

BÁO CÁO THỰC HÀNH

Lab04: Inheritance Polymorphism

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Mục lục

1. Import the existing project into the workspace of IntelliJ	2
2. Additional requirements of AIMS	2
3. Creating the Book class	2
4. Creating the abstract Media class	4
5. Creating the CompactDisc Class	9
5.1 . Create the Disc class extending the Media Class.....	9
5.2 . Create the <i>Track</i> class which models a track on a compact disc and will store information including the <i>title</i> and <i>length</i> of the track	11
5.3 . Open the CompactDisc.....	12
6. Create the Playable interface	14
7. Update the Cart class to work with Media	15
8. Update the Store class to work with Media.....	17
9. Constructor of whole classes and parent classes	19
10. Unique item in a list	20
11. Polymorphism with toString() method.....	21
12. Sort media in the cart.....	22
13. Create a complete console application in the Aims class.....	23

1. Import the existing project into the workspace of IntelliJ
2. Additional requirements of AIMS
3. Creating the Book class

```
© Cart.java ×  © Aims.java  © Book.java ×  © DigitalVideoDisc.java
1  package hust.soict.dsai.aims.media;
2
3  import java.util.ArrayList;
4  import java.util.List;
5
6  public class Book {
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     public int getID(){
14         return id;
15     }
16
17     public void setID(int id){
18         this.id = id;
19     }
20
21     public String getTitle(){
22         return title;
23     }
24
25     public void setTitle(String title){
26         this.title = title;
27     }
28
29     public String getCategory(){
30         return category;
```

```
29     public String getCategory(){
30         return category;
31     }
32
33     public void setCategory(String category){
34         this.category = category;
35     }
36
37     public float getCost(){
38         return cost;
39     }
40
41     public void setCost(float cost){
42         this.cost = cost;
43     }
```

```
45     public List<String> getAuthors(){
46         return authors;
47     }
48
49     public void setAuthors(List<String> authors){
50         this.authors = authors;
51     }
52
53     private Book(){
54         //TODO Auto-generated constructor stub
55         super();
56     }
57
58     private void addAuthor(String author){
59         for (String s: this.authors){
60             if (s == author){
61                 System.out.println("The author already exists!");
62                 return;
63             }
64         }
65         this.authors.add(author);
66         System.out.println("The author has been added!");
67     }
68 }
```

```
70     public void removeAuthor(String author){
71         boolean check = false;
72         for (String s: this.authors){
73             if (s == author){
74                 check = true;
75                 break;
76             }
77         }
78         if (check){
79             this.authors.remove(author);
80             System.out.println("The author has been removed!");
81         }
82         else{
83             System.out.println("The author does not exist!");
84         }
85     }
86 }
87 }
```

4. Creating the abstract Media class

- **Book class:**

```
Aims.java    © Book.java ×    © DigitalVideoDisc.java    © TestPassingParameter.java    © Di
1  package hust.soict.dsai.aims.media;
2
3  import java.util.ArrayList;
4  import java.util.List;
5
6  public class Book {
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     public int getID(){
14         return id;
15     }
16
17     public void setID(int id){
18         this.id = id;
19     }
20
21     public String getTitle(){
22         return title;
23     }
24
25     public void setTitle(String title){
26         this.title = title;
27     }
28
29     public String getCategory(){
30         return category;
31     }
32
33     public void setCategory(String category){
34         this.category = category;
35     }
36
37     public float getCost(){
38         return cost;
39     }
40
41     public void setCost(float cost){
42         this.cost = cost;
43     }
```

```
45     public List<String> getAuthors(){
46         return authors;
47     }
48
49     public void setAuthors(List<String> authors){
50         this.authors = authors;
51     }
52
53     private Book(){
54         //TODO Auto-generated constructor stub
55         super();
56     }
57
58     private void addAuthor(String author){
59         for (String s: this.authors){
60             if (s == author){
61                 System.out.println("The author already exists!");
62                 return;
63             }
64         }
65         this.authors.add(author);
66         System.out.println("The author has been added!");
67     }
68 }
```

```
69
70     public void removeAuthor(String author){
71         boolean check = false;
72         for (String s: this.authors){
73             if (s == author){
74                 check = true;
75                 break;
76             }
77         }
78         if (check){
79             this.authors.remove(author);
80             System.out.println("The author has been removed!");
81         }
82         else{
83             System.out.println("The author does not exist!");
84         }
85     }
86 }
87 }
```

- **Media class:**

```
Cart.java  Media.java ×  Aims.java  Book.java  DigitalVideoDisc.java

1  package hust.soict.dsai.aims.media;
2
3  import javax.swing.undo.AbstractUndoableEdit;
4
5  1 inheritor
6  public abstract class Media {
7      private int id;
8      private String title;
9      private String category;
10     private float cost;
11
12     public Media(){
13         //TODO Auto-generated constructor stub
14     }
15
16     public int getId(){
17         return id;
18     }
19
20     public void setID(int id){
21         this.id = id;
22     }
23
24     public String getTitle(){
25         return 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 }
```

```
43     public Media(String title){
44         this.title = title;
45     }
46
47     public Media(String title, String category, float cost){
48         this.title = title;
49         this.category = category;
50         this.cost = cost;
51     }
52
53 }
```

