

BÁO CÁO LAB 04: INHERITANCE AND POLYMORPHISM

Trần Khánh Quỳnh – 20225762

I. Trả lời câu hỏi

Phần 9:

1. Xác định các lớp nào là tập hợp (aggregates) của các lớp khác?

- Cart là tập hợp của Media
- CompactDisc là tập hợp của Track

Phần 10:

- Nếu đối tượng truyền vào không phải là thể hiện của lớp Media, khi gọi phương thức equals() trong lớp Media, bạn cần kiểm tra kiểu của đối tượng đó trước khi thực hiện so sánh. Cụ thể trong phương thức equals(), bạn có thể sử dụng instanceof để kiểm tra xem đối tượng truyền vào có phải là một thể hiện của lớp Media hay không. Nếu không, phương thức sẽ trả về false, vì hai đối tượng không thể so sánh được nếu chúng không phải cùng kiểu.

Phần 12:

1. Lớp nào nên implement interface Comparable?

Interface Comparable nên được implement trong các lớp cần sắp xếp theo một quy tắc mặc định, ví dụ như lớp Media và các lớp con của nó như Book, CompactDisc, DigitalVideoDisc. Điều này giúp các đối tượng của các lớp này có thể được so sánh và sắp xếp trong các cấu trúc dữ liệu như TreeSet hoặc khi sử dụng Collections.sort().

2. Cách cài đặt phương thức compareTo() trong các lớp?

Phương thức compareTo() yêu cầu bạn định nghĩa cách so sánh đối tượng hiện tại với đối tượng khác. Ví dụ, trong lớp Media, có thể so sánh các đối tượng theo tiêu đề hoặc giá. Cách so sánh này phải đảm bảo thứ tự nhất quán và hợp lý.

3. Có thể có hai quy tắc sắp xếp của các mục (theo tiêu đề rồi đến giá và theo giá rồi đến tiêu đề) nếu sử dụng Comparable không?

Không, trong một phương thức compareTo() duy nhất, bạn chỉ có thể thực hiện một quy tắc sắp xếp. Nếu bạn cần sử dụng nhiều quy tắc sắp xếp khác nhau (ví dụ: theo tiêu đề rồi đến giá và theo giá rồi đến tiêu đề), bạn nên sử dụng Comparator thay vì Comparable.

4. DVD có một quy tắc sắp xếp khác biệt so với các loại media khác, tức là theo tiêu đề, sau đó là chiều dài giảm dần, rồi đến giá. Làm thế nào để thay đổi mã nguồn để hỗ trợ điều này?

Để DVD có một quy tắc sắp xếp khác biệt, bạn cần override phương thức compareTo() trong lớp DigitalVideoDisc để thực hiện quy tắc sắp xếp riêng cho DVD. Trong trường hợp này, sắp xếp sẽ theo thứ tự: tiêu đề -> chiều dài giảm dần -> giá.

II. Code

1. Book class

```
1 package hust.soict.dsai.aims.media;
2
3 import java.util.*;
4
5 public class Book extends Media{
6     private List<String> authors = new ArrayList<String>();
7
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     //addAuthor
18     public void addAuthor(String authorName) {
19         for(String author : authors) {
20             if(author == authorName) {
21                 System.out.println("This author already exist!");
22                 return;
23             }
24         }
25         authors.add(authorName);
26         System.out.println("Author was added sucessfully!");
27     }
28
29
30     //removeAuthor
31     public void removeAuthor(String authorName) {
32         for(String author : authors) {
33             if(author == authorName) {
34                 authors.remove(authorName);
35                 System.out.println("Author was removed sucessfully!");
36             }
37         }
38         System.out.println("Author's name not found!");
39     }
40 }
```

```

41 ● @Override
42 public String toString() {
43     return "Book - " + getTitle() + " - " + getCategory() + " - " + getAuthors() +
44         " - Cost: $" + getCost();
45 }
46 ●
47 public Book() {
48     super();
49     // TODO Auto-generated constructor stub
50 }
51 ●
52 public Book(String title, String category, float cost) {
53     super(title, category, cost);
54     // TODO Auto-generated constructor stub
55 }
56 ●
57 public Book(String title, String category) {
58     super(title, category);
59     // TODO Auto-generated constructor stub
60 }
61 ●
62 public Book(String title) {
63     super(title);
64     // TODO Auto-generated constructor stub
65 }
66
67
68
69
70
71

```

Tạo class Book trong package hust.soict.dsai.aims.media với các trường là id, title, category, cost và authors. Sau đó tạo các phương thức addAuthor() và removeAuthor() cho class.

