

1. Good Evening

2. We begin at 9:10 pm

3. Topic - Flyweight & Decorator

Structural Agenda

1. Flyweight Design Pattern

: UI of Games

2. Decorator Design Pattern [LL]

: Add dynamic responsibilities

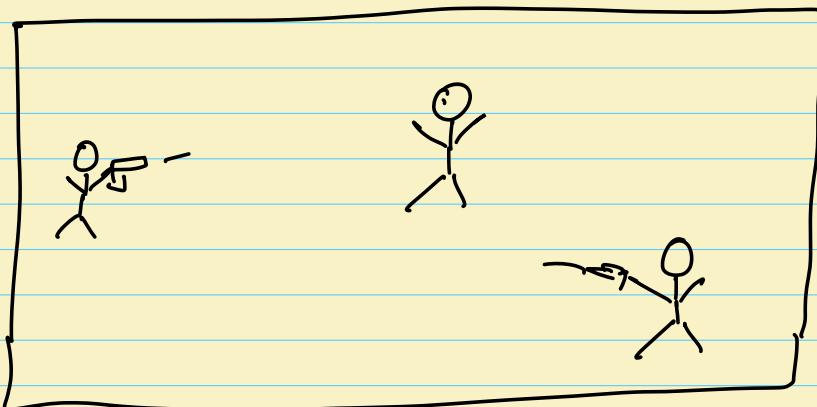
3. Message.

Flyweight Design Pattern

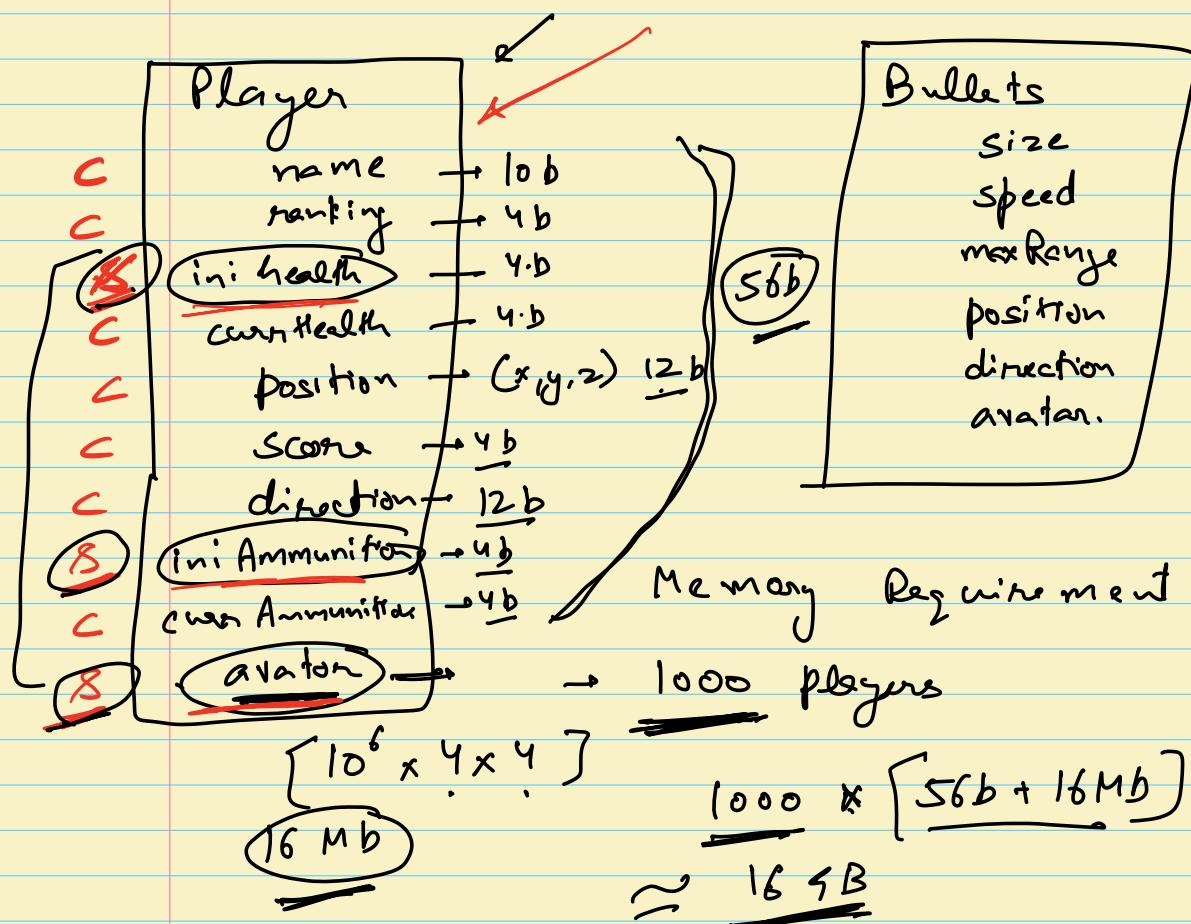
→ UI of Games

* PUBS, Contracts

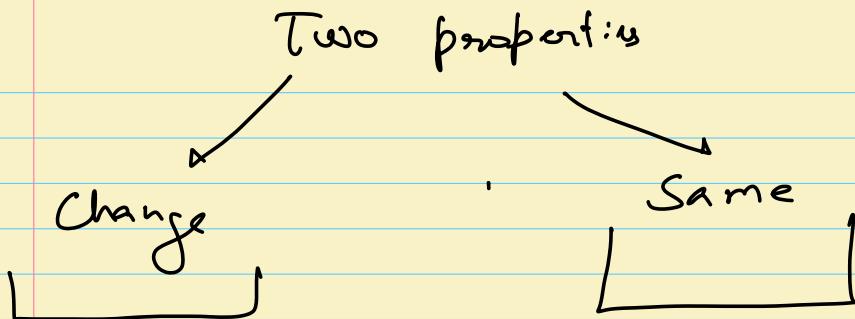
* [To save memory / RAM space.]



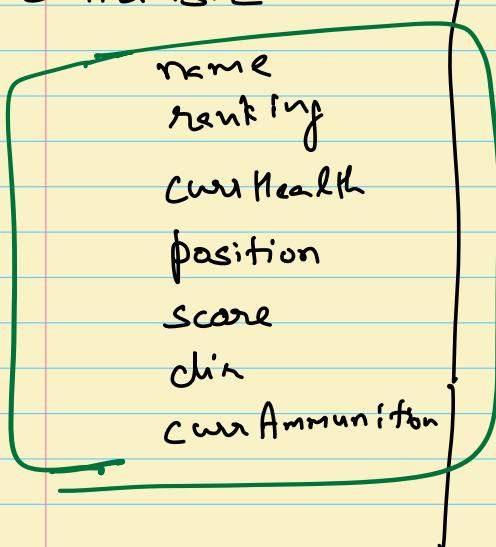
Battle Ground / Scene



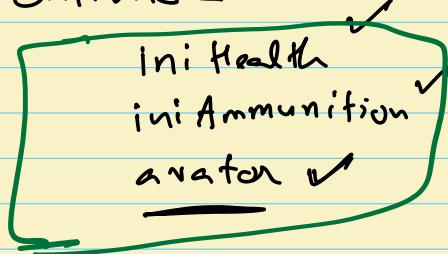
256 MB RAM allowed to play if.



