

# **Das Oracle JDK 8 ist etabliert und die Innovation geht weiter**

**Wolfgang Weigend, Peter Doschkinow**

ORACLE Deutschland B.V. & Co. KG

## **Bestandteile der Java-Plattform, OpenJDK, JDK 7, JDK 8, JDK 9, Entwicklungsumgebungen, Open Source Community, JavaFX**

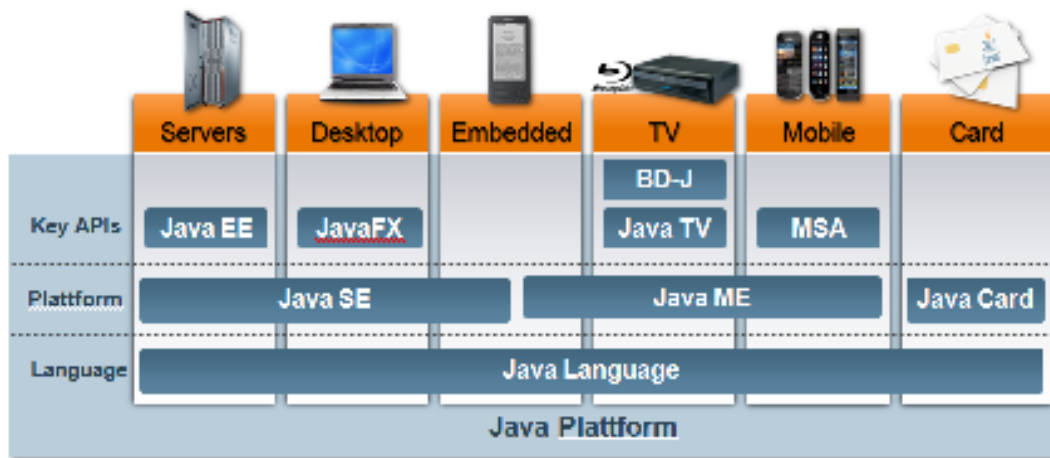
Java ist seit fast zwei Jahrzehnten im unternehmensweiten Einsatz und viele Firmen verwenden das JDK 7. Da steht bereits das JDK 8 für erste Erfahrungen bereit. Die Funktionalität des JDK 8 erstreckt sich über die neuen Lambda-Ausdrücke für funktionale Programmierung, die Nashorn-JavaScript-Engine für JavaScript-JVM-Interoperabilität und zahlreiche Neuerungen in der Java-Sprache, den Core-Bibliotheken, über Java-Security bis zur JVM. Weitere Verbesserungen sind im JDK 8 bei Type-Annotationen, dem Date & Time API und bei JDBC 4.0 zu finden. Die mit Java SE 8 eingeführten Compact-Profile stehen, als Subsets von Java SE, in drei Varianten zur Verfügung, um die Modularisierung für Java Embedded bereits im JDK 8 vorzubereiten. Doch damit nicht genug, JavaFX hat nach dem Start vom OpenJFX Projekt den Sprung ins JDK 8 gemacht und hat sich dort für Rich Internet Anwendungen manifestiert. Im Vortrag wird die gesamte Java Plattform Strategie besprochen, vom OpenJDK, Java SE 7, Java SE 8, Java SE 9 bis zur Java Enterprise Edition und die Zusammenarbeit mit der Java Community betont. Es geht im folgenden um die weitere Zukunft der Java-Sprache und der Java-Plattform.

## **Wie sich die Java Plattform verändert**

Der Vortrag beginnt mit der Java Plattform und dem bestehenden Ökosystem und unterstreicht die hohe Verbreitung der Java-Technologieplattform mit weltweit über 9 Millionen registrierten Java Entwicklern, 930 Millionen Java Runtime Environment (JRE) Downloads pro Jahr, mehr als 1.1 Milliarden Java Desktops, 3 Milliarden mobilen Endgeräten mit Java und 1.4 Milliarden Java Cards die Jahr für Jahr hergestellt werden. Java führt die Liste der am häufigsten verwendeten Programmiersprachen an und ist in allen Schulen und Universitäten fester Bestandteil der Ausbildung. Darin liegt auch die Stärke von Java: Die Community vergrößert sich und tauscht sich global über die Kodierung aus, sodass ohne komplexe Einarbeitung der Programmiercode gelesen, nachvollzogen und verändert werden kann. Aus diesem Grund wird der freie Zugang zu aktuellen Informationen für die Weiterentwicklung der Java Community uneingeschränkt zur Verfügung gestellt. Die einheitliche Entwicklungsbasis bildet dabei das offene und frei verfügbare OpenJDK, als zentrale Grundlage für die Aktivitäten der Java Standard Edition 7 (Java SE 7) und der Java Standard Edition 8 (Java SE 8).

