

Öffnen Sie EXCEL und wechseln Sie mit der Tastenkombination Alt + F11 in den *Visual Basic-Editor*. Alternativ können Sie auch das Tabellenregister ('Tabelle1') mit der rechten Maustaste anklicken und mit 'Code anzeigen' den Visual Basic-Editor öffnen. Klicken Sie anschließend im Menü *Einfügen* auf den Eintrag *UserForm*.

In Ihrem Codefenster wird ein Formular in der Entwurfsansicht, gem. nachfolgendem Muster angezeigt und die Symbolleiste *Werkzeugsammlung* eingeblendet.

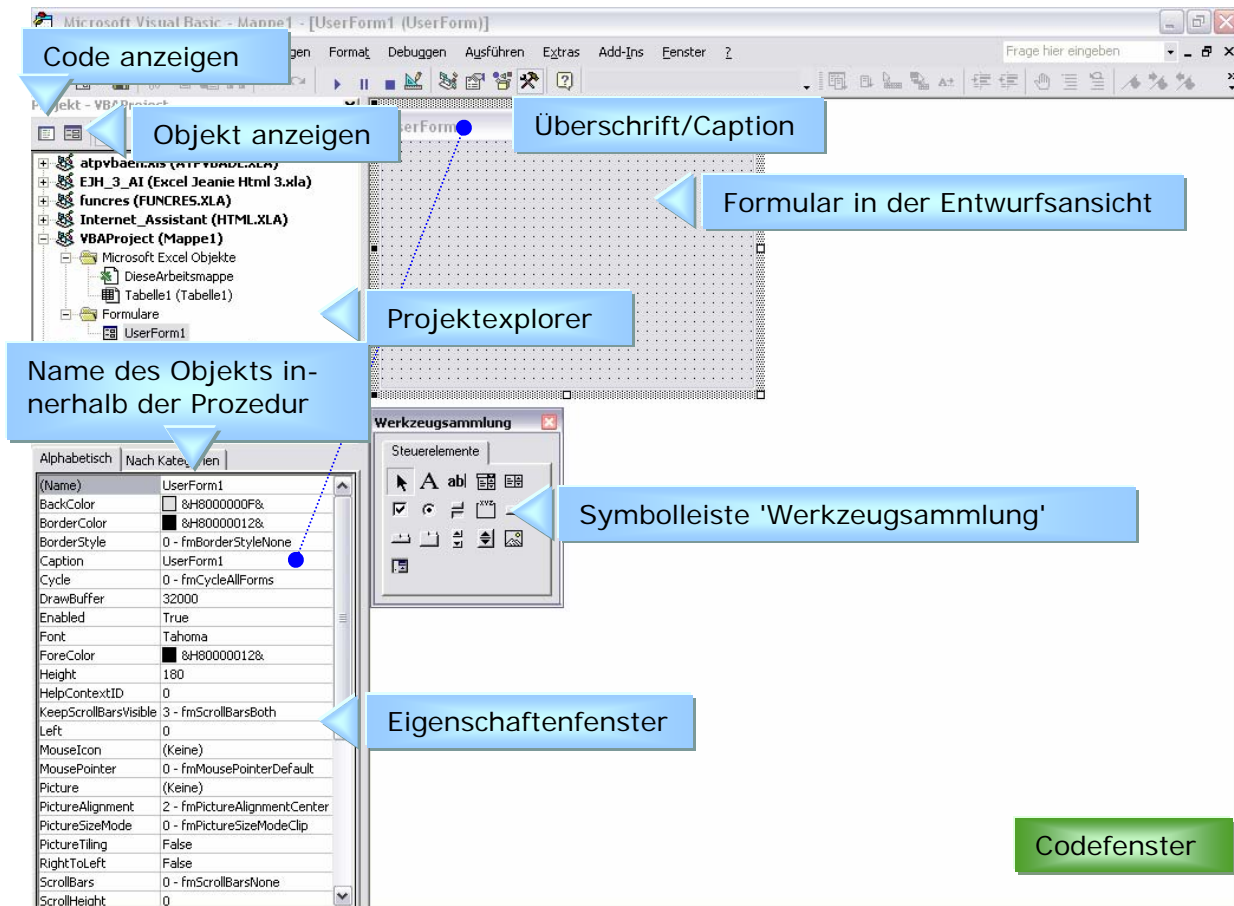


Abb. 1

Um der UserForm einzelne Objekte, wie z.B. Schaltflächen, Textfelder etc. hinzuzufügen, klicken Sie auf das jeweilige Steuerelement und ziehen es per Drag & Drop auf die Fläche der UserForm. Weisen Sie allen Objekten im Eigenschaftfenster eindeutige Namen zu und verwenden Sie dabei die gängigen Typkürzel, wie z.B. txtName für ein Textfeld 