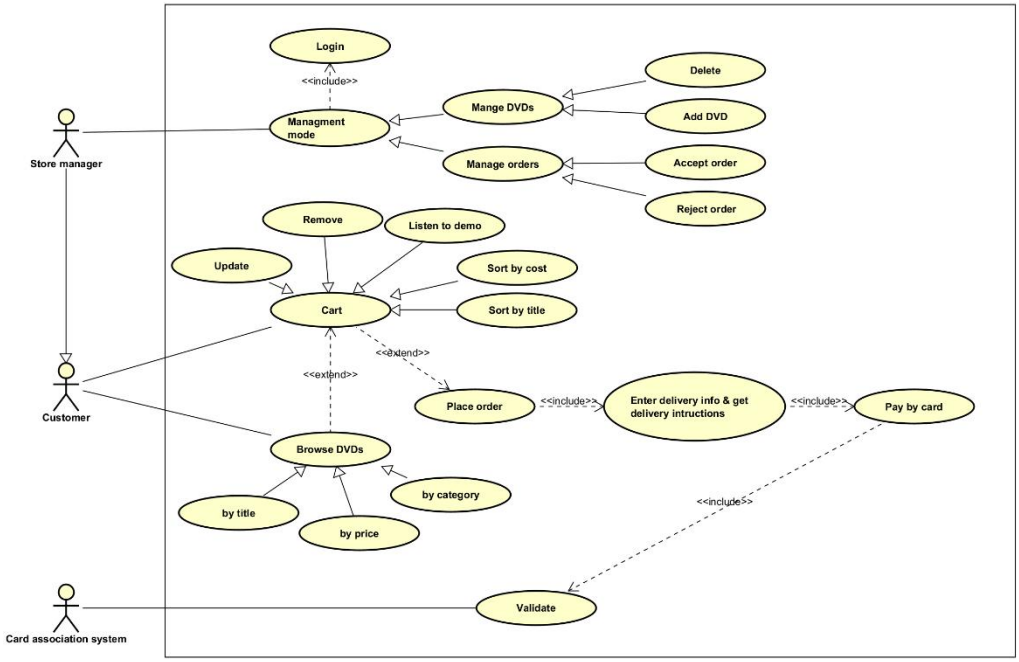
Lab 2. Problem Modeling and Encapsulation

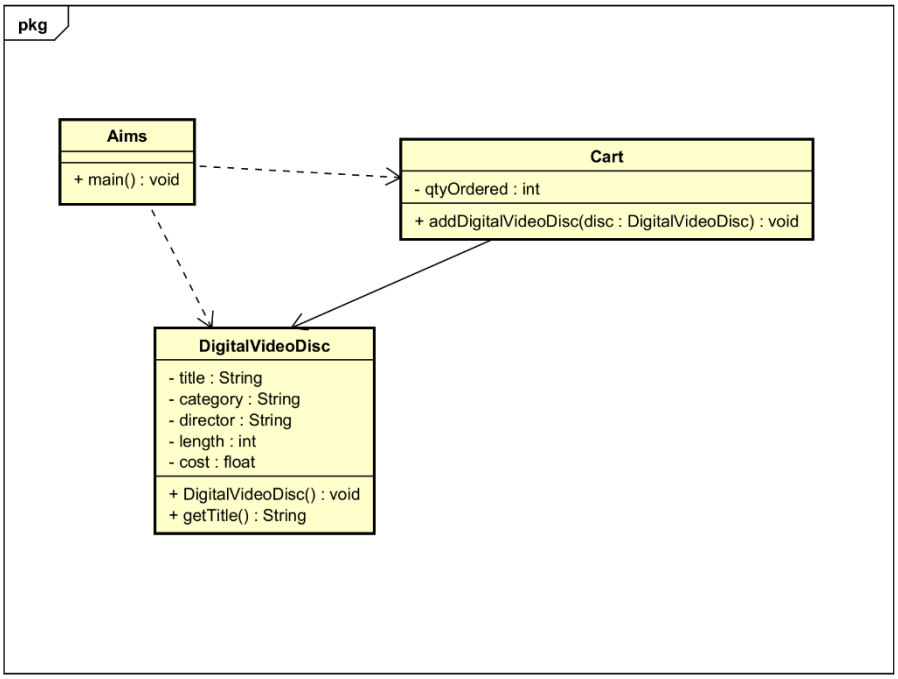
IT3103 – 7850868 – Thực hành Lập Trình Hướng Đối Tượng

Họ và tên: Lê Đức Anh **MSSV: 202417097**

Usecase Diagram:



Class Diagram:



DigitalVideoDisc class code:

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


    public String getTitle(){
        return title;
    }
    public String getCategory(){
        return category;
    }
    public String getDirector(){
        return director;
    }
    public int length(){
        return length;
    }
    public float getCost(){
        return cost;
    }

}
```

Cart class:

```java
import java.util.LinkedList;

public class Cart {
    public static final int MAX_NUMBERS_ORDERED = 20;
    LinkedList<DigitalVideoDisc> itemsOrdered = new LinkedList<>
();
    public void addDigitalVideoDisc(DigitalVideoDisc a) {
        if (itemsOrdered.size() < MAX_NUMBERS_ORDERED) {
            itemsOrdered.add(a);
            System.out.println(a.getTitle()+" has been added.");
            System.out.println("Cost: " + totalCost());
        } else {
            System.out.println("The cart is full!");
        }
    }

    public double totalCost() {
        double sum = 0;
        for (int i = 0; i < itemsOrdered.size(); i++) {
            sum += itemsOrdered.get(i).getCost();
        }
        return sum;
    }

    public void removeDigitalVideoDisc(DigitalVideoDisc a) {
        if (itemsOrdered.remove(a)) {
            System.out.println("The disc has been removed.");
        } else {
            System.out.println("The disc is not in the cart.");
        }
    }
}
```

Aim class:

```
public class Aims {
    public static void main(String[] args) {
        Cart anOrder = new Cart();

        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87,
(float)19.95);
        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Fiction", "George Lucas", 124,
(float)24.95);
        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladdin", "Animation", "John Musker", 90, (float)18.99);

        anOrder.addDigitalVideoDisc(dvd1);
        anOrder.addDigitalVideoDisc(dvd2);
        anOrder.addDigitalVideoDisc(dvd3);

        System.out.println("Total cost is: " + anOrder.totalCost());

        anOrder.removeDigitalVideoDisc(dvd1);

        System.out.println("Total cost is: " + anOrder.totalCost());
    }
}
```

Result:

```
The Lion King has been added.
Cost: 19.950000762939453
Star Wars has been added.
Cost: 44.900001525878906
Aladdin has been added.
Cost: 63.89000129699707
Total cost is: 63.89000129699707
The disc has been removed.
Total cost is: 43.94000053405762
```

Question:

Reading Assignment: When should accessor methods be used? Read the following article and find the best possible answer to the above question: Holub, Allen. "Why getter and setter methods are evil" JavaWorld, 5 Sep. 2003

- It violates Encapsulations rule by exposing private data.
- You should use accessor methods (getters and setters) when you need to protect an object's data and control how it's accessed or changed.
- Don't add getter and setter for all variables.

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?

- No, Java does not allow you to do that. This will result in a compiler error.