

# Report OOP – Lab 04

**Họ và tên:** Nguyễn Kiều Duyên

**MSSV:** 20225618

**Mã lớp:** 744520      **Kỳ** 2024-1

## Table of Contents

I.	New written code	2
1.	Tạo lớp Book	2
2.	Tạo lớp trừu tượng Media	3
3.	Tạo lớp CompactDisc	6
4.	Tạo Interface Playable	9
5.	Cập nhật lớp Cart để làm việc với Media	10
6.	Cập nhật lớp Store để hoạt động với Media	11
7.	Cập nhật sơ đồ UML và Constructor cho các lớp cha và lớp con	11
8.	Đảm bảo phần tử duy nhất trong danh sách Media hoặc Track	13
9.	Tính đa hình với phương thức toString()	14
10.	Sắp xếp Media trong Cart	15
11.	Tạo Console hoàn chỉnh trong lớp Aims	17
II.	UML Diagram	26
1.	Use-Case Diagram	26
2.	Class Diagram (Final Lab04)	27
III.	Answer Question	28

# I. New written code

## 1. Tạo lớp Book

*Branch: topic/aims-project/add-book-class*

Tạo Class **Book** trong package **hust.soict.hedspi.aims.media** .

```
1 package hust.soict.hedspi.aims.media;
2
3 import java.util.*;
4
5 public class Book {
6
7     private int id;
8     private String title;
9     private String category;
10    private float cost;
11    private List<String> authors = new ArrayList<String>();
12
13
14    public int getId() {
15        return id;
16    }
17
18
19    public void setId(int id) {
20        this.id = id;
21    }
22
23
24    public String getTitle() {
25        return title;
26    }
27
28
29    public void setTitle(String title) {
30        this.title = title;
31    }
32
33
34    public String getCategory() {
35        return category;
36    }
37
38
39    public void setCategory(String category) {
40        this.category = category;
41    }
```

```
40        this.category = category;
41    }
42
43
44    public float getCost() {
45        return cost;
46    }
47
48
49    public void setCost(float cost) {
50        this.cost = cost;
51    }
52
53
54    public List<String> getAuthors() {
55        return authors;
56    }
57
58
59    public void setAuthors(List<String> authors) {
60        this.authors = authors;
61    }
62
63
64    public Book(String title) {
65        this.title = title;
66    }
67
68    public Book(String title, String category) {
69        this.title = title;
70        this.category = category;
71    }
72
73    public Book(String title, String category, float cost) {
74        this.title = title;
75        this.category = category;
76        this.cost = cost;
77    }
78
79    public void addAuthor(String authorName) {
80        if(!authors.contains(authorName)) {
```

Tạo các phương thức **addAuthor(String authorName)** và **removeAuthor(String authorName)** cho lớp **Book**:

```
79    public void addAuthor(String authorName) {
80        if(!authors.contains(authorName)) {
81            authors.add(authorName);
82            System.out.println("Add author successfully!");
83        } else {
84            System.out.println("This author has already been in the list!");
85        }
86    }
87
88    public void removeAuthor(String authorName) {
89        if (authors.contains(authorName)) {
90            authors.remove(authorName);
91            System.out.println("Remove author successfully!");
92        } else {
93            System.out.println("No author has been found to remove!");
94        }
95    }
96
97 }
98
```

## 2. Tạo lớp trừu tượng Media

*Branch: topic/aims-project/add-media-class*

Tạo lớp trừu tượng **Media** có chứa các thông tin chung của **DVD** và **Book**.

```
Media.java ×
1 package hust.soict.hedspi.aims.media;
2
3 public abstract class Media {
4
5     private static int nbMedia = 0;
6     private int id;
7     private String title;
8     private String category;
9     private float cost;
10
11
12     public int getId() {
13         return id;
14     }
15
16     public void setId(int id) {
17         this.id = id;
18     }
19
20     public String getTitle() {
21         return title;
22     }
23
24     public void setTitle(String title) {
25         this.title = title;
26     }
27
28     public String getCategory() {
29         return category;
30     }
31
32     public void setCategory(String category) {
33         this.category = category;
34     }
35
36     public float getCost() {
37         return cost;
38     }
39
40     public void setCost(float cost) {
41         this.cost = cost;
42     }
43
44     public Media(String title) {
45         this.title = title;
46         this.id = ++nbMedia;
47     }
48
49     public Media(String title, String category) {
50         this.title = title;
51         this.category = category;
52         this.id = ++nbMedia;
53     }
54
55     public Media(String title, String category, float cost) {
56         this.title = title;
57         this.category = category;
58         this.cost = cost;
59         this.id = ++nbMedia;
60     }
61
62 }
63
```

Đồng thời, xóa các trường và phương thức khỏi lớp **Book** và **DigitalVideoDisc**.

Chuyển **DigitalVideoDisc** vào package **hust.soict.hedspi.aims.media** và xóa package **hust.soict.dsai.aims.disc**

```
hust.soict.hedspi.aims.media
> Book.java
> DigitalVideoDisc.java
> Media.java
```

Kế thừa lớp **Media** cho cả 2 lớp **Book** và **DigitalVideoDisc**.

- Lớp **Book**:



```
Media.java Book.java ×
5 public class Book extends Media {
6
7     private List<String> authors = new ArrayList<String>();
8
9     public List<String> getAuthors() {
10         return authors;
11     }
12
13     public void setAuthors(List<String> authors) {
14         this.authors = authors;
15     }
16
17     public Book(String title) {
18         super(title);
19     }
20
21     public Book(String title, String category) {
22         super(title, category);
23     }
24
25     public Book(String title, String category, float cost) {
26         super(title, category, cost);
27     }
28
29     public void addAuthor(String authorName) {
30         if(!authors.contains(authorName)) {
31             authors.add(authorName);
32             System.out.println("Add author successfully!");
33         } else {
34             System.out.println("This author has already been in the list!");
35         }
36     }
37
38     public void removeAuthor(String authorName) {
39         if (authors.contains(authorName)) {
40             authors.remove(authorName);
41             System.out.println("Remove author successfully!");
42         } else {
43             System.out.println("No author has been found to remove!");
44         }
45     }
}
```

- Lớp DigitalVideoDisc

```
3 public class DigitalVideoDisc extends Media {
4
5     private String director;
6     private int length;
7
8     public String getDirector() {
9         return director;
10    }
11    public int getLength() {
12        return length;
13    }
14
15    public boolean isMatch(String keyword)
16    {
17        return this.getTitle().toLowerCase().contains(keyword.toLowerCase());
18    }
19
20    public DigitalVideoDisc(String title) {
21        super(title);
22    }
23    public DigitalVideoDisc(String title, String category, float cost) {
24        super(title, category, cost);
25    }
26    public DigitalVideoDisc(String title, String category, String director, float cost) {
27        super(title, category, cost);
28        this.director = director;
29    }
30    public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
31        super(title, category, cost);
32        this.director = director;
33        this.length = length;
34    }
35
36    @Override
37    public String toString()
38    {
39        return "DVD: " + this.getTitle() +
40            " - Category: " + this.getCategory() +
41            " - Director: " + this.director +
```

### 3. Tạo lớp CompactDisc

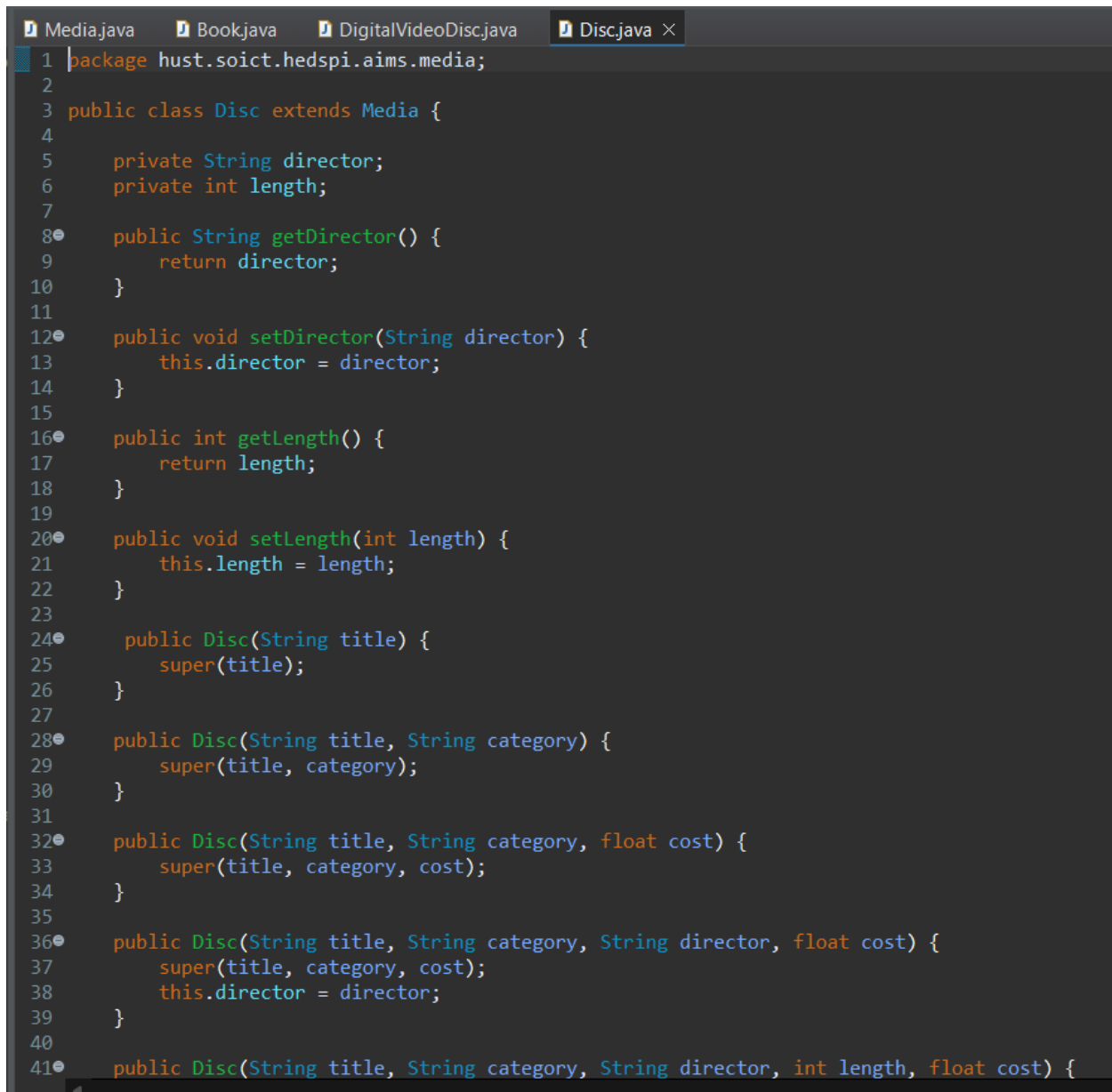
*Branch: topic/aims-project/create-CompactDisc-class*

#### 3.1. Tạo lớp Disc kế thừa từ Media

Tạo lớp **Disc** có 2 trường length và director.

Làm cho **DigitalVideoDisc** và **CompactDisc** kế thừa từ lớp **Disc**.

- Lớp Disc



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

## - Lớp DigitalVideoDisc

```
Media.java Book.java DigitalVideoDisc.java Disc.java
1 package hust.soict.hedspi.aims.media;
2
3 public class DigitalVideoDisc extends Disc {
4
5     public boolean isMatch(String keyword)
6     {
7         return this.getTitle().toLowerCase().contains(keyword.toLowerCase());
8     }
9
10    public DigitalVideoDisc(String title) {
11        super(title);
12    }
13    public DigitalVideoDisc(String title, String category, float cost) {
14        super(title, category, cost);
15    }
16    public DigitalVideoDisc(String title, String category, String director, float cost) {
17        super(title, category, director, cost);
18    }
19    public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
20        super(title, category, director, length, cost);
21    }
22
23    @Override
24    public String toString()
25    {
26        return "DVD: " + this.getTitle() +
27            " - Category: " + this.getCategory() +
28            " - Director: " + this.getDirector() +
29            " - DVD length: " + this.getLength() +
30            " - Cost: " + this.getCost() + "$";
31    }
32
33 }
```

