

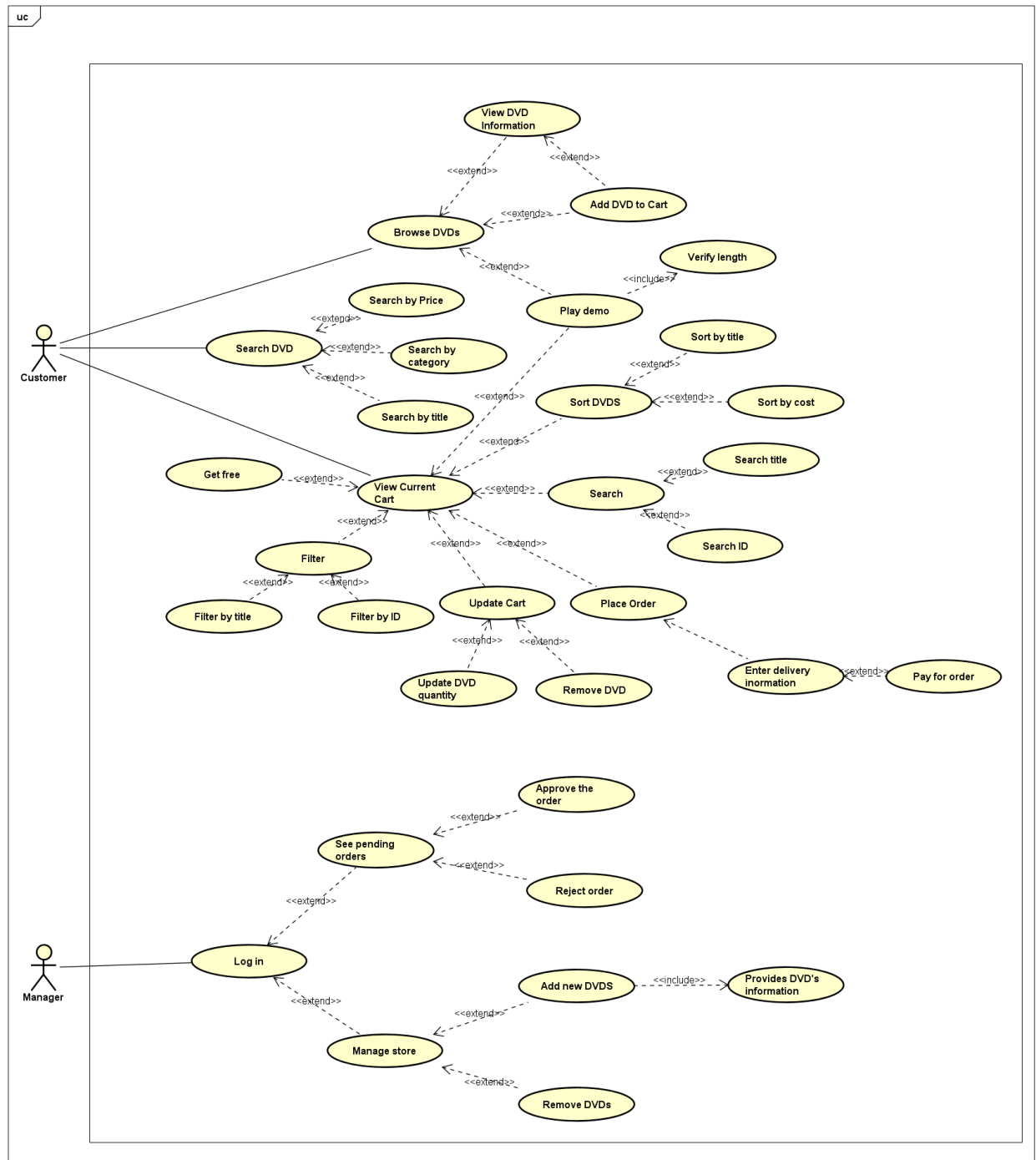
**Báo cáo thực hành Lab02 OOP**

Họ và tên: Lê Nguyễn Minh Đức

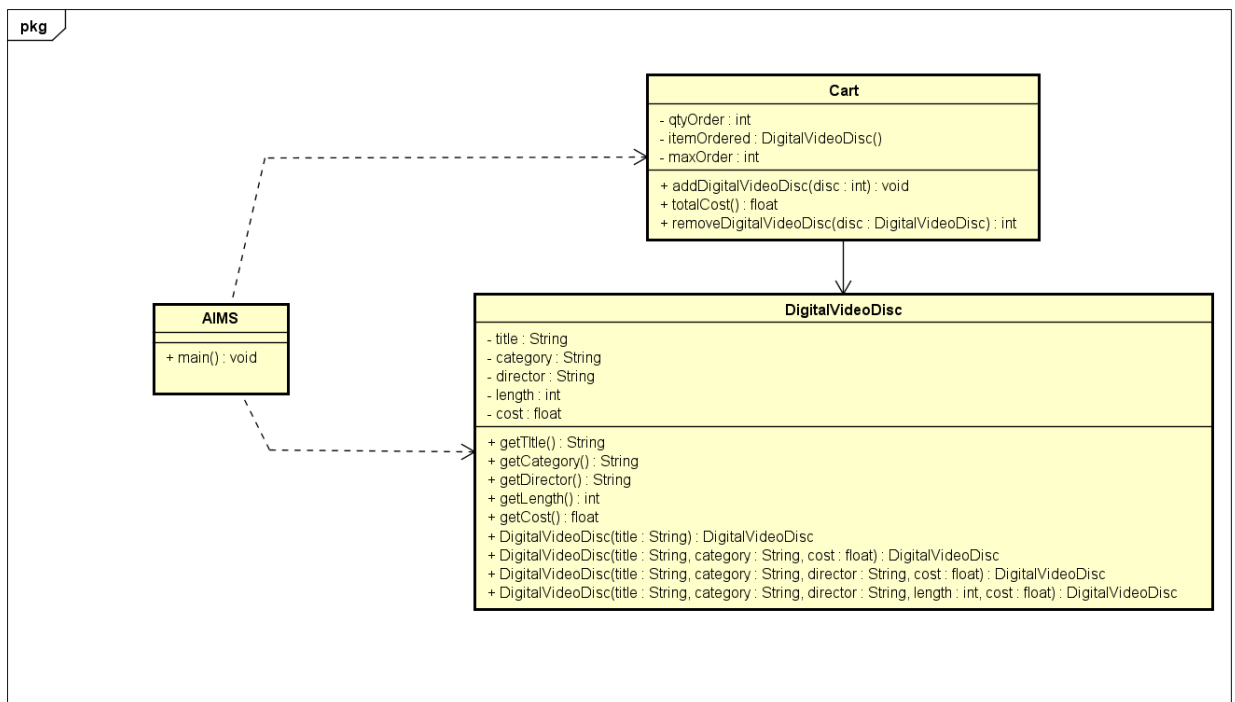
MSSV: 20225611

Mã lớp: 744523

## • Use Case Diagram:



- **Class Diagram:**



- **Reading assignment: Khi nào nên sử dụng Accessor?**
  - **Trả về dưới dạng Interface:** Khi cần trả về một đối tượng dưới dạng Interface mà nó triển khai, giúp mã bên ngoài không bị ảnh hưởng bởi các thay đổi bên trong. Điều này không phải là "getter" thông thường mà nhằm bảo vệ mã khỏi phụ thuộc vào chi tiết triển khai.
  - **Tương tác với hệ thống thủ tục:** Khi làm việc với hệ thống thủ tục (như hệ điều hành hoặc cơ sở dữ liệu), accessor giúp giữ tính linh hoạt cần thiết để tương tác với các hệ thống này, vì chúng thường có giao diện chung và linh hoạt.
  - **Tăng tính linh hoạt cho các thư viện Java:** Trong các thư viện Java, accessor thường được sử dụng để đảm bảo

tính linh hoạt khi cách sử dụng chưa rõ ràng. Tuy nhiên, nếu bạn nắm rõ cách các lớp sẽ được dùng, có thể giảm thiểu việc sử dụng accessor để giữ thiết kế đơn giản và dễ bảo trì.

- **Question:** Nếu bạn tạo một phương thức khởi tạo để xây dựng một DVD theo tiêu đề, thì bạn có thể tạo một phương thức khởi tạo để xây dựng một DVD theo thể loại. JAVA có cho phép bạn làm điều này không?
  - Có, Java cho phép bạn tạo nhiều phương thức khởi tạo (constructor) trong cùng một lớp, miễn là chúng có chữ ký khác nhau (có thể khác nhau về số lượng tham số hoặc kiểu dữ liệu của tham số). Điều này được gọi là nạp chồng phương thức khởi tạo (constructor overloading).