Java ist der technologische Ausgangspunkt der meisten Hardware- und Software-Hersteller und bildet auch die Basis für die Oracle Fusion Middleware mit Oracle Fusion Applications. Dies verdeutlicht auch das Geschäftsmodell für die Java-Entwickler, das die anhand der gelernten Programmiersprache und der frei zugänglichen Java-Technologie die von ihnen erstellte Programmierlogik in Form von Anwendungen und neuen Services in die kommerzielle Vermarktung bringt. Die Verwendung von Java in Open Source Projekten macht einen Großteil der IT-Landschaft aus, bietet doch der kommerzielle Einsatz des Java-Programmier-Codes die Möglichkeit einer Einnahmequelle für die Entwickler. Bereits bei der Verwendung vom OpenJDK ist der Entwickler integraler Bestandteil einer klar umrissenen Java-Strategie. Die neuen OpenJDK Community Richtlinien wurden in Zusammenarbeit mit IBM, Eclipse Foundation, Oracle und Einzelpersonen erstellt und führen die Arbeiten des OpenJDK Governance Board weiter. Das Gremium hat Regeln aufgestellt, die den langfristigen Bestand und das Wachstum der OpenJDK Community fördern und sicherstellen, dass die Mitglieder in klarer und offener Weise agieren und die administrative Governance nach dem Leistungsprinzip erfolgt. So wird ein hohes Qualitätsniveau für das OpenJDK erreicht. An der Weiterentwicklung vom OpenJDK sind neben Oracle große Hersteller wie IBM, Apple, SAP, HP, Twitter, Azul Systems und VMware beteiligt. Alle setzen auf die einheitliche Java Plattform, die aus der Java Sprache, der Java Virtual Machine (JVM) und den Java APIs für unterschiedliche Funktionalitäts- und Hardware-Anforderungen wie Java Enterprise Edition (Java EE), Java Standard Edition (Java SE) und Java Micro Edition (Java ME) besteht (siehe Abbildung 1).

**Abbildung 1: Die Java Plattform**



### Java Virtual Machine und Java Development Kit

Durch die Akquisition von SUN Microsystems durch Oracle, sind die beiden Java Virtuellen Maschinen HotSpot und JRockit unter einem Dach. Die Hotspot JVM ist allgemein einsetzbar, parametrisierbar, qualitativ hochwertig und am Markt sehr stark verbreitet. Die JRockit JVM ist eine spezielle Server-JVM mit hoher Leistungsfähigkeit, die für den Serverbereich und die Oracle Fusion Middleware optimiert wurde. Mit JRockit verbessert sich das Laufzeitverhalten von Anwendungen die effizienter mit den Ressourcen: Speichergröße, Anzahl der Threads, Netzwerk, IO-Aktivitäten umgehen und durch deterministische Garbage-Collection und minimalen Pausenzeiten niedrige

