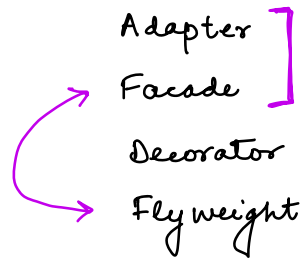
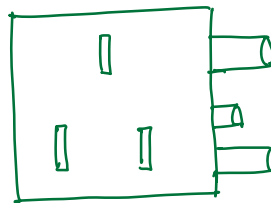
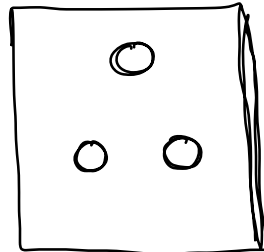
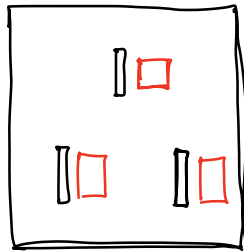


## Structural design



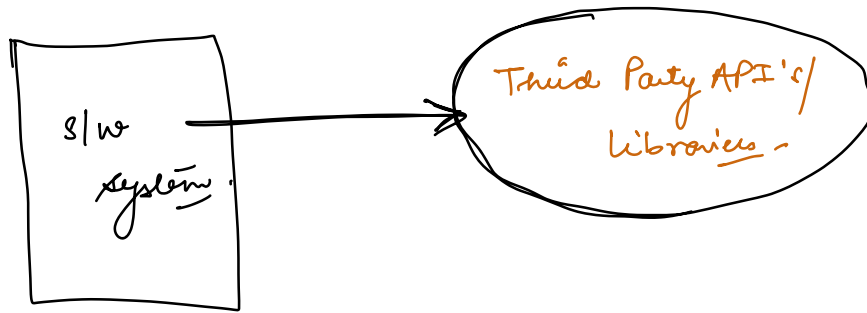
### Adapter Design Pattern

↓  
power adapter.



Adapter.

↓  
intermediate layer which  
converts one form to  
another.



1) change  $III^{rd}$  party API

2) add  $III^{rd}$  party API

3) update

razorpay  
↓  
phonepe -

x() → y()

Phonepe  
{ YesBankAPI yb = new  
          YesBankAPI();

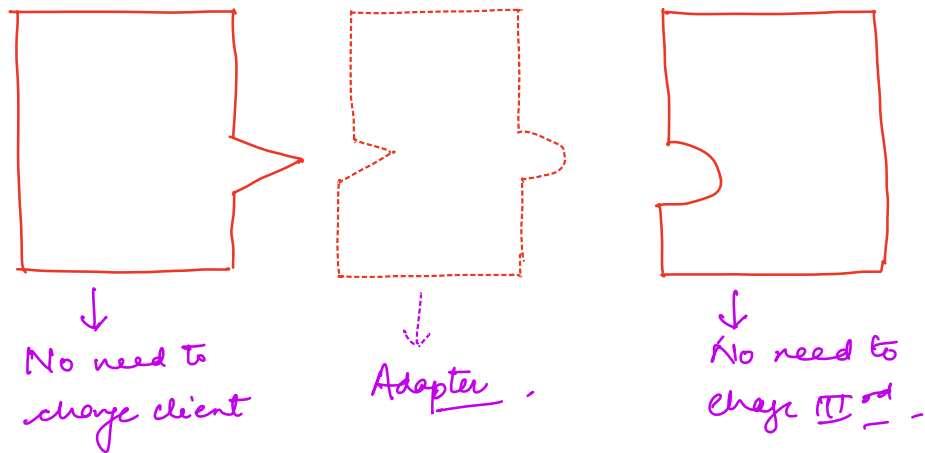
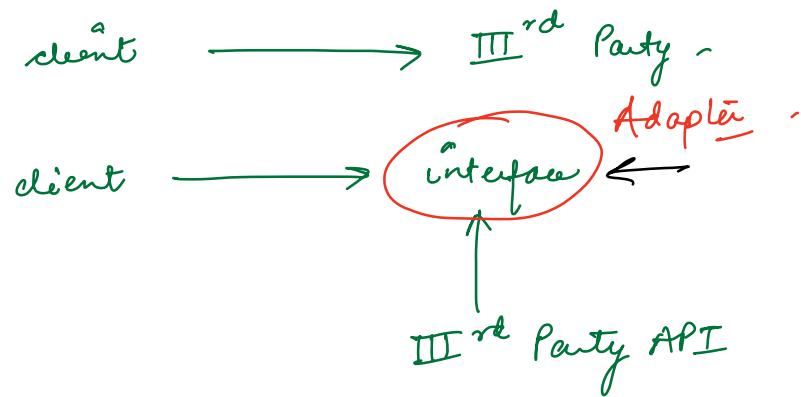
yb.getBalance();  
yb.moneyTransfer();

X  
YesBankAPI.

✓  
ICICI BankAPI.  
- balance@cheek() ↓  
- transfer() ↓

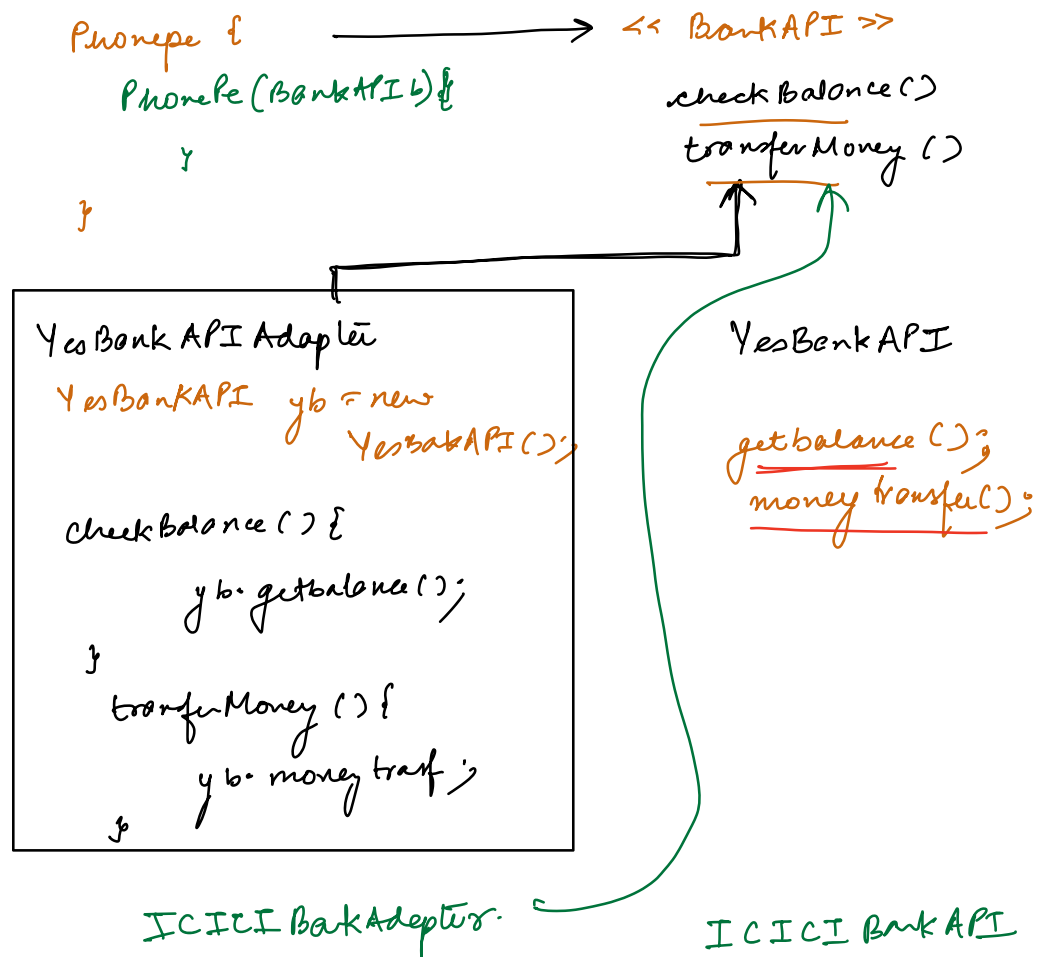
# 2 concrete classes are dependent on each other. → DI is violated.

