ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO THỰC HÀNH
IT3103-744528-2024.1

BÀI THỰC HÀNH LAB 04

Họ và tên sv: Nguyễn Trịnh Hoàng Nguyên
Lớp: K67-Việt Nhật 06
GVHD: Lê Thị Hoa
TA: Đặng Mạnh Cường

Hà Nội 12/2024

BÁO CÁO THỰC HÀNH LAB 4 LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

I. Create the Book Class

```java
package hust.soict.dsai.aims.media;

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

public class Book extends Media {  15 usages  Hoàng Nguyên

    //Attribute
    private List<String> authors = new ArrayList<String>();  6 usages

    //Constructor
    public Book(String title, String category, float cost) {  1 usage  Hoàng Nguyên
        super(title, category, cost);
    }

    public Book(String title, String category, float cost,List<String> authors) {  6 usages  Hoàng Nguyên
        super(title, category, cost);
        this.authors = authors;
    }

    @Override  Hoàng Nguyên
    //toString
    public String toString()
    {
        return "BOOK - " +getId() +". "+ getTitle() + " - " + "category: "+getCategory() + " - "+
                "list authors: "+ authors +": " + "Cost: "+getCost() +" $ ";
    }
```

Figure I-1.Book Class 1

```java
    //Add Authors
    public void addAuthor(String authorName) {  no usages  Hoàng Nguyên
        if(!authors.contains(authorName)){
            authors.add(authorName);
            System.out.println("Author added: " + authorName);
        }else
        {
            System.out.println("Author already added: " + authorName);
        }
    }

    //Remove Author
    public void removeAuthor(String authorName) {  no usages  Hoàng Nguyên
        if(authors.contains(authorName)){
            authors.remove(authorName);
            System.out.println("Author removed: " + authorName);
        }else{
            System.out.println("Author does not exist: " + authorName);
        }
    }
}
```

Figure I-2.Book Class 2

II. Creating the abstract Media class

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

```java
package hust.soict.dsai.aims.media;

import java.util.Comparator;

public class Media {  50 usages  4 inheritors
    //Attribute
    private static int nbMedia = 0;  2 usages
    private int id;  5 usages
    private String title;  4 usages
    private String category;  4 usages
    private float cost;  4 usages
    public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost();  1 usage
    public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle();  1 usage

    // Constructor
    public Media(String title, String category, float cost) {  3 usages
        this.id = nbMedia++;
        this.title = title;
        this.category = category;
        this.cost = cost;
    }

    public Media() {  1 usage
        this.id = nbMedia++;
    }

    //Getter
    public int getId() {  6 usages
        return id;
    }
    public String getTitle() {
        return title;
    }
    public String getCategory() {  3 usages
        return category;
    }
    public float getCost() {  12 usages
        return cost;
    }
```

Figure II-1.Media Class 1

```java
    //Setter
    public void setId(int id) {  no usages
        this.id = id;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public void setCategory(String category) {  3 usages
        this.category = category;
    }
    public void setCost(float cost) {  3 usages
        this.cost = cost;
    }
    //Method
    public boolean isMatch(String title)  1 usage  1 override
    {
        return this.getTitle().equals(title);
    }
    @Override
    public boolean equals(Object o){
        Media media = (Media) o;
        try{
            String title = media.getTitle();
            return title.equals(this.getTitle());
        } catch (NullPointerException e){
            return false;
        }
    }
    public String toString() {  3 overrides
        return "Media{" +
                "id=" + id + '\'' +
                "title='" + title + '\'' +
                ", category='" + category + '\'' +
                ", cost=" + cost +
                '}';
    }
}
```

