

# Report OOP – Lab03

## Basic Object-Oriented Techniques

Kỳ 2024-1

- **Họ và tên:** Lê Hải Yến
- **MSSV:** 20225780
- **Mã lớp:** 744520

## Contents

<b>1</b>	<b>New Written Code</b>	<b>2</b>
1.1	Working with Method Overloading . . . . .	2
1.1.1	Overloading by differing types of parameter . . . . .	2
1.1.2	Overloading by differing the number of parameters . . . . .	2
1.2	Passing Parameter . . . . .	2
1.3	Classifier Member and Instance Member . . . . .	3
1.4	Print and Search Cart . . . . .	4
1.4.1	Print Cart . . . . .	4
1.4.2	Search Cart . . . . .	5
1.5	Implement the Store Class . . . . .	6
1.6	String, StringBuilder, and StringBuffer . . . . .	7
<b>2</b>	<b>Debug</b>	<b>9</b>
<b>3</b>	<b>UML Diagram</b>	<b>10</b>
3.1	Use-Case Diagram . . . . .	10
3.2	Class Diagram . . . . .	10
<b>4</b>	<b>Answer Questions</b>	<b>11</b>

# 1 New Written Code

## 1.1 Working with Method Overloading

Thêm nạp chồng phương thức addDigitalVideoDiscLHY tại class Cart.java.

### 1.1.1 Overloading by differing types of parameter

```
20 public void addDigitalVideoDiscLHY(DigitalVideoDiscLHY [] dvdList) {
21     if (dvdList.length > MAX_NUMBERS_ORDERED) {
22         System.out.println("Your cart is full, cannot add a new item!");
23     } else {
24         for (int i = 0; i < dvdList.length; i++) {
25             itemsOrdered[qtyOrdered] = dvdList[i];
26             System.out.println("DigitalVideoDisc: " + dvdList[i].getTitleLHY() + " has been added!");
27             qtyOrdered ++ ;
28         }
29     }
30 }
```

### 1.1.2 Overloading by differing the number of parameters

```
32 public void addDigitalVideoDiscLHY(DigitalVideoDiscLHY dvd1,DigitalVideoDiscLHY dvd2) {
33     DigitalVideoDiscLHY [] dvdList = {dvd1, dvd2};
34     addDigitalVideoDiscLHY(dvdList);
35 }
--
```

## 1.2 Passing Parameter

- Thêm setter cho title của lớp DigitalVideoDiscLHY.

```
32 public void setterTitleLHY(String title) {
33     this.title = title;
34 }
```

- Tạo lớp DigitalVideoDiscWrapper để bao bọc lớp DigitalVideoDiscLHY, cho phép thay đổi giá trị thông qua phương thức swap chính xác trong lớp TestPassingParameter.

```
1 package hust.soict.ITE6.test.disc;
2 import hust.soict.ITE6.aims.disc.DigitalVideoDiscLHY;
3
4 public class DigitalVideoDiscWrapper {
5     DigitalVideoDiscLHY dvd;
6     DigitalVideoDiscWrapper(DigitalVideoDiscLHY dvd) {
7         super();
8         this.dvd = dvd;
9     }
10 }
```

```

1 package hust.soict ITE6.test.disc;
2 import hust.soict.ITE6.aims.disc.DigitalVideoDiscLHY;
3
4 public class TestPassingParameter {
5     public static void main(String[] args) {
6         DigitalVideoDiscLHY jungleDVD = new DigitalVideoDiscLHY("Jungle");
7         DigitalVideoDiscLHY cinderellaDVD = new DigitalVideoDiscLHY("Cinderella");
8
9         // Wrap the DVD objects into wrapper classes
10        DigitalVideoDiscWrapper jungleWrapper = new DigitalVideoDiscWrapper(jungleDVD);
11        DigitalVideoDiscWrapper cinderellaWrapper = new DigitalVideoDiscWrapper(cinderellaDVD);
12
13        // Call the wrong swap() method
14        swap(jungleDVD, cinderellaDVD);
15        System.out.println("Wrong swapped jungle dvd title: " + jungleDVD.getTitleLHY());
16        System.out.println("Wrong swapped cinderella dvd title: " + cinderellaDVD.getTitleLHY());
17
18        // Call the correct swap() method
19        swap(jungleWrapper, cinderellaWrapper);
20        System.out.println("Correct swapped jungle dvd title: " + jungleWrapper.dvd.getTitleLHY());
21        System.out.println("Correct swapped cinderella dvd title: " + cinderellaWrapper.dvd.getTitleLHY());
22
23        changeTitleLHY(jungleDVD, cinderellaDVD.getTitleLHY());
24        System.out.println("Change jungle dvd title: " + jungleDVD.getTitleLHY());
25    }
26
27    // Wrong
28    public static void swap(Object o1, Object o2)
29    {
30        Object tmp = o1;
31        o1 = o2;
32        o2 = tmp;
33    }
34
35    // Correct
36    public static void swap(DigitalVideoDiscWrapper o1, DigitalVideoDiscWrapper o2)
37    {
38        DigitalVideoDiscLHY tmp = o1.dvd;
39        o1.dvd = o2.dvd;
40        o2.dvd = tmp;
41    }
42
43    public static void changeTitleLHY(DigitalVideoDiscLHY dvd, String title)
44    {
45        String oldTitle = dvd.getTitleLHY();
46        dvd.setterTitleLHY(title);
47        dvd = new DigitalVideoDiscLHY(oldTitle);
48    }
49 }

