ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO THỰC HÀNH

IT1130-744528-2024.1

BÀI THỰC HÀNH 4

Họ và tên sv: Chu Đình hà

Lớp: **K67 – VN03**

GVHD: Lê Thị Hoa

TA: **Tạ Trung Hiếu**

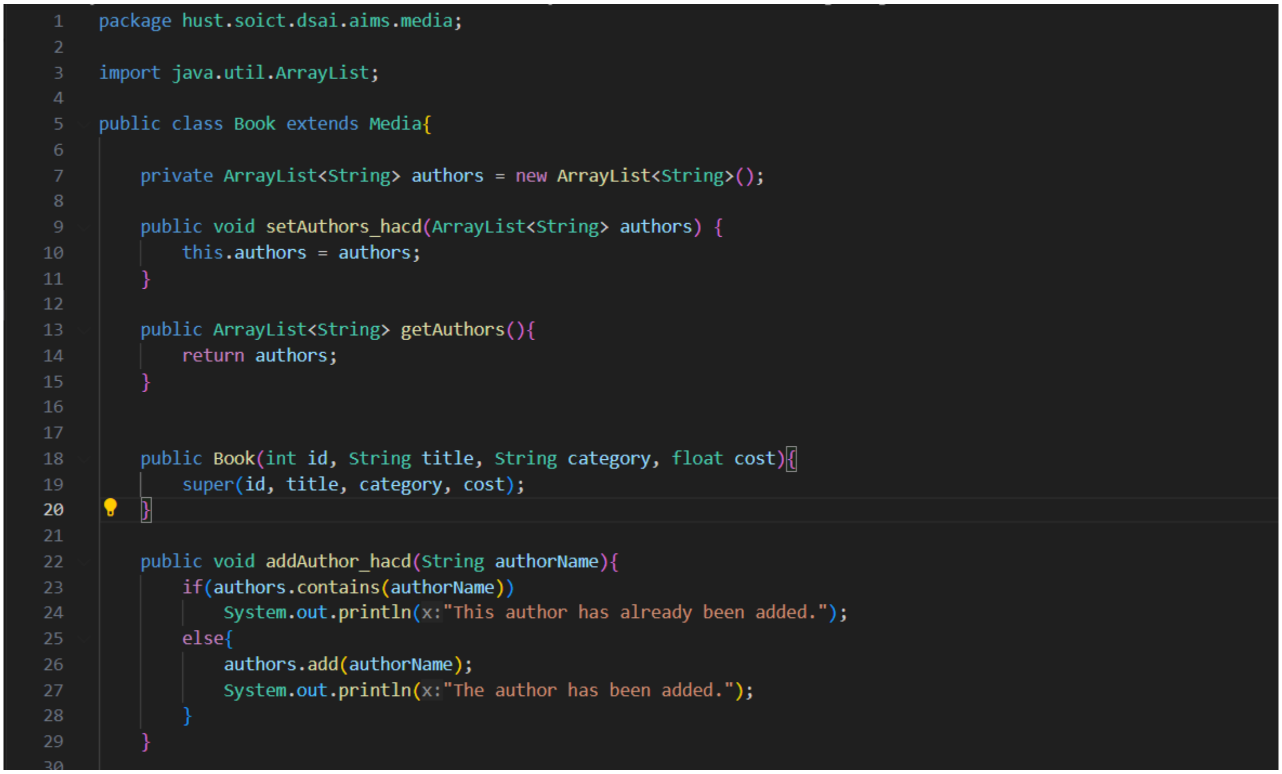
Hà Nội 12/2024

BÁO CÁO THỰC HÀNH LAB 4 LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

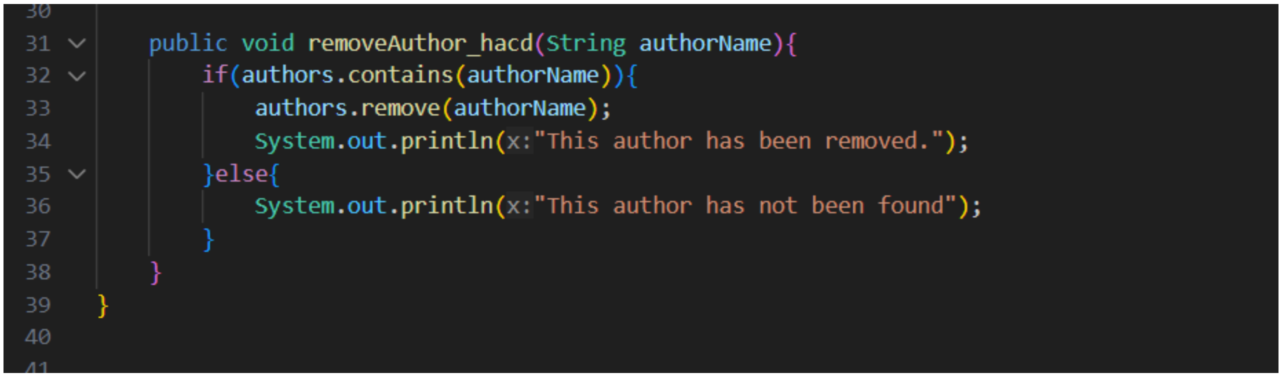
Table of Figures

1 Create the Book class



```
package hust.soict.dsai.aims.media;

import java.util.ArrayList;

public class Book extends Media{

    private ArrayList<String> authors = new ArrayList<String>();

    public void setAuthors_hacd(ArrayList<String> authors) {
        this.authors = authors;
    }

    public ArrayList<String> getAuthors(){
        return authors;
    }


    public Book(int id, String title, String category, float cost){
        super(id, title, category, cost);
    }

    public void addAuthor_hacd(String authorName){
        if(authors.contains(authorName))
            System.out.println(x:"This author has already been added.");
        else{
            authors.add(authorName);
            System.out.println(x:"The author has been added.");
        }
    }
```

Figure 1.1: Book Class 1

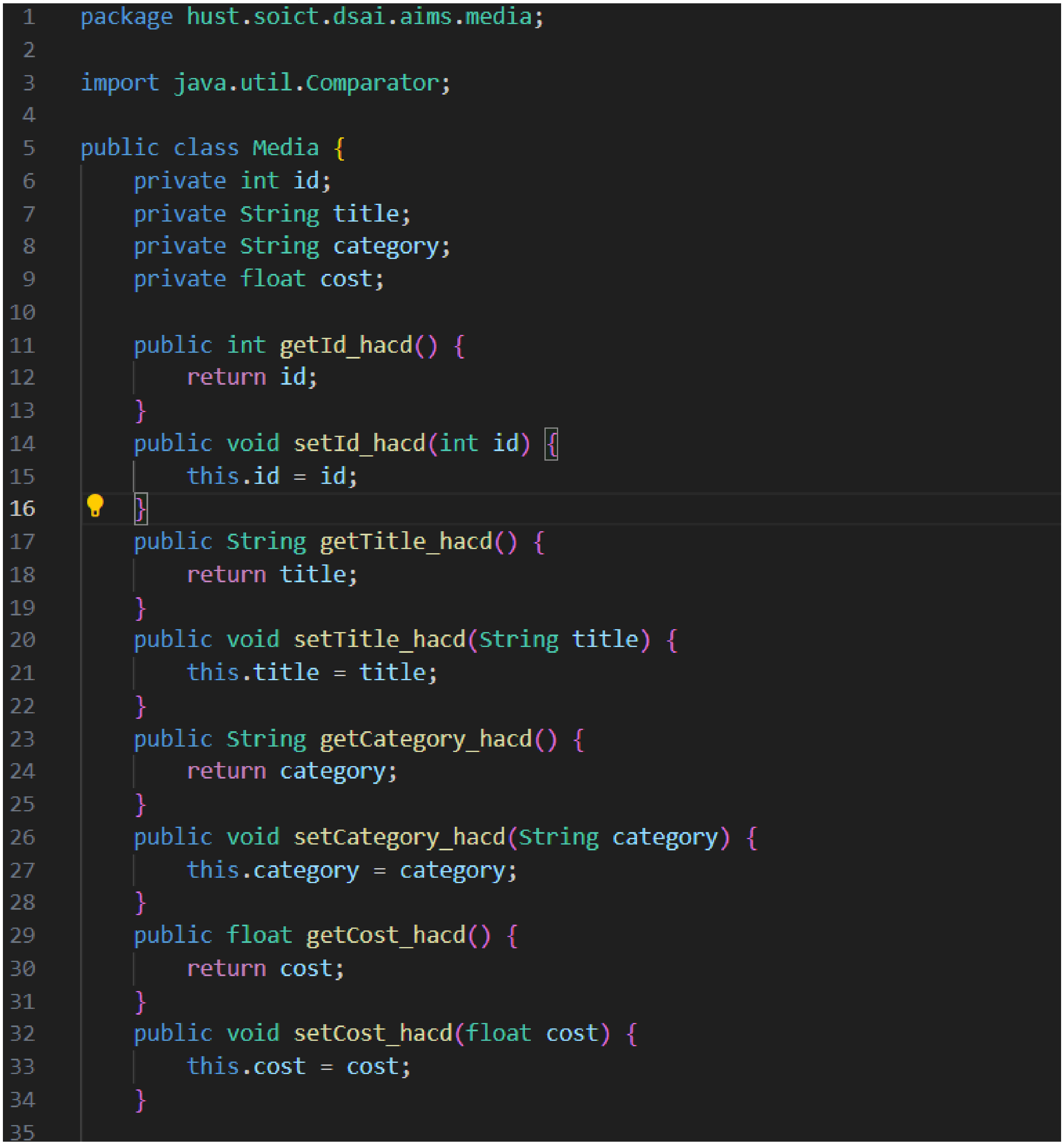


```
    public void removeAuthor_hacd(String authorName){
        if(authors.contains(authorName)){
            authors.remove(authorName);
            System.out.println(x:"This author has been removed.");
        }else{
            System.out.println(x:"This author has not been found");
        }
    }
}
```

Figure 1.2: Book Class 2

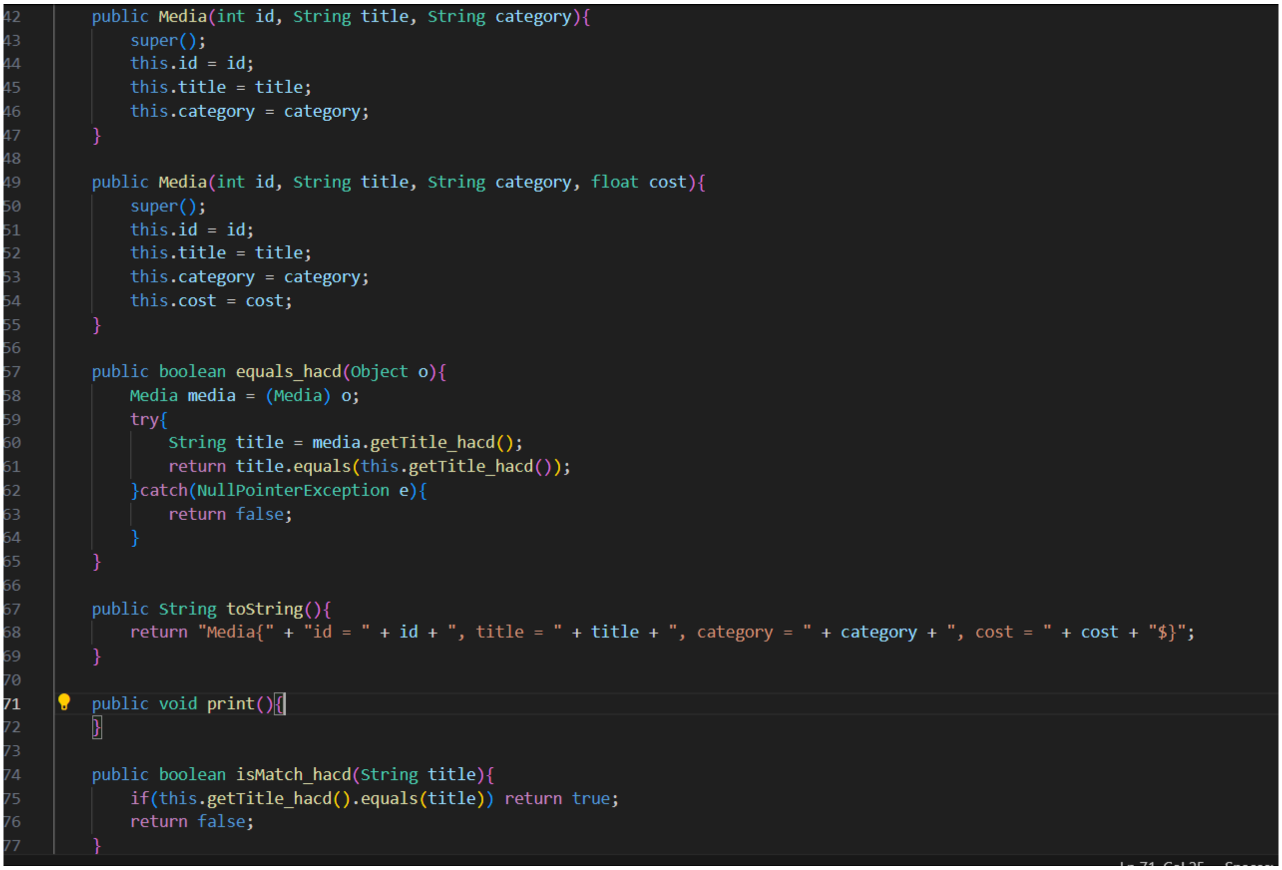
2 Creating the abstract Media class

Đây sẽ là lớp cha để các lớp DigitalVideoDisc, Book kế thừa.



```java
package hust.soict.dsai.aims.media;

import java.util.Comparator;

public class Media {
    private int id;
    private String title;
    private String category;
    private float cost;

    public int getId_hacd() {
        return id;
    }
    public void setId_hacd(int id) {
        this.id = id;
    }
    public String getTitle_hacd() {
        return title;
    }
    public void setTitle_hacd(String title) {
        this.title = title;
    }
    public String getCategory_hacd() {
        return category;
    }
    public void setCategory_hacd(String category) {
        this.category = category;
    }
    public float getCost_hacd() {
        return cost;
    }
    public void setCost_hacd(float cost) {
        this.cost = cost;
    }

```

Figure 2.1: Media Class 1

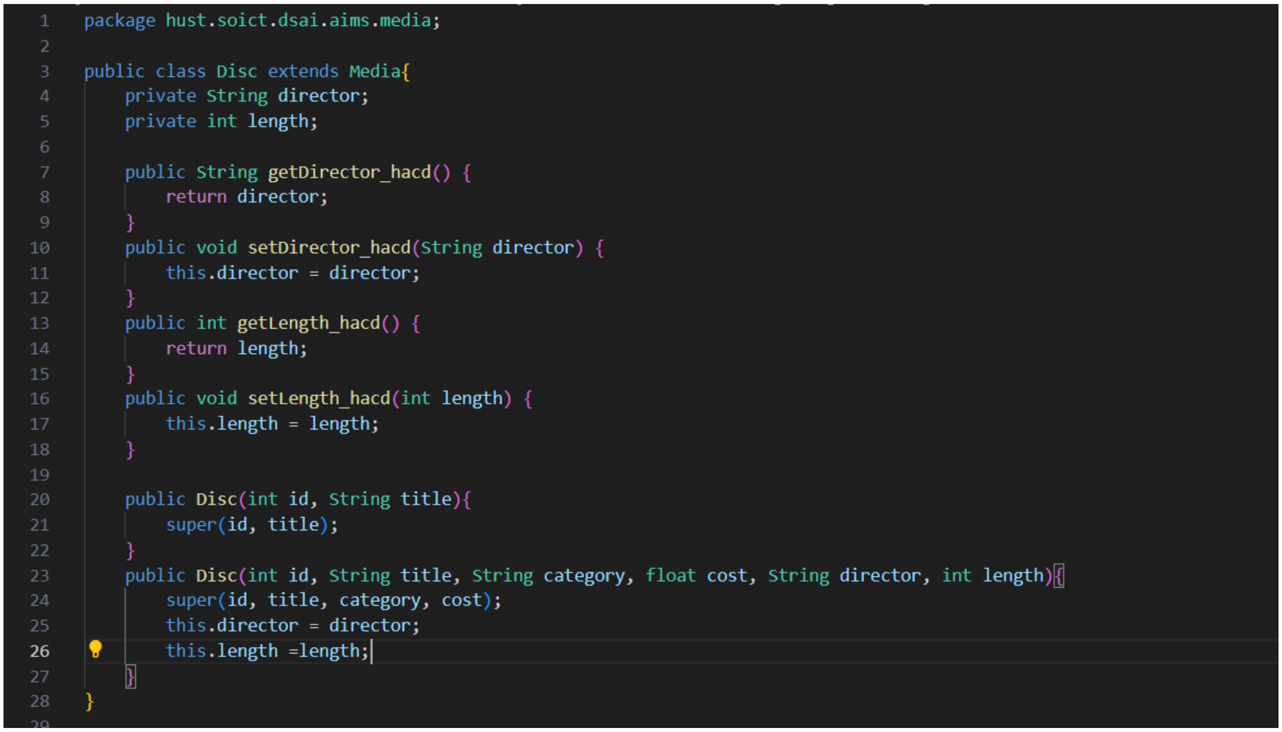


```
    public Media(int id, String title, String category){
        super();
        this.id = id;
        this.title = title;
        this.category = category;
    }

    public Media(int id, String title, String category, float cost){
        super();
        this.id = id;
        this.title = title;
        this.category = category;
        this.cost = cost;
    }

    public boolean equals_hacd(Object o){
        Media media = (Media) o;
        try{
            String title = media.getTitle_hacd();
            return title.equals(this.getTitle_hacd());
        }catch(NullPointerException e){
            return false;
        }
    }

    public String toString(){
        return "Media{" + "id = " + id + ", title = " + title + ", category = " + category + ", cost = " + cost + "$}";
    }

    public void print(){
    }

    public boolean isMatch_hacd(String title){
        if(this.getTitle_hacd().equals(title)) return true;
        return false;
    }
```

Figure 2.2: Media Class 2

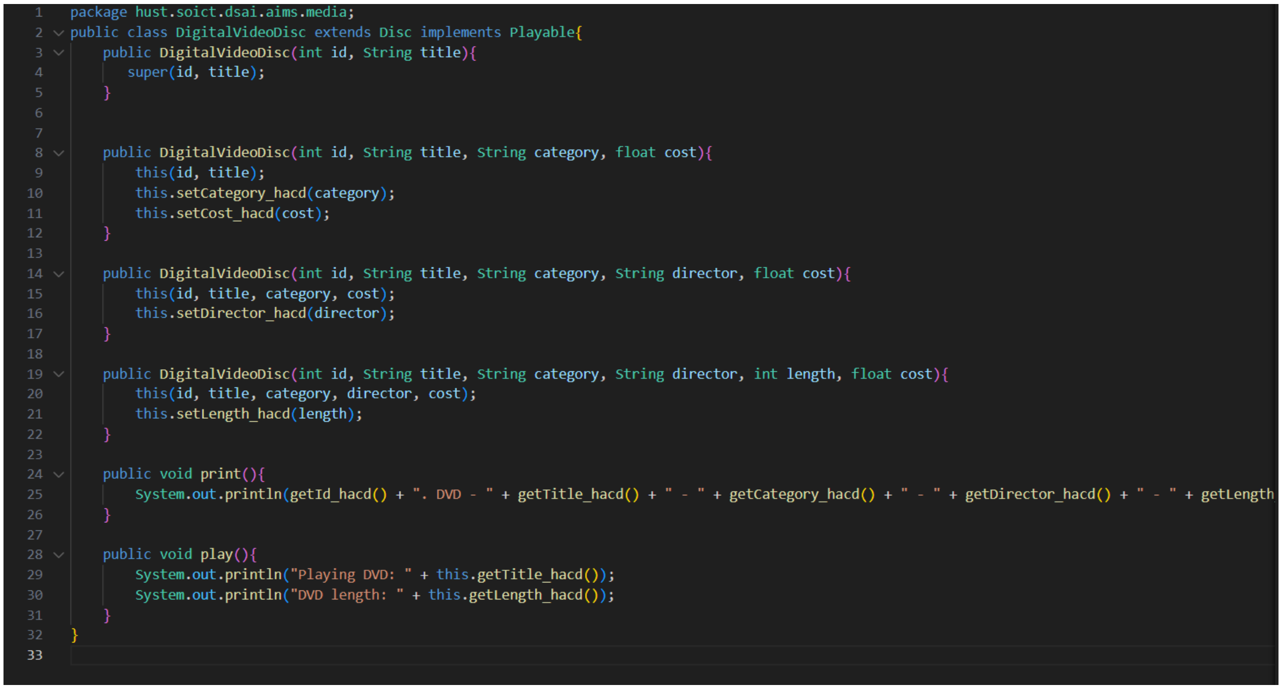
3 Creating the CompactDisc class

3.1 Create the Disc class extending the Media clas



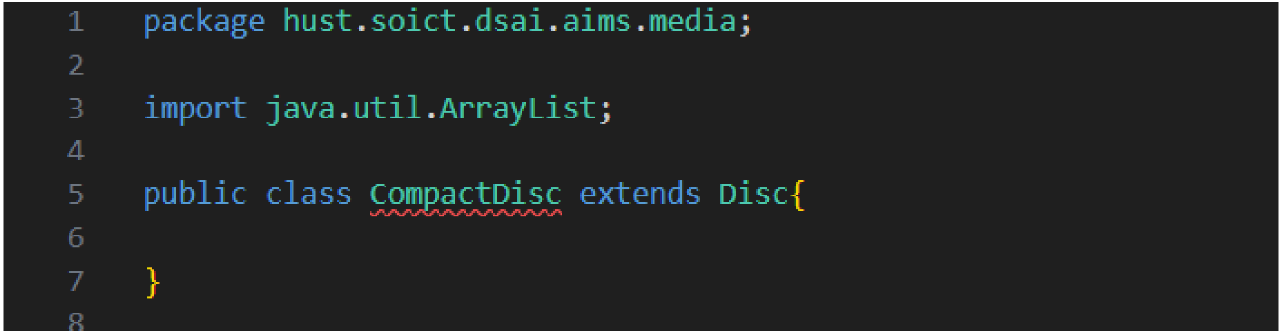
```
package hust.soict.dsai.aims.media;

public class Disc extends Media{
    private String director;
    private int length;

    public String getDirector_hacd() {
        return director;
    }
    public void setDirector_hacd(String director) {
        this.director = director;
    }
    public int getLength_hacd() {
        return length;
    }
    public void setLength_hacd(int length) {
        this.length = length;
    }

    public Disc(int id, String title){
        super(id, title);
    }
    public Disc(int id, String title, String category, float cost, String director, int length){
        super(id, title, category, cost);
        this.director = director;
        this.length =length;
    }
}
```

