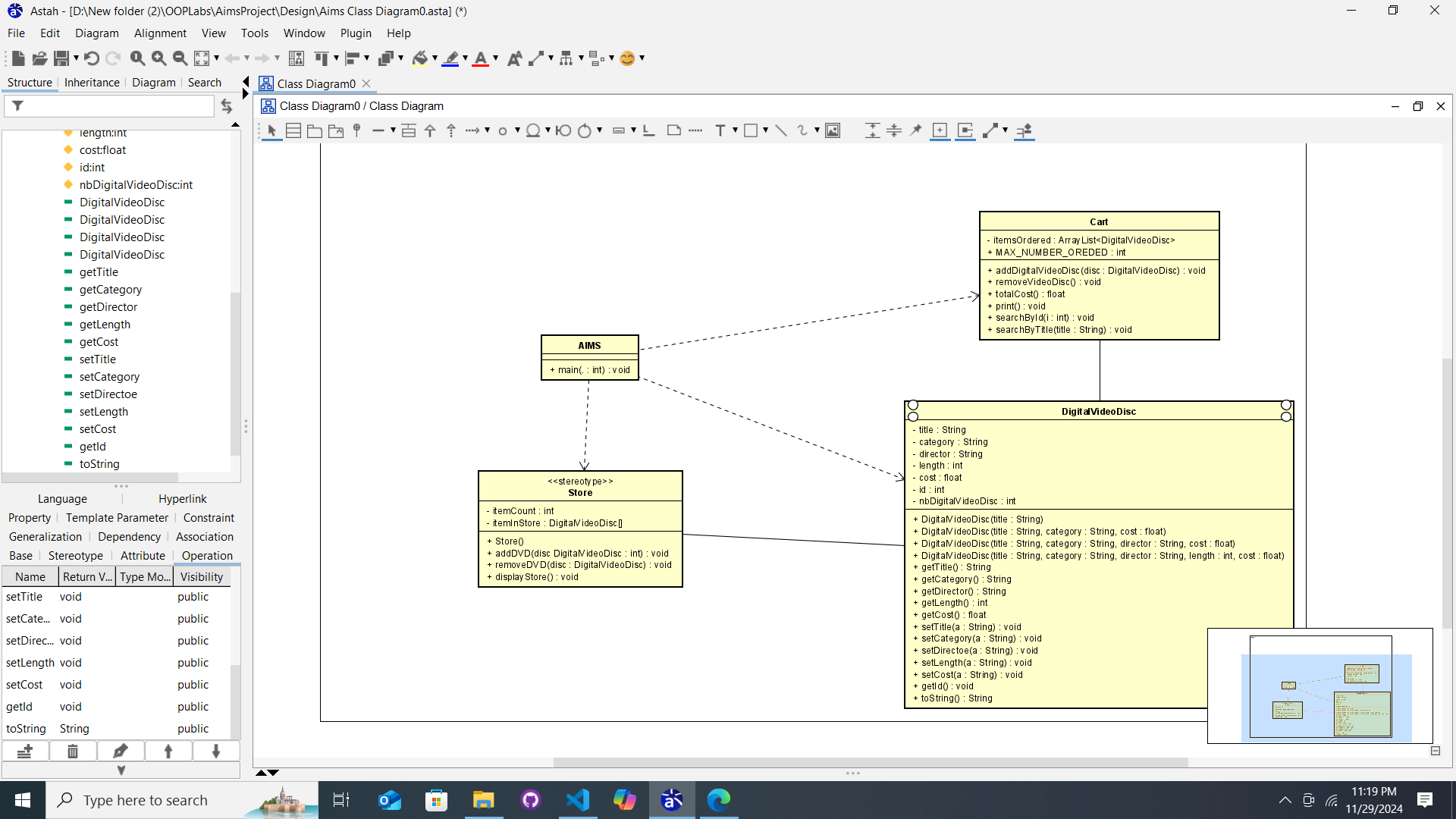
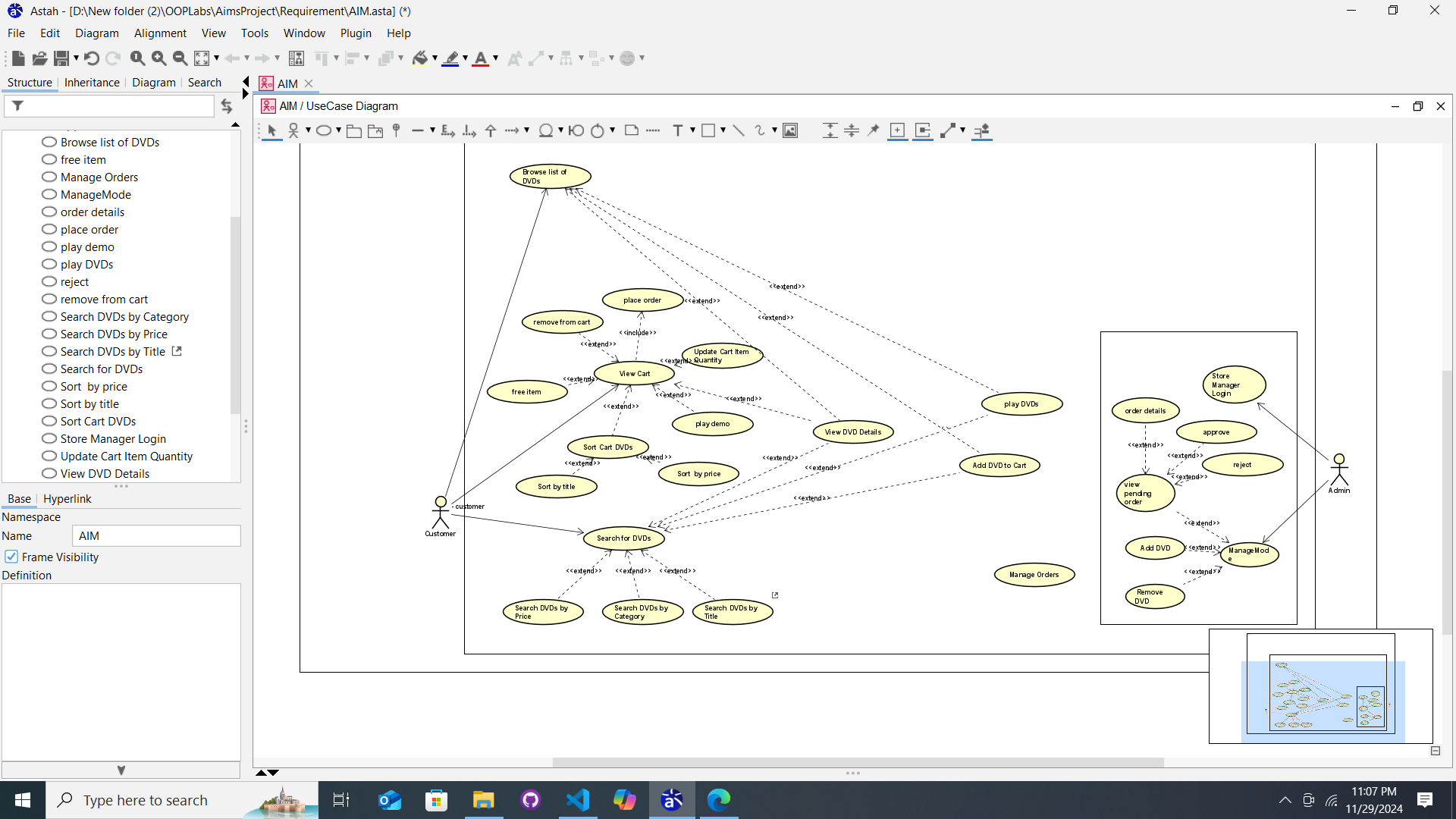
