Delphi leicht & verständlich

Autor: Hans-Georg Schumann (Neuauflage von **Delphi für Einsteiger**)
ISBN 978-87-91364-35-8, 4. Auflage: 2008-12

Cover: Sylvio Droigk, Lektorat: Joh.-Chr. Hanke
Published by KnowWare Vertrieb GmbH

Bestellung für Endverbraucher und für den Buchhandel
KnowWare-Vertrieb GmbH
Postfach 3920
D-49029 Osnabrück
Tel.: +49 (0)541 33145-20
Fax: +49 (0)541 33145-33
E-Mail: bestellung@knowware.de
Web: www.knowware.de

Worum es geht
Hinter KnowWare steht der Gedanke, Wissen leicht verständlich und preisgünstig zu vermitteln.

Wo sind die Hefte erhältlich
Neue Hefte sind im Allgemeinen in verschiedenen Buchhandlungen vor Ort, im Bahnhofs-Buchhandel und in Internet-Buchhandlungen vorrätig.

Alle beim Verlag vorrätigen Titel kannst du immer bestellen, und zwar:

Bestellungen:
- am einfachsten über unsere Webseite www.knowware.de
- oder mit dem Bestellformular am Ende eines KnowWare Heftes
- oder per Fax, Telefon sowie Email Details siehe oben

unter www.knowware.de findest du:

- Beschreibungen und Bilder aller Hefte.
- Bei unseren Heften stehen dir die ersten 15-20 Seiten pro Titel als kostenlose PDF-Datei zur Verfügung. So lässt sich jedes Heft online testen.
- Ausverkaufte Hefte: viele Titel sind als PDF-Datei - teilweise sogar kostenlos - herunterladbar
- Online-Bestellung
- Kostenloser Newsletter mit vielen Vorteilen und Informationen
- Interne Suchfunktion nach Schlagworten. Du findest schnell, was du suchst.

Inhaltsverzeichnis

Vorwort

Von Pascal nach Delphi

In diesem Kurs geht es ums Programmieren. Dabei gäbe es einiges an Sprachen, in denen man das Programmieren erlernen kann. Jede hat ihre Vor- und Nachteile, und du wirst immer jemanden finden, der die eine Sprache ganz besonders lobt, während die übrigen bei ihm nicht allzu gut abschneiden.

Ich programmiere am liebsten in Delphi bzw. in Pascal. Also was denn nun? Pascal oder Delphi?

Am Anfang war da ein Schweizer Professor, der eine Programmiersprache erfand. Er gab ihr den Namen eines berühmten französischen Mathematikers: Pascal (Blaise Pascal 1623-1662).

Die Programmiersprache *Pascal* verbreitete sich schnell, vor allem an den Universitäten. Irgendwann kam eine kleine amerikanische Firma namens Borland auf die Idee, ein ganzes Programmiersystem auf der Basis der Sprache Pascal zu entwickeln. Und weil dieses System für die damalige Zeit (wo PC-Prozessoren noch mit rund 5 MHz getaktet wurden), recht rasant lief, wurde es unter dem Namen „Turbo Pascal“ weltbekannt.

Nach dem Siegeszug von Windows gab es anfangs auch ein *Turbo Pascal* für Windows, aus dem schließlich das heutige Delphi entstand, benannt nach der berühmten Orakelstätte des alten Griechenland.

Unter *Delphi* versteht man eine komplette Entwicklungsumgebung: Die Sprache ist ein stark weiterentwickeltes Pascal, genannt *Object Pascal*. Es gibt einen Editor, in dem du den Programmtext eintippen und bearbeiten kannst. Es gibt einen Compiler, der deinen Text so übersetzt, dass er für den Computer verständlich und als Programm ausführbar wird. Und es gibt eine Reihe von Hilfsprogrammen, die dich bei der Fehlersuche unterstützen – denn Fehler sind beim Programmieren alltäglich.

Weshalb sollte man ausgerechnet in Delphi bzw. Pascal programmieren? Für mich ein wesentlicher subjektiver Grund: Weil mir diese Sprache gefällt – so wie dir z. B. eine bestimmte Automarke.

