



Hans-Georg
Schumann



Visual Basic 2015

FÜR KIDS



Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)

Der Verlag räumt Ihnen mit dem Kauf des ebooks das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. 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 Einspeicherung und Verarbeitung in elektronischen Systemen.

Der Verlag schützt seine ebooks vor Missbrauch des Urheberrechts durch ein digitales Rechtemanagement. Bei Kauf im Webshop des Verlages werden die ebooks mit einem nicht sichtbaren digitalen Wasserzeichen individuell pro Nutzer signiert.

Bei Kauf in anderen ebook-Webshops erfolgt die Signatur durch die Shopbetreiber. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Hans-Georg Schumann



Visual Basic 2015 für Kids



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 978-3-95845-207-7

1. Auflage 2015

www.mitp.de

E-Mail: mitp-verlag@sigloch.de

Telefon: +49 7953 / 7189 - 079

Telefax: +49 7953 / 7189 - 082

© 2015 mitp-Verlags GmbH & Co. KG

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.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Lektorat: Katja Völpel

Sprachkorrektorat: Petra Heubach-Erdmann

Covergestaltung: Christian Kalkert

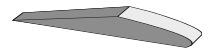
Satz: III-Satz, Husby, www.drei-satz.de

Inhalt

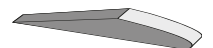


Vorwort	9
Einleitung	13
Das erste Projekt	17
Visual Studio starten	17
Kleine Spritztour durch Visual Studio	19
Hallo auf Knopfdruck	24
Eine Methode zum Drücken	28
Von »Drück mich« zu »Hallo«	34
Das Projekt speichern	35
Visual Studio beenden	36
Zusammenfassung	37
Ein paar Fragen	38
... aber noch keine Aufgabe	38
Buttons und Labels	39
Ein Projekt wieder öffnen	39
Noch ein Knopf	41
Titelleiste oder Label?	45
Gut oder schlecht?	48
Speichern und ausprobieren	50
Ereignisse und Methoden	50

1

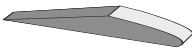


2



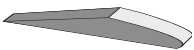
Veredelung	52
Ein ganz neues Projekt	54
Komponentenschwemme	55
Zusammenfassung	57
Ein paar Fragen	58
... und eine Aufgabe	58

3



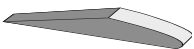
Von Zahlen, Zeichen und Operatoren	59
Die passende Optik	59
Alignment und Margin	61
Zufallszahlen und Variablen	66
Zeichenketten	70
Vereinbarungen sind nötig	70
Kommentare	72
Plus oder minus, mal oder durch	73
Zahlen mit Format	77
Zusammenfassung	79
Keine Fragen	80
... aber ein paar Aufgaben	80

4



Bedingungen	81
Von 1 bis 6	81
Wenn ... dann	86
Die If-Then-Struktur	88
Die Sache mit Try und Catch	89
Punkt für Punkt	93
Und und Oder, oder?	95
Zusammenfassung	97
Nur eine Frage	98
... und ein paar Aufgaben	98

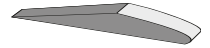
5



Ein Ratespiel mit Zahlen	99
Zu groß, zu klein	99
Es kann geraten werden	101
Dein PC zählt mit	105
Die If-Then-Else-Struktur	107
Neues Spiel oder wirklich Schluss?	109
Zusammenfassung	113

Eine Frage ...	114
... und eine Aufgabe	114
Kontrollstrukturen	115
Auf dem Weg zum Millionär	115
While oder Until?	117
Schleifenvariationen	120
If und Exit	122
Kleine Knopfparade	124
Diagnose auf Knopfdruck	126
Listenwahl	128
Von Fall zu Fall	134
Zusammenfassung	136
Ein paar Fragen ...	137
... und ein paar Aufgaben	137
Combo, Radio oder Check?	139
Alles in einer Box	139
Den XAML-Text ändern	146
Von Pünktchen ...	150
... und Häkchen	153
Antworten für die Optionsfelder	157
Wirklich fertig?	159
Variablenfelder und Startwerte	159
Die richtige Wahl	162
Die For-Next-Struktur	164
Zusammenfassung	167
Nur eine Frage ...	168
... und nur eine Aufgabe	168
Aktion Seelenklempner	169
Zwei Buttons und ein paar Gruppenfelder	170
TextBox, Label, ScrollBar	172
Vor der Sprechstunde	177
Bereit zur Diagnose	178
Schiebereien	179
Noch mehr Diagnosen?	181
Neue Sub – selbst gemacht	183

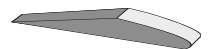
6



7

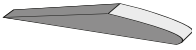


8



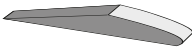
Datentransfer	185
Strings aus der Liste	187
Keine Sprechstunde?	188
Therapieprotokoll	190
Zusammenfassung	193
Ein paar Fragen	194
... und ein paar Aufgaben	194

