# BÀI 1: Nhận diện Class từ bài toán

|  |  |
| --- | --- |
| Tên lớp (Class) | Vai trò / Mô tả |
| Book (Sách) | Quản lý thông tin của từng cuốn sách trong thư viện, gồm các thuộc tính như mã sách, tên sách, tác giả, năm xuất bản. |
| Reader (Độc giả) | Đại diện cho người mượn hoặc trả sách trong hệ thống. Lưu thông tin cá nhân của độc giả. |
| Librarian (Nhân viên thư viện) | Đại diện cho người quản lý thư viện, chịu trách nhiệm nhập sách, cho mượn và nhận trả sách. |
| Borrowing (Phiếu mượn / Giao dịch mượn sách) | Lưu thông tin chi tiết của mỗi lần độc giả mượn hoặc trả sách (ngày mượn, ngày trả, trạng thái). |

# Bài 2: Phân biệt các loại quan hệ giữa class

|  |  |  |  |
| --- | --- | --- | --- |
| **Mối quan hệ** | **Loại** | **Giải thích** | **Ký hiệu UML** |
| Giáo viên – Lớp học (Một giáo viên giảng dạy nhiều lớp học.) | Association (Kết hợp) | Hai lớp liên kết về mặt nghiệp vụ nhưng tồn tại độc lập. Xóa giáo viên, lớp học vẫn tồn tại (và ngược lại). | Đường nối bình thường (không hình thoi). |
| Đơn hàng – Sản phẩm (Một đơn hàng gồm nhiều sản phẩm.) | Aggregation (Kết tập/Tổng hợp) | Quan hệ toàn-bộ lỏng. Sản phẩm là thành phần nhưng có thể tồn tại độc lập ngoài đơn hàng. Xóa đơn hàng, sản phẩm vẫn còn. | Hình thoi rỗng ở phía 'toàn bộ' (Đơn hàng) — nối tới thành phần (Sản phẩm). |
| Cơ thể – Bộ phận (Một cơ thể gồm nhiều bộ phận không thể tách rời.) | Composition (Hợp thành) | Quan hệ toàn-bộ chặt. Bộ phận không thể tồn tại độc lập. Xóa cơ thể thì các bộ phận bị hủy theo. | Hình thoi đặc ở phía 'toàn bộ' (Cơ thể) — nối tới thành phần (Bộ phận). |

# Bài 3: Phân tích phạm vi truy cập

|  |  |  |
| --- | --- | --- |
| **Thành phần** | **Modifier** | **Giải thích** |
| username | private | Dữ liệu thuộc về người dùng, không cho truy cập trực tiếp từ bên ngoài để đảm bảo tính đóng gói; truy cập qua getter/setter nếu cần. |
| password | private | Thông tin nhạy cảm, phải ẩn hoàn toàn; chỉ được thao tác qua phương thức (hash/verify), không lộ ra ngoài. |
| lastLoginTime | protected | Cho phép lớp con (ví dụ AdminUser) đọc/cập nhật phục vụ audit/report; bên ngoài không được đụng tới. |
| login() | public | Hành vi cần gọi từ bên ngoài hệ thống (UI/Service) để xác thực đăng nhập. |
| resetPassword() | public | Được gọi từ luồng quên mật khẩu/ quản trị; triển khai kiểm tra/OTP bên trong, không cho truy cập dữ liệu thô. |

# Bài 4: Vẽ sơ đồ Class đơn giản

|  |  |  |  |
| --- | --- | --- | --- |
| Lớp | Thuộc tính | Phương thức | Chức năng mô tả |
| Product | - productId: String - name: String - price: double - stockQty: int | - getPrice() - updateStock(qty) - toString() | Lưu thông tin sản phẩm và hỗ trợ cập nhật tồn kho, tra cứu giá. |
| Customer | - customerId: String - fullName: String - phone: String - address: String | - getContactInfo() - updateAddress(addr) | Lưu thông tin khách hàng và hỗ trợ liên hệ/cập nhật địa chỉ. |
| Order | - orderId: String - orderDate: Date - totalAmount: double | - addProduct(product, qty) - removeProduct(product) - calculateTotal() | Đại diện đơn hàng của khách; quản lý danh sách sản phẩm, tính tổng tiền. |

Quan hệ giữa các lớp (Association) và bội số:

|  |  |  |  |
| --- | --- | --- | --- |
| Mối quan hệ | Loại | Bội số UML | Giải thích |
| Customer — Order | Association | Customer (1) ↔ Order (0..\*) | Một khách hàng có thể có nhiều đơn hàng; mỗi đơn hàng thuộc về đúng 1 khách hàng. |
| Order — Product | Association (nhiều-nhiều) | Order (\* ) ↔ Product (\* ) | Một đơn hàng gồm nhiều sản phẩm; một sản phẩm có thể nằm trong nhiều đơn hàng. |

* Vẽ sơ đồ:

A diagram with text on it

AI-generated content may be incorrect.

