ORACLE

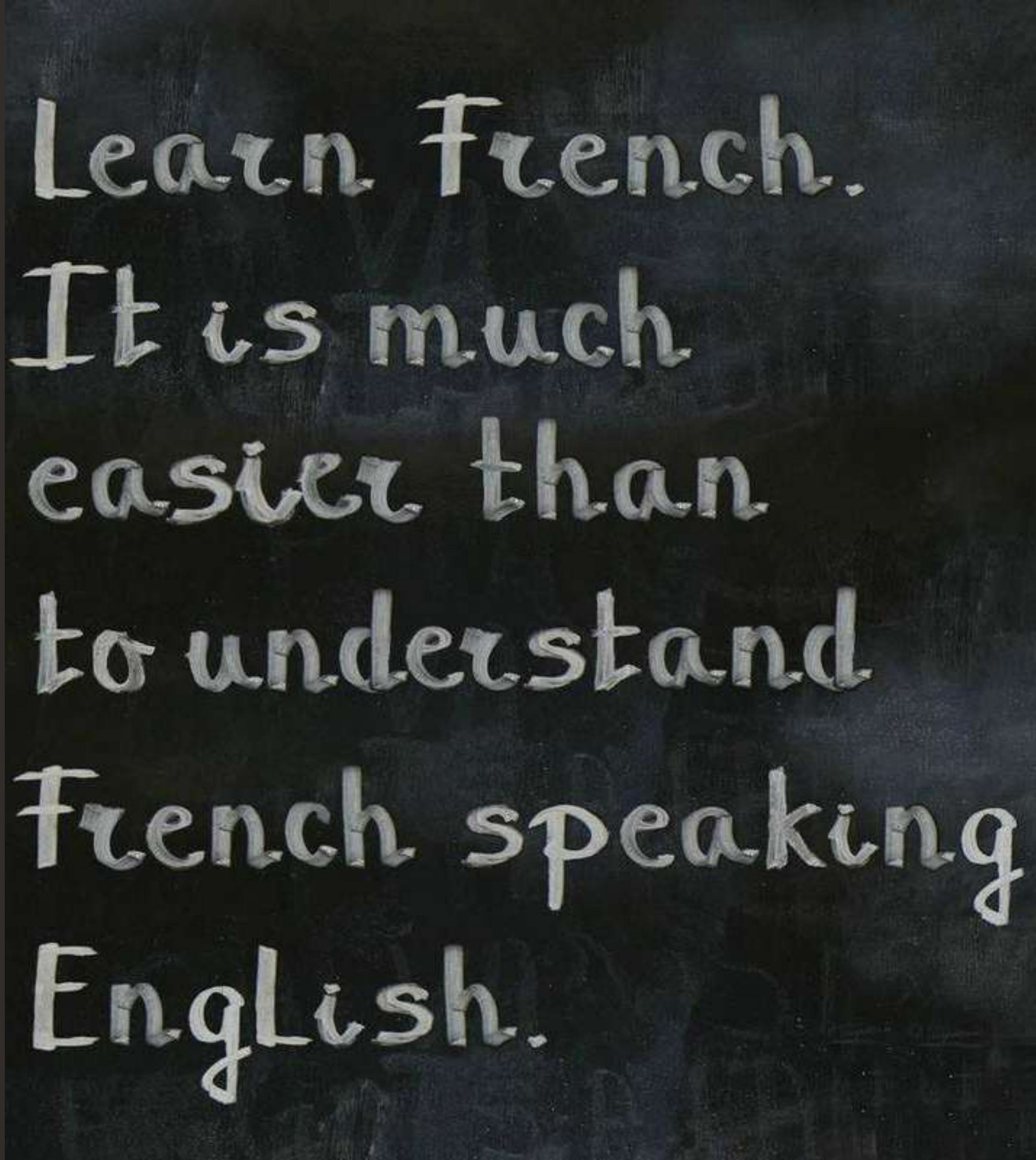
Java and the forty versions

David Delabassée
@delabassee
DevRel Java Platform Group

ConFoo.CA

February 2020

David Delabassée
@delabassee
Java Platform Group



Learn French.
It is much
easier than
to understand
French speaking
English.

