

1

Der Einstieg in die Makroprogrammierung



Geschätzte Dauer:
60 Minuten

In dieser Lektion lernen Sie

- wie Sie ein Makro mit dem Makrorecorder aufzeichnen.
 - wie Sie mit dem Visual Basic-Editor Ihre Makros bearbeiten.
 - wie Sie durch die Hilfefunktion schnelle Unterstützung erhalten.
-

Anwender von Microsoft Word beginnen meist dann, sich für die Makroprogrammierung zu interessieren, wenn sie wiederkehrende Bearbeitungsschritte automatisieren möchten. Manch einen nervt es einfach, wenn er wiederkehrende Befehle immer wieder von vorn eingeben muss. Andere Anwender gehen noch einen Schritt weiter. Sie überlegen sich, wie sie ihre Arbeitsprozesse insgesamt durch Automatisierung effektiver organisieren können. Auch Anwendungsentwickler beginnen zunehmend, sich für die mächtige Programmiersprache Visual Basic for Applications (VBA) zu interessieren, die in Word verwendet wird.

Die VBA-Programmierung hat mit der Makroprogrammierung der frühen Tage kaum noch etwas gemeinsam. Was damals ein einfaches Mittel zur Aufzeichnung von Tastenfolgen war, hat sich zur objektorientierten Sprache gemausert. Der Anfänger kann nach wie vor die gewünschten Tastenfolgen aufzeichnen. Der Anwendungsentwickler kann aber darüber hinaus komplette Anwendungen aus den Bausteinen erstellen, die Word und die übrigen Office-Programme zur Verfügung stellen.

Anfänger, Power-User und Anwendungsentwickler finden in diesem Buch gleichermaßen nützliche Informationen. Der Anfänger wird Schritt für Schritt zur Programmierung hingeführt. Der Power-User erhält das nötige Handwerkszeug auf seinem Weg zum Experten. Der Anwendungsentwickler schließlich bekommt Ideen für Projekte und Hinweise zur Realisierung.

Aber genug der Vorrede. Warum Sie sich für dieses Buch interessieren, das wissen Sie selbst am besten. Springen wir also mitten hinein in die Praxis.

Jedes Buch über die Programmierung von Word beginnt wahrscheinlich mit dem gleichen Thema: Ein Makro soll aufgezeichnet, wiedergegeben und bearbeitet werden. Auch dieses Buch macht da keine Ausnahme.

Ein Makro aufzeichnen

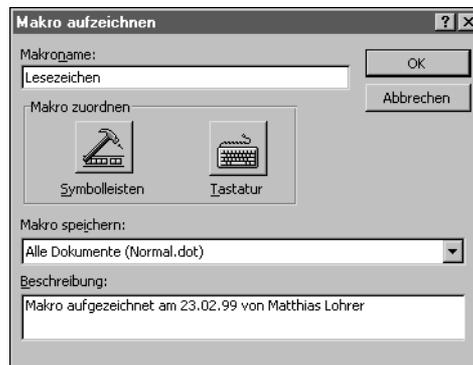
Angenommen, Sie erstellen zurzeit ein etwas umfangreicheres Dokument, beispielsweise einen Bericht, ein Referat oder eine Broschüre. Während Sie ihren Text überarbeiten, fallen ihnen gelegentlich Passagen auf, bei denen Sie sich nicht sicher sind, ob die Angaben stimmen. Wenn Sie die Stellen nicht sofort korrigieren können, erstellen Sie hier so etwas wie ein Lesezeichen, indem Sie beispielsweise *!Überprüfen!* in Ihren Text hineinschreiben. Wenn Sie sich Ihr Dokument später erneut vornehmen, können Sie gezielt nach diesen Stellen suchen und sie bearbeiten.

Für das Einfügen Ihres Lesezeichens wünschen Sie sich ein Makro. Auf einen Tastendruck hin soll der Text *!Überprüfen!* in das Dokument eingefügt werden. Wenn Sie die Eingabe nicht jedes Mal manuell vornehmen, hat das außerdem den Vorteil, dass Sie sich nicht vertippen können. Mit einem Suchen-Befehl können Sie die entsprechenden Passagen zuverlässiger finden, als wenn Sie mögliche Tippfehler berücksichtigen müssten.

Dieses Makro werden Sie nun mit wenigen Handgriffen erstellen.

- ❶ Öffnen Sie in Word 2000 ein neues, leeres Dokument.
- ❷ Wählen Sie im Menü *Extras* die Option *Makro* und dann die Option *Aufzeichnen*. Es erscheint das Dialogfeld *Makro aufzeichnen*.
- ❸ Geben Sie im bereits aktivierten Feld *Makroname* einen Namen Ihrer Wahl ein, z.B. **Lesezeichen** (siehe Abbildung 1.1).

Abbildung 1.1
Mit dem Makrorecorder lassen sich Makros bequem aufzeichnen





- 4 Klicken Sie auf *OK*. Das Dialogfeld verschwindet und Sie haben wieder die gewohnte Word-Oberfläche vor sich. Außerdem sehen Sie eine kleine Symbolleiste über dem Dokument schweben. Ab dem jetzigen Zeitpunkt ist die Makroaufzeichnung aktiv. Alle Befehle, die Sie jetzt eingeben, werden in Ihrem Makro *Lesezeichen* gespeichert. Mit den beiden Schaltflächen auf der Symbolleiste können Sie die Makroaufzeichnung vorübergehend anhalten oder komplett beenden.
- 5 Geben Sie nun den Text ein, der jeweils als Lesezeichen in Ihre Dokumente eingefügt werden soll. Tippen Sie **!Überprüfen!** in das momentan geöffnete Dokument.
- 6 Beenden Sie die Makroaufzeichnung, indem Sie auf das kleine Quadrat in der Makrosymbolleiste klicken.

Fertig. Damit ist Ihr Makro erstellt.

Während der Makroaufzeichnung ist die Mausbedienung eingeschränkt. Sie können mit Hilfe der Maus zwar Menüs selektieren, jedoch beispielsweise keinen Text markieren.



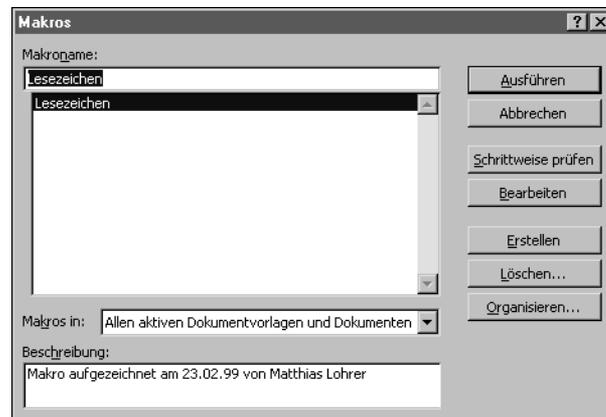
Ein Makro über das Menü ausführen

Nun wollen Sie testen, ob Ihr Makro auch tatsächlich funktioniert.

- 1 Wählen Sie aus dem Menü *Extras* die Option *Makro* und dann *Makros*. Auch die Tastenkombination **(Alt) + (F8)** ruft dieses Fenster auf.
- 2 Das Dialogfeld *Makros* öffnet sich (siehe Abbildung 1.2). Hier sehen Sie eine Liste aller Makros, die Sie bislang erstellt haben. Selektieren Sie das Makro *Lesezeichen* und klicken Sie auf *Ausführen*. Das Dialogfeld wird geschlossen, und der Text des Lesezeichens wird in das Dokument eingefügt.

Abbildung 1.2

Das Dialogfeld *Makros* ist die Schaltzentrale für die Makroverwaltung in Word



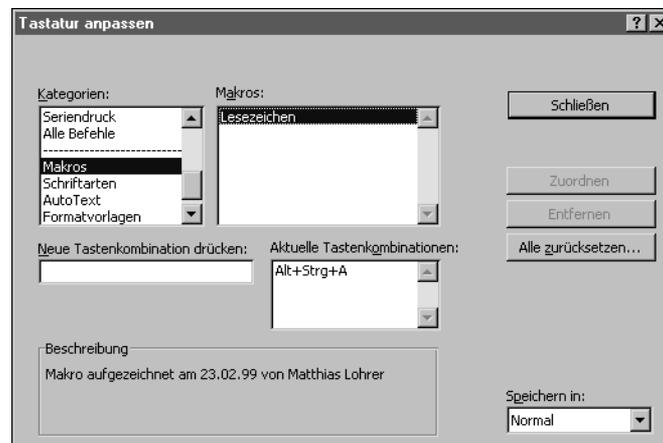
Einem Makro eine Tastenkombination zuweisen

Na schön, werden Sie jetzt sagen. Geht es vielleicht noch umständlicher? Ich wollte doch eine Tastenkombination haben. Stattdessen muss ich jetzt x-mal mit der Maus irgendwo hinklicken. Gemach, gemacht! Das war nur der erste Schritt. Zunächst einmal mussten Sie das Makro erstellen. In einem zweiten Schritt werden Sie Ihrem Makro nun noch eine Tastenkombination zuweisen.

