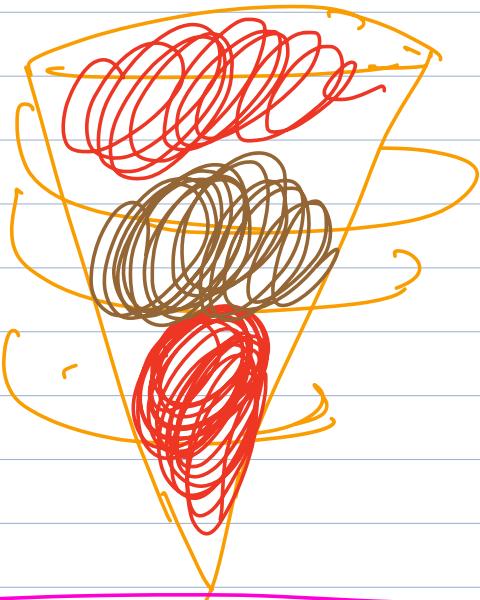


Agenda

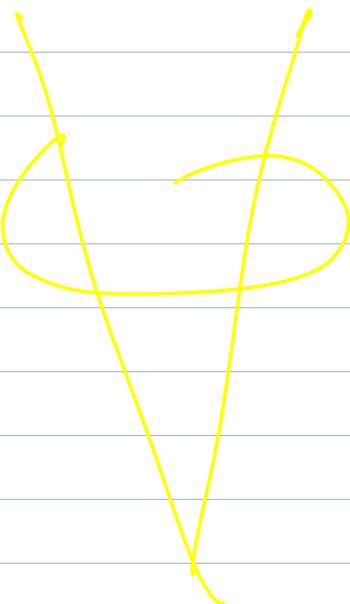
- ① Decorator Design Pattern
- ② Facade Design Pattern
- ③ Flyweight Design Pattern

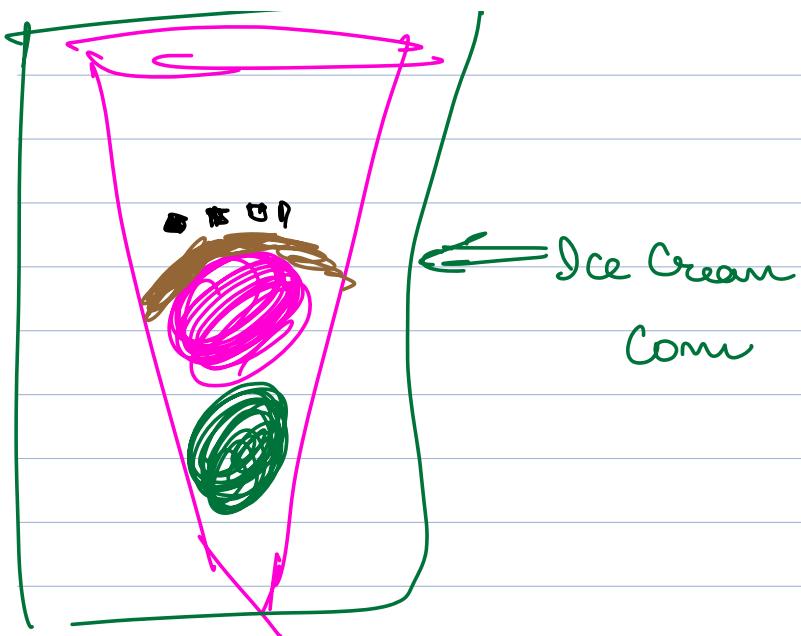
DECORATOR DESIGN PATTERN

| Ice Cream Cone



| Ice Cream Cone

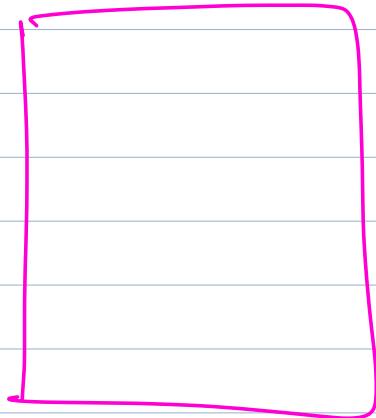
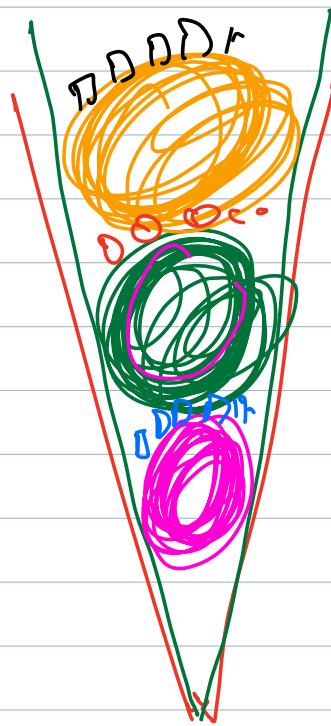
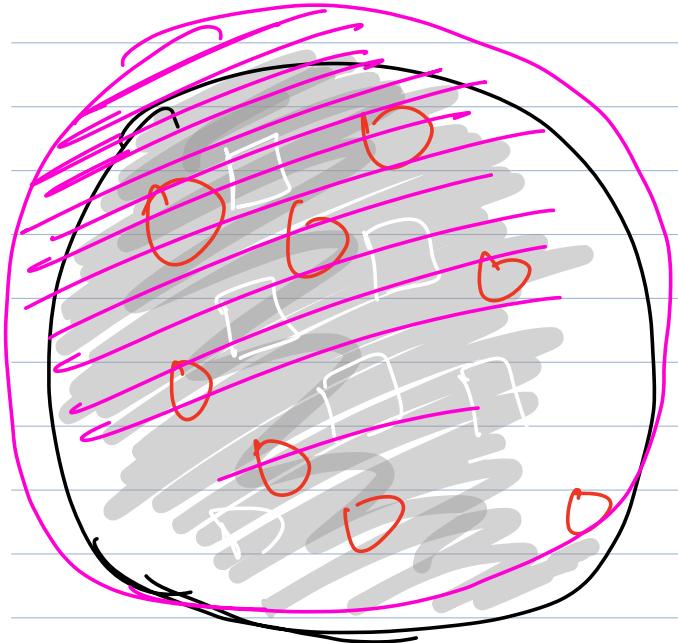




