Library System

Design Pattern:

Factory Design Pattern

The factory design pattern is used when we have a superclass with multiple sub-classes and based on input, we need to return one of the sub-class. This pattern takes out the responsibility of the instantiation of a class from the client program to the factory class.

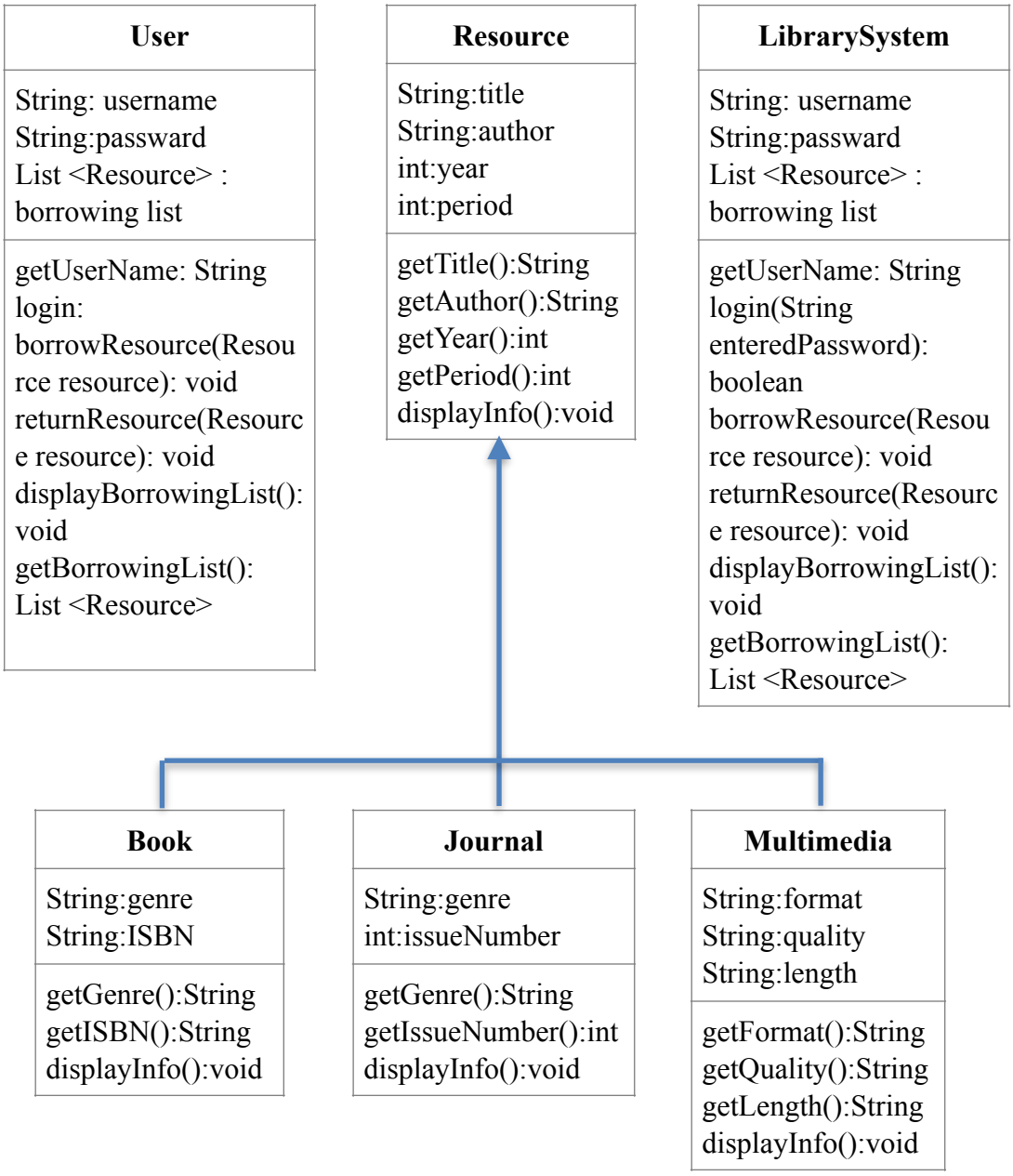
But sample factory pattern still has some cons. If the products keep increasing, the logic focus on factory will be more complicated and hard to maintain efficiency.

