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

BÁO CÁO THỰC HÀNH
IT3011-744528-2024.1

BÀI THỰC HÀNH 4

Họ và tên sv: Nguyễn Đăng Phúc Hưng
Lớp: **K67 – CNTT Việt Pháp**
GVHD: Lê Thị Hoa
TA: Đặng Mạnh Cường

Hà Nội 12/2024

BÁO CÁO THỰC HÀNH LAB 4 LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

1 Create the Book class 4
2 Creating the abstract Media class 6
3 Creating the CompactDisc class 8
3.1 Create the Disc class extending the Media clas 8
3.2 Create the Track class which models a track on a compact disc and will store information incuding the title and length of the track. 10
3.3 Open the CompactDisc class 11
4 Create the Playable interface 13
5 Update the Cart class to work with Media 14
6 Update the Store class to work with Media 18
7 Constructors of whole classes and parent classes 20
8 Unique item in a list 21
9 Polymorphism with toString() method 22
10 Sort media in the car 23
11 Create a complete console application in the Aims class 24
11.1 Người dùng chọn 1: View store 25
11.1.1 Người dùng tiếp tục chọn 1. See a media's details 26
11.1.2 Người dùng chọn 2: Add a media to the cart 27
11.1.3 Người dùng chọn 3: Play a media 27
11.1.4 Người dùng chọn 4: See current cart 28
11.2 Người dùng chọn 2: Update store 29
11.2.1 Người dùng chọn 1: Add a media to the store 29
11.2.2 Người dùng chọn 2: Remove a media from the store 30
11.3 Người dùng chọn 3: See current cart 31
11.3.1 Người dùng chọn 1: Filter medias in cart 32
11.3.2 Người dùng chọn 2: Sort medias in cart 34
11.3.3 Người dùng chọn 3: Remove media from cart 35
11.3.4 Người dùng chọn 4: Play a media 36
11.3.5 Người dùng chọn 5: Place order 36

12 Class Diagram 38
13 UseCase Diagram 39
14 Answer Questions 39

Table of Figures

Figure 1.1: Book Class 1 4
Figure 1.2: Book Class 2 5
Figure 2.1: Media Class 1 6
Figure 2.2: Media Class 2 7
Figure 3.1: Disc Class 8
Figure 3.2: DigitalVideoDisc Class 9
Figure 3.3: CompactDisc Class 9
Figure 3.4: Track Class 10
Figure 3.5: CompactDisc Class 1 11
Figure 3.6: CompactDisc Class 2 12
Figure 4.1: Playable interface 13
Figure 4.2: Method play() của DigitalVideoDisc 13
Figure 4.3: Method play() của Track 13
Figure 4.4: Method play() của CompactDisc 13
Figure 5.1: Cart Class 1 14
Figure 5.2: Cart Class 2 15
Figure 5.3: Cart Class 3 16
Figure 5.4: Cart Class 4 17
Figure 6.1: Store Class 1 18
Figure 6.2: Store Class 2 19
Figure 7.1: Constructor Track Class 20
Figure 7.2: Constructor CompactDisc Class 20
Figure 7.3: Constructor Media Class 20
Figure 7.4: Constructor Disc Class 21
Figure 8.1: Override equals in Media Class 21
Figure 8.2: Override equals in Track Class 21
Figure 9.1: Code mô phỏng Polymorphism 22
Figure 9.2: Override toString() in Media Class 22
Figure 9.3: Result demo Polymorphism 22
Figure 10.1: Add the comparators as attributes of the Media class 23
Figure 10.2: MediaComparatorByCostTitle Class 23
Figure 10.3: MediaComparatorByTitleCost Class 24
Figure 11.1: Màn hình chính 24
Figure 11.2: Vào Trang View Store 25
Figure 11.3: See a media's details 26
Figure 11.4: Thêm vào Cart 26
Figure 11.5: Thêm media vào Cart 27

Figure 11.6: Play a media 27
Figure 11.7: See current cart after sort 28
Figure 11.8: Vào Trang Update Store 29
Figure 11.9: Add a media to store 29
Figure 11.10: Result after add media to store 30
Figure 11.11: Remove a media from the store 30
Figure 11.12: Result after remove a media 31
Figure 11.13: Vào trang See current cart 31
Figure 11.14: Media in Cart 32
Figure 11.15: Filter Cart By id 32
Figure 11.16: Filter Cart By Title 33
Figure 11.17: Sort Cart By Title 34
Figure 11.18: Sort Cart By Cost 34
Figure 11.19: Remove media by id 35
Figure 11.20: Result after remove media in cart by id 35
Figure 11.21: Play a media in cart 36
Figure 11.22: Order 36
Figure 11.23: Result after order 37
Figure 12.1: Class Diagram 38
Figure 13.1: UseCase Diagram 39
Figure 14.1: Triển khai Comparable trong lớp trừu tượng Media 40
Figure 14.2: Mở rộng để so sánh nhiều thuộc tính hơn 40
Figure 14.3: Triển khai tại lớp con 40

