

Die Entwicklung Ihrer ersten Benutzeroberfläche

3

In diesem Kapitel

- ▶ Die Komponenten der Benutzeroberfläche kennenlernen
- ▶ Die Benutzeroberfläche zeichnen
- ▶ Die Eigenschaften der Benutzeroberfläche ändern

Der erste Schritt, um ein Visual Basic-Programm zu erstellen, besteht in der Erstellung der Benutzeroberfläche. Dabei müssen Sie die folgenden Schritte ausführen:

- ✓ Zeichnen Sie die Benutzeroberfläche
- ✓ Definieren Sie die Eigenschaften der Benutzeroberfläche
- ✓ Schreiben Sie BASIC-Code

Bevor Sie also Ihr erstes Programm in Visual Basic schreiben können, müssen Sie wissen, wie man eine Benutzeroberfläche erstellt.

Gemeinsame Komponenten einer Benutzeroberfläche

Trotz ihrer Unterschiede sehen graphische Benutzeroberflächen ähnlich aus, weil sie aus ähnlichen Komponenten bestehen. Jede graphische Benutzeroberfläche zeigt Informationen in einem *Fenster* an. Ein Fenster kann den ganzen Bildschirm oder nur einen Teil davon bedecken. Zwei oder mehr Fenster können gleichzeitig auf dem Bildschirm angezeigt werden, wobei sie sich entweder wie Spielkarten überlappen oder wie Kacheln Seite an Seite liegen können. In Visual Basic werden Fenster als *Formulare* bezeichnet.

Wenn Sie ein neues Formular erstellen, ist es vollkommen leer. Um Ihr Formular nützlich zu machen, müssen Sie so genannte *Objekte* in das Formular einfügen. Ein Objekt kann ein Bild, ein Rahmen, eine Schaltfläche oder ein Kontrollkästchen sein. Der Benutzer kommuniziert mit Ihrem Programm, indem er die Objekte anklickt, Daten in sie eingibt oder in anderer Form manipuliert.

Auf sich allein gestellt, machen Objekte absolut nichts. Um sie zu veranlassen, etwas zu tun, müssen Sie BASIC-Code schreiben (worüber Sie in Kapitel 4 mehr erfahren).



Eine kurze Geschichte der Benutzeroberflächen

In der Frühzeit der Computer (damals in den Fünfzigern) musste man einen Computer öffnen und einige Drähte umstecken, wenn man ihn dazu bringen wollte, etwas Nützliches zu tun. Man musste wissen, wie ein Computer programmiert wird, und man musste wissen, wie man Drähte verbindet, ohne sich dabei durch einen elektrischen Schlag umzubringen.

Um einen Computer in den Sechzigern zu benutzen, hat man ihn mit einem Stapel Lochkarten gefüttert. Wenn der Computer auf Befehle reagierte, druckte er etwas auf Papier aus, ein Vorgang, der Stunden oder gar Tage dauern konnte. Deshalb war die Benutzung eines Computers immer noch langsam, mühsam und langweilig.

Die Siebziger verbanden eine Tastatur mit einem Fernsehapparat und nannten das ganze Ding Computer-Terminal. Zum ersten Mal konnte man einen Befehl direkt in den Computer eintippen und sofort vom Computer eine Antwort erhalten. Diese Möglichkeit hätte die Bedienung der Computer eigentlich erleichtern und zur Abwechslung auch mal etwas angenehmer machen sollen.

Doch diese ersten primitiven Benutzeroberflächen bestanden nur aus einem leeren Bildschirm und einem blinkenden Punkt, der als Cursor bezeichnet wurde. Um den Computer dazu zu bringen, irgend etwas zu tun, musste man die passenden Befehle eintippen können. Leider verweigerte der Computer die Zusammenarbeit, wenn man nicht die richtigen Befehle kannte und nicht wusste, auf welche Tasten man drücken sollte, so dass man sich ziemlich dumm vorkam. Und wieder war die Benutzung eines Computers langsam, mühsam und langweilig.

In einem verzweifelten Versuch, die Bedienung von Computern zu vereinfachen, erfanden Programmierer bald danach etwas, das als *Graphical User Interface* oder *GUI* (wird so ausgesprochen wie es geschrieben wird) bezeichnet wurde. Ein GUI besteht im wesentlichen aus Menüs und Symbolen, die der Benutzer mit der Maus anklicken kann, um Befehle auszuführen.

Der Apple-Macintosh hatte das erste populäre GUI, aber Microsoft zog schnell nach und führte unter dem Namen *Microsoft Windows* ein sehr erfolgreiches GUI ein. Leider kann auch die Benutzung eines GUIs auf einem Computer langsam, mühsam und total langweilig sein. Jetzt wissen Sie, dass die Computerindustrie die Schuld daran hat. Das löst zwar die Probleme nicht, behebt aber vielleicht für einen Moment Ihre Sorgen.

