Desde Java 8 hasta Java 17

Víctor Orozco - @tuxtor

5 de octubre de 2021

Nabenik

¿Como se hace Java?

De Java 8 hasta Java 17

Java 9

Java 10

Java 11

Java 12

Java 13

Java 14

Java 15

Java 16

Java 17

¿Como se hace Java?

¿Java?

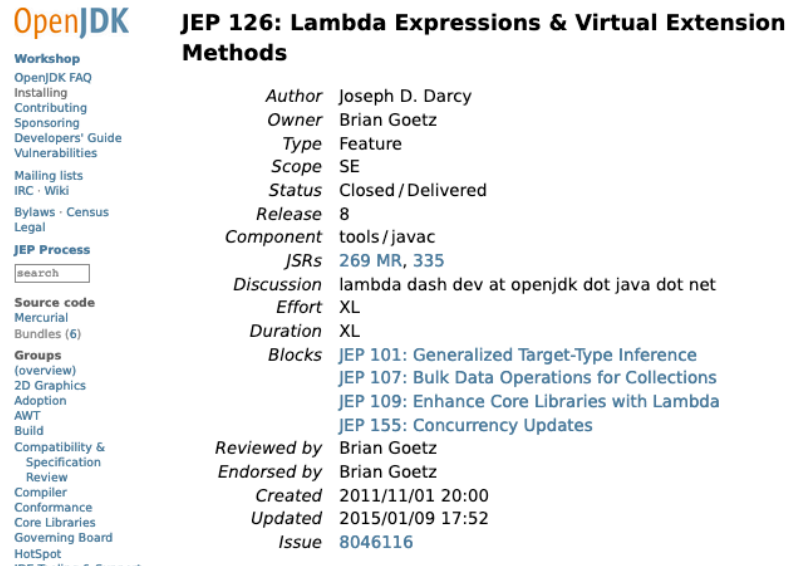
- Lenguaje
- VM
- Bibliotecas/API

El conjunto es la plataforma Java (TM)

¿Como se actualiza Java?

- JCP - Java Community Process
- JSR - Java Specification Request
- JEP - Java Enhancement Proposal
- JCK - Java Compatibility Kit

¿Como se actualiza Java? - Java Enhancement Proposal



OpenJDK

Workshop
OpenJDK FAQ
Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities

Mailing lists
IRC · Wiki

Bylaws · Census
Legal

JEP Process

search

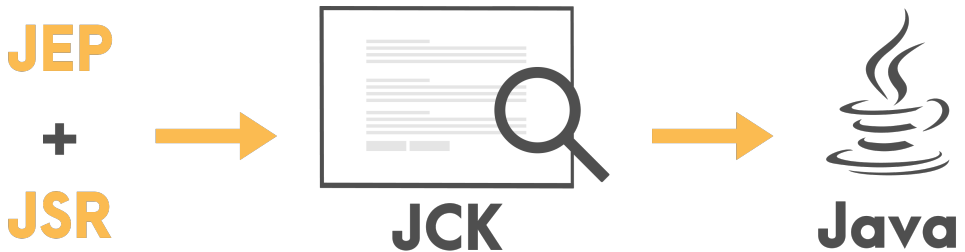
Source code
Mercurial
Bundles (6)

Groups
(overview)
2D Graphics
Adoption
AWT
Build
Compatibility & Specification Review
Compiler
Conformance
Core Libraries
Governing Board
HotSpot

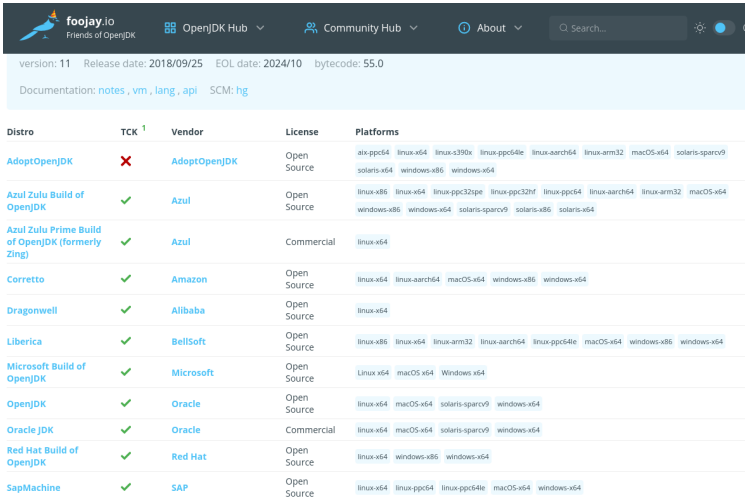
JEP 126: Lambda Expressions & Virtual Extension Methods

