

EJERCICIO 1

Crea en el proyecto un paquete “modelo” que cubra la funcionalidad para trabajar con una librería. Los datos que tenemos que gestionar son:

- Libro: De cada libro registraremos título, autor, ISBN (String) y precio (BigDecimal). También tenemos libros de segunda mano, cuyo precio lleva un 50% de descuento y tienen un atributo adicional que es la antigüedad (Integer)
- Librería: Tendrá un nombre y un mapa de libros. Este mapa de libros estará compuesto por una clave que será el ISBN (String) de cada libro y un valor que será el objeto Libro en sí.

Crea las clases necesarias. Incluye métodos get() y set() para todo. Además, incluye método equals teniendo en cuenta que dos libros son iguales si tienen el mismo ISBN y dos librerías son iguales si tienen el mismo nombre.

Añade los siguientes métodos adicionales en la clase Librería

- getTasacionCompleta() que no recibe nada y devuelve un BigDecimal con la suma del precio de todos los libros.
- getPrecioMedio() que devuelva el precio medio del libro
- getLibrosBaratos() que devuelva una lista (List) con todos los libros (sean del tipo que sean) que cuesten menos de 10 euros.

EJERCICIO 2

Crea un paquete “servicios” donde tengas una clase “LibreriaService” con lo siguiente:

- Un método llamado insertarLibroNuevo() que recibe un objeto Libro y lo guarda en la siguiente tabla. No devuelve nada. Si hay algún error en la BBDD, lanzará LibroException. Los errores al cerrar conexión o statement se pueden ignorar.
NOTA: servirá tanto para insertar libros nuevos como de segunda mano, pero en este último caso, el atributo antigüedad no se guarda en BBDD.

#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir NULL
1	ISBN	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>
2	TITULO	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>
3	AUTOR	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>
4	PRECIO	DECIMAL	20,6	<input checked="" type="checkbox"/>	<input type="checkbox"/>

MYSQL MARIADB	ORACLE
<pre>CREATE TABLE IF NOT EXISTS libros (ISBN varchar(50) NOT NULL, TITULO varchar(50) NOT NULL, AUTOR varchar(50) NOT NULL, PRECIO decimal(20,6) NOT NULL, PRIMARY KEY (ISBN));</pre>	<pre>CREATE TABLE libros (ISBN varchar2(50) NOT NULL, TITULO varchar2(50) NOT NULL, AUTOR varchar2(50) NOT NULL, PRECIO NUMERIC(20,6) NOT NULL, PRIMARY KEY (ISBN));</pre>

- Un método llamado actualizarPrecioLibroNuevo() que recibe un objeto Libro. Tiene que actualizar su precio en la tabla anterior. (Sólo su precio). Devolverá un booleano indicando si se ha actualizado correctamente o si no existe dicho libro. Si hay algún error en BBDD, lanzará LibroException. Los errores al cerrar conexión o statement se pueden ignorar

EJERCICIO 3

Crea un paquete “app” con una clase “App”. En esta clase, crea los siguientes métodos:

- solicitarLibreria() que tiene que devolver un objeto Libreria. El nombre de la librería será “Páginas desiertas” y el mapa de libros tendrá que ser rellenado solicitando libros al usuario. Se solicitarán libros indefinidamente hasta que el usuario escriba la palabra “FIN”. Todos los libros indicados por el usuario serán nuevos, no serán de segunda mano. Cada libro será indicado por el usuario en una única línea en este formato: ISBN; TITULO; AUTOR; PRECIO

Por ejemplo:

987654651;Crimen y Castigo;Dostoievski;12.34

NOTA: No es necesario validar los datos de entrada. Asumimos que el usuario escribirá todo correctamente.

- main() que llamará al método anterior. Luego recorrerá el mapa de la librería e insertará en BBDD todos los libros que tengan un precio inferior a 10 euros. Para insertar utilizamos el método de la clase LibroService. Si hay algún error al insertar un libro, se mostrará un mensaje indicándolo y se continuará con el resto de libros. No es necesario controlar la transacción para que se inserten todos o ninguno.
 - Al terminar de insertar lo solicitado, mostrará un mensaje indicando que todo OK. Mostrará también el precio medio del libro y la tasación completa, formateando dichos importes correctamente.