Einleitung

Herzlich willkommen bei diesem Kurzlehrgang! Wenn ihr Delphi neu erworben und das erste Mal gestartet habt und dann nicht mehr weiter wisst, oder auch wenn ihr überlegt, ob Programmieren mit Delphi überhaupt was für euch ist, dann seid ihr hier genau richtig.

Eins jedoch gleich vorweg: Von Programmieren solltet ihr bereits ein klein wenig Ahnung haben, am besten in Pascal. Denn eine Programmiersprache werde ich hier nicht erläutern. Die „besondere Denkweise“ von Programmierern sollte euch also etwas vertraut sein.

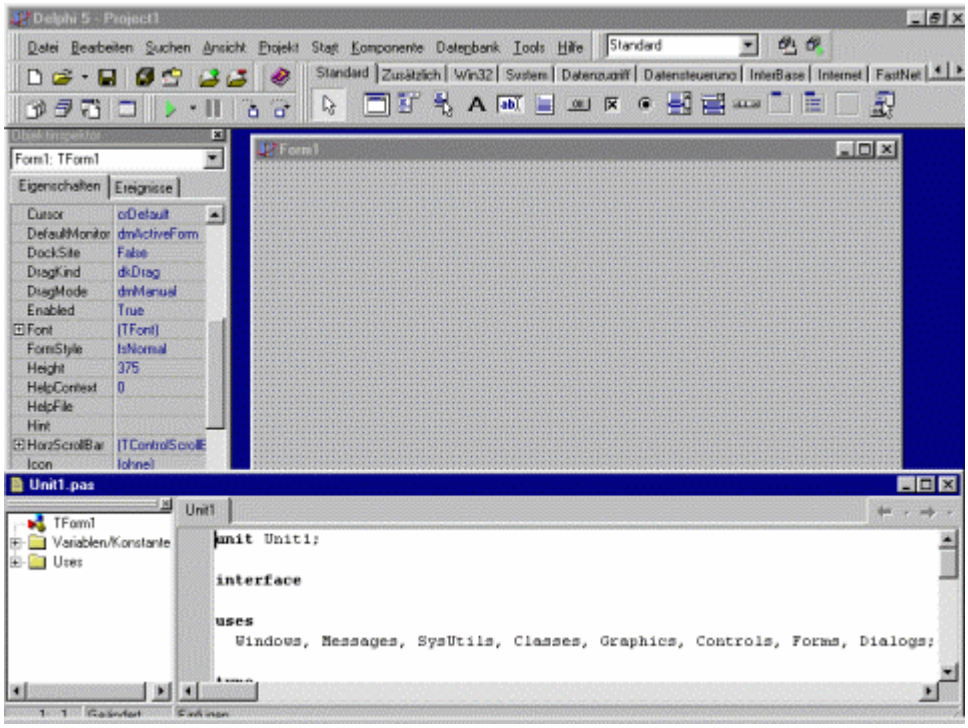
In diesem Zusammenhang kann gleich klargestellt werden: Delphi ist keine Programmiersprache, sondern eine visuelle Entwicklungsumgebung. Die dabei verwendetete Sprache heißt Object Pascal.

Aber jetzt zur Sache: Nach dem ersten Delphi-Start ist ein Neuling erst einmal von der gigantischen Anzahl von Menüpunkten und Symbolleisten erschlagen. Wo klicke ich denn nun als erstes, um eine kleine Windows-Anwendung zu erstellen? Das soll in diesem Tutorial geklärt werden.

Anmerkung: Die Screenshots stammen von Delphi 5 Professional. Das Erklärte funktioniert jedoch in allen Delphi Versionen.

Oberfläche

Hier habt Delphi also gestartet und nun in etwa folgenden Bildschirmaufbau vor euch. Die einzelnen Fenster können natürlich frei angeordnet werden. Die abgebildete Anordnung ist jedoch empfehlenswert.