### 3.2. Tạo lớp Track

Lớp **Track** mô phỏng một bài nhạc trên đĩa **CompactDisc**.

```
Media.java Book.java DigitalVideoDisc.java Disc.java Track.java
1 package hust.soict.hedspi.aims.media;
2
3 public class Track {
4
5     private String title;
6     private int length;
7
8     public String getTitle() {
9         return title;
10    }
11
12    public int getLength() {
13        return length;
14    }
15
16    public Track(String title, int length) {
17        this.title = title;
18        this.length = length;
19    }
20
21 }
22
```

### 3.3. Sửa 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.

```
Media.java Book.java DigitalVideoDisc.java Disc.java Track.java CompactDisc.java X
1 package hust.soict.hedspi.aims.media;
2
3 import java.util.ArrayList;
4
5 public class CompactDisc extends Disc {
6
7     private String artist;
8     private ArrayList<Track> tracks;
9
10    public String getArtist() {
11        return artist;
12    }
13
14    public CompactDisc(String title) {
15        super(title);
16    }
17
18    public CompactDisc(String title, String category) {
19        super(title, category);
20    }
21
22    public CompactDisc(String title, String category, float cost) {
23        super(title, category, cost);
24    }
25
26    public CompactDisc(String title, String category, String artist, float cost) {
27        super(title, category, cost);
28        this.artist = artist;
29    }
30
31    public void addTrack(Track track) {
32        if(!tracks.contains(track)) {
33            tracks.add(track);
34            System.out.println("Add track successfully!");
35        } else {
36            System.out.println("Track already exists in CD.");
37        }
38    }
39
40    public void removeTrack(Track track) {
41        if (tracks.contains(track)) {
42            tracks.remove(track);
43            System.out.println("Remove track successfully!");
44        } else {
45            System.out.println("Track does not exist in CD.");
46        }
47    }
48
49    public int getLength() {
50        int totalLength = 0;
51        for (Track track : tracks) {
52            totalLength += track.getLength();
53        }
54        return totalLength;
55    }
56
57 }
```



## 4. Tạo Interface Playable

*Branch: topic/aims-project/playable-interface*

Interface **Playable** cho phép các lớp chỉ định chúng triển khai phương thức **play()**

- Tạo interface **Playable** và thêm phương thức nguyên mẫu:

```
1 package hust.soict.hedspi.aims.media;
2
3 public interface Playable {
4
5     public void play();
6
7 }
```

- Triển khai interface **Playable** và phương thức **play()** trong **DigitalVideoDisc**

```
2
3 public class DigitalVideoDisc extends Disc implements Playable {
4
5     public void play() {
6         System.out.println("Playing DVD: " + this.getTitle());
7         System.out.println("DVD length: " + this.getLength());
8     }
9 }
```

- Triển khai interface **Playable** và phương thức **play()** trong **Track**

```
3 public class Track implements Playable {
4
5     public void play() {
6         System.out.println("Playing track: " + this.getTitle());
7         System.out.println("Track length: " + this.getLength());
8     }
9 }
```

- Triển khai interface **Playable** và phương thức **play()** trong **CompactDisc**

```
4
5 public class CompactDisc extends Disc implements Playable {
6
7     public void play() {
8         System.out.println("Playing CD: " + this.getTitle());
9         System.out.println("CD length: " + this.getLength());
10        for (Track track : tracks) {
11            track.play();
12        }
13    }
14 }
```

## 5. Cập nhật lớp Cart để làm việc với Media

*Branch: feature/aims-project/update-cart-to-work-with-media*

Xóa các phương thức **addDigitalVideoDisc()** và **removeDigitalVideoDisc()** để tạo các phương thức tổng quát hơn là **addMedia()** và **removeMedia()**.

```
1 package hust.soict.hedspi.aims.cart;
2
3 import java.util.*;
4 import hust.soict.hedspi.aims.media.Media;
5
6 public class Cart {
7
8     public static final int MAX_NUMBERS_ORDERED = 20;
9     private ArrayList<Media> itemsOrdered = new ArrayList<Media>();
10
11     public int qtyOrdered = 0;
12
13     public void addMedia(Media media) {
14         if (itemsOrdered.size() >= MAX_NUMBERS_ORDERED) {
15             System.out.println("The cart is almost full!");
16         } else {
17             itemsOrdered.add(media);
18             System.out.println(media.getTitle() + "has been added!");
19         }
20     }
21
22     public void removeMedia(Media media) {
23         if (itemsOrdered.size() == 0) {
24             System.out.println("Nothing to remove!");
25         } else {
26             if (itemsOrdered.remove(media)) {
27                 System.out.println(media.getTitle() + "has been remove from the cart.");
28             } else {
29                 System.out.println("Media not found in cart!");
30             }
31         }
32     }
33 }
```

Cập nhật **totalCost()** để tính tổng chi phí tất cả Media

```
34     public float totalCost() {
35         float totalCost = 0;
36         for (Media media : itemsOrdered) {
37             totalCost += media.getCost();
38         }
39         return totalCost;
40     }
```

