

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h1>Installer l'environnement de développement java</h1> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

# Installer l'environnement de développement Java

Sous Windows

|   |          |
|---|----------|
| <b>1. PROLEGOMENES</b>  | <b>3</b> |
| 1.1. HERITAGE   | 3        |
| 1.2. COMPILATIONS D'UN PROGRAMME                                    | 4        |
| 1.2.1. <i>Compilation classique d'un programme</i>                  | 4        |
| 1.2.1.1. La compilation classique                                   | 4        |
| 1.2.1.2. La compilation avec l'exemple du C                         | 5        |
| 1.2.2. <i>Java est un langage compilé et interprété</i>             | 6        |
| 1.2.2.1. Compilation et interprétation du langage Java              | 6        |
| 1.2.2.2. Audience du langage Java                                   | 9        |
| 1.3. LES CONCEPTS DE JVM, JRE ET JDK                                | 9        |
| 1.3.1. <i>JVM</i>   | 10       |
| 1.3.1.1. Le principe de machine virtuelle                           | 10       |
| 1.3.1.2. Fonctionnement de la JVM                                   | 10       |
| 1.3.1.3. Les fonctions principales de la JVM                        | 11       |
| 1.3.2. <i>JRE</i>   | 11       |
| 1.3.2.1. Présentation du JRE  | 11       |
| 1.3.2.2. JRE versus JDK   | 11       |
| 1.3.3. <i>JDK</i>   | 11       |
| 1.3.3.1. Définition du JDK  | 12       |
| 1.3.3.2. Classification de JDK JRE JVM                              | 13       |
| 1.4. LE CHOIX DU JDK  | 13       |
| 1.4.1. <i>Rappel du contexte historique de Java</i>                 | 13       |
| 1.4.1.1. Timeline de Java   | 13       |
| 1.4.1.2. Les fournisseurs pour OpenJDK                              | 15       |
| 1.5. ELEMENTS TECHNIQUES  | 15       |
| 1.5.1. <i>Architectures x86 32 bits et x64 64 bits</i>              | 16       |
| 1.5.2. <i>Architectures de processeurs</i>                          | 17       |
| 1.5.2.1. L'architecture de Von Neumann                              | 17       |
| 1.5.2.2. Caractéristiques d'un microprocesseur                      | 17       |
| 1.5.2.2.1. Jeu d'instructions                                       | 17       |
| 1.5.2.2.2. Longueur de mot  | 18       |
| 1.5.2.2.3. Vitesse d'horloge système                                | 18       |
| 1.5.3. <i>Principales types d'architectures de processeur</i>       | 18       |
| 1.5.3.1. L'architecture CISC  | 18       |
| 1.5.3.1.1. Caractéristiques de l'architecture CISC                  | 18       |
| 1.5.3.2. L'architecture RISC  | 19       |
| 1.5.3.2.1. Caractéristiques de l'architecture RISC                  | 19       |
| 1.5.3.2.2. Fournisseurs de JDK par type d'architecture pour Windows | 20       |
| 1.6. PREREQUIS HARDWARE   | 21       |
| 1.6.1. <i>Les types de configurations de travail</i>                | 21       |
| 1.6.1.1. Configuration minimale                                     | 21       |
| 1.6.1.2. Configuration normale                                      | 22       |
| 1.6.1.3. Configuration avancée                                      | 22       |
| 1.6.1.3.1. Le goulot d'étranglement de Von Neumann                  | 22       |
| 1.6.1.3.2. Les technique pour remédier au Von Neumann bottleneck    | 23       |

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

|   |           |
|---|-----------|
| 1.6.1.3.2.1. La mémoire cache   | 23        |
| 1.6.1.3.2.2. Installer un type de mémoire adapté  | 24        |
| 1.6.1.3.3. Les autres techniques plus globales pour améliorer les performances du système     | 25        |
| 1.6.1.3.3.1. Augmenter la taille du fichier d'échange   | 25        |
| 1.6.1.3.3.2. Arrêt des services inutilisés  | 28        |
| 1.6.1.4. Configuration professionnelle  | 34        |
| 1.6.1.4.1. Le multithreading  | 34        |
| 1.6.1.4.2. L'hyper-threading  | 35        |
| 1.6.1.4.2.1. Vérifier si l'hyper-threading est activé   | 36        |
| 1.6.1.4.3. Les outils de profilage pour Java  | 37        |
| 1.6.1.4.3.1. Finalité d'un outil de profilage   | 37        |
| 1.6.1.4.3.2. Quelques outils de profilage pour Windows  | 38        |
| 1.6.2. Connaître le type d'architecture installée sur la machine                              | 40        |
| 1.6.2.1. Utiliser PowerShell  | 40        |
| 1.6.2.2. Utiliser le gestionnaire des tâches  | 41        |
| 1.6.2.3. Utiliser l'explorateur de fichiers   | 42        |
| 1.6.2.3.1. Program Files(x86) versus Program Files  | 42        |
| 1.6.2.3.2. System32 versus SysWOW64   | 43        |
| <b>2. INSTALLATION DE L'ENVIRONNEMENT DE DEVELOPPEMENT JAVA</b>                               | <b>44</b> |
| 2.1. PROCEDURE D'INSTALLATION DE JAVA   | 45        |
| 2.1.1. Téléchargement du JDK  | 45        |
| 2.1.1.1. Vérifier l'intégrité du fichier téléchargé   | 46        |
| 2.1.2. Installation de Java   | 48        |
| 2.1.2.1. Procédure d'installation de Java avec l'installateur automatique                     | 48        |
| 2.1.2.2. Configuration des variables d'environnement Java                                     | 53        |
| 2.1.2.2.1. Configurer la variable d'environnement Java via les propriétés avancées du système | 53        |
| 2.1.2.2.2. Configurer la variable d'environnement Java à l'aide d'un fichier batch            | 57        |
| 2.1.2.3. Vérification de l'installation du JDK  | 57        |
| 2.1.2.3.1. Vérifier la version de Java installée  | 58        |
| 2.1.2.3.2. Afficher le dossier d'installation du JDK  | 58        |
| 2.1.2.3.3. Vérifier la version du compilateur Java  | 59        |
| 2.1.2.4. Vérification avec un programme Java en ligne de commande                             | 59        |
| 2.1.2.4.1. Code de la classe HelloWorld   | 59        |
| 2.2. INSTALLATION DES OUTILS DE DEVELOPPEMENT   | 61        |
| 2.2.1. Les éditeurs de code   | 62        |
| 2.2.1.1. Les principaux éditeurs de code  | 62        |
| 2.2.1.1.1. Notepad++  | 63        |
| 2.2.1.1.2. Atom   | 63        |
| 2.2.1.1.3. Sublime Text   | 63        |
| 2.2.1.1.4. Bluefish   | 63        |
| 2.2.1.1.5. UltraEdit  | 63        |
| 2.2.1.2. Visual Studio code et VSCodium   | 64        |
| 2.2.1.2.1. Visual Studio Code   | 64        |
| 2.2.1.2.2. VSCodium   | 64        |
| 2.2.2. Les IDE  | 64        |
| 2.3. INSTALLATION DE L'IDE  | 64        |
| 2.3.1. Les principaux IDE pour coder en Java  | 64        |
| 2.3.2. Installation de l'IDE Eclipse  | 64        |
| 2.3.2.1. Téléchargement de l'IDE  | 64        |
| 2.3.2.2. Téléchargement de l'IDE  | 64        |
| 2.3.2.3. Configuration de l'IDE   | 64        |

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h1>Installer l'environnement de développement java</h1> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

|   |           |
|---|-----------|
| <b>3. TESTS</b>   | <b>67</b> |
| 3.1. BYTECODE   | 67        |
| 3.2. PROGRAMME DE TEST  | 67        |
| 3.3. JUNIT  | 67        |
| <b>4. DE JAVA 8 A JAVA 18 : 8 ANNEES DE NOUVELLES FONCTIONNALITES</b> | <b>67</b> |
| 4.1.1. Release table  | 67        |
| <b>5. CRITIQUES DE LA TECHNOLOGIE JAVA</b>                            | <b>69</b> |
| <b>6. ICONES</b>  | <b>69</b> |
| <b>7. LINKS</b>   | <b>70</b> |

## 1. Prolégomènes

Avant d'attaquer les modalités d'installation de l'environnement de développement java sous Windows, il convient de rappeler le contexte global de la technologie java.

La technologie Java définit à la fois un langage de programmation orienté objet et une plateforme informatique.

La technologie Java a été développée, en 1995, par James A. Gosling chez Sun Microsystems<sup>1</sup> ; cette entreprise a été rachetée par Oracle en 2009. Java est devenu donc propriété d'Oracle Corporation.

### 1.1. Héritage

Java est un langage de programmation multiplateformes<sup>2</sup>, il dérive des langages C et C++, dont il reprend, en partie, la syntaxe.

<sup>1</sup> [https://fr.wikipedia.org/wiki/Sun\\_Microsystems](https://fr.wikipedia.org/wiki/Sun_Microsystems)

<sup>2</sup> Logiciels(OS) ou matériels(x86,ARM).

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <b>Installer l'environnement de développement java</b> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



Figure 1 Timeline 3 langages

## 1.2. Compilations d'un programme

Une des particularités principales qui différencie le langage Java des autres langages comme le C ou le C++ est la manière dont il est exécuté et compilé sur une machine.

En effet, un programme C / C++, compilé, binaire, ne fonctionne que sur la plateforme pour laquelle il a été compilé.

Pour comprendre comment fonctionne Java, il est utile de rappeler la compilation classique, avec l'exemple du C.

### 1.2.1. Compilation classique d'un programme

#### 1.2.1.1. La compilation classique

La compilation classique suit ce schéma :

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h1>Installer l'environnement de développement java</h1> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

**Un programme compilé, binaire, ne fonctionne que sur la plateforme pour laquelle il a été compilé**



Figure 2 Compilation et exécutable

### 1.2.1.2. La compilation avec l'exemple du C

Voici un schéma qui synthétise la compilation en C :

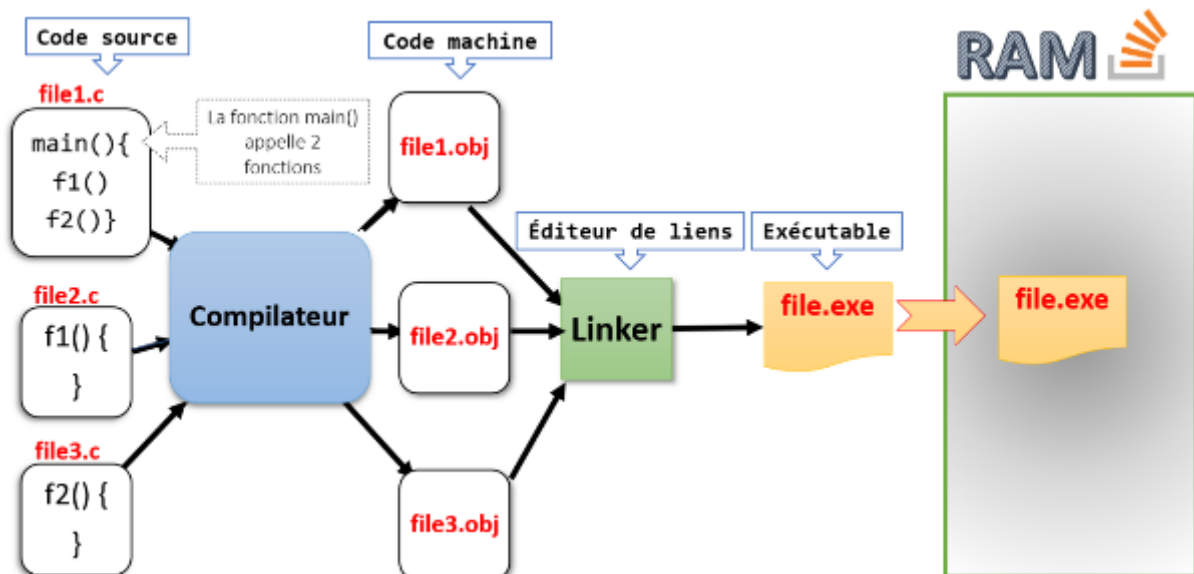


Figure 3 La compilation en C

Quant au compilateur Java, nommé **javac**, il ne traduit pas **directement** le code source, du fichier **.java**, en langage machine comme les compilateurs de C ou C++.

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <b>Installer l'environnement de développement java</b> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

Il le traduit en un langage intermédiaire appelé **bytecode** représenté par les fichiers **.class**.

Ce bytecode est ensuite interprété par un autre programme : la machine virtuelle java ou **JVM** (pour Java Virtual Machine).

Donc, un programme java contiendra, toujours, deux types de fichiers : les fichiers sources en java (extension. java) et le résultat de leur compilation en byte code (fichier d'extension .class).

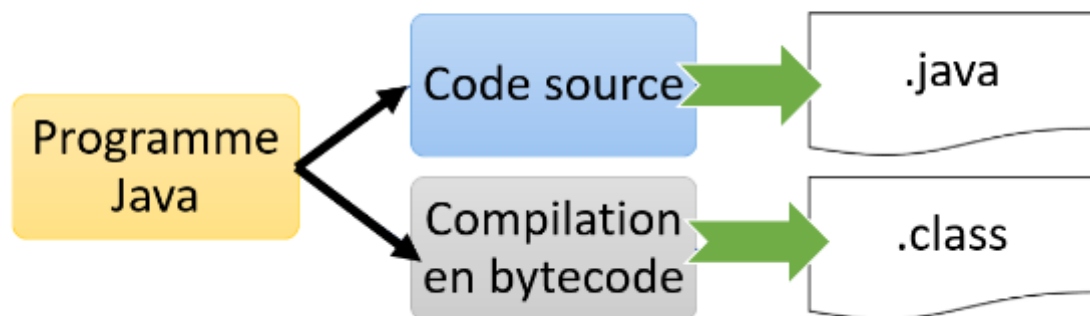


Figure 4 Composition d'un programme Java

## 1.2.2. Java est un langage compilé et interprété

### 1.2.2.1. Compilation et interprétation du langage Java

**Java peut être considéré à la fois comme un langage compilé et interprété car** son code source est d'abord compilé en un bytecode binaire. Ce bytecode s'exécute sur la machine virtuelle Java (JVM), qui est un interpréteur et/ou un compilation juste-à-temps (*just-in-time compilation* ou *JIT compilation*)<sup>3</sup>.

De plus, ce bytecode donne à Java sa portabilité : il fonctionnera sur n'importe quelle JVM correctement implémentée, quelle que soit la configuration matérielle ou logicielle de l'ordinateur.

Voici un schéma illustrant ce mécanisme :

<sup>3</sup> <https://www.ibm.com/docs/fr/sdk-java-technology/8?topic=reference-jit-compiler>

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

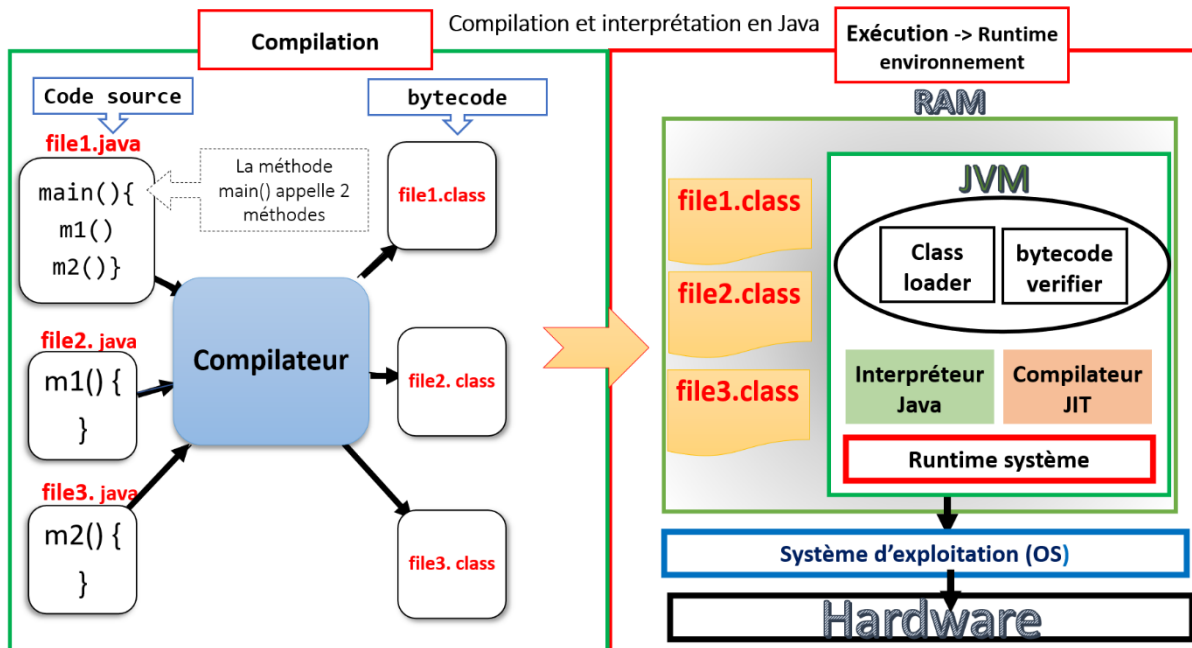


Figure 5 Compilation et interprétation en Java

La JVM est dépendante de la plate-forme, c'est-à-dire que sa mise en œuvre diffère d'une plate-forme à l'autre (Windows, Linux, Mac, etc...). Il en va de même pour le Java Runtime Environment (JRE) et le Java Development Kit (JDK).

