

Einleitung

In diesem Kapitel lernen Sie

- welche Themenschwerpunkte in diesem Buch behandelt werden
- was sich hinter .NET Framework verbirgt
- wie dieses Buch aufgebaut ist
- wie Sie am besten mit diesem Buch arbeiten

1.1 Die Themenschwerpunkte dieses Buches

Herzlich willkommen bei »Richtig einsteigen: Programmieren lernen mit Visual Basic 2010«. Dieses Buch erläutert die Programmierung in Visual Basic und zeigt, wie Sie mit Visual Basic 2010 Express Anwendungen erstellen, die unter .NET Framework ausgeführt werden können. Mit dieser Zielformulierung sind die drei Themenschwerpunkte beschrieben, die in diesem Buch besprochen und in den nächsten Abschnitten kurz vorgestellt werden.

Visual Basic 2010 Express

Visual Basic 2010 Express ist eine integrierte Entwicklungsumgebung, in der Sie den Code für Ihre Programme erstellen und schreiben und von der aus Sie Ihre Anwendungen ausführen und testen können. Die Express Editionen von Visual Studio sind die »kleineren Schwestern« des großen Visual Studio 2010. Im Unterschied zu den verschiedenen Versionen von Visual Studio 2010 sind die Express Editionen jedoch kostenlos. Trotzdem finden Sie in den Express Editionen alles, was Sie brauchen, um eigene Anwendungen zu programmieren. Ein paar der Highlights finden Sie in der folgenden Aufstellung:

- Ausgefeilter Code-Editor mit Code-Farbgebung, der beispielsweise Schlüsselwörter farbig hervorhebt und Syntax- und andere Fehler mit einer Wellenlinie markiert
- IntelliSense-Auswahllisten, die sich öffnen, während Sie Code eintippen, und sinnvolle Vorschläge zur automatischen Vervollständigung des Programmcodes liefern
- Integrierter grafischer Debugger, der Sie beim Erkennen und Beheben von Fehlern in Ihren Programmen unterstützt
- Formular-Designer, mit dem Sie Windows Forms-Anwendungen mit einer grafischen Benutzeroberfläche erstellen, indem Sie Steuerelemente via Drag & Drop auf dem Fenster der Anwendung ablegen
- Zahlreiche Steuerelemente, wie Sie sie von anderen Windows-Anwendungen her kennen und mit denen das Erstellen von Programmen mit einer grafischen Benutzeroberfläche sowohl schnell als auch einfach ist

- Unterschiedliche Projektvorlagen, die das Erstellen von Anwendungen einfacher machen
- Umfangreiche, kontextsensitive Hilfe, die gerade bei den ersten Schritten und den ersten Programmierprojekten sehr hilfreich ist

All diese Bestandteile von Visual Basic 2010 Express u.v.m. werden Sie im Laufe dieses Buches kennenlernen. Kapitel 2 zeigt, wie Sie Visual Basic 2010 Express installieren, und in Kapitel 3 erhalten Sie einen ausführlicheren Überblick über die integrierte Entwicklungsumgebung und erfahren, wie Sie schnell loslegen können. Viele der weiteren Features von Visual Basic 2010 Express lernen Sie dann im Laufe des Buches kennen; die Informationen sind dabei immer so aufbereitet, dass sie in einem konkreten Bezug zu einem konkreten Projekt oder einer bestimmten Aufgabe stehen, um so die Praxisnähe der vermittelten Kenntnisse zu gewährleisten.

Visual Basic 2010

Visual Basic 2010 ist eine der Programmiersprachen, in denen Sie Anwendungen für .NET Framework erstellen können. Bevor wir uns mit Visual Basic 2010 im Besonderen beschäftigen, wollen wir kurz noch erklären, was Programmierung ist und warum Programmierung nötig ist. Ein Computer an sich ist eigentlich nicht sehr intelligent. Im Prinzip ist ein Computer nichts weiter als eine Vielzahl winziger elektronischer Schalter, die entweder ein- oder ausgeschaltet sind. Durch unterschiedliche Kombinationen dieser Schalter können Sie erreichen, dass ein Computer eine bestimmte Aktion ausführt, z.B. etwas auf dem Bildschirm anzeigt oder einen Ton wiedergibt. Das ist es, worum es bei Programmierung eigentlich geht – dem Computer mitzuteilen, was er zu tun hat.

