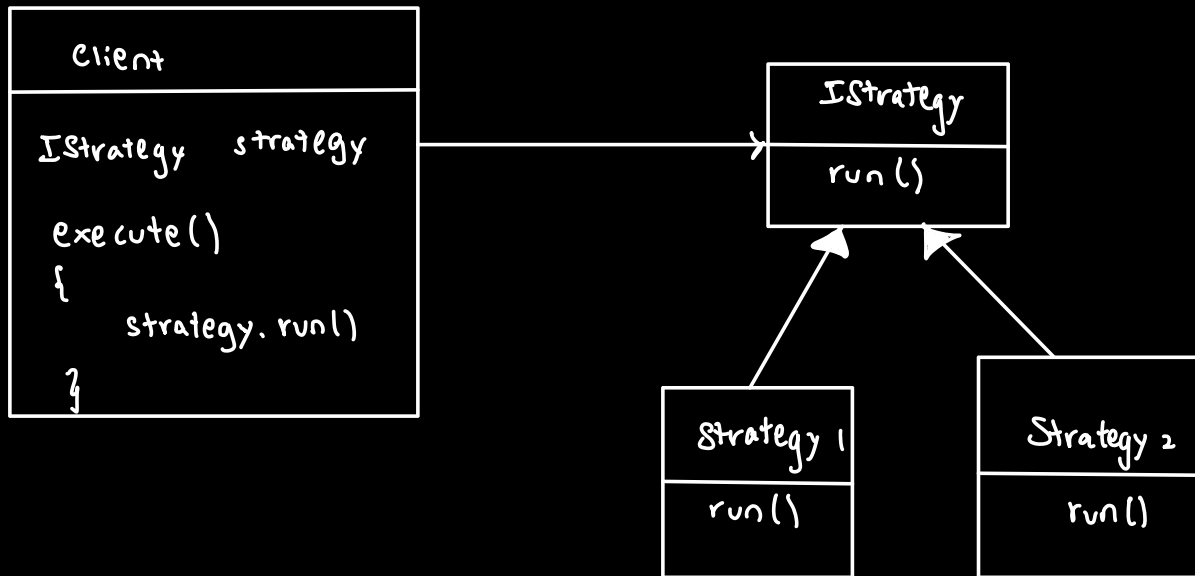


STRATEGY PATTERN: [BEHAVIORAL]

Use composition instead of inheritance.



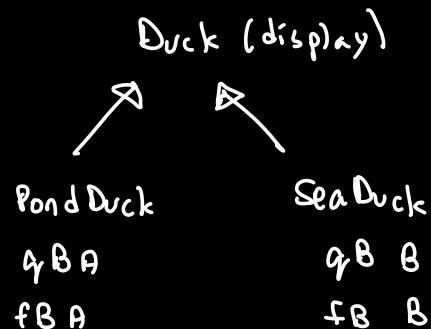
Example:

Ducks can be displayed, can quack and can fly.

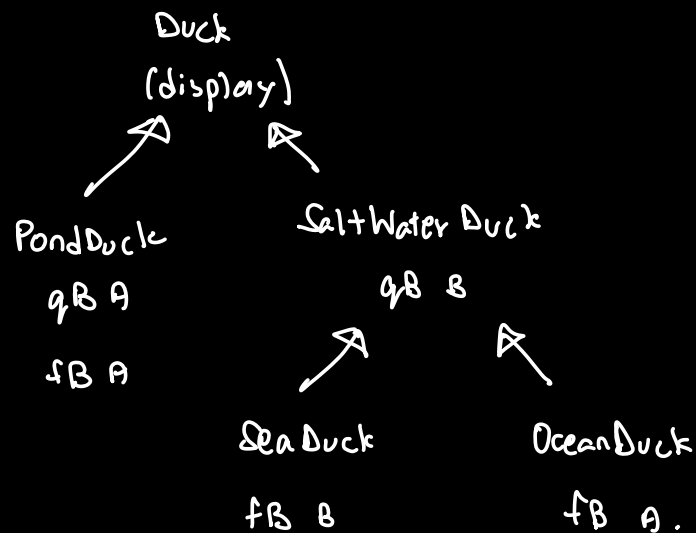
Pond Duck have displayBehavior A, quackBehavior A, Fly Behavior A

Sea Duck have dB A, qB B, fB B

We can use inheritance



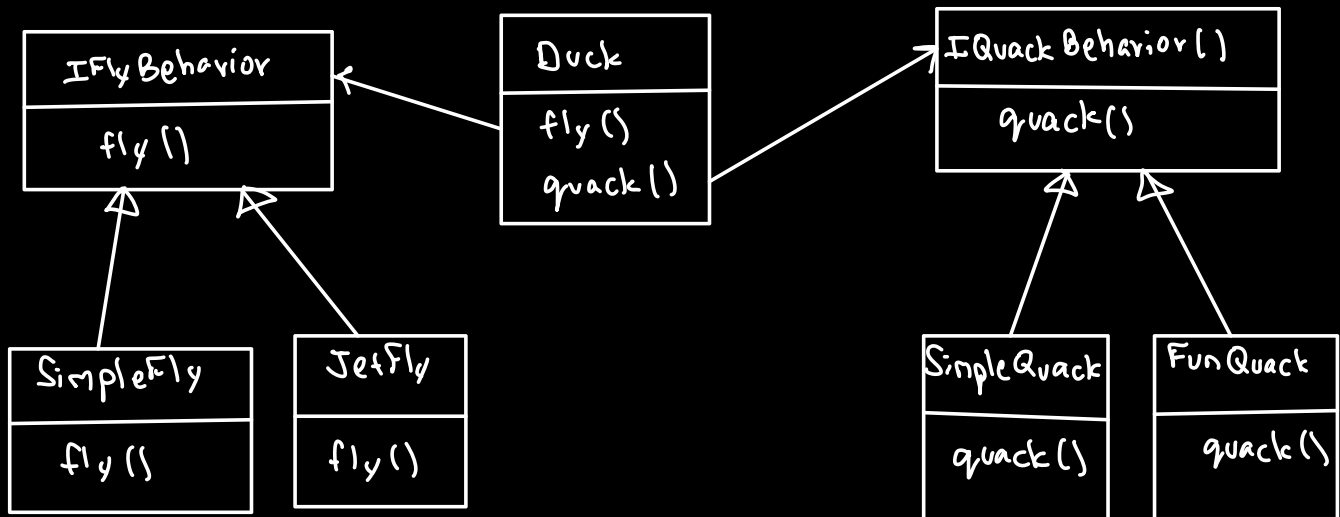
What if OceanDuck comes with all same as
SeaDuck except fB which is A.



If there are 3 types of each behavior, there
are 27 possible types of ducks.

Inheritance is not a good
solution!

USE STRATEGY PATTERN



```

class Duck
{
    IFlyBehavior fb ;
    IQuackBehavior qb ,

    public Duck ( IFlyBehavior fb , IQuackBehavior qb)
    {
        this.fb = fb ;
        this.qb = qb ;
    }

    public fly ()
    {
        this.fb.fly () ;
    }
} ;
    
```