2. *DigitalVideoDisc class*

```

1 package hust.soict.dsai.aims.media;
2
3 public class DigitalVideoDisc extends Disc implements Playable{
4     private String director;
5     private int length;
6
7
8 ● public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
9     super(title,category,cost);
10    this.director = director;
11    this.length = length;
12 }
13
14 ● public DigitalVideoDisc(String title, String category, String director, float cost) {
15     super(title,category,cost);
16     this.director = director;
17 }
18
19 ● public DigitalVideoDisc(String title, String category, float cost) {
20     super(title,category,cost);
21 }
22
23 ● public String getDirector() {
24     return director;
25 }
26 ● public int getLength() {
27     return length;
28 }
29
30 ● @Override
31 public void play() {
32     // TODO Auto-generated method stub
33     System.out.println("Playing DVD: " + this.getTitle());
34     System.out.println("DVD length: " + this.getLength());
35 }
36
37
38 ● @Override

```

```

38●  @Override
39  public String toString() {
40      return "DVD - " + getTitle() + " - " + getCategory() + " - Director: " + getDirector() +
41          " - Length: " + getLength() + " mins - Cost: $" + getCost();
42  }
43  }
44

```

- Sau khi tạo class Media thì xóa các trường id, title, category, cost, chuyển class Book extends Media
- Sau khi tạo class Disc thì chuyển thành class DigitalVideoDisc extends Disc
- Sau khi tạo xong Playable interface thì áp dụng nó cho class DigitalVideoDisc
- Tạo play() cho DigitalVideoDisc class

3. *Disc class*

```

1  package hust.soict.dsai.aims.media;
2
3  public class Disc extends Media {
4      private int length;
5      private String director;
6
7      //getters
8●  public int getLength() {
9      return length;
10 }
11
12●  public String getDirector() {
13      return director;
14 }
15
16  //Constructor
17●  public Disc(String title, String category, float cost) {
18      super(title, category, cost);
19 }
20
21
22●  public Disc(String title, String category, float cost, int length, String director) {
23      super(title, category, cost);
24      this.length = length;
25      this.director = director;
26 }
27
28●  public Disc() {
29      // TODO Auto-generated constructor stub
30 }
31●  @Override
32  public String toString() {
33      return "Disc - " + getTitle() + " - " + getCategory() + " - Director: " + getDirector() +
34          " - Length: " + getLength() + " mins - Cost: $" + getCost();
35 }
36
37 }
38

```

- Class Disc có hai trường là length và director
- Sau khi tạo xong Media class thì chuyển Disc extends đến Media

4. *Track class*

```

1 package hust.soict.dsai.aims.media;
2
3 public class Track implements Playable{
4     private String title;
5     private int length;
6
7     //Getter
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        super();
19        this.title = title;
20        this.length = length;
21    }
22
23    public Track() {
24        // TODO Auto-generated constructor stub
25    }
26
27    @Override
28    public void play() {
29        // TODO Auto-generated method stub
30        System.out.println("Playing DVD: " + this.getTitle());
31        System.out.println("DVD length: " + this.getLength());
32    }
33 }

```

```

33 }
34
35 @Override
36 public boolean equals(Object object) {
37     if (object instanceof Track) {
38         Track otherTrack = (Track) object;
39         return this.title.equalsIgnoreCase(otherTrack.getTitle()) && this.length == otherTrack.getLength();
40     }
41     return false;
42 }
43
44 @Override
45 public String toString() {
46     return "Track - " + getTitle() + " - " + getLength() + " mins";
47 }

```

- Class Track có hai trường title và length.
- Sau khi tạo xong Playable interface thì áp dụng cho class Track.
- Tạo play() cho classTrack