Aber schauen wir uns kurz die „Konkurrenz“ an:

Sehr bekannt ist *Visual Basic*. Dieses Entwicklungssystem ist nicht allzu schwer zu erlernen und zu bedienen – ähnlich wie Delphi. Allerdings reicht die Leistungsfähigkeit nicht an die von Delphi heran.

C++ ist möglicherweise die „stärkste“ Sprache überhaupt, allerdings auch am schwierigsten zu erlernen.

Java wiederum ist leichter zu erlernen als C++, ist ihm jedoch in seiner Leistungsfähigkeit unterlegen.

Und Delphi?

Delphi bietet für mich den besten Kompromiss: Mit Delphi das Programmieren zu lernen ist etwas schwieriger als mit Visual Basic. Aber in Punkto Leistungsfähigkeit reicht es durchaus an Entwicklungssysteme für C++ heran und kann mit Java locker mithalten.

Was hält uns also ab, endlich unser System zu starten und mit dem Programmieren zu beginnen?

Du hast noch kein Delphi?

Möglicherweise bist du Schüler oder Student oder sonst wie „Lernender“. Dann gibt es bei vielen Software-Anbietern einen Sonderpreis für Delphi.

Es muss ja nicht unbedingt die neueste Version sein. Oft bekommt man eine ältere zum Schleuderpreis. (Die meisten meiner Programmbeispiele funktionieren mit neueren wie älteren Delphi-Versionen.)

Und auf der Homepage von Borland gibt es eine kostenlose 30-Tage-Test-Version von Delphi 7, die du dir hier herunterladen kannst:

http://www.borland.com/products/downloads/download_delphi.html

Allerdings kann das dauern, denn es handelt sich um immerhin rund 170 MB!

