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

BÁO CÁO THỰC HÀNH
IT3103-744528-2024.1

BÀI THỰC HÀNH 04

Họ và tên sv: Nguyễn Hoàng Phúc
Lớp: **K67-CNTT-VN**

GVHD: Lê Thị Hoa

TA: Đặng Mạnh Cường

Hà Nội 12/2024

BÁO CÁO THỰC HÀNH LAB 4 LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

1 Create the Book class 4

2 Creating the abstract Media class 5

3 Creating the CompactDisc class 7

3.1 Create the Disc class extending the Media clas 7

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 10

3.3 Open the CompactDisc class 12

4 Create the Playable interface 13

5 Update the Cart class to work with Media 14

6 Update the Store class to work with Media 16

7 Constructors of whole classes and parent classes 18

8 Unique item in a list 19

9 Polymorphism with toString() method 20

10 Sort media in the car 21

11 Create a complete console application in the Aims class 22

11.1 Người dùng chọn 1: View store 22

11.1.1 Người dùng tiếp tục chọn 1. See a media’s details 23

11.1.2 Người dùng chọn 2: Add a media to the cart 24

11.1.3 Người dùng chọn 3: Play a media 25

11.1.4 Người dùng chọn 4: See current cart 25

11.2 Người dùng chọn 2: Update store 25

11.2.1 Người dùng chọn 1: Add a media to the store 26

11.2.2 Người dùng chọn 2: Remove a media from the store 28

11.3 Người dùng chọn 3: See current cart 29

11.3.1 Người dùng chọn 1: Filter medias in cart 29

11.3.2 Người dùng chọn 2: Sort medias in cart 30

11.3.3 Người dùng chọn 3: Remove media from cart 31

11.3.4 Người dùng chọn 4: Play a media 32

11.3.5 Người dùng chọn 5: Place order 32

12 Class Diagram 34

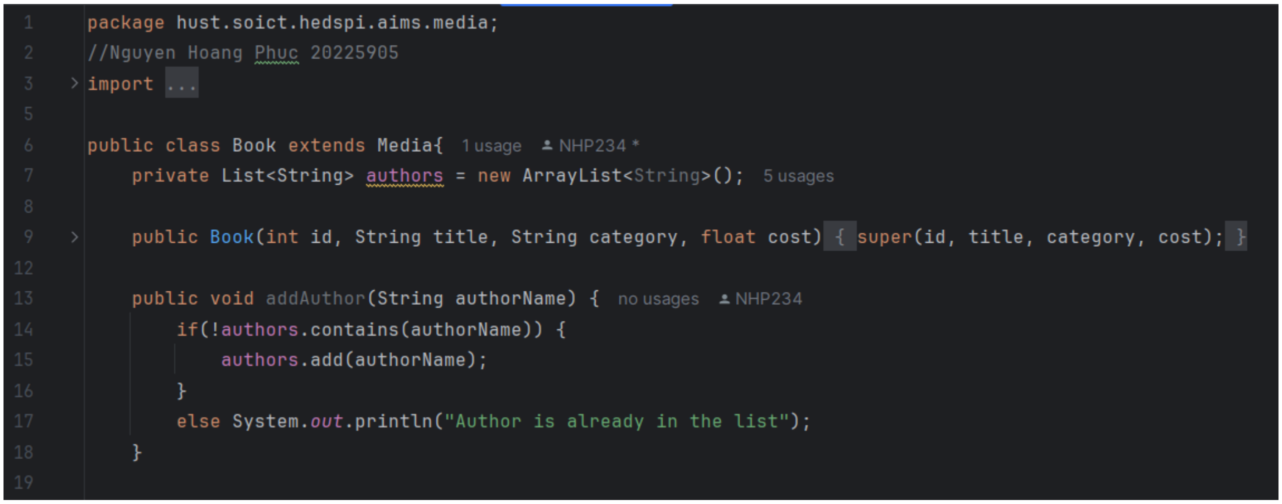
1 Class Diagram 38
2 UseCase Diagram 39
3 Answer Questions 39

Table of Figures

Figure 1.1: Book Class 1 4
Figure 1.2: Book Class 2........ 5
Figure 2.1: Media Class 1 6
Figure 2.2: Media Class 2 7
Figure 3.1: Disc Class 8
Figure 3.2: DigitalVideoDisc Class........ 9
Figure 3.3: CompactDisc Class 9
Figure 3.4: Track Class 10
Figure 3.5: CompactDisc Class 1 11
Figure 3.6: CompactDisc Class 2 12
Figure 4.1: Playable interface 13
Figure 4.2: Method play() của DigitalVideoDisc........ 13
Figure 4.3: Method play() của Track 13
Figure 4.4: Method play() của CompactDisc........ 13
Figure 5.1: Cart Class 1........ 14
Figure 5.2: Cart Class 2........ 15
Figure 5.3: Cart Class 3........ 16
Figure 5.4: Cart Class 4........ 17
Figure 6.1: Store Class 1 18
Figure 6.2: Store Class 2 19
Figure 7.1: Constructor Track Class 20
Figure 7.2: Constructor CompactDisc Class........ 20
Figure 7.3: Constructor Media Class 20
Figure 7.4: Constructor Disc Class........ 21
Figure 8.1: Override equals in Media Class 21
Figure 8.2: Override equals in Track Class 21
Figure 9.1: Code mô phỏng Polymorphism........ 22
Figure 9.2: Override toString() in Media Class........ 22
Figure 9.3: Result demo Polymorphism 22
Figure 10.1: Add the comparators as attributes of the Media class........ 23
Figure 10.2: MediaComparatorByCostTitle Class........ 23
Figure 10.3: MediaComparatorByTitleCost Class........ 24
Figure 11.1: Màn hình chính 24
Figure 11.2: Vào Trang View Store........ 25
Figure 11.3: See a media's details 26
Figure 11.4: Thêm vào Cart 26
Figure 11.5: Thêm media vào Cart 27
Figure 11.6: Play a media 27

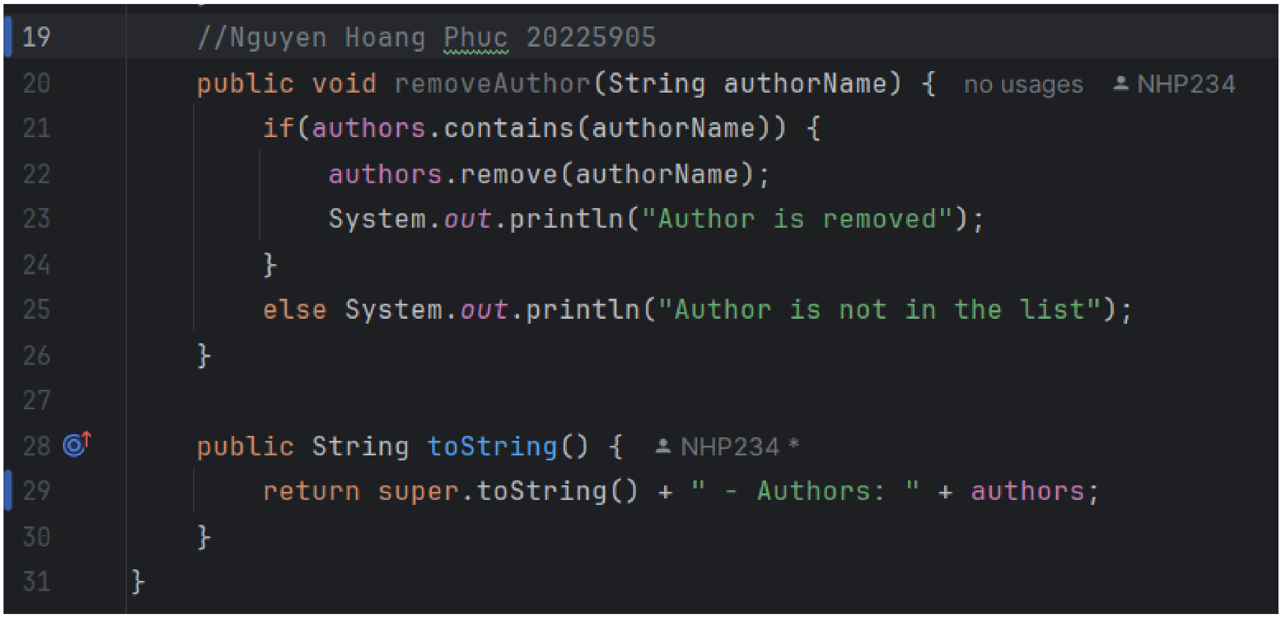
Figure 11.7: See current cart after sort........28
Figure 11.8: Vào Trang Update Store........29
Figure 11.9: Add a media to store........29
Figure 11.10: Result after add media to store........30
Figure 11.11: Remove a media from the store........30
Figure 11.12: Result after remove a media........31
Figure 11.13: Vào trang See current cart........31
Figure 11.14: Media in Cart........32
Figure 11.15: Filter Cart By id........32
Figure 11.16: Filter Cart By Title........33
Figure 11.17: Sort Cart By Title........34
Figure 11.18: Sort Cart By Cost........34
Figure 11.19: Remove media by id........35
Figure 11.20: Result after remove media in cart by id........35
Figure 11.21: Play a media in cart........36
Figure 11.22: Order........36
Figure 11.23: Result after order........37
Figure 12.1: Class Diagram........38
Figure 13.1: UseCase Diagram........39
Figure 14.1: Triển khai Comparable trong lớp trừu tượng Media........40
Figure 14.2: Mở rộng để so sánh nhiều thuộc tính hơn........40
Figure 14.3: Triển khai tại lớp con........40

1 Create the Book class



```java
package hust.soict.hedspi.aims.media;
//Nguyen Hoang Phuc 20225905
import ...

public class Book extends Media{
    private List<String> authors = new ArrayList<String>();

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

    public void addAuthor(String authorName) {
        if(!authors.contains(authorName)) {
            authors.add(authorName);
        }
        else System.out.println("Author is already in the list");
    }
```

Figure 1.1: Book Class 1

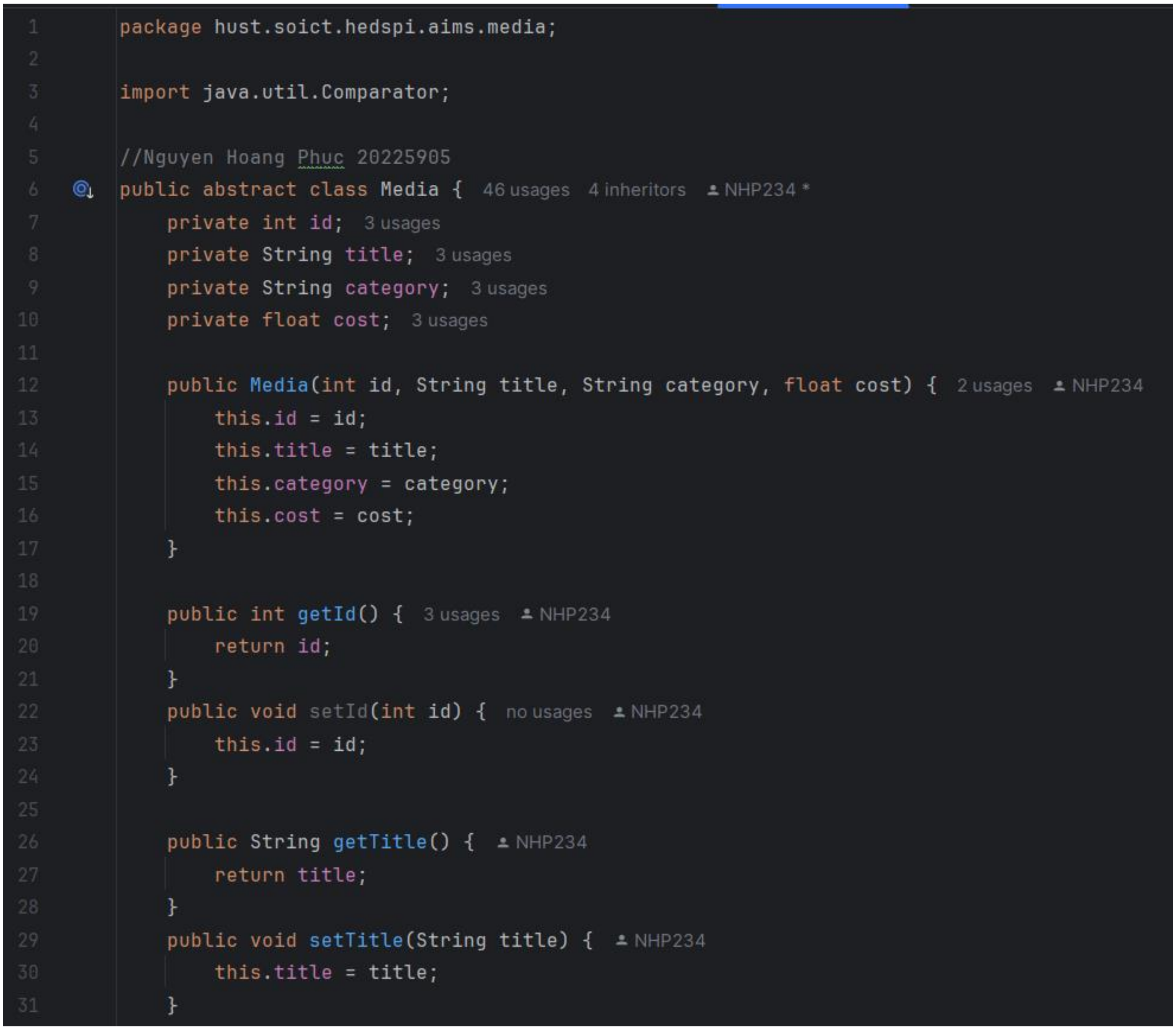


```java
    //Nguyen Hoang Phuc 20225905
    public void removeAuthor(String authorName) {
        if(authors.contains(authorName)) {
            authors.remove(authorName);
            System.out.println("Author is removed");
        }
        else System.out.println("Author is not in the list");
    }

    public String toString() {
        return super.toString() + " - Authors: " + authors;
    }
}
```

Figure 1.2: Book Class 2

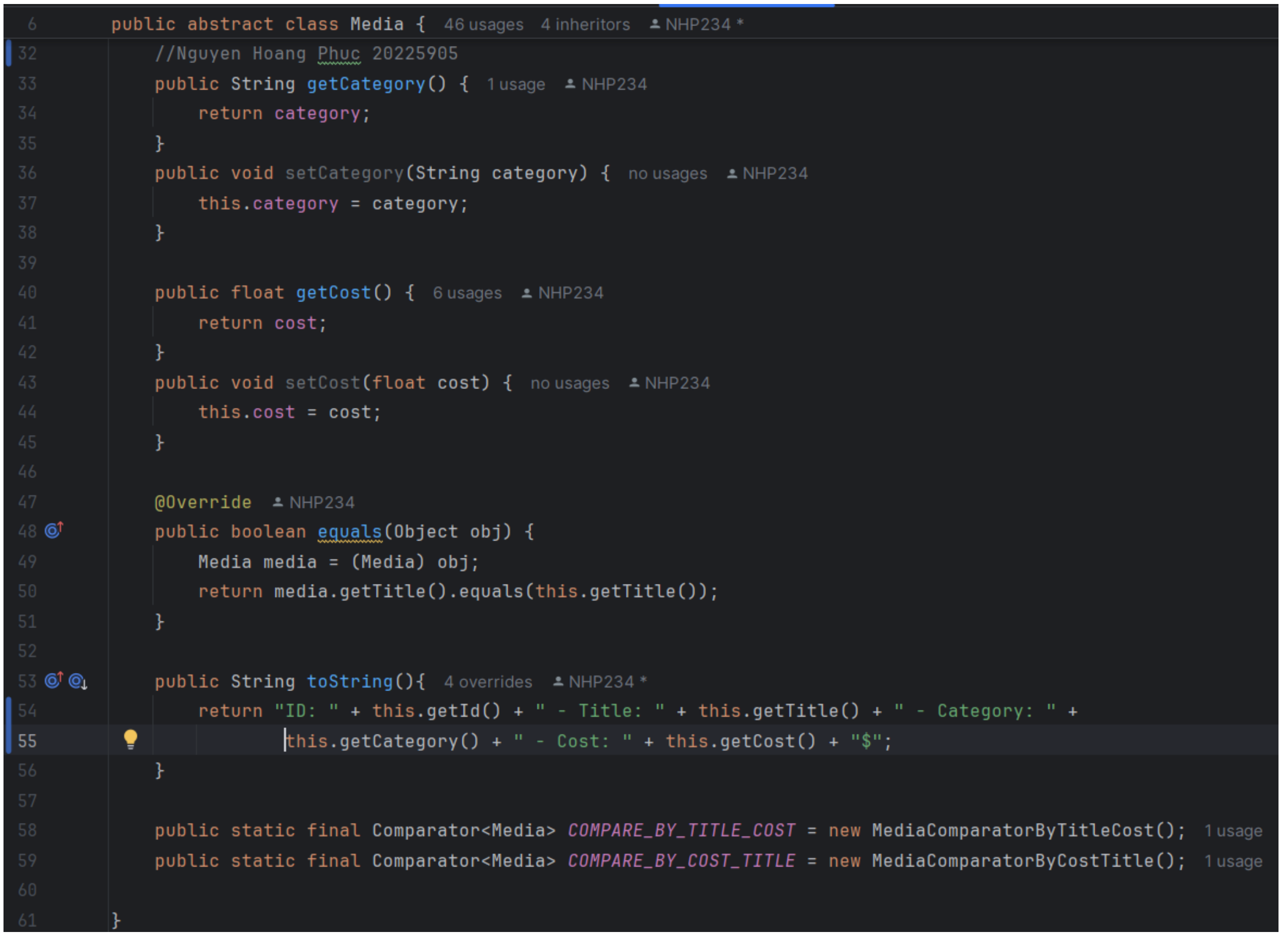
2 Creating the abstract Media class

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



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

import java.util.Comparator;

//Nguyen Hoang Phuc 20225905
public abstract class Media {  46 usages  4 inheritors  NHP234 *
    private int id;  3 usages
    private String title;  3 usages
    private String category;  3 usages
    private float cost;  3 usages

    public Media(int id, String title, String category, float cost) {  2 usages  NHP234
        this.id = id;
        this.title = title;
        this.category = category;
        this.cost = cost;
    }

    public int getId() {  3 usages  NHP234
        return id;
    }
    public void setId(int id) {  no usages  NHP234
        this.id = id;
    }

    public String getTitle() {  NHP234
        return title;
    }
    public void setTitle(String title) {  NHP234
        this.title = title;
    }
```

Figure 2.1: Media Class 1

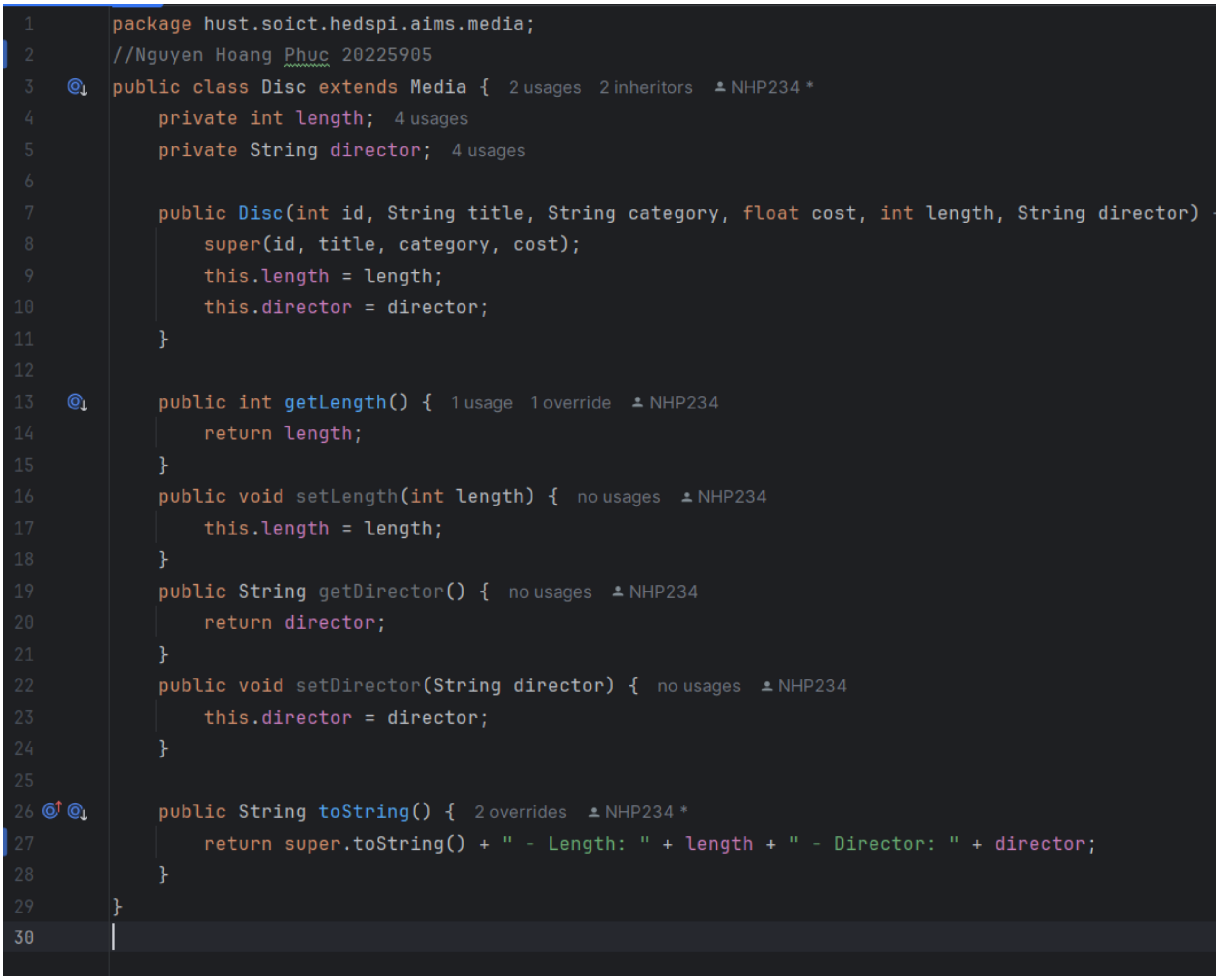


```java
public abstract class Media {
    //Nguyen Hoang Phuc 20225905
    public String getCategory() {
        return category;
    }
    public void setCategory(String category) {
        this.category = category;
    }

    public float getCost() {
        return cost;
    }
    public void setCost(float cost) {
        this.cost = cost;
    }

    @Override
    public boolean equals(Object obj) {
        Media media = (Media) obj;
        return media.getTitle().equals(this.getTitle());
    }

    public String toString(){
        return "ID: " + this.getId() + " - Title: " + this.getTitle() + " - Category: " +
                this.getCategory() + " - Cost: " + this.getCost() + "$";
    }

