

Lab 03: Basic Object-Oriented Techniques – Report

Trần Khánh Quỳnh – 20225762

1. New written code

- a) Method addDigitalVideoDisc which allows to pass an arbitrary number of arguments for dvd

```
public void addDigitalVideoDisc(DigitalVideoDisc... dvdList) {
    //arbitrary number of arguments instead of array parameter
    if (qtyOrdered == MAX_NUMBERS_ORDERED) {
        System.out.println("The cart is full");
        return;
    }

    int numberOfDVDs = dvdList.length;

    if (qtyOrdered + numberOfDVDs > MAX_NUMBERS_ORDERED) {
        System.out.println("Cannot add all DVDs. Cart would exceed maximum capacity.");
        return;
    }

    for(DigitalVideoDisc dvd: dvdList) {
        itemsOrdered[qtyOrdered++] = dvd;
    }

    System.out.println("List of DVDs has been added");
}
```

```
List of DVDs has been added
Total cost is: 63.89
```

- b) Method addDigitalVideoDisc with 2 parameters

```
public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
    if (qtyOrdered == MAX_NUMBERS_ORDERED) {
        System.out.println("The cart is full");
        return;
    }
    if (qtyOrdered + 2 > MAX_NUMBERS_ORDERED) {
        System.out.println("Cannot add all DVDs. Cart would exceed maximum capacity.");
        return;
    }
    itemsOrdered[qtyOrdered++] = dvd1;
    itemsOrdered[qtyOrdered++] = dvd2;

    System.out.println("2 DVDs has been added");
}
```

```
2 DVDs has been added  
Total cost is: 44.90
```

c) TestPassingParameter.java

```
package aims;  
  
public class TestPassingParameter {  
    public static void main(String[] args) {  
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");  
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");  
  
        swap(jungleDVD, cinderellaDVD);  
        System.out.println("Jungle DVD title: " + jungleDVD.getTitle());  
        System.out.println("Cinderella DVD title: " + cinderellaDVD.getTitle());  
  
        changeTitle(jungleDVD, cinderellaDVD.getTitle());  
        System.out.println("Jungle DVD title: " + jungleDVD.getTitle());  
    }  
  
    public static void swap(Object o1, Object o2) {  
        Object tmp = o1;  
        o1 = o2;  
        o2 = tmp;  
    }  
  
    public static void changeTitle(DigitalVideoDisc dvd, String title) {  
        String oldTitle = dvd.getTitle();  
        dvd.setTitle(title);  
        dvd = new DigitalVideoDisc(oldTitle);  
    }  
}
```

```
Jungle DVD title: Jungle  
Cinderella DVD title: Cinderella  
Jungle DVD title: Cinderella
```

d) Keep track of the number of DVDs created

```

1 package aims;
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
10    //New
11    private static int nbDigitalVideoDiscs = 0;
12    private int id;
13
14    public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
15        super();
16        this.title = title;
17        this.category = category;
18        this.director = director;
19        this.length = length;
20        this.cost = cost;
21
22        //New
23        nbDigitalVideoDiscs++;
24        this.id = nbDigitalVideoDiscs;
25    }

```

```

26    public DigitalVideoDisc(String title, String category, String director, float cost) {
27        super();
28        this.title = title;
29        this.category = category;
30        this.director = director;
31        this.cost = cost;
32
33        //New
34        nbDigitalVideoDiscs++;
35        this.id = nbDigitalVideoDiscs;
36    }
37    public DigitalVideoDisc(String title, String category, float cost) {
38        super();
39        this.title = title;
40        this.category = category;
41        this.cost = cost;
42
43        //New
44        nbDigitalVideoDiscs++;
45        this.id = nbDigitalVideoDiscs;
46    }
47    public DigitalVideoDisc(String title) {
48        super();
49        this.title = title;
50
51        //New
52        nbDigitalVideoDiscs++;
53        this.id = nbDigitalVideoDiscs;
54    }
55

```

```

•    public int getId() {
        return id;
    }

```

e) Method to print the list of ordered items of a cart, the price of each item, and the total price:

-toString() method in DigitalVideoDisc class:

```

81 public String toString(){
82     return String.format("DVD - %s - %s - %s -%d: %.2f $",
83         title,
84         category,
85         director,
86         length,
87         cost);
88 }
89

```

-printCart() method in Cart class:

```

89 public void printCart() {
90     System.out.println("*****CART*****");
91     System.out.println("Ordered Items:");
92     float totalcost = 0;
93     for(int i = 0; i < qtyOrdered; i++) {
94         System.out.printf("%d. %s\n",i+1,itemsOrdered[i].toString());
95         totalcost += itemsOrdered[i].getCost();
96     }
97     System.out.println(String.format("Total cost: %.2f", totalcost));
98     System.out.println("*****");
99 }
100 }

```

-Run printCart():

```

3 public class Aims {
4     public static void main(String[] args) {
5         Cart anOrder = new Cart();
6
7         DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King","Animation", "Roger Allers", 87, 19.95f);
8         //anOrder.addDigitalVideoDisc(dvd1);
9         DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star wars","Science Fiction", "Geogre Lucas",87,24.95f);
10        //anOrder.addDigitalVideoDisc(dvd2);
11        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin","Animation",18.99f);
12        anOrder.addDigitalVideoDisc(dvd1, dvd2, dvd3);
13
14        System.out.println("Total cost is: " + String.format("%.2f", anOrder.totalCost()));
15        //anOrder.removeDigitalVideoDisc(dvd1);
16        //System.out.println("Total cost is: " + String.format("%.2f", anOrder.totalCost()));
17        anOrder.printCart();
18    }
19 }

```

```

<terminated> Aims (3) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Nov 25, 2024, 1:40:06 PM – 1:40:07 PM) [pid: 1824]
List of DVDs has been added
Total cost is: 63.89
*****CART*****
Ordered Items:
1. DVD - The Lion King - Animation - Roger Allers -87: 19.95 $
2. DVD - Star wars - Science Fiction - Geogre Lucas -87: 24.95 $
3. DVD - Aladin - Animation - null -0: 18.99 $
Total cost: 63.89
*****

```

f) Search for DVDs in the cart by ID and display the search results:

```

101 //Search for DVDs in the cart by ID
102 public DigitalVideoDisc searchByID(int id) {
103     for(int i = 0; i < qtyOrdered; i++) {
104         if(itemsOrdered[i].getId() == id) {
105             System.out.println("DVD found:");
106             System.out.println(itemsOrdered[i].toString());
107             return itemsOrdered[i];
108         }
109     }
110
111     System.out.println("No match is found!");
112     return null;
113 }

```

Result:

If ID = 1

```

17         anOrder.printCart();
18         anOrder.searchByID(1);
19     }

```

```

Total cost is: 63.89
*****CART*****
Ordered Items:
1. DVD - The Lion King - Animation - Roger Allers -87: 19.95 $
2. DVD - Star wars - Science Fiction - Geogre Lucas -87: 24.95 $
3. DVD - Aladin - Animation - null -0: 18.99 $
Total cost: 63.89
*****
DVD found:
DVD - The Lion King - Animation - Roger Allers -87: 19.95 $

```

If ID = 0

```

17         anOrder.printCart();
18         anOrder.searchByID(0);
19     }

```

```

List of DVDs has been added
Total cost is: 63.89
*****CART*****
Ordered Items:
1. DVD - The Lion King - Animation - Roger Allers -87: 19.95 $
2. DVD - Star wars - Science Fiction - Geogre Lucas -87: 24.95 $
3. DVD - Aladin - Animation - null -0: 18.99 $
Total cost: 63.89
*****
No match is found!

