

Comment générer un fichier exécutable ?

a) Situation

On suppose que le code de l'application est réparti dans deux classes Java *Principal.java* et *OutilFichier.java*. Ces deux classes sont situées dans le *package* nommé *application*. De plus l'application utilise un fichier qui a pour nom *donnees.txt*.

L'arborescence est la suivante, à partir d'un dossier nommé *programmation* :

```
→ programmation
    → donnees.txt
    → application
        → OutilFichier.java
        → Principal.java
```

b) Compilation

Comme habituellement, il faut compiler les fichiers *.java* pour obtenir des fichiers *.class*.

```
C:\ ... \programmation>javac application.OutilFichier.java
C:\ ... \programmation>javac application.Principal.java
```

c) Préparation d'un fichier *MANIFEST.MF*

Il faut créer un fichier texte dans le dossier *programmation*, fichier qui aura pour nom *MANIFEST.MF*. Ce fichier doit contenir la ligne de code suivante :

```
Main-Class: application.Principal
```

Notez bien d'une part la présence d'un espace après le caractère ':', et d'autre part la présence d'un retour à la ligne, après la ligne citée ci-dessus (le fichier comporte donc 2 lignes, dont une vide). Vous l'avez compris, on indique dans ce fichier le nom de la classe contenant la fonction *main*.

d) Génération d'une archive contenant l'exécutable (à partir des fichiers *.class*)

Il faut utiliser la commande *jar* de la manière suivante (notez la présence d'un point en fin de commande, voir les explications à la page suivante):

```
C:\ ... \programmation>jar cvfm monapplication.jar MANIFEST.MF .
```

Les options de la commande ont la signification suivante :

- c indique que l'on souhaite créer un fichier *jar*
- f indique que le résultat de la commande doit se placer dans un fichier
- v pour *verbose* : des messages apparaîtront sur la console montrant les étapes de la création de l'archive
- m pour demander à utiliser le fichier *MANIFEST.MF*

Le fichier *monapplication.jar* est le fichier qui sera créé par la commande et qui contiendra l'exécutable. Il sera placé dans le dossier *programmation*.

Le symbole *.* donné en tant que dernier argument de la commande correspond au dossier courant. La commande ira chercher les fichiers *.class* à partir de ce dossier, et dans les sous-dossiers éventuels de celui-ci.

e) Création d'un fichier de commande

A ce stade, si un utilisateur souhaite lancer l'exécutable, il devra taper la commande :

```
C:\ ... \programmation>java -jar monapplication.jar
```

Pour lui faciliter la tâche, on peut placer cette commande dans un fichier de commande ayant l'extension *.bat* ou *.cmd*.

Il reste un dernier détail à gérer : on souhaite empêcher la fenêtre console de se refermer automatiquement lorsque le programme se terminera. Pour ce faire, dans un fichier de commande *lancermonapplication.bat*, il faut écrire la ligne suivante :

```
cmd /k java -jar monapplication.jar
```

Sous Windows, dans un explorateur de fichiers, il suffit de double-cliquer sur le fichier *lancermonapplication.bat* pour lancer l'application.

f) Situation finale

On obtient finalement l'arborescence suivante :

- programmation
 - donnees.txt
 - application
 - OutilFichier.java
 - Principal.java
 - OutilFichier.class
 - Principal.class
 - MANIFEST.MF
 - monapplication.jar
 - lancermonapplication.bat