

OOP Lab 04

Họ và tên: Lê Bá Ngọc Hiểu

MSSV: 20225627

9. Constructors of whole classes and parent classes

Question 1: Which classes are aggregates of other classes? Checking all constructors of whole classes if they initialize for their parts?

Answer: Trong AimsProject, lớp media là một lớp tổng hợp (aggregate) của Media (bao gồm các lớp con như Book, DigitalVideoDisc) vì nó chứa danh sách các đối tượng Media. Các đối tượng Media có thể tồn tại độc lập với Order và được thêm vào thông qua phương thức như addMedia() hoặc thông qua constructor nhận danh sách Media làm tham số. Ngoài ra, lớp Disc cũng được xem là tổng hợp các thuộc tính length và director, mặc dù đây là các phần tử logic hơn là các lớp riêng biệt. Khi kiểm tra các constructor, có thể thấy các lớp tổng hợp thường khởi tạo hoặc chấp nhận đối tượng thành phần thông qua tham số, điều này khẳng định mối quan hệ tổng hợp.

10. Unique item in a list

Question 2: When overriding the equals() method of the Object class, you will have to cast the Object parameter obj to the type of Object that you are dealing with. For example, in the Media class, you must cast the Object obj to a Media, and then check the equality of the two objects' attributes as the above requirements (i.e. title for Media; title and length for Track). If the passing object is not an instance of Media, what happens?

Answer: Nếu đối tượng truyền vào không phải là một instance của Media (hoặc Track), phương thức equals() sẽ kiểm tra bằng lệnh instanceof. Nếu điều kiện không thỏa mãn, phương thức sẽ trả về false, tức đối tượng không bằng nhau. Điều này giúp ngăn chặn lỗi ép kiểu sai và đảm bảo chỉ so sánh các đối tượng phù hợp.

11. Polymorphism with toString() method

Explain:

Khi gọi phương thức toString() trên từng phần tử của danh sách mediaList, Java sẽ sử dụng đa hình để gọi phương thức toString() tương ứng với loại đối tượng cụ thể (Book,

DigitalVideoDisc, CompactDisc). Mặc dù danh sách được khai báo là kiểu Media, nhưng các đối tượng thực sự bên trong danh sách quyết định hành vi của phương thức. Điều này minh họa đa hình ở mức hành vi, giúp mã linh hoạt và dễ mở rộng mà không cần sửa đổi logic in ra.

12.

Class Implementing Comparable Interface

Lớp Media nên triển khai Comparable<Media> nếu muốn sử dụng phương thức compareTo() để sắp xếp theo một tiêu chí mặc định.

Implement compareTo

Triển khai compareTo để so sánh Media dựa trên một tiêu chí, ví dụ:

@Override

```
public int compareTo(Media o) {  
    return this.title.compareTo(o.title); // So sánh theo tiêu đề  
}
```

Two Ordering Rules

Không thể hỗ trợ đồng thời hai quy tắc sắp xếp trong một lớp triển khai Comparable. Để làm điều đó, cần sử dụng Comparator.

Different Ordering for DVDs

Thêm Comparator dành riêng cho DigitalVideoDisc:

```
public static final Comparator<DigitalVideoDisc> COMPARE_BY_TITLE_LENGTH_COST =  
    Comparator  
        .comparing(DigitalVideoDisc::getTitle)  
        .thenComparing(Comparator.comparing(DigitalVideoDisc::getLength).reversed())
```

.thenComparing(DigitalVideoDisc::getCost);

Sử dụng comparator này khi cần.