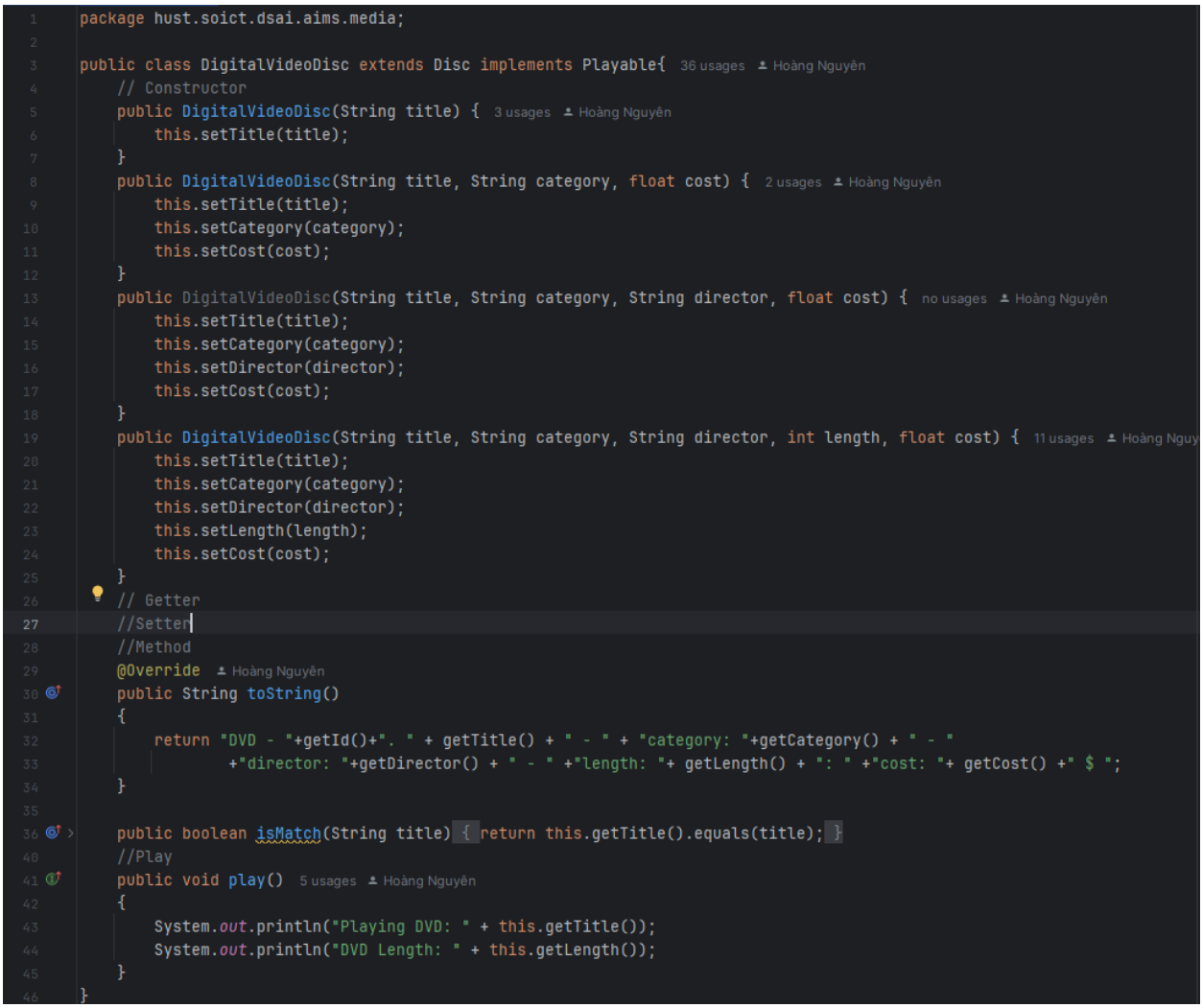
Figure II-2. Media Class 2

III. Creating the CompactDisc class

A. Create the Disc class extending the Media class

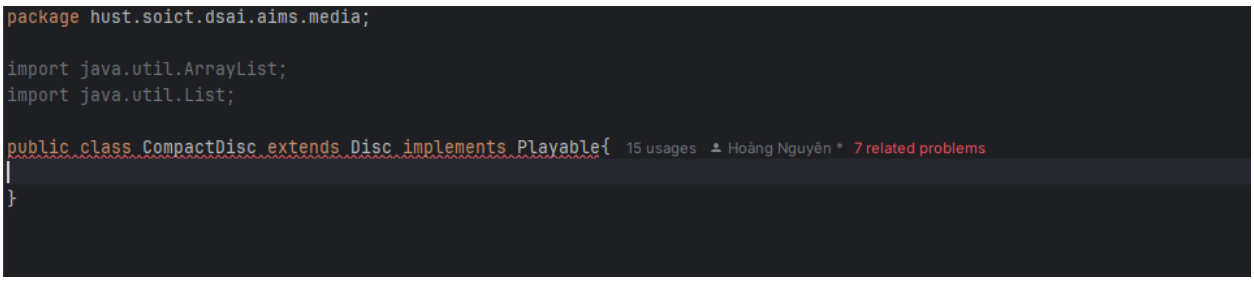
```java
package hust.soict.dsai.aims.media;

public class Disc extends Media {  2 usages  2 inheritors  Hoàng Nguyên

    //Attribute
    private int length;  3 usages
    private String director;  3 usages

    //Getter
    public int getLength() { return length; }
    public String getDirector() { return director; }

    //Setter
    public void setLength(int length) { this.length = length; }
    public void setDirector(String director) { this.director = director; }

    //Constructor
    public Disc(String title, String category, float cost, int length, String director) {  2 usages  Hoàng Nguyên
        super(title,category,cost);
        this.length = length;
        this.director = director;
    }
    public Disc() {  4 usages  Hoàng Nguyên
    }
}
```

Figure III-1. Disc Class



```java
package hust.soict.dsai.aims.media;

public class DigitalVideoDisc extends Disc implements Playable{ 36 usages  Hoàng Nguyên
    // Constructor
    public DigitalVideoDisc(String title) { 3 usages  Hoàng Nguyên
        this.setTitle(title);
    }
    public DigitalVideoDisc(String title, String category, float cost) { 2 usages  Hoàng Nguyên
        this.setTitle(title);
        this.setCategory(category);
        this.setCost(cost);
    }
    public DigitalVideoDisc(String title, String category, String director, float cost) { no usages  Hoàng Nguyên
        this.setTitle(title);
        this.setCategory(category);
        this.setDirector(director);
        this.setCost(cost);
    }
    public DigitalVideoDisc(String title, String category, String director, int length, float cost) { 11 usages  Hoàng Nguy
        this.setTitle(title);
        this.setCategory(category);
        this.setDirector(director);
        this.setLength(length);
        this.setCost(cost);
    }
    // Getter
    //Setter
    //Method
    @Override  Hoàng Nguyên
    public String toString()
    {
        return "DVD - "+getId()+". " + getTitle() + " - " + "category: "+getCategory() + " - "
                +"director: "+getDirector() + " - " +"length: "+ getLength() + ": " +"cost: "+ getCost() +" $ ";
    }

    public boolean isMatch(String title) { return this.getTitle().equals(title); }
    //Play
    public void play() 5 usages  Hoàng Nguyên
    {
        System.out.println("Playing DVD: " + this.getTitle());
        System.out.println("DVD Length: " + this.getLength());
    }
}
```

Figure III-2.DigitalVideoDisc Class



```java
package hust.soict.dsai.aims.media;

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

public class CompactDisc extends Disc implements Playable{ 15 usages  Hoàng Nguyên * 7 related problems

}
```

Figure III-3.CompactDisc Class

B. Create the Track class which models a track on a compact disc and will store information incuding the title and length of the track.

```java
package hust.soict.dsai.aims.media;

public class Track implements Playable {  31 usages  Hoàng Nguyên
    private String title;  3 usages
    private int length;  3 usages
    //Getter
    public String getTitle() { return title; }
    public int getLength() { return length; }
    //Constructor
    public Track(int length,String title){  20 usages  Hoàng Nguyên
        this.title = title;
        this.length = length;
    }
    //Method
    public void play()  5 usages  Hoàng Nguyên
    {
        System.out.println("Playing track: " + title);
        System.out.println("Length: " + length);
    }
    @Override  Hoàng Nguyên
    public boolean equals(Object o){
        Track track = (Track) o;
        try{
            String title = track.getTitle();
            int length = track.getLength();
            return title.equals(this.getTitle()) && length == this.getLength();
        } catch (NullPointerException e){
            return false;
        }
    }
}
```

Figure III-4.Track Class

C. Open the CompactDisc class

```java
package hust.soict.dsai.aims.media;

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

public class CompactDisc extends Disc implements Playable{ 15 usages  Hoàng Nguyên *

    //Attribute
    private String artist; 4 usages
    private List<Track> tracks = new ArrayList<Track>(); 7 usages

    //Getter
    public String getArtist() { 1 usage  Hoàng Nguyên
        return artist;
    }

    //Constructor
    public CompactDisc(String title,String category,float cost,int length,String director, 1 usage  Hoàng Nguyên
                       String artist) {
        super(title,category,cost,length,director);
        this.artist = artist;
    }
    public CompactDisc(String title,String category,float cost,int length,String director, 6 usages  Hoàng Nguyên
                       String artist, List<Track> tracks) {
        super(title,category,cost,length,director);
        this.artist = artist;
        this.tracks = tracks;
    }

    //toString
    @Override  Hoàng Nguyên *
    public String toString()
    {
        return "CD - " + getId() +". "+getTitle() + " - " + "category: "+getCategory() + " - "
                +"director: "+ getDirector()+" - " + "artist: "+artist
                +" - " + "length: "+getLength() + " - cost: " + getCost() +" $ ";
    }
    //addTrack
    public void addTrack(Track track) { no usages  Hoàng Nguyên
        int index = tracks.indexOf(track);
        if(index == -1) {
            System.out.println("Track is already in the list");
            return;
        }
```

