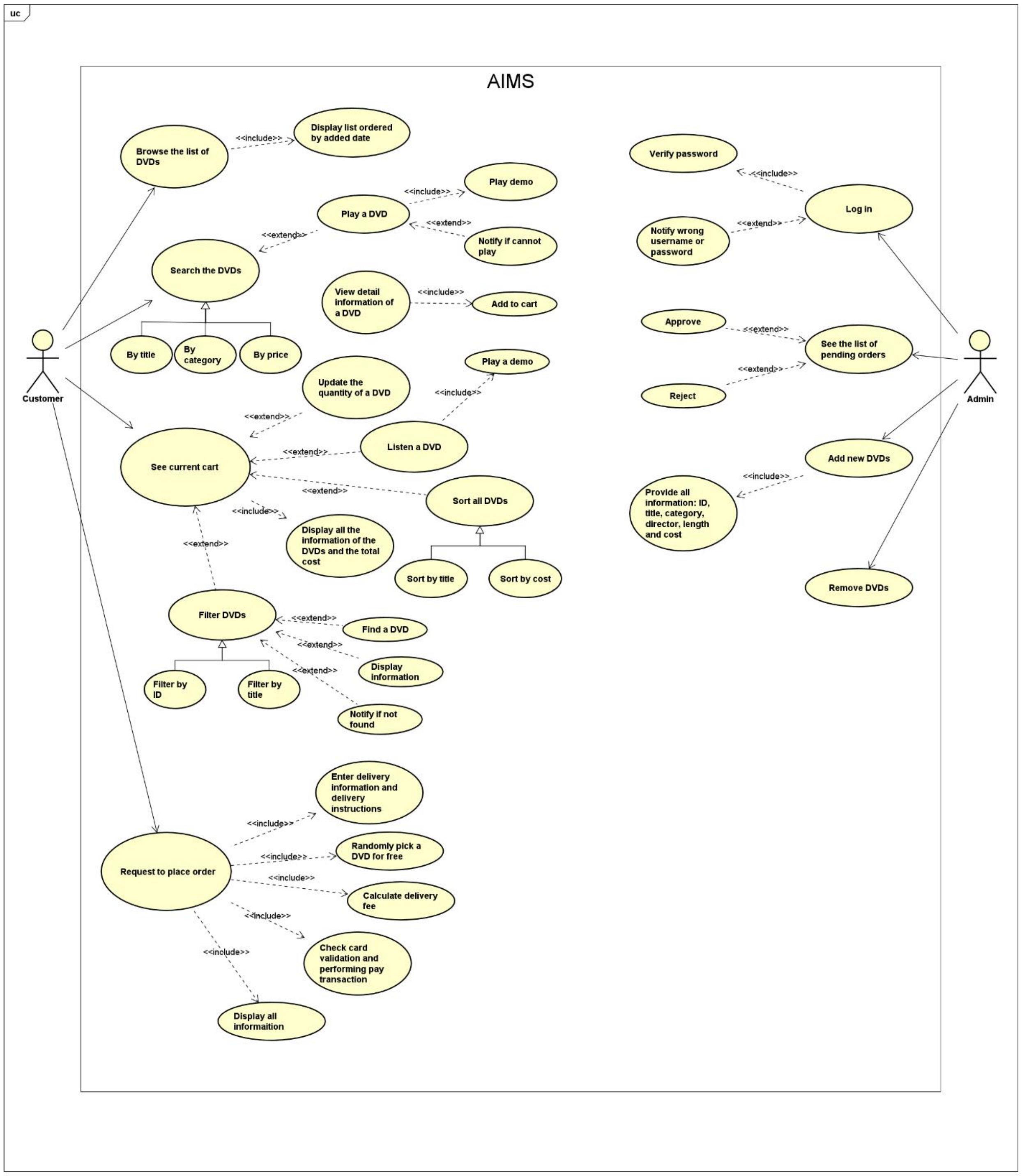
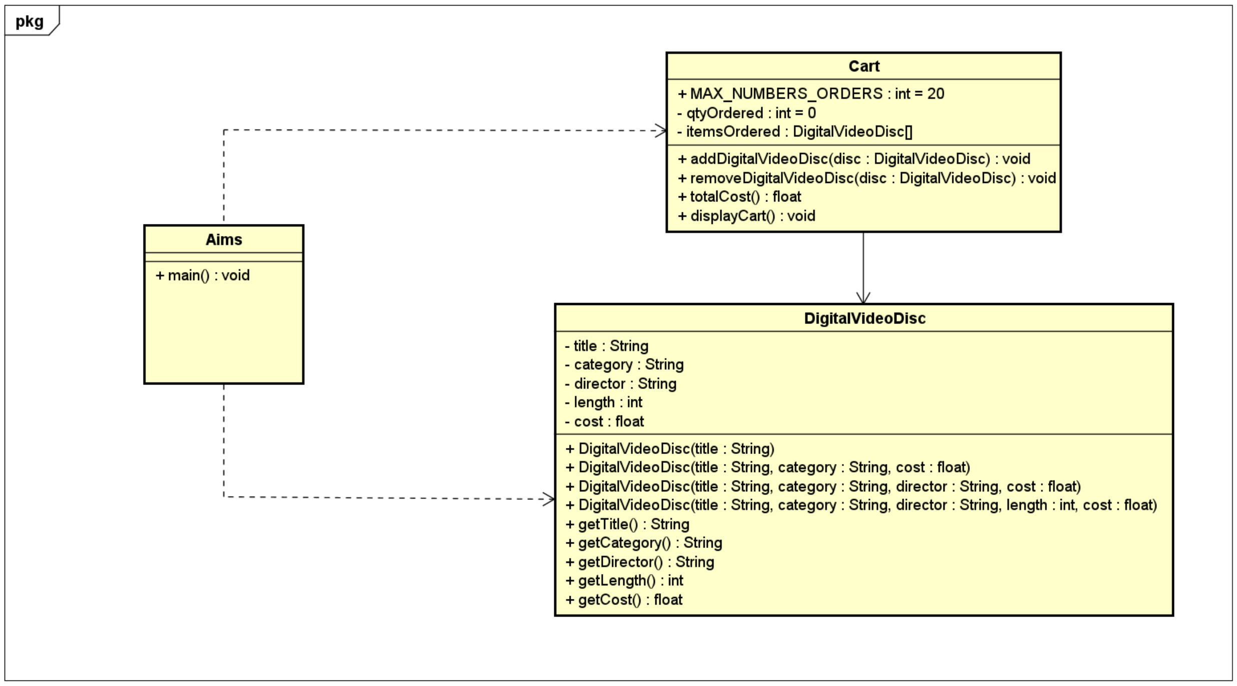
Lab02 Report