Donc JVM, JRE et JDK sont dépendants de chaque plateforme, en raison du type de système d'exploitation de la machine et/ou de processeur sous-jacent.

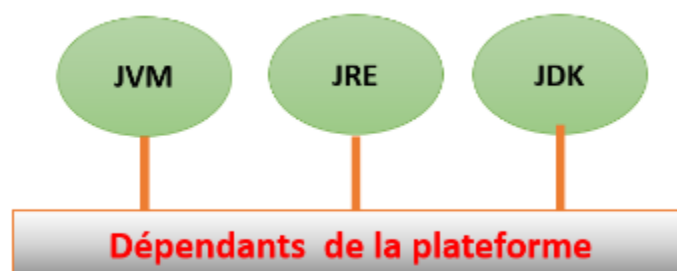


Figure 6 Dépendance de la plateforme

Mais toutes les JVM peuvent exécuter le même bytecode java. Car Java est indépendant de la plateforme.

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h1>Installer l'environnement de développement java</h1> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



Figure 7 Indépendance de la plateforme de Java

C'est l'approche « write once and run anywhere »<sup>4</sup>, en français « écrire une fois et exécuter partout ».

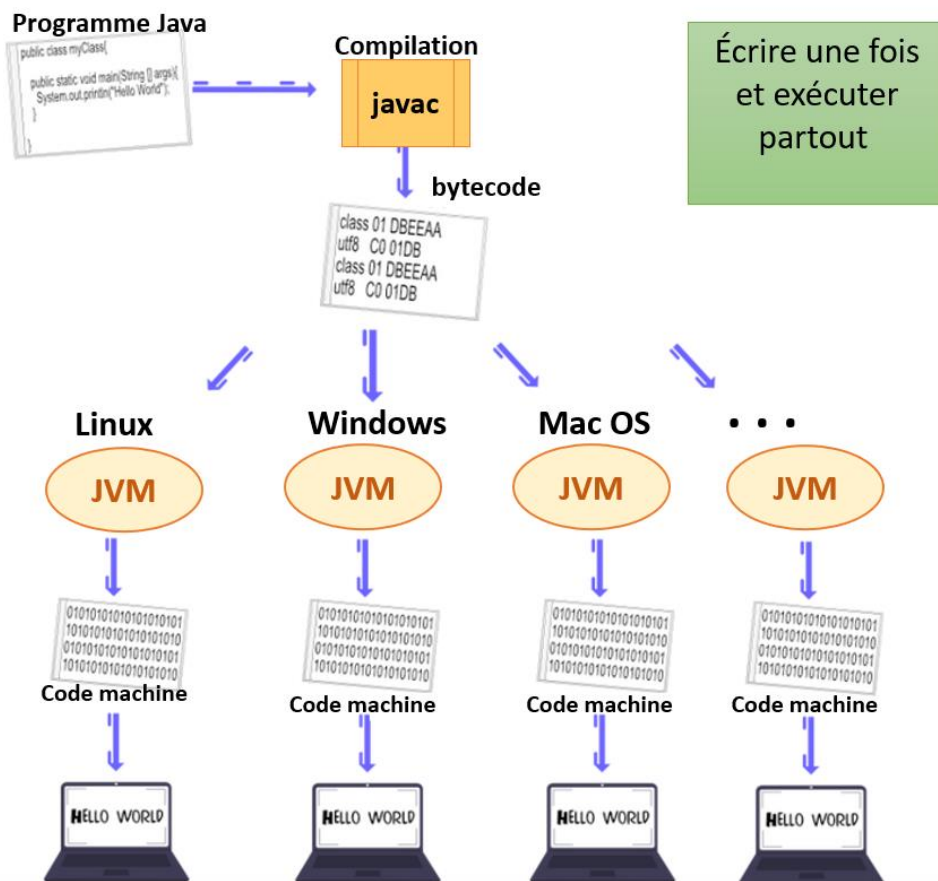


Figure 8 write once and run anywhere (WORA)

Cette indépendance de la plate-forme est l'une des caractéristiques qui ont fait de Java l'une des plates-formes de programmation les plus utilisées.

<sup>4</sup> En abrégé : WOA



|  |  |              |                  |
|--|--|--------------|------------------|
| <br>Nombre de pages :<br>71 | <b>Installer l'environnement de développement java</b> | Réalisé le : | 14-07-2022       |
|  |  | Modifié le : | 02-10-2022 19:21 |

### 1.2.2.2. Audience du langage Java

En 2019, Java était l'un des langages de programmation les plus populaires, en particulier pour les applications Web client-serveur, avec **9 millions de développeurs**<sup>5</sup>.

En effet selon l'index TIOBE Java est classé troisième :

| Jul 2022 | Jul 2021 | Change | Programming Language   | Ratings | Change |
|----------|----------|--------|--|---------|--------|
| 1        | 3        | ▲      |  Python | 13.44%  | +2.48% |
| 2        | 1        | ▼      |  C      | 13.13%  | +1.50% |
| 3        | 2        | ▼      |  Java   | 11.59%  | +0.40% |
| 4        | 4        |        |  C++    | 10.00%  | +1.98% |
| 5        | 5        |        |  C#     | 5.65%   | +0.82% |

Tableau 1 Index Tiobe juillet 2022

6

Toujours selon cette même source, on constate que le langage Java a occupé la **première place** de 2002 à 2017 :

| Programming Language | 2022 | 2017 | 2012 | 2007 | 2002 | 1997 | 1992 | 1987 |
|----------------------|------|------|------|------|------|------|------|------|
| Python               | 1    | 5    | 8    | 7    | 12   | 28   | -    | -    |
| C                    | 2    | 2    | 2    | 2    | 2    | 1    | 1    | 1    |
| Java                 | 3    | 1    | 1    | 1    | 1    | 13   | -    | -    |
| C++                  | 4    | 3    | 3    | 3    | 3    | 2    | 2    | 5    |
| C#                   | 5    | 4    | 4    | 8    | 18   | -    | -    | -    |
| Visual Basic         | 6    | 15   | -    | -    | -    | -    | -    | -    |
| JavaScript           | 7    | 8    | 10   | 9    | 9    | 20   | -    | -    |
| Assembly language    | 8    | 10   | -    | -    | -    | -    | -    | -    |
| SQL                  | 9    | -    | -    | -    | 7    | -    | -    | -    |
| PHP                  | 10   | 7    | 6    | 5    | 6    | -    | -    | -    |

Tableau 2 Index Tiobe sur 10 ans

Précisons les concepts sur lesquels repose la technologie Java.

## 1.3. Les concepts de JVM, JRE et JDK

En effet, il convient de présenter les notions de JRE, JVM et JDK, que tout développeur Java se doit de connaître.

<sup>5</sup> [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))

<sup>6</sup> <https://www.tiobe.com/tiobe-index/>

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

### 1.3.1. JVM

Avant de détailler la JVM, il y a lieu de présenter le concept de machine virtuelle, dans le cas général.

#### 1.3.1.1. Le principe de machine virtuelle

Prenons la définition de Wikipédia :



En informatique, une machine virtuelle<sup>7</sup> est une illusion d'un appareil informatique créée par un logiciel d'émulation. Le logiciel d'émulation simule la présence de ressources matérielles et logicielles telles que la mémoire, le processeur, le disque dur, voire le système d'exploitation et les pilotes, permettant d'exécuter des programmes dans les mêmes conditions que celles de la machine simulée.

Un des intérêts des machines virtuelles est de pouvoir s'abstraire des caractéristiques de la machine physique utilisée (matérielles et logicielles — notamment système d'exploitation), permettant une forte portabilité des logiciels<sup>8</sup>

**L'usage de machines virtuelles est l'un des principes fondamentaux de la technologie Java.**

Dans le cas particulier de Java la machine virtuelle prend le nom de : Java Virtual Machine (JVM).  
Donc, la JVM est une machine **abstraite** c'est une machine virtuelle, elle n'existe pas physiquement.

#### 1.3.1.2. Fonctionnement de la JVM

Il existe donc, une étape intermédiaire entre le code l'interprété et celui qui est compilé: la JVM.

- La JVM est un programme, qui permet d'isoler l'application qu'il doit faire tourner, du matériel et même du système d'exploitation.
- Le programme n'a aucun accès aux spécificités du matériel, l'ensemble de ses besoins lui étant fourni par la JVM.
- Ainsi, tout programme conçu pour cette machine virtuelle Java pourra fonctionner sur n'importe quel système d'exploitation (OS<sup>9</sup>), du moment que la dite machine virtuelle Java existe pour cet OS en question.
- Le programme fonctionnant depuis la JVM a déjà subi une première phase de compilation pour le transformer non pas en langage machine propre à l'ordinateur, mais dans un langage "machine virtuelle": le **bytecode**.
  - Ensuite la machine virtuelle compile ce bytecode à la volée juste au moment de son utilisation (technologie JIT, Just in Time).

<sup>7</sup> VM en abrégé.

<sup>8</sup> [https://fr.wikipedia.org/wiki/Machine\\_virtuelle](https://fr.wikipedia.org/wiki/Machine_virtuelle)

<sup>9</sup> Operating System

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

- La JVM sert, donc, à exécuter du code managé : code dans un langage intermédiaire : le bytecode.
- **La machine virtuelle ne connaît pas le langage Java : elle ne connaît que le bytecode qui est issu de la compilation de codes sources écrits en Java.**

### 1.3.1.3. Les fonctions principales de la JVM

Donc, pour résumer, la JVM :

- Charge le code
- Vérifie le code
- Interprète ce code
- Compile ce bytecode à la volée
- Exécute le code
- Fournit l'environnement d'exécution

## 1.3.2. JRE

### 1.3.2.1. Présentation du JRE

Le Java Runtime Environment, c'est l'**implémentation** de JVM<sup>10</sup>. Donc le JRE **dépend** du type de la plateforme.

Il est utilisé pour fournir l'**environnement d'exécution des programmes Java**. Il se compose d'un ensemble de bibliothèques et d'autres fichiers Java nécessaires à la JVM lors de l'exécution.

Il existe de nombreuses implémentations de JVM<sup>11</sup>, celles-ci sont open source :

- HotSpot, la principale implémentation de référence de Java VM
- Eclipse OpenJ9 d'IBM J9, pour Windows, Linux, macOS.

### 1.3.2.2. JRE versus JDK

- Le JRE est nécessaire, pour **tous type d'utilisateur**, pour faire tourner les applications Java ; tandis que le JDK est nécessaire, **au développeur**, pour coder des applications Java.
- Le JRE ne contient aucun outil de développement, contrairement au JDK.
- Les JRE sont disponibles pour une gamme de plates-formes beaucoup plus large que celle de JDK.
- JRE et JDK sont donc, **tous les deux dépendants** de la plateforme.

## 1.3.3. JDK

Voyons la définition du JDK, ainsi que ses différents fournisseurs.

<sup>10</sup> Il contient donc la JVM.

<sup>11</sup> [https://en.wikipedia.org/wiki/List\\_of\\_Java\\_virtual\\_machines](https://en.wikipedia.org/wiki/List_of_Java_virtual_machines)

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

### 1.3.3.1. Définition du JDK

Prenons la définition de Wikipédia :



Le Java Development Kit (JDK) désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être **compilé**, transformé en bytecode destiné à la machine virtuelle Java.

Il existe plusieurs éditions de JDK, selon la plate-forme Java considérée (et bien évidemment la version de Java ciblée) :

JSE pour la Java Standard Edition ;

JEE, sigle de Java Enterprise Edition ;

JME 'Micro Edition', destinée au marché mobiles.<sup>12</sup>

Le JDK contient donc des outils permettant de développer des applications Java.

Ces principaux composants sont :

- ✓ **JRE** : environnement d'exécution des programmes Java, qui contient la **JVM**.
- ✓ Le compilateur : **`javac`**
- ✓ L'interpréteur/chargeur : **`java`**
- ✓ Le débogueur : **`jdb`**
- ✓ Le désassembleur des fichiers .class : **`javap`**
- ✓ L'archiveur : **`jar`**
- ✓ Le générateur de documentation : **`javadoc`**

On peut synthétiser ces trois notions : JVM, JRE et JDK à l'aide d'un schéma.

<sup>12</sup> [https://fr.wikipedia.org/wiki/Java\\_Development\\_Kit](https://fr.wikipedia.org/wiki/Java_Development_Kit)

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <b>Installer l'environnement de développement java</b> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

### 1.3.3.2. Classification de JDK JRE JVM



Figure 9 Classification de JDK JRE JVM



Il est possible d'installer plusieurs versions du JDK sur la même machine<sup>13</sup>.  
 Dans le cas où l'on désire tester les performances de chaque édition du JDK, par exemple.

## 1.4. Le choix du JDK

Tous les JDK ne sont pas open source. Pour bien comprendre cette situation, il y a lieu de rappeler l'historique de Java, pour ensuite présenter les principales distributions des JDK, disponibles actuellement.

### 1.4.1. Rappel du contexte historique de Java

Les principales dates qui ont marquées l'histoire de la technologie Java.

#### 1.4.1.1. Timeline de Java

- 1982 Fondation de Sun Microsystems<sup>14</sup>, en Californie, par Andy Bechtolsheim, Bill Joy, Vinod Khosla et Scott McNealy.
- 1995 Lancement du langage Java, développé par James Gosling et Patrick Naughton au sein de Sun Microsystems.

<sup>13</sup> Si celle-ci dispose d'assez de ressources !

<sup>14</sup> [https://fr.wikipedia.org/wiki/Sun\\_Microsystems](https://fr.wikipedia.org/wiki/Sun_Microsystems)

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

- 1996 JDK 1.0
- 1997 JDK 1.1
- 1998 J2SE<sup>15</sup> 1.2
- 2000 J2SE 1.3
- 2002 J2SE 1.4
- 2004 J2SE 5.0
- 2006 Java SE 6. Java devient open source. En effet Sun Microsystems publie le code source de Java sous la Licence publique générale GNU (GPL) .
- 2009 Sun Microsystems est rachetée<sup>16</sup> par Oracle qui devient donc le nouveau propriétaire de Java.
- 2011 Java SE 7 est présenté par Oracle avec deux variantes du JDK
  - OpenJDK qui est open source. Il est maintenu et développé par Oracle, et permet aux communautés de développeurs et aux entreprises<sup>17</sup> de participer à son développement.
  - Oracle JDK qui n'est pas open source.
    - Il est maintenu et développé par Oracle et est conforme aux spécifications OpenJDK.
    - Ce JDK d'Oracle fournit des extensions additionnels mais dont l'usage est payant.
    - Oracle JDK est publié sous le "Oracle Binary Code License Agreement".
- 2014 Java SE 8
- 2017 Java SE 9
- 2018-03 Java SE 10
- 2018-09 Java SE 11
- 2019-03 Java SE 12
- 2019-09 Java SE 13
- 2020-03 Java SE 14
- 2020-09 Java SE 15
- 2021-03 Java SE 16
- 2021-04 Microsoft<sup>18</sup> sort sa propre distribution OpenJDK.
- 2021-09 Java SE 17

Pour développer en Java on a le choix entre des JDK open source ou soumis à une licence commerciale.

<sup>15</sup> Java 2 Standard Edition

<sup>16</sup> Pour un montant de 7 400 000 000 \$.

<sup>17</sup> Red Hat, Azul Systems, IBM, Apple, SAP, Microsoft, ...

<sup>18</sup> <https://devblogs.microsoft.com/java/microsoft-deepens-its-investments-in-java/>

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

La version open source repose sur OpenJDK. En effet, OpenJDK est une implémentation officielle, gratuite et open source de la plate-forme Java SE, tel que défini par le Java Community Process<sup>19</sup>. et ce, depuis sa version 7. Il a été initialement publié en 2007 comme le résultat du développement que Sun Microsystems a commencé en 2006.