Figure 3.1: Disc Class



```java
package hust.soict.dsai.aims.media;
public class DigitalVideoDisc extends Disc implements Playable{
    public DigitalVideoDisc(int id, String title){
       super(id, title);
    }


    public DigitalVideoDisc(int id, String title, String category, float cost){
        this(id, title);
        this.setCategory_hacd(category);
        this.setCost_hacd(cost);
    }

    public DigitalVideoDisc(int id, String title, String category, String director, float cost){
        this(id, title, category, cost);
        this.setDirector_hacd(director);
    }

    public DigitalVideoDisc(int id, String title, String category, String director, int length, float cost){
        this(id, title, category, director, cost);
        this.setLength_hacd(length);
    }

    public void print(){
        System.out.println(getId_hacd() + ". DVD - " + getTitle_hacd() + " - " + getCategory_hacd() + " - " + getDirector_hacd() + " - " + getLength
    }

    public void play(){
        System.out.println("Playing DVD: " + this.getTitle_hacd());
        System.out.println("DVD length: " + this.getLength_hacd());
    }
}
```

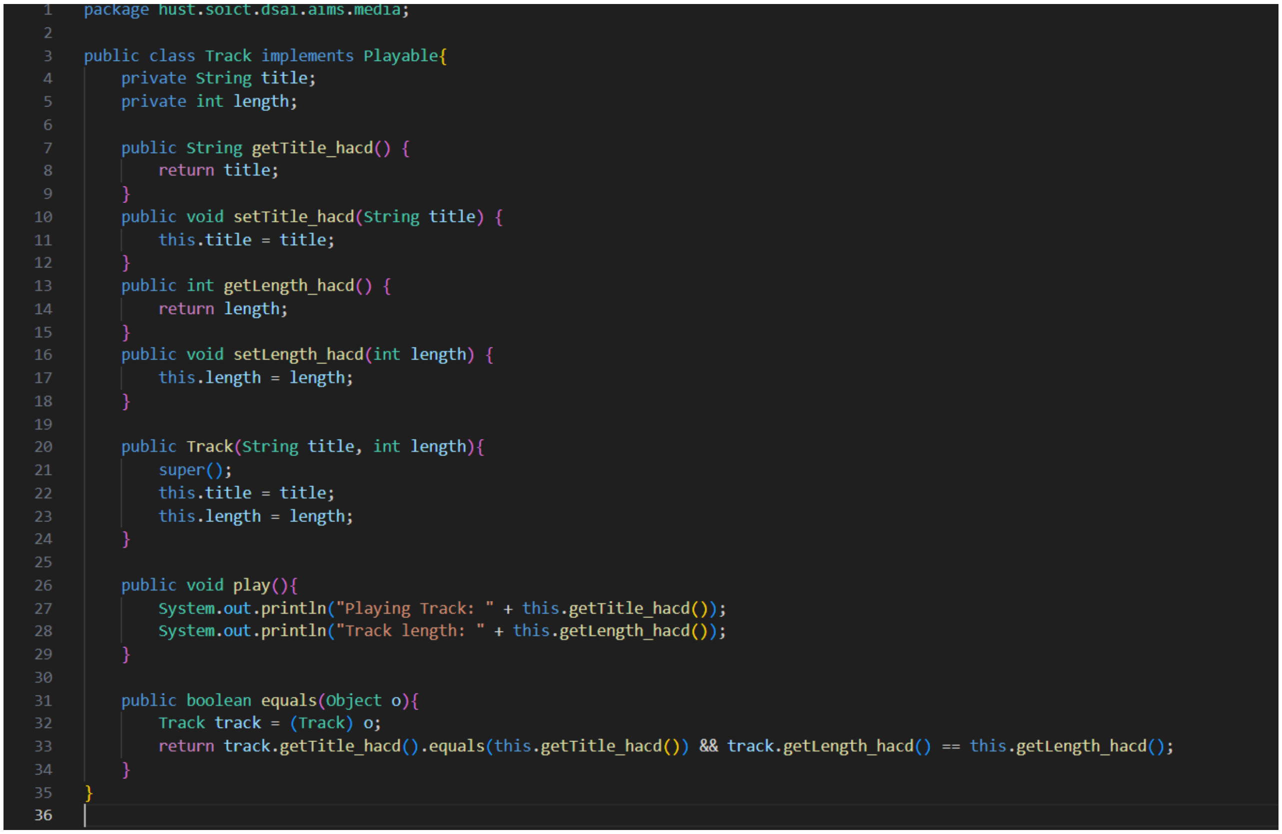
Figure 3.2: DigitalVideoDisc Class



```java
package hust.soict.dsai.aims.media;

import java.util.ArrayList;

public class CompactDisc extends Disc{

}
```

Figure 3.3: CompactDisc Class

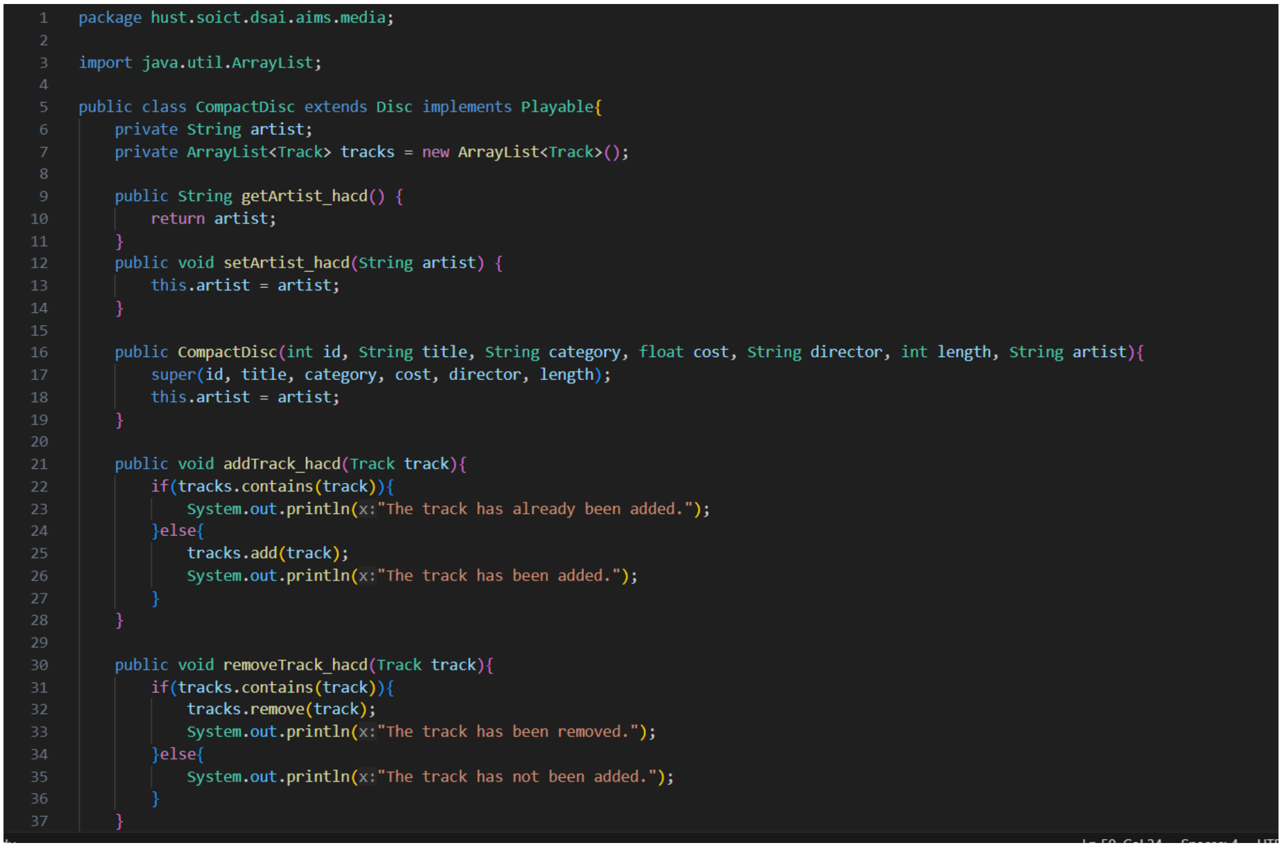
3.2 Create the Track class which models a track on a compact disc and will store information incuding the title and length of the track.



```java
package hust.soict.dsai.aims.media;

public class Track implements Playable{
    private String title;
    private int length;

    public String getTitle_hacd() {
        return title;
    }
    public void setTitle_hacd(String title) {
        this.title = title;
    }
    public int getLength_hacd() {
        return length;
    }
    public void setLength_hacd(int length) {
        this.length = length;
    }

    public Track(String title, int length){
        super();
        this.title = title;
        this.length = length;
    }

    public void play(){
        System.out.println("Playing Track: " + this.getTitle_hacd());
        System.out.println("Track length: " + this.getLength_hacd());
    }

    public boolean equals(Object o){
        Track track = (Track) o;
        return track.getTitle_hacd().equals(this.getTitle_hacd()) && track.getLength_hacd() == this.getLength_hacd();
    }
}

```

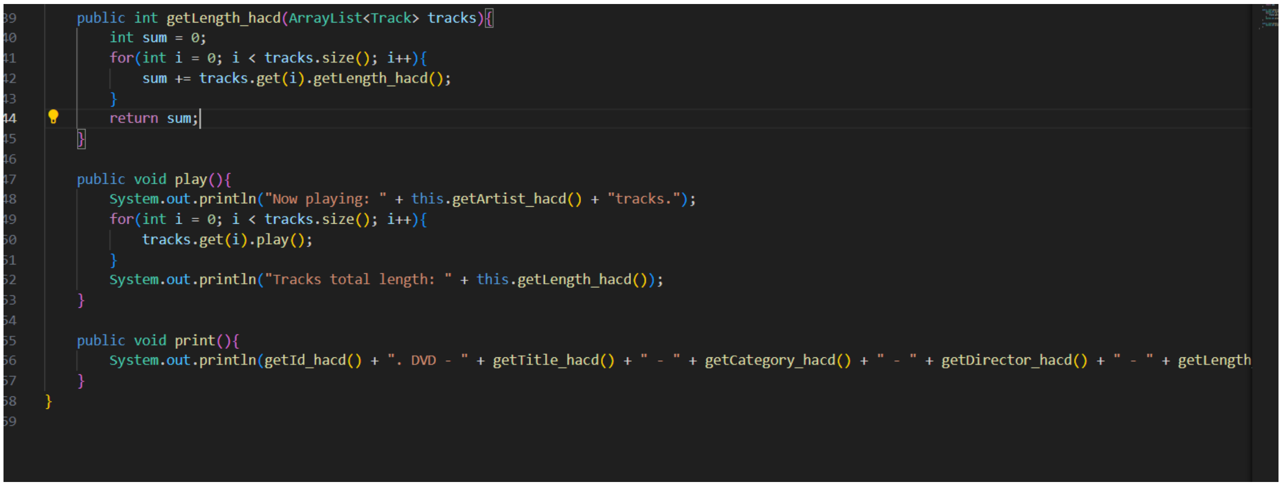
Figure 3.4: Track Class

3.3 Open the CompactDisc class



```java
package hust.soict.dsai.aims.media;

import java.util.ArrayList;

public class CompactDisc extends Disc implements Playable{
    private String artist;
    private ArrayList<Track> tracks = new ArrayList<Track>();

    public String getArtist_hacd() {
        return artist;
    }
    public void setArtist_hacd(String artist) {
        this.artist = artist;
    }

    public CompactDisc(int id, String title, String category, float cost, String director, int length, String artist){
        super(id, title, category, cost, director, length);
        this.artist = artist;
    }

    public void addTrack_hacd(Track track){
        if(tracks.contains(track)){
            System.out.println(x:"The track has already been added.");
        }else{
            tracks.add(track);
            System.out.println(x:"The track has been added.");
        }
    }

    public void removeTrack_hacd(Track track){
        if(tracks.contains(track)){
            tracks.remove(track);
            System.out.println(x:"The track has been removed.");
        }else{
            System.out.println(x:"The track has not been added.");
        }
    }
```

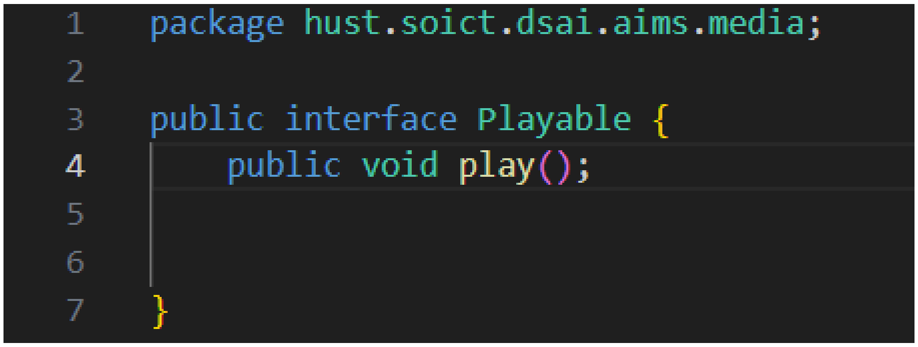
Figure 3.5: CompactDisc Class 1



```
    public int getLength_hacd(ArrayList<Track> tracks){
        int sum = 0;
        for(int i = 0; i < tracks.size(); i++){
            sum += tracks.get(i).getLength_hacd();
        }
        return sum;
    }

    public void play(){
        System.out.println("Now playing: " + this.getArtist_hacd() + "tracks.");
        for(int i = 0; i < tracks.size(); i++){
            tracks.get(i).play();
        }
        System.out.println("Tracks total length: " + this.getLength_hacd());
    }

    public void print(){
        System.out.println(getId_hacd() + ". DVD - " + getTitle_hacd() + " - " + getCategory_hacd() + " - " + getDirector_hacd() + " - " + getLength
    }
}
```

Figure 3.6: CompactDisc Class 2

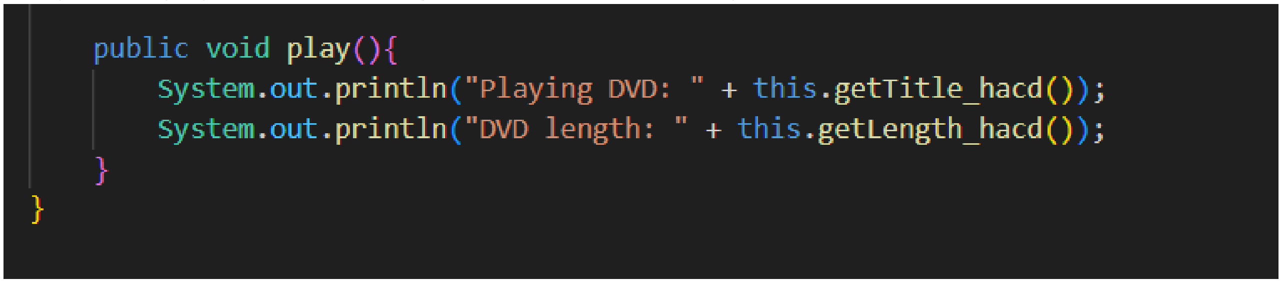
4 Create the Playable interface



```
package hust.soict.dsai.aims.media;

public interface Playable {
    public void play();


}
```

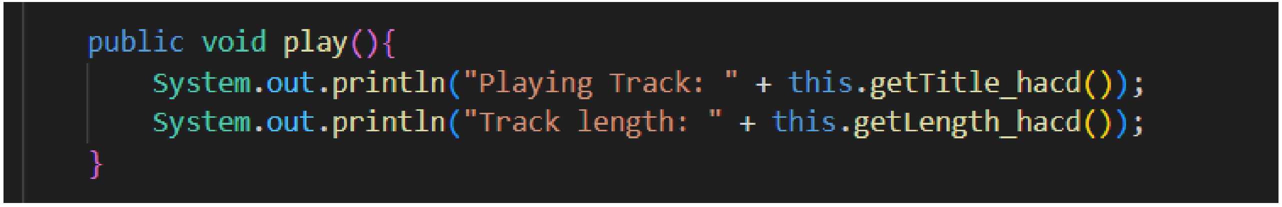
Figure 4.1: Playable interface

Implement play() cho các class DigitalVideoDisc, Track, CompactDisc



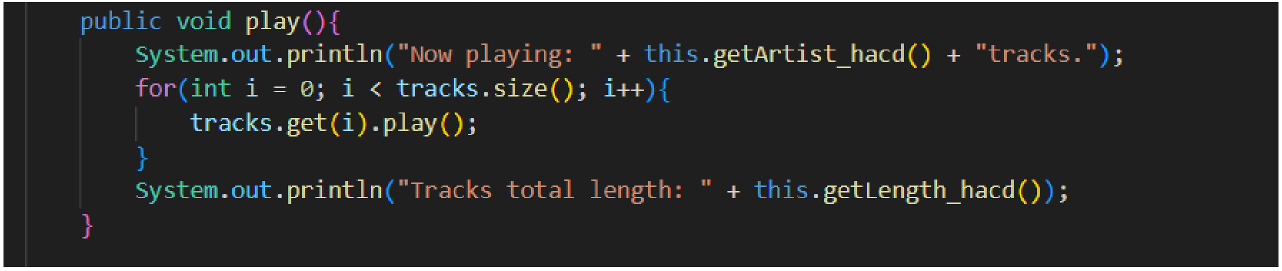
```
    public void play(){
        System.out.println("Playing DVD: " + this.getTitle_hacd());
        System.out.println("DVD length: " + this.getLength_hacd());
    }
}
```

Figure 4.2: Method play() của DigitalVideoDisc



```
    public void play(){
        System.out.println("Playing Track: " + this.getTitle_hacd());
        System.out.println("Track length: " + this.getLength_hacd());
    }
```

Figure 4.3: Method play() của Track

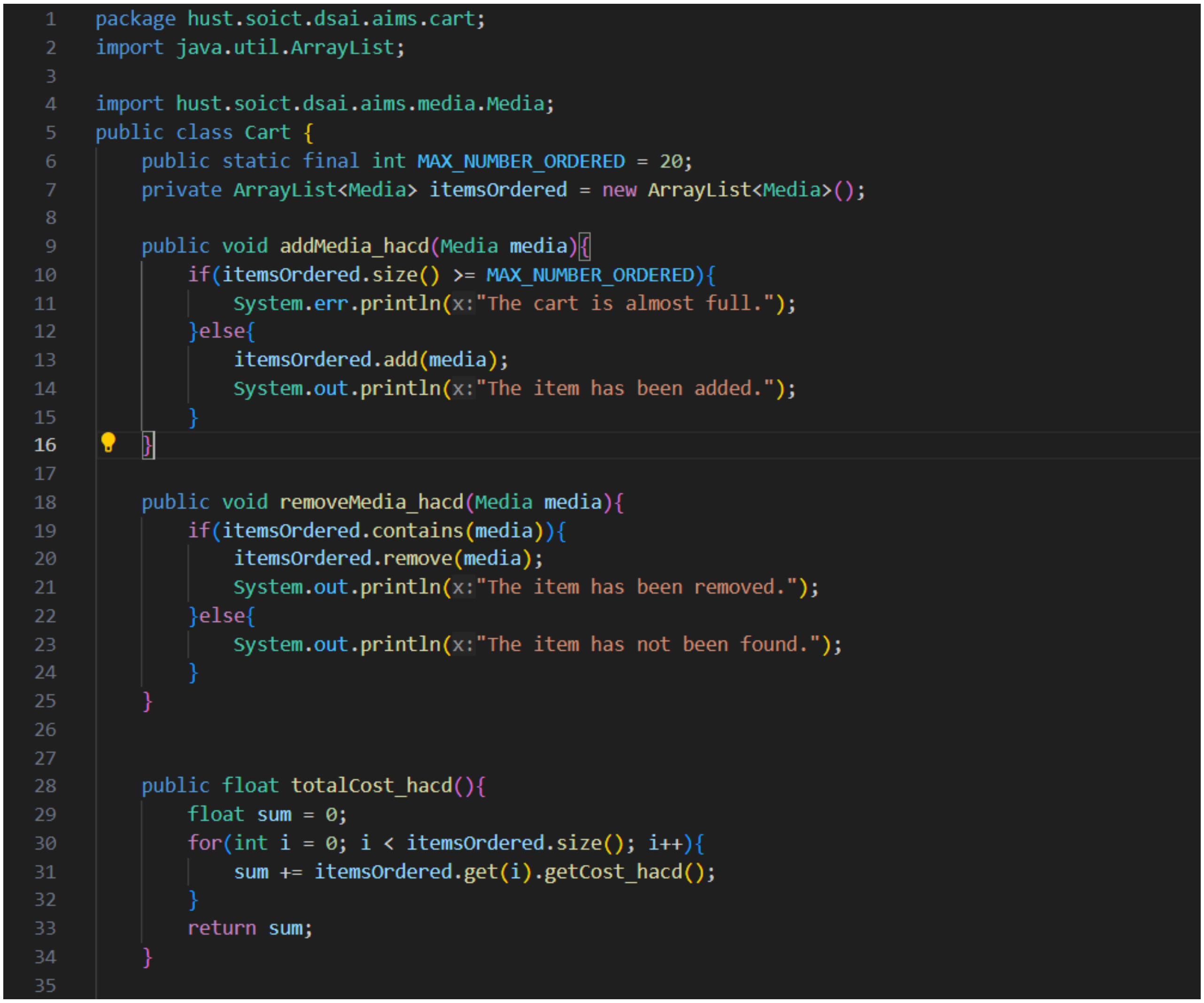


```
    public void play(){
        System.out.println("Now playing: " + this.getArtist_hacd() + "tracks.");
        for(int i = 0; i < tracks.size(); i++){
            tracks.get(i).play();
        }
        System.out.println("Tracks total length: " + this.getLength_hacd());
    }
```