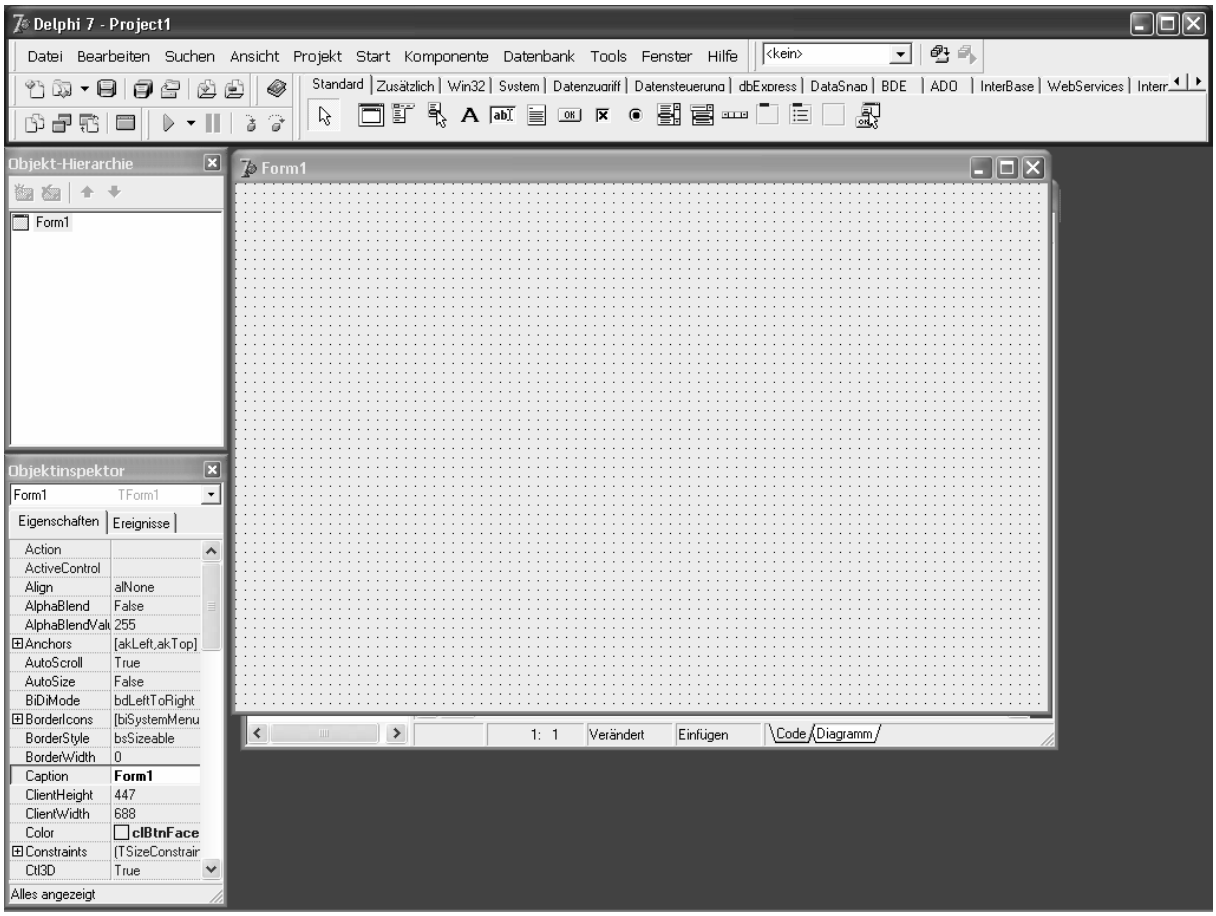
Viel Spaß und Erfolg mit Delphi wünscht

Hans-Georg Schumann

Auf zur Delphi-Tour!

Ich gehe davon aus, dass du weißt, wie man unter Windows ein Programm installiert und wie man es startet. Auch solltest du wissen, wie man bei Windowsprogrammen Menüs öffnet und Dialogfelder bedient, um z. B. Dateien zu speichern oder zu laden.

Nachdem Delphi in den Startlöchern steht, kann es losgehen – es erscheint dieses oder ein ähnliches Bild:



Ganz oben siehst du die Hauptleiste. Hier tummeln sich Menüeinträge und Symbole – wie bei anderen Windowsprogrammen auch.

Ein Formular

Darunter finden wir eine Art Arbeitsplatte wie in der Küche beim Zubereiten von Speisen; hier heißt diese Platte *Formular* oder *Formfenster*.

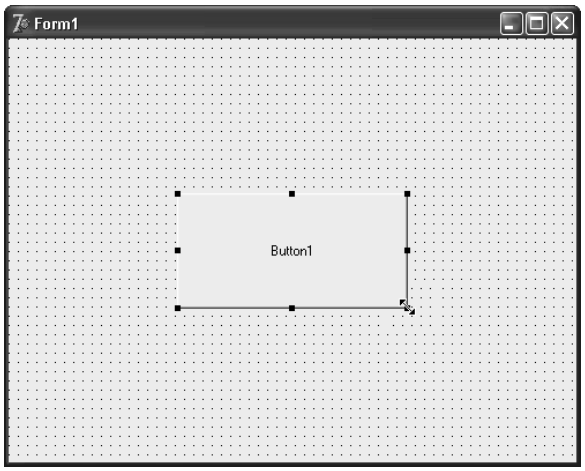
Links davon steht etwas von Objekt-Hierarchie und Objektinspektor. Mit einem dieser beiden Hilfsfenster befassen wir uns etwas später – das andere benötigen wir hier nicht unbedingt.

Das erste kleine Projekt

Jetzt aber wollen wir gleich unser erstes kleines Projekt erstellen.

- Suche in der Symbolleiste das Symbol für BUTTON und klicke darauf.

- Im Formfenster drückst du die linke Maustaste und ziehst damit einen Rahmen auf, der die Größe deines Button festlegt. Seine Lage und Maße lassen sich jederzeit ändern.



Statt vom Button spricht man im Deutschen auch von Schaltfläche, wie du aus deiner eigenen Erfahrung mit Windows weißt.

Nun haben wir eine Art Knopf (zum Drücken oder Klicken) in einem Formular – immerhin. Mich stören zunächst die Bezeichnungen: Statt „Form1“ und „Button1“ könnte man da doch etwas anderes hinsetzen?

Objekte

Und nun kommt der *Objektinspektor* zum Zug.

