

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 04

Họ và tên: Phạm Quốc Cường

MSSV: 20225604

Lớp: **K67-Việt Nhật 01**

GVHD: Lê Thị Hoa

TA: **Phạm 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 .....	6
3	Creating the CompactDisc class .....	8
3.1	Create the Disc class extending the Media clas .....	8
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 .....	11
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.....	18
7	Constructors of whole classes and parent classes .....	20
8	Unique item in a list .....	21
9	Polymorphism with <code>toString()</code> method .....	22
10	Sort media in the car.....	23
11	Create a complete console application in the Aims class .....	24
11.1	Nguời dùng chọn 1: View store .....	25
11.1.1	Nguời dùng tiếp tục chọn 1. See a media's details .....	26
11.1.2	Nguời dùng chọn 2: Add a media to the cart .....	27
11.1.3	Nguời dùng chọn 3: Play a media.....	27
11.1.4	Nguời dùng chọn 4: See current cart .....	28
11.2	Nguời dùng chọn 2: Update store .....	29
11.2.1	Nguời dùng chọn 1: Add a media to the store .....	29
11.2.2	Nguời dùng chọn 2: Remove a media from the store.....	30
11.3	Nguời dùng chọn 3: See current cart .....	31
11.3.1	Nguời dùng chọn 1: Filter medias in cart.....	32
11.3.2	Nguời dùng chọn 2: Sort medias in cart .....	34
11.3.3	Nguời dùng chọn 3: Remove media from cart.....	35
11.3.4	Nguời dùng chọn 4: Play a media.....	36
11.3.5	Nguời dùng chọn 5: Place order .....	36

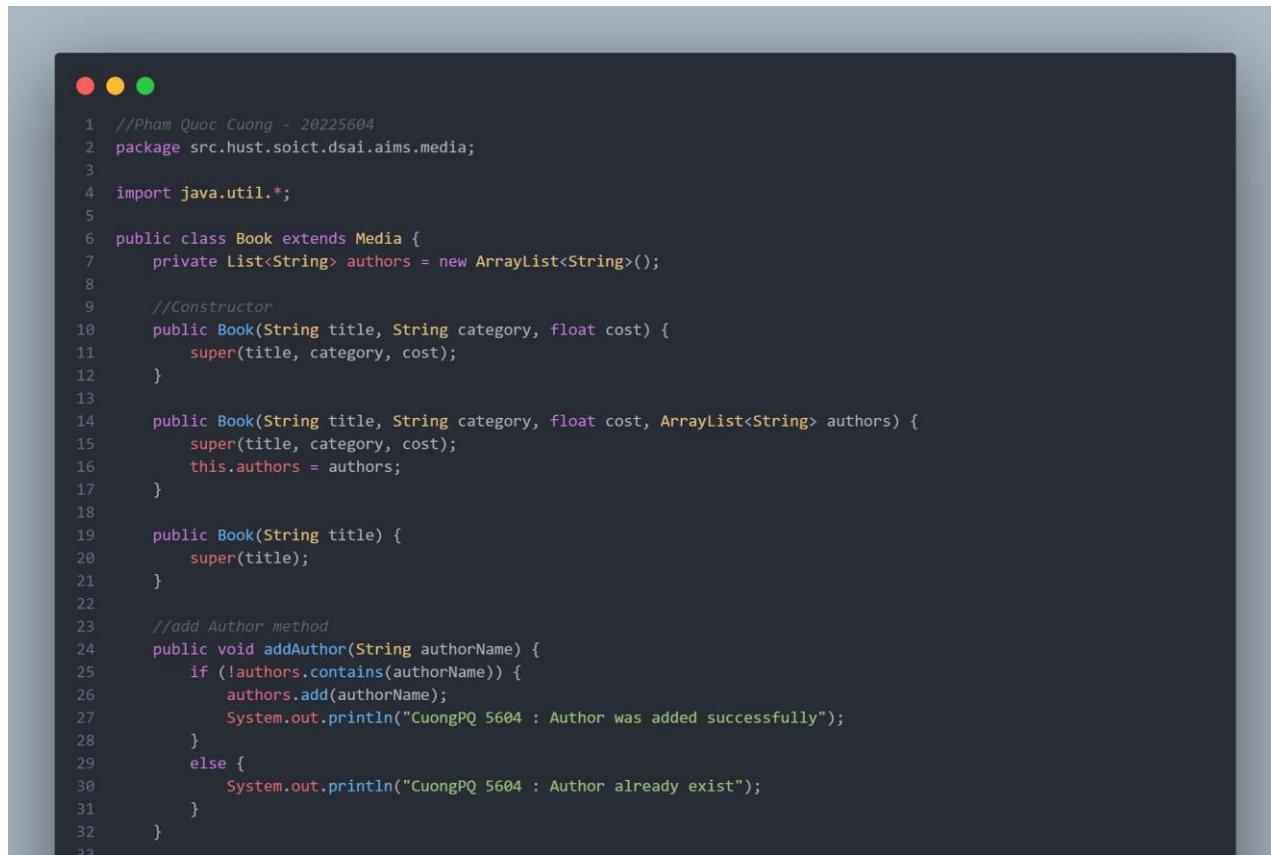
12	Class Diagram .....	38
13	UseCase Diagram.....	39
14	Answer Questions.....	39

## 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



```
1 //Pham Quoc Cuong - 20225604
2 package src.hust.soict.dsai.aims.media;
3
4 import java.util.*;
5
6 public class Book extends Media {
7     private List<String> authors = new ArrayList<String>();
8
9     //Constructor
10    public Book(String title, String category, float cost) {
11        super(title, category, cost);
12    }
13
14    public Book(String title, String category, float cost, ArrayList<String> authors) {
15        super(title, category, cost);
16        this.authors = authors;
17    }
18
19    public Book(String title) {
20        super(title);
21    }
22
23    //add Author method
24    public void addAuthor(String authorName) {
25        if (!authors.contains(authorName)) {
26            authors.add(authorName);
27            System.out.println("CuongPQ 5604 : Author was added successfully");
28        }
29        else {
30            System.out.println("CuongPQ 5604 : Author already exist");
31        }
32    }
33}
```

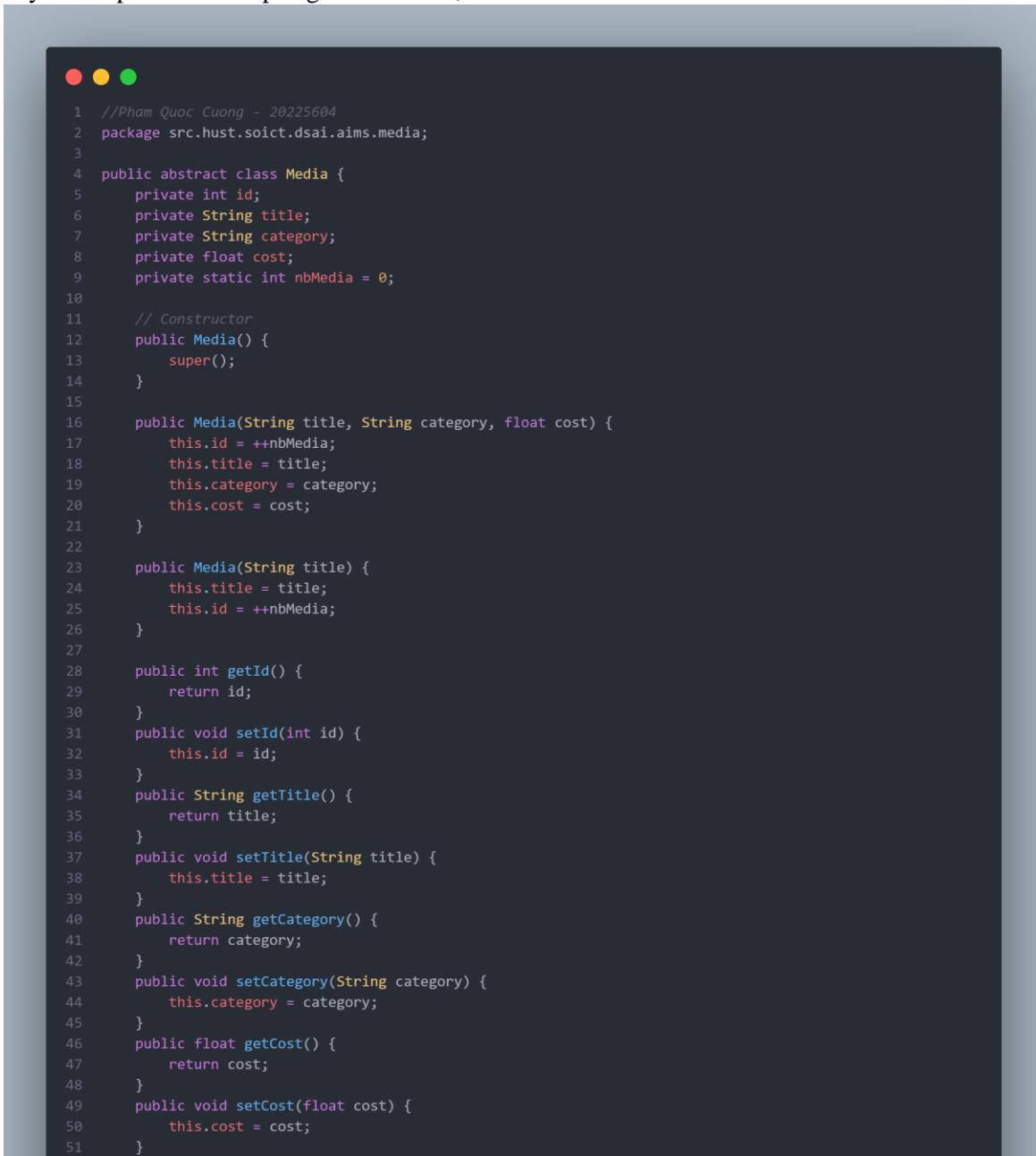
Figure 1.1: Book Class 1

```
33 // remove Author method
34 public void removeAuthor(String authorName) {
35     if (authors.contains(authorName)) {
36         authors.remove(authorName);
37         System.out.println("CuongPQ 5604 : Removed author successfully");
38     } else {
39         System.out.println("CuongPQ 5604 : Author not found");
40     }
41 }
42
43
44 public List<String> getAuthors() {
45     return authors;
46 }
47
48 public void setAuthors(List<String> authors) {
49     this.authors = authors;
50 }
51
52 @Override
53 public String toString() {
54     return "Book - Title: " + this.getTitle() + ", Category: " + this.getCategory() + ", Cost: " + this.getCost();
55 }
56 }
57 }
```

