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

BÁO CÁO THỰC HÀNH  
 **IT1130-744360-2024.1**

BÀI THỰC HÀNH 4

Họ và tên sv: Nguyễn Đức Bình

MSSV:20225791  
 Lớp: **K67 Việt Nhật 03**

GVHD: Lê Thị Hoa

TA: **Bùi Trọng Dũng**

Hà Nội 11/2024

BÁO CÁO THỰC HÀNH LAB 4 LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

1. [Create the Book class 4](#_bookmark0)
2. [Creating the abstract Media class 6](#_bookmark3)
3. [Creating the CompactDisc class 8](#_bookmark6)
   1. [Create the Disc class extending the Media clas 8](#_bookmark7)
   2. [Create the Track class which models a track on a compact disc and will store information](#_bookmark11) [incuding the title and length of the track. 10](#_bookmark11)
   3. [Open the CompactDisc class 11](#_bookmark13)
4. [Create the Playable interface 13](#_bookmark16)
5. [Update the Cart class to work with Media 14](#_bookmark21)
6. [Update the Store class to work with Media 18](#_bookmark26)
7. [Constructors of whole classes and parent classes 20](#_bookmark29)
8. [Unique item in a list 21](#_bookmark33)
9. [Polymorphism with toString() method 22](#_bookmark37)
10. [Sort media in the car 23](#_bookmark41)
11. [Create a complete console application in the Aims class 24](#_bookmark45)
    1. [Người dùng chọn 1: View store 25](#_bookmark47)
       1. [Người dùng tiếp tục chọn 1. See a media’s details 26](#_bookmark49)
       2. [Người dùng chọn 2: Add a media to the cart 27](#_bookmark52)
       3. [Người dùng chọn 3: Play a media 27](#_bookmark54)
       4. [Người dùng chọn 4: See current cart 28](#_bookmark56)
    2. [Người dùng chọn 2: Update store 29](#_bookmark58)
       1. [Người dùng chọn 1: Add a media to the store 29](#_bookmark60)
       2. [Người dùng chọn 2: Remove a media from the store 30](#_bookmark63)
    3. [Người dùng chọn 3: See current cart 31](#_bookmark66)
       1. [Người dùng chọn 1: Filter medias in cart 32](#_bookmark69)
       2. [Người dùng chọn 2: Sort medias in cart 34](#_bookmark72)
       3. [Người dùng chọn 3: Remove media from cart 35](#_bookmark75)
       4. [Người dùng chọn 4: Play a media 36](#_bookmark78)
       5. [Người dùng chọn 5: Place order 36](#_bookmark80)
12. [Class Diagram 38](#_bookmark83)
13. [UseCase Diagram 39](#_bookmark85)
14. [Answer Questions 39](#_bookmark86)

Table of Figures

[Figure 1.1: Book Class 1 4](#_bookmark1)

[Figure 1.2: Book Class 2 5](#_bookmark2)

[Figure 2.1: Media Class 1 6](#_bookmark4)

[Figure 2.2: Media Class 2 7](#_bookmark5)

[Figure 3.1: Disc Class 8](#_bookmark8)

[Figure 3.2: DigitalVideoDisc Class 9](#_bookmark9)

[Figure 3.3: CompactDisc Class 9](#_bookmark10)

[Figure 3.4: Track Class 10](#_bookmark12)

[Figure 3.5: CompactDisc Class 1 11](#_bookmark14)

[Figure 3.6: CompactDisc Class 2 12](#_bookmark15)

[Figure 4.1: Playable interface 13](#_bookmark17)

[Figure 4.2: Method play() của DigitalVideoDisc 13](#_bookmark18)

[Figure 4.3: Method play() của Track 13](#_bookmark19)

[Figure 4.4: Method play() của CompactDisc 13](#_bookmark20)

[Figure 5.1: Cart Class 1 14](#_bookmark22)

[Figure 5.2: Cart Class 2 15](#_bookmark23)

[Figure 5.3: Cart Class 3 16](#_bookmark24)

[Figure 5.4: Cart Class 4 17](#_bookmark25)

[Figure 6.1: Store Class 1 18](#_bookmark27)

[Figure 6.2: Store Class 2 19](#_bookmark28)

[Figure 7.1: Constructor Track Class 20](#_bookmark30)

[Figure 7.2: Constructor CompactDisc Class 20](#_bookmark31)

[Figure 7.3: Constructor Media Class 20](#_bookmark32)

[Figure 7.4: Constructor Disc Class 21](#_bookmark34)

[Figure 8.1: Override equals in Media Class 21](#_bookmark35)

[Figure 8.2: Override equals in Track Class 21](#_bookmark36)

[Figure 9.1: Code mô phỏng Polymorphism 22](#_bookmark38)

[Figure 9.2: Override toString() in Media Class 22](#_bookmark39)

[Figure 9.3: Result demo Polymorphism 22](#_bookmark40)

[Figure 10.1: Add the comparators as attributes of the Media class 23](#_bookmark42)

[Figure 10.2: MediaComparatorByCostTitle Class 23](#_bookmark43)

[Figure 10.3: MediaComparatorByTitleCost Class 24](#_bookmark44)

[Figure 11.1: Màn hình chính 24](#_bookmark46)

[Figure 11.2: Vào Trang View Store 25](#_bookmark48)

[Figure 11.3: See a media's details 26](#_bookmark50)

[Figure 11.4: Thêm vào Cart 26](#_bookmark51)

[Figure 11.5: Thêm media vào Cart 27](#_bookmark53)

[Figure 11.6: Play a media 27](#_bookmark55)

[Figure 11.7: See current cart after sort 28](#_bookmark57)

[Figure 11.8: Vào Trang Update Store 29](#_bookmark59)

[Figure 11.9: Add a media to store 29](#_bookmark61)

