Library Management System

Project Overview:

This Java application is a library management system that allows librarians to manage library items, authors, and patrons, along with their various sub classes. The system also allows patrons to search, borrow, and return items.

Features and Functionality:

Item Management:

- Librarians can add, edit, and delete books and periodicals from the library inventory
- Periodicals can be either printed or electronic, while books can be printed, electronic, or audio.

Author Management:

- Librarians can add, edit, and delete authors, each with a name, date of birth, and a list of written items.

Patron Management:

- Librarians can manage patrons by adding, editing, or deleting patrons, including students and employees.

Borrow and Return Library Items:

- Patrons can borrow library items if available, and the system will update the item inventory accordingly.

- Patrons can search for items by title, author, or ISBN and can return borrowed items when done.

Class Explanations:

LibraryItem: This is the base class that represents all items in the library. Some attributes include ID, title, author, ISBN, publisher, number of copies. The methods get and set details for each attribute, along with a toString to print them in a format.

Periodicals: This is a subclass of LibraryItem (extends), and it represents periodicals available in the library, either as printed or electronic types. The attribute is type, with an Enum for the PRINTED or ELECTRONIC PeriodicalType. It also has constructors, getter, setter, and a toString method.

Books: Like the periodicals, books is also a subclass of LibraryItem, and it represents books, available in printed, electronic, or audio format. The attribute is also type, with an Enum for the PRINTED, ELECTRONIC, or AUDIO BookType. It also has constructors, getter, setter, and a toString method.

Author: A class that manages author attributes such as name, date of birth, and list of library items they've written. Authors are stored in the library's author list. It has constructors, getters, setters for the attributes, and methods to add items to a list, edit an author, and another toString.

Patron: A base class for library patrons. Contains attributes like name, address, phone number, and items borrowed by the patron. It has constructors, getters, setters for the attributes, and methods to borrow and return an item, to edit a patron, along with a toString.

Student: Extends `Patron` and represents student patrons with constructors, and a toString method.

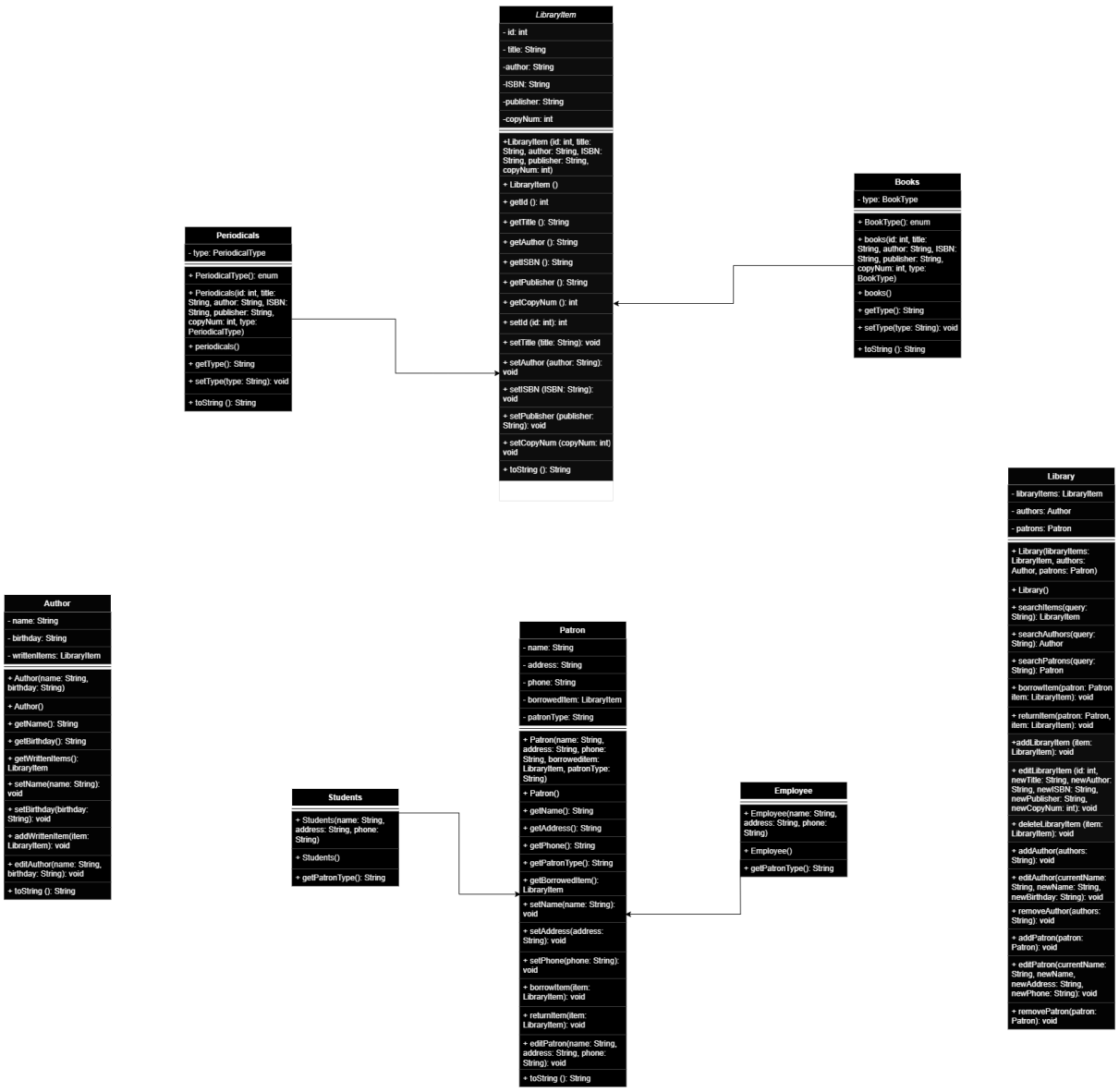
Employee: Extends `Patron` and represents employee patrons with constructors, and a toString method.

