

# **Oracle JDK 9 dreht die Innovationsschraube weiter**

**Wolfgang Weigend, Peter Doschkinow**

ORACLE Deutschland B.V. & Co. KG

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

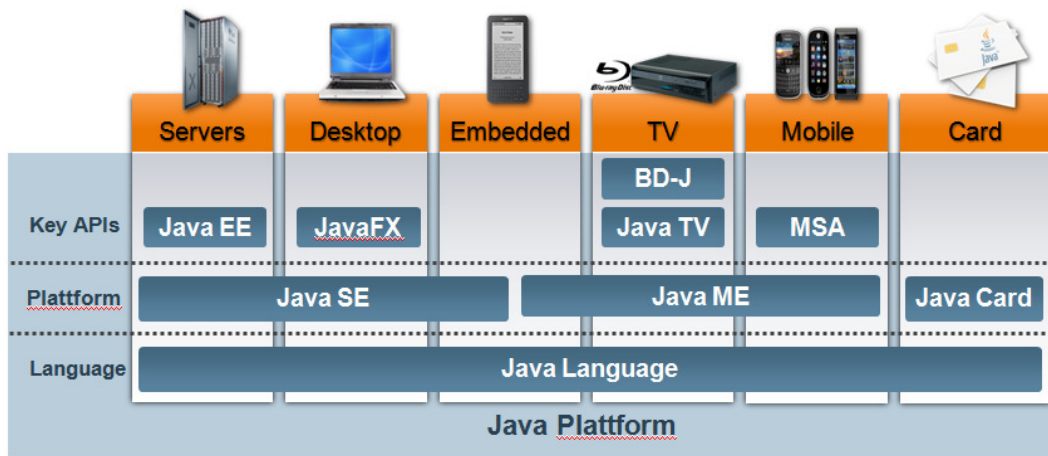
Java ist seit fast zwei Jahrzehnten im unternehmensweiten Einsatz und das neue JDK 9 wurde im September 2017 freigegeben. Die Java-Plattformmodularisierung mit dem Projekt Jigsaw ist die größte Änderung im Oracle JDK 9. Über die JDK-Enhancement-Vorschläge JEP 200 (Modulares JDK), JEP 201 (Modularer Source-Code) und JEP 220 (Modulare Laufzeit-Images) kamen die technischen Vorschläge ins JDK 9. Der JSR 376 (Java Platform Module System) schafft die Grundlage für die Referenzimplementierung vom OpenJDK Projekt Jigsaw. Diese Veränderungen lassen es zu, mit dem JDK 9 die gewünschte technische Paketierung von ausgewählter Java-Funktionalität selbst zu bestimmen. Das Projekt Jigsaw hat die primäre Aufgabe, das Design und die Implementierung eines Standard-Modulsystems für die Java-Plattform und für das JDK 9 bereitzustellen. Dabei soll die durchgängige, dynamische und einfache Anpassbarkeit der Java-SE-Plattform und dem JDK, auch für kleine Endgeräte berücksichtigt sowie die Verbesserung von Sicherheit und Wartbarkeit von Java-SE-Plattform-Implementierungen, speziell vom JDK, verwirklicht werden. Im Vortrag werden die JDK 9 Hauptmerkmale und die Unterschiede zu früheren Java-Versionen dargestellt, sowie die Bedeutung der Java-Anwendungsentwicklung hervorgehoben. Im Vortrag wird die gesamte Java Plattform Strategie besprochen, vom OpenJDK, 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 10 Millionen registrierten Java Entwicklern, eine Milliarde Java Runtime Environment (JRE) Downloads pro Jahr, mehr als eine Milliarde 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 9 (Java SE 9). 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 9. Diese Java SE Version bieten eine höhere Entwicklerproduktivität, bessere Ausnutzung von Multi-Core Prozessoren und Unterstützung großer Hauptspeicher und Hochgeschwindigkeitsnetze. Der Funktionsumfang vom Oracle JDK kann mit der NetBeans Entwicklungsumgebung und dem GlassFish Server getestet werden. JDK 9 umfasst neben dem Java Plattform Modulsystem weitere Merkmale, wie das neue Version-String-Format, ein verbessertes Werkzeugset (JVM, GC, JRE, jlink), neue Sicherheitsfunktionen, zusätzliche Tuning-Maßnahmen, Neuerungen in den Java-Core-Bibliotheken und weitere Sprachverbesserungen.

Tabelle 1 zeigt eine Übersicht der Funktionalität vom JDK 9 mit den darin enthaltenen JDK Enhancement Proposals (JEP's).

### **Java Micro Edition**

Oracle begann mit der Modernisierung der Java Micro Edition (Java ME) Plattform und arbeitet mit der Java Mobile Community gemeinsam an der evolutionären Weiterentwicklung von neuen Java-Versionen für moderne Geräte-/Hardware-Funktionalität, um 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. Durch das Oracle JDK 9 mit dem Java-Plattform-Modul-System und dem Werkzeug jlink ist es möglich, aus der Java SE kleinere Java-Lauzeitumgebungen zu erzeugen.

