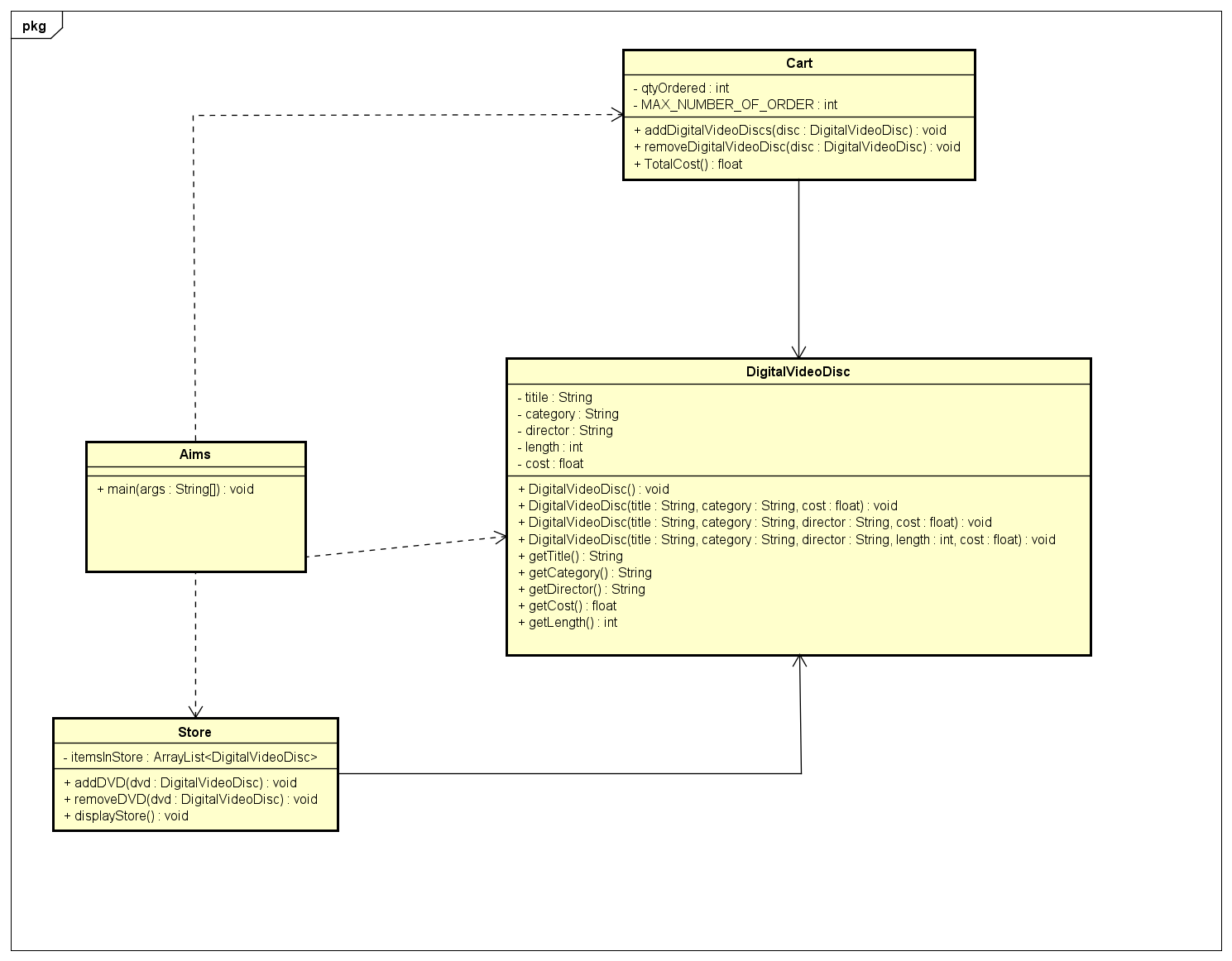