Wir Menschen drücken uns in sehr komplexen Sprachen aus, die aus einer Vielzahl von Wörtern und Regeln bestehen. Computer verwenden eine sehr einfache Sprache, die nur aus Nullen und Einsen besteht. 0 bedeutet aus und 1 bedeutet an. Der Versuch, mit einem Computer in seiner eigenen Sprache zu sprechen, ähnelt dem Versuch, mit einem Freund per Morsezeichen zu kommunizieren. Es ist möglich, aber warum so kompliziert?

Eine Programmiersprache fungiert als Übersetzer zwischen Ihnen und dem Computer. Anstatt die computereigene Sprache (die Maschinensprache genannt wird) zu erlernen, können Sie eine Programmiersprache verwenden, um dem Computer auf leichter erlernbare sowie besser verständliche Weise seine Anweisungen zu geben.

Ein spezielles Programm (der sogenannte Compiler) übersetzt die Anweisungen aus der Programmiersprache in die Maschinensprache. Das bedeutet, dass Sie als Visual Basic-Programmierer nicht verstehen müssen, was der Computer macht und wie er es macht. Es reicht, dass Sie wissen, wie die Programmiersprache Visual Basic funktioniert.

Der Schwerpunkt beim Erlernen einer Programmiersprache besteht also darin, deren Syntax, ihr Regelwerk zu erlernen. Dies lässt sich am einfachsten mit dem Erlernen einer neuen Fremdsprache vergleichen. Dabei müssen Sie ebenfalls lernen, wie ein Satz gebildet wird, welche grammatikalischen Regeln der Sprache zugrunde liegen, wie die Zeichensetzung funktioniert und welches Vokabular die Sprache besitzt. Das Vokabular einer Programmiersprache sind die sogenannten Schlüsselwörter. Dies sind reservierte Begriffe (im Falle von Visual Basic ans Englische angelehnt), mit denen Sie formulieren können, was das Programm machen soll. Fast alle Schlüsselwörter von Visual Basic werden nach und nach in diesem Buch erklärt. Mit den Grundlagen der Sprache Visual Basic 2010 befasst sich Teil II des Buches; die objektorientierten Konzepte von Visual Basic 2010 lernen Sie in Teil III kennen.

Bei den Programmiersprachen, die zur Programmierung für .NET Framework zur Verfügung stehen, nimmt Visual Basic 2010, besonders im Vergleich zu C# (gesprochen: »C sharp«), eine Sonderrolle ein. C# wurde speziell für .NET Framework entwickelt, Basic hingegen weist eine lange Tradition auf und die Sprachversion Visual Basic 2010 versucht, viele der vertrauten und Basic-Programmierern bekannten Sprachkonstrukte und Konzeptionen mit den Features der Objektorientierung zu verbinden. Daher werden Ihnen viele Sprachelemente vertraut vorkommen, wenn Sie beispielsweise Programmiererfahrungen in Visual Basic 6.0 oder in einer der für die Office-Programme verfügbaren Implementierungen von VBA (Visual Basic for Applications) haben. Andere Sprachelemente wiederum sind neu und spiegeln die Tatsache wider, dass mit Visual Basic 2010 Anwendungen erstellt werden, die unter .NET Framework ausgeführt werden (mehr dazu etwas später in diesem Kapitel). Doch gerade mit Visual Basic 2010 ist das Erstellen von Anwendungen einfacher als zuvor.

.NET Framework

.NET Framework (ausgesprochen: dot-net) ist eine Entwicklungsplattform, die von Microsoft im Jahre 2000 eingeführt wurde und auf der Anwendungen erstellt, verteilt und ausgeführt werden können, die .NET-Technologien nutzen. Das Problem beim Begriff .NET Framework besteht vor allem darin, dass dieser Begriff eine Mischung aus Marketing, Technologien, Visionen und Realität darstellt und je nach Kontext, in dem der Begriff verwendet wird, andere Aspekte durchschimmern. So liest sich die Definition von .NET auf der Microsoft-Website beispielsweise so: »Microsoft .NET umfasst eine Reihe von Microsoft-Softwaretechnologien, über die Daten, Systeme und Geräte miteinander verbunden werden und Mitarbeiter sich besser austauschen können.« Wirklich schlauer wird man durch diese Definition nicht.

Wenn wir die visionären und eher marketingorientierten Aspekte einmal außen vor lassen und eher auf die technische Seite fokussieren, lässt sich .NET Framework vereinfacht dargestellt in verschiedene Bereiche unterteilen, die Sie in Abbildung 1.1 auf der folgenden Seite sehen.

