Lecture 32:- Vendor machine Build (state Design pattern).

Introduction

Suppose there is an object which can exist in limited no. of forms states at a time.

State machine diagram.

objects.	methods
A	$m1() \Rightarrow A \to B$
B	$m2() \Rightarrow B \to C$
C	$m3() \Rightarrow C \to D$
D	$m4() \Rightarrow D \to A$

$$\begin{tikzcd} A \arrow[r, "m1()"] & B \arrow[d, "m2()"] \\ D \arrow[u, "m4()"] & C \arrow[l, "m3()"] \end{tikzcd}$$

When to Use state Design pattern:-

When object changes it's states after particular operation / method, we can use state design pattern.

Example:- Building vendong machine

vendor machine working:- we will enter item needed through keypad then vm give us that item in dispenser.

States in vendor machines:-

1. No coin state - when No coin inserted.
2. HasCoin state - when coin is inserted.
3. Dispense state - item is being dispensed.
4. Soldout state - item is sold | finished.
5. Return coin state - vm return coin when item is unavailable.

① scenarios when object doesn't change state when object does not perform that operation on that state.

ex:- At Has coin state $\longrightarrow$ disepense().

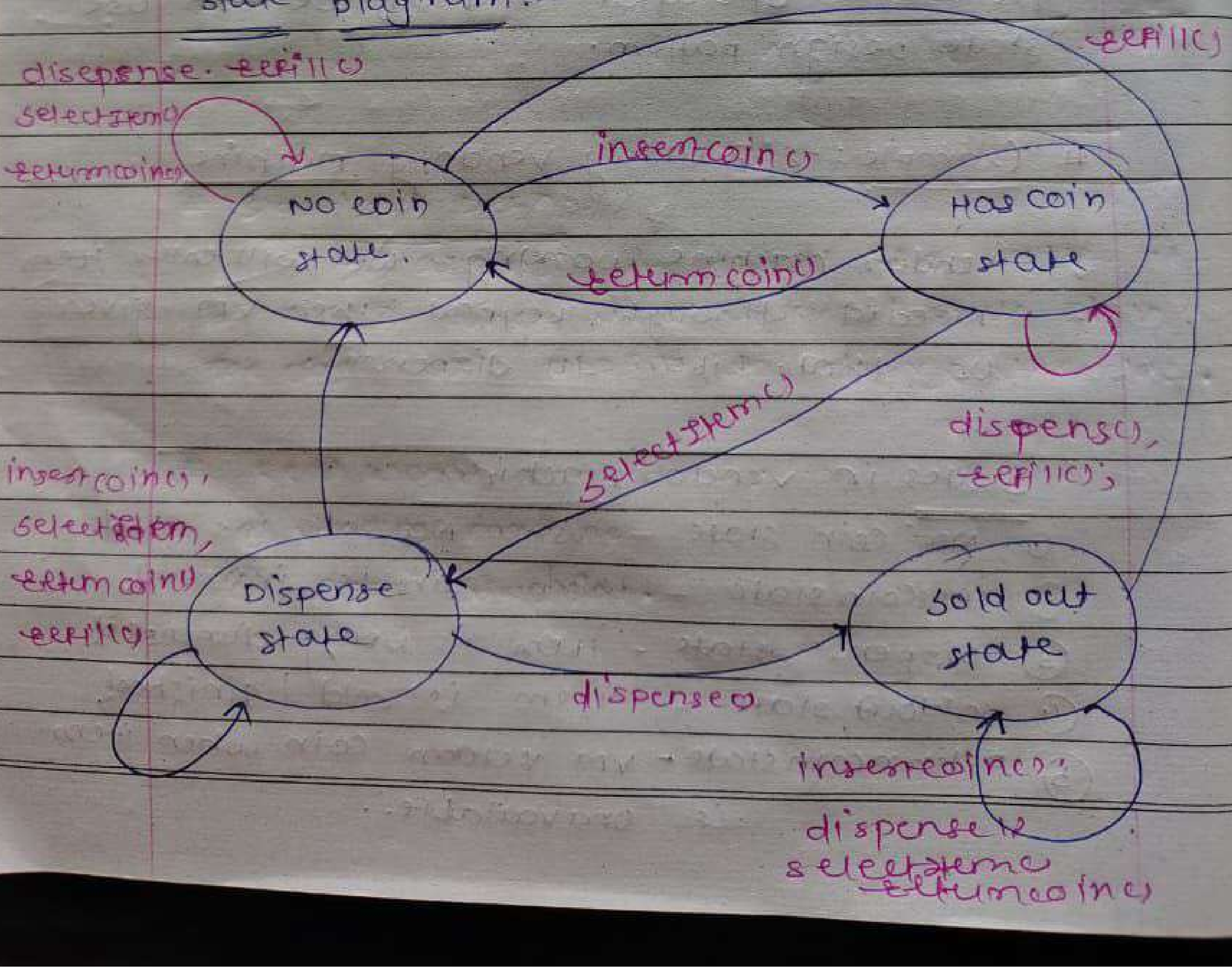
② when object perform operation but result of that operation remain vm in same state.

ex:- refill() $\rightarrow$ Nocoin state.

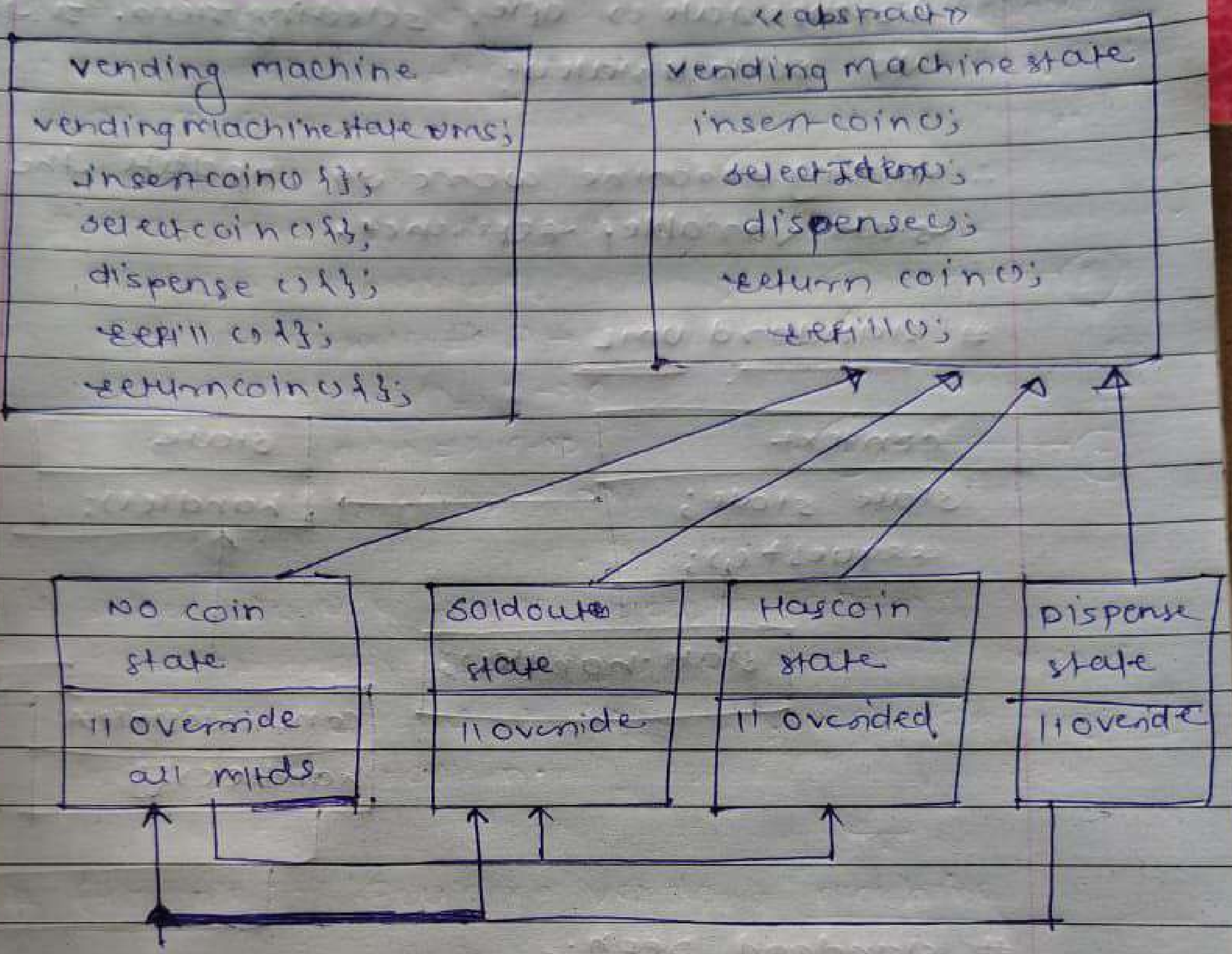
ex: insertcoin() $\longrightarrow$ Hascoin state()

It will accept $\overset{\text{coin}}{}$ but does not change state as item is not selected to get item dispensed.

