

# Report Lab 03

Student: Pham Quang Huy – ID: 20215207

## Method Overloading

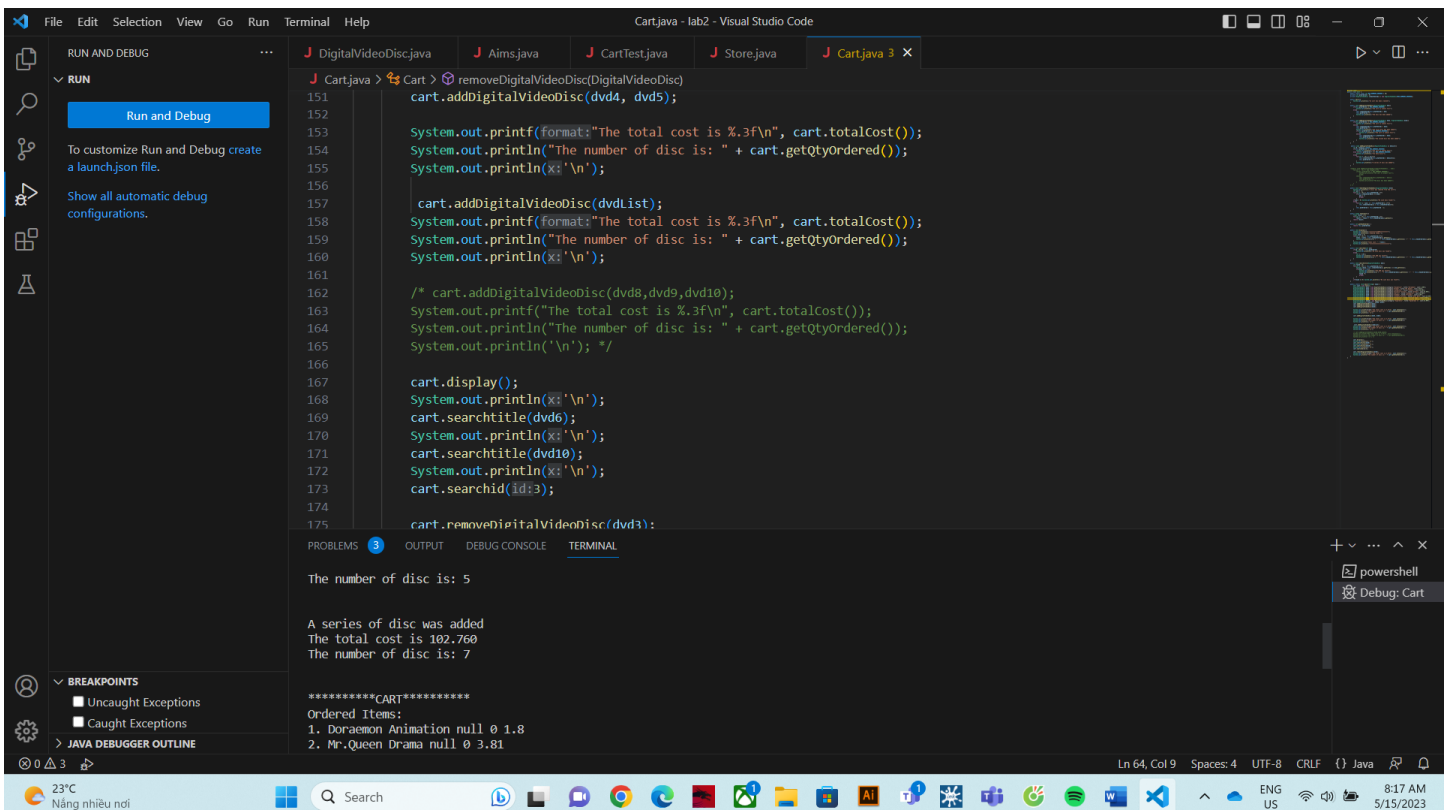
### 1. Overloading with different types of parameter

Add a list of DVD into the cart:

Source code:

```
public void addDigitalVideoDisc(DigitalVideoDisc [] dvdList){
    int N = dvdList.length;
    if(this.qtyOrdered >= MAX_NUMBERS_ORDERED)
        System.out.println(x:"The cart is almost full");
    else if(this.qtyOrdered + N > MAX_NUMBERS_ORDERED)
        System.out.println(x:"Too many discs");
    else{
        for(int i = 0;i < N;i++){
            this.itemsOrdered[this.qtyOrdered] = dvdList[i];
            this.qtyOrdered ++;
        }
        System.out.println(x:"A series of disc was added");
    }
}
```

Running result:



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like DigitalVideoDisc.java, Aims.java, CartTest.java, Store.java, and Cart.java 3.
- Code Editor:** Displays the code for Cart.java, showing the addDigitalVideoDisc method and its usage in the main method.
- Terminal:** Shows the output of the program, including the number of discs added and the total cost.
- Output:** Displays the output of the program, showing the number of discs added and the total cost.