Folgende Fenster sind zu sehen:

- Am oberen Fensterrand befindet sich das „eigentliche“ Delphi: die Symbolleisten und Menüs.
- Am linken Rand ist der sog. „Objektinspektor“ zu sehen. Er hat am oberen Rand eine Combobox (oder Dropdownliste), in der sich alle Komponenten aufgelistet sind, die sich im gerade aktuellen Fenster (=Form) der zu entwickelnden Anwendung befinden. Jetzt beim Start wird hier Form1: TForm1 angezeigt. Alle darunter aufgeführten Eigenschaften beziehen sich also auf das rechts angezeigte Fenster. Im

Objektinspektor lassen sich nun Eigenschaftswerte manuell verändern - ohne eine Zeile zu programmieren.

÷! Am unteren Fensterrand ist das Fenster zu sehen, in dem sich der eigentliche Quellcode befindet. Der linke Teil davon, der sog. Explorer, ist in den ersten Delphi-Versionen noch nicht verfügbar. Er dient dazu, schneller im Code navigieren zu können.

÷! Und schließlich in der Mitte das bereits erwähnte Hauptfenster unserer neuen Anwendung.

Projekte

Bevor wir nun loslegen, unser neues Programm zu erstellen, müssen zuerst ein paar Formalitäten klären, damit klar wird, wie Delphi ein solches "Projekt" verwaltet.

Wie unter Windows üblich erzeugt Delphi durch Kompilieren ausführbare Dateien mit der Endung EXE. Solche Programme bestehen in der Regel aus mehreren Fenstern. Wenn wir unserem Programm ein zweites Fenster hinzufügen, was wir nachher auch tun werden, erzeugt Delphi gleichzeitig eine neue Quellcode-„Unit“. Jedem Fenster („Form“) wird also eine Unit zugeordnet. Es gibt auch Units ohne Fenster, jedoch keine Fenster ohne Unit. Irgendwie muss sich Delphi merken, welche Units am Ende zusammen in eine Anwendung gehören. Das wird in der Projektdatei gespeichert. Wir können auch jetzt sehen: Am oberen Fensterrand steht "Project1", Delphi hat also ein neues, leeres Projekt erstellt, das bereits ein Fenster (Form1) mit zugehöriger Unit (Unit1) enthält.

Klicken wir nun auf den Button mit den drei Disketten („Alles speichern“) werden wir nacheinander nach zwei Dateinamen gefragt: Der erste ist der Name, den Unit1 bekommen soll. Nennen wir sie Hauptfenster. Die Endung .pas wird automatisch hinzugefügt. Gleichzeitig erstellt Delphi auch eine gleichnamige Datei mit der Endung .dfm, in der die Fenstereinstellungen von Form1 gespeichert werden. Der zweite Name ist für die Projektdatei, bezeichnet also das ganze Projekt und wird auch der Dateiname der später zu erzeugenden EXE. Nennen wir das Projekt Test. Es erhält die Endung .dpr (=Delphi Project).

Programme starten

Klicken wir nun einmal testweise auf den grünen Pfeil in der Menüleiste bzw. drücken die Taste F9.

Je nach Einstellung (Menü Tools/Umgebungsoptionen/Präferenzen/Compilerfortschritt anzeigen) erscheint nun ein kleines Dialogfenster, in dem ihr sehen könnt, was der Compiler gerade macht. Da keine Fehler auftreten (wer nichts macht, kann auch nichts falsch machen), wird unser Mini-Programm auch gleich gestartet: Ein leeres Fenster mit dem Titel "Form1".

Jetzt denkt ihr vielleicht, das war ja noch kein großes Kunststück. Für Delphi wahrlich nicht. Aber diejenigen, die sich mal an Borland Pascal 1.0 für Windows versucht haben und stundenlang tippen mussten, um solch ein leeres Fenster auf den Bildschirm zu bekommen, sind vielleicht beeindruckt. Aber wir machen weiter.

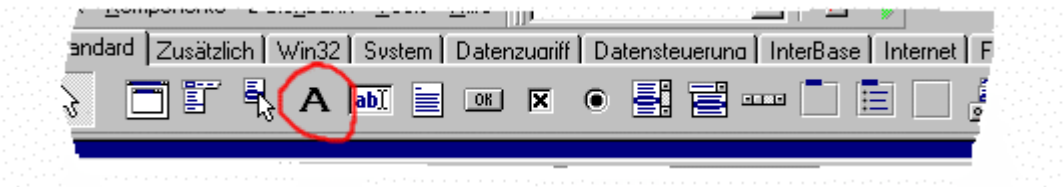
Komponenten (1)

Wir beenden also unser kleines Programm durch Klick auf das X-Symbol in der Titelleiste.

