```java
import java.util.*;
/**
 * The Lbrary class takes the values from various classes i.e. ElectronicResource,
 * PhysicalBook, LibraryMember.
 *
 * @author Deepak
 * @version 30.11.2023
 */
public class Library
{
    //instance field
    private ArrayList<Object> catalogue;
    /**
     * Constructor for class Library objects.
     */
    public Library()
    {
        catalogue = new ArrayList<>();
    }

    /**
     * A method to get the Catalogue.
     */
    public ArrayList<Object> getCatalogue()
    {
        return catalogue;
    }

    /**
     * A method to set the Catalogue.
     */
    public void setCatalogue(ArrayList<Object> cat)
    {
        catalogue = cat;
    }

    /**
     * A method to print the details of the Catalogue.
     */
    public void printDetails()
    {
        System.out.println("Catalogue:- ");
        for (Object resource : catalogue)
        {
            if (resource instanceof PhysicalBook)
            {
                ((PhysicalBook) resource).printDetails();
            }
            else if (resource instanceof ElectronicResource)
            {
```

```
                ((ElectronicResource) resource).printDetails();
            }
        }
    }

    /**
     * A method to check if the Catalogue contains a particular resource.
     */
    public boolean containResource(Object resource)
    {
        return catalogue.contains(resource);
    }

    /**
     * A method to modify the author name in the resource of the Catalogue.
     */
    public void modifyAuthorName(PhysicalBook book, String newAuthorFirstName)
    {
        if (catalogue.contains(book))
        {
            if (book.getAuthor() != null)
            {
                book.getAuthor().setAuthorFirstName(newAuthorFirstName);
            }
            else
            {
                System.out.println("Cannot find Author Data");
            }
        }
        else
        {
            System.out.println("Resurce object not found in catalogue");
        }
    }

    /**
     * A method to search a resource by its title in the Catalogue.
     */
    public void searchTitle(String bookTitle)
    {
        int count = 0;
        for (Object resource : catalogue)
        {
            if (resource instanceof PhysicalBook)
            {
                if (((PhysicalBook) resource).getBookTitle().equals(bookTitle))
                {
                    ((PhysicalBook) resource).printDetails();
                    count += 1;
                }
```

```
            }
        }
        System.out.println("\nTotal resources: " + count);
    }

    /**
     * A method to search a resource by the author's surname in the Catalogue.
     */
    public void searchAuthorSurname(String authSurname)
    {
        int count = 0;
        for (Object resource : catalogue)
        {
            if (resource instanceof PhysicalBook)
            {
                if (((PhysicalBook) resource).getAuthor() != null)
                {
                    String authorSurname = ((PhysicalBook)
resource).getAuthor().getAuthorSurname();
                    if (authorSurname.equalsIgnoreCase(authSurname))
                    {
                        ((PhysicalBook) resource).printDetails();
                        count += 1;
                    }
                }
            }
        }
        System.out.println("\nTotal resource found: " + count);
    }

    /**
     * A method to remove a resource by from the Catalogue.
     */
    public void removeResource(Object resource)
    {
        if (catalogue.contains(resource))
        {
            if (resource instanceof PhysicalBook)
            {
                PhysicalBook book = (PhysicalBook) resource;
                if (book.getLibMember() != null)
                {
                    System.out.println("Resource on loan");
                }
                else
                {
                    catalogue.remove(resource);
                }
            } else
            {
```

```
                catalogue.remove(resource);
            }
        }
        else
        {
            System.out.println("Resource object not found");
        }
    }

    /**
     * A method to remove a resource by its position in the Catalogue.
     */
    public void removeResourceFromPosition(int position)
    {
        if (position >= 0 && position < catalogue.size())
        {
            Object resource = catalogue.get(position);
            if (resource instanceof PhysicalBook)
            {
                PhysicalBook book = (PhysicalBook) resource;
                if (book.getLibMember() != null)
                {
                    System.out.println("Resource on loan");
                }
                else
                {
                    catalogue.remove(position);
                }
            }
            else
            {
                catalogue.remove(position);
            }
        }
        else
        {
            System.out.println("Resource not found at this position");
        }
    }

    /**
     * A method to print the details of the books available in the Catalogue.
     */
    public void printBooksAvailable()
    {
        System.out.println("Books Available: \n");
        for (Object resource : catalogue)
        {
            if (resource instanceof PhysicalBook)
            {
```

```
                PhysicalBook book = (PhysicalBook) resource;
                if (book.isAvailable())
                {
                    book.printDetails();
                }
            }
        }
    }
    /**
     * A method to get the number of resources in the Catalogue.
     */
    public int getNumberOfResources()
    {
        return catalogue.size();
    }

    /**
     * A method to add a resource in the Catalogue.
     */
    public void addResource(Object resource)
    {
        if (catalogue.contains(resource))
        {
            System.out.println("Resource object already exists in the catalogue");
        }
        else
        {
            catalogue.add(resource);
        }
    }

    /**
     * A method to lend a book in the Catalogue to a library member.
     */
    public void lendBook(PhysicalBook book, LibraryMember libmember) {
        if (catalogue.contains(book))
        {
            if (book.isAvailable())
            {
                book.setLibMember(libmember);
                libmember.addBooksBorrowed(book);
            }
            else
            {
                System.out.println("Book already borrowed");
            }
        }
        else
        {
            System.out.println("Book not found");
```

```
        }
    }

    /**
     * A method to return a book in the Catalogue lent by a  library member.
     */
    public void returnBook(PhysicalBook book, boolean recordDamage, String
damageDescription)
    {
        if (catalogue.contains(book))
        {
            book.setLibMember(null);
            if (recordDamage)
            {
                book.addDamage(damageDescription);
            }
            System.out.println("Book returned");
        }
        else
        {
            System.out.println("Book not found");
        }
    }

    /**
     * A method to print the details of all physical books in the Catalogue.
     */
    public void allPhysicalBook()
    {
        System.out.println("Physical Books:");
        for (Object resource : catalogue)
        {
            if (resource instanceof PhysicalBook)
            {
                ((PhysicalBook) resource).printDetails();
            }
        }
    }

    /**
     * A method to print the details of all electronic resources in the Catalogue.
     */
    public void allElectronicResources()
    {
        System.out.println("Electronic Resources: ");
        for (Object resource : catalogue)
        {
            if (resource instanceof ElectronicResource)
            {
                ((ElectronicResource) resource).printDetails();
```

```
            }
        }
    }
}
```