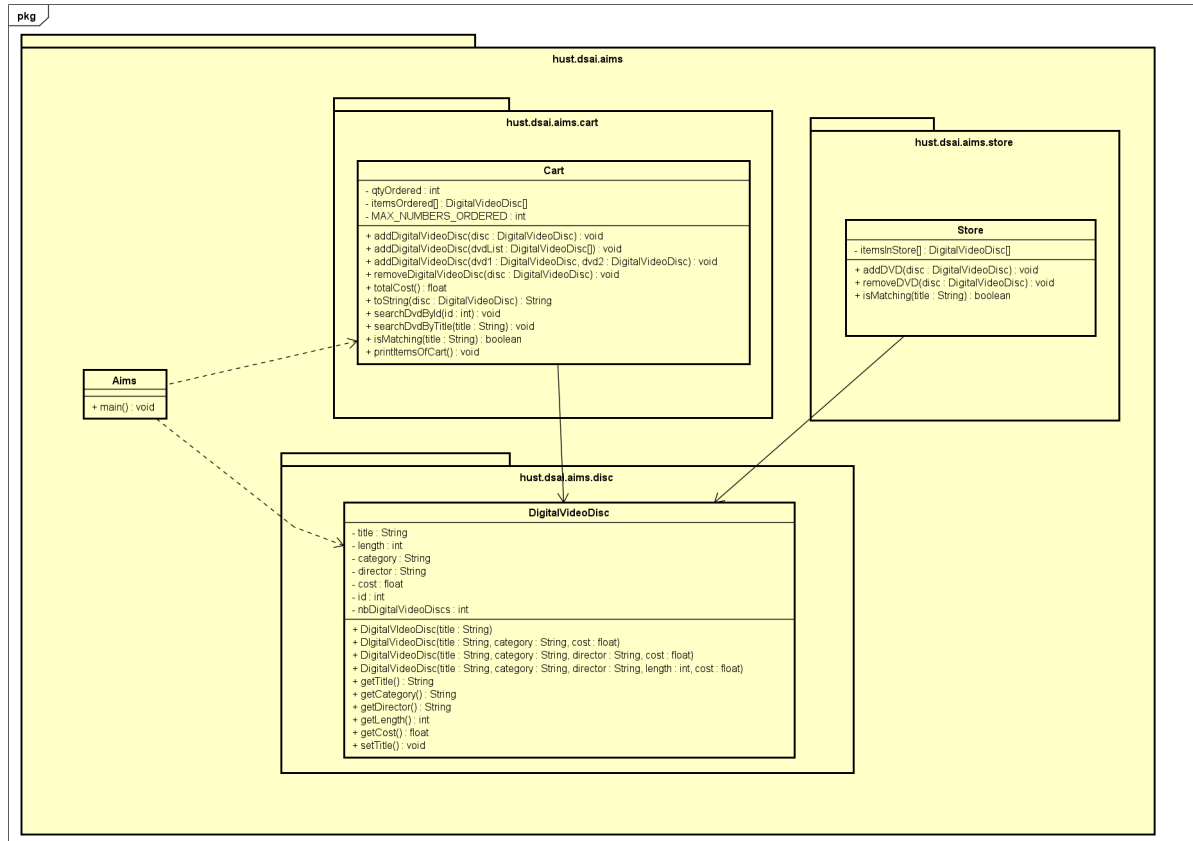


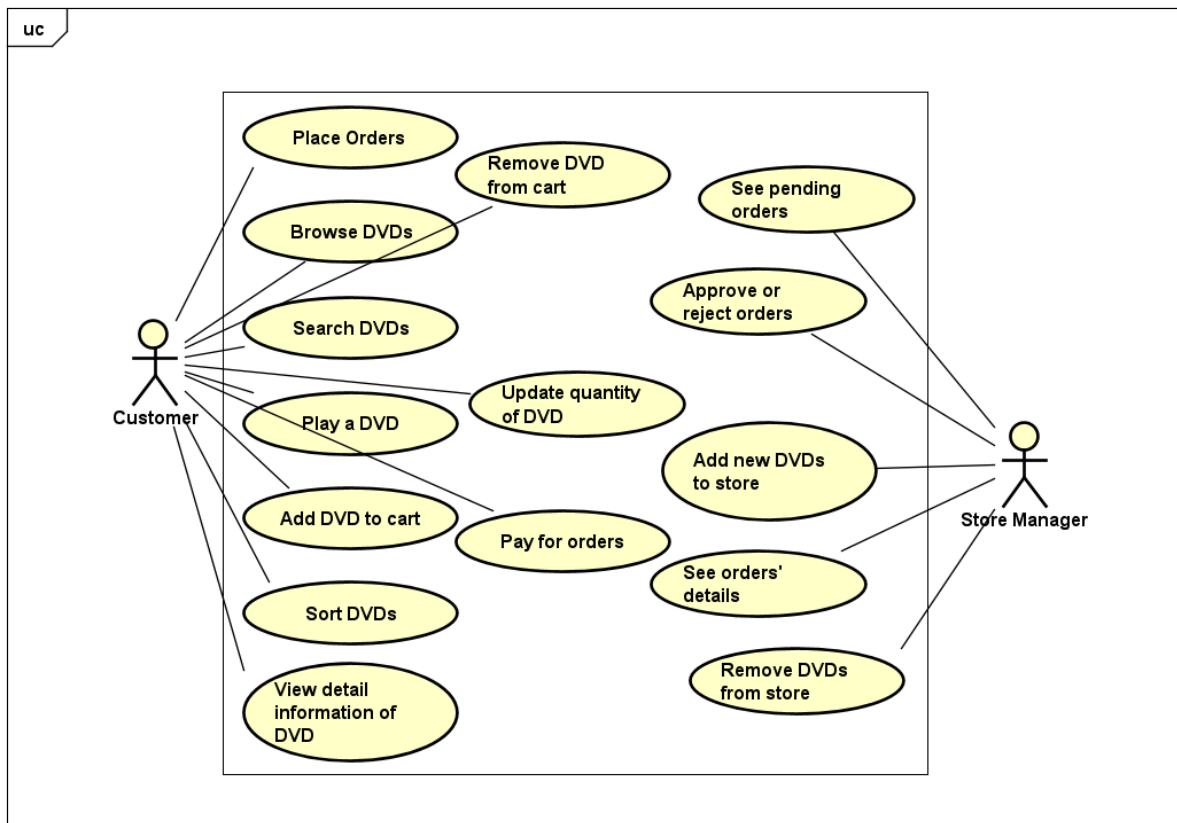
# Lab03 Report

## 1. Updated Diagram.

### 1.1. Class Diagram.



## 1.2. Use case Diagram.



## 2. Source Code.

### 2.1. Overloading Method in Cart.

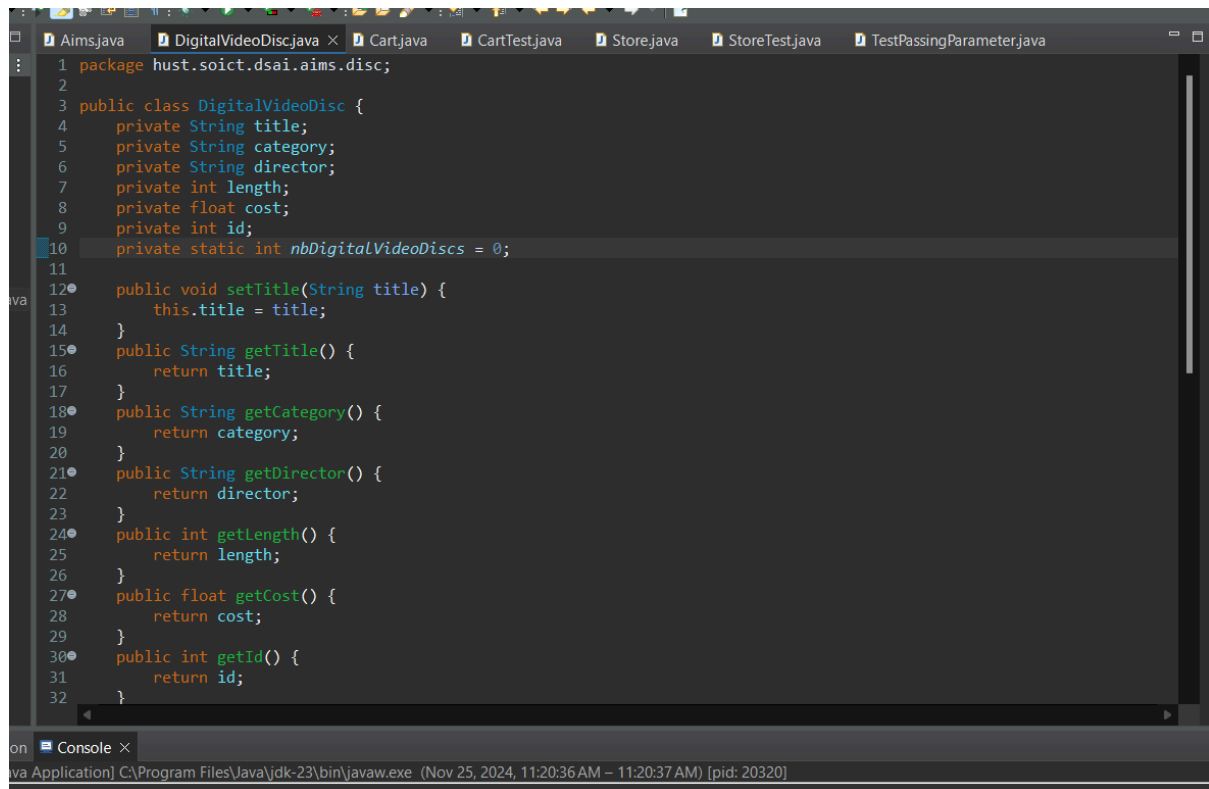
```
10
11 int qtyOrdered = itemsOrdered.size();
12
13 public void addDigitalVideoDisc(DigitalVideoDisc disc) {
14     boolean check = true;
15     if (qtyOrdered < MAX_NUMBERS_ORDERED) {
16         for (int i = 0; i < itemsOrdered.size(); i++) {
17             if (disc.getTitle() == itemsOrdered.get(i).getTitle()) {
18                 System.out.println("The " + disc.getTitle() + "disc has already been in the cart");
19                 check = false;
20             }
21         }
22         if (check == true) {
23             itemsOrdered.add(disc);
24             qtyOrdered++;
25             System.out.println("The " + disc.getTitle() + "disc has been added successfully");
26         }
27     } else {
28         System.out.println("The cart is almost full");
29     }
30 }
31
32 public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList) {
33     if (qtyOrdered + dvdList.length >= MAX_NUMBERS_ORDERED) {
34         System.out.println("The cart is almost full");
35     }
36 }
```

```
27     } else {
28         System.out.println("The cart is almost full");
29     }
30 }
31
32 public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList) {
33     if (qtyOrdered + dvdList.length >= MAX_NUMBERS_ORDERED) {
34         System.out.println("The cart is almost full");
35     } else {
36         for (int i = 0; i < dvdList.length; i++) {
37             addDigitalVideoDisc(dvdList[i]);
38         }
39     }
40 }
41
42 public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