1. Create the Book class

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

import java.util.*;

public class Book extends Media {
    private ArrayList<String> authors = new ArrayList<String>();

    public ArrayList<String> getAuthors() {
        return authors;
    }

    public void setAuthors(ArrayList<String> authors) {
        this.authors = authors;
    }

    public Book() {
    }

    public Book(int id, String title, String category, float cost, ArrayList<String> authors) {
        super(id, title, category, cost);
        this.authors = authors;
    }

    public void addAuthor(String authorName) {
        if (authors.contains(authorName)) {
            System.out.println("Author " + authorName + " already exists in the authors list.");
        } else {
            authors.add(authorName);
            System.out.println("Author " + authorName + " successfully added.");
        }
    }

    public void removeAuthor(String authorName) {
        if (authors.contains(authorName)) {
            authors.remove(authorName);
            System.out.println("Author " + authorName + " removed.");
        } else {
            System.out.println("Author " + authorName + " not found.");
        }
    }

    public String authorToString() {
        String author = String.join(", ", this.getAuthors());
        return author;
    }

    public String toString() {
        return "Book - title: " + this.getTitle() +
                " - category: " + this.getCategory() +
                " - authors: " + this.authorToString() +
                " - cost: " + this.getCost() + "$";
    }
}
```

Hình 1: Book class

2. Create the abstract Media class

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

import java.util.Comparator;

public abstract class Media {
    private int id;
    private String title;
    private String category;
    private float cost;

    public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost();
    public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle();

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getTitle() {
        return title;
    }

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

    public String getCategory() {
        return category;
    }

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

    public float getCost() {
        return cost;
    }

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

    public Media() {
    }

    public Media(int id, String title, String category, float cost) {
        super();
        this.id = id;
        this.title = title;
        this.category = category;
        this.cost = cost;
    }

    public String toString() {
        return "Media - id: " + this.getId() +
                " - title: " + this.getTitle() +
                " - category: " + this.getCategory() +
                " - cost: " + this.getCost() + "$";
    }

    public boolean isMatch(String title) {
        if (title == null || this.getTitle() == null || title.trim().isEmpty())
            return false;

        String[] words = title.toLowerCase().split("\\s+");
        String loweredThisTitle = this.getTitle().toLowerCase();

        for (String word : words) {
            if (loweredThisTitle.contains(word)) return true;
        }

        return false;
    }
```

Hình 2: Abstract Media class

3. Create the CompactDisc class

3.1. Create the Disc class extending the Media class

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

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

    public int getLength() {
        return length;
    }

    public String getDirector() {
        return director;
    }

    public Disc(int id, String title, String category, float cost, int length, String director) {
        super(id, title, category, cost);
        this.length = length;
        this.director = director;
    }

    public Disc(int id, String title, String category, float cost, String director) {
        super(id, title, category, cost);
        this.director = director;
    }

    public Disc() {
    }

}
```

Hình 3: Disc class extends Media class

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

public class DigitalVideoDisc extends Disc implements Playable {
    private static int nbDigitalVideoDiscs = 0;

    public DigitalVideoDisc(int id, String title, String category, float cost, int length, String director)
        super(id, title, category, cost, length, director);
        nbDigitalVideoDiscs++;
    }

    public void play() {
        System.out.println("Playing DVD: " + this.getTitle());
        System.out.println("DVD length: " + this.getLength());
    }

    public String toString() {

        return "DVD - title: " + this.getTitle() +
                " - category: " + this.getCategory() +
                " - director: " + this.getDirector() +
                " - length: " + this.getLength() +
                " - cost: " + this.getCost() + "$";
    }
}
```

Hình 4: DigitalVideoDisc class extends Disc class

```java
package hust.soict.dsai.aims.media;
import java.util.*;

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

    public String getArtist() {
        return artist;
    }

    public CompactDisc() {
    }
```

Hình 5: CompactDisc class extends Disc class

3.2. Create the Track class

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

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

    public String getTitle() {
        return title;
    }

    public int getLength() {
        return length;
    }

    public Track() {
    }

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

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

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;

        if (o == null || getClass() != o.getClass()) return false;

        Track track = (Track) o;
        return this.title.equals(track.title) && this.length == track.length;
    }

}
```

Hình 6: Track clas

3.3. Open the CompactDisc class

```java
package hust.soict.dsai.aims.media;
import java.util.*;

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

    public String getArtist() {
        return artist;
    }

    public CompactDisc() {
    }

    public CompactDisc(int id, String title, String category, float cost, String artist,
            ArrayList<Track> tracks, int length, String director) {
        super(id, title, category, cost, length, director);
        this.artist = artist;
        this.tracks = tracks;
    }

    public CompactDisc(int id, String title, String category, float cost, String artist,
            ArrayList<Track> tracks, String director) {
        super(id, title, category, cost, director);
        this.artist = artist;
        this.tracks = tracks;
    }

    public void addTrack(Track track) {
        boolean isIdentical = false;

        for (Track track : tracks) {
            if (track.equals(track)) {
                isIdentical = true;
                break;
            }
        }

        if (isIdentical) {
            System.out.println("Track \"" + track + "\" already exists in the tracks list.");
        } else {
            tracks.add(track);
            System.out.println("Track \"" + track + "\" successfully added.");
        }
    }

    public void removeTrack(Track track) {
        if (tracks.contains(track)) {
            tracks.remove(track);
            System.out.println("Track \"" + track + "\" removed.");
        } else {
            System.out.println("Track \"" + track + "\" not found.");
        }
    }
```

Hình 7: CompactDisc class (1)

```
    public int getLength() {
        int sumLength = 0;
        for (Track track : tracks) {
            sumLength += track.getLength();
        }

        return sumLength;
    }

    public void play() {
        System.out.println("Playing CD: " + this.getTitle());
        System.out.println("CD artist: " + this.getArtist());
        System.out.println("CD length: " + this.getLength());
        for (Track track : tracks) {
            track.play();
        }
    }

    public String toString() {
        return "CD - title: " + this.getTitle() +
                " - category: " + this.getCategory() +
                " - director: " + this.getDirector() +
                " - length: " + this.getLength() +
                " - cost: " + this.getCost() + "$";
    }
}
```

Hình 8: CompactDisc class (2)

4. Create the Playable interface

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

public interface Playable {
    public void play();
}
```

