

MIDI TECH

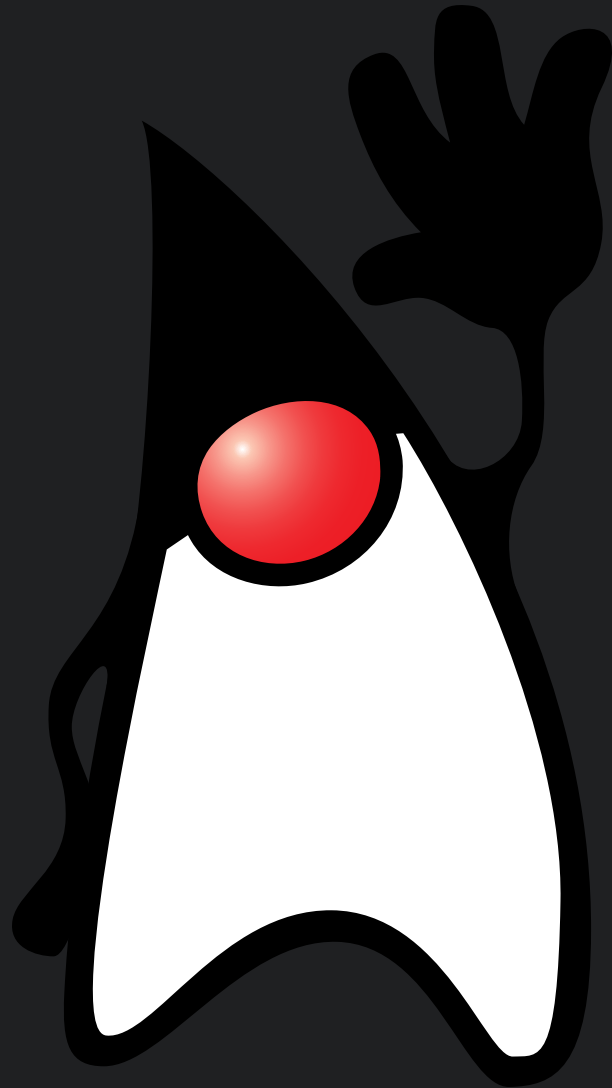
L'après Java 8, les nouveautés depuis Java 9 !

François Delbrayelle (@fdelbrayelle)

BIENVENUE !

- Nouvelle saison des midis techniques
- Sujet dense = questions à la fin =)
- **Sondage : qui utilise quelle version ?**

Java évolue !

