

Und im letzten Fenster *Verankern* (Bild 1-3) sind alle möglichen Fenster der IDE aufgeführt. Hauptsächlich benötigen wir zur Entwicklung und Ausführung den Projekt-Explorer und das Eigenschaftsfenster. Die größte Fläche zeigt den Code des jeweils ausgewählten Objekts und wird als VBA-Editor bezeichnet. Mit einem Doppelklick auf ein Objekt im Projekt-Explorer öffnet sich das zugehörige Code-Fenster automatisch. Lediglich die Formulare (*UserForms*) machen da eine Ausnahme. Sie besitzen eine eigene Oberfläche und es kann zwischen Code-Ansicht und Objekt-Ansicht (Bild 1-4) umgeschaltet werden. Alle Einstellungen in den Optionen bleiben bis zu ihrer Änderung bestehen, auch über eine Excel-Mappe hinaus.

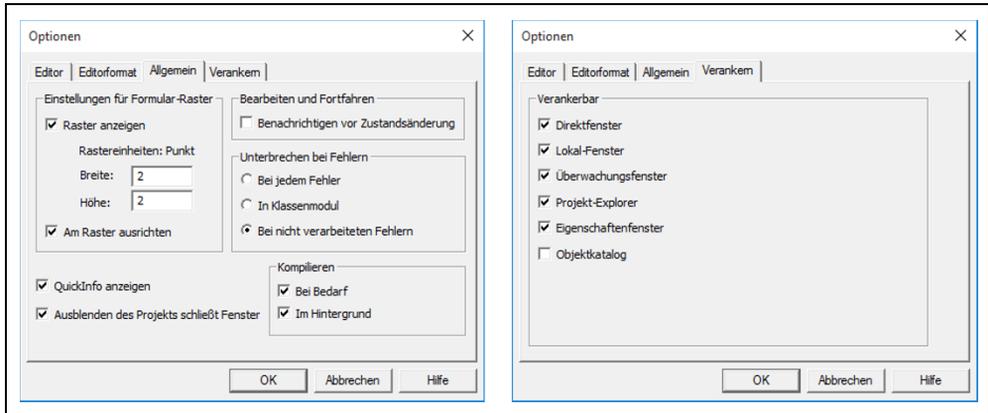


Bild 1-3. Optionen Allgemein und Verankern

## 1.1.2 Projekt und Projekt-Explorer

Eine Excel-Anwendung, zu der neben Tabellen auch Benutzerflächen, Programmmodule, Objekte und Prozeduren gehören, wird in der IDE als Projekt verwaltet. Das ist die Aufgabe des Projekt-Explorers (Bild 1-4). Jedes Projekt besitzt einen Namen, der beim Neustart einer Excel-Anwendung immer *VBAProjekt* lautet. Jedes Projekt sollte aber einen eigenen aussagekräftigen Namen bekommen, in dem keine Sonderzeichen und Leerzeichen vorkommen. Ziffern sind nur innerhalb des Namens erlaubt.

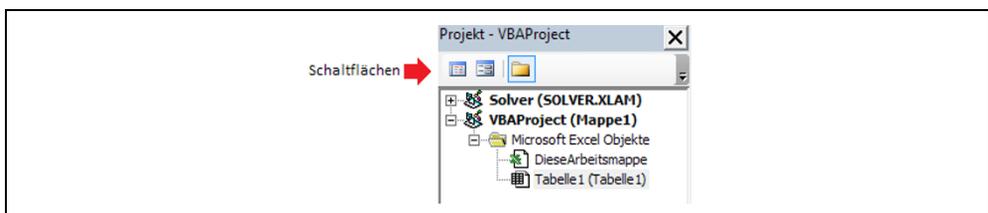


Bild 1-4. Der Projekt-Explorer im Visual Basic-Editor

Wichtig ist noch, dass in der Menüleiste des Projekt-Explorers zwei Schaltflächen von großer Bedeutung liegen. Mit diesen Schaltflächen wird die Darstellung des aktiven Objekts bestimmt. Also ob im VBA-Editor ein Objekt (z. B. eine Benutzerfläche) als Ansicht oder die zugehörigen Prozeduren als Programmcode dargestellt werden. Ein im Projekt-Explorer ausgewähltes Objekt ist auch immer im VBA-Editor sichtbar, soweit dies von der Art des Objekts möglich ist.

### 1.1.3 Der Objektkatalog

Der Objektkatalog (Bild 1-5), der sich genauso wie alle anderen Fenster in der IDE über die Registerkarte *Ansicht* im Editor einschalten lässt, zeigt die Objekt-Klassen, Eigenschaften, Methoden, Ereignisse und Konstanten an, die in den Objektbibliotheken und Prozeduren dem jeweiligen Projekt zur Verfügung stehen. Mithilfe dieses Dialogfensters lassen sich Objekte der Anwendung, Objekte aus anderen Anwendungen oder selbstdefinierte Objekte suchen und verwenden. Der Objektkatalog wird eigentlich nur bei Bedarf geöffnet.

