

Inhaltsverzeichnis

Über die Autoren	7
Einleitung	27
Themen dieses Buches	27
Was Sie wissen sollten	28
Was Sie brauchen	28
Konventionen in diesem Buch	28
Excel-Befehle	29
Visual-Basic-Editor-Befehle	29
Tastatureingaben	29
Was die Symbole bedeuten	30
Wie dieses Buch aufgebaut ist	31
Teil I: Einführung in Excel-VBA	31
Teil II: Fortgeschrittene VBA-Techniken	31
Teil III: Mit UserForms arbeiten	31
Teil IV: Excel-Anwendungen entwickeln	32
Teil V: Anhang	32
Über die Website zum Buch	32
Teil I	
Einführung in Excel-VBA	33
Kapitel 1	
Grundlagen der Entwicklung von Arbeitsblattanwendungen	35
Was ist eine Arbeitsblattanwendung	35
Schritte bei der Entwicklung der Anwendung	36
Die Anforderungen der Anwender ermitteln	37
Eine Anwendung planen, die die Anforderungen der Anwender erfüllt	38
Die am besten geeignete Benutzeroberfläche festlegen	40
Anpassen des Menübands	41
Anpassen der Kontextmenüs	41
Tastenkombinationen definieren	41
Benutzerdefinierte Dialogfelder erstellen	42
ActiveX-Steuerelemente auf einem Arbeitsblatt verwenden	43
Die Anwendung entwickeln	45

Denken Sie immer an den Anwender	45
Die Anwendung testen	45
Die Anwendung narrensicher machen	46
Erstellen Sie eine ästhetisch ansprechende und intuitive Anwendung	48
Ein Hilfesystem erstellen	49
Dokumentieren Sie Ihre Entwicklungsarbeit	50
Die Anwendung an die Benutzer verteilen	50
Die Anwendung updaten, falls nötig	51
Weitere Punkte, die für die Entwicklung wichtig sind	51
Die beim Anwender installierte Excel-Version	51
Probleme aufgrund der Sprache	51
Systemgeschwindigkeit	52
Videomodi	52

Kapitel 2

Einführung in Visual Basic for Applications

55

Sich mit dem Makrorekorder einen Vorsprung verschaffen	55
Ihr erstes Makro erstellen	56
Das Makro untersuchen	57
Ihr Makro testen	58
Ihr Makro bearbeiten	59
Absolute und relative Makroaufzeichnung im Vergleich	59
Makros mit absoluten Verweisen aufzeichnen	59
Makros mit relativen Verweisen aufzeichnen	62
Weitere Konzepte für die Makroaufzeichnung	64
Dateierweiterungen für Arbeitsmappen mit Makros	65
Makrosicherheit in Excel	65
Vertrauenswürdige Speicherorte	65
Makros in Ihrer persönlichen Makroarbeitsmappe abspeichern	66
Ein Makro einer Schaltfläche oder einem anderen Formularsteuerelement zuweisen	67
Ein Makro in die Symbolleiste für den Schnellzugriff einfügen	69
In Visual Basic-Editor arbeiten	70
Die VBE-Komponenten verstehen	70
Mit dem Projekt-Explorer arbeiten	72
Ein neues VBA-Modul einfügen	73
Ein VBA-Modul entfernen	74
Im Codefenster arbeiten	75
Fenster minimieren und maximieren	75
VBA-Code in ein Modul einfügen	75
Die VBA-Entwicklungsumgebung anpassen	78
Die Registerkarte »Editor«	78
Die Registerkarte »Editorformat«	80
Die Registerkarte »Allgemein«	81
Die Registerkarte »Verankern«	82