[Figure 11.10: Result after add media to store 30](#_bookmark62)

[Figure 11.11: Remove a media from the store 30](#_bookmark64)

[Figure 11.12: Result after remove a media 31](#_bookmark65)

[Figure 11.13: Vào trang See current cart 31](#_bookmark67)

[Figure 11.14: Media in Cart 32](#_bookmark68)

[Figure 11.15: Filter Cart By id 32](#_bookmark70)

[Figure 11.16: Filter Cart By Title 33](#_bookmark71)

[Figure 11.17: Sort Cart By Title 34](#_bookmark73)

[Figure 11.18: Sort Cart By Cost 34](#_bookmark74)

[Figure 11.19: Remove media by id 35](#_bookmark76)

[Figure 11.20: Result after remove media in cart by id 35](#_bookmark77)

[Figure 11.21: Play a media in cart 36](#_bookmark79)

[Figure 11.22: Order 36](#_bookmark81)

[Figure 11.23: Result after order 37](#_bookmark82)

[Figure 12.1: Class Diagram 38](#_bookmark84)

[Figure 13.1: UseCase Diagram 39](#_bookmark87)

[Figure 14.1: Triển khai Comparable trong lớp trừu tượng Media 40](#_bookmark88)

[Figure 14.2: Mở rộng để so sánh nhiều thuộc tính hơn 40](#_bookmark89)

[Figure 14.3: Triển khai tại lớp con 40](#_bookmark90)

# Create the Book class

# 

Figure 1 create the book

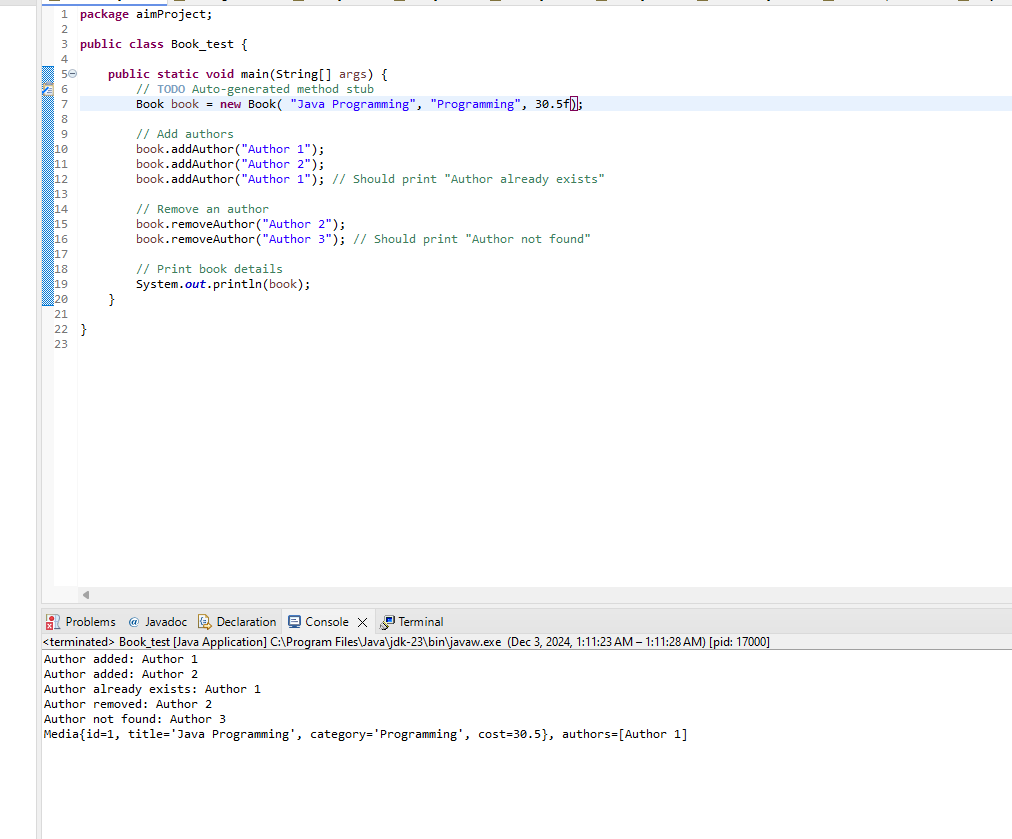
Result:  
 

