



Pearson

Java 9 to 17: The New Feature Benefits



TM Henri Tremblay

Managing Director, Head of TS Imagine Canada
TS Imagine

[@henri_tremblay](https://twitter.com/henri_tremblay)

Henri Tremblay

EASYSOCK

ObjENESIS

EHCACHE



- More or less made possible class mocking and proxying
- Coined the term “partial mocking”



Oracle
Groundbreaker
Ambassador
Alumni



Good old

8

New & shiny

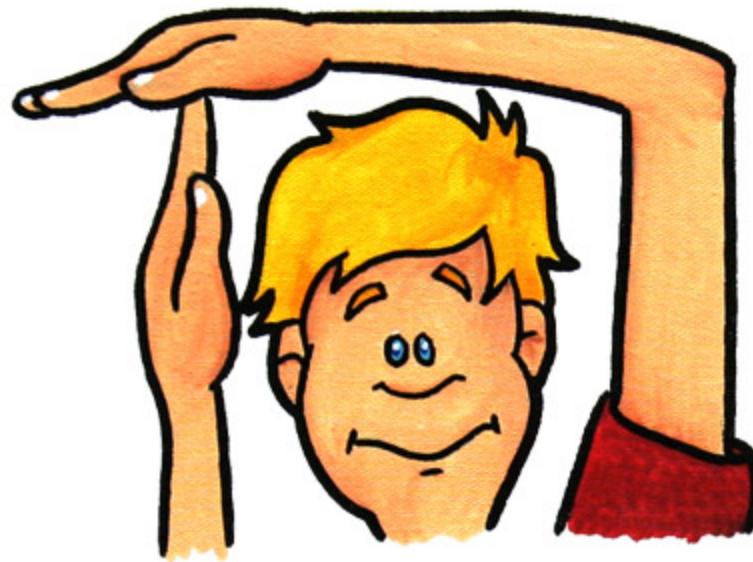
17



[Source code](#)

<https://github.com/henri-tremblay/java91011beyond>

- > Which version of Java are you currently using?
 - + Java 6 or less
 - + Java 7
 - + Java 8
 - + Java 9-10
 - + Java 11
 - + Java 12-16
 - + Java 17
 - + Java 18+

