

# Fichiers JAR

# CLASS\_PATH

Philippe Genoud

*Philippe.Genoud@univ-grenoble-alpes.fr*



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).

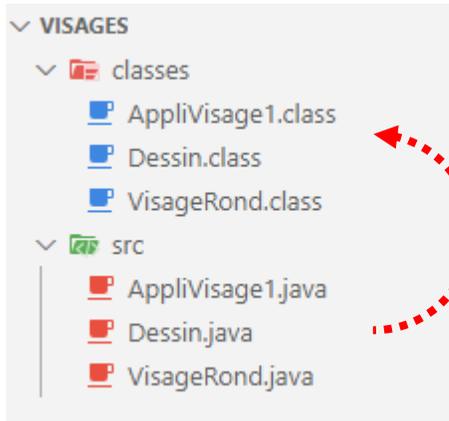
# Qu'est-ce qu'un fichier JAR ?

- fichier JAR (Java ARchive) fichier compressé utilisé pour regrouper en un seul fichier
  - plusieurs fichiers de classe Java (`.class`),
  - des métadonnées
  - des ressources (comme des images et des fichiers de configuration)
- Point clés :
  - **Format** : un fichier JAR utilise le format de fichier ZIP.
  - **Utilisation** : couramment utilisés pour faciliter la distribution des bibliothèques et des applications Java.
  - **Manifest** : un fichier JAR contient un fichier manifeste (META-INF/MANIFEST.MF) qui peut spécifier des informations comme le point d'entrée de l'application.
  - **Exécution** : Les fichiers JAR peuvent être exécutés directement si le manifeste spécifie la classe principale.



# Fichier JAR exemple

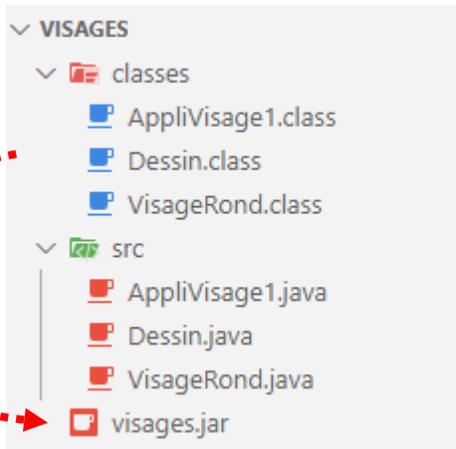
Code compilé  
(byte code)



Code source

Compilation

```
PS C:\Philippe\Visages\src> javac -d ../classes AppliVisage1.java
```



Création du fichier jar

**c** : create  
**v** : verbose  
**f** : file

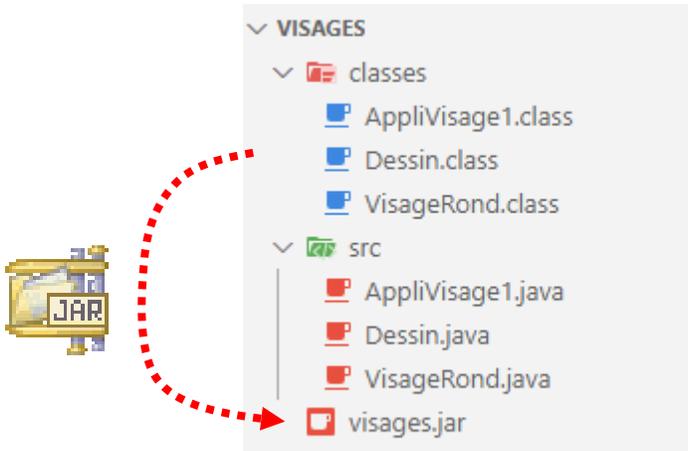
fichier JAR  
à créer

**-C chemin** change de répertoire avant  
d'ajouter des fichiers au fichier JAR

```
PS C:\Philippe\Visages> jar cvf visages.jar -C ./classes .  
added manifest  
adding: AppliVisage1.class(in = 926) (out= 645)(deflated 30%)  
adding: Dessin.class(in = 1383) (out= 853)(deflated 38%)  
adding: VisageRond.class(in = 2688) (out= 1323)(deflated 50%)
```