5. *CompactDisc class*

```

1 package hust.soict.dsai.aims.media;
2
3 import java.util.ArrayList;
4
5 public class CompactDisc extends Disc implements Playable {
6     private String artist;
7     private ArrayList<Track> tracks = new ArrayList<Track>();
8
9     //getter for artist
10    public String getArtist() {
11        return artist;
12    }
13
14    //Constructor
15    public CompactDisc(String title, String category, float cost, String artist, ArrayList<Track> tracks) {
16        super(title, category, cost);
17        this.artist = artist;
18        this.tracks = tracks;
19    }
20
21    public CompactDisc(String title, String category, float cost, int length, String director) {
22        super(title, category, cost, length, director);
23        // TODO Auto-generated constructor stub
24    }
25
26    public CompactDisc(String title, String category, float cost) {
27        super(title, category, cost);
28        // TODO Auto-generated constructor stub
29    }
30
31    //addTrack
32    public void addTrack(Track track) {
33        if(tracks.contains(track)) {
34            System.out.println(track.getTitle() + " already exist in the CD");
35        }
36        tracks.add(track);
37        System.out.println("Track " + track.getTitle() + " added sucessfully!");
38    }
39

```

```

41    //removeTrack
42    public void removeTrack(Track track) {
43        if(tracks.contains(track)) {
44            System.out.println("Track " + track.getTitle() + " remove successfully!");
45            tracks.remove(track);
46        }
47        System.out.println("Track not found!");
48    }
49
50    //getLength method
51    public int totalLength() {
52        int total = 0;
53        for(Track track : tracks) {
54            total += track.getLength();
55        }
56        return total;
57    }
58
59    }
60
61    public CompactDisc() {
62        // TODO Auto-generated constructor stub
63    }
64
65    @Override
66    public void play() {
67        // TODO Auto-generated method stub
68        System.out.println("Playing DVD: " + this.getTitle());
69        System.out.println("DVD length: " + this.getLength());
70
71        for(Track track : tracks) {
72            track.play();
73        }
74    }
75
76

```

```

76
77    @Override
78    public String toString() {
79        return "CD - " + getTitle() + " - " + getCategory() + " - Director: " + getDirector() +
80            " - Length: " + getLength() + " mins - Cost: $" + getCost();
81    }
82
83
84
85
86
87 }
88

```

- Tạo class CompactDisc, thêm các trường là artist và tracks. Sau đó thêm các phương thức addTrack() để thêm 1 track mới vào và removeTrack() để xóa 1 track đã tồn tại.
- Tạo totalLength() để đo độ dài tổng của CD (1CD có nhiều track).
- Sau đó, áp dụng Playable interface.

6. *Media class*

```

1 package hust.soict.dsai.aims.media;
2
3 import java.util.Comparator;
4
5 public abstract class Media {
6     public static final Comparator<Media> COMPARE_BY_TITLE_COST
7         = new MediaComparatorByTitleCost();
8     public static final Comparator<Media> COMPARE_BY_COST_TITLE
9         = new MediaComparatorByCostTitle();
10
11     private int id;
12     private String title;
13     private String category;
14     private float cost;
15     private static int nbDigitalVideoDiscs = 0;
16     //Getters & setters
17
18
19     public Media(String title, String category, float cost) {
20         super();
21         this.title = title;
22         this.category = category;
23         this.cost = cost;
24         nbDigitalVideoDiscs++;
25         this.id = nbDigitalVideoDiscs;
26     }
27
28     public Media(String title, String category) {
29         super();
30         this.title = title;
31         this.category = category;
32     }
33
34     public Media(String title) {
35         super();
36         this.title = title;
37     }
38

```

```

40
41● public int getId() {
42    return id;
43}
44
45● public void setId(int id) {
46    this.id = id;
47}
48
49● public String getTitle() {
50    return title;
51}
52
53● public void setTitle(String title) {
54    this.title = title;
55}
56
57● public String getCategory() {
58    return category;
59}
60
61● public void setCategory(String category) {
62    this.category = category;
63}
64
65● public float getCost() {
66    return cost;
67}
68
69● public void setCost(float cost) {
70    this.cost = cost;
71}
72
73● public static int getNbDigitalVideoDiscs() {
74    return nbDigitalVideoDiscs;
75}
76
77● public static void setNbDigitalVideoDiscs(int nbDigitalVideoDiscs) {
78
79●     public static void setNbDigitalVideoDiscs(int nbDigitalVideoDiscs) {
80        Media.nbDigitalVideoDiscs = nbDigitalVideoDiscs;
81    }
82
83● @Override
84 public boolean equals(Object object) {
85     if (object instanceof Media) {
86         Media other = (Media) object;
87         return this.title.equalsIgnoreCase(other.getTitle());
88     }
89     return false;
90 }
91
92● public Media() {
93    // TODO Auto-generated constructor stub
94 }
95
96● @Override
97 public String toString() {
98     return "Media - " + getTitle() + " - " + getCategory() + " - Cost: $" + getCost();
99 }
100 }
101