Extrinsic

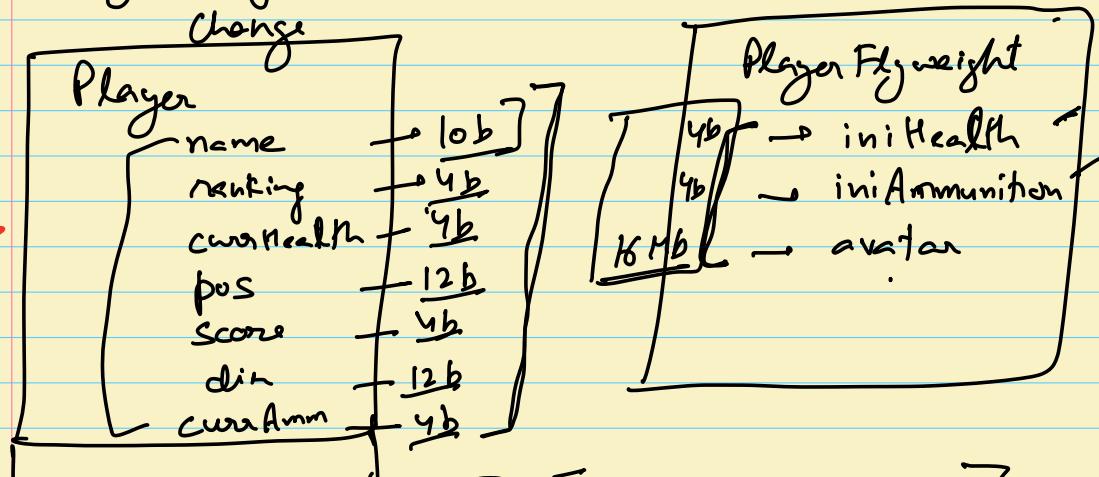


Intrinsic



→ Many Players

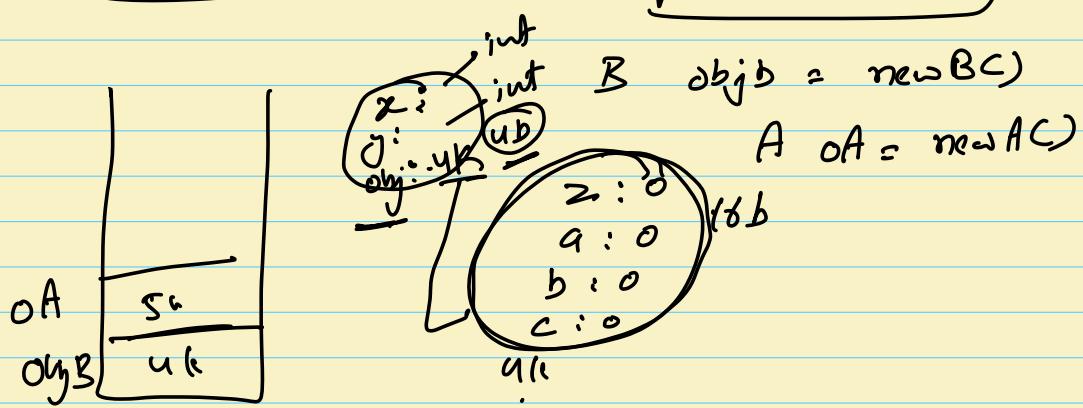
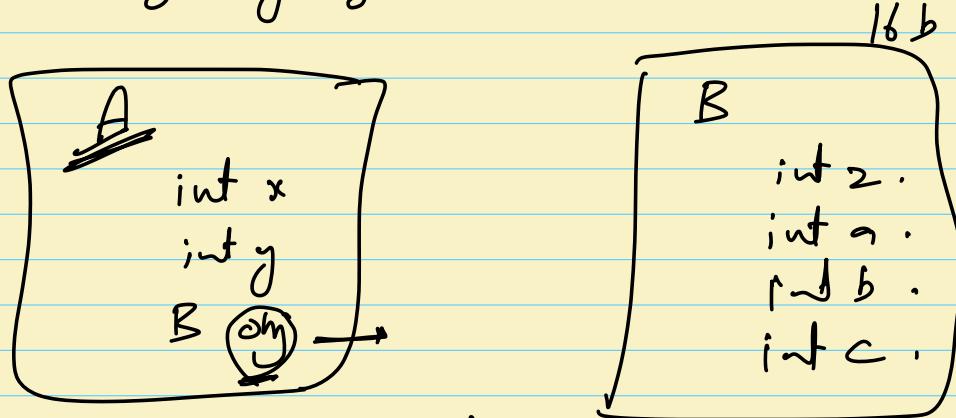
→ Player types are limited



PlayerFW ~~fw~~ ~~star~~. Reference = 4b

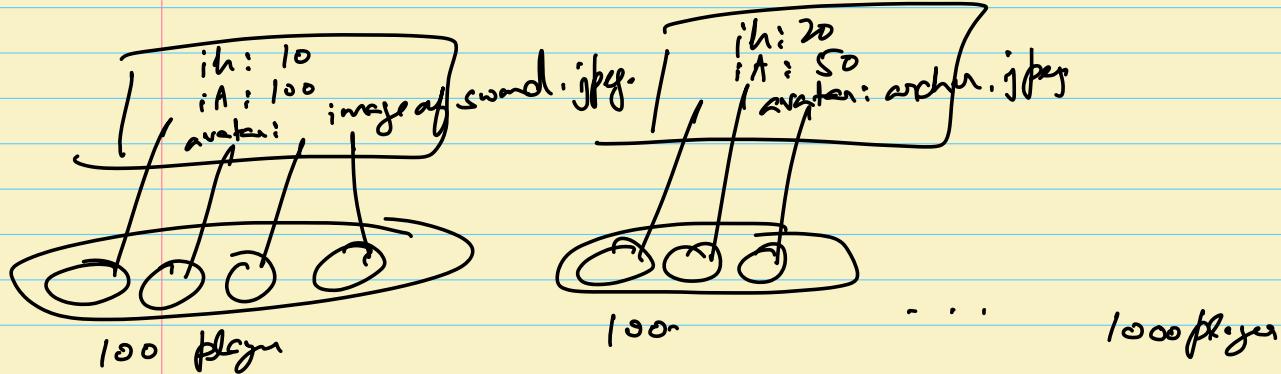
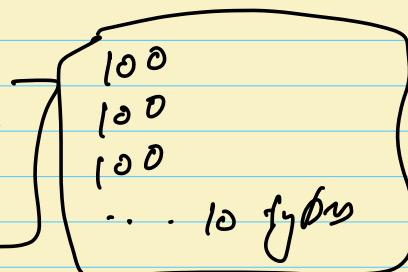
→ 1000 players

