**LA CLASE FILE**

**MANEJO DE FICHEROS EN JAVA**

*http://puntocomnoesunlenguaje.blogspot.com.es/2013/05/clase-file-java.html*

* La clase File se usa para obtener información sobre archivos y directorios.
* La clase File permite crear y eliminar archivos y directorios.
* Un objeto de la clase Java File representa un archivo o directorio.

**CONSTRUCTORES**

La clase proporciona los siguientes constructores para crear objetos File:

public File(String nombreFichero|path);

public File(String path, String nombreFichero|path);

public File(File path, String nombreFichero|path);

**Ejemplos utilizando el primer constructor:**

1. Crea un Objeto File asociado al fichero personas.dat que se encuentra en el directorio de trabajo:

File f = new File("personas.dat");

En este caso no se indica path. Se supone que el fichero se encuentra en el directorio actual de trabajo.

2. Crea un Objeto File asociado al fichero personas.dat que se encuentra en el directorio ficheros dentro del directorio actual.

File f = new File("ficheros/personas.dat");

En este caso se indica la ruta relativa tomando como base el directorio actual de trabajo. Se supone que el fichero personas.dat se encuentra en el directorio ficheros. A su vez el directorio ficheros se encuentra dentro del directorio actual de trabajo.

3. Crea un Objeto File asociado al fichero personas.dat dando la ruta absoluta:

File f = new File("c:/ficheros/personas.dat");

El fichero se encuentra en el directorio ficheros. A su vez el directorio ficheros se encuentra en la raíz de la unidad C:

Si se omite la letra de la unidad, por defecto se asume la letra de la unidad en la que se encuentra el proyecto:

File f = new File("/ficheros/personas.dat");

**Ejemplos utilizando el segundo constructor:**

En este caso se crea un objeto File cuya ruta (absoluta o relativa) se indica en el primer String.

1. Crea un Objeto File asociado al fichero personas.dat que se encuentra en el directorio ficheros dentro del directorio actual.

File f = new File("ficheros", "personas.dat" );

En este caso se indica la ruta relativa tomando como base el directorio actual de trabajo.

2. Crea un Objeto File asociado al fichero personas.dat dando la ruta absoluta:

File f = new File("/ficheros", "personas.dat" );

En este caso se indica la ruta absoluta, indicada por la barra del principio.

**Ejemplos utilizando el tercer constructor:**

Este constructor permite crear un objeto File cuya ruta se indica a través de otro objeto File.

1. Crea un Objeto File asociado al fichero personas.dat que se encuentra en el directorio ficheros dentro del directorio actual.

File ruta = new File("ficheros");

File f = new File(ruta, "personas.dat" );

2. Crea un Objeto File asociado al fichero personas.dat dando la ruta absoluta:

File ruta = new File("/ficheros");

File f = new File(ruta, "personas.dat" );

**¡!!** Debemos tener en cuenta que crear un objeto File no significa que deba existir el fichero o el directorio o que el path sea correcto. Si no existen no se lanzará ningún tipo de excepción ni tampoco serán creados.

**MÉTODOS**

Se pueden consultar en la API de Java: *http://docs.oracle.com/javase/7/docs/api/java/io/File.html*

