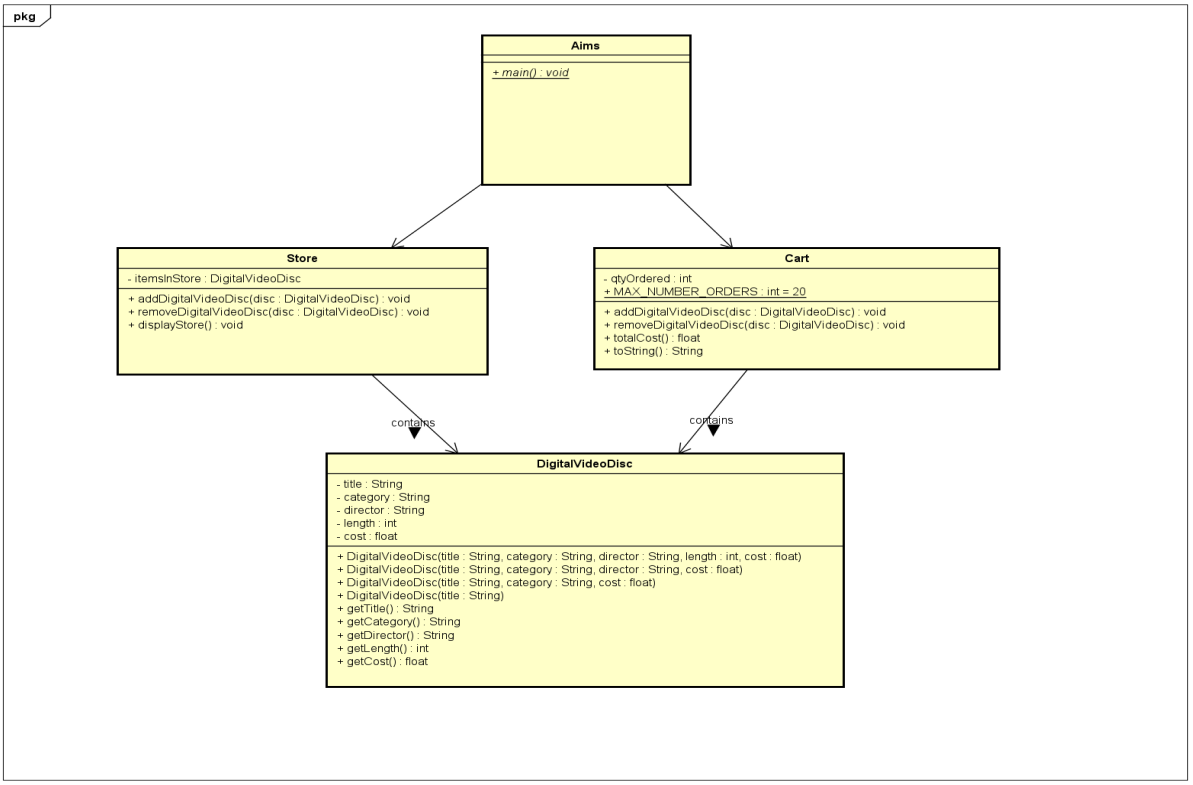
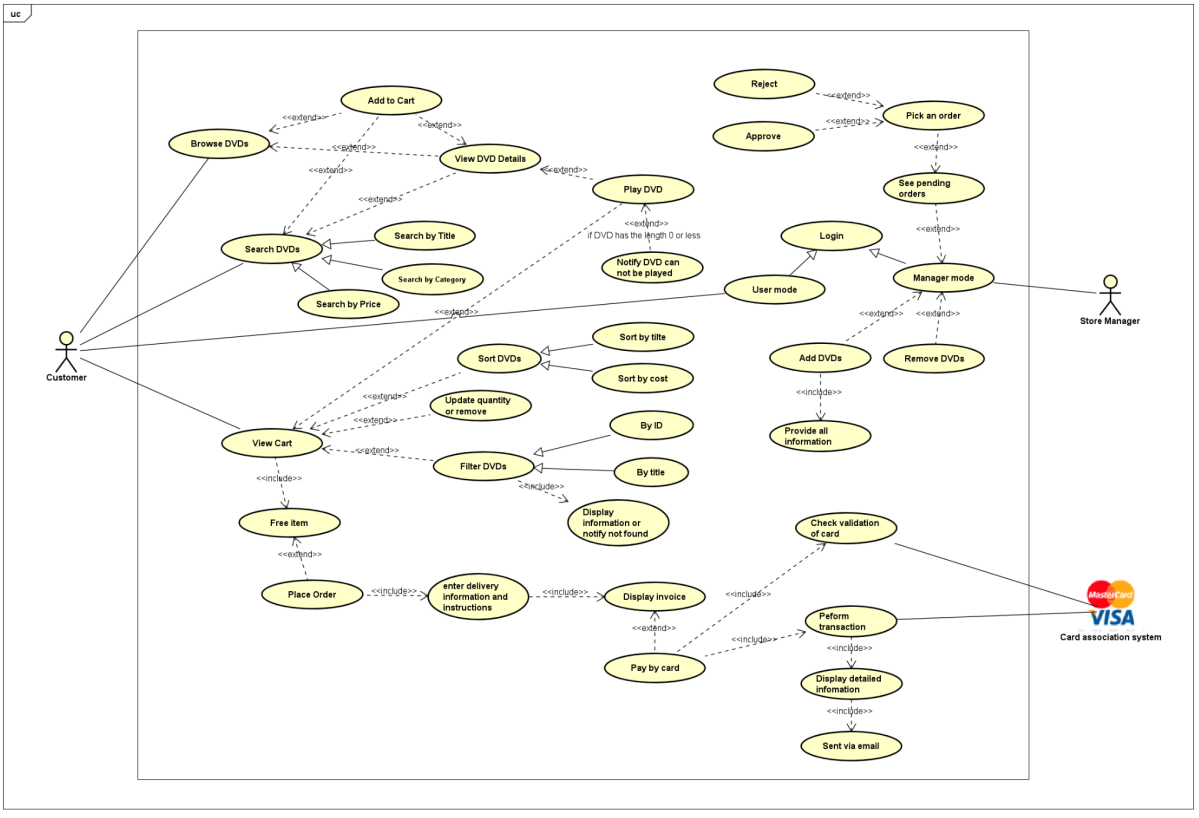
REPORT LAB 03

