

Báo cáo thực hành 4

Lập trình hướng đối tượng

Contents

1.	Creating the abstract Media class.....	2
2.	Create the Disc class extending the Media class.....	4
3.	Creating the Playable interface	5
4.	Creating the Book class.....	5
5.	Creating the Track class	6
6.	Creating the CompactDisc class	7
7.	Update the DigitalVideoDisc class	8
8.	Update the Cart class	9
9.	Update the Store class	11
10.	Update the Aims class	12
11.	Demo Aims	21
12.	Update the class diagram	22

1. Creating the abstract Media class

```
1 package lap_04.AimesProject.Media;
2
3 import java.util.Comparator;
4
5 @Explain | Test | Document | Fix | Ask
6 4 inheritors new *
7 public abstract class Media {
8     9 usages
9     protected int id;
10    7
11    protected String title;
12    12 usages
13    protected String category;
14    8
15    protected float cost;
16    8 usages
17    int nbMedia=0;
18    1 usage
19    public static final Comparator<Media> COMPARE_BY_TITLE_COST =
20    12
21        new MediaComparatorByTitleCost();
22    1 usage
23    public static final Comparator<Media> COMPARE_BY_COST_TITLE =
24    14
25        new MediaComparatorByCostTitle();
26
27    new *
28    public int getId() {
29        return id;
30    }
31
32    new *
33    public void setId(int id) {
34        this.id = id;
35    }
36
37    new *
38    public String getTitle() {
39        return title;
40    }
41
42    new *
43    public void setTitle(String title) {
44        this.title = title;
45    }
46
47    new *
48    public String getCategory() {
49        return category;
50    }
51 }
```

```
        return cost;
```

3 ^

```
    }
```

```
new *
```

```
public void setCost(float cost) {  
    this.cost = cost;  
}
```

[Explain](#) | [Test](#) | [Document](#) | [Fix](#) | [Ask](#)

```
new *
```

```
public Media() {  
    nbMedia++;  
    this.id = nbMedia;  
}
```

[Explain](#) | [Test](#) | [Document](#) | [Fix](#) | [Ask](#)

```
new *
```

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

[Explain](#) | [Test](#) | [Document](#) | [Fix](#) | [Ask](#)

```
new *
```

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

[Explain](#) | [Test](#) | [Document](#) | [Fix](#) | [Ask](#)

```
new *
```

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

[Explain](#) | [Test](#) | [Document](#) | [Fix](#) | [Ask](#)

```
new *
```

```
@Override  
public boolean equals(Object obj) {  
    if(obj instanceof Media) {  
        Media mda = (Media) obj;  
        return this.title.equals(mda.title);  
    }  
    return false;  
}
```

```
}
```

2. Create the Disc class extending the Media class

```
1 package lap_04.AimesProject.Media;
2
3 🔍 Explain | Test | Document | Fix | Ask
4 4 usages 2 inheritors new *
5 @ public class Disc extends Media {
6
7     protected int length;
8     6 usages
9     protected String director;
10    1 override new *
11    public int getLength() { return length; }
12    no usages new *
13    public String getDirector() { return director; }
14    2 usages new *
15    public Disc() {
16        super();
17    }
18    🔍 Explain | Test | Document | Fix | Ask
19    1 usage new *
20    public Disc(String title) {
21        super(title);
22        this.title = title;
23    }
24    🔍 Explain | Test | Document | Fix | Ask
25    1 usage new *
26    public Disc(String title, String category, float cost) {
27        super(title,category,cost);
28        this.title = title;
29        this.category = category;
30        this.cost = cost;
31    }
32    🔍 Explain | Test | Document | Fix | Ask
33    no usages new *
34    public Disc(String title, String category) {
35        super(title,category);
36        this.title = title;
37        this.category = category;
38    }
39    🔍 Explain | Test | Document | Fix | Ask
40    1 usage new *
41    public Disc(String title, String category, String director, int length, float cost) {
42        super(title,category,cost);
43        this.title = title;
44        this.category = category;
45        this.director = director;
46        this.length = length;
47        this.cost = cost;
48    }
49    🔍 Explain | Test | Document | Fix | Ask
50    2 usages new *
51    public Disc(String title, String category, String director, float cost) {
52        super(title,category,cost);
53        this.title = title;
54        this.category = category;
55        this.director = director;
56        this.cost = cost;
57    }
58    new *
59    public void setLength(int length) { this.length = length; }
60    2 usages new *
61    public void setDirector(String director) { this.director = director; }
62 }
```

