte erwähnt werden, dass Office 2000 bis 2003 keine .NET-Programme sind. Die Makroprogrammiersprache VBA basiert auf Visual Basic 6, nicht auf Visual Basic .NET. Eine wesentliche Funktion von .NET, nämlich so genannte Web Services, können Sie freilich schon jetzt nutzen: Das Web Services Toolkit, eine einfache und kostenlose Erweiterung für die VBA-Entwicklungsumgebung, macht es möglich. (Details zur Anwendung dieses Toolkits finden Sie in Kapitel 15.)

Warum dieses Buch?

Das eigentliche Problem bei der VBA-Programmierung ist nicht VBA an sich: Die Formulierung einer Schleife oder Abfrage, der Umgang mit Methoden und Eigenschaften etc. ist wie in jeder anderen Programmiersprache leicht zu verstehen und wird Ihnen nach zwei oder drei Tagen keine Schwierigkeiten mehr machen. Das Problem besteht darin, die komplexen Objektbibliotheken von Excel mit ihren Tausenden von Schlüsselwörtern zu überblicken und effizient zu nutzen. Der Schwerpunkt dieses Buchs und der vorgestellten Beispiele liegt daher bei der Erläuterung eben jener Objektbibliotheken.

Anders als bei der Excel-Hilfe besteht das Ziel dieses Buchs nicht darin, eine – oft nichts sagende – Referenz von Schlüsselwörtern zu bieten. Vielmehr geht es darum, die Zusammenhänge zwischen den zahlreichen Objekten zu verdeutlichen und konkrete Anwendungsmöglichkeiten aufzuzeigen, wie sie in der Praxis tatsächlich vorkommen. Gleichzeitig werden aber auch die Probleme Excels beim Namen genannt, um Ihnen die langwierige Suche nach Fehlern zu ersparen, die Sie gar nicht selbst verursacht haben.

Viel Erfolg!

Die Beispiele dieses Buchs zeigen, wie weit Excel-Programmierung gehen kann. Die Möglichkeiten sind wirklich beinahe unbegrenzt! Dieses Buch soll Ihnen helfen, sich so rasch wie möglich in das Objektmodell von Excel einzuarbeiten. Wenn diese Hürde einmal überwunden ist (rechnen Sie mit mindestens ein, zwei Wochen!), beginnt die VBA-Programmieren richtig Spaß zu machen. In diesem Sinne wünsche ich Ihnen viel Erfolg und vielleicht sogar Spaß und Freude bei der Excel-Programmierung!

Michael Kofler, August 2004

<http://www.kofler.cc>

Konzeption des Buchs

Visual Basic für Applikationen ist zwar eine moderne objektorientierte Programmiersprache, die schier unglaubliche Zahl von Schlüsselwörtern bringt aber auch viele Probleme mit sich. Während des Einstiegs ist es so gut wie unmöglich, auch nur einen Überblick über VBA zu gewinnen. Und selbst nach monatelanger Programmierung mit VBA wird die Hilfe der wichtigste Ratgeber zu den Details eines bestimmten Schlüsselwortes bleiben. Dieses Buch versucht deswegen ganz bewusst das zu bieten, was in der Originaldokumentation bzw. in der Hilfe zu kurz kommt:

- »echte« Anwendungen in Form von konkreten, realitätsbezogenen Beispielen
- themenorientierte Syntaxzusammenfassungen (z. B. alle Eigenschaften und Methoden zur Bearbeitung von Zellbereichen)
- detaillierte Informationen zur Anwendung von DLL-Funktionen, zu ActiveX-Automation, zur Programmierung eigener Add-Ins, zur Verwendung von Web Services etc.
- aussagekräftige Beschreibungen aller Objekte von VBA und ihre Einordnung in die Objekthierarchie
- Informationen über Ungereimtheiten und Fehler von VBA

Einem Anspruch wird das Buch aber ganz bewusst nicht gerecht: dem der Vollständigkeit. Es erscheint mir sinnlos, Hunderte von Seiten mit einer Referenz aller Schlüsselwörter zu füllen, wenn Sie beinahe dieselben Informationen auch in der Hilfe finden können. Anstatt auch nur den Anschein der Vollständigkeit zu vermitteln zu versuchen, habe ich mich bemüht, wichtigeren Themen den Vorrang zu geben und diese ausführlich und fundiert zu behandeln.

Formalitäten

Tastaturabkürzungen werden in der Form Strg+F2 angegeben, Dateinamen als c:\readme.txt. Menükommandos, Symbole (also deren gelber Info-Text) und Buttontexte werden in Kapitälchen dargestellt: DATEI|ÖFFNEN, ABBRUCH oder OK. Die Anweisung BEARBEITEN|INHALTE EINFÜGEN|VERKNÜPFUNG EINFÜGEN meint, dass Sie zuerst das Menükommando BEARBEITEN|INHALTE EINFÜGEN ausführen und im daraufhin erscheinenden Dialog den Button VERKNÜPFUNG EINFÜGEN auswählen sollen.

VBA-Schlüsselwörter sowie eigene Variablen und Prozeduren werden *kursiv* angegeben, etwa *Application*-Objekt oder *Visible*-Eigenschaft. Tabellenfunktionen sind in der gleichen Schrift, aber in Großbuchstaben angegeben, etwa WENN. (Tabellefunktionen sind auch anhand der Sprache von VBA-Schlüsselwörtern zu unterscheiden:

VBA-Schlüsselwörter sind englisch, Tabellenfunktionen deutsch.) Schlüsselwörter, die zum ersten Mal im Text auftreten und näher erklärt werden oder die gerade besonders wichtig sind, werden durch *diese Schrift* hervorgehoben.

Beispielcode, Beispieldateien

Aus Platzgründen sind in diesem Buch immer nur die wichtigsten Codepassagen der Beispielprogramme abgedruckt. Den vollständigen Code finden Sie in den Beispielprogrammen auf der beiliegenden CD. Die Beispieldateien sind kapitelweise organisiert. Code in diesem Buch beginnt immer mit einem Kommentar, der auf die entsprechende Beispieldatei verweist.

```
' Beispiel 01\format.xls
Sub FormatAsResult()
    Selection.Style = "result"
End Sub
```

Excel XP oder Excel 2002? Office XP oder Office 2002?

Microsoft bezeichnet die 2002er-Generation des Office-Pakets als Office XP, wohl in Anlehnung an Windows XP. Die einzelnen Komponenten von Office XP heißen aber Word 2002, Excel 2002 etc. Mit anderen Worten: Excel XP = Excel 2002. Dieses Buch verwendet generell nur die Bezeichnung Excel 2002 bzw. Office 2002.

Bei Office 2003 hat der Benennungsunfug zum Glück wieder aufgehört, sowohl das Gesamtpaket als auch die Einzelkomponenten werden gleich bezeichnet.

Und eine Entschuldigung

Ich bin mir bewusst, dass unter den Lesern dieses Buchs auch zahlreiche Frauen sind. Dennoch ist in diesem Buch immer wieder von *dem Anwender* die Rede, wenn ich keine geschlechtsneutrale Formulierung gefunden habe. Ich bitte dafür alle Leserinnen ausdrücklich um Entschuldigung. Ich bin mir des Problems bewusst, empfinde Doppelgleisigkeiten der Form *der/die Anwender/in* oder kurz *AnwenderIn* aber sprachlich nicht schön – und zwar sowohl beim Schreiben als auch beim Lesen.