mitp für Kids

Visual Basic 2010 für Kids

von
Hans-Georg Schumann

1. Auflage

mitp/bhv 2010

Verlag C.H. Beck im Internet:
www.beck.de
ISBN 978 3 8266 8680 1

Über den Autor:
Hans-Georg Schumann ist Informatik- und Mathematiklehrer an einer Gesamtschule.
Er hat bereits viele erfolgreiche Bücher in der Reihe *„... für Kids“* geschrieben.

Ab 12 Jahre, aber auch für Erwachsene, die eine wirklich einfache Einführung suchen

Wolltest du schon immer in die Liga der Programmierer und Software-Entwickler aufsteigen? *Visual Basic 2010 für Kids* ist deine Eintrittskarte dafür!

Hans-Georg Schumann führt dich Schritt für Schritt in die Visual-Basic-Programmierung ein. Er zeigt dir, wie man Buttons und Labels anlegt und mit Operatoren und Methoden umgeht. Selbst vor der Objektorientierten Programmierung macht er nicht halt. Anhand spannender Beispiele mit viel Praxisbezug kommt der Spaß beim Lernen nicht zu kurz. Zwischendurch gibt es immer wieder Fragen und Aufgaben zum Lösen, um das Gelernte zu festigen.

Alle Projektbeispiele und die Lösungen zu den Aufgaben gibt es auf der beiliegenden DVD und zum Herunterladen im Internet. Und das Tollste: Auf der DVD findest du auch die Software Visual Basic 2010 Express, mit der du sofort loslegen kannst.

mit DVD

Auf der DVD:

Vollversion von Visual Basic 2010 Express sowie die Beispielprojekte und Lösungen aus dem Buch

mitp

www.mitp.de

Ebenfalls in dieser Reihe erscheinen:

ISBN 978-3-8266-8658-0

Schumann

Visual Basic 2010 FÜR KIDS

ISBN 978-3-8266-8680-1

9 783826 686801 (D)€ 24,95

mitp

mitp

Hans-Georg Schumann

Visual Basic 2010 FÜR KIDS

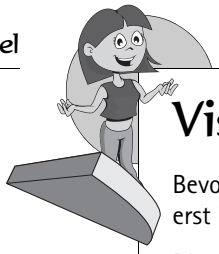
Inklusive DVD-ROM

1 Das erste Projekt

Du willst gleich loslegen? Dem Computer endlich mal etwas sagen, was er für dich tun kann? Na, dann schalte deinen PC an und lass erst mal Windows auftauchen. Von da aus geht es dann direkt zum ersten Programmprojekt in Visual Basic.

In diesem Kapitel lernst du

- wie man Visual Basic startet
- wie man ein Projekt erstellt und ausführt
- was ein Formular ist und wie man damit arbeitet
- was eine Komponente ist und wie man sie einsetzt
- die Komponente `Button` kennen
- wie man ein Projekt speichert
- wie man Visual Basic beendet



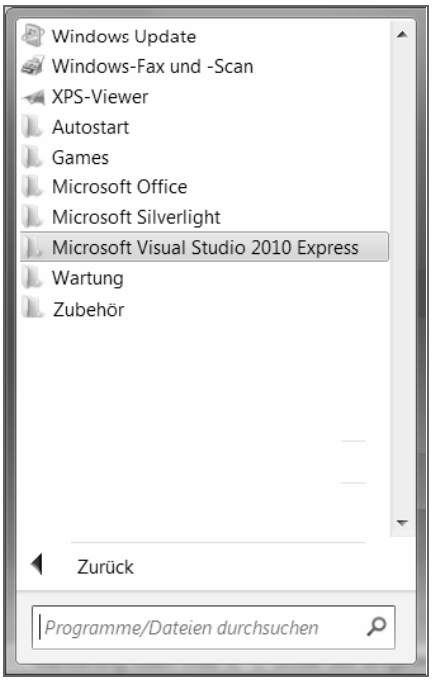
Visual Basic starten

Bevor wir mit dem Programmieren anfangen können, muss **Visual Basic** erst installiert werden.

Die Installation übernimmt ein Programm namens SETUP. Genaues erfährst du im **Anhang B**. Hier musst du dir von jemandem helfen lassen, wenn du dir die Installation nicht allein zutraust.

Die einfachste Möglichkeit, Visual Basic zu starten, ist diese:

- Klicke mit der Maus auf START und dann auf ALLE PROGRAMME.
- Dann klicke weiter auf MICROSOFT VISUAL BASIC EXPRESS EDITION. (Falls du einen anderen Namen gewählt hast, musst du dich darüber zu diesem Eintrag durchklicken.)



oder:

- Klicke mit der Maus auf START und dann auf AUSFÜHREN.
- Tippe `vbexpress.exe` ein und klicke dann auf OK. Wenn das nicht klappt, musst du den kompletten Pfad mit allen Ordnern eingeben, z.B.:

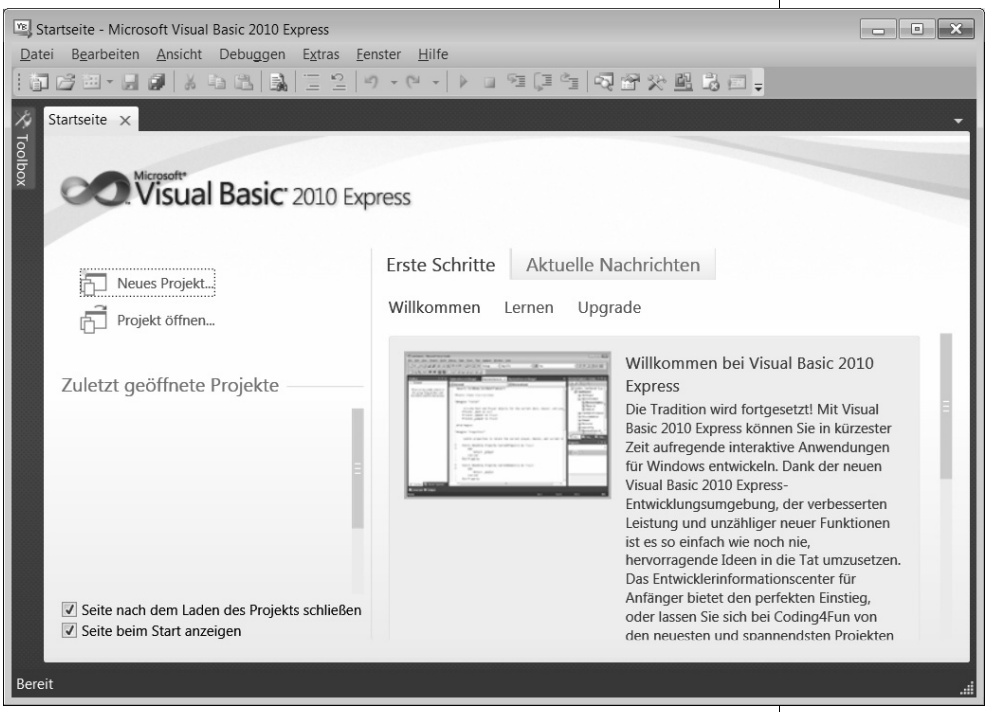
```
C:\Programme\Microsoft Visual Studio\Common7\IDE\vbexpress.exe

oder

C:\Basic\Common7\IDE\vbexpress.exe
```

