**THỰC HÀNH LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG LAB 02**

**GVHD: Đàm Quang Tuấn**

**Mã lớp: 744520**

Phạm Mạnh Quyết

20225663

Contents

[**I.** **Use case diagram** 2](#_Toc180876747)

[**II.** **Class Diagram** 3](#_Toc180876748)

[**III.** **Reading assignment** 3](#_Toc180876749)

[**IV.** **Source code** 4](#_Toc180876750)

[***1.*** ***Class: DigitalVideoDisc*** 4](#_Toc180876751)

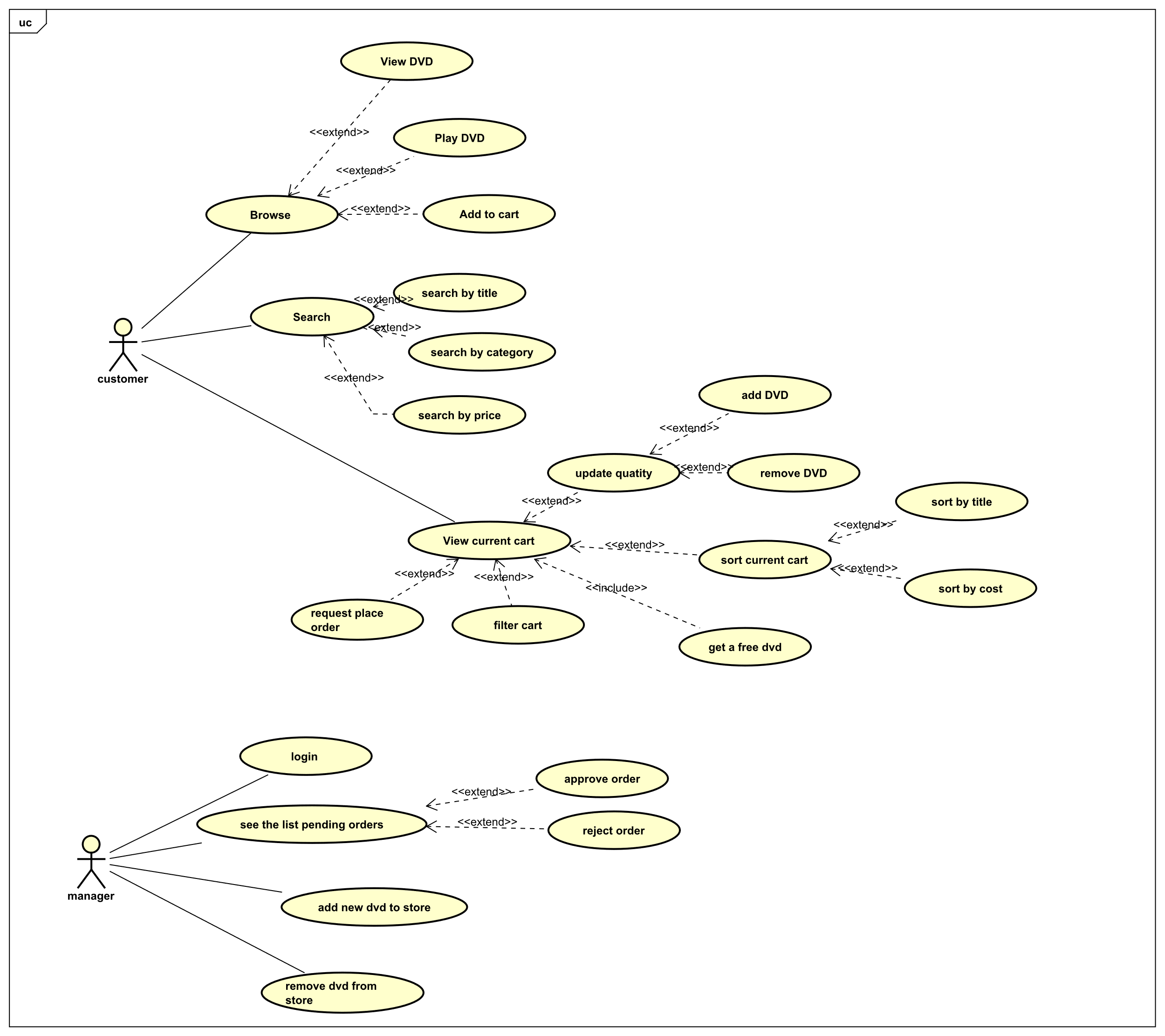
[***2.*** ***Class: Cart*** 6](#_Toc180876752)

