



Hanspeter Mössenböck ist Professor für Informatik an der Universität Linz. Seine Interessen liegen auf dem Gebiet der Programmiersprachen und der Systemsoftware, insbesondere des Übersetzerbaus. Von 1988 bis 1994 war er Assistenzprofessor an der ETH Zürich und Mitarbeiter von Prof. Niklaus Wirth im Oberon-Projekt. Seit 2000 kooperieren er und sein Institut mit dem Java-Team bei Sun Microsystems bzw. Oracle in Kalifornien. Dabei wurden neue Optimierungen im Java-Compiler und in der Java-VM entwickelt, die heute Teil der Java-Distribution sind. Ferner hatte er Gastprofessuren in Oxford und Budapest inne, wo ihm 2006 ein Ehrendoktorat verliehen wurde.

Mössenböck ist Verfasser der Bücher »Kompaktkurs C#« und »Objektorientierte Programmierung in Oberon-2« sowie Mitverfasser der Bücher »Die .NET-Technologie«, »Ein Compiler-Generator für Mikrocomputer« und »Informatik-Handbuch«.

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

Sprechen Sie Java?

Eine Einführung in das systematische
Programmieren

5., überarbeitete 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, Düsseldorf
Druck: Koninklijke Wöhrmann B.V., Zutphen, Niederlande

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 978-3-86490-099-0
5., überarbeitete und erweiterte Auflage 2014
Copyright © 2014 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

Vorwort zur 5. Auflage

Lange haben die Java-Entwickler darauf gewartet. Mit Java 8, das 2014 auf den Markt kam, wurden in Java endlich *Lambda-Ausdrücke* eingeführt.

Lambda-Ausdrücke sind namenlose Methoden, die man in Variablen speichern, als Parameter an andere Methoden übergeben und zu einer späteren Zeit ausführen kann. Sie erlauben es, viele Algorithmen der Java-Bibliothek mit Code zu parametrisieren. Ursprünglich stammen Lambda-Ausdrücke aus der Welt der funktionalen Sprachen. Dass sie nun auch in immer mehr imperative Sprachen Eingang finden, zeigt, dass Programmierparadigmen in mancher Hinsicht näher zusammenrücken und bewährte Konzepte zusammenwachsen.

Java 8 bringt noch einige weitere Neuerungen, die sich aber meist nur auf die Java-Klassenbibliothek beziehen oder fortgeschrittene Details betreffen, die den Rahmen dieses einführenden Lehrbuchs sprengen würden. Auf eine dieser Neuerungen wird allerdings dennoch kurz eingegangen, und zwar auf *Default-Methoden* in Interfaces. Um Lambda-Ausdrücke in der Klassenbibliothek einsetzen zu können, mussten einige Interfaces um neue Methoden erweitert werden. Da diese Interfaces jedoch weltweit in Verwendung sind, hätte das bedeutet, dass Millionen von Programmierern diese neuen Methoden implementieren hätten müssen. Java 8 erlaubt daher, Interface-Methoden nun auch mit Code zu versehen. Klassen, die solche Interfaces implementieren, erben diesen Code und müssen die betreffenden Methoden nicht selbst implementieren. Da solche Default-Methoden vor allem in Zusammenhang mit Lambda-Ausdrücken eingeführt wurden, werden sie kurz im Kapitel über Lambda-Ausdrücke beschrieben.

Die neue Auflage dieses Buches enthält auch eine kurze Beschreibung von *anonymen Klassen*, die seit Langem Teil von Java sind, aber als fortgeschrittenes Sprachmerkmal bisher in diesem Buch weggelassen wurden. Da Lambda-Ausdrücke aber in Java mittels anonymer Klassen implementiert werden, gibt es nun auch ein kurzes Kapitel über sie.