Author	Joseph D. Darcy
Owner	Brian Goetz
Type	Feature
Scope	SE
Status	Closed / Delivered
Release	8
Component	tools / javac
JSRs	269 MR, 335
Discussion	lambda dash dev at openjdk dot java dot net
Effort	XL
Duration	XL
Blocks	JEP 101: Generalized Target-Type Inference JEP 107: Bulk Data Operations for Collections JEP 109: Enhance Core Libraries with Lambda JEP 155: Concurrency Updates
Reviewed by	Brian Goetz
Endorsed by	Brian Goetz
Created	2011/11/01 20:00
Updated	2015/01/09 17:52
Issue	8046116

¿Como se actualiza Java? - Java Compatibility Kit



Distros



Distro	TCK [1]	Vendor	License	Platforms
AdoptOpenJDK	✗	AdoptOpenJDK	Open Source	aix-ppc64 linux-x64 linux-s390x linux-ppc64le linux-aarch64 linux-arm32 macOS-x64 solaris-sparcv9 solaris-x64 windows-x86 windows-x64
Azul Zulu Build of OpenJDK	✓	Azul	Open Source	linux-x86 linux-x64 linux-ppc32spe linux-ppc32hf linux-ppc64 linux-aarch64 linux-arm32 macOS-x64 windows-x86 windows-x64 solaris-sparcv9 solaris-x86 solaris-x64
Azul Zulu Prime Build of OpenJDK (formerly Zing)	✓	Azul	Commercial	linux-x64
Corretto	✓	Amazon	Open Source	linux-x64 linux-aarch64 macOS-x64 windows-x86 windows-x64
Dragonwell	✓	Alibaba	Open Source	linux-x64
Liberica	✓	BellSoft	Open Source	linux-x86 linux-x64 linux-arm32 linux-aarch64 linux-ppc64le macOS-x64 windows-x86 windows-x64
Microsoft Build of OpenJDK	✓	Microsoft	Open Source	Linux x64 macOS x64 Windows x64
OpenJDK	✓	Oracle	Open Source	linux-x64 macOS-x64 solaris-sparcv9 windows-x64
Oracle JDK	✓	Oracle	Commercial	linux-x64 macOS-x64 solaris-sparcv9 windows-x64
Red Hat Build of OpenJDK	✓	Red Hat	Open Source	linux-x64 windows-x86 windows-x64
SapMachine	✓	SAP	Open Source	linux-x64 linux-ppc64 linux-ppc64le macOS-x64 windows-x64

De Java 8 hasta Java 17

¿Que versión de Java se usa más en Guatemala?

Java 8
https://guatejug.github.io/jvm2021/

¿Que recibo con cada versión nueva de Java?

- Java - **Lenguaje**
- Java - Bibliotecas e APIs
- Java - Maquina Virtual de Java

Mejoras de tipo

- Incubation
- Preview (–flag)
- Definitiva

Java - Las mejoras que resaltan

- Java 9
  - Modulos
  - JShell
  - HTTP/2
  - Factory methods
- Java 10
  - Type Inference
  - Class Data Sharing
  - Time based release
- Java 11
  - String methods
  - File methods
  - Direct .java execution
- Java 12
  - Switch expressions
- Java 13
  - Text blocks
- Java 14
  - Pattern matching
  - Records
  - Helpfull NPE