3. Creating the Playable interface

```
package lap_04.AimesProject.Media;

10 usages 3 implementations new *
public interface Playable {
    4 usages 3 implementations new *
    public void play();
}
```

4. Creating the Book class

```
1 package lap_04.AimesProject.Media;
2
3 > import ...
4
5 Explain | Test | Document | Fix | Ask
6 usages new *
6 public class Book extends Media {
    7 usages
    7 private List<String> authors = new ArrayList<>();
    no usages new *
    8 > public List<String> getAuthors() { return authors; }
    no usages new *
    11 > public void setAuthors(List<String> authors) { this.authors = authors; }
    Explain | Test | Document | Fix | Ask
    1 usage new *
    14 public void addAuthor(String authorName) {
    15     if(authors.contains(authorName)) {
    16         System.out.println("Author has already in list");
    17         return;
    18     }
    19     authors.add(authorName);
    20 }
    21
    Explain | Test | Document | Fix | Ask
    no usages new *
    22 public void removeAuthor(String authorName) {
    23     if(authors.contains(authorName)) {
    24         authors.remove(authorName);
    25         return;
    26     }
    27     System.out.println("Cant find author!");
    28 }
    Explain | Test | Document | Fix | Ask
    no usages new *
    29 public Book(String title, String category, float cost) {
    30     super(title,category,cost);
    31     this.title = title;
    32     this.category = category;
    33     this.cost = cost;
    34 }
    2 usages new *
    35 public Book() {
    36
    37 }
    Explain | Test | Document | Fix | Ask
    new *
    38 @Override
    39 @ public String toString() {
    40     return "Book: ID" + id + " Title:" + title + ", " + " Authors:" + authors
    41         + ", Category:" + category + ",Cost:" + cost;
    42 }
    43
```

5. Creating the Track class

```
1 package lap_04.AimesProject.Media;
2
3 public class Track implements Playable{
4     private String title;
5     private int length;
6
7     public String getTitle() { return title; }
8
9     public int getLength() { return length; }
10
11     public Track(String title, int length) {
12         this.title = title;
13         this.length = length;
14     }
15
16     public void play() {
17         System.out.println("Playing Track: " + this.getTitle());
18         System.out.println("Track length: " + this.getLength());
19     }
20
21     @Override
22     public boolean equals(Object obj) {
23         if(obj instanceof Track) {
24             Track trk = (Track) obj;
25             return this.title.equals(trk.title) && this.length==trk.length;
26         }
27         return false;
28     }
29
30     @Override
31     public String toString() { return "Track: title=" + title + ", length=" + length + "]; }
32 }
```

6. Creating the CompactDisc class

```
1 package lap_04.AimesProject.Media;
2
3 > import ...
4
5 Explain | Test | Document | Fix | Ask
6 usages new *
7
8 public class CompactDisc extends Disc implements Playable{
9     3 usages
10     private String artist;
11     10 usages
12     private List<Track>tracks = new ArrayList<>();
13
14     no usages new *
15     > public String getArtist() { return artist; }
16
17 Explain | Test | Document | Fix | Ask
18 1 usage new *
19 public CompactDisc(String title, String category, String director, float cost,
20                     String artist, List<Track> tracks) {
21     super(title,category,director,cost);
22     this.artist = artist;
23     this.tracks = tracks;
24 }
25
26 1 usage new *
27 public CompactDisc() {
28
29 }
30 Explain | Test | Document | Fix | Ask
31 // public boolean equals(Object obj) {
32 //     if(obj instanceof Track) {
33 //         Track trk = (Track) obj;
34 //         return this.equals(trk.getTitle()) && this == trk.getLength();
35 //     }
36 // }
37
38 1 usage new *
39 public void addTrack(Track track) {
40     if(tracks.contains(track)) {
41         System.out.println("The track is already in the list!");
42         return;
43     }
44     tracks.add(track);
45 }
```

```

35     }
    Explain | Test | Document | Fix | Ask
    no usages new *
36     public void removeTrack(Track track) {
37         if(tracks.contains(track)) {
38             tracks.remove(track);
39             return;
40         }
41         System.out.println("Cant find the track!");
42     }
43
    Explain | Test | Document | Fix | Ask
    new *
44     public int getLength() {
45         int sumLen = 0;
46         for(int i=0; i<tracks.size();i++) {
47             sumLen += tracks.get(i).getLength();
48         }
49         return sumLen;
50     }
    Explain | Test | Document | Fix | Ask
    4 usages new *
51     public void play() {
52         System.out.println("Playing CD: " + this.getTitle());
53         System.out.println("CD Length: " + this.getLength());
54         for(int i=0; i<tracks.size();i++) {
55             tracks.get(i).play();
56         }
57     }
58
    Explain | Test | Document | Fix | Ask
    new *
59     @Override
60     public String toString() {
61         return "CompactDisc: id"+ id +"artist=" + artist + ", tracks=" + tracks + ", length=" + length + ", director=" + director
62             + ", title=" + title + ", category=" + category + ", cost=" + cost + "]";
63     }
64 }
65

