

Einleitung

Der Einstieg in die Programmierung mit Excel beginnt für die meisten von uns mit einer bestimmten Aufgabe, die sich mit den Standardwerkzeugen von Excel nicht erledigen lässt. Welche Aufgabe das ist, ist für jeden von uns anders. Vielleicht soll für alle Zeilen eines Datasets eine eigene Arbeitsmappe erstellt werden. Oder die Aufgabe besteht darin, den Mail-Versand Dutzender Berichte zu automatisieren. Was immer auch für Sie der Auslöser war, Sie können sicher sein, dass jemand anderes mit den gleichen Anforderungen wie Sie seine Reise in die Welt von Excel-VBA gestartet hat.

Das Schöne an Excel-VBA ist, dass Sie kein Experte sein müssen, um mit dieser Programmiersprache Probleme zu lösen. Sie können einfach nur so viel lernen, bis Sie ein bestimmtes Problem gelöst haben. Oder Sie tauchen tiefer in die Materie ein und lernen alle möglichen Verfahren kennen, um mit den unterschiedlichsten Automatisierungsszenarien umgehen zu können.

Was immer auch Ihre persönlichen Ziele sein mögen, *Excel-VBA Power-Programmierung für Dummies* wird Ihnen helfen, die Mächtigkeit der Programmiersprache VBA zu nutzen, um Aufgaben zu automatisieren, smarter zu arbeiten und produktiver zu sein.

Themen dieses Buches

Der Schwerpunkt dieses Buches ist Visual Basic for Applications (VBA); dies ist die Programmiersprache, die in Excel (und in anderen Microsoft-Office-Anwendungen) eingebaut ist. Genauer ausgedrückt zeigt Ihnen dieses Buch, wie Sie Programme schreiben können, die unterschiedliche Aufgaben in Excel automatisieren. Das Buch behandelt dabei das gesamte Themenspektrum, angefangen bei der Aufzeichnung einfacher Makros bis hin zu ausgefallenen, benutzerorientierten Anwendungen und Werkzeugen.

Sie können das Buch so verwenden, wie es Ihnen am meisten zusagt. Sie können es von vorne nach hinten lesen oder im Buch blättern und hier und dort nützliche Tipps und Tricks finden. Die Programmierung mit VBA ist oft ein aufgabenorientiertes Unternehmen. Falls Sie also vor einer herausfordernden Aufgabe stehen, können Sie zuerst den Index konsultieren und dort nachschlagen, an welcher Stelle im Buch Ihr Problem behandelt wird.

Microsoft Visual Studio Tools für Office (VSTO) wird in diesem Buch nicht behandelt. Hierbei handelt es sich um eine Technologie, die die Programmiersprachen Visual Basic.NET und Visual C# verwendet. VSTO kann ebenfalls verwendet werden, um Excel und andere Microsoft-Office-Anwendungen zu steuern.

Wie Sie vielleicht wissen, steht Excel 2016 für verschiedene Plattformen zur Verfügung. So können Sie beispielsweise Microsoft Excel Web App im Browser verwenden; andere Versionen stehen für das iPad und andere Tablets bereit. Diese Versionen bieten jedoch keine Unterstützung für VBA. Mit anderen Worten: Dieses Buch behandelt die Desktopversion von Excel 2016 für Windows.

Was Sie wissen sollten

Dieses Buch ist kein Buch für Excel-Einsteiger. Falls Sie noch keine Erfahrungen mit Excel und Kenntnisse darüber besitzen, dann ist ein Buch wie *Excel 2016 für Dummies* eine bessere Wahl.

