University of Miskolc Full-time programme
Department of Information Technology

GEIAL31A-B2a
Java programming

Semester project

The goal of this project is to develop a complete Java desktop application that demonstrates the practical use of:

- object-oriented design
- abstract classes
- interfaces
- collections
- file handling
- graphical user interface programming using Swing / AWT

1. Model Layer

1.1 Abstract Class – Item

Create an abstract class called Item, which represents a generic item stored in the library.
The class must contain:

- Data members:
  - id (int)
  - title (String)
- A constructor that initializes all data members
- An abstract method: String getDescription();
- A concrete method: public int getId();

1.2 Interface – Borrowable

Create an interface called Borrowable, which describes borrowable behavior.
The interface must declare the following methods:
boolean isBorrowed();
void borrow();
void returnItem();

1.3 Class – Book

Create a class called Book that:

- extends the Item abstract class
- implements the Borrowable interface

Additional data members:

- author (String)
- price (int)
- borrowed (boolean)

The class must:

- Implement all methods declared in Borrowable
- Override the getDescription() method to return a formatted description
- Override toString() so that a book can be saved to a file in text form

2. Collection Handling

2.1 Class – Library

Create a class called Library, which stores an arbitrary number of books using an ArrayList<Book>. The class must provide the following operations:

- Add a new book to the library
- Remove a book by ID
- Get the number of stored books
- Retrieve a book by its ID
- Retrieve all books
- Retrieve only the available (not borrowed) books
- Retrieve the most expensive book

3. File Handling

3.1 Loading Books from File

- At program startup, book data must be loaded from a text file
- Use BufferedReader for file reading
- Each line in the file represents one book. Example file format:

  1;The Hobbit;Tolkien;4500;false
  2;1984;Orwell;3800;true

3.2 Saving Books into file

- When the program terminates, all book data must be saved back to a file
- Use BufferedWriter for file writing
- The saved format must be compatible with the loading format

4. Graphical User Interface

4.1 Main Window

Create a Swing-based GUI (JFrame) that contains:

- A list component (JList or JTable) displaying all books
- Buttons for the following actions:
  - Add Book
  - Delete Book
  - Borrow / Return Book
  - Load from File
  - Save to File

4.2 GUI Behavior

- Selecting a book displays its details
- Borrowing or returning a book updates its status
- The displayed list updates dynamically after each operation
- Use proper event handling (ActionListener)

5. Runnable Application Class

Create a runnable class (e.g. LibraryApp) that:

- Initializes the Library
- Loads data from the file
- Creates and displays the graphical user interface
- Saves data when the application exits

7. Optional Extensions (Bonus)

- Search by author or title
- Sorting books using Comparator
- Dialog windows (JDialog) for data input
- Exception handling with user-friendly error messages

The UML class diagram of the project: