BÁO CÁO OOP LAB 04

Họ tên: Bùi Ý Nhi – 20225657

1. **Answer Question:**

1.1. Which classes are aggregates of other classes(9)?

- Store aggregates Media (like DigitalVdDisc, CompactDisc, etc.)

  Cart aggregates Media (like DigitalVdDisc, CompactDisc, etc.)

  CompactDisc aggregates Track

  Book aggregates authors (in the form of a list of strings)

1.2. If the passing object is not an instance of Media, what happens(10)?

- If the method is defined to accept a parameter of type Media (e.g., addMedia(Media media)), then only instances of Media or its subclasses can be passed. If you attempt to pass an object that is not a Media instance or a subclass of Media, a compilation error will occur. This is because Java enforces type safety at compile time.

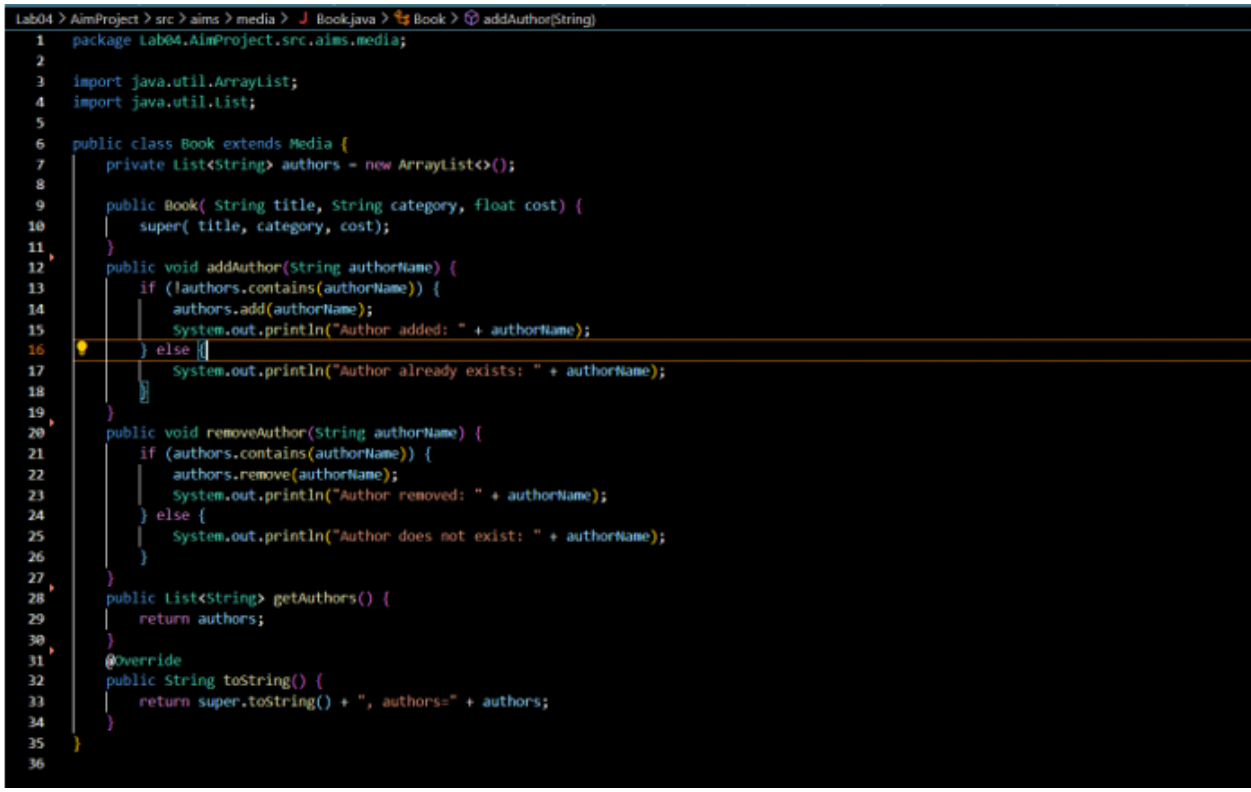
1.3. **(11)**

- What class should implement the Comparable interface?
  - ⇨ Media (or possibly subclasses of Media, such as DigitalVdDisc, CompactDisc, Book, etc.):
- In those classes, how should you implement the compareTo()method be to reflect the ordering that we want? (exg
  - ⇨ First compare by title: We use String's compareTo() method to compare the titles. If the titles are different, it will return a value indicating the order (< 0 if this.title is lexicographically less than other.title, > 0 if it is greater).
  - ⇨ If the titles are the same, it compares by cost using Float.compare(), which ensures the correct comparison of floating-point numbers.
- 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?
  - ⇨ Yes, we can have two ordering rules, but the Comparable interface only defines one natural ordering for the class.
- 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?