Als erstes ändern wir den Fenstertitel. Form1 ist schließlich nicht besonders aussagekräftig. Dazu verwenden wir den Objektinspektor. Hier sind alle Eigenschaften von Form1 aufgelistet, die wir bearbeiten können. Die Eigenschaft, die für den Fenstertitel zuständig ist, heißt „Caption“. Tragen wir hier nun etwas Sinnvolleres ein, z. B. Mein erstes Delphi-Programm. Während ihr tippt, könnt ihr beobachten, wie sich gleichzeitig der Titel im Fenster rechts tatsächlich ändert.

Nun wollen wir noch einen kleinen Text in dem Fenster platzieren.

Alles, was in einem Fenster angezeigt werden kann, bekommt ihr aus der Komponentenpalette. Da Borland dem Paket bereits knapp 100 verschiedene Komponenten beilegt, sind diese auf verschiedene Registerseiten verteilt, damit Übersicht nicht verloren geht. Wir begnügen uns mit der Seite „Standard“.



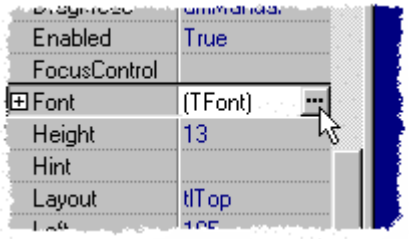
Und für Text ist hier die Komponente „Label“ zuständig. Klickt also darauf und anschließend auf die Stelle im Formular, wo der Text beginnen soll.

Komponenten (2)

Nun haben wir den Text „Label1“ im Fenster stehen. Aber wie vorhin beim Fenstertitel, kann auch die Eigenschaft Caption der Label-Komponente verändert werden.

Nun ist aber darauf zu achten, dass wir zwei Komponenten im Einsatz haben: das Fenster und den Text. Im Objektinspektor muss also die richtige Komponente in der Combobox angezeigt werden, ansonsten können wir hier die andere auswählen. Schneller geht das, wenn wir mit der Maus das Label bzw. eine leere Stelle von Form1 anklicken.

Damit wir uns nicht nur auf die Eigenschaft „Caption“ beschränken, wollen wir unseren Text etwas deutlicher hervorheben. Dazu hat die Label-Komponente die Eigenschaft „Font“. Hier gibt es am Ende der Zeile einen Button mit drei Punkten. Wenn wir darauf klicken, erscheint das bekannte Schriftart-Auswahlfenster.



Ereignisse

Der Text kann also nach Belieben formatiert werden.

Als nächstes wollen wir einen Button in unser Programm einbauen. Die Komponente „Button“ befindet sich drei Plätze rechts neben dem Label in der Standard-Komponentenleiste. Draufklicken und ihn irgendwo platzieren.

Wie wohl die Beschriftung geändert wird? Richtig, über die Eigenschaft Caption.

Nun liegt der Sinn eines Buttons darin, dass man auf ihn klickt und daraufhin etwas passiert. Wie können wir auf solche Aktionen des Programmbenutzers reagieren?

Vielleicht habt ihr schon bemerkt, dass es im Objektinspektor nicht nur die Seite „Eigenschaften“, sondern auch eine Seite „Ereignisse“ gibt. Dorthin wechseln wir jetzt. Der Button sollte dabei markiert sein, wir wollen schließlich seine Ereignisse festlegen und nicht die des Fensters oder des Labels.

Mit etwas Schulenglisch lässt sich leicht herausfinden, worauf die einzelnen Ereignisse reagieren. Ein Klick auf den Button wird natürlich ein „OnClick“-Ereignis auslösen.

Nun muss also etwas Programmiertes kommen, denn bisher haben wir ja noch keine Zeile Code getippt. Aber Delphi hilft uns noch etwas. Wir machen einen Doppelklick auf das leere Feld hinter „OnClick“. Daraufhin wechseln wir automatisch in das Quellcode-Fenster. Hier hat Delphi selbstständig eine leere Prozedur für das Ereignis angelegt. Wir müssen also nur noch tippen, was passieren soll.

