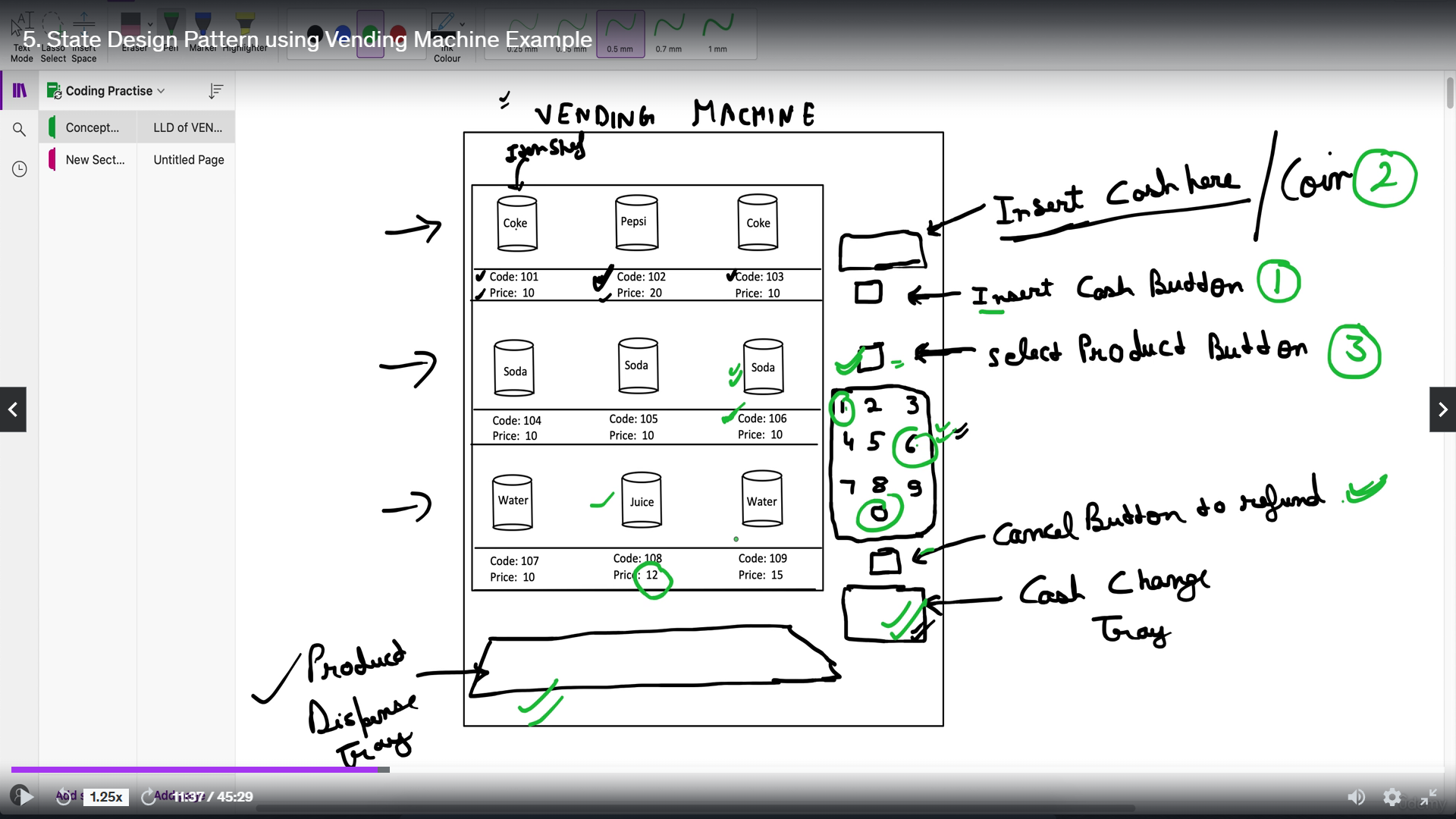
**State Design Pattern**1. In the interview, If you’re told that a product has specific states with fixed operations, then we use this pattern.   
2. The State Design Pattern is particularly useful in situations where an object needs to change its behavior when its internal state changes.   
3. This pattern helps to manage state-specific behavior in a clean, modular, and maintainable way.