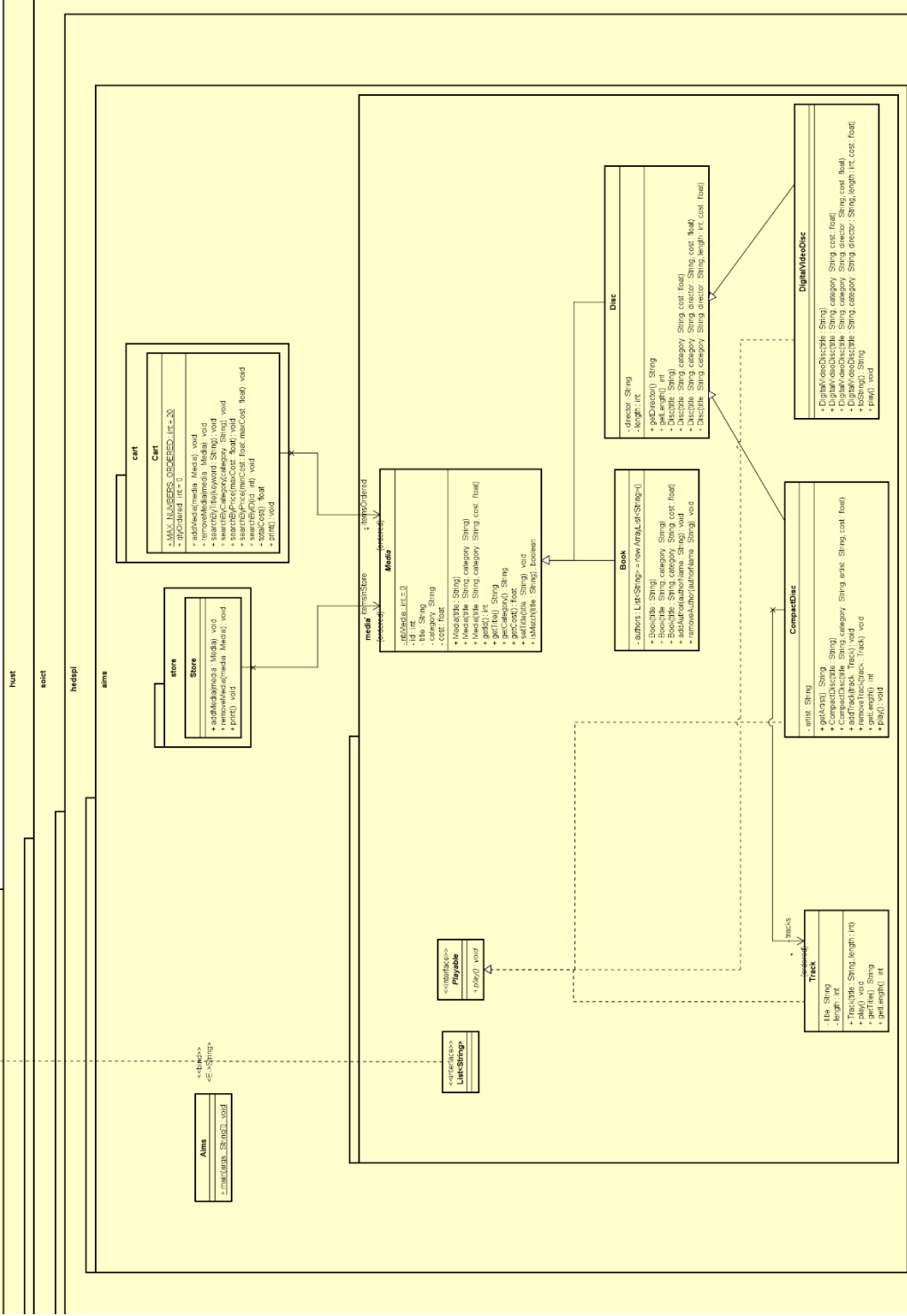
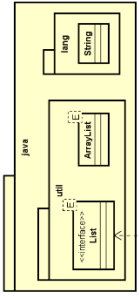
## 6. Cập nhật lớp Store để hoạt động với Media

*Branch: feature/aims-project/update-store-to-work-with-media*

```
Store.java ×
1 package hust.soict.hedspi.aims.store;
2 import java.util.ArrayList;
3
4
5
6
7 public class Store {
8
9     private ArrayList<Media> itemsInStore = new ArrayList<Media>();
10
11     public void addMedia(Media media) {
12         if (itemsInStore.contains(media)) {
13             System.out.println("The media " + media.getTitle() + " is already in the store!");
14         } else {
15             itemsInStore.add(media);
16             System.out.println("The media " + media.getTitle() + " has been added to the store.");
17         }
18     }
19
20     public void removeMedia(Media media) {
21         if (itemsInStore.remove(media)) {
22             System.out.println("The media " + media.getTitle() + " has been removed from the store.");
23         } else {
24             System.out.println("The media " + media.getTitle() + " is not in the store!");
25         }
26     }
27 }
```

## 7. Cập nhật sơ đồ UML và Constructor cho các lớp cha và lớp con

- Đã constructor ở các bước trước.
- Chỉ xóa những setter không cần thiết.
- Class Diagram UML



## 8. Đảm bảo phần tử duy nhất trong danh sách Media hoặc Track

*Branch: feature/aims-project/override-equals-method*

Ghi đè phương thức `equals(Object o)`

- Lớp **Media**

```
56  @Override
57  public boolean equals(Object obj) {
58      if (obj == this) {
59          return true;
60      }
61      if (!(obj instanceof Media)) {
62          return false;
63      }
64      return ((Media)obj).getTitle() == this.getTitle();
65  }
```

- Lớp **Track**

```
26  @Override
27  public boolean equals(Object obj) {
28      if (obj == this) {
29          return true;
30      }
31      if (!(obj instanceof Track)) {
32          return false;
33      }
34      return ((Track)obj).getTitle() == this.getTitle() && ((Track)obj).getLength() == this.getLength();
35  }
```

- Cập nhật lại lớp **Cart** để kiểm tra phần tử trùng lặp

```
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!");
    }
}
```

- Cập nhật lớp **CartTest** để kiểm tra

```
3      DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin",
4          "Animation", 18.99f);
5      cart.addMedia(dvd3);
6
7      DigitalVideoDisc dvd4 = new DigitalVideoDisc("Aladin",
8          "Animation", 18.99f);
9      cart.addMedia(dvd4);
10
```

## 9. Tính đa hình với phương thức toString()

**Branch: *feature/aims-project/polymorphism-with-toString-method***

- Tùy vào từng loại đối tượng, override phương thức **toString()** thì sẽ hiện ra thông tin khác nhau tùy thuộc vào **toString()** tương ứng.

