



La clase File en Java

Características

- ▶ Es una clase asociada a las operaciones de gestión de ficheros.
- ▶ Proporciona información sobre los ficheros a nivel de sistema operativo, como nombres, directorios, distintos tipos de rutas, o atributos.
- ▶ Permite realizar operaciones globales sobre ficheros y directorios, como por ejemplo borrados o cambios de nombre.
- ▶ Se disponen de varios constructores para crear objetos de tipo File. Es necesario pasarles el nombre del fichero y el del directorio donde se va a crear.



Creación de objetos File

- ▶ Constructores:

- ▶ File (String ruta_y_nombrefichero)
- ▶ File (String ruta, String nombrefichero)
- ▶ File (File ruta, String nombrefichero)

- ▶ Ejemplos:

- ▶ `File fichWin = new File("C:\direct1\direct2\fichero.txt");`
- ▶ `File fichLin = new File ("/direct1/direct2/fichero.txt");`

- ▶ `String direct= "C:\direct1\direct2";`
- ▶ `File fichWin2= new File(direct, "fichero2.txt");`

- ▶ `File descrip = new File(direct);`
- ▶ `File fichWin3= new File (descrip, "fichero3.txt");`



Principales métodos

- ▶ **getName()**, devuelve el nombre del fichero o directorio.
 - ▶ **getPath()**, devuelve la ruta relativa.
 - ▶ **getAbsolutePath()**, devuelve la ruta absoluta.
 - ▶ **getParent()**, devuelve el directorio padre o null.
 - ▶ **canRead()**, devuelve cierto si el fichero se puede leer.
 - ▶ **canWrite()**, devuelve cierto si el fichero se puede escribir.
 - ▶ **length()**, devuelve el tamaño del fichero en bytes.
 - ▶ **createNewFile()**, crea un nuevo fichero si no existe.
 - ▶ **mkdir()**, crea un directorio.
 - ▶ **delete()**, borra el fichero o directorio.
 - ▶ **renameTo()**, cambia el nombre con el nuevo argumento.
 - ▶ **exists()**, devuelve cierto si el fichero o directorio existe.
 - ▶ **isDirectory()**, devuelve cierto si el objeto File está asociado a un directorio.
 - ▶ **isFile()**, devuelve cierto si el objeto File está asociado a un fichero.
 - ▶ **list()**, devuelve una lista con los nombres de los ficheros o directorios.
-



Ejemplo

```
Import java.io.File
Public class ListarDirectorio{
public static void main(String[] args){
    System.out.println("Directorio actual: ");
    File f = new File(".");
    String[] lista = f.list();
    for (String nombre : lista) {
        System.out.println(nombre);
    }
}
```



Atributos de la clase File

- ▶ Hay que tener en cuenta los distintos separadores de directorios o carpetas que utilizan los diferentes sistemas operativos, por ejemplo en Linux será “/”, mientras que en Windows es “\”.
- ▶ También los sistemas operativos difieren en el carácter utilizado para separar rutas, por ejemplo en variables del sistema.
- ▶ Para ayudar a las aplicaciones para que puedan intentar ser independientes del sistema operativo, se proporcionan los siguientes atributos de File:
 - ▶ **File.pathSeparator** (devuelve un String). Separador de rutas.
 - ▶ **File.pathSeparatorChar** (devuelve un char). Separador de rutas.
 - ▶ **File.separator** (devuelve String). Separador de carpetas.
 - ▶ **File.separatorChar** (devuelve char). Separador de carpetas.

