

عمل الطلاب :

1- محمد أحمد العمر 444101583

Library & Books Classes Coding & Debugging

2- مشاري ناصر الفرهود 443102320

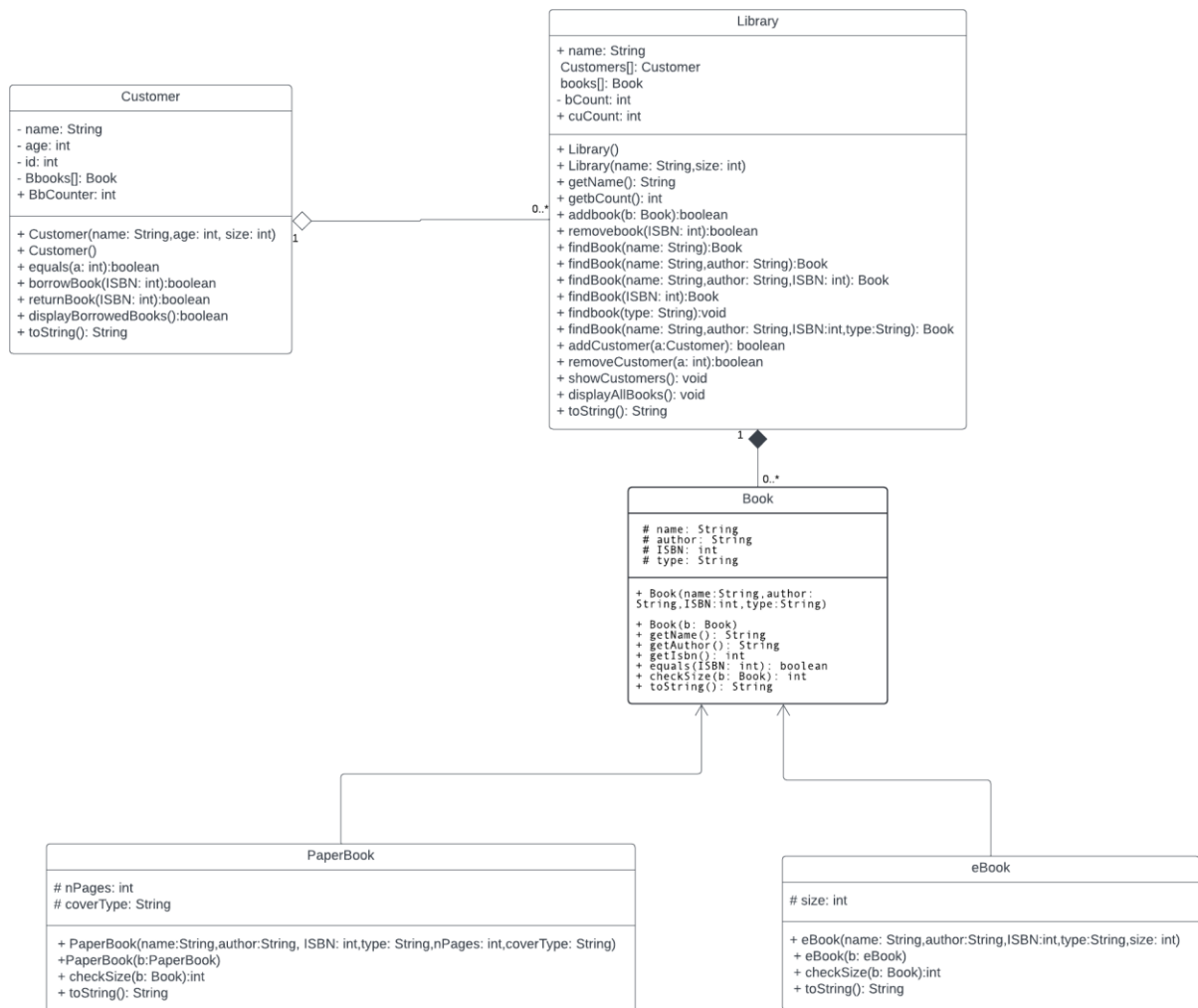
UML & Implementation and design details with describing all the methods.

3- يوسف الحامد 444101414

LibraryTest & Interface Programming & Debugging

Project idea: Library management

Project introduction: a simple Library management application to register, borrow, add, and browse books



1- Class Customer

Attributes:

- name: a String Stores the name of the customer
- age: an integer Stores the age of the customer
- id: an integer Stores the ID of the customer
- Bbooks[]: Book` : An array representing books borrowed by the customer
- BbCounter: an integer to keep track of the number of books borrowed by the customer

Methods:

- Customer(name: String, age: int, size: int)` : Constructor with parameters for initializing the customer's name, age, and the size of the `Bbooks` array
- equals(a: int): boolean` : Compares the ID of the customer with the provided ID `a`. Returns `true` if they are equal, `false` otherwise.
- borrowBook(ISBN: int): boolean` : Allows the customer to borrow a book from the library based on its ISBN. Adds the book to the customer's `Bbooks` array and removes it from the library's collection.
- returnBook(ISBN: int): boolean` : Allows the customer to return a book to the library based on its ISBN. Removes the book from the customer's `Bbooks` array and adds it back to the library's collection.
- displayBorrowedBooks(): boolean` : Displays the books currently borrowed by the customer. Returns `true` if there are borrowed books, `false` otherwise.
- toString(): String` : Returns a string representation of the `Customer` object, including the customer's name, age, ID, and the list of borrowed books.

