

# Why and How to Upgrade to Java 17

Johan Janssen – Sanoma Learning

# Agenda

- » Why upgrade?
- » Release cadence
- » How to upgrade
- » Upgrade guidance per Java version

?

# Questions

- » [@johanjanssen42](https://twitter.com/johanjanssen42) 
- » <https://www.linkedin.com/in/johanjanssen2001>
- » <https://github.com/johanjanssen/JavaUpgrades>

# Background

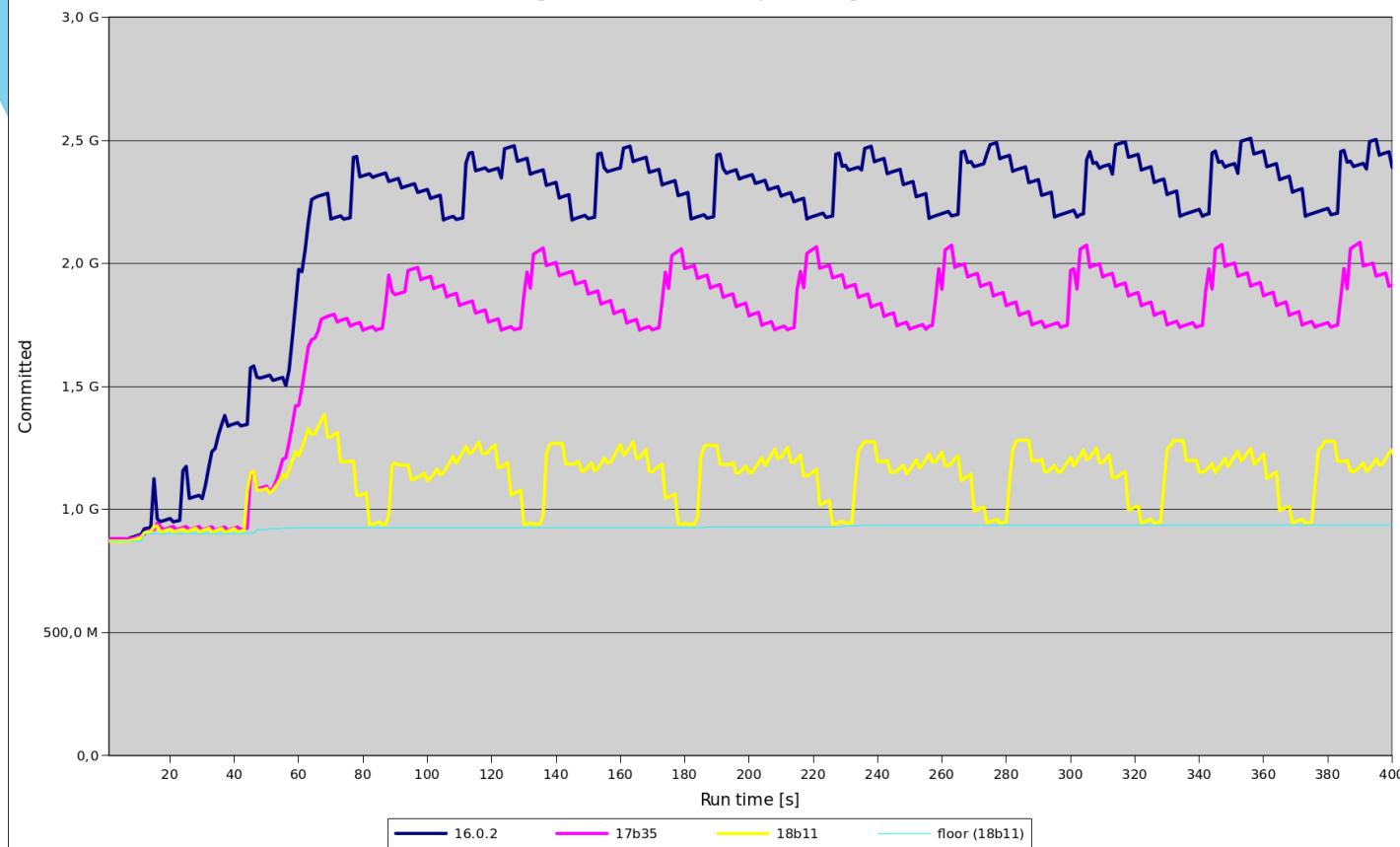
- » Software architect at Sanoma Learning
- » Upgrading Java for years
- » Applications with hundreds of thousands of users a day

