**Bài 1:**

|  |  |  |
| --- | --- | --- |
| **Class** | **Vai trò** | **Mô tả** |
| Sach | Lưu trữ thông tin sách | Đại diện cho 1 cuốn sách trong thư viện. Lớp này chứa các thuộc tính như mã sách, tên sách |
| DocGia | Đại diện cho người dùng | Lưu trữ thông tin cho người dùng. Lớp này chứa các thuộc tính như mã độc giả, tên độc giả |
| NhanVien | Quản lý và vận hành hệ thống thư viện | Là người có quyền truy cập các chức năng quản lý trong thư viện. Lớp này bao gồm các thuộc tính như mã nhân viên, tên nhân viên, chức vụ |

**Bài 2:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mối quan hệ** | **Loại** | **Giải thích** | **Ký hiệu UML** |
| Một giáo viên giảng dạy nhiều lớp học | Association (Liên kết) | Đây là mối quan hệ hai chiều, thể hiện rằng một giáo viên có thể dạy nhiều lớp, và một lớp có thể được dạy bởi một hoặc nhiều giáo viên. Không có ràng buộc về sở hữu hay phụ thuộc sinh-tử giữa các đối tượng. | Đường nối giữa hai lớp, có thể ghi bội số như 1..\* hoặc \* |
| Một đơn hàng gồm nhiều sản phẩm | Aggregation (Kết tập) | Là mối quan hệ “có – gồm có” nhưng các thành phần vẫn tồn tại độc lập. Ví dụ: sản phẩm vẫn tồn tại dù đơn hàng bị xóa. | Đường nối với hình thoi rỗng ở phía đơn hàng |
| Một cơ thể gồm nhiều bộ phận không thể tách rời | Composition (Thành phần) | Là mối quan hệ “chứa – là một phần của” có ràng buộc sinh-tử: khi cơ thể bị hủy thì các bộ phận cũng bị hủy theo. | Đường nối với hình thoi đặc ở phía cơ thể |

**Bài 3:**

|  |  |  |
| --- | --- | --- |
| **Thành phần** | **Modifier (Phạm vi truy cập)** | **Giải thích** |
| username | private | Thông tin nhạy cảm, chỉ nên được truy cập nội bộ trong lớp để bảo mật. Có thể cung cấp getter/setter công khai nếu cần truy xuất. |
| password | private | Là dữ liệu bảo mật cao, không được phép truy cập trực tiếp từ bên ngoài. Chỉ nội bộ lớp User có thể kiểm tra hoặc thay đổi. |
| login() | public | Là hành vi chính của người dùng, cần được gọi từ bên ngoài (ví dụ: từ giao diện đăng nhập), nên công khai. |
| resetPassword() | public | Cũng là hành vi người dùng có thể thực hiện (hoặc quản trị viên gọi), nên cần được truy cập công khai. |
| lastLoginTime | protected | Có thể được sử dụng trong các lớp kế thừa (ví dụ: AdminUser, GuestUser) để theo dõi hoạt động đăng nhập, nhưng không cần công khai với bên ngoài. |

**Bài 5:**

Lớp: Sách

1. Thuộc tính

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| id | string | Mã định danh duy nhất của sách |
| title | string | Tên sách |
| author | string | Tên tác giả |
| publisher | string | Nhà xuất bản |
| quantity | int | Số lượng bản sao trong kho/ thư viện |

1. Phương thức

|  |  |  |
| --- | --- | --- |
| Tên phương thức | Kiểu trả về | Mô tả |
| displayInfo() | void | Hiển thị thông tin chi tiết của sách |
| updateQuantity(int newQty) | void | Cập nhật số lượng bản sao của sách |
| isAvailable() | boolean | Kiểm tra xem sách còn bản sao để mượn hay không |

1. UML dạng text

|  |
| --- |
| Book |
| - bookID: String  - title: String  - author: String  - publisher: String  - quantity: int |
| + displayInfo(): void  + updateQuantity(newQty:int): void  + isAvailable(): boolean |

**Bài 9:**

|  |  |
| --- | --- |
| **Loại lỗi** | **Mô tả chi tiết** |
| 1. Lỗi quan hệ (Relationship) | Trong thực tế, một User có thể có nhiều Order, không phải chỉ 1. Nhưng sơ đồ lại thể hiện là 1–1, tức một User chỉ có một đơn hàng — sai với logic hệ thống bán hàng. |
| 2. Lỗi multiplicity (bội số) | Quan hệ đúng phải là 1--\* (một User có thể tạo nhiều Order). Còn một Order chỉ thuộc về một User → multiplicity phải là 1 ở phía Order → User. |
| 3. Lỗi đặt modifier hoặc thiếu thông tin truy cập | Các thuộc tính nên để ở phạm vi private (-) thay vì public (+) để đảm bảo tính đóng gói (encapsulation). Hiện tại sơ đồ để tất cả + là không đúng nguyên tắc hướng đối tượng. |

**Sơ đồ cải tiến:**