'Name', lblName für ein Bezeichnungsfeld 'Name' oder frmPersonal für ein Formular mit dem Namen 'Personal'. Über diese Namen werden die einzelnen Steuerelemente innerhalb Ihrer Prozedur identifiziert und können so auch im Code direkt angesprochen werden. Auch andere Eigenschaften, wie z.B. die Größe des Objekts, dessen Positionierung im Formular, Hintergrund- oder Schriftfarbe usw. können im Eigenschaftfenster definiert und auch innerhalb einer Prozedur angesprochen und ggf. verändert werden.

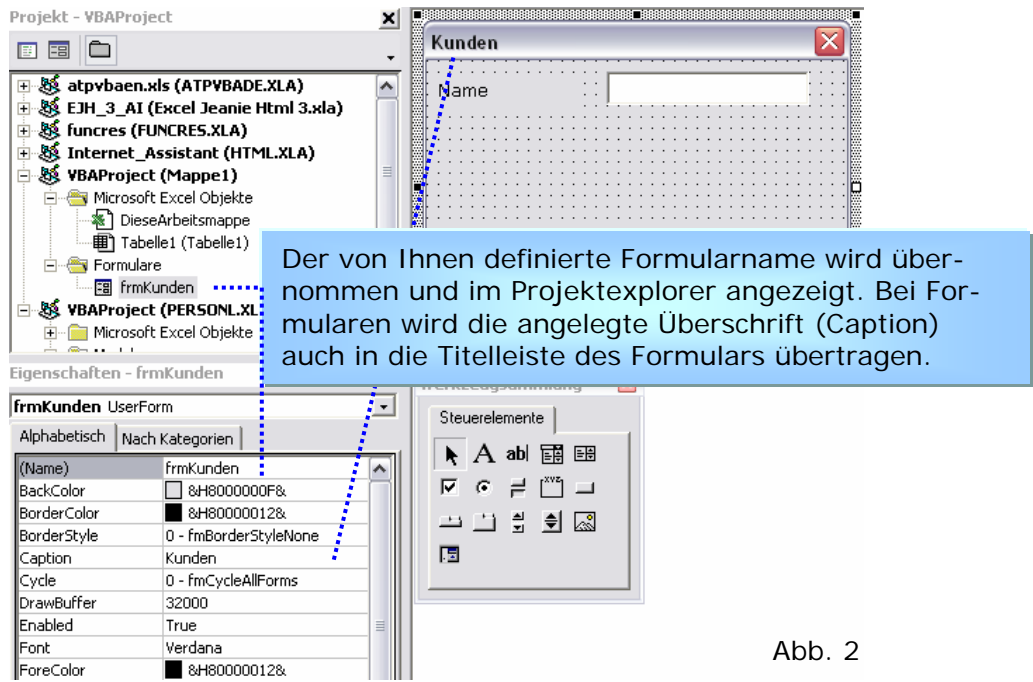


Abb. 2

Mit einem Doppelklick auf das jeweilige Steuerelement (bei der UserForm über den Button 'Code anzeigen', Abb.1 oder das gleichnamige Kontextmenü) öffnet sich das Codefenster mit zwei Codezeilen, die Excel automatisch erstellt hat. Bei einem angeklickten Bezeichnungs- oder Textfeld beispielsweise mit folgendem Aussehen:



Abb. 3

Bei den blau formatierten Worten handelt es sich um sog. Schlüsselwörter die von Excel fest vergeben werden und deren Bedeutung innerhalb einer Prozedur definiert ist. `Option Explicit` bedeutet beispielsweise, dass alle Variablen innerhalb des Moduls 'explizit' definiert werden müssen. Excel überwacht dann, ob Variablennamen innerhalb der Prozedur angelegt und auch richtig geschrieben werden. Diese Option kann im Visual Basic-Editor – Menü: Extras – Optionen – Register: Editor - 'Variablendeklaration erforderlich' aktiviert und ggf. auch deaktiviert werden.

Mit den Schlüsselwörtern `Sub` und `End Sub` wird eine Prozedur aufgerufen und nach der Ausführung wieder beendet, mit `Exit Sub` kann sie auch an einer bestimmten Position unmittelbar verlassen werden.

Mit `Private` wird eine Variable oder Anweisung nur innerhalb des aktuellen Moduls definiert – im Gegensatz zu `Public`. Mit diesem Schlüsselwort wird eine Variable als 'öffent-

lich' deklariert, was bedeutet, dass von allen Prozeduren und in allen Modulen auf sie zugegriffen werden kann.

Hilfe zu den einzelnen Schlüsselwörtern, aber auch zu Eigenschaften und Methoden etc. erhalten Sie, indem Sie den Cursor in das entsprechende Wort setzen und die F1-Taste drücken.

Im Projekttexplorer sehen Sie, dass Ihre Arbeitsmappe (Workbook) und die einzelnen darin enthaltenen Tabellen (Worksheets), einschließlich der Formulare und - falls vorhanden - auch Module (hier werden Makros und Prozeduren abgelegt) etc. unter dem Begriff 'VBA-Project' zusammengefasst werden.

Da Sie jedem einzelnen Objekt Ihres VBA-Projects auch Code zuweisen können, kann mit einem Doppelklick auf das jeweilige Objekt, dessen Codefenster geöffnet werden. Mit Hilfe der beiden Kombinationsfelder 'Objekte' und 'Prozeduren', die sich oberhalb des Codefensters (s. Abb. 3) befinden, können nun jedem Element ggf. vorhandene Objekte und entsprechende Ereignisse zugeordnet werden.

Beispiel:

alle Objekte - nach einem Doppelklick auf 'DieseArbeitsmappe':

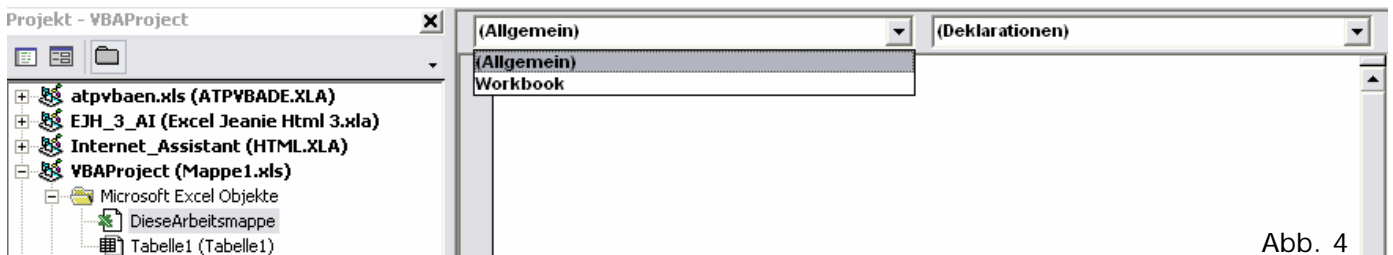


Abb. 4

Beispiel:

alle Objekte und Ereignisse nach einem Doppelklick auf 'Tabelle1':