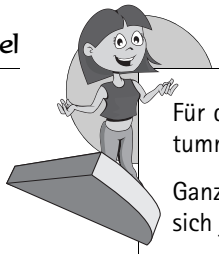
Oder du suchst über den Knopf DURCHSUCHEN nach der Datei VBEXPRESS.EXE.

Kleine Spritztour durch Visual Basic

Je nach Computer kann es eine Weile dauern, bis Visual Basic geladen ist.

Was dich schließlich erwartet, könnte ungefähr so aussehen – wobei das aktuelle Bild unter anderem davon abhängt, ob du eine Verbindung zum Internet hast oder nicht:





Für den ersten Augenblick ist das vielleicht ein bisschen verwirrend. Da tummelt sich ja einiges auf dem Bildschirm.

Ganz oben kann man die Menüleiste erkennen. Links darunter befinden sich jede Menge Symbole, die man mit der Maus anklicken kann.

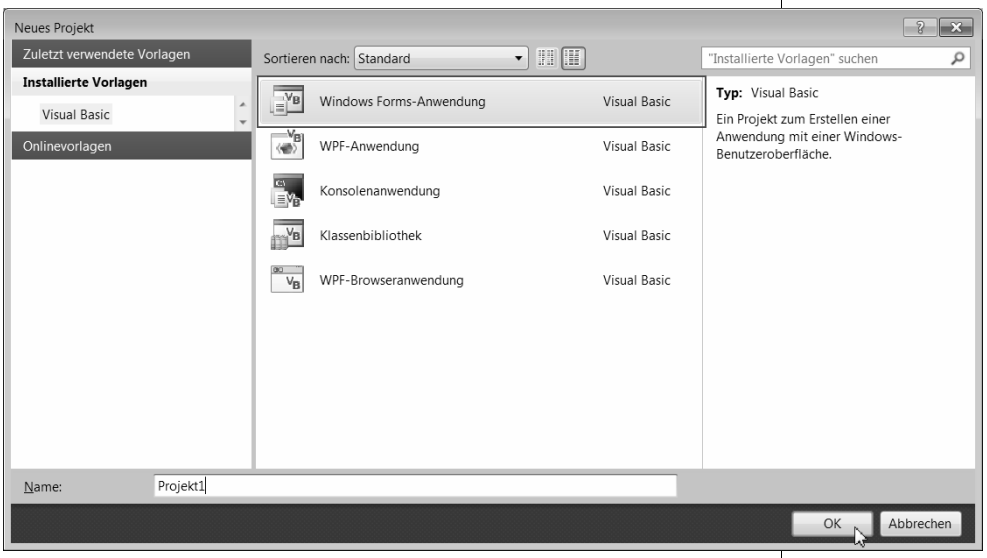
Diese Menüs von Visual Basic wirst du wahrscheinlich am meisten benutzen:

- Über das DATEI-Menü kannst du Dateien speichern, laden (öffnen), ausdrucken, neu erstellen oder Visual Basic beenden.
- Das BEARBEITEN-Menü hilft dir bei der Bearbeitung deines Programmtextes, aber auch bei anderen Programmelementen. Außerdem kannst du dort bestimmte Arbeitsschritte rückgängig machen oder wiederherstellen.
- Im ANSICHT-Menü hast du unter anderem die Möglichkeit, zusätzliche Hilfsfenster und Boxen ein- oder auszublenden.
- Über das (erst später erscheinende) DEBUGGEN-Menü sorgst du dafür, dass dein Profgrammprojekt ausgeführt wird.
- Und das HILFE-Menü bietet dir vielfältige Hilfe-Informationen an.

Einige wichtige Menüeinträge sind in einem so genannten **Popup**-Menü zusammengefasst. Das heißt so, weil es dort aufklappt, wo du gerade mit der **rechten** Maustaste hinklickst.

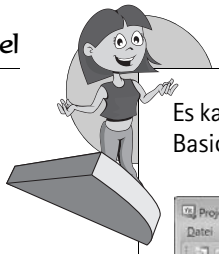
Ein Editorfenster, wie du es vielleicht von einem Editor oder Textverarbeitungsprogramm her kennst, ist gerade nicht in Sicht. Was tun? Unser Ziel ist es, ein neues Projekt – unser Erstlingswerk – zu erstellen. Also los!

- Klicke in der Menüleiste auf DATEI und im sich öffnenden Menü auf NEUES PROJEKT.
- Im folgenden Dialogfeld klickst du auf das Symbol für WINDOWS-ANWENDUNG.

