**Bài 1:**

Sách(book)

Vai trò: Đại diện cho một cuốn sách trong thư viện, lưu trữ thông tin chi tiết của sách.

Thuộc tính ví dụ:maSach, tenSach, macGia, namXuatBan, trangThai,

Độc giả (Reader)

Vai trò: Đại diện cho người dùng thư viện, có thể mượn và trả sách.

Thuộc tính ví dụ:maDocGia, hoTenemail, soDienThoai

Nhân Viên Thư Viện(Librarian**)**

Vai trò: Đại diện cho nhân viên quản lý thư viện, hỗ trợ quản lý sách và giao dịch mượn trả.

Thuộc tính ví dụ:maNhanVien, hoTen ,chucVu,

Phiếu Mượn(Borrow Transaction)

Vai trò: Ghi nhận một lần mượn sách, liên kết độc giả với sách và ngày mượn.

Thuộc tính ví dụ: maPhieuMuon, maDocGia, maSach, ngayMuon, ngayHanTra

**Bài 2:**

Mối quan hệ: Một giáo viên giảng dạy nhiều lớp học.

Loại: Association

Giải thích: Giáo viên và lớp học tồn tại độc lập, một giáo viên dạy nhiều lớp, một lớp có thể có nhiều giáo viên theo thời gian. Không ai sở hữu ai.

Ký hiệu UML: Đường thẳng nối giữa hai lớp, ghi số lượng (ví dụ: 1 giáo viên – \* lớp học).

Mối quan hệ: Một đơn hàng gồm nhiều sản phẩm.

Loại: Aggregation

Giải thích: Đơn hàng chứa sản phẩm, nhưng sản phẩm vẫn tồn tại độc lập (có thể dùng cho nhiều đơn hàng). Quan hệ “có” nhưng không bắt buộc cùng sống cùng chết.

Ký hiệu UML: Đường thẳng với hình thoi rỗng ở phía Đơn hàng.

Mối quan hệ: Một cơ thể gồm nhiều bộ phận không thể tách rời.

Loại: Composition

Giải thích: Bộ phận chỉ tồn tại trong cơ thể, nếu cơ thể mất thì bộ phận cũng mất. Quan hệ “có” chặt chẽ, cùng vòng đời.

Ký hiệu UML: Đường thẳng với hình thoi đặc ở phía Cơ thể.

**Bài 3:**

Thành phần: username

Modifier: private

Giải thích: Tên người dùng là thông tin nhạy cảm, chỉ nên truy cập qua phương thức (getter/setter) để kiểm soát dữ liệu, tránh thay đổi trực tiếp.

Thành phần: password

Modifier: private

Giải thích: Mật khẩu là dữ liệu cực kỳ nhạy cảm, phải được mã hóa và chỉ cho phép thay đổi qua phương thức bảo mật như resetPassword(), không cho truy cập trực tiếp.

Thành phần: login()

Modifier: public

Giải thích: Phương thức đăng nhập cần được gọi từ bên ngoài (giao diện, hệ thống khác), nên để public để người dùng hoặc hệ thống có thể sử dụng.

Thành phần: resetPassword()

Modifier: public

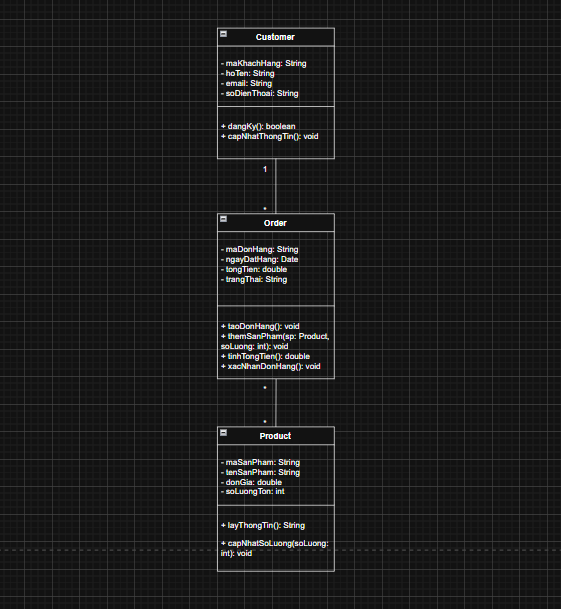
Giải thích: Người dùng cần có cách khôi phục hoặc đổi mật khẩu từ bên ngoài, nên để public. Có thể yêu cầu xác thực trước khi thực thi.

