

	<h1>Installer l'environnement de développement java</h1>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27

# Installer l'environnement de développement Java

Sous Windows

<b>1. PROLEGOMENES</b>	<b>2</b>
1.1. HERITAGE	3
1.2. COMPILATIONS D'UN PROGRAMME	3
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	4
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	8
1.3. LES CONCEPTS DE JVM, JRE ET JDK	9
1.3.1. <i>JVM</i>	9
1.3.1.1. Le principe de machine virtuelle	9
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	11
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>	15
1.5.2. <i>Architectures de processeurs</i>	16
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	17
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

	<h1>Installer l'environnement de développement java</h1>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27

En effet, dans cette situation 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. ....	22
1.6.1.3.2. Les technique pour remédier au Von Neumann bottleneck.....	23
1.6.1.4. Configuration professionnelle .....	24
<b>1.6.2. Connaître le type d'architecture installée sur la machine</b> .....	24
1.6.2.1. Program Files(x86) vs Program Files .....	26
<b>2. INSTALLATION</b> .....	26
<b>2.1. PROCEDURE D'INSTALLATION</b> .....	26
2.1.1. Téléchargement .....	26
2.1.2. Procédure.....	26
2.1.3. Installation .....	26
2.1.3.1. Installeur automatique.....	26
2.1.3.2. Vérifier Java_home.....	26
<b>2.1.4. Installation du Java SE</b> .....	29
<b>2.1.5. Sha256</b> .....	31
2.1.5.1. Programme en C .....	31
2.1.5.2. Checksum et Sha512 .....	31
2.1.6. Installation de l'IDE Eclipse .....	31
2.1.6.1. Configuration de l'environnement de développement.....	31
<b>3. TESTS</b> .....	31
3.1. BYTECODE.....	31
3.2. PROGRAMME DE TEST .....	32
3.3. JUNIT .....	32
<b>4. DE JAVA 8 A JAVA 18 : 8 ANNEES DE NOUVELLES FONCTIONNALITES</b> .....	32
4.1.1. Release table .....	32
<b>5. CRITIQUES DE LA TECHNOLOGIE JAVA</b> .....	34
<b>6. ICONES</b> .....	34
<b>7. LINKS</b> .....	34

## 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.

	<h2 style="text-align: center;">Installer l'environnement de développement java</h2>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27

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.

