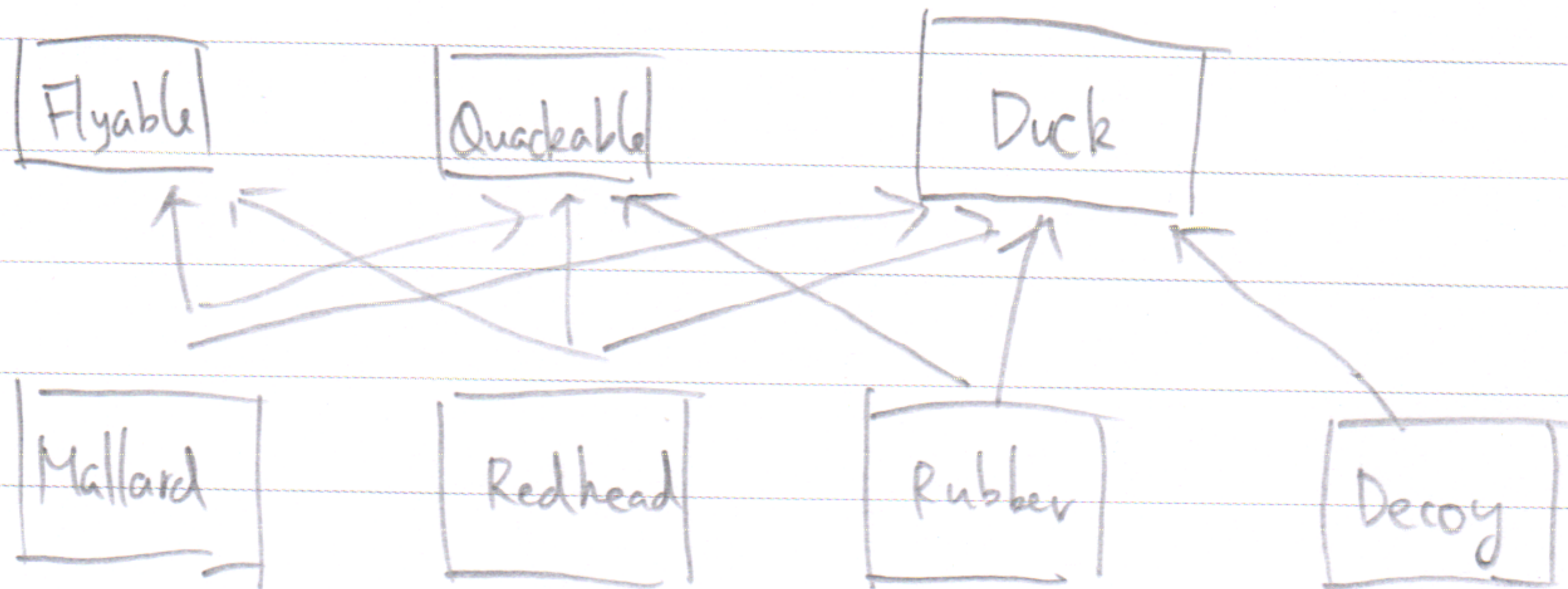


* Duck class	display	quack	fly	
- Mallard Duck	(green head)			implements Quackable Flyable
- Redhead Duck	(red head)			"
- Rubber Duck	(yellow)	squeak		x Implements Quackable
- Decoy Duck	(colorful)	x		x



* Principle :

Identify what varies and separate them from what stays the same

* class Duck {

```

    Flying Behavior fb ;
    Quacking Behavior qb ;

    void display () {
    void performFly () {
    void performQuack () {
    }
    }
    }
    
```

) what varies (Behavior can be extended and set flexibly)
- same
) "almost" same

* Principle :

Favor composition over inheritance
Program to interface: has a behavior is a flyable duck

* strategy pattern : encapsulate some "isolated" and "interchangeable" part about an object so the objects can "freely" change it

Subject :

* use of strategy pattern

- ① Layout Manager for Containers
- ① POOSkill for POOPet (?), not currently)
- ② PlayStrategy for Poker Player
- ③ SpeedupStrategy for Car

