

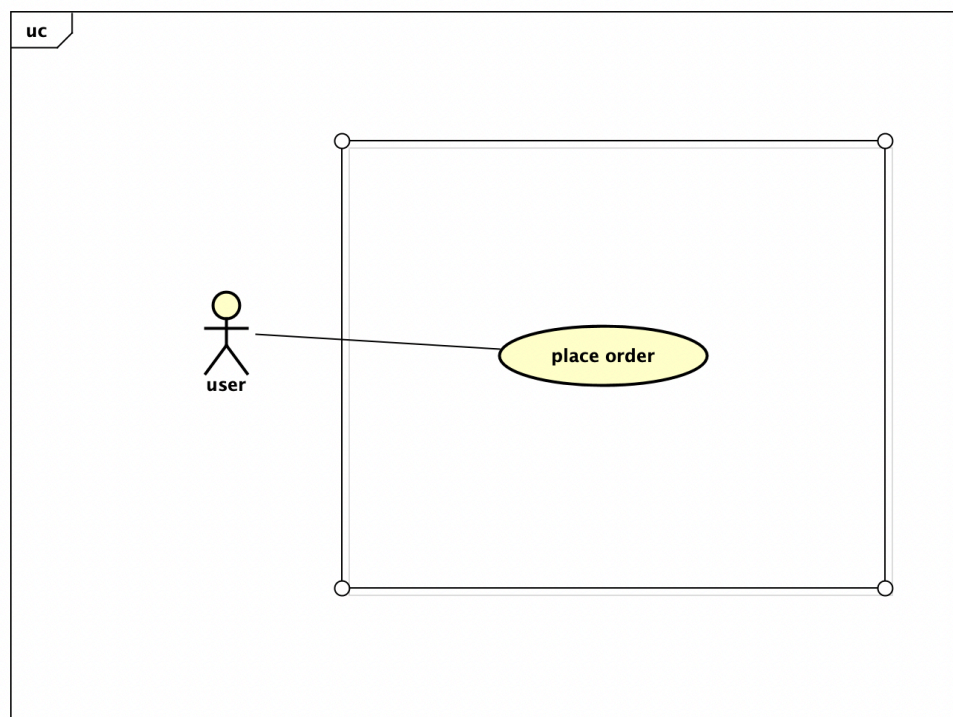
OOP- Lab 02: Problem Modeling and Encapsulation