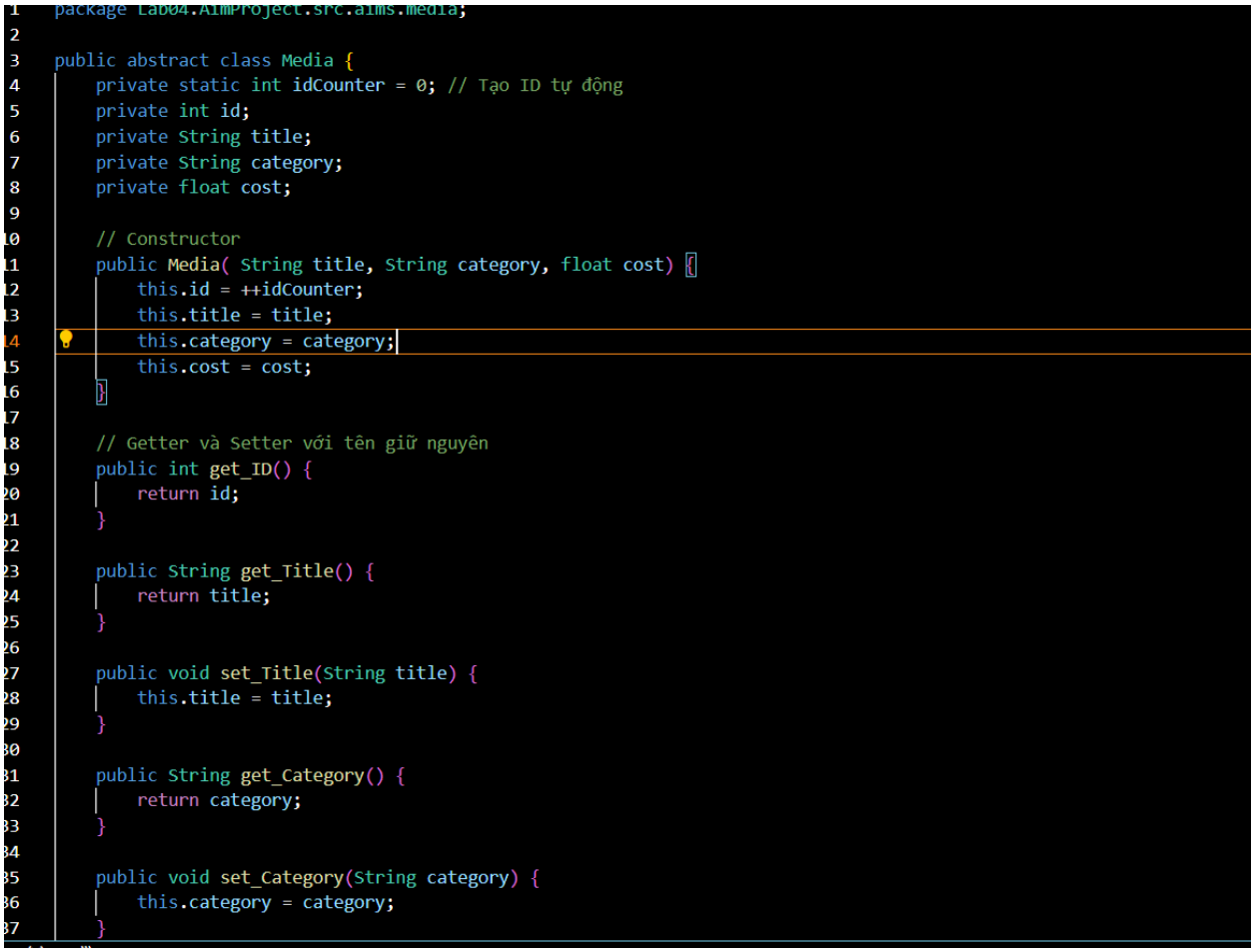
- ⇨ If DVDs need to have a specific ordering rule (by title, then decreasing length, and then cost), we can override the compareTo() method in the DigitalVdDisc class (or any other class where this behavior is required).
- ⇨ We can implement multiple ordering rules by defining different comparators for different sorting criteria (e.g., by title then cost, or by cost then title).
- ⇨ For DigitalVdDisc, we can override the compareTo() method to implement a custom sorting order, like title, then decreasing length, then cost.
- ⇨ Using Comparable or Comparator, we can achieve flexible sorting for our media objects based on different attributes.

