

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

**BÁO CÁO THỰC HÀNH**  
**IT3103-744527-2024.1**  
**BÀI THỰC HÀNH -LAB02**

Họ và tên sv: Trần Cao Bảo Phúc  
MSSV:20225756  
Lớp:IT-E601-K67  
GVHD: Lê Thị Hoa  
HTGD: Đặng Mạnh Cường

Hà Nội 9/2024

# BÁO CÁO THỰC HÀNH LAB 2

## LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

### Table of Contents

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

### Table of Figures

Figure 1: Use Case Diagram .....	3
Figure 2: Class Diagram.....	4
Figure 3: Aims Class .....	5
Figure 4: DigitalVideoDisc Class 1.....	6
Figure 5: DigitalVideoDisc Class 2.....	7
Figure 6: Cart Class 1.....	8
Figure 7: Cart Class 2.....	9
Figure 8: Code Demo .....	10
Figure 9: Result .....	11
Figure 10: Getter and Setter Methods .....	12

## 1. Bài toán đặt ra

Có thể trong tương lai, Tiki và Sendo sẽ thảo luận về khả năng sáp nhập nhằm cạnh tranh với các nền tảng thương mại điện tử khác, đặc biệt là những nền tảng có sự hỗ trợ từ nước ngoài. Việc sáp nhập của hai công ty này sẽ tạo ra công ty Ti-do, trong đó “Ti” là từ Tiki, và “do” là từ Sendo, có nghĩa là một công ty tỷ đô la tại Việt Nam. Công ty này, Ti-do, muốn bạn giúp họ tạo ra một hệ thống hoàn toàn mới cho dự án AIMS (AIMS viết tắt là An Internet Media Store). Hiện tại, chỉ có một loại phương tiện là Đĩa Video Kỹ thuật số (DVD).

## 2. Yêu cầu hệ thống

### 2.1. Đối với Customer:

#### a) **Duyệt danh sách DVD:**

- Khách hàng có thể xem danh sách các DVD có sẵn trong cửa hàng, được sắp xếp theo thứ tự mới nhất đến cũ nhất dựa trên ngày thêm vào.

#### b) **Tìm kiếm DVD:**

- Khách hàng có thể tìm kiếm DVD dựa trên ba tiêu chí:
  - Theo tiêu đề:** Tìm kiếm dựa trên chuỗi từ khóa, hệ thống sẽ tìm những DVD có tiêu đề chứa bất kỳ từ nào trong chuỗi từ khóa (không phân biệt chữ hoa hay chữ thường).
  - Theo thể loại:** Tìm kiếm dựa trên tên thể loại (không phân biệt chữ hoa hay chữ thường).
  - Theo giá:** Khách hàng có thể cung cấp giá tối thiểu và tối đa, hoặc chỉ cần cung cấp giá tối đa.

#### c) **Nghe thử DVD:**

- Khách hàng có thể chọn phát một phần demo của DVD. Nếu DVD có độ dài bằng 0 hoặc ít hơn, hệ thống sẽ thông báo rằng DVD không thể phát.

#### d) **Xem chi tiết DVD:**

- Khách hàng có thể xem thông tin chi tiết của một DVD từ danh sách hiển thị.

#### e) **Thêm DVD vào giỏ hàng:**

- Khách hàng có thể thêm DVD vào giỏ hàng từ danh sách hiển thị hoặc từ màn hình chi tiết của DVD.

#### f) **Xem và quản lý giỏ hàng:**

- Khách hàng có thể xem tất cả thông tin về các DVD trong giỏ hàng cùng với tổng chi phí.
- Khách hàng có thể nghe thử các DVD trong giỏ trước khi xác nhận đặt hàng.
- Sắp xếp giỏ hàng:**