- DigitalVideoDisc Class:

```
© Cart.java  © Media.java  © Aims.java  © Book.java  © DigitalVideoDisc.java ×
1  package hust.soict.dsai.aims.media;
2
3  public class DigitalVideoDisc extends Media {
4      //Attributes
5      private String director;
6      private int length;
7
8      //Getter and setter
9      public String getDirector() {
10         return director;
11     }
12
13     public void setDirector(String director) {
14         this.director = director;
15     }
16
17     public int getLength(){
18         return length;
19     }
20
21     public void setLength(int length) {
22         this.length = length;
23     }
24
25     public int getnbDigitalVideoDisc(){
26         //Class variables to manage IDs and count of DVDs
27         return 0;
28     }
29     
```

```
30 //Constructor
31 //Default constructor
32
33 //Create a DVD object by title
34 public DigitalVideoDisc(String title){
35     super();
36     setTitle(title);
37 }
38
39
40
41 //Create a DVD object by category, title and cost
42 public DigitalVideoDisc(String title, String category, float cost){
43     super();
44     setTitle(title);
45     setCategory(category);
46     setCost(cost);
47 }
48
49 public void setTitle(String title) {
50 }
51
52 //Create a DVD object by director, category, title and cost
53 public DigitalVideoDisc(String title, String category, String director, float cost){
54     super();
55     setTitle(title);
56     setCategory(category);
57     setDirector(director);
58     setCost(cost);
59 }
```

```
60
61 //Create a DVD object by all attributes: title, category,director, cost, length
62 public DigitalVideoDisc(String title, String category, String director, int length, float cost){
63     super();
64     setTitle(title);
65     setCategory(category);
66     setDirector(director);
67     setLength(length);
68     setCost(cost);
69 }
70
71 }
```


5. Creating the CompactDisc Class

5.1. Create the Disc class extending the Media Class

- Disc Class:

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

- DigitalVideoDisc Class:

```
1 package hust.soict.dsai.aims.media;
2
3 public class DigitalVideoDisc extends Disc{
4
5     public int getNbDigitalVideoDisc(){
6         //Class variables to manage IDs and count DVDs
7         return 0;
8     }
9
10    //Constructor
11    public DigitalVideoDisc(String director, int length){
12        super(director, length);
13    }
14
15    public DigitalVideoDisc(String title, String category, String director, int length, float cost){
16        super(title, category, director, length, cost);
17    }
18    public DigitalVideoDisc(String title, String category, float cost){
19        super(title, category, cost);
20    }
21
22    public DigitalVideoDisc(String title){
23        super(title);
24    }
25
26    public DigitalVideoDisc(){
27        super();
28    }
29
30 }
```

- CompactDisc Class:

```
Book.java  DigitalVideoDisc.java  CompactDisc.java  Disc.java
1 package hust.soict.dsai.aims.media;
2
3 public class CompactDisc extends Disc {
4 }
5
```

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

- Track class:



```
1 package hust.soict.dsai.aims.media;
2
3 public class Track {
4     private final String title;
5     private final int length;
6
7     //Getter methods
8     public String getTitle() {
9         return title;
10    }
11
12    public int getLength() {
13        return length;
14    }
15
16    //Constructor
17    public Track(String title, int length) {
18        this.title = title;
19        this.length = length;
20    }
21 }
```

5.3. Open the CompactDisc

- CompactDisc Class:



```
1 package hust.soict.dsai.aims.media;
2
3 import java.util.ArrayList;
4
5 public class CompactDisc extends Disc {
6     private final String artist;
7     private final ArrayList<Track> tracks;
8
9     //Getter method for artist
10    public String getArtist() {
11        return artist;
12    }
13
14    //Constructor
15    public CompactDisc(String artist){
16        super();
17        this.artist = artist;
18        this.tracks = new ArrayList<>();
19    }
20
21    //Method for tracks
22    public void addTrack(Track track){
23        if (!tracks.contains(track)){
24            tracks.add(track);
25            System.out.println("Track added successfully!");
26        }
27
28        else{
29            System.out.println("Track exists in CD!");
30        }
31    }
```

```
33     public void removeTrack(Track track){
34         if(tracks.contains(track)){
35             tracks.remove(track);
36             System.out.println("Track removed successfully");
37         }
38     }
39     else{
40         System.out.println("Track does not exist in CD!");
41     }
42 }
43 //Method to get length of CD
44 public int getLength(){
45     int totalLength = 0;
46     for(Track track : tracks){
47         totalLength += track.getLength();
48     }
49     return totalLength;
50 }
51 }
```

6. Create the Playable interface

- Playable Interface

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

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

- DigitalVideoDisc class:

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

public class DigitalVideoDisc extends Disc implements Playable{
    public DigitalVideoDisc (String director, int length){
        super(director, length);
    }

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

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

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

    public DigitalVideoDisc (){
        super();
    }

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

- Track class:

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

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

    //Getter methods
    public String getTitle() {
        return title;
    }
}
```