# use interfaces



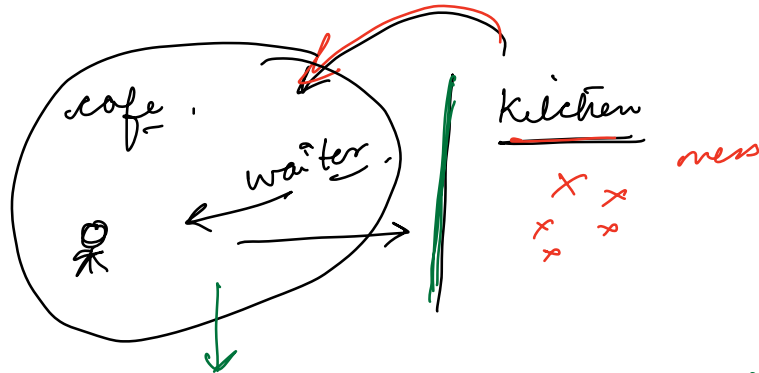
## How to use Adapter ?

- ① create an interface acc to how your client wants it to be.



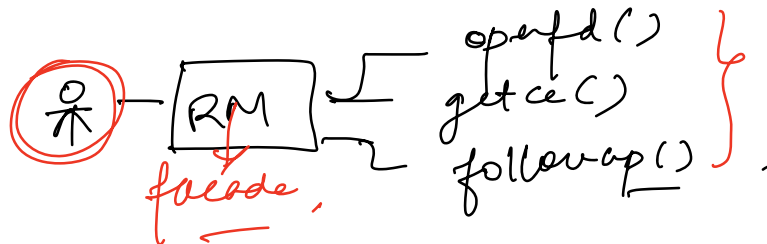
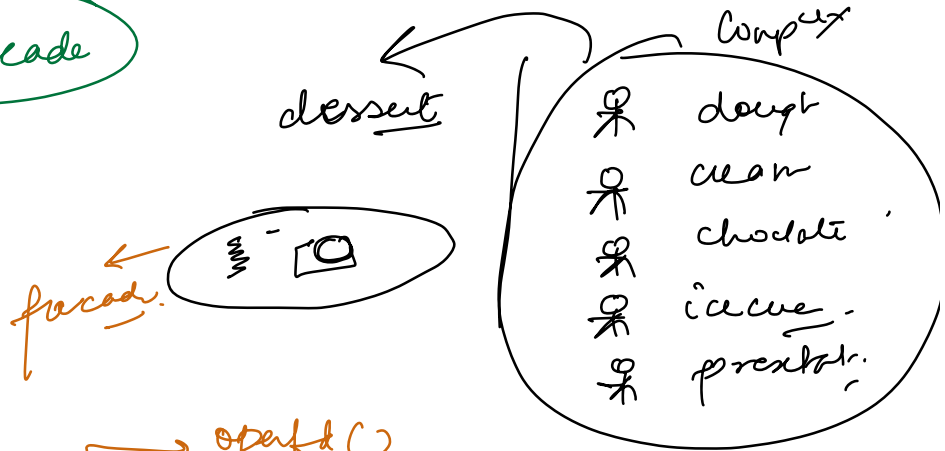
# Facade

