

# Trabajar con java desde línea de comandos

## Compilar

Vamos al directorio donde está nuestro proyecto (o el directorio donde hemos decidido que van a ir nuestros fuentes java). Ahí creamos y hacemos nuestros fuentes .java. Es importante tener en cuenta lo siguiente:

- Los ficheros .java que dentro no ponen package, deben colocarse en el directorio de nuestro proyecto. El siguiente fuente debería ir en c:\path\_mi\_proyecto\fuentes1.java

```
public class fuentes1 {  
    ...  
}
```

- Si un fichero .java lleva dentro package paquete;, debemos crear en nuestro proyecto un subdirectorio paquete y meter dentro el fichero .java. El siguiente fuente debería ir en c:\path\_mi\_proyecto\paquete\fuentes2.java

```
package paquete;  
public class fuentes2 {  
    ...  
}
```

- Si un fichero .java lleva dentro package paquete.subpaquete;, debemos crear en nuestro proyecto un subdirectorio paquete y dentro de este otro subdirectorio subpaquete y dentro de este nuestro fuente .java. El siguiente fuente debería ir en c:\path\_mi\_proyecto\paquete\subpaquete\fuentes3.java

```
package paquete.subpaquete;  
public class fuentes3 {  
    ...  
}
```

Es decir, los package y los directorios deben coincidir.

Para compilar, situados en el directorio de nuestro proyecto, debemos compilar así

```
javac fuentes1.java paquete\fuentes2.java paquete\subpaquete\fuentes3.java
```

es decir, debemos compilar desde el directorio de nuestro proyecto y debemos poner, si hace falta, los path para llegar desde ahí a los fuentes. Esto generará los ficheros fuentes1.class, paquete\fuentes2.class y paquete\subpaquete\fuentes3.class

## ***Ejecutar sin crear el .jar***

Una vez que tenemos los .class generados, ya podemos ejecutar. Nuevamente debemos estar situados en el directorio de nuestro proyecto y, según cual sea la clase que contiene el main, debemos ejecutar

```
java fuente1
java paquete.fuente2
java paquete.subpaquete.fuente3
```

Hay que fijarse en varias cosas importantes. La primera es que estamos en el directorio de nuestro proyecto, no en los subdirectorios paquete ni subpaquete. También es importante ver que hemos separado paquete, subpaquete y el nombre de las clases con puntos, y no con las barras \. Finalmente, es importante fijarse que no hemos añadido .class detrás.

## ***Creamos un jar***

Podemos crear un fichero .jar con todas nuestras clases. El fichero .jar no es más que un fichero dentro del cual se guardarán juntos todos los ficheros .class ya compilados, similar a un fichero .zip de winzip o .rar de winrar, donde se guardan empaquetados otros ficheros. Para generar el jar, debemos ejecutar el siguiente comando, situados, como siempre, en el directorio de nuestro proyecto

```
jar cf fichero.jar fuente1.class paquete\fuente2.class
paquete\subpaquete\fuente3.class
```

Esto generará un fichero.jar que tendrá dentro los tres ficheros .class. Es importante fijarse que hemos ejecutado el comando desde nuestro directorio de proyecto y que hemos puesto los paths. Se puede hacer más rápido ejecutando este comando

```
jar cf fichero.jar fuente1.class paquete
```

Esto mete dentro de fichero.jar el fuente1.class y el directorio paquete completo, con todo su contenido. Tal cual lo tenemos, los ficheros paquete\fuente2.java y paquete\subpaquete\fuente3.java también se meterán dentro, pero no molestarán.

Un jar generado de esta forma se puede ejecutar, según qué fuente sea el que tenga el main, con

```
java -cp fichero.jar fuente1
java -cp fichero.jar paquete.fuente2
java -cp fichero.jar paquete.subpaquete.fuente3
```

Hay que advertir que estamos en el directorio de nuestro proyecto, que es donde está el fichero.jar. También que hemos separado paquetes, subpaquetes y clases con punto y no con barras \, además que no hemos puesto .class detrás.

## ***Fichero de manifiesto***

Es posible meter un fichero dentro del jar que indique cual es la clase que tiene el main, de forma que al ejecutar no tengamos que decirle cual es. Para ello, en un fichero de texto normal, de nombre por ejemplo manifiesto.txt, y creado en el directorio de nuestro proyecto, ponemos una línea como esta

```
Main-Class: fuente1
```

o bien, si el main está en alguna de las otras clases, el contenido podría ser

```
Main-Class: paquete.fuente2
```

o bien

```
Main-Class: paquete.subpaquete.fuente3
```

Nuevamente, advertir que hemos puesto todo separado con puntos y sin .class detrás.

Ahora, para generar el jar, escribimos

```
jar      cmf      manifiesto.txt      fichero.jar      fuente1.class  
paquete\fuente2.class paquete\subpaquete\fuente3.class
```

Es importante fijarse que si las opciones son cmf, hay que colocar primero el fichero de manifiesto y luego el fichero.jar. Si las opciones fueran al revés, cfm, habría que invertir el nombre del fichero de manifiesto y del jar

```
jar      cfm      fichero.jar      manifiesto.txt      fuente1.class  
paquete\fuente2.class paquete\subpaquete\fuente3.class
```

Una vez generado el jar con su fichero de manifiesto dentro, podemos ejecutarlo así

```
java -jar fichero.jar
```

La opción -jar hará que java mire el contenido del fichero de manifiesto para buscar el nombre de la clase con el main.

**Adaptado de:**

[http://chuwiki.chuidiang.org/index.php?title=Compilar java desde l%C3%ADnea de comandos](http://chuwiki.chuidiang.org/index.php?title=Compilar_java_desde_l%C3%ADnea_de_comandos)