Wie kommt es zu diesem Namen? Unter einem *Objekt* versteht man normalerweise Dinge, die sich in unserer Umgebung befinden: Häuser, Autos, aber auch Menschen, Tiere, Pflanzen. Objekte haben Eigenschaften und Verhaltensweisen.

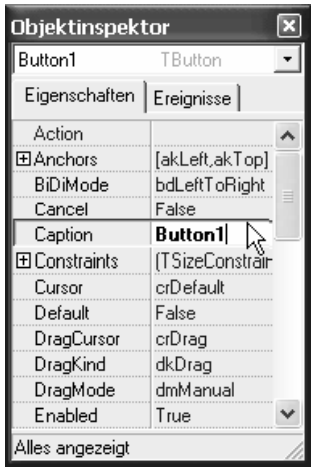
Auch in Delphi gibt es Objekte – dort spricht man neben Eigenschaften aber von Methoden.

Der Objektinspektor ist für die Verwaltung von Eigenschaften und Methoden eines Objektes zuständig. Und du vermutest richtig, dass Komponenten wie Formulare und Schaltflächen Objekte sind.

Weil es in Delphi – wie im richtigen Leben – mehrere Objekte ein und desselben Typs geben kann, fasst man dort alle Gemeinsamkeiten unter dem Begriff *Klasse* zusammen.

Der Objektinspektor zeigt die Eigenschaften der jeweils markierten Komponente an.

Eine Komponente in Windows – das ist z. B. ein Fenster, ein Menü, eine Schaltfläche usw. Und Delphi bietet die Möglichkeit, alle diese Komponenten in einem Programm einzusetzen.



Zu unserem Projekt gehören augenblicklich zwei Komponenten: Ein Formular mit dem Namen `Form1` und eine Schaltfläche namens `Button1`.

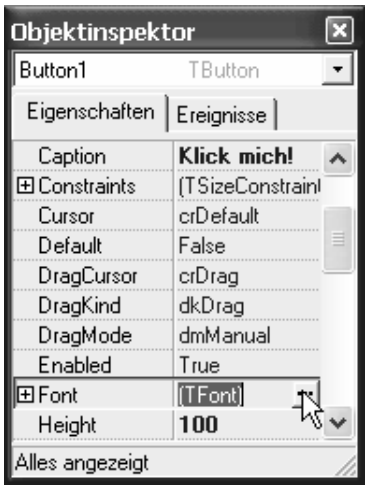
Eine Aufschrift für den Button

Normalerweise müsste die Schaltfläche noch markiert sein – ist sie das nicht, hilft ein Mausklick.

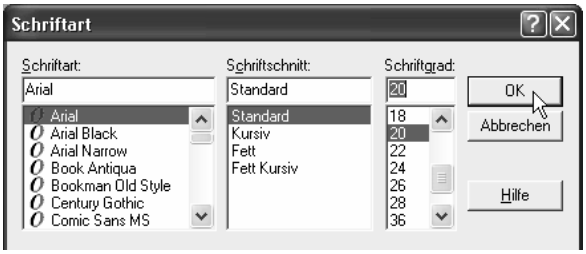
1. Klicke auf den Text hinter CAPTION und lösche ihn.
2. Gib einen neuen Text ein, z. B. „Klick mich!“

Und siehe da: Der Button hat eine neue Aufschrift, die allerdings noch ein bisschen mickrig aussieht.

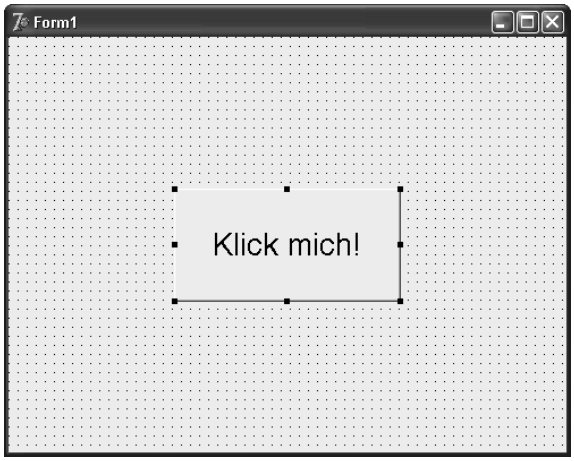
3. Deshalb suchst du im Objektinspektor die Eigenschaft FONT, unter der du u.a. Schriftart und Schriftgröße einstellen kannst.