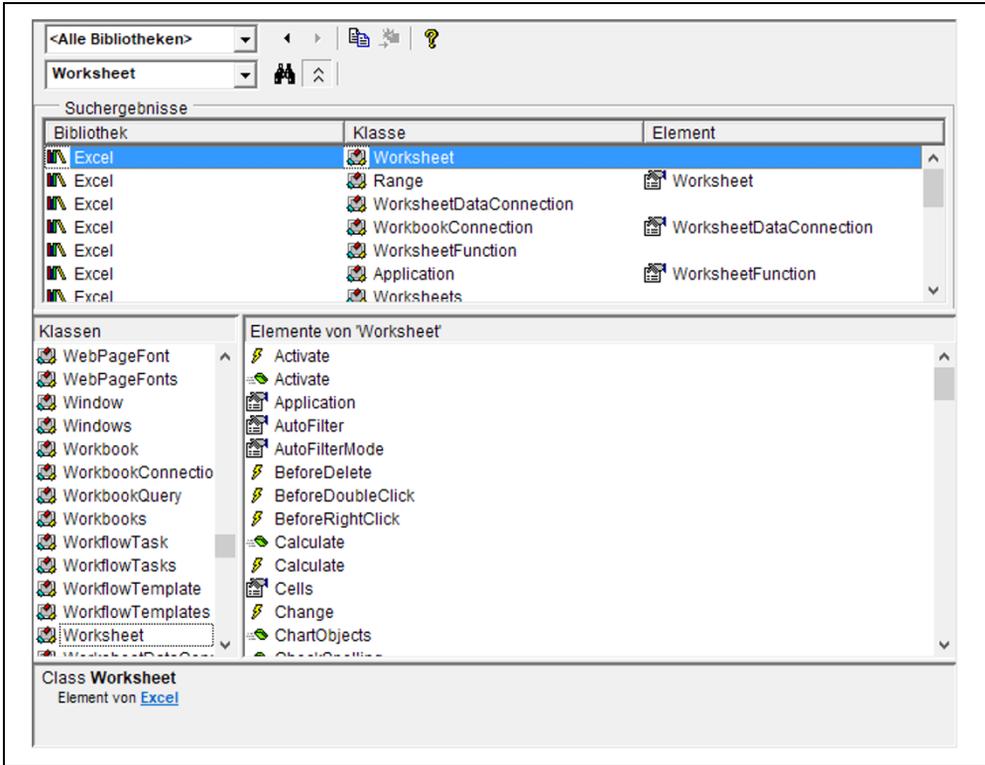


Bild 1-5. Der Objektkatalog im Visual Basic-Editor

### 1.1.4 Das Eigenschaftsfenster

Das Eigenschaftsfenster ist für die Entwicklung sehr wichtig. Es listet für das ausgewählte Objekt die Eigenschaften zur Entwurfszeit und deren aktuelle Einstellungen auf. Im oberen Auswahlfenster steht in fetter Schrift das Objekt und dahinter in einfacher Schrift der Objekttyp (Bild 1-6). Das Auswahlfenster lässt sich auch öffnen und es kann ein anderes Objekt aus einer Liste aller vorhanden Objekte ausgewählt werden. Darunter stehen die Eigenschaften, die das ausgewählte Objekt besitzt. Einfacher ist die Auswahl aber durch Anklicken eines Objekts im Projekt-Explorer oder VBA-Editor.

Die angezeigten Eigenschaften lassen sich zur Entwurfszeit durch Anklicken bzw. durch Doppelklick ändern. Werden mehrere Objekte gleichzeitig ausgewählt, dann enthält das Eigenschaftsfenster eine Liste nur der Eigenschaften, die die ausgewählten Objekte gemeinsam besitzen. So müssen mehrere Objekte nicht einzeln geändert werden. Die Darstellung des Eigenschaftsfensters wird unter dem Register *Ansicht* in der IDE geregelt, so

wie alle anderen Fenster auch. Es lässt sich durch Anfassen der Kopfzeile mit der Maus frei verschieben, ist aber in der Regel an den linken Rand der IDE geheftet.

