

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**BÁO CÁO THỰC HÀNH**  
**IT3103 – 744528 - 2024.1**  
**BÀI THỰC HÀNH 2**

Họ và tên sinh viên: Phạm Quốc Cường

MSSV: 20225604

Lớp: Việt Nhật 01 – K67

GVHD: Lê Thị Hoa

HTGD: Đặng Mạnh Cường

Hà Nội 10/2024

# BÁO CÁO THỰC HÀNH LAB 2

## LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

### Contents

Table of Figure: .....	3
1. Bài toán đặt ra .....	4
2. Yêu cầu hệ thống.....	4
2.1. Đối với Customer:.....	4
2.2. Đối với Store Manager:.....	4
3. Use Case Diagram.....	5
4. Class Diagram .....	6
5. Source Code .....	7
5.1. Aims Class .....	7
5.2. DigitalVideoDisc Class .....	8
5.3. Cart Class .....	9
6. Kết quả demo .....	10
7. Reading Assignment .....	12
8. Answer the question.....	12

## Table of Figure:

Figure 1: Use Case Diagram .....	5
Figure 2: Class Diagram.....	6
Figure 3: Aims Class .....	7
Figure 4: DigitalVideoDiscClass.....	8
Figure 5: Cart Class.....	9
Figure 6: Code Demo .....	10
Figure 7: Result .....	11
Figure 8: Getters & Setter methods .....	12

## 1. Bài toán đặt ra

Trong bối cảnh cạnh tranh mạnh mẽ giữa các nền tảng thương mại điện tử, Ti-do (sự kết hợp giữa Tiki và Sendo) cần xây dựng một hệ thống mới mang tên **AIMS (An Internet Media Store)** để cung cấp một nền tảng mua sắm trực tuyến dành riêng cho các sản phẩm phương tiện truyền thông, với loại phương tiện hiện tại là **DVD**. Hệ thống này cần có khả năng hỗ trợ người dùng tìm kiếm và mua sản phẩm, cũng như quản lý đơn hàng và hàng tồn kho hiệu quả cho quản lý cửa hàng.

## 2. Yêu cầu hệ thống

### 2.1. Đối với Customer:

- **Xem danh sách DVD**
- **Tìm kiếm DVD:** Người dùng có thể tìm kiếm DVD theo các tiêu chí:
  - **Theo tên:** Nhập chuỗi từ khóa và hệ thống sẽ tìm các DVD có từ khóa trong tiêu đề, không phân biệt chữ hoa, chữ thường.
  - **Theo thể loại:** Tìm các DVD theo thể loại mong muốn.
  - **Theo giá:** Tìm các DVD dựa trên giá tối thiểu và tối đa hoặc chỉ theo giá tối đa.
- **Xem thông tin DVD:** Cho phép người dùng xem chi tiết các thông tin về DVD.
- **Thêm DVD vào giỏ hàng:** Cho phép thêm DVD vào giỏ hàng từ danh sách tìm kiếm hoặc từ màn hình chi tiết.
- **Quản lý giỏ hàng:**
  - **Xem giỏ hàng:** Hiển thị tất cả thông tin về DVD trong giỏ cùng với tổng giá.
  - **Sắp xếp DVD:** Có thể sắp xếp các DVD trong giỏ hàng theo:
    - **Tên:** Theo thứ tự bảng chữ cái. Nếu cùng tên, sắp xếp theo giá từ cao xuống thấp.
    - **Giá:** Sắp xếp giảm dần theo giá. Nếu giá giống nhau, sắp xếp theo tên tăng dần.
  - **Bộ lọc DVD:** Có thể tìm kiếm trong giỏ hàng dựa trên **ID** hoặc **Tên** của DVD.
  - **Cập nhật số lượng DVD:** Cho phép thay đổi số lượng hoặc xóa DVD khỏi giỏ.
  - **Nhận một DVD miễn phí:** Một sản phẩm ngẫu nhiên trong giỏ sẽ được miễn phí.
- **Đặt hàng:**
  - Nhập thông tin giao hàng và hướng dẫn giao hàng.
  - Hệ thống tính toán phí giao hàng dựa trên khối lượng và vị trí giao hàng.
  - Hiển thị hóa đơn với danh sách DVD, tổng giá trước và sau VAT, cùng với phí giao hàng.
- **Thanh toán đơn hàng:**
  - Thanh toán bằng thẻ tín dụng qua kết nối với hệ thống xác thực thẻ.
  - Sau khi thanh toán, hiển thị thông tin chi tiết giao dịch và gửi email xác nhận đơn hàng.