```

7. Update the DigitalVideoDisc class

```

1 package lap_04.AimesProject.Media;
2
    Explain | Test | Document | Fix | Ask
    new *
3 public class DigitalVideoDisc extends Disc implements Playable{
4
    new *
5     public DigitalVideoDisc(String title) {
6         super(title);
7     }
    new *
8     public DigitalVideoDisc(String title, String category, float cost) {
9         super(title,category,cost);
10    }
    new *
11    public DigitalVideoDisc(String title, String category, String director, float cost) {
12        super(title,category,director,cost);
13    }
    new *
14    public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
15        super(title,category,director,length,cost);
16    }
    new *
17    public DigitalVideoDisc() {
18        super();
19    }
    Explain | Test | Document | Fix | Ask
    4 usages new *
20    public void play() {
21        System.out.println("Playing DVD: " + this.getTitle());
22        System.out.println("DVD length: " + this.getLength());
23    }
    Explain | Test | Document | Fix | Ask
    new *
24    @Override
25    public String toString() {
26        return "DigitalVideoDisc: ID:" + id + " Title:" + title + ", Length=" + length + ", Director:" + director + ", "
27            + "Category:" + category + " Cost=" + cost;
28    }
29 }
30

```


8. Update the Cart class

```
1 package lap_04.AimesProject;
2
3 > import ...
4
5
6 Explain | Test | Document | Fix | Ask
7 12 usages
8 public class Cart {
9     no usages
10    public static final int MAX_NUMBERS_ORDERED =20;
11    19 usages
12    private ArrayList<Media> itemsOrdered = new ArrayList<>();
13
14    Explain | Test | Document | Fix | Ask
15    2 usages
16    public void addMedia(Media media) {
17        if(itemsOrdered.contains(media)) {
18            System.out.println("the media has already in the cart");
19            return;
20        }
21        itemsOrdered.add(media);
22    }
23    Explain | Test | Document | Fix | Ask
24    1 usage
25    public void removeMedia(Media media) {
26        if(itemsOrdered.contains(media)) {
27            itemsOrdered.remove(media);
28            System.out.println("A media has been removed from cart");
29            return;
30        }
31        System.out.println("Can't find the media in cart");
32    }
33    Explain | Test | Document | Fix | Ask
34    no usages
35    float totalCost() {
36        float total = 0;
37        for(int i=0;i<itemsOrdered.size();i++) {
38            total += itemsOrdered.get(i).getCost();
39        }
40        return total;
41    }
42    Explain | Test | Document | Fix | Ask
43    // public void printOrder() {
44    //     System.out.println( "*****CART*****");
45    //     System.out.println("Ordered Items: ");
46    //     for(int i=0;i<itemsOrdered.size();i++) {
47    //         System.out.println(i+" "+itemsOrdered.get(i).getTitle()+
48    //             "- "+itemsOrdered.get(i).getCategory()+"- "+itemsOrdered.get(i).getDirector()+
49    //             "- "+itemsOrdered.get(i).getLength()+": "+itemsOrdered.get(i).getCost());
50    //     }
51    //     System.out.println("Total cost: "+totalCost());
52    //     System.out.println("*****");
53    // }
```

```

1 usage
45  ✓ public void placeOrder() {
46      System.out.println("An order is created");
47      itemsOrdered.removeAll(itemsOrdered);
48  }

1 usage
49  > public void printSize() { System.out.println("Number of media in cart:" + itemsOrdered.size()); }
    Explain | Test | Document | Fix | Ask

1 usage
52  ✓ public Media findMediaID(int id) {
53      ✓ for(int i=0;i<itemsOrdered.size();++i) {
54          if(id == itemsOrdered.get(i).getId());
55          return (itemsOrdered.get(i));
56      }
57      return null;
58  }
    Explain | Test | Document | Fix | Ask

3 usages
59  ✓ public Media findMediaTitle(String title) {
60      ✓ for(int i=0;i<itemsOrdered.size();i++) {
61          if(title.equals(itemsOrdered.get(i).getTitle()));
62          return (itemsOrdered.get(i));
63      }
64      return null;
65  }
    Explain | Test | Document | Fix | Ask

1 usage
66  ✓ public void sortTitleCost() {
67      Collections.sort(itemsOrdered,Media.COMPARE_BY_TITLE_COST);
68      System.out.println(itemsOrdered);
69  }
    Explain | Test | Document | Fix | Ask

1 usage
70  ✓ public void sortCostTitle() {
71      Collections.sort(itemsOrdered,Media.COMPARE_BY_COST_TITLE);
72      System.out.println(itemsOrdered);
73  }
74  }