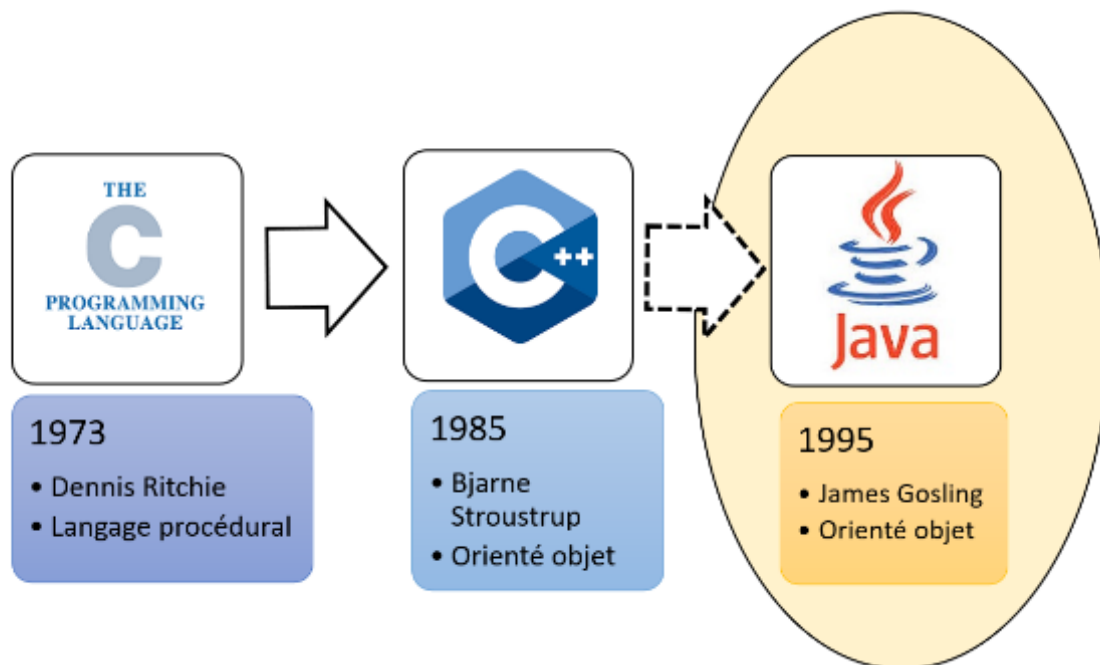


Figure 1 Timeline 3 langages

### 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.

<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 : 35		Modifié le :	18-08-2022 22:27

### 1.2.1. Compilation classique d'un programme

#### 1.2.1.1. La compilation classique

La compilation classique suit ce schéma :

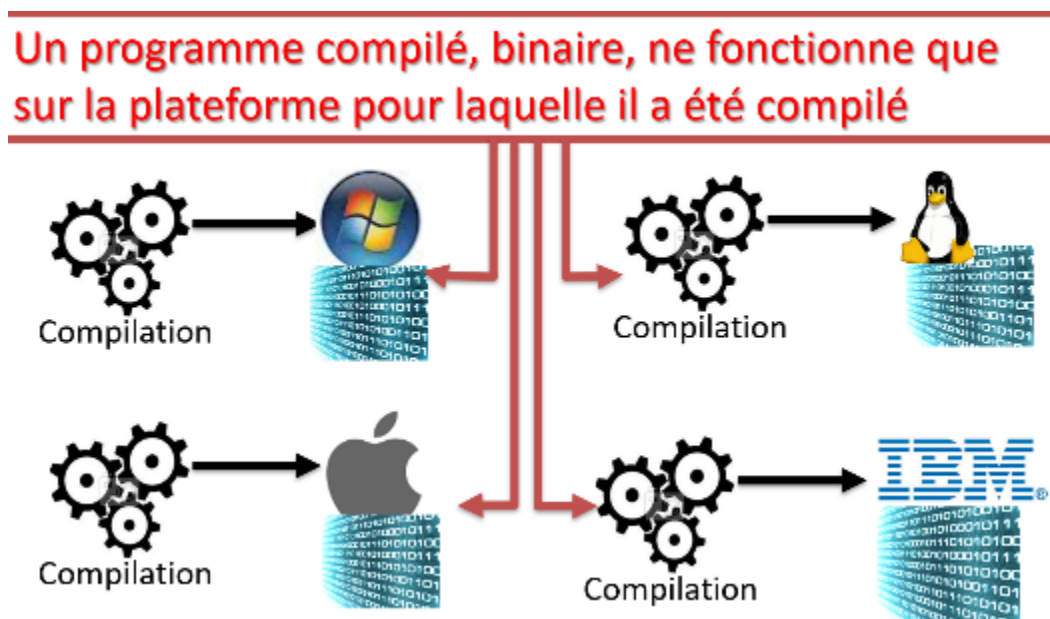


Figure 2 Compilation et exécutable

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

Voici un schéma qui synthétise la compilation en C :

	<h2 style="text-align: center;">Installer l'environnement de développement java</h2>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27



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++.

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

	<h2 style="text-align: center;">Installer l'environnement de développement java</h2>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27

### 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 une 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 :

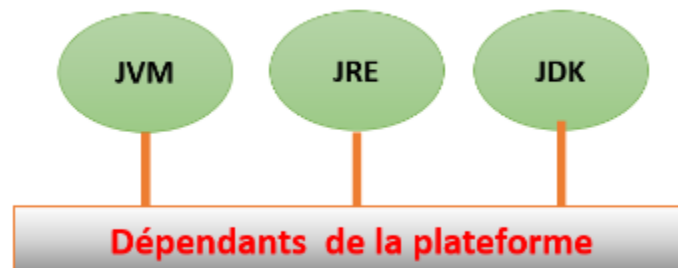


Figure 5 Compilation et interprétation en Java

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).