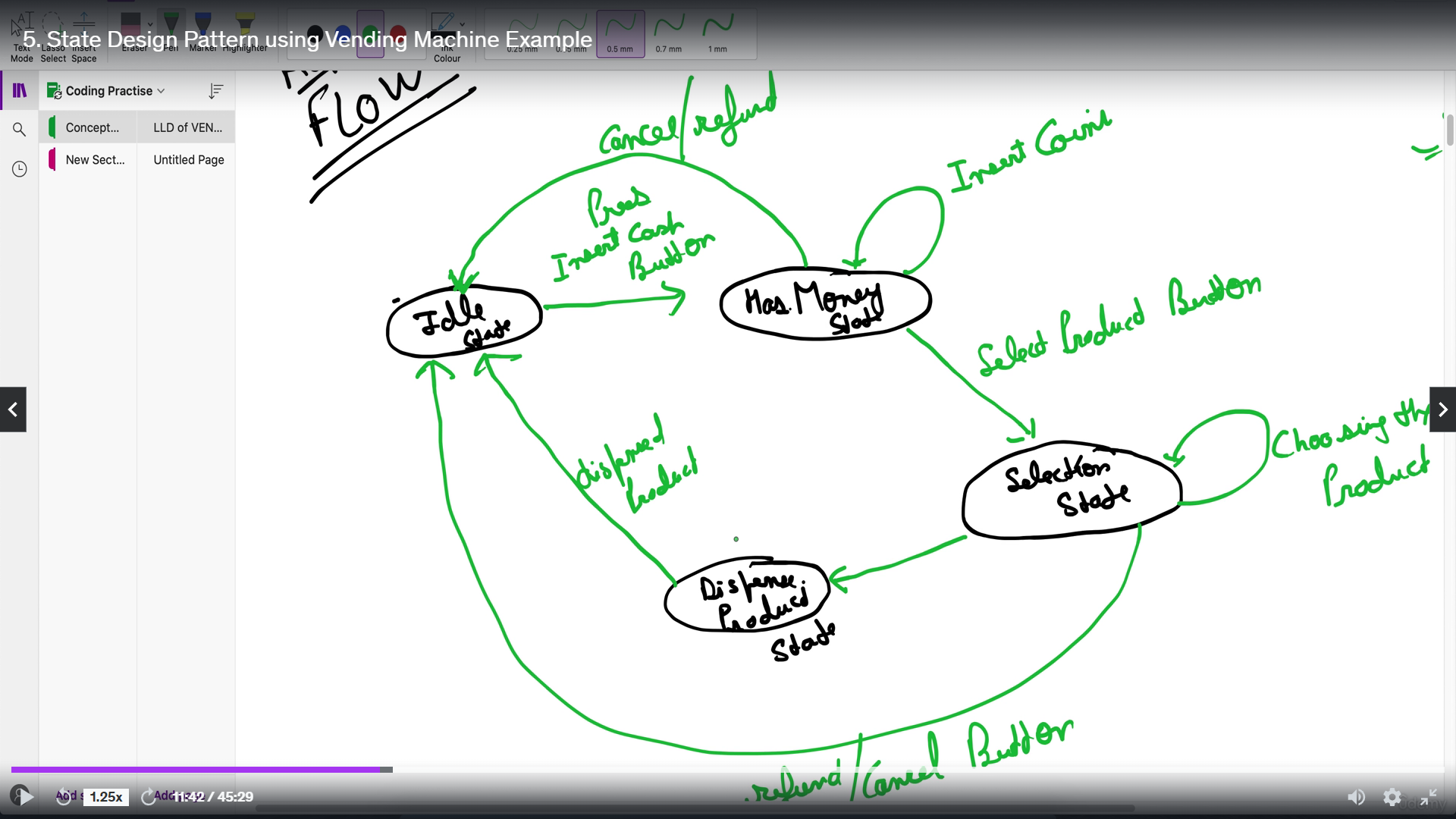
Follows Single responsibility & Open/Close principles.