Auf der untersten Ebene der Abbildung sehen Sie die Betriebssystemebene; auf dieser setzt .NET Framework auf. Eines der Ziele bei .NET Framework war, eine Entwicklungsplattform zu schaffen, die so übergreifend definiert ist, dass die Möglichkeit besteht, einmal erstellte .NET-Anwendungen auf unterschiedlichen Betriebssystemplattformen ausführen zu können. So gibt es neben der Implementierung für Windows auch das Projekt Mono, mit dem .NET Framework für andere Plattformen, u.a. für Linux und Mac OS, zur Verfügung steht.

.NET Framework selbst lässt sich grob in drei Komponenten aufteilen (die jede wiederum weitere Subsysteme enthalten, die der Übersichtlichkeit halber nicht alle in der Abbildung aufgeführt sind):

- **Common Language Infrastructure** Auf der untersten Ebene befindet sich die sogenannte *Common Language Infrastructure*, abgekürzt CLI, bei der es sich um ein Regelwerk handelt, das genau beschreibt, wie bestimmte Dinge im .NET Framework implementiert werden müssen. Die CLI ist also eine Spezifikation und keine Implementierung, obwohl die CLI oft mit der Common Language Runtime (siehe nächsten Punkt) verwechselt wird. Lassen Sie uns zwei Subsysteme der CLI hervorheben:
 - **Common Type System** Ein Problem älterer Programmierplattformen unter Windows besteht darin, dass es extrem aufwendig bis unmöglich ist, Komponenten, die in verschiedenen Programmiersprachen erstellt wurden, zusammenarbeiten zu lassen. Eine der Ursachen hierfür ist, dass es schwierig war, Daten von der Komponente in Sprache A mit der Komponente in Sprache B auszutauschen, da die Sprachen die Daten unterschiedlich interpretierten. Abhilfe schafft hier

.NET Framework mit dem allgemeinen Typsystem (Common Type System, abgekürzt CTS), das vorschreibt, wie Datentypen zu implementieren sind. Dadurch ist es problemlos möglich, in einer Anwendung Komponenten zu verwenden, die mit verschiedenen Programmiersprachen erstellt wurden. Alle .NET-Sprachen, und das gilt sowohl für die von Microsoft selbst zur Verfügung gestellten Sprachen wie C#, Visual Basic 2010, C++ usw. als auch für Programmiersprachen von Drittanbietern, müssen die gleichen Datentypen gleich implementieren. Wie die Datentypen von .NET Framework mit den Visual Basic-Datentypen zusammenhängen, wird uns noch ausführlich in Kapitel 4 beschäftigen.

- **Common Language Specification** Ein weiterer Baustein der CLI ist die sogenannte *Common Language Specification*, die sich vor allem an Autoren von Compilern für Programmiersprachen wendet und in der festgelegt ist, wie der Quellcode von der Programmiersprache in die Sprache übersetzt wird, die dann von der Common Language Runtime (siehe weiter unten) ausgeführt werden kann. Ein eigener Abschnitt im dritten Kapitel des Buches ermöglicht Ihnen, einen Blick auf den Code zu werfen, der vom Visual Basic Compiler aus Ihrem Quellcode erstellt wird.

