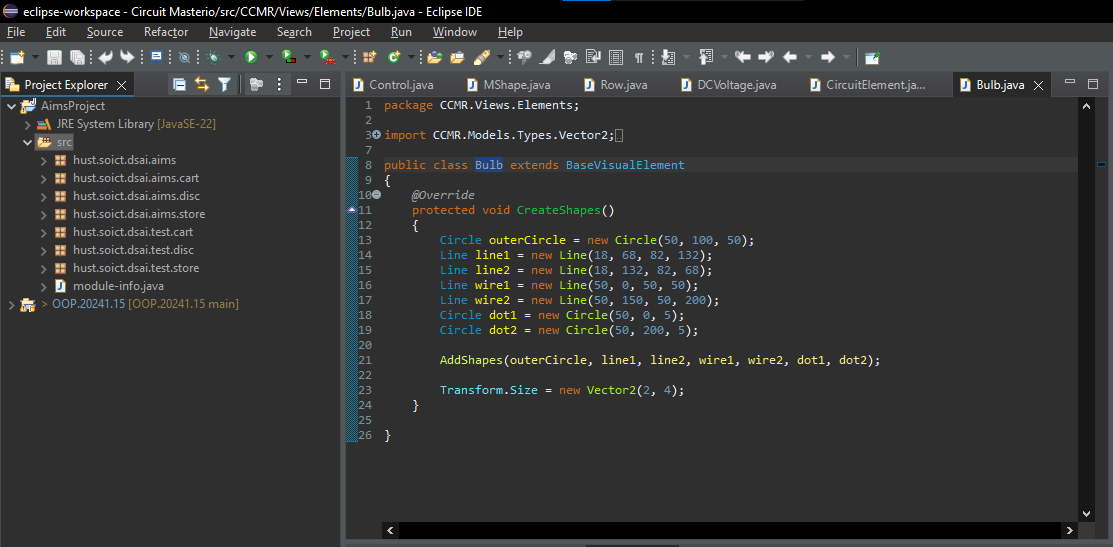
Báo cáo thực hành lab4

1. Import the existing project into the workspace of Eclipse

* Kết quả sau khi import 1 project:
* 

1. Additional requirements of AIMS
2. Creating the Book class

* Đoạn mã của class Book:

public class Book {

private int id;

private String title;

private String category;

private float cost;

private ArrayList<String> authors = new ArrayList<String>();

public Book() {

}

public int getId() {

return id;

}

public String getTitle() {

return title;

}

public String getCategory() {

return category;

}

public float getCost() {

return cost;

}

public ArrayList<String> getAuthors() {

return authors;

}

public void addAuthor(String author) {

if (author != null && !author.isEmpty()) {

authors.add(author);

}

}

public void removeAuthor(String author) {

if (author != null && !author.isEmpty() && authors.contains(author)) {

authors.remove(author);

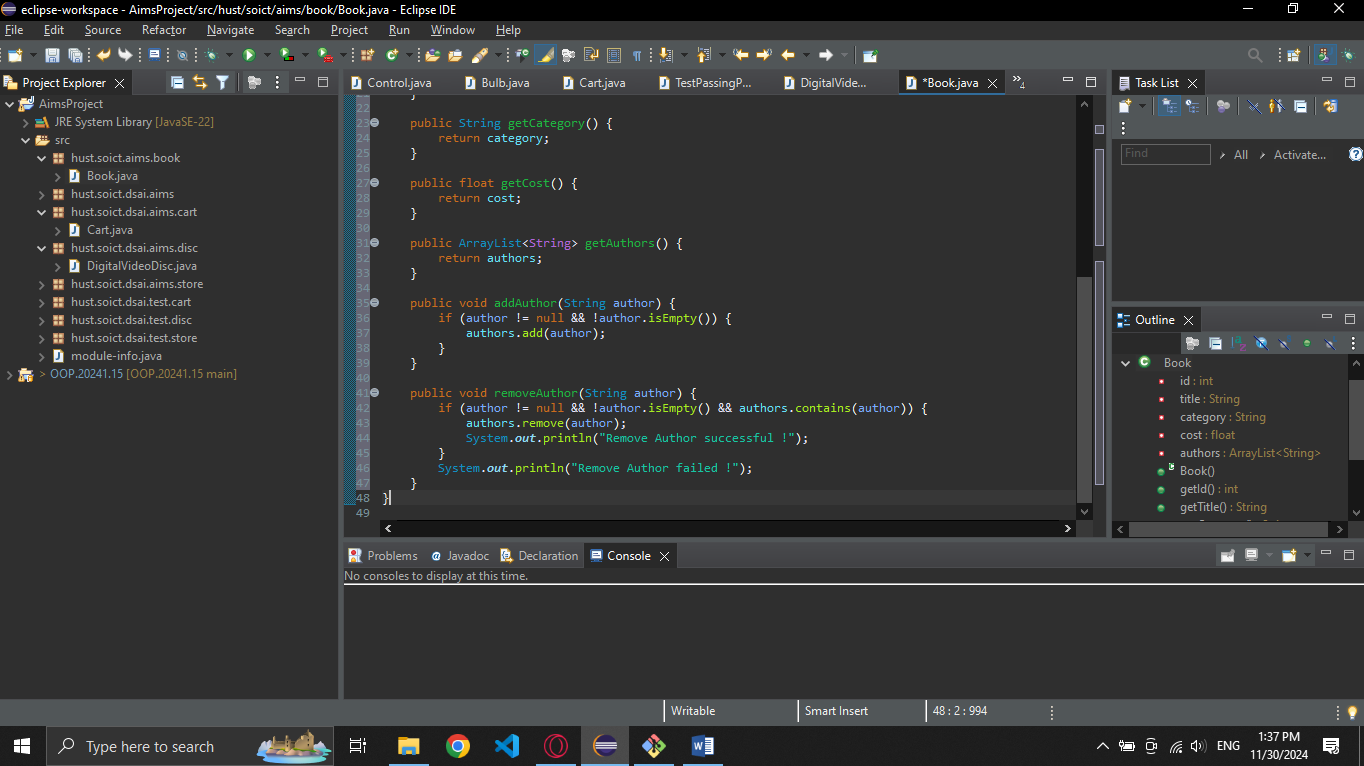
System.***out***.println("Remove Author successful !");