```

g) Search for DVDs in the cart by title

```

115 //Search for DVDs by title
116 public DigitalVideoDisc searchByTitle(String title) {
117     for(int i = 0; i < qtyOrdered; i++) {
118         if(itemsOrdered[i].getTitle() == title) {
119             System.out.println("DVD found:");
120             System.out.println(itemsOrdered[i].toString());
121             return itemsOrdered[i];
122         }
123     }
124     System.out.println("No match is found!");
125     return null;
126 }
127 }
128

```

Result:

If title is "Star wars":

```

        anOrder.printCart();
        //anOrder.searchByID(0);
        anOrder.searchByTitle("Star wars");
    }
}

```

```

DVD found:
DVD - Star wars - Science Fiction - Geogre Lucas - 87: 24.95 $

```

If title is Cinderella

```

17         anOrder.printCart();
18         //anOrder.searchByID(0);
19         anOrder.searchByTitle("Cinderella");
20     }
21 }

```

```

*****
No match is found!

```

h) CartTest() method:

```

3 public class CartTest {
4     public static void main(String[] args) {
5         Cart cart = new Cart();
6
7         DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
8         cart.addDigitalVideoDisc(dvd1);
9         DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star wars", "Science Fiction", "George Lucas", 87, 24.95f);
10        cart.addDigitalVideoDisc(dvd2);
11        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
12        cart.addDigitalVideoDisc(dvd3);
13
14        //Test print method
15        cart.printCart();
16        //test search method
17        cart.searchByID(0);
18        cart.searchByID(2);
19        cart.searchByTitle("Aladin");
20        cart.searchByTitle("Sofia the first");
21
22    }
23 }
24 }
25

```

Result:

```
The DVD has been added
The DVD has been added
The DVD has been added
*****CART*****
Ordered Items:
1. DVD - The Lion King - Animation - Roger Allers - 87: 19.95 $
2. DVD - Star wars - Science Fiction - George Lucas - 87: 24.95 $
3. DVD - Aladin - Animation - null - 0: 18.99 $
Total cost: 63.89
*****
No match is found!
DVD found:
DVD - Star wars - Science Fiction - George Lucas - 87: 24.95 $
DVD found:
DVD - Aladin - Animation - null - 0: 18.99 $
No match is found!
```

i) Implement the Store class

```
1 package aims;
2
3 public class Store {
4     private static final int MAX_ITEMS = 100;
5     private DigitalVideoDisc[] itemsinStore;
6     private int itemCount = 0;
7
8     public Store() {
9         itemsinStore = new DigitalVideoDisc[MAX_ITEMS];
10    }
11
12    public void addDVD(DigitalVideoDisc dvd) {
13        if (itemCount >= MAX_ITEMS) {
14            System.out.println("The store is full!");
15            return;
16        }
17
18        itemsinStore[itemCount++] = dvd;
19        System.out.println("DVD added successfully to the store!");
20    }
21
22    public void removeDVD(DigitalVideoDisc dvd) {
23        if (itemCount == 0) {
24            System.out.println("The store is empty!");
25            return;
26        }
27
28        boolean found = false;
29        for (int i = 0; i < itemCount; i++) {
30            if (itemsinStore[i] == dvd) {
31
32                for (int j = i; j < itemCount - 1; j++) {
33                    itemsinStore[j] = itemsinStore[j + 1];
34                }
35                itemsinStore[itemCount - 1] = null;
36                itemCount--;
37                found = true;
38                System.out.println("DVD removed successfully from the store");
39                break;
40            }
41        }
42
43        if (!found) {
44            System.out.println("DVD not found in the store");
45        }
46    }
47 }
```

-StoreTest class:

```

1 package aims;
2
3 public class StoreTest {
4     public static void main(String[] args) {
5         Store store = new Store();
6
7         DigitalVideoDisc dvd1 = new DigitalVideoDisc("Jurassic Park", "Adventure", "Steven Spielberg", 127, 22.99f);
8         DigitalVideoDisc dvd2 = new DigitalVideoDisc("The Matrix", "Science Fiction", "Wachowski Sisters", 136, 25.99f);
9
10        // Test adding DVDs
11        store.addDVD(dvd1);
12        store.addDVD(dvd2);
13
14        // Test removing DVDs
15        store.removeDVD(dvd1);
16        store.removeDVD(dvd1);
17    }
18 }
19

```

Result:

```

DVD added successfully to the store!
DVD added successfully to the store!
DVD removed successfully from the store
DVD not found in the store