ORACLE

fourteen
Java and the ~~forty~~ versions

David Delabassée
@delabassee
DevRel Java Platform Group

ConFoo.CA

February 2020

Safe harbor statement

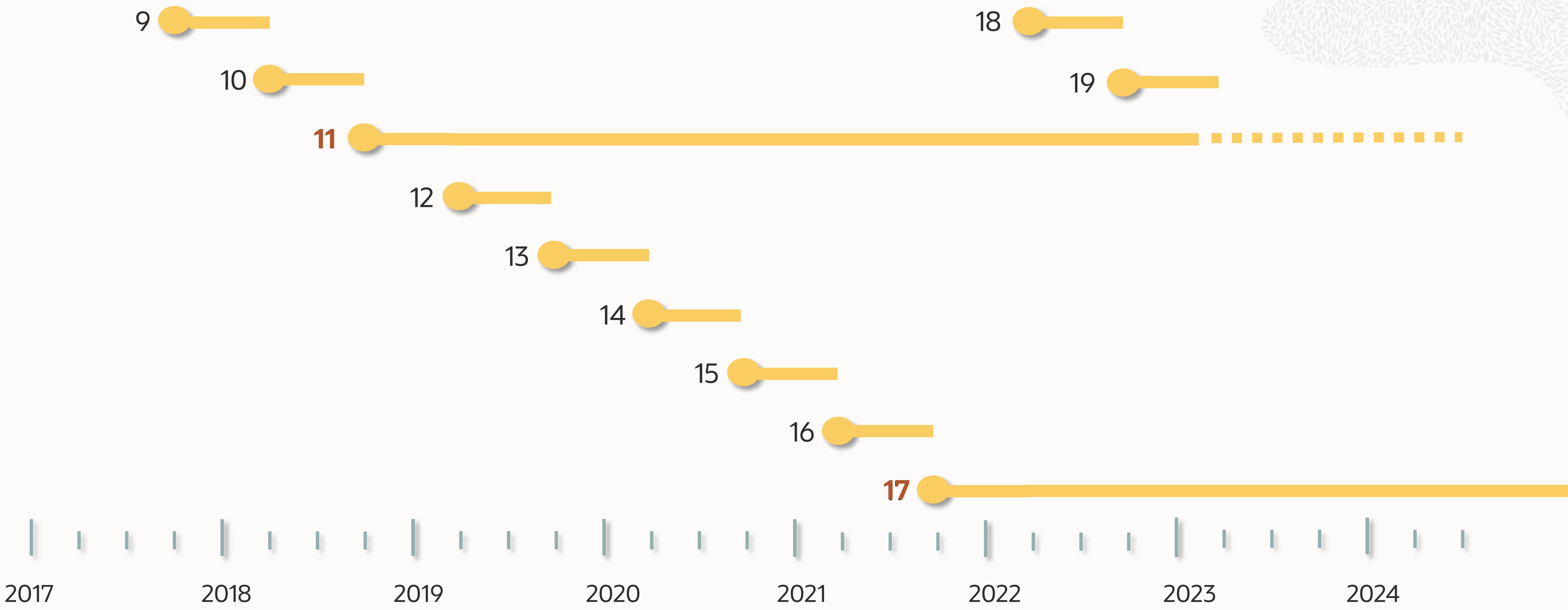
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Developer productivity
Application performance

In the face of constantly-evolving
programming paradigms,
application styles,
deployment styles
and hardware.

Java 9 / JDK 9 - 90 JEPs

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 Stats.
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	jshellThe 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 ALPN 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 Modul.
254	Compact Strings
255	Merge Selected Xerces 2.11.0 Updates into JAXP
256	BeanInfo Annotations
257	Update JavaFX/Media to Newer Ver 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 Link of Language-Def. Object Models
277	Enhanced Deprecation
278	Additional Tests for Humongous Objs in G1
279	Improve Test-Failure Troubleshooting
280	Indify String Concatenation
281	HotSpot C++ Unit-Test Framework
282	jlinkThe 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 GC
292	Implement Selected ES6 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

