

# Report OOP – Lab04

## Inheritance and Polymorphism

Kỳ 2024-1

- **Họ và tên:** Lê Hải Yến
- **MSSV:** 20225780
- **Mã lớp:** 744520

## Contents

<b>1</b>	<b>New Written Code</b>	<b>2</b>
1.1	Create the Book class . . . . .	2
1.2	Creating the abstract Media class . . . . .	3
1.3	Creating the CompactDisc class . . . . .	5
1.3.1	Create the Disc class extending the Media class . . . . .	6
1.3.2	Create the Track class . . . . .	7
1.3.3	Open the CompactDisc class . . . . .	8
1.4	Create the Playable interface . . . . .	9
1.5	Update the Cart class to work with Media.....	10
1.6	Update the Store class to work with Media.....	11
1.7	Update UML and Constructors of whole classes and parent classes.....	12
1.8	Unique item in a list .....	13
<b>2</b>	<b>UML Diagram Final</b>	<b>14</b>
2.1	Use-Case Diagram.....	14
2.2	Class Diagram.....	14
<b>3</b>	<b>Answer Questions</b>	<b>15</b>

# 1 New Written Code

## 1.1 Create the Book class

```
1 package hust.soict.ITE6.aims.media;
2
3 import java.util.*;
4
5 public class Book {
6
7     private int id;
8     private String title;
9     private String category;
10    private float cost;
11    private List<String> authors = new ArrayList<String>();
12
13    public int getId() {
14        return id;
15    }
16    public void setId(int id) {
17        this.id = id;
18    }
19
20    public String getTitle() {
21        return title;
22    }
23    public void setTitle(String title) {
24        this.title = title;
25    }
26
27    public String getCategory() {
28        return category;
29    }
30    public void setCategory(String category) {
31        this.category = category;
32    }
33
34    public float getCost() {
35        return cost;
36    }
37    public void setCost(float cost) {
38        this.cost = cost;
39    }
40
41    public Book(String title) {
42        this.title = title;
43    }
44    public Book(String title, String category) {
45        this.title = title;
46        this.category = category;
47    }
48    public Book(String title, String category, float cost) {
49        this.title = title;
50        this.category = category;
51        this.cost = cost;
52    }
53
54    public void addAuthor(String authorName) {
55        if (!authors.contains(authorName)) {
56            authors.add(authorName);
57            System.out.println("Author added successfully.");
58        } else {
59            System.out.println("This author is already in the list!");
60        }
61    }
62
63    public void removeAuthor(String authorName) {
64        if (authors.contains(authorName)) {
65            authors.remove(authorName);
66            System.out.println("Author removed successfully.");
67        } else {
68            System.out.println("No author with this name found in the list!");
69        }
70    }
71
72 }
```

## 1.2 Creating the abstract Media class

- Đây sẽ là lớp cha để các lớp DigitalVideoDisc, Book kế thừa.
- Tạo ra lớp Media:

```
1 package hust.soict.ITE6.aims.media;
2
3 public abstract class Media {
4     private int id;
5     private String title;
6     private String category;
7     private float cost;
8     private static int nbMedia = 0;
9
10    public int getId() {
11        return id;
12    }
13
14    public void setId(int id) {
15        this.id = id;
16    }
17
18    public String getTitle() {
19        return title;
20    }
21
22    public void setTitle(String title) {
23        this.title = title;
24    }
25
26    public String getCategory() {
27        return category;
28    }
29
30    public void setCategory(String category) {
31        this.category = category;
32    }
33
34    public float getCost() {
35        return cost;
36    }
37
38    public void setCost(float cost) {
39        this.cost = cost;
40    }
41
42    public Media(String title) {
43        this.title = title;
44        this.id = ++nbMedia;
45    }
46
47    public Media(String title, String category) {
48        this.title = title;
49        this.category = category;
50        this.id = ++nbMedia;
51    }
52
53    public Media(String title, String category, float cost) {
54        this.title = title;
55        this.category = category;
56        this.cost = cost;
57        this.id = ++nbMedia;
58    }
59 }
60
61
```

- Xóa các trường và phương thức từ các lớp Book và DigitalVideoDisc
- Chuyển DigitalVideoDisc vào package hust.soict.ITE6.aims.media và xóa package hust.soict.ITE6.aims.disc

```
hust.soict.ITE6.aims.media
> Book.java
> DigitalVideoDiscLHV.java
> Media.java
```

- Extend lớp Media cho cả Book và DigitalVideoDisc
- Lớp Book

```

1 package hust.soict.ITE6.aims.media;
2
3 import java.util.*;
4
5 public class Book extends Media {
6
7     private List<String> authors = new ArrayList<String>();
8
9     public Book(String title) {
10         super(title);
11     }
12
13     public Book(String title, String category) {
14         super(title, category);
15     }
16
17     public Book(String title, String category, float cost) {
18         super(title, category, cost);
19     }
20
21     public void addAuthor(String authorName) {
22         if (!authors.contains(authorName)) {
23             authors.add(authorName);
24             System.out.println("Author added successfully.");
25         } else {
26             System.out.println("This author is already in the list!");
27         }
28     }
29
30     public void removeAuthor(String authorName) {
31         if (authors.contains(authorName)) {
32             authors.remove(authorName);
33             System.out.println("Author removed successfully.");
34         } else {
35             System.out.println("No author with this name found in the list!");
36         }
37     }
38 }
39

```

- Lớp DigitalVideoDisc

```

1 package hust.soict ITE6.aims.media;
2
3 public class DigitalVideoDiscLHY extends Media {
4     private int length;
5     private String director;
6
7     public int getLengthLHY() {
8         return this.length;
9     }
10
11    public String getDirectorLHY() {
12        return this.director;
13    }
14
15    public boolean isMatch(String keyword) {
16        return this.getTitle().toLowerCase().contains(keyword.toLowerCase());
17    }
18
19    public DigitalVideoDiscLHY(String title) {
20        super(title);
21    }
22
23    public DigitalVideoDiscLHY(String title, String category, float cost) {
24        super(title, category, cost);
25    }
26
27    public DigitalVideoDiscLHY(String director, String category, String title, float cost) {
28        super(title, category, cost);
29        this.director = director;
30    }
31
32    public DigitalVideoDiscLHY(String title, String category, String director, int length, float cost) {
33        super(title, category, cost);
34        this.director = director;
35        this.length = length;
36    }
37
38    public String toString() {
39        return this.getId() + ". DVD: " + this.getTitle() +
40            " - Category: " + this.getCategory() +
41            " - Director: " + this.director +
42            " - DVD length: " + this.length +
43            " - Cost: " + this.getCost() + "$";
44    }
45 }
46 }