Figure 1.2: Book Class 2

## 2 Creating the abstract Media class

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



```
1 //Pham Quoc Cuong - 20225604
2 package src.hust.soict.dsai.aims.media;
3
4 public abstract class Media {
5     private int id;
6     private String title;
7     private String category;
8     private float cost;
9     private static int nbMedia = 0;
10
11    // Constructor
12    public Media() {
13        super();
14    }
15
16    public Media(String title, String category, float cost) {
17        this.id = ++nbMedia;
18        this.title = title;
19        this.category = category;
20        this.cost = cost;
21    }
22
23    public Media(String title) {
24        this.title = title;
25        this.id = ++nbMedia;
26    }
27
28    public int getId() {
29        return id;
30    }
31    public void setId(int id) {
32        this.id = id;
33    }
34    public String getTitle() {
35        return title;
36    }
37    public void setTitle(String title) {
38        this.title = title;
39    }
40    public String getCategory() {
41        return category;
42    }
43    public void setCategory(String category) {
44        this.category = category;
45    }
46    public float getCost() {
47        return cost;
48    }
49    public void setCost(float cost) {
50        this.cost = cost;
51    }
}
```

Figure 2.1: Media Class I

```
52     public boolean isMatch(String title) {
53         return this.getTitle().equalsIgnoreCase(title);
54     }
55
56
57 //Pham Quoc Cuong - 20225604
58 @Override
59     public boolean equals(Object obj) {
60         if (this == obj) { // Check if the references are the same
61             return true;
62         }
63         if (obj == null || getClass() != obj.getClass()) { // Check if obj is null or not the same class
64             return false;
65         }
66         Media media = (Media) obj; // Cast obj to Media
67         return title != null && title.equals(media.title); // Compare titles
68     }
69
70 //Pham Quoc Cuong - 20225604
71 @Override
72     public String toString() {
73         return "Media {id=" + id
74                 + ", title:'" + title + "\'"
75                 + ", category: " + category + "\'
76                 + ", cost: $" + cost + "}";
77     }
78 }
79 }
```

Figure 2.2: Media Class 2



```
1 //Pham Quoc Cuong - 20225604
2 package src.hust.soict.dsai.aims.media;
3
4 import java.util.*;
5
6 public class Book extends Media {
7     private List<String> authors = new ArrayList<String>();
8
9     //Constructor
10    public Book(String title, String category, float cost) {
11        super(title, category, cost);
12    }
13
14    public Book(String title, String category, float cost, ArrayList<String> authors) {
15        super(title, category, cost);
16        this.authors = authors;
17    }
18
19    public Book(String title) {
20        super(title);
21    }
22
23    //add Author method
24    public void addAuthor(String authorName) {
25        if (!authors.contains(authorName)) {
26            authors.add(authorName);
27            System.out.println("CuongPQ 5604 : Author was added successfully");
28        }
29        else {
30            System.out.println("CuongPQ 5604 : Author already exist");
31        }
32    }
33
34    // remove Author method
35    public void removeAuthor(String authorName) {
36        if (authors.contains(authorName)) {
37            authors.remove(authorName);
38            System.out.println("CuongPQ 5604 : Removed author successfully");
39        } else {
40            System.out.println("CuongPQ 5604 : Author not found");
41        }
42    }
43
44    public List<String> getAuthors() {
45        return authors;
46    }
47
48    public void setAuthors(List<String> authors) {
49        this.authors = authors;
50    }
51
52    // Pham Quoc Cuong - 20225604
53    @Override
54    public String toString() {
55        return "Book - Title: " + this.getTitle() + ", Category: " + this.getCategory() + ", Cost: " + this.getCost();
56    }
57 }
58
```

Figure 2.3: Book class extends Media



The screenshot shows a Java code editor with a dark theme. At the top left, there are three circular icons: red, yellow, and green. The code itself is as follows:

```
1 // Pham Quoc Cuong - 20225604
2 package src.hust.soict.dsai.aims.media;
3
4 public class DigitalVideoDisc extends Disc implements Playable {
5     // Constructors
6     public DigitalVideoDisc(String title) {
7         super(title);
8     }
9     public DigitalVideoDisc(String title, String category, float cost) {
10        super(title, category, cost);
11    }
12    public DigitalVideoDisc(String title, String category, String directory, float cost) {
13        super(title, category, directory, cost);
14    }
15    public DigitalVideoDisc(String title, String category, String directory, int length, float cost) {
16        super(title, category,directory, length, cost);
17    }
18
19 // Pham Quoc Cuong - 20225604
20 // Play Method
21 @Override
22 public void play() {
23     System.out.println("Playing DVD: " + this.getTitle());
24     System.out.println("DVD length: " + this.getLength());
25 }
26
27 // Pham Quoc Cuong - 20225604
28 @Override
29 public String toString() {
30     return "DVD - Title: " + this.getTitle() + ", Category: " + this.getCategory() + ", Cost: " + this.getCost();
31 }
32 }
33
34
35
```

Figure 2.4: DigitalVideoDisc class extends Media

### 3 Creating the CompactDisc class

#### 3.1 Create the Disc class extending the Media clas

```
● ● ●  
1 // Pham Quoc Cuong - 20225604  
2 package src.hust.soict.dsai.aims.media;  
3  
4 public class Disc extends Media {  
5     private int length;  
6     private String director;  
7  
8     // Pham Quoc Cuong - 20225604  
9     //Constructors  
10    public Disc(String title) {  
11        super(title);  
12    }  
13  
14    public Disc(String title, String category, String director, int length, float cost) {  
15        super(title, category, cost);  
16        this.director = director;  
17        this.length = length;  
18    }  
19  
20    public Disc(String title, String category, String director, float cost) {  
21        super(title, category, cost);  
22        this.director = director;  
23    }  
24  
25    public Disc(String title, String category, float cost) {  
26        super(title, category, cost);  
27    }  
28  
29    public int getLength() {  
30        return length;  
31    }  
32    public void setLength(int length) {  
33        this.length = length;  
34    }  
35    public String getDirector() {  
36        return director;  
37    }  
38    public void setDirector(String director) {  
39        this.director = director;  
40    }  
41}  
42  
43
```

Figure 3.1: Disc Class



The screenshot shows a Java code editor with a dark theme. At the top left, there are three colored circular icons (red, yellow, green). The code itself is a Java class named `DigitalVideoDisc`. It includes several constructors, a `play()` method, and a `toString()` method. The code is annotated with comments and copyright notices.

```
1 // Pham Quoc Cuong - 20225604
2 package src.hust.soict.dsai.aims.media;
3
4 public class DigitalVideoDisc extends Disc implements Playable {
5     // Constructors
6     public DigitalVideoDisc(String title) {
7         super(title);
8     }
9     public DigitalVideoDisc(String title, String category, float cost) {
10        super(title, category, cost);
11    }
12    public DigitalVideoDisc(String title, String category, String directory, float cost) {
13        super(title, category, directory, cost);
14    }
15    public DigitalVideoDisc(String title, String category, String directory, int length, float cost) {
16        super(title, category,directory, length, cost);
17    }
18
19 // Pham Quoc Cuong - 20225604
20 // Play Method
21 @Override
22 public void play() {
23     System.out.println("Playing DVD: " + this.getTitle());
24     System.out.println("DVD length: " + this.getLength());
25 }
26
27 @Override
28 public String toString() {
29     return "DVD - Title: " + this.getTitle() + ", Category: " + this.getCategory() + ", Cost: " + this.getCost();
30 }
31 }
32
33
34
```

Figure 3.2: *DigitalVideoDisc Class*

```
1 //Pham Quoc Cuong - 20225604
2 package src.hust.soict.dsai.aims.media;
3
4 import java.util.*;
5
6 public class CompactDisc extends Disc implements Playable {
7     private String artist;
8     private List<Track> tracks = new ArrayList<Track>();
9
10    // Constructor
11    public CompactDisc(String title) {
12        super(title);
13    }
14    public CompactDisc(String title, String category, String director, int length, float cost) {
15        super(title, category, director, length, cost);
16    }
17
18    public CompactDisc(String title, String category, String director, ArrayList<Track> tracks, float cost) {
19        super(title, category, director, cost);
20        this.tracks = tracks;
21        this.setLength(this.getLength());
22    }
23    public CompactDisc(String title, String artist, ArrayList<Track> tracks) {
24        super(title);
25        this.artist = artist;
26        this.tracks = tracks;
27    }
28    public CompactDisc(String title, String artist) {
29        super(title);
30        this.artist = artist;
31    }
32    public CompactDisc(String title, String category, String director, int length, float cost, String artist, ArrayList<Track> tracks) {
33        super(title, category, director, length, cost);
34        this.artist = artist;
35        this.tracks = tracks;
36    }
37
38    // Getter for artist
39    public String getArtist() {
40        return artist;
41    }
42
43    // Methods to manage tracks
44    public void addTrack(Track track) {
45        if (tracks.contains(track)) {
46            System.out.println("CuongPQ 5604 : Track already exist");
47        } else {
48            tracks.add(track);
49            System.out.println("CuongPQ 5604 : The track " +"\\"+ track.getTitle()+"\\"+" have been added! ");
50        }
51    }
52
53    public void removeTrack(Track track) {
54        if (tracks.contains(track)) {
55            tracks.remove(track);
56            System.out.println("CuongPQ 5604 : The track " +"\\"+ track.getTitle()+"\\"+" have been removed! ");
57        } else {
58            System.out.println("Track not found" );
59        }
60    }
61}
```

```
60      }
61
62     // Get total length of CD
63     public int getLength() {
64         int totallength = 0;
65         for (Track track : tracks) {
66             totallength += track.getLength();
67         }
68         return totallength;
69     }
70
71     // Pham Quoc Cuong - 20225604
72     @Override
73     public void play() {
74         System.out.println("Playing CD: " + this.getTitle());
75         System.out.println("Artist: " + this.getArtist());
76         System.out.println("CD length: " + this.getLength());
77         System.out.println("Tracks:");
78         for (Track track : tracks) {
79             track.play();
80         }
81     }
82
83     // Pham Quoc Cuong - 20225604
84     @Override
85     public String toString() {
86         return "CD - Title: " + this.getTitle() + ", Category: " + this.getCategory() + ", Artist: " + this.getArtist() + ", Cost: " + this.getCost();
87     }
88 }
89
```

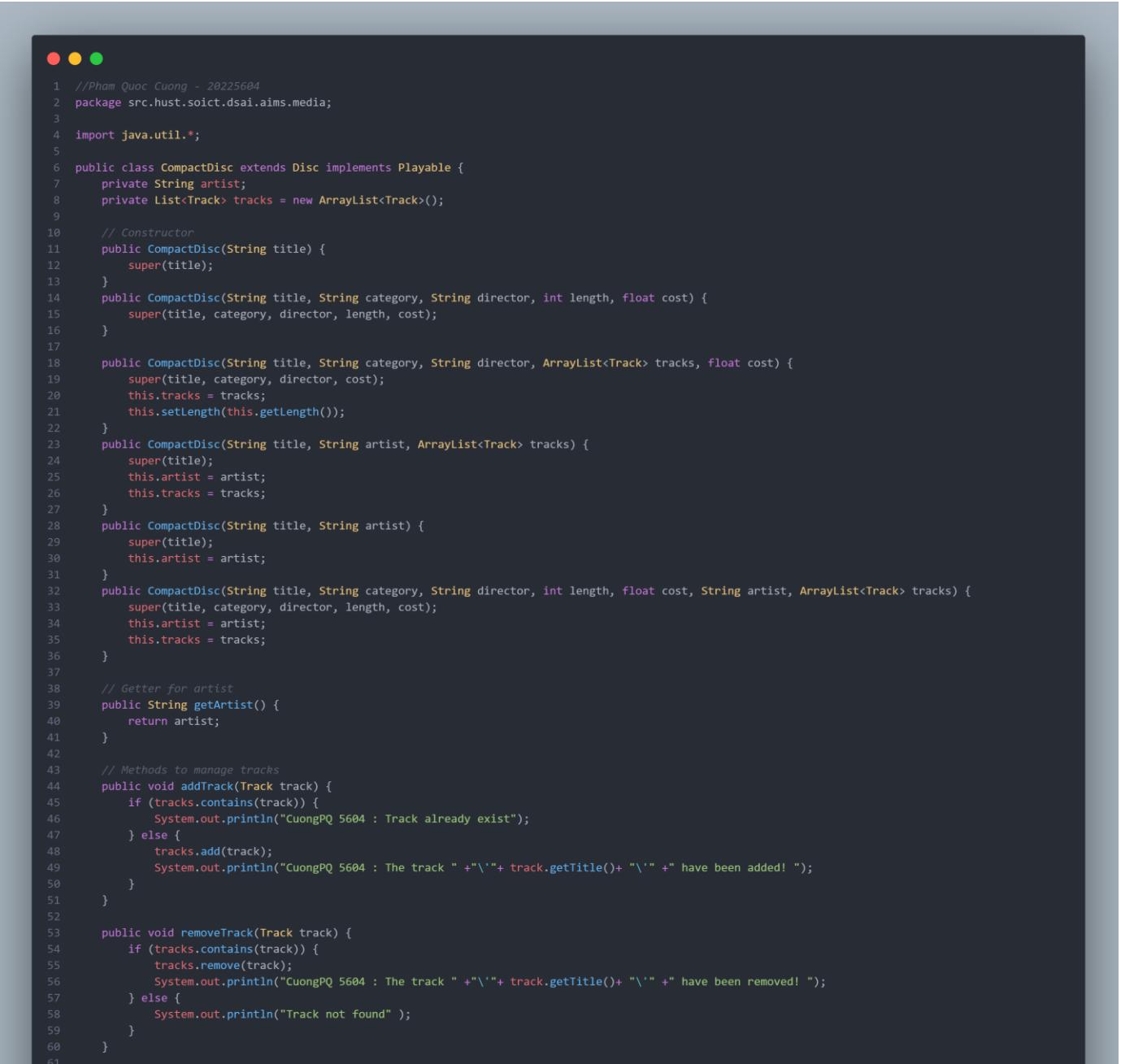
Figure 3.3: CompactDisc Class

### 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.

```
1 // Pham Quoc Cuong - 20225604
2 package src.hust.soict.dsai.aims.media;
3
4 public class Track implements Playable {
5     private String title;
6     private int length;
7
8     // Constructor
9     public Track(String title, int length) {
10         this.title = title;
11         this.length = length;
12     }
13
14     // Getters
15     public String getTitle() {
16         return title;
17     }
18     public int getLength() {
19         return length;
20     }
21
22     @Override
23     public void play() {
24         System.out.println("Playing Track: " + this.getTitle());
25         System.out.println("Track length: " + this.getLength());
26     }
27
28     @Override
29     public boolean equals(Object obj) {
30         if (this == obj) { // Check if the references are the same
31             return true;
32         }
33         if (obj == null || getClass() != obj.getClass()) { // Check if obj is null or not the same class
34             return false;
35         }
36         Track track = (Track) obj; // Cast obj to Track
37         return length == track.length && (title != null && title.equals(track.title)); // Compare title and length
38     }
39
40 }
41
```