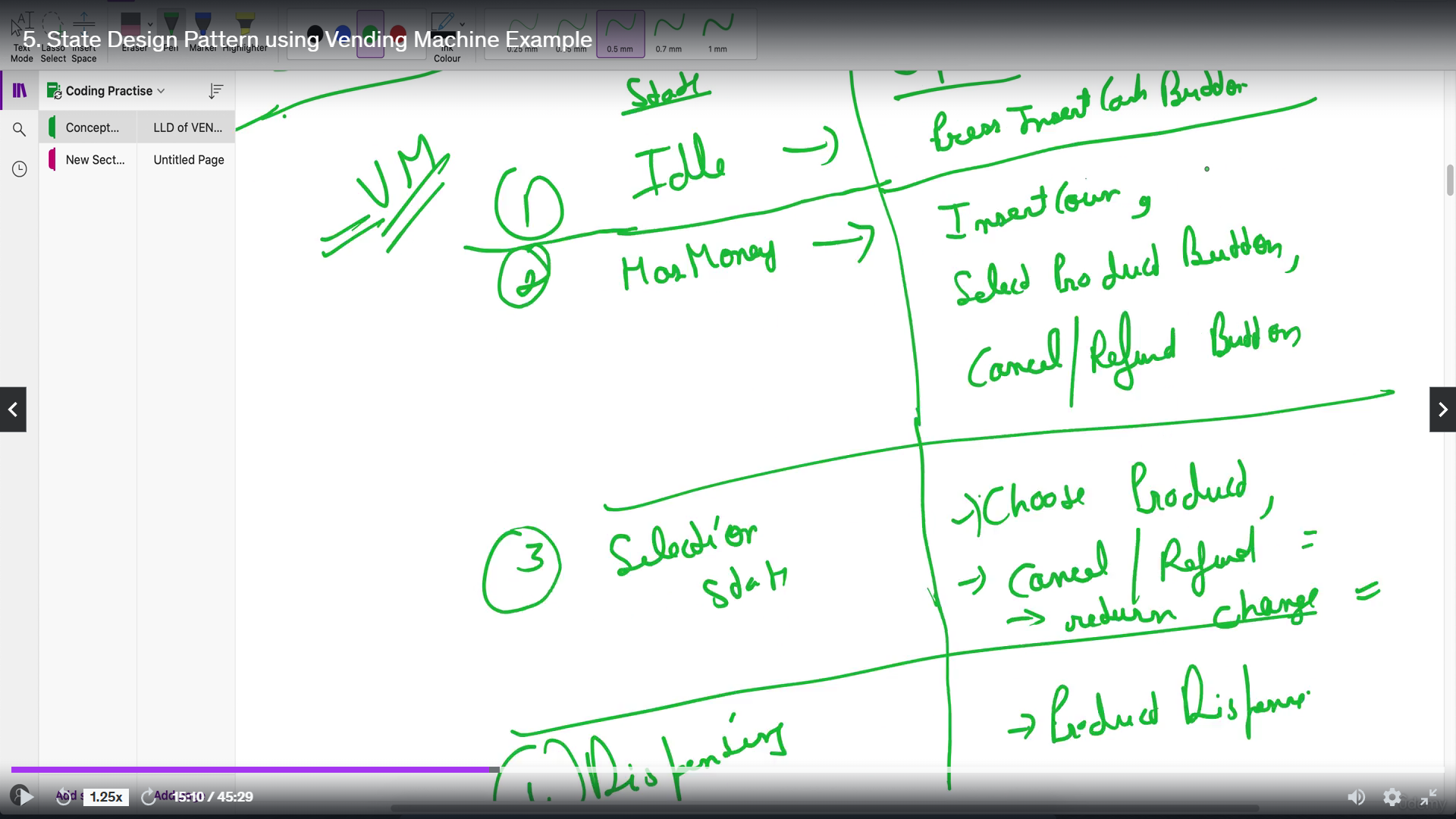
Eg use cases: Vending machine, TV, ATM, Traffic Light, Order processing, Elevator, Customer support ticket, Audio player, Task management, Parking lot, etc.

Eg2: In a game, a character might have different states like Idle, Running, Jumping, and Attacking. Each state has specific behaviour and transitions.  
 **Given product – Discuss the design and operations with the interviewer**



**Product Operations flow**



**Product State and Operation chart**

NOTE: In this pattern, we make a state interface containing all the states and operations. Each concrete state class has all the operations.   
Eg: We have states A,B,C,D and operations P,Q,R,S. Suppose A can perform P and R only, so here if Q or S are performed for A, it will throw a default exception/error.