10 Player Flyweights



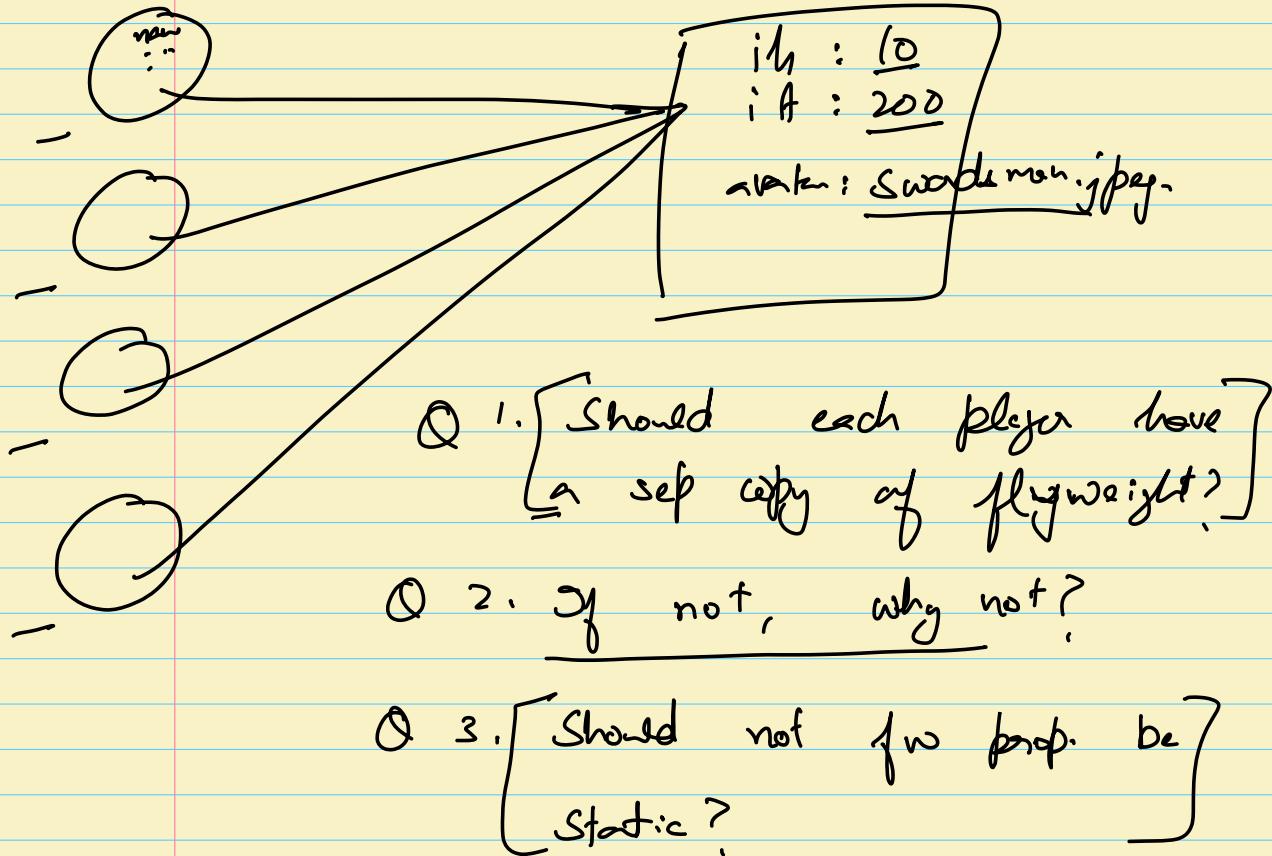
→ 1000 players

→ 10 fws.



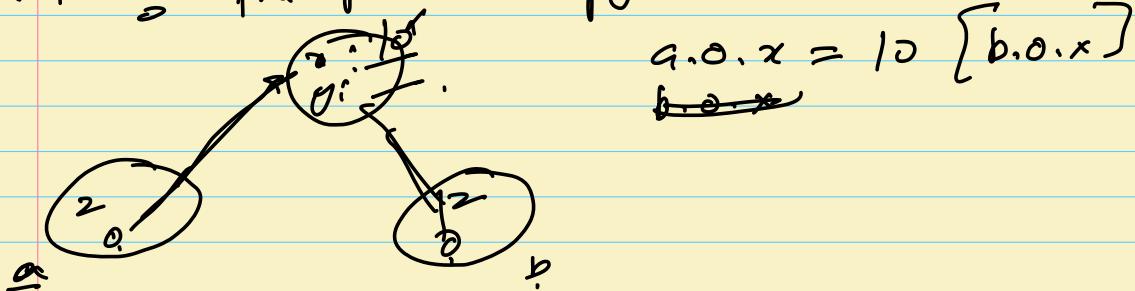
1000 player

100 players



A1 \rightarrow No separate copy.

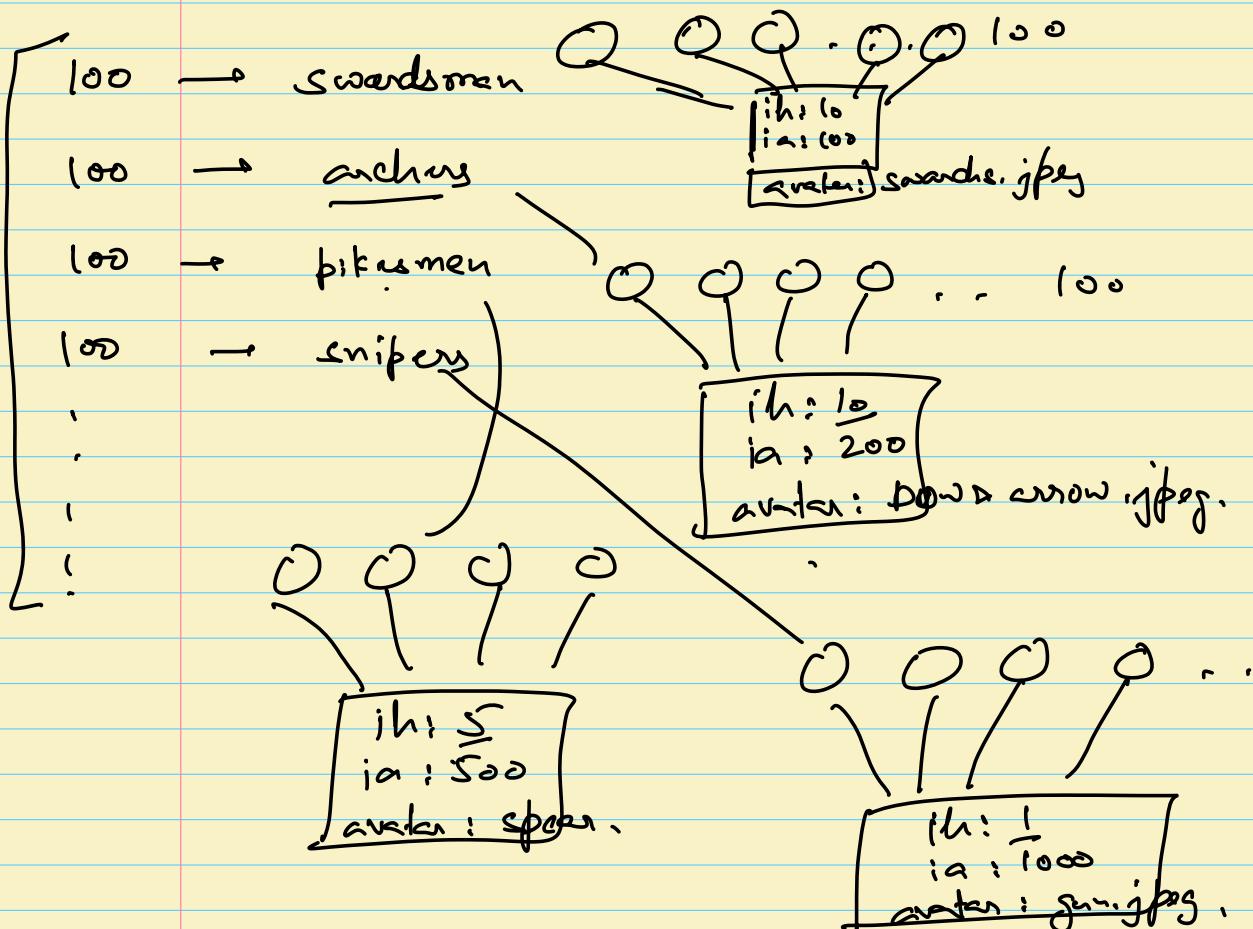
A2 \rightarrow ∞ the prop in flyweight don't change.

