Report OOP – Lab03

Họ và tên: Nguyễn Kiều Duyên

MSSV: 20225618

Mã lớp: 744520 **Kỳ 2024-1**

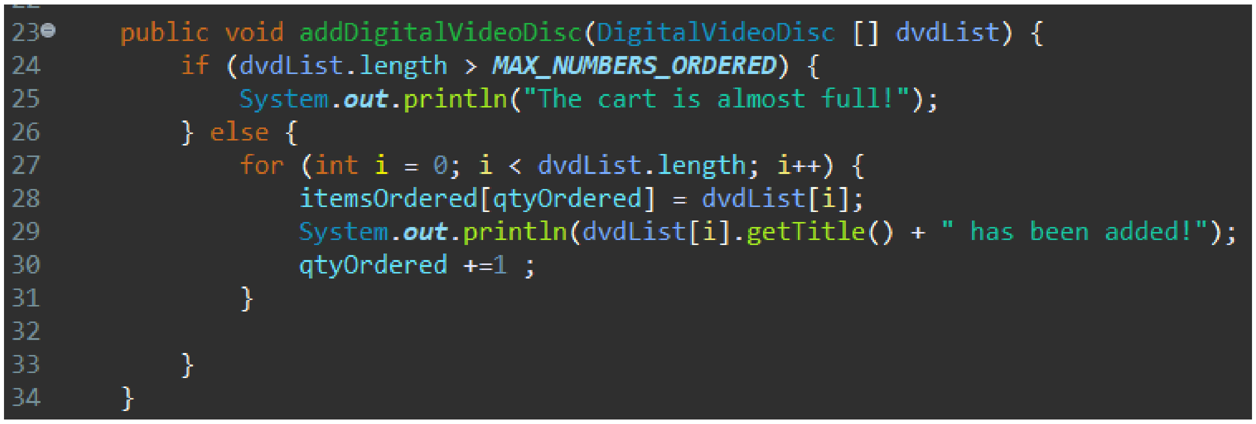
Table of Contents

I. New written code

1. Working with method overloading

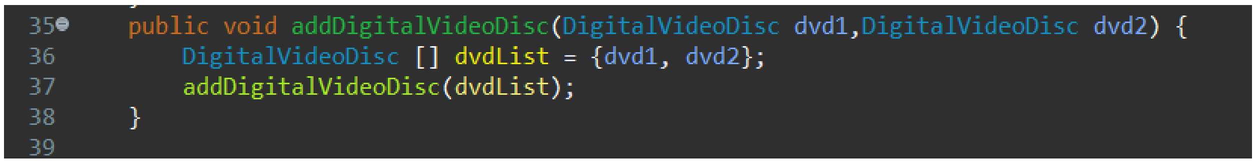
Tại class **Cart.java**: nạp chồng phương thức **addDigitalVideoDisc.**

1.1 Overloading by differing types of parameter



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

    }
}
```

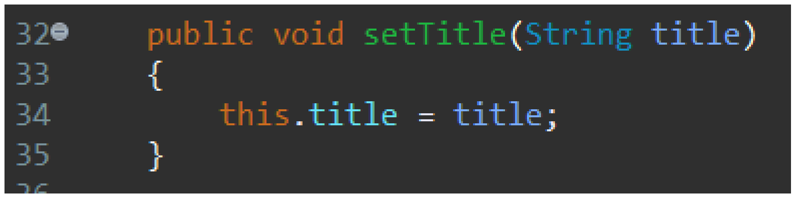
1.2 Overloading by differing the number of parameters



```java
public void addDigitalVideoDisc(DigitalVideoDisc dvd1,DigitalVideoDisc dvd2) {
    DigitalVideoDisc [] dvdList = {dvd1, dvd2};
    addDigitalVideoDisc(dvdList);
}
```

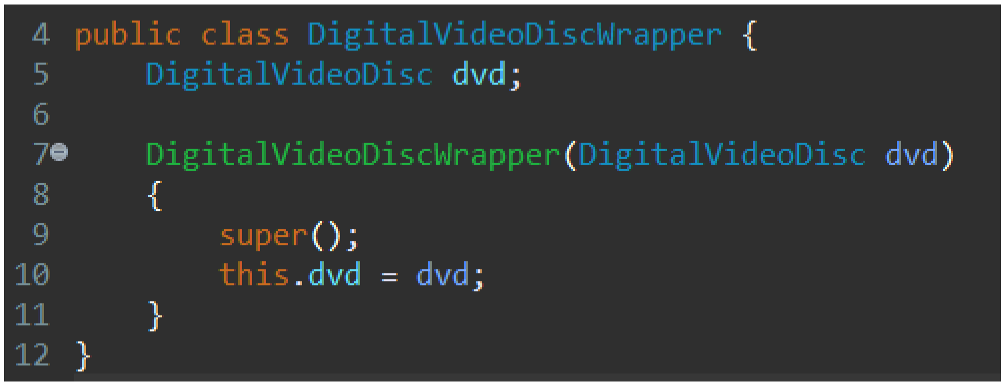
2. Passing parameter

- Thêm setter cho title của lớp **DigitalVideoDisc**.

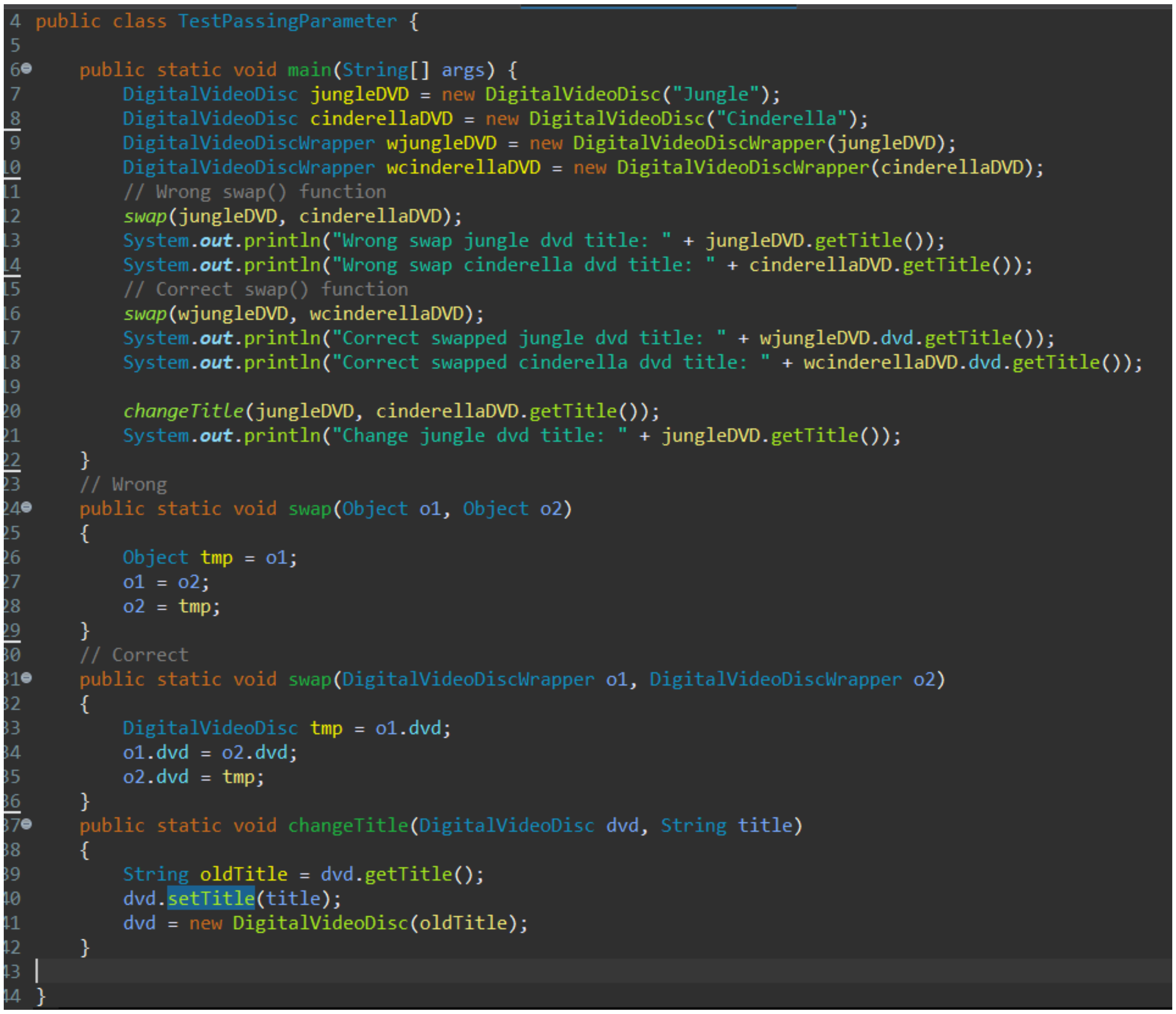


```java
public void setTitle(String title)
{
    this.title = title;
}
```

- Tạo lớp **DigitalVideoDiscWrapper** để wrap lên class **DigitalVideoDisc** từ đó có thể thay đổi được giá trị qua hàm swap đúng ở class **TestPassingParameter** như dưới đây



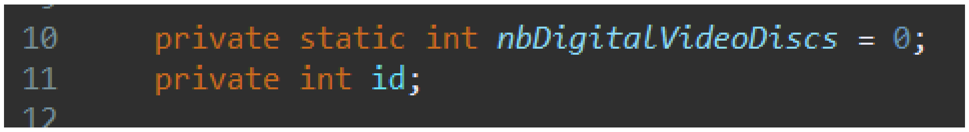
```java
public class DigitalVideoDiscWrapper {
    DigitalVideoDisc dvd;