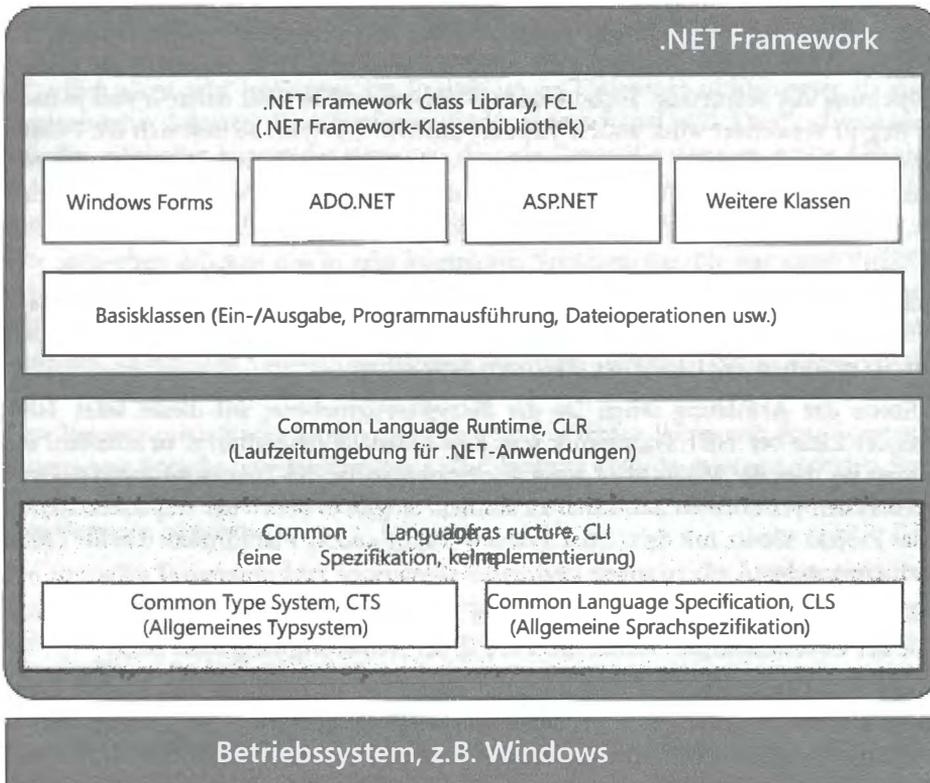


Abbildung 1.1: .NET Framework im Überblick

- **Common Language Runtime** Die *Common Language Runtime*, abgekürzt CLR, stellt für Anwendungen eine Laufzeitumgebung zur Verfügung, in der diese kontrolliert ausgeführt werden. Die Laufzeitumgebung kümmert sich darum, dass die Applikation gestartet wird, und stellt sicher, dass Programme nur solchen Code ausführen, den sie auch ausführen dürfen. So lässt sich beispielsweise konfigurieren, dass Programme, die aus dem Internet heruntergeladen wurden, keine Daten lesen

oder verändern dürfen, die sich auf dem Zielcomputer befinden. Dies bringt eine erhebliche Verbesserung beim Thema Sicherheit mit sich. Auch kümmert sich die Laufzeitumgebung um Programmfehler, die in einer Anwendung auftreten können. Das Stichwort lautet hier Ausnahmebehandlung, ein Thema, dem wir Kapitel 5 dieses Buches gewidmet haben.

- **Framework Class Library** Die *Framework Class Library*, abgekürzt FCL und im Deutschen Framework-Klassenbibliothek genannt, ist eine wahre Fundgrube von unterschiedlichsten, nützlichen und getesteten Funktionen, die Sie in Ihren Anwendungen verwenden können, um so schneller ans Ziel zu kommen. Die Klassenbibliothek ist sehr übersichtlich organisiert. Ein Baustein sind die sogenannten Basisklassen, mit denen es z.B. möglich ist, Dateien zu öffnen oder zu erstellen, Daten in diese Dateien zu schreiben oder sie aus ihnen zu lesen, Texte auf den Bildschirm auszugeben oder Tastatureingaben entgegenzunehmen, und das alles mit nur wenigen eigenen Codezeilen.

Für andere Bereiche der Programmierung existieren zahlreiche Funktionssammlungen mit spezialisierten Funktionen. Hier sei exemplarisch der Baustein *Windows Forms* erwähnt, mit dem es möglich ist, Anwendungen mit einer grafischen Benutzeroberfläche einfacher als mit Fenstern und Steuerelementen zu erstellen. Ein anderer ist die Komponente ADO.NET, die wiederum Funktionen enthält, um auf Datenbanken zuzugreifen. Diese beiden Komponenten der Klassenbibliothek werden Sie anhand von interessanten Projekten in Teil IV des Buches kennenlernen. Eine dritte Komponente ist ASP.NET, bei der es sich um das Pendant zu *Windows Forms* handelt, nur dass mit dieser Funktionssammlung keine Desktop-, sondern Webanwendungen erstellt werden können.

Visual Basic 2010 enthält zusätzlich zu den Klassen, die .NET Framework zur Verfügung stellt, eine eigene Visual Basic-Klassenbibliothek. In dieser Bibliothek finden Sie viele der Funktionen wieder, die auch in den Vorgängerversionen zur Verfügung standen.

Sollte Ihnen nun etwas der Kopf rauchen, dann machen Sie sich keine Sorgen: Dieses Phänomen kennt fast jeder, der sich zum ersten Mal mit .NET Framework auseinandersetzt. Bei den Ausführungen auf den vorigen Seiten ging es darum, einen groben Rahmen dessen abzustecken, was Ihnen mit viel mehr Praxisnähe auf den verbleibenden Seiten dieses Buches begegnen wird. Schauen Sie ab und zu in diese Einleitung, während Sie das Buch durcharbeiten, und Sie werden sehen, dass viele der Puzzlestücke aus den folgenden Kapiteln auf einmal ein übersichtlicheres Gesamtbild ergeben.