Ice Cream Cone

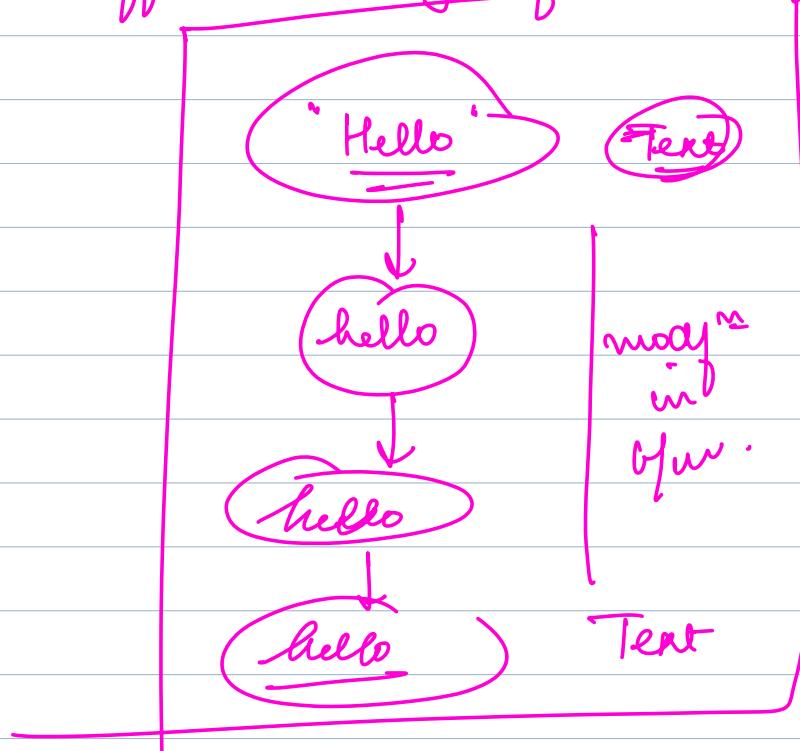
class IceCreamCone {
 list < Scoop > Scoops;
 list < Topping > toppings;
}



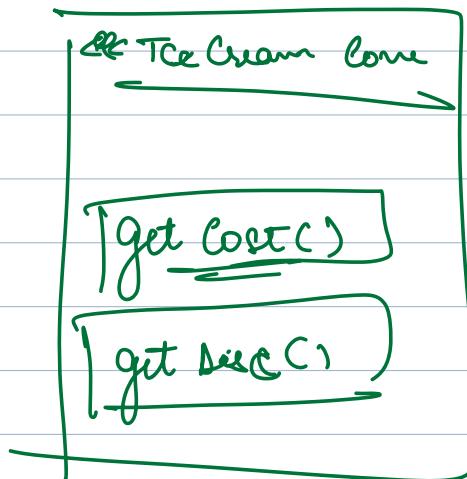


We start from something of type of X
and we also end with something of type X

Problem: Start with something and keep modifying it to reach another something of same type



DECORATOR DESIGN PATTERN =



"Orange Cone + Choco Scoop + Vanille Scoop + Choco
Cone
Choc

£2.00



«Ice Cream Cone»
get Cost()
get Disc()

Step 1

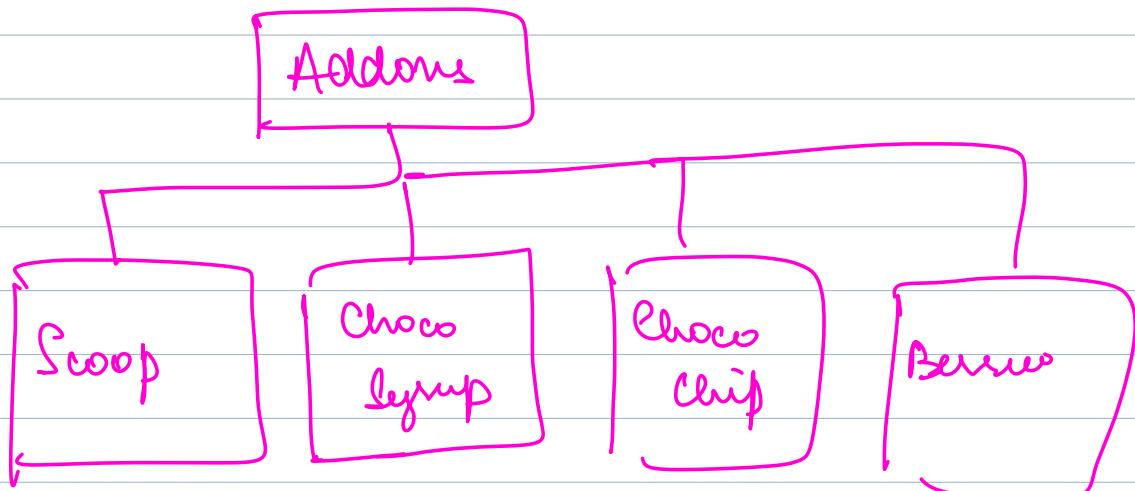
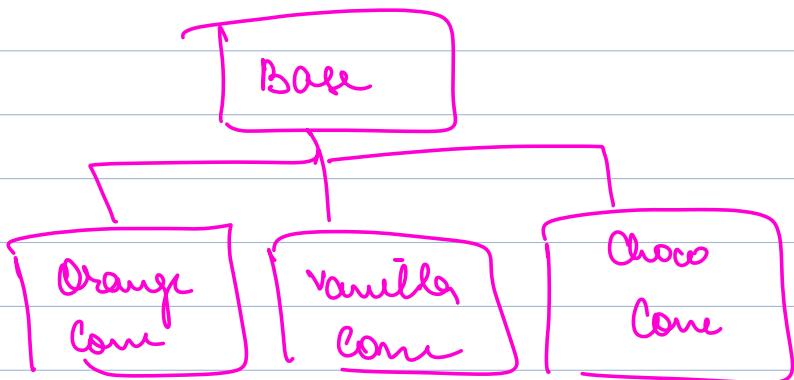
Define an interface / abstract class. that
rep the thing that we are constructing