9



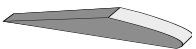
Menüs und Dialoge	195
Ein Menü für den Klempner	195
Menü-Methoden	202
Dialogfelder und Bibliotheken	205
Öffnen und Speichern	206
Sicherheitsabfrage	208
... und Schluss	213
Zusammenfassung	214
Ein paar Fragen	215
... aber keine Aufgabe	215

10



Einstieg in die Grafik	217
Von Punkten und Koordinaten	217
Das erste »Gebilde«	220
XAML oder Visual Basic?	225
Jetzt wird's bunt	228
Eigene Methode?	231
Eckig und rund	232
Farbtupfer	234
Noch ein Button?	237
Zusammenfassung	238
Ein paar Fragen	239
... und ein paar Aufgaben	239

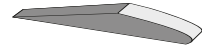
11



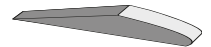
Animationen	241
Erst mal ein Kreis	242
Und es bewegt sich doch	244
Mehr Animationen?	246
Eine neue Klasse	249
Eigenschaften, Methoden und ein Konstruktor	251

Erscheinen, Bewegen, Verschwinden	254
Funktion oder Prozedur	256
Zusammenfassung	258
Zwei Fragen	260
... und zwei Aufgaben	260
Kapselung und Vererbung	261
Alles unter einem Hut	262
Es gibt was zu erben	264
Ein neues Baby?	265
Ein Zinsobjekt mit lauter Nullen	269
Kapital, Prozent und Zinsen	271
Eine Datei für neue Klassen	273
Erbfolge	276
Zusammenfassung	278
Ein paar Fragen	279
... und eine Aufgabe	279
Jetzt wird geOOPT	281
Mehr als ein Kreis	281
Bild aus Datei	284
Movelmage statt MoveCircle	286
Figur statt nur Bild	288
Die Zeit tickt	291
Es bewegt sich was	295
Zusammenfassung	298
Zwei Fragen	299
... und ein paar Aufgaben	299
Polymorphie	301
Von alten und neuen Methoden	301
Eine kleine Monsterfamilie	304
Mehr Schein als Sein?	306
Polymorphie	308
Monstershow	311
Überschreiben und überladen	313
Eines für alle?	315
Noch mal Vielgestalt?	318

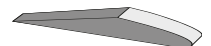
12



13



14



15

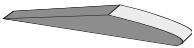


Zusammenfassung	319
Ein paar Fragen	320
... aber keine Aufgaben	320

Einblick in die Spieleprogrammierung 321

Spielfeld und Spielfigur	321
Game-New	324
Game-Run	327
Das richtige Timing	330
Ball oder Käfer?	331
Tastensteuerung	333
Drehmoment	335
Maussteuerung	338
Winkelfunktionen	341
Zusammenfassung	344
Zum Schluss	345
Keine Fragen	345
... aber noch zwei Aufgaben	345

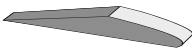
A



Anhang A 347

Für Eltern	347
... und für Lehrer	348

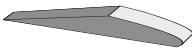
B



Anhang B 351

Visual Studio installieren	351
Einsatz der Buch-Dateien	355

C



Anhang C 357

Kleine Checkliste	357
Dem Fehler auf der Spur	358

Stichwortverzeichnis 363



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 strohdumm. 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 öfter mal, dass der Computer nicht so will, wie du es gerne hättest. Meistens ist das ein Feh-

ler 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**.

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. Und ein solches (sehr mächtiges) System ist **Visual Studio** von Microsoft.

Warum gerade (Visual) 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.

