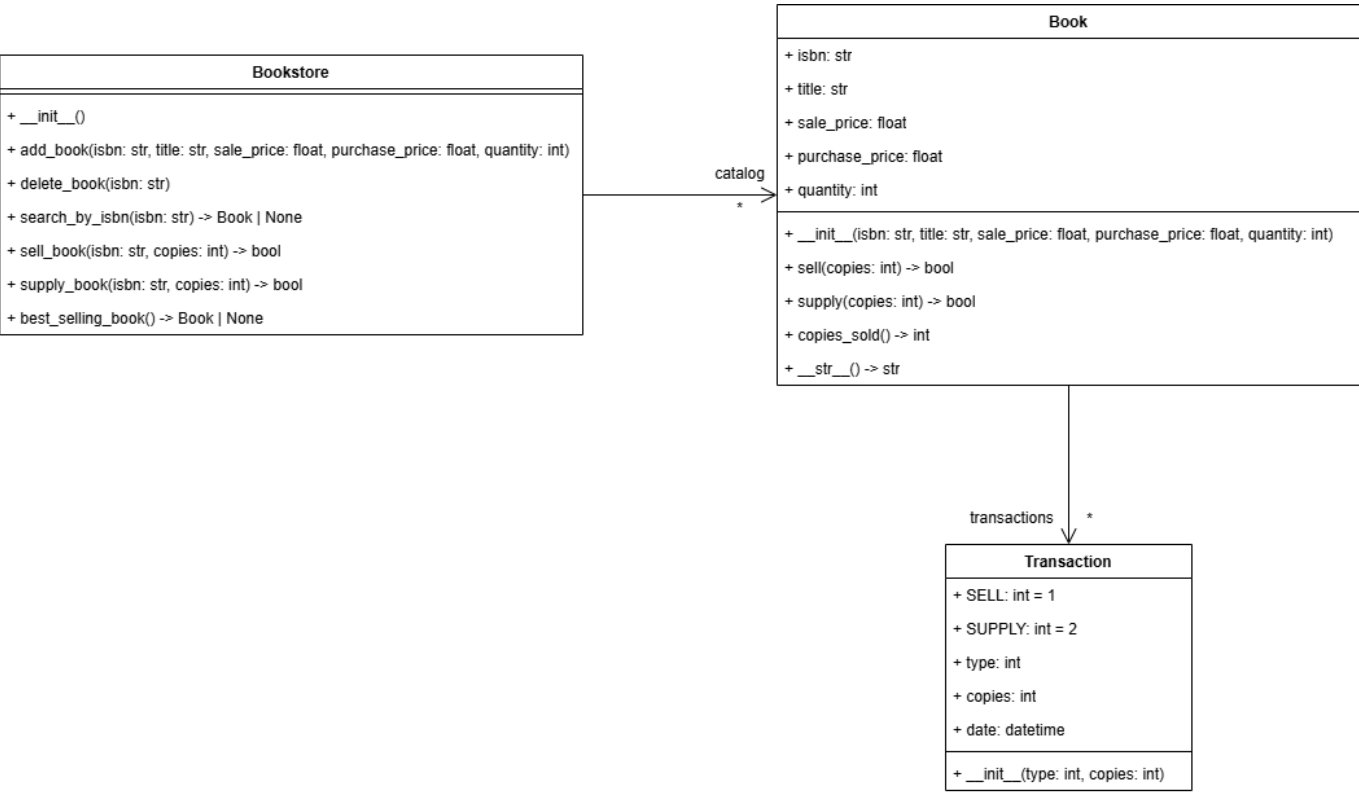


Tienda de libros

La tienda de libros es una aplicación utilizada para evaluar el conocimiento de los conceptos de POO en Python. La aplicación es una simple tienda de libros que permite a los usuarios agregar, eliminar, listar y buscar libros. La aplicación está implementada utilizando clases y objetos en Python.

El modelo de la aplicación es el siguiente:



El código de la aplicación está incompleto, la idea es completarlo teniendo en cuenta los siguientes pasos.

1. Completa la clase **Transaction** teniendo en cuenta los siguientes requisitos:
 - La clase debe tener una constante **SELL** de tipo **int** con valor **1**.
 - La clase debe tener una constante **SUPPLY** de tipo **int** con valor **2**.
 - La clase debe tener un método **__init__** que reciba los siguientes parámetros:
 - **type** de tipo **int**.
 - **copies** de tipo **int**.

En el método **__init__**, la clase debe inicializar los atributos **type** y **copies** con los valores recibidos como parámetros.