}

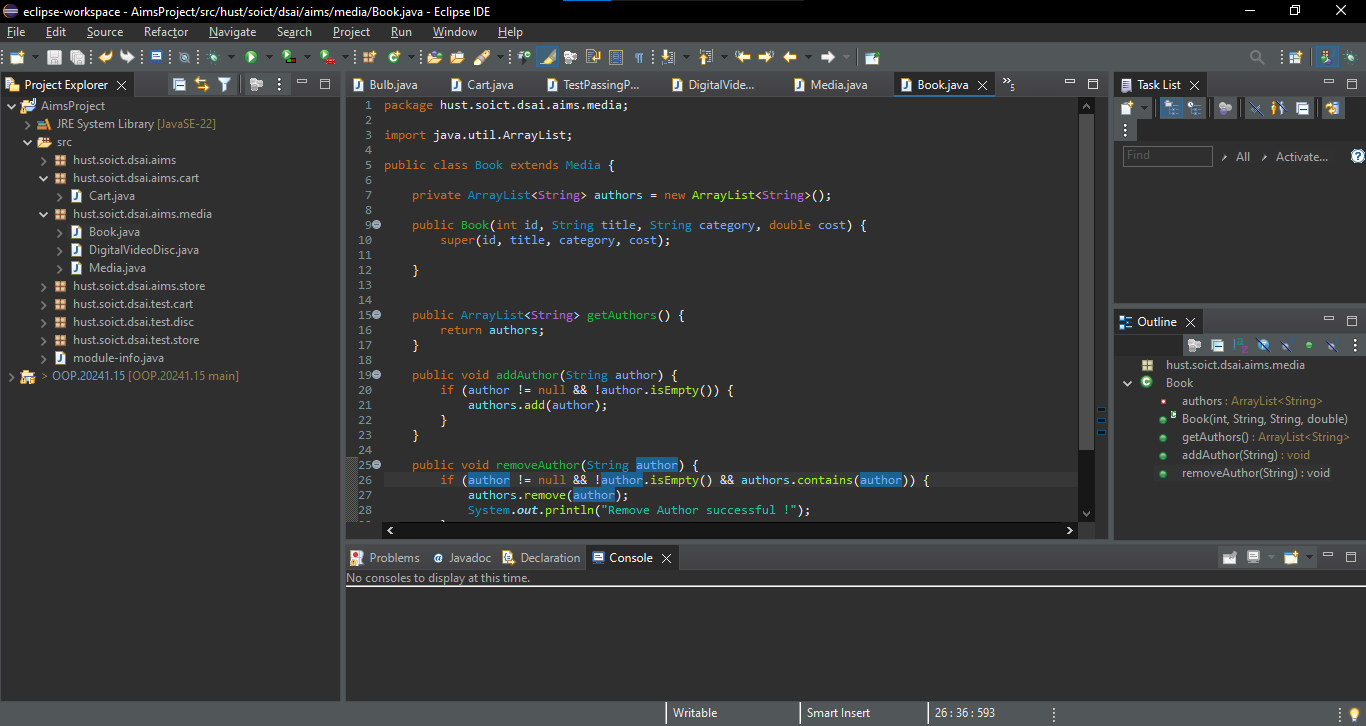
System.***out***.println("Remove Author failed !");