Figure 3.4: Track Class

### 3.3 Open the CompactDisc class



```

1 //Pham Quoc Cuong - 20225604
2 package src.hust.soict.dsai.aims.media;
3
4 import java.util.*;
5
6 public class CompactDisc extends Disc implements Playable {
7     private String artist;
8     private List<Track> tracks = new ArrayList<Track>();
9
10    // Constructor
11    public CompactDisc(String title) {
12        super(title);
13    }
14    public CompactDisc(String title, String category, String director, int length, float cost) {
15        super(title, category, director, length, cost);
16    }
17
18    public CompactDisc(String title, String category, String director, ArrayList<Track> tracks, float cost) {
19        super(title, category, director, cost);
20        this.tracks = tracks;
21        this.setLength(this.getLength());
22    }
23    public CompactDisc(String title, String artist, ArrayList<Track> tracks) {
24        super(title);
25        this.artist = artist;
26        this.tracks = tracks;
27    }
28    public CompactDisc(String title, String artist) {
29        super(title);
30        this.artist = artist;
31    }
32    public CompactDisc(String title, String category, String director, int length, float cost, String artist, ArrayList<Track> tracks) {
33        super(title, category, director, length, cost);
34        this.artist = artist;
35        this.tracks = tracks;
36    }
37
38    // Getter for artist
39    public String getArtist() {
40        return artist;
41    }
42
43    // Methods to manage tracks
44    public void addTrack(Track track) {
45        if (tracks.contains(track)) {
46            System.out.println("CuongPQ 5604 : Track already exist");
47        } else {
48            tracks.add(track);
49            System.out.println("CuongPQ 5604 : The track " +"\\"+ track.getTitle()+"\\"+" have been added! ");
50        }
51    }
52
53    public void removeTrack(Track track) {
54        if (tracks.contains(track)) {
55            tracks.remove(track);
56            System.out.println("CuongPQ 5604 : The track " +"\\"+ track.getTitle()+"\\"+" have been removed! ");
57        } else {
58            System.out.println("Track not found" );
59        }
60    }
61

```

Figure 3.5: CompactDisc Class 1

```
60
61 // Get total length of CD
62 public int getLength() {
63     int totalLength = 0;
64     for (Track track : tracks) {
65         totalLength += track.getLength();
66     }
67     return totalLength;
68 }
69
70 // Pham Quoc Cuong - 20225604
71 @Override
72 public void play() {
73     System.out.println("Playing CD: " + this.getTitle());
74     System.out.println("Artist: " + this.getArtist());
75     System.out.println("CD length: " + this.getLength());
76     System.out.println("Tracks:");
77     for (Track track : tracks) {
78         track.play();
79     }
80 }
81
82 // Pham Quoc Cuong - 20225604
83 @Override
84 public String toString() {
85     return "CD - Title: " + this.getTitle() + ", Category: " + this.getCategory() + ", Artist: " + this.getArtist() + ", Cost: " + this.getCost();
86 }
87
88 }
89
```

Figure 3.6: CompactDisc Class 2

## 4 Create the Playable interface



Figure 4.1: Playable interface

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



Figure 4.2: Method play() của DigitalVideoDisc



Figure 4.3: Method play() của Track

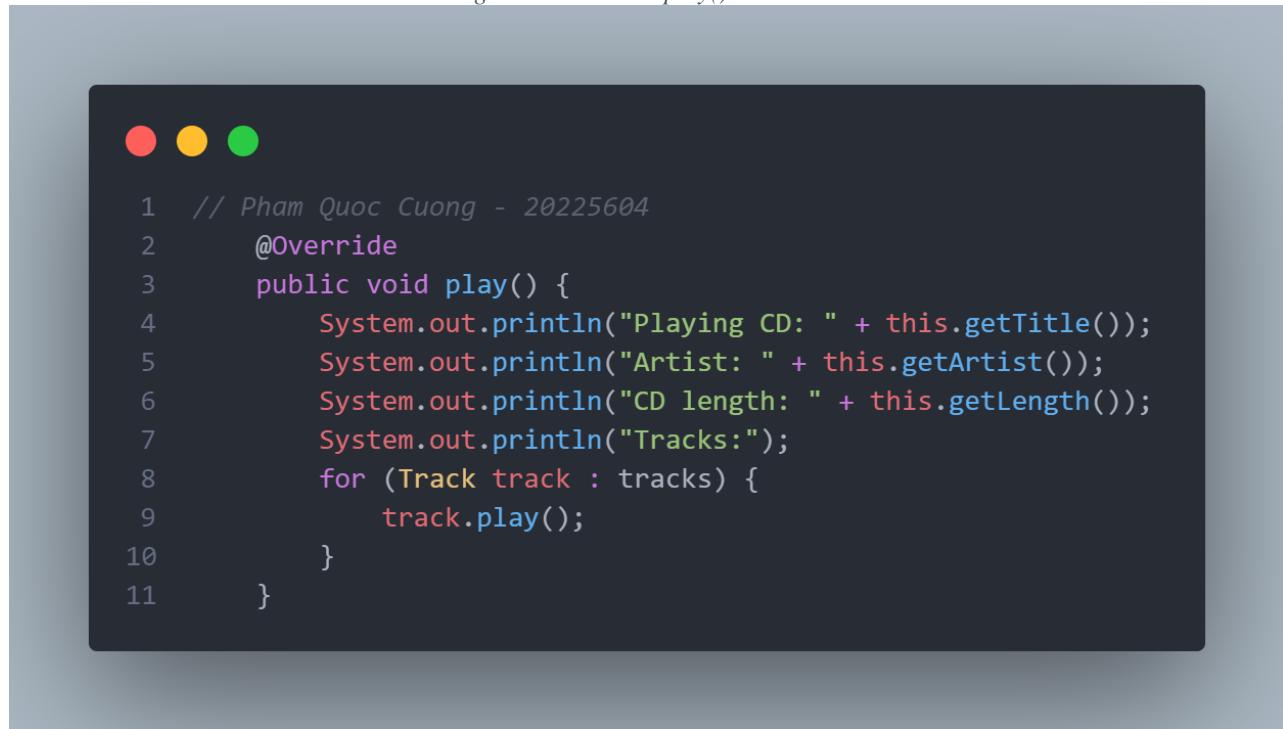
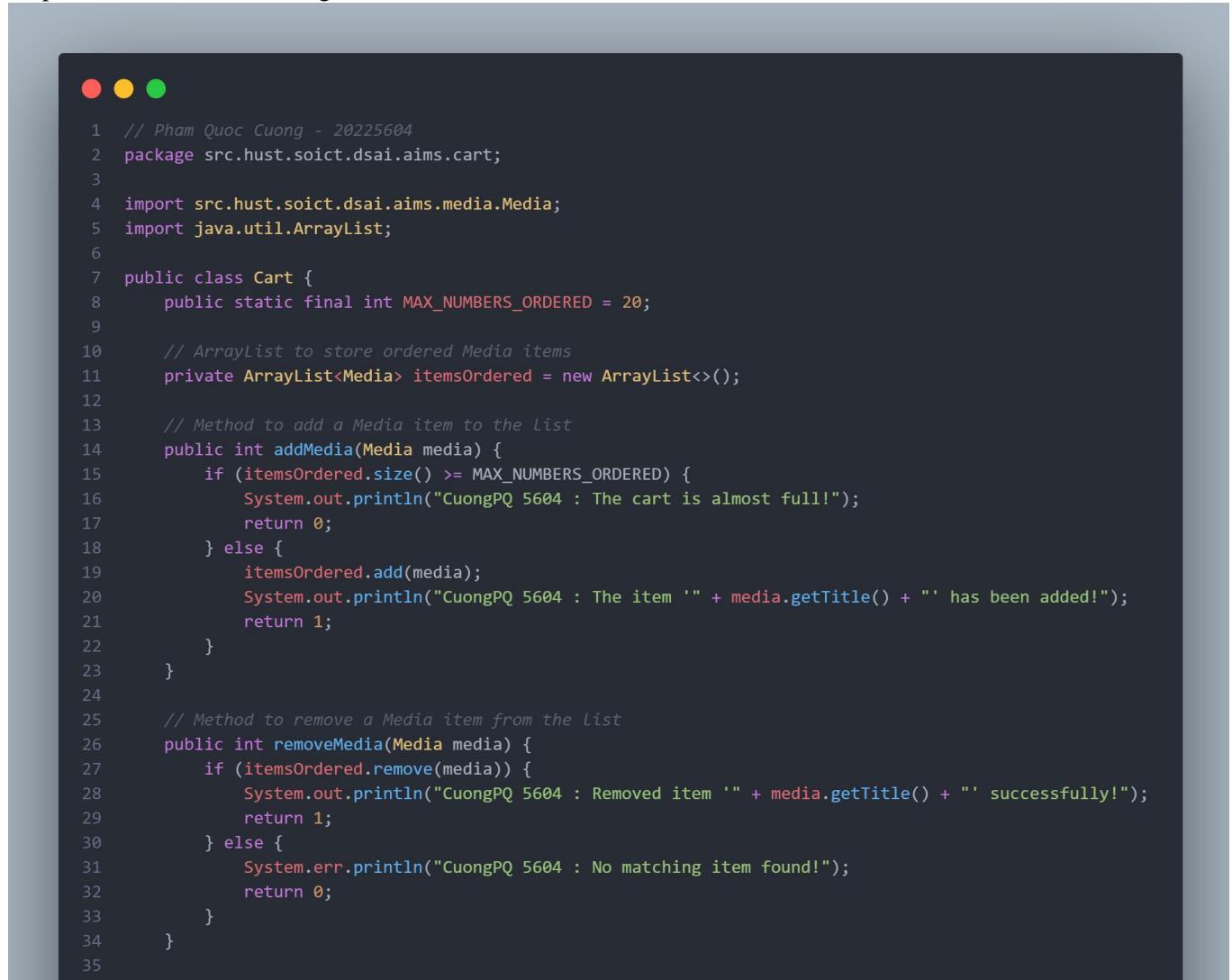