#### 1.4.1.2. Les fournisseurs pour OpenJDK

Il existe de nombreux d'éditeurs pour OpenJDK. Ces éditions ne sont pas toutes open source. On a retenu les principales distributions.

| Fournisseur                | Gratuite libre et open source | Site  |
|----------------------------|-------------------------------|---|
| AdoptOpenJDK <sup>20</sup> | <b>OUI</b>                    | <a href="https://adoptium.net/temurin/releases">https://adoptium.net/temurin/releases</a>   |
| Alibaba_Dragonwell         | <b>OUI</b>                    | <a href="https://dragonwell-jdk.io/#/index">https://dragonwell-jdk.io/#/index</a>   |
| Amazon Corretto            | <b>OUI</b>                    | <a href="https://aws.amazon.com/fr/corretto">https://aws.amazon.com/fr/corretto</a>   |
| Azul Zulu                  | <b>OUI</b>                    | <a href="https://www.azul.com/downloads">https://www.azul.com/downloads</a>   |
| IBM                        | <b>OUI</b>                    | <a href="https://www.ibm.com/support/pages/java-sdk-downloads-eclipse">https://www.ibm.com/support/pages/java-sdk-downloads-eclipse</a> |
| Microsoft                  | <b>OUI</b>                    | <a href="https://www.microsoft.com/openjdk">https://www.microsoft.com/openjdk</a>   |
| OpenJDK                    | <b>OUI</b>                    | <a href="https://openjdk.org">https://openjdk.org</a>   |
| Oracle JDK                 | <b>NON<sup>21</sup></b>       | <a href="https://www.oracle.com/java/technologies/downloads">https://www.oracle.com/java/technologies/downloads</a>                     |
| SapMachine                 | <b>OUI</b>                    | <a href="https://sap.github.io/SapMachine">https://sap.github.io/SapMachine</a>   |

Tableau 3 Principales distributions OpenJDK

On va choisir une version libre du JDK. Donc on optera pour une distribution de type openJDK, ou bien pour des fins de développement à une version d'Oracle.

Mais avant d'installer l'environnement de développement Java, il faut s'assurer de disposer du matériel adéquat.

## 1.5. Eléments techniques

Si l'on veut pouvoir faire de la conception et du développement d'applications, que ce soit dans n'importe quel langage, il faut disposer d'une machine en conséquence.

Pour bien comprendre l'enjeu, un bref rappel technique est nécessaire.

<sup>19</sup> [https://fr.wikipedia.org/wiki/Java\\_Community\\_Process](https://fr.wikipedia.org/wiki/Java_Community_Process)

<sup>20</sup> Devenu Adoptium

<sup>21</sup> Gratuit pour un usage personnel et non commercial

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <b>Installer l'environnement de développement java</b> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

### 1.5.1. Architectures x86 32 bits et x64 64 bits

Lorsque les processeurs 64 bits ont été introduits, ils étaient appelés x64 pour distinguer la nouvelle architecture 64 bits des anciennes gammes de processeurs 32 bits. Ainsi, on utilise les termes de : «x86» et «x64» pour désigner les architectures «32 bits» et «64 bits».

Aujourd'hui, les ordinateurs vendus dans le commerce sont équipés d'un processeur 64 bits et fonctionnent avec un système d'exploitation 64 bits.

Mais il existe encore des anciennes machines qui fonctionnent, encore avec un processeur 32 bits.

Il convient donc de connaître les deux types d'architectures, car cela a, évidemment, un impact sur le développement d'applications.

Les deux différences fondamentales entre un processeur 32 bits et 64 bits concernent : la taille des registres généraux du processeur et la quantité de mémoire supportée.

Ce tableau reprend les principales caractéristiques des deux systèmes.

|                                   | Système 32-bit   | Système 64-bit  |
|-----------------------------------|--|---|
| <b>Nombre de bits</b>             | 32   | 64  |
| <b>L'architecture du logiciel</b> | L'architecture 32 bits repose sur des registres, des adresses ou des <b>bus de données de 32 bits</b> (4 octets) de large. Pour les logiciels, 32 bits signifie généralement l'utilisation d'un espace d'adresse linéaire 32 bits. | L'architecture 64 bits est basée sur des registres, des adresses ou <b>des bus de données de 64 bits</b> (8 octets). Pour les logiciels, 64 bits signifie que le code est utilisé avec les adresses de mémoire virtuelle 64 bits. |
| <b>Compatibilité</b>              | Dans les systèmes d'exploitation de 32 bits, les applications requièrent des CPU <sup>22</sup> de 32 bits  | Le système d'exploitation de 64 bits nécessite un processeur de 64 bits et les applications 64 bits nécessitent un système d'exploitation et un processeur de 64 bits   |
| <b>Limites de mémoire</b>         | Les systèmes 32 bits sont limités à 3,2 gigaoctets (Go) de RAM <sup>23</sup> . Windows 32 bits avec une limitation d'adresse ne dépassant pas 4 Go. Cela dépend du matériel, généralement de 3,25 Go.                              | Les systèmes 64 bits permettent, théoriquement, jusqu'à 17 milliards de Go de RAM. <sup>24</sup>  |

Tableau 4 Comparaison systèmes 32 bits et 64 bits

<sup>22</sup> Central Processor Unit , autrement dit : processeur.

<sup>23</sup> Random Access Memory (La mémoire vive)

<sup>24</sup> D'après : <https://waytolearnx.com/2018/11/difference-entre-32-bits-et-64-bits.html>



|   |  |              |                  |
|---|--|--------------|------------------|
|  | <b>Installer l'environnement de développement java</b> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

### 1.5.2. Architectures de processeurs

Pour situer cette technologie de processeur par rapport aux deux, précédentes, il convient de faire un rappel sur les principales types d'architectures.

#### 1.5.2.1. L'architecture de Von Neumann

Tous les processeurs, du premier d'entre-deux le Intel 4004<sup>25</sup>, sorti en 1971, aux plus récents comme les puces ARM, tous obéissent à « la loi de Von Neumann ».

En effet, les architectures de processeurs découlent du modèle d'architecture d'ordinateur de Von Neumann<sup>26</sup>.



27

Figure 10 Le modèle original de von Neumann pour l'architecture des ordinateurs.<sup>28</sup>



Il existe une autre architecture, mais destinée à un usage plus restreint et spécialisé : **l'architecture de Harvard**<sup>29</sup>.

Actuellement, la plupart des ordinateurs sont des machines de Von Neumann.

#### 1.5.2.2. Caractéristiques d'un microprocesseur

Un microprocesseur est, fondamentalement, caractérisé par : son jeu d'instructions, la longueur de mot et la vitesse d'horloge.

##### 1.5.2.2.1. Jeu d'instructions

Le jeu d'instructions est l'ensemble des instructions-machine qu'un processeur d'ordinateur peut exécuter. Ces instructions-machines permettent d'effectuer des opérations élémentaires (addition, ET logique...) ou plus complexes (division, passage en mode basse consommation...)<sup>30</sup>.

<sup>25</sup> [https://fr.wikipedia.org/wiki/Intel\\_4004](https://fr.wikipedia.org/wiki/Intel_4004)

<sup>26</sup> [https://en.wikipedia.org/wiki/Von\\_Neumann\\_architecture](https://en.wikipedia.org/wiki/Von_Neumann_architecture)

<sup>27</sup> <https://interstices.info/wp-content/uploads/jalios/modele-neumann/modele-origine2.gif>

<sup>28</sup> <https://interstices.info/le-modele-darchitecture-de-von-neumann/>

<sup>29</sup> [https://fr.wikipedia.org/wiki/Architecture\\_de\\_type\\_Harvard](https://fr.wikipedia.org/wiki/Architecture_de_type_Harvard)

<sup>30</sup> [https://fr.wikipedia.org/wiki/Jeu\\_d%27instructions](https://fr.wikipedia.org/wiki/Jeu_d%27instructions)

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

#### 1.5.2.2.2. Longueur de mot

C'est le nombre de bits traités dans une seule instruction, par le processeur. La taille du mot est proportionnelle à la puissance de traitement du CPU. Les processeurs actuels ont une longueur de mot de 32 ou **64 bits**.

#### 1.5.2.2.3. Vitesse d'horloge système

Le rôle de l'horloge est de cadencer le rythme du travail du microprocesseur. La **fréquence** correspond à ce que l'on appelle un cycle d'horloge. Une instruction, selon le type de processeur et d'instruction peut prendre un ou plusieurs cycles d'horloge. A technologie égale, plus la fréquence est élevée, plus le nombre d'instruction pouvant être exécutée est élevée.<sup>31</sup>

Un microprocesseur repose sur une architecture.

### 1.5.3. Principales types d'architectures de processeur

Les deux catégories principales<sup>32</sup> d'architectures de processeur les plus connues et qui nous intéressent, ici pour le développement, sont:

1. CISC: Complex Instruction Set Computer = ordinateur à jeu d'instruction complexe
2. RISC: Reduced Instruction-Set Computer = *ordinateur à jeu d'instructions réduit*

#### 1.5.3.1. L'architecture CISC

##### 1.5.3.1.1. Caractéristiques de l'architecture CISC

- Un microprocesseur CISC, désigne un microprocesseur possédant un **jeu d'instructions**<sup>33</sup> comprenant de très nombreuses instructions<sup>34</sup>.
- **Les instructions interagissent avec la mémoire en utilisant des modes d'adressage complexes**
- Les processeurs CISC réduisent la taille du programme et donc moins de cycles de mémoire sont nécessaires pour exécuter les programmes.
- Une instruction est écrite en assembleur, donc plus proche d'un langage de programmation standard comme le C.
- Une instruction, de taille importante et variable, fonctionne en plusieurs étapes de bas niveau.

<sup>31</sup> Idem

<sup>32</sup> Les autres architectures sont : VLIW (Very long instruction word), vectorielle et dataflow.

<sup>33</sup> [https://fr.wikipedia.org/wiki/Jeu\\_d%27instructions](https://fr.wikipedia.org/wiki/Jeu_d%27instructions)

<sup>34</sup> [https://fr.wikipedia.org/wiki/Microprocesseur\\_%C3%A0\\_jeu\\_d%27instruction\\_%C3%A9tendu](https://fr.wikipedia.org/wiki/Microprocesseur_%C3%A0_jeu_d%27instruction_%C3%A9tendu)

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

L'architecture CISC se retrouve, notamment, dans les processeurs des deux principaux fabricants que sont Intel et AMD. Actuellement les processeurs les plus utilisés sont ceux des fabricants Intel et AMD.

Il est à noter que l'architecture ARM commence également à être utilisée dans les configurations d'ordinateurs.

### 1.5.3.2. L'architecture RISC

#### 1.5.3.2.1. Caractéristiques de l'architecture RISC

- Un processeur RISC est un type d'architecture de processeur qui se caractérise par des instructions de base aisées à décoder, uniquement composées **d'instructions simples**<sup>35</sup>.
- Ce sont des instructions, de longueur uniforme, qui sont exécutées en **un cycle d'horloge**.
- Conséquence : l'ordinateur doit effectuer à plusieurs reprises des opérations simples pour exécuter un programme.

Les principales implémentations de l'architecture RISC sont :

- SPARC, (Scalable Processor Architecture), est une architecture de processeur de type RISC, originellement développée par Sun Microsystems<sup>36</sup>.  
Les spécifications de SPARC sont entièrement libres.
- PowerPC est une architecture de processeur RISC POWER d'IBM, et développée conjointement par Apple, IBM et Motorola<sup>37</sup>.  
PowerPC est sous licence Open Source.
- ARM est une architecture de processeur RISC développée par l'entreprise ARM.<sup>38</sup>  
La première utilisation de cette technologie a été faite par Acorn, pour ses ordinateurs.

Ces architectures sont représentées dans le schéma suivant :

<sup>35</sup> [https://fr.wikipedia.org/wiki/Processeur\\_%C3%A0\\_jeu\\_d%27instructions\\_r%C3%A9duit](https://fr.wikipedia.org/wiki/Processeur_%C3%A0_jeu_d%27instructions_r%C3%A9duit)

<sup>36</sup> [https://fr.wikipedia.org/wiki/Architecture\\_SPARC](https://fr.wikipedia.org/wiki/Architecture_SPARC)

<sup>37</sup> <https://fr.wikipedia.org/wiki/PowerPC>

<sup>38</sup> [https://fr.wikipedia.org/wiki/ARM\\_\(entreprise\)](https://fr.wikipedia.org/wiki/ARM_(entreprise))

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <b>Installer l'environnement de développement java</b> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



Graphe 1 Principales architectures de processeur

En ayant à l'esprit ces différentes architectures, on est plus à même de choisir la bonne version d'un programme. En effet celui-ci doit être doublement compatible : avec l'architecture du processeur et avec le système d'exploitation supporté par l'ordinateur.

C'est ainsi que pour ce qui est de la technologie Java , elle est déclinée en plusieurs distributions, selon donc, le couple architecture /OS.

Mais dans ce cas particulier, on va présenter uniquement les distributions du JDK relatives à l'écosystème Windows.

#### 1.5.3.2.2. Fournisseurs de JDK par type d'architecture pour Windows

Les fournisseurs de JDK pour Windows par type d'architecture x86 x64 et ARM :

| Fournisseur de JDK pour Windows | Disponibilité du JDK par architecture |
|---------------------------------|---------------------------------------|
| AdoptOpenJDK - Adoptium         | x86; x64                              |
| Alibaba_Dragonwell              | x64                                   |
| Amazon Corretto                 | x64                                   |
| Azul Zulu                       | x86; x64; AArch64                     |
| IBM                             | x64                                   |

|  |  |              |                  |
|--|--|--------------|------------------|
| <br>Nombre de pages :<br>71 | <b>Installer l'environnement de développement java</b> | Réalisé le : | 14-07-2022       |
|  |  | Modifié le : | 02-10-2022 19:21 |

|                  |              |
|------------------|--------------|
| <b>Microsoft</b> | x64; AArch64 |
| OpenJDK          | x64          |
| Oracle JDK       | x64          |
| SapMachine       | x64          |

Tableau 5 Les fournisseurs de JDK pour Windows par type d'architecture

On constate que l'architecture RISC avec la distribution AArch64 est, librement, disponible, notamment, chez Microsoft. C'est la preuve que l'architecture ARM suscite un grand intérêt. Ce qui laisse présager une plus grande diffusion à l'avenir.

## 1.6. Prérequis hardware

Si l'on veut pouvoir faire de la conception et du développement d'applications, que ce soit dans n'importe quel langage, il faut disposer d'une machine en conséquence.

### 1.6.1. Les types de configurations de travail

En matière de développement, on peut considérer quatre, principales, configurations matérielles. La composition de ces environnements est résumée dans un tableau.

| Configurations                   |       | Minimale     | Normale     | Avancée             | Professionnelle       |
|----------------------------------|-------|--------------|-------------|---------------------|-----------------------|
| Type d'ordinateur                |       | Laptop       | Desktop     | Desktop             | Work station          |
| Ecran                            |       | 17"          | 2 X 21"     | 2 X 24"             | 2 X 27"               |
| Processeur                       | Intel | I3           | I5          | I7/ I9              | Xeon (x2)             |
|                                  | AMD   | Ryzen 3      | Ryzen 5     | Ryzen 7/<br>Ryzen 9 | Ryzen<br>Threadripper |
| RAM                              |       | 8 Go         | 16 Go       | 32 Go               | 32 Go                 |
| Stockage                         |       | 250 Go       | 500 Go      | 1 To                | 1 To                  |
| Système d'exploitation (64 bits) |       | Windows Home | Windows Pro | Windows Pro         | Windows Entreprise    |

Tableau 6 Configurations matérielles

Ces quatre configurations appellent quelques précisions.

#### 1.6.1.1. Configuration minimale

Il n'est pas impossible de développer avec un laptop. C'est l'idéal pour écrire de petits programmes, des maquettes ou pour faire des tests.

En revanche pour de longues heures de programmation, cela n'est pas très ergonomique. De plus le processeur d'un ordinateur portable chauffe rapidement, au bout de quelques heures d'utilisation, ce qui peut entraîner des plantages, apparemment, inexplicables.

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

De plus s'il y a beaucoup d'outils d'applications et d'outils de développement installés, le système va ralentir. Si plusieurs applications sont ouvertes simultanément, alors le système va utiliser beaucoup de mémoire virtuelle et va paginer, ce qui fait grossir le fichier d'échange<sup>39</sup>.

On va donc se trouver limité, en termes de performance ; à moins de passer à une configuration plus puissante.

### 1.6.1.2. Configuration normale

Ici il faut entendre « normale » au sens de norme ou de standard ou de bonnes pratiques.

En effet, c'est la constatation en situation réelle, qui amène à considérer cette configuration de normale. Car il faut une quantité suffisante de RAM et un processeur avec une bonne fréquence<sup>40</sup>, pour pouvoir bien travailler.

L'ergonomie est assurée avec, notamment, deux écrans, ce n'est pas un luxe si l'on veut avoir un œil sur le code et un autre sur l'IHM ou la Vue<sup>41</sup>.

Avec cette configuration on peut coder, convenablement, la plupart des applications classiques, qu'elles soient desktop, web ou autres.

Mais dans certaines situations, il peut arriver que le système ralentisse. C'est par exemple le cas si on développe une application qui nécessite de lancer, simultanément : l'IDE<sup>42</sup>, le serveur, la base de données, l'outil de gestion de versions du code source, le navigateur, une ou plusieurs applications bureautiques, la messagerie...etc. Dans cette situation la latence ne cesse d'augmenter.

Dans ce cas il peut être judicieux de passer à une configuration avancée.

### 1.6.1.3. Configuration avancée

Avec cette configuration ayant un processeur plus puissant et plus de RAM, on peut ouvrir plusieurs applications simultanément. Cela ne ralentira pas beaucoup, en principe, le système.

Normalement les performances de la machine ne seront pas affectées. En théorie donc, car malgré un processeur de haut de gamme et 32 Go de RAM, il peut se produire une certaine latence.