```

### 1.3 Creating the CompactDisc class

### 1.3.1 Create the Disc class extending the Media class

- Tạo lớp Disc có 2 trường length và director.
- DigitalVideoDisc và CompactDisc kế thừa từ lớp Disc.
- Lớp Disc

```
1 package hust.soict.ITE6.aims.media;
2
3 public class Disc extends Media {
4     private String director;
5     private int length;
6
7     public String getDirector() {
8         return director;
9     }
10
11     public void setDirector(String director) {
12         this.director = director;
13     }
14
15     public int getLength() {
16         return length;
17     }
18
19     public void setLength(int length) {
20         this.length = length;
21     }
22
23     public Disc(String title) {
24         super(title);
25     }
26
27     public Disc(String title, String category) {
28         super(title, category);
29     }
30
31     public Disc(String title, String category, float cost) {
32         super(title, category, cost);
33     }
34
35     public Disc(String title, String category, float cost, String director) {
36         super(title, category, cost);
37         this.director = director;
38     }
39
40     public Disc(String title, String category, float cost, String director, int length) {
41         super(title, category, cost);
42         this.director = director;
43         this.length = length;
44     }
45 }
46 }
```

- Lớp DigitalVideoDisc

```

1 package hust.soict.ITE6.aims.media;
2
3 public class DigitalVideoDiscLHY extends Disc {
4     public boolean isMatch(String keyword) {
5         return this.getTitle().toLowerCase().contains(keyword.toLowerCase());
6     }
7
8     public DigitalVideoDiscLHY(String title) {
9         super(title);
10    }
11
12    public DigitalVideoDiscLHY(String title, String category, float cost) {
13        super(title, category, cost);
14    }
15
16    public DigitalVideoDiscLHY(String director, String category, String title, float cost) {
17        super(title, category, cost, director);
18    }
19
20    public DigitalVideoDiscLHY(String title, String category, float cost, String director, int length) {
21        super(title, category, cost, director, length);
22    }
23
24    public String toString() {
25        return this.getId() + ". DVD: " + this.getTitle() +
26            " - Category: " + this.getCategory() +
27            " - Director: " + this.getDirector() +
28            " - DVD length: " + this.getLength() +
29            " - Cost: " + this.getCost() + "$";
30    }
31
32 }

```

### 1.3.2 Create the Track class

- Tạo lớp Track để mô phỏng một bản nhạc trên đĩa CD và lưu trữ thông tin bao gồm tiêu đề title và thời lượng length của bản nhạc.
- Lớp Track

```

1 package hust.soict.ITE6.aims.media;
2
3 public class Track {
4     private String title;
5     private int length;
6
7     public String getTitle() {
8         return title;
9     }
10
11    public void setTitle(String title) {
12        this.title = title;
13    }
14
15    public int getLength() {
16        return length;
17    }
18
19    public void setLength(int length) {
20        this.length = length;
21    }
22
23    public Track(String title, int length) {
24        this.title = title;
25        this.length = length;
26    }
27
28 }

```

### 1.3.3 Open the CompactDisc class

- Thêm trường artist và danh sách tracks.
- Tạo phương thức addTrack() và removeTrack().
- Tạo phương thức getLength() để tính tổng độ dài các bài nhạc trong tracks của CD.

```
1 package hust.soict.ITE6.aims.media;
2
3 import java.util.ArrayList;
4
5 public class CompactDisc extends Disc {
6     private String artist;
7     private ArrayList<Track> tracks;
8
9     public String getArtist() {
10         return artist;
11     }
12
13     public void setArtist(String artist) {
14         this.artist = artist;
15     }
16
17     public CompactDisc(String title, String category, float cost, String director, int length, String artist) {
18         super(title, category, cost, director, length);
19         this.artist = artist;
20     }
21
22     public CompactDisc(String title, String category, float cost, String director) {
23         super(title, category, cost, director);
24     }
25
26     public CompactDisc(String title, String category, float cost) {
27         super(title, category, cost);
28     }
29
30     public CompactDisc(String title, String category) {
31         super(title, category);
32     }
33
34     public CompactDisc(String title) {
35         super(title);
36     }
37
38     public void addTrack(Track track) {
39         if (tracks.contains(track)) {
40             System.out.println("Track is already in the list!");
41         } else {
42             tracks.add(track);
43             System.out.println("Track added successfully.");
44         }
45     }
46
47     public void removeTrack(Track track) {
48         if (tracks.contains(track)) {
49             tracks.remove(track);
50             System.out.println("Track removed successfully.");
51         } else {
52             System.out.println("Track not found in the list!");
53         }
54     }
55
56     public int getLength() {
57         int totalLength = 0;
58         for (int i = 0; i < tracks.size(); i++) {
59             totalLength += tracks.get(i).getLength();
60         }
61         return totalLength;
62     }
63 }
64
```



## 1.4 Create the Playable interface

- Interface Playable cho phép các lớp chỉ định chúng triển khai phương thức play()
- Tạo Interface Playable và thêm phương thức nguyên mẫu

```
1 package hust.soict.ITE6.aims.media;  
2  
3 public interface Playable {  
4     public void play();  
5 }
```

- Triển khai interface Playable và phương thức play() cho lớp DigitalVideoDisc.

```
public class DigitalVideoDiscLHY extends Disc implements Playable {  
    public void play() {  
        System.out.println("Playing DVD: " + this.getTitle());  
        System.out.println("DVD length: " + this.getLength());  
    }  
}
```

- Triển khai interface Playable và phương thức play() cho lớp Track.

```
public class Track implements Playable {  
    public void play() {  
        System.out.println("Playing track: " + this.getTitle());  
        System.out.println("Track length: " + this.getLength());  
    }  
}
```

- Triển khai interface Playable và phương thức play() cho lớp CompactDisc.

```
public class CompactDisc extends Disc implements Playable {  
    public void play() {  
        System.out.println("Playing CD: " + this.getTitle());  
        System.out.println("Artist: " + this.getArtist());  
        System.out.println("CD length: " + this.getLength());  
        for (Track track : tracks) {  
            track.play();  
        }  
    }  
}
```

