

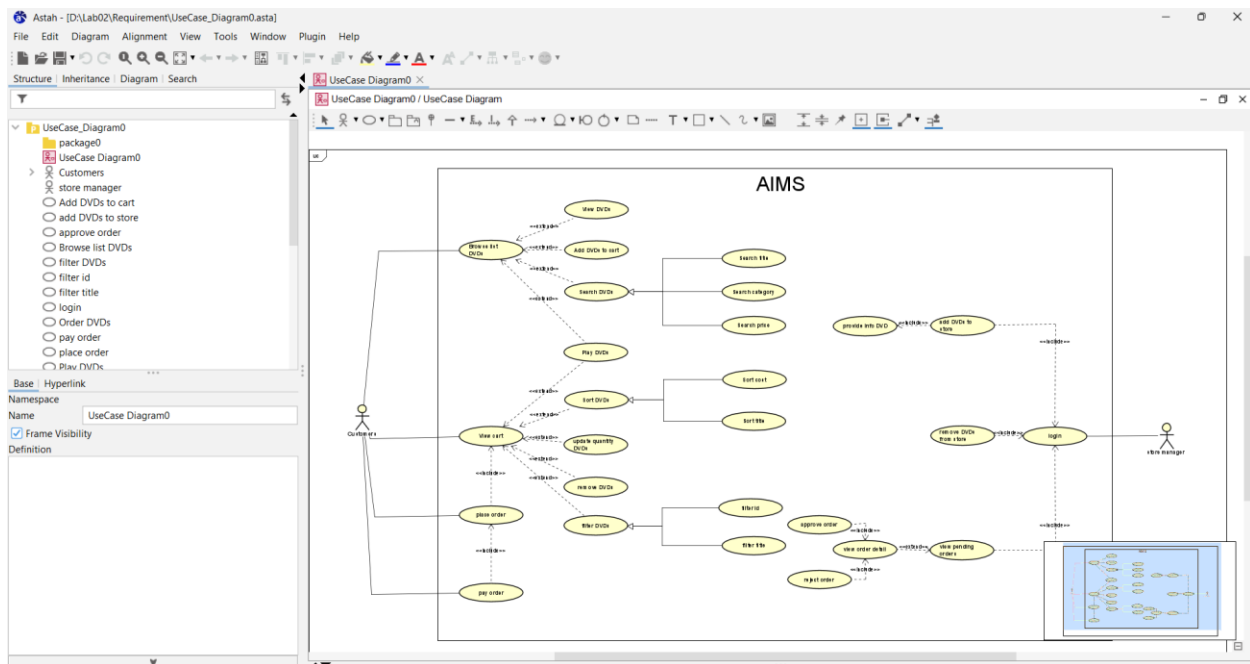
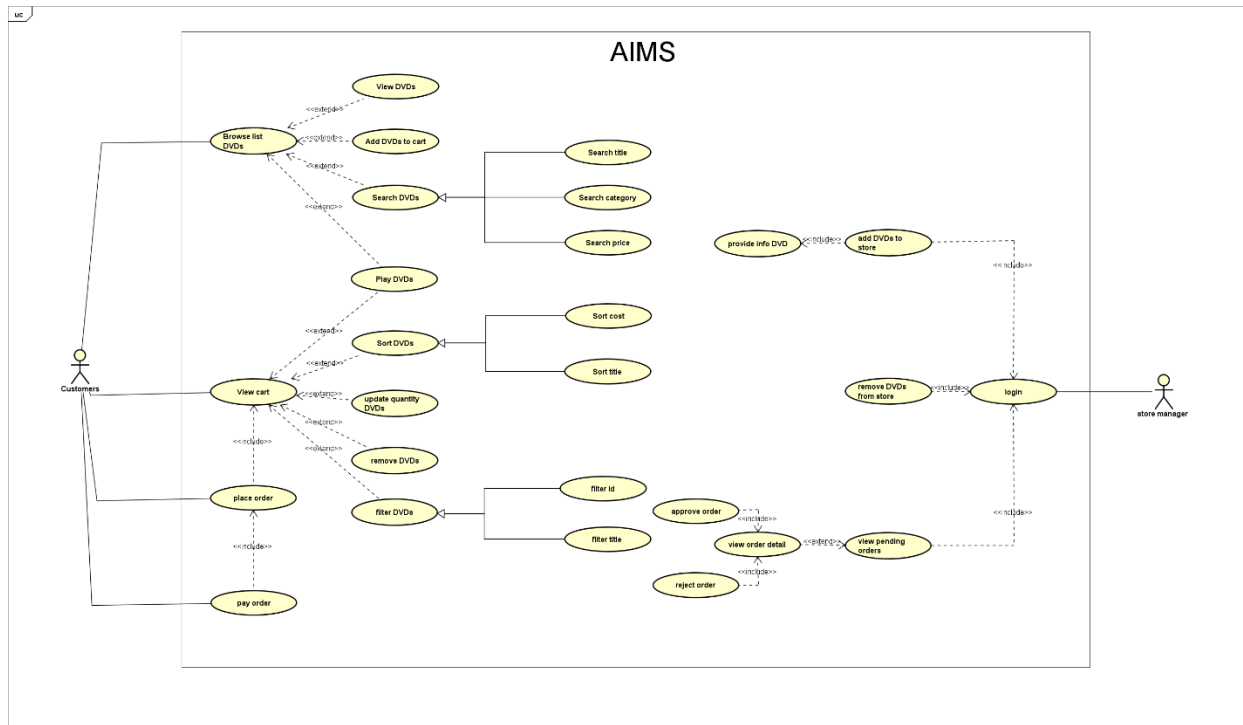
# Object-Oriented Programming