The output in the terminal is as follows:

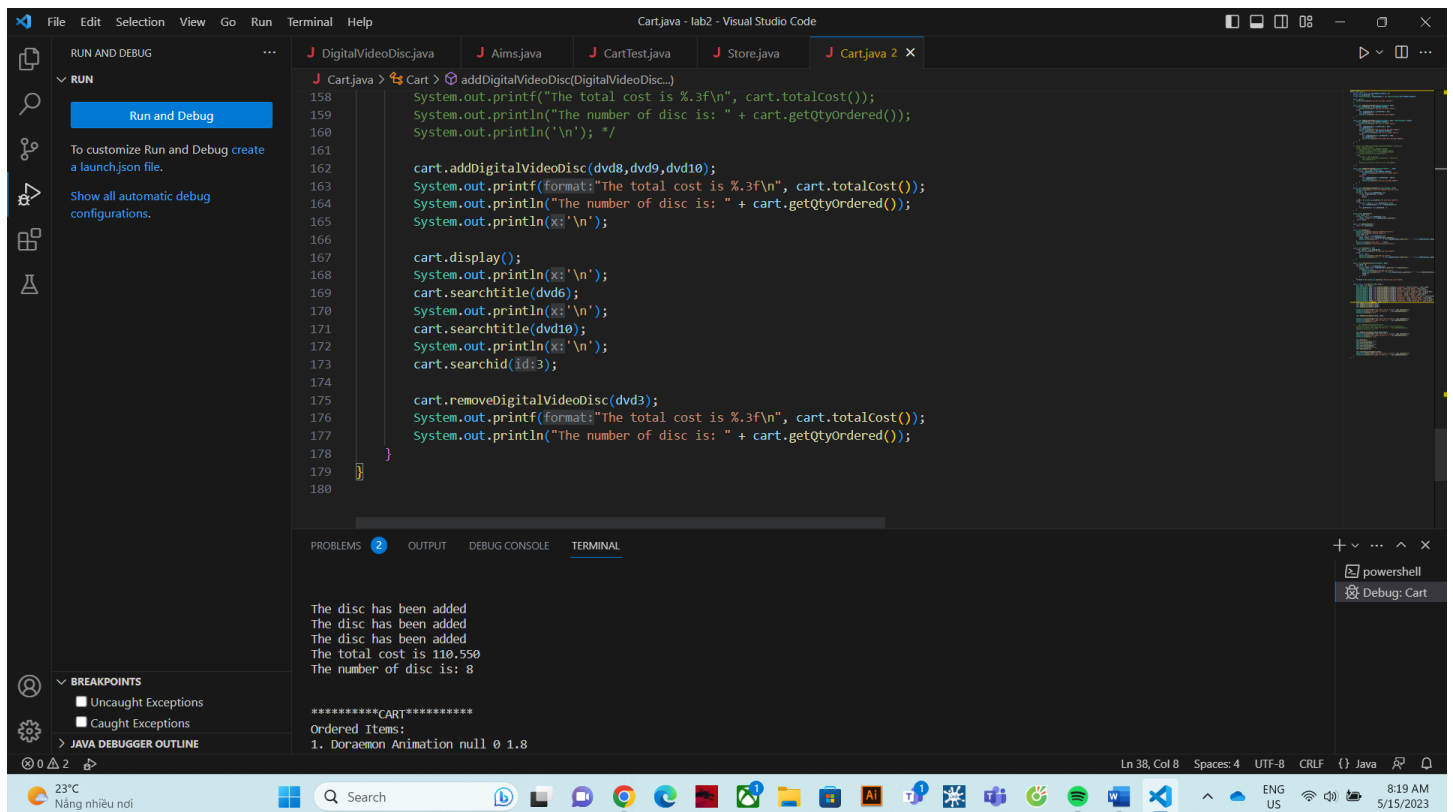
```
The number of disc is: 5
A series of disc was added
The total cost is 102.760
The number of disc is: 7
*****CART*****
Ordered Items:
1. Doraemon Animation null 0 1.8
2. Mr.Queen Drama null 0 3.81
```

Add an arbitrary number of DVD into the cart

Source code:

```
public void addDigitalVideoDisc(DigitalVideoDisc... dvd){
    for(int i = 0; i < dvd.length; i++){
        if(this.qtyOrdered >= MAX_NUMBERS_ORDERED){
            System.out.println(x:"The cart is almost full");
            break;
        }else{
            this.itemsOrdered[this.qtyOrdered] = dvd[i];
            this.qtyOrdered ++;
            System.out.println(x:"The disc has been added");
        }
    }
}
```