Figure 4.4: Method play() của CompactDisc

## 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.



```

1 // Pham Quoc Cuong - 20225604
2 package src.hust.soict.dsai.aims.cart;
3
4 import src.hust.soict.dsai.aims.media.Media;
5 import java.util.ArrayList;
6
7 public class Cart {
8     public static final int MAX_NUMBERS_ORDERED = 20;
9
10    // ArrayList to store ordered Media items
11    private ArrayList<Media> itemsOrdered = new ArrayList<>();
12
13    // Method to add a Media item to the list
14    public int addMedia(Media media) {
15        if (itemsOrdered.size() >= MAX_NUMBERS_ORDERED) {
16            System.out.println("CuongPQ 5604 : The cart is almost full!");
17            return 0;
18        } else {
19            itemsOrdered.add(media);
20            System.out.println("CuongPQ 5604 : The item '" + media.getTitle() + "' has been added!");
21            return 1;
22        }
23    }
24
25    // Method to remove a Media item from the List
26    public int removeMedia(Media media) {
27        if (itemsOrdered.remove(media)) {
28            System.out.println("CuongPQ 5604 : Removed item '" + media.getTitle() + "' successfully!");
29            return 1;
30        } else {
31            System.err.println("CuongPQ 5604 : No matching item found!");
32            return 0;
33        }
34    }
35

```

Figure 5.1: Cart Class I

```
34     }
35
36     // Method to calculate the total cost of items in the cart
37     public float totalCost() {
38         float sum = 0.00f;
39         for (Media media : itemsOrdered) {
40             sum += media.getCost();
41         }
42         return sum;
43     }
44
45     // Display items in the cart
46     public void printCart() {
47         System.out.println("*****CART*****");
48         System.out.println("Ordered Items:");
49         for (int i = 0; i < itemsOrdered.size(); i++) {
50             System.out.println((i + 1) + ". " + itemsOrdered.get(i).toString());
51         }
52         System.out.println("Total cost: " + totalCost() + "$");
53         System.out.println("*****");
54     }
55
56     // Search by ID method
57     public Media searchById(int id) {
58         for (Media media : itemsOrdered) {
59             if (media != null && media.getId() == id) {
60                 return media;
61             }
62         }
63         return null;
64     }
65
66     // Search by title method
67     public Media searchByTitle(String title) {
68         for (Media media : itemsOrdered) {
69             if (media != null && media.isMatch(title)) {
70                 return media;
71             }
72         }
73         return null;
74     }
75
76 }
77 }
```

Figure 5.2: Cart Class

## 6 Update the Store class to work with Media



```

1 // Pham Quoc Cuong - 20225604
2 package src.hust.soict.dsai.aims.store;
3
4 import java.util.ArrayList;
5 import src.hust.soict.dsai.aims.media.Media;
6
7 public class Store {
8     // List of Media items in Store
9     private ArrayList<Media> itemsInStore = new ArrayList<>();
10
11    // Add a Media item to the store
12    public void addMedia(Media media) {
13        if (!itemsInStore.contains(media)) {
14            itemsInStore.add(media);
15            System.out.println("CuongPQ 5604 - '" + media.getTitle() + "' has been added to the store!");
16        } else {
17            System.out.println("CuongPQ 5604 - '" + media.getTitle() + "' already exists in the store!");
18        }
19    }
20
21    // Remove a Media item from the store
22    public void removeMedia(Media media) {
23        if (itemsInStore.contains(media)) {
24            itemsInStore.remove(media);
25            System.out.println("CuongPQ 5604 - '" + media.getTitle() + "' has been removed from the store!");
26        } else {
27            System.out.println("CuongPQ 5604 - There is no '" + media.getTitle() + "' in the store!");
28        }
29    }
30
31    // Search by ID method
32    public Media searchById(int id) {
33        for (Media media : itemsInStore) {
34            if (media != null && media.getId() == id) {
35                return media;
36            }
37        }
38        return null;
39    }
40
41    // Search by title method
42    public Media searchByTitle(String title) {
43        for (Media media : itemsInStore) {
44            if (media != null && media.isMatch(title)) {
45                return media;
46            }
47        }
48        return null;
49    }
50
51    // Print all items in the store
52    public void printStore() {
53        System.out.println("***** STORE *****");
54        if (itemsInStore.isEmpty()) {
55            System.out.println("CuongPQ 5604 - The store is currently empty.");
56        } else {
57            for (int i = 0; i < itemsInStore.size(); i++) {
58                System.out.println("CuongPQ 5604 - " + (i + 1) + ". " + itemsInStore.get(i).toString());
59            }
60        }
61        System.out.println("*****");
62    }
63 }
64

```

Figure 6.1: Store Class

## 7 Constructors of whole classes and parent classes

```
// Pham Quoc Cuong - 20225604          You, 1 second ago •
// Constructor
public Track(String title, int length) {
    this.title = title;
    this.length = length;
}
```

Figure 7.1: Constructor Track Class

```
// Constructor
public CompactDisc(String title) {
    super(title);
}
public CompactDisc(String title, String category, String director, int length, float cost) {
    super(title, category, director, length, cost);
}
public CompactDisc(String title, String category, String director, ArrayList<Track> tracks, float cost) {
    super(title, category, director, cost);
    this.tracks = tracks;
    this.setLength(this.getLength());
}
public CompactDisc(String title, String artist, ArrayList<Track> tracks) {
    super(title);
    this.artist = artist;
    this.tracks = tracks;
}
public CompactDisc(String title, String artist) {
    super(title);
    this.artist = artist;
}
public CompactDisc(String title, String category, String director, int length, float cost, String artist, ArrayList<Track> tracks) {
    super(title, category, director, length, cost);
    this.artist = artist;
    this.tracks = tracks;
}
```