4. Hier wählst du unter SCHRIFTGRAD eine Größe um die 20 aus.



Dann dürfte das Ganze in etwa so aussehen:



Ganz nett – aber wo ist nun das Programm? Und was bringt es, wenn ich auf diesen Button klicken würde?

Ehrlich gesagt, noch gar nichts. Bis jetzt besteht das Projekt auch nur aus einem Formular mit einer Schaltfläche.

Deshalb müssen wir jetzt klären, was passieren soll, wenn auf diesen Button geklickt wird.

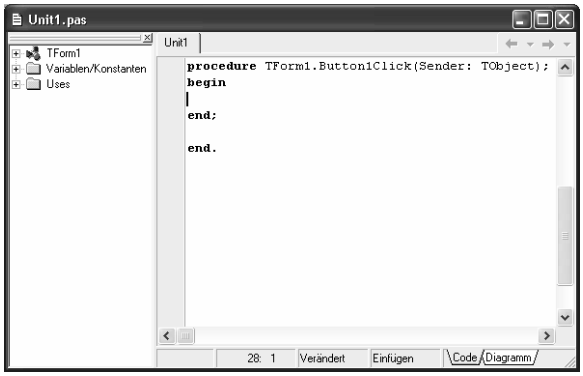
Beginnen wir einfach: Auf Mausklick soll der Button seine Aufschrift ändern, z. B. in „Autsch“ oder – noch eindrucksvoller – in „AUTSCH“.

Ereignis und Methode

Und damit landen wir bei den Methoden. Delphi bietet für den Button eine ganze Reihe von Methoden. Wir benötigen eine, die auf Mausklick reagiert.

- Dazu doppelklickst du einfach auf den Button.

Und schon landest du in einem neuen Fenster, das sich die ganze Zeit hinter dem Formular versteckt hat. In diesem Fenster findest du den *Editor*, in den du den Programmtext eingeben kannst. Ein Profi – der du ja werden willst – sagt übrigens statt Programmtext auch *Quelltext*.



Im Editorfenster steht der Doppelname `TForm1.Button1Click`. Zuerst kommt der Name des Objekttyps bzw. der Klasse, zu der die eigentliche Methode gehört: `TForm1`. Der folgende Methodenname erklärt sich dann selbst: `Button1Click`.

Die Zahlen darin besagen einfach nur, dass Delphi natürlich auch mehr als ein Formular und einen Button zu bieten hat – wenn nötig.

Der Punkt – so klein er ist – ist wichtig. Er verbindet beide Teile, den Klassennamen und den Methodennamen. Weil die Methode für alle Objekte gleich ist, wird sie unter dem Klassennamen vereinbart. Anders sieht es bei den Eigenschaften aus, denn diese können natürlich von Objekt zu Objekt verschieden sein – z. B. die Größe.

Genug geredet!

- Vollende die Methode, indem du sie so ergänzt:

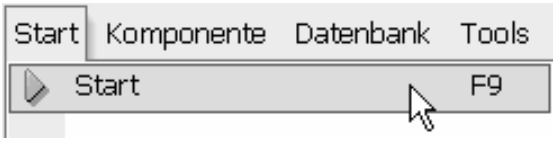
```
procedure TForm1.Button1Click ⏎
(Sender: TObject);
begin
  Button1.Caption := 'AUTSCH!';
end;
```

Weil’s hier ziemlich eng zugeht, passt nicht immer alles in eine Zeile. Das macht aber nichts, weil Delphi nicht zeilenorientiert arbeitet.

Hinweis zu den Quelltexten: Dieses Zeichen (⏎) bedeutet, dass die Zeile weitergeht und nicht umbrochen wird. Drücke also *nicht* auf [Enter]! Wenn vor dem Pfeil zusätzlich ein Leerzeichen steht (⏎), tippst du das Leerzeichen mit. Falls dort jedoch kein Leerzeichen eingefügt wurde, setzt du die Zeile ohne Leerzeichen fort.

