



---

# BÁO CÁO THỰC HÀNH LAB 03

---

Lập trình hướng đối tượng



NOVEMBER 11, 2023

## Contents

Mục tiêu .....	2
1. Phân nhánh repository.....	2
2. Làm việc với ghi đè phương thức .....	2
2.1 Ghi đè bằng các kiểu tham số khác nhau.....	2
2.2 Ghi đè bằng cách thay đổi số lượng tham số đầu vào .....	5
3. Truyền tham số.....	7
4. Sử dụng tính năng debug .....	9
5. Class Member, Instance Member.....	9
6. Mở lớp Cart.....	10
7. Thực hiện lớp Store.....	13
8. Tái tổ chức project .....	15
9. String, StringBuilder and StringBuffer .....	15
10. Mã nguồn .....	18
11. Class Diagram .....	18
Table of Figures.....	20

## Mục tiêu

- Làm việc với Release workflow
- Ghi đè phương thức
- Truyền tham số
- Classifier member vs. Instance member
- Quản lý bộ nhớ với String và StringBuffer và các trường hợp khác
- Debugging
- Tái tổ chức lại project bằng cách tạo các package để quản lý các lớp.

## 1. Phân nhánh repository

Trong bài này để áp dụng Release Flow, ta có thể tạo 9 branch:

- Tạo một nhánh có tên là refactor/apply-release-flow để thực hiện việc tái cấu trúc kho lưu trữ theo Release Flow.
- Tạo một nhánh có tên là topic/method-overloading để thực hiện bài tập về method overloading.
- Tạo một nhánh có tên là topic/passing-parameter để thực hiện bài tập nghiên cứu về cách Java truyền tham số.
- Tạo một nhánh có tên là topic/class-members để thực hiện bài tập với thành viên của lớp (class members) và thành viên của thể hiện (instance members).
- Tạo một nhánh có tên là feature/print-cart để triển khai tính năng in các đĩa trong giỏ hàng.
- Tạo một nhánh có tên là feature/search-cart để triển khai tính năng tìm kiếm các đĩa trong giỏ hàng.
- Tạo một nhánh có tên là topic/store để triển khai lớp Store.
- Tạo một nhánh có tên là refactor/packages để tái cấu trúc dự án trong kho lưu trữ của bạn bằng cách sử dụng gói (packages).
- Tạo một nhánh có tên là topic/memory-management-string để thực hiện bài tập về quản lý bộ nhớ cho String, StringBuffer và StringBuilder.

## 2. Làm việc với ghi đè phương thức

Tạo nhánh topic/method-overloading

### 2.1 Ghi đè bằng các kiểu tham số khác nhau

Trong class `Cart`:

- Tạo một phương thức mới giống tên nhưng khác kiểu dữ liệu đầu vào  
addDigitalVideoDisc(DigitalVideoDisc [] dvdList)

Phương thức này sẽ thêm một danh sách các DVD vào trong giỏ hiện tại.

```
/**
 * Add a list of discs to the cart if the cart is not full
 * @param dvdList list of discs to be added
 */
1 usage new *
public void addDigitalVideoDisc(DigitalVideoDisc [] dvdList) {
    if(qtyOrdered + dvdList.length <= MAX_NUMBER_ORDERED) {
        for(int i = 0; i < dvdList.length; i++) {
            itemsOrdered[qtyOrdered] = dvdList[i];
            qtyOrdered++;
        }
        System.out.println("The discs have been added");
    }
    else {
        System.out.println("The cart is almost full");
    }
}
```

Figure 1 Tạo một phương thức mới ghi đè phương thức cũ, đầu vào là mảng DVD

```
± dvmduc5269 *
public static void main(String args[]) {
    //Create a new cart
    Cart anOrder = 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");
    //anOrder.addDigitalVideoDisc(dvd1);

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

    DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladin", category: "Animation", cost: 18.99f);
    // anOrder.addDigitalVideoDisc(dvd3);
    DigitalVideoDisc []dvds = {dvd1, dvd2, dvd3};
    anOrder.addDigitalVideoDisc(dvds);

    //Print total costs of all items in the cart
    System.out.println("Total Cost is: ");
    System.out.println(anOrder.totalCost());

    //Remove a DVD from the cart
    anOrder.removeDigitalVideoDisc(dvd3);
    //Print all DVDs in the cart
    anOrder.printCart();
}
```

