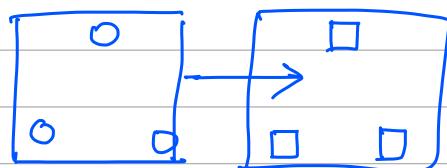
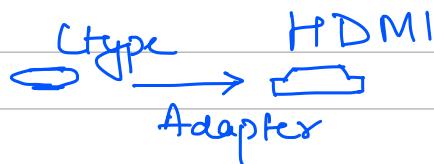


Agenda

Start @ 9.05 pm

- 1) Adapter design pattern
- 2) Facade design pattern

Adapter design pattern

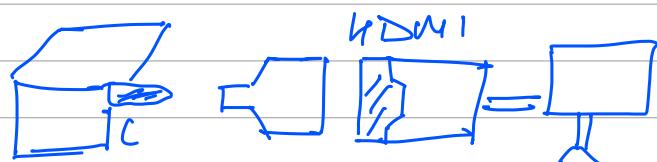


Structural design Pattern

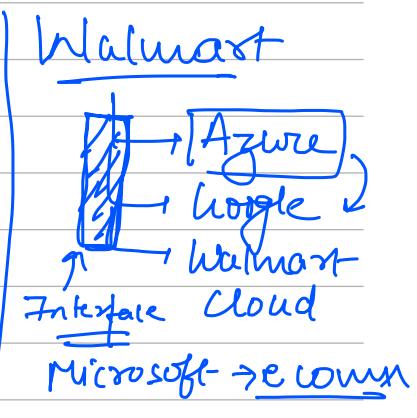
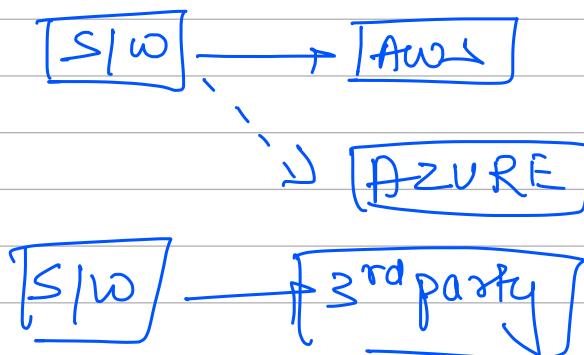
→ How should we structure codebase