Figure 2 result

# Creating the abstract Media class

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



Figure 3 media class

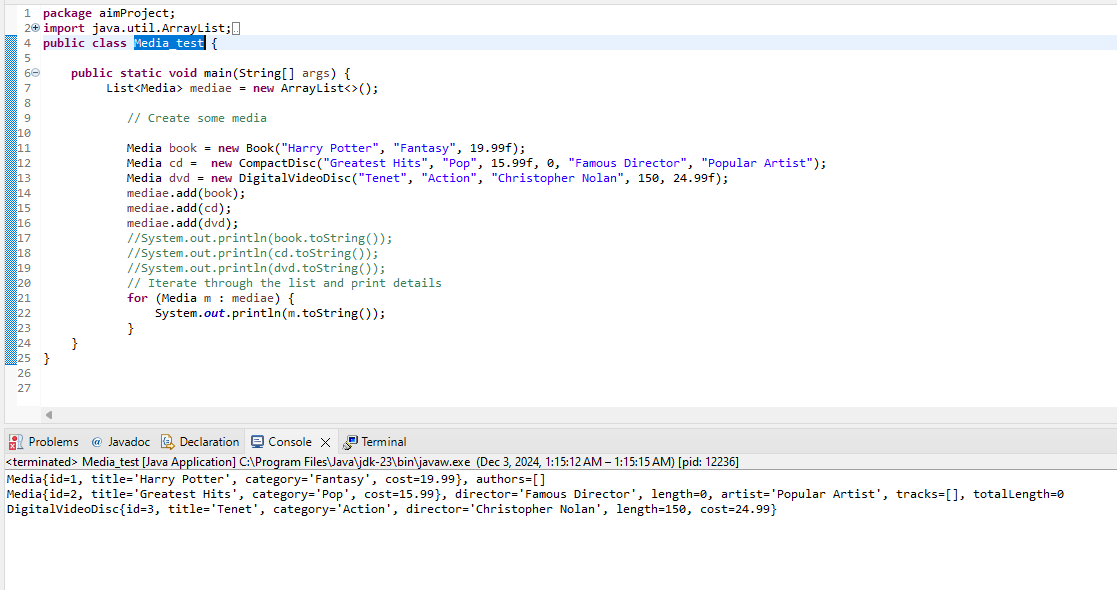
**

Figure 4 result

# Creating the CompactDisc class

# 

# 

Figure 5 compactdics

## Create the Disc class extending the Media clas

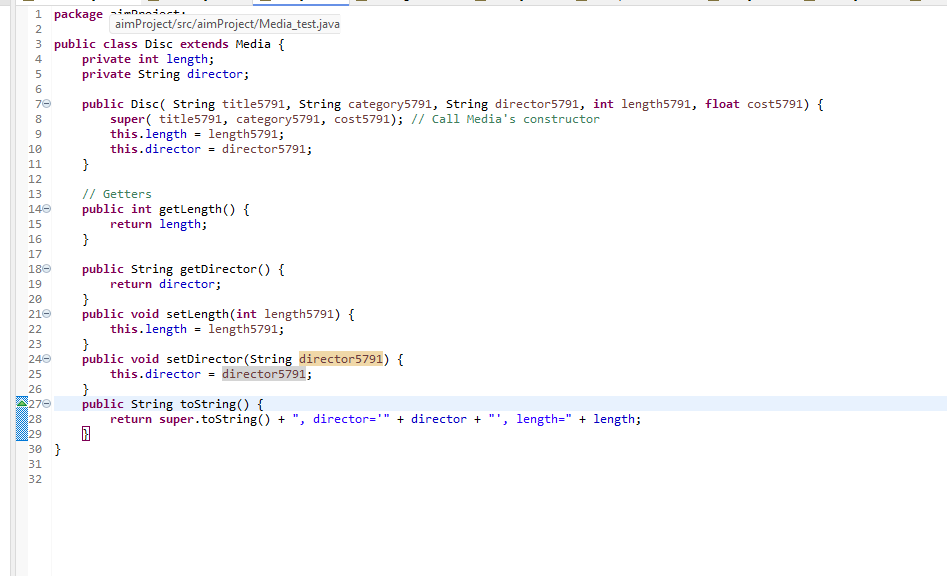


Figure 6Disk class

