**Báo cáo thực hành Lập trình hướng đối tượng Lab 4**

**Họ và tên: Dương Thái Anh**

**MSSV: 20226099**

**Lớp: 744520**

1. **Tạo Book class**

A computer screen shot of text

Description automatically generated

A screen shot of a computer code

Description automatically generated

**II. Tạo Media class**

1. **Media class**

A screen shot of a computer program

Description automatically generated

**A screenshot of a computer program

Description automatically generated**

**A screen shot of a computer code

Description automatically generated**

1. **Sửa Book class**

A screen shot of a computer program

Description automatically generated

**A computer screen with text and images

Description automatically generated**

**III. Create CompactDisc class**

1. **Create Disc class extends Media class**

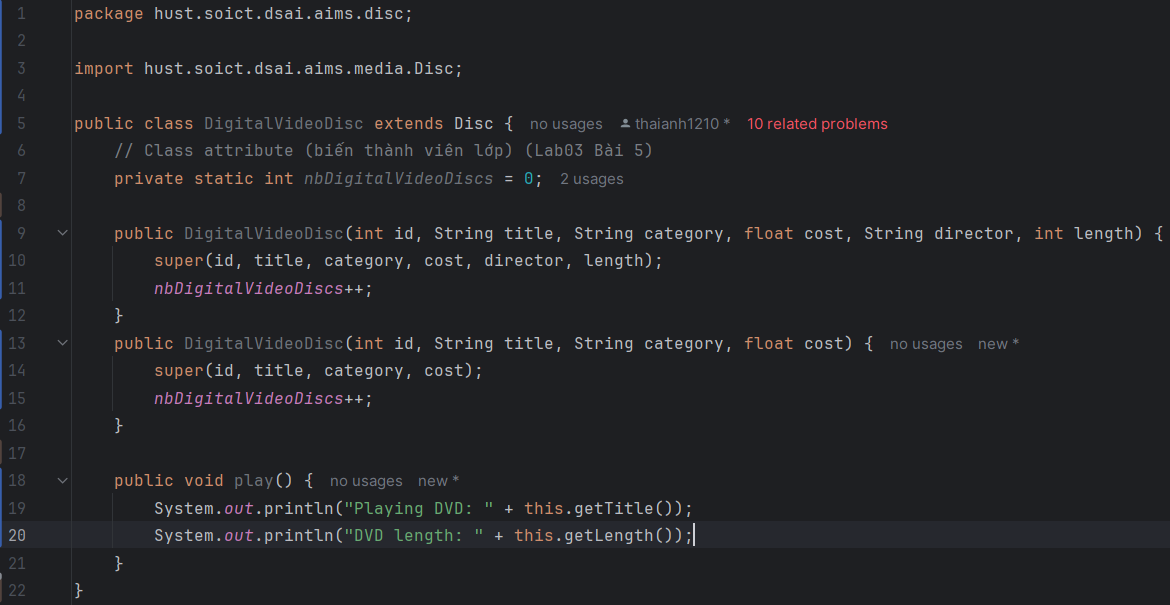
**A screen shot of a computer

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

1. **Sửa DigitalVideoDisc class**



1. **Create Track class**

A computer screen shot of a program

Description automatically generated

1. **Create CompactDisc class extends Disc class**

A screen shot of a computer program

Description automatically generated

A computer screen shot of a program code

