

# 1 Einleitung

Am 18. März 2014 war es endlich so weit: Das zuvor mehrfach verschobene und lang erwartete Java 8 ist erschienen. Dieses Release enthält diverse wegweisende Erweiterungen und mit Lambda-Ausdrücken ein neues Sprachkonstrukt, das die funktionale Programmierung in Java erlaubt. Durch ein sorgfältiges API-Design von Massenoperationen auf Collections (Bulk Operations on Collections), wie Filterung, Transformation und Sortierung, lässt sich die funktionale gut mit der objektorientierten Programmierung verbinden. Dadurch ergeben sich vollkommen neue Gestaltungsmöglichkeiten, die jeder ambitionierte Java-Entwickler beherrschen sollte. Auch die lange Zeit stiefmütterlich behandelte Verarbeitung von Datumswerten wurde in Java 8 vollständig überarbeitet. Doch damit nicht genug: JavaFX als GUI-Technologie wurde aufpoliert. Darüber hinaus gibt es eine Vielzahl weiterer Funktionalitäten in Java 8 zu entdecken. Dieses Buch gibt einen Überblick über folgende wesentliche Erweiterungen in JDK 8:

## ■ **Lambda-Ausdrücke, Default-Methoden und Methodenreferenzen**

Zunächst werden wir mit Lambdas, Methodenreferenzen und Default-Methoden die Grundlagen zur funktionalen Programmierung mit Java kennenlernen.

## ■ **Bulk Operations on Collections**

Viele Aufgabenstellungen wie Filterungen und Transformationen erfordern den Einsatz von Collections und spezieller Algorithmen. Bis JDK 8 fehlte es allerdings an geeigneter Unterstützung, was sich nun ändert. Die neu eingeführten Streams bieten weitreichende Möglichkeiten zur Datenverarbeitung und zur Parallelisierung von Verarbeitungsschritten.

## ■ **Date And Time API**

Die Verarbeitung von Datums- und Zeitangaben war bis Java 8 immer mühselig und fehlerträchtig. Das neue Date And Time API vereinfacht das Ganze erheblich.

## ■ **JavaFX 8**

Viele GUIs, die auf Swing basieren, wirken altbacken. Abhilfe schafft hier JavaFX als neue Oberflächentechnologie. Wir betrachten Verbesserungen in JavaFX 8 und schauen kurz auf die Realisierung von Darstellungen in 3D.

## ■ **Diverse weitere Änderungen**

Im JDK findet sich eine Vielzahl größerer und kleinerer Verbesserungen und Neuerungen. Von den diversen weiteren Änderungen in JDK 8 behandle ich stellvertretend und überblicksartig verschiedenste Themen wie Parallel Array Sorting, Überarbeitung der Garbage Collection, die Unterstützung von Base64 u. v. m.

Die aufgelisteten Themen werden in jeweils separaten Kapiteln behandelt. Den Abschluss dieses Buchs bildet eine Beschreibung über bereits für JDK 7 vorgesehene Sprachfeatures, die es leider auch nicht in JDK 8 geschafft haben.

### **Hinweis: Umfang von JDK 8**

Schon der Releasetermin von JDK 7 wurde einige Male nach hinten verlegt. Das geschah unter anderem, weil die Fertigstellung verschiedener Features wie Lambda-Ausdrücke, Modularisierung, ein neues Date And Time API usw. viel länger dauerte als geplant. Schließlich wurden etliche Funktionalitäten aus Zeitmangel von JDK 7 auf JDK 8 verschoben.

Wer nun gedacht hätte, dass alle diese Funktionalitäten tatsächlich im JDK 8 enthalten sein würden, wird enttäuscht. Die Geschichte des Verschiebens und Abkündigens von Features wiederholt sich auch für JDK 8. Trotz mehrfacher Verzögerungen wurden einige Features leider nicht realisiert, wie z. B. die Modularisierung. Somit verbleibt als bedeutendste Neuerung in JDK 8 die Einführung von Lambda-Ausdrücken – allerdings sind deren Auswirkungen wirklich umfangreich und ziehen sich durch verschiedenste Teile des JDks.

### **Entdeckungsreise JDK 8 – Wünsche an die Leser**

Ich wünsche allen Lesern viel Freude mit diesem Buch zu Java 8 sowie einige neue Erkenntnisse und viel Spaß bei eigenen Experimenten mit JDK 8. Möge Ihnen der Umstieg auf Lambdas und die funktionale Programmierung mit meinem Buch ein wenig leichter fallen.

Lassen Sie uns nun die Entdeckungsreise durch JDK 8 und seine Neuerungen mit einem Einstieg in das Thema Lambda-Ausdrücke beginnen. Wenn Sie zunächst ein wenig mehr zu funktionaler Programmierung erfahren wollen, bietet sich ein Blick in den Anhang A an.