In our case, the factory is the Library. The product is resource and concrete products are book, journal, and multimedia. The resource is the super class and concrete products inherits its variable like years, title, and author. In our library systems, user can register their account and then log in to borrow books or return their borrowed ones.

In this situation, concrete products can be more efficiently used when the resource class exists. Unique details for each concrete products are described in their class use override method. Such as the genre in book and quality in multimedia.

Situation:

In today's modern era of technology, libraries have adapted and changed to meet the needs of our society. And now, we come out an idea of creating a code for library systems. We try to create a code that will assist library users in planning their borrowing and making good use of their time and resources. This code not only allows users to choose books from a list, but it also offers a range of functions such as login and registration, borrowing and returning resources, displaying borrowed items, exploring the library's offerings, and logging out when finished. With the help of this application, users can effortlessly and smoothly pick out books from the library and customize their plan for visiting based on their individual preferences. Users have the option to log in and enjoy a personalized service, or they can create their own user accounts through registration. The application also facilitates borrowing and returning resources. When needed, users can easily manage their borrowing list and return resources.


User
String: username
String:passward
List <Resource> :
borrowing list
getUserName: String
login:
borrowResource(Resou
rce resource): void
returnResource(Resourc
e resource): void
displayBorrowingList():
void
getBorrowingList():
List <Resource>
Resource
String:title
String:author
int:year
int:period
getTitle():String
getAuthor():String
getYear():int
getPeriod():int
displayInfo():void
LibrarySystem
String: username
String:passward
List <Resource> :
borrowing list
getUserName: String
login(String
enteredPassword):
boolean
borrowResource(Resou
rce resource): void
returnResource(Resourc
e resource): void
displayBorrowingList():
void
getBorrowingList():
List <Resource>
Book
String:genre
String:ISBN
getGenre():String
getISBN():String
displayInfo():void
Journal
String:genre
int:issueNumber
getGenre():String
getIssueNumber():int
displayInfo():void
Multimedia
String:format
String:quality
String:length
getFormat():String
getQuality():String
getLength():String
displayInfo():void

1. Create **Resource** class

Resource	
Modifier and type	**Method(or Variable) and description**
Instance variable	
String	title
String	author
int	year
Int	period
Constructor	
Resource(String title, String author, int year, int period) Enable to instantiate a Resource object with given title, author, year and period.	
Instance Methods	
-	4 getter for 4 attributes(getTitle(), getAuthor(), getYear(), getPeriod()).
void	displayInfo() Return the information of the resource.

2. Create **Journal** class

Journal	
Modifier and type	**Method(or Variable) and description**
Instance variable	
Int	issueNumber
String	Genre
Constructor	
Journal(String title, String author, int year, int issueNumber, int period, String genre) Enable to instantiate a Journal object with given genre and issueNumber and use super(...) keyword to instantiate the object of superclass by title, author, year, period.	
Instance Methods	
-	2 getter for 2 attributes(getIssueNumber(), getGenre()).
void	displayInfo() Return the information of the journal.

3. Create **Book** class

Book	
Modifier and type	**Method(or Variable) and description**
Instance Variable	
String	genre
String	ISBN
Constructor	
Book(String title, String author, int year, String genre, int period, String ISBN) Enable to instantiate a Book object with given genre and ISBN and use super(...)	
Instance Methods	
-	2 getter for 2 attributes(getISBN(), getGenre()).
void	displayInfo() Return the information of the book.

4. Create Multimedia class

Multimedia	
Modifier and type	**Method(or Variable) and description**
Instance variable	
String	format
String	quality
String	length
Constructor	
Multimedia(String title, String author, int year, String format, int period, String quality, String length) Enable to instantiate a Multimedia object with given length, format and quality and use super(...) keyword to instantiate the object of superclass by title, author, year, period..	
Instance Methods	
-	3 getter for 3 attributes(getFormat(), getQuality(), getLength()).
void	displayInfo() Return the information of the multimedia.