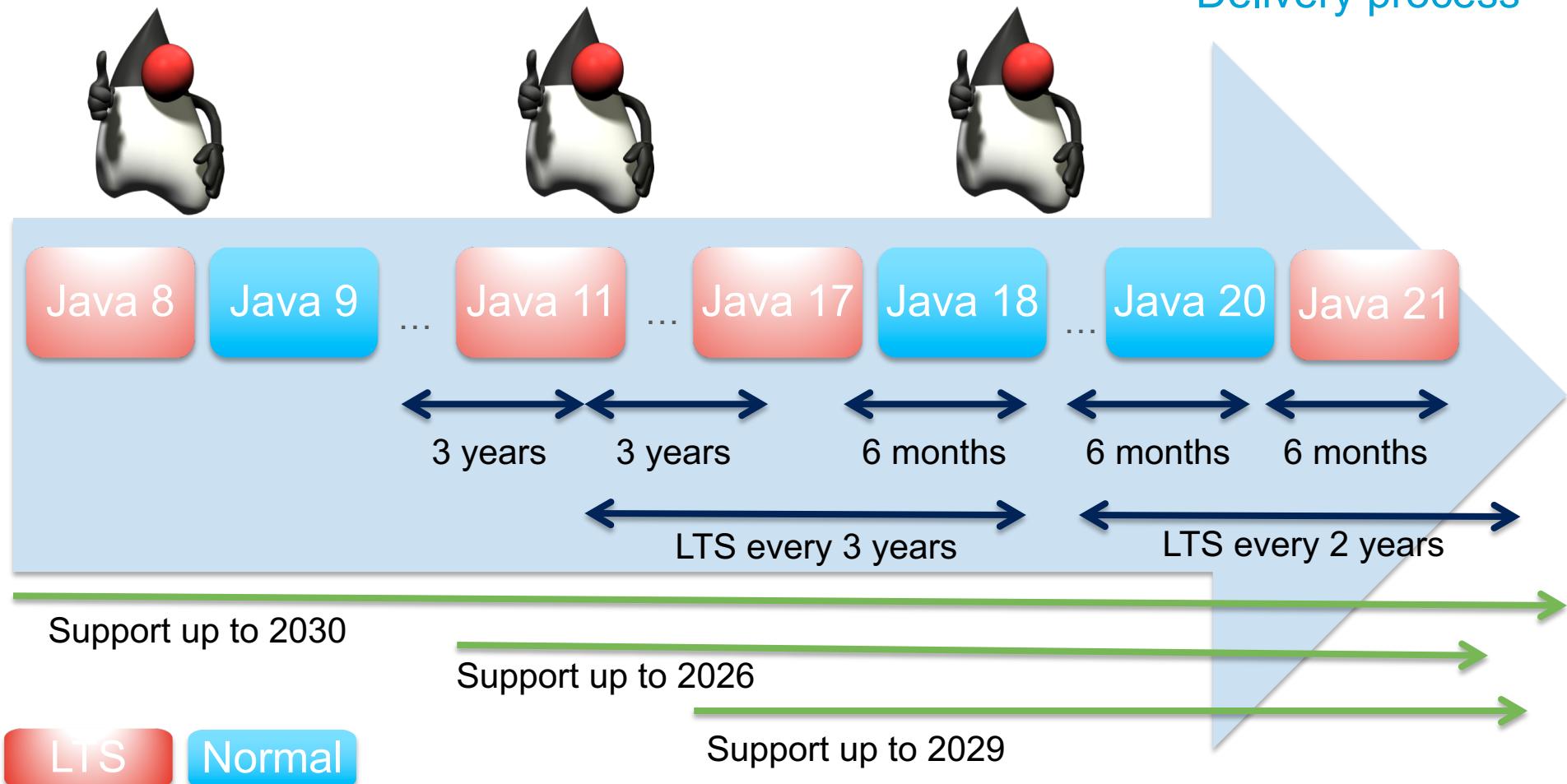


Questions



Java Delivery Process

Delivery process



- > Which build of Java are you currently using?
 - + Oracle JDK
 - + Oracle OpenJDK
 - + Eclipse Adoptium Temurin (formerly AdoptOpenJDK)
 - + Amazon Corretto
 - + IBM J9
 - + Azul Zing
 - + Azul Zulu
 - + Bellsoft Liberica JDK
 - + Alibaba Dragonwell
 - + RedHat
 - + Microsoft
 - + SAP
 - + Something else
 - + I don't know

- > **OracleJDK:**
 - + Until Java 17: Free for development, supported for 6 months
 - + Starting with Java 17: Free, but ask your lawyer
- > **Oracle OpenJDK:** Free forever, supported for 6 months
- > **Other OpenJDK:** Free forever, LTS supported after 6 months
 - + Built by **Adoptium**, Azul Zulu, IBM, Amazon, Bellsoft...
- > For more details see
 - + <https://medium.com/@javachampions/java-is-still-free-3-0-0-ocrt-2021-bca75c88d23b>

- > **JCP:** Java Community Process
 - + Responsible of Java governance
- > **JLS:** Java Language Specification
 - + This is a specification for the Java Language
- > **JSR:** Java Specification Request
 - + Big piece of Java change submitted to JCP
- > **JEP:** Java Enhancement Proposal
 - + Little pieces of Java change
- > <https://openjdk.java.net/projects/jdk/ii/>

- > **Coin**: Syntactic sugar in Java 7
- > **Valhalla**: Value types
- > **Loom**: New type of threads called Fiber
- > **Panama**: Improved IO
- > **Amber**: Developer productivity

<https://openjdk.java.net/projects/>

Amber:

- + JEP 361: Switch Expressions
- + JEP 389: Text Blocks
- + JEP 394: Pattern Matching for instanceof
- + JEP 395: Record
- + JEP 405: Record Patterns & Array Patterns (Preview)
- + JEP 406: Pattern matching for switch (Preview)
- + JEP 409: Sealed Classes

Valhalla:

- + JEP 374: Disable and Deprecate Biased Locking
- + JEP 390: Warnings for Value-Based Classes
- + JEP 401: Primitive Objects (Preview)
- + JEP 402: Unify the Basic Primitives with Objects (Preview)

Loom:

- + JEP 353: Reimplement the Legacy Socket API
- + JEP 373: Reimplement the Legacy DatagramSocket API

Trends

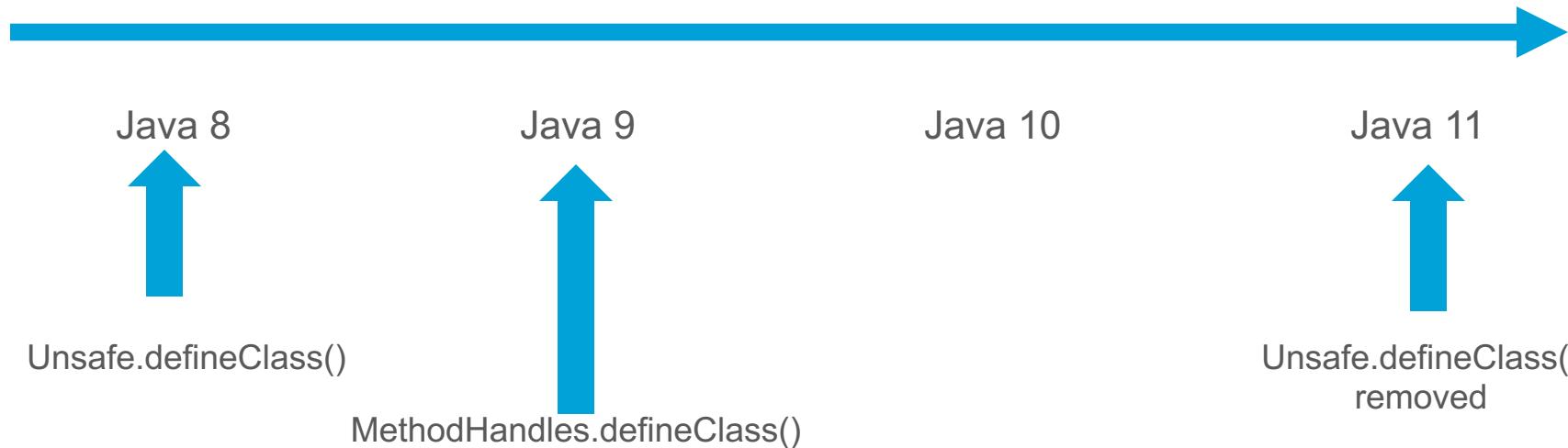
@Deprecated

is back

Stronger than ever

```
/**  
 * Counts the number of stack frames in this thread. The thread must  
 * be suspended.  
 *  
 * @return the number of stack frames in this thread.  
 * @throws IllegalThreadStateException if this thread is not  
 * suspended.  
 * @deprecated The definition of this call depends on {@link #suspend},  
 * which is deprecated. Further, the results of this call  
 * were never well-defined.  
 * This method is subject to removal in a future version of Java SE.  
 * @see StackWalker  
 */  
@Deprecated(since="1.2", forRemoval=true)  
public native int countStackFrames();
```

Classes and methods removed



Entire module removed

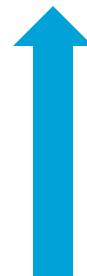
What any sane person will want his/her framework to support

Java 8



`import java.xml.bind`

Java 9



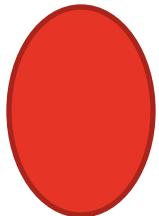
`--add-modules javax.xml.bind`

Java 10

Java 11



`--module-path containing
jaxb-api-2.4.0-xxx.jar`



Encapsulation

The party is over

Exception in thread "main" java.lang.reflect.InaccessibleObjectException: Unable to make field private final byte[] java.lang.String.value accessible: module java.base does not "opens java.lang" to unnamed module @2ff4acd0

Garbage Collector

Java < 8

- Serial
- Parallel
- ParallelOld
- CMS
- iCMS

Java 8

- G1

Java 9

- Removed iCMS
- G1 (Default GC)

Java 11

- Epsilon
- Z (Experimental)

Java 12

- Shenandoah (Experimental)