```

### 1.3 Classifier Member and Instance Member

Tại class DigitalVideoDisc, thêm các instance như sau:

```

9     private static int nbDigitalVideoDiscs = 0;
10    private int id;

36    public int getId() {
37        return id;
38    }

```

```

44- public DigitalVideoDiscLHY(String title) {
45     super();
46     this.title = title;
47     this.id = ++nbDigitalVideoDiscs;
48 }
49
50- public DigitalVideoDiscLHY(String title, String category, float cost) {
51     super();
52     this.title = title;
53     this.category = category;
54     this.cost = cost;
55     this.id = ++nbDigitalVideoDiscs;
56 }
57
58- public DigitalVideoDiscLHY(String director, String category, String title, float cost) {
59     super();
60     this.title = title;
61     this.category = category;
62     this.director = director;
63     this.cost = cost;
64     this.id = ++nbDigitalVideoDiscs;
65 }

67- public DigitalVideoDiscLHY(String title, String category, String director, int length, float cost) {
68     super();
69     this.title = title;
70     this.category = category;
71     this.director = director;
72     this.length = length;
73     this.cost = cost;
74     this.id = ++nbDigitalVideoDiscs;
75 }
--

```

## 1.4 Print and Search Cart

### 1.4.1 Print Cart

- Hàm toString() thay đổi kiểu trả về của String trong class DigitalVideoDisc.

```

77- public String toString() {
78     return this.id + ". DVD: " + this.title +
79         " - Category: " + this.category +
80         " - Director: " + this.title +
81         " - DVD length: " + this.length +
82         " - Cost: " + this.cost + "$";
83 }

```

- Phương thức print() để in danh sách các mặt hàng trong class Cart theo mẫu.

```

65- public void print() {
66     System.out.println("*****CART*****");
67     System.out.println("Ordered Items:");
68     for (int i = 0; i < qtyOrdered; i++) {
69         System.out.println(itemsOrdered[i]);
70     }
71     System.out.println("Total cost: " + totalCost());
72     System.out.println("*****");
73 }
--

```

### 1.4.2 Search Cart

- Xây dựng phương thức tìm kiếm theo ID và Title trong class Cart:

```
75 public void searchByID(int id) {
76     boolean isFound = false;
77     for (int i = 0; i < qtyOrdered; i++) {
78         if (itemsOrdered[i].getId() == id) {
79             System.out.println("Found: " + itemsOrdered[i]);
80             isFound = true;
81         }
82     }
83     if (isFound == false) {
84         System.out.println("No DVDs were found with the given ID.");
85     }
86 }
87
88 public void searchByTitle(String keyword) {
89     boolean isMatchFound = false;
90     for (int i=0; i < qtyOrdered; i++) {
91         if (itemsOrdered[i].isMatch(keyword)) {
92             System.out.println("Found: " + itemsOrdered[i]);
93             isMatchFound = true;
94         }
95     }
96     if (isMatchFound == false) {
97         System.out.println("No DVDs found with the title containing \"" + keyword + "\".");
98     }
99 }
100
101 }
```

