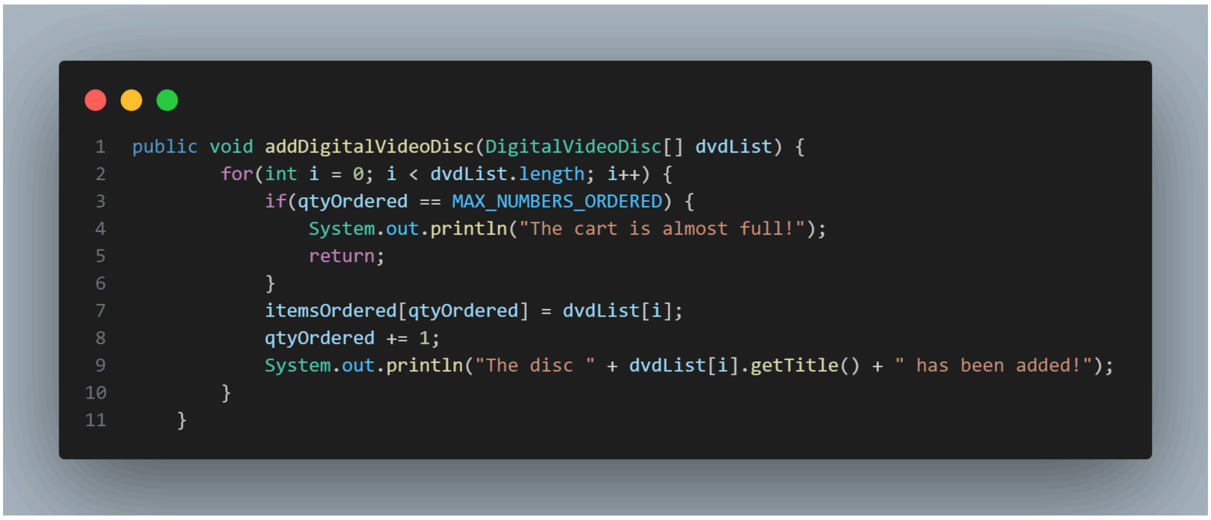
# Báo cáo thực hành Lập trình hướng đối tượng Lab 3
## Họ và tên: Nguyễn Lương Hoàng Tùng
## Lớp: 744520
## MSSV: 20226129

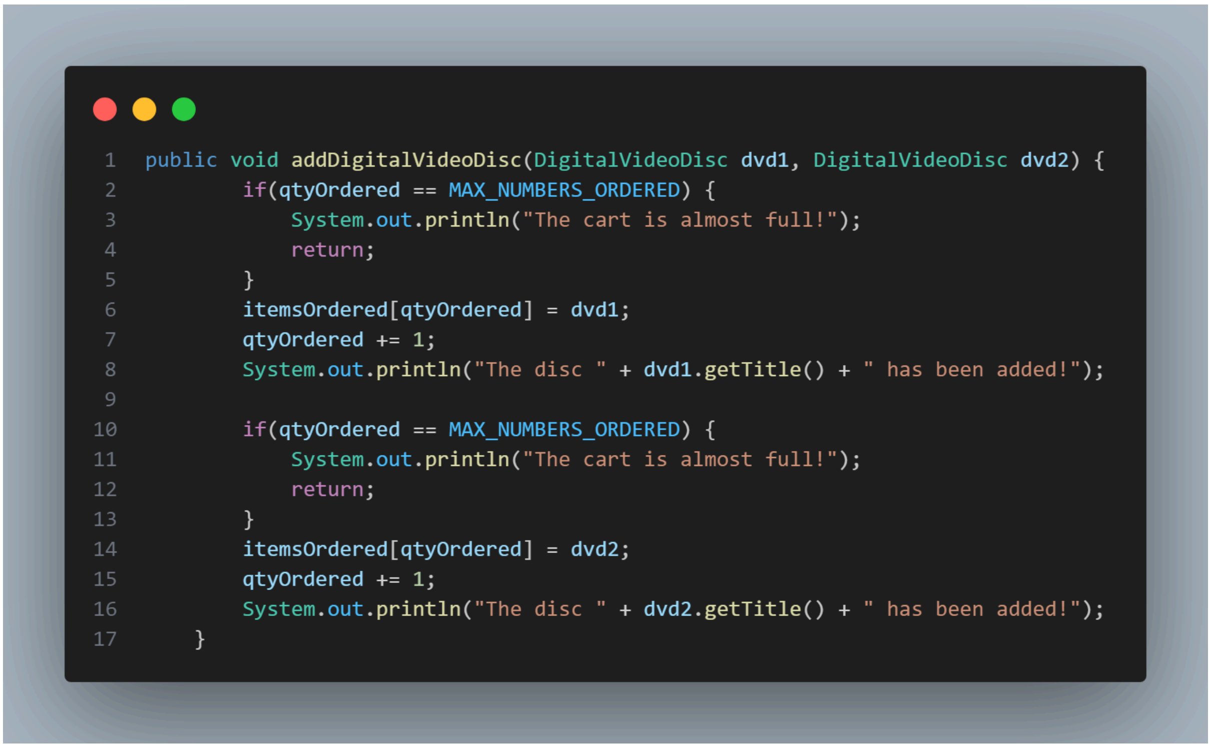
## I. Working with method overloading

### 1. Overloading by differing types of parameter



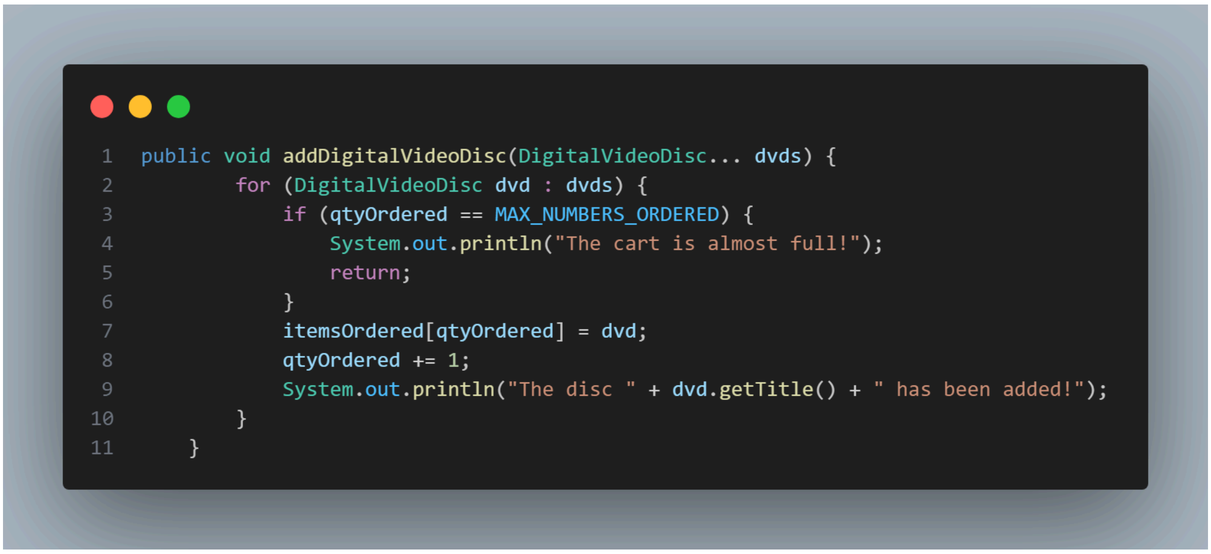
```java
public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList) {
        for(int i = 0; i < dvdList.length; i++) {
            if(qtyOrdered == MAX_NUMBERS_ORDERED) {
                System.out.println("The cart is almost full!");
                return;
            }
            itemsOrdered[qtyOrdered] = dvdList[i];
            qtyOrdered += 1;
            System.out.println("The disc " + dvdList[i].getTitle() + " has been added!");
        }
    }
```

### 2. Overloading by differing the number of parameters



```java
public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
        if(qtyOrdered == MAX_NUMBERS_ORDERED) {
            System.out.println("The cart is almost full!");
            return;
        }
        itemsOrdered[qtyOrdered] = dvd1;
        qtyOrdered += 1;
        System.out.println("The disc " + dvd1.getTitle() + " has been added!");

        if(qtyOrdered == MAX_NUMBERS_ORDERED) {
            System.out.println("The cart is almost full!");
            return;
        }
        itemsOrdered[qtyOrdered] = dvd2;
        qtyOrdered += 1;
        System.out.println("The disc " + dvd2.getTitle() + " has been added!");
    }
```

Try to add a method **addDigitalVideoDisc** which allows to pass an arbitrary number of arguments for dvd. Compare to an array parameter. What do you prefer in this case?



