

2. Auflage

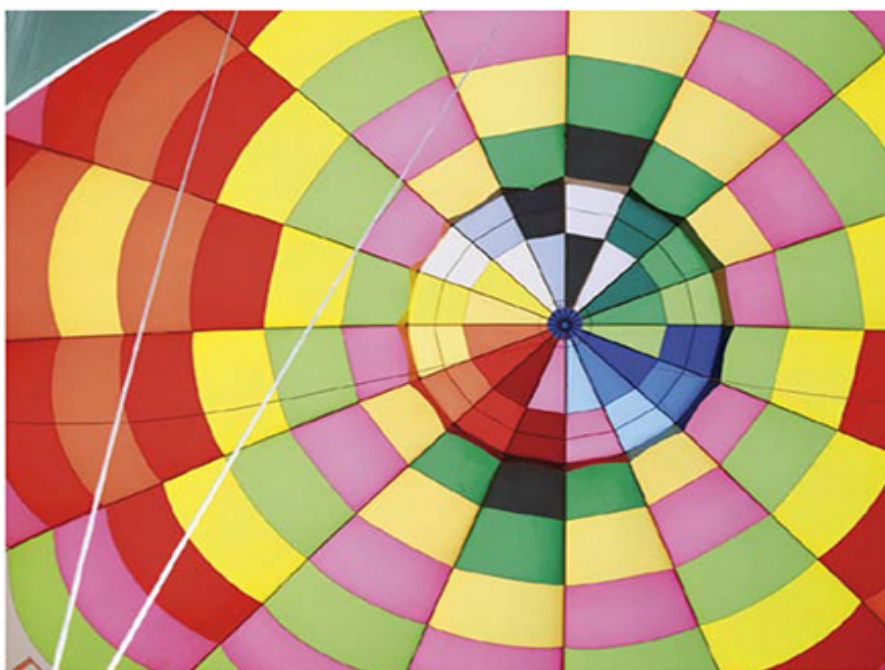
Richtig einsteigen:

Excel VBA- Programmierung

Für Excel 2007 bis 2016



Bernd Held



O'REILLY®

Richtig einsteigen: Excel VBA-Programmierung

- Für Microsoft Excel 2007 bis 2016

Bernd Held



Bernd Held

Lektorat: Ariane Hesse

Korrektorat: Sibylle Feldmann

Satz: Gerhard Alfes, www.mediaservice.tv

Herstellung: Susanne Bröckelmann

Umschlaggestaltung: Michael Oréal, www.oreal.de

unter Verwendung eines Fotos von francescodemarco/Fotolia.com

Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH,
www.mediaprint-druckerei.de

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der

Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über
<http://dnb.d-nb.de> abrufbar.

ISBN:

Print 978-3-96009-003-8

PDF 978-3-96010-027-0

epub 978-3-96010-028-7

mobi 978-3-96010-029-4

1. Auflage 2016

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint
»O'REILLY«. O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly
Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

Copyright © 2016 dpunkt.verlag GmbH

Wieblinger Weg 17

69123 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die
Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche
Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die
Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen
sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen
warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Die Informationen in diesem Buch wurden mit größter Sorgfalt erarbeitet. Dennoch können
Fehler nicht vollständig ausgeschlossen werden. Verlag, Autoren und Übersetzer übernehmen

keine juristische Verantwortung oder irgendeine Haftung für eventuell verbliebene Fehler und deren Folgen.

5 4 3 2 1 0

Inhalt

Einleitung

Wie dieses Buch aufgebaut ist

Download der Beispieldateien

Die Icons

Unterstützung für dieses Buch

Über den Autor

1 Die Entwicklungsumgebung von Excel

Excel für die Programmierung vorbereiten

Die Sicherheitsstufe heruntersetzen

Das Werkzeug Entwicklertools einblenden

Die Entwicklungsumgebung kennenlernen

Die Entwicklungsumgebung aufrufen

Der Projekt-Explorer

Das Eigenschaftenfenster

Das Codefenster

Das Direktfenster – die Testhilfe

Der Objektkatalog – das Nachschlagewerk

Der Makrorekorder – zu Beginn eine gute Hilfe

Die ersten Makros und deren Handhabung

Wert in eine Zelle einer Tabelle schreiben

Eine Meldung am Bildschirm ausgeben

Mehrzeilige Meldung am Bildschirm ausgeben

Eine Eingabe vom Anwender verlangen

Einen individuellen Tabellenkopf erstellen

Die wichtigsten Tastenkombinationen

Variablen und Konstanten

Variablen deklarieren

Konstanten einsetzen

Zusammenfassung

Die Lernkontrolle

2 Die wichtigsten Sprachelemente von Excel-VBA

Bedingungen erstellen und üben

Die Anweisungen If/Then/Else einsetzen

Die Anweisung Select Case einsetzen

Schleifen erstellen und verstehen

Die For...Next-Schleifen

Die For Each...Next-Schleifen

Die Schleife Do Until...Loop

Die Schleife Do While...Loop

Sonstige Sprachelemente

Die Struktur With

Zusammenfassung

Lernkontrolle

3 Das Objekt Range (Zellen und Bereiche programmieren)

Zellen und Bereiche formatieren

Zahlenformat und Schriftschnitt festlegen

Zellenfarbe und Schriftfarbe festlegen

Das Gitternetz und den Gesamtrahmen formatieren

Daten in Zellen konvertieren

Korrektur nach fehlerhaftem Datenimport

Unerwünschte führende und nachgestellte Leerzeichen entfernen

Bestimmte Zeichen in Zellen ersetzen/entfernen

- Die Position des Minuszeichens umstellen
- Verwendete Datumsformate vereinheitlichen
- Daten in Zellen und Bereichen suchen
 - Suche nach exakter Übereinstimmung
 - Suche nach exakter Übereinstimmung (Schreibweise spielt keine Rolle)
 - Suche auch in Teilen der Zelle (Schreibweise spielt keine Rolle)
 - Daten anhand eines eindeutigen Schlüssels suchen
- Bereiche Zelle für Zelle verarbeiten
 - Daten aus einem Bereich löschen
 - Extremwerte in einem Bereich ermitteln und kennzeichnen
 - Mehrere nicht zusammenhängende Bereiche verarbeiten
- Zusammenfassung
- Die Lernkontrolle

4 Das Objekt Worksheet (Tabellen programmieren)

- Tabellen dokumentieren, filtern und durchsuchen
 - Tabelleninhaltsverzeichnis erstellen und verlinken
 - Tabellen durchsuchen und dokumentieren
 - Tabellen filtern mit einem Kriterium
 - Tabellen filtern mit mehreren Kriterien
- Tabellen einrichten und schützen
 - Bildlaufbereiche für Tabellen festlegen
 - Bereiche in Tabellen sperren
 - Tabellenschutz für eine Tabelle einstellen und zurücksetzen
 - Alle Tabellen einer Mappe schützen
- Tabellenblätter anlegen, drucken und exportieren
 - Tabellen anlegen und benennen
 - Eine Tabelle drucken

- Alle sichtbaren Tabellen einer Mappe ausdrucken
- Tabelle als PDF ausgeben
- Eine Tabelle exportieren
- Individuelle Kopf- und Fußzeilen erstellen
- Tabellen verstecken oder löschen
 - Tabellen ein- und ausblenden
 - Alle Tabellen bis auf eine ausblenden
 - Tabellen löschen
- Zusammenfassung
- Die Lernkontrolle

5 Das Objekt Workbook (Arbeitsmappen programmieren)

- Arbeitsmappen abarbeiten und schließen
- Arbeitsmappe anlegen, verarbeiten, speichern und schließen
- Dokumenteigenschaften abfragen und auswerten
- Externe Verknüpfungen verarbeiten
 - Externe Verknüpfungen ermitteln
 - Verknüpfte Arbeitsmappen automatisch öffnen
 - Externe Verknüpfungen entfernen
- Arbeitsmappe löschen
- Sicherheitskopie einer Arbeitsmappe erstellen
- Daten aus einer anderen Mappe synchronisieren
- Zusammenfassung
- Die Lernkontrolle

6 Standardfunktionen nutzen, eigene Funktionen schreiben

- Die integrierten Tabellenfunktionen von Excel anzapfen
 - Einen Bereich summieren
 - Eine bedingte Summierung durchführen
 - Extremwerte ermitteln

Leere Tabellen aus einer Arbeitsmappe entfernen
Min- und Max-Wert in einem Bereich finden und einfärben
Leere Zeilen aus einer Tabelle entfernen

Eigene Funktionen schreiben

Der Aufbau einer Funktion

Aktuelle Arbeitsmappe ermitteln

Funktionen testen

Bestimmte Zeichen aus einer Zelle entfernen

Kalenderwoche nach DIN ermitteln

Die Existenz einer Tabelle prüfen

Die Existenz einer Datei prüfen

Die Existenz eines Verzeichnisses prüfen

Funktionen im Funktionsassistenten einsehen

Funktionen in eine andere Funktionskategorie hängen

Zusammenfassung

Die Lernkontrolle

7 Die Ereignisprogrammierung in Excel

Die Arbeitsmappenereignisse

Das Ereignis Workbook_Open

Das Ereignis Workbook_BeforeClose

Das Ereignis Workbook_BeforeSave

Das Ereignis Workbook_NewSheet

Das Ereignis Workbook_BeforePrint

Die wichtigsten Ereignisse auf Arbeitsmappenebene

Die Tabellenereignisse

Das Ereignis Worksheet_Change

Das Ereignis Worksheet_SelectionChange

Das Ereignis Worksheet_BeforeDoubleClick

..

Die wichtigsten Tabellenereignisse im Überblick
Excel über Tastenkombinationen steuern

Formelzellen in Festwertzellen wandeln

Makros zeitgesteuert starten

Excel nach einer bestimmten Zeit beenden

Makro zu einer bestimmten Uhrzeit starten

Zusammenfassung

Die Lernkontrolle

8 Die Dialogprogrammierung mit Excel

UserForms einfügen, beschreiben und anzeigen

Die Eigenschaften einer UserForm festlegen

Die wichtigsten Steuerelemente

Das Steuerelement TextBox

Das Steuerelement ComboBox

Das Steuerelement ListBox

Das Steuerelement CheckBox

Das Steuerelement OptionButton

Das Steuerelement Image

Zusammenfassung

Die Lernkontrolle

9 Das Fehler-Handling

Die Laufzeitfehler von Excel

Typische Stolperfallen bei der Programmierung

Einen Laufzeitfehler abfangen

Laufzeitfehler bereits im Voraus verhindern

Zusammenfassung

Die Lernkontrolle

10 Fragen & Antworten

Kapitel 1

Kapitel 2

Kapitel 3

Kapitel 4

Kapitel 5

Kapitel 6

Kapitel 7

Kapitel 8

Kapitel 9

Index

Einleitung

Wie dieses Buch aufgebaut ist

Sie wollten schon immer einmal mit Makros arbeiten? Sie möchten noch tiefer in Excel einsteigen, um damit Abläufe zu automatisieren, zu sichern und zu optimieren?

Sie haben Freude daran, sich mit zukunftssträchtigen und wertvollen Techniken zu beschäftigen? Wenn ja, dann ist dieses Buch das Richtige für Sie!

Dieses Buch führt Sie Schritt für Schritt richtig in die VBA-Programmiersprache ein. Die Grundlage für dieses Buch ist ein dreitägiger VBA-Kurs, den ich seit über 10 Jahren fast jeden Monat in Vaihingen-Enz und an anderen Standorten in Deutschland durchführe.

Nach Durcharbeiten dieses Buchs werden Sie in der Lage sein, große Teile Ihrer täglichen Arbeit zu automatisieren und lästige Routinearbeiten über Makros erledigen zu lassen. Mit diesem Buch als Grundlage werden Sie viel Spaß daran haben, auf diesem Gebiet weiter zu machen. Da Excel in fast jeder Firma vertreten und Automatisierung beinahe immer notwendig ist, werden Sie sich durch Ihr neu erworbenes Know-how viele Freunde machen.

Lernen Sie zu Beginn des Buchs zunächst einmal Ihr Werkzeug kennen. Dabei erfahren Sie, wie Sie sich in der Entwicklungsumgebung von Excel zurechtfinden, und Sie werden erfolgreich Schleifen und Verzweigungen anwenden können. Danach lernen Sie die Programmierung der wirklich wichtigen Excel-Objekte (Zellen, Tabellen und Mappen) kennen. Erfahren Sie, welche Methoden und Eigenschaften dieser Objekte dringend gebraucht werden, um die wichtigsten Arbeiten automatisiert und schnell erledigen zu können.

Am Ende eines jeden Kapitels erwarten Sie Übungsaufgaben und Verständnisfragen, sodass Sie sicher sein können, den optimalen Nutzen

aus diesem Buch zu ziehen.

Die Beispiele, die im Buch vorgestellt werden, wurden mit der Excel-Version 2016 geschrieben. Alle hier vorgestellten Makros können aber genauso gut in allen Versionen ab Excel 2002 genutzt werden.

Sollten Sie Fragen oder Anregungen haben, dann scheuen Sie sich nicht, mich unter meiner Mailadresse

info@held-office.de

zu kontaktieren.

Download der Beispieldateien

Die für dieses Buch bereitgestellten Dateien können Sie sich unter folgender Adresse herunterladen:

www.Held-office.de/Downloads/RichtigEinsteigen.zip

Sie finden sie auch direkt auf der Verlagsseite:

http://downloads.oreilly.de/9783960090038

Die Icons

Dieses Buch führt Sie in die VBA-Programmierung mit Excel ein. Schon nach kurzer Einarbeitungszeit werden Sie über das nötige Know-how verfügen, um ansprechende Ergebnisse zu erzielen. Zugleich erwerben Sie das entsprechende Know-why, das heißt, Sie erfahren immer auch, warum etwas in einer bestimmten Weise funktioniert. Hin und wieder empfiehlt es sich für eine bessere Übersicht, bestimmte Informationen in eigenen Absätzen auszuzeichnen.