Figure 7.2: Constructor CompactDisc Class

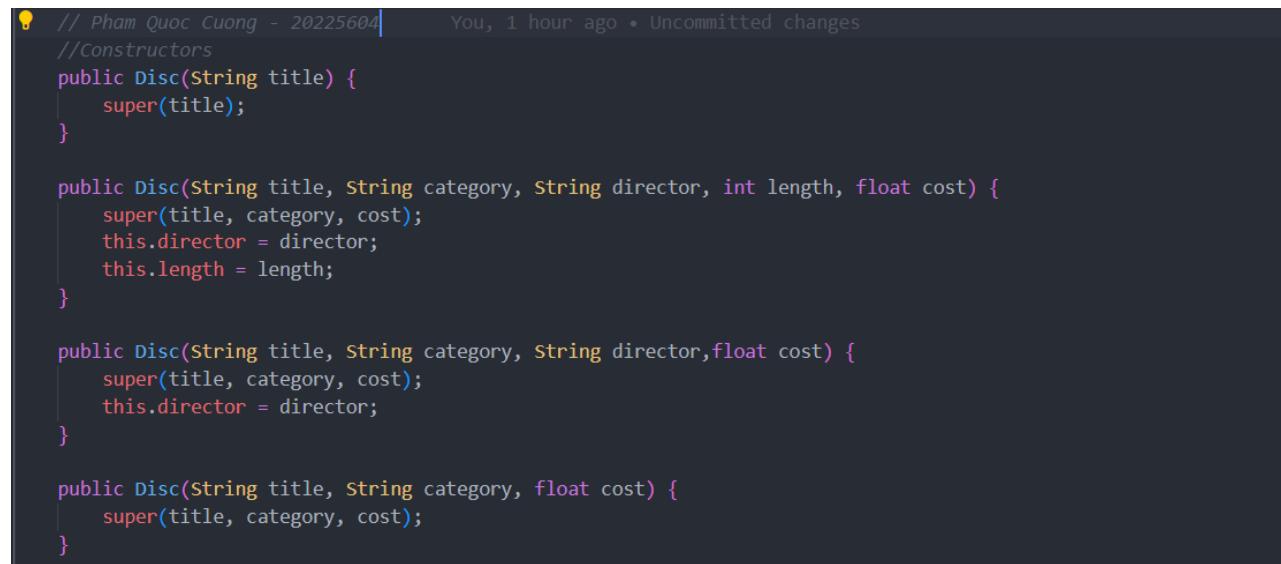
Lớp Disc kế thừa lớp Media, khi đó lớp Media là lớp cha, lớp Disc là lớp con.

```
// Constructor          You, 1 hour ago • Uncommitted changes
public Media() {
    super();
}

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

public Media(String title) {
    this.title = title;
    this.id = ++nbMedia;
}
```

Figure 7.3: Constructor Media Class



```

1 // Pham Quoc Cuong - 20225604      You, 1 hour ago • Uncommitted changes
2 //Constructors
3 public Disc(String title) {
4     super(title);
5 }
6
7 public Disc(String title, String category, String director, int length, float cost) {
8     super(title, category, cost);
9     this.director = director;
10    this.length = length;
11 }
12
13 public Disc(String title, String category, String director, float cost) {
14     super(title, category, cost);
15     this.director = director;
16 }
17
18 public Disc(String title, String category, float cost) {
19     super(title, category, cost);
20 }

```

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 lặp vào danh sách.



```

1 //Pham Quoc Cuong - 20225604
2 @Override
3 public boolean equals(Object obj) {
4     if (this == obj) { // Check if the references are the same
5         return true;
6     }
7     if (obj == null || getClass() != obj.getClass()) { // Check if obj is null or not the same class
8         return false;
9     }
10    Media media = (Media) obj; // Cast obj to Media
11    return title != null && title.equals(media.title); // Compare titles
12 }

```

Figure 8.1: Override equals in Media Class



```
1 // Pham Quoc Cuong - 20225604
2     @Override
3     public boolean equals(Object obj) {
4         if (this == obj) { // Check if the references are the same
5             return true;
6         }
7         if (obj == null || getClass() != obj.getClass()) { // Check if obj is null or not the same class
8             return false;
9         }
10        Track track = (Track) obj; // Cast obj to Track
11        return length == track.length && (title != null && title.equals(track.title)); // Compare title and Length
12    }
```

Figure 8.2: Override equals in Track Class

## 9 Polymorphism with `toString()` method

```
● ● ●

1 // Phạm Quốc Cường - 20225604
2 package src.hust.soict.dsai.test;
3
4 import java.util.ArrayList;
5 import java.util.List;
6
7 import src.hust.soict.dsai.aims.media.Book;
8 import src.hust.soict.dsai.aims.media.CompactDisc;
9 import src.hust.soict.dsai.aims.media.DigitalVideoDisc;
10 import src.hust.soict.dsai.aims.media.Media;
11
12 public class PolymorphismTest {
13     public static void main(String[] args) {
14         List<Media> media = new ArrayList<>();
15
16         CompactDisc cd = new CompactDisc("CD1");
17         DigitalVideoDisc dvd = new DigitalVideoDisc("DVD1");
18         Book book = new Book("Book1");
19
20         media.add(cd);
21         media.add(book);
22         media.add(dvd);
23
24         for(Media m : media) {
25             System.out.println(m.toString());
26         }
27     }
28 }
29
```

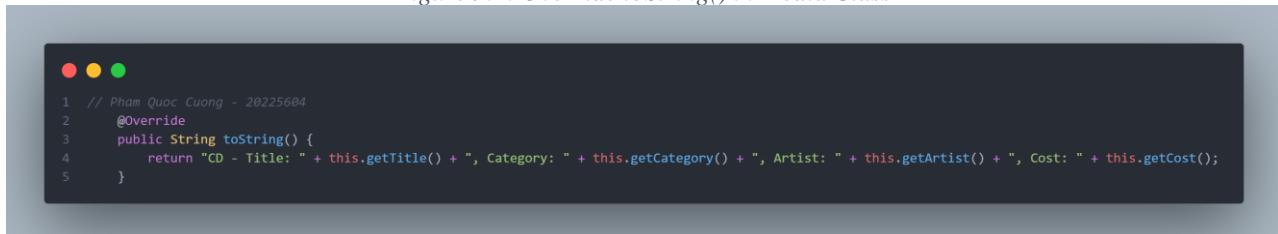
Figure 9.1: Code mô phỏng Polymorphism



```

1 //Pham Quoc Cuong - 20225604
2     @Override
3     public String toString() {
4         return "Media {id=" + id
5                 + ", title:'" + title + "'"
6                 + ", category: " + category + "'"
7                 + ", cost: $" + cost + "}";
8     }

```

Figure 9.2: Override `toString()` in `Media` Class


```

1 // Pham Quoc Cuong - 20225604
2     @Override
3     public String toString() {
4         return "CD - Title: " + this.getTitle() + ", Category: " + this.getCategory() + ", Artist: " + this.getArtist() + ", Cost: " + this.getCost();
5     }

```

Figure 9.2: Override `toString()` in `CompactDisc`


```

1 // Pham Quoc Cuong - 20225604
2     @Override
3     public String toString() {
4         return "DVD - Title: " + this.getTitle() + ", Category: " + this.getCategory() + ", Cost: " + this.getCost();
5     }

```

Figure 9.2: Override `toString()` in `DigitalVideoDisc`


```

1 // Pham Quoc Cuong - 20225604
2     @Override
3     public String toString() {
4         return "Book - Title: " + this.getTitle() + ", Category: " + this.getCategory() + ", Cost: " + this.getCost();
5     }

```

Figure 9.2: Override `toString()` in `Book`

Kết quả

```
CD - Title: CD1, Category: null, Artist: null, Cost: 0.0
Book - Title: Book1, Category: null, Cost: 0.0
DVD - Title: DVD1, Category: null, Cost: 0.0
```

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

```
// Pham Quoc Cuong - 20225604      You, 3 seconds ago • Uncommitted changes
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



```
1 // Pham Quoc Cuong - 20225604
2 package src.hust.soict.dsai.aims.media;
3
4 import java.util.Comparator;
5
6 public class MediaComparatorByCostTitle implements Comparator<Media> {
7     @Override
8     public int compare(Media o1, Media o2) {
9         if(o1.getCost() != o2.getCost()) {
10             return (int) (o1.getCost() - o2.getCost());
11         } else {
12             return o1.getTitle().compareTo(o2.getTitle());
13         }
14     }
15 }
16
```

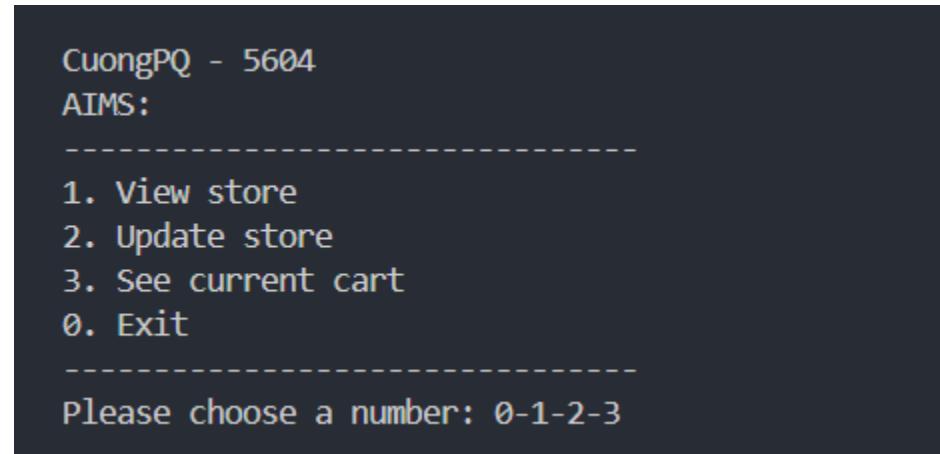
Figure 10.2: MediaComparatorByCostTitle Class



