

VBA mit WORD-97 lernen



Einstieg in die Welt der Makro-Programmierung

Werner Holler



Als Textverarbeitung – Anwender steht man oft vor demselben Problem: Man hat gelernt, Texte mit einem guten Programm zu erstellen, zu speichern, zu korrigieren oder auch bereits etwas schöner zu gestalten. Trotzdem gibt es immer wiederkehrende Routineaufgaben, und für diese sich wiederholenden und teilweise stumpfsinnigen Arbeiten wurde die Makrosprache Visual Basic for Applications (kurz VBA) geschaffen. Mit deren Hilfe gelingt es, Arbeitsschritte zu vereinfachen oder Prozesse zu verkürzen und damit zu beschleunigen.

Das Buch von Rene Martin: *VBA mit WORD97 lernen* - Einstieg in die Welt der Makro – Programmierung ist dafür ein nettes Einstiegswerk, das sich in erster Linie an alle richtet, die keine oder nur geringe Vorkenntnisse in VBA haben. Allerdings finden sicher auch Profis in diesem Buch interessante Anregungen für die Programmierpraxis.

Der Autor demonstriert anhand gut nachvollziehbarer Eigenschaften, wie ausgewählte Routineaufgaben oder Probleme mit dem Programm *Word 97* mittels Makros gelöst werden können. Dabei geht es vor allem um praxisorientierte Anwendungen, die der Autor mittels fünf fiktiver Figuren (Kunstgeschichtestudent, Soziologe, Biologiestudent, Romanistikstudent, Geschäftsführerin eines EDV – Schu-

lungsunternehmens) bestimmte fachspezifische Probleme in VBA lösen läßt. So wird aus einem ansonsten relativ trockenen Thema ein lebendiges, anschauliches und lehrreiches Buch, das auch ohne Computer sehr gut lesbar ist (selbst probiert auf schattigen Plätzen im sonnigen Süden ohne PC – es sollte schließlich ja auch ein echter Urlaub sein!).

Das Buch bietet auch viele nützliche und im Buch beschriebene Makros zum Lernen und Gebrauchen auf beiliegender Diskette an.

Im ersten Teil des Buches wird gezeigt, daß in vielen Bereichen allerdings Makros gar nicht nötig sind. Das wird anhand einiger ausgewählter Beispiele aus der Praxis gezeigt. Dabei geht es nicht nur um Feldfunktionen und Serienbriefe, sondern auch um Einführung in die Denkweise des Programmierens. Es wird gezeigt, wie alltagsprachlich formulierte Probleme mit Hilfsmitteln von *Word* gelöst werden können. Es ist nämlich auch die Umgebung wichtig: Der Programmierer sollte nicht nur wissen, wie er sein Makro erstellen muß, sondern sollte auch ein wenig Ahnung von der Umgebung von *Word* haben. Etwa, wie ein Makro in der richtigen Dokumentvorlage gespeichert wird oder ihm ein Symbol, eine Tastenkombination oder ein Menüeintrag zugewiesen wird.

Im zweiten Teil des Buches wird der Makro-Rekorder erläutert, mit dem man sich wiederholende Arbeitsschritte einfach aufzeichnen und gegebenenfalls später noch händisch nachbearbeiten kann. Dieser Makro-Rekorder ist vor allem für den Einsteiger in die VBA-Programmierung ein sehr nützliches Hilfsmittel.

Das Einrichten von Masken mit den dazugehörigen Programmierschritten wird im dritten Teil des Buches vorgestellt.

Ausblicke auf fortgeschrittene Programmierung, die über die alleinige Textverarbeitung hinausgehen, werden im letzten Teil des Buches gegeben. Ziel ist es, Programme zu entwickeln, die Texte erstellen und weitere Programme aus dem Office-Paket benutzen.

Ein Paar nette Ausschnitte aus dem Buch gefällig? Bitte sehr:

Auswahlbeispiel 1

Sehr oft hat man viele Fenster gleichzeitig geöffnet. Im Menüteil *Fenster* sind aber nur bis zu 9 Fenster sichtbar. Das folgende Makro zeigt alle zur Zeit geöffneten Fenster an:



Sub Fensterzählen()

```
Dim Zähler As Integer
Dim Fenstername As String
Zähler = 0

For Each Fenster In Windows
    Fenstername = Fenster.Caption & Chr(13) & Fenstername
    Zähler = Zähler + 1
Next Fenster

MsgBox prompt:=Fenstername,
Title:="Folgende " & Str$(Zähler) & " Fenster sind geöffnet"
End Sub
```

Auswahlbeispiel 2

Ein in manchen Fällen sicherlich sehr nützliches Makro ist untenstehendes: Es berechnet nach Eingabe des Geburtsdatums die noch verbleibenden Tage bis zum Geburtstag.

Dieses Makro ist beliebig ausbaufähig, man kann etwa wichtige Namen und deren Geburtsdaten als Konstante eingeben und auf diese Art einen Geburtstagskalender erstellen, der dann – möglicherweise sogar in ein automatisches Startfenster eingebaut (siehe unten) - vergeblichen Mitmenschen so manche unangenehme Situation ersparen hilft.