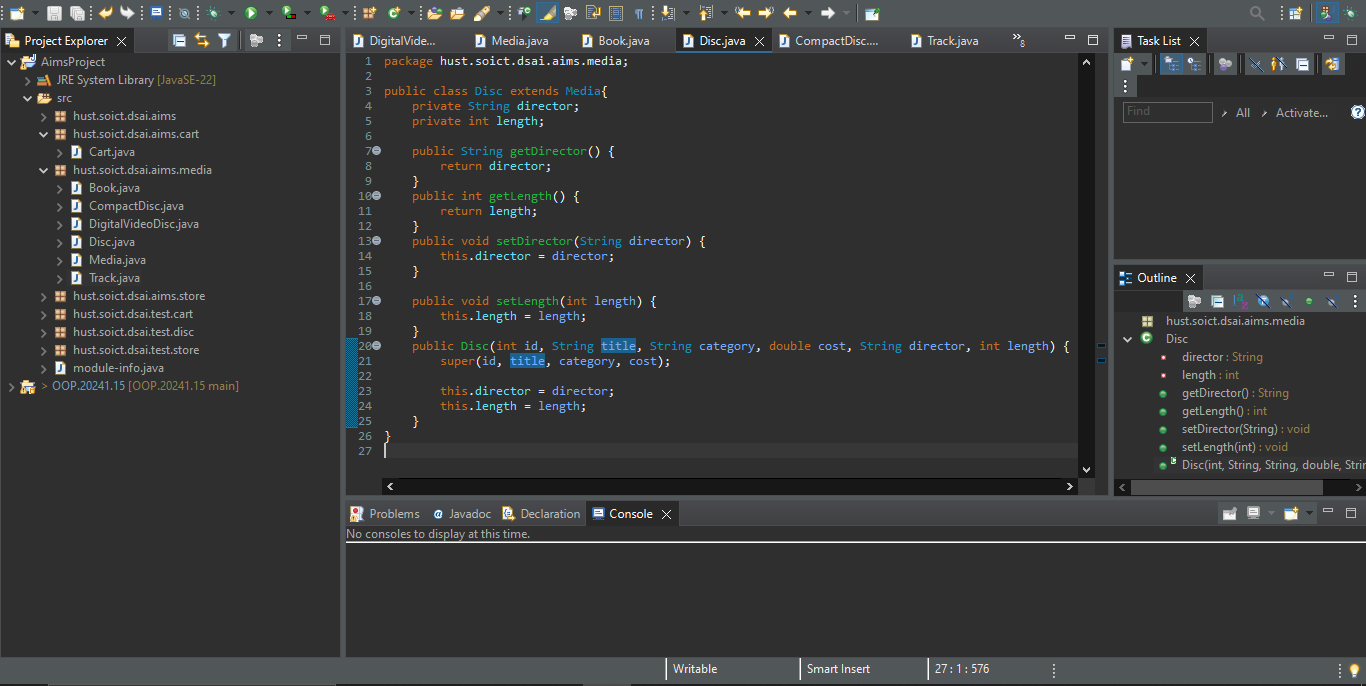
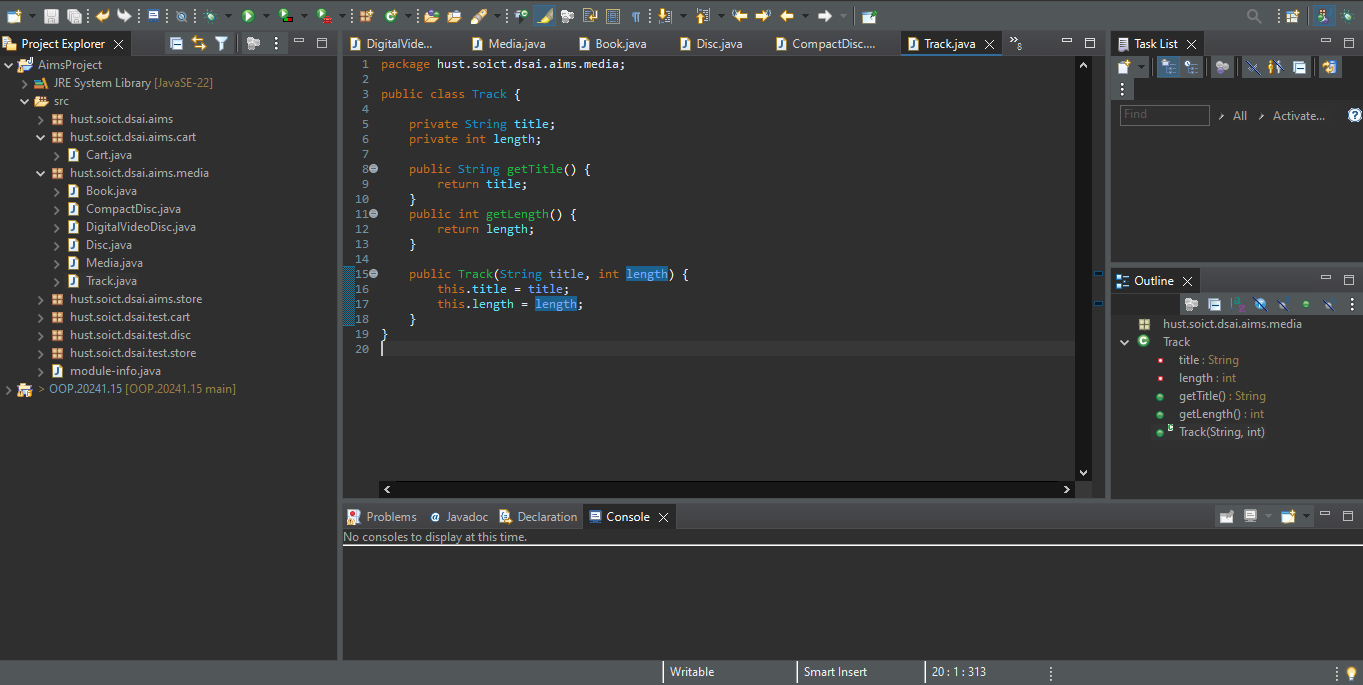
}

