Bài 1

1. Lớp Book (Sách)

* Vai trò: Đại diện cho một cuốn sách trong thư viện.
* Thuộc tính chính:
  + code (Mã sách): String
  + title (Tên sách): String
  + author (Tác giả): String
  + publicationYear (Năm xuất bản): int
  + publisher (Nhà xuất bản): String (tuỳ chọn)
  + copiesAvailable (Số bản đang có sẵn): int
  + totalCopies (Tổng số bản): int
* Phương thức chính:
  + isAvailable(): boolean
  + checkOut(): boolean (giảm copiesAvailable)
  + returnCopy(): void (tăng copiesAvailable)
  + addCopy(): void (tăng totalCopies và copiesAvailable)
  + removeCopy(): void (giảm tổng số bản và có thể cập nhật copiesAvailable)

1. Lớp Reader (Độc giả)

* Vai trò: Đại diện cho người dùng có quyền mượn sách.
* Thuộc tính chính:
  + readerId (Mã độc giả): String
  + name (Tên): String
  + email (Email): String
  + phone (Số điện thoại): String
  + address (Địa chỉ): String
  + issuedBooks (Danh sách sách đã mượn): List<BookBorrowRecord>
  + maxBorrowLimit (Giới hạn mượn): int
* Phương thức chính:
  + canBorrow(): boolean (kiểm tra còn chỗ trống trong giới hạn)
  + borrowBook(book: Book): boolean (thêm vào issuedBooks nếu hợp lệ)
  + returnBook(bookCode: String): boolean (gỡ khỏi issuedBooks)
  + getOverdueCount(): int (nếu có quản lý ngày mượn/trả)

1. Lớp Librarian (Nhân viên thư viện)

* Vai trò: Quản lý vận hành hệ thống, xử lý các thao tác quản trị và hỗ trợ độc giả.
* Thuộc tính chính:
  + staffId (Mã nhân viên): String
  + name (Tên): String
  + role (Vai trò trong thư viện): String
  + position (Chức danh): String
* Phương thức chính:
  + addBook(book: Book): void
  + removeBook(bookCode: String): boolean
  + updateBook(book: Book): boolean
  + registerReader(reader: Reader): void
  + lendBook(readerId: String, bookCode: String): boolean
  + receiveReturn(readerId: String, bookCode: String): boolean

1. Lớp BorrowRecord (Đơn mượn sách)

* Vai trò: Ghi nhận chi tiết một lần mượn sách giữa độc giả và sách.
* Thuộc tính chính:
  + recordId (Mã phiếu): String
  + reader (Reader)
  + book (Book)
  + borrowDate (Ngày mượn): Date
  + dueDate (Ngày trả dự kiến): Date
  + returnDate (Ngày trả thực tế): Date (nếu đã trả)
  + status (Trạng thái: Mượn/Đã trả/Quá hạn)
* Phương thức chính:
  + isOverdue(): boolean
  + markReturned(returnDate): void

1. Lớp LoanPolicy (Chính sách mượn)

* Vai trò: Quản lý các quy định liên quan đến mượn và trả (giới hạn số sách, thời hạn mượn, phí phạt…).
* Thuộc tính chính:
  + maxBorrowLimit (Giới hạn mượn cho mỗi độc giả)
  + loanPeriodDays (Thời hạn mượn theo ngày)
  + finePerDay (Phí phạt mỗi ngày quá hạn)
* Phương thức chính:
  + isWithinLimit(reader: Reader, book: Book): boolean
  + calculateFine(overdueDays: int): double

Bài 2

|  |  |  |  |
| --- | --- | --- | --- |
| **Mối quan hệ** | **Loại quan hệ** | **Giải thích** | **Ký hiệu UML** |
| Một giáo viên giảng dạy nhiều lớp học | Association (kết hợp) | Hai thực thể có liên kết chức năng, có thể tồn tại độc lập. Giáo viên và lớp học vẫn tồn tại nếu xóa quan hệ dạy–học. | — hoặc đường nối thường, có thể ghi multiplicity 1..\* |
| Một đơn hàng gồm nhiều sản phẩm | Aggregation (kết tập) | “Có–thuộc” nhưng lỏng lẻo: sản phẩm vẫn tồn tại ngay cả khi đơn hàng bị hủy. Phần–tổng nhưng không sống chết cùng nhau. | ◇ (hình thoi rỗng ở phía “đơn hàng”) |
| Một cơ thể gồm nhiều bộ phận không thể tách rời | Composition (thành phần) | Quan hệ toàn–phần mạnh, sinh mệnh phụ thuộc: bộ phận không thể tồn tại nếu không có cơ thể. Xóa cơ thể → bộ phận cũng mất. | ◆ (hình thoi đặc ở phía “cơ thể”) |

Bài 3

|  |  |  |
| --- | --- | --- |
| **Thành phần** | **Modifier** | **Giải thích** |
| username | public hoặc protected (tùy vào mục đích dùng ngoài class) | Nếu cần cho các lớp khác truy cập và hiển thị thông tin người dùng, để ở public. Nếu chỉ dùng trong cùng hoặc các lớp con, có thể để protected. Tránh để private nếu cần truy cập từ bên ngoài trong các module/đối tượng khác. |
| password | private | Bảo vệ an toàn thông tin nhạy cảm, không cho truy cập trực tiếp từ bên ngoài. Phương thức thao tác với password (ví dụ đổi/mật khẩu, xác thực) nên thông qua các phương thức công khai khi cần. |
| login() | public | Cho phép người dùng hoặc hệ thống gọi để đăng nhập từ bất kỳ đâu trong chương trình. |
| resetPassword() | public (hoặc protected tùy ngữ cảnh) | public nếu cần cho người dùng hoặc hệ thống tự động gọi để thiết lập lại mật khẩu; có thể để protected nếu chỉ gọi từ bên trong class hoặc từ các lớp con trong kế thừa (ví dụ khi có luồng xác thực/khôi phục mật khẩu nội tại). |
| lastLoginTime | private hoặc protected (nếu chỉ đọc nội bộ) | private để bảo vệ dữ liệu đăng nhập cuối cùng và chỉ cho phép thông qua phương thức getter công khai. Nếu bạn muốn các lớp con có thể đọc được mà không sửa, có thể để protected hoặc cung cấp public getter. |

