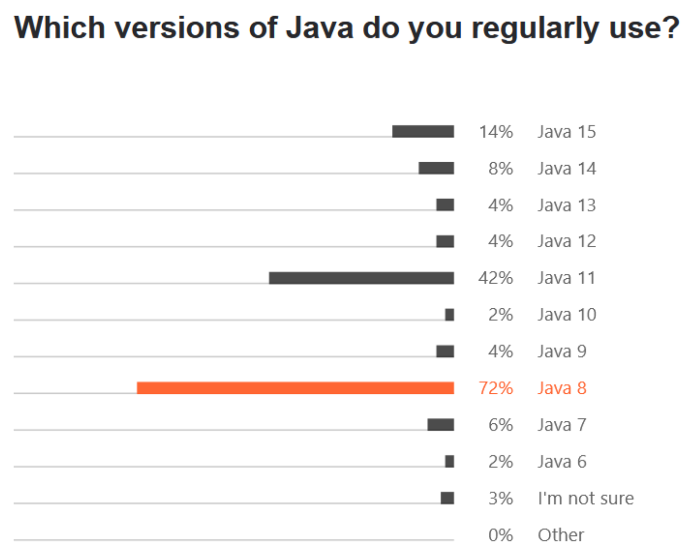
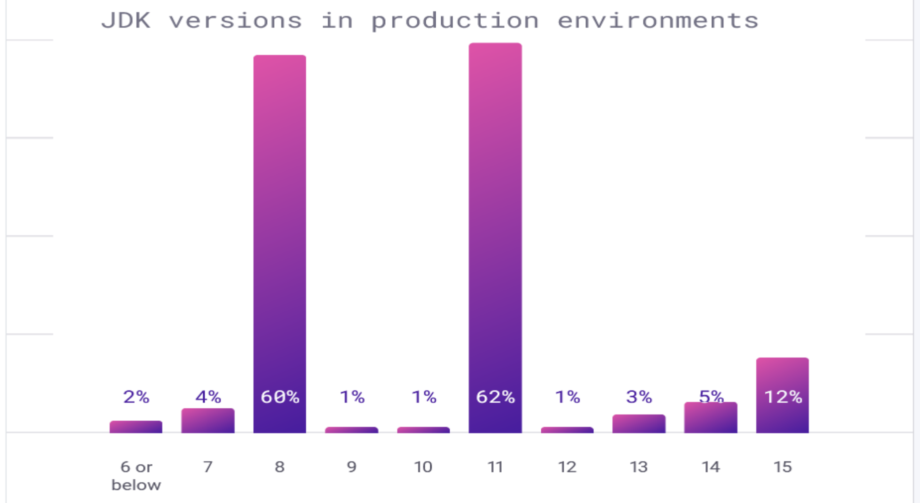
LIFE AFTER JAVA 8

