

Report OOP – Lab 04

Họ và tên: Bùi Quốc Bảo

MSSV: 20225601

Mã lớp: 744520

1. Lớp Book

```
import java.util.*;

public class Book {

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

    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 List<String> getAuthors() {
    return authors;
}

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

• public Book(String title) {
    this.title = title;
}

• public Book(String title, String category) {
    this.title = title;
    this.category = category;
}

• public Book(String title, String category, float cost) {
    this.title = title;
    this.category = category;
    this.cost = cost;
}

• public void addAuthor(String authorName) {

```

- Phương thức addAuthor và removeAuthor:

```

    public void addAuthor(String authorName) {
        if(!authors.contains(authorName)) {
            authors.add(authorName);
            System.out.println("Add author successfully!");
        } else {
            System.out.println("This author has already been in the list!");
        }
    }

    public void removeAuthor(String authorName) {
        if (authors.contains(authorName)) {
            authors.remove(authorName);
            System.out.println("Remove author successfully!");
        } else {
            System.out.println("No author has been found to remove!");
        }
    }
}

```

2. Lớp Media

```

1 public abstract class Media {
2
3     private static int nbMedia = 0;
4     private int id;
5     private String title;
6     private String category;
7     private float cost;
8
9
10
11     • public int getId() {
12         return id;
13     }
14
15     • public void setId(int id) {
16         this.id = id;
17     }
18
19     • public String getTitle() {
20         return title;
21     }
22
23     • public void setTitle(String title) {
24         this.title = title;
25     }
26
27     • public String getCategory() {
28         return category;
29     }
30
31     • public void setCategory(String category) {
32         this.category = category;
33     }
34
35     • public float getCost() {
36         return cost;
37     }
38
39     • public void setCost(float cost) {
40         this.cost = cost;
41     }
42
43 }

```

```

    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(String title) {
        this.title = title;
        this.id = ++nbMedia;
    }

    public Media(String title, String category) {
        this.title = title;
        this.category = category;
        this.id = ++nbMedia;
    }

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

```

- Đồng thời, xóa các trường và phương thức khỏi lớp Book và DigitalVideoDisc. Chuyển DigitalVideoDisc vào package hust.soict.ite6.aims.media và xóa package hust.soict.ite6.aims.disc
- Kế thừa lớp Media cho cả 2 lớp Book và DigitalVideoDisc.

```

public class Book extends Media {

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

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

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

    public Book(String title) {
        super(title);
    }

    public Book(String title, String category) {
        super(title, category);
    }

    public Book(String title, String category, float cost) {
        super(title, category, cost);
    }

    public void addAuthor(String authorName) {
        if(!authors.contains(authorName)) {
            authors.add(authorName);
            System.out.println("Add author successfully!");
        } else {
            System.out.println("This author has already been in the list!");
        }
    }

    public void removeAuthor(String authorName) {
        if (authors.contains(authorName)) {
            authors.remove(authorName);
            System.out.println("Remove author successfully!");
        } else {
            System.out.println("No author has been found to remove!");
        }
    }
}

```

```

public class DigitalVideoDisc extends Media {

    private String director;
    private int length;

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

    public boolean isMatch(String keyword)
    {
        return this.getTitle().toLowerCase().contains(keyword.toLowerCase());
    }

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

    @Override
    public String toString()
    {
        return "DVD: " + this.getTitle() +
            " - Category: " + this.getCategory() +
            " - Director: " + this.director +

```

3. Lớp CompactDisc

- Lớp Disc:

```

public class Disc extends Media {

    private String director;
    private int length;

    public String getDirector() {
        return director;
    }

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

    public int getLength() {
        return length;
    }

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

    public Disc(String title) {
        super(title);
    }

    public Disc(String title, String category) {
        super(title, category);
    }

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

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

    public Disc(String title, String category, String director, int length, float cost) {

```

- Lớp DigitalVideoDisc

```

public class DigitalVideoDisc extends Disc {

    public boolean isMatch(String keyword)
    {
        return this.getTitle().toLowerCase().contains(keyword.toLowerCase());
    }

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

    @Override
    public String toString()
    {
        return "DVD: " + this.getTitle() +
            " - Category: " + this.getCategory() +
            " - Director: " + this.getDirector() +
            " - DVD length: " + this.getLength() +
            " - Cost: " + this.getCost() + "$";
    }

}

```

- Lớp Track:

```

public class Track {

    private String title;
    private int length;

    public String getTitle() {
        return title;
    }

    public int getLength() {
        return length;
    }

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

}

