

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



BÁO CÁO THỰC HÀNH LAB 04
Học phần: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Mã học phần: IT3103
Mã lớp: 744530

Giảng viên hướng dẫn: GV Lê Thị Hoa
Sinh viên thực hiện: Tạ Hồng Phúc
MSSV: 20225906

Hà Nội, tháng 11 năm 2024

Báo cáo thực hành Lab 04

Contents

1	Create the Book class	5
2	Creating the abstract Media class	6
3	Creating the CompactDisc class.....	7
3.1	Create the Disc class extending the Media class.....	7
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.....	9
3.3	Open the CompactDisc class	9
4	Create the Playable interface	11
5	Update the Cart class to work with Media.....	12
6	Update the Store class to work with Media.....	14
7	Constructors of whole classes and parent classes.....	15
8	Unique item in a list	16
9	Polymorphism with <i>toString()</i> method.....	17
10	Sort media in the car.....	18
11	Create a complete console application in the Aims class	19
11.1	Nguời dùng chọn 1: View store.....	19
11.1.1	Người dùng tiếp tục chọn 1. See a media's details.....	20
11.1.2	Người dùng chọn 2: Add a media to the cart.....	21
11.1.3	Người dùng chọn 3: Play a media.....	21
11.1.4	Người dùng chọn 4: See current cart.....	22
11.2	Nguời dùng chọn 2: Update store	23
11.2.1	Người dùng chọn 1: Add a media to the store	23
11.2.2	Người dùng chọn 2: Remove a media from the store	24
11.3	Nguời dùng chọn 3: See current cart.....	25
11.3.1	Người dùng chọn 1: Filter medias in cart	26
11.3.2	Người dùng chọn 2: Sort medias in cart.....	27
11.3.3	Người dùng chọn 3: Remove media from cart	28
11.3.4	Người dùng chọn 4: Play a media.....	29
11.3.5	Người dùng chọn 5: Place order	29
12	Class Diagram	30
13	UseCase Diagram	31
14	Answer Question	31

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

Source Code:

```
1 package hust.soict.dsai.aims.media;
2 import java.util.List; // Import the List interface
3 import java.util.ArrayList; // Import the ArrayList class
4
5 public class Book_phucth {
6     private int id;
7     private String title;
8     private String category;
9     private float cost;
10    private List<String> authors = new ArrayList<String>();
11
12
13    public int getId() {
14        return id;
15    }
16    public void setId(int id) {
17        this.id = id;
18    }
19    public String getTitle() {
20        return title;
21    }
22    public void setTitle(String title) {
23        this.title = title;
24    }
25    public String getCategory() {
26        return category;
27    }
28    public void setCategory(String category) {
29        this.category = category;
30    }
31    public float getCost() {
32        return cost;
33    }
34    public void setCost(float cost) {
35        this.cost = cost;
36    }
37    public Book_phucth() {
38        // TODO Auto-generated constructor stub
39    }
40}
```

Figure 1.1: Book Class 1

```
41    public void addAuthor(String authorName) {
42        if(!authors.contains(authorName)) {
43            authors.add(authorName);
44            System.out.println("Author " + authorName + " added successfully!");
45        }
46        else {
47            System.out.println("Author already exists!");
48        }
49    }
50
51    public void removeAuthor(String authorName) {
52        if(!authors.contains(authorName)) {
53            authors.remove(authorName);
54            System.out.println("Author " + authorName + " removed!");
55        }
56        else {
57            System.out.println("Author do not exist.");
58        }
59    }
60 }
```

Figure 1.2: Book Class 2

2 Creating the abstract Media class

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

Source Code:

```
1 package hust.soict.dsai.aims.media;
2
3 public class Media_phucth {
4
5     private int id;
6     private String title;
7     private String category;
8     private float cost;
9
10    public int getId() {
11        return id;
12    }
13
14    public void setId(int id) {
15        this.id = id;
16    }
17
18    public String getTitle() {
19        return title;
20    }
21
22    public void setTitle(String title) {
23        this.title = title;
24    }
25
26    public String getCategory() {
27        return category;
28    }
29
30    public void setCategory(String category) {
31        this.category = category;
32    }
33
34    public float getCost() {
35        return cost;
36    }
37
38    public void setCost(float cost) {
39        this.cost = cost;
40    }
```

Figure 2.1: Media Class 1

```
1 public Media_phucth() {
2     // TODO Auto-generated constructor stub
3 }
4 public Media_phucth(int id, String title, String category, float cost) {
5     this.id = id;
6     this.title = title;
7     this.category = category;
8     this.cost = cost;
9 }
```