POPESCU IONUT COSMIN – ERICSSON ROMANIA

JAVA 8 RECAP I

- Almost 10 years
- Still strong

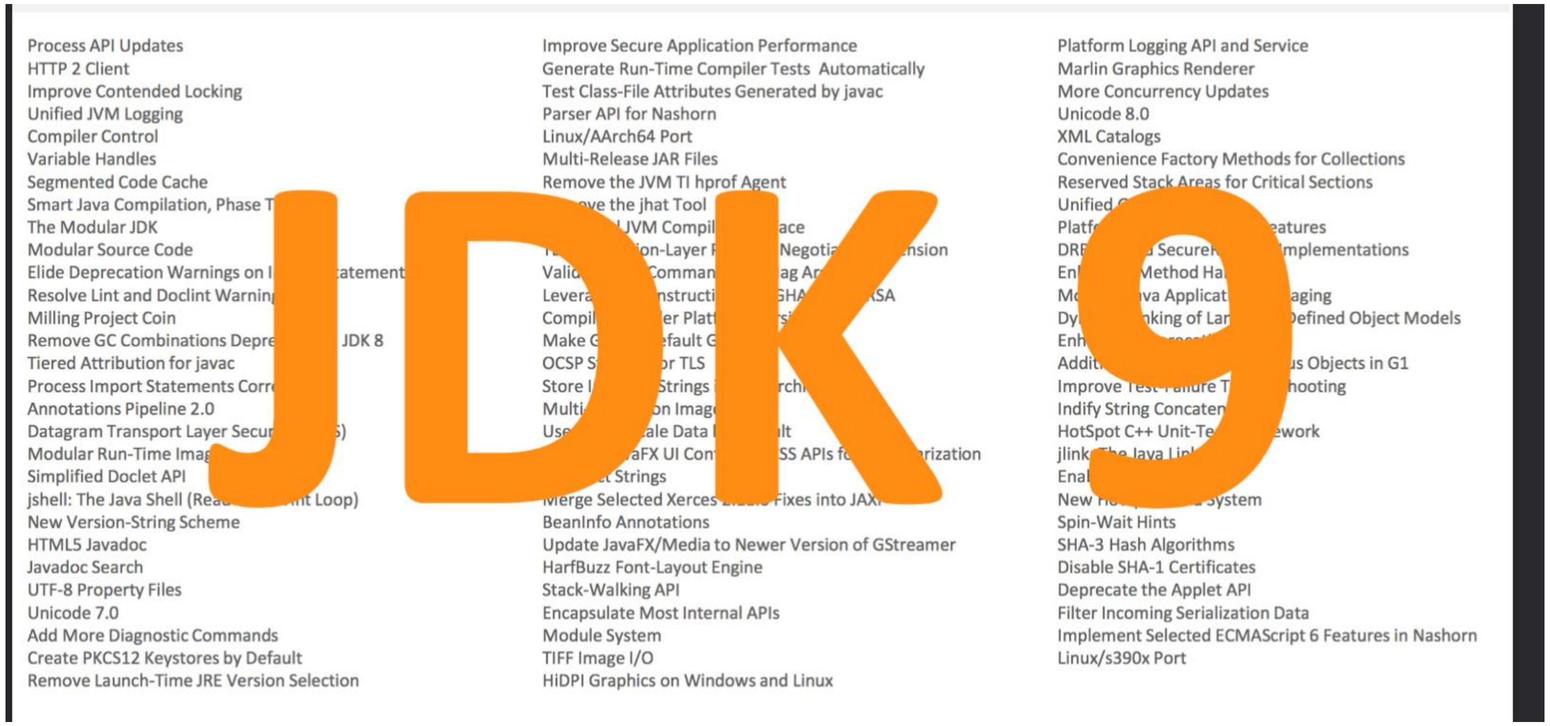




JAVA 8 RECAP II

- Version that modernized the language
- Streams API (map, filter, flatMap, collect, anyMatch)
- Lambda expressions (C# had them since 3.0)
- New API for date and time ()
- Default methods in interfaces (C# will have it 5years later ☺)

JDK 9



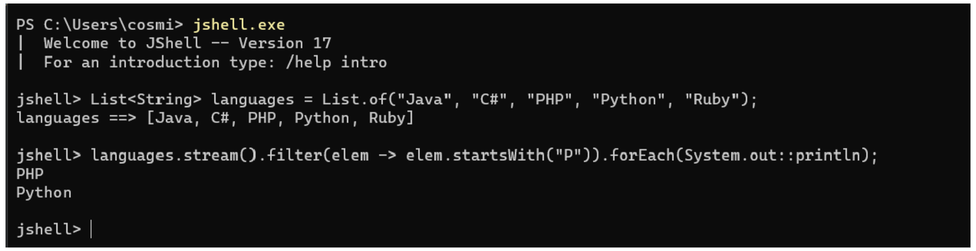
Process API Updates
HTTP 2 Client
Improve Contended Locking
Unified JVM Logging
Compiler Control
Variable Handles
Segmented Code Cache
Smart Java Compilation, Phase T
The Modular JDK
Modular Source Code
Elide Deprecation Warnings on I tatement
Resolve Lint and Doclint Warnin
Milling Project Coin
Remove GC Combinations Depre JDK 8
Tiered Attribution for javac
Process Import Statements Corr
Annotations Pipeline 2.0
Datagram Transport Layer Secur S)
Modular Run-Time Imag
Simplified Doclet API
jshell: The Java Shell (Rea nt Loop)
New Version-String Scheme
HTML5 Javadoc
Javadoc Search
UTF-8 Property Files
Unicode 7.0
Add More Diagnostic Commands
Create PKCS12 Keystores by Default
Remove Launch-Time JRE Version Selection

Improve Secure Application Performance
Generate Run-Time Compiler Tests Automatically
Test Class-File Attributes Generated by javac
Parser API for Nashorn
Linux/AArch64 Port
Multi-Release JAR Files
Remove the JVM TI hprof Agent
ve the jhat Tool
JVM Compil ace
ion-Layer Negoti nsion
Vali Comman ag Ar
Levera nstructi SHA SA
Compil er Platt
Make G efault G
OCSP S or TLS
Store I Strings rch
Multi on Imag
Use ale Data lt
aFX UI Con SS APIs f rization
Strings
Merge Selected Xerces Fixes into JAX
BeanInfo Annotations
Update JavaFX/Media to Newer Version of GStreamer
HarfBuzz Font-Layout Engine
Stack-Walking API
Encapsulate Most Internal APIs
Module System
TIFF Image I/O
HiDPI Graphics on Windows and Linux

Platform Logging API and Service
Marlin Graphics Renderer
More Concurrency Updates
Unicode 8.0
XML Catalogs
Convenience Factory Methods for Collections
Reserved Stack Areas for Critical Sections
Unified
Platf eatures
DRE SecureR Implementations
En Method Ha
Mo va Applicat aging
Dy nking of Lar Defined Object Models
Enh
Addit us Objects in G1
Improve T hooting
Indify String Concaten
HotSpot C++ Unit-Te ework
jlink
Enal
New System
Spin-Wait Hints
SHA-3 Hash Algorithms
Disable SHA-1 Certificates
Deprecate the Applet API
Filter Incoming Serialization Data
Implement Selected ECMAScript 6 Features in Nashorn
Linux/s390x Port

JDK 9 - I

- Big And Small Changes
- Java platform module system(some devs hate them.)
- jlink – generate custom java runtime image with the necessary modules and dependencies to run the application.
- Jshell – REPL shell for running Java code



```
PS C:\Users\cosmi> jshell.exe
|  Welcome to JShell -- Version 17
|  For an introduction type: /help intro

jshell> List<String> languages = List.of("Java", "C#", "PHP", "Python", "Ruby");
languages ==> [Java, C#, PHP, Python, Ruby]

jshell> languages.stream().filter(elem -> elem.startsWith("P")).forEach(System.out::println);
PHP
Python

jshell>
```

JDK 9 - II

- HTTP/2 Client (Incubator) – HTTP2 support, WebSockets

- Support for Unicode 7 and 8
- Convenience Factory Methods for Collections

```
Set<Integer> mySet = Set.of(1, 2, 3);
List<Integer> myList = List.of(1, 2, 3);
Map<String, Integer> myMap = Map.of("one", 1, "two", 2);
```

JDK 9 - III

- New methods in Streams API (dropWhile, takeWhile, iterate, ofNullable)
- Reactive Streams publish-subscribe framework – interface level. Implementations are provided by RxJava for example
- Arrays.mismatch ☺
- The great cleanup starts (deprecate Applet API ☺, Concurrent Mark Sweep Garbage Collector, Object.finalize()). This work will be carried on by the future JDKs.
- Encapsulate Most Internal APIs (`sun.misc.Unsafe`)

JDK 10 - I

- March 2018
- Something amazing is happening ☺
- Java gets var (C# has it since 2007 ☺)

```
var userList = new ArrayList<String>(); // infers ArrayList<String>
var stream = list.stream(); // infers Stream<String>
```

Verbosity will gradually start to reduce.

JDK 10 - II

- Application Class-Data Sharing – improving startup footprint
- New APIs (`List`, `Set`, `Map.copyOf(Collection)`, `Optional.orElseThrow()`, `Collectors.toUnmodifiable/List/Map/Set`)
- New Java release cycle. Two versions per years. Usually, spring and autumn.