- Sử dụng phương thức boolean isMatch(String title) trong DigitalVideoDisc để kiểm tra xem đĩa có khớp với tiêu đề đã cho hay không.

```
40 public boolean isMatch(String keyword) {
41     return this.title.toLowerCase().contains(keyword.toLowerCase());
42 }
```

- Tạo class CartTest để kiểm tra các hàm trên.

```
1 package hust.soict.ITE6.test.cart;
2 import hust.soict.ITE6.aims.cart.CartLHY;
3
4
5 public class CartTest {
6     public static void main(String[] args) {
7
8         CartLHY cart = new CartLHY();
9
10        DigitalVideoDiscLHY dvd1 = new DigitalVideoDiscLHY("The Lion King",
11            "Animation", "Roger Allers", 87, 19.95f);
12        cart.addDigitalVideoDiscLHY(dvd1);
13        DigitalVideoDiscLHY dvd2 = new DigitalVideoDiscLHY("Star War",
14            "Science Fiction", "George Lucas", 87, 24.95f);
15        cart.addDigitalVideoDiscLHY(dvd2);
16        DigitalVideoDiscLHY dvd3 = new DigitalVideoDiscLHY("Aladin",
17            "Animation", 18.99f);
18        cart.addDigitalVideoDiscLHY(dvd3);
19        cart.print();
20        cart.searchByID(1);
21        cart.searchByTitle("Star");
22        cart.searchByTitle("Baby");
23    }
24 }
```

## 1.5 Implement the Store Class

- Tạo class `Store`, chứa thuộc tính `itemsInStore[]` là một mảng các DVD có sẵn trong cửa hàng.
- Thêm hai phương thức `addDVD` và `removeDVD`.

```
1 package hust.soict.ITE6.aims.store;
2 import java.util.ArrayList;
3
4
5
6
7 public class Store {
8     private List<DigitalVideoDiscLHY> itemsInStore = new ArrayList<DigitalVideoDiscLHY>();
9
10    public void addDVD(DigitalVideoDiscLHY dvd)
11    {
12        int index = itemsInStore.indexOf(dvd);
13        if (index != -1) {
14            System.out.println(dvd.getTitleLHY() + " is already available in the store.");
15        } else {
16            itemsInStore.add(dvd);
17            System.out.println(dvd.getTitleLHY() + " has been successfully added to the store.");
18        }
19    }
20
21    public void removeDVD(DigitalVideoDiscLHY dvd)
22    {
23        boolean removed = itemsInStore.remove(dvd);
24        if (removed)
25        {
26            System.out.println(dvd.getTitleLHY() + " has been successfully removed from the store.");
27        } else {
28            System.out.println(dvd.getTitleLHY() + " was not found in the store.");
29        }
30    }
31
32    public void displayDVD() {
33        if (itemsInStore.isEmpty()) {
34            System.out.println("No DVDs available in the store.");
35        } else {
36            for (int i = 0; i < itemsInStore.size(); i++) {
37                System.out.println((i + 1) + ". " + itemsInStore.get(i));
38            }
39        }
40    }
41 }
```

- Tạo class `StoreTest` để kiểm tra hai phương thức trên.

```

1 package hust.soict.ITE6.test.store;
2 import hust.soict.ITE6.aims.disc.DigitalVideoDiscLHY;
3
4
5 public class StoreTest {
6     public static void main(String[] args) {
7         Store store = new Store();
8
9         DigitalVideoDiscLHY dvd1 = new DigitalVideoDiscLHY("The Lion King",
10             "Animation", "Roger Allers", 87, 19.95f);
11         store.addDVD(dvd1);
12
13         DigitalVideoDiscLHY dvd2 = new DigitalVideoDiscLHY("Star Wars",
14             "Science Fiction", "George Lucas", 87, 24.95f);
15         store.addDVD(dvd2);
16
17         DigitalVideoDiscLHY dvd3 = new DigitalVideoDiscLHY("Aladin",
18             "Animation", 18.99f);
19         store.addDVD(dvd3);
20
21         store.displayDVD();
22
23         store.addDVD(dvd2);
24         store.removeDVD(dvd2);
25
26         store.displayDVD();
27
28         store.addDVD(dvd2);
29
30         store.displayDVD();
31     }
32 }