```

    public int getLength() {
        return length;
    }

    //Constructor
    public Track(String title, int length) {
        this.title = title;
        this.length = length;
    }
    @Override
    public void play(){
        System.out.println("Playing track: " + this.getTitle());
        System.out.println("Track length: " + this.getLength());
    }
}

```

- ➔ Với các thay đổi trên, đã thêm khả năng phát của CompactDisc DigitalVideoDisc và Track thông qua việc triển khai interface Playable, đồng thời triển khai phương thức play() cho mỗi lớp tương ứng.

7. Update the Cart class to work with Media

- Cart class

```

package hust.soict.dsai.aims.cart;
import hust.soict.dsai.aims.media.DigitalVideoDisc;
import hust.soict.dsai.aims.media.Media;

import java.io.PrintStream;
import java.util.ArrayList;

public class Cart {

    // Max Number of order
    public static final int MAX_NUMBERS_ORDERED = 20;

    //ArrayList store orders
    private ArrayList<Media> itemsOrdered = new ArrayList<>();

    //add a media item to the list
    public void addMedia(Media media) {
        //check the current quantity to assure that the cart isn't full
        if (itemsOrdered.size() < MAX_NUMBERS_ORDERED) {
            itemsOrdered.add(media);
            System.out.println("The media has been added!");
        } else {
            System.out.println("The cart is full!");
        }
    }

    //remove the item passed by argument from the list
    public void removeMedia(Media media) {
        if (itemsOrdered.remove(media)) {
            System.out.println("The media has been removed!");
        } else {

```

```

        System.out.println("The media not found!");
    }
}

//method which loops through the values of the array and sums the cost
of the individual Media items
public float totalCost(){
    float total = 0;
    for (Media media : itemsOrdered){
        total += media.getCost();
    }
    return total;
}

public void listCart(){
    for (Media media : itemsOrdered){
        System.out.println("id: " + media.getId() + " " +
media.getTitle());
    }
}

//Search by ID
public boolean search(int id){
    for (Media media : itemsOrdered){
        if (media.getId() == id){
            return true;
        }
    }
    return false;
}

//Search by Title
public boolean search(String title){
    for (Media media : itemsOrdered){
        if (media.getTitle().equals(title){
            return true;
        }
    }
    return false;
}

public void printCart() {
    System.out.println("*****CART*****");
    System.out.println(" Ordered Items:");
    for (Media media : itemsOrdered) {
        System.out.printf("id: %d. %s\n" + media.getId(),
media.getTitle());
    }
    float totalCost = totalCost();
    System.out.printf("Total cost: %.2f $\n", totalCost);
    System.out.println("*****");
}
}

```


- Sử dụng `ArrayList<Media>` thay vì mảng để lưu trữ các đối tượng `Media`
- Sử dụng phương thức `addMedia()` thay vì `addDigitalVideoDisc()`
- Sử dụng phương thức `removeMedia()` thay vì `removeDigitalVideoDisc()`
- Sử dụng `size()` để kiểm tra số lượng phần tử trong giỏ hàng
- Sử dụng vòng lặp `for-each` để duyệt qua danh sách các đối tượng `Media`
- Sửa lại phương thức `printCart()` để in thông tin đối tượng `Media`

8. Update the Store class to work with Media

- Store class:

```

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

import java.util.ArrayList;

public class Store {

    //Max Number of items in store
    public static final int MAX_NUMBERS_STORED = 100;

    //ArrayList store items
    private ArrayList<Media> itemsInStore = new ArrayList<>();

    //add a media item to the store
    public void addMedia(Media media) {

        //check the current quantity to assure that the store is not
full
        if (itemsInStore.size() < MAX_NUMBERS_STORED) {
            itemsInStore.add(media);
            System.out.println("The media has been added to the
store!");
        }
        else {
            System.out.println("The store is full!");
        }
    }

    //remove items
    public void removeMedia(Media media) {
        if (itemsInStore.remove(media)) {
            System.out.println("The media has been removed!");
        }
        else{
            System.out.println("The media not found!");
        }
    }

