

TEMA 2 – MANEJO DE FICHEROS

RELACIÓN 2

1. Realizar un programa que utilice `DataOutputStream` y `DataInputStream` para almacenar y posteriormente leer los datos más importantes de los libros de una biblioteca.
 - Deberá crearse una clase `Libro` que tendrá los siguientes atributos y métodos:
 - Atributos: ISBN de tipo `long`, título de tipo `String` y autor de tipo `String`. Todos ellos privados.
 - Métodos:
 - `Libro(long cod, String s1, String s2)` Constructor de la clase.
 - `Public void grabarLibro(DataOutputStream fich) throws IOException`. Guarda en el fichero que se le pasa como argumento los datos del libro.
 - `Public void leerLibro(DataInputStream fich) throws IOException`. Lee del fichero los datos de un libro y los almacena en los atributos del libro.
 - `Public void mostrarLibro()`. Muestra en pantalla los datos del libro.
 - `Public long consultarISBN()`. Devuelve el valor de ISBN.
 - `Public String consultarTitulo()`. Devuelve el valor de título.
 - `Public String consultarAutor()`. Devuelve el valor del autor.
 - Crear una clase `Biblioteca` que contendrá el método `main` y que realizará las siguientes operaciones:
 - Deberá crear los flujos de entrada y salida (`DataOutputStream` y `DataInputStream`).
 - Solicitar por pantalla los datos de libros (varios) y almacenarlos en el fichero.
 - Leer el contenido del fichero mostrando en pantalla todos los libros insertados anteriormente. Se puede hacer uso del método `available` de `DataInputStream` (`int available()`: devuelve el número de bytes que están disponibles para leerse).
2. Se tiene un archivo con los códigos de los productos y sus precios (`fproducto.dat`), campos:
codproducto | precio

El código del producto de un registro coincide con su posición relativa en el archivo.

Realizar un programa que genere el fichero **fproducto.dat** y posteriormente lea del teclado códigos de productos y les aumente el precio en un 10%. Se utilizará acceso directo al archivo. Se terminará el proceso cuando se introduzca el código 0.

3. Realizar un programa en java con las siguientes especificaciones:

- El primer paso será crear un fichero binario de trabajadores **ftrabajador.dat**. En este fichero se guardarán objetos de la clase **trabajador**. Para cada trabajador los datos a almacenar serán DNI, nombre completo, teléfono, fecha de nacimiento y salario.
- Se creará la clase trabajador con los atributos que se indican en el punto anterior. Crear también los métodos de esta clase, constructor, getters y setters. La clase trabajador deberá implementar la interfaz serializable.
- Los datos de los trabajadores serán solicitados por pantalla. Con los datos introducidos se crearán los objetos y se almacenarán en un fichero binario.
- A continuación, se calculará la media de los salarios de todos los trabajadores que se encuentran en el fichero.
- Por último, se generará un nuevo fichero binario con los trabajadores cuyo salario sea inferior a la media. A estos trabajadores se les incrementará su salario en un 20%. En este fichero también se almacenarán objetos de tipo trabajador.
- Se mostrará en pantalla el contenido de este último fichero generado.