



Java aktuell



Sicherheit

Schützen Sie Ihre Anwendung
vor unbekannten Bedrohungen

DevOps

Worauf es bei einer erfolgreichen
DevOps-Kultur ankommt

GraalVM

Der heilige Gral
der Compiler?

Immer auf der sicheren Seite



ORAWORLD

Das e-Magazine für alle Oracle-Anwender!

EOUC
MEA
RACLE
SERGROUP
COMMUNITY

- Spannende Geschichten aus der Oracle-Welt
- Technologische Hintergrundartikel
- Leben und Arbeiten heute und morgen
- Einblicke in andere User Groups weltweit
- Neues (und Altes) aus der Welt der Nerds
- Comics, Fun Facts und Infografiken

Jetzt Artikel
einreichen
oder
Thema
vorschlagen!

Jetzt e-Magazine herunterladen
www.oraworld.org



Shannon ja bereits vorgeschlagen [6]: Einfach mal was wegwerfen und sonst eher mal langsam machen. Na prima! „Wir machen es wie immer nur noch doller“, wie ein Kollege von mir immer zu sagen pflegt. In diesem Sinne daher mein üblicher Aufruf: Wenn ihr wollt, dass sich was ändert, dann wirkt bitte aktiv an den Open-Source-Projekten mit! Alles, was ich bislang gespendet habe, ist nach kurzer Zeit angenommen worden. Wenn also jeder Leser nur ein einziges Feature liefert oder nur einen einzigen Bug fixt, haben wir in zwei Wochen mehr geändert als die EF in zwei Jahren...

- [1] <https://blogs.eclipse.org/post/mike-milinkovich/welcome-future-cloud-native-java> - „Welcome to the Future of Cloud Native Java“ / (Mike MILINKOVIC, Eclipse Foundation)
- [2] <https://jakartaone.org/> - JakartaOne Livestream (Eclipse Foundation)
- [3] <https://blogs.eclipse.org/post/tanja-obradovic/update-jakarta-ee-community-july-2019> - „Update for Jakarta EE community: July 2019“ (Tanja OBRADOVIC, Eclipse Foundation)
- [4] <https://jakarta.ee/compatibility/> - Jakarta EE Compatible Products (Eclipse Foundation)
- [5] <https://jakarta.ee/specifications/> - Jakarta EE Specifications (Eclipse Foundation)

- [6] <https://www.eclipse.org/lists/jakartaee-platform-dev/msg00648.html> - Oracle's position on Jakarta EE 9 (Bill SHANNON, Oracle)



Markus Karg

ProSeS BDE GmbH

markus@headcrashing.eu

Markus Karg ist Entwicklungsleiter eines mittelständischen Softwarehauses sowie Autor, Konferenzsprecher und Consultant. JAX-RS hat der Sprecher der Java User Group Goldstadt von Anfang an mitgestaltet, zunächst als freier Contributor, seit JAX-RS 2.0 als Mitglied der Expert Groups JSR 339 und JSR 370.



Unbekannte Kostbarkeiten des SDK Heute: try-with-resource-Statement mit „effectively final“ Ressourcen

Bernd Müller, Ostfalia

Das Java SDK enthält eine Reihe von Features, die wenig bekannt sind. Wären sie bekannt und würden sie verwendet, könnten Entwickler viel Arbeit und manchmal sogar zusätzliche Frameworks einsparen. Wir wollen in dieser Reihe derartige Features des SDK vorstellen: die unbekannten Kostbarkeiten.

Das Ziel dieser Ausgabe unserer unbekannten Kostbarkeiten ist nicht das Reduzieren von Schreibaufwand, sondern die Verbesserung von Code-Lesbarkeit. Mit Java 9 wurde das try-with-resource-Statement überarbeitet und lässt nun Ressourcen zu, die nicht innerhalb des Statements definiert sein müssen, wenn sie „final“ oder „effectively final“ sind.

Automatic Resource Management (ARM) mit Java 7

Java 7 verbesserte das automatisierte Verwalten von Ressourcen mit dem sogenannten try-with-resource-Statement, dessen konkrete syntaktische Form weder „with“ noch „resource“ enthält. Es dürfte allen Java-Entwicklern bekannt sein und breite Verwendung finden, da es verwendeten Ressourcen automatisch schließt und uns so Arbeit abnimmt. Der in *Listing 1* beschriebene Code-Ausschnitt verdeutlicht dies.

Die Klassen `Scanner` und `PrintWriter` implementieren beide das Interface `AutoClosable`, sodass ihre Instanzen automatisch beim Verlassen des Blocks geschlossen werden. Vor Java 7 musste im „finally“-Teil des try-Statements zunächst geprüft werden, ob die beiden Ressourcen tatsächlich geöffnet sind, um sie dann zu schließen. Dies kann nun komplett entfallen.

Das Beispiel kopiert lediglich eine Datei, wofür deutlich besserer Alternativen existieren. Es soll hier nur dazu dienen, die Verwendung der beiden Variablen `scanner` und `writer` zu verdeutlichen. Sie müssen innerhalb der runden Klammern des Statements erzeugt und mit einem Semikolon getrennt werden. Der Autor ist der Meinung, dass dies der Lesbarkeit des Codes nicht guttut.

```
try (Scanner scanner = new Scanner(new File(FILE_TO_READ));
    PrintWriter writer = new PrintWriter(new File(FILE_TO_WRITE))) {
    while (scanner.hasNext()) {
        writer.print(scanner.nextLine());
    }
}
```

Listing 1

```
Scanner scanner = new Scanner(new File(FILE_TO_READ));
PrintWriter writer = new PrintWriter(new File(FILE_TO_WRITE));
try (scanner; writer) {
    while (scanner.hasNext()) {
        writer.print(scanner.nextLine());
    }
}
```

Listing 2

Finale Ressourcen

Seit Java 9 dürfen die Ressourcen des try-with-resource-Statements außerhalb des Statements definiert werden, wenn sie „final“ oder „effectively final“ sind. Wenn man das obige Beispiel derart überarbeitet, ändert sich der Code wie in *Listing 2* abgebildet.

Der Autor findet, dass diese Version besser lesbar ist und damit eine höhere Code-Qualität aufweist und hofft, dass die Leser diese Meinung teilen.

Zusammenfassung

Seit Java 9 verlangt das try-with-resource-Statement nicht mehr, dass die verwendeten Ressourcen innerhalb der runden Klammern deklariert werden, sondern erlaubt deren Deklaration auch außerhalb des Statements, wenn die Variablen final oder effectively final deklariert werden. Das kommt der Lesbarkeit des Codes zugute.



Bernd Müller

Ostfalia

bernd.mueller@ostfalia.de

Nach seinem Studium der Informatik und der Promotion arbeitete Bernd Müller für die IBM und die HDI Informationssysteme. Er ist Professor, Geschäftsführer, Autor mehrerer Bücher zu den Themen JSF und JPA, sowie Speaker auf nationalen und internationalen Konferenzen. Er engagiert sich im iJUG und speziell in der JUG Ostfalen.



Data Analytics 2020

28. & 29. April | in Düsseldorf

DOAG

analytics.doag.org





Early Bird
bis 21. Januar 2020

JavaLand²⁰²⁰

17. - 19. März 2020 in Brühl bei Köln

Ab sofort Ticket & Hotel buchen!

www.javaland.eu