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

}
```

Figure 2.2: Media Class 2

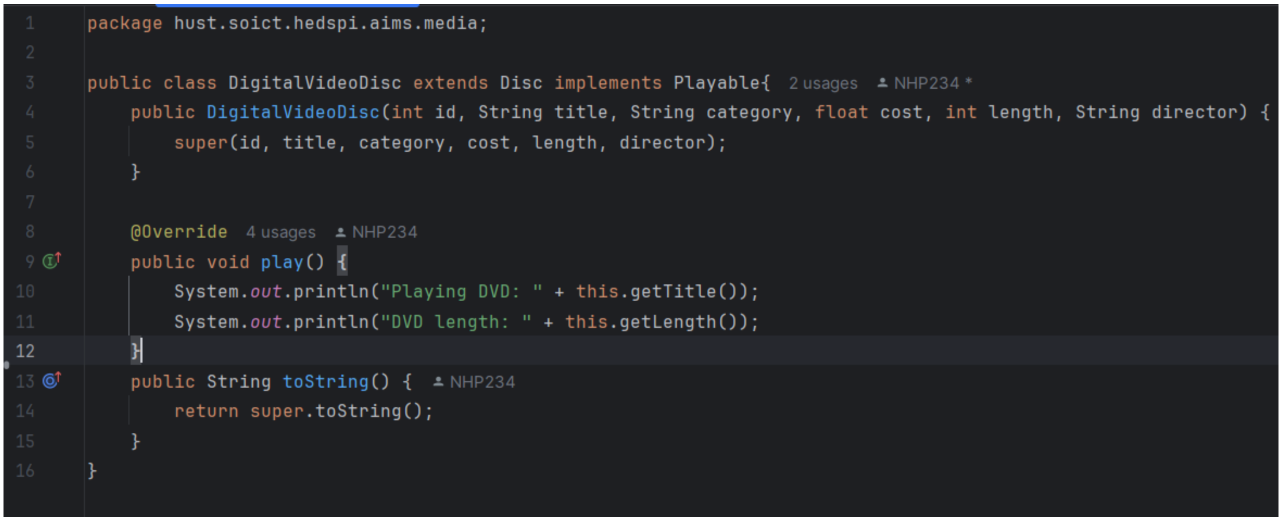
3 Creating the CompactDisc class

3.1 Create the Disc class extending the Media class



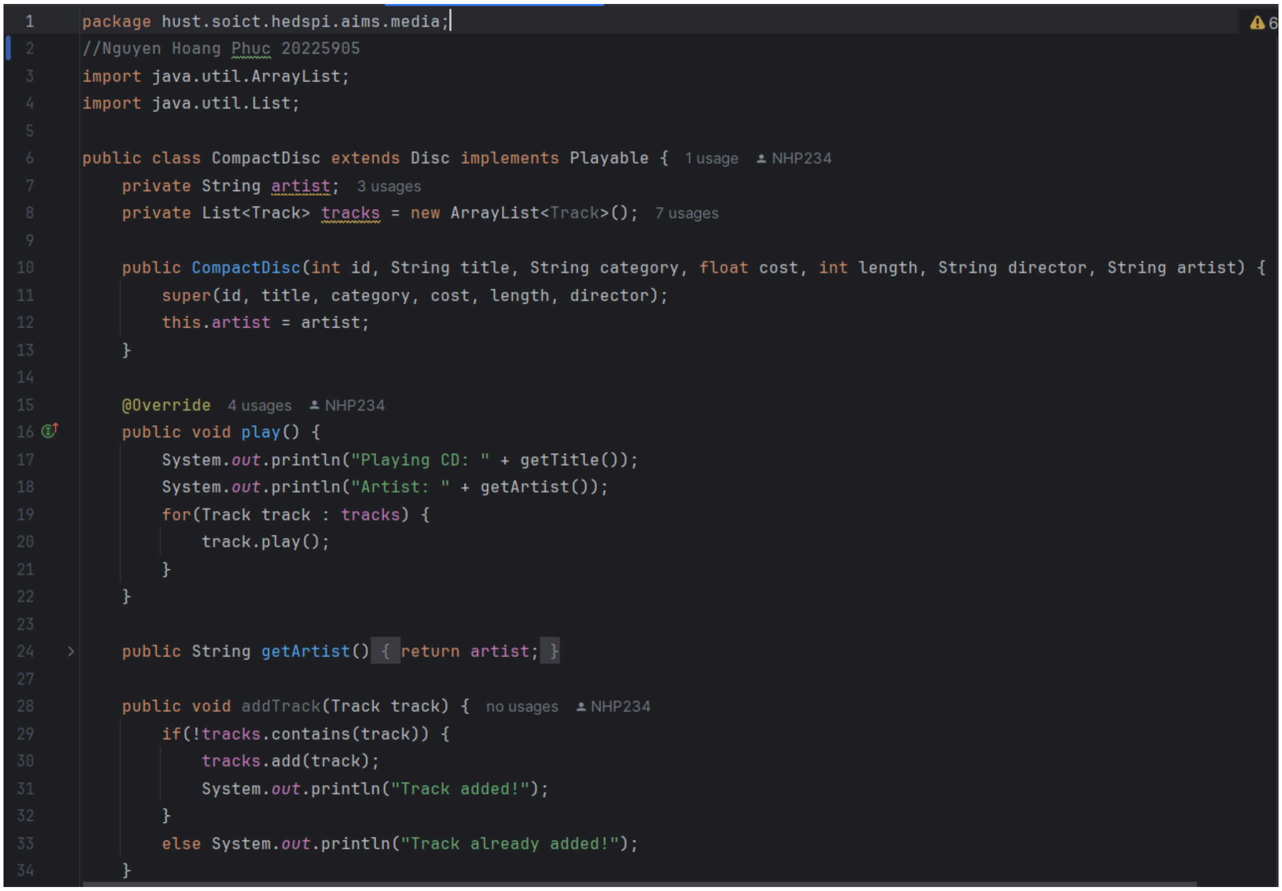
```java
package hust.soict.hedspi.aims.media;
//Nguyen Hoang Phuc 20225905
public class Disc extends Media {
    private int length;
    private String director;

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

    public int getLength() {
        return length;
    }
    public void setLength(int length) {
        this.length = length;
    }
    public String getDirector() {
        return director;
    }
    public void setDirector(String director) {
        this.director = director;
    }

    public String toString() {
        return super.toString() + " - Length: " + length + " - Director: " + director;
    }
}
```

Figure 3.1: Disc Class

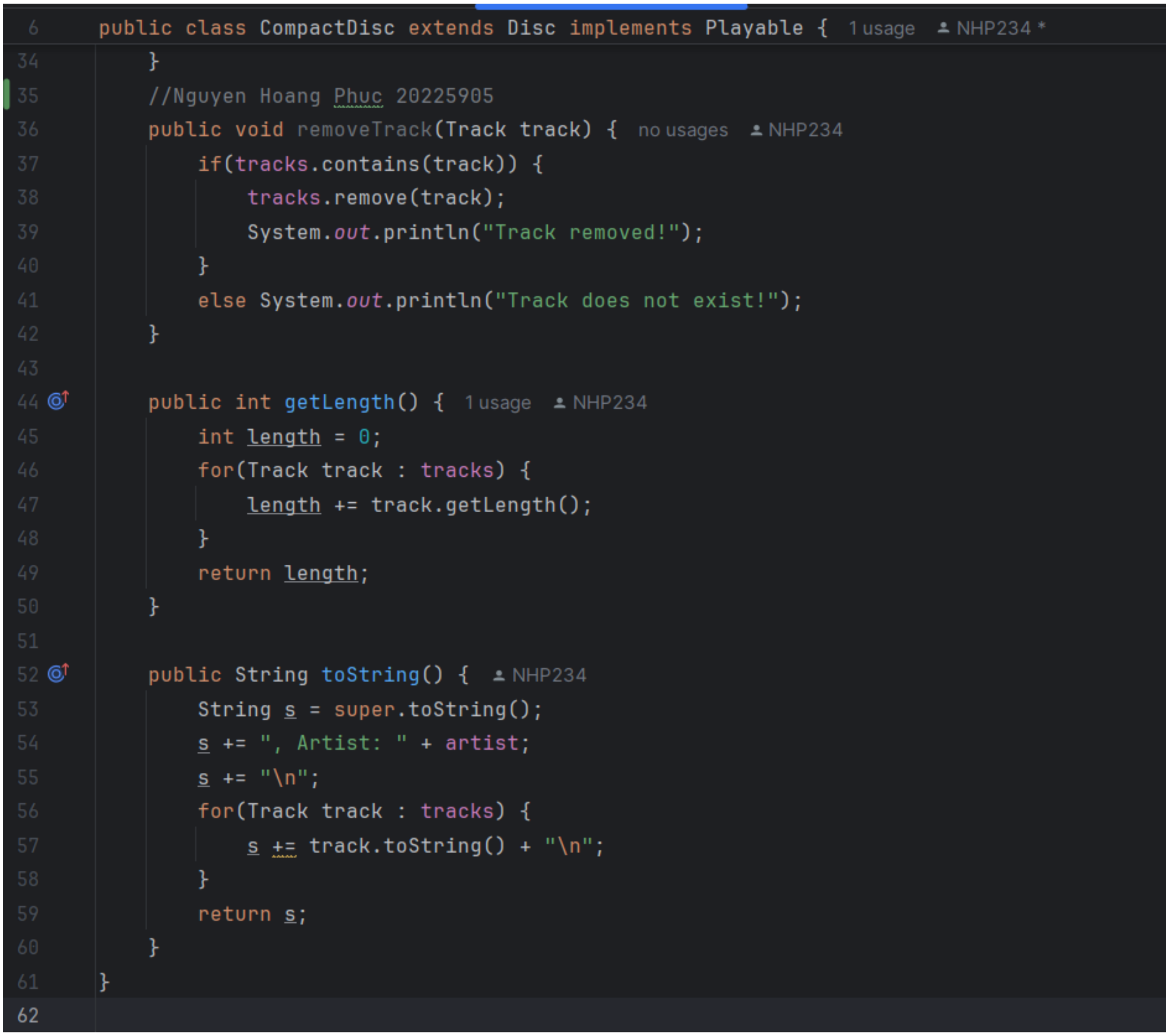


```java
package hust.soict.hedspi.aims.media;

public class DigitalVideoDisc extends Disc implements Playable{
    public DigitalVideoDisc(int id, String title, String category, float cost, int length, String director) {
        super(id, title, category, cost, length, director);
    }

    @Override
    public void play() {
        System.out.println("Playing DVD: " + this.getTitle());
        System.out.println("DVD length: " + this.getLength());
    }
    public String toString() {
        return super.toString();
    }
}
```

Figure 3.2: DigitalVideoDisc Class



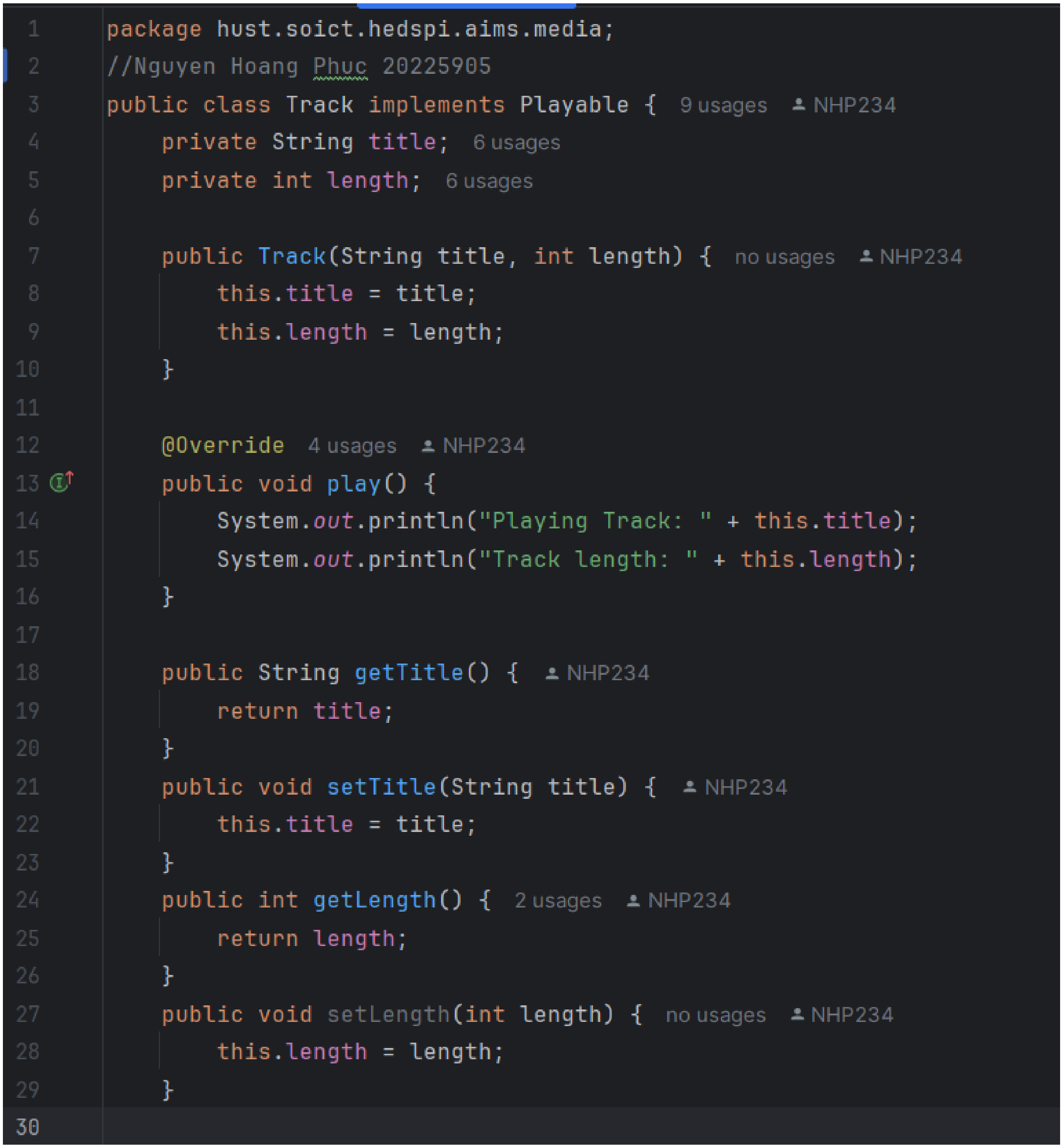
```
package hust.soict.hedspi.aims.media;
//Nguyen Hoang Phuc 20225905
import java.util.ArrayList;
import java.util.List;

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

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

    @Override
    public void play() {
        System.out.println("Playing CD: " + getTitle());
        System.out.println("Artist: " + getArtist());
        for(Track track : tracks) {
            track.play();
        }
    }

    public String getArtist() { return artist; }

    public void addTrack(Track track) {
        if(!tracks.contains(track)) {
            tracks.add(track);
            System.out.println("Track added!");
        }
        else System.out.println("Track already added!");
    }
```



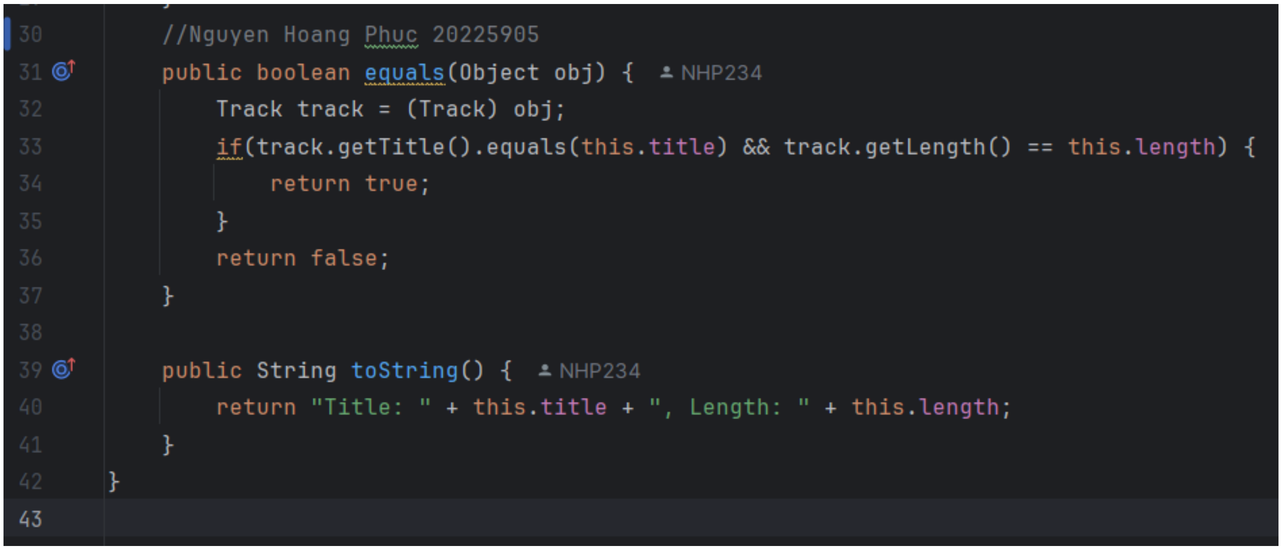
```
public class CompactDisc extends Disc implements Playable {  1 usage  NHP234 *
    }
    //Nguyen Hoang Phuc 20225905
    public void removeTrack(Track track) {  no usages  NHP234
        if(tracks.contains(track)) {
            tracks.remove(track);
            System.out.println("Track removed!");
        }
        else System.out.println("Track does not exist!");
    }

    public int getLength() {  1 usage  NHP234
        int length = 0;
        for(Track track : tracks) {
            length += track.getLength();
        }
        return length;
    }

    public String toString() {  NHP234
        String s = super.toString();
        s += ", Artist: " + artist;
        s += "\n";
        for(Track track : tracks) {
            s += track.toString() + "\n";
        }
        return s;
    }
}
```

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.



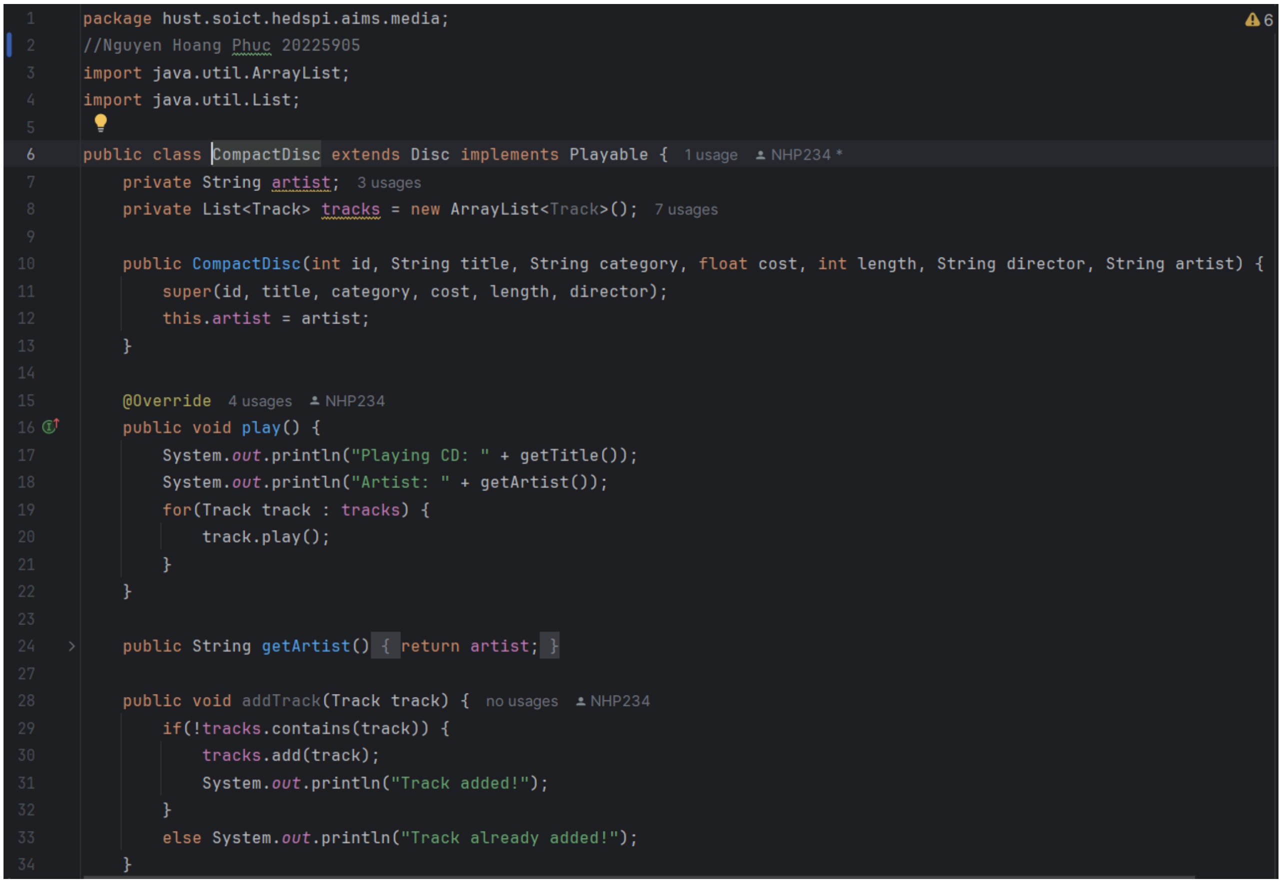
```java
package hust.soict.hedspi.aims.media;
//Nguyen Hoang Phuc 20225905
public class Track implements Playable {  9 usages  NHP234
    private String title;  6 usages
    private int length;  6 usages

    public Track(String title, int length) {  no usages  NHP234
        this.title = title;
        this.length = length;
    }

    @Override  4 usages  NHP234
    public void play() {
        System.out.println("Playing Track: " + this.title);
        System.out.println("Track length: " + this.length);
    }

    public String getTitle() {  NHP234
        return title;
    }
    public void setTitle(String title) {  NHP234
        this.title = title;
    }
    public int getLength() {  2 usages  NHP234
        return length;
    }
    public void setLength(int length) {  no usages  NHP234
        this.length = length;
    }

```