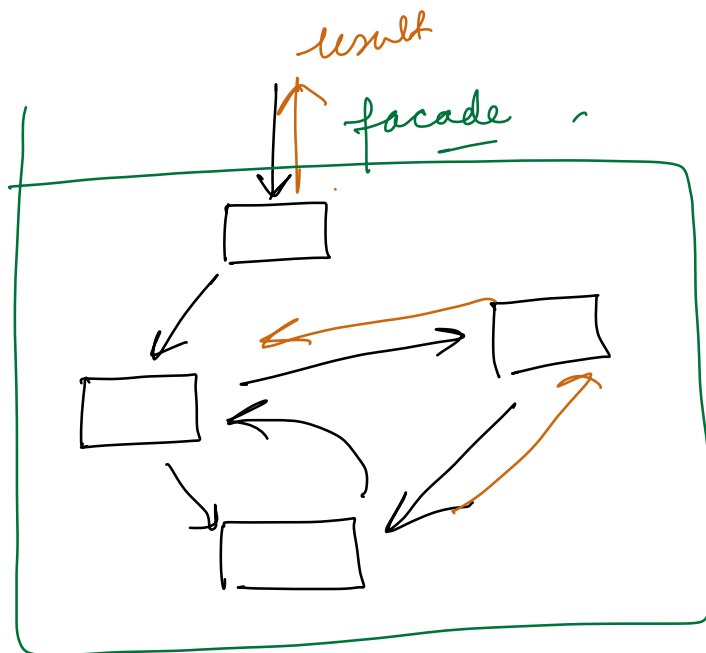
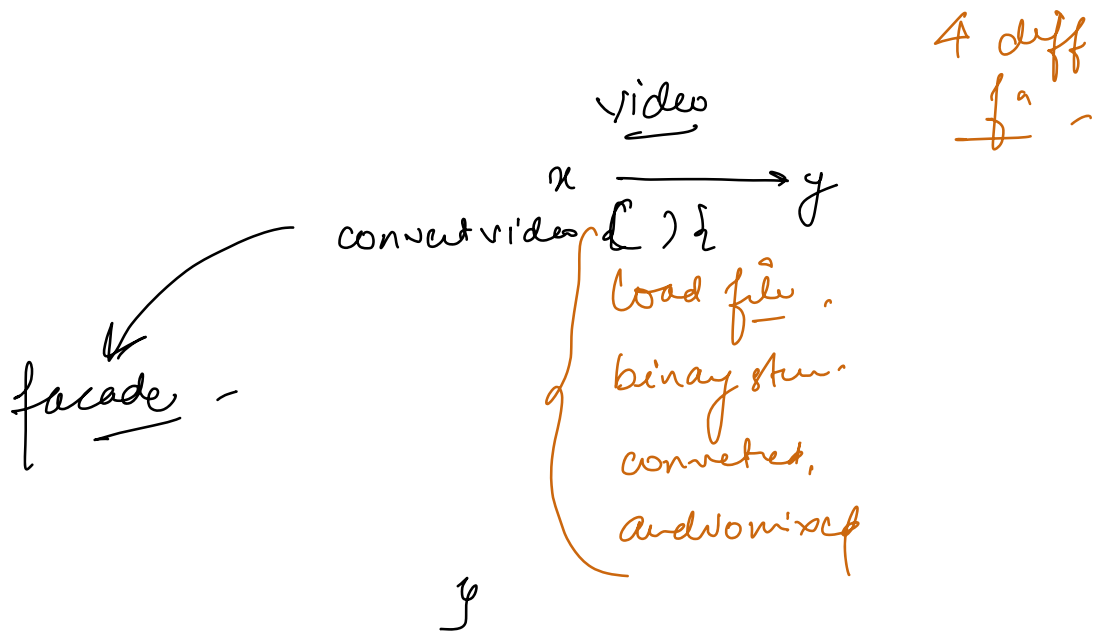
↳ boundary / outside of a building  
deceptive!