1. Update use-case diagram and class diagram





2. Method overloading

```java
public void addDigitalVideoDisc(DigitalVideoDisc... discs) {
    for (DigitalVideoDisc disc : discs) {
        if (qtyOrdered == MAX_ORDERED) {
            System.out.println(x:"The cart is full. Cannot add more discs.");
            break;
        } else {
            itemsOrdered[qtyOrdered] = disc;
            qtyOrdered++;
            System.out.println(x:"The disc has been added");
        }
    }
}
```

- In this case, varargs is the better choice because it provides cleaner and more intuitive usage for adding multiple DVDs to the cart. It avoids the extra step of creating an array.

```java
public void addDigitalVideoDisc(DigitalVideoDisc disc1, DigitalVideoDisc disc2) {
    if (qtyOrdered >= MAX_ORDERED){
        System.out.println(x:"The cart is full");
    }
    else {
        itemsOrdered[qtyOrdered] = disc1;
        qtyOrdered++;
        System.out.println(x:"The first cart has been added successfully");
        if (qtyOrdered == MAX_ORDERED){
            System.out.println(x:"The cart is full");
        }else{
            System.out.println(x:"The second cart has been added successfully");
        }
    }
}
```

3. Passing parameter

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

public class TestPassingParameter {
    Run | Debug
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc(title:"Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc(title:"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(DigitalVideoDisc o1, DigitalVideoDisc o2) {···

