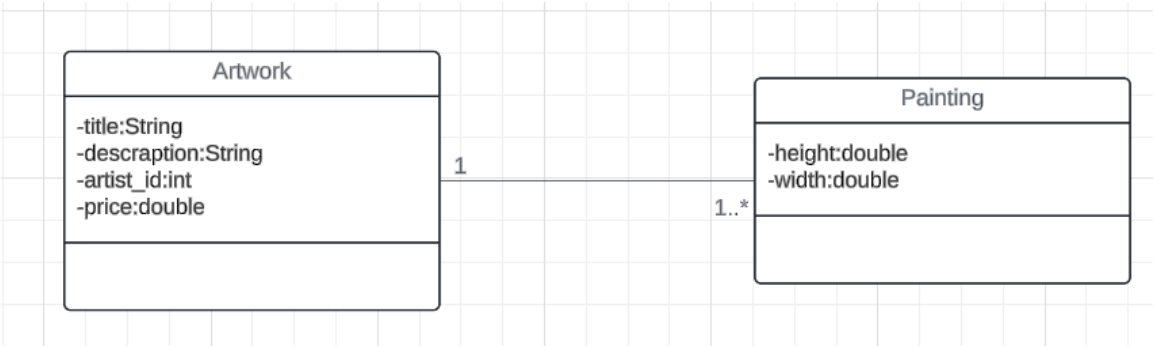
Three Design Patterns Applied :

1-Structural: Abstraction occurrence pattern :

Problem: What is the best way to represent such sets of occurrences?

Solution:

- Create an “abstraction” class that contains the common data.
- Then create an “occurrence” class representing the occurrences of this abstraction.
- Connect these classes with a one-to-many association

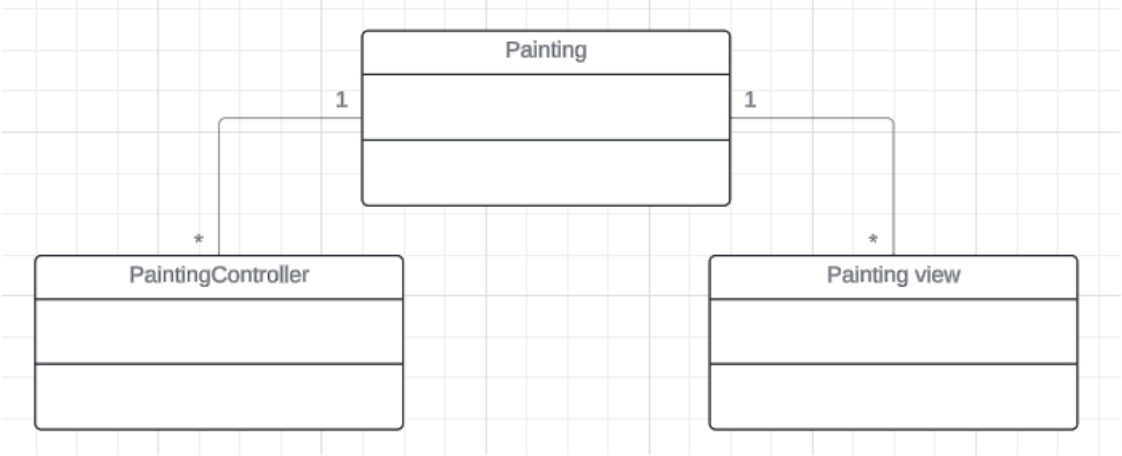


2-Behavioural: Observer Pattern :

Problem: A mechanism is needed to ensure that when the state of an object changes related objects are updated to keep them in step

Solution:

• One object has the role of the subject/publisher and one or more other objects the role of observers/subscribers. The observers register themselves with the subject, & if the state of the subject changes the observers are notified & can the update themselves.



3-Creational: Singleton Pattern :

Problem: How do you ensure that it is never possible to create more than one instance of a singleton class. And provide a global point of access to it.

Solution:

- Have the constructor private to ensure that no other class will be able to create an instance of the class singleton.