Grundsätzlich sollten solche von Delphi angelegten Prozeduren nicht gelöscht werden. Falls man sich vertan hat und ein Ereignis doch nicht braucht, reicht es den selbstgetippten Code zwischen begin und end zu löschen und anschließend auf „Speichern“ zu klicken - schon entfernt Delphi die überflüssige Prozedur wieder.

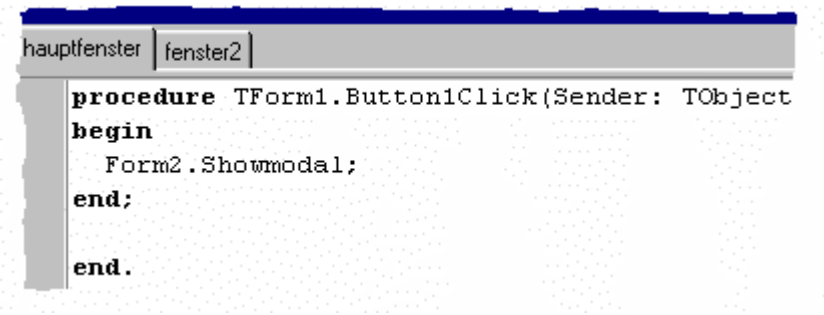
Ein zweites Fenster

Wie angekündigt wollen wir also nun ein zweites Fenster öffnen. Dazu müssen wir jedoch erst mal eins erzeugen. Klicken wir also auf das Menü Datei und wählen „Neues Formular“. Schon bekommen wir ein „Form2“ zu sehen zusammen mit einer neuen Unit. Diese speichern wir unter dem Namen fenster2.pas. In dem Fenster platzieren wir ein Label mit der Beschriftung „Ich bin das zweite Fenster“. Dann können wir es schließen und uns wieder der Unit Hauptfenster.pas zuwenden.

Jedes Fenster (allgemein: jede Komponente) hat einen eindeutigen Namen, das ist der Wert, der hinter der Eigenschaft "Name" angegeben ist. Dieser Wert kann natürlich verändert werden, um das Programm übersichtlicher zu machen. Der Name unseres zweiten Fensters ist nach wie vor „Form2“. Diesen Namen müssen wir wissen, um das Fenster anzeigen zu können.

Wir tippen in unser OnClick-Ereignis also folgenden Text:

Form2.ShowModal;



Im Gegensatz zu „Show“ kann bei „ShowModal“ auf andere Fenster der Anwendung erst wieder zugegriffen werden, wenn das modale Fenster geschlossen wurde.

Unsere Unit Hauptfenster kann nun aber noch nichts mit dem Ausdruck "Form2" anfangen. Wir müssen ihr sagen, wo Form2 definiert ist. Wer sich in Pascal auskennt, weiß, dass eine andere Unit in der uses-Klausel eingebunden wird. In Delphi kann man das per

Klick machen: Menü Datei, Unit verwenden. Hier einfach Fenster2 (so haben wir die Unit2 ja gespeichert und in ihr ist das Form2 deklariert) auswählen - fertig.

Über F9 können wir unser Programm jetzt ausprobieren. Um eine ordentliche EXE-Datei zu erzeugen, kann der Menüpunkt "Projekt/Projekt xy compilieren" bzw. die Tastenkombination Strg+F9 verwendet werden.

Im Gegensatz zu Visual Basic benötigt eine Delphi-Anwendung keine Laufzeitbibliotheken, es genügt also, die fertige EXE-Datei weiterzugeben.

Ausblick

Das war jetzt natürlich ein ziemlich simples Beispiel für den Einstieg. Jede einzelne Komponente hat natürlich noch zahlreiche weitere Eigenschaften, die verändert werden können. Hier hilft am besten die Online-Hilfe: Einfach eine Eigenschaft im Objektinspektor markieren und F1 drücken.

Allgemeine Informationen über eine Komponente sind so ebenfalls erhältlich.

Viel Spaß beim Experimentieren!