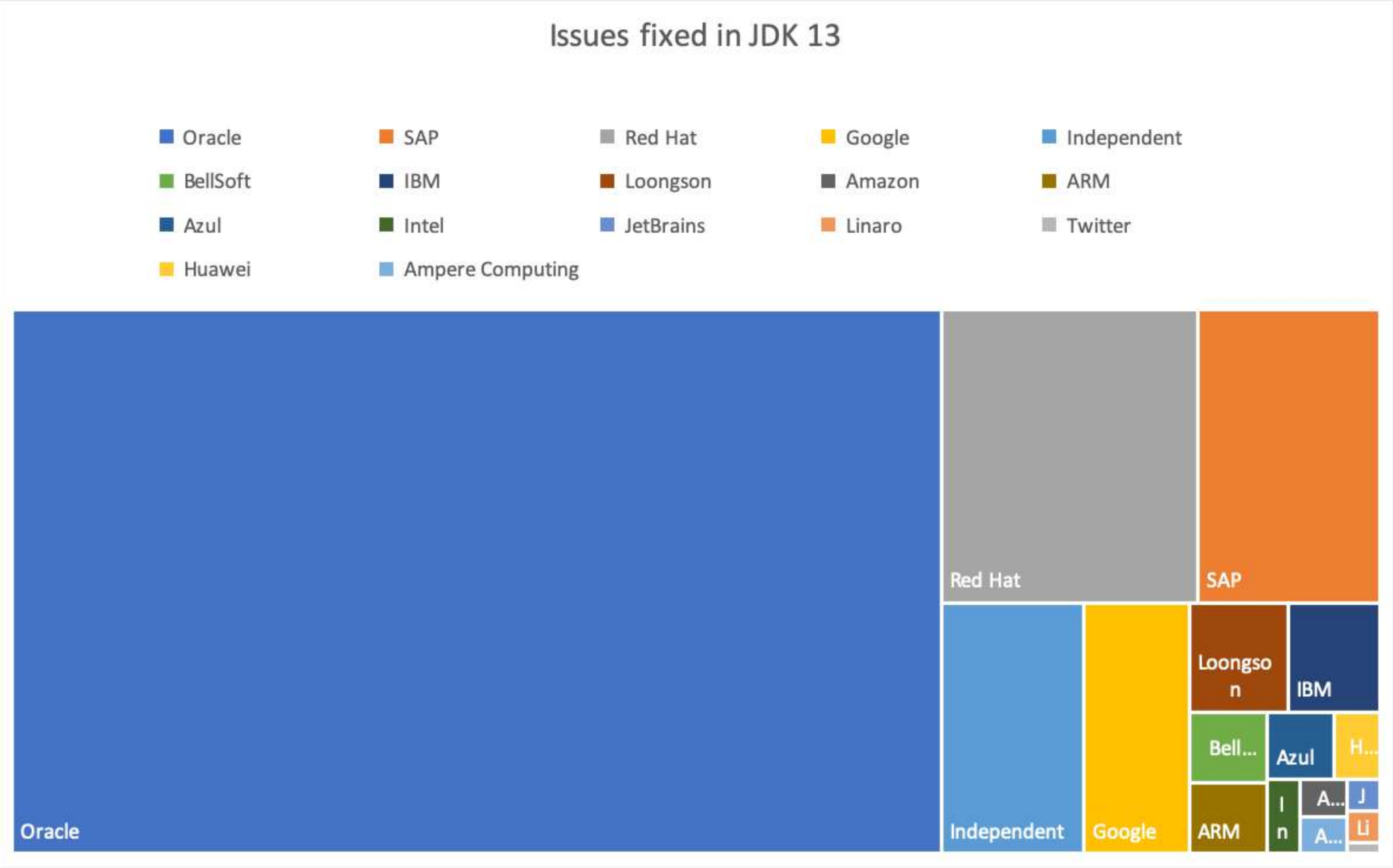

9
18
10
19
11
12
13
14
15
16
17
2017
2018
2019
2020
2021
2022
2023
2024

Oracle offers users choice

https://oracle.com/java

https://jdk.java.net

Oracle


Issues fixed in JDK 13
Oracle
SAP
Red Hat
Google
Independent
BellSoft
IBM
Loongson
Amazon
ARM
Azul
Intel
JetBrains
Linaro
Twitter
Huawei
Ampere Computing
Oracle
Red Hat
SAP
Independent
Google
Loongso
n
IBM
Bell...
Azul
H...
ARM
I
n
A...
J
A...
Li

Java is still free!

Foster Faster Innovations

- Predictable release cadence
- JEP
  - Incubator, Preview, Standard
  - https://openjdk.java.net/jeps
- Project Skara
  - Investigate alternative SCM and code review options for OpenJDK
  - Git, hosted Git provider & tooling
  - Already moved : Skara, jfx, jmc, Loom, Panama-foreign + R/O mirrors
  - https://github.com/openjdk/

Java is still free!
Delivering Faster

Java 10 / JDK 10 - 12 JEPs

286	Local-Variable Type Inference
296	Consolidate the JDK Forest into a Single Repository
304	Garbage-Collector Interface
307	Parallel Full GC for G1
310	Application Class-Data Sharing
312	Thread-Local Handshakes
313	Remove the Native-Header Generation Tool (javah)
314	Additional Unicode Language-Tag Extensions
316	Heap Allocation on Alternative Memory Devices
317	Experimental Java-Based JIT Compiler
319	Root Certificates
322	Time-Based Release Versioning

Java 11 / JDK 11 - 17 JEPs

181	Nest-Based Access Control
309	Dynamic Class-File Constants
315	Improve Aarch64 Intrinsics
318	Epsilon: A No-Op Garbage Collector (Experimental)
320	Remove the Java EE and CORBA Modules
321	HTTP Client (Standard)
323	Local-Variable Syntax for Lambda Parameters
324	Key Agreement with Curve25519 and Curve448
327	Unicode 10
328	Flight Recorder
329	ChaCha20 and Poly1305 Cryptographic Algorithms
330	Launch Single-File Source-Code Programs
331	Low-Overhead Heap Profiling
332	Transport Layer Security (TLS) 1.3
333	ZGC: A Scalable Low-Latency Garbage Collector (Experimental)
335	Deprecate the Nashorn JavaScript Engine
336	Deprecate the Pack200 Tools and API

Java 12 / JDK 12 - 8 JEPs

189	Shenandoah: A Low-Pause-Time Garbage Collector (Experimental)
230	Microbenchmark Suite
325	Switch Expressions (Preview)
334	JVM Constants API
340	One AArch64 Port, Not Two
341	Default CDS Archives
344	Abortable Mixed Collections for G1
346	Promptly Return Unused Committed Memory from G1

Java 13 / JDK 13 - 5 JEPs

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

Java 14 / JDK 14 - 16 JEPs

14

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

20/2/2020 Final Release Candidate
17/3/2020 General Availability

https://jdk.java.net/14

Java 15 / JDK 15

?