Jetzt soll der Computer das Projekt ausführen.

- Dazu klickst du im Menü auf START und nochmals auf START oder drückst [F9].



Eine weitere Möglichkeit ist ein Klick auf das Start-Symbol unter der Menüleiste:

Das Formular hat seine Musterung verloren. Und wenn du jetzt auf unseren Button klickst, dann ändert sich dessen Aufschrift:

Immerhin: Es ist etwas passiert. Dazu brauchten wir außer einem Button auch nur eine einzige (zusätzliche) Programmzeile.

Wir haben es hier mit einer Kombination von Ereignis und Methode zu tun: Das *Ereignis* heißt Mausklick bzw. `OnClick`, und es hat die dazugehörige *Methode* `Button1Click` ausgelöst bzw. aktiviert.

Wechselhaft

Bevor es weitergeht, müssen wir dieses Programm beenden, um zu unserem Delphiprojekt zurückzukehren.

- Klicke dazu auf die Schaltfläche mit dem kleinen X – wie du es auch von anderen Windowsprogrammen gewöhnt bist. (Abbildung siehe nächste Spalte oben.)

Nun widmen wir uns der Zeile, um die wir die Methode `Button1Click` erweitert haben:

```
Button1.Caption := 'AUTSCH!';
```

Auch hier taucht wieder der Punktoperator als Verbindung zwischen zwei Namen auf: Das Objekt, um das es geht, heißt `Button1`, die Eigenschaft heißt `Caption`, und gemeint ist damit so etwas wie „Aufschrift“.

Dieser wird der Text „AUTSCH!“ zugewiesen. Und dafür ist der Zuweisungsoperator := da: Ein Doppelpunkt und ein Gleichheitszeichen **ohne** Leerzeichen dazwischen.

Man spricht deshalb auch von einer *Zuweisung*. Weitere Zuweisungen könnten sein:

```
X := 5;
Name := 'Erich';
```

Hier bekommt eine Variable namens *X* (die du ja aus dem Matheunterricht kennst) den Wert 5 zugewiesen. Mit Text geht das natürlich auch. Im zweiten Beispiel heißt die Variable *Name* und der zugewiesene Wert ist `'Erich'`.

Mich stört bei dem bisherigen Projekt, dass der Button nach dem Klicken die Aufschrift „Autsch“ behält. Schöner wäre es, würde nach einer kleinen Weile wieder der alte Text „Klick mich!“ erscheinen.

Damit kommen wir zu unserer nächsten Erweiterung:

```
procedure TForm1.Button1Click ↵
(Sender: TObject);
begin
  Button1.Caption := 'AUTSCH!';
  sleep (1000);
  Button1.Caption := 'Klick mich!';
end;
```

Nun wird nach einem Mausklick auf die Schaltfläche für eine Sekunde ein klagendes „Autsch“ angezeigt, um dann wieder der ursprünglichen Aufforderung „Klick mich!“ Platz zu machen.

Dafür, dass der zweite Text eine Sekunde Zeit hat, sorgt die Anweisung `sleep (1000)`, die die Programmausführung für 1000 Millisekunden „schlafen legt“.

Ein Projekt speichern

Wenn du das Projekt lange genug ausprobiert hast, solltest du es sicher auf der Festplatte unterbringen.

Das geschieht am besten über DATEI|ALLES SPEICHERN.

Da es vermutlich dein erstes Projekt ist, kannst du die vorgegebenen Namen übernehmen:

PROJECT1.DPR	Die Projektdatei mit den Verwaltungsdaten (DPR = Delphi PRojekt)
UNIT1.PAS	Die Datei mit dem Quelltext (PAS = Pascal), auch **Unit** genannt.

Hinzu kommen weitere Dateien, die Delphi automatisch erstellt, wie z. B. UNIT1.DFM mit den Formulardaten (DFM = Delphi ForMular).

Alle im Heft behandelten Beispiele wie auch die Lösungen zu den Aufgaben kannst du dir von der KnowWare-Homepage herunterladen. Du findest sie unter der Adresse *http://www.knowware.de/?book=delphi*

Reicht dir das fürs Erste, kannst du Delphi jetzt wie jedes normale Windowsprogramm beenden.