    DigitalVideoDiscWrapper(DigitalVideoDisc dvd)
    {
        super();
        this.dvd = dvd;
    }
}
```



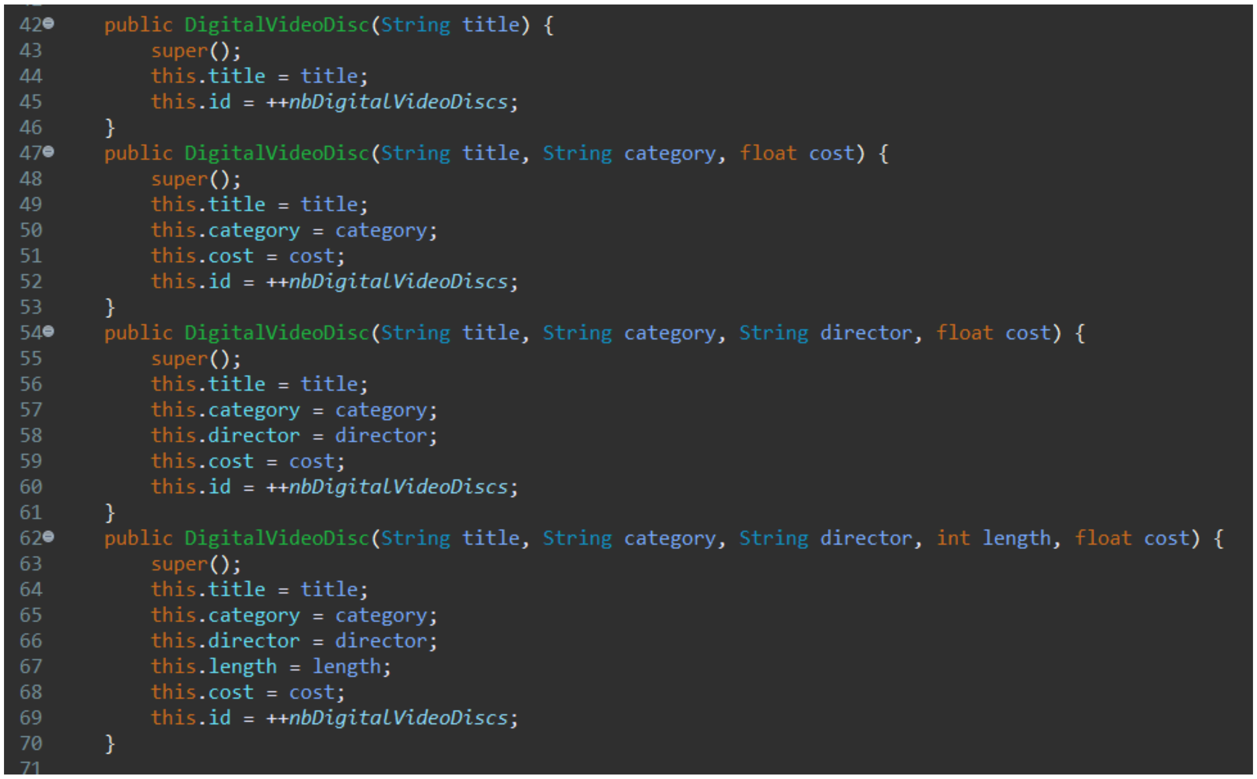
```java
public class TestPassingParameter {

    public static void main(String[] args) {
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");
        DigitalVideoDiscWrapper wjungleDVD = new DigitalVideoDiscWrapper(jungleDVD);
        DigitalVideoDiscWrapper wcinderellaDVD = new DigitalVideoDiscWrapper(cinderellaDVD);
        // Wrong swap() function
        swap(jungleDVD, cinderellaDVD);
        System.out.println("Wrong swap jungle dvd title: " + jungleDVD.getTitle());
        System.out.println("Wrong swap cinderella dvd title: " + cinderellaDVD.getTitle());
        // Correct swap() function
        swap(wjungleDVD, wcinderellaDVD);
        System.out.println("Correct swapped jungle dvd title: " + wjungleDVD.dvd.getTitle());
        System.out.println("Correct swapped cinderella dvd title: " + wcinderellaDVD.dvd.getTitle());

        changeTitle(jungleDVD, cinderellaDVD.getTitle());
        System.out.println("Change jungle dvd title: " + jungleDVD.getTitle());
    }
    // Wrong
    public static void swap(Object o1, Object o2)
    {
        Object tmp = o1;
        o1 = o2;
        o2 = tmp;
    }
    // Correct
    public static void swap(DigitalVideoDiscWrapper o1, DigitalVideoDiscWrapper o2)
    {
        DigitalVideoDisc tmp = o1.dvd;
        o1.dvd = o2.dvd;
        o2.dvd = tmp;
    }
    public static void changeTitle(DigitalVideoDisc dvd, String title)
    {
        String oldTitle = dvd.getTitle();
        dvd.setTitle(title);
        dvd = new DigitalVideoDisc(oldTitle);
    }

}
```

3. Classifier Member and Instance Member

- Tại class **DigitalVideoDisc** thêm instance như sau



```java
    private static int nbDigitalVideoDiscs = 0;
    private int id;
```

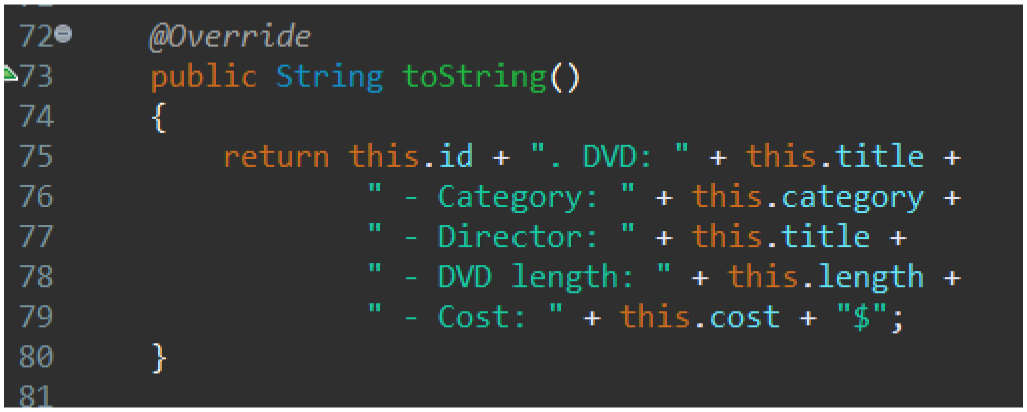


```java
    public DigitalVideoDisc(String title) {
        super();
        this.title = title;
        this.id = ++nbDigitalVideoDiscs;
    }
    public DigitalVideoDisc(String title, String category, float cost) {
        super();
        this.title = title;
        this.category = category;
        this.cost = cost;
        this.id = ++nbDigitalVideoDiscs;
    }
    public DigitalVideoDisc(String title, String category, String director, float cost) {
        super();
        this.title = title;
        this.category = category;
        this.director = director;
        this.cost = cost;
        this.id = ++nbDigitalVideoDiscs;
    }
    public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
        super();
        this.title = title;
        this.category = category;
        this.director = director;
        this.length = length;
        this.cost = cost;
        this.id = ++nbDigitalVideoDiscs;
    }
```

4. Print and Search Cart

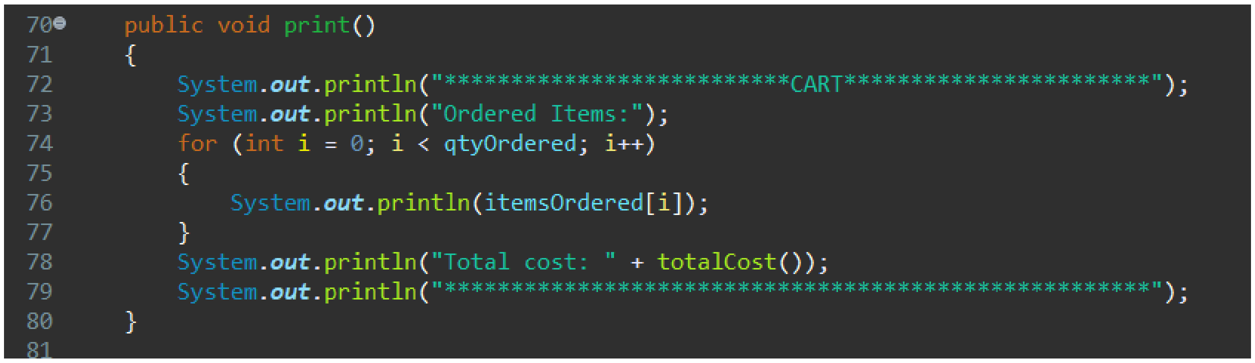
4.1 Print Cart

- Viết lại hàm **toString()** để thay đổi kiểu trả về của string ở trong class **DigitalVideoDisc**.