### Lớp Book

```
@Override
public String toString() {
    return "Book: " + this.getTitle() +
        " - Category: " + this.getCategory() +
        " - Cost: " + this.getCost() + "$";
}
```

### Lớp Media

```
@Override
public String toString() {
    return "Media: " + this.getTitle() +
        " - Category: " + this.getCategory() +
        " - Cost: " + this.getCost() + "$";
}
```

### Lớp DigitalVideoDisc

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

### Lớp CompactDisc

```
@Override
public String toString() {
    return "CD: " + this.getTitle() +
        " - Category: " + this.getCategory() +
        " - Artist: " + this.getArtist() +
        " - Length: " + this.getLength() +
        " - Cost: " + this.getCost() + "$";
}
```

- Kết quả chạy lớp **CartTest**

```
public class CartTest {
    public static void main(String[] args) {

        Cart cart = new Cart();

        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
        cart.addMedia(dvd1);

        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star War", "Science Fiction", "George Lucas", 87, 24.95f);
        cart.addMedia(dvd2);

        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
        cart.addMedia(dvd3);

        DigitalVideoDisc dvd4 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
        cart.addMedia(dvd4);

        cart.print();

        cart.searchByID(3);
        cart.searchByTitle("Lion");
    }
}
```

```
The Lion Kinghas been added!
Star Warhas been added!
Aladinhas been added!
Aladin is already in the cart!
*****CART*****
Ordered Items:
1.DVD: The Lion King - Category: Animation - Director: Roger Allers - DVD length: 87 - Cost: 19.95$
2.DVD: Star War - Category: Science Fiction - Director: George Lucas - DVD length: 87 - Cost: 24.95$
3.DVD: Aladin - Category: Animation - Director: null - DVD length: 0 - Cost: 18.99$
Total cost: 63.89
*****
FoundDVD: Aladin - Category: Animation - Director: null - DVD length: 0 - Cost: 18.99$
FoundDVD: The Lion King - Category: Animation - Director: Roger Allers - DVD length: 87 - Cost: 19.95$
```

## 10. Sắp xếp Media trong Cart

*Branch: feature/aims-project/sort-media-in-cart-with-comparator*

Sử dụng **Comparator**

- Tạo 2 lớp **Comparator**

```
MediaComparatorByCostTitle.java ×
1 package hust.soict.hedspi.aims.media;
2
3 import java.util.Comparator;
4
5 public class MediaComparatorByCostTitle implements Comparator<Media>{
6
7     @Override
8     public int compare(Media o1, Media o2) {
9
10         int costComparison = Double.compare(o2.getCost(), o1.getCost());
11         if (costComparison != 0) {
12             return costComparison;
13         }
14
15         return o1.getTitle().compareTo(o2.getTitle());
16     }
17 }
```

```
MediaComparatorByCostTitle.java  MediaComparatorByTitleCost.java ×
1 package hust.soict.hedspi.aims.media;
2
3 import java.util.Comparator;
4
5 public class MediaComparatorByTitleCost implements Comparator<Media> {
6
7     @Override
8     public int compare(Media o1, Media o2) {
9
10         int titleComparison = o1.getTitle().compareTo(o2.getTitle());
11         if (titleComparison != 0) {
12             return titleComparison;
13         }
14
15         return Double.compare(o2.getCost(), o1.getCost());
16     }
17
18 }
```

- Thêm các **Comparator** làm thuộc tính của lớp **Media**

```
5 public abstract class Media implements Comparable<Media>{
6
7     public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost();
8     public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle();
9 }
```

- Tạo lớp **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("*****");
```

Phần so sánh **Comparator** và interface **Comparable** sẽ được triển khai ở phần [III. Answer Question](#)



## 11. Tạo Console hoàn chỉnh trong lớp Aims

**Branch: topic/aims-project/complete-console-application**

- Hàm **main**

- Clear Console

```
// Clear Console
public static void clearConsole() {
    for (int i=0; i<50; i++)
    {
        System.out.println();
    }
}
```

- Khởi tạo Store gồm các Media (CD-Track, DVD, Book)

```
public static void initSetup() {

    DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
    DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star War", "Science Fiction", "George Lucas", 87, 24.95f);
    DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
    store.addMedia(dvd1);
    store.addMedia(dvd2);
    store.addMedia(dvd3);

    Book book = new Book("The Valley of Fear", "Detective", 20.00f);
    Book book1 = new Book("A Living Remedy: A Memoir", "Biography", 202.00f);
    Book book2 = new Book("On the Origin of Time: Stephen Hawking's Final Theory", "Science", 120.00f);
    store.addMedia(book);
    store.addMedia(book1);
    store.addMedia(book2);

    CompactDisc cd1 = new CompactDisc("Adele - 30", "Music", "Adele", 1500.98f);
    Track track1CD1 = new Track("All Night Parking (interlude)", 161);
    Track track2CD1 = new Track("To Be Loved", 403);
    Track track3CD1 = new Track("Woman Like Me", 300);
    cd1.addTrack(track1CD1);
    cd1.addTrack(track2CD1);
    cd1.addTrack(track3CD1);

    CompactDisc cd2 = new CompactDisc("The Gods We Can Touch", "Music", "Aurora", 2000.22f);
    Track track1CD2 = new Track("Everything Matters", 180+34);
    Track track2CD2 = new Track("Blood in the Wine", 180+30);
    Track track3CD2 = new Track("Artemis", 60*2+39);
    cd2.addTrack(track1CD2);
    cd2.addTrack(track2CD2);
    cd2.addTrack(track3CD2);

    CompactDisc cd3 = new CompactDisc("Purpose", "Music", "Justin Bieber", 1000.98f);
    Track track1CD3 = new Track("The Feeling", 4*60+5);
    Track track2CD3 = new Track("No Sense", 4*60+35);
    cd3.addTrack(track1CD3);
    cd3.addTrack(track2CD3);

    store.addMedia(cd1);
    store.addMedia(cd2);
    store.addMedia(cd3);

    clearConsole();
}
```

- Menu chính được gọi trong hàm main()

```
// Show menu
public static void showMenu() {
    System.out.println("AIMS: ");
    System.out.println("-----");
    System.out.println("1. View store");
    System.out.println("2. Update store");
    System.out.println("3. See current cart");
    System.out.println("0. Exit");
    System.out.println("-----");
    System.out.println("Please choose a number: 0-1-2-3");
}
```

- Menu Store khi ấn vào View Store

```
public static void storeMenu(Scanner scanner) {
    boolean back = false;
    while(back == false)
    {
        store.print();

        System.out.println("Options: ");
        System.out.println("-----");
        System.out.println("1. See a media's details");
        System.out.println("2. Add a media to cart");
        System.out.println("3. Play a media");
        System.out.println("4. See current cart");
        System.out.println("0. Back");
        System.out.println("-----");
        System.out.println("Please choose a number: 0-1-2-3-4");

        int option = scanner.nextInt();
        scanner.nextLine();

        switch (option) {
            case 0:
                clearConsole();
                back = true;
                break;
            case 1:
                boolean foundDetails = false;
                while (foundDetails == false) {
                    System.out.println("Enter the title of the media (type 0 to stop): ");
                    String title = scanner.nextLine();
                    if (title.equals("0")) {
                        clearConsole();
                        break;
                    }
                    Media media = store.search(title);
                    if (media != null) {
                        clearConsole();
                        System.out.println("Details: ");
                        System.out.println(media);
                        mediaDetailsMenu(scanner, media);
                        foundDetails = true;
                    } else {
                        System.out.println("***MEDIA NOT FOUND***");
                    }
                }
                break;
        }
    }
}
```