```

- Tạo Media class với các trường id, title, category và cost
- Thêm các comparators thành trường trong Media Class để sort media trong cart

7. Playable interface


```

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

```

Tạo Playable interface và áp dụng cho CompactDisc, DigitalVideoDisc và Track

8. Cart

```

1 package hust.soict.dsai.aims.cart;
2
3 import java.util.ArrayList;
4
5 public class Cart {
6     public static final int MAX_NUMBERS_ORDERED = 20;
7     private ArrayList<Media> itemsOrdered = new ArrayList<Media>();
8
9     public float totalCost() {
10         float sum = 0.00f;
11         for(Media itemOrdered : itemsOrdered) {
12             sum += itemOrdered.getCost();
13         }
14         return sum;
15     }
16
17     public ArrayList<Media> getItemsOrdered() {
18         return itemsOrdered;
19     }
20
21     //Print the list of ordered items, price and total price
22     public void printCart() {
23         System.out.println("*****CART*****");
24         System.out.println("Ordered Items:");
25         float totalcost = 0;
26         for(Media itemOrdered : itemsOrdered){
27             int i = 0;
28             System.out.printf("%d. %s\n",itemOrdered.getId(),itemOrdered);
29             totalcost += itemOrdered.getCost();
30         }
31         System.out.println(String.format("Total cost: %.2f", totalcost));
32         System.out.println("***** ");
33     }
34
35     //Search for DVDs in the cart by ID
36     public Media searchByID(int id) {
37         for(Media itemOrdered : itemsOrdered) {
38             if(itemOrdered.getId() == id) {
39                 System.out.println("DVD found:");
40             }
41         }
42     }
43 }

```

```

41         System.out.println("DVD found:");
42         System.out.println(itemOrdered);
43         return itemOrdered;
44     }
45 }
46
47     System.out.println("No match is found!");
48     return null;
49 }
50
51 //Search for DVDs by title
52 public Media searchByTitle(String title) {
53     for(Media itemOrdered : itemsOrdered) {
54         if(itemOrdered.getTitle() == title) {
55             System.out.println("DVD found:");
56             System.out.println(itemOrdered);
57             return itemOrdered;
58         }
59     }
60
61     System.out.println("No match is found!");
62     return null;
63 }
64
65 //AddMedia
66 public void addMedia(Media media) {
67     if(itemsOrdered.size() > MAX_NUMBERS_ORDERED) {
68         System.out.println("The cart is full!");
69     }
70     if(itemsOrdered.contains(media)) {
71         System.out.println(media.getTitle() + " already exists!");
72         return;
73     }
74     itemsOrdered.add(media);
75     System.out.println(media.getTitle() + " has been added");
76 }
77
78 //removeMedia
79 public void removeMedia(Media media) {
80     if(!itemsOrdered.contains(media)) {
81         System.out.println(media.getTitle() + " is not found");
82     }
83     itemsOrdered.remove(media);
84     System.out.println(media.getTitle() + " is removed successfully");
85 }
86
87 public int getNumItemsInCart() {
88     return itemsOrdered.size();
89 }
90 }
91
92 }
93
94

```

Sửa class Cart để nó hoạt động với Media: thay itemsOrdered bằng ArrayList, tạo thêm addMedia() và removeMedia (); update totalCost() để nó làm việc với arrayList

9. Store

```

1 package hust.soict.dsai.aims.store;
2
3 import java.util.ArrayList;
4
5
6
7 public class Store {
8     private static final int MAX_ITEMS = 100;
9     private ArrayList<Media> itemsinStore = new ArrayList<Media>();
10
11     public void addMedia(Media media) {
12         if (itemsinStore.size() >= MAX_ITEMS) {
13             System.out.println("The store is full!");
14             return;
15         }
16         if(itemsinStore.contains(media)) {
17             System.out.println(media.getTitle() + " already exists!");
18             return;
19         }
20         itemsinStore.add(media);
21         System.out.println(media.getTitle() + " is added successfully to the store!");
22     }
23
24     public void removeMedia(Media media) {
25         if(!itemsinStore.contains(media)) {
26             System.out.println(media.getTitle() + " is not in the store");
27             return;
28         }
29         itemsinStore.remove(media);
30         System.out.println(media.getTitle() + " is removed successfully");
31     }
32
33     //Search for DVDs in the cart by ID
34     public Media searchByID(int id) {
35         for(Media iteminStore : itemsinStore) {
36             if(iteminStore.getId() == id) {
37                 System.out.println("Item found:");
38                 System.out.println(iteminStore);
39                 return iteminStore;
40             }