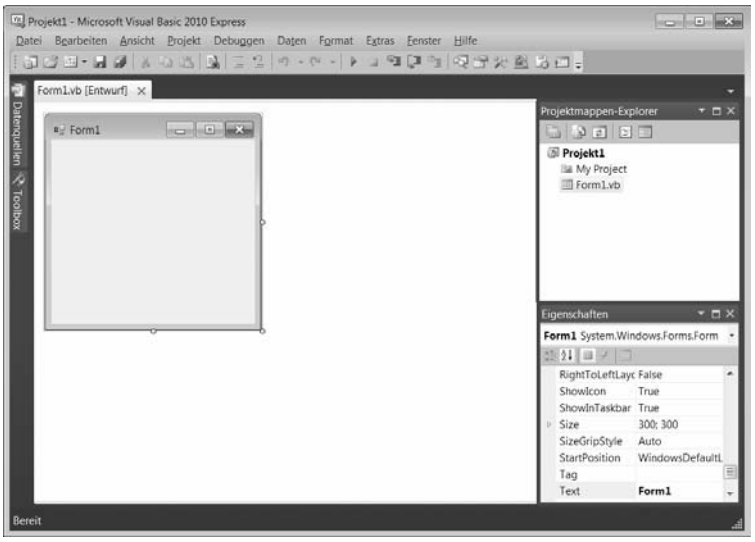


Wenn du willst, kannst du außerdem einen Namen für dein Projekt eingeben. Das ist beim ersten Versuch noch unwichtig, aber später solltest du schon einen Namen finden, der zum jeweiligen Projekt passt.

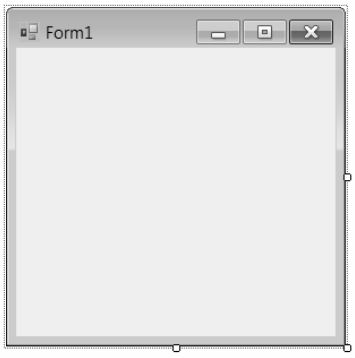
≫ Klicke zur Bestätigung auf OK.



Es kann etwas dauern, bis sich schließlich das Erscheinungsbild von Visual Basic etwa so geändert hat:

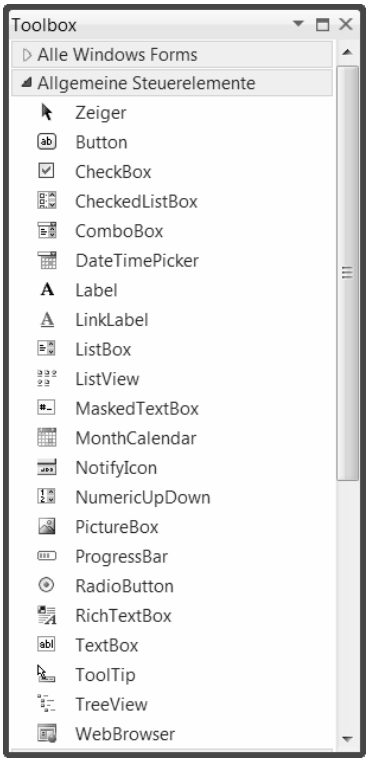


Was du links siehst, ist das so genannte **Formular**, in dem wir unsere Programmoberfläche zusammenbasteln.



Die benötigten Zutaten finden sich direkt daneben. Wenn du ganz links mit dem Mauszeiger auf den Eintrag TOOLBOX fährst, öffnet sich ein (ellenlanges) Menü. Darin stehen die **Komponenten**, die Visual Basic dir zur Verfügung stellt.

Damit lassen sich z.B. Schaltflächen, Dialogfelder oder Menüs selbst gestalten und in ein Programm einbauen. Schon für unser erstes Visual-Basic-Projekt werden wir uns in dieser Komponentensammlung bedienen.

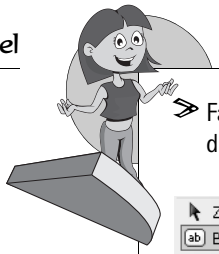


Hallo auf Knopfdruck

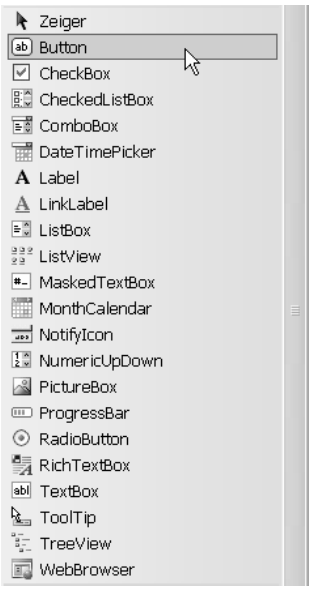
Eigentlich kann's jetzt schon losgehen. Den Umgang mit Menüs und Dialogfenstern kennst du bereits von Windows. Deshalb müssen wir uns damit nicht mehr aufhalten. Bauen wir uns jetzt ein kleines Projekt, das auf Knopfdruck funktioniert.

Wir brauchen dazu eine Komponente, die **Button** genannt wird. Man sagt dazu auch **Schaltfläche**. Und viele sprechen einfach von einem Knopf. Wichtig ist, dass man darauf mit der Maus klicken kann.

Nun hat die Toolbox von Visual Basic nicht nur eine Knopfart zu bieten. Der Button, den wir suchen, befindet sich im Bereich ALLGEMEINE STEUERELEMENTE.

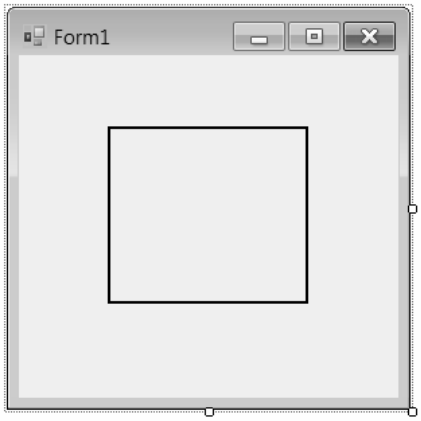


≫ Fahre mit der Maus langsam über die Einträge. Wenn du BUTTON gefunden hast, klicke darauf.

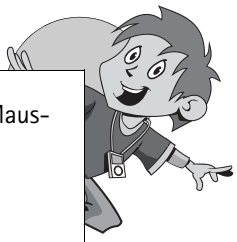


≫ Wechsle mit dem Mauszeiger nun in das Formular.

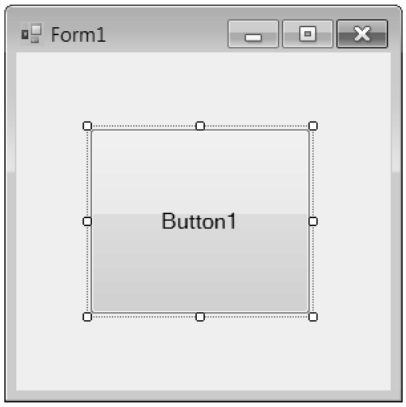
≫ Drücke die linke Maustaste, halte sie gedrückt und ziehe mit der Maus schräg nach unten.