    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);
    }
}
```

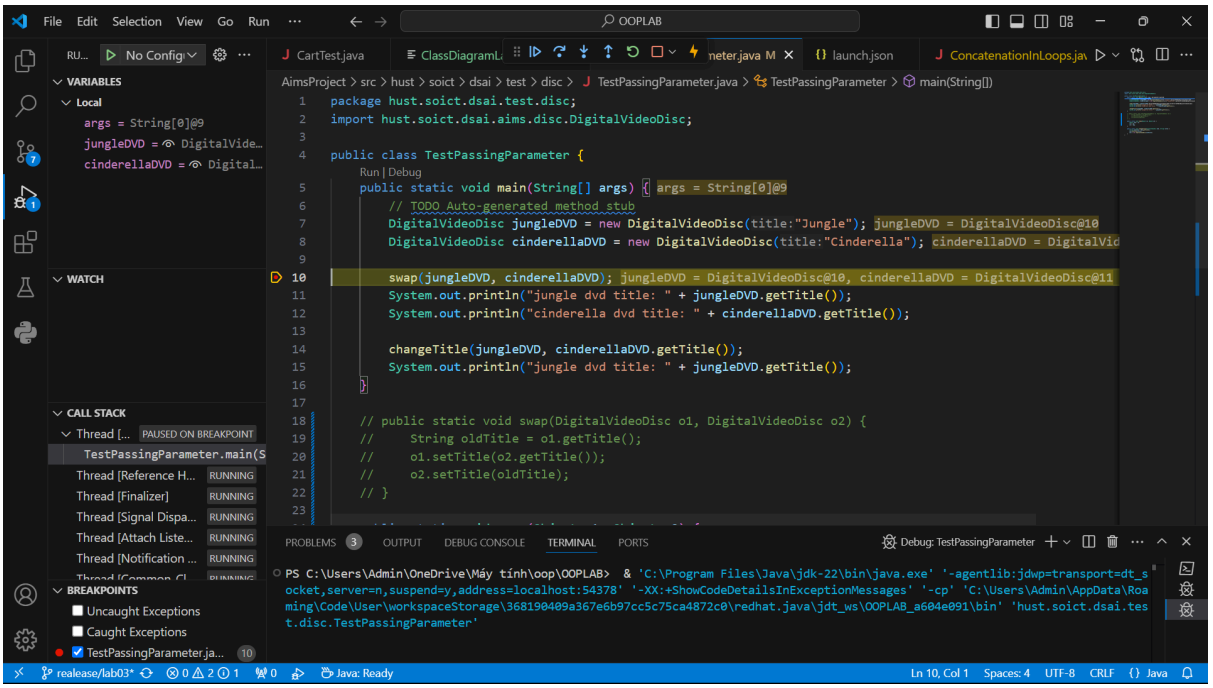
```
jungle dvd title: Jungle
cinderella dvd title: Cinderella
cinderella dvd title: Cinderella
jungle dvd title: Cinderella
```

- ***Is JAVA a Pass by Value or a Pass by Reference programming language?***