```
1 // Pham Quoc Cuong - 20225604
2 package src.hust.soict.dsai.aims.media;
3
4 import java.util.Comparator;
5 public class MediaComparatorByTitleCost implements Comparator<Media> {
6     @Override
7     public int compare(Media o1, Media o2) {
8         if(o1.getTitle().compareTo(o2.getTitle()) != 0) {
9             return o1.getTitle().compareTo(o2.getTitle());
10        } else {
11            if(o1.getCost() > o2.getCost()) return 1;
12            else return -1;
13        }
14    }
15 }
```

Figure 10.3: MediaComparatorByTitleCost Class

## 11 Create a complete console application in the Aims class



```
CuongPQ - 5604
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

```
Please choose a number: 0-1-2-3

1
***** STORE *****
CuongPQ 5604 - 1. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5
CuongPQ 5604 - 2. DVD - Title: Tay Du Ky, Category: Fantasy, Cost: 18.55
CuongPQ 5604 - 3. Book - Title: Harry Potter, Category: Novel, Cost: 25.2
*****
CuongPQ - 5604: 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

```
CuongPQ - 5604: 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  
  
1  
CuongPQ - 5604: Enter media's title: Tom & Jerry  
CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5  
CuongPQ - 5604: Options:  
-----  
1. Add to cart  
2. Play  
0. Back  
-----  
Please choose a number: 0-1-2
```

Figure 11.3: See a media's details

```
CuongPQ - 5604: Options:  
-----  
1. Add to cart  
2. Play  
0. Back  
-----  
Please choose a number: 0-1-2  
1  
CuongPQ 5604 : The item 'Tom & Jerry' has been added!
```

Figure 11.4: Thêm vào Cart

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

```
CuongPQ - 5604: 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  
  
2  
Enter media's title: Tom & Jerry  
CuongPQ 5604 : The item 'Tom & Jerry' has been added!
```

Figure 11.5: Thêm media vào Cart

### 11.1.3 Người dùng chọn 3: Play a media

```
CuongPQ - 5604: 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  
  
3  
Enter media's title: Tom & Jerry  
Playing CD: Tom & Jerry  
Artist: null  
CD length: 7  
Tracks:  
Playing Track: Nhac Trinh (Tran Phuong Tuan)  
Track length: 3  
Playing Track: Tuyen tap Dam Vinh Hung  
Track length: 4
```

Figure 11.6: Play a media

#### 11.1.4 Người dùng chọn 4: See current cart

```

-----
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
*****
CART*****
Ordered Items:
1. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5
2. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5
3. DVD - Title: Tay Du Ky, Category: Fantasy, Cost: 18.55
Total cost: 69.55$
*****
CuongPQ - 5604: 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
1. Sort by title cost
2. Sort by cost title
-----
your option:

1
*****
CART*****
Ordered Items:
1. DVD - Title: Tay Du Ky, Category: Fantasy, Cost: 18.55
2. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5
3. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5
Total cost: 69.55$
*****
```

Figure 11.7: See current cart after sort

## 11.2 Người dùng chọn 2: Update store

```
CuongPQ - 5604
AIMS:
-----
1. View store
2. Update store
3. See current cart
0. Exit
-----
Please choose a number: 0-1-2-3

2
CuongPQ - 5604: Update Store Menu:
=====
1. Add Media
2. Delete Media
3. Update Media in Store
0. Back
=====
Option:
```

Figure 11.8: Vào Trang Update Store

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

```
CuongPQ - 5604: Update Store Menu:  
=====  
1. Add Media  
2. Delete Media  
3. Update Media in Store  
0. Back  
=====  
Option:  
  
1  
CuongPQ - 5604: Add Media Menu:  
1. DigitalVideoDisc  
2. CompactDisc  
3. Book  
----  
-> Your type:  
3  
CuongPQ - 5604: Enter title: Giai Tich 1  
CuongPQ - 5604: Enter category: Giai Tich  
CuongPQ - 5604: Enter cost: 15  
CuongPQ - 5604: Enter author's name (Enter nothing to skip): Cuong  
CuongPQ - 5604: Enter author's name (Enter nothing to skip):  
CuongPQ 5604 - 'Giai Tich 1' has been added to the store!
```

Figure 11.9: Add a media to store

=> Kết quả sau khi thêm

```
***** STORE *****
CuongPQ 5604 - 1. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5
CuongPQ 5604 - 2. DVD - Title: Tay Du Ky, Category: Fantasy, Cost: 18.55
CuongPQ 5604 - 3. Book - Title: Harry Potter, Category: Novel, Cost: 25.2
CuongPQ 5604 - 4. Book - Title: Giai Tich 1, Category: Giai Tich, Cost: 15.0
*****
```

Figure 11.10: Result after add media to store

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

```
CuongPQ - 5604: Update Store Menu:
=====
1. Add Media
2. Delete Media
3. Update Media in Store
0. Back
=====
Option:
2
CuongPQ - 5604: Enter item's title: Giai Tich 1
CuongPQ - 5604: Book Giai Tich 1 has been deleted from the store!
```

Figure 11.11: Remove a media from the store

=> Kết quả sau khi remove

```
***** STORE *****
CuongPQ 5604 - 1. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5
CuongPQ 5604 - 2. DVD - Title: Tay Du Ky, Category: Fantasy, Cost: 18.55
CuongPQ 5604 - 3. Book - Title: Harry Potter, Category: Novel, Cost: 25.2
*****
```

Figure 11.12: Result after remove a media

### 11.3 Người dùng chọn 4: See current cart

```
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
*****
CART*****
Ordered Items:
1. DVD - Title: Tay Du Ky, Category: Fantasy, Cost: 18.55
2. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5
3. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5
Total cost: 69.55$
*****
CuongPQ - 5604: 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
```

Figure 11.13: Vào trang See current cart

Giả sử lúc này trong Cart sẽ có các Media sau

```
*****CART*****
```

Ordered Items:

1. DVD - Title: Tay Du Ky, Category: Fantasy, Cost: 18.55
  2. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5
  3. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5
- Total cost: 69.55\$

*Figure 11.14: Media in Cart*

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

```
*****CART*****
```

Ordered Items:

1. DVD - Title: Tay Du Ky, Category: Fantasy, Cost: 18.55
  2. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5
  3. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5
- Total cost: 69.55\$

CuongPQ - 5604: 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

1. Filter by title
  2. Filter by id
- 

Your option:

2

Enter id:

1

DVD - Title: Tay Du Ky, Category: Fantasy, Cost: 18.55

*Figure 11.15: Filter Cart By id*

```
CuongPQ - 5604: 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  
1. Filter by title  
2. Filter by id  
-----  
Your option:  
1  
Enter title:  
Tay Du Ky  
DVD - Title: Tay Du Ky, Category: Fantasy, Cost: 18.55
```

Figure 11.16: Filter Cart By Title

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

```
CuongPQ - 5604: 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  
1. Sort by title cost  
2. Sort by cost title  
-----  
your option:  
2  
*****CART*****  
Ordered Items:  
1. DVD - Title: Tay Du Ky, Category: Fantasy, Cost: 18.55  
2. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5  
3. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5  
Total cost: 69.55$  
*****
```

Figure 11.17: Sort Cart By Title

```
CuongPQ - 5604: 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  
1. Sort by title cost  
2. Sort by cost title  
-----  
your option:  
1  
*****CART*****  
Ordered Items:  
1. DVD - Title: Tay Du Ky, Category: Fantasy, Cost: 18.55  
2. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5  
3. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5  
Total cost: 69.55$  
*****
```

Figure 11.18: Sort Cart By Cost

## 11.3.3 Người dùng chọn 3: Remove media from cart

```
CuongPQ - 5604: 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 media's title: Tom & Jerry
CuongPQ 5604 : Removed item 'Tom & Jerry' successfully!"
```

Figure 11.19: Remove media by title

=> Kết quả

```
CuongPQ - 5604: 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
*****CART*****
Ordered Items:
1. DVD - Title: Tay Du Ky, Category: Fantasy, Cost: 18.55
2. CD - Title: Tom & Jerry, Category: Cartoon, Artist: null, Cost: 25.5
Total cost: 44.05$
*****
```

Figure 11.20: Result after remove media in cart

#### 11.3.4 Người dùng chọn 4: Play a media

```
CuongPQ - 5604: 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 media's title: Tom & Jerry
Playing CD: Tom & Jerry
Artist: null
CD length: 7
Tracks:
Playing Track: Nhac Trinh (Tran Phuong Tuan)
Track length: 3
Playing Track: Tuyen tap Dam Vinh Hung
Track length: 4
```

Figure 11.21: Play a media in cart

#### 11.3.5 Người dùng chọn 5: Place order