Figure 2 Aims với phương thức mới

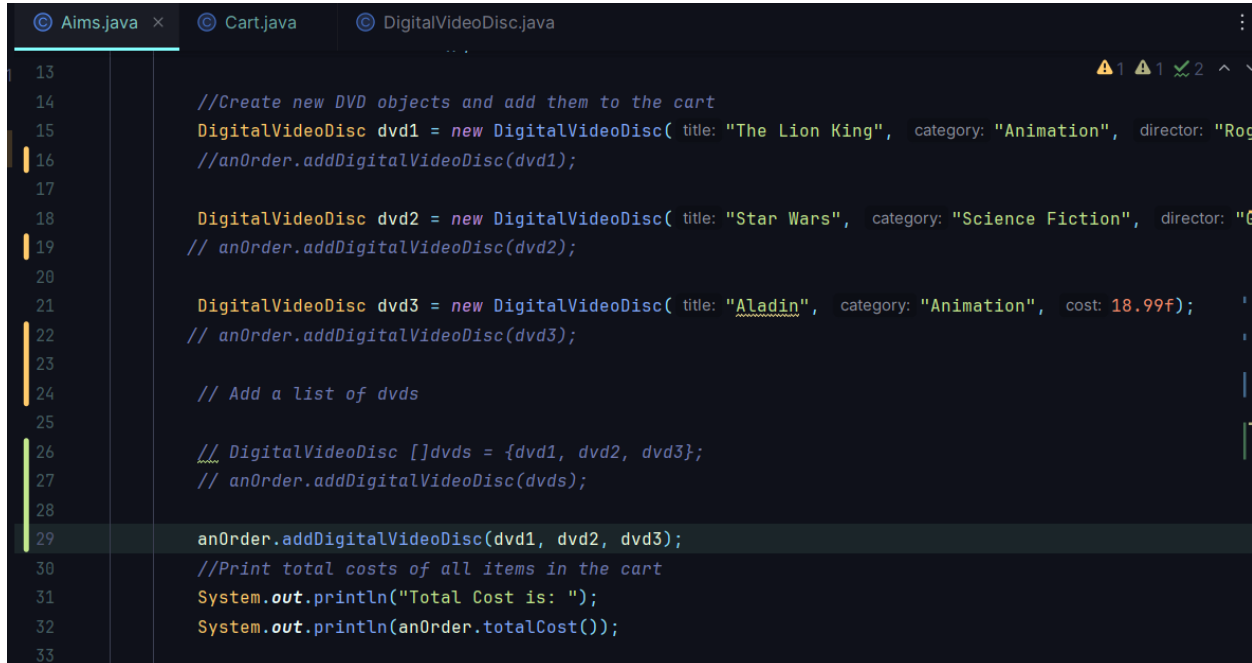
```
The discs have been added
Total Cost is:
63.89
The disc has been removed
DVD - The Lion King - Animation - Roger Allers - 87: 19.95$
DVD - Star Wars - Science Fiction - George Lucas - 87: 24.95$
```

Figure 3 Kết quả với phương thức mới

- Tạo thêm một phương thức mới addDigitalVideoDisc mà đầu vào là một số lượng DVD bất kì, so sánh nó với kiểu dữ liệu đầu vào là mảng. Bạn ưa thích phương thức nào hơn trong trường hợp này?

```
1 usage new *
public void addDigitalVideoDisc(DigitalVideoDisc... dvds) {
    if(qtyOrdered + dvds.length <= MAX_NUMBER_ORDERED) {
        for(int i = 0; i < dvds.length; i++) {
            itemsOrdered[qtyOrdered] = dvds[i];
            qtyOrdered++;
        }
        System.out.println("The discs have been added");
    }
    else {
        System.out.println("The cart is almost full");
    }
}
```

Figure 4 Phương thức mới, đầu vào là các DVD với số lượng bất kì

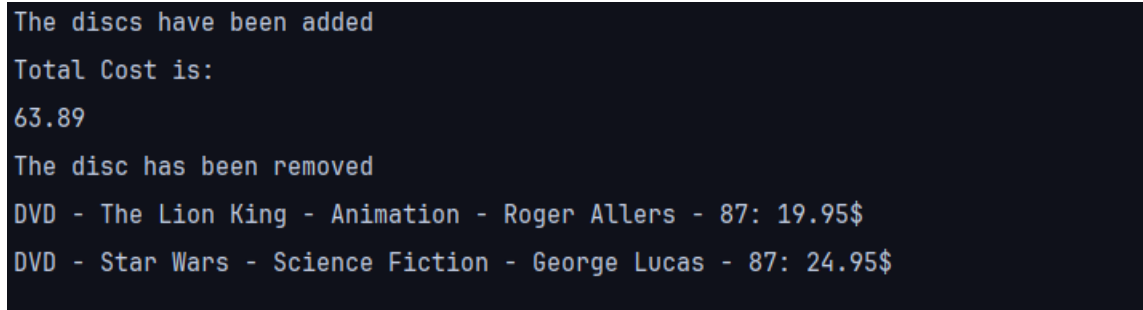


```

13
14 //Create new DVD objects and add them to the cart
15 DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King", category: "Animation", director: "Rog
16 //anOrder.addDigitalVideoDisc(dvd1);
17
18 DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star Wars", category: "Science Fiction", director: "G
19 // anOrder.addDigitalVideoDisc(dvd2);
20
21 DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladin", category: "Animation", cost: 18.99f);
22 // anOrder.addDigitalVideoDisc(dvd3);
23
24 // Add a list of dvds
25
26 // DigitalVideoDisc []dvds = {dvd1, dvd2, dvd3};
27 // anOrder.addDigitalVideoDisc(dvds);
28
29 anOrder.addDigitalVideoDisc(dvd1, dvd2, dvd3);
30 //Print total costs of all items in the cart
31 System.out.println("Total Cost is: ");
32 System.out.println(anOrder.totalCost());
33

