

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



BÁO CÁO THỰC HÀNH
IT3103-744529-2024.1
BÀI THỰC HÀNH 4

Họ và tên sinh viên: Đoàn Thanh Tùng
MSSV: 20225946
Lớp: Việt Nhật 02 – K67
GVHD: Lê Thị Hoa
HTGD: Bùi Trọng Dũng

Hà Nội 11/2024

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 including 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 6.1: Store Class 1 | 18 |
| 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 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

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

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

public class Book {
    private int id;
    private String title;
    private String category;
    private float cost;
    private List<String> authors = new ArrayList<>();

    • public Book() {
        // TODO Auto-generated constructor stub
    }

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

    }

    • public Book(int id, String title, String category, float cost, List<String> authors) {

    }

    • public String getTitle() {
        return title;
    }

    • public String getCategory() {
        return category;
    }
}
```

Figure 1.1: Book Class 1

```
public String getCategory() {  
    return category;  
}  
  
public float getCost() {  
    return cost;  
}  
  
public List<String> getAuthors() {  
    return authors;  
}  
  
public void addAuthor(String authorName) {  
    if (!authors.contains(authorName)) {  
        authors.add(authorName);  
        System.out.println("Doan Thanh Tung || " + authorName + "is added successfully");  
    } else {  
        System.out.println("Doan Thanh Tung || " + authorName + "is already in the list");  
    }  
}  
  
public void removeAuthor(String authorName) {  
    if (authors.contains(authorName)) {  
        authors.remove(authorName);  
        System.out.println("Doan Thanh Tung || " + authorName + "is removed successfully");  
    } else {  
        System.out.println("Doan Thanh Tung || " + authorName + "is not in the list");  
    }  
}  
}
```

Figure 1.2: Book Class 2

2 Creating the abstract Media class

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

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

//Doan Thanh Tung - 20225946
public abstract class Media {
    int id;
    String title;
    String category;
    float cost;

    public Media() {

    }

    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;
    }
}
```

Figure 2.1: Media Class 1


```
//Doan Thanh Tung - 20225946
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;
}
}
```

Figure 2.2: Media Class 2

3 Creating the CompactDisc class

3.1 Create the Disc class extending the Media clas

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

//Doan Thanh Tung - 20225946
public class Disc extends Media {
    private int length;
    private String director;

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

    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 void setLength(int length) {
        this.length = length;
    }

    public void setDirector(String director) {
        this.director = director;
    }
}
```

Figure 3.1: Disc Class

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

// Doan Thanh Tung - 20225946
public class DigitalVideoDisc extends Disc {
    private String director;
    private int length;

    public DigitalVideoDisc(int id, String title, String category, float cost) {
        super(id, title, category, cost);
    }

    public String getDirector() {
        return director;
    }

    public int getLength() {
        return length;
    }
}
```

Figure 3.2: DigitalVideoDisc Class

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

//Doan Thanh Tung - 20225946
public class CompactDisc extends Disc {
    private String artist;
    private ArrayList<Track> tracks = new ArrayList<Track>();
    public CompactDisc(int id, String title, String category, float cost) {
        super(id, title, category, cost);
    }
}
```

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

//Doan Thanh Tung - 20225946
public class CompactDisc extends Disc {

}
}
```

Figure 3.3: CompactDisc Class

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

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

//Doan Thanh Tung - 20225946
public class Track {
    private int length;
    private String title;
    public Track() {

    }
    public void setLength(int length) {
        this.length = length;
    }
    public void setTitle(String title) {
        this.title = title;
    }
}
```

Figure 3.4: Track Class

3.3 Open the CompactDisc class

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

//Doan Thanh Tung - 20225946

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

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

    public String getArtist() {
        return artist;
    }

    public void addTrack(Track track) {
        if(!tracks.contains(track)) {
            tracks.add(track);
            System.out.println("Doan Thanh Tung || " + track + "is added successfully");
        }
        System.out.println(track+" is already in the track list");
    }
}
```

Figure 3.5: CompactDisc Class 1

```
public void addTrack(Track track) {
    if(!tracks.contains(track)) {
        tracks.add(track);
        System.out.println("Doan Thanh Tung || " + track + "is added successfully");
    }
    System.out.println(track+" is already in the track list");
}

public void removeTrack(Track track) {
    if (tracks.contains(track)) {
        tracks.remove(track);
        System.out.println("Doan Thanh Tung || " + track + "is removed successfully");
    } else {
        System.out.println("Doan Thanh Tung || " + track + "is not in the track list");
    }
}

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

Figure 3.6: CompactDisc Class 2

4 Create the Playable interface

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

//Doan Thanh Tung - 20225946
public interface Playable {

    public void play();

}
```

