

2 Workshop: Die Entwicklungsumgebung

Visual Basic ist zum einen eine Programmiersprache, zum anderen jedoch auch ein hochintegriertes Werkzeug, um Programme zu erstellen.

Sie erhalten Visual Basic immer als Komplettpaket, bestehend aus der Programmiersprache selbst, den notwendigen Werkzeugen, die Sie bei der Erstellung eigener Programme unterstützen, und einer Anzahl von Zusatzprogrammen, die oft benötigte Aufgaben stark vereinfachen. All dies ist bei Visual Basic in der so genannten Entwicklungsumgebung zusammengefasst.

In diesem Kapitel wird die Entwicklungsumgebung in aller Kürze vorgestellt. Die wichtigsten Oberflächenelemente und Werkzeuge werden benannt und gezeigt. Zudem wird besprochen, was ein Visual Basic-Projekt ist.

Programmiersprache und Entwicklungsumgebung

2.1 Die Oberfläche von Visual Basic 6.0

Wenn Sie Visual Basic 6.0 starten, sehen Sie das Fenster aus Abbildung 2.1. Falls Sie Visual Basic zum ersten Mal gestartet haben, werden Sie feststellen, dass eine gewisse Einarbeitungszeit benötigt wird, um alle Elemente der Oberfläche kennen zu lernen und mit ihnen arbeiten zu können.

Oberfläche

Aber denken Sie daran, diese Entwicklungsumgebung wird Ihnen die Entwicklung Ihrer Programme auf einfachste Weise erlauben, Sie müssen sie eben nur erst kennen lernen.

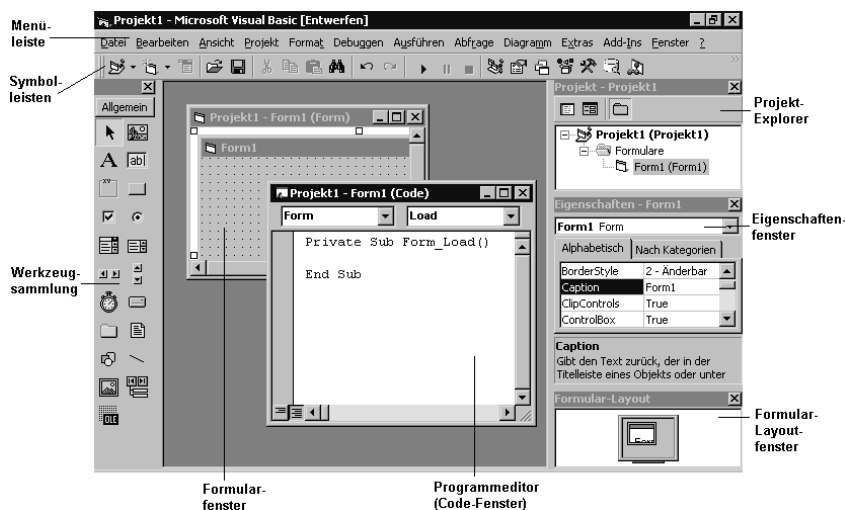


Abbildung 2.1: Die Visual Basic 6.0 Entwicklungsumgebung

Um Ihnen einen leichten Einstieg zu ermöglichen, werden in den folgenden Abschnitten zunächst die wichtigsten Elemente der Oberfläche benannt und erläutert.

2.2 Die wichtigsten Menüs

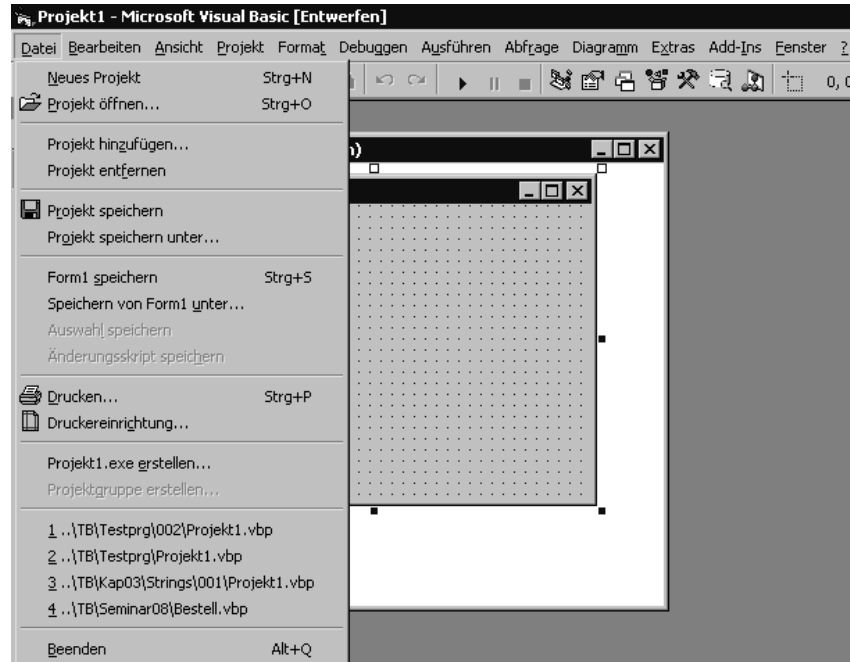
Für einen ersten Überblick werden in den folgenden Abschnitten die Menüs von Visual Basic gezeigt und kurz erläutert. Viele Funktionen der Entwicklungsumgebung können nicht nur über die *Menüs*, sondern auch über *HotKeys*, *Funktionstasten* oder objektbezogene *PopUp-Menüs* aufgerufen werden.

2.2.1 Menü Datei

Menü Datei Im Menü DATEI (Abbildung 2.2) finden Sie alle Funktionen, die notwendig sind, um ein Projekt oder einzelne Dateien des Projektes zu erstellen, zu laden oder zu speichern.

Sie können in diesem Menü Ausdrücke starten, das Programm erstellen (kompilieren) und Visual Basic beenden. Darüber hinaus wird in diesem Menü eine Liste der zuletzt geöffneten Projekte geführt, die dazu verwendet werden kann, diese Projekte zu laden, ohne dafür einen entsprechenden Dialog öffnen zu müssen.

Abbildung 2.2:
Das Menü Datei



2.2.2 Übung: Anlegen eines neuen Projekts

Legen Sie ein Standardprojekt an und speichern Sie es.

Da dieses Projekt keinen Programmcode hat, ist es auf der beiliegenden CD nicht gespeichert.

