**¿Qué clase permite escribir datos elementales (*int*, *float*, *double*, etc.) en un fichero?**



*DataInputStream*



*FileOutputStream*



*DataOutputStream*

La clase DataOutputStream permite escribir datos en un fichero directamente como tipos de datos primitivos (int, float, double, etc.). Actúa como filtro y trabaja en colaboración con otra clase iniciadora como FileOutputStream.

--BIEN—

**Selecciona la afirmación correcta:**



Todas las clases que representan flujos de datos están ubicadas en el paquete *java.io*.



Todas las clases que representan flujos de datos están ubicadas en el paquete*java.util*.



Todas las clases que representan flujos de datos están ubicadas en el paquete *java.lang*.

Las clases que representan flujos de datos están en el paquete java.io.

---BIEN--

**¿Qué retorna el método *list()* de la clase *File*?**



Un array de objetos *File*.



Un array de objetos *String*.



Un array de objetos genéricos *Object*; se aplica polimorfismo.

--MAL--

**La interfaz *Serializable*...**



Aporta a las clases que implementan la capacidad de persistencia.



Hace que los objetos de las clases que la implementan se guarden automáticamente en disco.



*Serializable* no es una interfaz, es una clase abstracta.

Para que un objeto pueda ser guardado en disco debe pertenecer a una clase que implemente la interfaz Serializable, es decir, una clase que tenga capacidad de persistencia.

---BIEN--

**ReadFloat() es un método de la clase...**



*DataInputStream*



*DataOutputStream*



*BufferedReader*

*--MAL---*

**Dado el siguiente código:**

**String texto="Luna nueva;Luna creciente;Luna menguante";**

**Scanner lector = new Scanner(texto);**

**lector.useDelimiter(";");**

**lector.next();**

**System.out.println(lector.next());**

**lector.close();**

**¿Cuál será el resultado de su ejecución?**



Luna



Luna creciente



Luna nueva

El primer next() lee el fragmento “Luna nueva”, pero no se muestra en pantalla; el segundo next() lee “Luna creciente” y está vez si se muestra en pantalla, porque está encerrado en una sentencia System.out.println(…).

--BIEN—

**Para que el usuario pueda introducir una frase por teclado es necesario:**



Un flujo de datos de escritura.



Un flujo de datos de lectura.



Ninguna de las anteriores, los flujos de datos sólo permiten la entrada/salida con ficheros.

CREO QUE ES “Un flujo de datos de lectura.”

--MAL—

**Indica la afirmación correcta sobre el lenguaje Java.**



Dispone de clases que representan flujos de datos que son simultáneamente de lectura/escritura.



Dispone de clases que representan flujos de lectura, pero no de escritura.



Dispone de clases que representan flujos de lectura y de clases que representan flujos de escritura, pero no existen clases que representen flujos de lectura/escritura.

Java no dispone de clases que representen flujos de lectura y escritura. Si necesitamos leer y escribir en un fichero necesitamos dos flujos distintos: un flujo de entrada o lectura, y otro de salida o escritura.

---BIEN—

**¿Cuál de estas clases representa un flujo de datos que permite escribir en un fichero datos de tipo primitivo (*int*, *double*, *float*, etc.)?**



*BufferedReader*



*DataOutputStream*



*FileReader*



*FileInputStream*

DataInputStream permite escribir datos en un fichero directamente como tipos de datos primitivos (int, float, double, etc.). Actúa como filtro y trabaja en colaboración con otra clase iniciadora, como FileOutputStream.

--BIEN--

**La clase abstracta que sirve como base para todos los flujos de salida de datos en formato binario es...**



*OutputStream*



*Writer*



*Reader*



*InputSream*

*--MAL-- Flujos de Byte InputSream y OutputStream.*

**WriteInt(int valor) es un método de la clase...**



*OutputStreamReader*



*DataInputStream*



*DataOutputStream*



*BufferedReader*

*--BIEN—*

**Dada la siguiente sentencia:**

**File fich = new File("coches\\ford\\modelos.txt");**

**Marca qué operaciones podemos realizar con la variable *fich*. Hay varias respuestas correctas.**



Comprobar si existe el fichero modelos.txt.



Añadir nuevos modelos de coche en el fichero modelos.txt.



Obtener el número de bytes que ocupa el fichero dentro del sistema de archivos.



Borrar una línea de texto dentro del contenido del fichero modelos.txt.



Modificar la matrícula de uno de los coches guardados en modelos.txt.



Comprobar si el fichero modelos.txt tiene permisos de escritura.

La clase File no nos permite efectuar operaciones de lectura/escritura de ficheros; para eso están los flujos de datos.  
La clase File no nos permite efectuar operaciones de lectura/escritura de ficheros; para eso están los flujos de datos.  
La clase File no nos permite efectuar operaciones de lectura/escritura de ficheros; para eso están los flujos de datos.

--BIEN—

**Marca las afirmaciones correctas sobre la clase *File*.**



Permite crear un fichero.



Permite crear un fichero y escribir una línea en él.



Permite crear una carpeta.



Permite crear un fichero y lo abre para escritura.



Permite borrar una carpeta.

La clase File nos permite crear un fichero, pero no escribir en él; para eso están los flujos de datos.  
La clase File nos permite crear un fichero, pero no escribir en él; para eso están los flujos de datos.

--MAL—

**¿Cuáles de estas clases representan flujos de datos que actúan como filtros? Hay varias respuestas correctas.**



*BufferedReader*



*DataInputStream*



*FileInputStream*



*FileOutputStream*



*DataOutputStream*



*FileReader*



*BufferedWriter*



*FileWriter*

Los flujos de datos iniciadores se interactúan directamente con el dispositivo para leer o escribir información. Los flujos de datos que representan filtros se sitúan entre el flujo iniciador y el programa.  
Los flujos de datos iniciadores se interactúan directamente con el dispositivo para leer o escribir información. Los flujos de datos que representan filtros se sitúan entre el flujo iniciador y el programa.  
Los flujos de datos iniciadores se interactúan directamente con el dispositivo para leer o escribir información. Los flujos de datos que representan filtros se sitúan entre el flujo iniciador y el programa.  
Los flujos de datos iniciadores se interactúan directamente con el dispositivo para leer o escribir información. Los flujos de datos que representan filtros se sitúan entre el flujo iniciador y el programa.

--BIEN--

**¿Cómo se obtiene el tamaño de un archivo con la clase *File*?**



Con el método *length()*.



Con el método *size()*.



Con el método *getSize()*.

Aplicado sobre un objeto File que representa un fichero, el método length() retorna el tamaño del fichero expresado en bytes.

--BIEN--