BÁO CÁO THỰC HÀNH LAB 3
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

Table of Figures

I. Mã nguồn chương trình:

1. Branch your respository:

2. Method overloading:

```
//overload phuong thuc addDigitalVideoDisc khac kieu tham so
no usages  tuandattt
public void addDigitalVideoDisc(DigitalVideoDisc [] dvdList){
    for(DigitalVideoDisc disc: dvdList){
        addDigitalVideoDisc(disc);
    }
}

//overload phuong thuc addDigitalVideoDisc voi nhieu tham so
no usages  tuandattt
public void addDigitalVideoDisc(DigitalVideoDisc disc1, DigitalVideoDisc disc2){
    addDigitalVideoDisc(disc1);
    addDigitalVideoDisc(disc2);
}
```

Figure 1: Method overloading

3. Passing Parameter:

```
package Lab_02;

tuandattt *
public class TestPasssingParameter {
    tuandattt *
    public static void main(String[] args){
        //TODO Auto-generated methor 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());


        //doi title cua 2 dvd
        changeTitle(jungleDVD, cinderellaDVD.getTitle());
        System.out.println("jungle dvd title: " + jungleDVD.getTitle());
    }

    no usages  tuandattt
    public static void swap(Object o1, Object o2){
        Object tmp = o1;
        o1 = o2;
        o2 = tmp;
    }

    1 usage  tuandattt
    public static void changeTitle(DigitalVideoDisc disc, String title){
        String oldTitle = disc.getTitle();
        disc.setTitle(title);
        disc = new DigitalVideoDisc(oldTitle);
```

Figure 2: TestPassingParameter class

```java
// viet lai phuong thuc swap
2 usages  tuandattt
public static void swap(DigitalVideoDisc o1, DigitalVideoDisc o2){
    String title = o2.getTitle();
    o2.setTitle(o1.getTitle());
    o1.setTitle(title);
}

}
```

Figure 3: Phương thức swap mới

```java
//test phuong thuc swap moi
swap(jungleDVD, cinderellaDVD);
System.out.println("Sau khi swap: ");
System.out.println("ungle DVD title: " + jungleDVD.getTitle());
System.out.println("cinderella DVD title: " + cinderellaDVD.getTitle());
```

Figure 4: Chạy thử phương thức swap mới

4. Use debug run:

```java
14        //test phuong thuc swap moi
          swap(jungleDVD, cinderellaDVD);
16        System.out.println("Sau khi swap: ");
17        System.out.println("ungle DVD title: " + jungleDVD.getTitle());
18        System.out.println("cinderella DVD title: " + cinderellaDVD.getTitle());
19
```

Figure 5: Thêm breakpoint

```java
14        //test phuong thuc swap moi
15        swap(jungleDVD, cinderellaDVD);
16        System.out.println("Sau khi swap: ");
17        System.out.println("ungle DVD title: " + jungleDVD.getTitle());
18        System.out.println("cinderella DVD title: " + cinderellaDVD.getTitle());
19
```