Step 2

There are 2 types of entities that
I have



There can be multiple entities of each type

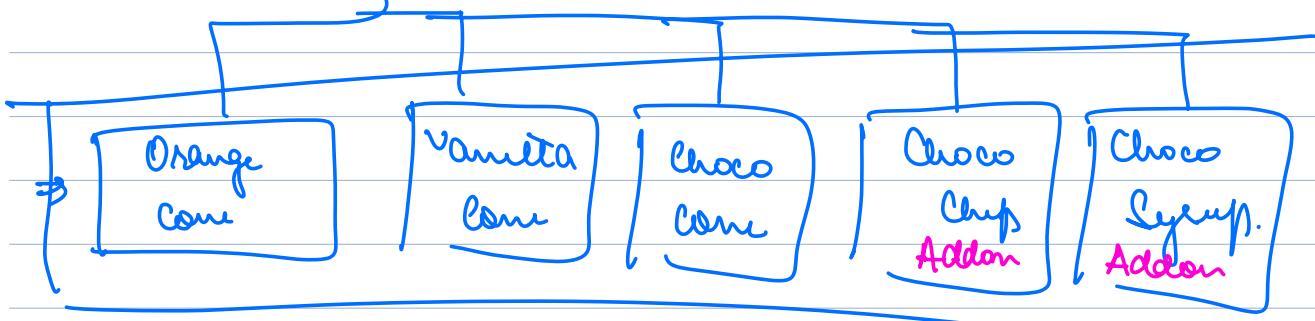


Step 3

Create a class for each of them

to implement uniform
abs class Step 1

IceCream Cone



Code for each

for Base Entities

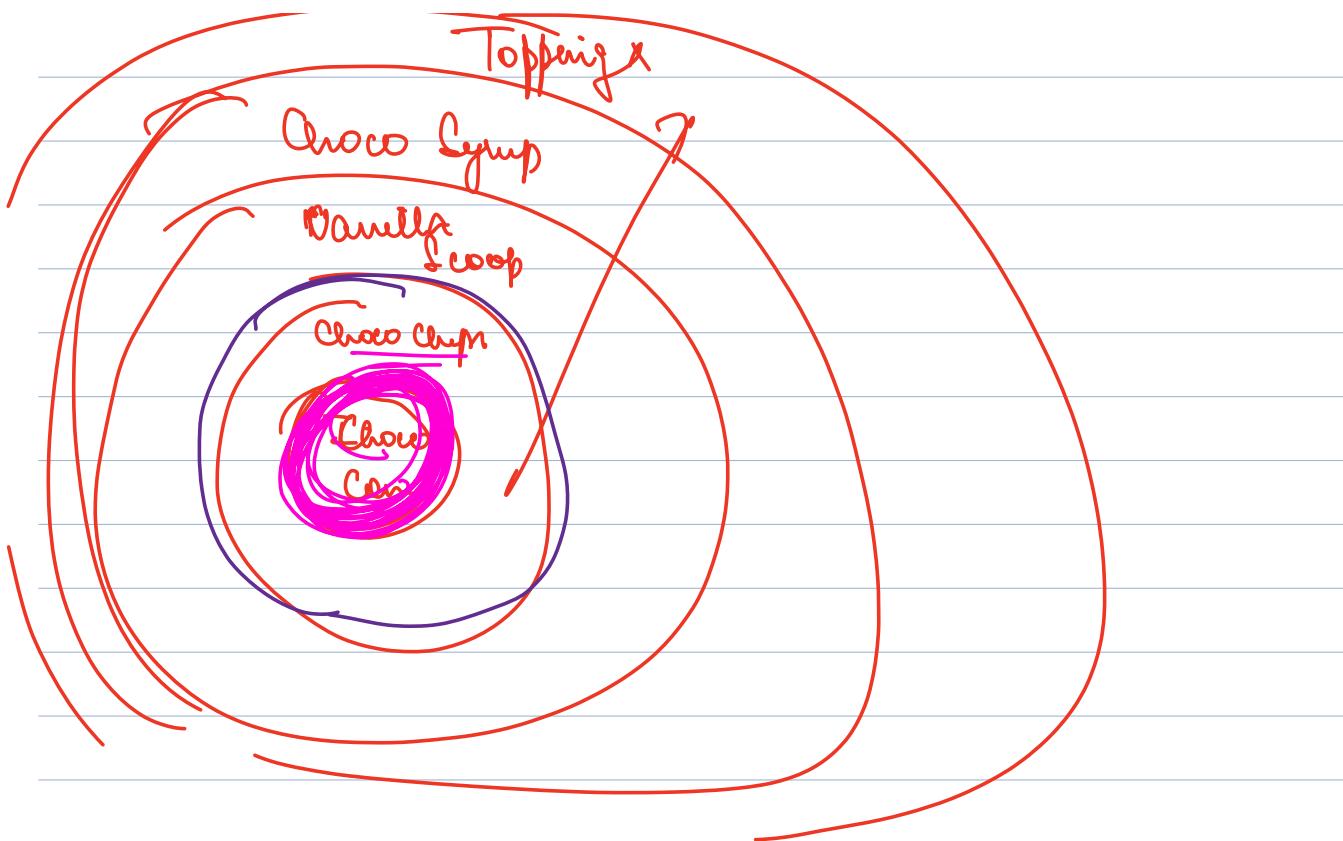
class ChocolateCone implements IceCreamCone {

get Cost() {

return 10;

get Size() {

return "ChocoLan";

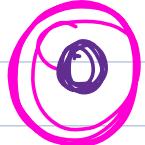


- ① Choco Chip sits on Ice Cream Cone
- ② After adding it combined entity is still an ice cream cone
- ③ Choco Chip is an add on entity

Class Choco Chip Addon implements Ice Cream Cone {
Ice Cream Cone ~~(cone;)~~
Choco Chip Addon (Ice Cream Cone) {
 this - cone = cone;

} getCost() {
 return cone.getCost() + 10;

getDesc()

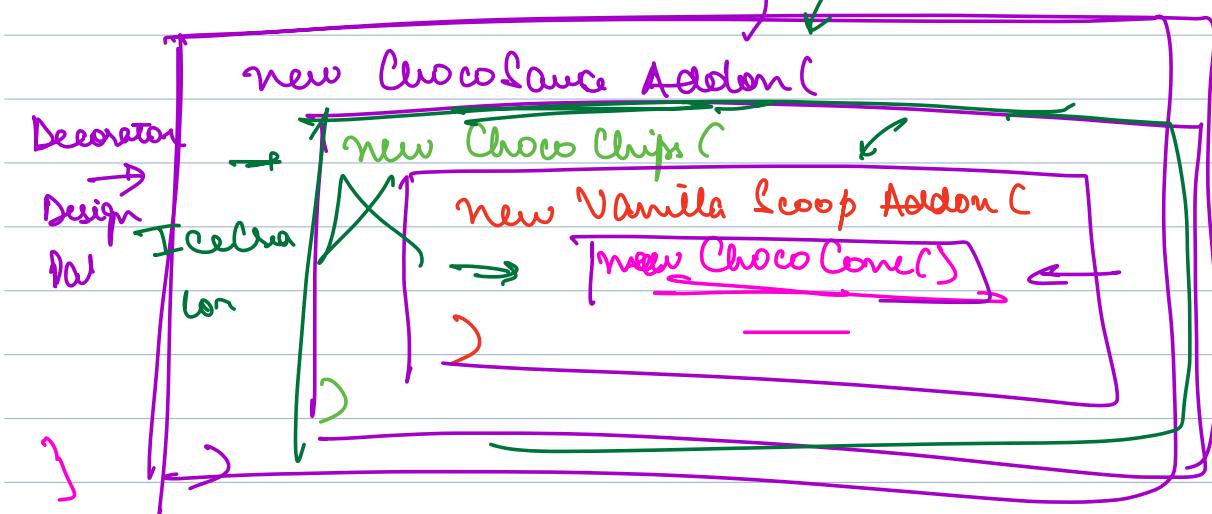


return cone.getDesc() + " Choco Chef"