```

Figure 5 Aims với phương thức mới



```

The discs have been added
Total Cost is:
63.89
The disc has been removed
DVD - The Lion King - Animation - Roger Allers - 87: 19.95$
DVD - Star Wars - Science Fiction - George Lucas - 87: 24.95$

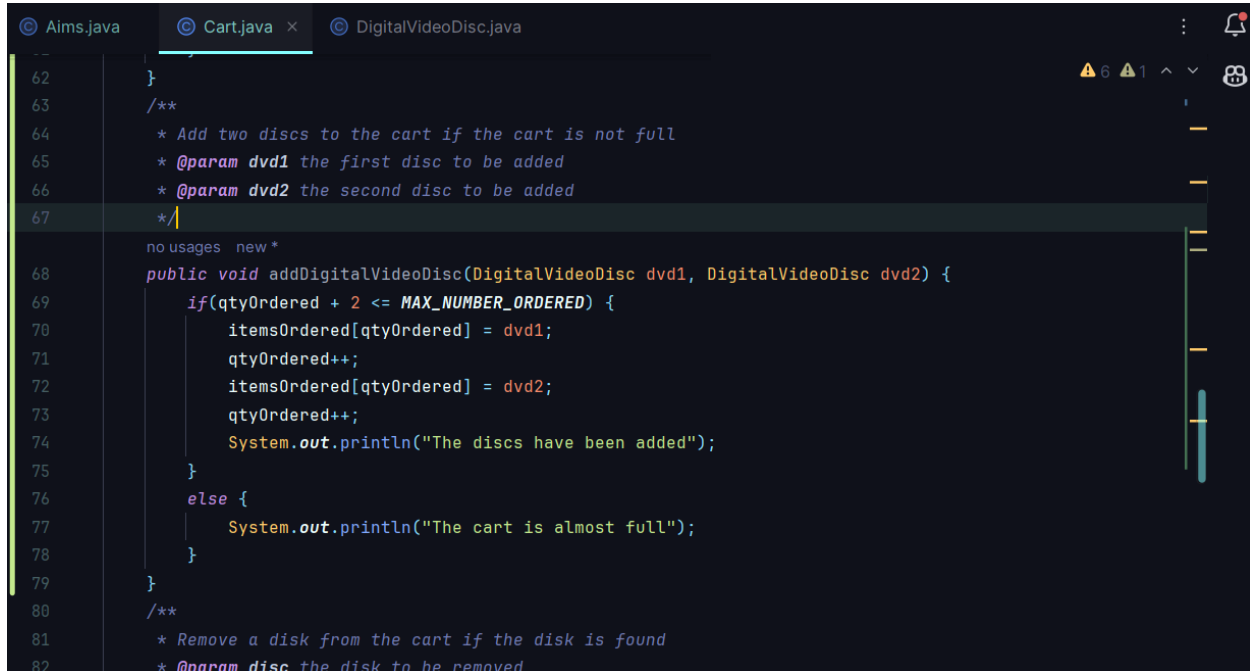
```

Figure 6 Kết quả với phương thức mới

Trong trường hợp này, em thấy phương thức với tham số biến động là tiện lợi hơn, bởi vì đầu vào chỉ là các tham số có cùng kiểu dữ liệu nên:

- Khi muốn thêm DVD vào thì chỉ cần viết DVD đó trực tiếp vào đầu vào của phương thức mà không phải tạo một mảng mới.
- Ta có thể thêm một số lượng DVD bất kì vào đầu vào của phương thức.