- 1 Wählen Sie aus dem Menü *Extras* die Option *Anpassen*. Es erscheint das Dialogfeld *Anpassen*.
- 2 Klicken Sie unten auf die Schaltfläche *Tastatur*. Zusätzlich wird das Dialogfeld *Tastatur anpassen* geöffnet (siehe Abbildung 1.3).
- 3 Wählen Sie im Listenfeld *Kategorien* die Option *Makros*. Diese Option steht relativ weit unten, noch hinter einer abtrennenden Linie.
- 4 Im Listenfeld *Makros* erscheinen die Makros. Selektieren Sie hier Ihr Makro *Lesezeichen*.
- 5 Aktivieren Sie nun das Feld *Neue Tastenkombination drücken*.
- 6 Geben Sie eine von Ihnen gewünschte Tastenkombination ein. Versuchen Sie es für den Anfang mit **Alt+Gr** + **A**.
- 7 Unter dem Feld erscheint der Hinweis: *Derzeit zugeordnet zu [nicht zugewiesen]*. Damit wissen Sie, dass dieses Kürzel zurzeit keinem anderen Befehl oder Makro zugeordnet ist.
- 8 Klicken Sie auf die Schaltfläche *Zuordnen*. Ihre Tastenkombination wird in das Listenfeld *Aktuelle Tastenkombinationen* übernommen (siehe Abbildung 1.3).

Abbildung 1.3

Ihren Makros können Sie beliebige Tastenkombinationen zuordnen



- 9 Damit ist alles erledigt. Schließen Sie das Fenster *Tastatur anpassen*, indem Sie auf *Schließen* klicken.
- 10 Schließen Sie auch das Dialogfeld *Anpassen*.

Zuordnen nicht vergessen!

Anfänger übersehen oft, dass es nicht ausreicht, die Tastenkombination einzugeben und anschließend auf die Schaltfläche *Schließen* zu klicken. Sie müssen auf jeden Fall erst *Zuordnen* aufrufen! Als Resultat wird die Tastenkombination dann im Feld *Aktuelle Tastenkombinationen* angezeigt. Falls Sie es sich doch noch anders überlegen sollten, hätten Sie jetzt auch Gelegenheit, das Kürzel wieder zu löschen, indem Sie auf die Schaltfläche *Entfernen* klicken. Anschließend könnten Sie ein neues Kürzel eingeben und mit *Zuordnen* speichern.



Ein Makro über seine Tastenkombination aufrufen

Damit haben Sie Ihrem Makro *Lesezeichen* die Tastenkombination **AltGr**+**A** zugewiesen. Probieren Sie es aus!

- Stellen Sie die Schreibmarke an eine beliebige Stelle in Ihrem Dokument und drücken Sie **AltGr**+**A**.

Automatisch wird Ihr Lesezeichentext eingefügt. Na, ging das jetzt schnell genug? War doch gar nicht schwer.

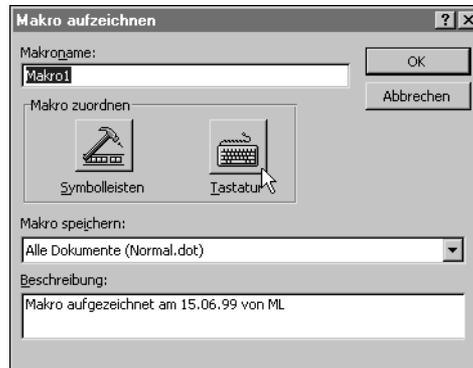
Die Tastenkombination schon beim Aufzeichnen zuweisen

Und nun verrate ich Ihnen noch einen Trick! Es ist auch möglich, in einem einzigen Arbeitsschritt ein Makro aufzuzeichnen und die Tastenkombination zuzuweisen. Wenn Sie die Makroaufzeichnung starten, sehen Sie unter dem Feld *Makroname* den Bereich *Makro zuordnen* (siehe Abbildung 1.4). Hier gibt es zwei Schaltflächen. Eine führt zur Anpassung der Symbolleisten. Eine zweite zur Anpassung der Tastatur. Wenn Sie hier auf *Tastatur* klicken, erscheint das Dialogfeld *Tastatur anpassen*, das Sie bereits kennen gelernt haben. Hier können Sie Ihre Tastenkombination definieren. Nachdem Sie eine Tastenkombination zugewiesen und das Fenster geschlossen haben, befinden Sie sich im Makro-Aufzeichnungsmodus. Geben Sie hier die Befehle ein und beenden Sie die Makroaufzeichnung über die Makrosymbolleiste, so wie Sie es bereits gelernt haben.

Sie haben vorhin zunächst das Makro aufgezeichnet und erst nachträglich die Tastenkombination zugeordnet. Was Ihnen vielleicht umständlich vorkommt, hat zwei Vorteile:

Abbildung 1.4

Mit den beiden Schaltflächen können Sie einem Makro bereits beim Aufzeichnen eine Symbolschaltfläche oder ein Tastenkürzel zuweisen



- Sie haben gelernt, dass Makro und Tastenkombination voneinander unabhängig sind, und beliebig einander zugeordnet werden können.
- Zwei kleine Schritte nacheinander sind für den Anfang leichter als ein großer Schritt auf einmal.

Gute und weniger gute Tastenkombinationen

Beim Zuweisen von Tastenkombinationen sollten Sie vorsichtig sein, da Sie hier leicht in die eine oder andere Falle tappen können. Wenn Sie beispielsweise statt $\text{AltGr} + \text{A}$ die Tastenkombination $\text{Alt} + \text{A}$ verwenden, erhalten Sie die Rückmeldung, dass auch dieses Kürzel zurzeit nicht verwendet wird. Das stimmt aber nicht so ganz, denn mit $\text{Alt} + \text{A}$ lässt sich über die Tastatur das Menü *Ansicht* öffnen. Nachdem Sie dieses Kürzel Ihrem Makro zugeordnet haben, können Sie das Menü *Ansicht* nicht mehr über die Tastatur öffnen. Stattdessen wird bei der Eingabe von $\text{Alt} + \text{A}$ das Makro ausgeführt.

Ähnlich tückisch verhalten sich einige Kombinationen mit der Taste AltGr . Über AltGr erreichen Sie neben der Klein- und Großschreibung sozusagen die dritte Ebene auf der Tastatur. $\text{AltGr} + \text{Q}$ fügt beispielsweise das @-Zeichen ein, das stets bei E-Mail-Adressen verwendet wird. Wenn Sie dieses Kürzel in dem Feld *Neue Tastenkombination drücken* eingeben, erhalten Sie ebenfalls die irreführende Meldung, dass dieses Kürzel derzeit nicht zugewiesen sei. Wenn Sie jedoch einem Makro dieses Kürzel zuweisen, erhalten Sie genau den gegenteiligen Effekt, wie bei einigen Kombinationen mit der Alt -Taste: Beim Aufruf des Kürzels wird nach wie vor das @-Zeichen eingegeben, das Makro wird aber nicht ausgeführt.

Viele Leute überschreiben einfach die vordefinierte Tastenkombination, die sie selbst nicht verwenden. $\text{Strg} + \text{Z}$ führt beispielsweise standardmäßig den Befehl *Zeilenabstand2* aus. „Na gut,“ sagen sich manche Programmierer, „dieses Kürzel verwende ich ohnehin nicht, also über-





schreibe ich es einfach.“ Solange Sie nur selbst an Ihrem PC arbeiten, ist diese Vorgehensweise völlig unproblematisch. Aber schon wenn Ihr Freund oder Kollege ausnahmsweise Ihren PC benutzt, kann dieses Verfahren zu Problemen führen. Wenn der Kollege gewohnt ist, Word genau mit diesen vordefinierten Kürzeln zu bedienen, wird er völlig irritiert sein, wenn sich Word nicht wie erwartet verhält.

Bevor Sie jetzt verzweifeln und ihre Finger gleich ganz von den Tastenkombinationen lassen, sei Ihnen folgender einfache Rat gegeben: Probieren Sie das Kürzel Ihrer Wahl aus, noch ehe Sie den Makrorecorder einschalten, bzw. bevor Sie über die Option *Menü/Anpassen* ein Kürzel zuweisen. Wenn das gewünschte Kürzel auch im Normalzustand bereits irgendeine Aktivität hervorruft, dann sollten Sie lieber eine andere Tastenkombination verwenden.

Die dritte Ebene, die über **AltGr** erreichbar ist, ist übrigens nur in Ausnahmefällen belegt. Sehen Sie einfach auf Ihre Tastatur. Dort sind die jeweiligen Zeichen in der Regel mit angegeben. Hier werden Sie noch auf die wenigsten Probleme stoßen. Die Hinweise in der Dialogbox sind für die Wahl des Kürzels jedenfalls unzureichend.

Das Makro muss einen Namen haben

Bei der Wahl eines Namens für ein Makro sind Sie nicht so frei wie bei der Wahl eines Namens für ein Dokument. So darf im Makronamen beispielsweise kein Leerzeichen vorkommen. Außerdem sind einige Sonderzeichen wie beispielsweise das %-Zeichen nicht erlaubt. Die Verwendung von Umlauten, dem „ß“ und dem Unterstrich „_“ ist aber gestattet.

