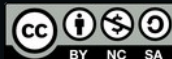


Ficheros en Java



Cristina Moreno Ruiz, basada en material de Lionel Tarazón Alcocer

Entrada y salida de datos

- Teclado y pantalla
- Ficheros
- Bases de datos

package java.io

Librería con las clases necesarias para manejar ficheros.

- Clase File
- Clase FileWriter

Clase File

```
import java.io.File
```

Representa el acceso a una ruta dentro del sistema de archivos, puede ser un archivo o una carpeta.

```
File f = new File (String ruta);
```

Rutas en Windows vs Linux

Ejemplo de ruta Unix: `/usr/bin`

Ejemplo de ruta Windows: `C:\Windows\System32`

Java nos permite utilizar la barra de Unix ("/") para representar rutas en sistemas Windows.

Podremos usar **`File f = new File ("miCarpeta/ej.txt");`** en ambos sistemas operativos.

Rutas absolutas vs relativas

Una ruta **absoluta** es aquella que se refiere a un elemento a partir del raíz del sistema de ficheros. Por ejemplo “C:/Fotos/Foto1.png”

Una ruta **relativa** es aquella que no incluye el raíz y por ello se considera que parte desde el directorio de trabajo de la aplicación.

Por ejemplo “misFotos/Foto2.png”

Rutas absolutas vs relativas

```
File carpetaFotos = new File("C:/Fotos");
```

```
File unaFoto = new File("C:/Fotos/Foto1.png");
```

Los siguientes ejemplos, de rutas relativas, se refieren a una carpeta y un archivo, ambos dentro de la carpeta del proyecto en Netbeans:

```
File otraCarpeta = new File ("Unidad11/apartado1");
```

```
File f = new File ("Actividades.txt");
```

Rutas relativas y ruta real

```
File f = new File ("Unidad11/apartado1/Actividades.txt");
```

Directorio de trabajo	Ruta real
C:/Proyectos/Java	C:/Proyectos/Java/Unidad11/apartado1/Actividades.txt
X:/Unidades	X:/Unidades/Unidad11/apartado1/Actividades.txt
/Programas	/Programas/Unidad11/apartado1/Actividades.txt

Métodos de la clase File

String getParent()

String getName()

String getAbsolutePath()

boolean exists()

boolean isFile()

boolean isDirectory()

long length()

long lastModified()

boolean mkdir()

boolean delete()

boolean renameTo(File destino)

File[] listfiles()

Métodos de la clase File

String getParent()

devuelve la ruta de la carpeta padre, que almacena al elemento referido por esta ruta.

Básicamente la cadena de texto resultante es idéntica a la ruta original, eliminando el último elemento.

Si la ruta tratada se refiere a la carpeta raíz de un sistema de archivos ("C:\\", "/", etc.), este método devuelve null.

En el caso de tratarse de una ruta relativa, este método no incluye la parte de la carpeta de trabajo.

Métodos de la clase File

String getName()

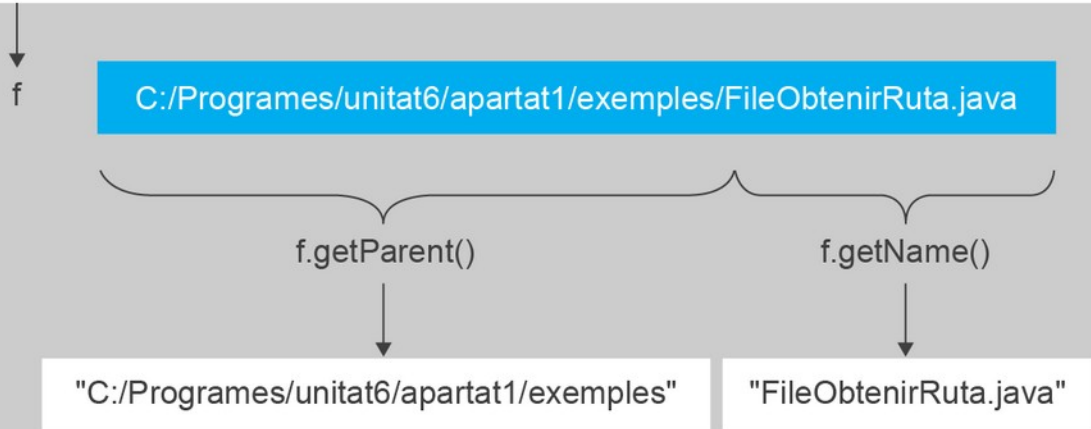
devuelve el nombre del elemento que representa la ruta, ya sea una carpeta o un archivo. Es el caso inverso del método getParent(), ya que el texto resultante es solo el último elemento.