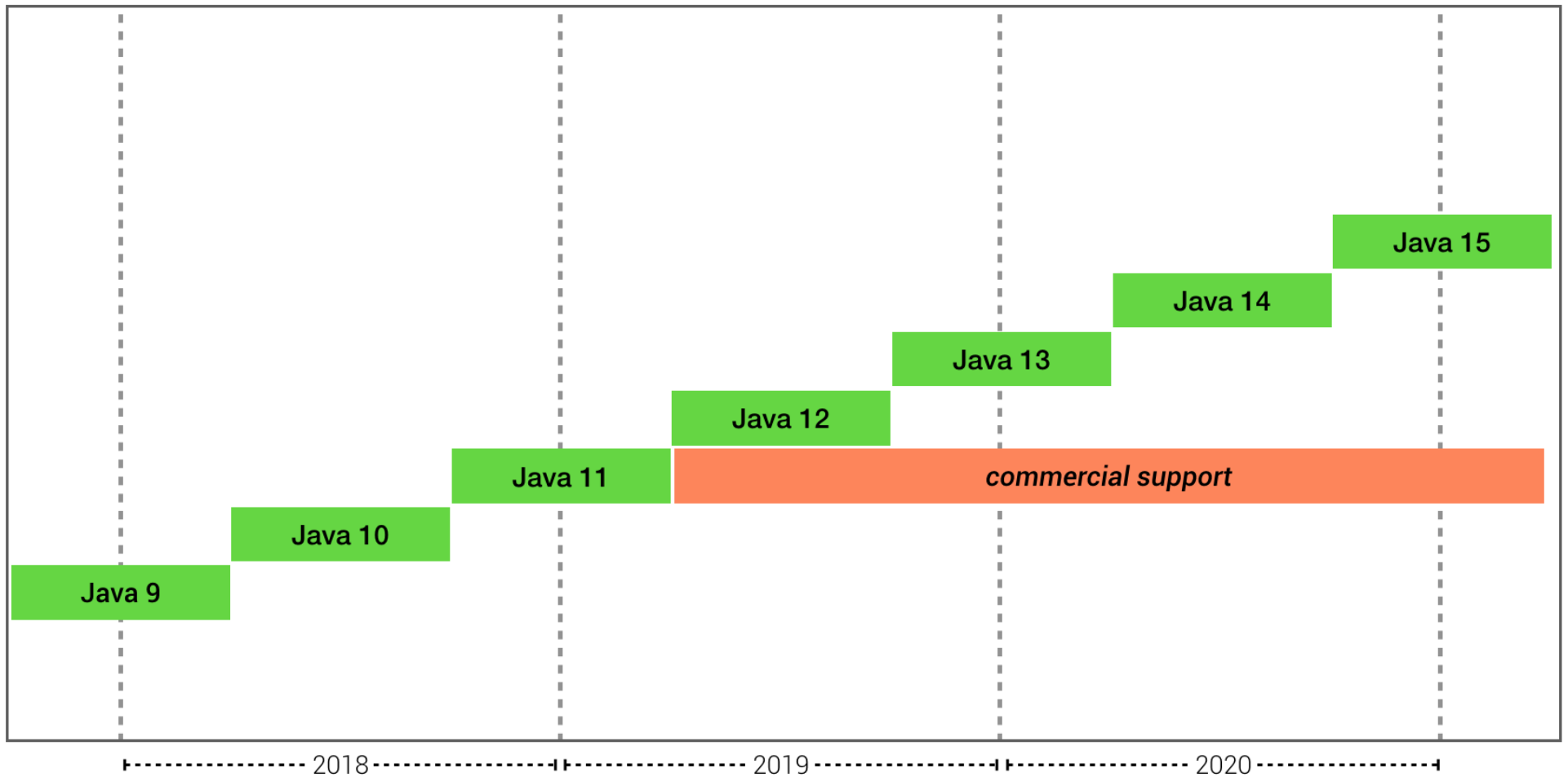


AVANT JAVA 10

- Releases de tailles trop importantes
- Sorties trop espacées dans le temps
- Nouvelles versions plus compliquées

À PARTIR DE JAVA 10

- Release train avec sortie tous les 6 mois
- Release à date de ce qui est intégré
- Moins de nouvelles fonctionnalités



Source : <https://dzone.com/articles/what-the-future-java-releases-will-mean-for-legacy>

SUPPORT

- Support LTS toutes les 6 releases pour 3 ans
- Support non LTS pour 6 mois
- Support payant étendu possible par Oracle (licence par CPU ou par nom d'utilisateur)
- Java 8 en fin de vie depuis janvier 2019 (support payant possible jusqu'en 2025)
- <https://www.oracle.com/technetwork/java/java-se-support-roadmap.html>

Issues fixed in JDK 11 per organization



Source : <https://blogs.oracle.com/java-platform-group/building-jdk-11-together>

QUELLE JDK ?

- Deux versions identiques en contenu, différentes en contrat : OpenJDK et OracleJDK
- OpenJDK est open-source (version Oracle par défaut)
- Support limité dans le temps mais gratuit pour OpenJDK
- Autres JDKs avec support plus étendu : AdoptOpenJDK, Amazon Corretto, Azul, Eclipse OpenJ9, ...

Java SE 9

Septembre 2017



<https://openjdk.java.net/projects/jdk9/>

PROJET JIGSAW (JAVA PLATFORM MODULE SYSTEM) - JSR 376

- JDK découpée en modules : `java --list-modules`
- Un module définit les services qu'il expose et les modules qu'il utilise (packages et ressources)
- Nouveau fichier descripteur de modules à la racine des packages : `module-info.java`

PROJET JIGSAW (JAVA PLATFORM MODULE SYSTEM) - JSR 376

- Possibilité de créer une JRE personnalisée avec uniquement les modules nécessaires grâce à `jlink`
- Modules pas obligatoires si migration
- Adaptés pour les architectures microservices

PROJET JIGSAW (JAVA PLATFORM MODULE SYSTEM) - JSR 376

```
module world.gfi.intranet {  
  exports world.gfi.intranet.users;  
  exports world.gfi.intranet.projects;  
}
```

```
module world.gfi.cra {  
  exports world.gfi.cra;  
  requires world.gfi.intranet;  
  requires org.apache.commons.lang3;  
}
```

COMPILATION AOT (AHEAD-OF-TIME) - JEP 295

- JIT : JVM HotSpot exécute du bytecode et compile fréquemment le code exécuté vers du code natif
- Compiler les classes en code natif avant de lancer la JVM
- Commande `jaotc` après `java`
- Génération de fichiers librairies `.so`

AMÉLIORATION DES PERFORMANCES

- Garbage-first collector (G1) - JEP 248 : GC par défaut
- Compact Strings - JEP 254
(implémentation plus compacte si String ISO-8859-1 ou Latin-1). Avant : Une `String` était une séquence de `char` (unitairement stocké en interne en UTF-16 = 2 octets)
- Indify String Concatenation - JEP 280 : intrasification de la concaténation des chaînes (utilisation de "`+`" optimisée)

ENHANCED DEPRECATION - JEP 277

- Attribut `since` sur `@Deprecated` (depuis quelle version de Java ?)
- Attribut `forRemoval` sur `@Deprecated` (indique la suppression future)

"SUCRE SYNTAXIQUE" - JEP 213

- Autorisation des variables finales dans les `try-with-resource`
- `@SafeVarargs` sur méthodes privées (auparavant uniquement sur statiques ou finales)
- Autorisation de l'opérateur diamant sur les classes abstraites quand le type est implicite
- Méthodes privées dans les interfaces

THE JAVA SHELL (A JAVA REPL) - JEP 222

- Read-eval-print loop (interpréteur en ligne de commandes)
- Commande `jshe11`
- **Démo !**

CONVENIENCE FACTORY METHODS FOR COLLECTIONS - JEP 269

- Méthodes statiques pour créer des collections immutables
- Petites collections (`List`, `Set`, `Map`)
- Objectif : performances
- **Démo !**

MORE CONCURRENCY UPDATES - JEP 266

- Implémentation des Reactive Streams via la classe `Flow`
- Programmation réactive : Publisher avec `subscribe(Subscriber)`, Subscriber avec `onNext(T)`, Subscription
- Alternative à Vert.x, RxJava ou Akka
- Améliorations pour l'API `CompletableFuture` (timeout, delay)

NOUVEAUTÉS STREAM

- `Stream::takeWhile`
- `Stream::dropWhile`
- `Stream::ofNullable`
- `Stream::iterate` (avec un `Predicate` : boucle for standard)
- **Démo !**

NOUVEAUTÉS COLLECTORS

- `Collectors::filtering`
- `Collectors::flatMaping`
- **Démo !**

NOUVEAUTÉS OPTIONAL

- `Optional::or(Supplier)`
- `Optional::ifPresentOrElse(Consumer, Runnable)`
- `Optional::stream()`
- **Démo !**

NOUVEAUTÉS JAVA.TIME

- `LocalDate::datesUntil(LocalDate)`
- `LocalDate::datesUntil (LocalDate, Period)`
- **Démo !**

NOUVEAUTÉS ARRAYS

- `Arrays.equals()`
- `Arrays.compare()`
- `Arrays.compareUnsigned()`
- `Arrays.mismatch()`
- **Démo !**

NOUVEAUTÉS PROCESS API

- Possibilité d'exécuter des commandes externes depuis Java
- Compatible avec les différents OS supportés par Java
- Processus de la JVM :
`ProcessHandle.current()`
- Processus fils :
`Runtime.getRuntime().exec("...")`

Java SE 10

Mars 2018



<https://openjdk.java.net/projects/jdk/10/>

LOCAL-VARIABLE TYPE INFERENCE - JEP 286

- Nouveau mot-clé `var` (variable locale uniquement)
- Possibilité d'utiliser `final`
- Inutilisable avec l'opérateur diamant
- Possible dans une boucle `for` ou un `try-with-resource`
- **Démo !**

COLLECTION COPY FACTORY METHODS

- Nouvelles méthodes statiques `copyOf` sur les collections (`List`, `Set`, `Map`)
- Retourne une collection immuable
- **Démo !**

NOUVEAUTÉS DIVERSES

- `Collectors::toUnmodifiableList`
- `Optional.orElseThrow()` : à préférer à `Optional.get()`
- `Reader.transferTo(Writer)`
- **Démo !**