Figure 4.4: Method play() của CompactDisc

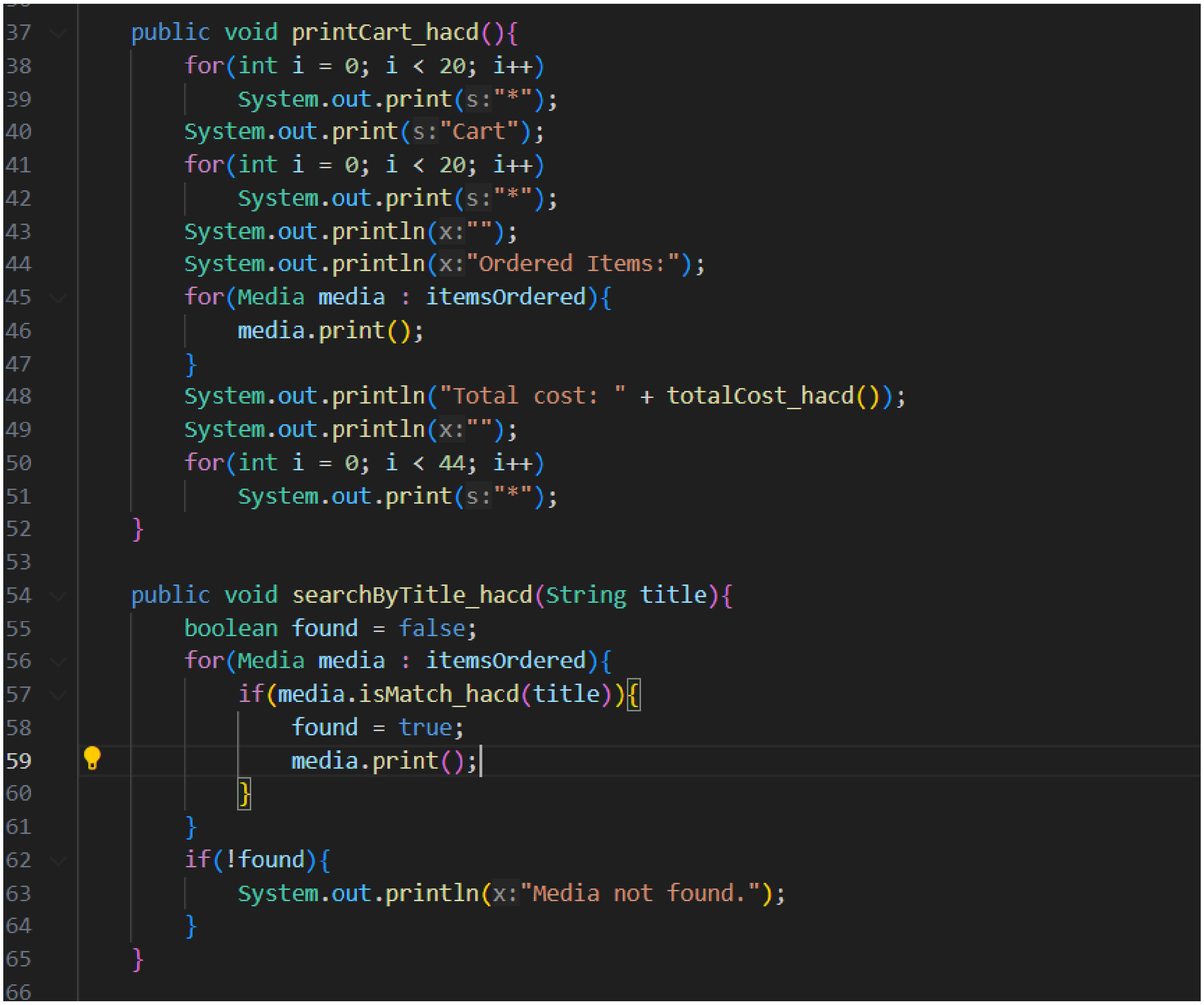
5 Update the Cart class to work with Media

Lớp Cart bây giờ cần có khả năng tương tác với các đối tượng DVD, CD và Book. Vì các lớp DVD, CD và Book đều kế thừa từ lớp Media, nên thay vì làm việc trực tiếp với từng lớp con, lớp cart chỉ cần giao tiếp với lớp Media là có thể hoạt động được với tất cả.



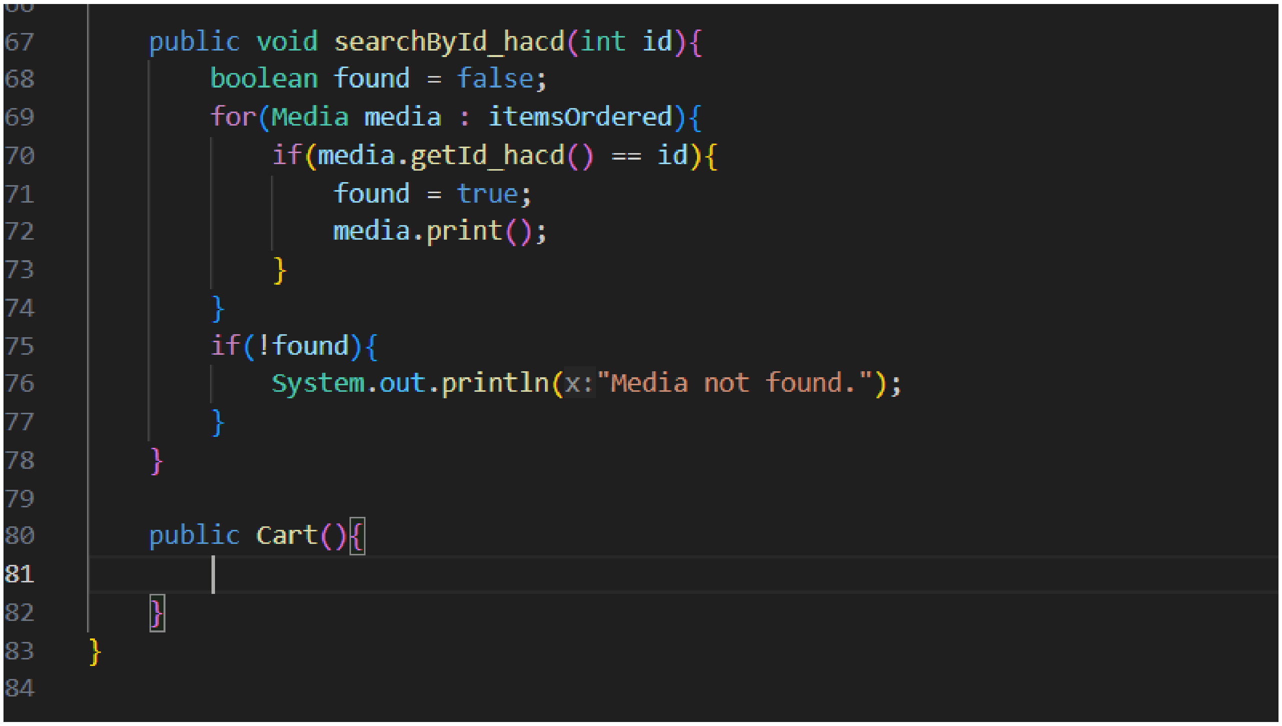
```java
package hust.soict.dsai.aims.cart;
import java.util.ArrayList;

import hust.soict.dsai.aims.media.Media;
public class Cart {
    public static final int MAX_NUMBER_ORDERED = 20;
    private ArrayList<Media> itemsOrdered = new ArrayList<Media>();

    public void addMedia_hacd(Media media){
        if(itemsOrdered.size() >= MAX_NUMBER_ORDERED){
            System.err.println(x:"The cart is almost full.");
        }else{
            itemsOrdered.add(media);
            System.out.println(x:"The item has been added.");
        }
    }

    public void removeMedia_hacd(Media media){
        if(itemsOrdered.contains(media)){
            itemsOrdered.remove(media);
            System.out.println(x:"The item has been removed.");
        }else{
            System.out.println(x:"The item has not been found.");
        }
    }


    public float totalCost_hacd(){
        float sum = 0;
        for(int i = 0; i < itemsOrdered.size(); i++){
            sum += itemsOrdered.get(i).getCost_hacd();
        }
        return sum;
    }

```

Figure 5.1: Cart Class 1



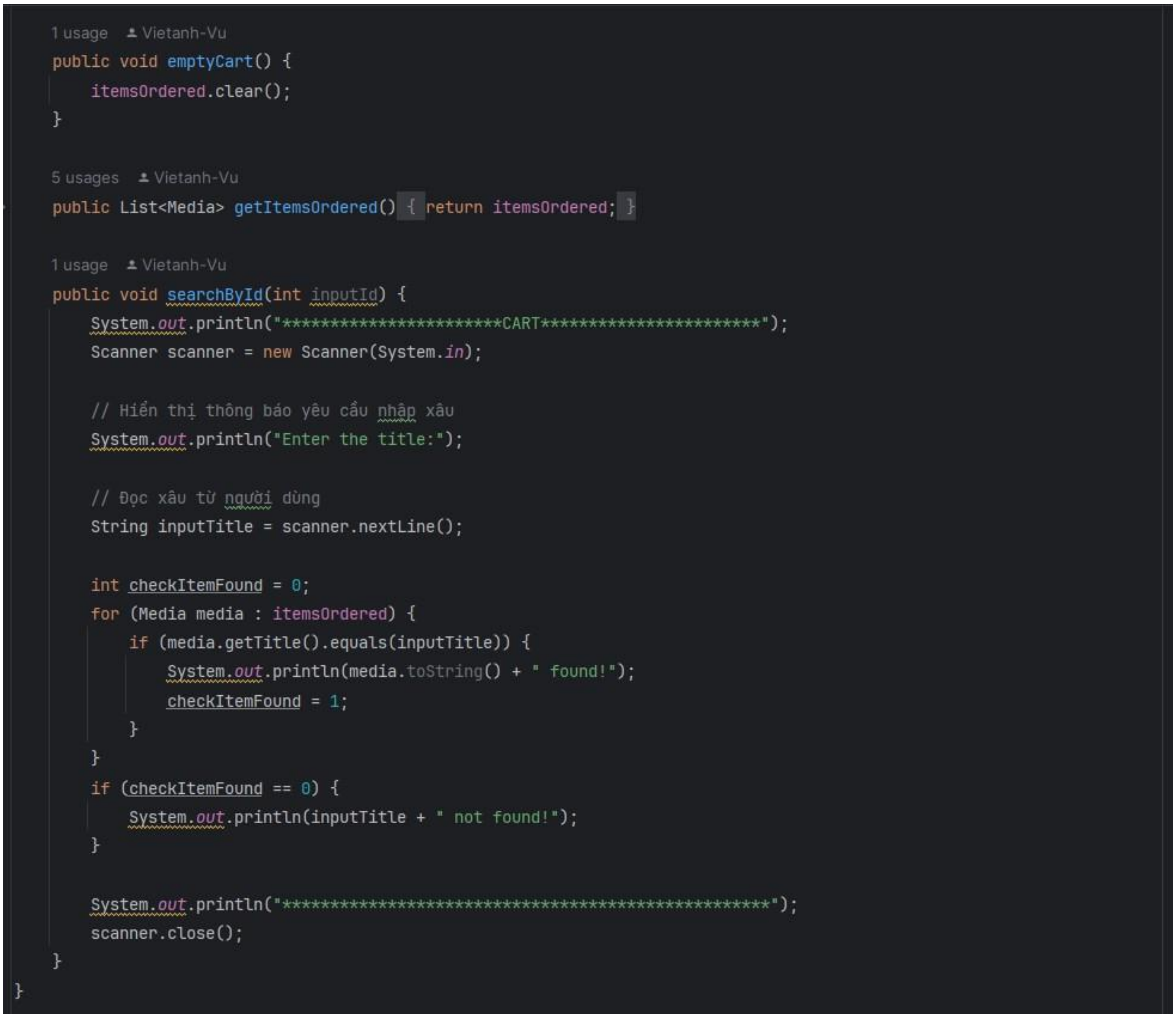
```
    public void printCart_hacd(){
        for(int i = 0; i < 20; i++)
            System.out.print(s:"*");
        System.out.print(s:"Cart");
        for(int i = 0; i < 20; i++)
            System.out.print(s:"*");
        System.out.println(x:"");
        System.out.println(x:"Ordered Items:");
        for(Media media : itemsOrdered){
            media.print();
        }
        System.out.println("Total cost: " + totalCost_hacd());
        System.out.println(x:"");
        for(int i = 0; i < 44; i++)
            System.out.print(s:"*");
    }

    public void searchByTitle_hacd(String title){
        boolean found = false;
        for(Media media : itemsOrdered){
            if(media.isMatch_hacd(title)){
                found = true;
                media.print();
            }
        }
        if(!found){
            System.out.println(x:"Media not found.");
        }
    }

```

Figure 5.2: Cart Class 2



```
    public void searchById_hacd(int id){
        boolean found = false;
        for(Media media : itemsOrdered){
            if(media.getId_hacd() == id){
                found = true;
                media.print();
            }
        }
        if(!found){
            System.out.println(x:"Media not found.");
        }
    }

    public Cart(){

    }
}
```

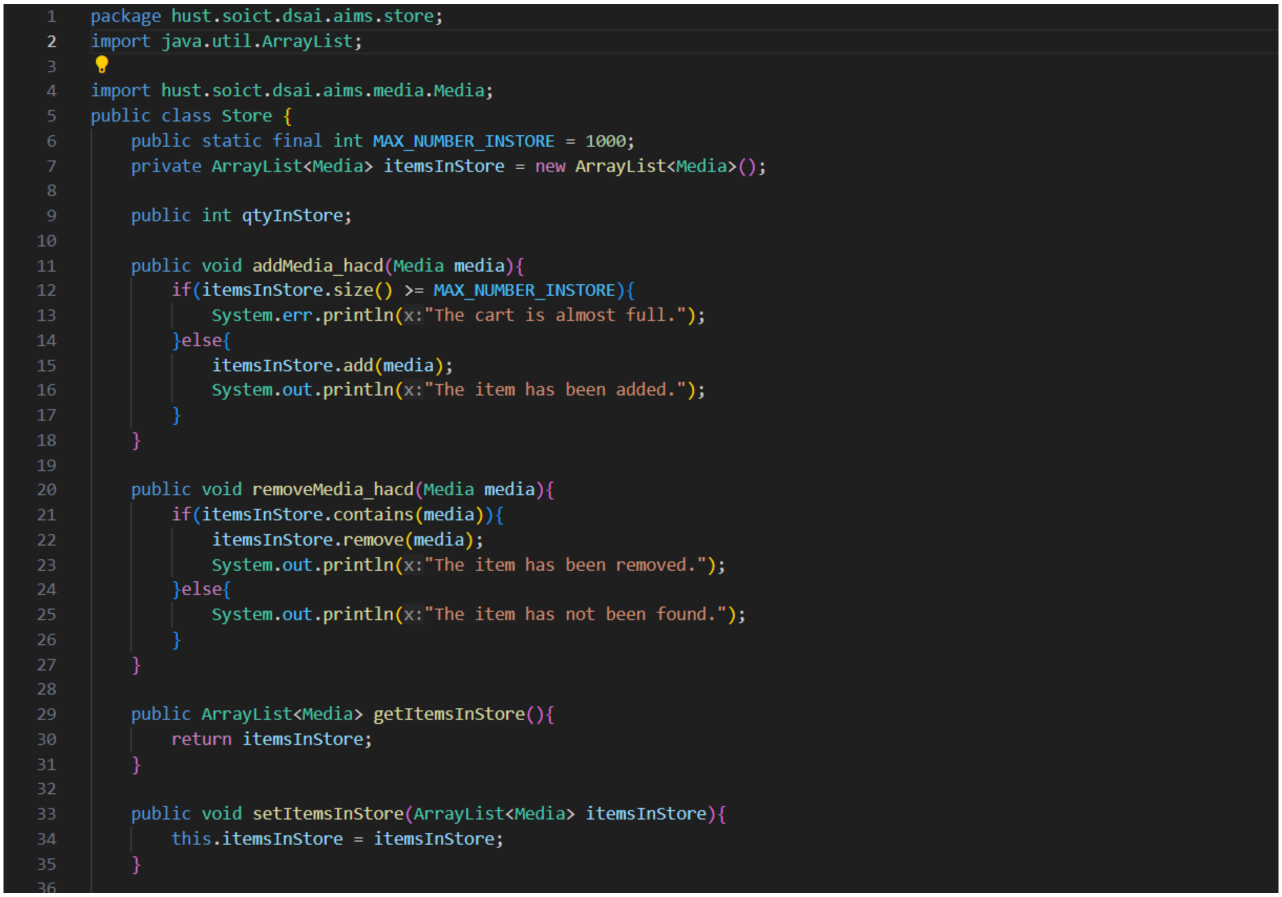
Figure 5.3: Cart Class 3



```
    1 usage   Vietanh-Vu
    public void emptyCart() {
        itemsOrdered.clear();
    }

    5 usages   Vietanh-Vu
    public List<Media> getItemsOrdered() { return itemsOrdered; }

    1 usage   Vietanh-Vu
    public void searchById(int inputId) {
        System.out.println("***********************CART***********************");
        Scanner scanner = new Scanner(System.in);

        // Hiển thị thông báo yêu cầu nhập xâu
        System.out.println("Enter the title:");

        // Đọc xâu từ người dùng
        String inputTitle = scanner.nextLine();

        int checkItemFound = 0;
        for (Media media : itemsOrdered) {
            if (media.getTitle().equals(inputTitle)) {
                System.out.println(media.toString() + " found!");
                checkItemFound = 1;
            }
        }
        if (checkItemFound == 0) {
            System.out.println(inputTitle + " not found!");
        }

        System.out.println("***************************************************");
        scanner.close();
    }
}
```

