



mitp

Hans-Georg
Schumann



Visual Basic 2010

FÜR **KIDS**



Inklusive DVD-ROM

Hans-Georg Schumann



Visual Basic 2010 für Kids



mitp

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte bibliografische
Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN 978-3-8266-8470-8
1. Auflage 2010

E-Mail: kundenbetreuung@hjr-verlag.de

Telefon: +49 89/2183-7928
Telefax: +49 89/2183-7620

www.mitp.de

© 2010 mitp, eine Marke der Verlagsgruppe Hüthig Jehle Rehm GmbH
Heidelberg, München, Landsberg, Frechen, Hamburg

Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt.
Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist
ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere
für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die
Einspeicherung und Verarbeitung in elektronischen Systemen.

Lektorat: Katja Völpel
Korrektur: Petra Heubach-Erdmann
Satz: Ill-satz, Husby, www.drei-satz.de

Inhalt

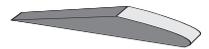


Vorwort	15
Einleitung	19
Das erste Projekt	23
Visual Basic starten	24
Kleine Spritztour durch Visual Basic	25
Hallo auf Knopfdruck	29
Eine Methode zum Drücken	34
Von »Drück mich« zu »Hallo«	39
Das Projekt speichern	40
Visual Basic beenden	42
Zusammenfassung	44
Ein paar Fragen	45
... aber noch keine Aufgabe	45
Buttons und Labels	47
Ein Projekt wieder öffnen	48
Noch ein Knopf	49
Gut oder schlecht?	54

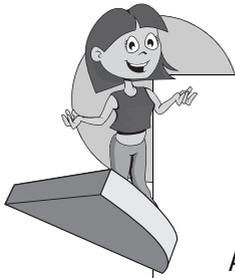
1



2



5



Antwort per Label	56
Speichern und ausprobieren	59
Ereignisse und Methoden	60
Veredelung	61
Ein ganz neues Projekt.	64
Komponentenschwemme.	66
Zusammenfassung	68
Ein paar Fragen	69
... und eine Aufgabe	69

3

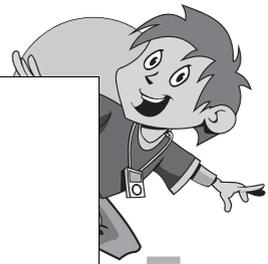


Von Zahlen, Zeichen und Operatoren	71
Die passende Optik.	72
Zufallszahlen und Variablen	74
Zeichenketten.	76
Vereinbarungen sind nötig	78
Kommentare	80
Plus oder minus, mal oder durch.	80
Zahlen mit Format	85
Zusammenfassung	86
Keine Fragen	87
... aber ein paar Aufgaben	87

4



Bedingungen	89
Von 1 bis 6	90
Wenn ... dann	92
Die If-Then-Struktur	94
Die Sache mit Try und Catch.	96
Punkt für Punkt	98
Und und Oder, oder?	101
Zusammenfassung	103
Nur eine Frage	104
... und ein paar Aufgaben.	104



Ein Ratespiel mit Zahlen	105
Zu groß, zu klein	106
Es kann geraten werden	108
Dein PC zählt mit.	109
Die If-Then-Else-Struktur	111
Neues Spiel oder wirklich Schluss?	113
Zusammenfassung.	117
Eine Frage	117
... und ein paar Aufgaben	118

5



Kontrollstrukturen	119
Auf dem Weg zum Millionär.	120
While oder Until?	121
Schleifenvariationen	123
If und Exit.	126
Kleine Knopfparade	127
Diagnose auf Knopfdruck	128
Listenwahl	130
Von Fall zu Fall.	134
Zusammenfassung.	135
Ein paar Fragen	136
... und ein paar Aufgaben	136

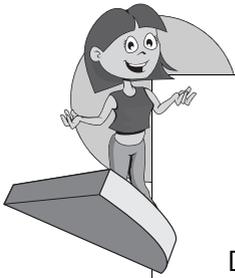
6



Combo, Radio oder Check?	139
Alles in einer Box.	140
Von Pünktchen	143
... und Häkchen	145
Der letzte Schliff	147
Antworten für die Optionsfelder	149
Wirklich fertig?	151
Variablenfelder und Startwerte	152
Die richtige Wahl.	154

7





8

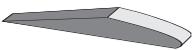


Die For-Next-Struktur	156
Zusammenfassung	158
Nur eine Frage	160
... und nur eine Aufgabe.....	160