## 2.2 Ghi đè bằng cách thay đổi số lượng tham số đầu vào



```

62     }
63     /**
64      * Add two discs to the cart if the cart is not full
65      * @param dvd1 the first disc to be added
66      * @param dvd2 the second disc to be added
67      */
68     no usages new *
69     public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
70         if(qtyOrdered + 2 <= MAX_NUMBER_ORDERED) {
71             itemsOrdered[qtyOrdered] = dvd1;
72             qtyOrdered++;
73             itemsOrdered[qtyOrdered] = dvd2;
74             qtyOrdered++;
75             System.out.println("The discs have been added");
76         }
77         else {
78             System.out.println("The cart is almost full");
79         }
80     }
81     /**
82      * Remove a disk from the cart if the disk is found
83      * @param disc the disk to be removed

```

Figure 7 Phương thức mới với 2 tham số đầu vào



```

13
14     //Create new DVD objects and add them to the cart
15     DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King", category: "Animation", director: "R
16     //anOrder.addDigitalVideoDisc(dvd1);
17
18     DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star Wars", category: "Science Fiction", director: '
19     // anOrder.addDigitalVideoDisc(dvd2);
20
21     DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladdin", category: "Animation", cost: 18.99f);
22     // anOrder.addDigitalVideoDisc(dvd3);
23
24     // Add a list of dvds
25
26     // DigitalVideoDisc []dvds = {dvd1, dvd2, dvd3};
27     // anOrder.addDigitalVideoDisc(dvds);
28
29     anOrder.addDigitalVideoDisc(dvd1, dvd2, dvd3);
30     anOrder.addDigitalVideoDisc(dvd1, dvd2);
31
32     //Print total costs of all items in the cart
33     System.out.println("Total Cost is: ");

```

Figure 8 Aims với phương thức mới ở 2.2 và phương thức ở 2.1

```
The discs have been added
The discs have been added
Total Cost is:
108.78999
The disc has been removed
DVD - The Lion King - Animation - Roger Allers - 87: 19.95$
DVD - Star Wars - Science Fiction - George Lucas - 87: 24.95$
DVD - The Lion King - Animation - Roger Allers - 87: 19.95$
DVD - Star Wars - Science Fiction - George Lucas - 87: 24.95$
```

Figure 9: Kết quả khi dùng 2 phương thức

### 3. Truyền tham số

#### Tạo nhánh topic/passing-parameter

Tạo một class mới tên là `TestPassingParameter`, với mục đích ban đầu là hoán đổi tên của hai đối tượng DVD cho nhau.

```
new *
public class TestPassingParameter {
    // TODO auto-generated method stub
    new *
    public static void main(String[] args) {
        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());
    }

    1usage new *
    public static void swap(Object o1, Object o2) {
        Object tmp = o1;
        o1 = o2;
        o2 = tmp;
    }

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

Figure 10 TestPassingParameter main method



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

Figure 11 Kết quả

jungleDVD, cinderellaDVD là 2 con trỏ đến 2 đối tượng DVD trong bộ nhớ heap

Kết quả:

- Sau khi gọi phương thức swap (jungleDVD, cinderellaDVD), giá trị title của hai đối tượng DVD vẫn giữ nguyên. Điều này là bởi vì khi gọi phương thức swap, JAVA sẽ copy giá trị của các tham số đầu vào vào jungleDVD, cinderellaDVD, sau đó truyền vào phương thức swap để thực hiện. Do đó hàm swap chỉ thực hiện hoán đổi **bản sao** của 2 tham số đầu vào, chứ không thực hiện hoán đổi chính 2 tham số đó.

- Sau khi gọi phương thức  
changeTitle(jungleDVD, cinderellaDVD.getTitle()),

giá trị title của JungleDVD bị thay đổi từ Jungle thành Cinderella. Điều này là vì mặc dù phương thức này làm việc với bản sao các tham số đầu vào, nhưng trong phương thức này có câu lệnh

```
dvd.setTitle(title);
```

Câu lệnh này trực tiếp thay đổi giá trị thuộc tính title của đối tượng mà bản sao của jungleDVD trỏ tới, ở đây đổi thành Cinderella.

Hàm swap có thể hoán đổi 2 đối tượng dvd1 và dvd2 bằng cách đặt giá trị các thuộc tính của DVD này bằng các giá trị của các thuộc tính tương ứng của DVD kia và ngược lại.

Không thể hoán đổi giá trị tham chiếu đến 2 đối tượng trong một phương thức vì Java thực hiện « pass-by-value » cho các tham số.

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

Figure 12 Hàm swap

Kết quả :

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

Figure 13 Kết quả hoán đổi hai đối tượng

## 4. Sử dụng tính năng debug

## 5. Class Member, Instance Member

Tạo nhánh topic/class-members

### Class Member (Thành viên của lớp):

- **Định nghĩa:** Là thành phần thuộc về lớp chứ không phải của một đối tượng cụ thể.
- **Khai báo:** Dùng từ khóa **static** để khai báo.
- **Truy cập:** Có thể truy cập trực tiếp thông qua tên lớp mà không cần tạo một đối tượng của lớp đó.

### Instance Member (Thành viên của thể hiện):

- **Định nghĩa:** Là thành phần thuộc về một đối tượng cụ thể của lớp.
- **Khai báo:** Không sử dụng từ khóa **static**.
- **Truy cập:** Cần tạo một đối tượng của lớp để truy cập.

### Ở trong lớp DigitalVideoDisc:

Tạo một thành viên của lớp tên là `nbDigitalVideoDiscs`, (chỉ số lượng đĩa) một thành viên của đối tượng là `id`. (chỉ id của đĩa).