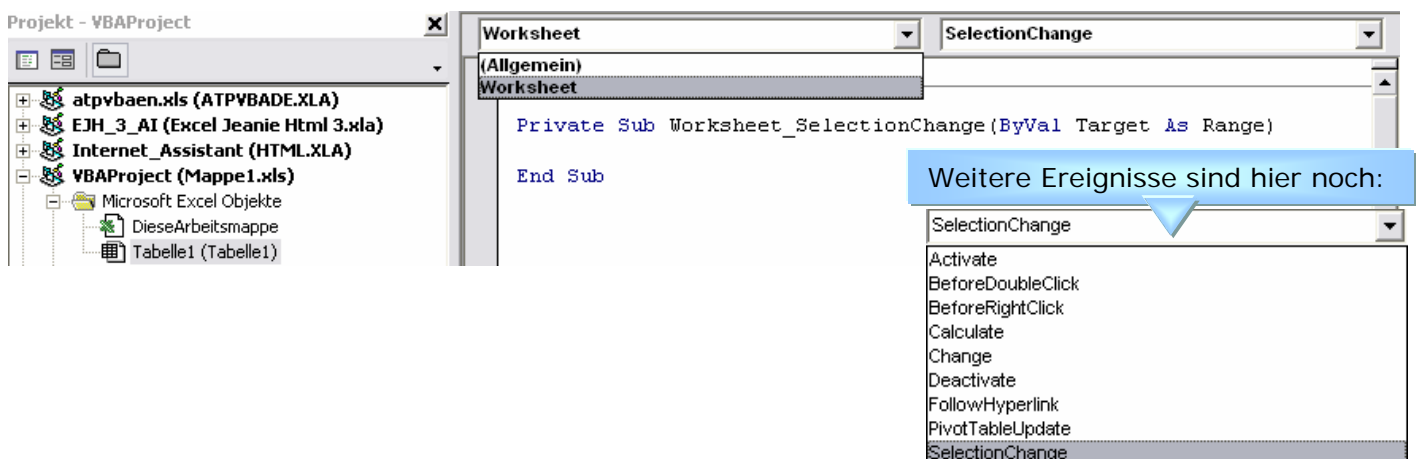


Abb. 5

Beispiel:

Objekte und Ereignisse nach einem Doppelklick auf das Textfeld 'txtName' im Formular 'frmKunden':

Bei diesem Objekt können noch weitere Ereignisse definiert werden:

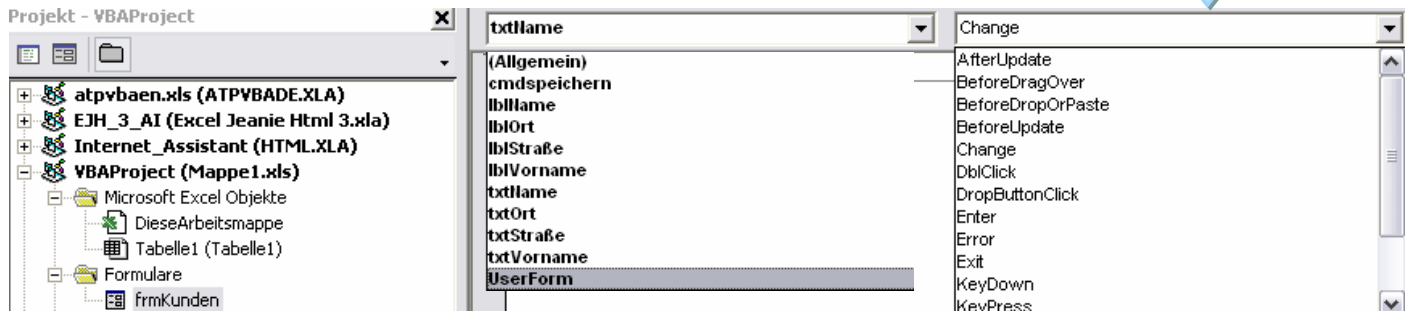


Abb. 6

Durch Anklicken des Buttons 'Sub/UserForm ausführen' kann von der Entwurfsansicht in die Formularansicht gewechselt werden, um entsprechende Eingaben vorzunehmen oder Aktionen auszulösen.

