

JDK

Alexandre Clenet

Quelle est la version de la plateforme Java installée sur votre poste ?

```
java -version

java version "20.0.2" 2023-07-18
Java(TM) SE Runtime Environment (build 20.0.2+9-78)
Java HotSpot(TM) 64-Bit Server VM (build 20.0.2+9-78, mixed mode, sharing)
```

Où est-elle?

```
which java

/c/Program Files/Java/jdk-20/bin/java
```

A quoi ressemble le bytecode?

```
javac Exemple1.java

-Création de Exemple1.class
```

```
less Exemple1.class

<CA><FE><BA><BE>^@^@^@^@I
^@^B^@^C^G^@^D^L^@^E^@^F^A^@^Pjava/lang/Object^A^@^F<init>^A^@^C()V      ^@^H^@
^G^@
^L^@^K^@^L^A^@^HExemple1^A^@^Ax^A^@^AI
^@^H^@^N^L^@^E^@^O^A^@^D(I)V
^@^Q^@^R^G^@^S^L^@^T^@^U^A^@^Pjava/lang/System^A^@^Cout^A^@^ULjava/io/PrintStream;
...
```

```
xxd Exemple1.class

00000000: cafe babe 0000 0040 0049 0a00 0200 0307  ....@.I.....
00000010: 0004 0c00 0500 0601 0010 6a61 7661 2f6c  ....java/l
00000020: 616e 672f 4f62 6a65 6374 0100 063c 696e  ang/Object...<in
...
```

```
javap -c -s Exemple1.class | less

Compiled from "Exemple1.java"
public class Exemple1 {
    public Exemple1(int);
        descriptor: (I)V
        Code:
            0: aload_0
            1: invokespecial #1           // Method java/lang/Object."<init>":
()V
            4: aload_0
            5: iload_1
            6: putfield      #7           // Field x:I
            9: return

    public int compute(int);
        descriptor: (I)I
        Code:
            0: aload_0
            1: getfield      #7           // Field x:I
            4: iload_1
            5: iadd
            6: ireturn
    ...
}
```

Le main de cette classe n'en fait rien. Comprenez sa (basique) intention et corrigez le.

```
Exemple1 e1 = new Exemple1(1);
    System.out.println("The result is " + e1.compute(5));
    new Scanner(System.in).nextLine();
    System.exit(1);
```

```
EXEC :
java Exemple1.java
The result is 6
```

Exercice Package

```
IUT3\Build\jdk\fr\nantes\iut\lpmiar> java .\Exemple1.java
The result is 6
```

Exercice

Donnez les classpath pour les ensembles suivants

1. toutes les classes et toutes les archives dans vrac
/home/student/java/encours/vrac/*
2. les paquetages utils.text et utils.io
/home/student/java/encours/Utils/* /home/student/java/stables/Utils/*
3. toutes les archives dans libs + les paquetages utils.text et utils.io
/home/student/java/stables/libs /home/student/java/encours/Utils/* /home/student/java/stables/Utils/*

Que font les commandes suivantes ?

`java -cp encours/vrac:stables/libs/* Test 3 carottes :`

Exécute le programme Java Test avec les arguments 3 et carottes. Utilise les fichiers de classe dans les répertoires encours/vrac et stables/libs/*.

`java -jar stables/libs/Utils.jar 3 radis :`

Exécute le programme Java contenu dans le fichier JAR Utils.jar. Passe les arguments 3 et radis au programme.

`java -cp stables/libs/* utils.io.TestIOUtils :`

Exécute la classe Java utils.io.TestIOUtils. Utilise tous les fichiers JAR et répertoires du répertoire stables/libs/ comme classpath.

`java -version:1.5 -verbose:Class -jar stables/libs/a.jar :`

Exécute le fichier JAR a.jar. Utilise Java version 1.5. Active la sortie détaillée du chargement des classes (-verbose:Class).

Exercices

1. Créez un jar exécutable pour la classe fr.nantes.iut.lpmiar.Exemple1 :

Après avoir créé un manifeste à la source

`jar cfm Exemple1.jar manifest.txt \fr\nantes\iut\lpmiar\Exemple1.class`

2. Archivage d'une application : Structure :

```
project/
├── src/
│   ├── app/
│   │   └── ... (sources de l'application)
│   └── Utils/
│       └── ... (sources d'utilitaires)
├── build/
│   ├── app/
│   │   └── ... (bytecode de l'application)
│   └── Utils/
│       └── ... (bytecode d'utilitaires)
├── resources/
│   ├── manifest
│   └── ...
```

`jar cfm MonApplication.jar resources/manifest -C build app -C build Utils`

3. Archivage d'une bibliothèque : Structure :

```
mylibrary/
├── src/
```

```
| └─ malib/
| └─ ... (sources du package malib)
└─ build/
  | └─ malib/
  | └─ ... (bytecode du package malib)
  └─ doc/
    | └─ api/
    |   └─ ... (documentation d'API)
    └─ ...

jar cfe MalibLibrary.jar malib -C src malib -C build malib -C doc/api malib
```