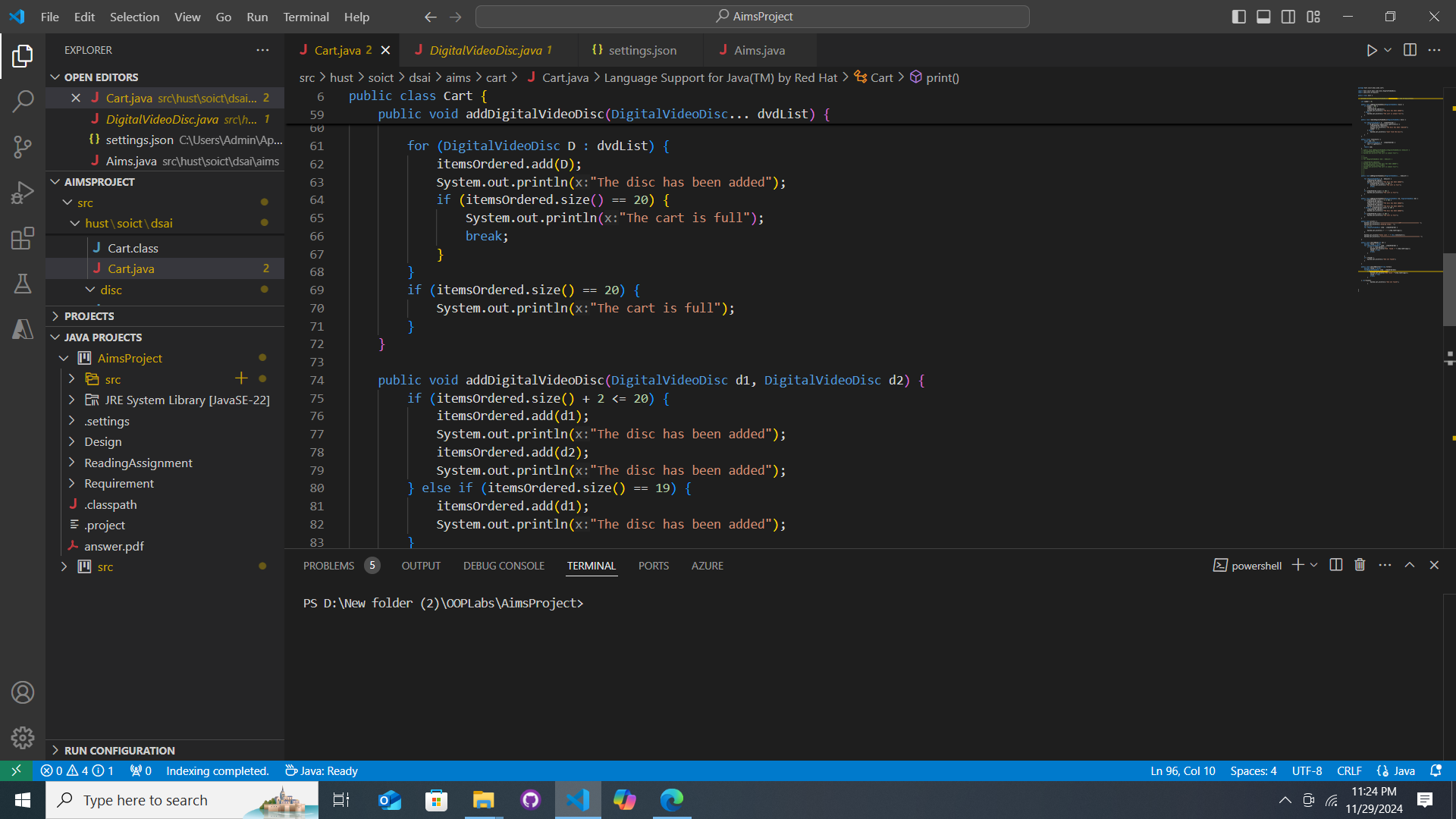