VBA-Grundlagen	82
Objekte verstehen	83
Sammlungen verstehen	84
Eigenschaften verstehen	84
Methoden verstehen	87
Tiefer einsteigen: Range-Objekte einsetzen	89
Die Eigenschaften des Range-Objekts finden	89
Die Eigenschaft Range	89
Die Eigenschaft Cells	91
Die Eigenschaft Offset	93
Wichtige Konzepte, die Sie sich merken sollten	95
Keine Panik – Sie sind nicht allein	97
Lesen Sie den Rest dieses Buches	97
Lassen Sie sich von Excel beim Schreiben Ihres Makros helfen	97
Verwenden Sie die VBA-Hilfdateien	97
Verwenden Sie den Objektkatalog	98
Stibitzen Sie Code im Internet	100
Userforen optimal nutzen	100
Besuchen Sie Experten-Blogs	101
Suchen Sie auf YouTube nach Schulungsvideos	102
Vom Office Dev Center lernen	102
Analysieren Sie andere Excel-Dateien in Ihrem Unternehmen	102
Fragen Sie die Excel-Experten vor Ort	102

Kapitel 3

VBA-Programmiergrundlagen **103**

VBA-Sprachelemente – Ein Überblick	103
Kommentare	105
Variablen, Datentypen und Konstanten	107
Datentypen definieren	108
Variablen deklarieren	110
Den Datentyp ermitteln	111
Zwingen Sie sich, alle Variablen zu deklarieren	111
Gültigkeitsbereich von Variablen	112
Öffentliche Variablen	115
Konstanten verwenden	115
Mit Zeichenfolgen arbeiten	117
Mit Datumswerten arbeiten	117
Zuweisungsanweisungen	119
Arrays	121
Arrays deklarieren	121
Mehrdimensionale Arrays deklarieren	122
Dynamische Arrays deklarieren	123
Objektvariablen	123
Benutzerdefinierte Datentypen	125

Eingebaute Funktionen	126
Mit Objekten und Sammlungen arbeiten	129
Das With-End With-Konstrukt	129
Die For Each-Next-Schleife	130
Die Codeausführung steuern	132
GoTo-Anweisungen	133
If-Then-Verzweigung	133
Select Case-Verzweigung	138
Anweisungsblöcke in einer Schleife ausführen	141
Do Until-Schleifen	148

Kapitel 4

***VBA-Sub-Prozeduren verwenden* 151**

Über Prozeduren	151
Eine Sub-Prozedur deklarieren	152
Gültigkeitsbereich und Sichtbarkeit einer Prozedur festlegen	153
Sub-Prozeduren ausführen	154
Eine Prozedur mit dem Befehl »Sub/UserForm ausführen« starten	155
Eine Prozedur vom Dialogfeld »Makro« aus ausführen	155
Eine Prozedur durch Drücken einer Tastenkombination ausführen	156
Eine Prozedur über das Menüband ausführen	158
Eine Prozedur über ein benutzerdefiniertes Kontextmenü ausführen	158
Eine Prozedur von einer anderen Prozedur aus ausführen	158
Eine Prozedur in einem anderen Modul ausführen	160
Eine Prozedur in einer anderen Arbeitsmappe aufrufen	160
Eine Prozedur durch Anklicken eines Objekts ausführen	163
Eine Prozedur beim Eintritt eines Ereignisses ausführen	165
Eine Prozedur vom Direktfenster aus ausführen	165
Argumente an Prozeduren übergeben	166
Techniken zur Behandlung von Fehlern	170
Fehler abfangen	170
Beispiele für die Fehlerbehandlung	172
Ein realistisches Beispiel, das Sub-Prozeduren verwendet	175
Das Ziel	176
Anforderungen an das Projekt	176
Was Sie wissen	176
Der Ansatz	177
Ein paar vorbereitende Aufzeichnungen	178
Das anfängliche Setup	179
Den Code schreiben	180
Die Prozedur für die Sortierung schreiben	181
Mehr Tests	186
Die Probleme beheben	187
Verfügbarkeit des Tools	190
Das Projekt evaluieren	191