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

	<b>Installer l'environnement de développement java</b>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27



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.



*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 ».

<sup>4</sup> En abrégé : WOA

	<h1>Installer l'environnement de développement java</h1>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27



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.

#### 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 :

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



 Nombre de pages : 35	<b>Installer l'environnement de développement java</b>	Réalisé le :	14-07-2022
		Modifié le :	18-08-2022 22:27

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.

#### 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 :

<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 : 35		Modifié le :	18-08-2022 22:27



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).
  - 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.**

<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 : 35		Modifié le :	18-08-2022 22:27

### 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.

#### 1.3.3.1. Définition du JDK

Prenons la définition de Wikipédia :

<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 : 35		Modifié le :	18-08-2022 22:27



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 : 35		Modifié le :	18-08-2022 22:27

### 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 : 35		Modifié le :	18-08-2022 22:27

- 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 : 35		Modifié le :	18-08-2022 22:27

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>
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.

### 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».

<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

 Nombre de pages : 35	<b>Installer l'environnement de développement java</b>	Réalisé le :	14-07-2022
		Modifié le :	18-08-2022 22:27

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 concernant :  
la taille des registres généraux du processeur et la quantité de mémoire supportée.

Ce tableau reprend les principales caractéristiques des deux systèmes.

	<b>Système 32-bit</b>	<b>Système 64-bit</b>
<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

### 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.

<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>



	<h2 style="text-align: center;">Installer l'environnement de développement java</h2>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27

### 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>.

#### 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**.

<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-originel2.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 : 35		Modifié le :	18-08-2022 22:27

### 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.

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.

<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 : 35		Modifié le :	18-08-2022 22:27

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 : 35		Modifié le :	18-08-2022 22:27



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
Amazon Corretto	x64
Azul Zulu	x86; x64; AArch64
IBM	x64
Microsoft	x64; AArch64

 Nombre de pages : 35	<b>Installer l'environnement de développement java</b>	Réalisé le :	14-07-2022
		Modifié le :	18-08-2022 22:27

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/ Ryzen 9	Ryzen 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 : 35		Modifié le :	18-08-2022 22:27

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 de la machine.

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

En effet, dans cette situation 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 : 35		Modifié le :	18-08-2022 22:27



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.

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

Pour y remédier on peut appliquer les techniques les plus courantes :

#### Von neuman botlneck

<https://itigic.com/fr/von-neumann-architecture-characteristics-and-limitations/>

<file:///C:/Users/Djamel/Downloads/Cours1%20Architecture%20des%20Ordinateurs.pdf>

Le mot *informatique* vient de la contraction des mots **information** et **automatique**.



	<h2 style="text-align: center;">Installer l'environnement de développement java</h2>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27