    - \+ Java always **pass by value**, so the method swap only swap the copy value of object -> the actual title won't change
    - \+ the reason why method changeTitle works is by using setTitle to modify directly the object
- New code that will work

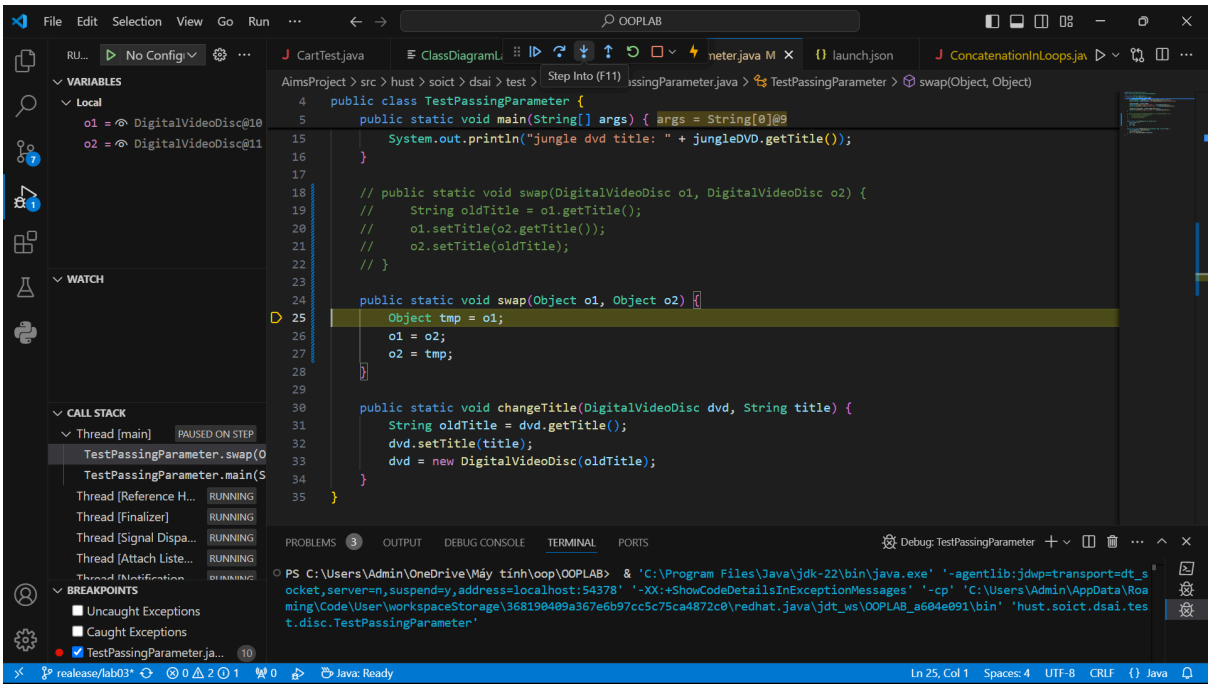
```java
public static void swap(DigitalVideoDisc o1, DigitalVideoDisc o2) {
    String oldTitle = o1.getTitle();
