UT1.Manejo de ficheros.

Actividad 1. Ficheros.

Es opcional, pero recomendable que algunos de los ejercicios los hagas para Linux ;-)

(A) **Listas de ficheros.**

La propiedad **list() de la clase *File*** nos devuelve un vector que contiene el nombre de todos los ficheros y directorios contenidos en el directorio actual o el que estemos posicionados.

Se pide:

1) Desarrollar el código Java que muestre la lista de ficheros en el directorio actual.

2) Desarrollar el código Java que muestre la lista de ficheros en un directorio facilitado por el usuario.

(B) **Propiedades de ficheros.**

Desarrollar el código Java que solicite el nombre de un directorio del que se mostrarán los ficheros y subdirectorios y todas sus propiedades, tales como:

- *Nombre del fichero/directorio*
- *Ruta relativa*
- *Ruta absoluta*
- *Atributo de lectura*
- *Atributo de escritura*
- *Tamaño*
- *Si es un directorio*
- *Si es un fichero*

(C) Texto.

- Desarrolla un programa en java que realice lo siguiente:

- ❖ Abre un fichero de texto => **File**
- ❖ Genera el flujo de entrada => con **FileReader**
- ❖ Muestra los caracteres recuperados por pantalla uno a uno => **read**
- ❖ Cierra el fichero => **close**

Con el mismo fichero anterior, haz la lectura del fichero ***de 10 caracteres en 10 caracteres***

Desarrolla un programa en java que cree un fichero nuevo y escribe una cadena de caracteres:

cadena="Primera prueba con FileWriter"

- ❖ Crea el fichero
- ❖ Crea el flujo de salida => **FileWriter**
- ❖ Escribe los caracteres de cadena uno a uno => **write**
- ❖ Cierra el fichero

En el mismo fichero anterior y **añadiéndolo al final,** escribe la cadena " y esto es el final." de manera completa, no carácter a carácter.

(D) Con BUFFERED.

El objetivo es hacer una lectura y una escritura desde/hacia un archivo, **línea a línea**. Utiliza un fichero de texto con diferentes líneas (de diferente longitud).

Para la lectura:

- ❖ Identifica un fichero de texto => **BufferedReader** hace uso de FileReader

- Define una cadena String para la lectura
- Haz la lectura línea a línea y muéstrala por pantalla=> readLine
- Cierra el fichero => close

Para la escritura:

- Identifica un fichero de texto => B**ufferedWriter** hace uso de FileWriter
- Define una cadena String para la escritura, de manera que siempre escriba lo mismo seguido de "fila " y el nº de fila, para apreciar que se trata de una nueva línea.

  Ejemplo: Fila 1 Dam2

  Fila 2 Dam2...
- Y recuerda usar newLine
- Cierra el fichero => close

(E) **Ficheros Binarios**.

Desarrolla un programa en java que escriba bytes en un fichero y posteriormente lo recupere del fichero y lo muestra por pantalla (lo lee):

- Crea el flujo de salida al fichero (**FileOutputstream**)
- Crea el flujo de entrada (**FileInputStream**)
- Escribe datos en el fichero (50 iteraciones de contador i, u otro método) => **write**
- Lee el fichero => **read**

Al fichero anterior, añade al final 10 caracteres “+” (de uno en uno)

(F) **Ficheros de Objetos**.

a) **Crea una clase Personas** con varios atributos: nombre, DNI, edad (para forzar un integer) y salario. Crea los métodos get y set y otros que consideres.

Recuerda que la clase debe implementar la interfaz ***Serializable***

```
public class Personas implements Serializable  {
```

b) **Crea un fichero** que contenga datos fuentes de Personas, es decir **escribe objetos Persona**, desde Java, siguiendo los pasos:

- Crea el flujo de salida ObjectOutputStream
- Crea los arrays con los nombres, edades… o léelos de teclado (como mínimo 6 personas)
- Recorre los arrays creando cada OBJ y escribiendo cada "persona" en el fichero

c) **Cierra** el stream de salida.

d) **Lee objetos Persona** del fichero creado anteriormente, para lo que tendrás que crear el flujo de entrada ObjectInputStream y Muestra los datos por pantalla.