

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



BÁO CÁO THỰC HÀNH
IT3103-744529-2024.1

BÀI THỰC HÀNH 2

Họ và tên sinh viên: Đoàn Thanh Tùng

MSSV: 20225946

Lớp: Việt Nhật 02 – K67

GVHD: Lê Thị Hoa

HTGD: Bùi Trọng Dũng

Hà Nội 10/2024

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

2. Yêu cầu hệ thống

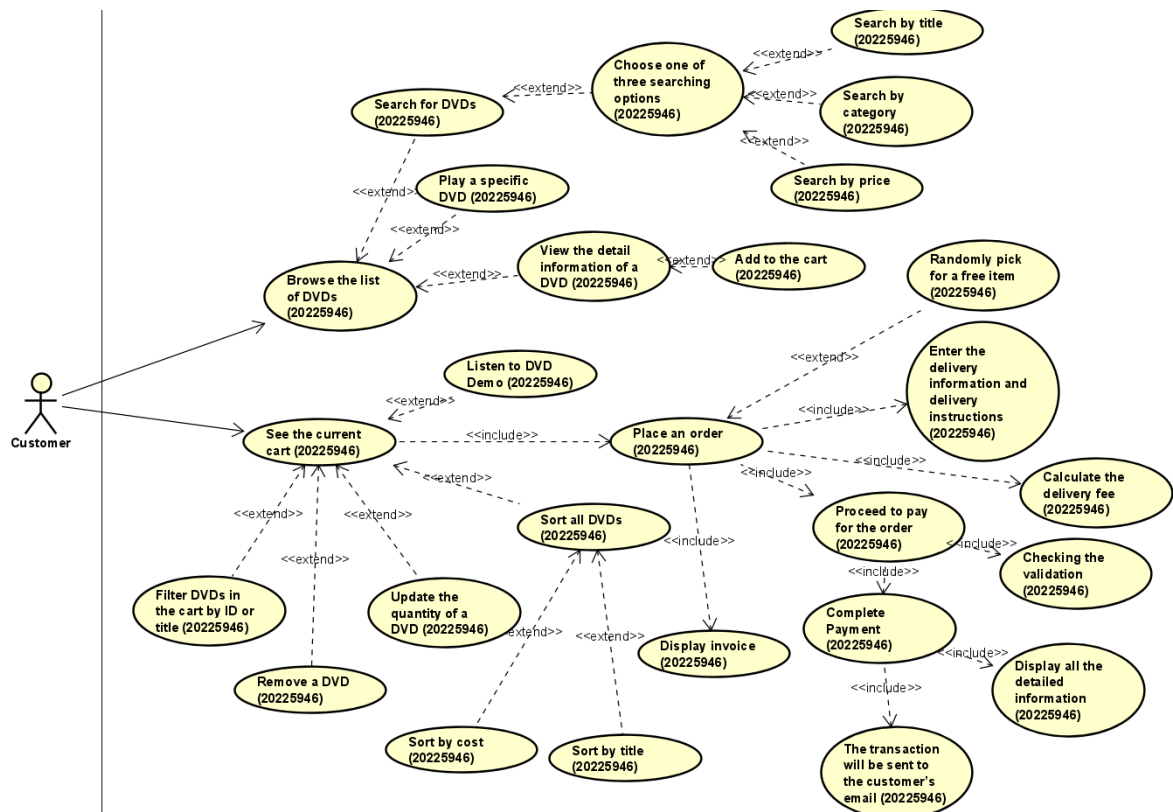
2.1. Đối với Customer:

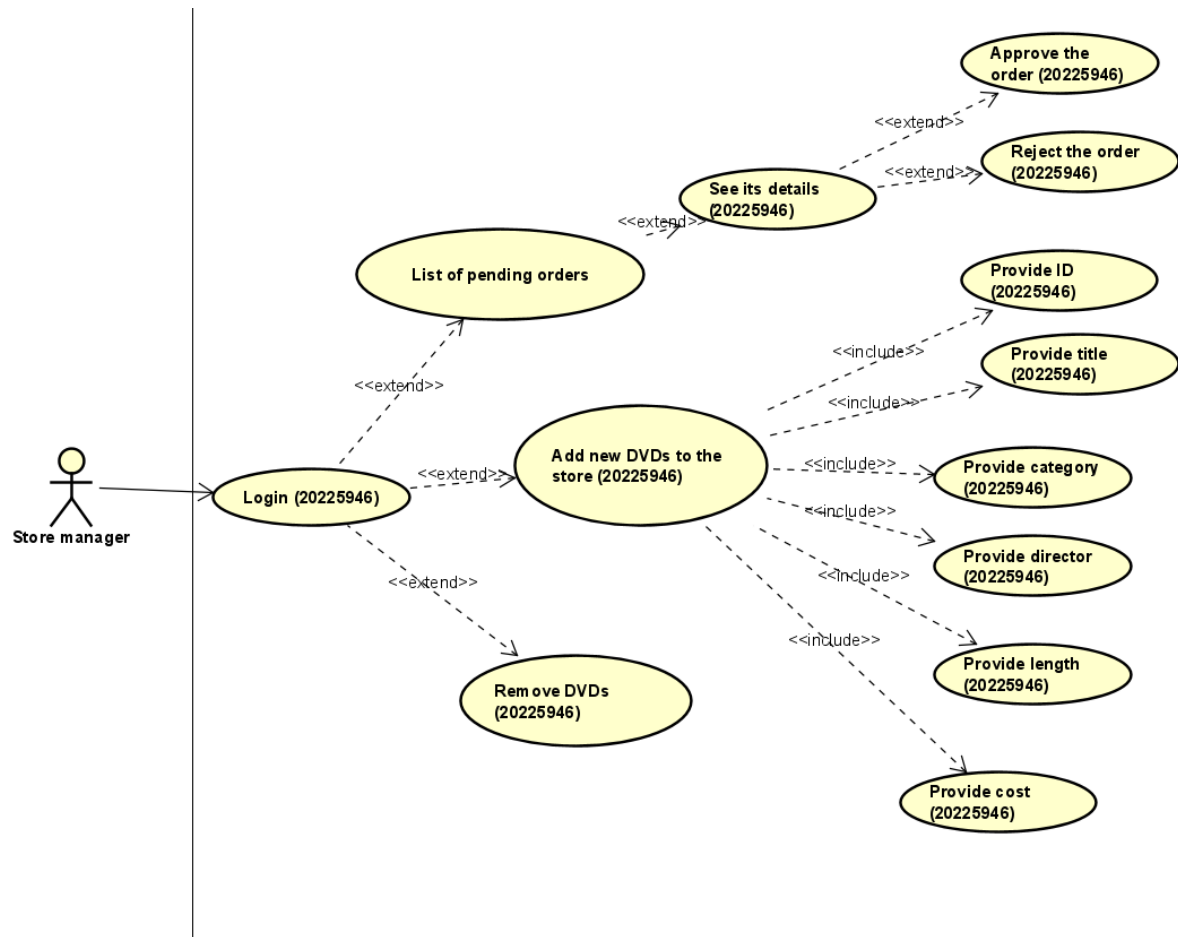
- Duyệt danh sách các DVD có sẵn trong cửa hàng.
- Tìm kiếm DVDs theo tiêu đề, thể loại và giá cả.
- Xem thông tin chi tiết của DVD để quyết định thêm vào giỏ hàng.
- Xem giỏ hàng.
- Xem bản demo của DVDs.
- Đặt DVD.
- Sắp xếp DVDs theo tiêu đề hoặc giá cả.
- Điều chỉnh số lượng DVDs muốn mua.
- Bỏ DVDs ra khỏi giỏ hàng.
- Đặt hàng

2.2. Đối với Store Manager:

- Đăng nhập.
- Xem danh sách các đơn hàng đang chờ xử lí.
- Xem chi tiết các đơn đặt DVDs rồi có thể chọn chấp nhận hoặc từ chối đơn hàng
- Thêm DVDs mới vào cửa hàng.
- Bỏ DVDs ra khỏi cửa hàng.