Figure III-5.CompactDisc Class 1

```java
    //addTrack
    public void addTrack(Track track) {  no usages  Hoàng Nguyên
        int index = tracks.indexOf(track);
        if(index == -1) {
            System.out.println("Track is already in the list");
            return;
        }
        tracks.add(track);
        System.out.println("Track added to the list");
    }
    //Remove Track
    public void removeTrack(Track track) {  no usages  Hoàng Nguyên
        int index = tracks.indexOf(track);
        if(index == -1) {
            System.out.println("Track is not in the list");
            return;
        }
        tracks.remove(index);
        System.out.println("Track removed from the list");
    }
    //Get track length
    public int getLength() {  4 usages  Hoàng Nguyên
        int length = 0;
        for(Track track : tracks) {
            length += track.getLength();
        }
        setLength(length);
        return length;
    }
    //Play
    public void play()  5 usages  Hoàng Nguyên
    {
        System.out.println("Playing CD: " + this.getTitle());
        System.out.println("Artist: " + this.getArtist());
        System.out.println("Length: " + this.getLength());
        for(Track track : tracks) {
            track.play();
        }
    }
}
```

Figure III-6.CompactDisc Class 2

IV. Create the Playable interface

```java
package hust.soict.dsai.aims.media;

public interface Playable {  11 usages  3 implementations
    //Play
    public void play();  5 usages  3 implementations
}
```

Figure IV-1.Playable interface

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

```java
public void play()  5 usages  Hoàng Nguyên
{
    System.out.println("Playing DVD: " + this.getTitle());
    System.out.println("DVD Length: " + this.getLength());
}
```

Figure IV-2.Method play() của DigitalVideoDisc

```
public void play()  5 usages  Hoàng Nguyên
{
    System.out.println("Playing track: " + title);
    System.out.println("Length: " + length);
}
```

Figure IV-3.Method play() của Track

```
//Play
public void play()  5 usages  Hoàng Nguyên
{
    System.out.println("Playing CD: " + this.getTitle());
    System.out.println("Artist: " + this.getArtist());
    System.out.println("Length: " + this.getLength());
    for(Track track : tracks) {
        track.play();
    }
}
```

Figure IV-4.Method play() của CompactDisc

V. 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.

```java
package hust.soict.dsai.aims.cart;

import hust.soict.dsai.aims.media.Media;

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

public class Cart {  6 usages  Hoàng Nguyên
    // Attribute
    public static final int MAX_NUMBERS_ORDERED = 20;  1 usage
    private List<Media> itemsOrdered = new ArrayList<Media>();  14 usages
    // Method to add Media to the cart
    public void addMedia(Media media) {  6 usages  Hoàng Nguyên
        if(media!=null) {
            if (itemsOrdered.size() < MAX_NUMBERS_ORDERED) {
                for(Media m: itemsOrdered) {
                    if(media.equals(m)) {
                        System.out.println("Media is already in the Cart");
                        return;
                    }
                }
                itemsOrdered.add(media);
                System.out.println("Added media: " + media);
            } else {
                System.out.println("The cart is full. Can't add more media");
            }
        }
        System.out.println("Number of DVDs in the current cart: "+ itemsOrdered.size());
    }
    // Method to delete Media in the cart
    public void removeMedia(Media media) {  3 usages  Hoàng Nguyên
        //Search for media
        for(Media item: itemsOrdered) {
            if(item.equals(media)) {
                itemsOrdered.remove(media);
                System.out.println("Removed media: " + media);
                return;
            }
        }
        System.out.println("Not found media: "+ media.toString()+". Can't remove media");

    }
```

Figure V-1.Cart Class 1

```java
    // Method to calculate the total cost
    public float totalCost(){  1 usage  Hoàng Nguyên
        if (itemsOrdered.isEmpty()) {
            return 0;
        }
        float total = 0;
        for(Media item : itemsOrdered){
            total += item.getCost();
        }
        return total;
    }
    //Print Cart
    public void print(){  4 usages  Hoàng Nguyên
        System.out.println("**********************************CART*********************************");
        System.out.println("Ordered Items:");
        for(Media item : itemsOrdered){
            System.out.println(item.getId() + ". "+item.toString());
        }
        System.out.println("Total cost :" + totalCost());
        System.out.println("***********************************************************************");
    }

    //Search DVD in Cart by ID
    public Media searchByID(int id)  5 usages  Hoàng Nguyên
    {
        for(Media item : itemsOrdered){
            if(item.getId() == id)
            {
                System.out.println("DVD founded: " + item.toString());
                return item;
            }
        }
        System.out.println("DVD not found");
        return null;
    }
```

Figure V-2.Cart Class 2

```
79      //Search DVD in Cart by Title
80      public Media searchByTitle(String title)  5 usages  Hoàng Nguyên
81      {
82          for(Media item : itemsOrdered){
83              if(item.isMatch(title))
84              {
85                  System.out.println("DVD founded: " + item.toString());
86                  return item;
87              }
88          }
89          System.out.println("DVD not found");
90          return null;
91      }
92      //Sort list media in Cart by Title -> Cost
93      public void sortTitleCost()  2 usages  Hoàng Nguyên
94      {
95          Collections.sort(itemsOrdered, Media.COMPARE_BY_TITLE_COST);
96          System.out.println("Cart sorted by title.");
97          print();
98      }
99      //Sort list media in Cart by Cost -> Title
100     public void sortCostTitle()  2 usages  Hoàng Nguyên
101     {
102         Collections.sort(itemsOrdered, Media.COMPARE_BY_COST_TITLE);
103         System.out.println("Cart sorted by cost.");
104         print();
105     }
106     //Clear cart
107     public void clear()  1 usage  Hoàng Nguyên
108     {
109         itemsOrdered.clear();
110     }
111 }
112
113
```

Figure V-3.Cart Class 3

VI. Update the Store class to work with Media

```java
package hust.soict.dsai.aims.store;

import hust.soict.dsai.aims.media.Media;

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

public class Store {  6 usages  Hoàng Nguyên
    //Attribute
    private List<Media> itemsInStore = new ArrayList<~>();  7 usages
    //Add DVD to Store
    public void addMedia(Media media) {  21 usages  Hoàng Nguyên
        if(media != null) {
            for(Media m : itemsInStore) {
                if(media.equals(m)) {
                    System.out.println("Media is already in the store");
                    return;
                }
            }
            itemsInStore.add(media);
            System.out.println("Added media: " + media);
        }
    }
    //Remove DVD in Store
    public void removeMedia(Media media) {  4 usages  Hoàng Nguyên
        for(Media item: itemsInStore) {
            if(item.equals(media)) {
                itemsInStore.remove(media);
                System.out.println("Removed media: " + media);
                return;
            }
        }
        System.out.println("Not found media: "+ media.toString()+". Can't remove media");
    }

    //Print list DVD in Store
    public void print(){  3 usages  Hoàng Nguyên
        for (Media item : itemsInStore) {
            System.out.println(item.toString());
        }
    }

}
```

