Bài 1

**I. Danh sách lớp và vai trò**

**1. Class Sach (Book)  
Vai trò: Đại diện cho thông tin của một cuốn sách trong thư viện.  
Thuộc tính:**

* **maSach: string — Mã định danh duy nhất của sách**
* **tenSach: string — Tên sách**
* **tacGia: string — Tác giả**
* **namXuatBan: int — Năm xuất bản**
* **tinhTrang: boolean — Trạng thái (true = còn trong thư viện, false = đã được mượn)  
  Phương thức:**
* **hienThiThongTin() — Hiển thị thông tin chi tiết sách**
* **capNhatTinhTrang() — Cập nhật trạng thái khi mượn hoặc trả**

**2. Class DocGia (Reader)  
Vai trò: Quản lý thông tin của người đọc/mượn sách.  
Thuộc tính:**

* **maDocGia: string — Mã độc giả**
* **tenDocGia: string — Họ tên độc giả**
* **soDienThoai: string — Số điện thoại**
* **diaChi: string — Địa chỉ liên hệ  
  Phương thức:**
* **dangKy() — Đăng ký trở thành độc giả**
* **muonSach(Sach s) — Thực hiện mượn sách**
* **traSach(Sach s) — Thực hiện trả sách**

**3. Class NhanVien (Staff)  
Vai trò: Đại diện cho nhân viên thư viện – người quản lý hoạt động mượn/trả sách.  
Thuộc tính:**

* **maNhanVien: string — Mã nhân viên**
* **tenNhanVien: string — Họ tên nhân viên**
* **chucVu: string — Chức vụ  
  Phương thức:**
* **quanLyMuonTra() — Xác nhận và ghi nhận các giao dịch mượn/trả**
* **themSachMoi() — Thêm sách mới vào hệ thống**

**4. Class PhieuMuon (BorrowSlip / LoanRecord)  
Vai trò: Ghi lại thông tin mỗi lần mượn hoặc trả sách.  
Thuộc tính:**

* **maPhieu: string — Mã phiếu mượn**
* **docGia: DocGia — Thông tin người mượn**
* **sach: Sach — Thông tin sách được mượn**
* **ngayMuon: Date — Ngày mượn**
* **ngayTra: Date — Ngày trả (nếu có)**
* **tinhTrang: string — “Đang mượn” hoặc “Đã trả”  
  Phương thức:**
* **xacNhanMuon() — Xác nhận việc mượn**
* **xacNhanTra() — Cập nhật khi trả sách**

**5. Class ThuVien (Library)  
Vai trò: Lớp tổng quản lý hệ thống, lưu danh sách sách, độc giả và nhân viên.  
Thuộc tính:**

* **danhSachSach: List<Sach>**
* **danhSachDocGia: List<DocGia>**
* **danhSachNhanVien: List<NhanVien>**
* **danhSachPhieuMuon: List<PhieuMuon>  
  Phương thức:**
* **timSachTheoTen(ten) — Tìm kiếm sách theo tên**
* **themDocGia(docGia) — Thêm độc giả mới**
* **xoaSach(maSach) — Xóa sách khỏi hệ thống**

**II. Tổng kết mối quan hệ giữa các lớp**

* **DocGia ↔ PhieuMuon: Một độc giả có thể có nhiều phiếu mượn.**
* **Sach ↔ PhieuMuon: Một sách có thể xuất hiện trong nhiều phiếu (theo thời gian).**
* **NhanVien ↔ PhieuMuon: Nhân viên quản lý và xác nhận các phiếu mượn/trả.**
* **ThuVien: Là lớp tổng hợp (aggregate) quản lý toàn bộ dữ liệu: sách, độc giả, nhân viên và phiếu mượn.**

**Bài 2**

**Phân biệt các quan hệ trong UML**

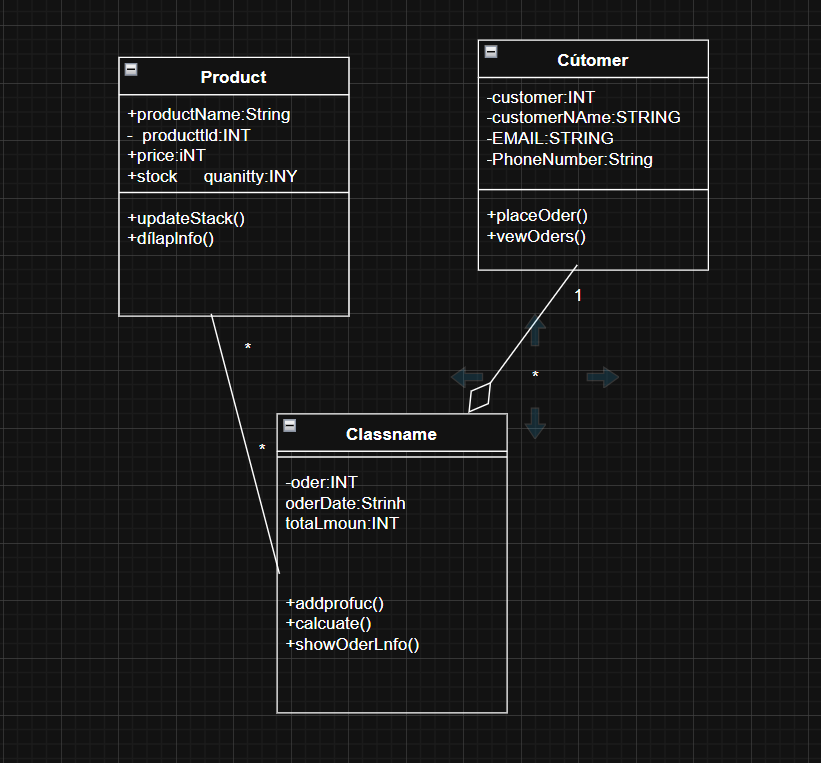
|  |  |  |  |
| --- | --- | --- | --- |
| **Mối quan hệ** | **Loại** | **Giải thích** | **Ký hiệu UML** |
| **Giáo viên – Lớp học** | **Association** | **Hai lớp liên kết độc lập, không phụ thuộc nhau.** | **Đường nối có nhãn 1..\*** |
| **Đơn hàng – Sản phẩm** | **Aggregation** | **Có chứa nhưng tách rời được, xóa đơn hàng sản phẩm vẫn còn.** | **Hình thoi rỗng (◊)** |
| **Cơ thể – Bộ phận** | **Composition** | **Chứa chặt chẽ, xóa cơ thể thì bộ phận cũng mất.** | **Hình thoi đặc (◆)** |

Bài 3

**Phạm vi truy cập cho class User**

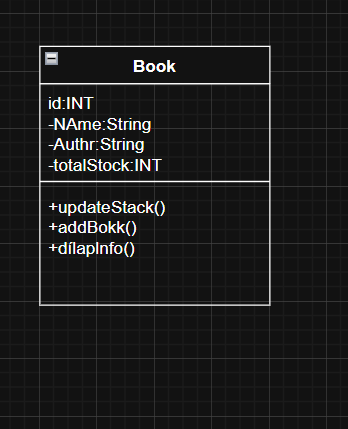
|  |  |  |
| --- | --- | --- |
| **Thành phần** | **Modifier (Phạm vi truy cập)** | **Giải thích** |
| username | private | Chỉ được truy cập trong lớp User, bảo vệ thông tin người dùng. |
| password | private | Bảo mật tuyệt đối, không cho truy cập trực tiếp từ bên ngoài. |
| lastLoginTime | private *(hoặc protected nếu dùng kế thừa)* | Dữ liệu nội bộ, chỉ lớp User (hoặc lớp con) mới được truy cập. |
| login() | public | Cho phép bên ngoài (ứng dụng) gọi để đăng nhập. |
| resetPassword() | public | Cho phép bên ngoài thực hiện thao tác đổi mật khẩu hợp lệ. |

Bài 4

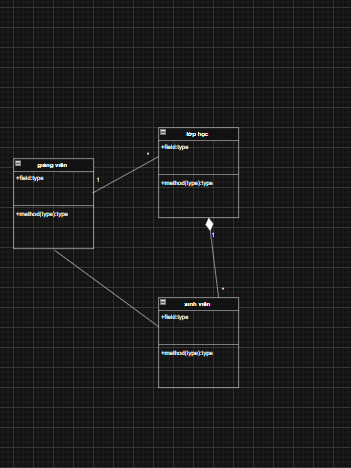


|  |  |  |
| --- | --- | --- |
| Thực thể | Quan hệ trực tiếp với | Giải thích |
| Customer | Order | Mỗi khách hàng có thể tạo ra nhiều đơn hàng. Đây là quan hệ 1–n (một-nhiều). |
| Order | Product | Mỗi đơn hàng có thể chứa nhiều sản phẩm, và một sản phẩm có thể nằm trong nhiều đơn hàng -> quan hệ n–n (nhiều-nhiều). |
| Customer | *(Không trực tiếp)* | Khách hàng không trực tiếp “sở hữu” hay “chứa” sản phẩm. Việc mua sản phẩm chỉ diễn ra thông qua đơn hàng. |

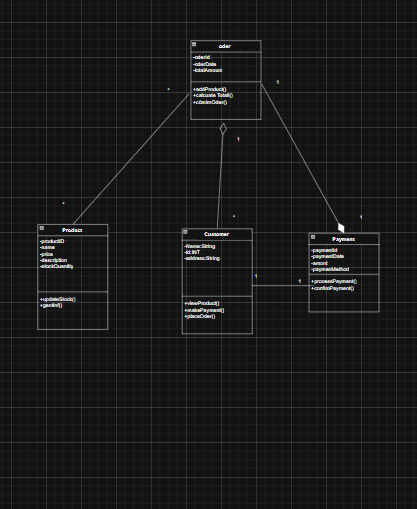
Bài 5



Bài 6



Bài 7



Bài 8