```java
    //Nguyen Hoang Phuc 20225905
    public boolean equals(Object obj) {
        Track track = (Track) obj;
        if(track.getTitle().equals(this.title) && track.getLength() == this.length) {
            return true;
        }
        return false;
    }

    public String toString() {
        return "Title: " + this.title + ", Length: " + this.length;
    }
}
```

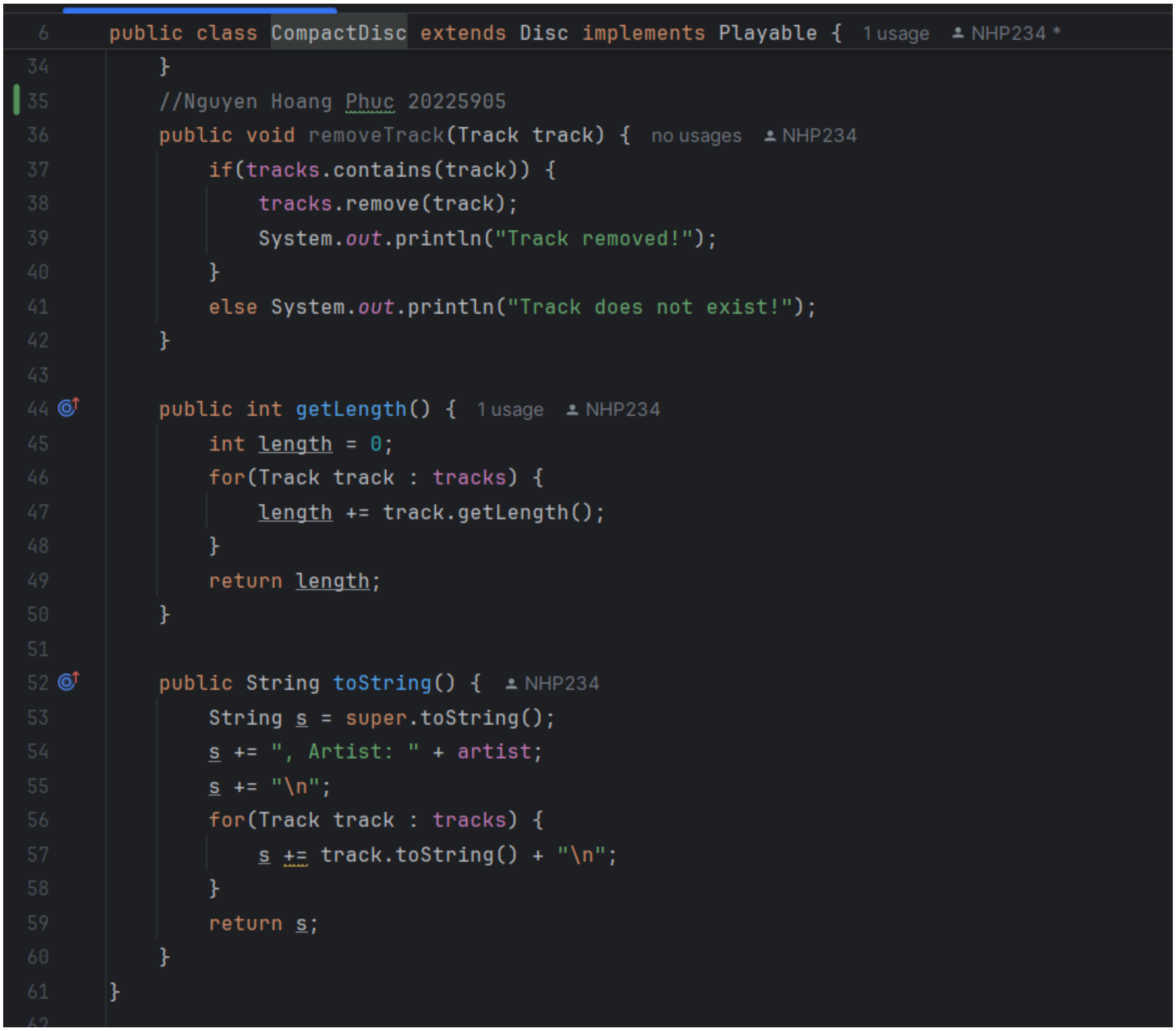
Figure 3.4: Track Class

3.3 Open the CompactDisc class



```java
package hust.soict.hedspi.aims.media;
//Nguyen Hoang Phuc 20225905
import java.util.ArrayList;
import java.util.List;

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

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

    @Override
    public void play() {
        System.out.println("Playing CD: " + getTitle());
        System.out.println("Artist: " + getArtist());
        for(Track track : tracks) {
            track.play();
        }
    }

    public String getArtist() { return artist; }

    public void addTrack(Track track) {
        if(!tracks.contains(track)) {
            tracks.add(track);
            System.out.println("Track added!");
        }
        else System.out.println("Track already added!");
    }
```

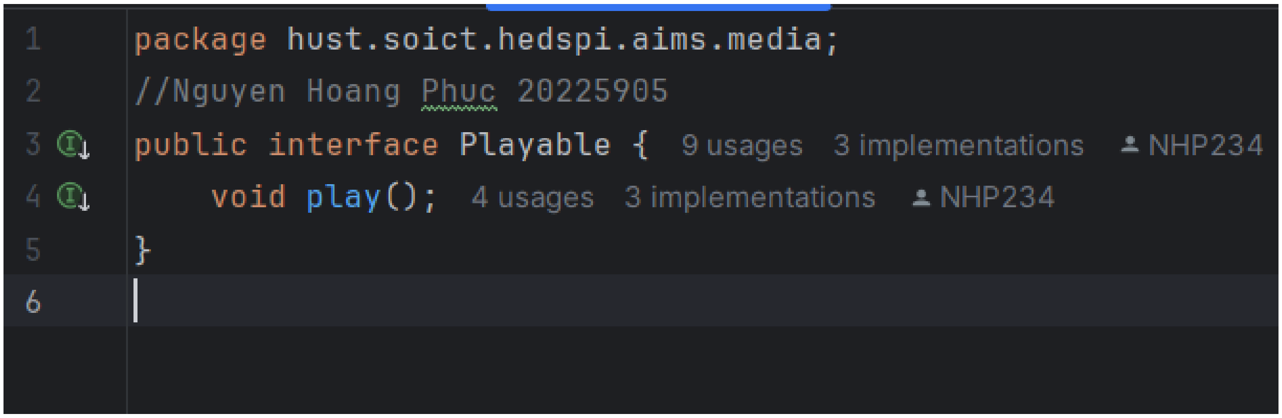
Figure 3.5: CompactDisc Class 1



```java
public class CompactDisc extends Disc implements Playable {
    }
    //Nguyen Hoang Phuc 20225905
    public void removeTrack(Track track) {
        if(tracks.contains(track)) {
            tracks.remove(track);
            System.out.println("Track removed!");
        }
        else System.out.println("Track does not exist!");
    }

    public int getLength() {
        int length = 0;
        for(Track track : tracks) {
            length += track.getLength();
        }
        return length;
    }

    public String toString() {
        String s = super.toString();
        s += ", Artist: " + artist;
        s += "\n";
        for(Track track : tracks) {
            s += track.toString() + "\n";
        }
        return s;
    }
}
```

Figure 3.6: CompactDisc Class 2

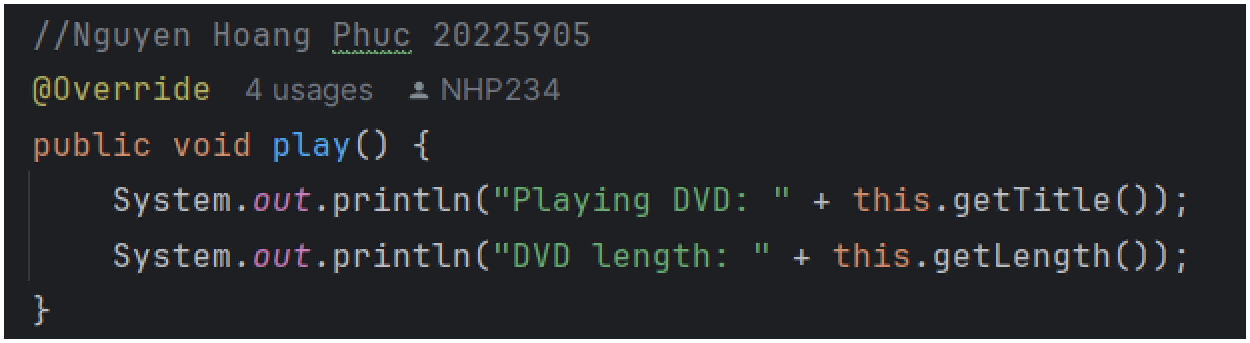
4 Create the Playable interface



```java
package hust.soict.hedspi.aims.media;
//Nguyen Hoang Phuc 20225905
public interface Playable {
    void play();
}
```

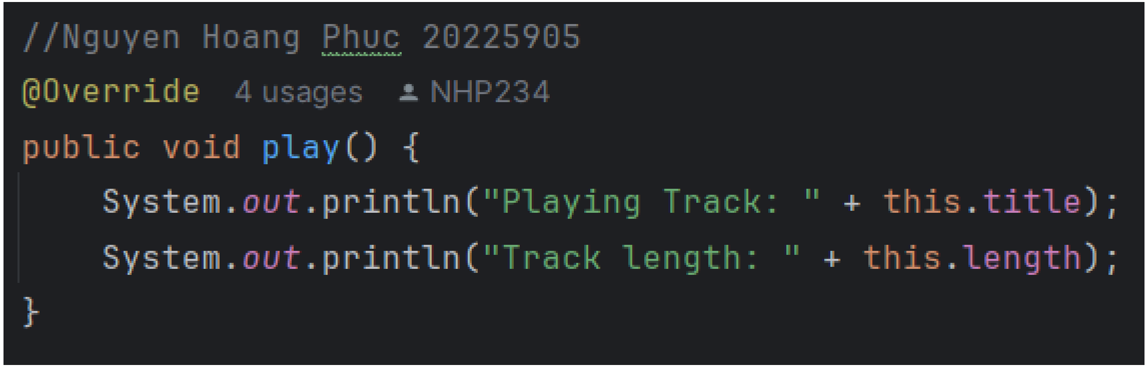
Figure 4.1: Playable interface

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



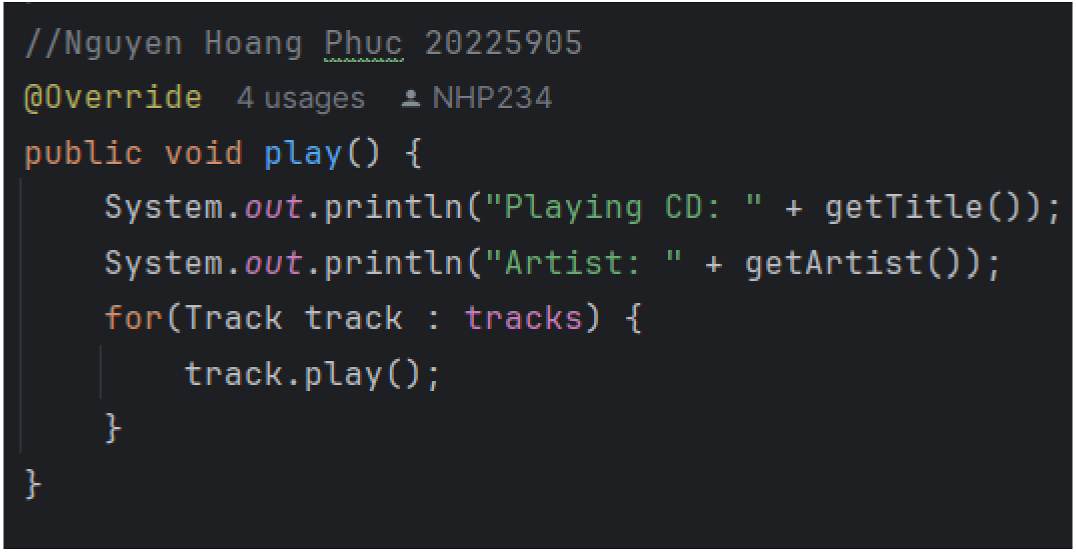
```java
//Nguyen Hoang Phuc 20225905
@Override  4 usages  NHP234
public void play() {
    System.out.println("Playing DVD: " + this.getTitle());
    System.out.println("DVD length: " + this.getLength());
}
```

Figure 4.2: Method play() của DigitalVideoDisc



```java
//Nguyen Hoang Phuc 20225905
@Override  4 usages  NHP234
public void play() {
    System.out.println("Playing Track: " + this.title);
    System.out.println("Track length: " + this.length);
}
```

Figure 4.3: Method play() của Track

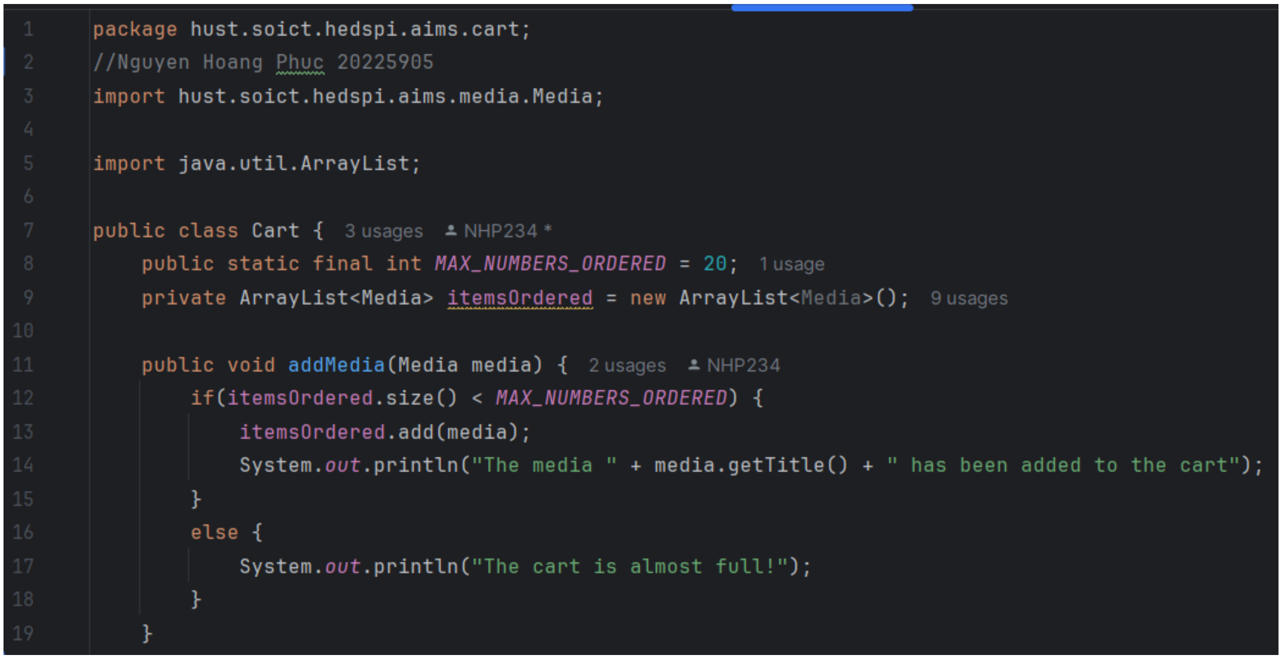


```java
//Nguyen Hoang Phuc 20225905
@Override  4 usages  NHP234
public void play() {
    System.out.println("Playing CD: " + getTitle());
    System.out.println("Artist: " + getArtist());
    for(Track track : tracks) {
        track.play();
    }
}
```

Figure 4.4: Method play() của CompactDisc

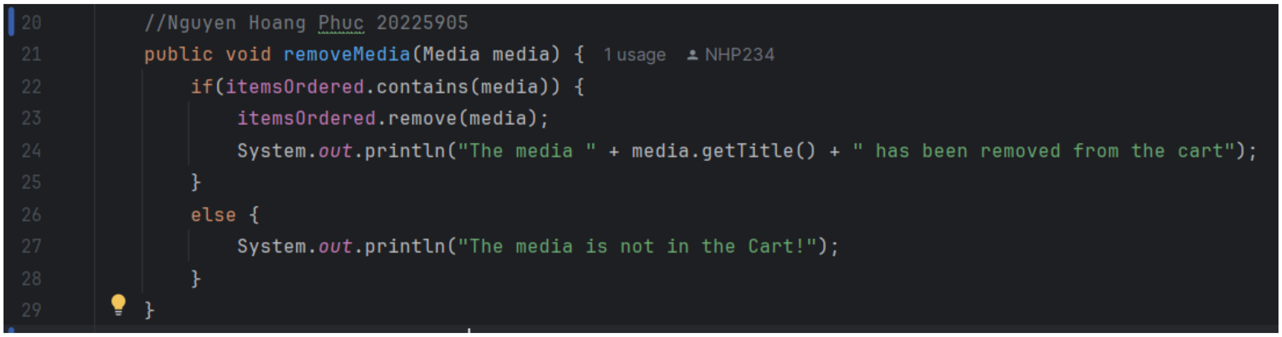
5 Update the Cart class to work with Media

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.



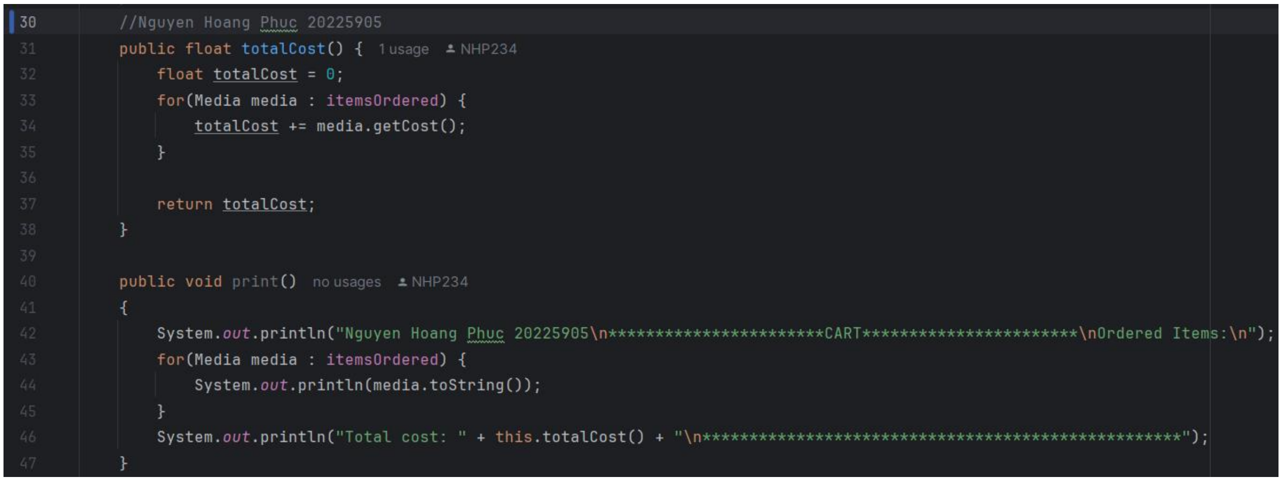
```
package hust.soict.hedspi.aims.cart;
//Nguyen Hoang Phuc 20225905
import hust.soict.hedspi.aims.media.Media;

import java.util.ArrayList;

public class Cart {  3 usages  NHP234 *
    public static final int MAX_NUMBERS_ORDERED = 20;  1 usage
    private ArrayList<Media> itemsOrdered = new ArrayList<Media>();  9 usages

    public void addMedia(Media media) {  2 usages  NHP234
        if(itemsOrdered.size() < MAX_NUMBERS_ORDERED) {
            itemsOrdered.add(media);
            System.out.println("The media " + media.getTitle() + " has been added to the cart");
        }
        else {
            System.out.println("The cart is almost full!");
        }
    }
```

Figure 5.1: Cart Class 1



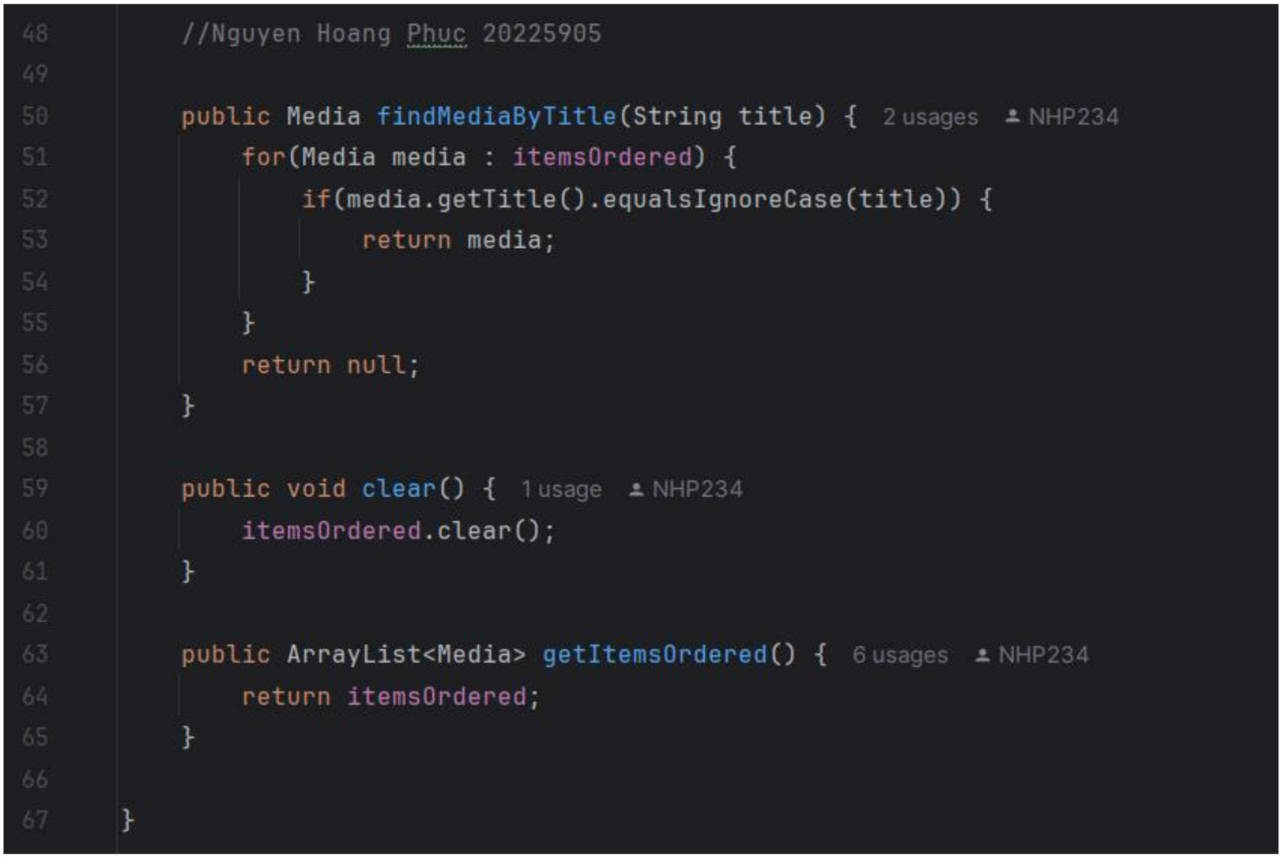
```
    //Nguyen Hoang Phuc 20225905
    public void removeMedia(Media media) {  1 usage  NHP234
        if(itemsOrdered.contains(media)) {
            itemsOrdered.remove(media);
            System.out.println("The media " + media.getTitle() + " has been removed from the cart");
        }
        else {
            System.out.println("The media is not in the Cart!");
        }
    }
```

Figure 5.2: Cart Class 2



```
    //Nguyen Hoang Phuc 20225905
    public float totalCost() {  1 usage  NHP234
        float totalCost = 0;
        for(Media media : itemsOrdered) {
            totalCost += media.getCost();
        }

        return totalCost;
    }

    public void print()  no usages  NHP234
    {
        System.out.println("Nguyen Hoang Phuc 20225905\n************************CART************************\nOrdered Items:\n");
        for(Media media : itemsOrdered) {
            System.out.println(media.toString());
        }
        System.out.println("Total cost: " + this.totalCost() + "\n****************************************************");
    }
```

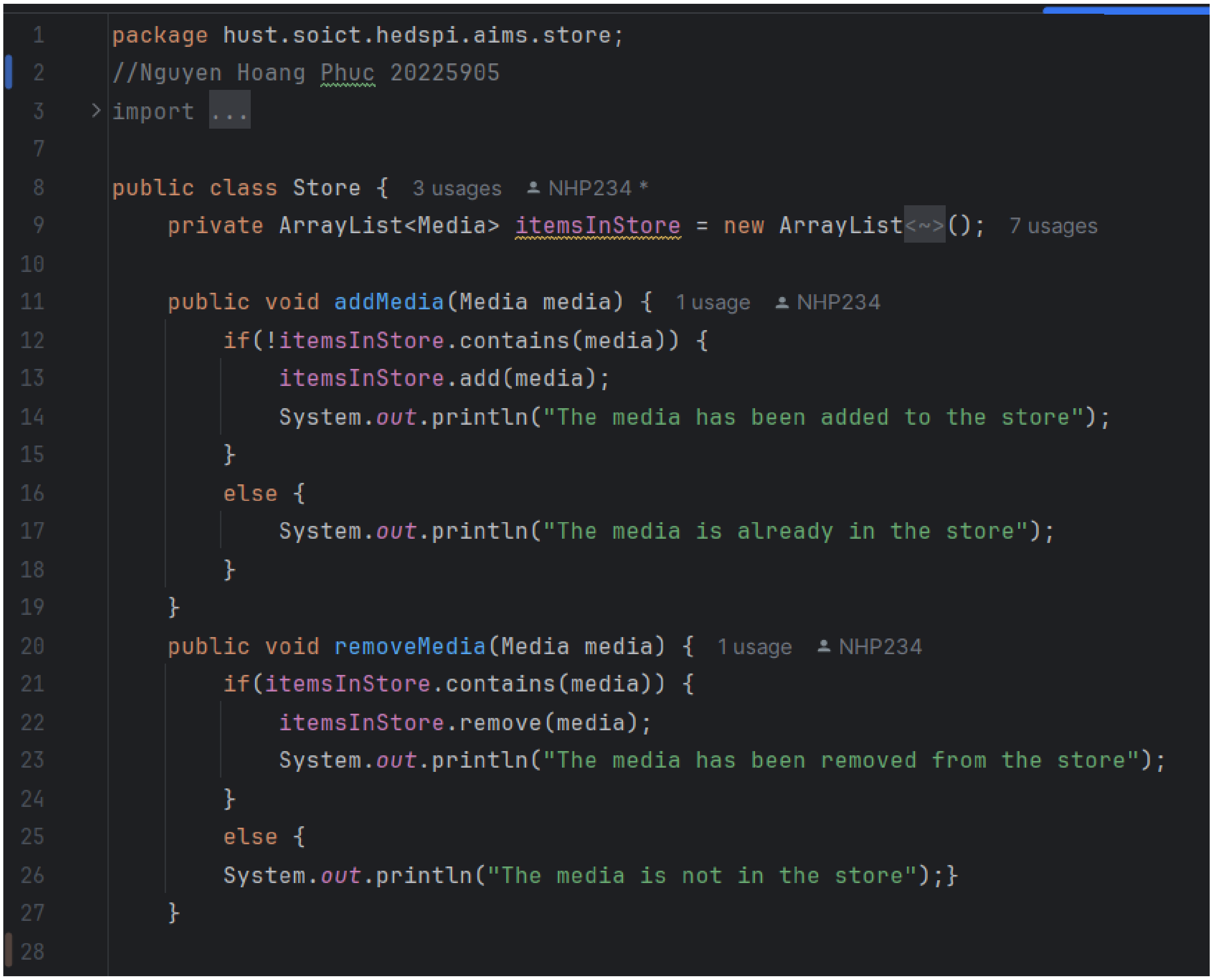
Figure 5.3: Cart Class 3



```java
    //Nguyen Hoang Phuc 20225905

    public Media findMediaByTitle(String title) {
        for(Media media : itemsOrdered) {
            if(media.getTitle().equalsIgnoreCase(title)) {
                return media;
            }
        }
        return null;
    }

    public void clear() {
        itemsOrdered.clear();
    }

    public ArrayList<Media> getItemsOrdered() {
        return itemsOrdered;
    }

}
```

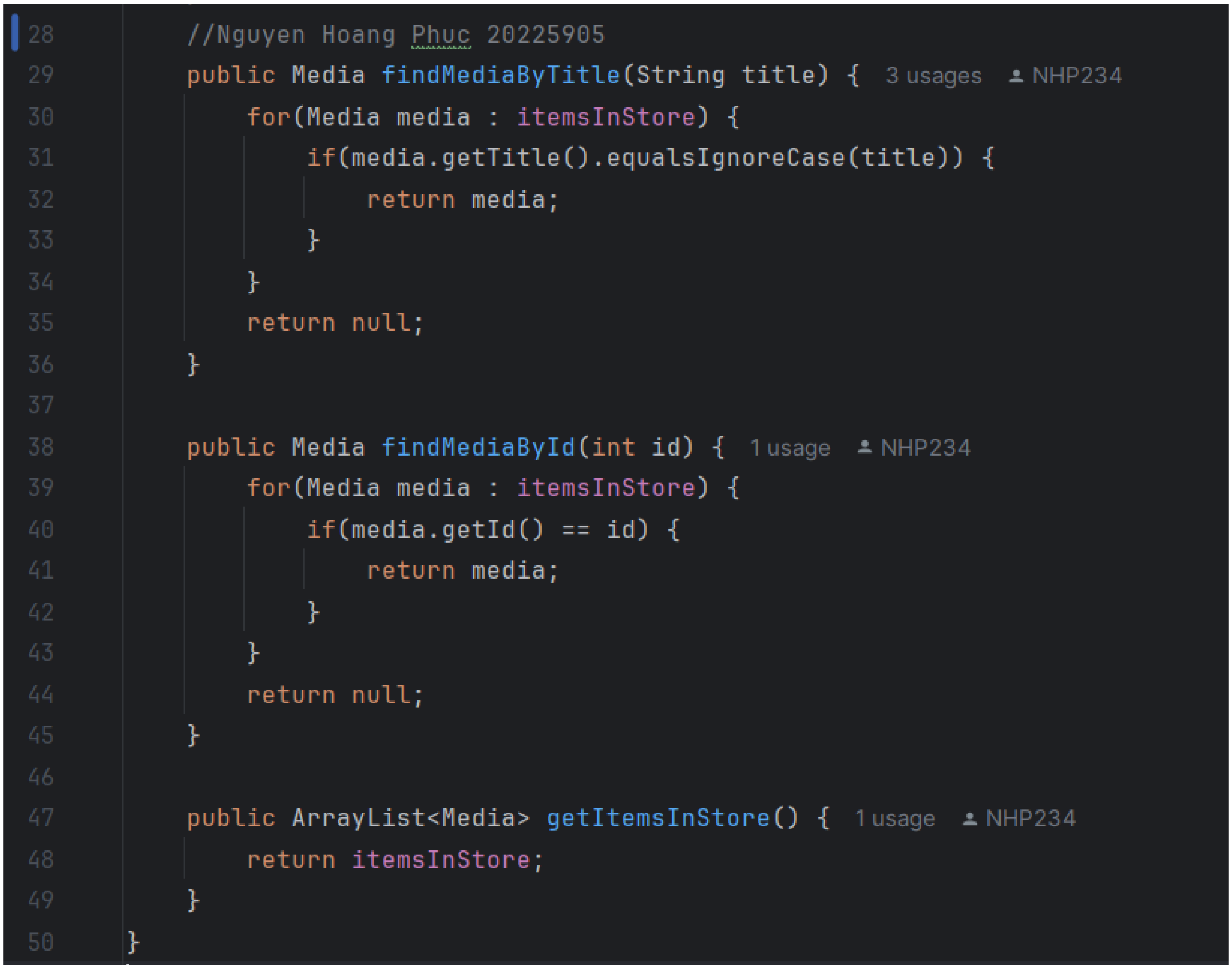
Figure 5.4: Cart Class 4

6 Update the Store class to work with Media



```java
package hust.soict.hedspi.aims.store;
//Nguyen Hoang Phuc 20225905
import ...

public class Store {
    private ArrayList<Media> itemsInStore = new ArrayList<~>();

    public void addMedia(Media media) {
        if(!itemsInStore.contains(media)) {
            itemsInStore.add(media);
            System.out.println("The media has been added to the store");
        }
        else {
            System.out.println("The media is already in the store");
        }
    }
    public void removeMedia(Media media) {
        if(itemsInStore.contains(media)) {
            itemsInStore.remove(media);
            System.out.println("The media has been removed from the store");
        }
        else {
        System.out.println("The media is not in the store");}
    }

```

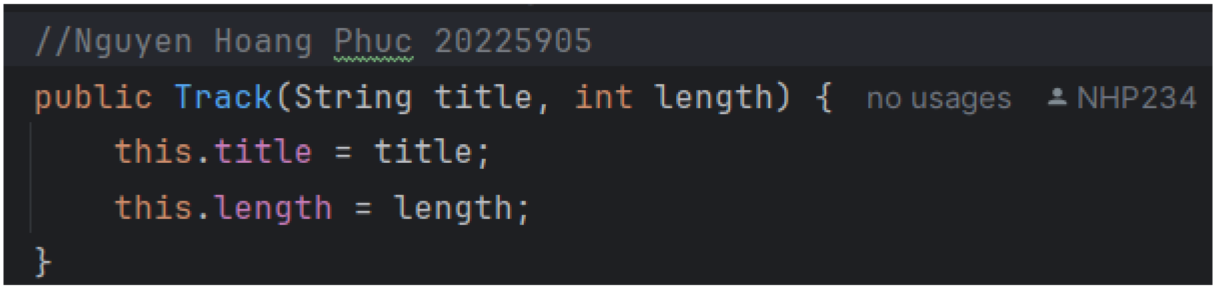
Figure 6.1: Store Class 1



```
    //Nguyen Hoang Phuc 20225905
    public Media findMediaByTitle(String title) {  3 usages  NHP234
        for(Media media : itemsInStore) {
            if(media.getTitle().equalsIgnoreCase(title)) {
                return media;
            }
        }
        return null;
    }

    public Media findMediaById(int id) {  1 usage  NHP234
        for(Media media : itemsInStore) {
            if(media.getId() == id) {
                return media;
            }
        }
        return null;
    }

    public ArrayList<Media> getItemsInStore() {  1 usage  NHP234
        return itemsInStore;
    }
}
```

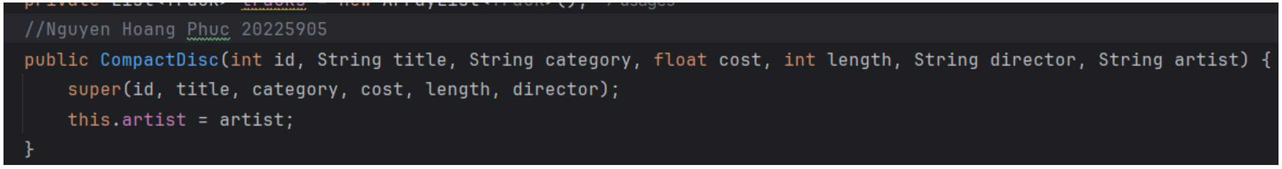
Figure 6.2: Store Class 2

7 Constructors of whole classes and parent classes



```
//Nguyen Hoang Phuc 20225905
public Track(String title, int length) {  no usages  NHP234
    this.title = title;
    this.length = length;
}
```

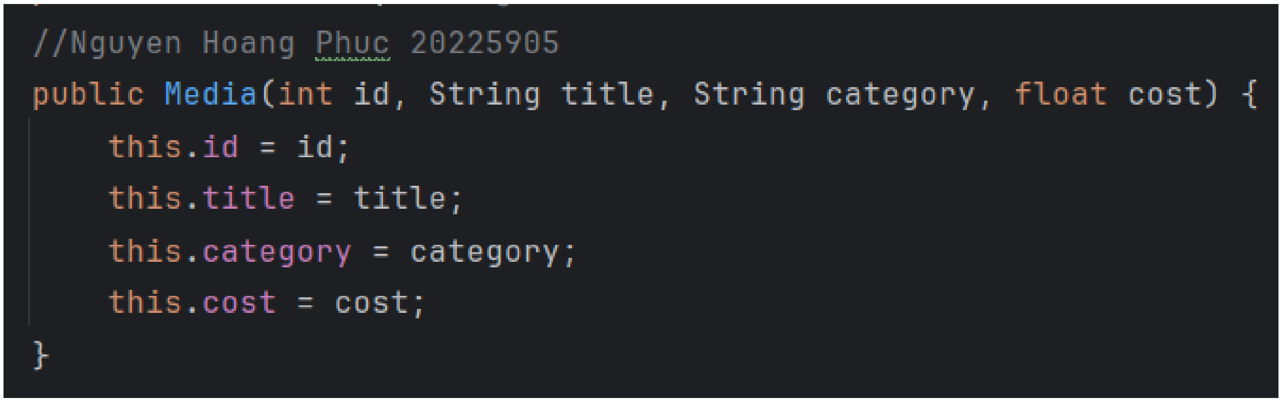
Figure 7.1: Constructor Track Class



```
private List<Track> tracks = new ArrayList<Track>();  7 usages
//Nguyen Hoang Phuc 20225905
public CompactDisc(int id, String title, String category, float cost, int length, String director, String artist) {
    super(id, title, category, cost, length, director);
    this.artist = artist;
}
```

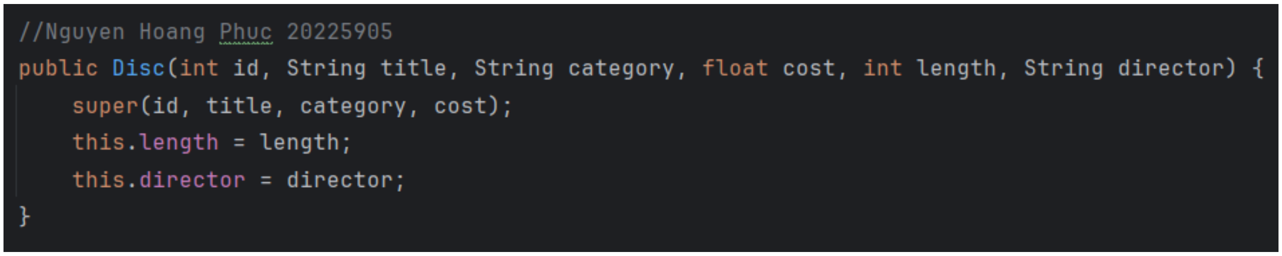
Figure 7.2: Constructor CompactDisc Class

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



```
//Nguyen Hoang Phuc 20225905
public Media(int id, String title, String category, float cost) {
    this.id = id;
    this.title = title;
    this.category = category;
    this.cost = cost;
}
```

Figure 7.3: Constructor Media Class

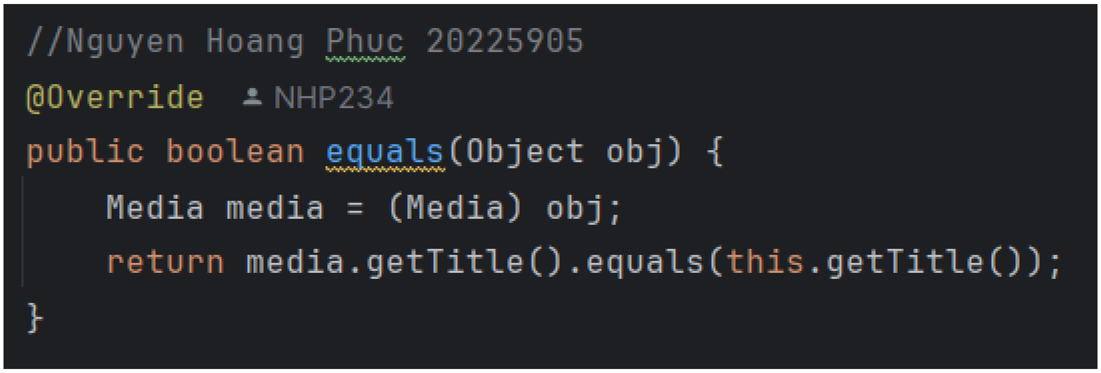


```
//Nguyen Hoang Phuc 20225905
public Disc(int id, String title, String category, float cost, int length, String director) {
    super(id, title, category, cost);
    this.length = length;
    this.director = director;
}
```

Figure 7.4: Constructor Disc Class

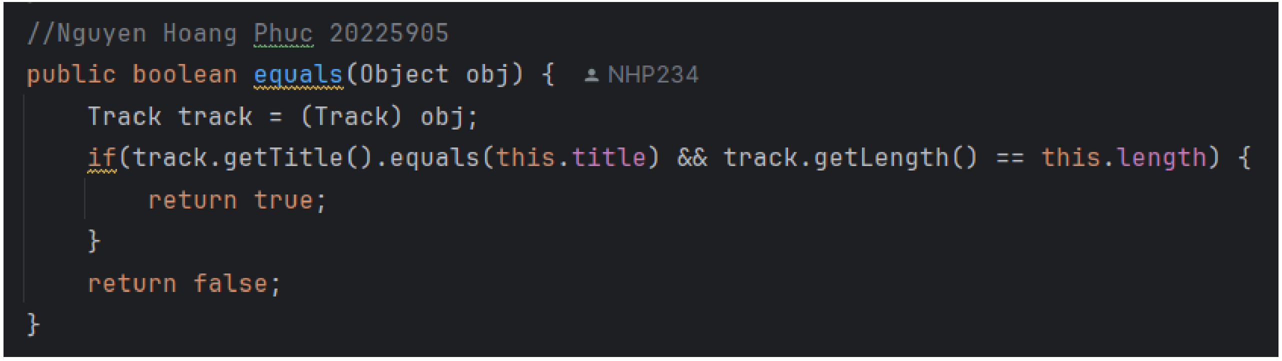
8 Unique item in a list

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.



```
//Nguyen Hoang Phuc 20225905
@Override  NHP234
public boolean equals(Object obj) {
    Media media = (Media) obj;
    return media.getTitle().equals(this.getTitle());
}
```

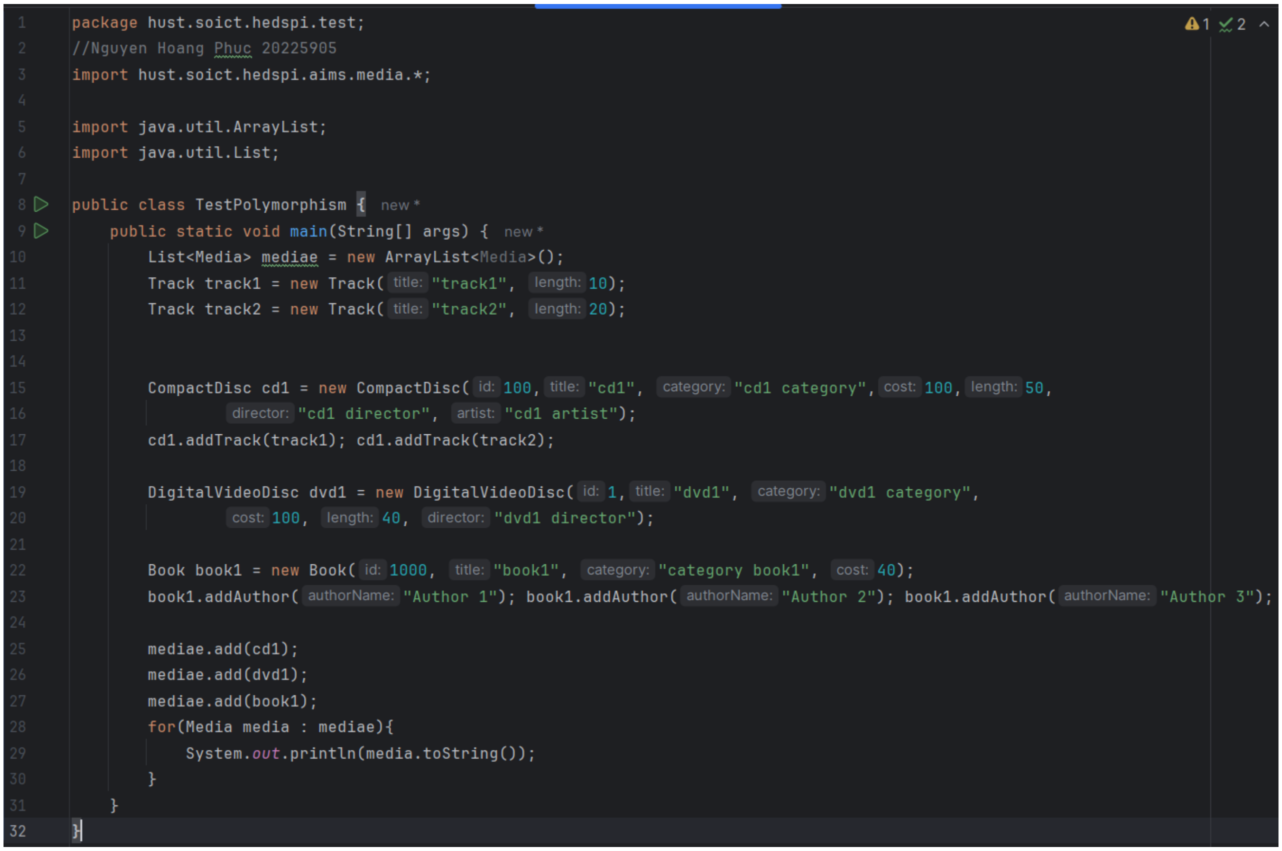
Figure 8.1: Override equals in Media Class



```
//Nguyen Hoang Phuc 20225905
public boolean equals(Object obj) {  NHP234
    Track track = (Track) obj;
    if(track.getTitle().equals(this.title) && track.getLength() == this.length) {
        return true;
    }
    return false;
}
```

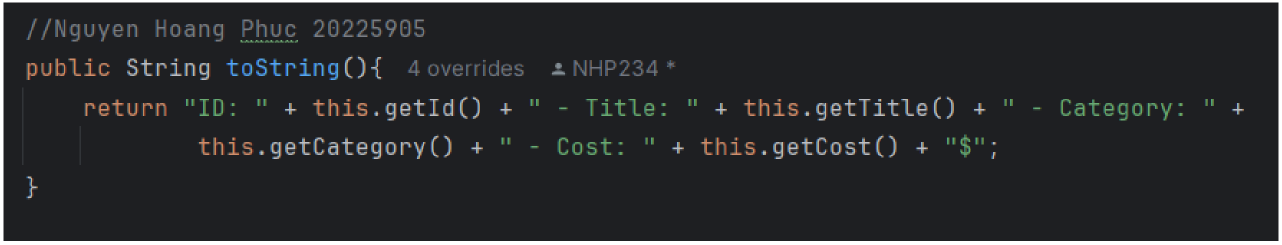
Figure 8.2: Override equals in Track Class

9 Polymorphism with toString() method



```
package hust.soict.hedspi.test;
//Nguyen Hoang Phuc 20225905
import hust.soict.hedspi.aims.media.*;

import java.util.ArrayList;
import java.util.List;

public class TestPolymorphism {  new *
    public static void main(String[] args) {  new *
        List<Media> mediae = new ArrayList<Media>();
        Track track1 = new Track( title: "track1",  length: 10);
        Track track2 = new Track( title: "track2",  length: 20);


        CompactDisc cd1 = new CompactDisc( id: 100, title: "cd1",  category: "cd1 category", cost: 100, length: 50,
                 director: "cd1 director",  artist: "cd1 artist");
        cd1.addTrack(track1); cd1.addTrack(track2);

        DigitalVideoDisc dvd1 = new DigitalVideoDisc( id: 1, title: "dvd1",  category: "dvd1 category",
                 cost: 100,  length: 40,  director: "dvd1 director");

        Book book1 = new Book( id: 1000,  title: "book1",  category: "category book1",  cost: 40);
        book1.addAuthor( authorName: "Author 1"); book1.addAuthor( authorName: "Author 2"); book1.addAuthor( authorName: "Author 3");

        mediae.add(cd1);
        mediae.add(dvd1);
        mediae.add(book1);
        for(Media media : mediae){
            System.out.println(media.toString());
        }
    }
}
```

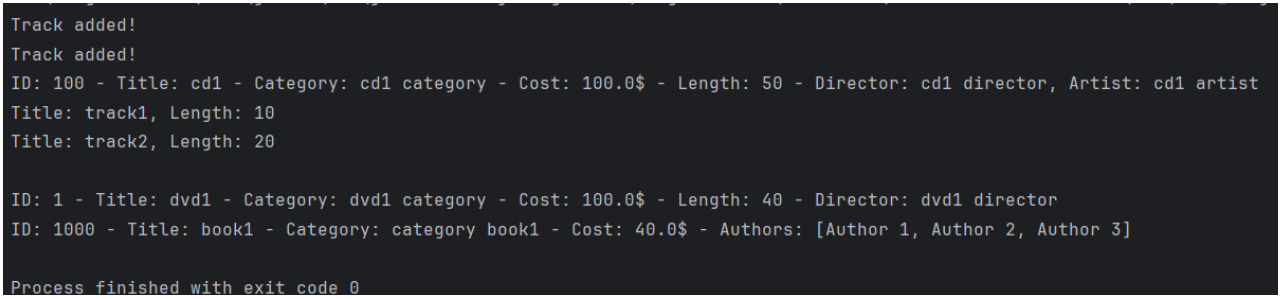
Figure 9.1: Code mô phỏng Polymorphism



```
//Nguyen Hoang Phuc 20225905
public String toString(){  4 overrides  NHP234 *
    return "ID: " + this.getId() + " - Title: " + this.getTitle() + " - Category: " +
            this.getCategory() + " - Cost: " + this.getCost() + "$";
}
```

Figure 9.2: Override toString() in Media Class

Kết quả



```
Track added!
Track added!
ID: 100 - Title: cd1 - Category: cd1 category - Cost: 100.0$ - Length: 50 - Director: cd1 director, Artist: cd1 artist
Title: track1, Length: 10
Title: track2, Length: 20

ID: 1 - Title: dvd1 - Category: dvd1 category - Cost: 100.0$ - Length: 40 - Director: dvd1 director
ID: 1000 - Title: book1 - Category: category book1 - Cost: 40.0$ - Authors: [Author 1, Author 2, Author 3]

Process finished with exit code 0
```

Figure 9.3: Result demo Polymorphism

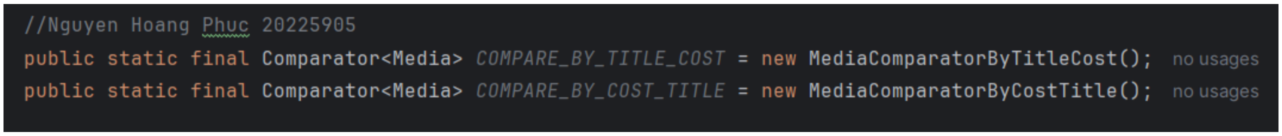
Lớp Media là lớp cơ sở được kế thừa bởi các lớp cụ thể hơn là CompactDisc, DigitalVideoDisc và Book. Khi khởi tạo các đối tượng cd, dvd, book thuộc lớp con rồi gán chúng cho biến kiểu Media, ta áp dụng kỹ thuật gọi là upcasting.

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

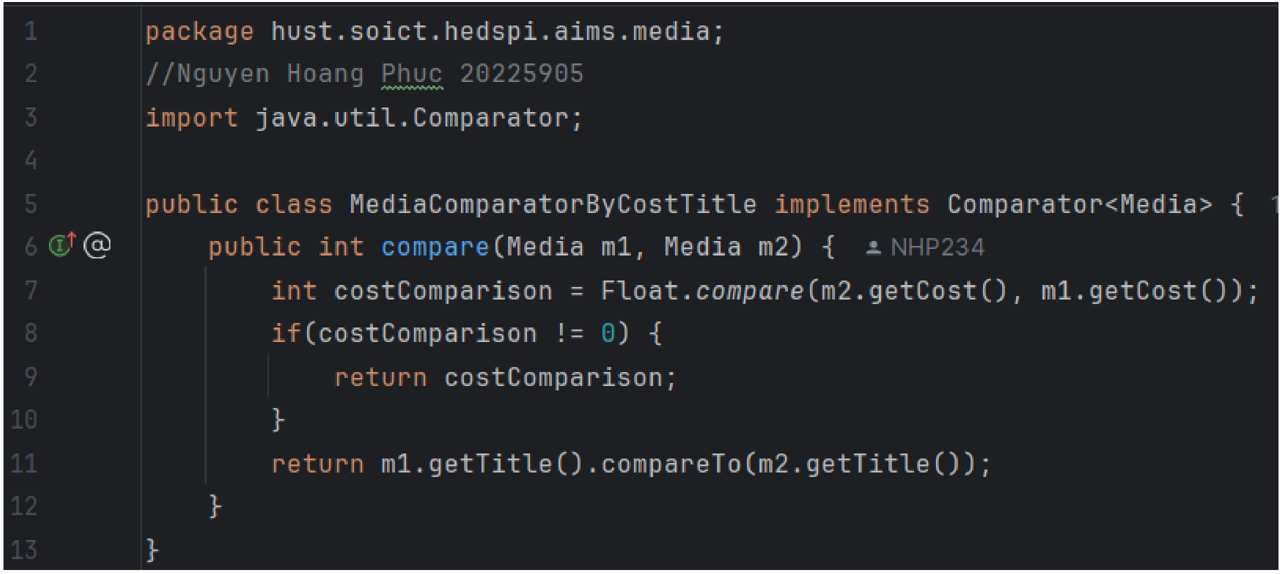
Sắp xếp các media trong giỏ hàng theo hai tiêu chí:

- 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



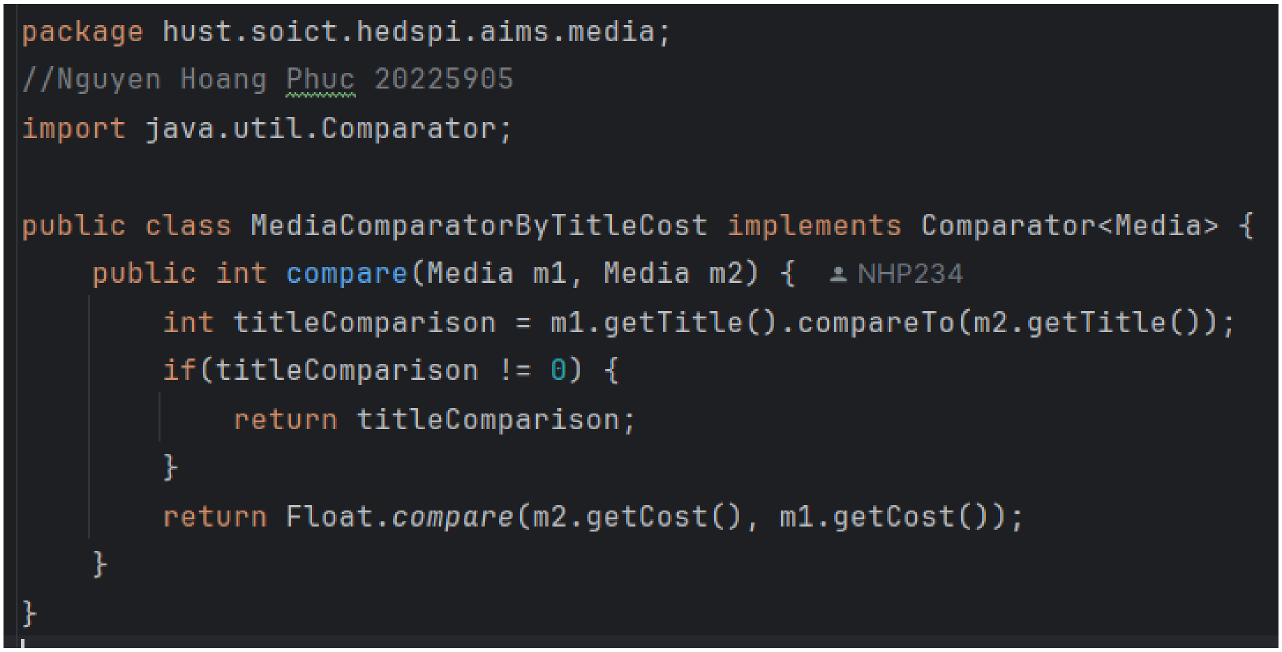
```java
//Nguyen Hoang Phuc 20225905
public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost();
public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle();
```

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



```java
package hust.soict.hedspi.aims.media;
//Nguyen Hoang Phuc 20225905
import java.util.Comparator;

public class MediaComparatorByCostTitle implements Comparator<Media> {
    public int compare(Media m1, Media m2) {
        int costComparison = Float.compare(m2.getCost(), m1.getCost());
        if(costComparison != 0) {
            return costComparison;
        }
        return m1.getTitle().compareTo(m2.getTitle());
    }
}
```

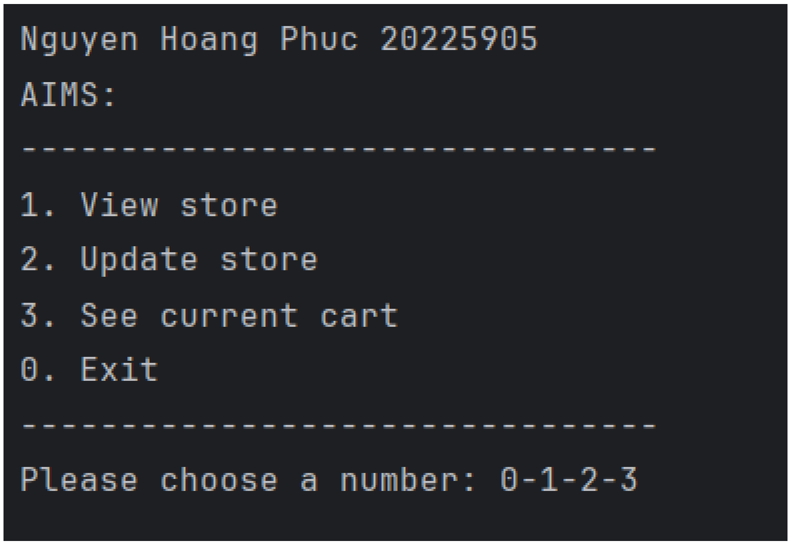
Figure 10.2: MediaComparatorByCostTitle Class



```java
package hust.soict.hedspi.aims.media;
//Nguyen Hoang Phuc 20225905
import java.util.Comparator;

public class MediaComparatorByTitleCost implements Comparator<Media> {
    public int compare(Media m1, Media m2) {  NHP234
        int titleComparison = m1.getTitle().compareTo(m2.getTitle());
        if(titleComparison != 0) {
            return titleComparison;
        }
        return Float.compare(m2.getCost(), m1.getCost());
    }
}
```

Figure 10.3: MediaComparatorByTitleCost Class

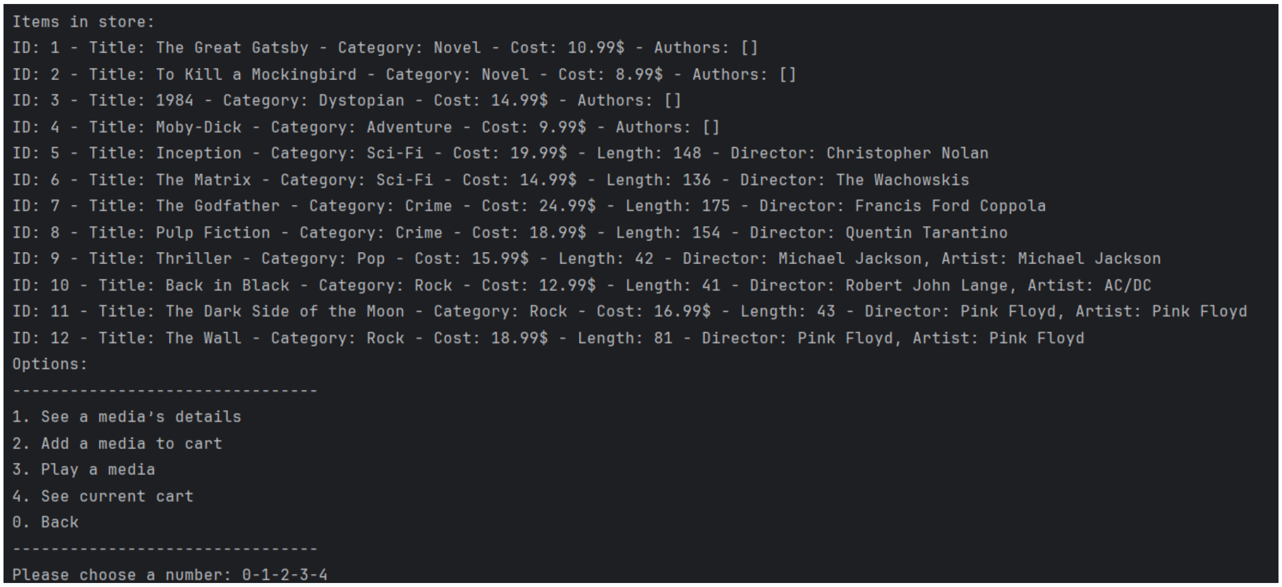
11 Create a complete console application in the Aims class



```
Nguyen Hoang Phuc 20225905
AIMS:
--------------------------------
1. View store
2. Update store
3. See current cart
0. Exit
--------------------------------
Please choose a number: 0-1-2-3
```

Figure 11.1: Màn hình chính

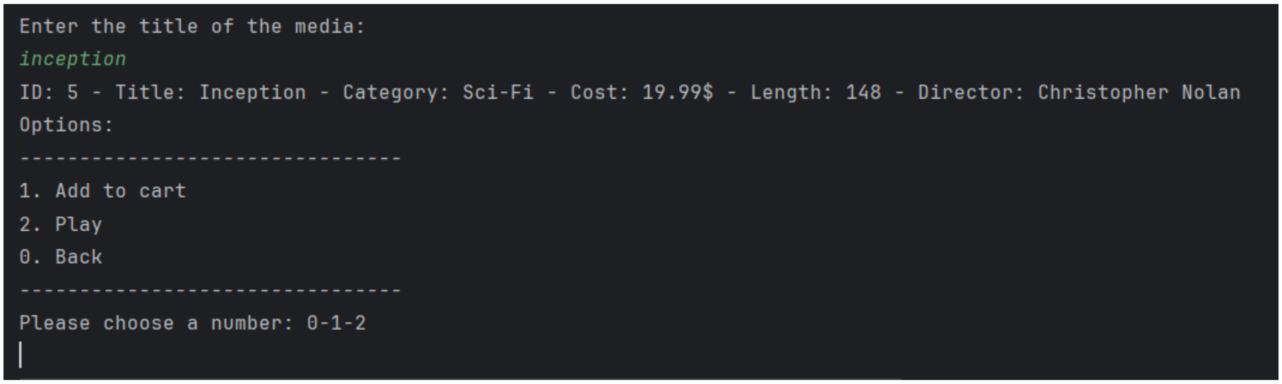
11.1 Người dùng chọn 1: View store



```
Items in store:
ID: 1 - Title: The Great Gatsby - Category: Novel - Cost: 10.99$ - Authors: []
ID: 2 - Title: To Kill a Mockingbird - Category: Novel - Cost: 8.99$ - Authors: []
ID: 3 - Title: 1984 - Category: Dystopian - Cost: 14.99$ - Authors: []
ID: 4 - Title: Moby-Dick - Category: Adventure - Cost: 9.99$ - Authors: []
ID: 5 - Title: Inception - Category: Sci-Fi - Cost: 19.99$ - Length: 148 - Director: Christopher Nolan
ID: 6 - Title: The Matrix - Category: Sci-Fi - Cost: 14.99$ - Length: 136 - Director: The Wachowskis
ID: 7 - Title: The Godfather - Category: Crime - Cost: 24.99$ - Length: 175 - Director: Francis Ford Coppola
ID: 8 - Title: Pulp Fiction - Category: Crime - Cost: 18.99$ - Length: 154 - Director: Quentin Tarantino
ID: 9 - Title: Thriller - Category: Pop - Cost: 15.99$ - Length: 42 - Director: Michael Jackson, Artist: Michael Jackson
ID: 10 - Title: Back in Black - Category: Rock - Cost: 12.99$ - Length: 41 - Director: Robert John Lange, Artist: AC/DC
ID: 11 - Title: The Dark Side of the Moon - Category: Rock - Cost: 16.99$ - Length: 43 - Director: Pink Floyd, Artist: Pink Floyd
ID: 12 - Title: The Wall - Category: Rock - Cost: 18.99$ - Length: 81 - Director: Pink Floyd, Artist: Pink Floyd
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
```

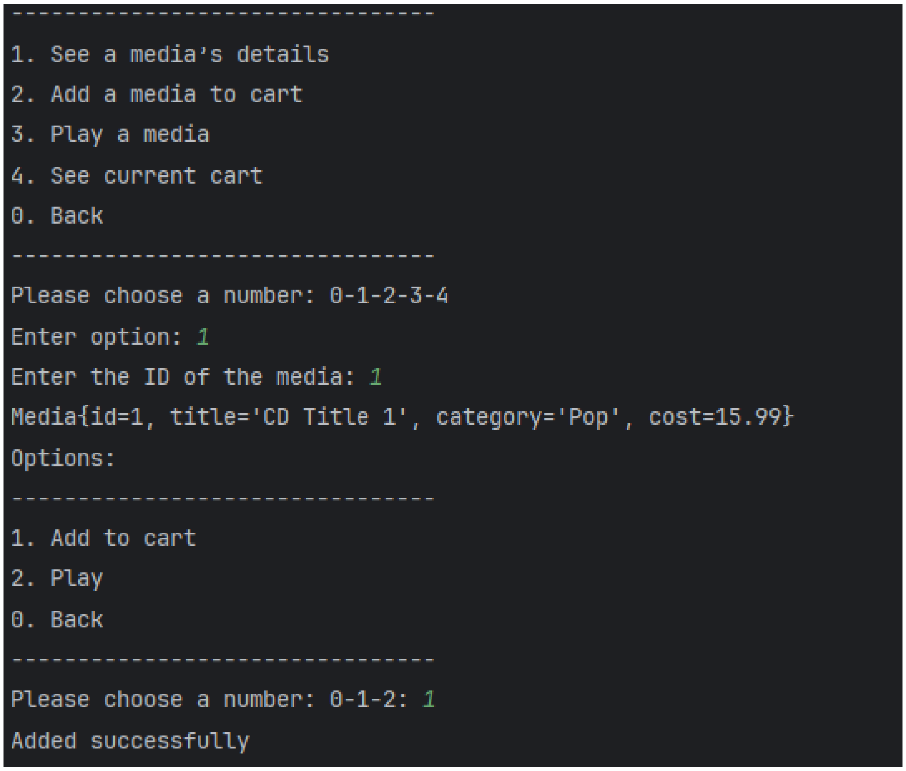
Figure 11.2: Vào Trang View Store

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



```
Enter the title of the media:
inception
ID: 5 - Title: Inception - Category: Sci-Fi - Cost: 19.99$ - Length: 148 - Director: Christopher Nolan
Options:
--------------------------------
1. Add to cart
2. Play
0. Back
--------------------------------
Please choose a number: 0-1-2
```

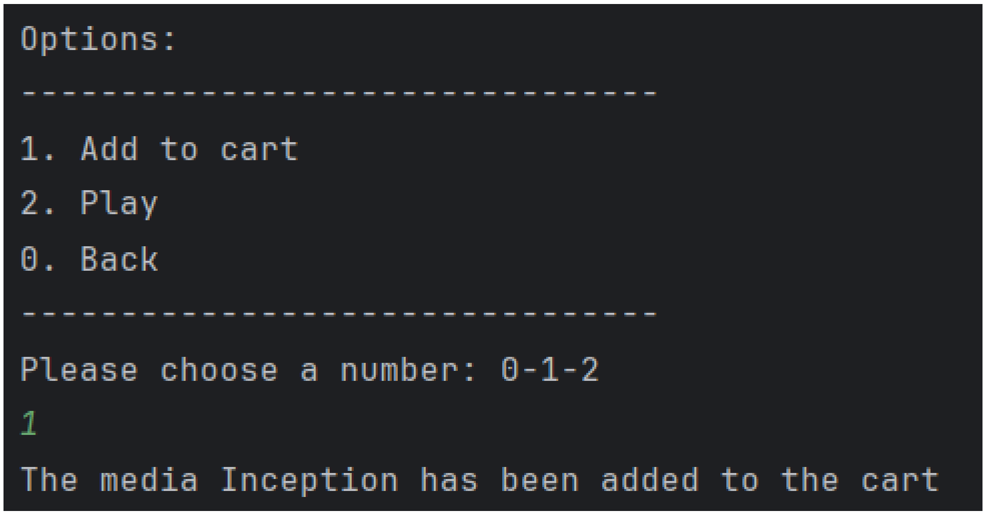
Figure 11.3: See a media's details