    //list all items
    public void listItems() {
        for (Media media : itemsInStore) {
            System.out.println("id: " + media.getId() + " " +
media.getTitle());
        }
    }
}

```

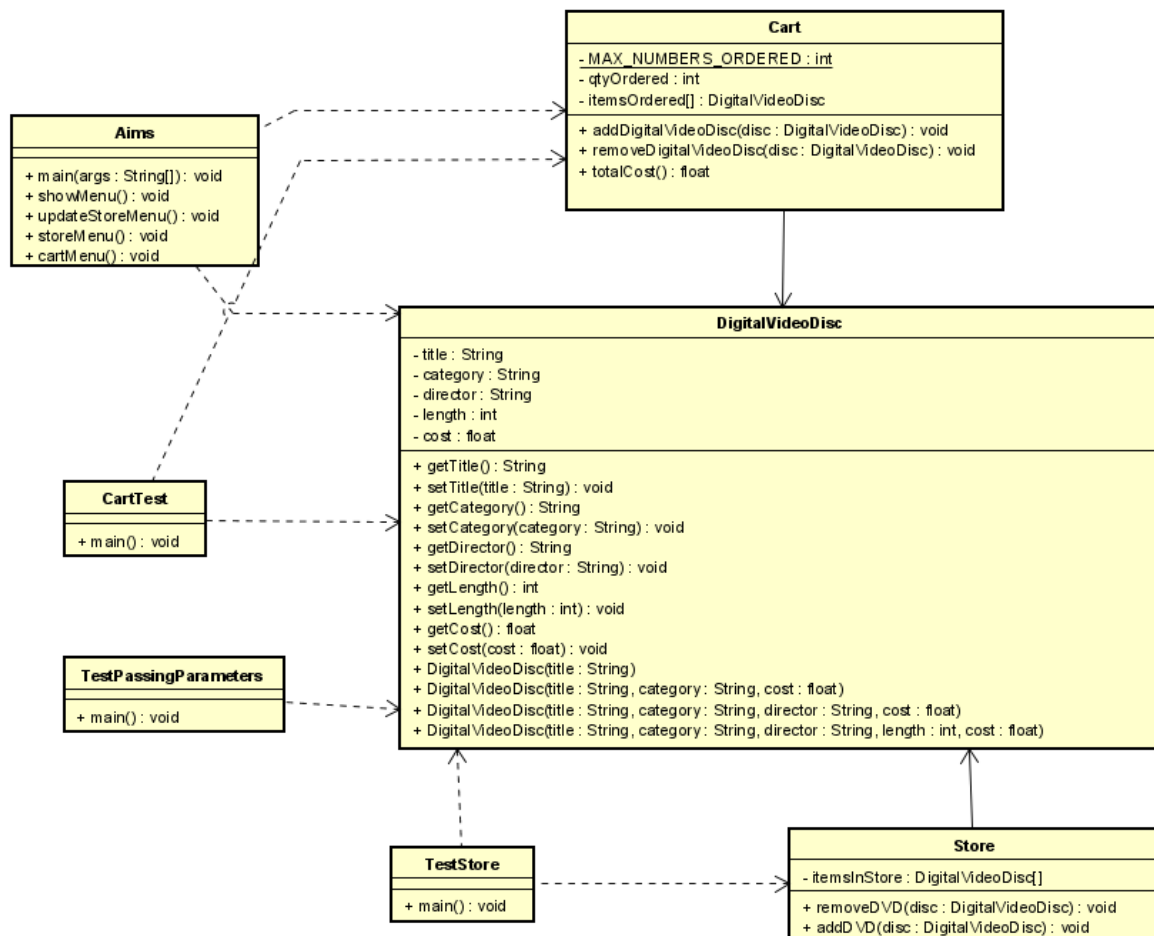
```
    }

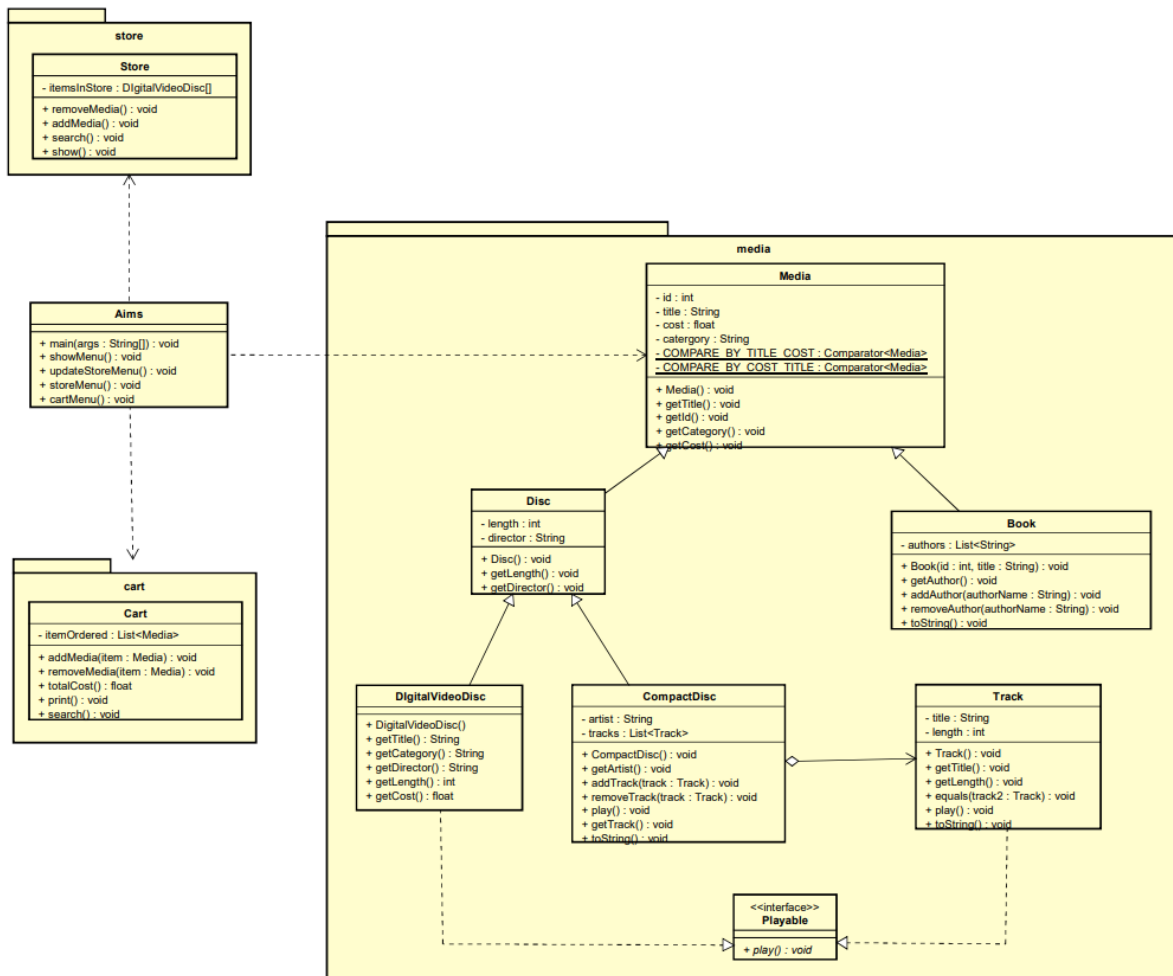
    //search by ID
    public boolean search(int id){
        for (Media media : itemsInStore){
            if (media.getId() == id){
                return true;
            }
        }
        return false;
    }