43     if (qtyOrdered + 2 >= MAX_NUMBERS_ORDERED) {
44         System.out.println("The cart is almost full");
45     } else {
46         addDigitalVideoDisc(dvd1);
47         addDigitalVideoDisc(dvd2);
48     }
49 }
50
51 public void removeDigitalVideoDisc(DigitalVideoDisc disc) {
52     boolean check = true;
53     for (int i = 0; i < itemsOrdered.size(); i++) {
54         if (disc.getTitle() == itemsOrdered.get(i).getTitle()) {
55             itemsOrdered.remove(i);
56             System.out.println("The disc has been removed successfully");
57             qtyOrdered--;
58             check = false;
59         }
60     }
61 }
```

## 2.2. Passing Parameter.

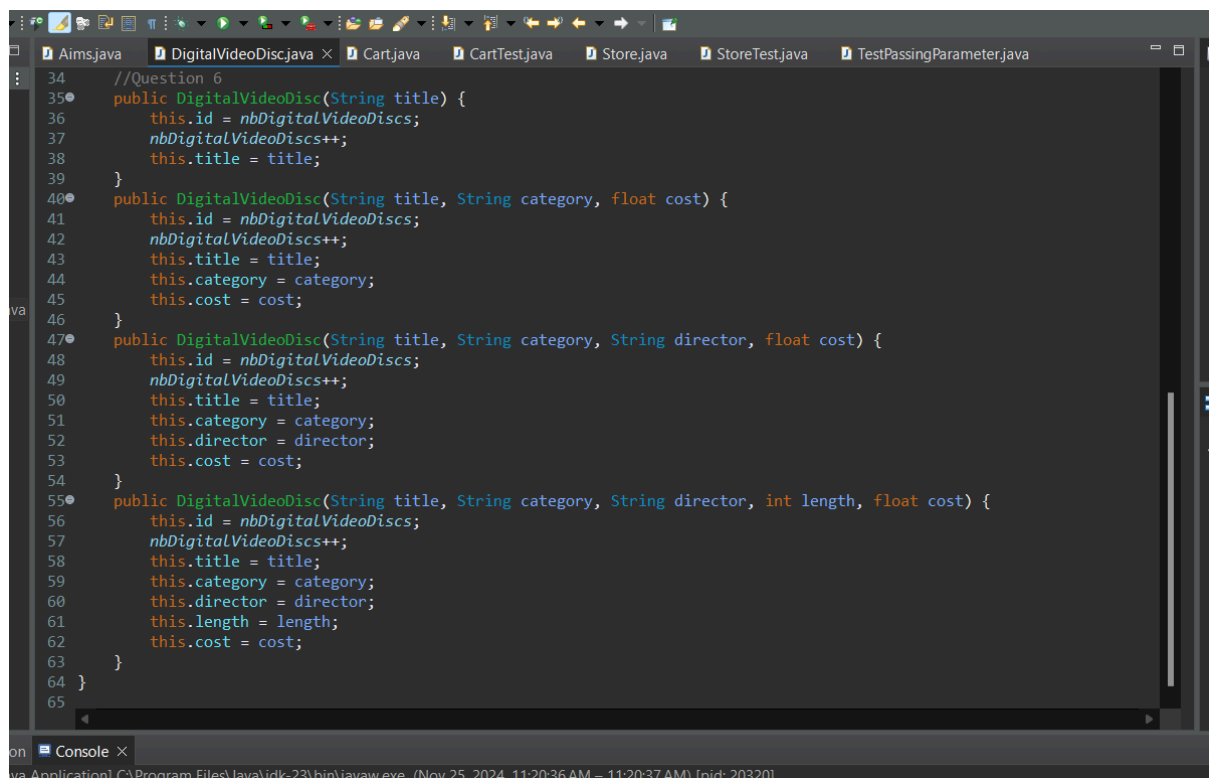
```
1 package hust.soict.dsai.test.disc;
2
3 import hust.soict.dsai.aims.disc.DigitalVideoDisc;
4
5 public class TestPassingParameter {
6     public static void main(String[] args) {
7         DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
8         DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");
9
10        swap(jungleDVD, cinderellaDVD);
11        System.out.println("Jungle dvd title: " + jungleDVD.getTitle());
12        System.out.println("Cinderella dvd title: " + cinderellaDVD.getTitle());
13
14        changeTitle(jungleDVD, cinderellaDVD.getTitle());
15        System.out.println("jungle dvd title: " + jungleDVD.getTitle());
16    }
17
18    public static void swap(Object o1, Object o2) {
19        Object tmp = o1;
20        o1 = o2;
21        o2 = tmp;
22    }
23
24    public static void changeTitle(DigitalVideoDisc dvd, String title) {
25        String oldTitle = dvd.getTitle();
26        dvd.setTitle(title);
27        dvd = new DigitalVideoDisc(oldTitle);
28    }
29 }
30
```