```

j) String, StringBuilder and StringBuffer

-ConcatenationInLoops.java:

```

1 package hust.soict.globalict.garbage;
2
3 import java.util.Random;
4
5 public class ConcatenationInLoops {
6     public static void main(String[] args) {
7         Random r = new Random(123);
8         long start = System.currentTimeMillis();
9         String s = "";
10        for(int i = 0; i < 65536; i++) {
11            s += r.nextInt(2);
12            System.out.println(System.currentTimeMillis() - start);
13        }
14
15        r = new Random(123);
16        start = System.currentTimeMillis();
17        StringBuilder sb = new StringBuilder();
18        for(int i = 0; i < 65536; i++) {
19            sb.append(r.nextInt(2));
20        }
21        s = sb.toString();
22        System.out.println(System.currentTimeMillis() - start);
23    }
24 }

```

-GarbageCreator.java


```

1 package hust.soict.globalict.garbage;
2
3 import java.io.IOException;
4
5
6
7 public class GarbageCreator {
8     public static void main(String[] args) {
9         String filename = "D:/OOPLab/OtherProjects/hust/soict/globalict/garbage/quynh.txt";
10        String outputString = "";
11        byte[] inputBytes = null;
12        long startTime, endTime;
13
14        try {
15            inputBytes = Files.readAllBytes(Paths.get(filename));
16        } catch (IOException e) {
17            e.printStackTrace();
18            return;
19        }
20
21        startTime = System.currentTimeMillis();
22        for (byte b : inputBytes) {
23            outputString += (char) b;
24        }
25        endTime = System.currentTimeMillis();
26
27        System.out.println("Thời gian sử dụng toán tử '+': " + (endTime - startTime) + " ms");
28    }
29 }
30

```

Result:

```

Thời gian sử dụng toán tử '+': 39 ms

```

-NoGarbage.java

```

1 package hust.soict.globalict.garbage;
2
3 import java.io.IOException;
4 import java.nio.file.Files;
5 import java.nio.file.Paths;
6
7 public class NoGarbage {
8     public static void main(String[] args) {
9         String filename = "D:/OOPLab/OtherProjects/hust/soict/globalict/garbage/quynh.txt";
10        StringBuffer outputBuffer = new StringBuffer();
11        byte[] inputBytes = null;
12        long startTime, endTime;
13
14        try {
15            inputBytes = Files.readAllBytes(Paths.get(filename));
16        } catch (IOException e) {
17            e.printStackTrace();
18            return;
19        }
20
21        startTime = System.currentTimeMillis();
22        for (byte b : inputBytes) {
23            outputBuffer.append((char) b);
24        }
25        endTime = System.currentTimeMillis();
26
27        System.out.println("Thời gian sử dụng StringBuffer: " + (endTime - startTime) + " ms");
28    }
29 }

```

Result:

```
<terminated> NoGarbage [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Nov 25, 2024, 3:24:16 PM - 3:24:16 PM) [pid: 11232]  
Thời gian sử dụng StringBuffer: 1 ms
```

2. Code debugging and results

-Set a breakpoint:

```
2  
3 public class TestPassingParameter {  
4     public static void main(String[] args) {  
5         DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");  
6         DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");  
7  
8         swap(jungleDVD, cinderellaDVD);  
9         System.out.println("Jungle DVD title: " + jungleDVD.getTitle());  
10        System.out.println("Cinderella DVD title: " + cinderellaDVD.getTitle());  
11  
12        changeTitle(jungleDVD, cinderellaDVD.getTitle());  
13        System.out.println("Jungle DVD title: " + jungleDVD.getTitle());  
14    }  
15 }  
16
```

-Step into swap function:

```
16  
17 public static void swap(Object o1, Object o2) {  
18     Object tmp = o1;  
19     o1 = o2;  
20     o2 = tmp;  
21 }  
22
```

+Variable perspective

The image displays three sequential screenshots of the Variable Perspective in an IDE, illustrating the state of variables during the execution of the `swap` function.

