

Ejercicio: Diseño y Desarrollo de un Sistema de Gestión de Biblioteca

Objetivo: Desarrollar un sistema de gestión para una biblioteca utilizando los conceptos de **programación orientada a objetos**. Este ejercicio incluye la creación de **diagramas UML** y la implementación de código en **Java**. Este proyecto les permitirá practicar el uso de **clases abstractas**, **interfaces**, **clases concretas**, y **clases genéricas** en el contexto de gestión de materiales en una biblioteca.

Problema

La biblioteca desea implementar un sistema de gestión de materiales que permita almacenar y gestionar diferentes tipos de materiales, como **libros** y **revistas**. Algunos materiales pueden prestarse a los usuarios, y otros no. El sistema también debe tener una colección centralizada donde puedan agruparse los materiales para fácil acceso y búsqueda.

Requisitos

1. **Materiales de la biblioteca:**
 - Todos los materiales deben compartir ciertos atributos, como el título y el autor.
 - Los materiales deben tener una forma de mostrar sus detalles específicos.
2. **Libros:**
 - Los libros son prestables, por lo que deben poder marcarse como **prestados o devueltos**.
 - Los libros también deben permitir consultar si están actualmente prestados.
3. **Revistas:**
 - Las revistas no se prestan, por lo que no requieren métodos de préstamo.
 - Cada revista debe tener un número de edición para identificarla.
4. **Colección de Materiales:**
 - La biblioteca necesita una colección donde almacenar todos los materiales. Esta colección debe poder contener cualquier tipo de **Material**.
 - La colección debe permitir mostrar los detalles de todos los materiales almacenados.

Instrucciones

1. Fase 1: Diseño UML

- Antes de escribir código, desarrolle los **diagramas UML** que representen la estructura del sistema.
- El diagrama debe incluir:
 - Una **clase abstracta Material** que defina los atributos y métodos comunes para todos los materiales de la biblioteca.
 - Una **interfaz Prestamo** que defina los métodos necesarios para los materiales que pueden ser prestados.
 - Clases **concretas Libro y Revista** que hereden de **Material**. La clase **Libro** debe implementar la interfaz **Prestamo**.
 - Una **clase genérica ColeccionMateriales<T extends Material>** que administre los materiales de la biblioteca y permita almacenar cualquier tipo de **Material**.
- Indiquen todas las relaciones entre las clases (herencia, implementación) y las firmas de los métodos principales.

2. Fase 2: Implementación del Código en Java

- Usando los diagramas UML como referencia, implementen el código en Java.
- Sigán las especificaciones detalladas a continuación para cada clase:
 - **Clase abstracta Material:**
 - Atributos: **titulo** y **autor**.
 - Métodos: Definir métodos **getTitulo()**, **getAutor()**, y un método abstracto **mostrarDetalles()**.
 - **Interfaz Prestamo:**
 - Métodos: **prestar()**, **devolver()**, **estaPrestado()**. Cualquier material que implemente **Prestamo** debe definir estos métodos para indicar su estado de préstamo.
 - **Clase concreta Libro:**
 - Atributos adicionales: un indicador de si el libro está prestado o no.
 - Implementación: Hereda de **Material** e implementa **Prestamo**. Debe implementar el método **mostrarDetalles()** para mostrar el título, autor y estado de préstamo del libro.
 - **Clase concreta Revista:**
 - Atributos adicionales: **numeroEdicion**.

- Implementación: Hereda de **Material**. Debe implementar el método **mostrarDetalles()** para mostrar el título, autor y número de edición de la revista.
- **Clase genérica **ColeccionMateriales<T extends Material>**:**
 - Atributos: una lista genérica que almacene objetos de tipo **Material**.
 - Métodos: **agregarMaterial()** para añadir materiales a la colección, y **mostrarColeccion()** para mostrar los detalles de todos los materiales en la colección.

3. Fase 3: Pruebas en la Clase Principal

- Crear una clase **Principal** para probar el sistema.
- Crear al menos:
 - **Cuatro libros.**
 - **Tres revistas.**
- Agregar estos materiales a la colección y demostrar el uso de los métodos de la interfaz **Prestamo** para prestar y devolver libros.
- Mostrar los detalles de todos los materiales en la colección antes y después de realizar préstamos y devoluciones.

Nota: Este ejercicio les permitirá integrar y aplicar conocimientos clave en POO, diseño de sistemas y estructura de código Java. Les recomiendo analizar y planificar bien el diseño antes de iniciar la codificación.