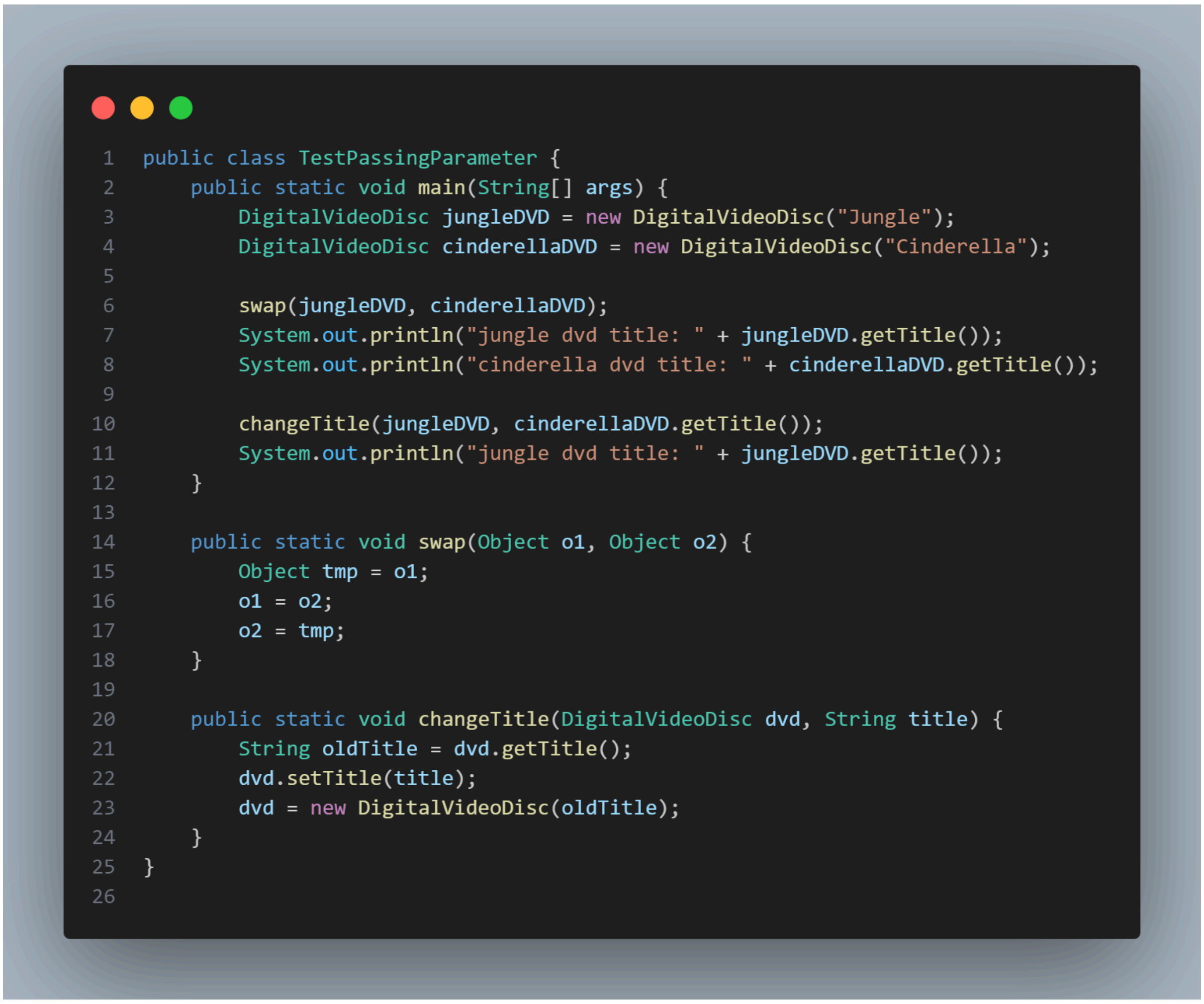
```
public void addDigitalVideoDisc(DigitalVideoDisc... dvds) {
        for (DigitalVideoDisc dvd : dvds) {
            if (qtyOrdered == MAX_NUMBERS_ORDERED) {
                System.out.println("The cart is almost full!");
                return;
            }
            itemsOrdered[qtyOrdered] = dvd;
            qtyOrdered += 1;
            System.out.println("The disc " + dvd.getTitle() + " has been added!");
        }
    }
```

Phương thức này sử dụng varargs để cho phép thêm một số lượng tùy ý các đĩa dvd vào giỏ hàng. Cách này linh hoạt hơn vì nó cho phép thêm bất kỳ số lượng đĩa dvd mà không cần chỉ định mảng một cách rõ ràng. Điều này làm cho mã nguồn sạch sẽ hơn và thuật tiện hơn khi gọi phương thức.

## II. Passing parameter

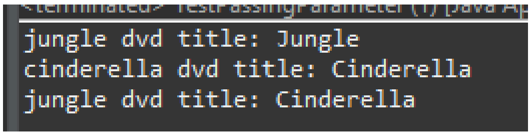
Question: Is JAVA a Pass by Value or a Pass by Reference programming language?
Answer: Java is a Pass by Value programming language. In Java, when you pass an argument to a method, you are passing a copy of the value. This means that changes made to the parameter inside the method do not affect the original value.

1. Create class TestPassingParameter



```java
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);
    }
}

```

Result:



```
<terminated> TestPassingParameter (1) [Java Ap
jungle dvd title: Jungle
cinderella dvd title: Cinderella
jungle dvd title: Cinderella
```

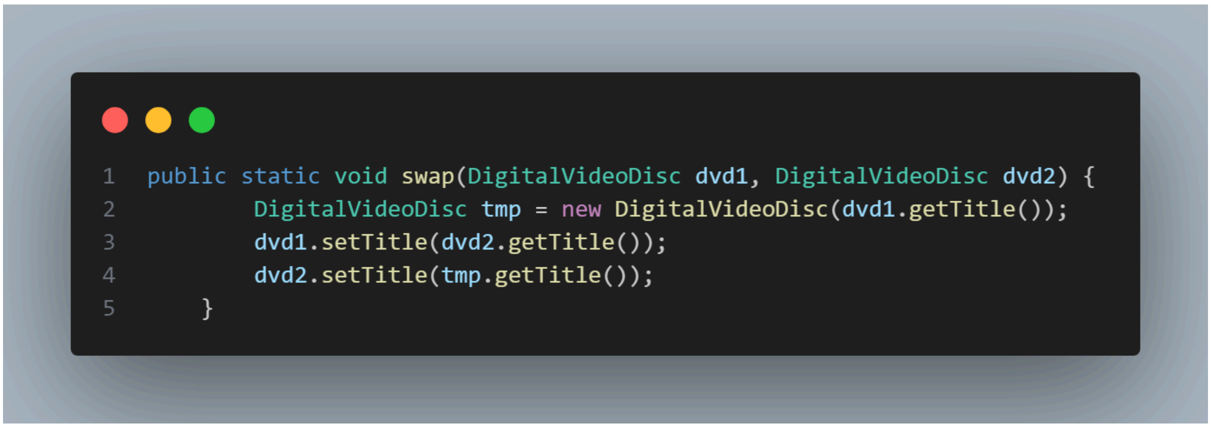
After the call of swap(jungleDVD, cinderellaDVD) why does the title of these two objects still remain?

-> Vì trong Java, tham số được truyền vào phương thức là giá trị của đối tượng, không phải là tham chiếu đến đối tượng. Khi ta thay đổi giá trị của tham số bên trong phương thức, sự thay đổi này không ảnh hưởng đến giá trị của các đối tượng gốc.

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

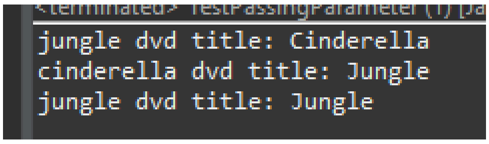
-> Vì trong phương thức changeTitle, ta thực hiện thay đổi trực tiếp trên đối tượng dvd bằng cách gọi dvd.setTitle(title). Điều này ảnh hưởng trực tiếp đến đối tượng gốc được truyền vào phương thức.

2. Fix swap



```java
public static void swap(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
        DigitalVideoDisc tmp = new DigitalVideoDisc(dvd1.getTitle());
        dvd1.setTitle(dvd2.getTitle());
        dvd2.setTitle(tmp.getTitle());
    }
```

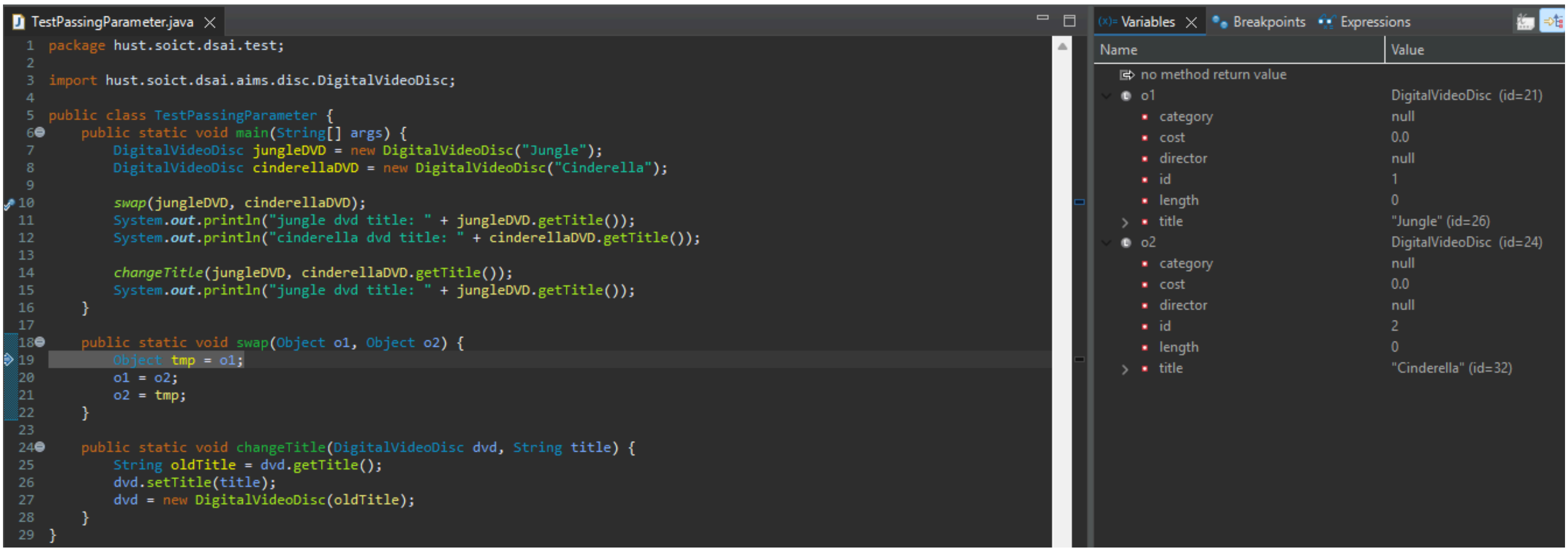
Result:



```
<terminated> TestPassingParameter (1) [Ja
jungle dvd title: Cinderella
cinderella dvd title: Jungle
jungle dvd title: Jungle
```

