Java Programming Exercise: Library Management System

Background

You are tasked with developing a Library Management System using Java. This system will manage books and borrowers using core principles of Object-Oriented Programming (OOP) such as inheritance, polymorphism, abstraction, and encapsulation. The system will use `HashMaps` to efficiently handle relationships between books and borrowers.

Detailed Task Instructions

1. **Base Class Creation**
    - Create a class `Person` with the following private properties:
        - `name` (String)
        - `age` (int)
    - Implement a constructor to initialize these properties.
    - Provide getters and setters for each property.

2. **Derived Class for Borrowers**
    - Create a derived class `Borrower` from `Person`.
    - Add a property `borrowerId` (String) unique to each borrower.
    - Implement methods to manage the borrower's list of borrowed books.
3. **Book Class**
    - Create a class `Book` with properties:
        - `title` (String)
        - `author` (String)
        - `isbn` (String)
    - Include a constructor and appropriate getters and setters.

4. **Library Management System**
    - Use `HashMaps` for managing books and borrowers:
        - `HashMap<String, Book>` to store books, where keys are `isbn` values.
        - `HashMap<String, Borrower>` to store borrowers, where keys are `borrowerId` values.
    - Implement the following methods:
        - `addBook(Book book)`: Adds a new book to the system.
        - `removeBook(String isbn)`: Removes a book from the system.
        - `addBorrower(Borrower borrower)`: Adds a new borrower.
        - `removeBorrower(String borrowerId)`: Removes a borrower.
        - `checkoutBook(String isbn, String borrowerId)`: Associates a book with a borrower, marking it as checked out if not already loaned.
        - `returnBook(String isbn)`: Marks a book as returned and available for borrowing again.
5. **Additional Methods**
    - `listAllAvailableBooks()`: Lists all books that are not currently checked out.
    - `findBooksByAuthor(String author)`: Returns a list of all books by a specific author.
    - `listBooksBorrowedBy(String borrowerId)`: Lists all books currently borrowed by a specific borrower.

Implementation Notes

- Use proper encapsulation to ensure that the fields of the classes are not directly accessible from outside the classes.
- Consider how to manage the state of a book (e.g., whether it is checked out or available).
- Ensure that the `checkoutBook` method checks if a book is already loaned out before associating it with a borrower.