```
8 usages  
public static int nbDigitalVideoDiscs = 0;  
5 usages  
public String id;
```

Figure 14 Tạo thành viên của lớp, thành viên của thể hiện

Mỗi khi một đĩa DVD mới được tạo ra, số lượng đĩa sẽ tăng lên, id sẽ được gán cho một đĩa.

```

2 usages  @ dvmduc5269 *
public DigitalVideoDisc(String title) {
    this.title = title;
    nbDigitalVideoDiscs++;
    id = "DVD" + nbDigitalVideoDiscs;
}

1 usage  @ dvmduc5269 *
public DigitalVideoDisc(String title, String category, float cost) {
    this.title = title;
    this.category = category;
    this.cost = cost;
    nbDigitalVideoDiscs++;
    id = "DVD" + nbDigitalVideoDiscs;
}

9 ▶ public static void main(String args[]) {
10
11     //Create a new cart
12     Cart anOrder = new Cart();
13
14     //Create new DVD objects and add them to the cart
15     DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King", category: "Animation", director: "Roger Allers", cost: 19.95f);
16     anOrder.addDigitalVideoDisc(dvd1);
17
18     DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star Wars", category: "Science Fiction", director: "George Lucas", cost: 24.95f);
19     anOrder.addDigitalVideoDisc(dvd2);
20
21     DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladin", category: "Animation", cost: 18.99f);
22     anOrder.addDigitalVideoDisc(dvd3);
23
24
25     //Print all DVDs in the cart
26     System.out.println("There are " + anOrder.getQtyOrdered() + " DVDs in the cart:");
27     anOrder.printCart();

```

```

The disc The Lion King has been added
The disc Star Wars has been added
The disc Aladin has been added
There are 3 DVDs in the cart:
DVD1 - The Lion King - Animation - Roger Allers - 87: 19.95$
DVD2 - Star Wars - Science Fiction - George Lucas - 87: 24.95$
DVD3 - Aladin - Animation - null - 0: 18.99$

```

Figure 15: Cập nhật số lượng đĩa, gán ID cho từng đĩa

## 6. Mở lớp Cart

Tạo một phương thức mới để in ra các thông tin của một giỏ hàng theo định dạng cho trước.

Tạo một nhánh mới feature/print-cart để thực hiện phương thức này.

```

2 usages  ± dvmduc5269 *
public void printCart() {
    System.out.println("*****CART*****");
    System.out.println("Ordered items:");
    for(int i = 0; i < qtyOrdered; i++) {
        System.out.println(i + 1 + ". DVD - " + itemsOrdered[i].getTitle() + " - " + itemsOrdered[i].getCategory());
    }
    System.out.println("Total cost: " + totalCost());
    System.out.println("*****");
}
}

```

Figure 16: tạo một phương thức để in thông tin

- Tìm một đĩa trong giỏ hàng bằng ID hoặc bằng tên, in ra kết quả. Nếu không tìm thấy thì thông báo cho người dùng là không tìm thấy. Tạo một nhánh mới tên là feature/search-cart

```

/**
 * Method to search for a DigitalVideoDisc with a given ID
 * @param id the given ID
 * @return true if the DigitalVideoDisc has the same ID as the given ID, false otherwise
 */
1 usage  ± dvmduc5269 *
boolean isMatchId(String id) {
    for(int i = 0; i < qtyOrdered; i++) {
        if(itemsOrdered[i].getId().equals(id)) {
            //print out the information of the DigitalVideoDisc
            itemsOrdered[i].print();
            return true;
        }
    }
    return false;
}

```

Figure 17 Tìm đĩa trong giỏ bằng ID

```

/**
 * Method to check if a DigitalVideoDisc has the same title as a given string
 * @param title the given string
 * @return true if the DigitalVideoDisc has the same title as the given string, false otherwise
 */
1 usage  new *
boolean isMatchTitle(String title) {
    for(int i = 0; i < qtyOrdered; i++) {
        if(itemsOrdered[i].getTitle().equals(title)) {
            //print out the information of the DigitalVideoDisc
            itemsOrdered[i].print();
            return true;
        }
    }
    return false;
}

```

Figure 18 Tìm đĩa trong giỏ bằng tên

- Viết một lớp CartTest để thực hiện tất cả các phương thức trên.

