

# Vorwort

Zunächst einmal bedanke ich mich bei Ihnen, dass Sie sich für dieses Buch entschieden haben, um sich über die Neuerungen von Java 8 zu informieren. In den nachfolgenden Kapiteln möchte ich Ihnen diese brandaktuelle Java-Version mit ihren umfangreichen Erweiterungen näherbringen. Insbesondere sind Lambda-Ausdrücke und das Stream-API wegweisende Neuerungen, durch die nun neben der objektorientierten auch die funktionale Programmierung in Java möglich wird. Auch die lange Zeit stiefmütterlich behandelte Datumsarithmetik wurde in Java 8 aufpoliert. Doch damit nicht genug: Die GUI-Technologie JavaFX wurde sowohl um neue Bedienelemente als auch um die Unterstützung für Darstellungen in 3D erweitert. JavaFX schickt sich an, Swing bald als GUI-Framework abzulösen.

## Wer sollte dieses Buch lesen?

Dieses Buch ist kein Buch für Programmierneulinge, sondern richtet sich an all diejenigen Leser, die einen fundierten Überblick über die mit Java 8 eingeführten Neuerungen erhalten wollen. Ich gehe also davon aus, dass Sie bereits einiges an Erfahrung mit Java mitbringen. Damit der Umstieg und das Nachvollziehen der Beispiele zu Java 8 leichter fällt, wird oftmals ein Vergleich zu einer herkömmlichen Lösung mit Java 7 dargestellt.

Ich setze zwar ein gutes Java-Grundwissen voraus, allerdings werden ausgewählte Themengebiete etwas genauer und gegebenenfalls einführend betrachtet, wenn dies das Verständnis der nachfolgenden Inhalte einfacher macht. Dies ist etwa für JavaFX der Fall: Dort beginne ich mit einer Darstellung der Grundlagen, weil die deutschsprachige Literatur gerade auf diesem Gebiet recht spärlich ist und sich sonst die Neuerungen aus JavaFX 8 nicht so gut nachvollziehen lassen würden.

Das Buch richtet sich im Speziellen an zwei Zielgruppen: Zum einen sind dies engagierte Hobbyprogrammierer, Informatikstudenten und Berufseinsteiger, die Java als Sprache beherrschen und nun neugierig auf die weitreichenden Änderungen in Java 8 sind. Zum anderen ist das Buch für erfahrenere Softwareentwickler und -architekten gedacht, die ihr Wissen ergänzen wollen, um für zukünftige Projekte abschätzen zu können, wann und in welchen Bereichen Java 8 eine gewinnbringende Alternative darstellen kann.

## Was vermittelt dieses Buch?

Sie als Leser erhalten neben Theoriewissen eine Vertiefung durch praktische Beispiele, sodass der Umstieg auf Java 8 in eigenen Projekten erfolgreich gemeistert werden kann. Der Fokus dieses Buchs liegt auf dem praktischen Nutzen und den zugrunde liegenden Konzepten. Zur Verdeutlichung werden vereinfachte Beispiele aus dem realen Programmiereralltag genutzt. Um den Rahmen des Buchs nicht zu sprengen, stellen die abgebildeten Programmlistings häufig nur Ausschnitte aus lauffähigen Programmen dar. Deren Name wird in Kapitälchenschrift, etwa `DATEPICKEREXAMPLE`, angegeben.

## Sourcecode und ausführbare Programme

Der Sourcecode kann auf der Webseite [www.dpunkt.de/java-8](http://www.dpunkt.de/java-8) heruntergeladen werden. Zudem befindet sich dort ein Eclipse-Projekt, über das sich alle Programme ausführen lassen. Beachten Sie bitte, dass Sie dazu mindestens Eclipse 4.3.2 mit dem speziellen Update zur JDK-8-Unterstützung benötigen (frei verfügbar unter <http://download.eclipse.org/eclipse/updates/4.3-P-builds/>).

## Aufbau dieses Buchs

**Kapitel 1** Einführend erhalten Sie einen kurzen Überblick zu Java 8 und seine wegweisenden Neuerungen.

**Kapitel 2** Kapitel 2 startet dann mit Lambdas, einer der bedeutsamsten Änderungen der Sprache seit der Einführung von Generics in Java 5. Lambdas führen zu einer vollkommen neuen Denkweise und bilden die Grundlage für die funktionale Programmierung mit Java.