Figure 2.2: Media Class 2

3 Creating the CompactDisc class

3.1 Create the Disc class extending the Media class

Source Code:

```
1 package hust.soict.dsai.aims.media;
2
3 public class Disc_phucth extends Media_phucth{
4     private String director;
5     private int length;
6
7
8     public String getDirector() {
9         return director;
10    }
11
12    public int getLength() {
13        return length;
14    }
15
16    public Disc_phucth() {
17        super();
18    }
19    // Create Disk with title
20    public Disc_phucth(String title) {
21        super();
22        setTitle(title);
23    }
24
25    // Create Disk with title, category, cost
26    public Disc_phucth(String title, String category, float cost) {
27        super();
28        setTitle(title);
29        setCategory(category);
30        setCost(cost);
31    }
32
33    // Create Disk with title, category, director, length, cost
34    public Disc_phucth(String title, String category, String director, int length, float cost) {
35        super();
36        setTitle(title);
37        setCategory(category);
38        this.director = director;
39        this.length = length;
40        setCost(cost);
41    }
42 }
```

Figure 3.1: Disc Class

```
1 package hust.soict.dsai.aims.media;
2
3 public class DigitalVideoDisc_phucth extends Disc_phucth implements Playable_phucth{
4     private String director;
5     private int length;
6     private static int nbDVD_phucth = 0;
7     public String getDirector() {
8         return director;
9     }
10    public void setDirector(String director) {
11        this.director = director;
12    }
13    public int getLength() {
14        return length;
15    }
16    public void setLength(int length) {
17        this.length = length;
18    }
19    //Create DVD by Title
20    public DigitalVideoDisc_phucth(String title) {
21        super();
22        setTitle(title);
23        nbDVD_phucth++;
24        setId(nbDVD_phucth);
25    }
26    //Create DVD by Title, Category, Cost
27    public DigitalVideoDisc_phucth(String title, String category, float cost) {
28        super();
29        setTitle(title);
30        setCategory(category);
31        setCost(cost);
32        nbDVD_phucth++;
33        setId(nbDVD_phucth);
34    }
35    //Create DVD by Title, Category, Director, Cost
36    public DigitalVideoDisc_phucth(String title, String category, String director, float cost) {
37        super();
38        setTitle(title);
39        setCategory(category);
40        setDirector(director);
41        setCost(cost);
42        nbDVD_phucth++;
43        setId(nbDVD_phucth);
44    }
45    //Create DVD by Title, Category, Director, Length, Cost
46    public DigitalVideoDisc_phucth(String title, String category, String director, int length, float cost) {
47        super();
48        setTitle(title);
49        setCategory(category);
50        setDirector(director);
51        setLength(length);
52        setCost(cost);
53        nbDVD_phucth++;
54        setId(nbDVD_phucth);
55    }
}
```

Figure 3.2: DigitalVideoDisc Class

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

public class CompactDisc_phucth extends Disc_phucth {

    public CompactDisc_phucth() {
        // TODO Auto-generated constructor stub
        super();
    }
}
```

Figure 3.3: CompactDisc Class

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.

Source code:

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

public class Track_phucth{
    private String title;
    private int length;
    public String getTitle() {
        return title;
    }
    public int getLength() {
        return length;
    }
    public Track_phucth() {
        // TODO Auto-generated constructor stub
        super();
    }
    // Create Track with title
    public Track_phucth(String title) {
        super();
        this.title = title;
    }
    // Create Track with title and length
    public Track_phucth(String title, int length) {
        super();
        this.title = title;
        this.length = length;
    }
}
```

Figure 3.4: Track Class

3.3 Open the CompactDisc class

Source code:

```
1 package hust.soict.dsai.aims.media;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class CompactDisc_phucth extends Disc_phucth{
7     private String artist;
8     private List<Track_phucth> tracks_phucth = new ArrayList<Track_phucth>();
9
10    public String getArtist() {
11        return artist;
12    }
13    //Default constructor
14    public CompactDisc_phucth() {
15        // TODO Auto-generated constructor stub
16        super();
17    }
}
```