Họ và Tên: Nguyễn Thị Kiều Oanh

MSSV: 20225659

1. UML & Astah

Vẽ thử 1 sơ đồ use case diagram trong Uml&astah



2. Problem Statement of AIMS Project

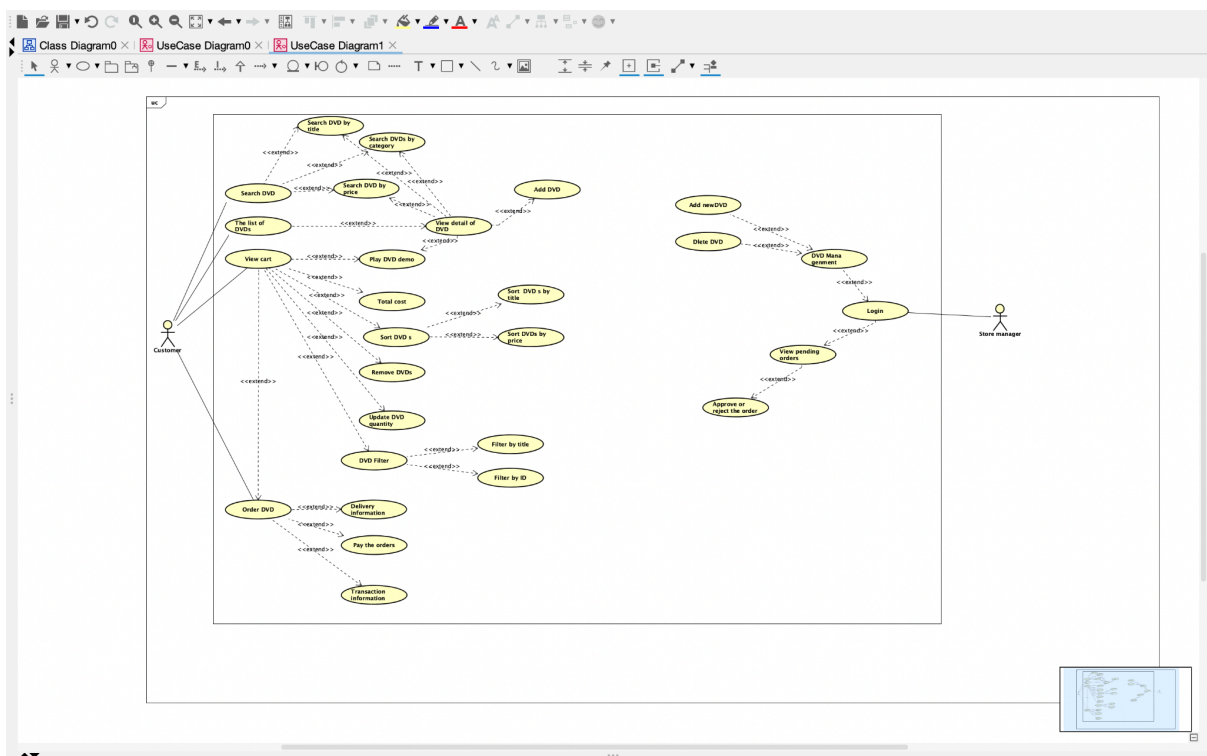
- **Yêu cầu đề bài :** Thiết kế hệ thống cho dự án AIMS. (Chỉ có 1 loại phương tiện DVD)
- **Yêu cầu hệ thống :**
 - + **Đối với Customers:**
 - Duyệt danh sách các DVD có sẵn
 - Tìm kiếm DVD theo: Tiêu đề, danh mục, giá cả
 - Xem demo 1 đoạn của DVD
 - Xem thông tin chi tiết của 1 DVD
 - Thêm DVD vào giỏ hàng
 - Xem giỏ hàng và tổng tiền, nghe demo trong giỏ hàng của DVD
 - Sắp xếp DVD theo tiêu đề hoặc chi phí trong giỏ

- Cập nhật số lượng DVD trong giỏ hàng hoặc xóa DVD trong giỏ
- Lọc DVD trong giỏ theo : ID, tiêu đề
- Đặt hàng

+ Đối với Store Manager:

- Đăng nhập, kiểm tra quyền
- Xem danh sách các đơn hàng đang được xử lý
- Xem chi tiết đơn hàng, chấp nhận/từ chối đơn hàng
- Thêm mới/ xóa danh sách sản phẩm trong Store

3. Use case diagram



4. UML Class Diagram for use cases related to cart management

• Nội dung, các đối tượng

AIMS (Đối tượng chính):

Phương thức:

+ void main()

Cart

Thuộc tính:

- qtyOrdered: int

- MAX_NUMBER_ORDER: int
- itemOrdered: DigitalVideoDisc[]

Phương thức:

- + addDigitalVideoDisc(disc: DigitalVideoDisc): int
- + totalCost(): float
- + print() : void
- + removeDigitalVideoDisc(disc: DigitalVideoDisc): int
- + searchByTitle(title : String) : void
- + searchByID(id : int) : void

DigitalVideoDisc

Thuộc tính:

- title: String
- category: String
- director: String
- length: int
- cost: float
- id : int

Phương thức:

- + DigitalVideoMusic(title: String, category: String, director: String, length: int, cost: float): void
- + getTitle(): String
- + getCategory(): String
- + getDirector(): String
- + getLength(): int
- + getCost(): float
- + setTitle(titleSet : String) : void
- + setDirector(directorSet : String) : void
- + setCategory(categorySet : String) : void
- + setLength(lengthSet : int) : void
- + setCost(costSet : float) : void
- + setID(idSet : int) : void

Store (class phụ trợ cho DigitalVideoDisc)

Thuộc tính :