**Kapitel 3** Kapitel 3 zeigt dann, wie sich Lambdas gewinnbringend mit den diversen Erweiterungen im Collections-Framework, insbesondere dem Stream-API, kombinieren lassen. Dort wird unter anderem eine mächtige Filter-Map-Reduce-Funktionalität bereitgestellt – ähnlich wie man dies von NoSQL-Datenbanken zur Verarbeitung großer Datenmengen (Stichwort: Big Data) kennt.

**Kapitel 4** Lange Zeit war die Verarbeitung von Datums- und Zeitangaben mit Java-Bordmitteln eher mühsam und zudem fehlerträchtig. Mit Java 8 ändert sich dies grundlegend. Das neue Date And Time API bereitet in seiner Nutzung viel Freude. Kapitel 4 gibt dazu einen Überblick.

**Kapitel 5** Nicht nur intern, sondern auch auf Seite der Benutzeroberflächen wurde in Java 8 einiges verbessert. JavaFX ist nun fester Bestandteil des JDKs und wurde an die Versionsnummer von Java angepasst. Neben Detailverbesserungen sind die zwei Bedienelemente `DatePicker` und `TreeTableView` sowie die Unterstützung von

3D-Darstellungen bedeutende Erweiterungen. Kapitel 5 beginnt mit einem allgemeinen Einstieg in JavaFX und geht danach auf die Besonderheiten von JavaFX 8 ein.

**Kapitel 6** Neben den in den vorangegangenen Kapiteln behandelten recht fundamentalen Änderungen enthält Java 8 noch eine Vielzahl weitere, zum Teil kleinere Verbesserungen, die aber allesamt das Programmiererleben deutlich erleichtern. Einige wesentliche werden in Kapitel 6 vorgestellt.

**Kapitel 7** Kapitel 7 rekapituliert noch einmal die Neuerungen in Java 8 und zieht ein Fazit. Zudem wagen wir dort einen Ausblick auf mögliche Funktionalitäten in JDK 9. Abgerundet wird das Kapitel durch eine Übersicht zu weiterer Literatur zu Java 8, Lambdas und funktionaler Programmierung.

**Anhang A** Anhang A liefert eine knappe Einführung in verschiedene Programmierparadigmen und insbesondere in die funktionale Programmierung, die nun mit JDK 8 Einzug in Java gehalten hat.

## Konventionen

### Verwendete Zeichensätze

In diesem Buch gelten folgende Konventionen bezüglich der Schriftart: Neben der vorliegenden Schriftart werden wichtige Textpassagen *kursiv* oder ***kursiv und fett*** markiert. Englische Fachbegriffe werden eingedeutscht groß geschrieben, etwa Event Handling. Zusammensetzungen aus englischen und deutschen (oder eingedeutschten) Begriffen werden mit Bindestrich verbunden, z. B. Plugin-Manager. Namen von Programmen sowie Entwurfsmustern werden bei ihrer Verwendung in KAPITÄLCHEN dargestellt. Sourcecode-Listings sind in der Schrift `courier` gesetzt, um zu verdeutlichen, dass dieser Text einen Ausschnitt aus einem Java-Programm darstellt. Auch im normalen Text werden Klassen, Methoden, Konstanten und Übergabeparameter in dieser Schriftart angegeben.

### Verwendete Klassen aus dem JDK

Werden Klassen des JDKs zum ersten Mal im Text erwähnt, so wird deren voll qualifizierter Name, d. h. inklusive der Package-Struktur, angegeben: Für die Klasse `String` würde dann etwa `java.lang.String` notiert. Dies erleichtert eine Orientierung und ein Auffinden im JDK. Dies gilt insbesondere, da in den Listings nur selten `import`-Anweisungen abgebildet werden. Im nachfolgenden Text wird zur besseren Lesbarkeit auf diese Angabe verzichtet und nur der Klassenname genannt.

