

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: Lê Đồng Cảnh Phú
MSSV: 20225755
Lớp: VN01-K67
GVHD: Lê Thị Hoa
HTGD: Đặng Mạnh Cường

Hà Nội 10/2024

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

Mục tiêu là phát triển một hệ thống quản lý cửa hàng trực tuyến dành cho các sản phẩm truyền thông số, với trọng tâm hiện tại là đĩa DVD. Hệ thống này, AIMS (An Internet Media Store), sẽ hỗ trợ khách hàng tìm kiếm, duyệt qua danh sách DVD, xem thông tin chi tiết, phát thử, thêm sản phẩm vào giỏ hàng và thực hiện mua hàng. Đồng thời, hệ thống cũng phải cho phép quản lý cửa hàng xem và xử lý các đơn hàng, quản lý danh sách sản phẩm và cập nhật kho hàng.

2. Yêu cầu hệ thống

2.1. Đối với Customer:

➤ Duyệt và tìm kiếm DVD:

- Xem danh sách DVD sắp xếp theo thứ tự từ mới đến cũ.
- Tìm kiếm DVD theo ba cách:
 - **Theo tiêu đề:** So khớp bất kỳ từ nào trong chuỗi từ khóa với tiêu đề của DVD, không phân biệt chữ hoa chữ thường.
 - **Theo danh mục:** So khớp danh mục DVD, không phân biệt chữ hoa chữ thường.
 - **Theo giá:** Cung cấp khoảng giá tối thiểu và tối đa, hoặc chỉ giá tối đa.

➤ Xem thông tin và thêm vào giỏ hàng:

- Xem chi tiết từng DVD từ danh sách DVD.
- Thêm DVD vào giỏ hàng từ màn hình chi tiết hoặc từ danh sách.

➤ Quản lý giỏ hàng:

- Xem danh sách DVD trong giỏ hàng kèm tổng giá trị.
- Nghe thử một phần của DVD trước khi xác nhận đặt hàng.
- Sắp xếp DVD trong giỏ hàng:
 - **Theo tiêu đề:** Sắp xếp theo thứ tự bảng chữ cái của tiêu đề. Nếu trùng tiêu đề, ưu tiên DVD có giá cao hơn.
 - **Theo giá:** Sắp xếp giảm dần theo giá. Nếu trùng giá, sắp xếp tăng dần theo tiêu đề.
- Cập nhật giỏ hàng:
 - Cập nhật số lượng DVD hoặc xóa DVD khỏi giỏ hàng.
 - Một DVD ngẫu nhiên trong giỏ hàng sẽ được miễn phí để khuyến khích khách hàng.
- Lọc DVD trong giỏ hàng dựa trên ID hoặc tiêu đề. Nếu tìm thấy, hiển thị thông tin DVD; nếu không, thông báo không tìm thấy.

➤ Đặt hàng:

- Yêu cầu thông tin giao hàng và hướng dẫn giao hàng.
- Hệ thống tính phí giao hàng dựa trên tổng khối lượng đơn hàng và vị trí giao hàng.
- Hiển thị hóa đơn gồm danh sách DVD, tổng chi phí trước và sau thuế VAT, phí giao hàng.
- Thanh toán: Hiện chỉ chấp nhận thanh toán qua thẻ tín dụng và kết nối với hệ thống xác thực thẻ.
- Sau giao dịch, hiển thị thông tin chi tiết giao dịch và gửi thông tin đơn hàng qua email.

2.2. Đối với Store Manager:

➤ **Quản lý đơn hàng:**

- Đăng nhập vào hệ thống để vào chế độ quản lý.
- Xem danh sách các đơn hàng đang chờ xử lý.
- Xem chi tiết và duyệt hoặc từ chối từng đơn hàng.

➤ **Quản lý DVD:**

- Thêm DVD mới vào cửa hàng với đầy đủ thông tin gồm ID, tiêu đề, danh mục, đạo diễn, thời lượng và giá.
- Xóa DVD khỏi cửa hàng khi cần thiết.

3. Use Case Diagram

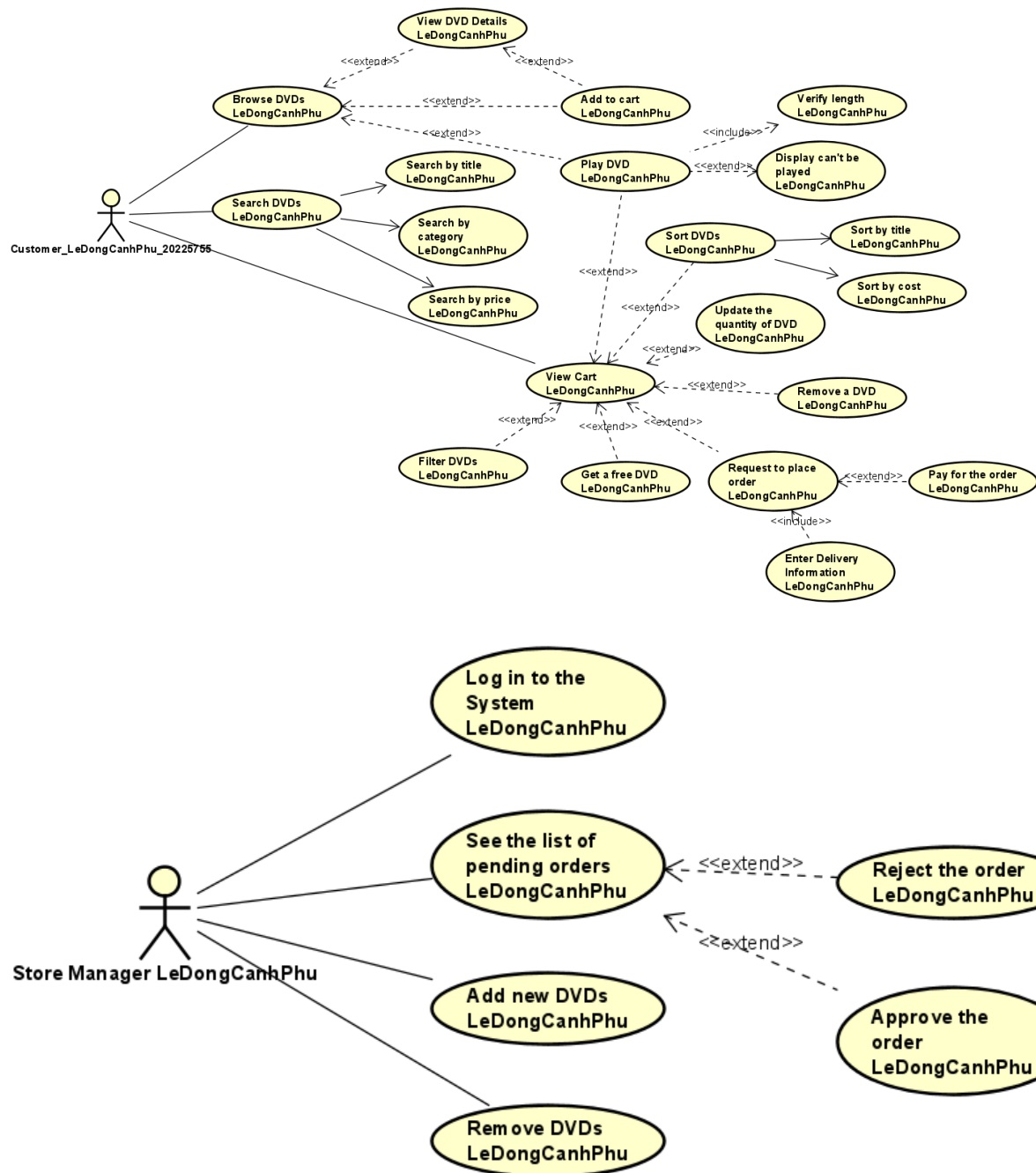


Figure 1: Use Case Diagram

4. Class Diagram

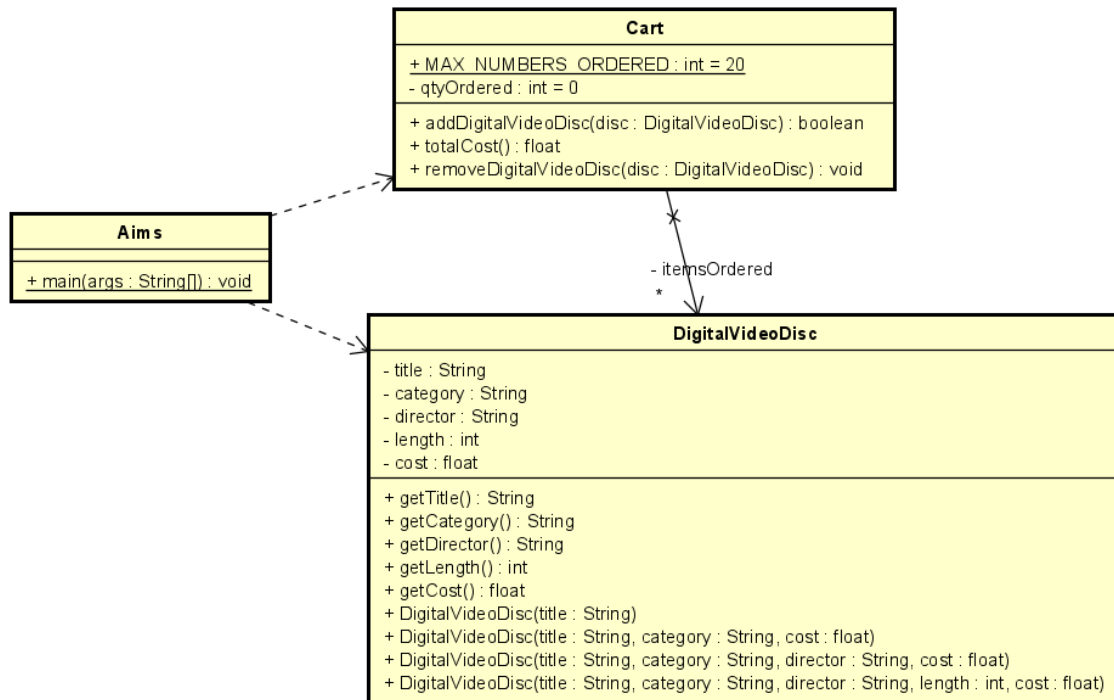
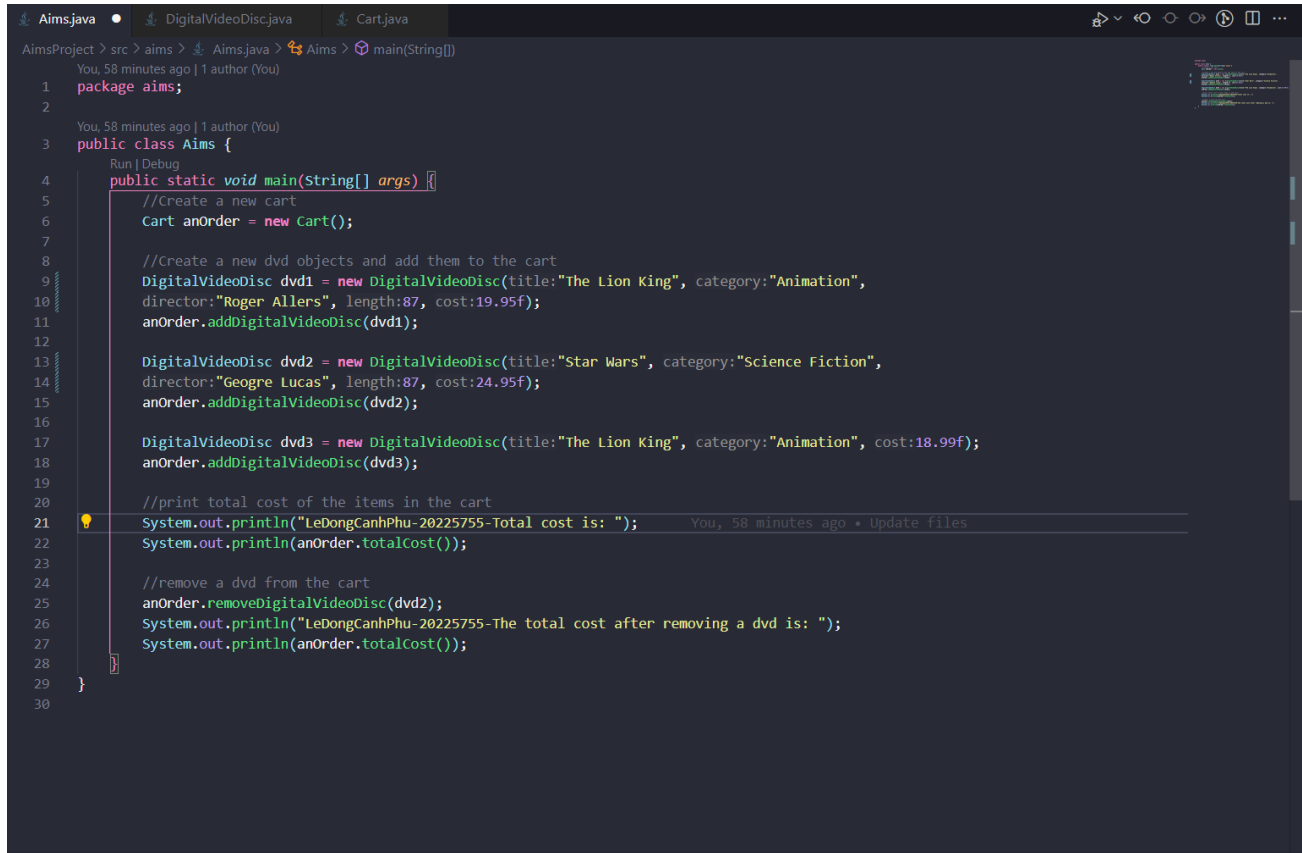


Figure 2: Class Diagram

5. Source Code

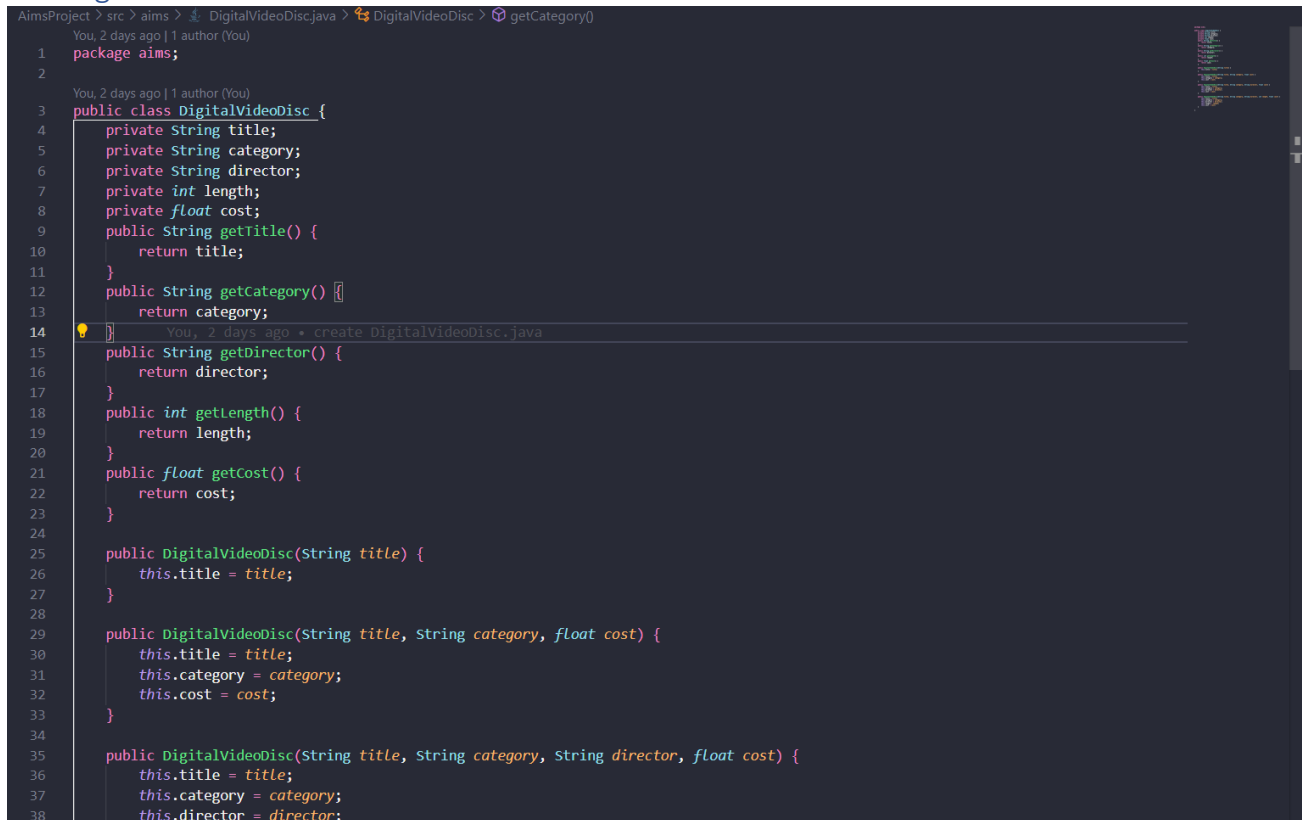
5.1. Aims Class



```
Aims.java • DigitalVideoDisc.java • Cart.java
AimsProject > src > aims > Aims.java > Aims > main(String[])
You, 58 minutes ago | 1 author (You)
1 package aims;
2
3 public class Aims {
4     Run | Debug
5     public static void main(String[] args) {
6         //Create a new cart
7         Cart anOrder = new Cart();
8
9         //Create a new dvd objects and add them to the cart
10        DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion King", category:"Animation",
11        director:"Roger Allers", length:87, cost:19.95f);
12        anOrder.addDigitalVideoDisc(dvd1);
13
14        DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star Wars", category:"Science Fiction",
15        director:"Geogre Lucas", length:87, cost:24.95f);
16        anOrder.addDigitalVideoDisc(dvd2);
17
18        DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"The Lion King", category:"Animation", cost:18.99f);
19        anOrder.addDigitalVideoDisc(dvd3);
20
21        //print total cost of the items in the cart
22        System.out.println("LeDongCanhPhu-20225755-Total cost is: ");
23        System.out.println(anOrder.totalCost());
24
25        //remove a dvd from the cart
26        anOrder.removeDigitalVideoDisc(dvd2);
27        System.out.println("LeDongCanhPhu-20225755-The total cost after removing a dvd is: ");
28        System.out.println(anOrder.totalCost());
29    }
30 }
```

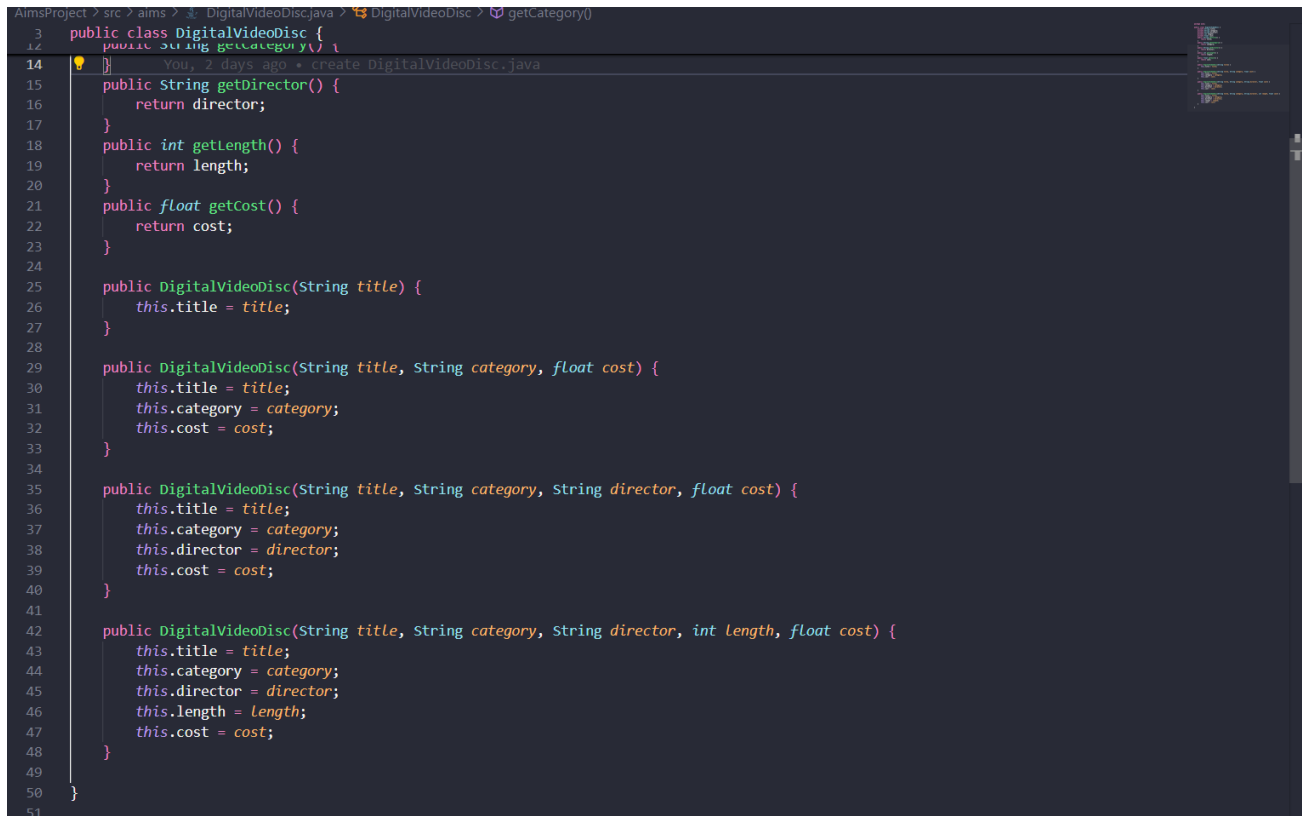
Figure 3: Aims Class

5.2. DigitalVideoDisc Class



```
AimsProject > src > aims > DigitalVideoDisc.java > DigitalVideoDisc > getCategory()
You, 2 days ago | 1 author (You)
1 package aims;
2
3 You, 2 days ago | 1 author (You)
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    public String getTitle() {
11        return title;
12    }
13    public String getCategory() {
14        return category;
15    }
16    public String getDirector() {
17        return director;
18    }
19    public int getLength() {
20        return length;
21    }
22    public float getCost() {
23        return cost;
24    }
25    public DigitalVideoDisc(String title) {
26        this.title = title;
27    }
28    public DigitalVideoDisc(String title, String category, float cost) {
29        this.title = title;
30        this.category = category;
31        this.cost = cost;
32    }
33    public DigitalVideoDisc(String title, String category, String director, float cost) {
34        this.title = title;
35        this.category = category;
36        this.director = director;
37    }
38 }
```

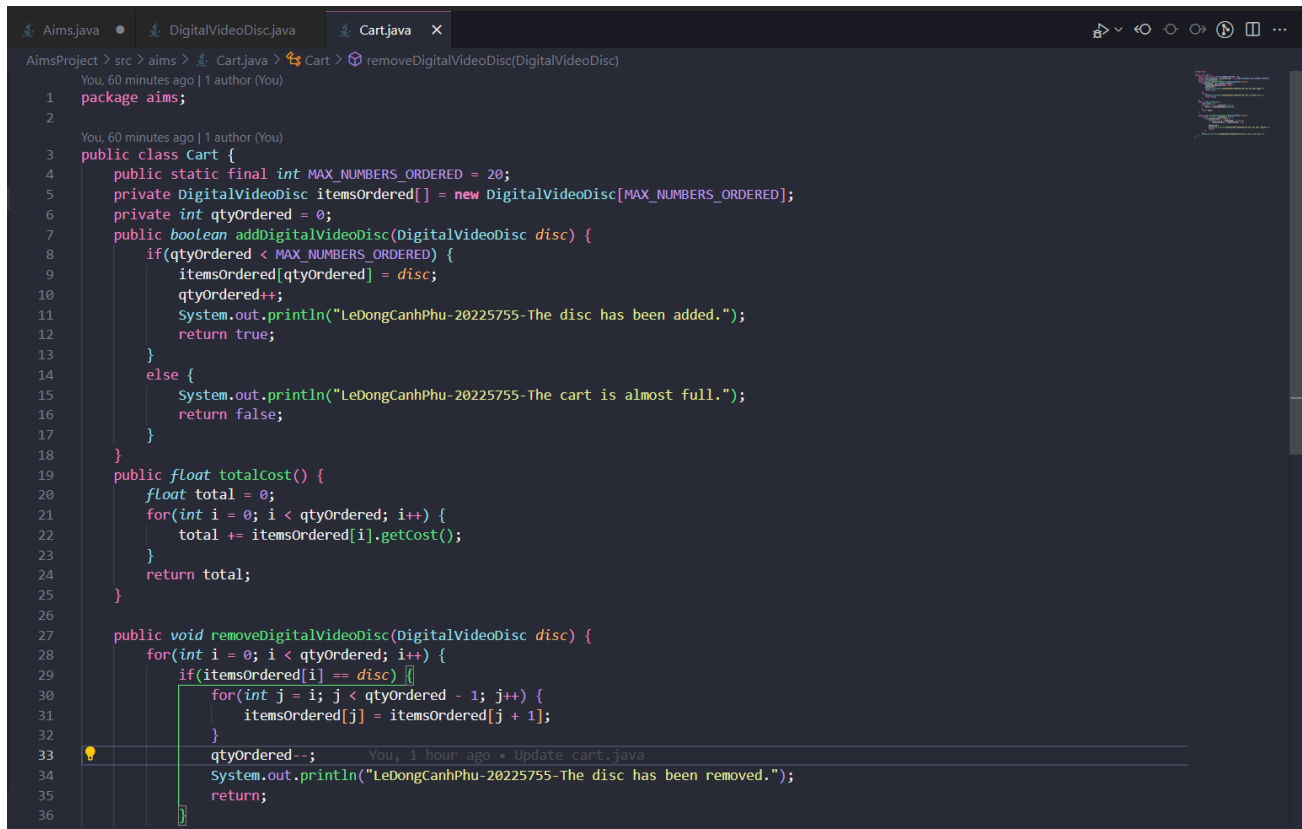
Figure 4: DigitalVideoDisc Class 1

A screenshot of an IDE window showing the code for the DigitalVideoDisc class. The code is written in Java and includes several methods for getting and setting attributes like title, category, director, length, and cost. The IDE has a dark theme and a sidebar on the right showing a file explorer.

```
3 public class DigitalVideoDisc {
4     public String getCategory() {
5     }
6 }
7
8 You, 2 days ago • create DigitalVideoDisc.java
9
10 public String getDirector() {
11     return director;
12 }
13
14 public int getLength() {
15     return length;
16 }
17
18 public float getCost() {
19     return cost;
20 }
21
22 public DigitalVideoDisc(String title) {
23     this.title = title;
24 }
25
26 public DigitalVideoDisc(String title, String category, float cost) {
27     this.title = title;
28     this.category = category;
29     this.cost = cost;
30 }
31
32 public DigitalVideoDisc(String title, String category, String director, float cost) {
33     this.title = title;
34     this.category = category;
35     this.director = director;
36     this.cost = cost;
37 }
38
39 public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
40     this.title = title;
41     this.category = category;
42     this.director = director;
43     this.length = length;
44     this.cost = cost;
45 }
46
47 }
```

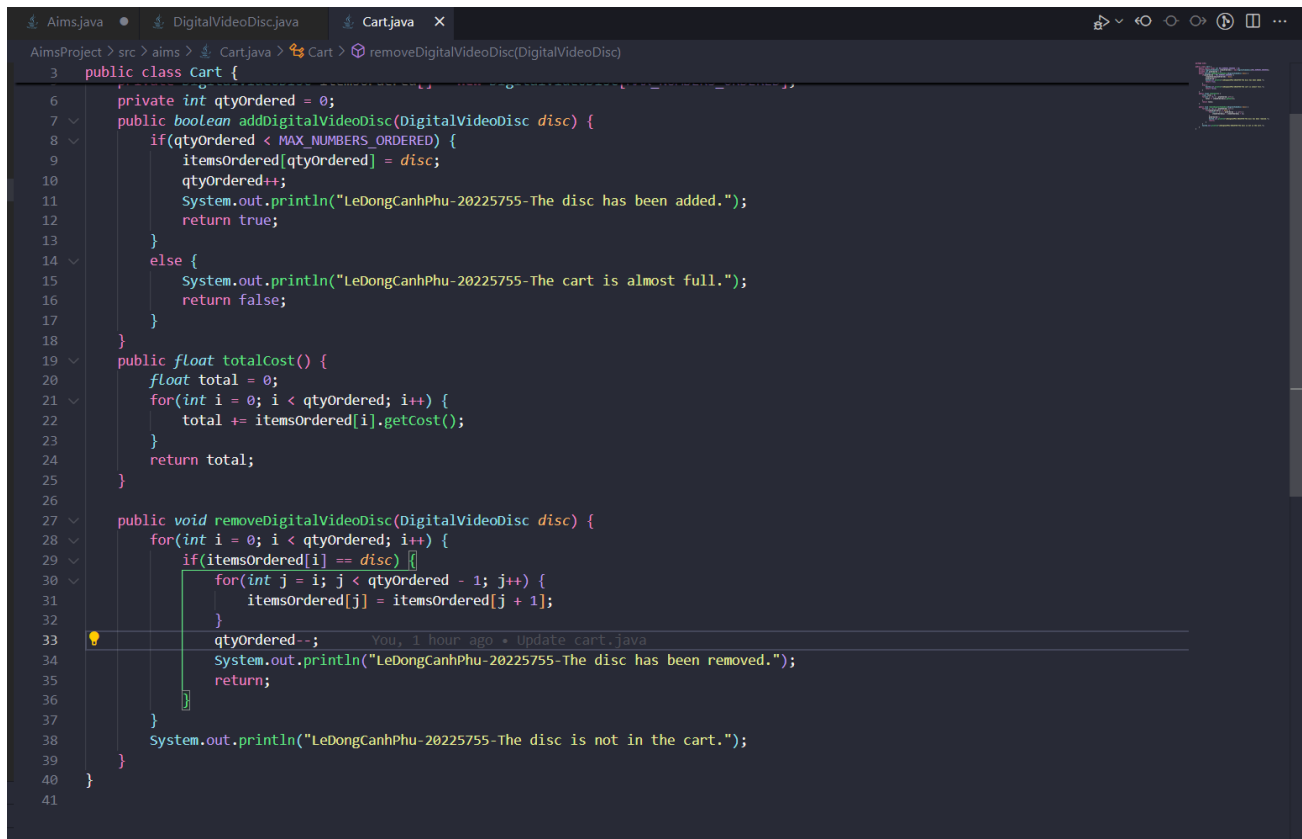
Figure 5: DigitalVideoDisc Class 2

5.3. Cart Class



```
1 package aims;
2
3 public class Cart {
4     public static final int MAX_NUMBERS_ORDERED = 20;
5     private DigitalVideoDisc itemsOrdered[] = new DigitalVideoDisc[MAX_NUMBERS_ORDERED];
6     private int qtyOrdered = 0;
7     public boolean addDigitalVideoDisc(DigitalVideoDisc disc) {
8         if(qtyOrdered < MAX_NUMBERS_ORDERED) {
9             itemsOrdered[qtyOrdered] = disc;
10            qtyOrdered++;
11            System.out.println("LeDongCanhPhu-20225755-The disc has been added.");
12            return true;
13        }
14        else {
15            System.out.println("LeDongCanhPhu-20225755-The cart is almost full.");
16            return false;
17        }
18    }
19    public float totalCost() {
20        float total = 0;
21        for(int i = 0; i < qtyOrdered; i++) {
22            total += itemsOrdered[i].getCost();
23        }
24        return total;
25    }
26
27    public void removeDigitalVideoDisc(DigitalVideoDisc disc) {
28        for(int i = 0; i < qtyOrdered; i++) {
29            if(itemsOrdered[i] == disc) {
30                for(int j = i; j < qtyOrdered - 1; j++) {
31                    itemsOrdered[j] = itemsOrdered[j + 1];
32                }
33                qtyOrdered--;
34                System.out.println("LeDongCanhPhu-20225755-The disc has been removed.");
35                return;
36            }
37        }
38    }
39 }
```

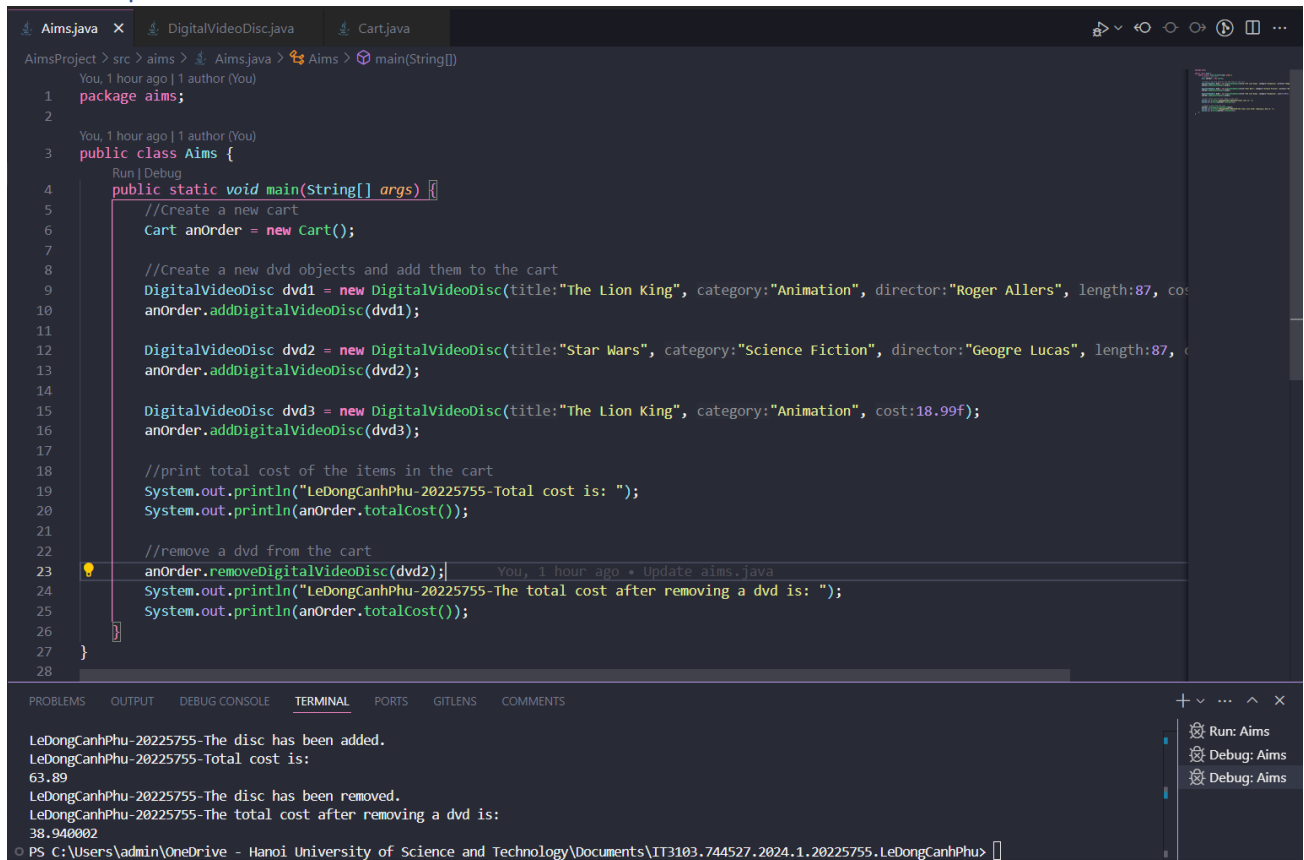
Figure 6: Cart Class 1



```
3 public class Cart {
4     private int qtyOrdered = 0;
5     public boolean addDigitalVideoDisc(DigitalVideoDisc disc) {
6         if (qtyOrdered < MAX_NUMBERS_ORDERED) {
7             itemsOrdered[qtyOrdered] = disc;
8             qtyOrdered++;
9             System.out.println("LeDongCanhPhu-20225755-The disc has been added.");
10            return true;
11        } else {
12            System.out.println("LeDongCanhPhu-20225755-The cart is almost full.");
13            return false;
14        }
15    }
16    public float totalCost() {
17        float total = 0;
18        for (int i = 0; i < qtyOrdered; i++) {
19            total += itemsOrdered[i].getCost();
20        }
21        return total;
22    }
23    public void removeDigitalVideoDisc(DigitalVideoDisc disc) {
24        for (int i = 0; i < qtyOrdered; i++) {
25            if (itemsOrdered[i] == disc) {
26                for (int j = i; j < qtyOrdered - 1; j++) {
27                    itemsOrdered[j] = itemsOrdered[j + 1];
28                }
29                qtyOrdered--;
30                System.out.println("LeDongCanhPhu-20225755-The disc has been removed.");
31                return;
32            }
33        }
34        System.out.println("LeDongCanhPhu-20225755-The disc is not in the cart.");
35    }
36 }
```

Figure 7: Cart Class 2

6. Kết quả demo



The screenshot displays an IDE with three tabs: `Aims.java`, `DigitalVideoDisc.java`, and `Cart.java`. The `Aims.java` tab is active, showing the following code:

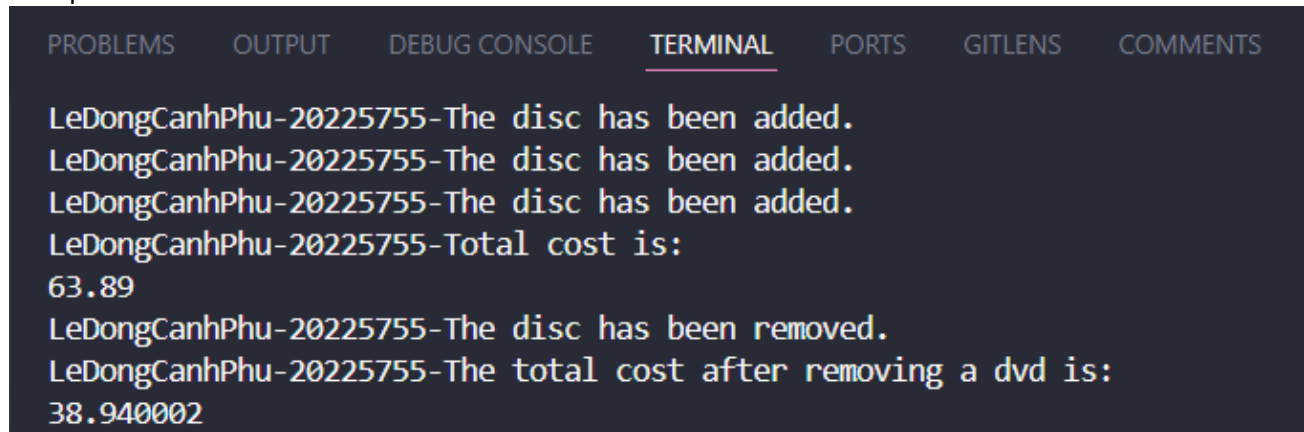
```
1 package aims;
2
3 public class Aims {
4     //Create a new cart
5     Cart anOrder = new Cart();
6
7     //Create a new dvd objects and add them to the cart
8     DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion King", category:"Animation", director:"Roger Allers", length:87, cost:18.99f);
9     anOrder.addDigitalVideoDisc(dvd1);
10
11     DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star Wars", category:"Science Fiction", director:"Geogre Lucas", length:87, cost:18.99f);
12     anOrder.addDigitalVideoDisc(dvd2);
13
14     DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"The Lion King", category:"Animation", cost:18.99f);
15     anOrder.addDigitalVideoDisc(dvd3);
16
17     //print total cost of the items in the cart
18     System.out.println("LeDongCanhPhu-20225755-Total cost is: ");
19     System.out.println(anOrder.totalCost());
20
21     //remove a dvd from the cart
22     anOrder.removeDigitalVideoDisc(dvd2);
23     System.out.println("LeDongCanhPhu-20225755-The total cost after removing a dvd is: ");
24     System.out.println(anOrder.totalCost());
25 }
26
27
28
```

The terminal output at the bottom shows the execution results:

```
LeDongCanhPhu-20225755-The disc has been added.
LeDongCanhPhu-20225755-Total cost is:
63.89
LeDongCanhPhu-20225755-The disc has been removed.
LeDongCanhPhu-20225755-The total cost after removing a dvd is:
38.940002
PS C:\Users\admin\OneDrive - Hanoi University of Science and Technology\Documents\IT3103.744527.2024.1.20225755.LeDongCanhPhu>
```

Figure 8: Code Demo

Kết quả:

A screenshot of a terminal window with a dark background. At the top, there is a horizontal menu bar with several tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is highlighted with a red underline), PORTS, GITLENS, and COMMENTS. Below the menu bar, the terminal displays the following text in a monospaced font: LeDongCanhPhu-20225755-The disc has been added. (repeated three times), LeDongCanhPhu-20225755-Total cost is: 63.89, LeDongCanhPhu-20225755-The disc has been removed., and LeDongCanhPhu-20225755-The total cost after removing a dvd is: 38.940002.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  COMMENTS

LeDongCanhPhu-20225755-The disc has been added.
LeDongCanhPhu-20225755-The disc has been added.
LeDongCanhPhu-20225755-The disc has been added.
LeDongCanhPhu-20225755-Total cost is:
63.89
LeDongCanhPhu-20225755-The disc has been removed.
LeDongCanhPhu-20225755-The total cost after removing a dvd is:
38.940002
```

Figure 9: Result

7. Reading Assignment

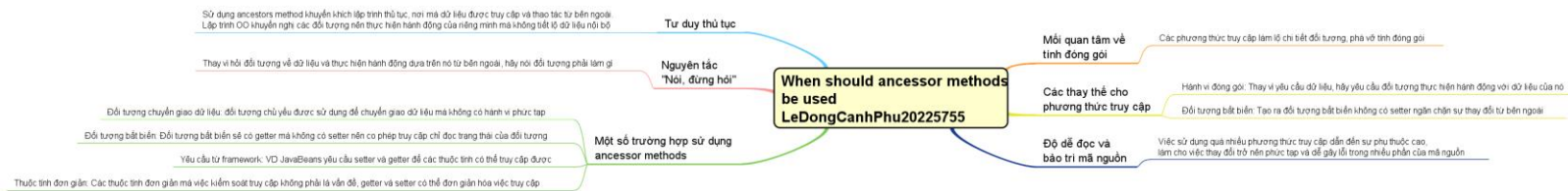


Figure 10: Getter and Setter Methods

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?

Có, Java cho phép bạn tạo nhiều constructor cho một lớp, tính năng này được gọi là constructor overloading. Điều này có nghĩa là bạn có thể định nghĩa các constructor khác nhau trong cùng một lớp miễn là danh sách tham số của chúng là duy nhất.