September 2020!

https://bugs.openjdk.java.net/secure/Dashboard.jspa?selectPageId=19114
https://openjdk.java.net/projects/jdk/15/spec/

Java is still free!
Delivering Faster
Richest Feature Pipeline Ever

Innovating for the Future

ZGC
Create a scalable low latency garbage collector capable of handling large heaps

Loom
Massively scale lightweight threads, making concurrency simple again

Amber
Continuously improve developer productivity through evolutions of the Java language

Panama
Higher performance and easier development of I/O intensive applications through Java-native platform enhancements

Valhalla
Higher density and performance of machine learning and big data applications through the introduction of Value Types

Metropolis
Implement more of the JVM in Java starting with the JIT complier "Java-on-Java"

Zero GC

- A Scalable Low Latency GC
  - Low Latency ⇨ pause times stay below 10 ms, typically within 2 ms
  - Scalable ⇨ pause times do not increase with the heap or live-set size
  - Handle heaps ranging from a few hundred megabytes to multi terabytes in size

- Use
  - `-XX:+UnlockExperimentalVMOption -XX:+UseZGC`
- And tune
  - `-Xmx<size>`

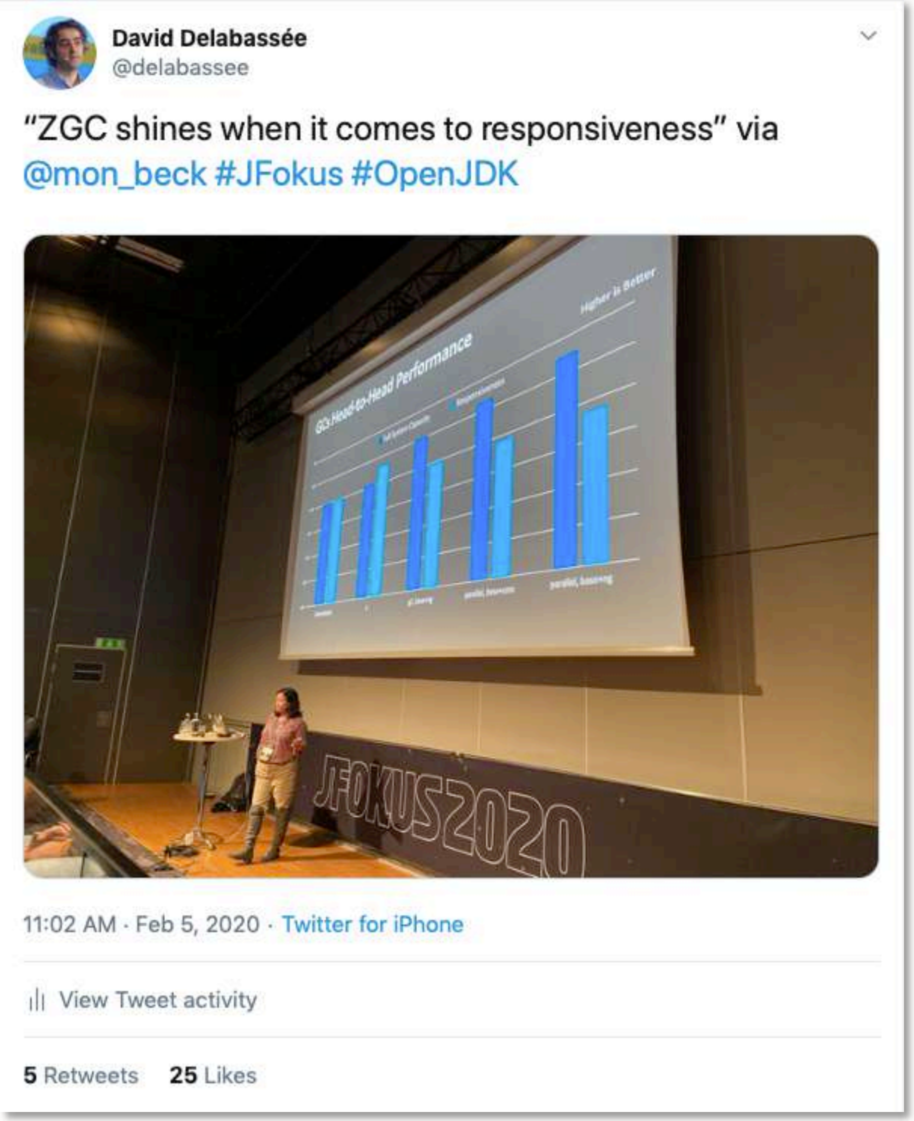
Zero GC

14

- JDK 11
  - Initial ZGC support on Linux (experimental)
- JDK 12
  - Support for concurrent class unloading, further pause time reduction
- JDK 13
  - Linux/AArch64 support, max heap size increased to 16TB
- JDK 14
  - MacOS (JEP 364) & Windows (JEP 365) support, JFR leak profiler, tiny heaps support (8mb), …