Figure 4.1: Playable interface

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

```
@Override
public void play() {
    System.out.println("DOAN THANH TUNG || Playing DVD: " + this.getTitle());
    System.out.println("DOAN THANH TUNG || DVD length: " + this.getLength());
}
```

Figure 4.2: Method play() của DigitalVideoDisc

```
@Override
public void play() {
    System.out.println("DOAN THANH TUNG || Playing DVD: " + this.getTitle());
    System.out.println("DOAN THANH TUNG || DVD length: " + this.getLength());
}
```

Figure 4.3: Method play() của Track

```
@Override
public void play() {
    System.out.println("DOAN THANH TUNG || Playing CD: " + this.getTitle());
    System.out.println("DOAN THANH TUNG || CD length: " + this.getLength());
    for (Track track : tracks) {
        track.play();
    }
}
```

Figure 4.4: Method play() của CompactDisc

5 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.

```
import java.util.ArrayList;[]

// Doan Thanh Tung - 20225946
public class Cart {
    public static final int MAX_NUMBERS_ORDERED = 20;
    private List<Media> itemsOrdered = new ArrayList<Media>();
    private int qtyOrdered;

    public void addMedia(Media media) {
        if (qtyOrdered < MAX_NUMBERS_ORDERED) {
            itemsOrdered.add(media);
            System.out.println(media.getTitle() + " is added successfully (DOAN THANH TUNG)");
            qtyOrdered++;
        } else
            System.out.println("The cart is full");
    }

    public void removeMedia(Media media) {
        if (itemsOrdered.size() == 0) {
            System.out.println("nothing in the cart!");
        }
        for (Media item : itemsOrdered) {
            if (item.equals(media)) {
                itemsOrdered.remove(item);
                System.out.println(media.getTitle() + " is removed successfully");
                qtyOrdered--;
                break;
            }
        }
        System.out.println("there is no DVD like that in your cart!");
    }
}
```

Figure 5.1: Cart Class 1


```
public float totalcost() {
    float sum = 0;
    for (int i = 0; i < qtyOrdered; i++) {
        sum += itemsOrdered.get(i).getCost();
    }
    return sum;
}

// Doan Thanh Tung - 20225946
public void print() {
    System.out.println("\n*****CART*****");
    System.out.println("Ordered Items:");
    for (int i = 0; i < qtyOrdered; i++) {
        System.out.println(itemsOrdered.get(i).toString());
    }
    System.out.printf("Total cost: $%.2f\n", totalcost());
    System.out.println("*****");
}

// Doan Thanh Tung - 20225946
public void search(int id) {
    for (int i = 0; i < qtyOrdered; i++) {
        if (itemsOrdered.get(i).getId() == id) {
            System.out.println("\nSuccessfull");
            System.out.println(itemsOrdered.get(i).toString());
            return;
        }
    }
    System.out.println("\nNot Found!");
}
```

Figure 5.2: Cart Class 2

```
// Doan Thanh Tung - 20225946
public void search(int id) {
    for (int i = 0; i < qtyOrdered; i++) {
        if (itemsOrdered.get(i).getId() == id) {
            System.out.println("\nSuccessfull");
            System.out.println(itemsOrdered.get(i).toString());
            return;
        }
    }
    System.out.println("\nNot Found!");
}

// Doan Thanh Tung - 20225946

public void search(String title) {
    for (int i = 0; i < qtyOrdered; i++) {
        if (itemsOrdered.get(i).getTitle() == title) {
            System.out.println("\nSuccessfull");
            System.out.println(itemsOrdered.get(i).toString());
            return;
        }
    }
    System.out.println("\nNot Found!");
}
}
```

Figure 5.3: Cart Class 3

6 Update the Store class to work with Media

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

import java.util.ArrayList;