2. Source code:

- *Book.java*



- *Media.java*



```java
package Lab04.AimProject.src.aims.media;

public abstract class Media {
    private static int idCounter = 0; // Tạo ID tự động
    private int id;
    private String title;
    private String category;
    private float cost;

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

    // Getter và Setter với tên giữ nguyên
    public int get_ID() {
        return id;
    }

    public String get_Title() {
        return title;
    }

    public void set_Title(String title) {
        this.title = title;
    }

    public String get_Category() {
        return category;
    }

    public void set_Category(String category) {
        this.category = category;
    }
```

```java
    public float get_Cost() {
        return cost;
    }

    public void set_Cost(float cost) {
        this.cost = cost;
    }

    // Override equals() để so sánh các Media theo title
    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;
        Media media = (Media) obj;
        return title != null && title.equals(media.title);
    }

    // Override toString() để hiển thị thông tin cơ bản của Media
    @Override
    public String toString() {
        return "Media{id=" + id +
                ", title='" + title + '\'' +
                ", category='" + category + '\'' +
                ", cost=" + cost + '}';
    }
}
```

- *CompactDisc.java*

```java
package Lab04.AimProject.src.aims.media;

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

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

    // Constructor
    public CompactDisc(String title, String category, float cost, String artist) {
        super(title, category, cost);
        this.artist = artist;
    }

    // Getter
    public String getArtist() {
        return artist;
    }

    // Thêm bản nhạc vào CD
    public void addTrack(Track track) {
        if (!tracks.contains(track)) {
            tracks.add(track);
            System.out.println("Track added: " + track.getTitle());
        } else {
            System.out.println("Track already exists: " + track.getTitle());
        }
    }
```

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

    // Xóa bản nhạc khỏi CD
    public void removeTrack(Track track) {
        if (tracks.contains(track)) {
            tracks.remove(track);
            System.out.println("Track removed: " + track.getTitle());
        } else {
            System.out.println("Track does not exist: " + track.getTitle());
        }
    }

    // Tính tổng độ dài của tất cả các bản nhạc
    public int getLength() {
        int totalLength = 0;
        for (Track track : tracks) {
            totalLength += track.getLength();
        }
        return totalLength;
    }
```

```java
    // Triển khai phương thức play()
    @Override
    public void play() {
        if (tracks.isEmpty()) {
            System.out.println(x:"No tracks to play.");
        } else {
            System.out.println("Playing CD: " + get_Title());
            System.out.println("Artist: " + artist);
            for (Track track : tracks) {
                track.play(); // Gọi play() của từng Track
            }
        }
    }

    // Override toString() để hiển thị thông tin
    @Override
    public String toString() {
        StringBuilder trackInfo = new StringBuilder();
        for (Track track : tracks) {
            trackInfo.append(track.toString()).append(str:"\n");
        }
        return super.toString() +", artist='" + artist + '\'' +", totalLength=" + getLength() +"\nTracks:\n" + trackInfo;
    }
}
```

- *Disc.java*

```java
package Lab04.AimProject.src.aims.media;


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

    public Disc(int id, String title, String category, float cost, int length, String director) {
            super( title, category, cost);  // Gọi constructor của lớp Media
            this.length = length;
            this.director = director;
        }
    public int getLength() {
        return length;
    }
    public String getDirector() {
        return director;
    }

}
```

- *Track.java*

```java
package Lab04.AimProject.src.aims.media;

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

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

    // Getter
    public String getTitle() {
        return title;
    }

    public int getLength() {
        return length;
    }
    // Override equals() để so sánh Track theo title và length
    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;
        Track track = (Track) obj;
        return length == track.length && title.equals(track.title);
    }
```

```java
    // Triển khai play() từ Playable
    @Override
    public void play() {
        if (length > 0) {
            System.out.println("Playing Track: " + title);
            System.out.println("Track length: " + length);
        } else {
            System.out.println(x:"Cannot play track. Invalid length.");
        }
    }