Thành phần: lastLoginTime

Modifier: private

Giải thích: Thời gian đăng nhập cuối chỉ dùng nội bộ để theo dõi, không cho bên ngoài sửa trực tiếp. Có thể cung cấp getter nếu cần hiển thị.

**Bài 4:**

****

Lớp: Customer

Thuộc tính: customerId: int, name: string, email: string

Phương thức: viewOrder(): void, makeOrder(): void

Vai trò: Đại diện cho khách hàng, thực hiện việc xem và tạo đơn hàng.

Lớp: Order

Thuộc tính: orderId: int, orderDate: Date, totalAmount: double

Phương thức: calculateTotal(): double, addProduct(p: Product): void

Vai trò: Quản lý thông tin đơn hàng, tính tổng tiền và thêm sản phẩm vào đơn.

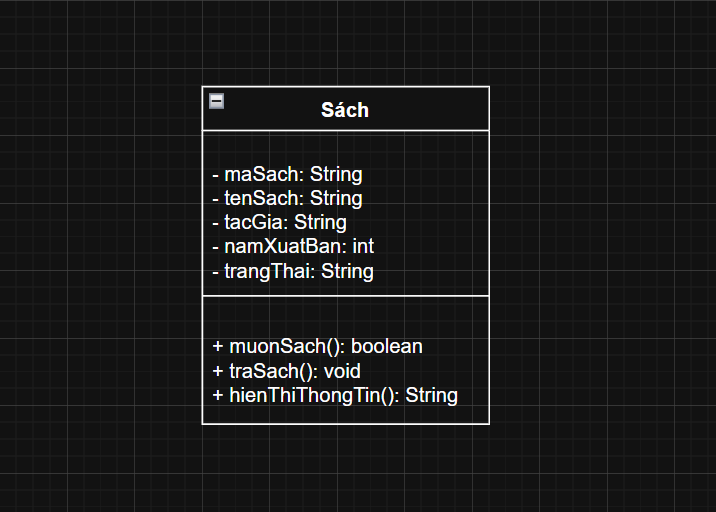
Lớp: Product

Thuộc tính: productId: int, productName: string, price: double

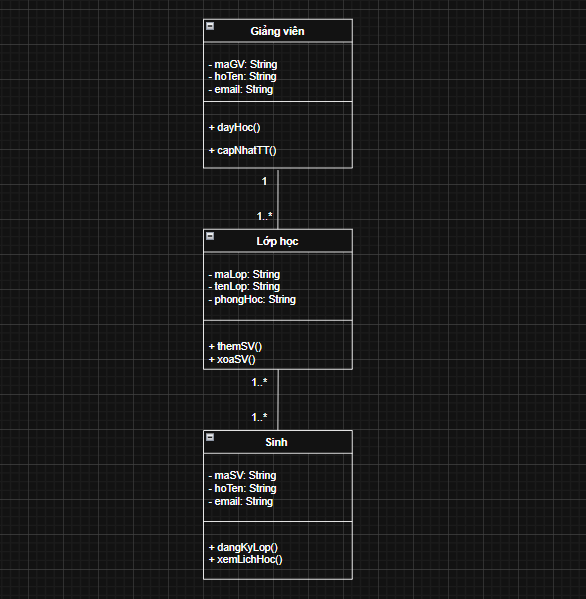
Phương thức: getInfo(): string, updatePrice(newPrice: double): void

Vai trò: Lưu trữ và quản lý thông tin sản phẩm, hỗ trợ truy xuất và cập nhật giá.