Source code Cart.java

```
1  package aims.cart;
2
3  import aims.media.DigitalVideoDisc;
4  import aims.media.Media;
5
6  import java.util.ArrayList;
7
8  public class Cart { 6 usages  ▲ Hyuht17
9      public static final int MAX_NUMBERS_ORDERED = 20; 2 usages
10     private ArrayList<Media> itemsOrdered = new ArrayList<Media>(); 23 usages
11
12     public void addMedia(Media media) { 5 usages  ▲ Hyuht17
13         if ( itemsOrdered.size() < MAX_NUMBERS_ORDERED){
14             itemsOrdered.add(media);
15             System.out.println("The disc has been added");
16         }
17         else{
18             System.out.println("The cart is almost full");
19         }
20     }
21
22     public void removeMedia(Media media) { 1 usage  ▲ Hyuht17
23         int findItem = -1;
24         if (itemsOrdered.size() > 0){
25             if (itemsOrdered.contains(media)){
26                 findItem = itemsOrdered.indexOf(media);
27             }
28             if (findItem != -1){
29                 itemsOrdered.remove(findItem);
30                 System.out.println("The disc has been removed");
31             }
32             else {
33                 System.out.println("The disc is not in the cart");
34             }
35         }
36     }
37
38     public float totalCost(){ no usages  ▲ Hyuht17
39         float total = 0;
40         for (int i = 0; i < itemsOrdered.size(); i++){
41             total += itemsOrdered.get(i).getCost();
42         }
43         return total;
44     }
45
46     // public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList) {
47     //     for (int i = 0; i < dvdList.length; i++){
48     //         if (qtyOrdered < MAX_NUMBERS_ORDERED){
49     //             itemsOrdered[qtyOrdered] = dvdList[i];
50     //             qtyOrdered++;
51     //         }
52     //         else{
53     //             System.out.println("The cart is almost full");
54     //         }
55     //     }
56     // }
57
58  @ public void addMedia(Media...mediaList){ no usages  ▲ Hyuht17
59      for (Media media : mediaList){
60          if (itemsOrdered.size() < MAX_NUMBERS_ORDERED){
61              itemsOrdered.add(media);
62          }
63      }
64  }
```

```

81  ~ public void printCart(){ 2 usages ± Hyuht17
82      System.out.println("*****CART*****");
83      System.out.println("Ordered Items:");
84      float totalCost = 0;
85  ~   for (int i = 0; i < itemsOrdered.size(); i++){
86       System.out.println((i + 1) + "." + itemsOrdered.get(i).toString());
87       totalCost += itemsOrdered.get(i).getCost();
88   }
89   System.out.println("Total cost: " + totalCost);
90   System.out.println("*****");
91 }
92
93
94  ~ public void searchById(int id) { 3 usages ± Hyuht17
95      boolean found = false;
96  ~   for (int i = 0; i < itemsOrdered.size(); i++) {
97  ~       if (itemsOrdered.get(i).getId() == id) {
98           System.out.println("Found: " + itemsOrdered.get(i).toString());
99           found = true;
100      }
101  }
102  ~   if (!found) {
103       System.out.println("No match found for ID: " + id);
104   }
105 }
106
107  ~ public void searchByTitle(String title) { 3 usages ± Hyuht17
108      boolean found = false;
109  ~   for (int i = 0; i < itemsOrdered.size(); i++) {
110  ~       if (itemsOrdered.get(i).isMatch(title)) {

```

```

124  ~ public void sortCartByTitle() { 1 usage ± Hyuht17
125      itemsOrdered.sort((media1, media2) -> {
126          media1.getTitle().compareTo(media2.getTitle());
127          return 0;
128      });
129      System.out.println("The cart has been sorted by Title" + ".");
130  }
131
132  ~ public void sortCartByCost() { 1 usage ± Hyuht17
133      itemsOrdered.sort((media1, media2) -> {
134          return Float.compare(media1.getCost(), media2.getCost());
135      });
136      System.out.println("The cart has been sorted by Title" + ".");
137  }
138
139  ~ public void clearCart() { 1 usage ± Hyuht17
140      itemsOrdered.clear();
141      System.out.println("The cart has been cleared.");
142  }
143 }