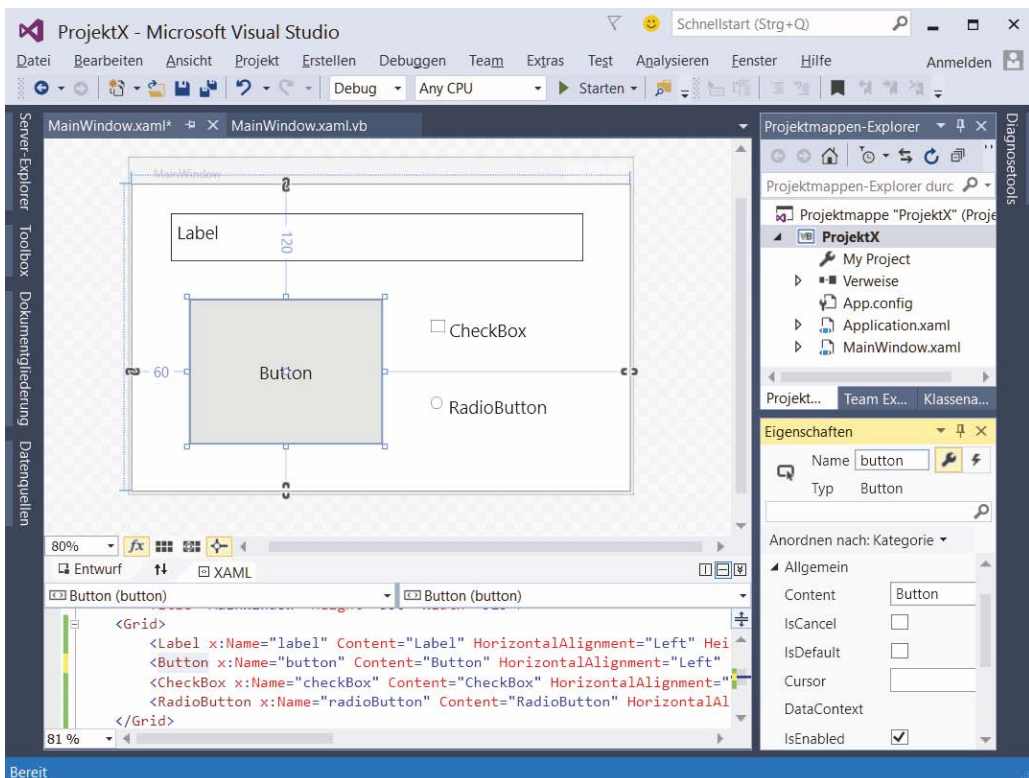
Warum gerade [Visual] Basic?

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, die zuvor noch nie mit irgendeiner Programmiersprache Kontakt hatten, sehr gut geeignet. (Willst du auch mal in andere Sprachen hineinschnuppern, dann empfehle ich dir eines der anderen Kids-Bücher z.B. über Delphi 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.

Mit der von Microsoft kostenlos zur Verfügung gestellten Software **Visual Studio** hast du eine sehr leistungsfähige Entwicklungsumgebung für die neuesten Windows-Versionen.



Und was bietet dieses Buch?

Über eine ganze Reihe von Kapiteln verteilt lernst du

- ◇ Grundlagen der Sprache Basic kennen
- ◇ mit Visual Studio unter Windows umzugehen
- ◇ mit Komponenten zu arbeiten (das sind Bausteine, mit denen du dir viel Programmierarbeit sparen kannst)
- ◇ einiges über Grafik, Animation und Spiele
- ◇ 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 sich alle Projekte im Buch von der Verlags-Homepage herunterladen lassen, findest du hinter einem Programmierschritt auch den jeweiligen Namen des Projekts oder einer Datei (z.B. PROJEKT1, GAME1). Wenn du also das Projekt nicht selbst erstellen willst, kannst du stattdessen die 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 Auf-

gaben kannst du dir von der Verlags-Homepage herunterladen: <http://www.mitp.de/206>. Sie lassen sich z.B. 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?

Du findest Visual Studio 2015 als komplette Entwicklungsumgebung für Windows-Programme zum Download auf der entsprechenden Seite von Microsoft:

<https://www.visualstudio.com/de-de/>

Kostenlos ist die **Community**-Version, damit arbeiten wir auch hier im Buch. Nach dem Download wird alles mit dem **Setup**-Programm in ein Verzeichnis deiner Wahl installiert, z.B. C:\PROGRAMME\VISUAL STUDIO.

Wie gut kannst du schon mit dem Computer umgehen?

Die Beispielprogramme in diesem Buch findest du auf der Homepage des Verlages in der gerade aktuellen Version:

<http://www.mitp.de/206>

Betriebssystem

Die meisten Computer arbeiten heute mit dem Betriebssystem Windows. Davon brauchst du eine der Versionen 7, 8 oder 10.

USB-Stick oder SD-Card

Auf jeden Fall benötigst du einen USB-Stick oder ein anderes externes Speichermedium, auch wenn du deine Programme auf der Festplatte unterbringen willst. In einem zusätzlichen 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 Studio 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. **Computer für Kids**.



Kennst du dich aber schon gut mit dem Computer und mit Windows aus, dann lass uns gleich 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 Studio.

In diesem Kapitel lernst du

- ⊙ wie man Visual Studio 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 Studio beendet

Visual Studio starten

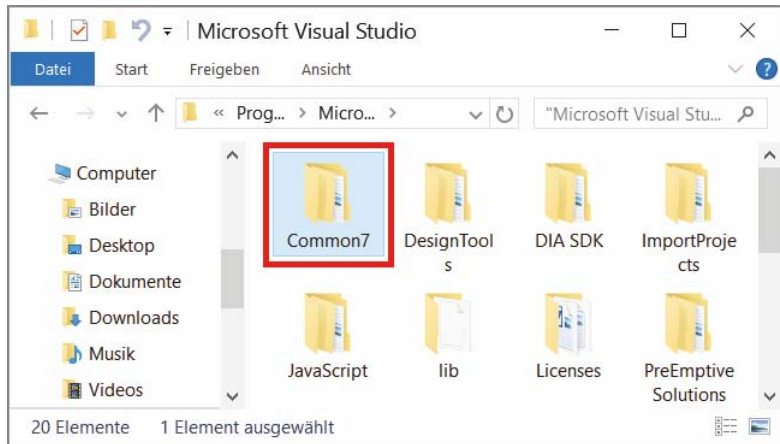
Bevor wir mit dem Programmieren anfangen können, muss Visual Studio erst installiert werden.