Hình 9: Giao diện Playable

```
public void play() {
    System.out.println("Playing DVD: " + this.getTitle());
    System.out.println("DVD length: " + this.getLength());
}
```

Hình 10: Phương thức play() của DigitalVideoDisc

```
public void play() {
    System.out.println("Playing CD: " + this.getTitle());
    System.out.println("CD artist: " + this.getArtist());
    System.out.println("CD length: " + this.getLength());
    for (Track track : tracks) {
        track.play();
    }
}
```

Hình 11: Phương thức play() của CompactDisc

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

Hình 12: Phương thức play() của Track

5. Update the Cart class to work with Media class

```java
package hust.soict.dsai.aims.cart;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.Scanner;

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

public class Cart {
    public static final int MAX_NUMBERS_ORDERED = 20;
    private ArrayList<Media> itemsOrdered = new ArrayList<Media>();

    public ArrayList<Media> getItemsOrdered() {
        return itemsOrdered;
    }

    public void sortCart(Comparator<Media> comparator) {
        Collections.sort(this.itemsOrdered, comparator);
    }

    public void addMedia(Media media) {
        if (itemsOrdered.size() >= MAX_NUMBERS_ORDERED) {
            System.out.println("Maximum number of medias reached, unable to add anymore to cart.");
        } else {
            boolean isIdentical = false;
            for (Media media : itemsOrdered) {
                if (media.equals(media)) {
                    isIdentical = true;
                    break;
                }
            }

            if (isIdentical) {
                System.out.println("Media \"" + media.getTitle() + "\" is already added to cart.");
                return;
            }

            itemsOrdered.add(media);
            System.out.println("Media \"" + media.getTitle() + "\" added to cart.");
        }
    }

    public void removeMedia(Media media) {
        if (itemsOrdered.contains(media)) {
            itemsOrdered.remove(media);
            System.out.println("Media \"" + media.getTitle() + "\" removed from cart.");
        } else {
            System.out.println("Media \"" + media.getTitle() + "\" not found in cart.");
        }
    }
```

Hình 13: Cart class (1)

```
    public float totalCost() {
        float sumCost = (float)0.0;
        for (Media media : itemsOrdered) {
            sumCost += media.getCost();
        }

        return sumCost;
    }

    public void printCart() {
        System.out.println("********************** CART **********************");
        System.out.println("Ordered items");
        int i = 1;
        for (Media media : itemsOrdered) {
            System.out.println(i + ". " + media.toString());
            i++;
        }
        System.out.println("Total cost: " + Math.round(this.totalCost() * 100.0) / 100.0 + "$");
        System.out.println("***************************************************");
    }

    public void searchMediabyID(Scanner input) {
        System.out.println("***************************************************");
        System.out.print("Input ID to search: ");
        int inputID = input.nextInt();
        int resultCount = 0;
        for (Media media : itemsOrdered) {
            if (inputID == media.getId()) {
                resultCount++;
                if (resultCount == 1) System.out.println("Media(s) with ID = " + inputID + ":");
                System.out.println(resultCount + ". " + media.toString());
            }
        }

        if (resultCount == 0) System.out.println("No match is found.");
        System.out.println("***************************************************");
    }

    public void searchMediabyTitle(Scanner input) {
        System.out.println("***************************************************");
        System.out.print("Input title to search: ");
        String inputTitle = input.nextLine();
        int resultCount = 0;

        for (Media media : itemsOrdered) {
            if (media.isMatch(inputTitle)) {
                resultCount++;
                if (resultCount == 1) System.out.println("Media(s) with title that matches \"" + inputTitle + "\":");
                System.out.println(resultCount + ". " + media.toString());
            }
        }

        if (resultCount == 0) System.out.println("No match is found.");
        System.out.println("***************************************************");
    }
}
```

Hình 14: Cart class (2)

6. Update the Store class to work with Media class

```java
package hust.soict.dsai.aims.store;
import java.util.ArrayList;

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

public class Store {
    private ArrayList<Media> itemsInStore;

    public ArrayList<Media> getItemsInStore() {
        return itemsInStore;
    }

    public Store() {
        itemsInStore = new ArrayList<Media>();
    }

    public void addMedia(Media media) {
        itemsInStore.add(media);
        System.out.println("Media \"" + media.getTitle() + "\" added to store.");
    }

    public void removeMedia(Media media) {
        itemsInStore.remove(media);
        System.out.println("Media \"" + media.getTitle() + "\" removed from store.");
    }
}
```

Hình 15: Store class

7. Constructors of whole classes and parent classes

```java
public CompactDisc() {
}

public CompactDisc(int id, String title, String category, float cost, String artist,
        ArrayList<Track> tracks, int length, String director) {
    super(id, title, category, cost, length, director);
    this.artist = artist;
    this.tracks = tracks;
}

public CompactDisc(int id, String title, String category, float cost, String artist,
        ArrayList<Track> tracks, String director) {
    super(id, title, category, cost, director);
    this.artist = artist;
    this.tracks = tracks;
}
```

Hình 16: Constructor của CompactDisc class

```java
public Track() {
}

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

Hình 17: Constructor của Track class

Lớp Disc kế thừa từ lớp Media. Constructor của lớp Media và lớp Disc lần lượt như sau:

```java
public Media() {
}

public Media(int id, String title, String category, float cost) {
    super();
    this.id = id;
    this.title = title;
    this.category = category;
    this.cost = cost;
}
```

Hình 18: Constructor của Media class

```java
public Disc(int id, String title, String category, float cost, int length, String director) {
    super(id, title, category, cost);
    this.length = length;
    this.director = director;
}

public Disc(int id, String title, String category, float cost, String director) {
    super(id, title, category, cost);
    this.director = director;
}
```

Hình 19: Constructor của Disc class

8. Unique item in a list

```java
@Override
public boolean equals(Object o) {
    if (this == o) return true;

    if (o == null || getClass() != o.getClass()) return false;

    Media media = (Media) o;
    return this.title.equals(media.title);
}
```

Hình 20: Override phương thức equals() của Media class

```java
@Override
public boolean equals(Object o) {
    if (this == o) return true;

    if (o == null || getClass() != o.getClass()) return false;

    Track track = (Track) o;
    return this.title.equals(track.title) && this.length == track.length;
}
```

Hình 21: Override phương thức equals() của Track class

9. Polymorphism with toString() method

```java
@Override
public boolean equals(Object o) {
    if (this == o) return true;

    if (o == null || getClass() != o.getClass()) return false;

    Media media = (Media) o;
    return this.title.equals(media.title);
}
```

Hình 22: Override phương thức toString() trong Media class

```java
public String authorToString() {
    String author = String.join(", ", this.getAuthors());
    return author;
}

public String toString() {
    return "Book - title: " + this.getTitle() +
            " - category: " + this.getCategory() +
            " - authors: " + this.authorToString() +
            " - cost: " + this.getCost() + "$";
}
```

Hình 23: Override phương thức toString() trong Book class

```java
public String toString() {

    return "DVD - title: " + this.getTitle() +
            " - category: " + this.getCategory() +
            " - director: " + this.getDirector() +
            " - length: " + this.getLength() +
            " - cost: " + this.getCost() + "$";
}
```

Hình 24: Override phương thức toString() trong DigitalVideoDisc class

```java
public String toString() {
    return "Book - title: " + this.getTitle() +
            " - category: " + this.getCategory() +
            " - authors: " + this.authorToString() +
            " - cost: " + this.getCost() + "$";
}
```

Hình 25: Override phương thức toString() trong CompactDisc class

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

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

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 hust.soict.dsai.aims.media.Track;

public class TestPolymorphism {

    Run | Debug
    public static void main(String[] args) {

        List<Media> media = new ArrayList<Media>();

        CompactDisc cd = new CompactDisc(1, "CDx", "Hardcore", 10.55f, "Avicii", new ArrayList<Track>(), 115, "Dylan Baubach");
        DigitalVideoDisc dvd = new DigitalVideoDisc(2, "The Lion King", "Animation", 19.95f, 87, "Roger Allers");
        Book book = new Book(3, "War and Peace", "Novel", 21.75f, new ArrayList<>(Arrays.asList("Lev Tolstoy")));

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

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

Hình 26: Code chương trình mô phỏng polymorphism

Kết quả:

```
CD - title: CDx - category: Hardcore - director: Dylan Baubach - length: 0 - cost: 10.55$
DVD - title: The Lion King - category: Animation - director: Roger Allers - length: 87 - cost: 19.95$
Book - title: War and Peace - category: Novel - authors: Lev Tolstoy - cost: 21.75$
```

Hình 27: Kết quả chạy chương trình polymorphism

The Media class is the base class that is inherited by the subclasses CompactDisc, DigitalVideoDisc, and Book. When we instantiate objects such as cd, dvd, and book from these subclasses and assign them to a variable of type Media, we apply the upcasting technique.
Adding these objects to the media list and iterating through the list to print the information of each element using the toString() method is a typical example of dynamic polymorphism. Each subclass can implement its own version of the toString() method, so the output will vary depending on the type of the object without explicitly specifying the type of each element in the code.

10. Sort medias in the cart

```java
public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost();
public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle();
```

Hình 28: Thêm các đối tượng Comparator là thuộc tính của Media class

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

import java.util.Comparator;

public class MediaComparatorByCostTitle implements Comparator<Media> {

    @Override
    public int compare(Media o1, Media o2) {
        return Comparator.comparing(Media::getCost)
                .thenComparing(Media::getTitle)
                .compare(o1, o2);
    }
}
```

Hình 29: MediaComparatorByCostTitle class

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

import java.util.Comparator;

public class MediaComparatorByTitleCost implements Comparator<Media> {

    @Override
    public int compare(Media o1, Media o2) {
        return Comparator.comparing(Media::getTitle)
                .thenComparing(Media::getCost)
                .compare(o1, o2);
    }
}
```

Hình 30: MediaComparatorByTitleCost class

11. Console application

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

Hình 31: Màn hình chính

11.1. Người dùng chọn View store

```
AIMS:
--------------------------------
    1. View store
    2. Update store
    3. See current cart
    0. Exit
--------------------------------
Please choose a number: 0-1-2-3
Enter option: 1
*************** LIST OF ITEMS IN STORE ***************
CD - title: Thriller - category: Pop - director: Quincy Jones - length: 810 - cost: 20.95$
DVD - title: Finding Nemo - category: Animation - director: Adrew Stanton - length: 119 - cost: 25.49$
Book - title: To Kill a Mockingbird - category: Novel - authors: Harper Lee - cost: 49.75$
Book - title: Don Quixote - category: Novel - authors: Miguel de Cervantes - cost: 37.49$
CD - title: Loin des yeux - category: Acoustic music - director: Sylvain Duthu - length: 540 - cost: 29.99$
DVD - title: The Dark Knight - category: Action film - director: Christopher Nolan - length: 134 - cost: 21.35$
**************************************************
================================
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
Enter option:
```

Hình 32: Màn hình View store

11.1.1. Người dùng tiếp tục chọn See a media's details