3. Use Case Diagram



*Figure 1: Use Case Diagram*

4. Class Diagram

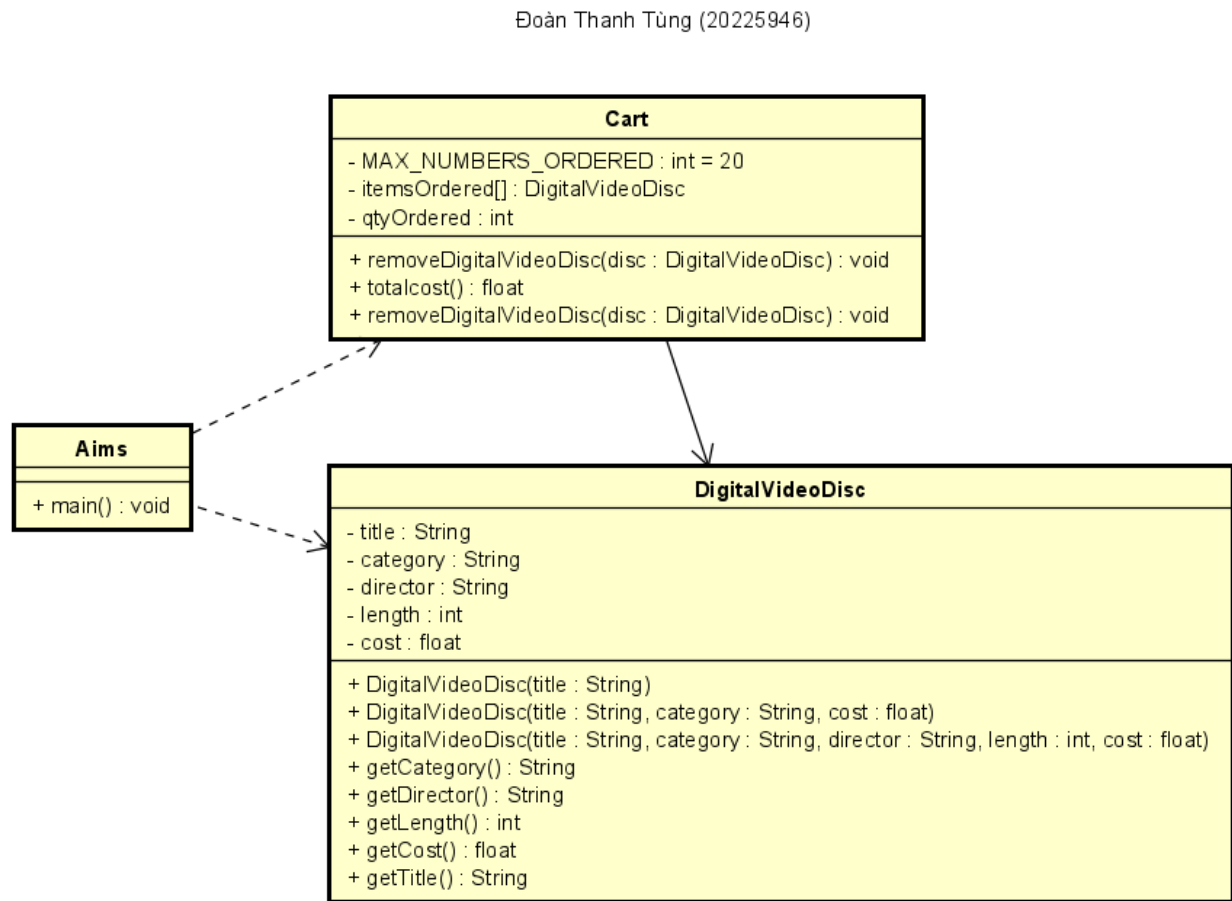


Figure 2: Class Diagram

5. Source Code

5.1. Aims Class

```
1 package lab2;  
2  
3 // Doan Thanh Tung - 20225946  
4 public class Aims {  
5     public static void main(String args[]) {  
6         Cart order = new Cart();  
7     }  
8 }  
9 |
```

Figure 3: Aims Class

5.2. DigitalVideoDisc Class

```
1 package lab2;
2
3 // Doan Thanh Tung - 20225946
4 public class DigitalVideoDisc {
5     private String title;
6     private String category;
7     private String director;
8     private float cost;
9     private int length;
10
11     public DigitalVideoDisc(String title) {
12         super();
13         this.title = title;
14     }
15
16     public DigitalVideoDisc(String title, String category, float cost) {
17         super();
18         this.title = title;
19         this.category = category;
20         this.cost = cost;
21     }
22
23     public DigitalVideoDisc(String title, String category, String director, float cost) {
24         super();
25         this.title = title;
26         this.category = category;
27         this.director = director;
28         this.cost = cost;
29     }
30 }
```

Figure 4: DigitalVideoDisc Class 1


```
30
31• public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
32    super();
33    this.title = title;
34    this.category = category;
35    this.director = director;
36    this.length = length;
37    this.cost = cost;
38 }
39
40• public String getCategory() {
41    return category;
42 }
43
44• public String getDirector() {
45    return director;
46 }
47
48• public int getLength() {
49    return length;
50 }
51
52• public float getCost() {
53    return cost;
54 }
55
56• public String getTitle() {
57    return title;
58 }
59 }
60
```