}



1. Creating the abstract Media class

* Thay đổi đoạn mã và di chuyển 2 lớp Book và DigitalVideoDisc vào bên trong package hust.soict.dsai.aims.media.
* 

1. Creating the CompactDisc class
   1. Create the Disc class extending the Media class
   * Đoạn mã của class Disc
   * 
   1. Create the Track class which models a track on a compact disc and will store information incuding the title and length of the track.
   * Đoạn mã của lớp Track
   * 
   1. Open the CompactDisc class
   * Đoạn mã tương ứng theo yêu cầu:

public class CompactDisc extends Disc {

private String artist;

private ArrayList<Track> tracks = new ArrayList<Track>();

public String getArtist() {

return artist;

}

public ArrayList<Track> getTracks() {

return tracks;

}

public CompactDisc(int id, String title, String category, double cost, String director,

int length, String artist, ArrayList<Track> tracks) {

super(id, title, category, cost, director, length);

this.artist = artist;

this.tracks = tracks;

}

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());

}

}

public void removeTrack(Track track) {

if (tracks.contains(track)) {

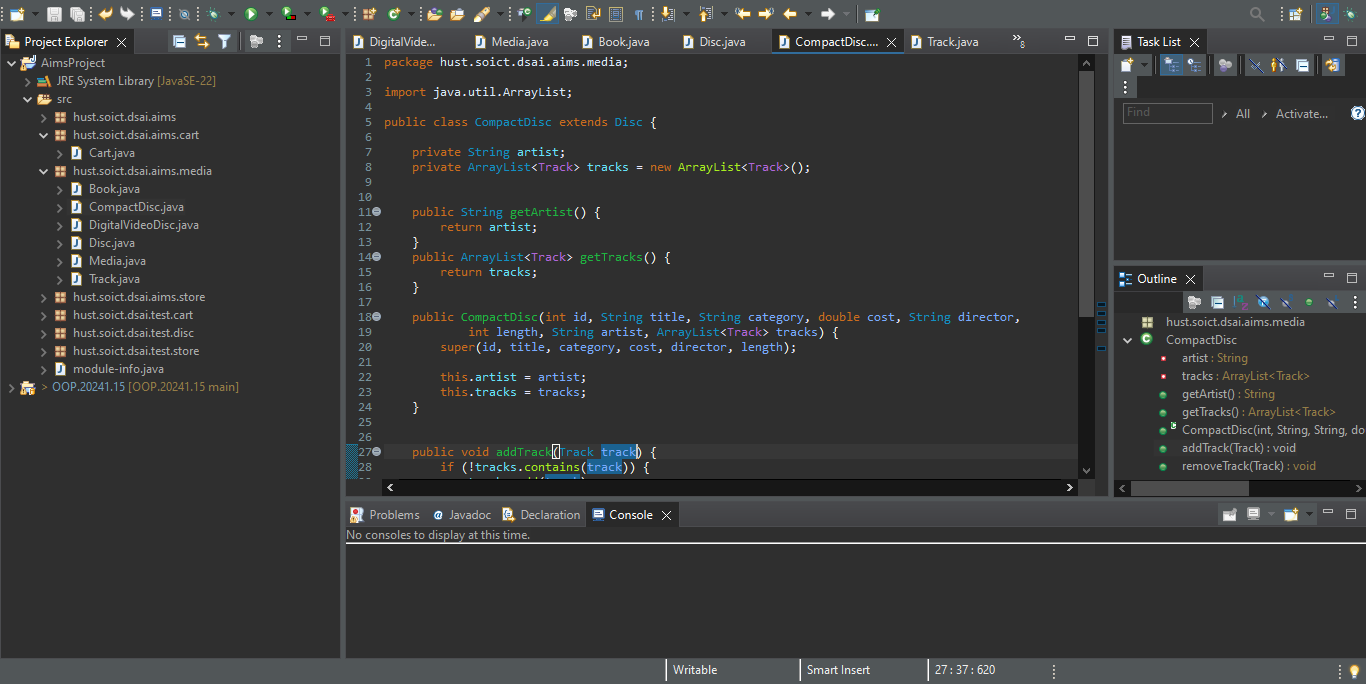
tracks.remove(track);

System.***out***.println("Track removed: " + track.getTitle());

