

## BÁO CÁO THỰC HÀNH LAB 2 LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

### Mục lục nội dung

1. Đề bài.....	2
2. Yêu cầu bài toán.....	2
3. Use case diagram .....	2
4. Class diagram.....	3
5. Mã nguồn của chương trình.....	4
Aims class:.....	4
Cart class:.....	5
DigitalVideoDisc class:.....	9
6. Demo chương trình.....	11
7. Reading Assignment.....	11
8. Answer the questions.....	11

### Mục lục hình ảnh

Figure 1: Use case diagram.....	2
Figure 2: Class diagram .....	3
Figure 3: Aims class code.....	4
Figure 4: Aims class code.....	4
Figure 5: Cart class code.....	5
Figure 6: Cart class code.....	6
Figure 7: Cart class code.....	7
Figure 8: Cart class code.....	8
Figure 9: Cart class code.....	8
Figure 10: DVD class code .....	9
Figure 11: DVD class code .....	10
Figure 12: DVD class code .....	10
Figure 13: Result.....	11
Figure 14: Setter and Getter method .....	11

## 1. Đề bài

Thiết kế hệ thống mới cho dự án AIMS. (Hiện tại chỉ có 1 loại phương tiện DVD)

## 2. Yêu cầu bài toán

Đối với người mua hàng:

- Duyệt danh sách các DVD có sẵn trong cửa hàng.
- Tìm kiếm DVD theo: tiêu đề, danh mục, giá cả.
- Xem thông tin chi tiết của 1 DVD.
- Thêm DVD vào giỏ hàng.
- Xem giỏ hàng.
- Sắp xếp DVD trong giỏ hàng theo tiêu đề hoặc chi phí
- Cập nhật số lượng DVD trong giỏ hàng
- 4- Đặt hàng:

Đối với người quản lý cửa hàng

- Đăng nhập, kiểm tra quyền.
- Xem danh sách các đơn hàng đang chờ xử lý.
- Xem chi tiết 1 đơn hàng và xử lý nó (đồng ý/ từ chối đặt hàng).
- Thêm(xóa) DVD mới vào(ra) danh sách sản phẩm.