Figure 5: DigitalVideoDisc Class 2

5.3. Cart Class

```
1 package lab2;
2
3 // Doan Thanh Tung - 20225946
4 public class Cart {
5     public static final int MAX_NUMBERS_ORDERED = 20;
6     private DigitalVideoDisc itemsOrdered[] = new DigitalVideoDisc[MAX_NUMBERS_ORDERED];
7     private int qtyOrdered;
8
9     public void addDigitalVideoDisc(DigitalVideoDisc disc) {
10         if (qtyOrdered <= MAX_NUMBERS_ORDERED) {
11             itemsOrdered[qtyOrdered++] = disc;
12             System.out.println(disc.getTitle() + " is sucessfully added");
13             return;
14         }
15         System.out.println("the cart is full");
16     }
17
18     public void removeDigitalVideoDisc(DigitalVideoDisc disc) {
19         if (itemsOrdered.length == 0) {
20             System.out.println("nothing in the cart!");
21         }
22         for (int i = 0; i < qtyOrdered; i++) {
23             if (itemsOrdered[i].equals(disc)) {
24                 for (int j = i; j < qtyOrdered; j++) {
25                     itemsOrdered[j] = itemsOrdered[j + 1];
26                 }
27                 qtyOrdered--;
28                 System.out.println(disc.getTitle() + " is sucessfully removed");
29                 return;
30             }
31         }
32     }
33 }
```

Figure 6: Cart Class 1

```
18• public void removeDigitalVideoDisc(DigitalVideoDisc disc) {
19    if (itemsOrdered.length == 0) {
20        System.out.println("nothing in the cart!");
21    }
22    for (int i = 0; i < qtyOrdered; i++) {
23        if (itemsOrdered[i].equals(disc)) {
24            for (int j = i; j < qtyOrdered; j++) {
25                itemsOrdered[j] = itemsOrdered[j + 1];
26            }
27            qtyOrdered--;
28            System.out.println(disc.getTitle() + " is sucessfully removed");
29            return;
30        }
31    }
32    System.out.println("there is no DVD like that in your cart!");
33 }
34
35• public float totalcost() {
36    float sum = 0;
37    for (int i = 0; i < qtyOrdered; i++) {
38        sum += itemsOrdered[i].getCost();
39    }
40    return sum;
41 }
42
43• public void print() {
44    for (int i = 0; i < qtyOrdered; i++) {
45        System.out.println(itemsOrdered[i].getTitle());
46    }
47 }
48 }
49
```

Figure 7: Cart Class 2

6. Kết quả demo

Thêm 4 phim vào giỏ hàng.

Xóa dvd2

Xóa dvd2 lần nữa để kiểm tra thông báo lỗi

In ra giá tiền tổng

```
1 package lab2;
2
3 // Doan Thanh Tung - 20225946
4 public class Aims {
5     public static void main(String args[]) {
6
7         Cart order = new Cart();
8
9         DigitalVideoDisc dvd1 = new DigitalVideoDisc("Doraemon the movie", "Anime", "Zen Soichiro", 90, 25.50f);
10        order.addDigitalVideoDisc(dvd1);
11        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Spiderman no way home", "Science Fiction", "Jon Watts", 90,
12            30.15f);
13        order.addDigitalVideoDisc(dvd2);
14
15        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Naruto", "Anime", 21.30f);
16        DigitalVideoDisc dvd4 = new DigitalVideoDisc("Diary of a Wimpy Kid", "Comedy", "Thor Freudenthal", 40.50f);
17        order.addDigitalVideoDisc(dvd3);
18        order.addDigitalVideoDisc(dvd4);
19        order.removeDigitalVideoDisc(dvd2);
20        order.removeDigitalVideoDisc(dvd2);
21        System.out.printf("Total cost is: %.2f\n", order.totalcost());
22    }
23 }
24
```

Figure 8: Code Demo

Kết quả:

```
Doraemon the movie is sucessfully added  
Spiderman no way home is sucessfully added  
Naruto is sucessfully added  
Diary of a Wimpy Kid is sucessfully added  
Spiderman no way home is sucessfully removed  
there is no DVD like that in your cart!  
Total cost is: 87.30
```