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

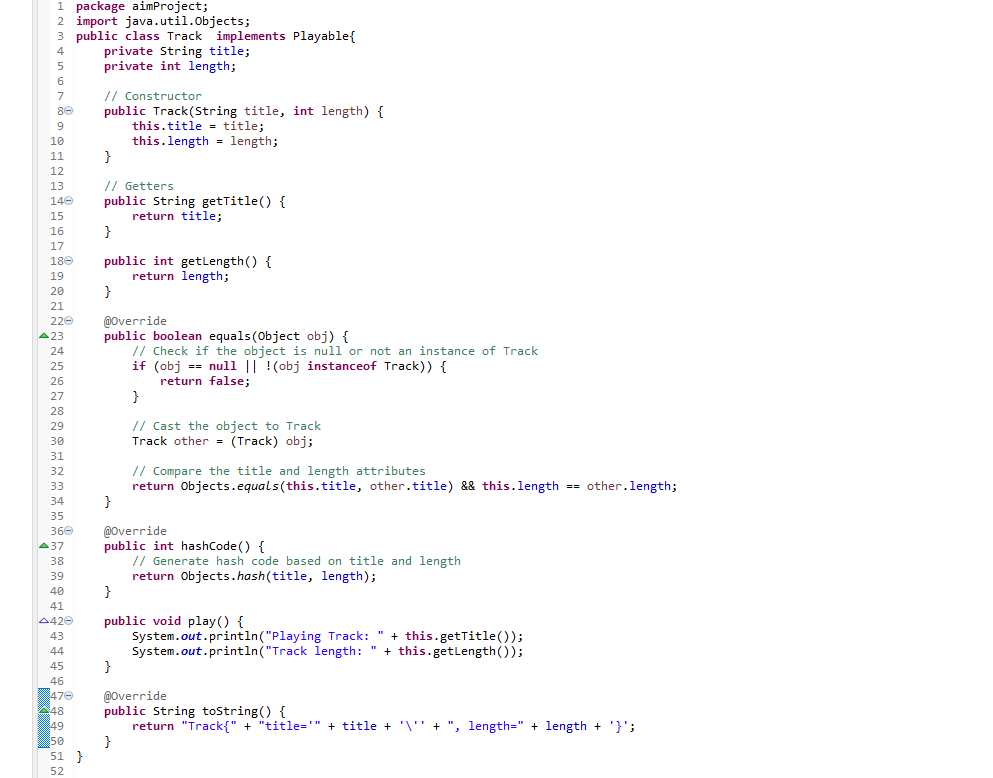


Figure 7track class

## Open the CompactDisc class

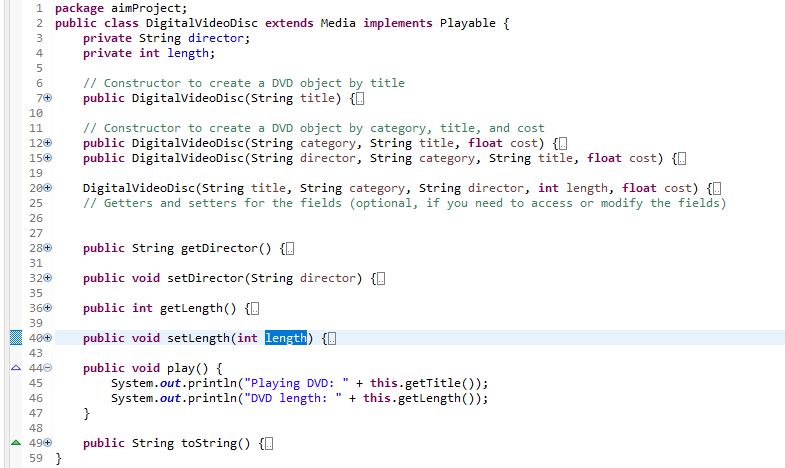
Figure 8 cd class

# Create the Playable interface

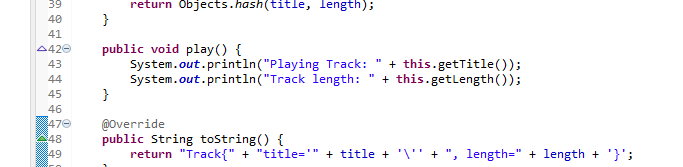
# 

Figure 9 playable

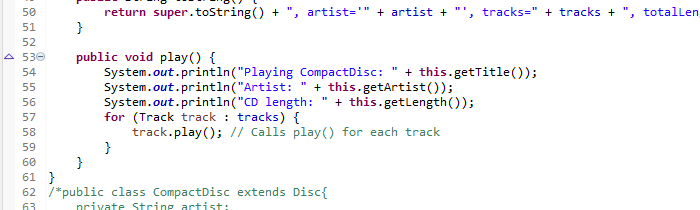
Implement play() cho các class DigitalVideoDisc, Track, CompactDisc