### 2.3. Classifier Member and Instance Member.



This screenshot shows the first part of the `DigitalVideoDisc` class in the `hust.soict.dsai.aims.disc` package. The class has several private attributes: `title`, `category`, `director`, `length`, `cost`, `id`, and a static counter `nbDigitalVideoDiscs` initialized to 0. It also includes getter and setter methods for `title`, `category`, `director`, `length`, and `cost`.

```
1 package hust.soict.dsai.aims.disc;
2
3 public class DigitalVideoDisc {
4     private String title;
5     private String category;
6     private String director;
7     private int length;
8     private float cost;
9     private int id;
10    private static int nbDigitalVideoDiscs = 0;
11
12    public void setTitle(String title) {
13        this.title = title;
14    }
15    public String getTitle() {
16        return title;
17    }
18    public String getCategory() {
19        return category;
20    }
21    public String getDirector() {
22        return director;
23    }
24    public int getLength() {
25        return length;
26    }
27    public float getCost() {
28        return cost;
29    }
30    public int getId() {
31        return id;
32    }
33 }
```



This screenshot shows the constructors for the `DigitalVideoDisc` class. There are four constructors: a no-argument constructor, a constructor with `title`, a constructor with `title`, `category`, and `cost`, and a constructor with `title`, `category`, `director`, and `cost`. Each constructor initializes the `id` attribute using the static counter and increments the counter. The last constructor also initializes the `length` attribute.

```
34 //Question 6
35 public DigitalVideoDisc(String title) {
36     this.id = nbDigitalVideoDiscs;
37     nbDigitalVideoDiscs++;
38     this.title = title;
39 }
40 public DigitalVideoDisc(String title, String category, float cost) {
41     this.id = nbDigitalVideoDiscs;
42     nbDigitalVideoDiscs++;
43     this.title = title;
44     this.category = category;
45     this.cost = cost;
46 }
47 public DigitalVideoDisc(String title, String category, String director, float cost) {
48     this.id = nbDigitalVideoDiscs;
49     nbDigitalVideoDiscs++;
50     this.title = title;
51     this.category = category;
52     this.director = director;
53     this.cost = cost;
54 }
55 public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
56     this.id = nbDigitalVideoDiscs;
57     nbDigitalVideoDiscs++;
58     this.title = title;
59     this.category = category;
60     this.director = director;
61     this.length = length;
62     this.cost = cost;
63 }
64 }
65 }
```

#### 2.4. toString() and print() methods in Cart class.

```
1 Aims.java 2 DigitalVideoDisc.java 3 *Cart.java X 4 CartTest.java 5 Store.java 6 StoreTest.java 7 TestPassingParameter.java
61
62 }
63
64 public float totalCost() {
65     float sum = 0.0f;
66     for (int i = 0; i < itemsOrdered.size(); i++) {
67         sum += itemsOrdered.get(i).getCost();
68     }
69     return sum;
70 }
71
72 public String toString(DigitalVideoDisc disc) {
73     return "DVD - " + disc.getTitle() + " - " + disc.getCategory() + " - " + disc.getDirector()
74         + " - " + disc.getLength() + ": " + disc.getCost() + "$";
75 }
76
77 public void printItemsOfCart() {
78     System.out.println("*****CART*****");
79     for (int i = 0; i < itemsOrdered.size(); i++) {
80         System.out.println(toString(itemsOrdered.get(i)));
81     }
82     System.out.println("Total cost: " + totalCost());
83     System.out.println("*****");
84 }
85
```