```

```

39         return iteminStore;
40     }
41 }
42
43 System.out.println("No match is found!");
44 return null;
45 }
46
47 //Search for DVDs by title
48 public Media searchByTitle(String title) {
49     for(Media iteminStore : itemsinStore) {
50         if(iteminStore.getTitle() == title) {
51             System.out.println("DVD found:");
52             System.out.println(iteminStore);
53             return iteminStore;
54         }
55     }
56
57     System.out.println("No match is found!");
58     return null;
59 }
60
61 public void printStore() {
62     System.out.println("*****STORE*****");
63     int i = 0;
64
65     for (Media itemInStore : itemsinStore) {
66         i++;
67         System.out.printf("%d. %s\n", i, itemInStore);
68     }
69
70     System.out.println("***** ");
71 }
72
73 }

```

Sửa class Store, tương tự class Cart

10.Unique item in a list

Trong Class Media:

```
80
81  @Override
82  public boolean equals(Object object) {
83      if (object instanceof Media) {
84          Media other = (Media) object;
85          return this.title.equalsIgnoreCase(other.getTitle());
86      }
87      return false;
88  }
89
```

Trong Class Track:

```
35  @Override
36  public boolean equals(Object object) {
37      if (object instanceof Track) {
38          Track otherTrack = (Track) object;
39          return this.title.equalsIgnoreCase(otherTrack.getTitle()) && this.length == otherTrack.getLength();
40      }
41      return false;
42  }
43
```

11.Sort media in the cart

```
1  package hust.soict.dsai.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          int costComparision = Float.compare(o1.getCost(), o2.getCost());
10         if (costComparision != 0) {
11             return costComparision;
12         }
13         return o1.getTitle().compareToIgnoreCase(o2.getTitle());
14     }
15 }
16
17
18
```

```
1  package hust.soict.dsai.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          int titleComparision = o1.getTitle().compareToIgnoreCase(o2.getTitle());
10         if (titleComparision != 0) {
11             return titleComparision;
12         }
13         return Float.compare(o1.getCost(), o2.getCost());
14     }
15 }
16
17
```

12. Tạo console

```
15
16 public class Aims {
17     //Main
18     public static void showMenu() {
19         System.out.println("AIMS: ");
20         System.out.println("-----");
21         System.out.println("1.View store");
22         System.out.println("2.Update store");
23         System.out.println("3.See current cart");
24         System.out.println("0.Exit");
25         System.out.println("-----");
26         System.out.println("Please choose a number: 0-1-2-3");
27     }
28 }
29
30 public static void handleShowMenu(Scanner scanner, Store store, Cart cart) {
31     while(true) {
32         showMenu();
33         int choice = scanner.nextInt();
34         scanner.nextLine();
35
36         switch(choice) {
37             case 1:
38                 handleStoreMenu(scanner, store, cart);
39                 break;
40             case 2:
41                 handleUpdateStore(scanner, store, cart);
42                 break;
43             case 3:
44                 handleCartMenu(scanner, store, cart);
45                 break;
46             case 0:
47                 return;
48             default:
49                 System.out.println("Invalid choice. Please try again.");
50         }
51     }
52 }
```

```

//Option 1
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 handleStoreMenu(Scanner scanner, Store store, Cart cart) {
    while(true) {
        storeMenu();
        int choice = scanner.nextInt();
        scanner.nextLine();

        switch(choice) {
            case 1:
                handleMediaDetailsMenu(scanner, store, cart);
                break;
            case 2:
                addMediaToCart(scanner, cart, store);
                break;
            case 3:
                playMedia(scanner, cart, store);
                break;
            case 4:
                handleCartMenu(scanner, store, cart);
                break;
            case 0:
                return;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    }
}

