

## Fragen, Antworten und Kommentare zur aktuellen Vorlesung

Frage: Was sind Java Agents?

Java Agents sind normale Java-Programme, die als Besonderheit den Byte-Code neu geladener Klassen verändern können. Generell liegen Klassen in systemunabhängigen Byte-Code vor und werden bei der ersten Nutzung geladen und passend zur genutzten Maschine umgesetzt. Dies erklärt auch die langen Startzeiten beim Start von Java-Programmen. Mit dem genutzten Class-Loader können Java-Agents zusammenarbeiten und dabei das Verhalten verändern. Ein Ansatz ist z. B., dass am Anfang eines jeden Methodenaufrufs Programmcode ergänzt wird, mit dem die Aufrufinformationen in der Konsole ausgegeben werden. Die Eingriffe können noch viel weitergehen und das gesamte Verhalten beeinflussen. Der Ansatz wird z. B. von JPA-Werkzeugen genutzt, die eine enge teil-automatisierte Anbindung an eine Datenbank ermöglichen. In Richtung C und C++ ist der Ansatz mit der Nutzung von Pre-Compilern verwandt.

(graue Blöcke sind Fragen, die ich mir gestellt hatte und die ich mir einfach von einer KI beantworten lassen habe. Die generelle Korrektheit habe ich geprüft, aber nicht alle Links validiert)

Sie

Nutzt Mockito Java Agents? Wie kann die zugehörige Warning vermieden werden?

Copilot

**das web**wird verwendet

Ja, Mockito nutzt Java Agents. [Es verwendet die Byte Buddy-Bibliothek, um Klassen zur Laufzeit zu manipulieren und Mock-Objekte zu erstellen](#)<sup>1</sup>.

[Die Warnung, die Sie sehen, könnte mit der dynamischen Agentenladung in neueren Java-Versionen zusammenhängen](#)<sup>2</sup>. [Ab Java 9 wurde eine neue Funktion eingeführt, die das dynamische Laden von Java Agents ermöglicht, aber diese Funktion ist standardmäßig deaktiviert](#)<sup>2</sup>. [Wenn Mockito versucht, einen Agenten zur Laufzeit zu laden, kann dies zu einer Warnung führen](#)<sup>2</sup>.

[Um diese Warnung zu vermeiden, können Sie die JVM-Option -XX:+EnableDynamicAgentLoading hinzufügen, wenn Sie Ihre Tests ausführen](#)<sup>2</sup>. [Bitte beachten Sie, dass dies nur die Warnung verbirgt und in zukünftigen Java-Versionen möglicherweise nicht mehr funktioniert](#)<sup>2</sup>.

Eine andere Möglichkeit, die Warnung zu vermeiden, besteht darin, Byte Buddy als Agenten (d.h. über `-javaagent`) anstelle von als Bibliothek (d.h. [über `--class-path` oder `--module-path`](#)) zu laden<sup>2</sup>. [Allerdings machen dies weder IDEs noch Build-Tools besonders einfach, so dass eine ordnungsgemäße Behebung des Problems wahrscheinlich warten muss, bis IDEs und Build-Tools "aufholen"](#)<sup>2</sup>.

Bitte beachten Sie, dass Sie immer die neueste Version von Mockito und Ihren anderen Abhängigkeiten verwenden sollten, um sicherzustellen, dass Sie von den neuesten Verbesserungen und Fehlerbehebungen profitieren.

Weitere Informationen

1 github.com 2 stackoverflow.com 3 baeldung.com 4 baeldung.com  
5 stackoverflow.com 6 stackoverflow.com — weniger anzeigen

Frage: Beim Mocken der Klasse XY erhalte ich folgende Fehlermeldung, was kann ich machen?

[org.mockito.exceptions.base.MockitoException](#):

Mockito cannot mock this class: interface java.util.List.

If you're not sure why you're getting this error, please open an issue on GitHub.

Java : 23

JVM vendor name : Azul Systems, Inc.

JVM vendor version : 23.0.1+11

JVM name : OpenJDK 64-Bit Server VM

JVM version : 23.0.1+11

JVM info : mixed mode, sharing

OS name : Windows 10

OS version : 10.0

...