Objekte mit Visual Basic erstellen

Objekte in einem Formular werden mit der Werkzeugsammlung erstellt, die auf der linken Seite des Bildschirms angezeigt wird. Diese Werkzeugsammlung enthält kleine Zeichnungen, welche die verschiedenen Objekte repräsentieren, die Sie auf ein Formular zeichnen können (siehe Abbildung 3.1).



Zwei Methoden, um eine Benutzeroberfläche zu erstellen

Visual Basic bietet Ihnen zwei Methoden, um eine Benutzeroberfläche zu erstellen:

- ✓ Sie können die Benutzeroberfläche mit dem Visual Basic-Anwendungsassistenten automatisch erstellen.
- ✓ Sie können die Benutzeroberfläche manuell neu erstellen.

Welche Methode ist besser? Wenn Sie ein Programm für Windows 95/98/NT mit den Standardmenüs *Datei*, *Bearbeiten*, *Fenster* und *Hilfe* erstellen wollen, lassen Sie Visual Basic die Arbeit für Sie erledigen.

Aber wenn Sie nur ein einfaches Programm ohne Pulldown-Menüs erstellen wollen, geht es manuell leichter.

Welchen Weg Sie auch wählen, Sie können Ihre Benutzeroberfläche später jederzeit modifizieren.



Abbildung 3.1: Die Visual Basic-Werkzeugsammlung.

Um ein Objekt auf einem Formular zu erstellen, gehen Sie immer folgendermaßen vor:

1. **Klicken Sie auf das Objekt in der Werkzeugsammlung, das Sie zeichnen wollen.**
2. **Gehen Sie mit dem Mauszeiger an die Stelle des Formulars, an der Sie das Objekt zeichnen wollen.**
3. **Drücken und halten Sie die linke Maustaste nieder, und ziehen Sie die Maus. Damit zeichnen Sie das Objekt auf dem Formular.**



Um ein Objekt schnell auf einem Formular zu zeichnen, doppelklicken Sie auf das Symbol des Objekts in der Werkzeugsammlung. Visual Basic erstellt dann das betreffende Objekt automatisch auf dem Formular.

Der Entwurf einer Benutzeroberfläche hat viel mit Zeichnen zu tun. In Teil II erfahren Sie mehr über bestimmte Objekte, ihr Einsatzgebiet und ihre Funktion. Hier und jetzt ist es wichtig, dass Sie zwei Dinge verstehen: Alle Ihre Programme brauchen eine Benutzeroberfläche; und Visual Basic stellt Ihnen die Komponenten zur Verfügung, mit denen Sie ein Programm schnell und einfach zusammensetzen können.

Die erste Benutzeroberfläche manuell neu erstellen



Falls Sie die folgenden 14 Schritte nicht manuell eingeben wollen, enthält die beiliegende CD die Datei HELLO1.VBP im Verzeichnis *Kapitel 3*, die Sie öffnen und studieren können.

Damit Sie jetzt gleich mit Visual Basic vertraut werden, finden Sie hier einige Schritte, mit denen Sie eine lebensechte Benutzeroberfläche erstellen können:

1. **Starten Sie Visual Basic unter Windows 95/98/NT, falls Sie das noch nicht getan haben. Falls Visual Basic bereits läuft, wählen Sie den Menübefehl DATEI/NEUES PROJEKT.**

Visual Basic zeigt das Dialogfeld *Neues Projekt* an und fragt Sie, welche Art von Programm Sie erstellen wollen.

2. **Klicken Sie auf *Standard-EXE* und dann auf *Öffnen*.**

Visual Basic zeigt ein leeres Formular mit dem Titel *Form1* an. (Wenn Sie den Visual Basic-Anwendungsassistenten aktivieren wollen, um die Benutzeroberfläche mit seiner Hilfe automatisch zu erstellen, klicken Sie an dieser Stelle auf sein Symbol, aber nicht jetzt. Am Ende dieses Kapitels erfahren Sie, wie Sie den Visual Basic-Anwendungsassistenten benutzen können.)

3. **Fahren Sie mit dem Mauszeiger auf das kleine Quadrat (einen so genannten *Ziehpunkt*) in der rechten unteren Ecke des Formulars, so dass er die Form eines doppelköpfigen Pfeils annimmt. Drücken und halten Sie die linke Maustaste nieder, und ziehen Sie die Maus, um das Formular zu vergrößern.**

4. **Wählen Sie den Menübefehl ANSICHT/WERKZEUGSAMMLUNG, um die Werkzeugsammlung anzuzeigen.**

(Überspringen Sie diesen Schritt, falls die Werkzeugsammlung bereits auf dem Bildschirm sichtbar ist.)