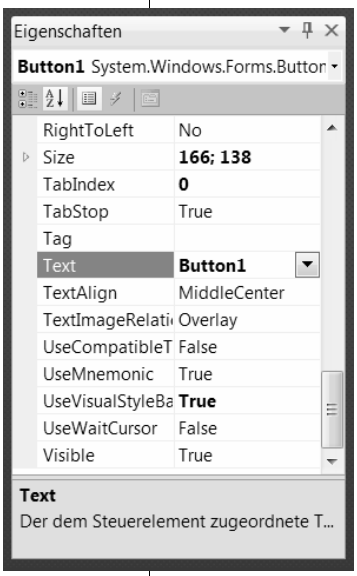
Ein Rahmen für den Button

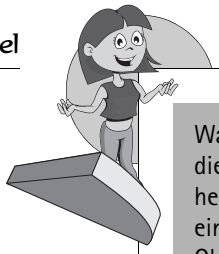


≫ Wenn du meinst, dass die Schaltfläche groß genug ist, lass die Maustaste wieder los.



Da hast du nun eine Schaltfläche (oder einen Knopf) mit der Aufschrift BUTTON1. Eigentlich ein blöder Name! Den sollten wir gleich ändern. Dazu brauchen wir ein Fenster, das wir bisher gar nicht weiter beachtet haben: Dort stehen alle **Eigenschaften**, die ein Objekt bzw. eine Komponente betreffen, also z.B. die Größe, die Lage oder der Name.

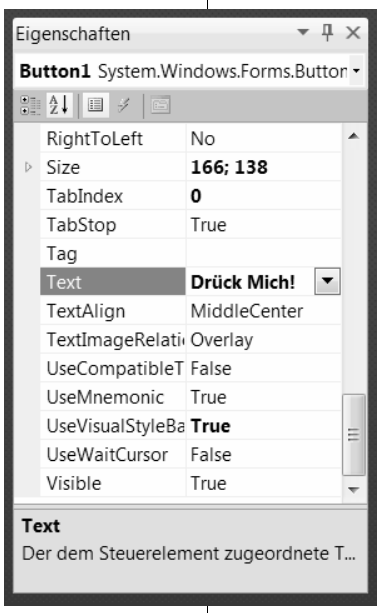




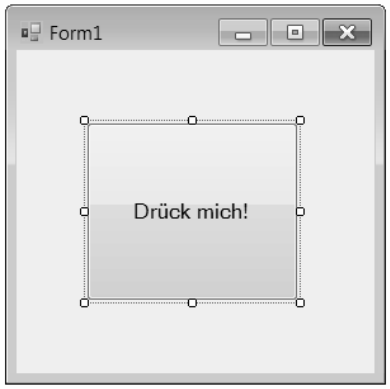
Was versteht man hier unter einem **Objekt**? Das sind doch eigentlich diese Dinger, die ständig irgendwo herumstehen oder sich um uns herum bewegen. Also z.B. Häuser, Bäume, Autos, Leute. Auch du bist ein Objekt. Und zwar vom Typ Mensch. Ebenso gibt es in Visual Basic Objekte. Die sind natürlich nur künstlich. So ein Objekt ist beispielsweise das Formular, und auch der Knopf, den du gerade dort hinein gesetzt hast, ist ein Objekt. Alle Komponenten sind Objekte.

Dabei kann es auch in Visual Basic mehrere Objekte eines Typs geben – so wie im richtigen Leben auch. Dann spricht man von einer **Klasse**, womit dasselbe gemeint ist wie mit **Objekttyp**. Und ein Objekt wird auch als **Instanz** einer Klasse bezeichnet. Demnach bist du eine Instanz der Klasse Mensch.

➢ Klicke auf den Text hinter TEXT und lösche ihn. Dann tippe dort ein: `Drück mich!`



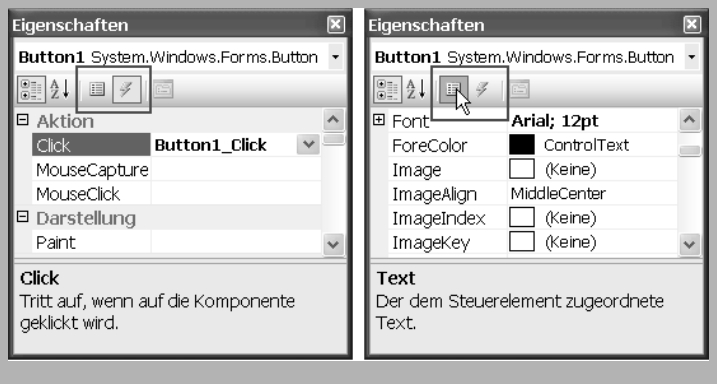
Und schon erscheint dieser Text auch als Aufschrift auf der Schaltfläche:

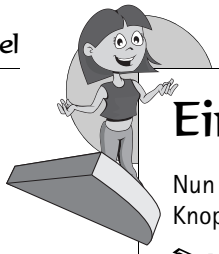


Ein Button zum Drücken

Du findest die Eigenschaft nicht? Stattdessen stehen da lauter Namen wie z.B. »Click«, »Paint« oder »DragDrop«. Da bist du wohl im Eigenschaftenfenster irgendwie auf die falsche Seite geraten. Dort stehen nämlich die Ereignisse.

Klicke ganz oben im Fenster auf das Symbol **links neben** dem Blitz. So bist du wieder auf der richtigen Seite.