1.2 Wie Sie am besten mit diesem Buch arbeiten

Dieses Buch hat sich folgende Ziele gesetzt: Sie mit der Programmiersprache Visual Basic 2010 vertraut zu machen, zu zeigen, dass Programmieren gar nicht so schwierig ist und Spaß macht, und es möchte erreichen, dass Sie beim Programmieren verstehen, was Sie machen und warum Sie es machen.

Programmieren verstehen lernen

Wenn man das Erlernen einer Programmiersprache noch mal mit dem Erlernen einer Fremdsprache vergleicht, dann lassen sich grob zwei Lernansätze unterscheiden. Bei dem einen Ansatz würden Sie von einer neuen Fremdsprache fertige Sätze erlernen, ohne dabei jedoch mit den Grundlagen der Sprache, ihrer Grammatik, vertraut zu werden. Sie wären dann in der Lage, die einmal erlernten Sätze anzuwenden; schwierig wird dann aber das Führen eines Gesprächs, in dem Sie eigene Sätze bauen wollen. Übertragen auf das Erlernen einer Programmiersprache finden Sie häufig Bücher, die die gleiche Konzeption

verfolgen, die dann eben auch die gleichen Nachteile mit sich bringen: In deren ersten Kapiteln lernen Sie schnell, eine Anwendung mit einer grafischen Oberfläche zu erstellen, die meisten Schrittfolgen bestehen aus »Klicken Sie hier, klicken Sie dort«-Anweisungen, aber Sie erhalten wenig Informationen darüber, warum Sie dies machen sollen und in welchem größeren Kontext die Dinge stehen.

Da wir gerne möchten, dass Sie nach dem Durcharbeiten des Buches in der Lage sind – um den Vergleich zum Erlernen einer Fremdsprache noch einmal aufzugreifen –, eigene Sätze zu bilden, also eigene Programme zu erstellen, die sich von den Beispielen in diesem Buch unterscheiden, haben wir, vor allem was die Auswahl der Beispielprojekte angeht, einen etwas anderen Ansatz gewählt. Die meisten Beispielprojekte im zweiten und dritten Teil des Buches sind einfache Konsolenanwendungen. Konsolenanwendungen sind Programme, die keine grafische Benutzeroberfläche besitzen, in der Eingabeaufforderung von Windows ablaufen (also dem Programm, das Sie im Startmenü von Windows über *Alle Programme/Zubehör/Eingabeaufforderung* ausführen können). Und wir konzentrieren uns vor allem darauf, fundiertes Wissen über die Programmiersprache und deren Syntax zu vermitteln. Der Vorteil beim Erstellen von Konsolenanwendungen besteht darin, dass sie schneller zu programmieren sind und Sie sich nicht mit all den Themen, die im Zusammenhang mit der Programmierung einer grafischen Oberfläche auftauchen, auseinandersetzen müssen.

Die Windows-Anwendungen mit grafischer Oberfläche kommen aber in diesem Buch auch nicht zu kurz. Auch hier werden Sie an zahlreichen, unterschiedlichen Projekten lernen, wie eine Windows Forms-Anwendung erstellt wird, und auch dabei einen Blick hinter die Kulissen werfen können.

Alle Beispielprojekte des Buches können Sie von der Website zum Buch herunterladen und sofort starten. Wir empfehlen Ihnen, sich die Mühe zu machen, die Programme selbst einzutippen, da Sie so viel schneller die Syntax von Visual Basic 2010 erlernen, schneller mit der Funktionsweise der integrierten Entwicklungsumgebung vertraut werden und folgerichtig dadurch auch schneller in der Lage sein werden, eigene Projekte zu erstellen.

Diesem Ansatz folgend ist dieses Buch in vier Teile untergliedert, die Sie am besten von vorne nach hinten lesen und durcharbeiten, da die Informationen der einzelnen Kapitel aufeinander aufbauen.

Aufbau des Buches

Teil I, Erste Schritte, enthält diese Einleitung und zeigt, wie Sie Ihren Computer für die weitere Verwendung des Buches vorbereiten und Visual Basic 2010 Express auf Ihrem Computer installieren.