```
public class CartTest {  
    new *  
    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 Lion King", category: "Animation", director: "Rog  
        cart.addDigitalVideoDisc(dvd1);  
  
        DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star Wars", category: "Science Fiction", director: "G  
        cart.addDigitalVideoDisc(dvd2);  
  
        DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladin", category: "Animation", cost: 18.99f);  
        cart.addDigitalVideoDisc(dvd3);  
  
        //Test the print method  
        cart.printCart();  
        //To-do: test the search methods here  
        //Search for a DVD with the ID "DVD1"  
        System.out.println("Search for a DVD with the ID \"DVD1\":");  
        if (!cart.isMatchId("DVD1"))  
            System.out.println("Not found");  
  
        //Search for a DVD with the title "The Lion King"  
        System.out.println("Search for a DVD with the title \"The Lion King\":");  
        if (!cart.isMatchTitle("The Lion King"))  
            System.out.println("Not found");  
  
        //Search for a DVD with the title "Aladi"  
        System.out.println("Search for a DVD with the title \"Aladi\":");  
        if (!cart.isMatchTitle("Aladi"))  
            System.out.println("Not found");  
    }  
}
```

Figure 19 Lớp CartTest

Kết quả:

- In thông tin các đĩa được thêm vào giỏ
- In tổng giá tiền của các đĩa
- In ra thông tin của các đĩa tìm được bằng tên hoặc ID, nếu không tìm được thì thông báo không tìm được.

```

*****CART*****
Ordered items:
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
*****
Search for a DVD with the ID "DVD1":
DVD1 - The Lion King - Animation - Roger Allers - 87: 19.95$
Search for a DVD with the title "The Lion King":
DVD1 - The Lion King - Animation - Roger Allers - 87: 19.95$
Search for a DVD with the title "Aladi":
Not found

```

Figure 20 Kết quả chương trình

## 7. Thực hiện lớp Store

### Tạo nhánh mới tên topic/store

Tạo một lớp Store, lớp này chứa một thuộc tính itemsInStore [] – một mảng các DVDs có trong cửa hàng.

```

public class Store {
    4 usages
    DigitalVideoDisc[] itemsInStore = new DigitalVideoDisc[100];
}

```

Figure 21 Lớp Store với thuộc tính itemsInStore

Tạo hai phương thức addDVD và removeDVD để thêm và loại bỏ một DVD từ cửa hàng

```
/**
 * Add a disc to the store if the store is not full
 * @param disc the disc to be added
 */
3 usages new *
public void addDVD(DigitalVideoDisc disc) {
    if(qtyInStore < 100) {
        itemsInStore[qtyInStore] = disc;
        qtyInStore++;
        System.out.println("The disc has been added");
    }
    else {
        System.out.println("The store is almost full");
    }
}
```

Figure 22 Phương thức addDVD

```
/**
 * Remove a disk from the store if the disk is found
 * @param disc the disk to be removed
 */
1 usage new *
public void removeDVD(DigitalVideoDisc disc) {
    for(int i = 0; i < qtyInStore; i++) {
        if(itemsInStore[i].getTitle().equals(disc.getTitle())) {
            for(int j = i; j < qtyInStore - 1; j++) {
                itemsInStore[j] = itemsInStore[j + 1];
            }
            qtyInStore--;
            System.out.println("The disc has been removed");
            return;
        }
    }
    System.out.println("The disc is not in the store");
}
```

Figure 23 Phương thức removeDVD

Kiểm tra hai phương thức này ở trong lớp StoreTest

```

public class StoreTest {
    new *
    public static void main(String args[]) {
        DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King", category: "Animation", director: "Roge
        DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star Wars", category: "Science Fiction", director: "Ge
        DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladdin", category: "Animation", cost: 18.99f);;

        Store store = new Store();
        store.addDVD(dvd1); // The disc has been added
        store.addDVD(dvd2); // The disc has been added
        store.addDVD(dvd3); // The disc has been added
        store.removeDVD(dvd2); // The disc has been removed
        store.removeDVD(dvd2); // The disc is not in the store
    }
}

```

Figure 24 Lớp StoreTest

Kết quả:

```

The disc has been added
The disc has been added
The disc has been added
The disc has been removed
The disc is not in the store

```

Figure 25 Kết quả của chương trình

## 8. Tái tổ chức project

Tạo một nhánh có tên là refactor/packages để tái cấu trúc dự án trong kho lưu trữ của bạn bằng cách sử dụng gói (packages).

## 9. String, StringBuilder and StringBuffer

Tạo một nhánh có tên là topic/memory-management-string để thực hiện bài tập về quản lý bộ nhớ cho String, StringBuffer và StringBuilder.

Tạo một package mới như đề bài, rồi tạo một class có tên ContenationInLoops, mục đích là kiểm tra thời gian xử lý khi tạo String dùng toán tử +, StringBuffer và StringBuilder.

Với toán tử + :



