

Group Members:

- 1)Ibrahim Abu Talib (Cs221061)
- 2)Radhika Rajwani (Cs221055)
- 3)Muhammad Daniyal (Cs221087)
- 4)Naheek Rahim (Cs221105)

Project: Library Management System**Aim:**

The aim of this project is to create a Library Management System using Java programming language. The system should allow users to add and remove books from the library and provide a transcript of the borrowing activity.

Code Explanation:

1. The code defines a class hierarchy for different types of library items, such as FictionBooks, NonFictionBooks, ThrillerBooks, and DVD, which inherit from the abstract class LibraryItems.
2. The LibraryItems class contains common properties and methods for library items, including Library_id, Student_name, Department, category, item, option_number, Book_number, add_remove, yes_no, title, titleToRemove, and a LinkedList of books.
3. The addBook() method allows users to add a book to the library by entering its title.
4. The removeBook() method allows users to remove a book from the library by entering its title.
5. The chooseBook() method in each subclass of LibraryItems allows users to choose a specific book from the respective category.
6. The Borrow() method provides a menu for selecting the category and book to borrow.
7. The Transcript() method generates a transcript with the user's name, ID, department, borrowed item name, added item, and removed item.
8. The main() method serves as the entry point of the program, displaying a welcome message and offering options for browsing categories, borrowing, and adding/removing books.

Project Steps:

1. Start by understanding the code provided and the class hierarchy for library items.
2. Identify the required classes and methods to implement the lab.
3. Create a new Java project in your preferred IDE.
4. Create a new class named "LibraryManagementSystem" in the project.
5. Copy and paste the provided code into the "LibraryManagementSystem" class.
6. Implement the missing methods, such as FictionBooks, NonFictionBooks, ThrillerBooks, DVD, and the abstract methods in LibraryItems.
7. Test the program by running the main() method and interacting with the menu options.
8. Verify that you can add and remove books, select categories, and generate a transcript.
9. Make any necessary adjustments or improvements to the code and functionality.
10. Document the steps taken, the code modifications made, and the results obtained during the lab.

Note:

Ensure that you have the necessary input/output handling and error checking in place. The code provided may have some missing or incomplete parts that you need to implement or modify to make the program work correctly.

Project Output:

The expected output of the project is a functional Library Management System that allows users to browse categories, borrow books, add/remove books, and generate a transcript of their borrowing activity.

Report:

The given code represents a library management system in Java. It includes classes and methods to perform operations like adding books, removing books, borrowing books, and generating a transcript of borrowed books. Let's go through the code and understand its functionality and structure.

1. The code starts with importing required classes and defining a package called "filing" (not included in the provided code snippet).

2. The code includes an abstract class called "LibraryItems" which serves as the base class for different types of library items.

3. The class "LibraryItems" contains several static variables like "Library_id," "Student_name," "Department," "category," "item," "option_number," "Book_number," "add_remove," "yes_no," "title," "titleToRemove," and a static LinkedList named "books" to store Book objects. These variables hold information related to library items and user inputs.

4. The "LibraryItems" class also includes methods like "addBook" and "removeBook" to add and remove books from the library. These methods prompt the user to enter the title of the book and perform the corresponding action.

5. The "LibraryItems" class contains abstract methods like "chooseBook" and "getId" and non-abstract methods like "getName" and "getDepartment." These methods are overridden in the subclasses.

6. The code includes subclasses like "FictionBooks," "NonFictionBooks," "ThrillerBooks," and "DVD" that extend the "LibraryItems" class. Each subclass overrides the "chooseBook" method to provide specific book choices based on the category selected.

7. The "Borrow" method allows the user to select a category and then choose a book to borrow. Based on the user's input, an instance of the respective subclass is created, and the "chooseBook" method is called to perform the borrowing operation.

8. The "Transcript" method prompts the user to enter their name, ID, and department. It then generates a transcript of the borrowed book, including the user's information, borrowed item, added item, and removed item.

9. The "main" method serves as the entry point of the program. It displays a welcome message and prompts the user to enter an option number. Based on the user's input, it performs different operations like displaying book categories, borrowing books, adding or removing books, and generating a transcript.

10. The code uses the "Scanner" class to read user inputs from the console.

11. The "Book" class is a simple class that represents a book with a title. It has a constructor to initialize the title and a getter method to retrieve the title.

Overall, the given code provides a basic library management system where users can add, remove, and borrow books. It uses classes and inheritance to organize and perform different operations. However, some improvements can be made, such as separating the functionality into different methods, using instance variables instead of static variables, and handling user inputs more robustly (e.g., handling invalid inputs).