Lösung

Über das Menü DATEI->NEUES PROJEKT wird der Dialog NEUES PROJEKT (Abbildung 2.3) geöffnet.

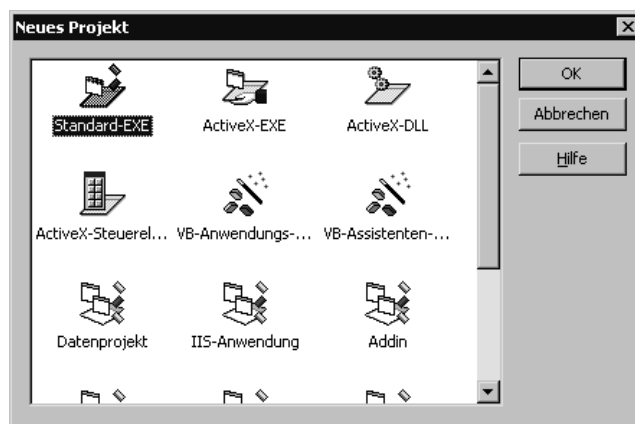


Abbildung 2.3: Dialog Neues Projekt

Wählen Sie den Eintrag *Standard-EXE* und betätigen Sie die Schaltfläche *OK*. Visual Basic erstellt jetzt automatisch ein Projekt mit allen notwendigen Projektdateien. Das zugehörige Modul wird automatisch geöffnet, so dass Sie mit dem Oberflächendesign sofort anfangen können.

Standard-EXE

Bei einer *Standard-EXE* wird dem Projekt ein Formular hinzugefügt, wie Sie im Projekt-Explorer (Abbildung 2.4) erkennen können.

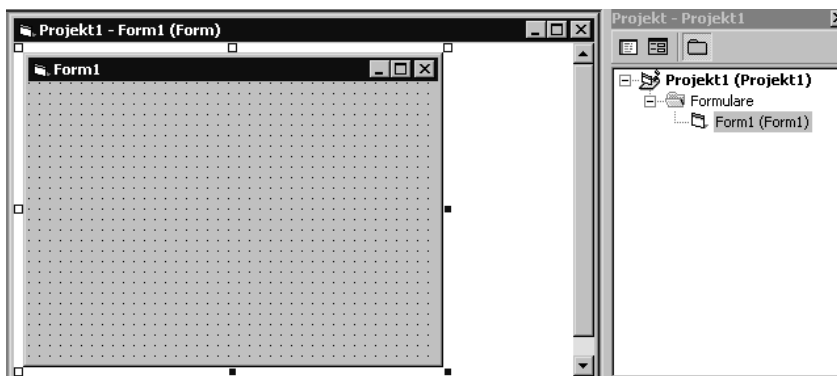


Abbildung 2.4: Ein Standard-Projekt wurde angelegt

2 Workshop: Die Entwicklungsumgebung

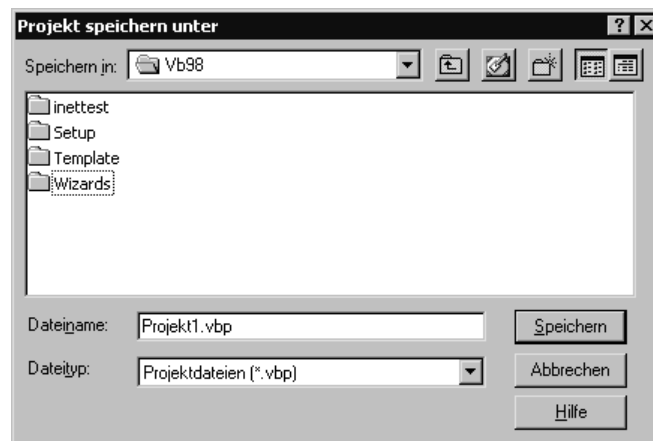
Projekt speichern Um das Projekt zu speichern, ist wiederum eine Auswahl des Menüs DATEI erforderlich. Die einfachste Art ist die Auswahl des Menüpunktes PROJEKT SPEICHERN. Dieser sorgt dafür, dass alle dem Projekt zugehörigen Dateien gespeichert werden.

Wurde das Projekt bisher nicht gespeichert, so erhalten Sie Dialoge, die Ihnen die Möglichkeit geben jeder einzelnen Projektdatei einen Namen zu geben und das Verzeichnis auszuwählen, in dem sie gespeichert werden soll.

das Formular speichern Bei unserem Projekt werden nacheinander zwei Dialoge aufgeblendet. Der erste Dialog verlangt die Speicherung der Datei *Form1.frm*. Mit der Endung *frm* werden Formulare gespeichert. Es handelt sich um das Formular *Form1*, das gespeichert werden soll.

die Projektdatei speichern Ein zweiter Dialog (Abbildung 2.5) verlangt das Speichern der Datei *Projekt1.vbp*. Mit dieser Endung wird das Projekt selbst gespeichert. In der Projektdatei sind alle Informationen enthalten, die notwendig sind um das Projekt zu einem späteren Zeitpunkt wieder zu laden.

Abbildung 2.5:
Speichern der
Projektdatei



Verwenden Sie beim Speichern von Projekten sprechende Namen. Es ist sonst mühsam, unter durchnummerierten Dateien eine bestimmte zu finden.

2.2.3 Menü Bearbeiten

Menü Bearbeiten Das Menü BEARBEITEN (Abbildung 2.6) bietet neben den nun fast zum Standard gewordenen Punkten wie *Rückgängig*, *Kopieren*, *Ausschneiden*, *Einfügen*, *Suchen und Ersetzen* etc. einige neue und hilfreiche Funktionen.

Diese betreffen vor allem den Editor und dienen der Fehlerminimierung bei der Eingabe von Code. Eine nähere Beschreibung dieser Hilfsmittel finden Sie im Abschnitt über den *Editor*.

