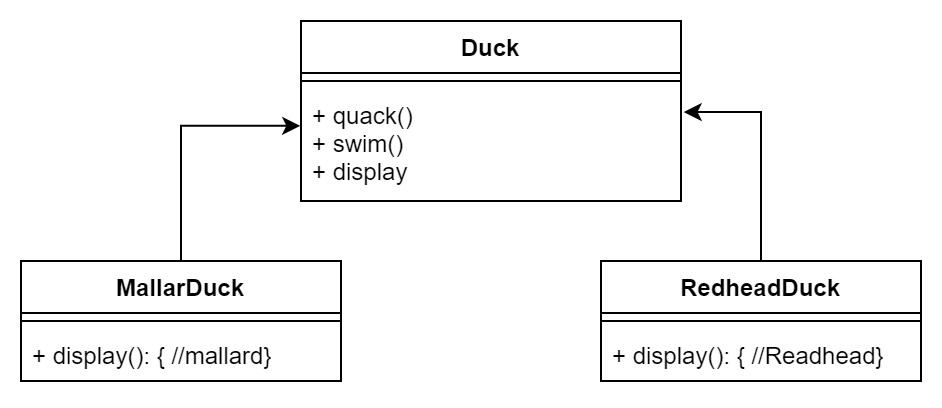
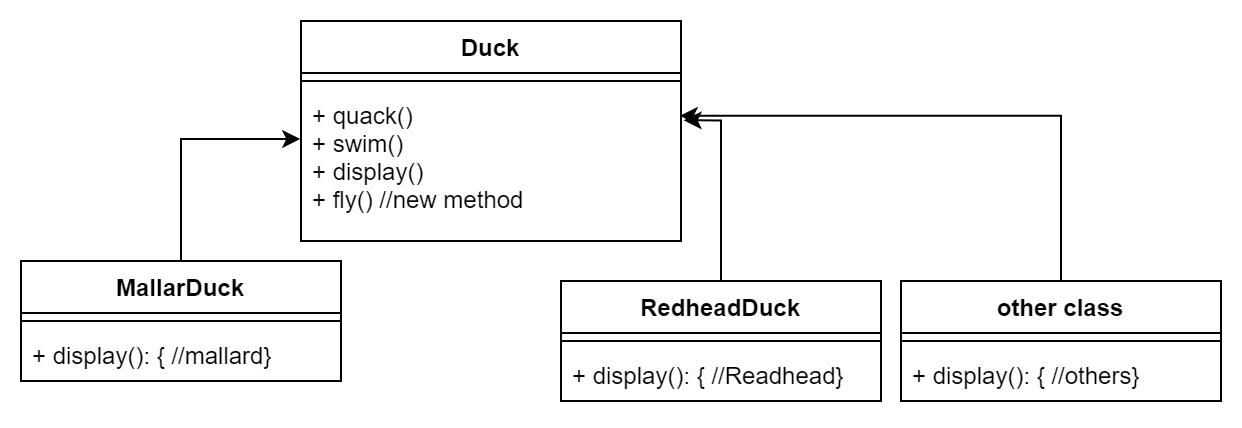
1. Design Pattern cơ bản 1

* Khởi đầu với 1 chương trình game với nhân vật là các con vịt (Duck)
* Joe dùng pattern OOP thiết kế diagram như sau



* Nhu cầu công ty cần thêm tính năng bay -> Joe quyết định thêm phương thức fly() vào class Duck

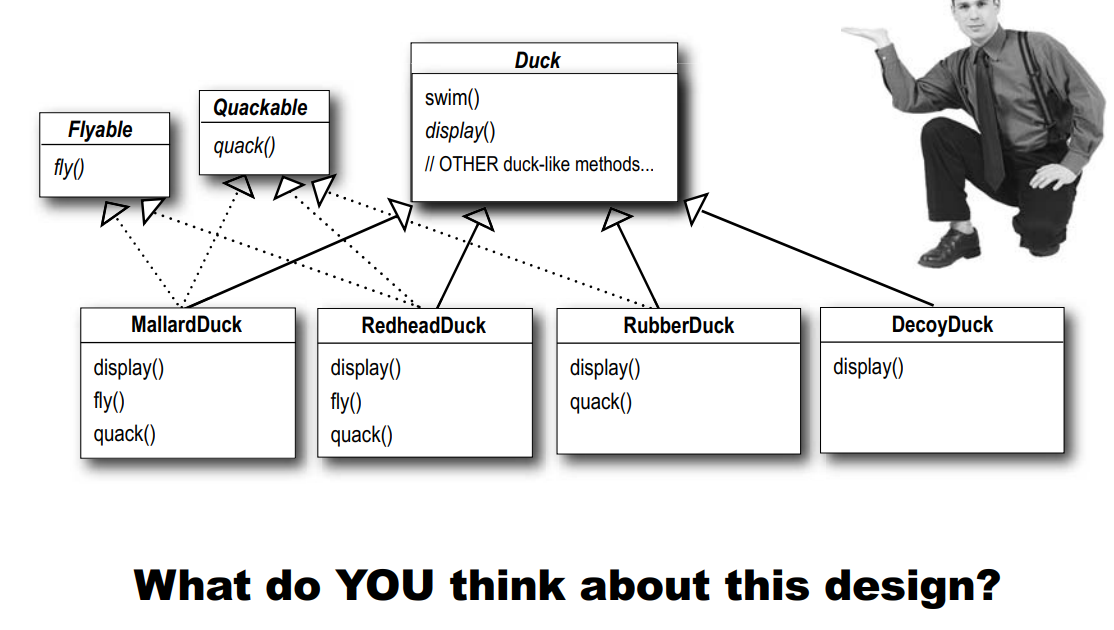


* Nhưng vấn đề xảy ra khi thêm class RubberDuck (vịt cao su):

+ quack() => squeak() ? không kêu như bình thường

+ fly() => không thể bay?

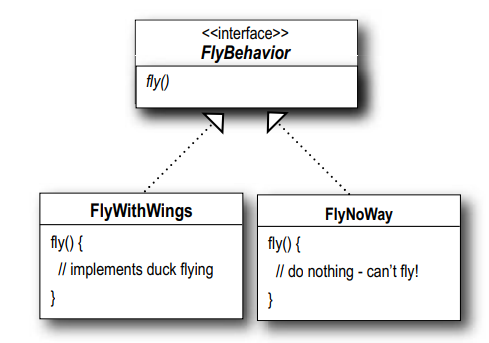
* Cần phải verride lại cái method này
* Khi thêm 1 class DecoyDuck() (vịt gỗ) => vẫn phải override lại quack() và fly()
* Cách này làm cho code không thể REUSE
* Joe quyết định tách các method quack(), fly() thành các interface

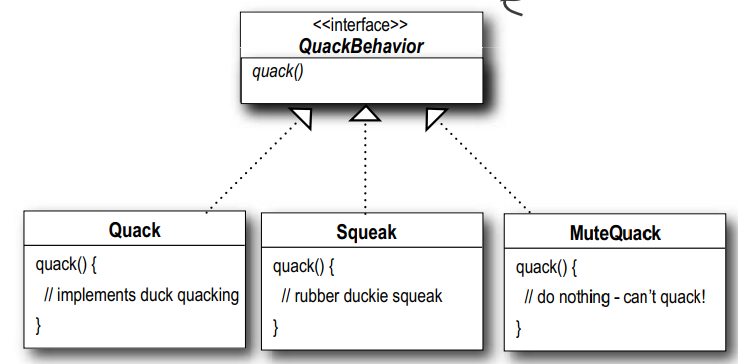


Ghi chú:

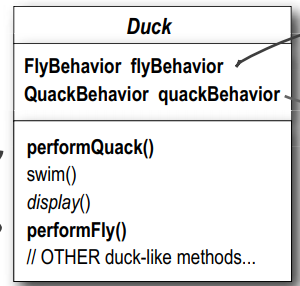
* - - - - - ->: implement các interface
* 🡪 kế thừa từ class Duck
* Vấn đề xảy ra khi có quá nhiều class cần implement Interface Fly và Quack
* Code lại không thể REUSE là bị DUPLICATE quá nhiều lần
* Một hướng đặt ra:

+ Tách các phương thức thành các class => Capsulate chúng lại => dung lại khi cần





+ Duck giờ sẽ ủy thác các interface Fly, Quack



public class Duck {

QuackBehavior quackBehavior;

public void performQuack(){

quackBehavior.quack();}}

Diagram đầy đủ