Damit Sie das meiste aus diesem Buch herausholen können, sollten Sie ein relativ erfahrener Excel-Anwender sein und wissen, wie Sie die folgenden Dinge erledigen:

- ✓ Arbeitsmappen erstellen, Arbeitsblätter einfügen, Dateien speichern und so weiter
- ✓ Sich in einer Arbeitsmappe bewegen
- ✓ Das Excel-Menüband verwenden
- ✓ Formeln eingeben
- ✓ Die Arbeitsblattfunktionen von Excel verwenden
- ✓ Zellen und Zellbereiche benennen
- ✓ Die grundlegenden Windows-Features verwenden, wie Techniken für die Dateiverwaltung und die Zwischenablage

Was Sie brauchen

Es ist wichtig, dass auf Ihrem Computer eine vollständige Installation von Excel vorhanden ist; falls Sie die in den fortgeschritteneren Kapiteln beschriebenen Kommunikationstechniken zwischen Excel und anderen Office-Anwendungen ausprobieren wollen, benötigen Sie eine Komplettinstallation von Microsoft Office.

Auch wenn das meiste Material mit älteren Versionen von Excel verwendet werden kann, gehen wir davon aus, dass Sie Excel 2016 verwenden. Falls Sie vorhaben, Anwendungen zu entwickeln, die mit älteren Excel-Versionen eingesetzt werden, empfehlen wir Ihnen dringend, die aktuellste Version von Excel zu verwenden, die bei Ihrer Zielgruppe im Einsatz ist.

Welche Windows-Version Sie verwenden, ist nicht so wichtig. Jeder Computer, auf dem Windows läuft, reicht aus; jedoch sind Sie mit einer schnellen Maschine und genügend Arbeitsspeicher besser dran. Excel ist ein umfangreiches Programm und seine Verwendung auf einem langsamen Computer mit minimaler Speicherausstattung kann schnell zu einer frustrierenden Erfahrung werden.

Dieses Buch behandelt keine der Excel-Versionen, die für den Mac zur Verfügung stehen.

Konventionen in diesem Buch

Nehmen Sie sich einen Moment Zeit und überfliegen Sie diesen Abschnitt, um sich mit ein paar typografischen Konventionen vertraut zu machen, die in diesem Buch verwendet werden.

Excel-Befehle

Excel verwendet ein kontextbezogenes Menüband. Die Begriffe, die ganz oben im Menüband stehen (wie **EINFÜGEN** und **ANSICHT**), werden *Registerkarten* genannt. Wenn Sie eine der Registerkarten anklicken, zeigt das Menüband Symbole und Schaltfläche der Befehle an, die zur Kategorie der Registerkarte passen. Jedes Symbol besitzt einen Namen, der häufig neben oder unter dem Symbol angezeigt wird. Auf den Registerkarten sind die Befehle/Schaltflächen in Gruppen zusammengefasst. Der Name der Gruppe wird ganz unten im Menüband angezeigt.

Um einen bestimmten Befehl anzugeben, wird in diesem Buch zuerst der Name der Registerkarte, dann der Name der Gruppe und abschließend der Name der Schaltfläche wiedergegeben. So wird beispielsweise der Befehl, mit dem der Zeilenumbruch einer Zelle umgeschaltet werden kann, folgendermaßen dargestellt:

START|AUSRICHTUNG|ZEILENUMBRUCH

Wenn Sie die erste Registerkarte anklicken, die mit **DATEI** beschriftet ist, öffnen Sie die Backstage-Ansicht. In der Backstage-Ansicht finden Sie die Befehle an der linken Seite des Fensters. Um einen Befehl der Backstage-Ansicht anzugeben, verwenden wir das Wort **DATEI**, gefolgt vom Namen des Befehls. So wird beispielsweise mit dem folgenden Befehl das Dialogfeld **EXCEL-OPTIONEN** geöffnet.

DATEI|OPTIONEN

Visual-Basic-Editor-Befehle

Visual Basic-Editor (VBE) ist das Fenster, in dem Sie mit Ihrem VBA-Code arbeiten. VBE verwendet die traditionelle Benutzeroberfläche mit Menüleiste und Symbolleisten. Wenn Sie beispielsweise im Buch die folgende Befehlssequenz sehen, bedeutet dies, das Menü **EXTRAS** zu öffnen und dort den Befehl **VERWEISE** anzuklicken:

EXTRAS|VERWEISE

Tastatureingaben

Sie müssen die Tastatur verwenden, um Daten einzugeben. Auch das Menü und die Dialogfelder lassen sich direkt über die Tastatur bedienen; diese Vorgehensweise ist praktisch, wenn sich Ihre Hände bereits an den Tasten befinden.