## **Lab 02: Problem Modeling and Encapsulation**

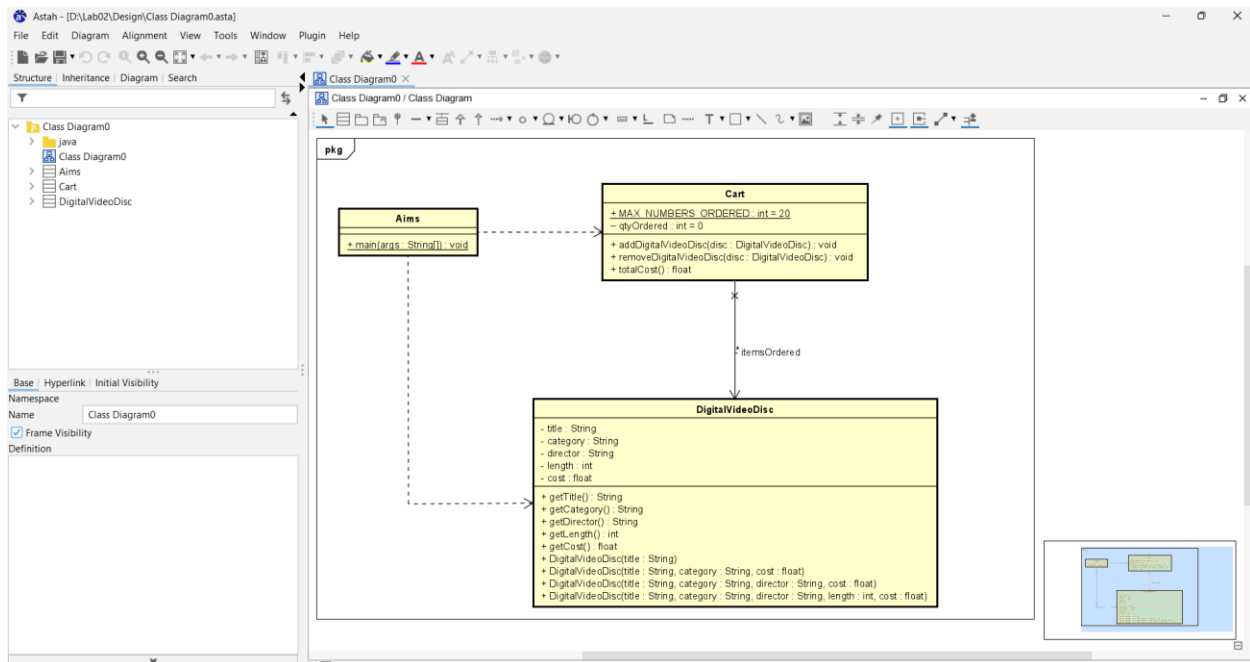