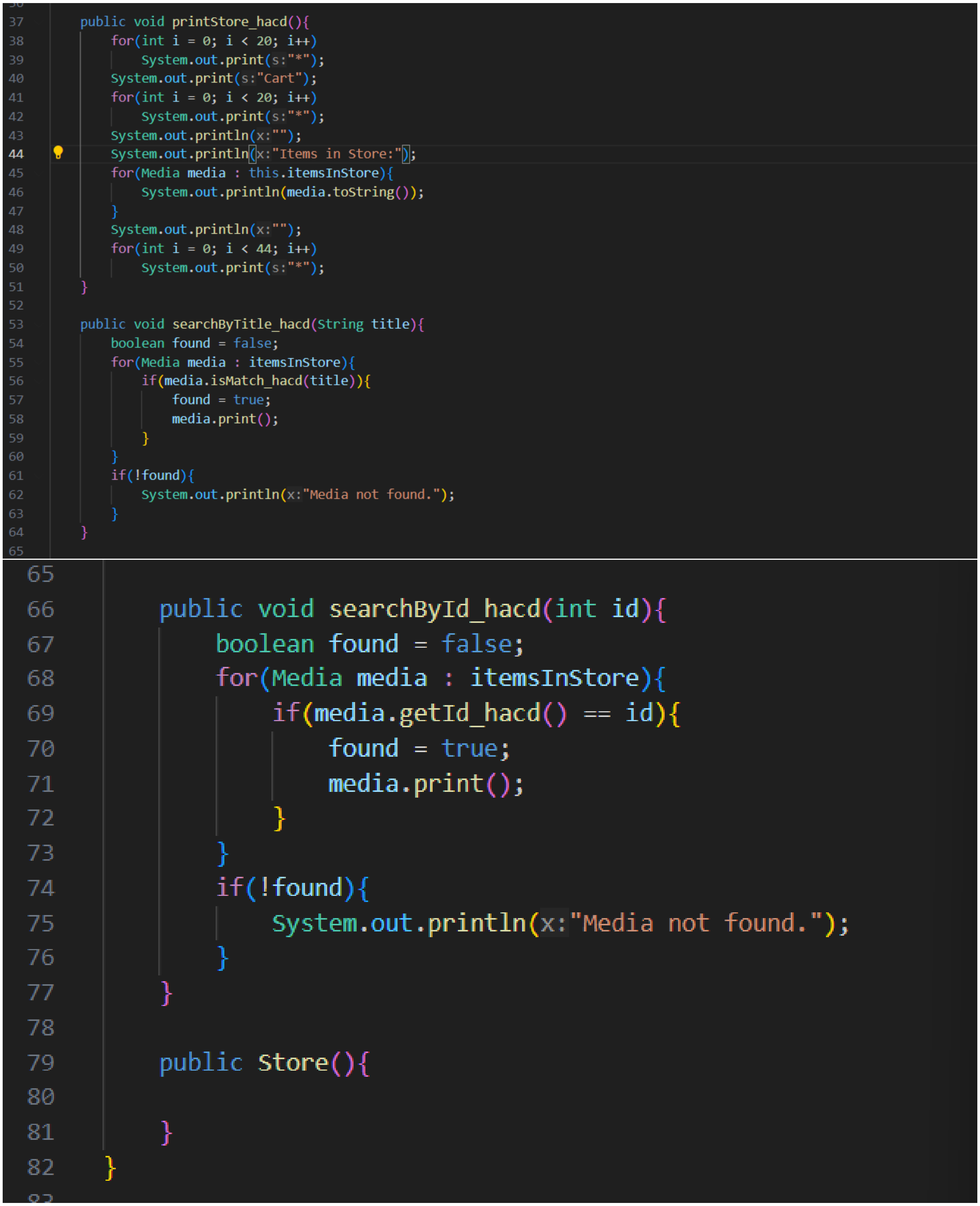
Figure 5.4: Cart Class 4

6 Update the Store class to work with Media



```java
package hust.soict.dsai.aims.store;
import java.util.ArrayList;

import hust.soict.dsai.aims.media.Media;
public class Store {
    public static final int MAX_NUMBER_INSTORE = 1000;
    private ArrayList<Media> itemsInStore = new ArrayList<Media>();

    public int qtyInStore;

    public void addMedia_hacd(Media media){
        if(itemsInStore.size() >= MAX_NUMBER_INSTORE){
            System.err.println(x:"The cart is almost full.");
        }else{
            itemsInStore.add(media);
            System.out.println(x:"The item has been added.");
        }
    }

    public void removeMedia_hacd(Media media){
        if(itemsInStore.contains(media)){
            itemsInStore.remove(media);
            System.out.println(x:"The item has been removed.");
        }else{
            System.out.println(x:"The item has not been found.");
        }
    }

    public ArrayList<Media> getItemsInStore(){
        return itemsInStore;
    }

    public void setItemsInStore(ArrayList<Media> itemsInStore){
        this.itemsInStore = itemsInStore;
    }

```

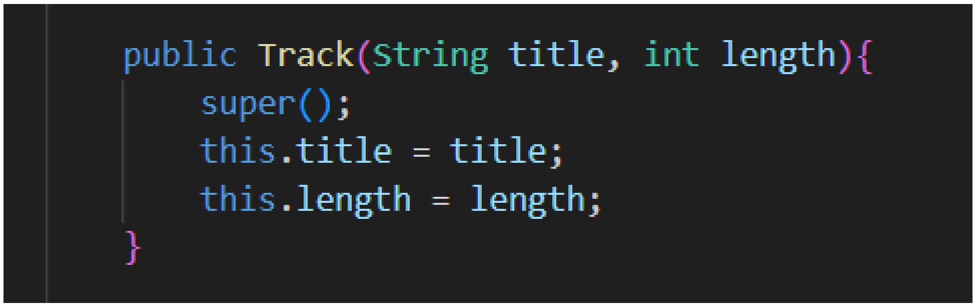
Figure 6.1: Store Class 1



```
    public void printStore_hacd(){
        for(int i = 0; i < 20; i++)
            System.out.print(s:"*");
        System.out.print(s:"Cart");
        for(int i = 0; i < 20; i++)
            System.out.print(s:"*");
        System.out.println(x:"");
        System.out.println(x:"Items in Store:");
        for(Media media : this.itemsInStore){
            System.out.println(media.toString());
        }
        System.out.println(x:"");
        for(int i = 0; i < 44; i++)
            System.out.print(s:"*");
    }

    public void searchByTitle_hacd(String title){
        boolean found = false;
        for(Media media : itemsInStore){
            if(media.isMatch_hacd(title)){
                found = true;
                media.print();
            }
        }
        if(!found){
            System.out.println(x:"Media not found.");
        }
    }

    public void searchById_hacd(int id){
        boolean found = false;
        for(Media media : itemsInStore){
            if(media.getId_hacd() == id){
                found = true;
                media.print();
            }
        }
        if(!found){
            System.out.println(x:"Media not found.");
        }
    }

    public Store(){

    }
}
```

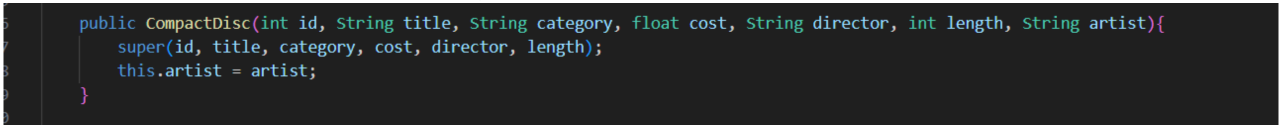
Figure 6.2: Store Class 2

7 Constructors of whole classes and parent classes



```java
public Track(String title, int length){
    super();
    this.title = title;
    this.length = length;
}
```

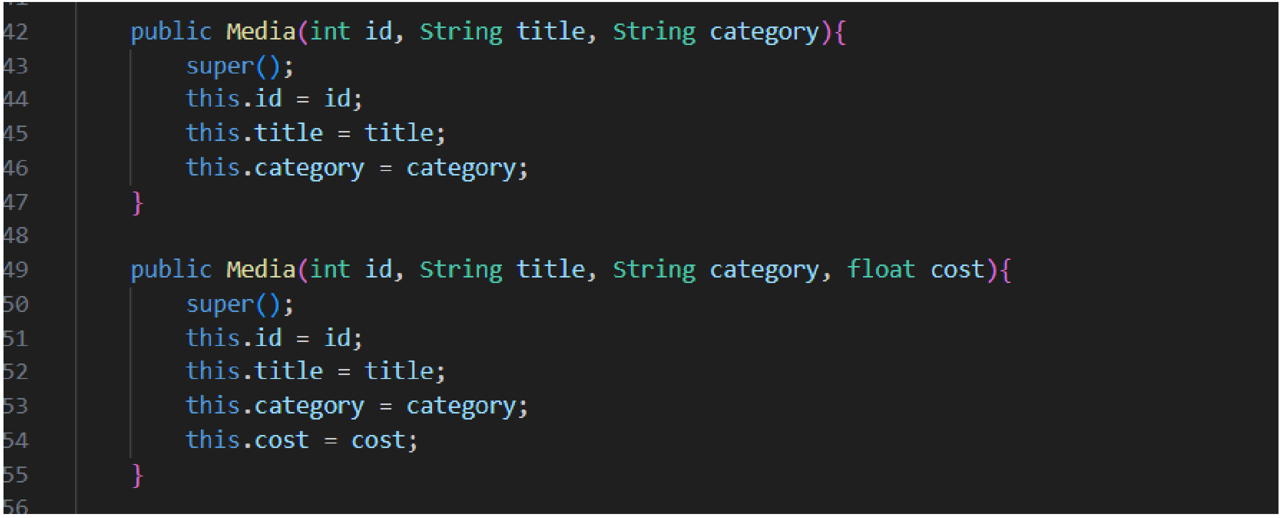
Figure 7.1: Constructor Track Class



```java
public CompactDisc(int id, String title, String category, float cost, String director, int length, String artist){
    super(id, title, category, cost, director, length);
    this.artist = artist;
}
```

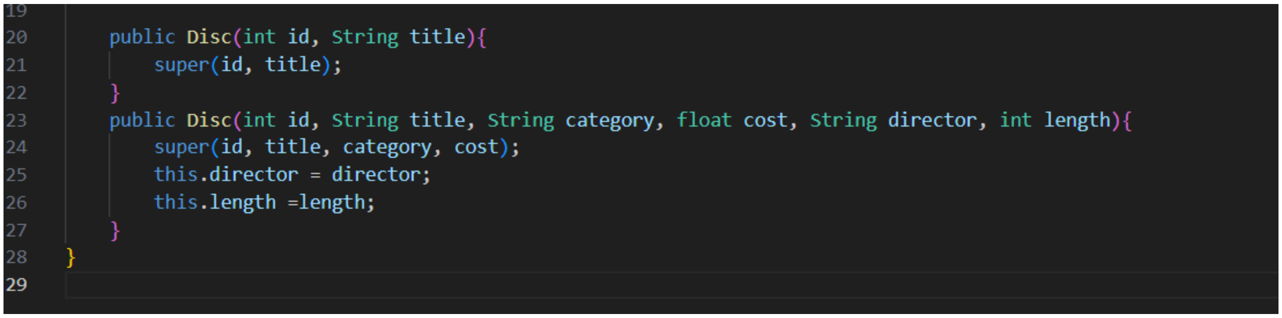
Figure 7.2: Constructor CompactDisc Class

Lớp Disc kế thừa lớp Media, khi đó lớp Media là lớp cha, lớp Disc là lớp con.



```java
public Media(int id, String title, String category){
    super();
    this.id = id;
    this.title = title;
    this.category = category;
}

public Media(int id, String title, String category, float cost){
    super();
    this.id = id;
    this.title = title;
    this.category = category;
    this.cost = cost;
}
```

Figure 7.3: Constructor Media Class

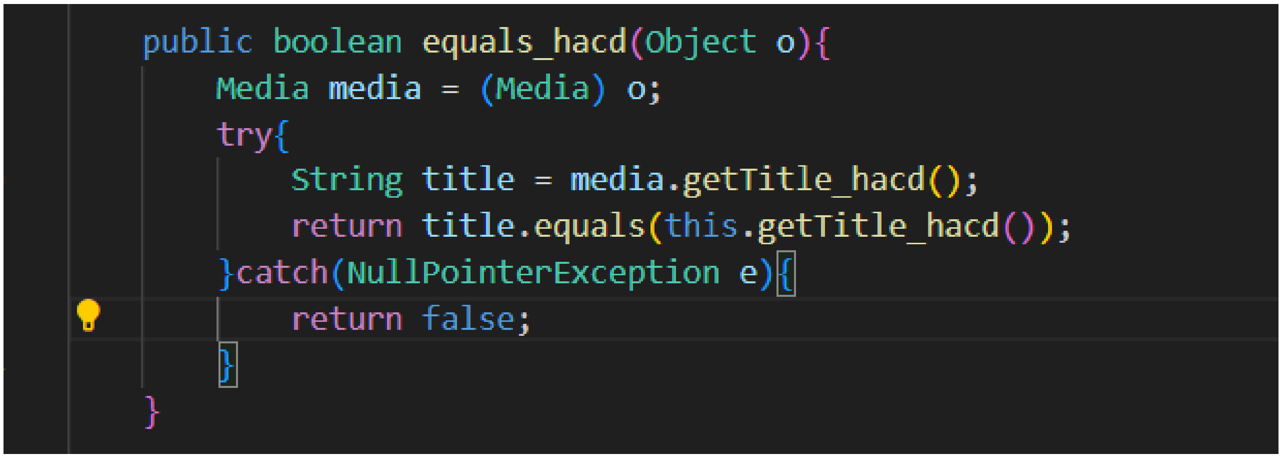


```
    public Disc(int id, String title){
        super(id, title);
    }
    public Disc(int id, String title, String category, float cost, String director, int length){
        super(id, title, category, cost);
        this.director = director;
        this.length =length;
    }
}
```

Figure 7.4: Constructor Disc Class

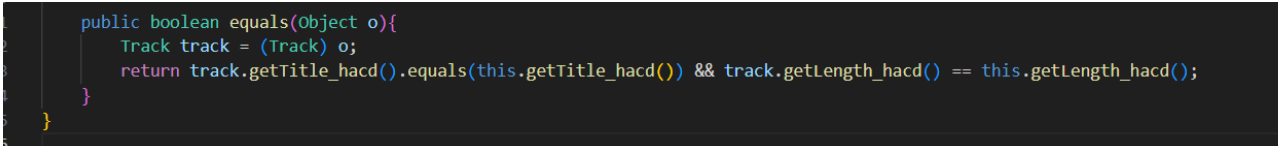
8 Unique item in a list

Để tránh trùng lặp các phần tử media trong giỏ hàng hoặc các track trong một đĩa CD, chúng ta có thể ghi đè lại phương thức equals() mặc định kế thừa từ lớp Object. Việc này cho phép so sánh bản chất thay vì so sánh vị trí ô nhớ của các đối tượng, qua đó ngăn chặn thêm các phần tử bị trùng lắp vào danh sách.



```
    public boolean equals_hacd(Object o){
        Media media = (Media) o;
        try{
            String title = media.getTitle_hacd();
            return title.equals(this.getTitle_hacd());
        }catch(NullPointerException e){
            return false;
        }
    }
```

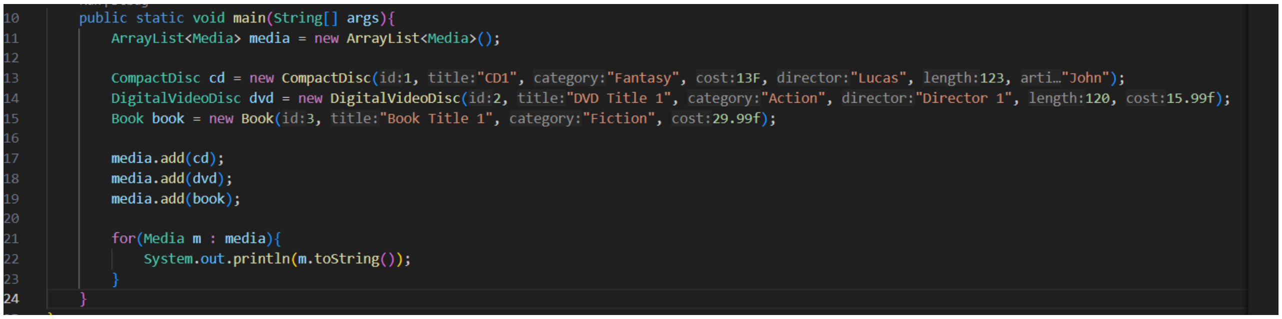
Figure 8.1: Override equals in Media Class



```
    public boolean equals(Object o){
        Track track = (Track) o;
        return track.getTitle_hacd().equals(this.getTitle_hacd()) && track.getLength_hacd() == this.getLength_hacd();
    }
}
```

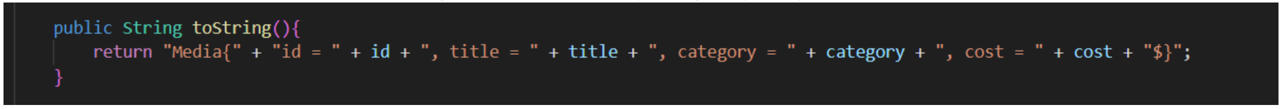
Figure 8.2: Override equals in Track Class

9 Polymorphism with toString() method



```java
    public static void main(String[] args){
        ArrayList<Media> media = new ArrayList<Media>();

        CompactDisc cd = new CompactDisc(id:1, title:"CD1", category:"Fantasy", cost:13F, director:"Lucas", length:123, arti…"John");
        DigitalVideoDisc dvd = new DigitalVideoDisc(id:2, title:"DVD Title 1", category:"Action", director:"Director 1", length:120, cost:15.99f);
        Book book = new Book(id:3, title:"Book Title 1", category:"Fiction", cost:29.99f);

        media.add(cd);
        media.add(dvd);
        media.add(book);

        for(Media m : media){
            System.out.println(m.toString());
        }
    }
```

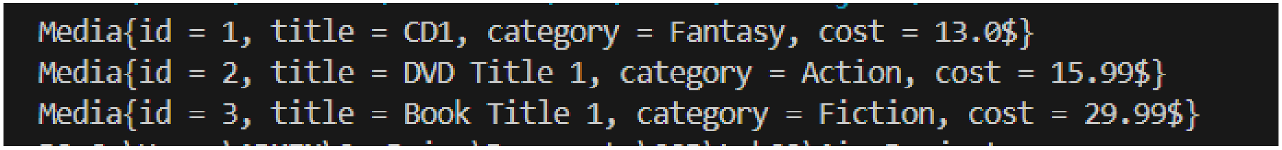
Figure 9.1: Code mô phỏng Polymorphism



```java
public String toString(){
    return "Media{" + "id = " + id + ", title = " + title + ", category = " + category + ", cost = " + cost + "$}";
}
```

Figure 9.2: Override toString() in Media Class

Kết quả



```
Media{id = 1, title = CD1, category = Fantasy, cost = 13.0$}
Media{id = 2, title = DVD Title 1, category = Action, cost = 15.99$}
Media{id = 3, title = Book Title 1, category = Fiction, cost = 29.99$}
```

Figure 9.3: Result demo Polymorphism

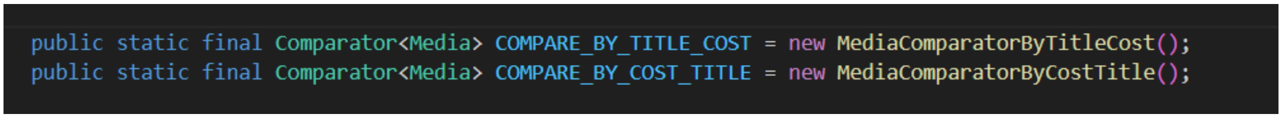
Lớp Media là lớp cơ sở được kế thừa bởi các lớp cụ thể hơn là CompactDisc, DigitalVideoDisc và Book. Khi khởi tạo các đối tượng cd, dvd, book thuộc lớp con rồi gán chúng cho biến kiểu Media, ta áp dụng kỹ thuật gọi là upcasting.

Việc thêm chúng vào danh sách media và duyệt danh sách để in ra thông tin mỗi phần tử bằng phương thức toString() là ví dụ điển hình cho tính đa hình động. Mỗi lớp con có thể cài đặt riêng toString() nên kết quả sẽ khác nhau dựa theo loại đối tượng, mà không cần quan tâm đến kiểu cụ thể của từng phần tử.

10 Sort media in the car

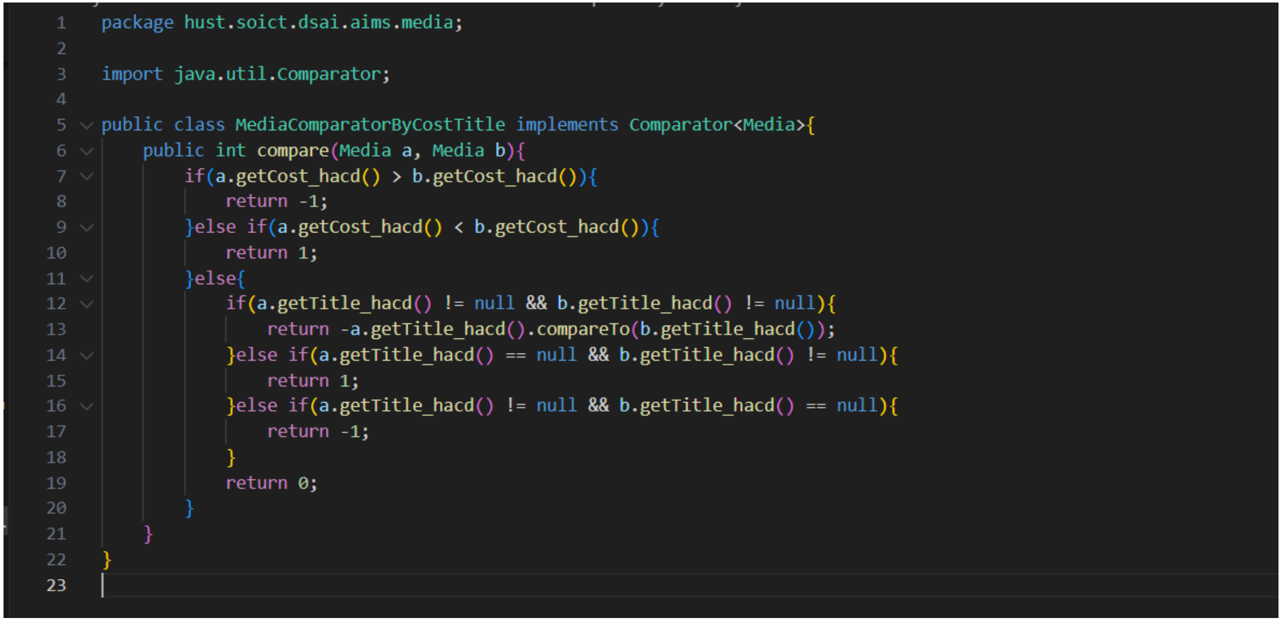
Sắp xếp các media trong giỏ hàng theo hai tiêu chí:

- Bằng title: Hiển thị tất cả các media theo thứ tự bảng chữ cái. Trong trường hợp cùng title, media có cost cao hơn sẽ được hiển thị trước.
- Bằng cost: Hiển thị theo thứ tự cost giảm dần. Trong trường hợp cost như nhau, sắp xếp media theo thứ tự bảng chữ cái