JDK 11 - I

- GA 09/2018
- Long live the new LTS after Java 8
- Java 11(Oracle JDK) trap (https://blog.joda.org/2018/09/do-not-fall-into-oracles-java-11-trap.html)
- Our old friend, HTTP client from JDK 9 is standard now. Production ready
- Local-Variable Syntax for Lambda Parameters (`(var s1, var s2) -> s1 + s2`). We can use also type annotations

JDK 11 - II

- Launch Single-File Source-Code Programs – launch a java program like a script, like python or php script
- chmod +x MyProgram.java
- ./MyProgram.java 1 2 3

```
#!/usr/local/bin/java --source 11

import java.util.Arrays;

public class Addition
{
    public static void main(String[] args) {
        Integer sum = Arrays.stream(args)
          .mapToInt(Integer::parseInt)
          .sum();

        System.out.println(sum);
    }
}
```

JDK 11 - III

- Unicode 10
- Deprecate the Nashorn JavaScript Engine
- New cryptographic algorithms (ChaCha20 and Poly1305)
- TLS 1.3
- ZGC: Experimental low-latency garbage collector and Epsilon garbage collector
- Flight Recorder
- Remove the Java EE and CORBA Modules – careful with the update. No more JAXB

JDK 11 - IV

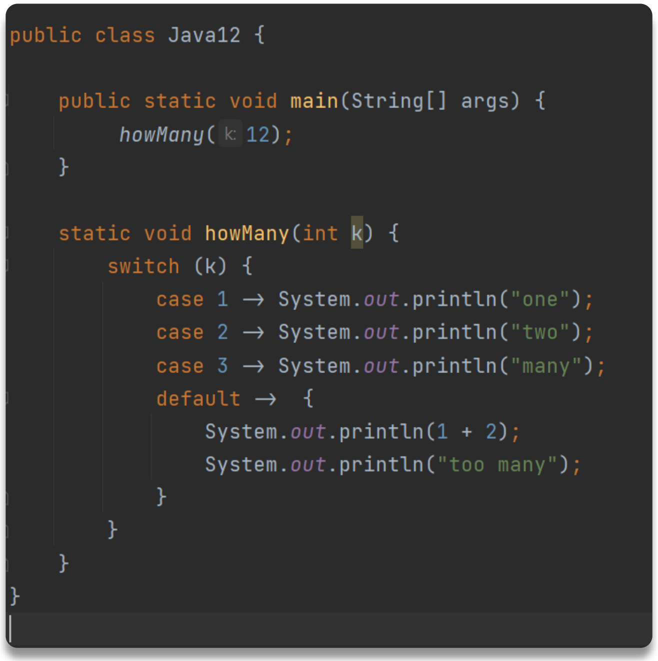
- New APIs
  - `Optional.isEmpty()`
  - `String.isBlank`
  - `String.repeat(int)`
  - `String strip()`
  - `String stripLeading()`
  - `String stripTrailing()`
  - `Predicate not(Predicate`

JDK 12 - I

- 03/2019
- Shenandoah: A Low-Pause-Time Garbage Collector (Experimental)
- Switch Expressions (Preview)

  Not included in the Java SE standard
  Preview features require use of —enable–preview
flag

  In JDK 13, it will be in the second preview



```
public class Java12 {

    public static void main(String[] args) {
        howMany( k: 12);
    }

    static void howMany(int k) {
        switch (k) {
            case 1 -> System.out.println("one");
            case 2 -> System.out.println("two");
            case 3 -> System.out.println("many");
            default -> {
                System.out.println(1 + 2);
                System.out.println("too many");
            }
        }
    }
}
```

JDK 12 - II

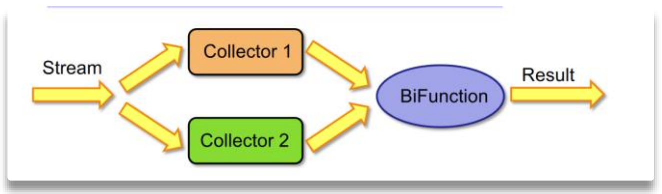
- Stream API gets a new collector – teeing (https://blog.frankel.ch/teeing-java-api/)

  `teeing(Collector, Collector, BiFunction)`

  Collect a stream using two collectors

  Use a BiFunction to merge the two collections

  tee is a Linux command ☺



JDK 13 - I

- 09/2019
- Only 5 JEPs ☺
- Switch expressions (Second preview)
- Text blocks (preview)

```
String html = """
          <html>
              <body>
                  <p>Hello, world</p>
              </body>
          </html>
          """;
```