Name: Lương Minh Hiếu ID: 20230083

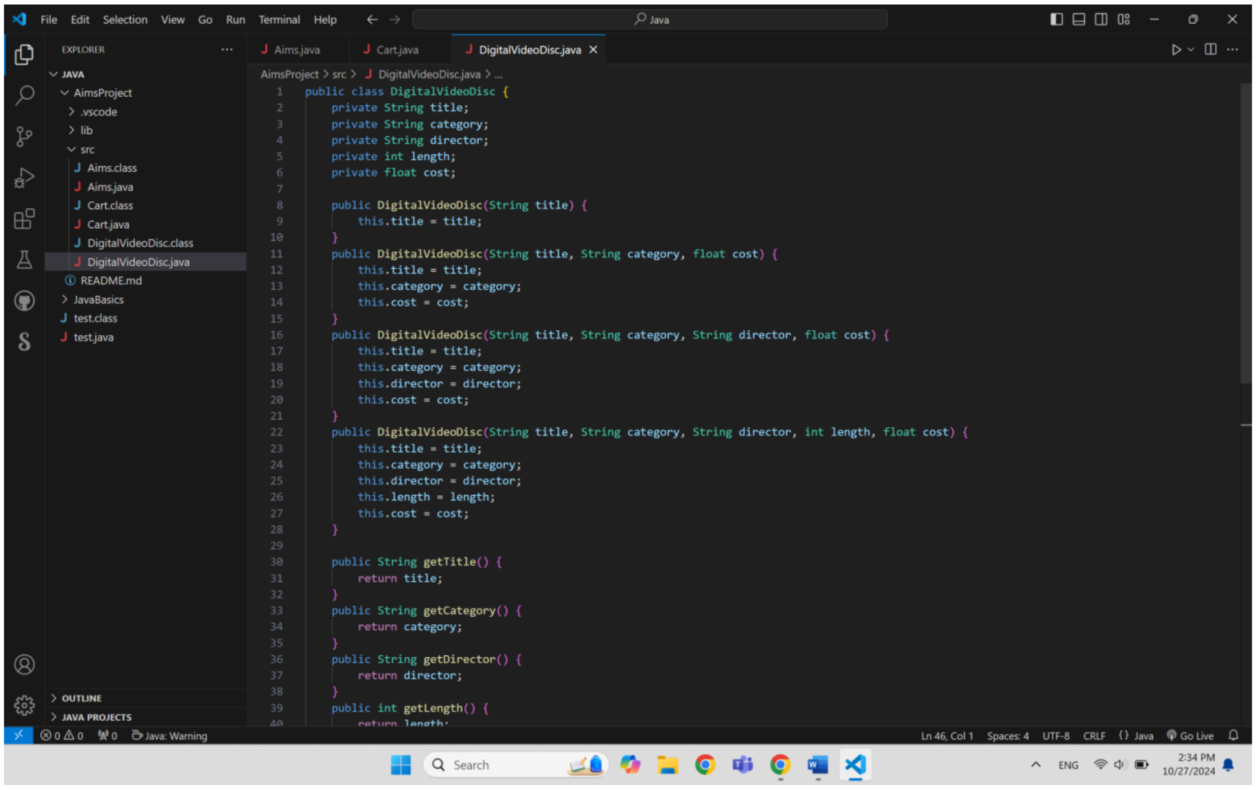
1. Use-case diagram



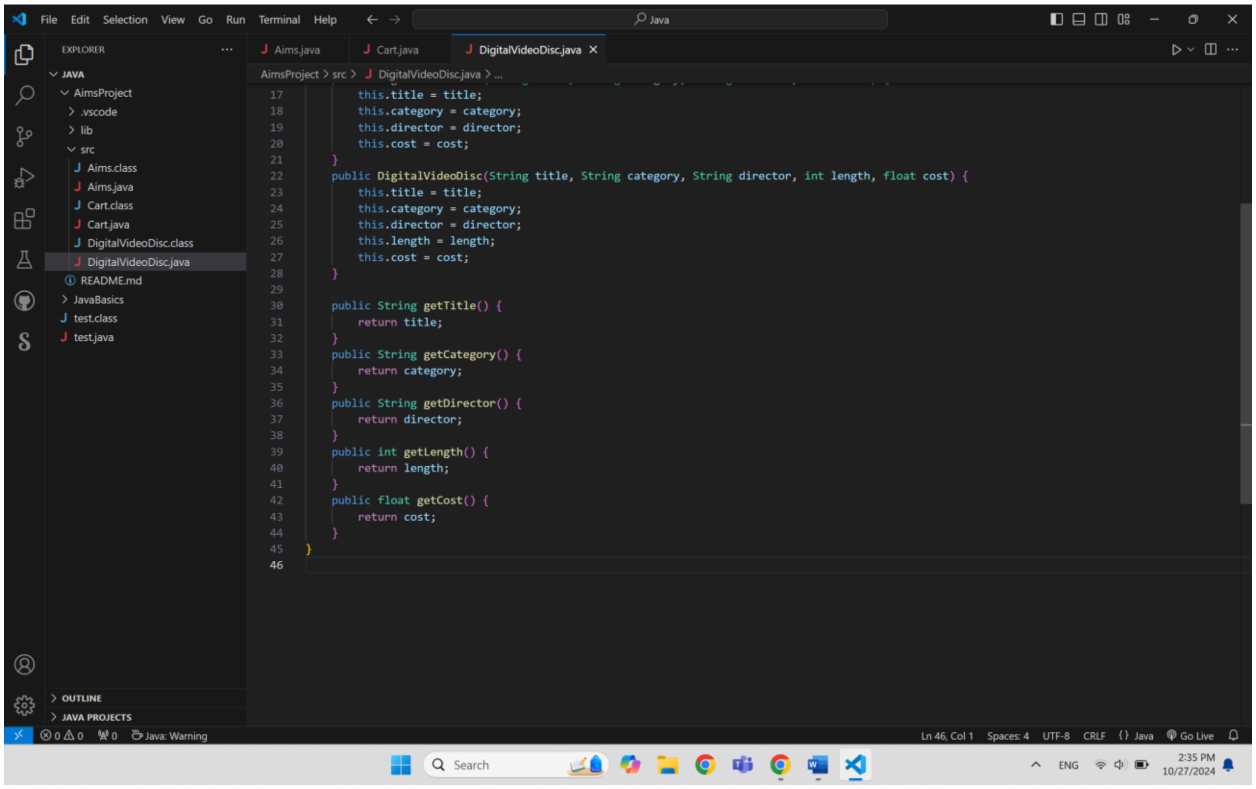
2. Class diagram


pkg
Cart
+ MAX_NUMBERS_ORDERS : int = 20
- qtyOrdered : int = 0
- itemsOrdered : DigitalVideoDisc[]
+ addDigitalVideoDisc(disc : DigitalVideoDisc) : void
+ removeDigitalVideoDisc(disc : DigitalVideoDisc) : void
+ totalCost() : float
+ displayCart() : void
Aims
+ main() : void
DigitalVideoDisc
- title : String
- category : String
- director : String
- length : int
- cost : float
+ DigitalVideoDisc(title : String)
+ DigitalVideoDisc(title : String, category : String, cost : float)
+ DigitalVideoDisc(title : String, category : String, director : String, cost : float)
+ DigitalVideoDisc(title : String, category : String, director : String, length : int, cost : float)
+ getTitle() : String
+ getCategory() : String
+ getDirector() : String
+ getLength() : int
+ getCost() : float