```
================================
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
Enter option: 1
Enter media's title (ignore case): don quixote
Book - title: Don Quixote - category: Novel - authors: Miguel de Cervantes - cost: 37.49$
================================
Options:
--------------------------------
    1. Add to cart
    0. Back
--------------------------------
Please choose a number: 0-1
Enter option:
```

Hình 33: Màn hình See a media's details

```
Book - title: Don Quixote - category: Novel - authors: Miguel de Cervantes - cost: 37.49$
================================
Options:
--------------------------------
    1. Add to cart
    0. Back
--------------------------------
Please choose a number: 0-1
Enter option: 1
Media "Don Quixote" added to cart.
```

Hình 34: Màn hình Add to cart từ View store

11.1.2. Người dùng chọn Add a media to cart

```
********** LIST OF ITEMS IN STORE **********
CD - title: Thriller - category: Pop - director: Quincy Jones - length: 810 - cost: 20.95$
DVD - title: Finding Nemo - category: Animation - director: Adrew Stanton - length: 119 - cost: 25.49$
Book - title: To Kill a Mockingbird - category: Novel - authors: Harper Lee - cost: 49.75$
Book - title: Don Quixote - category: Novel - authors: Miguel de Cervantes - cost: 37.49$
CD - title: Loin des yeux - category: Acoustic music - director: Sylvain Duthu - length: 540 - cost: 29.99$
DVD - title: The Dark Knight - category: Action film - director: Christopher Nolan - length: 134 - cost: 21.35$
**************************************************
================================
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
Enter option: 2
Enter media's title (ignore case): finding nemo
Media "Finding Nemo" added to cart.
Number of DVD(s) in cart: 1
```

Hình 35: Màn hình Add a media to cart

11.1.3. Người dùng chọn Play a media

```
================================
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
Enter option: 3
Enter media's title (ignore case): loin des yeux
Playing CD: Loin des yeux
CD artist: Boulevard des Airs
CD length: 540
Playing Track: Et nous vraiment
Track length: 195
Playing Track: Cielo Ciego
Track length: 166
Playing Track: Bruxelles
Track length: 179
```

Hình 36: Màn hình Play a media từ View store

11.1.4. Người dùng chọn See current 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
Enter option: 4
********************* CART *********************
Ordered items
1. Book - title: Don Quixote - category: Novel - authors: Miguel de Cervantes - cost: 37.49$
2. DVD - title: Finding Nemo - category: Animation - director: Adrew Stanton - length: 119 - cost: 25.49$
Total cost: 62.98$
************************************************
==============================
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
Enter option:
```

Hình 37: Màn hình See current cart

- Người dùng chọn Filter medias in cart:

```
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
Enter option: 1
Choose a filter criteria (1 = by id; 2 = by Title): 1
************************************************
Input ID to search: 4
Media(s) with ID = 4:
1. Book - title: Don Quixote - category: Novel - authors: Miguel de Cervantes - cost: 37.49$
************************************************
```

Hình 38: Màn hình Filter medias in cart bằng id

```
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
Enter option: 1
Choose a filter criteria (1 = by id; 2 = by Title): 2
************************************************
Input title to search: finding
Media(s) with title that matches "finding":
1. DVD - title: Finding Nemo - category: Animation - director: Adrew Stanton - length: 119 - cost: 25.49$
************************************************
```

Hình 39: Màn hình Filter medias in cart bằng title

- Người dùng chọn Sort medias in cart:

```
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
Enter option: 2
Choose a sort criteria (1 = by Title; 2 = by Cost): 1
Cart sorted by title.
********************** CART **********************
Ordered items
1. Book - title: Don Quixote - category: Novel - authors: Miguel de Cervantes - cost: 37.49$
2. DVD - title: Finding Nemo - category: Animation - director: Adrew Stanton - length: 119 - cost: 25.49$
Total cost: 62.98$
************************************************
```

Hình 40: Màn hình Sort medias in cart bằng title

```
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
Enter option: 2
Choose a sort criteria (1 = by Title; 2 = by Cost): 2
Cart sorted by cost.
********************** CART **********************
Ordered items
1. DVD - title: Finding Nemo - category: Animation - director: Adrew Stanton - length: 119 - cost: 25.49$
2. Book - title: Don Quixote - category: Novel - authors: Miguel de Cervantes - cost: 37.49$
Total cost: 62.98$
************************************************
```

Hình 41: Màn hình Sort medias in cart bằng cost

- Người dùng chọn Remove media from cart:

```
================================
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
Enter option: 3
Enter title of media to remove (ignore case): don quixote
Media "Don Quixote" removed from cart.
```

Hình 42: Màn hình Remove media from cart

Kết quả:

```
********************* CART *********************
Ordered items
1. DVD - title: Finding Nemo - category: Animation - director: Adrew Stanton - length: 119 - cost: 25.49$
Total cost: 25.49$
************************************************
```

Hình 43: Kết quả sau khi xóa media khỏi cart

- Người dùng chọn Play a media:

```
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
Enter option: 4
Enter media's title (ignore case): finding nemo
Playing DVD: Finding Nemo
DVD length: 119
```