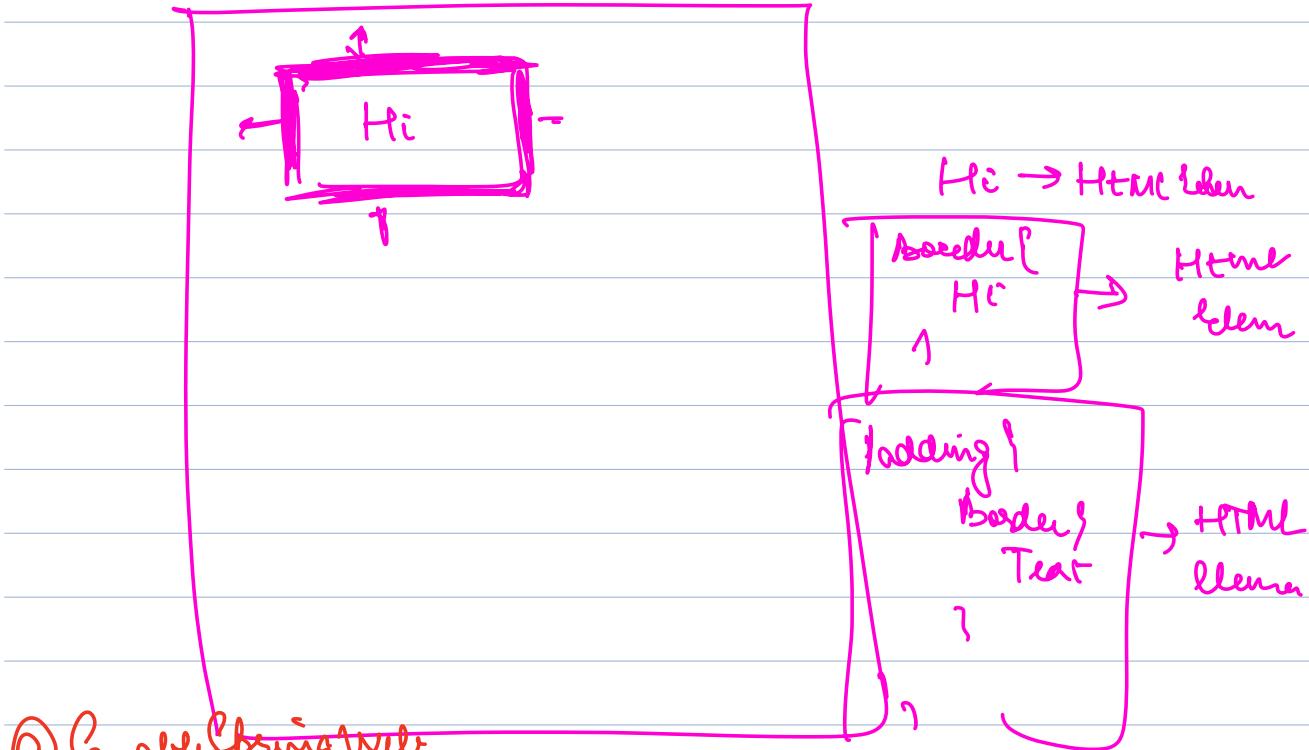
}

Client {

IceCreamCone i = <--



Class Vanilla Scoop Addon implements IceCreamCone



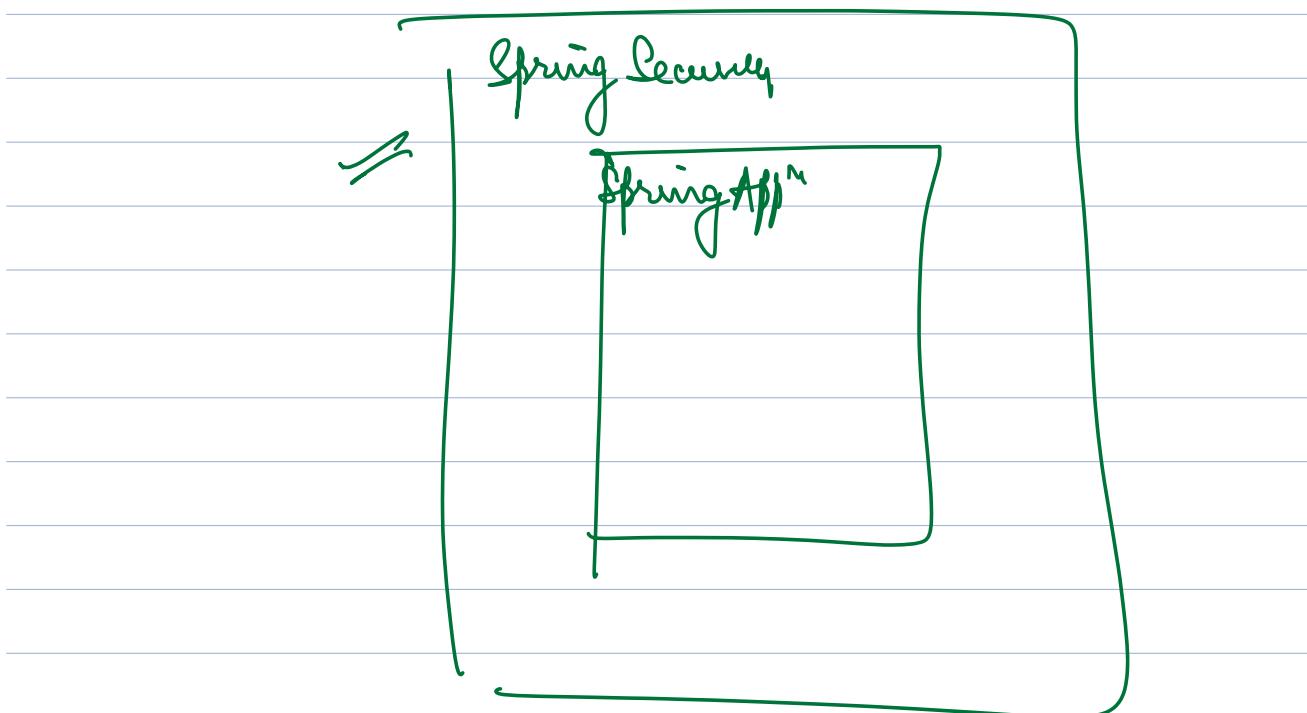
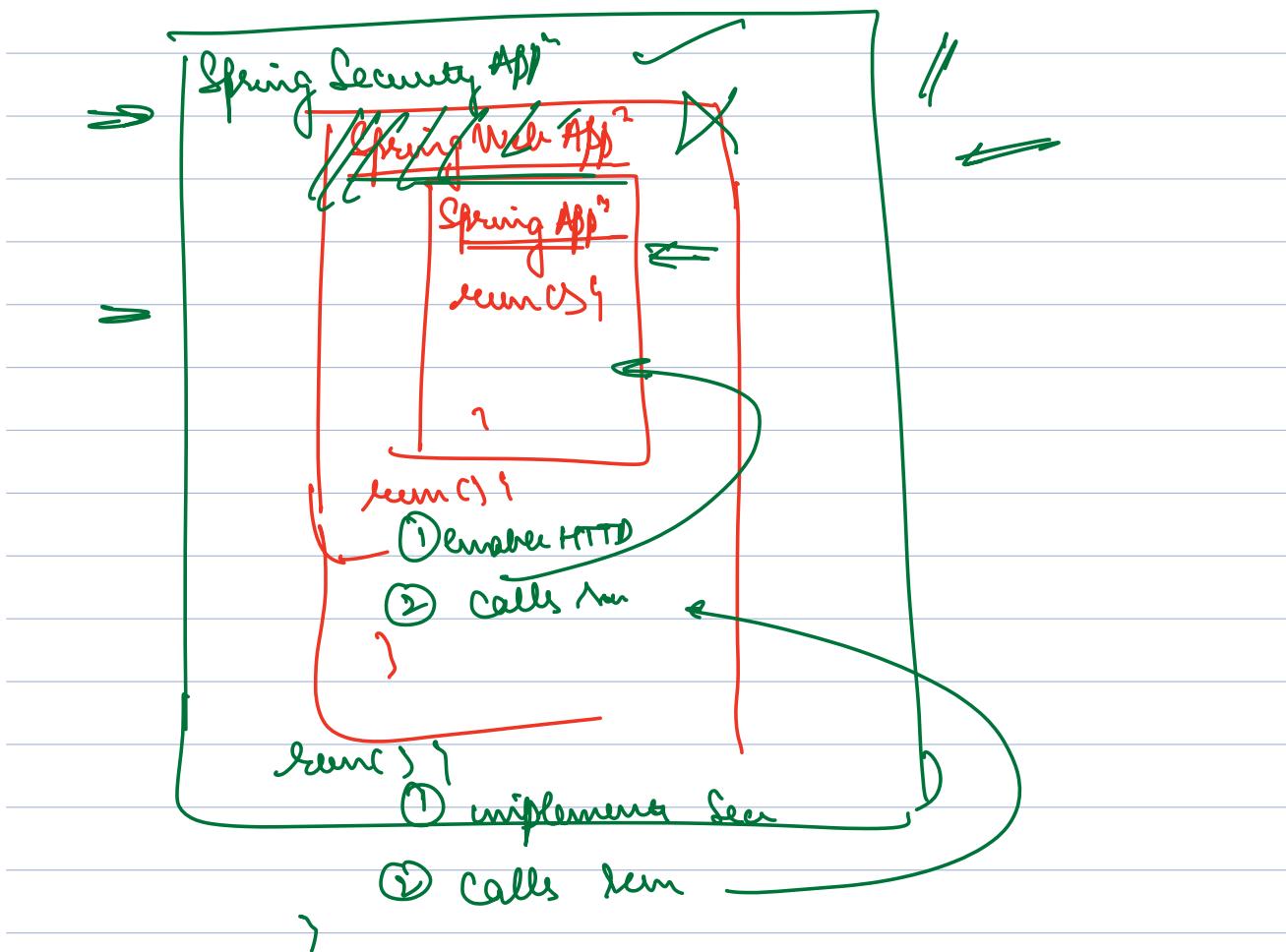
@EnableSpringWeb