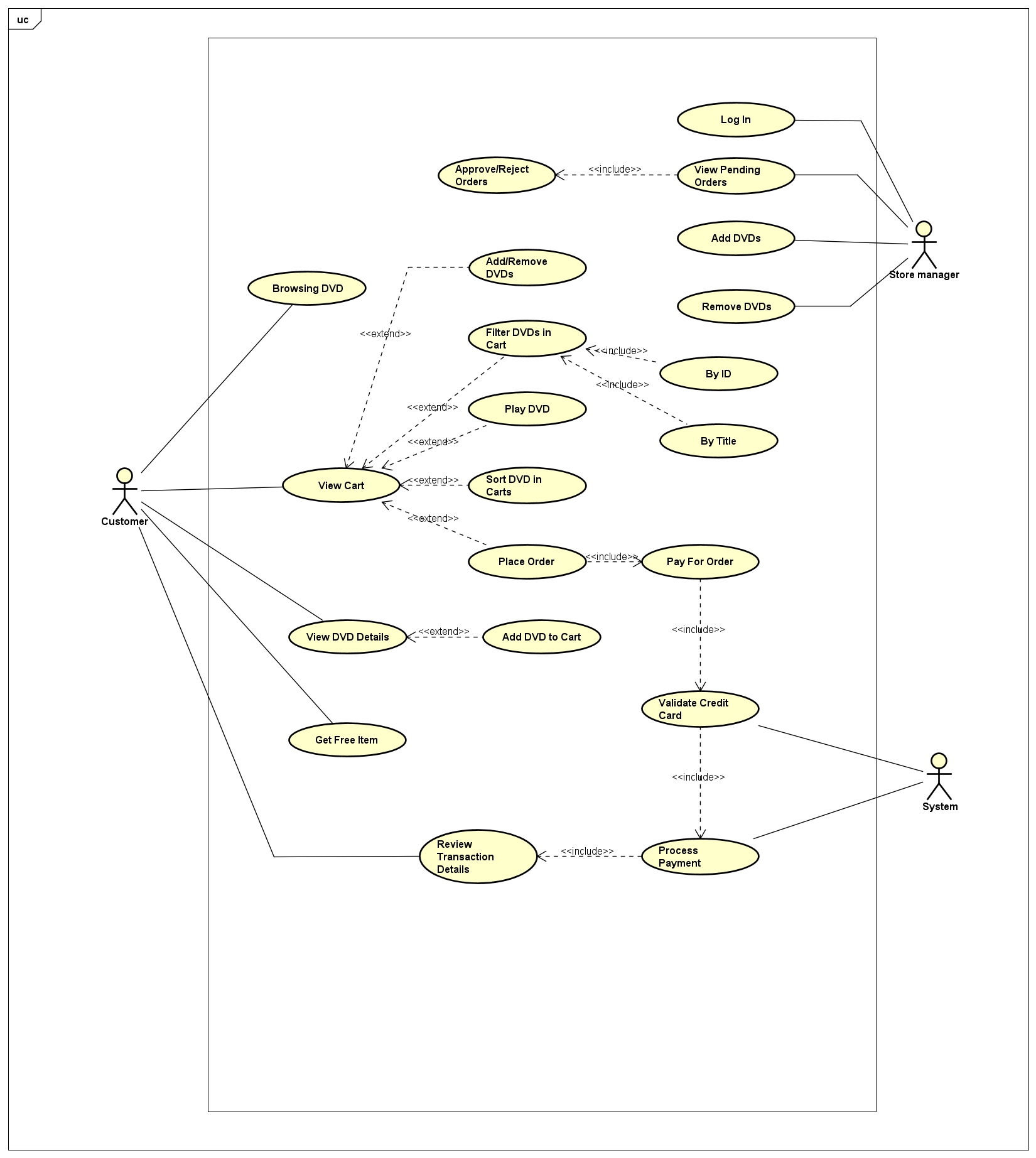
**LÊ NHẬT HOÀNG – 20235498**