Sub Geburtstag()

```
Dim Datum As Date

GebDatum = InputBox("Wie lautet Dein
Geburtsdatum?")

Tag% = Day(GebDatum)
Monat% = Month(GebDatum)

If Tag% = Day(Date) And Monat% = Month(Date)
Then
    MsgBox "Happy Birthday"
Else
    AnzTage = DateDiff("D", Date,
DateSerial(Year(Date) + 1, Monat%, Tag%))

    If AnzTage = 365 Then AnzTage = AnzTage -
365
    ' OK. Für Schaltjahre sollte hier etwas
geändert werden ...
    MsgBox "Sorry, Du hast erst in " &
AnzTage & " Tagen Geburtstag"
End If

End Sub
```

Auswahlbeispiel 3

Wer E-Mails schreibt, wird schon festgestellt haben, dass aufgrund der verschiedenen Online-Dienste besser keine Umlaute wie "ä", und "ö", aber auch keine "ß" im Text stehen sollten. (Zwischenfrage: Gibt's nach der Rechtschreibreform eigentlich noch ein "ß", oder gibt's überhaupt noch die Rechtschreibreform?). Wer seine Texte in Word schreiben und sie nur noch in seine Mails kopieren will, dem wird folgendes Makro sicherlich nützlich sein:

```
Option Base 1
' Bewirkt, dass bei Arrays bei der
Indizierung mit 1 statt mit 0 begonnen wird
Sub Umlautedrehen()
Dim Umlaut(4)
Dim BuchstabKombin(4)
Dim i%

Umlaut(1) = "ä"
Umlaut(2) = "ö"
Umlaut(3) = "ü"
Umlaut(4) = "ß"
BuchstabKombin(1) = "ae"
BuchstabKombin(2) = "oe"
BuchstabKombin(3) = "ue"
BuchstabKombin(4) = "ss"

For i% = 1 To 4
    Selection.Find.ClearFormatting

    Selection.Find.Replacement.ClearFormatting
    With Selection.Find
        .Text = Umlaut(i%)
        .Replacement.Text =
BuchstabKombin(i%)
    End With
    Selection.Find.Execute
Replace:=wdReplaceAll
Next i%
End Sub
```

Automatisches Starten von Makros

Als Abschluß soll hier noch eine kurze überarbeitete Abhandlung über einen sehr interessanten Abschnitt aus dem Buch zum Thema "Automatisches Starten von Makros" kurz vorgestellt werden:

Üblicherweise werden Makros nur über direkte Aktivierung durch den Anwender, also etwa durch Anklicken oder durch eine Tastenkombination, aktiviert. Makros können aber auch direkt beim Start von Word, beim Start eines bestimmten Dokuments oder auch beim Start einer bestimmten Dokumentvorlage beginnen.

Eine automatische Prozedur wird dann aufgerufen, wenn sie einen bestimmten Namen hat und sich im Ordner *Microsoft Word Objekte / ThisDokument* befindet. Die Makros müssen unter folgenden Namen gespeichert werden:

<i>AutoExec</i>	Dieses Makro wird beim Start von Word ausgeführt
<i>AutoNew</i>	Wird beim Erstellen eines neuen Makros ausgeführt.
<i>AutoOpen</i>	Wird gestartet, wenn der Benutzer ein vorhandenes Dokument öffnet
<i>AutoClose</i>	Wird beim Schließen eines Dokumentes aktiviert
<i>AutoExit</i>	Startet beim Beenden von Word. Gut geeignet für Speicherungen und Sicherheitsaktionen

Obiges Makro etwa startet Word mit einem freundlichen Begrüßungsbildschirm, schaltet das Lineal ein und stellt die Ansicht auf Normal sowie Zoom auf 91%.

Weitere im Buch beschriebene Beispiele für automatisch startende Makros:

- Zuletzt bearbeitetes Dokument wird nach Aufruf von Word gestartet, die Stelle im Text wird angesprungen, von der aus man das Dokument geschlossen hat.
- Jeder Benutzer wird individuell mit dem gleichen Makro begrüßt, indem das Makro in der Registry nachschaut, welcher Benutzer eingetragen ist.
- Zum Abfangen von Fehlern: Beginnt ein Dateiname mit DOKUMENT (etwa DOKUMENT1, wie es bei Erst-Erstellen von neuen Texten üblich ist), so muss das Dokument gespeichert werden, und zwar unter einem anderen Namen. Wer hat nicht schon auf die Frage "Wollen Sie die Änderungen in Dokument1 speichern?" auf "Nein" geklickt, und weg war der wichtige Text?
- Makros, die beispielsweise 1 Stunde nach dem Word-Start oder zu einem bestimmten einzustellenden Zeitpunkt starten.

Rene Martin; VBA mit WORD97 lernen; Einstieg in die Welt der Makro – Programmierung; Addison – Wesley;



ISBN: 3 8273 1198 5; ATS 291,00