5. **Klicken Sie in der Werkzeugsammlung auf das *Befehlsschaltfläche*-Symbol (*QuickInfo CommandButton*; siehe Abbildung 3.2).**

3 ► Die Entwicklung Ihrer ersten Benutzeroberfläche

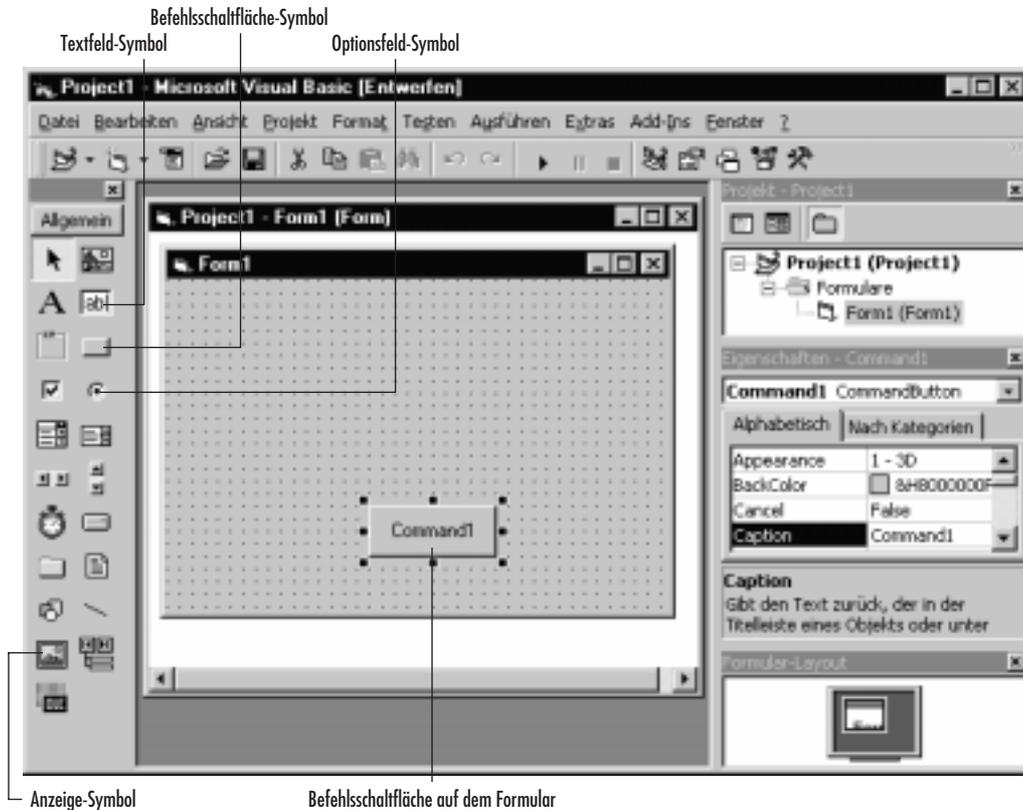


Abbildung 3.2: Das Befehlsschaltfläche-Symbol in der Werkzeugsammlung und eine Schaltfläche auf dem Formular.

6. Ziehen Sie die Maus über das Formular, so dass die Schaltfläche aussieht, wie in Abbildung 3.2 auf dem Formular abgebildet.

7. Klicken Sie in der Werkzeugsammlung auf das *Optionsfeld*-Symbol, und zeichnen Sie ein Optionsfeld.

Wiederholen Sie diesen Vorgang noch zweimal, um insgesamt drei Optionsfelder zu erstellen (siehe Abbildung 3.3).

8. Klicken Sie in der Werkzeugsammlung auf das *Anzeige*-Symbol, und zeichnen Sie einen Bilderrahmen.

Wiederholen Sie diesen Vorgang noch zweimal, um insgesamt drei Bilderrahmen zu erstellen (siehe Abbildung 3.4).

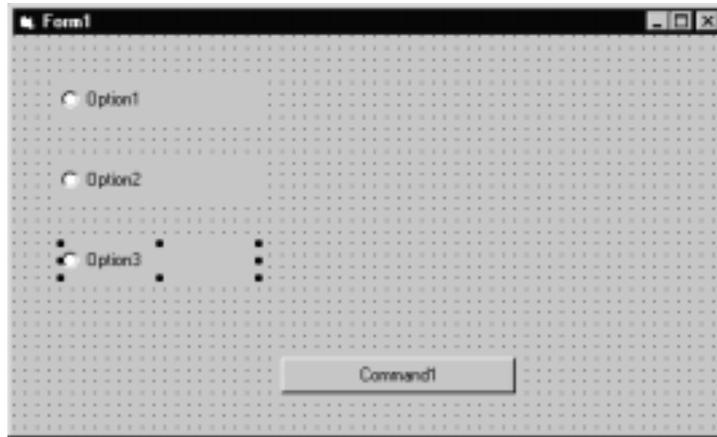


Abbildung 3.3: Drei Optionsfelder auf dem Formular.