https://wiki.openjdk.java.net/display/zgc/Main

Zero GC



David Delabassée
@delabassee

"ZGC shines when it comes to responsiveness" via @mon_beck #JFokus #OpenJDK

11:02 AM · Feb 5, 2020 · Twitter for iPhone

View Tweet activity

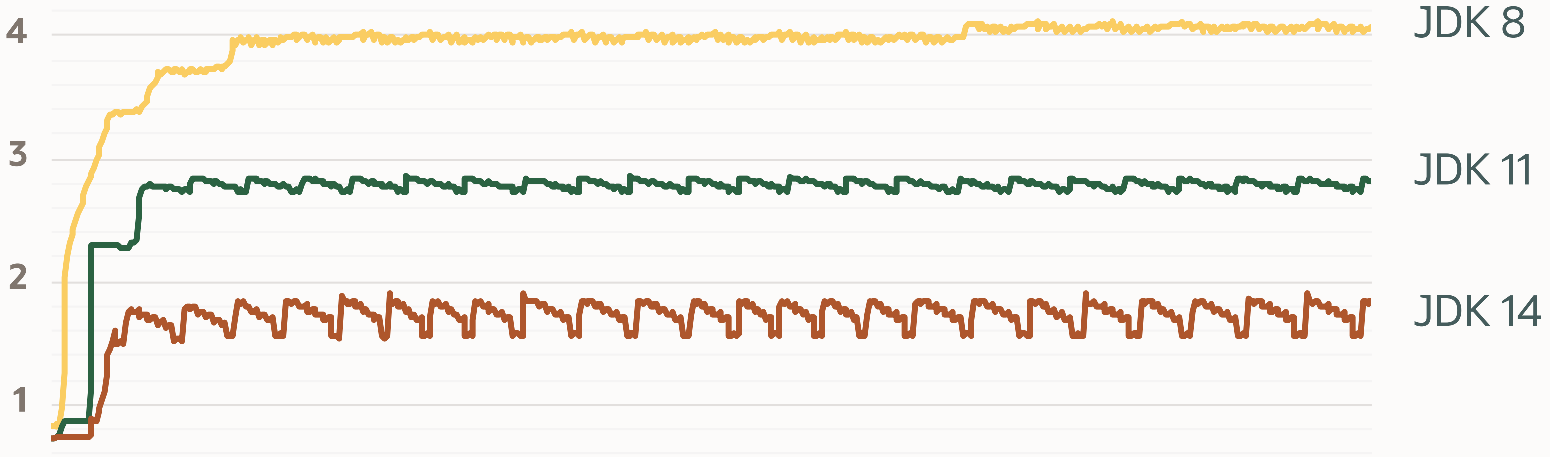
5 Retweets **25** Likes

