Modelo Segundo Parcial

Programación II

Contexto

Se te solicita desarrollar una aplicación en Java que gestione una biblioteca. Los libros se deben poder ordenar y filtrar utilizando diferentes criterios. Además se tienen que serializar para poder persistir en archivos.
La aplicación debe permitir iterar sobre la colección de libros para mostrar información específica y permitir generar un CRUD sobre la biblioteca.

Requisitos

1. **Interfaz Genérica**:
   - Crear una interfaz genérica ***Repository<T>***, la cual define métodos para agregar (***add(T)***), eliminar (***remove(T)***) y obtener todos los elementos(***getAll()***).

2. **Interfaz Comparable**:

   - Implementar la interfaz *Comparable* en la clase **Book** para que los libros se puedan comparar por su **título** de forma natural.

4. **Comparator**:

   - Crear varias clases que implementen la interfaz *Comparator* para ordenar los libros por **autor** (**BookAuthorComparator**) y por **año** de publicación (**BookYearComparator**).

5. **Iterator**:

   - Implementar la interfaz *Iterable* en una clase (**Library**) que contenga la colección de libros para poder iterar sobre ella.

6. **Ordenamiento**:

   - Agregar métodos (en Library) para ordenar los libros por diferentes criterios (por título, autor y año).

7. **Serialización**:

   - Crear la interfaz genérica ***Serializer<T>***, con métodos para serializar y deserializar elementos, tanto en formato ***binario*** como en formato ***JSON***.
   - Adecuar los métodos para que permitan guardar / leer desde archivos binarios y archivos JSON. (**writeBinary**, **readBinary**, **writeJSON**, **readJSON**).

Detalles de Implementación

1. Clase **Book**:
   - ❖ Atributos: String title, String author, int year.

2. Clase **Library**:
   - ❖ Atributos: ArrayList<Book> books.

Instrucciones para la resolución del modelo:

1. Implementar las clases y métodos descritos en los requisitos.

2. Escribir un programa principal que demuestre el uso de las diferentes funcionalidades: agregar libros, eliminarlos, modificarlos, ordenarlos por diferentes criterios e iterar sobre la colección.

3. Serializar a binario en ***libros.dat*** (comprobar la correcta deserialización).

4. Serializar a JSON en ***libros.json*** (comprobar la correcta deserialización).

5. Asegurarse de manejar correctamente las excepciones y de seguir las buenas prácticas de programación.