# Why upgrade?

- » Performance improvements
- » Security fixes
- » End of support, no more bugfixes
- » ...
- » Use cool new features such as Records and Pattern Matching
- » Keep and attract employees

### GC component committed memory

BigRAMTester 20GB Heap/16M regions



From: <https://tschatzl.github.io/2021/09/16/jdk17-g1-parallel-gc-changes.html>

# Why not upgrade?

- » Update everything (regularly)
  - ◊ Local machine
  - ◊ Build environment
  - ◊ Deploy environments
  - ◊ Libraries, frameworks, tools
  - ◊ Application
- » Seen as a lot of work
- » Scared of Java 9 modules
- » Support clients with old Java versions

# Multi release JAR

- » `java/com/example/Application`
- » `java/com/example/Student`
- » `java17/com/example/Student`

# Multi release JAR

- » All versions should implement the same public API
- » Unfortunately not enforced by build tools
- » IDE gives warning
- » Validate the JAR's content with Java 17:
  - ◊ `jar --validate`





## Why this session?

- » Upgrading is seen as challenging
  - » Estimating the amount of work is difficult
  - » Projects stay on old Java versions
- 

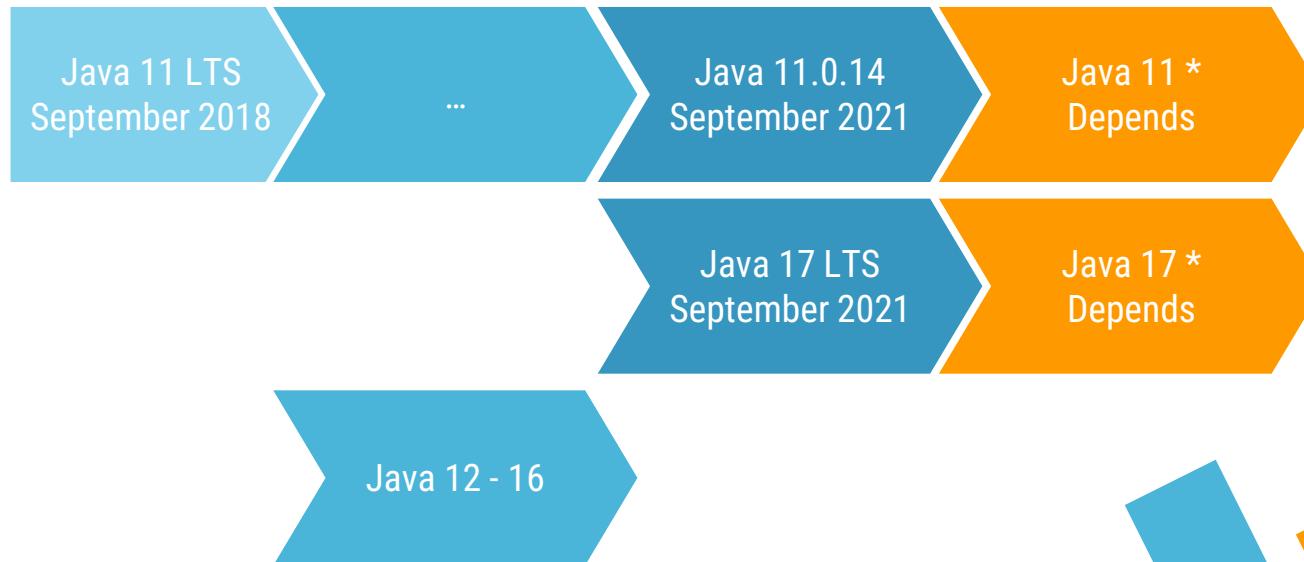


# Goal of the session

- » Share recurring issues and solutions
- » Provide guidance
- » Ease the migration
- » *Maven is used as an example*



# Releases





# What to upgrade?

Application code

Dependencies

Java

# What to upgrade?

Application code

Dependencies



Java



delete X

# Deprecations

- » java.xml.bind (JAXB)
- » Java 8
- » Java 9 - DEPRECATED
- » Java 10 - DEPRECATED
- » Java 11 - REMOVED

# Removed

- » Root certificates
- » Encryption algorithms
- » JVM flags/options (-XX or -X)
- » Garbage collector (For instance CMS in 14)
- » Tools (JMC)



# New APIs in Java 17

Comparing Java 17 (17-ea+17-1401-open) with Java 8 (8.0.275-adpt).