JDK 13 - II

- Reimplement the Legacy Socket API – easy to adopt to work with Project Loom

JDK 14 - I

- 03/2020 ☺
- Pattern Matching for instanceof (Preview) – and pattern matching is in Scala, C#, Haskell. It allows the desired 'shape' of an object to be expressed concisely (the *pattern*), and for various statements and expressions to test that 'shape' against their input (the *matching*).

```
if (obj instanceof String s) {
    // can use s here
} else {
    // can't use s here
}
```

JDK 14 - II

- Helpful NullPointerExceptions - improve the usability of NullPointerExceptions generated by the JVM by describing precisely which variable was null.
- Records – Java will have it first. In Kotlin – data class.
  - Compact syntax for declaring classes which are transparent holders for shallowly immutable data.

```
record Point(int x, int y) {} // the compiler takes care of
everything
```

JDK 14 - III

- Text Blocks (Second Preview) – two new escape sequences
- Foreign-Memory Access API (Incubator) - allow Java programs to safely and efficiently access foreign memory outside of the Java heap.
- Packaging Tool (Incubator) – package the applications as a msi or exe, deb, rpm
- Deprecations
  - Solaris and SPARC Ports
  - Pack200 Tools and API
  - ParallelScavenge + SerialOld GC Combination

JDK 15 - I

- 09/2020
- Sealed Classes (Preview) - Enhance the Java programming language with *sealed classes and interfaces*. Sealed classes and interfaces restrict which other classes or interfaces may extend or implement them.

```
package com.example.geometry;

public abstract sealed class Shape

    permits Circle, Rectangle, Square {...}
```

JDK 15 - II

- Pattern Matching for instanceof (Second Preview)
- Shenandoah: A Low-Pause-Time Garbage Collector (Production) – from experimental to profuction
- Records (Second Preview) – local records, annotations on records
- Foreign-Memory Access API (Second Incubator) – refinements
- ZGC (Production)
- Remove the Nashorn JavaScript Engine
- Reimplement the Legacy DatagramSocket API

JDK 16 - I

- 03/2021
- Records and pattern matching for instanceof are final in this JDK
- Sealed classes (second preview)
- Vector API (Incubator) – vector computations to use hardware instructions on supported CPU architectures
- Ports for Alpine Linux and Windows/AArch64
- Unix-Domain Socket Channels (https://www.morling.dev/blog/talking-to-postgres-through-java-16-unix-domain-socket-channels/)

JDK 17 I

- 09/2021
- Long live again the new LTS
- Oracle JDK – free ☺
- Sealed Classes – delivered
- Pattern Matching for switch (Preview) - https://openjdk.java.net/jeps/406
- Strongly Encapsulate JDK Internals – developers should use standard apis for increased security and maintainability.
- Sealed Classes – no changes from JDK 16
- macOS/AArch64 Port – Port for the new Apple M1

JDK 17 - II

- Vector API (Second Incubator)
- Foreign Function & Memory API (Incubator)
- Deprecate the Security Manager for Removal
- Deprecate the Applet API for Removal – will be removed for sure

Project to follow in Java world

- Besides the new shiny JEPs there are many projects in OpenJDK world for further improving the language
- Project Loom
- Project Valhalla
- GraalVM

Project Loom

- Easy-to-use, high-throughput lightweight concurrency and new programming models on the Java platform
- Virtual threads (Go has goroutines) – threads managed by JVM
- Delimited continuations
- Tail-call elimination
- There are builds - http://jdk.java.net/loom/ - based on incomplete versions of JDKs
- A nice presentation about green threads

Project Valhalla - I

- Inline types - *Codes like a class, works like an int!*
- Immutable and not nullable, have a default value

```
inline public class Point {
public int x;
public int y;

public Point(int x, int y) {
    this.x = x;
    this.y = y;
}
```

Project Valhalla - II

- Generics over Primitive Types – `List<Integer>` vs `List<int>`
- No one knows when will be finished

Project Panama

- Interconnecting JVM and native code
- The Foreign-Memory Access API
- The Foreign Linker API
- The Vector API
- https://github.com/carldea/panama4newbies - explore the project

Java for cloud, microservices

- Yeah, Java can do that ☺
- Spring Boot
- Quarkus, Micronaut
- Docker, K8S

Conclusion

- Is Java a bad language – NO
- Is Java sometimes too verbose – mmmmmyeah ☺
- Is <insert_your_favourite_language> better than Java – Some ideas can be better executed. ☺

THANK YOU ☺