Linz, im Jänner 2014
Hanspeter Mössenböck

Vorwort zur 4. Auflage

Nach der Übernahme von Sun Microsystems durch Oracle im Januar 2010 wurde auch die Zukunft der Programmiersprache Java neu geplant. Im Herbst 2011 ist es nun endlich so weit: Die neue Version von Java kommt unter dem Namen Java 7 auf den Markt. Java 7 ist allerdings eher eine Zwischenversion, in der es auf Programmiersprachenebene nur wenige Neuigkeiten gibt. Erst die nächste Version (Java 8), die für Ende 2012 angekündigt ist, soll wieder mehr neue Konzepte bringen.

In Java 7 wurden einige Vereinfachungen eingebaut. So ist es nun zum Beispiel möglich, switch-Anweisungen auf String-Ausdrücke anzuwenden, mehrere Ausnahmetypen in einer einzigen catch-Klausel zusammenzufassen oder die Erzeugung von Objekten generischer Typen mit vereinfachter Syntax zu schreiben. Eine spezielle Form der try-Anweisung erlaubt es außerdem, Ressourcen wie Dateien oder Netzwerkverbindungen, die in der try-Anweisung geöffnet wurden, am Ende der try-Anweisung wieder automatisch zu schließen.

Trotz der geringen Änderungen schien eine Neuauflage dieses Buches angebracht, da damit der aktuelle Stand der Sprache dokumentiert wird und bei dieser Gelegenheit auch kleine Fehler eliminiert werden konnten. Als Neuerung gegenüber der letzten Auflage habe ich mich entschlossen, ein Kapitel über Annotationen einzufügen, also über Metainformationen, die an Programmelemente hängt und zur Laufzeit ausgewertet werden können. Annotationen gehören zwar nicht zu den Grundlagen der Programmierung, finden aber in letzter Zeit vor allem in Bibliotheken immer stärkere Verbreitung, so dass Java-Programmierer mit ihnen vertraut sein sollten.

Wie immer bedanke ich mich bei den Leserinnen und Lesern für Feedback und Verbesserungsvorschläge, für die ich immer ein offenes Ohr habe.

Linz, im April 2011

Hanspeter Mössenböck

Vorwort zur 3. Auflage

In den letzten Jahren hat sich Java nicht nur in der Industrie immer mehr durchgesetzt, sondern ist auch zur primären Ausbildungssprache an vielen Universitäten und Fachhochschulen geworden.

Seit Sommer 2004 gibt es eine neue Version von Java (Java 5, in älteren Dokumenten auch Java 1.5 genannt), die als wesentliche Neuerungen *generische Typen*, *Enumerationstypen*, eine neue Form der *for-Schleife*, Methoden mit *variabler Parameteranzahl*, *Auto-Boxing* sowie *statisch importierte Klassen* zur Verfügung stellt. Diese Spracherweiterungen machten eine Neuauflage des vorliegenden Buches nötig. Seiner Intention gemäß, ein einführendes Lehrbuch zu sein, werden die neuen Sprachmerkmale jedoch nur so weit beschrieben, als sie für Programmieranfänger von Bedeutung sind. Fortgeschrittene Konzepte wie *Attribute*, die ebenfalls Teil von Java 5 sind, werden nicht behandelt.

Die wichtigsten Neuerungen sind zweifellos generische Typen und Enumerationstypen, denen je ein eigenes Kapitel gewidmet ist. Die anderen Neuerungen wurden in die bestehenden Kapitel eingebaut.

Auf Wunsch vieler Leser wurde ein weiteres Kapitel hinzugefügt, das einen Überblick über die wichtigsten Klassen der Java-Bibliothek gibt. Dort werden Collection-Klassen beschrieben (Listen, Mengen und Abbildungen) sowie Klassen für die Ein- und Ausgabe. Das Kapitel bietet auch einen Einstieg in die Programmierung grafischer Benutzeroberflächen mit *Swing*.

Die für dieses Buch entwickelten einfachen Ein-/Ausgabeklassen `In` und `Out` können wie bisher von ssw.jku.at/JavaBuch/ heruntergeladen werden. Ferner enthält diese Webseite auch *Musterlösungen* zu den Übungsaufgaben am Ende jedes Kapitels.

Ich danke allen Lesern, die mir Feedback und Verbesserungsvorschläge geschickt haben, und bin auch in Zukunft für Fehlerhinweise und Verbesserungswünsche dankbar.