String getAbsolutePath()

devuelve la ruta absoluta. Si el objeto File se inicializó usando una ruta relativa, el resultado incluye también la carpeta de trabajo.

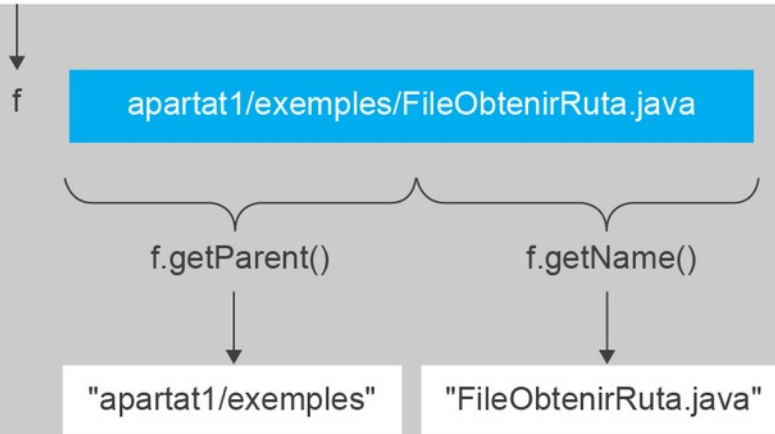
```
File f = new File("C:/Programes/unitat6/apartat1/exemples/FileObtenirRuta.java");
```

Ruta absoluta



```
File f = new File("apartat1/exemples/FileObtenirRuta.java");
```

Ruta relativa



Métodos de la clase File

boolean exists()

comprueba si la ruta existe dentro del sistema de ficheros. Devolverá true si existe y false en caso contrario.

boolean isFile()

comprueba el sistema de ficheros en busca de la ruta y devuelve true si existe y es un fichero. Devolverá false si no existe, o si existe pero no es un fichero.

boolean isDirectory()

funciona como el anterior pero comprueba si es una carpeta.

Métodos de la clase File

long length()

devuelve el tamaño de un archivo en bytes. Este método solo puede ser llamado sobre una ruta que represente un archivo, de lo contrario no se puede garantizar que el resultado sea válido.

long lastModified()

devuelve la última fecha de edición del elemento representado por esta ruta. El resultado se codifica en un único número entero cuyo valor es el número de milisegundos que han pasado desde el 1 de junio de 1970.

Métodos de la clase File

boolean mkdir()

permite crear la carpeta indicada en la ruta. La ruta debe indicar el nombre de una carpeta que no existe en el momento de invocar el método. Por ejemplo, dado un objeto File instanciado con la ruta "C:/Fotos/Albania" que no existe, el método mkdir() creará la carpeta "Albania" dentro de "C:/Fotos". Devuelve true si se ha creado correctamente, en caso contrario devuelve false (por ejemplo si la ruta es incorrecta, la carpeta ya existe o el usuario no tiene permisos de escritura).

boolean delete()

borra el archivo o carpeta indicada en la ruta. La ruta debe indicar el nombre de un archivo o carpeta que sí existe, en el momento de invocar el método. Se podrá borrar una carpeta solo si está vacía (no contiene ni carpetas ni archivos). Devuelve true o false según si la operación se ha podido llevar a cabo.

Métodos de la clase File

boolean renameTo(File destino)

el nombre de este método es algo engañoso ("renombrar", en inglés), ya que su función real no es simplemente cambiar el nombre de un archivo o carpeta, sino cambiar la ubicación completa.

El método se invoca en el objeto File con la ruta origen (donde se encuentra el archivo o carpeta), y se le da como argumento otro objeto File con la ruta destino.

Devuelve true o false según si la operación se ha podido llevar a cabo correctamente o no (la ruta origen y destino son correctos, no existe ya un archivo con este nombre en el destino, etc.).

Nótese que, en el caso de carpetas, es posible moverlas aunque contengan archivos.

Métodos de la clase File

File[] listfiles()

devuelve un vector de tipo File (File[]) con todos los elementos contenidos en la carpeta, es decir subcarpetas y archivos (representados por objetos File, uno por elemento).

Para que se ejecute correctamente la ruta debe indicar una carpeta.

El tamaño del vector será igual al número de elementos que contiene la carpeta. Si el tamaño es 0, el valor devuelto será null y toda operación posterior sobre el vector será errónea.

El orden de los elementos es aleatorio (al contrario que en el explorador de archivos del sistema operativo, no se ordena automáticamente por tipo ni alfabéticamente).

Métodos de la clase File

String getParent()

String getName()

String getAbsolutePath()

boolean exists()

boolean isFile()

boolean isDirectory()

long length()

long lastModified()

boolean mkdir()

boolean delete()

boolean renameTo(File destino)

File[] listfiles()