Figure VI-1.Store Class 1

```
42
43      //Search Media in Store by Title
44      public Media SearchByTitle(String title) {  4 usages  Hoàng Nguyên
45          if(title != null &&!title.trim().isEmpty()) {
46              title = title.trim();
47              for(Media item: itemsInStore) {
48                  if(item.getTitle().equalsIgnoreCase(title)) {
49                      return item;
50                  }
51              }
52          }
53          return null;
54      }
55      //Search Media in Store by ID
56      public Media SearchById(int id) {  4 usages  Hoàng Nguyên
57              for(Media item: itemsInStore) {
58                  if(item.getId() == id) {
59                      return item;
60                  }
61              }
62          return null;
63      }
64  }
65
```

Figure VI-2.Store Class 2

VII. Constructors of whole classes and parent classes

```
//Constructor
public Track(int length,String title){  20 usages  Hoàng Nguyên
    this.title = title;
    this.length = length;
}
```

Figure VII-1.Constructor Track Class

```
//Constructor
public CompactDisc(String title,String category,float cost,int length,String director,  1 usage  Hoàng Nguyên
                   String artist) {
    super(title,category,cost,length,director);
    this.artist = artist;
}
public CompactDisc(String title,String category,float cost,int length,String director,  6 usages  Hoàng Nguyên
                   String artist, List<Track> tracks) {
    super(title,category,cost,length,director);
    this.artist = artist;
    this.tracks = tracks;
}
```

Figure VII-2.Constructor CompactDisc Class

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

```java
    // Constructor
    public Media(String title, String category, float cost) { 3 usages  Hoàng Nguyên
        this.id = nbMedia++;
        this.title = title;
        this.category = category;
        this.cost = cost;
    }

    public Media() { 1 usage  Hoàng Nguyên
        this.id = nbMedia++;
    }
```

Figure VII-3.Constructor Media Class

```java
//Constructor
public Disc(String title, String category, float cost, int length, String director) { 2 usages  Hoàng Nguyên
    super(title,category,cost);
    this.length = length;
    this.director = director;
}
public Disc() { 4 usages  Hoàng Nguyên
}
```

Figure VII-4.Constructor Disc Class

VIII. 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 lắp vào danh sách.

```java
@Override  Hoàng Nguyên
public boolean equals(Object o){
    Media media = (Media) o;
    try{
        String title = media.getTitle();
        return title.equals(this.getTitle());
    } catch (NullPointerException e){
        return false;
    }
}
```

Figure VIII-1.Override equals in Media Class

```java
@Override  Hoàng Nguyên
public boolean equals(Object o){
    Track track = (Track) o;
    try{
        String title = track.getTitle();
        int length = track.getLength();
        return title.equals(this.getTitle()) && length == this.getLength();
    } catch (NullPointerException e){
        return false;
    }
}
```

Figure VIII-2.Override equals in Track Class

IX. Polymorphism with toString() method

```java
package hust.soict.dsai.test.media;

import hust.soict.dsai.aims.media.Book;
import hust.soict.dsai.aims.media.CompactDisc;
import hust.soict.dsai.aims.media.DigitalVideoDisc;
import hust.soict.dsai.aims.media.Media;

import java.util.ArrayList;
import java.util.List;
public class TestMedia {  Hoàng Nguyên *
    public static void main(String[] args) {  Hoàng Nguyên *
        List<Media> mediae = new ArrayList<>();

        CompactDisc cd = new CompactDisc( title: "CD01", category: "Fantasy", cost: 13f, length: 123, director: "Lucas", artist: "John");
        DigitalVideoDisc dvd = new DigitalVideoDisc( title: "DVD Title 1", category: "Action", director: "Director 1", length: 120, cost: 15.99f);
        Book book = new Book( title: "Book Title 1", category: "Fiction", cost: 29.99f,List.of( e1: "Robert C. Martin"));

        mediae.add(cd);
        mediae.add(dvd);
        mediae.add(book);

        for(Media media : mediae){
            System.out.println(media.toString());
        }
    }
}
```

Figure IX-1.Code mô phỏng Polymorphism

```java
@Override  Hoàng Nguyên
//toString
public String toString()
{
    return "BOOK - " +getId() +". "+ getTitle() + " - " + "category: "+getCategory() + " - "+
            "list authors: "+ authors +": " + "Cost: "+getCost() +" $ ";
}
```

Figure IX-2.Override toString() in Book Class

```java
//toString
@Override  Hoàng Nguyên
public String toString()
{
    return "CD - " + getId() +". "+getTitle() + " - " + "category: "+getCategory() + " - " +"director: "+ getDirector()+" - " +
            "artist: "+artist
            +" - " + "length: "+getLength() + " - cost: " + getCost() +" $ ";
}
```

Figure IX-3.Override toString() in Compact Disc Class

```java
@Override  Hoàng Nguyên
public String toString()
{
    return "DVD - "+getId()+". " + getTitle() + " - " + "category: "+getCategory() + " - "
            +"director: "+getDirector() + " - " +"length: "+ getLength() + ": " +"cost: "+ getCost() +" $ ";
}
```

Figure IX-4.Override toString() in DigitalVideoDisc Class

```
public String toString() {  3 overrides  Hoàng Nguyên
    return "Media{" +
            "id=" + id + '\'' +
            "title='" + title + '\'' +
            ", category='" + category + '\'' +
            ", cost=" + cost +
            '}';
}
```

Figure IX-5.Override toString() in Media Class

Kết quả:

```
CD - 0. CD01 - category: Fantasy - director: Lucas - artist: John - length: 0 - cost: 13.0 $
DVD - 1. DVD Title 1 - category: Action - director: Director 1 - length: 120: cost: 15.99 $
BOOK - 2. Book Title 1 - category: Fiction - list authors: [Robert C. Martin]: Cost: 29.99 $

Process finished with exit code 0
```

Figure IX-6.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.

X. 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

```
13    public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost();  1 usage
14    public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle();  1 usage
```

Figure X-1.Add the comparators as attributes of the Media class

```java
package hust.soict.dsai.aims.media;

import java.util.Comparator;

public class MediaComparatorByCostTitle implements Comparator<Media> { 1 usage  Hoàng Nguyên
    @Override  Hoàng Nguyên
    public int compare(Media m2, Media m1) {
        if(m1.getCost() > m2.getCost())
        {
            return -1;
        }else if(m1.getCost() < m2.getCost())
        {
            return 1;
        }else {
            if(m1.getTitle() != null && m2.getTitle() != null)
            {
                return - m1.getTitle().compareTo( m2.getTitle());
            }
            if(m1.getTitle() == null && m2.getTitle() != null)
                return 1;
            if(m1.getTitle() != null && m2.getTitle() == null)
                return -1;
            return 0;
        }
    }
}
```