```java
    @Override
    public String toString()
    {
        return this.id + ". DVD: " + this.title +
                 " - Category: " + this.category +
                 " - Director: " + this.title +
                 " - DVD length: " + this.length +
                 " - Cost: " + this.cost + "$";
    }
```

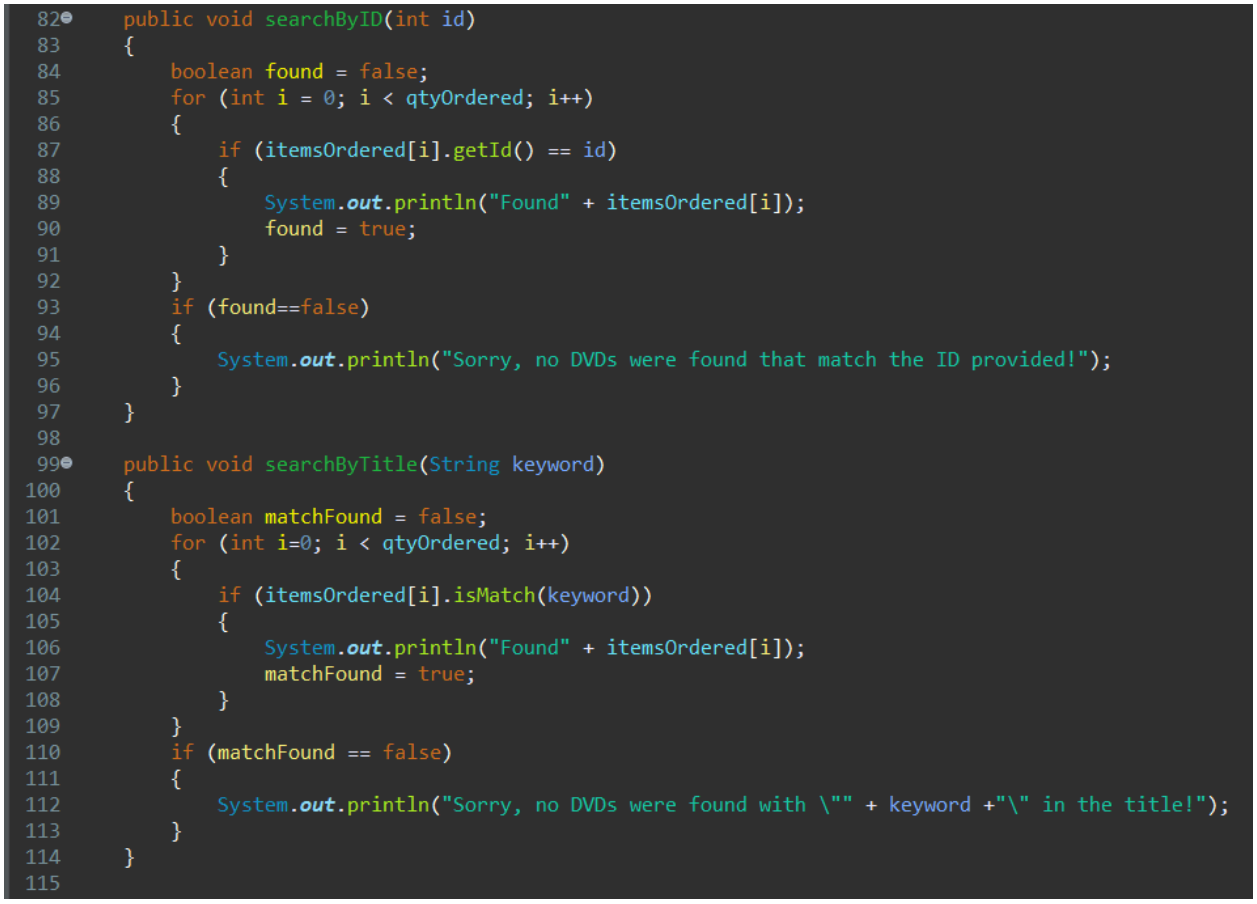
- Tạo phương thức **print()** in danh sách các mặt hàng trong class **Cart**.



```java
public void print()
{
    System.out.println("**************************CART************************");
    System.out.println("Ordered Items:");
    for (int i = 0; i < qtyOrdered; i++)
    {
        System.out.println(itemsOrdered[i]);
    }
    System.out.println("Total cost: " + totalCost());
    System.out.println("******************************************************");
}
```

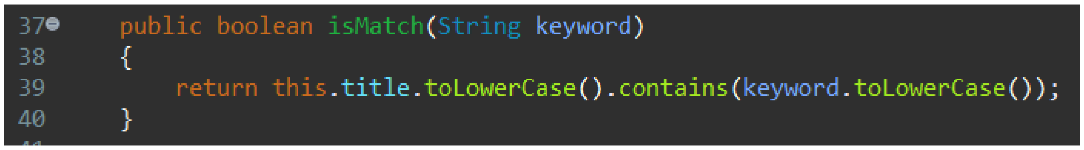
4.2 Search Cart

- Tìm kiếm theo ID và tìm kiếm theo Title ở trong class **Cart**.



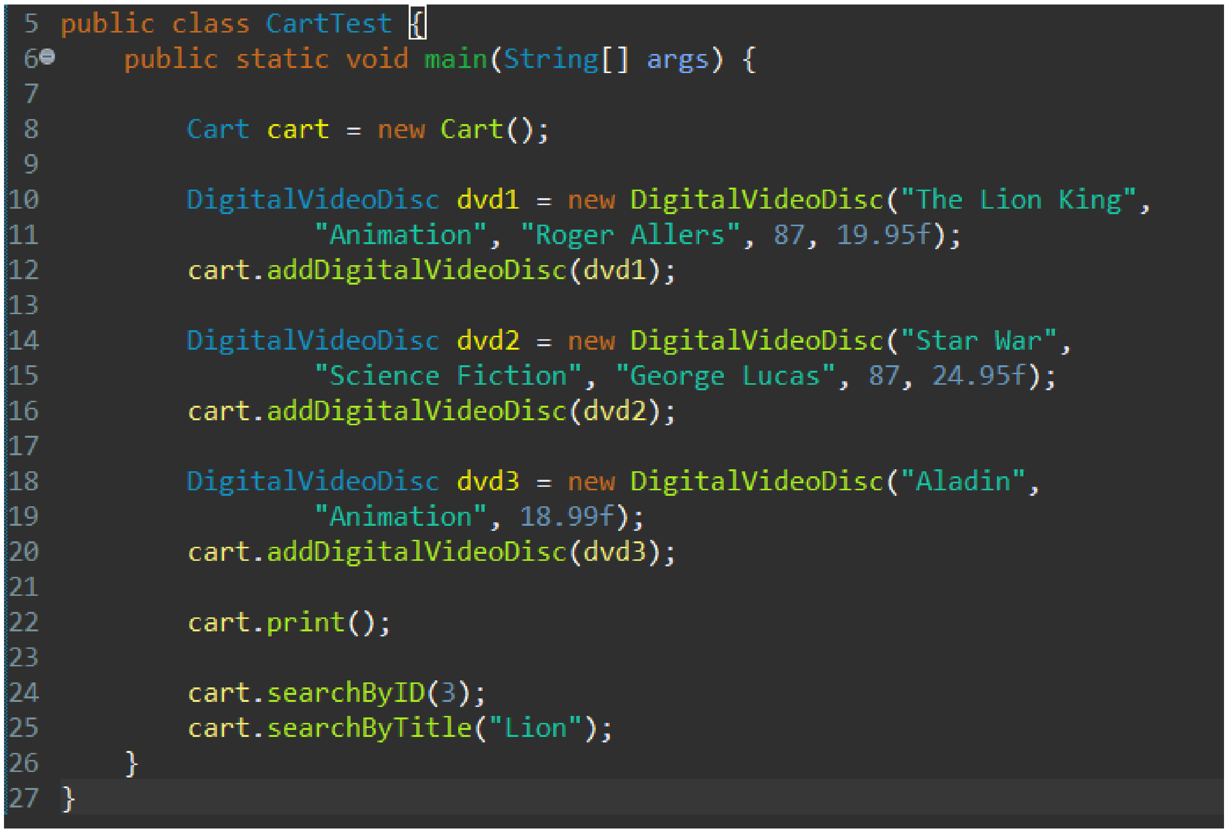
```java
public void searchByID(int id)
{
    boolean found = false;
    for (int i = 0; i < qtyOrdered; i++)
    {
        if (itemsOrdered[i].getId() == id)
        {
            System.out.println("Found" + itemsOrdered[i]);
            found = true;
        }
    }
    if (found==false)
    {
        System.out.println("Sorry, no DVDs were found that match the ID provided!");
    }
}