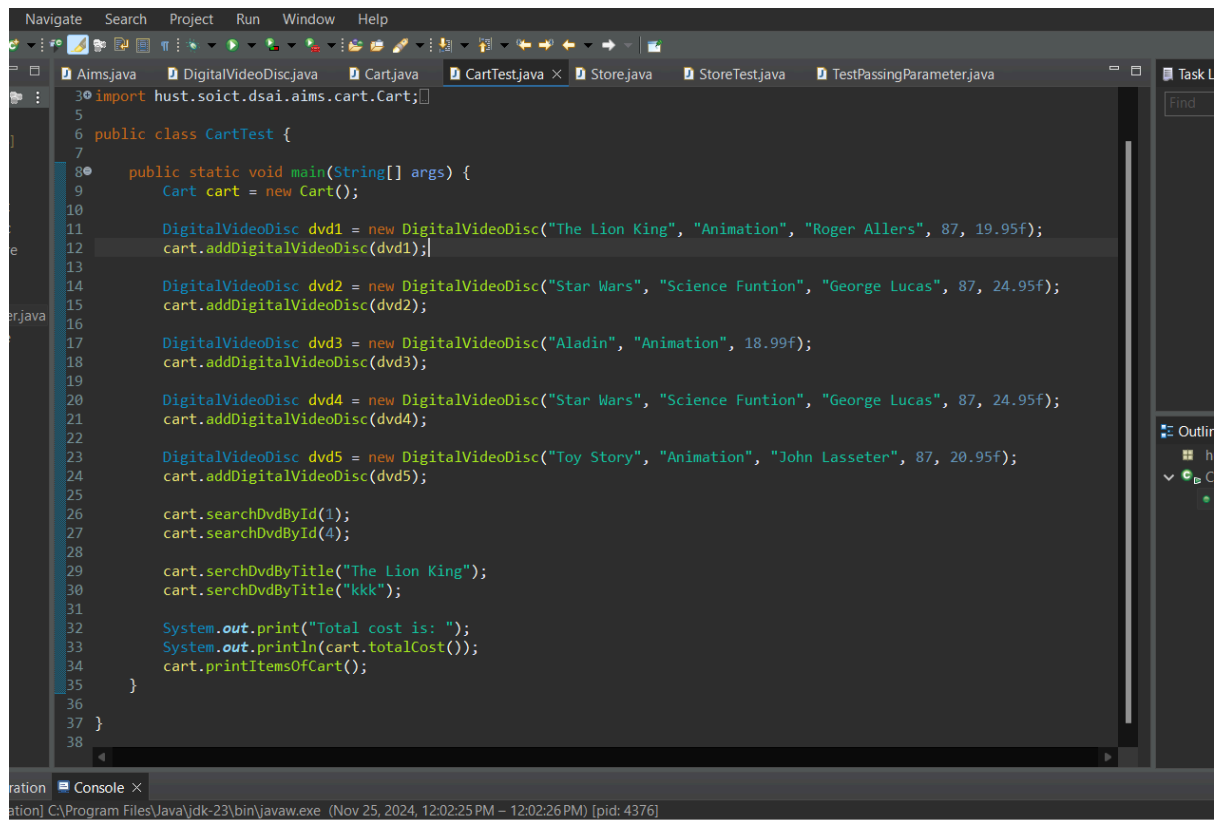
## 2.5. Search by ID and Title method and boolean isMatch(String title) method.

```

86 public void searchDvdById(int id) {
87     boolean check = false;
88     for (int i = 0; i < itemsOrdered.size(); i++) {
89         if (id == itemsOrdered.get(i).getId()) {
90             System.out.println("The DVD is found: " + toString(itemsOrdered.get(i)));
91             check = true;
92         }
93     }
94
95     if (check == false) {
96         System.out.println("The dvd with the id: " + id + " is not found");
97     }
98 }
99
100 public void serchDvdByTitle(String title) {
101     if (isMatching(title)) {
102         for (int i = 0; i < itemsOrdered.size(); i++) {
103             if (title == itemsOrdered.get(i).getTitle()) {
104                 System.out.println("The DVD is found: " + toString(itemsOrdered.get(i)));
105             }
106         }
107     } else {
108         System.out.println("The dvd with the title: " + title + " is not found");
109     }
110 }
111
112 public boolean isMatching(String title) {
113     for (int i = 0; i < itemsOrdered.size(); i++) {
114         if (title == itemsOrdered.get(i).getTitle()) {
115             return true;
116         }
117     }
118 }

```

## 2.6. CartTest class and result.



The screenshot shows an IDE with the `CartTest.java` file open. The code defines a `CartTest` class with a `main` method that creates a `Cart` object, adds five `DigitalVideoDisc` objects, and then performs searches and prints the total cost and items.

```
30 import hust.soict.dsai.aims.cart.Cart;
5
6 public class CartTest {
7
8     public static void main(String[] args) {
9         Cart cart = new Cart();
10
11         DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
12         cart.addDigitalVideoDisc(dvd1);
13
14         DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Funtion", "George Lucas", 87, 24.95f);
15         cart.addDigitalVideoDisc(dvd2);
16
17         DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
18         cart.addDigitalVideoDisc(dvd3);
19
20         DigitalVideoDisc dvd4 = new DigitalVideoDisc("Star Wars", "Science Funtion", "George Lucas", 87, 24.95f);
21         cart.addDigitalVideoDisc(dvd4);
22
23         DigitalVideoDisc dvd5 = new DigitalVideoDisc("Toy Story", "Animation", "John Lasseter", 87, 20.95f);
24         cart.addDigitalVideoDisc(dvd5);
25
26         cart.searchDvdById(1);
27         cart.searchDvdById(4);
28
29         cart.serchDvdByTitle("The Lion King");
30         cart.serchDvdByTitle("kkk");
31
32         System.out.print("Total cost is: ");
33         System.out.println(cart.totalCost());
34         cart.printItemsOfCart();
35     }
36 }
37
38
```

The console output shows the results of the execution:

```
ation Console ×
ation] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 25, 2024, 12:02:25 PM – 12:02:26 PM) [pid: 4376]
Total cost is: 109.74
The Lion King
Star Wars
Aladin
Star Wars
Toy Story
```

The screenshot shows an IDE with a project named 'AIMSProject'. The package explorer on the left shows a hierarchy of packages: 'src' contains 'hust.soict.dsai.aims', 'hust.soict.dsai.aims.cart', 'hust.soict.dsai.aims.disc', 'hust.soict.dsai.aims.store', 'hust.soict.dsai.test.cart', 'hust.soict.dsai.test.disc' (containing 'TestPassingParameter.java'), 'hust.soict.dsai.test.store', 'soict.dsai', and 'module-info.java'. Other packages like 'DaysOfMonth', 'HelloJava', 'JavaBasics', 'OtherProjects', and 'StartTriangle' are also visible.

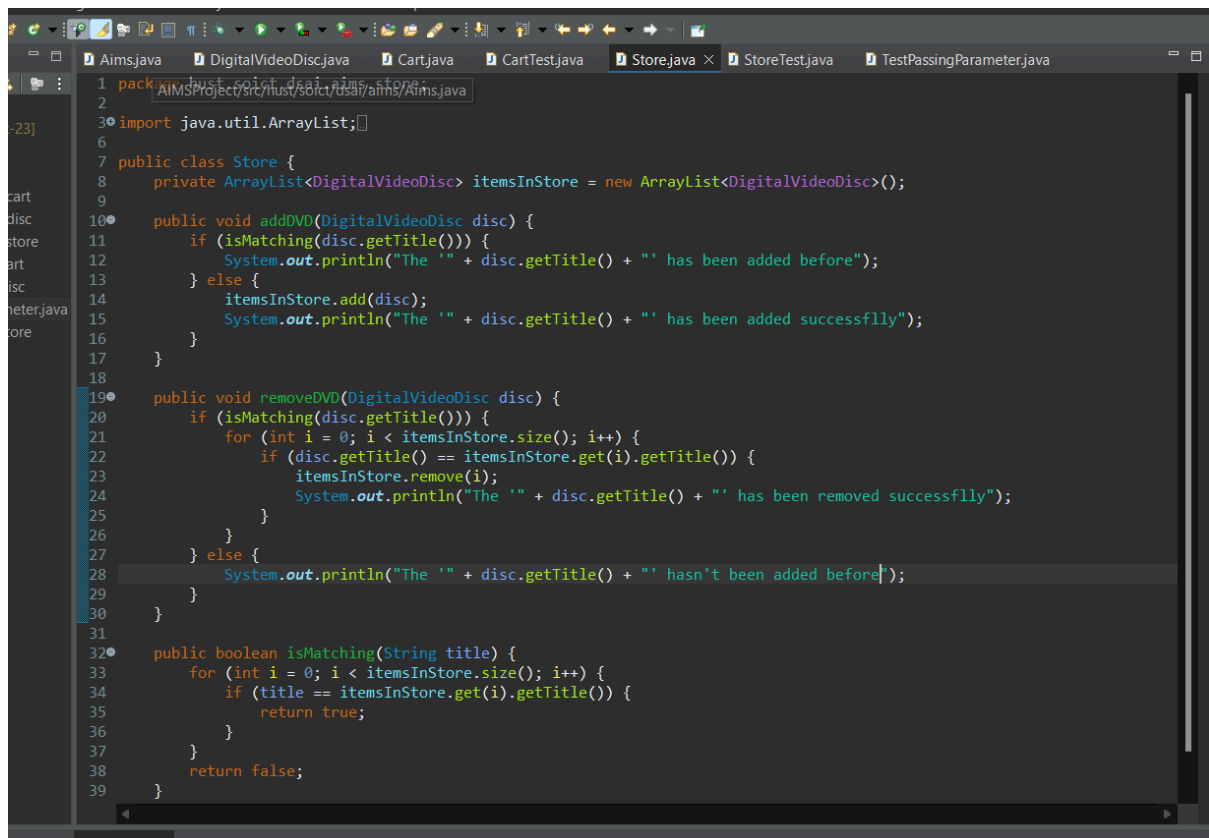
The main editor displays the 'CartTest.java' file with the following code:

```
1 import hust.soict.dsai.aims.cart.Cart;
2
3
4
5
6 public class CartTest {
7
8     public static void main(String[] args) {
9         Cart cart = new Cart();
10
11         DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers");
12         cart.addDigitalVideoDisc(dvd1);
13
14         DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Funtion", "George Lucas");
15         cart.addDigitalVideoDisc(dvd2);
16
17         DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
18         cart.addDigitalVideoDisc(dvd3);
19
20         DigitalVideoDisc dvd4 = new DigitalVideoDisc("Star Wars", "Science Funtion", "George Lucas");
21         cart.addDigitalVideoDisc(dvd4);
22
23         DigitalVideoDisc dvd5 = new DigitalVideoDisc("Toy Story", "Animation", "John Lasseter");
24         cart.addDigitalVideoDisc(dvd5);
25
26         cart.searchDvdById(1);
27         cart.searchDvdById(4);
28     }
29 }
```