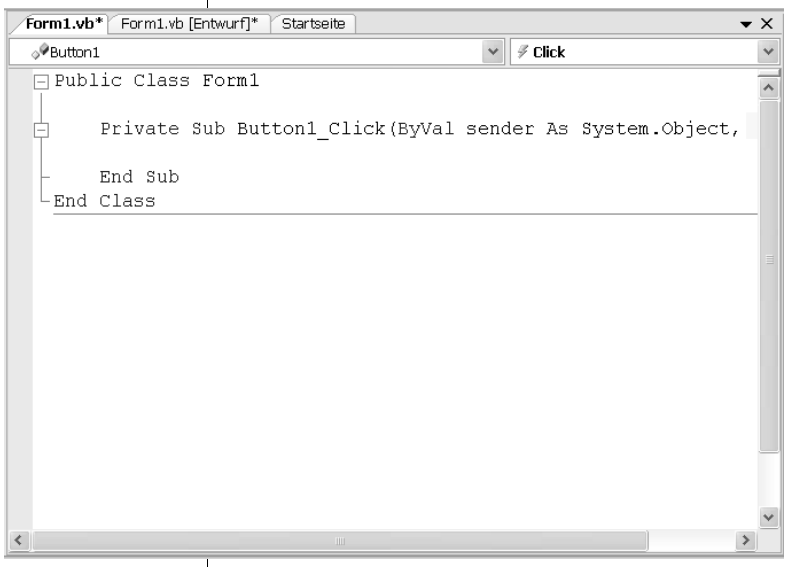


Eine Methode zum Drücken

Nun müssen wir noch erreichen, dass unser Programm auch auf den Knopfdruck einen Gruß losschickt.

≫ Doppelklicke im Formular auf die Schaltfläche mit dem Text DRÜCK MICH.

Ein neues Fenster kommt zum Vorschein. Das ist das **Editorfenster** von Visual Basic, dort steht der Quelltext deines Programms. Man kann also auch von Quelltextfenster sprechen.

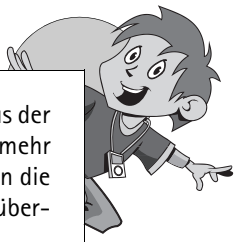


Dort findest du schon den Namen der Methode, die für die Schaltfläche DRÜCK MICH zuständig ist:

```
Private Sub Button1_Click
```

Groß oder klein?

Im Gegensatz zu manchen anderen Programmiersprachen ist es in Visual Basic gleichgültig, ob du deine Wörter groß- oder kleinschreibst. Lass am besten alles stehen, was Visual Basic dir bereits vorgibt. Ansonsten kannst du meinen Gewohnheiten in diesem Buch folgen oder deine eigene Schreibweise benutzen.



Ganz oben steht mit `Public Class Form1` der Name der Klasse, aus der das Formular abgeleitet ist (und die 1 weist darauf hin, dass es auch mehr als ein Formular geben kann). `Form1` ist sozusagen die »Backform«, in die unser Programm eingepackt ist. Und `Button1_Click` heißt frei übersetzt »Knopf1druck« oder »Drücke auf Knopf Nr.1«.

Und über den Unterstrich (_) verbindet Visual Basic den Namen einer Komponente und einer solchen Methode. Man kann also auch von **Verbindungsoperator** sprechen. Allerdings hat der Unterstrich noch eine andere Bedeutung – wie du bald sehen wirst. Die Klammern hinter der Methode samt langem Inhalt müssen uns jetzt und hier nicht interessieren.

Wie du weiter oben schon erfahren hast, besitzen Objekte **Eigenschaften**. Ein Objekt mit Eigenschaften allein ist aber ziemlich leblos. Wie ein Stein z.B. oder eine Straße. Lebendig wird ein Objekt erst durch seine **Methoden.**

Ein Auto beispielsweise hat nicht nur bestimmte Eigenschaften, sondern es kann sich bewegen, es kann beschleunigen oder bremsen, es lässt sich lenken, man kann es starten und auch wieder anhalten. All das sind Prozesse, die man als Methoden bezeichnen würde.

Damit nun ein Knopf (Button) überhaupt z.B. auf einen Mausklick reagieren kann, braucht er (mindestens) eine Methode. Gleiches gilt für alle anderen Komponenten und auch für die meisten Objekte. Manche von ihnen haben gleich zehn bis zwanzig Methoden (oder noch mehr).

Viele Methoden reagieren auf **Ereignisse**, die ebenfalls (wie die Eigenschaften) mit dem betreffenden Objekt verknüpft sind.

Wir müssen nun zwischen `Private Sub` und `End Sub` etwas einfügen. Dort gehört eine Anweisung hin, die Visual Basic sagt, was auf Knopfdruck geschehen soll.

```
Private Sub Button1_Click (ByVal sender As _
System.Object, ByVal e As System.EventArgs) _
Handles Button1.Click
  Button1.Text = "Hallo!"
End Sub
```

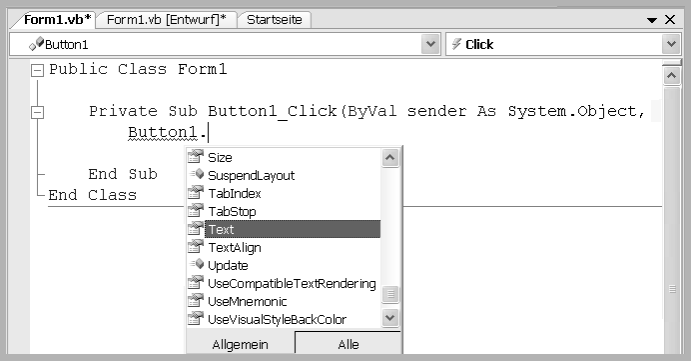
Hier ist er wieder: der Unterstrich (_). Er gehört eigentlich nicht zum Programmtext, sondern zeigt nur an, dass auch die nachfolgenden Zeilen im Editor von Visual Basic zusammen mit der ersten in **einer einzigen** Zeile stehen müssten! Denn Visual Basic erwartet eine Anweisung eigentlich komplett in einer Zeile.

Sollte eine solche Zeile aber mal zu lang werden oder möchtest du aus optischen Gründen mehrere Zeilen verwenden, benutzt du einfach am Zeilenende einen Unterstrich. Dann weiß Visual Basic, dass es mit der Anweisung in der folgenden Zeile weitergeht.

Wichtig ist, dass vor dem Unterstrich ein Leerzeichen stehen muss!

≫ Ergänze die Methode um die eine Zeile `Button1.Text = "Hallo!"`.

Wenn du dir beim Eintippen etwas zu viel Zeit lässt, ist das nicht schlimm – im Gegenteil: Du bekommst von Visual Basic sogar Unterstützung. Zu jedem Objekt, das Visual Basic bekannt ist, werden dir in einer Liste dessen Eigenschaften und Methoden angeboten.