Information + atomique = informatique

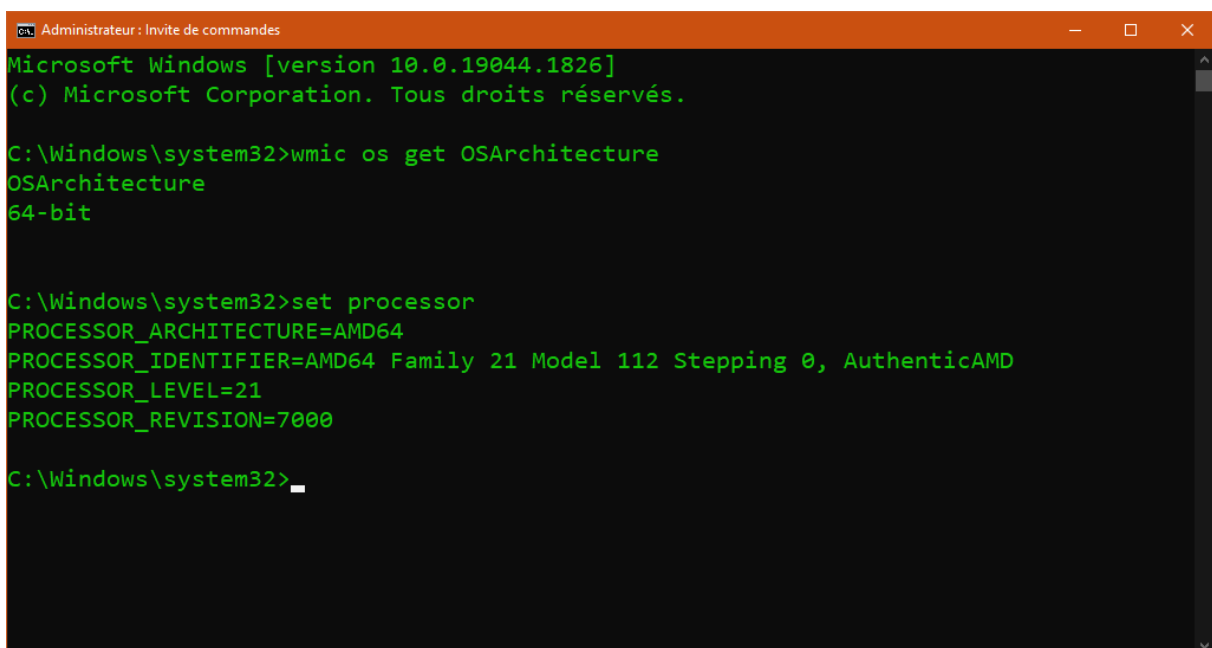
1.6.1.4. *Configuration professionnelle*  
Multithreading.

Programmation concurrentielle => bi-proc.

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

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

Sous Windows, plusieurs possibilités :



```

Administrateur : Invite de commandes

Microsoft Windows [version 10.0.19044.1826]
(c) Microsoft Corporation. Tous droits réservés.

C:\Windows\system32>wmic os get OSArchitecture
OSArchitecture
64-bit

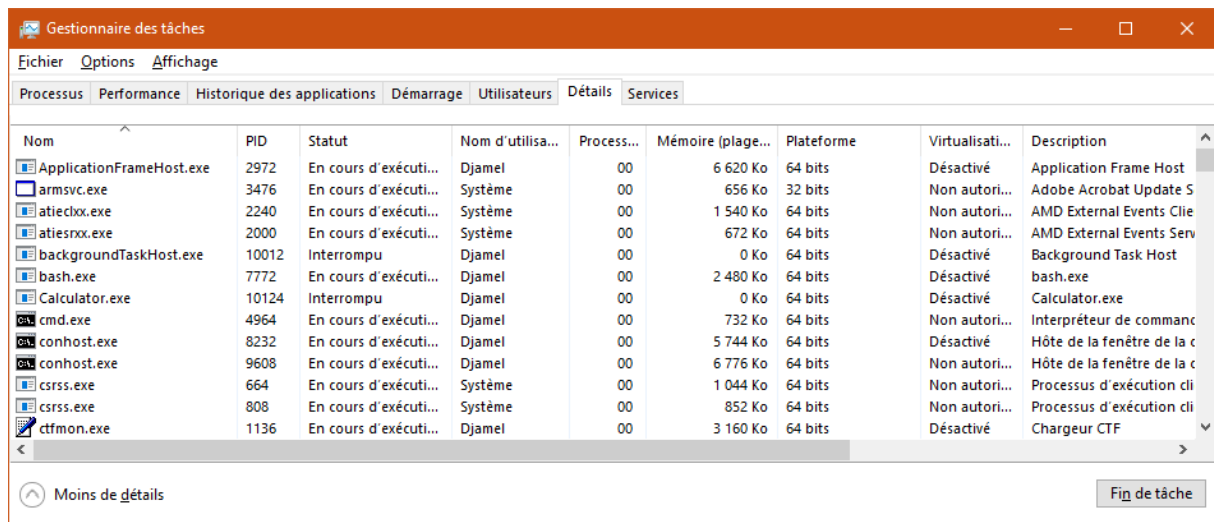
C:\Windows\system32>set processor
PROCESSOR_ARCHITECTURE=AMD64
PROCESSOR_IDENTIFIER=AMD64 Family 21 Model 112 Stepping 0, AuthenticAMD
PROCESSOR_LEVEL=21
PROCESSOR_REVISION=7000

C:\Windows\system32>_

