

LA CLASE FILE

- ✚ No sirve para leer ni para escribir en un archivo sino que permite crear o borrar archivos o directorios, cambiar el nombre de un archivo, leer el nombre del directorio, consultar si un nombre representa un fichero o directorio, listar el contenido de un directorio ...
- ✚ Independiente de la plataforma (constante File.separator):

```
import java.io.File;

String sFichero = "mifichero.txt";
String sDirectorio = "midirectorio";
String sPath = File.separator + sDirectorio + File.separator +
sFichero;
System.out.println(sPath);
/*
\midirectorio\mifichero.txt
*/
```

```
String sFichero = "mifichero.txt";
String sDirectorio = "midirectorio";
String so = System.getProperty("os.name");
String separador = System.getProperty("file.separator");
String sPath = separador + sDirectorio + separador +
sFichero;
System.out.println(sPath);
System.out.println(so);

/*
\midirectorio\mifichero.txt
Windows 7
*/
```

- ✚ Constructores:

- File (String viaAcceso)
- File (String directorio, String fichero)
- File (File directorio, String fichero)

```
import java.io.*;
import java.util.*;
/*
 * La clase File tiene tres constructores
```

```
    File(String path)
    File(String path, String name)
    File(File dir, String name)
```

El parámetro path indica el camino hacia el directorio donde se encuentra el archivo, y name indica el nombre del archivo.
Los métodos más importantes que describe esta clase son los siguientes:

```
    String getName() --> Nombre
    String getPath() --> Camino Relativo
    String getAbsolutePath() --> Camino absoluto
    boolean exists()
    boolean canWrite() --> Si el archivo se puede escribir
    boolean canRead --> Si el archivo se puede leer
    boolean isFile()
    boolean isDirectory()
    boolean isAbsolute()--> Si un Path es absoluto
    long lastModified()
    long length() --> Longitud en bytes.
    boolean mkdir() --> crea directorio
    boolean mkdirs()-->crea direct. y los superiores si no existen.
    boolean renameTo(File dest);
    boolean delete()
    String[] list()
    String[] list(FilenameFilter filter)
```

```
...
```

```
...
```

```
*/
```

```
public class EjemploClaseFile {
    public static void main (String arg[]){
        String directorio;
        if (arg.length > 0)
            directorio=arg[0];
        else
            directorio=".";
        File actual = new File(directorio);
        System.out.println("El directorio es: ");
        try {
            System.out.println(actual.getCanonicalPath());
        } catch (IOException e) {
        }
        System.out.println("Su contenido es: ");
        File[] archivos = actual.listFiles();
        for (File archivo : archivos) {
            if(archivo.isFile()){
                System.out.println("Nombre "+ archivo.getName());
                System.out.println("Longitud en caracteres "+ archivo.length());
                System.out.println("Modificado "+ new Date(archivo.lastModified()));
                System.out.println("Camino "+archivo.getPath());
                System.out.println("Camino absoluto "+archivo.getAbsolutePath());
                System.out.println("Se puede escribir "+archivo.canRead());
                System.out.println("Se puede leer "+archivo.canWrite());
            }
            System.out.println();
        }
    }
}
```

```
/*
 * El directorio es:
C:\
Su contenido es:

Nombre autoexec.bat
Longitud en caracteres 24
Modificado Wed Jun 10 23:42:20 CEST 2009
Camino c:\autoexec.bat
Camino absoluto c:\autoexec.bat
Se puede escribir true
Se puede leer true

Nombre config.sys
Longitud en caracteres 10
Modificado Wed Jun 10 23:42:20 CEST 2009
Camino c:\config.sys
Camino absoluto c:\config.sys
Se puede escribir true
Se puede leer true

Nombre hiberfil.sys
Longitud en caracteres 2337484800
Modificado Thu Jan 01 01:00:00 CET 1970
Camino c:\hiberfil.sys
Camino absoluto c:\hiberfil.sys
Se puede escribir false
Se puede leer false

Nombre pagefile.sys
Longitud en caracteres 3116646400
Modificado Thu Jan 01 01:00:00 CET 1970
Camino c:\pagefile.sys
Camino absoluto c:\pagefile.sys
Se puede escribir false
Se puede leer false

 */
}
```

```
import java.io.*;

public class EjemploOperacionesArchivo {
    public static void main (String arg[]){
        // Creación de un fichero
        //
        try {
            File file = new File("D:\\Alberto.txt");
            boolean resultado = file.createNewFile();
            if (resultado)
                System.out.println("Archivo creado");
            else
                System.out.println("No se pudo crear el archivo");
            //Posiblemente exista
        } catch (IOException e) {

            System.out.println("Se produjo el error: " + e.getMessage());
            //Acceso denegado por protección C:
            // bloque catch vacío --> ignora la excepción

        }

        //Borrado de un fichero
        //
        try {
            File file = new File("D:\\Alberto.txt");
            boolean resultado = file.delete();
            if (resultado)
                System.out.println("Archivo borrado");
            else
                System.out.println("No se pudo borrar el archivo");
            //P.ej. por protección C:, porque no exista
        } catch (Exception e) {
            System.out.println("Se produjo el error: " + e.getMessage());
            //El bloque try nunca provoca una IOException
        }

    }
}
```

FILTROS Y ARCHIVOS DE ACCESO DIRECTO → CURSO PRÓXIMO, MÓDULO “ACCESO A DATOS”