

**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 tên SV: Lý Công Tiến  
MSSV: 20225934  
Lớp: Việt-Nhật 04  
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

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.....	11
7. Reading Assignment .....	13
8. Answer the question.....	13

### TABLE OF FIGURES

Figure 1: Use Case Diagram .....	5
Figure 2: Class Diagram.....	6
Figure 3: Aims Class .....	7
Figure 4: DigitalVideoDisc Class1 .....	8
Figure 5: DigitalVideoDisc Class2 .....	8
Figure 6: Cart Class1 .....	9
Figure 7: Cart Class2 .....	10
Figure 8: Cart Class3 .....	10
Figure 9: Cart Class4 .....	11
Figure 10: Code Demo1 .....	11
Figure 11: Code Demo2 .....	11
Figure 12: Code Demo3 .....	12

Figure 13: Result Demo1 .....	12
Figure 14: Result Demo2 .....	12
Figure 15: Result Demo3 .....	12
Figure 16: Getter and Setter Methods .....	13

## 1. Bài toán đặt ra

Có thể trong tương lai Tiki và Sendo sẽ đàm phán về khả năng sáp nhập để cạnh tranh với các nền tảng thương mại điện tử khác và đặc biệt là những nền tảng có sự ủng hộ nước ngoài. Việc sáp nhập hai công ty này sẽ tạo ra một công ty Ti-do, trong đó "Ti" là từ Tiki và "do" là từ Sendo, có nghĩa là một công ty tỷ đô trong tiếng Việt. Công ty Ti-do muốn bạn giúp họ tạo ra một hệ thống hoàn toàn mới cho dự án AIMS (AIMS là viết tắt của An Internet Media Store).

## 2. Yêu cầu hệ thống

### 2.1. Đối với Customer:

AIMS hiện tại chỉ có một loại phương tiện: đĩa video kỹ thuật số (DVD). Khách hàng có thể duyệt danh sách các đĩa DVD có sẵn trong cửa hàng, thứ tự hiển thị dựa trên ngày thêm vào của họ, từ mới nhất đến cũ nhất và có thể tìm kiếm DVD theo tiêu đề, thể loại hoặc giá cả. Khách hàng có thể xem thông tin chi tiết, thêm vào giỏ hàng và xem trước một phần demo của DVD. Nếu DVD có độ dài từ 0 trở xuống, hệ thống phải thông báo cho khách hàng rằng không thể phát DVD đó.

Khi khách hàng muốn xem giỏ hàng hiện tại, hệ thống sẽ hiển thị tất cả thông tin của các đĩa DVD cùng với tổng chi phí. Họ có thể sắp xếp tất cả các đĩa DVD trong giỏ hàng theo tiêu đề hoặc theo chi phí.

Khách hàng có thể cập nhật số lượng DVD trong giỏ hàng hoặc xóa DVD khỏi giỏ hàng. Để tăng doanh số bán hàng khách hàng được phép có một mặt hàng miễn phí được hệ thống chọn ngẫu nhiên trong giỏ hàng. Họ có thể lọc DVD trong giỏ hàng bằng cách cung cấp ID hoặc tiêu đề của nó. Nếu tìm thấy mặt hàng thì hiển thị thông tin trong giỏ hàng. Nếu không thì thông báo mặt hàng không được tìm thấy trong giỏ hàng hiện tại.

Khách hàng có thể đặt hàng không cần đăng nhập, cung cấp thông tin giao hàng và trả phí qua thẻ tín dụng. Sau khi thanh toán, hệ thống hiển thị chi tiết giao dịch và gửi thông tin đơn hàng qua email cho khách.

### 2.2. Đối với Store Manager:

Quản lý cửa hàng có thể đăng nhập để xem, duyệt hoặc từ chối đơn hàng. Quản lý có quyền thêm hoặc xóa DVD khỏi cửa hàng và phải cung cấp đầy đủ thông tin DVD khi thêm mới.