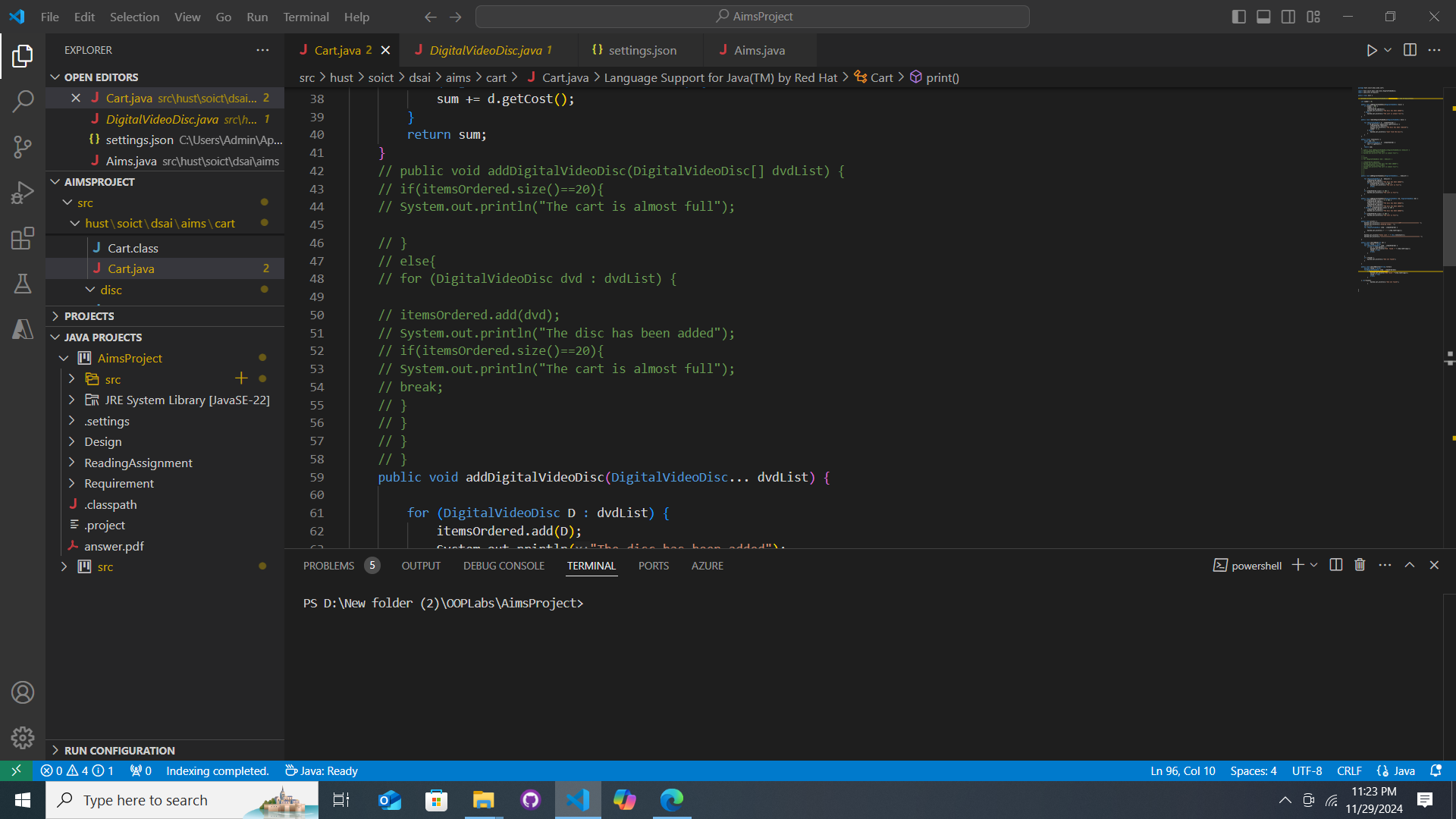
1. Update use-case diagram and class diagram