1

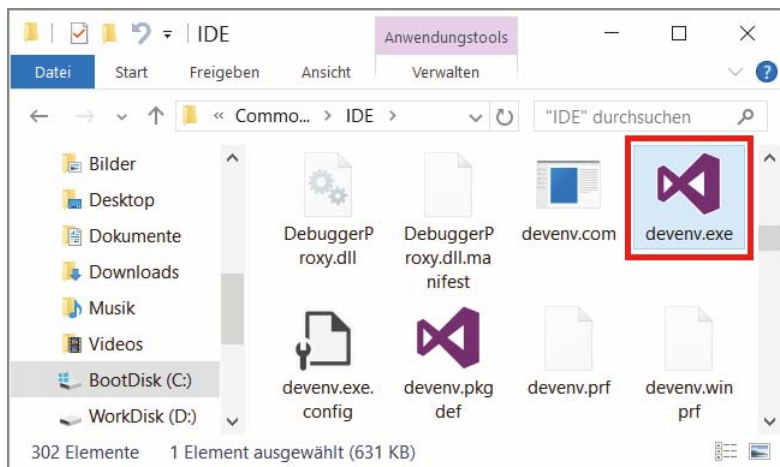
Die Installation übernimmt ein Programm namens SETUP. Genaues erfährst du im **Anhang B**. Hier musst du dir von jemandem helfen lassen, wenn du dir die Installation nicht allein zutraust.

Eine Möglichkeit, Visual Studio zu starten, ist diese:

- Öffne den Ordner, in den du Visual Studio untergebracht hast (z.B. C:\PROGRAMME\VISUAL STUDIO).



- Dort musst du nun weiter in einige Unterordner mit den Namen COMMON7\IDE wechseln:



- Hier suchst du unter den vielen Symbolen eines derjenigen heraus, die aussehen wie eine gekippte Lila-8, und zwar das mit dem Namen DEVENV.EXE.

- Nun kannst du das Programm mit einem Doppelklick auf das Symbol starten:



Ich empfehle dir, eine **Verknüpfung** auf dem Desktop anzulegen:

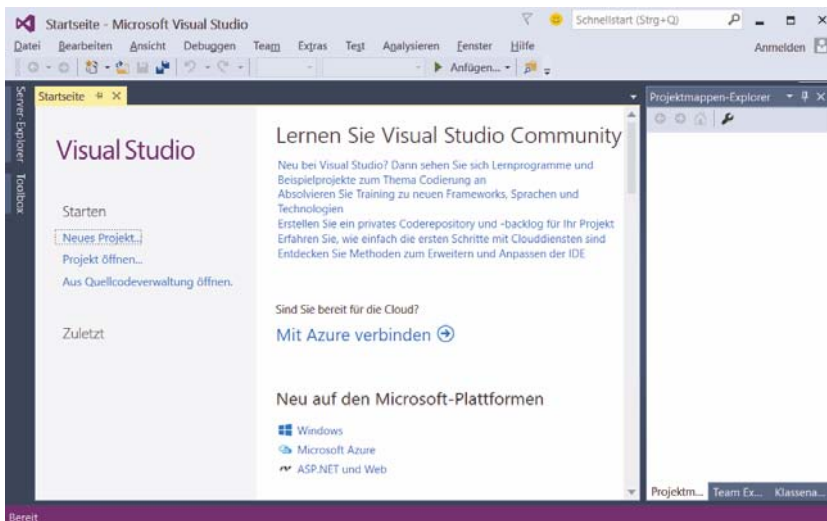
- ❖ Dazu klickst du mit der rechten Maustaste auf das Symbol für Visual Studio (DEVENV.EXE). Im Kontextmenü wählst du KOPIEREN.
- ❖ Dann klicke auf eine freie Stelle auf dem Desktop, ebenfalls mit der rechten Maustaste. Im Kontextmenü wählst du VERKNÜPFUNG EINFÜGEN.
- ❖ Es ist sinnvoll, für das neue Symbol auf dem Desktop den Text devenv.exe - Verknüpfung durch Visual Studio zu ersetzen.

Von nun an kannst du auf das neue Symbol **doppelklicken** und damit Visual Studio starten.