```
--------------------------------
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 ID of the media: 1
Media{id=1, title='CD Title 1', category='Pop', cost=15.99}
Options:
--------------------------------
1. Add to cart
2. Play
0. Back
--------------------------------
Please choose a number: 0-1-2: 1
Added successfully
```

Figure 11.4: Thêm vào Cart

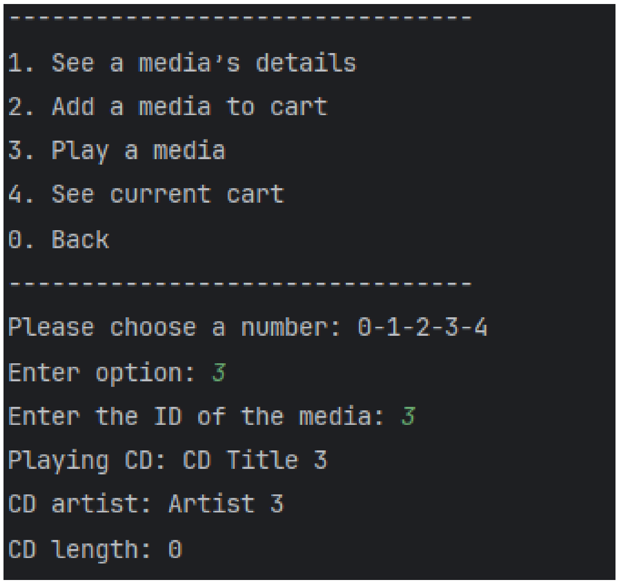
11.1.2 Người dùng chọn 2: Add a media to the cart



```
Options:
--------------------------------
1. Add to cart
2. Play
0. Back
--------------------------------
Please choose a number: 0-1-2
1
The media Inception has been added to the cart
```

Figure 11.5: Thêm media vào Cart

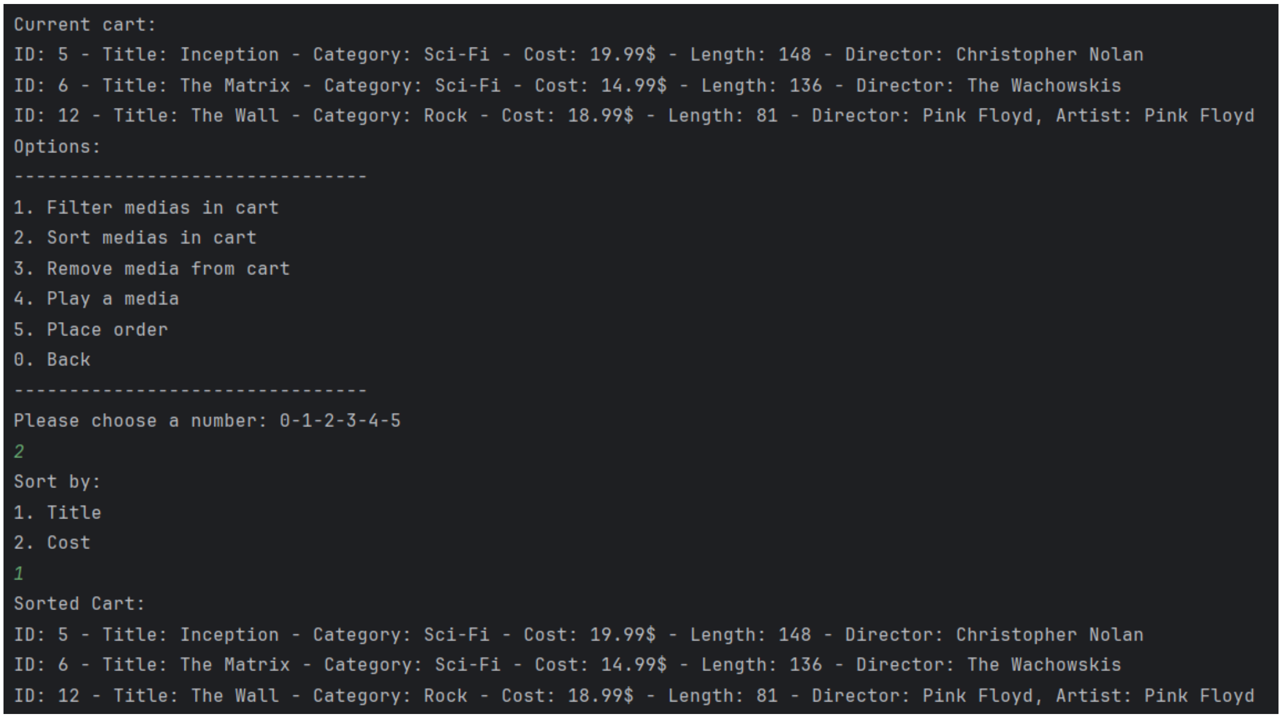
11.1.3 Người dùng chọn 3: Play a media



```
--------------------------------
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 ID of the media: 3
Playing CD: CD Title 3
CD artist: Artist 3
CD length: 0
```

Figure 11.6: Play a media

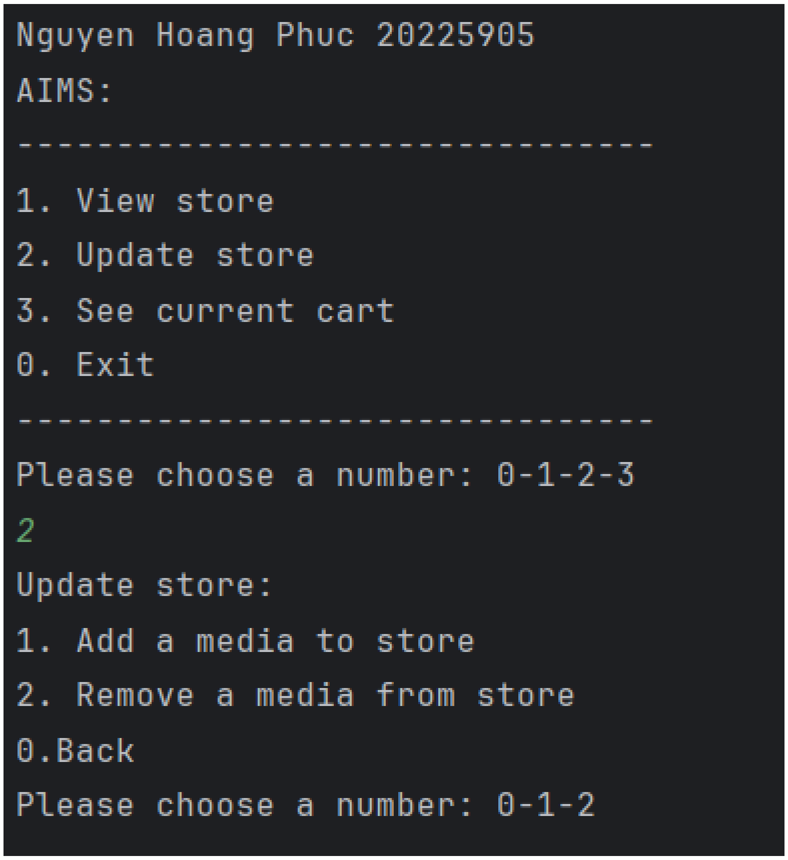
11.1.4 Người dùng chọn 4: See current cart



```
Current cart:
ID: 5 - Title: Inception - Category: Sci-Fi - Cost: 19.99$ - Length: 148 - Director: Christopher Nolan
ID: 6 - Title: The Matrix - Category: Sci-Fi - Cost: 14.99$ - Length: 136 - Director: The Wachowskis
ID: 12 - Title: The Wall - Category: Rock - Cost: 18.99$ - Length: 81 - Director: Pink Floyd, Artist: Pink Floyd
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
2
Sort by:
1. Title
2. Cost
1
Sorted Cart:
ID: 5 - Title: Inception - Category: Sci-Fi - Cost: 19.99$ - Length: 148 - Director: Christopher Nolan
ID: 6 - Title: The Matrix - Category: Sci-Fi - Cost: 14.99$ - Length: 136 - Director: The Wachowskis
ID: 12 - Title: The Wall - Category: Rock - Cost: 18.99$ - Length: 81 - Director: Pink Floyd, Artist: Pink Floyd
```

Figure 11.7: See current cart after sort

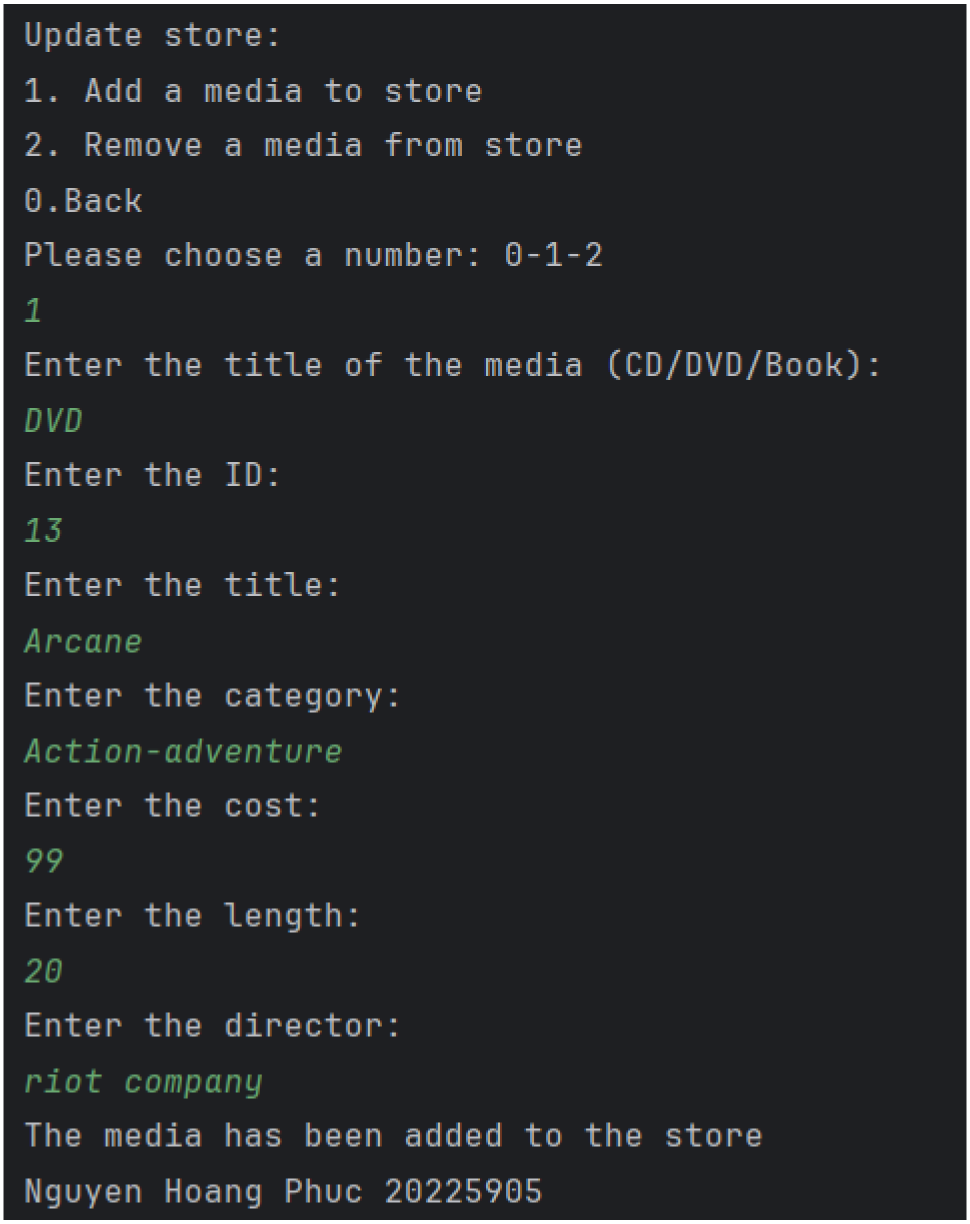
11.2 Người dùng chọn 2: Update store



```
Nguyen Hoang Phuc 20225905
AIMS:
--------------------------------
1. View store
2. Update store
3. See current cart
0. Exit
--------------------------------
Please choose a number: 0-1-2-3
2
Update store:
1. Add a media to store
2. Remove a media from store
0.Back
Please choose a number: 0-1-2
```

Figure 11.8: Vào Trang Update Store

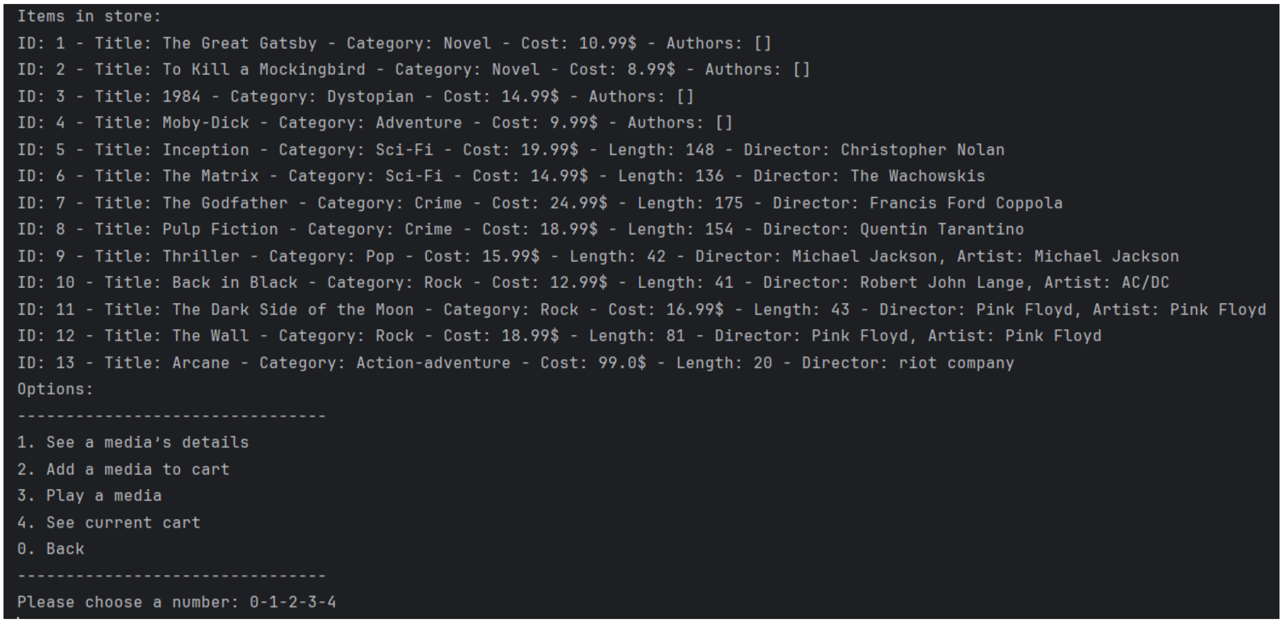
11.2.1 Người dùng chọn 1: Add a media to the store



```
Update store:
1. Add a media to store
2. Remove a media from store
0.Back
Please choose a number: 0-1-2
1
Enter the title of the media (CD/DVD/Book):
DVD
Enter the ID:
13
Enter the title:
Arcane
Enter the category:
Action-adventure
Enter the cost:
99
Enter the length:
20
Enter the director:
riot company
The media has been added to the store
Nguyen Hoang Phuc 20225905
```

Figure 11.9: Add a media to store

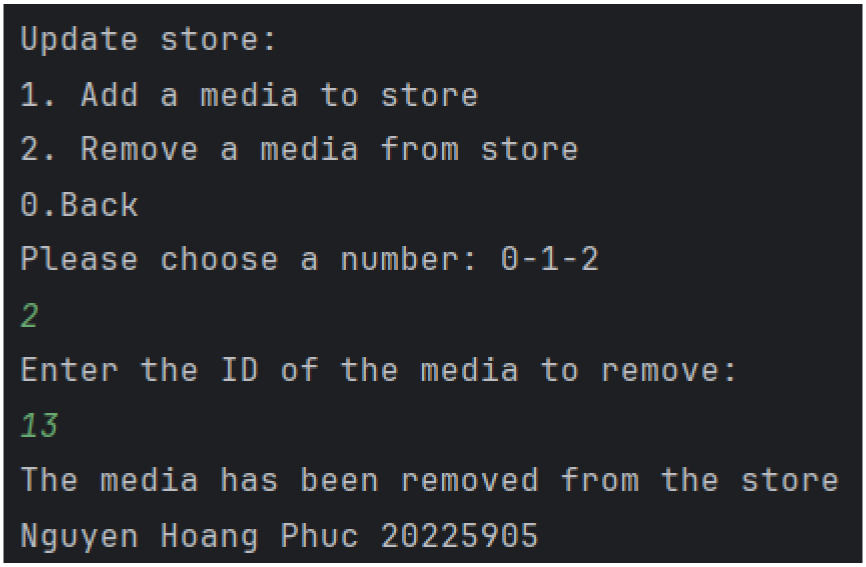
=> Kết quả sau khi thêm



```
Items in store:
ID: 1 - Title: The Great Gatsby - Category: Novel - Cost: 10.99$ - Authors: []
ID: 2 - Title: To Kill a Mockingbird - Category: Novel - Cost: 8.99$ - Authors: []
ID: 3 - Title: 1984 - Category: Dystopian - Cost: 14.99$ - Authors: []
ID: 4 - Title: Moby-Dick - Category: Adventure - Cost: 9.99$ - Authors: []
ID: 5 - Title: Inception - Category: Sci-Fi - Cost: 19.99$ - Length: 148 - Director: Christopher Nolan
ID: 6 - Title: The Matrix - Category: Sci-Fi - Cost: 14.99$ - Length: 136 - Director: The Wachowskis
ID: 7 - Title: The Godfather - Category: Crime - Cost: 24.99$ - Length: 175 - Director: Francis Ford Coppola
ID: 8 - Title: Pulp Fiction - Category: Crime - Cost: 18.99$ - Length: 154 - Director: Quentin Tarantino
ID: 9 - Title: Thriller - Category: Pop - Cost: 15.99$ - Length: 42 - Director: Michael Jackson, Artist: Michael Jackson
ID: 10 - Title: Back in Black - Category: Rock - Cost: 12.99$ - Length: 41 - Director: Robert John Lange, Artist: AC/DC
ID: 11 - Title: The Dark Side of the Moon - Category: Rock - Cost: 16.99$ - Length: 43 - Director: Pink Floyd, Artist: Pink Floyd
ID: 12 - Title: The Wall - Category: Rock - Cost: 18.99$ - Length: 81 - Director: Pink Floyd, Artist: Pink Floyd
ID: 13 - Title: Arcane - Category: Action-adventure - Cost: 99.0$ - Length: 20 - Director: riot company
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
```

Figure 11.10: Result after add media to store

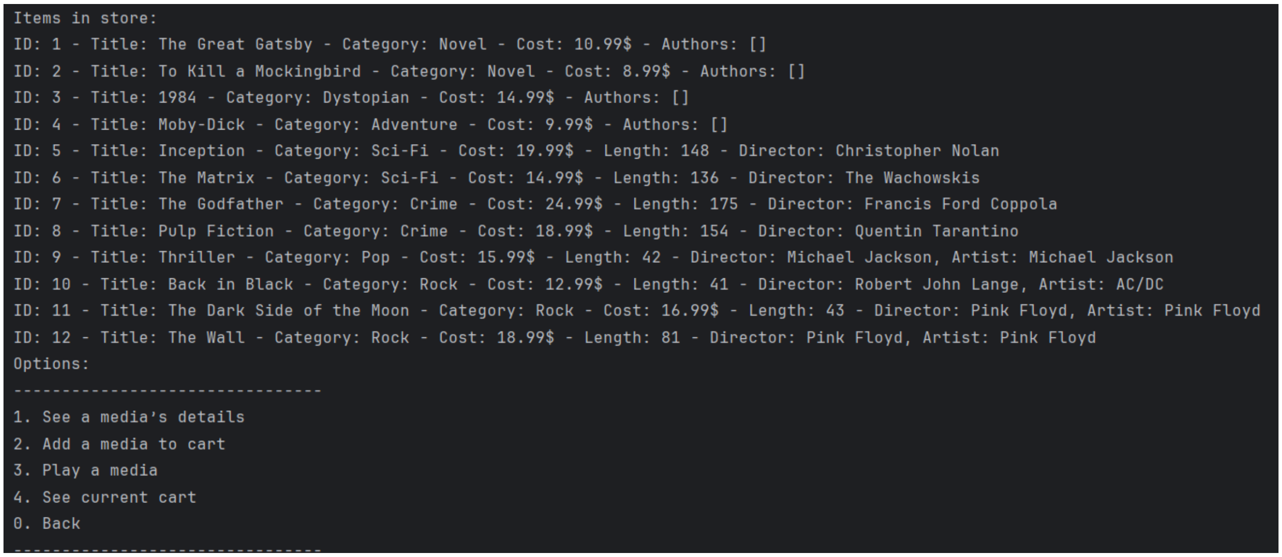
11.2.2 Người dùng chọn 2: Remove a media from the store



```
Update store:
1. Add a media to store
2. Remove a media from store
0.Back
Please choose a number: 0-1-2
2
Enter the ID of the media to remove:
13
The media has been removed from the store
Nguyen Hoang Phuc 20225905
```

Figure 11.11: Remove a media from the store

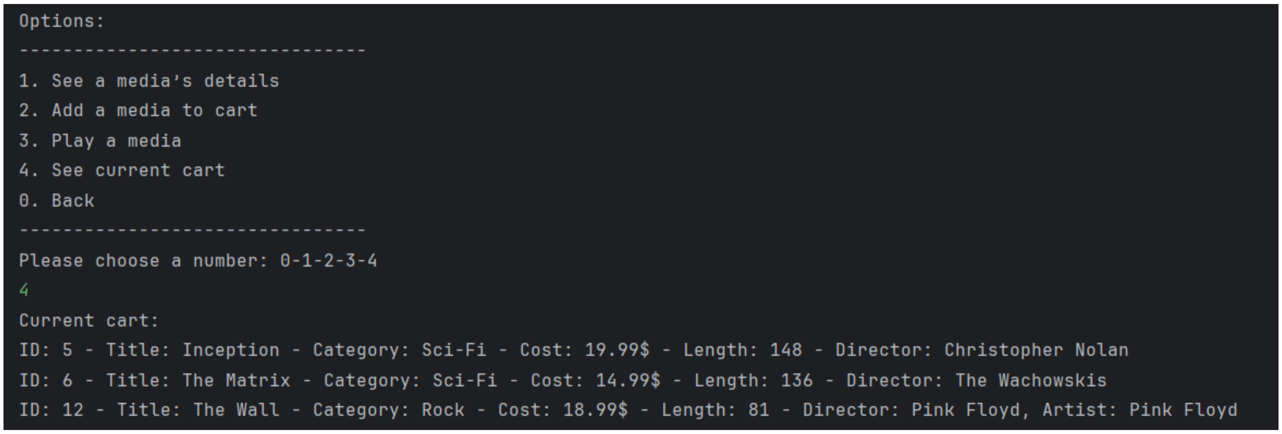
=> Kết quả sau khi remove