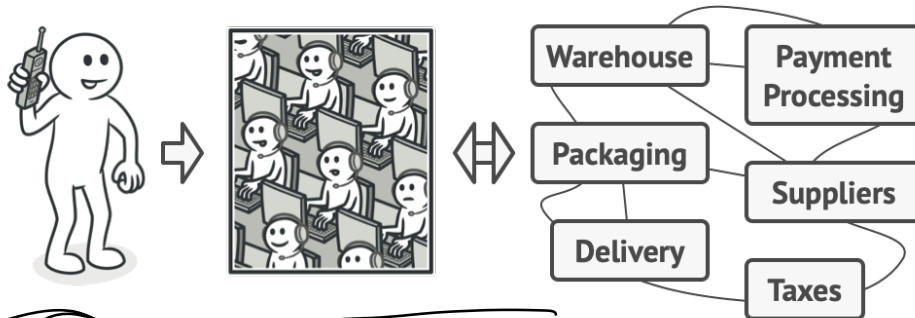
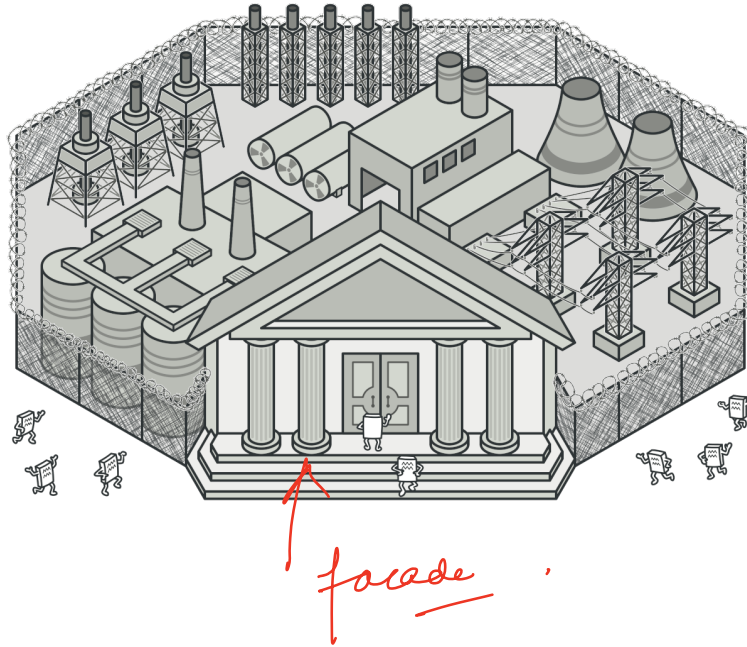


provides cleaner / simplified  
view of a complex trip.

Facade







order placed helper →

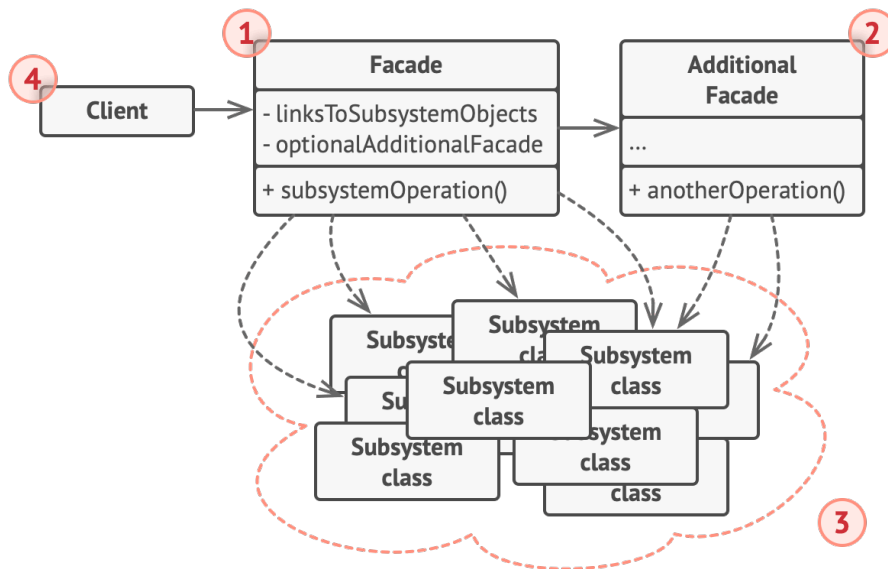
inventory mgmt

sell Notify

Buyer Notify

Email

SMS



⇒ {   
   ✓ Factory → 15   
   ✓ Prototype → 20.   
 }

✓ notification {   
   simple   
   abstract   
 }

✓ prototype

→ { Singleton Builder } — Logger ✓

Strategy   
 UML

LLD-2      mock LLD-1   
 MCO ←      ↓ 30 day



Monday Tue Wed Friday Sun ,

The diagram shows a sequence of days: Monday, Tuesday, Wednesday, Friday, and Sunday. Monday, Friday, and Sunday are underlined. Wednesday has an 'X' over it. A curved arrow points from Tuesday to Sunday.