- itemsInStore : LinkedList<DigitalVideoDisc>

Phương thức

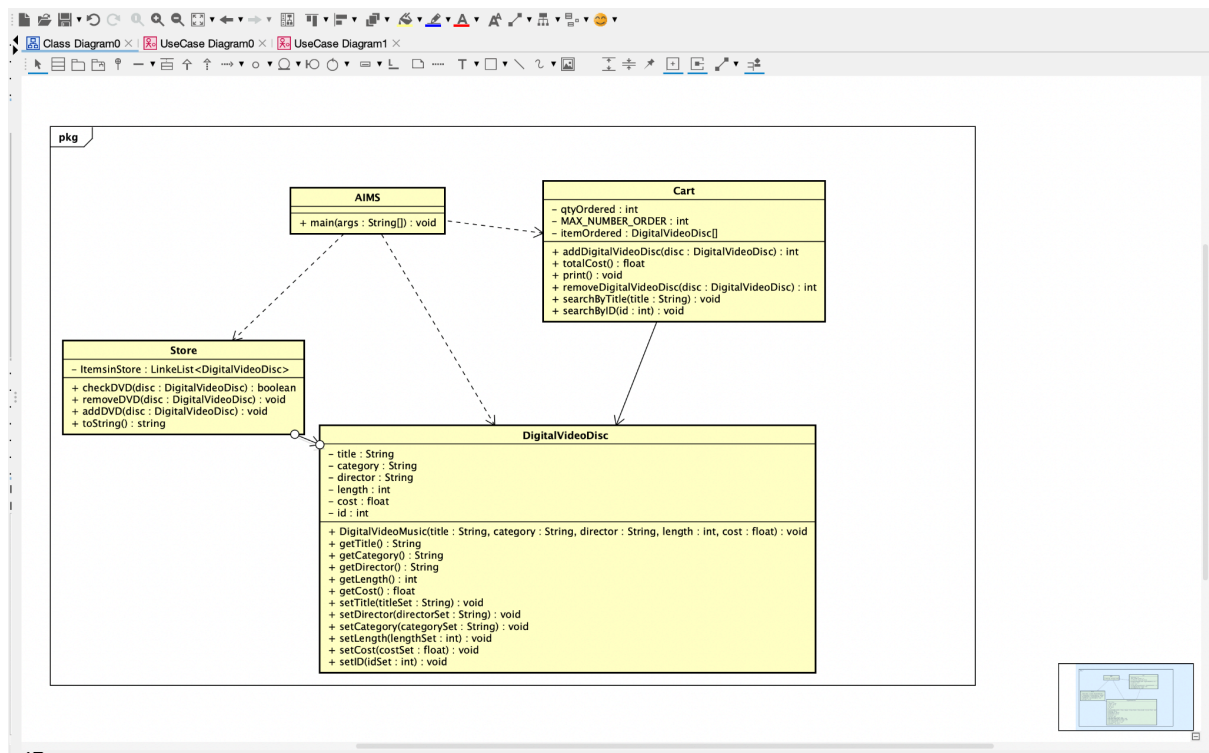
- checkDVD(disc : DigitalVideoDisc) : boolean

+ removeDVD(disc : DigitalVideoDisc) : void

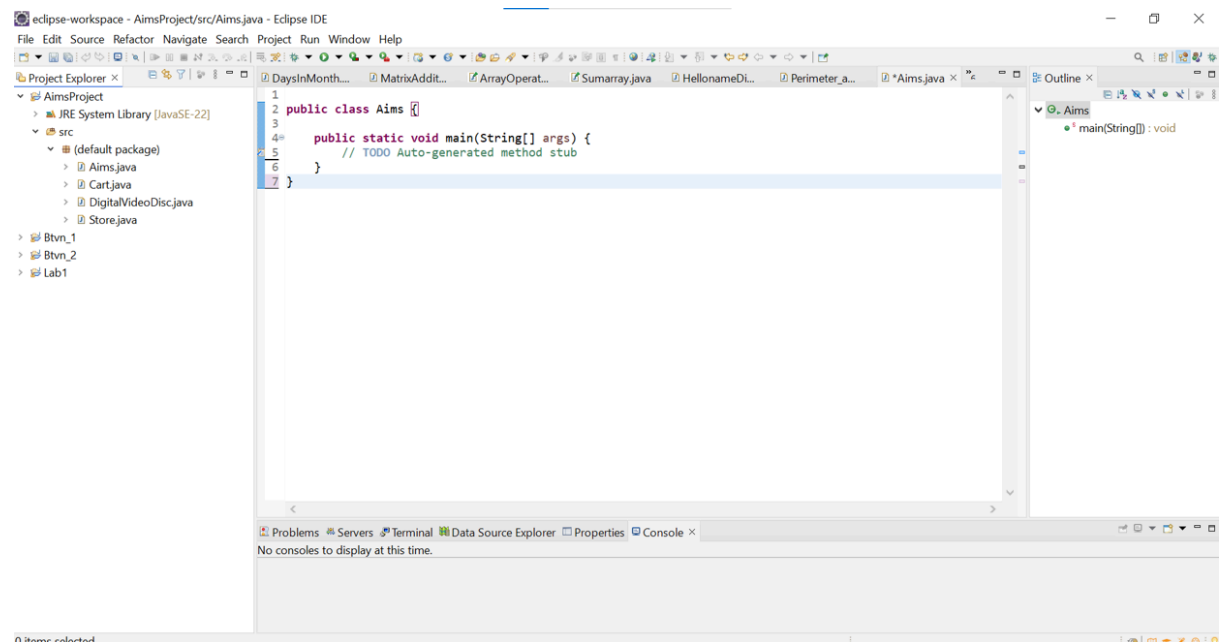
+ addDVD(disc : DigitalVideoDisc) : void