Running result:



```
File Edit Selection View Go Run Terminal Help
Cart.java - lab2 - Visual Studio Code
RUN AND DEBUG
RUN
To customize Run and Debug create a launch.json file.
Show all automatic debug configurations.
BREAKPOINTS
Uncaught Exceptions
Caught Exceptions
JAVA DEBUGGER OUTLINE
J DigitalVideoDisc.java J Aims.java J CartTest.java J Store.java J Cart.java 2 x
J Cart.java > Cart > addDigitalVideoDisc(DigitalVideoDisc...)
158 System.out.printf("The total cost is %.3f\n", cart.totalCost());
159 System.out.println("The number of disc is: " + cart.getQtyOrdered());
160 System.out.println('\n');
161
162 cart.addDigitalVideoDisc(dvd8,dvd9,dvd10);
163 System.out.printf(format:"The total cost is %.3f\n", cart.totalCost());
164 System.out.println("The number of disc is: " + cart.getQtyOrdered());
165 System.out.println(x:'\n');
166
167 cart.display();
168 System.out.println(x:'\n');
169 cart.searchtitle(dvd6);
170 System.out.println(x:'\n');
171 cart.searchtitle(dvd10);
172 System.out.println(x:'\n');
173 cart.searchid(id3);
174
175 cart.removeDigitalVideoDisc(dvd3);
176 System.out.printf(format:"The total cost is %.3f\n", cart.totalCost());
177 System.out.println("The number of disc is: " + cart.getQtyOrdered());
178
179
180
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL
The disc has been added
The disc has been added
The disc has been added
The total cost is 110.550
The number of disc is: 8
*****CART*****
Ordered Items:
1. Doraemon Animation null 0 1.8
Ln 38, Col 8 Spaces: 4 UTF-8 CRLF () Java 8:19 AM 5/15/2023
```

## 2. Overloading by differing the number of parameters

Source code:

```

public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2){
    if(this.qtyOrdered >= MAX_NUMBERS_ORDERED){
        System.out.println(x:"The cart is almost full");
    }else{
        this.itemsOrdered[this.qtyOrdered] = dvd1;
        this.qtyOrdered ++;
        System.out.println(x:"The first disc has been added");
        if(this.qtyOrdered >= MAX_NUMBERS_ORDERED){
            System.out.println(x:"The cart is almost full");
        }else{
            this.itemsOrdered[this.qtyOrdered] = dvd1;
            this.qtyOrdered ++;
            System.out.println(x:"The second disc has been added");
        }
    }
}
}

```

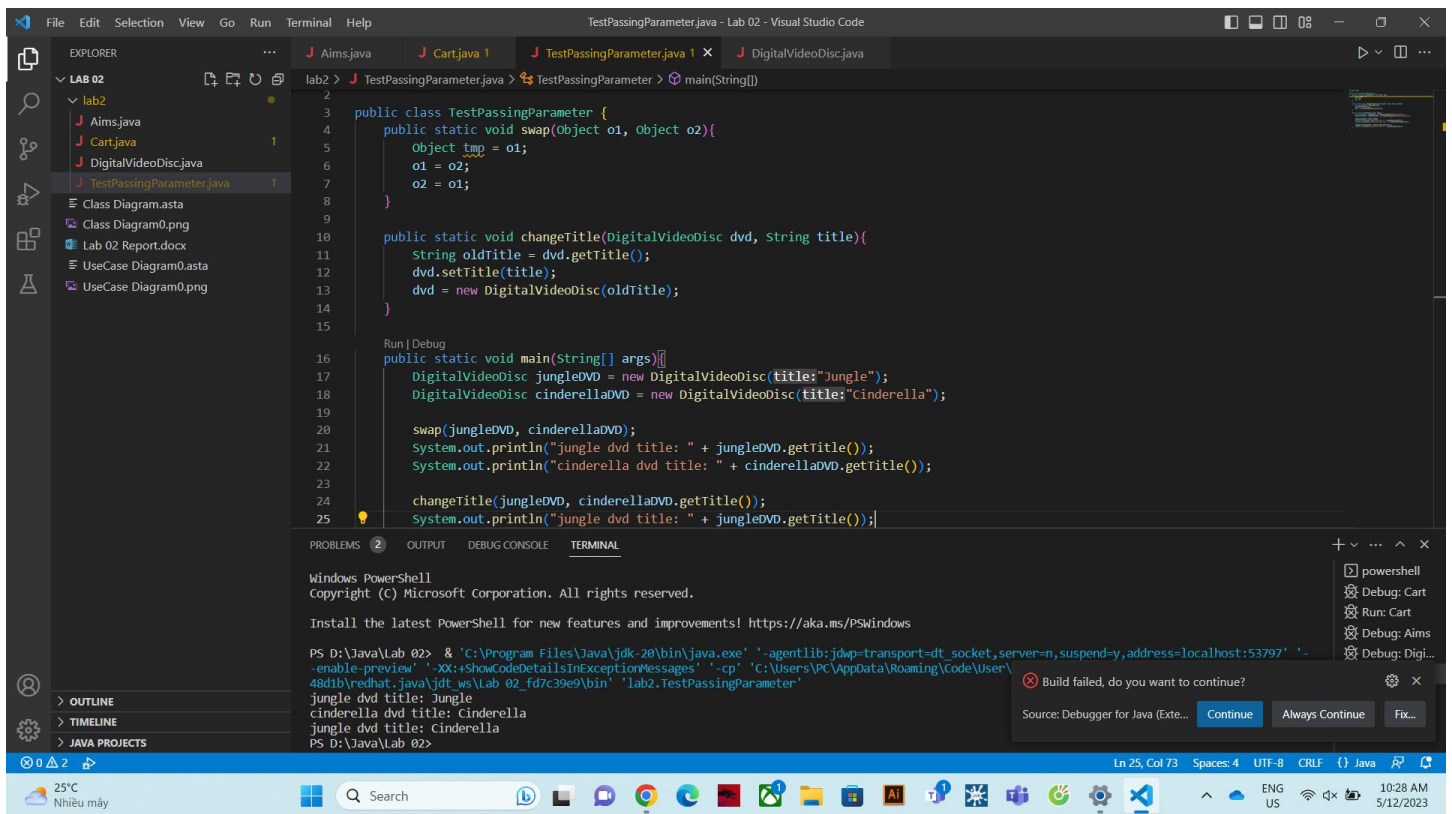
Running result:

## Passing by references or Passing by values:

Question: Is JAVA a Pass by Value or a Pass by Reference programming language?

Answer: JAVA is a Pass by Value only programming language.

Swapping functions: Source code and running result

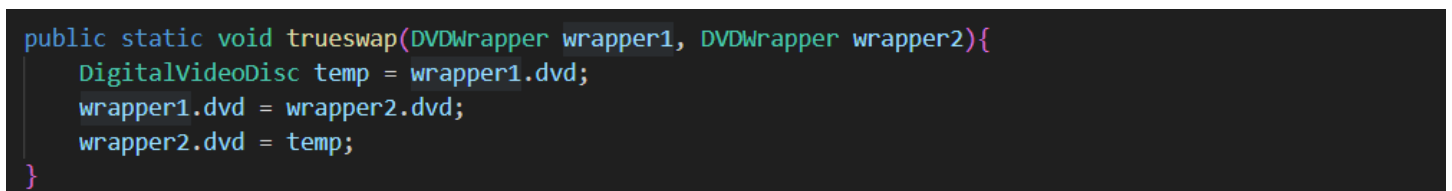


True swapping:

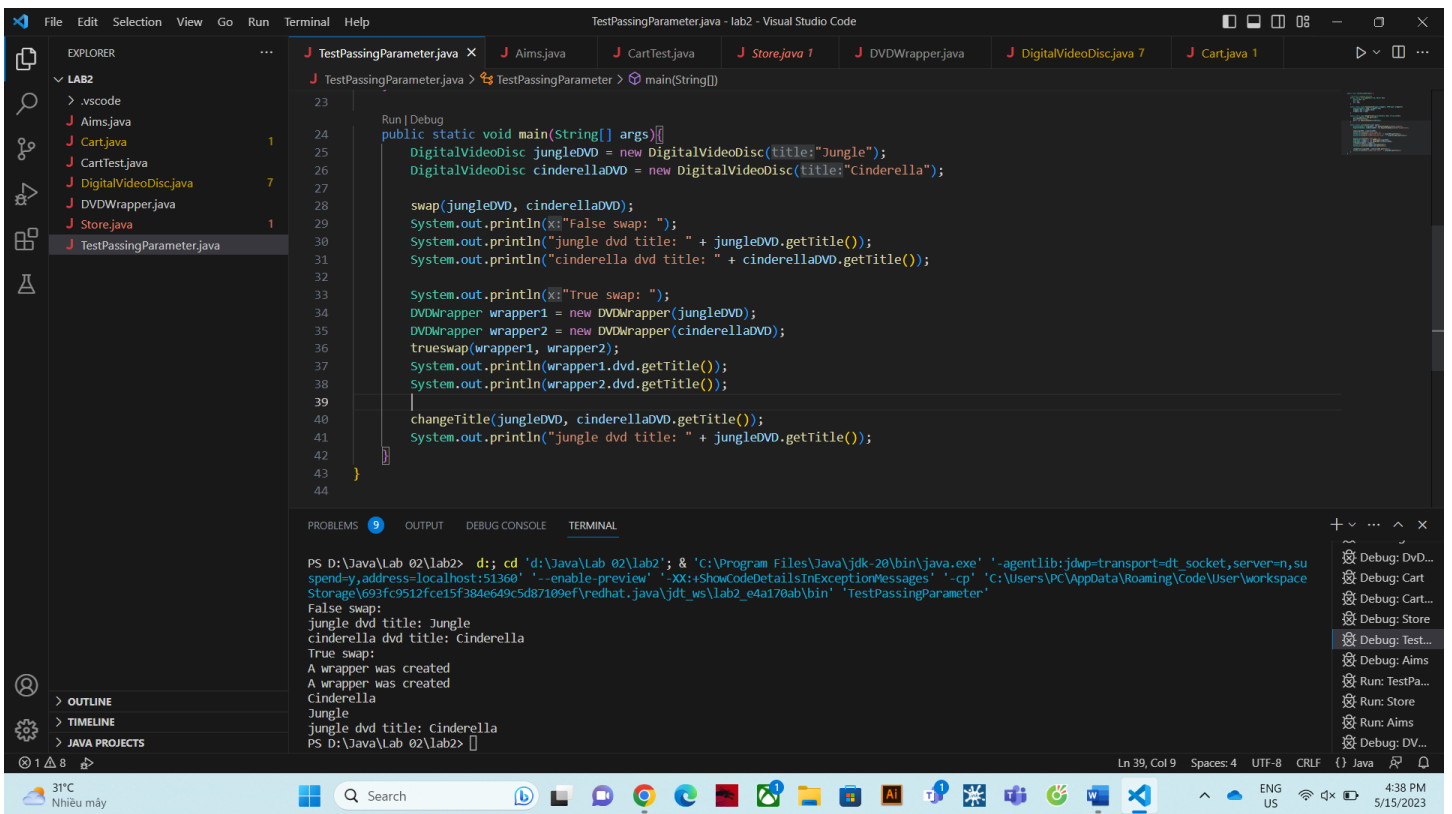
First, I create a wrapper class:



Swapping function:



Running result:



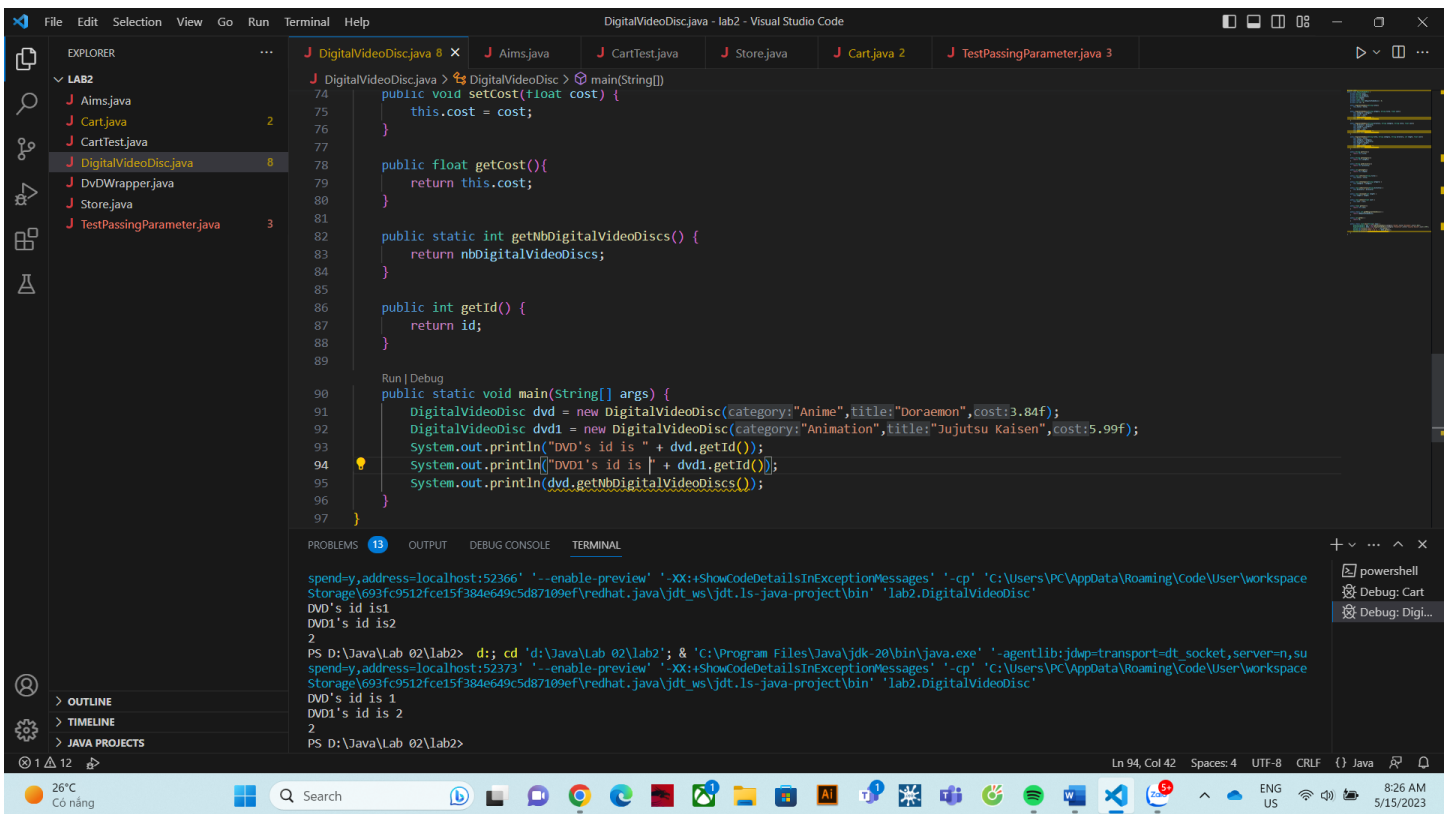
**Classifier members and Instance members:**