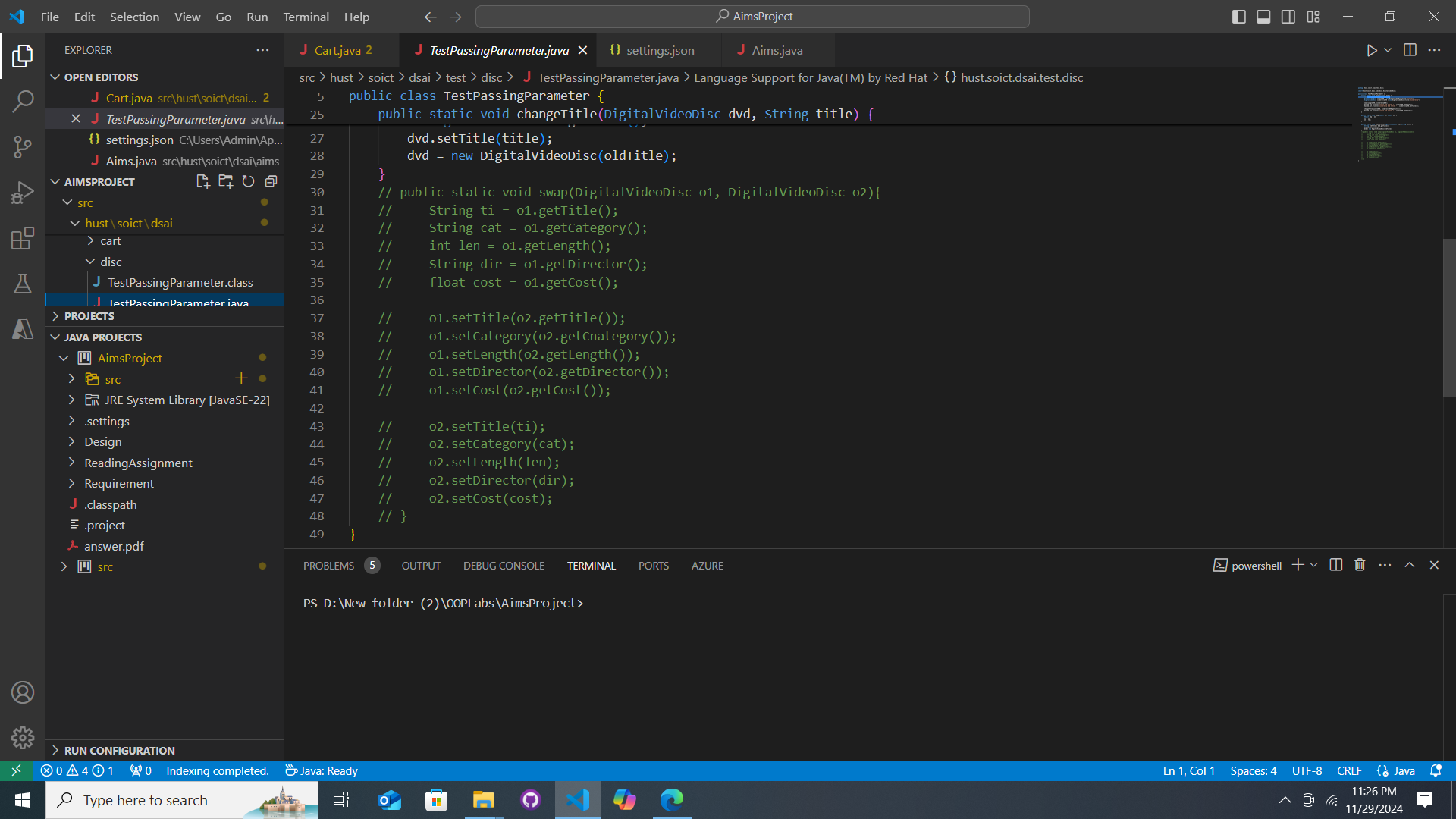
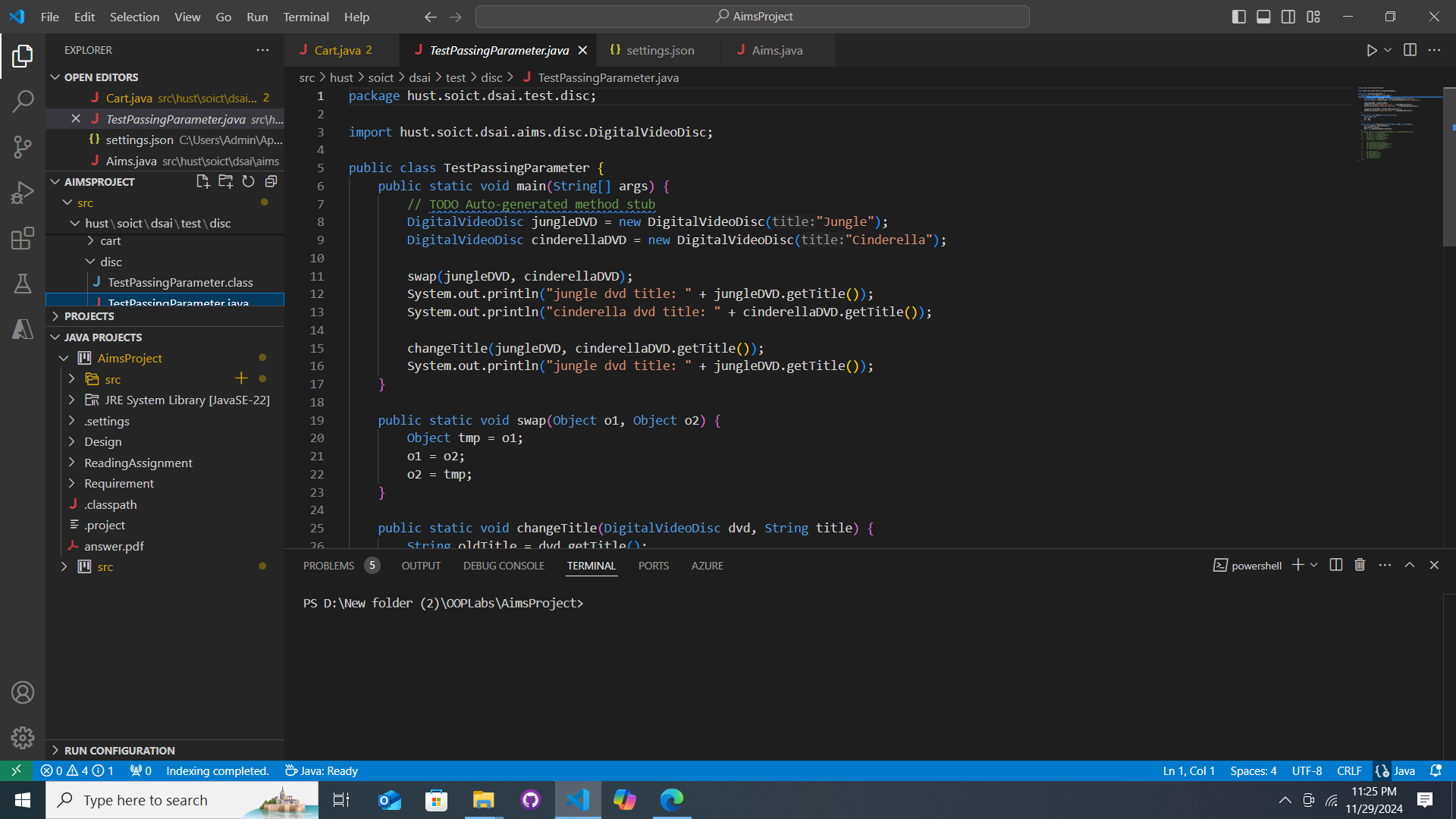