The bottom panel shows the 'Console' output for the 'CartTest [Java Application]' run. The output is as follows:

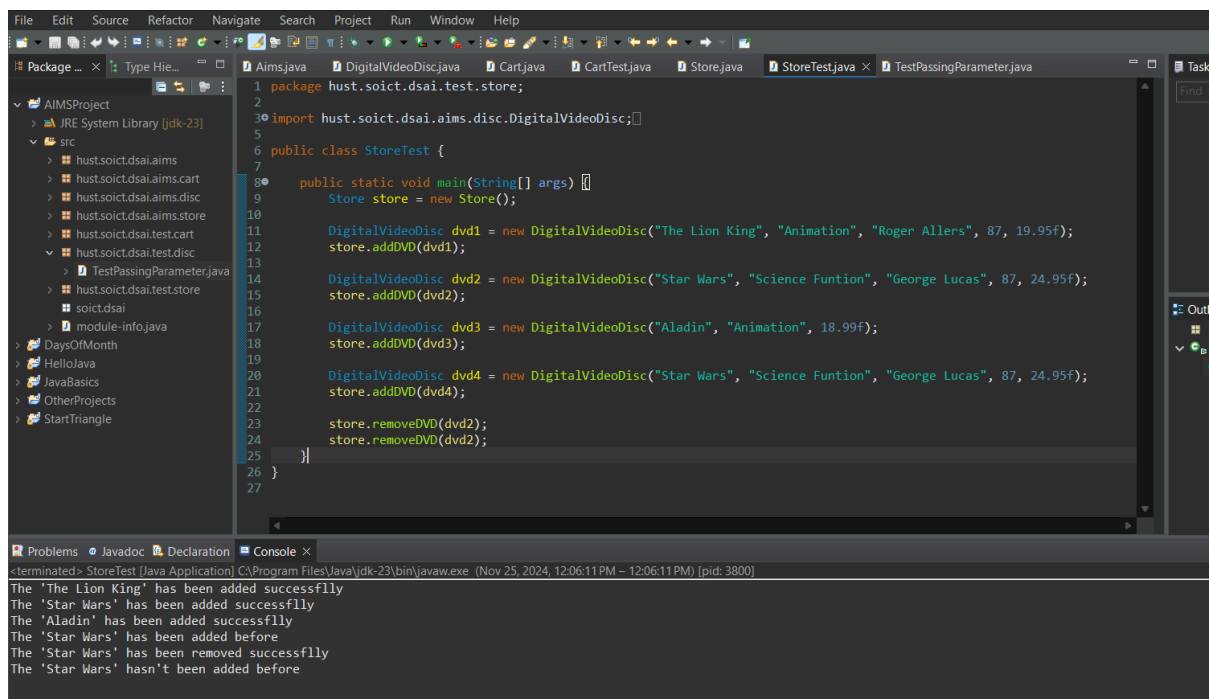
```
<terminated> CartTest [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 25, 2024, 12:02:25 PM - 12:02:26 PM) [pid: 4376]
The The Lion Kingdisc has been added successfully
The Star Warsdisc has been added successfully
The Aladindisc has been added successfully
The Star Warsdisc has already been in the cart
The Toy Storydisc has been added successfully
The DVD is found: DVD - Star Wars - Science Funtion - George Lucas - 87: 24.95$
The DVD is found: DVD - Toy Story - Animation - John Lasseter - 87: 20.95$
The DVD is found: DVD - The Lion King - Animation - Roger Allers - 87: 19.95$
The dvd with the title: 'kkk' is not found
Total cost is: 84.84
*****CART*****
DVD - The Lion King - Animation - Roger Allers - 87: 19.95$
DVD - Star Wars - Science Funtion - George Lucas - 87: 24.95$
DVD - Aladin - Animation - null - 0: 18.99$
DVD - Toy Story - Animation - John Lasseter - 87: 20.95$
Total cost: 84.84
*****
```

## 2.7. Store class.



```
1 package hust.soiect.dsai.aims.store;
2
3 import java.util.ArrayList;
4
5
6
7 public class Store {
8     private ArrayList<DigitalVideoDisc> itemsInStore = new ArrayList<DigitalVideoDisc>();
9
10    public void addDVD(DigitalVideoDisc disc) {
11        if (isMatching(disc.getTitle())) {
12            System.out.println("The '" + disc.getTitle() + "' has been added before");
13        } else {
14            itemsInStore.add(disc);
15            System.out.println("The '" + disc.getTitle() + "' has been added successffly");
16        }
17    }
18
19    public void removeDVD(DigitalVideoDisc disc) {
20        if (isMatching(disc.getTitle())) {
21            for (int i = 0; i < itemsInStore.size(); i++) {
22                if (disc.getTitle() == itemsInStore.get(i).getTitle()) {
23                    itemsInStore.remove(i);
24                    System.out.println("The '" + disc.getTitle() + "' has been removed successffly");
25                }
26            }
27        } else {
28            System.out.println("The '" + disc.getTitle() + "' hasn't been added before");
29        }
30    }
31
32    public boolean isMatching(String title) {
33        for (int i = 0; i < itemsInStore.size(); i++) {
34            if (title == itemsInStore.get(i).getTitle()) {
35                return true;
36            }
37        }
38        return false;
39    }
40 }
```