public void searchByTitle(String keyword)
{
    boolean matchFound = false;
    for (int i=0; i < qtyOrdered; i++)
    {
        if (itemsOrdered[i].isMatch(keyword))
        {
            System.out.println("Found" + itemsOrdered[i]);
            matchFound = true;
        }
    }
    if (matchFound == false)
    {
        System.out.println("Sorry, no DVDs were found with \"" + keyword +"\" in the title!");
    }
}
```

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



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

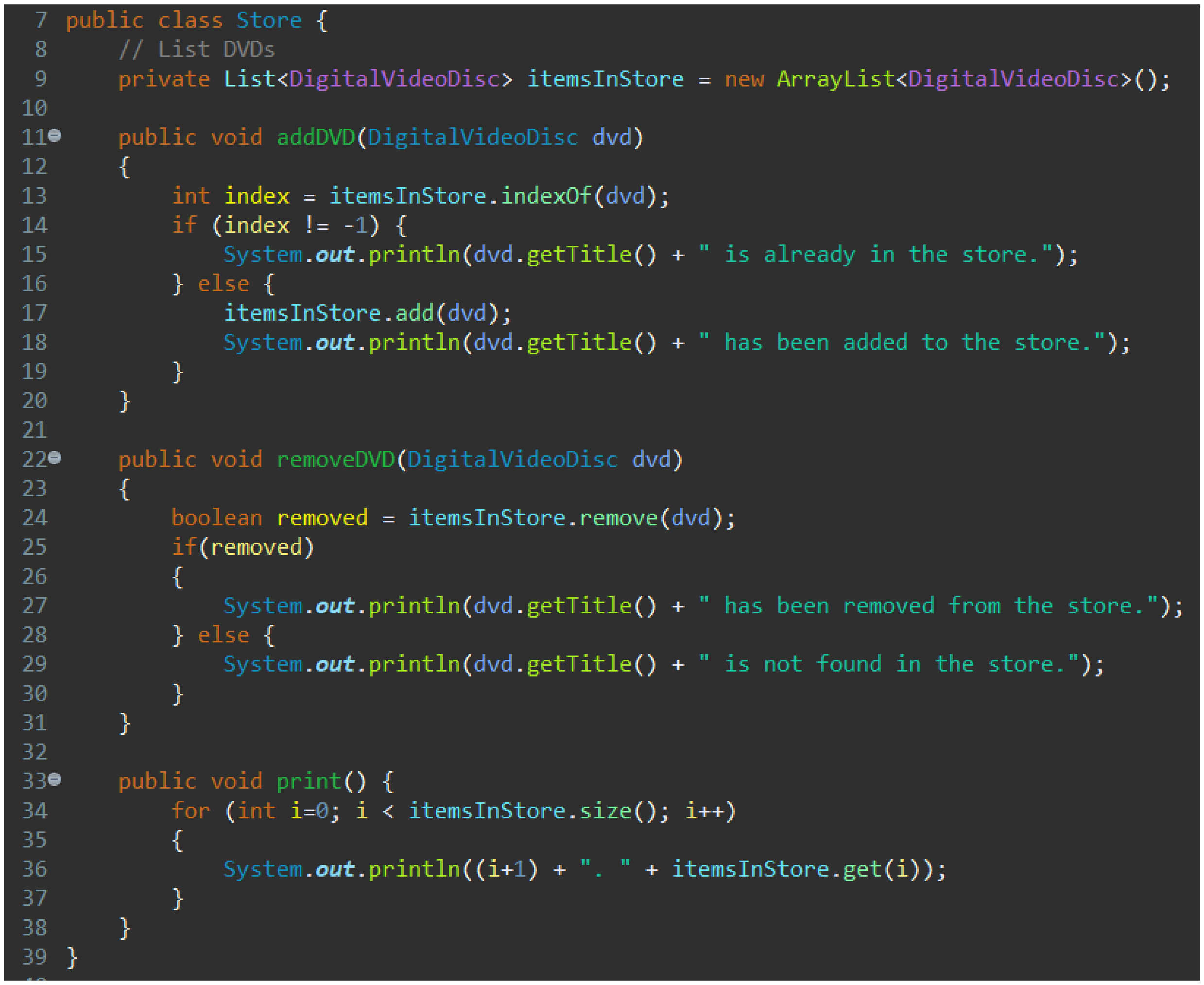
- Tạo class **CartTest** để kiểm tra các hàm trên



```
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 War",
                "Science Fiction", "George Lucas", 87, 24.95f);
        cart.addDigitalVideoDisc(dvd2);

        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin",
                "Animation", 18.99f);
        cart.addDigitalVideoDisc(dvd3);

        cart.print();

        cart.searchByID(3);
        cart.searchByTitle("Lion");
    }
}
```

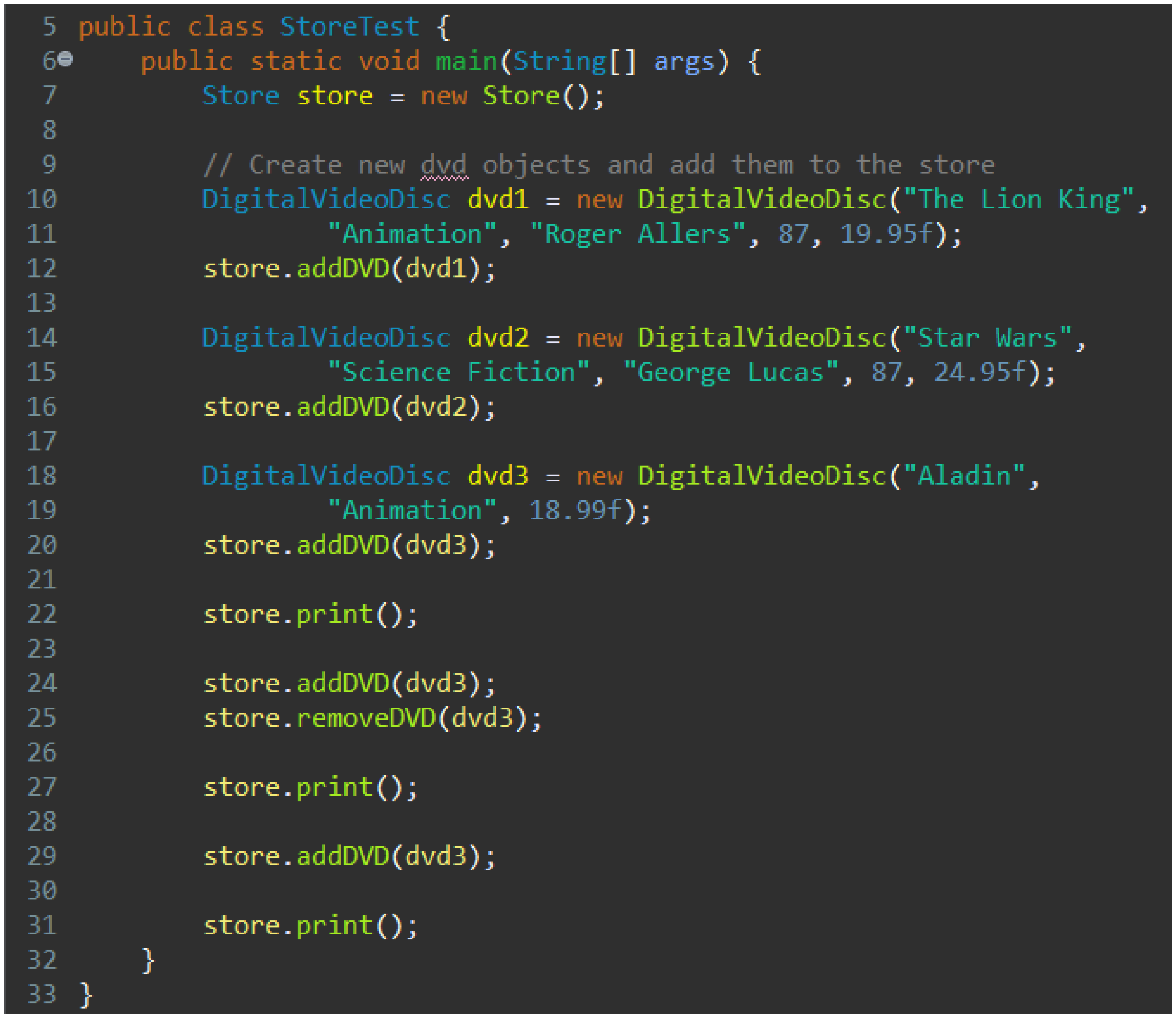
5. Implement the Store class

- Tạo class **Store**, chứa thuộc tính **itemsInStore[]** là một mảng các DVD có sẵn trong cửa hàng.
- Thêm 2 phương thức **addDVD** và **removeDVD**