Antwortzeiten erreichen. Zudem wurden alle Bestandteile der JRockit JVM (Code Generierung, Speicher Management, Thread Management, I/O, Reflection) optimiert. Die beiden JVMs JRockit und HotSpot wurden in einem mehrjährigen Entwicklungsprozess zu einer einheitlichen JVM unter dem Namen „HotRockit“ verschmolzen, die sich dann aus den besten Funktionsmerkmalen beider JVMs zusammensetzt. Die Arbeitsergebnisse sind inkrementell dem OpenJDK Projekt zugeflossen, wobei die JRockit Funktionalität den bisher größten Code-Beitrag zum OpenJDK darstellt. Vorhandene JRockit basierte Produkte, wie JRockit Mission Control, JRockit Real Time, JRockit Virtual Edition bleiben separat und werden weiterhin kommerziell lizenziert. Das Java Development Kit (JDK) und Java Runtime Environment (JRE) steht weiterhin kostenfrei zur Verfügung und Oracle wird eine offene Java Implementierung (OpenJDK) dauerhaft unterstützen, die reine Open Source Komponenten beinhaltet. Mit dem zentralen Einstiegspunkt vom OpenJDK gelangen die erreichten Ergebnisse in die Java SE 7, die Java SE 8 und die Java SE 9. Diese Java SE Versionen bieten eine höhere Entwicklerproduktivität, bessere Ausnutzung von Multi-Core Prozessoren und Unterstützung großer Hauptspeicher und Hochgeschwindigkeitsnetze. Wie bereits JDK 7, wurde auch JDK 8 nach der Fertigstellung auf [openjdk.java.net](http://openjdk.java.net) veröffentlicht. Der Funktionsumfang vom JDK kann mit NetBeans und GlassFish Server getestet werden. JDK 7 umfasst im wesentlichen Sprachverbesserungen aus dem Projekt Coin, die Concurrency und Collections Updates und die Unterstützung für dynamisch typisierte Sprachen. Das JDK 7 ist seit Mitte 2011 verfügbar, JDK 8 steht seit dem 18. März 2014 zur Verfügung. Die wichtigsten Inhalte des JDK 8 sind die Lambda-Ausdrücke, die Fertigstellung der konvergierenden JVM und das Projekt Nashorn mit einer verbesserten Interoperabilität von Java Script mit der JVM. Tabelle 1 zeigt eine Übersicht der Funktionalität vom JDK 8 mit den darin enthaltenen JDK Enhancement Proposals (JEP's). Vorab hatte sich das JCP-SE/EE-Executive-Committee in der Abstimmung mit teils deutlichen Mehrheiten für die zugrundeliegenden Java Specification Requests entschieden: JSR 334 „Small Enhancements to the Java Programming Language“, JSR 335 „Lambda Expressions for the Java Programming Language“, JSR 336 "Java 7 SE Release Contents“, JSR 337 "Java 8 SE Release Contents. Die kommende Java Plattform Modularisierung im Projekt Jigsaw wird voraussichtlich im JDK 9 enthalten sein, welches zwei Jahre nach der Veröffentlichung vom JDK 8 erscheinen soll.

### **Java Micro Edition**

Oracle beginnt mit der Modernisierung der Java Micro Edition (Java ME) Plattform und arbeitet mit der Java Mobile Community gemeinsam am Projekt Java ME.next, der evolutionären Weiterentwicklung der neuen Version von Java ME. Ziel ist es, die zugrundeliegende Sprachspezifikation zu aktualisieren und moderne Geräte-/Hardware-Funktionalität wie Near Field Communication, IP Multimedia Subsystem (IMS), Sensoren, Telefonie und Lokation durch neue Java APIs besser zu unterstützen. Wie auf dem Java Client werden sowohl native Java-Anwendungen, als auch auf Web-Technologien basierende Anwendungen unterstützt.

### **JavaFX**

Für die Entwicklung von Rich Internet Anwendungen (RIA) mit Unterstützung von Multimedia und modernen Hardware-GPUs (Graphics Processing Unit) wird JavaFX angeboten. Sie bestand früher aus der JavaFX Script Sprache, den JavaFX Script APIs, den Runtime-Libraries und lief auf der Java Virtual Machine. Für die neue JavaFX Plattform kommt ein Sprachwechsel. Damit wird seit JavaFX 2.0 JavaFX Script nicht mehr fortgeführt, sondern die JavaFX APIs sind vollständig in Java implementiert. Damit stehen viele Vorteile der Java Plattform wie Generics, Annotations und Multithreading unmittelbar auch für JavaFX zur Verfügung. Java Programmierer können JavaFX

nutzen, ohne eine weitere Programmiersprache lernen zu müssen und es wird leichter JavaFX innerhalb von Swing zu benutzen. Andere Scripting-Sprachen wie JRuby, Groovy und Scala, die auf der JVM laufen können für JavaFX Anwendungen benutzt werden und vergleichbare Merkmale wie JavaFX Script bereitstellen. JavaFX 8 ist vollständig mit Java SE 8 integriert.

