

# Übersicht

<b>Vorwort und Einleitung</b>	<b>17</b>
<b>1 Die ersten Schritte mit Visual Basic 2010</b>	<b>23</b>
<b>2 Rundgang durch die Visual Basic 2010-IDE</b>	<b>57</b>
<b>3 Formulare und Steuerelemente</b>	<b>73</b>
<b>4 Die Befehle von Visual Basic 2010</b>	<b>91</b>
<b>5 Programmierung mit Objekten</b>	<b>131</b>
<b>6 Streifzug durch die .NET-Klassenbibliothek</b>	<b>159</b>
<b>7 Collections und LINQ</b>	<b>191</b>
<b>8 Formulare und Steuerelemente für etwas Fortgeschrittene</b>	<b>207</b>
<b>9 Benutzeroberflächen gestalten</b>	<b>245</b>
<b>10 Fehler abfangen und aufspüren</b>	<b>271</b>
<b>11 Multithreading mit Visual Basic</b>	<b>287</b>
<b>12 Visual Basic und die Datenbanken</b>	<b>305</b>
<b>13 WPF mit Visual Basic 2010</b>	<b>353</b>
<b>14 Von der Idee zum fertigen Windows-Programm</b>	<b>387</b>
<b>Anhang A: Das kleine .NET-Lexikon</b>	<b>419</b>
<b>Anhang B: Visual Basic 2010-Befehlswörter</b>	<b>423</b>
<b>Anhang C: Lösungen zu den Übungsaufgaben</b>	<b>435</b>
<b>Stichwortverzeichnis</b>	<b>457</b>

# Inhaltsverzeichnis

<b>Vorwort und Einleitung</b>	<b>17</b>
<b>1 Die ersten Schritte mit Visual Basic 2010</b>	<b>23</b>
1.1 Eine Checkliste für den Start	24
1.1.1 Die Installation im Detail	25
1.1.2 Ein erster Probestart	27
1.1.3 Visual Basic 2010 Express registrieren	28
1.2 Der Idealgewichtsrechner als erstes Beispielprojekt	30
1.2.1 Version 1 – der Idealgewichtsrechner berechnet ein Idealgewicht	31
1.2.2 Version 2 – der Idealgewichtsrechner überprüft die Eingabe	35
1.2.3 Version 3 – der Idealgewichtsrechner unterscheidet zwischen Männlein und Weiblein	36
1.2.4 Version 4 – der Idealgewichtsrechner zeigt Bildchen an	39
1.2.5 Version 5 – der Idealgewichtsrechner berechnet den BMI	41
1.2.6 Version 6 – der Idealgewichtsrechner merkt sich Einstellungen	49
1.2.7 Version 7 – der Idealgewichtsrechner lernt sprechen	51
1.3 Den Idealgewichtsrechner weitergeben	55
1.3.1 Die Rolle der Programmdatei	55
1.4 Zusammenfassung	56
1.5 Fragen zum Abschluss des Kapitels	56

<b>2</b>	<b>Rundgang durch die Visual Basic 2010-IDE</b>	<b>57</b>
2.1	Der erste Start von Visual Basic 2010	58
2.1.1	Die Rolle der Projekte	58
2.1.2	Der Projektmappen-Explorer	59
2.1.3	Ein vorhandenes Projekt öffnen	60
2.1.4	Ein neues Projekt anlegen	61
2.1.5	Projekte starten, anhalten und beenden	61
2.1.6	Projekte speichern	62
2.1.7	Listings ausdrucken	63
2.1.8	Projekteigenschaften	64
2.1.9	Der Aufbau eines Visual Basic-Projekts	64
2.2	Die IDE als Nachhilfelehrer	66
2.3	Der Programmcode-Editor	67
2.4	Die IDE nach eigenem Geschmack anpassen	68
2.4.1	Schriftart und Schriftgröße einstellen	69
2.4.2	Farben des Hintergrunds und der Schrift einstellen	69
2.4.3	Zeilenumbruch und Zeilennummern einschalten	69
2.4.4	Menübefehle mit eigenen Tastenkombinationen belegen	69
2.4.5	Das Extras-Menü um neue Einträge erweitern	70
2.4.6	Funktionen, die es bei Visual Basic Express nicht gibt	70
2.5	Zusammenfassung	71
2.6	Fragen zum Abschluss des Kapitels	71
<b>3</b>	<b>Formulare und Steuerelemente</b>	<b>73</b>
3.1	Die Anatomie eines Formulars	74
3.2	Steuerelemente und die Toolbox	74
3.2.1	Steuerelemente auf dem Formular anordnen	75
3.2.2	Die Rolle der Aufgabenliste	75
3.2.3	Die Rolle der Komponenten	76
3.2.4	Die Toolbox anpassen	77
3.3	Das Eigenschaftenfenster	78
3.4	Steuerelemente positionieren	79
3.4.1	Ein Blick in die Designerdatei eines Formulars	79
3.5	Ereignisse und Ereignisprozeduren	80
3.6	Tipps zu Formularen und Steuerelementen	82
3.7	Die wichtigsten Steuerelemente (für den Anfang)	82
3.7.1	Das Label	83
3.7.2	Der Button	83
3.7.3	Die TextBox	84
3.7.4	Der RadioButton	84