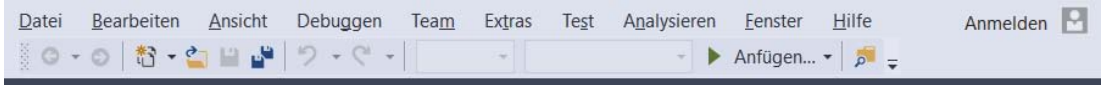
Kleine Spritztour durch Visual Studio

Je nach Computer kann es eine Weile dauern, bis Visual Studio 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 Studio wirst du wahrscheinlich am meisten benutzen:

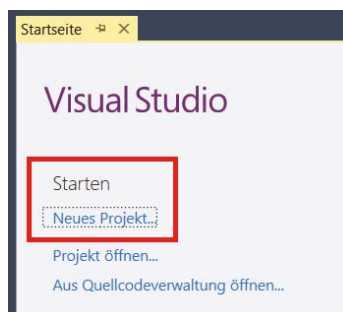
- ❖ Über das DATEI-Menü kannst du Dateien speichern, laden (öffnen), ausdrucken, neu erstellen oder Visual Studio beenden.
- ❖ Das BEARBEITEN-Menü hilft dir bei der Bearbeitung deines Programmtextes, aber auch bei anderen Programmelementen. Außerdem kannst du dort bestimmte Arbeitsschritte rückgängig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



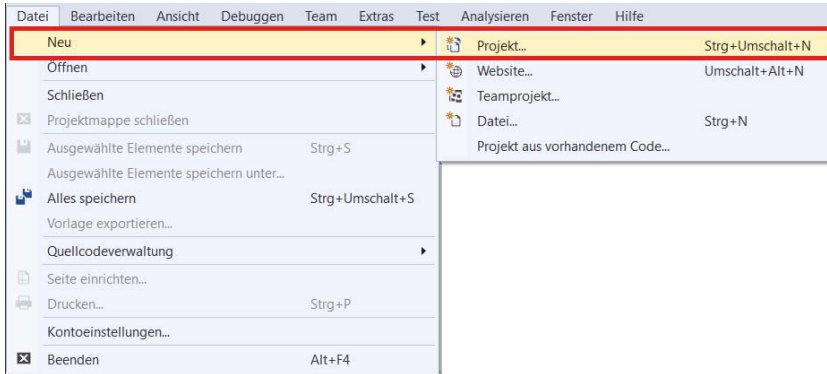
Einige wichtige Menüeinträge sind in einem sogenannten **Popup-Menü** zusammengefasst. Das heißt so, weil es dort aufklappt, wo du gerade mit der rechten Maustaste hinklickst.

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!

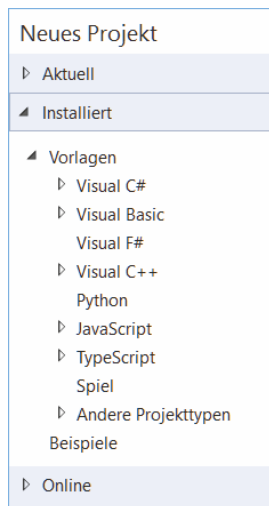
➤ Klicke unter STARTEN auf NEUES PROJEKT.



- Oder du klickst in der Menüleiste auf DATEI und im sich öffnenden Menü auf NEU und dann auf PROJEKT.

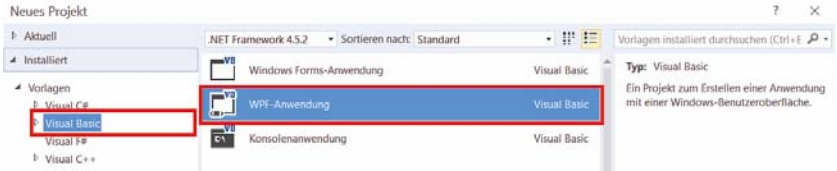


Im folgenden Dialogfenster wird dir nun einiges angeboten. Das merkst du, wenn du auf der linken Seite mal einzelne Einträge anklickst. Eigentlich so viel, dass es schon ein bisschen verwirrend sein kann. Wie du siehst, gibt es nicht nur **mehrere** Sprachen, mit denen Visual Studio funktioniert, sondern du kannst deine Projekte auch für andere **Plattformen** (Geräte und Betriebssysteme) als Windows-Computer entwickeln.



- Suche links den Eintrag VISUAL BASIC und klicke darauf. Das ist wichtig, damit dein Projekt auch in dieser Sprache erstellt werden kann. Danach klickst du auf das Symbol für WPF-ANWENDUNG.

1



WPF, was ist das? Zunächst ist es die Abkürzung für »Windows Presentation Foundation«. Gemeint ist damit ein sogenanntes **Framework**, ein Programmiergerüst, das einen umfangreichen Vorrat an Elementen anbietet, um eine grafische Oberfläche zu gestalten. Man kann auch sagen: um ein Hauptfenster mit Inhalt zu füllen, und zwar z. B. mit Komponenten und Multimedia-Objekten.

Neben WPF gibt es noch die sogenannten **Windows Forms**. Die waren die Vorläufer des WPF, noch heute kann man aber mit Visual Basic auch Projekte erstellen, die mit Windows Forms funktionieren.