Figure X-2.MediaComparatorByCostTitle Class

```java
package hust.soict.dsai.aims.media;

import java.util.Comparator;

public class MediaComparatorByTitleCost implements Comparator<Media> { 1 usage  Hoàng Nguyên
    @Override  Hoàng Nguyên
    public int compare(Media m2, Media m1) {
        try {
            if (m1.getTitle().compareTo(m2.getTitle()) > 0) {
                return -1;
            } else if (m1.getTitle().compareTo(m2.getTitle()) < 0)
                return 1;
            else {
                if(m1.getCost() > m2.getCost())
                    return -1;
                else if(m1.getCost() < m2.getCost())
                    return 1;
            }
        }catch (NullPointerException e) {
            if(m1.getTitle() == null && m2.getTitle() == null)
                return -1;
            if(m1.getTitle() != null && m2.getTitle() == null)
                return -1;
            if(m1.getTitle() == null && m2.getTitle() != null)
                return 1;
        }
        return 0;
    }
}
```

Figure X-3.MediaComparatorByTitleCost Class

```java
//Sort list media in Cart by Title -> Cost
public void sortTitleCost() 2 usages  Hoàng Nguyên
{
    Collections.sort(itemsOrdered, Media.COMPARE_BY_TITLE_COST);
    System.out.println("Cart sorted by title.");
    print();
}
//Sort list media in Cart by Cost -> Title
public void sortCostTitle() 2 usages  Hoàng Nguyên
{
    Collections.sort(itemsOrdered, Media.COMPARE_BY_COST_TITLE);
    System.out.println("Cart sorted by cost.");
    print();
}
```

Figure X-4. Áp dụng 2 Comparator vào Cart Class

XI. Create a complete console application in the Aims class

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

Figure XI-1.Màn hình chính

A. Người dùng chọn 1: View store

```
1
BOOK - 0. Clean Code - category: Programming - list authors: [Robert C. Martin]: Cost: 45.5 $
BOOK - 1. The Pragmatic Programmer - category: Programming - list authors: [Andrew Hunt David Thomas]: Cost: 50.0 $
BOOK - 2. Introduction to Algorithms - category: Computer Science - list authors: [Thomas H. Cormen Charles E. Leiserson Ronald L. Rivest Clifford Stein]: Cost: 60.0 $
BOOK - 3. Design Patterns - category: Programming - list authors: [Erich Gamma Richard Helm Ralph Johnson John Vlissides]: Cost: 40.0 $
BOOK - 4. Artificial Intelligence: A Modern Approach - category: AI - list authors: [Stuart Russell Peter Norvig]: Cost: 70.0 $
DVD - 5. The Matrix - category: Science Fiction - director: Lana Wachowski - length: 136: cost: 15.0 $
DVD - 6. Inception - category: Science Fiction - director: Christopher Nolan - length: 148: cost: 18.0 $
DVD - 7. The Godfather - category: Crime - director: Francis Ford Coppola - length: 175: cost: 20.0 $
DVD - 8. Pulp Fiction - category: Crime - director: Quentin Tarantino - length: 154: cost: 16.0 $
DVD - 9. The Lord of the Rings: The Fellowship of the Ring - category: Fantasy - director: Peter Jackson - length: 178: cost: 25.0 $
CD - 10. Thriller - category: Music - director: Quincy Jones - artist: Michael Jackson - length: 22 - cost: 30.0 $
CD - 11. Back in Black - category: Music - director: Robert John Lange - artist: AC/DC - length: 14 - cost: 25.0 $
CD - 12. Abbey Road - category: Music - director: George Martin - artist: The Beatles - length: 13 - cost: 35.0 $
CD - 13. The Dark Side of the Moon - category: Music - director: Alan Parsons - artist: Pink Floyd - length: 17 - cost: 40.0 $
CD - 14. Rumours - category: Music - director: Ken Caillat - artist: Fleetwood Mac - length: 15 - cost: 28.0 $
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 XI-2.View Store

1. Người dùng chọn 1: See a media's details

```
1
Find with:
1. ID:
2. Title:
0. Back
1
Enter the ID of a media:
0
BOOK - 0. Clean Code - category: Programming - list authors: [Robert C. Martin]: Cost: 45.5 $
Options:
--------------------------------
1. Add to cart
2. Play
0. Back
--------------------------------
Please choose a number: 0-1-2
```

Figure XI-3.See a media's details by ID

```
Find with:
1. ID:
2. Title:
0. Back
2
Enter the name of a media:
rumours
CD - 14. Rumours - category: Music - director: Ken Caillat - artist: Fleetwood Mac - length: 15 - cost: 28.0 $
Options:
--------------------------------
1. Add to cart
2. Play
0. Back
--------------------------------
Please choose a number: 0-1-2
```

Figure XI-4.See a media's details by Title

```
BOOK - 0. Clean Code - category: Programming - list authors: [Robert C. Martin]: Cost: 45.5 $
Options:
--------------------------------
1. Add to cart
2. Play
0. Back
--------------------------------
Please choose a number: 0-1-2
1
Added media: BOOK - 0. Clean Code - category: Programming - list authors: [Robert C. Martin]: Cost: 45.5 $
Number of DVDs in the current cart: 1
```

Figure XI-5. Thêm vào giỏ hàng

```
BOOK - 1. The Pragmatic Programmer - category: Programming - list authors: [Andrew Hunt David Thomas]: Cost: 50.0 $
Options:
--------------------------------
1. Add to cart
2. Play
0. Back
--------------------------------
Please choose a number: 0-1-2
2
This media cannot be played.
```

Figure XI-6. Play() không được vì đây là Book

```
Enter the ID of a media:
10
CD - 10. Thriller - category: Music - director: Quincy Jones - artist: Michael Jackson - length: 22 - cost: 30.0 $
Options:
--------------------------------
1. Add to cart
2. Play
0. Back
--------------------------------
Please choose a number: 0-1-2
2
Playing CD: Thriller
Artist: Michael Jackson
Length: 22
Playing track: Wanna Be Startin' Somethin'
Length: 6
Playing track: Thriller
Length: 6
Playing track: Beat It
Length: 5
Playing track: Billie Jean
Length: 5
```

Figure XI-7.play() được vì đây là CD

2. Người dùng chọn 2: Add a media to cart

```
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
Find with:
1. ID:
2. Title:
0. Back
1
Enter the ID of a media:
10
Added media: CD - 10. Thriller - category: Music - director: Quincy Jones - artist: Michael Jackson - length: 22 - cost: 30.0 $
```

Figure XI-8.Add a media to cart

3. Người dùng chọn 3:

Phương thức play() giống với đã Demo ở trên

```
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
Find with:
1. ID:
2. Title:
0. Back
1
Enter the ID of a media:
11
Playing CD: Back in Black
Artist: AC/DC
Length: 14
Playing track: Hells Bells
Length: 5
Playing track: Shout to Thrill
Length: 5
Playing track: Blank in black
Length: 4
```

Figure XI-9.play()

4. Người dùng chọn 4: Xem Cart hiện tại có gì

Trước đó Add thêm vài Media vào Cart để theo dõi dễ dàng hơn :