Aktion Seelenklempner 161

Zwei Buttons und ein paar Gruppenfelder.....	162
Eingabefeld, Anzeigetafel und Bildlaufleiste	163
Vor der Sprechstunde.....	168
Bereit zur Diagnose	169
Schiebereien	170
Noch mehr Diagnosen?	173
Neue Sub – selbst gemacht.....	175
Datentransfer	176
Strings aus der Liste.....	178
Keine Sprechstunde?	180
Therapieprotokoll	181
Zusammenfassung	184
Ein paar Fragen	186
... und ein paar Aufgaben.....	186

9



Menüs und Dialoge 187

Ein Menü für den Klempner	188
Zwei Dialogfelder	192
Öffnen und Speichern	196
Diagnosen drucken.....	199
Sicherheitsabfrage	202
... und Schluss.....	207
Zusammenfassung	208
Ein paar Fragen	209
... aber keine Aufgabe.....	209



Grafik	211
Von Punkten und Koordinaten	212
Das erste Bild	214
Jetzt wird's bunt	216
Eckig und rund	219
Mit Text geht's auch	221
Farbtupfer	223
Zusammenfassung	225
Ein paar Fragen	226
... und ein paar Aufgaben	226

10



Eine eigene Klasse	227
Erst mal ein Kreis	228
Und es bewegt sich doch	230
Eine neue Klasse	232
Eigenschaften, Methoden und ein Konstruktor	234
Vereinbaren und initialisieren	235
Erscheinen, Bewegen und Verschwinden	236
Funktion oder Prozedur	239
Zusammenfassung	241
Nur eine Frage	242
... und eine Aufgabe	242

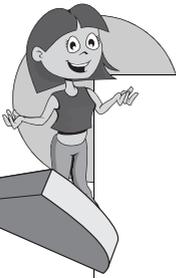
11



Kapselung und Vererbung	243
Alles unter einem Hut	244
Es gibt was zu erben	245
Ein neues Baby?	247
Ein Hand voll Set und Get	249
Ein Zinsobjekt mit lauter Nullen	252
Kapital, Prozent und Zinsen	253
Eine Datei für neue Klassen	255
Zusammenfassung	258
Ein paar Fragen	258
... und ein paar Aufgaben	259

12





13



Jetzt wird geOOpt	261
Mehr als ein Kreis.....	262
Bild ruft Datei.....	265
Figur statt nur Bild.....	267
Es bewegt sich was.....	270
Drehungen.....	273
Programm-Tuning.....	275
Zusammenfassung.....	277
Eine Frage.....	278
... und ein paar Aufgaben.....	278

14



Polymorphie	281
Von alten und neuen Methoden.....	282
Eine kleine Monsterfamilie.....	284
Mehr Schein als Sein?.....	286
Polymorphie.....	288
Monstershow.....	291
Überschreiben und überladen.....	293
Eines für alle?.....	296
Noch mal Polymorphie.....	298
Zusammenfassung.....	300
Ein paar Fragen.....	300
... aber keine Aufgaben.....	300

15



Buntes Allerlei	301
Wert oder Referenz?.....	302
Optionale Parameter.....	304
... und flexible Parameter.....	305
Welcher Fehler ist es?.....	307
Einfach nur Basic?.....	309
Zusammenfassung.....	312
Zum Schluss.....	313
Eine Frage.....	314
... und eine Aufgabe.....	314



Anhang A	315
Für Eltern	315
... und für Lehrer	316
 Anhang B	 319
Visual Basic installieren	319
Einsatz der Buchdateien	324
 Anhang C	 327
Kleine Checkliste	327
Dem Fehler auf der Spur	328
 Kleines OOP-Lexikon	 331
 Stichwortverzeichnis	 337

Für
Julia, Daniel, Katrin
und Janne



Vorwort

Es sind schon wahre Wunderwerke, die Computer! Man trifft diese Dinge überall, sie können allerhand Sachen, man kann damit Texte schreiben, Bilder malen und spielen.

In Wirklichkeit aber sind sie stohdumm. Denn sie machen nur das, was ihnen ein Programm sagt. Ist das ein Grafikprogramm, dann lassen sich mit dem Computer Bilder bearbeiten. Ist das ein Actionspiel, dann lässt sich der Computer mit einem Joystick oder den Tasten steuern und bewegt irgendwelche Figuren oder andere Objekte über den Bildschirm.