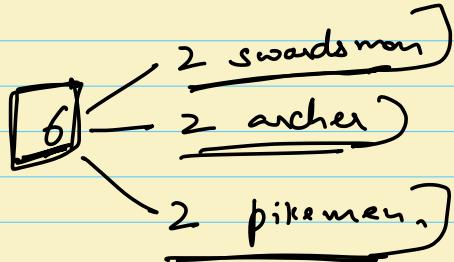
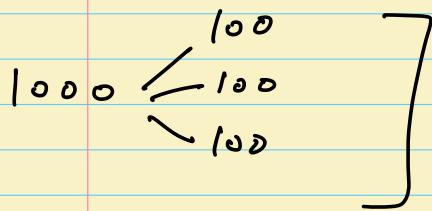
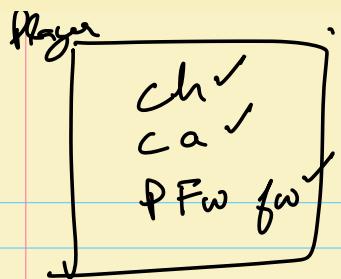


→ If few prop. don't change & are

same for all objects, why are they
not static?

1000 player

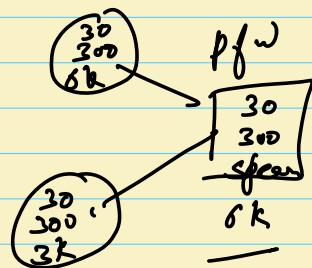
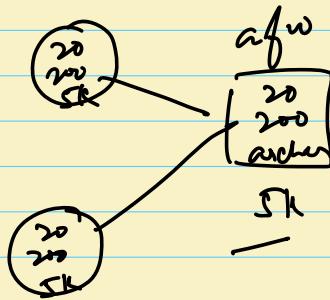
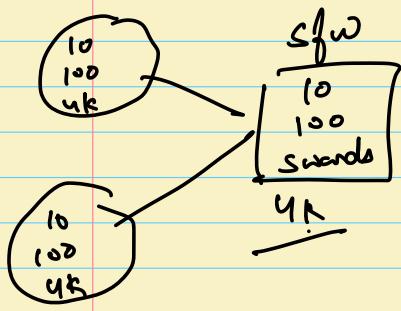
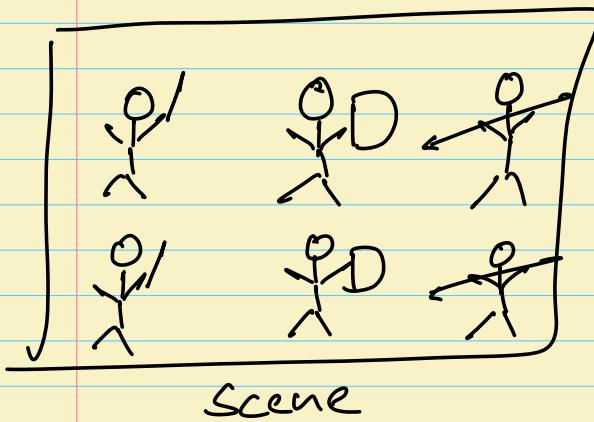




PlayerFW sfw = new PlayerFW(10, 100, swords)
ih ia & avatar.

" afw = new PlayerFW(20, 200, archer)

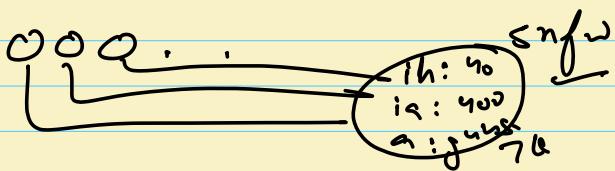
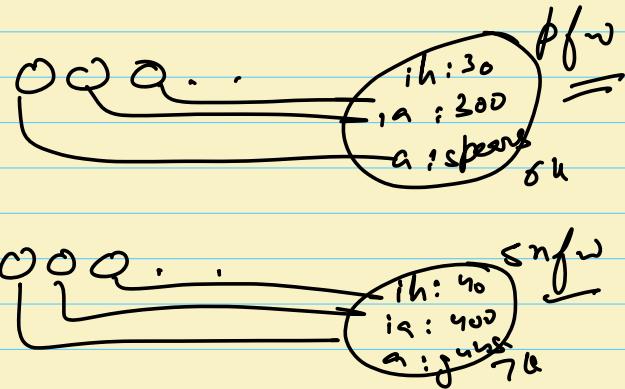
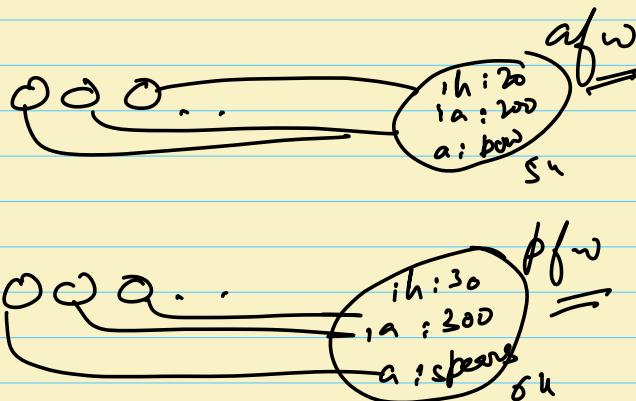
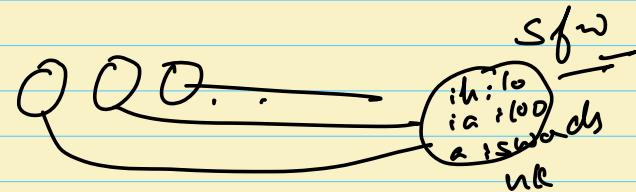
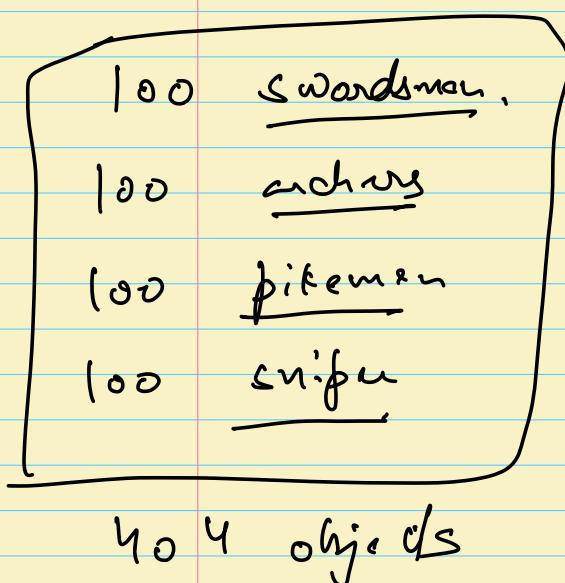
" pfw = new PlayerFW(30, 300, spears)



class

Player FW
→ ih
→ ia
→ quater

class Player Type



How much space is removed?