Der Textmarker weist Sie auf etwas hin, worauf Sie unbedingt achten sollten.



Hier erfahren Sie, wie Sie etwas besonders einfach und elegant erledigen können.



Absätze mit diesem Icon geben Ihnen wichtige Hintergrundinformationen und erklären, warum etwas in einer bestimmten Weise funktioniert.

Unterstützung für dieses Buch

Der Verlag hat auch von seiner Seite alles unternommen, um die Richtigkeit des Buchinhalts sicherzustellen. Etwaige Korrekturen und Änderungen finden Sie unter folgender Adresse:

<http://downloads.oreilly.de/9783960090038>

Kommentare, Fragen und Anregungen können Sie ebenfalls an den Verlag schicken. Wenden Sie sich dazu an kommentar@oreilly.de

Bitte beachten Sie, dass der Verlag keinen Support für Software-Produkte anbieten kann.

Über den Autor



Mein Name ist Bernd Held. Ich wurde am 02.04.1969 geboren, bin verheiratet und Vater von zwei Kindern. Während meines Abiturs und in der Zeit danach war ich Leistungssportler, wurde unter anderem zwei Mal Deutscher Jugendmeister über 400 Meter Hürden und nahm an

Europa- und Weltmeisterschaften teil. Nach meiner sportlichen Laufbahn habe ich mich ins Berufsleben gestürzt und mich auf die Themen Excel/Access, individuelle VBA-Schulungen/VBA-Kurse sowie VBA-Programmierung spezialisiert.

Von Haus aus bin ich gelernter Informatiker. Zunächst war ich zwei Jahre bei einer kleinen Softwarefirma in der Entwicklung und danach sechs Jahre bei T-Systems im Controlling beschäftigt. Dort war ich verantwortlich für das Berichtswesen, die Leistungsverrechnung, das Erstellen von betrieblichen Auswertungen und Wirtschaftlichkeitsrechnungen sowie für die Erstellung neuer Controlling-Tools auf Basis von Microsoft Office. Im Januar 2002 folgte dann der Schritt in die Selbstständigkeit.

Seit dieser Zeit konzentriere ich mich auf die Auftragsprogrammierung, die Unternehmensberatung sowie das Schreiben von Fachartikeln und Computerbüchern. Einige Bücher von mir wurden bereits ins Russische, Tschechische und Englische übersetzt. Weitere Aufgabengebiete sind das fachliche Überarbeiten von Computerbüchern sowie die Durchführung von VBASchulungen. Zu meinen Spezialgebieten zählen Excel, VBA-Programmierung und Access. Acht Jahre in Folge wurde ich als MVP (Most Valuable Professional) für den Bereich Excel von Microsoft ausgezeichnet.

Seit 2008 arbeite ich neben meinen drei Angestellten mit einem eigenen Team aus Experten erfolgreich zusammen. Wir führen in erster Linie Programmierprojekte und Schulungen durch, sind in der Beratung tätig und schreiben Bücher sowie Artikel für diverse Fachbuchverlage.

Kapitel 1

Die Entwicklungsumgebung von Excel

In diesem Kapitel:

Excel für die Programmierung vorbereiten

Die Entwicklungsumgebung kennenlernen

Die ersten Makros und deren Handhabung

Die wichtigsten Tastenkombinationen

Variablen und Konstanten

Zusammenfassung

Die Lernkontrolle

Excel für die Programmierung vorbereiten

Bevor Sie richtig mit Excel-VBA einsteigen können, müssen Sie Microsoft Excel erst einmal für den Gebrauch der zukünftigen Makros einrichten.

Die Sicherheitsstufe heruntersetzen

Seit der Version Excel 2007 ist der Umgang mit Makros in Office etwas erschwert worden. Standardmäßig ist nach der Installation des Office-Pakets in Excel die höchste Sicherheitsstufe eingestellt. In dieser

Einstellung sind die Nutzung und das Schreiben von Makros nicht möglich.

Bereiten Sie Excel für die zukünftigen Aufgaben einmalig wie folgt vor:

1. Starten Sie Microsoft Excel.
2. Klicken Sie im Menüband unter *Datei* auf den Befehl *Optionen*.
3. Im Dialog *Excel-Optionen* wählen Sie die Rubrik *Trust Center* aus.
4. Klicken Sie auf die Schaltfläche *Einstellungen für das Trust Center*.
5. Wählen Sie im Dialog *Trust Center* die Rubrik *Makroeinstellungen* aus.
6. Aktivieren Sie in den *Makroeinstellungen* die Option *Alle Makros aktivieren*.
7. Setzen Sie einen Haken in das Kontrollkästchen *Zugriff auf das VBA-Projektmodell vertrauen*.

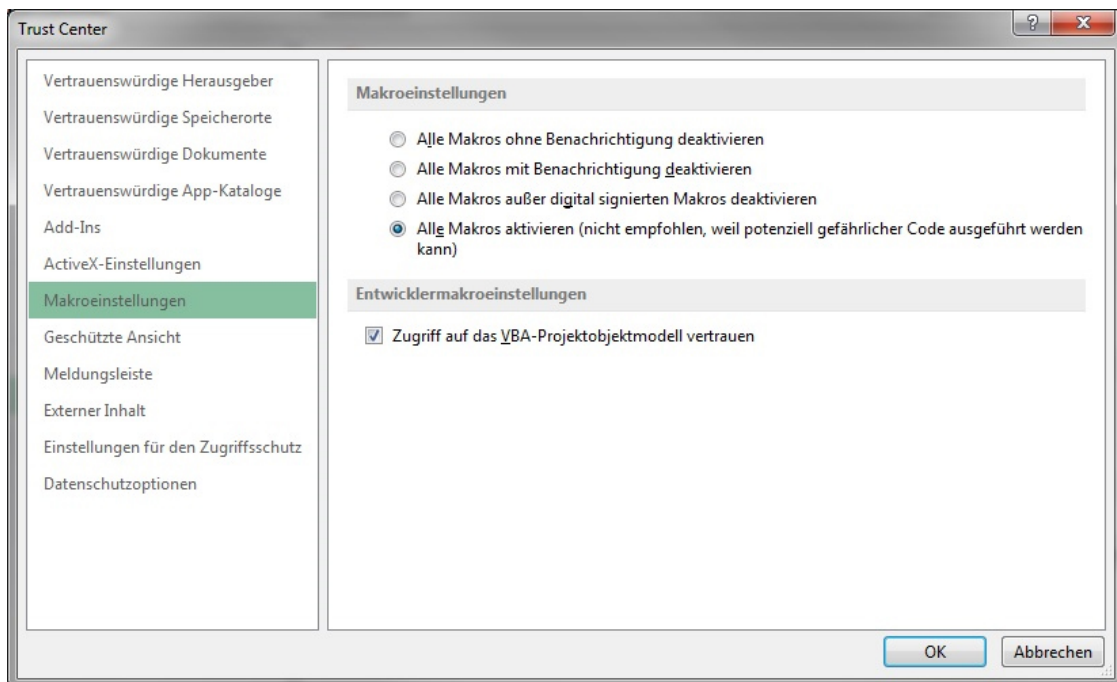


Abbildung 1.1: Die Makroeinstellungen anpassen und den Zugriff auf das VBA-Objektmodell erlauben.