Programme werden von Leuten erstellt, die sich in einer Programmiersprache auskennen, sich Nächte um die Ohren schlagen und nur in der Nähe ihres Computers schlafen. So sagt man, aber das muss ja nicht stimmen.

Was heißt eigentlich Programmieren?

Wenn du aufschreibst, was ein Computer tun soll, nennt man das **Programmieren**. Das Tolle daran ist, dass du selbst bestimmen kannst, was getan werden soll. Lässt du dein Programm laufen, macht der Computer die Sachen, die du ausgeheckt hast. Natürlich wird er dann dein Zimmer nicht aufräumen und dir auch keine Tasse Kakao ans Bett bringen. Aber kannst du erst mal programmieren, kannst du den Computer sozusagen nach deiner Pfeife tanzen lassen.



Allerdings passiert es gerade beim Programmieren, dass der Computer nicht so will, wie du es gerne hättest. Meistens ist das ein Fehler im Programm. Das Problem kann aber auch irgendwo anders im Computer oder im Betriebssystem liegen. Das Dumme bei Fehlern ist, dass sie sich gern so gut verstecken, dass die Suche danach schon manchen Programmierer zur Verzweiflung gebracht hat.

Vielleicht hast du nun trotzdem Lust bekommen, das Programmieren zu erlernen. Dann brauchst du ja nur noch eine passende **Entwicklungsumgebung**, und schon kann's losgehen.

Was ist eine Entwicklungsumgebung?

Um ein Programm zu erstellen, musst du erst mal etwas eintippen. Das ist wie bei einem Brief oder einer Geschichte, die man schreibt. Das Textprogramm dafür kann sehr einfach sein, weil es ja nicht auf eine besondere Schrift oder Darstellung ankommt wie bei einem Brief oder einem Referat. So etwas wird **Editor** genannt.

Ist das Programm eingetippt, kann es der Computer nicht einfach lesen und ausführen. Jetzt muss es so übersetzt werden, dass der PC versteht, was du von ihm willst. Weil er aber eine ganz andere Sprache spricht als du, muss ein Dolmetscher her.

Du programmierst in einer Sprache, die du verstehst, und der Dolmetscher übersetzt es so, dass es dem Computer verständlich wird. So was heißt dann **Compiler** (ausgesprochen: Kompailer).

Schließlich müssen Programme getestet, überarbeitet, verbessert, wieder getestet und weiterentwickelt werden. Dazu gibt es noch einige zusätzliche Hilfen. Daraus wird dann ein ganzes System, die Entwicklungsumgebung.

Warum gerade Basic?

Leider kannst du nicht so programmieren, wie dir der Mund gewachsen ist. Eine **Programmiersprache** muss so aufgebaut sein, dass möglichst viele Menschen in möglichst vielen Ländern einheitlich damit umgehen können.



Weil in der ganzen Welt Leute zu finden sind, die wenigstens ein paar Brocken Englisch können, besteht auch fast jede Programmiersprache aus englischen Wörtern. Es gab auch immer mal Versuche, z.B. in Deutsch zu programmieren, aber meistens klingen die Wörter dort so künstlich, dass man lieber wieder aufs Englische zurückgreift.

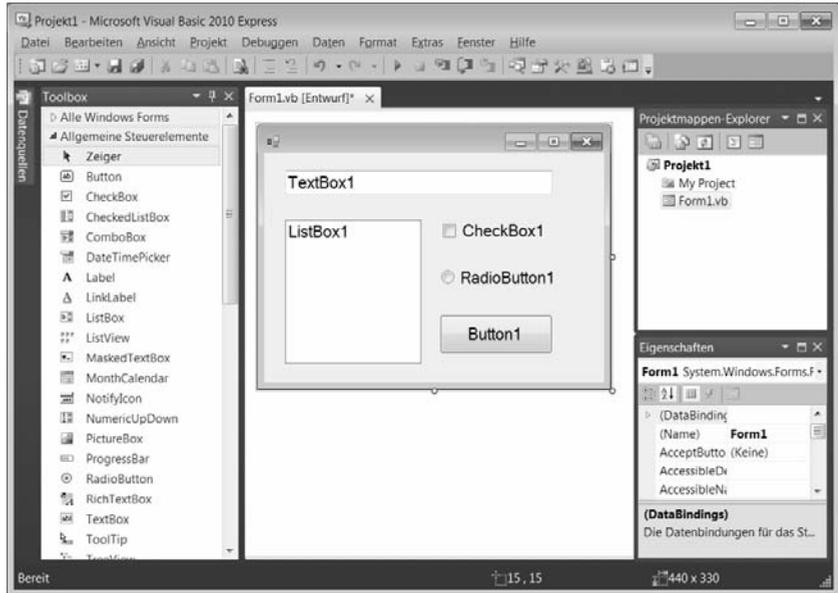
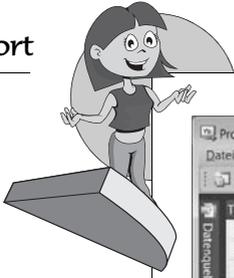
Eigentlich ist es egal, welche Programmiersprache du benutzt. Am besten eine, die möglichst leicht zu erlernen ist. In diesem Buch hast du es mit der Programmiersprache **BASIC** zu tun. Das ist die Abkürzung für den langen englischen Begriff »Beginner's All-purpose Symbolic Instruction Code«, der so viel bedeutet wie »Symbolische Allzweckprogrammiersprache für Anfänger«.