*Figure 4.2: Method play() của DigitalVideoDisc*



*Figure 4.3: Method play() của Track*



*Figure 4.4: Method play() của CompactDisc*

# Update the Cart class to work with Media

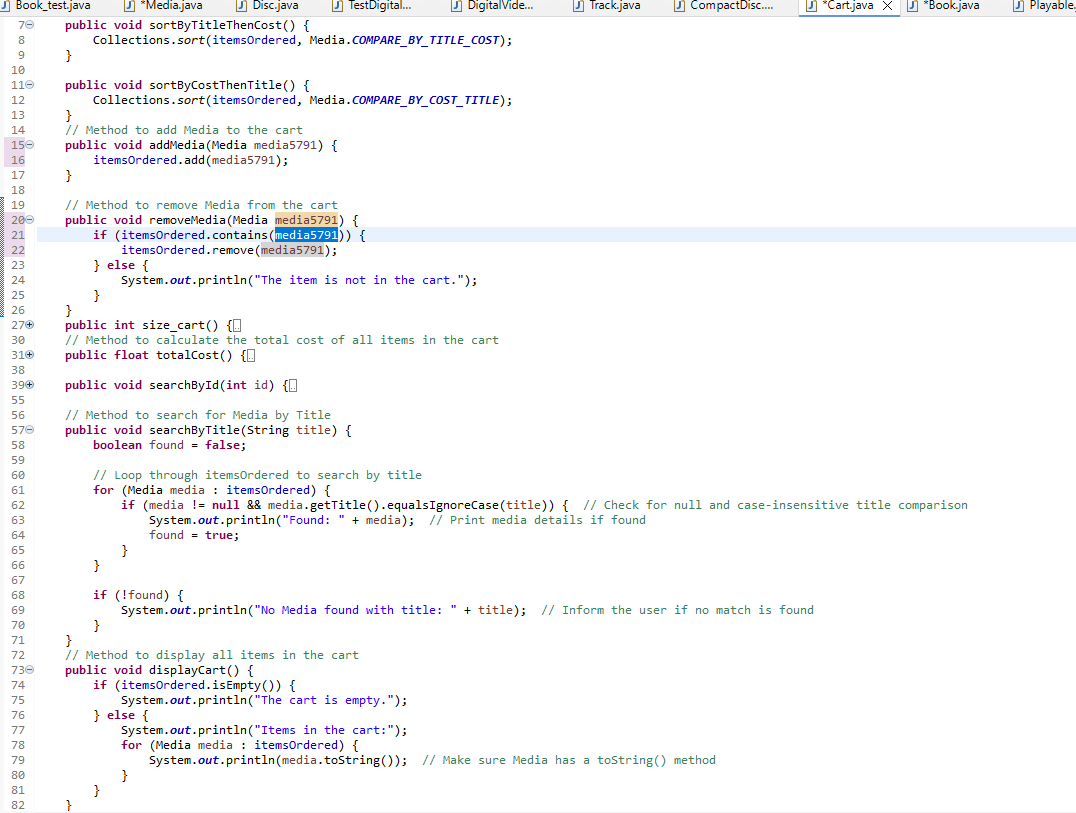
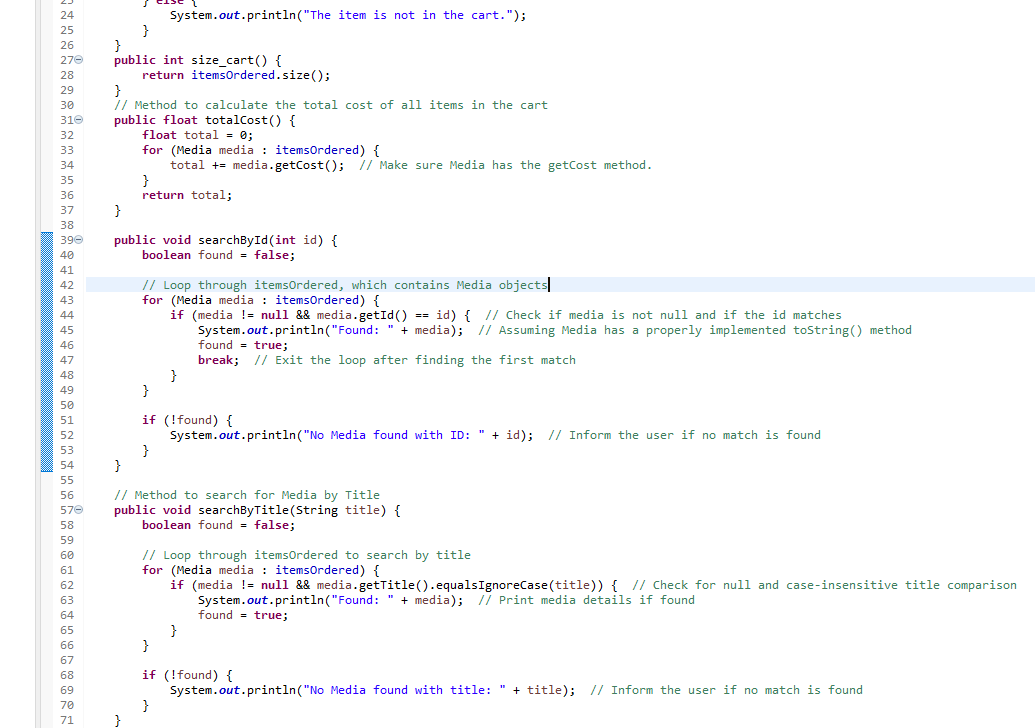
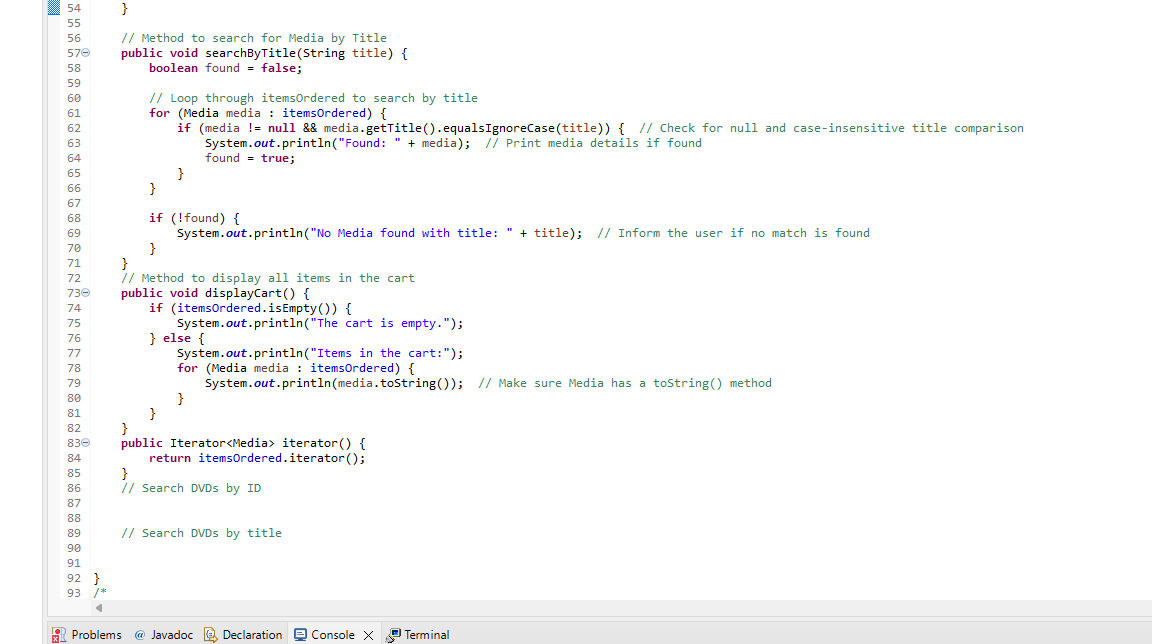