### 2.2. Đối với Store Manager:

- **Đăng nhập hệ thống:** Yêu cầu Store Manager đăng nhập để truy cập các chức năng quản lý.
- **Quản lý đơn hàng:**
  - Xem danh sách các đơn hàng đang chờ xử lý.
  - Chọn từng đơn để xem chi tiết và phê duyệt hoặc từ chối đơn.
- **Quản lý kho DVD:**
  - **Thêm DVD mới:** Cung cấp tất cả thông tin về DVD bao gồm ID, tên, thể loại, đạo diễn, độ dài và giá thành để thêm vào kho.
  - **Xóa DVD:** Xóa một DVD khỏi kho nếu không còn cần thiết.

### 3. Use Case Diagram

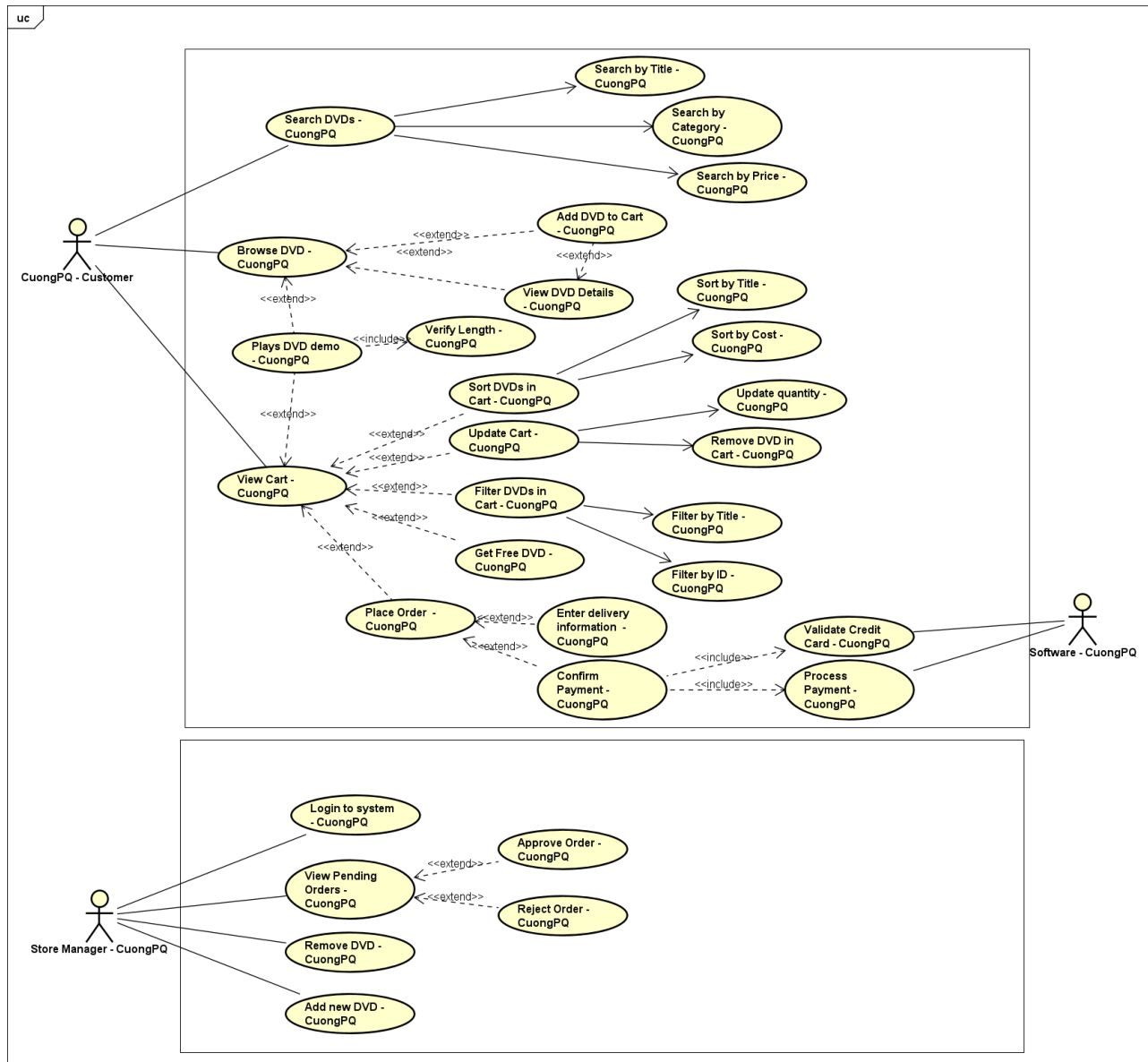


Figure 1: Use Case Diagram

## 4. Class Diagram

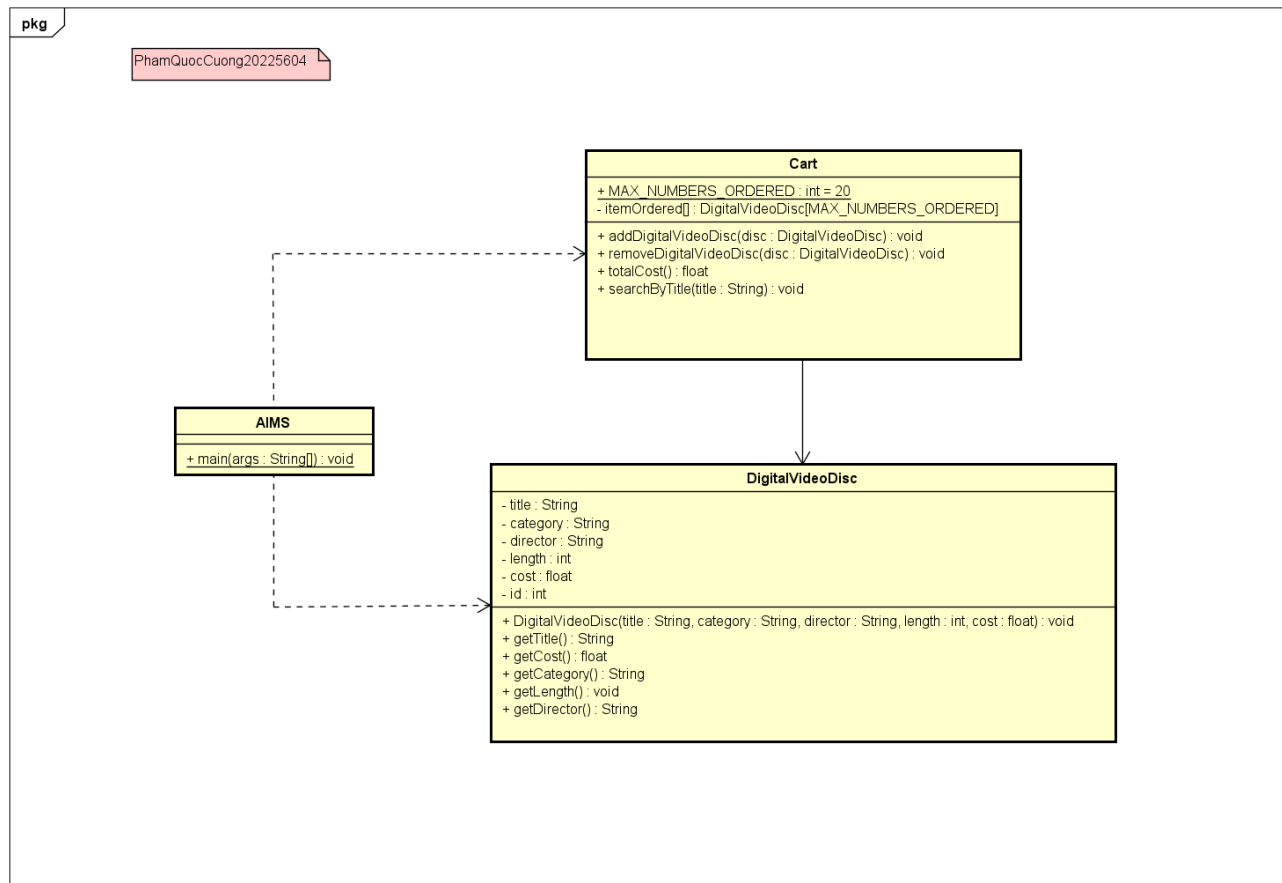


Figure 2: Class Diagram

## 5. Source Code

### 5.1. Aims Class



```
1 // Pham Quoc Cuong - 20225604
2 // Main class
3 package cuongpq;
4
5 public class Aims {
6     public static void main(String[] args) {
7         //Create a new cart
8         Cart anOrder = new Cart();
9
10        //Create new dvd objects and add them to the cart
11        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King","Animation",
12            "Roger Allers",87,19.95f);
13        anOrder.addDigitalVideoDisc(dvd1);
14        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star wars","Science Fiction",
15            "Geogre Lucas",87,24.95f);
16        anOrder.addDigitalVideoDisc(dvd2);
17        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin","Animation",18.99f);
18        anOrder.addDigitalVideoDisc(dvd3);
19
20        System.out.printf("CuongPQ 5604 : Total cost is: %.2f",anOrder.totalCost());
21        System.out.println();
22    }
23 }
```

