

**COMPILATION,
EXÉCUTION,
ARCHIVAGE**

INTRODUCTION

OBJECTIFS

Une introduction aux phases de

- Compilation,
- Exécution,
- Archivage,

en ligne de commande.

APPLICATION

APPLICATION

Une application Java nommée App se déclare

```
public class App {  
    ...  
}
```

dans un fichier App.java.

Elle déclare une fonction main de signature

```
public static void main(String[] args) {  
    ...  
}
```

(Sinon, c'est un module, pas une application.)

- `main` : point d'entrée de l'application,
- `args` : arguments passés à l'application.

App.java

```
import java.lang.System;
public class App {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```


COMPILATION

COMPILATION

- nécessite la commande javac,
- fournie par [Java Development Kit \(JDK\)](#).



```
$ javac App.java
```

ARTEFACT

- le fichier `App.class` généré par la compilation.

EXECUTION

EXECUTION

- nécessite la commande java,
- fournie par [Java Runtime Environment \(JRE\)](#).



Dans le répertoire contenant le fichier `App.class`:

```
$ java App  
Hello, World!
```

APPLICATIONS COMPOSITES

APPLICATION COMPOSITE

- module Printer,
- application App.

Printer.java

```
import java.lang.System;
public class Printer {
    public static void call(String text) {
        System.out.println(text);
    }
}
```

App.java

```
public class App {  
    public static void main(String[] args) {  
        Printer.call("Hello, World!");  
    }  
}
```

COMPILATION

```
$ javac Printer.java App.java
```

EXECUTION

```
$ java App  
Hello world!
```

ARCHIVAGE

OBJECTIF

Distribuer le programme comme un fichier unique.

Manifest.txt

Spécifier le point d'entrée de l'application :

```
Main-Class: App
```


ARCHIVAGE

- nécessite la commande jar (Java Archive Tool),
- fournie par [Java Development Kit \(JDK\)](#).



Compilez vos fichiers .java, puis :

```
$ jar cfm app.jar Manifest.txt *.class
```

Voir aussi : [Packaging Programs in JAR Files](#)

EXECUTION

Dans le répertoire contenant le fichier `app.jar` :

```
$ java -jar app.jar  
Hello, World!
```

BIBLIOTHÈQUES

BIBLIOTHÈQUE

- Ensemble de fonctionnalités,
- Disponibles sous forme de package(s),
- Pas de point d'entrée (pas une application).

printer/Printer.java

```
package printer;
import java.lang.System;
public class Printer {
    public static void call(String text) {
        System.out.println(text);
    }
}
```

➤ _ COMPILATION ET ARCHIVAGE

```
$ javac printer/Printer.java  
$ jar cf printer.jar printer
```

CLASSPATH

Pour utiliser la bibliothèque contenue dans l'archive `printer.par`, il faut indiquer à java de prendre en compte ce fichier au moyen du classpath

- à la compilation de l'application,
- lors de son exécution.

EXEMPLES

Ajoutez au classpath

- "." pour les fichiers .class du répertoire courant,
- "*" pour les fichiers .jar du répertoire courant.

Voir aussi [Setting the class path](#).

➤ SPÉCIFIER LE CLASSPATH

Comme :

- une variable d'environnement

```
$ export CLASSPATH=". : *"
```

- une option en ligne de commande de java/javac :

```
$ javac -cp ". : *" *.java
```

App.java

```
import printer.Printer;  
public class App {  
    public static void main(String[] args) {  
        Printer.call("Hello, World!");  
    }  
}
```



```
$ javac -cp ".:*" App.java
```

```
$ ls
```

```
App.class  App.java  printer.jar
```

EXECUTION

```
$ java -cp ".:*" App  
Hello, World!
```