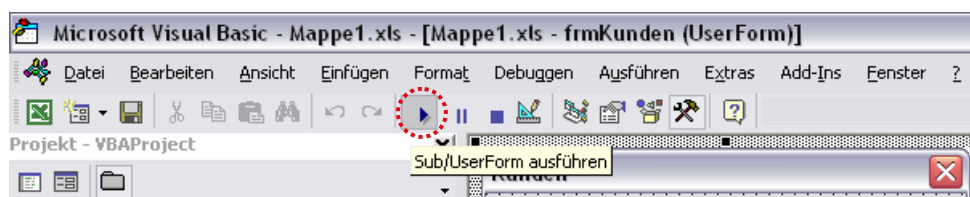


Abb. 7

Als nächstes muss nun festgelegt werden, bei welchem Ereignis das Formular angezeigt werden soll – beim Anklicken eines Buttons oder wenn die Arbeitsmappe geöffnet wird usw.

Soll es gleich beim Öffnen der Arbeitsmappe angezeigt werden, muss der entsprechende Code dem Objekt 'DieseArbeitsmappe' zugewiesen werden. Deshalb wird zunächst mit einem Doppelklick das Codefenster des Objekts geöffnet und im Kombinationsfeld der Eintrag 'Workbook' aktiviert. Als Ereignis muss im Kombinationsfeld 'Prozedur' - 'Open' aktiviert werden. Excel hat wieder zwei Codezeilen generiert, die entsprechende Öffnen-Anweisung muss nun dazwischen eingegeben werden. Dazu ist zunächst der Name des Formulars 'frmKunden' einzugeben und nach einem Punkt die Anweisung: 'Show'.

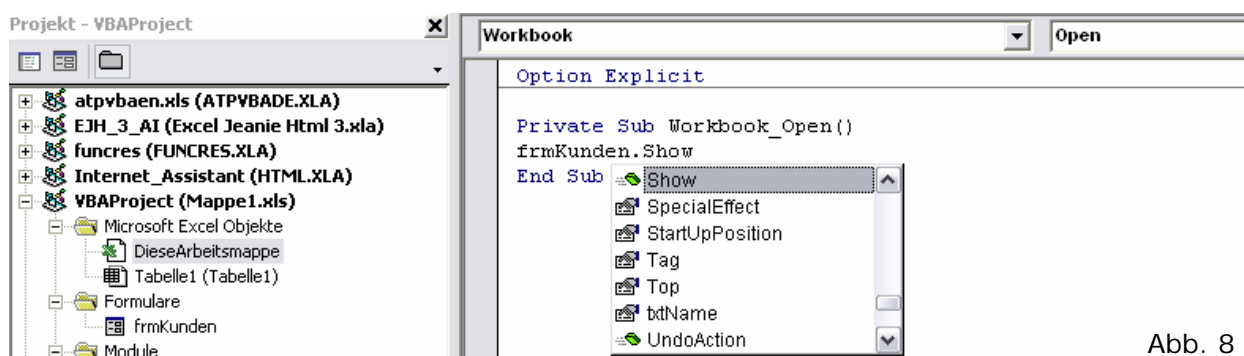




Abb. 8

Beim Erstellen dieser Codezeile werden Sie feststellen, dass Excel nach Schreiben des Punktes eine Auswahl an Eigenschaften und Methoden anbietet, die für dieses Objekt verfügbar sind. Den ausgewählten Eintrag können Sie auch durch Drücken der Tab-Taste an Ihrer Codeposition anfügen.