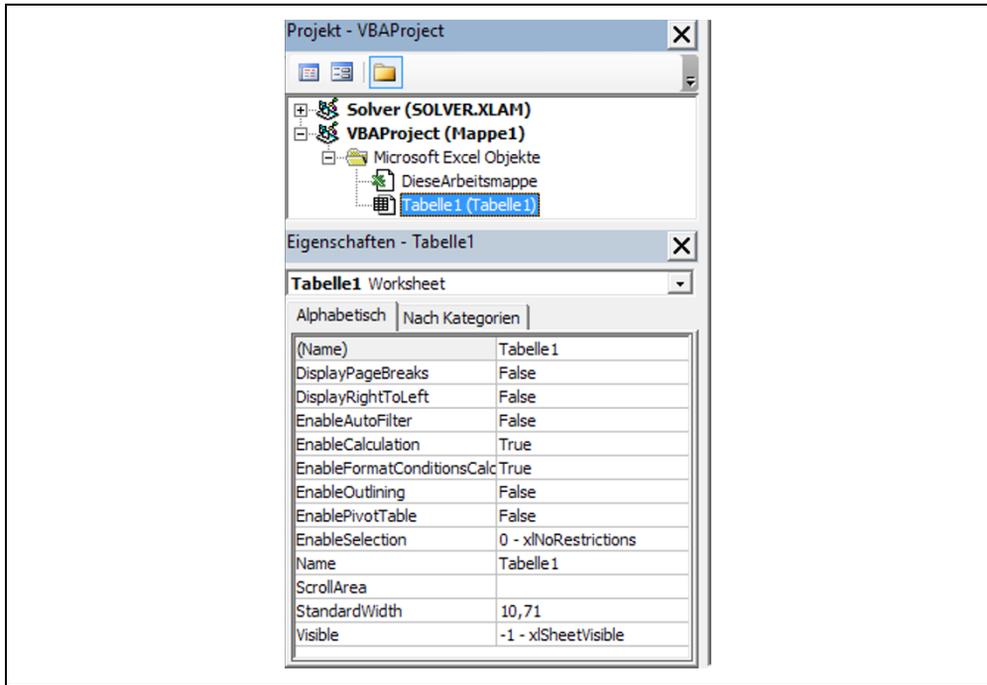


Bild 1-6. Das Eigenschaftsfenster im Visual Basic-Editor

### 1.1.5 Die Direkt-, Lokal- und Überwachungsfenster

Die Direkt-, Lokal- und Überwachungsfenster dienen zum Testen der Programme. Das Direktfenster dient zum Eingeben und zum Ausführen von Programmcode zur Fehlersuche. Das Lokalfenster zeigt alle deklarierten Variablen in der aktuellen Prozedur und deren Werte. Das Überwachungsfenster überwacht Ausdrücke und alarmiert beim Eintreten von Randbedingungen. Zur weiteren Einarbeitung empfehle ich die Literaturhinweise. Für unsere Programmierung sind sie zunächst von untergeordneter Bedeutung.

#### Übung 1.1 Neues Projekt anlegen

- Im ersten Schritt wird eine neue Excel-Anwendung geöffnet und die Einstellungen, wie unter 1.1 beschrieben, werden vorgenommen.
- Diese Einstellungen sind zukünftig nicht mehr nötig.
- Die IDE wird geöffnet (ALT + F11).
- Mit einem Mausklick im Projekt-Explorer auf den Projektnamen, kann dieser unter *Name* im Eigenschaftsfenster geändert werden, z. B. *ErstesProjekt*.
- Die Änderung ist anschließend im Projekt-Explorer zu sehen.
- Eine Namensänderung kann beliebig oft durchgeführt werden.

## 1.2 Objekte

### 1.2.1 Objekte, allgemein

Der grundlegende Begriff in der IDE ist das Objekt. Einem Objekt werden Eigenschaften, Methoden und Ereignisse zugeordnet. Die Schreibweise von Objekt, Unterobjekt und Methode oder Eigenschaft ist:

```
Objekt[.Unterobjekt ... ][.Methode]
```

oder

```
Objekt[.Unterobjekt ... ][.Eigenschaft]
```

Soll z. B. ein Textfeld *TextBox1* auf dem Formular *UserForm1* den Text „Dies ist ein Test!“ zugewiesen bekommen (Erläuterung kommt später), dann lautet die Anweisung:

```
UserForm1.TextBox1.Text = "Dies ist ein Test!"
```

Ein fester Text wird durch Anführungszeichen gekennzeichnet. Soll der Text nicht angezeigt werden, dann kann die Eigenschaft *Visible* des Textfeldes auf den Wert *False* gesetzt werden.

```
UserForm1.TextBox1.Visible = False
```

Ereignissen von Objekten sind in der Regel Ereignis-Prozeduren zugeordnet. Zum Beispiel ein Klick mit der linken Maustaste auf die Befehlsschaltfläche mit dem Namen *cmdStart* eines Formulars im Entwurfsmodus zeigt im Codefenster des Formulars die Prozedur:

```
Private Sub cmdStart_Click()  
End Sub
```

Der Name einer Ereignis-Prozedur besteht immer aus den Namen von Objekt und Ereignis, verbunden durch einen Tiefstrich. Diese Prozedur bekommt dann alle Anweisungen, die ausgeführt werden sollen, wenn die Schaltfläche im Ausführungsmodus angeklickt wird.

