Auto-executable JAR

I. Création une JAR auto-executable par commande

Les fichiers JAR (pour **J**ava **AR**chive) sont des fichiers archives qui permettent d'archiver des fichiers dans un format spécifique à Java, et donc portable. Les fichiers JAR ne sont en fait rien d'autre que des fichiers zip renommés (essayez d'ouvrir un fichier JAR avec Winzip ou Winrar, vous verrez que cela fonctionne). L'intérêt des archives JAR est aussi qu'elles peuvent être **auto-exécutables**, faisant office de *.exe* pour vos programmes Java sous Windows par exemple.
Ceci un repertoire comtenant Paire.java et PairHashMap.java, on veut creer un jar package de ces fichiers java.

1. On utilise javac pour générer les fichiers de class en spécifiant l'adresse: /build
   $javac -sourcepath src -d build/ src/Hashmap/*.java

```
tearsyu@tearsDog:~/UPEC/OOP/EXO/src/Hashmap$ tree
.
├── [tearsyu  4.0K]  build
└── [tearsyu  4.0K]  src
    └── [tearsyu  4.0K]  Hashmap
        ├── [tearsyu  3.3K]  PaireHashMap.java
        └── [tearsyu   584]  Paire.java
```

2. On crée un fichier de manifest en spécifiant l'entrée du programme, ce n'est pas forcément utile dans le cas d'une archive non-exécutable et n'oublie pas l'espace entre ":" et le nom de package:
   $echo "Main-Class: Hashmap.PaireHashMap" > manifest

```
tearsyu@tearsDog:~/UPEC/OOP/EXO/src/Hashmap$ cat manifest
Main-Class: Hashmap.PaireHashMap
```

3. Creer un jar file:
   $jar cfm pair.jar manifest -C build/ .

- 'c' :veut dire que vous voulez créer un fichier JAR.
- 'm' : indique que vous voulez utiliser votre propre fichier MANIFEST.
- 'f' : indique que le résultat(le nom du jar) sera un fichier.
- build/pair.jar: le nom du résultat.
- manifest: le fichier manifest qu'on a créé en indiquand l'entrée du programme.
- -C build/ . : changer le répertoire temporaire pendant l'exécution du fichier jar en ajoutant les fichiers class dans build/, le point '.' veut dire que tous les fichiers dans build/.

4. Pour faire le test du jar:

$ java -jar pair.jar

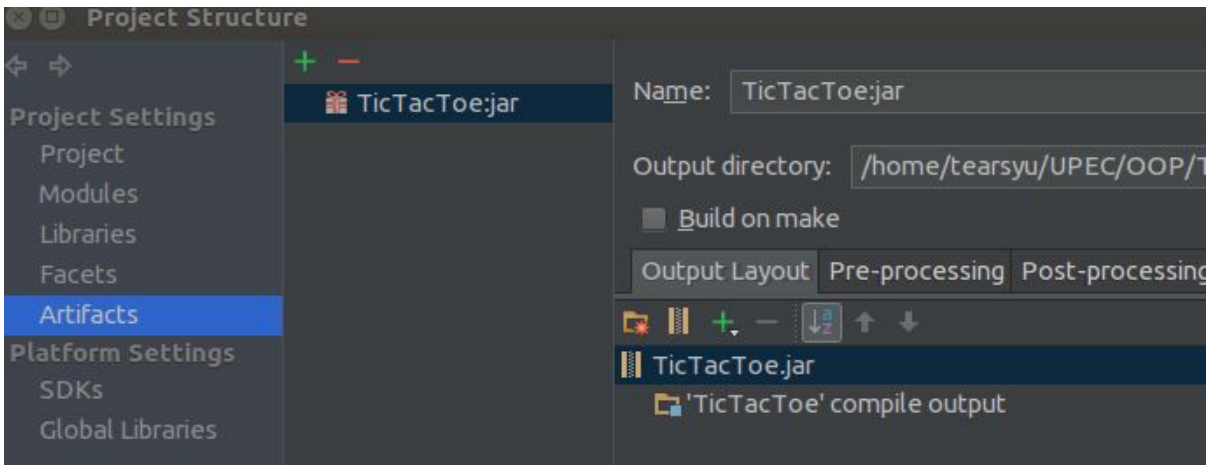
```
heck if contains
Insert] Not null leng but no this word.
Insert] NULL
heck if contains
Insert] Not null leng but no this word.
 | (do : 0.13333)(me : 0.06667)
 | (How : 0.06667)(you : 0.20000)(and : 0.06667)(are : 0.06667)(can : 0.06667)
 | (what : 0.06667)(yuan : 0.06667)
 | (doing : 0.06667)(bring : 0.06667)
 | (thounsand : 0.06667)
SUM] Total vocaburary : 15
```

II. Création une JAR auto-executable par IDE

On peut créer une JAR par IDE qu'on utilise, c'est plutôt simple.

a. Intellij IDEA

File→ Project Structure→ Artifacts et cliquer ajouter "+" → jar->from moduls with dependencies.. → Rebuild project.



b. Eclipse

File → Export → Jar file → Choisir vos packages → Spécifier le contentu du Manifest → Cliquer "OK"

III. Pour aller plus loin

Maven, Ant, Gradle servent à créer les règles en indiquant comment le projet se construit et se compile par le format XML(pas Gradle).