```
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import java.util.*;
import java.io.PrintStream;
import java.io.ByteArrayOutputStream;

/**
 * The test class for Library class.
 *
 * @author  Deepak
 * @version 30.11.2023
 */
public class LibraryTest
    {
        /**
         * Testing the method getCatalogue.
         */
        @Test public void tetCatalogue()
        {
            Library library1 = new Library();
            assertNotNull(library1.getCatalogue());
        }

        /**
         * Testing the method setCatalogue.
         */
        @Test public void setCatalogue()
        {
            Library library1 = new Library();
            ArrayList catalogue = new ArrayList<>();
            library1.setCatalogue(catalogue);
            assertEquals(catalogue, library1.getCatalogue());
        }

        /**
         * Testing the method containResource.
         */
        @Test public void containResource()
        {
            Library library1 = new Library();
            Author author1 = new Author();
            PhysicalBook physical1 = new PhysicalBook(author1, "ABC123",
"BookTitle");
            library1.addResource(physical1);
            assertTrue(library1.containResource(physical1));
        }

        /**
```

```
         * Testing the method modifyAuthorName.
         */
        @Test public void modifyAuthorName()
        {
            Library library1 = new Library();
            Author author1 = new Author();
            PhysicalBook physical1 = new PhysicalBook(author1, "ABC123",
"BookTitle");
            author1.setAuthorFirstName("Deepak");
            physical1.setAuthor(author1);
            library1.addResource(physical1);
            library1.modifyAuthorName(physical1, "Deepak");
            assertEquals("Deepak", physical1.getAuthor().getAuthorFirstName());
        }

        /**
         * Testing the method searchTitle.
         */
        @Test public void searchTitle()
        {
            Library library1 = new Library();
            Author author1 = new Author();
            PhysicalBook physical1 = new PhysicalBook(author1, "ABC123",
"BookTitle");
            PhysicalBook physical2 = new PhysicalBook(author1, "java", "java");
            library1.addResource(physical1);
            library1.addResource(physical2);
            library1.searchTitle("CS994");
            assertEquals(2, library1.getNumberOfResources());
        }

        /**
         * Testing the method searchAuthorSurname.
         */
        @Test public void searchAuthorSurname()
        {
            Library library1 = new Library();
            Author author1 = new Author();
            Author author2 = new Author();
            PhysicalBook physical1 = new PhysicalBook(author1, "ABC123",
"BookTitle");
            PhysicalBook physical2 = new PhysicalBook(author1, "java", "java");
            author1.setAuthorSurname("Raw");
            author2.setAuthorSurname("Rawal");
            physical1.setAuthor(author1);
            physical2.setAuthor(author2);
            library1.addResource(physical1);
            library1.addResource(physical2);
            library1.searchAuthorSurname("Raw");
            assertEquals(2, library1.getNumberOfResources());
        }
```

```
         * Testing the method removeResource.
         */
        @Test public void removeResource()
        {
            Library library1 = new Library();
            Author author1 = new Author();
            PhysicalBook physical1 = new PhysicalBook(author1, "ABC123",
"BookTitle");
            library1.addResource(physical1);
            library1.removeResource(physical1);
            assertFalse(library1.containResource(physical1));
        }

        /**
         * Testing the method removeResourceFromPosition.
         */
        @Test public void removeResourceFromPosition()
        {
            Library library1 = new Library();
            Author author1 = new Author();
            PhysicalBook physical1 = new PhysicalBook(author1, "ABC123",
"BookTitle");
            PhysicalBook physical2 = new PhysicalBook(author1, "java", "java");
            library1.addResource(physical1);
            library1.addResource(physical2);
            library1.removeResourceFromPosition(1);
            assertFalse(library1.containResource(physical2));
        }

        /**
         * Testing the method printBooksAvailable.
         */
        @Test public void printBooksAvailable()
        {
            Library library1 = new Library();
            Author author1 = new Author();
            PhysicalBook physical1 = new PhysicalBook(author1, "ABC123",
"BookTitle");
            PhysicalBook physical2 = new PhysicalBook(author1, "java", "java");
            library1.addResource(physical1);
            library1.addResource(physical2);
            library1.printBooksAvailable();
            String expectedOutput = "Books Available: \n" + "Title: BookTitle
\nAuthor: " + "Title: java  \nAuthor:  ";
            assertFalse(library1.toString().contains(expectedOutput));
        }

        /**
         * Testing the method getNumberPfResources.