    // Override toString() để hiển thị thông tin
    @Override
    public String toString() {
        return "Track{title='" + title + "', length=" + length + "}";
    }
}
```

- *Interface playable()*
- *updateCart*

```java
    // Sắp xếp giỏ hàng theo tiêu chí (title hoặc cost)
    public void sortCart(String sortBy) {
        if (sortBy.equalsIgnoreCase(anotherString:"title")) {
            Collections.sort(itemOrdered, MediaComparator.COMPARE_BY_TITLE);
        } else if (sortBy.equalsIgnoreCase(anotherString:"cost")) {
            Collections.sort(itemOrdered, MediaComparator.COMPARE_BY_COST);
        } else {
            System.out.println(x:"Invalid sorting option.");
        }
        print();  // In giỏ hàng sau khi sắp xếp
    }

    // Lọc giỏ hàng theo tiêu chí (ID hoặc title)
    public void filterCart(String filterBy, String value) {
        if (filterBy.equalsIgnoreCase(anotherString:"id")) {
            int id = Integer.parseInt(value);
            searchById(id);
        } else if (filterBy.equalsIgnoreCase(anotherString:"title")) {
            searchByTitle(value);
        } else {
            System.out.println(x:"Invalid filter option.");
        }
    }
}
```

- *updateStore*

```java
    // Xóa phương tiện khỏi cửa hàng
    public void removeMedia(Media media) {
        if (checkMedia(media)) {
            itemsInStore.remove(media);  // Xóa phương tiện khỏi danh sách
            System.out.println(x:"The media has been deleted.");
        } else {
            System.out.println(x:"Not found.");
        }
    }

    // Thêm phương tiện vào cửa hàng
    public void addMedia(Media media) {
        if (!checkMedia(media)) {  // Nếu phương tiện chưa có trong cửa hàng
            itemsInStore.add(media);  // Thêm phương tiện vào cửa hàng
            System.out.println(x:"The media has been added.");
        } else {
            System.out.println(x:"Media already exists in the store.");
        }
    }

    // In ra tất cả các phương tiện trong cửa hàng
    public void print() {
        StringBuilder output = new StringBuilder();
        for (Media media : itemsInStore) {
            output.append(media.get_Title()).append(str:" - ").append(media.get_Category()).append(str:" - ").append(media.get_Cost()).append(str:"
        }
        System.out.println(output.toString());
    }
}
```

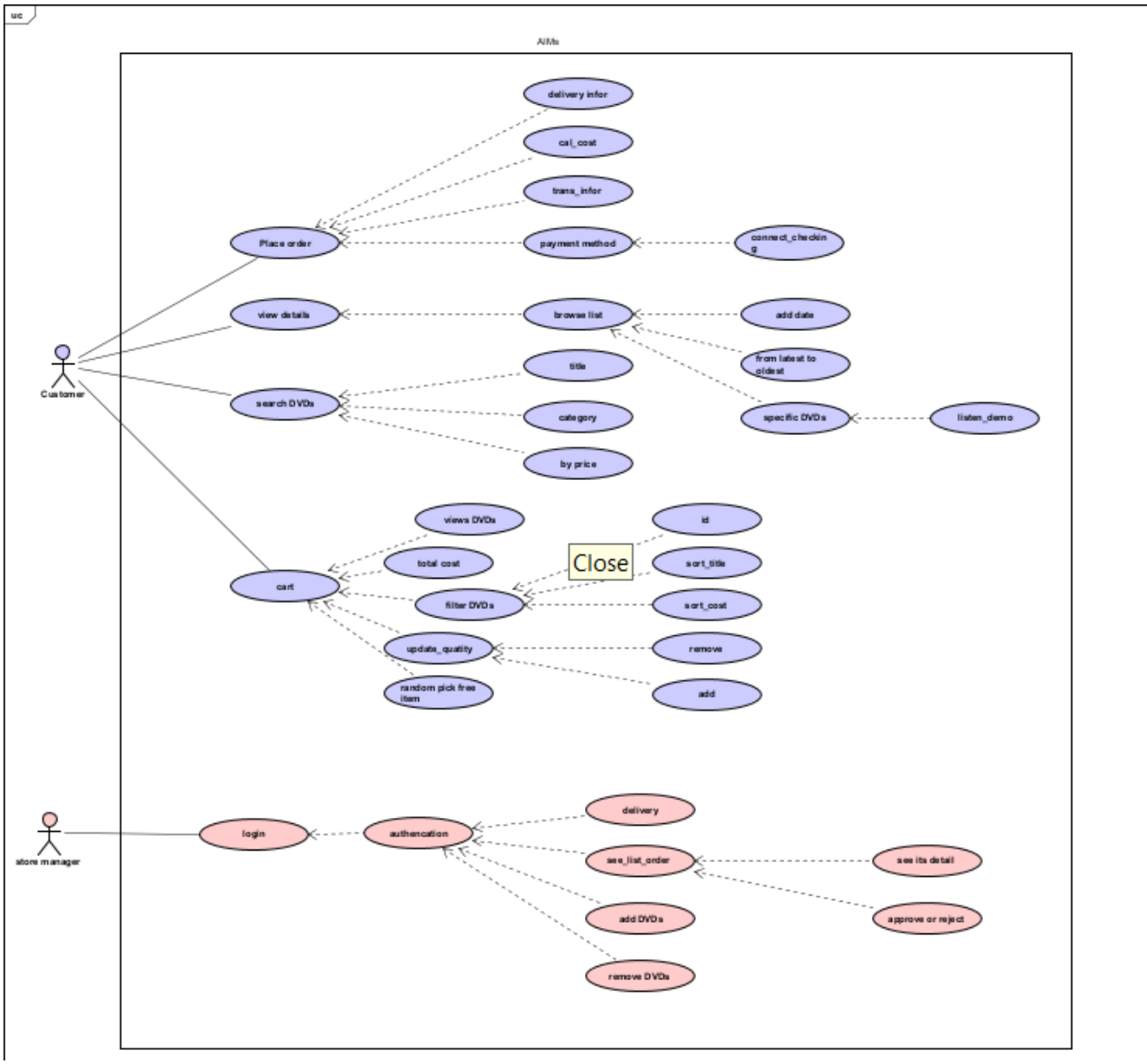
- *MediaCOmparator.java(cost & title)*