Aufgabe 1

Wenn du es dir zutraust, versuch doch mal, den Button „hüpfen“ zu lassen. Dazu müssen sich in `Button1Click` nur statt der (oder zusätzlich zur) Eigenschaft CAPTION die Werte für LEFT und TOP ändern.

Mathematik

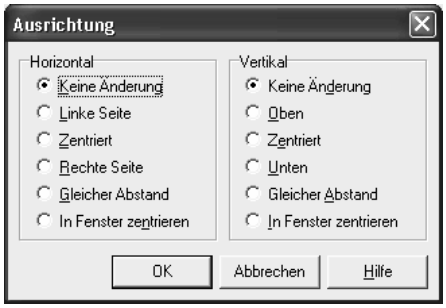
Und schon geht es weiter mit unserem nächsten Projekt (RECHNER1). Diesmal ist etwas Mathe angesagt. Aber keine Bange – nur die Grundrechenarten sind gefragt, Grundschulbildung genügt also – noch.

- Klicke auf das Symbol für Button und ziehe vier gleich große Schaltflächen im Formular auf.

Die Maße kannst du jeweils von Hand im Objektinspektor unter den Eigenschaften HEIGHT und WIDTH eintragen (z. B. einen Wert zwischen 60 und 90), die Position wird unter TOP und LEFT festgelegt. Die folgende Tabelle verschafft dir den nötigen Überblick:

WIDTH	Breite der Komponente
HEIGHT	Höhe der Komponente
LEFT	Position der Komponente links
TOP	Position der Komponente oben

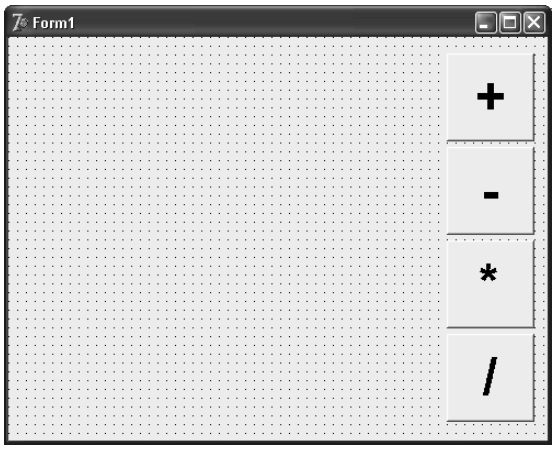
Wenn du mit der rechten Maustaste auf eine Komponente klickst, öffnest du ein Kontextmenü; über die Einträge POSITION | AUSRICHTEN erreichst du ein Dialogfeld, in dem du deine Buttons „in Position“ bringen kannst, wenn du sie zuvor alle zusammen markiert hast.



Über POSITION|GRÖßE im selben Kontextmenü kannst du auch die Maße für alle Buttons gleichzeitig festlegen.

- Bist du mit der Kosmetik fertig, bezeichnest du über CAPTION jeden Button mit einem Symbol für die vier Grundrechenarten (+, -, *, /)

Dann dürfte das ganze Formular etwa so aussehen:



Natürlich habe ich da bei der Eigenschaft FONT noch etwas nachgeholfen, damit alle Symbole auch groß genug sind.

Eine neue Komponente

Könnte eine Art Taschenrechner werden? Aber etwas Wichtiges fehlt: Die Anzeige. Dazu benötigen wir eine neue Komponente, ein so genanntes Label, auch Anzeigefeld genannt:

- Suche in der Symbolleiste das Symbol für LABEL und klicke darauf. Dann ziehe im Formular links neben der Buttongruppe ein Label auf.

Bevor du nun den Eintrag hinter CAPTION änderst, müssen wir uns zuerst um eine andere Eigenschaft kümmern.

- Stelle hinter AUTOSIZE den Wert auf FALSE. So wird verhindert, dass sich die von dir gewählte Größe der Anzeigefläche (automatisch) ändert und dem angezeigten Wert anpasst.