En effet, lorsque de la mémoire classique est utilisée avec un système disposant d'un processeur rapide ; il se produit un goulot d'étranglement affectant les performances du système.

#### 1.6.1.3.1. Le goulot d'étranglement de Von Neumann

En effet, dans cette situation, quand les charges de travail gérées par la machine atteignent un certain seuil, il se produit un ralentissement dans le mouvement des données entre le CPU et la RAM. C'est ce qu'on appelle le goulot d'étranglement de Von Neumann<sup>43</sup>.

<sup>39</sup> Ou fichier swap

<sup>40</sup> > à 2 GHz.

<sup>41</sup> Cela dépend évidemment du type d'application développée : desktop ou web.

<sup>42</sup> Environnement de développement intégré

<sup>43</sup> Ou Von Neumann bottleneck.

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



Figure 11 Von Neumann bottleneck

Ceci s'explique par le fait que la mémoire vive, où se trouvent les instructions et les données à traiter, est mutualisée dans le **même bus de données** avec un **adressage commun**. Conséquence : les instructions et les données doivent donc être saisies séquentiellement à partir de la RAM.

Comme le traitement par le processeur est plus rapide que celui de la RAM, il se produit un temps d'attente, avant que les données proviennent de la RAM au processeur. Donc le processeur peut rester, un certain temps, inactif. C'est ce qu'on appelle le « IDLE time ».

#### 1.6.1.3.2. Les technique pour remédier au Von Neumann bottleneck

Pour remédier au goulot d'étranglement de Von Neumann on peut appliquer les techniques les plus courantes, parmi celles-ci :

- La mémoire cache
- Installer un type de mémoire adapté

##### 1.6.1.3.2.1. La mémoire cache

La mémoire cache, ou antémémoire, est un type de mémoire beaucoup plus rapide que la RAM classique. On distingue trois types de mémoire cache : L1, L2 et L3 situées dans le processeur, sauf pour L3 qui peut se situer en dehors de ce dernier.

Donc pour ne pas ne pas limiter les performances du processeur, on lui a adjoint ces petites unités de mémoires, beaucoup plus rapides.

Evidemment dans cette configuration on choisira un processeur avec une mémoire cache, la plus importante possible, compte-tenu des contraintes budgétaires. Il faudra donc faire une optimisation sous contrainte.

Pour connaître la quantité de mémoire cache, il faut suivre ces 3 étapes:

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

1. Ouvrir une invite de commandes en mode administrateur
2. Taper la commande : `winsat features`
3. Et on obtient , entre autres, la quantité de mémoire cache : L1, L2 et L3 exprimée en octets.

Exemple :



```

Administrateur: C:\Windows\system32\cmd.exe
Microsoft Windows [version 10.0.19044.1889]
(c) Microsoft Corporation. Tous droits réservés.

C:\Windows\system32>winsat features
Outil d'évaluation du système Windows
> DWM running... leaving it on
> En cours d'exécution : Énumération de fonctions ''
> Gathering System Information
> Operating System                : 10.0 Build-19044
> Processor                      : Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz
> TSC Frequency                  : 0
> Number of Processors           : 1
> Number of Cores                 : 4
> Number of CPUs                 : 8
> Number of Cores per Processor  : 4
> Number of CPUs Per Core       : 2
> Cores have logical CPUs        : YES
> L1 Cache and line Size         : 32768 64
> L2 Cache and line Size         : 262144 64
> L3 Cache and line Size         : 6291456 64

```

Screenshot 1 Quantité de mémoire cache

#### 1.6.1.3.2.2. Installer un type de mémoire adapté

En effet, si on désire diminuer le goulot d'étranglement entre le processeur et la RAM, il faut installer de la mémoire de type DDR5 SDRAM (Double Data Rate 5 Synchronous Dynamic Random-Access Memory).

En effet, la mémoire à double canal (DDR) permet au processeur d'échanger des données avec la RAM par le biais de deux canaux, en lisant et en écrivant simultanément sur deux barrettes mémoire. **Cela augmente la bande passante disponible.**



Si on veut aller plus loin dans l'optimisation des performances du système ; il faut savoir que comme le CPU ou le GPU (Graphics Processing Unit), la RAM peut, également, être overclocker. En effet, il est possible d'augmenter la fréquence de celle-ci ainsi que de diminuer son temps de latence<sup>44</sup>.

<sup>44</sup> <https://www.intel.fr/content/www/fr/fr/gaming/resources/overclock-ram.html>



|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

1.6.1.3.3. Les autres techniques plus globales pour améliorer les performances du système  
On en mentionnera deux : la taille du fichier d'échange et les services.

#### 1.6.1.3.3.1. Augmenter la taille du fichier d'échange

Lorsque le système d'exploitation manque de mémoire vive , il en vient à utiliser une partie de l'espace libre du disque dur pour y stocker, temporairement, des données : c'est le fichier d'échange<sup>45</sup>. Cette technique est également désignée sous le terme de : **mémoire virtuelle**.

Le fichier d'échange pagefile.sys système d'exploitation se trouve à la racine du lecteur C :

C:\pagefile.sys

Pour augmenter la taille du fichier d'échange, il faut suivre ces étapes:

1. Ouvrir une fenêtre « Exécuter » Windows<sup>46</sup>
2. Taper la commande : `systempropertiesperformance`



- 3.
4. Dans la fenêtre qui s'ouvre, cliquez sur l'onglet Avancé.

<sup>45</sup> Ou fichier swap.

<sup>46</sup> Touches Windows + R

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



5. Cliquer sur le bouton **Modifier**.
6. Dans la fenêtre « Mémoire virtuelle », il faut **décocher** la case « Gestion automatique du fichier d'échange pour les lecteurs ».
7. Et **activer** le bouton radio « Taille personnalisée »

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



**Mémoire virtuelle**

☐ Gestion automatique du fichier d'échange pour les lecteurs

Taille du fichier d'échange pour chaque lecteur

| Lecteur [nom de volume] | Taille du fichier d'échange (Mo) |
|-------------------------|----------------------------------|
| C: [Windows]            | 4155 - 6704                      |
| D: [DEV]                | Aucun                            |
| E: [RECOVERY]           | Aucun                            |

Lecteur sélectionné : C: [Windows]  
Espace disponible : 90790 Mo

☒ Taille personnalisée :

Taille initiale (Mo) :

Taille maximale (Mo) :

☐ Taille gérée par le système  
☐ Aucun fichier d'échange

Taille totale du fichier d'échange pour tous les lecteurs

Minimale autorisée : 16 Mo  
Recommandée : 2363 Mo  
Allouée actuellement : 4155 Mo

8. Indiquer (en Mo) pour « Taille initiale » une capacité d'au moins celle mentionnée en « Recommandée », ou avec une taille initiale au moins égale à celle « Allouée actuellement ».
9. Indiquer une « Taille maximale » environ double de la « Taille initiale ».<sup>47</sup>

<sup>47</sup> Et compatible avec l'espace disque disponible.

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



10. Cliquer sur le bouton **Définir** pour confirmer les nouvelles valeurs.
11. Cliquer sur le bouton **OK**.
12. Cliquer encore sur le bouton **OK** de la fenêtre précédente.
13. Enfin, redémarrer l'ordinateur pour que les changements deviennent effectifs.

#### 1.6.1.3.3.2. Arrêt des services inutilisés

Les services sont des programmes configurés pour démarrer en même temps que Windows. Ils fonctionnent ensuite en arrière-plan tant que l'ordinateur est allumé.

Afin d'optimiser Windows Il est judicieux de désactiver les services inutiles. Ceci pour gagner en mémoire disponible et récupérer de précieuses ressources système.

Avant de désactiver un service inutile, il convient, au préalable, de lister ces services. Ceci peut être effectué avec PowerShell.

Pour lancer PowerShell : on peut ouvrir une fenêtre « Exécuter » avec la commande **powershell** :

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



Commande 1: commande PowerShell



Screenshot 2: fenêtre Powershell

Utiliser la commande PowerShell :

### **Get-Service**

Cette commande renvoi un tableau avec la liste des services :



Screenshot 3 commande PowerShell Get-Service

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

Mais le nom du service est parfois **tronqué**, pour y remédier il faut utiliser la commande :

***Get-Service | format-table -auto***



```

PS C:\Windows\system32> Get-Service | format-table -auto

Status Name                               DisplayName
-----
Stopped AarSvc_3ff90c9                     Agent Activation Runtime_3ff90c9
Running AdobeARMservice                   Adobe Acrobat Update Service
Stopped AJRouter                           Service de routeur AllJoyn
Stopped ALG                               Service de la passerelle de la couche Application
Running AMD External Events Utility        AMD External Events Utility
Stopped AppIDSvc                           Identité de l'application
Running Appinfo                            Informations d'application
Stopped AppMgmt                            Gestion d'applications
Stopped AppReadiness                       Préparation des applications
Stopped AppVClient                         Microsoft App-V Client
Stopped AppXSvc                            Service de déploiement AppX (AppXSVC)
Stopped AssignedAccessManagerSvc            Service AssignedAccessManager
Running AudioEndpointBuilder                Générateur de points de terminaison du service Audio Windows
Running Audiosrv                           Audio Windows

```

Commande 2: *Get-Service | format-table -auto*

Ce qui est plus lisible.

La commande suivante permet d'obtenir le **nombre total des services Windows** :

***(Get-Service | Measure-Object).Count***



```

PS C:\Windows\system32> (Get-Service | Measure-Object).Count
274
PS C:\Windows\system32>

```

Commande 3: *(Get-Service | Measure-Object).Count*

Donc sur cette machine il y a **274 services** Windows.

On peut filtrer les services, en ne prenant en compte que ceux qui sont en cours d'exécution :

***(Get-Service | Where-Object {\$\_.Status -eq "Running"} | Measure-Object).Count***



```

PS C:\Windows\system32> (Get-Service | Where-Object {$_.Status -eq "Running"} | Measure-Object).Count
102
PS C:\Windows\system32>

```

Commande 4: *(Get-Service | Where-Object {\$\_.Status -eq "Running"} | Measure-Object).Count*

On obtient donc **102 services actifs**; ce qui fait quand même beaucoup, pour une configuration de développement.

D'où la nécessité **d'optimiser le système** en réduisant le nombre de services, en désactivant ceux qui sont inutiles.

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

Pour désactiver les services inutiles, il convient de suivre ces étapes:

1. Ouvrir une fenêtre « Exécuter » Windows
2. Taper la commande : `services.msc`



- 3.
4. On obtient la fenêtre « Services »

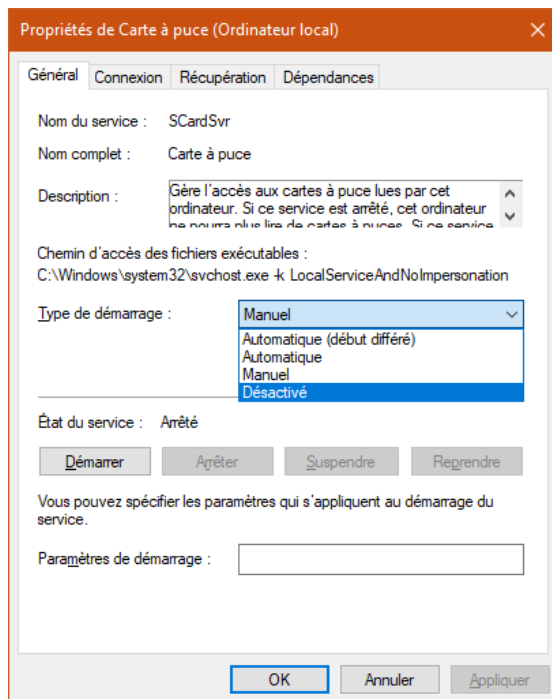


5. Choisir un service inutile à désactiver, ici dans le contexte, le service « Carte à puce ».  
Double-cliquer sur le service en question.



- 6.
7. Positionner le statut de « Type de démarrage » sur « Désactivé ».  
Cliquer sur « OK ».

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



#### 8. Le service est bien désactivé :



Une fois tous les services inutiles désactivés, on peut vérifier la liste des services en utilisant la commande :

```
Get-Service | Where-Object {$_.Status -eq "Running"} | out-gridview
```



|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

```

Administrateur : Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Get-Service | Where-Object {$_.Status -eq "Running"} | out-gridview

```

Commande 5: `Get-Service | Where-Object {$_.Status -eq "Running"} | out-gridview`

Ce qui donne une présentation sous forme de table :



| Status  | Name                        | DisplayName  |
|---------|-----------------------------|--|
| Running | AdobeARMSvc                 | Adobe Acrobat Update Service   |
| Running | AMD External Events Utility | AMD External Events Utility  |
| Running | Appinfo                     | Informations d'application   |
| Running | AudioEndpointBuilder        | Générateur de points de terminaison du service Audio Windows         |
| Running | Audiosrv                    | Audio Windows  |
| Running | BFE                         | Moteur de filtrage de base   |
| Running | BrokerInfrastructure        | Service d'infrastructure des tâches en arrière-plan                  |
| Running | Browser                     | Explorateur d'ordinateurs  |
| Running | BTAGService                 | Service de passerelle audio Bluetooth                                |
| Running | BthAvctpSvc                 | Service AVCTP  |
| Running | bthserv                     | Service de prise en charge Bluetooth                                 |
| Running | camsvc                      | Service Gestionnaire d'accès aux fonctionnalités                     |
| Running | cbdhsvc_3d10da9             | Service utilisateur du Presse-papiers_3d10da9                        |
| Running | CDPSvc                      | Service de plateforme des appareils connectés                        |
| Running | CDPUserSvc_3d10da9          | Service pour utilisateur de plateforme d'appareils connectés_3d10da9 |
| Running | ClickToRunSvc               | Service Microsoft Office « Démarrer en un clic »                     |
| Running | CoreMessagingRegistrar      | CoreMessaging  |
| Running | CryptSvc                    | Services de chiffrement  |
| Running | DcomLaunch                  | Lanceur de processus serveur DCOM                                    |

Screenshot 4: Powershell GridView



**Pour aller plus loin et pour avoir plus d'informations et des conseils sur la désactivation des services système sous Windows et Windows Server, il est conseillé de consulter la documentation Microsoft<sup>4849</sup>.**

**<https://docs.microsoft.com/en-us/windows-server/security/windows-services/security-guidelines-for-disabling-system-services-in-windows-server>**

<sup>48</sup> <https://docs.microsoft.com/fr-fr/windows-server/security/windows-services/security-guidelines-for-disabling-system-services-in-windows-server>

<sup>49</sup> <https://docs.microsoft.com/fr-fr/windows/application-management/per-user-services-in-windows>

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

<https://docs.microsoft.com/fr-fr/windows/application-management/per-user-services-in-windows>

#### 1.6.1.4. Configuration professionnelle

Cette configuration composée d'une station de travail, contiendra les mêmes caractéristiques de performance que la précédente. La différence majeure réside dans la présence de deux processeurs.

Disposer d'une machine bi-processeur n'est pas obligatoire pour faire du développement classique.

En fait cette configuration est utile si l'on développe de grosses applications et / ou pour des architectures distribuées.

Cette configuration est tout indiquée pour faire de la programmation concurrente<sup>50</sup> et donc du multithreading<sup>51</sup>. Ce qui est possible en Java. Ce qui l'est encore plus dans un langage reposant sur un paradigme de programmation plus fonctionnel<sup>52</sup>, comme Scala par exemple.

Les deux processeurs de cette configuration : le Xeon (Intel) et le Ryzen Threadripper (AMD) supportent le multithreading.

##### 1.6.1.4.1. Le multithreading

Un processeur ne réalise qu'une seule opération à la fois, en d'autres termes le CPU effectue des calculs en série. Contrairement au GPU, qui lui effectue des calculs en parallèle.

Donc pour augmenter la vitesse de calcul du CPU on a recours, notamment, au multithreading.

Un **thread** ou fil d'exécution est similaire à un processus car tous deux représentent l'exécution d'un ensemble d'instructions du langage machine d'un processeur.

Du point de vue de l'utilisateur, ces exécutions semblent se dérouler en parallèle.

Toutefois, là où chaque processus possède sa propre mémoire virtuelle, les threads d'un même processus se partagent sa mémoire virtuelle<sup>53</sup>.

<sup>50</sup> [https://fr.wikipedia.org/wiki/Programmation\\_concurrente](https://fr.wikipedia.org/wiki/Programmation_concurrente)

<sup>51</sup> [https://en.wikipedia.org/wiki/Multithreading\\_\(computer\\_architecture\)](https://en.wikipedia.org/wiki/Multithreading_(computer_architecture))

<sup>52</sup> [https://fr.wikipedia.org/wiki/Programmation\\_fonctionnelle](https://fr.wikipedia.org/wiki/Programmation_fonctionnelle)

<sup>53</sup> [https://fr.wikipedia.org/wiki/Thread\\_%28informatique%29](https://fr.wikipedia.org/wiki/Thread_%28informatique%29)

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



Figure 12 Un processus avec deux threads.

Un programme Java dispose d'au moins un thread, c'est le « thread principal » ; celui-ci est généré par la JVM au lancement du programme, lors de l'appel de la méthode « main() ».