```java
package Lab04.AimProject.src.aims.media;
import java.util.Comparator;


public class MediaComparator {
    // Comparator sắp xếp theo tên (title)
    public static Comparator<Media> COMPARE_BY_TITLE = new Comparator<Media>() {
        @Override
        public int compare(Media m1, Media m2) {
            return m1.get_Title().compareTo(m2.get_Title());
        }
    };

    // Comparator sắp xếp theo giá (cost)
    public static Comparator<Media> COMPARE_BY_COST = new Comparator<Media>() {
        @Override
        public int compare(Media m1, Media m2) {
            return Float.compare(m1.get_Cost(), m2.get_Cost());
        }
    };
}

```

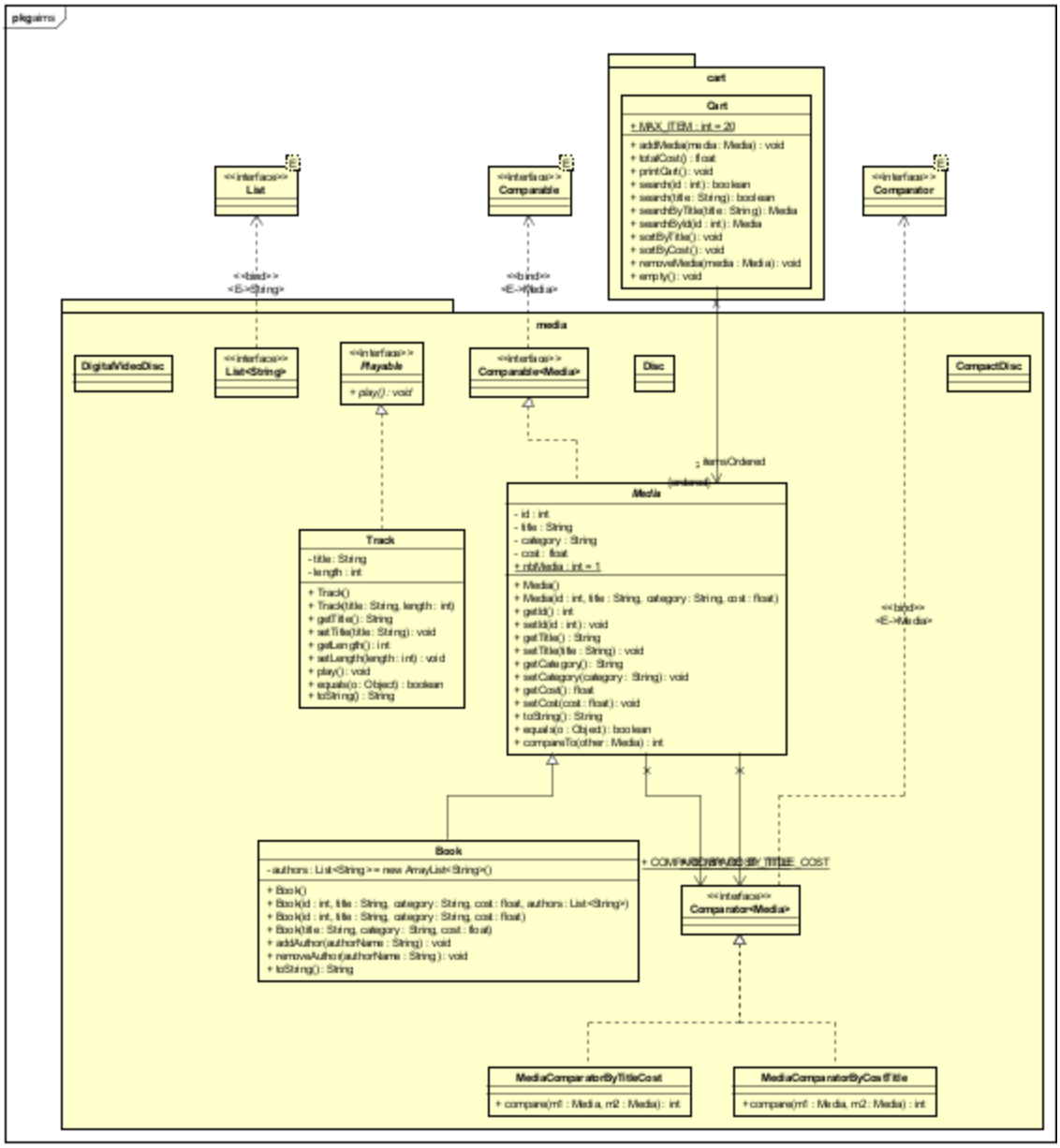
-

3. Usecase Diagram and Class Diagram

- **Usecase Diagram**

- 

- **Class Diagram**


pkgairro
cart
Cart
+ MAX_ITEM : int = 20
+ addMedia(media : Media) : void
+ totalCost() : float
+ printCart() : void
+ search(id : int) : boolean
+ search(title : String) : boolean
+ searchByTitle(title : String) : Media
+ searchById(id : int) : Media
+ sortByTitle() : void
+ sortByCost() : void
+ removeMedia(media : Media) : void
+ empty() : void
«interface» List
«interface» Comparable
«interface» Comparator
«bind» <E->String>
«bind» <E->Media>
media
DigitalVideoDisc
«interface» List<String>
«interface» Playable
+ play() : void
«interface» Comparable<Media>