//Doan Thanh Tung - 20225946
public class Store {
    private ArrayList<Media> itemsInStore = new ArrayList<Media>();

    public void addMedia(Media media) {
        itemsInStore.add(media);
        System.out.println(media.getTitle() + " is added successfully (DOAN THANH TUNG)");
    }

    public void removeMedia(Media media) {
        for (Media item : itemsInStore) {
            if (item.equals(media)) {
                itemsInStore.remove(item);
                System.out.println(media.getTitle() + " is removed successfully (DOAN THANH TUNG)");
                return;
            }
        }
        System.out.println("Not found");
    }
}
```

Figure 6.1: Store Class 1

7 Constructors of whole classes and parent classes

```
//Doan Thanh Tung - 20225946|
public Track(int length, String title) {
    this.length = length;
    this.title = title;
}
```

Figure 7.1: Constructor Track Class

```
//Doan Thanh Tung - 20225946|
public CompactDisc(int id, String title, String category, float cost, String artist,
    ArrayList<Track> tracks) {
    super(id, title, category, cost);
    this.artist = artist;
    this.tracks = tracks;
}
```

Figure 7.2: Constructor CompactDisc Class

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

```
//Doan Thanh Tung - 20225946
public Media(int id, String title, String category, float cost) {
    this.id = id;
    this.title = title;
    this.category = category;
    this.cost = cost;
}
```

Figure 7.3: Constructor Media Class

```
// Doan Thanh Tung - 20225946
public Disc(int id, String title, String category, float cost) {
    super(id, title, category, cost);
}

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;
}
```

Figure 7.4: Constructor Disc Class

8 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 lặp vào danh sách.

```
// Doan Thanh Tung
public boolean equals(Object o) {
    if (o instanceof Media) {
        Media media = (Media) o;
        if (media.getTitle() == this.title) {
            return true;
        }
    }
    return false;
}
```

Figure 8.1: Override equals in Media Class

```
//Doan Thanh Tung - 20225946
public boolean equals(Object o) {
    if(o instanceof Track) {
        Track track = (Track) o;
        if(track.getTitle().equals(title) && track.getLength() == length) {
            return true;
        }
    }
    return false;
}
```

Figure 8.2: Override equals in Track Class

9 Polymorphism with toString() method

```
//Doan Thanh Tung - 20225946|
public static void main(String[] args) {
    List<Media> mediae = new ArrayList<Media>();
    CompactDisc cd = new CompactDisc(1, "My Heart Will Go On", "Pop", 15.0f);
    DigitalVideoDisc dvd = new DigitalVideoDisc(2, "Inception", "Science Fiction", 199.99f);
    Book book = new Book(3, "Clean Code", "Programming", 35.5f);
    mediae.add(book);
    mediae.add(dvd);
    mediae.add(cd);
    for (Media m : mediae) {
        System.out.println(m.toString());
    }
}
```

Figure 9.1: Code mô phỏng Polymorphism

```
public String toString() {
    return "DOAN THANH TUNG ||" +
        "Media informations:\n" +
        "title: " + title +
        "\ncategory: " + category
        + "\ncost: " + cost + "\n";
}
```

Figure 9.2: Override toString() in Media Class

Kết quả

```
DOAN THANH TUNG ||Media informations:
title: Clean Code
category: Programming
cost: 35.5

DOAN THANH TUNG ||Media informations:
title: Inception
category: Science Fiction
cost: 199.99

DOAN THANH TUNG ||Media informations:
title: My Heart Will Go On
category: Pop
cost: 15.0

|
```

Figure 9.3: 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.

10 Sort media in the car

Sắp xếp các media trong giỏ hàng theo hai tiêu chí:

- Bảng 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.
- Bảng cost: Hiển thị theo thứ tự cost giảm dần. Trong trường hợp cost như nhau, sắp xếp media theo thứ tự bảng chữ cái

```
//Doan Thanh Tung - 20225946|
public static final Comparator<Media> COMPARE_BY_TITLE_COST =
    new MediaComparatorByTitleCost();
public static final Comparator<Media> COMPARE_BY_COST_TITLE =
    new MediaComparatorByCostTitle();
```

Figure 10.1: Add the comparators as attributes of the Media class

```
import java.util.Comparator;

public class MediaComparatorByCostTitle implements Comparator<Media> {

    //Doan Thanh Tung - 20225946
    public int compare(Media o1, Media o2) {
        return Comparator.comparing(Media::getCost)
            .thenComparing(Media::getTitle)
            .compare(o1, o2);
    }
}
```

Figure 10.2: MediaComparatorByCostTitle Class