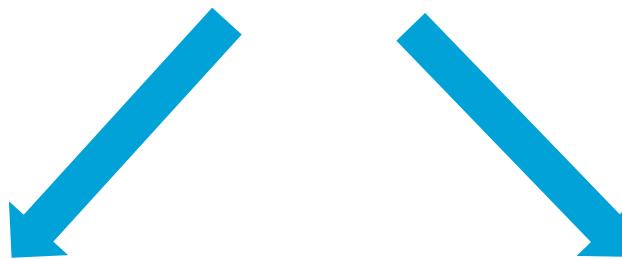
Java 14

- Remove CMS

Java 15

- Shenandoah
- Z

Developer efficiency



Inference

Pattern matching

```
// 71 characters
Map<String, String> map = new Map<String, String>();
map.put("a", "b");
// 27 characters
var map = Map.of("a", "b");
```



Performance

Deprecation

Garbage Collector

Encapsulation

Developer efficiency

Performance

Java 9

(September 2017)

Modules

- 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

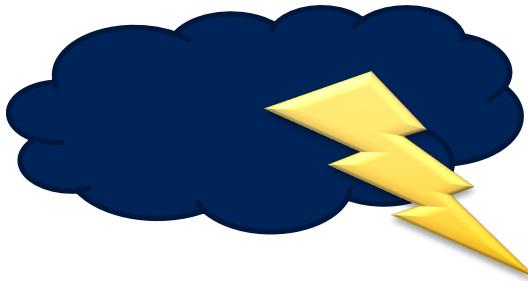
Deprecation

Garbage Collector

Encapsulation

Developer efficiency

Performance



```
Map<String, String> map = new HashMap<>() {{  
    put("key", "value");  
}};
```



```
Map<String, String> map =  
    Map.of("key", "value");
```



Slightly longer map instantiation

```
private Map<String, String> map = new HashMap<>();  
{  
    map.put("key", "value");  
}
```

```
ByteArrayOutputStream in = new ByteArrayOutputStream();
```