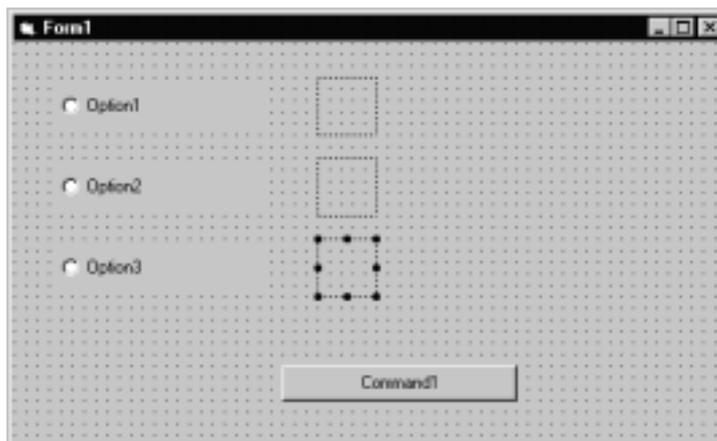


Abbildung 3.4: Drei Bilderrahmen auf dem Formular.

9. Klicken Sie in der Werkzeugsammlung auf das *Textfeld*-Symbol, und zeichnen Sie ein Textfeld.

Wiederholen Sie diesen Vorgang noch zweimal, um insgesamt drei Textfelder zu erstellen (siehe Abbildung 3.5).

10. Wählen Sie den Menübefehl DATEI/<FORMULAR> SPEICHERN, oder drücken Sie auf **Strg + **S**.**

Das Dialogfeld *Datei speichern unter* erscheint und fragt Sie, wie Sie Ihre Datei nennen wollen. (Sie können vor dem Speichern im Kombinationsfeld *Speichern in* den Ordner wählen, in dem Sie Ihr Visual Basic-Projekt und Ihre Formulare speichern wollen.)

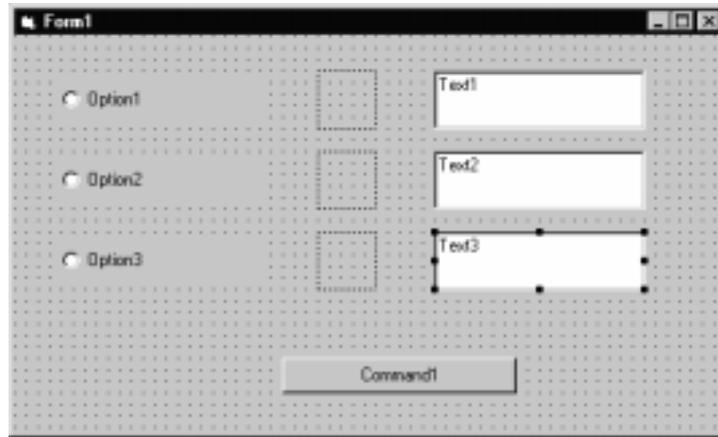


Abbildung 3.5: Drei Textfelder auf dem Formular.

11. Tippen Sie *HELLO* ein, und klicken Sie auf *Speichern*.

Damit wird Ihr Formular in einer Datei namens HELLO.FRM gespeichert.

12. Wählen Sie den Menübefehl *Datei/Projekt speichern*.

Das Dialogfeld *Datei speichern unter* erscheint und fragt Sie, wie Sie Ihr Projekt nennen wollen.

13. Tippen Sie *HELLO* ein, und klicken Sie auf *Speichern*.

Damit wird Ihr Visual Basic-Projekt in einer Datei namens HELLO.VBP gespeichert.

14. Wählen Sie den Menübefehl *Datei/Beenden*, oder drücken Sie auf **[Alt] + [Q], wenn Sie Visual Basic beenden wollen.**

Glückwunsch! Sie haben gerade eine generische Visual Basic-Benutzeroberfläche erstellt. Wenn sie nicht besonders eindrucksvoll aussieht, liegt das daran, dass die Benutzeroberfläche nicht an Ihr Programm angepasst wurde.

Um eine Visual Basic-Benutzeroberfläche anzupassen, müssen Sie für jedes Objekt *Eigenschaften* (engl. *properties*) definieren.

Definieren Sie die Eigenschaften Ihrer ersten Benutzeroberfläche

Das Zeichnen der Benutzeroberfläche ist der erste Schritt beim Schreiben eines Visual Basic-Programms. Der zweite Schritt besteht darin, die Eigenschaften für jedes Objekt Ihrer Benutzeroberfläche zu definieren.