Die wichtigsten Menüs

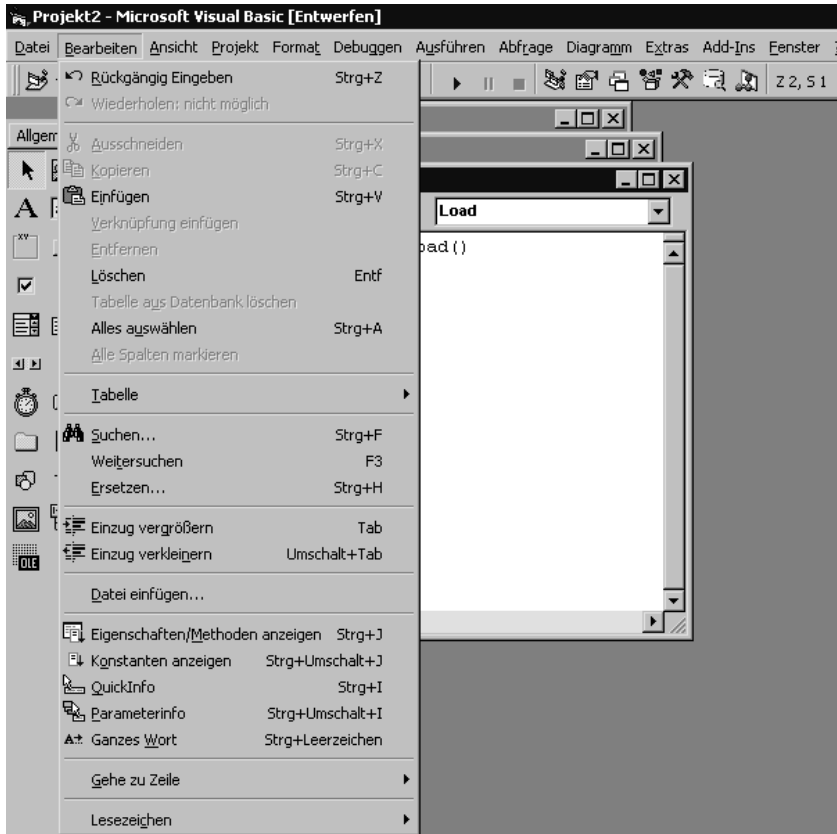


Abbildung 2.6:
Menü Bearbeiten

2.2.4 Menü Ansicht

Über die Menü-Befehle des Menüs ANSICHT (Abbildung 2.7) können Sie die Darstellung und Anzeige der Fenster in der Entwicklungsumgebung einstellen. Es stehen Menü-Befehle zum Öffnen des Codefensters oder der Objektansicht sowie zum Ein- und Ausblenden der verschiedenen Hilfsfenster zur Verfügung.

Menü Ansicht

2.2.5 Menü Projekt

Über das Menü PROJEKT (Abbildung 2.8) können Sie die dem Projekt zugehörigen Dateien und Objekte verwalten. Hier können Sie neue Komponenten wie *Formulare*, *Module*, *Klassen* etc. den offenen Projekten hinzufügen oder aus ihnen entfernen.

Menü Projekt

Mit dem Menü-Befehl VERWEISE... können Sie Bibliotheken und Anwendungen auswählen, deren Funktionen und Objekte Sie in Ihrem Projekt verwenden wollen. Es wird dann ein Verweis auf die Objektbibliothek dieser Anwendung angelegt. Nach Auswahl eines Eintrags aus dieser Liste stehen Ihnen alle Funktionen und Objekte der Bibliothek oder Anwendung in Ihrem aktuellen Projekt zur Verfügung.

**Menü-Befehl
VERWEISE...**

2 Workshop: Die Entwicklungsumgebung

Abbildung 2.7:
Menü Ansicht

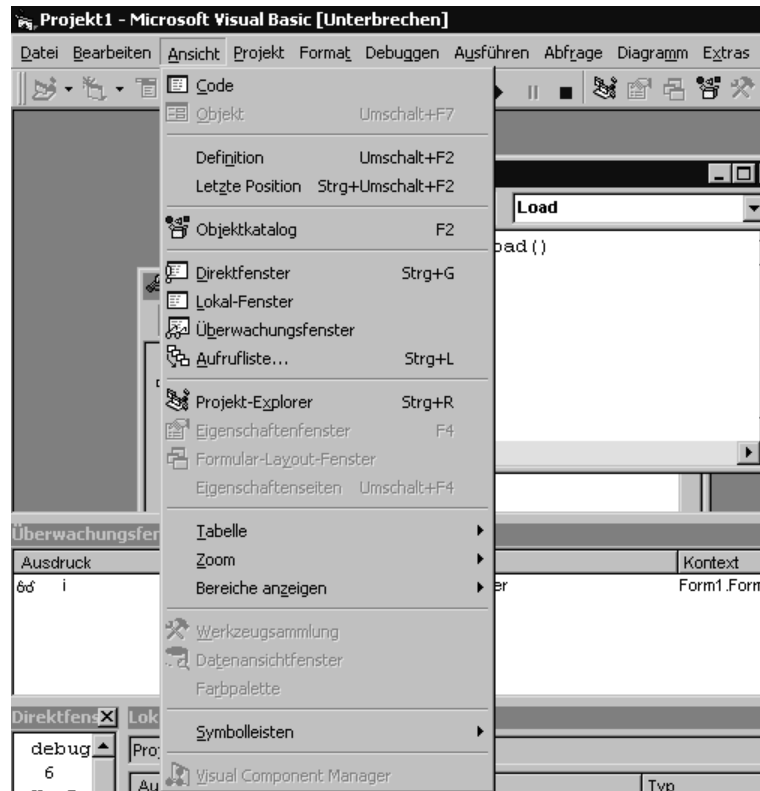
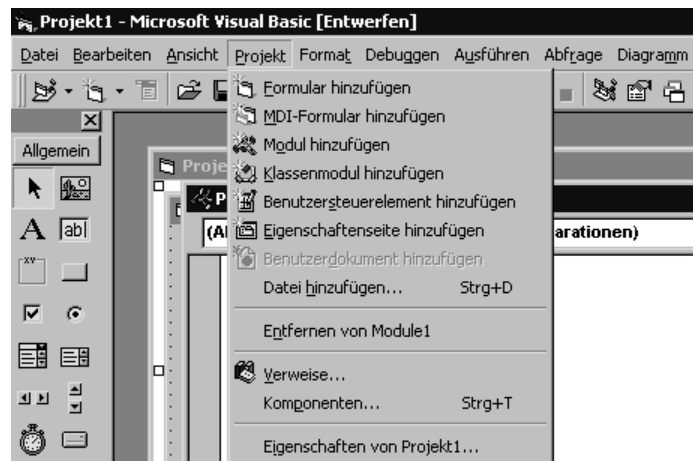


Abbildung 2.8:
Menü Projekt



Menü-Befehl Der Menü-Befehl **KOMPONENTEN...** ermöglicht es Ihnen schließlich, weitere Steuerelemente und Objekte in Ihr Projekt aufzunehmen.