- Left Screenshot:** Shows the initial state where `o1` and `o2` are objects of type `DigitalVideoDisc`. `o1` has a `title` of "Jungle" and `o2` has a `title` of "Cinderella". A temporary variable `tmp` is also present, holding the reference to `o1`.
- Middle Screenshot:** Shows the state after the first assignment `Object tmp = o1;`. The `tmp` variable now holds the reference to the object previously held by `o1`.
- Right Screenshot:** Shows the state after the second assignment `o1 = o2;`. The `o1` variable now holds the reference to the object previously held by `o2` (the "Cinderella" object).

-Return to main function

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

Console × Problems Debug Shell

TestPassingParameter [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Nov 25, 2024, 12:50:52 AM) [pid: 16624]

Jungle DVD title: Jungle

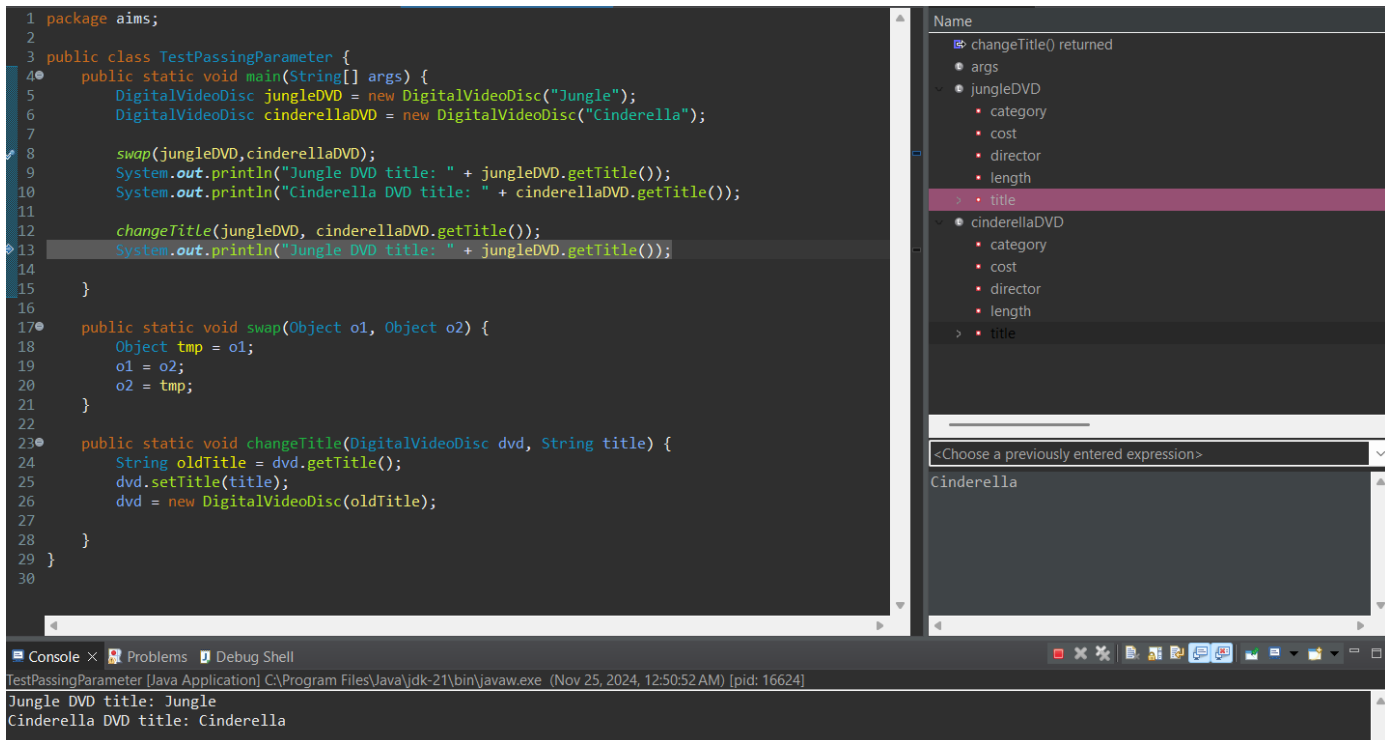
```
1 package aims;
2
3 public class TestPassingParameter {
4     public static void main(String[] args) {
5         DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
6         DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");
7
8         swap(jungleDVD, cinderellaDVD);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

Console × Problems Debug Shell

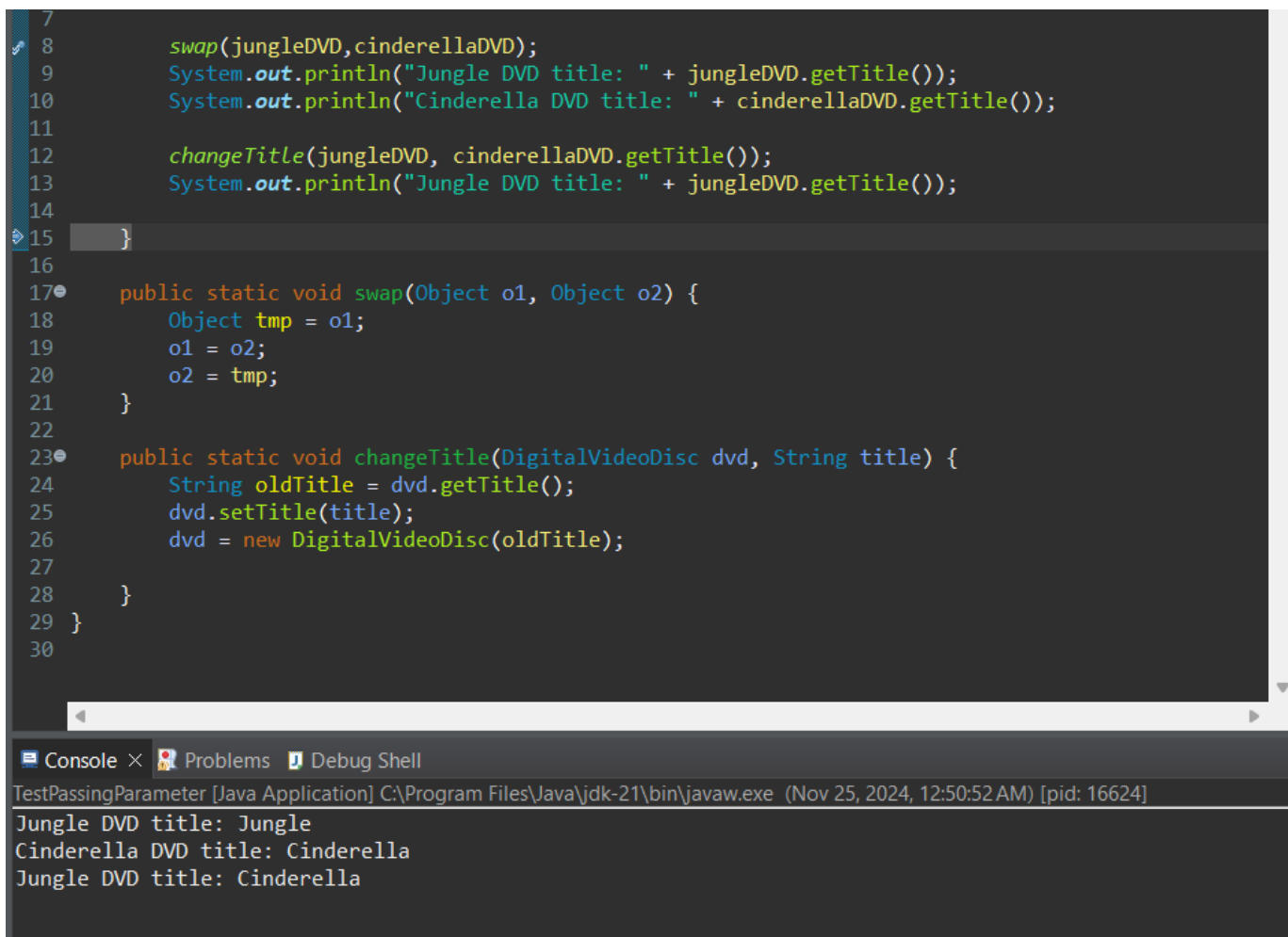
TestPassingParameter [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Nov 25, 2024, 12:50:52 AM) [pid: 16624]

