

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



**BÁO CÁO THỰC HÀNH – LAB02**

**IT3103-744528-2024.1**

**BÀI THỰC HÀNH - LAB02**

**Họ và tên sv : Nguyễn Trọng Hình**

**Mssv : 20225842**

**Lớp : Việt Nhật 04 – K67**

**GVHD : Lê Thị Hoa**

**HTGD : Đặng Mạnh Cường**

# BÁO CÁO THỰC HÀNH LAB 2

## LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

### Mục Lục

<b>Danh sách hình ảnh .....</b>	<b>2</b>
<b>1. Bài toán đặt ra .....</b>	<b>3</b>
<b>2. Yêu cầu của hệ thống .....</b>	<b>3</b>
2.1 Đối với Customer (Khách hàng): .....	3
2.2 Đối với Store Manager (Người quản lý cửa hàng):.....	3
<b>3. Use Case Diagram.....</b>	<b>3</b>
<b>4. Class Diagram .....</b>	<b>4</b>
<b>5. Source Code .....</b>	<b>5</b>
5.1. Aims Class .....	5
5.2. DigitalVideoDisc Class.....	6
5.3. Cart Class .....	7
<b>6. Kết quả demo .....</b>	<b>8</b>
<b>7. Reading Assignment.....</b>	<b>8</b>
<b>8. Answer the question .....</b>	<b>9</b>

### Danh sách hình ảnh

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

## 1. Bài toán đặt ra

- Dự án AIMS (An Internet Media Store) yêu cầu xây dựng một hệ thống quản lý cửa hàng trực tuyến, nơi khách hàng có thể tìm kiếm và đặt mua các sản phẩm là DVD. Trong hệ thống này, khách hàng có thể duyệt danh sách DVD, tìm kiếm và sắp xếp DVD theo nhiều tiêu chí, thêm sản phẩm vào giỏ hàng, và thực hiện thanh toán. Bên cạnh đó, hệ thống cũng hỗ trợ người quản lý cửa hàng trong việc quản lý các đơn hàng và cập nhật danh mục sản phẩm.

## 2. Yêu cầu của hệ thống

### 2.1 Đối với Customer (Khách hàng):

- **Duyệt và tìm kiếm DVD:** Khách hàng có thể tìm kiếm DVD theo tiêu đề, danh mục, hoặc giá và xem thông tin chi tiết của DVD.
- **Xem giỏ hàng:** Khách hàng có thể xem các DVD trong giỏ hàng cùng với tổng chi phí, sắp xếp DVD trong giỏ hàng theo tiêu đề hoặc chi phí.
- **Cập nhật giỏ hàng:** Khách hàng có thể thêm, xóa, hoặc thay đổi số lượng DVD trong giỏ hàng. Hệ thống còn chọn ngẫu nhiên một DVD trong giỏ hàng làm quà tặng miễn phí.
- **Đặt hàng:** Khách hàng có thể yêu cầu đặt hàng, nhập thông tin giao hàng và thực hiện thanh toán qua thẻ tín dụng.
- **Lọc và xem DVD trong giỏ hàng:** Khách hàng có thể lọc DVD trong giỏ hàng bằng ID hoặc tiêu đề và xem thông tin chi tiết.

### 2.2 Đối với Store Manager (Người quản lý cửa hàng):

- **Đăng nhập:** Người quản lý cần đăng nhập để truy cập chế độ quản lý.
- **Quản lý đơn hàng:** Người quản lý có thể xem các đơn đặt hàng đang chờ xử lý và phê duyệt hoặc từ chối các đơn hàng.
- **Quản lý DVD trong cửa hàng:** Người quản lý có thể thêm DVD mới vào cửa hàng hoặc xóa DVD khỏi cửa hàng.