Klickst du auf den Namen und drückst dann die Eingabe-Taste, so wird der Name in den Quelltext übernommen.

Denn Visual Basic erwartete in früheren Versionen eine Anweisung eigentlich komplett in einer Zeile. Seit Version 2010 kann dieser Unterstrich auch weggelassen werden.

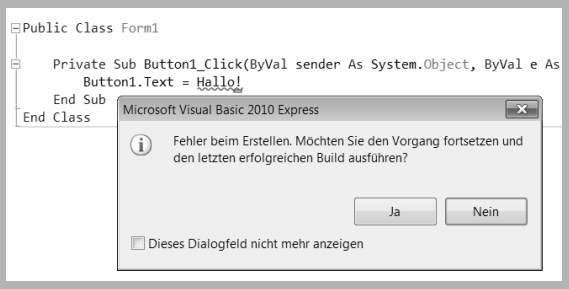
Vielleicht hast du schon eine Ahnung, was `Button1.Text` bedeutet. Aber lass das Programm erst mal laufen.

≫ Klicke dazu auf DEBUGGEN und dann auf DEBUGGEN STARTEN.

Noch kürzer und schneller geht es, wenn du einfach auf die Taste F5 drückst oder links oben unter der Menüleiste auf diesen Knopf klickst:

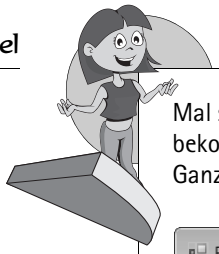
Diesmal erscheint das Formular wie losgelöst (von Visual Basic). Und die Schaltfläche lädt dich zum Draufdrücken ein.

Bei dir läuft das Programm gar nicht? Stattdessen erscheint ein Dialogfeld mit einer solchen oder ähnlichen Fehlermeldung:

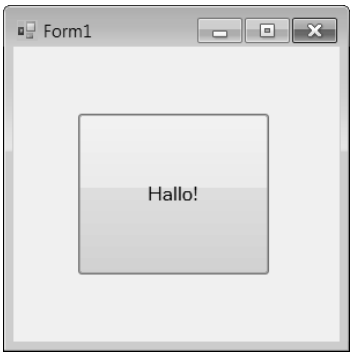


Zusätzlich ist die Stelle markiert, in der Visual Basic den Fehler vermutet. Das kann ein Schreibfehler sein, hier wurden die Anführungsstriche vergessen. (Nähere Informationen bekommst du über ein Extrafenster, das sich in der Regel unter dem Quelltext öffnet.)

Bessere die Stelle aus und starte dann das Programm einfach noch einmal.

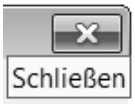


Mal sehen, was ein Mausklick bringt: Eigentlich nicht allzu viel, denn nun bekommt der Button nur eine neue Aufschrift. Ein bisschen mickrig, das Ganze. Aber das Wörtchen »Hallo« ist doch zu erkennen.



Einen wichtigen Unterschied zwischen Programmbearbeitung und Programmausführung erkennt man nicht nur daran, dass sich das Äußere des Formulars ändert:

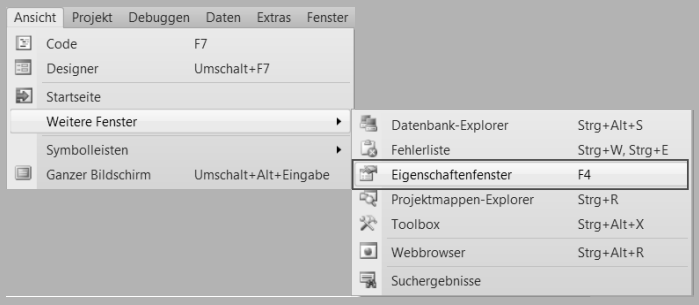
- Bei der **Programmbearbeitung** können Formular und Schaltfläche beliebig geändert werden. Außerdem lassen sich weitere Eigenschaften über das entsprechende Fenster festlegen. Und nur bei der Bearbeitung kannst du Quelltext im Editorfenster eintippen. Man nennt diese Phase auch **Entwicklungszeit**.
- Bei der **Programmausführung** dagegen treten die Komponenten in Aktion. Du kannst eine Schaltfläche anklicken und damit eine Methode aktivieren, die z.B. die Aufschrift der Schaltfläche ändert. Ein laufendes Programm muss ordnungsgemäß beendet werden, beispielsweise durch Klick auf ein Schließsymbol oder per Tastendruck. Man spricht bei dieser Phase auch von **Laufzeit**.



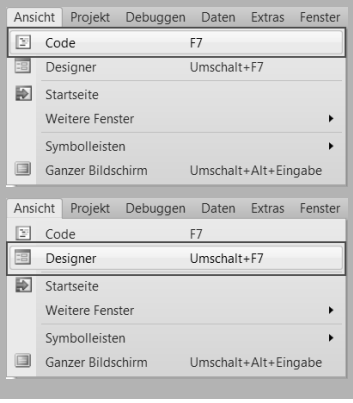
➢ Beende das Programm, indem du im Formular auf das kleine **X** ganz oben rechts klickst. Oder du drückst die Tastenkombination Alt F4.

Du hast aus Versehen zu viele Fenster zugeklickt? Das Formular ist weg? Oder das Eigenschaftenfenster? Dann bekommst du die verschwundenen Fenster so wieder zusammen:

- Klicke auf einen der Registereinträge oberhalb der Fenstergruppe, um zwischen Quelltext- und Formularanzeige zu wechseln.
- Klicke auf ANSICHT und dann auf EIGENSCHAFTENFENSTER.



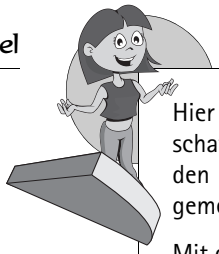
- Oder auf ANSICHT und auf CODE (für den Quelltext) oder auf DESIGNER (für das Formular).



Von »Drück mich« zu »Hallo«

Jetzt ist es an der Zeit, die Anweisungszeile in der Methode `Button1_Click` ein bisschen genauer unter die Lupe zu nehmen:

```
Button1.Text = "Hallo!"
```



Hier wird einer Eigenschaft namens `Text` etwas zugewiesen. Diese Eigenschaft kennst du bereits, denn du hast ihr im Eigenschaftenfenster schon den Text »Drück mich!« verpasst. Dass mit `Button1` die Schaltfläche gemeint ist, die dich zum Drücken einlädt, ist also klar.