```

## 1.6 String, StringBuilder, and StringBuffer

- Tạo một lớp mới tên là ConcatenationInLoops để kiểm tra thời gian xử lý khi xây dựng chuỗi String bằng toán tử +, StringBuilder, và StringBuffer.

```

1 package src.hust.soict.ITE6.garbage;
2 import java.util.Random;
3 public class ConcatenationInLoops {
4     public static void main(String[] args) {
5         Random randomGenerator = new Random(123);
6
7         long startTime = System.currentTimeMillis();
8         String concatenatedString = "";
9         for (int i = 0; i < 65536; i++) {
10             concatenatedString += randomGenerator.nextInt(2);
11         }
12         System.out.println("Time taken with + operator: " + (System.currentTimeMillis() - startTime) + "ms");
13         System.out.println("Length of concatenatedString using + operator: " + concatenatedString.length());
14
15         randomGenerator = new Random(123);
16         startTime = System.currentTimeMillis();
17         StringBuffer stringBuffer = new StringBuffer();
18         for (int i = 0; i < 65536; i++) {
19             stringBuffer.append(randomGenerator.nextInt(2));
20         }
21         concatenatedString = stringBuffer.toString();
22         System.out.println("Time taken with StringBuffer: " + (System.currentTimeMillis() - startTime) + "ms");
23
24         randomGenerator = new Random(123);
25         startTime = System.currentTimeMillis();
26         StringBuilder stringBuilder = new StringBuilder();
27         for (int i = 0; i < 65536; i++) {
28             stringBuilder.append(randomGenerator.nextInt(2));
29         }
30         concatenatedString = stringBuilder.toString();
31         System.out.println("Time taken with StringBuilder: " + (System.currentTimeMillis() - startTime) + "ms");
32     }
33 }

```

- Tạo một class mới là GarbageCreator để đọc tệp văn bản vào chuỗi dùng +.

```

1 package src.hust.soict.ITE6.garbage;
2
3 import java.io.IOException;
4
5
6
7 public class GarbageCreator {
8     public static void main(String[] args) {
9
10         String filename = "src/test.txt";
11         byte[] inputBytes = { 0 };
12         long startTime, endTime;
13
14         try {
15             inputBytes = Files.readAllBytes(Paths.get(filename));
16         } catch (IOException e) {
17             e.printStackTrace();
18         }
19
20         startTime = System.currentTimeMillis();
21
22         String outputString = "";
23         for (byte b : inputBytes) {
24             outputString += (char)b;
25         }
26
27         endTime = System.currentTimeMillis();
28
29         System.out.println("Time: " + (endTime - startTime) + " ms");
30
31         System.out.println("First 100 characters: " + outputString.substring(0, Math.min(outputString.length(), 100)));
32     }
33 }

```

- Tạo một class mới là NoGarbage để đọc tệp văn bản vào chuỗi dùng StringBuilder.

```

1 package src.hust.soict.ITE6.garbage;
2
3 import java.io.IOException;
4
5
6
7 public class NoGarbage {
8     public static void main(String[] args) {
9
10         String filename = "src/test.txt";
11         byte[] inputBytes = { 0 };
12         long startTime, endTime;
13
14         try {
15             inputBytes = Files.readAllBytes(Paths.get(filename));
16         } catch (IOException e) {
17             e.printStackTrace();
18         }
19
20         startTime = System.currentTimeMillis();
21
22         StringBuilder outputStringBuilder = new StringBuilder();
23         for (byte b : inputBytes) {
24             outputStringBuilder.append((char)b);
25         }
26
27         endTime = System.currentTimeMillis();
28
29         System.out.println("Time: " + (endTime - startTime) + " ms");
30
31         System.out.println("First 100 characters: " + outputStringBuilder.substring(0, Math.min(outputStringBuilder.length(), 100)));
32     }
33 }

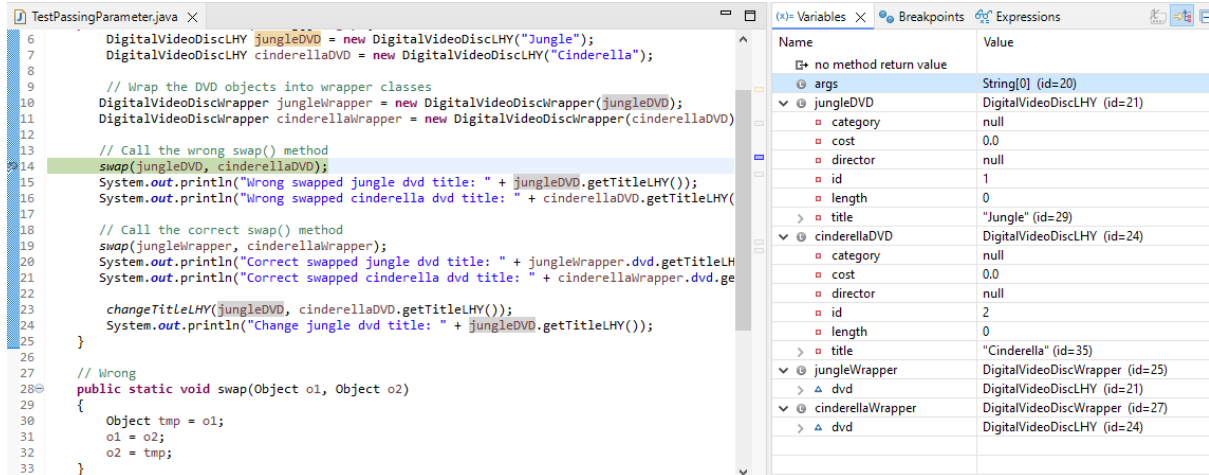
```