(@EnableWebSecurity) ← Decorator
Book My Show SpringApp

run();

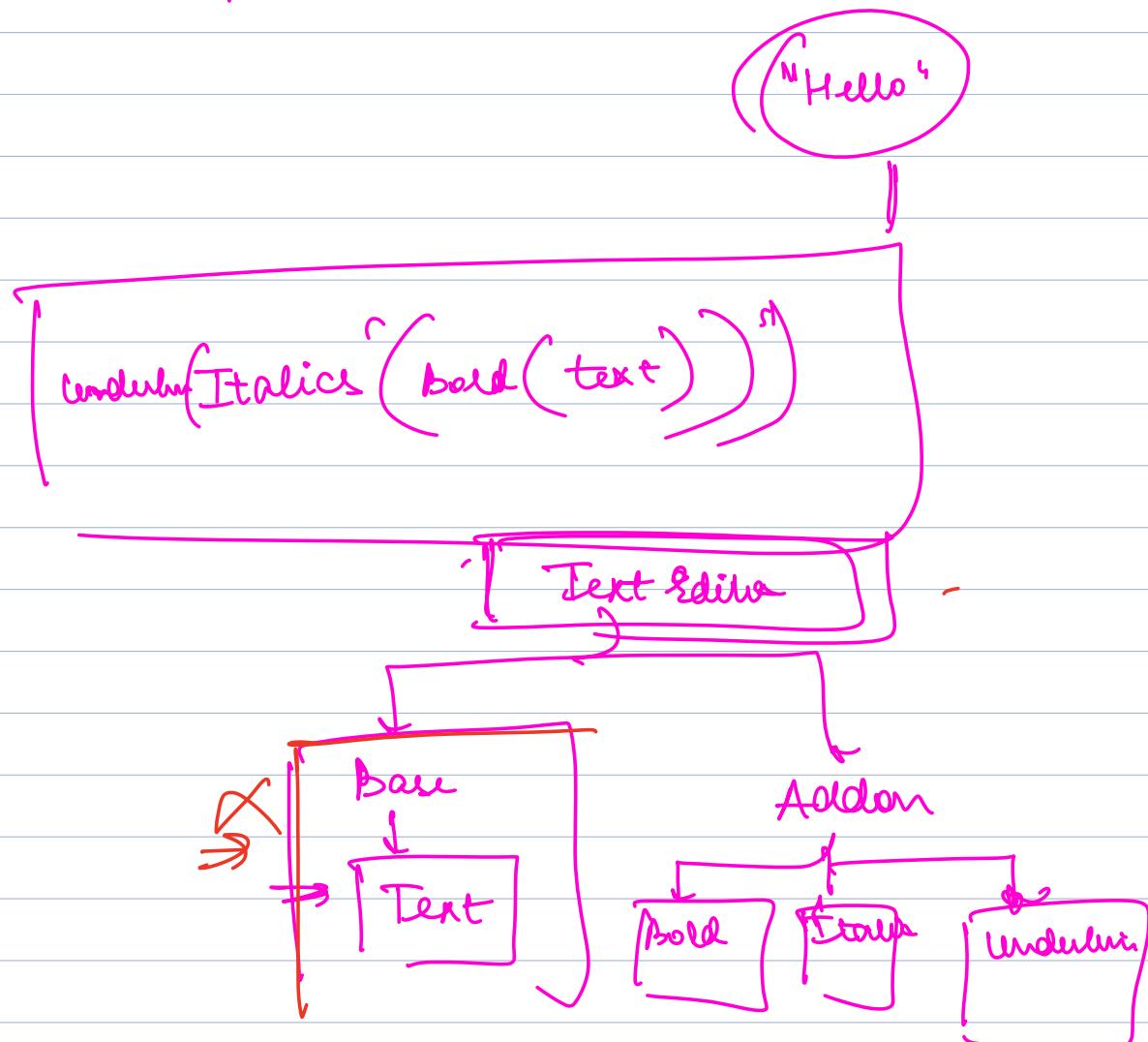
?

?