Hình 44: Màn hình Play a media từ cart

- Người dùng chọn Place order:

```
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
Enter option: 5
Your order is created.
Cart is now emptied.
```

Hình 45: Màn hình Place order

Mọi item trong cart đã bị xóa:

```
********************** CART **********************
Ordered items
Total cost: 0.0$
**************************************************
```

Hình 46: Mọi media trong cart đã bị xóa sau khi người dùng chọn Place order

11.2. Người dùng chọn Update store

```
AIMS:
--------------------------------
    1. View store
    2. Update store
    3. See current cart
    0. Exit
--------------------------------
Please choose a number: 0-1-2-3
Enter option: 2
================================
Update store options:
--------------------------------
    1. Add a media to store
    2. Remove a media from store
    0. Back
--------------------------------
Please choose a number: 0-1-2
Enter option:
```

Hình 47: Màn hình Update store

11.2.1. Người dùng chọn Add a media to store

```
Update store options:
------------------------------
    1. Add a media to store
    2. Remove a media from store
    0. Back
------------------------------
Please choose a number: 0-1-2
Enter option: 1
Choose a type of media to add (1 = CD; 2 = DVD; 3 = Book): 1
Media id: 7
CD title: Myles Smith's Collect - Best songs
CD category: Pop/Acoustic
CD cost: 36.55
CD artist: Myles Smith
CD director: Myles Smith
CD track(s) (each track separated by ';' | title and length of each track separated by '/'): Stargazing/180; Solo/198;My Home/212;Nice to Meet You/207
Media "Myles Smith's Collect - Best songs" added to store.
```

Hình 48: Màn hình Add a media to store

Kết quả:

```
*************** LIST OF ITEMS IN STORE ***************
CD - title: Thriller - category: Pop - director: Quincy Jones - length: 810 - cost: 20.95$
DVD - title: Finding Nemo - category: Animation - director: Adrew Stanton - length: 119 - cost: 25.49$
Book - title: To Kill a Mockingbird - category: Novel - authors: Harper Lee - cost: 49.75$
Book - title: Don Quixote - category: Novel - authors: Miguel de Cervantes - cost: 37.49$
CD - title: Loin des yeux - category: Acoustic music - director: Sylvain Duthu - length: 540 - cost: 29.99$
DVD - title: The Dark Knight - category: Action film - director: Christopher Nolan - length: 134 - cost: 21.35$
CD - title: Myles Smith's Collect - Best songs - category: Pop/Acoustic - director: Myles Smith - length: 797 - cost: 36.55$
*****************************************************
```

Hình 49: Kết quả sau khi thêm vào store

11.2.2. Người dùng chọn Remove a media from store

```
Update store options:
------------------------------
    1. Add a media to store
    2. Remove a media from store
    0. Back
------------------------------
Please choose a number: 0-1-2
Enter option: 2
Enter id of media to remove: 4
Media "Don Quixote" removed from store.
```

Hình 50: Màn hình Remove a media from store

Kết quả:

```
*************** LIST OF ITEMS IN STORE ***************
CD - title: Thriller - category: Pop - director: Quincy Jones - length: 810 - cost: 20.95$
DVD - title: Finding Nemo - category: Animation - director: Adrew Stanton - length: 119 - cost: 25.49$
Book - title: To Kill a Mockingbird - category: Novel - authors: Harper Lee - cost: 49.75$
CD - title: Loin des yeux - category: Acoustic music - director: Sylvain Duthu - length: 540 - cost: 29.99$
DVD - title: The Dark Knight - category: Action film - director: Christopher Nolan - length: 134 - cost: 21.35$
CD - title: Myles Smith's Collect - Best songs - category: Pop/Acoustic - director: Myles Smith - length: 797 - cost: 36.55$
*****************************************************
```