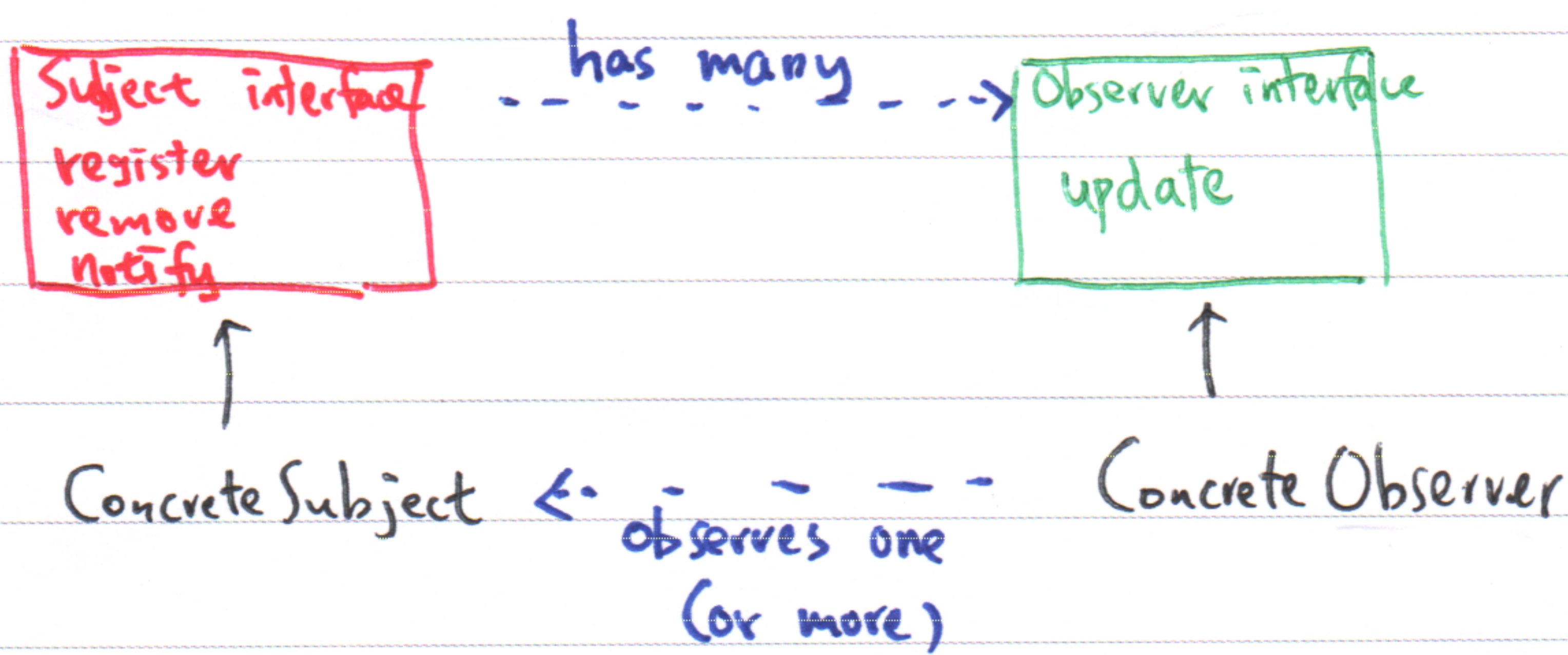
* Weather Sensors
↑ pull

Subject Weather Data getTemp, getHumid, getPressure measureChanged
↓ push
Observer's "Weather Display's"

* Observer : "subscribe" to Subject's changes
Subject : "notify" Observers the changes in measureChanged

One to many

"observer pattern"



"loosely coupled objects"

* class java.util.Observable
addObserver
deleteObserver
{ notifyObservers
& setChanged

interface java.util.Observer
update

Subject :