https://www.jfokus.se/jfokus20-preso/OpenJDK-in-the-new-age-of-GC.pdf

G1 GC

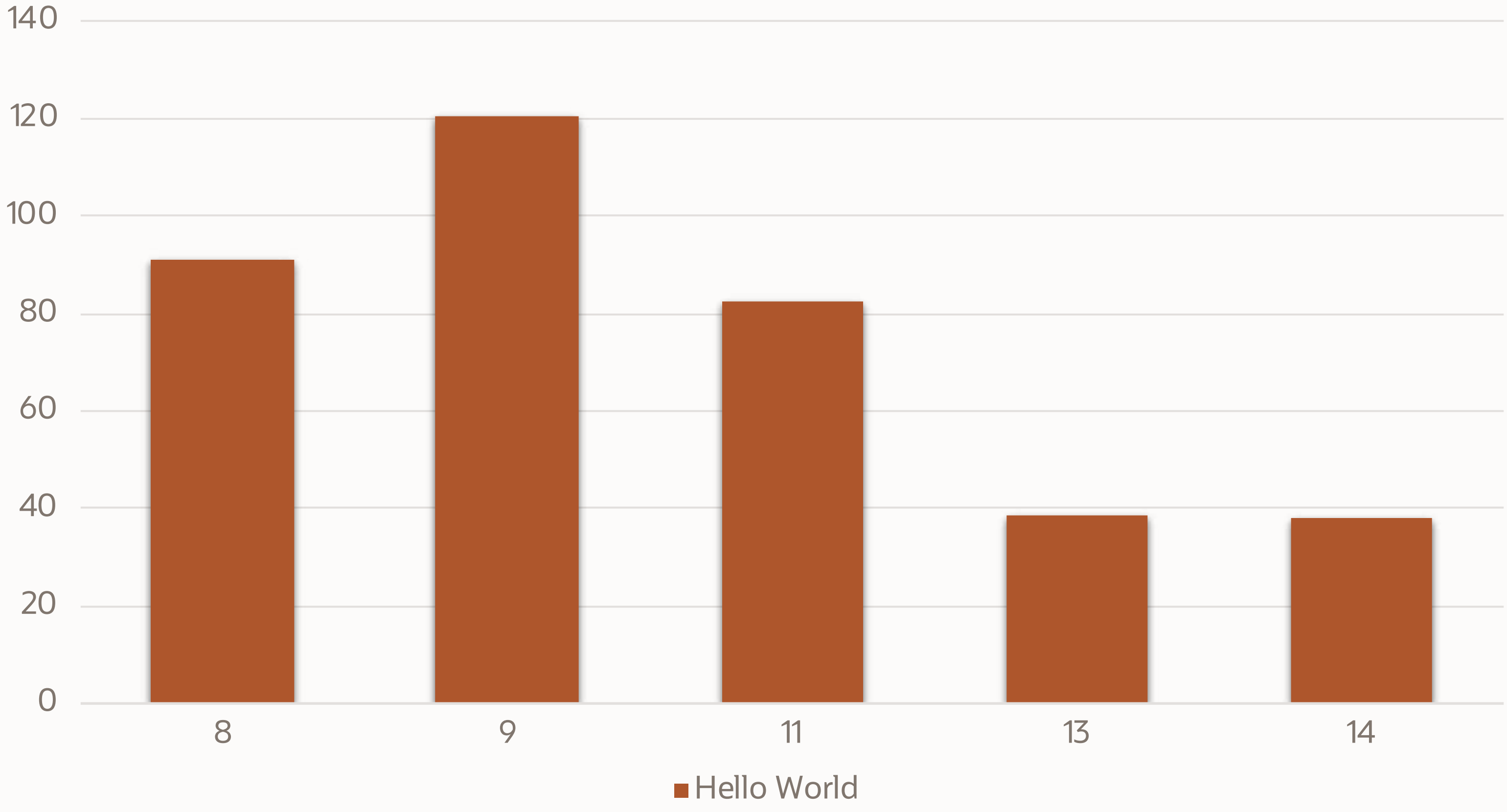
14

- NUMA-Aware Memory Allocation for G1 - JEP 345
- ~ 700 enhancements since JDK 8, across all areas!
  - Across all areas ⇨ significant improvements
- Ex. Native memory usage over time (GB)
  - BigRamTester, w. 16GB heap