- What all classes should be there
- What all attrs and behavior should be there

Adapter: Intermediary layer that converts one form to another

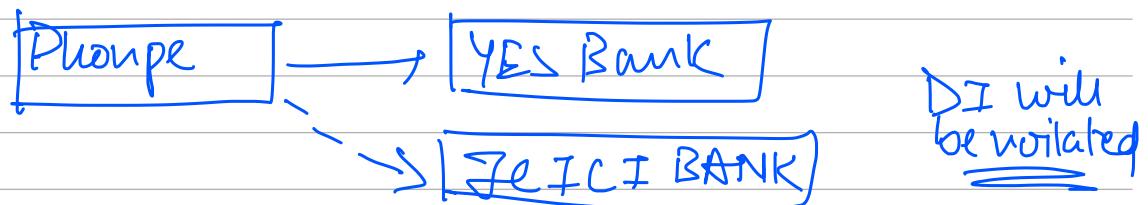


Adapter Design Pattern

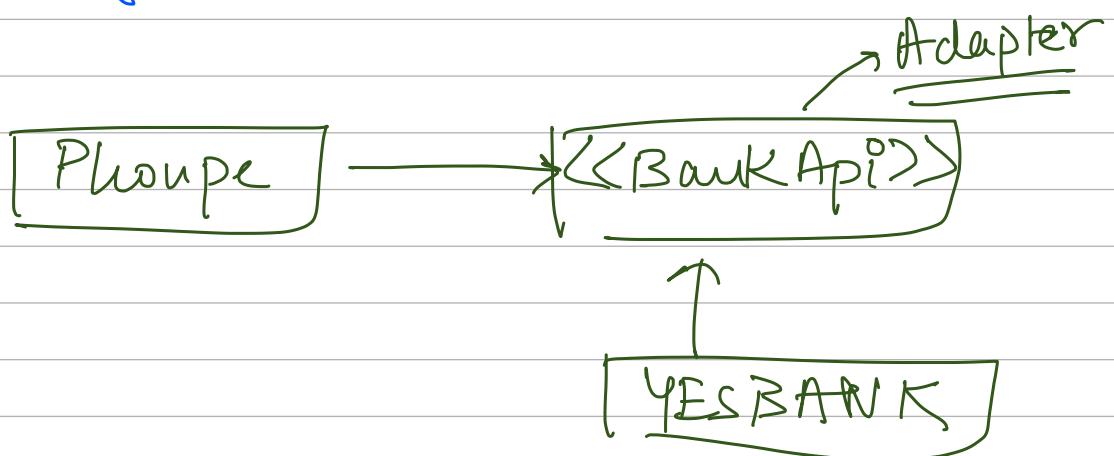


Problem Statement

- 1) We need to migrate from 3rd party service in any scenario, it should be easy.
- 2) If 3rd party company goes down then we will be able to migrate to some other 3rd party