* use of observer pattern

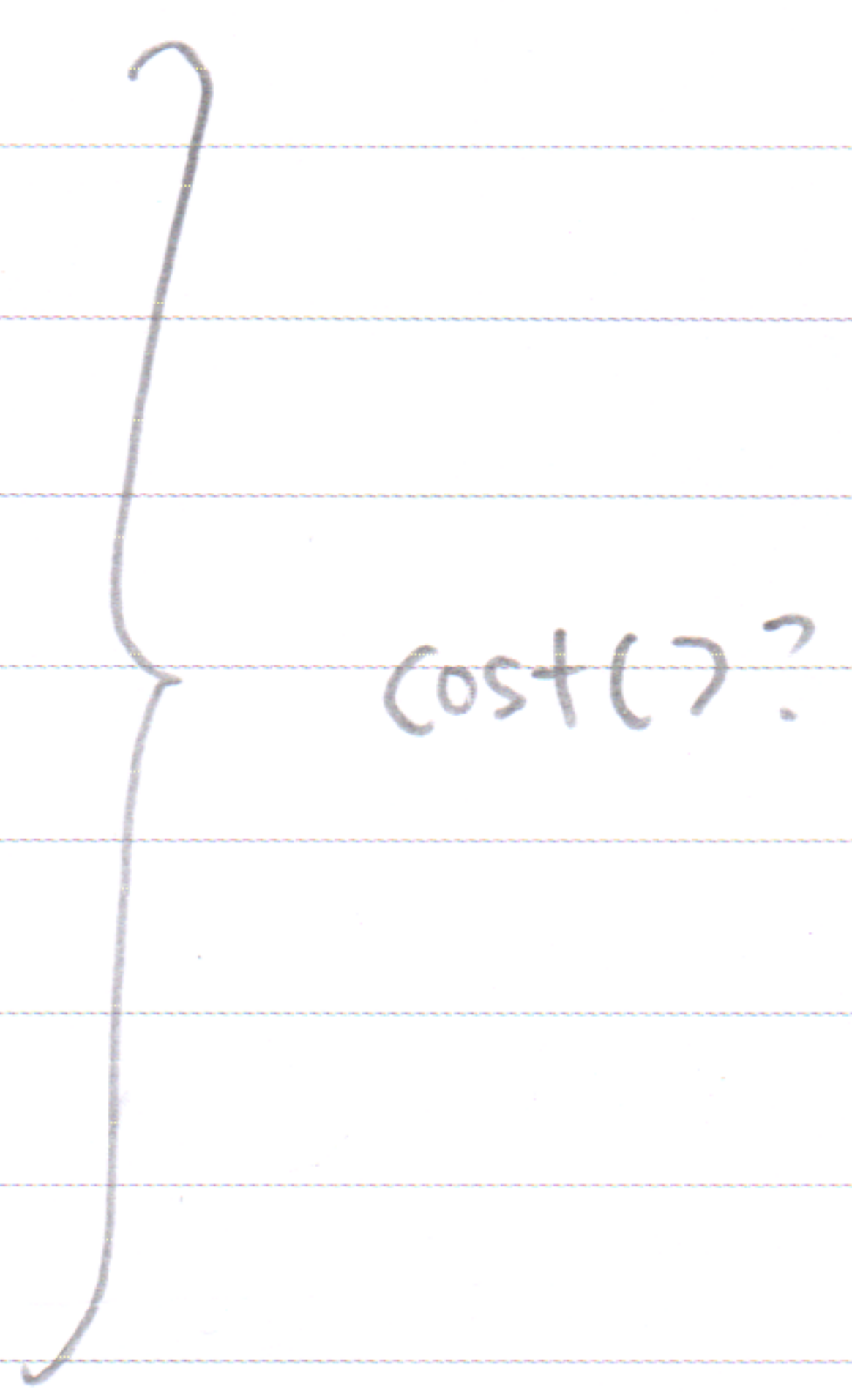
- ① ActionListeners for Component
- ① Cars (front) for Car (back) ?
- ② POOPets for POOArena ?