```
public class MediaComparatorByTitleCost implements Comparator<Media> {  
    //Doan Thanh Tung - 20225946  
    public int compare(Media o1, Media o2) {  
        return Comparator.comparing(Media::getTitle)  
            .thenComparing(Media::getCost)  
            .compare(o1, o2);  
    }  
}
```

Figure 10.3: MediaComparatorByTitleCost Class

11 Create a complete console application in the Aims class

```
Doan Thanh Tung - 20225946 - AIMS:  
-----  
1.View store  
2.Update store  
3.See current cart  
0. Exit  
-----  
Please choose a number: 0-1-2-3  
|
```

Figure 11.1: Màn hình chính

11.1 Người dùng chọn 1: View store

```
Doan Thanh Tung - 20225946 - AIMS:
-----
1.View store
2.Update store
3.See current cart
0. Exit
-----
Please choose a number: 0-1-2-3
1
Book: Clean Code - Programming - [] - $35.5
CD: My Heart Will Go On - Pop - null - null - 0s - $15.0
DVD: Inception - Science Fiction - null - 0s - $199.99
DVD: alone in the dark - Science Fiction - null - 0s - $199.99

Doan Thanh Tung - 20225946 - 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 11.2: Vào Trang View Store

11.1.1 Người dùng tiếp tục chọn 1. See a media's details

```
Doan Thanh Tung - 20225946 - 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
1
Doan Thanh Tung || Enter the media's title:
Inception
DVD: Inception - Science Fiction - null - 0s - $199.99

Doan Thanh Tung - 20225946 - Options:
-----
1. Add to cart
2. Play
0. Back
-----
Please choose a number: 0-1-2
```

Figure 11.3: See a media's details

```
Doan Thanh Tung - 20225946 - Options:
-----
1. Add to cart
2. Play
0. Back
-----
Please choose a number: 0-1-2
1
Inception is added successfully (DOAN THANH TUNG)
Number of DVDs in the current cart: 1
```

Figure 11.4: Thêm vào Cart

11.1.2 Người dùng chọn 2: Add a media to the cart

```
Doan Thanh Tung - 20225946 - 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
Doan Thanh Tung || Enter the media's title:
Clean Code
Clean Code is added successfully (DOAN THANH TUNG)
Number of DVDs in the current cart: 2
```

Figure 11.5: Thêm media vào Cart

11.1.3 Người dùng chọn 3: Play a media

```
Doan Thanh Tung - 20225946 - 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
Doan Thanh Tung || Enter the media's title:
Inception
DOAN THANH TUNG || Playing DVD: Inception
DOAN THANH TUNG || DVD length: 0
```

Figure 11.6: Play a media

11.1.4 Người dùng chọn 4: See current cart

```
Doan Thanh Tung - 20225946 - 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
4
DVD: Inception - Science Fiction - null - 0s - $199.99
Book: Clean Code - Programming - [] - $35.5
```

Figure 11.7: See current cart after sort

11.2 Người dùng chọn 2: Update store

```
Doan Thanh Tung - 20225946 - AIMS:
-----
1.View store
2.Update store
3.See current cart
0. Exit
-----
Please choose a number: 0-1-2-3
2
Enter 1 to add media, 2 to remove media or 0 to back
```

Figure 11.8: Vào Trang Update Store

11.2.1 Người dùng chọn 1: Add a media to the store

```
Doan Thanh Tung - 20225946 - AIMS:
-----
1.View store
2.Update store
3.See current cart
0. Exit
-----
Please choose a number: 0-1-2-3
2
Enter 1 to add media, 2 to remove media or 0 to back
1
Media's ID:
5
Media title:
happy world ?
Media category:
horror
Media cost:
20.50
Kind of media: 1 for Disc, 2 for Book:
1
happy world ? is added successfully (DOAN THANH TUNG)
```

Figure 11.9: Add a media to store

=> Kết quả sau khi thêm

```
Doan Thanh Tung - 20225946 - AIMS:
-----
1.View store
2.Update store
3.See current cart
0. Exit
-----
Please choose a number: 0-1-2-3
1
Book: Clean Code - Programming - [] - $35.5
CD: My Heart Will Go On - Pop - null - null - 0s - $15.0
DVD: Inception - Science Fiction - null - 0s - $199.99
DVD: alone in the dark - Science Fiction - null - 0s - $199.99
CD: happy world - horror - null - null - 0s - $20.5
```

