**Installer l’environnement de développement java**

[1. Prolégomènes 1](#_Toc109411334)

[1.1. Héritage 1](#_Toc109411335)

[1.2. Compilations d’un programme 2](#_Toc109411336)

[1.2.1. Compilation classique d’un programme 2](#_Toc109411337)

[1.2.1.1. La compilation classique 2](#_Toc109411338)

[1.2.1.2. La compilation avec l’exemple du C 3](#_Toc109411339)

[1.2.2. Java est un langage compilé et interprété 4](#_Toc109411340)

[1.3. Les concepts de JVM, JRE et JDK 8](#_Toc109411341)

[1.3.1. Focus sur la JVM 9](#_Toc109411342)

[1.4. Historique des versions de Java 9](#_Toc109411343)

[1.4.1. Release table 9](#_Toc109411344)

[2. Installation 11](#_Toc109411345)

[2.1. Prérequis hardware 11](#_Toc109411346)

[2.1.1. Les trois types configurations hardware 11](#_Toc109411347)

[3. Tests 12](#_Toc109411348)

[3.1. Bytecode 12](#_Toc109411349)

[4. Links 12](#_Toc109411350)

# 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[[1]](#footnote-1) ; cette entreprise a été rachetée par Oracle en 2009. Java est devenu donc propriété d’Oracle Corporation.

## Héritage

Java est un langage de programmation multiplateformes[[2]](#footnote-2), il dérive des langages C et C++, dont il reprend, en partie, la syntaxe.



Figure Timeline 3 langages

## 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.

### Compilation classique d’un programme

#### La compilation classique

La compilation classique suit ce schéma :



Figure Compilation et exécutable

#### La compilation avec l’exemple du C

Voici un schéma qui synthétise la compilation en C :



Figure 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 Composition d'un programme Java

### Java est un langage compilé et interprété

<https://www.sciencedirect.com/topics/engineering/interpreted-language>

**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)[[3]](#footnote-3)*.

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 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).



Dépendants de chaque plateforme, en raison du type de système d’exploitation de la machine et/ou de processeur sous-jacent.

Figure Dépendance de la plateforme

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



Figure Indépendance de la plateforme de Java

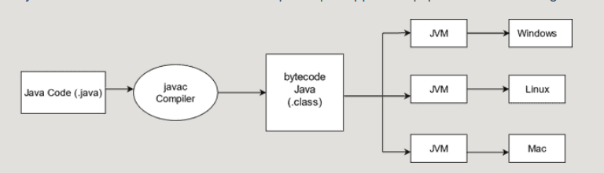
C’est l’approche « write once and run anywhere » [[4]](#footnote-4), en français « écrire une fois et exécuter partout ».



Figure write once and run anywhere (WORA)

…///…

Figure le concept de WOA



## Les concepts de JVM, JRE et JDK

En effet, il convient de présenter les notions de JRE, JVM et JDK.

Java reprend la syntaxe du C.

### Focus sur la JVM



## Historique des versions de Java

<https://en.wikipedia.org/wiki/Java_version_history>

Java 8, 11, 17

### Release table

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Release date** | **End of Free Public Updates**[**[3]**](https://en.wikipedia.org/wiki/Java_version_history#cite_note-auto9-3)[**[8]**](https://en.wikipedia.org/wiki/Java_version_history#cite_note-8)[**[9]**](https://en.wikipedia.org/wiki/Java_version_history#cite_note-9)[**[10]**](https://en.wikipedia.org/wiki/Java_version_history#cite_note-10) | **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[[11]](https://en.wikipedia.org/wiki/Java_version_history#cite_note-Azul-11) |
| Java SE 7 | July 2011 | July 2019 | July 2022 |
| Java SE 8 (LTS) | March 2014 | **March 2022 for Oracle (commercial)** December 2030 for Oracle (non-commercial) December 2030 for Azul May 2026 for IBM Semeru[[12]](https://en.wikipedia.org/wiki/Java_version_history" \l "cite_note-IBM_Semeru-12) At least May 2026 for Eclipse Adoptium At least May 2026 for Amazon Corretto | December 2030[[13]](https://en.wikipedia.org/wiki/Java_version_history#cite_note-:1-13) |
| 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 October 2024 for IBM Semeru[[12]](https://en.wikipedia.org/wiki/Java_version_history" \l "cite_note-IBM_Semeru-12) At least October 2024 for Eclipse Adoptium At least September 2027 for Amazon Corretto At least October 2024 for Microsoft[[14]](https://en.wikipedia.org/wiki/Java_version_history#cite_note-Microsoft"-14)[[15]](https://en.wikipedia.org/wiki/Java_version_history#cite_note-15) | September 2026 September 2026 for Azul[[11]](https://en.wikipedia.org/wiki/Java_version_history#cite_note-Azul-11) |
| 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 March 2023 for Azul[[11]](https://en.wikipedia.org/wiki/Java_version_history#cite_note-Azul-11) | — |
| Java SE 16 | March 2021 | September 2021 for OpenJDK | — |
| Java SE 17 (LTS) | September 2021 | September 2029 for Azul At least September 2027 for Microsoft[[14]](https://en.wikipedia.org/wiki/Java_version_history#cite_note-Microsoft"-14) At least September 2027 for Eclipse Adoptium | September 2029 or later 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[[13]](https://en.wikipedia.org/wiki/Java_version_history#cite_note-:1-13) |
| **Legend:**  Old version  Older version, still maintained  **Latest version**  Future release | | | |

<https://en.wikipedia.org/wiki/Java_version_history>

# Installation

## Prérequis hardware

Si l’on veut pouvoir faire du développement, dans n’importe quel langage, il faut disposer d’un ordinateur en conséquence.

En matière de développement, on peut considérer trois types configurations matérielles.

### Les trois types configurations hardware

# Tests

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

## Bytecode

Décompilateur

***javap***

# 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/

1. https://fr.wikipedia.org/wiki/Sun\_Microsystems [↑](#footnote-ref-1)
2. Logiciels(OS) ou matériels(x86,ARM). [↑](#footnote-ref-2)
3. <https://www.ibm.com/docs/fr/sdk-java-technology/8?topic=reference-jit-compiler> [↑](#footnote-ref-3)
4. En abrégé : WOA [↑](#footnote-ref-4)