Bài 1 :

- Hệ thống quản lí thư viện : Sách , Độc giả , Nhân viên thư viện

- Chức năng chính :

+ Sách : Biểu diễn thông tin của tỗi cuốn sách trong thư viện

+ Độc giả : Người mượn , trả sách trong thư viện

+ Nhân viên thư viện : quản lí thư viện

- Vai trò và mô tả :

|  |  |
| --- | --- |
| Tên Lớp | Vai trò và mô tả |
| Sach | - thuộc tính : maSach , tenSach , tacGia...  - vai trò : Lưu chi tiết thông tin cuốn sách  - phuơng thức : hienThiThongTin() , capNhatCuonSach() |
| DocGia | - thuộc tính : maDocGia, tenDocGia, diaChi, soDienThoa  - vai trò :  Đại diện cho người mượn sách.  - phuơng thức :muonSach() , traSach() |
| NhanVien | - thuộc tính : maNV, tenNV, chucVu  - vai trò : Quản lý hoạt động mượn/trả, thêm hoặc xóa sách khỏi hệ thống  - phuơng thức :nhanTraSach() , capNhatThongTinSach() |

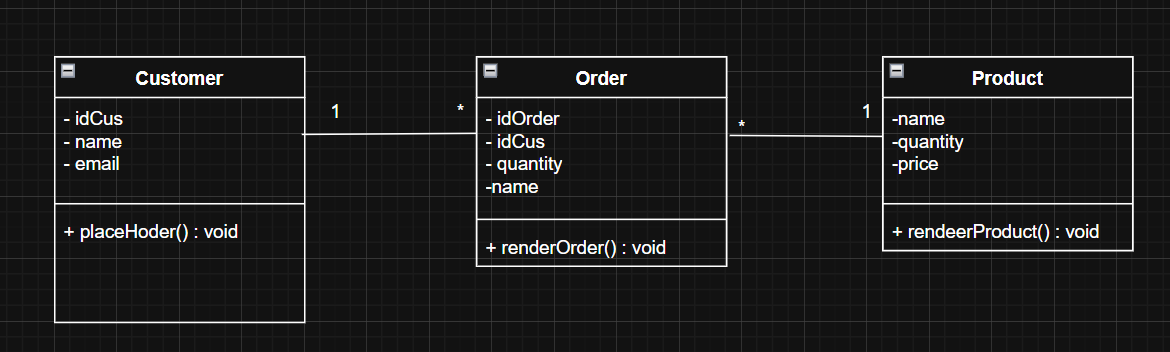
Bài 2

|  |  |  |  |
| --- | --- | --- | --- |
| Mối quan hệ | Loại quan hệ | Giải thích | Kí 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) |

Bài 3 :

|  |  |  |
| --- | --- | --- |
| **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 |

Bài 4 :



### Mô tả chức năng từng lớp

#### ****1. Lớp Product****

**- Thuộc tính:** productID , name , price

**- Chức năng:** Lưu thông tin sản phẩm , được tham chiếu trong Order để biết sản phẩm nào đang được đặt

#### ****2. Lớp Customer****

**- Thuộc tính:** customerID , name , email

**- Chức năng:** Quản lý thông tin khách hàng , Thực hiện hành động placeOrder() để tạo đơn hàng

#### ****3. Lớp Order****

**- Thuộc tính:** orderID , quantity , date , customer , product

**Chức năng:**

Lưu thông tin đơn hàng

Liên kết 1 sản phẩm cho 1 đơn hàng (đơn giản)

Gắn với đúng khách hàng đã đặt

Bài 5 :