Die wichtigsten Menüs

Der Menü-Befehl EIGENSCHAFTEN VON PROJEKT1... öffnet einen Dialog, in welchem alle relevanten, projektbezogenen Eigenschaften eingestellt werden können.

**Menü-Befehl
EIGENSCHAFTEN VON
PROJEKT1...**

2.2.6 Menü Format

Mit dem Menü FORMAT (Abbildung 2.9) erreichen Sie eine ganze Reihe von Befehlen, die es Ihnen erlauben, die Anordnung der Steuerelemente zu verändern und sie auszurichten.

Menü Format

Dies bezieht sich vor allem auf Gruppen von Steuerelementen. Ein einzelnes Steuerelement lässt sich mit den Menü-Befehlen lediglich am Raster ausrichten.

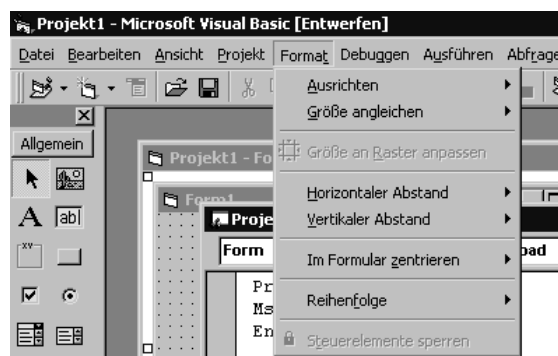


Abbildung 2.9:
Menü Format

2.2.7 Menü Debuggen

Das Menü DEBUGGEN (Abbildung 2.10) dient dem Überwachen und Testen einer Anwendung. Es beinhaltet verschiedene Menüpunkte zur Prozessablaufsteuerung, die in der Regel erst nach einer Unterbrechung des Programmablaufs wirksam werden.

Menü Debuggen



Abbildung 2.10:
Menü Debuggen



2.2.8 Übung: Programm Debuggen

Laden Sie das der Übung zugehörige Projekt in die Entwicklungsumgebung. Starten Sie eine Debug-Sitzung. Setzen Sie einen Haltepunkt. Arbeiten Sie im Einzelschritt-Modus und benutzen Sie die Funktionen um Variablen zu betrachten.

Lösung

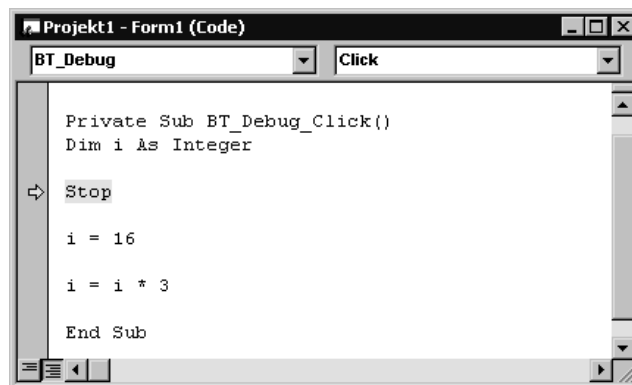
Projekt starten Nachdem Sie das Projekt geladen haben, können Sie es direkt starten. Benutzen Sie hierfür die Funktionstaste **[F5]**. Sie sehen die Oberfläche aus Abbildung 2.11.

Abbildung 2.11:
Das Programm wurde gestartet



Nachdem Sie die Schaltfläche *Starten* ausgewählt haben, öffnet sich ein Code-Fenster (Abbildung 2.12).

Abbildung 2.12:
Programmlauf wurde durch
Stop-Anweisung unterbrochen



Programmlauf unterbrechen Wenn Sie ein Programm debuggen möchten, so müssen Sie den Programmlauf unterbrechen, denn nur dann haben Sie die Möglichkeit auf den Programm-
lauf Einfluss zu nehmen bzw. einen aktuellen Status der Programmzustände zu erfahren.

Haltepunkt setzen In Abbildung 2.12 wurde der Programmlauf durch eine *Stop*-Anweisung unterbrochen. Eine weitere Möglichkeit ist das Setzen eines Haltepunktes. Hierzu müssen Sie den Eingabecursor in die Zeile stellen, in welcher Sie anhalten möchten. Dann drücken Sie die Taste **[F9]** oder wählen den Menüpunkt **DEBUGGEN-> HALTEPUNKT EIN/AUS**.

Die wichtigsten Menüs

Sie erkennen den Haltepunkt im Code-Fenster am roten Punkt links neben der Programmzeile (Abbildung 2.13).

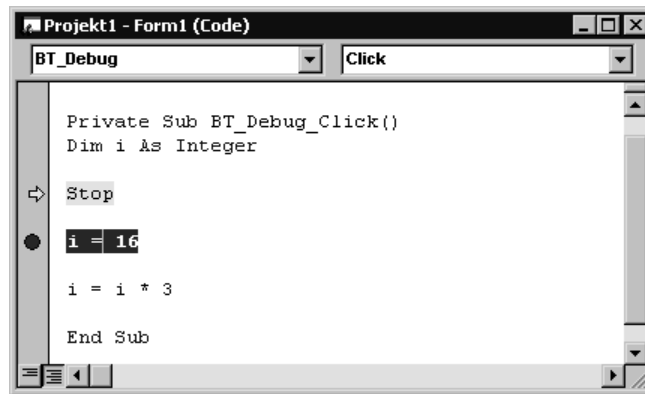


Abbildung 2.13:
Haltepunkt wurde
gesetzt

Durch Betätigen der Taste **[F5]** wird der Programmlauf fortgesetzt. Der nächste Stop ist auf dem gesetzten Haltepunkt.

Sie können im Haltemodus jede Anweisung im Code einzeln ausführen und das Ergebnis dann sofort analysieren. Sehr hilfreich sind hier das *Direktfenster*, das *Lokalfenster* und das *Überwachungsfenster* (Abbildung 2.14).