+ toString() : String

***Hình ảnh**



5. Create Aims class



6 . Create the DigitalVideoDisc (accessors, mutators and constructors)

Code :

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

    public DigitalVideoDisc(String title, String category, String director, int
length, float cost,int id) {
        this.title = title;
        this.category = category;
        this.director = director;
        this.length = length;
        this.cost = cost;
        this.id = id;
    }

    public String getTitle() {
        return title;
    }
    public int getLength() {
        return length;
    }
    public String getDirector() {
        return director;
    }
    public String getCategory() {
        return category;
    }
}
```

```

    }
    public float getCost() {
        return cost;
    }
    public int getId() {
        return id;
    }

    public void setTitle(String titleSet) {
        this.title = titleSet;
    }

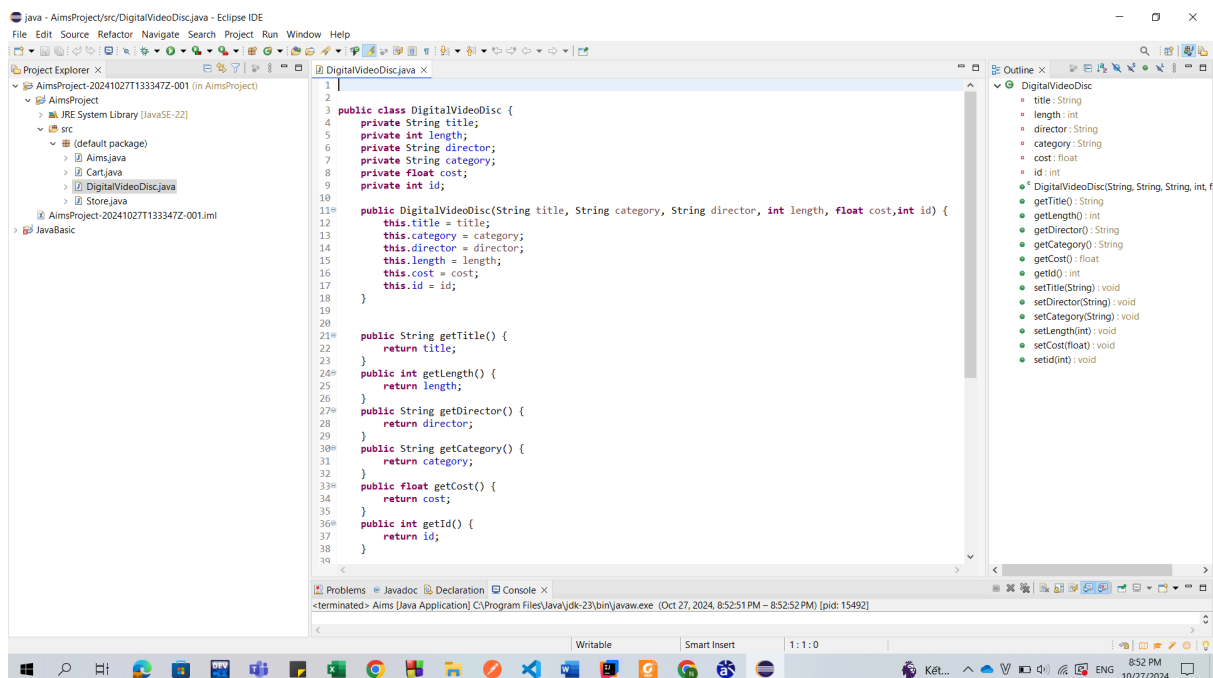
    public void setDirector(String directorSet) {
        this.director = directorSet;
    }