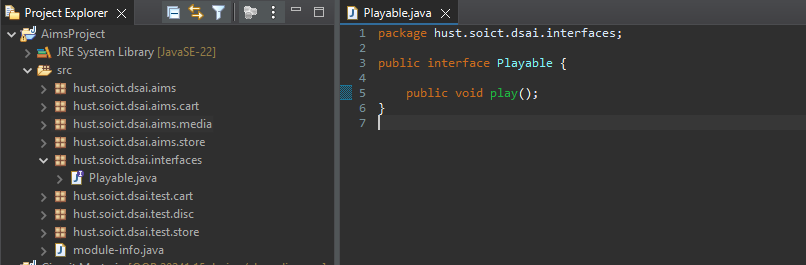
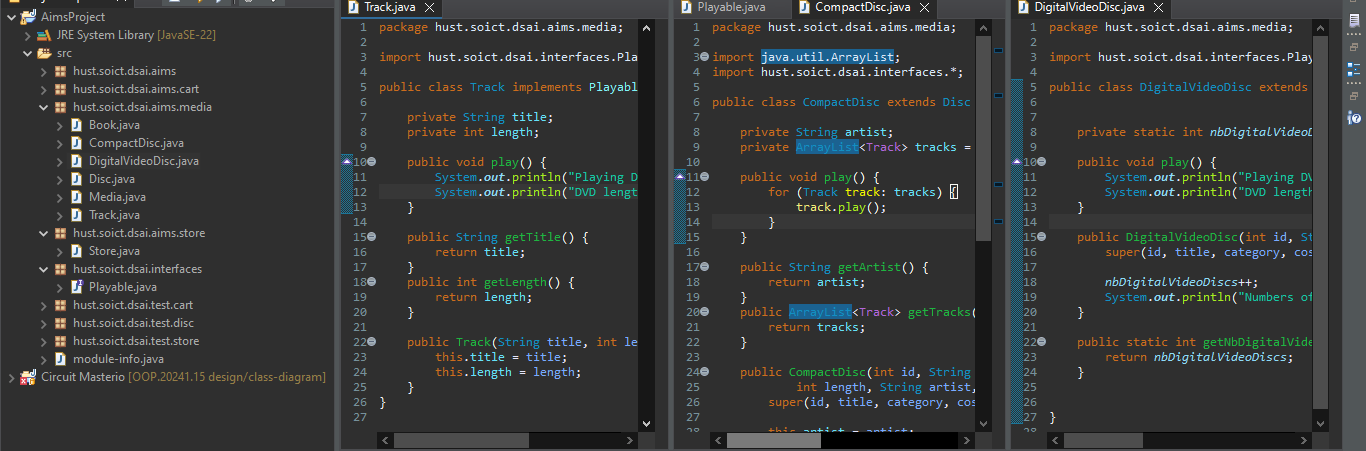
} else {

System.***out***.println("Track not found: " + track.getTitle());

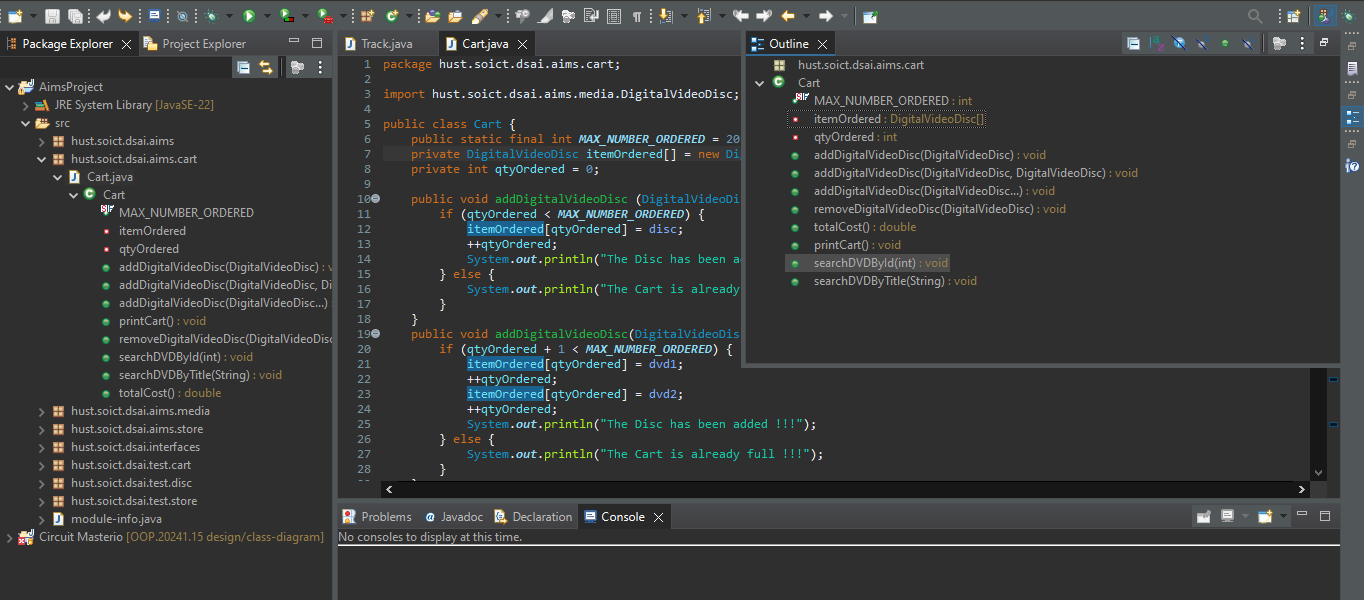
}



1. Create the Playable interface

* Tạo interface Playable:
* 
* Thêm phương thức Play() cho CompactDisc, DigitalVideoDisc và Track
* 

1. Update the Cart class to work with Media.

* Hình ảnh mở phần outline cho cart.
* 
* Đoạn mã cho hàm addMedia, removeMedia và totalCost mới:

public void addMedia(Media media) {

if (qtyOrdered < ***MAX\_NUMBER\_ORDERED***) {

itemsOrdered.add(media);

qtyOrdered ++;

System.***out***.println("Media has been added.");

} else {

System.***out***.println("There are not enough space.");

