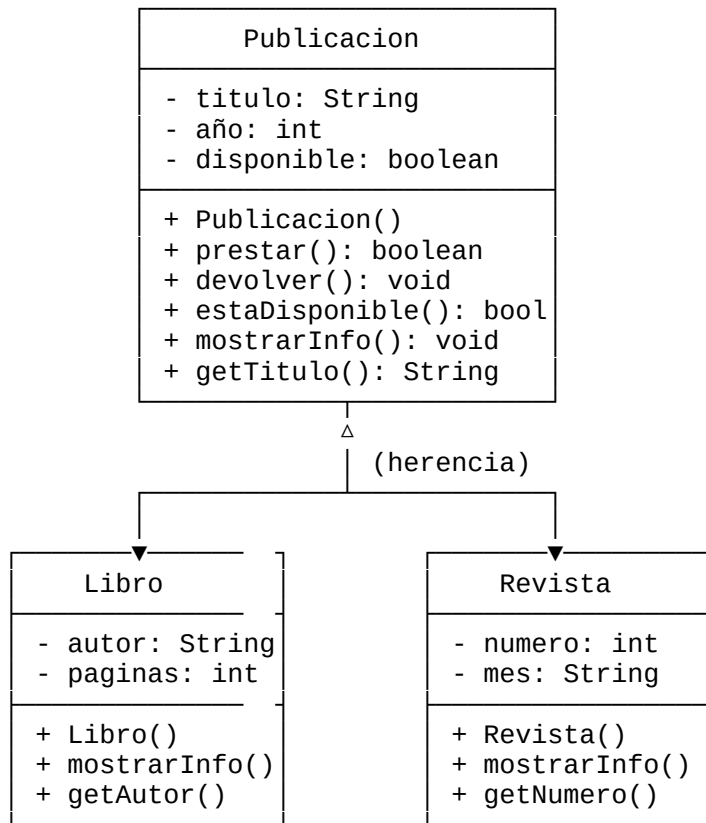


1) A partir del siguiente diagrama UML, debes crear las clases Java correspondientes. Lee atentamente el diagrama y implementa todas las clases con sus atributos, métodos y relaciones.



Hay que implementar:

A. Clase **Publicacion** (clase padre):

Atributos: `titulo`, `año`, `disponible` (TODOS `private`)

Constructor: que reciba `titulo` y `año`, `disponible` inicia en `true`

Métodos:

`prestar()`: si está disponible, la marca como no disponible y retorna `true`, sino `false`

`devolver()`: marca como disponible

`estaDisponible()`: retorna el estado de disponible

`mostrarInfo()`: imprime `titulo`, `año` y estado

`getTitulo()`: getter del `titulo`

B. Clase **Libro** (hereda de **Publicacion**):

Atributos adicionales: `autor`, `paginas` (`private`)

Constructor: que reciba todos los datos (`titulo`, `año`, `autor`, `paginas`)

Métodos:

`mostrarInfo()`: override - debe mostrar toda la info incluyendo `autor` y `páginas`

`getAutor()`: getter del `autor`

C. Clase Revista (hereda de Publicacion):

Atributos adicionales: numero, mes (private)

Constructor: que reciba todos los datos (titulo, año, numero, mes)

Métodos:

mostrarInfo(): override - debe mostrar toda la info incluyendo número y mes

getNumero(): getter del número

D. EN EL MAIN:

- Crear 2 libros y una revista

- Mostrar informacion de los 3

- Prestar uno

- Intentar volver a prestarlo (debe imprimir un error)

- Volver a mostrar info del libro prestado

- Devolver dicho libro

2) Instrumentos musicales. Dibujar las clases en UML con draw.io o a mano y luego escribir el código.

CLASES;

A) Clase padre: Instrumento

Atributos:

nombre (String, private)

marca (String, private)

precio (double, private)

Métodos:

Constructor que reciba todos los atributos

tocar(): método que imprima "Tocando [nombre] de marca [marca]"

afinar(): método que imprima "Afinando [nombre]..."

mostrarInfo(): muestra todos los datos del instrumento

Getters para todos los atributos

B) Clase hija: InstrumentoCuerda

Hereda de: Instrumento (EXTENDS)

Atributo adicional:

numeroCuerdas (int, private)

Métodos:

Constructor que reciba todos los datos

tocar(): Override - "Tocando las [numeroCuerdas] cuerdas de [nombre]"

mostrarInfo(): Override - debe mostrar toda la información incluyendo número de cuerdas

getNumeroCuerdas(): getter

C) Clase hija: InstrumentoViento

Hereda de: Instrumento

Atributo adicional:

material (String, private)

Métodos:

Constructor que reciba todos los datos

tocar(): Override - "Soplando [nombre] de [material]"

mostrarInfo(): Override - debe mostrar toda la información incluyendo material

getMaterial(): getter

-Mostrar info una última vez

D) EN EL MAIN:

-Crear dos instrumentos de cuerda y dos de viento.

-Ponerlos en un array: Instrumento[] orquesta = {guitarra, violin, flauta, saxo};

-Usando la sintaxis de for each recorrer ese array y llamar a los metodos de mostrar info, tocar y afinar (en bucles separados)

POR EJEMPLO:

```
for (Instrumento inst : orquesta) {  
    inst.mostrarInfo();  
    System.out.println();  
}
```