Figure 6: Xóa checkpoint

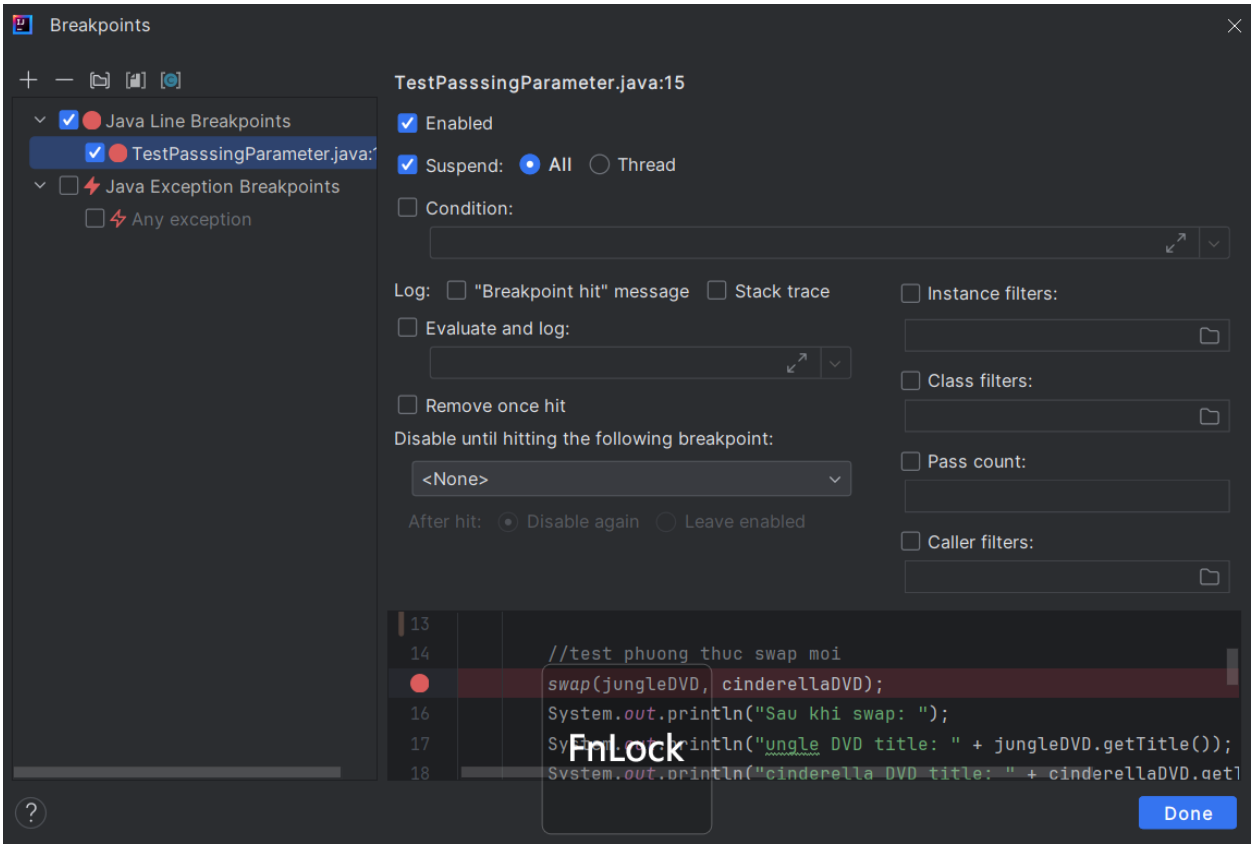


Figure 7: Breakpoint view

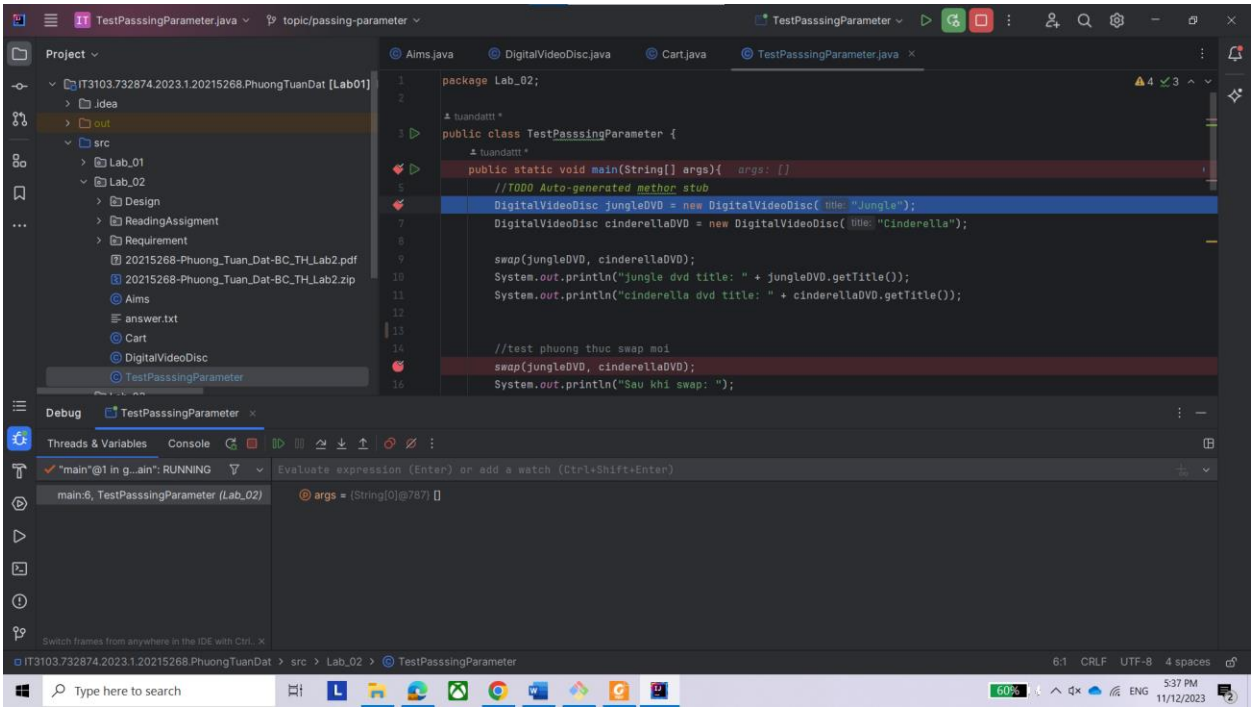


Figure 8: Debug mode

Figure 9: Biến trong Variables View

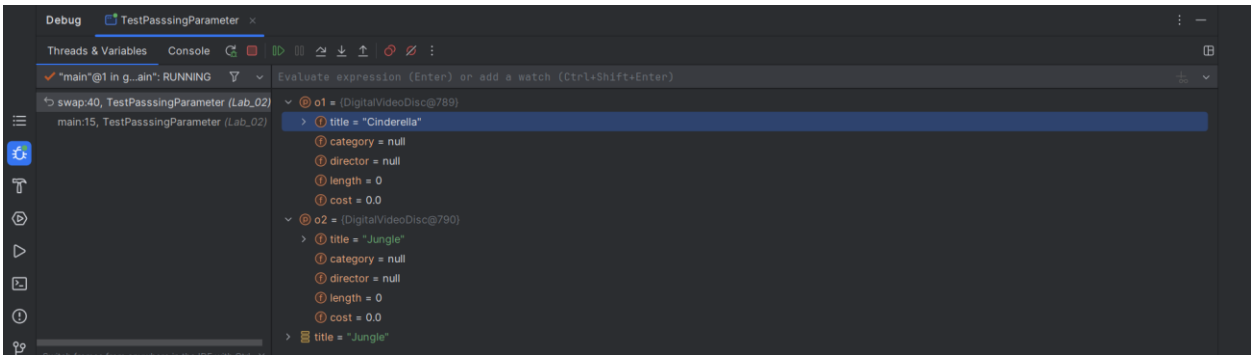


Figure 10: Step over dòng 39

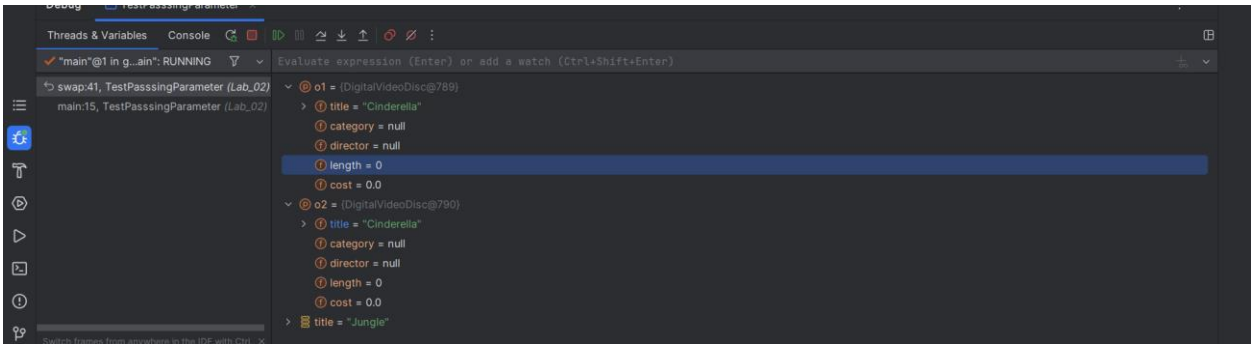


Figure 11: Step over dòng 40

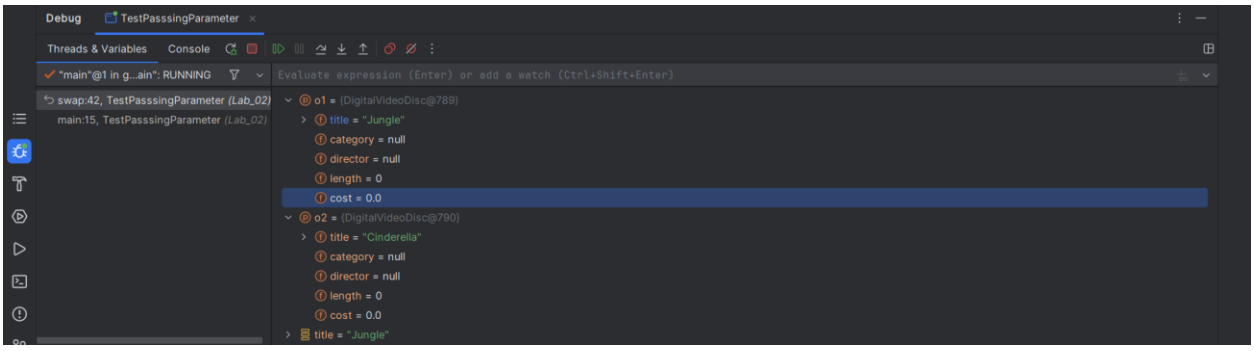


Figure 12: Step over dòng 41

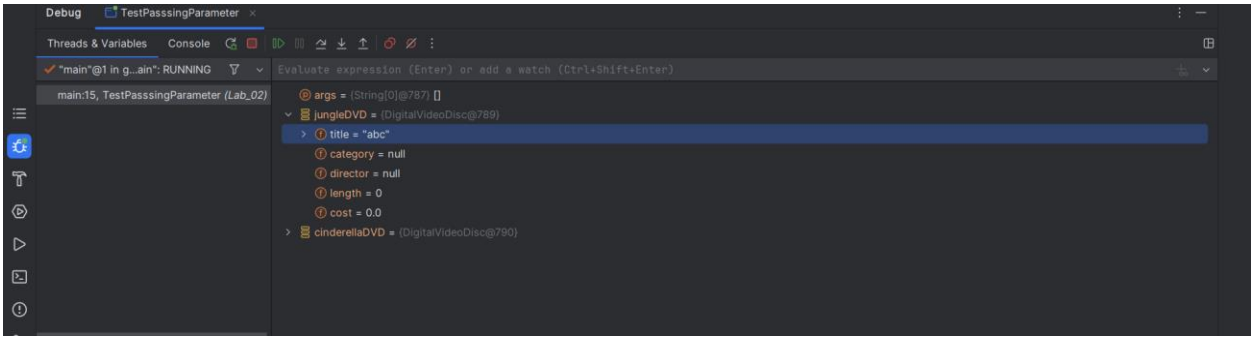


Figure 13: Sửa giá trị biến trong debug mode

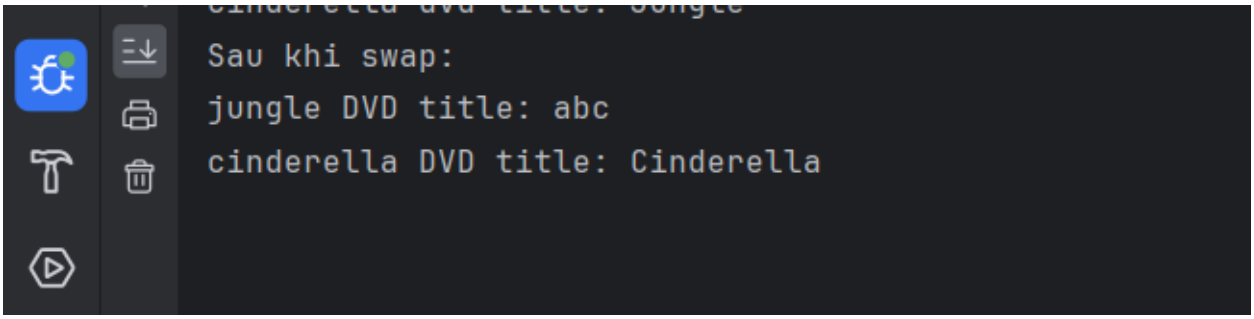


Figure 14: Kết quả trên console

5. Classifier Member and Instance Member:

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

1 usage  tuandattt
public DigitalVideoDisc(String title, String category, float cost) {
    nbDigitalVideoDiscs++;
    this.title = title;
    this.category = category;
    this.cost = cost;
    this.id = nbDigitalVideoDiscs;
}

no usages  tuandattt
public DigitalVideoDisc(String title, String category, String director, float cost) {
    nbDigitalVideoDiscs++;
    this.title = title;
    this.category = category;
    this.director = director;
    this.cost = cost;
    this.id = nbDigitalVideoDiscs;
}

2 usages  tuandattt
public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
    nbDigitalVideoDiscs++;
    this.title = title;
    this.category = category;
    this.director = director;
    this.length = length;
```

Figure 15: Class member và Object member

6. Open Cart Class:

```java
//Phuong thuc print
no usages  new *
public void print(){
    System.out.println("**********CART**********");
    for(int i = 0; i < qtyOrdered; i++){
        System.out.println("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("**************************************************");
}
```

Figure 16: Phương thức Print