Figure 11.10: Result after add media to store

11.2.2 Người dùng chọn 2: Remove a media from the store

```
Doan Thanh Tung - 20225946 - AIMS:
-----
1.View store
2.Update store
3.See current cart
0. Exit
-----
Please choose a number: 0-1-2-3
2
Enter 1 to add media, 2 to remove media or 0 to back
2
Please enter the media's title:
happy world
happy world is removed successfully (DOAN THANH TUNG)
```

Figure 11.11: Remove a media from the store

=> Kết quả sau khi remove

```
Doan Thanh Tung - 20225946 - AIMS:
-----
1.View store
2.Update store
3.See current cart
0. Exit
-----
Please choose a number: 0-1-2-3
1
Book: Clean Code - Programming - [] - $35.5
CD: My Heart Will Go On - Pop - null - null - 0s - $15.0
DVD: Inception - Science Fiction - null - 0s - $199.99
DVD: alone in the dark - Science Fiction - null - 0s - $199.99
```

Figure 11.12: Result after remove a media

11.3 Người dùng chọn 3: See current cart

```
Doan Thanh Tung - 20225946 - 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
4
DVD: Inception - Science Fiction - null - 0s - $199.99
Book: Clean Code - Programming - [] - $35.5

Doan Thanh Tung - 20225946 - 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
```

Figure 11.13: Vào trang See current cart

Giả sử lúc này trong Cart sẽ có các Media sau

```
DVD: Inception - Science Fiction - null - 0s - $199.99
Book: Clean Code - Programming - [] - $35.5
```

Figure 11.14: Media in Cart

11.3.1 Người dùng chọn 1: Filter medias in cart

```
Doan Thanh Tung - 20225946 - 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 an option to filter Media: 1: By title, 2: By id:
2
Please enter the id: 2
DVD: Inception - Science Fiction - null - 0s - $199.99
```

Figure 11.15: Filter Cart By id

```
Doan Thanh Tung - 20225946 - 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 an option to filter Media: 1: By title, 2: By id:
1
Please enter the title: Inception
DVD: Inception - Science Fiction - null - 0s - $199.99
```

Figure 11.16: Filter Cart By Title

11.3.2 Người dùng chọn 2: Sort medias in cart

```
Doan Thanh Tung - 20225946 - 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
Please choose if you'd like to sort by title or cost: 0 - title, 1 - cost
0
cart is sorted by title successfully
Book: Clean Code - Programming - [] - $35.5
DVD: Inception - Science Fiction - null - 0s - $199.99
```

Figure 11.17: Sort Cart By Title

```
Doan Thanh Tung - 20225946 - 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
Please choose if you'd like to sort by title or cost: 0 - title, 1 - cost
1
cart is sorted by cost successfully
Book: Clean Code - Programming - [] - $35.5
DVD: Inception - Science Fiction - null - 0s - $199.99
```

Figure 11.18: Sort Cart By Cost

11.3.3 Người dùng chọn 3: Remove media from cart

```
Doan Thanh Tung - 20225946 - 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
please enter a media's title to remove:
Inception
Inception is removed successfully
```

Figure 11.19: Remove media by id

=> Kết quả

```
Doan Thanh Tung - 20225946 - 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
Please choose if you'd like to sort by title or cost: 0 - title, 1 - cost
0
cart is sorted by title successfully
Book: Clean Code - Programming - [] - $35.5
```