Ist es wirklich notwendig, die Eigenschaften zu definieren? Ja und nein. Visual Basic weist den Eigenschaften aller Objekte Ihrer Benutzeroberfläche automatisch bestimmte Standardwerte zu. Damit sehen Ihre Programme jedoch ziemlich häßlich aus. Wenn Sie Ihre Benutzeroberfläche attraktiver gestalten wollen, müssen Sie Eigenschaften für Ihre Objekte definieren.



Obwohl jedes Objekt zehn bis dreißig veränderbare Eigenschaften besitzt, brauchen Sie nicht jede einzelne davon zu ändern. Meistens ändern Sie nur ein oder zwei Eigenschaften jedes Objekts.

Was Eigenschaften bewirken

Ehe Sie die Eigenschaften eines Objekts ändern, möchten Sie wahrscheinlich wissen, wozu Eigenschaften überhaupt gut sind. Im wesentlichen definieren Eigenschaften das Erscheinungsbild eines Objekts auf dem Bildschirm und den Namen des Objekts.



Der Name eines Objekts dient nur Ihrer Bequemlichkeit. Visual Basic gibt Objekten automatisch langweilige Namen wie *Text1* oder *Command1*. Wenn Sie BASIC-Code schreiben, machen Sie sich die Arbeit viel leichter, wenn Sie jedem Objekt einen aussagekräftigen Namen geben, um die Objekte in Ihrem Programm zu identifizieren. (In Teil IV lernen Sie, BASIC-Code zu schreiben.)

Das Erscheinungsbild eines Objekts umfasst seine Größe, Farbe und Position auf dem Bildschirm. Es lässt Ihre Benutzeroberfläche gut aussehen.

Werte von Eigenschaften ändern

Sie können den Wert von Eigenschaften zu zwei verschiedenen Zeiten ändern:

- ✓ Zur Entwurfszeit
- ✓ Zur Laufzeit

Die *Entwurfszeit* ist der Zeitraum, in dem Sie Ihre Benutzeroberfläche entwerfen, also der Zeitraum ehe Sie Ihr Programm tatsächlich ausführen.



Meistens werden Sie die Werte der Eigenschaften eines Objekts zur Entwurfszeit ändern. Wenn Sie den Namen eines Objekts ändern wollen, müssen Sie dies zur Entwicklungszeit tun.

Die *Laufzeit* ist die Zeit, in der Ihr Programm ausgeführt wird und die Werte der Eigenschaften eines Objekts mit BASIC-Code ohne Ihren Eingriff geändert werden. Ehe Ihr Programm die Eigenschaften eines Objekts ändern kann, müssen Sie dem Programm natürlich mit BASIC-Code genau vorschreiben, welche Eigenschaften eines Objekts geändert werden sollen.

Durch die Änderung einer Eigenschaft zur Laufzeit können Sie Animationen ablaufen lassen oder Meldungen auf dem Bildschirm anzeigen. Dazu zählen z.B. Fehlermeldungen, Programm-Statusanzeigen oder Meldungen, die den Benutzer daran erinnern, Ihr Programm zu bezahlen.

Wie Sie Eigenschaftswerte zur Entwurfszeit ändern

Jedes Objekt hat zehn oder mehr Eigenschaften (siehe Abbildung 3.6). Um die Eigenschaften eines Objekts zu ändern, stellen Sie sicher, dass ein Formular angezeigt wird, und führen Sie dann folgende Schritte aus:

1. **Klicken Sie auf das Objekt, dessen Eigenschaften Sie ändern wollen.**
2. **Öffnen Sie das Eigenschaftenfenster, und klicken Sie auf die Eigenschaft, die Sie ändern wollen.**

Wenn das Eigenschaftenfenster nicht sichtbar ist, drücken Sie auf **F4**, um das Fenster zu öffnen.

3. **Geben Sie den neuen Wert dieser Eigenschaft ein.**



Abbildung 3.6: Das Eigenschaftenfenster mit mehreren Eigenschaften des Textfelds Text3.

Einfach, nicht wahr? Wenn Sie die Eigenschaftswerte für ein oder mehrere Objekte ändern müssen, zeigt dieses Buch eine Tabelle, die folgendermaßen aussieht:

Objekt	Eigenschaft	Wert
Form	Name	frmHello
	Caption	Hallo Welt!