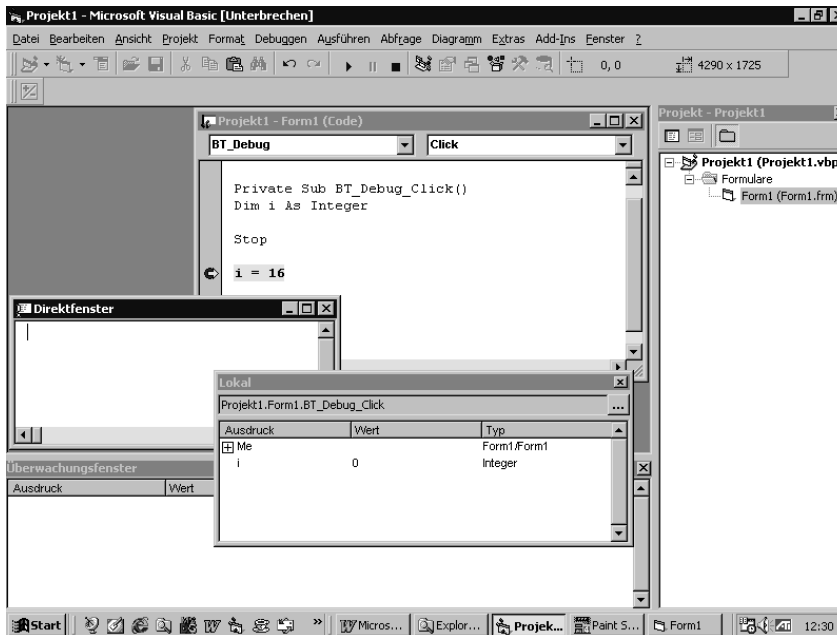
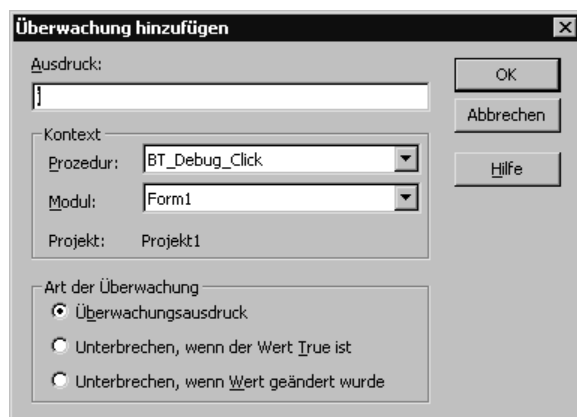


Abbildung 2.14:
Direktfenster,
Lokalfenster und
Überwachungs-
fenster

2 Workshop: Die Entwicklungsumgebung

- das Lokalfenster** Im Lokalfenster werden alle Variablen der aktuellen Prozedur mit ihren Werten angezeigt. Es wird nur dann aktualisiert, wenn der Programmablauf unterbrochen wurde.
- das Überwachungsfenster** Gleiches gilt für das Überwachungsfenster. Im Gegensatz zum Lokalfenster werden im Überwachungsfenster allerdings nur diejenigen Variablen angezeigt, die explizit von Ihnen zur Überwachung ausgewählt wurden. Dies kann durch das Menü DEBUGGEN->ÜBERWACHUNG HINZUFÜGEN erfolgen.
- das Direktfenster** Das Direktfenster schließlich erlaubt Ihnen die Eingabe von Programmanweisungen und deren Ausführung durch Betätigen der Taste [ENTER]. Außerdem können mit der Anweisung *Debug.Print* Ausgaben im Direktfenster angezeigt werden.
- Überwachung hinzufügen** Um eine Variable zu überwachen, muss Visual Basic über das Menü DEBUGGEN->ÜBERWACHUNG HINZUFÜGEN... dazu veranlasst werden. Setzen Sie die Eingabemarke auf eine Variable. Hat die Variable nur einen Buchstaben, wie im Beispiel, so können Sie die Eingabemarke direkt vor oder nach den Variablennamen stellen. Rufen Sie über den Menü-Befehl den Dialog *Überwachung Hinzufügen* (Abbildung 2.15) auf und bestätigen Sie ihn mit *Ok*.

Abbildung 2.15:
Dialog
Überwachung
hinzufügen

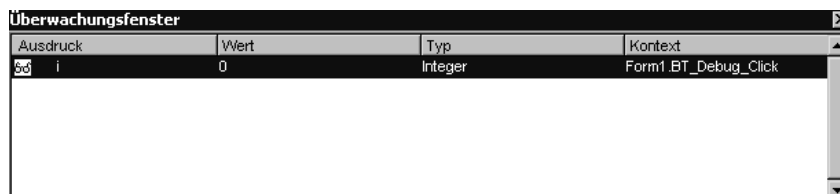


Das Überwachungsfenster zeigt jetzt die Variable an. Sie hat den Wert 0 (Abbildung 2.16). Die Programmzeile weist der Variablen aber den Wert 16 zu.



Das Programm wird durch einen Haltepunkt gestoppt, bevor die Programmzeile, auf der sich der Haltepunkt befindet, ausgeführt wird.

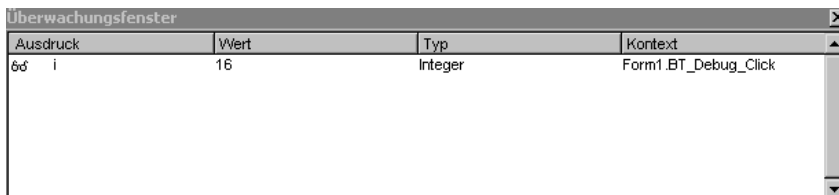
Abbildung 2.16:
Wert der Variablen
auf dem
Haltepunkt



Die wichtigsten Menüs

Um das Programm im Einzelschrittmodus weiterzuführen, benutzen Sie die Funktionstaste **F8** oder den Menü-Befehl **DEBUGGEN->EINZELSCHRITT**. In Abbildung 2.17 können Sie sehen, dass sich der Wert der Variablen *i* im Überwachungsfenster geändert hat.

Einzelschrittmodus



Ausdruck	Wert	Typ	Kontext
i	16	Integer	Form1.BT_Debug_Click

Abbildung 2.17: Wert der Variablen nach Ausführen der Programmzeile im Einzelschrittmodus

Sie können jetzt das Programm durch mehrfaches Betätigen der Funktionstaste **F8** oder durch einmaliges Betätigen der Funktionstaste **F5** zu Ende laufen lassen.

Sie haben jetzt die Grundtechniken des Debuggens mit Visual Basic kennen gelernt. Der Debugger von Visual Basic ist aber ein weitaus mächtigeres Werkzeug, als bisher gezeigt wurde. Es lohnt sich auf jeden Fall, hierzu die Online-Hilfe von Visual Basic zusätzlich zu konsultieren.



2.2.9 Menü Ausführen

Im Menü **AUSFÜHREN** (Abbildung 2.18) sind alle Befehle zusammengefasst, die benötigt werden, um ein Programm in der Entwicklungsumgebung zu starten.