```

Store.java

```
1 package aims.store;
2
3 import aims.media.DigitalVideoDisc;
4 import aims.media.Media;
5
6 import java.util.ArrayList;
7
8 public class Store { 6 usages 1 Hyuht17
9     public static final int MAX_ITEMS_IN_STORE = 100; 1 usage
10     private ArrayList<Media> itemsInStore = new ArrayList<Media>(); 10 usages
11
12     public void addMedia (Media media){ 3 usages 1 Hyuht17
13         if (itemsInStore.size() < MAX_ITEMS_IN_STORE){
14             itemsInStore.add(media);
15             System.out.println("The media has been added");
16         }
17         else{
18             System.out.println("The store is full");
19         }
20     }
21
22 @ public void removeMedia(Media media) { 2 usages 1 Hyuht17
23     int findIndex = -1;
24     String title = media.getTitle();
25     findIndex = itemsInStore.indexOf(itemsInStore.equals(title));
26
27     if (findIndex != -1) {
28         itemsInStore.remove(findIndex);
29         System.out.println("The Media has been removed from the store.");
30     } else {
31         System.out.println("No Media with the title \"\" + title + \"\" was found in the store.");
32     }
33 }
```

```
8 public class Store { 6 usages 1 Hyuht17
22 public void removeMedia(Media media) { 2 usages 1 Hyuht17
33 }
34
35 public void listDVDs(){ 2 usages 1 Hyuht17
36     for (Media media : itemsInStore) {
37         System.out.println(media.toString());
38     }
39 }
40
41 public void printStore(){ 1 usage 1 Hyuht17
42     System.out.println("*****CART*****");
43     System.out.println("Ordered Items:");
44     float totalCost = 0;
45     for (int i = 0; i < itemsInStore.size(); i++){
46         System.out.println((i + 1) + ". " + itemsInStore.get(i).toString());
47         totalCost += itemsInStore.get(i).getCost();
48     }
49     System.out.println("Total cost: " + totalCost);
50     System.out.println("*****");
51 }
52
53 public Media searchByTitle(String title){ 4 usages 1 Hyuht17
54     Media findMedia = null;
55     for (Media media : itemsInStore){
56         if (media.getTitle().toLowerCase().contains(title.toLowerCase()))
57         {
58             findMedia = media;
59             break;
60         }
61     }
62     return findMedia;
63 }
```

Aim.java

```
1  package aims;
2
3  import aims.cart.Cart;
4  import aims.media.Book;
5  import aims.media.CompactDisc;
6  import aims.media.DigitalVideoDisc;
7  import aims.media.Media;
8  import aims.store.Store;
9
10 import java.util.Scanner;
11
12 public class Aims {
13     private static Store store = new Store();
14     private static Cart cart = new Cart();
15
16     public static void main(String[] args) {
17         Scanner scanner = new Scanner(System.in);
18         int choice;
19
20         do {
21             showMenu();
22             choice = scanner.nextInt();
23             scanner.nextLine(); // Đọc bỏ dòng trống
24
25             switch (choice) {
26                 case 1:
27                     viewStore(scanner);
28                     break;
29                 case 2:
30                     updateStore(scanner);
31                     break;
32                 case 3:
33                     viewCart(scanner);
34                     break;
35                 case 0:
36                     System.out.println("Exiting the application. Goodbye!");
37                     break;
38                 default:
39                     System.out.println("Invalid choice! Please choose again.");
40             }
41         } while (choice != 0);
42
43         scanner.close();
44     }
45
46     public static void showMenu() {
47         System.out.println("AIMS: ");
48         System.out.println("-----");
49         System.out.println("1. View store");
50         System.out.println("2. Update store");
51         System.out.println("3. See current cart");
52         System.out.println("0. Exit");
53         System.out.println("-----");
54         System.out.print("Please choose a number: 0-1-2-3: ");
55     }
56
57     public static void viewStore(Scanner scanner) {
58         System.out.println("Store contents:");
59         store.printStore();
60
61         int choice;
62         do {
```

```

63     storeMenu();
64     choice = scanner.nextInt();
65     scanner.nextLine(); // Đọc bỏ dòng trống
66
67     switch (choice) {
68         case 1:
69             seeMediaDetails(scanner);
70             break;
71         case 2:
72             addMediaToCart(scanner);
73             break;
74         case 3:
75             playMedia(scanner);
76             break;
77         case 4:
78             viewCart(scanner);
79             break;
80         case 0:
81             System.out.println("Returning to main menu.");
82             break;
83         default:
84             System.out.println("Invalid choice! Please choose again.");
85     }
86 } while (choice != 0);
87 }
88
89 public static void storeMenu() { 1 usage  ± Hyuht17
90     System.out.println("Options: ");
91     System.out.println("-----");
92     System.out.println("1. See a media's details");

```

125/26 2 8 CRLF UTF-8 4 spaces

```

93     System.out.println("2. Add a media to cart");
94     System.out.println("3. Play a media");
95     System.out.println("4. See current cart");
96     System.out.println("0. Back");
97     System.out.println("-----");
98     System.out.print("Please choose a number: 0-1-2-3-4: ");
99 }
100
101 @ public static void seeMediaDetails(Scanner scanner) { 1 usage  ± Hyuht17
102     System.out.print("Enter the title of the media: ");
103     String title = scanner.nextLine();
104     Media media = store.searchByTitle(title);
105
106     if (media != null) {
107         System.out.println(media.toString());
108
109         if (media instanceof DigitalVideoDisc || media instanceof CompactDisc) {
110             mediaDetailsMenu();
111             int choice = scanner.nextInt();
112             scanner.nextLine(); // Đọc bỏ dòng trống
113
114             switch (choice) {
115                 case 1:
116                     cart.addMedia(media);
117                     break;
118                 case 2:
119                     if (media instanceof DigitalVideoDisc) {
120                         DigitalVideoDisc dvd = (DigitalVideoDisc) media; // Ép kiểu
121                         dvd.play();
122                     } else if (media instanceof CompactDisc) {

```

```

120         DigitalVideoDisc dvd = (DigitalVideoDisc) media; // Ép kiểu
121         dvd.play();
122     } else if (media instanceof CompactDisc) {
123         CompactDisc cd = (CompactDisc) media; // Ép kiểu
124         cd.play();
125     }
126     break;
127     case 0:
128         System.out.println("Returning to store menu.");
129         break;
130     default:
131         System.out.println("Invalid choice! Please choose again.");
132     }
133 }
134 } else {
135     System.out.println("Media not found in the store.");
136 }
137 }
138
139 public static void mediaDetailsMenu() { 1 usage  ▲ Hyuht17
140     System.out.println("Options: ");
141     System.out.println("-----");
142     System.out.println("1. Add to cart");
143     System.out.println("2. Play");
144     System.out.println("0. Back");
145     System.out.println("-----");
146     System.out.print("Please choose a number: 0-1-2: ");
147 }
148
149 @ public static void addMediaToCart(Scanner scanner) { 1 usage  ▲ Hyuht17

```

```

149 @ public static void addMediaToCart(Scanner scanner) { 1 usage  ▲ Hyuht17
150     System.out.print("Enter the title of the media: ");
151     String title = scanner.nextLine();
152     Media media = store.searchByTitle(title);
153
154     if (media != null) {
155         cart.addMedia(media);
156         if (media instanceof DigitalVideoDisc) {
157             System.out.println("Number of DVDs in cart: " + countDVDsInCart());
158         }
159     } else {
160         System.out.println("Media not found in the store.");
161     }
162 }
163
164 @ public static void playMedia(Scanner scanner) { 2 usages  ▲ Hyuht17
165     System.out.print("Enter the title of the media: ");
166     String title = scanner.nextLine();
167     Media media = store.searchByTitle(title);
168
169     if (media instanceof DigitalVideoDisc) {
170         DigitalVideoDisc dvd = (DigitalVideoDisc) media; // Ép kiểu
171         dvd.play();
172     } else if (media instanceof CompactDisc) {
173         CompactDisc cd = (CompactDisc) media; // Ép kiểu
174         cd.play();
175     } else {
176         System.out.println("This media cannot be played.");
177     }
178 }
179

```



```

181     public static int countDVDsInCart() { 1 usage  ± Hyuht17
182         return (int) cart.getItemsOrdered().stream()
183             .filter(media -> media instanceof DigitalVideoDisc)
184             .count();
185     }
186
187 ② public static void updateStore(Scanner scanner) { 1 usage  ± Hyuht17
188     System.out.println("Update store:");
189     System.out.println("1. Add a media");
190     System.out.println("2. Remove a media");
191     System.out.print("Choose an option: ");
192     int choice = scanner.nextInt();
193     scanner.nextLine(); // Đọc bỏ dòng trống
194
195     switch (choice) {
196     case 1:
197         System.out.print("Enter media details (id, title, category, cost): ");
198         int id = scanner.nextInt();
199         scanner.nextLine();
200         String title = scanner.nextLine();
201         String category = scanner.nextLine();
202         float cost = scanner.nextFloat();
203         scanner.nextLine();
204
205         Media newMedia = new Book(id, title, category, cost);
206         store.addMedia(newMedia);
207         System.out.println("Media added to store.");
208         break;
209     case 2:
210         System.out.print("Enter the title of the media to remove: ");
211         String removeTitle = scanner.nextLine();