Kapitel 5
Funktionsprozeduren erstellen
193

Sub-Prozeduren und Funktionsprozeduren im Vergleich	193
Warum benutzerdefinierte Funktionen erstellen	194
Ein einführendes Beispiel für eine Funktion	194
Die Funktion in einem Tabellenblatt verwenden	195
Die Funktion in einer VBA-Prozedur verwenden	196
Die benutzerdefinierte Funktion analysieren	197
Funktionsprozeduren	199
Gültigkeitsbereich von Funktionsprozeduren	200
Funktionsprozeduren ausführen	201
Funktionsargumente	205
Beispiele für Funktionsprozeduren	205
Funktionsprozeduren ohne Argumente	205
Eine Funktionsprozedur mit einem Argument	208
Eine Funktionsprozedur mit zwei Argumenten	211
Eine Funktionsprozedur mit einem Array-Argument	212
Eine Funktionsprozedur mit optionalen Argumenten	213
Eine Funktionsprozedur, die ein VBA-Array zurückgibt	215
Eine Funktionsprozedur, die einen Fehlerwert zurückgibt	218
Eine Funktionsprozedur mit einer beliebigen Anzahl Argumenten	219
Die Excel-Funktion SUMME emulieren	221
Erweiterte Datumsfunktionen	224
Funktionen debuggen	226
Das Dialogfeld »Funktion einfügen« berücksichtigen	228
Die Methode »MacroOptions« verwenden	228
Eine Funktionskategorie angeben	230
Eine Funktionsbeschreibung von Hand hinzufügen	232
Benutzerdefinierte Funktionen in Add-Ins speichern	232
Die Windows-API verwenden	233
Windows-API-Beispiele	234
Das Windows-Verzeichnis ermitteln	234
Den Status der Umschalttasten ermitteln	235
Mehr über API-Funktionen lernen	236

Kapitel 6
Die Excel-Ereignisse verstehen
237

Was Sie über Ereignisse wissen sollten	237
Ereignissequenzen verstehen	238
Wo kommen die Prozeduren für die Ereignisverarbeitung hin	238
Ereignisse deaktivieren	240
Den Code für eine Ereignisprozedur eingeben	242
Prozeduren zur Ereignisverarbeitung, die Argumente verwenden	243

Sich mit den Ereignissen auf der Arbeitsmappenebene vertraut machen	245
Das Ereignis Open	246
Das Ereignis Activate	247
Das Ereignis SheetActivate	247
Das Ereignis NewSheet	247
Das Ereignis BeforeSave	248
Das Ereignis Deactivate	248
Das Ereignis BeforePrint	249
Das Ereignis BeforeClose	250
Die Arbeitsblattereignisse untersuchen	253
Das Ereignis Change	253
Einen bestimmten Bereich auf Änderungen überwachen	254
Das Ereignis SelectionChange	260
Das Ereignis BeforeDoubleClick	261
Das Ereignis BeforeRightClick	262
Ereignisse auf Anwendungsebene überwachen	263
Ereignisse auf Anwendungsebene aktivieren	264
Feststellen, wann eine Arbeitsmappe geöffnet wird	265
Ereignisse auf Anwendungsebene überwachen	266
Ereignisse verwenden, die nicht mit einem Objekt verknüpft sind	267
Das Ereignis OnTime	267
Das Ereignis OnKey	269

Kapitel 7

VBA-Programmierung: Beispiele und Techniken

275

Am Beispiel lernen	275
Mit Bereichen arbeiten	276
Einen Bereich kopieren	276
Einen Bereich verschieben	277
Einen Bereich variabler Größe kopieren	277
Unterschiedliche Arten von Bereichen auswählen und erkennen	279
Die Größe eines Bereichs ändern	282
Einen Zellwert anfordern	282
Einen Wert in die nächste leere Zelle eingeben	284
Ein Makro anhalten, damit der Anwender einen Bereich auswählen kann	285
Ausgewählte Zellen zählen	287
Den Typ des ausgewählten Bereichs bestimmen	288
Einen Bereich effizient in einer Schleife durchlaufen	290
Alle leeren Zeilen löschen	294
Zeilen beliebig oft duplizieren	295
Feststellen, ob ein Bereich in einem anderen enthalten ist	296
Den Datentyp einer Zelle bestimmen	297
Bereiche lesen und schreiben	298
Ein besserer Ansatz, um einen Bereich zu schreiben	300