BASIC ist inzwischen eine der Sprachen, die am meisten verbreitet sind. Sie ist einfach und für Anfänger sehr gut geeignet, die zuvor noch nie mit irgendeiner Programmiersprache Kontakt hatten. (Willst du auch mal in andere Sprachen hineinschnuppern, dann empfehle ich dir eines der Kids-Bücher über Delphi, C++, C# oder Java.)

Da es eine ganze Reihe verschiedener Basic-Dialekte gibt, musste ich mir einen herausuchen. Besonders stark verbreitet und dazu noch vielseitig ist Visual Basic, weshalb ich mich auch hier für diesen Dialekt entschieden habe.

Visual Basic, das Entwicklungspaket zum Buch

Um den Kauf einer Entwicklungsumgebung für Basic musst du dich nicht weiter kümmern, wenn du dieses Buch erst mal besitzt. Auf der DVD zum Buch ist ein komplettes System für Basic enthalten. Mit der von Microsoft kostenlos zur Verfügung gestellten Software **Visual Basic** hast du eine sehr leistungsfähige Entwicklungsumgebung für die neuesten Versionen ab Windows XP.



Und was bietet dieses Buch?

Über eine ganze Reihe von Kapiteln verteilt lernst du

- ❖ Grundlagen der Sprache Basic kennen
- ❖ mit Visual Basic unter Windows umzugehen
- ❖ mit Komponenten zu arbeiten (das sind Bausteine, mit denen du dir viel Programmierarbeit sparen kannst)
- ❖ die Geheimnisse der Objektorientierten Programmierung (kurz: OOP) kennen

Im **Anhang** gibt es dann noch einiges an Informationen und Hilfen. Auch für deine Eltern und Lehrer, aber vor allem für dich!



Einleitung

Wie arbeitest du mit diesem Buch?

Grundsätzlich besteht dieses Buch aus einer Menge Text mit vielen Abbildungen dazwischen. Natürlich habe ich mich bemüht, alles so zuzubereiten, dass daraus lauter gut verdauliche Happen werden. Damit das Ganze noch genießbarer wird, gibt es zusätzlich noch einige Symbole, die ich dir hier gern erklären möchte:

Arbeitsschritte

- Wenn du dieses Zeichen siehst, heißt das: Es gibt etwas zu tun. Damit kommen wir beim Programmieren Schritt für Schritt einem neuen Ziel immer näher.
- Grundsätzlich lernt man besser, wenn man einen Programmtext selbst eintippt oder ändert. Aber nicht immer hat man große Lust dazu. Weil du alle Projekte im Buch von der Verlags-Homepage herunterladen kannst, findest du hinter einem Programmierschritt auch den jeweiligen Namen des Projekts oder einer Datei (z.B. PROJEKT1, GRAFIK1). Wenn du also das Projekt nicht selbst erstellen willst, kannst du stattdessen diese Datei aus dem Download verwenden.



Aufgaben

Am Ende eines Kapitels wirst du jeweils eine Reihe von Fragen und Aufgaben entdecken. Diese Übungen sind nicht immer ganz einfach, aber sie helfen dir, noch besser zu programmieren. Auch die Lösungen zu den Aufgaben kannst du dir von der Verlags-Homepage herunterladen. Du kannst sie dir alle im Editor von Windows oder auch in deinem Textverarbeitungsprogramm anschauen. Oder du lässt sie dir ausdrucken und hast sie dann schwarz auf weiß, um sie neben deinen PC zu legen. (Auch die Programme zu den Aufgaben liegen zum Download bereit.)