    //search by title
    public boolean search(String title){
        for (Media media : itemsInStore){
            if (media.getTitle().equals(title)){
                return true;
            }
        }
        return false;
    }
}
```

- itemsInStore từ mảng DigitalVideoDisc[] sang ArrayList<Media> trong lớp Store.
- Sau đó, cần thay thế các phương thức addDVD() và removeDVD() bằng addMedia() và removeMedia().

9. Constructor of whole classes and parent classes





10. Unique item in a list

- Track class:

```

@Override
public boolean equals(Object o) {
    Track tmp = (Track) o;
    if (this.getTitle() == tmp.getTitle() && this.getLength() ==
tmp.getLength()) {
        return true;
    } else return false;
}

```

- Media class:

```

@Override
public boolean equals(Object o){
    Media tmp = (Media) o;
    if (this.getTitle() == tmp.getTitle()){
        return true;
    }
    else return false;
}

```

11. Polymorphism with toString() method

- MediaTest class:

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

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

public class MediaTest {
    public static void main(String args[]) {
        List<Media> media = new ArrayList<Media>();

        DigitalVideoDisc dvd = new DigitalVideoDisc("Conan", "Comic",
"aidji", 87, 19.95f);

        Book book = new Book();

        CompactDisc cd = new CompactDisc();

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

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

- Kết quả:



```
Run  MediaTest x

"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
hust.soict.dsai.aims.media.CompactDisc@1b6d3586
hust.soict.dsai.aims.media.DigitalVideoDisc@4554617c
hust.soict.dsai.aims.media.Book@74a14482

Process finished with exit code 0
```

12.Sort media in the cart

- MediaComparatorByCostTitle Class:

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

import java.util.Comparator;

public class MediaComparatorByCostTitle implements Comparator<Media>{

    @Override
    public int compare(Media o1, Media o2) {
        // TODO Auto-generated method stub
        return o1.getTitle().compareTo(o2.getTitle());
    }

}
```

- MediaComparatorByTitleCost class:

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

import java.util.Comparator;

public class MediaComparatorByTitleCost implements Comparator<Media>{
    @Override
    public int compare(Media o1, Media o2) {
        // TODO Auto-generated method stub
        return o1.getTitle().compareTo(o2.getTitle());
    }
}
```

- Media class:

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

13. Create a complete console application in the Aims class

- Menu chính:

```
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
The media has been added
The media has been added
The media has been added
The media has been added
The media has been added
AIMS:
-----
1. View store
2. Update store
3. See current cart
0. Exit
-----
Please choose a number: 0-1-2-3
```

- View Store

```
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
The media has been added
The media has been added
The media has been added
The media has been added
The media has been added
AIMS:
-----
1. View store
2. Update store
3. See current cart
0. Exit
-----
Please choose a number: 0-1-2-3
1
*****STORE*****
1  | abc (Book) --- 100.0$
2  | xyz (Book) --- 85.0$
3  | nch (Book) --- 197.0$
4  | IU (CD) --- 90.0$
5  | DVD 1 (DVD) --- 35.88$
*****
```

- Media's Details:

```
Options:
-----
1. See a media's details
2. Add a media to cart
3. Play a media
4. See current cart
0. Back
-----
Please choose a number: 0-1-2-3-4
1
>>>>>Enter the title of the media:
>>>>>abc
```

```
|| Title: abc
|| Authors: h, ani
|| Cost: $100.00
```

- **Play Media:**

```
Options:
-----
1. See a media's details
2. Add a media to cart
3. Play a media
4. See current cart
0. Back
-----
Please choose a number: 0-1-2-3-4
3
>>>>>Enter the title of the media you want to play:
>>>>>IU
```

[illegible]

- Thêm vào Cart:

```
Options:
-----
1. See a media's details
2. Add a media to cart
3. Play a media
4. See current cart
0. Back
-----
Please choose a number: 0-1-2-3-4
2
>>>>Enter the title of the media you want to add to the cart:
>>>>xyz

>>>>>>>>>>=====<<<<<<<<<<<
The media has been added
>>>>>>>>>>=====<<<<<<<<<<<
```

- Xem Cart:

```
Options:
-----
1. See a media's details
2. Add a media to cart
3. Play a media
4. See current cart
0. Back
-----
Please choose a number: 0-1-2-3-4
4
*****CART*****
Ordered Items:
#1: xyz. 85.00 $
-----
==> Total cost: 85.00 $
*****
```

- **Filter theo title:**

```
Options:
-----
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
-----
Please choose a number: 0-1-2-3-4-5
1
>>>>Enter the title of the media you want to filter:
>>>>xyz
*****CART*****
Ordered Items by Title:
#1: xyz. 85.00 $
#2: xyz2. 197.00 $
*****
```

- **Sort theo Cost/Tite:**

```
Options:
-----
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
-----
Please choose a number: 0-1-2-3-4-5
2
>>>>>>Sort by:
1. Cost
2. Title
1
Cart sorted by cost.
*****CART*****
Ordered Items:
#1: xyz. 85.00 $
#2: xyz2. 197.00 $
-----
==> Total cost: 282.00 $
*****
```

- Xóa Media khỏi Order:

```
Options:
-----
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
-----
Please choose a number: 0-1-2-3-4-5
3
>>>>>Enter the title of the media you want to remove:
>>>>>xyz

>>>>>>>>>>=====<<<<<<<<<<<<
The disc has been removed
The media does not exist in the cart.
>>>>>>>>>>=====<<<<<<<<<<<<
```

- Place order

```
Options:
-----
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
-----
Please choose a number: 0-1-2-3-4-5
5
**** Order is created ****
```

- Update Store:

```
-----  
1. View store  
2. Update store  
3. See current cart  
0. Exit  
-----  
Please choose a number: 0-1-2-3  
2  
#####Update Store#####  
-----  
1. Add a Media  
2. Remove a Media  
0. Back  
-----  
Please choose a number: 0-1-2  
1
```

```
What type of media do you want to add?  
-----  
1. Book  
2. DVD  
3. CD  
0. Back  
-----  
Please choose a number: 0-1-2-3  
1  
-----Book-----  
Title:  
hgx  
Cost:  
92  
Author (enter 'done' to finish):  
kadj  
done  
The media has been added  
Added Book to Store successfully
```

- Remove Media khỏi Store:

```

-----
1. Add a Media
2. Remove a Media
0. Back
-----
Please choose a number: 0-1-2
2
>>>>>Enter the title of the media you want to remove:
>>>>>hgx

>>>>>>>>>>=====<<<<<<<<<<<<
The disc has been removed
>>>>>>>>>>=====<<<<<<<<<<<<

#####Update Store#####

```

- Code:

```

package hust.soict.dsai.aims.aims;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Scanner;

import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.media.*;
import hust.soict.dsai.aims.store.Store;

public class Aims {

    private static Scanner sc = new Scanner(System.in);
    private static Cart order = new Cart();
    private static Store store = new Store();
    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");
    }

    public static void storeMenu() {

```

```

        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");
    }

    public static void mediaDetailsMenu() {
        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");
    }

    public static void cartMenu() {
        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");
    }

    //=====View
store=====//
    public static void test() {}
    public static void showDetailsOfEachMediaType(Media media) {

        int maxLen = 20;
        if (media instanceof DigitalVideoDisc) {

            System.out.println("┌───────────────────────────────────┐");
            System.out.printf("│ Title: %-" + maxLen + "s \n",
media.getTitle());
            System.out.printf("│ Director: %-" + maxLen + "s \n",
((DigitalVideoDisc) media).getDirector());
            System.out.printf("│ Length: %-3d minutes \n",
((DigitalVideoDisc) media).getLength());
            System.out.printf("│ Cost: $%-5.2f \n", media.getCost());
            System.out.println("└───────────────────────────────────┘");

        } else if (media instanceof CompactDisc) {

            System.out.println("┌───────────────────────────────────┐");
            System.out.printf("│ Title: %-" + maxLen + "s \n",
media.getTitle());
            System.out.printf("│ Artist: %-" + maxLen + "s \n",
((CompactDisc) media).getArtist());

```

[illegible]


```

        case 1:
            addToCart(m);
            break;

        // 2. Play
        case 2:
            playMedia(m);
            break;

        // 0. Back
        case 0:
            return true;

        default:

System.out.println("=====");
        System.out.println("===>>>This option is not
available<<<===");

System.out.println("=====");
        return false;
    }
    return false;
}

public static void showMediaDetails() {

    sc.nextLine();
    System.out.println(">>>>>Enter the title of the media: ");
    System.out.print(">>>>>");
    Media tmp = store.searchMedia(sc.nextLine());
    if(tmp!=null){
        showDetailsOfEachMediaType(tmp);

        // Menu Media Details
        boolean backToMenuStore = false;

        while (!backToMenuStore) {
            mediaDetailsMenu();
            backToMenuStore =
solveOptionOfMediaDetailsSelected(tmp);
        }
    }
    else {

System.out.println("=====");
        System.out.println("===>>>The Media does not exist in the
store<<<===");