```

9. Update the Store class

```
1 package lap_04.AimesProject;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import lap_04.AimesProject.Media.Media;
7
8 Explain | Test | Document | Fix | Ask
9 10 usages
10 public class Store {
11     1 usage
12     private static final int MAX_IN_STORE = 10000;
13     8 usages
14     private List<Media> itemsInStore = new ArrayList<>();
15
16     Explain | Test | Document | Fix | Ask
17     3 usages
18     public void addMedia(Media media) {
19         if(itemsInStore.size() == MAX_IN_STORE) {
20             System.out.println("The store is full");
21             return;
22         }
23         if(itemsInStore.contains(media)) {
24             System.out.println("This media has already in store");
25             return;
26         }
27         itemsInStore.add(media);
28         System.out.println("A media has been added in store");
29     }
30
31     Explain | Test | Document | Fix | Ask
32     1 usage
33     public void removeMedia(Media media) {
34         if(itemsInStore.contains(media)) {
35             itemsInStore.remove(media);
36             System.out.println("A media has been removed from store");
37             return;
38         }
39         System.out.println("Can't find the media in store");
40     }
41
42     Explain | Test | Document | Fix | Ask
43     4 usages
44     public Media findMedia(String title) {
45         for(int i=0; i<itemsInStore.size();i++) {
46             if(title.equals(itemsInStore.get(i).getTitle())) {
47                 return itemsInStore.get(i);
48             }
49         }
50         System.out.println("Can't find the media");
51         return null;
52     }
53 }
```