[Initially 1000 players → 16 GB

1000 players = 100 players of each type
10 types

1000 player objects : $1000 \times 54 = 54000\text{b}$

10 " flyweight objects. $\rightarrow 10 \times 16 \text{ Mb} = 160 \text{ Mb}$

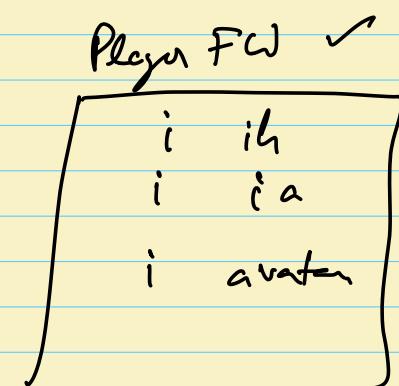
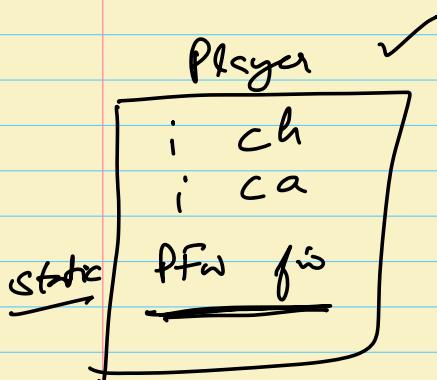


$$54000\text{b} + 160 \text{ Mb}$$

$$54 \text{ kb} + 160 \text{ Mb} \sim 160 \text{ Mb}$$

16 $\leq B$

160 Mb



↑
Player sow1 = new Player()

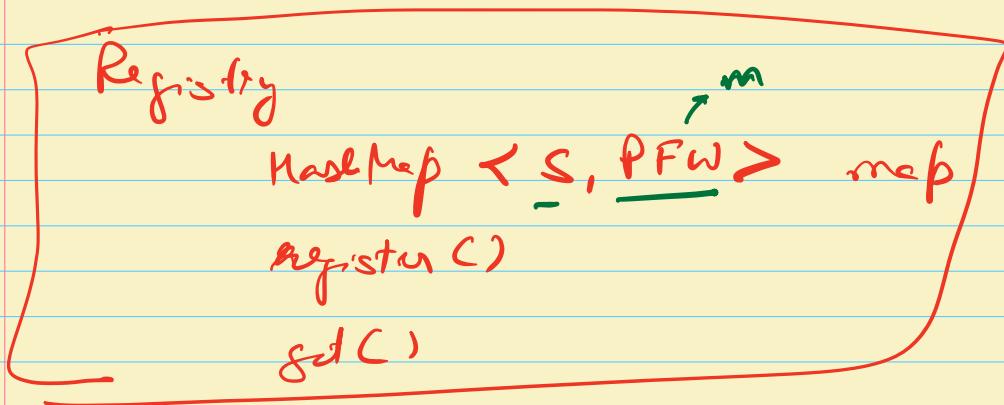
↓
Player sow2 = new Player()

PlayerFW sfw = new PlayerFW(10, 100, sow2)

$$sw1.fw = \boxed{sfw}.$$

$$sw2.fw = \boxed{sfw}.$$

Registry : Flyweight & Registry
can be used together.



Player sw1 =

sw1.fw = Registry.get("swordman fw")

Player sw2 =

sw2.fw :

Break

10:30 - 10:40

Assignment:

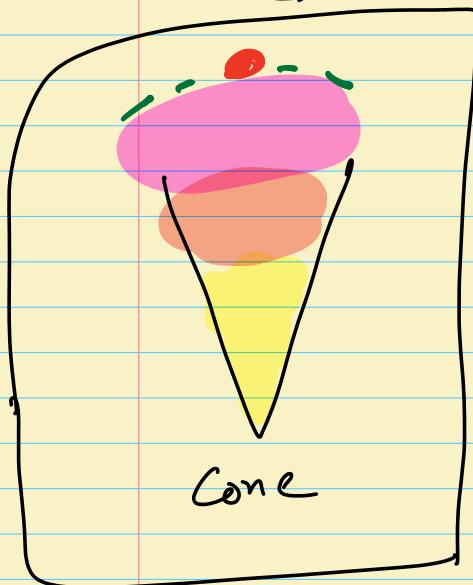
- Code the Flyweight scenario
- Do it with Registry

Decorator Design Pattern.

- ❖ Used to dynamically add responsibilities / decorations to an object at run-time.
- Story
- Code
- Production Level Code.

Story : Icecream Vending Machine

- 1. Help order ice-cream combinations.
- 2. Paint the order.
- 3. Give the cost



Cone + Buttn Scatch
Scoop

+ strawberry
Scoop

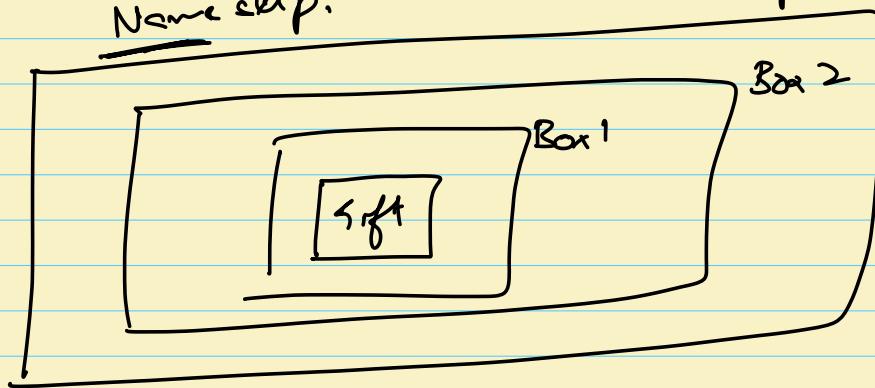
+ Blueberry
Scoop

+ Cherry

+ Choc Chips

Name slip.

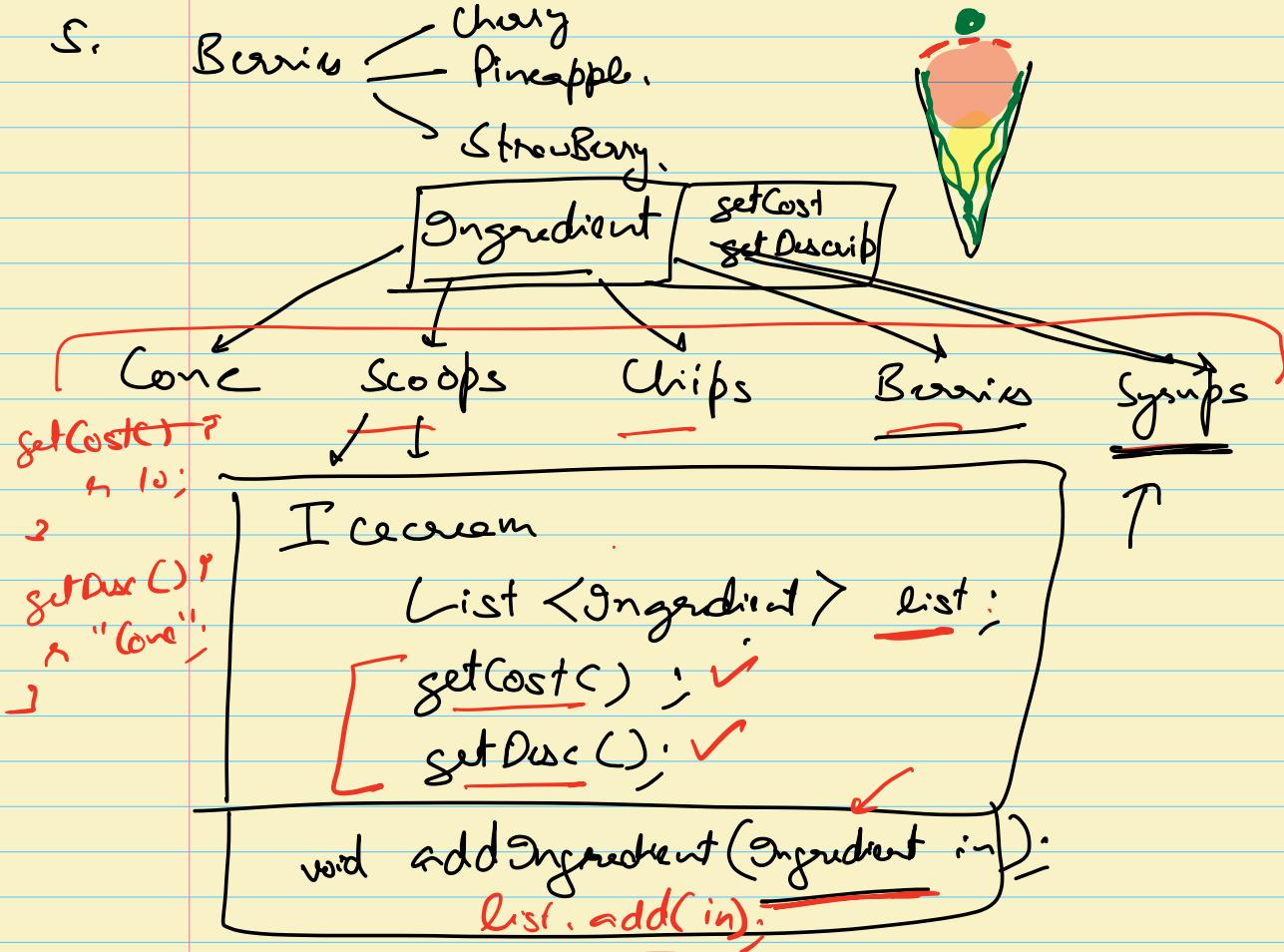
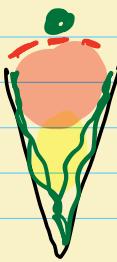
change