Schutz vor Makroviren

Viren sind Programme, die Ihrem Computer in irgendeiner Form Schaden zufügen. Wenn der Virencode in einer Makrosprache wie VBA erstellt wurde, spricht man von Makroviren. Leider können auch Word-Dokumente potenziell Makroviren enthalten.

Unter dem Menübefehl *Extras/Makro/Sicherheit* können Sie aus drei verschiedenen Sicherheitsstufen auswählen, wie sich Word verhalten soll, wenn ein Dokument geöffnet wird, das irgendwelche Makros (nicht unbedingt Viren) enthält (siehe Abbildung 1.5). Bei der Option *Niedrig* erhalten Sie keinen Warnhinweis beim Öffnen von Dokumenten, die Makros enthalten. Bei *Mittel* können Sie jeweils auswählen, ob die Makros aktiviert oder deaktiviert werden sollen (siehe Abbildung 1.6). Bei *Hoch* werden nur signierte Makros aus vertrauenswürdigen Quellen ausgeführt.

Solange Sie mit Ihren selbst geschriebenen Makros experimentieren, können Sie die niedrige Sicherheitsstufe wählen, da Sie sonst stets erst die Makroaktivierung bestätigen müssen. Ansonsten ist es für Ihre Sicherheit

Abbildung 1.5
Das Dialogfeld
Sicherheit

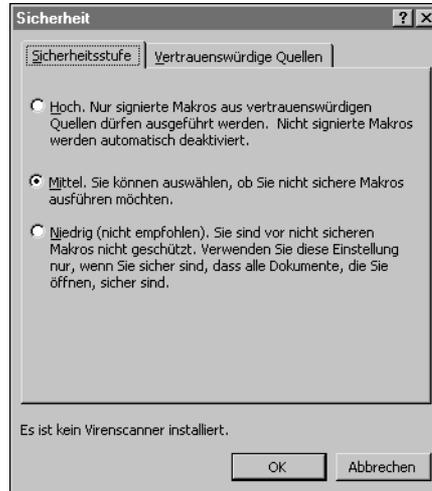
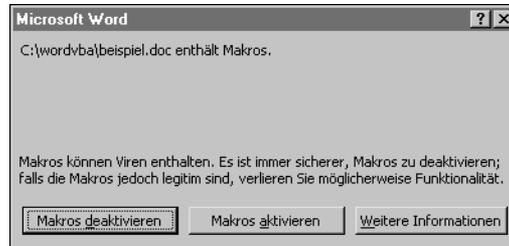


Abbildung 1.6
Bei der Wahl der
mittleren Sicher-
heitsstufe müssen
Sie vor dem Öffnen
von Dokumenten,
die Makros enthal-
ten, stets erst die
Aktivierung der
Makros bestätigen



vorteilhaft, zumindest die mittlere Stufe zu wählen. Im Übrigen empfiehlt Microsoft, auf jeden Fall einen aktuellen Virens Scanner zu installieren.

In der zehnten Lektion werden Sie sich noch eingehender mit dem Thema „Makroviren und Sicherheit“ auseinandersetzen. Dort werden Sie auch einen Weg kennen lernen, wie Sie auch bei der Verwendung der höchsten Sicherheitsstufe Ihre eigenen Makros stets ohne Rückfrage ablaufen lassen können.

Makros im Visual Basic-Editor bearbeiten

Angenommen, Sie haben sich bei der Eingabe Ihres Makros vertippt. Was nun? Müssen Sie Ihr Makro jetzt völlig neu erstellen? Nein, Sie können Ihr Makro nachträglich bearbeiten. Dafür und für viele weitere Aufgaben stellt Word den Visual Basic-Editor zur Verfügung.

Von Word zum Visual Basic-Editor ...

Es gibt mehrere Möglichkeiten, von Word aus den Visual Basic-Editor zu öffnen. Für den ersten Einstieg ist der folgende Weg am einfachsten:

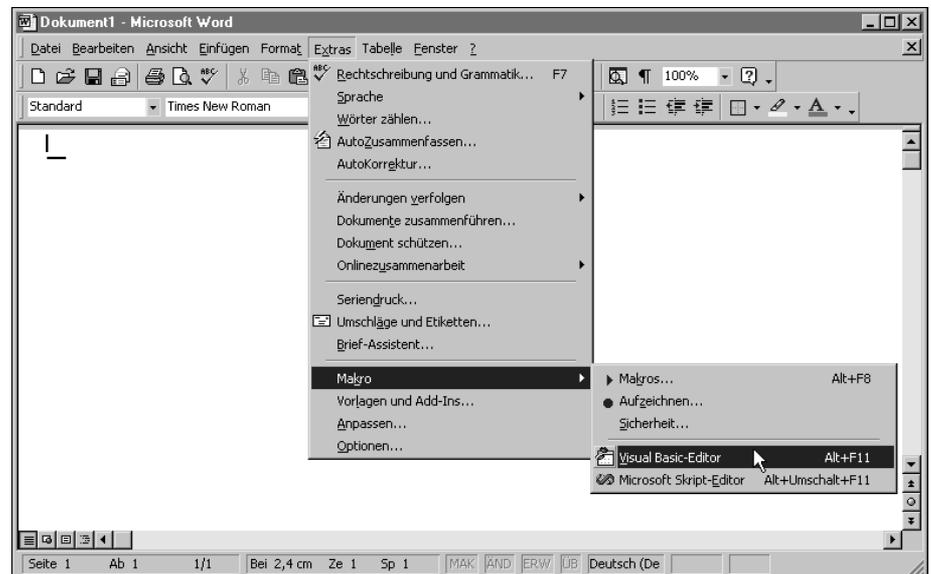
- 1 Wählen Sie aus dem Menü *Extras* die Option *Makro* und danach die Option *Makros*.
- 2 Das Dialogfeld *Makros* öffnet sich. Markieren Sie das Makro, das sie bearbeiten wollen und klicken Sie auf *Bearbeiten*. Der Visual Basic-Editor wird geöffnet.

Wenn Sie direkt zum Visual Basic-Editor gelangen wollen, ohne gleich das passende Makro auf dem Bildschirm zu haben, können Sie auch den Menübefehl *Extras/Makro/Visual Basic-Editor* auswählen (siehe Abbildung 1.7).

Diesem Menübefehl ist außerdem die Tastenkombination **Alt + F11** zugeordnet.

Abbildung 1.7

Sie können den Visual Basic-Editor auch direkt aufrufen



... und wieder zurück zu Word

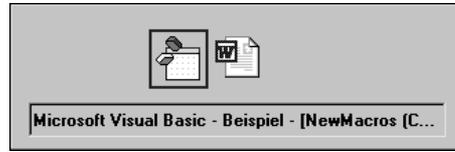
Klicken Sie im Visual Basic-Editor auf der Symbolleiste links oben auf die Schaltfläche mit dem *W* für Word. Anschließend sind Sie wieder in Ihrer gewohnten Word-Umgebung.

Da der Visual Basic-Editor kein Teil von Word, sondern ein eigenständiges Programm ist, können Sie ihn auch über die Programmfensterleiste erreichen.



Abbildung 1.8

Auch mit **[Alt]+[F8]** können Sie zwischen Word und dem Visual Basic-Editor hin- und herwechseln



- ❶ Drücken Sie **[Alt]** und halten Sie diese Taste gedrückt. Tippen Sie dann einmal kurz auf **[F8]**. Es erscheint ein Fenster mit den Symbolen der im Moment geöffneten Programme (siehe Abbildung 1.8).
- ❷ Indem Sie wiederholt bei gedrücktem **[Alt]** auf **[F8]** drücken, wird jeweils das folgende Symbol selektiert.
- ❸ Lassen Sie **[Alt]** los, wenn das gewünschte Symbol markiert ist. Das gewünschte Programm, Word, der Visual Basic-Editor oder ein anderes Programm wird aktiviert. In Abbildung 1.8 ist das Symbol für den Visual Basic-Editor markiert.

Und schließlich führt Sie auch die Tastenkombination **[Alt]+[F11]** wieder zum Word-Bildschirm zurück.

- ❶ Drücken Sie in Word **[Alt]+[F8]**. Das Dialogfeld *Makros* öffnet sich.
- ❷ Markieren Sie Ihr Makro *Lesezeichen* und klicken Sie auf *Bearbeiten*. Der Visual Basic-Editor wird geöffnet.
- ❸ Wechseln Sie mit den beschriebenen Methoden einige Male zwischen Word und dem Visual Basic-Editor hin und her. Lassen Sie schließlich den Visual Basic-Editor geöffnet.