Eindimensionale Arrays übertragen	302
Einen Bereich in ein Variant-Array übertragen	302
Zellen anhand ihres Wertes auswählen	303
Einen nicht zusammenhängenden Bereich kopieren	305
Mit Arbeitsmappen und Blättern arbeiten	307
Alle Arbeitsmappen speichern	307
Alle Arbeitsmappen speichern und schließen	308
Alles, bis auf die Auswahl ausblenden	308
Ein Inhaltsverzeichnis mit Hyperlinks erstellen	310
Tabellenblätter synchronisieren	311
VBA-Techniken	312
Eine boolesche Eigenschaft umschalten	312
Das Datum und die Zeit anzeigen	313
Eine freundliche Zeit anzeigen	315
Liste der installierten Schriftarten erstellen	317
Ein Array sortieren	318
Eine Reihe von Dateien bearbeiten	320
Einige nützliche Funktionen, die Sie in Ihrem Code verwenden können	322
Die Funktion FileExists	322
Die Funktion FileNameOnly	322
Die Funktion PathExists	323
Die Funktion RangeNameExists	323
Die Funktion SheetExists	324
Die Funktion WorkbookIsOpen	324
Einen Wert aus einer geschlossenen Arbeitsmappe abrufen	325
Einige nützliche Arbeitsblatffunktionen	327
Informationen über die Zellformatierung zurückgeben	328
Ein sprechendes Tabellenblatt	329
Datum anzeigen, an dem eine Datei gespeichert oder gedruckt wurde	330
Übergeordnete Objekte verstehen	331
Zellen zählen, deren Wert zwischen zwei Werten liegen	332
Die letzte, nicht leere Zelle einer Spalte oder Zeile ermitteln	333
Entspricht eine Zeichenfolge einem Muster?	334
Das n-te Elemente aus einer Zeichenfolge extrahieren	336
Eine Zahl in Worten	336
Eine Funktion mit mehreren Funktionen	337
Die Funktion SHEETOFFSET	338
Den größten Wert einer Zelle in allen Arbeitsblättern	339
Ein Array mit zufälligen Ganzzahlen ohne Duplikate zurückgeben	341
Einen Bereich zufällig anordnen	342
Einen Bereich sortieren	344
Windows-API-Aufrufe	345
API-Deklarationen verstehen	345
32-Bit- und 64-Bit-Deklarationen	346
Dateiverknüpfung ermitteln	347
Informationen zum Standarddrucker abrufen	348

Informationen zu den Bildschirmen abrufen	350
Lesen aus und schreiben in die Registrierungsdatenbank	351
Aus der Registrierungsdatenbank lesen	352
In die Registrierungsdatenbank schreiben	352

Teil II

Fortgeschrittene VBA-Techniken 355

Kapitel 8

Mit PivotTables arbeiten 357

Ein einführendes Beispiel in PivotTables	357
Eine PivotTable erstellen	358
Den aufgezeichneten Code für die PivotTable untersuchen	360
Den aufgezeichneten PivotTable-Code aufräumen	361
Eine komplexere PivotTable erstellen	363
Der Code, der die PivotTable erstellt	365
So funktioniert der Code für die komplexe PivotTable	366
Mehrere PivotTables erstellen	368
Eine PivotTable in eine normale Tabelle konvertieren	372

Kapitel 9

Mit Diagrammen arbeiten 375

Das Innenleben von Diagrammen	375
Speicherorte von Diagrammen	375
Der Makrorekorder und Diagramme	376
Das Chart-Objektmodell	377
Ein eingebettetes Diagramm erstellen	378
Ein Diagramm auf einem Diagrammblatt erstellen	380
Diagramme bearbeiten	380
Mit VBA ein Diagramm aktivieren	381
Ein Diagramm verschieben	382
Mit VBA ein Diagramm deaktivieren	384
Prüfen, ob ein Diagramm aktiviert ist	384
Diagramme aus den Sammlungen ChartObjects und Charts entfernen	385
Alle Diagramme in einer Schleife durchlaufen	386
Größe und Ausrichtung eines ChartObject ändern	389
Ganz viele Diagramme erstellen	390
Ein Diagramm exportieren	393
Alle Grafiken exportieren	393
Die im Diagramm verwendeten Daten ändern	395
Diagrammdaten in Abhängigkeit von der aktiven Zelle ändern	396
Mit VBA die in einem Diagramm dargestellten Datenbereiche ermitteln	398

Mit VBA beliebige Datenbeschriftungen in einem Diagramm anzeigen	401
Ein Diagramm in einem UserForm anzeigen	405
Diagrammereignisse verstehen	408
Ein Beispiel für die Verwendung von Diagrammereignissen	409
Ereignisse für ein eingebettetes Diagramm aktivieren	411
Beispiel: Diagrammereignisse für ein eingebettetes Diagramm verwenden	413
VBA-Tricks für Diagramme	415
Eingebettete Diagramme auf einer eigenen Seite ausdrucken	415
Verknüpfung zwischen Diagramm und Datenquelle aufheben	416
Mit dem Ereignis MouseOver Text anzeigen	419
Ein Diagramm scrollen	421
Mit Sparklinediagrammen arbeiten	423

Kapitel 10

Zusammenarbeit mit anderen Anwendungen

427

Microsoft-Office-Automatisierung verstehen	427
Das Konzept der Bindung verstehen	427
Frühe Bindung	428
Späte Bindung	429
Ein einfaches Automatisierungsbeispiel	430
Access von Excel aus automatisieren	431
Eine Access-Abfrage von Excel aus ausführen	431
Ein Access-Makro von Excel aus starten	432
Word von Excel aus automatisieren	433
Excel-Daten an ein Word-Dokument senden	434
Seriendruck mit einem Word-Dokument simulieren	435
PowerPoint von Excel aus automatisieren	438
Excel-Daten an eine PowerPoint-Präsentation senden	438
Alle Excel-Diagramme an eine PowerPoint-Präsentation senden	439
Eine Arbeitsmappe in eine PowerPoint-Präsentation konvertieren	441
Outlook von Excel aus automatisieren	443
Die aktive Arbeitsmappe als Anhang mailen	443
Einen bestimmten Zellbereich als Anlage mailen	444
Ein einzelnes Arbeitsblatt als Anhang mailen	446
E-Mail an alle Adressen in Ihrer Kontaktliste senden	447
Andere Anwendungen von Excel aus starten	448
Die VBA-Funktion Shell verwenden	449
Die Windows-API-Funktion ShellExecute verwenden	451
AppActivate verwenden	453
Dialogfelder der Systemsteuerung starten	454

Kapitel 11

Externe Daten und Dateien verwenden 455

Externe Datenverbindungen verwenden	455
Manuell eine Datenverbindung einrichten	455
Datenverbindungen manuell bearbeiten	460
Mit VBA dynamische Datenverbindungen erstellen	461
Alle Verbindungen in einer Arbeitsmappe durchlaufen	464
Externe Daten mit ADO und VBA abrufen	465
Die Verbindungszeichenfolge	465
Ein Recordset deklarieren	467
Auf die ADO-Objektbibliothek verweisen	467
Alles im Code zusammenbringen	469
ADO mit der aktiven Arbeitsmappe verwenden	470
Grundlagen für das Arbeiten mit Textdateien	472
Eine Textdatei öffnen	473
Eine Textdatei lesen	474
In eine Textdatei schreiben	474
Eine Dateinummer abrufen	474
Lese-/Schreibposition ermitteln oder festlegen	475
Anweisungen für das Lesen und das Schreiben	476
Beispiele für das Arbeiten mit Textdateien	476
Daten aus einer Textdatei importieren	476
Einen Zellbereich in eine Textdatei exportieren	477
Eine Textdatei in einen Bereich importieren	478
Die Excel-Nutzung protokollieren	479
Eine Textdatei filtern	480
Gebäuchliche Dateioperationen durchführen	481
Die dateibezogenen VBA-Anweisungen verwenden	481
Das Objekt FileSystemObject verwenden	487
Zippen und Entzippen von Dateien	489
Dateien zippen	490
Eine Datei entzippen	491

Teil III

Mit UserForms arbeiten 493

Kapitel 12

Benutzerdefinierte Dialogfelder 495

Bevor Sie dieses UserForm erstellen ...	495
Eingabefelder	495
Die VBA-Funktion InputBox	495
Die Methode Application.InputBox	498
Die VBA-Funktion MsgBox	501