Notfälle



Vielleicht hast du irgendetwas falsch gemacht oder etwas vergessen. Oder es wird gerade knifflig. Dann fragst du dich, was du nun tun sollst. Bei diesem Symbol findest du eine Lösungsmöglichkeit. Notfalls kannst du aber auch ganz hinten im Anhang C nachschauen, wo ein paar Hinweise zur Pannenhilfe aufgeführt sind.

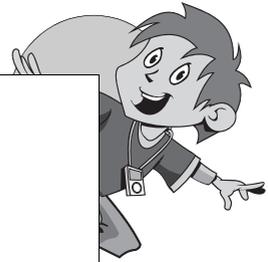
Wichtige Stellen im Buch



Hin und wieder siehst du ein solch dickes Ausrufezeichen im Buch. Dann ist das eine Stelle, an der etwas besonders Wichtiges steht.



Wenn es um eine ausführlichere Erläuterung geht, tritt Buffi in Erscheinung und schnuppert in seiner Kiste mit Tipps & Tricks.



Was brauchst du für dieses Buch?

Die DVD zum Buch

Du findest Visual Basic Express als komplette Entwicklungsumgebung für Windows-Programme auf der beiliegenden DVD. Installiert wird alles mit dem Programm SETUP in ein Verzeichnis deiner Wahl, z.B. C:\BASIC.

Die Beispielprogramme in diesem Buch findest du im Ordner BUCH. Falls du mal keine Lust zum Abtippen hast, kannst du sie also ebenso wie die Lösungen zu den Fragen und Aufgaben von dort auf deinen PC laden. Zu finden sind die Dateien auch auf der Homepage des Verlages in der gerade aktuellen Version:

<http://www.mitp.de/8680>

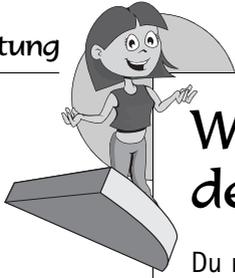
Betriebssystem

Die meisten Computer arbeiten heute mit dem Betriebssystem Windows. Davon brauchst du eine Version ab XP.

Übungsdisketten oder USB-Stick

Auf jeden Fall benötigst du mindestens eine Diskette oder z.B. einen USB-Stick, auch wenn du deine Programme auf die Festplatte speichern willst. Auf einem externen Speicher sind deine Arbeiten auf jeden Fall zusätzlich sicher aufgehoben.

Gegebenenfalls bitte deine Eltern oder Lehrer um Hilfe: Sie sollen den Anhang A (und vielleicht auch noch Anhang B) lesen. Dann können sie dir bei den ersten Schritten besser helfen.



Wie gut kannst du schon mit dem Computer umgehen?

Du musst dich mit deinem PC nicht perfekt auskennen, um mit Basic zu programmieren. Es ist aber gut zu wissen, wie man Visual Basic startet und beendet. Das erfährst du gleich im ersten Kapitel.



Wenn du noch Schwierigkeiten mit dem PC hast, ist es besser, sich erst mal ein grundlegendes Buch über Computer anzuschauen, z.B. PCs für Kids.

Kennst du dich aber schon gut mit dem Computer und mit Windows aus, dann lass uns beginnen!

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

1



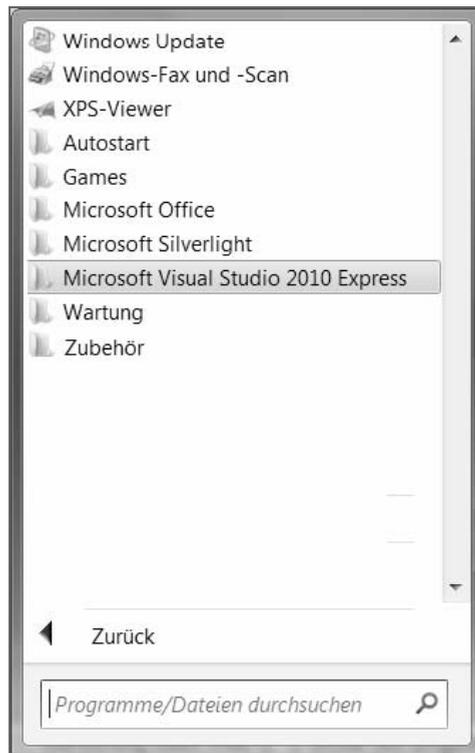
Visual Basic starten

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.:

Kleine Spritztour durch Visual Basic



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

oder

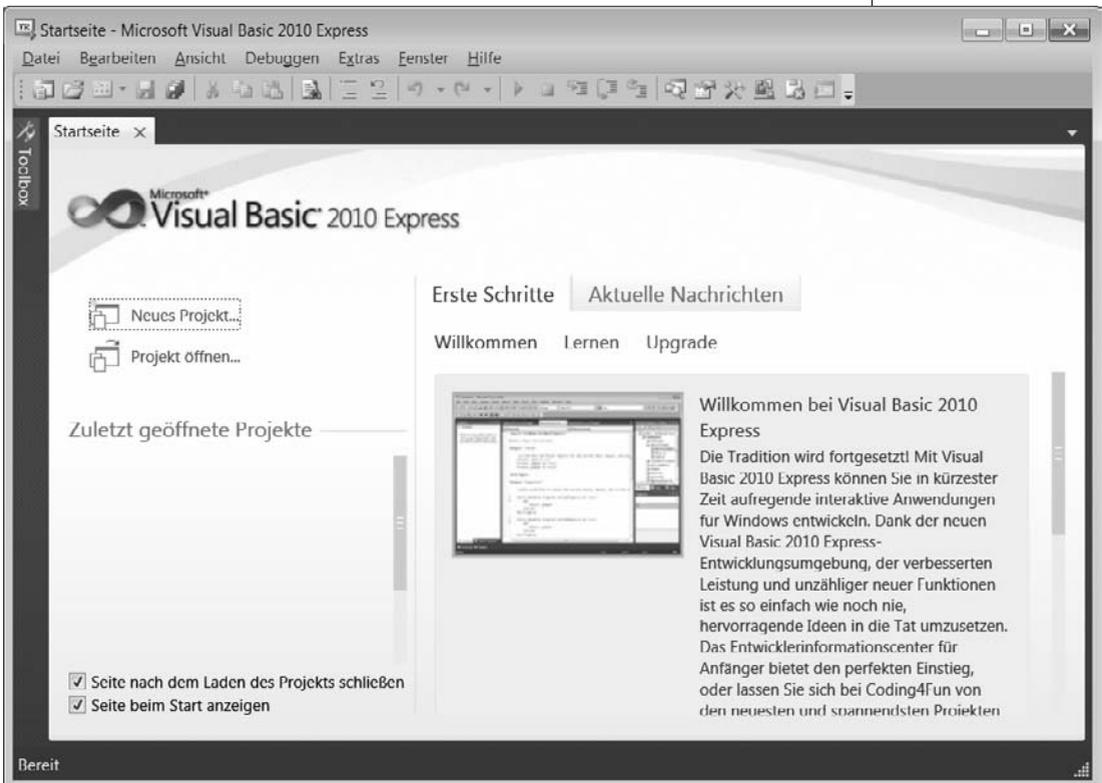
C:\Basic\Common7\IDE\vbexpress.exe

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:

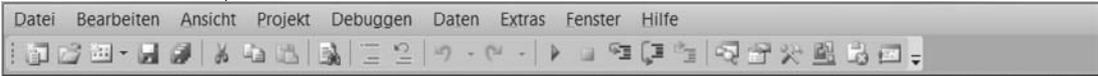


1



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 Programmprojekt ausgeführt wird.
- ❖ Und das HILFE-Menü bietet dir vielfältige Hilfe-Informationen an.