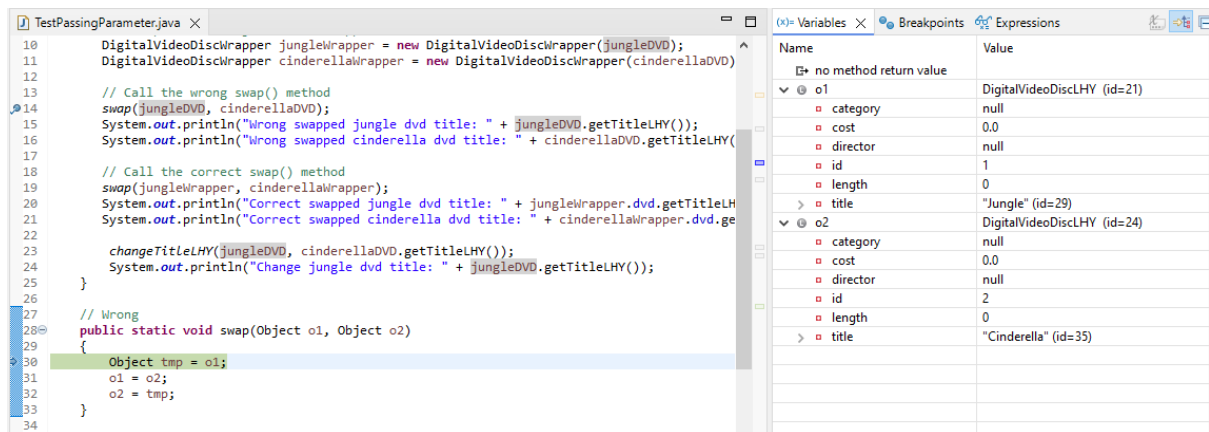
## 2 Debug

Tại class TestPassingParameter với hàm swap():

- Đặt breakpoint tại dòng swap(jungleDVD, cinderellaDVD);



- Sử dụng Step into (F5) để nhảy vào hàm.