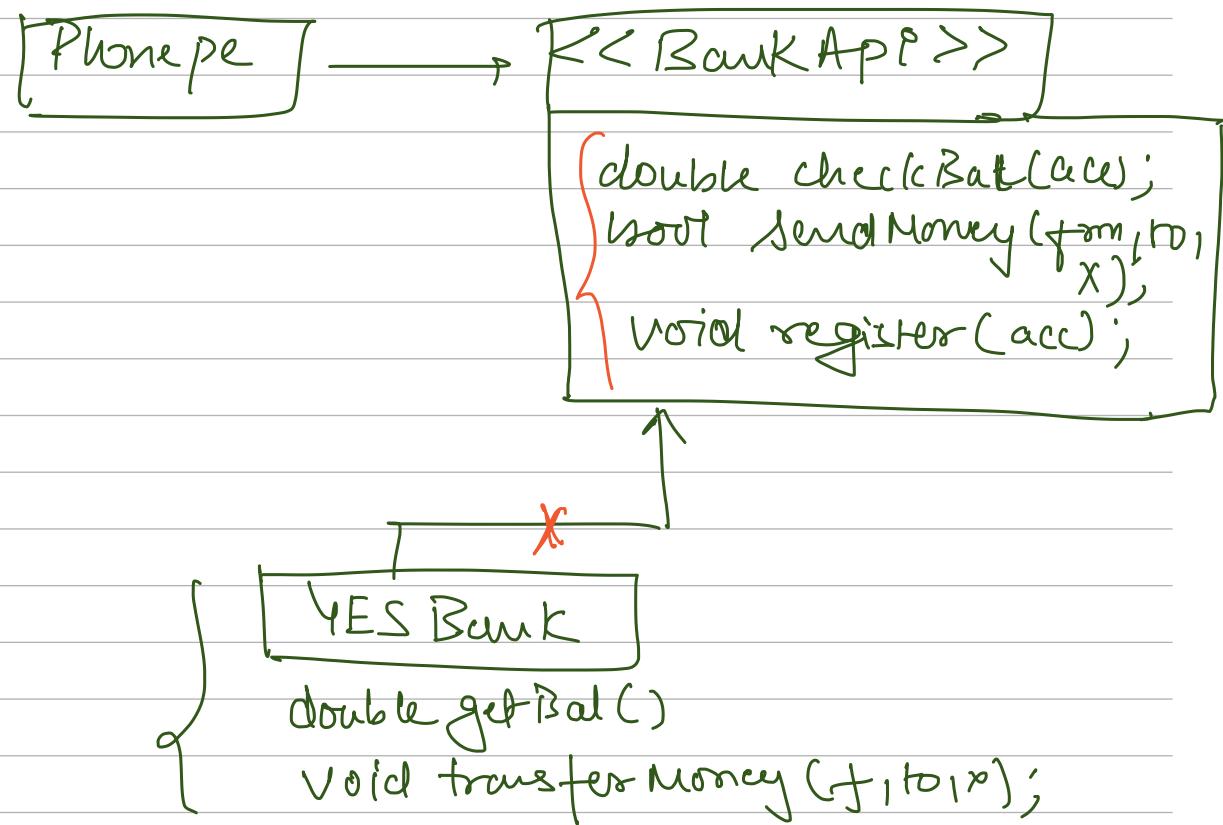


- 2) Adapter design pattern ensures that our codebase remains maintainable in case we are talking to a 3rd party.
- 1) If our code base has tight coupling with 3rd party then our code base violates DI and It will not be maintainable
- 2) Whenever we are connecting to a 3rd party then never connect with them directly, always use an interface



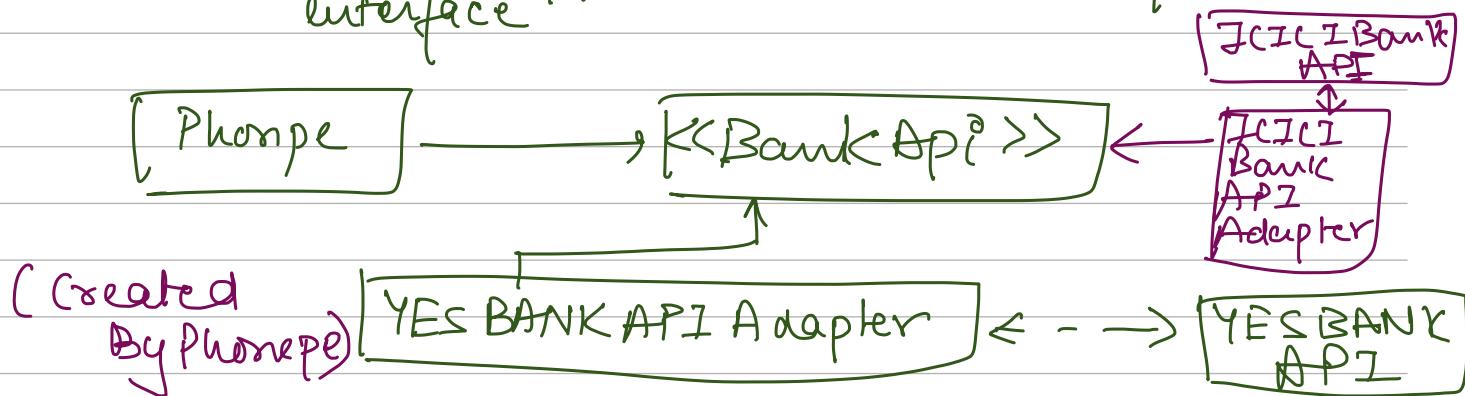
How to use adapter pattern

- 1) Whenever we are connecting to a 3rd party service, create an interface with all methods that we need from 3rd party



- 2) As 3rd party won't implement this interface for us, we should implement the interface & internally use the 3rd party APIs

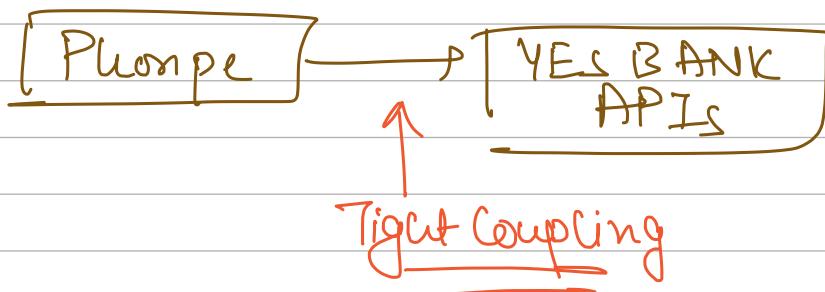
→ Create a wrapper class which implements interface