```



	<h2 style="text-align: center;">Installer l'environnement de développement java</h2>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27



```

PS C:\> # method 4 - System.IntPtr type
PS C:\> if([System.IntPtr]::Size -eq 4){
>>     "32-bit"
>> }else{
>>     "64-bit"
>> }
64-bit
PS C:\>

```

- Go to Registry Editor
- Navigate to the following location:
- **HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment**
- In Right Pane, you will see a registry entry; **PROCESSOR\_ARCHITECTURE**. This entry stores either x86 or x64.

### ► commandes

Microsoft Windows [version 10.0.19044.1826]  
(c) Microsoft Corporation. Tous droits réservés.

	<h2 style="text-align: center;">Installer l'environnement de développement java</h2>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27

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

C:\Windows\system32>
```

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

### 1.6.2.1. Program Files(x86) vs Program Files

## 2. Installation

### 2.1. Procédure d'installation

#### 2.1.1. Téléchargement

Choisir Java EE et ensuite on sélectionne la bonne perspective

#### 2.1.2. Procédure

#### 2.1.3. Installation

##### 2.1.3.1. Installeur automatique

##### 2.1.3.2. Vérifier Java\_home

► environnement de développement java



► pas le jre , car il contient le jdk

► **java 1.8.0 = Java SE 8**

	<h2 style="text-align: center;">Installer l'environnement de développement java</h2>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27

► java 1.8.0 => système de versionning (officiel) utilisé dans la doc Oracle ou Sun

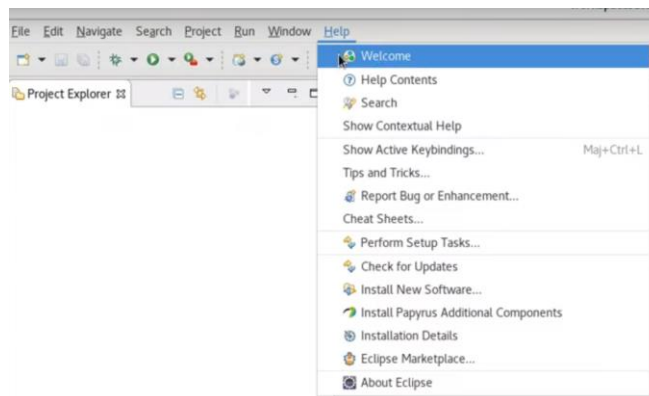
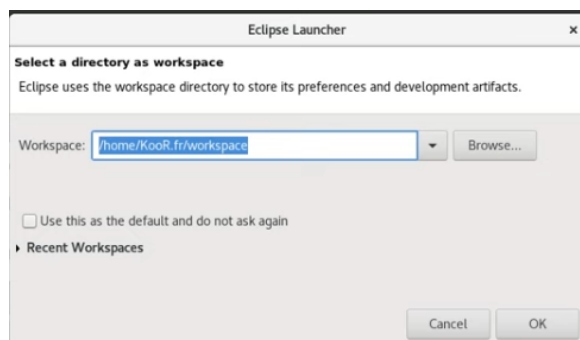
► openjdk et java pour system Unix like