Geschieht die obere Textzuweisung im Formular *UserForm1*, auf dem sich auch das Textfeld *TextBox1* befindet, dann genügt es zu schreiben:

```
TextBox1.Text = "Dies ist ein Test!"
```

Es muss also nicht immer die gesamte Objekthierarchie genannt werden, wenn das Objekt eindeutig bestimmt ist. Ebenso gibt es gewisse Grundeinstellungen (Default-Werte), die wir in der Aufzählung von Objekt und Eigenschaft nicht nennen müssen. Bei einem Textfeld ist der Default-Wert *Text* und so genügt es zu schreiben:

```
TextBox1 = "Dies ist ein Test!"
```

damit der Eigenschaft *Text* des Objekts *TextBox1* der konstante Text zugewiesen wird. Objekte in VBA sind Dokumente, Tabellen, Folien, Abfragen etc. Aber auch Fenster, Module, Steuerelemente, Statusleisten, Menüelemente usw. Grundsätzlich ist zwischen Objektlisten und den Objekten selbst zu unterscheiden. Objekte (Workbook, Worksheet etc.) in Objektlisten (Workbooks, Worksheets etc.) werden über ihren Namen oder einen Index angesprochen.

Einige wichtige Objektlisten in Office-Anwendungen:

- Documents enthält alle Word-Dokumente einer Word-Anwendung
- Workbooks enthält geöffnete Excel-Arbeitsmappen
- Worksheets enthält alle Tabellen einer Excel-Anwendung
- Windows enthält alle aktiven Fenster.

Um ein Objekt in einer solchen Objektliste anzusprechen, wird ein Index benutzt. So liefert

```
MsgBox ThisWorkbook.Worksheets(2).Name
```

den Namen des zweiten Arbeitsblattes (eine Eigenschaft des Objekts Worksheet) in der aktuellen Excel-Arbeitsmappe. Die Ausgabe erfolgt mit der Funktion *MsgBox*. Sie zeigt den Namen in einem Dialogfeld und wartet auf eine Bestätigung. Der Zugriff auf ein nicht vorhandenes Objekt erzeugt den Laufzeitfehler 1004. Aus diesen Listen ergeben sich dann je nach Anwendung die Objekte:

- Application – aktives Objekt
- FileSearch – Objekt zum Suchen nach Dateien
- Assistant – Objekt des Office-Assistenten
- Document – Word-Dokument
- Workbook – Excel-Arbeitsmappe
- Presentation – Objektklasse PowerPoint
- MsgBox – Meldefenster
- InputBox – Anzeige eines Eingabedialogs.

Objekten können wir im Eigenschaftsfenster unter Namen einen anderen Namen zuordnen. Dabei geben wir dem Objektamen ein Präfix, mit dem wir die Objektart kennzeichnen. Dies ist nicht zwingend einzuhalten, dient aber der Übersichtlichkeit im Programmcode. Ein einmal festgelegtes Präfix sollte dann nicht mehr geändert werden. Beispiele:

- tbl – Tabelle (sheet)
- frm – Form (user form)
- mod – Modul (module)
- com – Befehlsschaltfläche (command)
- wbk – Excel-Arbeitsmappe (workbook)
- wsh – Excel-Arbeitsblatt (worksheet).

Ein schneller Zugang zu Eigenschaften und Methoden eines Objekts ist durch das *Kontextmenü* möglich. Dabei wird das Objekt mit einem Klick der linken Maustaste im Entwurfsmodus markiert. Mit einem Klick auf die rechte Maustaste öffnet sich ein Dialogfenster, das sogenannte *Kontextmenü*. Darin werden die wichtigsten Methoden und Eigenschaften des markierten Objekts zur Auswahl angezeigt. Das Kontextmenü existiert auch in der Anwendung.

## 1.2.2 Anwendungen und Makros

Unter Anwendungen sind die Office Programme Word, Excel, Outlook, PowerPoint etc. zu verstehen. In diesen Anwendungen lassen sich Prozeduren und Funktionen programmieren oder mit der Makro-Funktion erzeugen.

Die Makro-Funktion wird mit der Registerkarte *Entwicklertools* und der Auswahl *Makro aufzeichnen* in der Menügruppe *Code* gestartet. Eine zweite Startmöglichkeit besteht darin,