Mit dem Punkt (**.**) werden in Visual Basic Objekt und Eigenschaft verbunden. Man nennt dieses Symbol hier auch **Zugriffsoperator**. (Man könnte aber auch hier von einem Verbindungsoperator sprechen.)

Was ist eigentlich mit diesem Gleichheitszeichen (**=**)? Man nennt es Zuweisungszeichen. Es wird ja auch einer Eigenschaft (`Text`) etwas (`"Hallo!"`) zugewiesen. Weshalb diese Anweisung gleichzeitig eine **Zuweisung** ist.

Mit der Anweisung `Button1.Text = "Hallo!"` wird bei jedem erneuten Drücken auf den Button dessen Anzeige immer wieder auf »Hallo« gesetzt, das »Drück mich« bleibt nach dem ersten Mal leider verschwunden.

Das lässt sich auch nur mit etwas Aufwand ändern. Dazu kommen wir erst später. Wenn du willst, kannst du mit anderen Texten statt dem einfachen »Hallo« experimentieren.

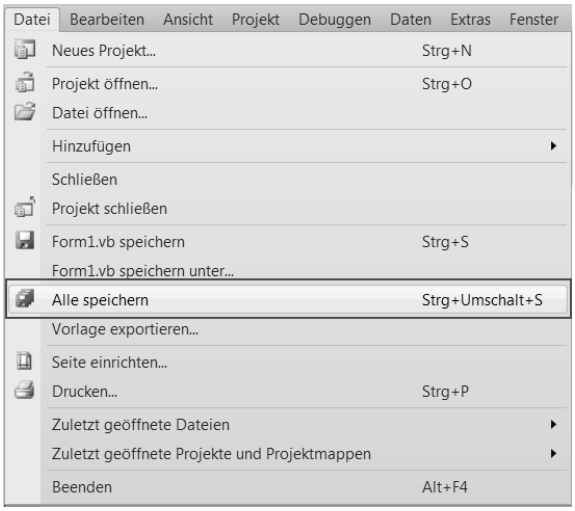
- Das Editorfenster erreichst du durch Doppelklick auf die Schaltfläche DRÜCK MICH. Oder du klickst auf den Eintrag FORM1.VB.
- Und das Programm wird dann über DEBUGGEN und DEBUGGEN STARTEN zum Laufen gebracht. Oder du drückst F5.
- Um das Programm zu beenden, klicke auf das kleine **X** rechts oben im Formular. Oder du drückst Alt F4.

Das Projekt speichern

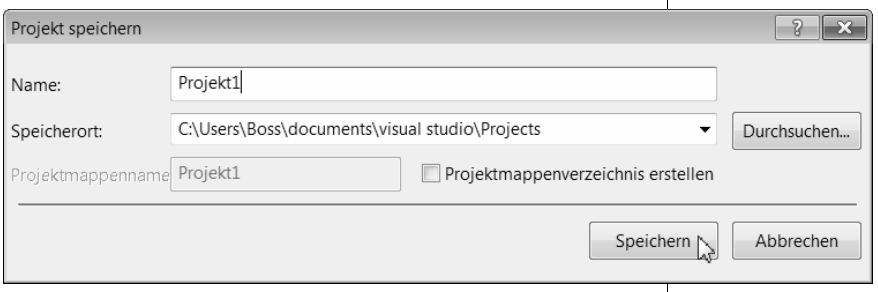
Allzu umfangreich ist unser erstes Programmprojekt bis jetzt zwar nicht, aber du solltest dennoch schon mal speichern, was du bis jetzt geschafft hast.

≫ Klicke auf DATEI und dann auf ALLES SPEICHERN.

Das Projekt speichern

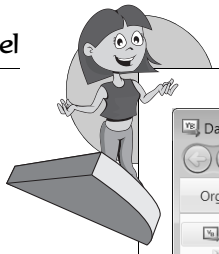


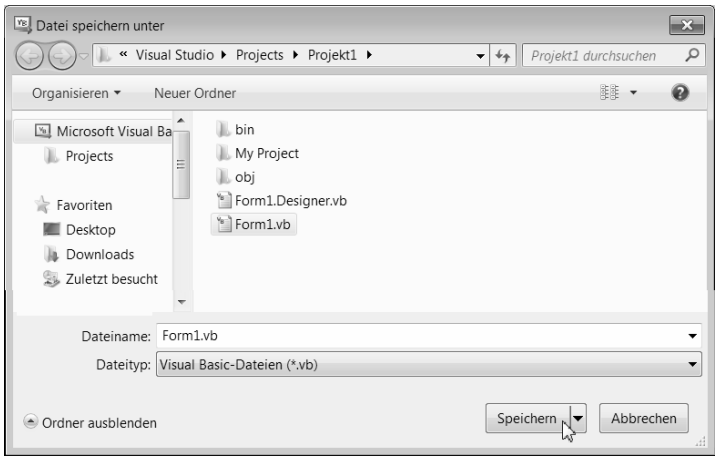
Nun öffnet sich ein Dialogfeld und bietet dir ein Verzeichnis an, in dem du deine Projektdateien ablegen kannst:



≫ Gib dort `Projekt1` (oder einen Namen deiner Wahl) ein. Dann bestätige mit Klick auf SPEICHERN.

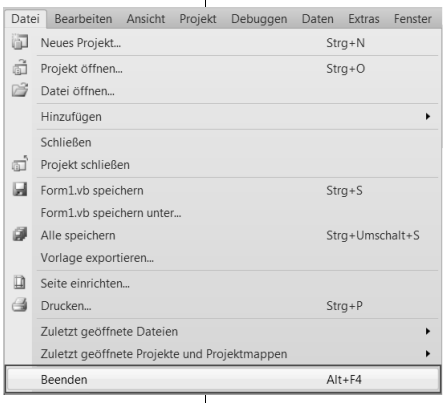
Natürlich kannst du Dateien auch einzeln speichern (über ... SPEICHERN UNTER), z.B. wenn du den vorgegebenen Namen FORM1.VB nicht übernehmen oder eine Kopie davon speichern willst.