```
openjdk version "1.8.0_121"
OpenJDK Runtime Environment (build 1.8.0_121-b14)
OpenJDK 64-Bit Server VM (build 25.121-b14, mixed mode)
```

► Eclipse

► Eclipse version JEE

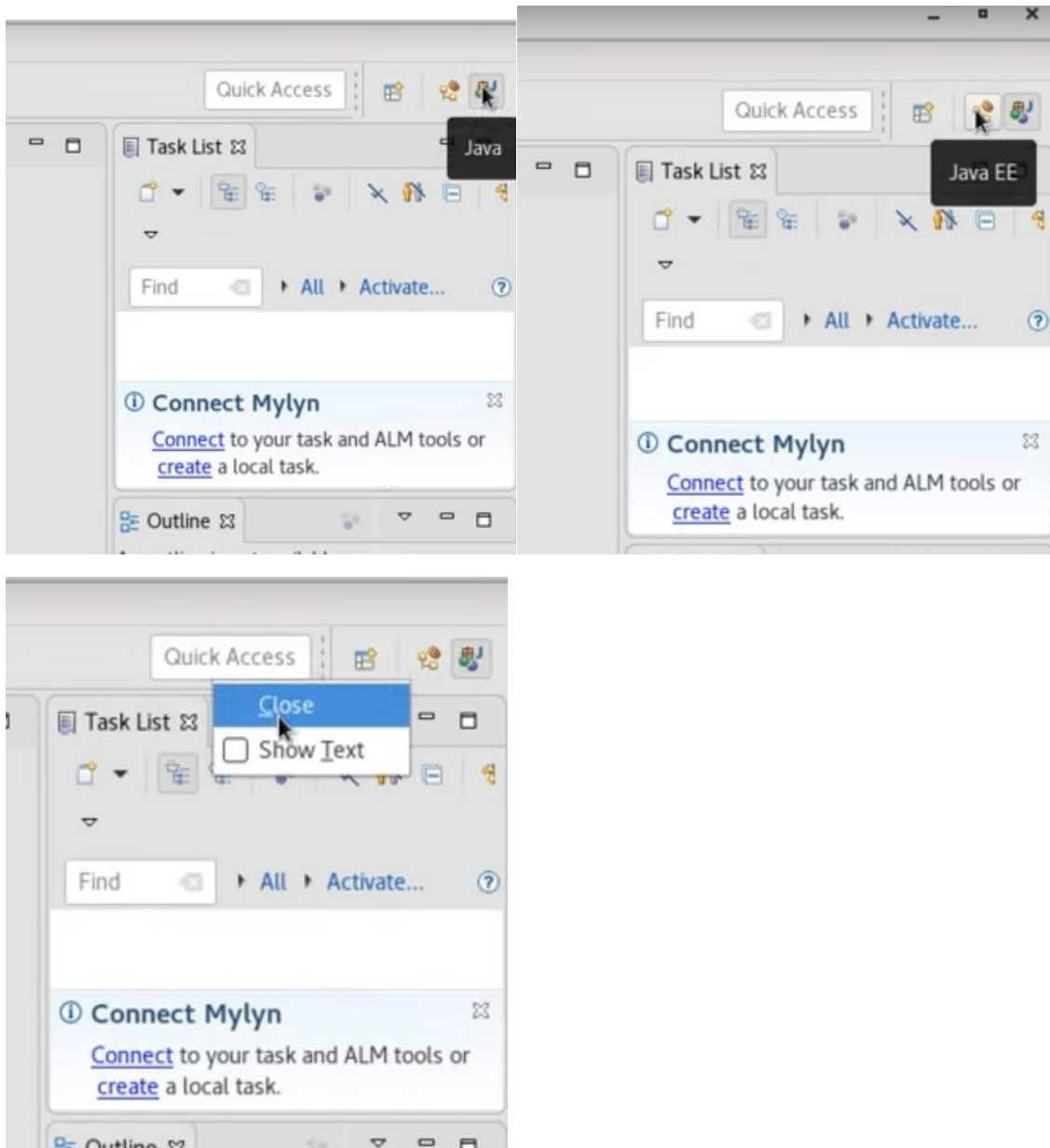
► Workspace



-welcome page

► Perspectives

	<h1>Installer l'environnement de développement java</h1>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27



► 3 types de commentaires

	<b>Installer l'environnement de développement java</b>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27