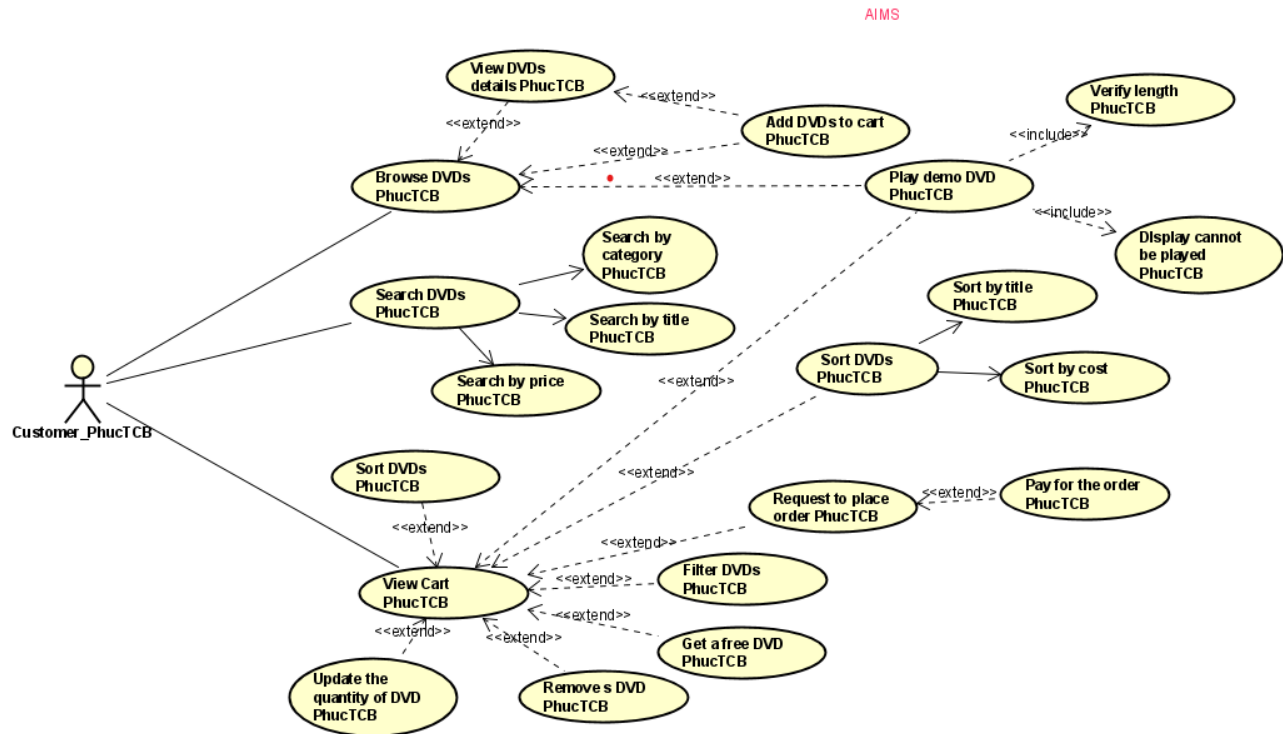
- Theo tiêu đề: Sắp xếp theo thứ tự bảng chữ cái của tiêu đề, nếu có tiêu đề trùng nhau, DVD có giá cao hơn sẽ hiển thị trước.
- Theo giá: Sắp xếp giảm dần theo giá, nếu giá trùng nhau, DVD có tiêu đề nhỏ hơn sẽ hiển thị trước.
- **Cập nhật giỏ hàng:** Khách hàng có thể cập nhật số lượng hoặc xóa DVD khỏi giỏ hàng.
- **Chọn DVD miễn phí:** Hệ thống sẽ chọn ngẫu nhiên một sản phẩm trong giỏ hàng để tặng miễn phí cho khách hàng.
- g) **Lọc DVD trong giỏ hàng:**
  - Khách hàng có thể lọc DVD trong giỏ hàng dựa trên ID hoặc tiêu đề. Nếu không tìm thấy, hệ thống sẽ thông báo.
- h) **Đặt hàng:**
  - Khách hàng có thể yêu cầu đặt hàng mà không cần đăng nhập.
  - Hệ thống sẽ yêu cầu khách hàng cung cấp thông tin giao hàng và hướng dẫn giao hàng.
  - Hệ thống tính phí giao hàng dựa trên tổng khối lượng của đơn hàng và địa điểm giao hàng.
  - Hiển thị hóa đơn bao gồm danh sách DVD, tổng chi phí trước và sau thuế VAT, phí giao hàng.
  - Khách hàng có thể thanh toán bằng thẻ tín dụng thông qua hệ thống kiểm tra và thực hiện giao dịch.
  - Sau khi giao dịch hoàn tất, hệ thống hiển thị thông tin chi tiết giao dịch và gửi email thông báo đến khách hàng.