Optisch erkennbar sind Methoden, also Anweisungen was das ausgewählte Objekt zu tun, bzw. auszuführen hat, auch an diesem Symbol: 

Eigenschaften sind Merkmale eines Objekts, beispielsweise die Größe oder Farbe, aber auch Informationen, die seinen Zustand definieren; sie werden mit diesem Symbol gekennzeichnet: 

Tipp: Wenn Sie Code mit Kleinbuchstaben schreiben wird er – sofern die Syntax richtig ist – von Excel automatisch beim Verlassen der Codezeile in Groß- und Kleinbuchstaben umformatiert. So können Sie selbst Schreibfehler sofort erkennen und entsprechend korrigieren.

Soll das Formular mit Hilfe eines Buttons angezeigt werden, zeichnen Sie einen 'Command-Button' (Symbolleiste: 'Steuerelement-Toolbox') in Ihre Tabelle. Beachten Sie, dass Excel nun in einen Entwurfsmodus wechselt, den Sie wieder 'beenden' müssen, wenn Sie keine weiteren Steuerelemente einfügen wollen und auch die Codebearbeitung abgeschlossen ist.



Abb. 9

Wenn Sie nun die Schaltfläche mit der rechten Maustaste anklicken, sehen Sie im Kontextmenü u.a. die Einträge 'Eigenschaften' und 'Code anzeigen'. Nach Anklicken dieser Menüs öffnen sich auch hier die bereits beschriebenen Objekte und Sie können im Codefenster sehen, dass Excel der Schaltfläche gleich das 'Click'-Ereignis zugewiesen hat. Auch hier lässt sich nun die 'Show-Anweisung' einbinden, so wie bereits beschrieben.

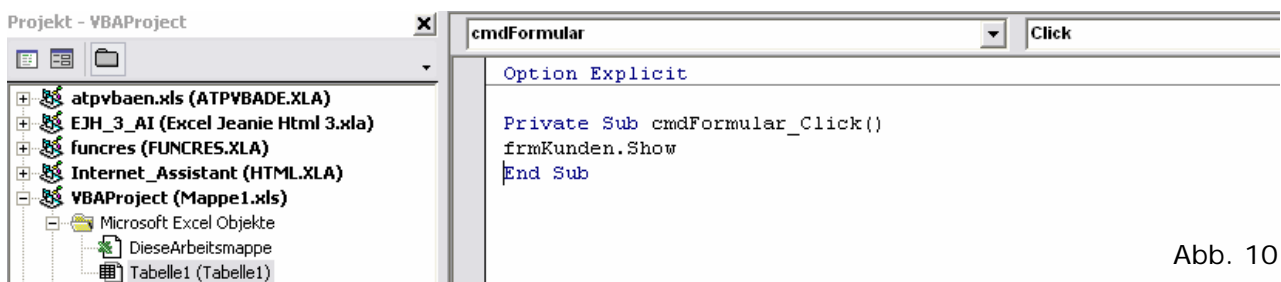


Abb. 10

Eine wichtige Rolle im Zusammenhang mit dem Anzeigen einer UserForm spielt auch das Ereignis: 'Initialize'. Mit Hilfe der Initialisierung können bereits vor dem Anzeigen des Formulars verschiedene Einstellungen festgelegt werden. So z.B. ob beim Öffnen des Formulars einzelnen Steuerelementen ein bestimmter Wert zugewiesen werden soll oder ob Objekte zunächst ausgeblendet sein sollen, um sie erst zu einem späteren Zeitpunkt einzublenden.

Im folgenden Beispiel soll der Speichern-Button beim Öffnen eines Formulars ausgeblendet sein und erst eingeblendet werden, wenn das Textfeld 'Ort' nach einer Eingabe wieder verlassen wird.