Figure 11.20: Result after remove media in cart by id

11.3.4 Người dùng chọn 4: Play a media

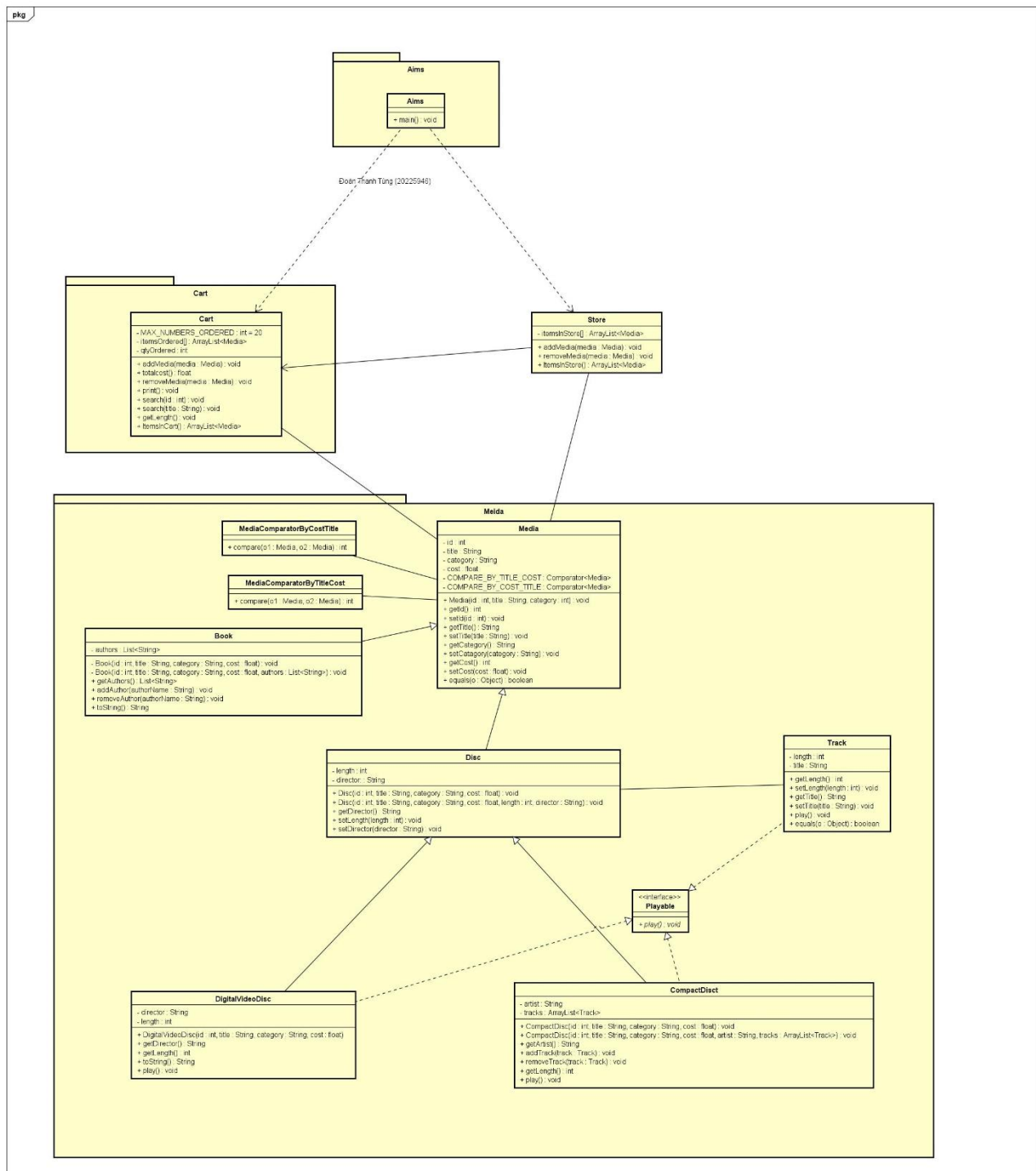
```
Doan Thanh Tung - 20225946 - Options:
-----
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
-----
Please choose a number: 0-1-2-3-4-5
4
Doan Thanh Tung || Enter the media's title:
Inception
DOAN THANH TUNG || Playing DVD: Inception
DOAN THANH TUNG || DVD length: 0
```

Figure 11.21: Play a media in cart

11.3.5 Người dùng chọn 5: Place order

```
Doan Thanh Tung - 20225946 - 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
An order is created!
```