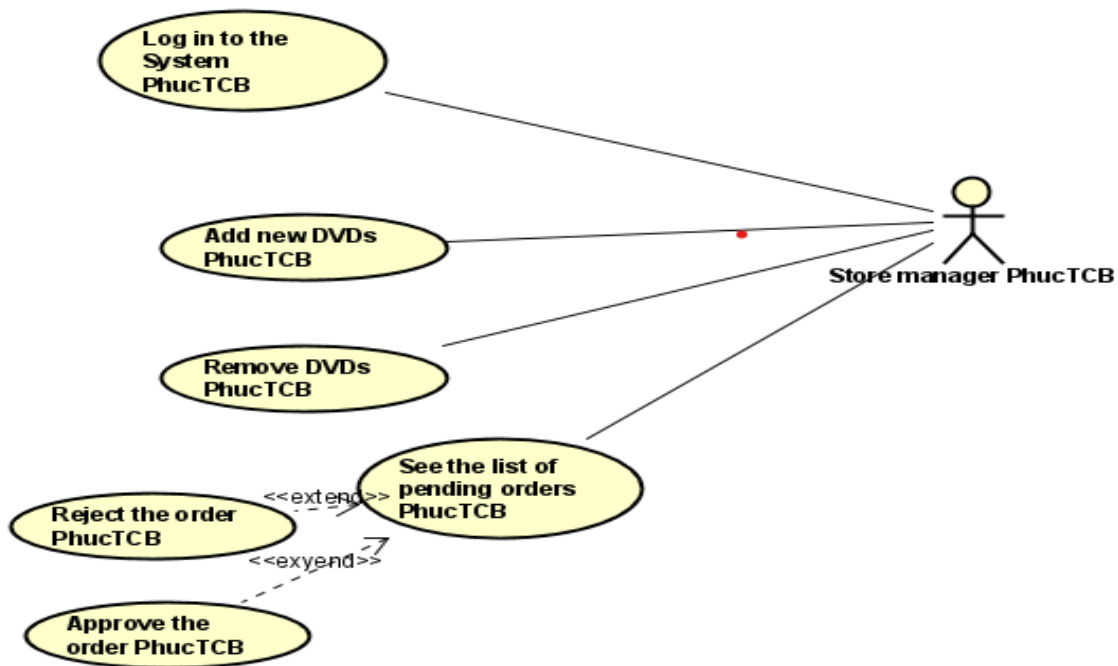
## 2.2. Đối với Store Manager:

- a) **Đăng nhập và quản lý cửa hàng:**
  - Quản lý cửa hàng phải đăng nhập vào hệ thống để điều hành chế độ quản lý.
- b) **Quản lý đơn hàng:**
  - Xem danh sách các đơn hàng đang chờ xử lý.
  - Quản lý có thể chọn đơn hàng để xem chi tiết và quyết định phê duyệt hoặc từ chối đơn hàng đó.
- c) **Quản lý sản phẩm:**
  - Quản lý có thể thêm sản phẩm mới (DVD) vào cửa hàng bằng cách cung cấp thông tin chi tiết bao gồm ID, tiêu đề, thể loại, đạo diễn, độ dài và giá.

- Quản lý có thể xóa DVD khỏi danh sách sản phẩm của cửa hàng.