```

- Lớp CompactDisc: Thêm trường artist và danh sách tracks. Tạo phương thức addTrack() và removeTrack(). Tạo phương thức getLength() để tính tổng độ dài các bài nhạc trong tracks của CD


```
import java.util.ArrayList;

public class CompactDisc extends Disc {

    private String artist;
    private ArrayList<Track> tracks;

    public String getArtist() {
        return artist;
    }

    public CompactDisc(String title) {
        super(title);
    }

    public CompactDisc(String title, String category) {
        super(title, category);
    }

    public CompactDisc(String title, String category, float cost) {
        super(title, category, cost);
    }

    public CompactDisc(String title, String category, String artist, float cost) {
        super(title, category, cost);
        this.artist = artist;
    }
}
```

```
public void addTrack(Track track) {
    if(!tracks.contains(track)) {
        tracks.add(track);
        System.out.println("Add track successfully!");
    } else {
        System.out.println("Track already exists in CD.");
    }
}

public void removeTrack(Track track) {
    if (tracks.contains(track)) {
        tracks.remove(track);
        System.out.println("Remove track successfully!");
    } else {
        System.out.println("Track does not exist in CD.");
    }
}

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

4. Lớp Playable

- Lớp Playable:

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

- Sau đó, Triển khai interface Playable và phương thức play() trong DigitalVideoDisc - Triển khai interface Playable và phương thức play() trong Track - Triển khai interface Playable và phương thức play() trong CompactDisc bằng implements:

```
public class DigitalVideoDisc extends Disc implements Playable {
```

```
3 public class Track implements Playable {
```

```
public class CompactDisc extends Disc implements Playable {
```

5. Cập nhật lớp Cart

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

    public int qtyOrdered = 0;

    public void addMedia(Media media) {
        if (itemsOrdered.size() >= MAX_NUMBERS_ORDERED) {
            System.out.println("The cart is almost full!");
        } else {
            itemsOrdered.add(media);
            System.out.println(media.getTitle() + "has been added!");
        }
    }

    public void removeMedia(Media media) {
        if (itemsOrdered.size() == 0) {
            System.out.println("Nothing to remove!");
        } else {
            if (itemsOrdered.remove(media)) {
                System.out.println(media.getTitle() + "has been remove from the cart.");
            } else {
                System.out.println("Media not found in cart!");
            }
        }
    }
}
```

- Phương thức totalCost:

```

public float totalCost() {
    float totalCost = 0;
    for (Media media : itemsOrdered) {
        totalCost += media.getCost();
    }
    return totalCost;
}

```

6. Cập nhật lớp Store

```

public class Store {

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

    public void addMedia(Media media) {
        if (itemsInStore.contains(media)) {
            System.out.println("The media " + media.getTitle() + " is already in the store!");
        } else {
            itemsInStore.add(media);
            System.out.println("The media " + media.getTitle() + " has been added to the store.");
        }
    }

    public void removeMedia(Media media) {
        if (itemsInStore.remove(media)) {
            System.out.println("The media " + media.getTitle() + " has been removed from the store.");
        } else {
            System.out.println("The media " + media.getTitle() + " is not in the store!");
        }
    }
}

```

7. Sắp xếp Media

- 2 lớp comparator:

```

import java.util.Comparator;

public class MediaComparatorByCostTitle implements Comparator<Media>{

    @Override
    public int compare(Media o1, Media o2) {

        int costComparison = Double.compare(o2.getCost(), o1.getCost());
        if (costComparison != 0) {
            return costComparison;
        }

        return o1.getTitle().compareTo(o2.getTitle());
    }
}

```

```
import java.util.Comparator;

public class MediaComparatorByTitleCost implements Comparator<Media> {

    @Override
    public int compare(Media o1, Media o2) {

        int titleComparison = o1.getTitle().compareTo(o2.getTitle());
        if (titleComparison != 0) {
            return titleComparison;
        }

        return Double.compare(o2.getCost(), o1.getCost());
    }
}
```

- Tạo MediaCompareTest để kiểm tra:

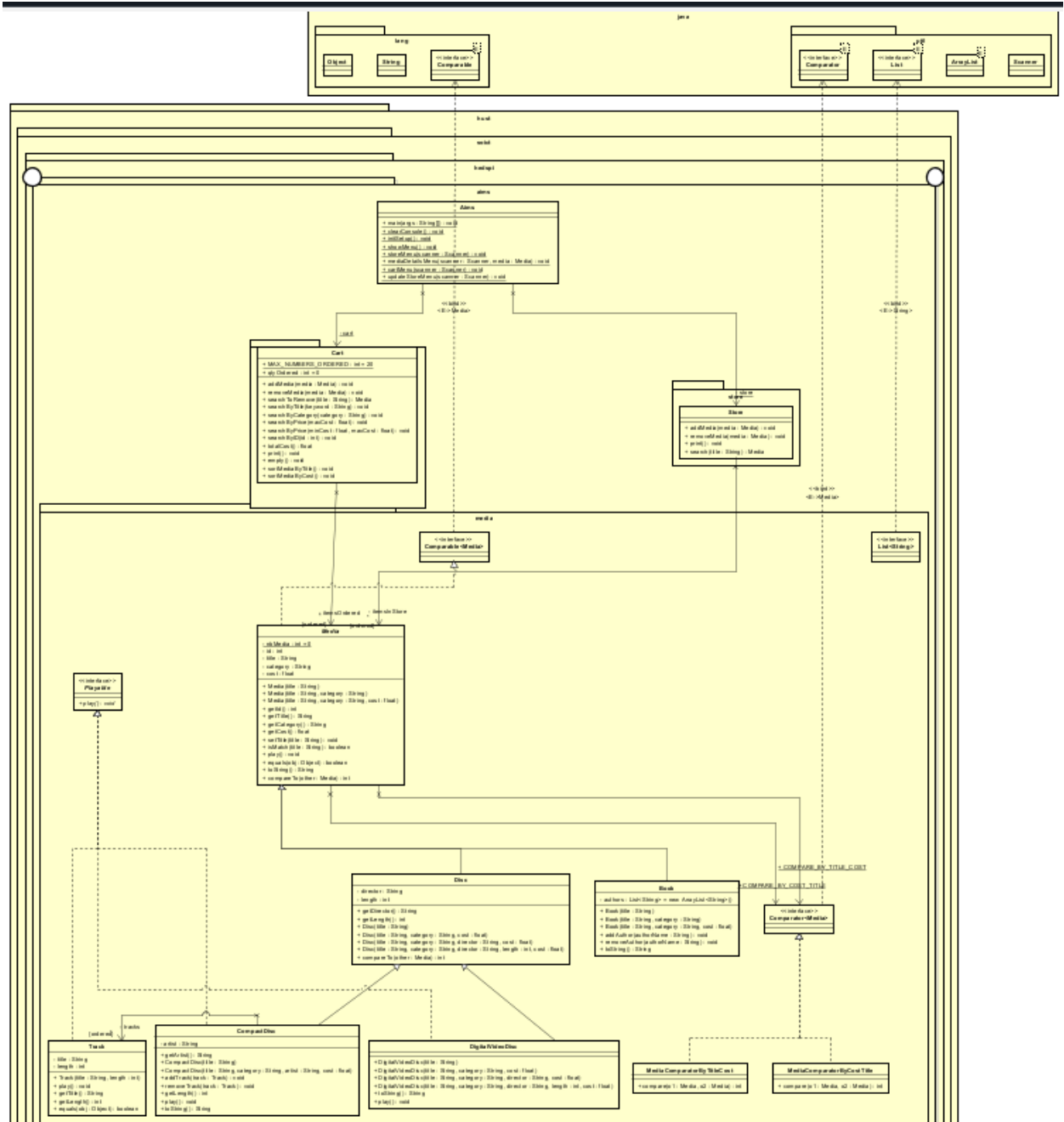
```
Iterator<Media> iterator = mediae.iterator();
// Sort by title using comparator
System.out.println();
System.out.println("*****SORT BY TITLE USING COMPARATOR*****");
Collections.sort((List<Media>)mediae, Media.COMPARE_BY_TITLE_COST);
iterator = mediae.iterator();

while (iterator.hasNext()) {
    System.out.println(((Media)iterator.next()).toString());
}
System.out.println("*****");

// Sort by cost using comparator
System.out.println();
System.out.println("*****SORT BY COST USING COMPARATOR*****");
Collections.sort((List<Media>)mediae, Media.COMPARE_BY_COST_TITLE);
iterator = mediae.iterator();

while (iterator.hasNext()) {
    System.out.println(((Media)iterator.next()).toString());
}
System.out.println("*****");
```

8. Cập nhật Class Diagram



9. Trả lời câu hỏi

- 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.
- What class should implement the Comparable interface?
The Media class should implement the Comparable interface.

-In those classes, how should you implement the compareTo() method to reflect the ordering that we want?

The implementation of the compareTo() method can be found in the src/hust/socit/hedspi/aims/Media.java file.

-Can we have two ordering rules for the items (e.g., by title then cost, and by cost then title) if we use this Comparable interface approach?

No, we cannot. The Comparable interface supports only one natural ordering for the objects being compared.

-Suppose the DVDs have a different ordering rule from other media types (by title, then decreasing length, then cost). How would you modify your code to allow this?

To handle this, you can override the compareTo() method in the Disc class to reflect the new ordering rule.

The modified implementation is available in the src/hust/soict/ite6/aims/Disc.java file.