```java
public class Store {
    // List DVDs
    private List<DigitalVideoDisc> itemsInStore = new ArrayList<DigitalVideoDisc>();

    public void addDVD(DigitalVideoDisc dvd)
    {
        int index = itemsInStore.indexOf(dvd);
        if (index != -1) {
            System.out.println(dvd.getTitle() + " is already in the store.");
        } else {
            itemsInStore.add(dvd);
            System.out.println(dvd.getTitle() + " has been added to the store.");
        }
    }

    public void removeDVD(DigitalVideoDisc dvd)
    {
        boolean removed = itemsInStore.remove(dvd);
        if(removed)
        {
            System.out.println(dvd.getTitle() + " has been removed from the store.");
        } else {
            System.out.println(dvd.getTitle() + " is not found in the store.");
        }
    }

    public void print() {
        for (int i=0; i < itemsInStore.size(); i++)
        {
            System.out.println((i+1) + ". " + itemsInStore.get(i));
        }
    }
}
```

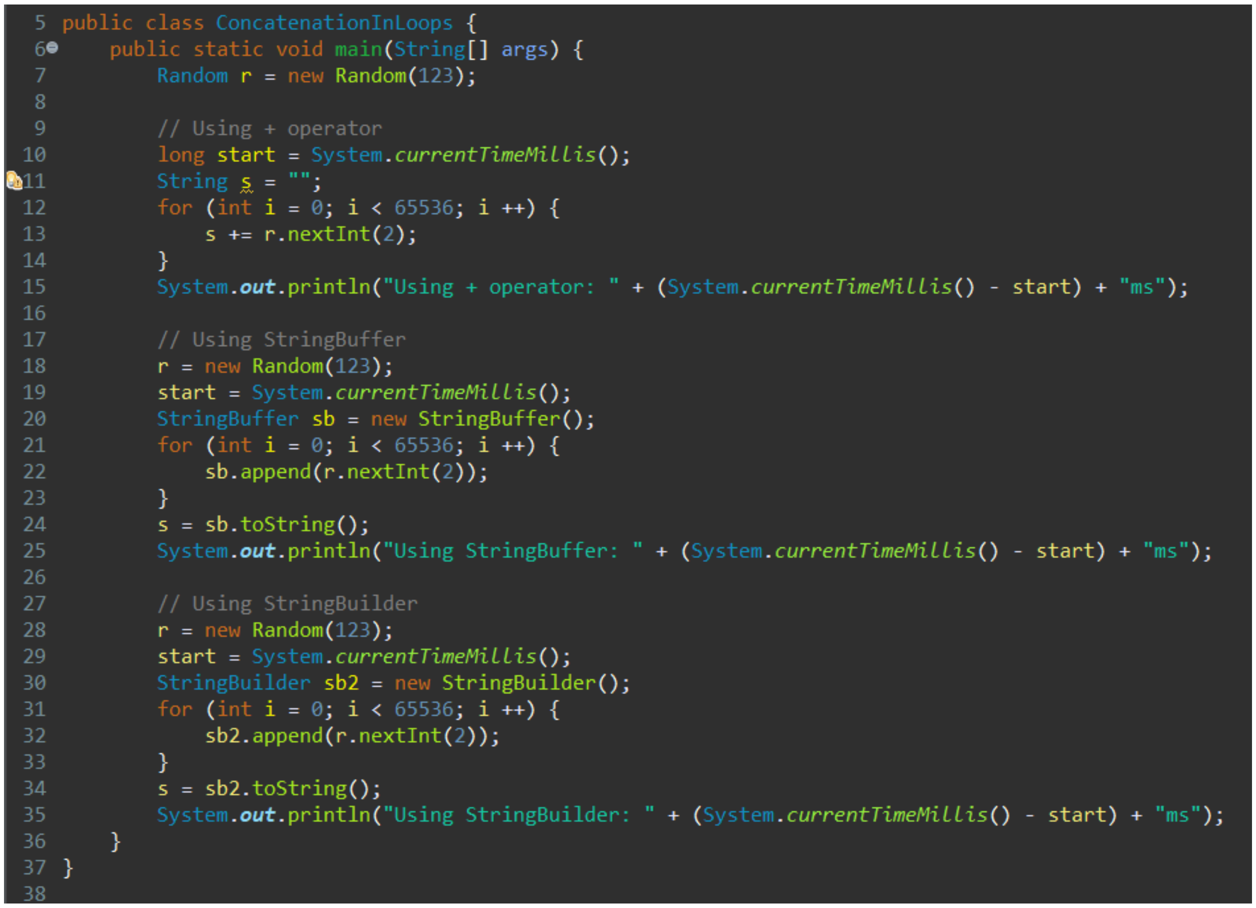
- Tạo class **StoreTest** để kiểm tra 2 phương thức trên



```java
public class StoreTest {
    public static void main(String[] args) {
        Store store = new Store();

        // Create new dvd objects and add them to the store
        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King",
                "Animation", "Roger Allers", 87, 19.95f);
        store.addDVD(dvd1);

        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars",
                "Science Fiction", "George Lucas", 87, 24.95f);
        store.addDVD(dvd2);

        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin",
                "Animation", 18.99f);
        store.addDVD(dvd3);

        store.print();

        store.addDVD(dvd3);
        store.removeDVD(dvd3);

        store.print();

        store.addDVD(dvd3);

        store.print();
    }
}
```

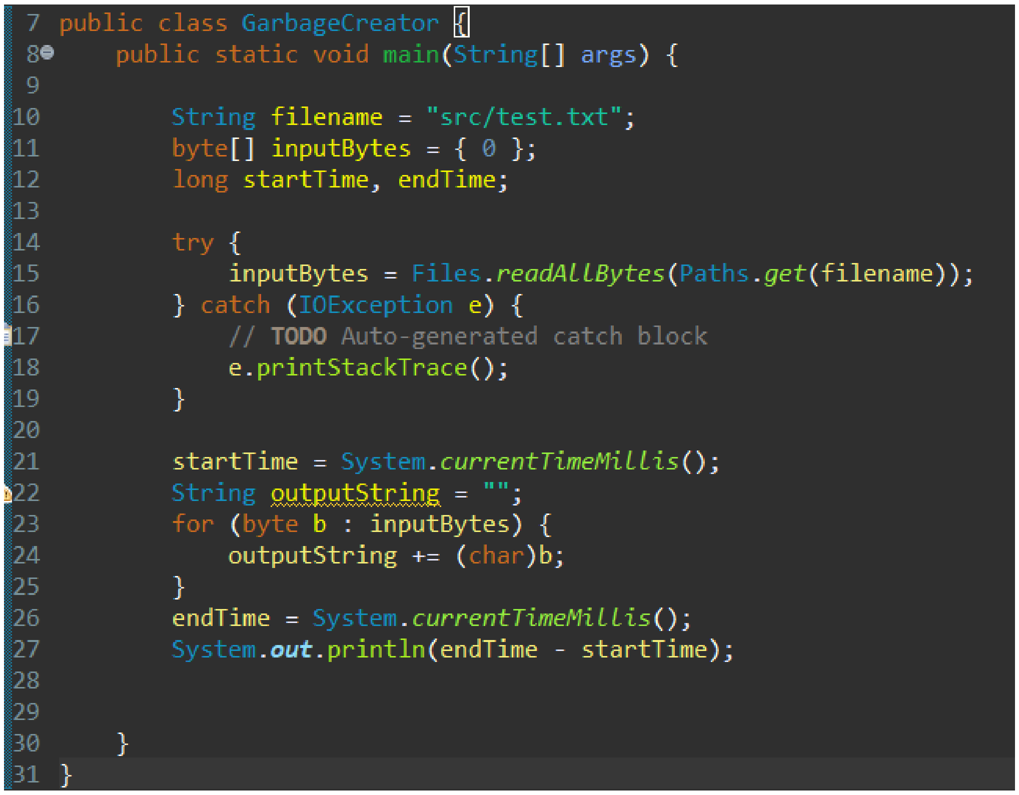
6. String, StringBuilder and StringBuffer

- 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**.



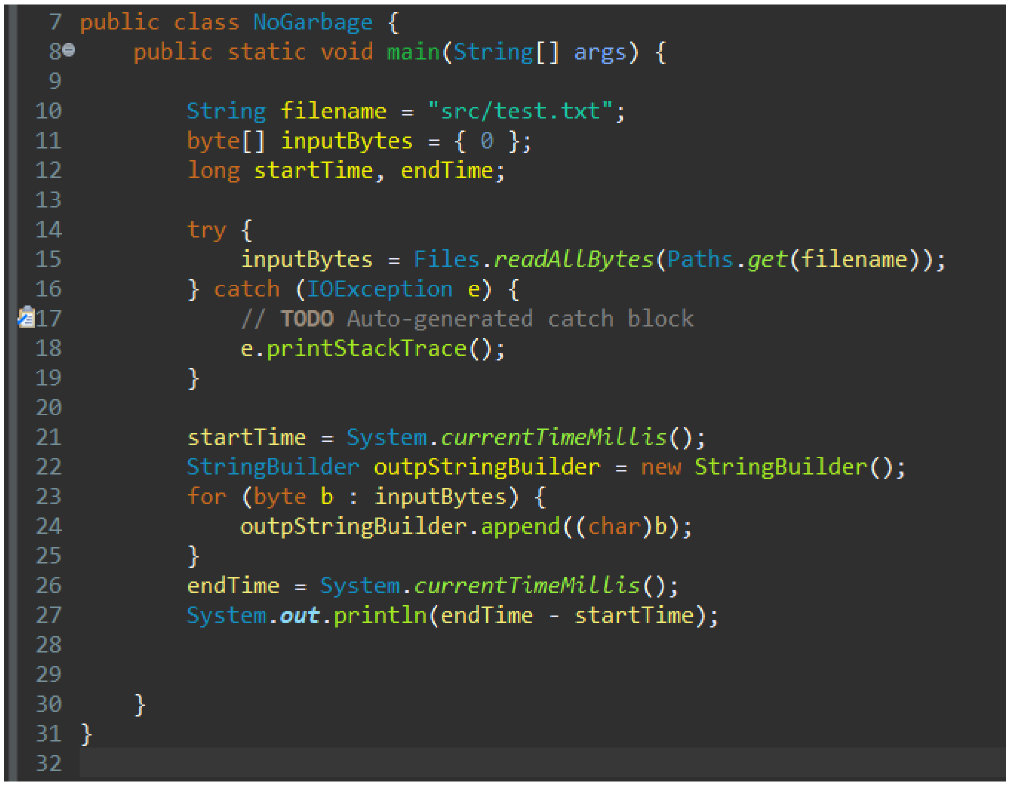
```java
public class ConcatenationInLoops {
    public static void main(String[] args) {
        Random r = new Random(123);