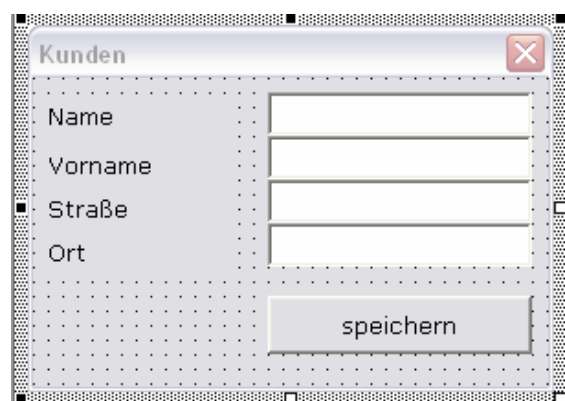
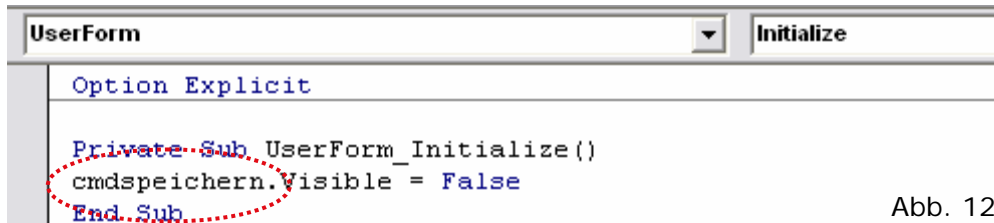


Abb. 11

Um das zu realisieren, muss das Formular wie folgt initialisiert werden:

Klicken Sie im Projekt-Explorer auf Ihr Formular, damit es im Entwurfsmodus angezeigt wird und anschließend im Kontextmenü auf den Eintrag 'Code anzeigen'. Wählen Sie im Kombinationsfeld 'Objekte' (oberhalb des Codefensters) den Eintrag 'UserForm' aus und im Kombinationsfeld 'Prozedur' (rechts daneben) den Eintrag 'Initialize'.

Schreiben Sie zwischen die beiden automatisch generierten Codezeilen die folgende Anweisung:



```
Option Explicit

Private Sub UserForm_Initialize()
    cmdspeichern.Visible = False
End Sub
```

Abb. 12

Anmerkung:

cmdspeichern ist der in den Eigenschaften des Buttons definierte Name. Wenn Sie nach diesem Eintrag einen Punkt schreiben, wird