### 3. Use Case Diagram

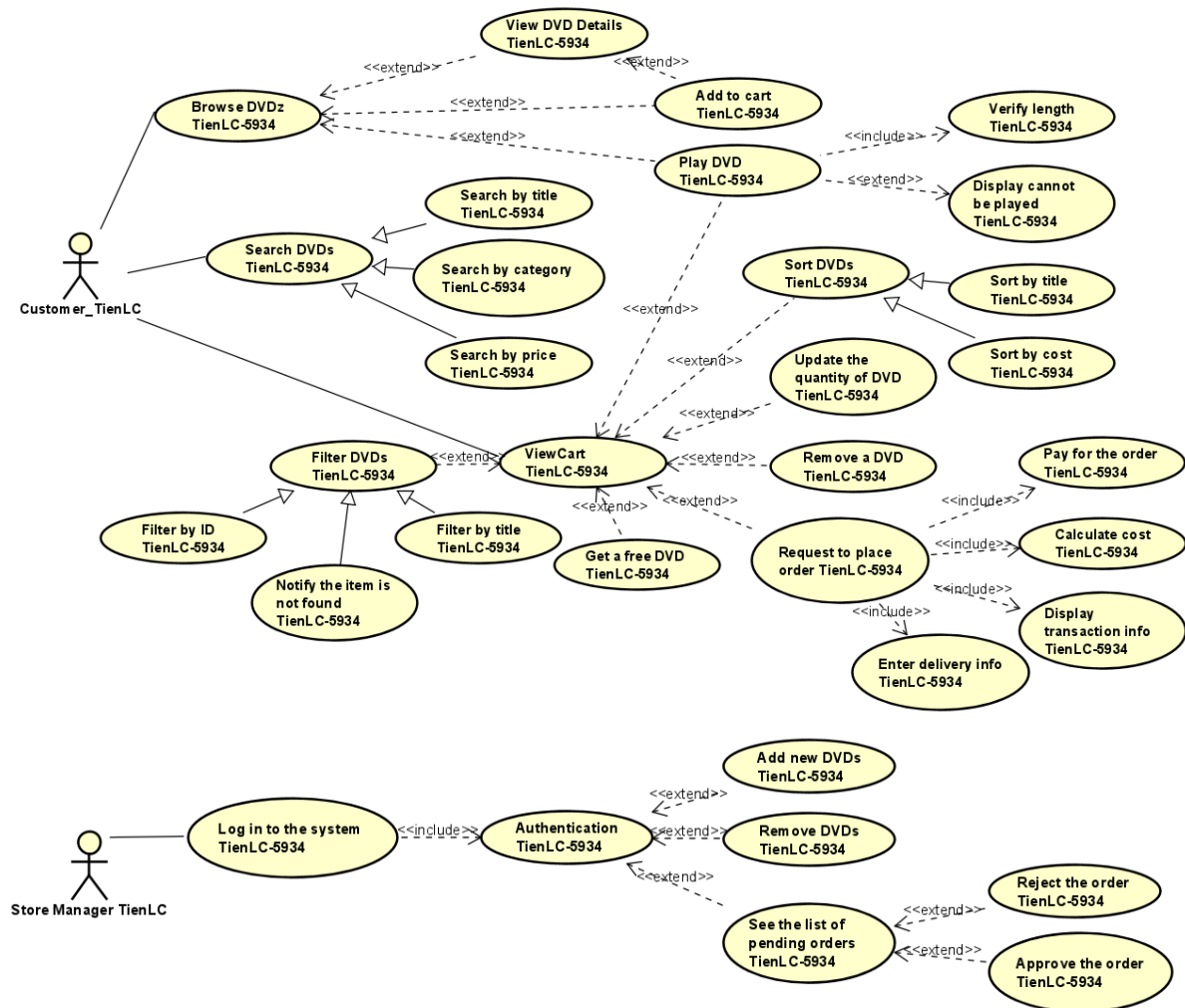


Figure 1: Use Case Diagram

## 4. Class Diagram

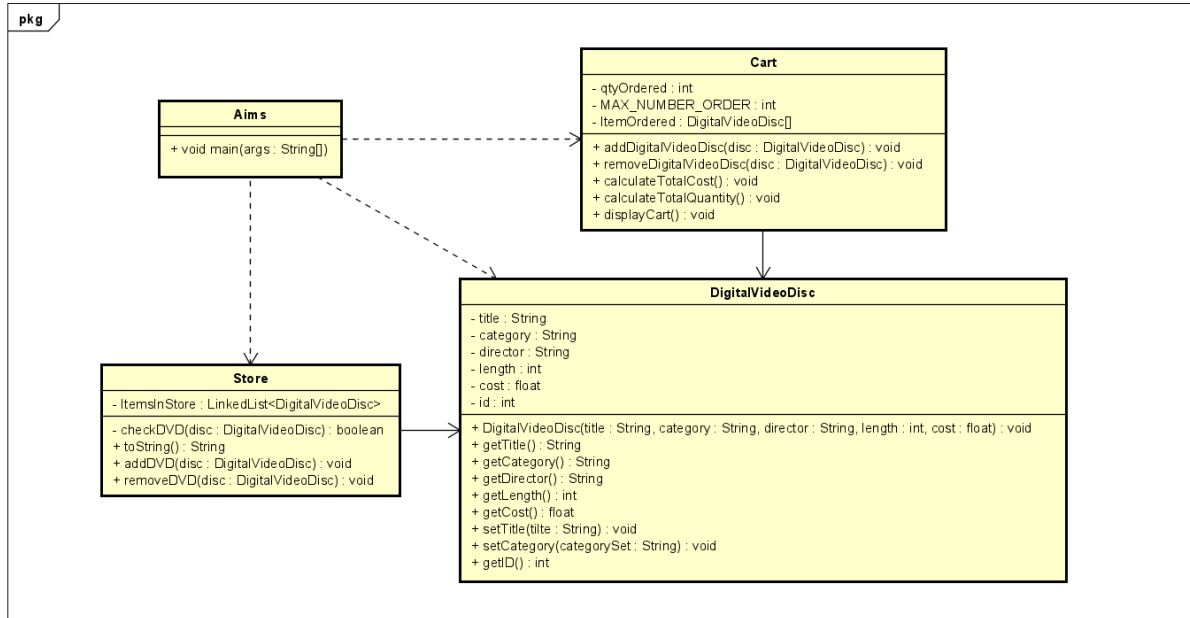


Figure 2: Class Diagram

## 5. Source Code

### 5.1. Aims Class

```
package aimsproject;

import aimsproject.DigitalVideoDiscTienLC;
import aimsproject.CartTienLC;
import aimsproject.StoreTienLc;

public class Aims {
    public static void main(String[] args) {
        CartTienLC anOrder = new CartTienLC();
    }
}
```

Figure 3: Aims Class

## 5.2. DigitalVideoDisc Class

```

3  package aimsproject;
4  public class DigitalVideoDiscTienLC {
5      private static int nbDigitalVideoDiscs = 0;
6      private String title;
7      private String category;
8      private String director;
9      private int length;
10     private float cost;
11     private int id;
12
13     public DigitalVideoDiscTienLC(String title){
14         super();
15         this.title = title;
16         this.id = ++nbDigitalVideoDiscs;
17     }
18     public DigitalVideoDiscTienLC(String title, String category, float cost){
19         this.title = title;
20         this.category = category;
21         this.cost = cost;
22         this.id = ++nbDigitalVideoDiscs;
23     }
24     public DigitalVideoDiscTienLC(String title, String category, float cost, String director){
25         this.title = title;
26         this.category = category;
27         this.cost = cost;
28         this.director = director;
29         this.id = ++nbDigitalVideoDiscs;
30     }

```

Figure 4: DigitalVideoDisc Class1

```

31     public DigitalVideoDiscTienLC(String title, String category, float cost, String director, int length){
32         this.title = title;
33         this.category = category;
34         this.cost = cost;
35         this.director = director;
36         this.length = length;
37         this.id = ++nbDigitalVideoDiscs;
38     }
39
40     public String getTitle(){ return title; }
41     public void setTitle(String title){
42         this.title = title;
43     }
44     public String getCategory(){ return category; }
45     public void setCategory(String category){
46         this.category = category;
47     }
48     public float getCost(){ return cost; }
49     public void setCost(float cost){
50         this.cost = cost;
51     }
52     public String getDirector(){ return director; }
53     public void setDirector(String director){
54         this.director = director;
55     }
56     public int getLength(){ return length; }
57     public void setLength(int length){
58         this.length = length;
59     }
60     public int getID(){ return id; }
61 }

```

Figure 5: DigitalVideoDisc Class2



### 5.3. Cart Class

```
2 package aimsproject;
3 import aimsproject.DigitalVideoDiscTienLC;
4
5 public class CartTienLC {
6     public static final int MAX_NUMBERS_ORDERED = 20;
7     private DigitalVideoDiscTienLC itemsOrdered[] = new DigitalVideoDiscTienLC[MAX_NUMBERS_ORDERED];
8     private int qtyOrdered;
9
10    public int addDigitalVideoDisc(DigitalVideoDiscTienLC disc){
11        if(qtyOrdered == MAX_NUMBERS_ORDERED){
12            System.out.println("Full cart. Can't add disc");
13            return 0;
14        }
15        else{
16            itemsOrdered[qtyOrdered] = disc;
17            qtyOrdered++;
18            System.out.println("The DVD " + "\"" + disc.getTitle() + "\"" + " have been added.");
19            return 1;
20        }
21    }
22    public int addDigitalVideoDisc(DigitalVideoDiscTienLC... dvdArray){
23        int addCount = 0;
24        for (DigitalVideoDiscTienLC disc : dvdArray){
25            if (qtyOrdered == MAX_NUMBERS_ORDERED){
26                System.out.println("The cart is almost full. Can't add more discs.");
27                break;
28            }
29            else{
30                itemsOrdered[qtyOrdered] = disc;
31                qtyOrdered++;
32                System.out.println("The DVD " + "\"" + disc.getTitle() + "\"" + " has been added.");
```

Figure 6: Cart Class1

```

33         addCount++;
34     }
35 }
36 return addCount;
37 }
38 //Ham them 2 dia DVD
39 public int addDigitalVideoDisc(DigitalVideoDiscTienLC dvd1, DigitalVideoDiscTienLC dvd2) {
40     if (qtyOrdered + 1 >= MAX_NUMBERS_ORDERED) {
41         System.out.println("The cart is almost full. Can't add more discs.");
42         return 0;
43     }
44     else{
45         itemsOrdered[qtyOrdered] = dvd1;
46         qtyOrdered++;
47         System.out.println("The DVD " + "\"" + dvd1.getTitle() + "\" + " has been added.");
48         itemsOrdered[qtyOrdered] = dvd2;
49         qtyOrdered++;
50         System.out.println("The DVD " + "\"" + dvd2.getTitle() + "\" + " has been added.");
51         return 2; //Tra ve so dia DVD da them duoc
52     }
53 }
54 public int removeDigitalVideoDisc(DigitalVideoDiscTienLC disc){
55     if(itemsOrdered[0] == null){
56         System.out.println("Your cart is empty.");
57         return 0;
58     }
59     for(int i = 0; i < qtyOrdered; i++){
60         if(itemsOrdered[i].equals(disc)){
61             for(int j = i; j < qtyOrdered - 1; j++){
62                 itemsOrdered[j] = itemsOrdered[j+1];
63             }
64             itemsOrdered[qtyOrdered - 1] = null;

```

Figure 7: Cart Class2

```

65         qtyOrdered--;
66         System.out.println("Remove DVD " + "\"" + disc.getTitle() + "\" + " done.");
67         return 1;
68     }
69 }
70 System.out.println("No DVD match.");
71 return 0;
72 }
73 public float totalCost(){
74     float sum = 0.00f;
75     for(int i = 0; i < qtyOrdered; i++){
76         sum += itemsOrdered[i].getCost();
77     }
78     return sum;
79 }
80 public void print(){
81     StringBuilder output = new StringBuilder("*****CART*****\nOrdered items: \n");
82     for(int i = 0; i < qtyOrdered; i++){
83         output.append(i+1 + ". [" + itemsOrdered[i].getTitle() + "] - [" + itemsOrdered[i].getCategory() + "] - ["
84             + itemsOrdered[i].getDirector() + "] - [" + itemsOrdered[i].getLength() + "]: "
85             + itemsOrdered[i].getCost() + " $\n");
86     }
87     output.append("total: ").append(totalCost()).append(" $\n");
88     output.append("*****\n");
89     System.out.println(output);
90 }