```

```

89         return;
90     default:
91         System.out.println("Invalid choice. Please try again.");
92     }
93
94
95     }
96
97 }
98
99 //See media details
100 public static void mediaDetailsMenu() {
101     System.out.println("Options: ");
102     System.out.println("-----");
103     System.out.println("1.Add to cart");
104     System.out.println("2.Play");
105     System.out.println("0.Back");
106     System.out.println("-----");
107     System.out.println("Please choose a number: 0-1-2");
108
109 }
110
111 public static void handleMediaDetailsMenu(Scanner scanner, Store store, Cart cart) {
112     while(true) {
113         mediaDetailsMenu();
114         int choice = scanner.nextInt();
115         scanner.nextLine();
116
117         switch(choice) {
118             case 1:
119                 addMediatoCart(scanner, cart, store);
120                 break;
121             case 2:
122                 playMedia(scanner, cart, store);
123                 break;
124             case 0:
125                 return;
126             default:
127                 System.out.println("Invalid choice. Please try again.");

```

```

127         System.out.println("Invalid choice. Please try again.");
128     }
129
130
131     }
132
133 }
134
135 //Add media to cart
136 public static void addMediatoCart(Scanner scanner, Cart cart, Store store) {
137     System.out.println("Enter the title of the media to add to the cart:");
138     String title = scanner.nextLine();
139
140     Media media = store.searchByTitle(title);
141     //Nếu tìm thấy media trong store --> media khác null
142     if (media != null) {
143         cart.addMedia(media);
144         System.out.println("Media " + media.getTitle() + " has been added to the cart.");
145         System.out.println("Number of items in cart: " + cart.getNumItemsInCart()); // Display numb
146     } else {
147         //nếu media = null
148         System.out.println("Media not found.");
149     }
150 }
151
152 public static void removeMediafromCart(Scanner scanner, Cart cart) {
153     System.out.println("Enter the title of the media to remove from the cart:");
154     String title = scanner.nextLine();
155
156     Media media = cart.searchByTitle(title);
157     //Nếu tìm thấy media trong cart
158     if (media != null) {
159         cart.removeMedia(media);
160         System.out.println("Media " + media.getTitle() + " has been removed from the cart.");
161     } else {
162         System.out.println("Media " + title + " not found.");
163     }
164 }

```

```

166
167● public static void removeMediafromStore(Scanner scanner, Store store) {
168     System.out.println("Enter the title of the media to remove from the store:");
169     String title = scanner.nextLine();
170
171     Media media = store.searchByTitle(title);
172     //Nếu tìm thấy media trong store
173     if (media != null) {
174         store.removeMedia(media);
175         System.out.println("Media " + media.getTitle() + " has been removed from the store.");
176     } else {
177         System.out.println("Media " + title + " not found.");
178     }
179 }
180
181 //Play a media
182● public static void playMedia(Scanner scanner, Cart cart, Store store) {
183     System.out.println("Enter the title of the media to play:");
184     String title = scanner.nextLine();
185
186     Media media = store.searchByTitle(title);
187     if(media != null) {
188         if(media instanceof Playable) {
189             ((Playable) media).play();
190         } else {
191             System.out.println("Cannot play");
192         }
193     }
194     System.out.println("Media not found");
195 }
196
197
198
199 //Option 2
200● public static void updateStore() {
201     System.out.println("Options: ");
202     System.out.println("-----");
203     System.out.println("1.Add a media");

```

```

5     System.out.println("0.Back");
6     System.out.println("-----");
7     System.out.println("Please choose a number: 0-1-2");
8
9 }
0
1● public static void handleUpdateStore(Scanner scanner, Store store, Cart cart) {
2     while(true) {
3         updateStore();
4         int choice = scanner.nextInt();
5         scanner.nextLine();
6         switch(choice) {
7             case 1:
8                 System.out.println("Enter media title: ");
9                 String title = scanner.nextLine();
0                 System.out.println("Enter media type:");
1                 String type = scanner.nextLine();
2                 if(type.equals("Book")) {
3                     System.out.print("Enter category, cost: ");
4                     String category = scanner.nextLine();
5                     float cost = scanner.nextFloat();
6                     Book book = new Book(title,category, cost);
7                     store.addMedia(book);
8                 } else if(type.equals("CD")) {
9                     System.out.print("Enter category, cost: ");
0                     String category = scanner.nextLine();
1                     float cost = scanner.nextFloat();
2                     CompactDisc cd = new CompactDisc(title,category, cost);
3                     store.addMedia(cd);
4                 } else if(type.equals("DVD")) {
5                     System.out.print("Enter category, cost: ");
6                     String category = scanner.nextLine();
7                     float cost = scanner.nextFloat();
8                     DigitalVideoDisc dvd = new DigitalVideoDisc(title,category, cost);
9                     store.addMedia(dvd);
0                 }
1                 break;
2             case 2:

```



```

241         break;
242     case 2:
243         removeMediafromStore(scanner, store);
244         break;
245     case 0:
246         return;
247     default:
248         System.out.println("Invalid choice. Please try again.");
249
250     }
251 }
252 }
253
254 //Option 3: see current cart
255 public static void cartMenu() {
256     System.out.println("Options: ");
257     System.out.println("-----");
258     System.out.println("1.Filter medias in cart");
259     System.out.println("2.Sort medias in cart");
260     System.out.println("3.Remove media from cart");
261     System.out.println("4.Play a media");
262     System.out.println("5.Place order");
263     System.out.println("0.Back");
264     System.out.println("-----");
265     System.out.println("Please choose a number: 0-1-2-3-4-5");
266 }

```

```

6
7
8 public static void handleCartMenu(Scanner scanner, Store store, Cart cart) {
9     while(true) {
10         cartMenu();
11         int choice = scanner.nextInt();
12         scanner.nextLine();
13         switch(choice) {
14             case 1:
15                 filterMenu(scanner, store, cart);
16                 break;
17             case 2:
18                 sortCart(scanner, store, cart);
19                 break;
20             case 3:
21                 removeMediafromCart(scanner, cart);
22                 break;
23             case 4:
24                 playMedia(scanner, cart, store);
25                 break;
26             case 5:
27                 placeOrder(cart);
28                 break;
29             case 0:
30                 return;
31             default:
32                 System.out.println("Invalid choice. Please try again.");
33
34         }
35     }
36 }
37
38
39 public static void placeOrder(Cart cart) {
40     System.out.println("Order placed successfully! Total Cost: $" + cart.totalCost());
41     cart.getItemsOrdered().clear();
42 }

```

```

304● public static void sortCart(Scanner scanner, Store store, Cart cart) {
305     while(true) {
306         System.out.println("Sort by: ");
307         System.out.println("1. Title");
308         System.out.println("2. Cost");
309         int sortChoice = scanner.nextInt();
310         scanner.nextLine();
311         switch (sortChoice) {
312             case 1:
313                 Collections.sort(cart.getItemsOrdered(), new MediaComparatorByTitleCost());
314                 break;
315             case 2:
316                 Collections.sort(cart.getItemsOrdered(), new MediaComparatorByCostTitle());
317                 break;
318             default:
319                 System.out.println("Invalid choice. Please try again.");
320         }
321         cart.printCart();
322     }
323 }

```

```

324● public static void filterMenu(Scanner scanner, Store store, Cart cart) {
325     while(true) {
326         System.out.println("Options: ");
327         System.out.println("-----");
328         System.out.println("1.Filter by ID");
329         System.out.println("2.Filter by title");
330         System.out.println("0.Back");
331         System.out.println("-----");
332         System.out.println("Please choose a number: 0-1-2");
333
334         int choice = scanner.nextInt();
335         scanner.nextLine();
336         switch(choice) {
337             case 1:
338                 int id = scanner.nextInt();
339                 cart.searchByID(id);
340                 break;
341             case 2:
342                 String title = scanner.next();
343                 scanner.nextLine();
344                 cart.searchByTitle(title);
345                 break;
346             case 0:
347                 return;
348             default:
349                 System.out.println("Invalid choice. Please try again.");
350         }
351     }
352 }
353 }

```

```
354
355● public static void filterCartMenu() {
356     System.out.println("Options: ");
357     System.out.println("-----");
358     System.out.println("1.Filter by ID");
359     System.out.println("2.Filter by title");
360     System.out.println("0.Back");
361     System.out.println("-----");
362     System.out.println("Please choose a number: 0-1-2-3-4-5");
363 }
364
365
366● public static void main(String[] args) {
367     Scanner scanner = new Scanner(System.in);
368     Store store = new Store();
369     Cart cart = new Cart();
370     handleShowMenu(scanner, store, cart);
371
372
373
374 }
375 }
```