## 3. Use Case Diagram

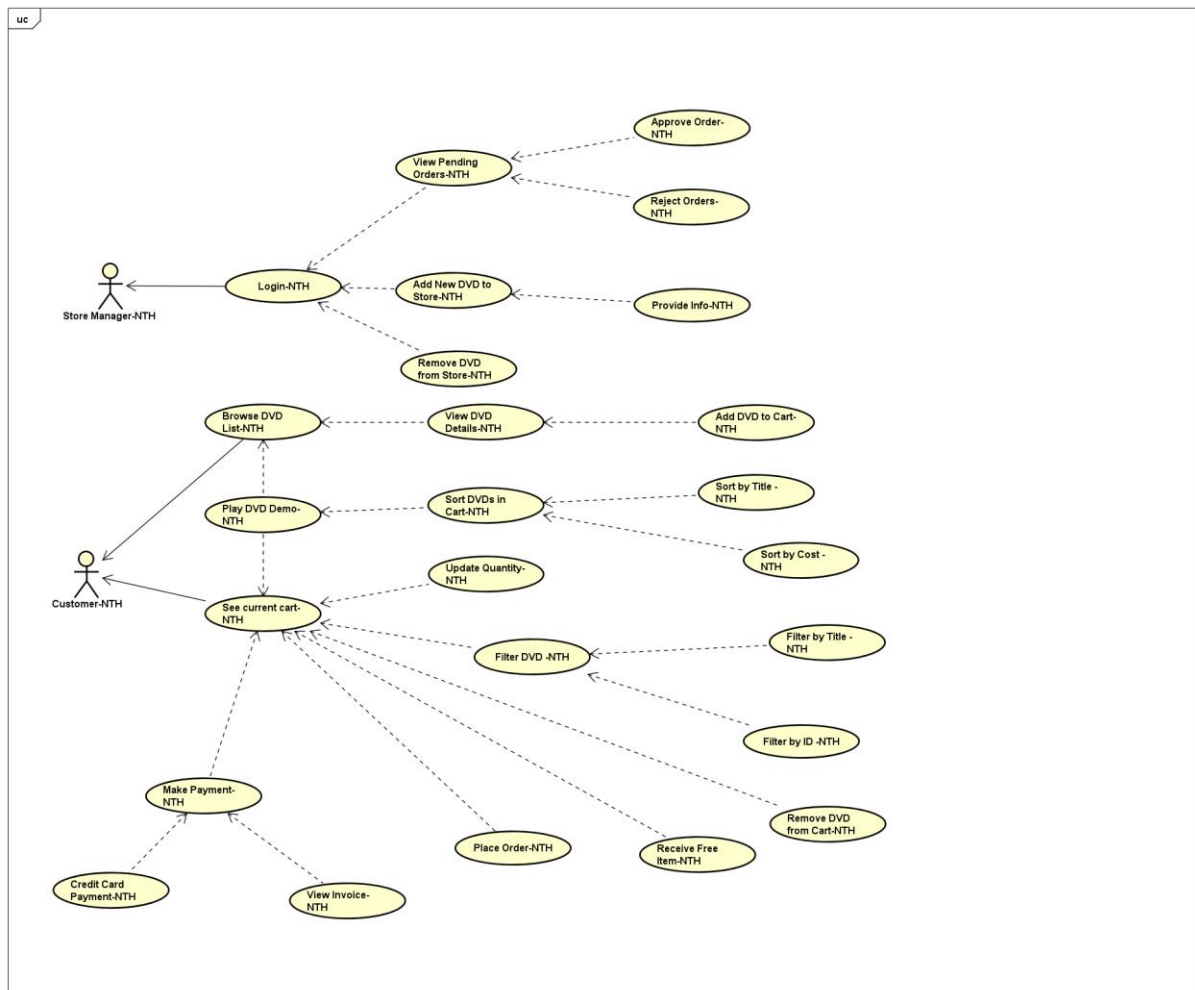