Figure 3.5: CompactDisc Class 1

```
// Create CD with ID and title
public CompactDisc_phucth(int id, String title) {
    super();
    setId(id);
    setTitle(title);
}
// Create CD with ID, title and category
public CompactDisc_phucth(int id, String title, String category) {
    super();
    setId(id);
    setTitle(title);
    setCategory(category);
}
// Create CD with ID, title, category and cost
public CompactDisc_phucth(int id, String title, String category, float cost) {
    super();
    setId(id);
    setTitle(title);
    setCategory(category);
    setCost(cost);
}
public void addTrack(Track_phucth track) {
    if(!tracks_phucth.contains(track)) {
        tracks_phucth.add(track);
        System.out.println("Track added successfully!");
    }
    else {
        System.out.println(track + " already exists!");
    }
}
public void removeTrack(Track_phucth track) {
    if(tracks_phucth.contains(track)) {
        tracks_phucth.remove(track);
        System.out.println("Track removed!");
    }
    else {
        System.out.println(track + " not exist!");
    }
}
public int getLength(){
    int totalLength = 0;
    for(Track_phucth track: tracks_phucth) {
        totalLength += track.getLength();
    }
    return totalLength;
}
```

Figure 3.6: CompactDisc Class 2

4 Create the Playable interface

Source code:

```
1 package hust.soict.dsai.aims.media;  
2  
3 public interface Playable_phucth {  
4     public void play();  
5 }  
6
```

Figure 4.1: Playable interface

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

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

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

Figure 4.2: Method play() của DigitalVideoDisc

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

Figure 4.3: Method play() của Track

```
public void play() {  
    System.out.println("Playing DVD: " + this.getTitle());  
    System.out.println("Number of track: " + tracks_phucth.size());  
  
    for (int i = 0; i < tracks_phucth.size(); i++) {  
        System.out.print("Track " + (i + 1) + ": ");  
        tracks_phucth.get(i).play(); // Call the play method of each Track  
    }  
}
```

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.

Source Code:

```
package lab04;

import java.util.ArrayList;
import java.util.List;
import hust.soict.dsai.aims.media.Media_phucth;

public class Cart_phucth {
    public static final int MAX_NUMBERS_ORDERED = 20;
    private List<Media_phucth> itemsOrdered = new ArrayList<Media_phucth>();

    public void addMedia_phucth(Media_phucth product) {
        if(!itemsOrdered.contains(product)) {
            itemsOrdered.add(product);
            System.out.println("Media added successfully!");
        }
        else {
            System.out.println("Media existed!");
        }
    }

    public void removeMedia_phucth(Media_phucth product) {
        if(itemsOrdered.contains(product)) {
            itemsOrdered.remove(product);
            System.out.println("Media removed.");
        }
        else {
            System.out.println("Media not exist!");
        }
    }
}
```

Figure 5.1: Cart Class 1

```
public float totalCost() {  
    float totalCost = 0.0f;  
  
    for (int i=0; i<itemsOrdered.size(); i++) {  
        if (itemsOrdered.get(i) != null) {  
            totalCost += itemsOrdered.get(i).getCost();  
        }  
    }  
    return totalCost;  
}  
  
// Print items in cart  
public void printItemsInCart() {  
    System.out.println("*****CART*****");  
    System.out.println("Ordered Items:");  
    for(int i=0; i<itemsOrdered.size(); i++) {  
        Media_phucth dvd = itemsOrdered.get(i);  
        System.out.printf("%d. %s - %d: %.2f$ \n", i+1,  
                          dvd.getTitle(), dvd.getCategory(), dvd.getCost());  
    }  
  
    float totalCost = totalCost();  
    System.out.printf("Total cost: %.2f $ \n", totalCost);  
    System.out.println("*****");  
}
```

Figure 5.2: Cart Class 2

```
// Search by ID  
public boolean searchById(int id) {  
    int n = itemsOrdered.size();  
    for(int i=0; i<n; i++) {  
        if(itemsOrdered.get(i).getId() == id) return true;  
    }  
    return false;  
}  
  
// Search by title  
public void searchByTitle(String title) {  
    int n = itemsOrdered.size();  
    boolean check = false;  
    for(int i=0; i<n; i++) {  
        if(itemsOrdered.get(i).isMatch(title)) {  
            System.out.println("Your cart has a DVD as title name is: " + title);  
            check = true;  
        }  
    }  
    if(!check) {  
        System.out.println("No data is matching");  
    }  
}
```

Figure 5.3: Cart Class 3

6 Update the Store class to work with Media

Source code:

```
package lab04;

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

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

public class Store_phucth {
    public static final int MAX_DVD = 50;
    private List<Media_phucth> itemsInStore = new ArrayList<Media_phucth>();
    private int qtyOrdered = 0;

    public int getQtyOrdered() {
        return qtyOrdered;
    }

    public void setQtyOrdered(int n) {
        qtyOrdered += n;
    }
}
```