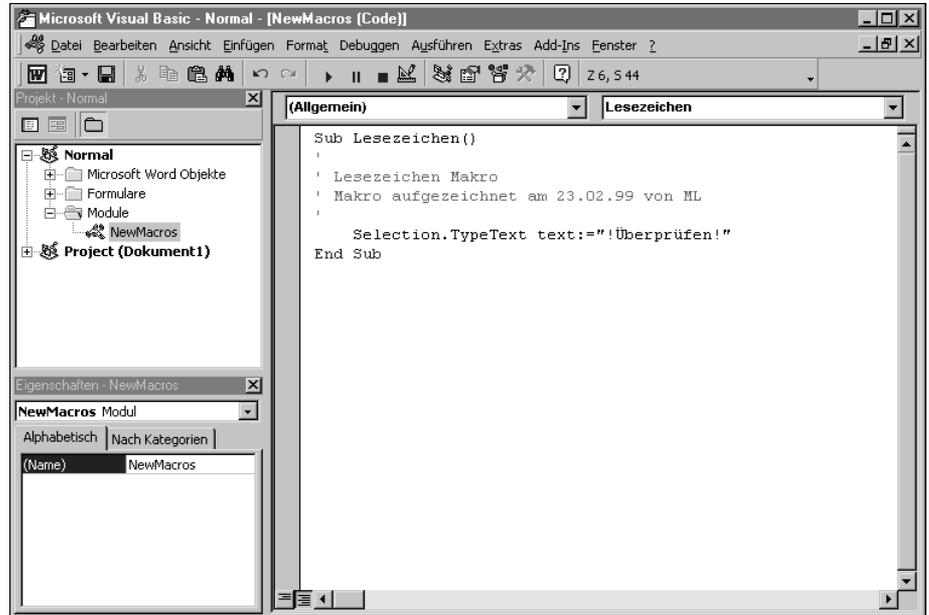
Sich im Visual Basic-Editor zurechtfinden

Der erste Eindruck des VB-Editors auf den Neueinsteiger ist immer derselbe. „Das sind hier aber viele Fenster mit vielen Schaltflächen und Symbolen. Wie soll ich mich denn da zurechtfinden?“ Lassen Sie sich nicht entmutigen! Für den ersten Einstieg müssen Sie nur einen kleinen Teil der Optionen beherrschen, die der Editor bietet. Diese werden Sie jetzt kennen lernen. In den kommenden beiden Lektionen werden Sie sich dann Schritt für Schritt auch in die weiterführenden Möglichkeiten einarbeiten.

Drei Fenster: Projekte, Eigenschaften und Code

Beim Öffnen des Visual Basic-Editors werden standardmäßig drei Fenster eingeblendet (siehe Abbildung 1.9). Dominierend ist das große Codefenster. Auf der linken Seite sehen Sie oben das Projektfenster und darunter das Eigenschaftfenster. Außerdem sehen Sie über die gesamte Fensterbreite eine Menüzeile und eine Symbolleiste. Als Einsteiger müssen Sie das Projektfenster und das Eigenschaftfenster erst einmal überhaupt nicht interessieren. Wenn Sie möchten, können Sie diese beiden

Abbildung 1.9
Der Visual Basic-
Editor



Fenster auch erst einmal schließen, dann hat das Codefenster mehr Platz und Sie werden nicht unnötig abgelenkt. Später können Sie diese Fenster jederzeit wieder über *Ansicht/Projektexplorer* und *Ansicht/Eigenschaftenfenster* öffnen.

Makro oder Prozedur?

Vorab ein kleiner Hinweis: Der Visual Basic-Editor kennt den Ausdruck „Makro“ nicht. Was in Word „Makro“ heißt, wird im Visual Basic-Editor „Prozedur“ genannt. Lassen Sie sich dadurch nicht stören. Beides ist das gleiche. In Word heißt es „Makro“, weil es hier schon immer so geheißen hat. Der Visual Basic-Editor nennt es „Prozedur“, weil diese Bezeichnung in der Softwareentwicklung gebräuchlich ist. Da der Visual Basic-Editor erst nachträglich aus der Visual Basic-Programmierungsumgebung zu Word hinzugefügt wurde, ist es zu dieser kleinen Sprachverwirrung gekommen. Im Code selbst kommen außerdem die beiden Ausdrücke *Sub* und *Function* vor. *Sub* steht für „Subprozedur“ und ist das gleiche wie ein Makro. Eine Funktion ist eine Prozedur, die einen Wert zurückliefert. Später werden Sie hierfür Beispiele kennen lernen.

Das Codefenster

Wenn Sie in Word den Menübefehl *Extras/Makro/Makros* ausgewählt, ein Makro markiert und auf *Bearbeiten* geklickt haben, sehen Sie im Codefenster den Code genau dieses Makros. Wenn Sie gar kein Codefenster sehen sollten, können Sie es nachträglich über *Ansicht/Code* einblenden.



Ihr Lesezeichen-Makro besteht aus den folgenden Codezeilen:

```
Sub Lesezeichen()  
' Lesezeichen Makro  
' Makro aufgezeichnet am 23.02.99 von Matthias Lohrer  
,  
    Selection.TypeText Text:="!Überprüfen!"  
End Sub
```

Das Codefenster zeigt den Quelltext der Makros. Wenn Sie möchten, können Sie hier bereits kleine Änderungen im Quelltext vornehmen. Vielleicht gefällt es Ihnen nicht, dass Ihr Lesezeichen ohne Leerzeichen zu Beginn und am Ende eingefügt wird. Das können Sie hier problemlos ändern.

- ❶ Fügen Sie im Quelltext vor und nach dem Lesezeichen-Wort einige Leerzeichen ein. Achten Sie dabei darauf, dass die Leerzeichen innerhalb der beiden Anführungsstriche stehen.
- ❷ Wechseln Sie zurück zu Word.
- ❸ Führen Sie das Makro hier erneut aus. Nun wird Ihr Lesezeichen bereits in der geänderten Form ausgegeben.
- ❹ Wechseln Sie wieder zum Visual Basic-Editor.



Das Codefenster bietet zwei verschiedene Modi. Sie können entweder jeweils ein einzelnes Makro anzeigen, oder sich alle verfügbaren Makros hintereinander als Fließtext anzeigen lassen. Mit den beiden kleinen Schaltflächen links unten am Codefenster können Sie zwischen der Prozedur- und der Modulansicht hin- und herschalten.

Die Bestandteile einer Prozedur

Nun sollen Sie die Bestandteile dieses kurzen Quelltextes im Detail kennen lernen. Als Erstes fällt Ihnen vielleicht auf, dass der Code farbig dargestellt ist. Diese farbige Auszeichnung erledigt der Visual Basic-Editor automatisch. Textbestandteile werden je nach ihrer Bedeutung anders eingefärbt. Der Fachausdruck dafür lautet Syntax-Coloring.

Der Prozedurrumpf

Das Makro wird eingefasst von den Schlüsselworten *Sub* und *End Sub*. Hiermit erkennt Word, wo ein Makro beginnt und wo es aufhört. Dieses umklammernde Begriffspaar wird auch als Prozedurrumpf bezeichnet. Schlüsselworte sind Bezeichnungen, die in Visual Basic fest eingebaut sind. Visual Basic färbt Schlüsselworte blau ein.

Der Prozedurname

Hinter dem einleitenden *Sub* steht der Name des Makros, das Sie selbst gewählt haben. Diesen Namen können Sie, wenn Sie möchten, problemlos hier im Quelltext ändern. Auch wenn Sie später das Makrofenster beispielsweise mit **Alt+F8** öffnen, werden Sie den geänderten Namen sehen.

Hinter dem Namen sehen Sie ein leeres Paar Klammern, das Visual Basic automatisch hinzugefügt hat. Prozeduren oder Funktionen können Werte übergeben werden. Diese so genannten Argumente oder Parameter werden in diesem Klammernpaar definiert. Später werden Sie hierzu Beispiele kennen lernen. Da Sie für die Prozedur *Lesezeichen* keine Argumente definiert haben, ist das Klammernpaar leer.

Kommentare

Unter der Zeile *Sub Lesezeichen()* folgen einige Zeilen, die alle mit einem Apostroph beginnen. Alles, was hinter dem Apostroph steht, wird von Visual Basic als Kommentar behandelt. Das Programm versucht also nicht, diese Zeilen auszuführen. Hier können Sie Anmerkungen zur Aufgabe und zur Funktionsweise Ihres Programms hinzufügen. Bei größeren Makros sollten Sie von der Möglichkeit, Ihren Quellcode mit Kommentaren zu dokumentieren, regen Gebrauch machen. Wenn Sie selbst etwas im Kommentarblock eingeben, wird es automatisch grün eingefärbt. Sie können auch weitere Kommentarzeilen eingeben, müssen aber dabei darauf achten, dass die Zeile mit einem Apostroph beginnt.

Das Apostrophzeichen geben Sie mit **Alt+#** ein, rechts neben dem „ä“. Verwenden Sie nicht die Akzentzeichen!

Befehle

Normaler Code besteht aus einzelnen Zeilen. Er wird schwarz dargestellt. Falls einmal eine Zeile in rot dargestellt wird, bedeutet das, dass der Compiler hier einen Fehler gefunden hat, aber dazu kommen wir später. In Ihrem Makro steht hier eine einzelne Zeile mit dem Befehl:

```
Selection.TypeText Text: "!Überprüfen!"
```

Dieser Befehl lässt sich folgendermaßen in seine Bestandteile aufdröseln:

- *Selection* ist das zugrunde liegende Objekt.
- *TypeText* ist eine Methode des Objekts *Selection*. Methoden werden aufgerufen, indem man sie mit einem Punkt an das Objekt anhängt.
- Einem Argument mit Namen *Text* wird mit dem Zuweisungsoperator `:=` der Wert `"!Überprüfen!"` zugewiesen.