```

```

212         Media mediaToRemove = store.searchByTitle(removeTitle);
213
214         if (mediaToRemove != null) {
215             store.removeMedia(mediaToRemove);
216             System.out.println("Media removed from store.");
217         } else {
218             System.out.println("Media not found in store.");
219         }
220         break;
221     default:
222         System.out.println("Invalid choice! Returning to main menu.");
223     }
224 }
225
226 ② public static void viewCart(Scanner scanner) { 2 usages  ± Hyuht17
227     cart.printCart();
228
229     int choice;
230     do {
231         cartMenu();
232         choice = scanner.nextInt();
233         scanner.nextLine(); // Đọc bỏ dòng trống
234
235         switch (choice) {
236         case 1:
237             filterCart(scanner);
238             break;
239         case 2:
240             sortCart(scanner);
241             break;

```

```

239         case 2:
240             sortCart(scanner);
241             break;
242         case 3:
243             removeMediaFromCart(scanner);
244             break;
245         case 4:
246             playMedia(scanner);
247             break;
248         case 5:
249             placeOrder();
250             break;
251         case 0:
252             System.out.println("Returning to main menu.");
253             break;
254         default:
255             System.out.println("Invalid choice! Please choose again.");
256     }
257 } while (choice != 0);
258 }
259
260 public static void cartMenu() { 1 usage  ± Hyuht17
261     System.out.println("Options: ");
262     System.out.println("-----");
263     System.out.println("1. Filter medias in cart");
264     System.out.println("2. Sort medias in cart");
265     System.out.println("3. Remove media from cart");
266     System.out.println("4. Play a media");
267     System.out.println("5. Place order");
268     System.out.println("0. Back");

```

```

266     System.out.println("4. Play a media");
267     System.out.println("5. Place order");
268     System.out.println("0. Back");
269     System.out.println("-----");
270     System.out.print("Please choose a number: 0-1-2-3-4-5: ");
271 }
272
273 @  ✓ public static void filterCart(Scanner scanner) { 1 usage  ± Hyuht17
274     System.out.println("Filter options:");
275     System.out.println("1. By ID");
276     System.out.println("2. By Title");
277     System.out.print("Choose an option: ");
278     int choice = scanner.nextInt();
279     scanner.nextLine(); // Đọc bỏ dòng trống
280
281     ✓ if (choice == 1) {
282         System.out.print("Enter ID: ");
283         int id = scanner.nextInt();
284         cart.searchById(id);
285     } else if (choice == 2) {
286         System.out.print("Enter title: ");
287         String title = scanner.nextLine();
288         cart.searchByTitle(title);
289     } else {
290         System.out.println("Invalid choice!");
291     }
292 }
293
294 @  ✓ public static void sortCart(Scanner scanner) { 1 usage  ± Hyuht17
295     System.out.println("Sort options:");

```

```

287         String title = scanner.nextLine();
288         cart.searchByTitle(title);
289     } else {
290         System.out.println("Invalid choice!");
291     }
292 }
293
294 @ public static void sortCart(Scanner scanner) { 1 usage 1 Hyuht17
295     System.out.println("Sort options:");
296     System.out.println("1. By Title");
297     System.out.println("2. By Cost");
298     System.out.print("Choose an option: ");
299     int choice = scanner.nextInt();
300     scanner.nextLine(); // Đọc bỏ dòng trống
301
302     if (choice == 1) {
303         cart.sortCartByTitle();
304     } else if (choice == 2) {
305         cart.sortCartByCost();
306     } else {
307         System.out.println("Invalid choice!");
308     }
309 }
310
311 @ public static void removeMediaFromCart(Scanner scanner) { 1 usage 1 Hyuht17
312     System.out.print("Enter the title of the media to remove: ");
313     String title = scanner.nextLine();
314     Media media = cart.getItemsOrdered().stream() Stream<Media>
315         .filter(m -> m.getTitle().equalsIgnoreCase(title))
316         .findFirst() Optional<Media>

```

```

307         System.out.println("Invalid choice!");
308     }
309 }
310
311 @ public static void removeMediaFromCart(Scanner scanner) { 1 usage 1 Hyuht17
312     System.out.print("Enter the title of the media to remove: ");
313     String title = scanner.nextLine();
314     Media media = cart.getItemsOrdered().stream() Stream<Media>
315         .filter(m -> m.getTitle().equalsIgnoreCase(title))
316         .findFirst() Optional<Media>
317         .orElse(null);
318
319     if (media != null) {
320         cart.removeMedia(media);
321     } else {
322         System.out.println("Media not found in cart.");
323     }
324 }
325
326 public static void placeOrder() { 1 usage 1 Hyuht17
327     System.out.println("Order placed successfully!");
328     cart.clearCart();
329 }
330 }
331

```