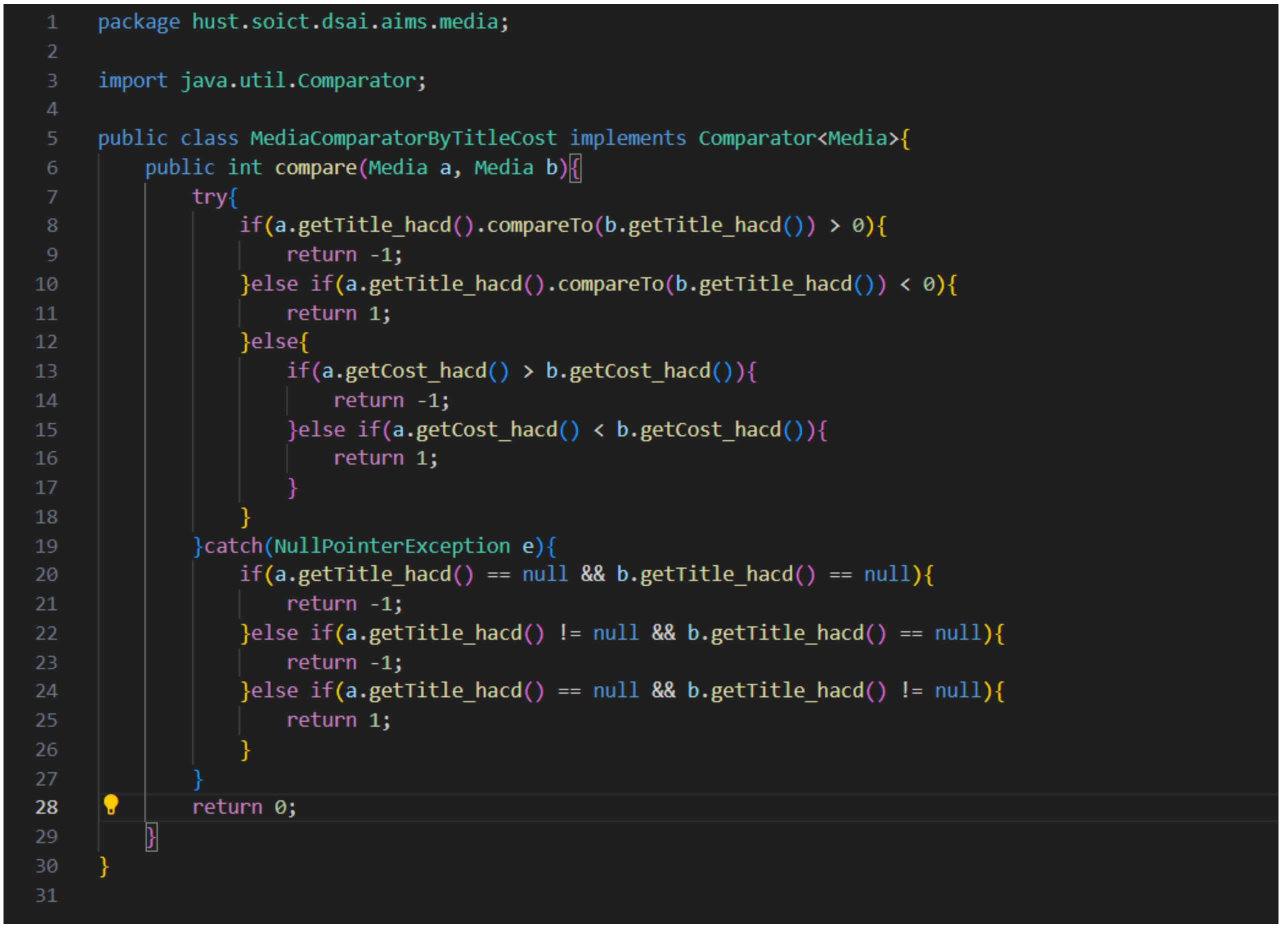
```java
public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost();
public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle();
```

Figure 10.1: Add the comparators as attributes of the Media class



```java
package hust.soict.dsai.aims.media;

import java.util.Comparator;

public class MediaComparatorByCostTitle implements Comparator<Media>{
    public int compare(Media a, Media b){
        if(a.getCost_hacd() > b.getCost_hacd()){
            return -1;
        }else if(a.getCost_hacd() < b.getCost_hacd()){
            return 1;
        }else{
            if(a.getTitle_hacd() != null && b.getTitle_hacd() != null){
                return -a.getTitle_hacd().compareTo(b.getTitle_hacd());
            }else if(a.getTitle_hacd() == null && b.getTitle_hacd() != null){
                return 1;
            }else if(a.getTitle_hacd() != null && b.getTitle_hacd() == null){
                return -1;
            }
            return 0;
        }
    }
}

```

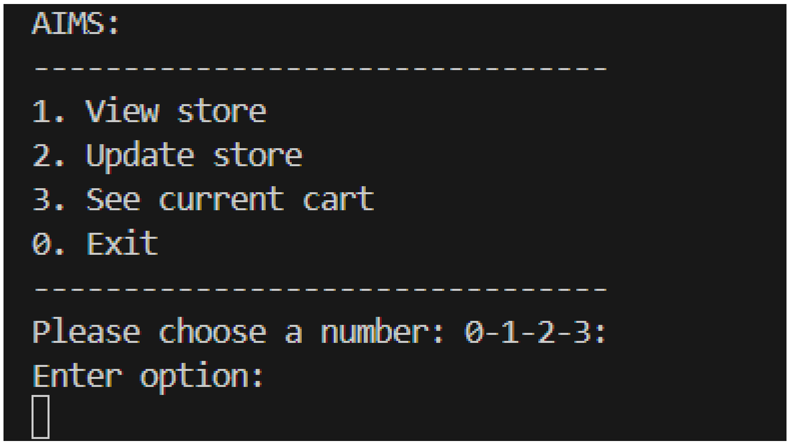
Figure 10.2: MediaComparatorByCostTitle Class



```java
package hust.soict.dsai.aims.media;

import java.util.Comparator;

public class MediaComparatorByTitleCost implements Comparator<Media>{
    public int compare(Media a, Media b){
        try{
            if(a.getTitle_hacd().compareTo(b.getTitle_hacd()) > 0){
                return -1;
            }else if(a.getTitle_hacd().compareTo(b.getTitle_hacd()) < 0){
                return 1;
            }else{
                if(a.getCost_hacd() > b.getCost_hacd()){
                    return -1;
                }else if(a.getCost_hacd() < b.getCost_hacd()){
                    return 1;
                }
            }
        }catch(NullPointerException e){
            if(a.getTitle_hacd() == null && b.getTitle_hacd() == null){
                return -1;
            }else if(a.getTitle_hacd() != null && b.getTitle_hacd() == null){
                return -1;
            }else if(a.getTitle_hacd() == null && b.getTitle_hacd() != null){
                return 1;
            }
        }
        return 0;
    }
}
```

Figure 10.3: MediaComparatorByTitleCost Class

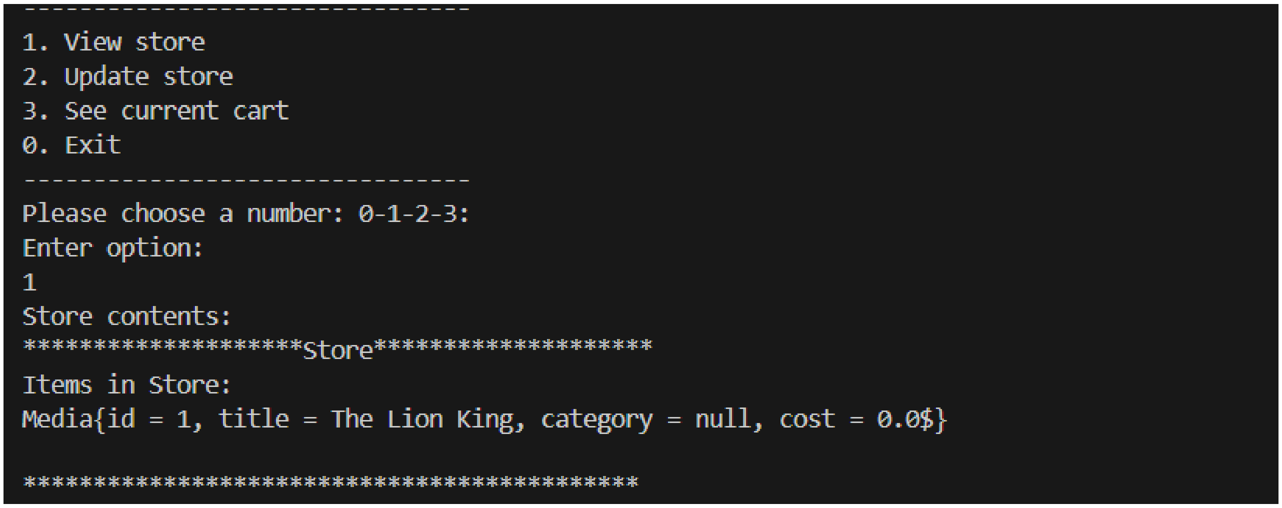
11 Create a complete console application in the Aims class



```
AIMS:
--------------------------------
1. View store
2. Update store
3. See current cart
0. Exit
--------------------------------
Please choose a number: 0-1-2-3:
Enter option:
```

Figure 11.1: Màn hình chính

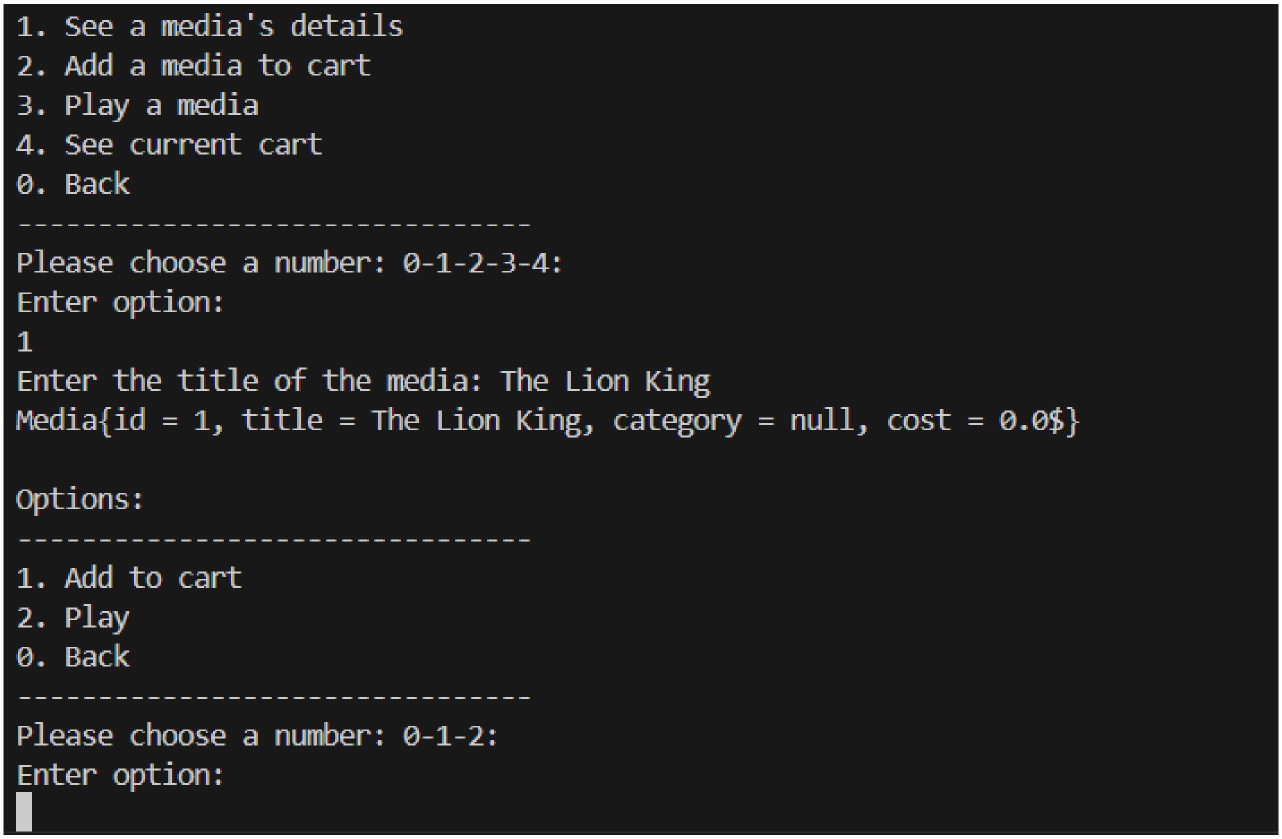
11.1 Người dùng chọn 1: View store



```
--------------------------------
1. View store
2. Update store
3. See current cart
0. Exit
--------------------------------
Please choose a number: 0-1-2-3:
Enter option:
1
Store contents:
********************Store********************
Items in Store:
Media{id = 1, title = The Lion King, category = null, cost = 0.0$}

********************************************
```

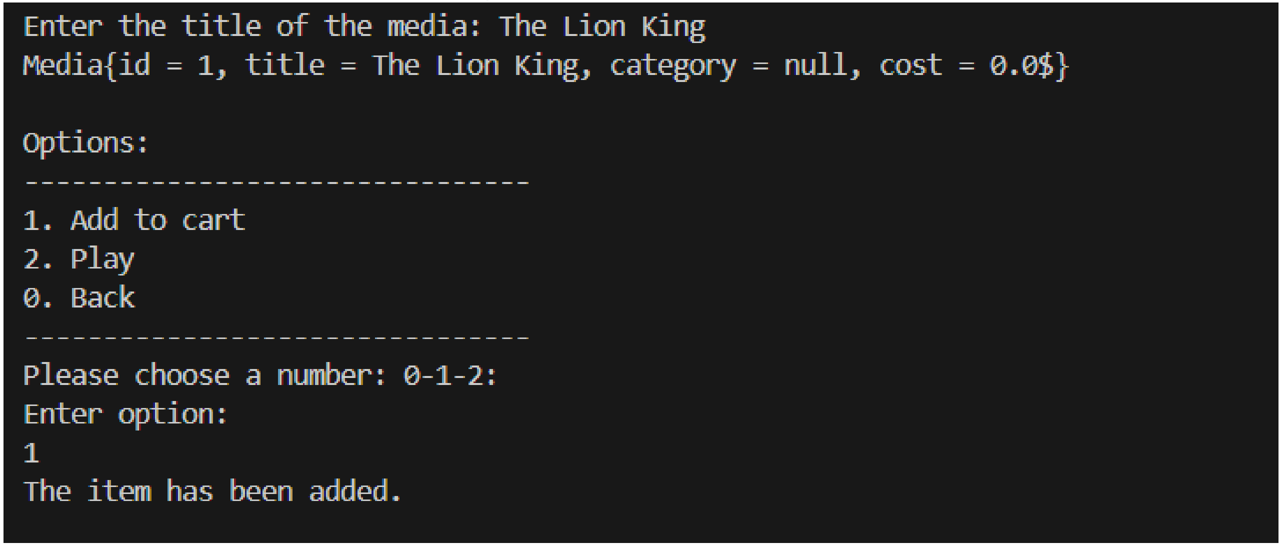
Figure 11.2: Vào Trang View Store

11.1.1 Người dùng tiếp tục chọn 1. See a media’s details



```
1. See a media's details
2. Add a media to cart
3. Play a media
4. See current cart
0. Back
--------------------------------
Please choose a number: 0-1-2-3-4:
Enter option:
1
Enter the title of the media: The Lion King
Media{id = 1, title = The Lion King, category = null, cost = 0.0$}

Options:
--------------------------------
1. Add to cart
2. Play
0. Back
--------------------------------
Please choose a number: 0-1-2:
Enter option:
```

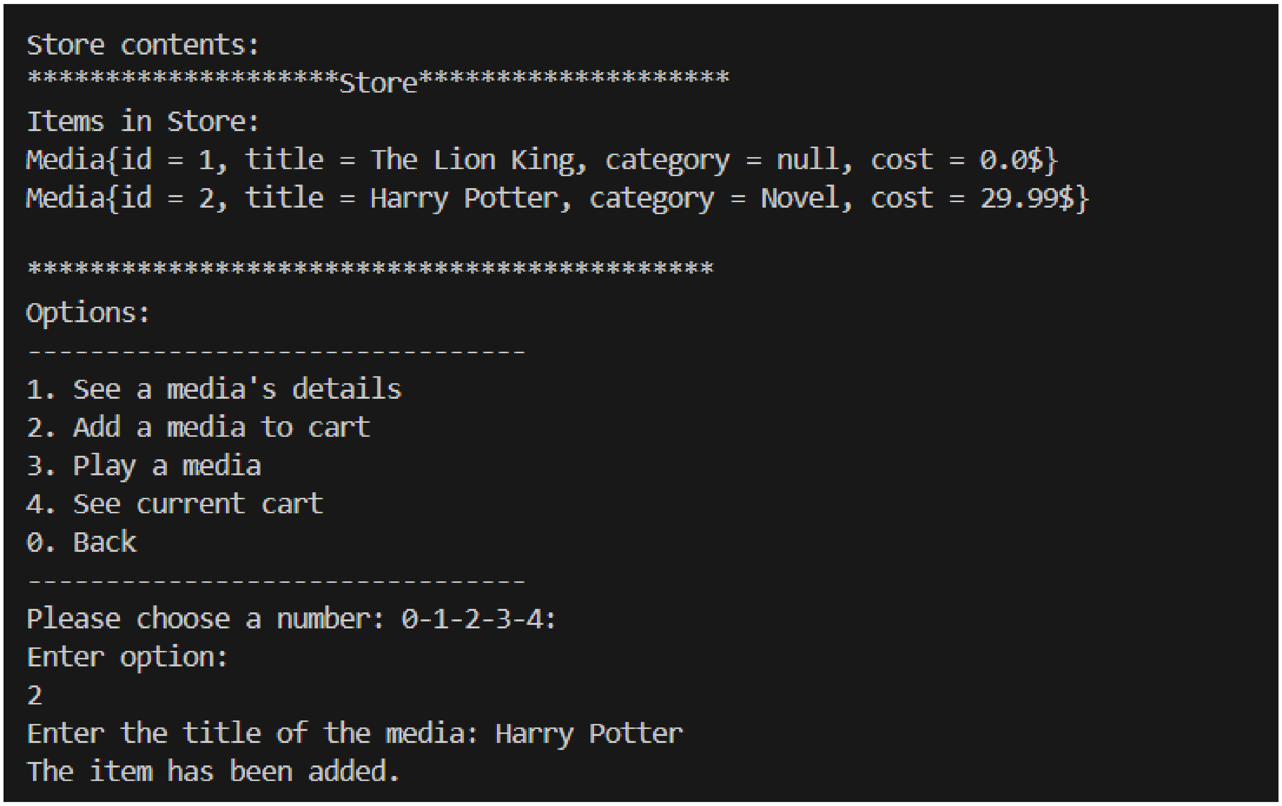
Figure 11.3: See a media's details



```
Enter the title of the media: The Lion King
Media{id = 1, title = The Lion King, category = null, cost = 0.0$}

Options:
--------------------------------
1. Add to cart
2. Play
0. Back
--------------------------------
Please choose a number: 0-1-2:
Enter option:
1
The item has been added.
```

Figure 11.4: Thêm vào Cart

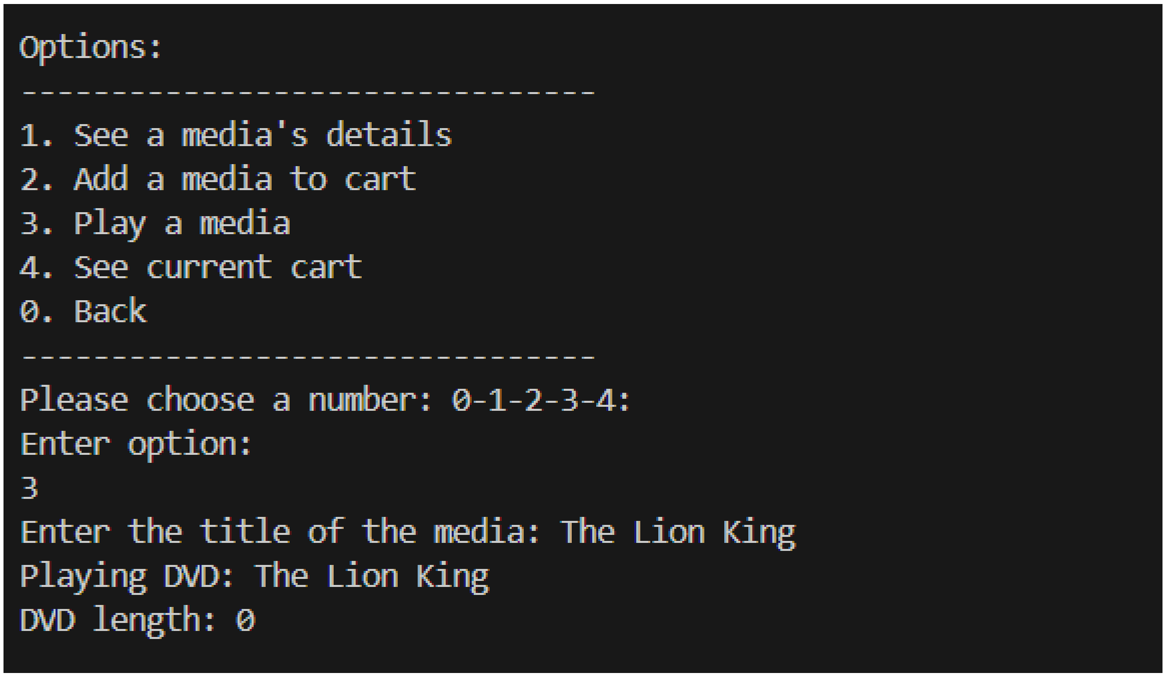
11.1.2 Người dùng chọn 2: Add a media to the cart



```
Store contents:
********************Store********************
Items in Store:
Media{id = 1, title = The Lion King, category = null, cost = 0.0$}
Media{id = 2, title = Harry Potter, category = Novel, cost = 29.99$}

********************************************
Options:
--------------------------------
1. See a media's details
2. Add a media to cart
3. Play a media
4. See current cart
0. Back
--------------------------------
Please choose a number: 0-1-2-3-4:
Enter option:
2
Enter the title of the media: Harry Potter
The item has been added.
```