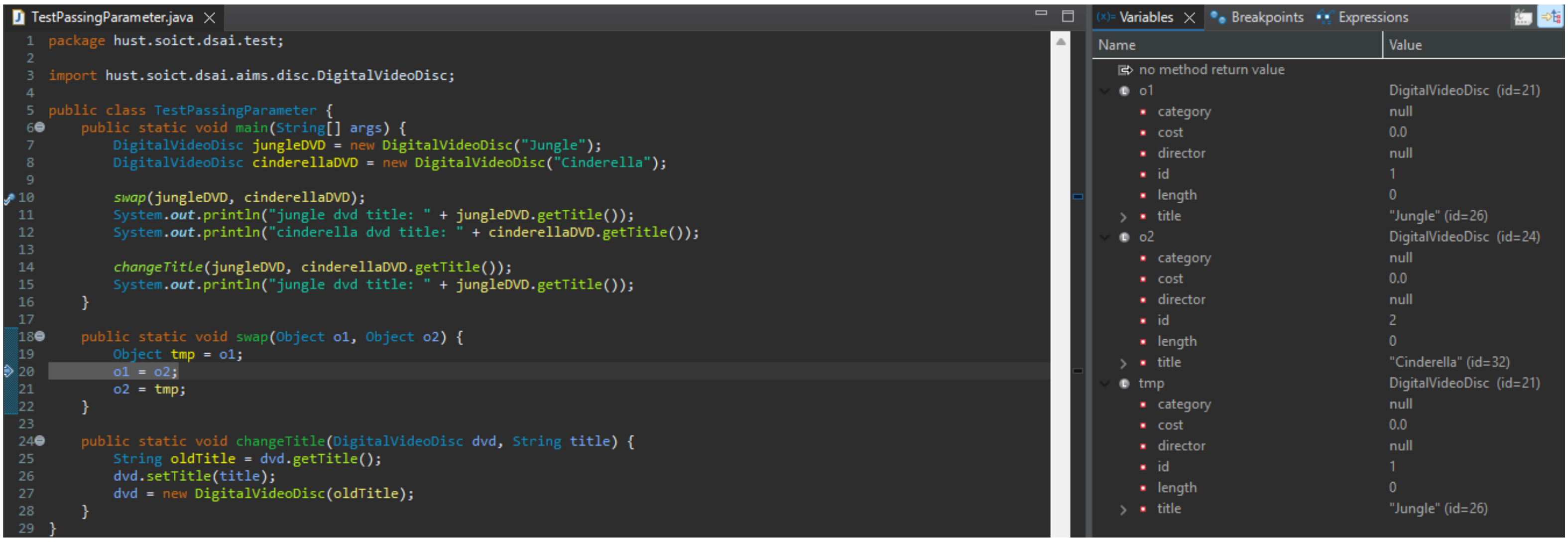
## III. Use debug run

1. Investigate value of variables



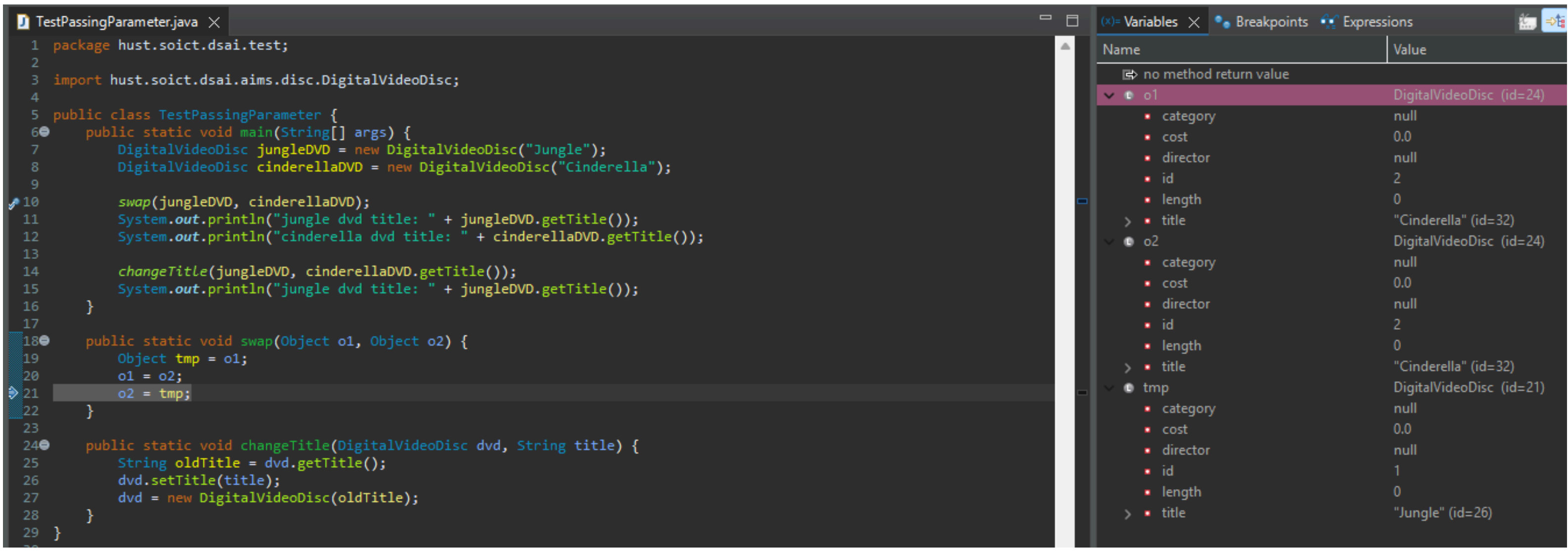
```java
package hust.soict.dsai.test;

import hust.soict.dsai.aims.disc.DigitalVideoDisc;

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);
    }
}
```

| Name | Value |
|---|---|
| no method return value | |
| o1 | DigitalVideoDisc (id=21) |
| category | null |
| cost | 0.0 |
| director | null |
| id | 1 |
| length | 0 |
| title | "Jungle" (id=26) |
| o2 | DigitalVideoDisc (id=24) |
| category | null |
| cost | 0.0 |
| director | null |
| id | 2 |
| length | 0 |
| title | "Cinderella" (id=32) |



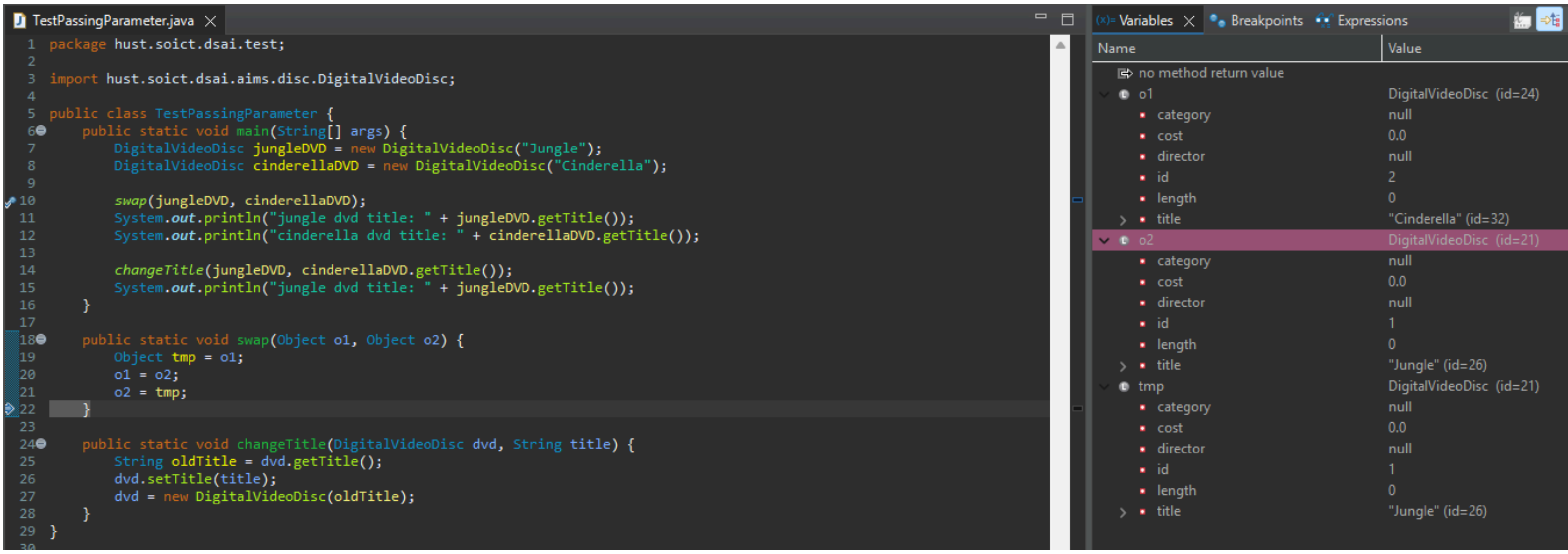
```java
package hust.soict.dsai.test;

import hust.soict.dsai.aims.disc.DigitalVideoDisc;

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);
    }
}
```

| Name | Value |
|---|---|
| no method return value | |
| o1 | DigitalVideoDisc (id=21) |
| category | null |
| cost | 0.0 |
| director | null |
| id | 1 |
| length | 0 |
| title | "Jungle" (id=26) |
| o2 | DigitalVideoDisc (id=24) |
| category | null |
| cost | 0.0 |
| director | null |
| id | 2 |
| length | 0 |
| title | "Cinderella" (id=32) |
| tmp | DigitalVideoDisc (id=21) |
| category | null |
| cost | 0.0 |
| director | null |
| id | 1 |
| length | 0 |
| title | "Jungle" (id=26) |



```java
package hust.soict.dsai.test;

import hust.soict.dsai.aims.disc.DigitalVideoDisc;

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);
    }
}
```