```
Enter the ID of a media:
14
Added media: CD - 14. Rumours - category: Music - director: Ken Caillat - artist: Fleetwood Mac - length: 15 - cost: 28.0 $
Number of DVDs in the current cart: 3
```

Figure XI-10. Thêm Media vào Cart 1

```
Enter the ID of a media:
12
Added media: CD - 12. Abbey Road - category: Music - director: George Martin - artist: The Beatles - length: 13 - cost: 35.0 $
Number of DVDs in the current cart: 4
```

Figure XI-11.Thêm Media vào Cart 2

```
5
Added media: DVD - 5. The Matrix - category: Science Fiction - director: Lana Wachowski - length: 136: cost: 15.0 $
Number of DVDs in the current cart: 5
```

Figure XI-12.Thêm Media vào Cart 3

Sắp xếp theo tên -> giá

```
Please choose a number: 0-1-2-3-4
4
Choose a Sort:
------------------------------
1. Media Comparator By Title - Cost
2. Media Comparator By Cost - Title
0. Back
------------------------------
Please choose a number: 0-1-2
1
Cart sorted by title.
*****************************CART*****************************
Ordered Items:
12. CD - 12. Abbey Road - category: Music - director: George Martin - artist: The Beatles - length: 13 - cost: 35.0 $
0. BOOK - 0. Clean Code - category: Programming - list authors: [Robert C. Martin]: Cost: 45.5 $
14. CD - 14. Rumours - category: Music - director: Ken Caillat - artist: Fleetwood Mac - length: 15 - cost: 28.0 $
5. DVD - 5. The Matrix - category: Science Fiction - director: Lana Wachowski - length: 136: cost: 15.0 $
10. CD - 10. Thriller - category: Music - director: Quincy Jones - artist: Michael Jackson - length: 22 - cost: 30.0 $
Total cost :153.5
*****************************************************************
```

Figure XI-13.Sắp xếp theo tên -> giá

Sắp xếp theo giá -> tên

```
Please choose a number: 0-1-2-3-4
4
Choose a Sort:
------------------------------
1. Media Comparator By Title - Cost
2. Media Comparator By Cost - Title
0. Back
------------------------------
Please choose a number: 0-1-2
2
Cart sorted by cost.
*****************************CART*****************************
Ordered Items:
5. DVD - 5. The Matrix - category: Science Fiction - director: Lana Wachowski - length: 136: cost: 15.0 $
14. CD - 14. Rumours - category: Music - director: Ken Caillat - artist: Fleetwood Mac - length: 15 - cost: 28.0 $
10. CD - 10. Thriller - category: Music - director: Quincy Jones - artist: Michael Jackson - length: 22 - cost: 30.0 $
12. CD - 12. Abbey Road - category: Music - director: George Martin - artist: The Beatles - length: 13 - cost: 35.0 $
0. BOOK - 0. Clean Code - category: Programming - list authors: [Robert C. Martin]: Cost: 45.5 $
Total cost :153.5
*****************************************************************
```

Figure XI-14.Sắp xếp theo giá -> tên

B. Người dùng chọn 2: Update Store

1. Them vao Store

```
2
Options:
--------------------------------
1. Add to Store
2. Remove from Store
0. Back
--------------------------------
Please choose a number: 0-1-2
1
Choose a media type:
--------------------------------
1. Book
2. Digital Video Disc
3. Compact Disc
0. Back
--------------------------------
Please choose a number: 0-1-2-3
```

Figure XI-15.Thêm vào Store

Chọn loại Media để thêm:

```
1
Enter title of new Book: Conan
Enter category: TrinhTham
Enter cost:15
Enter authors (separated by commas): Author1,Author2
Added media: BOOK - 15. Conan  - category: TrinhTham - list authors: [Author1, Author2]: Cost: 15.0 $
```

Figure XI-16.Thêm Book vào Store

```
Please choose a number: 0-1-2-3
2
Enter title of new DVD: DVD
Enter category: Category
Enter Director: Director
Enter length: 15
Enter cost: 20
Added media: DVD - 16. DVD - category: Category - director: Director - length: 15: cost: 20.0 $
```

Figure XI-17.Thêm DVD vào Store

```
3
Enter title of new CD: CD
Enter category: Cate
Enter Director: Direc
Enter length: 30
Enter cost: 40
Enter artist: Art
Enter number of tracks: 2
Enter title of track 1: Track1
Enter length of track 1: 14
Enter title of track 2: Track2
Enter length of track 2: 16
Added media: CD - 17. CD - category: Cate - director: Direc - artist: Art - length: 30 - cost: 40.0 $
```

Figure XI-18.Thêm CD vào Store

2. Xóa Media trong Store

```
Options:
------------------------------
1. Add to Store
2. Remove from Store
0. Back
------------------------------
Please choose a number: 0-1-2
2
Find with:
1. ID:
2. Title:
0. Back
1
Enter the ID of a media:
14
Removed media: CD - 14. Rumours - category: Music - director: Ken Caillat - artist: Fleetwood Mac - length: 15 - cost: 28.0 $
```

Figure XI-19.Xóa Media trong Store

In ra Store

```
BOOK - 0. Clean Code - category: Programming - list authors: [Robert C. Martin]: Cost: 45.5 $
BOOK - 1. The Pragmatic Programmer - category: Programming - list authors: [Andrew Hunt David Thomas]: Cost: 50.0 $
BOOK - 2. Introduction to Algorithms - category: Computer Science - list authors: [Thomas H. Cormen Charles E. Leiserson Ronald L. Rivest Clifford Stein]: Cost: 60.0 $
BOOK - 3. Design Patterns - category: Programming - list authors: [Erich Gamma Richard Helm Ralph Johnson John Vlissides]: Cost: 40.0 $
BOOK - 4. Artificial Intelligence: A Modern Approach - category: AI - list authors: [Stuart Russell Peter Norvig]: Cost: 70.0 $
DVD - 5. The Matrix - category: Science Fiction - director: Lana Wachowski - length: 136: cost: 15.0 $
DVD - 6. Inception - category: Science Fiction - director: Christopher Nolan - length: 148: cost: 18.0 $
DVD - 7. The Godfather - category: Crime - director: Francis Ford Coppola - length: 175: cost: 20.0 $
DVD - 8. Pulp Fiction - category: Crime - director: Quentin Tarantino - length: 154: cost: 16.0 $
DVD - 9. The Lord of the Rings: The Fellowship of the Ring - category: Fantasy - director: Peter Jackson - length: 178: cost: 25.0 $
CD - 10. Thriller - category: Music - director: Quincy Jones - artist: Michael Jackson - length: 22 - cost: 30.0 $
CD - 11. Back in Black - category: Music - director: Robert John Lange - artist: AC/DC - length: 14 - cost: 25.0 $
CD - 12. Abbey Road - category: Music - director: George Martin - artist: The Beatles - length: 13 - cost: 35.0 $
CD - 13. The Dark Side of the Moon - category: Music - director: Alan Parsons - artist: Pink Floyd - length: 17 - cost: 40.0 $
BOOK - 15. Conan  - category: TrinhTham - list authors: [Author1, Author2]: Cost: 15.0 $
DVD - 16. DVD - category: Category - director: Director - length: 15: cost: 20.0 $
CD - 17. CD - category: Cate - director: Direc - artist: Art - length: 30 - cost: 40.0 $
```

Figure XI-20.In ra Store

C. Nguoi dung chon 3: See Curent Cart

1. Filter