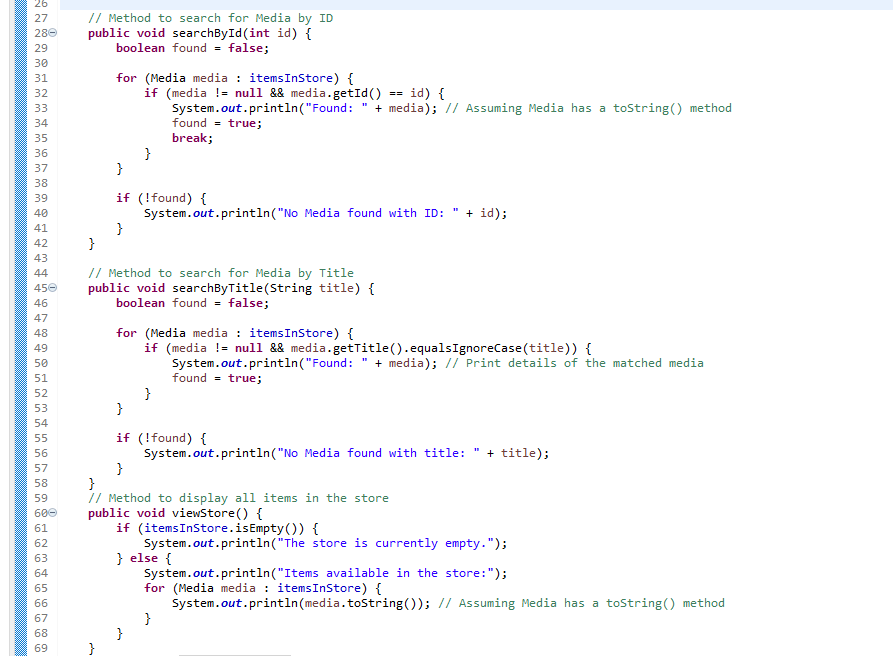
Figure 10 cart\_update





# Update the Store class to work with Media

*Figure 6.1: Store Class 1*



*Figure 6.2: Store Class 2*

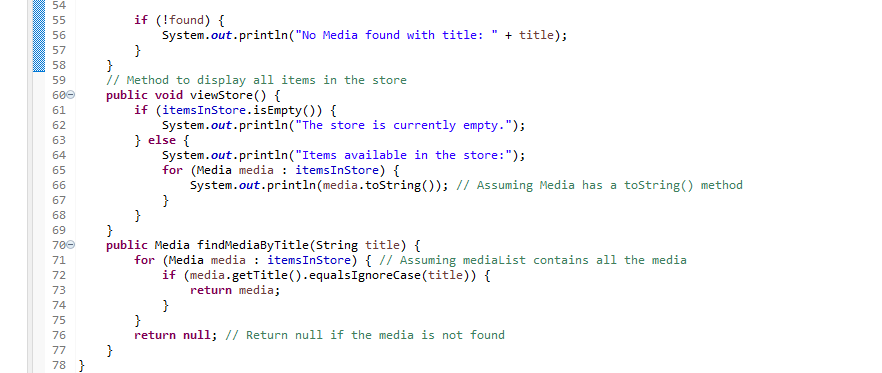
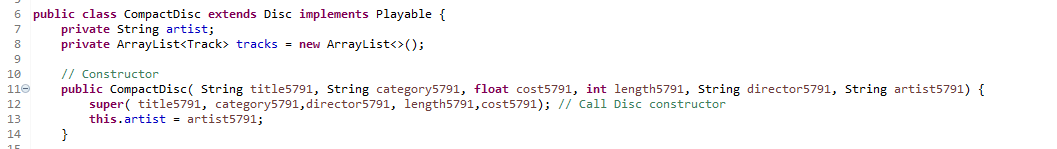


Figure 11 store3

# Constructors of whole classes and parent classes

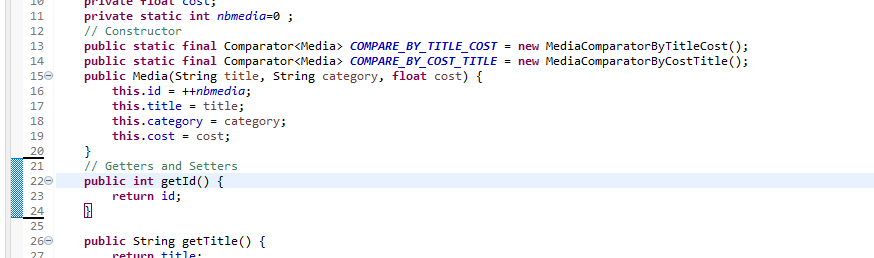
# 

*Figure 7.1: Constructor Track Class*

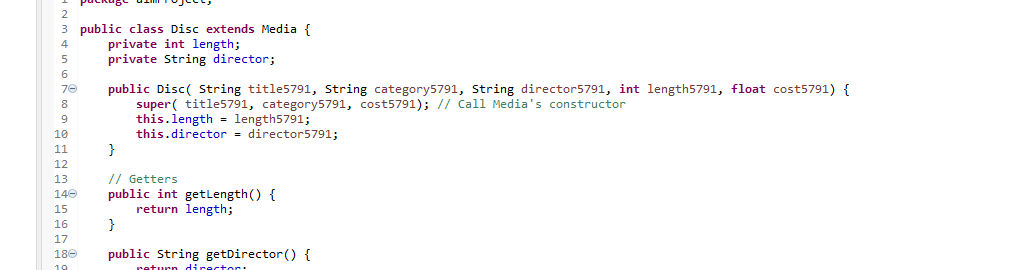
**

*Figure 7.2: Constructor CompactDisc Class*

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



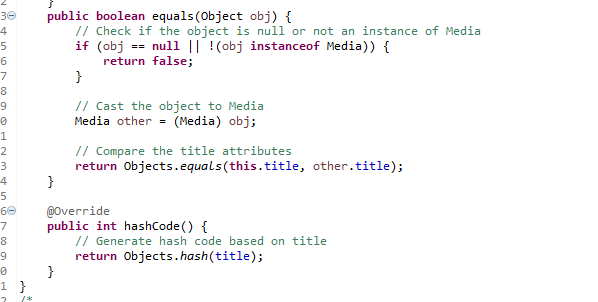
*Figure 7.3: Constructor Media Class*



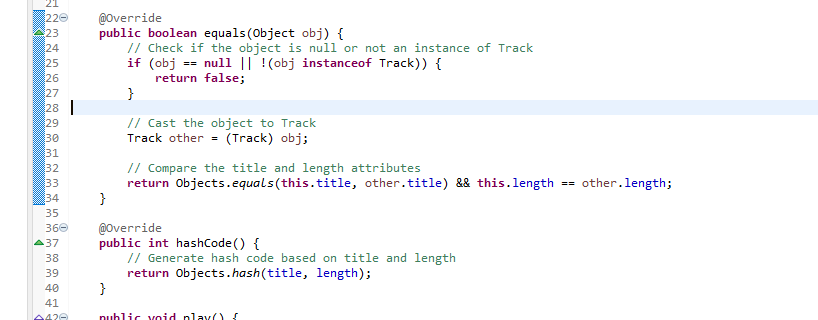
# Unique item in a list

*Figure 7.4: Constructor Disc Class*

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.