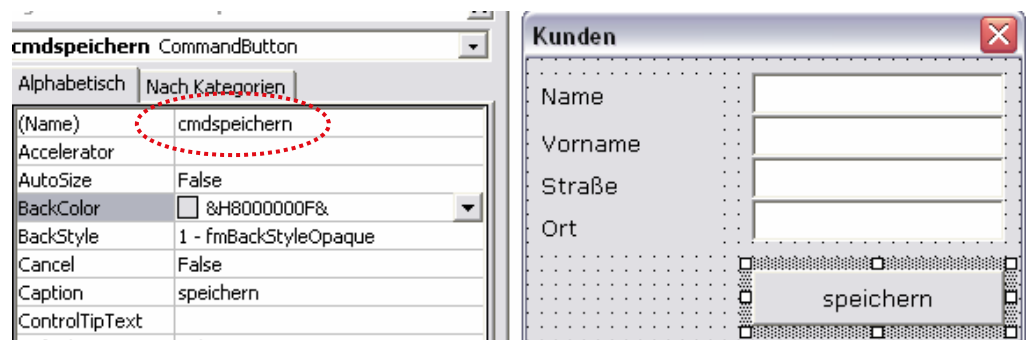


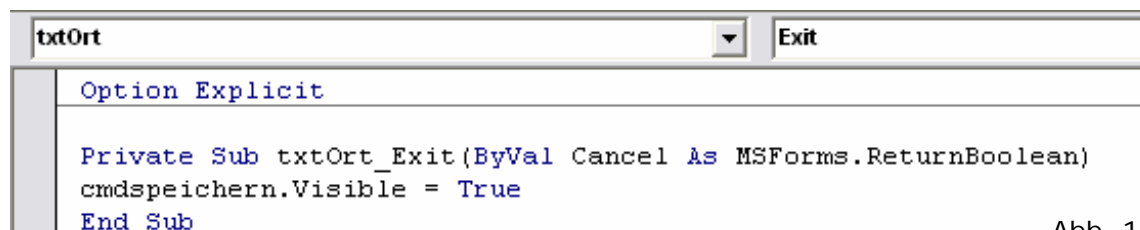
Abb. 13

Ihnen wieder eine Auswahlliste der Methoden und Eigenschaften angezeigt und Sie können den Eintrag 'visible' – sichtbar – durch Anklicken der Tab-Taste an Ihren bisherigen Code anhängen. Da der Standardwert der Eigenschaften immer 'True' – also Wahr - ist, muss der Wert für einen auszublendenden Button auf 'False' – Falsch – gesetzt werden.

Um den Button nun beim Verlassen des Textfeldes 'Ort' einzublenden, müssen Sie wie folgt verfahren:

Das Ereignis ist zunächst dem Textfeld 'Ort' zuzuweisen. Das Codefenster dieses Steuerelements öffnen Sie mit einem Doppelklick auf dieses Objekt. Excel bietet Ihnen hier standardmäßig das Ereignis 'Change' – Änderung - an und Sie müssen im Kombinationsfeld 'Prozedur' den Eintrag 'Exit' – beim Verlassen – aktivieren.

Schreiben Sie anschließend die 'visible'-Anweisung wieder zwischen die beiden Codezeilen:



```
Option Explicit

Private Sub txtOrt_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    cmdspeichern.Visible = True
End Sub
```

Abb. 14

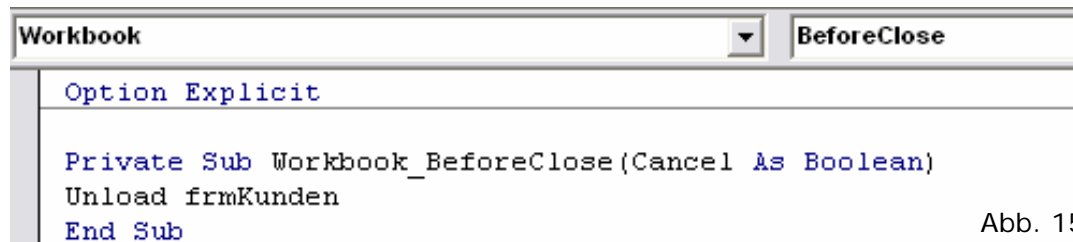
Anmerkung:

Beim Verlassen des Textfeldes 'Ort' wird die Eigenschaft 'visible' (sichtbar) der Befehlschaltfläche 'cmdspeichern' auf 'True' (Wahr) gesetzt.

In Ihrem Formular wird nun, nachdem Sie in dieses Textfeld einen Ortsnamen eingegeben und mit der Enter-Taste bestätigt haben, der Speichern-Button eingeblendet.

Zum Schluss noch ein Hinweis...

Wenn Sie innerhalb Ihrer Anwendung, beispielsweise mit Hilfe von Schaltflächen Formulare öffnen und schließen, sollten Sie diese – sicherheitshalber - wenn sie nicht mehr benötigt werden, wieder 'entladen'. Die Anweisung hierfür lautet: `Unload frmKunden` - in unserem Fall, erfolgt dies beim Ereignis 'BeforeClose' des Objekts 'Arbeitsmappe':



```
Workbook BeforeClose
Option Explicit
Private Sub Workbook_BeforeClose(Cancel As Boolean)
Unload frmKunden
End Sub
```

Abb. 15

Wird ein Formular mit `visible = false` ausgeblendet, ist es immer noch im Speicher verfügbar. Mit der Unload-Anweisung wird es aus dem Speicher entfernt und dieser wird wieder freigegeben.