Define an **Interface** ILibraryRepository: void **SaveToFile**(string filePath, List<LibraryItem> items) , List<LibraryItem> **LoadFromFile(string filePath):** Loads the list of library items from a file and returns a list of LibraryItem objects.

Implement the Repository: Create a class FileLibraryRepository that implements the ILibraryRepository interface:

For loading, read the file**, split each line by commas**, and instantiate the appropriate LibraryItem objects based on the data (e.g., if the item has a **genre**, it's a **Book**; if it has a **publication date**, it's a **Journal**).

```
public void SaveToFile(String filePath, List<LibraryItem> items ) {
    try {
      FileWriter writer = new FileWriter(filePath, false);                    // Tạo đối tượng ghi vào File
      for ( LibraryItem item : items ) {                           // core
        if ( item instanceof Book ) {
          Book book = (Book) item ; // ép kiểu
          writer.write(book.getId() + ", " + book.getTitle() + ", " + book.getAuthorOrPublisher() +", " + book.getGenre() + ", " + book.isIsAvailable() );
          writer.write("\n");
        } else if ( item instanceof Journal ) {
          Journal journal = (Journal) item ;
          writer.write(journal.getId() + ", " + journal.getTitle() + ", " + journal.getAuthorOrPublisher() + ", " + journal.getPublicationDate() + ", " + journal.isIsAvailable() );
          writer.write("\n");
        }
      }
      System.out.println("Save data to File successfully");
      writer.close();
    } catch ( IOException e) {
      System.out.println("Error Save to File r kia cha" + e);
    }
List<LibraryItem> listItem = new ArrayList<>();        // Create a listItem
  @Override
  public List<LibraryItem> LoadFromFile(String filePath) {
    try {
```

```
    File inputFile = new File(filePath);             // Tên file đầu vào và read file lấy dữ liệu lên
    if ( !inputFile.exists() ) {
       throw new FileNotFoundException("File not found: " + filePath );
    }
    Scanner inputScanner = new Scanner(inputFile);
    while ( inputScanner.hasNext() ) {
       String currentLine = inputScanner.nextLine();
       String[] result = currentLine.split(", ");
String id = result[0].trim();         // Core
String title = result[1].trim();
String authorOrPublisher = result[2].trim();
String genreOrDate = result[3].trim();
boolean available = Boolean.parseBoolean(result[4].trim());
       String regex = "^\\d{4}-(0[1-9]|1[0-2])-(0[1-9]|[12]\\d|3[01])$";          // Cách search : regex
       Pattern pattern = Pattern.compile(regex);
       Matcher matcher = pattern.matcher(genreOrDate);
       if ( matcher.matches()  ) {                              // Định dạng ngày trong mảng
 DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd");
 LocalDate date = LocalDate.parse( genreOrDate , formatter );
          Journal journal1 = new Journal(id, title, authorOrPublisher, date, available)  ;
          listItem.add( journal1 );
       } else {   // nếu là thể loại
          Book book1 = new Book(id, title, authorOrPublisher, available, genreOrDate) ;
          listItem.add( book1 );
       }
    inputScanner.close();
  } catch ( IOException e ) { System.out.println("Error Load from file r kia cha" + e);    }
  return listItem;
```

```
// Hàm main
List<LibraryItem> items = new ArrayList<>();
    items.add(book);                          items.add(journal);             // Create 2 Book and Journal
FileLibraryRepository repository = new FileLibraryRepository() ;
    String filePath = "Test.txt";
    repository.SaveToFile( filePath , items );    // Save to File
    items = repository.LoadFromFile(filePath);    // Load From File to List User

for ( LibraryItem item : items ) {          // Show out all items loaded from file
      item.getDetails();
    }
```