**1. Update Usecase and Class Diagram**

****

**2. Method overloading**

**A screen shot of a computer program

Description automatically generated**

I would prefer array parameter as it is easier to handle because the number of DVDs is determined, otherwise it would be way too hard to handle.

**3. Passing parameter**

**A screen shot of a computer program

Description automatically generated**

***Is Java Pass by Value or Pass by Reference?***

Java is a pass-by-value programming language. This means that when you pass an argument to a method, Java creates a copy of the value and passes that to the method.

For objects, the value being passed is the reference to the object, not the object itself. If you try to swap two objects (e.g., swap(DVD dvd1, DVD dvd2)), the method receives copies of the references pointing to the objects. Therefore, reassigning these references (dvd1 = dvd2) only changes the local copies of the references, not the original references. This is why swapping objects in this way does not work.

***Why does the title of the objects remain unchanged after calling swap(jungleDVD, cinderellaDVD)?***

In the swap method, the parameters o1 and o2 are local variables. When the method is called, copies of the references jungleDVD and cinderellaDVD are passed to the method. Reassigning o1 and o2 within the method only changes the local copies of the references, not the actual references jungleDVD and cinderellaDVD in the calling context. As a result, the original objects (jungleDVD and cinderellaDVD) remain unaffected, and their titles stay the same.

***Why is the title of jungleDVD changed after calling changeTitle(jungleDVD, cinderellaDVD.getTitle())?***

In the changeTitle method, the reference to jungleDVD is passed to the method. This reference allows the method to directly access and modify the object's internal state. When the method calls dvd.setTitle(title), it changes the title of the object that jungleDVD references. Because jungleDVD and the method parameter dvd both point to the same object, any modifications made to the object inside the method are reflected in the original object. Hence, the title of jungleDVD is changed. However, when the method reassigns dvd to a new object (dvd = new DigitalVideoDisc(oldTitle)), this reassignment only affects the local copy of the reference, not the original jungleDVD. Thus, the original jungleDVD reference remains unchanged but retains the modified title.