➤ Wenn du willst, kannst du außerdem einen Namen für dein Projekt eingeben – z. B. Projekt1.

Das ist beim ersten Versuch noch unwichtig, aber später solltest du schon einen Namen finden, der zum jeweiligen Projekt passt.

➤ Durchaus wichtig ist der **SPEICHERORT**. Du kannst die Vorgabe so stehen lassen, solltest dir aber dann merken, wo deine Projekte sind.

Ich habe auf Laufwerk D: einen Ordner **VISUAL BASIC** erstellt, in den packe ich meine Projekte.

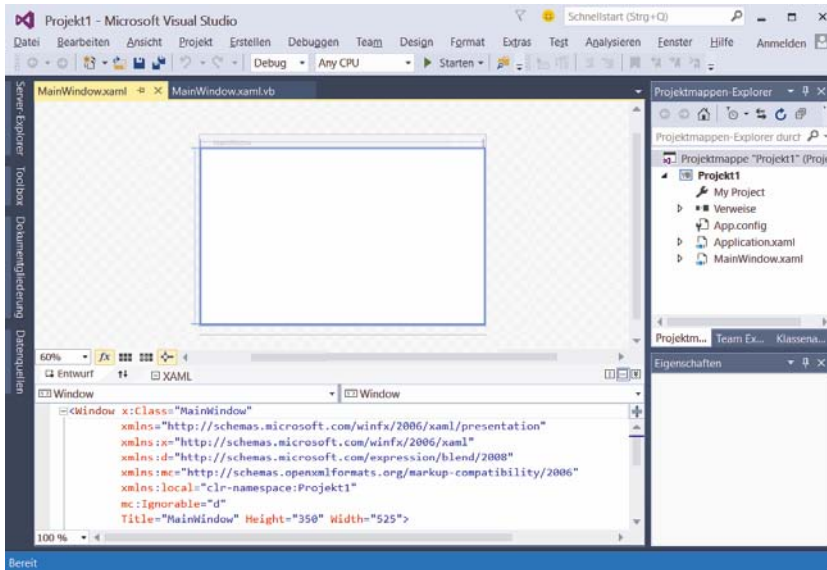


➤ Klicke schließlich zur Bestätigung auf **OK**.

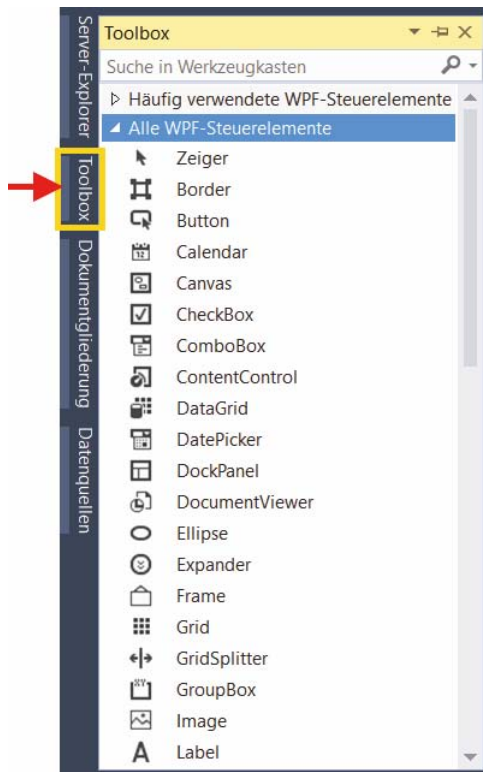
Es kann etwas dauern, bis sich schließlich das Erscheinungsbild von Visual Studio etwa so geändert hat:

Das große Fenster links oben ist das Designfenster und trägt den Titel **MAINWINDOW.XAML**. Was du darin siehst, ist das sogenannte **Formular** oder auch **Projekt-Hauptfenster**, in dem wir unsere Programmoberfläche zusammenbasteln.

Kleine Spritztour durch Visual Studio



Darunter in einem schmalen Fenster steht einiger Text. Der beschreibt das darüber liegende Formular. Damit beschäftigen wir uns erst später.



1



Die benötigten Zutaten finden sich direkt daneben. Wenn du ganz links am Rand mit dem Mauszeiger auf den Eintrag **TOOLBOX** klickst, öffnet sich ein (ellenlanges) Menü. Darin stehen die **Komponenten**, die Visual Studio 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.

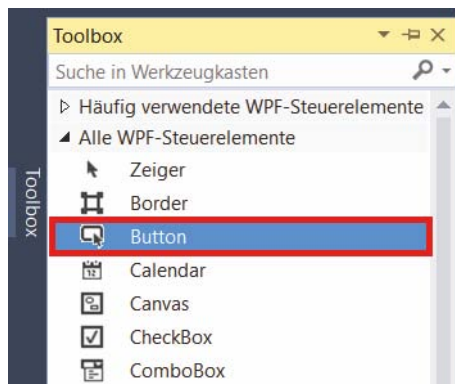
Bei dir gibt es nichts mit Toolbox? Dann öffne das Menü **ANSICHT**, suche dort den Eintrag **TOOLBOX** und klicke darauf.

