

Compilation Java & Maven

Cours de POO avec Java

Kevin Heirich (A403)

HEPIA

22.09.2023

Compilation Java

JRE, JDK, JVM

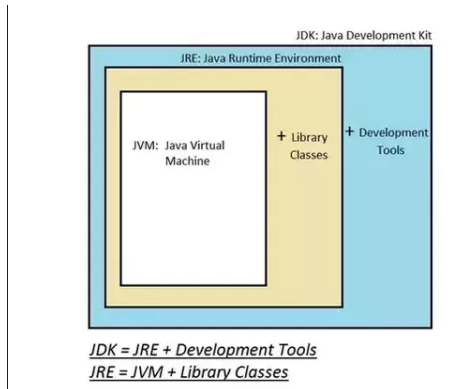


Figure 1: JRE, JDK, JVM

Compilation simple : Javac

```
javac Main.java
```

```
// Javac compile le fichier.java  
// Cette commande génère un fichier .class  
// Nous executons notre code avec
```

```
java Main
```

Hello World!

// Le fichier doit porter le nom de la classe !

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

La version de java

Pensez à vérifier votre version de Java

```
java -version
```

- La dernière LTS est la version 21
- La version 17 fonctionnera aussi pour ce cours (LTS précédente)

Maven

Pourquoi Maven ?

- Maven est un gestionnaire de projet
 - Compilation multi-fichiers
 - Tests unitaires
 - Installation de librairies
 - Production d'exécutables

- Maven utilise un fichier `POM.xml` pour définir la structure, et les librairies à employer
- Le dossier `src` contient le code source & les tests
- Le dossier `target` contient les fichiers de sortie comme le `.jar`

Installation 1/3

L'archive de maven est téléchargée sur le site d'oracle

```
tar zxvf apache-maven-3.8.6.tar.gz  
mv apache-maven-3.8.6/ /usr/local/  
export PATH=/usr/local/apache-maven-3.8.6/bin:$PATH
```

Ne copiez pas *bêtement* ces lignes si le nom du fichier est différent

Installation 2/3

Vérifiez la bonne installation en vérifiant que les versions concordent

```
mvn --version
```

```
Apache Maven 3.8.6
```

```
Maven home: /usr/local/apache-maven-3.8.6
```

```
Java version: 18.0.2.1
```

Si la version de `maven` ou de `java` est incorrecte, vos variable d'environnement sont peut-être fausses

La variable `JAVA_HOME` doit pointer sur votre installation Java.

La variable `PATH` doit bien inclure le chemin vers l'installation de `maven`

Ces variables doivent être incluses dans votre fichier `.bashrc` avec la commande `export`

Les commandes de base

// Compiler du code & installer les dépendances (génère un dossier target)

```
mvn compile
```

// Exécuter les tests

```
mvn test
```

// Exécuter le programme

```
mvn exec:java
```

// Nettoyer le dossier target

```
mvn clean
```

Demonstration live