Underlying exception : [org.mockito.exceptions.base.MockitoException](#): Could not modify all classes [interface java.util.List, interface java.lang.Iterable, interface java.util.SequencedCollection, interface java.util.Collection]

at qsMockito/bsp.MeinInterfaceMockTest.setUp([MeinInterfaceMockTest.java:25](#))

at java.base/java.lang.reflect.Method.invoke([Method.java:580](#))

at java.base/java.util.ArrayList.forEach([ArrayList.java:1597](#))

at java.base/java.util.ArrayList.forEach([ArrayList.java:1597](#))

Caused by: [org.mockito.exceptions.base.MockitoException](#): Could not modify all classes [interface java.util.List, interface java.lang.Iterable, interface java.util.SequencedCollection, interface java.util.Collection]

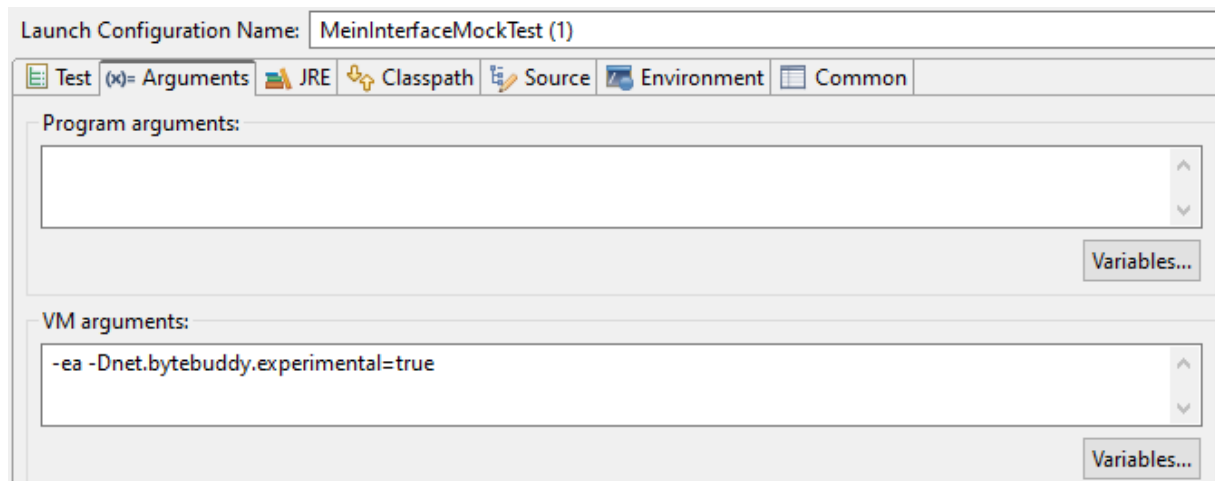
Antwort: Generell schwierig, da Mockito von ByteBuddy abhängt und Zugriffe sowie Veränderungen von Klassen, die nicht selbst geschrieben wurden immer mehr eingeschränkt werden. Wenn solche Fehlermeldungen genauer betrachtet werden, enthalten Sie manchmal einen Teil der folgenden Form.

... 4 more

Caused by: [java.lang.IllegalArgumentException](#): Java 23 (67) is not supported by the current version of Byte Buddy which officially supports Java 22 (66) - update Byte Buddy or set net.bytebuddy.experimental as a VM property

Die sauberste Antwort ist dann, eine ältere Java-Version zu installieren, nur die JDK Compliance beim Java Compiler zu setzen reicht leider nicht, und dann diese Version für das Projekt zu nutzen. In Eclipse muss das weitere JDK bekannt gemacht werden, damit es in Projekten genutzt werden kann. Alternativ ist der Vorschlag aus der Meldung oft ein Workaround, den folgenden Parameter an die VM zu übergeben:

-Dnet.bytebuddy.experimental=true



Sollen Standard-Java-Interface gemockt werden, empfiehlt es sich im Projekt auf das Java-Module-System, konkret module-info.java zu verzichten.