```

1
2 public class Start {
3
4
5     /*
6      * commentaire
7      */
8
9     /**
10      *
11      * @param args
12      */
13
14
15     public static void main(String[] args) {
16
17         int a = 0; // Ceci est une déclaration

```

► Portabilité du code java

Compilation javac

Exécution java

#### 2.1.4. Installation du Java SE

[https://www.youtube.com/watch?v=y\\_Ye9-s7h1w](https://www.youtube.com/watch?v=y_Ye9-s7h1w)

	<h2 style="text-align: center;">Installer l'environnement de développement java</h2>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27

## Download & Install Oracle JDK 18

- Oracle JDK 18 is regular update (Non-LTS) of Java SE
- Released on March, 22<sup>nd</sup> 2022
- Download page:  
<https://www.oracle.com/java/technologies/downloads/>
- Verify SHA256 checksum after download
- Installer automatically updates PATH environment variable

Transcription

0:56 installer automatically updates the path

0:58 environment variable that means the

1:01 installation of oracle jdk 18 will be

1:04 quick and easy

1:06 all right lets head over to the

1:08 homepage of oracle jdk download at the

1:12 url oracle.com

1:15 java slash technologies slash downloads

1:20 okay you see this is the official

1:22 download based on oracle jdk 18

1:26 and for

1:28 installer for windows click windows tab

1:31 here and you can download the installer

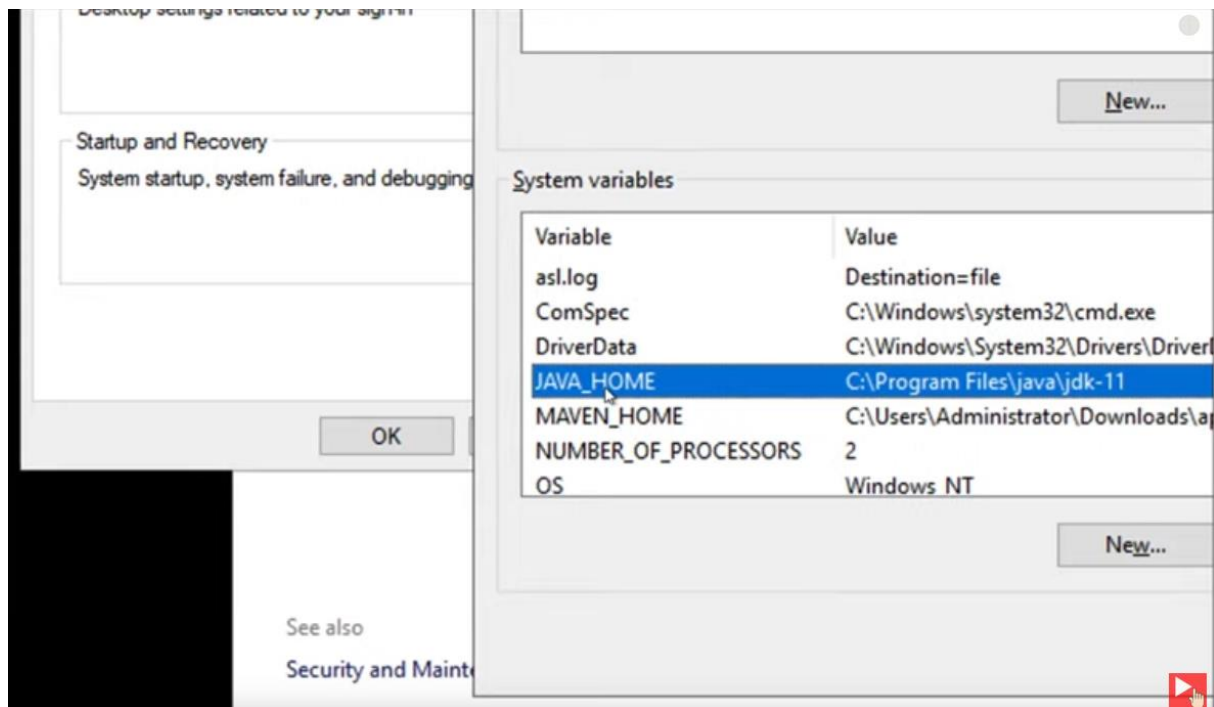
1:35 exit for installer x64 msi installer

1:39 here i choose download the x64 msi

1:42 installer so click the download link

Anglais (généré automatiquement)

<https://www.youtube.com/watch?v=CluB3qwLdbk>



Mm

Step by step

	<h2 style="text-align: center;">Installer l'environnement de développement java</h2>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27

<https://www.youtube.com/watch?v=CluB3qwLdbk>

On va installer Oracle JDK, puisque c'est à des fins de développement.

Téléchargement

Certutil

2.1.5. **Sha256**

PowerShell

Pour comparer le Sha256 => faire un programme en C pour comparer 2 chaînes de caractères.

**2.1.5.1. Programme en C**

**2.1.5.2. Checksum et Sha512**

F41047493698E0E77D6AA88D490656998EE4FD634776B040CA60C69621A36C8FFB02D36579941554  
75A09774839F030094B043B28B3132B6ED56831B05FD73F8

2.1.6. Installation de l'IDE Eclipse

*2.1.6.1. Configuration de l'environnement de développement*