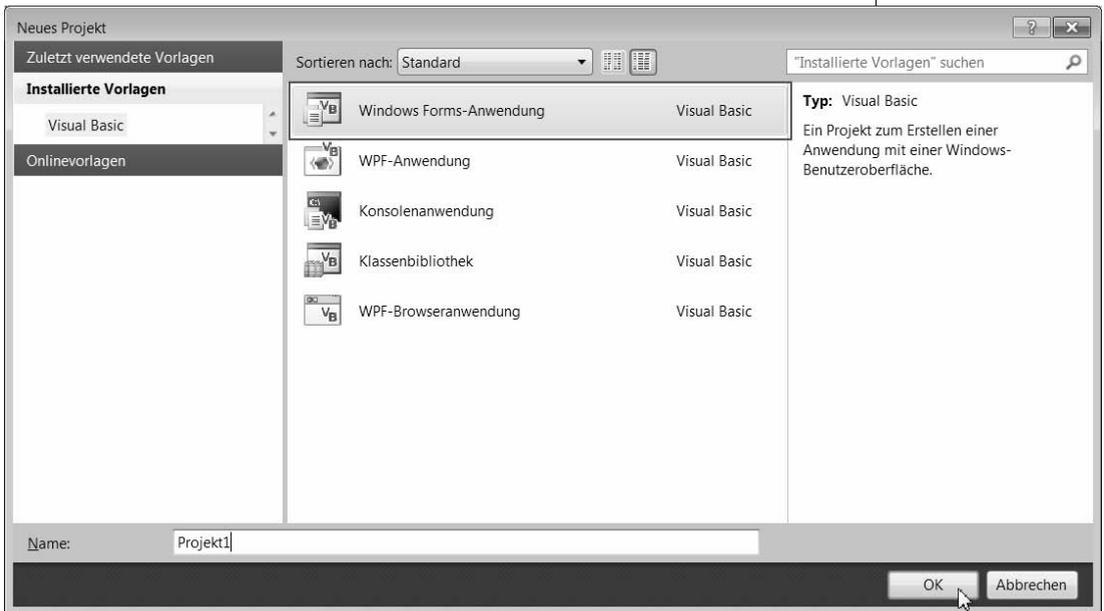
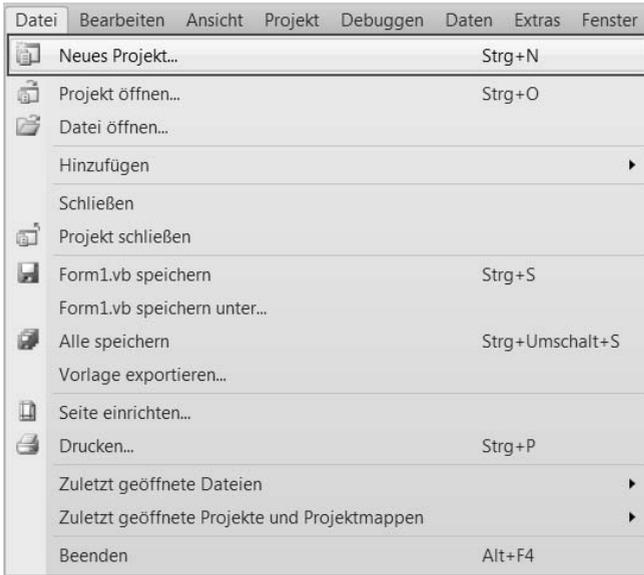
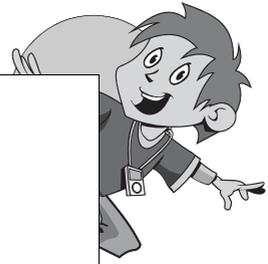


Einige wichtige Menüeinträge sind in einem so genannten **Popup-Menü** zusammengefasst. Das heißt so, weil es dort aufklappt, wo du gerade mit der rechten Maustaste hinklickst.

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.

Kleine Spritztour durch Visual Basic



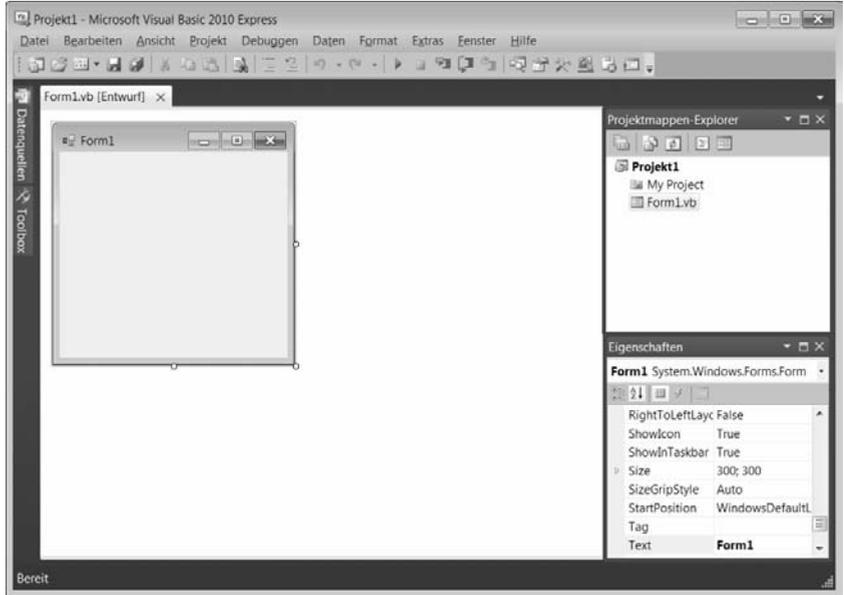
Wenn du willst, kannst du außerdem einen Namen für dein Projekt eingeben. Das ist beim ersten Versuch noch unwichtig, aber später solltest du schon einen Namen finden, der zum jeweiligen Projekt passt.

