Zach Williams

Multiple Inheritance

The problem is that there are going to be instances where there are several classes with common behaviors. One class may have common behaviors with another class but not another. The diagram will explain what I mean.

Cruiser

Sports Bike

Towing

Sedan

Car

Sports Car

Sports Boat

Cruiser

Motorcycle

Truck

Watercraft

Vehicles

There are common behaviors of each vehicle; start, shut\_off, accelerate, brake.

As we can see though, there are three types of sports vehicles that will share common behaviors but other classes won’t.

* For instance: Sports Boat, Sports Car, Sports Bike all have Nitrous in them and is activated by some type of behavior
* Cruiser for both Watercraft and Motorcycle don’t have Nitrous nor do Sedans.

Because our Sports Vehicles all share a common behavior but also share common behaviors with another class as well, we have a multiple inheritance issue. We could create a Sports Vehicle class and then have a classification between Boat, Car, Motorcycle, but then we run into the issue of having all other watercraft characteristics, car characteristics, motorcycle characteristics, etc. being inherited among into the sports vehicle class so this idea is not plausible. Refactoring could be done but you’d waste a lot of time and the code base would have several repeat common behaviors therefore it wouldn’t be DRY.

This is our problem in terms of multiple inheritance. We then solve this issue by mixing in modules. We could create a module for the example above called `module Sportable` and share the common characteristics of nitrous with the three different types of sports vehicles while still inheriting the more common features of each vehicle from the parent class (Watercraft, Car, Motorcycle).