```

Figure 8: Cart Class3

```

90     }
91     public void searchById(int i){
92         if(i > qtyOrdered){
93             System.out.println("No match found.");
94         }
95         else{
96             System.out.println("Result: " + "[" + itemsOrdered[i-1].getTitle() +
97                 "]" - "[" + itemsOrdered[i-1].getCategory() + "]" - "["
98                 + itemsOrdered[i-1].getDirector() + "]" - "["
99                 + itemsOrdered[i-1].getLength() + "]: " + itemsOrdered[i-1].getCost() + " $" + "\n");
100         }
101     }
102     public void searchByTitle(String title){
103         for(int i = 0; i < qtyOrdered; i++){
104             if(itemsOrdered[i].getTitle().equals(title)){
105                 System.out.println("Result: " + "[" + itemsOrdered[i].getTitle() +
106                     "]" - "[" + itemsOrdered[i].getCategory() + "]" - "["
107                     + itemsOrdered[i].getDirector() + "]" - "["
108                     + itemsOrdered[i].getLength() + "]: " + itemsOrdered[i].getCost() + " $" + "\n");
109                 return;
110             }
111         }
112         System.out.println("No match found.");
113     }
114 }

```

Figure 9: Cart Class4

## 6. Kết quả demo

```

8     public class Aims {
9         public static void main(String[] args) {
10             CartTienLC anOrder = new CartTienLC();
11
12             //Them dia vao gio hang
13             DigitalVideoDiscTienLC dvd1 = new DigitalVideoDiscTienLC("Tom and Jerry", "Cartoon", 20.05f, "Gene Deitch", 120);
14             DigitalVideoDiscTienLC dvd2 = new DigitalVideoDiscTienLC("The Lion king", "anime", 15.55f, "Roger Allers", 87);
15             DigitalVideoDiscTienLC dvd3 = new DigitalVideoDiscTienLC("Toy Story", "anime", 17.75f, "John Lasseter", 150);
16             anOrder.addDigitalVideoDisc(dvd1, dvd2, dvd3);
17
18             //Kiem tra khi chua thuc hien ham xoa
19             anOrder.print();
20         }

```

Figure 10: Code Demo1

```

12         //Them dia vao gio hang
13         DigitalVideoDiscTienLC dvd1 = new DigitalVideoDiscTienLC("Tom and Jerry", "Cartoon", 20.05f, "Gene Deitch", 120);
14         DigitalVideoDiscTienLC dvd2 = new DigitalVideoDiscTienLC("The Lion king", "anime", 15.55f, "Roger Allers", 87);
15         DigitalVideoDiscTienLC dvd3 = new DigitalVideoDiscTienLC("Toy Story", "anime", 17.75f, "John Lasseter", 150);
16         anOrder.addDigitalVideoDisc(dvd1, dvd2, dvd3);
17
18         //Kiem tra khi chua thuc hien ham xoa
19         anOrder.print();
20         //Kiem tra khi da thuc hien ham xoa
21         anOrder.removeDigitalVideoDisc(dvd2);
22         anOrder.print();
23     }

```

Figure 11: Code Demo2

```

8   public class Aims {
9       public static void main(String[] args) {
10           CartTienLC anOrder = new CartTienLC();
11
12           //Them dia vao gio hang
13           DigitalVideoDiscTienLC dvd1 = new DigitalVideoDiscTienLC("Tom and Jerry", "Cartoon", 20.05f, "Gene Deitch", 120);
14           DigitalVideoDiscTienLC dvd2 = new DigitalVideoDiscTienLC("The Lion king", "anime", 15.55f, "Roger Allers", 87);
15           DigitalVideoDiscTienLC dvd3 = new DigitalVideoDiscTienLC("Toy Story", "anime", 17.75f, "John Lasseter", 150);
16           anOrder.addDigitalVideoDisc(dvd1, dvd2, dvd3);
17           //Kiem tra khi chua thuc hien ham xoa
18           anOrder.print();
19           //Kiem tra khi da thuc hien ham xoa
20           anOrder.removeDigitalVideoDisc(dvd2);
21           anOrder.print();
22           //Tim kiem DVD trong gio hang sau khi xoa
23           anOrder.searchById(3);
24       }
25   }

```

Figure 12: Code Demo3

Kết quả:

```

run:
The DVD "Tom and Jerry" has been added.
The DVD "The Lion king" has been added.
The DVD "Toy Story" has been added.
*****CART*****
Ordered items:
1.[Tom and Jerry] - [Cartoon] - [Gene Deitch] - [120]: 20.05 $
2.[The Lion king] - [anime] - [Roger Allers] - [87]: 15.55 $
3.[Toy Story] - [anime] - [John Lasseter] - [150]: 17.75 $
total: 53.35 $
*****