8. Bestätigen Sie diese Einstellung, indem Sie die geöffneten Dialoge jeweils mit *OK* beenden.

Nachdem Sie diese Einstellungen vorgenommen haben, können Sie zukünftig mit Makros arbeiten, außerdem können Sie Dateien, die Makros enthalten, ebenfalls nutzen.



Der Haken im Trust Center bei *Zugriff auf das VBA-Projektmodell vertrauen* versetzt Sie in die Lage, zu einem späteren Zeitpunkt, wenn Sie schon etwas geübter im Umgang mit VBA sind, Makros zu schreiben, mit denen Sie beispielsweise Makros aus anderen Excel-Dateien ersetzen und bearbeiten können. Sie können aufgrund dieser Trust-Einstellung dann im Prinzip zur Laufzeit auf Makros zugreifen sowie ganze Programmteile austauschen und durch neue Makros ersetzen.

Das Werkzeug Entwicklertools einblenden

Um praktikabel mit Makros umgehen zu können, bietet Microsoft eine eigene Registerkarte für die Verwaltung und Programmierung von Makros in der Oberfläche von Excel an, die zunächst jedoch dem Standardanwender verborgen bleibt. Diese Registerkarte trägt den Namen *Entwicklertools*. Mithilfe der Werkzeuge auf dieser Registerkarte können Sie beispielsweise Schaltflächen in Tabellen einfügen und diesen Schaltflächen Makros zuweisen, um diese später mit einem Klick auf die Schaltfläche starten zu können. Des Weiteren beinhaltet dieser Werkzeugkasten die Möglichkeit, Makros mittels des Makrorekorders automatisch aufzuzeichnen.

Blenden Sie das Menüband *Entwicklertools* wie folgt ein:

1. Klicken Sie im Menüband unter *Datei* auf den Befehl *Optionen*.

2. Im Dialog *Excel-Optionen* wählen Sie die Rubrik *Menüband anpassen* aus.
3. Aktivieren Sie im Feld *Hauptregisterkarten* das Kontrollkästchen *Entwicklertools*.

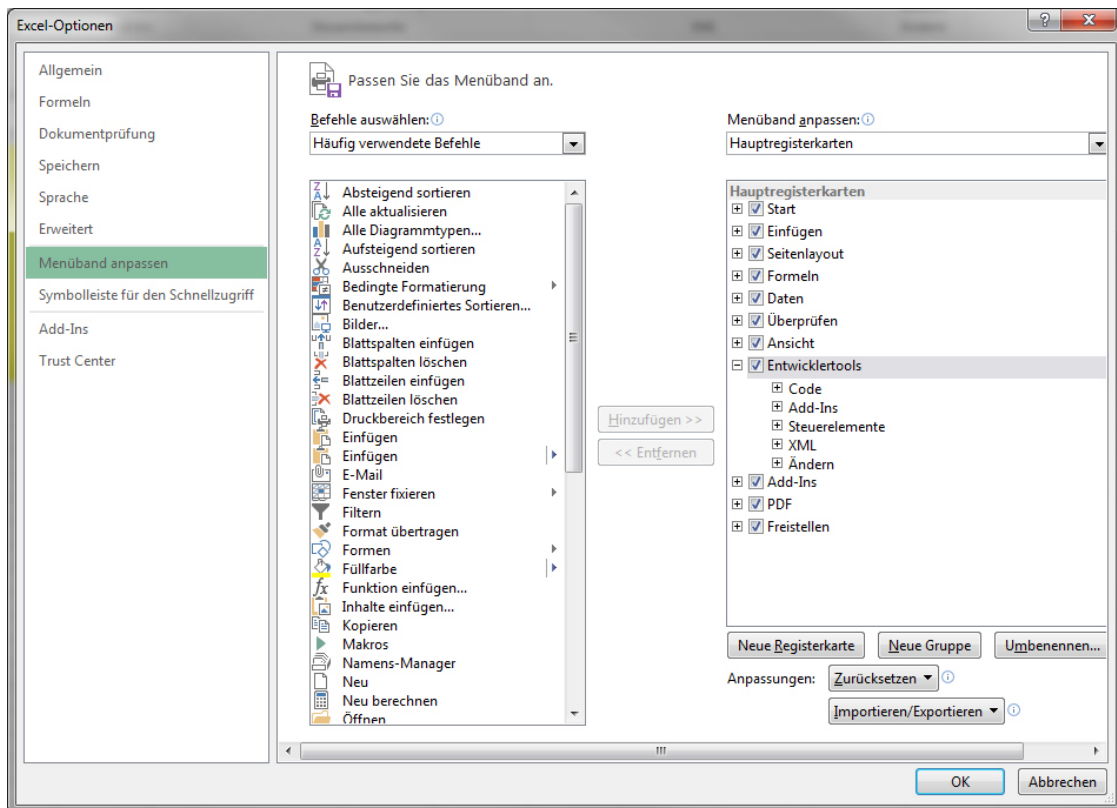


Abbildung 1.2: Das Werkzeug Entwicklertools einblenden.

4. Bestätigen Sie Ihre Einstellung mit einem Klick auf *OK*.

In der Excel-Oberfläche wird jetzt ein neues Menüband mit dem Namen *Entwicklertools* angeboten. Die wichtigsten Funktionen aus diesem Register werden auf den folgenden Seiten nach und nach beschrieben.

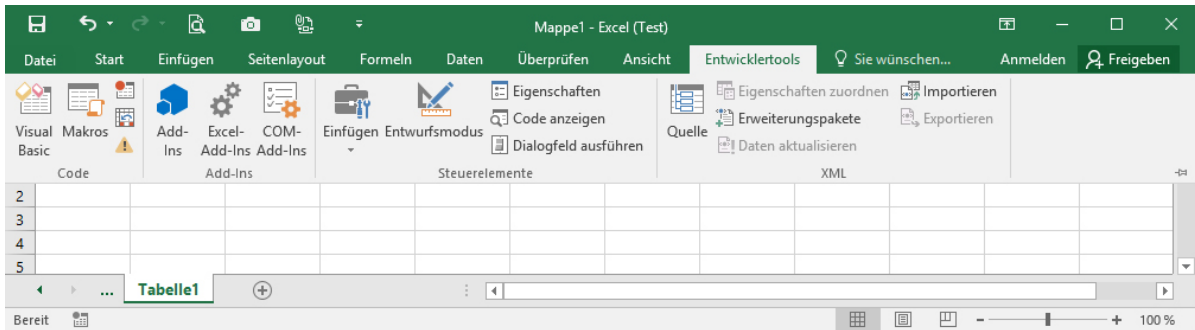


Abbildung 1.3: Das Register *Entwicklertools* steht für die Bearbeitung und Verwaltung der Makros bereit.