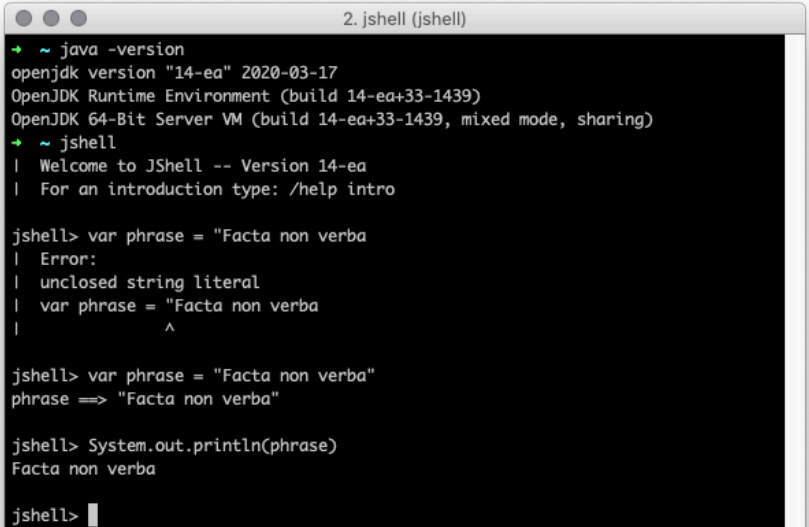
Java - Las mejoras que resaltan

- Java 15
  - Sealed classes
  - Nashorn removal
  - SPARC/Solaris removal
- Java 16
  - Vector API (Incubator)
  - C++ 14
  - Records

- Java 17
  - MacOS Metal
  - Strong encapsulation
  - Switch expression + pattern matching
  - AOT removal
  - Applet deprecation

Java 9

JEP 222: jshell: The Java Shell (Read-Eval-Print Loop)



JEP 110: HTTP/2 Client

```
1  HttpRequest request = HttpRequest.newBuilder()
2      .uri(new URI("https://swapi.co/api/starships/9"))
3      .GET()
4      .build();
5
6  HttpResponse<String> response = HttpClient.newHttpClient()
7      .send(request, BodyHandlers.ofString());
8
9  System.out.println(response.body());
```

JEP 269: Convenience Factory Methods for Collections

Antes

```
1 Set<String> set = new HashSet<>();
2 set.add("a");
3 set.add("b");
4 set.add("c");
5 set = Collections.unmodifiableSet(set);
```

”Pro”

```
1 Set<String> set = Collections.unmodifiableSet(new HashSet<>(
      Arrays.asList("a", "b", "c")));
```

Ahora

```
1 Set<String> set = Set.of("a", "b", "c");
```

JEP 213: Milling Project Coin - Private methods in interfaces

Antes

```
1  public interface Vehicle{
2      public void move();
3  }
```

Ahora

```
1  public interface Vehicle{
2      public default void makeNoise(){
3          System.out.println("Making noise!");
4          createNoise();
5      }
6
7      private void createNoise(){
8          System.out.println("Run run");
9      }
10 }
```

JEP 213: Milling Project Coin - Try-with-resources

Antes

```
1  BufferedReader reader = new BufferedReader(new FileReader("
       langs.txt"));
2
3  try(BufferedReader innerReader = reader){
4      System.out.println(reader.readLine());
5  }
```

Ahora

```
1  BufferedReader reader = new BufferedReader(new FileReader("
       langs.txt"));
2
3  try(reader){
4      System.out.println(reader.readLine());
5  }
```

Java 10

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

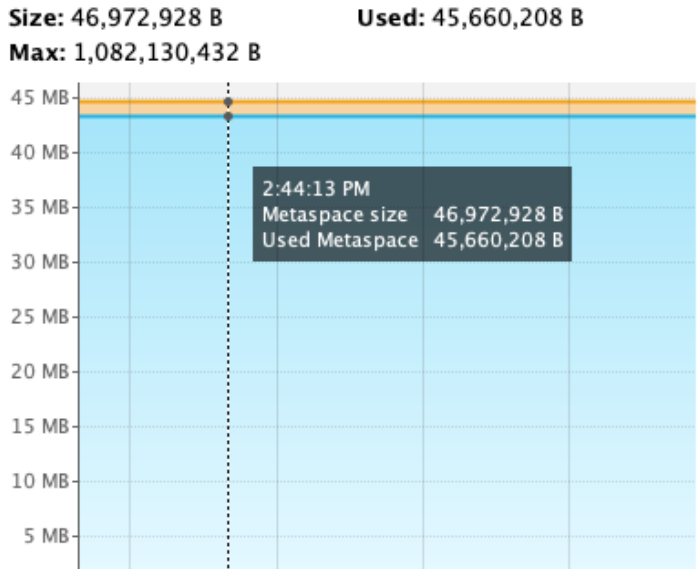
JEP 286: Local-Variable Type Inference

```
1 public static void main(String args[]){
2     var localValue = 99;
3     System.out.println(++localValue);
4     //localValue = "Foo"
5 }
```

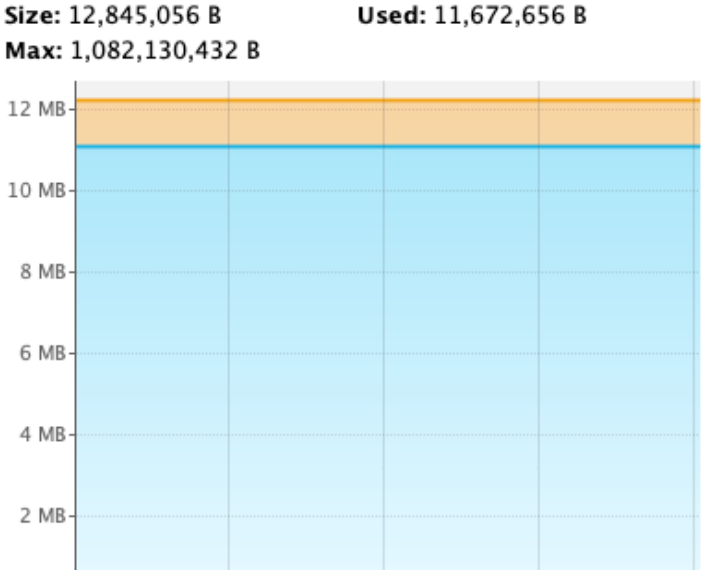
JEP 310: Application Class-Data Sharing

```
java -XX:+UseAppCDS -XX:DumpLoadedClassList=classes.lst -jar demo-microservicio-ee-microbundle.jar                    1
java -XX:+UseAppCDS -Xshare:dump -XX:SharedClassListFile=classes.lst -XX:SharedArchiveFile=app-cds.jsa demo-microservicio-ee-microbundle.jar    2
java -XX:+UseAppCDS -Xshare:on -XX:SharedArchiveFile=app-cds.jsa -jar demo-microservicio-ee-microbundle.jar           3
```