Jungle DVD title: Jungle
Cinderella DVD title: Cinderella

-After running the changeTitle function: the title of jungle DVD is now Cinderella

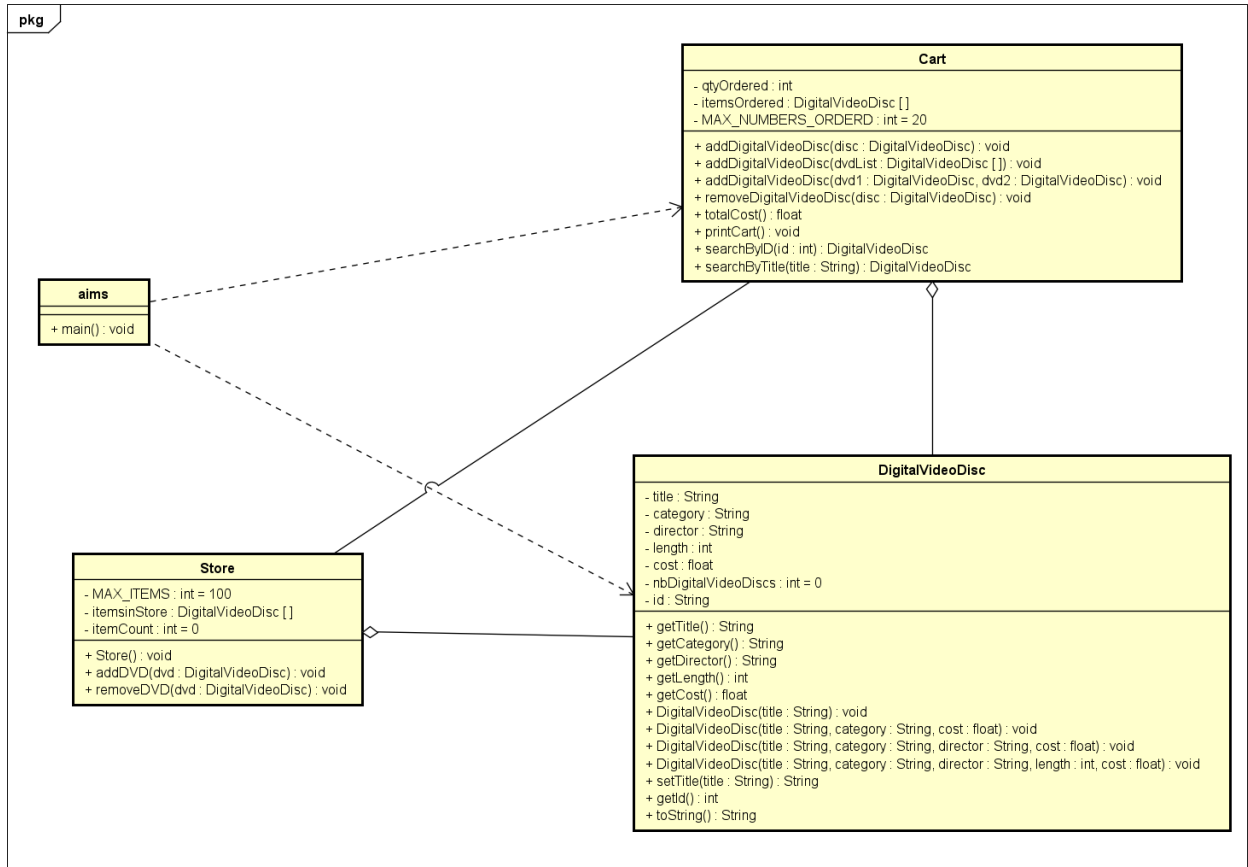


-Print the name of Jungle DVD:



3. Use-case diagram and Class diagram updates

a) Class diagram



b) Use-case diagram

The parts in red are the new additions, or are related to this lab assignment

- The JungleDVD's title changed because the line `dvd.setTitle(title)` changes the content of the object. Although `dvd` is just a copy of the reference, it still points to the same object in memory. So when we call the `setTitle()` method, we're changing the data inside that object.