Figure 6.1: Store Class 1

```
public void addMedia_phucth(Media_phucth product) {
    if(!itemsInStore.contains(product)) {
        itemsInStore.add(product);
        System.out.println("Media added successfully!");
    }
    else {
        System.out.println("Media existed!");
    }
}

public void removeMedia_phucth(Media_phucth product) {
    if(itemsInStore.contains(product)) {
        itemsInStore.remove(product);
        System.out.println("Media removed.");
    }
    else {
        System.out.println("Media not exist!");
    }
}
```

Figure 6.2: Store Class 2

7 Constructors of whole classes and parent classes

```
public Track_phucth() {
    // TODO Auto-generated constructor stub
    super();
}
// Create Track with title
public Track_phucth(String title) {
    super();
    this.title = title;
}
// Create Track with title and length
public Track_phucth(String title, int length) {
    super();
    this.title = title;
    this.length = length;
}
```

Figure 7.1: Constructor Track Class

```
//Default constructor
public CompactDisc_phucth() {
    // TODO Auto-generated constructor stub
    super();
}
// Create CD with ID and title
public CompactDisc_phucth(int id, String title) {
    super();
    setId(id);
    setTitle(title);
}
// Create CD with ID, title and category
public CompactDisc_phucth(int id, String title, String category) {
    super();
    setId(id);
    setTitle(title);
    setCategory(category);
}
// Create CD with ID, title, category and cost
public CompactDisc_phucth(int id, String title, String category, float cost) {
    super();
    setId(id);
    setTitle(title);
    setCategory(category);
    setCost(cost);
}
```

Figure 7.2: Constructor CompactDisc Class

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

```
//Default constructor
public Media_phucth() {
    // TODO Auto-generated constructor stub
}
// Constructor with title
public Media_phucth(String title) {
    this(title, "", 0.0f);
}
// Constructor with title and category
public Media_phucth(String title, String category) {
    this(0, title, category, 0.0f);
}
// Constructor with title, category, and cost
public Media_phucth(String title, String category, float cost) {
    this(0, title, category, cost);
}
public Media_phucth(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

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.

```
@Override
public boolean equals(Object obj_phucth) {
    // Check if the object is null or not an instance of Media
    if (this == obj_phucth) {
        return true; // If they refer to the same object, they are equal
    }
    if (obj_phucth == null || !(obj_phucth instanceof Media_phucth)) {
        return false; // Not equal if the object is null or not a Media
    }

    // Cast obj to Media and compare their titles
    Media_phucth other = (Media_phucth) obj_phucth;
    return this.title.equals(other.title);
}
```

Figure 8.1: Override equals in Media Class

```
@Override
public boolean equals(Object obj_phucth) {
    // Check if the object is null or not an instance of Track
    if (this == obj_phucth) {
        return true; // If they refer to the same object, they are equal
    }
    if (obj_phucth == null || !(obj_phucth instanceof Track_phucth)) {
        return false; // Not equal if the object is null or not a Track
    }

    // Cast obj to Track and compare their titles and lengths
    Track_phucth other = (Track_phucth) obj_phucth;
    return this.title.equals(other.title) && this.length == other.length;
}
```

Figure 8.2: Override equals in Track Class

9 Polymorphism with `toString()` method

```
// Print with toString()
public void printCart() {
    for (Media_phucth m : itemsOrdered) {
        System.out.println(m.toString());
    }
}
```

Figure 9.1: Code mô phỏng Polymorphism

```
@Override
public String toString() {
    return "Media_phucth{" +
        "title=''" + title + '\'' +
        ", cost=" + cost +
        '}';
}

@Override
public String toString() {
    return "DVD_phucth{" +
        "director=''" + director + '\'' +
        ", length=" + length +
        '}' + super.toString();
}
```

Figure 9.2: Override `toString()` in Media Class

Kết quả

```
Media added successfully!
Media added successfully!
DVD_phucth{director='Victor Vũ', length=120}Media_phucth{title='Mắt biếc', cost=20.5}
DVD_phucth{director='Thánh Trần', length=87}Media_phucth{title='Bố già', cost=24.15}

Total cost is: 44.65$
```

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 kiểu cụ thể của từng phần tử.

- Nhận xét: Do Media class chứa các thuộc tính chung và mỗi lớp con có thể có thuộc tính riêng (Ví dụ như class DVD có director và length) nên ta sẽ tiến hành override ở cả lớp cha và các lớp con và thu được kết quả như trong hình.

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

```
public class Media_phucth {  
    public static final Comparator<Media_phucth> COMPARE_BY_TITLE_COST =  
        new MediaComparatorByTitleCost();  
  
    public static final Comparator<Media_phucth> COMPARE_BY_COST_TITLE =  
        new MediaComparatorByCostTitle();
```

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