3.7.5	Die CheckBox	85
3.7.6	Die ListBox	86
3.7.7	Die ComboBox	87
3.7.8	Die PictureBox	87
3.7.9	Der Rauf- und Runterzähler (NumericUpDown)	88
3.8	Zusammenfassung	89
3.9	Fragen zum Abschluss des Kapitels	89
<b>4</b>	<b>Die Befehle von Visual Basic 2010</b>	<b>91</b>
4.1	Schritt für Schritt oder wie wird ein Computerprogramm ausgeführt?	92
4.1.1	Computerprogramm = Befehle, Befehle und noch mehr Befehle	92
4.1.2	Fehler in einem Programm – die Bugs lauern überall	93
4.2	Variablen als das Gedächtnis des Computers	93
4.2.1	Variablen besitzen Werte	94
4.2.2	Variablen und ihr Anfangswert	95
4.3	Im Rechnen unschlagbar? Die Operatoren	96
4.3.1	Ein erstes Beispiel – ein kleiner Mehrwertsteuerrechner	97
4.4	Computerprogramme treffen Entscheidungen	99
4.4.1	Bedingungen verknüpfen – Mr. Boole lässt grüßen	101
4.4.2	Sind Sie volljährig oder bist Du es noch nicht? Der Computer findet es heraus	102
4.4.3	Der Mehrwertsteuerrechner wird erweitert	106
4.4.4	Mehrfachabfragen mit Select Case	106
4.5	Wiederholungen erwünscht – die Programmschleifen	108
4.5.1	Die For Next-Schleife – die Anzahl der Durchläufe steht fest	108
4.5.2	Wiederholen, bis eine Abbruchbedingung erfüllt ist	109
4.5.3	Die While End While-Schleife – wie Do Loop, nur ein wenig einfacher	109
4.5.4	Die For Each-Schleife – Spezialist für Collections	110
4.5.5	Warten auf Godot – die Endlosschleife	110
4.6	Befehle zu Gruppen zusammenfassen – die Funktionen und Prozeduren	111
4.6.1	Prozeduren definieren	112
4.6.2	Variablendeklarationen innerhalb einer Prozedur – die lokalen Variablen	112
4.6.3	Prozeduren mit Parameter	112
4.6.4	Funktionen definieren	114