    o1.setTitle(o2.getTitle());
    o2.setTitle(oldTitle);
}
```

4. Debugging

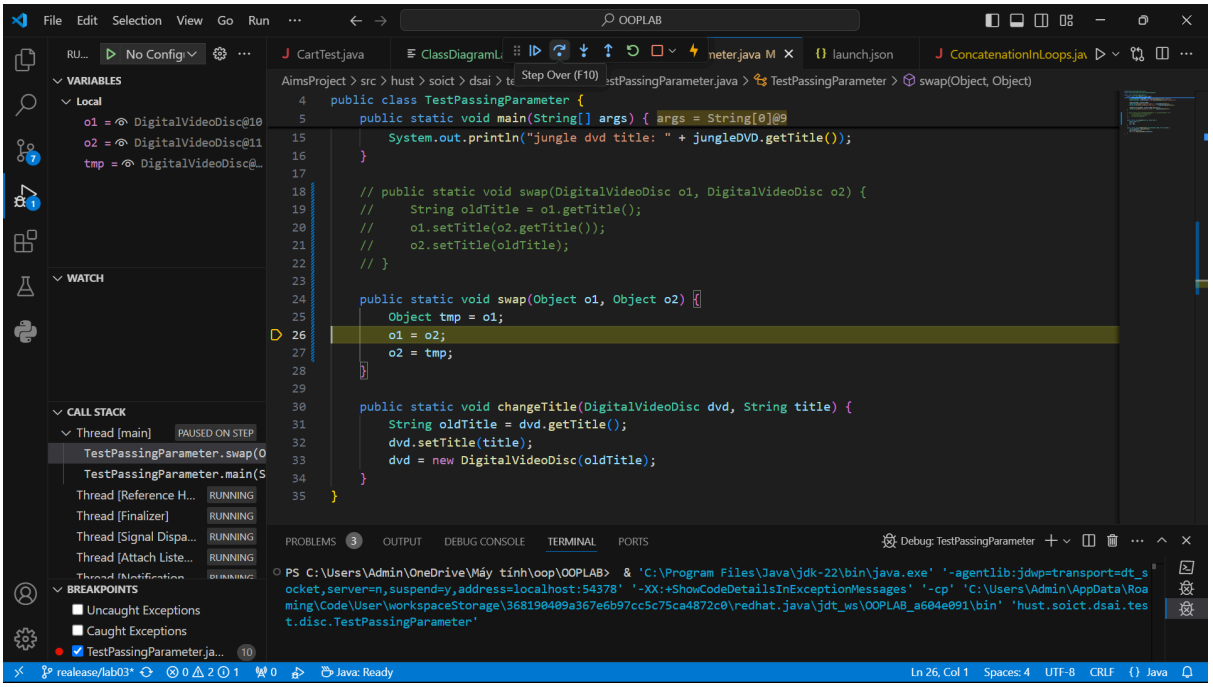
- break point



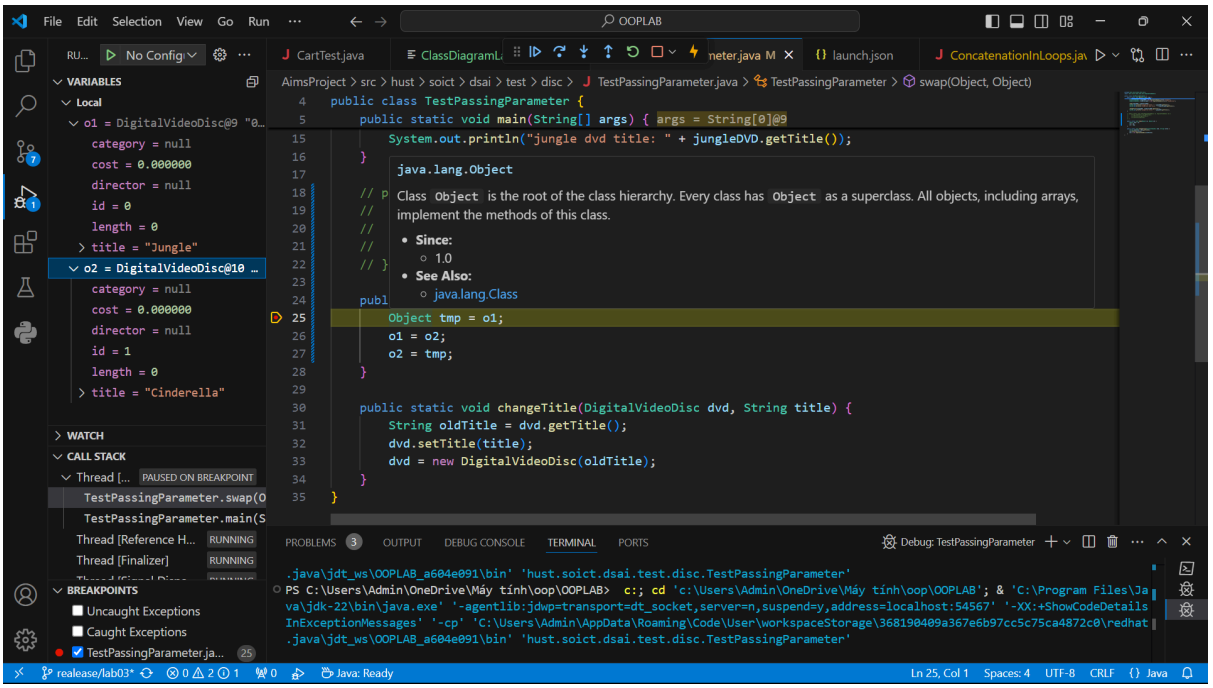
- step into



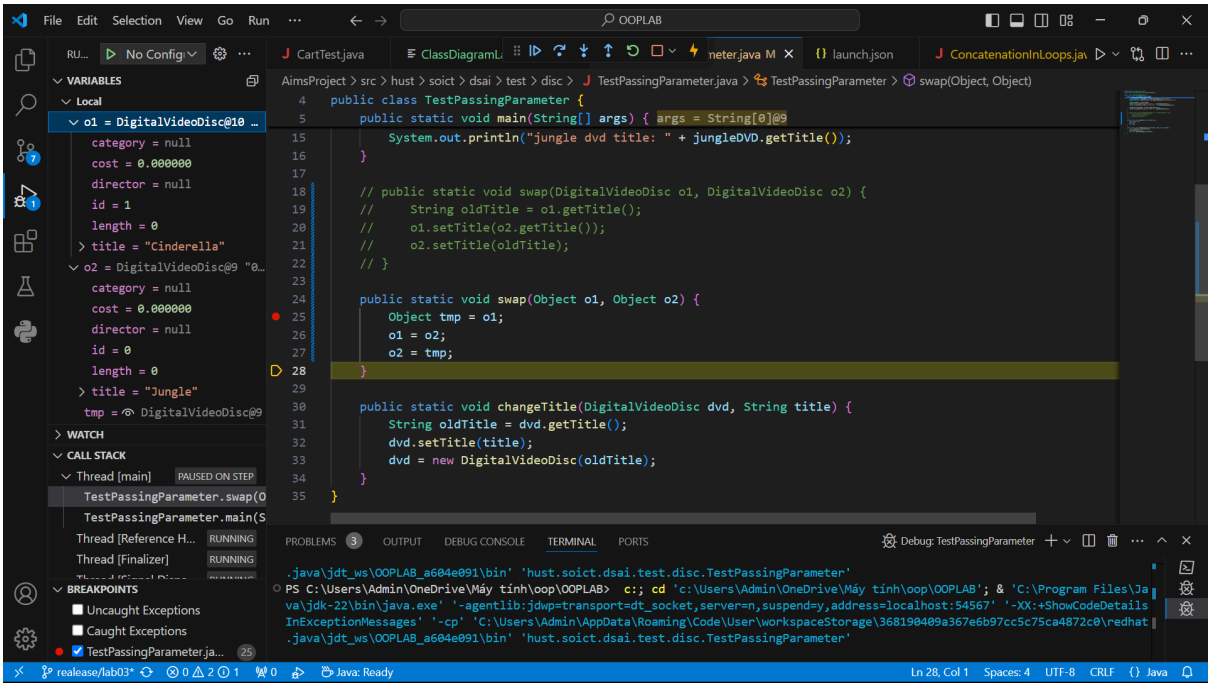
- step over



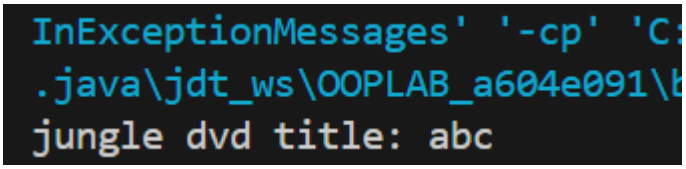
- investigate attributes of variables



- step over swap function



- change title attribute value with “abc”



5. Classifier Member and Instance Member

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

public DigitalVideoDisc(String title) {
    super();
    this.title = title;
    this.id = nbDigitalVideoDiscs;
    nbDigitalVideoDiscs ++;
}

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

6. Open the Cart class