Jetzt wissen Sie zwar, wie dieser Befehl prinzipiell aufgebaut ist und wie man seine Bestandteile nennt. Vielleicht wirft das Beispiel bei Ihnen aber auch mehr Fragen auf, als es Antworten bietet:

- Was ist ein Objekt und was soll eine Methode sein?
- Woher weiß ich, welche Objekte es gibt, und wofür sie gut sind?
- Woher weiß ich, über welche Methoden diese Objekte verfügen?
- Woher weiß ich, wie die Argumente heißen und
- woher weiß ich, welche Werte eingesetzt werden können?

Nur keine Panik! Jetzt holen Sie erst einmal tief Luft!

Bestandteile eines Befehls

Objekte

Es ist nicht leicht zu sagen, was ein Objekt ist. Am richtigsten, aber auch am nichts sagendsten wäre die Antwort: Ein Objekt kann so ziemlich alles sein. Oder sagen wir einfach: Objekte sind die Bausteine, aus denen Word und die Word-Dokumente aufgebaut sind.

Es ist viel Arbeit, alle Objekte kennen zu lernen, über die Word verfügt. Hinzu kommt, dass es nicht einfach soundsoviele Objekte gibt, sondern diese Objekte sind hierarchisch zueinander aufgebaut. An der Spitze steht das *Application*-Objekt, das für die gesamte Word-Anwendung steht. Dieses Oberobjekt enthält nun eine Reihe Unterobjekte, die selbst wieder weitere Objekte enthalten und so weiter. Für den Anfang ist es aber weder sinnvoll noch nötig, gleich alle Objekte auf einmal kennen lernen zu wollen. In den kommenden Lektionen lernen Sie diejenigen Objekte kennen, die Sie für die tägliche Arbeit benötigen und die Ihnen helfen, produktiver zu arbeiten.

Methoden, Eigenschaften, Auflistungen

Typischerweise sind einem Objekt weitere Elemente zugeordnet. Objekte verfügen in der Regel über Eigenschaften, über Methoden, und sie enthalten weitere Objekte, meist so genannte Auflistungen gleichartiger Objekte. Eigenschaften kann man abfragen oder neu bestimmen. Methoden lassen sich ausführen. Ein Objekt aus einer Auflistung kann man bestimmen und mit ihm entsprechend verfahren.

Argumente, Parameter, Werte

Den Methoden werden oft eine oder auch mehrere Werte übergeben. Der Methode *TypeText* wird beispielsweise der Text, der eingefügt werden soll, als Wert übergeben. Wenn man sich nicht auf den jeweils konkreten

Wert bezieht, spricht man von den Parametern, die einer Methode zugeordnet sind.

Für den Anfang sind Sie nun gut beraten, einfach die Beispiele in dieser und in den kommenden Lektionen selbst nachzuvollziehen. Aus der großen Menge an Befehlen, Objekten, Methoden etc. werden vorrangig solche behandelt, die Sie bei der täglichen Arbeit oft einsetzen.

Eine Reise durch das Hilfesystem

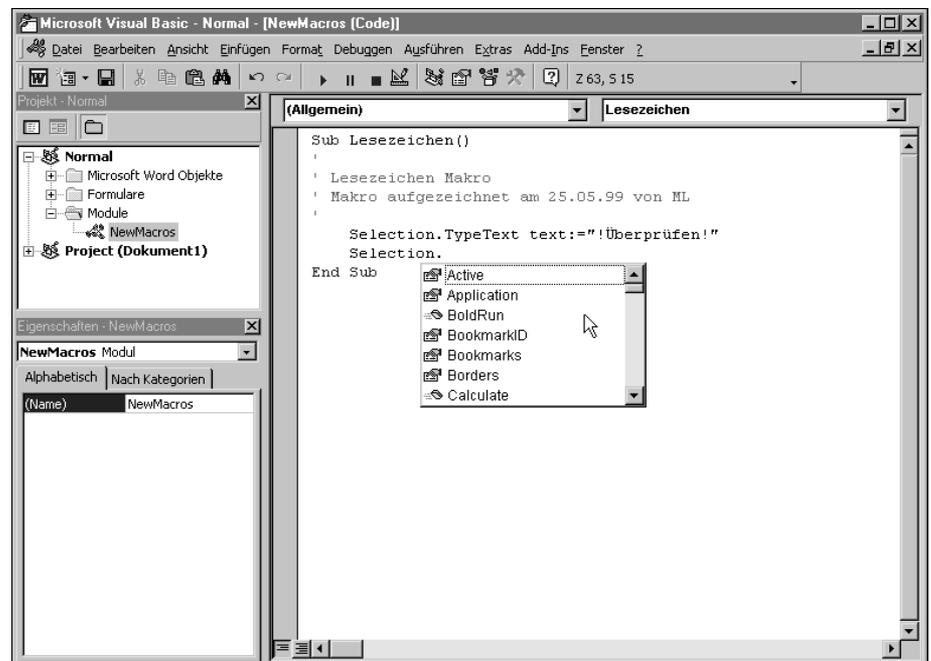
Nach Ihrer ersten Begegnung der dritten Art mit solch schwer fassbaren Erscheinungen wie Objekten, die so gut wie alles sein können und selbst wieder Auflistungen von Objekten in sich enthalten, ist es an der Zeit, Sie auf die Hilfe-Funktion des Visual Basic-Editors hinzuweisen. Die Entwicklerwerkzeuge von Microsoft genießen generell den Ruf, dass sie den Entwickler hervorragend bei der Arbeit unterstützen. Der Visual Basic-Editor macht da keine Ausnahme.

QuickInfos während der Codeeingabe

Um die praktischen QuickInfos kennen zu lernen, führen Sie folgende Schritte aus:

- 1 Setzen Sie im Codefenster im Makro *Lesezeichen* die Einfügemarke an den Anfang der Zeile *End Sub*. Drücken Sie einmal auf . Gehen Sie

Abbildung 1.10
QuickInfos unterstützen Sie bei der Codeeingabe



dann eine Zeile nach oben in die neu entstandene Leerzeile und drücken Sie einmal auf .

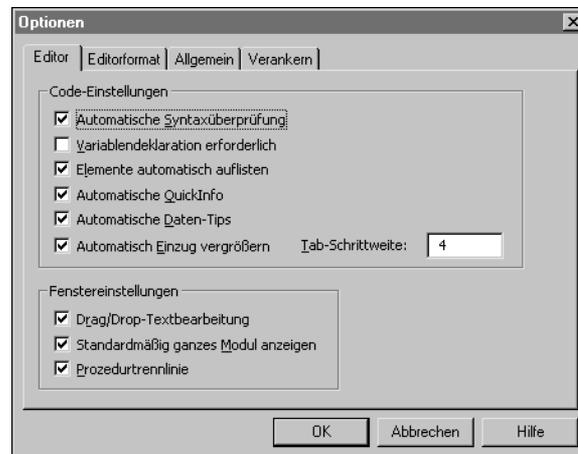
- 2 In die Leerzeile schreiben Sie **Selection.**, also **Selection** und einen Punkt dahinter.

Ein kleines Fenster öffnet sich. In diesem Fenster stehen alle Elemente, die zum Objekt *Selection* gehören, und das sind nicht gerade wenige (siehe Abbildung 1.10).

Falls bei Ihnen nichts erscheinen sollte, dann wählen Sie im Menü *Extras* den Befehl *Optionen ...* und selektieren im sich öffnenden Fenster das Kontrollkästchen *Elemente automatisch auflisten* (siehe Abbildung 1.11). Klicken Sie auf *OK*. Anschließend müsste bei dem entsprechenden Vorgehen das Fenster mit den Hinweisen auch bei Ihnen erscheinen.



Abbildung 1.11
Mit dem Menübefehl *Extras/Optionen* und der Option *Automatische QuickInfo* aktivieren Sie die Anzeige von QuickInfos



- 3 Mit  und  können Sie sich in dieser langen Auflistung bewegen. Sie können auch beginnen, einen Text einzugeben. Word springt dann automatisch an den Eintrag, der mit den Buchstaben beginnt, die sie eingegeben haben. Den jeweils selektierten Eintrag können Sie übernehmen, indem sie einfach auf  oder auf **Leer** drücken. Selektieren Sie nun auf diese Weise die Methode *MoveDown*.

- 4 Geben Sie hinter *MoveDown* ein Leerzeichen ein. Auch jetzt erscheint ein kleines Fensterchen, das Ihnen die Parameter anzeigt, die zu dieser Methode gehören. Geben Sie ein: **Unit:=wdLine**

- 5 Geben Sie ein Komma ein und ergänzen Sie: **Count:=5**. Drücken Sie dann .

```
Selection.MoveDown Unit:=wdLine, Count:=5
```

Die einzige Stelle, an der diese QuickInfo-Hilfe Sie leider im Stich lässt, ist der Moment, wenn Sie dem Argument *Unit* einen Wert zuordnen sollen. Woher sollten Sie wissen, um welche „Einheiten“ Sie sich hier fortbewegen können? Im weiteren Verlauf werden Sie auch hierzu einige Tricks kennen lernen. Fürs Erste nehmen Sie die Angabe *wdLine*. Bei *Count* handelt es sich offenkundig um die Anzahl Einheiten, um die nach unten gegangen werden soll. Hier kann eine beliebige ganze Zahl eingetragen werden.