        // Using + operator
        long start = System.currentTimeMillis();
        String s = "";
        for (int i = 0; i < 65536; i ++) {
            s += r.nextInt(2);
        }
        System.out.println("Using + operator: " + (System.currentTimeMillis() - start) + "ms");

        // Using StringBuffer
        r = new Random(123);
        start = System.currentTimeMillis();
        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < 65536; i ++) {
            sb.append(r.nextInt(2));
        }
        s = sb.toString();
        System.out.println("Using StringBuffer: " + (System.currentTimeMillis() - start) + "ms");

        // Using StringBuilder
        r = new Random(123);
        start = System.currentTimeMillis();
        StringBuilder sb2 = new StringBuilder();
        for (int i = 0; i < 65536; i ++) {
            sb2.append(r.nextInt(2));
        }
        s = sb2.toString();
        System.out.println("Using StringBuilder: " + (System.currentTimeMillis() - start) + "ms");
    }
}

```

- Tạo một class mới là **GarbageCreator** đọc tệp văn bản vào 1 chuỗi dùng **+**



```java
public class GarbageCreator {
    public static void main(String[] args) {

        String filename = "src/test.txt";
        byte[] inputBytes = { 0 };
        long startTime, endTime;

        try {
            inputBytes = Files.readAllBytes(Paths.get(filename));
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        startTime = System.currentTimeMillis();
        String outputString = "";
        for (byte b : inputBytes) {
            outputString += (char)b;
        }
        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);


    }
}
```

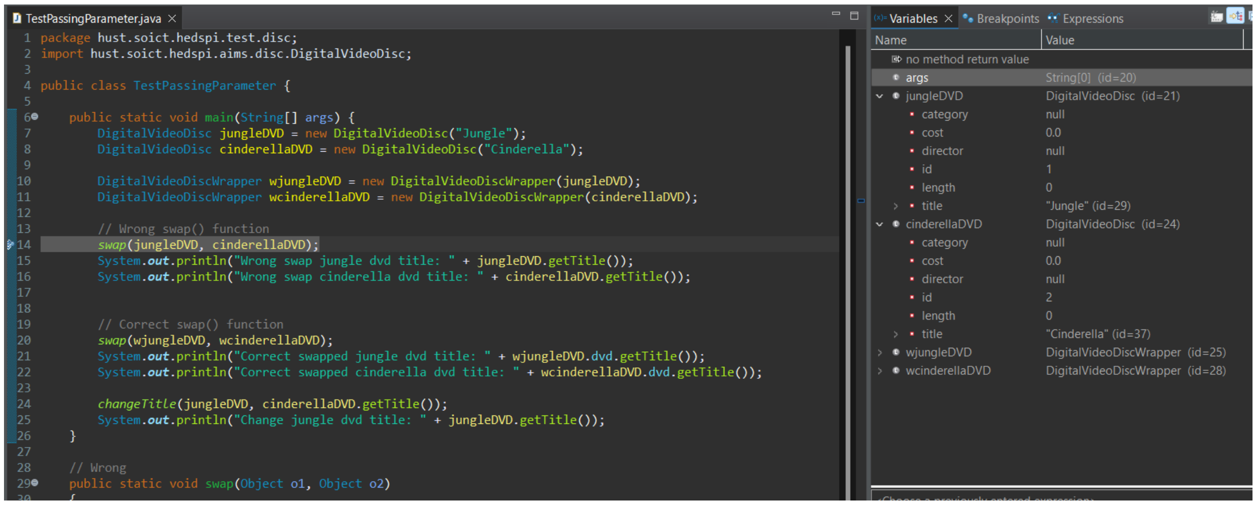
- Tạo một class mới là **NoGarbage** đọc tệp văn bản vào 1 chuỗi dùng **StringBuilder**



```java
public class NoGarbage {
    public static void main(String[] args) {

        String filename = "src/test.txt";
        byte[] inputBytes = { 0 };
        long startTime, endTime;

        try {
            inputBytes = Files.readAllBytes(Paths.get(filename));
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        startTime = System.currentTimeMillis();
        StringBuilder outpStringBuilder = new StringBuilder();
        for (byte b : inputBytes) {
            outpStringBuilder.append((char)b);
        }
        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);


    }
}

```

II. Debug

- Debug tại class **TestPassingParameter** với hàm **swap()**
- Đặt breakpoint tại swap(jungleDVD, cinderellaDVD);



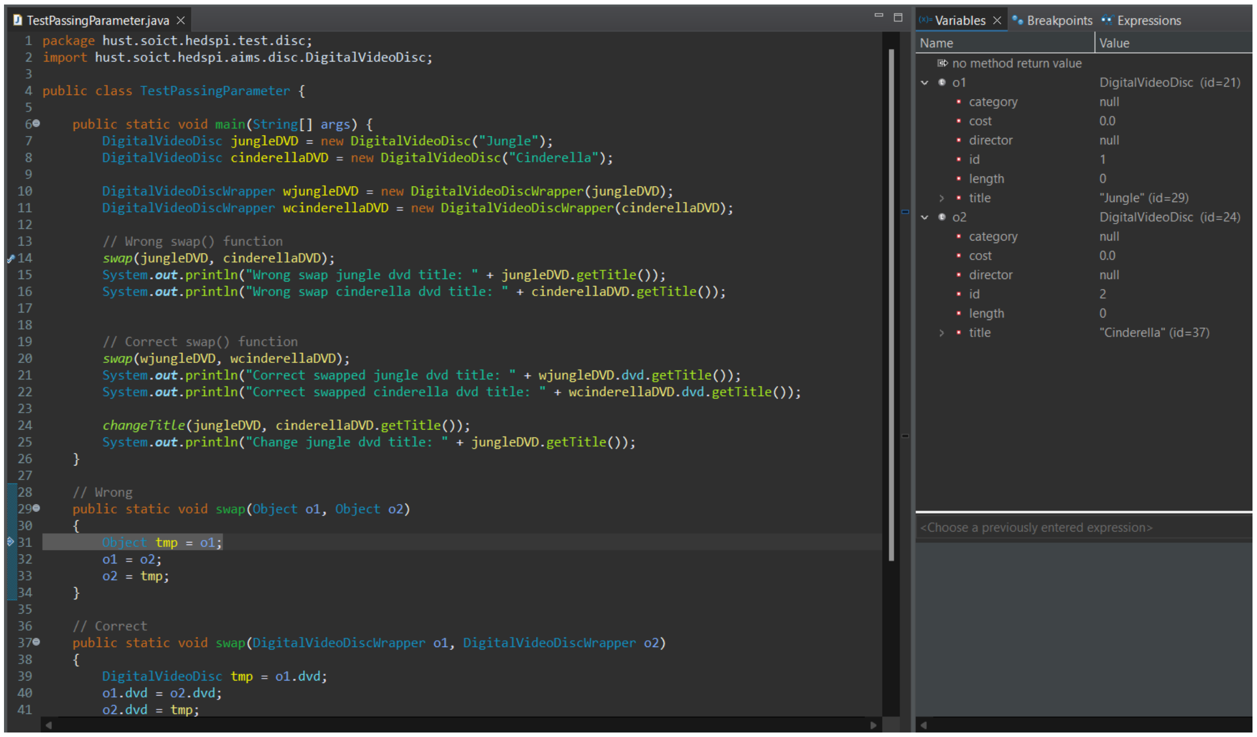
TestPassingParameter.java

```java
package hust.soict.hedspi.test.disc;
import hust.soict.hedspi.aims.disc.DigitalVideoDisc;

public class TestPassingParameter {

    public static void main(String[] args) {
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");

        DigitalVideoDiscWrapper wjungleDVD = new DigitalVideoDiscWrapper(jungleDVD);
        DigitalVideoDiscWrapper wcinderellaDVD = new DigitalVideoDiscWrapper(cinderellaDVD);

        // Wrong swap() function
        swap(jungleDVD, cinderellaDVD);
        System.out.println("Wrong swap jungle dvd title: " + jungleDVD.getTitle());
        System.out.println("Wrong swap cinderella dvd title: " + cinderellaDVD.getTitle());


        // Correct swap() function
        swap(wjungleDVD, wcinderellaDVD);
        System.out.println("Correct swapped jungle dvd title: " + wjungleDVD.dvd.getTitle());
        System.out.println("Correct swapped cinderella dvd title: " + wcinderellaDVD.dvd.getTitle());

        changeTitle(jungleDVD, cinderellaDVD.getTitle());
        System.out.println("Change jungle dvd title: " + jungleDVD.getTitle());
    }