```
package hust.soict.dsai.aims.media;  
  
import java.util.Comparator;  
  
public class MediaComparatorByCostTitle implements Comparator<Media_phucth> {  
  
    public MediaComparatorByCostTitle() {}  
  
    @Override  
    public int compare(Media_phucth m1, Media_phucth m2) {  
        return Comparator.comparing(Media_phucth::getCost) // Descending cost  
            .thenComparing(Media_phucth::getTitle) // Ascending title  
            .compare(m1, m2);  
    }  
}
```

Figure 10.2: MediaComparatorByCostTitle Class

```
package hust.soict.dsai.aims.media;  
  
import java.util.Comparator;  
  
public class MediaComparatorByTitleCost implements Comparator<Media_phucth> {  
  
    public MediaComparatorByTitleCost() {}  
  
    @Override  
    public int compare(Media_phucth m1, Media_phucth m2) {  
        return Comparator.comparing(Media_phucth::getTitle)  
            .thenComparing((m1Inner, m2Inner) -> Float.compare(m2Inner.getCost(), m1Inner.getCost()))  
            .compare(m1, m2);  
    }  
}
```

Figure 10.3: MediaComparatorByTitleCost Class

11 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:

Figure 11.1: Màn hình chính

11.1 Người dùng chọn 1: View store

Please choose a number:

1

*****CART*****

Available Items:

- 1. Doraemon – Viễn tưởng: 20.00\$
 - 2. Conan – Trinh thám: 50.00\$
 - 3. Shin – crayon boy – Hài hước: 10.00\$
 - 4. Miko – Tình cảm: 40.00\$
 - 5. Dragonball – Hành động: 20.00\$
- *****

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

Please choose a number: 0-1-2-3-4

1

Enter the title of the media:

Conan

DVD_phucth{director='Gosho Aoyama', length=120}Media_phucth{title='Conan', cost=50.0}

Options:

-
- 1. Add to cart
 - 2. Play
 - 0. Back

Figure 11.3: See a media's details

Please choose a number: 0-1-2

1

Media added successfully!

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 11.4: Thêm vào Cart

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

```
Please choose a number: 0-1-2-3-4
2
Enter the title of the media to add to cart:
Dragonball
Media added successfully!
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 11.5: Thêm media vào Cart

11.1.3 Người dùng chọn 3: Play a media

```
Please choose a number: 0-1-2-3-4
3
Enter the title of the media to play:
Miko
Playing DVD: Miko
DVD length: 120
Options:
```

Figure 11.6: Play a media

11.1.4 Người dùng chọn 4: See current cart

```
Please choose a number: 0-1-2-3-4
4
*****CART*****
Ordered Items:
1. Conan – Trinh thám: 50.00$
2. Dragonball – Hành động: 20.00$
Total cost: 70.00 $
*****
```

Figure 11.7: See current cart after sort

11.2 Người dùng chọn 2: Update store

AIMS:

-
- 1. View store
 - 2. Update store
 - 3. See current cart
 - 0. Exit
-

Please choose a number:

2

Enter 1 to add media, 2 to remove media:

Figure 11.8: Vào Trang Update Store

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

Please choose a number:

2

Enter 1 to add media, 2 to remove media:

1

Enter media details (title, category, cost):

J97

Children

5000000

Media added successfully!

Figure 11.9: Add a media to store

=> Kết quả sau khi thêm

```
Please choose a number:  
1  
*****CART*****  
Available Items:  
1. Doraemon - Viễn tưởng: 20.00$  
2. Conan - Trinh thám: 50.00$  
3. Shin - crayon boy - Hài hước: 10.00$  
4. Miko - Tình cảm: 40.00$  
5. Dragonball - Hành động: 20.00$  
6. J97 - Children: 5000000.00$  
*****
```

Figure 11.10: Result after add media to store

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

```
Please choose a number:  
2  
Enter 1 to add media, 2 to remove media:  
2  
Enter the title of the media to remove:  
J97  
Media with title 'J97' removed.
```

Figure 11.11: Remove a media from the store

=> Kết quả sau khi remove

```
*****CART*****  
Available Items:  
1. Doraemon - Viễn tưởng: 20.00$  
2. Conan - Trinh thám: 50.00$  
3. Shin - crayon boy - Hài hước: 10.00$  
4. Miko - Tình cảm: 40.00$  
5. Dragonball - Hành động: 20.00$  
*****
```