System.out.println("=====");
    }

}

public static void addAMediaToCart() {
    sc.nextLine();
    System.out.println(">>>>>Enter the title of the media you want
to add to the cart: ");
    System.out.print(">>>>>");
    Media tmp = store.searchMedia(sc.nextLine());
    if(tmp!=null){

```

[illegible]

```

available<<<==");

System.out.println("=====");
        return false;
    }
    return false;
}

public static void viewStore() {

    boolean backToMenuMain = false;

    while (!backToMenuMain) {
        store.printStore();
        storeMenu();
        backToMenuMain = solveOptionOfStoreMenuSelected();
    }
}

//=====
=====//

//=====Update
store=====//

public static void updateStoreMenu() {
    System.out.println("#####Update Store#####");
    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");
}

public static void addAMediaToStoreMenu() {
    System.out.println("What type of media do you want to add?");
    System.out.println("-----");
    System.out.println("1. Book");
    System.out.println("2. DVD");
    System.out.println("3. CD");
    System.out.println("0. Back");
    System.out.println("-----");
    System.out.println("Please choose a number: 0-1-2-3");
}

public static void addABookToStore() {
    System.out.println("-----Book-----");
    sc.nextLine();
    System.out.println("Title: ");
    String title = sc.nextLine();
    System.out.println("Cost: ");
    float cost = sc.nextFloat();
    List<String> authors = new ArrayList<>();
    sc.nextLine();
    System.out.println("Author (enter 'done' to finish): ");
    while (true) {

        String author = sc.nextLine();
    }
}

```

```

        if (author.equalsIgnoreCase("done")) {
            break;
        }

        authors.add(author);
    }
    store.addMedia(new Book(title, "Book", cost, authors));
    System.out.println("Added Book to Store successfully");
    System.out.println("-----");
}
public static void addADVDToStore() {
    System.out.println("-----DVD-----");
    sc.nextLine(); // Consume the newline character

    System.out.print("Title: ");
    String title = sc.nextLine();

    System.out.print("Cost: ");
    float cost = sc.nextFloat();
    sc.nextLine(); // Consume the newline character

    System.out.print("Director: ");
    String director = sc.nextLine();

    System.out.print("Length: ");
    int length = sc.nextInt();
    sc.nextLine(); // Consume the newline character

    store.addMedia(new DigitalVideoDisc(title, "DVD", director,
length, cost));
    System.out.println("Added DVD to Store successfully");
    System.out.println("-----");
}
public static void addACDToStore() {
    sc.nextLine(); // Consume the newline character
    System.out.println("-----CD-----");
    System.out.print("Title: ");
    String title = sc.nextLine();

    System.out.print("Cost: ");
    float cost = sc.nextFloat();
    sc.nextLine(); // Consume the newline character

    System.out.print("Artist: ");
    String artist = sc.nextLine();

    System.out.println("Enter Track Information for CD (Enter '0' to
finish):");
    ArrayList<Track> tracks = new ArrayList<>();
    int trackNumber = 1;
    while (true) {
        System.out.print("Track " + trackNumber + " Title (Enter '0'
to finish): ");
        String trackTitle = sc.nextLine();
        if (trackTitle.equals("0")) {
            break;
        }

        System.out.print("Track " + trackNumber + " Length: ");
        int trackLength = sc.nextInt();

```

```

        sc.nextLine(); // Consume the newline character

        Track track = new Track(trackTitle, trackLength);
        tracks.add(track);
        trackNumber++;
    }

    store.addMedia(new CompactDisc(title, "CD", cost, artist,
tracks));
    System.out.println("Added CD to Store successfully");
    System.out.println("-----");
}

public static boolean solveOptionOfAddAMediaToStoreSelected(){
    switch (readOption()){

        // 1. Add a Book
        case 1:
            addABookToStore();
            break;

        // 2. Add a DVD
        case 2:
            addADVDTToStore();
            break;

        // 3. Add a CD
        case 3:
            addACDToStore();
            break;

        // 0. Back
        case 0:
            return true;

        default:

System.out.println("=====");
            System.out.println("===>>>This option is not
available<<<===");

System.out.println("=====");
            return false;
        }
        return false;
    }
    public static void addAMediaToStore(){

        boolean backToUpdateStoreMenu = false;

        while (!backToUpdateStoreMenu) {
            addAMediaToStoreMenu();
            backToUpdateStoreMenu =
solveOptionOfAddAMediaToStoreSelected();
        }
    }
    public static void removeAMediaFromStore(){
        sc.nextLine();
        System.out.println(">>>>>Enter the title of the media you want
to remove: ");
        System.out.print(">>>>>");
    }

```

```
Media tmp = store.searchMedia(sc.nextLine());
if(tmp!=null){
    System.out.println("\n>>>>>>>>>>=====<<<<<<<<<<<");
    store.removeMedia(tmp);
    System.out.println(">>>>>>>>>>=====<<<<<<<<<<<\n");
}
else {

System.out.println("=====");
    System.out.println("==>>>The Media does not exist in the
store<<<==");

System.out.println("=====");
}
}

public static boolean solveOptionOfUpdateStoreMenuSelected(){
    switch (readOption()){

        // 1. Add a Media
        case 1:
            addAMediaToStore();
            break;

        // 2. Remove a Media
        case 2:
            removeAMediaFromStore();
            break;