Figure 11.5: Thêm media vào Cart

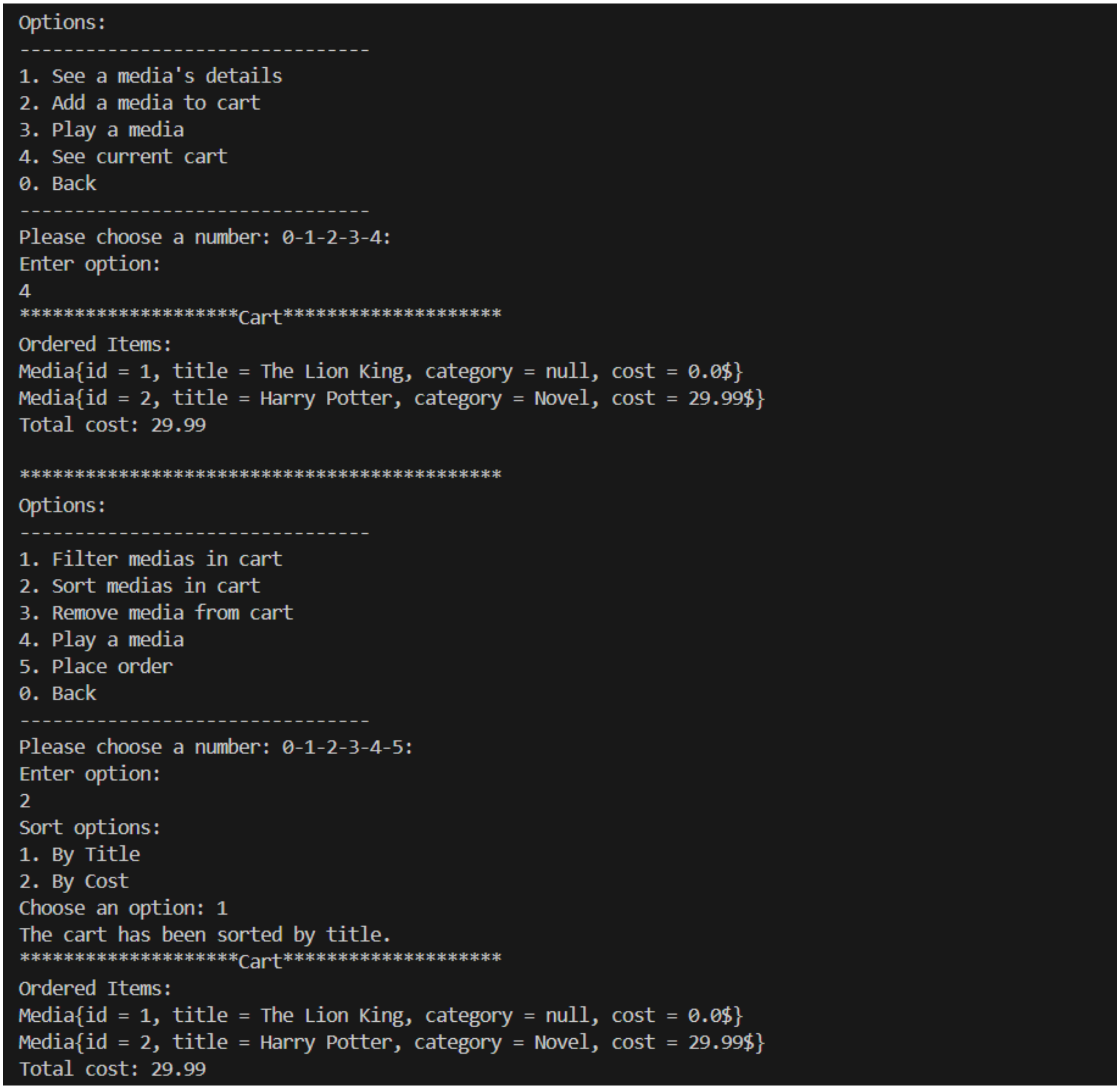
11.1.3 Người dùng chọn 3: Play a media



```
Options:
--------------------------------
1. See a media's details
2. Add a media to cart
3. Play a media
4. See current cart
0. Back
--------------------------------
Please choose a number: 0-1-2-3-4:
Enter option:
3
Enter the title of the media: The Lion King
Playing DVD: The Lion King
DVD length: 0
```

Figure 11.6: Play a media

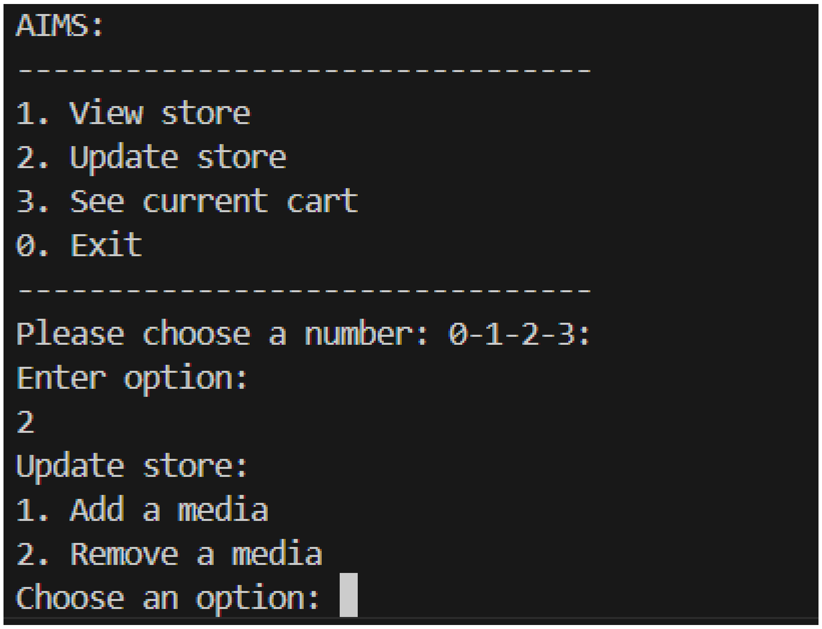
11.1.4 Người dùng chọn 4: See current cart



```
Options:
--------------------------------
1. See a media's details
2. Add a media to cart
3. Play a media
4. See current cart
0. Back
--------------------------------
Please choose a number: 0-1-2-3-4:
Enter option:
4
********************Cart********************
Ordered Items:
Media{id = 1, title = The Lion King, category = null, cost = 0.0$}
Media{id = 2, title = Harry Potter, category = Novel, cost = 29.99$}
Total cost: 29.99

********************************************
Options:
--------------------------------
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
--------------------------------
Please choose a number: 0-1-2-3-4-5:
Enter option:
2
Sort options:
1. By Title
2. By Cost
Choose an option: 1
The cart has been sorted by title.
********************Cart********************
Ordered Items:
Media{id = 1, title = The Lion King, category = null, cost = 0.0$}
Media{id = 2, title = Harry Potter, category = Novel, cost = 29.99$}
Total cost: 29.99
```

Figure 11.7: See current cart after sort

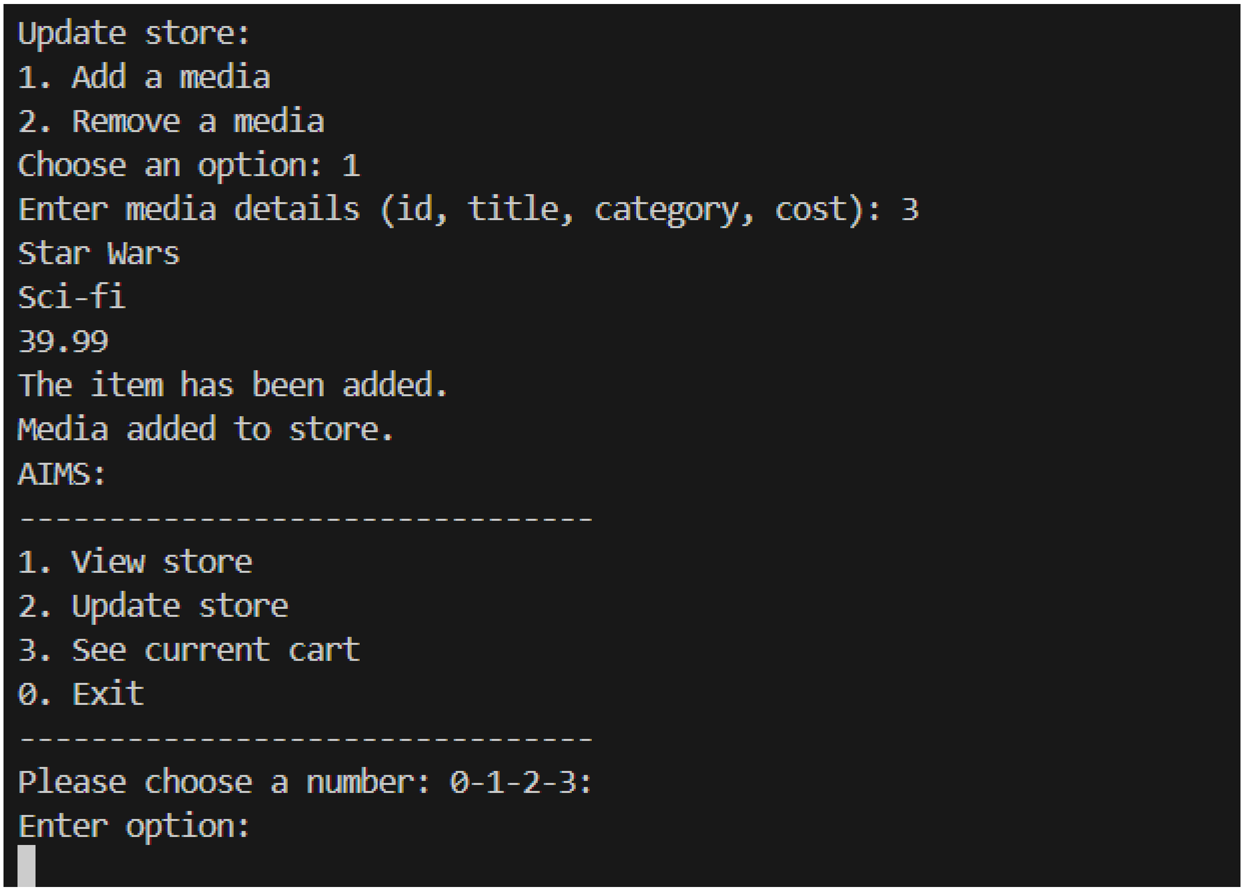
11.2 Người dùng chọn 2: Update store



```
AIMS:
--------------------------------
1. View store
2. Update store
3. See current cart
0. Exit
--------------------------------
Please choose a number: 0-1-2-3:
Enter option:
2
Update store:
1. Add a media
2. Remove a media
Choose an option:
```

Figure 11.8: Vào Trang Update Store

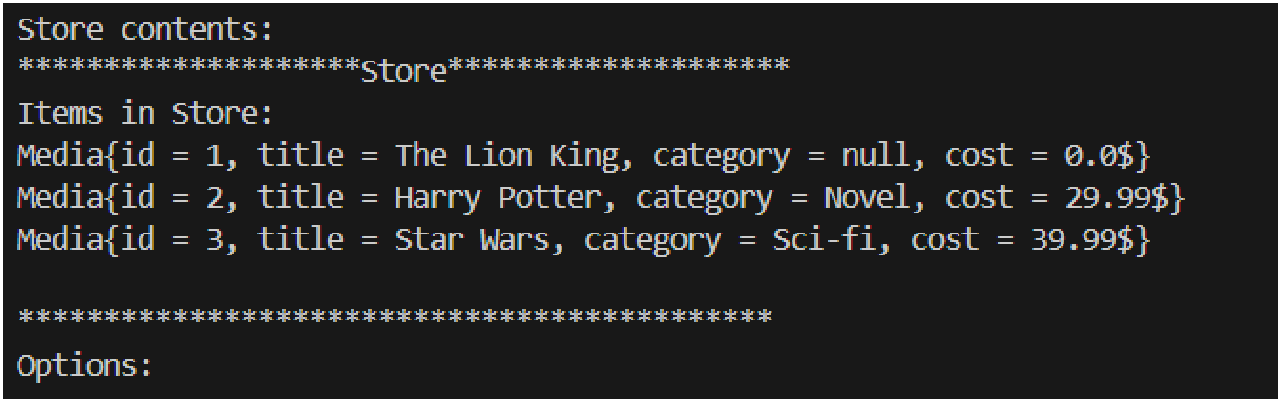
11.2.1 Người dùng chọn 1: Add a media to the store



```
Update store:
1. Add a media
2. Remove a media
Choose an option: 1
Enter media details (id, title, category, cost): 3
Star Wars
Sci-fi
39.99
The item has been added.
Media added to store.
AIMS:
--------------------------------
1. View store
2. Update store
3. See current cart
0. Exit
--------------------------------
Please choose a number: 0-1-2-3:
Enter option:
```

Figure 11.9: Add a media to store

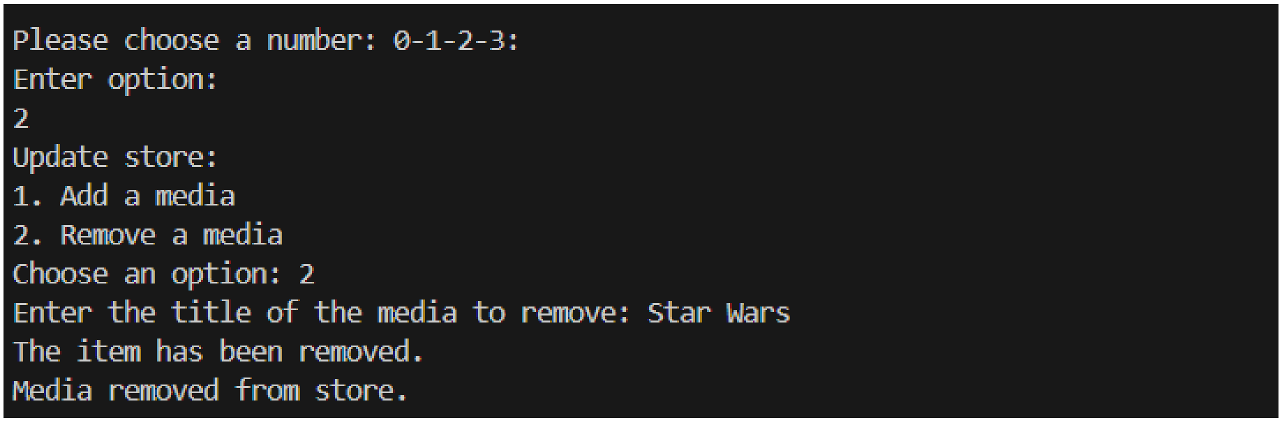
=> Kết quả sau khi thêm



```
Store contents:
********************Store********************
Items in Store:
Media{id = 1, title = The Lion King, category = null, cost = 0.0$}
Media{id = 2, title = Harry Potter, category = Novel, cost = 29.99$}
Media{id = 3, title = Star Wars, category = Sci-fi, cost = 39.99$}

********************************************
Options:
```

Figure 11.10: Result after add media to store

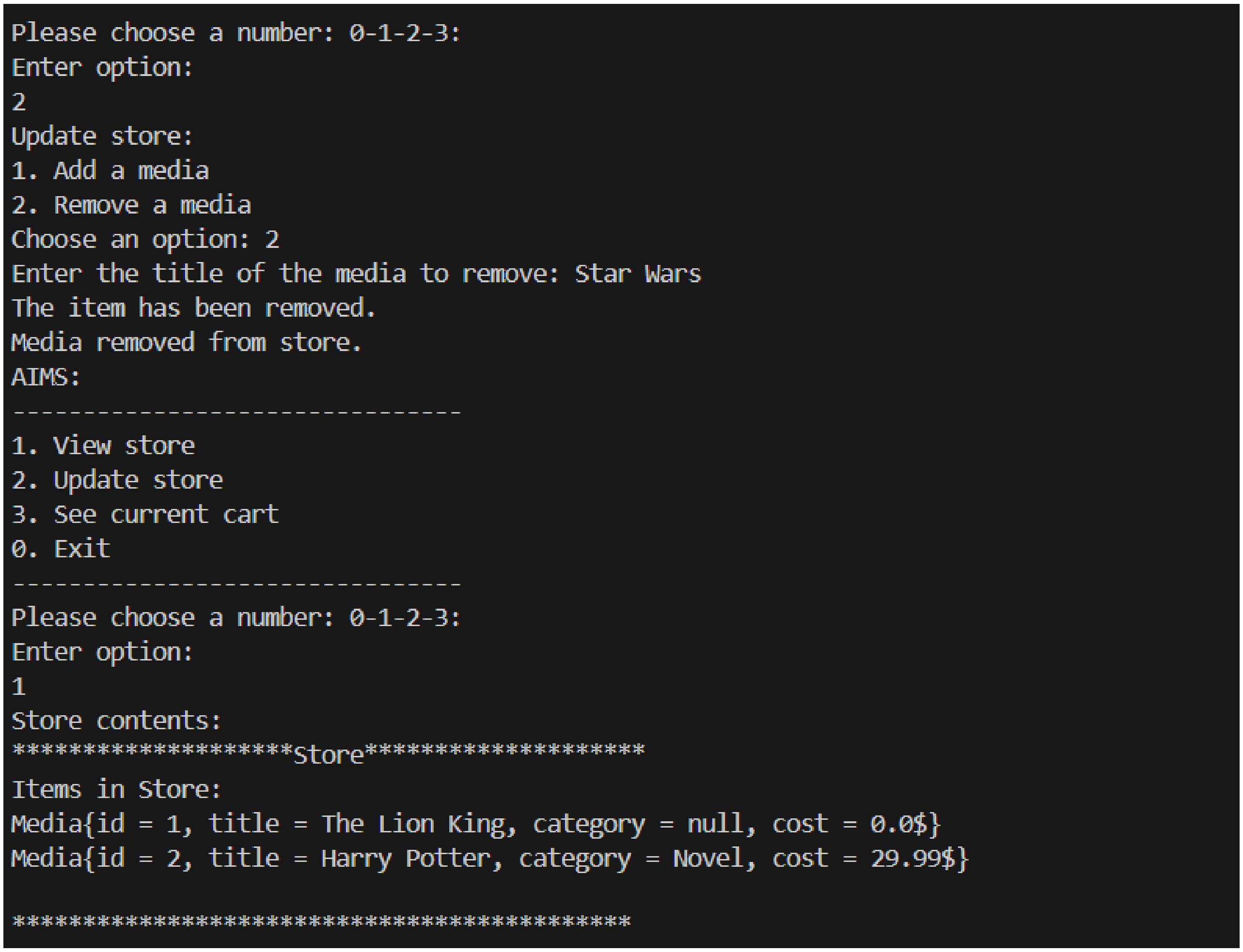
11.2.2 Người dùng chọn 2: Remove a media from the store



```
Please choose a number: 0-1-2-3:
Enter option:
2
Update store:
1. Add a media
2. Remove a media
Choose an option: 2
Enter the title of the media to remove: Star Wars
The item has been removed.
Media removed from store.
```

Figure 11.11: Remove a media from the store

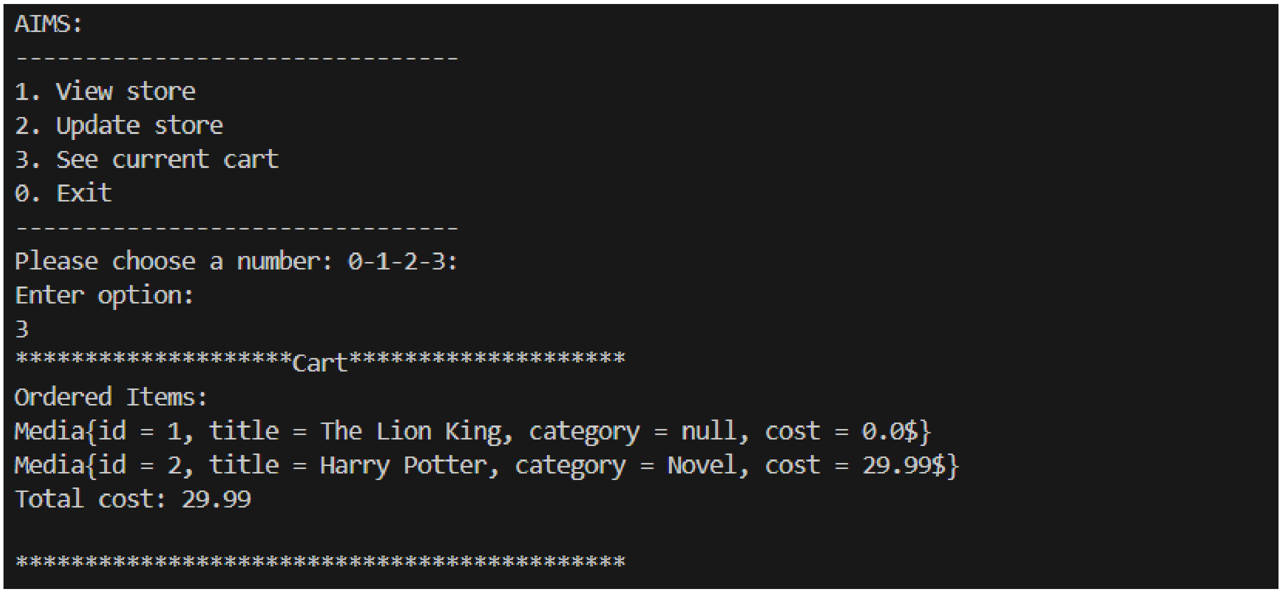
=> Kết quả sau khi remove



```
Please choose a number: 0-1-2-3:
Enter option:
2
Update store:
1. Add a media
2. Remove a media
Choose an option: 2
Enter the title of the media to remove: Star Wars
The item has been removed.
Media removed from store.
AIMS:
--------------------------------
1. View store
2. Update store
3. See current cart
0. Exit
--------------------------------
Please choose a number: 0-1-2-3:
Enter option:
1
Store contents:
********************Store********************
Items in Store:
Media{id = 1, title = The Lion King, category = null, cost = 0.0$}
Media{id = 2, title = Harry Potter, category = Novel, cost = 29.99$}

********************************************
```