Die Tabelle sagt Ihnen Folgendes:

1. **Klicken Sie auf das Formular.**
2. **Klicken Sie im Eigenschaftenfenster auf die Eigenschaft *Name*.**
3. **Ändern Sie den Wert, indem Sie *frmHello* eintippen.**
4. **Klicken Sie im Eigenschaftenfenster auf die Eigenschaft *Caption*.**
5. **Ändern Sie den Wert, indem Sie *Hallo Welt!* eingeben.**

Testen Sie Ihr neu erworbenes Wissen

1. **Was sind die zwei allgemeinen und in fast jeder Benutzeroberfläche enthaltenen Komponenten?**
 - a) Die leicht bedienbare Benutzeroberfläche und ein 500-Seiten-Handbuch, das erklärt, wie einfach die Benutzeroberfläche zu benutzen ist.
 - b) Menüs, die niemand versteht, und eine Maus, die niemand benutzen kann.
 - c) Nutzlose Symbole, deren Funktion unklar ist, und Text, der nichts aussagt.
 - d) Formulare und Objekte.
2. **Wie können Sie Eigenschaftswerte ändern?**
 - a) Gar nicht. Die Eigenschaften müssen selbst den Wunsch haben, sich zu ändern.
 - b) Durch permanente Drohungen und Einschüchterungen.
 - c) Indem Sie das Eigenschaftenfenster benutzen, nachdem Sie die Benutzeroberfläche entworfen haben (zur so genannten *Entwurfszeit*), oder indem Sie BASIC-Code schreiben, um die Werte zu ändern, wenn das Programm ausgeführt wird (zur so genannten *Laufzeit*).
 - d) Indem Sie einen Magneten an eine Seite Ihres Bildschirms halten und beobachten, wie sich die Bilder verzerren und Ihr Monitor kaputtgeht.

Wie Sie die Eigenschaften Ihrer ersten Benutzeroberfläche definieren



Falls Sie die folgenden 17 Schritte nicht manuell nachvollziehen wollen, können Sie die Datei HELLO2.VBP aus dem Ordner *Kapitel 3* von der beiliegenden CD laden und studieren.

Sie können die Eigenschaften Ihrer Benutzeroberfläche definieren, indem Sie folgende Schritte ausführen:

1. Starten Sie Visual Basic unter Windows 95/98/NT. Falls Visual Basic bereits läuft, wählen Sie den Menübefehl *Datei/Neu*.

Visual Basic zeigt das Dialogfeld *Neues Projekt* an.

2. Klicken Sie auf die Registerkarte *Aktuell*.
3. Klicken Sie auf die Datei *HELLO.VBP* und dann auf *Öffnen*.

Visual Basic zeigt das Formular *HELLO.FRM* auf dem Bildschirm an.

4. Klicken Sie auf das Optionsfeld *Option1*, um es zu markieren. Die Markierung besteht aus kleinen schwarzen Rechtecken an den Ecken und Rändern des Objekts.
5. Klicken Sie im Eigenschaftfenster auf die Eigenschaft *Name*, und tippen Sie *optGrin* ein. Klicken Sie auf die Eigenschaft *Caption*, und geben Sie *Ich bin glücklich!* ein.
6. Klicken Sie auf das Feld *Image1* (am oberen Rand des Formulars).
7. Klicken Sie im Eigenschaftfenster auf die Eigenschaft *Name*, und tippen Sie *imgGrin* ein.
8. Klicken Sie im Eigenschaftfenster auf die Eigenschaft *Picture*. Klicken Sie auf die drei Punkte (...).

Visual Basic zeigt das Dialogfeld *Bild laden* an (siehe Abbildung 3.7).

9. Öffnen Sie den Ordner *Icons*. (Es könnte sein, dass Sie sich erst den Ordner *Graphics* suchen müssen.) Öffnen Sie dann den Ordner *Misc* im Ordner *Icons*.
10. Doppelklicken Sie auf das Symbol *Face03.ico*.

Visual Basic zeigt ein wirklich glückliches Gesicht in dem Bilderrahmen an.

11. Klicken Sie auf die Eigenschaft *Visible*. Klicken Sie dann auf den Pfeil in dem Wertfeld, und wählen Sie *False*.
12. Klicken Sie auf das Feld *Text1*.
13. Klicken Sie auf die Eigenschaft *BorderStyle*. Klicken Sie auf den Pfeil in dem Wertfeld, und wählen Sie *0 - Kein*.
14. Klicken Sie auf die Eigenschaft *Name*, und tippen Sie *txtGrin* ein.
15. Doppelklicken Sie auf die Eigenschaft *Text*. Drücken Sie auf die Rück-Taste, um der Eigenschaft *Text* eine Leerzeile zuzuweisen.
16. Ändern Sie die Eigenschaften für die anderen Objekte gemäß Tabelle 3.1.
17. Wählen Sie den Menübefehl *Datei/Speichern*, um Ihre Änderungen zu speichern.

Glückwunsch! Sie haben gerade alle notwendigen Eigenschaften für Ihre erste Benutzeroberfläche definiert.

