

BÁO CÁO THỰC HÀNH LAB 2 LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

1. Đề bài	1
2. Yêu cầu bài toán.....	1
3. Use case diagram	3
4. UML Class Diagram for use cases related to cart management	4
6,7,8. Create the DigitalVideoDisc class and its attributes, accessors, Constructor method	4
Reading Assignment :	6
Question 8. : 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?	7
9. Create the Cart class to work with DigitalVideoDisc.....	7
10,11. Create Carts of DigitalVideoDiscs and check the remove method	9

Table of Figure

Figure 1: UseCase Diagram	3
Figure 2 Class Diagram	4
Figure 3 : Mind map getter	7
Figure 4: Code class Aims.....	9
Figure 5: Kết quả	9

1. Đề bài

Thiết kế hệ thống mới cho dự án AIMS (hiện tại chỉ có 1 phương tiện : DVD)

2. Yêu cầu bài toán

Yêu Cầu của Người Mua Hàng:

1. Xem Danh Sách DVD:

- Người mua có thể xem danh sách DVD có sẵn trong cửa hàng trên trang chủ.
- Danh sách sẽ được sắp xếp theo thứ tự ngày thêm mới, từ mới nhất đến cũ nhất.
- Khách hàng có thể phát một đĩa DVD cụ thể; tuy nhiên, các đĩa có thời lượng là 0 hoặc âm không thể phát được.

2. Tìm Kiếm DVD:

- Người mua có thể tìm kiếm DVD theo ba phương pháp: theo tiêu đề, theo thể loại hoặc theo giá.
- Khi tìm kiếm theo tiêu đề, họ nhập một chuỗi từ khóa. Nếu tiêu đề của DVD chứa bất kỳ từ khóa nào trong chuỗi, DVD đó sẽ được hiển thị.
- Khi tìm kiếm theo thể loại, họ nhập tên thể loại. Nếu DVD có thể loại tương ứng, nó sẽ được hiển thị.
- Khi tìm kiếm theo giá, họ có thể nhập giá tối thiểu và tối đa hoặc chỉ giá tối đa.

3. Xem Chi Tiết và Thêm vào Giỏ Hàng:

- Người mua có thể xem thông tin chi tiết của một DVD và thêm vào giỏ hàng từ danh sách hoặc trang chi tiết của DVD.

4. Quản Lý Giỏ Hàng:

- Khách hàng có thể thêm các đĩa DVD vào giỏ hàng từ danh sách hoặc từ trang chi tiết.
- Hệ thống hiển thị tất cả các đĩa DVD trong giỏ cùng với tổng giá.
- Khách hàng có thể nghe một đĩa DVD trong giỏ trước khi xác nhận đơn hàng.
- Khách hàng có thể cập nhật số lượng đĩa DVD trong giỏ hoặc loại bỏ chúng.
- Hệ thống tự động chọn ngẫu nhiên một sản phẩm để khách hàng nhận miễn phí.

5. Đặt Hàng:

- Người mua có thể yêu cầu đặt hàng khi đang xem giỏ hàng.
- Họ sẽ nhập thông tin giao hàng và hướng dẫn giao hàng.
- Hệ thống sẽ tính phí giao hàng dựa trên trọng lượng tổng đơn hàng và địa chỉ giao hàng.
- Sau khi thanh toán bằng thẻ tín dụng, họ sẽ nhận được hóa đơn chi tiết qua email và đơn hàng sẽ ở trạng thái chờ xác nhận.

Yêu Cầu của Người Quản Lý Cửa Hàng:

1. Đăng Nhập và Quản Lý Đơn Hàng:

- Người quản lý cửa hàng cần đăng nhập vào hệ thống để truy cập chế độ quản lý.
- Người quản lý có thể xem danh sách các đơn hàng đang chờ xử lý.
- Có thể chọn một đơn hàng để xem chi tiết và duyệt hoặc từ chối đơn hàng đó.

2. Thêm và Xóa Đĩa DVD:

- Người quản lý có thể thêm đĩa DVD mới vào cửa hàng.
- Cần cung cấp thông tin đầy đủ về đĩa DVD mới, bao gồm ID, tiêu đề, danh mục, đạo diễn, thời lượng và giá.
- Có khả năng xóa các đĩa DVD khỏi cửa hàng.

3. Xác Nhận Đơn Hàng và Thanh Toán:

- Người quản lý xác nhận hoặc từ chối các đơn hàng đang chờ xử lý.
- Hệ thống tính phí giao hàng dựa trên tổng khối lượng đơn hàng và địa chỉ giao hàng.
- Hiển thị hóa đơn cho khách hàng, bao gồm danh sách các đĩa DVD, tổng chi phí trước và sau thuế VAT, cùng với phí giao hàng.
- Chỉ có một phương thức thanh toán được chấp nhận, đó là thẻ tín dụng. Hệ thống kết nối với hệ thống liên kết thẻ để kiểm tra tính hợp lệ của thẻ hoặc thực hiện giao dịch thanh toán.
- Sau giao dịch, hiển thị thông tin chi tiết như mã giao dịch, chủ thẻ, số tiền giao dịch, thông báo giao dịch, số dư, ngày giao dịch cho khách hàng.

3. Use case diagram

Đối với yêu cầu bài toán như đã mô tả ở phần 2 gồm 2 actor là người mua hàng và người quản lý cửa hàng, ta có use case như hình dưới đây :

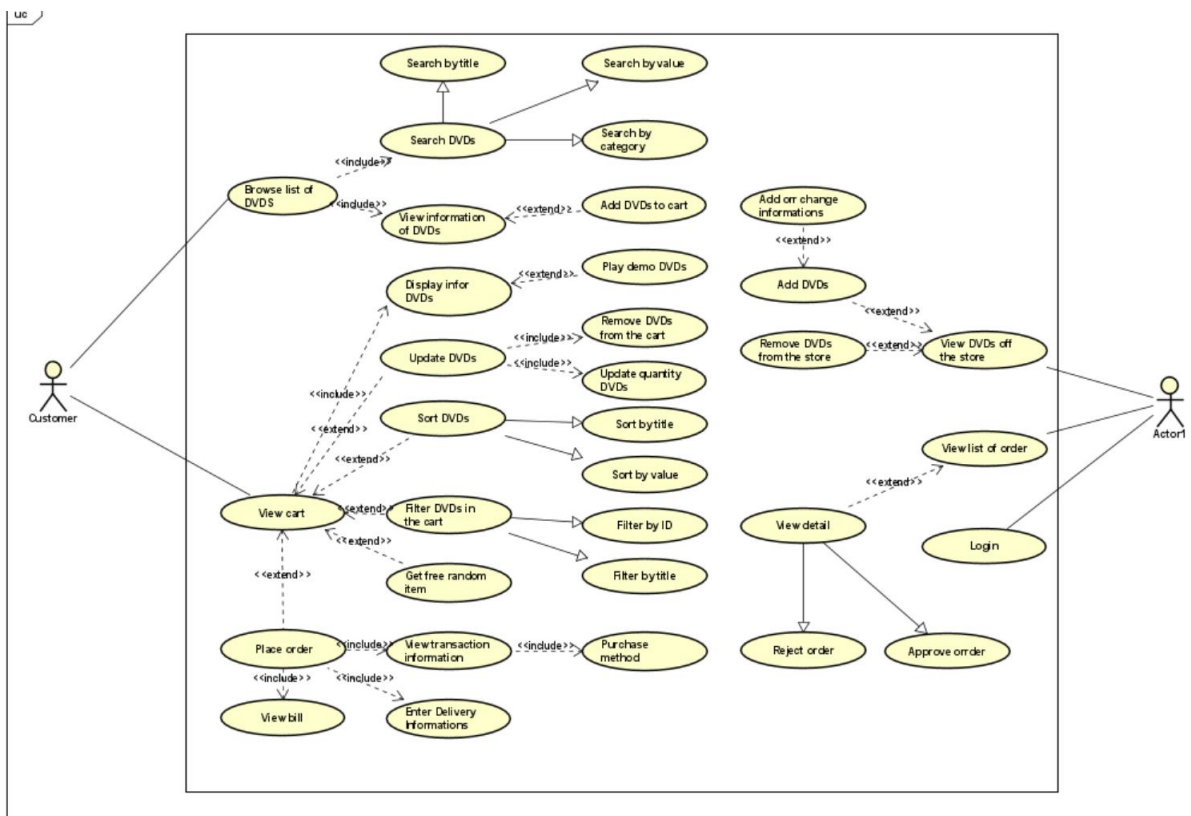


Figure 1: UseCase Diagram

4. UML Class Diagram for use cases related to cart management

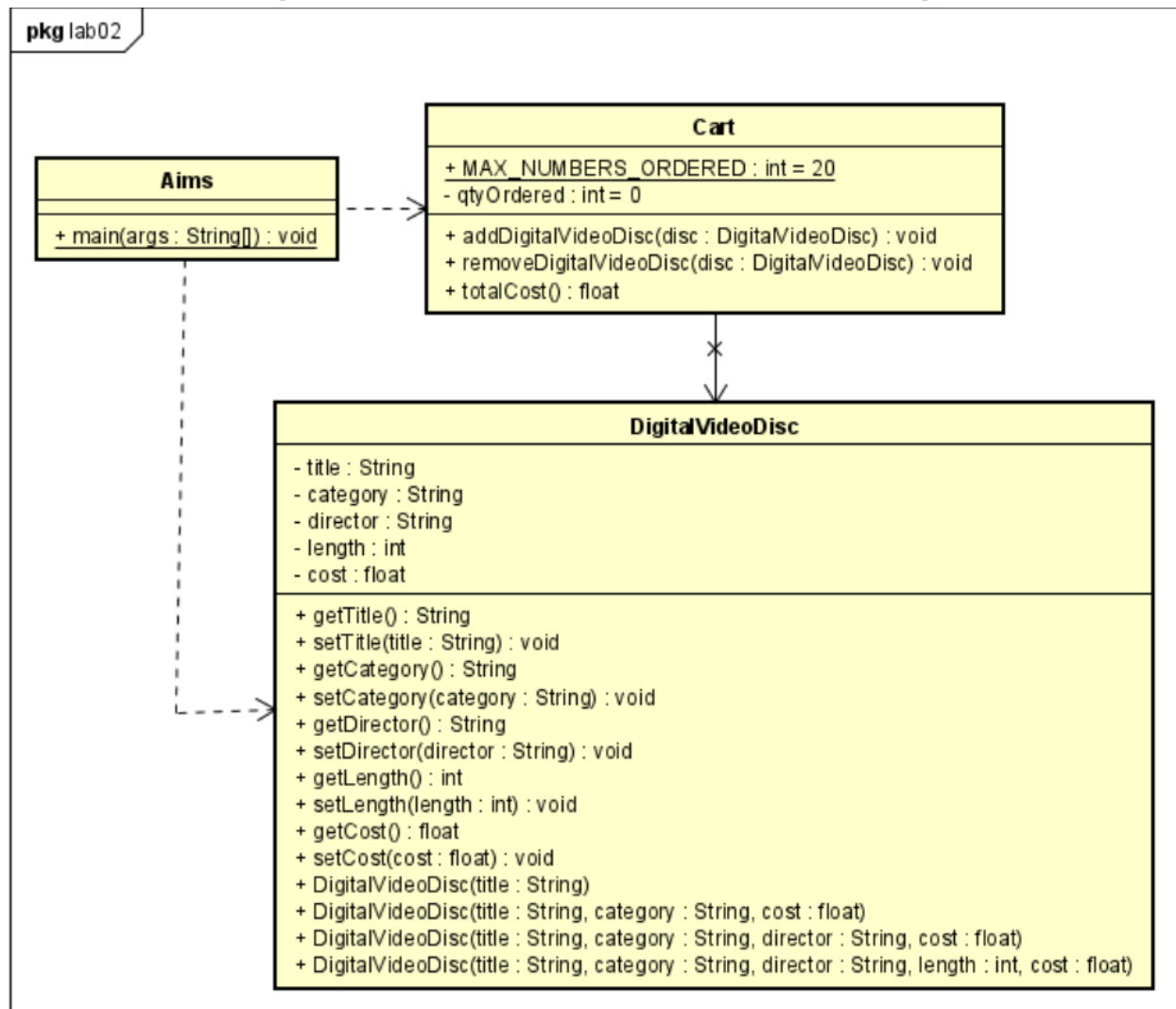


Figure 2 Class Diagram

6,7,8. Create the DigitalVideoDisc class and its attributes, accessors, Constructor method

```
package lab02;
```

```
public class DigitalVideoDisc {
    private String title;
    private String category;
    private String director;
    private int length;
    private float cost;
```

```
// getter setter của các thuộc tính trên
public String getTitle() {
    return title;
}

public void setTitle(String title) {
    this.title = title;
}

public String getCategory() {
    return category;
}

public void setCategory(String category) {
    this.category = category;
}

public String getDirector() {
    return director;
}

public void setDirector(String director) {
    this.director = director;
}

public int getLength() {
    return length;
}

public void setLength(int length) {
    this.length = length;
}

public float getCost() {
    return cost;
}

public void setCost(float cost) {
    this.cost = cost;
}

// các constructor
public DigitalVideoDisc(String title) {
    super();
    this.title = title;
}
```

```
}

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

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;
}
}
```

Reading Assignment :

Accessor methods should be used whenever you want to provide controlled, secure, and abstracted access to the private fields of a class. They are a fundamental practice in object-oriented programming and are crucial for building maintainable, flexible, and secure code.



Figure 3 : Mind map getter

Question 8. : 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?

- Không , vì khi đấy sẽ bị giống chữ ký và không chồng phương thức được, có thể trùng tên nhưng không thể trùng chữ ký

9. Create the Cart class to work with DigitalVideoDisc

```
package lab02;

public class Cart {
    public static final int MAX_NUMBERS_ORDERED = 20;
    private DigitalVideoDisc itemsOrdered[] = new DigitalVideoDisc[MAX_NUMBERS_ORDERED];

    private int qtyOrdered = 0; // to show quantity in the cart
    // Method to add a DigitalVideoDisc to the cart
    public void addDigitalVideoDisc(DigitalVideoDisc disc) {
        if(qtyOrdered < MAX_NUMBERS_ORDERED) {
            itemsOrdered[qtyOrdered] = disc;
            qtyOrdered++;
            System.out.println("The disc has been added");
        }
        else System.out.println("The cart is almost full");
    }
}
```

```
}  
// Method to remove a DigitalVideoDisc to the cart  
public void removeDigitalVideoDisc(DigitalVideoDisc disc) {  
    boolean found = false; // Sử dụng biến boolean để theo dõi xem đĩa đã được tìm thấy hay không  
    for (int i = 0; i < qtyOrdered; i++) {  
        if (itemsOrdered[i] != null && itemsOrdered[i].equals(disc)) {  
            found = true; // Đặt giá trị found thành true khi tìm thấy đĩa  
            itemsOrdered[i] = itemsOrdered[qtyOrdered - 1]; // Di chuyển đĩa cuối cùng lên vị trí của đĩa  
            cần xóa  
            itemsOrdered[qtyOrdered - 1] = null; // Gán giá trị null cho phần tử cuối cùng  
            qtyOrdered--; // Giảm số lượng đĩa trong danh sách  
            System.out.println("The disc has been removed.");  
            break;  
        }  
    }  
    if (!found) {  
        System.out.println("The disc was not found in the cart.");  
    }  
}  
// Method to calculate totalcost of the items in the cart  
public float totalCost() {  
    float sum = 0.0f;  
    for(int i = 0; i < qtyOrdered; i++) {  
        sum += itemsOrdered[i].getCost();  
    }  
    return Float.parseFloat(String.format("%.2f", sum));  
}  
}
```


10,11. Create Carts of DigitalVideoDiscs and check the remove method

```
package lab02;

new *
public class Aims {
    new *
    public static void main(String[] args) {
        System.out.println("Pham Duc Dung 20215265");
        //TODO Auto-generated method stub
        //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: "Aladdin", category: "Animation", cost: 18.99F);
anOrder.addDigitalVideoDisc(dvd3);
        // print total cost
        System.out.println("Total Cost is: "+ anOrder.totalCost());
        anOrder.removeDigitalVideoDisc(dvd2);
        System.out.println("Total Cost is: ");
        System.out.println(anOrder.totalCost());
    }
}
```

Figure 4: Code class Aims

```
C:\Java\bin\java.exe "-javaagent:C:\Program Files\JetBrain
Pham Duc Dung 20215265
The disc has been added
The disc has been added
The disc has been added
Total Cost is: 63.89
The disc has been removed.
Total Cost is:
38.94

Process finished with exit code 0
```

Figure 5: Kết quả