- La clase debe tener un atributo **date** de tipo **datetime** que debe inicializarse con la fecha y hora actual

Sugerencia: puedes usar la función `datetime.now()` para obtener la fecha y hora actual.

2. Completa la clase **Book** teniendo en cuenta los siguientes requisitos:

- La clase debe tener un método `__init__` que reciba los siguientes parámetros:

- `isbn` de tipo `str`.
- `title` de tipo `str`.
- `sale_price` de tipo `float`.
- `purchase_price` de tipo `float`.
- `quantity` de tipo `int`.

En el método `__init__`, la clase debe inicializar los atributos `isbn`, `title`, `sale_price`, `purchase_price` y `quantity` con los valores recibidos como parámetros.

- La clase debe tener un atributo `transactions` de tipo `list[Transaction]` que debe inicializarse como una lista vacía.
- La clase debe tener un método de instancia `sell` que reciba un parámetro `copies` de tipo `int` y haga lo siguiente:
 - Si el parámetro `copies` es mayor que el atributo `quantity` del libro, el método debe devolver `False`.
 - De lo contrario, el método disminuye el atributo `quantity` del libro por el valor del parámetro `copies` y agrega un nuevo objeto `Transaction` a la lista `transactions` con el tipo `Transaction.SELL` y el número de `copies` vendidas.
 - El método debe devolver `True`.
- La clase debe tener un método de instancia `supply` que reciba un parámetro `copies` de tipo `int` y haga lo siguiente:
 - Aumenta el atributo `quantity` del libro por el valor del parámetro `copies`.
 - Agrega un nuevo objeto `Transaction` a la lista `transactions` con el tipo `Transaction.SUPPLY` y el número de `copies` suministradas.
- La clase debe tener un método de instancia `copies_sold` que devuelva un `int` con el número total de copias vendidas.

Sugerencia: puedes sumar el número de copias de cada transacción de tipo `Transaction.SELL`.

- La clase debe tener un método de instancia `__str__` que devuelva un `str` con el siguiente formato:

```
ISBN: {isbn}
Title: {title}
Sale Price: {sale_price}
Purchase Price: {purchase_price}
Quantity: {quantity}
```

Donde `{isbn}`, `{title}`, `{sale_price}`, `{purchase_price}` y `{quantity}` deben ser reemplazados por los valores de los atributos del libro.

Sugerencia: puedes usar un f-string (`f""`) para formatear la cadena y `\n` dentro de la cadena para una nueva línea.

3. Completa la clase `Bookstore` teniendo en cuenta los siguientes requisitos:

- La clase debe tener un método `__init__` que inicialice el atributo `catalog` de tipo `dict[str, Book]` como un diccionario vacío.
- La clase debe tener un método de instancia `add_book` que reciba los parámetros `isbn` de tipo `str`, `title` de tipo `str`, `sale_price` de tipo `float`, `purchase_price` de tipo `float` y `quantity` de tipo `int` y haga lo siguiente:
 - Verifica si el `isbn` no está en el diccionario `catalog`.
 - Si el `isbn` no está en el diccionario `catalog`, el método crea un nuevo objeto `Book` con los parámetros recibidos y lo agrega al diccionario `catalog` utilizando el `isbn` como clave.
- La clase debe tener un método de instancia `delete_book` que reciba el parámetro `isbn` de tipo `str` y haga lo siguiente:
 - Verifica si el `isbn` está en el diccionario `catalog`.
 - Si el `isbn` está en el diccionario `catalog`, el método elimina el libro del diccionario `catalog`.
- La clase debe tener un método de instancia `search_by_isbn` que reciba el parámetro `isbn` de tipo `str` y devuelva `Book` | `None` con el libro que tiene el `isbn` recibido o `None` si el libro no está en el diccionario `catalog`.
- La clase debe tener los métodos de instancia `sell_book` y `supply_book` que reciban los parámetros `isbn` de tipo `str` y `copies` de tipo `int`. Copia el siguiente código en la clase `Bookstore` para completar los métodos:

```
def sell_book(self, isbn: str, copies: int) -> bool:
    book = self.search_by_isbn(isbn)
    if book is None:
        return False
    return book.sell(copies)

def supply_book(self, isbn: str, copies: int) -> bool:
    book = self.search_by_isbn(isbn)
    if book is None:
        return False
    book.supply(copies)
    return True
```

- La clase debe tener un método de instancia `best_selling_book` que devuelva `Book` | `None` con el libro que ha vendido más copias o `None` si no hay libros vendidos.