Im Text beschriebene Methodenaufrufe enthalten in der Regel die Typen der Übergabeparameter, etwa `substring(int, int)`. Sind die Parameter in einem Kontext nicht entscheidend, wird mitunter auf deren Angabe aus Gründen der besseren Lesbarkeit verzichtet – das gilt insbesondere für generische Methoden.

## Verwendete Abkürzungen

Im Buch verwende ich die in der nachfolgenden Tabelle aufgelisteten Abkürzungen. Weitere Abkürzungen werden im laufenden Text in Klammern nach ihrer ersten Definition aufgeführt und anschließend bei Bedarf genutzt.

Abkürzung	Bedeutung
JDK	Java Development Kit
JLS	Java Language Specification
JRE	Java Runtime Environment
JSR	Java Specification Request
JVM	Java Virtual Machine
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
(G)UI	(Graphical) User Interface
IDE	Integrated Development Environment
XML	Extensible Markup Language

## Danksagung

Bei der Erstellung des Manuskripts konnte ich auf ein starkes Team an Korrekturlesern zurückgreifen. Es ist mir eine große Freude, von den unterschiedlichen Sichtweisen und Erfahrungen vieler Leute profitieren zu dürfen.

Den einen oder anderen Tipp erhielt ich von Stefan Bartels, Tim Böttmeyer, Sven Bremerstein, Philipp Dössegger, Peter Kehren, Dirk Lemmermann und Florian Messerschmidt. Insbesondere Merten Driemeyer und Dr. Carsten Kern haben mit verschiedenen hilfreichen Anmerkungen zu einer Verbesserung beigetragen. Außerdem danke ich Johannes Weigend für die hilfreichen Anregungen und Ergänzungsvorschläge sowie die Erlaubnis, seine Blog-Beiträge leicht abgewandelt nutzen zu dürfen.

Auch einige Kollegen meines Arbeitgebers Zühlke Engineering AG haben mich direkt oder indirekt unterstützt. Zunächst möchte ich meinem Chef Kai Schwidder für die Unterstützung und die Freiräume zur Ausgestaltung von internen Java-8-Kursen danken. Tatkräftig haben die Zühlkianer Wolfgang Giersche, Nikolaos Kaintantzis, Jörg Keller, Franziska Meyer, Sagi Nedunkanal und Christoph Süess durch ihre Kommentare zur Klarheit und Präzisierung beigetragen. Vielen Dank dafür!

Ein ganz besonderer Dank geht an Ralph Willenborg und Andreas Schöneck. Ralph hat mit seinen Adleraugen viele Tippfehler gefunden und sprachliche Verbesserungen angeregt. Andreas hat dieses Buch von seinen frühen Anfängen bis hin zum Endstadium durch viele Hinweise und Anregungen tatkräftig unterstützt. Vielen, vielen Dank für die Mühen und die schnellen Rückmeldungen auch zu später Stunde ;-)

Ebenso geht ein Dankeschön an das Team des dpunkt.verlags (Dr. Michael Barabas, Martin Wohlrab, Vanessa Wittmer und Birgit Bäuerlein) für die tolle Zusammenarbeit. Außerdem möchte ich mich bei Torsten Horn für die fundierte fachliche Durchsicht sowie bei Ursula Zimpfer für ihre Adleraugen beim Copy-Editing bedanken.

Abschließend geht ein lieber Dank an meine Frau Lilija für ihr Verständnis und die Unterstützung. Glücklicherweise musste sie bei der Erstellung dieses Buchs zu Java 8 einen weit weniger gestressten Autor ertragen, als dies früher bei der Erstellung meines Buchs »Der Weg zum Java-Profi« der Fall war.

## **Anregungen und Kritik**

Trotz großer Sorgfalt und mehrfachem Korrekturlesen lassen sich missverständliche Formulierungen oder sogar Fehler leider nicht vollständig ausschließen. Falls Ihnen etwas Derartiges auffällt, so zögern Sie bitte nicht, mir dies mitzuteilen. Gerne nehme ich auch sonstige Anregungen oder Verbesserungsvorschläge entgegen. Kontaktieren Sie mich bitte per Mail unter:

`michael_inden@hotmail.com`

Zürich und Aachen, im Mai 2014  
Michael Inden