    public void setCategory(String categorySet) {
        this.category = categorySet;
    }

    public void setLength(int lengthSet) {
        this.length = lengthSet;
    }

    public void setCost(float costSet) {
        this.cost = costSet;
    }
    public void setId(int idSet) {
        this.id = idSet;
    }
}

```



7. Create the Cart class to work with DigitalVideoDisc
Code :

```

public class Cart {
    private static final int MAX_NUMBERS_ORDER = 100;
    private DigitalVideoDisc[] itemOrdered = new
DigitalVideoDisc[MAX_NUMBERS_ORDER];
    private int qtyOrdered = 0;

    public void addDigitalVideoDisc(DigitalVideoDisc disc) {
        if (qtyOrdered < MAX_NUMBERS_ORDER) {
            itemOrdered[qtyOrdered] = disc;
            qtyOrdered++;
            System.out.println("The disc has been added.");
        } else {
            System.out.println("The cart is full!");
        }
    }

    public void removeDigitalVideoDisc(DigitalVideoDisc disc) {
        for (int i = 0; i < qtyOrdered; i++) {
            if (itemOrdered[i] == disc) {
                for (int j = i; j < qtyOrdered - 1; j++) {
                    itemOrdered[j] = itemOrdered[j + 1];
                }
                itemOrdered[qtyOrdered - 1] = null;
                qtyOrdered--;
                System.out.println("The disc has been removed.");
                return;
            }
        }
        System.out.println("The disc is not found in the cart.");
    }

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

    public void print() {
        System.out.println("Cart contents:");
        for (int i = 0; i < qtyOrdered; i++) {
            System.out.println((i + 1) + ". " + itemOrdered[i].getTitle() + " - "
+ itemOrdered[i].getCost() + "$");
        }
    }

    public void searchByTitle(String title) {
        for (int i = 0; i < qtyOrdered; i++) {
            if (itemOrdered[i].getTitle().equalsIgnoreCase(title)) {
                System.out.println("Found: " + itemOrdered[i].getTitle());
                return;
            }
        }
        System.out.println("No disc with title: " + title);
    }

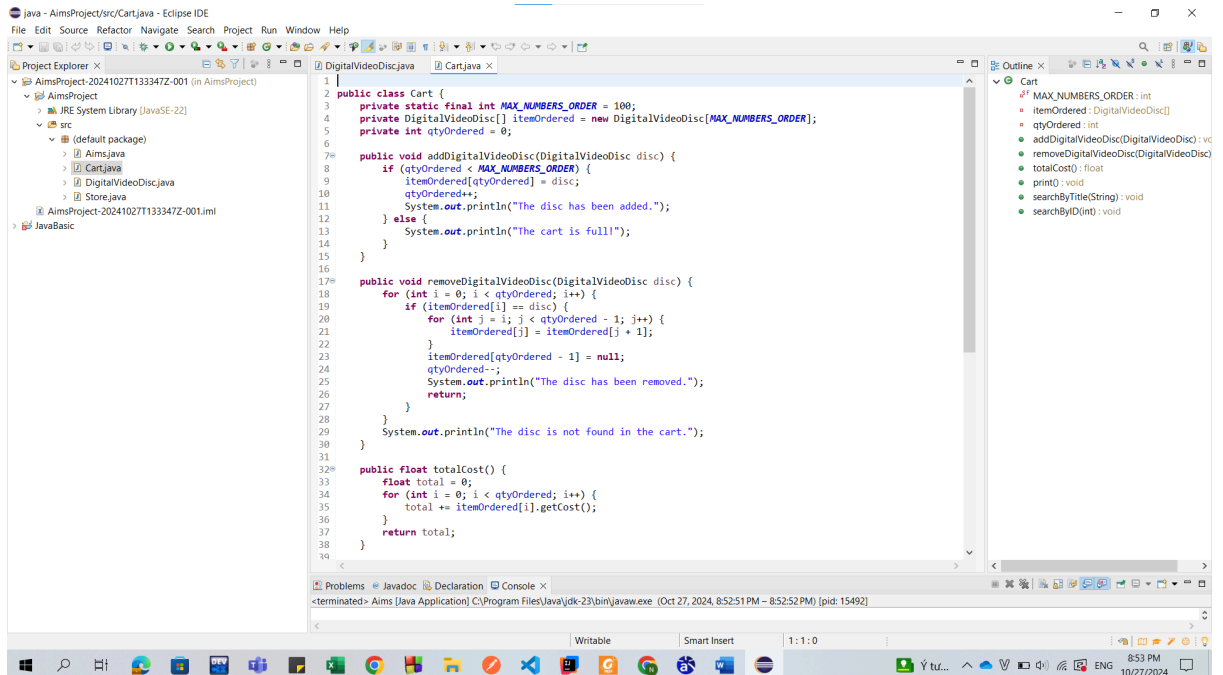
    public void searchByID(int id) {
        for (int i = 0; i < qtyOrdered; i++) {

```