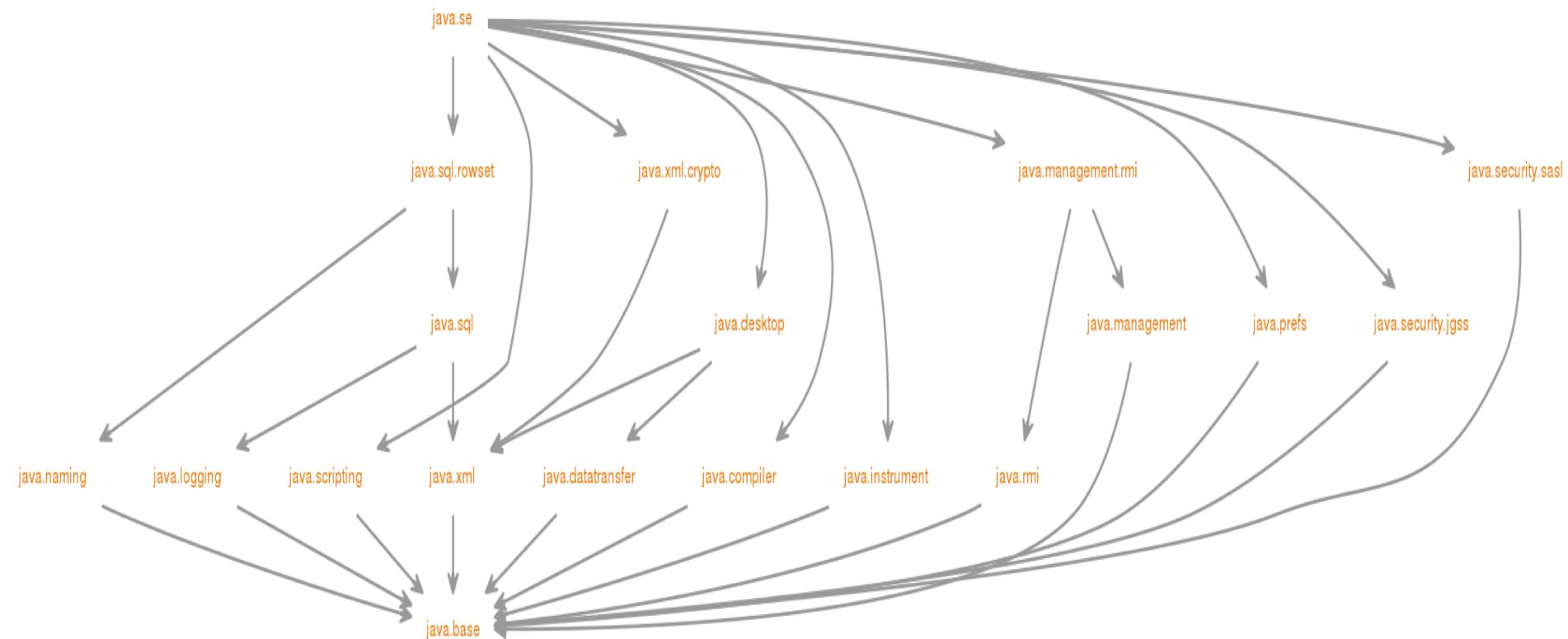
```
try(in) {  
}
```

Instead of

```
try(ByteArrayOutputStream other = in) {  
}
```

```
public interface Printer {  
    private void print(String s) {  
        System.out.println(s);  
    }  
  
    default void printAll(String[] list) {  
        Arrays.stream(list).forEach(this::print);  
    }  
}
```

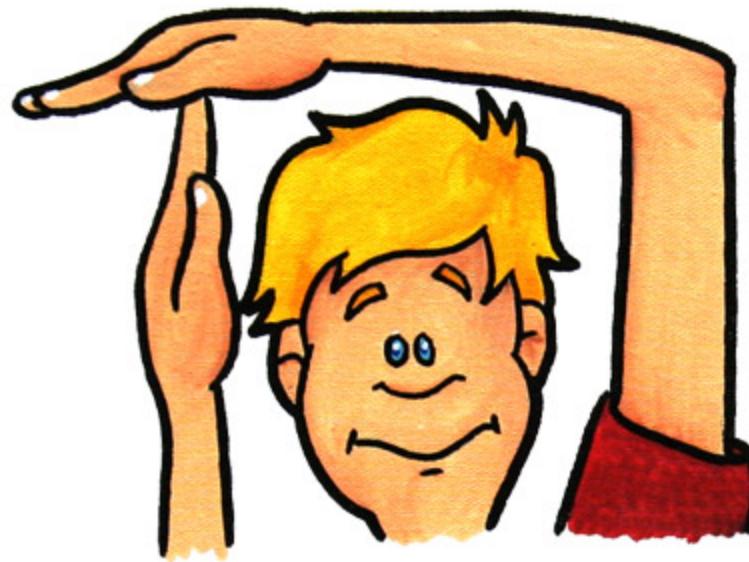
Java SE Modules



Credits: Oracle

DEMO

5 minutes break



Java 10

JSR 383 (March 2018)

JEP 286: Local-Variable Type Inference

JEP 296: Consolidate the JDK Forest into a Single Repository

JEP 304: Garbage-Collector Interface

JEP 307: Parallel Full GC for G1

JEP 310: Application Class-Data Sharing

JEP 312: Thread-Local Handshakes

JEP 313: Remove the Native-Header Generation Tool (javah)

JEP 314: Additional Unicode Language-Tag Extensions

JEP 316: Heap Allocation on Alternative Memory Devices

JEP 317: Experimental Java-Based JIT Compiler

JEP 319: Root Certificates

JEP 322: Time-Based Release Versioning

Deprecation
Garbage Collector
Encapsulation
Developer efficiency
Performance

DEMO

> **Keywords**

- + while, if, abstract, public, default, class, enum (Java 5), _ (Java 9)

> **Restricted Keywords**

- + open, module, to, with

> **Literals**

- + true, false, null

> **Reserved identifier**

- + var

DEMO

DEMO

Java 11

JSR 384 (September 2018)

JEP 181: Nest-Based Access Control

JEP 309: Dynamic Class-File Constants

JEP 315: Improve Aarch64 Intrinsics

JEP 318: Epsilon: A No-Op Garbage Collector

JEP 320: Remove the Java EE and CORBA Modules

JEP 321: HTTP Client (Standard)

JEP 323: Local-Variable Syntax for Lambda Parameters

JEP 324: Key Agreement with Curve25519 and Curve448

JEP 327: Unicode 10

JEP 328: Flight Recorder

JEP 329: ChaCha20 and Poly1305 Cryptographic Algorithms

JEP 330: Launch Single-File Source-Code Programs

JEP 331: Low-Overhead Heap Profiling

JEP 332: Transport Layer Security (TLS) 1.3

JEP 333: ZGC: A Scalable Low-Latency Garbage Collector (Experimental)