## Inhaltsverzeichnis

4.6.5	Statische Variablen	115
4.7	Wie schnell ist der Computer?	115
4.7.1	Den Ausweis bitte – Prüfsummen bilden	118
4.8	Feldvariablen (Arrays)	120
4.8.1	Arrays definieren	120
4.8.2	Einen Wert in einem Array ansprechen	121
4.8.3	Die Größe eines Arrays dynamisch ändern	121
4.8.4	Arrays Element für Element durchlaufen	122
4.9	Etwas fortgeschrittenere Themen der Visual Basic- Programmierung	123
4.9.1	Die Rolle der Datentypen	123
4.9.2	Übergaben von Argumenten als Referenz	125
4.9.3	Eine beliebige Anzahl an Parametern übergeben mit einem ParamArray	126
4.9.4	Zeichenkettenoperationen mit der String-Klasse	126
4.10	Zusammenfassung	128
4.11	Fragen zum Abschluss des Kapitels	128
<b>5</b>	<b>Programmierung mit Objekten</b>	<b>131</b>
5.1	Was ist ein Objekt?	132
5.1.1	Auf zur objektorientierten Programmierung	133
5.2	Eine Klasse wird definiert	133
5.2.1	Eine Klasse besitzt Mitglieder	133
5.2.2	Felder definieren	134
5.3	Eigenschaften definieren	134
5.4	Aus der Klasse wird ein Objekt	135
5.4.1	Ich instanziiere, also bin ich – das Schlüsselwort Me	136
5.4.2	Die Rolle des Konstruktors	136
5.4.3	Eigenschaften direkt einen Wert zuweisen	137
5.5	Methoden definieren und überladen – was steckt hinter dem Gültigkeitsbereich?	137
5.5.1	Methoden überladen	138
5.6	Klassen über freigegebene Mitglieder ansprechen	139
5.7	Klassen ableiten	141
5.8	Mitglieder überschreiben	143
5.9	Abstrakte Basisklassen – ab jetzt wird es ein wenig anspruchsvoller	145
5.10	Schnittstellen	147
5.10.1	Schnittstellen bei der Variablendeklaration	148
5.10.2	Schnittstellen führen zu (etwas) mehr Flexibilität	149

5.10.3	Ein praktisches Beispiel für die Implementierung einer Schnittstelle	149
5.10.4	Polymorphie – damit auch das einmal geklärt wird	151
5.11	Events	153
5.12	Alles besitzt einen Typ	155
5.12.1	Den Typ eines Objekts feststellen	155
5.12.2	Alles über einen Typen – das Type-Objekt gibt Auskunft	156
5.13	Zusammenfassung	157
5.14	Fragen zum Abschluss des Kapitels	157
<b>6</b>	<b>Streifzug durch die .NET-Klassenbibliothek</b>	<b>159</b>
6.1	Ein erster Überblick	160
6.1.1	Die Environment-Klasse zum Kennenlernen	160
6.1.2	Instanzenmitglieder bei Basisklassen benutzen	161
6.1.3	Ohne Hilfe geht es (fast) nicht	162
6.2	Die Rolle der Namespaces	163
6.2.1	Der Imports-Befehl	164
6.2.2	Das Global-Schlüsselwort von Visual Basic	165
6.3	Die Klassenbibliothek enthält nicht nur Klassen	165
6.4	Die wichtigsten Namespaces im Schnellüberblick	166
6.4.1	Die Namespace-Hierarchie mit dem Objektkatalog erforschen	167
6.4.2	Die Suche nach einem Namen	168
6.4.3	Mathemattikklassen	168
6.5	Der My-Namespace	169
6.5.1	Mehr über den My-Namespace	170
6.6	Dateizugriffe mit Streams	171
6.6.1	Im Mittelpunkt steht die File-Klasse	171
6.6.2	Im Mittelpunkt steht der Stream	176
6.6.3	Streams im Arbeitsspeicher anlegen	177
6.6.4	Der große Visual Basic-Ratgeber für alle Lebenslagen	178
6.6.5	Eine Frage der Zeichensätze	182
6.6.6	Mit Dateien, Verzeichnissen und Laufwerken umgehen	183
6.7	Zusammenfassung	189
6.8	Fragen	190
<b>7</b>	<b>Collections und LINQ</b>	<b>191</b>
7.1	Eine Collection ist ...	191
7.1.1	Die universelle ArrayList-Collection	192
7.1.2	Elemente hinzufügen	192