```java
package Lab_02;

new *
public class CartTest {
    new *
    public static void main(String[] args){
        //Create new cart
        Cart cart = new Cart();

        //Create new dvd objects and add them to the cart
        DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King", category: "Animation", director: "Roger Allers",
        cart.addDigitalVideoDisc(dvd1);

        DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star War", category: "Science Fiction", director: "George Lucas",
        cart.addDigitalVideoDisc(dvd2);

        DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladin", category: "Animation", cost: 18.95f);
        cart.addDigitalVideoDisc(dvd3);

        //test print method
        cart.print();
    }
}
```

Figure 17: CartTest Class test print method

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

Figure 18: Phương thức toString

```java
1 usage  tuandattt
public boolean isMatch(String title){
    if(title.equals(this.title)) return true;
    else {
        return false;
    }
}
```

Figure 19: Phương thức boolean isMatch

```java
1 usage  tuandattt
public void search(int id){
    boolean found = false;
    for(int i = 0; i < qtyOrdered; i++){
        if(itemsOrdered[i].getId() == id) {
            System.out.println(itemsOrdered[i]);
            found = true;
        }
    }
    if(!found){
        System.out.println("No match is found");
    }
}

1 usage  tuandattt
public void search(String title){
    boolean found = false;
    for(int i = 0; i < qtyOrdered; i++){
        if(itemsOrdered[i].isMatch(title)){
            System.out.println(itemsOrdered[i]);
            found = true;
        }
    }
    if(!found){
        System.out.println("No match is found");
    }
}
```

Figure 20: Phương thức search theo title và id

```java
        //test search by id method
        System.out.println("Search by id:");
        cart.search( id: 1);
        //test search by title method
        System.out.println("Search by title:");
        cart.search( title: "Aladin");
    }
}
```

Figure 21: CartTest class test tìm kiếm theo title và id

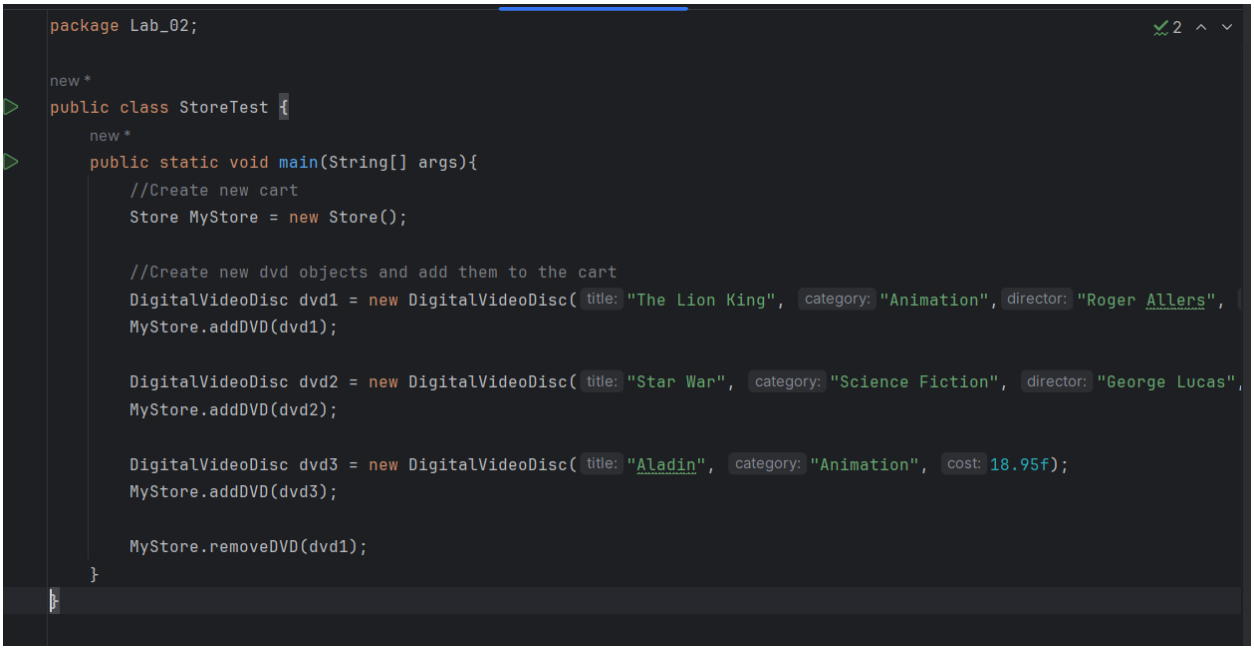
7. Implement the Store Class:

```
2 usages  new *
public class Store {
    //Danh sach dia trong cua hang
    5 usages
    private DigitalVideoDisc itemsInStore[] = new DigitalVideoDisc[100];
    //So luong dia trong cua hang
    6 usages
    int numberItem;

    //Them dia vao danh sach cua cua hang
    3 usages  new *
    public void addDVD(DigitalVideoDisc disc){
        itemsInStore[numberItem] = disc;
        numberItem++;
        System.out.println("Added " + disc.getTitle() + " to the store");
    }
```

```
    1 usage  new *
    public void removeDVD(DigitalVideoDisc disc){
        boolean found = false;
        for(int i = 0; i < numberItem; i++){
            if(itemsInStore[i] == disc){
                //lùi dvd tìm thấy xuống cuối danh sách
                for(int j = i; j < numberItem - 1; j++){
                    itemsInStore[j] = itemsInStore[j+1];
                }
                //xóa dvd tìm được
                itemsInStore[numberItem - 1] = null;
                //cập nhật số lượng dvd
                numberItem--;
                //cập nhật đã tìm thấy dvd muốn xóa
                found = true;
                System.out.println("Removed " + disc.getTitle() + " from the store");
                break;
            }
        }
        if(!found){
            System.out.println("Cannot find " + disc.getTitle());
        }
    }
```

Figure 22: Store Class



```
package Lab_02;

new *
public class StoreTest {
    new *
    public static void main(String[] args){
        //Create new cart
        Store MyStore = new Store();

        //Create new dvd objects and add them to the cart
        DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King", category: "Animation", director: "Roger Allers",
        MyStore.addDVD(dvd1);

        DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star War", category: "Science Fiction", director: "George Lucas",
        MyStore.addDVD(dvd2);

        DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladin", category: "Animation", cost: 18.95f);
        MyStore.addDVD(dvd3);

        MyStore.removeDVD(dvd1);
    }
}
```

Figure 23: Chạy thử phương thức addDVD và removeDVD

8. Re-organize projects:

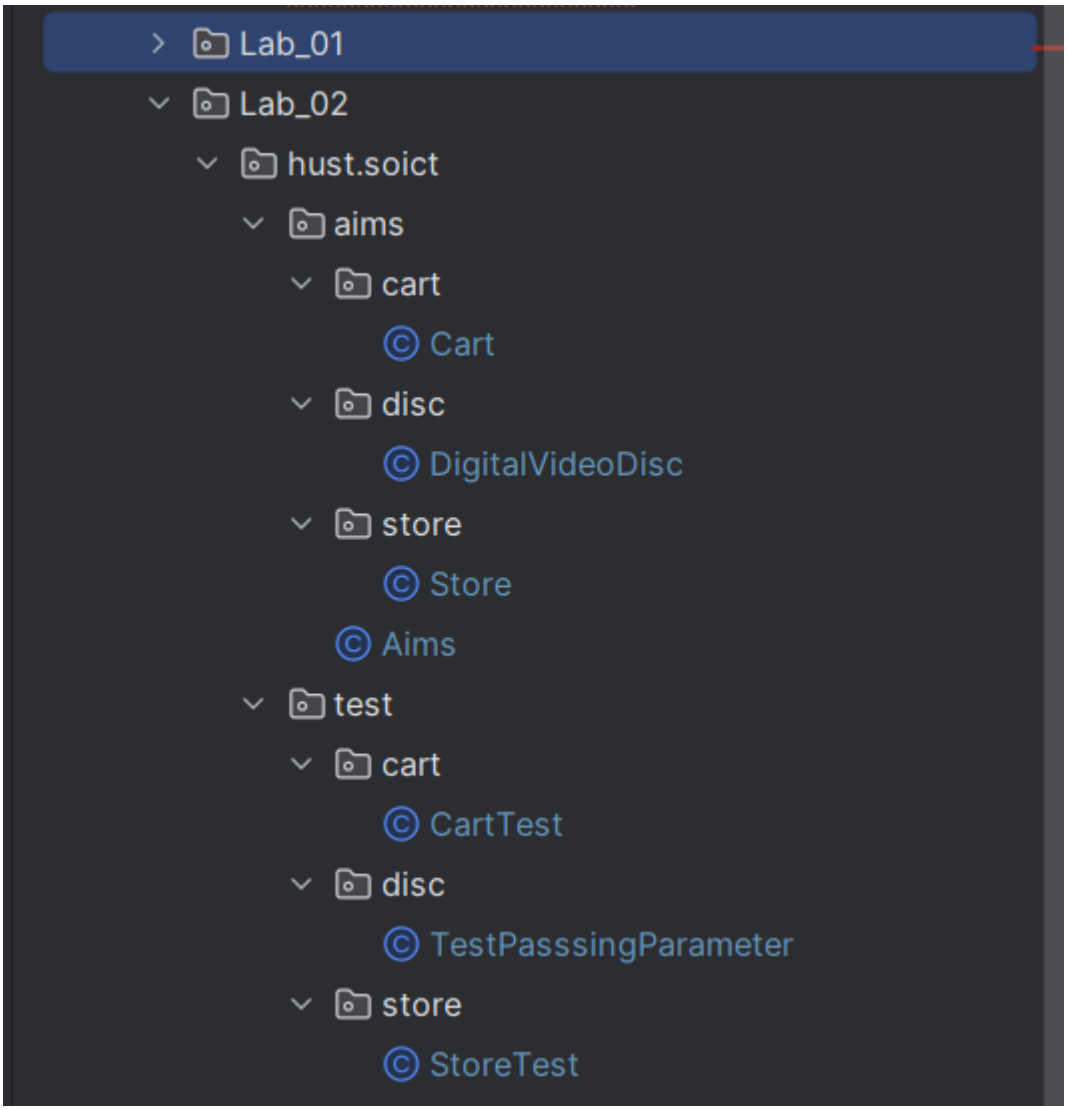


Figure 24: Sắp xếp các Class

9. String, StringBuilder and StringBuffer:

```java
package hust.soict.garbage;

import java.util.Random;

new *
public class ConcatenationInLoops {
    new *
    public static void main(String[] args){
        Random r = new Random( seed: 123);
        long start = System.currentTimeMillis();
        String s = "";
        for (int i = 0; i < 65536; i++){
            s += r.nextInt( bound: 2);
        }
        System.out.println(System.currentTimeMillis() - start); //print 4500

        r = new Random( seed: 123);
        start = System.currentTimeMillis();
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < 65536; i++)
            sb.append(r.nextInt( bound: 2));
        s = sb.toString();
        System.out.println(System.currentTimeMillis() - start); //print 5
    }
}
```

Figure 25: ConcatenationInLoops Class

```java
package hust.soict.garbage;

import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.io.IOException;

no usages  new *
public class GarbageCreator {
    no usages  new *
    public void createGarbage(){
        String filename = "test.exe";
        byte[] inputBytes = { 0 };
        long startTime, endTime;

        try {
            inputBytes = Files.readAllBytes(Paths.get(filename));
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
        startTime = System.currentTimeMillis();
        String outputString = "";
        for(byte b : inputBytes){
            outputString += (char)b;
        }
        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }


}
```

Figure 26: GarbageCreator Class

```java
public class NoGarbage {
    3 usages
    private StringBuilder outputStringBuilder;

    no usages  new *
    public NoGarbage() {
        outputStringBuilder = new StringBuilder();
    }

    no usages  new *
    public void createGarbage() {
        String filename = "test.exe";
        byte[] inputBytes = { 0 };
        long startTime, endTime;

        try {
            inputBytes = Files.readAllBytes(Paths.get(filename));
        } catch (IOException e) {
            throw new RuntimeException(e);
        }

        startTime = System.currentTimeMillis();

        outputStringBuilder.setLength(0); // Clear the StringBuilder for reuse

        for (byte b : inputBytes) {
            outputStringBuilder.append((char) b);
        }

        endTime = System.currentTimeMillis();

        System.out.println(endTime - startTime);
```

Figure 27: NoGarbage Class

```java
package hust.soict.garbage;

import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.io.IOException;

no usages  new *
public class GarbageCreator {
    no usages  new *
    public void createGarbage(){
        String filename = "test.exe";
        byte[] inputBytes = { 0 };
        long startTime, endTime;

        try {
            inputBytes = Files.readAllBytes(Paths.get(filename));
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
        startTime = System.currentTimeMillis();
        //Su dung StringBuffer
        StringBuilder outputStringBuilder = new StringBuilder();
        //String outputString = "";
        for(byte b : inputBytes){
            outputStringBuilder.append((char)b);
        }
        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }
```

Figure 28: CreatorGarbage Class sử dụng StringBuffer

II. Demo chương trình:

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


Process finished with exit code 0
```