}

}

public void removeMedia(Media media) {

if (itemsOrdered.contains(media)) {

itemsOrdered.remove(media);

qtyOrdered --;

System.***out***.println("Media has been removed.");

} else {

System.***out***.println("Media does not exsist.");

}

}

public double totalCost() {

double totalCost = 0;

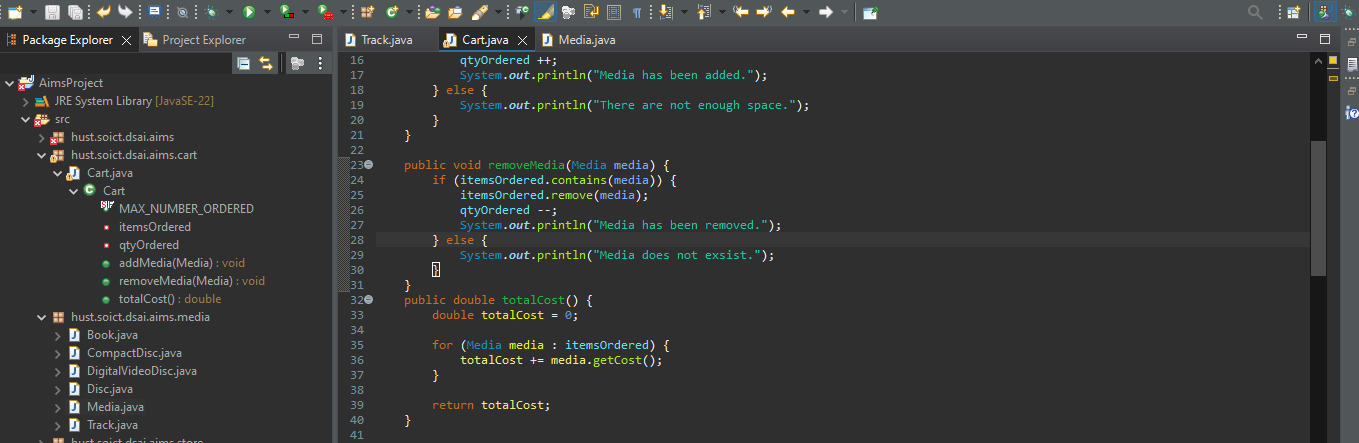
for (Media media : itemsOrdered) {

totalCost += media.getCost();

}

return totalCost;

}

* 

1. Update the Store class to work with Media

* Đoạn mã cho các hàm trong Store:

public class Store {

public static final int ***MAX\_NUMBER\_ORDERED*** = 20;

private ArrayList<Media> itemsInStore = new ArrayList<Media>();

private int qtyOrdered = 0;

public void addMedia(Media media) {

if (qtyOrdered < ***MAX\_NUMBER\_ORDERED***) {

itemsInStore.add(media);

qtyOrdered ++;

System.***out***.println("Media has been added.");

} else {

System.***out***.println("There are not enough space.");

}

}

public void removeMedia(Media media) {

if (itemsInStore.contains(media)) {

itemsInStore.remove(media);

qtyOrdered --;

System.***out***.println("Media has been removed.");

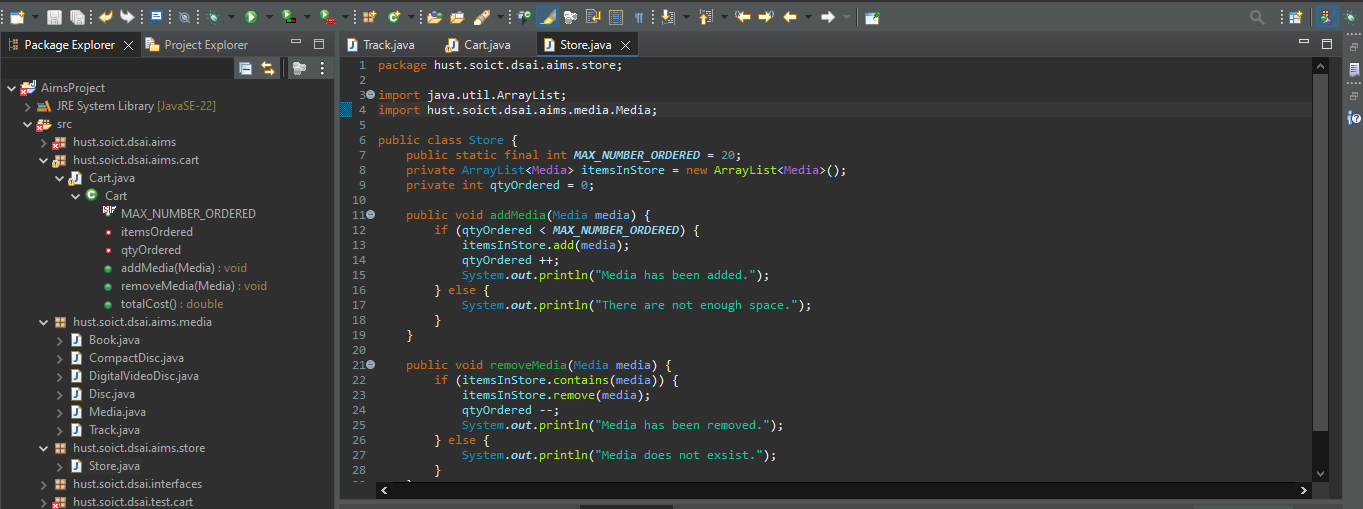
} else {

System.***out***.println("Media does not exsist.");