- Sử dụng Step over (F6) để chạy dòng lệnh tiếp theo.

```

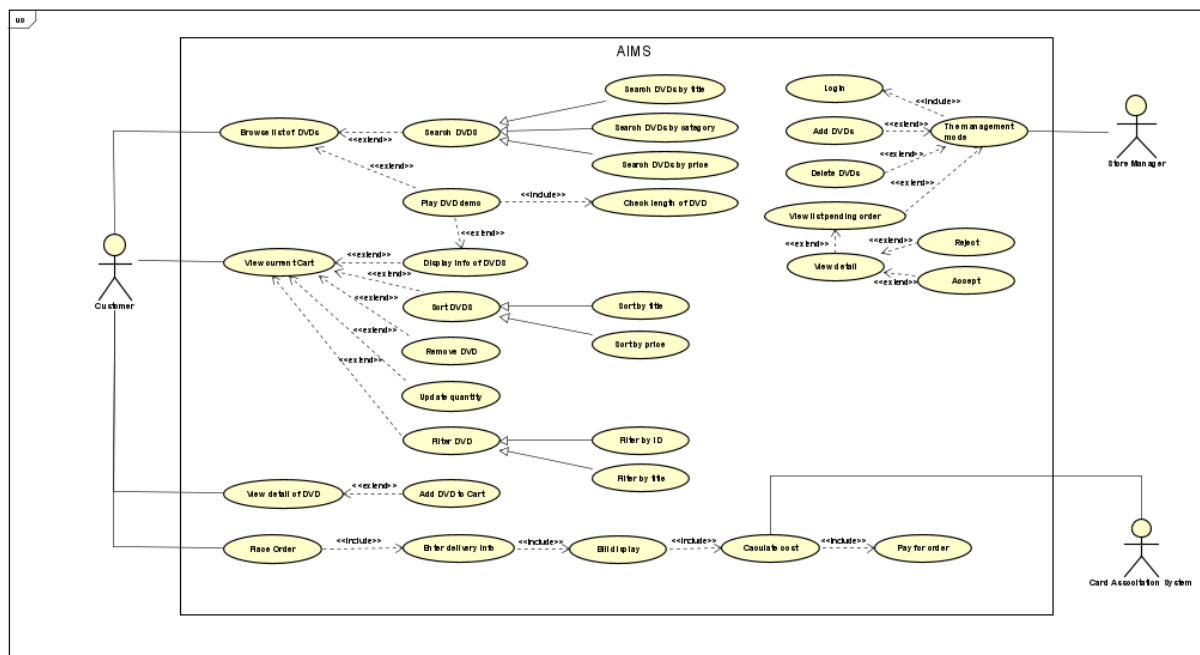
TestPassingParameter.java
10 DigitalVideoDiscWrapper jungleWrapper = new DigitalVideoDiscWrapper(jungleDVD);
11 DigitalVideoDiscWrapper cinderellaWrapper = new DigitalVideoDiscWrapper(cinderellaDVD);
12
13 // Call the wrong swap() method
14 swap(jungleDVD, cinderellaDVD);
15 System.out.println("Wrong swapped jungle dvd title: " + jungleDVD.getTitleLHY());
16 System.out.println("Wrong swapped cinderella dvd title: " + cinderellaDVD.getTitleLHY());
17
18 // Call the correct swap() method
19 swap(jungleWrapper, cinderellaWrapper);
20 System.out.println("Correct swapped jungle dvd title: " + jungleWrapper.dvd.getTitleLHY());
21 System.out.println("Correct swapped cinderella dvd title: " + cinderellaWrapper.dvd.getTitleLHY());
22
23 changeTitleLHY(jungleDVD, cinderellaDVD.getTitleLHY());
24 System.out.println("Change jungle dvd title: " + jungleDVD.getTitleLHY());
25
26
27 // Wrong
28 public static void swap(Object o1, Object o2)
29 {
30     Object tmp = o1;
31     o1 = o2;
32     o2 = tmp;
33 }
34
35 // Correct
36 public static void swap(DigitalVideoDiscWrapper o1, DigitalVideoDiscWrapper o2)
37 {
38     DigitalVideoDiscWrapper tmp = o1;
39     o1 = o2;
40     o2 = tmp;
41 }

