



Hanspeter Mössenböck ist Professor für Informatik an der Universität Linz und Leiter des Instituts für Systemsoftware. Er beschäftigt sich vor allem mit Programmiersprachen, Compilern und Systemsoftware.

Als ehemaliger Mitarbeiter von Prof. Niklaus Wirth an der ETH Zürich war er Mitglied des Oberon-Teams, in dem ein Pascal-Nachfolger samt innovativem Betriebssystem entwickelt wurde. Ferner ist er Autor des Compiler-Generators Coco/R, der heute weltweit als Public-Domain-Software eingesetzt wird. Neben einem Forschungsaufenthalt bei Sun Microsystems in Kalifornien hatte er Gastprofessuren in Oxford und Budapest inne. Er ist Verfasser der Bücher »Sprechen Sie Java?« und »Objektorientierte Programmierung in Oberon-2« sowie Mitverfasser der Bücher »Die .NET-Technologie« und »Ein Compiler-Generator für Mikrocomputer«.

dpunkt.lehrbuch

Bücher und Teachware für die moderne Informatikausbildung

Berater für die dpunkt.lehrbücher sind:

Prof. Dr. Gerti Kappel, E-Mail: gerti@big.tuwien.ac.at

Prof. Dr. Ralf Steinmetz, E-Mail: Ralf.Steinmetz@kom.tu-darmstadt.de

Prof. Dr. Martina Zitterbart, E-Mail: zit@telematik.informatik.uni-karlsruhe.de

Papier
plus⁺
PDF.

Zu diesem Buch – sowie zu vielen weiteren dpunkt.büchern – können Sie auch das entsprechende E-Book im PDF-Format herunterladen. Werden Sie dazu einfach Mitglied bei [dpunkt.plus⁺](http://dpunkt.plus+):

www.dpunkt.de/plus

Hanspeter Mössenböck

Kompaktkurs C# 5.0

4., aktualisierte und erweiterte Auflage



dpunkt.verlag

Prof. Dr. Hanspeter Mössenböck
Johannes Kepler Universität Linz
Institut für Systemsoftware
Altenbergerstraße 69 · A-4040 Linz
E-Mail: hanspeter.moessenboeck@jku.at
<http://ssw.jku.at>

Lektorat: Christa Preisendanz
Copy-Editing: Ursula Zimpfer, Herrenberg
Satz: FrameMaker-Dateien vom Autor
Herstellung: Birgit Bäuerlein
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

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 978-3-86490-227-7

4., aktualisierte und erweiterte Auflage 2015
Copyright © 2015 dpunkt.verlag GmbH
Wieblinger Weg 17
69123 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Geleitwort

C# und dessen Entwicklung sind untrennbar mit der darunterliegenden Laufzeitumgebung – dem .NET-Framework – verbunden. Denn obwohl es unter .NET eine ganze Reihe von Programmiersprachen gibt, nimmt C# als die Implementierungssprache von .NET eine Sonderstellung ein. Das .NET-Framework verfolgt das Ziel, die Entwicklung diverser Anwendungen auf unterschiedlichen Plattformen wie Windows, Windows Phone, iOS oder Android zu vereinfachen und den Entwickler dabei von Routineaufgaben, wie etwa dem Memory Management, zu entlasten.

Heute findet man .NET – und damit auch C# – nicht nur auf klassischen Desktops und Servern, sondern auch auf mobilen Geräten, auf Mikrocontrollern, auf Sensoren und ähnlichen Geräten, die im Umfeld von IoT (Internet of Things) eine zentrale Rolle spielen.

Um dieser Diversität gerecht zu werden, müssen das .NET-Framework und dessen Entwicklung ständig an neue Anforderungen angepasst werden. Beispiele dafür sind moderne CPUs, die Entwickler vor die Aufgabe stellen, sich mit parallelen Anwendungen zu beschäftigen, oder SIMD-Technologien für die Beschleunigung spezieller Berechnungen. Aber auch Spezifika bestimmter Geräteklassen wie kleine Geräte mit limitierten Ressourcen versus große Cloud-Anwendungen werden zunehmend berücksichtigt. Um diese Weiterentwicklung auf eine breite Basis zu stellen, werden zahlreiche .NET-Technologien – z.B. der C#-Compiler – mittlerweile im Rahmen der .NET Foundation als Open-Source-Projekte entwickelt.

Was hat das alles aber mit C# zu tun? Nun, C# wurde so entworfen, dass die Eigenschaften von .NET in dieser Sprache optimal genutzt werden und Teile von .NET selbst, wie etwa ein Großteil der Base Class Library, in dieser Sprache möglichst einfach implementiert werden können. So ermöglichen zum Beispiel partielle Klassen die effiziente Erweiterung generierter Klassen, Language Integrated Queries (LINQ) erlauben die einfachere Verarbeitung und Parallelisierung von Daten, und asynchrone Methoden ermöglichen eine bessere Nutzung verfügbarer Ressourcen bzw. die Erstellung benutzerfreundlicherer Anwendungen.