Die Entwicklungsumgebung kennenlernen

Erfahren Sie auf den nächsten Seiten alles, was Sie von Beginn an brauchen, damit Sie Ihre Makros schnell und sicher erfassen, starten und testen können.

Die fertig ausgefüllte Arbeitsmappe *Start.xlsm* mit allen folgenden Beispielen können Sie unter dieser Adresse herunterladen: <http://downloads.oreilly.de/9783960090038>.

Die Entwicklungsumgebung aufrufen

Wagen Sie den ersten Sprung in die Entwicklungsumgebung von Excel, indem Sie im Register *Entwicklertools* den Befehl *Visual Basic* anklicken. Alternativ dazu können Sie sich die Tastenkombination **Alt + F11** merken, die Sie ebenfalls direkt in die Entwicklungsumgebung von Microsoft Excel bringt.

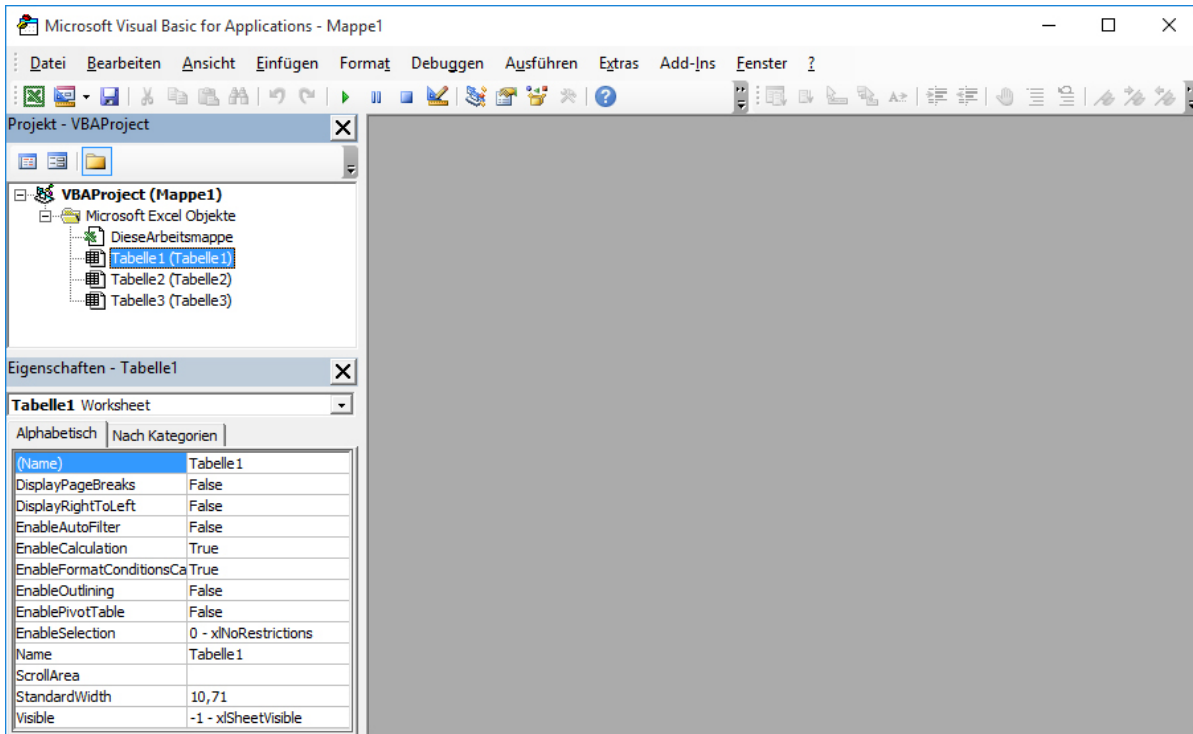


Abbildung 1.4: Die Entwicklungsumgebung von Microsoft Excel.

Die Entwicklungsumgebung in Excel-VBA beinhaltet mehrere Fenster, über die Sie Makros einsehen, erfassen und testen können.

Der Projekt-Explorer

Standardmäßig oben links ist der sogenannte Projekt-Explorer zu finden. Sollte dieser Explorer nicht angezeigt werden, können Sie ihn über das Menü *Ansicht* und den Befehl *Projekt-Explorer* einblenden. Alternativ rufen Sie den Projekt-Explorer über die Tastenkombination **Strg** + **R** auf.

Der Projekt-Explorer zeigt Ihnen alle aktuell geöffneten Arbeitsmappen sowie die darin enthaltenen Tabellen an.

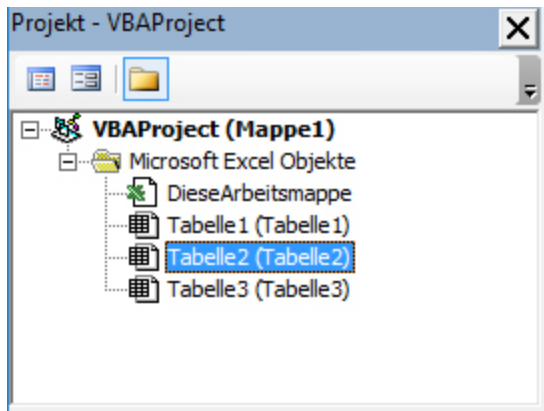


Abbildung 1.5: Der Projekt-Explorer gibt Auskunft über den Inhalt einer Mappe.


Die beiden ersten Symbole *Code anzeigen* und *Objekt anzeigen* werden später für Sie von Interesse sein, wenn Sie beispielsweise eigene benutzerfreundliche Dialoge entwerfen und zwischen den beiden Ebenen Objekt und Code hin- und herspringen müssen.

Das doppelt belegte Symbol *Ordner wechseln* ist eine reine Ansichtsoption. Damit wird die Ansicht der Objekte in der Arbeitsmappe angezeigt. Die standardmäßig eingestellte Ansicht ordnet die Objekte schön ordentlich in Rubriken an. Mit einem weiteren Klick auf dieses Symbol werden alle Objekte alphabetisch und ohne Darstellung in Rubriken angezeigt. Hier ist der Übersichtlichkeit halber die Standardeinstellung zu empfehlen.

Der Projekt-Explorer arbeitet sehr eng mit dem darunterliegenden Fenster, dem Fenster *Eigenschaften*, zusammen. Diese beiden Werkzeuge müssen in einem Atemzug genannt werden, da sie sich einander bedingen. Genau aus diesem Grund sind beide Fenster standardmäßig untereinander angeordnet.

