Bai1:

Hệ thống quản lý thư viện có 3 đối tượng chính:

* **Sách**
* **Độc giả**
* **Nhân viên thư viện**

Chức năng chính:

* Độc giả **mượn** và **trả sách**.
* Mỗi sách có: mã, tên, tác giả, năm xuất bản.

| **Tên lớp** | **Vai trò / Mô tả** |
| --- | --- |
| **Sach** | Biểu diễn thông tin của mỗi cuốn sách trong thư viện. |
| **DocGia** | Lưu thông tin người đọc, người có thể mượn hoặc trả sách. |
| **NhanVien** | Đại diện cho nhân viên thư viện, người quản lý việc mượn/trả và cập nhật kho sách. |
| **PhieuMuon** | Thể hiện thông tin mượn sách (ai mượn, mượn cuốn nào, ngày mượn – ngày trả). |
| **ThuVien** *(tùy chọn)* | Quản lý tập hợp sách, độc giả và phiếu mượn (đóng vai trò lớp điều phối hệ thống). |

### **1. Lớp Sach**

* **Thuộc tính**: maSach, tenSach, tacGia, namXuatBan, tinhTrang (ví dụ: còn / đã mượn)
* **Vai trò**: Lưu thông tin chi tiết của từng cuốn sách.
* **Hành vi (phương thức)**: capNhatTinhTrang(), hienThiThongTin()

### **2. Lớp DocGia**

* **Thuộc tính**: maDocGia, tenDocGia, diaChi, soDienThoai
* **Vai trò**: Đại diện cho người mượn sách.
* **Hành vi**: muonSach(), traSach()

### **3. Lớp NhanVien**

* **Thuộc tính**: maNV, tenNV, chucVu
* **Vai trò**: Quản lý hoạt động mượn/trả, thêm hoặc xóa sách khỏi hệ thống.
* **Hành vi**: lapPhieuMuon(), nhanTraSach(), capNhatSach()

### **4. Lớp PhieuMuon**

* **Thuộc tính**: maPhieu, ngayMuon, ngayTra, maDocGia, maSach, tinhTrangTra
* **Vai trò**: Ghi lại lịch sử mượn/trả của từng độc giả.
* **Hành vi**: xacNhanTra(), kiemTraQuaHan()

### **5. Lớp ThuVien *(tùy chọn)***

* **Thuộc tính**: danhSachSach, danhSachDocGia, danhSachNhanVien
* **Vai trò**: Lớp trung tâm, điều phối các chức năng quản lý.
* **Hành vi**: timSach(), themSach(), quanLyPhieuMuon()

Bai2:

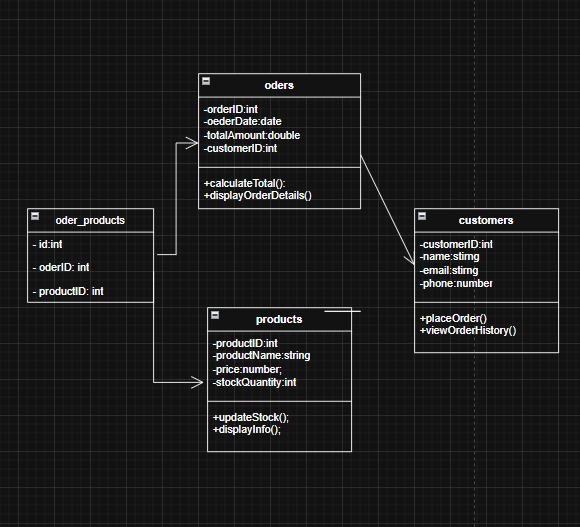
| **Mối quan hệ** | **Loại quan hệ** | **Giải thích** | **Ký hiệu UML** |
| --- | --- | --- | --- |
| **Một giáo viên giảng dạy nhiều lớp học.** | **Association (Liên kết)** | Là quan hệ **liên kết lỏng lẻo**: giáo viên và lớp học có thể tồn tại **độc lập**. Một giáo viên có thể dạy nhiều lớp, và một lớp cũng có thể do nhiều giáo viên dạy (tùy hệ thống). Nếu giáo viên nghỉ việc, lớp học vẫn tồn tại. | Đường nối **bình thường** giữa hai lớp, có thể ghi **bội số (1..\*, 1..n)** |
| **Một đơn hàng gồm nhiều sản phẩm.** | **Aggregation (Kết tập)** | Là quan hệ **“có – chứa” (has-a)** nhưng **các đối tượng con vẫn có thể tồn tại độc lập**. Sản phẩm có thể tồn tại mà không cần đơn hàng (vẫn nằm trong kho). Nếu đơn hàng bị xóa, sản phẩm không bị xóa khỏi hệ thống. | Đường nối với **hình thoi rỗng (◇)** ở phía **lớp “chủ sở hữu”** (Đơn hàng ◇— Sản phẩm) |
| **Một cơ thể gồm nhiều bộ phận không thể tách rời.** | **Composition (Cấu thành)** | Là quan hệ **“phụ thuộc chặt chẽ” (part-of)**. Bộ phận không thể tồn tại nếu cơ thể bị hủy — khi cơ thể mất đi, các bộ phận cũng không còn. | Đường nối với **hình thoi đen (◆)** ở phía **lớp “toàn thể”** (Cơ thể ◆— Bộ phận) |

Bai3:

| **Thành phần** | **Modifier (Phạm vi truy cập)** | **Giải thích** |
| --- | --- | --- |
| username | private | Dữ liệu cá nhân, chỉ nên được truy cập thông qua phương thức getter/setter để đảm bảo an toàn và tránh sửa trực tiếp. |
| password | private | Rất nhạy cảm, chỉ lớp User được phép truy cập và xử lý, không được truy cập trực tiếp từ bên ngoài. |
| lastLoginTime | private | Là thông tin hệ thống, không cần cho bên ngoài biết; có thể cho phép đọc qua phương thức getLastLoginTime(). |
| login() | public | Là hành vi mà các đối tượng khác (ví dụ: hệ thống, giao diện đăng nhập) cần gọi để xác thực người dùng. |
| resetPassword() | public | Cho phép gọi từ bên ngoài (ví dụ: trang “Quên mật khẩu”), nên để công khai để người dùng hoặc hệ thống có thể yêu cầu đổi mật khẩu. |

Bai4:

| **Lớp** | **Vai trò / Chức năng** |
| --- | --- |
| **Product** | Lưu trữ thông tin sản phẩm, cập nhật số lượng tồn kho. |
| **Customer** | Đại diện cho khách hàng, thực hiện đặt hàng và xem lịch sử mua. |
| **Order** | Quản lý thông tin đơn hàng, tính tổng tiền và liệt kê sản phẩm trong đơn. |



Bai5:

**Lớp: Độc giả (DocGia)**

**1. Thuộc tính (Attributes)**

| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| --- | --- | --- |
| **maDocGia** | **String** | **Mã định danh duy nhất của độc giả.** |
| **tenDocGia** | **String** | **Họ và tên của độc giả.** |
| **diaChi** | **String** | **Địa chỉ nơi cư trú của độc giả.** |
| **soDienThoai** | **String** | **Số điện thoại liên hệ.** |
| **soSachDangMuon** | **int** | **Số lượng sách hiện độc giả đang mượn.** |

**2. Phương thức (Methods)**

| **Tên phương thức** | **Kiểu trả về** | **Mô tả** |
| --- | --- | --- |
| **muonSach()** | **void** | **Thực hiện thao tác mượn sách, cập nhật số sách đang mượn.** |
| **traSach()** | **void** | **Thực hiện trả sách, giảm số lượng sách đang mượn.** |
| **xemThongTin()** | **void** | **Hiển thị thông tin chi tiết của độc giả.** |
| **kiemTraQuaHan()** | **boolean** | **Kiểm tra xem độc giả có sách mượn quá hạn hay không.** |