## 2.8. StoreTest class and result.



```
1 package hust.soiect.dsai.test.store;
2
3 import hust.soiect.dsai.aims.disc.DigitalVideoDisc;
4
5
6 public class StoreTest {
7
8    public static void main(String[] args) {
9        Store store = new Store();
10
11        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
12        store.addDVD(dvd1);
13
14        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Funtion", "George Lucas", 87, 24.95f);
15        store.addDVD(dvd2);
16
17        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
18        store.addDVD(dvd3);
19
20        DigitalVideoDisc dvd4 = new DigitalVideoDisc("Star Wars", "Science Funtion", "George Lucas", 87, 24.95f);
21        store.addDVD(dvd4);
22
23        store.removeDVD(dvd2);
24        store.removeDVD(dvd2);
25    }
26 }
27
```

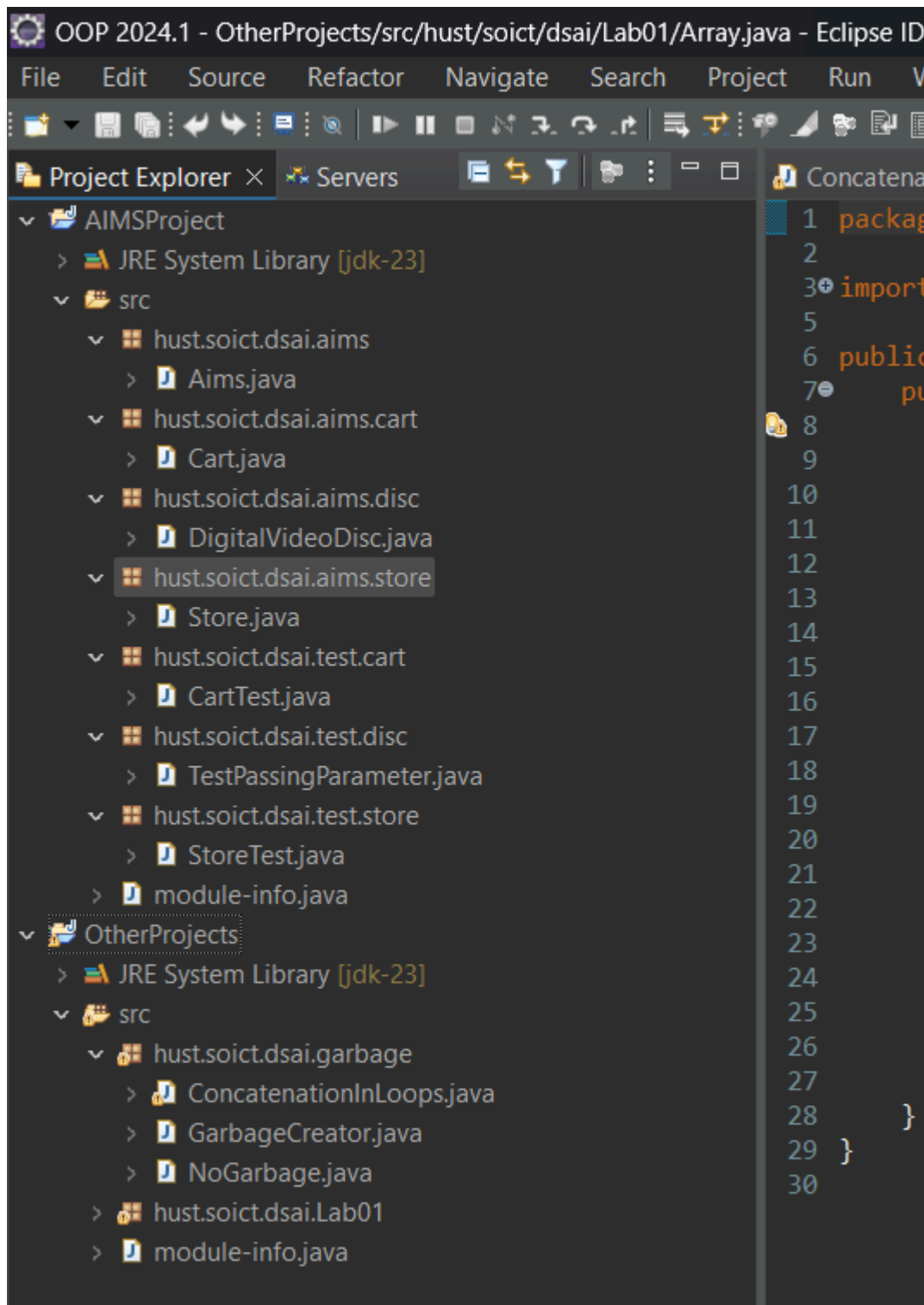
Problems Javadoc Declaration Console

<terminated> StoreTest [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 25, 2024, 12:06:11 PM – 12:06:11 PM) [pid: 3800]

The 'The Lion King' has been added successffly  
The 'Star Wars' has been added successffly  
The 'Aladin' has been added successffly  
The 'Star Wars' has been added before  
The 'Star Wars' has been removed successffly  
The 'Star Wars' hasn't been added before