5. Create User class

User	
Modifier and type	**Method(or Variable) and description**
Instance variable	
String	username
String	passord
List <Resource>	borrowinglist
Constructor	
public User(String username, String password): Enable to instantiate a User object with the given username and password. Initializes the borrowingList as an empty ArrayList**.**	
Instance Methods	

-	2 getter for 2 attributes(getUserName(), getBorrowingList()).
boolean	login(String enteredPassword) Verifies if the entered password matches the user's password. Returns **true** if the passwords match, otherwise **false**.
void	borrowResource() Adds a resource to the user's borrowing list and prints a success message
void	displayBorrowingList() Displays the borrowing list for the user, including information about each borrowed resource.

6. Create LibrarySystem class

LibrarySystem	
Modifier and type	**Method(or Variable) and description**
Instance variable	
List<Resource>	resource
List<User>	users
User	currentUser
Constructor	
public LibrarySystem(): Enable to initialize the lists of resources and users and populate the resources with some initial items (books, journals, multimedia).	
Instance methods	
-	1 getter for 1 attribute(getCurrentUser()).

void	login(String username, String password) Allows a user to log in by providing a username and password. Checks the provided credentials against the registered users and sets the **currentUser** if the login is successful.
void	registerUser(String username, String password) Registers a new user with the given username and password, provided the username is at least 4 characters long and the password is at least 8 characters long.
void	borrowResource() Allows the currently logged-in user to borrow a resource from the library by entering the title of the desired resource.
void	returnResource() Allows the currently logged-in user to return a borrowed resource by entering the title of the resource to be returned.
void	displayBorrowingList() Displays the borrowing list of the currently logged-in user.
void	displayLibraryResource() Displays information about all available resources in the library.
void	logout() Logs out the current user by setting **currentUser** to **null**.
main	public static void main(String[] args) The main method that creates an instance of **LibrarySystem**, initializes it with some data, and runs a **LibrarySystemTester** to interact with the library system.
Private Helper Method	
Resource	findResourceByTitle(String title) Searches for a resource in the library by its title and returns the first match.

Resource	findBorrowedResourceByTitle(String title) Searches for a resource in the currently logged-in user's borrowing list by its title and returns the first match.

7. Create **Test** class

   a. The program begins by creating a **Scanner** object to read user input, and the system displays a menu with three options: Login (1), Register (2), and Exit (3), in which Users are prompted to enter a choice (1-3). By doing this, a perpetual loop (**outerLoop**) is established to keep the program running until the user chooses to exit.
   b. User input is obtained and stored in the variable **choice**, and An **InputMismatchException** is caught if the input is not an integer. In such cases, an error message is displayed, the invalid input is consumed, and the loop restarts.
   c. Use a switch statement to process user's input (variable choice)
   d. Use a loop to display the Library User Menu, with options for various actions until the users chooses to logout.
   e. Depending on the user's input, implement different methods(borrow, return, etc.) based on the LibrarySystem class.

Sample Output

```
Library System Menu:
1. Login
2. Register
3. Exit
Enter your choice (1-3): 2
Enter a new username > 4 characters: MDFK
Enter a new password > 8 characters: NCCU0001
User 'MDFK' registered successfully.

Library System Menu:
1. Login
2. Register
3. Exit
Enter your choice (1-3): 1
Enter your username: MDFK
Enter your password: NCCU0001
Login successful. Welcome, MDFK!

Library User Menu:
1. Borrow a resource
2. Return a resource
3. Display borrowing list
4. Display library resources
5. Logout
Enter your choice (1-5): 1
Library Resources:

（此指令將展現圖書館所有的資源，礙於篇幅，在此省略）

Enter the title of the resource you want to borrow: The Art of Painting
Resource 'The Art of Painting' borrowed successfully.

Library User Menu:
1. Borrow a resource
2. Return a resource
3. Display borrowing list
4. Display library resources
5. Logout
Enter your choice (1-5): 3
Borrowing List for MDFK:
--------------
Kind: Multimedia
Title: The Art of Painting
Author: John Smith
Year: 2021
Lending period: 7 days
Format: Video
Quality: 4K
Length or Page:130min
--------------
```