

JAVA “BAREBONES”

INTRODUCTION

OBJECTIFS

Une introduction aux phases de

- Compilation,
- Packaging,
- Execution,

SANS BlueJ, Visual Studio Code, Eclipse, etc.

1 – APPLICATION MINIMALE

APPLICATION

- Une application Java suppose un programme déclarant une méthode avec le prototype:

```
public static void main(String[] args)
```

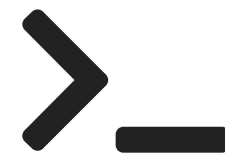
- Le tableau args contient les arguments passés à l'application lors de son invocation.

App.java

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

COMPILATION

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



```
$ javac App.java
```


DISTRIBUTION

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

EXECUTION

- nécessite la commande java.
- elle est fournie par **Java Runtime Environment (JRE)**.



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

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

2 - DISTRIBUER UNE ARCHIVE

App.java

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

Printer.java

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

Manifest.txt

```
Main-Class: App
```

ARCHIVAGE

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



Après compilation, dans le répertoire contenant les fichiers *.class:

```
$ 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!
```

3 - DISTRIBUER UNE BIBLIOTHÈQUE

printer/Printer.java

```
package printer;

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



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

CLASS PATH

- Pour utiliser la bibliothèque contenue dans l'archive `printer.par`, il faudra indiquer à java d'explorer ce fichier au moyen de la variable `CLASSPATH`.
- Par exemple: ajoutez `*` au class path pour que java prenne en compte tous les fichiers jar contenus dans le répertoire courant.
- Voir aussi [Setting the class path](#)

App.java

```
import printer.Printer;

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



```
$ javac App.java
```

```
$ ls
```

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


EXECUTION

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

```
$ java App
```

```
Hello, World!
```