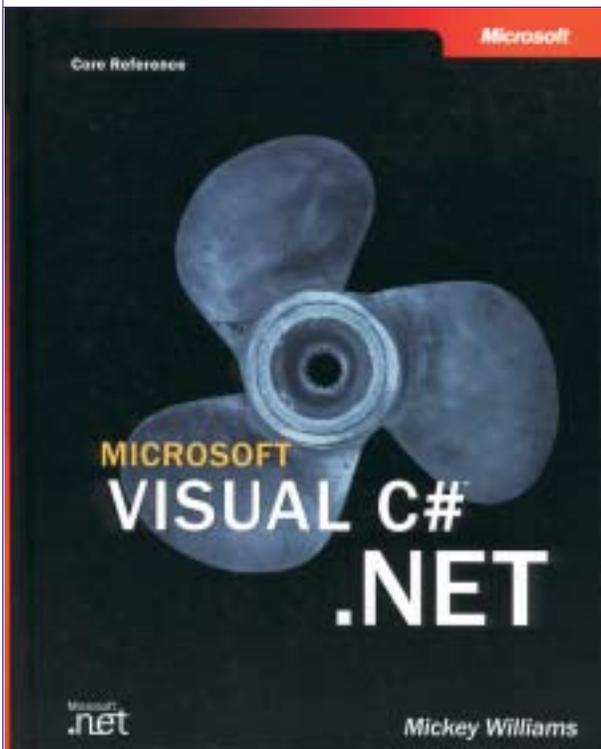


Microsoft Visual C# .NET

Martin Schönhacker



Mickey Williams; Microsoft Press; ISBN 0-7356-1290-0; 746 Seiten + CD-ROM (ca. 100 MB) + DVD-ROM (ca. 2,15 GB); ca. Euro 57,90

Der erste Absatz aus der Besprechung des Buches „*Programming Microsoft .NET*“ von Jeff Prosise (ebenfalls in diesem Heft der PCNEWS) gilt sinngemäß auch hier: Das vorliegende Buch ist ebenso massiv. Inhaltlich handelt es aber um eine nicht ganz so schwere Lektüre. Es wird zwar Erfahrung mit der objektorientierten Programmierung vorausgesetzt, aber die Bestandteile von C# (englisch ausgesprochen: „C sharp“) und einigen ausgewählten Teilen des Microsoft .NET Framework werden in einem deutlich gemütlicheren Tempo eher ausführlich vorgestellt.

Der Autor konzentriert sich allerdings fast ausschließlich auf den Einsatz von Visual Studio. Wer also diese Entwicklungsumgebung nicht besitzt oder (aus welchen Gründen auch immer, zum Beispiel wegen einer Abneigung gegen per Mausklick automatisch generierten Programmcode) nicht verwenden will, sollte sich vielleicht lieber gleich nach einem anderen Buch umsehen.

Im ersten großen Abschnitt, den **Kapiteln 1-5**, wird Visual C# .NET zunächst einmal im Überblick und mit seinen grundlegendsten Bausteinen vorgestellt. Nach einer „Tour“ durch die Entwicklungsumgebung, die sich natürlich auf das übliche minimalistische Programm „Hello, World“ stützt, geht es um Datentypen, Operatoren, Referenzen, Eigenschaften und die Steuerung des Kontrollflusses.

Mit fortgeschrittenen Elementen von C# geht es in den **Kapiteln 6-10** weiter. Delegates, Attribute, Enumeratoren, Kollektionen und vieles mehr werden mit Beispielen illustriert und sehr gründlich erklärt. Die Fehlersuche wird in verschiedenen Varianten diskutiert, und auch das für komplexere Anwendungen oft unverzichtbare Multithreading kommt nicht zu kurz.

Mehr als ein Drittel des Buches ist dem deutlich umfangreichsten dritten Teil gewidmet, der sich mit Windows Forms beschäftigt und aus den **Kapiteln 11-17** besteht. Es handelt sich dabei nicht etwa um ein völlig neues Konzept, sondern im .NET Framework werden Programme mit einer unter Windows lauffähigen grafischen Benutzeroberfläche einfach als „Windows Forms“ bezeichnet. Die einzelnen Kapitel beschäftigen sich mit den grundlegenden Kontrollelementen und der Behandlung von Benutzereingaben, aber auch mit der grafischen Schnittstelle GDI+ und komplexen Elementen wie z.B. Grids, Listen oder Baumdarstellungen.

Die **Kapitel 18-19** beschreiben den Zugriff auf Datenbanken mit ADO.NET und die Speicherung strukturierter Daten mit XML.

Erst der fünfte und letzte Teil, bestehend aus den **Kapiteln 20-21**, behandelt schließlich die Implementierung von Web-Formularen mit ASP.NET sowie SOAP und Web Services. Der vergleichsweise vernachlässigbare Umfang von ganzen 20 Seiten für das gesamte Kapitel über Web Services zeigt schon, dass die Prioritäten dieses Buches eindeutig bei Windows-Anwendungen liegen. Das ist ja auch nicht prinzipiell etwas Schlechtes, aber es sollte einem vor dem Kauf bewusst sein.

Auf der beigegepackten CD-ROM ist neben allen Beispieldateien auch das komplette Buch selbst als eBook in Form einer Hilfedatei mit ca. 5,0 MB enthalten. Dadurch ist der Inhalt nicht nur leicht zu durchsuchen, sondern auch bei der Arbeit immer direkt verfügbar. Damit man sich gleich ins Programmieren stürzen kann, gibt es dazu auch noch eine auf 60 Tage befristete Testversion von Microsoft Visual Studio .NET Professional auf einer separaten DVD-ROM.

Unter dem Titel „Microsoft Visual C# Entwicklerbuch“ gibt es dieses Buch auch in deutscher Übersetzung. Die CD-ROM enthält hier als Bonus das eBook in beiden Sprachen.

Während das Buch definitiv nicht für den Einstieg in die Programmierung gedacht ist, eignet es sich sehr gut zum Umstieg aus einer anderen Programmiersprache zu Visual C#. Vor allem mit Vorkenntnissen in C++ oder Java ist es relativ leicht, die Beispiele nachzuvollziehen und rasch ein Gefühl für C# zu bekommen.

Using the Stack Class

The Stack class implements a last-in, first-out data structure that stores its objects such that the last object inserted appears to be at the top of the stack, as shown in Figure 8-4.

Figure 8-4.
A stack, which is a last-in, first-out collection. In addition to the *ICollection*, *IEnumerable*, and *ICollection* interfaces, the most commonly used