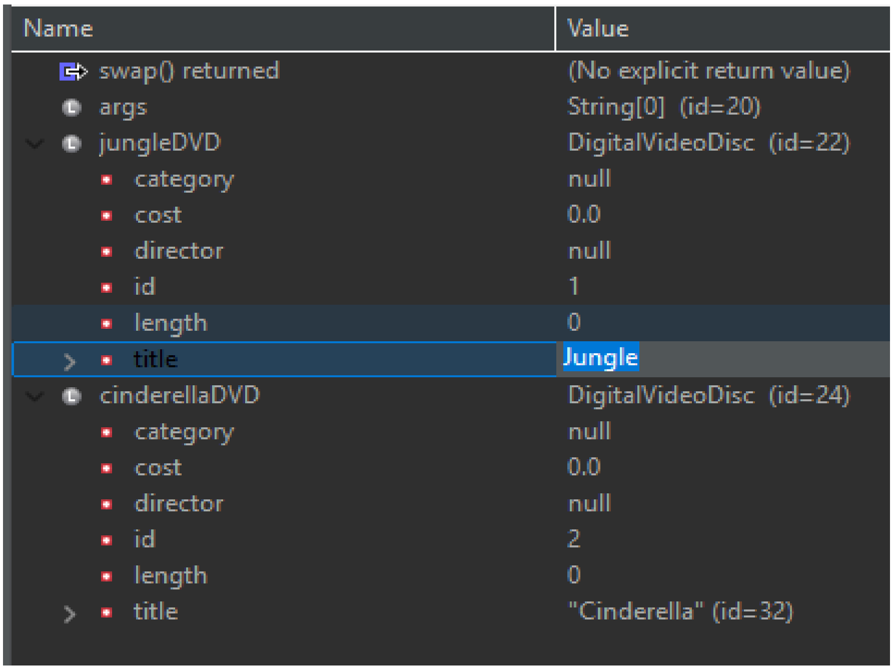
```
Name                      Value
  no method return value
o1                        DigitalVideoDisc  (id=24)
    category              null
    cost                  0.0
    director              null
    id                    2
    length                0
    title                 "Cinderella" (id=32)
o2                        DigitalVideoDisc  (id=24)
    category              null
    cost                  0.0
    director              null
    id                    2
    length                0
    title                 "Cinderella" (id=32)
tmp                       DigitalVideoDisc  (id=21)
    category              null
    cost                  0.0
    director              null
    id                    1
    length                0
    title                 "Jungle" (id=26)
```



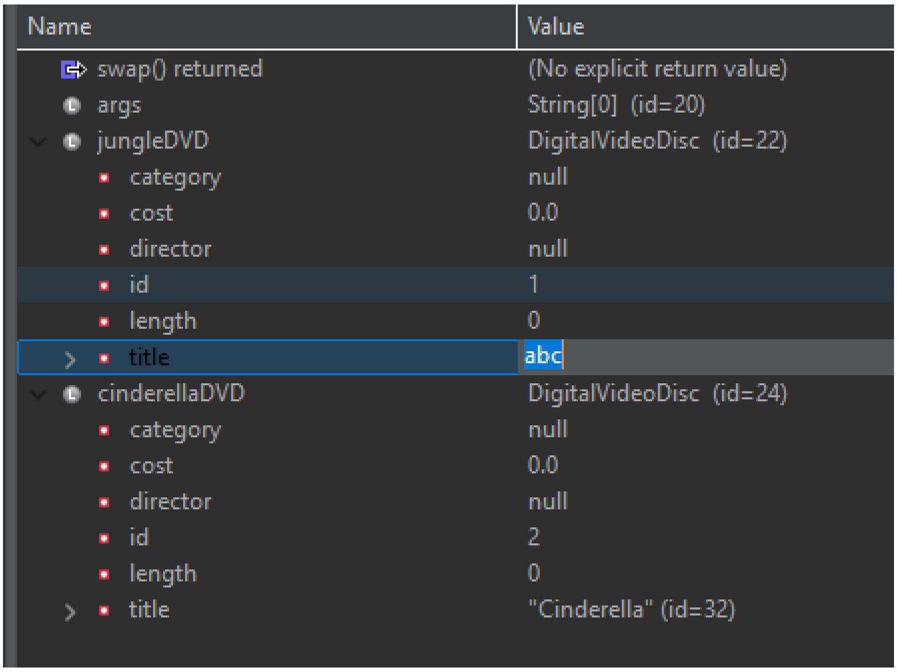
```java
package hust.soict.dsai.test;

import hust.soict.dsai.aims.disc.DigitalVideoDisc;

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);
    }
}
```

```
Name                      Value
  no method return value
o1                        DigitalVideoDisc  (id=24)
    category              null
    cost                  0.0
    director              null
    id                    2
    length                0
    title                 "Cinderella" (id=32)
o2                        DigitalVideoDisc  (id=21)
    category              null
    cost                  0.0
    director              null
    id                    1
    length                0
    title                 "Jungle" (id=26)
tmp                       DigitalVideoDisc  (id=21)
    category              null
    cost                  0.0
    director              null
    id                    1
    length                0
    title                 "Jungle" (id=26)
```

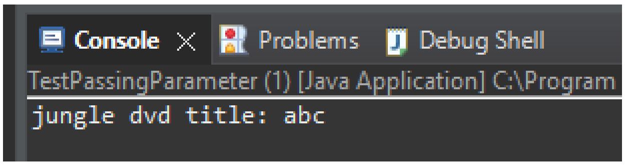
## 2. Change value of variables



| Name | Value |
|---|---|
| swap() returned | (No explicit return value) |
| args | String[0] (id=20) |
| jungleDVD | DigitalVideoDisc (id=22) |
| category | null |
| cost | 0.0 |
| director | null |
| id | 1 |
| length | 0 |
| title | Jungle |
| cinderellaDVD | DigitalVideoDisc (id=24) |
| category | null |
| cost | 0.0 |
| director | null |
| id | 2 |
| length | 0 |
| title | "Cinderella" (id=32) |



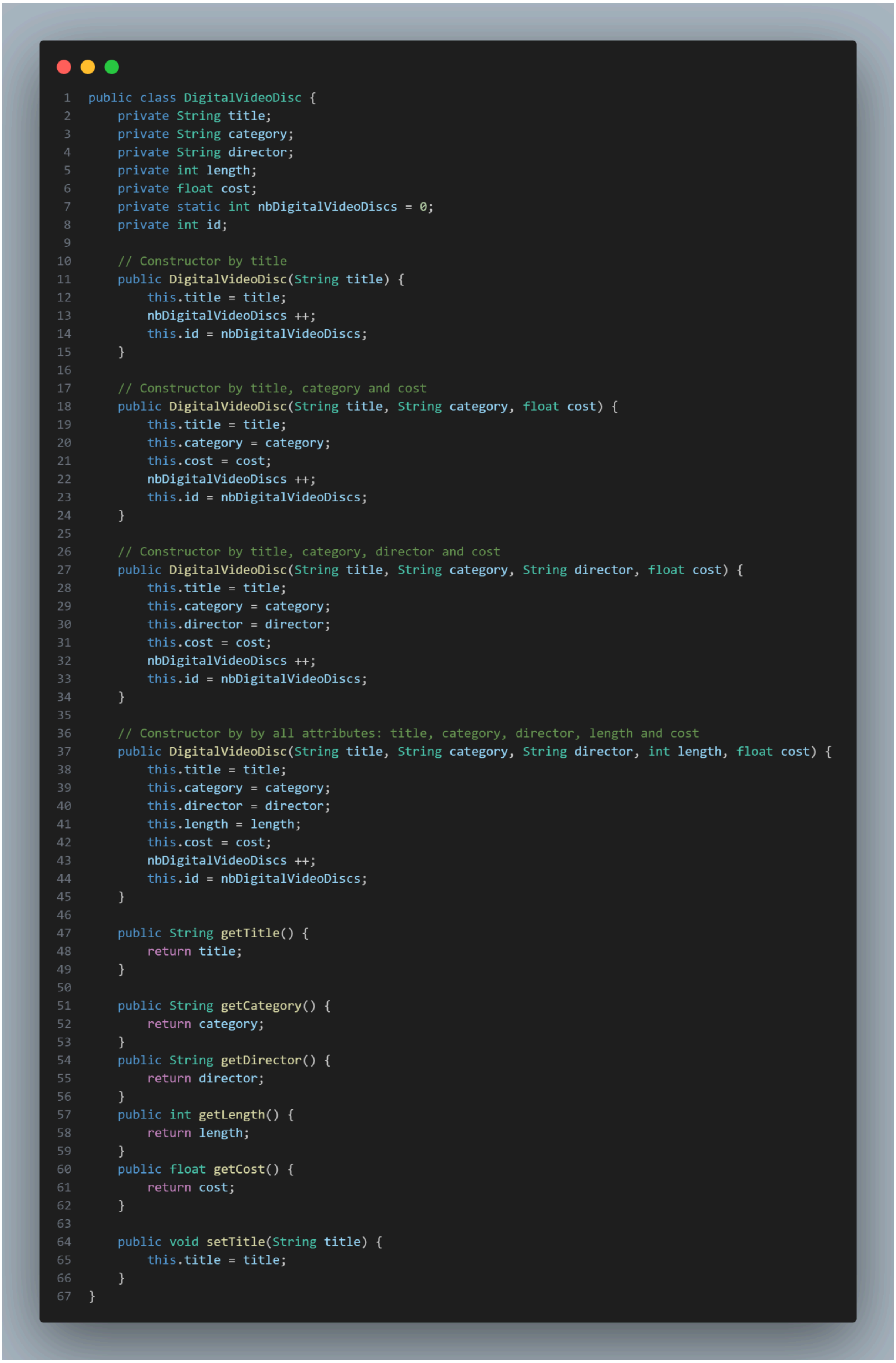
| Name | Value |
|---|---|
| swap() returned | (No explicit return value) |
| args | String[0] (id=20) |
| jungleDVD | DigitalVideoDisc (id=22) |
| category | null |
| cost | 0.0 |
| director | null |
| id | 1 |
| length | 0 |
| title | abc |
| cinderellaDVD | DigitalVideoDisc (id=24) |
| category | null |
| cost | 0.0 |
| director | null |
| id | 2 |
| length | 0 |
| title | "Cinderella" (id=32) |



TestPassingParameter (1) [Java Application] C:\Program

```
jungle dvd title: abc
```