Figure 11.12: Result after remove a media

11.3 Người dùng chọn 3: See current cart

```
Please choose a number:  
3  
*****CART*****  
Ordered Items:  
1. Doraemon – Viễn tưởng: 20.00$  
2. Conan – Trinh thám: 50.00$  
3. Shin – crayon boy – Hài hước: 10.00$  
4. Miko – Tình cảm: 40.00$  
5. Dragonball – Hành động: 20.00$  
Total cost: 140.00 $  
*****  
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

```
*****CART*****
Ordered Items:
1. Doraemon - Viễn tưởng: 20.00$
2. Conan - Trinh thám: 50.00$
3. Shin - crayon boy - Hài hước: 10.00$
4. Miko - Tình cảm: 40.00$
5. Dragonball - Hành động: 20.00$
Total cost: 140.00 $
*****
```

Figure 11.14: Media in Cart

11.3.1 Người dùng chọn 1: Filter medias in cart

```
Please choose a number: 0-1-2-3-4-5
1
1. Filter by ID
2. Filter by title
1
Nhập ID cần tìm: 3
DVD_phucth{director='Yoshito Usui', length=120}Media_phucth{title='Shin - crayon boy', cost=10.0}
```

Figure 11.15: Filter Cart By id

```
Please choose a number: 0-1-2-3-4-5
1
1. Filter by ID
2. Filter by title
2
Nhập title cần tìm: Conan
DVD_phucth{director='Gosho Aoyama', length=120}Media_phucth{title='Conan', cost=50.0}
```

Figure 11.16: Filter Cart By Title

11.3.2 Người dùng chọn 2: Sort medias in cart

- 1. Sort medias in cart by title
- 2. Sort medias in cart by cost
- 0. Back

Please choose a number: 0-1-2

1

*****CART*****

Ordered Items:

- 1. Conan – Trinh thám: 50.00\$
- 2. Doraemon – Viễn tưởng: 20.00\$
- 3. Dragonball – Hành động: 20.00\$
- 4. Miko – Tình cảm: 40.00\$
- 5. Shin – crayon boy – Hài hước: 10.00\$

Total cost: 140.00 \$

Figure 11.17: Sort Cart By Title

- 1. Sort medias in cart by title
- 2. Sort medias in cart by cost
- 0. Back

Please choose a number: 0-1-2

2

*****CART*****

Ordered Items:

- 1. Conan – Trinh thám: 50.00\$
- 2. Miko – Tình cảm: 40.00\$
- 3. Doraemon – Viễn tưởng: 20.00\$
- 4. Dragonball – Hành động: 20.00\$
- 5. Shin – crayon boy – Hài hước: 10.00\$

Total cost: 140.00 \$

Figure 11.18: Sort Cart By Cost

11.3.3 Người dùng chọn 3: Remove media from cart

```
Please choose a number: 0-1-2-3-4-5
3
Enter ID to delete:
4
Media removed: Shin - crayon boy
```

Figure 11.19: Remove media by id

=> Kết quả

```
*****CART*****
Ordered Items:
1. Doraemon - Viễn tưởng: 20.00$
2. Conan - Trinh thám: 50.00$
3. Miko - Tình cảm: 40.00$
4. Dragonball - Hành động: 20.00$
Total cost: 130.00 $
*****
```

Figure 11.20: Result after remove media in cart by id

11.3.4 Người dùng chọn 4: Play a media

```
Please choose a number: 0-1-2-3-4-5
4
Enter the title of the media to play:
Conan
Playing DVD: Conan
DVD length: 120
```

Figure 11.21: Play a media in cart

11.3.5 Người dùng chọn 5: Place order

```
Please choose a number: 0-1-2-3-4-5
5
Your order is submitted!
Options:
```

Figure 11.22: Order

=> Kết quả sau khi order