Figure 11.22: Order



12 Class Diagram

Figure 12.1: Class Diagram

13 UseCase Diagram

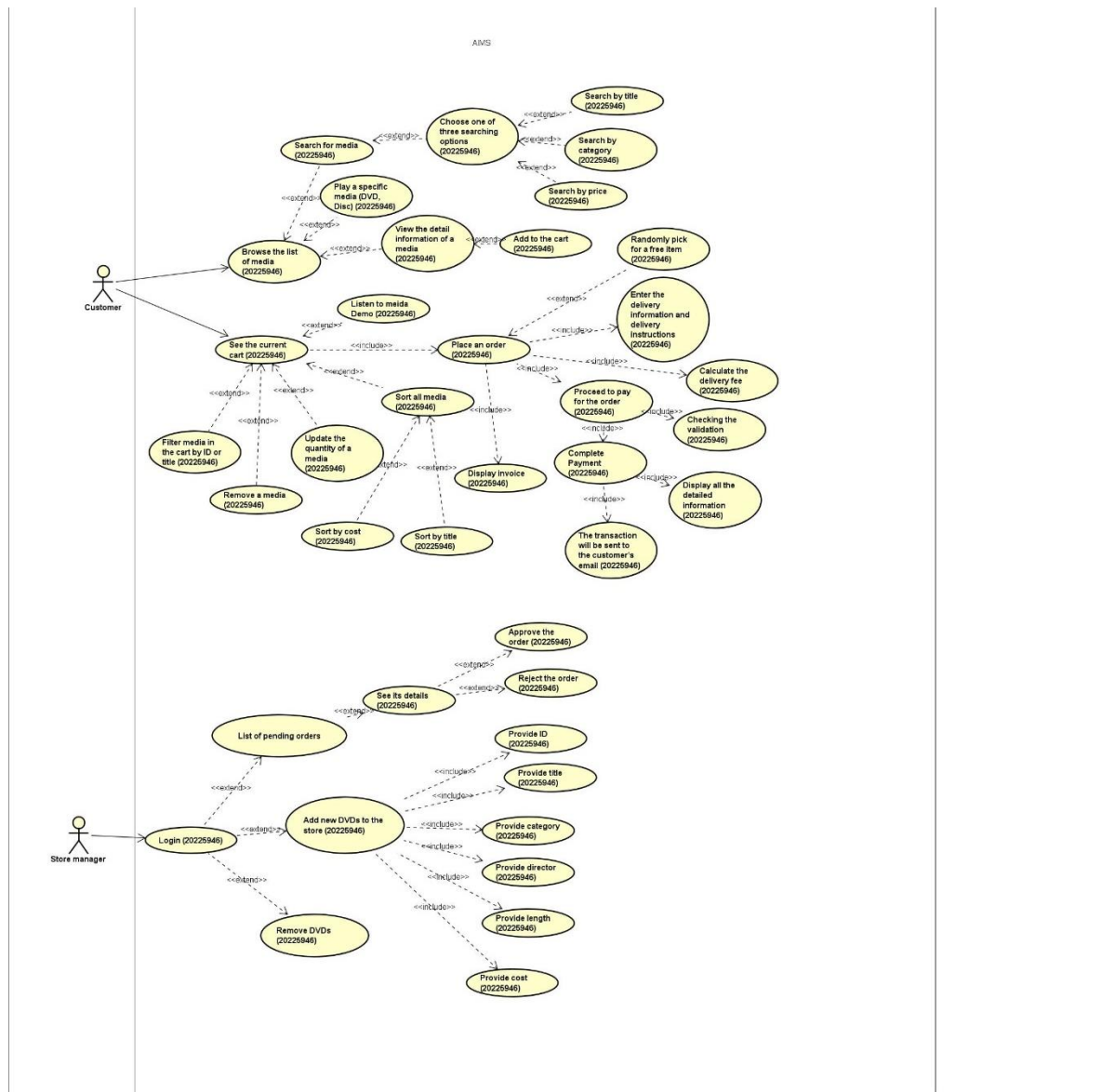


Figure 13.1: UseCase Diagram

14 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.

```

1 package hust.soict.dsai.aims.media;
2
3 import java.util.Comparator;
4
5 public class MediaComparatorByCostTitle implements Comparator<Media> {
6
7     // Doan Thanh Tung - 20225946
8     public int compare(Media o1, Media o2) {
9         return Comparator.comparing(Media::getCost).compare(o1, o2);
10    }
11 }
12

```

Figure 14.1: Triển khai Comparable trong lớp trừu tượng Media

```

1 package hust.soict.dsai.aims.media;
2
3 import java.util.Comparator;
4
5 public class MediaComparatorByCostTitle implements Comparator<Media> {
6
7     // Doan Thanh Tung - 20225946
8     public int compare(Media o1, Media o2) {
9         return Comparator.comparing(Media::getCost).thenComparing(Media::getTitle).compare(o1, o2);
10    }
11 }
12

```

Figure 14.2: Mở rộng để so sánh nhiều thuộc tính hơn

```

8 public abstract class Media {
9     int id;
10    String title;
11    String category;
12    float cost;
13    // Doan Thanh Tung - 20225946
14    public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost();
15    public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle();
16

```

Figure 14.3: Triển khai tại lớp con

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.