### 3. Use Case Diagram





*Figure 1: Use Case Diagram*

## 4. Class Diagram

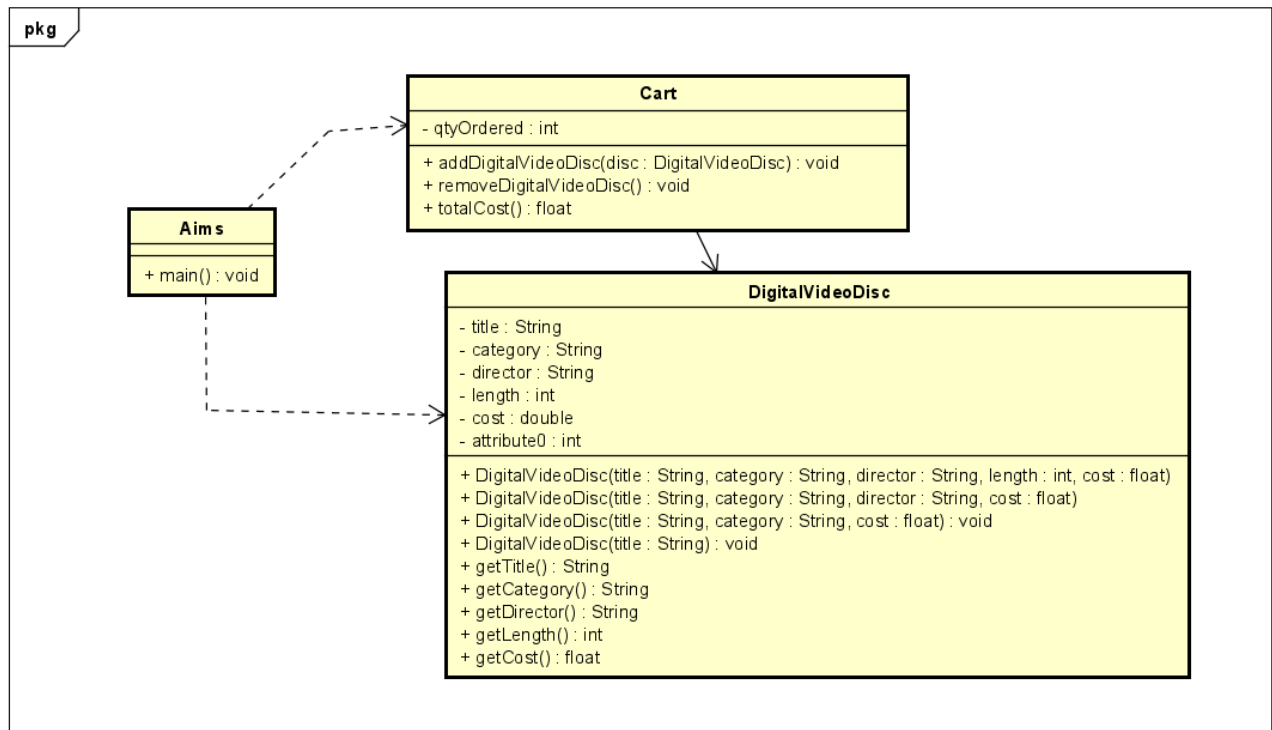


Figure 2: Class Diagram

## 5. Source Code

### 5.1. Aims Class

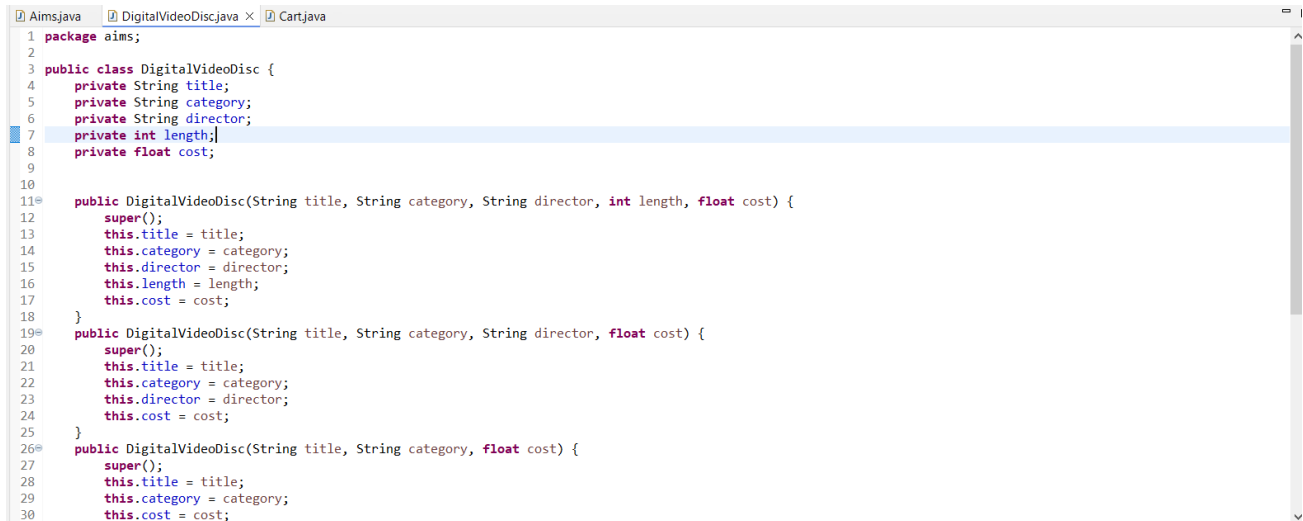
```

1 package aims;
2
3 public class Aims {
4     public static void main(String[] args) {
5         Cart anOrder = new Cart();
6
7         DigitalVideoDisc dvd1 = new DigitalVideoDisc("BaoPhuc-The Lion King",
8             "Animation", "Roger Allers", 87, 19.95f);
9         anOrder.addDigitalVideoDisc(dvd1);
10
11        DigitalVideoDisc dvd2 = new DigitalVideoDisc("BaoPhuc-Star Wars",
12            "Science Fiction", "George Lucas", 87, 24.95f);
13        anOrder.addDigitalVideoDisc(dvd2);
14
15        DigitalVideoDisc dvd3 = new DigitalVideoDisc("BaoPhuc-Aladin",
16            "Animation", 18.99f);
17        anOrder.addDigitalVideoDisc(dvd3);
18
19        DigitalVideoDisc dvd4 = new DigitalVideoDisc("BaoPhuc-Two broken girls",
20            "Comedy", "BaoPhuc", 22.22f);
21        anOrder.addDigitalVideoDisc(dvd4);
22
23        System.out.println("Total cost is:");
24        System.out.println(anOrder.totalCost());
25
26        anOrder.removeDigitalVideoDisc(dvd3);
27
28        System.out.println("Total cost is:");
29        System.out.println(anOrder.totalCost());
30
31    }
32 }