## 1.5 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.
- Xóa addDigitalVideoDisc() và removeDigitalVideoDisc() thay bằng các phương thức tổng quát hơn là addMedia() và removeMedia() đồng thời cập nhật totalCost() để tính tổng chi phí tất cả Media

```
1 package hust.soict.ITE6.aims.cart;
2 import java.util.*;
3
4 import hust.soict.ITE6.aims.media.Media;
5
6 public class CartLHY {
7     public static final int MAX_NUMBERS_ORDERED = 20;
8     private ArrayList<Media> itemsOrdered = new ArrayList<Media>();
9
10    public void addMedia(Media media) {
11        if (itemsOrdered.size() >= MAX_NUMBERS_ORDERED) {
12            System.out.println("The cart is almost full!");
13        } else {
14            itemsOrdered.add(media);
15            System.out.println("Added: " + media.getTitle());
16        }
17    }
18
19    public void addMedia(Media[] mediaList) {
20        if (mediaList == null || mediaList.length == 0) {
21            System.out.println("The media list is empty.");
22            return;
23        }
24        for (Media media : mediaList) {
25            if (itemsOrdered.size() >= MAX_NUMBERS_ORDERED) {
26                System.out.println("The cart is full! Cannot add more items.");
27                break;
28            }
29            itemsOrdered.add(media);
30            System.out.println("The media " + media.getTitle() + " has been added.");
31        }
32    }
33
34    public void addMedia(Media firstMedia, Media... otherMedia) {
35        if (itemsOrdered.size() < MAX_NUMBERS_ORDERED) {
36            itemsOrdered.add(firstMedia);
37            System.out.println("Added: " + firstMedia.getTitle());
38        } else {
39            System.out.println("The cart is full! Cannot add more items.");
40            return;
41        }
42
43        for (Media media : otherMedia) {
44            if (itemsOrdered.size() >= MAX_NUMBERS_ORDERED) {
45                System.out.println("The cart is full! Cannot add more items.");
46                break;
47            }
48            itemsOrdered.add(media);
49            System.out.println("Added: " + media.getTitle());
50        }
51    }
52
53    public void addMedia(Media media1, Media media2) {
54        addMedia(media1, media2);
55    }
56
57    public void removeMedia(Media media) {
58        if (itemsOrdered.isEmpty()) {
59            System.out.println("The cart is empty! Nothing to remove.");
60        } else {
61            if (itemsOrdered.remove(media)) {
62                System.out.println("Removed: " + media.getTitle());
63            } else {
64                System.out.println("Media not found in the cart!");
65            }
66        }
67    }
68 }
```

```

69 public float totalCost() {
70     float totalCost = 0;
71     for (Media media : itemsOrdered) {
72         totalCost += media.getCost();
73     }
74     return totalCost;
75 }
76
77 public void print() {
78     System.out.println("*****CART*****");
79     System.out.println("Ordered Items:");
80     int i = 0;
81     for (Media media : itemsOrdered) {
82         i += 1;
83         System.out.println(i + ". " + media);
84     }
85     System.out.println("Total cost: " + totalCost());
86     System.out.println("*****");
87 }
88
89 public void searchByID(int id) {
90     boolean isFound = false;
91     for (Media media : itemsOrdered) {
92         if (media.getId() == id) {
93             System.out.println("Found " + media);
94             isFound = true;
95         }
96     }
97     if (isFound == false) {
98         System.out.println("No DVDs were found with the given ID.");
99     }
100 }
101
102 public void searchByTitle(String keyword) {
103     boolean isMatchFound = false;
104     for (Media media : itemsOrdered) {
105         if (media.isMatch(keyword)) {
106             System.out.println("Found " + media);
107             isMatchFound = true;
108         }
109     }
110     if (isMatchFound == false) {
111         System.out.println("No DVDs found with the title containing \"" + keyword + "\".");
112     }
113 }
114 }
115 }

```

## 1.6 Update the Store class to work with Media

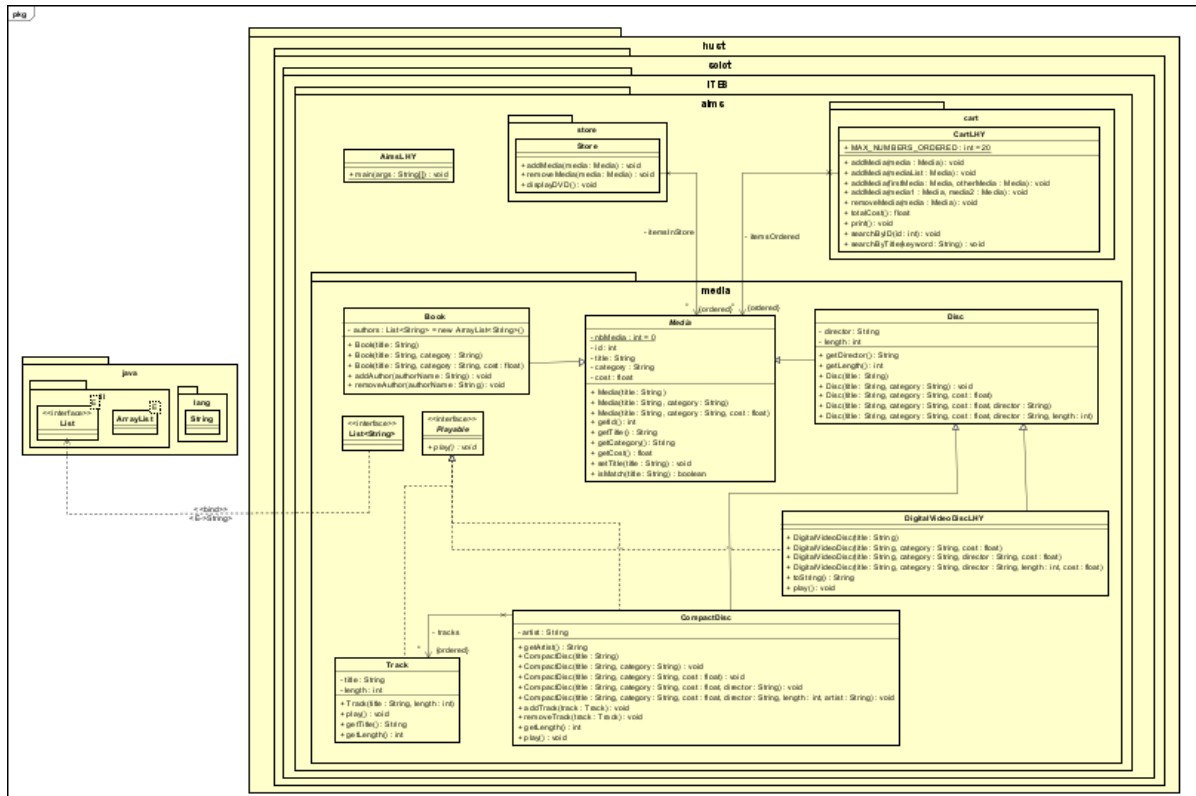
```

1 package hust.soict.ITE6.aims.store;
2 import java.util.ArrayList;
3 import hust.soict.ITE6.aims.media.Media;
4
5 public class Store {
6     private ArrayList<Media> itemsInStore = new ArrayList<Media>();
7
8     public void addMedia(Media media) {
9         if (!itemsInStore.contains(media)) {
10             itemsInStore.add(media);
11             System.out.println("Added: " + media.getTitle());
12         } else {
13             System.out.println(media.getTitle() + " is already in the store.");
14         }
15     }
16
17     public void removeMedia(Media media) {
18         if (itemsInStore.contains(media)) {
19             itemsInStore.remove(media);
20             System.out.println("Removed: " + media.getTitle());
21         } else {
22             System.out.println(media.getTitle() + " is not in the store.");
23         }
24     }
25
26     public void displayDVD() {
27         if (itemsInStore.isEmpty()) {
28             System.out.println("The store is currently empty. No media available.");
29         } else {
30             System.out.println("*****List of media in the store*****");
31             for (int i = 0; i < itemsInStore.size(); i++) {
32                 System.out.println((i + 1) + ". " + itemsInStore.get(i));
33             }
34             System.out.println("*****");
35         }
36     }
37 }