Teil II, Grundlagen von Visual Basic 2010, stellt Ihnen an vielen praktischen Beispielen die grundlegenden Sprachelemente der Programmiersprache Visual Basic 2010 vor. Wie oben erwähnt, sind die zahlreichen Beispielprojekte der Kapitel sogenannte Konsolenanwendungen. Wir haben diesen Ansatz gewählt, damit Sie sich in diesem Teil voll auf das Erlernen der Programmiersprache konzentrieren können und sich nicht noch gleichzeitig mit den Aufgaben auseinandersetzen müssen, die bei der Erstellung einer grafischen Benutzeroberfläche anfallen.

In Teil III, Konzepte der objektorientierten Programmierung, lernen Sie u.a., wie Sie ein Programm modularisieren und in kleine funktionale Einheiten aufteilen können, wie Sie die Bibliotheksfunktionen für das Arbeiten mit Zeichenketten verwenden und wie Sie eigene Klassen erstellen, um so Ihren selbst geschriebenen Code leicht wiederverwendbar zu machen. Auch alle Projekte in diesem Teil des Buches sind aus den gleichen Gründen wie bei Teil II Konsolenanwendungen.

Teil IV schließlich, Windows Forms-Anwendungen, beschäftigt sich mit den verschiedenen Aspekten der Programmierung von Anwendungen, die eine grafische Benutzeroberfläche besitzen. Sie werden

dort lernen, wie Sie die fertigen Steuerelemente der Klassenbibliothek verwenden, wie Sie eigene Steuerelemente erstellen und wie sich mit ADO.NET der Zugriff auf Datenbanken realisieren lässt.

Aufbau der einzelnen Kapitel

Die einzelnen Kapitel des Buches sind folgendermaßen aufgebaut:

- **Lernziele** Sie finden am Anfang jedes Kapitels eine Übersicht zu den Lernzielen und Themen des Kapitels.
- **Beispielprojekte** Jedes Kapitel enthält ein oder mehrere Beispielprojekte, an denen die Lerninhalte des Kapitels praxisnah erläutert werden.
- **Schrittfolgen** Die Anleitungen für das Erstellen der Projekte sind als Schrittfolgen dargestellt. Auch wenn Sie alle Beispielprojekte von der Website zum Buch herunterladen können, empfehlen wir Ihnen, alle Beispiele selbst zu erstellen und den Programmcode einzutippen. Sie werden auf diese Weise schneller mit der Syntax von Visual Basic vertraut und lernen so auch die integrierte Entwicklungsumgebung mit all ihren nützlichen Features rascher kennen.
- **Übungen** Am Ende jedes Kapitels finden Sie einen Abschnitt mit Übungen zu dem jeweiligen Kapitel. Manchmal handelt es sich dabei um einfache Fragen, manchmal enthält eine Übung eine Aufgabe, um das im Kapitel vorgestellte Projekt weiter zu verbessern. Alle Antworten finden Sie auf der Website zu diesem Buch. Versuchen Sie, die Übungen selbstständig durchzuführen, und schauen Sie sich erst dann die Lösungen an. So werden Sie schneller in der Lage sein, eigene Programmierprojekte zu realisieren.

Konventionen in diesem Buch

Dieses Buch verwendet einige wenige Konventionen. Wenn Sie mit ihnen vertraut sind, können Sie einfacher mit dem Buch arbeiten:

- **Fette Schrift** Text, den Sie in den Schrittfolgen eintippen müssen, wird in fett gedruckten Buchstaben dargestellt. Auch in Codelistings werden Änderungen, die in einem bestimmten Schritt im Quellcode vorgenommen wurden, fett, aber dann in Listingschrift, dargestellt.
- **Kursive Schrift** Die kursive Schrift wird für Dateinamen verwendet, bezeichnet Elemente der Benutzeroberfläche (also die Menüs von Visual Basic, Dialogfelder und auch die Namen der Eigenschaften, wie sie in der IDE erscheinen) und wird in seltenen Fällen auch für Hervorhebungen verwendet.
- **Listingschrift** Die Listingschrift wird verwendet, wenn Codefragmente aus dem Beispielcode abgedruckt werden. Im Fließtext werden die Begriffe in Listingschrift abgedruckt, die auf Stellen im Quellcode verweisen.

Zusätzlich finden Sie im ganzen Buch verteilt Absätze, die mit den folgenden Icons versehen sind und die auf bestimmte, hilfreiche Elemente hinweisen:



Wichtig

Absätze mit diesem Icon enthalten wichtige Informationen, auf die Sie unbedingt achten sollten.