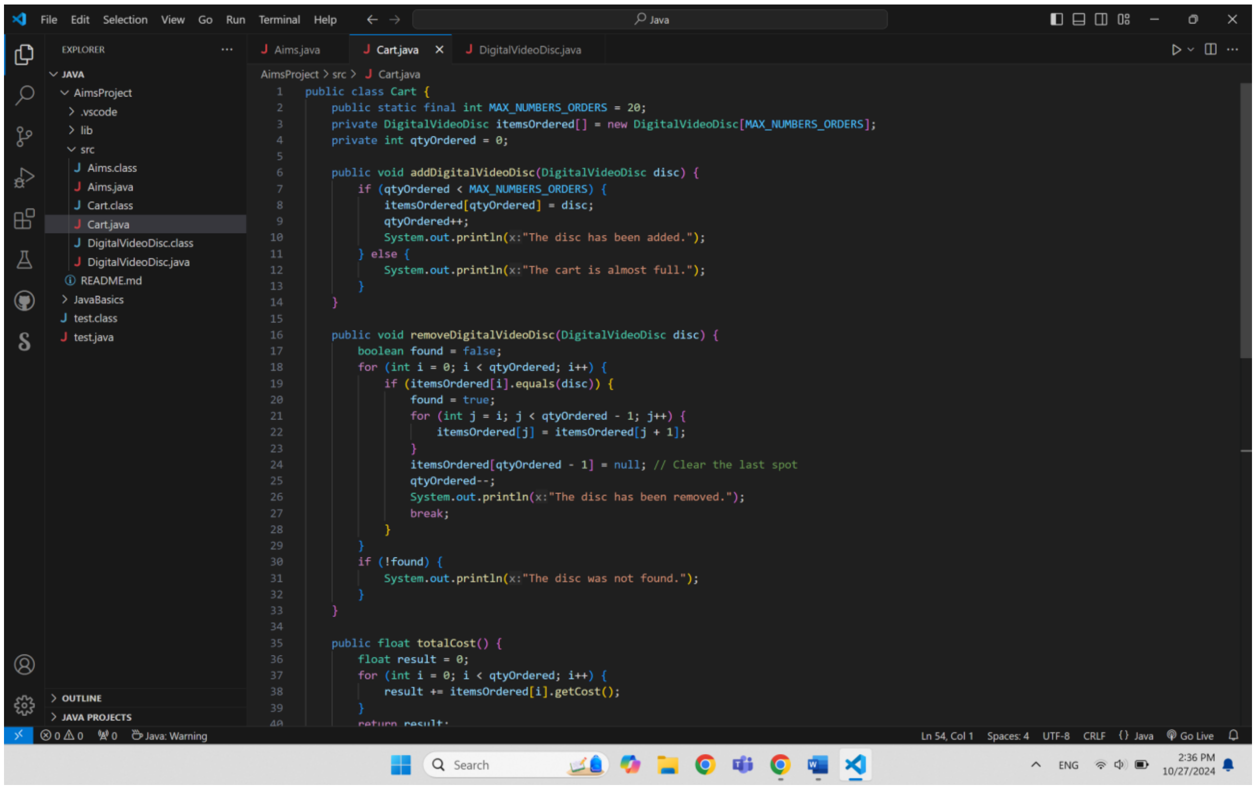
3. Source code



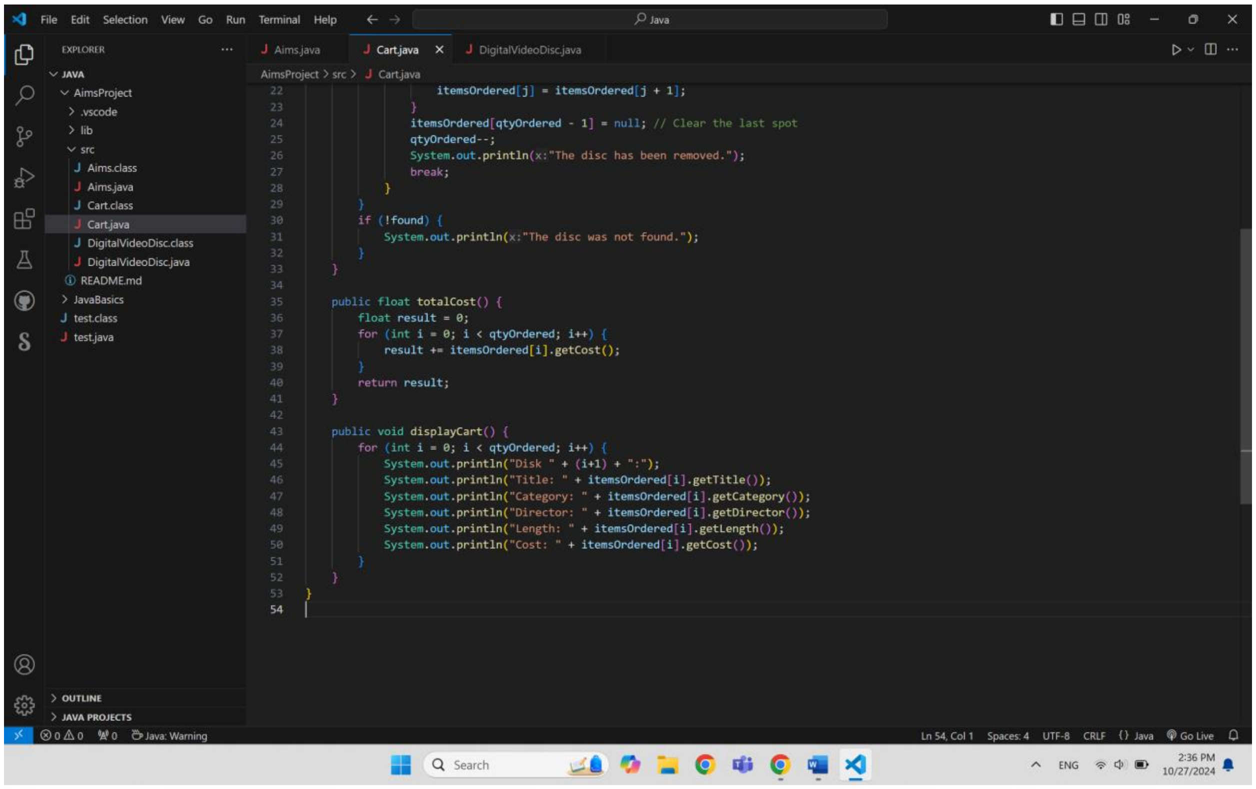
```java
public class DigitalVideoDisc {
    private String title;
    private String category;
    private String director;
    private int length;
    private float cost;

    public DigitalVideoDisc(String title) {
        this.title = title;
    }
    public DigitalVideoDisc(String title, String category, float cost) {
        this.title = title;
        this.category = category;
        this.cost = cost;
    }
    public DigitalVideoDisc(String title, String category, String director, float cost) {
        this.title = title;
        this.category = category;
        this.director = director;
        this.cost = 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 getLength() {
        return length;
```



```java
        this.title = title;
        this.category = category;
        this.director = director;
        this.cost = 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 getLength() {
        return length;
    }
    public float getCost() {
        return cost;
    }
}
```