Die Excel-Methode GetOpenFilename	507
Die Excel-Methode GetSaveAsFilename	510
Abfrage eines Verzeichnisses	511
Die eingebauten Dialogfelder von Excel anzeigen	511
Ein Dateneingabeformular anzeigen	514
Das Datenformular bereitstellen	515
Ein Dateneingabeformular mit VBA anzeigen	516

Kapitel 13

UserForms – eine Einführung **517**

Wie Excel benutzerdefinierte Dialogfelder verarbeitet	517
Ein neues UserForm einfügen	518
Einem UserForm Steuerelemente hinzufügen	518
Steuerelemente in der Toolsammlung	519
Kontrollkästchen – CheckBox	520
Kombinationsfeld – ComboBox	520
CommandButton – Befehlsschaltfläche	521
Frame – Rahmen	521
Image – Bildfeld	521
Label – Beschriftung	521
ListBox – Listenfeld	521
MultiPage – Element mit mehreren Registerkarten	521
OptionButton – Optionsfelder	522
RefEdit – Zellbereich	522
ScrollBar – Bildlaufleiste	522
SpinButton – Drehfeld	522
TabStrip – Registerfeld	522
TextBox – Textfeld	522
ToggleButton – Umschalter	523
Anpassung von UserForm-Steuerelementen	524
Die Eigenschaften eines Steuerelements anpassen	526
Das Eigenschaften-Fenster	526
Gemeinsame Eigenschaften	528
Bequemlichkeit für Tastaturbenutzer	530
Ein UserForm anzeigen	532
Die Anzeigeposition anpassen	533
Ein nicht modales UserForm anzeigen	533
Ein UserForm basierend auf einer Variablen anzeigen	533
Ein UserForm laden	534
Ereignisprozeduren	534
Ein UserForm schließen	534
Ein UserForm erstellen: Ein Beispiel	536
Das UserForm erstellen	536
Code schreiben, mit dem das Dialogfeld angezeigt wird	539

Das Dialogfeld testen	540
Ereignisprozeduren hinzufügen	541
Das fertige Dialogfeld	543
UserForm-Ereignisse	543
Informationen über Ereignisse	543
UserForm-Ereignisse	545
Drehfeld-Ereignisse	545
Ein Drehfeld mit einem Textfeld kombinieren	547
Verweise auf UserForm-Steuerelemente	550
Die Werkzeugsammlung anpassen	551
Der Werkzeugsammlung neue Seiten hinzufügen	551
Steuerelemente anpassen oder kombinieren	551
Neue ActiveX-Steuerelemente hinzufügen	553
UserForm-Schablonen erstellen	554
Eine UserForm-Checkliste	555

Kapitel 14

UserForms – Beispiele

557

Ein UserForm als »Menü« anlegen	557
Befehlsschaltflächen in einem UserForm verwenden	557
Ein Listenfeld in einem UserForm verwenden	558
Auswahl von Bereichen in einem UserForm	560
Einen Begrüßungsbildschirm erstellen	561
Die Schließen-Schaltfläche eines UserForms deaktivieren	564
Die Größe eines UserForms ändern	565
Ein Blatt aus einem UserForm heraus vergrößern und blättern	567
Listenfeld-Techniken	569
Einem ListBox-Steuerelement Elemente hinzufügen	570
Das in einem Listenfeld ausgewählte Element ermitteln	575
Mehrfachauswahlen in einem Listenfeld bestimmen	576
Mehrere Listen in einem einzelnen Listenfeld	577
Übertragung von Listenfeldelementen	578
Elemente in einem Listenfeld verschieben	580
Mehrspaltige Listenfelder	582
Mit einem Listenfeld Zeilen in einem Arbeitsblatt auswählen	584
Mit einem Listenfeld ein Blatt aktivieren	586
Ein Listenfeld über ein Textfeld filtern	589
Das MultiPage-Steuerelement in einem UserForm verwenden	592
Ein externes Steuerelement verwenden	593
Ein Beschriftungsfeld animieren	596