Description automatically generated

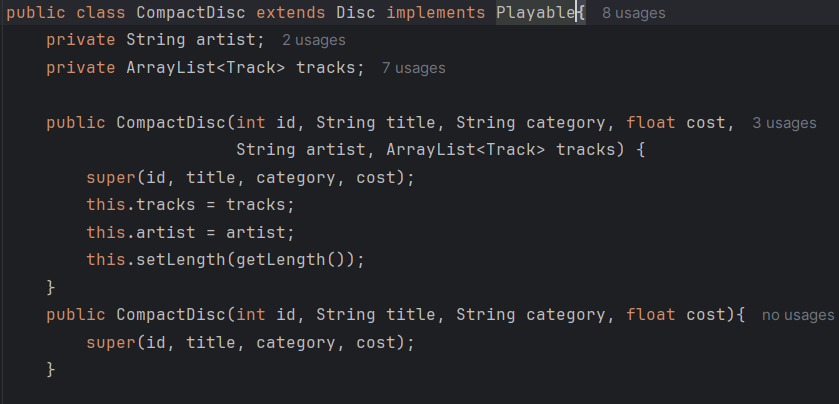
**IV. Create interface Playable**

1. **Interface Playable**

A screen shot of a computer

Description automatically generated

1. **Fix CompactDisc**



A screen shot of a computer code

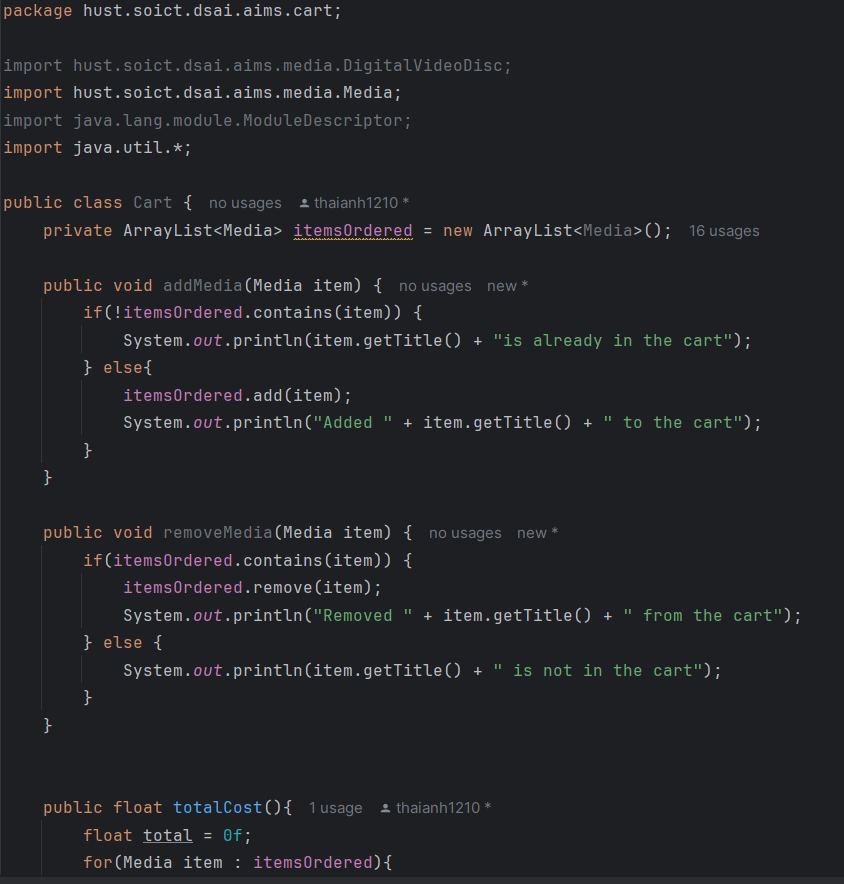
Description automatically generated

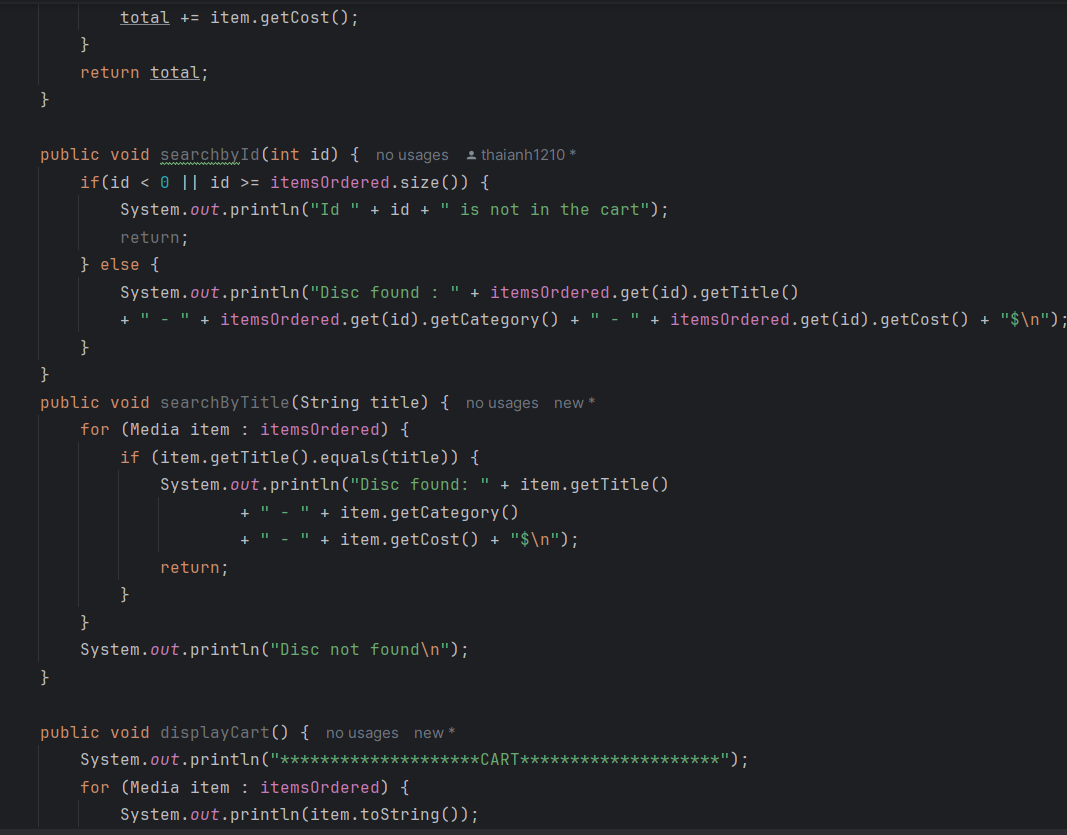
1. **Fix DigitalVideoDisc**

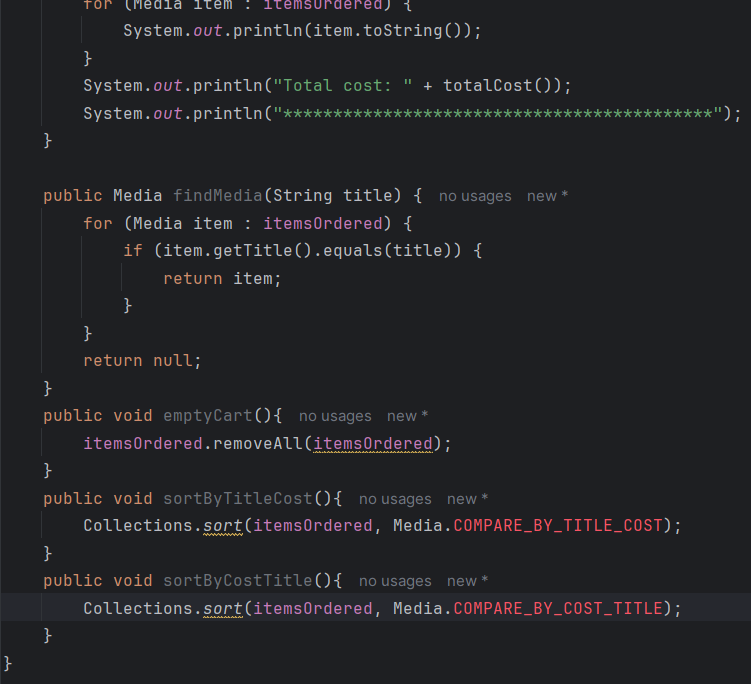
A screen shot of a computer program

Description automatically generated

**V. Update Cart class to work with Media**







**VI. Update Store class to work with Media**

****

**A computer screen shot of text

Description automatically generated**

**A screen shot of a computer code

Description automatically generated**

**VII. Unique item in a list**

**A screen shot of a computer code

Description automatically generated**

1. **Sửa lại Media**

A computer screen shot of a program code

Description automatically generated

A screen shot of a computer program

Description automatically generated

A computer screen shot of a program code

Description automatically generated

**VIII. Polymorphism with toString method;**

**A computer screen shot of a program code

Description automatically generated**

**A black background with white text

Description automatically generated**

**IX. Sort media in the cart**

**A computer screen shot of a program code

Description automatically generated** **A computer screen with white and orange text

Description automatically generated**

**X. Create a complete console application in the Aims class**

**A computer screen shot of a program code

Description automatically generated**

**A computer screen shot of a program code

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**A screen shot of a computer code

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer screen

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

****

**A computer screen shot of a program code

Description automatically generated**

**XI. Trả lời câu hỏi**

Question 1: Which classes are aggregates of other classes? Checking all constructors of whole classes if they initialize for their parts? - Write constructors for parent and child classes. Remove redundant setter methods if any

Trả lời: Lớp **Book** và **CompactDisc** là các lớp con của lớp **Media**, trong đó:

* **Book** chứa danh sách các tác giả (authors).
* **CompactDisc** chứa danh sách các bài hát (tracks).