**Lớp: Độc giả (DocGia)**

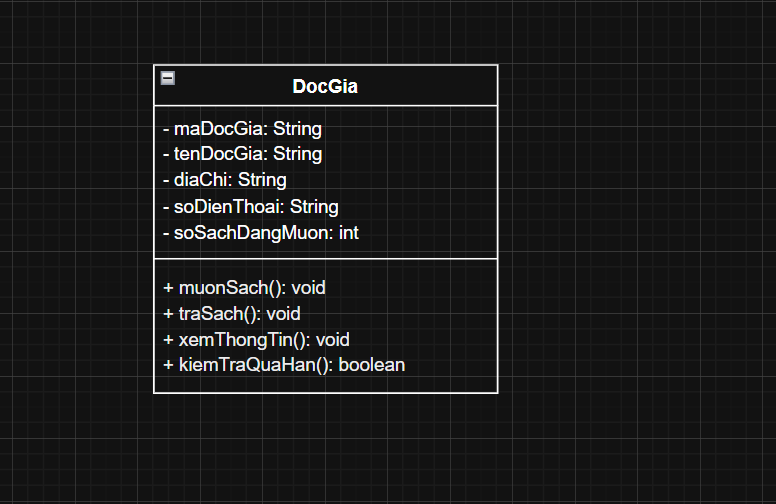
**1. Thuộc tính (Attributes)**

|  |  |  |
| --- | --- | --- |
| **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. |

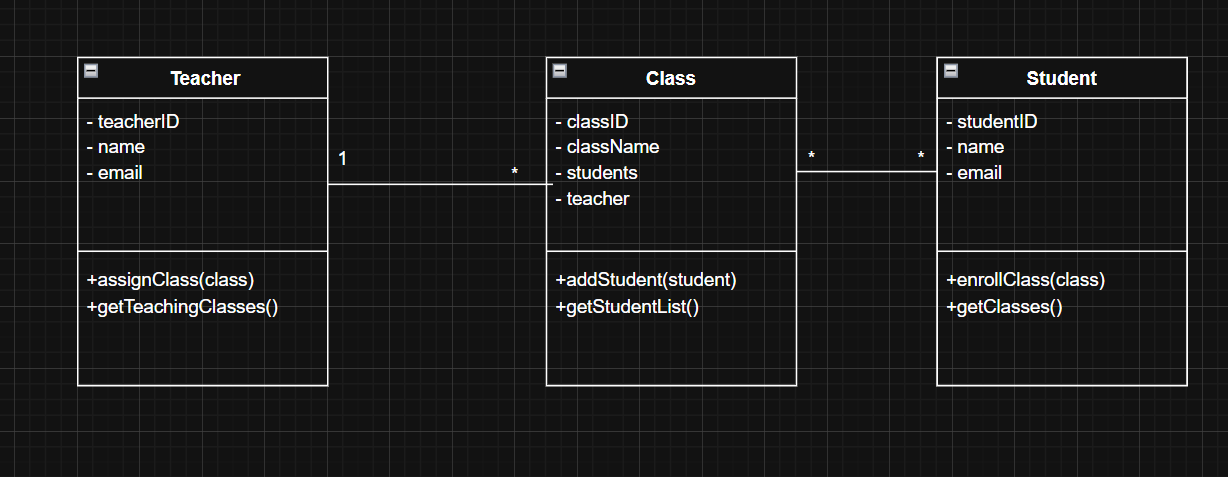
**2. Phương thức (Methods)**

|  |  |  |
| --- | --- | --- |
| **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. |