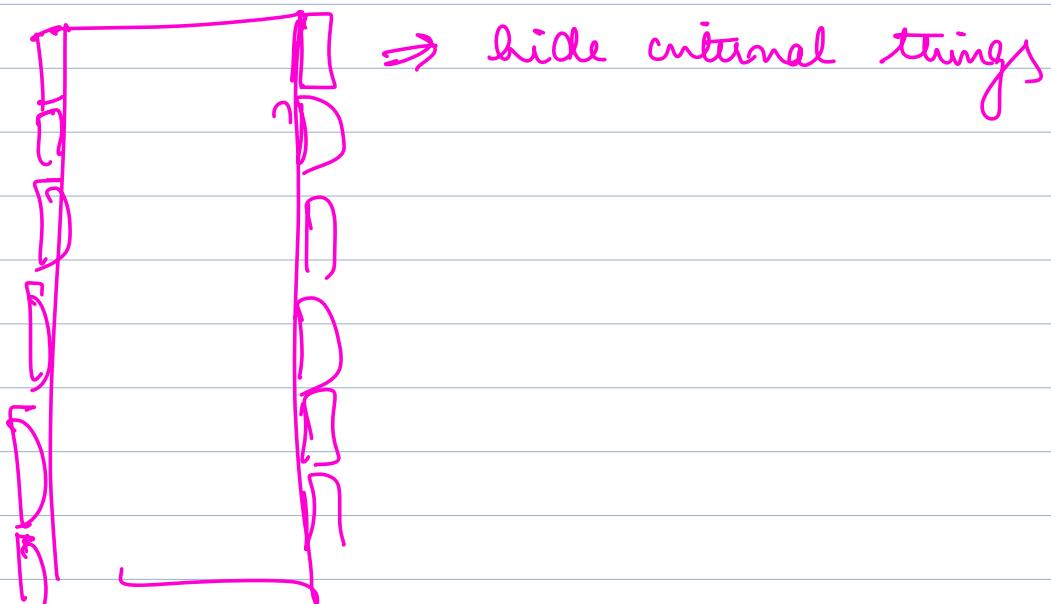
→ I have a text editor
get text()

→ I want to restructure how text is output to client



```
Class TextEditor {  
    private String text;  
    getText() {  
        return text;  
    }  
}
```

Facade Design Pattern
Boundary
External] of a building



```

class Flipkart {
    InvoiceGenerator ig;
    Inventory MS ms;
    NotificationService nfc;
    String OrderPlaced ( int orderId ) {
        ig. SendInvoice ( ^ )
        ms. updateInventory ( productId )
        nfc. notify ( buyer, __ )
        [ seller, __ ]
    }
}

```

{ Order Cancelled }

{ Order Refunded }

Facade

```

Flipkart
- OrderPlacedUtility utility
OrderPlaced()
utility. onOrderPlaced()

```

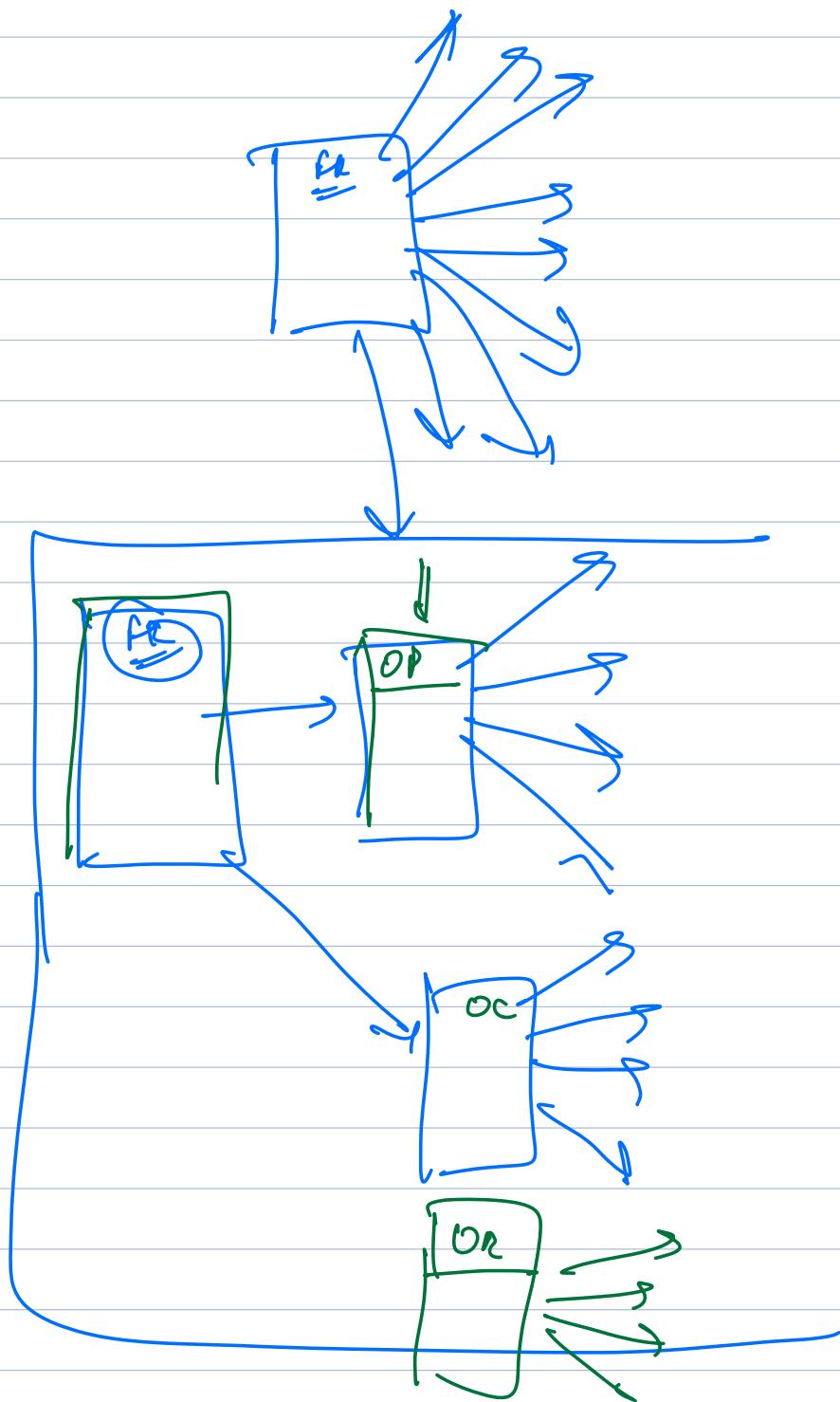
```

OrderPlaced Utility
InvoiceGenerator ig;
Inventory MS ms;
NotificationService nfc;
OrderPlaced()

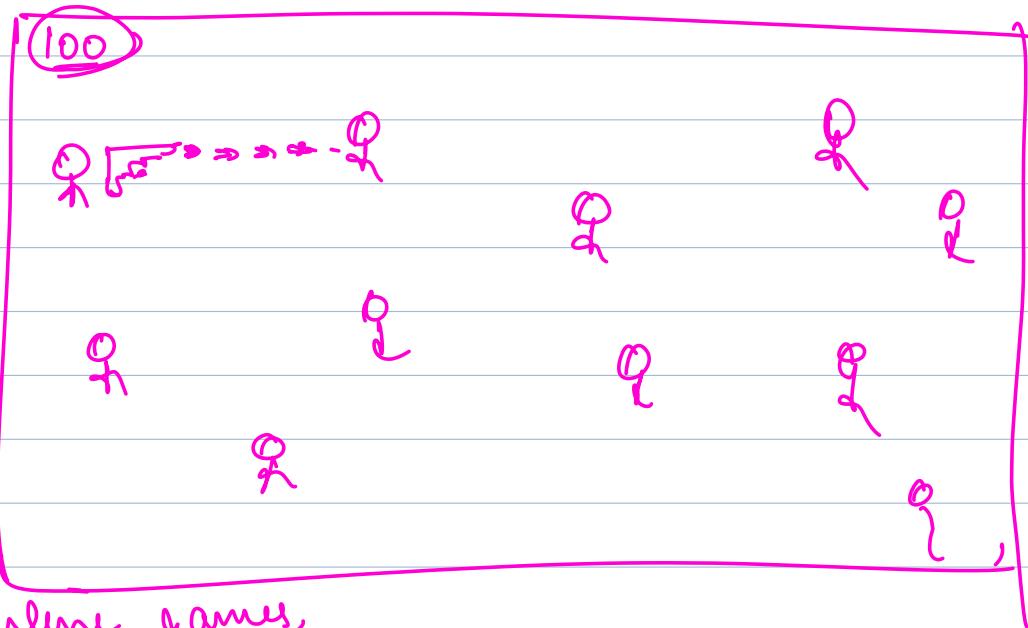
ig. SendInvoice ( ^ )
ms. updateInventory ( productId )
nfc. notify ( buyer, __ )
[ seller, __ ]