Damit haben Sie Ihr Makro um einen weiteren Befehl ergänzt. Der neue Befehl soll nur der Demonstration dienen. Nachdem der Text eingefügt wurde, geht die Einfügemarke um fünf Zeilen nach unten. Versuchen Sie es.



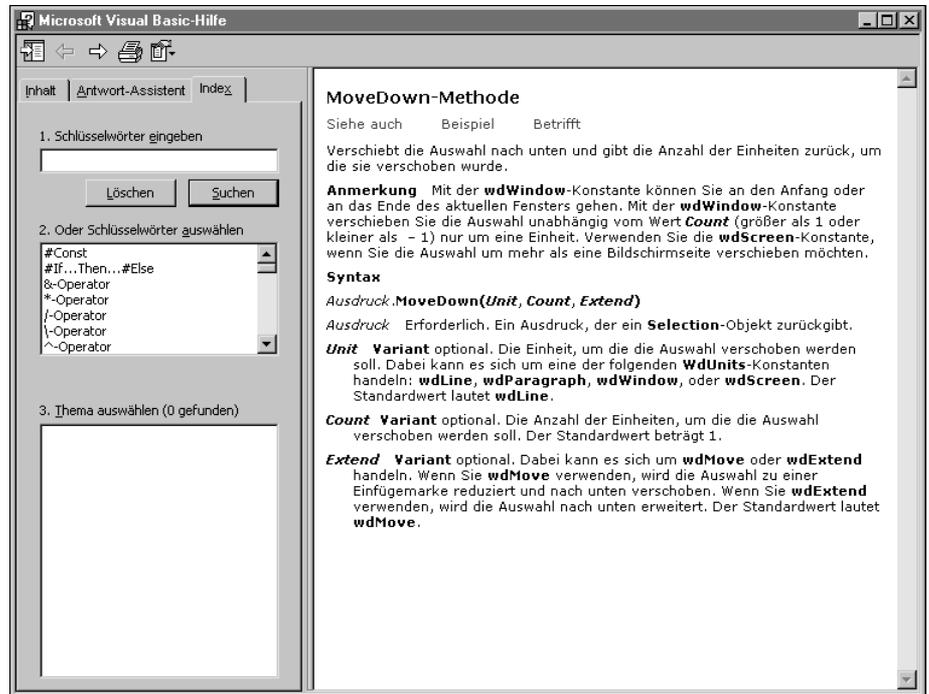
- 1 Wechseln Sie zum Word-Bildschirm.
- 2 Geben Sie einige Absatzzeichen ein, damit die Einfügemarke um fünf Zeilen nach unten verschoben werden kann, und stellen Sie die Einfügemarke an den Anfang des Textes.
- 3 Öffnen Sie das Dialogfeld *Makros*.
- 4 Lassen Sie das Makro *Lesezeichen* erneut ablaufen. Der Text wird eingefügt und die Einfügemarke wird um fünf Zeilen nach unten verschoben.
- 5 Wechseln Sie wieder zum Visual Basic-Editor.

Ausführliche Informationen bei der Codeeingabe

Die QuickInfos dienen der raschen Orientierung und erleichtern außerdem die Codeeingabe. Aber auch ausführlichere Informationen zu den einzelnen Befehlen sind leicht und schnell verfügbar.

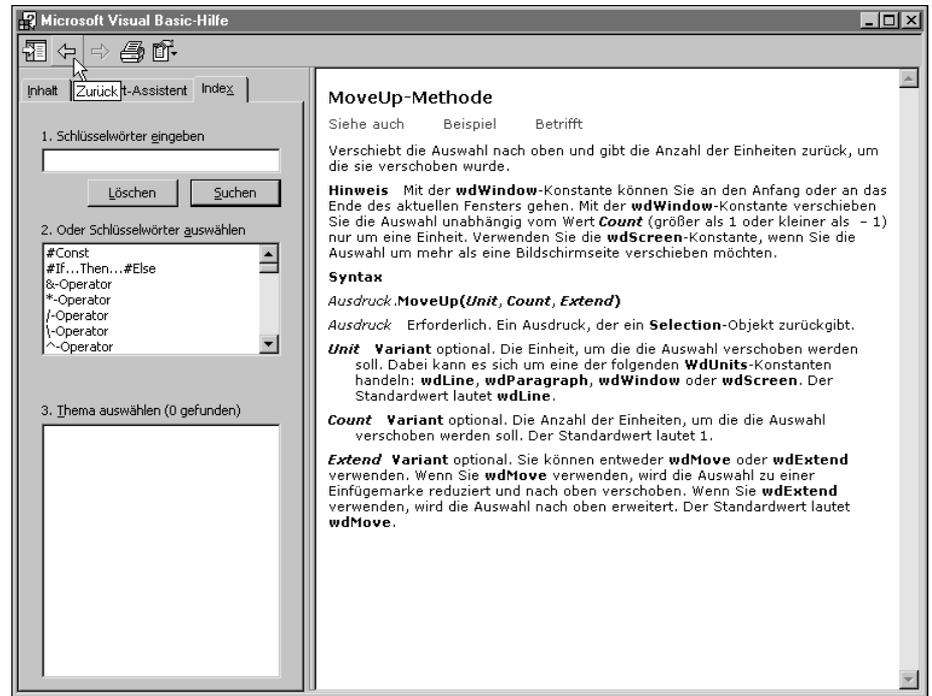
- 1 Bewegen Sie die Einfügemarke im Codefenster in das Wort *MoveDown*, das Sie gerade eingegeben haben, und drücken Sie **F1**.
Das Hilfefenster öffnet sich. Im rechten Teil sehen Sie Informationen zur *MoveDown*-Methode. Im linken Teil sehen Sie drei Registerkarten. Zunächst überblicken Sie den Hilfetext zur *MoveDown*-Methode (siehe Abbildung 1.12). Hier erfahren Sie zum Beispiel, welche Word-Konstanten Sie für den Parameter *Unit* verwenden können. Außer *wdLine* könnten Sie zum Beispiel auch *wdParagraph* verwenden. Die Bewegung erfolgt dann nicht zeilenweise, sondern absatzweise. Hier ist also bereits die Antwort auf die Frage, wo Sie die Werte finden, wenn ein Parameter als Wert eine Word-Konstante benötigt. Einfach in der Hilfe nachschlagen!
- 2 Unter der Titelleiste *MoveDown-Methode* sehen Sie drei Links zu *Siehe auch*, *Beispiel* und *Betrifft*. Klicken Sie auf *Siehe auch*: Ein Fenster *Themen gefunden* öffnet sich und listet zwei weitere Themen auf: die *Move*-Methode und die *MoveUp*-Methode.

Abbildung 1.12
Der Visual Basic-Editor bietet eine mächtige Hilfe-Funktion



- 3 Selektieren Sie die *MoveUp*-Methode, und klicken Sie auf *Anzeigen*. Im Anzeigefenster erscheint der Hilfetext zur *MoveUp*-Methode. Offenkundig funktioniert diese Methode sehr ähnlich.
- 4 Nun wollen Sie zurück zur vorhergehenden Anzeige. Klicken Sie in der Symbolleiste auf die Schaltfläche *Zurück* mit dem nach links zeigenden Pfeil (siehe Abbildung 1.13). Die beiden Pfeilschaltflächen können Sie dazu verwenden, sich auf dem Weg, den Sie bislang in der Hilfe zurückgelegt haben, vor- und zurückzubewegen.
- 5 Klicken Sie auf *Beispiel*. Im Fenster werden einige komplette Codestücke vorgestellt und erläutert. Der erläuternde Text auf der Eingangsseite kann oft kompliziert und unübersichtlich anmuten. Beim Studium der Beispiele wird vieles schneller klar. Oft können Sie sich aus dem Beispielpcode auch Anregungen für die eigene Arbeit holen.
- 6 Klicken Sie in der Symbolleiste auf die Schaltfläche *Zurück*, um zur Hauptseite von *MoveDown* zu gelangen.
- 7 Klicken Sie abschließend auf *Betrifft*. Auch hier erscheint das Dialogfeld *Themen gefunden*, das jetzt aber nur eine einzige Fundstelle auflistet: das *Selection*-Objekt.
- 8 Klicken Sie auf *Anzeigen*.

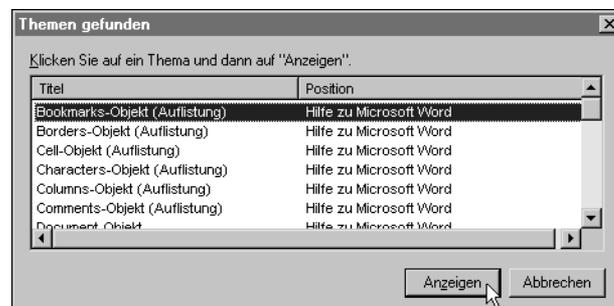
Abbildung 1.13
Mit Hilfe der Schaltflächen können Sie in der Hilfe wie in einem Web-Browser navigieren



Es erscheinen ausführliche Erläuterungen zu diesem mächtigen Objekt. Unter anderem finden Sie hier Verweise zu den *Eigenschaften* und den *Methoden*, die dieses Objekt zur Verfügung stellt.

