

EJERCICIOS ACCESO ENTRADA-SALIDA Y FICHEROS

Los flujos estándar

1.- (clase LeeNombre) Escribir un programa que solicite al usuario su nombre y, utilizando directamente System.in, lo lea de teclado y muestre por pantalla un mensaje del estilo *"Su nombre es Miguel"*. Recuerda que System.in es un objeto de tipo **InputStream**. La clase InputStream permite **leer bytes** utilizando el método **read()**. Será tarea nuestra ir construyendo un String a partir de los bytes leídos. Prueba el programa de manera que el usuario incluya en su nombre algún carácter "extraño", por ejemplo el símbolo "€" ¿Funciona bien el programa? ¿Por qué?

2.- (clase LeeEdad). Escribir un programa que solicite al usuario su edad y, utilizando directamente System.in, la lea de teclado y muestre por pantalla un mensaje del estilo *"Su edad es 32 años"*. En este caso, será tarea nuestra construir un String a partir de los bytes leídos y transformarlo posteriormente en un entero.

3.-(SumarEdades)

Escribir un método void sumaEdades() que lea de teclado las edades de una serie de personas y muestre cuanto suman. El método finalizará cuando el usuario introduzca una edad negativa

Escribe un método main que llame al método anterior para probarlo.

4.- (leerByte) System.in (InputStream) está orientado a lectura de bytes. Escribe un programa que lea un byte de teclado y muestre su valor (int) por pantalla. Pruébalo con un carácter "extraño", por ejemplo '€'.

5.- (leerCaracter) InputStreamReader (StreamReader) está orientado a caracteres. Escribe un programa que lea un carácter de teclado usando un InputStreamReader y muestre su valor (int) por pantalla. Pruébalo con un carácter "extraño", por ejemplo '€'. ¿Se obtiene el mismo resultado que en el ejercicio anterior?.

Entrada “orientada a líneas”.

En los ejercicios anteriores, las limitaciones de la clase utilizada (InputStream), nos obliga a incluir en el programa instrucciones que detecten que el usuario ha terminado su entrada (ha pulsado *INTRO*). La clase *BufferedReader* dispone del método *readLine()*, capaz de leer una línea completa (la propia instrucción detecta el final de la línea) y devolver un *String*.

7.- Repite el ejercicio 1 utilizando un *BufferedReader* asociado a la entrada estándar. La clase *BufferedReader*, está orientada a leer caracteres en lugar de bytes. ¿Qué ocurre ahora si el usuario introduce un carácter “extraño” en su nombre?

Lectura /escritura en ficheros

8.- Crea un fichero de texto y escribe una frase en él. Utiliza las clase *FileWriter* y *FileReader*

9.- (EscribirFichero1) Escribe un programa que, usando las clases *FileOutputStream* y *FileInputStream*.

- escriba los caracteres de tu nombre en un fichero (nombre.txt).
- lea el fichero creado y lo muestre por pantalla.
- Si abrimos el fichero creado con un editor de textos, ¿su contenido es legible?

10.- (EscribirFichero2) Repetir el ejercicio anterior utilizando las clases *FileReader* y *FileWriter*.

Streams para información binaria

11.- (Personas) Escribe un programa que, utilizando entre otras la clase *DataOutputStream*, almacene en un fichero llamado *personas.dat* la información relativa a una serie de personas que va introduciendo el usuario desde teclado:

- Nombre (*String*)
- Edad (entero)
- Peso (*double*)
- Estatura (*double*)

La entrada del usuario terminará cuando se introduzca un nombre vacío.

Nota: Utiliza la clase *Scanner* para leer desde teclado y los métodos *writeDouble*, *writeInt* y *writeUTF* de la clase *DataInputStream* para escribir en el fichero)

Al finalizar el programa, abre el fichero resultante con un editor de texto (notepad o wordpad) ¿La información que contiene es legible?.

12.- (AñadirPersonas) Modifica el programa anterior para que el usuario, al comienzo del programa, pueda elegir si quiere añadir datos al fichero o sobre escribir la información que contiene.

13.- (MostrarPersonas) Realizar un programa que lea la información del fichero *personas.dat* y la muestre por pantalla. Para determinar que no quedan más datos en el fichero podemos capturar la excepción *EOFException*

14.- (CalculosPersonas) Realizar un programa, similar al anterior, que lea la información del fichero personas.dat y muestre por pantalla a estatura que tienen de media las personas cuya edad está entre 20 y 30 años.

Manejo de ficheros y carpetas

1. Mostrar ruta absoluta de la ruta actual
2. Crear un archivo temporal y mostrar por pantalla la ruta en que se encuentra y el nombre del fichero (en dos líneas distintas).
3. Mostrar el nº de archivos y directorios que contiene una carpeta.
4. Calcular y mostrar el tamaño de los archivos contenidos en una carpeta (sin incluir las subcarpetas)
5. ¿Se puede utilizar `createNewFile` para crear una carpeta? ¿Si creamos un directorio, crea toda la ruta? Pruébalo.
6. ¿Hay algún método para determinar qué tamaño tiene una carpeta?
7. Dada una carpeta, renombrar todos sus archivos. El nombre será el original al que añadiremos el sufijo "(copia)". Si no fuera posible renombrar el fichero porque ya existe uno con ese nombre, iremos añadiendo números (copia1) o (copia2)
8. Dada una carpeta mostrar su contenido: tamaño de los ficheros, o indicación de que es una carpeta.
9. Dada una carpeta, mostrar la fecha de última modificación de cada uno de los archivos que contiene.
10. Dada una carpeta, calcular su tamaño. El tamaño será la suma del tamaño de los archivos que contiene y carpetas que contiene (recursivo).
11. Dada una carpeta, calcular la fecha de su última modificación. La fecha será la fecha de última modificación del archivo que se ha modificado más recientemente (recursivo).
12. Dada una carpeta cuya ruta indica el usuario, crear un fichero de texto cuyo nombre sea el de la carpeta indicada y extensión txt. El fichero contendrá el nombre de los ficheros que contiene la carpeta (solo el nombre y la extensión, sin la ruta). Cada nombre aparecerá en una línea distinta, en minúsculas y ordenados alfabéticamente de menor a mayor.