Element	Modification
▪ java.applet	
● Applet	+ deprecated + forRemoval
● ApplicationContext	+ deprecated + forRemoval
● AppletStub	+ deprecated + forRemoval
● AudioClip	+ deprecated + forRemoval
▪ java.awt.color	
● ICC_Profile	
● finalize()	+ deprecated + forRemoval
▪ java.awt.datatransfer	
● DataFlavor	
○ allHtmlFlavor	+ final
○ fragmentHtmlFlavor	+ final
○ selectionHtmlFlavor	+ final
▪ java.awt.desktop	added
▪ java.awt.dnd	
● DragSourceAdapter	
● DragSourceAdapter()	- public + protected
● DragSourceContext	
● DragSourceContext(DragGestureEvent, Cursor, Image,...)	added
● DragSourceContext(DragSourceContextPeer,...)	removed
● DragSource	
● createDragSourceContext(DragGestureEvent, Cursor,...)	added
● createDragSourceContext(DragSourceContextPeer,...)	removed

## Detailed release info

- » <https://openjdk.java.net/>
- » <https://openjdk.java.net/projects/jdk/17>
  - ◊ Available from Java 10
- » <https://www.oracle.com/java/technologies/javase/16-relnote-issues.html>
  - ◊ Available from Java 10

# Oracle JDK Migration Guide

- » <https://docs.oracle.com/en/java/javase/17/migrate/getting-started.html>

# Running multiple JDK's

- » Replace current one
- » Change JAVA\_HOME
- » SDKMAN!
- » Docker

# Dockerfile

```
FROM maven:3.8.1-openjdk-17-slim
```

```
ADD . /yourproject
```

```
WORKDIR /yourproject
```

```
RUN mvn test --fail-at-end
```

# Docker

» docker build -t javaupgrade .

# Rolling out a new Java version







# Preparations

- » IDE with support for Java 16 / 17
- » Upgrade build tool and plugins
  - » Compiler plugin
  - » Unit test plugin (surefire)
  - » Integration test plugin (failsafe)
  - » ...

# Configure Maven compiler from Java 9

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>3.8.1</version>
    <configuration>
        <release>17</release>
    </configuration>
</plugin>
```

# Upgrade dependencies

- » On current or new Java version
- » Upgrade the dependencies you created as well
- » **Check for new artifact/group names**

# Upgrade dependencies

- » Maven Versions Plugin
  - ◊ mvn versions:display-dependency-updates
- » Gradle Versions Plugin
  - plugins {  
 id "com.github.ben-manes.versions" version "\$version"  
}  
◊ gradle dependencyUpdates -Drevision=release

# New package names

- » javax.xml.bind:jaxb-api
  - ◊ Latest: 2.4.0-x August 2018
- » jakarta.xml.bind:jakarta.xml.bind-api
  - ◊ Latest: 3.0.0 November 2020

# Old GroupIds Alerter

- » Maven Plugin
- » Gradle Plugin

# Cooking

- » Compile source code
- » Run unit tests
- » Package application
- » Test running application



# Java 11

- » JavaFX removed
  - ◊ Use OpenJFX build: Gluon
  - ◊ Use JDK with OpenJFX: Liberica JDK / ojdkbuild
  - ◊ Use Maven dependencies: <https://openjfx.io/>
- » JDK fonts removed
  - ◊ apt install fontconfig
  - ◊ Optional: libfreetype6 fontconfig fonts-dejavu

# Java 11

- » Java Mission Control removed
  - ◊ AdoptOpenJDK and Oracle offer JDK Mission Control builds
- » JEP 320: Remove the Java EE and CORBA Modules

# Change

» `import javax.xml.bind.*;`



» `import jakarta.xml.bind.*;`

» Dependencies

Module	Packages	Replacement groupId	Replacement artifactId
java.activation	javax.activation	com.sun.activation	jakarta.activation
java.xml.ws.annotation	javax.annotation	jakarta.annotation	jakarta.annotation-api
java.transaction	javax.transaction	jakarta.transaction	jakarta.transaction-api
java.xml.bind	javax.xml.bind.*	jakarta.xml.bind com.sun.xml.bind	jakarta.xml.bind-api jaxb-impl
java.xml.ws	javax.jws.* javax.xml.soap javax.xml.ws.*	jakarta.xml.ws com.sun.xml.ws	jakarta.xml.ws-api jaxws-rt
java.corba	javax.activity javax.rmi.*	No official standalone glassfish-corba	

# Java 15

- » 15: JEP 372: Remove the Nashorn JavaScript Engine

