Class Structure

The Hogwarts Archive system is composed of five main classes:

- The **Archive** class act as an entry point and takes all user input and delegates processing to the CommandHandler class's handleCommand method.
- **CommandHandler** then takes care of data processing, through the handleCommand method calling methods for each type of user command and taking care of print logic such as unique success and error messages.
- **CommonErrorChecker** however is also called by CommandHandler for handling common cases of repetitive checking for and printing of errors such as no students in the system.
- **Student** includes main student data such as creating a unique student ID for each student, and storing name, currently rented spellbooks, and rental history. It also includes methods for clearing all spellbooks, removing a specific spellbook, as well as adding to history and current spellbooks.
- **SpellBook** contains all properties related to the book, including serial number, title, inventor, type, rental status, and rental history. It also contains methods for getting renting status as well as title, inventor and type in nice strings.

Single Responsibility Principle

Each class in this system has only one reason to change, meaning it has only one clear responsibility, hence following the Single Responsibility Principle. This promotes maintainability with reduced risks of bugs if one class is modified. Testability has also been simplified due to easier debugging when locating any errors in the system as each class has a separate responsibility. For example, the main handling of a command and calling relevant functions is managed by the CommandHandler class.

Encapsulation

The SpellBook and Student classes are the main data models in this system, and all properties of both classes are set to private to protect their internal data fields. They contain public getter and setter methods to ensure controlled access of all data. For instance, spellbooks can be removed from and added to a student's currently rented spellbooks list through specific methods as oppose to allowing direct list modification which encapsulates list handling logic.

Abstraction

Abstraction is used extensively in the Hogwarts Archive system to hide internal complexities of handling commands and exposes intuitive behaviour. The CommandHandler extensively utilises this principle, and only handleCommand is called by the Archive class after taking input. This method then splits the string into an array and figures out which specific method to call and passes in the array. The methods called by handleCommand are all private and have abstracted the internal logic of data processing. The Archive class has also been intentionally designed to only call the handleCommand method to abstract the processing of each command away from the class.

The use of functional interfaces in the methods printSpellbookUniqueAttributes and searchSpellbooksByAttribute abstracts the process of how the attribute is extracted. This allows for these methods to not depend on the specific attribute, but only on extracting a string from a spellbook.

Polymorphism

An instance of overloading is present in the Spellbook class for the getPrintableForm which can optionally have a Boolean passed for ease of implementation.