

**My Notes for Design Pattern**

Contents

[1. Strategy Pattern 2](#_Toc155737570)

[What it is? 2](#_Toc155737571)

# Strategy Pattern

## What it is?

**Strategy** is a behavioral design pattern that lets you define a family of algorithms, put each of them into a separate class, and make their objects interchangeable.

## How to Detect?

Basically see if a specific task is being solved by different methods.

## Problem

I made a middleware which let me authenticate user using “email and password” and then I added a feature to authenticate user using “Oauth” too and then another feature of inserted to authenticate user using “Pass code”… This causes [Authenticate class] to have double code each time another method of authentication is added.

## Solution

Make a class Authentication Context and create a member named authStrategy which is instance of an interface

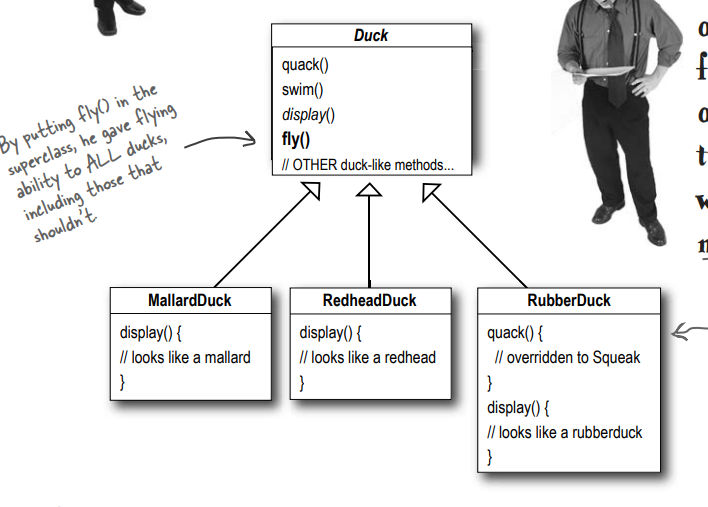
## Other Point of Views

Using of Composition and Interfaces for inheritance instead of Inheritance from Class.

“The Solution to Problems with Inheritance, is certainly not more inheritance”

## Duck Problem

Problem



Solution:

