Question

You have to implement an automated order system using the Command design pattern, including at least the following three commands. You will then expand this basic system in any case as you choose to demonstrate the use of additional design patterns.

- Display Book List
- Submit Order
- Display Invoice

Let's start with a basic system, where we assume that all orders originate from the same user (i.e. the system serves only one user at a time). As a result, an invoice is generated by simply summing up all books ordered.

The list of books in the basic system will contain information about the books. Each book has an ISBN, a title, an author and a prize,

Approach

You should implement the Command design pattern. This involves the following:

- Text-based user interface
- Invoker class
- Command interface
- Command classes (one for each command)
- Business classes:
    - BookList class
    - Book Class
    - Orders class
    - OrderItem Class
    - Invoice class

The user interface should simply be a text-based, numbered list of options implemented in the main method.

The BookList and Orders classes keep a collection of Book and OrderItem items, respectively. A Book object will store the isbn#, the title, the author and its price. An OrderItem object will store a bookusing its isbn only (not its title) and quantity.

The Invoker class contains equivalent methods for each item in the application's main menu. Each method creates a Command object of the appropriate Command class (constructed with a reference to the business object, and any other parameter providing the necessary information for the Command object), calls its execute method and returns a single object's result to the application user interface. (Execute methods do not have any parameter values - any needed values are either passed to the constructor or are inputted by the user inside the methods.)

An invoice will be constructed from the Invoice class containing all of the ordered items, returned as an array of strings for the user interface to display. Note that an invoice needs information from both the BookList and the

Orders objects. (The Orders object specifies which books were ordered, and the BookList class has the title of each book to include in the invoice.)