Menü Ausführen

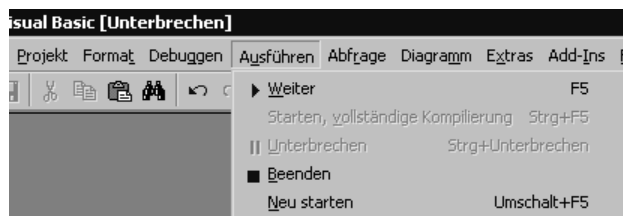


Abbildung 2.18: das Menü Ausführen

2.2.10 Menü Extras

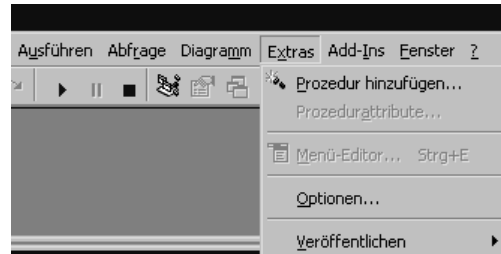
Das Menü **EXTRAS** (Abbildung 2.19) beinhaltet unter anderem den *Menü-Editor*. Ein geöffnetes und aktives Formularfenster kann mit Hilfe dieses Editors mit einem Menü versehen werden.

Menü Extras

Ein weitaus wichtigerer Punkt im Menü **EXTRAS** ist jedoch das Menü **OPTIONEN**. Dieser Menü-Befehl öffnet den Dialog *Optionen*, über welchen globale Einstellungen der Entwicklungsumgebung vorgenommen werden können.

Visual Basic Optionen

Abbildung 2.19:
Menü Extras



2.2.11 Menü Add-Ins

Menü Add-Ins Neben den von anderen Windows-Programmen her bekannten Menüpunkten *Fenster* und *?* gibt es noch den Menüpunkt *Add-Ins*.

Hierin sind alle installierten und freigeschalteten Add-Ins aufgelistet. Der *Add-In-Manager* in diesem Menü regelt diese Auflistung.

Add-Ins sind eigenständige Programme, die in die Entwicklungsumgebung auch nachträglich integriert werden können. Die Anzahl der mitgelieferten Add-Ins hängt von Ihrer Edition ab. Weitere Add-Ins können von anderen Herstellern zugekauft und in die Visual Basic-Entwicklungsumgebung integriert werden.

2.3 Symbolleisten

Symbolleisten Die Symbolleisten stellen ein *Mittelding* zwischen Menü-Befehlen und Tastaturkürzeln dar. Sie ermöglichen Ihnen, einen Menü-Befehl mit einem Mausklick auf ein Symbol auszuführen. Diese Symbole sind, nach Themenbereichen geordnet, in so genannten Leisten untergebracht. Sie befinden sich unterhalb der Menüleiste.

Sie haben auch die Möglichkeit, den Inhalt der Symbolleisten anzupassen oder gar eigene Symbolleisten zu entwerfen. Hierzu dient der Menü-Befehl ANSICHT>SYMBOLLEISTEN->ANPASSEN... .

2.4 Das Eigenschaftfenster

das Eigenschaftfenster Im *Eigenschaftfenster* (Abbildung 2.20) werden während der Programmentwicklung die Eigenschaften von Objekten (Formulare, Klassen, Steuerelemente) angezeigt. Sie können dort auch verändert werden. Das *Eigenschaftfenster* zeigt immer die Eigenschaften des aktuell markierten Objekts an.

Falls das Eigenschaftfenster nicht sichtbar ist, kann es über das Menü ANSICHT->EIGENSCHAFTENFENSTER, über die Taste **F4** oder über die Symbolleiste aktiviert werden.

Das Eigenschaftfenster

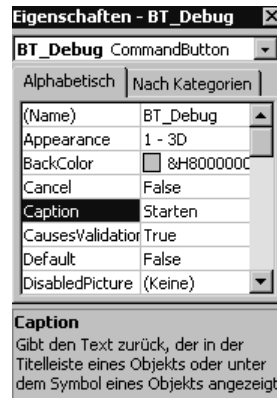


Abbildung 2.20:
Die Eigenschaften
der Schaltfläche
BT_Debug im
Eigenschaften-
fenster

Das Eigenschaftfenster besteht aus den Teilen *Objekt-Listbox* und der *Eigenschaften-Liste*.

Objekt-Listbox

In der *Objekt-Listbox*, die sich direkt unterhalb des Fenstertitels befindet, sind alle Objekte des aktuell aktiven Moduls eingetragen. Falls nur ein Objekt markiert ist, wird der Name des markierten Objekts in der *Textbox* angezeigt. Mit der *DropDown*-Liste kann, ohne das Eigenschaftfenster zu verlassen, auf jedes andere Objekt des aktiven Moduls gewechselt werden.

**alle Objekte des
aktiven Moduls**

Eigenschaftenliste

Die *Eigenschaftenliste* ist eine Tabelle mit zwei Spalten. Die linke Spalte enthält die Eigenschaften des markierten Objekts. Die rechte Spalte enthält die aktuellen Werte dieser Eigenschaften.

**Eigenschaften und
Werte**



Abbildung 2.21:
Eigenschaften nach
Kategorien
geordnet

Über die Registerkarten der Liste können die Eigenschaften alphabetisch oder, wie in Abbildung 2.21 sichtbar, nach Kategorien geordnet angezeigt werden.

**nach Kategorien
geordnet**



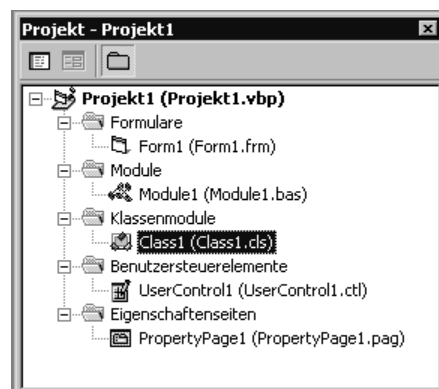
Das Setzen von Eigenschaften ist in Visual Basic der halbe Weg zum fertigen Programm. Unterschätzen Sie nicht die Macht der Eigenschaften. Viele Zeilen Code meiner Programme zeigten sich im Nachhinein als überflüssig, nachdem ich die Eigenschaften der Objekte besser definierte.

2.5 Der Projekt-Explorer

**alle Dateien
des/der
Projekts/Projekte**

Der *Projekt-Explorer* liefert eine Liste aller im Projekt oder der Projektgruppe verwendeten Module. Abbildung 2.22 zeigt dabei nur eine kleine Auswahl der möglichen Projektdateien. Ob normale *Formulare* oder *MDI-Formen*, *Module*, *Klassen*, *Steuerelementdateien* oder *Eigenschaftenseiten*, im Projektfenster sind alle Projektdateien aufgelistet.