7.1.3	Elemente wieder entfernen	193
7.1.4	Eine Collection durchlaufen	193
7.1.5	Eine Collection durchsuchen	194
7.1.6	Eine Collection in ein Array umwandeln	194
7.1.7	Die HashTable – Zugriff nur über einen Schlüssel	195
7.2	Generische Collections – nur ein Typ erlaubt	196
7.2.1	Die Dictionary-Klasse – mit Typ und Schlüssel	197
7.3	Funktionsausdrücke – praktischer, als man denkt	198
7.4	LINQ für den Einstieg	202
7.5	LINQ für etwas Fortgeschrittenere	203
7.6	Zusammenfassung	204
7.7	Fragen zum Abschluss des Kapitels	205
<b>8</b>	<b>Formulare und Steuerelemente für etwas Fortgeschrittene</b>	<b>207</b>
8.1	Die Rolle der Events	208
8.2	Tastatureingaben	209
8.2.1	Was ist genau passiert? Die Aufgabe des e-Arguments	209
8.2.2	Der Unterschied zwischen KeyDown, KeyUp und KeyPress	210
8.2.3	Die Rolle des Eingabefokus	211
8.2.4	Den Eingabefokus per Programmcode setzen	213
8.2.5	Wie das Formular Tastaturereignisse erhält	213
8.2.6	Automatische Zeilenschaltung – kurzer Ausflug zu SendKeys	213
8.3	Mausaktionen	214
8.3.1	Den Mauszeiger »verschönern«	215
8.4	Ein einfacher Kreditrechner	216
8.5	Listen mit Details und Spalten – das ListView-Control	220
8.6	Es lebe die Hierarchie – das TreeView-Control	222
8.6.1	Relative Pfade beim Umgang mit Bildern	224
8.6.2	Woher bekommen ListView und TreeView ihre Bilder?	225
8.7	Das TabControl-Control für ein in Registern unterteiltes Dialogfeld	225
8.8	Ein Aufgabenplaner – Schritt für Schritt	229
8.9	Die ganze Welt im eigenen Programm – das Webbrowser-Control	238
8.10	Fast wie eine kleine Textverarbeitung – die RichTextBox	241

8.11	Zusammenfassung	243
8.12	Fragen zum Abschluss des Kapitels	243
<b>9</b>	<b>Benutzeroberflächen gestalten</b>	<b>245</b>
9.1	Auf die richtigen »Zutaten« kommt es an	246
9.2	Die Menüleiste – das MenuStrip-Steuererelement	247
9.3	Die Symbolleiste – das ToolStrip-Steuererelement	248
9.3.1	Die Statusleiste	250
9.3.2	Kontextmenüs mit dem ContextMenuStrip- Steuererelement	251
9.3.3	Die passenden Icons finden	251
9.4	Dialogfelder	252
9.4.1	Eingebaute Dialogfelder	252
9.4.2	Den Rückgabewert eines Dialogfelds festlegen	253
9.5	Der Bildbetrachter stellt sich vor	254
9.5.1	Die Umsetzung Schritt für Schritt	255
9.5.2	Erweiterung 1 – eine Dateiliste	266
9.5.3	Erweiterung 2 – ein Begrüßungsbildschirm	267
9.5.4	Die Anwendung weitergeben	267
9.5.5	Weitere Erweiterungen	268
9.6	Zusammenfassung	269
9.7	Fragen zum Abschluss des Kapitels	269
<b>10</b>	<b>Fehler abfangen und aufspüren</b>	<b>271</b>
10.1	Was genau sind Programmfehler?	272
10.2	Fehler = Ausnahme	273
10.2.1	Ausnahmen basieren auf regulären Klassen	275
10.3	Das Ausnahmefenster	275
10.4	Ausnahmen abfangen	276
10.4.1	Variablen in einem Try-Block	277
10.4.2	Mehrere Ausnahmen abfangen	278
10.4.3	Damit in jedem Fall etwas passiert – der Finally-Befehl	278
10.4.4	Ausnahmen im Debugger abfangen	279
10.5	Der integrierte Debugger stellt sich vor	280
10.5.1	Ein erster Überblick über den Debugger	280
10.5.2	Den Debugger über die Tastatur steuern	280
10.5.3	Das Direktfenster	281
10.5.4	Ein Programm im Einzelschritt-Modus ausführen	281
10.5.5	Festlegen der nächsten Anweisung	281



