Paquetes

1 Paquetes

Los paquetes en Java nos permiten organizar mejor el código de nuestra aplicación, separando en distintos directorios los diferentes ficheros fuente. Un paquete, asociado a un determinado directorio, agrupará en ese directorio aquellos ficheros que tienen una determinada relación funcional.

Todos los ficheros fuente de un mismo paquete llevarán como primera línea la palabra clave `package`, con la ruta del directorio del paquete, empleando el carácter '.' como separador.

```
package test;
package test.store;
package test.interface;
package test.interface.errors;
```

Por tanto, si un determinado fichero pertenece al paquete `test.store`, deberá tener esa línea al principio del fichero, y debe existir un directorio `test/store`, donde estará ubicado ese fichero.

1.1 Compilar las clases de un paquete

Supongamos que:

- Estamos en el directorio ***src***, y dentro está el directorio ***test***, y dentro de este último el directorio ***store***.
- Dentro de ***store*** tenemos el fichero ***Program.java***, que pertenece al paquete `test.store`, y, por tanto, su primera línea es "`package test.store;`", y posiblemente otras clases de las que depende `Program.java`.

Para compilar una clase que se encuentra en un paquete hay varias opciones:

- Dentro del directorio previo al paquete (`src`):

  ~/src> *javac test/store/*.java (compila todas las clases)*

  o bien

  ~/src> *javac test/store/P2.java (compila P2.java y todas sus dependencias)*

- Dentro del directorio del paquete (`store`):

  ~/src/test/store> *javac *.java (compila todas las clases)*

  Lo que no funcionará será intentar compilar solamente `Program.java` (si depende de otras clases del mismo paquete, ya que no las buscará en el directorio actual, estén ya compiladas o no).

  ~/src/test/store> *javac Program.java (error si depende de otras clases)*

  Para que compile correctamente, habría que indicarle la ruta en el *classpath*:

  ~/src/test/store> *javac -cp .. Program.java (correcto)*

También podemos querer que el destino de los **.class** sea otro distinto. En este caso, emplearemos el conocido parámetro ***-d***:

En este último caso, si no existe, se creará automáticamente el directorio ***classes/test/store*** dentro de ***src***, y dentro de ***store*** se guardará ***Program.class***. Es decir, se cree donde se cree ***Program.class***, siempre estará en la jerarquía de directorios que se define en el paquete.

```
~/src> javac  -d classes  test/store/Program.java
```

1.2 Ejecutar una clase de un paquete

Para ejecutar una clase de un paquete, hay que tener en cuenta que la clase se llama realmente `test.store.Program`., y que allí donde lo ejecutemos, el programa `java` buscará el directorio `test/store`.

Podemos situarnos en el directorio padre del paquete y ejecutar cualquiera de los dos siguientes comandos, con sintaxis alternativa:

```
~/src> java test/store/Program
~/src> java test.store.Program
```

Y, por supuesto, empleando el CLASSPATH o el parámetro ***-cp***, podemos ejecutarlo desde otro punto que no sea el padre del paquete. Por ejemplo,

```
~/src/test/store> java  -cp ../..  test.store.Program
```

Es decir, el CLASSPATH (o lo indicado en ***-cp***) debe apuntar al directorio padre en el cual comience la estructura del paquete donde se encuentren los **.class**. En general, si esa estructura está dentro del directorio */dir*:

```
~/src> java  -cp  /dir  test.store.Program
```

Cosas que no funcionan serían:

```
~/src/test/store> java  Program   (no hay clase Program aquí)
~/src/test/store> java  test.store.Program  (no hay directorio test aquí)
~/src/test/store> java  -cp ../.. Program  (no hay clase Program en src)
```

2 Los ficheros .jar

Los ficheros **.jar** son el resultado de empaquetar en un fichero comprimido uno o varios paquetes, junto con otros ficheros de diverso tipo (datos, declaraciones, configuración…).

Para crearlos se usa el comando ***jar***, por ejemplo, con la opción ***-help*** para conocer sus posibilidades:

```
~/src> jar -help
```

Para crear un fichero **.jar** con el contenido de un paquete nos situaríamos en el directorio padre del paquete y ejecutaríamos el comando indicando el primer directorio del paquete:

```
~/src> jar  -cvf  test.jar test
```

- La opción **–c** indica que se va a **c**rear el fichero.
- La opción **–f** especifica el nombre del fichero JAR (**test.jar**, que debe ir a continuación).
- La opción -**v** (opcional) hace que se vaya imprimiendo en pantalla el nombre de lo que se va añadiendo.

Se puede comprobar el contenido del fichero **.jar** mediante el comando:

```
~/src> jar  -tf  test.jar
```

Es posible ejecutar las clases de un fichero .**jar** sin necesidad de desempaquetarlo, incluyendo la ruta al fichero **.jar** (su ruta completa incluido su nombre) en el CLASSPATH:

```
~/src> java  -cp test.jar  test/store/Program
```

Incluso podemos ejecutar el propio fichero **.jar**, pero para ello hay que indicar cuál es la clase contenida en el **.jar** que hay que ejecutar (en nuestro caso `test/store/Program`). Eso se hace creando un fichero conocido como ***manifiesto***, que se coloca dentro del **.jar**. Por ejemplo, si el contenido de ***manifest.mf*** es:

```
Main-Class: test.store.Program
<línea en blanco>
```

Crearíamos el **.jar** con:

```
~/src> jar  -cvfm  test.jar manifest.fm test
```

Y podríamos ejecutar `Program` con el comando:

```
~/src> java  -jar  test.jar
```