Figure 3: Aims Class

## 5.2. DigitalVideoDisc Class

```
1 // Pham Quoc Cuong - 20225604
2 package cuongpq;
3
4 public class DigitalVideoDisc {
5     private String title;
6     private String category;
7     private String director;
8     private int length;
9     private float cost;
10
11     //Constructor by title
12     public DigitalVideoDisc(String title) {
13         super();
14         this.title = title;
15     }
16     // Constructor by category, title and cost
17     public DigitalVideoDisc(String title, String category, float cost) {
18         this.title = title;
19         this.category = category;
20         this.cost = cost;
21     }
22     // Constructor by director, category, title and cost
23     public DigitalVideoDisc(String title, String category, String director, float cost) {
24         this.title = title;
25         this.category = category;
26         this.director = director;
27         this.cost = cost;
28     }
29     // Constructor by all attributes
30     public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
31         this.title = title;
32         this.category = category;
33         this.director = director;
34         this.length = length;
35         this.cost = cost;
36     }
37
38     //Getter
39     public String getTitle() {
40         return title;
41     }
42     public String getCategory() {
43         return category;
44     }
45     public String getDirector() {
46         return director;
47     }
48     public int getLength() {
49         return length;
50     }
51     public float getCost() {
52         return cost;
53     }
54 }
55
```

Figure 4: DigitalVideoDiscClass



### 5.3. Cart Class

```

1 // Pham Quoc Cuong - 20225604
2 package cuongpq;
3
4 public class Cart {
5     public static final int MAX_NUMBERS_ORDERED = 20;
6
7     // Array to store ordered DVDs
8     private DigitalVideoDisc itemOrdered[] =
9         new DigitalVideoDisc[MAX_NUMBERS_ORDERED];
10
11     private int qtyOrdered; // how many DigitalVideoDiscs in cart
12
13     // Method to add an item to the List
14     public int addDigitalVideoDisc(DigitalVideoDisc disc) {
15         if (qtyOrdered + 1 > MAX_NUMBERS_ORDERED) {
16             System.out.println("CuongPQ 5604 : The cast is almost full!");
17             return 0;
18         } else {
19             itemOrdered[qtyOrdered] = disc;
20             qtyOrdered++;
21             System.out.println("CuongPQ 5604 : The DVD " + "(" + qtyOrdered + ") " + "\"" + disc.getTitle() + "\" have been added! ");
22             return 1;
23         }
24     }
25
26     // Method to remove an item from the List
27     public int removeDigitalVideoDisc(DigitalVideoDisc disc){
28         // Check if cart is empty
29         if (qtyOrdered == 0) {
30             System.out.println("CuongPQ 5604 : Your cart is empty!");
31             return 0;
32         }
33         // Find the disc in the array
34         boolean found = false;
35         for (int i = 0; i < qtyOrdered; i++) {
36             if (itemOrdered[i].equals(disc)) { // Check if the DVD matches
37                 found = true;
38                 // Shift all items after the removed one to the Left
39                 for (int j = i; j < qtyOrdered - 1; j++) {
40                     itemOrdered[j] = itemOrdered[j + 1];
41                 }
42                 itemOrdered[qtyOrdered - 1] = null; // Clear the Last item
43                 qtyOrdered--; // Reduce the number of items
44                 System.out.println("CuongPQ 5604 : Removed DVD '" + disc.getTitle() + "' successfully!");
45                 break;
46             }
47         }
48         // If no matching DVD was found
49         if (!found) {
50             System.err.println("CuongPQ 5604 : No matching DVD found!");
51             return 0;
52         }
53         return 1;
54     }
55
56     // Method to caculate the total cost DVDS in the cart
57     public float totalCost() {
58         float sum = 0.00f;
59         for(int i=0; i<qtyOrdered; i++) {
60             sum += itemOrdered[i].getCost();
61         }
62         return sum;
63     }
64 }
65 }
66