2-Class Library

Attributes:

- name: a String Stores the name of the
- Customers[]: Customer` : An array of `Customer` objects
- books[]: Book` : An array of `Book` objects
- bCount: int` : Tracks the number of books in the library
- cuCount: int` : Tracks the number of customers in the library

Methods:

- Library()` : Default constructor to initialize the arrays

- `getName(): String`` : Returns the name of the library.
- `getbCount(): int`` : Returns the count of books in the library.
- `addbook(b: Book): boolean`` : checks if there is Space in the books array or if the book is already added
- `removebook(ISBN: int): boolean`` : Removes the given Book from the library and decrements the counter
- `findBook(name: String): Book`` : `Book`` : a method that Searches for a book in the Library using it's name
- `findBook(name: String, author: String): a method that Searches for a book in the Library using it's name & author`
- `findBook(name: String, author: String, ISBN: int): Book`` : a method that Searches for a book in the Library using it's name,author, and ISBN
- `findBook(ISBN: int): Book`` : a method that Searches for a book in the Library using it's ISBN
- `findbook(type: String): void`` : a method that Searches for a book in the Library using it's type
- `findBook(name: String, author: String, ISBN: int, type: String): Book`` : a method that Searches for a book in the Library using it's name,author,ISBN,Type
- `addCustomer(a: Customer): boolean`` : Adds a ``Customer`` object to the ``Customers`` array. Returns ``true`` if the customer was added successfully, ``false`` otherwise.
- `removeCustomer(a: int): boolean`` : Removes a customer from the library based on their ID. Returns ``true`` if the customer was removed successfully, ``false`` otherwise.
- `showCustomers(): void`` : Displays all the customers registered in the library.
- `displayAllBooks(): void`` : Displays all the books in the library.
- `toString(): String`` : Returns a string representation of the ``Library`` object, including its name, books, and customers.

3- Class Book

Attributes:

- `name`: a String Represents the name of the book.
- `author`: a String Represents the author of the book.
- `ISBN`: an intger Represents the ISBN number of the book.
- `type`: a String Represents the type of the book.

Methods:

- `Book(name: String, author: String, ISBN: int, type: String)` : Constructor method that initializes the `Book` object with the provided attributes.
- `Book(b: Book)` : Copy constructor method
- `getName(): String` : Getter method for retrieving the name of the book.
- `getAuthor(): String` : Getter method for retrieving the author of the book.
- `getIsbn(): int` : Getter method for retrieving the ISBN number of the book.
- `equals(ISBN: int): boolean` : Method to check if the provided ISBN matches the ISBN of the book.
- `checkSize(b: Book): int` : Abstract method that checks the size of Ebook and the size of PaperBook
- `toString(): String` : Method that returns a `String` representation of the `Book` object.

4- Class eBook

Attributes:

- `size`: an integer Stores the size of the eBook

Methods:

- `eBook(name: String, author: String, ISBN: int, type: String, size: int)` : Constructor with parameters for initializing the attributes of the `eBook` class.
- `eBook(b: eBook)` : Copy constructor that takes another `eBook` object as a parameter.
- `checkSize(b: Book): int` : Overrides the abstract method `checkSize` from the `Book` class. It compares the size of this `eBook` object with another `Book` object. If the other object is also an `eBook`, it returns the size of the larger eBook. Otherwise, it returns the size of the `eBook` object.
- `toString(): String` : Overrides the `toString` method from the `Book` class. It returns a string representation of the `eBook` object, including its name, author, ISBN, type, and size in megabytes.

5- Class PaperBook

Attributes:

- `nPages`: an integer Stores the number of pages of the paper book

- coverType: a String Stores the type of cover of the paper book

Methods:

- PaperBook(name: String, author: String, ISBN: int, type: String, nPages: int, coverType: String) : Constructor with parameters for initializing the attributes of the `PaperBook` class.
- PaperBook(b: PaperBook) : Copy constructor
- checkSize(b: Book): int : Overrides the abstract method `checkSize` from the `Book` class. It compares the number of pages of this `PaperBook` object with another `Book` object. If the other object is also a `PaperBook`, it returns the number of pages of the larger book. Otherwise, it returns the number of pages of the `PaperBook` object.
- toString(): String : Overrides the `toString` method from the `Book` class. It returns a string representation of the `PaperBook` object, including its name, author, ISBN, type, number of pages, and cover type.

Sample Run:

```
Welcome to Barnes & Noble! Select an option from the menu:
1. Add Book
2. Remove Book
3. Find Book
4. Display All Books.
5. Borrow Book
6. Return Book
7. Display All Borrowed Books
8. Exit
```