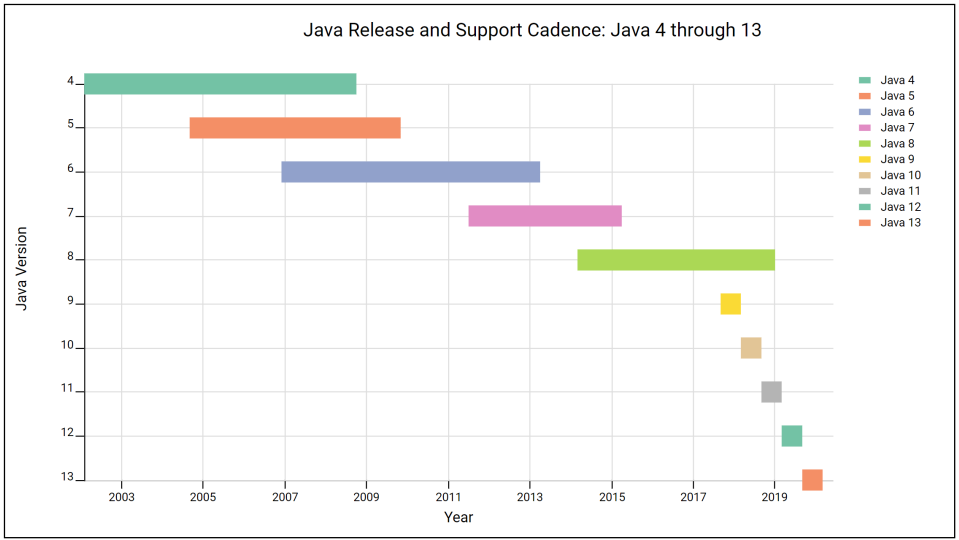
JEP 310: Application Class-Data Sharing



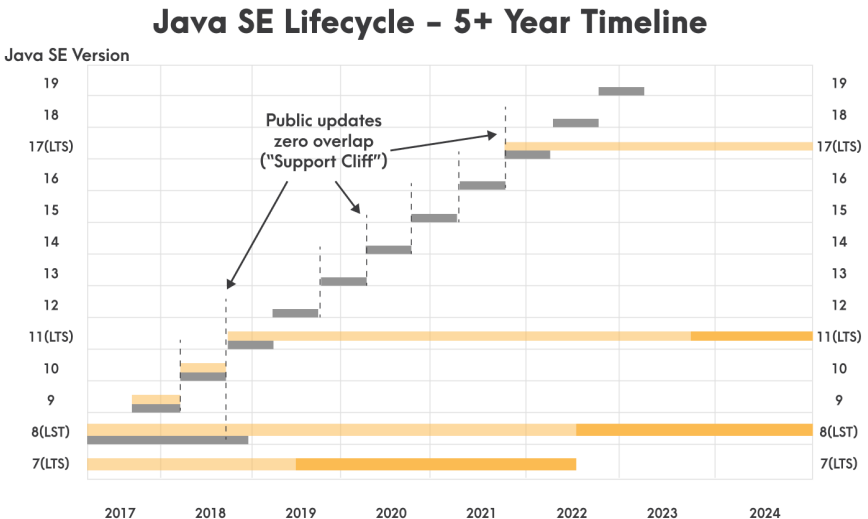
JEP 310: Application Class-Data Sharing



JEP 322: Time-Based Release Versioning



JEP 322: Time-Based Release Versioning



Java 11

Java 11

181: Nest-Based Access Control
309: Dynamic Class-File Constants
315: Improve Aarch64 Intrinsics
318: Epsilon: A No-Op Garbage Collector
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

JEP 323: Local-Variable Syntax for Lambda Parameters

Antes

```
1 BiPredicate<String,String> demoPredicate =
2     (String a, String b) -> a.equals(b);
3 BiPredicate<String,String> demoPredicate =
4     (a, b) -> a.equals(b);
```

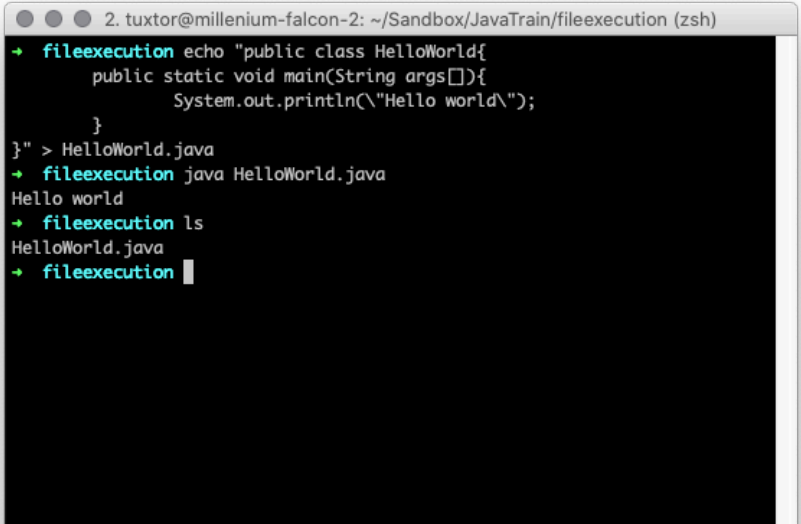
Ahora

```
1 BiPredicate<String,String> demoPredicate =
2     (var a, var b) -> a.equals(b);
```

Posibilidades

```
1 (@Nonnull var x, @Nullable var y) -> x.process(y)
```

JEP 330: Launch Single-File Source-Code Programs



Java 12

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

325: Switch Expressions (Preview)

Antes

```
1  String langType = "";
2  switch (args[0]) {
3      case "Java":
4      case "Scala":
5      case "Kotlin":
6          langType = "Static typed";
7          break;
8      case "Groovy":
9      case "JavaScript":
10         langType = "Dynamic typed";
11         break;
12 }
13 System.out.println(langType);
```

325: Switch Expressions (Preview)

Ahora

```
1  String langType = switch (args[0]) {
2      case "Java", "Scala", "Kotlin" -> "Static typed";
3      case "Groovy", "JavaScript" -> "Dynamic typed";
4      default -> {
5          System.out.println("This meant to be a processing
               block");
6          yield "Probably LISP :)";
7      }
8  };
9  System.out.println(langType);
```

Java 13

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

355: Text Blocks (Preview)

Antes

```
1 String html = "<html>\n" +
2 "    <body>\n" +
3 "        <p>Hello, world</p>\n" +
4 "    </body>\n" +
5 "</html>\n";
```

Ahora

```
1 String html = """
2 <html>
3     <body>
4         <p>Hello, world</p>
5     </body>
6 </html>
7 """;
```

Java 14

Java 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)

JEP 359: Records (Preview)

Data carrier

```
1 record Person(String name, String email, int age) {}
```

Uso

```
1 Person foo = new Person("Marco", "example@mail.com",99);
2 System.out.println(foo);
3 //foo.name = "Polo";
```

305: Pattern Matching for instanceof (Preview)

Antes

```
1  if(o instanceof Person){
2      Person p = (Person)o;
3      System.out.println("Hello " + p.name());
4  }else{
5      System.out.println("Unknown object");
6  }
```

Ahora

```
1  if(o instanceof Person p){
2      System.out.println("Hello " + p.name());
3  }else{
4      System.out.println("Unknown object");
5  }
```

Java 15

Java 15

339: Addition of EdDSA (Edwards-Curve Digital Signature Algorithm)
360: Preview for sealed classes
371: Addition of hidden classes in Java
372: Removal of the Nashorn JavaScript Engine
373: Legacy DatagramSocket API reimplementation
374: Biased locking disablement and deprecation
375: Second preview for instanceof pattern matching
377: Addition of ZGC, the scalable, low-latency, garbage collector for Java
378: Full inclusion of Java Text Blocks
379: Addition and enhancements of the low-pause time Shenandoah garbage collector
381: Final remove of Solaris and SPARC Ports
383: Incubation of the Foreign-Memory Access API
384: Second preview inclusion of Java Records
385: RMI Activation deprecation with the goal of future removal

360: Sealed Classes (Preview)

Antes

```
1 public class Animal { ... }
2
3 public class Porsche extends Animal { ... }
```

Ahora

```
1 public abstract sealed class Animal
2     permits Dog, Cat, Wolf { ... }
```

360: Sealed Classes (Preview)

Antes

```
1 public class Animal { ... }
2
3 public class Porsche extends Animal { ... }
```

Pro

```
1 abstract sealed class Animal {
2         final class Dog extends Root { ... }
3         final class Cat extends Root { ... }
4         final class Wolf extends Root { ... }
5 ... }
```

371: Hidden Classes

Antes

```
1 ClassLoader::defineClass
```

Ahora

```
1 Lookup::defineHiddenClass
```

381: Remove the Solaris and SPARC Ports

383: Foreign-Memory Access API (Second Incubator)

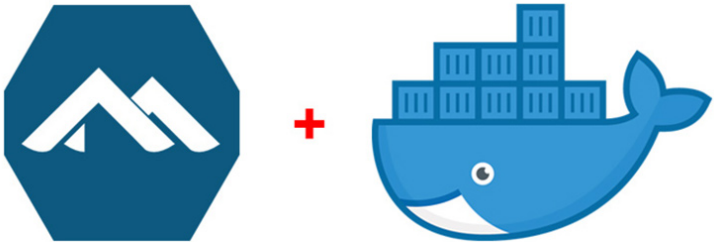
```
1  VarHandle intHandle = MemoryHandles.varHandle(int.class,
2          ByteOrder.nativeOrder());
3
4  try (MemorySegment segment = MemorySegment.allocateNative(100)
       ) {
5      for (int i = 0; i < 25; i++) {
6          intHandle.set(segment, i * 4, i);
7      }
8      ...
```