```
Items in store:
ID: 1 - Title: The Great Gatsby - Category: Novel - Cost: 10.99$ - Authors: []
ID: 2 - Title: To Kill a Mockingbird - Category: Novel - Cost: 8.99$ - Authors: []
ID: 3 - Title: 1984 - Category: Dystopian - Cost: 14.99$ - Authors: []
ID: 4 - Title: Moby-Dick - Category: Adventure - Cost: 9.99$ - Authors: []
ID: 5 - Title: Inception - Category: Sci-Fi - Cost: 19.99$ - Length: 148 - Director: Christopher Nolan
ID: 6 - Title: The Matrix - Category: Sci-Fi - Cost: 14.99$ - Length: 136 - Director: The Wachowskis
ID: 7 - Title: The Godfather - Category: Crime - Cost: 24.99$ - Length: 175 - Director: Francis Ford Coppola
ID: 8 - Title: Pulp Fiction - Category: Crime - Cost: 18.99$ - Length: 154 - Director: Quentin Tarantino
ID: 9 - Title: Thriller - Category: Pop - Cost: 15.99$ - Length: 42 - Director: Michael Jackson, Artist: Michael Jackson
ID: 10 - Title: Back in Black - Category: Rock - Cost: 12.99$ - Length: 41 - Director: Robert John Lange, Artist: AC/DC
ID: 11 - Title: The Dark Side of the Moon - Category: Rock - Cost: 16.99$ - Length: 43 - Director: Pink Floyd, Artist: Pink Floyd
ID: 12 - Title: The Wall - Category: Rock - Cost: 18.99$ - Length: 81 - Director: Pink Floyd, Artist: Pink Floyd
Options:
--------------------------------
1. See a media’s details
2. Add a media to cart
3. Play a media
4. See current cart
0. Back
--------------------------------
```

Figure 11.12: Result after remove a media

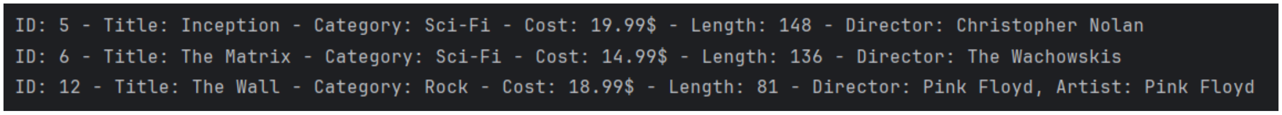
11.3 Người dùng chọn 3: See current cart



```
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
4
Current cart:
ID: 5 - Title: Inception - Category: Sci-Fi - Cost: 19.99$ - Length: 148 - Director: Christopher Nolan
ID: 6 - Title: The Matrix - Category: Sci-Fi - Cost: 14.99$ - Length: 136 - Director: The Wachowskis
ID: 12 - Title: The Wall - Category: Rock - Cost: 18.99$ - Length: 81 - Director: Pink Floyd, Artist: Pink Floyd
```

Figure 11.13: Vào trang See current cart

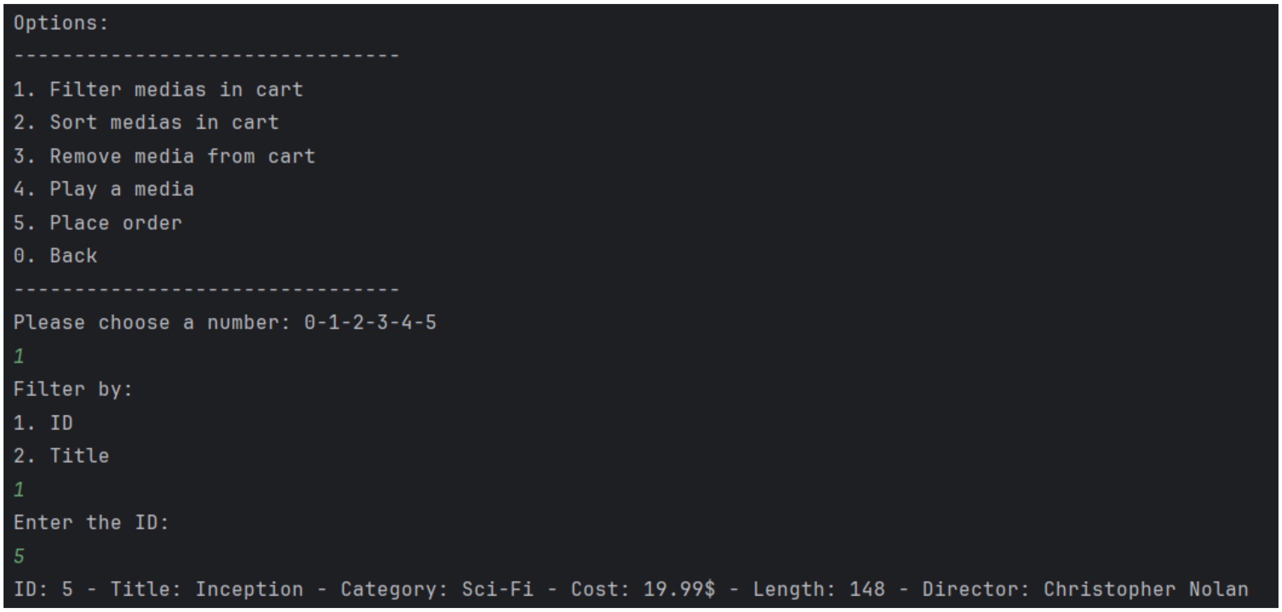
Giả sử lúc này trong Cart sẽ có các Media sau



```
ID: 5 - Title: Inception - Category: Sci-Fi - Cost: 19.99$ - Length: 148 - Director: Christopher Nolan
ID: 6 - Title: The Matrix - Category: Sci-Fi - Cost: 14.99$ - Length: 136 - Director: The Wachowskis
ID: 12 - Title: The Wall - Category: Rock - Cost: 18.99$ - Length: 81 - Director: Pink Floyd, Artist: Pink Floyd
```

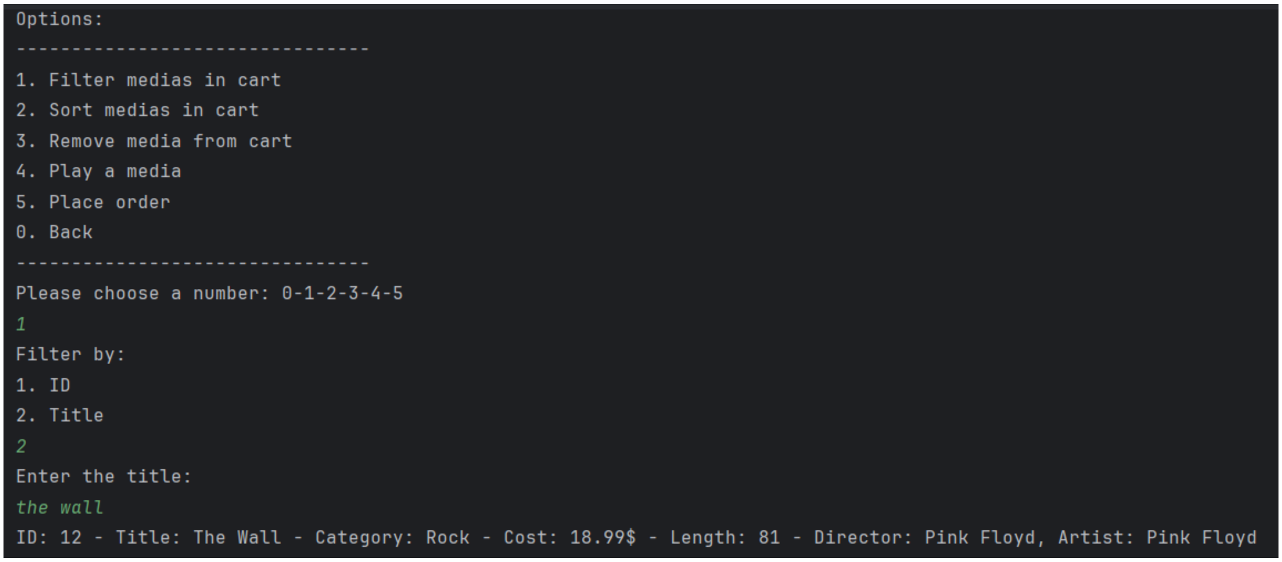
Figure 11.14: Media in Cart

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
1
Filter by:
1. ID
2. Title
1
Enter the ID:
5
ID: 5 - Title: Inception - Category: Sci-Fi - Cost: 19.99$ - Length: 148 - Director: Christopher Nolan
```

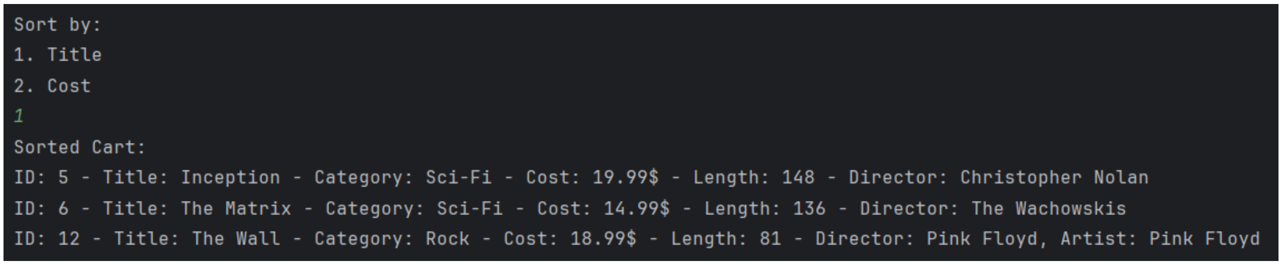
Figure 11.15: Filter Cart By id



```
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
1
Filter by:
1. ID
2. Title
2
Enter the title:
the wall
ID: 12 - Title: The Wall - Category: Rock - Cost: 18.99$ - Length: 81 - Director: Pink Floyd, Artist: Pink Floyd
```

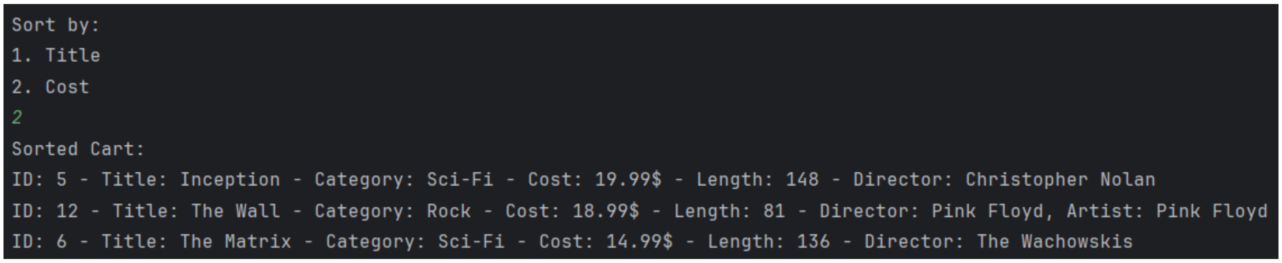
Figure 11.16: Filter Cart By Title

11.3.2 Người dùng chọn 2: Sort medias in cart



```
Sort by:
1. Title
2. Cost
1
Sorted Cart:
ID: 5 - Title: Inception - Category: Sci-Fi - Cost: 19.99$ - Length: 148 - Director: Christopher Nolan
ID: 6 - Title: The Matrix - Category: Sci-Fi - Cost: 14.99$ - Length: 136 - Director: The Wachowskis
ID: 12 - Title: The Wall - Category: Rock - Cost: 18.99$ - Length: 81 - Director: Pink Floyd, Artist: Pink Floyd
```

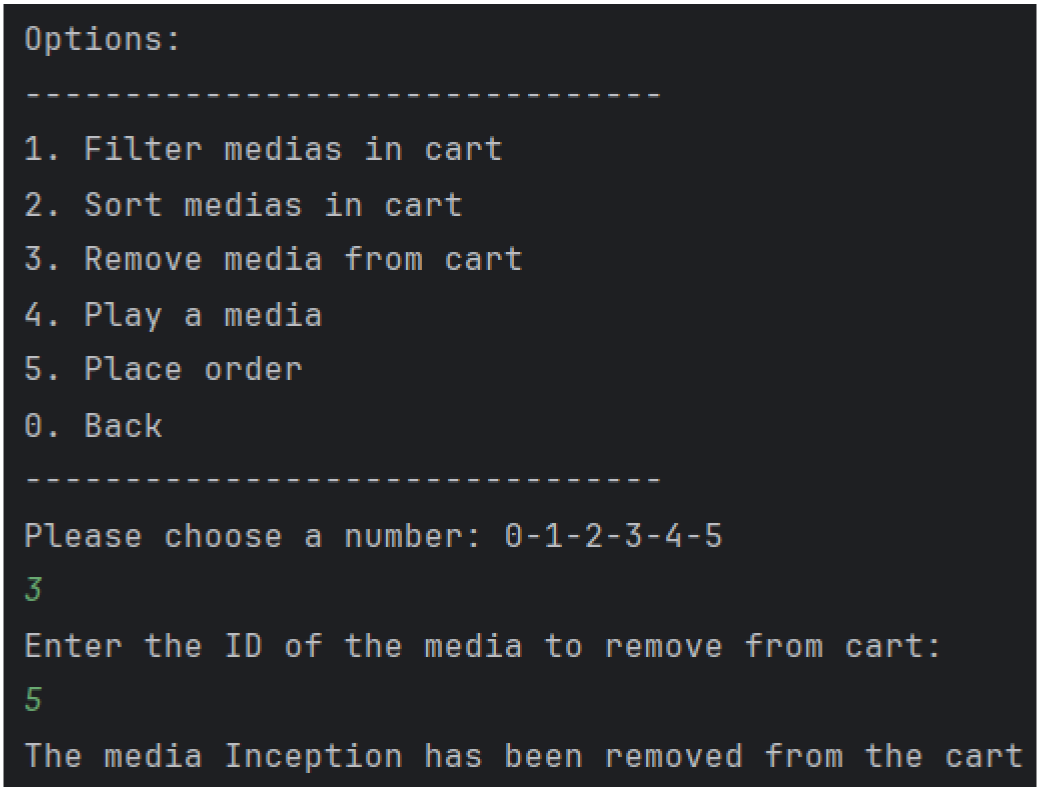
Figure 11.17: Sort Cart By Title



```
Sort by:
1. Title
2. Cost
2
Sorted Cart:
ID: 5 - Title: Inception - Category: Sci-Fi - Cost: 19.99$ - Length: 148 - Director: Christopher Nolan
ID: 12 - Title: The Wall - Category: Rock - Cost: 18.99$ - Length: 81 - Director: Pink Floyd, Artist: Pink Floyd
ID: 6 - Title: The Matrix - Category: Sci-Fi - Cost: 14.99$ - Length: 136 - Director: The Wachowskis
```

Figure 11.18: Sort Cart By Cost

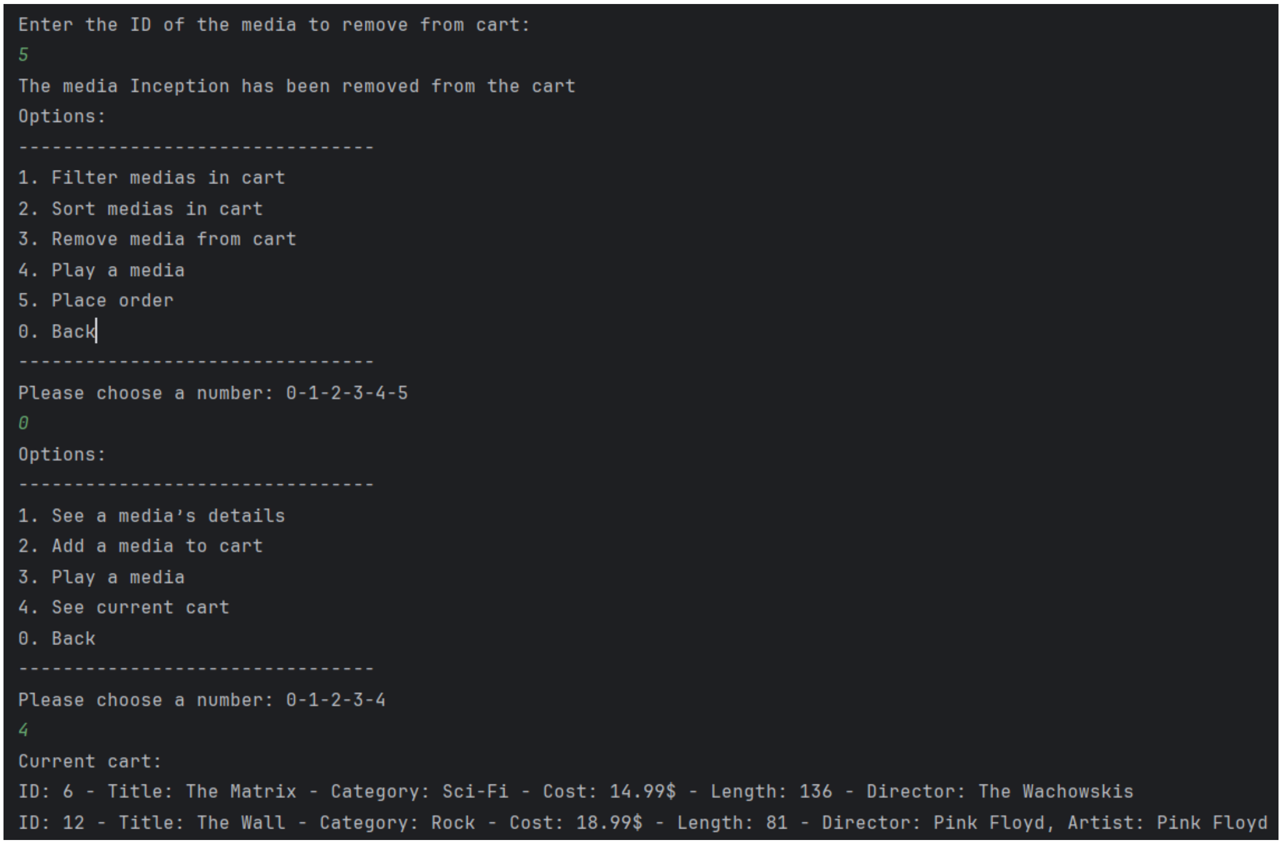
11.3.3 Người dùng chọn 3: Remove media from cart



```
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
3
Enter the ID of the media to remove from cart:
5
The media Inception has been removed from the cart
```

Figure 11.19: Remove media by id

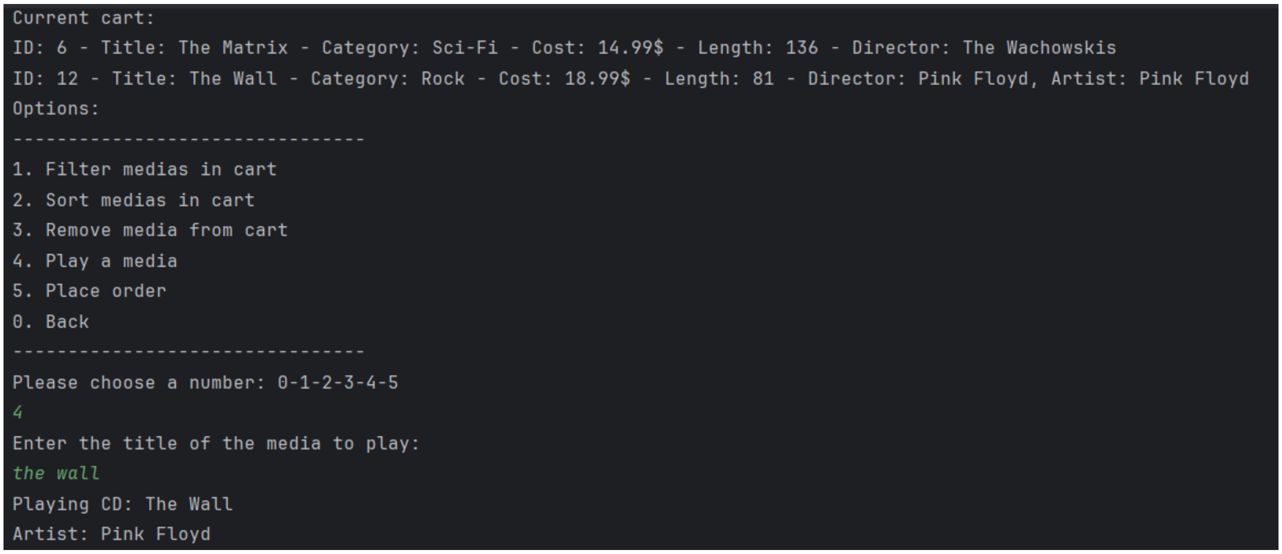
=> Kết quả



```
Enter the ID of the media to remove from cart:
5
The media Inception has been removed from the 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
0
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
4
Current cart:
ID: 6 - Title: The Matrix - Category: Sci-Fi - Cost: 14.99$ - Length: 136 - Director: The Wachowskis
ID: 12 - Title: The Wall - Category: Rock - Cost: 18.99$ - Length: 81 - Director: Pink Floyd, Artist: Pink Floyd
```

Figure 11.20: Result after remove media in cart by id

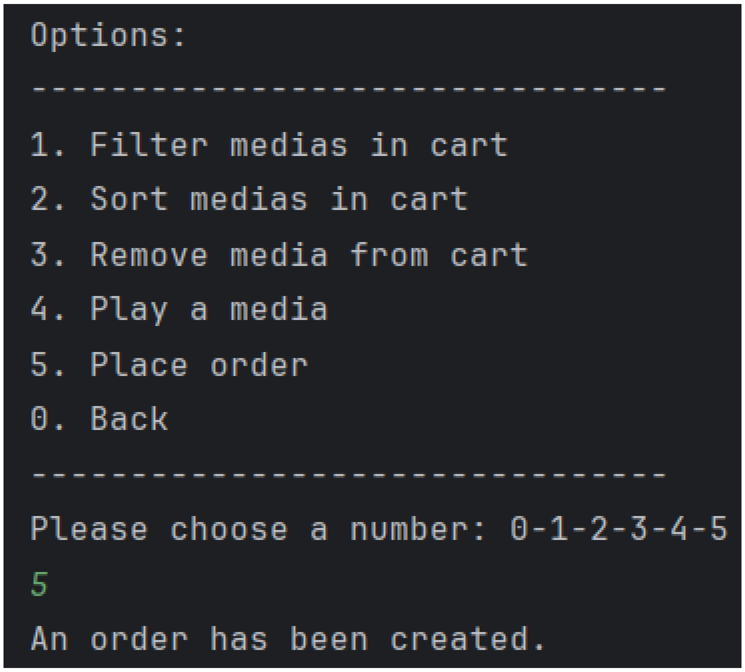
11.3.4 Người dùng chọn 4: Play a media



```
Current cart:
ID: 6 - Title: The Matrix - Category: Sci-Fi - Cost: 14.99$ - Length: 136 - Director: The Wachowskis
ID: 12 - Title: The Wall - Category: Rock - Cost: 18.99$ - Length: 81 - Director: Pink Floyd, Artist: Pink Floyd
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
4
Enter the title of the media to play:
the wall
Playing CD: The Wall
Artist: Pink Floyd
```