Un processeur qui exécute plusieurs threads, simultanément, est dit **multithread**.

Dans cette configuration avec deux processeurs, pour une application multithread, chaque processeur va exécuter un thread différent.

Donc un programme multithread disposant de plusieurs threads va fonctionner plus efficacement en exécutant plusieurs tâches simultanément (plusieurs unités d'exécution).



Le multithreading est une forme de parallélisation ou de division du travail pour un traitement simultané. Au lieu de donner une charge de travail importante à un seul cœur, les programmes multithreads divisent le travail en plusieurs tâches (threads) logicielles. Ces tâches sont traitées en parallèle par différents cœurs de processeurs pour gagner du temps<sup>55</sup>.

#### 1.6.1.4.2. L'hyper-threading

Hyper-threading (officiellement appelée Hyper-Threading Technology (HTT) ou HT Technology (HT)) est la mise en œuvre par l'entreprise Intel du simultaneous multithreading (SMT) à deux voies dans ses microprocesseurs<sup>56</sup>.

<sup>54</sup> [https://upload.wikimedia.org/wikipedia/commons/thumb/a/a5/Multithreaded\\_process.svg/330px-Multithreaded\\_process.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/a/a5/Multithreaded_process.svg/330px-Multithreaded_process.svg.png)

<sup>55</sup> <https://www.intel.com/content/www/us/en/gaming/resources/hyper-threading.html>

<sup>56</sup> <https://fr.wikipedia.org/wiki/Hyper-threading>

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

La technologie Intel® Hyper-Threading est une innovation matérielle qui permet d'exécuter plusieurs threads sur chaque cœur. Davantage de threads signifie que davantage de tâches peuvent être exécutées en parallèle.<sup>57</sup>

Les CPU considérés ici sont donc multi-cœurs, ceci leur permet donc d'exécuter plusieurs calculs en parallèle. En effet, les différentes sollicitations faites au processeur sont réparties au niveau de chaque cœur, physique, du CPU.

Avec la technologie du SMT, chaque cœur **physique** du processeur est divisé en deux cœurs **logiques**.

Dans ce cas, donc au niveau du système d'exploitation, un cœur du processeur physique est considéré comme deux processeurs logiques. Ce qui permet d'effectuer deux tâches simultanément.

Mais pour pouvoir bénéficier de l'hyper-threading, encore faut-il qu' il soit bien activé sur la machine.

#### 1.6.1.4.2.1. Vérifier si l'hyper-threading est activé

Voici les étapes à suivre pour vérifier si l'hyper-threading est activé sur le système d'exploitation :

1. Ouvrir une fenêtre « Exécuter » Windows<sup>58</sup>
2. Taper la commande : `taskmgr`



- 3.
4. Cliquer sur l'onglet « Performances »
5. Sur le volet de gauche cliquer sur le graphique « Processeur »

<sup>57</sup> <https://www.intel.com/content/www/us/en/gaming/resources/hyper-threading.html>

<sup>58</sup> Touches Windows + R

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



Figure 13 Hyper-threading et processeur

- En bas de la fenêtre sous le graphique « Processeur », on doit avoir un nombre de processeurs logiques double de celui du nombre de cœurs.  
Ici c'est bien le cas : le processeur possède 4 cœurs physiques et 8 cœurs logiques => donc l'hyper-threading est, bien, activé.

Si l'hyper-threading n'est pas activé, alors il faut en passer par le Bios ou l'UEFI de la machine pour changer cet état.

Pour aller plus loin, afin d'analyser, dans les détails, le comportement de l'application, on peut utiliser un outil de profilage pour Java.

#### 1.6.1.4.3. Les outils de profilage pour Java

Effectivement, si l'on veut connaître, en temps réel, l'état des threads, de la mémoire, du processeur et l'exécution du code, on peut utiliser un outil de profilage.

##### 1.6.1.4.3.1. Finalité d'un outil de profilage

Un outils de profilage analyse, dans les détails, le déroulement des opérations au niveau de la JVM. Ceci dans le but d'en déduire des mesures sur les exécution de threads, l'utilisation de la mémoire (fuite de mémoire), ou encore l'utilisation du processeur. La finalité étant d'identifier les goulots d'étranglements, maintenant bien connu.

Avec ce diagnostic, le développeur pourra identifier les parties du code source qu'il faudra optimiser.

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

### 1.6.1.4.3.2. Quelques outils de profilage pour Windows

Voici quelques outils de profilage, dont certains sont libres et d'autres commerciaux.

#### 1.6.1.4.3.2.1. *Profilers commerciaux*

##### **JProfiler**

JProfiler<sup>59</sup> est un outil de profilage Java qui aide les utilisateurs à résoudre les goulots d'étranglement liés aux performances, à localiser les fuites de mémoire et à comprendre les problèmes de thread.

Il existe d'autres outils open source, mais JProfiler reste un logiciel incontournable dans le domaine du profiling en Java.

##### **YourKit Java Profiler**

YourKit Java Profiler<sup>60</sup> comme JProfiler est un profiler pour les applications écrites en Java. Il reprend les fonctionnalités de l'outil précédent. Il existe sous forme de plugins pour les principaux IDE Java : Eclipse, IntelliJ IDEA et NetBeans.

#### 1.6.1.4.3.2.2. *Alternative aux profilers commerciaux*

##### **VisualVM**

VisualVM<sup>61</sup> est un outil visuel intégrant plusieurs outils JDK en ligne de commande et des capacités de profilage légères. Conçu pour des utilisations de production ou de développement.

##### **JConsole**

JConsole<sup>62</sup> est un outil de surveillance graphique pour surveiller la machine virtuelle Java (JVM) et les applications Java. Il utilise les fonctionnalités sous-jacentes de Java Virtual Machine pour fournir des informations sur les performances et la consommation des ressources des applications exécutées sur la plate-forme Java à l'aide de la technologie Java Management Extensions (JMX).

JConsole fait partie du kit de développement Java (JDK) ; on le retrouve dans le répertoire d'installation du JDK dans le dossier **bin**.

Pour retrouver ce répertoire, il suffit d'afficher la variable d'environnement **JAVA\_HOME**, grâce à une CLI<sup>63</sup> et en saisissant la commande : **echo %JAVA\_HOME%**

<sup>59</sup> <https://www.ej-technologies.com/products/jprofiler/overview.html>

<sup>60</sup> <https://www.yourkit.com/java/profiler>

<sup>61</sup> <https://visualvm.github.io/>

<sup>62</sup> <https://docs.oracle.com/javase/10/management/using-jconsole.htm>

<sup>63</sup> Command-line interface (Invite de commandes)

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



```

Microsoft Windows [version 10.0.19044.1889]
(c) Microsoft Corporation. Tous droits réservés.

C:\Windows\system32>echo %JAVA_HOME%
C:\Program Files\Java\jdk-18.0.1.1

C:\Windows\system32>

```

Commande 6: `echo %JAVA_HOME%`

Et le dossier bin :

**`C:\Program Files\Java\jdk-18.0.1.1\bin`**



Figure 14 jconsole.exe

Cette console graphique peut être démarrée à l'aide de la commande : **`jconsole`**



Commande 7 jconsole

On obtient la fenêtre « JConsole New Connection » :

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



Screenshot 5 :fenêtre JConsole New Connection

Il ne reste plus qu'à commencer à l'utiliser<sup>64</sup> pour inspecter une application Java.

Avant de passer à l'installation de l'environnement de développement Java, encore faut-il connaître le type d'architecture du système installé sur l'ordinateur.

### 1.6.2. Connaître le type d'architecture installée sur la machine

Avant de pouvoir télécharger un JDK et un IDE, il convient, au préalable, de connaître les architectures du processeur et du système d'exploitation présents sur l'ordinateur.

Sous Windows on a trois possibilités : PowerShell, le gestionnaire des tâches ou l'explorateur de fichiers.

#### 1.6.2.1. Utiliser PowerShell

Avec deux commandes :

- ✓ `$env:PROCESSOR_ARCHITECTURE`
- ✓ `(Get-WmiObject Win32_OperatingSystem).OSArchitecture`

<sup>64</sup> <https://docs.oracle.com/javase/7/docs/technotes/guides/management/jconsole.html>



|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



```

Administrateur : Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> $env:PROCESSOR_ARCHITECTURE
AMD64
PS C:\Windows\system32> (Get-WmiObject Win32_OperatingSystem).OSArchitecture
64 bits
PS C:\Windows\system32>

```

Commande 8: \$env et WmiObject

On peut utiliser une CLI plus classique, avec la commande : `echo %PROCESSOR_ARCHITECTURE%`



```

Administrateur : C:\Windows\system32\cmd.exe

C:\Windows\system32>echo %PROCESSOR_ARCHITECTURE%
AMD64

C:\Windows\system32>

```

Commande 9: `echo %PROCESSOR_ARCHITECTURE%`

### 1.6.2.2. Utiliser le gestionnaire des tâches

Pour vérifier l'architecture sur laquelle repose le système d'exploitation, on peut suivre ces étapes :

1. Ouvrir une fenêtre « Exécuter » Windows<sup>65</sup>
2. Taper la commande : `taskmgr`
3. La fenêtre gestionnaire des tâches s'ouvre
4. Cliquer sur l'onglet « Détails »

<sup>65</sup> Touches Windows + R

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



- On constate dans la colonne « Plateforme » que tous les processus tournent bien en 64 bits ; cela confirme donc bien que l'architecture du processeur est bien en 64 bits.

### 1.6.2.3. Utiliser l'explorateur de fichiers

Si les dossiers : **C:\Program Files** et **C:\Program Files (x86)** sont présents dans l'explorateur de fichiers de Windows, alors cela signifie que Windows est en version 64 bits et , donc que nécessairement, l'architecture le processeur est également en 64 bits.

#### 1.6.2.3.1. Program Files(x86) versus Program Files

Il faut cliquer sur le dossier « Programmes » pour que **C:\Program Files** , qui est le vrai chemin complet s'affiche dans la barre d'adresse :

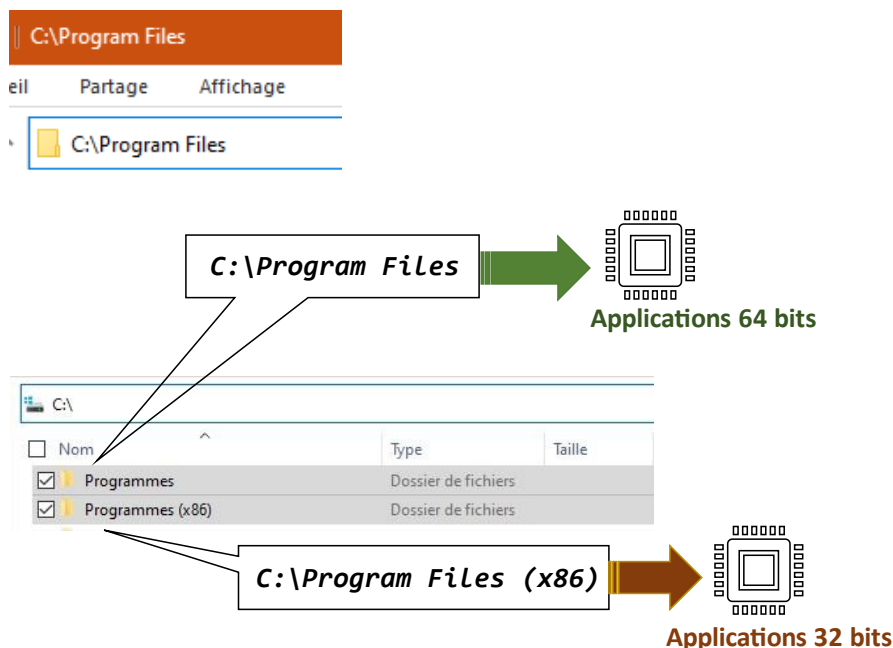


Figure 15; Program Files(x86) vs Program Files

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

Mais pourquoi existe-t-il deux dossiers différents pour stocker les fichiers de programmes dans Windows ?

La réponse est : pour séparer les applications 64 bits et 32 bits.

En effet, sur les versions 64 bits de Windows, les applications 64 bits sont installées dans le dossier Program Files. Cependant, les versions 64 bits de Windows prennent également en charge les programmes 32 bits<sup>66</sup> en utilisant une couche de compatibilité appelée WoW64<sup>67</sup> et un redirecteur de système de fichiers<sup>68</sup>.

Pour séparer les programmes 32 bits et 64 bits sur le PC, les concepteurs de Windows ont choisi d'installer les programmes 32 bits dans le dossier C:\Program Files (x86).

#### 1.6.2.3.2. System32 versus SysWOW64

Sur un ordinateur 64 bits, les programmes 64 bits stockent leurs fichiers dans C: Program Files et le dossier C: Windows\System32 à l'échelle du système contient des bibliothèques 64 bits.

Les programmes 32 bits stockent leurs fichiers dans C: Program Files (x86) et le dossier à l'échelle du système est C: Windows\SysWOW64.

Ces deux répertoires qui contiennent les bibliothèques de code système d'exploitation trouvent dans le dossier système : C:\Windows :

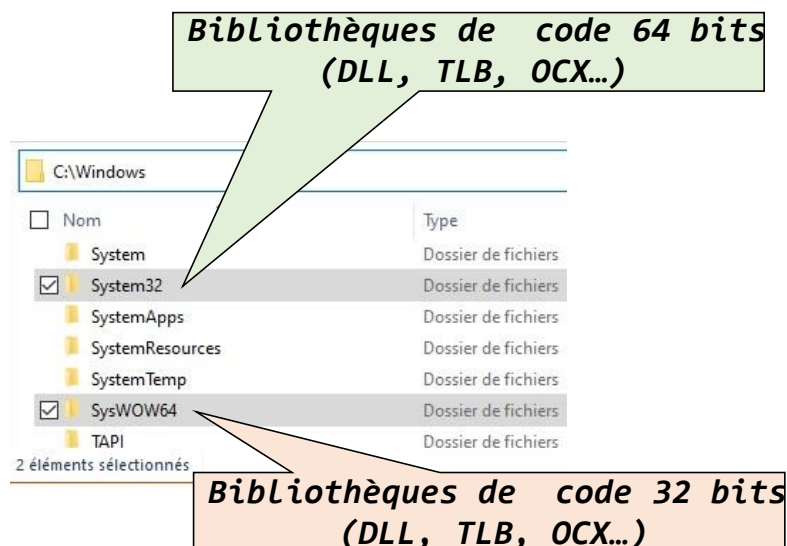


Figure 16: Répertoires System32 SysWoW64

<sup>66</sup> Un système avec un processeur 64 bits peut faire tourner des applications 32 bits, mais l'inverse est faux.

<sup>67</sup> Qui signifie: «Windows 32 bits sur Windows 64 bits»

<sup>68</sup> <https://msdn.microsoft.com/en-us/library/windows/desktop/aa384187%28v=vs.85%29.aspx?f=255&MSPPErr=-2147217396>

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

Le schéma suivant résume le processus de traitement d'un programme ou d'une application qui tourne sous Windows 64 bits :



Algorithme 1 :traitement d'un programme sous Windows

Après cette longue introduction , utile pour rappeler les bases matérielles et techniques sur lesquels reposent la programmation, on peut maintenant passer à l'étape de l'installation de l'environnement de développement en Java.

## 2. Installation de l'environnement de développement Java

Il sera question d'installer une version de Java, ainsi que l'outil, principal, de développement, en l'occurrence un IDE.

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2>Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

## 2.1. Procédure d'installation de Java

Cette procédure d'installation est composée de trois, principales, étapes : le téléchargement, sa vérification son installation sur le système d'exploitation.

### 2.1.1. Téléchargement du JDK

Parmi toutes les distributions du JDK vues précédemment,, on choisira d'installer Oracle JDK, à des fins de développement, donc il n'y aura pas de problème de licence.

Evidemment, si on le souhaite, on peut installer plusieurs<sup>69</sup> autres distributions du JDK.

La première étape consiste donc à installer le Java SE (de type JDK) sur le poste de développement. Pour ce faire on système d'exploitation rend sur ce site d'Oracle :

<https://www.oracle.com/java/technologies/downloads>

et , ensuite on choisit de télécharger une version du JDK ; ici on optera pour le JDK 18.



**Java 18 and Java 17 available now**

Java 17 LTS is the latest long-term support release for the Java SE platform. JDK 18 and JDK 17 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions](#).

JDK 18 will receive updates under these terms, until September 2022 when it will be superseded by JDK 19

JDK 17 will receive updates under these terms, until at least September 2024.

**Java 18** Java 17

**Java SE Development Kit 18.0.2.1 downloads**

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications and components using the Java programming language.

The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.

Linux macOS **Windows**

| Product/file description | File size | Download   |
|--------------------------|-----------|--|
| x64 Compressed Archive   | 172.93 MB | <a href="https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.zip">https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.zip</a> (sha256 <a href="#">L2</a> ) |
| x64 Installer            | 153.45 MB | <a href="https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.exe">https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.exe</a> (sha256 <a href="#">L2</a> ) |
| x64 MSI Installer        | 152.33 MB | <a href="https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.msi">https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.msi</a> (sha256 <a href="#">L2</a> ) |

Screenshot 6: Java SE Development Kit 18 downloads

<sup>69</sup> Sous réserve d'avoir des ressources système suffisantes : une machine assez puissante pour supporter les autres versions du JDK.