➤ Klicke zur Bestätigung auf OK.

1



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.

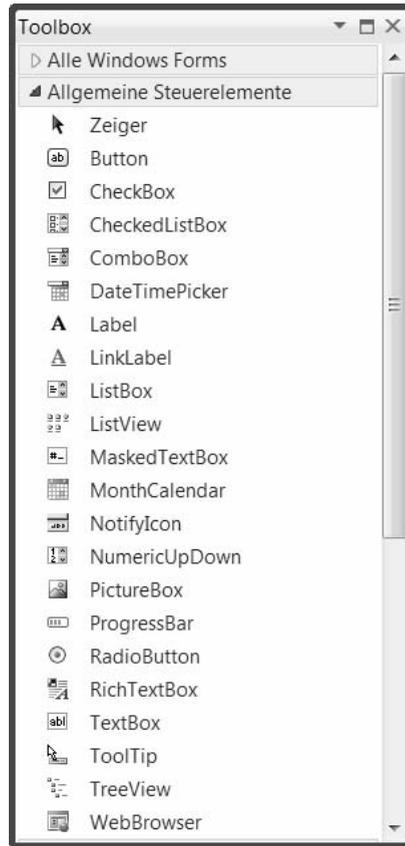


Die benötigten Zutaten finden sich direkt daneben. Wenn du ganz links mit dem Mauszeiger auf den Eintrag **TOOLBOX** fährst, öffnet sich ein (ellenlanges) Menü. Darin stehen die **Komponenten**, die Visual Basic dir zur Verfügung stellt.

Hallo auf Knopfdruck



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.



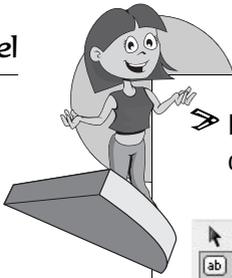
Hallo auf Knopfdruck

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.

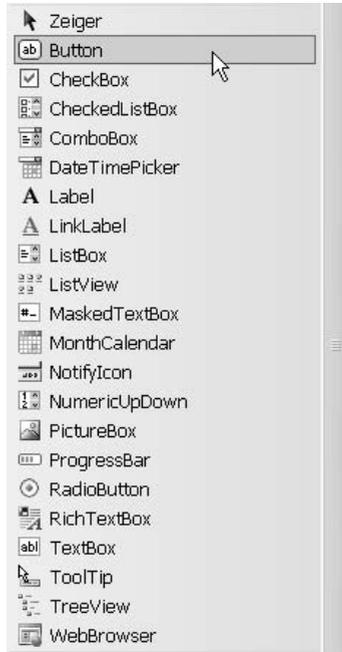
Wir brauchen dazu eine Komponente, die **Button** genannt wird. Man sagt dazu auch **Schaltfläche**. Und viele sprechen einfach von einem Knopf. Wichtig ist, dass man darauf mit der Maus klicken kann.

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

1

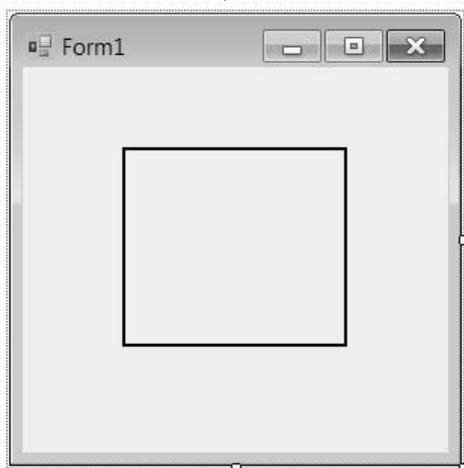


➤ Fahre mit der Maus langsam über die Einträge. Wenn du BUTTON gefunden hast, klicke darauf.



➤ 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