```

## 1.7 Update UML and Constructors of whole classes and parent classes

- Constructor ở các bước trước, xóa các setter không cần thiết
- Cập nhật Class Diagram



## 1.8 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.

- Lớp Media

```
public boolean equals(Object obj) {
    if (obj == this) {
        return true;
    }
    if (!(obj instanceof Media)) {
        return false;
    }
    Media other = (Media) obj;
    return this.getTitle().equals(other.getTitle());
}
```

- Lớp Track

```
public boolean equals(Object obj) {
    if (obj == this) {
        return true;
    }
    if (!(obj instanceof Track)) {
        return false;
    }
    Track other = (Track) obj;
    return this.getTitle().equals(other.getTitle()) && this.getLength() == other.getLength();
}
```

## 1.9 Polymorphism with toString() method

- Tùy vào từng loại đối tượng, override phương thức toString() thì sẽ hiện ra thông tin khác nhau tùy thuộc vào toString() tương ứng.

- Lớp Book

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

- Lớp Media

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

- Lớp DigitalVideoDisc

```
public String toString() {
    return this.getId() + ". DVD: " + this.getTitle() +
        " - Category: " + this.getCategory() +
        " - Director: " + this.getDirector() +
        " - DVD length: " + this.getLength() +
        " - Cost: " + this.getCost() + "$";
}
```

- Lớp CompactDisc

```
public String toString() {
    return "CD: " + this.getTitle() +
        " - Category: " + this.getCategory() +
        " - Artist: " + this.getArtist() +
        " - Length: " + this.getLength() +
        " - Cost: " + this.getCost() + "$";
}
```

- Kết quả

```

1 package hust.soict.ITE6.test.cart;
2 import hust.soict.ITE6.aims.cart.CartLHY;
3
4
5 public class CartTest {
6     public static void main(String[] args) {
7
8         CartLHY cart = new CartLHY();
9
10        DigitalVideoDiscLHY dvd1 = new DigitalVideoDiscLHY("The Lion King",
11            "Animation", 19.95f, "Roger Allers", 87);
12        cart.addMedia(dvd1);
13        DigitalVideoDiscLHY dvd2 = new DigitalVideoDiscLHY("Star War",
14            "Science Fiction", 24.95f, "George Lucas", 87);
15        cart.addMedia(dvd2);
16        DigitalVideoDiscLHY dvd3 = new DigitalVideoDiscLHY("Aladin",
17            "Animation", 18.99f);
18        cart.addMedia(dvd3);
19        DigitalVideoDiscLHY dvd4 = new DigitalVideoDiscLHY("Aladin",
20            "Animation", 18.99f);
21        cart.addMedia(dvd4);
22        cart.print();
23        cart.searchByID(1);
24        cart.searchByTitle("Star");
25        cart.searchByTitle("Baby");
26    }
27 }

```

---

```

Added: The Lion King
Added: Star War
Added: Aladin
Aladin is already in the cart!
*****CART*****
Ordered Items:
1. 1. DVD: The Lion King - Category: Animation - Director: Roger Allers - DVD length: 87 - Cost: 19.95$
2. 2. DVD: Star War - Category: Science Fiction - Director: George Lucas - DVD length: 87 - Cost: 24.95$
3. 3. DVD: Aladin - Category: Animation - Director: null - DVD length: 0 - Cost: 18.99$
Total cost: 63.89
*****
Found 1. DVD: The Lion King - Category: Animation - Director: Roger Allers - DVD length: 87 - Cost: 19.95$
Found 2. DVD: Star War - Category: Science Fiction - Director: George Lucas - DVD length: 87 - Cost: 24.95$
No DVDs found with the title containing "Baby".

```

- 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.
- Nhận xét: Do Media class chứa các thuộc tính chung và mỗi lớp con có thể có thuộc tính riêng (Ví dụ như class DVD có director và length) nên ta sẽ tiến hành override ở cả lớp cha và các lớp con và thu được kết quả như trong hình.

## 1.10 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
- Tạo 2 lớp Comparator

```

1 package hust.soict.ITE6.aims.media;
2 import java.util.Comparator;
3
4 public class MediaComparatorByCostTitle implements Comparator<Media> {
5     public int compare(Media m1, Media m2) {
6         int costComparison = Float.compare(m2.getCost(), m1.getCost());
7         if (costComparison != 0) {
8             return costComparison;
9         }
10        return m1.getTitle().compareTo(m2.getTitle());
11    }
12 }
13 }
14

```

```

1 package hust.soict.ITE6.aims.media;
2
3 import java.util.Comparator;
4
5 public class MediaComparatorByTitleCost implements Comparator<Media> {
6     public int compare(Media m1, Media m2) {
7         int titleComparison = m1.getTitle().compareTo(m2.getTitle());
8         if (titleComparison != 0) {
9             return titleComparison;
10        }
11        return Float.compare(m2.getCost(), m1.getCost());
12    }
13 }
14 }
15

```

- Thêm các Comparator làm thuộc tính của lớp Media

```

public abstract class Media implements Comparable<Media> {
    public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost();
    public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle();
}