State Diagram:-



UML diagram:-



Now main-class:- vending machine which comes to delegate tasks then it's ~~[illegible]~~ known as context class.

How mtd works?

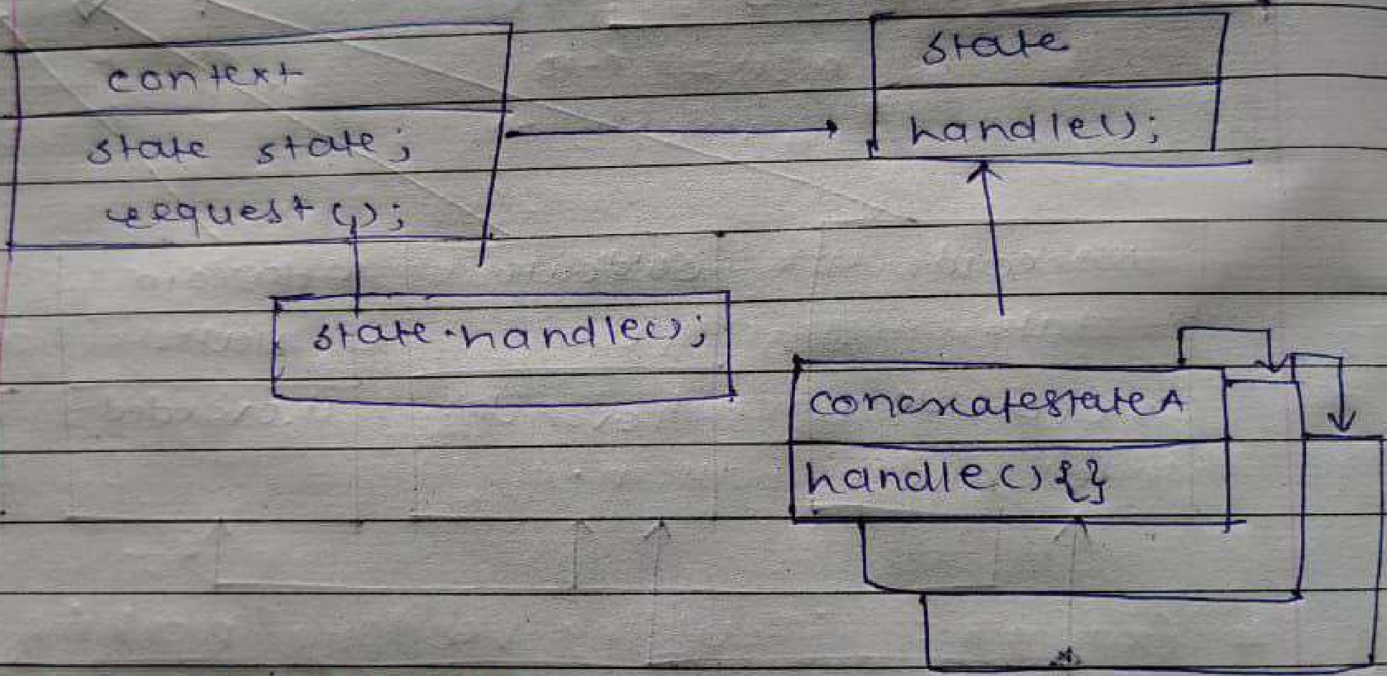
```
- insertcoin()
  { state = state → insertcoin();
  }
```

Assume current state is Nocoinstate then insertcoin() call to hascoinstate();

∴ Now state changes from hascoinstate() to dispensestate () after selecting item. It leads to soldout state().

∴ vending machine work to update the state variable, reference.

Standard UML



Standard Defn :-

It allows an object to alter its behaviour when it's internal state changes The object will appear to change the class.

Real world use cases :-

① vending machine
② ATM machine.