        // 0. Back
        case 0:
            return true;

        default:

System.out.println("=====");
        System.out.println("==>>>This option is not
available<<<==");

System.out.println("=====");
        return false;
    }
    return false;
}

public static void updateStore(){

    boolean backToMenuMain = false;

    while (!backToMenuMain) {
        updateStoreMenu();
        backToMenuMain = solveOptionOfUpdateStoreMenuSelected();
    }

}

//=====
=====//

//=====See current
cart=====//
```

```

        public static void filterMediaInCart() {

            sc.nextLine();
            System.out.println(">>>>>Enter the title of the media you want
to filter: ");
            System.out.print(">>>>>");
            String title = sc.nextLine();

            ArrayList<Media> itemsOrdered = order.getOrderedList();

System.out.println("*****CART*****");
;
            System.out.println("Ordered Items by Title:");
            int i = 0;

            String lowerCaseKeyword = title.toLowerCase();
            for (Media media : itemsOrdered) {

                String lowerCaseTitle = media.getTitle().toLowerCase();
                if (lowerCaseTitle.contains(lowerCaseKeyword)) {
                    System.out.printf("#%d: %s. %.2f $\\n", ++i,
media.getTitle(), media.getCost());
                }

            }

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

        public static void sortMediasInCart() {
            System.out.println(">>>>>>Sort by: \\n1. Cost\\n2. Title");
            switch (readOption()) {
                case 1:
                    order.sortByCost();
                    break;
                case 2:
                    order.sortByTitle();
                    break;
            }
        }

        public static void removeAMediaFromCart() {

            sc.nextLine();
            System.out.println(">>>>>Enter the title of the media you want
to remove: ");
            System.out.print(">>>>>");
            Media tmp = order.searchMedia(sc.nextLine());
            if (tmp != null) {
                System.out.println("\\n>>>>>>>>>>=====<<<<<<<<<<<<");
                order.removeMedia(tmp);
                System.out.println(">>>>>>>>>>=====<<<<<<<<<<<<\\n");
            }
            else {

System.out.println("=====");
                System.out.println("===>>>The Media does not exist in the
cart<<<===");
            }
        }
    }
}

```

```

System.out.println("=====");
    }
}

public static void placeOrder(){
    if(order.getOrderedList().isEmpty()){
        System.out.println("!!!! Your order is EMPTY !!!!");
    }
    else{
        System.out.println("**** Order is created ****");
    }
}

public static boolean solveOptionOfCartMenuSelected(){
    switch (readOption()){

        // 1. Filter medias in cart
        case 1:
            filterMediaInCart();
            break;

        // 2. Sort medias in cart
        case 2:
            sortMediasInCart();
            break;

        // 3. Remove media from cart
        case 3:
            removeAMediaFromCart();
            break;

        // 4. Play a media
        case 4:
            playAMedia();
            break;

        // 5. Place order
        case 5:
            placeOrder();
            break;

        // 0. Back
        case 0:
            return true;

        default:

System.out.println("=====");
        System.out.println("==>>>This option is not
available<<<==");

System.out.println("=====");
        return false;
    }
    return false;
}

public static void seeCurrentCart(){

    order.printCart();
    boolean back= false;

```



```

        while (!back) {
            cartMenu();
            back = solveOptionOfCartMenuSelected();
        }

    }

//=====
//=====//

//=====Menu
Main=====//
    public static void solveOptionSelected() {
        switch (readOption()) {

            // 1. View store
            case 1:
                viewStore();
                break;

            // 2. Update store
            case 2:
                updateStore();
                break;

            // 3. See current cart
            case 3:
                seeCurrentCart();
                break;

            // 0. Exit
            case 0:
                System.out.println("Are you sure you want to exit the
system ? (y/n)");
                if (sc.next().charAt(0) == 'y') {
                    System.exit(0);
                    sc.close();
                }
                else {
                    break;
                }
                break;

            default:

System.out.println("=====");
                System.out.println("==>>>This option is not
available<<<==");

System.out.println("=====");
                break;
        }
    }

    public static int readOption() {
        return sc.nextInt();
    }

    public static void initData() {

```

```
// Book
Book initBook = new Book("abc", "Book", 100.00f,
    Arrays.asList("h", "ani"));

// Book
Book initBook2 = new Book("xyz", "Book", 85.00f,
    Arrays.asList("a", "ki"));

// Book
Book initBook3 = new Book("xyz2", "Book", 197.00f,
    Arrays.asList("u", "ai"));

// DVD
DigitalVideoDisc initDVD = new DigitalVideoDisc("DVD 1", "DVD",
    "abc ", 150, 35.88f);

// CD
ArrayList<Track> tracks = new ArrayList<>();
tracks.add(new Track("GD", 3));
tracks.add(new Track("MCK", 4));
CompactDisc initCD = new CompactDisc("IU", "CD", 90.00f,
    "YG", tracks);

store.addMedia(initBook);
store.addMedia(initBook2);
store.addMedia(initBook3);
store.addMedia(initCD);
store.addMedia(initDVD);
}

//=====
=====//

public static void main(String[] args) {

    initData();
    while(true){
        showMenu();
        solveOptionSelected();
    }
}
}
```