```

Figure 3: Aims Class

## 5.2. DigitalVideoDisc Class



```
1 package aims;
2
3 public class DigitalVideoDisc {
4     private String title;
5     private String category;
6     private String director;
7     private int length;
8     private float cost;
9
10
11     public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
12         super();
13         this.title = title;
14         this.category = category;
15         this.director = director;
16         this.length = length;
17         this.cost = cost;
18     }
19     public DigitalVideoDisc(String title, String category, String director, float cost) {
20         super();
21         this.title = title;
22         this.category = category;
23         this.director = director;
24         this.cost = cost;
25     }
26     public DigitalVideoDisc(String title, String category, float cost) {
27         super();
28         this.title = title;
29         this.category = category;
30         this.cost = cost;
31     }
32 }
```

*Figure 4: DigitalVideoDisc Class 1*



```
23     this.director = director;
24     this.cost = cost;
25 }
26 public DigitalVideoDisc(String title, String category, float cost) {
27     super();
28     this.title = title;
29     this.category = category;
30     this.cost = cost;
31 }
32 public DigitalVideoDisc(String title) {
33     super();
34     this.title = title;
35 }
36 public String getTitle() {
37     return title;
38 }
39 public String getCategory() {
40     return category;
41 }
42 public String getDirector() {
43     return director;
44 }
45 public int getLength() {
46     return length;
47 }
48 public float getCost() {
49     return cost;
50 }
51 }
52 }
```

*Figure 5: DigitalVideoDisc Class 2*



### 5.3. Cart Class

```

1 package aims;
2
3 public class Cart {
4     public static final int MAX_NUMBERS_ORDERED = 20;
5     private int qtyOrdered = 0;
6     private DigitalVideoDisc itemOrdered[] =
7         new DigitalVideoDisc[MAX_NUMBERS_ORDERED];
8
9
10    public void addDigitalVideoDisc(DigitalVideoDisc disc) {
11        if (qtyOrdered < MAX_NUMBERS_ORDERED) {
12            itemOrdered[qtyOrdered] = disc;
13            qtyOrdered++;
14            System.out.println("BaoPhuc-The disc has been added: " + disc.getTitle());
15        } else {
16            System.out.println("BaoPhuc-The cart is almost full.");
17        }
18    }
19
20
21    public void removeDigitalVideoDisc(DigitalVideoDisc disc) {
22        boolean found = false;
23        for (int i = 0; i < qtyOrdered; i++) {
24            if (itemOrdered[i].getTitle().equals(disc.getTitle())) { //so sánh qua tên sản phẩm
25                found = true;
26                //dịch chuyển các sản phẩm sau sản phẩm cần xóa lên 1 đơn vị
27                for (int j = i; j < qtyOrdered - 1; j++) {
28                    itemOrdered[j] = itemOrdered[j + 1];
29                }