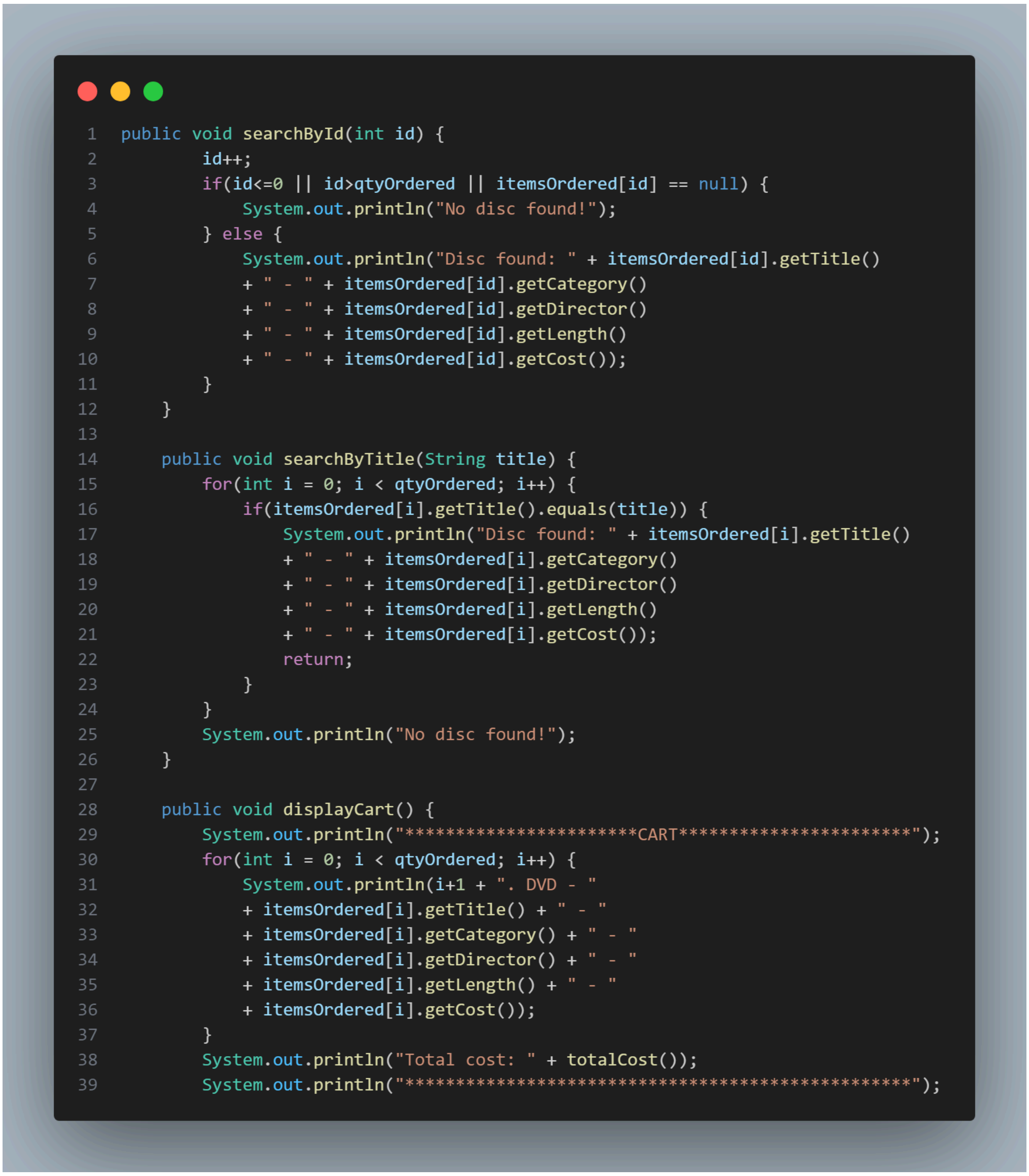
## IV. Classifier Member and Instance Member



```java
public class DigitalVideoDisc {
    private String title;
    private String category;
    private String director;
    private int length;
    private float cost;
    private static int nbDigitalVideoDiscs = 0;
    private int id;

    // Constructor by title
    public DigitalVideoDisc(String title) {
        this.title = title;
        nbDigitalVideoDiscs ++;
        this.id = nbDigitalVideoDiscs;
    }

    // Constructor by title, category and cost
    public DigitalVideoDisc(String title, String category, float cost) {
        this.title = title;
        this.category = category;
        this.cost = cost;
        nbDigitalVideoDiscs ++;
        this.id = nbDigitalVideoDiscs;
    }

    // Constructor by title, category, director and cost
    public DigitalVideoDisc(String title, String category, String director, float cost) {
        this.title = title;
        this.category = category;
        this.director = director;
        this.cost = cost;
        nbDigitalVideoDiscs ++;
        this.id = nbDigitalVideoDiscs;
    }

    // Constructor by by all attributes: title, category, director, length and cost
    public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
        this.title = title;
        this.category = category;
        this.director = director;
        this.length = length;
        this.cost = cost;
        nbDigitalVideoDiscs ++;
        this.id = nbDigitalVideoDiscs;
    }

    public String getTitle() {
        return title;
    }

    public String getCategory() {
        return category;
    }
    public String getDirector() {
        return director;
    }
    public int getLength() {
        return length;
    }
    public float getCost() {
        return cost;
    }

    public void setTitle(String title) {
        this.title = title;
    }
}
```

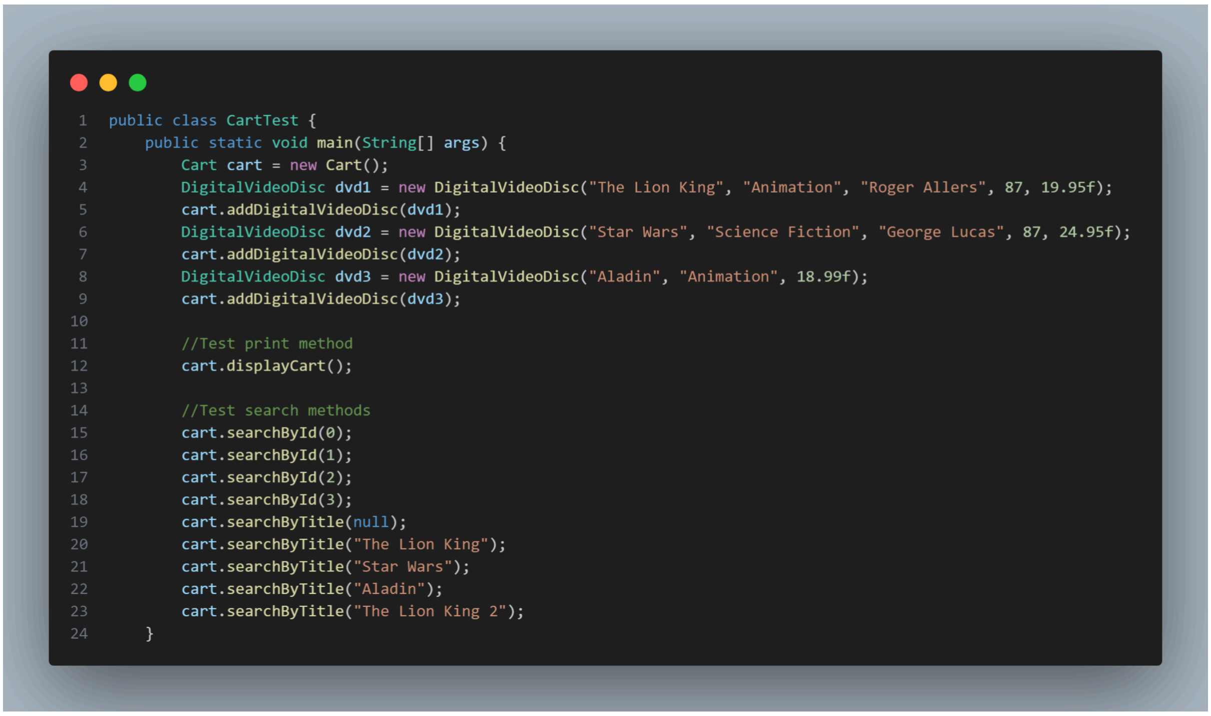
## V. Open the Cart class

1. Create print and search methods



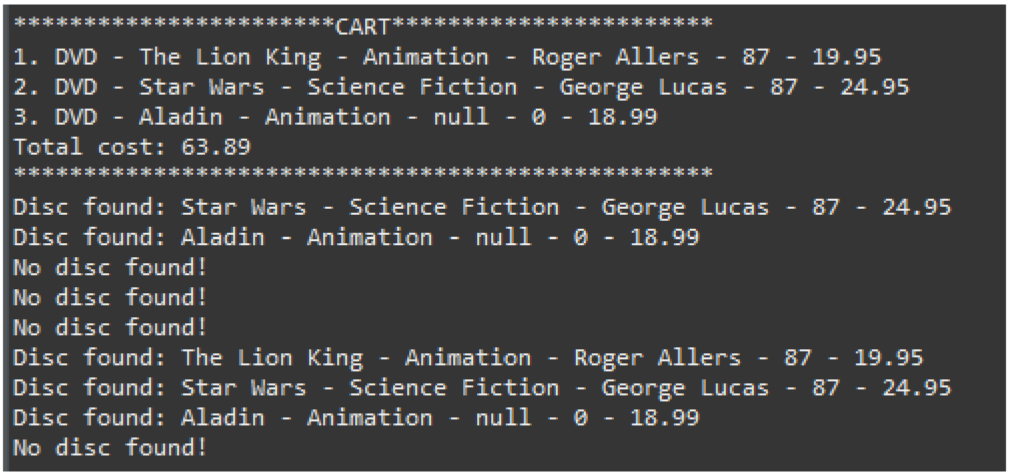
```java
public void searchById(int id) {
        id++;
        if(id<=0 || id>qtyOrdered || itemsOrdered[id] == null) {
            System.out.println("No disc found!");
        } else {
            System.out.println("Disc found: " + itemsOrdered[id].getTitle()
            + " - " + itemsOrdered[id].getCategory()
            + " - " + itemsOrdered[id].getDirector()
            + " - " + itemsOrdered[id].getLength()
            + " - " + itemsOrdered[id].getCost());
        }
    }

    public void searchByTitle(String title) {
        for(int i = 0; i < qtyOrdered; i++) {
            if(itemsOrdered[i].getTitle().equals(title)) {
                System.out.println("Disc found: " + itemsOrdered[i].getTitle()
                + " - " + itemsOrdered[i].getCategory()
                + " - " + itemsOrdered[i].getDirector()
                + " - " + itemsOrdered[i].getLength()
                + " - " + itemsOrdered[i].getCost());
                return;
            }
        }
        System.out.println("No disc found!");
    }

    public void displayCart() {
        System.out.println("*********************CART*********************");
        for(int i = 0; i < qtyOrdered; i++) {
            System.out.println(i+1 + ". DVD - "
            + itemsOrdered[i].getTitle() + " - "
            + itemsOrdered[i].getCategory() + " - "
            + itemsOrdered[i].getDirector() + " - "
            + itemsOrdered[i].getLength() + " - "
            + itemsOrdered[i].getCost());
        }
        System.out.println("Total cost: " + totalCost());
        System.out.println("**********************************************");
```