```
<dependency>
    <groupId>org.openjdk.nashorn</groupId>
    <artifactId>nashorn-core</artifactId>
    <version>15.2</version>
</dependency>
```

# Java 16

- » JEP 396: Strongly Encapsulate JDK Internals by Default

# Solution

- » Upgrade dependencies
- » Lombok 1.18.20 supports Java 16

# Java 16 workaround

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.8.1</version>
  <configuration>
    <fork>true</fork>
    <compilerArgs> <!-- See next slide-->
    </compilerArgs>
  </configuration>
</plugin>
```

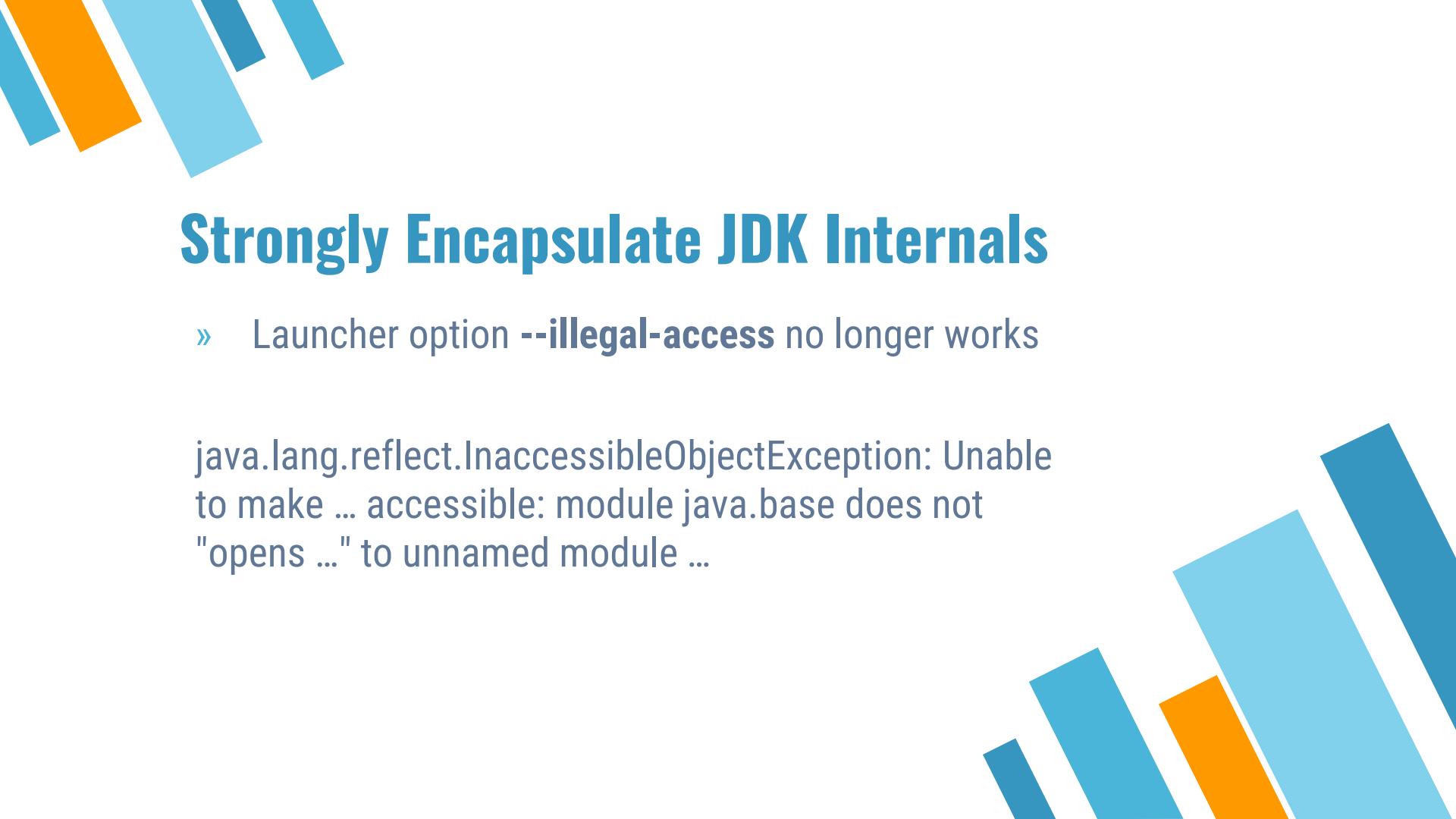
```
<arg>-J--add-opens=jdk.compiler/com.sun.tools.javac.comp=ALL-UNNAMED</arg>
<arg>-J--add-opens=jdk.compiler/com.sun.tools.javac.file=ALL-UNNAMED</arg>
<arg>-J--add-opens=jdk.compiler/com.sun.tools.javac.main=ALL-UNNAMED</arg>
<arg>-J--add-opens=jdk.compiler/com.sun.tools.javac.model=ALL-UNNAMED</arg>
<arg>-J--add-opens=jdk.compiler/com.sun.tools.javac.parser=ALL-UNNAMED</arg>
<arg>-J--add-opens=jdk.compiler/com.sun.tools.javac.processing=ALL-
UNNAMED</arg>
<arg>-J--add-opens=jdk.compiler/com.sun.tools.javac.tree=ALL-UNNAMED</arg>
<arg>-J--add-opens=jdk.compiler/com.sun.tools.javac.util=ALL-UNNAMED</arg>
<arg>-J--add-opens=jdk.compiler/com.sun.tools.javac.jvm=ALL-UNNAMED</arg>
```