**4. Debugging in Java**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

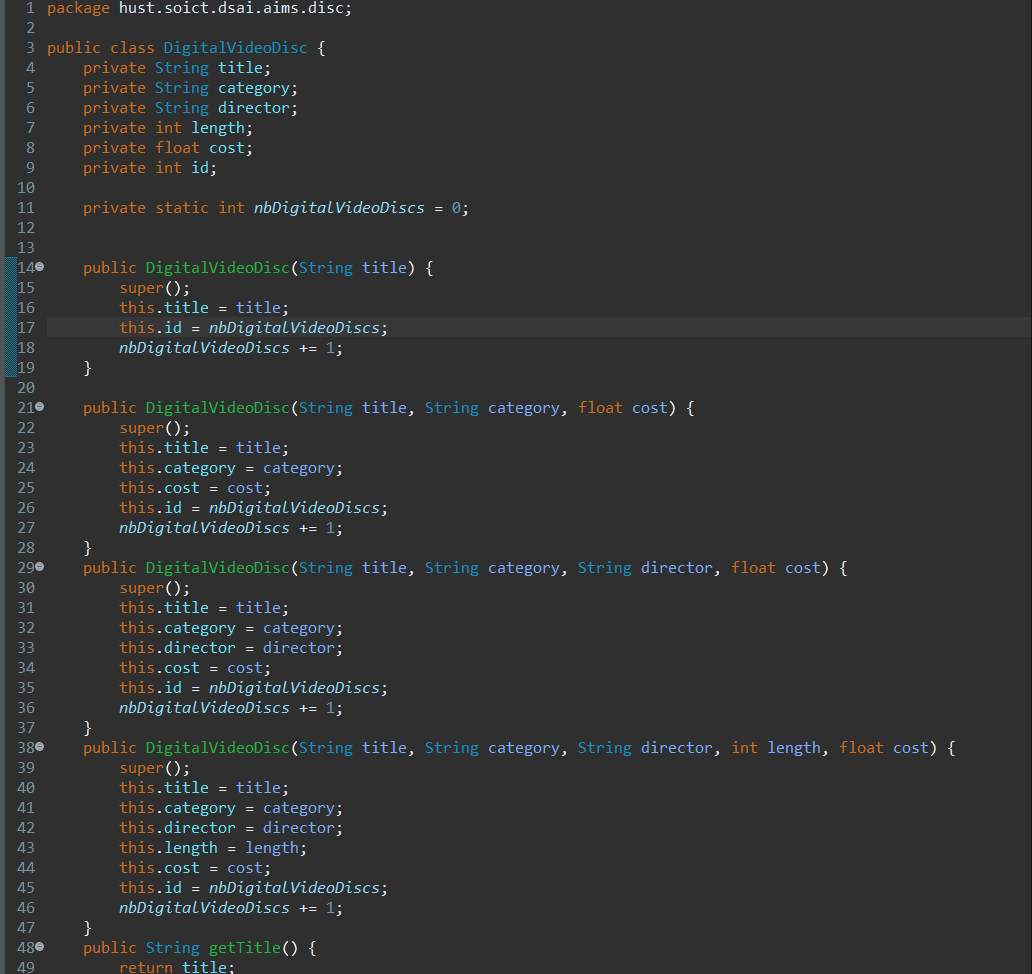
A screenshot of a computer program

Description automatically generated

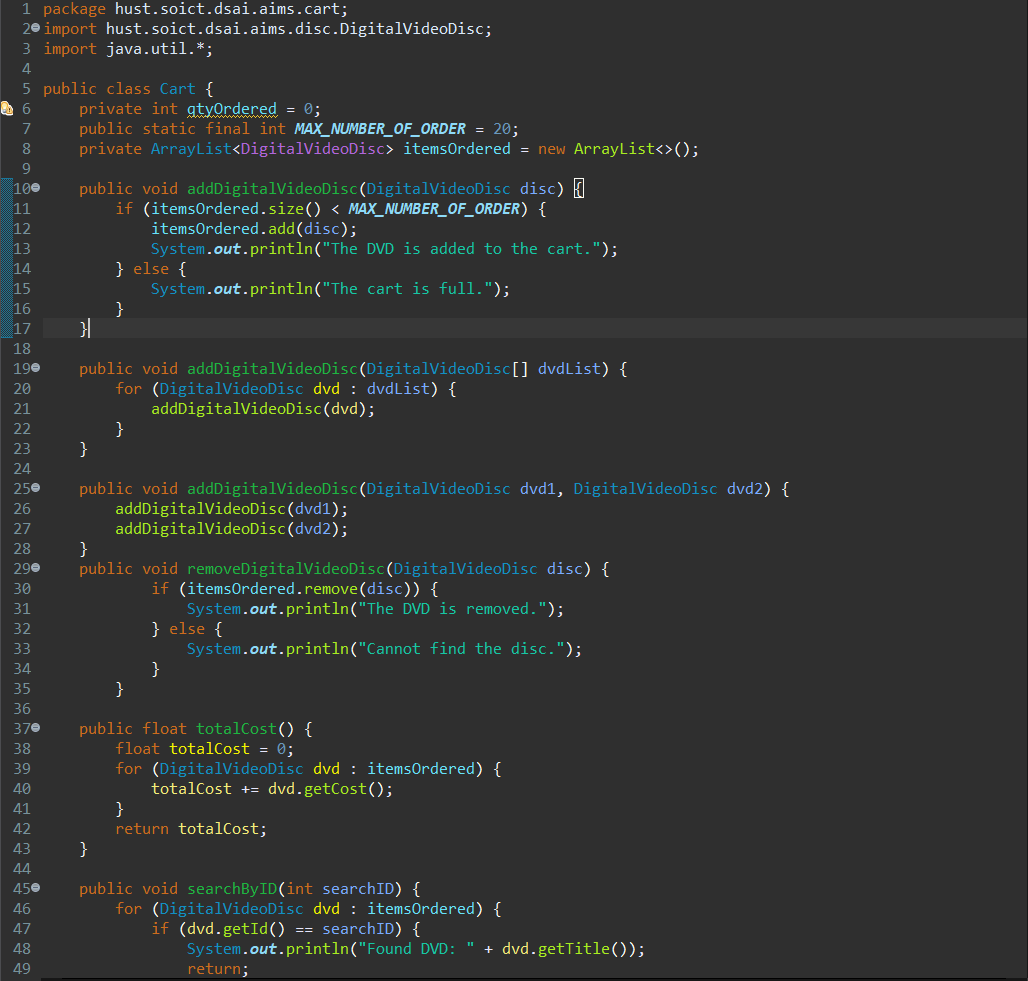
A screenshot of a computer program

Description automatically generatedAnd the result:

A screen shot of a computer code

Description automatically generated**5. Classifier Member and Instance Member**

**6. Open the Cart Class**



A screenshot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated

- Write a **toString()** method for the **DigitalVideoDisc** class. What should be the return type of this method? -> The method should return a String.

**7. Implement Store class**

A black rectangular object with white lines

Description automatically generatedA screen shot of a computer program

Description automatically generated

**9. String, StringBuilder and StringBuffer**

A screenshot of a computer

Description automatically generated**String Concatenation**

**A screenshot of a computer program

Description automatically generated**

Way too slow

**Using StringBuffer & StringBuilder:**

- A screenshot of a computer program

Description automatically generatedWe can see it’s way more faster and more efficient than string concatenation

- In most cases, StringBuilder is a preferred choice. In case there are problems relating to thread safety, we may consider between StringBuffer and StringBuilder