**Tabelle 1: JDK 8 Funktionalität und JDK Enhancement Proposals (JEP's)**

JEP	Funktionalität
126	Lambda Expressions & Virtual Extension Methods
138	Autoconf-Based Build System
160	Lambda-Form Representation for Method Handles
161	Compact Profiles
162	Prepare for Modularization
164	Leverage CPU Instructions for AES Cryptography
174	Nashorn JavaScript Engine
142	Reduce Cache Contention on Specified Fields
165	Compiler Control
122	Remove the Permanent Generation
173	Retire Some Rarely-Used GC Combinations
136	Enhanced Verification Errors
143	Improve Contended Locking
147	Reduce Class Metadata Footprint
148	Small VM
171	Fence Intrinsic
153	Launch JavaFX Applications
101	Generalized Target-Type Inference
104	Annotations on Java Types
105	DocTree API
106	Add Javadoc to javax.tools
117	Remove the Annotation-Processing Tool (apt)
118	Access to Parameter Names at Runtime
120	Repeating Annotations
139	Enhance javac to Improve Build Speed
172	DocLint
103	Parallel Array Sorting

107	Bulk Data Operations for Collections
109	Enhance Core Libraries with Lambda
112	Charset Implementation Improvements
119	javax.lang.model Implementation Backed by Core Reflection
135	Base64 Encoding & Decoding
149	Reduce Core-Library Memory Usage
150	Date & Time API
155	Concurrency Updates
170	JDBC 4.2
127	Improve Locale Data Packaging and Adopt Unicode CLDR Data
128	BCP 47 Locale Matching
133	Unicode 6.2
113	MS-SFU Kerberos 5 Extensions
114	TLS Server Name Indication (SNI) Extension
115	AEAD CipherSuites
121	Stronger Algorithms for Password-Based Encryption
123	Configurable Secure Random-Number Generation
124	Enhance the Certificate Revocation-Checking API
129	NSA Suite B Cryptographic Algorithms
130	SHA-224 Message Digests
131	PKCS#11 Crypto Provider for 64-bit Windows
140	Limited doPrivileged
166	Overhaul JKS-JCEKS-PKCS12 Keystores

## Java SE Support

Die aktuelle Java SE Version von Oracle, mit JDK 8.x, ist in einem Zeitfenster von 3-4 Jahren nach Fertigstellung frei verfügbar. Bug-Fixes und kritische Security-Updates, werden in diesem Zeitraum kostenfrei zum Download bereitgestellt. Ist eine ältere Java-Version, wie beispielsweise das Oracle JDK 6 oder JDK 7 nicht mehr öffentlich erhältlich, so ist sie nur für Kunden mit einem Java SE Support Vertrag zugänglich, weil das Oracle JDK 6 und JDK 7 nicht End-of-Life ist, sondern weiterhin Sicherheitsverbesserungen erfährt. Abhängigkeiten von anderen Software-Herstellern und extrem langes Anwendungsversions-Management führen dazu, dass viele Java-Benutzer ihre unternehmenskritische Java-Anwendungen oft auf älteren Java-Versionen betreiben müssen. Nur mit einem Java SE Support Vertrag sind sie dann in der Lage, sich gegen bekannt gewordene Fehler oder

Security-Schwachstellen abzusichern. Seit zwei Jahren ist Java SE 7 aus dem kostenfreien Release-Update-Zeitfenster herausgefallen, so dass der Umstieg auf das Oracle JDK 8 empfohlen wird.

**Kontaktadresse:**

**Wolfgang Weigend**

**ORACLE Deutschland B.V. & Co. KG**

**Robert-Bosch-Strasse 5**

**63303 Dreieich**

**Telefon:** +49 (0) 6106-397-785

**Fax:** +49 (0) 6106-397-105

**E-Mail:** [wolfgang.weigend@oracle.com](mailto:wolfgang.weigend@oracle.com)

**Internet:** [www.oracle.com](http://www.oracle.com)

**Peter Doschkinow**

**ORACLE Deutschland B.V. & Co. KG**

**Riesstr. 25**

**D-80992 München**

**Telefon:** +49 (0) 1802672253

**Fax:** +49 (0) 1802672329

**E-Mail** [peter.doschkinow@oracle.com](mailto:peter.doschkinow@oracle.com)

**Internet:** [www.oracle.com](http://www.oracle.com)