- 9 Klicken Sie auf *Methoden*, markieren Sie in der dann erscheinenden langen Liste die *TypeText*-Methode, und klicken Sie auf *Anzeigen*. Ein kurzer Hinweis zu dieser einfachen Methode erscheint.
- 10 Gehen Sie zurück zum *Selection*-Objekt.
- 11 Klicken Sie auf den Kasten *Multiple Objects* unter dem Kasten *Selection*. Nun erscheint eine Auflistung mit allen Objekten, die dieses Objekt selbst

Abbildung 1.14
Dem *Selection*-Objekt sind zahlreiche Elemente zugeordnet



enthält (siehe Abbildung 1.14). Die meisten Objekte sind Auflistungen, z.B. *Characters*, *Columns*, *Footnotes* usw.

- 12 Wählen sie *Abbrechen*, da Sie hierfür zurzeit keine weiteren Informationen benötigen.
- 13 Schließen Sie die Hilfe.

Hilfe über die Registerkarten anfordern

Der Ausgangspunkt Ihrer vorigen Reise durch das Hilfesystem war die Taste **F1**. Es gibt weitere Zugänge zur Hilfefunktion.

- 1 Wählen Sie im Visual Basic-Editor aus dem Menü ? die Option *Microsoft Visual Basic-Hilfe*.

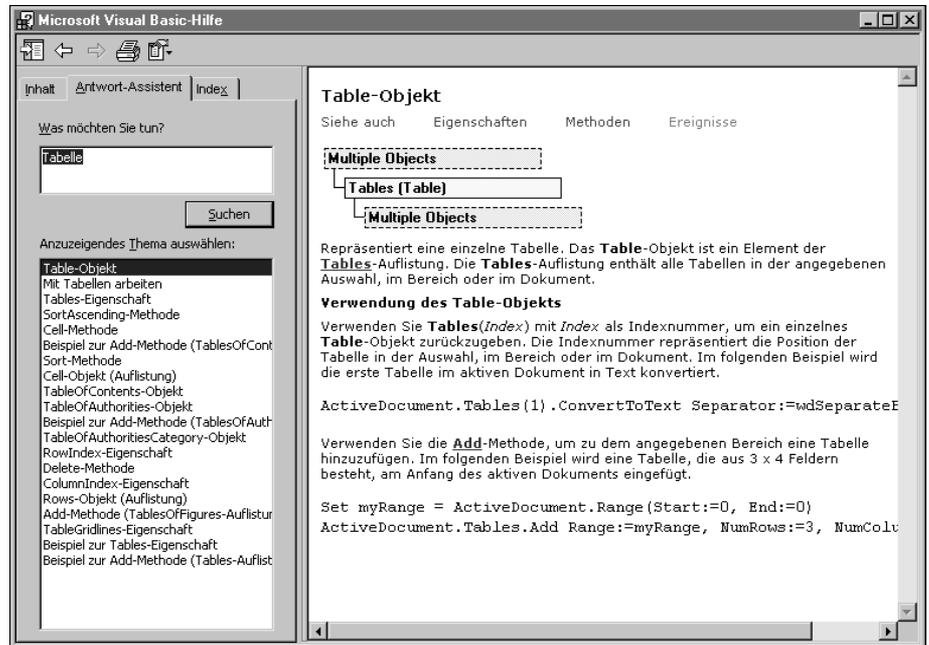
Das bereits bekannte Hilfefenster öffnet sich und zeigt einen allgemeinen Text zu Visual Basic an. Hier können Sie nun über die Registerkarten gezielt Informationen anfordern. Über die Registerkarte *Index* haben Sie beispielsweise gezielt Zugriff auf bestimmte Schlüsselwörter. Entweder geben Sie das gesuchte Schlüsselwort im Feld 1. *Schlüsselwörter ein* oder Sie selektieren es aus dem zweiten Listenfeld 2. *Oder Schlüsselwörter auswählen*. Wenn Sie dann auf *Suchen* klicken, werden im Feld 3. *Thema auswählen* entsprechende Fundstellen aufgelistet. Wenn Sie eines der Themen markieren, werden die entsprechenden Informationen angezeigt.

- 2 Klicken Sie auf die Registerkarte *Index*.
- 3 Klicken Sie auf einen beliebigen Eintrag im Listenfeld 2. *Oder Schlüsselwörter auswählen*, und klicken Sie auf *Suchen*. Im Listenfeld 3. *Thema auswählen* erscheinen anschließend die zugehörigen Themen.
- 4 Klicken Sie auf eines der Themen, um den Inhalt anzuzeigen.

Wenn Sie Informationen zu einem bestimmten Themengebiet suchen, von dem Sie nicht wissen, wie Visual Basic damit umgeht, können Sie die Registerkarte mit dem *Antwort-Assistent* verwenden. Im Eingabefeld *Was möchten Sie tun?* können Sie einen oder mehrere Begriffe eingeben, die mit Ihrem Thema zusammenhängen. Wenn Sie beispielsweise Tabellen mit Makros bearbeiten möchten, dann können Sie Informationen hierzu folgendermaßen erhalten.

- 5 Klicken Sie auf die Registerkarte *Antwort-Assistent*.
- 6 Geben Sie im Feld *Was möchten Sie tun?* nur das Wort **Tabelle** ein.
- 7 Klicken Sie auf *Suchen*. Im Listenfeld *Anzuzeigendes Thema auswählen* erscheinen zahlreiche Einträge (siehe Abbildung 1.15).
- 8 Klicken Sie auf ein Thema, um es anzuzeigen.

Abbildung 1.15
Der Antwort-Assistent erleichtert die Suche nach Themengebieten



Die Registerkarte *Inhalt* ermöglicht Ihnen schließlich einen streng systematischen Zugang zur Online-Hilfe. Die Einträge sind hier ähnlich strukturiert wie die Verzeichnisse und die Dateien im Windows-Explorer.

- 9 Klicken Sie auf die Registerkarte *Inhalt*. Sie sehen eine Liste von Einträgen mit führenden +-Zeichen.
- 10 Klicken Sie auf eines der +-Zeichen. Der Eintrag wird dadurch erweitert und zeigt die enthaltenen Themen an.
- 11 Markieren Sie eines der Themen. Der Eintrag wird angezeigt.
- 12 Schließen Sie das Hilfefenster.

Gesammelte Reise-Impressionen

Drei Eindrücke sollten Sie bei dieser kurzen Reise durch das Hilfesystem gewinnen.

- 1 Es ist meist sehr leicht, zu einem Begriff oder zu einem umfassenderen Thema gezielt Hilfe zu finden.
- 2 Der Aufbau des Hilfesystems ermöglicht es außerdem sehr einfach, von einem Begriff zum nächsten zu finden, der mit dem ersten zusammenhängt. Zu den Objekten finden Sie also sehr schnell die Methoden, hierzu die benötigten Parameter usw.

- 3 Es ist letztlich gleichgültig, über welchen Zugang Sie die Hilfe öffnen, ob beispielsweise gezielt zu einem Wort im Codefenster über (F1), oder über die diversen Registerkarten. Im Endeffekt werden Sie stets zu den gleichen Informationen geführt. Sie „verpassen“ also nichts, wenn Sie die eine oder andere Methode bevorzugen.

Die Menüs des Visual Basic-Editors – ein Überblick

Sicher werden Sie schon einmal die verschiedenen Menüs des Visual Basic-Editors angeklickt haben, um zu sehen, was sich dahinter so alles verbirgt. In einem ersten Schritt sollen Sie zunächst eine ungefähre Vorstellung davon erhalten, welche Optionen die zahlreichen Menüs bieten.

Datei

Die Optionen *Speichern*, *Beenden* und *Drucken* sind Ihnen bereits aus anderen Programmen vertraut. Außerdem können Sie hier Codeteile aus anderen Dateien importieren oder Ihre Makros exportieren.

Bearbeiten

Auch in diesem Menü finden Sie vieles, was Ihnen bereits bekannt vorkommen dürfte. Dazu zählen die Optionen *Einfügen*, *Löschen*, *Suchen*, *Ersetzen* und *Rückgängig*. Mit *Einzug vergrößern* können Sie nachträglich die Einrückungstiefe der jeweils markierten Codebestandteile anpassen. Außerdem können Sie Lesezeichen setzen. In größeren Projekten hilft es Ihnen, sich in Ihrem Code zurechtzufinden. Schließlich können Sie hier gezielt Informationen zu Methoden, Konstanten und den Parametern anzeigen lassen.

Ansicht

In diesem Menü können Sie in erster Linie die zahlreichen Fenster, die der Visual Basic-Editor zur Verfügung stellt, öffnen oder aktivieren. Außerdem können Sie hier die Anzeige der Symbolleisten steuern.