Das Eigenschaftenfenster

Direkt unterhalb des Projekt-Explorers befindet sich standardmäßig das Eigenschaftenfenster. Sollte es nicht eingeblendet sein, können Sie es

über den Menübefehl *Ansicht/Eigenschaftenfenster* einblenden. Alternativ dazu können Sie auch die Taste  drücken, um das Eigenschaftenfenster einzublenden.

Wie gerade schon erwähnt, sind Explorer und Eigenschaftenfenster als Einheit zu verstehen. Wenn Sie beispielsweise im Projekt-Explorer die *Tabelle1* anklicken, werden im darunterliegenden Eigenschaftenfenster alle Eigenschaften zu dieser Tabelle angezeigt. Durch Setzen dieser Eigenschaften können Verhalten und Aussehen der Tabellen beeinflusst werden.



Generell können Eigenschaften entweder im Eigenschaftenfenster eingestellt oder auch direkt durch Makros gesetzt werden.

Die wichtigsten Eigenschaften lernen Sie anhand von praktischen Aufgaben auf den folgenden Seiten kennen.

Tabellen ein- und ausblenden

Mithilfe des Eigenschaftenfensters lassen sich in Excel Tabellen ein- und ausblenden. Dabei erleben Sie gleich zu Beginn eine kleine Überraschung. Achten Sie darauf, dass Ihre Arbeitsmappe mehr als nur eine Tabelle enthalten muss.

Gehen Sie wie folgt vor, um beispielsweise *Tabelle1* auszublenden:

1. Klicken Sie im Projekt-Explorer auf das Objekt *Tabelle1* (*Tabelle1*).
2. Klicken Sie im Eigenschaftenfenster ganz unten rechts neben die Eigenschaft *Visible*.

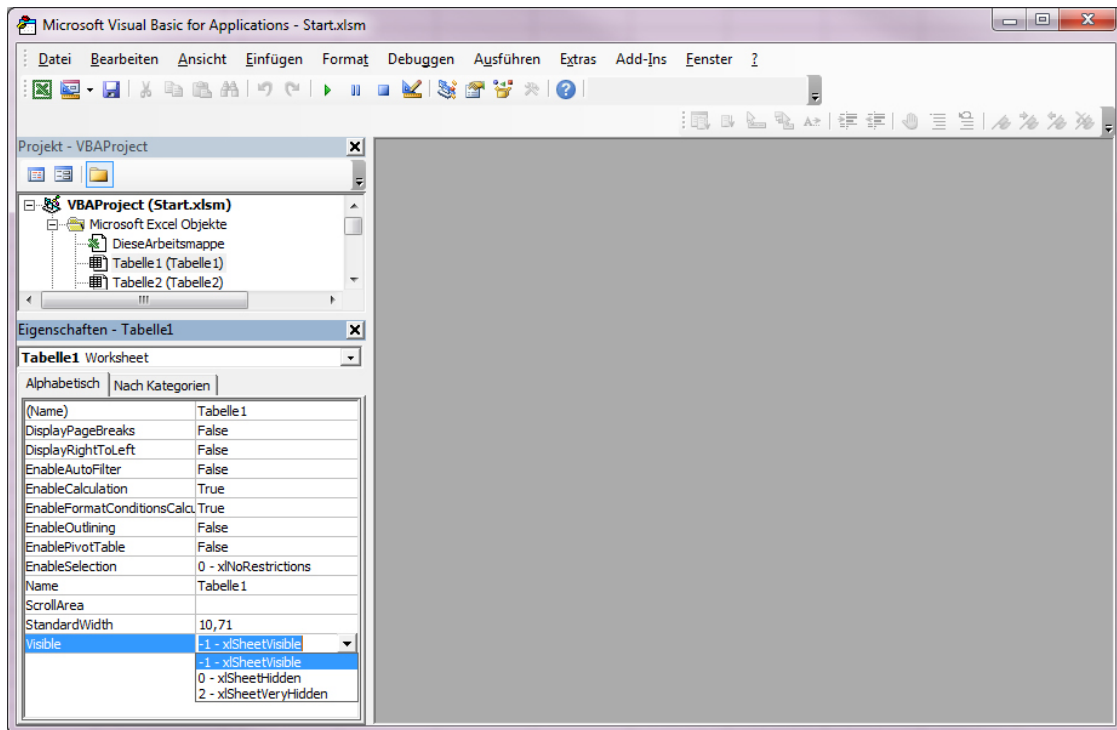


Abbildung 1.6: Die Eigenschaft *Visible* bietet drei Zustände der Sichtbarkeit an.

3. Wählen Sie aus dem Drop-down die Konstante *0 – xlSheetHidden*.

Jetzt ist *Tabelle1* ausgeblendet. Der Mauszeiger springt dabei, was zunächst vielleicht etwas irritiert, automatisch auf *Tabelle2*. Um den Status von *Tabelle1* wieder einzusehen bzw. zu ändern, klicken Sie einfach auf das Objekt *Tabelle1 (Tabelle1)* im Projekt-Explorer und weisen im Eigenschaftfenster wieder die Konstante *-1- xlSheetVisible* zu.



Sicher ist Ihnen nicht entgangen, dass das Eigenschaftfenster bei der Eigenschaft *Visible* eine weitere Konstante anbietet: *2 - xlSheetVeryHidden*. Diese Konstante sorgt dafür, dass die so eingestellte Tabelle sicher ausgeblendet wird. Das bedeutet, dass der »normale« Anwender keine Möglichkeit hat, diese Tabelle über die Oberfläche von Excel einzublenden. Tabellen werden in der Praxis gern sicher versteckt,

wenn es darum geht, Daten zu verbergen, damit diese nicht gelöscht werden können.



In einer Arbeitsmappe muss mindestens eine Tabelle immer eingeblendet sein. Sie können daher nicht alle Tabellen ausblenden!

Standardspaltenbreite festlegen

Über die Eigenschaft *StandardWidth* im Eigenschaftfenster können Sie die Standardspaltenbreite der Tabelle festlegen. Standardmäßig ist die Breite einer Spalte mit 10,71 festgelegt.



Eine Einheit der Spaltenbreite entspricht der Breite eines Zeichens in der standardmäßig eingestellten Schriftart. Je nach eingestellter Schriftart ist eine Umrechnung mehr oder weniger genau. Wenn Sie eine Umrechnung in Millimeter durchführen möchten, kann man gedanklich näherungsweise die Breite nach der Formel $-0,71 + 5,1425 * \text{Wunsch in mm} / 10$ berechnen. Hier wäre natürlich eine einfachere Umrechnung wünschenswert.






Zu jeder Eigenschaft können Sie übrigens die Taste F1 drücken, um weiterführende Informationen in der Onlinehilfe abzurufen.

Die ScrollArea setzen

Unter einer ScrollArea versteht man den Bereich einer Tabelle, in dem sich der »normale« Anwender aufhalten darf. Dieser Aufenthaltsbereich kann über das Eigenschaftfenster festgelegt werden.