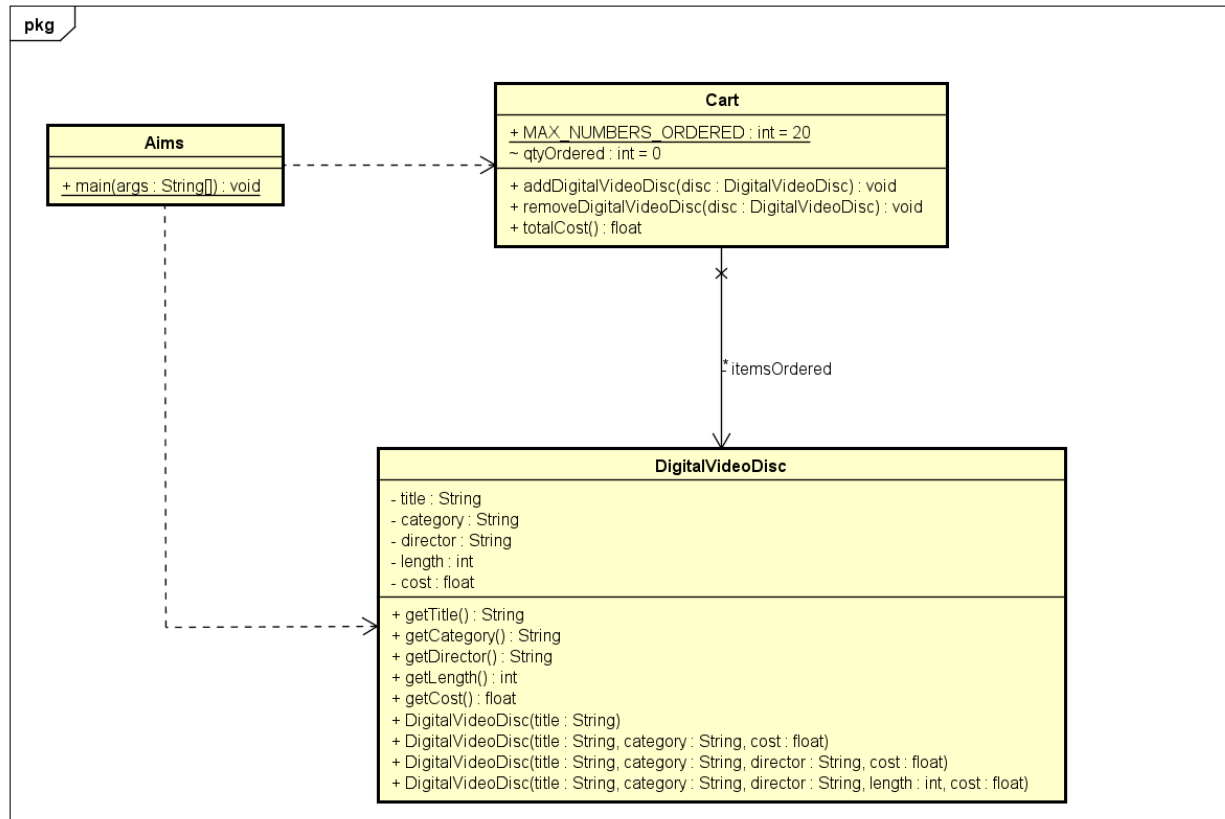
Họ và tên: Nguyễn Bá Hoàng