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

On fera l'hypothèse, réaliste, que le développeur est administrateur de sa machine. On choisit donc de télécharger le fichier *.msi*.

Dans le cas où l'on n'est pas administrateur, il faut télécharger l'archive *.zip*.

Dans le cas du fichier *msi*, une fois le téléchargement terminé, on devrait obtenir dans le répertoire local :



Screenshot 7 : JDK-18.msi

Une fois ce fichier msi téléchargé, il est prudent de vérifier son intégrité avant de cliquer sur installer.

#### 2.1.1.1. Vérifier l'intégrité du fichier téléchargé

Il est conseillé de vérifier l'intégrité du fichier téléchargé pour être sûr et quasiment certain qu'il n'est pas corrompu.

A cette fin, il faut calculer la somme de contrôle<sup>70</sup> ou *checksum*, en anglais, ou « empreinte numérique » de ce fichier téléchargé. Pour calculer celle-ci, il faut appliquer une fonction de hachage sur ce fichier et la comparer avec celle du fichier original :



Screenshot 8 : MSI\_SHA-256

Il existe plusieurs fonction de hachage dont SHA-256, qui nous intéresse ici, puisque c'est celle utilisée par Oracle dans ce contexte.

En cliquant sur le lien on obtient, dans la fenêtre du navigateur, le *checksum* suivant :



Screenshot 9 : checksum Oracle JDK 18

Donc l'empreinte du fichier d'origine fournie par Oracle est :

71d547c0ae19fd8423afaca1bc9f197a105098c0ac31d20611c44d31aeffdca5

<sup>70</sup> [https://fr.wikipedia.org/wiki/Somme\\_de\\_contr%C3%B4le](https://fr.wikipedia.org/wiki/Somme_de_contr%C3%B4le)

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

C'est cette suite de nombres au format hexadécimal que l'on va maintenant comparer au fichier téléchargé, qui se trouve en local.

Avec PowerShell on peut vérifier l'intégrité du fichier en comparant des deux checksum, en suivant ces étapes :

1. Se placer dans le répertoire du fichier téléchargé (ici D:\)
2. Ouvrir la fenêtre PowerShell
3. Utiliser *Get-FileHash*<sup>71</sup> en entrant la commande :  
**`Get-FileHash jdk-18_windows-x64_bin.msi -Algorithm SHA256 | Format-List`**



```

Windows PowerShell
PS D:\> Get-FileHash jdk-18_windows-x64_bin.msi -Algorithm SHA256 | Format-List

Algorithm : SHA256
Hash      : 71D547C0AE19FD8423AFACA1BC9F197A105098C0AC31D20611C44D31AEFFDCA5
Path      : D:\jdk-18_windows-x64_bin.msi

PS D:\>

```

Commande 10 : *Get-FileHash*

4. On obtient, donc, le hash du fichier téléchargé, qu'il faut maintenant comparer avec celui qui se trouve sur la page du site d'Oracle qui est:  
71d547c0ae19fd8423afaca1bc9f197a105098c0ac31d20611c44d31aeffdca5
5. Entrer la commande : **`"HASH1" -eq "HASH2"`** (Renvoie ***True*** si les empreintes sont identiques, ***False*** sinon)



```

Windows PowerShell
PS D:\> Get-FileHash jdk-18_windows-x64_bin.msi -Algorithm SHA256 | Format-List

Algorithm : SHA256
Hash      : 71D547C0AE19FD8423AFACA1BC9F197A105098C0AC31D20611C44D31AEFFDCA5
Path      : D:\jdk-18_windows-x64_bin.msi

PS D:\> "71d547c0ae19fd8423afaca1bc9f197a105098c0ac31d20611c44d31aeffdca5" -eq "71D547C0AE19FD8423AFACA1BC9F197A105098C0AC31D20611C44D31AEFFDCA5"
True
PS D:\>

```

Commande 11: **`"HASH1" -eq "HASH2"`**

6. C'est parfait , le fichier téléchargé est identique à l'original, l'installation du JDK peut, maintenant, vraiment débuter.

<sup>71</sup> <https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/get-filehash?view=powershell-7.2>

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



On peut, également, pour calculer l'empreinte d'un fichier, utiliser la commande `certutil.exe`<sup>72</sup> dans une invite de commandes avec `-hashfile`. Comme ici :

```

C:\Windows\System32\cmd.exe
Microsoft Windows [version 10.0.19044.1889]
(c) Microsoft Corporation. Tous droits réservés.

D:\>certutil -hashfile jdk-18_windows-x64_bin.msi SHA256
Hachage SHA256 de jdk-18_windows-x64_bin.msi :
71d547c0ae19fd8423afaca1bc9f197a105098c0ac31d20611c44d31aeffdca5
CertUtil: -hashfile La commande s'est terminée correctement.

D:\>

```

Commande 12: `certutil -hashfile`

Il suffit ensuite, comme précédemment, de comparer l'empreinte du fichier avec celle fournie par Oracle, en utilisant la commande : `if "HASH1" == "HASH2" echo True`.

```

C:\Windows\System32\cmd.exe
Microsoft Windows [version 10.0.19044.1889]
(c) Microsoft Corporation. Tous droits réservés.

D:\>certutil -hashfile jdk-18_windows-x64_bin.msi SHA256
Hachage SHA256 de jdk-18_windows-x64_bin.msi :
71d547c0ae19fd8423afaca1bc9f197a105098c0ac31d20611c44d31aeffdca5
CertUtil: -hashfile La commande s'est terminée correctement.

D:\>if "71d547c0ae19fd8423afaca1bc9f197a105098c0ac31d20611c44d31aeffdca5" == "71d547c0ae19fd8423afaca1bc9f197a105098c0ac31d20611c44d31aeffdca5" echo true
true

D:\>

```

Commande 13 : `certutil if "HASH1" == "HASH2" echo True`

### 2.1.2. Installation de Java

La procédure d'installation de Java, sous Windows est classique, il suffit de cliquer sur le fichier `.msi`. L'installation se fait automatiquement.

#### 2.1.2.1. Procédure d'installation de Java avec l'installateur automatique

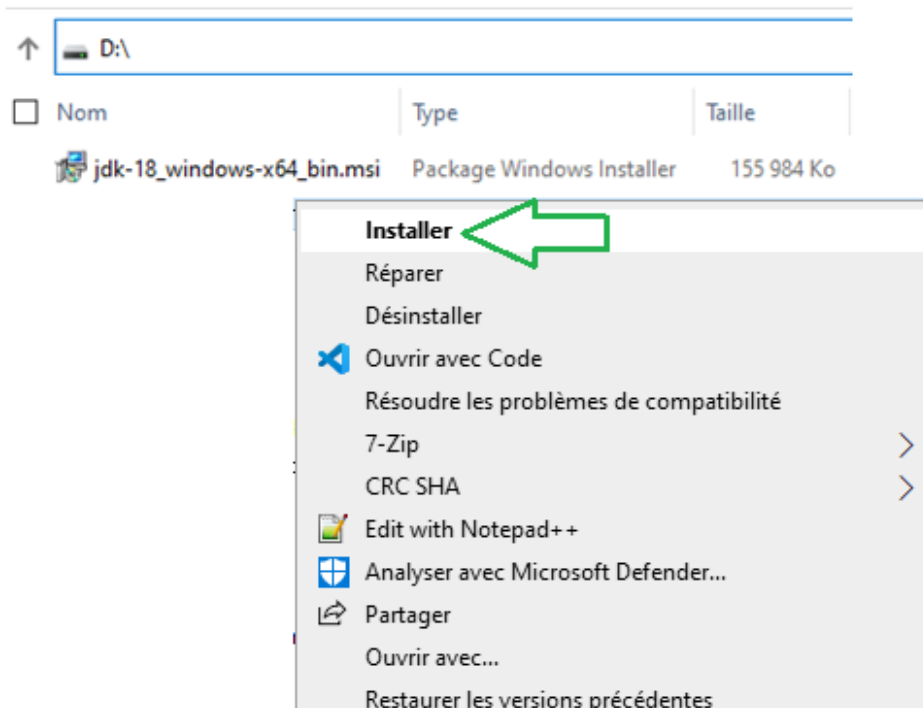
Voici les principales étapes de l'installation de Java:

1. Faire un clic droit sur le fichier `.msi` et choisir « Installer » :

<sup>72</sup> [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc732443\(v=ws.11\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc732443(v=ws.11))



|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



Screenshot 10: fichier msi : installation

2. L'installation démarre avec cet écran :



3. L'écran suivant indique le répertoire d'installation par défaut :

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



Screenshot 11 : Dossier d'installation du JDK par défaut

**Le mieux étant d'accepter les choix proposés par défaut** ; il suffira donc de cliquer sur « Next », sur chaque écran.

On peut, également, se référer à la documentation d'installation : [Installation of the JDK on Microsoft Windows Platforms](https://docs.oracle.com/en/java/javase/17/install/installation-jdk-microsoft-windows-platforms.html#GUID-A7E27B90-A28D-4237-9383-A58B416071CA)<sup>73</sup>, d'Oracle.

#### 4. Progression de l'installation



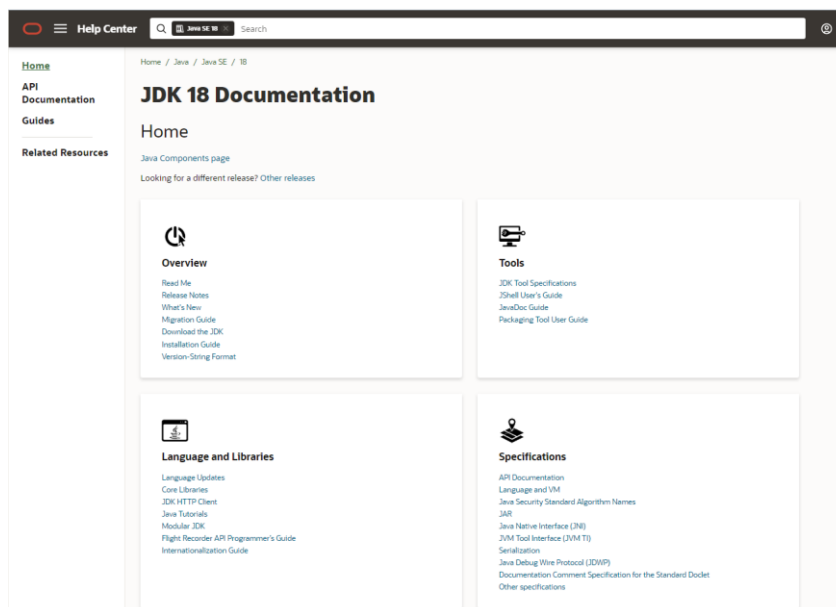
<sup>73</sup> <https://docs.oracle.com/en/java/javase/17/install/installation-jdk-microsoft-windows-platforms.html#GUID-A7E27B90-A28D-4237-9383-A58B416071CA>

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

5. A cette étape si on choisit de cliquer sur « Next Steps »...



6. On est dirigé sur la documentation en ligne<sup>74</sup> de Java 18 :



7. Si on choisit de cliquer sur « Close »...

<sup>74</sup> <https://docs.oracle.com/en/java/javase/18/index.html>

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



... l'installation se termine.

#### 8. Vérification du dossier d'installation du JDK

Le chemin du dossier d'installation du JDK, par défaut, est :

C:\Program Files\Java\jdk-18.0.2.1

C'est donc bien une version 64 bits du JDK, puisque le JDK a été automatiquement installé dans le dossier : C:\Program Files.



Screenshot 12 :Dossier d'installation du JDK

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

A l'issue de l'installation du JDK, il convient de vérifier la variable d'environnement **JAVA\_HOME**.

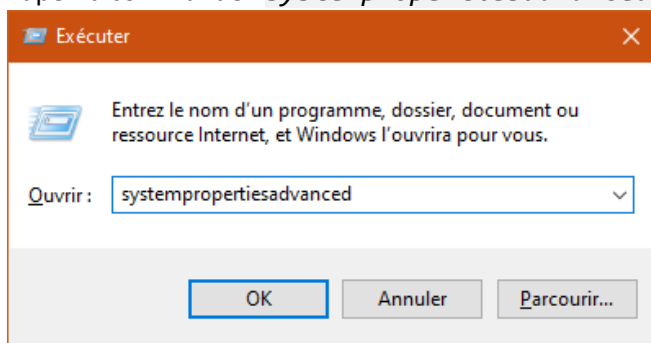
### 2.1.2.2. Configuration des variables d'environnement Java

Ensuite, nous configurons la variable d'environnement **JAVA\_HOME** pour Java.  
Ce n'est pas obligatoire puisque cela a été effectué automatiquement par l'installateur Windows .  
Mais si plusieurs versions de Java sont installées, il est nécessaire de configurer cette variable d'environnement, dans le cas où on souhaite utiliser une version particulière de Java.  
Il est donc utile de connaître la méthode à suivre.

il existe deux procédés pour configurer la variable d'environnement Java : en passant par les propriétés avancées du système ou bien via un fichier batch.

2.1.2.2.1. Configurer la variable d'environnement Java via les propriétés avancées du système  
Les principales étapes pour la première méthode sont:

1. Ouvrir la fenêtre exécuter, avec les touches **Windows + R**
2. Taper la commande : `systempropertiesadvanced`



- 3.
4. Dans la fenêtre suivante cliquer sur « Variables d'environnement » :

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



- 5.
6. Modifier la variable « Path » afin qu'elle contienne aussi le chemin vers l'exécutable Java
7. Pour cela dans la fenêtre suivante choisir la partie « Variables système », cliquer sur « Path » puis sur « Modifier » :



- 8.

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

9. Dans la fenêtre suivante cliquer sur «Nouveau» :



10.

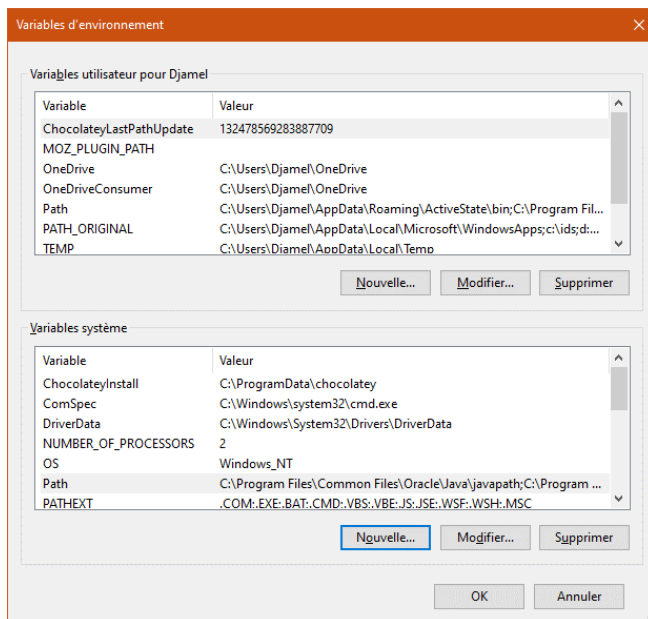
11. Dans la fenêtre suivante, entrer dans le nouveau champ le chemin du dossier bin du jdk :  
« C:\Program Files\Java\jdk-18.0.2.1\bin »



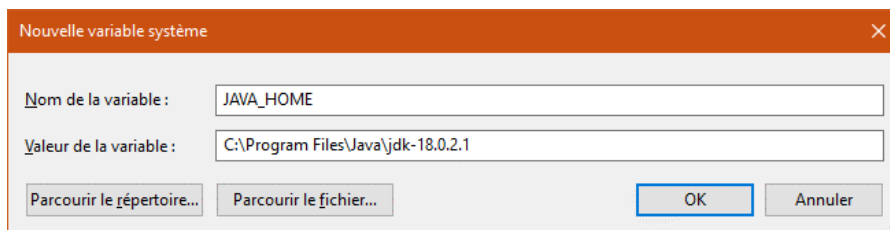
12. Ensuite il faut configurer la variable JAVA\_HOME

Pour cela : dans la fenêtre suivante, dans la partie « Variables système », cliquer sur  
« Nouvelle... »

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



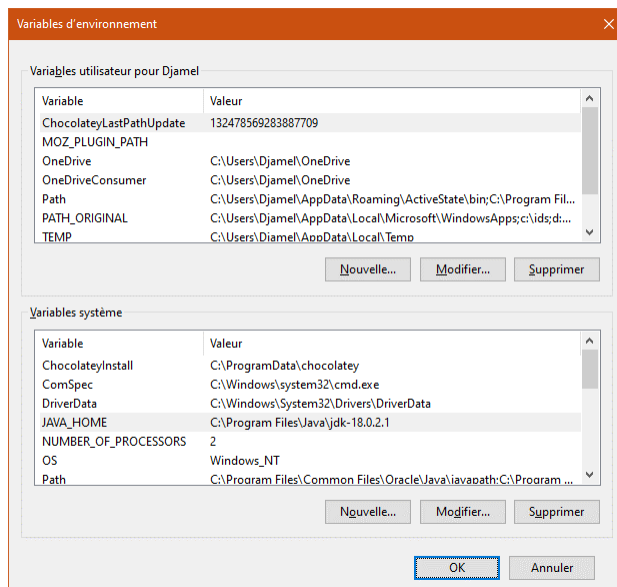
13. Dans la fenêtre suivante, dans le champ « Nom de la variable » entrer : JAVA\_HOME et dans le champ « Valeur de la variable » inscrire le chemin du répertoire d'installation du JDK, mais sans le dossier bin : C:\Program Files\Java\jdk-18.0.2.1



14. Cliquer sur OK



|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



15. On vérifie que la variable JAVA\_HOME a bien été prise en compte

16. Et pour terminer cliquer 2 fois sur OK pour valider et fermer les fenêtres précédentes.

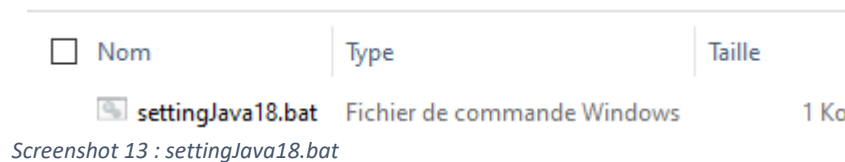
#### 2.1.2.2.2. Configurer la variable d'environnement Java à l'aide d'un fichier batch

Le second procédé pour configurer la variable d'environnement Java consiste à écrire un fichier batch.