```

        if (itemOrdered[i].getId() == id) {
            System.out.println("Found: " + itemOrdered[i].getTitle());
            return;
        }
    }
    System.out.println("No disc with ID: " + id);
}
}
}

```



8. Create Store Class

Code :

```

import java.util.LinkedList;

public class Store {

    private LinkedList<DigitalVideoDisc> itemsInStore;

    public Store() {
        itemsInStore = new LinkedList<>();
    }

    boolean checkDVD(DigitalVideoDisc disc) {
        return itemsInStore.contains(disc);
    }

    public void removeDVD(DigitalVideoDisc disc) {
        if (checkDVD(disc)) {
            itemsInStore.remove(disc);
            System.out.println("DVD removed from the store.");
        } else {
            System.out.println("DVD not found in the store.");
        }
    }
}

```

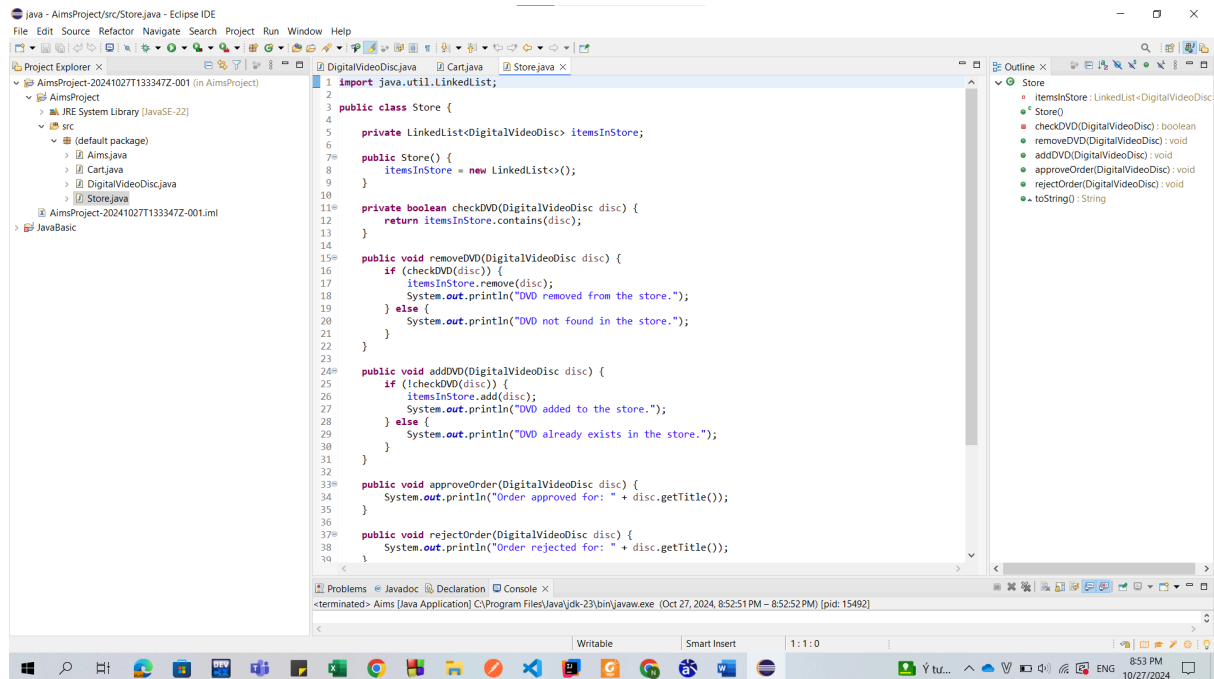