1. Cone Orange
 choc.
2. Syrups Chocolate
 mango
3. Scoops Vanilla
 BS
 SB
4. Chips Choc
 Vanille

5. Berries Cherry
 Pineapple
 Strawberry

→ Cost
→ Description



Icecream ic = new Icecream();

ic.addIngredient(new Cone());

ic. addIngredient (new Syrup())

ic. addIngredient (new ChocScoop())

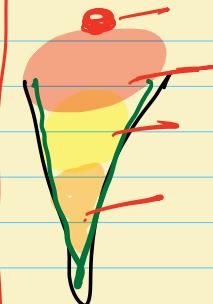
ic. addIngredient (new BSscoop());

ic. addIngredient (new StrawBScoop());

ic. addIngredient (new Cherry());

ic. getCost();

ic. getDesc();



Cone with Choclets

Symp with ChocScoop

with BSscoop with
sBSscoop with cherry.

Icecream

List<Ingredient> list

void addIngredient (Ingredient in)

i getCost() ↗

i c = 0
for (Ingredient in : list) ?
c += in.getCost();

← n c ;

s getDescription() ↗

Similar Code.

Problem Statement : Can an icecream
start with a cherry?

Base : Cone
Ingredients

AddOns : Bonus

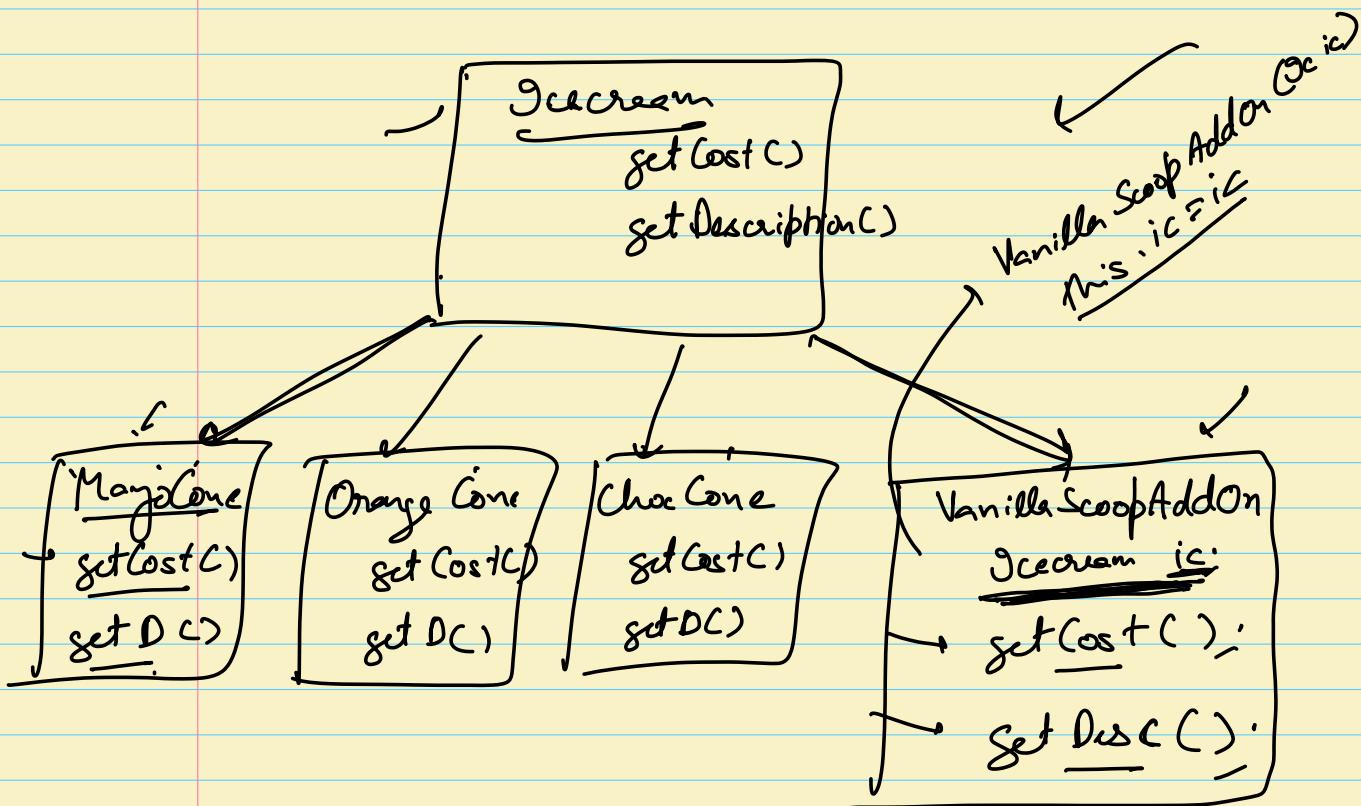
Scoops

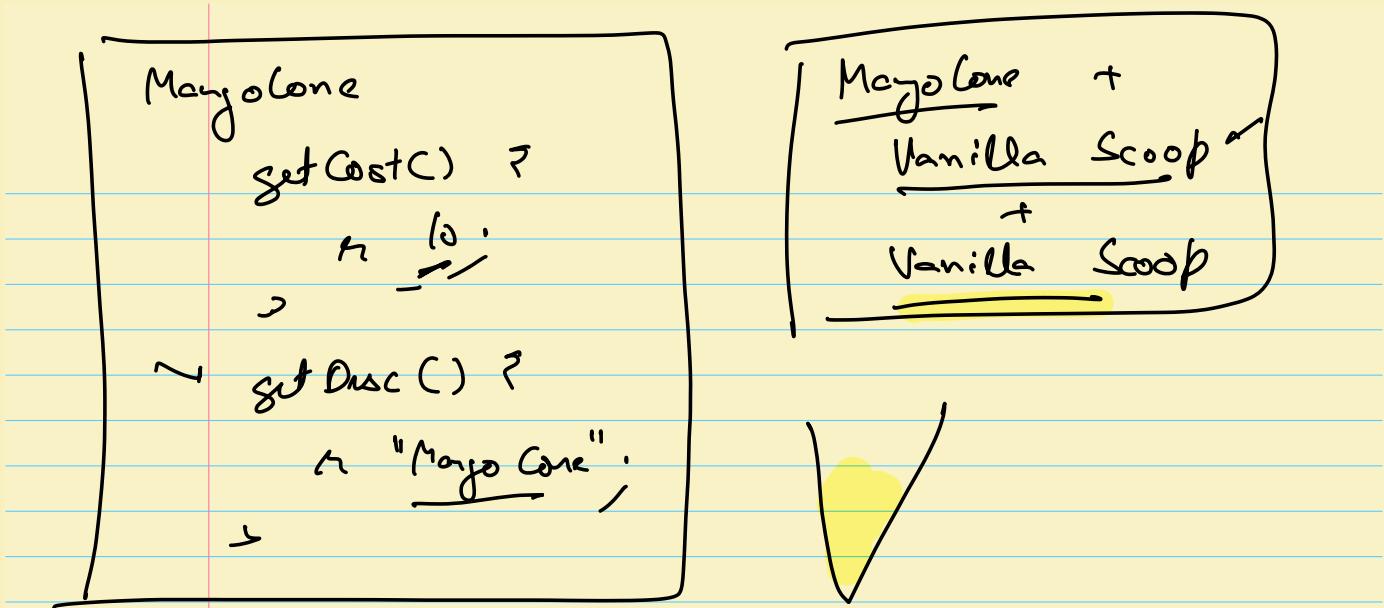
Syrups

Chips.

Icecream Cones

Solution : We start with Cone

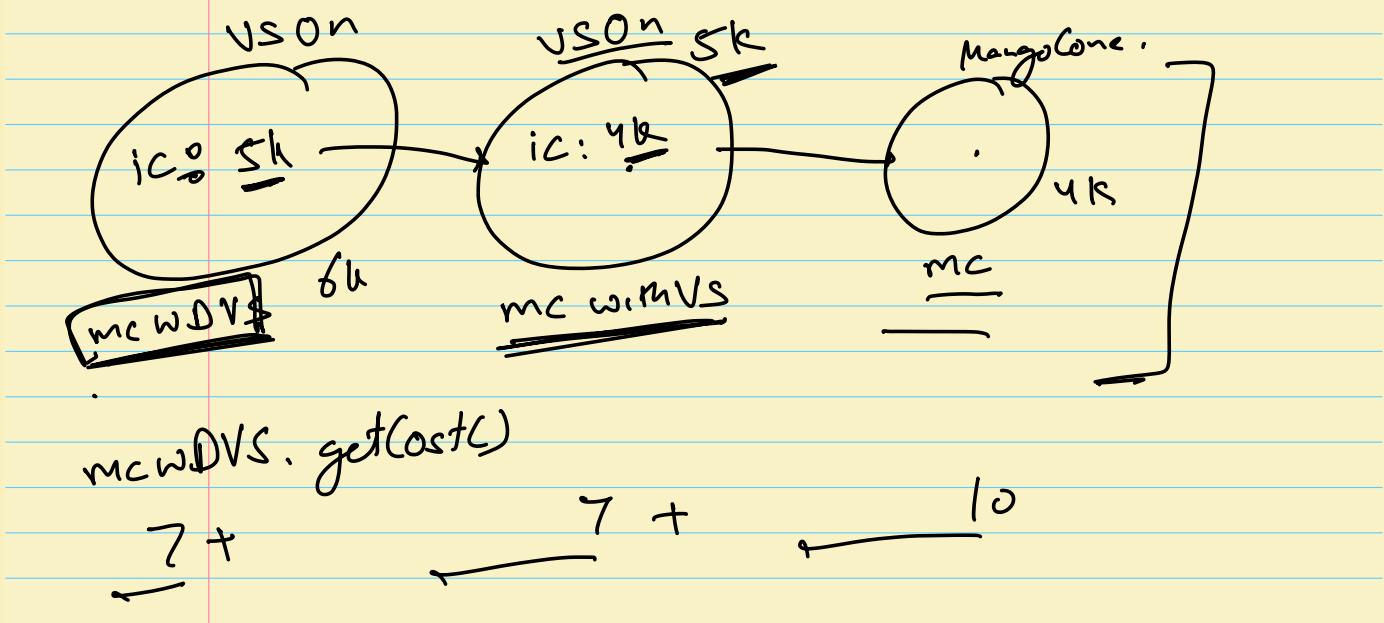




✓ Icecream mc = new MangoCone();

Icecream mc with VS = new VanillaScoopAddon(mc)

Ice cream mc with Double VS =
new VanillaScoop AddOn (mc with VS)



Icecream

Vanilla Scoop Add On

Icecream ic ;

i setCost() ?

return i + ic.getCost();

3

S getDisc() ?

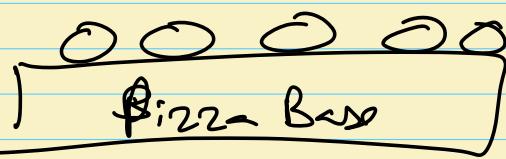
return ic.getDisc() +

"with Vanilla Scoop";

5

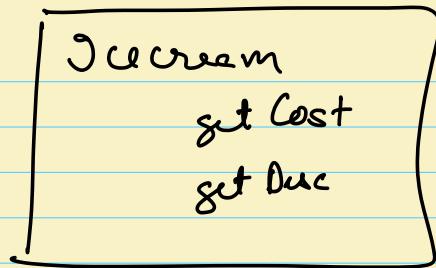


Thin Crust with
jalapenos



15 minutes

2 kind of add ons
Cone Addons



Mango Cone

gC: n 10

gD: n "MC"

Choc Cone

gC: n 12

gD: n "CC"

Vscoop Decorator

Icecream ic

VSDecor(GC ic) ?

this.ic = ic

CScoop Dec

GC ic

→ Ctor

↓

getCost ?

→ getCost

n 7 + ic.getCostC()

^ S' —

↓

getDuc() ?