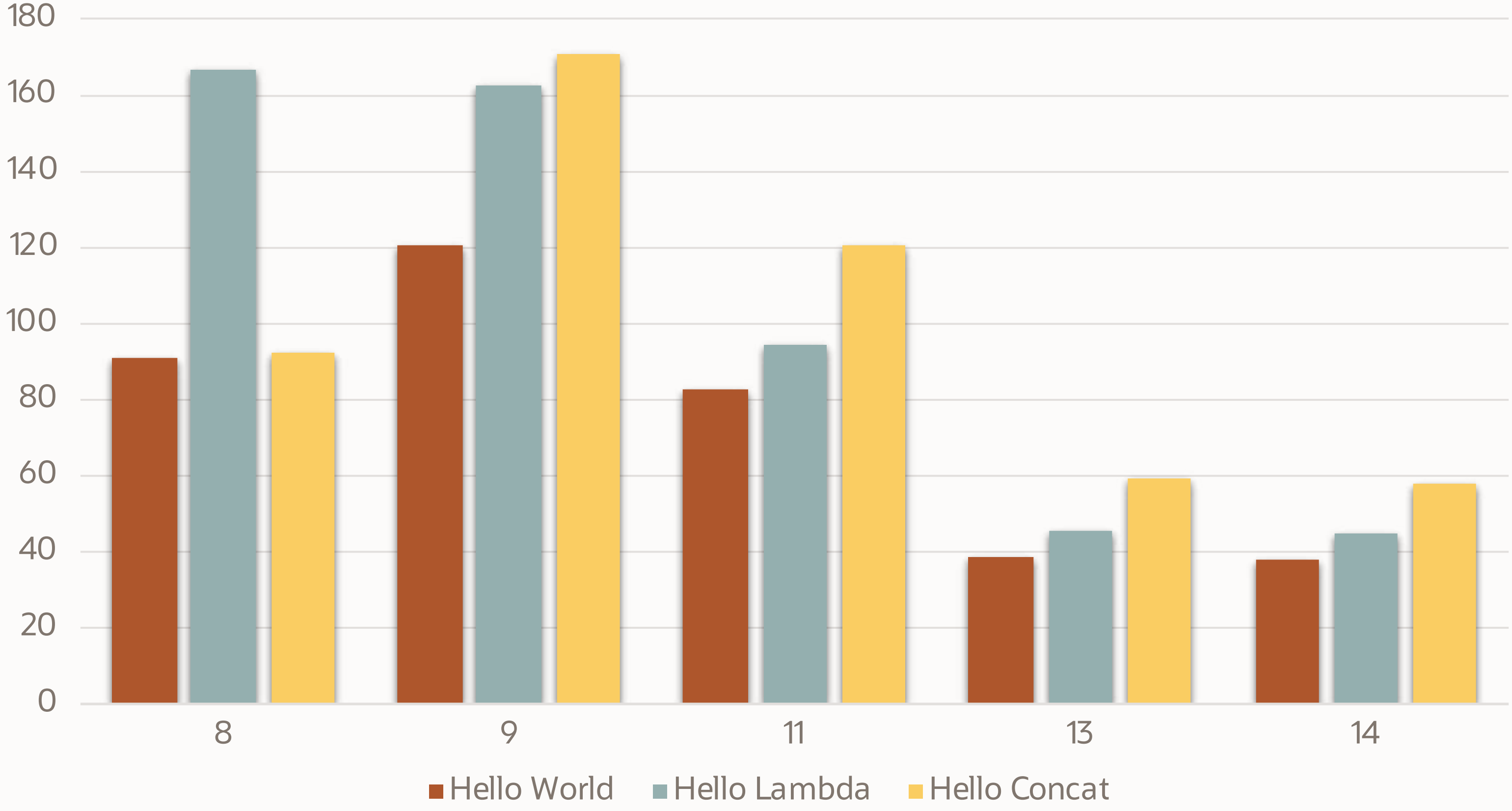
Startup Time

14



Startup Time

14



https://cl4es.github.io

Project Valhalla

- Enable flatter and denser memory layouts
- Main impediment to better object layout is object identity
    - Enables mutability, layout polymorphism, locking, etc.
    - Not all objects need it, but all objects pay for it!
    - Hard to dynamically determine at runtime whether identity will be relevant

Project Panama

- Foreign Function/Data interface
- Simple, safe, and performant replacement for JNI
- Access to low-level hardware functionality through normal Java code

Project Panama

14

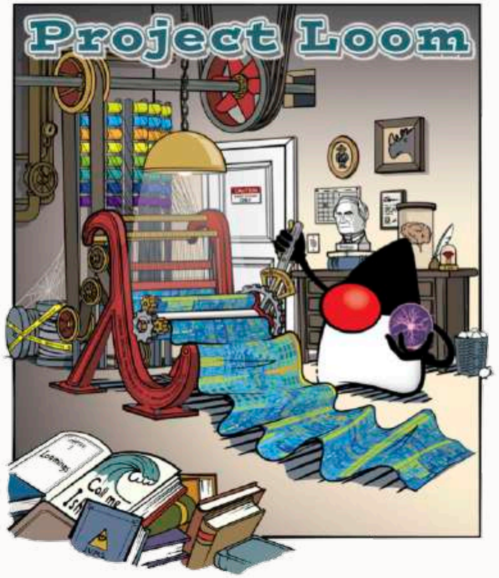
- Foreign-Memory Access API - JEP 370 (incubator)
  - Allows efficient off-heap memory access from Java
- “Extraction”
  - Tool to generate var/method handles from native library
  - API to customize the extraction process
- Vector API - JEP 338
  - Express vector computations that compile at runtime to optimal vector hardware instructions
- Non-Volatile Mapped Byte Buffers - JEP 352
  - Access to non-volatile memory (NVM) via MappedByteBuffer

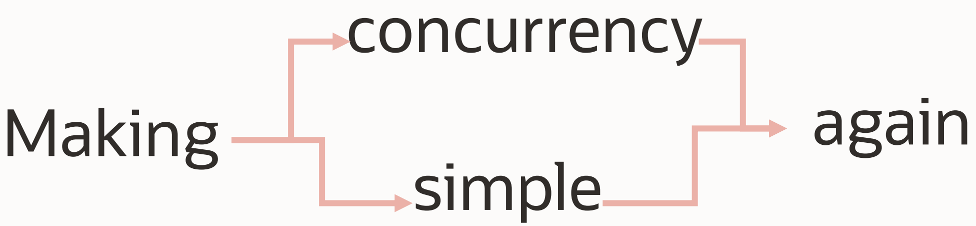
Panama

ITS'DEMO
Japan. Tax-Free Shop
ITS'DEMO
면세
TAX FREE
地下コスメ
TAX FREE

Project Loom

- Easier and more scalable concurrency model
- Virtual Threads vs. Kernel Threads
  - Making blocking calls virtually free
  - Millions of VT can be spawned in a single JVM instance!