```

    public void addDVD(DigitalVideoDisc disc) {
        if (!checkDVD(disc)) {
            itemsInStore.add(disc);
            System.out.println("DVD added to the store.");
        } else {
            System.out.println("DVD already exists in the store.");
        }
    }

    @Override
    public String toString() {
        StringBuilder storeContents = new StringBuilder("Store contents:\n");
        for (DigitalVideoDisc disc : itemsInStore) {
            storeContents.append(disc.getTitle()).append("\n");
        }
        return storeContents.toString();
    }
}

```



9. Create Carts of DigitalVideoDiscs

Ta phải chạy chương trình thêm DVD vào giỏ hàng trong class Aims:

Code :

```

public class Aims {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        //Câu lệnh trong class Cart
        // Tao giỏ hàng mới
    }
}

```

```

        Cart anOrder = new Cart();

        // Tạo các đối tượng DVD mới và thêm chúng vào giỏ hàng
        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King",
"Animation", "Roger Allers", 87, 19.95f,5);
        anOrder.addDigitalVideoDisc(dvd1);

        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars",
"Science Fiction", "George Lucas", 87, 24.95f,4);
        anOrder.addDigitalVideoDisc(dvd2);

        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin",
"Animation","", 0, 18.99f,3);
        anOrder.addDigitalVideoDisc(dvd3);

        // In tổng chi phí của các mục trong giỏ hàng
        System.out.println("Total Cost is: ");
        System.out.println(anOrder.totalCost());

    }

}

```

Kết quả chạy chương trình :

The screenshot shows the Eclipse IDE with the following components:

- Project Explorer:** Shows the project structure with files like Aims.java, Cart.java, DigitalVideoDisc.java, and Store.java.
- Editor:** Displays the code for Aims.java, which includes the main method and comments in Vietnamese.
- Outline:** Shows the class structure with the main method.
- Console:** Displays the output of the program:


```

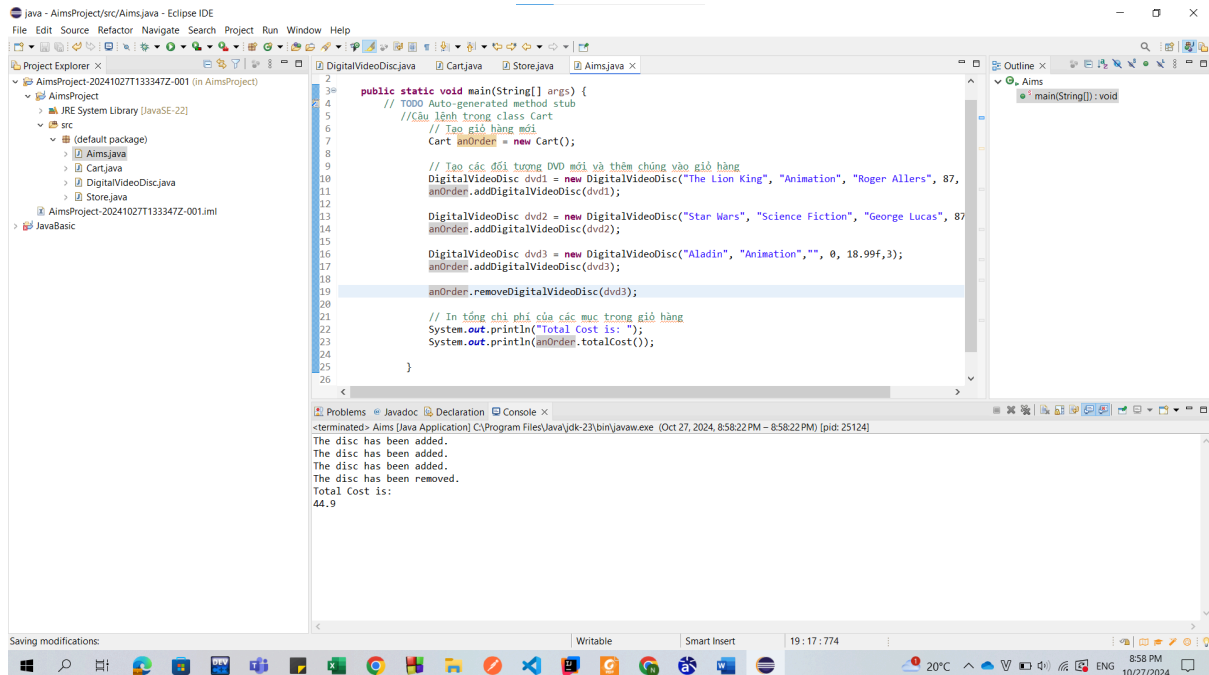
      <terminated> Aims [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Oct 27, 2024, 8:57:38 PM - 8:57:38 PM) [pid: 21012]
      The disc has been added.
      The disc has been added.
      The disc has been added.
      Total cost is:
      63.89
      