```

case 2:
    boolean foundToAdd = false;
    while (foundToAdd == false) {
        System.out.println("Enter the title of the media (type 0 to stop): ");
        String title = scanner.nextLine();
        if (title.equals("0")) {
            clearConsole();
            break;
        }
        Media media = store.search(title);
        if (media != null) {
            cart.addMedia(media);
            foundToAdd = true;
        } else {
            System.out.println("***MEDIA NOT FOUND***");
        }
    }
    break;
case 3:
    boolean foundToPlay = false;
    while (foundToPlay == false) {
        System.out.println("Enter the title of the media (type 0 to stop): ");
        String title = scanner.nextLine();
        if (title.equals("0")) {
            clearConsole();
            break;
        }
        Media media = store.search(title);
        if (media != null) {
            if (media instanceof Disc || media instanceof CompactDisc) {
                media.play();
            } else {
                System.out.println("This type of media is not supported!");
            }
            foundToPlay = true;
        } else {
            System.out.println("***MEDIA NOT FOUND***");
        }
    }
    break;

```

```

case 4:
    clearConsole();
    cartMenu(scanner);
    break;
default:
    clearConsole();
    System.out.println("Invalid option, please choose again.");
    break;
}
}
}

```

- Tùy chọn **See a media's details**

```
// Media Details Menu
public static void mediaDetailsMenu(Scanner scanner, Media media) {
    boolean back = false;
    while (back == false) {
        System.out.println("Options: ");
        System.out.println("-----");
        System.out.println("1. Add to cart");
        System.out.println("2. Play");
        System.out.println("0. Back");
        System.out.println("-----");
        System.out.println("Please choose a number: 0-1-2");

        int option = scanner.nextInt();
        scanner.nextLine();

        switch (option) {
            case 0:
                clearConsole();
                back = true;
                break;
            case 1:
                cart.addMedia(media);
                break;
            case 2:
                if (media instanceof Disc || media instanceof CompactDisc) {
                    media.play();
                } else {
                    System.out.println("This type of media is not supported!");
                }
                break;
            default:
                clearConsole();
                System.out.println("Invalid option, please choose again.");
                break;
        }
    }
}
```

- Từ menu chính chọn **See current cart**

```
// CartMenu
public static void cartMenu(Scanner scanner) {
    boolean back = false;
    while (back == false) {
        cart.print();
        System.out.println("Options: ");
        System.out.println("-----");
        System.out.println("1. Filter medias in cart");
        System.out.println("2. Sort medias in cart");
        System.out.println("3. Remove media from cart");
        System.out.println("4. Play a media");
        System.out.println("5. Place order");
        System.out.println("0. Back");
        System.out.println("-----");
        System.out.println("Please choose a number: 0-1-2-3-4-5");
        int option = scanner.nextInt();
        scanner.nextLine();
        switch (option) {
            case 0:
                clearConsole();
                back = true;
                break;
```

```
            case 1:
                System.out.println("Filter medias in cart by (1) id or (2) title:");
                int filterOption = scanner.nextInt();
                scanner.nextLine();
                boolean found = false;
                while (found == false) {
                    if (filterOption == 1) {
                        System.out.println("Enter the id to filter (type 0 to stop):");
                        int id = scanner.nextInt();
                        scanner.nextLine();
                        if (id == 0) {
                            clearConsole();
                            break;
                        }
                        cart.searchByID(id);
                        found = true;
                    } else if (filterOption == 2) {
                        System.out.println("Enter the title to filter (type 0 to stop):");
                        String title = scanner.nextLine();
                        if (title.equals("0")) {
                            clearConsole();
                            break;
                        }
                        cart.searchByTitle(title);
                        found = true;
                    } else if (filterOption == 0) {
                        clearConsole();
                        break;
                    } else {
                        System.out.println("Invalid option.");
                    }
                }
                break;
```

```

        case 2:
            System.out.println("Sort medias in cart by (1) title or (2) cost:");
            int sortOption = scanner.nextInt();
            scanner.nextLine();
            if (sortOption == 1) {
                cart.sortMediaByTitle();
            } else if (sortOption == 2) {
                cart.sortMediaByCost();
            } else {
                System.out.println("Invalid option.");
            }
            break;
        case 3:
            boolean foundToRemove = false;
            while (foundToRemove == false) {
                System.out.println("Enter the title of the media (type 0 to stop): ");
                String title = scanner.nextLine();
                if (title.equals("0")) {
                    clearConsole();
                    break;
                }
                Media media = cart.searchToRemove(title);
                if (media != null) {
                    clearConsole();
                    cart.removeMedia(media);
                    foundToRemove = true;
                } else {
                    System.out.println("***MEDIA NOT FOUND***");
                }
            }
            break;

```

```

        case 4:
            boolean foundToPlay = false;
            while (foundToPlay == false) {
                System.out.println("Enter the title of the media (type 0 to stop): ");
                String title = scanner.nextLine();
                if (title.equals("0")) {
                    clearConsole();
                    break;
                }
                Media media = store.search(title);
                if (media != null) {
                    if (media instanceof Disc || media instanceof CompactDisc) {
                        media.play();
                    } else {
                        System.out.println("This type of media is not supported!");
                    }
                    foundToPlay = true;
                } else {
                    System.out.println("***MEDIA NOT FOUND***");
                }
            }
            break;
        case 5:
            clearConsole();
            cart.empty();
            break;
        default:
            clearConsole();
            System.out.println("Invalid option, please choose again.");
            break;
    }
}
}
}

```

- Khi người dùng chọn **update store**