10. Update the Aims class

```
1 package lap_04.AimesProject;
2
3 import java.util.Scanner;
4 import lap_04.AimesProject.Cart;
5 import lap_04.AimesProject.Media.Book;
6 import lap_04.AimesProject.Media.CompactDisc;
7 import lap_04.AimesProject.Media.DigitalVideoDisc;
8 import lap_04.AimesProject.Media.Disc;
9 import lap_04.AimesProject.Media.Media;
10 import lap_04.AimesProject.Media.Playable;
11 import lap_04.AimesProject.Media.Track;
12 import lap_04.AimesProject.Store;
13
14 // AIMS PROJECT
15
16 public class Aims {
17     boolean exit;
18     Scanner kb = new Scanner(System.in);
19     public static void showMenu() {
20         System.out.println("AIMS: ");
21         System.out.println("-----");
22         System.out.println("1. View store");
23         System.out.println("2. Update store");
24         System.out.println("3. See current cart");
25         System.out.println("0. Exit");
26         System.out.println("-----");
27         System.out.println("Please choose a number: 0-1-2-3");
28     }
29     public static void storeMenu() {
30         System.out.println("Options: ");
31         System.out.println("-----");
32         System.out.println("1. See a media's details");
33         System.out.println("2. Add a media to cart");
34         System.out.println("3. Play a media");
35         System.out.println("4. See current cart");
36         System.out.println("0. Back");
37         System.out.println("-----");
38         System.out.println("Please choose a number: 0-1-2-3-4");
39     }
40 }
```

```

40 public static void mediaDetailsMenu() {
41     System.out.println("Options: ");
42     System.out.println("-----");
43     System.out.println("1. Add to cart");
44     System.out.println("2. Play");
45     System.out.println("0. Back");
46     System.out.println("-----");
47     System.out.println("Please choose a number: 0-1-2");
48 }
49
50 public static void cartMenu() {
51     System.out.println("Options: ");
52     System.out.println("-----");
53     System.out.println("1. Filter medias in cart");
54     System.out.println("2. Sort medias in cart");
55     System.out.println("3. Remove media from cart");
56     System.out.println("4. Play a media");
57     System.out.println("5. Place order");
58     System.out.println("0. Back");
59     System.out.println("-----");
60     System.out.println("Please choose a number: 0-1-2-3-4-5");
61 }
62
63 public static void UpdateStore() {
64     System.out.println("-----");
65     System.out.println("1. Add media to store");
66     System.out.println("2. Remove media to store");
67     System.out.println("0. Back");
68     System.out.println("-----");
69     System.out.println("Please choose a number: 0-1-2");
70 }
71
72 public static void addStoreMenu() {
73     System.out.println("-----");
74     System.out.println("1. Add Book");
75     System.out.println("2. Add CD");
76     System.out.println("3. Add DVD");
77     System.out.println("0. Back");
78     System.out.println("-----");
79     System.out.println("Please choose a number: 0-1-2-3");
80 }

```

[Explain](#) | [Test](#) | [Document](#) | [Fix](#) | [Ask](#)

1 usage

[Explain](#) | [Test](#) | [Document](#) | [Fix](#) | [Ask](#)

1 usage

[Explain](#) | [Test](#) | [Document](#) | [Fix](#) | [Ask](#)

1 usage

[Explain](#) | [Test](#) | [Document](#) | [Fix](#) | [Ask](#)

```

79 public void runMenu(Cart cart,Store store) {
80     while(!exit) {
81         boolean loop = true;
82         showMenu();
83         int choice = getInput( x: 3);
84         switch(choice) {
85             case 0:
86                 exit = true;
87                 break;
88             case 1:
89                 storeMenu();
90                 loop = true;
91                 int choice1 = getInput( x: 4);
92                 label0: while(loop==true) {
93                     switch(choice1) {
94                         case 0: loop = false; break;
95                         case 1: mediaDetails(store,card); break label0;
96                         case 2: addCart(card, store); break label0;
97                         case 3: playMediaStore(store); break label0;
98                         case 4: currentCard(card); break label0;
99                     }
100                 }
101                 break;
102             case 2:
103                 UpdateStore();
104                 loop = true;
105                 int choice2 = getInput( x: 2);
106                 label1: while(loop==true) {
107                     switch(choice2) {
108                         case 0:
109                             loop = false;
110                             break;
111                         case 1:
112                             addMediaStore(store);
113                             break label1;
114                         case 2:
115                             removeMediaStore(store);
116                             break label1;
117                     }
118                 }
119                 break;
120             case 3:
121                 currentCard(card);
122         }
123     }
124 }

```

 Explain | Test | Document | Fix | Ask

6 usages

```

125 private int getInput(int x) {
126     int choice = -1;
127     while(choice<0 || choice >x) {
128         try {
129             choice = Integer.parseInt(kb.nextLine());
130         }
131         catch(NumberFormatException e) {
132             System.out.println("Invalid selection. Please try again.");
133         }
134     }
135
136     return choice;
137 }
138
139 //-----
140 // function STORE MENU
141 // FUNCTION: outputTitle
142 // Lưu title từ bàn phím
143 // @return title
144 6 usages
145 private String outputTitle() {
146     System.out.println("Nhap title: ");
147     return(kb.nextLine());
148 }
149 // Explain | Test | Document | Fix | Ask
150 // 1.1 See a media's details
151 // Xem một thông tin media và in menu tiếp theo
152 // input: title in store
153 // output: print media information
154 // output: media's details menu
155 1 usage
156 @ private void mediaDetails(Store store, Cart cart) {
157     String title = outputTitle();
158     Media media = store.findMedia(title);
159     if(media!=null) {
160         System.out.println(media.toString());
161
162         mediaDetailsMenu();
163         int x = getInput( x: 2);
164         if(x==1) {
165             cart.addMedia(media); // 1.1.1 addCart tương đương 1.2
166         } else if(x==2) { //// 1.2.1 playMedia tương đương 1.3
167             if(media instanceof Playable) {
168                 ((Playable)media).play();
169             } else System.out.println("Can't find the media to play");
170         }
171     }
172 }