MSSV: 20225844

# 1. Use case diagram



## 2. Class Diagram



### 3. Aims class

```
1  import java.*;
2  public class Aims {
3
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", category: "Animation", director: "Roger Allers", length: 87, cost: 19.95f);
10         anOrder.addDigitalVideoDisc(dvd1);
11
12         DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star Wars", category: "Science Fiction", director: "George Lucas", length: 87, cost: 24.95f);
13         anOrder.addDigitalVideoDisc(dvd2);
14
15         DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladin", category: "Animation", cost: 18.99f);
16         anOrder.addDigitalVideoDisc(dvd3);
17
18         //print total cost of the items in the cart
19         System.out.println("Total Cost is: ");
20         System.out.println(anOrder.totalCost());
21
22
23         //removing items not found in the cart
24         DigitalVideoDisc dvd4 = new DigitalVideoDisc( title: "test_removeNotFound");
25         anOrder.removeDigitalVideoDisc(dvd4); // test ko tim thay dvd trong cart
26
27         //test removing dvd2
28         anOrder.removeDigitalVideoDisc(dvd2);
29         System.out.println("Total Cost khi xoa dvd 2: ");
30         System.out.println(anOrder.totalCost());
31     }
32 }
```

## 4. DigitalVideoDisc class

```
1 public class DigitalVideoDisc { 12 usages
2     private String title; 5 usages
3     private String category; 4 usages
4     private String director; 3 usages
5     private int length; 2 usages
6     private float cost; 4 usages
7
8     public String getTitle() { 2 usages
9         return title;
10    }
11
12    public String getCategory() { no usages
13        return category;
14    }
15
16    public String getDirector() { no usages
17        return director;
18    }
19
20    public int getLength() { no usages
21        return length;
22    }
23
24    public float getCost() { 1 usage
25        return cost;
26    }
27
28
29    public DigitalVideoDisc(String title) { 1 usage
30        this.title = title;
31    }
32 }
```

```
31    }
32
33    public DigitalVideoDisc(String title, String category, float cost) { 1 usage
34        this.title = title;
35        this.category = category;
36        this.cost = cost;
37    }
38
39    public DigitalVideoDisc(String title, String category, String director, float cost) { no usages
40        this.title = title;
41        this.director = director;
42        this.category = category;
43        this.cost = cost;
44    }
45
46    public DigitalVideoDisc(String title, String category, String director, int length, float cost) { 2 usages
47        this.title = title;
48        this.category = category;
49        this.director = director;
50        this.length = length;
51        this.cost = cost;
52    }
53
54
55 }
56 }
```

