

Ejercicio: Bibliotecas

Autor: Toñi Reina. **Revisores:** Fermín Cruz, José Riquelme, Mariano González. **Última modificación:** 23/3/2020

En este ejercicio vamos a implementar varios tipos para la gestión de una Biblioteca. Los tipos que forman el modelo de datos son los siguientes:

- **TipoPrestamo:** tipo enumerado con los valores posibles de un tipo de préstamo.
- **FormatoLibroElectrónico:** tipo enumerado con los valores posibles de un formato de libro electrónico.
- **FormatoAudio:** tipo enumerado con los valores posibles de un formato de audiolibro.
- **Libro:** tipo que representa un libro.
- **LibroElectrónico:** tipo que representa un libro electrónico. Es un subtipo de Libro.
- **Audiolibro:** tipo que representa un audiolibro. Es un subtipo de Libro.
- **Biblioteca:** tipo contenedor que representa una colección de libros.

Tipo Libro

Propiedades:

- **ISBN,** de tipo String. Consultable. El código ISBN, formado por una serie de dígitos y otros caracteres (por ejemplo, "9780385536516"), identifica de manera inequívoca a un libro.
- **título,** de tipo String. Consultable. Título del libro.
- **autor,** de tipo String. Consultable. Nombre del autor del libro.
- **número de páginas,** de tipo Integer. Consultable. Número de páginas del libro.
- **fecha de adquisición,** de tipo LocalDate. Consultable. Fecha de adquisición del libro.
- **precio,** de tipo Double. Consultable. Precio de compra del libro en euros.
- **estimación de ventas,** de tipo Integer. Consultable y modificable. Estimación del número de unidades vendidas del libro.
- **bestseller,** de tipo Boolean. Consultable. Un libro se considera un *bestseller* si el número de unidades vendidas es superior a 500.000.
- **tipo de préstamo,** del tipo enumerado TipoPrestamo, que puede tomar los valores DIARIO, SEMANAL y MENSUAL. Consultable y modificable. Tipo de préstamo del libro.

Constructores:

- Un constructor que recibe un parámetro por cada propiedad básica del tipo, en el mismo orden en el que están definidas.

Criterio de igualdad: dos libros son iguales si sus ISBN son iguales.

Criterio de ordenación: por ISBN.

Representación como cadena: el título seguido de un guion y el nombre del autor. Ejemplo: "Dexter: El oscuro pasajero - Jeff Lindsay".

Restricciones:

- El precio, la estimación de ventas y el número de páginas deben ser positivas.
- El ISBN debe tener 10 o 13 caracteres.
- La fecha de adquisición debe ser anterior al día de hoy.

Otras operaciones:

- *Integer getDiasPrestamo()*: calcula el número de días que puede ser prestado el libro, en función del tipo de préstamo (diario: 1 día; semanal: 7 días; mensual: 30 días).

Tipo LibroElectrónico

Propiedades: (se añaden o modifican las de Libro)

- **ISBN**, de tipo String. Consultable. Versión electrónica del ISBN. Es el ISBN del libro, al cual se le añade el sufijo '-e'.
- **Formato**, del tipo enumerado FormatoLibroElectrónico, que puede tomar los valores PDF, EPUB, AZW, DOC, RTF, DJVU y MOBI. Consultable. Formato de archivo del libro electrónico.

Constructores:

- Un constructor que recibe un parámetro por cada propiedad básica del tipo, en el mismo orden en el que están definidas.

Representación como cadena: igual que la de un libro, pero añadiendo al final "- ISBN: ", seguido del ISBN del libro electrónico.

Tipo Audiolibro

Propiedades: (se añaden a las de Libro)

- **Formato**, del tipo enumerado FormatoAudio, que puede tomar los valores MP3, WMA y M4B. Consultable. Formato de archivo del audiolibro.
- **urlDescarga**, de tipo String. Consultable. URL para la descarga del audiolibro.
- **Tamaño**, de tipo Integer. Consultable. Tamaño del archivo del audiolibro.

Constructores:

- Un constructor que recibe un parámetro por cada propiedad básica del tipo, en el mismo orden en el que están definidas.

Representación como cadena: igual que la de un libro, pero añadiendo al final "Formato: ", seguido del formato del audiolibro.

Restricciones:

- La url debe comenzar por "http:".
- El tamaño debe ser positivo.

Tipo Biblioteca

Propiedades:

- **Nombre**, de tipo String. Consultable. Nombre de la biblioteca.
- **Código Postal**, de tipo String. Consultable y modificable. Código postal de la biblioteca.
- **Email**, de tipo String. Consultable y modificable. Dirección de correo electrónico de la biblioteca.
- **Libros**, de tipo List<Libro>. Consultable. Lista de libros de la biblioteca.

Constructores:

- Un constructor con parámetros para el nombre, el código postal y el email. La propiedad libros se inicializa con una lista vacía.

Representación como cadena: el nombre de la biblioteca seguido del código postal entre paréntesis. Ejemplo: "Infanta Elena (41013)"

Restricciones:

- El nombre no puede ser nulo.
- El código postal debe estar formado por 5 dígitos.
- El email debe contener un solo carácter '@' y al menos un carácter '.'

Otras operaciones:

- *void añadeLibro(Libro l):* añade el libro que se pasa como parámetro a la lista de libros. Se admiten libros duplicados. Si el libro tiene el valor *null*, la operación no tiene efecto.
- *void eliminaLibro(Libro l):* elimina el libro que se pasa como parámetro de la lista de libros. Si el libro no existe, se lanza la excepción *IllegalArgumentException*.

Implemente los tipos descritos en el paquete `fp.bibliotecas`.

Implemente en el paquete `fp.bibliotecas.test` una clase `TestBiblioteca` con un método `main` en el que:

1. Se cree un objeto de tipo *Biblioteca*.
2. Se creen tres objetos, uno de tipo *Libro*, otro de tipo *LibroElectrónico* y otro de tipo *Audiolibro*.
3. Se añadan los tres libros a la biblioteca.
4. Se muestren por consola los libros añadidos a la biblioteca.