Disc
CompactDisc
; itemsOrdered
{ordered}
Track
- title : String
- length : int
+ Track()
+ Track(title : String, length : int)
+ getTitle() : String
+ setTitle(title : String) : void
+ getLength() : int
+ setLength(length : int) : void
+ play() : void
+ equals(o : Object) : boolean
+ toString() : String
Media
- id : int
- title : String
- category : String
- cost : float
+ nbMedia : int = 1
+ Media()
+ Media(id : int, title : String, category : String, cost : float)
+ getId() : int
+ setId(id : int) : void
+ getTitle() : String
+ setTitle(title : String) : void
+ getCategory() : String
+ setCategory(category : String) : void
+ getCost() : float
+ setCost(cost : float) : void
+ toString() : String
+ equals(o : Object) : boolean
+ compareTo(other : Media) : int
«bind» <E->Media>
Book
- authors : List<String> = new ArrayList<String>()
+ Book()
+ Book(id : int, title : String, category : String, cost : float, authors : List<String>)
+ Book(id : int, title : String, category : String, cost : float)
+ Book(title : String, category : String, cost : float)
+ addAuthor(authorName : String) : void
+ removeAuthor(authorName : String) : void
+ toString() : String
+ COMPARE_BY_TITLE_COST
«interface» Comparator<Media>
MediaComparatorByTitleCost
+ compare(m1 : Media, m2 : Media) : int
MediaComparatorByCostTitle
+ compare(m1 : Media, m2 : Media) : int