Setzen Sie einmal testweise die Eigenschaft *ScrollArea*, indem Sie wie folgt vorgehen:

1. Erfassen Sie im Eigenschaftenfenster bei der Eigenschaft *ScrollArea* den Bezug *A1:D10*.
2. Bestätigen Sie die Eingabe mit . Die Eingabe wird augenblicklich mit absoluten Bezügen umfasst. So wird aus der Eingabe *A1:D10* die Eingabe *\$A\$1:\$D\$10*.
3. Wechseln Sie, nachdem Sie die *ScrollArea* gesetzt haben, über die Tastenkombination  +  in die Standardoberfläche von Microsoft Excel.
4. Wechseln Sie zu *Tabelle1*.
5. Versuchen Sie, den Bereich *A1:D10* zu verlassen. Es wird Ihnen nicht gelingen.



In der Praxis wird diese Technik gern verwendet, um sensible Daten zu schützen. Daher können Sie, bevor Sie diese Eigenschaft verwenden, im Vorfeld Daten in einem entlegenen Teil der Tabelle erfassen und anschließend den Aufenthaltsbereich einschränken. So kommt der Standardanwender von Excel nicht an Ihre Daten und kann sie daher auch nicht ändern.

Leider wird ein geänderter Wert der Eigenschaft *ScrollArea* nicht aufrechterhalten, wenn Sie die Eigenschaft einstellen und danach die Arbeitsmappe speichern und schließen. Nach erneutem Öffnen der Arbeitsmappe ist die vorher gesetzte Eigenschaft wieder weg. Was nun tun, um sicherzustellen, dass diese Eigenschaft nach dem Öffnen der Arbeitsmappe gesetzt ist?

Die Lösung ergibt sich direkt aus dem Verhalten dieser Eigenschaft. Wenn der Eigenschaftswert nach dem Öffnen der Arbeitsmappe nicht mehr vorhanden ist, müssen Sie beim Öffnen dafür sorgen, dass er automatisch gesetzt wird.

Um die *ScrollArea* dauerhaft einzurichten, greifen Sie in die Trickkiste und richten ein sogenanntes Ereignis ein. Gehen Sie dabei wie folgt vor:

1. Klicken Sie im Projekt-Explorer doppelt auf das Objekt *DieseArbeitsmappe*.
2. Wählen Sie im rechts daneben eingeblendeten Fenster im linken Drop-down-Listefeld den Eintrag *Workbook*. Daraufhin wird automatisch das Ereignis *Workbook_Open* eingestellt.
3. Ergänzen Sie den noch leeren Rahmen des Ereignisses wie in Abbildung 1.7 gezeigt.

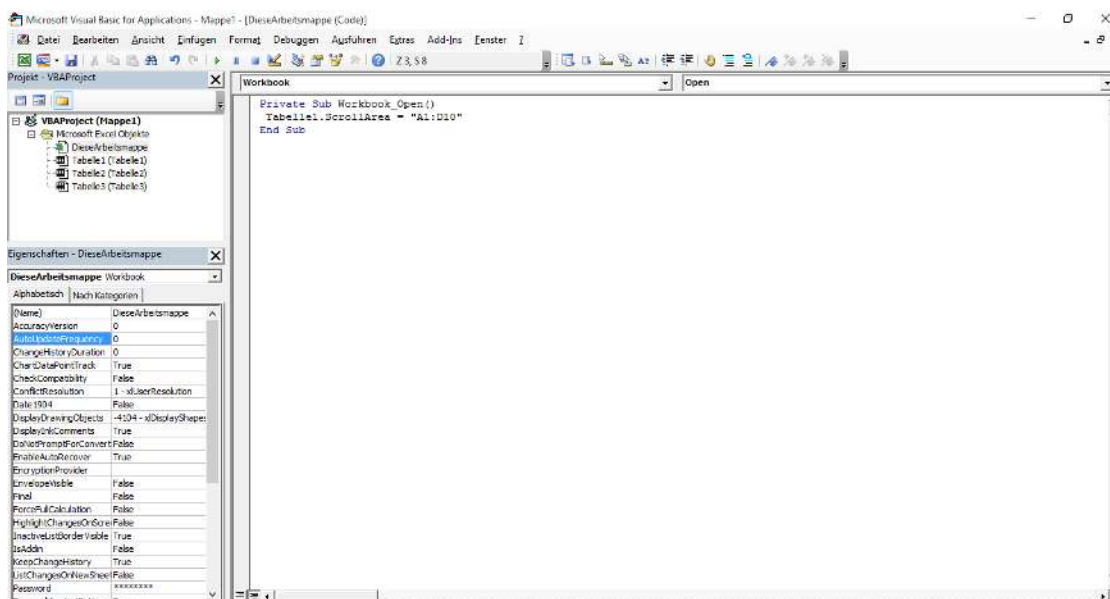


Abbildung 1.7: Das Ereignis Workbook_Open wird automatisch beim Öffnen der Arbeitsmappe ausgeführt.

4. Speichern Sie Ihre Arbeitsmappe. Danach schließen und öffnen Sie die Arbeitsmappe erneut. Sie werden feststellen, dass die ScrollArea jetzt »dauerhaft« eingestellt ist.

Auf diese Art und Weise können Sie zu jeder Tabelle ganz individuell eine ScrollArea einrichten.



Möchten Sie beispielsweise eine Tabelle gänzlich unveränderbar gestalten, ist die kleine ScrollArea durch den Zellenbezug *A1* zu definieren. In einer Tabelle, in der dieser Bezug angegeben wird, ist keinerlei




»Bewegung« möglich – auch eine Art von Datenschutz, meinen Sie nicht auch?



Eine ScrollArea ist ein zusammenhängender Bereich. Sie können nicht mehrere ScrollAreas auf einer Tabelle definieren. Wie Sie diese Standardeinstellung umgehen können, verrate ich Ihnen in Kapitel 7 »Ereignisprogrammierung in Excel«.

Den Namen einer Tabelle festlegen

Wenn Sie im Eigenschaftenfenster genau hinsehen, erkennen Sie, dass es zwei Eigenschaften gibt, um den Namen einer Tabelle festzulegen – die beiden Eigenschaften *Name* und *(Name)* im Eigenschaftenfenster. Standardmäßig sind beide Eigenschaften mit dem gleichen Wert belegt. Dieser doppelte Name spiegelt sich auch im Projekt-Explorer wider. Was hat es nun auf sich mit den zwei Namen? Die Beantwortung dieser Frage ist derart wichtig, dass Sie dazu erst einmal ein Beispiel erhalten. Gehen Sie wie folgt vor:

1. Wechseln Sie aus der Entwicklungsumgebung heraus in die Normalansicht von Excel.
2. Klicken Sie direkt im Tabellenregister unten den Namen *Tabelle1* doppelt an.
3. Erfassen Sie den Namen *Test* und bestätigen Sie mit .
4. Wechseln Sie über die Tastenkombination  +  zurück in die Entwicklungsumgebung.
5. Betrachten Sie das Ergebnis in Abbildung 1.8.