2. Create CartTest class



```java
public class CartTest {
    public static void main(String[] args) {
        Cart cart = new Cart();
        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
        cart.addDigitalVideoDisc(dvd1);
        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);
        cart.addDigitalVideoDisc(dvd2);
        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
        cart.addDigitalVideoDisc(dvd3);

        //Test print method
        cart.displayCart();

        //Test search methods
        cart.searchById(0);
        cart.searchById(1);
        cart.searchById(2);
        cart.searchById(3);
        cart.searchByTitle(null);
        cart.searchByTitle("The Lion King");
        cart.searchByTitle("Star Wars");
        cart.searchByTitle("Aladin");
        cart.searchByTitle("The Lion King 2");
    }
```

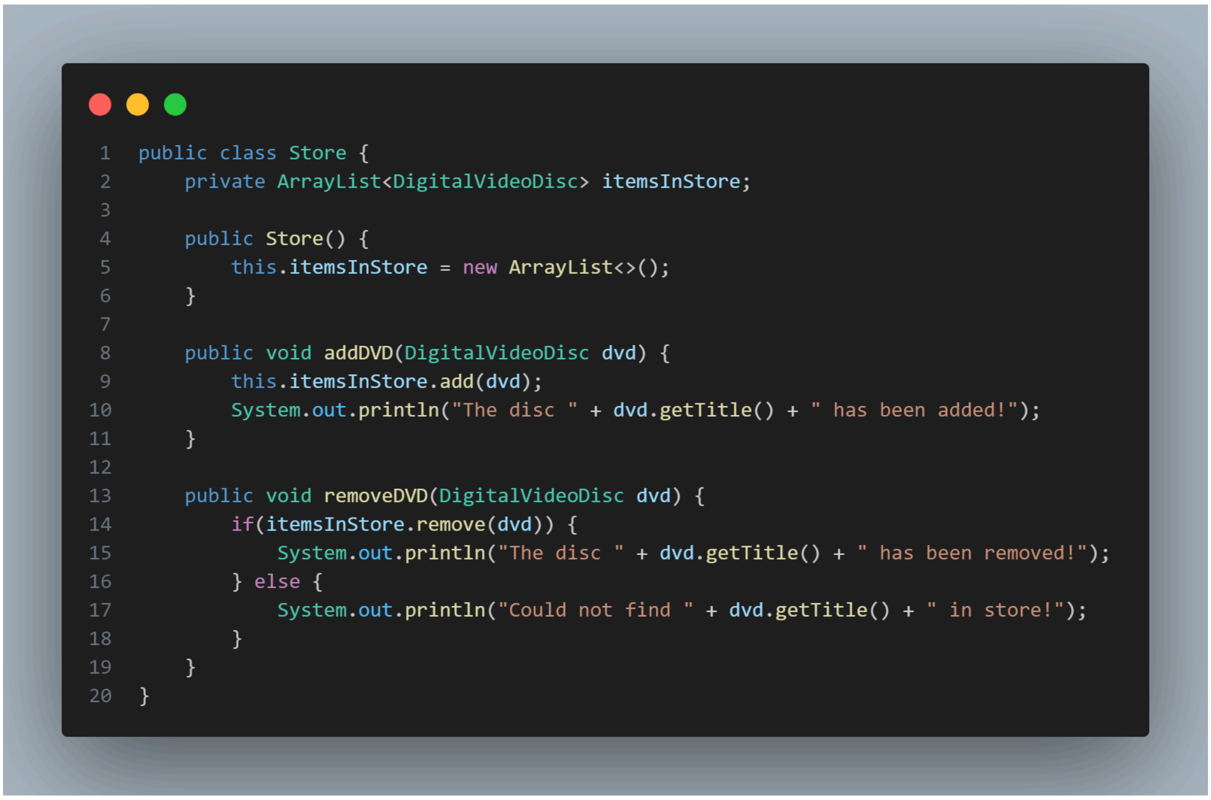
Result:



```
***********************CART***********************
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
**************************************************
Disc found: Star Wars - Science Fiction - George Lucas - 87 - 24.95
Disc found: Aladin - Animation - null - 0 - 18.99
No disc found!
No disc found!
No disc found!
Disc found: The Lion King - Animation - Roger Allers - 87 - 19.95
Disc found: Star Wars - Science Fiction - George Lucas - 87 - 24.95
Disc found: Aladin - Animation - null - 0 - 18.99
No disc found!
```

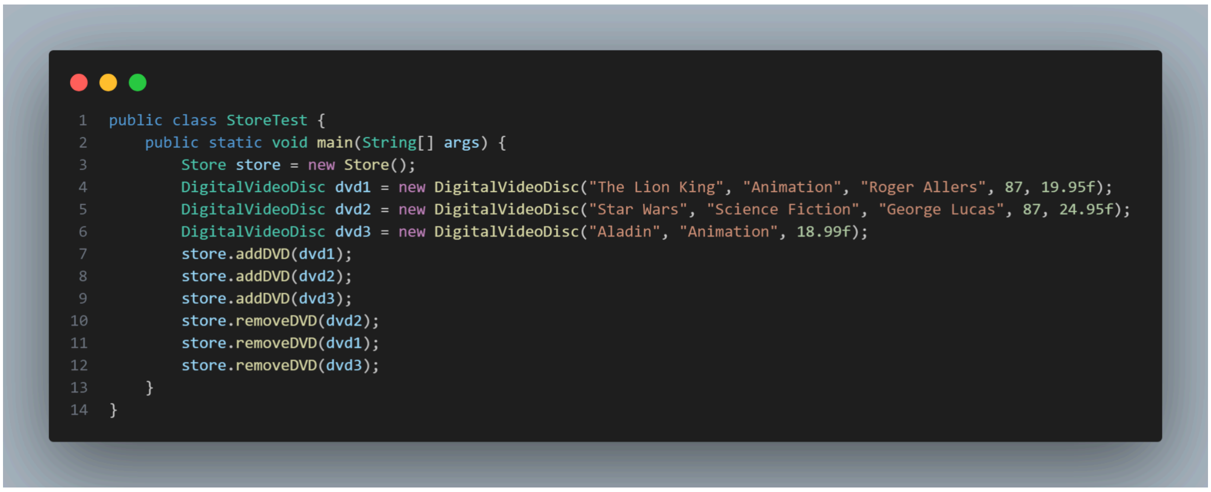
# VI. Implement the Store class

1. Create Store class



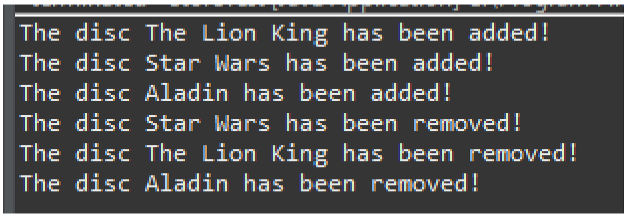
```java
public class Store {
    private ArrayList<DigitalVideoDisc> itemsInStore;

    public Store() {
        this.itemsInStore = new ArrayList<>();
    }

    public void addDVD(DigitalVideoDisc dvd) {
        this.itemsInStore.add(dvd);
        System.out.println("The disc " + dvd.getTitle() + " has been added!");
    }

    public void removeDVD(DigitalVideoDisc dvd) {
        if(itemsInStore.remove(dvd)) {
            System.out.println("The disc " + dvd.getTitle() + " has been removed!");
        } else {
            System.out.println("Could not find " + dvd.getTitle() + " in store!");
        }
    }
}
```

2. Create StoreTest class



```java
public class StoreTest {
    public static void main(String[] args) {
        Store store = new Store();
        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);
        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
        store.addDVD(dvd1);
        store.addDVD(dvd2);
        store.addDVD(dvd3);
        store.removeDVD(dvd2);
        store.removeDVD(dvd1);
        store.removeDVD(dvd3);
    }
}
```