```java
public String toString() {
    return this.id + " - " + this.title + " - " + this.category + " - " + this.director + " - " + this.length + ": " + this.cost + " $";
}
```

- Write a **toString()** method for the **DigitalVideoDisc** class. What should be the return type of this method?
    + it should return a String
- Write a boolean isMatch(String title) method in the DigitalVideoDisc which finds out if the corresponding disk is a match given the title.

```java
public boolean isMatch (String title){
    return this.title.equals(title);
}
```

- Search by id and title

```java
public void searchById(int idNum) {
    int found = 0;
    for(int i = 0; i < qtyOrdered; i++) {
        if (idNum == i){
            System.out.println(itemsOrdered[i].toString());
            found = 1;
            break;
        }
    }
    if(found == 0){
        System.out.println(x:"DVD not found");
    }
}

public void searchByTitle(String title) {
    int found = 0;
    for(int i = 0; i < qtyOrdered; i++){
        if(itemsOrdered[i].isMatch(title)){
            System.out.println(itemsOrdered[i].toString());
            found = 1;
            break;
        }
    }
    if(found == 0){
        System.out.println(x:"DVD not found");
    }
}
```

- CartTest class

```java
package hust.soict.dsai.test.cart;
import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.disc.DigitalVideoDisc;
public class CartTest {
    Run | Debug
    public static void main(String[] args) {

        //Create a new cart
        Cart cart = new Cart();
        //Create new dvd objects and add them to the cart
        DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Liems King",
            category:"Animation", director:"Roger Allers", length:87, cost:19.95f);
        cart.addDigitalVideoDisc(dvd1);

        DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star Wars",
            category:"Science Fiction", director:"George Lucas", length:87, cost:24.95f);
        cart.addDigitalVideoDisc(dvd2);

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

        //Test the print method
        cart.printCart();

        //Test the search methods
        cart.searchById(idNum:5);
        cart.searchByTitle(title:"Aladin");
    }
}
```

7. Implement the Store class

- Store class

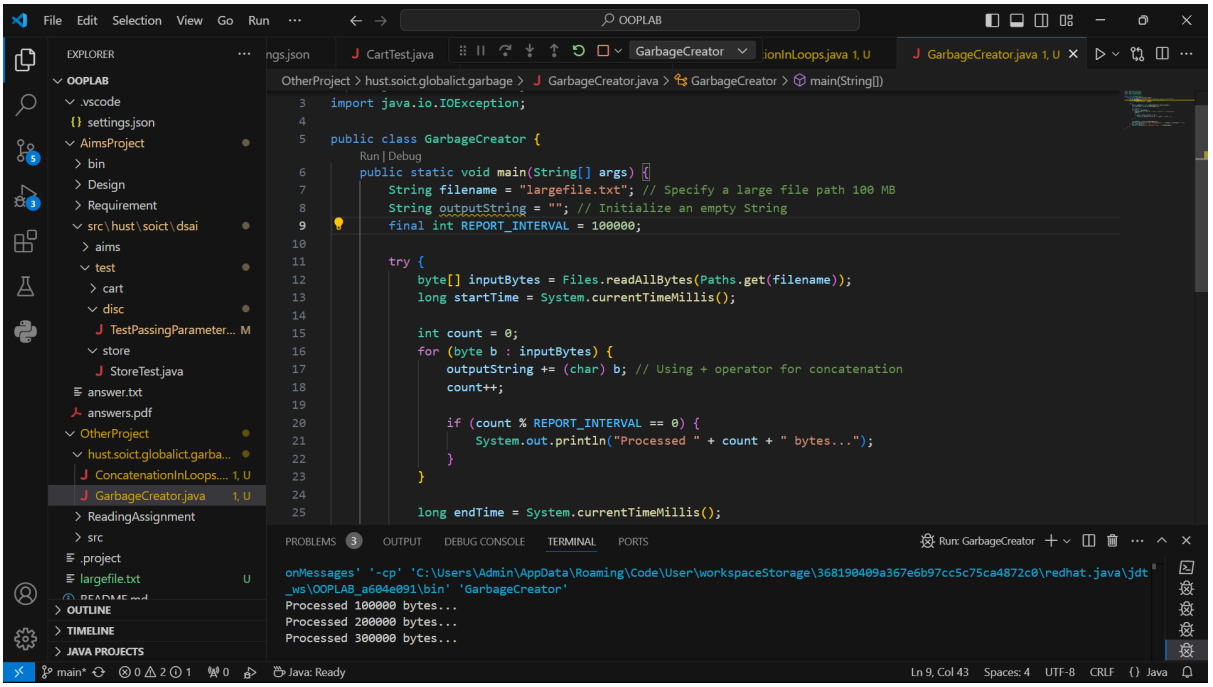
```java
package hust.soict.dsai.aims.store;
import hust.soict.dsai.aims.disc.DigitalVideoDisc;

