

Kapitel 1

Einleitung

In diesem Kapitel lernen Sie

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

1.1 Die Themenschwerpunkte dieses Buches

Herzlich willkommen bei »Richtig einsteigen: Programmieren lernen mit Visual C# 2010«. Es erläutert die Programmierung in C# und zeigt, wie Sie mit Visual C# 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 C# 2010 Express

Visual C# 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 Editions 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 Editions jedoch kostenlos. Trotzdem finden Sie in den Express Editions alles, was Sie brauchen, um eigene Anwendungen zu programmieren. Ein paar der Highlights finden Sie in der folgenden Aufstellung:

- Ausgefeilter Code-Editor mit Codefarbgebung, 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 machen
- Integrierter grafischer Debugger, der Sie beim Erkennen und Beheben von Fehlern in Ihren Programmen unterstützt
- Ansicht-Designer, mit dem Sie Windows Forms-Anwendungen mit einer grafischen Benutzeroberfläche erstellen, indem Sie Steuerelemente mit Drag & Drop auf dem Anwendungsfenster 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 zu bewerkstelligen ist
- Unterschiedliche Projektvorlagen, die das Erstellen von Anwendungen einfacher machen

Kapitel 1 Einleitung

- Umfangreiche, kontextsensitive Hilfe, die gerade bei den ersten Schritten und den ersten Programmierprojekten sehr hilfreich ist

All diese Bestandteile von Visual C# 2010 Express u.v.m. werden Sie im Laufe dieses Buches kennenlernen. Kapitel 2 zeigt, wie Sie Visual C# 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 C# 2010 Express lernen Sie dann in den weiteren Kapiteln kennen; die Informationen sind dabei immer so aufbereitet, dass sie in einem Bezug zu einem konkreten Projekt oder einer bestimmten Aufgabe stehen, um so die Praxisnähe der vermittelten Kenntnisse zu gewährleisten.

C#

C# ist eine der Programmiersprachen, in denen Sie Anwendungen für .NET Framework erstellen können. Wenn man sich den Stammbaum der verschiedenen Programmiersprachen anschaut, lässt sich feststellen, dass C# (gesprochen: »C sharp«) viel Ähnlichkeiten mit den Programmiersprachen C und C++ aufweist, aber auch Elemente, die von Java her bekannt sind. C# wurde von Microsoft entwickelt; an ihr hat auch der Entwickler von Turbo Pascal und Delphi, Anders Hejlsberg, mitgearbeitet.

C# wurde speziell für .NET Framework entwickelt und ist eine objektorientierte Sprache. C# weist viel von der Einfachheit auf, die Visual Basic auszeichnet, verbindet diese aber mit der Leistungsfähigkeit von C++, ohne dabei die Konstrukte zu verwenden, die C++ recht kompliziert macht. Wenn Sie das Gerücht gehört haben sollten, dass C# schwer zu erlernen sei, dann sei Ihnen versichert, dass dies nicht so ist. Den Beweis hierfür halten Sie mit dem vorliegenden Buch in Händen.

Der Schwerpunkt beim Erlernen einer Programmiersprache besteht 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 C# ans Englische angelehnt), mit denen Sie formulieren können, was das Programm machen soll. Eine Liste der C#-Schlüsselwörter finden Sie in Kapitel 4 (Seite 44); fast alle der Schlüsselwörter werden nach und nach in diesem Buch erklärt.

Mit den Grundlagen der Sprache C# befasst sich Teil II des Buches; die objektorientierten Konzepte von C# lernen Sie in Teil III kennen.

.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 sehen.