|  |  |
| --- | --- |
| **MÉTODO** | **DESCRIPCIÓN** |
| boolean canRead() | Devuelve true si se puede leer el fichero |
| boolean canWrite() | Devuelve true si se puede escribir en el fichero |
| boolean createNewFile() | Crea el fichero asociado al objeto File. Devuelve true si se ha podido crear. Para poder crearlo el fichero no debe existir. Lanza una excepción del tipo **IOException**. |
| boolean delete() | Elimina el fichero o directorio. Si es un directorio debe estar vacío. Devuelve true si se ha podido eliminar. |
| boolean exists() | Devuelve true si el fichero o directorio existe |
| String getName() | Devuelve el nombre del fichero o directorio |
| String getAbsolutePath() | Devuelve la ruta absoluta asociada al objeto File. |
| String getCanonicalPath() | Devuelve la ruta única absoluta asociada al objeto File. Puede haber varias rutas absolutas asociadas a un File pero solo una única ruta canónica. Lanza una excepción del tipo**IOException**. |
| String getPath() | Devuelve la ruta con la que se creó el objeto File. Puede ser relativa o no. |
| String getParent() | Devuelve un String conteniendo el directorio padre del File. Devuelve null si no tiene directorio padre. |
| File getParentFile() | Devuelve un objeto File conteniendo el directorio padre del File. Devuelve null si no tiene directorio padre. |
| boolean isAbsolute() | Devuelve true si es una ruta absoluta |
| boolean isDirectory() | Devuelve true si es un directorio válido |
| boolean isFile() | Devuelve true si es un fichero válido |
| long lastModified() | Devuelve un valor en milisegundos que representa la última vez que se ha modificado (medido desde las 00:00:00 GMT, del 1 de Enero de 1970). Devuelve 0 si el fichero no existe o ha ocurrido un error. |
| long length() | Devuelve el tamaño en bytes del fichero. Devuelve 0 si no existe. Devuelve un valor indeterminado si es un directorio. |
| String[] list() | Devuelve un array de String con el nombre de los archivos y directorios que contiene el directorio indicado en el objeto File. Si no es un directorio devuelve null. Si el directorio está vacío devuelve un array vacío. |
| String[] list(FilenameFilter filtro) | Similar al anterior. Devuelve un array de String con el nombre de los archivos y directorios que contiene el directorio indicado en el objeto File que cumplen con el filtro indicado. |
| boolean mkdir() | Crea el directorio. Devuelve true si se ha podido crear. |
| boolean mkdirs() | Crea el directorio incluyendo los directorios no existentes especificados en la ruta *padre* del directorio a crear. Devuelve true si se ha creado el directorio y los directorios no existentes de la ruta padre. |
| boolean renameTo(File dest) | Cambia el nombre del fichero por el indicado en el parámetro dest. Devuelve true si se ha realizado el cambio. |

**Ejemplo 1 (Averiguar qué hace el siguiente programa)**

**import** java.io.File;

**import** java.io.IOException;

public class File1 {

public static void main(String[] args) throws IOException {

File ruta = new File("c:/ficheros");

File f = new File(ruta, "datos.txt");

System.out.println(f.getAbsolutePath());

System.out.println(f.getParent());

System.out.println(ruta.getAbsolutePath());

System.out.println(ruta.getParent());

if (!f.exists()) { //se comprueba si el fichero existe o no

System.out.println("Fichero " + f.getName() + " no existe");

if (!ruta.exists()) { //se comprueba si la ruta existe o no

System.out.println("El directorio " + ruta.getName() + " no existe");

if (ruta.mkdir()) { //se crea la ruta. Si se ha creado correctamente

System.out.println("Directorio creado");

if (f.createNewFile()) { //se crea el fichero. Si se ha creado correctamente

System.out.println("Fichero " + f.getName() + " creado");

} else {

System.out.println("No se ha podido crear " + f.getName());

}

} else {

System.out.println("No se ha podido crear " + ruta.getName());

}

} else { //si la ruta existe creamos el fichero

if (f.createNewFile()) {

System.out.println("Fichero " + f.getName() + " creado");

} else {

System.out.println("No se ha podido crear " + f.getName());

}

}

} else { //el fichero existe. Mostramos el tamaño

System.out.println("Fichero " + f.getName() + " existe");

System.out.println("Tamaño " + f.length() + " bytes");

}

}

}

**Ejemplo 2**

public class File2 {

public static void main(String[] args) {

System.out.println("Ficheros en el directorio actual: ");

File f=new File(".");

*//Probad con "..", ¿qué archivos muestra?"*

String []archivos=f.list();

for (int i=0; i< archivos.length; i++)

{

System.out.println(archivos[i]);

}

}

}

**Ejercicio 1**

Pide al usuario un nombre absoluto de directorio y muestra su contenido en la pantalla:

**Ejercicio 2:**

Solicita un nombre de fichero o directorio al usuario y muestra en pantalla:

* Su ruta
* Ruta absoluta
* Tamaño
* Si se puede leer
* Si se puede escribir
* Si es fichero
* Si es directorio

Nota: Prueba la salida tanto para un archivo como para un directorio

**Ejercicio 3:** (Ojo! Si el directorio ya existe no se crea y en el caso de ficheros tb).

1. Crea un directorio nuevo en el directorio actual con nombre solicitado al usuario
2. A continuación crea dos ficheros vacíos dentro de dicho directorio (FICHERO1.TXT y FICHERO2.TXT).
3. Lista el contenido del directorio nuevo
4. Renombra FICHERO2.TXT y llámale FICHERO3.TXT
5. Lista el contenido del directorio

Nota: después de cada operación hay que informar al usuario del éxito o fracaso en la operación.