Updated DigitalVideoDisc class:

```

package lab2;
public class DigitalVideoDisc {
    private String title;
    private String category;
    private String directory;
    private int length;
    private float cost;
    private static int nbDigitalVideoDiscs = 0;
    private int id = 0;
}

```



## Display methods and Search methods:

### Display methods:

```
public void display(){
    System.out.println(x:"*****CART*****");
    System.out.println(x:"Ordered Items:");
    float total = 0;
    for(int i = 0;i < this.qtyOrdered;i++){
        total = total + this.itemsOrdered[i].getCost();
        System.out.println((i+1) + ". " + this.itemsOrdered[i].getTitle() + " " + this.itemsOrdered[i].getCategory() + " " + this.itemsOrdered[i].getCost());
    }
    System.out.println("Total cost: " + total);
    System.out.println(x:"*****");
}
```

### Search methods:

```
public void searchid(int id){
    if(id < 0||id > this.qtyOrdered)
        System.out.println(x:"No such disc was found");
    else{
        int i = id-1;
        System.out.println(x:"The DVD was found");
        System.out.println((i+1) + ". " + this.itemsOrdered[i].getTitle() + " " + this.itemsOrdered[i].getCategory() + " " + this.itemsOrdered[i].getCost());
    }
}

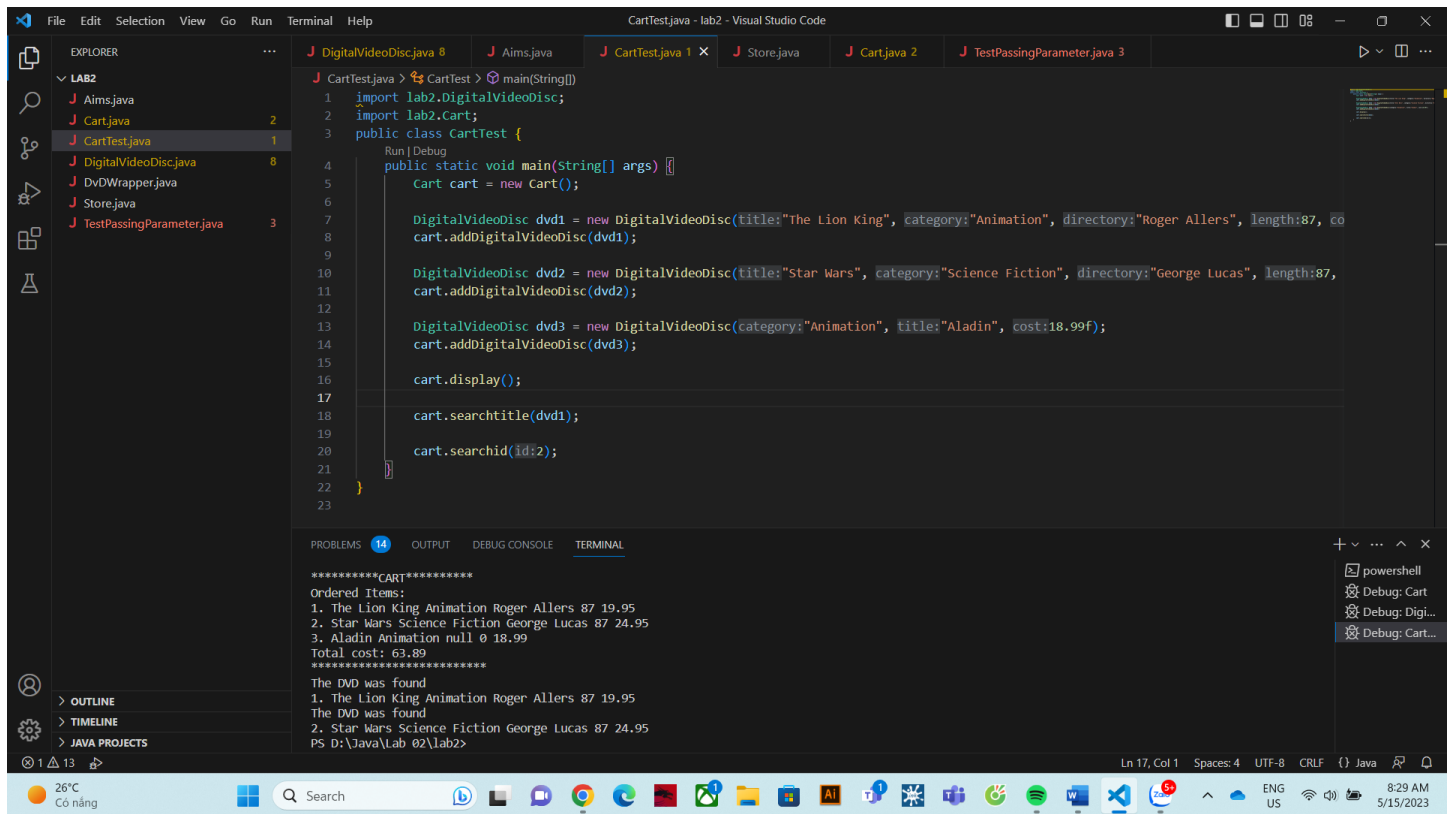
public void searchtitle(DigitalVideoDisc dvd){
    int found = 0;
    for(int i = 0;i < this.qtyOrdered;i++){
        boolean check = this.itemsOrdered[i].getTitle() == dvd.getTitle();
        if(check == true){
            System.out.println(x:"The DVD was found");
            System.out.println((i+1) + ". " + this.itemsOrdered[i].getTitle() + " " + this.itemsOrdered[i].getCategory() + " " + this.itemsOrdered[i].getCost());
            found ++;
            break;
        }
    }
}
```