Hallo auf Knopfdruck

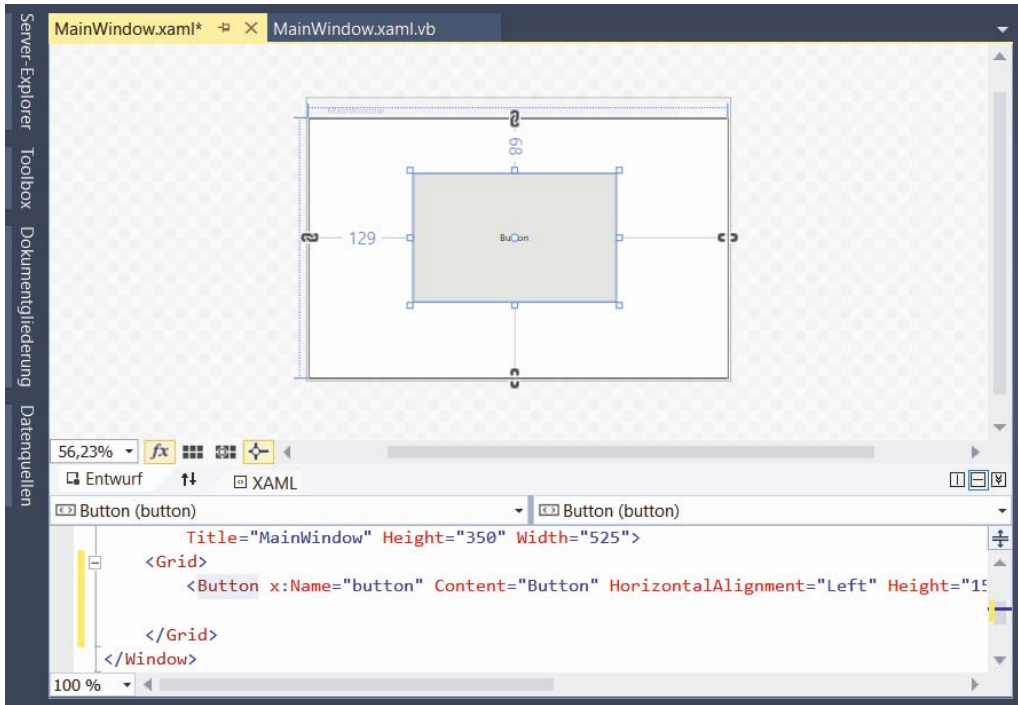
Eigentlich kann's jetzt schon losgehen. Den Umgang mit Menüs und Dialogfenstern kennst du bereits von Windows. Deshalb müssen wir uns damit nicht mehr aufhalten. Bauen wir uns jetzt ein kleines Projekt, das auf Knopfdruck funktioniert.

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.

➤ Fahre mit der Maus langsam über die Toolbox-Einträge. Wenn du bei **BUTTON** gelandet bist, klicke darauf.

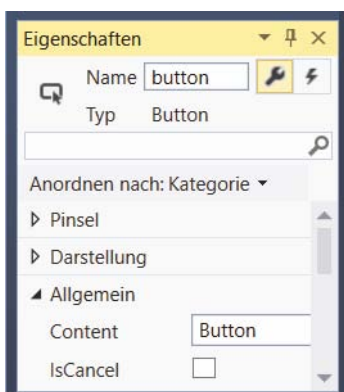


➤ Wechsle mit dem Mauszeiger nun in das Formular. Drücke die linke Maustaste, halte sie gedrückt und ziehe mit der Maus schräg nach unten. (Die Toolbox-Anzeige wird zugeklappt.)



➤ Wenn du meinst, dass die Schaltfläche groß genug ist, lass die Maus-taste wieder los.

Da hast du nun eine Schaltfläche (oder einen Knopf) mit der Aufschrift **BUTTON**. Eigentlich ein blöder Name! Den sollten wir gleich ändern. Dazu brauchen wir ein Fensterchen, das wir bisher noch 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. Du findest dieses Fenster rechts unten in Visual Studio.



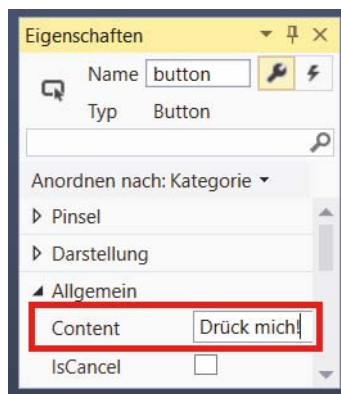
1



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 Studio Objekte. Die sind natürlich nur künstlich. So ein Objekt ist beispielsweise das Formular, und auch der Knopf, den du gerade dort hineingesetzt hast, ist ein Objekt. Alle Komponenten sind Objekte.