Bài 4

A screenshot of a computer

AI-generated content may be incorrect.

|  |  |  |
| --- | --- | --- |
| Mô tả chức năng từng lớp |  |  |
| Lớp | Vai trò | Chức năng chính |
| Product | Đại diện cho sản phẩm trong cửa hàng | Lưu thông tin sản phẩm (giá, tên...), cập nhật giá, hiển thị thông tin. |
| Customer | Người mua hàng | Lưu thông tin khách hàng, tạo/tạo mới đơn hàng. |
| Order | Đơn hàng do khách hàng tạo | Gồm nhiều sản phẩm, tính tổng số tiền, thêm sản phẩm vào đơn hàng. |

Bài 5

Chọn lớp: Sách

Dưới đây là bảng thuộc tính và phương thức cho lớp Sách (Book) theo mô hình UML đơn giản bằng văn bản.

Thuộc tính (Attributes)

* title: String — tiêu đề sách
* author: String — tác giả
* isbn: String — mã ISBN duy nhất
* publishedYear: int — năm xuất bản
* genre: String — thể loại
* availableCopies: int — số bản còn có sẵn

Phương thức (Methods)

* getTitle(): String — trả về tiêu đề sách
* getAuthor(): String — trả về tác giả
* isAvailable(): boolean — kiểm tra sách còn có sẵn để cho mượn
* borrowCopy(): void — cho mượn một bản sách (giảm availableCopies nếu có sẵn)
* returnCopy(): void — trả lại một bản sách (tăng availableCopies)
* setGenre(genre: String): void — cập nhật thể loại sách

Gợi ý cách trình bày UML (text)

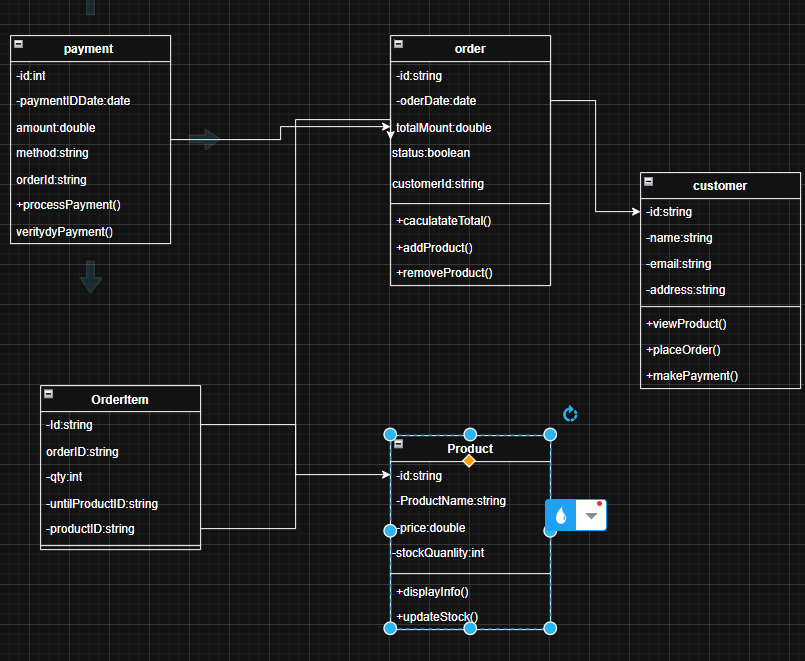
* Class Sách
  + Attributes:
    - * title: String
      * author: String
      * isbn: String
      * publishedYear: int
      * genre: String
      * availableCopies: int
  + Methods:
    - * getTitle(): String
      * getAuthor(): String
      * isAvailable(): boolean
      * borrowCopy(): void
      * returnCopy(): void
      * setGenre(genre: String): void

Bài 6

A screenshot of a computer

AI-generated content may be incorrect.

Bài 7



Bài 8

A computer screen shot of a computer screen

AI-generated content may be incorrect.

Bài 9

|  |  |  |
| --- | --- | --- |
| **Loại lỗi** | **Mô tả lỗi** | **Gợi ý sửa** |
| Multiplicity | Đang để User 1 — 1 Order, trong khi thực tế một User có thể có nhiều Order | Sửa thành: User 1 — \* Order |
| Relationship | Dùng mũi tên dependency → không đúng bản chất. Quan hệ giữa User và Order phải là Association / Aggregation / Composition | Dùng đường line với diamond nếu Order thuộc User (aggregation hoặc composition) |
| Modifier (phạm vi truy cập) | Các thuộc tính như password, email, totalAmount đang để + (public) → vi phạm bảo mật và OOP | Đổi thành - password, - email, - totalAmount (private) |
| Thiếu kiểu dữ liệu | Các attribute không có kiểu (String, Date, Double, …) | Bổ sung: username: String, OrderDate: Date |
| Logic nghiệp vụ thiếu | Order không có OrderItem, khó tính totalAmount đúng chuẩn | Thêm class OrderItem + quan hệ với Order |

A computer screen shot of a computer code

AI-generated content may be incorrect.