Eingabe

Eingaben, die Sie über die Tastatur vornehmen sollen, werden in Fettschrift angezeigt. Beispiel: Geben Sie in Zelle B51 die Formel **=SUMME(B2:B50)** ein.

Längere Eingaben werden in einer eigenen Zeile in einer nicht-proportionalen Schrift angezeigt. So können wir Sie beispielsweise auffordern, die folgende Formel einzugeben:

=SVERWEIS(Artikel;Preisliste;2)

VBA-Code

Sie finden in diesem Buch sowohl zahlreiche VBA-Codeschnipsel als auch Listings mit vollständigen Prozeduren. Für die Listings wird eine nicht-proportionale Schrift verwendet; jede Codezeile befindet sich im Listing in einer Zeile. (Ich habe diese Listings direkt im VBA-Modul kopiert und sie in mein Textverarbeitungsprogramm eingefügt.) Um den Code besser lesbar zu machen, haben wir häufig Tabstopps verwendet, um Einrückungen vorzunehmen. Diese Einrückungen sind optional, jedoch helfen sie dabei, Anweisungen zu erkennen, die zusammengehören.

Falls eine Codezeile in diesem Buch nicht in eine Zeile passt, haben wir die Standardsequenz von VBA für die Zeilenfortsetzung verwendet: Am Ende der Zeile steht dann ein Leerzeichen gefolgt von einem Unterstrich. Hiermit wird angegeben, dass diese Codezeile in der nächsten Zeile fortgesetzt wird. So handelt es sich bei den folgenden beiden Zeilen um eine einzeilige Code-Anweisung:

```
columnCount = Application.WorksheetFunction. _  
    CountA(Range("A:A")) + 1
```

Sie können diesen Code entweder exakt wie angegeben in zwei Zeilen eingeben oder auf einer einzigen Zeile, wobei Sie dann das Leerzeichen und den Unterstrich weglassen müssen.

Funktionen, Dateinamen und benannte Bereiche

Für die Namen der Arbeitsblatffunktionen von Excel werden Großbuchstaben verwendet. Beispiel: »Geben Sie in Zelle C20 eine SUMME-Formel ein.« Für die Namen von VBA-Prozeduren, Eigenschaften, Methoden und Objekten werden wir eine Mischung aus Groß- und Kleinschreibung verwenden, um die Namen so besser lesbar zu machen.

Was die Symbole bedeuten

Im gesamten Buch verwenden wir verschiedene Symbole, um Sie auf Stellen aufmerksam zu machen, die besonders wichtig sind.



Wir verwenden das Symbol »Nicht vergessen«, um Sie auf etwas Wichtiges hinzuweisen. Dies kann beispielsweise ein Konzept sein, mit dem Sie die vorliegende Aufgabe meistern können, oder grundlegende Hinweise, die dabei helfen, nachfolgende Informationen zu verstehen.



Mit dem Tippsymbol weisen wir auf effizientere Vorgehensweisen hin oder auf eine Technik, die nicht auf den ersten Blick ersichtlich ist.



Diese Symbole geben an, dass auf der Website zum Buch eine Beispieldatei verfügbar ist. Weitere Informationen finden Sie im Abschnitt »Über die Website zum Buch« weiter hinten in dieser Einleitung.



Wir verwenden das Symbol »Vorsicht«, wenn der Vorgang, den wir beschreiben, zu Problemen führen kann, falls Sie nicht vorsichtig sind.



Das Symbol »Querverweis« weist Sie auf andere Kapitel hin, in denen Sie ausführlichere Informationen zu einem Thema finden.

Wie dieses Buch aufgebaut ist

Die Kapitel dieses Buches sind in fünf Teilen organisiert.

Teil I: Einführung in Excel-VBA

In Teil I finden Sie eine Einführung in VBA. Sie lernen dort die Programmiergrundlagen, die Sie brauchen, um in Excel-VBA eigene Routinen und Funktionen zu erstellen. Kapitel 1 legt das Fundament und liefert Ihnen einen konzeptionellen Überblick zur Anwendungsentwicklung mit Excel. In den Kapiteln 2 bis 6 finden Sie alles, was Sie wissen müssen, um in VBA mit dem Schreiben von Programmcode loszulegen. Kapitel 7 rundet die Einführung in VBA mit zahlreichen, nützlichen Beispielen ab.

Teil II: Fortgeschrittene VBA-Techniken

In Teil II werden weitere Techniken vorgestellt, die häufig mit der Kategorie *fortgeschritten* versehen werden. Kapitel 8 und 9 beschreiben, wie Sie in VBA mit PivotTables und Diagrammen arbeiten (einschließlich Sparklines). Kapitel 10 stellt verschiedene Techniken vor, die Sie verwenden können, um von Excel aus mit anderen Anwendungen (wie beispielsweise Word oder Outlook) zu kommunizieren. Kapitel 11 schließt den zweiten Teil ab und beschreibt, wie Sie mit Dateien und externen Datenquellen arbeiten.

Teil III: Mit UserForms arbeiten

Die vier Kapitel in Teil III behandeln das Thema benutzerdefinierte Dialogfelder (*UserForms*). Kapitel 12 stellt einige eingebaute Alternativen zur Erstellung von benutzerdefinierten UserForms vor. In Kapitel 13 finden Sie eine Einführung in UserForms und die verschiedenen Steuerelemente, die Sie verwenden können. In Kapitel 14 und 15 präsentieren wir zahlreiche Beispiele für benutzerdefinierte Dialogfelder, angefangen von ganz einfachen, bis hin zu komplexeren.

Teil IV: Excel-Anwendungen entwickeln

Die Kapitel in Teil IV befassen sich mit wichtigen Aspekten bei der Programmierung von benutzerorientierten Anwendungen. Kapitel 16 beschreibt an einem praktischen Beispiel, wie Sie Add-Ins erstellen. In den Kapiteln 17 und 18 lernen Sie, wie Sie das Menüband und die Kontextmenüs von Excel anpassen können. Kapitel 19 stellt verschiedene Möglichkeiten vor, um Ihre Anwendung mit einer Onlinehilfe auszustatten. In Kapitel 20 beschreiben wir, wie Sie eigene Klassenmodule erstellen können. Kapitel 21 schließt Ihre Forschungsreise in die Welt der VBA-Programmierung mit Excel ab und Sie erhalten doch einige Informationen zu Fragen der Kompatibilität.

Teil V: Anhang

In Teil V finden Sie einen Anhang, den Sie als Referenz zu allen Anweisungen und Funktionen, die in VBA als Schlüsselwörter definiert sind, verwenden können.

Über die Website zum Buch

Zu fast allen Themen, die im Buch behandelt werden, finden Sie auf der Website zum Buch ein Beispiel. Sie können (und sollten) die zahlreichen nützlichen Beispiele dieses Buches unter dieser Adresse herunterladen:

<http://www.wiley-vch.de/publish/dt/books/ISBN3-527-71299-2>

Sie finden alle Beispiele des Buches in einer ZIP-Datei. Laden Sie die ZIP-Datei herunter, speichern Sie sie und entpacken Sie die Datei. Sie finden dann für jedes Kapitel einen eigenen Ordner (Kapitel01, Kapitel02 und so weiter). Falls für ein bestimmtes Beispiel mehrere Dateien benötigt werden, finden Sie diese in einem Unterordner des jeweiligen Kapitelordners.