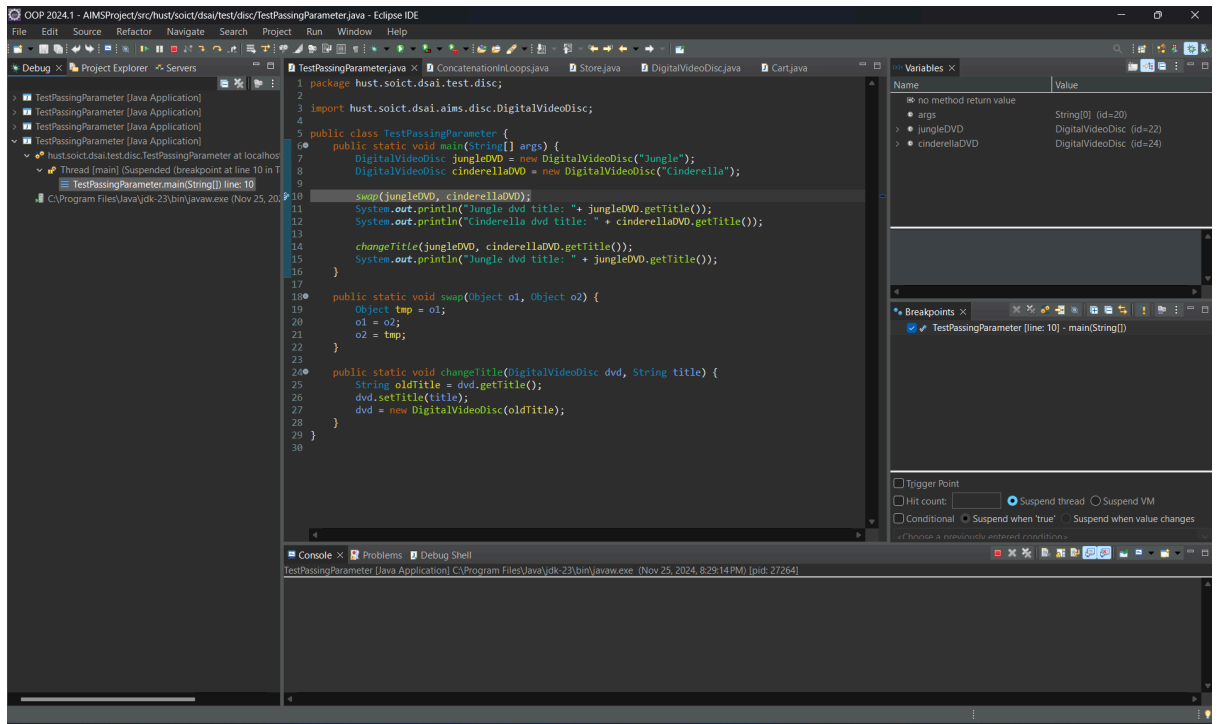


## 2.9. Re-organize project.

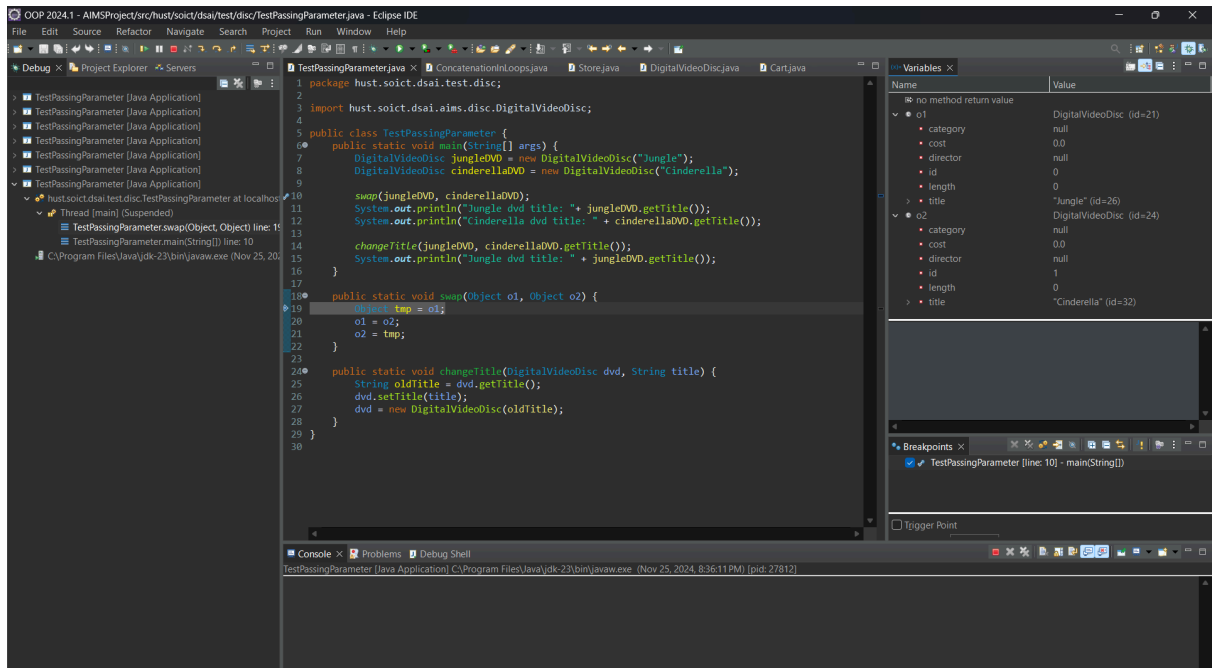


## 3. Code Debugging and Results.

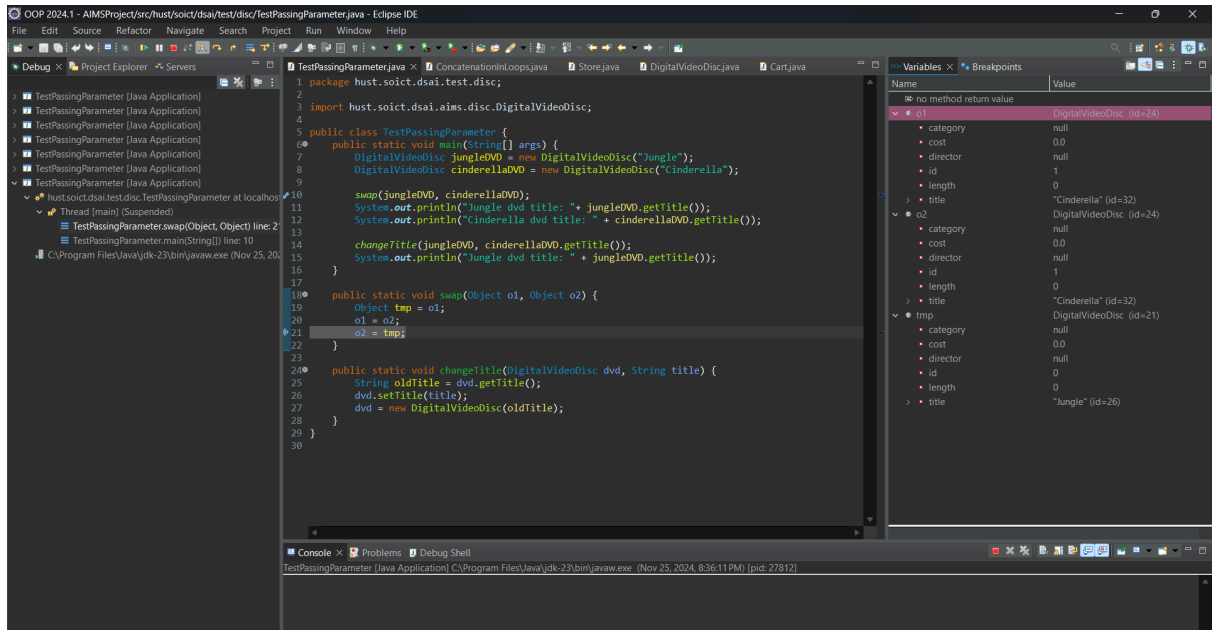
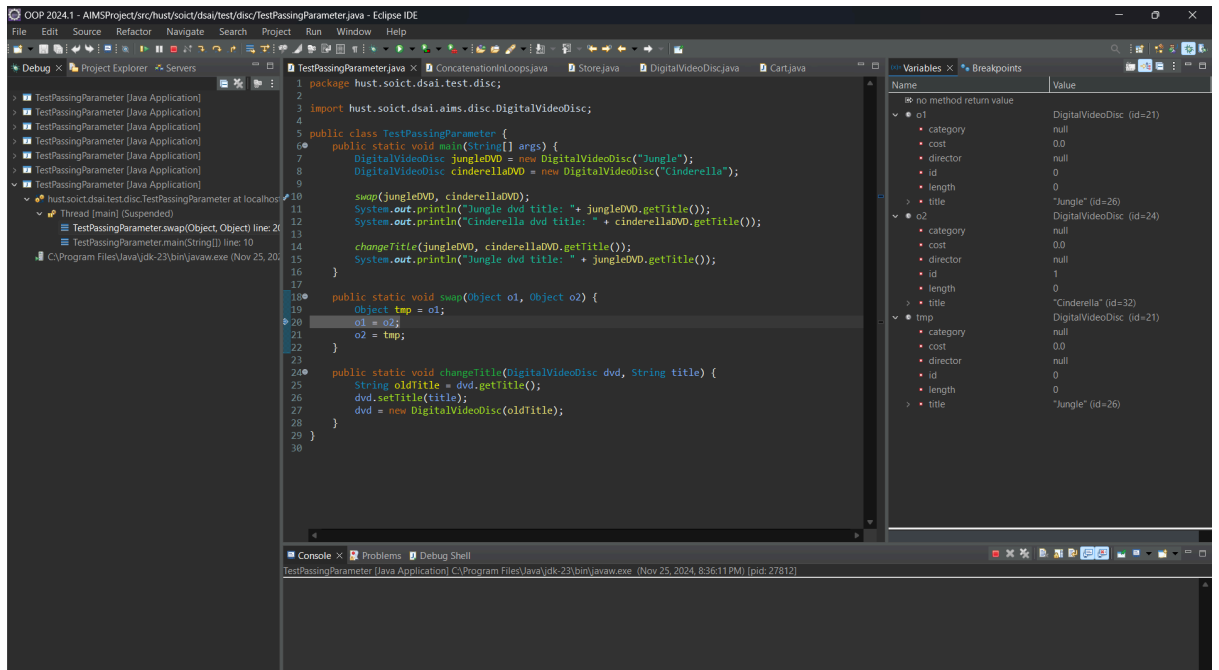
Breakpoint