### Testing both methods:

Source codes:

```
1 import lab2.DigitalVideoDisc;
2 import lab2.Cart;
3 public class CartTest {
4     public static void main(String[] args) {
5         Cart cart = new Cart();
6
7         DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion King", category:"Animation", directory:"Roger Allers", length:87, cost:19.95f);
8         cart.addDigitalVideoDisc(dvd1);
9
10        DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star Wars", category:"Science Fiction", directory:"George Lucas", length:87, cost:24.95f);
11        cart.addDigitalVideoDisc(dvd2);
12
13        DigitalVideoDisc dvd3 = new DigitalVideoDisc(category:"Animation", title:"Aladin", cost:18.99f);
14        cart.addDigitalVideoDisc(dvd3);
15
16        cart.display();
17
18        cart.searchtitle(dvd1);
19
20        cart.searchid(id:2);
21    }
22 }
```

Running results:



```
*****CART*****
Ordered Items:
1. The Lion King Animation Roger Allers 87 19.95
2. Star Wars Science Fiction George Lucas 87 24.95
3. Aladin Animation null 0 18.99
Total cost: 63.89
*****

The DVD was found
1. The Lion King Animation Roger Allers 87 19.95
The DVD was found
2. Star Wars Science Fiction George Lucas 87 24.95
PS D:\Java\Lab 02\lab2>
```

Implementing the Store class:

Source codes:

```

import lab2.DigitalVideoDisc;
public class Store {
    public static final int MAX = 200;
    private int qtyStored = 0;
    private DigitalVideoDisc itemsInStore[] = new DigitalVideoDisc[MAX];

    public void addDvd(DigitalVideoDisc dvd){
        if(this.qtyStored >= MAX)
            System.out.println(x:"The store is full");
        else{
            this.itemsInStore[qtyStored] = dvd;
            this.qtyStored++;
            System.out.println(x:"The DVD has been successfully added");
        }
    }

    public void removeDvd(DigitalVideoDisc dvd){
        int pos = -1;
        for(int i = 0; i < this.qtyStored; i++){
            if(this.itemsInStore[i] == dvd){
                pos = i;
                break;
            }
        }
        if(pos < 0) System.out.println(x:"No such disc found!");
        else{
            for(int i = pos; i < this.qtyStored-1; i++){
                this.itemsInStore[i] = this.itemsInStore[i+1];
            }
            this.qtyStored = this.qtyStored - 1;
            System.out.println(x:"The DVD was successfully removed");
        }
    }
}

```

Running results:

The screenshot shows the Visual Studio Code interface with the Java code from the previous block open in the editor. The Explorer pane on the left shows the project structure with files like Aims.java, Cart.java, CartTest.java, DigitalVideoDisc.java, DvdWrapper.java, Store.java, and TestPassingParameter.java. The Run and Debug pane on the right shows the execution of the code. The terminal output shows the following commands and results:

```

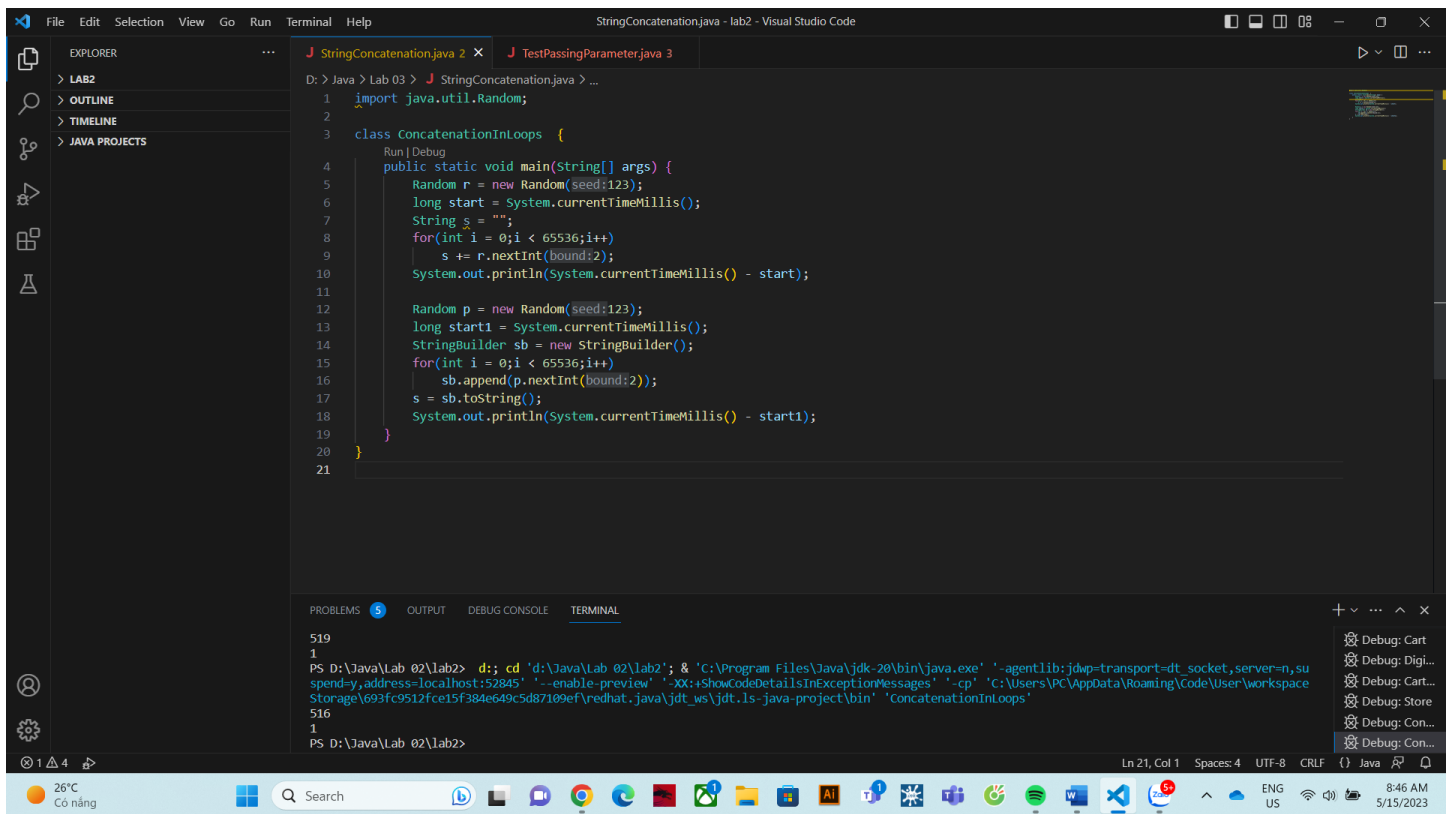
PS D:\Java\Lab 02\lab2> & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:5244 1' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\PC\AppData\Roaming\Code\User\workspaceStorage\693fc9512fce15f384e649c5d87109ef\redhat.java\jdt_ws\jdt.ls-java-project\bin' 'Store'
The DVD has been successfully added
The DVD has been successfully added
The DVD was successfully removed
PS D:\Java\Lab 02\lab2>

```

## String concatenation experiments

Source codes and running results





## Garbage collector