Einfügen

Über dieses Menü können Sie neue Programmbestandteile einfügen. Dazu gehören Prozeduren, Formulare und Module. Außerdem können Sie VBA-Code aus externen Dateien einfügen.

Format

Die Befehle in diesem Menü beziehen sich alle auf die Gestaltung von Formularen für den Anwenderdialog. Die Gestaltung von Formularen werden Sie in einer kommenden Lektion lernen.

Debuggen

Dies ist ein sehr wichtiges Menü. Mit diesen Optionen können Sie Ihre Makros kompilieren, schrittweise ausführen, Haltepunkte setzen und mehr. Das sind alles Dinge, die Sie gut brauchen können, wenn Sie einmal gelernt haben werden, sozusagen „freihändig“ Makros zu schreiben.

Ausführen

Die Optionen in diesem Menü erlauben Ihnen, ein Makro auszuführen, zu unterbrechen und zu beenden. Außerdem können Sie hier in den Bearbeitungsmodus bei Formularen wechseln.

Extras

Dieses Menü fasst sehr verschiedene Befehle zusammen. Mit den *Verweisen* und den *Zusätzlichen Steuerelementen* können Sie Verweise auf externe Programmbibliotheken organisieren. *Makro* öffnet das aus Word vertraute *Makros*-Fenster. *Optionen* erlaubt Ihnen, generelle Einstellungen zum Visual Basic-Editor vorzunehmen. Mit *Eigenschaften von Normal* bestimmen Sie allgemeine Einstellungen zur gerade bearbeiteten Vorlage, beziehungsweise zum bearbeiteten Projekt. Schließlich können Sie *Digitale Signaturen zuordnen* und verwalten.

Add-Ins

Hier können Sie den Add-In-Manager öffnen.

Fenster

Die Optionen in diesem Menü beziehen sich ausschließlich auf das bzw. die geöffneten Codefenster. Sie können wählen, ob diese Fenster sich überlagern, nur teilweise überlappen oder nebeneinander angezeigt werden sollen. Schließlich können Sie ein Fenster auch teilen und auf diese Weise an zwei verschiedenen Stellen gleichzeitig in einem Modul Änderungen vornehmen. Alle geöffneten Codefenster werden am Schluss aufgelistet und können hier gezielt aktiviert werden.

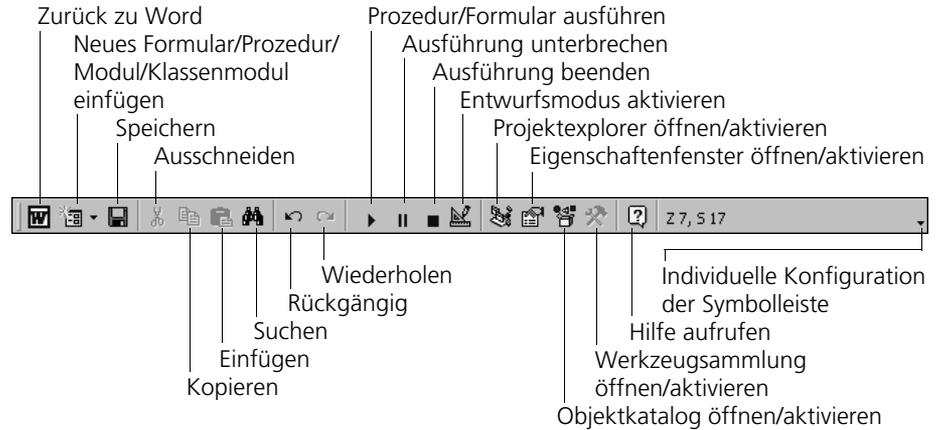
?

Mit diesem Menü können Sie die Hilfe öffnen und Informationen zum Editor erhalten. Mit *MSDN im Web* öffnen Sie den Internet Explorer und rufen die Microsoft Standardseite für Entwickler auf.

Die Schaltflächen der Symbolleiste

Schließlich sollen sie noch die Bedeutung der Schaltflächen auf der Symbolleiste kennen lernen. Bei etlichen werden Sie die Bedeutung bereits kennen oder sie werden die Bedeutung erraten können. Hier eine kurze Übersicht:

Abbildung 1.16
Symbolleiste des
Visual Basic-Editors



Zusammenfassung der Lektion

Möchten Sie

ein Makro aufzeichnen,

einem Makro eine Tastenkombination zuordnen oder die Tastenkombination löschen oder ändern,

ein Makro ausführen,

dann

wählen Sie im Menü *Extras* den Befehl *Makro* und dann die Option *Aufzeichnen*. Geben Sie hier einen Namen ein, und klicken Sie auf *OK*. Zum Beenden der Aufzeichnung klicken Sie auf die Schaltfläche *Aufzeichnung beenden* auf der Makro-Symbolschaltfläche oder wählen den Menübefehl *Extras/Makro/Aufzeichnung beenden*.

wählen Sie im Menü *Extras* den Befehl *Anpassen*. Klicken Sie hier auf die Schaltfläche *Tastatur*. Selektieren Sie im sich öffnenden Dialogfeld *Tastatur Anpassen* die Kategorie *Makros*. Selektieren Sie Ihr Makro. Geben Sie im Eingabefeld *Neue Tastenkombination drücken* die gewünschte Tastenkombination ein. Vergessen Sie nicht, jetzt erst auf *Zuordnen* zu klicken, ehe Sie dann auf *Schließen* klicken. Schließen Sie das Dialogfeld.

wählen Sie im Menü *Extras* den Befehl *Makro* und dann *Makros*. Das Dialogfeld *Makros* öffnet sich. Markieren Sie das gewünschte Makro. ►

Möchten Sie	dann
ein Makro bearbeiten,	Klicken Sie auf <i>Ausführen</i> . Wenn sie dem Makro eine Tastenkombination zugeordnet haben, können Sie alternativ auch diese Tastenkombination eingeben, während Sie Word geöffnet haben.
den Namen eines Makros ändern,	wählen Sie im Menü <i>Extras</i> den Befehl <i>Makro</i> und dann <i>Makros</i> . Das Dialogfeld <i>Makros</i> öffnet sich. Alternativ führt auch die Tastenkombination [Alt]+[F8] zu diesem Fenster. Selektieren Sie das gewünschte Makro, und klicken Sie auf <i>Bearbeiten</i> .
den Visual Basic-Editor aufrufen,	rufen Sie den Visual Basic-Editor auf. Ändern Sie im Codefenster den Namen des Makros, das hinter dem Wort <i>Sub</i> steht. Achten Sie darauf, dass Sie nicht die Klammern hinter dem Namen des Makros entfernen.
	wählen Sie im Menü <i>Extras</i> den Befehl <i>Makro</i> und dann <i>Visual Basic Editor öffnen</i> . Alternativ können sie auch die Tastenkombination [Alt]+[F11] verwenden.
 Suchen Sie Online-Hilfe über	wählen Sie im Visual Basic-Editor den Menübefehl <i>?/Microsoft Visual Basic-Hilfe</i>, klicken Sie auf die Registerkarte <i>Antwort-Assistent</i>, und suchen Sie nach diesem Eintrag:
<i>Selection</i> -Objekt,	<i>Selection-Objekt</i>
 Suchen Sie Online-Hilfe über	selektieren Sie im Visual Basic-Editor einen Menübefehl (ohne zu klicken) und drücken Sie [F1].
Verwendung des Visual Basic-Editors,	Es erscheint eine allgemeine Hilfe zum Thema <i>Visual Basic-Menüs</i> . Unter <i>Siehe auch</i> können Sie Einträge zu den verschiedenen Menüs auswählen.



Suchen Sie Online-Hilfe über

dann wählen Sie in Word den Menübefehl *?/Word-Hilfe*, klicken Sie auf die Registerkarte *Antwort-Assistent* und suchen Sie nach diesem Eintrag:

Makroaufzeichnung,
Tastenkombinationen,

Aufzeichnen eines Makros in Word
Anpassen von Belegungen mit
Tastenkombinationen

Vorschau auf die nächste Lektion

In dieser Lektion haben Sie viel Neues kennen gelernt. Makroaufzeichnung, Befehle, Objekte, den Visual-Basic-Editor und, und, und. Vielleicht ging es Ihnen aber zu langsam. Denn jetzt sind Sie zwar klüger und wissen vieles, von dem Sie vorher gar nicht dachten, dass Sie es zur Makroprogrammierung kennen müssen, aber so richtig programmieren können Sie noch nicht.

Daher verspreche ich Ihnen: Ab der nächsten Lektion wird richtig programmiert! Während Ihr Makro abläuft, wird der Anwender in einem Dialogfeld, das Sie selbst erstellt haben, Eingaben vornehmen. Die Eingaben des Anwenders werden Sie in Variablen speichern, Sie werden Berechnungen anstellen und die neuen Werte dem Anwender präsentieren. Für diese Aufgaben werden Sie Funktionen nutzen und die richtigen Operatoren einsetzen. Sie sehen, es erwartet Sie einiges! Also legen Sie bis zur nächsten Lektion jetzt lieber erst einmal eine kurze Pause ein.