# Java 16: Lombok

```
... class lombok.javac.apt.LombokProcessor (in unnamed module @0x21bd20ee) cannot access class com.sun.tools.javac.processing.JavacProcessingEnvironment (in module jdk.compiler) because module jdk.compiler does not export com.sun.tools.javac.processing to unnamed module ...
```

# Java 17

- » 398: Deprecate the Applet API for Removal
- » 410: Remove the Experimental AOT and JIT Compiler
- » 403: Strongly Encapsulate JDK Internals
- » 411: Deprecate the Security Manager for Removal



# Strongly Encapsulate JDK Internals

- » Launcher option **--illegal-access** no longer works

java.lang.reflect.InaccessibleObjectException: Unable  
to make ... accessible: module java.base does not  
"opens ..." to unnamed module ...

- » Upgrade dependencies
- » Use other JDK API's
- » Last resort
  - ◊ **--add-opens**

# Example --add-opens

```
<plugin>
  <artifactId>maven-surefire-plugin</artifactId>
  <configuration>
    <argLine>
      --add-opens java.base/java.lang=ALL-UNNAMED
      ...
    </argLine>
  </configuration>
</plugin>
```

# Mockito

```
public enum ExampleEnum {  
    TEST {  
        public String retrieve() {  
            return "test";  
        }  
    };  
}
```

# Mockito

```
@Mock  
private ExampleEnum exampleEnum;
```

# Mockito issue

You are seeing this disclaimer because Mockito is configured to create inlined mocks.

...

Underlying exception :

org.mockito.exceptions.base.MockitoException: Could  
not modify all classes

...

» Fixed in version 3.11.2

# Kotlin

- » Kotlin Maven plugin

```
<configuration>
  <jvmTarget>17</jvmTarget>
</configuration>
```

# Kotlin

[ERROR] Failed to execute goal  
org.jetbrains.kotlin:kotlin-maven-plugin:1.5.20-  
RC:compile (compile) on project ...: Compilation failure

**[ERROR] Unknown JVM target version: 17**

[ERROR] Supported versions: 1.6, 1.8, 9, 10, 11, 12, 13,  
14, 15, 16

# Gradle

> Task :buildSrc:compileGroovy

startup failed:

General error during semantic analysis: **Unsupported class file major version 61**

» <https://github.com/gradle/gradle/issues/16857>

# Tests: all Java versions

An error has occurred in JaCoCo Aggregate report generation. Error while creating report: Error while analyzing ... **Unsupported class file major version 61**

# Various: all Java versions

- » Execution default of goal org.pitest:**pitest-maven:1.4.10:mutationCoverage** failed:  
Unsupported **class file major version 61**
- » Execution repackage of goal  
org.springframework.boot:**spring-boot-maven-plugin:2.2.10.RELEASE:repackage** failed:  
Unsupported **class file major version 61**

# Class file major version

- » class file major version 61 is used for Java 17





# Summary

- » Amount of work
- » Keep dependencies up to date
- » Incremental steps to track progress
- » Everyone is responsible, you as well
- » Start upgrading now!



# THANK YOU!

## Any questions?

- » @johanjanssen42 
  - » <https://www.linkedin.com/in/johanjanssen2001>
  - » <https://github.com/johanjanssen/JavaUpgrades>
- 