```
// Update Store Menu
public static void updateStoreMenu(Scanner scanner) {
    boolean back = false;
    while (!back) {
        System.out.println("Options: ");
        System.out.println("-----");
        System.out.println("1. Add a media");
        System.out.println("2. Remove a media");
        System.out.println("0. Back");
        System.out.println("-----");
        System.out.println("Please choose a number: 0-1-2");
        int option = scanner.nextInt();
        scanner.nextLine();
        switch (option) {
            case 0:
                clearConsole();
                back = true;
                break;
```

```
        case 1:
            System.out.println("Enter the category of the media (1) Book, (2) CD, (3) DVD or (0) exit:");
            int categoryChoice = scanner.nextInt();
            scanner.nextLine();

            if (categoryChoice == 1) {
                System.out.println("Enter book title: ");
                String bookTitle = scanner.nextLine();
                System.out.println("Enter book category: ");
                String bookCategory = scanner.nextLine();
                System.out.println("Enter book cost: ");
                Float bookCost = scanner.nextFloat();
                scanner.nextLine();

                Book newBook = new Book(bookTitle, bookCategory, bookCost);
                store.addMedia(newBook);
            } else if (categoryChoice == 2) {
                System.out.println("Enter CD title: ");
                String cdTitle = scanner.nextLine();
                System.out.println("Enter CD category: ");
                String cdCategory = scanner.nextLine();
                System.out.println("Enter CD artist: ");
                String cdArtist = scanner.nextLine();
                System.out.println("Enter CD cost: ");
                Float cdCost = scanner.nextFloat();
                scanner.nextLine();

                CompactDisc newCD = new CompactDisc(cdTitle, cdCategory, cdArtist, cdCost);

                System.out.println("Do you want to add tracks to your CD? (1) Yes (0) No:");
                int addTrack = scanner.nextInt();
                scanner.nextLine();
```



```

        if (addTrack == 1) {
            System.out.println("How many tracks in your CD?");
            int numTrack = scanner.nextInt();
            scanner.nextLine();
            for (int i = 0; i < numTrack; i++) {
                System.out.println("Your " + (i+1) + " track: ");
                System.out.println("Enter track title: ");
                String trackTitle = scanner.nextLine();
                System.out.println("Enter track length: ");
                int trackLength = scanner.nextInt();
                scanner.nextLine();

                Track newTrack = new Track(trackTitle, trackLength);
                newCD.addTrack(newTrack);
            }
            store.addMedia(newCD);
        } else if (addTrack == 0) {
            store.addMedia(newCD);
        }
    }
} else if (categoryChoice == 3) {
    System.out.println("Enter DVD title: ");
    String dvdTitle = scanner.nextLine();
    System.out.println("Enter DVD category: ");
    String dvdCategory = scanner.nextLine();
    System.out.println("Enter book cost: ");
    Float dvdCost = scanner.nextFloat();
    scanner.nextLine();

    DigitalVideoDisc newDVD = new DigitalVideoDisc(dvdTitle, dvdCategory, dvdCost);
    store.addMedia(newDVD);
} else if (categoryChoice == 0) {
    clearConsole();
    break;
} else {
    System.out.println("Invalid option.");
}
break;

```

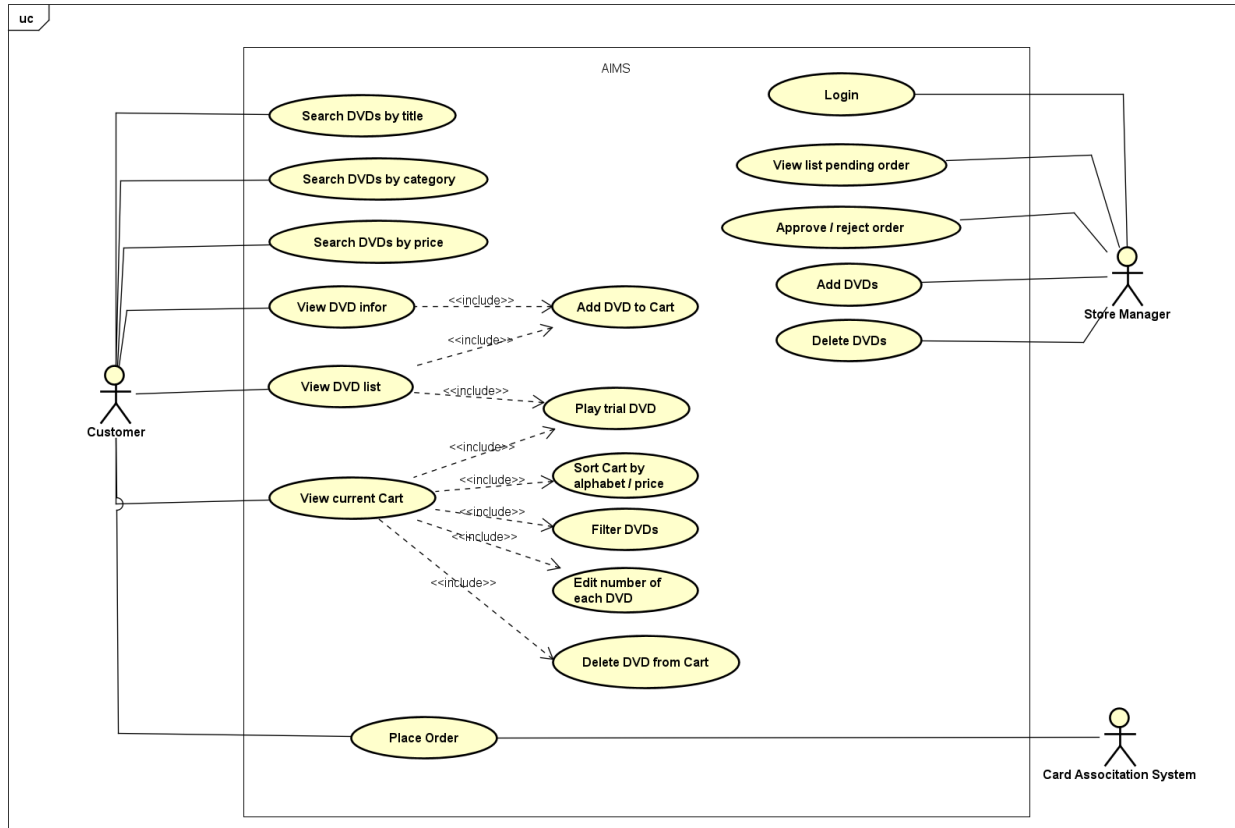
```

case 2:
    boolean foundToRemove = false;
    while (!foundToRemove) {
        System.out.println("Enter the title of the media (type 0 to stop): ");
        String titleForRemove = scanner.nextLine();
        if (titleForRemove.equals("0")) {
            clearConsole();
            break;
        }
        Media media = store.search(titleForRemove);
        if (media != null) {
            clearConsole();
            store.removeMedia(media);
            foundToRemove = true;
        } else {
            System.out.println("***MEDIA NOT FOUND***");
        }
    }
    break;
default:
    clearConsole();
    System.out.println("Invalid option, please choose again.");
    break;
}
}
}
}