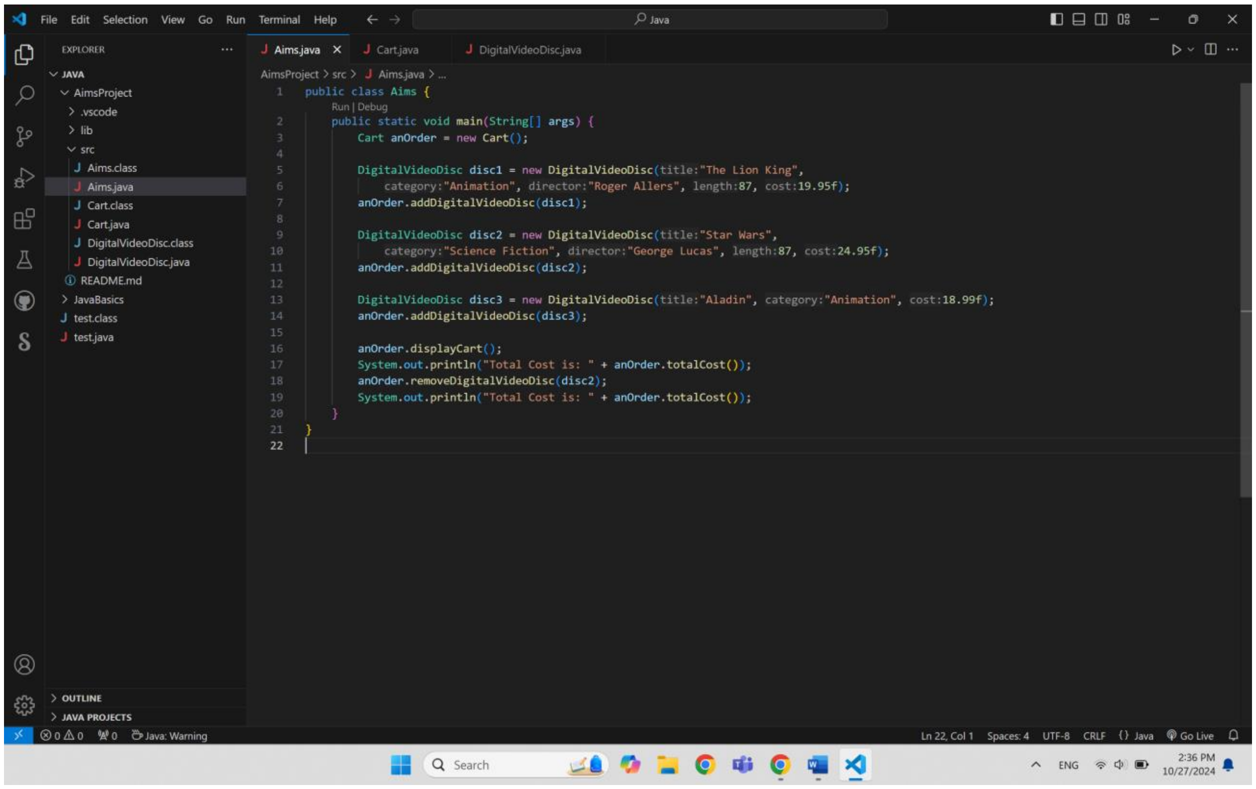


```java
public class Cart {
    public static final int MAX_NUMBERS_ORDERS = 20;
    private DigitalVideoDisc itemsOrdered[] = new DigitalVideoDisc[MAX_NUMBERS_ORDERS];
    private int qtyOrdered = 0;

    public void addDigitalVideoDisc(DigitalVideoDisc disc) {
        if (qtyOrdered < MAX_NUMBERS_ORDERS) {
            itemsOrdered[qtyOrdered] = disc;
            qtyOrdered++;
            System.out.println(x:"The disc has been added.");
        } else {
            System.out.println(x:"The cart is almost full.");
        }
    }

    public void removeDigitalVideoDisc(DigitalVideoDisc disc) {
        boolean found = false;
        for (int i = 0; i < qtyOrdered; i++) {
            if (itemsOrdered[i].equals(disc)) {
                found = true;
                for (int j = i; j < qtyOrdered - 1; j++) {
                    itemsOrdered[j] = itemsOrdered[j + 1];
                }
                itemsOrdered[qtyOrdered - 1] = null; // Clear the last spot
                qtyOrdered--;
                System.out.println(x:"The disc has been removed.");
                break;
            }
        }
        if (!found) {
            System.out.println(x:"The disc was not found.");
        }
    }

    public float totalCost() {
        float result = 0;
        for (int i = 0; i < qtyOrdered; i++) {
            result += itemsOrdered[i].getCost();
        }
        return result;
```



AimsProject > src > Cart.java

```java
                    itemsOrdered[j] = itemsOrdered[j + 1];
                }
                itemsOrdered[qtyOrdered - 1] = null; // Clear the last spot
                qtyOrdered--;
                System.out.println(x:"The disc has been removed.");
                break;
            }
        }
        if (!found) {
            System.out.println(x:"The disc was not found.");
        }
    }

    public float totalCost() {
        float result = 0;
        for (int i = 0; i < qtyOrdered; i++) {
            result += itemsOrdered[i].getCost();
        }
        return result;
    }

    public void displayCart() {
        for (int i = 0; i < qtyOrdered; i++) {
            System.out.println("Disk " + (i+1) + ":");
            System.out.println("Title: " + itemsOrdered[i].getTitle());
            System.out.println("Category: " + itemsOrdered[i].getCategory());
            System.out.println("Director: " + itemsOrdered[i].getDirector());
            System.out.println("Length: " + itemsOrdered[i].getLength());
            System.out.println("Cost: " + itemsOrdered[i].getCost());
        }
    }
}
```



AimsProject > src > Aims.java > ...

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

        DigitalVideoDisc disc1 = new DigitalVideoDisc(title:"The Lion King",
            category:"Animation", director:"Roger Allers", length:87, cost:19.95f);
        anOrder.addDigitalVideoDisc(disc1);

        DigitalVideoDisc disc2 = new DigitalVideoDisc(title:"Star Wars",
            category:"Science Fiction", director:"George Lucas", length:87, cost:24.95f);
        anOrder.addDigitalVideoDisc(disc2);

        DigitalVideoDisc disc3 = new DigitalVideoDisc(title:"Aladin", category:"Animation", cost:18.99f);
        anOrder.addDigitalVideoDisc(disc3);

        anOrder.displayCart();
        System.out.println("Total Cost is: " + anOrder.totalCost());
        anOrder.removeDigitalVideoDisc(disc2);
        System.out.println("Total Cost is: " + anOrder.totalCost());
    }
}
```

4. Reading Assignment

Question: When should accessor methods be used?

Answer: In my opinion, accessor methods shouldn't be used unless absolutely necessary because these methods violate the encapsulation principle and one basic principle of OO systems is data abstraction. Using these methods would make your code harder to maintain. We can use a method to return an object in terms of an interface that the object implement because that interface isolates you from changing the implementing class. We can focus on the action not the information, don't ask the information you need to do the work; ask the object that has the information to do the work for you.

5. Answering questions

- 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?

  My answer is no. This will violate the method of overloading principle since these constructor have the same name, the same number of parameters and the same type of parameters.