```

- Tạo lớp MediaCompareTest

```

1 package hust.soict.ITE6.test.media;
2 import java.util.*;
3
4
5 public class MediaCompareTest {
6
7     public static void main(String[] args) {
8         List<Media> mediae = new ArrayList<Media>();
9
10        DigitalVideoDiscLHY dvd1 = new DigitalVideoDiscLHY("Inception", "Sci-Fi", 29.99f, "Christopher Nolan", 148);
11        DigitalVideoDiscLHY dvd2 = new DigitalVideoDiscLHY("Avatar", "Adventure", 24.50f, "James Cameron", 162);
12        DigitalVideoDiscLHY dvd3 = new DigitalVideoDiscLHY("Finding Nemo", "Animation", 15.75f, "Andrew Stanton", 100);
13
14        Book book1 = new Book("Sherlock Holmes: A Study in Scarlet", "Mystery", 10.99f);
15        Book book2 = new Book("Becoming", "Biography", 25.50f);
16        Book book3 = new Book("The Theory of Everything", "Science", 18.00f);
17        Book book4 = new Book("Dracula", "Horror", 8.99f);
18
19        CompactDisc cd = new CompactDisc("Taylor Swift - 1989", "Pop", 200.00f, "Taylor Swift");
20        Track track1 = new Track("Blank Space", 231);
21        Track track2 = new Track("Style", 245);
22        Track track3 = new Track("Shake It Off", 219);
23
24        cd.addTrack(track1);
25        cd.addTrack(track2);
26        cd.addTrack(track3);
27
28        mediae.add(dvd1);
29        mediae.add(dvd2);
30        mediae.add(dvd3);
31        mediae.add(book1);
32        mediae.add(book2);
33        mediae.add(book3);
34        mediae.add(book4);
35        mediae.add(cd);
36
37        System.out.println("\n***** SORT USING OVERRIDE compareTo() METHOD *****");
38        Collections.sort(mediae);
39        printMediaList(mediae);
40
41        System.out.println("\n***** SORT BY TITLE USING COMPARATOR *****");
42        Collections.sort(mediae, Media.COMPARE_BY_TITLE_COST);
43        printMediaList(mediae);
44
45        System.out.println("\n***** SORT BY COST USING COMPARATOR *****");
46        Collections.sort(mediae, Media.COMPARE_BY_COST_TITLE);
47        printMediaList(mediae);
48    }
49
50    private static void printMediaList(List<Media> mediaList) {
51        for (Media media : mediaList) {
52            System.out.println(media.toString());
53        }
54        System.out.println("*****");
55    }
56 }

```

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

```

1 package hust.soict ITE6.aims;
2 import hust.soict ITE6.aims.cart.CartLHY;
3
4 public class AimsLHY {
5     private static Store store = new Store();
6     private static CartLHY cart = new CartLHY();
7
8     public static void initSetup() {
9         store.addMedia(new DigitalVideoDiscLHY("Frozen", "Fantasy", 21.95f, "Chris Buck", 102));
10        store.addMedia(new DigitalVideoDiscLHY("Interstellar", "Science Fiction", 29.99f, "Christopher Nolan", 169));
11        store.addMedia(new DigitalVideoDiscLHY("Mulan", "Adventure", 20.50f));
12
13        store.addMedia(new Book("The Hound of the Baskervilles", "Mystery", 22.00f));
14        store.addMedia(new Book("Educated: A Memoir", "Biography", 210.00f));
15        store.addMedia(new Book("Astrophysics for People in a Hurry", "Science", 125.00f));
16
17        CompactDisc cd1 = new CompactDisc("Adele - 25", "Music", 1400.75f, "Adele");
18        cd1.addTrack(new Track("Hello", 295));
19        cd1.addTrack(new Track("When We Were Young", 270));
20        cd1.addTrack(new Track("Water Under the Bridge", 245));
21        store.addMedia(cd1);
22
23        CompactDisc cd2 = new CompactDisc("Infections of a Different Kind", "Music", 1800.50f, "Aurora");
24        cd2.addTrack(new Track("Queendom", 233+28));
25        cd2.addTrack(new Track("Forgotten Love", 187+30));
26        cd2.addTrack(new Track("Churchyard", 26*5+12));
27        store.addMedia(cd2);
28
29        CompactDisc cd3 = new CompactDisc("Changes", "Music", 1200.50f, "Justin Bieber");
30        cd3.addTrack(new Track("Intentions", 5*40+24));
31        cd3.addTrack(new Track("Yummy", 4*60+22));
32        store.addMedia(cd3);
33
34        clearConsole();
35    }
36
37    public static void main(String[] args) {
38        initSetup();
39        boolean exit = false;
40
41        while (!exit) {
42            showMenu();
43            Scanner scanner = new Scanner(System.in);
44            int option = scanner.nextInt();
45            scanner.nextLine();
46
47            switch (option) {
48                case 0 -> {
49                    exit = true;
50                    System.out.println("Exiting AIMS. Goodbye!");
51                }
52                case 1 -> {
53                    clearConsole();
54                    storeMenu(scanner);
55                }
56                case 2 -> {
57                    clearConsole();
58                    updateStoreMenu(scanner);
59                }
60                case 3 -> {
61                    clearConsole();
62                    cartMenu(scanner);
63                }
64                default -> {
65                    clearConsole();
66                    System.out.println("Invalid choice. Please try again.");
67                }
68            }
69        }
70    }
71
72    public static void clearConsole() {
73        for (int i=0; i<50; i++) {
74            System.out.println();
75        }
76    }
77
78 }

```



```

82 public static void showMenu() {
83     System.out.println("""
84         AIMS:
85         -----
86         1. View store
87         2. Update store
88         3. See current cart
89         0. Exit
90         -----
91         Please choose a number: 0-1-2-3
92         """);
93 }
94
95 public static void storeMenu(Scanner scanner) {
96     boolean back = false;
97     while(back == false) {
98         store.displayDVD();
99
100         System.out.println("""
101             Options:
102             -----
103             1. See a media's details
104             2. Add a media to cart
105             3. Play a media
106             4. See current cart
107             0. Back
108             -----
109             Please choose a number: 0-1-2-3-4
110             """);
111
112         int option = scanner.nextInt();
113         scanner.nextLine();
114
115         switch (option) {
116             case 0 -> {
117                 clearConsole();
118                 back = true;
119             }
120             case 1 -> {
121                 boolean foundDetails = false;
122                 while (foundDetails == false) {
123                     System.out.println("Enter the title of the media (type 0 to stop): ");
124                     String title = scanner.nextLine();
125                     if (title.equals("0")) {
126                         clearConsole();
127                         break;
128                     }
129                     Media media = store.search(title);
130                     if (media != null) {
131                         clearConsole();
132                         System.out.println("Details: ");
133                         System.out.println(media);
134                         mediaDetailsMenu(scanner, media);
135                         foundDetails = true;
136                     } else {
137                         System.out.println("***MEDIA NOT FOUND***");
138                     }
139                 }
140             }
141             case 2 -> {
142                 boolean foundToAdd = false;
143                 while (foundToAdd == false) {
144                     System.out.println("Enter the title of the media (type 0 to stop): ");
145                     String title = scanner.nextLine();
146                     if (title.equals("0")) {
147                         clearConsole();
148                         break;
149                     }
150                     Media media = store.search(title);
151                     if (media != null) {
152                         cart.addMedia(media);
153                         foundToAdd = true;
154                     } else {
155                         System.out.println("***MEDIA NOT FOUND***");
156                     }
157                 }
158             }
159             case 3 -> {
160                 boolean foundToPlay = false;
161                 while (foundToPlay == false) {
162                     System.out.println("Enter the title of the media (type 0 to stop): ");
163                     String title = scanner.nextLine();
164                     if (title.equals("0")) {
165                         clearConsole();
166                         break;
167                     }
168                 }
169             }
170         }
171     }
172 }