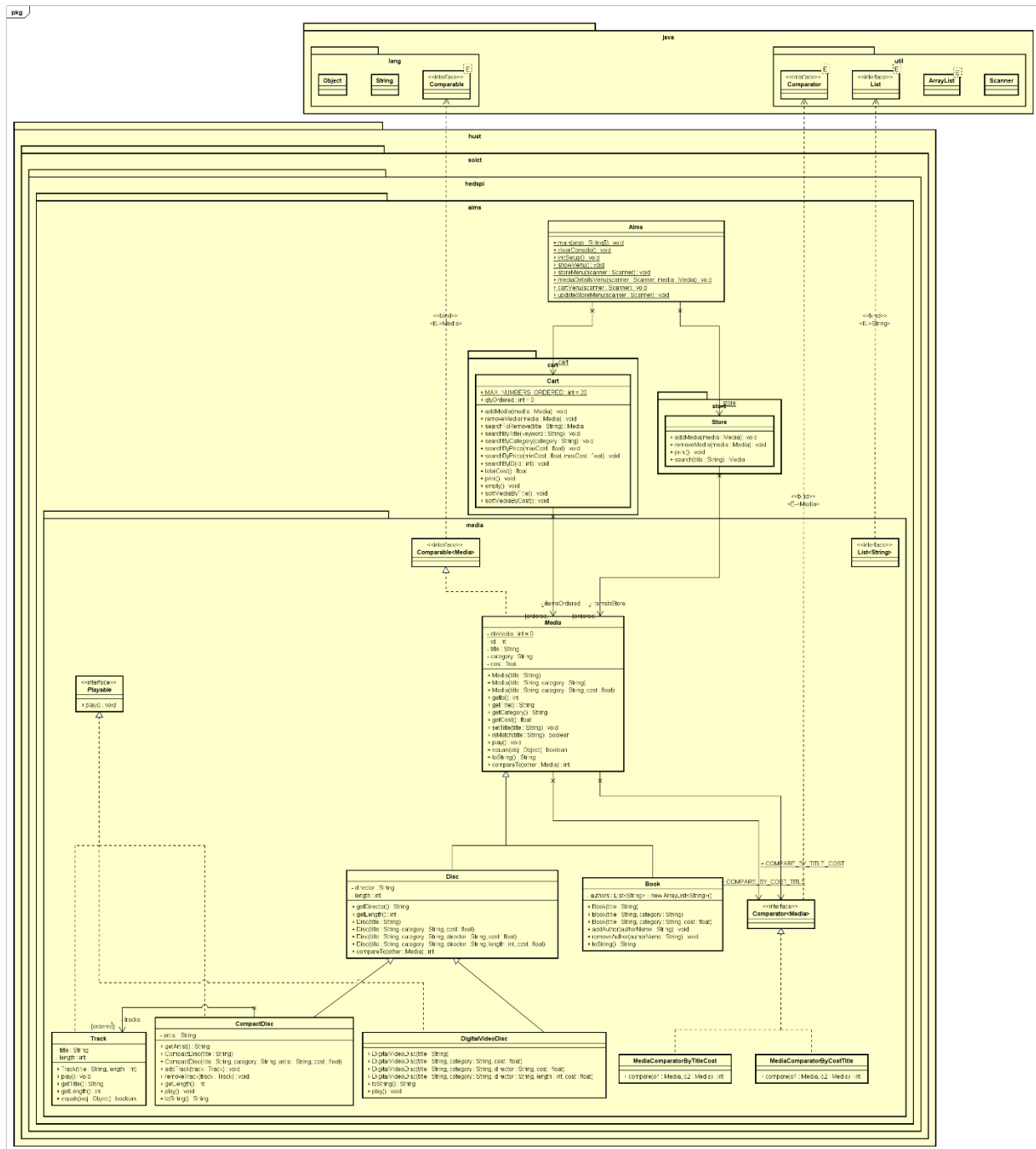
```

## II. UML Diagram

### 1. Use-Case Diagram



## 2. Class Diagram (Final Lab04)



### III. Answer Question

**Question:** 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?*

→ **Media** class should implement the **Comparable** interface

```
public abstract class Media implements Comparable<Media>{
```

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

The answer can be found in `src/hust/socit/hedspi/aims/Media.java` file.

```
@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. *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?*

No we cannot. The **Comparable** interface assumes that there is only one natural ordering for the objects being compared.

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?

We can override the **compareTo()** method in **Disc** class to reflect the new ordering rule.

The modified code can be found in `src/hust/soict/hedspi/aims/Disc.java` file

```
@Override
public int compareTo(Media other) {
    if (other instanceof Disc) {
        Disc otherDVD = (Disc) other;
        int titleComparison = this.getTitle().compareTo(otherDVD.getTitle());
        if (titleComparison != 0) {
            // Compare by title
            return titleComparison;
        } else {
            // Compare by decreasing length
            int lengthComparison = Integer.compare(otherDVD.getLength(), this.getLength());
            if (lengthComparison != 0) {
                return lengthComparison;
            } else {
                // Compare by cost
                return Double.compare(this.getCost(), otherDVD.getCost());
            }
        }
    } else {
        // If the media object is not a Disc, use the default method of the Media class
        return super.compareTo(other);
    }
}
```

MediaCompareTest

```
// Sort using override compareto method
System.out.println();
System.out.println("*****SORT USING OVERRIDE compareTo method*****");
Collections.sort(mediae);
Iterator<Media> iterator = mediae.iterator();
while (iterator.hasNext()) {
    System.out.println(((Media)iterator.next()).toString());
}

System.out.println("*****");
```