```
**********CART**********
Ordered Items:
12. CD - 12. Abbey Road - category: Music - director: George Martin - artist: The Beatles - length: 13 - cost: 35.0 $
0. BOOK - 0. Clean Code - category: Programming - list authors: [Robert C. Martin]: Cost: 45.5 $
14. CD - 14. Rumours - category: Music - director: Ken Caillat - artist: Fleetwood Mac - length: 15 - cost: 28.0 $
5. DVD - 5. The Matrix - category: Science Fiction - director: Lana Wachowski - length: 136: cost: 15.0 $
10. CD - 10. Thriller - category: Music - director: Quincy Jones - artist: Michael Jackson - length: 22 - cost: 30.0 $
Total cost :153.5
**********
```

Figure XI-21.Đây là Cart hiện tại

```
3
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
Choose a Filter:
--------------------------------
1. Filter by ID
2. Filter by Title
0. Back
--------------------------------
Please choose a number: 0-1-2
1
Enter ID: 0
DVD founded: BOOK - 0. Clean Code - category: Programming - list authors: [Robert C. Martin]: Cost: 45.5 $
```

Figure XI-22.Filter theo ID

```
Choose a Filter:
--------------------------------
1. Filter by ID
2. Filter by Title
0. Back
--------------------------------
Please choose a number: 0-1-2
2
Enter Title: Thriller
DVD founded: CD - 10. Thriller - category: Music - director: Quincy Jones - artist: Michael Jackson - length: 22 - cost: 30.0 $
```

Figure XI-23.Filter theo Title

2. Sort medias in cart

```
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
Choose a Sort:
--------------------------------
1. Media Comparator By Title - Cost
2. Media Comparator By Cost - Title
0. Back
--------------------------------
Please choose a number: 0-1-2
```

Figure XI-24.Chọn sắp xếp các Media trong Cart(tương tự như trên)

3. Xóa Media khỏi Cart

```
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
Remove by:
1. ID:
2. Title:
0. Back
1
Enter ID:
14
DVD founded: CD - 14. Rumours - category: Music - director: Ken Caillat - artist: Fleetwood Mac - length: 15 - cost: 28.0 $
Removed media: CD - 14. Rumours - category: Music - director: Ken Caillat - artist: Fleetwood Mac - length: 15 - cost: 28.0 $
```

Figure XI-25.Xóa Media khỏi Cart

```
Cart sorted by title.
***************************CART*****************************
Ordered Items:
12. CD - 12. Abbey Road - category: Music - director: George Martin - artist: The Beatles - length: 13 - cost: 35.0 $
0. BOOK - 0. Clean Code - category: Programming - list authors: [Robert C. Martin]: Cost: 45.5 $
5. DVD - 5. The Matrix - category: Science Fiction - director: Lana Wachowski - length: 136: cost: 15.0 $
10. CD - 10. Thriller - category: Music - director: Quincy Jones - artist: Michael Jackson - length: 22 - cost: 30.0 $
Total cost :125.5
************************************************************
```

Figure XI-26.Cart sau khi xóa

4. Play a media

```
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
Search media by:
1. ID:
2. Title:
0. Back
1
Enter ID:
10
DVD founded: CD - 10. Thriller - category: Music - director: Quincy Jones - artist: Michael Jackson - length: 22 - cost: 30.0 $
Playing CD: Thriller
Artist: Michael Jackson
Length: 22
Playing track: Wanna Be Startin' Somethin'
Length: 6
Playing track: Thriller
Length: 6
Playing track: Beat It
Length: 5
Playing track: Billie Jean
Length: 5
```

Figure XI-27.Play Media

5. Place Order

```
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
Order placed! Cart is now empty.
```

Figure XI-28.Place Order

Cart sau khi Place Order:

```
Cart sorted by title.
***************************CART*******************************
Ordered Items:
Total cost :0.0
******************************************************************
```

Figure XI-29.Cart sau khi Place Order

D. Exit

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

Process finished with exit code 0
```