```

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

```

Figure 26: Dùng toán tử + để dựng một String

Dùng StringBuilder:

```

// String concatenation with StringBuilder
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);
}

```

Figure 27 Dùng StringBuilder để dựng xâu

Kết quả: Dùng toán tử + chậm hơn đáng kể so với dùng phương thức append() của StringBuilder

538	625	548	514	519
3	5	6	5	5

Figure 28: Các kết quả: +(dòng trên), StringBuilder(dòng dưới)

Tạo một lớp GarbageCreator, tạo ra nhiều “rác” bằng cách dùng toán tử “+=” để thêm chuỗi.

Toán tử “+=” tạo thêm 1 đối tượng “String” (Hello World) để thêm vào String s. Sau đó đối tượng String (Hello World) này không còn cách nào khác để truy cập và trở thành rác.

Khi kích thước đầu vào trở nên rất lớn (1 triệu chuỗi “Hello World”), số lượng rác trở nên rất nhiều, ảnh hưởng đến hiệu suất của chương trình và chương trình sẽ mất rất nhiều thời gian để thực hiện.

```
new *
public class GarbageCreator {
    new *
    public static void main(String[] args) {
        Random r = new Random( seed: 123);
        long start = System.currentTimeMillis();
        String s = "Hello World";
        for (int i = 0; i < 1000000; i++) {
            s += "Hello World";
        }
        System.out.println(System.currentTimeMillis() - start);
    }
}
```

Figure 29 GarbageCreator



Figure 30 Kết quả chương trình: không hiển thị

Trong khi với StringBuilder, chương trình thực hiện xong gần như tức thì, với kích thước đầu vào tương tự (1 triệu chuỗi “Hello World”). Điều này là vì khi thêm chuỗi, StringBuilder không tạo thêm một đối tượng như toán tử “+=”, thay vào đó thông qua phương thức append(), chuỗi sẽ được thêm trực tiếp.