```
Please choose a number:
3
*****CART*****
Ordered Items:
Total cost: 0.00 $
*****
Options:
```

Figure 11.23: Result after order

12 Class Diagram

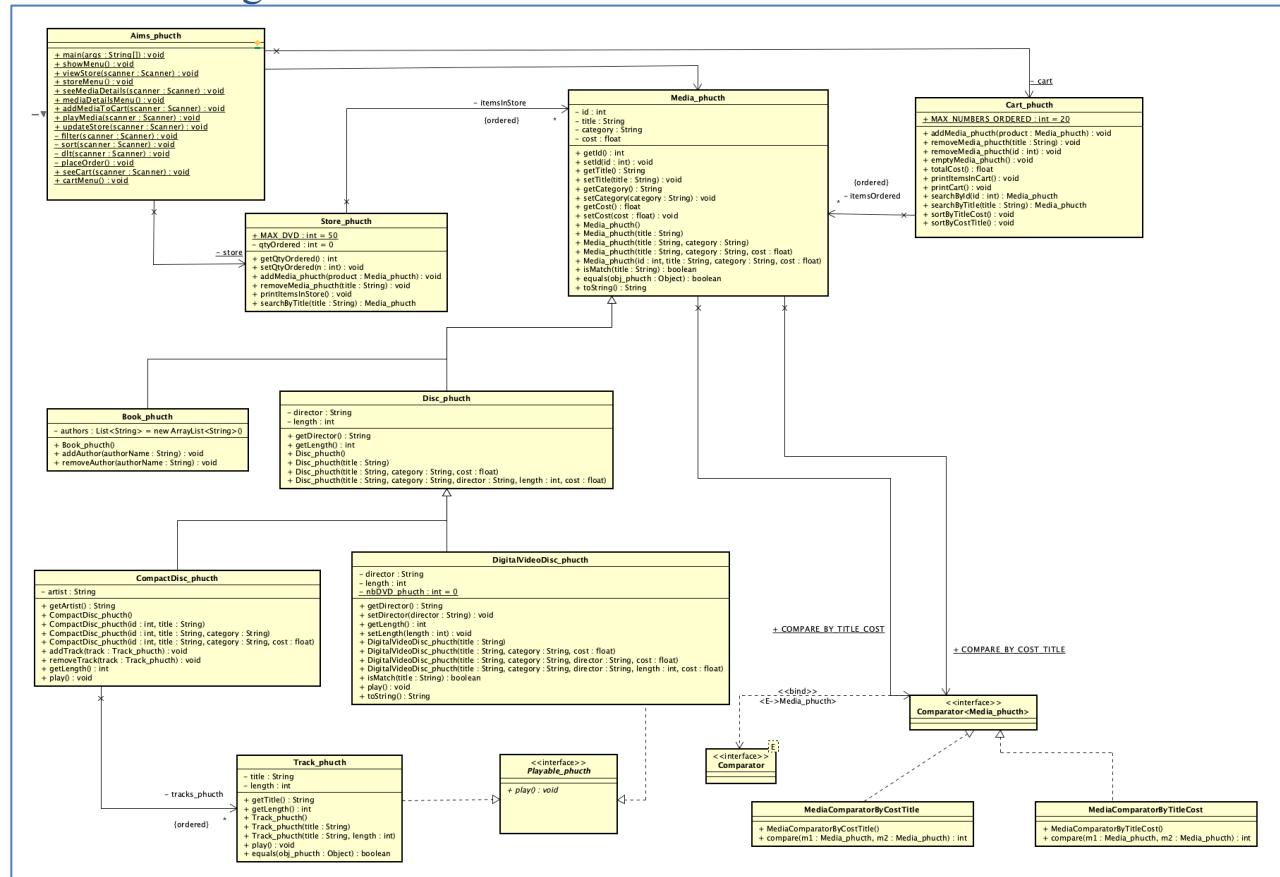


Figure 12.1: Class Diagram

13 UseCase Diagram

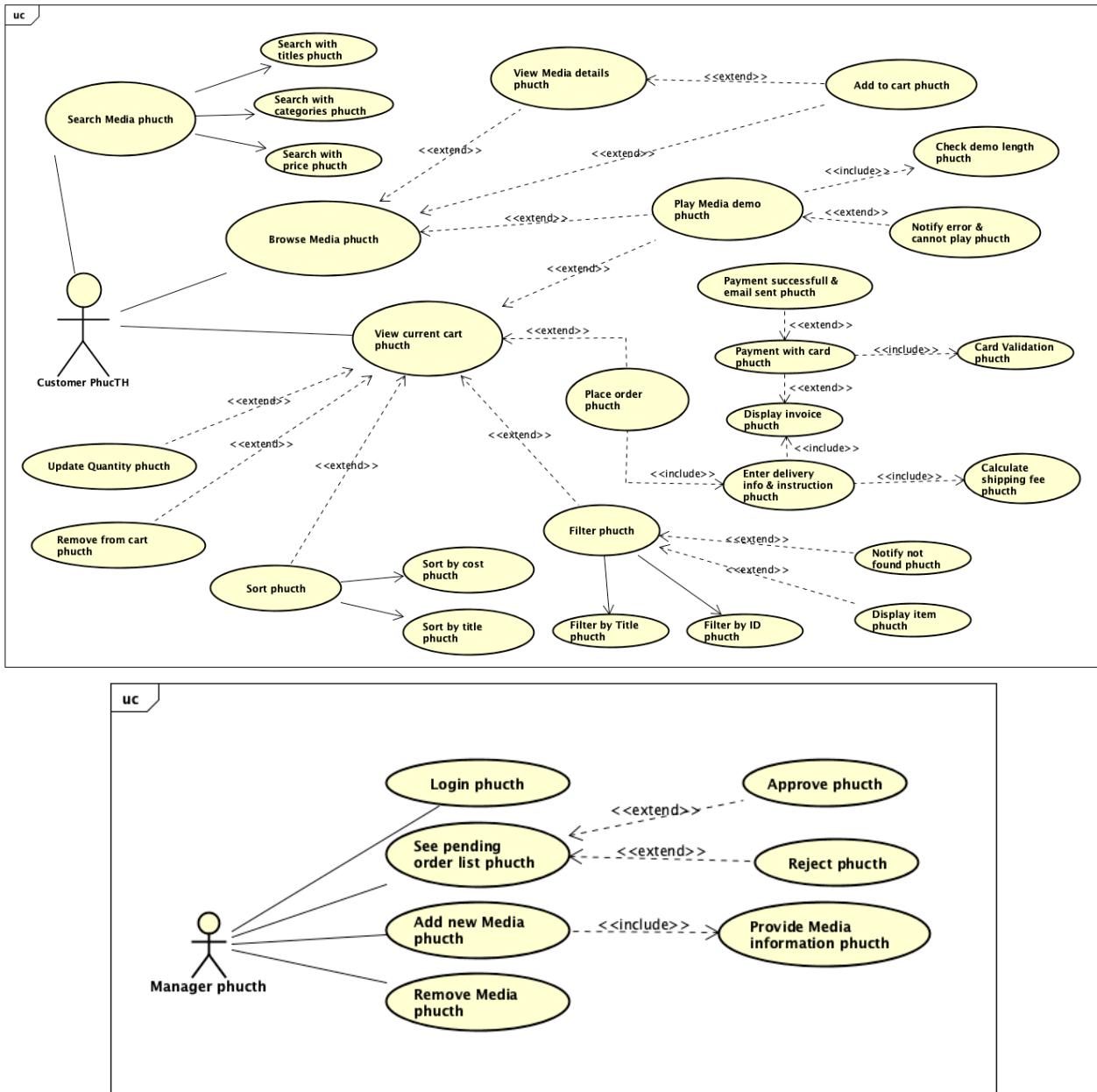


Figure 13.1: UseCase Diagram

14 Answer Question

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.

Lớp triển khai Comparable nên là lớp đại diện cho sản phẩm hoặc các mục trong giỏ hàng. Trong trường hợp này, lớp **Media** hoặc các lớp con của nó như **DVD** là ứng cử viên phù hợp để triển khai Comparable.