Kapitel 15
Erweiterte UserForm-Techniken
599

Ein nicht modales Dialogfeld	599
Eine Fortschrittsanzeige anzeigen	603
Eine unabhängige Fortschrittsanzeige erstellen	604
Eine in ein UserForm integrierte Fortschrittsanzeige anzeigen	608
Eine nicht grafische Fortschrittsanzeige erstellen	612
Assistenten erstellen	615
Einrichtung des »MultiPage«-Steuerelements für den Assistenten	616
Dem UserForm des Assistenten die Schaltflächen hinzufügen	617
Die Schaltflächen des Assistenten programmieren	617
Abhängigkeiten in einem Assistenten programmieren	619
Die Aufgabe mit dem Assistenten erledigen	621
Die »MsgBox«-Funktion emulieren	622
»MsgBox«-Emulation: »MyMsgBox«-Code	623
Wie die »MyMsgBox«-Funktion funktioniert	624
Verwendung der »MyMsgBox«-Funktion	626
Ein UserForm mit beweglichen Steuerelementen	626
Ein UserForm ohne Titelleiste	628
Mit einem UserForm eine Symbolleiste simulieren	630
Ein Aufgabenfeld mit einem UserForm emulieren	632
Ein größenverstellbares UserForm	634
Mehrere UserForm-Steuerelemente mit einer Ereignisprozedur verarbeiten	638
Auswahl einer Farbe in einem UserForm	641
Ein Diagramm in einem UserForm anzeigen	644
Ein Diagramm als GIF-Datei speichern	645
Die »Picture«-Eigenschaft des Image-Steuerelements ändern	645
Ein halbtransparentes UserForm erstellen	645
Ein Puzzle auf einem UserForm	647
Video Poker auf einem UserForm	649

Teil IV
Excel-Anwendungen entwickeln
651
Kapitel 16
Add-Ins erstellen und verwenden
653

Was ist ein Add-In?	653
Vergleich eines Add-Ins mit einer Standardarbeitsmappe	653
Warum erstellt man Add-Ins?	654
Der Add-In-Manager von Excel	656
Ein Add-In erstellen	658

Ein Add-In-Beispiel	659
Aussagekräftige Informationen für das Beispiel-Add-In hinzufügen	661
Ein Add-In erstellen	661
Ein Add-In installieren	662
Das Add-In testen	663
Ein Add-In weitergeben	664
Ein Add-In ändern	665
XLAM- und XLSM-Dateien vergleichen	666
Die XLAM-Datei als Element der VBA-Collection	666
Sichtbarkeit von XLSM- und XLAM-Dateien	666
Arbeitsblätter und Diagrammblätter in XLSM- und XLAM-Dateien	667
Zugriff auf VBA-Prozeduren in einem Add-In	668
Add-Ins mit VBA manipulieren	671
Der AddIns-Collection ein Element hinzufügen	672
Ein Element aus der AddIns-Collection entfernen	673
AddIn-Objekteigenschaften	673
Auf ein Add-In als Arbeitsmappe zugreifen	676
AddIn-Objekt ereignisse	677
Die Leistung von Add-Ins optimieren	677
Spezielle Probleme mit Add-Ins	678
Sicherstellen, dass ein Add-In installiert ist	678
Von einem Add-In aus auf andere Dateien verweisen	681
Die richtige Excel-Version für Ihr Add-In erkennen	681

Kapitel 17

Mit dem Menüband arbeiten

683

Menüband – Grundlagen	683
Das Menüband anpassen	685
Dem Menüband eine Schaltfläche hinzufügen	685
Der Symbolleiste für den Schnellzugriff eine Schaltfläche hinzufügen	688
Die Einschränkungen der Menüband-Anpassungen verstehen	690
Ein benutzerdefiniertes Menüband erstellen	691
Einer vorhandenen Registerkarte eine Schaltfläche hinzufügen	691
Einer vorhandenen Registerkarte ein Kontrollkästchen hinzufügen	696
Menüband-Steuerelemente – Demo	701
Beispiel für ein dynamicMenu-Steuerelement	710
Weitere Informationen zur Anpassung des Menübands	713
VBA für das Menüband verwenden	714
Zugriff auf ein Menüband-Steuerelement	715
Mit dem Menüband arbeiten	716
Eine Registerkarte aktivieren	718
Eine Symbolleiste im alten Stil erstellen	718
Einschränkungen der alten Symbolleisten aus Excel 2007 und früher	718
Code für die Erstellung einer Symbolleiste	719