}

}

}

* 

1. Constructors of whole classes and parent classes
2. Unique item in a list

* Đoạn mã override cho equals trong lớp Media

*@Override*

public boolean equals(Object o) {

Media media = (Media) o;

try {

String title = media.getTitle();

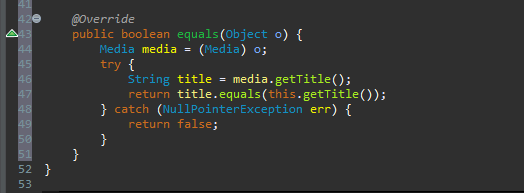
return title.equals(this.getTitle());

} catch (NullPointerException err) {

return false;

}

}

* 
* Đoạn mã override cho equals trong lớp Track

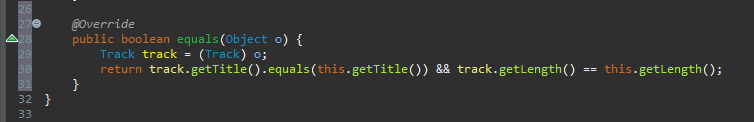
*@Override*

public boolean equals(Object o) {

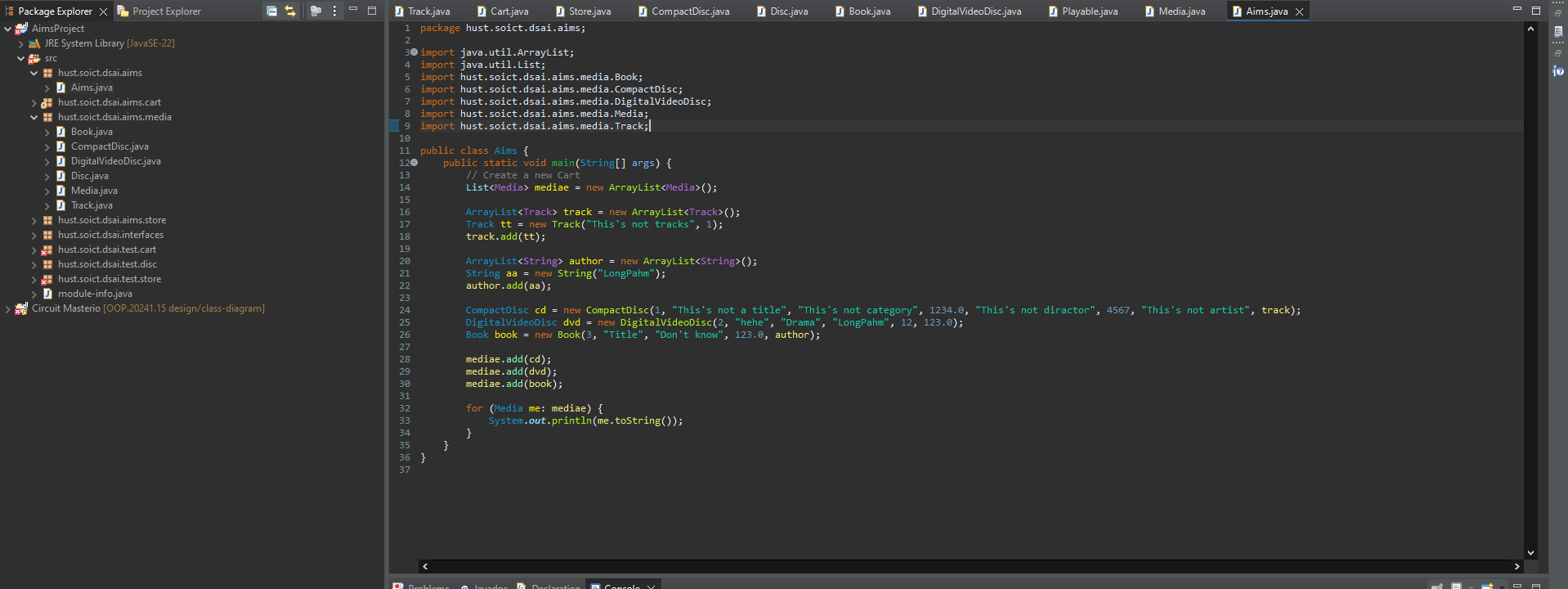
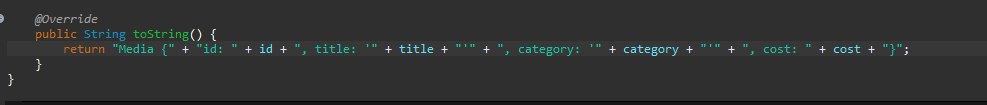
Track track = (Track) o;

return track.getTitle().equals(this.getTitle()) && track.getLength() == this.getLength();