```

```

168         Media media = store.search(title);
169         if (media != null) {
170             if (media instanceof Disc || media instanceof CompactDisc) {
171                 media.play();
172             } else {
173                 System.out.println("This type of media is not supported!");
174             }
175             foundToPlay = true;
176         } else {
177             System.out.println("****MEDIA NOT FOUND****");
178         }
179     }
180 }
181 case 4 -> {
182     clearConsole();
183     cartMenu(scanner);
184 }
185 default -> {
186     clearConsole();
187     System.out.println("Invalid option, please choose again.");
188 }
189 }
190 }
191 }
192
193 public static void mediaDetailsMenu(Scanner scanner, Media media) {
194     boolean back = false;
195     while (back == false) {
196         System.out.println("""
197             Options:
198             -----
199             1. Add to cart
200             2. Play
201             0. Back
202             -----
203             Please choose a number: 0-1-2
204             """);
205
206         int option = scanner.nextInt();
207         scanner.nextLine();
208
209         switch (option) {
210             case 0 -> {
211                 clearConsole();
212                 back = true;
213             }
214             case 1 -> {
215                 cart.addMedia(media);
216             }
217             case 2 -> {
218                 if (media instanceof Disc || media instanceof CompactDisc) {
219                     media.play();
220                 } else {
221                     System.out.println("This type of media is not supported!");
222                 }
223             }
224             default -> {
225                 clearConsole();
226                 System.out.println("Invalid option, please choose again.");
227             }
228         }
229     }
230 }
231
232 public static void cartMenu(Scanner scanner) {
233     boolean back = false;
234     while (back == false) {
235         cart.print();
236         System.out.println("""
237             Options:
238             -----
239             1. Filter medias in cart
240             2. Sort medias in cart
241             3. Remove media from cart
242             4. Play a media
243             5. Place order
244             0. Back
245             -----
246             Please choose a number: 0-1-2-3-4-5
247             """);
248
249         int option = scanner.nextInt();
250         scanner.nextLine();
251         switch (option) {
252             case 0 -> {
253                 clearConsole();
254                 back = true;
255             }

```

```

256 case 1 -> {
257     System.out.println("Filter medias in cart by (1) id or (2) title:");
258     int filterOption = scanner.nextInt();
259     scanner.nextLine();
260     boolean found = false;
261     while (found == false) {
262         if (filterOption == 1) {
263             System.out.println("Enter the id to filter (type 0 to stop):");
264             int id = scanner.nextInt();
265             scanner.nextLine();
266             if (id == 0) {
267                 clearConsole();
268                 break;
269             }
270             cart.searchByID(id);
271             found = true;
272         } else if (filterOption == 2) {
273             System.out.println("Enter the title to filter (type 0 to stop):");
274             String title = scanner.nextLine();
275             if (title.equals("0")) {
276                 clearConsole();
277                 break;
278             }
279             cart.searchByTitle(title);
280             found = true;
281         } else if (filterOption == 0) {
282             clearConsole();
283             break;
284         } else {
285             System.out.println("Invalid option.");
286         }
287     }
288 }
289 case 2 -> {
290     System.out.println("Sort medias in cart by (1) title or (2) cost:");
291     int sortOption = scanner.nextInt();
292     scanner.nextLine();
293     if (sortOption == 1) {
294         cart.sortMediaByTitle();
295     } else if (sortOption == 2) {
296         cart.sortMediaByCost();
297     } else {
298         System.out.println("Invalid option.");
299     }
300 }
301 case 3 -> {
302     boolean foundToRemove = false;
303     while (foundToRemove == false) {
304         System.out.println("Enter the title of the media (type 0 to stop): ");
305         String title = scanner.nextLine();
306         if (title.equals("0")) {
307             clearConsole();
308             break;
309         }
310         Media media = cart.searchToRemove(title);
311         if (media != null) {
312             clearConsole();
313             cart.removeMedia(media);
314             foundToRemove = true;
315         } else {
316             System.out.println("****MEDIA NOT FOUND****");
317         }
318     }
319 }
320 case 4 -> {
321     boolean foundToPlay = false;
322     while (foundToPlay == false) {
323         System.out.println("Enter the title of the media (type 0 to stop): ");
324         String title = scanner.nextLine();
325         if (title.equals("0")) {
326             clearConsole();
327             break;
328         }
329         Media media = store.search(title);
330         if (media != null) {
331             if (media instanceof Disc || media instanceof CompactDisc) {
332                 media.play();
333             } else {
334                 System.out.println("This type of media is not supported!");
335             }
336             foundToPlay = true;
337         } else {
338             System.out.println("****MEDIA NOT FOUND****");
339         }
340     }
341 }