Figure XI-30.Exit AIMS

XII. Class Diagram

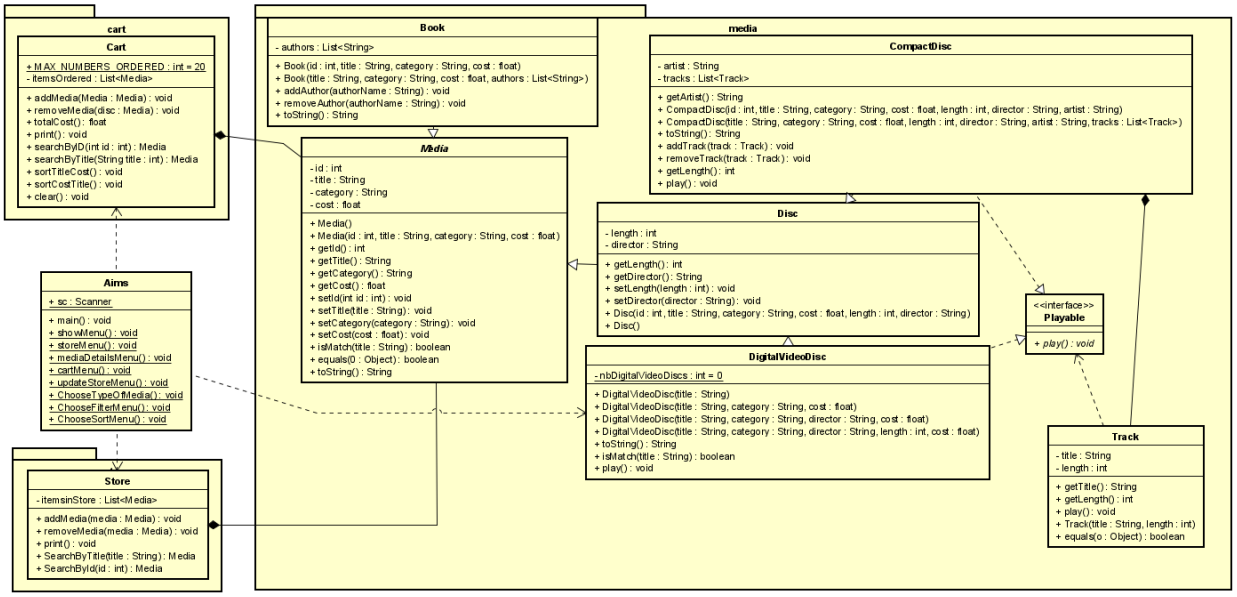


Figure XII-1.Class Diagram

XIII. UseCase Diagram

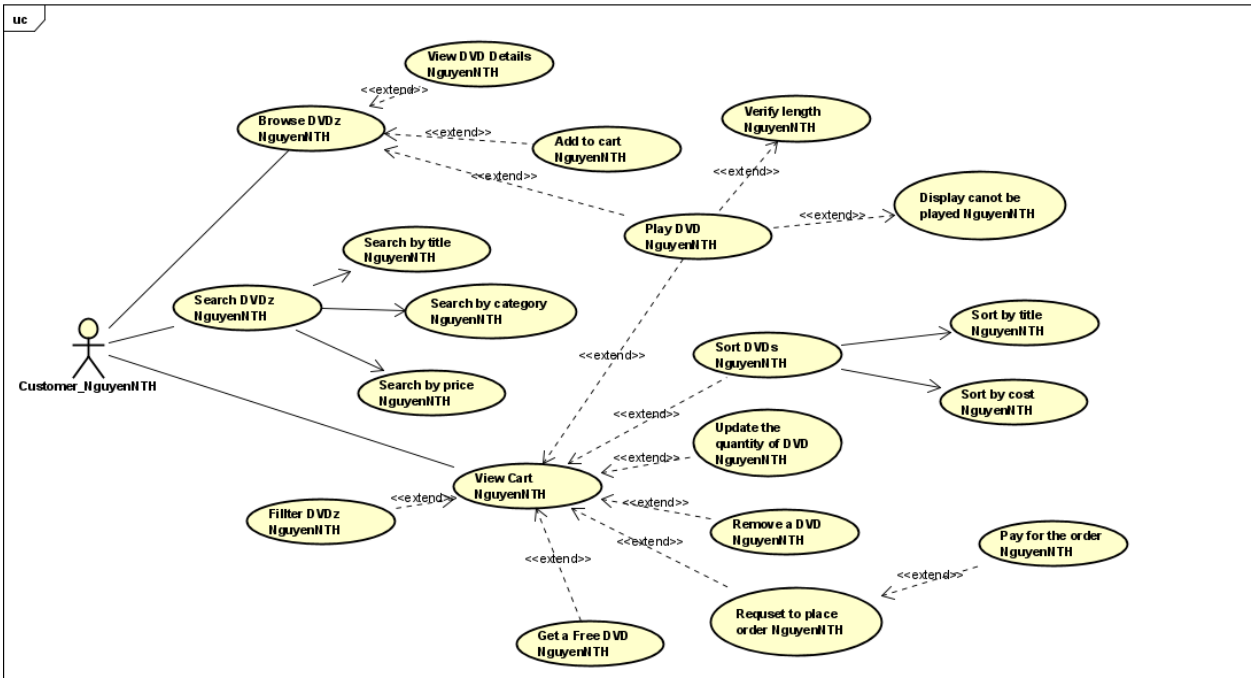


Figure XIII-1.Customer UseCase Diagram

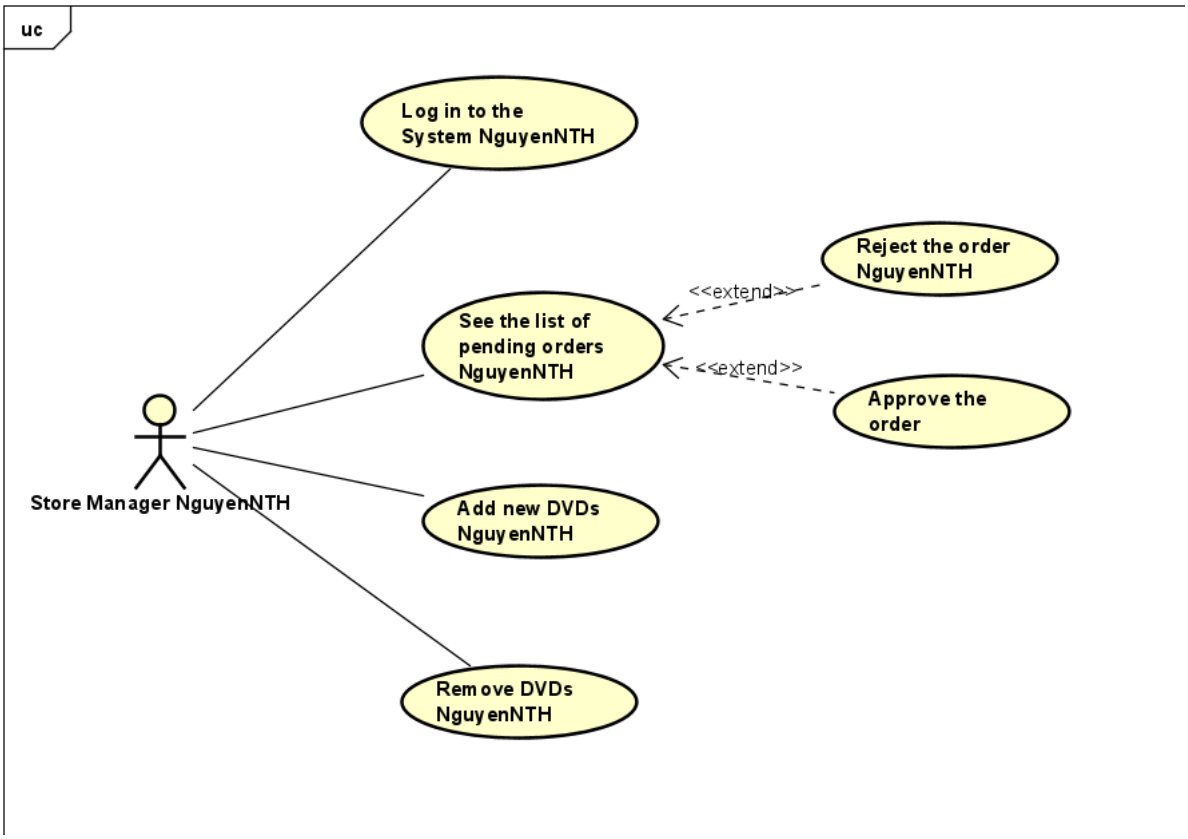


Figure XIII-2.StoreManager UseCase Diagram

XIV. Answer Questions

Trong trường hợp muốn so sánh các đối tượng Media với nhau bằng cách sử dụng Comparable thay vì Comparator, thì thay vì tạo ra các lớp riêng cho từng Comparator, chúng ta cần để lớp Media triển khai interface Comparable.Cách triển khai này giúp chúng ta linh hoạt hơn khi so sánh các đối tượng Media và cung cấp khả năng mở rộng cho các lớp con khác nếu cần thiết.