Linz, im Juli 2005

Hanspeter Mössenböck

Vorwort zur 2. Auflage

Die positive Aufnahme der ersten Auflage dieses Buches als Lehrbuch an zahlreichen Universitäten und Schulen hat zu Verbesserungsvorschlägen geführt, die nach einem ersten Nachdruck eine Neuauflage nahe legten.

In dieser zweiten Auflage wurden nicht nur Fehler korrigiert und Unklarheiten beseitigt, sondern vor allem auch die *objektorientierten Konzepte* von Java stärker betont. So gibt es jetzt ein neues Unterkapitel über *abstrakte Klassen* und eines über *Interfaces*. Auch die dynamische Bindung wurde mit weiteren Beispielen verdeutlicht.

Auf zahlreichen Wunsch gibt es jetzt auch Musterlösungen zu den Übungsaufgaben am Ende der einzelnen Kapitel. Damit aber Dozenten die Aufgaben in ihren Lehrveranstaltungen als Übungsbeispiele austeilen können, werden die Musterlösungen nicht allgemein zugänglich gemacht, sondern Dozenten, aber auch Leser, die das Buch im Selbststudium lesen, können die Musterlösungen beim Verlag (neumann@dpunkt.de) anfordern.

Ferner gibt es nun eine Webseite (www.ssw.uni-linz.ac.at/Misc/JavaBuch¹), auf der man begleitendes Material zu diesem Buch findet. Man kann von dieser Seite nicht nur die Klassen `In` und `Out` herunterladen, die für die Ein- und Ausgabe in diesem Buch verwendet werden, sondern man findet auch zahlreiche Verweise auf Java-Systeme, Java-Tutorials und weiterführende Dokumentationen. Insbesondere gibt es auf dieser Seite auch Folien im Powerpoint-Format, die ich für eine Lehrveranstaltung an der Universität Linz entwickelt habe und die sich an den Aufbau dieses Buches halten.

Für Verbesserungsvorschläge und Fehlermeldungen bin ich dankbar und erbitte eine Mitteilung an moessenboeck@ssw.uni-linz.ac.at.

Linz, im Januar 2003
Hanspeter Mössenböck

1. Die aktuelle URL lautet <http://ssw.jku.at/JavaBuch/>. Von dort können nun auch die Musterlösungen heruntergeladen werden.

Vorwort zur 1. Auflage

Als ich vor einiger Zeit vor der Aufgabe stand, eine einführende Programmier-Vorlesung mit Java zu halten, stellte ich fest, dass es zwar eine große Zahl von Büchern über Java-Programmierung gab, aber nur wenige, die sich als einführendes Lehrbuch eigneten. Die meisten Java-Bücher beginnen sofort mit Dingen wie Applets für das Internet, mit der Programmierung grafischer Benutzeroberflächen oder zumindest mit objektorientierten Konzepten. Wer noch nie programmiert hat, kann die Beispiele in diesen Büchern zwar nachcodieren und hat auf diese Weise auch Erfolgserlebnisse, er lernt aber nicht systematisch zu programmieren.

Ich ging also daran, ein eigenes Skriptum zu entwerfen, aus dem schließlich dieses Buch entstand. Mein Ziel war es, den Studenten¹ fundamentale Konzepte zu vermitteln, die sie auch in andere Sprachen übertragen konnten. Dazu gehören:

- *Algorithmisches Denken.* Wie formuliert man einen Algorithmus (d.h. ein Problemlösungsverfahren) für eine gegebene Aufgabe? Wie wählt man die richtigen Datenstrukturen und Anweisungsarten dafür? Wie führt man systematische Korrektheitsüberlegungen durch, die einem das Vertrauen geben, dass ein Programm auch wirklich das tut, was es soll?
- *Systematischer Programmentwurf.* Wie zerlegt man eine komplexe Aufgabe systematisch in kleinere Teilaufgaben, die dann als Bausteine (Pakete, Klassen und Methoden) einfach zu implementieren und modular zusammensetzen sind?
- *Moderne Softwarekonzepte.* Welche fundamentalen Konzepte gibt es in modernen Programmiersprachen? Dazu gehören zum Beispiel Rekursion, dynamische Datenstrukturen, Datenabstraktion, Vererbung, dynamische Bindung, Ausnahmebehandlung oder Parallelität. Die Beherrschung dieser (sprachunabhängigen) Konzepte zeichnet einen versierten Programmierer aus und gibt ihm einen Werkzeugkasten in die Hand, der ihn zum Meister macht und von Gelegenheitsprogrammierern unterscheidet.