JEP 335: Deprecate the Nashorn JavaScript Engine

JEP 336: Deprecate the Pack200 Tools and API

Deprecation

Garbage Collector

Encapsulation

Developer efficiency

Performance

DEMO

DEMO

DEMO

Java 12

JSR 386 (March 2019)

JEP 189: Shenandoah: A Low-Pause-Time Garbage Collector (Experimental)

JEP 230: Microbenchmark Suite

JEP 325: Switch Expressions (Preview)

JEP 334: JVM Constants API

JEP 340: One AArch64 Port, Not Two

JEP 341: Default CDS Archives

JEP 344: Abortable Mixed Collections for G1

JEP 346: Promptly Return Unused Committed Memory from G1



```
private boolean isWeekDay(DayOfWeek day) {  
    boolean weekDay;  
  
    switch(day) {  
        case MONDAY:  
        case TUESDAY:  
        case WEDNESDAY:  
        case THURSDAY:  
        case FRIDAY:  
            weekDay = true;  
            break;  
        case SATURDAY:  
        case SUNDAY:  
            weekDay = false;  
        default:  
            throw new IllegalStateException("A new day was added in my week: " + day);  
    }  
    return weekDay;  
}
```

```
private boolean isWeekDay(DayOfWeek day) {  
    boolean weekDay;  
  
    switch(day) {  
        case MONDAY:  
        case TUESDAY:  
        case WEDNESDAY:  
        case THURSDAY:  
        case FRIDAY:  
            weekDay = true;  
            break;  
        case SATURDAY:  
        case SUNDAY:  
            weekDay = false;  
            break;  
        default:  
            throw new IllegalStateException("A new day was added in my week: " + day);  
    }  
    return weekDay;  
}
```

```
private boolean isWeekDay(DayOfWeek day) {  
    boolean weekDay;  
  
    switch(day) {  
        case MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY -> weekDay = true;  
        case SATURDAY, SUNDAY -> weekDay = false;  
        default -> throw new IllegalStateException("A new day was added in my week: " + day);  
    }  
  
    return weekDay;  
}
```

```
private boolean isWeekDay(DayOfWeek day) {  
    boolean weekDay = switch(day) {  
        case MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY -> true;  
        case SATURDAY, SUNDAY -> false;  
        default -> throw new IllegalStateException("A new day was added in my week: " + day);  
    };  
  
    return weekDay;  
}
```

Switch expressions

```
private boolean isWeekDay(DayOfWeek day) {  
    return switch(day) {  
        case MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY -> true;  
        case SATURDAY, SUNDAY -> false;  
        default -> throw new IllegalStateException("A new day was added in my week: " + day);  
    };  
}
```

Java 13

JSR 388 (September 2019)

- JEP 350: Dynamic CDS Archives
- JEP 351: ZGC: Uncommit Unused Memory
- JEP 353: Reimplement the Legacy Socket API
- JEP 354: Switch Expressions (Preview)
- JEP 355: Text Blocks (Preview)



Switch expressions (still a preview)

```
private boolean isWeekDay(DayOfWeek day) {  
    return switch(day) {  
        case MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY -> true;  
        case SATURDAY, SUNDAY -> false;  
        default -> {  
            if(onEarth) {  
                yield false;  
            }  
            yield true;  
        }  
    };  
}
```

Switch expressions, two syntaxes

```
private boolean isWeekDay(DayOfWeek day) {  
    return switch(day) {  
        case MONDAY:  
        case TUESDAY:  
        case WEDNESDAY:  
        case THURSDAY:  
        case FRIDAY:  
            yield true;  
        case SATURDAY:  
        case SUNDAY:  
            yield false;  
    };  
}
```

```
String script = """
```