[***3.*** ***Class: Aims*** 8](#_Toc180876753)

[**V.** **Answer** 9](#_Toc180876754)

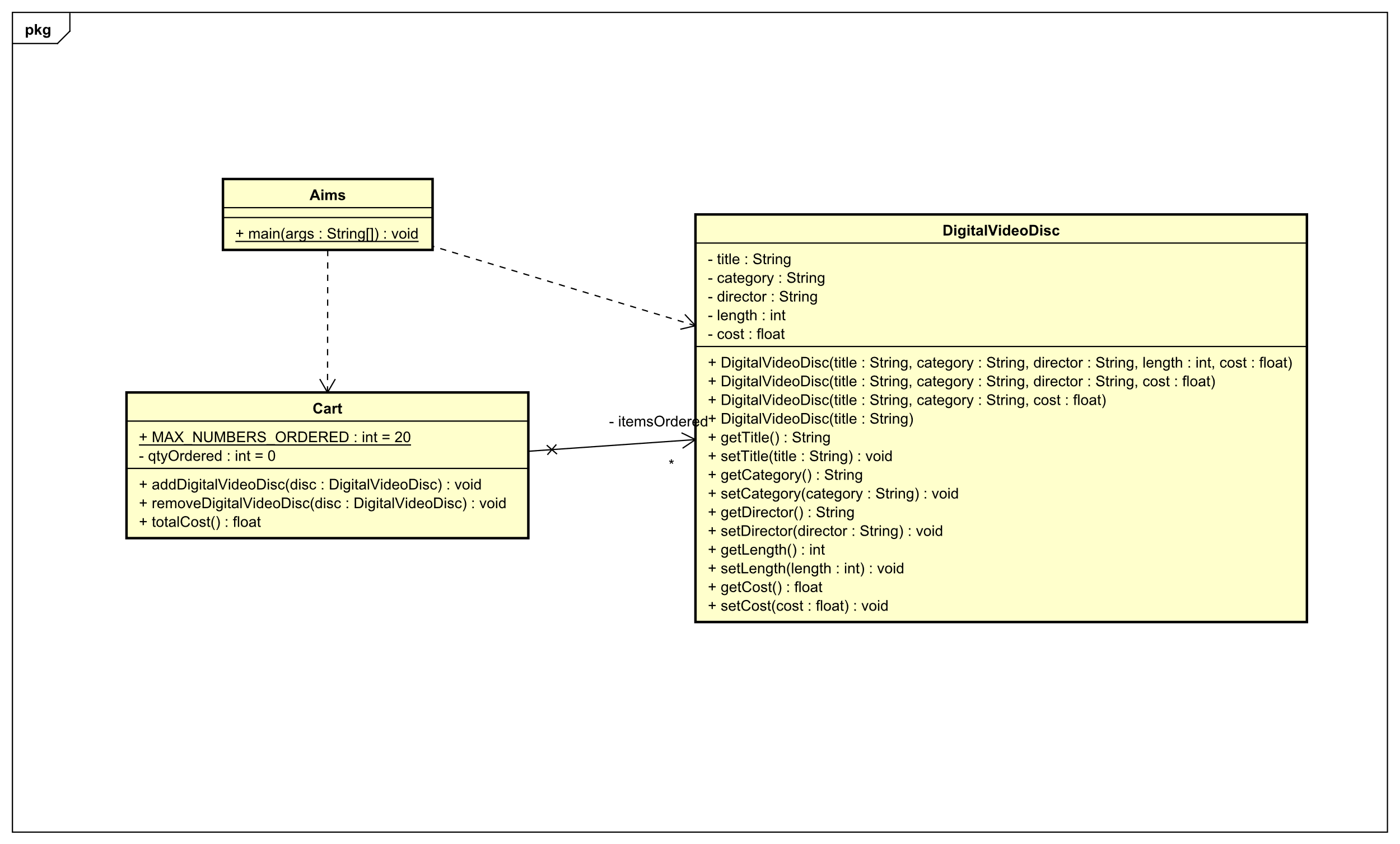
1. **Use case diagram**

* Source file: 
* Hình ảnh:



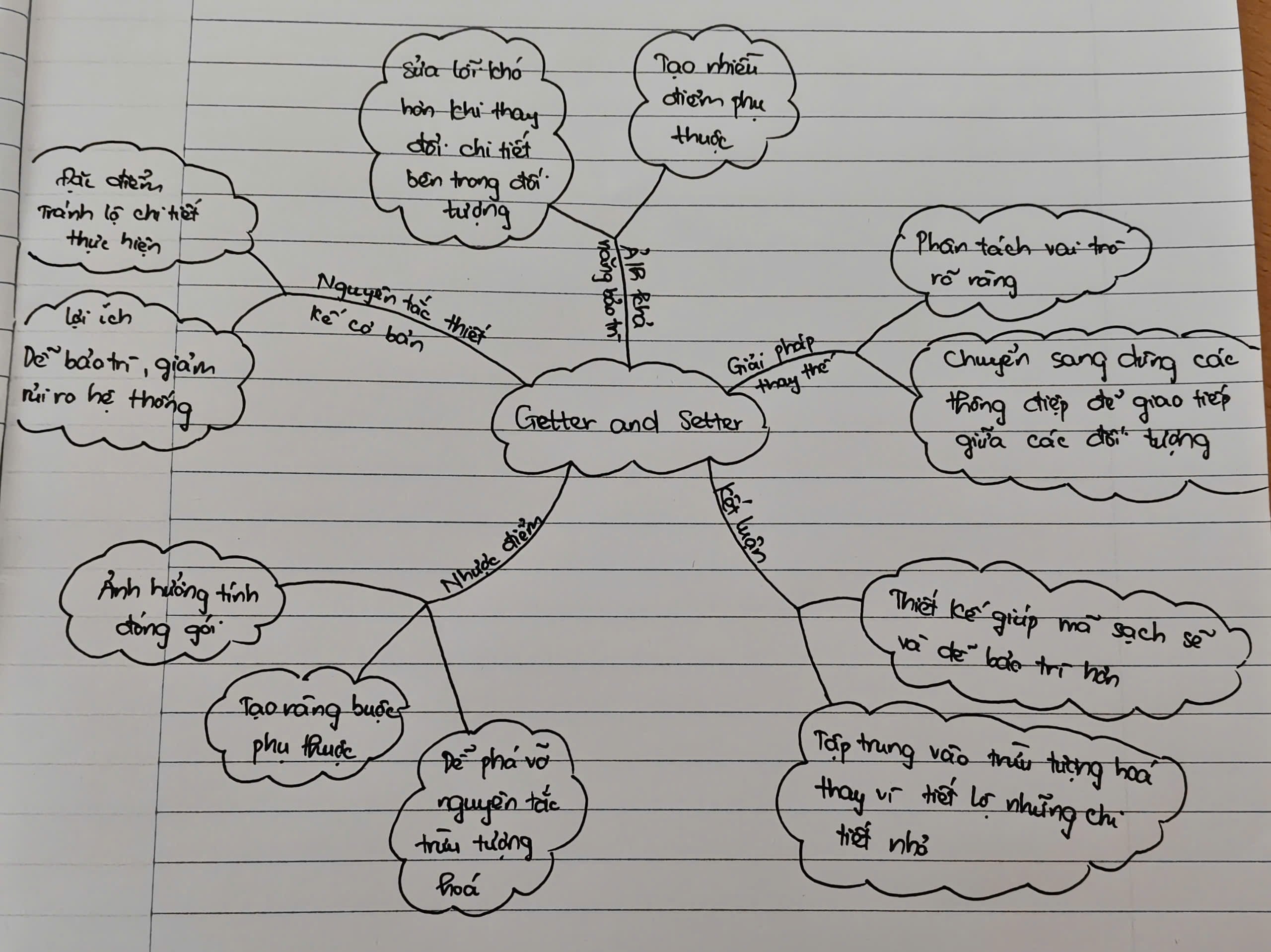
1. **Class Diagram**

* Source file: 
* Hình ảnh:



1. **Reading assignment**

* Câu hỏi: When should accessor methods be used?
  + Trả lời: Khi cho phép các lớp bên ngoài truy cập giá trị của thuộc tính mà vẫn đảm bảo tính đóng gói. Tức là hạn chế truy cập trực tiếp thuộc tính từ bên ngoài và chỉ cung cấp quyền đọc không cung cấp quyền ghi.
* Mind map:



1. **Source code**
2. ***Class: DigitalVideoDisc***

**package** src;

**public** **class** DigitalVideoDisc {

**private** String title;

**private** String category;

**private** String director;

**private** **int** length;

**private** **float** cost;

**public** DigitalVideoDisc(String title, String category, String director, **int** length, **float** cost) {

**super**();

**this**.title = title;

**this**.category = category;

**this**.director = director;

**this**.length = length;

**this**.cost = cost;

}

**public** DigitalVideoDisc(String title, String category, String director, **float** cost) {

**super**();

**this**.title = title;

**this**.category = category;

**this**.director = director;

**this**.cost = cost;

}

**public** DigitalVideoDisc(String title, String category, **float** cost) {

**super**();

**this**.title = title;

**this**.category = category;

**this**.cost = cost;

}

**public** DigitalVideoDisc(String title) {

**super**();

**this**.title = title;

}

**public** String getTitle() {

**return** title;

}

**public** **void** setTitle(String title) {

**this**.title = title;

}

**public** String getCategory() {

**return** category;

}

**public** **void** setCategory(String category) {

**this**.category = category;

}

**public** String getDirector() {

**return** director;

}

**public** **void** setDirector(String director) {

**this**.director = director;

}

**public** **int** getLength() {

**return** length;

}

**public** **void** setLength(**int** length) {

**this**.length = length;

}

**public** **float** getCost() {

**return** cost;

}

**public** **void** setCost(**float** cost) {

**this**.cost = cost;

}

}

1. ***Class: Cart***

**package** src;

**public** **class** Cart {

**public** **static** **final** **int** ***MAX\_NUMBERS\_ORDERED*** = 20;

**private** DigitalVideoDisc itemsOrdered[] = **new** DigitalVideoDisc[***MAX\_NUMBERS\_ORDERED***];

**private** **int** qtyOrdered = 0;

**public** **void** addDigitalVideoDisc(DigitalVideoDisc disc) {

**if**(qtyOrdered >= ***MAX\_NUMBERS\_ORDERED***) {

System.***out***.println("Gio hang da day!");

**return**;

}

itemsOrdered[qtyOrdered] = disc;

qtyOrdered += 1;

System.***out***.println("Dia da duoc them!");

}

**public** **void** removeDigitalVideoDisc(DigitalVideoDisc disc) {

**int** check = 0;

**for**(**int** i = 0; i < qtyOrdered; i++) {

**if**(itemsOrdered[i].equals(disc)) {

check = 1;

**for**(**int** j = i; j < qtyOrdered - 1; j++) {

itemsOrdered[j] = itemsOrdered[j + 1];

}

itemsOrdered[qtyOrdered - 1] = **null**;

qtyOrdered--;

System.***out***.println("Dia da duoc xoa!");

**break**;

}

}

**if**(check == 0) System.***out***.println("Khong tim thay dia de xoa!");

}

**public** **float** totalCost() {

**float** sum = 0;

**for**(**int** i = 0; i < qtyOrdered; i++) {

sum += itemsOrdered[i].getCost();

}

**return** sum;

}

}

1. ***Class: Aims***

**package** src;

**public** **class** Aims {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Cart anOrder = **new** Cart();

DigitalVideoDisc dvd1 = **new** DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);

anOrder.addDigitalVideoDisc(dvd1);

DigitalVideoDisc dvd2 = **new** DigitalVideoDisc("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);

anOrder.addDigitalVideoDisc(dvd2);

DigitalVideoDisc dvd3 = **new** DigitalVideoDisc("Aladin", "Animation", 18.99f);

anOrder.addDigitalVideoDisc(dvd3);

// Tạo đĩa dvd4 để test removeDigital

DigitalVideoDisc dvd4 = **new** DigitalVideoDisc("Vivo", "Animation", 25.99f);

System.***out***.print("Total Cose is: ");

System.***out***.println(anOrder.totalCost());

anOrder.removeDigitalVideoDisc(dvd4);

anOrder.removeDigitalVideoDisc(dvd3);

System.***out***.print("Total Cose after removed is: ");

System.***out***.println(anOrder.totalCost());

}

}

1. **Answer**

* Câu hỏi: If you create a constructor method to build a DVD by title then create a constructor method to build a DVD by category. Does JAVA allow you to do this?
* Trả lời: Java không chấp nhận. Vì:
* Khi tạo thêm 1 constructor :
* public DigitalVideoDisc(String category) {

super();

this.category = category;

}

* Contructor này đã ghi đè lên Constructor DigitalVideoDisc(String title) và "Java không cho ghi đè Constructor" nên sẽ báo lỗi!
* Trong Java chỉ cho nạp chồng (Overloading) Contructor.