**2.Working with the method overloading**

**2.1**

-I prefer the first one because it is familiar to me.

**3.Passing parameter**



- Question: ***Is JAVA a Pass by Value or a Pass by Reference programming language?***

Answer : pass by value

**Questions:**

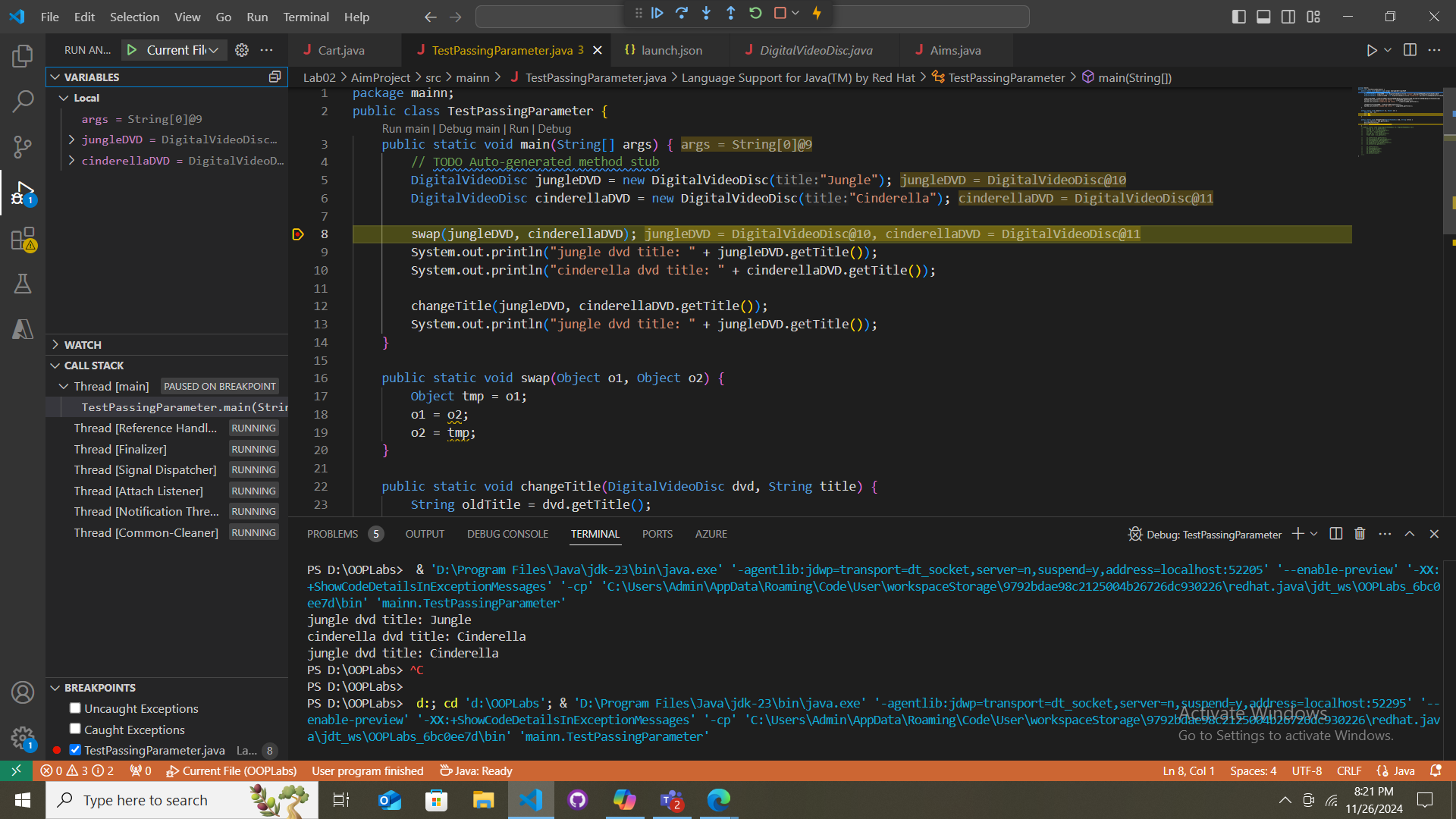
* After the call of **swap(jungleDVD, cinderellaDVD)** why does the title of these two objects still remain?

