Práctica SOLID

Dado el siguiente código considere el caso en el que se desea agregar otro método de pago como PayPal. Identifique los **principios SOLID que se está violando**, explique la razón y corrija el código de tal forma que ya no se viole ningún principio. Usted puede crear las interfaces y clases que considere o cambiar nombres de métodos si necesario.Incluso puede utilizar **diagramas de clase. Identificar:** 2 pt, **Explicar:** 3 pt y **Solución:** 5 pt**.**

**import** java.util.ArrayList;

**public** **class** Book {

String title;

String author;

String category;

**float** price;

**boolean** state;

**public** Book(String title, String author, String category,**float** price) {

**this**.title = title;

**this**.author = author;

**this**.category = category;

**this**.price = price;

state = **false**;

}

String getTitle() {

**return** title;

}

**void** setTitle(String title) {

**this**.title = title;

}

String getAuthor() {

**return** author;

}

**void** setAuthor(String author) {

**this**.author = author;

}

**void** searchBook() {

/\*

\* Search a book for author or title or category

\*/

}

}

**class** StoreBook {

**private** ArrayList<Book> books;

**private** Book book;

**public** StoreBook() {

/\*\*

\* Initialize array with books

\*/

}

**public** **void** pay(String method, Book book) {

**if**(method == "card") {

pay\_card("numberCard", book);

}**else** **if** ( method == "cash") {

pay\_cash( "money", book);

}

}

**public** **void** pay\_card(String card, Book book) {

/\*

\* change state to true

\*/

}

**public** **void** pay\_cash(String money, Book book) {

/\*

\* change state to true

\*/

}

**public** **void** removeBooksPurchased(ArrayList<Book> books) {

/\*

\* store books with state true in new array and remove the original array

\*

\*/

}

}

|  |  |  |
| --- | --- | --- |
| Principio | Problema | Solución |
| SRP | La clase Book tiene más de una responsabilidad, en este caso se le ha dado la responsabilidad de modelar un Libro y aparte la responsabilidad de Buscar | Realizar una Interfaz Search  searchByAuthor()  searchByTitle()  searchByCategory() |
| SRP | La clase StoreBook tiene la responsabilidad de almacenar libros y aparte la responsabilidad de realizar pagos | Realizar una Interfaz Payment que tenga los metodos:  payByCash()  payByCard() |
| OCP | Las clases existentes no cumplen con este principio debido a que no están abiertas ala extensión, es decir que si quiero agregar una nueva funcionalidad, tengo que modificar mi clase, en este caso la clase Store Book | Seria mejor implementar una nueva clase que extienda de Pago, para que de esta manera se pueda darla la funcionalidad de pago al programa. |

Se creó una clase para SearchBy y PaymentBy que implementan interfaces de Search y Payment debido a que estas interfaces no tienen relación directa con book y StoreBook, por lo que estaríamos violando el principio de inversión de dependencias.