Abbildung 2.22:
Alle Dateien eines
Projekts werden im
Projekt-Explorer
verwaltet



Vom *Projektfenster* aus kann das *Formularfenster* oder das *Code-Fenster* eines Moduls geöffnet werden. Es muss lediglich das entsprechende Modul in der Liste markiert werden. Jetzt wird entweder das Symbol *Code anzeigen* angewendet, um das zugehörige *Code-Fenster* zu öffnen, oder es wird das Symbol *Objekt anzeigen* angewendet, um das zugehörige *Formularfenster* zu öffnen.

einfach anklicken

Wird ein Modul im Projektfenster doppelt angeklickt, so wird für Formulare automatisch das *Formularfenster*, für Module automatisch das *Code-Fenster* geöffnet.

2.5.1 Kontextmenüs des Projekt-Explorers

Kontextmenüs

Wird über dem Projektfenster die rechte Maustaste gedrückt, so öffnet sich ein Kontextmenü (Abbildung 2.23).

In diesem Menü (Abbildung 2.23) können dem Projekt weitere Projekt-Module hinzugefügt werden.

Der Projekt-Explorer

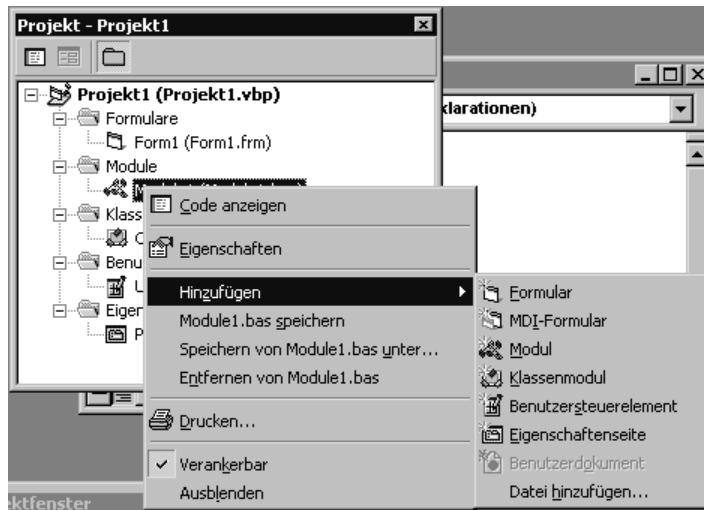


Abbildung 2.23:
Das Kontextmenü
des Projekt-
Explorers

Falls Sie beim Betätigen der rechten Maustaste nicht die Menüs aus Abbildung 2.23 gesehen haben, so befanden Sie sich wahrscheinlich mit dem Mauszeiger über dem Eintrag *Projekt* des Projekt-Explorers. In diesem Fall zeigt der Projekt-Explorer nämlich das Projekt-Kontextmenü aus Abbildung 2.24.

Projekt- Kontextmenü



Abbildung 2.24:
Das Projekt-
Kontextmenü des
Projekt-Explorers

Wenn Sie die beiden Menüs miteinander vergleichen, werden Sie feststellen, dass die Menü-Befehle, die sich um das Speichern und Entfernen von Projektmodulen drehen, jetzt auf das ganze Projekt beziehen, statt wie vorher nur auf ein einzelnes Modul.

Projekt speichern

Zudem können Sie den *Paket- und Weitergabeassistent* aufrufen, sofern Sie Ihre Software ausliefern möchten. Des Weiteren ist es möglich, für eine Projektgruppe das *Start-Projekt* festzulegen. Das Startprojekt ist jenes, welches bei Eingabe der Taste F5 gestartet wird. Diese Auswahl ist allerdings nur dann sinnvoll, wenn tatsächlich mehr als ein Projekt geladen ist.

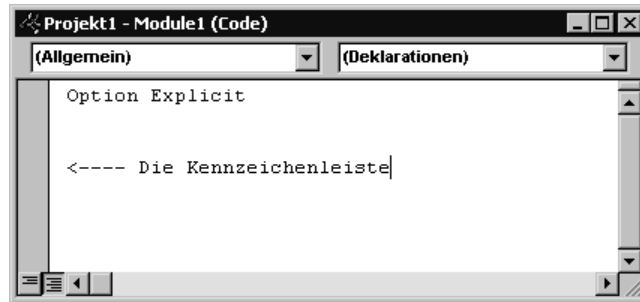
Paket- und Weiter- gabeassistent

2.6 Das Code-Fenster

Code-Fenster = Editor

Das *Code-Fenster*, auch Editor genannt, präsentiert sich mit einigen nützlichen Eigenschaften. Zuerst wird Ihnen wohl der senkrechte Strich am linken Rand auffallen. Dies ist die so genannte Kennzeichenleiste (Abbildung 2.25).

Abbildung 2.25:
Das Code-Fenster mit Kennzeichenleiste



Kennzeichenleiste

In der Kennzeichenleiste können Lesezeichen und Haltepunkte angezeigt und gesetzt werden. Im Abschnitt über das Debuggen konnten wir dies bereits sehen.

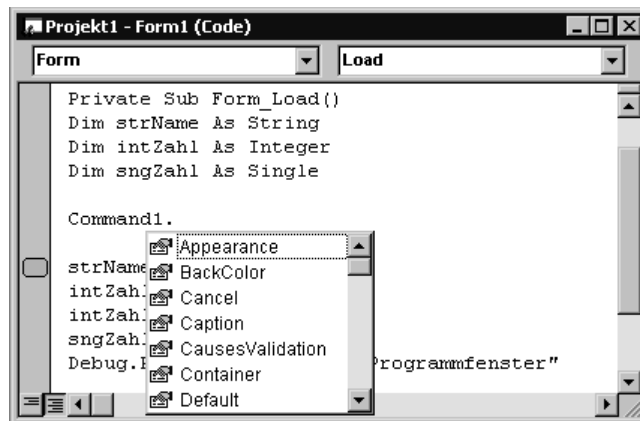
Das Code-Fenster hat aber einige weitere Funktionen, die das Programmieren sehr erleichtern:

Automatische Anweisungsergänzung

Anweisungsergänzung

Wenn Sie im Code auf ein Objekt mit einer Methode oder Eigenschaft Bezug nehmen, so wird Ihnen nach Eingabe des Punktes eine *Drop-Down-Liste* angezeigt, die alle Methoden und Eigenschaften des Objekts enthält (Abbildung 2.26).