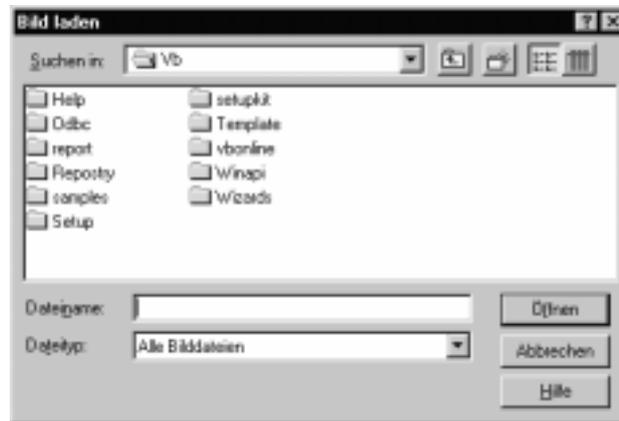


Abbildung 3.7: Das Dialogfeld Bild laden.

Objekt	Eigenschaft	Wert
Form	Name	frmHello
	Caption	Hallo Welt!
Option2	Name	optSmile
	Caption	Mir geht es gut.
Option3	Name	optFrown
	Caption	Ich bin traurig.
Image2	Name	imgSmile
	Picture	Face02
	Visible	False
Image3	Name	ImgFrown
	Picture	Face01
	Visible	False
Text2	Name	txtSmile
	BorderStyle	0 - Kein
	Text	(Leer)
Text3	Name	txtFrown
	BorderStyle	0 - Kein
	Text	(Leer)
Command1	Name	cmdExit
	Caption	Ende

Tabelle 3.1: Eigenschaften, die Sie ändern müssen, um den Entwurf Ihrer Benutzeroberfläche fertigzustellen.

Die Benutzeroberfläche mit Visual Basic automatisch erstellen

Wenn Sie die Schritte zur Erstellung der Benutzeroberfläche manuell nachvollzogen haben, haben Sie einen Eindruck davon bekommen, wie mühsam und zeitaufwendig dieser Prozess sein kann. Weil Computer doch eigentlich dazu da sind, Zeit einzusparen (damit Sie mehr Computerspiele spielen können), enthält Visual Basic den so genannten *Visual Basic-Anwendungsassistenten*.

Der Visual Basic-Anwendungsassistent erstellt das Gerüst eines kompletten Programms mit Pull-down-Menüs, Symbolleisten und Dialogfeldern. Wenn Sie ein Programm erstellen wollen, das der Benutzeroberfläche von Microsoft Word, Excel oder PowerPoint entsprechen soll, können Sie Zeit sparen, indem Sie die Benutzeroberfläche mit dem Visual Basic-Anwendungsassistenten erstellen.

Um zu sehen, wie der Visual Basic-Anwendungsassistent funktioniert, führen Sie die folgenden Schritte aus:

1. **Starten Sie Visual Basic unter Windows 95/98/NT. Falls Visual Basic bereits läuft, wählen Sie den Menübefehl DATEI/NEUES PROJEKT.**

Visual Basic öffnet das Dialogfeld *Neues Projekt*.

2. **Klicken Sie auf das Symbol *VB-Anwendungsassistent* und dann auf *Öffnen*.**

Das Einführungsfenster des Anwendungsassistenten wird angezeigt (siehe Abbildung 3.8). Dieses Fenster fordert Sie auf, ein Profil einzugeben. *Profile* sind Gerüste mit Standardeinstellungen, mit denen Sie den VB-Anwendungsassistenten anpassen können. Wählen Sie im Moment die vorgegebene Einstellung (*Keine*).



Abbildung 3.8: Das Einführungsfenster des Anwendungsassistenten.

3. Klicken Sie auf **Weiter**.

Das Fenster *Schnittstellentyp* des Anwendungsassistenten wird angezeigt (siehe Abbildung 3.9).



Abbildung 3.9: Das Fenster Schnittstellentyp des Anwendungsassistenten.

4. Markieren Sie das Optionsfeld **Multiple Document Interface (MDI)**, und klicken Sie dann auf **Weiter**.

Das Fenster *Menüs* des Anwendungsassistenten wird angezeigt (siehe Abbildung 3.10). (Falls Sie wollen, wählen Sie in Schritt 4 ein anderes Optionsfeld, um herauszufinden, wie die Optionen *Single Document Interface (SDI)* oder *In der Art des Explorers* aussehen.)



Abbildung 3.10: Das Fenster Menüs des Anwendungsassistenten.

5. Klicken Sie auf Weiter, um die Standardauswahl zu übernehmen.

Das Fenster *Symbolleiste anpassen* des Anwendungsassistenten wird angezeigt (siehe Abbildung 3.11). Per Drag & Drop können Sie aus den Symbolen eigene Symbolleisten zusammenstellen.



Abbildung 3.11: Das Fenster Symbolleiste anpassen des Anwendungsassistenten.