Figure 29: Kết quả PassingParameter

```
jungle dvd title: Cinderella
cinderella dvd title: Jungle
Sau khi swap:
ungle DVD title: Jungle
cinderella DVD title: Cinderella
```

Figure 30: Kết quả PassingParameter sau swap

```
**********CART**********
DVD - The Lion King - Animation - Roger Allers - 87: 19.95 $
Total Cost: 63.850002
DVD - Star War - Science Fiction - George Lucas - 87: 24.95 $
Total Cost: 63.850002
DVD - Aladin - Animation - null - 0: 18.95 $
Total Cost: 63.850002
**************************************************
```

Figure 31: Kết quả print-cart

```
Search by id:
DVD - The Lion King - Animation - Roger Allers - 87: 19.95 $
Search by title:
DVD - Aladin - Animation - null - 0: 18.95 $
```

Figure 32: Kết quả search-cart

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-
Added The Lion King to the store
Added Star War to the store
Added Aladin to the store
Removed The Lion King from the store

Process finished with exit code 0
```

Figure 33: Kết quả TestStore Class

III. Class Diagram

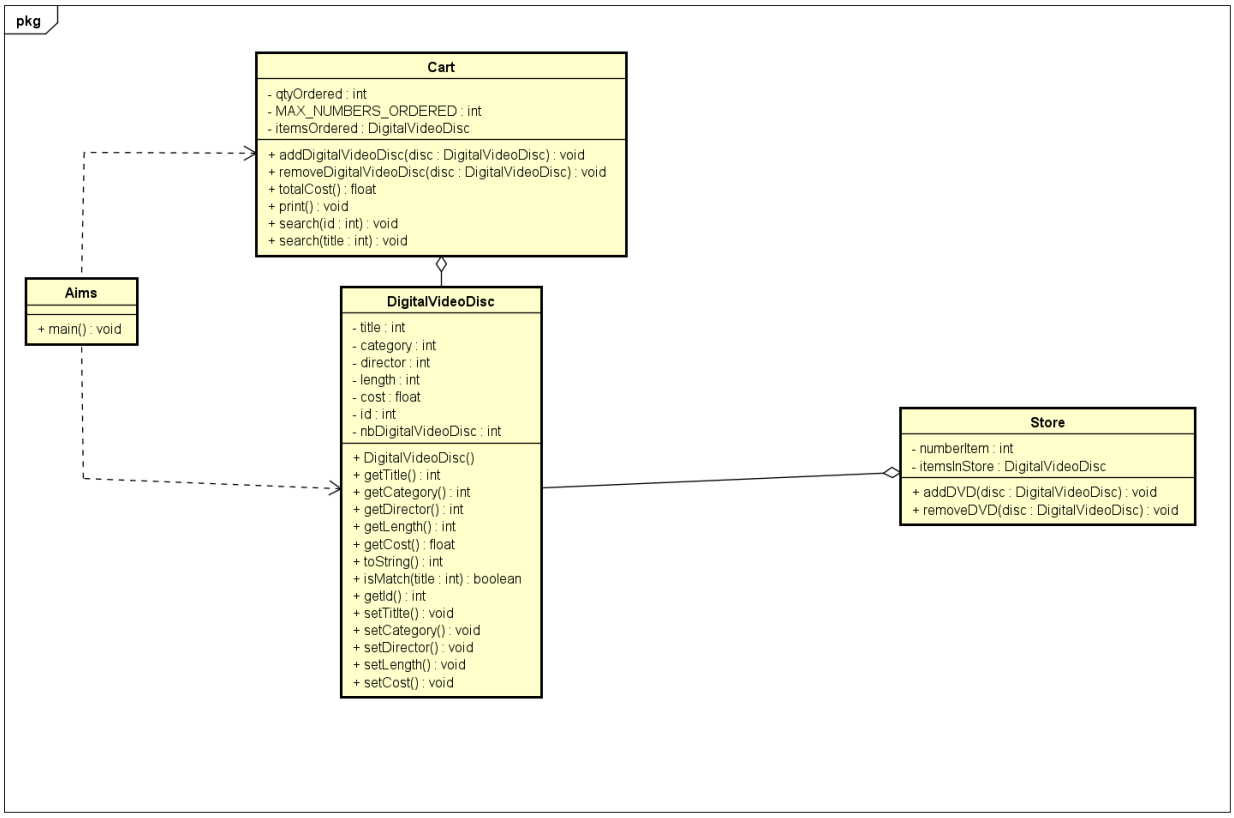


Figure 34: Update Class Diagram