```

```

342         case 5 -> {
343             clearConsole();
344             cart.empty();
345         }
346         default -> {
347             clearConsole();
348             System.out.println("Invalid option, please choose again.");
349         }
350     }
351 }
352 }
353
354 public static void updateStoreMenu(Scanner scanner) {
355     boolean back = false;
356     while (!back) {
357         System.out.println("""
358             Options:
359             -----
360             1. Add a media
361             2. Remove a media
362             0. Back
363             -----
364             Please choose a number: 0-1-2
365             """);
366
367         int option = scanner.nextInt();
368         scanner.nextLine();
369         switch (option) {
370             case 0 -> {
371                 clearConsole();
372                 back = true;
373             }
374             case 1 -> {
375                 System.out.println("Enter the category of the media (1) Book, (2) CD, (3) DVD or (0) exit:");
376                 int categoryChoice = scanner.nextInt();
377                 scanner.nextLine();
378
379                 if (categoryChoice == 1) {
380                     System.out.println("Enter book title: ");
381                     String bookTitle = scanner.nextLine();
382                     System.out.println("Enter book category: ");
383                     String bookCategory = scanner.nextLine();
384                     System.out.println("Enter book cost: ");
385                     Float bookCost = scanner.nextFloat();
386                     scanner.nextLine();
387
388                     Book newBook = new Book(bookTitle, bookCategory, bookCost);
389                     store.addMedia(newBook);
390                 } else if (categoryChoice == 2) {
391                     System.out.println("Enter CD title: ");
392                     String cdTitle = scanner.nextLine();
393                     System.out.println("Enter CD category: ");
394                     String cdCategory = scanner.nextLine();
395                     System.out.println("Enter CD artist: ");
396                     String cdArtist = scanner.nextLine();
397                     System.out.println("Enter CD cost: ");
398                     Float cdCost = scanner.nextFloat();
399                     scanner.nextLine();
400
401                     CompactDisc newCD = new CompactDisc(cdTitle, cdCategory, cdCost, cdArtist);
402
403                     System.out.println("Do you want to add tracks to your CD? (1) Yes (0) No:");
404                     int addTrack = scanner.nextInt();
405                     scanner.nextLine();
406
407                     if (addTrack == 1) {
408                         System.out.println("How many tracks in your CD?");
409                         int numTrack = scanner.nextInt();
410                         scanner.nextLine();
411                         for (int i = 0; i < numTrack; i++) {
412                             System.out.println("Your " + (i+1) + " track: ");
413                             System.out.println("Enter track title: ");
414                             String trackTitle = scanner.nextLine();
415                             System.out.println("Enter track length: ");
416                             int trackLength = scanner.nextInt();
417                             scanner.nextLine();
418
419                             Track newTrack = new Track(trackTitle, trackLength);
420                             newCD.addTrack(newTrack);
421                         }
422                         store.addMedia(newCD);
423                     } else if (addTrack == 0) {
424                         store.addMedia(newCD);
425                     }
426                 } else if (categoryChoice == 3) {
427                     System.out.println("Enter DVD title: ");
428

```

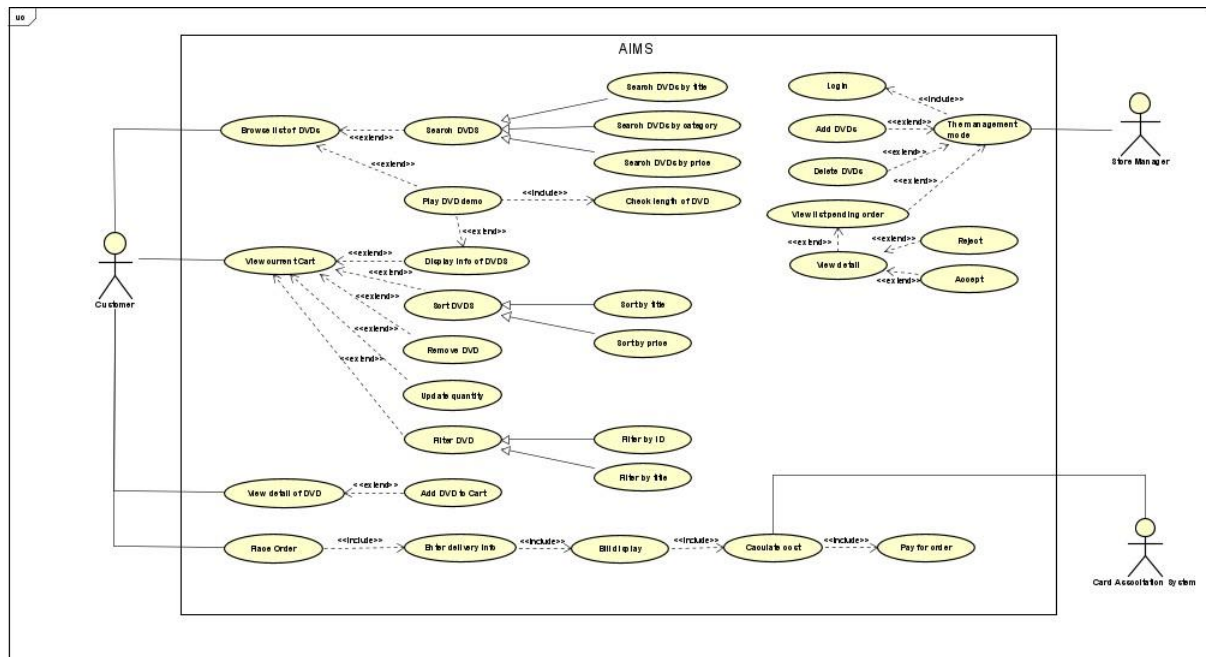
```

429     String dvdTitle = scanner.nextLine();
430     System.out.println("Enter DVD category: ");
431     String dvdCategory = scanner.nextLine();
432     System.out.println("Enter book cost: ");
433     Float dvdCost = scanner.nextFloat();
434     scanner.nextLine();
435
436     DigitalVideoDiscLHY newDVD = new DigitalVideoDiscLHY(dvdTitle, dvdCategory, dvdCost);
437     store.addMedia(newDVD);
438 } else if (categoryChoice == 0) {
439     clearConsole();
440     break;
441 } else {
442     System.out.println("Invalid option.");
443 }
444 }
445 case 2 -> {
446     boolean foundToRemove = false;
447     while (!foundToRemove) {
448         System.out.println("Enter the title of the media (type 0 to stop): ");
449         String titleForRemove = scanner.nextLine();
450         if (titleForRemove.equals("0")) {
451             clearConsole();
452             break;
453         }
454         Media media = store.search(titleForRemove);
455         if (media != null) {
456             System.out.println("Are you sure you want to remove this media? (yes/no): ");
457             String confirm = scanner.nextLine();
458             if (confirm.equalsIgnoreCase("yes")) {
459                 clearConsole();
460                 store.removeMedia(media);
461                 System.out.println("****MEDIA REMOVED****");
462                 foundToRemove = true;
463             } else {
464                 System.out.println("****MEDIA NOT REMOVED****");
465             }
466         } else {
467             System.out.println("****MEDIA NOT FOUND****");
468         }
469     }
470 }
471 default -> {
472     clearConsole();
473     System.out.println("Invalid option, please choose again.");
474 }
475 }
476 }
477 }
478 }

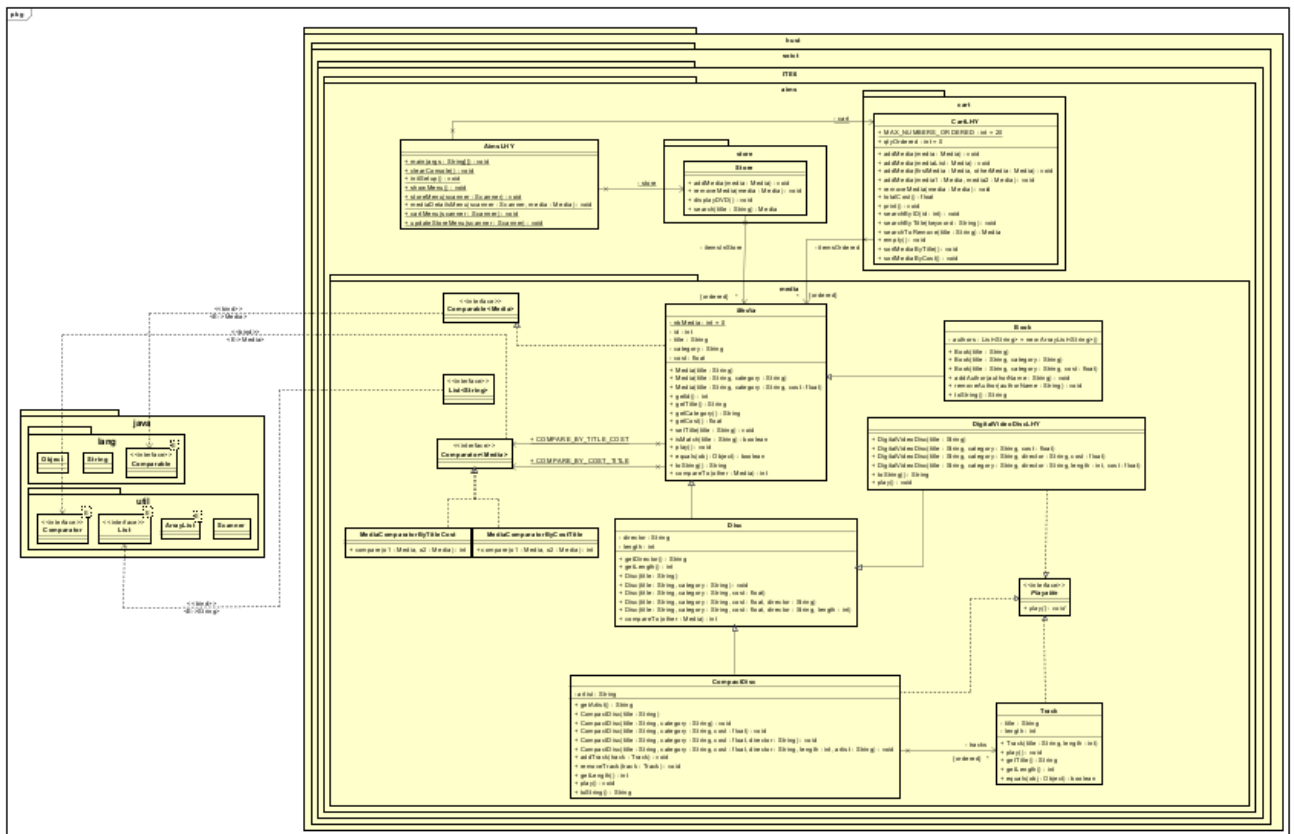
```