Summary

Adapter \Rightarrow 3rd party integration

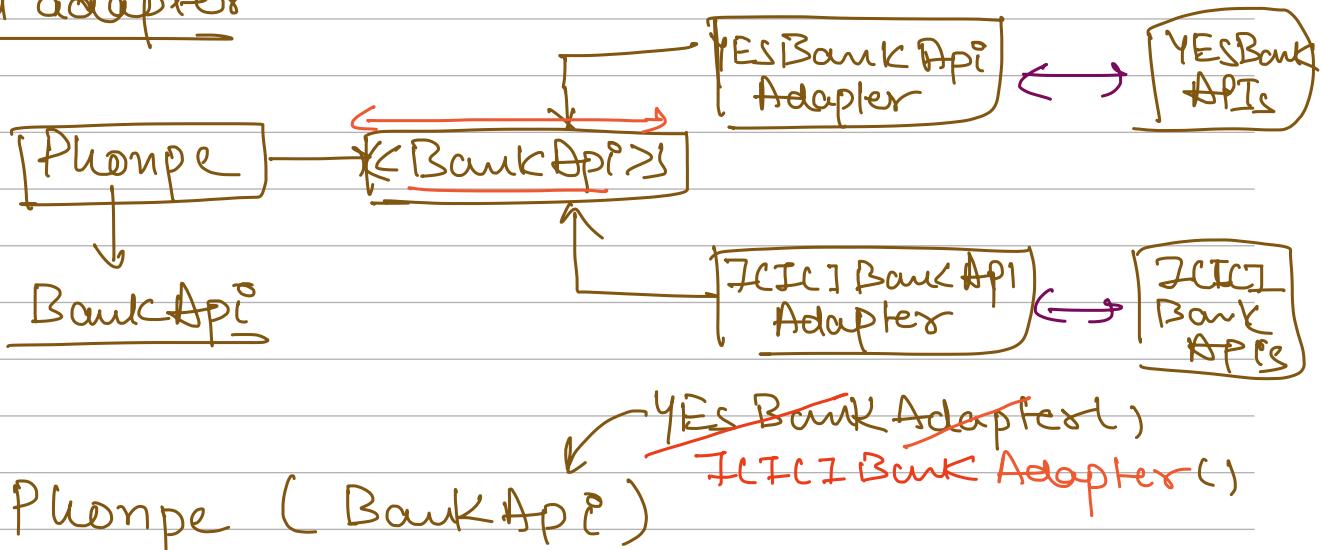
without adapter



1) Violating DIP

2) Testing & Regression will take huge effort & time

with adapter

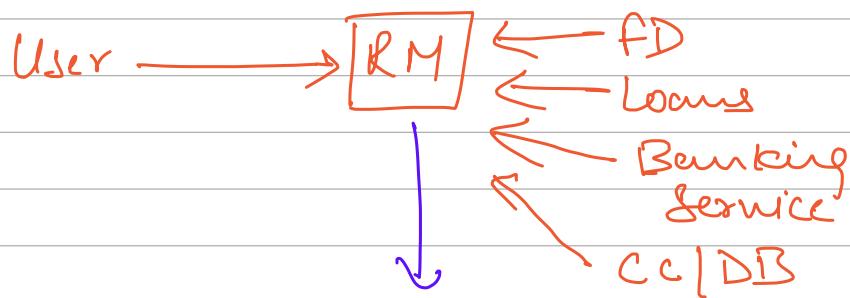


FAÇADE Design Pattern



Outer part of a Building

- provides a clean / simplified view by hiding internal complexity



Acting like a facade which simplifies all things by hiding complexities

~~leg~~

Amazon Order Placed	
Inv Mgmt Service	is;
Logistic Service	is;
Notification Ser	is;
Seller Service	ss;

- Currently all work is done by one class
- Making class **TOO BIG**

On Order Placed () {

- ls. update inventory
- ls. start tracking
- ss. initiate pickup
- ls. send email

4

Facade

Whenever we have a method / class that is doing a lot of work , instead of doing the work in the main class , move that to some helper / facade class -