## 3. Use case diagram

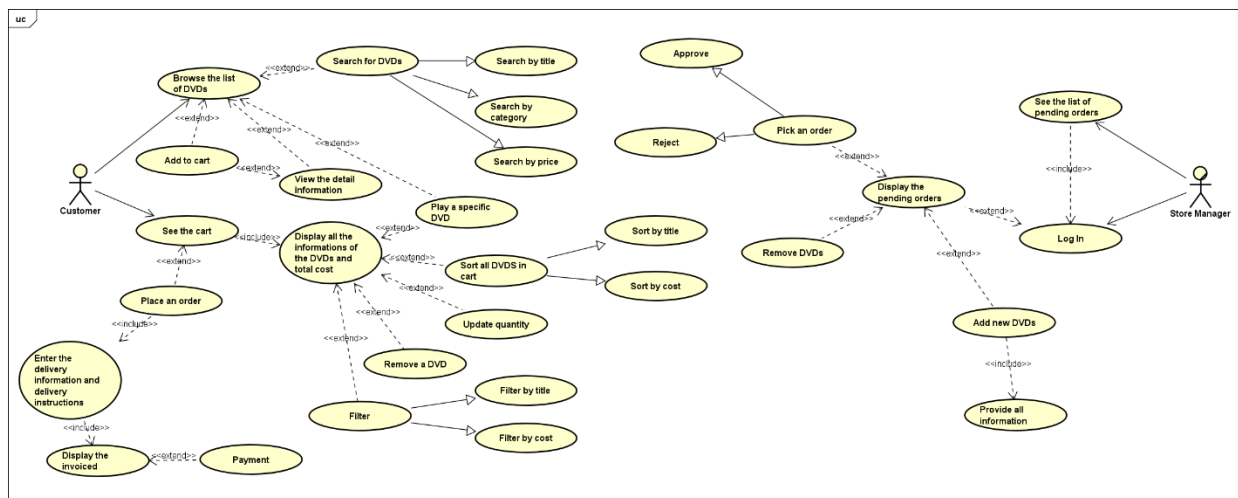


Figure 1: Use case diagram

## 4. Class diagram

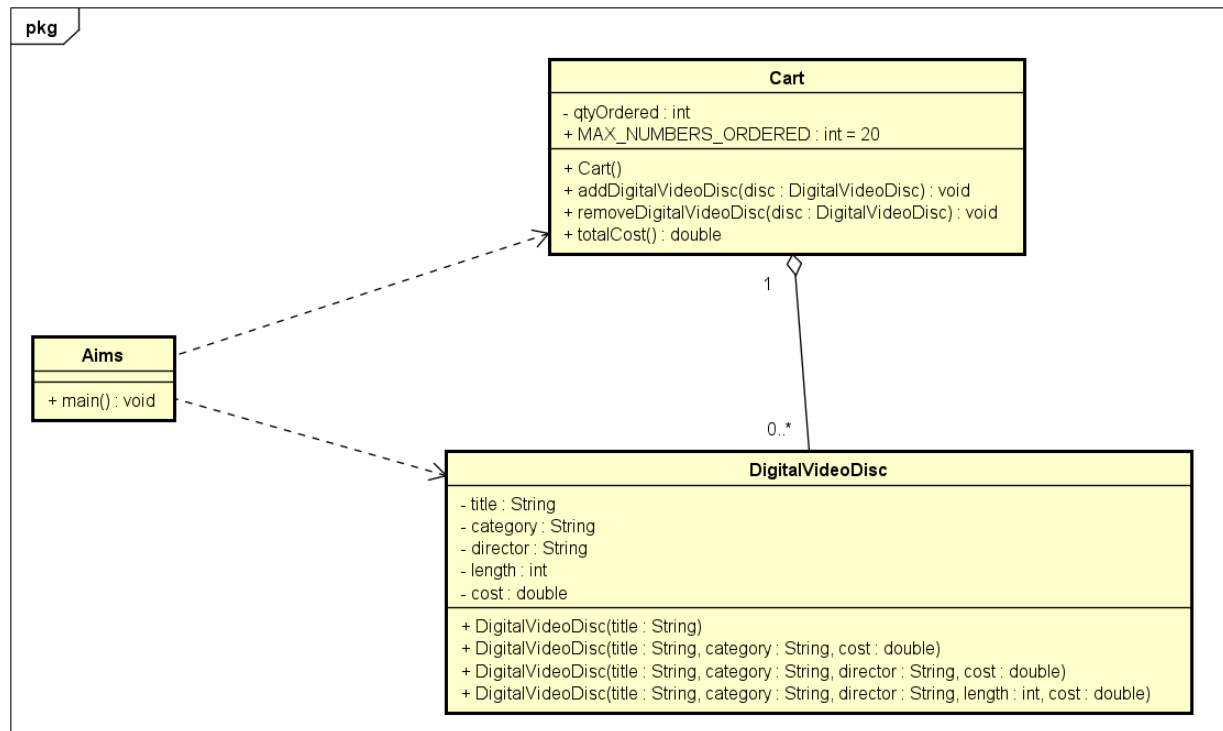


Figure 2: Class diagram

## 5. Mã nguồn của chương trình

Aims class:

```
1 package lab02.AimsProject;
2
3 public class Aims {
4     public static void main(String[] args) {
5         //Create a new cart
6         Cart anOrder = new Cart();
7
8         //Create new dvd objects and add them to the cart
9         DigitalVideoDisc dvd1 = new DigitalVideoDisc (title: "The Lion King",
10             category: "Animation", director: "Roger Allers", length: 87, cost: 19.95f);
11         anOrder.addDigitalVideoDisc (dvd1);
12
13         DigitalVideoDisc dvd2 = new DigitalVideoDisc (title: "Star Wars",
14             category: "Science Fiction", director: "George Lucas", length: 87, cost: 24.95f);
15         anOrder.addDigitalVideoDisc (dvd2);
16
17         DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladdin",
18             category: "Animation", cost: 18.99f);
19         anOrder.addDigitalVideoDisc (dvd3);
20
21         //print total cost of the items in the cart
22         System.out.println("Total Cost is: ");
23         System.out.println(anOrder.totalCost());
24
25         // Remove the dvd1
```

Figure 3: Aims class code

```
26         anOrder.removeDigitalVideoDisc(dvd1);
27
28         //print total cost of the items in the cart after remove
29         System.out.println("Total Cost is: ");
30         System.out.println(anOrder.totalCost());
31     }
32 }
33
```

Figure 4: Aims class code

Cart class:

```
1 package lab02.AimsProject;
2
3 public class Cart {
4     // Attribute
5     private DigitalVideoDisc []itemsOrdered =
6         new DigitalVideoDisc[MAX_NUMBERS_ORDERED];
7
8     private int qtyOrdered;
9
10    public static final int MAX_NUMBERS_ORDERED = 20;
11
12    // Constructor
13    public Cart() {
14        setQtyOrdered(0);
15    }
16
17    // Setter and Getter
18    public DigitalVideoDisc[] getItemsOrdered() {
19        return itemsOrdered;
20    }
```

Figure 5: Cart class code

```
21
    2 usages
    public void setItemsOrdered(DigitalVideoDisc[] itemsOrdered) {
23        this.itemsOrdered = itemsOrdered;
24    }
25
    7 usages
26    public int getQtyOrdered() { return qtyOrdered; }
29
    3 usages
30    public void setQtyOrdered(int qtyOrdered) { this.qtyOrdered = qtyOrdered; }
33
34    // Method to add a new DVD
    3 usages
35    public void addDigitalVideoDisc(DigitalVideoDisc disc) {
36        // If cart is full
37        if (getQtyOrdered() >= 20) {
38            System.out.println("The cart is almost full");
39            return;
40        }
41
42        // Increase the qtyOrdered
43        setQtyOrdered(getQtyOrdered() + 1);
44
45        // Add to cart
```