Library: The class that manages library operations for items, authors, and patrons, allowing adding, editing, searching, borrowing, and returning items, along with the ability to add, edit, and remove items, authors, and

patrons. It has attributes for LibraryItem, Author, and Patron, coupled with constructors, and methods for the operations above

LibraryDemo: The main entry point of the application. Demonstrates the core functionalities of item management, author and patron management, and borrowing and returning library items.

Class Diagram:



How to access and run the Application:

Requirements:

- Java Development Kit (JDK) 8 or higher
- A terminal or IDE (e.g., Eclipse, IntelliJ IDEA, VSCode)

Running the Application:

1. Click “Use this template” and clone the GitHub repository or download the project files.
2. Open the project in your preferred IDE or navigate to the project folder in your terminal.
3. Compile and run the Library`Demo.java` class.
4. Interact with the application through the console.

Development Documentation

Javadoc:

For documentation open the Javadoc index file located in the “doc” folder that is within the Java-Midterm-JY folder

Directory Structure:

Each file in the Java-Midterm-JY folder represents a class. Periodicals and Books extend LibraryItem, and Library, Students, and Employees extend Patron.

Build Process:

Compiling: In the terminal, navigate to Java-Midterm-JY and run the following command: javac *.java

Running: After compiling, execute the following to run the LibraryDemo class: java Demo

Compiler Dependencies:

- Java Development Kit (JDK) 8 or higher is required.
- No additional dependencies are necessary to compile and run this project.

Development Standards:

- **Code Naming Conventions**: Classes use PascalCase; variables, fields, and methods use camelCase.
- **Commenting**: Javadocs are used for each class and method, ensuring clear documentation.

Theoretical Database Design:

Within a theoretical database, we would likely have the following entities:

- **LibraryItem**: Stores item attributes like ID, title, author ID, ISBN, publisher, and number of copies.
- **Author**: Stores author details like ID, name, and date of birth.
- **Patron**: Stores patron details like ID, name, address, phone number, and borrowed items.

Entity-Relationship Diagram (ERD):

- Books to LibraryItem: subclass of LibraryItem
- Periodicals to LibraryItem: subclass of LibraryItem
- Employee to Patron: subclass of Patron

- Students to Patron: subclass of Patron
- Library to Patron: subclass of Patron

GitHub Repository:

Cloning the repository:

1. Click the “Use this template” button
2. Open your terminal
3. Run the following command: git clone <your-new-repo-url>

Deployment Documentation

Installation Steps:

Software Requirements:

- Java Development Kit (JDK) 8 or higher
- A compatible IDE (optional)

Deployment Steps:

1. Clone the repository or download the project files.
2. Open the project in your preferred IDE or terminal.
3. Compile all .java files.
4. Run the LibraryDemo.java file.

Running the Application: After running LibraryDemo.java, a console-based menu will appear. Use this to witness the system’s features.

GitHub Repository Link: https://github.com/JoshuaYouden/Java-Midterm-JY.git

End of Document