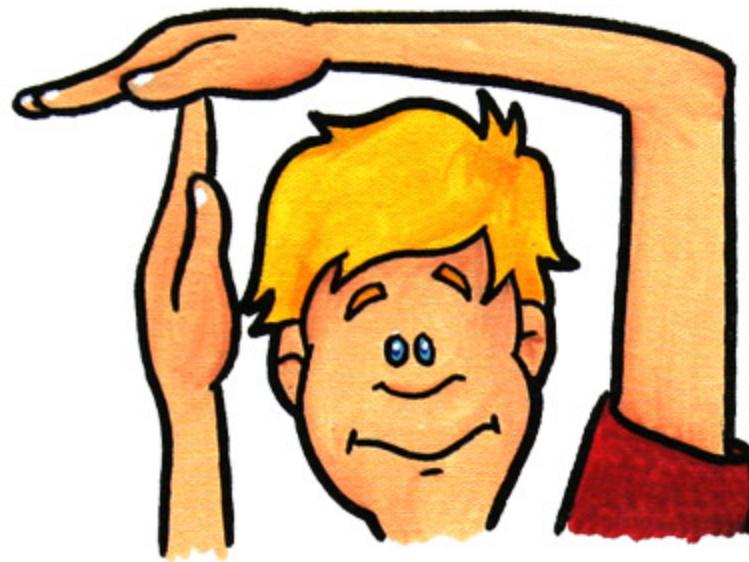
```
function hello() {  
    print("Hello, world\\");  
}
```

```
hello();  
""";  
;
```

Text Block 16

DEMO

5 minutes break



Java 14

JSR 389 (March 2020)

- JEP 305: Pattern Matching for instanceof (Preview)
- JEP 343: Packaging Tool (Incubator)
- JEP 345: NUMA-Aware Memory Allocation for G1
- JEP 349: JFR Event Streaming
- JEP 352: Non-Volatile Mapped Byte Buffers
- JEP 358: Helpful NullPointerExceptions
- JEP 359: Records (Preview)
- JEP 361: Switch Expressions (Standard)
- JEP 362: Deprecate the Solaris and SPARC Ports
- JEP 363: Remove the Concurrent Mark Sweep (CMS) Garbage Collector
- JEP 364: ZGC on macOS
- JEP 365: ZGC on Windows
- JEP 366: Deprecate the ParallelScavenge + SerialOld GC Combination
- JEP 367: Remove the Pack200 Tools and API
- JEP 368: Text Blocks (Second Preview)
- JEP 370: Foreign-Memory Access API (Incubator)

Deprecation
Garbage Collector
Encapsulation
Developer efficiency
Performance

DEMO

DEMO

DEMO

DEMO

Java 15

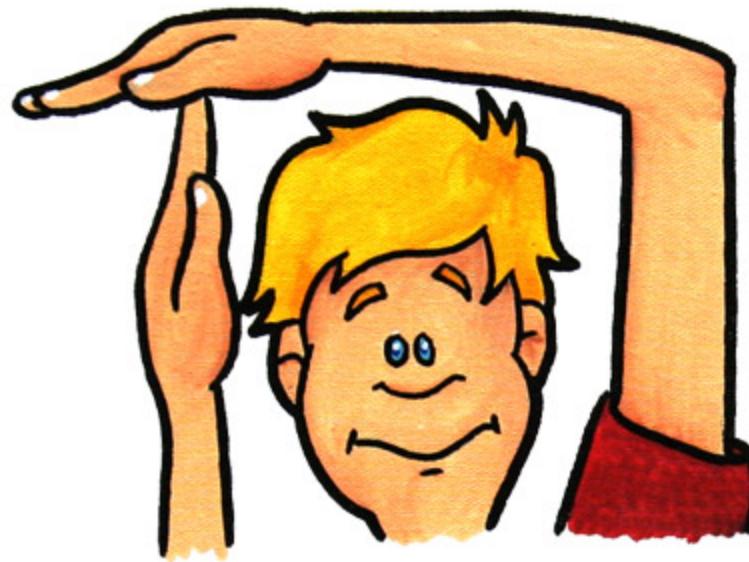
JSR 390 (September 2020)

- JEP 339: Edwards-Curve Digital Signature Algorithm (EdDSA)
- JEP 360: Sealed Classes (Preview)
- JEP 371: Hidden Classes
- JEP 372: Remove the Nashorn JavaScript Engine
- JEP 373: Reimplement the Legacy DatagramSocket API
- JEP 374: Disable and Deprecate Biased Locking
- JEP 375: Pattern Matching for instanceof (Second Preview)
- JEP 377: ZGC: A Scalable Low-Latency Garbage Collector
- JEP 378: Text Blocks
- JEP 379: Shenandoah: A Low-Pause-Time Garbage Collector
- JEP 381: Remove the Solaris and SPARC Ports
- JEP 383: Foreign-Memory Access API (Second Incubator)
- JEP 384: Records (Second Preview)
- JEP 385: Deprecate RMI Activation for Removal

Deprecation
Garbage Collector
Encapsulation
Developer efficiency
Performance

DEMO

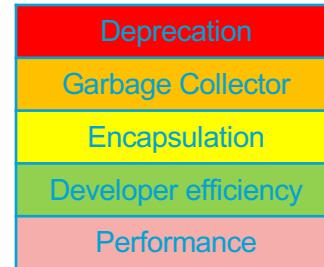
5 minutes break



Java 16

JSR 391 (March 2021)

- JEP 338: Vector API (Incubator)
- JEP 347: Enable C++14 Language Features
- JEP 357: Migrate from Mercurial to Git
- JEP 369: Migrate to GitHub**
- JEP 376: ZGC: Concurrent Thread-Stack Processing
- JEP 380: Unix-Domain Socket Channels
- JEP 386: Alpine Linux Port
- JEP 387: Elastic Metaspace
- JEP 388: Windows/AArch64 Port
- JEP 389: Foreign Linker API (Incubator)
- JEP 390: Warnings for Value-Based Classes**
- JEP 392: Packaging Tool**
- JEP 393: Foreign-Memory Access API (Third Incubator)
- JEP 394: Pattern Matching for instanceof
- JEP 395: Records
- JEP 396: Strongly Encapsulate JDK Internals by Default**
- JEP 397: Sealed Classes (Second Preview)**



Strongly encapsulate JDK internals by default

--illegal-access=permit → Open to unnamed modules. 1 warning on the console

--illegal-access=warn → Same. But tons of warnings

--illegal-access=debug → Same, but stack trace of the warnings

--illegal-access=deny → Forbidden unless --add-opens is used

This flag is now deprecated.

DEMO

DEMO

Java 17

JSR 392 (September 2021)

- JEP 306: Restore Always-Strict Floating-Point Semantics
- JEP 356: Enhanced Pseudo-Random Number Generators**
- JEP 382: New macOS Rendering Pipeline
- JEP 391: macOS/AArch64 Port
- JEP 398: Deprecate the Applet API for Removal
- JEP 403: Strongly Encapsulate JDK Internals**
- JEP 406: Pattern Matching for switch (Preview)**
- JEP 407: Remove RMI Activation
- JEP 409: Sealed Classes
- JEP 410: Remove the Experimental AOT and JIT Compiler**
- JEP 411: Deprecate the Security Manager for Removal**
- JEP 412: Foreign Function & Memory API (Incubator)
- JEP 414: Vector API (Second Incubator)**
- JEP 415: Context-Specific Deserialization Filters**

Deprecation
Garbage Collector
Encapsulation
Developer efficiency
Performance

DEMO

DEMO

DEMO

Conclusion

Who has learned
something today?



- > Java Champions – Java is still free
 - + <https://medium.com/@javachampions/java-is-still-free-3-0-0-oct-2021-bca75c88d23b>
- > Inside Java
 - + <https://inside.java/>
- > Dev Java
 - + <https://dev.java/>
- > Zeroturnaround Module Cheat Sheet
 - + <http://files.zeroturnaround.com/pdf/RebelLabs-Java-9-modules-cheat-sheet.pdf>
- > Var styleguide
 - + <http://openjdk.java.net/projects/amber/LVTIstyle.html>
- > Source code
 - + <https://github.com/henri-tremblay/java91011beyond>
- > JDK
 - + <https://openjdk.java.net/projects/jdk/>
- > Sdkman
 - + <https://sdkman.io/>

- > Courses by
 - + Cay S. Horstmann
 - + Heinz M. Kabutz
 - + Kirk C. Pepperdine



<http://objenesis.org>



<http://montreal-jug.org>

Questions?



Henri Tremblay
henri@tremblay.pro
<http://blog.tremblay.pro>
[@henri_tremblay](https://twitter.com/henri_tremblay)

EASYMOCK

<http://easymock.org>