## 2 UML Diagram Final

### 2.1 Use-Case Diagram



### 2.2 Class Diagram



### 3 Answer Questions

- Which classes are aggregates of other classes? Checking all constructors of whole classes if they initialize for their parts?
  - Các lớp là tập hợp (aggregates) của các lớp khác:
    - + Lớp CompactDisc là tập hợp của lớp Track (chứa danh sách các đối tượng Track).
    - + Lớp Cart là tập hợp của lớp Media (chứa danh sách các đối tượng Media).
  - Kiểm tra các constructor:
    - + Trong lớp CompactDisc, constructor cần khởi tạo danh sách tracks (ArrayList<Track>). Nếu không, việc thêm track sẽ gây ra lỗi NullPointerException.
    - + Trong lớp Cart, constructor cần khởi tạo danh sách Media (ArrayList<Media>).
- Write constructors for parent and child classes. Remove redundant setter methods if any
  - Constructor cha: Trong lớp Media, các thuộc tính như title, category, và cost nên được khởi tạo thông qua constructor.
  - Constructor con: Trong lớp Disc hoặc Book, các thuộc tính đặc thù như author (trong Book) hoặc director, length (trong Disc) cần được khởi tạo trong constructor.
  - Loại bỏ setter thừa: Nếu các thuộc tính được khởi tạo thông qua constructor và không cần thay đổi sau đó, các phương thức setter tương ứng có thể bị loại bỏ để đảm bảo tính bất biến.
- 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?
  - Lớp cần implement Comparable: Lớp Media nên implement giao diện Comparable vì đây là lớp cha của tất cả các loại phương tiện (CD, DVD, Book), và việc so sánh cần được áp dụng chung cho mọi loại.

```
public abstract class Media implements Comparable<Media>
```

- In those classes, how should you implement the compareTo() method to reflect the ordering that we want?

- Trong lớp Media, phương thức compareTo() nên sắp xếp theo: Tiêu đề (title) theo thứ tự tăng dần. Nếu tiêu đề giống nhau, so sánh giá (cost) theo thứ tự giảm dần.

```
public int compareTo(Media other) {
    int titleComparison = this.getTitle().compareTo(other.getTitle());
    if (titleComparison != 0) {
        return titleComparison;
    } else {
        return Float.compare(other.getCost(), this.getCost());
    }
}
```

- 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?

- Không, chúng ta không thể có hai quy tắc sắp xếp trong cùng một lớp nếu sử dụng giao diện Comparable. Comparable chỉ cho phép định nghĩa một thứ tự tự nhiên duy nhất.

- 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?

- Trong lớp Disc (hoặc cụ thể hơn là DigitalVideoDisc), chúng ta có thể ghi đè phương thức compareTo() để phản ánh quy tắc sắp xếp mới.
- Quy tắc mới: Sắp xếp theo tiêu đề (title) tăng dần. Nếu tiêu đề giống nhau, sắp xếp theo độ dài (length) giảm dần. Nếu độ dài giống nhau, sắp xếp theo giá (cost) tăng dần.

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