Le script batch ressemblera à celui-ci :

```
1 echo Setting Java 18
2 @echo off
3 set "C:\Program Files\Java\jdk-18.0.2.1"
4 echo %JAVA_HOME%
5 set PATH=%PATH%;%JAVA_HOME%\bin
6 pause
```

Il faut ensuite enregistrer le fichier sous, par exemple, *settingJava18.bat* :



Il suffit de cliquer sur le fichier pour exécuter la commande.

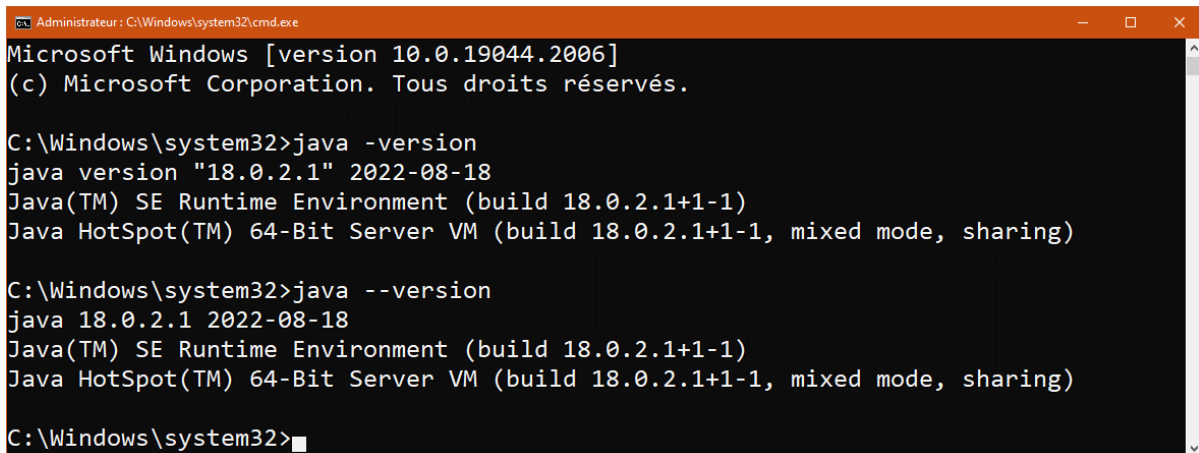
#### 2.1.2.3. Vérification de l'installation du JDK

Vérifications des points suivants : la version de Java, le dossier d'installation du JDK et la version du compilateur Java.

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

### 2.1.2.3.1. Vérifier la version de Java installée

Pour vérifier la version de Java, actuellement, installée on utilisera un shell cmd ou PowerShell ; ensuite il faut taper la commande : **java -version** ou bien : **java --version**



```

Microsoft Windows [version 10.0.19044.2006]
(c) Microsoft Corporation. Tous droits réservés.

C:\Windows\system32>java -version
java version "18.0.2.1" 2022-08-18
Java(TM) SE Runtime Environment (build 18.0.2.1+1-1)
Java HotSpot(TM) 64-Bit Server VM (build 18.0.2.1+1-1, mixed mode, sharing)

C:\Windows\system32>java --version
java 18.0.2.1 2022-08-18
Java(TM) SE Runtime Environment (build 18.0.2.1+1-1)
Java HotSpot(TM) 64-Bit Server VM (build 18.0.2.1+1-1, mixed mode, sharing)

C:\Windows\system32>

```

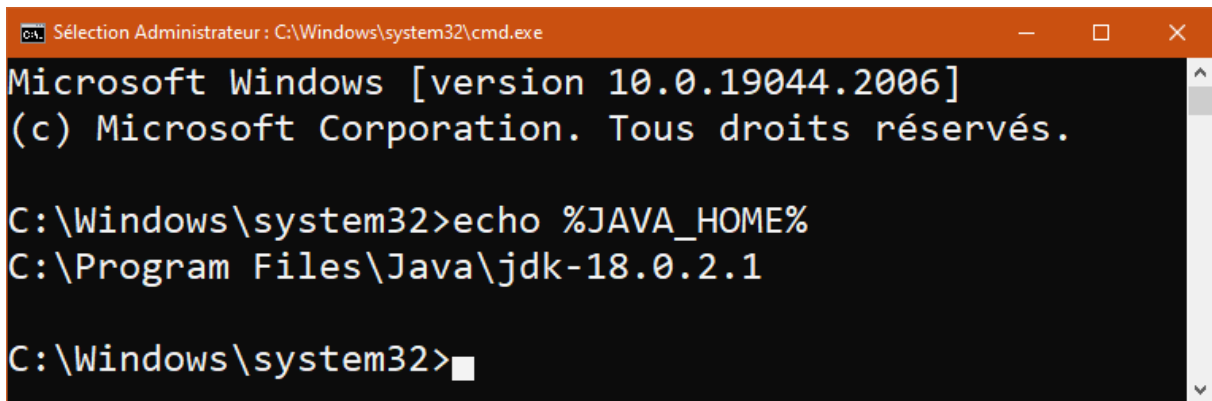
Commande 14 : java -version

Avec les deux syntaxes le résultat est équivalent.

### 2.1.2.3.2. Afficher le dossier d'installation du JDK

Pour afficher le contenu de la variable d'environnement Java, on utilisera les commandes :

✓ **echo %JAVA\_HOME%** → cmd



```

Microsoft Windows [version 10.0.19044.2006]
(c) Microsoft Corporation. Tous droits réservés.

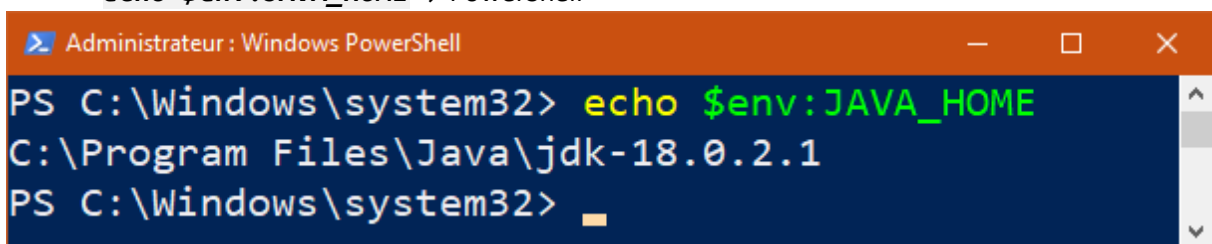
C:\Windows\system32>echo %JAVA_HOME%
C:\Program Files\Java\jdk-18.0.2.1

C:\Windows\system32>

```

Commande 15 : echo %JAVA\_HOME%

✓ **echo \$env:JAVA\_HOME** → PowerShell



```

Administrateur : Windows PowerShell

PS C:\Windows\system32> echo $env:JAVA_HOME
C:\Program Files\Java\jdk-18.0.2.1
PS C:\Windows\system32>

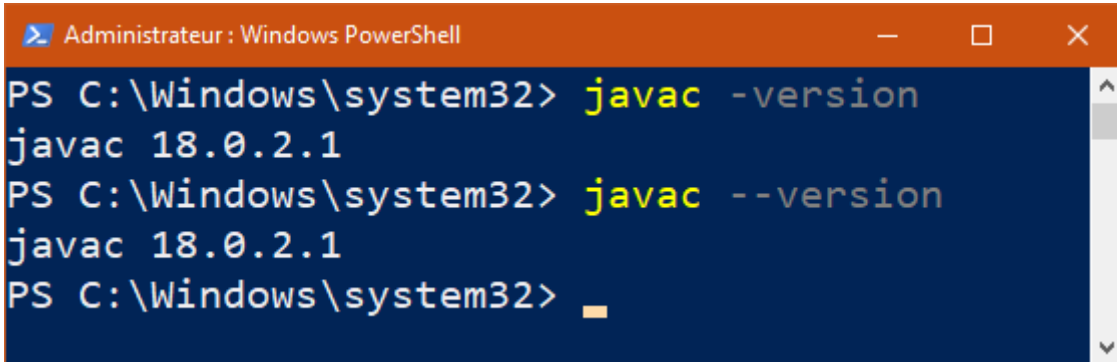
```

Commande 16 echo \$env:JAVA\_HOME

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

### 2.1.2.3.3. Vérifier la version du compilateur Java

Pour vérifier la version du compilateur Java, installé on utilisera la commande : **javac -version** ou bien : **javac --version**



```

PS C:\Windows\system32> javac -version
javac 18.0.2.1
PS C:\Windows\system32> javac --version
javac 18.0.2.1
PS C:\Windows\system32>

```

Commande 17: javac -version

### 2.1.2.4. Vérification avec un programme Java en ligne de commande

Il reste à tester le lancement du compilateur Java ; pour cela on va écrire une classe de test et travailler en CLI.

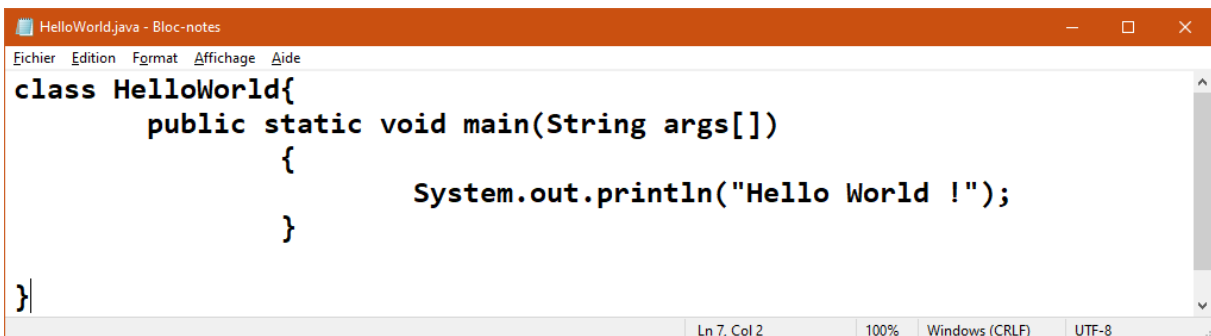
#### 2.1.2.4.1. Code de la classe HelloWorld

On ouvre Notepad et on écrit ces lignes de code dans un fichier nommé **HelloWorld.java** :

```

class HelloWorld{
    public static void main(String args[])
    {
        System.out.println("Hello World !");
    }
}

```



```

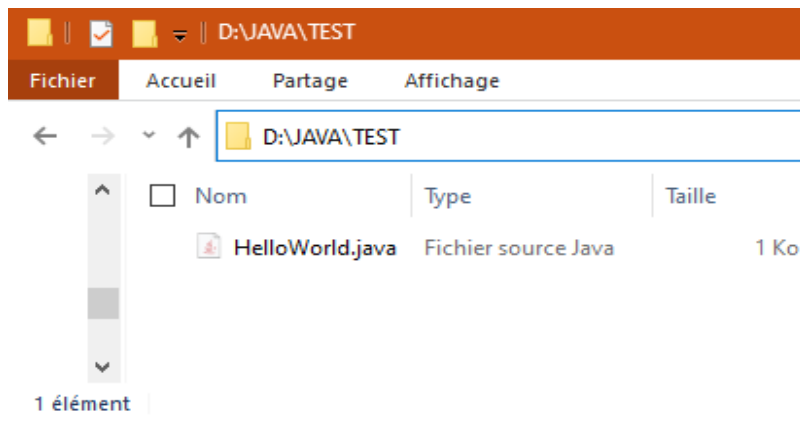
class HelloWorld{
    public static void main(String args[])
    {
        System.out.println("Hello World !");
    }
}

```

Screenshot 14: fichier HelloWorld.java

Le chemin du fichier est le suivant :

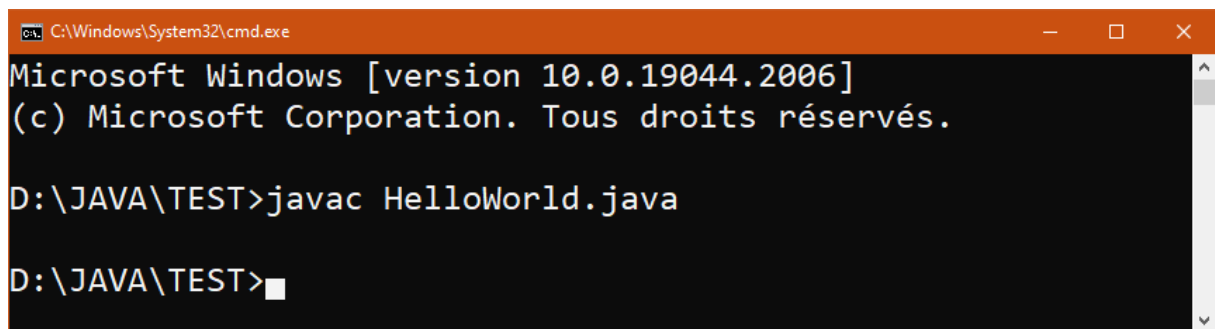
|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



Screenshot 15 : dossierTest

On se place dans ce dossier, on ouvre une invite de commandes et on compile le fichier en tapant la commande :

**`javac HelloWorld.java`**



Commande 18: `javac HelloWorld.java`

Cette compilation génère un fichier **`HelloWorld.class`**.




Screenshot 16 :dossierTest avec fichier .class

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

Ce fichier `.class`<sup>75</sup> renferme du code en langage machine pour le processeur virtuel Java. C'est un fichier de code compilé pour la JVM, mais cette machine virtuelle Java n'existe pas physiquement. On ne peut pas exécuter directement, ce code, avec le processeur, on doit passer par un émulateur : la JVM .

On donc va utiliser la JVM fournie par le JDK pour exécuter le code contenu dans le fichier **`HelloWorld.class`**.

Pour lancer la JVM, on reste dans le même dossier et on ouvre un shell et on entre la commande : **`java HelloWorld`**



```

C:\Windows\System32\cmd.exe
D:\JAVA\TEST>java HelloWorld
Hello World !

D:\JAVA\TEST>

```

Screenshot 17: commande `java HelloWorld`

Le programme s'exécute normalement.

Java est une technologie interopérable : on peut exécuter ce programme sur n'importe quel système d'exploitation, grâce à ce fichier `.class` (langage machine pour le processeur Java).

Si sur la machine cible un JRE (qui contient donc la JVM) a été installé, alors on pourra lancer la JVM qui exécutera ce fichier `.class`. Donc l'interopérabilité de Java passe par le JRE.

Après avoir installé et testé le JDK, il convient de choisir et de mettre en place les outils de développement.

## 2.2. Installation des outils de développement

Pour écrire du code en général et Java en particulier, outre le bloc-notes de Windows , on a le choix principalement, entre deux catégories d'outils : les éditeurs de code et les IDE.

<sup>75</sup> Ce fichier `.class` est l'équivalent du fichier `.o` ou `.obj` en C/C++.

|  |  |              |                  |
|--|--|--------------|------------------|
| <br>Nombre de pages :<br>71 | <b>Installer l'environnement de développement java</b> | Réalisé le : | 14-07-2022       |
|  |  | Modifié le : | 02-10-2022 19:21 |

### 2.2.1. Les éditeurs de code

Les éditeurs de code sont des logiciels conçus pour faciliter l'édition du code source. Ce sont en réalité des éditeurs de texte, classiques, auxquels ont été ajoutées des fonctionnalités complémentaires pour gérer et éditer le code source.

Un de leur avantage est qu'ils sont plus légers et flexibles que les IDE ; ce qui les destinent plutôt pour des petits projets.