- UML

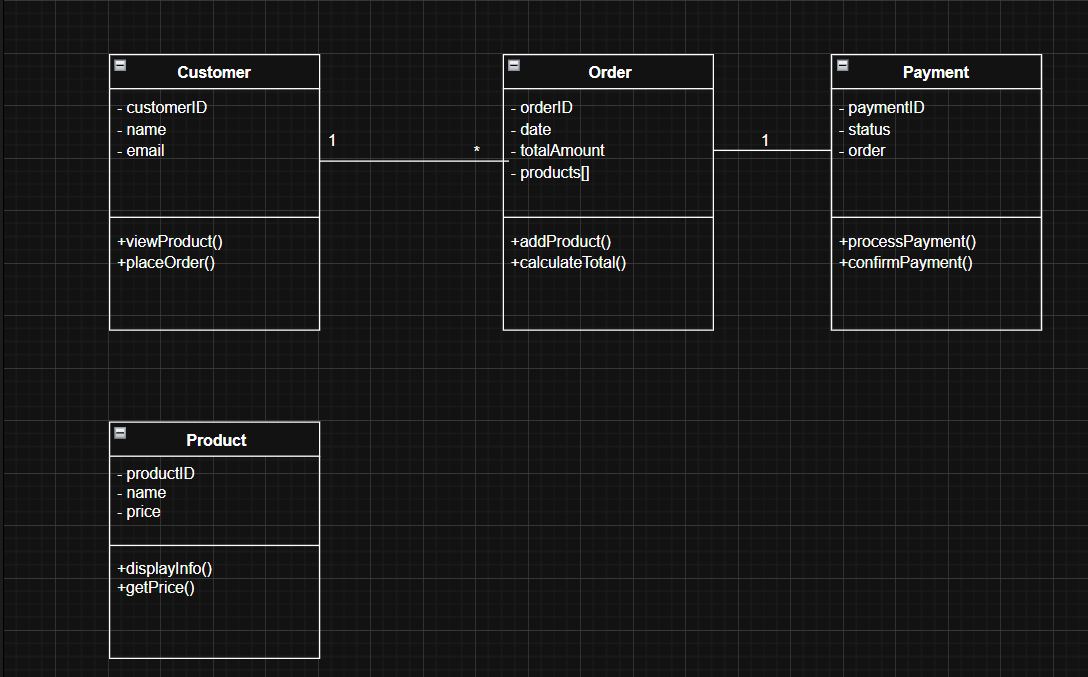


Bài 6 :

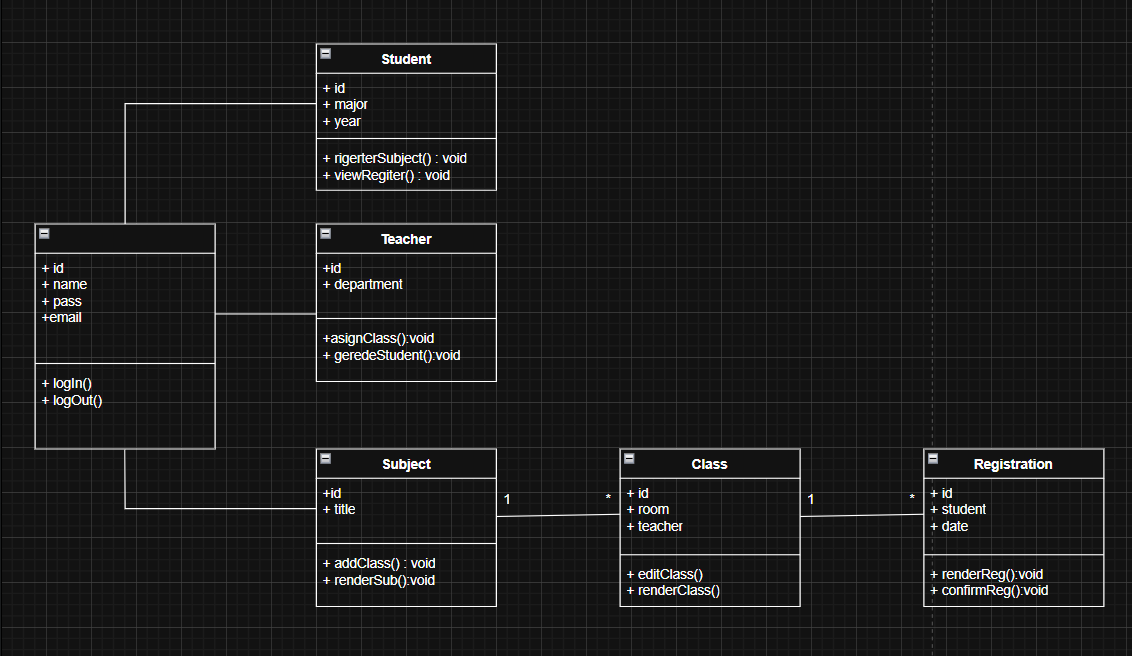


Bài 7:

Class : Xem sản phẩm : Product , Đặt hàng : Order và Customer , thanh toán : Payment



Bài 8 :



Bài 9 :

|  |  |  |
| --- | --- | --- |
| Loại lỗi | Mô tả lỗi chi tiết | Giải thích / Cách sửa |
| Lỗi quan hệ giữa các lớp | 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 | 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 | Tất cả các thuộc tính (username, password, email, OrderId, ...) đều để là + (public) | Trong UML chuẩn: thuộc tính nên là private (-), phương thức public (+). |