Kapitel 18
Mit Kontextmenüs arbeiten
723

CommandBar-Übersicht	723
CommandBar-Typen	723
Kontextmenüs auflisten	724
Verweise auf CommandBars	725
Verweis auf Steuerelemente in einer CommandBar	726
Eigenschaften von CommandBar-Steuerelementen	728
Anzeige aller Kontextmenüelemente	728
Mit VBA Kontextmenüs anpassen	731
Kontextmenü und das Single-Document Interface	731
Ein Kontextmenü zurücksetzen	734
Ein Kontextmenü deaktivieren	735
Einträge von Kontextmenüs deaktivieren	736
Dem Kontextmenü »Zelle« einen neuen Eintrag hinzufügen	736
Einem Kontextmenü ein Untermenü hinzufügen	739
Ein Kontextmenü auf eine einzelne Arbeitsmappe beschränken	741
Kontextmenüs und Ereignisse	742
Menüs automatisch hinzufügen und entfernen	742
Kontextmenüeinträge deaktivieren oder ausblenden	743
Ein kontextabhängiges Kontextmenü erstellen	743

Kapitel 19
Hilfe für Ihre Anwendungen bereitstellen
747

Hilfe für Ihre Excel-Anwendungen	747
Hilfesysteme, die Excel-Komponenten verwenden	749
Zellenkommentare für die Hilfe verwenden	750
Ein Textfeld für Hilfe verwenden	751
Hilfe auf einem Arbeitsblatt anzeigen	752
Hilfe in einem UserForm anzeigen	754
Hilfe in einem Webbrowser anzeigen	758
HTML-Dateien verwenden	758
Eine MHTML-Datei verwenden	759
Das HTML-Hilfesystem verwenden	761
Mit der Help-Methode HTML-Hilfe anzeigen	764
Ihrer Prozedur eine Hilfedatei zuordnen	765
Einer VBA-Funktion ein Hilfethema zuordnen	765

Kapitel 20
Klassenmodule
769

Was ist ein Klassenmodul?	769
Eingebaute Klassenmodule	770
Benutzerdefinierte Klassenmodule	770
Klassen und Objekte	770

Eine NumLock-Klasse erstellen	771
Ein Klassenmodul einfügen	772
Dem Klassenmodul VBA-Code hinzufügen	773
Die CNumLock-Klasse verwenden	774
Eigenschaften, Methoden und Ereignisse codieren	775
Eigenschaften von Objekten programmieren	775
Programmiermethoden für Objekte	777
Klassenmodul-Ereignisse	778
Anzeige eines QueryTable-Ereignisses	778
Eine Klasse für die Aufnahme von Klassen erstellen	782
Die Klassen »CSalesRep« und »CSalesReps« erstellen	782
Die Klassen »CInvoice« und »CInvoices« erstellen	784
Die übergeordneten Klassen mit Objekten füllen	786
Die Provisionen berechnen	787
Kapitel 21	
Kompatibilitätsprobleme	789
Was ist Kompatibilität?	789
Verschiedene Arten von Kompatibilitätsproblemen	790
Neue Funktionen vermeiden	791
Aber funktioniert es auch auf einem Mac?	793
64-Bit-Excel	794
Eine internationale Anwendung erstellen	795
Mehrsprachige Anwendungen	797
Sprachbetrachtungen für VBA	798
Lokale Eigenschaften verwenden	798
Systemeinstellungen identifizieren	799
Datums- und Zeiteinstellungen	801
Teil V	
Anhang	803
VBA-Anweisungen und Funktionsreferenz	804
Excel-Funktionen in VBA-Anweisungen aufrufen	807
Stichwortverzeichnis	812