Les caractéristiques principales de ces éditeurs de code sont :

- ✓ d'optimiser le temps d'écriture du code
- ✓ l'autocomplétion du code
- ✓ la coloration syntaxique
- ✓ l'indentation automatique
- ✓ extensibles à l'aide de plugins

#### 2.2.1.1. Les principaux éditeurs de code

Voici une sélection, non exhaustive, limitée aux éditeurs de code les plus utilisés par les développeurs, en environnement Windows.

Dans le tableau suivant la colonne « Dépôt code » indique si le code source est disponible sur un repository en accès libre ; quant à la colonne « SILL » elle précise si le logiciel fait partie du catalogue des logiciels libres du socle interministériel de logiciels libres<sup>76</sup>.

| Nom                | Concepteur        | License                                  | Écrit en  | Dépôt code | SILL |
|--------------------|-------------------|--|---|------------|------|
| Notepad++          | Don Ho            | Gratuitiel                               | C / C++   | OUI        | OUI  |
| Atom               | Github            | Gratuitiel                               | JavaScript, CoffeeScript, CSS, C++, HTML, Node.js | OUI        | NON  |
| Sublime Text       | Jon Skinner       | Harceliciel <sup>77</sup> / propriétaire | C++ et Python                                     | NON        | NON  |
| Bluefish           | Bluefish Dev Team | Freeware                                 | C   | OUI        | NON  |
| UltraEdit          | Ian D. Mea        | Propriétaire                             | ?   | NON        | NON  |
| Visual Studio Code | Microsoft         | Gratuitiel / propriétaire                | TypeScript et JavaScript                          | OUI        | NON  |
| VSCodium           | Microsoft         | Open source                              | TypeScript et JavaScript                          | OUI        | OUI  |

L'avantage principal de l'éditeur de code, c'est qu'il utilise beaucoup moins de ressources système qu'un EDI. On peut l'utiliser pour simplement écrire du code ou visualiser, rapidement, un fichier de code source.

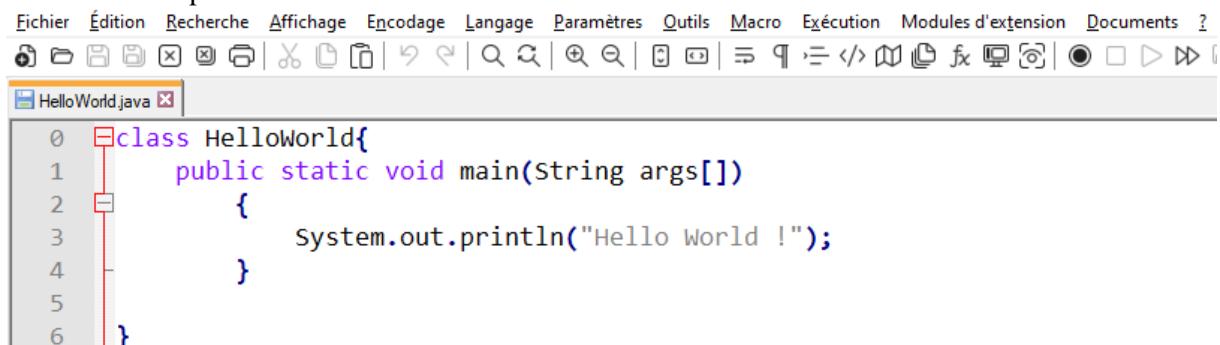
<sup>76</sup> <https://sill.etalab.gouv.fr/software>

<sup>77</sup> <https://fr.wikipedia.org/wiki/Harceliciel>

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

Voici un aperçu rapide pour chaque éditeur de code, consistant en un screenshot, de la fenêtre de code.

#### 2.2.1.1.1. Notepad++

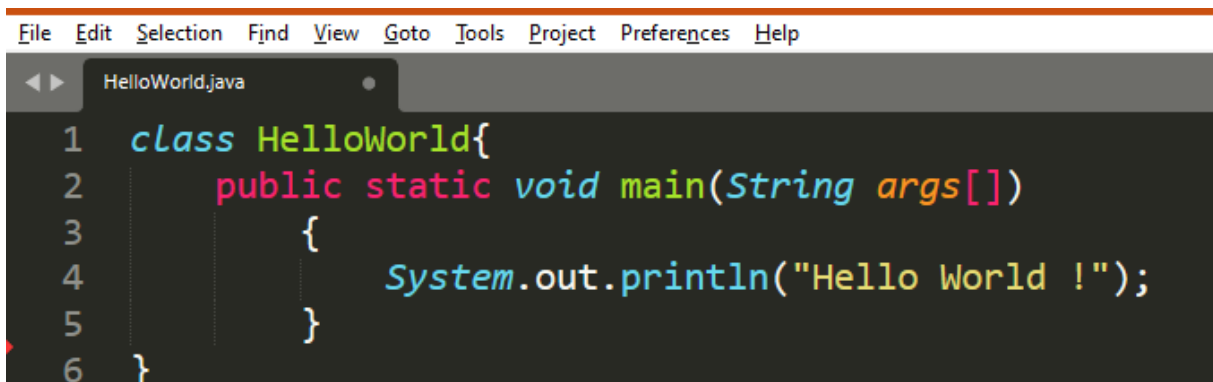


```

Fichier  Édition  Recherche  Affichage  Encodage  Langage  Paramètres  Outils  Macro  Exécution  Modules d'extension  Documents  ?
[Icons]
HelloWorld.java
0  class HelloWorld{
1      public static void main(String args[])
2      {
3          System.out.println("Hello World !");
4      }
5
6  }
```

#### 2.2.1.1.2. Atom

#### 2.2.1.1.3. Sublime Text



```

File  Edit  Selection  Find  View  Goto  Tools  Project  Preferences  Help
HelloWorld.java
1  class HelloWorld{
2      public static void main(String args[])
3      {
4          System.out.println("Hello World !");
5      }
6  }
```

#### 2.2.1.1.4. Bluefish

#### 2.2.1.1.5. UltraEdit

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

### 2.2.1.2. Visual Studio code et VSCodium

On peut envisager de coder en Java avec Visual Studio code, c'est parfaitement réalisable. Ceci est rendu possible grâce aux multiples extensions Java disponibles pour ce logiciel de l'entreprise de Redmond.

#### 2.2.1.2.1. Visual Studio Code

#### 2.2.1.2.2. VSCodium

### 2.2.2. Les IDE

## 2.3. Installation de l'IDE

Il faudra choisir un IDE pour le système d'exploitation de Microsoft et ensuite procéder à son installation.

.

► d'installation de l'IDE Eclipse en tant qu'administrateur

### 2.3.1. Les principaux IDE pour coder en Java

Les IDE majeurs pour coder en Java sont au nombre de trois :

- ✓ IntelliJ IDEA
- ✓ NetBeans
- ✓ Eclipse

### 2.3.2. Installation de l'IDE Eclipse

#### 2.3.2.1. Téléchargement de l'IDE

Choisir Java EE et ensuite on sélectionne la bonne perspective

#### 2.3.2.2. Téléchargement de l'IDE

Vérification checksum

#### 2.3.2.3. Configuration de l'IDE

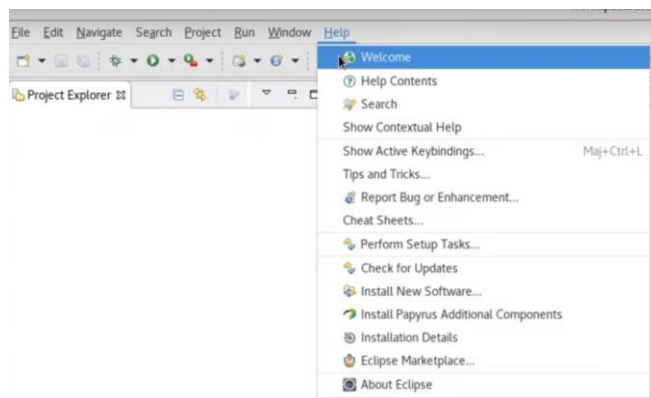
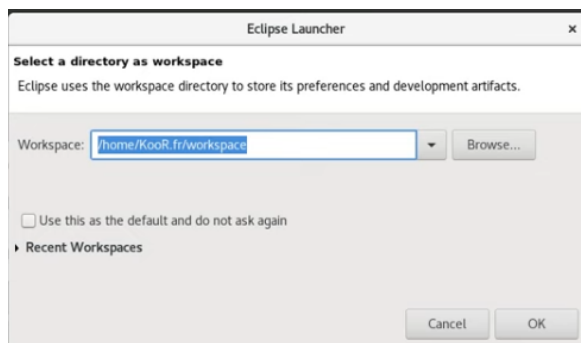


|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h1>Installer l'environnement de développement java</h1> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

## ► Eclipse

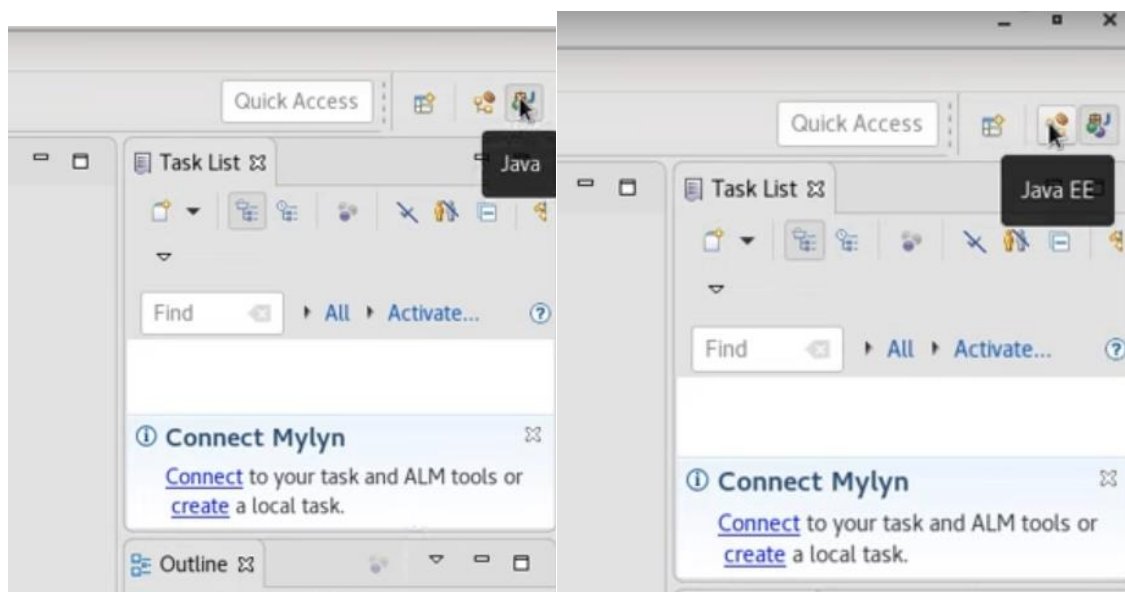
## ► Eclipse version JEE

## ► Workspace

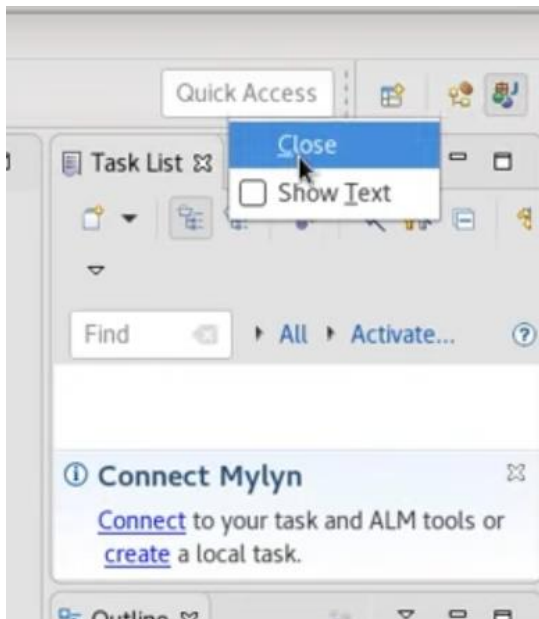


## -welcome page

## ► Perspectives



|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2 style="text-align: center;">Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



### ► 3 types de commentaires



### ► Portabilité du code java

Compilation javac

Exécution java

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <h2>Installer l'environnement de développement java</h2> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

### 3. Tests

Après avoir installé , il convient de vérifier le bon fonctionnement de l'installation.

#### 3.1. Bytecode

Décompilateur

**javap**

#### 3.2. Programme de test

#### 3.3. Junit

### 4. De Java 8 à Java 18 : 8 années de nouvelles fonctionnalités

Graphe versions : majeures / autres

[https://en.wikipedia.org/wiki/Java\\_version\\_history](https://en.wikipedia.org/wiki/Java_version_history)

Java 8, 11, 17

#### 4.1.1. Release table

| Version  | Release date  | End of Free Public Updates <sup>[3][8][9][10]</sup> | Extended Support Until |
|----------|---------------|---|------------------------|
| JDK Beta | 1995          | ?   | ?                      |
| JDK 1.0  | January 1996  | ?   | ?                      |
| JDK 1.1  | February 1997 | ?   | ?                      |

|  |  |              |                  |
|--|--|--------------|------------------|
| <br>Nombre de pages :<br>71 | <b>Installer l'environnement de développement java</b> | Réalisé le : | 14-07-2022       |
|  |  | Modifié le : | 02-10-2022 19:21 |

|                  |                |  |   |
|------------------|----------------|--|---|
| J2SE 1.2         | December 1998  | ?  | ?   |
| J2SE 1.3         | May 2000       | ?  | ?   |
| J2SE 1.4         | February 2002  | October 2008   | February 2013   |
| Java SE 5        | September 2004 | November 2009  | April 2015  |
| Java SE 6        | December 2006  | April 2013   | December 2018<br>December 2026 for Azul <sup>[11]</sup>   |
| Java SE 7        | July 2011      | July 2019  | July 2022   |
| Java SE 8 (LTS)  | March 2014     | <b>March 2022 for Oracle (commercial)</b><br>December 2030 for Oracle (non-commercial)   | December 2030 <sup>[13]</sup>                             |
|                  |                | December 2030 for Azul<br>May 2026 for IBM Semeru <sup>[12]</sup><br>At least May 2026 for Eclipse Adoptium<br>At least May 2026 for Amazon Corretto   |   |
| Java SE 9        | September 2017 | March 2018 for OpenJDK   | —   |
| Java SE 10       | March 2018     | September 2018 for OpenJDK   | —   |
| Java SE 11 (LTS) | September 2018 | September 2026 for Azul<br>October 2024 for IBM Semeru <sup>[12]</sup><br>At least October 2024 for Eclipse Adoptium<br>At least September 2027 for Amazon Corretto<br>At least October 2024 for Microsoft <sup>[14][15]</sup> | September 2026<br>September 2026 for Azul <sup>[11]</sup> |
|                  |                |  |   |
| Java SE 12       | March 2019     | September 2019 for OpenJDK   | —   |
| Java SE 13       | September 2019 | March 2020 for OpenJDK   | —   |
| Java SE 14       | March 2020     | September 2020 for OpenJDK   | —   |
| Java SE 15       | September 2020 | March 2021 for OpenJDK<br>March 2023 for Azul <sup>[11]</sup>  | —   |

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <b>Installer l'environnement de développement java</b> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |

|                  |                |  |  |
|------------------|----------------|--|--|
| Java SE 16       | March 2021     | September 2021 for OpenJDK   | —  |
| Java SE 17 (LTS) | September 2021 | September 2029 for Azul<br>At least September 2027 for Microsoft <sup>[14]</sup><br>At least September 2027 for Eclipse Adoptium | September 2029 or later<br>September 2029 for Azul |
| Java SE 18       | March 2022     | September 2022 for OpenJDK and Adoptium  | —  |
| Java SE 19       | September 2022 | March 2023 for OpenJDK   | —  |
| Java SE 20       | March 2023     | September 2023 for OpenJDK   | —  |
| Java SE 21 (LTS) | September 2023 | September 2028   | September 2031 <sup>[13]</sup>                     |

#### Legend:

Old version

Older version, still maintained

**Latest version**

Future release

[https://en.wikipedia.org/wiki/Java\\_version\\_history](https://en.wikipedia.org/wiki/Java_version_history)

## 5. Critiques de la technologie Java

## 6. Icones



|   |  |              |                  |
|---|--|--------------|------------------|
|  | <b>Installer l'environnement de développement java</b> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |



## 7. Links

### JAVA: INSTALLATION

- 

On aura besoin de trois prérequis

- 

Java: <https://www.java.com/fr/download/>

- 

Le JDK: <https://www.oracle.com/java/technologies/javase-downloads.html>

- 

Télécharger l'IDE que vous souhaitez

- 

Eclipse: <https://www.eclipse.org/downloads/>

- 

Netbeans : <https://netbeans.apache.org/download/index.html>

- 

Intelij : <https://www.jetbrains.com/fr-fr/idea/>

|   |  |              |                  |
|---|--|--------------|------------------|
|  | <b>Installer l'environnement de développement java</b> | Réalisé le : | 14-07-2022       |
| Nombre de pages :<br>71   |  | Modifié le : | 02-10-2022 19:21 |