public class Store {
    private DigitalVideoDisc[] itemsInStore; // Array to store DigitalVideoDiscs
    private int currentCount;   // Tracks the number of DigitalVideoDiscs in the store

    // Constructor to initialize the store with a given capacity
    public Store(int capacity) {
        itemsInStore = new DigitalVideoDisc[capacity];
        currentCount = 0;
    }

    public void addDigitalVideoDisc(DigitalVideoDisc DigitalVideoDisc) {
        if (currentCount < itemsInStore.length) {
            itemsInStore[currentCount] = DigitalVideoDisc;
            currentCount++;
            System.out.println("Added DigitalVideoDisc: " + DigitalVideoDisc.toString());
        } else {
            System.out.println(x:"Store is full. Cannot add more DigitalVideoDiscs.");
        }
    }

    public void removeDigitalVideoDisc(int id) {
```

```java
    public void removeDigitalVideoDisc(int id) {
        boolean found = false;
        for (int i = 0; i < currentCount; i++) {
            if (itemsInStore[i].getId() == id) { // Assuming DigitalVideoDisc has a getId() method
                found = true;
                // Shift all elements after the removed DigitalVideoDisc
                for (int j = i; j < currentCount - 1; j++) {
                    itemsInStore[j] = itemsInStore[j + 1];
                }
                itemsInStore[currentCount - 1] = null; // Clear the last slot
                currentCount--;
                System.out.println("Removed DigitalVideoDisc with ID: " + id);
                break;
            }
        }
        if (!found) {
            System.out.println("DigitalVideoDisc with ID " + id + " not found in the store.");
        }
    }

    // Method to display all DigitalVideoDiscs in the store
    public void displayStore() {
        System.out.println(x:"DigitalVideoDiscs currently in the store:");
        for (int i = 0; i < currentCount; i++) {
            System.out.println(itemsInStore[i].toString());
        }
    }
}
```

- StoreTest class

```java
package hust.soict.dsai.test.store;
import hust.soict.dsai.aims.disc.DigitalVideoDisc;
import hust.soict.dsai.aims.store.Store;

public class StoreTest {
    Run | Debug
    public static void main(String[] args) {
        // Create a store with a capacity of 5
        Store store = new Store(capacity:5);

        // Create some DigitalVideoDiscs
        DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Liems King", category:"Animation", director:"Roger Al
        DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star Wars", category:"Science Fiction", director:"George
        DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"Aladin",category:"Animation", cost:18.99f);

        // Test adDigitalVideoDiscng DigitalVideoDiscs
        store.addDigitalVideoDisc(dvd1);
        store.addDigitalVideoDisc(dvd2);
        store.addDigitalVideoDisc(dvd3);

        // DigitalVideoDiscsplay the store
        store.displayStore();;

        // Test removing a DigitalVideoDisc
        store.removeDigitalVideoDisc(id:2);

        // DigitalVideoDiscsplay the store again
        store.displayStore();

    }
}
```

9. String, StringBuilder and StringBuffer
    - use String took a lot of time can't even wait for the result



- using stringBuffer and stringBuilder is way faster