### **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. 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 9 ist vollständig mit Java SE 9 integriert.

**Tabelle 1: JDK 9 Funktionalität und JDK Enhancement Proposals (JEP's)**

| <b>JEP Funktionalität</b>                                    |
|--|
| 102 Process API Updates                                      |
| 110 HTTP 2 Client  |
| 143 Improve Contended Locking                                |
| 158 Unified JVM Logging                                      |
| 165 Compiler Control   |
| 193 Variable Handles   |
| 197 Segmented Code Cache                                     |
| 199 Smart Java Compilation, Phase Two                        |
| 200 The Modular JDK  |
| 201 Modular Source Code                                      |
| 211 Elide Deprecation Warnings on Import Statements          |
| 212 Resolve Lint and Doclint Warnings                        |
| 213 Milling Project Coin                                     |
| 214 Remove GC Combinations Deprecated in JDK 8               |
| 215 Tiered Attribution for javac                             |
| 216 Process Import Statements Correctly                      |
| 217 Annotations Pipeline 2.0                                 |
| 219 Datagram Transport Layer Security (DTLS)                 |
| 220 Modular Run-Time Images                                  |
| 221 Simplified Doclet API                                    |
| 222 jshell The Java Shell (Read-Eval-Print Loop)             |
| 223 New Version-String Scheme                                |
| 224 HTML5 Javadoc  |
| 225 Javadoc Search   |
| 226 UTF-8 Property Files                                     |
| 227 Unicode 7.0  |
| 228 Add More Diagnostic Commands                             |
| 229 Create PKCS12 Keystores by Default                       |
| 231 Remove Launch-Time JRE Version Selection                 |
| 232 Improve Secure Application Performance                   |
| 233 Generate Run-Time Compiler Tests Automatically           |
| 235 Test Class-File Attributes Generated by javac            |
| 236 Parser API for Nashorn                                   |
| 237 Linux/AArch64 Port                                       |
| 238 Multi-Release JAR Files                                  |
| 240 Remove the JVM TI hprof Agent                            |
| 241 Remove the jhat Tool                                     |
| 243 Java-Level JVM Compiler Interface                        |
| 244 TLS Application-Layer Protocol Negotiation Extension     |
| 245 Validate JVM Command-Line Flag Arguments                 |
| 246 Leverage CPU Instructions for GHASH and RSA              |
| 247 Compile for Older Platform Versions                      |
| 248 Make G1 the Default Garbage Collector                    |
| 249 OCSP Stapling for TLS                                    |
| 250 Store Interned Strings in CDS Archives                   |
| 251 Multi-Resolution Images                                  |
| 252 Use CLDR Locale Data by Default                          |
| 253 Prepare JavaFX UI Controls & CSS APIs for Modularization |
| 254 Compact Strings  |
| 255 Merge Selected Xerces 2.11.0 Updates into JAXP           |
| 256 BeanInfo Annotations                                     |
| 257 Update JavaFX/Media to Newer Version of GStreamer        |
| 258 HarfBuzz Font-Layout Engine                              |
| 259 Stack-Walking API  |

260 Encapsulate Most Internal APIs  
261 Module System  
262 TIFF Image I/O  
263 HiDPI Graphics on Windows and Linux  
264 Platform Logging API and Service  
  
265 Marlin Graphics Renderer  
  
266 More Concurrency Updates  
  
267 Unicode 8.0  
268 XML Catalogs  
269 Convenience Factory Methods for Collections  
270 Reserved Stack Areas for Critical Sections  
271 Unified GC Logging  
272 Platform-Specific Desktop Features  
273 DRBG-Based SecureRandom Implementations  
274 Enhanced Method Handles  
275 Modular Java Application Packaging  
276 Dynamic Linking of Language-Defined Object Models  
277 Enhanced Deprecation  
278 Additional Tests for Humongous Objects in G1  
279 Improve Test-Failure Troubleshooting  
280 Indify String Concatenation  
281 HotSpot C++ Unit-Test Framework  
282 jlink The Java Linker  
283 Enable GTK 3 on Linux  
284 New HotSpot Build System  
285 Spin-Wait Hints  
287 SHA-3 Hash Algorithms  
288 Disable SHA-1 Certificates  
289 Deprecate the Applet API  
290 Filter Incoming Serialization Data  
291 Deprecate the Concurrent Mark Sweep (CMS) Garbage Collector  
292 Implement Selected ECMAScript 6 Features in Nashorn  
294 Linux/s390x Port  
295 Ahead-of-Time Compilation  
297 Unified arm32/arm64 Port  
298 Remove Demos and Samples  
299 Reorganize Documentation

## Java SE Advanced Support

Die aktuelle Java SE Version von Oracle, mit JDK 9, 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, JDK 7 oder JDK 8 nicht mehr öffentlich erhältlich, so ist sie nur für Kunden mit einem Java SE Advanced Support Vertrag zugänglich, weil das Oracle JDK 7 und JDK 8 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.

**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)**