**Sơ đồ đơn giản dạng text**

**-------------------------**

**| DocGia |**

**-------------------------**

**| - maDocGia: String**

**| - tenDocGia: String**

**| - diaChi: String**

**| - soDienThoai: String**

**| - soSachDangMuon: int**

**-------------------------**

**| + muonSach(): void**

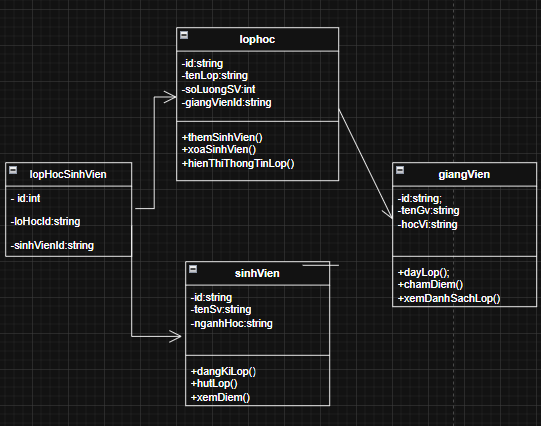
**| + traSach(): void**

**| + xemThongTin(): void**

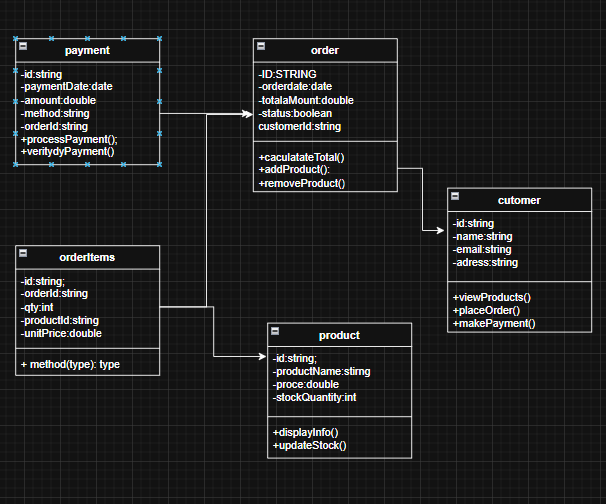
**| + kiemTraQuaHan(): boolean**

**-------------------------**

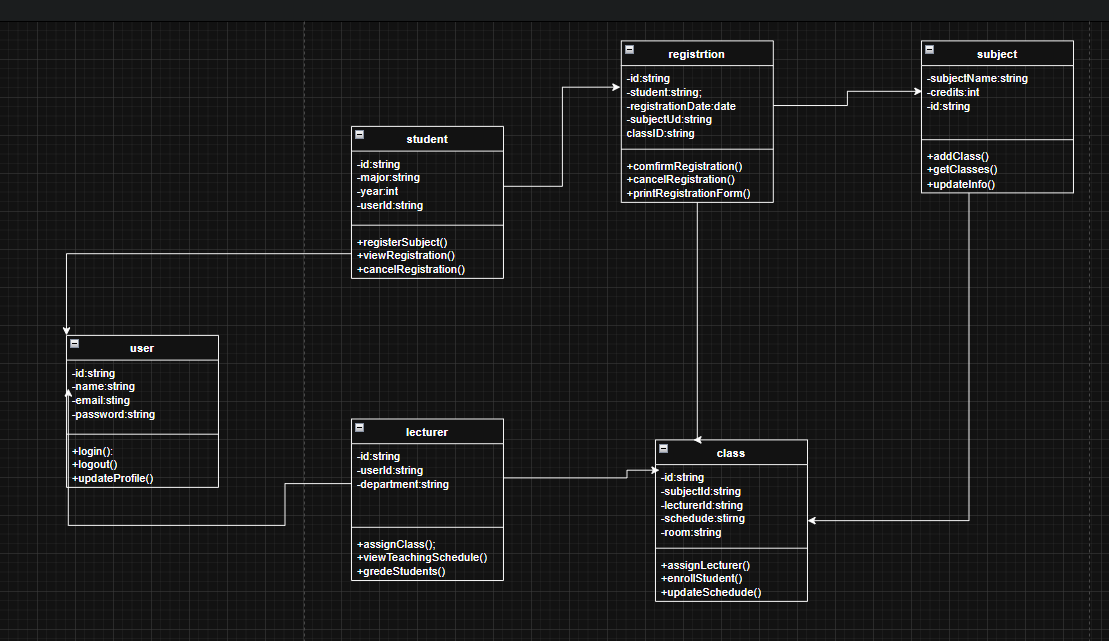
**Bai6:**

****

**Bai7:**

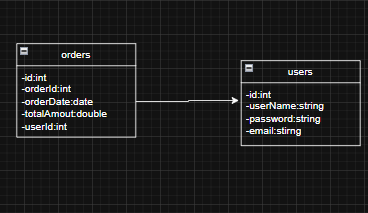
****

**Bai8:**

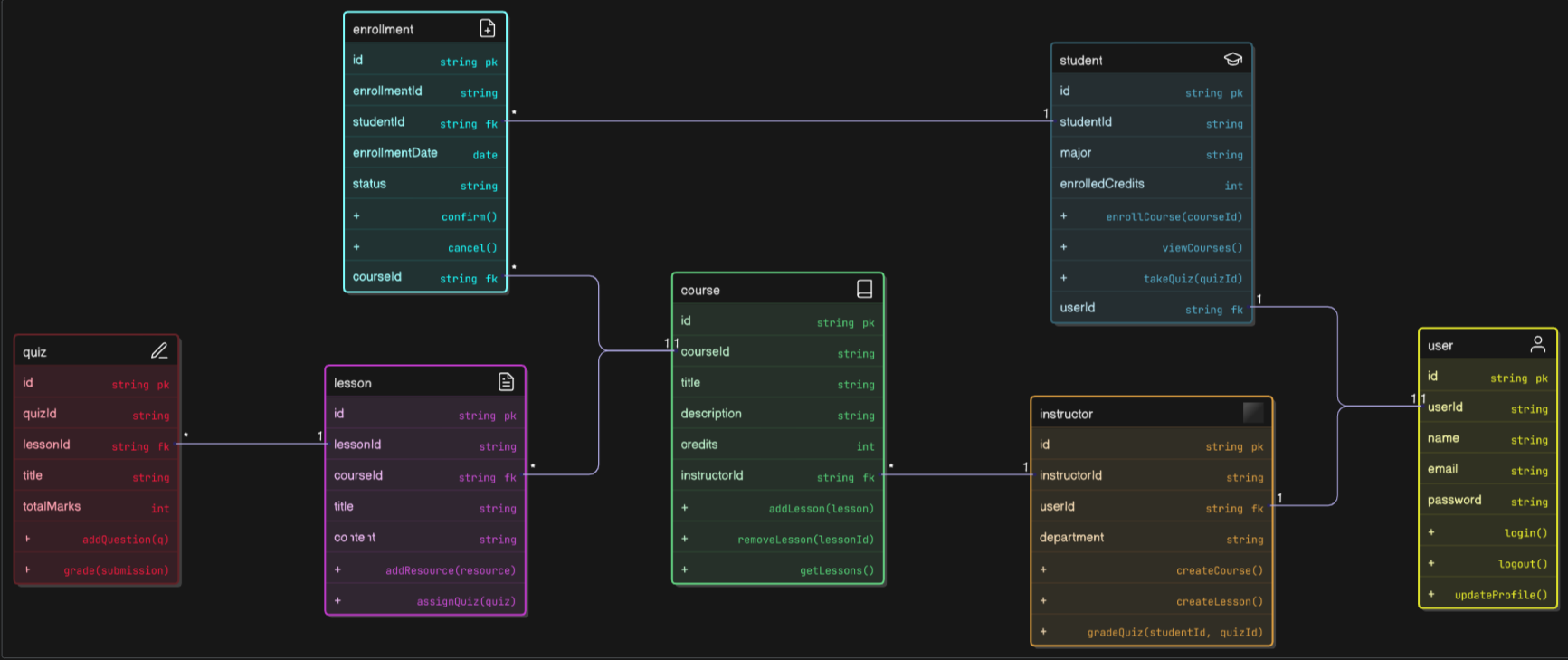
****

**Bai9:**

| **Loại lỗi** | **Mô tả lỗi chi tiết** | **Giải thích / Cách sửa** |
| --- | --- | --- |
| **Lỗi quan hệ giữa các lớp (Relationship)** | **Quan hệ giữa User và Order được biểu diễn là 1–1, tức là một người dùng chỉ có một đơn hàng.** | **Thực tế: Một người dùng có thể có nhiều đơn hàng. Quan hệ đúng là 1–\* (One-to-Many).** |
| **Lỗi Multiplicity (Bội số)** | **Cả hai đầu đều là 1, nghĩa là mỗi Order cũng chỉ thuộc một User, điều này đúng; nhưng đầu User phải là 1..\* hoặc 1–\* để thể hiện “một user có nhiều order”.** | **Sửa lại: User (1) --- (0..\*) Order.** |
| **Lỗi Modifier (Phạm vi truy cập)** | **Tất cả các thuộc tính (username, password, email, OrderId, ...) đều để là + (public).** | **Trong UML chuẩn: thuộc tính nên là private (-), phương thức public (+).** |
| **Lỗi đặt tên / nhất quán thuộc tính** | **OrderId và OrderDate nên viết đúng quy tắc đặt tên: orderId, orderDate.** | **UML không quy định chặt, nhưng giữ chuẩn *camelCase* giúp nhất quán và rõ ràng.** |
| **Thiếu quan hệ hướng dẫn logic** | **Order không có tham chiếu ngược tới User.** | **Thêm thuộc tính userId hoặc thể hiện liên kết trực tiếp.** |

****

**Bai10:**

****