Figure 11.12: Result after remove a media

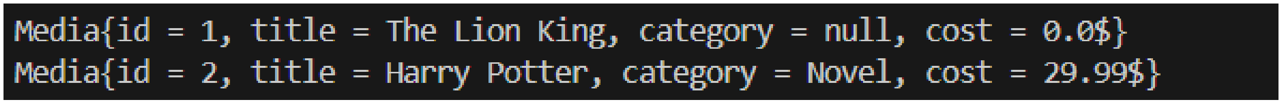
11.3 Người dùng chọn 3: See current cart



```
AIMS:
--------------------------------
1. View store
2. Update store
3. See current cart
0. Exit
--------------------------------
Please choose a number: 0-1-2-3:
Enter option:
3
********************Cart********************
Ordered Items:
Media{id = 1, title = The Lion King, category = null, cost = 0.0$}
Media{id = 2, title = Harry Potter, category = Novel, cost = 29.99$}
Total cost: 29.99

********************************************
```

Figure 11.13: Vào trang See current cart

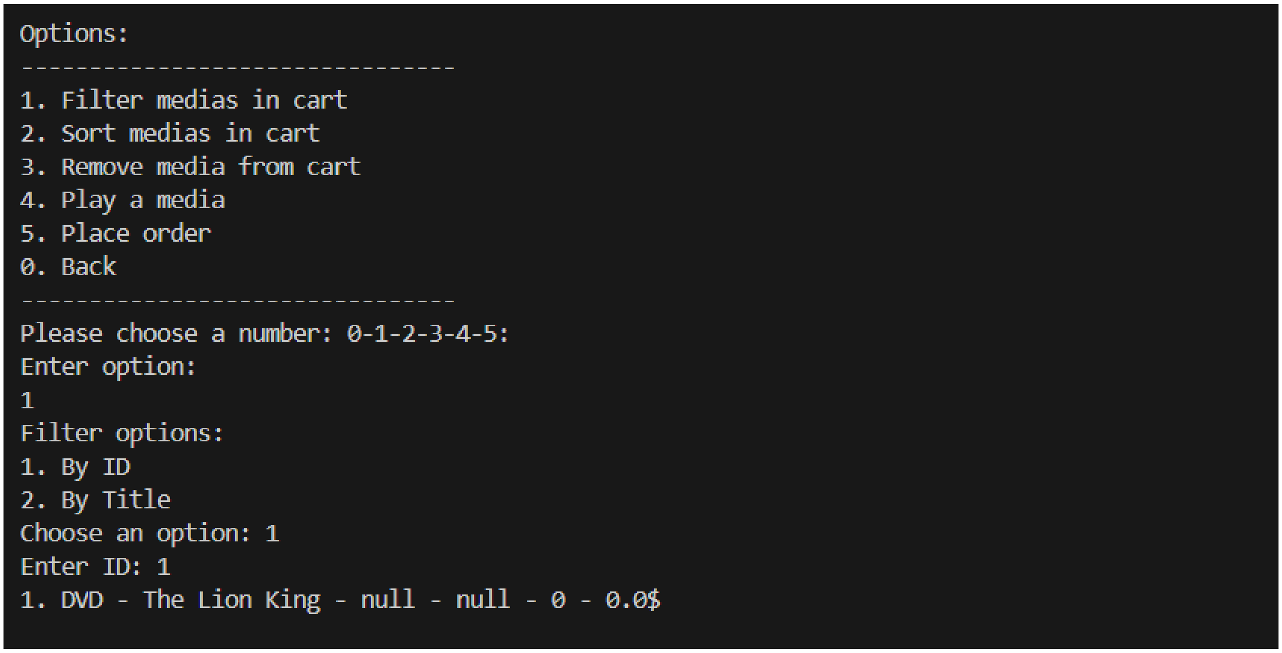
Giả sử lúc này trong Cart sẽ có các Media sau



```
Media{id = 1, title = The Lion King, category = null, cost = 0.0$}
Media{id = 2, title = Harry Potter, category = Novel, cost = 29.99$}
```

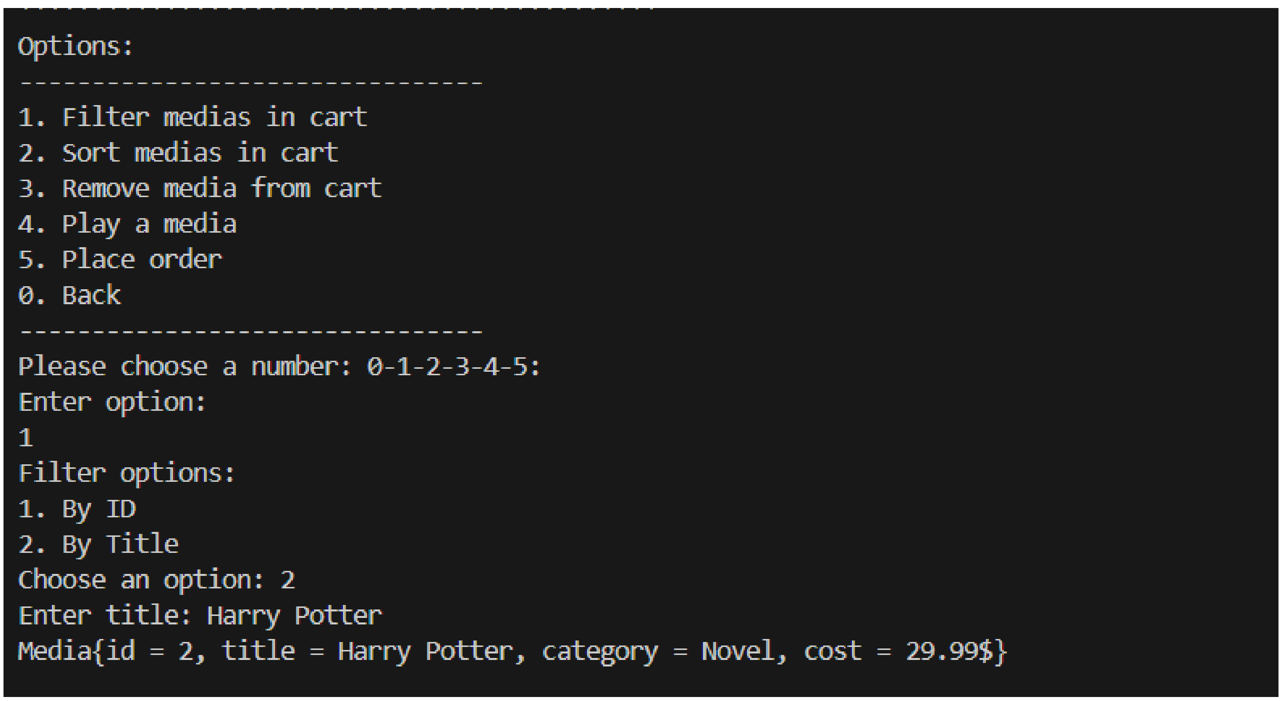
Figure 11.14: Media in Cart

11.3.1 Người dùng chọn 1: Filter medias in cart



```
Options:
--------------------------------
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
--------------------------------
Please choose a number: 0-1-2-3-4-5:
Enter option:
1
Filter options:
1. By ID
2. By Title
Choose an option: 1
Enter ID: 1
1. DVD - The Lion King - null - null - 0 - 0.0$
```

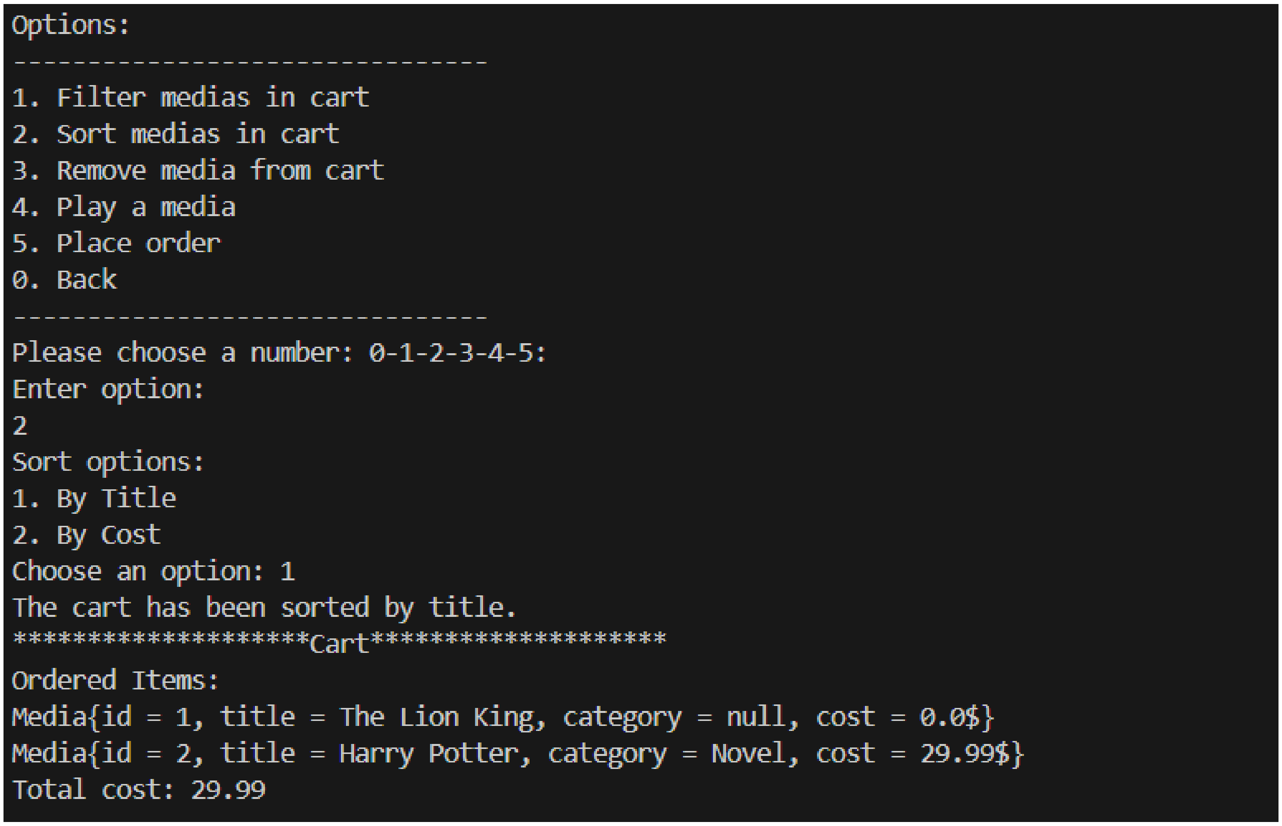
Figure 11.15: Filter Cart By id



```
--------------------------------------------
Options:
--------------------------------
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
--------------------------------
Please choose a number: 0-1-2-3-4-5:
Enter option:
1
Filter options:
1. By ID
2. By Title
Choose an option: 2
Enter title: Harry Potter
Media{id = 2, title = Harry Potter, category = Novel, cost = 29.99$}
```

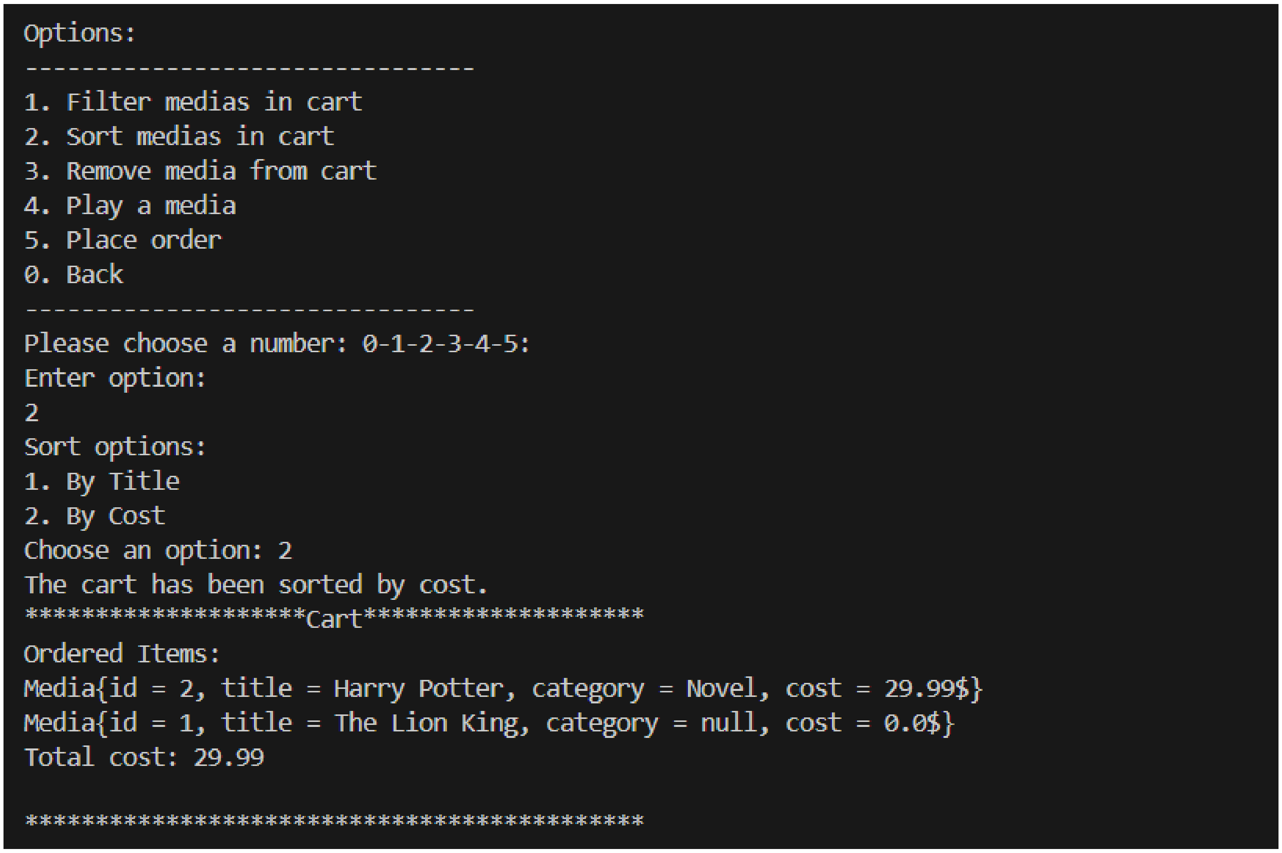
Figure 11.16: Filter Cart By Title

11.3.2 Người dùng chọn 2: Sort medias in cart



```
Options:
--------------------------------
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
--------------------------------
Please choose a number: 0-1-2-3-4-5:
Enter option:
2
Sort options:
1. By Title
2. By Cost
Choose an option: 1
The cart has been sorted by title.
********************Cart********************
Ordered Items:
Media{id = 1, title = The Lion King, category = null, cost = 0.0$}
Media{id = 2, title = Harry Potter, category = Novel, cost = 29.99$}
Total cost: 29.99
```

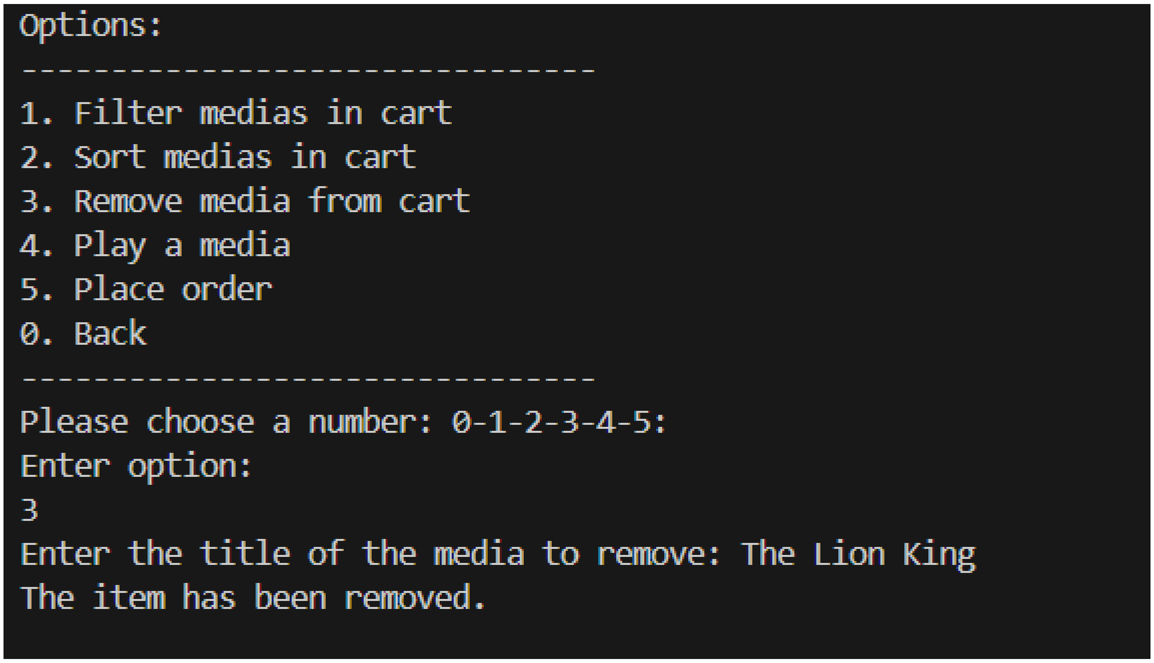
Figure 11.17: Sort Cart By Title



```
Options:
--------------------------------
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
--------------------------------
Please choose a number: 0-1-2-3-4-5:
Enter option:
2
Sort options:
1. By Title
2. By Cost
Choose an option: 2
The cart has been sorted by cost.
********************Cart********************
Ordered Items:
Media{id = 2, title = Harry Potter, category = Novel, cost = 29.99$}
Media{id = 1, title = The Lion King, category = null, cost = 0.0$}
Total cost: 29.99

********************************************
```

Figure 11.18: Sort Cart By Cost

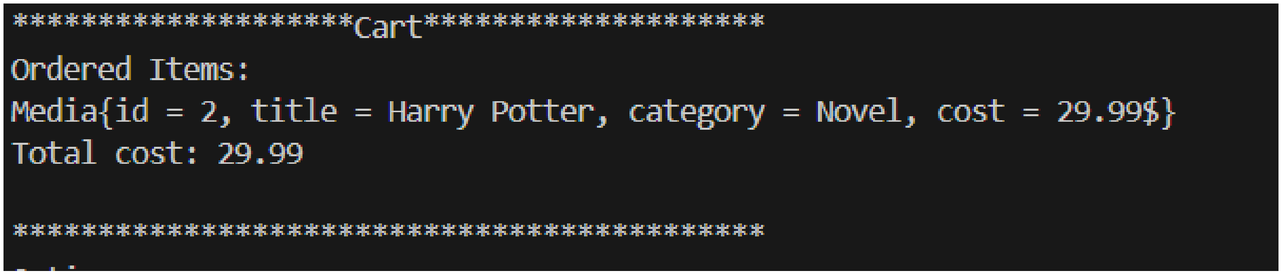
11.3.3 Người dùng chọn 3: Remove media from cart



```
Options:
--------------------------------
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
--------------------------------
Please choose a number: 0-1-2-3-4-5:
Enter option:
3
Enter the title of the media to remove: The Lion King
The item has been removed.
```