10.5.6	Haltepunkte setzen	282
10.5.7	Bedingte Haltepunkte einrichten	283
10.5.8	Ausdrücke überwachen	283
10.6	Tipps, um Fehler zu vermeiden	284
10.7	Zusammenfassung	284
10.8	Fragen	285
<b>11</b>	<b>Multithreading mit Visual Basic</b>	<b>287</b>
11.1	Ein »Hallo, Welt« mit mehreren Threads	288
11.1.1	Ein erstes Beispiel (das nicht auf Anhieb funktioniert)	289
11.1.2	Vorhang auf für die Delegaten	291
11.1.3	Das Beispiel wird funktionstüchtig gemacht	292
11.1.4	DoEvents macht Windows Forms-Anwendungen reaktionsfreudiger	294
11.2	Die BackgroundWorker-Komponente als komfortable Alternative	294
11.2.1	Fibonacci mit Background-Threads	297
11.3	Methodenaufrufe zur Abwechslung einmal asynchron	298
11.3.1	Asynchrones Lesen und Schreiben von Dateien	301
11.4	Timer als Alternative	302
11.5	Zusammenfassung	304
11.6	Fragen zum Abschluss des Kapitels	304
<b>12</b>	<b>Visual Basic und die Datenbanken</b>	<b>305</b>
12.1	Allgemeines über Datenbanken	306
12.1.1	Was steckt hinter einer Datenbank?	307
12.1.2	Was ist eigentlich ein SQL Server?	308
12.1.3	Welche Datenbank ist die beste?	308
12.1.4	Ein Wort zu den Access-Datenbanken	310
12.2	Visual Basic als Datenbankwerkzeug	310
12.3	Datenbanken ansprechen mit ADO.NET und SQL	312
12.3.1	ADO.NET in 5 Minuten	312
12.3.2	Die allgemeine Vorgehensweise beim Abrufen von Daten	312
12.3.3	Die Umsetzung Schritt für Schritt	314
12.3.4	Und nun mit einer SQL Server-Datenbank	320
12.3.5	Relative Pfade für die Mdf-Datei	321
12.3.6	Datenbankinhalte aktualisierbar machen	326
12.3.7	Wenn sich eine Abfrage aus mehreren Tabellen zusammensetzt	327

12.4	SQL-Abfragen	328
12.4.1	SQL-Kommandos ausprobieren	329
12.4.2	Das SQL Server Management Studio als Alternative	330
12.4.3	Anhängen einer Mdf-Datei	331
12.4.4	QueryExpress als praktische Alternative	332
12.4.5	SQL-Beispiele	333
12.5	Datenquellen für die Datenbindung	336
12.5.1	Eine Abfrage zur Datenquelle hinzufügen	342
12.5.2	Ein Blick hinter die Kulissen	344
12.6	LINQ to SQL	345
12.6.1	LINQ to SQL bei der Arbeit zusehen	350
12.7	Zusammenfassung	351
12.8	Fragen zum Kapitelabschluss	351
<b>13</b>	<b>WPF mit Visual Basic 2010</b>	<b>353</b>
13.1	WPF in 10 Minuten	354
13.1.1	Fenster statt Formulare	355
13.1.2	WPF-Controls statt Windows Forms-Controls	355
13.1.3	WPF-Designer statt Windows Forms-Designer	356
13.1.4	Die Rolle von XAML	357
13.2	Ein erstes WPF-Beispiel	362
13.3	WPF-Spielereien	368
13.3.1	Bunte Buttons	368
13.3.2	Schiefe Buttons	369
13.3.3	Sich automatisch ändernde Properties dank Trigger	371
13.3.4	Einfache Animationen	371
13.3.5	Bewegung dank Animation	373
13.3.6	Das Prinzip der Datenbindung	376
13.3.7	Runde Buttons	378
13.4	Ein Devisenrechner mit WPF	380
13.4.1	Devisenkurse über das Internet	381
13.5	Zusammenfassung	386
13.6	Fragen zum Abschluss des Kapitels	386
<b>14</b>	<b>Von der Idee zum fertigen Windows-Programm</b>	<b>387</b>
14.1	Die Idee	387
14.1.1	Die Idee wird konkreter	389
14.2	Die Umsetzung Schritt für Schritt	389
14.2.1	Die Versionsnummer und andere Programminformationen werden gesetzt	408

## *Inhaltsverzeichnis*

14.3	Die Weitergabe des Programms	409
14.3.1	Vom Debug- auf den Release-Modus umschalten	414
14.3.2	Die Rolle der .NET-Laufzeit	415
14.3.3	Wenn nur die .NET 2.0-Laufzeit zur Verfügung steht	415
14.3.4	Das Programm zum Download anbieten	416
<b>Anhang A: Das kleine .NET-Lexikon</b>		<b>419</b>
<b>Anhang B: Visual Basic 2010-Befehlswörter</b>		<b>423</b>
<b>Anhang C: Lösungen zu den Übungsaufgaben</b>		<b>435</b>
<b>Stichwortverzeichnis</b>		<b>457</b>