## 5. Cart class

```
import java.lang.*;

public class Cart {

    public static final int MAX_NUMBERS_ORDERED = 20;
    private DigitalVideoDisc itemsOrdered[] = new DigitalVideoDisc[MAX_NUMBERS_ORDERED];
    int qtyOrdered = 0;

    public void addDigitalVideoDisc(DigitalVideoDisc disc){
        for(int i = 0; i < 20; i++) {
            if (qtyOrdered != 20) {
                if(itemsOrdered[i] == null) {
                    itemsOrdered[i] = disc;
                    System.out.println("The disc has been added");
                    qtyOrdered += 1;
                    break;
                }
            } else {
                System.out.println("The cart is almost full");
                break;
            }
        }
    }
}
```

```
    public void removeDigitalVideoDisc(DigitalVideoDisc disc){
        for(int i = 0; i < 20; i++){
            if(itemsOrdered[i] != null) {
                if (itemsOrdered[i].getTitle().equals(disc.getTitle())) {
                    itemsOrdered[i] = null;
                    System.out.println("xoa thanh cong");
                    qtyOrdered -= 1;
                    break;
                }
            }
            if(i == 19){
                System.out.println("xoa khong thanh cong");
            }
        }
    }

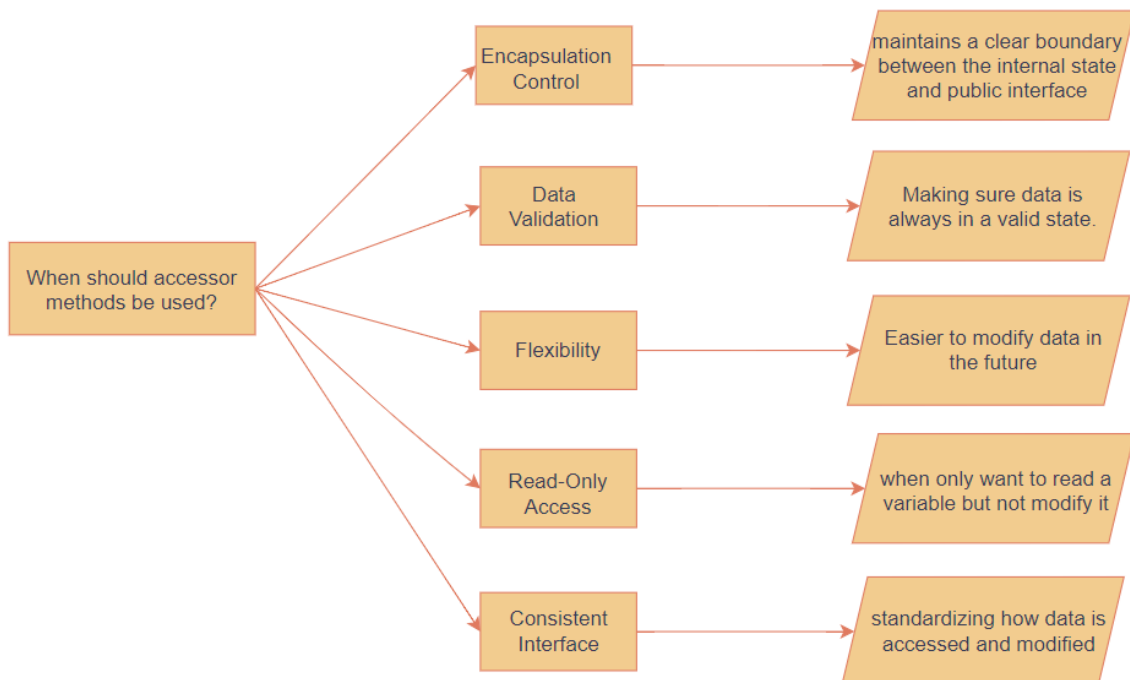
    public float totalCost(){
        float cost = 0;
        for(int i = 0; i < 20; i++){
            if(itemsOrdered[i] != null){
                cost += itemsOrdered[i].getCost();
            }
        }
        return cost;
    }
}
```

## 6. kết quả chạy thử

```
AP AimsProject Version control Current File
Aims.java DigitalVideoDisc.java Cart.java
1 import java.*;
2 public class Aims {
3
4     public static void main(String[] args){
5         //Create a new cart
6         Cart anOrder = new Cart();
7
Run Aims
C:\Program Files\Java\jdk-1.8\bin\java.exe ...
The disc has been added
The disc has been added
The disc has been added
Total Cost is:
63.89
xoa khong thanh cong
xoa thanh cong
Total Cost khi xoa dvd 2:
38.940002
Process finished with exit code 0
```

## 7. trả lời câu hỏi

**Question 1:** When should accessor methods be used?



**Question 2:** 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?

Trả lời: Java không cho phép điều này. Bởi vì do category và title cùng kiểu dữ liệu là String nên khi truyền tham số để khởi tạo thì java sẽ không xác định được phải khởi tạo bằng constructor nào (chữ ký giống nhau vì cùng tên phương thức, cùng kiểu tham số truyền vào, cùng số lượng tham số truyền vào)