*Figure 8.1: Override equals in Media Class*

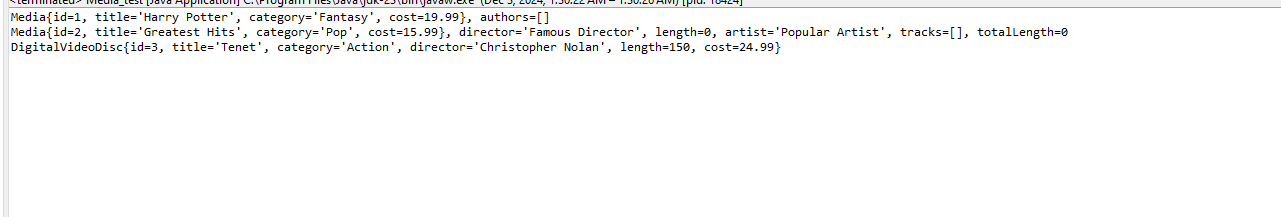
**

*Figure 8.2: Override equals in Track Class*

# Polymorphism with toString() method

# 

Kết quả



*Figure 9.3: Result demo Polymorphism*

When overriding the equals() method in Java, you need to follow the contract of the method, which includes:

1. **Reflexivity**: An object must be equal to itself.
2. **Symmetry**: If a.equals(b) is true, then b.equals(a) must also be true.
3. **Transitivity**: If a.equals(b) is true and b.equals(c) is true, then a.equals(c) must also be true.
4. **Consistency**: The result of a.equals(b) must not change unless one of the objects is modified.
5. **Null comparison**: An object is never equal to null.

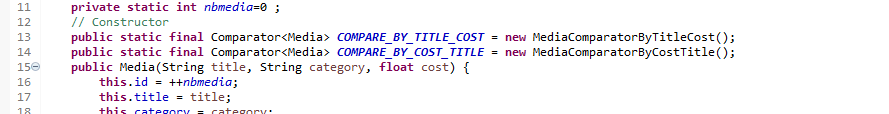
# Sort media in the car

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

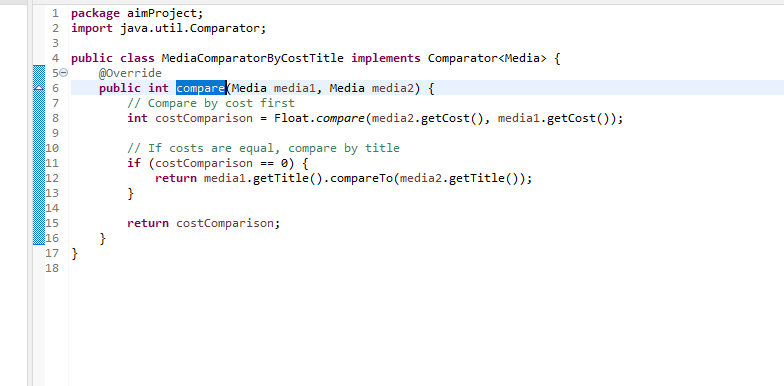
* Bằng title: Hiển thị tất cả các media theo thứ tự bảng chữ cái. Trong trường hợp cùng title,

media có cost cao hơn sẽ được hiển thị trước.

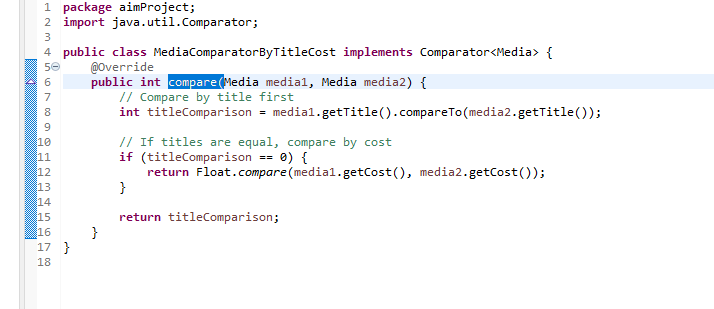
* Bằng cost: Hiển thị theo thứ tự cost giảm dần. Trong trường hợp cost như nhau, sắp xếp media theo thứ tự bảng chữ cái



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

**

*Figure 10.2: MediaComparatorByCostTitle Class*



*Figure 10.3: MediaComparatorByTitleCost Class*

# Create a complete console application in the Aims class

# 

*Figure 11.1: Màn hình chính*

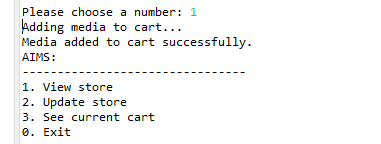
## Người dùng chọn 1: View store

## 

*Figure 11.2: Vào Trang View Store*

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

*Figure 11.3: See a media's details*

**

*Figure 11.4: Thêm vào Cart*

## Người dùng chọn 2: Update store

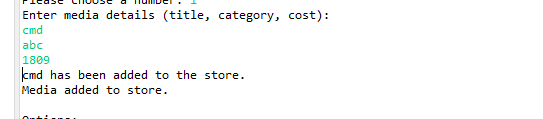
## 

*Figure 11.8: Vào Trang Update Store*

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

*Figure 11.9: Add a media to store*

=> Kết quả sau khi thêm



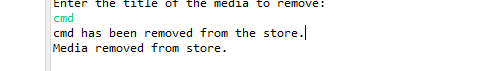
*Figure 11.10: Result after add media to store*

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

### 

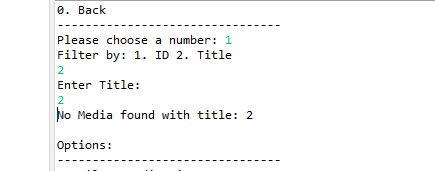
*Figure 11.11: Remove a media from the store*