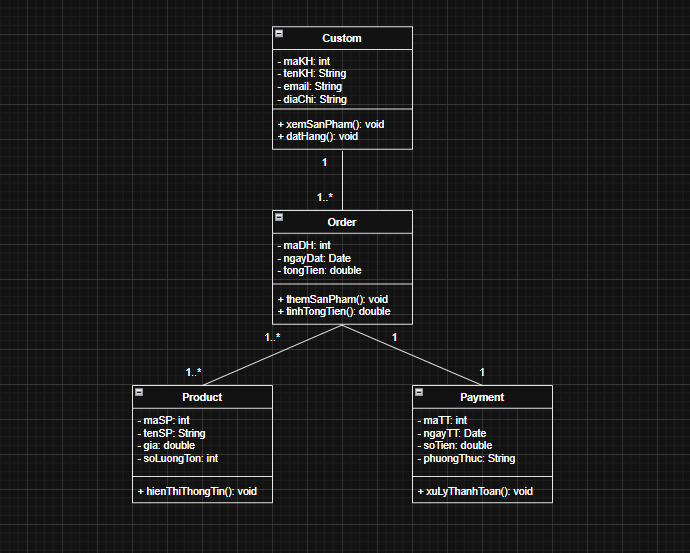
**Bài 5:**

****

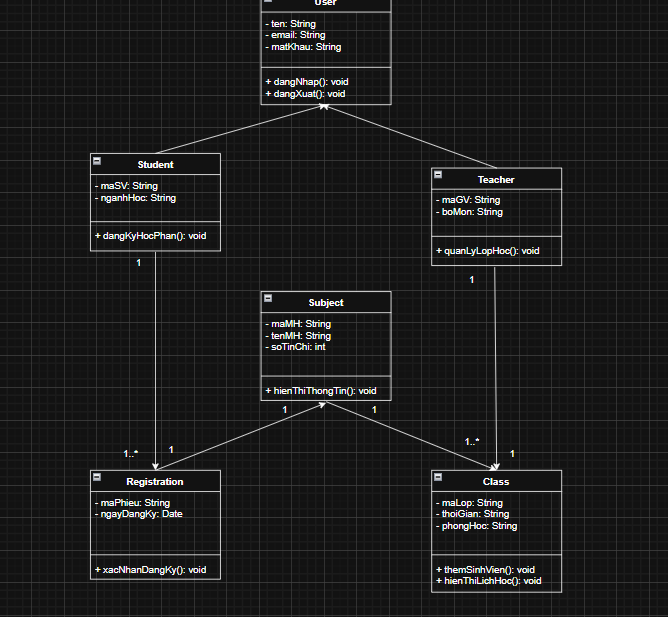
**Bài 6:**

****

**Bài 7:**

****

**Bài 8:**

****

**Bài 9:**

### Lỗi 1 – Quan hệ giữa các lớp (Relationship)

Một User (người dùng) có thể có nhiều Order (đơn hàng) → quan hệ 1–n, không phải 1–1.

Hiện sơ đồ để “1–1” là không hợp lý.

### Lỗi 2 – Modifier (dấu “+” ở thuộc tính)

Tất cả thuộc tính (username, password, email) không nên public (+) → nên là private (-).

Chỉ các phương thức (login(), resetPassword()) mới nên là public (+).

### Lỗi 3 – Multiplicity hoặc hướng mũi tên

Mũi tên giữa hai lớp đang thể hiện hướng phụ thuộc (dependency), nhưng ở đây phải là association (quan hệ sở hữu).  
 Cần dùng đường liền không mũi tên, chỉ kèm số lượng.

Bản cải tiến