```
CuongPQ - 5604: 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
Your cart have been paid
Thanks for using our service
```

Figure 11.22: Order

=> Kết quả sau khi order

```
CuongPQ - 5604: 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  
1. Sort by title cost  
2. Sort by cost title  
-----  
your option:  
1  
*****CART*****  
Ordered Items:  
Total cost: 0.0$  
*****
```

Figure 11.23: Result after order

## 12 Class Diagram

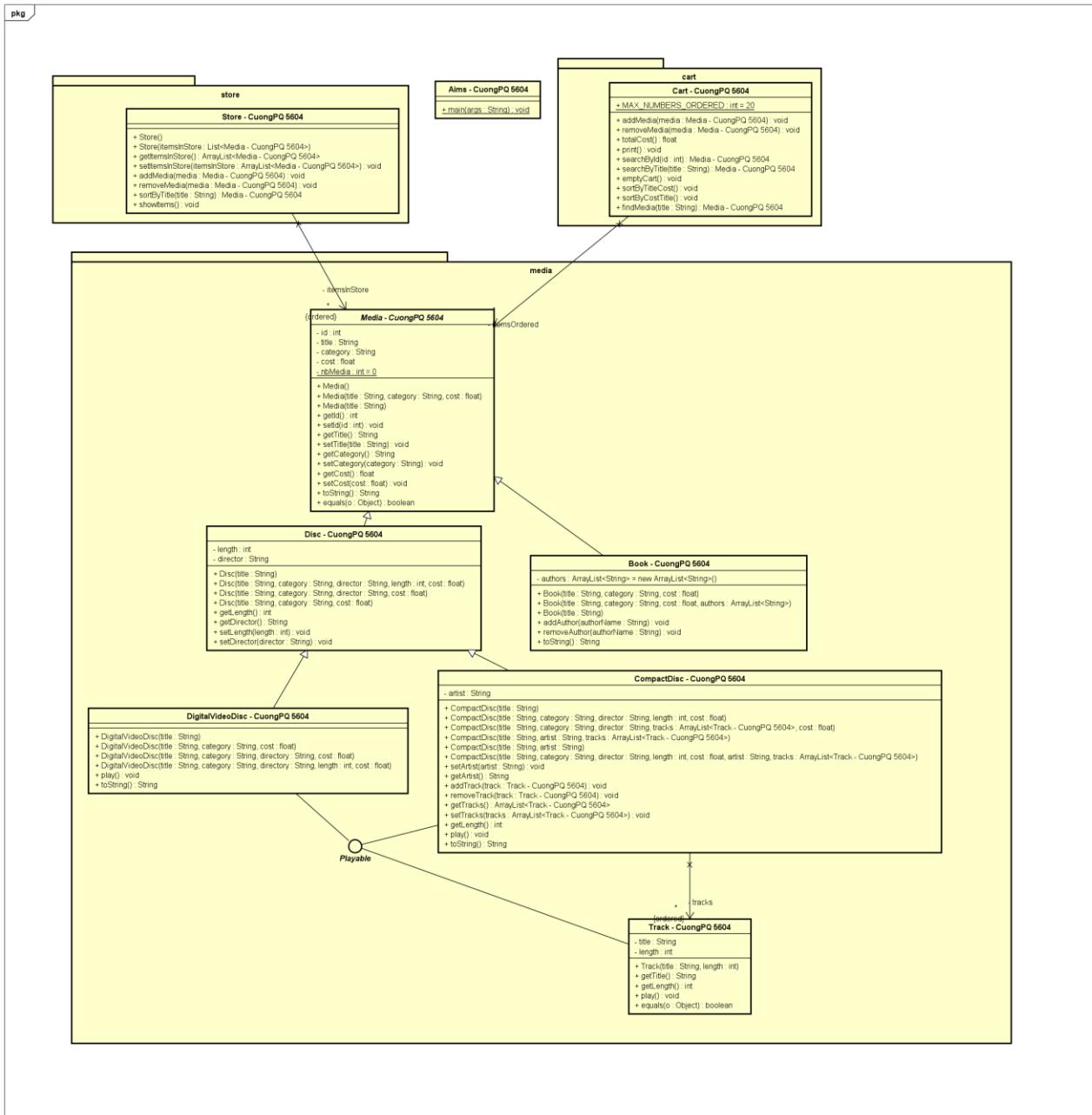


Figure 12.1: Class Diagram

## 13 UseCase Diagram

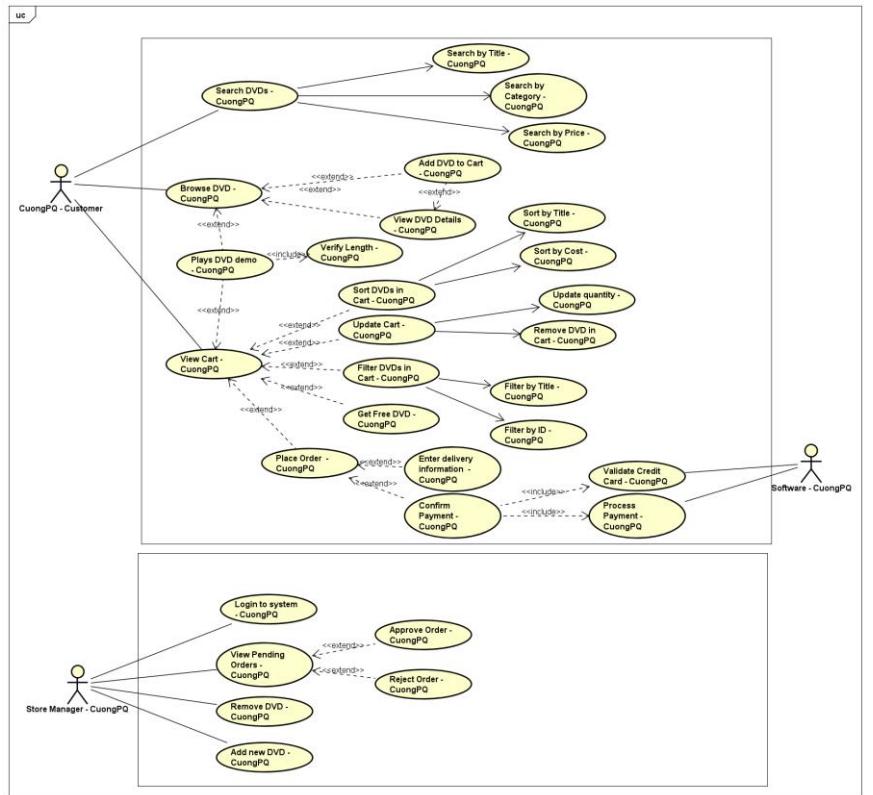


Figure 13.1: UseCase Diagram

## 14 Answer Question

Alternatively, to compare items in the cart, instead of using Comparator, we can use the Comparable interface and override the compareTo() method. You can refer to the Java docs to see the information of this interface. Suppose we are taking this Comparable interface approach.

- What class should implement the Comparable interface?
- In those classes, how should you implement the compareTo() method be to reflect the ordering that we want?
- 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?
- Suppose the DVDs has 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?

**Answer:**

- Lớp Media nên triển khai giao diện Comparable vì nó là lớp cơ sở cho các loại phương tiện khác như Book, DVD, và CD. Điều này đảm bảo rằng tất cả các phương tiện sẽ tuân theo một quy tắc sắp xếp mặc định.
  - Trong lớp Media, phương thức compareTo() cần được ghi đè để thể hiện quy tắc sắp xếp mặc định. Chẳng hạn, phương thức này có thể được thiết kế để sắp xếp theo thứ tự tiêu đề trước, sau đó đến giá cả.
  - Giao diện Comparable chỉ hỗ trợ **một** quy tắc sắp xếp, vì vậy không thể áp dụng hai quy tắc sắp xếp khác nhau (như "theo tiêu đề rồi giá cả" và "theo giá cả rồi tiêu đề") cùng một lúc khi chỉ sử dụng giao diện này. Để xử lý thêm các quy tắc sắp xếp khác, bạn cần sử dụng các lớp Comparator.
  - Để áp dụng quy tắc sắp xếp riêng cho lớp DigitalVideoDisc, bạn cần ghi đè phương thức compareTo() trong lớp này. Quy tắc có thể được triển khai như sau: so sánh theo tiêu đề, sau đó là độ dài giảm dần, và cuối cùng là giá cả. Lớp DigitalVideoDisc sẽ tự định nghĩa quy tắc này trong phương thức compareTo() của mình.