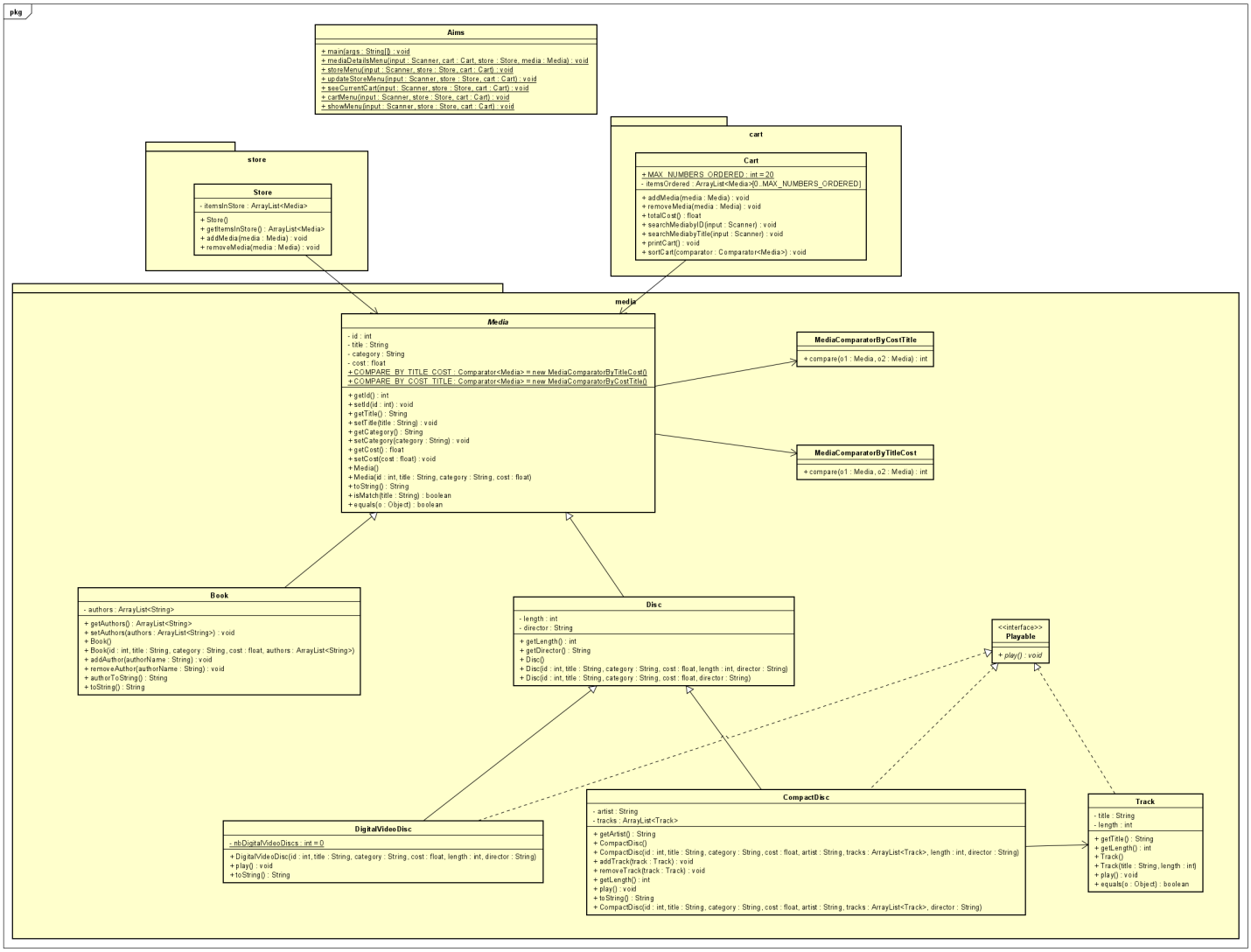
Hình 51: Kết quả sau khi xóa khỏi store

11.3. Người dùng chọn See current cart

Tương tự chức năng See current cart ở mục 11.1.4

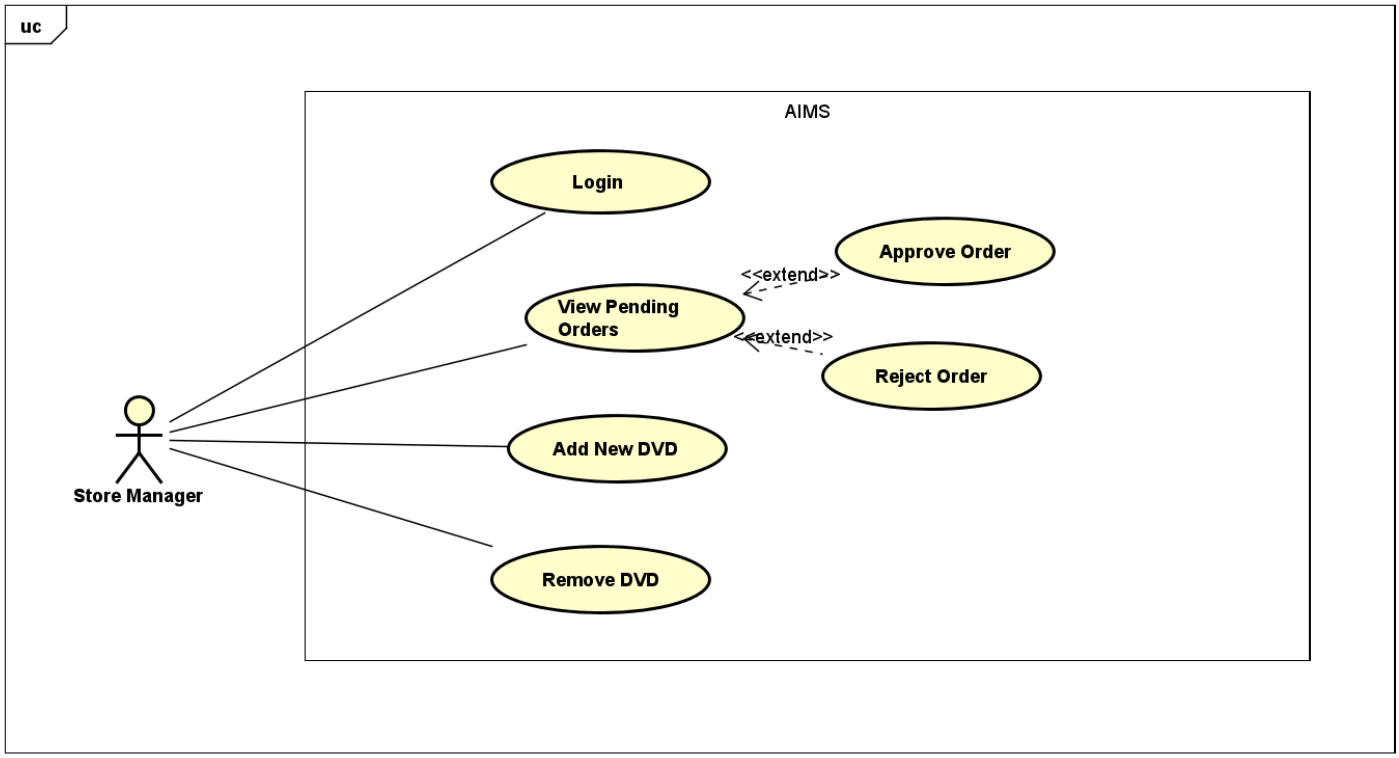
12. Class diagram and UseCase diagram

12.1. Class diagram

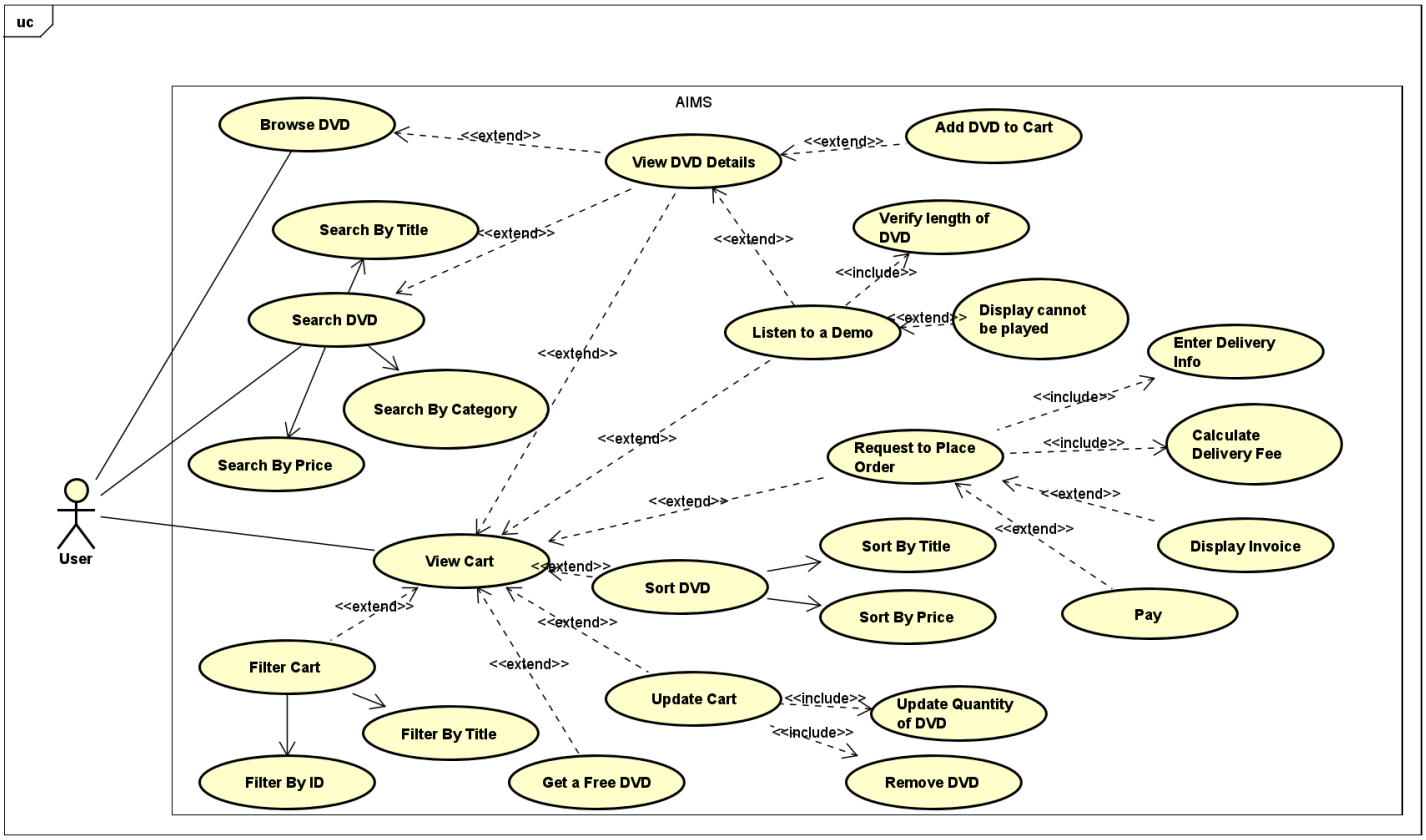


Hình 52: Class diagram

12.2. UseCase diagram



Hình 53: UseCase diagram của Store Manager



Hình 54: UseCase diagram của User

13. Answer questions

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.
1. What class should implement the Comparable interface?
2. In those classes, how should you implement the compareTo() method be to reflect the ordering that we want?
3. 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?
4. 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?

Answer:

1. It is recommended to implement the Comparable interface for the Media class and/or its subclasses (Book, Disc, DVD, CompactDisc), depending on which classes the comparison is meant for.
2. Example with SortByTitleThenCost: In the Media.java file:

```
public class Media implements Comparable<Media> {
   @Override
   public int compareTo(Media other) {
      int titleComparison = this.getTitle().compareTo(other.getTitle());
      if (titleComparison != 0) {
         return titleComparison;
      } else {
         return Double.compare(this.getCost(), other.getCost());
      }
   }
}
```

3. It is not possible to write two overloaded methods for two sorting rules at the same time in a single class when using the Comparable interface. However, it is possible to add parameters to the method for this purpose.
4. The compareTo method can be overridden as follows:

```
public class DVD extends Disc implements Playable {
   @Override
   public int compareTo(Media other) {
      if (other instanceof DVD) {
         DVD otherDVD = (DVD) 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 Double.compare(this.getCost(), otherDVD.getCost());
            }
         }
      } else {
         return super.compareTo(other);
      }
   }
}
```