Answer: Java swaps local references within the method but does not affect the original references outside the method. Therefore, the titles remain unchanged.

* After the call of **changeTitle(jungleDVD, cinderellaDVD.getTitle())** why is the title of the JungleDVD changed?

Answer : The object reference points to the same memory location, so changes made to the object inside the method are visible outside the method.

**4.Use debug run**

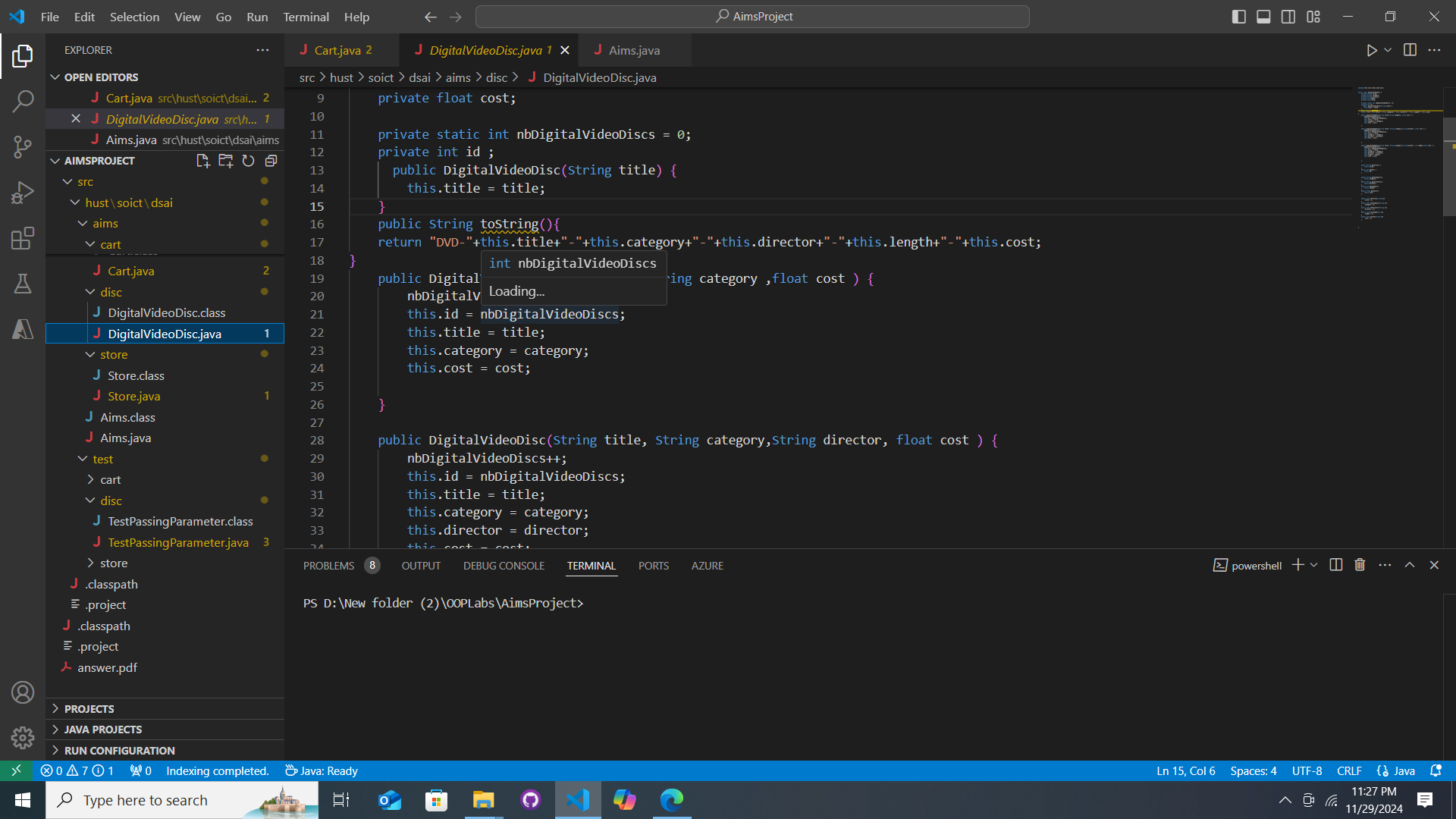


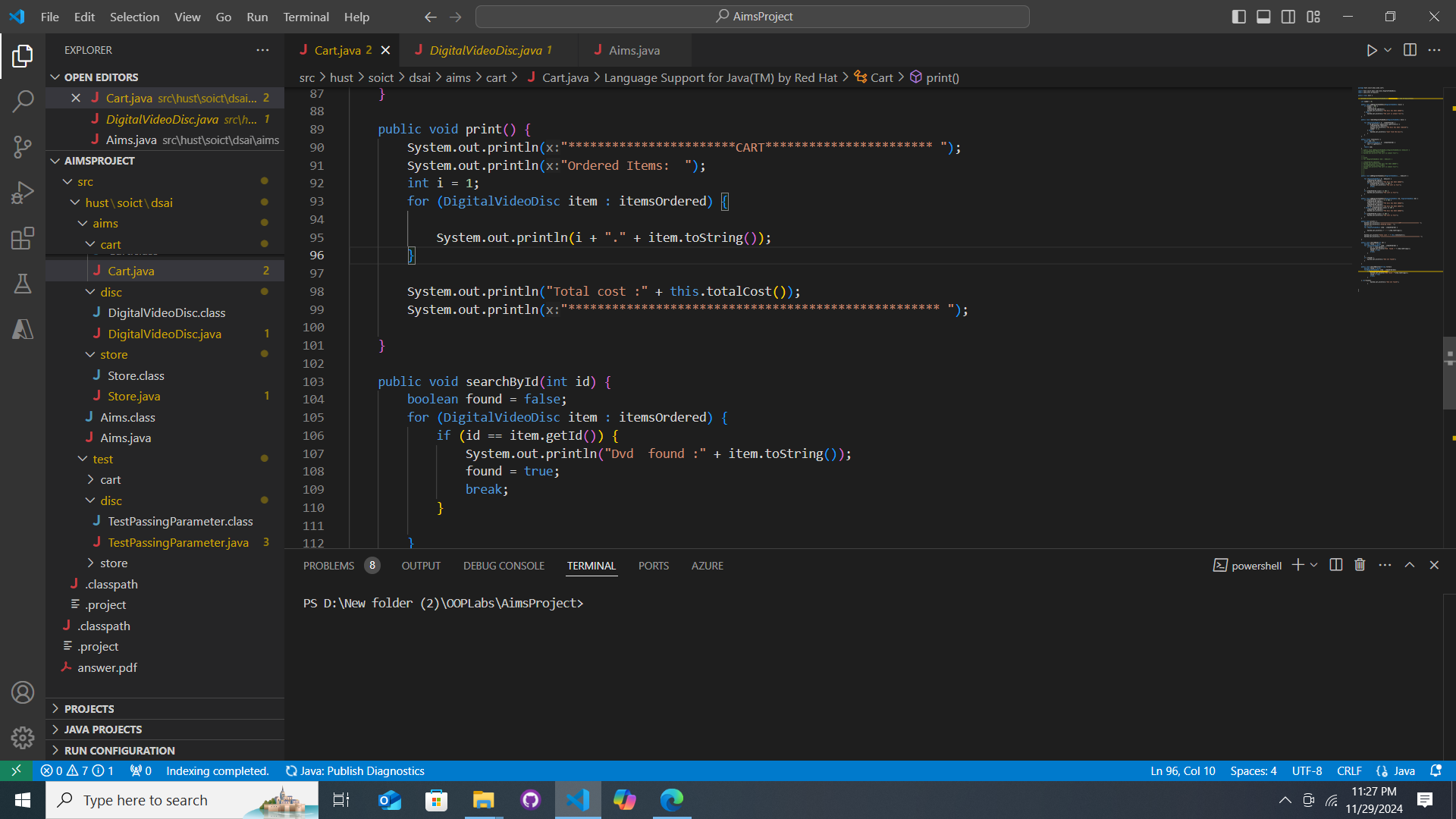
# 5.Classifier Member and Instance Member