```
new *
public class NoGarbage {
    new *
    public static void main(String[] args) {
        Random r = new Random( seed: 123);
        long start = System.currentTimeMillis();
        String s = "Hello World";
        StringBuilder sb = new StringBuilder(s);
        for (int i = 0; i < 1000000; i++) {
            sb.append("Hello World");
        }
        System.out.println(System.currentTimeMillis() - start);
    }
}
```

Figure 31 NoGarbage

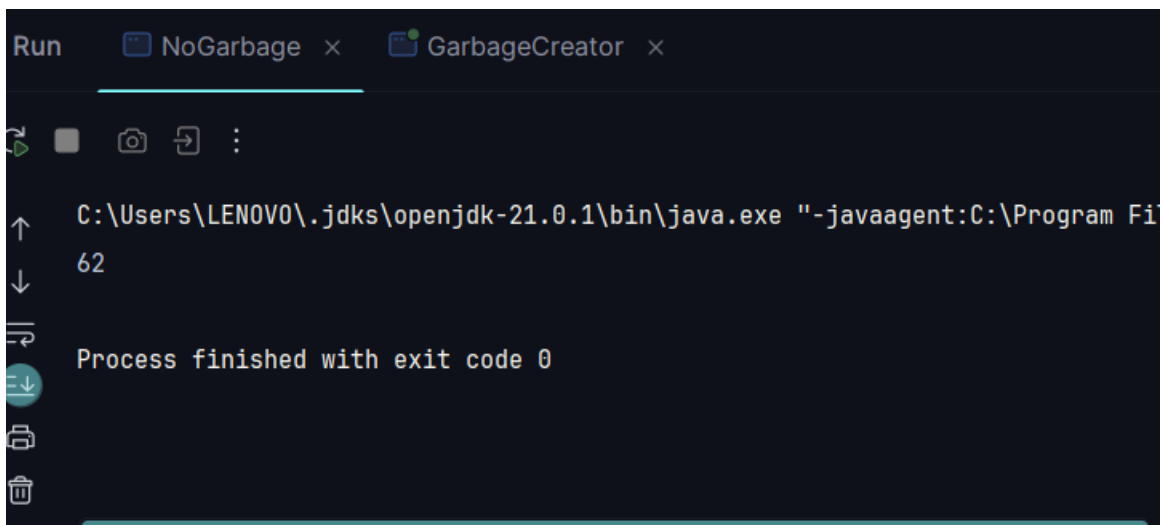


Figure 32 Kết quả chương trình: rất nhanh

## 10. Mã nguồn

Mã nguồn chương trình ở trong nhánh [release/lab03](#) của repo.

## 11. Class Diagram

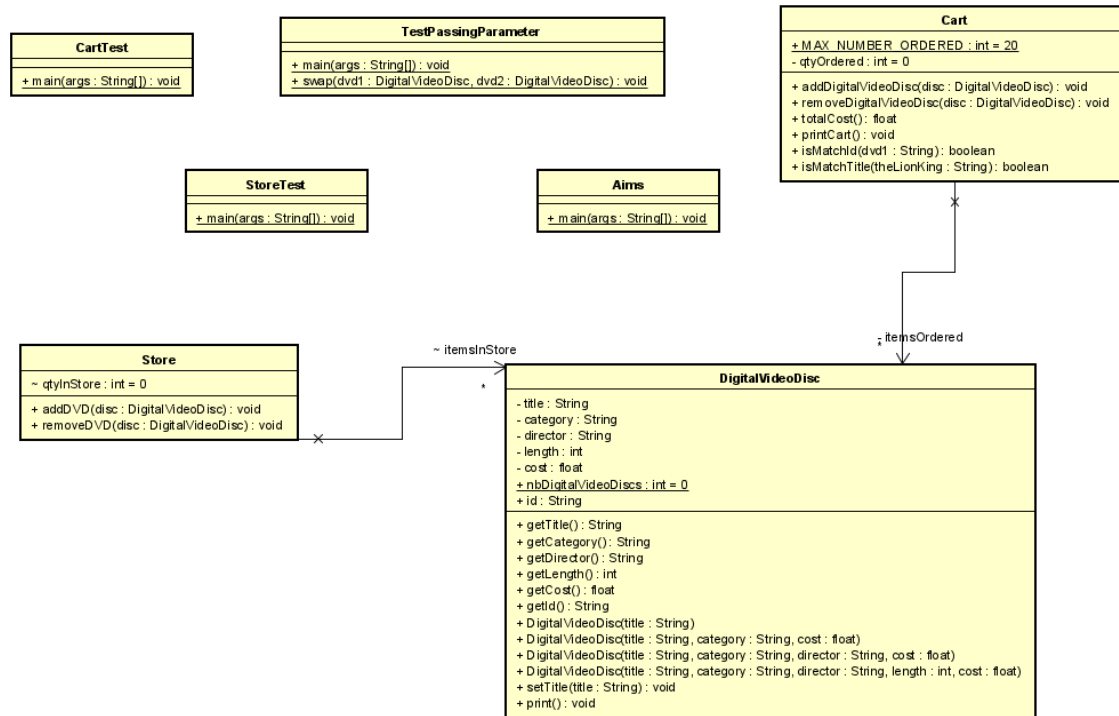


Figure 33 Class Diagram

## Table of Figures

Figure 1 Tạo một phương thức mới ghi đè phương thức cũ, đầu vào là mảng DVD3	
Figure 2 Aims với phương thức mới.....	3
Figure 3 Kết quả với phương thức mới.....	4
Figure 4 Phương thức mới, đầu vào là các DVD với số lượng bất kì .....	4
Figure 5 Aims với phương thức mới.....	5
Figure 6 Kết quả với phương thức mới.....	5
Figure 7 Phương thức mới với 2 tham số đầu vào.....	6
Figure 8 Aims với phương thức mới ở 2.2 và phương thức ở 2.1 .....	6
Figure 9: Kết quả khi dùng 2 phương thức .....	7
Figure 10 TestPassingParameter main method .....	7
Figure 11 Kết quả.....	8
Figure 12 Hàm swap .....	8
Figure 13 Kết quả hoán đổi hai đối tượng .....	9
Figure 14 Tạo thành viên của lớp, thành viên của thể hiện .....	9
Figure 15: Cập nhật số lượng đĩa, gán ID cho từng đĩa.....	10
Figure 16: tạo một phương thức để in thông tin .....	11
Figure 17 Tìm đĩa trong giỏ bằng ID .....	11
Figure 18 Tìm đĩa trong giỏ bằng tên .....	11
Figure 19 Lớp CartTest.....	12
Figure 20 Kết quả chương trình .....	13
Figure 21 Lớp Store với thuộc tính itemsInStore .....	13
Figure 22 Phương thức addDVD .....	14
Figure 23 Phương thức removeDVD .....	14
Figure 24 Lớp StoreTest.....	15
Figure 25 Kết quả của chương trình .....	15
Figure 26: Dùng toán tử + để dựng một String.....	16
Figure 27 Dùng StringBuilder để dựng xâu.....	16
Figure 28: Các kết quả: +(dòng trên), StringBuilder(dòng dưới).....	16
Figure 29 GarbageCreator.....	17
Figure 30 Kết quả chương trình: không hiển thị.....	17
Figure 31 NoGarbage.....	18
Figure 32 Kết quả chương trình: rất nhanh.....	18
Figure 33 Class Diagram.....	19