Figure 9: Result

7. Reading Assignment

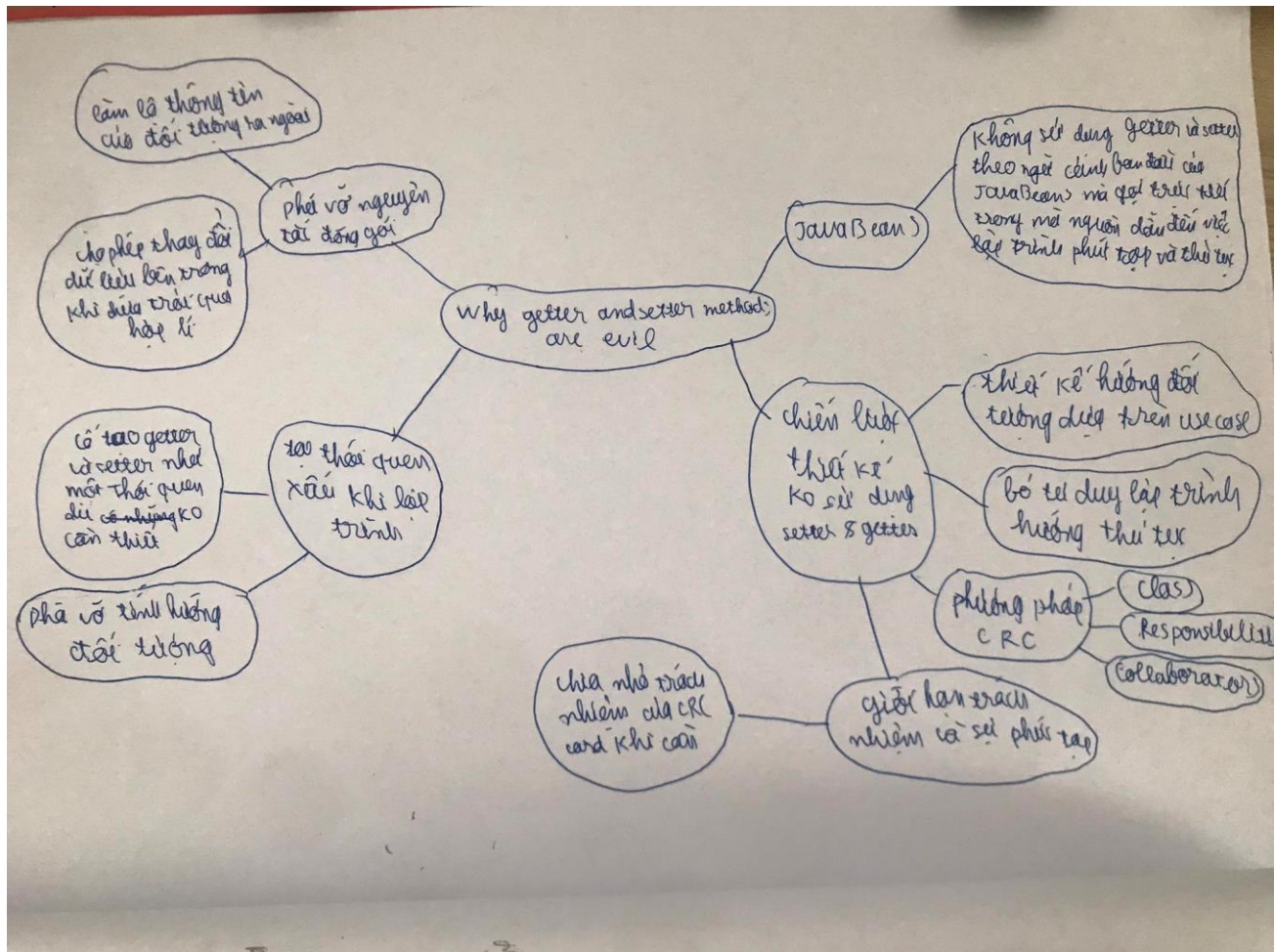


Figure 10: Getter and Setter Methods

8. Answer the question

Why getter and setter methods are evil

- Phá vỡ nguyên tắc đóng gói:
 - + Làm lộ thông tin của đối tượng ra ngoài.
 - + Cho phép thay đổi dữ liệu bên trong khi chưa trải qua xử lý hợp lý.
- Tạo thói quen xấu khi lập trình:
 - + Cố tạo getter và setter như một thói quen xấu khiến cho thông tin vẫn bị truy cập và làm phức tạp thêm vấn đề.
 - + Phá vỡ tính hướng đối tượng.
- Không đúng với mục đích ban đầu khi sử dụng getter và setter trong javaBeans,

gọi trực tiếp trong mã nguồn dẫn đến quá trình lập trình phức tạp và thủ tục.

- Có các giải pháp, chiến lược thiết kế khác không phụ thuộc vào setter và getter:
 - + Thiết kế hướng đối tượng dựa trên use case.
 - + Bỏ tư duy lập trình hướng thủ tục.
 - + Phương pháp CRC gồm class, responsibilities, collaborators.
 - + Giới hạn trách nhiệm và sự phức tạp bằng việc chia nhỏ trách nhiệm của CRC card khi cần.