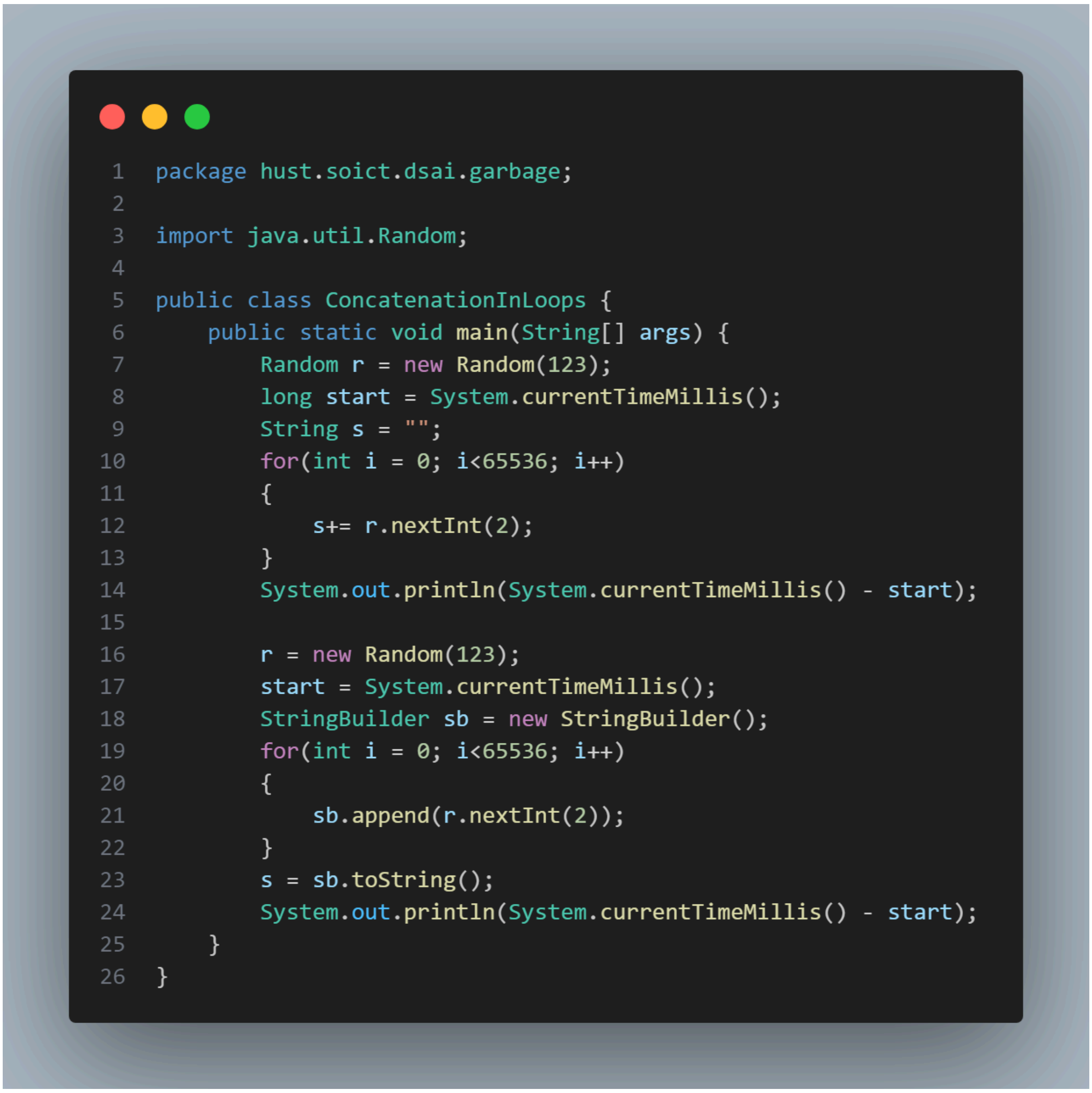
Result:



```
The disc The Lion King has been added!
The disc Star Wars has been added!
The disc Aladin has been added!
The disc Star Wars has been removed!
The disc The Lion King has been removed!
The disc Aladin has been removed!
```

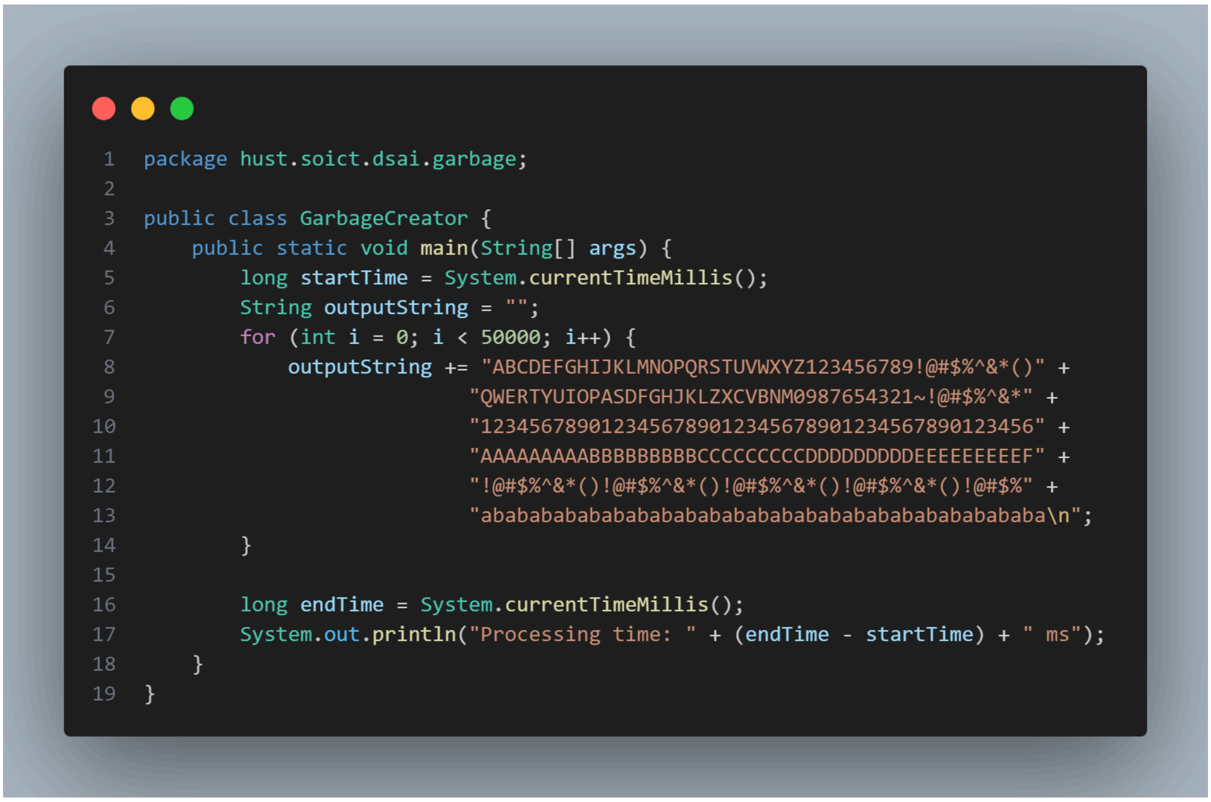
# VII. String, StringBuilder and StringBuffer

1. Create ConcatenationInLoops class



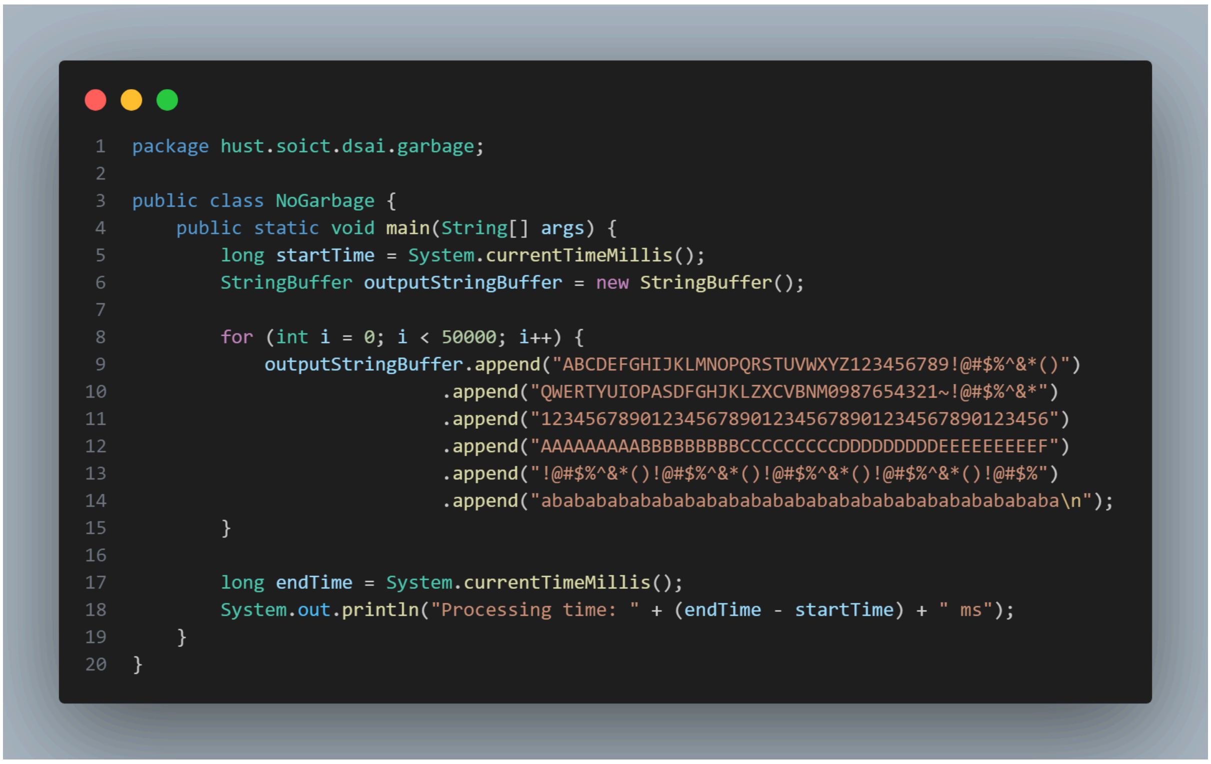
```java
package hust.soict.dsai.garbage;

import java.util.Random;

public class ConcatenationInLoops {
    public static void main(String[] args) {
        Random r = new Random(123);
        long start = System.currentTimeMillis();
        String s = "";
        for(int i = 0; i<65536; i++)
        {
            s+= r.nextInt(2);
        }
        System.out.println(System.currentTimeMillis() - start);

        r = new Random(123);
        start = System.currentTimeMillis();
        StringBuilder sb = new StringBuilder();
        for(int i = 0; i<65536; i++)
        {
            sb.append(r.nextInt(2));
        }
        s = sb.toString();
        System.out.println(System.currentTimeMillis() - start);
    }
}
```

2. Create GarbageCreator class



```java
package hust.soict.dsai.garbage;

public class GarbageCreator {
    public static void main(String[] args) {
        long startTime = System.currentTimeMillis();
        String outputString = "";
        for (int i = 0; i < 50000; i++) {
            outputString += "ABCDEFGHIJKLMNOPQRSTUVWXYZ123456789!@#$%^&*()" +
                           "QWERTYUIOPASDFGHJKLZXCVBNM0987654321~!@#$%^&*" +
                           "1234567890123456789012345678901234567890123456" +
                           "AAAAAAAAABBBBBBBBBCCCCCCCCCDDDDDDDDDEEEEEEEEEF" +
                           "!@#$%^&*()!@#$%^&*()!@#$%^&*()!@#$%^&*()!@#$%" +
                           "abababababababababababababababababababababababa\n";
        }

        long endTime = System.currentTimeMillis();
        System.out.println("Processing time: " + (endTime - startTime) + " ms");
    }
}
```