```

- seen, -



Flyweight Design Pattern

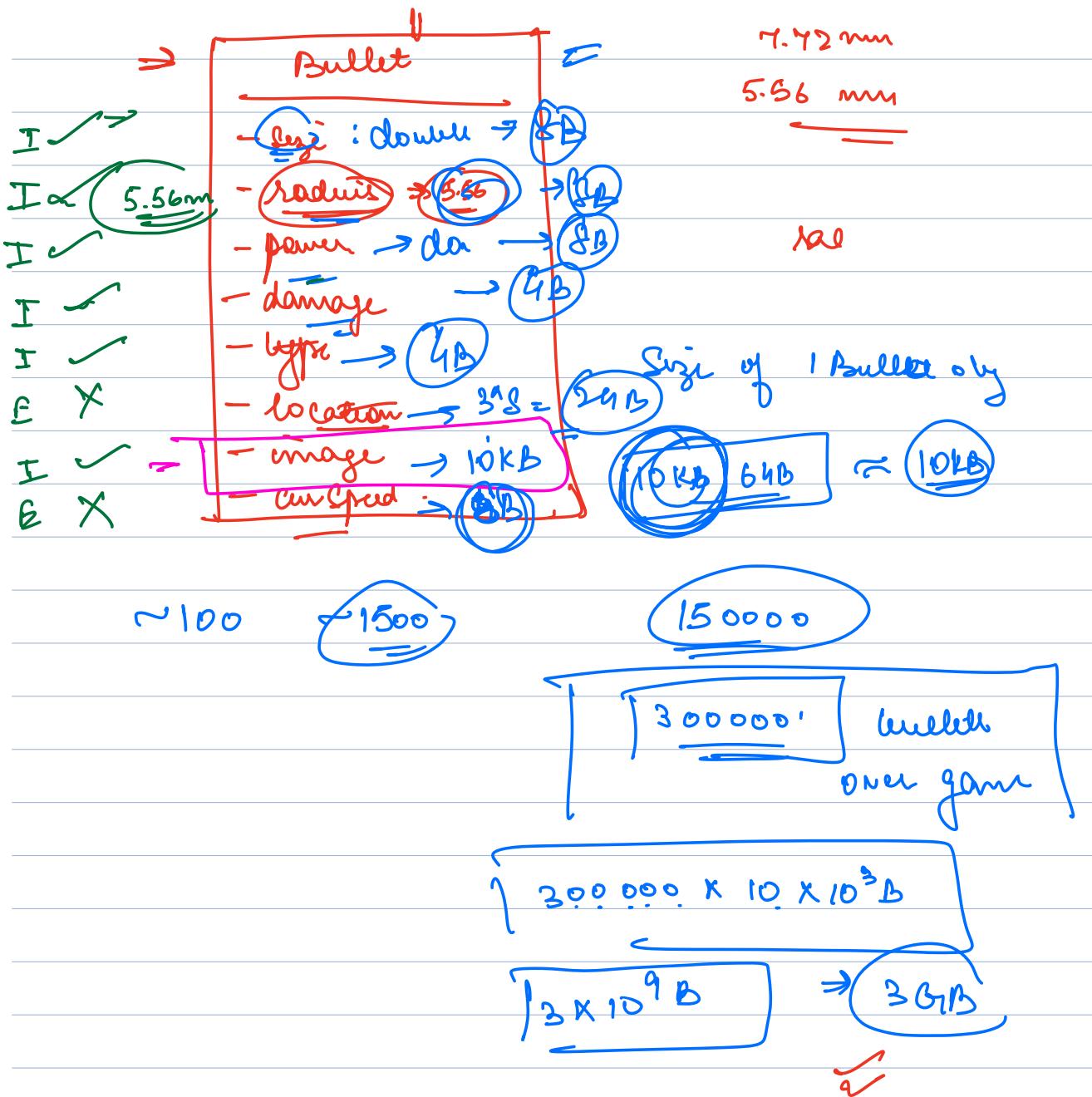


In Online games,

At the start of game complete
state of game universe is downloaded
to your machine

Any change that happens is broadcasted
to all players





Flyweight DP

- Objects have 2 types of properties:
- ① Intrinsic → that won't change
 - ② Extrinsic → that will change