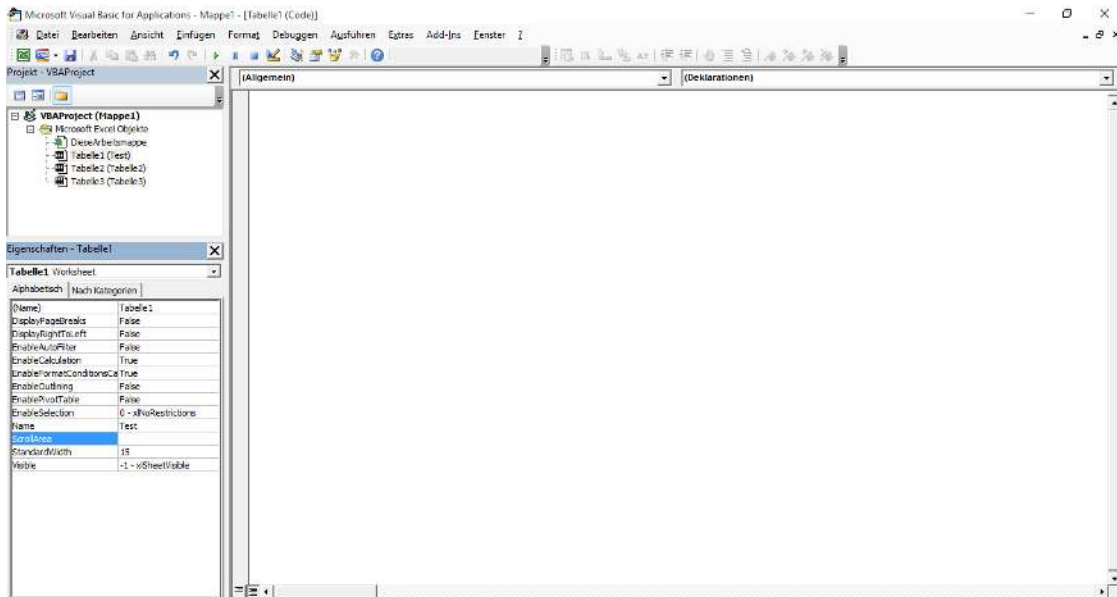


Abbildung 1.8: Wir stellen eine Abweichung der Tabellennamen fest.

Wie Sie sehen können, haben wir nun zwei verschiedene Tabellennamen. Beide Namen können auch jederzeit direkt über das Eigenschaftenfenster geändert werden. Worin besteht nun der Unterschied zwischen den beiden Tabellennamen, und welchen Tabellennamen sollten Sie für die Programmierung verwenden?

Nun, merken Sie sich Folgendes: Benutzen Sie immer den Tabellennamen, der im Projekt-Explorer an erster Stelle steht. Bei diesem Namen spricht man vom sogenannten Codenamen der Tabelle. Durch die Benutzung des Codenamens ersparen Sie sich gleich zu Beginn Ihrer Karriere als Entwickler viel Arbeit und Ärger. Auf die Vor- und Nachteile gehe ich im weiteren Verlauf des Kapitels noch genauer ein.



Bevor Sie überhaupt mit der Programmierung von Makros beginnen, sollten Sie die Tabellen so benennen, dass beide Namen idealerweise wieder gleich lauten, dabei aber auch sprechend sind. In der Praxis hat es sich bewährt, beim Tabellennamen ein Kürzel vorzugeben, damit auch im Makro später direkt erkannt

werden kann, dass es sich um eine Tabelle handelt. So könnten Sie beispielsweise die ehemalige *Tabelle1* in beiden Eigenschaftsfeldern mit dem Namen *tbl_ErsteMakros* benennen.

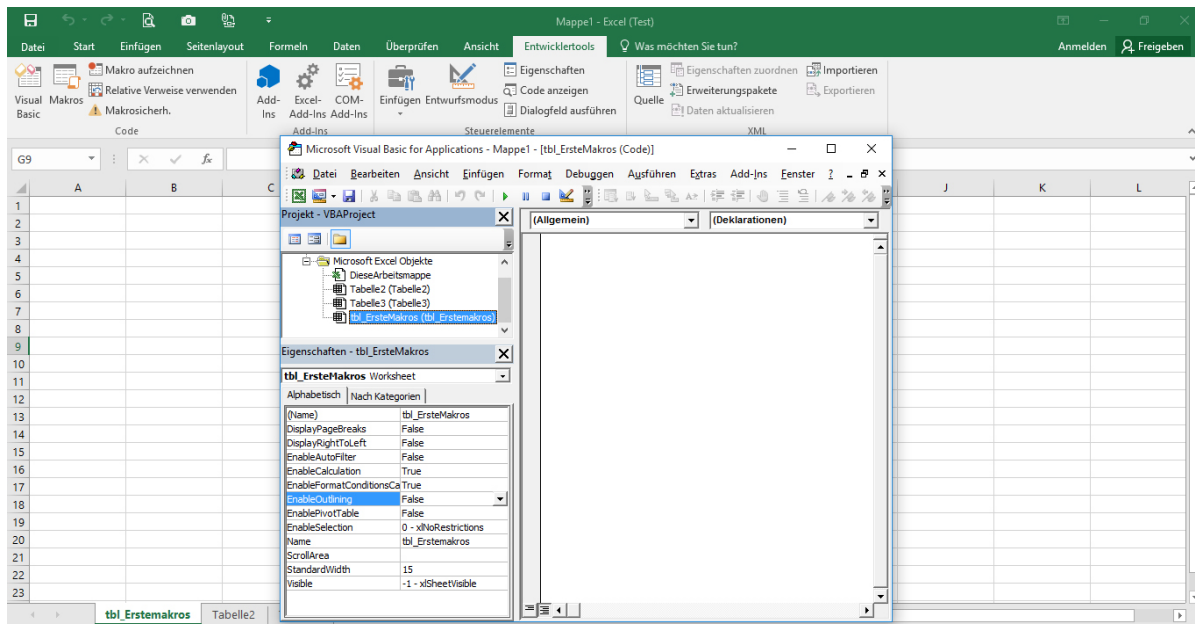


Abbildung 1.9: Sinnvolle Namen für Tabellen vergeben.



Bei der Benennung von Tabellen gilt es, bestimmte Einschränkungen zu beachten. So dürfen Sie beispielsweise bestimmte Zeichenfolgen beim Codenamen der Tabelle (die obere Eigenschaft im Eigenschaftsfenster) nicht verwenden. Unter anderem sind das Sonderzeichen sowie das Leerzeichen. Das erste Zeichen beim Codenamen der Tabelle darf auch keine Zahl sein.

Das Codefenster

Das Codefenster befindet sich standardmäßig rechts neben dem Projekt-Explorer. Dieses Fenster wird dann sichtbar, wenn Sie ein