```

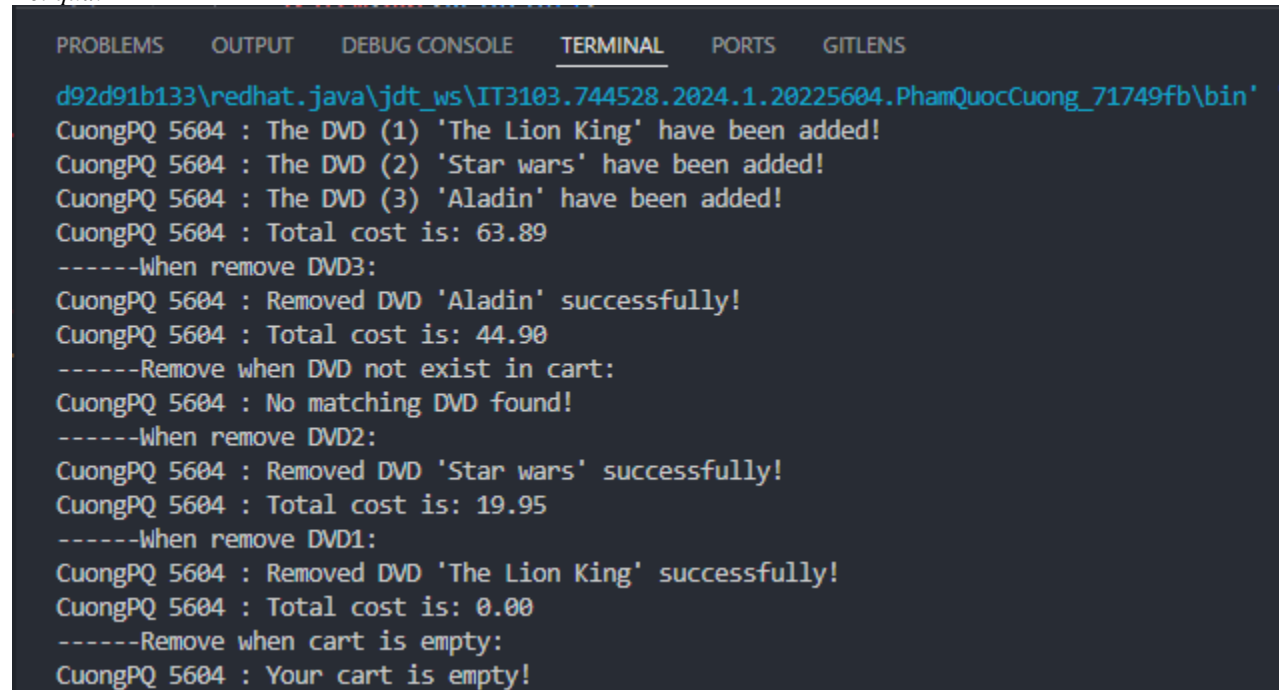
Figure 5: Cart Class

## 6. Kết quả demo

```
1 // Pham Quoc Cuong - 20225604
2 // Main class
3 package cuongpq;
4
5 public class Aims {
6     public static void main(String[] args) {
7         //Create a new cart
8         Cart anOrder = new Cart();
9
10        //Create new dvd objects and add them to the cart
11        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King","Animation",
12            "Roger Allers",87,19.95f);
13        anOrder.addDigitalVideoDisc(dvd1);
14        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star wars","Science Fiction",
15            "Geogre Lucas",87,24.95f);
16        anOrder.addDigitalVideoDisc(dvd2);
17        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin","Animation",18.99f);
18        anOrder.addDigitalVideoDisc(dvd3);
19
20        System.out.printf("CuongPQ 5604 : Total cost is: %.2f",anOrder.totalCost());
21        System.out.println();
22
23        // Test when remove DVD
24        System.out.println("-----When remove DVD3:");
25        anOrder.removeDigitalVideoDisc(dvd3);
26        System.out.printf("CuongPQ 5604 : Total cost is: %.2f",anOrder.totalCost());
27        System.out.println();
28
29        System.out.println("-----Remove when DVD not exist in cart:");
30        anOrder.removeDigitalVideoDisc(dvd3); // Test remove when DVD not exist
31
32        System.out.println("-----When remove DVD2:");
33        anOrder.removeDigitalVideoDisc(dvd2);
34        System.out.printf("CuongPQ 5604 : Total cost is: %.2f",anOrder.totalCost());
35        System.out.println();
36
37        System.out.println("-----When remove DVD1:");
38        anOrder.removeDigitalVideoDisc(dvd1);
39        System.out.printf("CuongPQ 5604 : Total cost is: %.2f",anOrder.totalCost());
40        System.out.println();
41
42        System.out.println("-----Remove when cart is empty:");
43        anOrder.removeDigitalVideoDisc(dvd1); // Test remove when cart is empty
44    }
45 }
46
```