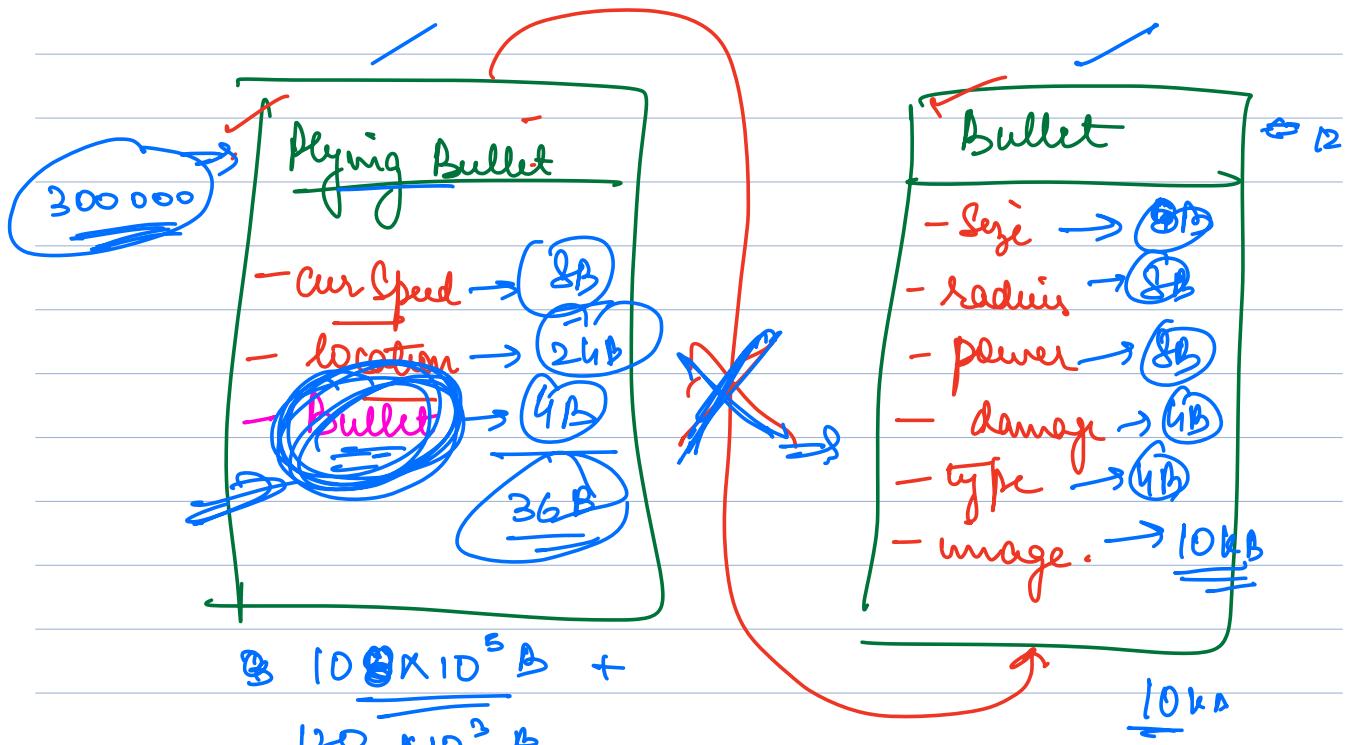
→ Sometimes such objects lead to heavy memory consumption

0 0 0 0 0 0 0 0 0 0 0 0

Split that class into 2 classes.

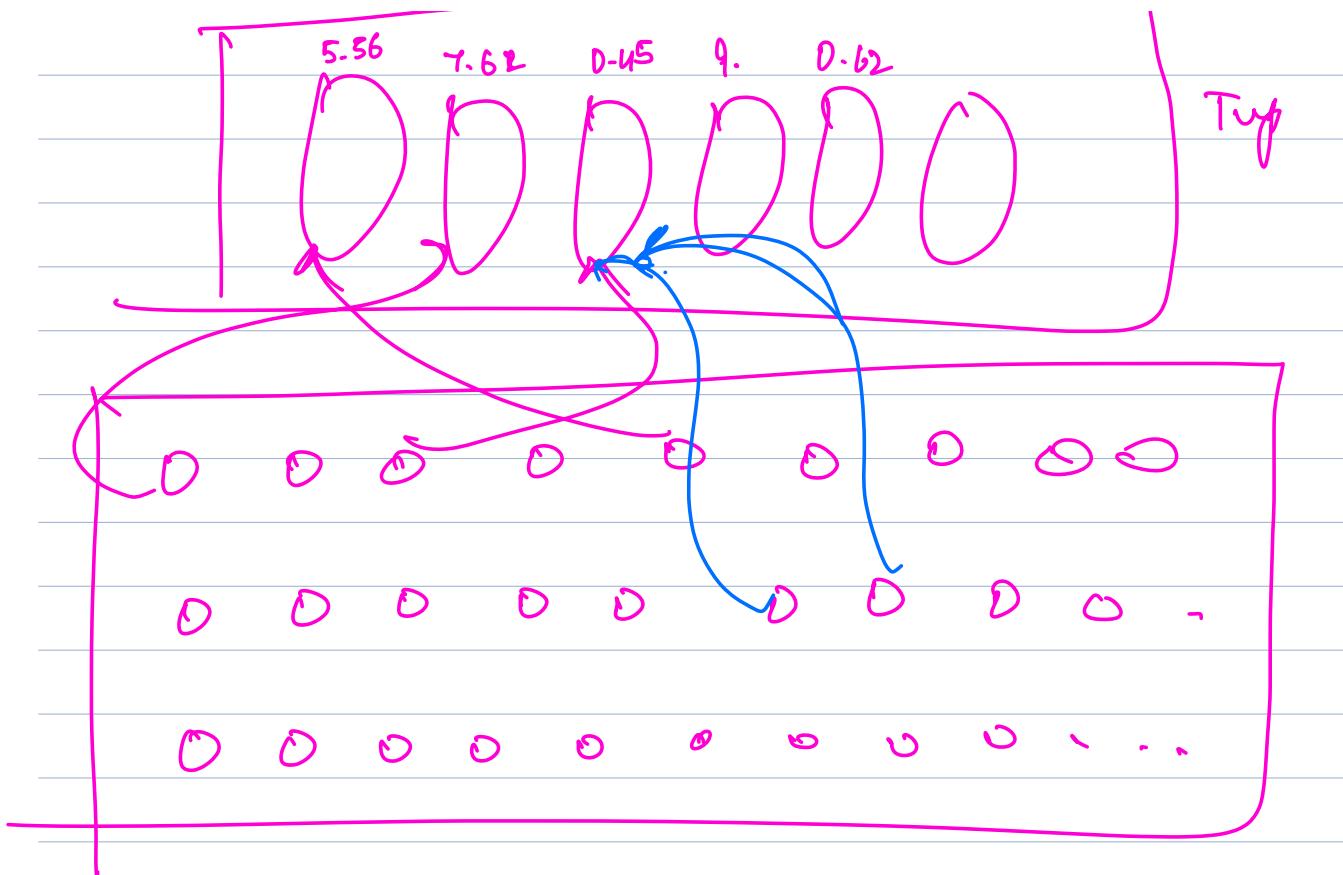
1 class \Rightarrow all extrinsic

2 class \Rightarrow all intrinsic



$$\begin{aligned}
 & \text{FlyingBullet} \rightarrow 10.8 \times 10^6 B + 120 \times 10^3 B \\
 & \rightarrow 10.8 \text{ MB} + 120 \text{ kB} \\
 & \approx 10.8 \text{ MB} \quad \leftarrow [3 \text{ GB}]
 \end{aligned}$$

→



var new Bullet = new Bullet(gm, —)