6. Klicken Sie auf Weiter, um die Standardauswahl zu übernehmen.

Das Fenster *Ressourcen* des Anwendungsassistenten wird angezeigt (siehe Abbildung 3.12). Mithilfe von Ressourcendateien können Sie fremdsprachige Versionen Ihres Programms erstellen. Statt die Ausdrücke der anderen Sprachen direkt in den Programmtext (beispielsweise in den Menüs und in Dialogfeldtiteln) eingeben zu müssen, können Sie diese Informationen mithilfe von Ressourcendateien separat speichern und benötigen für jede Sprache nur eine entsprechende Ressourcendatei.

7. Klicken Sie auf Weiter, um die Standardauswahl zu übernehmen.

Das Fenster *Internet-Verbindung* des Anwendungsassistenten wird angezeigt (siehe Abbildung 3.13). An dieser Stelle können Sie die Home-Page für Ihre Anwendung spezifizieren.

8. Klicken Sie auf Weiter, um die Standardauswahl zu übernehmen.

Das Fenster *Standardformulare* des Anwendungsassistenten wird angezeigt (siehe Abbildung 3.14).



Abbildung 3.12: Das Fenster Ressourcen des Anwendungsassistenten.



Abbildung 3.13: Das Fenster Internet-Verbindung des Anwendungsassistenten.

9. Aktivieren Sie das Kontrollkästchen Info-Feld, und klicken Sie dann auf Weiter.

Das Fenster *Datenzugriffsformulare* des Anwendungsassistenten wird angezeigt (siehe Abbildung 3.15). An dieser Stelle können Sie spezifizieren, mit welcher Datenbank Sie in Ihrem Programm arbeiten wollen.

3 ► Die Entwicklung Ihrer ersten Benutzeroberfläche



Abbildung 3.14: Das Fenster Standardformulare des Anwendungsassistenten.



Abbildung 3.15: Das Fenster Datenzugriffsformulare des Anwendungsassistenten.

10. Klicken Sie auf Weiter, um die Standardauswahl zu übernehmen.

Das Fenster *Fertig!* des Anwendungsassistenten wird angezeigt (siehe Abbildung 3.16).

11. Klicken Sie auf *Fertig*.

Visual Basic erstellt die Anwendung und zeigt die zugehörigen Fenster an.

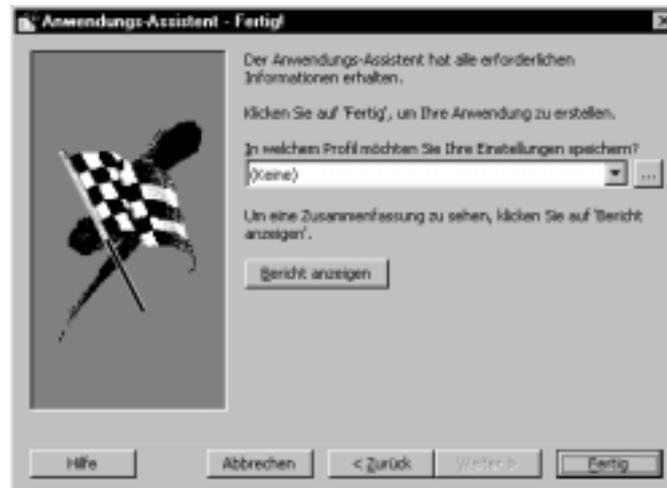


Abbildung 3.16: Das Fenster Fertig! des Anwendungsassistenten.

12. Drücken Sie auf F5, um das Programmgerüst auszuführen, das der Visual Basic-Anwendungsassistent für Sie erstellt hat.

Das Programm wird in einem separaten Fenster angezeigt (siehe Abbildung 3.17).



Abbildung 3.17: Das Programmgerüst, das von dem Anwendungsassistenten erstellt wurde.

13. Wählen Sie den Menübefehl ?/INFO.

Das Dialogfeld *Info Projekt1* wird angezeigt.

14. Klicken Sie auf *OK*.

15. Klicken Sie auf ein beliebiges Symbol in der Symbolleiste, oder wählen Sie einen beliebigen Menübefehl, um zu sehen, was passiert.

16. Wählen Sie den Menübefehl DATEI/BEENDEN, wenn Sie sich mit dem Programmgerüst vertraut gemacht haben.

17. Wählen Sie den Menübefehl DATEI/NEUES PROJEKT.

Ein Dialogfeld wird angezeigt, das Sie fragt, ob Sie Ihr neu erstelltes Projekt speichern wollen.

18. Klicken Sie auf *Nein*.

Wie Sie sehen, können Sie mit dem Visual Basic-Anwendungsassistenten schnell eine neue Benutzeroberfläche erstellen. Trotzdem müssen Sie die Benutzeroberfläche noch an Ihre Erfordernisse anpassen und BASIC-Code schreiben, um mit dem Programm eine nützliche Funktion zu erfüllen. Der Visual Basic-Anwendungsassistent hilft Ihnen nur dabei, schneller in die Gänge zu kommen.