Making

concurrency

simple

again

Project Amber

14

- Continuously improve developer productivity through evolutions of the Java language
- Delivered
  - Local-Variable Type Inference (`var`) - JDK 10
  - Local-Variable Syntax for Lambda Parameters - JDK 11
  - Switch Expressions - JDK 12 $^{\text{(Preview)}}$, JDK 13 $^{\text{(2nd Preview)}}$ & JDK 14 $^{\text{(Standard)}}$
  - Text Blocks - JDK 13 $^{\text{(Preview)}}$ & JDK 14 $^{\text{(2nd Preview)}}$
  - Records - JDK 14 $^{\text{(Preview)}}$
  - Pattern Matching `instanceof` - JDK 14 $^{\text{(Preview)}}$

Project Amber - Text Blocks JEP 368 (2nd preview)

14

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

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

Project Amber - Switch Expression JEP 361

14

```
switch(day){
   case MONDAY:
   case FRIDAY:
   case SUNDAY:
        numberOfChar = 6;
   case TUESDAY:
        numberOfChar = 7;
        break;
   case WEDNESDAY:
        numberOfChar = 9;
        break;
   case THURSDAY:
   case SATURDAY:
        numberOfChar = 8;
         break;
   default:
         throw new IllegalArgumentException…
}
```

```
int result = switch (day){
    case MONDAY, FRIDAY, SUNDAY -> 6;
    case TUESDAY -> 7;
    case WEDNESDAY -> 9;
    case THURSDAY, SATURDAY -> 8;
};
```

Project Amber - Records JEP 359 (preview)

14

- Provides a compact syntax for declaring classes which are transparent holders for shallowly immutable data
- Data carrier class with less code ceremony

Amber - Records

Project Amber - Pattern Matching with `InstanceOf` JEP 305 (preview)

14

```
if (obj instanceof String) {
   String s = (String) obj;
   // do something with s
}
```

```
if (obj instanceof String s) {
   // do something with s
}
else {
   // can’t use s here!
}
```

Amber - `instanceOf`

Innovating for the Future

ZGC
Create a scalable low latency garbage collector capable of handling large heaps

Loom
Massively scale lightweight threads, making concurrency simple again

Amber
Continuously improve developer productivity through evolutions of the Java language

Panama
Higher performance and easier development of I/O intensive applications through Java-native platform enhancements

Valhalla
Higher density and performance of machine learning and big data applications through the introduction of Value Types

Metropolis
Implement more of the JVM in Java starting with the JIT complier "Java-on-Java"

Helpful NullPointerExceptions JEP 358

14

```
java.lang.NullPointerException
        at Npe.locate(Npe.java:666)
        …
```

```
666 location.getCountry().getRegion().getProvince().getCity().getDistrict().getAddress()…
```

```
java -XX:+ShowCodeDetailsInExceptionMessages …
```

```
java.lang.NullPointerException:
        Cannot invoke "location$City.getDistrict()"
        because the return value of "Location$Province.getCity()" is null
        at Npe.locate(Npe.java:666)
        …
```

JDK Flight Recorder

- Event based tracing framework built into the JVM
  - High performance event recorder
  - Very low overhead, designed to be used in production
- Keeps history of tracing data always available, enables "after-the-fact" analysis
- Allows data from different subsystems and software layers to be correlated
- Interfaces
  - CLI: JVM flags, jfr, jcmd
  - GUI: JDK Mission Control
  - APIs: Java & JMX

JFR Event Streaming JEP 349

14

- Expose JFR data for continuous monitoring
- Stream event data as it is being produced, no need to dump data to a file
- API for the continuous consumption of events
  - In-process and out-of-process
- Low overhead (<1% overhead), safe for production

JDK Flight Recorder

14

- Event Types

JDK 10	125
JDK 11	131
JDK 12	136
JDK 13	143
JDK 14 $^{\text{(rc1)}}$	145
JDK 15 $^{\text{(ea-loom+3-2)}}$	154

https://docs.oracle.com/en/java/javase/13/docs/api/jdk.jfr/jdk/jfr/EventType.html

Packaging Tool JEP 343 (incubator)

14

- `jpackage`
- Give end users a natural, i.e. native, installation experience
  - Windows: msi & exe
  - macOS: pkg & dmg
  - Linux: deb & rpm
- Allows launch-time parameters to be specified at packaging time
- Can be invoked directly, from the command line, or programmatically, via the ToolProvider API

JVM Container Awareness

JDK-8186248	More flexibility in selecting Heap % of available RAM (8u144)
JDK-8179498	attach should be relative to `/proc/pid/root` and namespace aware as `jcmd`, `jstack`, ... fail to attach (10)
JDK-8146115	Improve Docker container detection & resource config usage (10)
JDK-8193710	`jcmd -l` & `jps` do not list Java processes running in containers (11)
JDK-8203357	Container Metrics (11)
JDK-8220786	Create new switch to redirect error reporting output to `stdout` or `stderr` (13)
JDK-8203359	JFR `jdk.CPUInformation` event reports incorrect information when running in container (in progress)
…	…
JDK-8230305	Cgroups v2: Container awareness (15)

https://bugs.openjdk.java.net

Wrap-up

Java is still free!
Delivering Faster
Richest Feature Pipeline Ever

https://openjdk.java.net

Java 14 / JDK 14 - 16 JEPs

14

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

20/2/2020	Final Release Candidate
17/3/2020	General Availability

https://jdk.java.net/14

Thanks!

David Delabassée
@delabassee