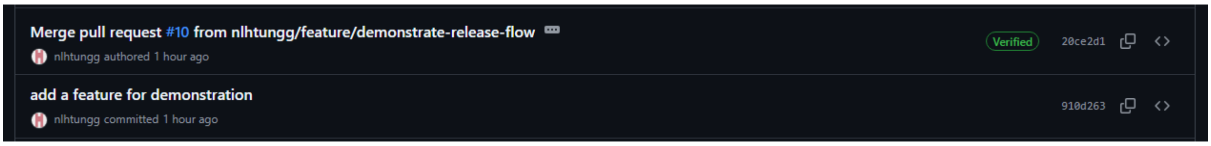
3. Create NoGarbage



```java
package hust.soict.dsai.garbage;

public class NoGarbage {
    public static void main(String[] args) {
        long startTime = System.currentTimeMillis();
        StringBuffer outputStringBuffer = new StringBuffer();

        for (int i = 0; i < 50000; i++) {
            outputStringBuffer.append("ABCDEFGHIJKLMNOPQRSTUVWXYZ123456789!@#$%^&*()")
                            .append("QWERTYUIOPASDFGHJKLZXCVBNM0987654321~!@#$%^&*")
                            .append("1234567890123456789012345678901234567890123456")
                            .append("AAAAAAAAABBBBBBBBBCCCCCCCCCDDDDDDDDDEEEEEEEEEF")
                            .append("!@#$%^&*()!@#$%^&*()!@#$%^&*()!@#$%^&*()!@#$%")
                            .append("abababababababababababababababababababababababa\n");
        }

        long endTime = System.currentTimeMillis();
        System.out.println("Processing time: " + (endTime - startTime) + " ms");
    }
}
```

## VIII. Release flow demonstration

Đã thực hiện thành công



Merge pull request #10 from nlhtungg/feature/demonstrate-release-flow

nlhtungg authored 1 hour ago

20ce2d1

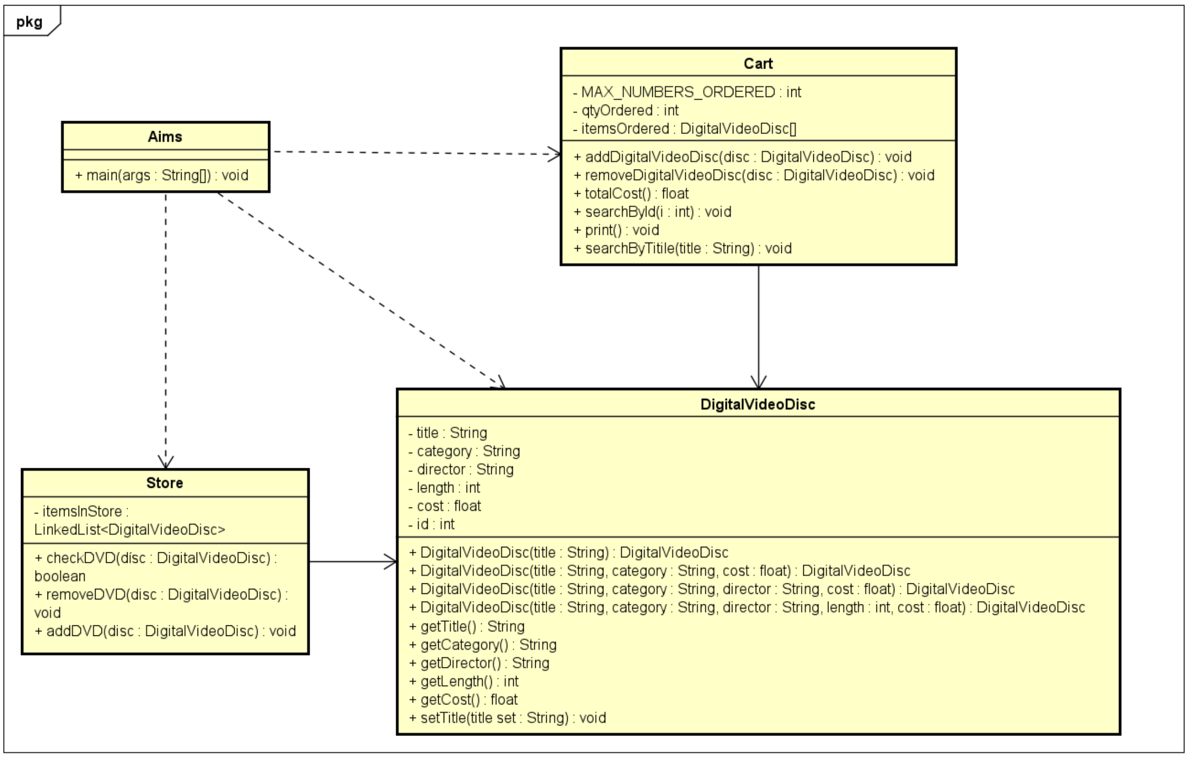
add a feature for demonstration

nlhtungg committed 1 hour ago

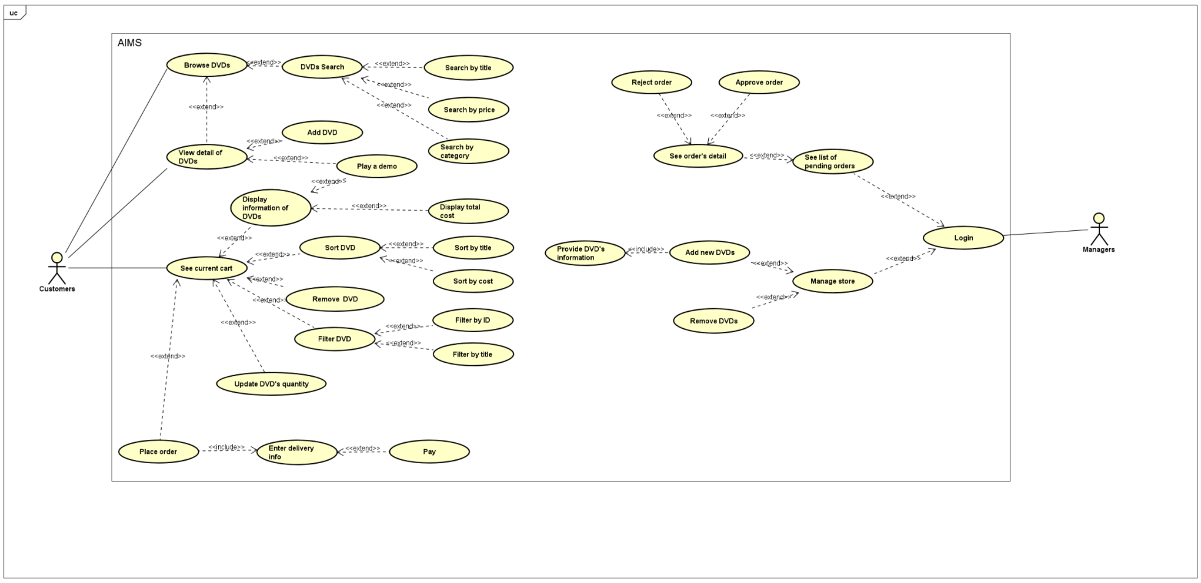
910d263

# IX. Update Diagrams

## 1. Class Diagram


pkg
Aims
+ main(args : String[]) : void
Cart
- MAX_NUMBERS_ORDERED : int
- qtyOrdered : int
- itemsOrdered : DigitalVideoDisc[]
+ addDigitalVideoDisc(disc : DigitalVideoDisc) : void
+ removeDigitalVideoDisc(disc : DigitalVideoDisc) : void
+ totalCost() : float
+ searchById(i : int) : void
+ print() : void
+ searchByTitile(title : String) : void
Store
- itemsInStore : LinkedList<DigitalVideoDisc>
+ checkDVD(disc : DigitalVideoDisc) : boolean
+ removeDVD(disc : DigitalVideoDisc) : void
+ addDVD(disc : DigitalVideoDisc) : void
DigitalVideoDisc
- title : String
- category : String
- director : String
- length : int
- cost : float
- id : int
+ DigitalVideoDisc(title : String) : DigitalVideoDisc
+ DigitalVideoDisc(title : String, category : String, cost : float) : DigitalVideoDisc
+ DigitalVideoDisc(title : String, category : String, director : String, cost : float) : DigitalVideoDisc
+ DigitalVideoDisc(title : String, category : String, director : String, length : int, cost : float) : DigitalVideoDisc
+ getTitle() : String
+ getCategory() : String
+ getDirector() : String
+ getLength() : int
+ getCost() : float
+ setTitle(title set : String) : void

## 2. Use Case Diagram


uc
AIMS
Browse DVDs
<<extend>>
DVDs Search
<<extend>>
Search by title
<<extend>>
Search by price
<<extend>>
Search by category
<<extend>>
Add DVD
<<extend>>
View detail of DVDs
<<extend>>
Play a demo
<<extend>>
Display information of DVDs
<<extend>>
Display total cost
<<extend>>
Sort DVD
<<extend>>
Sort by title
<<extend>>
Sort by cost
Customers
See current cart
<<extend>>
<<extend>>
Remove DVD
<<extend>>
<<extend>>
Filter DVD
<<extend>>
Filter by ID
<<extend>>
Filter by title
Update DVD's quantity
<<extend>>
Place order
<<include>>
Enter delivery info
<<extend>>
Pay
Reject order
Approve order
<<extend>>
<<extend>>
See order's detail
<<extend>>
See list of pending orders
<<extend>>
Login
Managers
Provide DVD's information
<<include>>
Add new DVDs
<<extend>>
Manage store
<<extend>>
<<extend>>
Remove DVDs