```

```

170 // 1.2. Add a media to cart
171 // input: title of media in store
172 // output: add to cart
173 // output: print numbers of dvd in cart
174 @ private void addCart(Cart cart, Store store) {
175     String title = outputTitle();
176     Media media = store.findMedia(title);
177     if(media != null) {
178         cart.addMedia(media);
179         System.out.println("You have added a media in cart");
180         cart.printSize();
181     }
182 }
183 // 1.3. Play a media in store
184 // input: title of media in store
185 // output: play media
186 @ private void playMediaStore(Store store) {
187     String title = outputTitle();
188     Media media = store.findMedia(title);
189     if(media instanceof Playable) {
190         ((Playable)media).play();
191     } else System.out.println("Can't find the media to play");
192 }
193 // 1.4. See current cart
194 // Tuong tu see current cart in showMenu
195 // END FUNCTION STORE
196 //-----
197
198 // -----
199 // 2. UPDATE STORE
200 // Function: InsertMedia
201 // set Title, Category, Cost for media
202 // use for addMedia
203 // 3 usages
204 @ private void insertMedia(Media media) {
205     System.out.println("Nhap title: ");
206     media.setTitle(kb.nextLine());
207     System.out.println("Nhap category: ");
208     media.setCategory(kb.nextLine());
209     System.out.println("Nhap cost: ");
210     media.setCost(Float.parseFloat(kb.nextLine()));
211 }

```



```

214 @ private void insertDisc(Disc disc) {
215     System.out.println("Nhập director: ");
216     disc.setDirector(kb.nextLine());
217     System.out.println("Nhập length: ");
218     disc.setLength(Integer.parseInt(kb.nextLine()));
219 }
    Explain | Test | Document | Fix | Ask
220 // 2.1 Add a media to store
221 // three Option: add Book, add CD or add DVD
222 // Book: Title, Category, Cost, Authors_List
223 // CD: Title, Category, Cost, Director, Length, Tracks_List(titleTrack, length)
224 // DVD: Title, Category, Cost, Director, Length
    1 usage
225 private void addMediaStore(Store store) {
226     addStoreMenu();
227     boolean loop = true;
228     int choice = getInput(x: 3);
229     label: while(loop== true) {
230         switch(choice) {
231             case 0: loop = false; break;
232             // add book
233             case 1:
234                 Book bk = new Book();
235                 insertMedia(bk);
236                 boolean x=true;
237                 while(x==true) {
238                     System.out.println("Nhập author:");
239                     bk.addAuthor(kb.nextLine());
240                     System.out.println("Bạn có muốn nhập author tiếp không?(True/False)");
241                     x =Boolean.parseBoolean(kb.nextLine());
242                 }
243                 store.addMedia(bk);
244                 break label;
245             // add cd
246             case 2:
247                 CompactDisc cd = new CompactDisc();
248                 insertMedia(cd);
249                 System.out.println("Nhập director: ");
250                 cd.setDirector(kb.nextLine());
251                 System.out.println("Nhập tracks:");
252                 int i=1;
253                 x=true;
254                 while(x==true) {
255                     System.out.println("Tracks "+i);
256                     System.out.println("Title:");
257                     String titleTrack = kb.nextLine();
258                     System.out.println("Length: ");
259                     int lengthTrack = Integer.parseInt(kb.nextLine());
260                     Track track = new Track(titleTrack,lengthTrack);