PARALLEL FULL GC FOR G1 - JEP 307

- Auparavant Full GC mono-threadé pour nettoyer la heap
- Nombre de threads avec `-XX:ParallelGCThreads`

APPLICATION CLASS- DATA SHARING - JEP 310

- CDS depuis JDK 1.5 seulement pour les classes système
- Classes applicatives dans une archive partagée
- Réduire le temps de démarrage

Java SE 11

Septembre 2018 (LTS)



<https://openjdk.java.net/projects/jdk/11/>

LOCAL-VARIABLE SYNTAX FOR LAMBDA PARAMETERS - JEP 323

- Utiliser des annotations sans perdre en concision
- `(var x, var y) -> x.process(y)`
- **Démo !**

HTTP CLIENT - JEP 321

- En incubation dans Java 9 et 10
- Nouvelle API pour remplacer `HttpURLConnection`
- Synchrone ou asynchrone (basé sur les `CompletableFuture`)
- **Démo !**

LAUNCH SINGLE-FILE SOURCE-CODE PROGRAMS - JEP 330

- Objectif : réduire la "cérémonie" (parfois inutile juste pour un simple test)
- Possibilité d'utiliser `java` sans passer par `javac` (compilation en mémoire) : `java MaClasse.java`
- Scripts avec le shebang : `#!/usr/bin/java : ./ma-classe` (sans extension)

NOUVEAUTÉS STRING

- `String::repeat`
- `String::strip`
- `String::stripLeading`
- `String::stripTrailing`
- `String::isBlank`
- `String::lines`
- **Démo !**

NOUVEAUTÉS DIVERSES

- `CharSequence::compareTo()`
- `StringBuilder::compareTo()`
- `StringBuffer::compareTo()`
- `Reader::nullReader()` et
`Writer::nullWriter()`
- `InputStream::readNBytes(int len)`
- `Pattern::asMatchPredicate`

NOUVEAUTÉS DIVERSES

- `Optional::isEmpty()`
- `Files::readString()` et `Files::writeString()`
- `Predicate::not()`
- `TimeUnit::convert(Duration)`
- `Collection::toArray()`

FLIGHT RECORDER - JEP 328

- Produire et consommer des événements (classe `Event` à étendre) pour la résolution d'incidents
- Pour démarrer l'enregistrement : `java -XX:StartFlightRecording` ou via la commande `jcmd`

PERFORMANCES

- Optimisations sur le G1
- ZGC - JEP 333 (expérimental, low-latency, big data)
- Epsilon GC - JEP 318

SUPPORTS DIVERS

- Unicode 10.0 - JEP 327
- Transport Layer Security (TLS) 1.3 - JEP 332

RETRAITS

- Java FX (openjfx.io), Java EE (dont JAX-WS et JAXB) et CORBA retirés
- Moteur JavaScript Nashorn introduit avec Java 8 (pour remplacer Rhino) : deprecated
- Oracle ne fournit plus de JRE (donc plus de mises à jour automatiques)
- `java.lang.Thread.destroy()` et `java.lang.Thread.stop(java.lang.Throwable)`

Java SE 12

Mars 2019



<https://openjdk.java.net/projects/jdk/12/>

EXTENDED SWITCH STATEMENT/EXPRESSION - JEP 325

- `case` à valeurs multiples : `case MONDAY, FRIDAY, SUNDAY -> System.out.println(6);`
- Affectation du résultat du `switch` dans une variable (`break` peut retourner une valeur)
- **Démo !**

NOUVEAUTÉS DIVERSES

- `Collectors::teeing`
- `String::align`, `String::indent` et `String::transform`
- `Files::isSameFile` et `Files::mismatch`
- `java.text.CompactNumberFormat` : norme LDML avec 1000 -> 1K, etc
- Améliorations pour l'API `CompletableFuture` (nouvelles méthodes)

PERFORMANCES

- Class-data sharing (CDS) par défaut
- Abortable Mixed Collections for G1 - JEP 344
- Promptly Return Unused Committed Memory from G1 - JEP 346
- Shenandoah: A Low-Pause-Time Garbage Collector - JEP 189 (expérimental, big data)

Java SE 13, 14...

Septembre 2019, Mars 2020...