Die Mächtigkeit von C# sowie seine enge Verflechtung mit .NET sind jedoch für manchen Einsteiger ein wenig verwirrend. Genau hier setzt das vorliegende Buch an. Der Autor gibt darin – basierend auf seiner langjährigen Erfahrung mit Programmiersprachen – einen kompakten Überblick über C# für Praktiker. Die

Querverweise zu Java sowie zahlreiche Beispiele und Übungsaufgaben mit Musterlösungen ermöglichen ein rasches Einarbeiten in die Materie.

Dem Entwickler wird aber auch ein Blick hinter die Kulissen der Sprache gewährt, damit er auch dann versteht, was abläuft, wenn ihn etwaige Wizards bei der Softwareentwicklung mit C# unterstützen.

Andreas Schabus

Technology Advisor

Microsoft Österreich GmbH

Vorwort zur 4. Auflage

C# ist heute eine der meistverwendeten Programmiersprachen – sowohl in der Industrie als auch an Schulen und Universitäten. Aber nicht nur die Anwendergemeinschaft ist gewachsen, sondern auch der Sprachumfang. Die dritte Auflage dieses Buchs beschrieb C# 3.0 und enthielt auch bereits eine Vorschau auf C# 4.0. Mittlerweile stehen wir bei C# 5.0, was eine vierte Auflage dieses Buchs notwendig machte. In C# 5.0 kamen vor allem Sprachelemente und Bibliotheksklassen für die parallele und asynchrone Programmierung hinzu.

Die Taktraten moderner Rechner sind an einer physikalischen Grenze angekommen und können nicht mehr erhöht werden. Die einzige Möglichkeit, Rechner weiterhin schneller zu machen, ist ab nun die Erhöhung der Anzahl ihrer Prozessoren. Mittlerweile sind 4 bis 8 Prozessoren pro Chip bereits Standard und bald wird es Arbeitsplatzrechner mit 16 und mehr Prozessoren geben.

Um all diese Prozessoren auszunutzen, muss man Programme parallelisieren. Leider geht das nicht automatisch, sondern muss vom Programmierer sorgfältig geplant und umgesetzt werden. Die parallele Programmierung ist nicht trivial, und die dabei eingesetzten Sprachmittel waren anfangs rudimentär und kompliziert. Mittlerweile bieten moderne Sprachen wie C# 5.0 jedoch elegante Anweisungsarten und Klassen, die die parallele Programmierung wesentlich erleichtern.

Zum einen gibt es in C# nun Klassen wie `Task` oder `Parallel`, mit denen man Codestücke parallel zueinander laufen lassen kann – idealerweise auf unterschiedlichen Prozessoren. Zum anderen wurde eine neue Art von Methoden eingeführt (sogenannte *asynchrone Methoden*), bei denen der Rufer nicht mehr warten muss, bis die aufgerufene Methode zu Ende gelaufen ist, sondern gleich nach dem Aufruf mit anderen Arbeiten weitermachen kann. Der Rufer und die aufgerufene Methode laufen also (weitgehend) parallel, was vor allem den Vorteil hat, dass der Rufer nicht durch lange Wartezeiten blockiert ist. Falls die Methode Ergebniswerte liefert, kann sich der Rufer diese zu einem späteren Zeitpunkt abholen.

Der C#-Compiler verwendet viel Aufwand dafür, diese Parallelität zu verstecken und dem Programmierer den Eindruck eines sequentiellen Programms zu geben, das im Hintergrund parallelisiert wird. Auf diese Weise wird die parallele und asynchrone Programmierung einfacher als je zuvor. Die neue Auflage des Buchs beschreibt die dazu notwendigen Sprachmittel samt Übungsaufgaben in einem eigenen Kapitel.

Unter <http://dotnet.jku.at> findet man wie bisher Musterlösungen zu den Übungsaufgaben am Ende jedes Kapitels. Die Webseite enthält außerdem Powerpoint-Folien einer Lehrveranstaltung über C# samt den neuen Sprachmerkmalen von C# 5.0.

Trotz des gewachsenen Sprachumfangs von C# ist auch die vierte Auflage dieses Buchs wieder kompakt, ohne auf Vollständigkeit oder Genauigkeit zu verzichten. Das Buch ermöglicht Studenten und Praktikern, sich rasch in C# einzuarbeiten und produktiv zu werden. Ich hoffe, dass es viele neue Freunde findet, und bin wie immer für Anregungen und Kritik offen.

Hanspeter Mössenböck
September 2014