Strategy Pattern

Used when there are a set of algorithms

Each algorithm is encapsulated.
The algorithms are interchangeable.

The user don't have to change if the algorithm is changed.

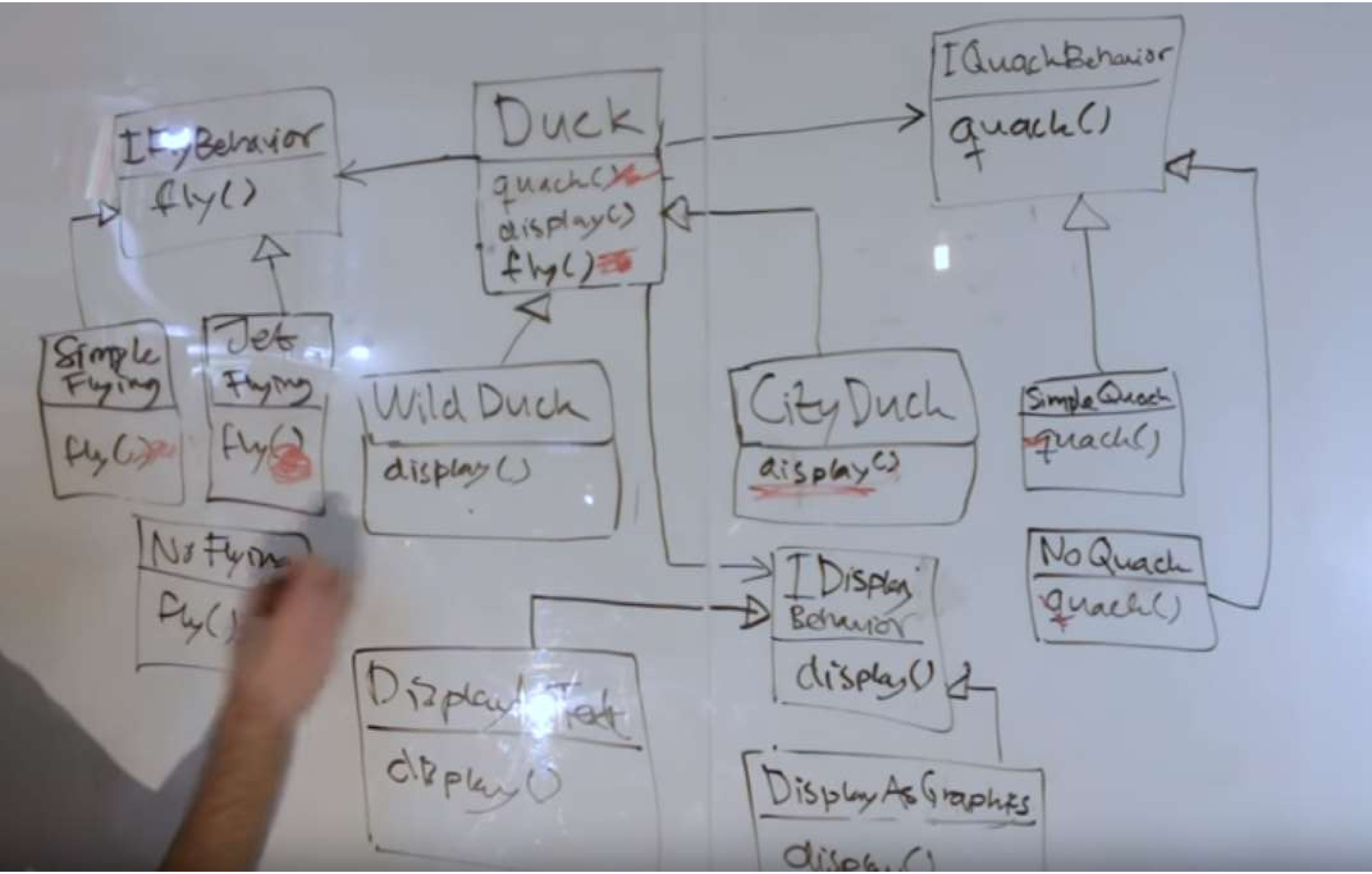
One example use-case is when you want to share the implementation of a method in sibling classes.

We create a strategy for each method

So the base class instead of having methods, has strategies/behavior

The strategies are interfaces

The derived class now can cherry pick the different behaviors it want to implement


IFlyBehavior
fly()
Duck
quack()
display()
fly()
IQuackBehavior
quack()
Simple Flying
fly()
Jets Flying
fly()
Wild Duck
display()
City Duck
display()
Simple Quack
quack()
No Flying
fly()
No Quack
quack()
IDisplay Behavior
display()
DisplayAsText
display()
DisplayAsGraphics
display()