Fichiers à ajouter :  
ici tout le contenu  
du répertoire  
**classes**

# Fichier JAR exemple



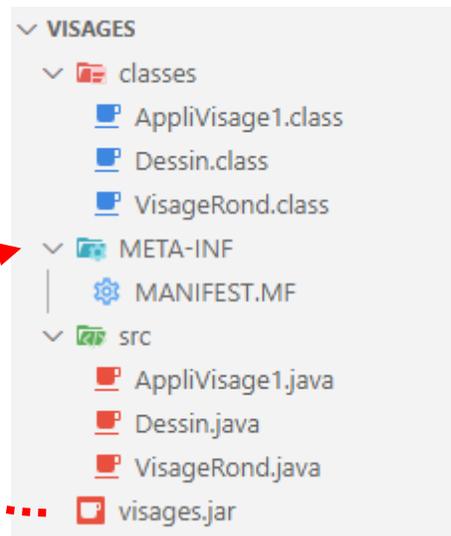
Voir le contenu d'un fichier JAR

t : table

```
PS C:\Philippe\Visages> jar tvf visages.jar
 0 Thu Dec 12 08:31:12 CET 2024 META-INF/
66 Thu Dec 12 08:31:12 CET 2024 META-INF/MANIFEST.MF
926 Thu Dec 12 08:20:34 CET 2024 AppliVisage1.class
1383 Thu Dec 12 08:20:34 CET 2024 Dessin.class
2688 Thu Dec 12 08:20:34 CET 2024 VisageRond.class
PS C:\Philippe\Visages>
```

Répertoire pour les méta données

Fichier manifeste

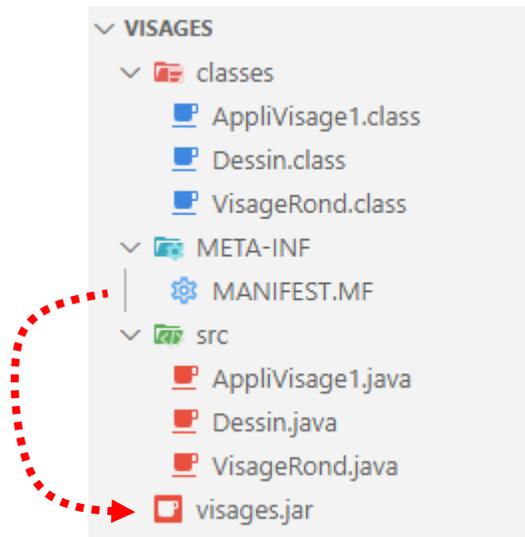


Extraire le contenu du manifeste

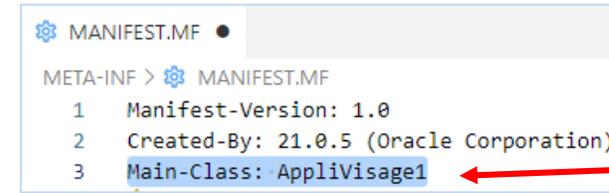
x : extract

```
PS C:\Philippe\Visages> jar xvf visages.jar META-INF/MANIFEST.MF
inflated: META-INF/MANIFEST.MF
PS C:\Philippe\Visages>
```

# Fichier JAR exemple



Modifier le manifeste



La classe contenant le programme principal

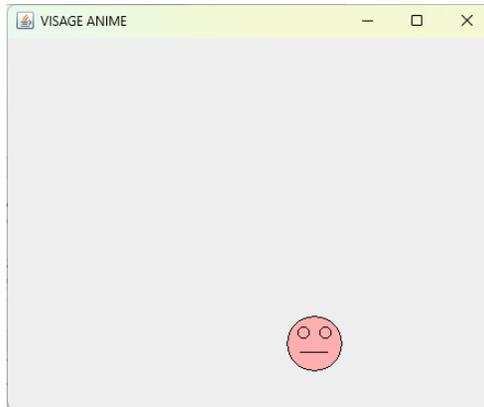
Mettre à jour le JAR avec le nouveau MANIFEST.MF

u : update

```
PS C:\Philippe\Visages> jar ufm visages.jar META-INF/MANIFEST.MF
```

Exécuter le JAR

```
PS C:\Philippe\Visages> java -jar visages.jar
```



# Classpath

- **Classpath** : paramètre utilisé par **javac** le compilateur Java et **java** la machine virtuelle Java (JVM) pour indiquer où trouver les fichiers de classe (fichiers **.class**) et les bibliothèques (fichiers **.jar**) nécessaires à l'exécution d'un programme Java.
- Peut être défini de plusieurs manières :
  - Via la ligne de commande : en utilisant l'option **-classpath** ou **-cp** lors de l'exécution de la commande **java** ou **javac**.
  - Via une variable d'environnement : en définissant la variable d'environnement **CLASSPATH**.
  - Via un fichier de configuration : dans certains environnements de développement intégrés (IDE) comme Eclipse ou IntelliJ IDEA.
- Exemple de définition du classpath via la ligne de commande :

```
java -cp .;lib/mylibrary.jar com.example.MyClass
```

Si la classe est définie dans un package, il faut utiliser un nom complètement qualifié (*Fully Qualified Name*) et préfixant le nom de la classe par son nom de package

- le classpath inclut le répertoire courant (**.**) et le fichier **mylibrary.jar** situé dans le répertoire **lib**