Figure 11.21: Play a media in cart

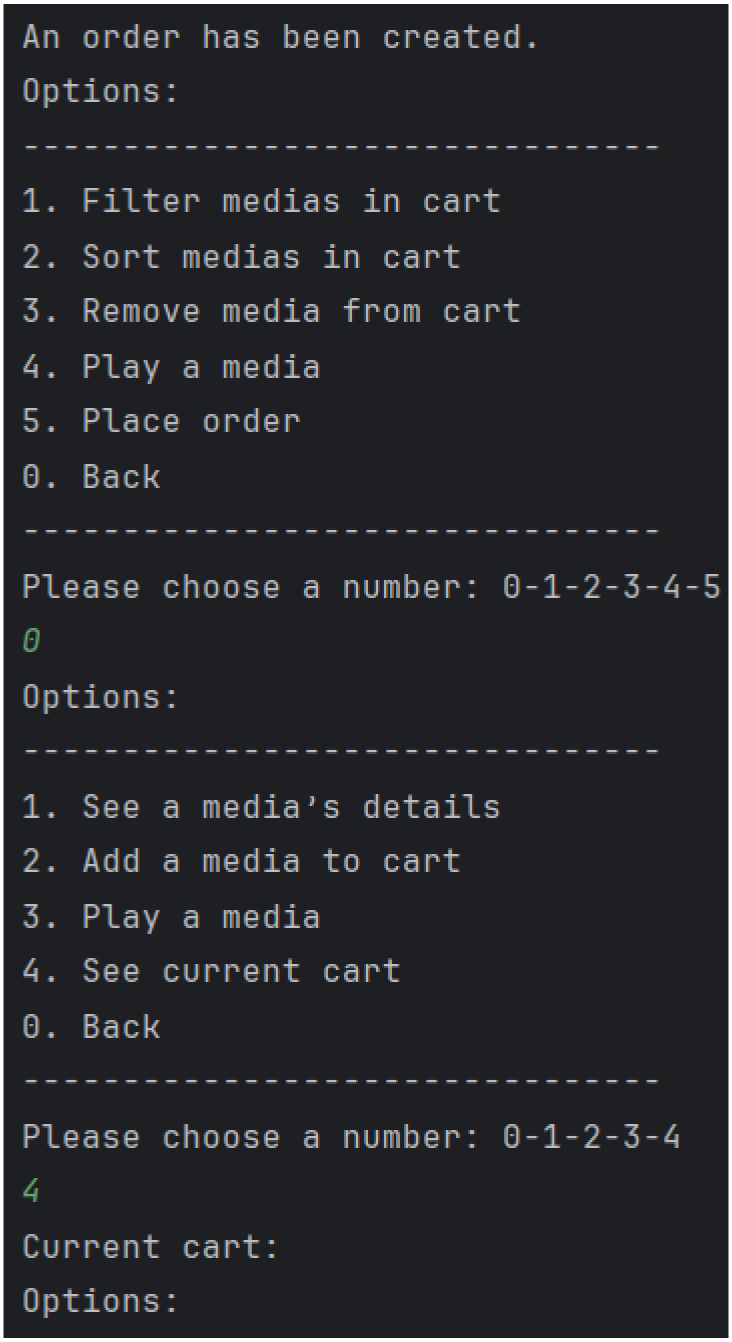
11.3.5 Người dùng chọn 5: Place order



```
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
5
An order has been created.
```

Figure 11.22: Order

=> Kết quả sau khi order



```
An order has been created.
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
0
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
4
Current cart:
Options:
```

Figure 11.23: Result after order

12 Class Diagram

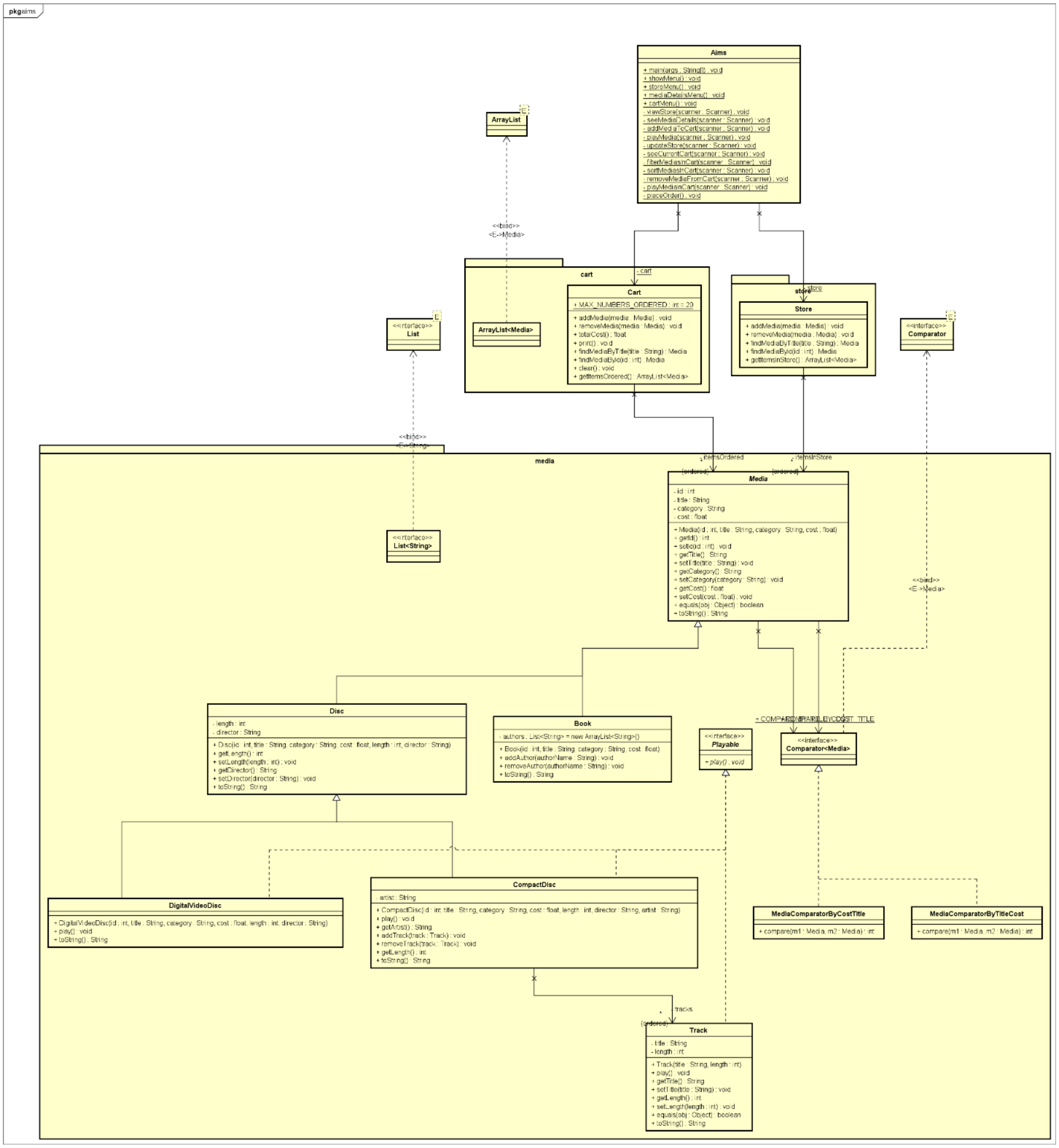

pkg aims
Aims
+ main(args : String[]) : void
+ showMenu() : void
+ storeMenu() : void
+ mediaDetailsMenu() : void
+ cartMenu() : void
- viewStore(scanner : Scanner) : void
- seeMediaDetails(scanner : Scanner) : void
- addMediaToCart(scanner : Scanner) : void
- playMedia(scanner : Scanner) : void
- updateStore(scanner : Scanner) : void
- seeCurrentCart(scanner : Scanner) : void
- filterMediasInCart(scanner : Scanner) : void
- sortMediasInCart(scanner : Scanner) : void
- removeMediaFromCart(scanner : Scanner) : void
- playMediaInCart(scanner : Scanner) : void
- placeOrder() : void
ArrayList
E
<<bind>>
<E->Media>
cart
- cart
Cart
+ MAX_NUMBERS_ORDERED : int = 20
+ addMedia(media : Media) : void
+ removeMedia(media : Media) : void
+ totalCost() : float
+ print() : void
+ findMediaByTitle(title : String) : Media
+ findMediaById(id : int) : Media
+ clear() : void
+ getItemsOrdered() : ArrayList<Media>
ArrayList<Media>
store
- store
Store
+ addMedia(media : Media) : void
+ removeMedia(media : Media) : void
+ findMediaByTitle(title : String) : Media
+ findMediaById(id : int) : Media
+ getItemsInStore() : ArrayList<Media>
<<interface>>
List
E
<<interface>>
Comparator
E
<<bind>>
<E->String>
media
* - itemsOrdered
{ordered}
* - itemsInStore
{ordered}
Media
- id : int
- title : String
- category : String
- cost : float
+ Media(id : int, title : String, category : String, cost : float)
+ getId() : int
+ setId(id : int) : void
+ getTitle() : String
+ setTitle(title : String) : void
+ getCategory() : String
+ setCategory(category : String) : void
+ getCost() : float
+ setCost(cost : float) : void
+ equals(obj : Object) : boolean
+ toString() : String
<<interface>>
List<String>
<<bind>>
<E->Media>
+ COMPARE_BY_COST_TITLE
+ COMPARE_BY_TITLE_COST
Disc
- length : int
- director : String
+ Disc(id : int, title : String, category : String, cost : float, length : int, director : String)
+ getLength() : int
+ setLength(length : int) : void
+ getDirector() : String
+ setDirector(director : String) : void
+ toString() : String
Book
- authors : List<String> = new ArrayList<String>()
+ Book(id : int, title : String, category : String, cost : float)
+ addAuthor(authorName : String) : void
+ removeAuthor(authorName : String) : void
+ toString() : String
<<interface>>
Playable
+ play() : void
<<interface>>
Comparator<Media>
DigitalVideoDisc
+ DigitalVideoDisc(id : int, title : String, category : String, cost : float, length : int, director : String)
+ play() : void
+ toString() : String
CompactDisc
- artist : String
+ CompactDisc(id : int, title : String, category : String, cost : float, length : int, director : String, artist : String)
+ play() : void
+ getArtist() : String
+ addTrack(track : Track) : void
+ removeTrack(track : Track) : void
+ getLength() : int
+ toString() : String
MediaComparatorByCostTitle
+ compare(m1 : Media, m2 : Media) : int
MediaComparatorByTitleCost
+ compare(m1 : Media, m2 : Media) : int
* - tracks
{ordered}
Track
- title : String
- length : int
+ Track(title : String, length : int)
+ play() : void
+ getTitle() : String
+ setTitle(title : String) : void
+ getLength() : int
+ setLength(length : int) : void
+ equals(obj : Object) : boolean
+ toString() : String

Figure 12.1: Class Diagram

13 UseCase Diagram

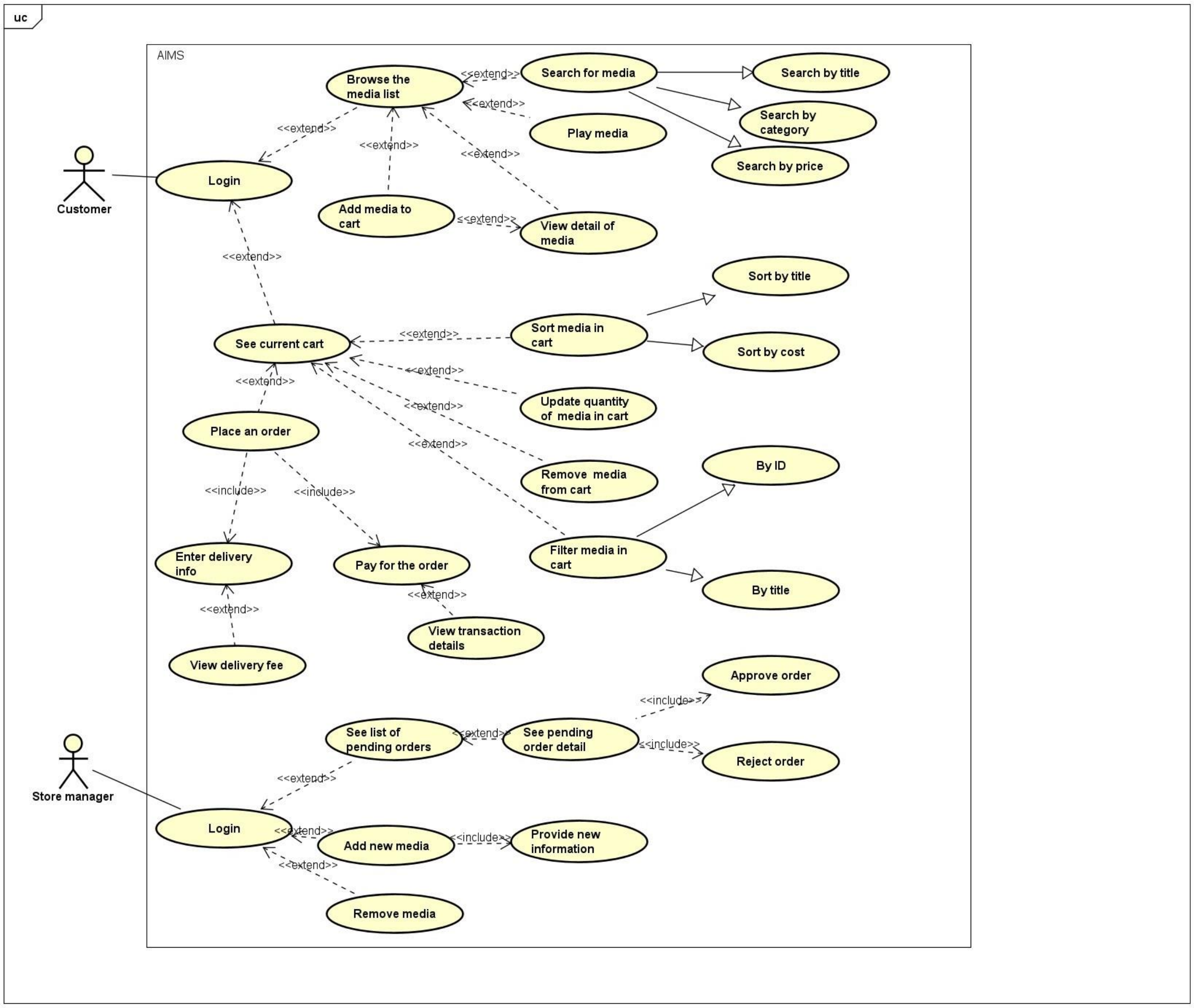
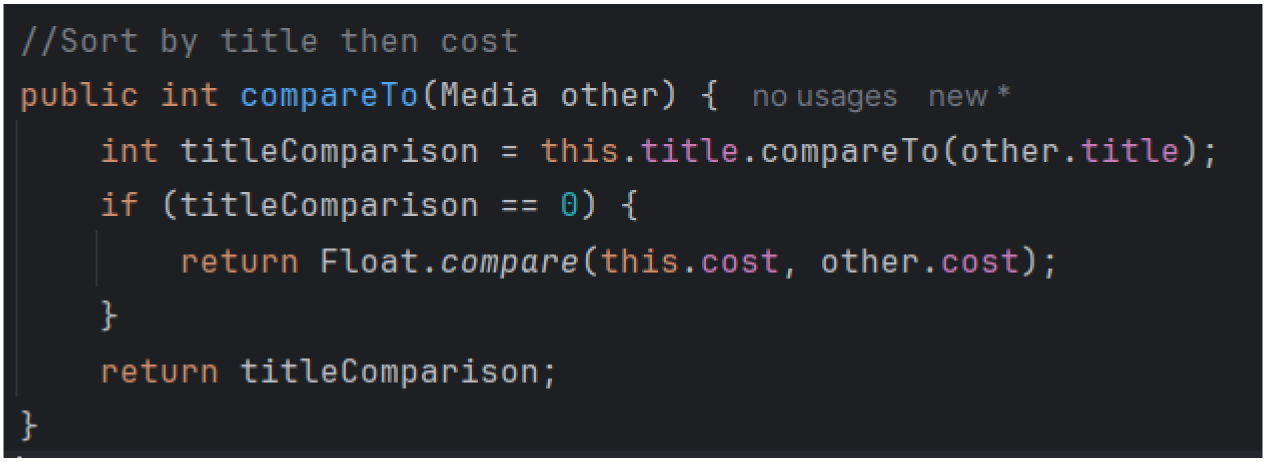


Figure 13.1: UseCase Diagram

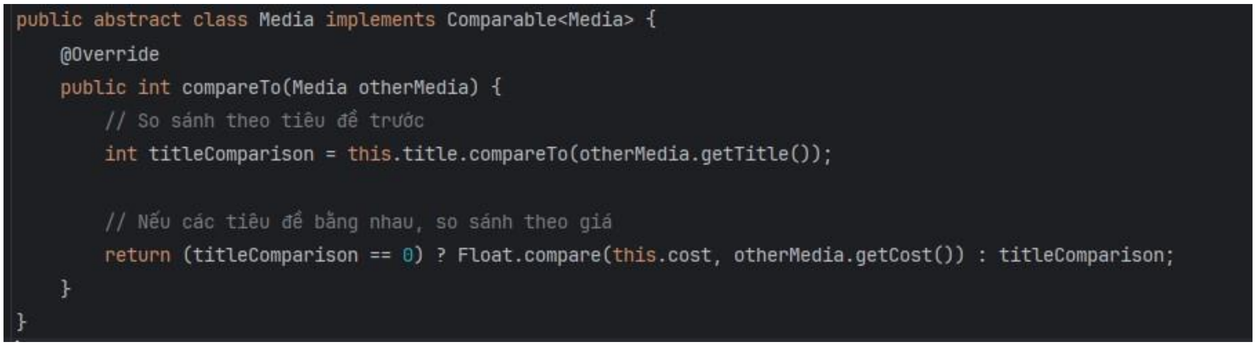
14 Answer questions

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.



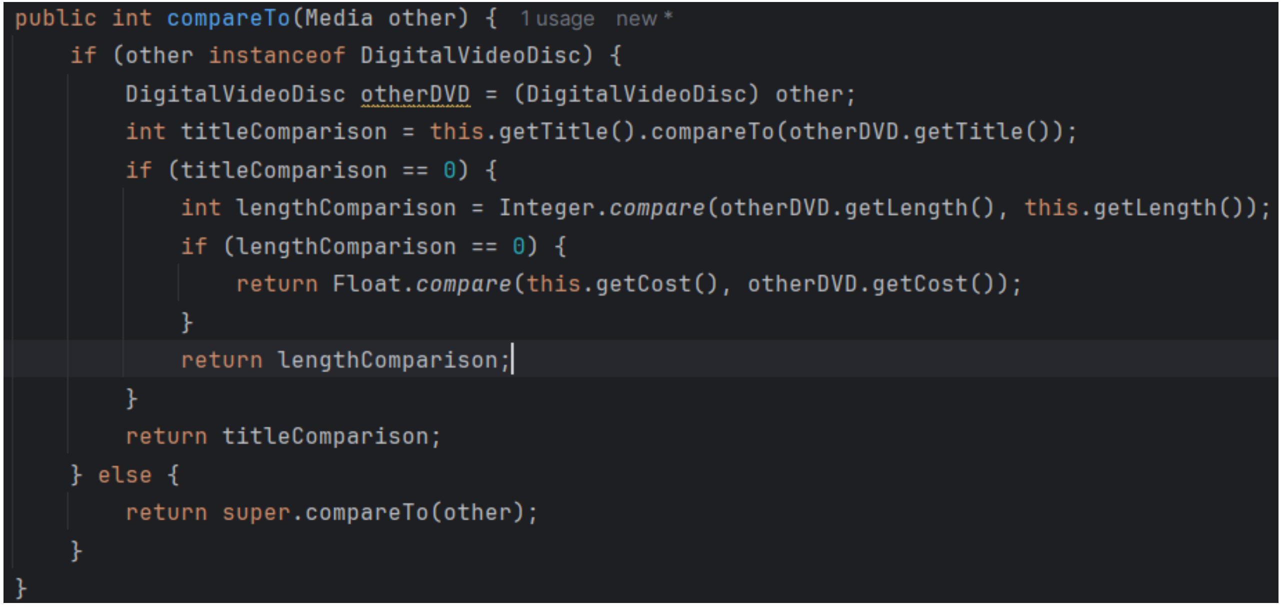
```
//Sort by title then cost
public int compareTo(Media other) {  no usages   new *
    int titleComparison = this.title.compareTo(other.title);
    if (titleComparison == 0) {
        return Float.compare(this.cost, other.cost);
    }
    return titleComparison;
}
```

Figure 14.1: Triển khai Comparable trong lớp trừu tượng Media



```
public abstract class Media implements Comparable<Media> {
    @Override
    public int compareTo(Media otherMedia) {
        // So sánh theo tiêu đề trước
        int titleComparison = this.title.compareTo(otherMedia.getTitle());

        // Nếu các tiêu đề bằng nhau, so sánh theo giá
        return (titleComparison == 0) ? Float.compare(this.cost, otherMedia.getCost()) : titleComparison;
    }
}
```

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



```
public int compareTo(Media other) {  1 usage   new *
    if (other instanceof DigitalVideoDisc) {
        DigitalVideoDisc otherDVD = (DigitalVideoDisc) other;
        int titleComparison = this.getTitle().compareTo(otherDVD.getTitle());
        if (titleComparison == 0) {
            int lengthComparison = Integer.compare(otherDVD.getLength(), this.getLength());
            if (lengthComparison == 0) {
                return Float.compare(this.getCost(), otherDVD.getCost());
            }
            return lengthComparison;
        }
        return titleComparison;
    } else {
        return super.compareTo(other);
    }
}
```

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.

Không thể có 2 tiêu chí sắp xếp khi chúng ta sử dụng cách tiếp cận giao diện Comparable.