Abbildung 2.26:
Automatische Anweisungsergänzung im Code-Fenster



Die Werkzeugsammlung und das Formularfenster

Mit den Pfeiltasten oder mit der Maus wählen Sie Ihr gewünschtes Element. Wenn Sie dann die Leertaste oder die Tabulatortaste drücken, wird das gewählte Element automatisch rechts hinter dem Punkt eingetragen.

Dies erleichtert Ihnen die Suche nach den verfügbaren Elementen des Objekts und verhindert darüber hinaus Schreibfehler. Die *Automatische Anweisungsergänzung* kann im Dialog *Optionen* unter *Editor->Elemente automatisch auflisten* aktiviert werden.

keine Schreibfehler

QuickInfo

Dies ist ein weiteres Hilfsmittel, um Fehleingaben zu verhindern. Wenn im Dialog *Optionen* in *Editor ->Automatische QuickInfo* aktiviert ist, werden Sie beim Eingeben eines Funktionsaufrufs automatisch von den zugehörigen Parametern in Kenntnis gesetzt (Abbildung 2.27).

QuickInfo

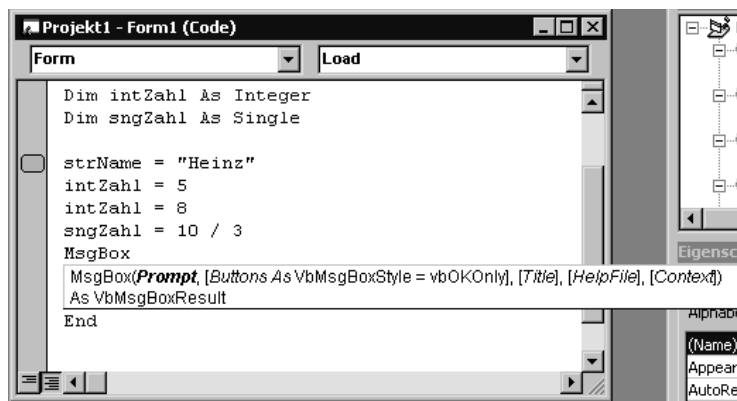


Abbildung 2.27: Automatische QuickInfo zeigt die Parameter der Methode MsgBox

Es wird ein Infofenster unter Ihrer aktuellen Zeile aufgeblendet, welches die Funktionssyntax anzeigt. Das erste Argument wird fett dargestellt. Wenn zu diesem ein Wert eingegeben wurde, wechselt die Markierung auf das zweite Argument, und so weiter. Es kann Ihnen somit nicht mehr passieren, dass Sie zu wenig Argumente oder einen falschen Typ an eine Funktion übergeben.

2.7 Die Werkzeugsammlung und das Formularfenster

Zwei weitere, nicht wegzudenkende Elemente der Visual Basic-Entwicklungs Oberfläche sind die *Werkzeugsammlung*, auch Toolbox genannt, und das *Formularfenster*.

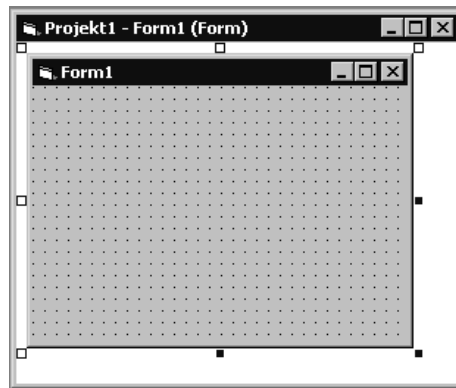
Mit diesen zwei Werkzeugen wird im Wesentlichen die Oberfläche eines Programms erstellt. Ich habe dabei bewusst das Wort »programmiert« nicht verwendet, denn die Erstellung einer Oberfläche ist in Visual Basic nichts anderes als das Zusammenstellen von mehreren Objekten im Formularfenster.

Oberflächen-design

2 Workshop: Die Entwicklungsumgebung

Ein Formularfenster kann für jedes Programm-Objekt mit grafischer Oberfläche (beispielsweise eine Form) geöffnet werden (Abbildung 2.28).

Abbildung 2.28:
Das Objekt Form1
im Formularfenster
(Designer)

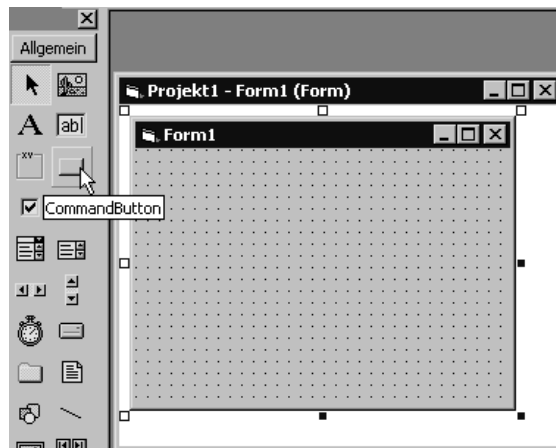


2.7.1 Die Werkzeugsammlung – Objekte hinzufügen

Die Werkzeugsammlung ist das einzige Instrument, mit dem man neue Objekte in ein Formularfenster aufnimmt. Dabei können Sie Objekte anhand ihrer grafischen Darstellung in der Werkzeugsammlung oder anhand der in Abbildung 2.29 dargestellten QuickInfo identifizieren.

Um die QuickInfo zu erhalten, müssen Sie lediglich mit dem Mauszeiger eine kurze Zeitspanne über der entsprechenden Grafik verharren.

Abbildung 2.29:
Ein Objekt in der
Werkzeugsammlung
identifizieren



Um eine Schaltfläche auf der Form zu platzieren, wird diese, wie in Abbildung 2.29 angedeutet, in der Werkzeugsammlung doppelt angeklickt. Daraufhin wird das neue Objekt, in diesem Fall eine Schaltfläche, auf dem Formular (Objekt *Form1*) mittig platziert.

Die Werkzeugsammlung und das Formularfenster

Um ein Objekt im Formularfenster an eine andere Position zu bewegen, wird es mit der Maus bei gedrückter Taste gezogen. Die Größe der Objekte kann ebenfalls per Drag&Drop geändert werden. Alle weiteren Eigenschaften werden über das Eigenschaftenfenster verändert.

**Position und
Größe von
Objekten ändern**