## 3. Tests

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

3.1. Bytecode

Décompilateur

 Nombre de pages : 35	<b>Installer l'environnement de développement java</b>	Réalisé le :	14-07-2022
		Modifié le :	18-08-2022 22:27

*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 <a href="#">[3][8][9][10]</a>	Extended Support Until
JDK Beta	1995	?	?
JDK 1.0	January 1996	?	?
JDK 1.1	February 1997	?	?
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 December 2026 for Azul <a href="#">[11]</a>
Java SE 7	July 2011	July 2019	July 2022



	<h2>Installer l'environnement de développement java</h2>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27

Java SE 8 (LTS)	March 2014	<b>March 2022 for Oracle (commercial)</b> December 2030 for Oracle (non-commercial) December 2030 for Azul May 2026 for IBM Semeru <sup>[12]</sup> At least May 2026 for Eclipse Adoptium At least May 2026 for Amazon Corretto	
		December 2030 <sup>[13]</sup>	
		March 2018 for OpenJDK	
		—	
		September 2018 for OpenJDK	
Java SE 11 (LTS)	September 2018	September 2026 for Azul October 2024 for IBM Semeru <sup>[12]</sup> At least October 2024 for Eclipse Adoptium At least September 2027 for Amazon Corretto At least October 2024 for Microsoft <sup>[14][15]</sup>	
		September 2026 September 2026 for Azul <sup>[11]</sup>	
		September 2019 for OpenJDK	
		—	
		March 2020 for OpenJDK	
Java SE 14	March 2020	September 2020 for OpenJDK	
		—	
		March 2021 for OpenJDK	
		—	
		March 2023 for Azul <sup>[11]</sup>	
Java SE 16	March 2021	September 2021 for OpenJDK	
		—	
		September 2029 for Azul At least September 2027 for Microsoft <sup>[14]</sup> At least September 2027 for Eclipse Adoptium	
		September 2029 or later September 2029 for Azul	
		September 2022 for OpenJDK and Adoptium	
Java SE 18	March 2022	—	
		—	
		March 2023 for OpenJDK	
		—	
		September 2023 for OpenJDK	
Java SE 20	March 2023	—	
		—	
		—	
		—	
		—	

	<h2>Installer l'environnement de développement java</h2>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27

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



## 7. Links

### JAVA: INSTALLATION

- 

On aura besoin de trois prérequis

-

	<b>Installer l'environnement de développement java</b>	Réalisé le :	14-07-2022
Nombre de pages : 35		Modifié le :	18-08-2022 22:27

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/>