```

Figure 13: Result Demo1

```

Remove DVD "The Lion king" done.
*****CART*****
Ordered items:
1.[Tom and Jerry] - [Cartoon] - [Gene Deitch] - [120]: 20.05 $
2.[Toy Story] - [anime] - [John Lasseter] - [150]: 17.75 $
total: 37.8 $
*****

```

Figure 14: Result Demo2

```
No match found.
```

Figure 15: Result Demo3

## 7. Reading Assignment

When should accessor methods be used? (Khi nào nên sử dụng các phương thức truy cập)

- Các phương thức truy cập (accessor methods) – getter và setter được sử dụng khi:
  - + Cần bảo vệ dữ liệu, getter và setter cho phép truy cập và chỉnh sửa giá trị mà vẫn đảm bảo tính an toàn cho dữ liệu bên trong lớp.
  - + Được dùng để kiểm tra và xác minh giá trị trước khi gán cho thuộc tính, tránh lỗi hoặc các giá trị không hợp lệ (setter).
  - + Được dùng để chỉ định quyền truy cập (getter và setter).

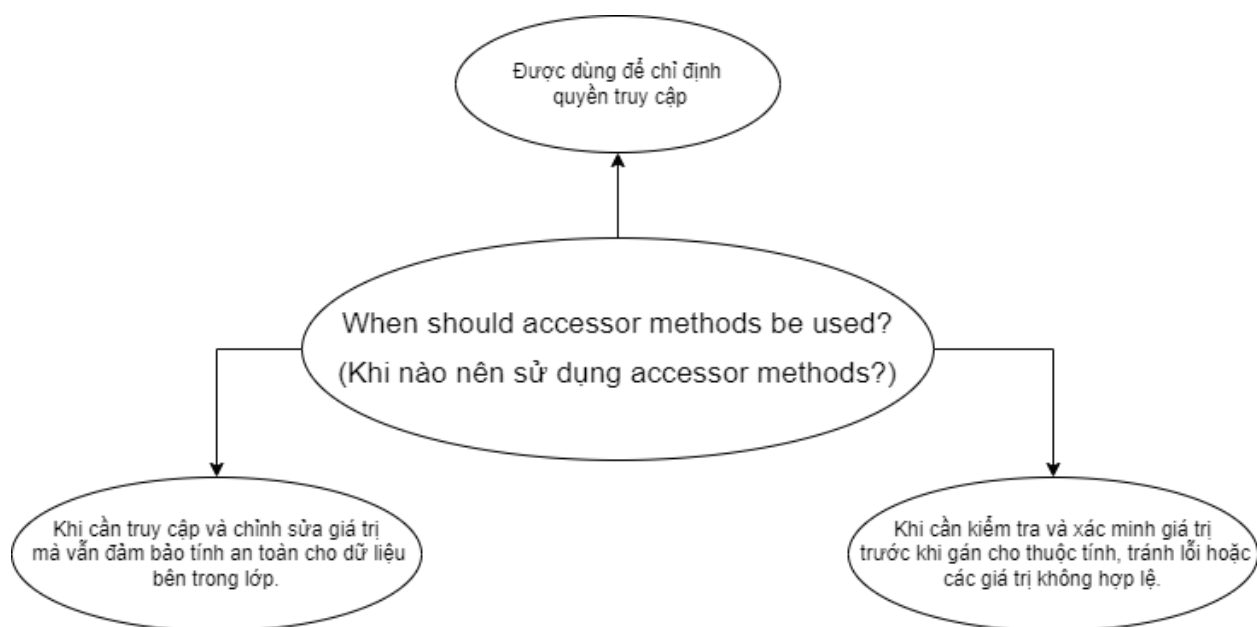


Figure 16: Getter and Setter Methods

## 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 ?

(Nếu bạn tạo một phương thức constructor để khởi tạo một đối tượng DVD bằng tiêu đề, sau đó tạo một phương thức constructor để khởi tạo một đối tượng DVD bằng thể loại. Java có cho phép bạn làm điều này không ?)

→ Có . Java cho phép thực hiện thông qua method overloading (nạp chồng phương thức).