    // Wrong
    public static void swap(Object o1, Object o2)
```

Variables | Breakpoints | Expressions

Name	Value
no method return value	
args	String[0] (id=20)
jungleDVD	DigitalVideoDisc (id=21)
category	null
cost	0.0
director	null
id	1
length	0
title	"Jungle" (id=29)
cinderellaDVD	DigitalVideoDisc (id=24)
category	null
cost	0.0
director	null
id	2
length	0
title	"Cinderella" (id=37)
wjungleDVD	DigitalVideoDiscWrapper (id=25)
wcinderellaDVD	DigitalVideoDiscWrapper (id=28)

- *Step into (F5)* để nhảy vào hàm



TestPassingParameter.java

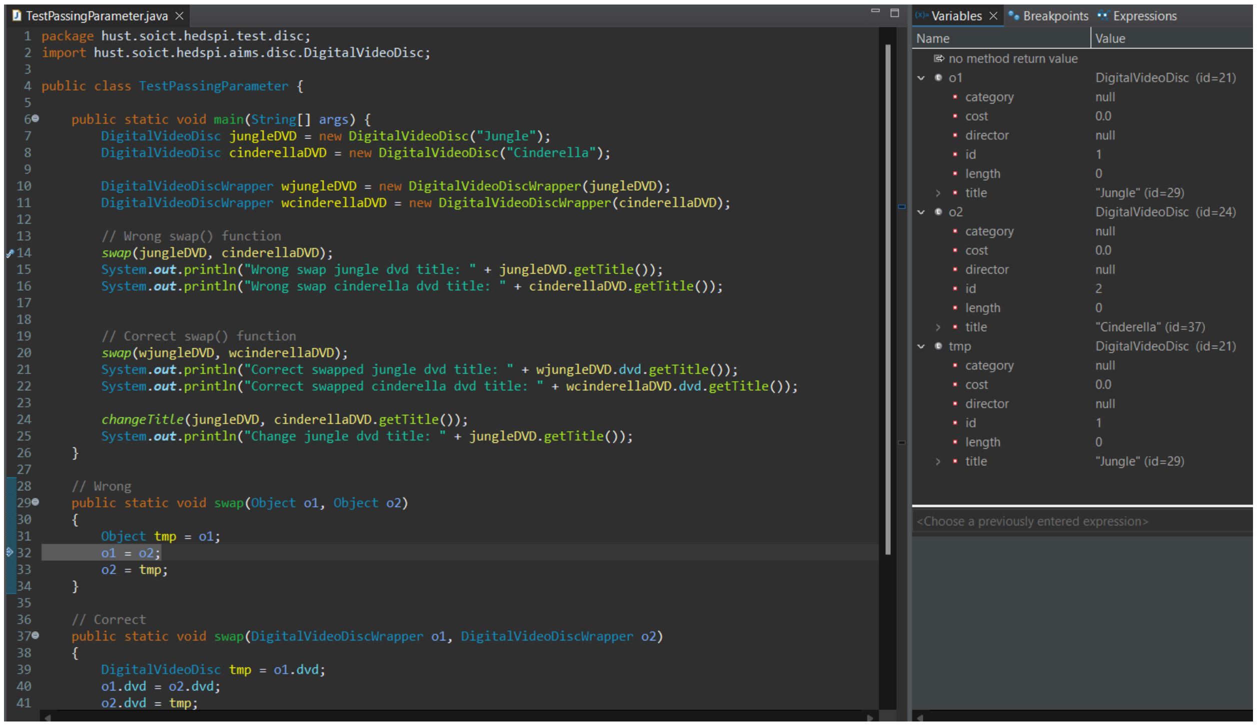
```java
package hust.soict.hedspi.test.disc;
import hust.soict.hedspi.aims.disc.DigitalVideoDisc;

public class TestPassingParameter {

    public static void main(String[] args) {
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");

        DigitalVideoDiscWrapper wjungleDVD = new DigitalVideoDiscWrapper(jungleDVD);
        DigitalVideoDiscWrapper wcinderellaDVD = new DigitalVideoDiscWrapper(cinderellaDVD);

        // Wrong swap() function
        swap(jungleDVD, cinderellaDVD);
        System.out.println("Wrong swap jungle dvd title: " + jungleDVD.getTitle());
        System.out.println("Wrong swap cinderella dvd title: " + cinderellaDVD.getTitle());


        // Correct swap() function
        swap(wjungleDVD, wcinderellaDVD);
        System.out.println("Correct swapped jungle dvd title: " + wjungleDVD.dvd.getTitle());
        System.out.println("Correct swapped cinderella dvd title: " + wcinderellaDVD.dvd.getTitle());

        changeTitle(jungleDVD, cinderellaDVD.getTitle());
        System.out.println("Change jungle dvd title: " + jungleDVD.getTitle());
    }

    // Wrong
    public static void swap(Object o1, Object o2)
    {
        Object tmp = o1;
        o1 = o2;
        o2 = tmp;
    }

    // Correct
    public static void swap(DigitalVideoDiscWrapper o1, DigitalVideoDiscWrapper o2)
    {
        DigitalVideoDisc tmp = o1.dvd;
        o1.dvd = o2.dvd;
        o2.dvd = tmp;
```

Variables | Breakpoints | Expressions

Name	Value
no method return value	
o1	DigitalVideoDisc (id=21)
category	null
cost	0.0
director	null
id	1
length	0
title	"Jungle" (id=29)
o2	DigitalVideoDisc (id=24)
category	null
cost	0.0
director	null
id	2
length	0
title	"Cinderella" (id=37)

<Choose a previously entered expression>

- *Step over (F6)* để chạy dòng lệnh tiếp theo



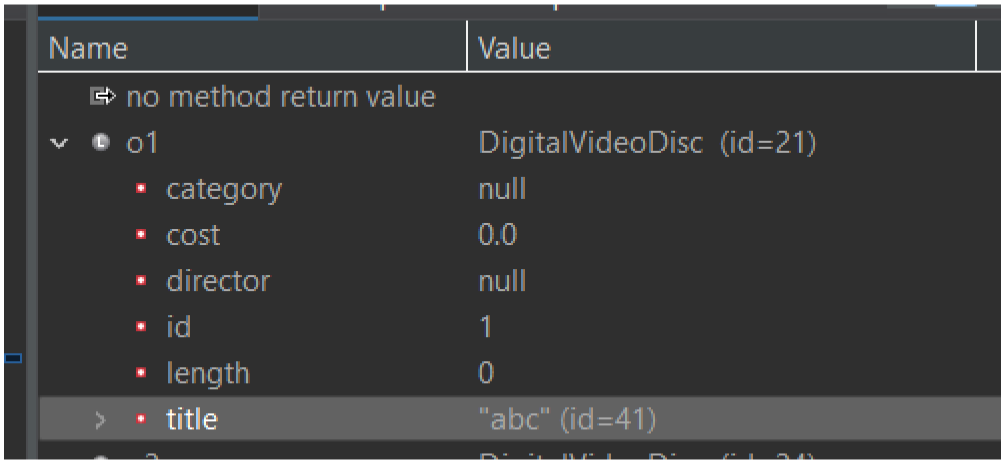
TestPassingParameter.java

```java
package hust.soict.hedspi.test.disc;
import hust.soict.hedspi.aims.disc.DigitalVideoDisc;

public class TestPassingParameter {

    public static void main(String[] args) {
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");

        DigitalVideoDiscWrapper wjungleDVD = new DigitalVideoDiscWrapper(jungleDVD);
        DigitalVideoDiscWrapper wcinderellaDVD = new DigitalVideoDiscWrapper(cinderellaDVD);

        // Wrong swap() function
        swap(jungleDVD, cinderellaDVD);
        System.out.println("Wrong swap jungle dvd title: " + jungleDVD.getTitle());
        System.out.println("Wrong swap cinderella dvd title: " + cinderellaDVD.getTitle());


        // Correct swap() function
        swap(wjungleDVD, wcinderellaDVD);
        System.out.println("Correct swapped jungle dvd title: " + wjungleDVD.dvd.getTitle());
        System.out.println("Correct swapped cinderella dvd title: " + wcinderellaDVD.dvd.getTitle());

        changeTitle(jungleDVD, cinderellaDVD.getTitle());
        System.out.println("Change jungle dvd title: " + jungleDVD.getTitle());
    }

    // Wrong
    public static void swap(Object o1, Object o2)
    {
        Object tmp = o1;
        o1 = o2;
        o2 = tmp;
    }