* Be careful

- ① order of evaluation
- ① Observable "class"

* Coffee

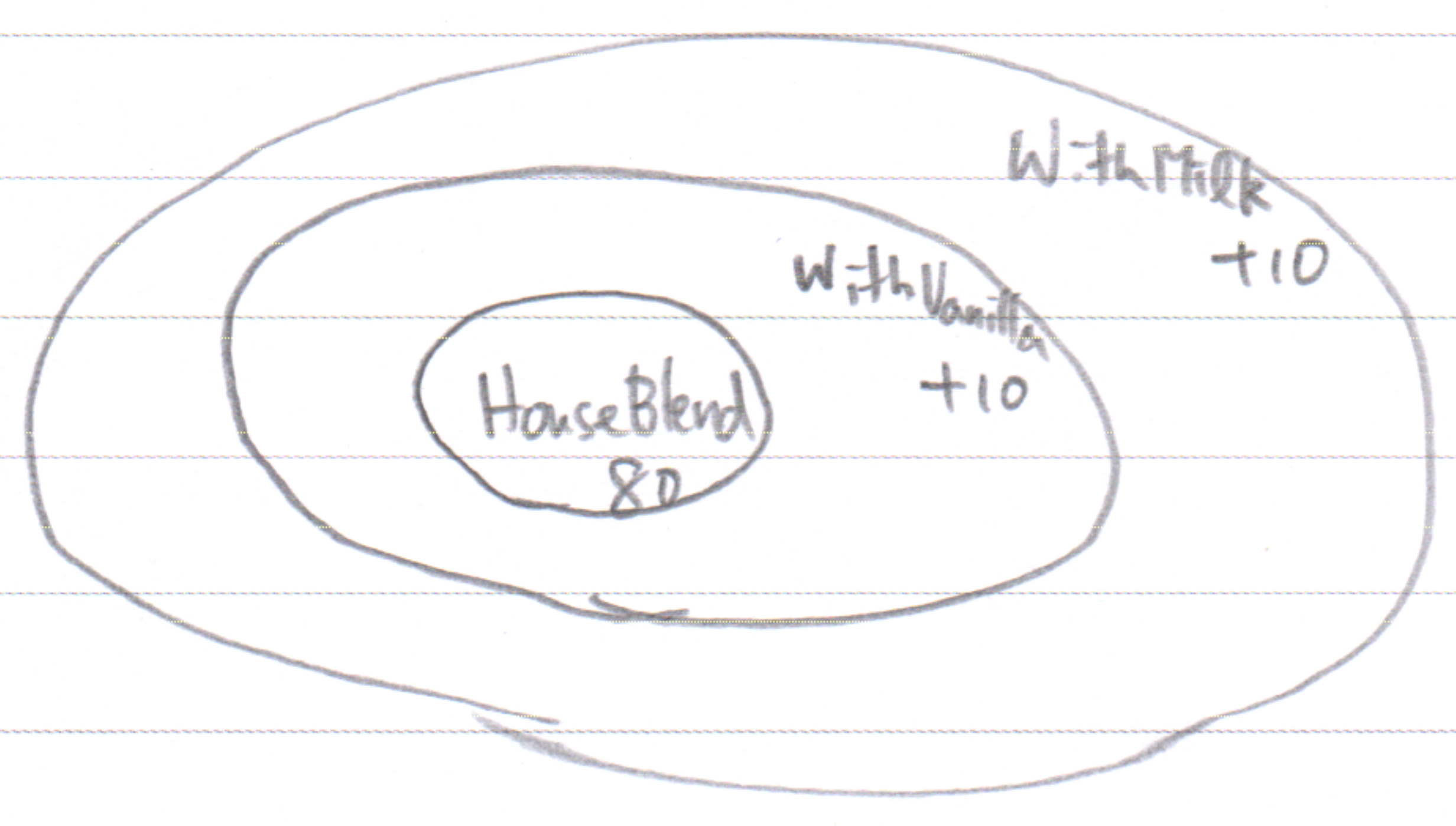
- House Blend
- Decaf
- Espresso
- House Blend With Milk
- Decaf With Milk
- House Blend With Vanilla
- House Blend With Vanilla And Milk
- ...