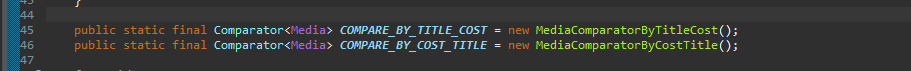
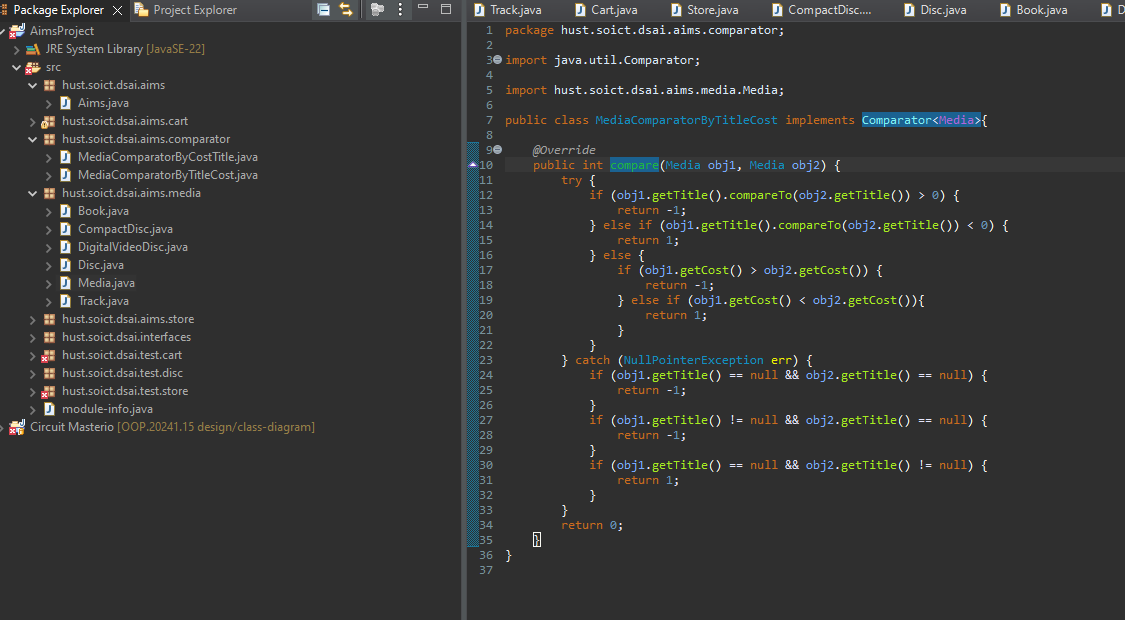
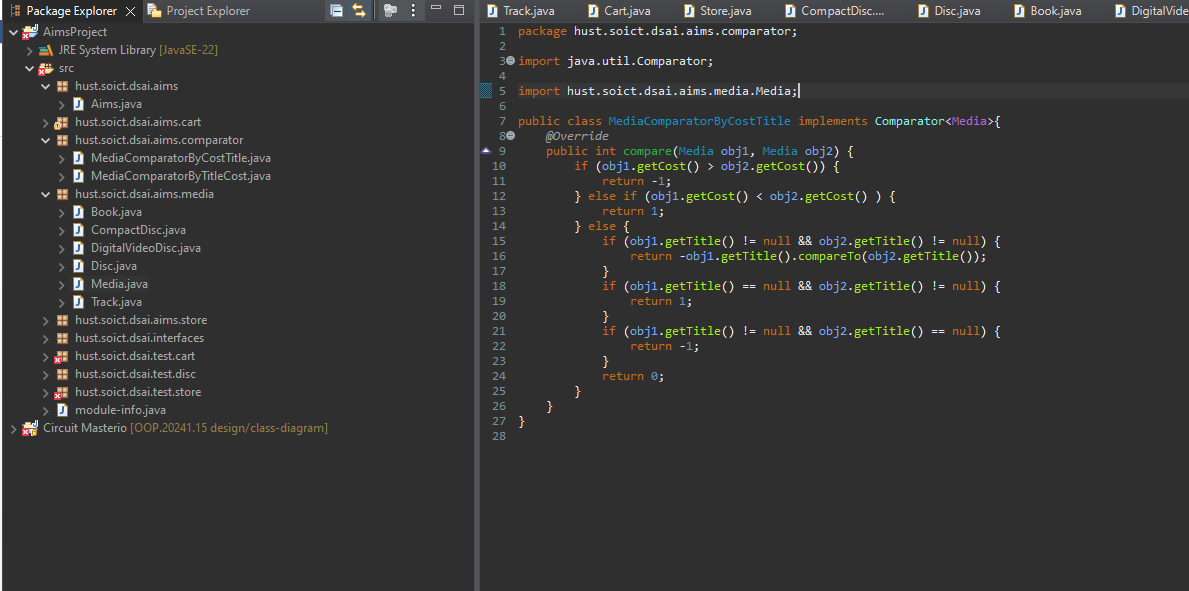
}

* 

1. Polymorphism with toString() method

* Đoạn mã mô phỏng tính đa hình
* 
* Ghi đè phương thức toString():
* 
* Output:
* 

1. Sort media in the cart

* Sắp xếp theo 2 tiêu chí bằng title và bằng cost:
* 
* Đoạn mã sắp xếp theo title:
* 
* Đoạn mã sắp xếp theo cost:
* 

1. Create a complete console application in the Aims class

* Mã nguồn trên đường dẫn: https://github.com/LongPhamplus/IT3103\_LTHDT