Dabei kann es auch in Visual Studio 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.

- Suche unter ALLGEMEIN den Eintrag CONTENT, klicke auf die Fläche dahinter und lösche den Text Button. Dann tippe dort ein: Drück mich!

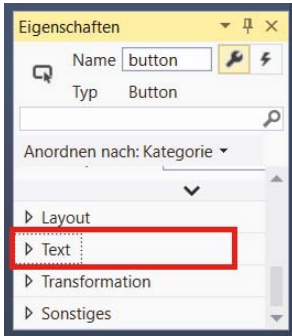


Wenn du zweimal auf eine Komponente klickst (kein Doppelklick!), dann erscheint der dortige Text markiert. Du kannst ihn dann durch direktes Eintippen ändern.

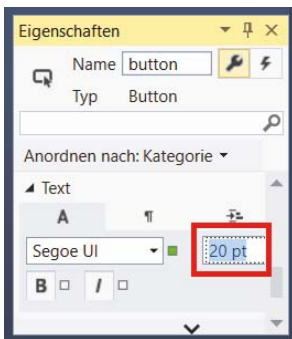
Und schon erscheint dieser Text auch als Aufschrift auf der Schaltfläche. Allerdings sieht er genauso mickrig aus wie der vorige Text. Auch da lässt sich was über das EIGENSCHAFTEN-Fenster machen.

- Blättere dich durch das Fenster, bis du zu dem Eintrag TEXT gelangst. Klicke darauf.

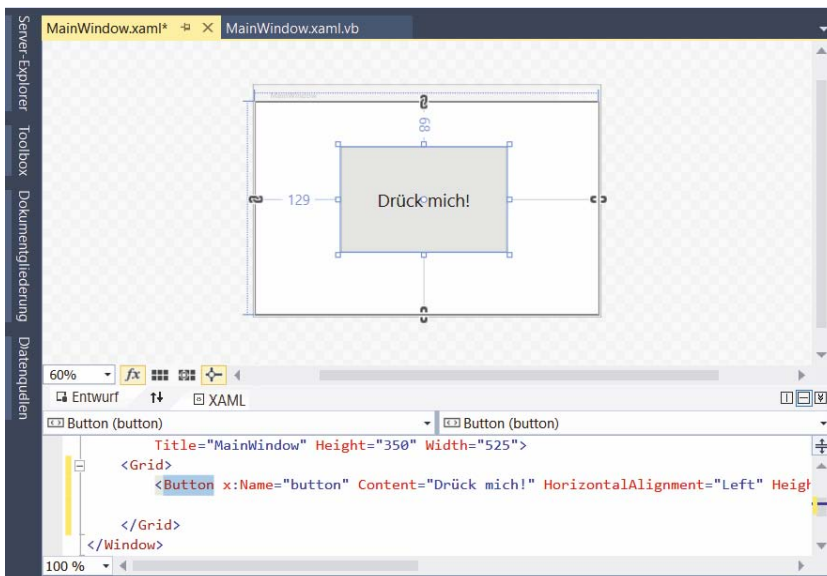
Hallo auf Knopfdruck



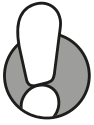
- Nun kannst du nicht nur eine neue Schriftart wählen (wenn du willst), sondern auch die Größe einstellen. Dort, wo eine Zahl mit »pt« steht (bei mir 9). Ändere das in einen Wert zwischen 16 und 24 um.



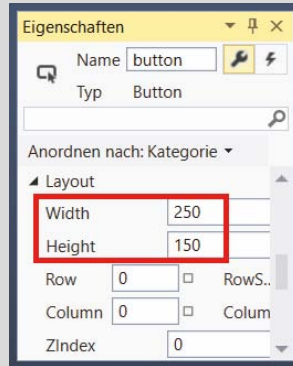
Und jetzt kann sich der Button schon besser sehen lassen.



1



Wenn du willst, kannst du dich im EIGENSCHAFTEN-Fenster noch ein bisschen weiter umschauen. So lassen sich z.B. über LAYOUT u.a. die Werte für die Maße der Schaltfläche ändern.



Eine Methode zum Drücken

Nun wäre es toll, wenn unser Programm auf den Knopfdruck einen Gruß losschickt.

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

Und ein neues Fenster kommt zum Vorschein. Es trägt den Titel MAIN-WINDOW.XAML.VB. Das ist das **Editorfenster** von Visual Studio, dort findest du den Quelltext deines Programms, also die Anweisungen und Funktionen, die vom Computer ausgeführt werden sollen. (Man kann also auch von Quelltext-Fenster sprechen.)

