



Hanspeter Mössenböck

# Compilerbau

Grundlagen und  
Anwendungen

dpunkt.verlag



**Hanspeter Mössenböck** ist Professor für Informatik an der Johannes Kepler Universität Linz und beschäftigt sich seit vielen Jahren mit Programmiersprachen und Compilern. Er war Mitarbeiter von Professor Niklaus Wirth an der ETH Zürich, einem der Pioniere des Compilerbaus, der unter anderem die Programmiersprache Pascal entwickelt hat. Seit über 20 Jahren kooperiert er mit Oracle Labs auf dem Gebiet der dynamischen Compileroptimierung für Java und andere Programmiersprachen. Viele der an seinem Institut entwickelten Techniken werden heute weltweit in Java-Systemen eingesetzt. Hanspeter Mössenböck ist Autor von Büchern über Java, C#, .NET sowie über compilererzeugende Systeme.

#### Copyright und Urheberrechte:

Die durch die dpunkt.verlag GmbH vertriebenen digitalen Inhalte sind urheberrechtlich geschützt. Der Nutzer verpflichtet sich, die Urheberrechte anzuerkennen und einzuhalten. Es werden keine Urheber-, Nutzungs- und sonstigen Schutzrechte an den Inhalten auf den Nutzer übertragen. Der Nutzer ist nur berechtigt, den abgerufenen Inhalt zu eigenen Zwecken zu nutzen. Er ist nicht berechtigt, den Inhalt im Internet, in Intranets, in Extranets oder sonst wie Dritten zur Verwertung zur Verfügung zu stellen. Eine öffentliche Wiedergabe oder sonstige Weiterveröffentlichung und eine gewerbliche Vervielfältigung der Inhalte wird ausdrücklich ausgeschlossen. Der Nutzer darf Urheberrechtsvermerke, Markenzeichen und andere Rechtsvorbehalte im abgerufenen Inhalt nicht entfernen.

Hanspeter Mössenböck

# Compilerbau

Grundlagen und Anwendungen



dpunkt.verlag

Hanspeter Mössenböck  
*hanspeter.moessenboeck@jku.at*

Lektorat: Christa Preisendanz  
Lektoratsbüro: Julia Griebel, Friederike Demmig  
Copy-Editing: Ursula Zimpfer, Herrenberg  
Satz: Hanspeter Mössenböck  
Herstellung: Stefanie Weidner, Frank Heidt  
Umschlaggestaltung: Eva Hepper, Silke Braun

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:  
Print 978-3-98889-008-5  
PDF 978-3-98890-145-3  
ePub 978-3-98890-146-0

1. Auflage 2024  
Copyright © 2024 dpunkt.verlag GmbH  
Wieblinger Weg 17  
69123 Heidelberg

*Schreiben Sie uns:*  
Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: *hallo@dpunkt.de*.

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.

# Geleitwort

Programmiersprachen spricht man nicht – es sind formale Systeme. Programme, die die Texte, die in Programmiersprachen formuliert sind, in Folgen von Computerinstruktionen übersetzen, nennt man Compiler. Es handelt sich dabei um komplexe Programme. Am Anfang der Compiler-Technologie standen die Sprachen Fortran (1957) und Algol (1960). Die Fertigung ihrer Compiler beschäftigte große Teams von Programmierern über Jahre. Durch die Systematisierung des Compilerbaus, wie sie in diesem Buch vorgestellt wird, konnte diese Arbeit für Einzelpersonen jedoch auf wenige Monate reduziert werden. Dies war ein gewaltiger Fortschritt.

Grundlegend neue Programmiersprachen gibt es heute eher selten und völlig neue Computerarchitekturen ebenfalls. Compilerbau scheint daher eine spezielle Sparte zu sein, die wenigen Spezialisten in großen Firmen vorbehalten bleibt. Wozu also dieses Buch?

Der Grund ist einfach. Jedes Programm, jede Anwendung, ist nach Regeln aufgebaut, die spezifizieren, wie Anweisungen und Datendeklarationen auszusehen haben. Wenn diese Regeln formalisiert werden, erhöht das die Klarheit und Verständlichkeit der Anwendung. Der Schlüssel dazu beruht auf Formalisierung, also auf der Spezifikation einer Syntax für die Eingabe. Dadurch wird eine Syntaxanalyse ermöglicht, die die Grundlage des Compilerbaus darstellt. Die Syntaxanalyse und weitere in diesem Buch beschriebene Techniken sind aber nicht nur zur Verarbeitung von Programmiersprachen nützlich, sondern lassen sich auch auf viele andere Probleme anwenden, bei denen es um die systematische Verarbeitung strukturierter Eingaben geht. Diese Techniken tragen maßgeblich zur Korrektheit und zum Verständnis solcher Anwendungen bei.

Möge dieses Buch bei dieser vielversprechenden Entwicklung behilflich sein!

*Prof. em. Dr. Niklaus Wirth*  
Zürich, im Dezember 2023<sup>1</sup>

---

1. Prof. Dr. Niklaus Wirth verstarb völlig unerwartet eine Woche nach Abfassung dieses Geleitworts im 90. Lebensjahr.