[https://openjdk.java.net/projects/jdk/13/ ...](https://openjdk.java.net/projects/jdk/13/)

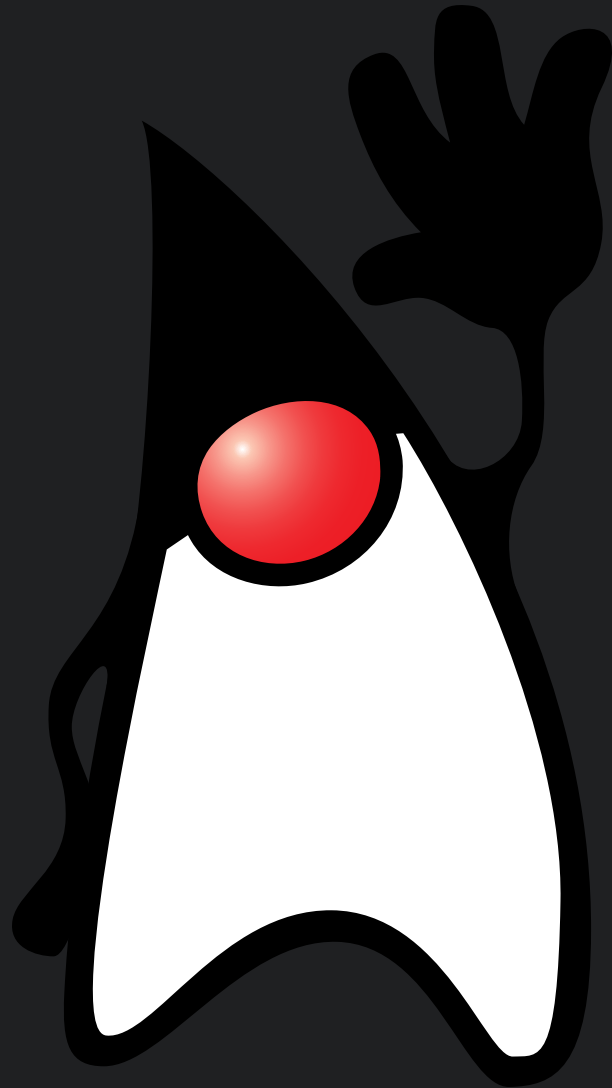
EN COURS OU À VENIR

- Raw String Literals - JEP 326 : ``chaîne sur plusieurs lignes``
- Pattern Matching for `instanceof` - JEP 305
- Packaging Tool (`jpackage`) - JEP 343 : `.msi`, `exe`, `deb`, `rpm`...
- Collection Literals - JEP 186 : `var list = #[1, 2, 3];`
- Foreign function interfaces (FFI) - JEP 191
- Détails dans `NullPointerException`

EN COURS OU À VENIR

- Continuation & Fibers : alternative légère au `Thread`
- Concise Method Bodies : `int length(String s) -> s.length();`
- Execution de méthodes Java sur du GPU (Stream API) : machine/deep learning
- Incubator Modules - JEP 11
- Preview Language and VM Features - JEP 12 : `--enable-preview`

Pour conclure !



POURQUOI MIGRER ?

- Java 8 en fin de vie depuis janvier 2019
- Performances largement améliorées
- Mises à jour de sécurité
- Nouvelles fonctionnalités (modules, `jshell`, `var`, ...)
- Support intégral des containers (Docker, Openshift)
- Vers plus de programmation fonctionnelle, réactive...

COMMENT MIGRER ?

- Se former, se renseigner
- Échanges SecDevOps (impacts environnements ?)
- Convaincre son équipe, son client

-

<https://docs.oracle.com/en/java/javase/11/migrate>

- <https://blog.codefx.org/java/java-11-migration-guide/>

COMMENT MIGRER ?

- Télécharger OpenJDK 11 (ou une autre)
- Mettre à jour l'IDE : IntelliJ IDEA \geq 2018.3 ou Eclipse \geq 2018-12
- Mettre à jour Maven (\geq 3.5.0) et Gradle (\geq 5.0) et leurs plugins (maven-compiler-plugin \geq 3.8.0)
- `rt.jar`, `tools.jar` et `dt.jar` retirés dans Java SE 9 (JEP 220)
- Mettre à jour les librairies (ajouter si besoin `javax.xml.bind:jaxb-api:2.3.1`)

SOURCES

- <https://openjdk.java.net>
- <https://medium.com/@javachampions>
- <https://www.loicmathieu.fr>
- <https://www.baeldung.com>
- <https://dzone.com>

Merci !