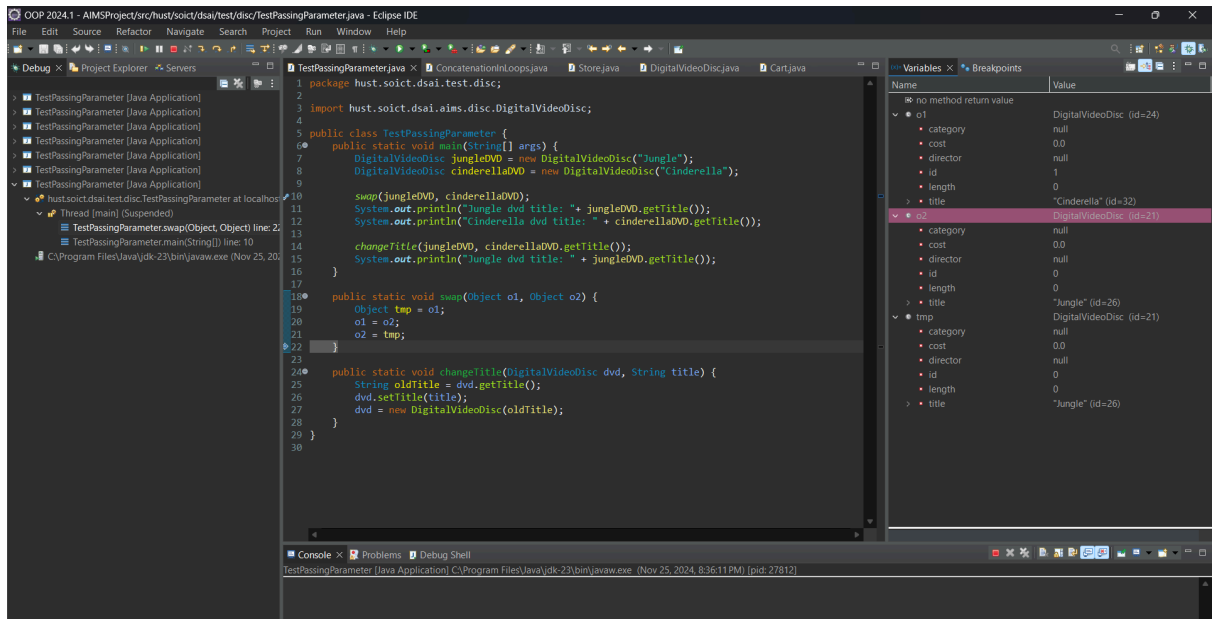


## Step into

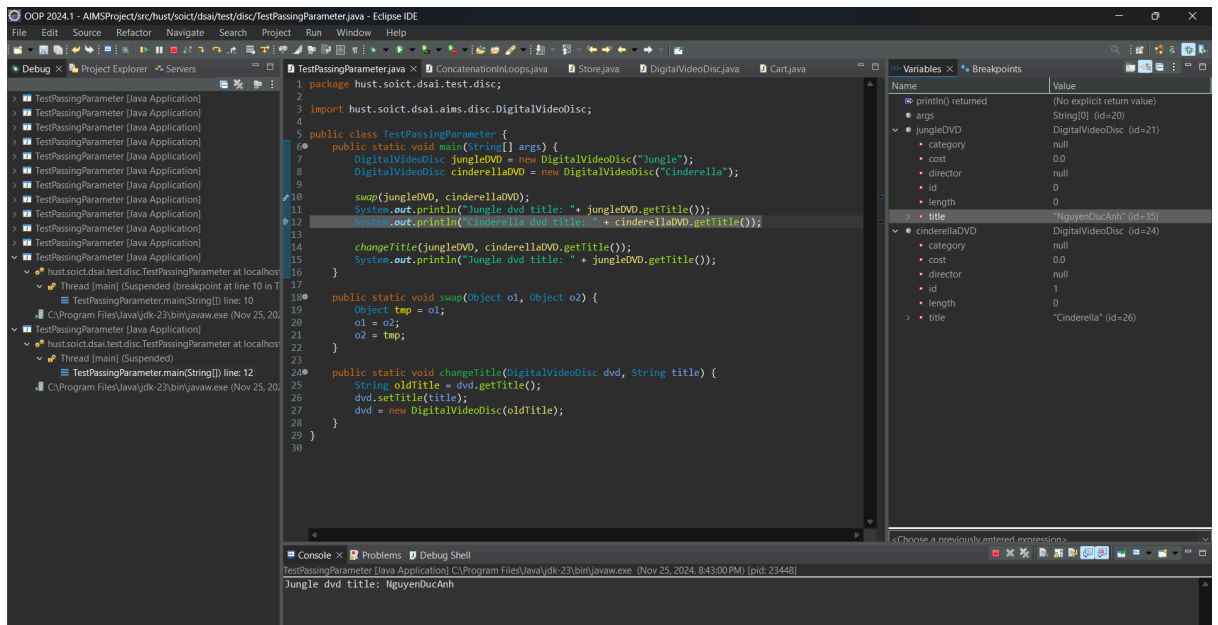


## Step over





## Result



## 4. Answer Questions.

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

- Java is a Pass by Value programming language.

4.2. After the call of **swap(jungleDVD, cinderellaDVD)** why does the title of these two objects still remain?

- Because the Java **swap ()** method is exchanging the values of the title fields between the two objects, but it's not changing the object references themselves. So the jungleDVD and cinderellaDVD object references still point to the same objects in memory as before the **swap ()** method call.

4.3. After the call of **changeTitle(jungleDVD, cinderellaDVD.getTitle())** why is the title of the JungleDVD changed?

- Because the changeTitle() method is modifying the title field of the jungleDVD object directly using the setter method.

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

- The return type of this method should be **String**.