```

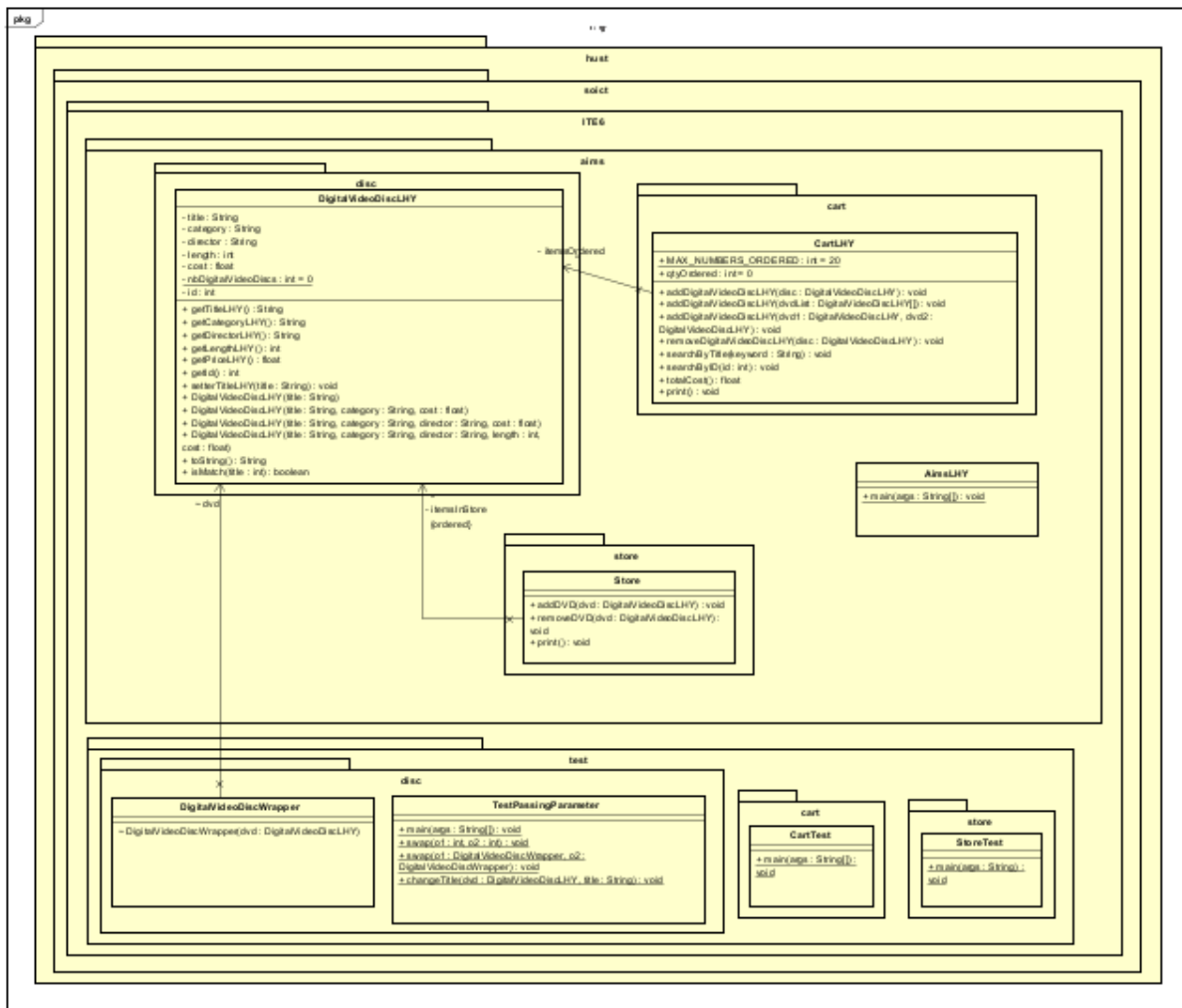
Name	Value
no method return value	
o1	DigitalVideoDiscLHY (id=21)
category	null
cost	0.0
director	null
id	1
length	0
title	"Jungle" (id=29)
o2	DigitalVideoDiscLHY (id=24)
category	null
cost	0.0
director	null
id	2
length	0
title	"Cinderella" (id=35)
tmp	DigitalVideoDiscLHY (id=21)

## 3 UML Diagram

### 3.1 Use-Case Diagram



### 3.2 Class Diagram



## 4 Answer Questions

### 1. Is JAVA a Pass by Value or a Pass by Reference programming language?

- Java là một pass-by-value programming language.

### 2. After the call of swap(jungleDVD, cinderellaDVD), why do the titles of these two objects remain unchanged?

- Bởi vì phương thức swap() chỉ hoán đổi giá trị của các trường title giữa hai đối tượng, nhưng không thay đổi tham chiếu của các đối tượng. Do đó, tham chiếu đến các đối tượng jungleDVD và cinderellaDVD vẫn trở về cùng một vùng nhớ như trước khi phương thức swap() được gọi. Do đó sự thay đổi chỉ phản ánh bên ngoài phương thức chứ không làm thay đổi giá trị thực của đối tượng.

### 3. After the call of changeTitle(jungleDVD, cinderellaDVD.getTitle()), why is the title of jungleDVD changed?

- Bởi vì phương thức changeTitle() trực tiếp thay đổi trường title của đối tượng jungleDVD thông qua phương thức setter.

4. **Write a `toString()` method for the `DigitalVideoDisc` class. What should be the return type of this method?**

- Viết phương thức `toString()` cho lớp `DigitalVideoDisc`. Kiểu trả về của phương thức này nên là `String`.