Cả hai lớp con này đều là tập hợp các lớp khác vì chúng chứa các đối tượng (danh sách) trong đó. Các constructor của chúng đều khởi tạo các giá trị cần thiết và không cần setter cho các danh sách đó, vì các phần tử trong danh sách chỉ được thay đổi qua các phương thức như thêm và xóa.

Ví dụ:  
**Lớp Cha Media**

public abstract class **Media** {

private int id;

private String title;

private String category;

private float cost;

// Constructor

public Media(int id, String title, String category, float cost) {

this.id = id;

this.title = title;

this.category = category;

this.cost = cost;

}

// Getters

public int getId() {

return id;

}

public String getTitle() {

return title;

} public String getCategory() { return category; }

public float getCost() { return cost; }

}

**Lớp con Book**

public class Book extends Media {

private ArrayList<String> authors;

// Constructor

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

super(id, title, category, cost);

this.authors = new ArrayList<>();

} // Thêm tác giả

public void addAuthor(String authorName) {

if (!authors.contains(authorName)) {

authors.add(authorName);

} else {

System.out.println("Author already exists."); } }

// Xóa tác giả

public void removeAuthor(String authorName) {

if (authors.contains(authorName)) {

authors.remove(authorName);

} else { System.out.println("Author not found."); } }

// Getter cho authors

public ArrayList<String> getAuthors() {

return authors; } }

**Lớp con CompactDisc**

public class CompactDisc extends Media {

private ArrayList<String> tracks;

// Constructor

public CompactDisc(int id, String title, String category, float cost) {

super(id, title, category, cost);

this.tracks = new ArrayList<>(); }

// Thêm track public void addTrack(String track) {

if (!tracks.contains(track)) {

tracks.add(track);

} else { System.out.println("Track already exists."); } }

// Xóa track

public void removeTrack(String track) {

if (tracks.contains(track)) { tracks.remove(track);

} else { System.out.println("Track not found."); } }

// Getter cho tracks

public ArrayList<String> getTracks() {

return tracks; } }

Question 2: 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?

Lớp **Media** nên implement **Comparable<Media>** vì tất cả các lớp kế thừa từ **Media** như **Book**, **CompactDisc**, và **DigitalVideoDisc** cần có khả năng so sánh với nhau. Việc này cho phép các đối tượng thuộc các lớp con được so sánh theo một tiêu chí chung mặc định.

+ @Override public int compareTo(Media other) {

// So sánh theo title

int titleComparison = this.title.compareTo(other.title);

if (titleComparison != 0) {

return titleComparison;

// Trả về kết quả so sánh title }

// Nếu title giống nhau, so sánh theo cost

return Float.compare(this.cost, other.cost); }

Không, nếu chỉ dùng Comparable, chỉ có thể áp dụng một quy tắc sắp xếp mặc định. Nếu muốn hỗ trợ nhiều quy tắc sắp xếp (ví dụ: title → cost và cost → title), nên sử dụng Comparator để có tính linh hoạt hơn. + Cần override lại phương thức compareTo() trong lớp con DigitalVideoDisc

package hust.soict.dsai.aims.media;

public class DigitalVideoDisc extends Media implements Comparable {

private int length;

public DigitalVideoDisc(int id, String title, String category, float cost, int length){

super(id, title, category, cost);

this.length = length; }

public int getLength() {

return length; }

@Override

public int compareTo(Media other) {

if (other instanceof DigitalVideoDisc) {

DigitalVideoDisc otherDVD = (DigitalVideoDisc) other;

// So sánh theo title

int titleComparison = this.getTitle().compareTo(otherDVD.getTitle());

if (titleComparison != 0) {

return titleComparison; }

// Nếu title giống nhau, so sánh length (giảm dần)

int lengthComparison = Integer.compare(otherDVD.getLength(), this.getLength());

if (lengthComparison != 0) {

return lengthComparison; }

// Nếu length giống nhau, so sánh cost

return Float.compare(this.getCost(), otherDVD.getCost()); }

// Nếu không phải DVD, sử dụng quy tắc của Media

return super.compareTo(other); } }

**XII. Update class diagram**