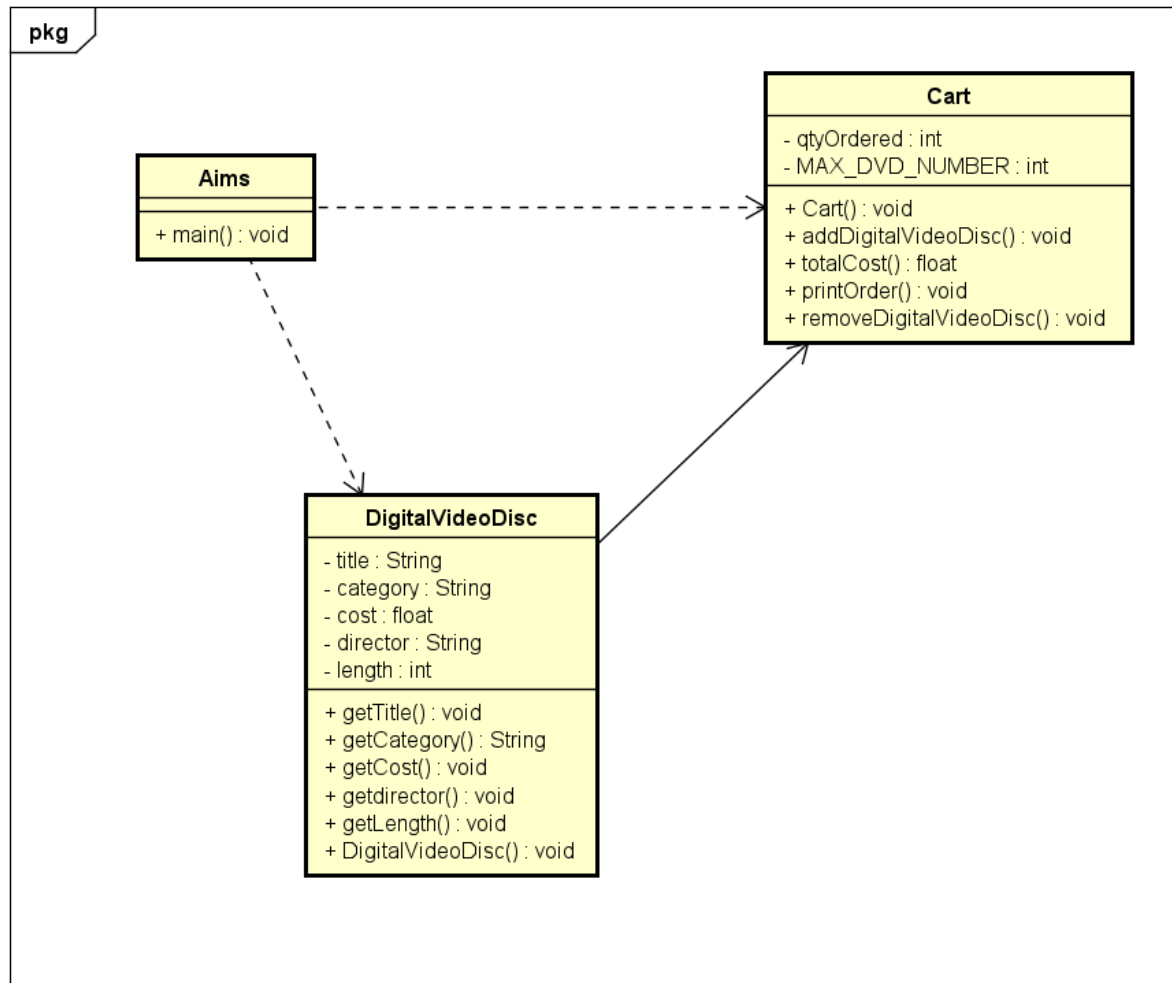


Figure 1: Use Case Diagram

## 4. Class Diagram

*Figure 2: Class Diagram*

## 5. Source Code

### 5.1. Aims Class

```

J Aims.java > Language Support for Java(TM) by Red Hat > Aims
1  public class Aims {
    Run main | Debug main | Run | Debug
2      public static void main(String [] args) {
3          Cart anOrder = new Cart();
4
5          DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"One Piece", category:"Anime", director:"Oda", length:148, cost:29.5f);
6          anOrder.addDigitalVideoDisc(dvd1);
7
8          DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Songoku", category:"Action", director:"Toriyama", length:136, cost:19.0f);
9          anOrder.addDigitalVideoDisc(dvd2);
10
11         DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"Naruto", category:"Anime", director:"Kishimoto", length:169, cost:24.5f);
12         anOrder.addDigitalVideoDisc(dvd3);
13
14         System.out.println(x:"NTH-Total cost is: ");
15         System.out.println(anOrder.totalCost());
16
17         anOrder.removeDigitalVideoDisc(dvd2);
18
19         System.out.println(x:"NTH-Total cost is: ");
20         System.out.println(anOrder.totalCost());
21     }
22 }
23

```

Figure 3: Aims Class

## 5.2. DigitalVideoDisc Class

```

import java.lang.reflect.Constructor;

public class DigitalVideoDisc {

    private String title;
    private String category;
    private String director;
    private int length;
    private double cost;

    public String getTitle() {
        return title;
    }

    public String getCategory() {
        return category;
    }

    public String getDirector() {
        return director;
    }

    public int getLength() {
        return length;
    }

    public double getCost() {
        return cost;
    }

    //Constructor with title, category, and cost parameters
    public DigitalVideoDisc(String title, String category, float cost) {
        this.title = title;
        this.category = category;
        this.cost = cost;
    }
}

```

Figure 4: DigitalVideoDisc Class 1

```

public DigitalVideoDisc(String title) {
    this.title = title;
}

public DigitalVideoDisc(String title, String category, String director, float cost) {
    this.title = title;
    this.category = category;
    this.director = director;
    this.cost = cost;
}

public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
    this.title = title;
    this.category = category;
    this.director = director;
    this.length = length;
    this.cost = cost;
}
}

```

Figure 5: DigitalVideoDisc Class 2

### 5.3. Cart Class

```

public class Cart {
    public static final int MAX_NUMBERS_ORDERED = 20;
    private DigitalVideoDisc[] dvds;
    private int qtyOrdered;

    public Cart() {
        dvds = new DigitalVideoDisc[MAX_NUMBERS_ORDERED];
        qtyOrdered = 0;
    }

    // Add a DVD to the cart
    public void addDigitalVideoDisc(DigitalVideoDisc disc) {
        if (qtyOrdered < MAX_NUMBERS_ORDERED) {
            dvds[qtyOrdered] = disc;
            qtyOrdered++;
            System.out.println("NTH-The DVD \"\" + disc.getTitle() + "\" has been successfully added to the cart.");
        } else {
            System.out.println(x:"NTH-The cart is full.");
        }
    }

    // Remove a DVD from the cart
    public void removeDigitalVideoDisc(DigitalVideoDisc disc) {
        for (int i = 0; i < qtyOrdered; i++) {
            if (dvds[i].equals(disc)) {
                for (int j = i; j < qtyOrdered - 1; j++) {
                    dvds[j] = dvds[j + 1];
                }
                dvds[qtyOrdered - 1] = null;
                qtyOrdered--;
                System.out.println("NTH-The disc \"\" + disc.getTitle() + "\" has been removed from the cart.");
                return;
            }
        }
        System.out.println(x:"NTH-DVD not found in the cart.");
    }
}

```

Figure 6: Cart Class 1

- addDigitalVideoDisc mục đích để thêm một DVD vào giỏ hàng
- removeDigitalVideoDisc mục đích để xóa một DVD khỏi giỏ hàng

```
// Calculate the total cost of DVDs in the cart
public float totalCost() {
    float totalCost = 0;
    for (int i = 0; i < qtyOrdered; i++) {
        totalCost += dvds[i].getCost();
    }
    return totalCost;
}

// Print the list of DVDs in the cart along with their costs
public void print() {
    System.out.println("NTH-Cart contents: ");
    for (int i = 0; i < qtyOrdered; i++) {
        System.out.println(dvds[i].getTitle() + " - $" + dvds[i].getCost());
    }
    System.out.println("NTH-Total cost: $" + totalCost());
}
}
```

Figure 7: Cart Class 2

- totalCost có nhiệm vụ tính toán tổng chi phí của tất cả các đĩa DVD có trong giỏ hàng.
- print có nhiệm vụ in ra danh sách các đĩa DVD có trong giỏ hàng, cùng với giá của từng đĩa.

## 6. Kết quả demo

Kết quả :

```
NTH-The DVD "One Piece" has been successfully added to the cart.
NTH-The DVD "Songoku" has been successfully added to the cart.
NTH-The DVD "Naruto" has been successfully added to the cart.
NTH-Total cost is:
73.0
NTH-The disc "Songoku" has been removed from the cart.
NTH-Total cost is:
54.0
```

Figure 8: Result

## 7. Reading Assignment



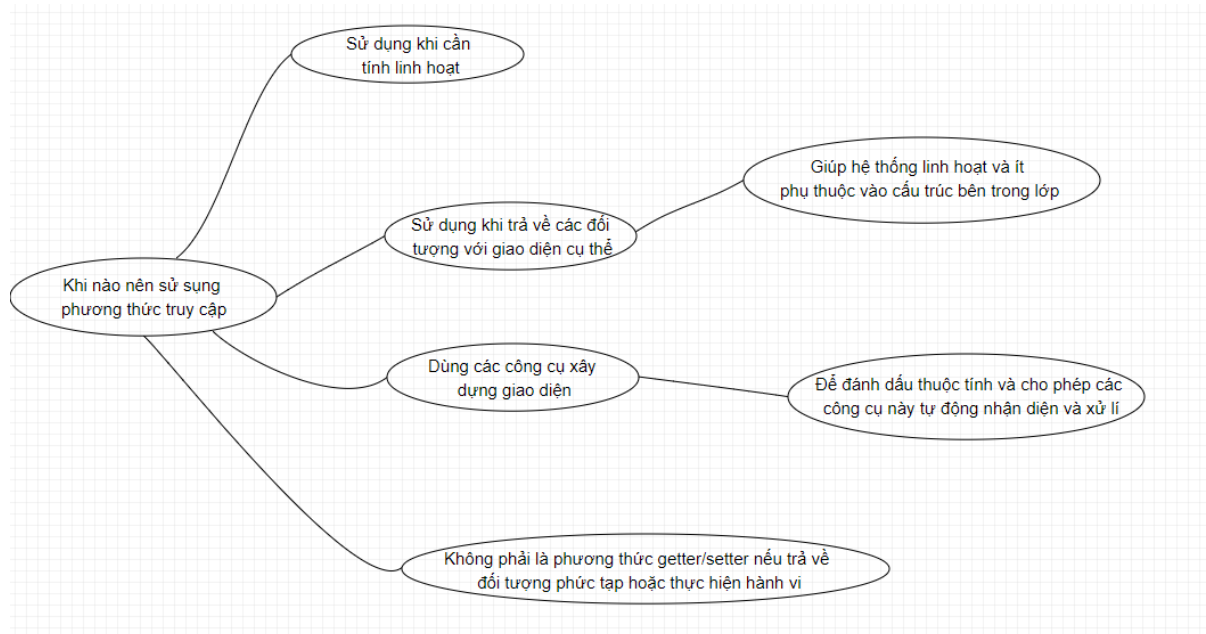


Figure 9: Getter and Setter Methods

## 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?

Java không chấp nhận, vì sẽ gây xung đột với constructor method có tham số là String title

Nếu bạn có một constructor nhận tham số là String title và một constructor khác cũng nhận tham số là String category, khi bạn truyền một tham số kiểu String cho hàm khởi tạo, Java sẽ không biết nên gọi constructor nào vì cả hai đều có kiểu tham số giống nhau. Điều này dẫn đến xung đột và gây lỗi biên dịch.