Java 16

Java 16

338: Incubation of the Vector API
347: C++14 language features enablement
357: Migration from Mercurial to Git
369: GitHub migration of OpenJDK repositories
376: Concurrent thread-stack processing for ZGC
380: Unix-Domain Socket Channels
386: Alpine Linux port
387: Support for elastic metaspace
388: The Windows AArch64 Port
389: Incubation of the Foreign Linker API
390: Value-based classes warnings
392: Addition of the Packaging Tool
393: Third incubation of the Foreign-Memory Access API
394: Full inclusion of pattern matching for instanceof
395: Full implementation of Java Records
396: Strongly encapsulation of JDK Internals
397: Second preview of sealed classes

386: Alpine Linux Port



389: Foreign Linker API (Incubator)

```
1 LibraryLookup libclang = LibraryLookup.ofLibrary("clang");
2 LibraryLookup.Symbol clangVersion = libclang.lookup("
    clang_getClangVersion");
```

396: Strongly Encapsulate JDK Internals by Default

Antes

```
--illegal-access=permit
```

Ahora

```
--illegal-access=deny
```

Java 17

Java 17

306: Restoration of always-strict semantics for floating-points
356: Pseudo-random number generators improvements
382: Support for a new macOS pipeline for rendering
391: The macOS AArch64 port
398: Deprecation of the infamous Applet API with the goal of full removal
403: Strong encapsulation of Java's internals
406: Switch pattern matching preview
407: RMI Activation removal
409: Full support for sealed Java classes
410: Removal of the experimental and largely unused ahead-of-time (AOT) and just-in-time (JIT) compilers
411: Deprecation of the Security Manager with removal being the eventual goal
412: Incubation of the Foreign Function & Memory API
414: Second incubation of the Vector API
415: Context-specific deserialization filters

406: Pattern Matching for switch (Preview)

```
1  static void testTriangle(Shape s) {
2      switch (s) {
3          case Triangle t && (t.calculateArea() > 100) ->
4              System.out.println("Large triangle");
5          case Square s ->
6              System.out.println("Is not a triangle is a square"
                   );
7          default ->
8              System.out.println("Non-triangle");
9      }
10 }
```

Nueva licencia Oracle Java 17

https://blogs.oracle.com/java/post/free-java-license 90%

Introducing the Free Java License

Donald Smith

September 14, 2021

JEPs in JDK 17 integrated since JDK 11

https://openjdk.java.net/projects/jdk/17/jeps-since-jdk-11

JEPs in JDK 17 integrated since JDK 11

Here are all of the JEPs integrated since the previous long-term-support (LTS) release, JDK 11. Incubator and Preview JEPs that were superseded by later JEPs in JDKs 12 through 17 are not included. The release in which a JEP was integrated is shown in parentheses after the JEP's title.

Additions

403: Strongly Encapsulate JDK Internals (17)
390: Warnings for Value-Based Classes (16)

HotSpot JVM

386: Alpine Linux Port (16)
391: macOS/AArch64 Port (17)
340: One AArch64 Port, Not Two (12)
388: Windows/AArch64 Port (16)

Flight Recorder
349: JFR Event Streaming (14)

Garbage Collectors
344: Abortable Mixed Collections for G1 (12)
345: NUMA-Aware Memory Allocation for G1 (14)
346: Promptly Return Unused Committed Memory from G1 (12)

Oracle Groundbreakers

Java Champions

ORACLE Certified Professional
Java SE 8 Programmer

ORACLE Certified Associate
Java SE 8 Programmer

- vorozco@nabenik.com
- @tuxtor
- http://vorozco.com
- http://tuxtor.shekalug.org