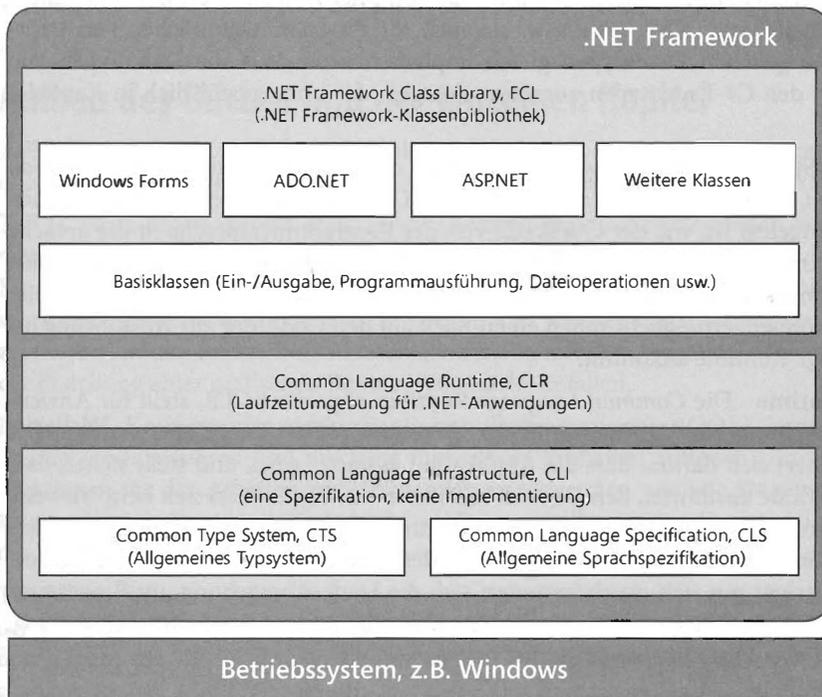


Abbildung 1.1: .NET Framework im Überblick

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 weiter unten) 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 zur Verfügung gestellten Sprachen wie C#, Visual Basic .NET, C++ usw. als auch für Programmiersprachen von Drittanbietern, müssen die gleichen Datentypen gleich implementieren. Wie die Datentypen von .NET Framework mit den C#-Datentypen zusammenhängen, wird uns 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 Bonuskapitel zum dritten Kapitel des Buches, das Sie von der Website zum Buch herunterladen können, ermöglicht Ihnen einen Blick auf den Code, der zur Ausführung bei der Common Language Runtime ankommt.
- **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 Anwendung verwenden können, um so schneller ans Ziel zu kommen. Die Klassenbibliothek ist sehr übersichtlich organisiert. Bei einem Baustein handelt es sich um die Basisklassen, mit denen es 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 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, grafische Benutzeroberflächen 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 an 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.

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. Mit den Ausführungen auf den vorigen Seiten ging es darum, einen groben Rahmen dessen abzustecken, was Ihnen mit viel mehr Praxisnähe auf den verbleibenden Seiten des 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 ist in vier Teile untergliedert, die Sie am besten von Anfang bis Ende lesen und durcharbeiten, da die Informationen der einzelnen Kapitel aufeinander aufbauen.

Aufbau des Buches und der einzelnen Kapitel

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

Teil II, Grundlagen von C#, stellt Ihnen an vielen praktischen Beispielen die grundlegenden Sprachelemente der Programmiersprache C# vor. Die zahlreichen Beispielprojekte der Kapitel sind sogenannte Konsolenanwendungen, die in der Windows-Eingabeaufforderung ausgeführt werden. 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 (OOP), 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.

Die einzelnen Kapitel des Buches sind folgendermaßen aufgebaut:

- Sie finden am Anfang jedes Kapitels eine Übersicht zu den Lernzielen und Themen des Kapitels.
- Jedes Kapitel enthält ein oder mehrere Beispielprojekte, an denen die Lerninhalte des Kapitels praxisnah erläutert werden. 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 C# vertraut und lernen so auch die integrierte Entwicklungsumgebung mit all ihren nützlichen Features rascher kennen.
- Text, den Sie in den Schrittfolgen eintippen müssen, wird in fett gedruckten Buchstaben dargestellt.
- 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.

Die DVD-ROM, Softlinks und die Website zu diesem Buch

Auf der Begleit-DVD zu diesem Buch finden Sie eine vollständige Version von Visual C# 2010 Express. Installieren Sie zuerst Visual C# 2010 Express; dies ist in Kapitel 2 ausführlich beschrieben. Surfen Sie anschließend zur Website dieses Buches unter www.vsexpress.de, von der Sie die Dateien mit den Beispielprojekten herunterladen können.

Neben den Beispieldateien finden Sie auf der Website zusätzliche Downloads, die in den jeweiligen Kapiteln beschrieben sind. Dies sind u.a. weitere Beispielprojekte und Kapitel aus anderen Microsoft Press-Büchern, mit denen Themen, die in diesem Buch aus Platzgründen nur einführend erläutert werden können, vertieft werden.

Im Text des Buches finden Sie außerdem zahlreiche Softlinks, die Sie auf interessante, unterhaltsame oder lehrreiche Websites verweisen. Sie können die Softlinks entweder direkt auf der Startseite von www.vsexpress.de eingeben oder in Ihren Browser die angegebene Adresse eintippen, um die jeweilige Website zu öffnen.

Wenn Sie nach dem Durcharbeiten an fortgeschritteneren Themen zur Programmierung mit Visual C# 2010 Express interessiert sind, werfen Sie einen Blick auf die Website www.tiefer-einsteigen.de. Dort finden Sie weitere Informationen zu einem Buch, das von Rainer G. Haselier, einem der Autoren dieses Buches verfasst wurde, und dessen Inhalte dort weitermachen, wo dieses Einführungsbuch aufhört.

Die Symbole in diesem Buch



Wichtig

Wichtige Informationen, auf die Sie unbedingt achten sollten.



Hinweis

Weiterführende Informationen, die Sie im Hinterkopf behalten sollten.



Tipp

Hier finden Sie Informationen, wie Sie Dinge besonders einfach und zeitsparend durchführen können.



Background

Wichtige Hintergrundinformationen. Die Informationen sind nicht erforderlich, um eine bestimmte Schrittfolge durchzuführen, sie vertiefen jedoch Ihr Wissen über die Zusammenhänge, in denen eine bestimmte Aktion steht.



Internet-Links

Links zu weiterführenden Kapiteln aus anderen Microsoft Press-Büchern, die Sie kostenlos herunterladen können und die Themen vertiefen, die in diesem Buch nur angerissen werden können.

Genug der Vorrede und hinein ins Vergnügen

Viel Spaß beim Programmieren wünschen Ihnen

Klaus Fahnenstich und Rainer G. Haselier