Bài 1 :

|  |  |
| --- | --- |
| Lớp | Vai trò/ mô tả |
| Book | Đại diện cho một cuốn sách trong hệ thống thư viện. Lớp này lưu các thông tin như mã sách, tên sách, tác giả, năm xuất bản, và có thể có trạng thái (còn hay đã mượn) hoặc số lượng bản sao nếu cần mở rộng. |
| Member | Đại diện cho người đọc (độc giả) thư viện. Chứa thông tin như mã bạn đọc, tên, thông tin liên hệ, danh sách sách đang mượn… và các phương thức như mượn sách, trả sách |
| Librarian | Đại diện cho nhân viên thư viện, người chịu trách nhiệm quản lý sách, thông tin độc giả, xử lý nghiệp vụ mượn/trả sách. Lớp này lưu thông tin như mã nhân viên, tên, và có các phương thức như đăng ký bạn đọc, thêm sách vào thư viện, xử lý mượn/trả. |
| Borrowing | Đại diện cho hành động mượn (và trả) sách. Lớp này lưu thông tin như mã phiếu mượn, ngày mượn, ngày trả dự kiến hoặc ngày trả thực tế, trạng thái phiếu (đang mượn/đã trả), và liên kết tới một độc giả (Member) và một cuốn sách (Book). |

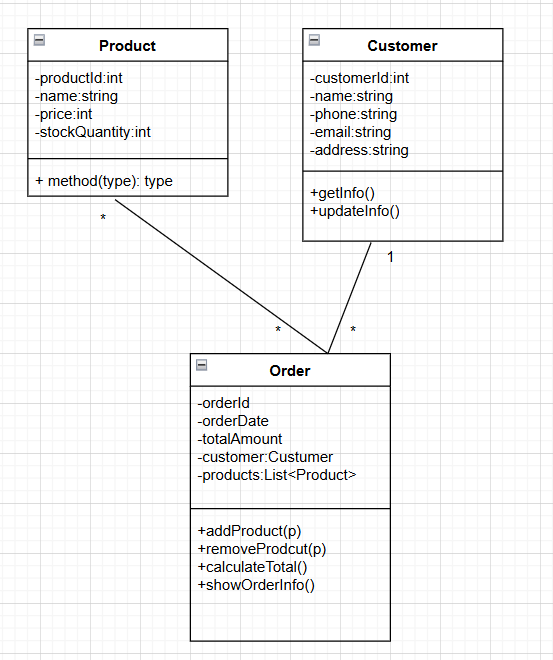
Bài 2:

|  |  |  |  |
| --- | --- | --- | --- |
| Mối quan hệ | Loại quan hệ | Giải thích | Ký hiệu UML |
| Một giáo viên giảng dạy nhiều lớp học | Association | Mối quan hệ bình thường giữa hai lớp “Giáo viên” và “Lớp học”. Mỗi giáo viên có thể dạy nhiều lớp, và mỗi lớp có thể có một hoặc nhiều giáo viên. Hai đối tượng tồn tại độc lập nhau — nếu xóa giáo viên, lớp học vẫn tồn tại. | Đường nối đơn giản giữa hai lớp, có thể ghi bội số |
| Một đơn hàng gồm nhiều sản phẩm | Aggregation | Mối quan hệ “toàn thể – bộ phận” nhưng các phần có thể tách rời. Một đơn hàng chứa nhiều sản phẩm, nhưng sản phẩm vẫn tồn tại độc lập ngoài đơn hàng (có thể thuộc đơn hàng khác). | Đường nối có hình thoi rỗng (◊) |
| Một cơ thể gồm nhiều bộ phận không thể tách rời | composotion | Mối quan hệ “toàn thể – bộ phận” nhưng gắn bó chặt chẽ, nghĩa là khi đối tượng “Cơ thể” bị hủy thì các “Bộ phận” cũng mất theo. Các phần không thể tồn tại độc lập. | Đường nối có hình thoi đặc |

Bài 3 :

|  |  |  |
| --- | --- | --- |
| Thành phần | Modifier | Giải thích |
| Username() | Private | Thuộc tính chỉ được truy cập nội bộ trong lớp User. Có thể truy cập thông qua getter/setter. |
| Password() | Private | Thông tin bảo mật, chỉ lớp User mới được phép truy cập và xử lý. Không cho phép truy cập trực tiếp từ bên ngoài để tránh lộ mật khẩu |
| lastLoginTime() | Protected | Chỉ lớp User và các lớp kế thừa mới có thể truy cập hoặc ghi lại thời gian đăng nhập. Không cho phép bên ngoài thay đổi trực tiếp |
| Login() | Public | Phương thức cho phép người dùng thực hiện đăng nhập từ bên ngoài hệ thống, nên cần công khai để các thành phần khác |
| ResetPassword() | public | Cho phép người dùng hoặc hệ thống gọi từ bên ngoài để đặt lại mật khẩu khi cần. Phải ở mức public để truy cập được từ giao diện hoặc module quản lý tài khoản |

Bài 4:

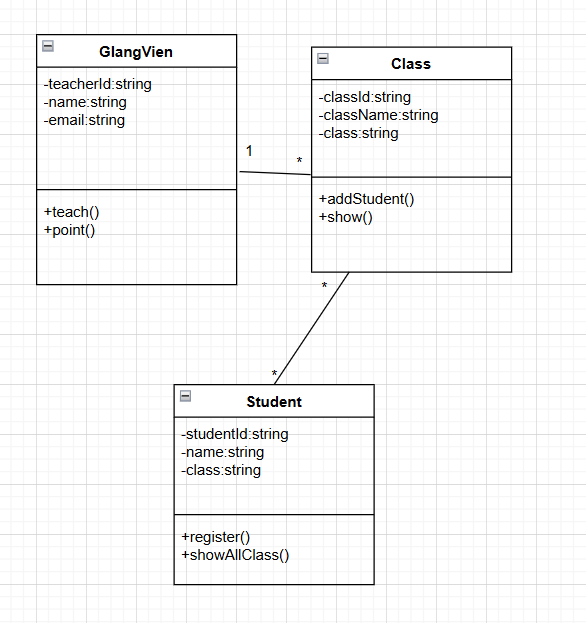


Bài 5:

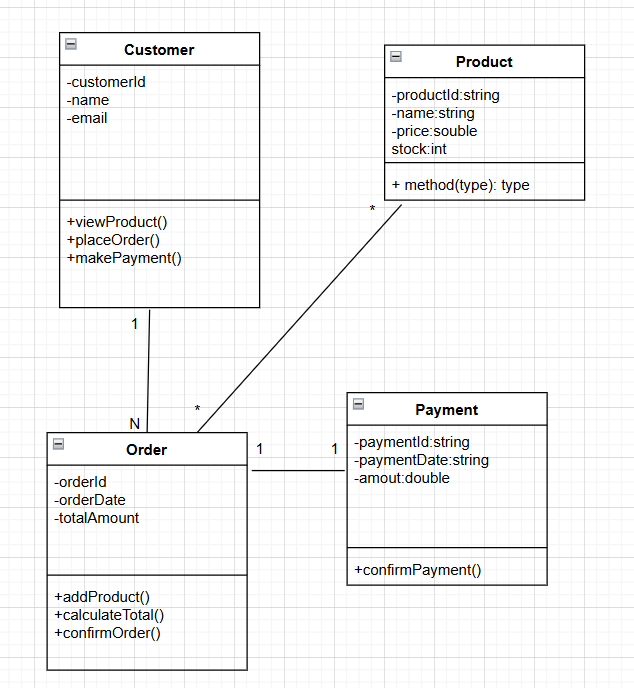
|  |  |  |  |
| --- | --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả | Phạm vi truy cập |
| bookId | String | Mã định danh duy nhất của sách | Private |
| Title | String | Tên sách | Private |
| Author | String | Tác giả của sách | Private |
| Price | Double | Giá bán hoặc giá mượn | Private |
| Quantity | Int | Số lượng sách cong trong kho / thư viện | Private |

|  |  |  |  |
| --- | --- | --- | --- |
| Tên pthuc | Kiểu trả về | Mô tả | Pvi truy cập |
| getInfo() | Void | Hiển thị thông tin chi tiết của sách | Public |
| updateQuantity(int newQuantity) | Void | Cập nhật số lượng tồn kho của sách | Public |
| applyDiscount(double percent) | Void | Giảm giá sách theo % khuyến mãi | Public |

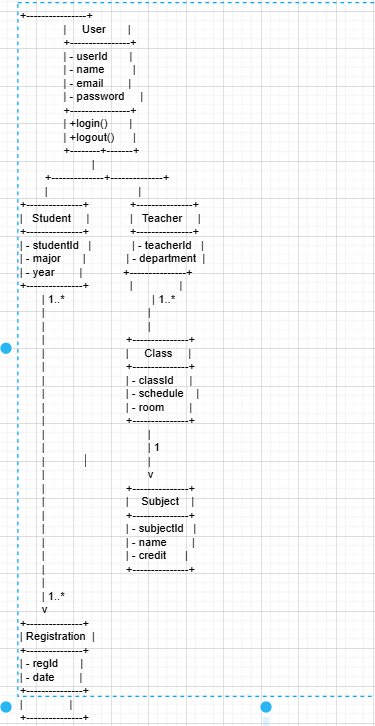
Bài 6:



Bài 7:



Bài 8:



Bài 9:

Lỗi 1 – Thiếu quan hệ giữa các lớp Hai lớp User và Order không có đường kết nối (association) nào.

Trong hệ thống bán hàng, một User (khách hàng) có thể tạo nhiều đơn hàng, do đó cần có mối quan hệ 1–N (User – Order).

Lỗi 2 – Thiếu kiểu dữ liệu cho thuộc tính Các thuộc tính (username, password, email, OrderId, OrderDate, totalAmount) chưa có kiểu dữ liệu.

UML chuẩn yêu cầu ghi rõ: tên: kiểu\_dữ\_liệu.

Lỗi 3 – Sai phạm vi truy cập (modifier) Trong UML, thông thường:

Thuộc tính là private (-) để bảo mật.

Phương thức là public (+) để truy cập được từ ngoài.