         */
        @Test public void getNumberOfResources()
```

```
            Author author1 = new Author();
            PhysicalBook physical1 = new PhysicalBook(author1, "ABC123",
"BookTitle");
            PhysicalBook physical2 = new PhysicalBook(author1, "java", "java");
            library1.addResource(physical1);
            library1.addResource(physical2);
            assertEquals(2, library1.getNumberOfResources());
        }

        /**
         * Testing the method addResources.
         */
        @Test public void addResource()
        {
            Library library1 = new Library();
            Author author1 = new Author();
            PhysicalBook physical1 = new PhysicalBook(author1, "ABC123",
"BookTitle");
            PhysicalBook physical2 = new PhysicalBook(author1, "java", "java");
            library1.addResource(physical1);
            library1.addResource(physical2);
            assertTrue(library1.containResource(physical1));
            assertTrue(library1.containResource(physical2));
        }

        /**
         * Testing the method lendBook.
         */
        @Test public void lendBook()
        {
            Library library1 = new Library();
            Author author1 = new Author();
            PhysicalBook physical1 = new PhysicalBook(author1, "ABC123",
"BookTitle");
            LibraryMember libraryM1 = new LibraryMember("memberid", "Deepak",
"membercontact");
            library1.addResource(physical1);
            library1.lendBook(physical1, libraryM1);
            assertTrue(library1.containResource(physical1));
        }

        /**
         * Testing the method returnBook.
         */
        @Test public void returnBook()
        {
            Library library1 = new Library();
            Author author1 = new Author();
            PhysicalBook physical1 = new PhysicalBook(author1, "ABC123",
"BookTitle");
            physical1.setLibMember(new LibraryMember("DR1", "DR", "dr"));
```

```
        }

        /**
         * Testing the method allPhysicalBook.
         */
        @Test public void allPhysicalBook()
        {
            Library library1 = new Library();
            Author author1 = new Author();
            PhysicalBook physical1 = new PhysicalBook(author1, "ABC123",
"BookTitle");
            PhysicalBook physical2 = new PhysicalBook(author1, "java", "java");
            ElectronicResource electron1 = new ElectronicResource("Ebook",
"Journal");
            library1.addResource(physical1);
            library1.addResource(physical2);
            library1.addResource(electron1);
            ByteArrayOutputStream outContent = new ByteArrayOutputStream();
            System.setOut(new PrintStream(outContent));
            library1.printDetails();
            System.setOut(System.out);
            String expectedOutput = "Books:\n" + "Physical Book " + "Author: \n" +
"Title: BookTitle " + "ISBN: ABC123 " + "Physical Book " + "Author: \n" + "Title:
java " + "ISBN: java";
            assertFalse(library1.toString().contains(expectedOutput));
        }

        /**
         * Testing the method allElectronicResources.
         */
        @Test public void allElectronicResources()
        {
            Library library1 = new Library();
            Author author1 = new Author();
            PhysicalBook physical1 = new PhysicalBook(author1, "ABC123",
"BookTitle");
            ElectronicResource electron1 = new ElectronicResource("ebook1",
"journal1");
            ElectronicResource electron2 = new ElectronicResource("ebook1",
"journa2");
            library1.addResource(physical1);
            library1.addResource(electron1);
            library1.addResource(electron2);
            ByteArrayOutputStream outContent = new ByteArrayOutputStream();
            System.setOut(new PrintStream(outContent));
            library1.printDetails();
            System.setOut(System.out);
            String expectedOutput = "E-Resources:\n"; expectedOutput += "Deepak";
expectedOutput += "Rawal"; expectedOutput += "Address1"; expectedOutput += "ebook1";
expectedOutput += "journal1"; expectedOutput += "ebook1"; expectedOutput += "journal2
";
            assertFalse(outContent.toString().contains(expectedOutput));
```