Figure 6: Code Demo

Kết quả:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  
d92d91b133\redhat.java\jdt_ws\IT3103.744528.2024.1.20225604.PhamQuocCuong_71749fb\bin'  
CuongPQ 5604 : The DVD (1) 'The Lion King' have been added!  
CuongPQ 5604 : The DVD (2) 'Star wars' have been added!  
CuongPQ 5604 : The DVD (3) 'Aladin' have been added!  
CuongPQ 5604 : Total cost is: 63.89  
-----When remove DVD3:  
CuongPQ 5604 : Removed DVD 'Aladin' successfully!  
CuongPQ 5604 : Total cost is: 44.90  
-----Remove when DVD not exist in cart:  
CuongPQ 5604 : No matching DVD found!  
-----When remove DVD2:  
CuongPQ 5604 : Removed DVD 'Star wars' successfully!  
CuongPQ 5604 : Total cost is: 19.95  
-----When remove DVD1:  
CuongPQ 5604 : Removed DVD 'The Lion King' successfully!  
CuongPQ 5604 : Total cost is: 0.00  
-----Remove when cart is empty:  
CuongPQ 5604 : Your cart is empty!
```

Figure 7: Result

## 7. Reading Assignment

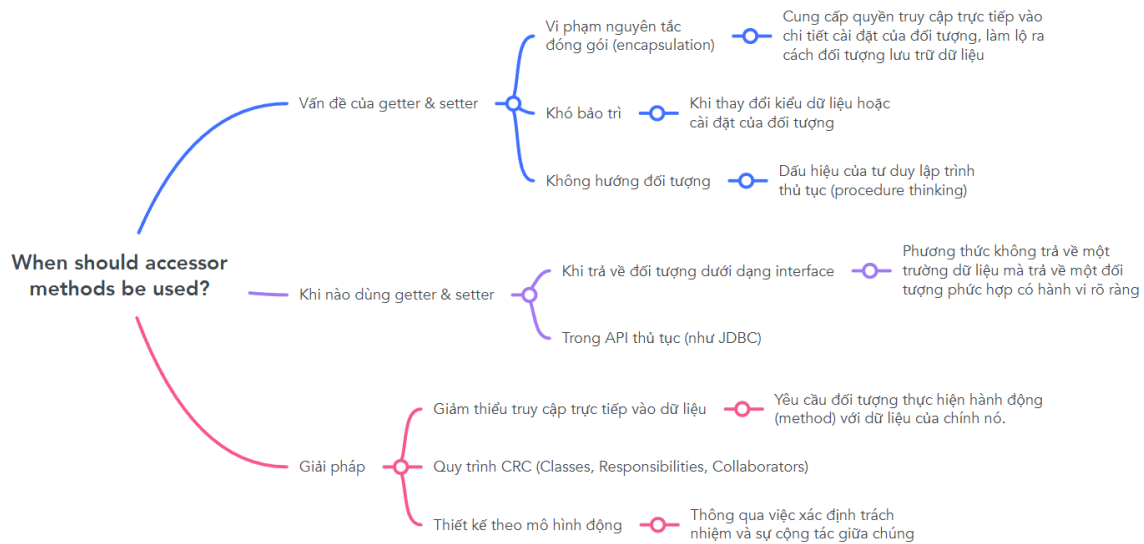


Figure 8: Getters & Setter methods

## 8. Answer the question

**Question:** If you create a constructor method to build a DVD by title then create a constructor method to build a DVD by category. Does JAVA allow you to do this?

**Answer:**

Trong ví dụ này, cả hai constructor đều nhận một String làm tham số, điều này sẽ dẫn đến lỗi vì Java không thể phân biệt được bạn muốn gọi constructor nào khi chỉ có một String được truyền vào. Điều này sẽ gây ra lỗi “duplicate method” vì Java dựa trên kiểu và số lượng tham số để phân biệt các phương thức (method signature).

Để giải quyết vấn đề này có thể phân biệt bằng số lượng hoặc loại tham số của các constructor để tránh trùng lặp.