1. Aus Gründen der Kürze und Lesbarkeit wird in diesem Buch nur die männliche Form von Personen verwendet. Selbstverständlich sind damit aber auch alle weiblichen Personen (Studentinnen, Programmierinnen, Benutzerinnen) gemeint.

- *Programmierstil.* Gute Programme sind nicht nur korrekt, sondern auch elegant, effizient und lesbar. Diese Eigenschaften sind besonders schwierig zu lehren und zu lernen. Andererseits sind sie für die Softwareentwicklung im größeren Umfang essenziell.

Dieses Buch ist keine Sprachspezifikation von Java, sondern ein Programmierlehrbuch, das Java als Werkzeug verwendet. Java ist eine moderne Programmiersprache, die vor allem im Bereich der Web-Programmierung häufig verwendet wird. Ihre Vorzüge machen sie aber auch für alle anderen Bereiche der Programmierung bestens geeignet. Java unterstützt moderne Konzepte der Softwaretechnik wie Sicherheit, Objektorientierung, Parallelität, Ausnahmebehandlung oder Komponentenorientierung. Ihre reichhaltige Bibliothek erlaubt die Erstellung grafischer Benutzeroberflächen, verteilter Anwendungen, Applikationen aus den Bereichen Multimedia, Computergrafik, E-Commerce und vieles andere.

Das vorliegende Buch geht allerdings kaum auf die Java-Bibliothek ein. Die Benutzung dieser Bibliothek ist Katalogwissen, das man jederzeit, auch über das Internet, beziehen kann. Für Programmieranfänger ist die Java-Bibliothek mit ihren Hunderten von Klassen und Tausenden von Methoden sogar eher verwirrend. Wir verwenden sie daher nur dort, wo es unumgänglich ist, nämlich für die Zeichenkettenverarbeitung und für einige mathematische Hilfsfunktionen. Für die Ein-/Ausgabe wurde für dieses Buch eine einfachere Bibliothek in Form der beiden Klassen `In` und `Out` entwickelt, die im Anhang A beschrieben wird und die man von `[JavaBuch]` laden kann.

Dieses Buch ist als Lehrbuch gedacht. Seine Kapitel sollten daher in der angegebenen Reihenfolge gelesen werden. Meist umfasst ein Kapitel genau den Stoff, der in einer Vorlesungseinheit von 90 Minuten bewältigt werden kann. Am Ende jedes Kapitels finden sich Übungsaufgaben, die den gelernten Stoff vertiefen und Lehrveranstaltungsleitern Gelegenheit für praktische Übungen geben.

Ich möchte an dieser Stelle meinen Assistenten Wolfgang Beer, Dietrich Birngruber, Markus Hof, Markus Knasmüller, Christoph Steindl und Albrecht Wöß danken, die die Übungen zu meiner Vorlesung über Jahre hinweg betreut und zahlreiche Übungsaufgaben zu diesem Buch beigesteuert haben. Wolfgang Beer hat außerdem geholfen, etliche Fehler im Manuskript dieses Buches zu entdecken.

Ferner danke ich den vom Verlag eingesetzten Begutachtern Prof. László Böszörményi, Prof. Dominik Gruntz und Prof. Martin Hitz für die zahlreichen nützlichen Anregungen, die sie zu diesem Buch beigetragen haben.

Nun möchte ich Sie als Leser einladen, die spannende Welt des Programmierens zu entdecken. Ich hoffe, dass Ihnen das Programmieren genauso viel Spaß und intellektuelle Befriedigung verschafft, wie das bei mir immer der Fall war.