```

```

261         cd.addTrack(track);
262         i++;
263         System.out.println("Ban co muon nhap track tiep khong?(True/False)");
264         x = Boolean.parseBoolean(kb.nextLine());
265     }
266     store.addMedia(cd);
267     break label;
268     // add dvd
269     case 3:
270         DigitalVideoDisc dvd = new DigitalVideoDisc();
271         insertMedia(dvd);
272         insertDisc(dvd);
273         store.addMedia(dvd);
274         break label;
275     }
276 }
277 }
278
279 // 2.2 Remove Media
280 // Remove media from store
281 // Lưu title từ bàn phím rồi tìm Media và xóa
282 @ 1 usage
283 private void removeMediaStore(Store store) {
284     String title = outputTitle();
285     Media media = store.findMedia(title);
286     store.removeMedia(media);
287 }
288 // END FUNCTION UPDATE STORE
289 //-----
290
291 Explain | Test | Document | Fix | Ask
292 // 3. SEE CURRENT CART
293 2 usages
294 private void currentCart(Cart cart) {
295     cartMenu();
296     boolean loop = true;
297     int choice1 = getInput(x: 5);
298     label2: while(loop==true) {
299         switch(choice1) {
300             case 0: loop = false; break label2;
301             case 1:
302                 filterMedia(cart);
303                 break label2;
304             case 2:
305                 sortMedia(cart);
306                 break label2;
307             case 3:
308                 removeMediaCart(cart);
309                 break label2;
310             case 4:

```

```

307         case 4:
308             playMediaCart(cart);
309             break label2;
310         case 5:
311             cart.placeOrder();
312             break label2;
313     }
314 }
315 }
316
317 Explain | Test | Document | Fix | Ask
318 // 3.1 Filter media
319 // Filter by ID or TITLE
320 // Nhập tu ban phim id hoặc title
321 // output: A media information
322 1 usage
323 private void filterMedia(Cart cart) {
324     System.out.println("Filter by id or title:");
325     String ans = kb.nextLine();
326     if(ans.equals("id")||ans.equals("ID")) {
327         System.out.println("Nhập ID:");
328         System.out.println(cart.findMediaID(Integer.parseInt(kb.nextLine())));
329     } else if(ans.equals("title")||ans.equals("Title")) {
330         System.out.println("Nhập Title: ");
331         System.out.println(cart.findMediaTitle(kb.nextLine()));
332     } else System.out.println("Nhập sai.");
333 }
334
335 Explain | Test | Document | Fix | Ask
336 // 3.2 Sort medias in cart
337 // Sort by TitleCost or CostTitle
338 // output: print Cart Media List by sort
339 1 usage
340 private void sortMedia(Cart cart) {
341     System.out.println("Sort by cost or title:");
342     String ans = kb.nextLine();
343     if(ans.equals("cost")||ans.equals("Cost")) {
344         cart.sortCostTitle();
345     } else if(ans.equals("title")||ans.equals("Title")) {
346         cart.sortTitleCost();
347     } else System.out.println("Nhập sai.");
348 }
349
350 Explain | Test | Document | Fix | Ask
351 // 3.3 Remove media from cart
352 // Tương tự remove from Store
353 1 usage
354 @ private void removeMediaCart(Cart cart) {
355     String title = outputTitle();
356     Media media = cart.findMediaTitle(title);
357     cart.removeMedia(media);

```

```

347 @  ✓ private void removeMediaCart(Cart cart) {
348         String title = outputTitle();
349         Media media = cart.findMediaTitle(title);
350         cart.removeMedia(media);
351     }
    Explain | Test | Document | Fix | Ask
352  ✓ // 3.4 Play a media in cart
353 // Tương tự play from Store
    1 usage
354 @  ✓ private void playMediaCart(Cart cart) {
355         String title = outputTitle();
356         Media media = cart.findMediaTitle(title);
357  ✓     if(media instanceof Playable) {
358         ((Playable)media).play();
359     } else System.out.println("Can't find the media to play");
360     }
361
    Explain | Test | Document | Fix | Ask
362 //MAIN
363 ▶  ✓ public static void main(String[] args) {
364         Aims menu = new Aims();
365         Cart cart = new Cart();
366         Store store = new Store();
367         menu.runMenu(cart,store);
368     }
369
370 }

```

11. Demo Aims

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

AIMS:
-----
1. View store
2. Update store
3. See current cart
0. Exit
-----
Please choose a number: 0-1-2-3
2
-----
1. Add media to store
2. Remove media to store
0. Back
-----
Please choose a number: 0-1-2
```

12. Update the class diagram