```
public abstract class Media_phucth implements Comparable<Media_phucth> {
    @Override
    public int compareTo(Media_phucth otherMedia) {
        int titleComparison = this.title.compareTo(otherMedia.getTitle());
        // So sánh theo tiêu đề trước
        return (titleComparison == 0) ? Float.compare(this.cost, otherMedia.getCost()) : titleComparison;
    }
}
```

Figure 14.1: Triển khai Comparable trong lớp trừu tượng Media

```
public abstract class Media_phucth implements Comparable<Media_phucth> {
    @Override
    public int compareTo(Media_phucth otherMedia) {
        // So sánh theo tiêu đề
        return this.title.compareTo(otherMedia.getTitle());
    }
}
```

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

```
public class DigitalVideoDisc_phucth extends Disc_phucth implements Playable_phucth{
    @Override
    public int compareTo(Media_phucth otherMedia) {
        if (otherMedia instanceof DigitalVideoDisc_phucth) {
            DigitalVideoDisc_phucth otherDVD = (DigitalVideoDisc_phucth) otherMedia;
            // So sánh theo tiêu đề trước
            int titleComparison = getTitle().compareTo(otherDVD.getTitle());
            // Nếu các tiêu đề bằng nhau, so sánh theo độ dài
            if (titleComparison == 0) {
                int lengthComparison = Integer.compare(otherDVD.getLength(), getLength());
                // Nếu độ dài bằng nhau, so sánh theo giá
                return (lengthComparison == 0) ? Float.compare(getCost(), otherDVD.getCost()) : lengthComparison;
            }
            return titleComparison;
        } else {
            // Với các đối tượng Media không phải DVD, sử dụng so sánh mặc định
            return super.compareTo(otherMedia);
        }
    }
}
```

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.
- Ta có thể sửa đổi quy tắc sắp xếp riêng trong lớp con như DVD để có thể áp dụng các quy tắc đặc biệt.