# Bài 5: Xác định thuộc tính và phương thức

|  |  |  |
| --- | --- | --- |
| **Loại** | **Tên** | **Mô tả / Vai trò** |
| Thuộc tính | orderId | Mã đơn hàng duy nhất để phân biệt từng đơn hàng. |
| Thuộc tính | orderDate | Ngày tạo đơn hàng. |
| Thuộc tính | totalAmount | Tổng số tiền của đơn hàng. |
| Thuộc tính | status | Trạng thái đơn hàng (ví dụ: Đang xử lý, Hoàn thành, Đã hủy). |
| Phương thức | addProduct() | Thêm sản phẩm vào đơn hàng. |
| Phương thức | calculateTotal() | Tính tổng số tiền dựa trên các sản phẩm trong đơn hàng. |
| Phương thức | updateStatus() | Cập nhật trạng thái đơn hàng (ví dụ: từ 'Đang xử lý' sang 'Hoàn thành'). |

Mô hình UML (dạng text):

Class: Order  
-----------------------  
- orderId  
- orderDate  
- totalAmount  
- status  
-----------------------  
+ addProduct()  
+ calculateTotal()  
+ updateStatus()

# Bài 6: Thêm Multiplicity vào sơ đồ

|  |  |  |  |
| --- | --- | --- | --- |
| **Mối quan hệ** | **Loại quan hệ** | **Bội số UML** | **Giải thích** |
| Giảng viên – Lớp học | Association | Giảng viên (1) ↔ Lớp học (0..\*) | Một giảng viên có thể dạy nhiều lớp, nhưng mỗi lớp chỉ có một giảng viên chính phụ trách. |
| Lớp học – Sinh viên | Association | Lớp học (1) ↔ Sinh viên (0..\*) | Một lớp có nhiều sinh viên, mỗi sinh viên thuộc ít nhất một lớp. |
| Sinh viên – Lớp học | Association (N–N) | Sinh viên (0..\*) ↔ Lớp học (0..\*) | Một sinh viên học nhiều lớp, và một lớp có nhiều sinh viên (mối quan hệ nhiều–nhiều). |

-Vẽ sơ đồ:

A white box with black text

AI-generated content may be incorrect.

# Bài 7: Từ Use Case → Class Diagram

|  |  |  |
| --- | --- | --- |
| Use Case | Class chính liên quan | Hành vi/Phương thức tiêu biểu |
| Xem sản phẩm | Product | getInfo() |
| Đặt hàng | Order, Customer | Customer.placeOrder(); Order.addProduct(); Order.submit() |
| Thanh toán | Payment, Order | Payment.process(); Order.calculateTotal() |

Danh sách Class (thuộc tính & phương thức cơ bản):

|  |  |  |
| --- | --- | --- |
| Class | Thuộc tính | Phương thức |
| Customer | - customerId - fullName - phone - address | - viewProduct() - placeOrder() - pay(order) |
| Product | - productId - name - price - stockQty | - getInfo() - updateStock(qty) |
| Order | - orderId - orderDate - totalAmount - status | - addProduct(product, qty) - calculateTotal() - submit() |
| Payment | - paymentId - amount - method - status | - process() - refund() |

Quan hệ giữa các lớp (Association) và bội số UML:

|  |  |  |  |
| --- | --- | --- | --- |
| Mối quan hệ | Loại | Bội số UML | Giải thích |
| Customer — Order | Association | Customer (1) ↔ Order (0..\*) | Một khách hàng có thể tạo nhiều đơn hàng; mỗi đơn hàng thuộc về một khách hàng. |
| Order — Product | Association (N–N) | Order (\*) ↔ Product (\*) | Một đơn hàng có nhiều sản phẩm; một sản phẩm có thể xuất hiện trong nhiều đơn hàng. |
| Order — Payment | Association | Order (1) ↔ Payment (0..1) | Mỗi đơn hàng có tối đa một thanh toán; trước khi trả tiền có thể là 0. |

* Vẽ sơ đồ:

A diagram of a company

AI-generated content may be incorrect.

# Bài 8: Thiết kế mô hình lớp cho hệ thống đăng ký học phần

A diagram of a student's program

AI-generated content may be incorrect.

# Bài 9: Tìm lỗi trong sơ đồ class

|  |  |  |
| --- | --- | --- |
| **Nhóm lỗi** | **Mô tả lỗi trong sơ đồ** | **Cách sửa** |
| Quan hệ (Relationship) | Không có đường liên kết giữa User và Order nên không thể hiện ai sở hữu đơn hàng. | Thêm association: User (1) ↔ Order (0..\*). |
| Multiplicity | Không ghi bội số ở hai đầu quan hệ (và thậm chí chưa có quan hệ). | Ghi rõ multiplicity: User (1) — Order (0..\*). |
| Modifier (Access) | Tất cả thuộc tính đang đánh dấu '+' (public) — sai nguyên tắc đóng gói. | Đổi thuộc tính sang '-' (private). Phương thức giữ '+'. |
| Quy ước đặt tên | Tên thuộc tính lẫn lộn kiểu: 'OrderId', ' OrderDate' (thừa khoảng trắng), 'totalAmount' (camelCase). | Chuẩn hóa camelCase: orderId, orderDate, totalAmount; username, password, email. |
| Thiếu tối thiểu | Order thiếu chủ sở hữu (user) ở cấp cấu trúc (thể hiện qua quan hệ). | Bổ sung association như trên; không thêm thuộc tính ngoài chương trình. |

* Vẽ sơ đồ:

A black line with numbers

AI-generated content may be incorrect.