=> Kết quả sau khi remove



### Người dùng chọn 1: Filter medias in cart

*Figure 11.15: Filter Cart By id*

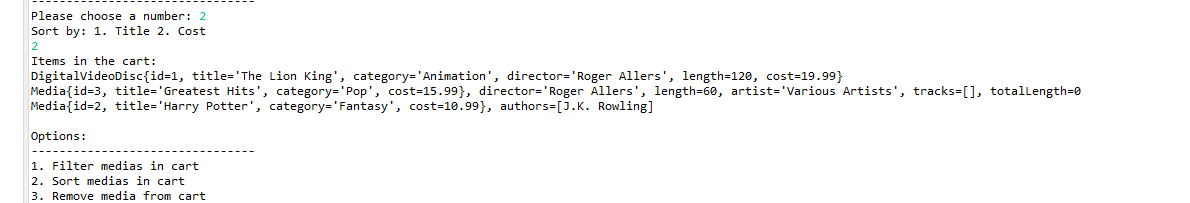


*Figure 11.16: Filter Cart By Title*

### Người dùng chọn 2: Sort medias in cart

### 

*Figure 11.17: Sort Cart By Title*

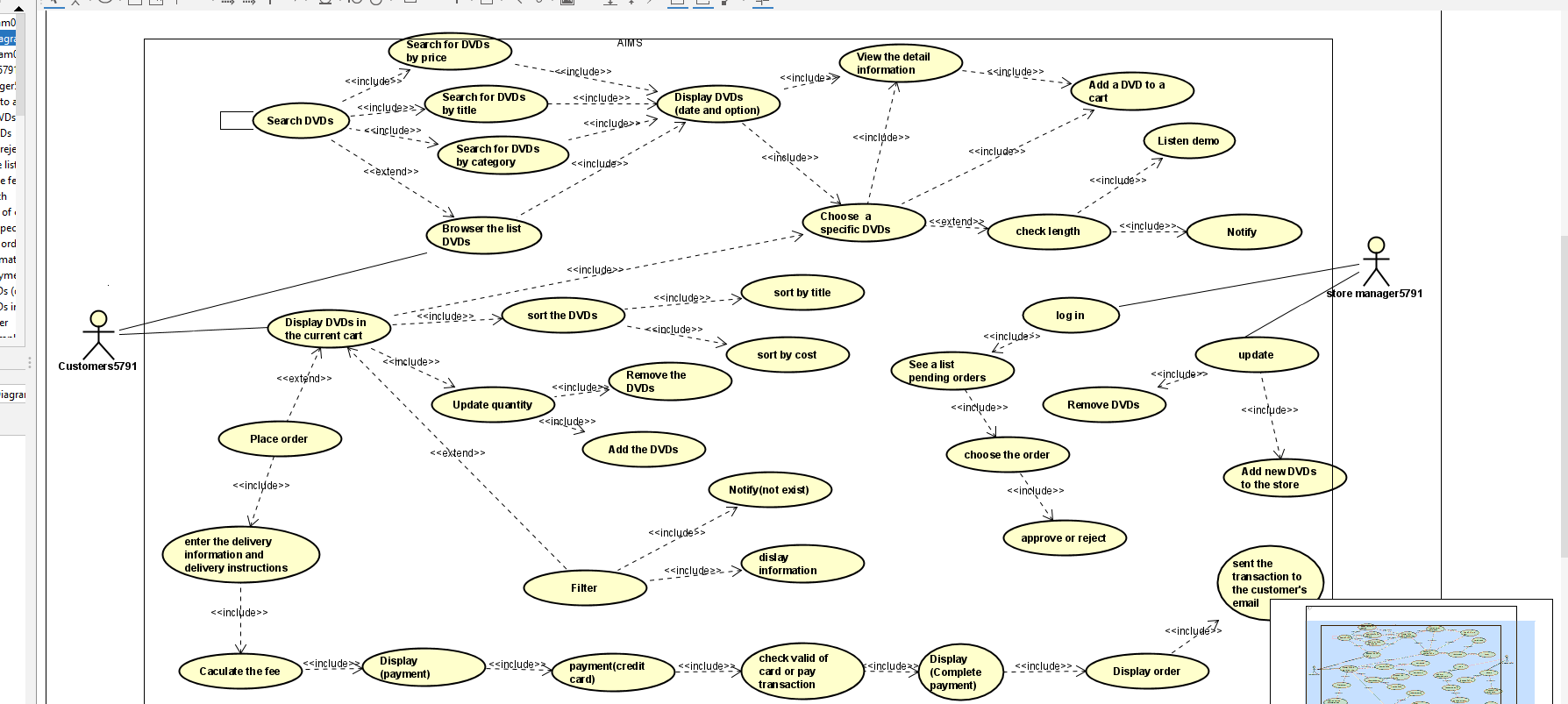
**

*Figure 11.18: Sort Cart By Cos*

# Class Diagram

# 

*Figure 12.1: Class Diagram*

1. UseCase Diagram
2. 
3. Answer Questions

*Figure 13.1: UseCase Diagram*

 **What class should implement the Comparable interface?**

* The class that represents the media types (such as Media, Book, DVD, CD, etc.) should implement the Comparable interface. This ensures that instances of these classes can be compared based on a specified order.

 **How should you implement the compareTo() method to reflect the ordering that we want?**

* You would implement the compareTo() method within the class (like Media or its subclasses) to define how objects of that class are compared. The compareTo() method returns a negative integer, zero, or a positive integer depending on whether the current object is less than, equal to, or greater than the object being compared, respectively. For example, to compare by title and cost

 **Can we have two ordering rules of the item (by title then cost and by cost then title) if we use this Comparable interface approach?**

* No, the Comparable interface only allows you to define one natural ordering for a class. If you want multiple sorting options (like by title then cost and by cost then title), you would need to use multiple Comparator objects instead, or use some kind of conditional logic within the compareTo() method to handle different cases.

 **Suppose the DVDs have a different ordering rule from the other media types, that is by title, then decreasing length, then cost. How would you modify your code to allow this?**

* To handle this, you could override the compareTo() method in the DVD class specifically. The DVD class would have its own implementation of compareTo() that considers the title, length (in descending order), and cost.