```

Figure 6: Cart Class 1

```

21    public void removeDigitalVideoDisc(DigitalVideoDisc disc) {
22        boolean found = false;
23        for (int i = 0; i < qtyOrdered; i++) {
24            if (itemOrdered[i].getTitle().equals(disc.getTitle())) { //so sánh qua tên sản phẩm
25                found = true;
26                //dịch chuyển các sản phẩm sau sản phẩm cần xóa lên 1 đơn vị
27                for (int j = i; j < qtyOrdered - 1; j++) {
28                    itemOrdered[j] = itemOrdered[j + 1];
29                }
30                itemOrdered[qtyOrdered - 1] = null; //xóa đối tượng thừa
31                qtyOrdered--;
32                System.out.println("BaoPhuc-The disc has been removed: " + disc.getTitle());
33                break;
34            }
35        }
36        if (found == false) {
37            System.out.println("BaoPhuc-The disc was not found in the cart.");
38        }
39    }
40
41
42    public float totalCost() {
43        float total = 0;
44        for (int i = 0; i < qtyOrdered; i++) {
45            total += itemOrdered[i].getCost();
46        }
47        return total;
48    }
49 }
50

```

Figure 7: Cart Class 2

## 6. Kết quả demo



```

1 package aims;
2
3 public class Aims {
4     public static void main(String[] args) {
5         Cart anOrder = new Cart();
6
7         DigitalVideoDisc dvd1 = new DigitalVideoDisc("BaoPhuc-The Lion King",
8             "Animation", "Roger Allers", 87, 19.95f);
9         anOrder.addDigitalVideoDisc(dvd1);
10
11        DigitalVideoDisc dvd2 = new DigitalVideoDisc("BaoPhuc-Star Wars",
12            "Science Fiction", "George Lucas", 87, 24.95f);
13        anOrder.addDigitalVideoDisc(dvd2);
14
15        DigitalVideoDisc dvd3 = new DigitalVideoDisc("BaoPhuc-Aladin",
16            "Animation", 18.99f);
17        anOrder.addDigitalVideoDisc(dvd3);
18
19        DigitalVideoDisc dvd4 = new DigitalVideoDisc("BaoPhuc-Two broken girls",
20            "Comedy", "BaoPhuc", 22.22f);
21        anOrder.addDigitalVideoDisc(dvd4);
22
23        System.out.println("Total cost is:");
24        System.out.println(anOrder.totalCost());
25
26        anOrder.removeDigitalVideoDisc(dvd3);
27
28        System.out.println("Total cost is:");
29        System.out.println(anOrder.totalCost());
30
31    }
32 }
33

```

Figure 8: Code Demo

Kết quả:



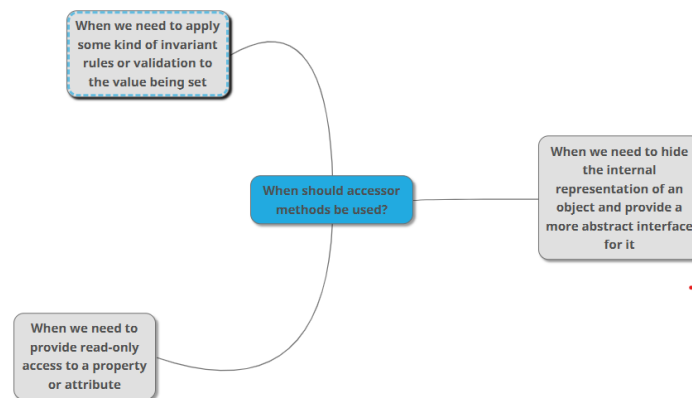
```

<terminated> Aims (1) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Oct 29, 2024, 10:19:01 PM - 10:19:02 PM) [pid: 4508]
BaoPhuc-The disc has been added: BaoPhuc-The Lion King
BaoPhuc-The disc has been added: BaoPhuc-Star Wars
BaoPhuc-The disc has been added: BaoPhuc-Aladin
BaoPhuc-The disc has been added: BaoPhuc-Two broken girls
Total cost is:
86.11
BaoPhuc-The disc has been removed: BaoPhuc-Aladin
Total cost is:
67.12

```

Figure 9: Result

## 7. Reading Assignment



*Figure 10: Getter and Setter Methods*

### 8. Answer the 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: Yes, Java allows you to create multiple constructor methods for a class, a feature known as constructor **overloading**. We can define different constructors with different parameter lists.