* a generic "cost" in Coffee? hard to extend
override "cost" for each class? repeated efforts
hard to modify

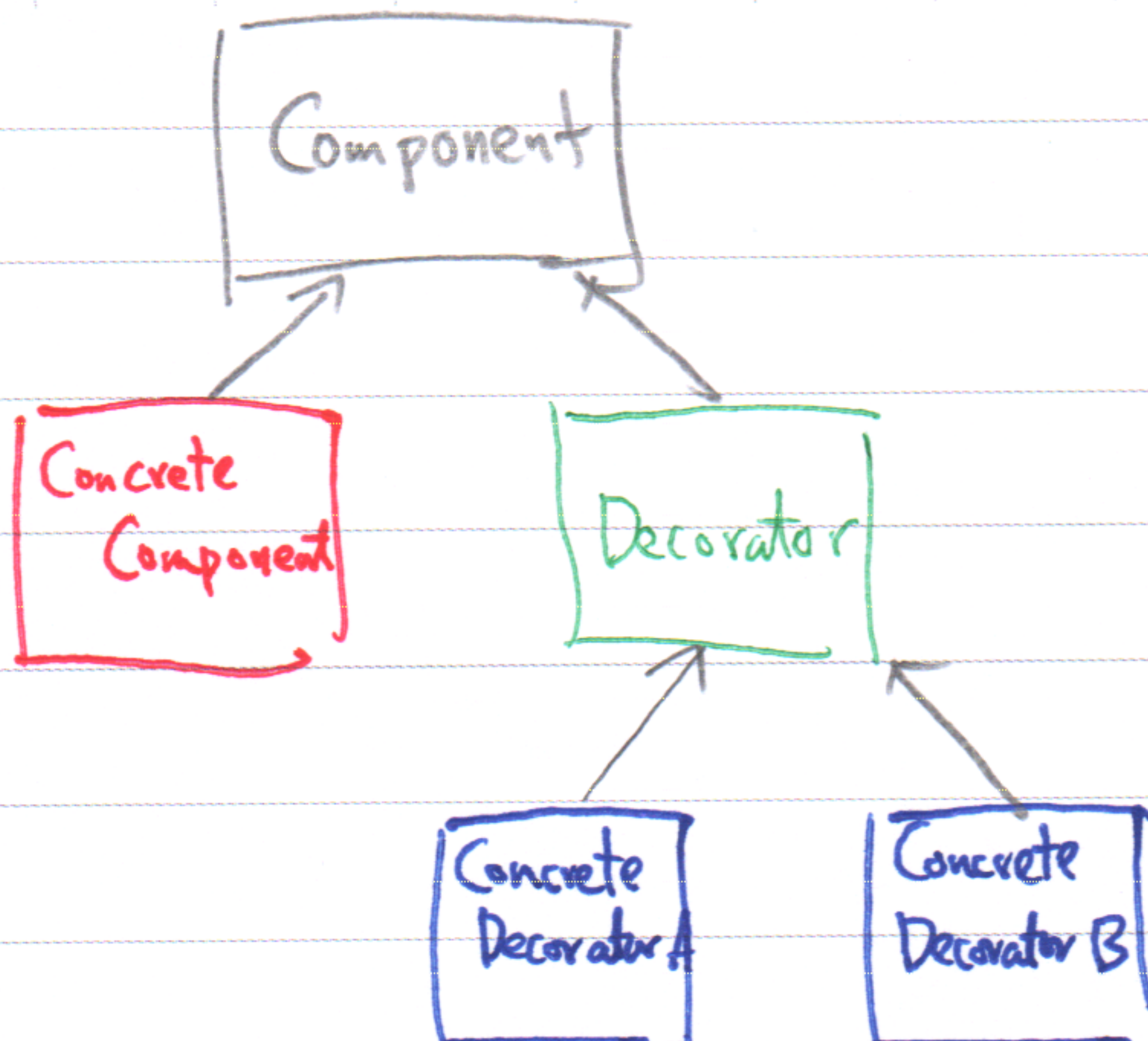
* open-closed principle: open for extension
closed for modification

* decorate by wrappers



Subject:

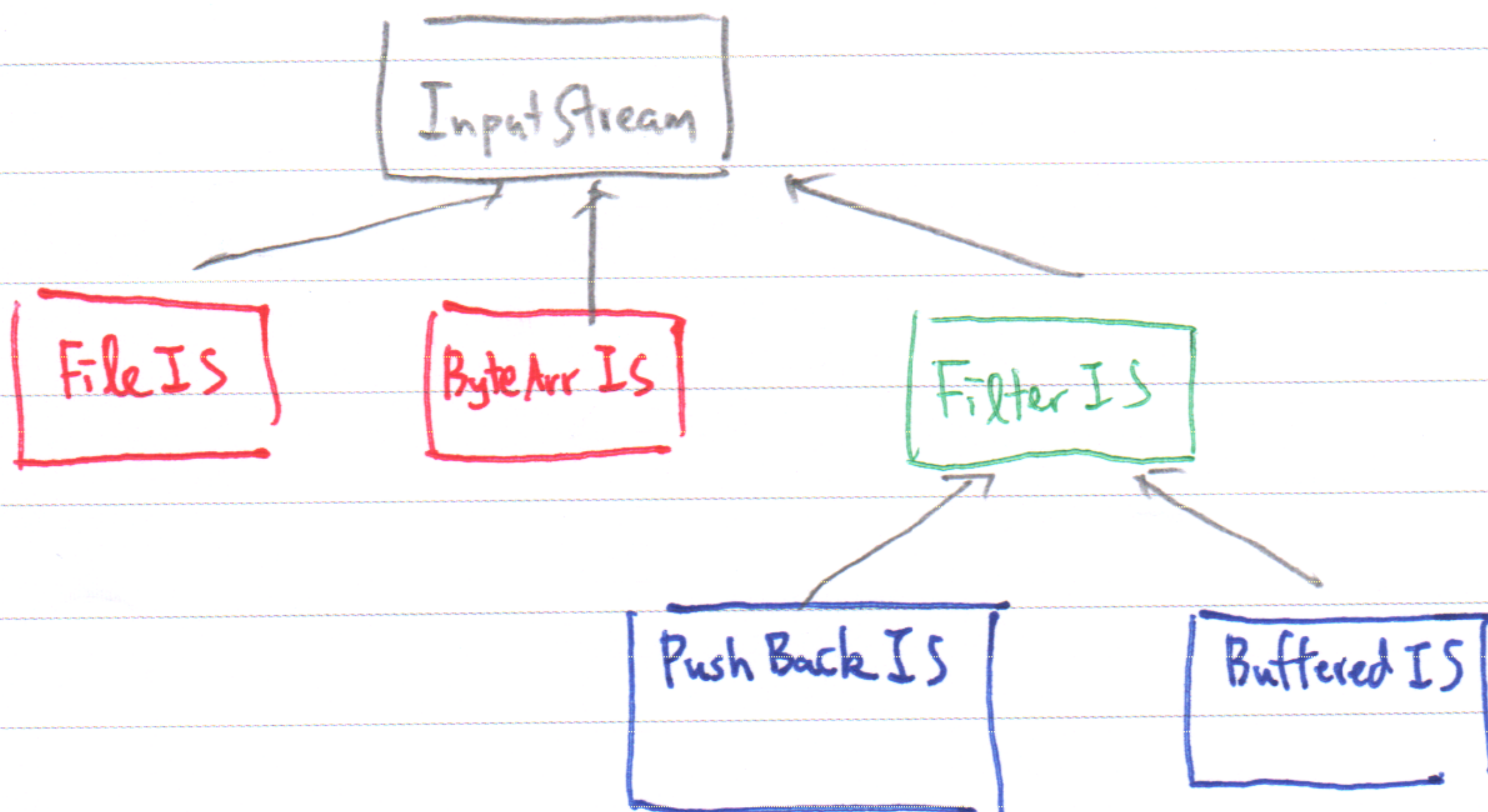
*



new ConcreteDecoratorA (new ConcreteComponent())
open for extension closed

* use of decorator pattern

①



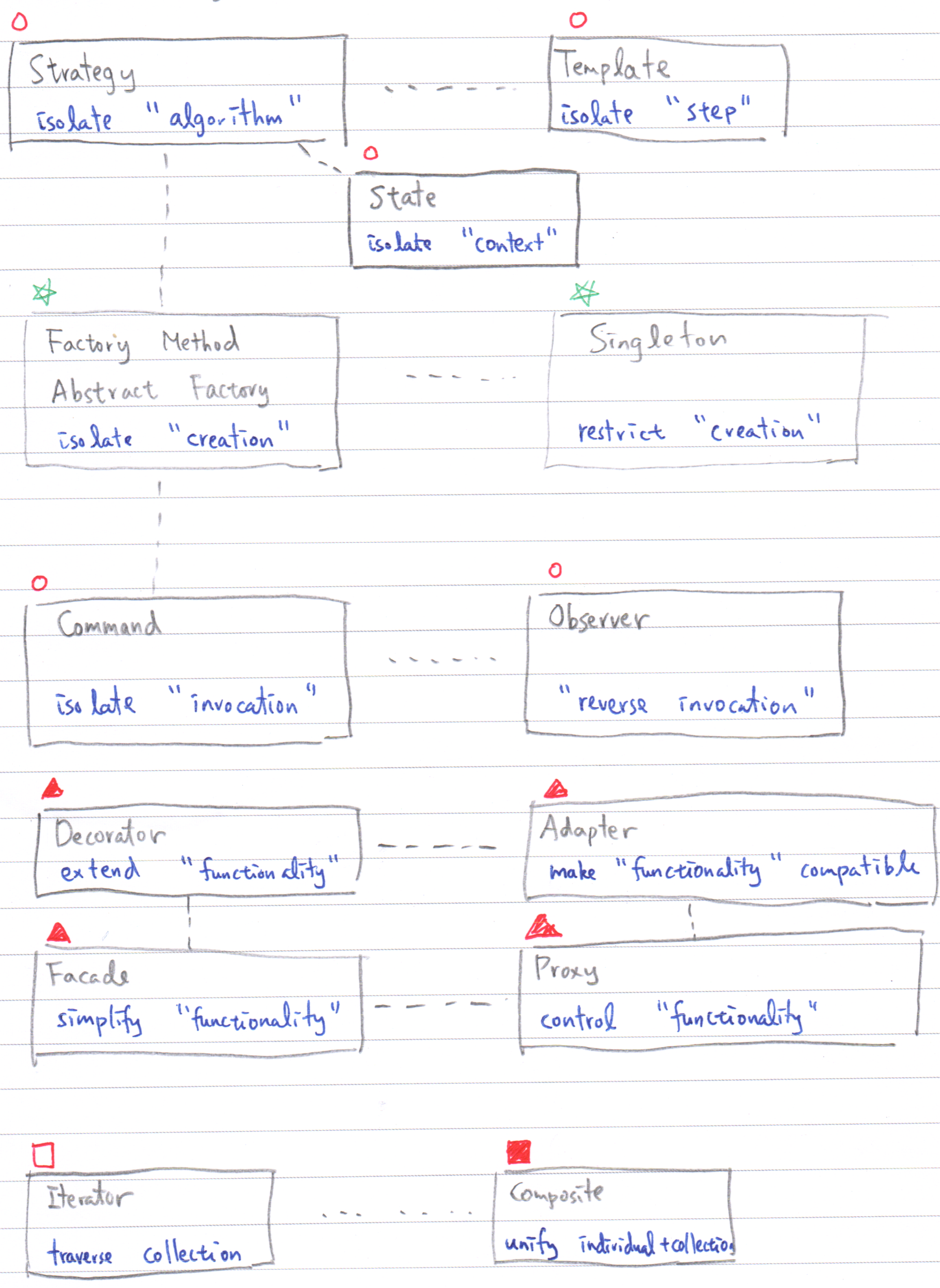
① IS, O/P for all the ...

* be careful:

① "slow" ?

Subject :

* Forest of Design Patterns (personal view)



☆ : Creational
○, □ : Behavioral
▲, ▣ : Structural