    // Correct
    public static void swap(DigitalVideoDiscWrapper o1, DigitalVideoDiscWrapper o2)
    {
        DigitalVideoDisc tmp = o1.dvd;
        o1.dvd = o2.dvd;
        o2.dvd = tmp;
```

Name	Value
no method return value	
o1	DigitalVideoDisc (id=21)
category	null
cost	0.0
director	null
id	1
length	0
title	"Jungle" (id=29)
o2	DigitalVideoDisc (id=24)
category	null
cost	0.0
director	null
id	2
length	0
title	"Cinderella" (id=37)
tmp	DigitalVideoDisc (id=21)
category	null
cost	0.0
director	null
id	1
length	0
title	"Jungle" (id=29)

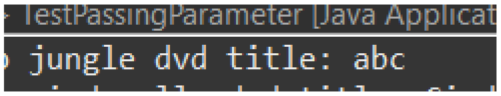
<Choose a previously entered expression>

- Quan sát giá trị của các biến và biểu thức trong **Variables/Expression View**
- Thay đổi giá trị của biến



Name	Value
no method return value	
o1	DigitalVideoDisc (id=21)
category	null
cost	0.0
director	null
id	1
length	0
title	"abc" (id=41)

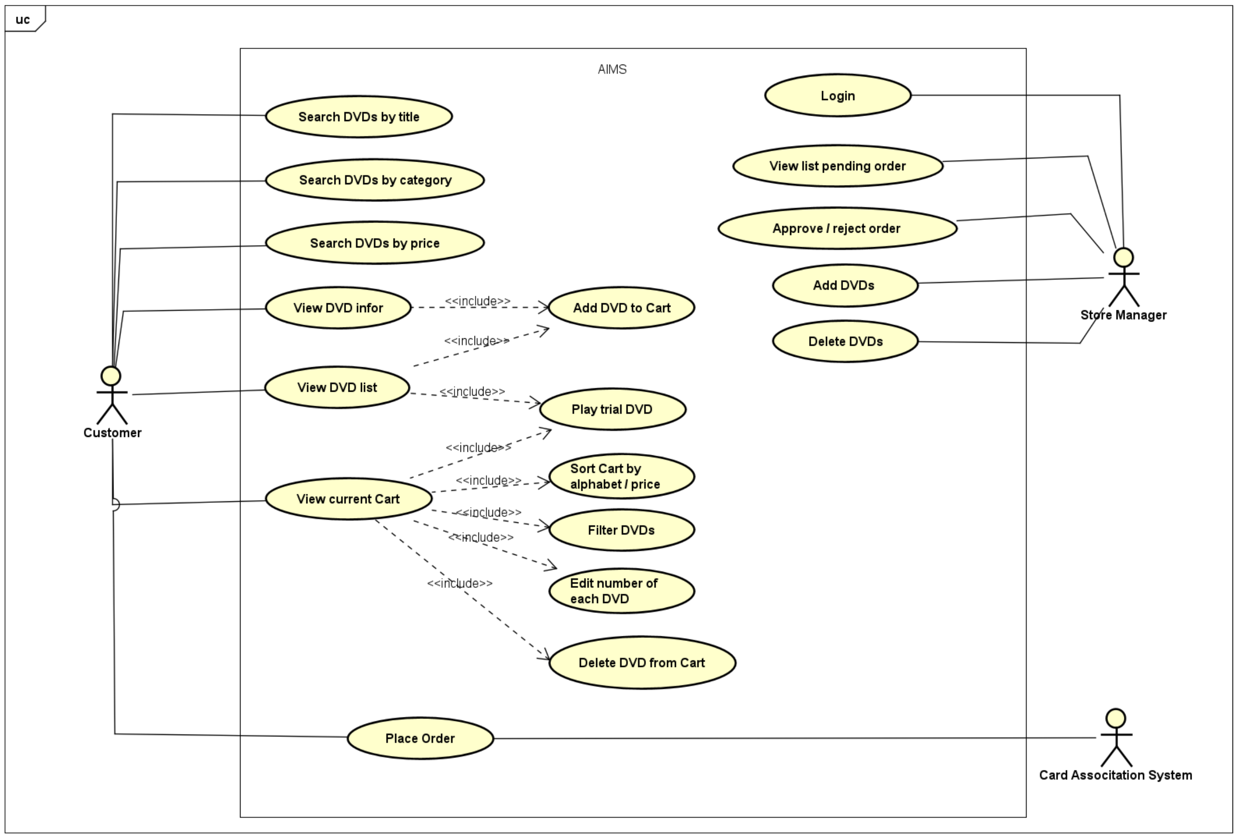
- Kết quả



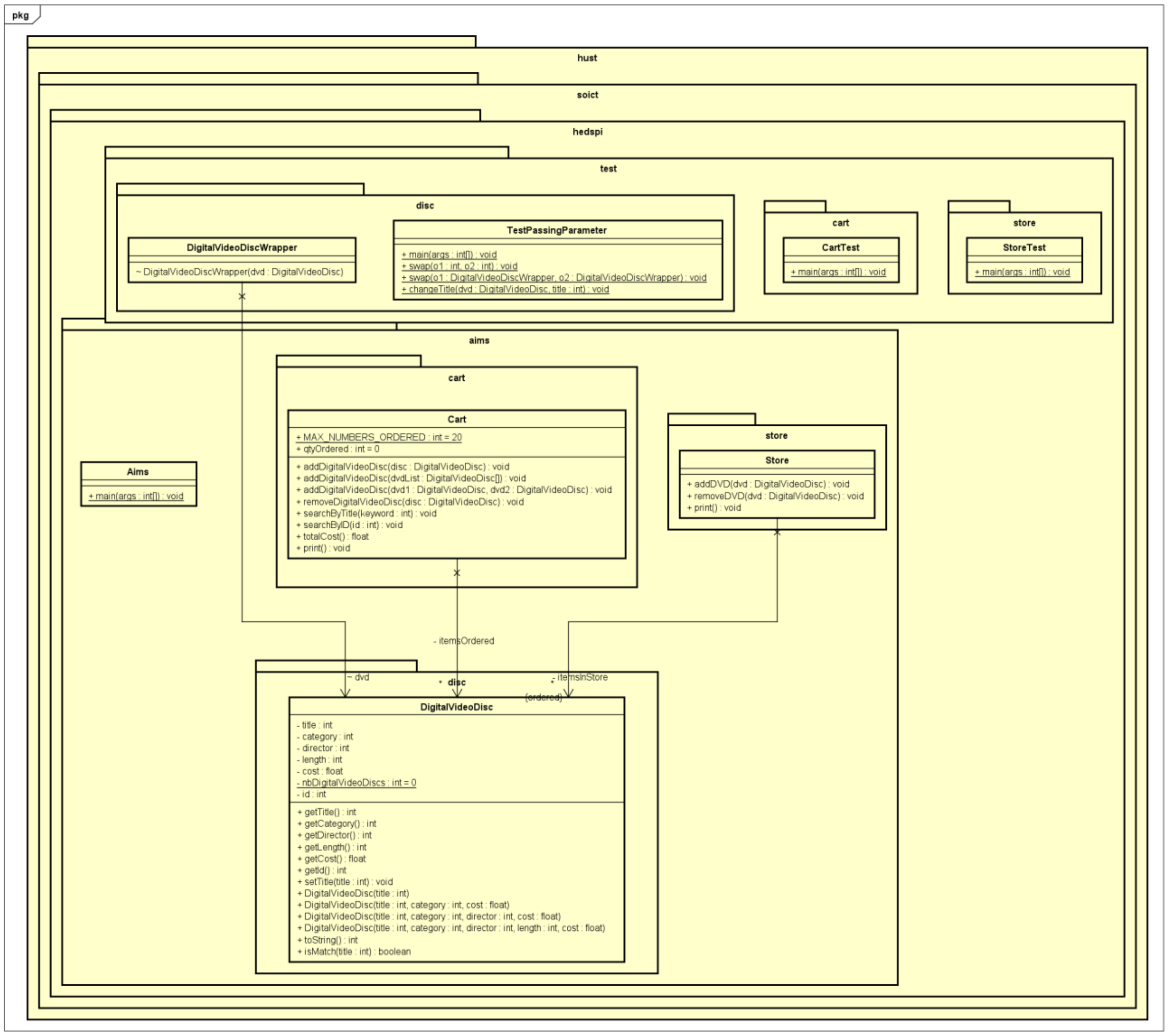
```
TestPassingParameter [Java Applicat
jungle dvd title: abc
```

III. UML Diagram

1. Use-Case Diagram



2. Class Diagram


pkg
hust
soict
hedspi
test
disc
DigitalVideoDiscWrapper
~ DigitalVideoDiscWrapper(dvd : DigitalVideoDisc)
TestPassingParameter
+ main(args : int[]) : void
+ swap(o1 : int, o2 : int) : void
+ swap(o1 : DigitalVideoDiscWrapper, o2 : DigitalVideoDiscWrapper) : void
+ changeTitle(dvd : DigitalVideoDisc, title : int) : void
cart
CartTest
+ main(args : int[]) : void
store
StoreTest
+ main(args : int[]) : void
aims
Aims
+ main(args : int[]) : void
cart
Cart
+ MAX_NUMBERS_ORDERED : int = 20
+ qtyOrdered : int = 0
+ addDigitalVideoDisc(disc : DigitalVideoDisc) : void
+ addDigitalVideoDisc(dvdList : DigitalVideoDisc[]) : void
+ addDigitalVideoDisc(dvd1 : DigitalVideoDisc, dvd2 : DigitalVideoDisc) : void
+ removeDigitalVideoDisc(disc : DigitalVideoDisc) : void
+ searchByTitle(keyword : int) : void
+ searchByID(id : int) : void
+ totalCost() : float
+ print() : void
store
Store
+ addDVD(dvd : DigitalVideoDisc) : void
+ removeDVD(dvd : DigitalVideoDisc) : void
+ print() : void
- itemsOrdered
~ dvd
*
- itemsInStore
*
{ordered}
disc
DigitalVideoDisc
- title : int
- category : int
- director : int
- length : int
- cost : float
- nbDigitalVideoDiscs : int = 0
- id : int
+ getTitle() : int
+ getCategory() : int
+ getDirector() : int
+ getLength() : int
+ getCost() : float
+ getId() : int
+ setTitle(title : int) : void
+ DigitalVideoDisc(title : int)
+ DigitalVideoDisc(title : int, category : int, cost : float)
+ DigitalVideoDisc(title : int, category : int, director : int, cost : float)
+ DigitalVideoDisc(title : int, category : int, director : int, length : int, cost : float)
+ toString() : int
+ isMatch(title : int) : boolean

IV. Answer Question

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

- Java is a **pass-by-value** programming language.

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

- Because the **swap()** method only swaps the values of the title fields between the two objects, but it does not change the object references themselves. Therefore, the object references **jungleDVD** and **cinderellaDVD** still point to the same objects in memory as they did before the **swap()** method was called.

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

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

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**.