→ getDuc

n ic.getDuc() +

n ic.getDuc() + "with

"with Vanilla Scoop")

Chocolate Scoop'

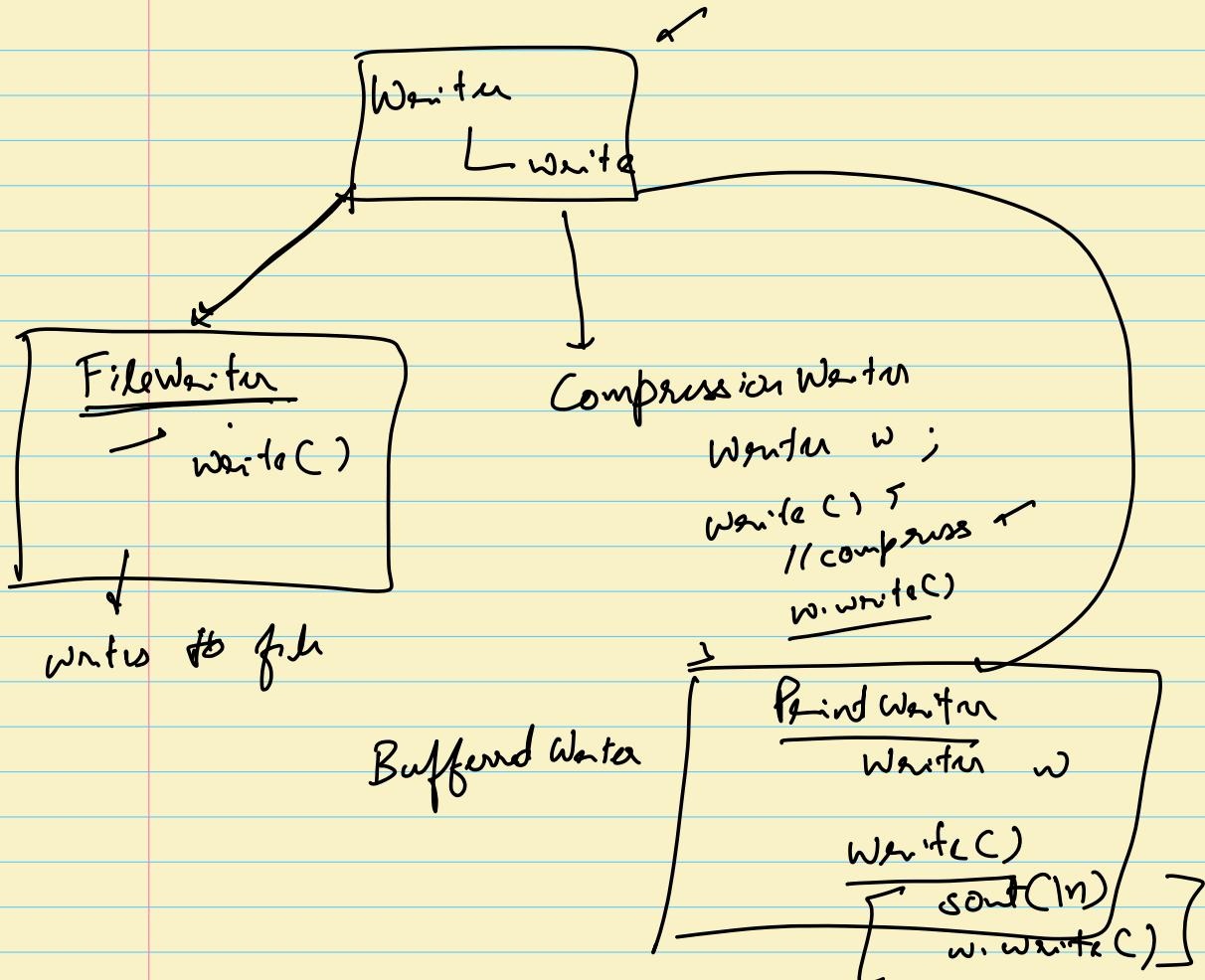
↓

→ Syrup Addons

→ Chip Addons

→ Berry Addons.

Writer [In JOKs]

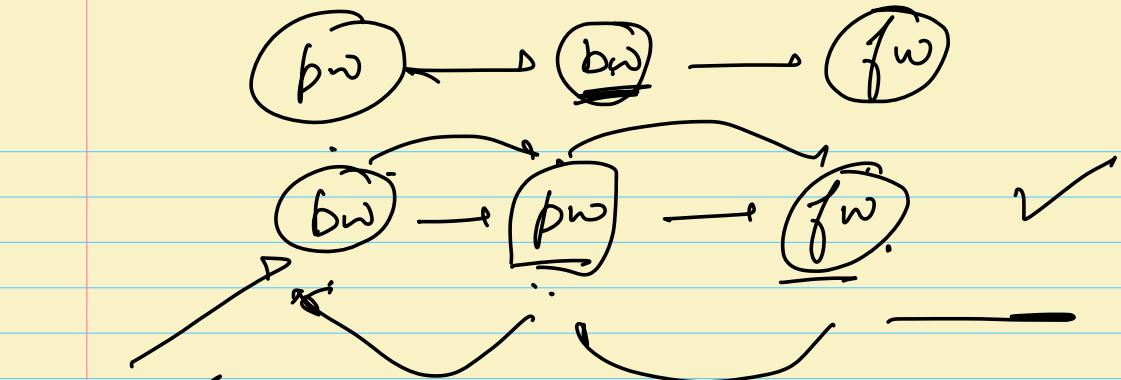


Writer fw = new FileWriter ("file.txt")

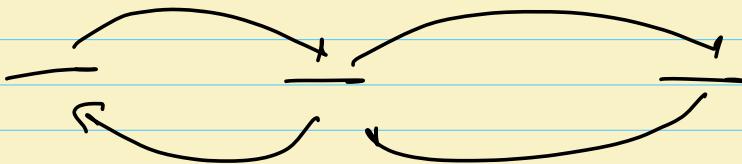
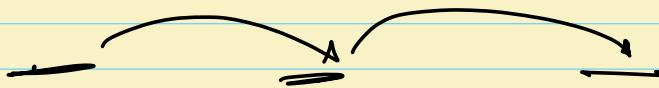
Writer fw = new PrintWriter (fw)

Writer cw = new CompressionWriter (fw)

(cw.write(c))

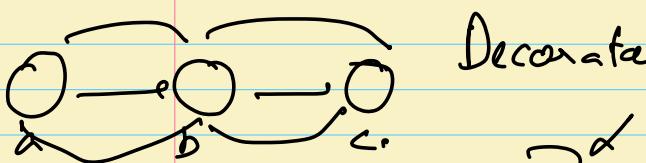


1000 ✓
14 ✓
50 ✓



Flyweight → Saving
space

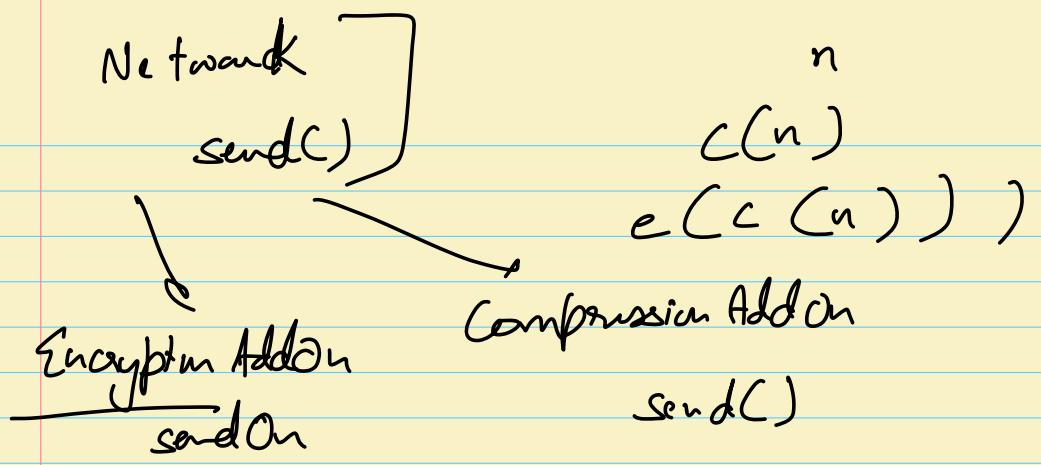
a b c



Decorator → Adding responsibilities dynamically.

w.writeC() ↑
// my work
w.writeC()

w.writeC() ↑
// my work
cba



$C \longrightarrow C \longrightarrow n$

