

Práctica 2 Java – Introducción a la POO

El objetivo de esta práctica es consolidar los conocimientos adquiridos relacionados con la POO (Programación orientada a objetos). Para ello, se propone:

1. Crea un proyecto Java llamado Biblioteca.
2. En este caso la Biblioteca del CIFP FBMoll necesita un programa para poder gestionar la biblioteca. Para poder llevarlo a cabo se solicitan realizar:
 - a. Una clase llamada **Libro**. Concretamente debemos almacenar: ISBN, título, autor, editorial, nº de copias y nº copias disponibles. Para esta clase debes crear constructor vacío, constructor con todos los parámetros, constructor copia, toString, getters/setters (DTO) y un contador de libros (que llevará el control de los diferentes libros que hay en la aplicación, por ejemplo, si tenemos el libro Javañol y de este libro tenemos 4 copias, nuestro contador de libro marcará un libro, es decir, es un contador de títulos distintos de la a). Debe haber un control en el setter de manera que el número de copias inicial nunca debe ser menos de 1. Para esta clase además necesitaremos los siguientes métodos:
 - i. Añadir libro. Solicitará los datos para crear un libro y añadirá dicho libro a una lista de libros que recibirá como parámetro el método. **(método static)**
 - ii. Eliminar libro. Solicitará al usuario un ISBN, lo buscará y lo eliminará de la lista que recibirá como parámetro el método. No se puede eliminar un libro que tiene reservas. **(método static)**
 - iii. Buscar libro por ISBN. Pedirá al usuario un ISBN, lo buscará en la lista que recibirá como parámetro el método. En caso de encontrarlo devolverá la posición en la que se encuentra, en caso contrario devolverá -1. **(método static)**
 - iv. Buscar libro por título. Pedirá al usuario un título, lo buscará en la lista que recibirá como parámetro el método. Mostrará cualquier libro que contenga la cadena buscada. **(método static)**
 - b. Una clase llamada **Biblioteca**. Concretamente debemos almacenar: nombre de la biblioteca, lista de libros y lista de personal que gestiona la biblioteca. Para esta clase debes crear constructor vacío, constructor con todos los parámetros, constructor copia, toString, getters/setters (DTO). Debe haber un control en el setter de manera que el nombre de la biblioteca empiece en mayúsculas.
 - i. Mostrar libros. Imprimirá por pantalla toda la lista de libros. **(método instancia)**
 - ii. Mostrar libros disponibles. Solo imprime la lista de los libros disponibles. **(método instancia)**
 - c. Una clase llamada **Persona**. Concretamente debemos almacenar: nombre, apellidos, NIF, contraseña. Para esta clase debes crear constructor con todos los parámetros, constructor copia, toString, getters/setters (DTO) y un control de la contraseña que debe tener al menos 8 caracteres. Crea los métodos

que consideres oportunos para poder añadir/eliminar personas que gestionarán la biblioteca (bibliotecarios).

3. Crea un menú (con switch), que permita gestionar la biblioteca dando soporte a todos los métodos que se han solicitado anteriormente. Puede ser interesante que haya varios menús, uno para gestionar el personal de la biblioteca y otro para gestionar las reservas.

4. OPCIONAL:

Además de lo realizado anteriormente, incluiremos:

- a. Una clase llamada **Usuario**. Concretamente debemos almacenar: nombre, apellidos, NIF, contraseña. Para esta clase debes crear constructor con todos los parámetros, constructor copia, toString, getters/setters (DTO) y un control de la contraseña que debe tener al menos 8 caracteres. Crea los métodos que consideres oportunos para poder añadir/eliminar usuarios a la biblioteca.
- b. Al incluir los usuarios serán estos los que reserven los libros. Piensa, que actualmente un usuario no puede reservar un libro si el usuario tiene más de 5 libros reservados, previamente tiene que devolverlos, por tanto, tu programa debe permitir la reserva y devolución de libros, así como listado de reservas realizadas por un usuario de la biblioteca.
- c. Analiza como le darías solución para tener un control de las reservas de libros realizadas por cada usuario. Teniendo en cuenta tú análisis aplica los cambios que requiera tu solución en el resto de clases y programa principal para dar soporte a lo solicitado.