Figure 11.19: Remove media by id

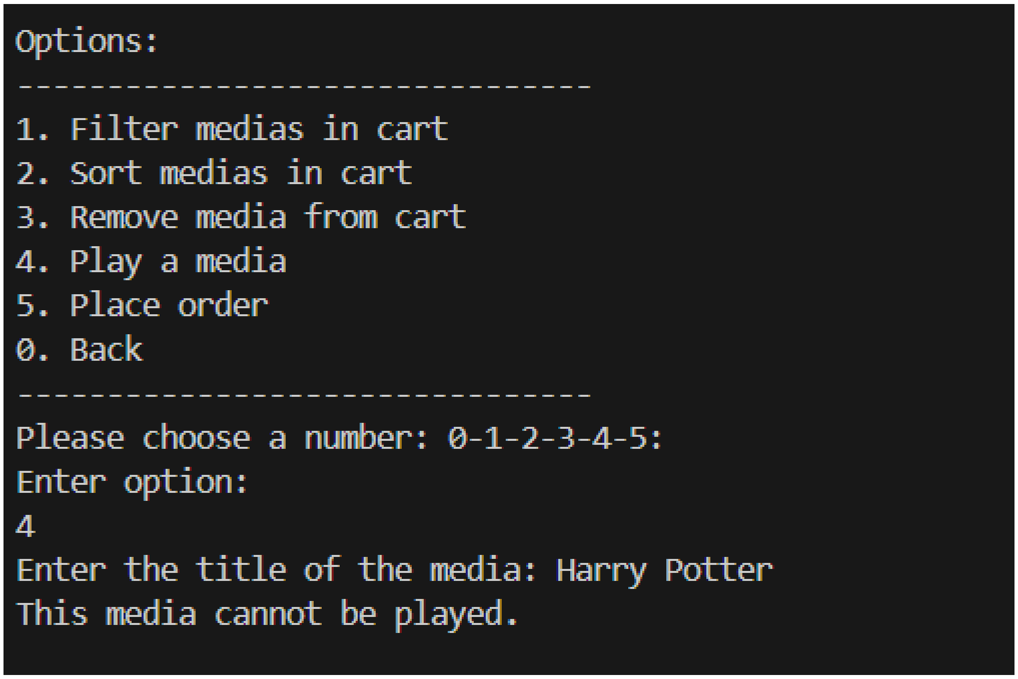
=> Kết quả



```
********************Cart********************
Ordered Items:
Media{id = 2, title = Harry Potter, category = Novel, cost = 29.99$}
Total cost: 29.99

********************************************
```

Figure 11.20: Result after remove media in cart by id

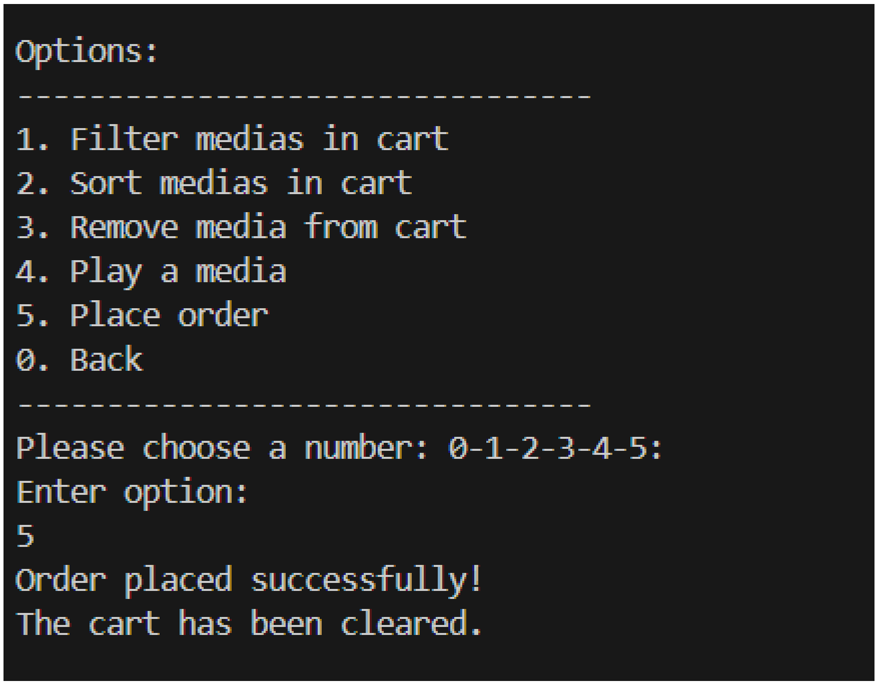
11.3.4 Người dùng chọn 4: Play a media



```
Options:
--------------------------------
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
--------------------------------
Please choose a number: 0-1-2-3-4-5:
Enter option:
4
Enter the title of the media: Harry Potter
This media cannot be played.
```

Figure 11.21: Play a media in cart

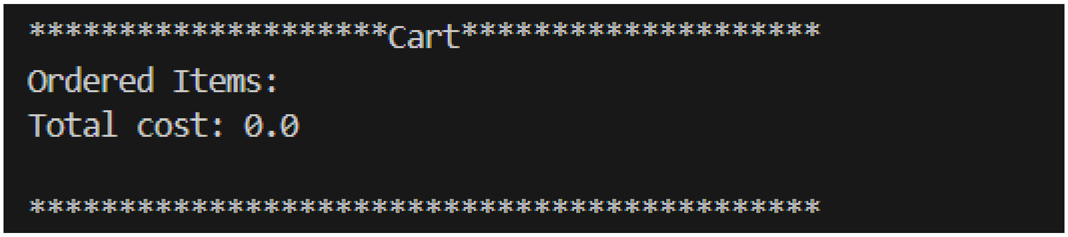
11.3.5 Người dùng chọn 5: Place order



```
Options:
--------------------------------
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
--------------------------------
Please choose a number: 0-1-2-3-4-5:
Enter option:
5
Order placed successfully!
The cart has been cleared.
```

Figure 11.22: Order

=> Kết quả sau khi order



```
********************Cart********************
Ordered Items:
Total cost: 0.0

********************************************
```

Figure 11.23: Result after order

12 Class Diagram

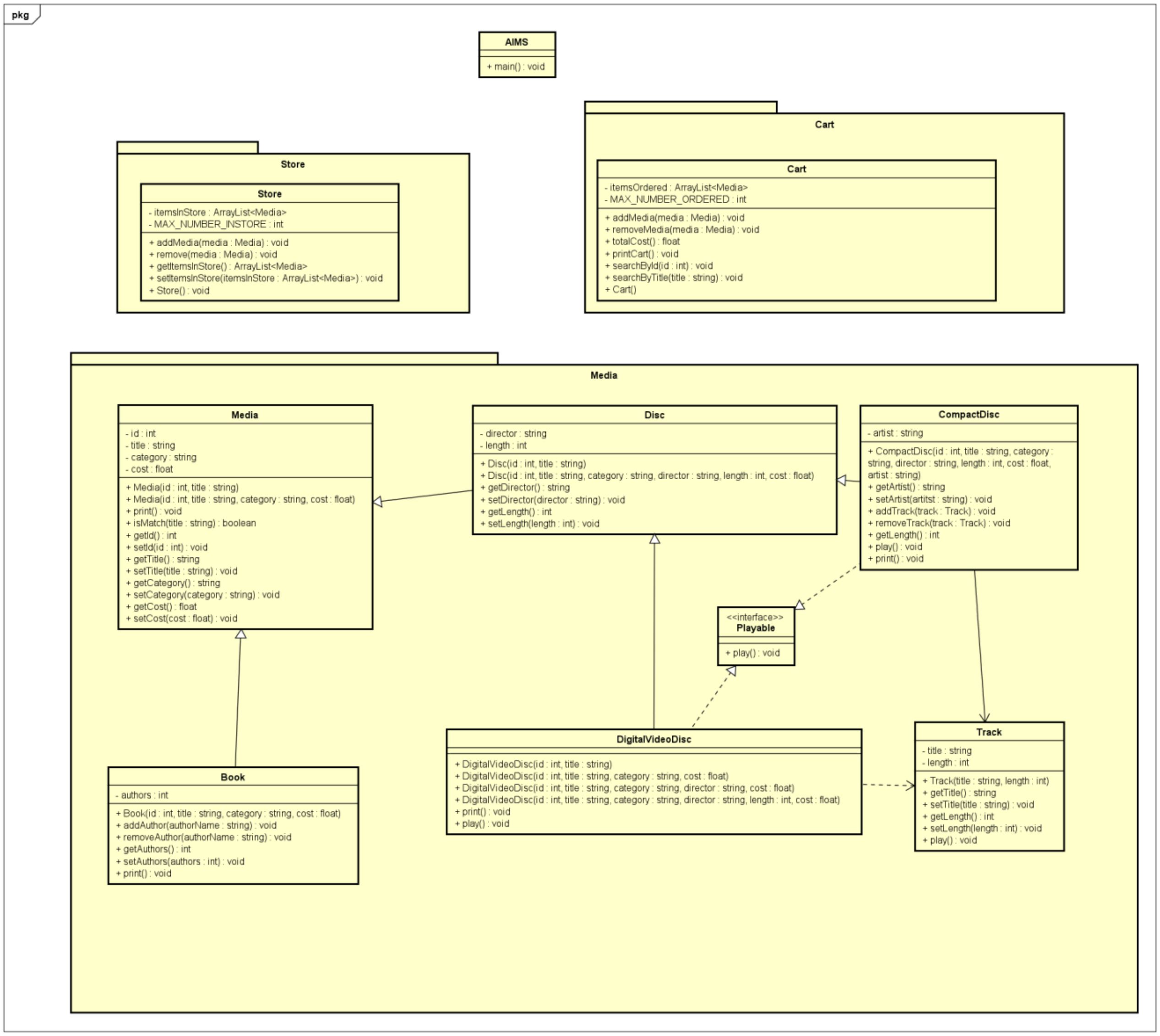

pkg
AIMS
+ main() : void
Store
Store
- itemsInStore : ArrayList<Media>
- MAX_NUMBER_INSTORE : int
+ addMedia(media : Media) : void
+ remove(media : Media) : void
+ getItemsInStore() : ArrayList<Media>
+ setItemsInStore(itemsInStore : ArrayList<Media>) : void
+ Store() : void
Cart
Cart
- itemsOrdered : ArrayList<Media>
- MAX_NUMBER_ORDERED : int
+ addMedia(media : Media) : void
+ removeMedia(media : Media) : void
+ totalCost() : float
+ printCart() : void
+ searchById(id : int) : void
+ searchByTitle(title : string) : void
+ Cart()
Media
Media
- id : int
- title : string
- category : string
- cost : float
+ Media(id : int, title : string)
+ Media(id : int, title : string, category : string, cost : float)
+ print() : void
+ isMatch(title : string) : boolean
+ getId() : int
+ setId(id : int) : void
+ getTitle() : string
+ setTitle(title : string) : void
+ getCategory() : string
+ setCategory(category : string) : void
+ getCost() : float
+ setCost(cost : float) : void
Disc
- director : string
- length : int
+ Disc(id : int, title : string)
+ Disc(id : int, title : string, category : string, director : string, length : int, cost : float)
+ getDirector() : string
+ setDirector(director : string) : void
+ getLength() : int
+ setLength(length : int) : void
CompactDisc
- artist : string
+ CompactDisc(id : int, title : string, category : string, director : string, length : int, cost : float, artist : string)
+ getArtist() : string
+ setArtist(artitst : string) : void
+ addTrack(track : Track) : void
+ removeTrack(track : Track) : void
+ getLength() : int
+ play() : void
+ print() : void
<<interface>>
Playable
+ play() : void
Book
- authors : int
+ Book(id : int, title : string, category : string, cost : float)
+ addAuthor(authorName : string) : void
+ removeAuthor(authorName : string) : void
+ getAuthors() : int
+ setAuthors(authors : int) : void
+ print() : void
DigitalVideoDisc
+ DigitalVideoDisc(id : int, title : string)
+ DigitalVideoDisc(id : int, title : string, category : string, cost : float)
+ DigitalVideoDisc(id : int, title : string, category : string, director : string, cost : float)
+ DigitalVideoDisc(id : int, title : string, category : string, director : string, length : int, cost : float)
+ print() : void
+ play() : void
Track
- title : string
- length : int
+ Track(title : string, length : int)
+ getTitle() : string
+ setTitle(title : string) : void
+ getLength() : int
+ setLength(length : int) : void
+ play() : void

Figure 12.1: Class Diagram

13 UseCase Diagram

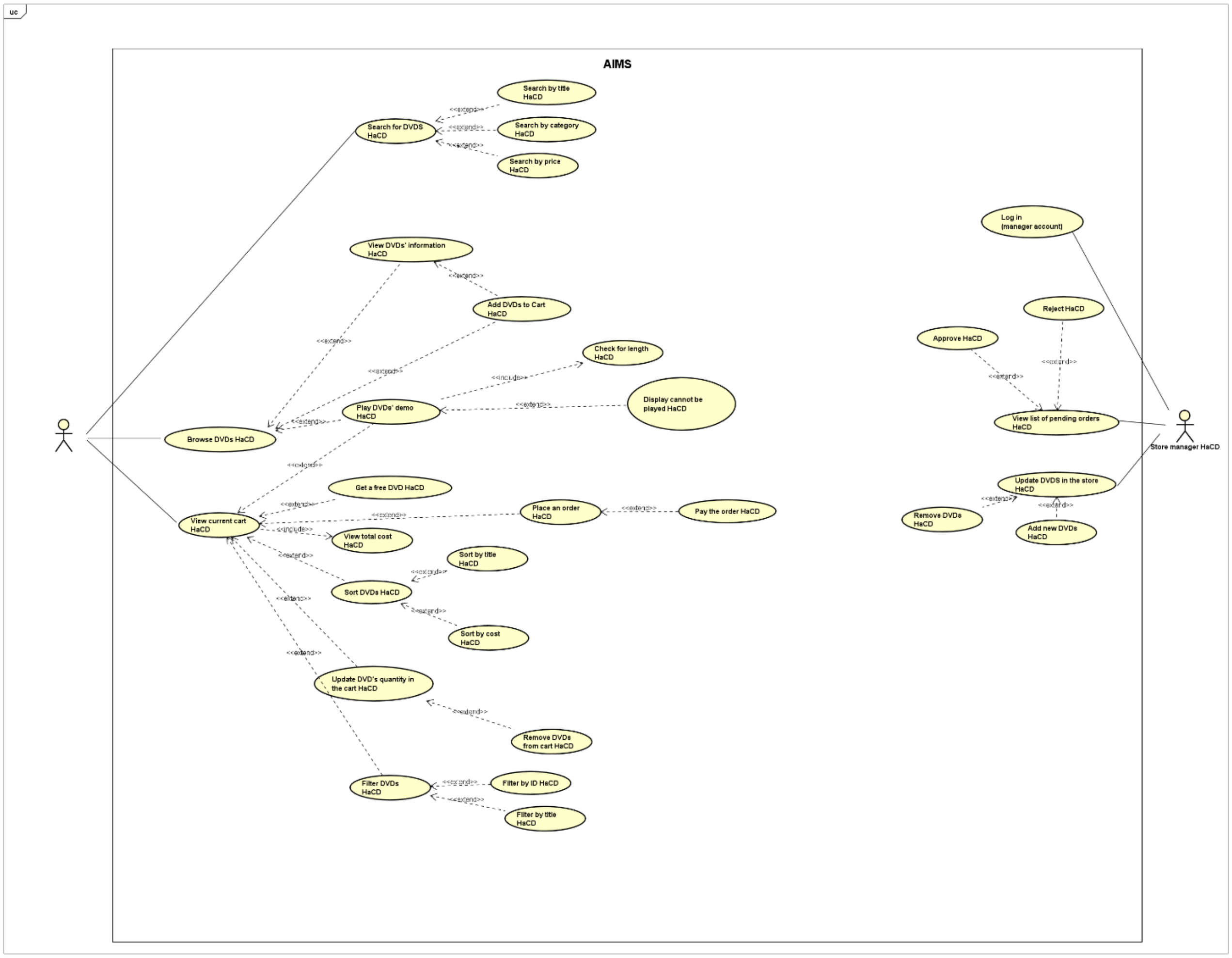
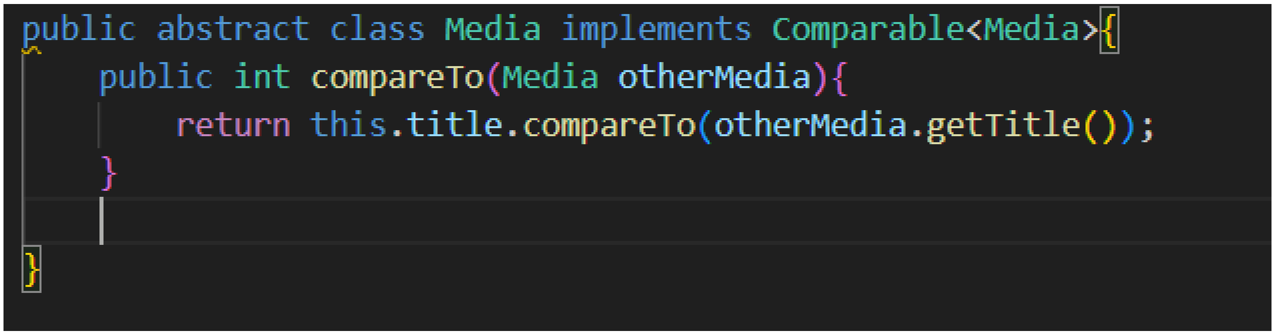


Figure 13.1: UseCase Diagram

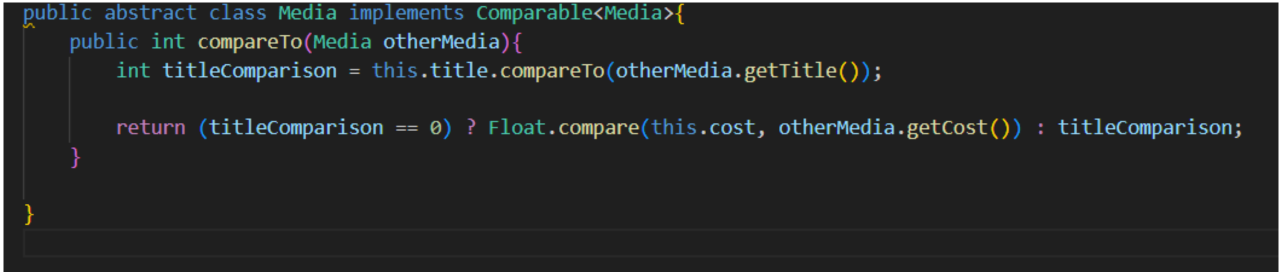
14 Answer Questions

Trong trường hợp muốn so sánh các đối tượng Media với nhau bằng cách sử dụng Comparable thay vì Comparator, thì thay vì tạo ra các lớp riêng cho từng Comparator, chúng ta cần để lớp Media triển khai interface Comparable.



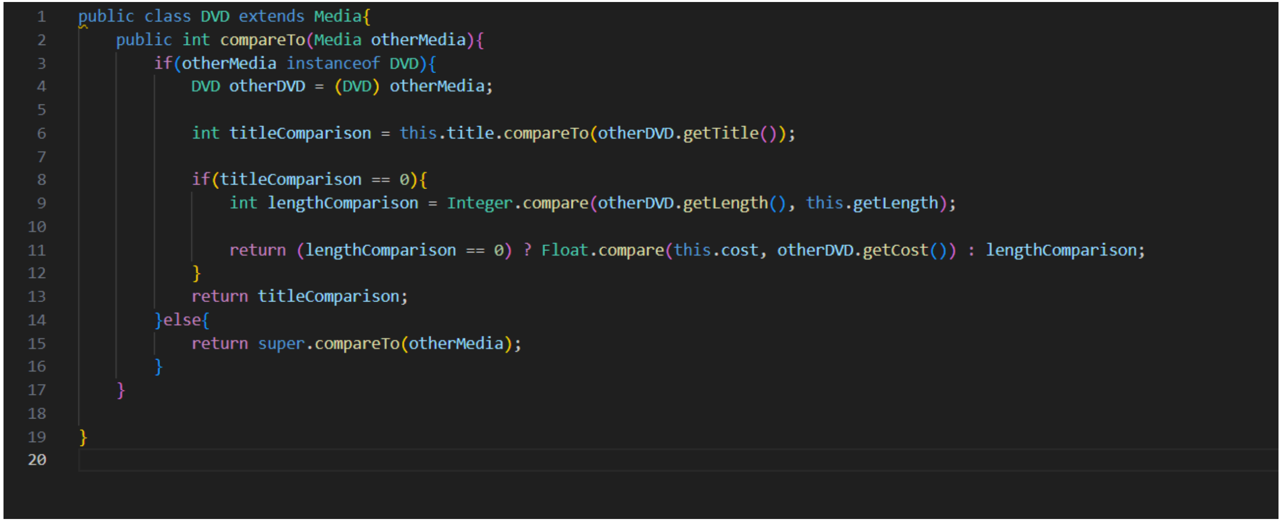
```java
public abstract class Media implements Comparable<Media>{
    public int compareTo(Media otherMedia){
        return this.title.compareTo(otherMedia.getTitle());
    }
    
}
```

Figure 14.1: Triển khai Comparable trong lớp trừu tượng Media



```java
public abstract class Media implements Comparable<Media>{
    public int compareTo(Media otherMedia){
        int titleComparison = this.title.compareTo(otherMedia.getTitle());

        return (titleComparison == 0) ? Float.compare(this.cost, otherMedia.getCost()) : titleComparison;
    }

}
```

Figure 14.2: Mở rộng để so sánh nhiều thuộc tính hơn



```java
public class DVD extends Media{
    public int compareTo(Media otherMedia){
        if(otherMedia instanceof DVD){
            DVD otherDVD = (DVD) otherMedia;

            int titleComparison = this.title.compareTo(otherDVD.getTitle());

            if(titleComparison == 0){
                int lengthComparison = Integer.compare(otherDVD.getLength(), this.getLength);

                return (lengthComparison == 0) ? Float.compare(this.cost, otherDVD.getCost()) : lengthComparison;
            }
            return titleComparison;
        }else{
            return super.compareTo(otherMedia);
        }
    }

}
```

Figure 14.3: Triển khai tại lớp con

Cách triển khai này giúp chúng ta linh hoạt hơn khi so sánh các đối tượng Media và cung cấp khả năng mở rộng cho các lớp con khác nếu cần thiết.