Figure 6: Cart class code

```
46     DigitalVideoDisc []newItemsOrdered = getItemsOrdered();
47     newItemsOrdered[getQtyOrdered() - 1] = disc;
48     setItemsOrdered(newItemsOrdered);
49
50     // Notify
51     System.out.println("The disc has been added");
52 }
53
54 // Method to remove a DVD
55 1 usage
56 public void removeDigitalVideoDisc(DigitalVideoDisc disc) {
57     // Search for disc
58     int indexOfRemoved = -1;
59     DigitalVideoDisc []newItemsOrdered = getItemsOrdered();
60     for (int i = 0; i < getQtyOrdered(); i++) {
61         if (newItemsOrdered[i].equals(disc)) {
62             indexOfRemoved = i;
63             break;
64         }
65     }
66
67     // If not found
68     if (indexOfRemoved == -1) {
69         System.out.println("The disc is not found");
70         return;
71     }
72 }
```

Figure 7: Cart class code

```

71
72     // Remove
73     for (int i = indexOfRemoved; i < getQtyOrdered() - 1; i++) {
74         newItemsOrdered[i] = newItemsOrdered[i + 1];
75     }
76     setQtyOrdered(getQtyOrdered() - 1);
77     newItemsOrdered[getQtyOrdered() - 1] = null;
78     setItemsOrdered(newItemsOrdered);
79
80     // Notify
81     System.out.println("The disc has been removed");
82 }
83
84 // Method to calculate the total cost
85 // 2 usages
86 public double totalCost() {
87     double cost = 0;
88     DigitalVideoDisc []itemsOrdered = getItemsOrdered();
89     for (DigitalVideoDisc disc: itemsOrdered) {
90         if (disc != null) {
91             cost += disc.getCost();
92         }
93     }
94 }

```

Figure 8: Cart class code

```

91     }
92 }
93
94     return Math.round(cost * 100.0) / 100.0;
95 }
96 }
97

```

Figure 9: Cart class code



DigitalVideoDisc class:

```
1 package lab02.AimsProject;
2
3 public class DigitalVideoDisc {
4     // Attribute
5     private String title;
6     private String category;
7     private String director;
8     private int length;
9     private double cost;
10
11     // Constructor
12     public DigitalVideoDisc(String title) {
13         this.title = title;
14     }
15
16     public DigitalVideoDisc(String title, String category, double cost) {
17         this(title);
18         this.category = category;
19         this.cost = cost;
20     }
21 }
```

Figure 10: DVD class code

```
21
    1 usage
22  ✓ public DigitalVideoDisc(String title, String category, String director, double cost) {
23      this(title, category, cost);
24      this.director = director;
25  }
26
    2 usages
27  ✓ public DigitalVideoDisc(String title, String category, String director, int length, double cost) {
28      this(title, category, director, cost);
29      this.length = length;
30  }
31
    // Getter
    no usages
33  ✓ public String getTitle() {
34      return title;
35  }
36
    no usages
37  ✓ public String getCategory() {
38      return category;
39  }
40
```

Figure 11: DVD class code

```
41  ✓ public String getDirector() {
42      return director;
43  }
44
    no usages
45  ✓ public int getLength() {
46      return length;
47  }
48
    1 usage
49  ✓ public double getCost() {
50      return cost;
51  }
52  }
53
```

Figure 12: DVD class code

## 6. Demo chương trình

```
C:\Users\Manh\.jdk\openjdk-21\bin\
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:
43.94

Process finished with exit code 0
```

Figure 13: Result

## 7. Reading Assignment

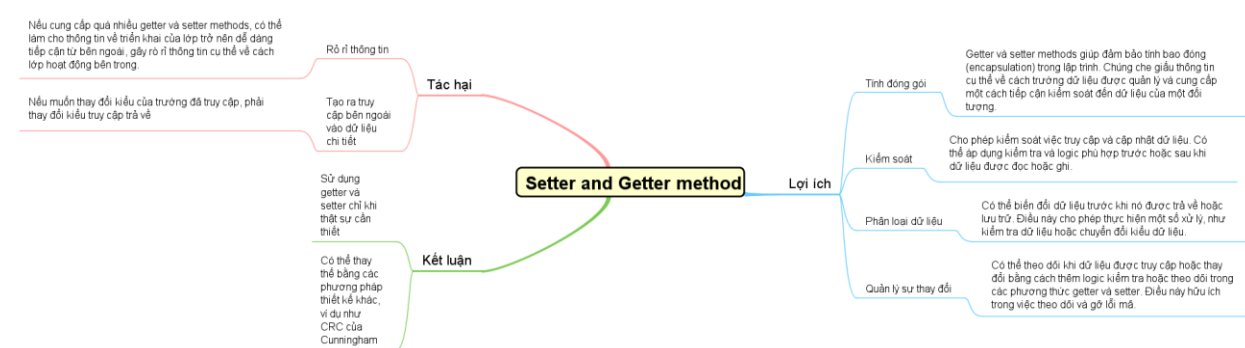


Figure 14: Setter and Getter method

## 8. Answer the questions

Java không cho phép tạo 2 constructor như thế, vì chúng có cùng chữ kí (DigitalVideoDisc(String))