Wie du siehst, gibt es in Visual Basic eine ganze Reihe von Dateien, aus der ein Projekt besteht:

- Eine davon ist der Programmtext, auch **Quelltext** genannt. Der wird als Datei mit der Kennung VB (für **V**isual **B**asic) versehen.
- Daneben verwaltet Visual Basic noch zusätzliche Dateien, die auch automatisch erzeugt werden. (Das siehst du, wenn du nach dem Speichern eines Projektes mal im Projektordner nachschaust, was da alles auf deiner Diskette oder Festplatte abgelegt wurde.)

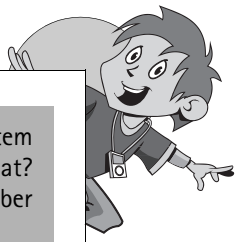


Visual Basic beenden

Dein allererstes Projekt ist sicher auf deiner Festplatte oder Diskette gelandet. Zeit also für eine kleine Pause. Willst du Visual Basic verlassen, dann geht das so:

≫ Klicke auf DATEI und dann auf BEENDEN.

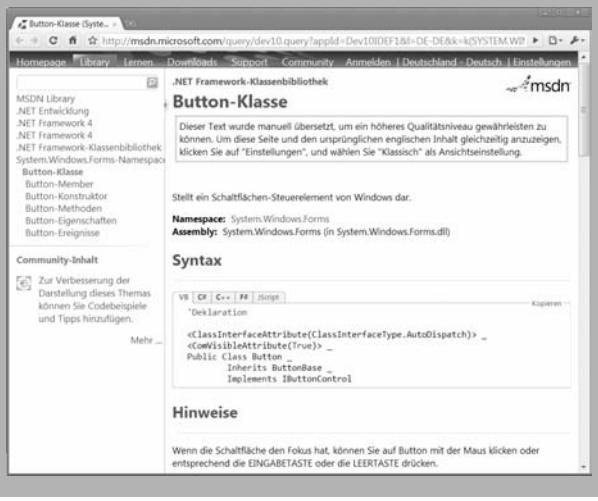
Oder du drückst die Tastenkombination Alt F4. Du kannst auch im Hauptfenster ganz oben rechts auf das kleine **X** klicken. Irgendwie kommst du also immer »nach Hause«.

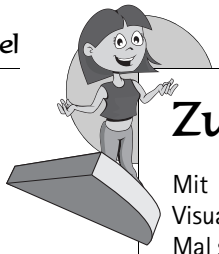


Bist du besonders neugierig und willst wissen, was das Hilfesystem von Visual Basic zu einem Wort oder einem Thema zu erzählen hat? Dann versuch es mal mit der Taste F1. Oder du bedienst dich über das HILFE-Menü.

Beim ersten »Hilferuf« musst du wählen, ob du die lokale Hilfe (auf der Festplatte) oder Hilfe übers Internet in Anspruch nehmen willst.

Nachdem dann ein neues Fenster erschienen ist, kannst du dich dort ein bisschen durch das **Hilfesystem** von Visual Basic hangeln und schon mal staunen, was diese Entwicklungsumgebung alles auf der Pfanne hat. Irgendwann aber solltest du das Hilfefenster mit Mausklick auf das **X** ganz oben rechts wieder schließen.





Zusammenfassung

Mit deinem ersten Projekt gehörst du zwar noch nicht zur Gilde der Visual-Basic-Programmierer, aber die Anfangsschritte hast du hinter dir. Mal sehen, was du von diesem Kapitel behalten hast. Da wären zuerst mal ein paar Operationen im Umgang mit Visual Basic:

Visual Basic starten	Doppelklicke auf das Visual-Basic-Symbol oder klicke auf START/ALLE PROGRAMME/VISUAL BASIC – o.Ä.
Komponente auswählen	Klicke in das TOOLBOX-Menü
Methode bearbeiten	Doppelklicke auf die Komponente
Eigenschaft festlegen	Klicke im Eigenschaftenfenster auf die Spalte hinter den Eigenschaftsnamen
Ganzes Projekt speichern	Klicke auf DATEI/ALLES SPEICHERN
Programmprojekt starten	Klicke auf DEBUGGEN/DEBUGGEN STARTEN
Programmprojekt beenden	Klicke auf das **X** oder drücke Alt F4
Hilfesystem aufrufen	Klicke auf HILFE oder drücke F1
Visual Basic beenden	Klicke auf DATEI/BEENDEN

Verschwundene Fenster lassen sich so wieder öffnen:

Formularfenster	Klicke auf ANSICHT/DESIGNER
Editorfenster	Klicke auf ANSICHT/CODE
Eigenschaftenfenster	Klicke auf ANSICHT/EIGENSCHAFTENFENSTER

Ein bisschen was vom Visual-Basic-Wortschatz hast du auch schon kennen gelernt:

`Form`	Das Formular, in dem sich alle Komponenten des Programms befinden. Das Formular selbst ist die Hauptkomponente.
`Button`	Eine Schaltfläche, auf die man mit der Maus klicken kann
`Button_Click`	Diese Methode wird mit Mausklick auf eine Schaltfläche aktiviert.
`Text`	Eine Eigenschaft, die viele Komponenten haben: Gemeint ist damit eine Aufschrift oder ein Anzeigetext.
`_` (Unterstrich) `.` (Punkt)	Operatoren für die Verbindung von Klasse/Objekt und Methoden/Eigenschaften (Unterstrich ohne Leerzeichen!)
`_` (Unterstrich)	Trennsymbol für lange Anweisungen, die auf mehrere Zeilen verteilt werden sollen (mit Leerzeichen!)

Deine Arbeit an einem Projekt spielte sich vorwiegend in drei Bereichen ab:

Im **Formularbereich** baust du deine Komponenten (wie z.B. Schaltflächen) zusammen.

Im **Editorfenster** tippst du deinen Quelltext (z.B. für Methoden) ein.

Im **Eigenschaftenfenster** bestimmst du die Eigenschaften einer Komponente.

Ein paar Fragen ...

1. Wie beendet man ein Programm in Visual Basic?
2. Warum spricht man in Visual Basic nicht nur von Programm, sondern von Projekt?
3. Was ist der Unterschied zwischen Formularfenster und Editorfenster?

... aber noch keine Aufgabe