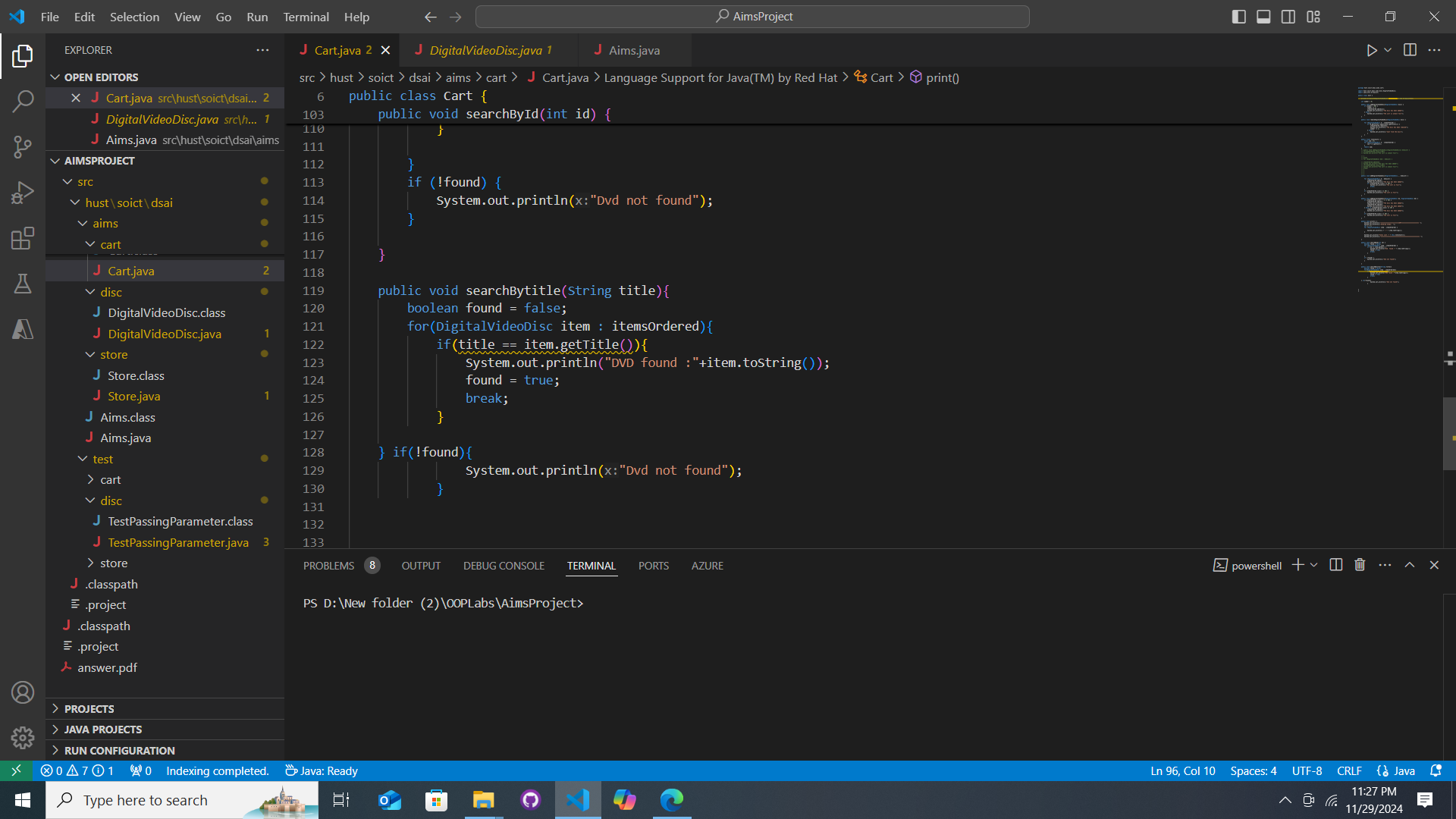
# 6. Open the **Cart** class

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

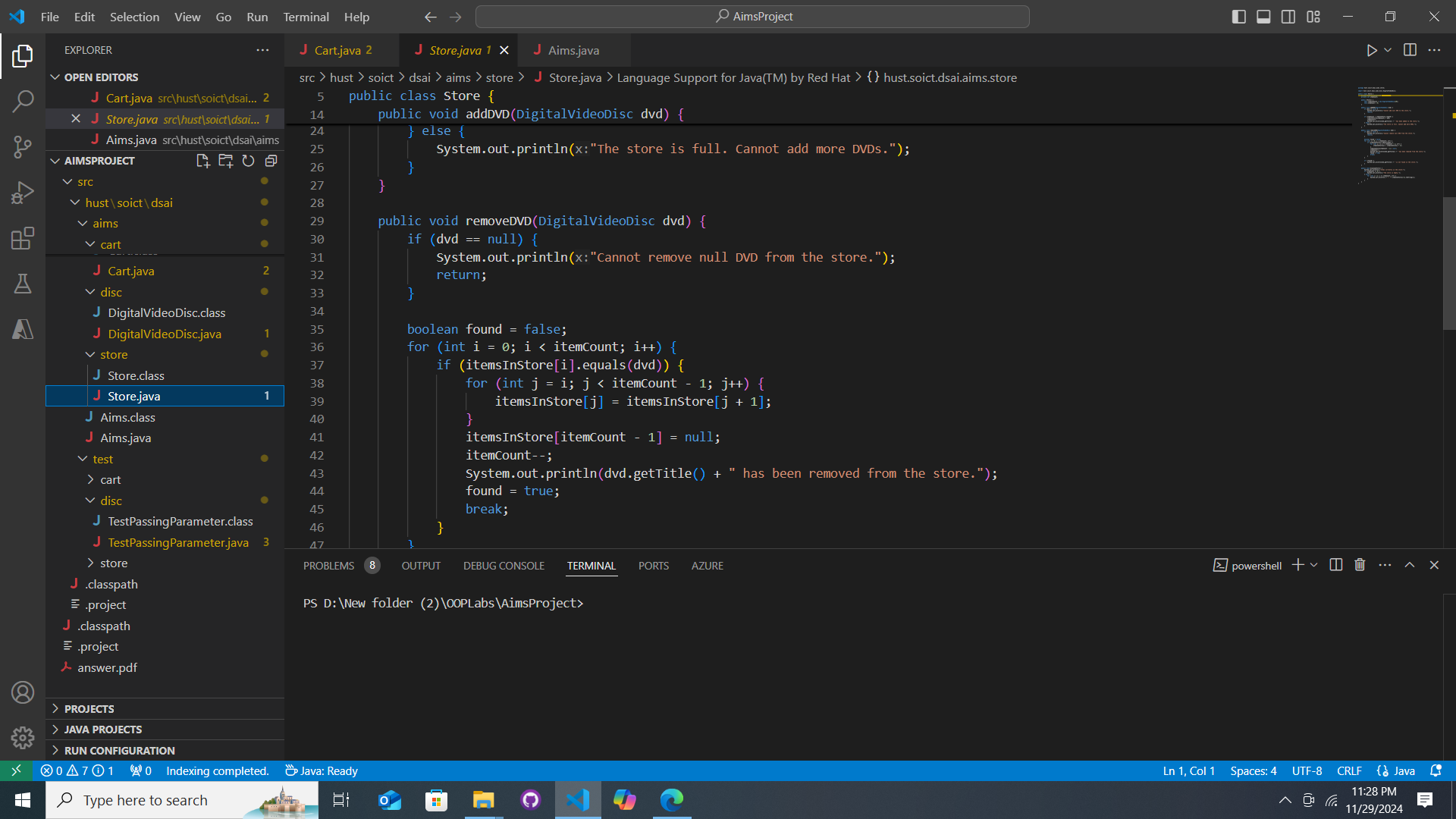
Answer : String







# 7. Implement the **Store** class



## 9. *String, StringBuilder and StringBuffer*