```

10. Requirements for cart

Đối với các yêu cầu , ta đều chạy trong Aims Class

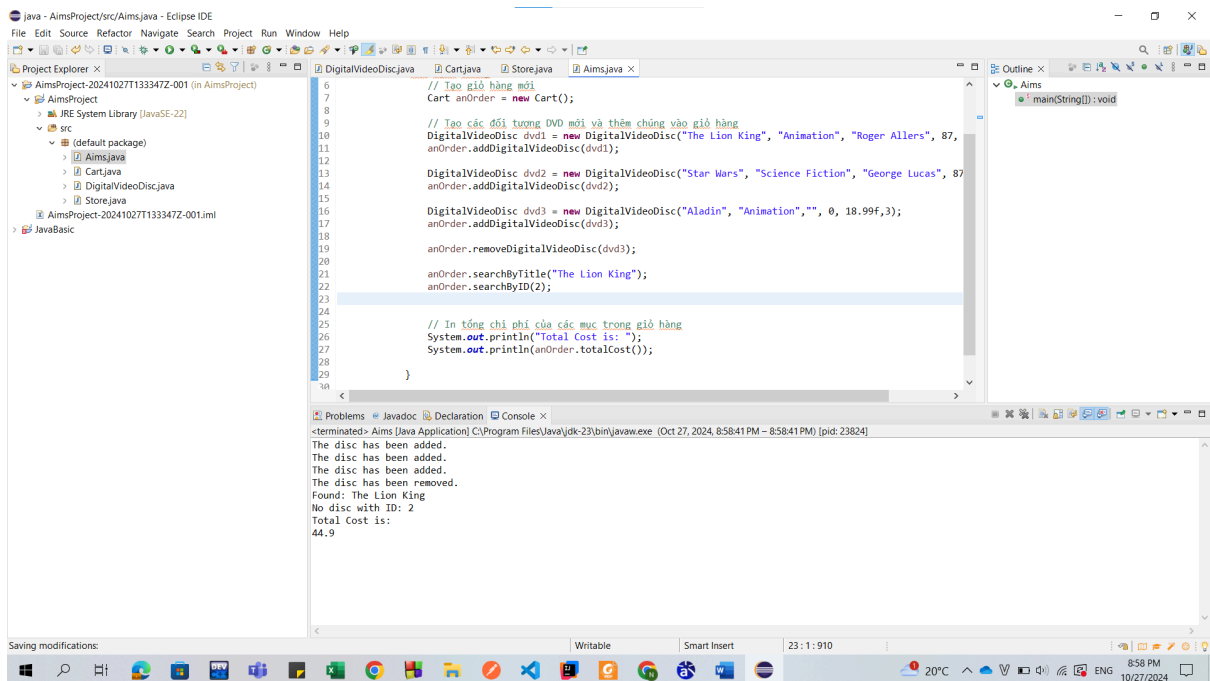
A, Remove DVD from Cart:

```
anOrder.removeDigitalVideoDisc(dvd3);
```



B. Search by title/ID

```
anOrder.searchByTitle("The Lion King");  
anOrder.searchByID(2);
```



C. Print title and cost DVDs in the Cart

```
anOrder.print();
```